

Dinámica de los Sistemas Físicos.
Notas de Clase – Año 2024

Cátedra de DSF

Coordinación: Ernesto Kofman

Departamento de Control
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Universidad Nacional de Rosario

Índice general

Introducción	1
1. Sistemas Dinámicos y Modelos Matemáticos	3
1.1. Sistemas Dinámicos	3
1.1.1. El Concepto de Sistema	3
1.1.2. Clases de Sistemas	4
1.2. Modelos Matemáticos	4
1.2.1. El Concepto de Modelo	4
1.2.2. Clases de Modelos	4
1.2.3. Clasificación de Modelos Matemáticos	5
1.2.4. Elementos de los Modelos Matemáticos	6
1.3. Representación de Modelos Matemáticos	7
1.3.1. Ecuaciones de Estado	7
1.3.2. Ecuaciones Diferenciales Algebraicas	9
1.4. Utilización de Modelos Matemáticos	10
1.4.1. Análisis Teórico de los Modelos Matemáticos	10
1.4.2. Simulación	10
1.4.3. Limitaciones al usar Modelos Matemáticos	11
1.5. Construcción de Modelos Matemáticos	11
1.5.1. Modelado por Primeros Principios	11
1.5.2. Modelado por Identificación	12
1.5.3. Modelado de Caja Gris	13
1.6. Herramientas de Software para Modelado y Simulación	13
1.7. Bibliografía Complementaria	14
1.8. Problemas Propuestos	14
2. Simulación de Sistemas Continuos	16
2.1. Principios de la Integración Numérica de ODEs	16
2.1.1. Métodos de Euler	18
2.1.2. La Iteración de Newton	20
2.1.3. Precisión de las Aproximaciones	20
2.1.4. Estabilidad Numérica	22
2.2. Métodos Monopaso	25
2.2.1. Métodos Explícitos de Runge Kutta	25
2.2.2. Métodos Monopaso Implícitos	27
2.2.3. Algoritmos de Control de Paso	27
2.3. Métodos Multipaso	30
2.3.1. Métodos Multipaso Explícitos	30
2.3.2. Métodos Multipaso Implícitos	31
2.3.3. Control de Paso	31
2.4. Algunos Casos Problemáticos	31
2.4.1. Sistemas Stiff	32

2.4.2.	Sistemas Marginalmente Estables	34
2.4.3.	Sistemas con Discontinuidades	34
2.5.	Simulación de DAEs	35
2.6.	Implementación de los Algoritmos en Software	37
2.7.	Bibliografía Complementaria	38
2.8.	Problemas Propuestos	38
3.	Causalización de Sistemas de DAEs	41
3.1.	Ordenamiento y Causalización de Ecuaciones	41
3.1.1.	Modelos Implícitos y Explícitos	41
3.1.2.	Algoritmo de Ordenamiento Horizontal y Vertical	44
3.1.3.	Eliminación de Ecuaciones Triviales	47
3.1.4.	Obtención de las Ecuaciones de Estado	47
3.2.	Lazos Algebraicos	48
3.2.1.	Ejemplo Introductorio	48
3.2.2.	Algoritmo de Rasgado	49
3.2.3.	Obtención de las Ecuaciones de Estado	51
3.3.	Sistemas de Índice Alto	52
3.3.1.	Ejemplo Introductorio	52
3.3.2.	Algoritmo de Pantelides	53
3.3.3.	El índice de perturbación de una DAE	56
3.3.4.	Elección de las Variables de Estado	58
3.4.	DAEs y Diagramas de Bloques	59
3.4.1.	Diagramas de Bloques y Relaciones Causales	59
3.4.2.	Procedimiento para obtener un DB desde una DAE	59
3.4.3.	Diagramas de Bloques y Lazos Algebraicos	61
3.4.4.	Diagramas de Bloques y Sistemas de Índice Alto	61
3.5.	Herramientas de Software para Causalización de DAEs	62
3.6.	Bibliografía Complementaria	63
3.7.	Problemas Propuestos	64
4.	Técnicas Básicas de Modelado	66
4.1.	Circuitos Eléctricos	66
4.1.1.	Componentes Básicos	66
4.1.2.	Relaciones Estructurales	68
4.1.3.	Elementos de Conmutación	69
4.1.4.	Consideraciones sobre el orden	71
4.2.	Sistemas Mecánicos	71
4.2.1.	Modelos Mecánicos Traslacionales	71
4.2.2.	Modelos Mecánicos Rotacionales	75
4.2.3.	Modelos Roto-Traslacionales	77
4.2.4.	Mecánica Planar	79
4.2.5.	Consideraciones sobre el orden	81
4.3.	Sistemas Hidráulicos	81
4.3.1.	Componentes Básicos y Relaciones Constitutivas	82
4.3.2.	Relaciones Estructurales	83
4.3.3.	Otros Componentes	84
4.3.4.	Consideraciones sobre el Orden	85
4.4.	Sistemas Térmicos	85
4.4.1.	Componentes y Relaciones Constitutivas Básicas	85
4.4.2.	Relaciones Estructurales	87
4.4.3.	Consideraciones Sobre el Orden y Analogía con otros Dominios	89
4.5.	Sistemas Multidominio	89
4.5.1.	Sistemas Electro-Mecánicos	89

4.5.2.	Sistemas Hidráulico-Mecánicos	90
4.5.3.	Componentes Modulados	91
4.6.	Modelos no Físicos	92
4.6.1.	Modelos Epidemiológicos	92
4.6.2.	Modelos Ecológicos y Poblacionales	94
4.6.3.	Modelos Económicos	96
4.6.4.	Parametrización de los Modelos	98
4.7.	Construcción Directa de Diagramas de Bloques	98
4.8.	Bibliografía Complementaria	100
4.9.	Problemas Propuestos	100
5.	Modelado Orientado a Objetos	106
5.1.	Modelica y Modelos Orientados a Objetos	106
5.1.1.	Modelado por Componentes	108
5.1.2.	Conectores y Conexiones	110
5.1.3.	Herencia	112
5.1.4.	Modelado Jerárquico	113
5.2.	Construcción de Librerías de Distintos Dominios	117
5.2.1.	Circuitos Eléctricos	117
5.2.2.	Sistemas Mecánicos	122
5.2.3.	Sistemas Hidráulicos	131
5.2.4.	Sistemas Térmicos	134
5.2.5.	Sistemas Multidominio	136
5.3.	Sistemas de Control y Modelos Causales	141
5.3.1.	Diagramas de Bloques	141
5.3.2.	Sistemas Muestreados	144
5.3.3.	Sistemas con Eventos	145
5.3.4.	Sensores y Actuadores	146
5.4.	DAE Definida por un Modelo Orientado a Objetos	148
5.5.	La Librería Estándar de Modelica	151
5.6.	Bibliografía Complementaria	152
5.7.	Problemas Propuestos	152
6.	Modelado Mediante Formalismos Energéticos	164
6.1.	Introducción a los Bond Graphs	164
6.1.1.	Componentes Elementales	164
6.1.2.	Sensores y Componentes Modulados	170
6.1.3.	Variables Generalizadas y Dominios Físicos	171
6.1.4.	Energía Asociada a un Bond Graph	171
6.2.	Modelado con Bond Graphs	172
6.2.1.	Circuitos Eléctricos	172
6.2.2.	Sistemas Mecánicos	173
6.2.3.	Sistemas Hidráulicos	174
6.2.4.	Sistemas Multidominio	176
6.3.	Bond Graphs y DAEs	177
6.3.1.	Obtención de las Relaciones Constitutivas y Estructurales	177
6.3.2.	Causalización	178
6.3.3.	Singularidades Estructurales y Lazos Algebraicos	180
6.3.4.	Orden de la DAE	181
6.4.	Bond Graphs y Modelica	183
6.4.1.	Componentes Básicos	183
6.4.2.	Vínculos Estructurales, Transformadores y Giradores	185
6.4.3.	Componentes Modulados	186
6.4.4.	Sensores	186

6.4.5.	Interfaz con Otras Librerías	188
6.5.	Ecuaciones de Euler-Lagrange	188
6.5.1.	Ejemplos Introdutorios	188
6.5.2.	El Principio de Acción Estacionaria	190
6.5.3.	Coordenadas Generalizadas	192
6.5.4.	Fuerzas no Conservativas	192
6.5.5.	Ecuaciones de Euler-Lagrange y DAEs	194
6.5.6.	Extensión a Otros Dominios	194
6.6.	Bibliografía Complementaria	194
6.7.	Problemas Propuestos	195
7.	Herramientas de Análisis de Sistemas Lineales	198
7.1.	Sistemas de Orden 1	198
7.1.1.	Ecuaciones de Estado	198
7.1.2.	Función Transferencia	200
7.2.	Sistemas de Orden 2	203
7.2.1.	Ecuaciones de Estado	203
7.2.2.	Función Transferencia	208
7.3.	Sistemas de Orden 2 Generales y de Orden Arbitrario	214
7.3.1.	Retratos de Fase	214
7.3.2.	Solución General de las Ecuaciones de Estado	219
7.3.3.	Puntos de Equilibrio y Estabilidad Interna	221
8.	Herramientas de Análisis de Sistemas No Lineales	223
8.1.	Trayectorias, Puntos de Equilibrio y Estabilidad	223
8.1.1.	Trayectorias en Sistemas No Lineales	223
8.1.2.	Determinación de los Puntos de Equilibrio	228
8.1.3.	Modelo en Torno al Equilibrio	229
8.2.	Linealización	229
8.2.1.	Linealización en Torno al Equilibrio	229
8.2.2.	Linealización en un Sistema de DAEs	230
8.2.3.	Linealización en OpenModelica	231
8.3.	Estabilidad de Lyapunov	235
8.3.1.	Definición de Estabilidad	235
8.3.2.	Método Directo de Lyapunov	236
8.3.3.	Principio de Invariancia de LaSalle	238
8.4.	Análisis Mediante Aproximaciones Lineales	239
8.4.1.	Método Indirecto de Lyapunov	239
8.4.2.	Teorema de Hartman-Grobman	241
8.5.	Bibliografía Complementaria	243
8.6.	Problemas Propuestos	243

Introducción

El presente curso tiene por objetivo estudiar diversos temas relacionados con el modelado, la simulación y el análisis de sistemas dinámicos provenientes de distintos dominios, principalmente de la física. El curso pretende con esto brindar herramientas fundamentales para el abordaje de problemas vinculados al control automático en los cuales es menester contar con modelos matemáticos y con herramientas que permiten analizar el comportamiento de los sistemas en su interacción con los mecanismos de control que los comandan.

En estos problemas, el procedimiento típico comienza con la construcción de un modelo matemático del que luego se estudia su dinámica mediante simulación y/o análisis. El proceso de construcción del modelo se suele realizar mediante la formulación de las leyes matemáticas que rigen los fenómenos físicos más relevantes que ocurren en el sistema en cuestión. En tal caso se habla de *modelado por primeros principios*, y es esta la manera de modelar sobre la que trabajaremos en el curso. Una alternativa la constituye el modelado por experimentación, el cual no abordaremos aquí aunque usaremos eventualmente algunos de sus principios para completar la parametrización de los modelos.

Las mencionadas leyes matemáticas que rigen los fenómenos más relevantes tendrán forma de ecuaciones diferenciales (leyes de Newton, por ejemplo) o bien de ecuaciones algebraicas (la ley de Ohm, por ejemplo) que vincularán las distintas magnitudes del sistema. Por este motivo, trabajaremos con modelos que, más allá del formalismo de representación que utilicemos, se corresponderán a sistemas de *ecuaciones diferenciales algebraicas*.

Aunque en casos simples obtener modelos de sistemas de ecuaciones algebraicas a partir de las leyes básicas de la física es relativamente sencillo, en sistemas complejos con muchos componentes esto se torna significativamente más difícil. Por esto es que se han desarrollado distintas técnicas y herramientas de modelado que facilitan notablemente el proceso permitiendo pensar los modelos como la composición y conexión de modelos más simples. Entre estas técnicas, el *modelado orientado a objetos* y, particularmente, el lenguaje Modelica se han impuesto en los últimos años como las herramientas más utilizadas para formular modelos matemáticos. Por lo tanto, estas técnicas de modelado constituirán también una parte importante del curso.

Además del modelado orientado a objetos, veremos en el curso otras herramientas más clásicas que permiten obtener las ecuaciones que conforman un modelo a partir de principios energéticos como son los Bond Graphs y las ecuaciones de Euler Lagrange. Veremos también que estas herramientas en algunos casos podrán simplificar el proceso de modelado y nos facilitarán además la interpretación física de los fenómenos que ocurren en el sistema.

Si bien las ecuaciones diferenciales algebraicas son la forma matemática natural de describir los fenómenos que ocurren en un sistema continuo, las mismas no permiten directamente estudiar sus propiedades ni tampoco realizar simulaciones eficientemente. Tanto para propósitos de análisis como de simulación eficiente, necesitaremos utilizar modelos en *Ecuaciones de Estado* que, como veremos en el curso, pueden obtenerse desde las ecuaciones diferenciales algebraicas ordenando las mismas adecuadamente y en algunos casos aplicando algunos procedimientos adicionales.

En el curso también estudiaremos distintos *Métodos de Integración Numérica* que nos permitirán simular los modelos una vez que los mismos estén expresados en forma de ecuaciones de estado. Asimismo, aprovechando esta forma de representación, veremos métodos para analizar la estabilidad y la forma cualitativa de las trayectorias tanto para sistemas lineales como no lineales.

La Figura 1 sintetiza entonces el procedimiento típico que se sigue desde el reconocimiento de las leyes que rigen los fenómenos más relevantes de un sistema hasta su análisis y simulación, pasando por la

construcción del modelo matemático (aprovechando eventualmente herramientas de modelado orientado a objetos o de formalismos energéticos) y por su posterior ordenamiento en forma de ecuaciones de estados. El curso, en definitiva, pretende abarcar los principales aspectos de este proceso con el ordenamiento de los capítulos mostrado en dicha figura.

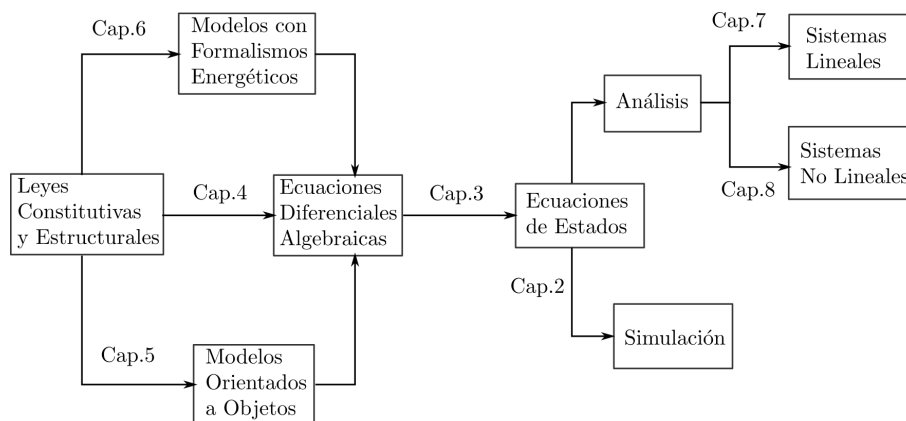


Figura 1: El proceso de modelado, simulación y análisis y su organización a lo largo del curso.

De esta manera, tras las definiciones y conceptos básicos que se brindan en el Capítulo 1, en el Capítulo 2 estudiaremos los principios de la simulación de sistemas continuos y presentaremos los principales métodos de integración numérica que se utilizan en la práctica y en las herramientas de software de simulación más difundidas. Luego, en el Capítulo 3 veremos como ordenar los modelos de ecuaciones diferenciales algebraicas para transformarlos en sistemas de ecuaciones de estado que puedan analizarse y simularse eficientemente.

En el Capítulo 4 introduciremos las técnicas básicas de modelado para distintos dominios (circuitos eléctricos, sistemas mecánicos, hidráulicos y térmicos), con sus leyes fundamentales y con distintas estrategias para organizar las ecuaciones que surgen de dichas leyes. Tras esto, en el Capítulo 5 estudiaremos cómo construir modelos orientados a objetos, utilizando en particular el ya mencionado lenguaje de modelado llamado Modelica. Completando luego las metodologías de modelado, en el Capítulo 6 estudiaremos dos formalismos para construir modelos a partir de fenómenos energéticos: los Bond Graphs y las ecuaciones de Euler-Lagrange.

Por último, en los Capítulos 7 y 8 estudiaremos distintas herramientas de análisis para sistemas lineales y no lineales respectivamente. En el primer caso veremos especialmente técnicas basadas en el análisis de la respuesta temporal y en el análisis de trayectorias en el espacio de estado. En el caso de los sistemas no lineales veremos técnicas de análisis que en algunos casos utilizarán linealización y en otros tratarán directamente con las no linealidades principalmente mediante las técnicas de Lyapunov.

Capítulo 1

Sistemas Dinámicos y Modelos Matemáticos

En este curso trabajaremos con modelos matemáticos de sistemas dinámicos. En este primer capítulo introductorio presentaremos los principales conceptos relacionados tanto a los sistemas dinámicos como a los modelos matemáticos.

El contenido de este capítulo está principalmente basado en el libro de Peter Fritzson *Introducción al Modelado y Simulación de Sistemas Técnicos y Físicos con Modelica* [Fri15] y en el apunte *Sistemas Dinámicos y Modelos Matemáticos* [Jun15] de la cátedra de Dinámica de los Sistemas Físicos de nuestra Facultad.

1.1. Sistemas Dinámicos

1.1.1. El Concepto de Sistema

La palabra *sistema* es utilizada de manera permanente en nuestro lenguaje cotidiano para referirnos a diversas cosas tanto en el lenguaje coloquial como en el técnico-científico. En el contexto de este curso, sin embargo, intentaremos utilizar una definición más precisa de este concepto.

La Teoría General de Sistemas brinda múltiples definiciones para el concepto. Una definición bastante habitual en la literatura [Sky05] es la siguiente: “Un sistema es un conjunto de entidades interactuantes que conforman una unidad”.

Otra definición muy similar proveniente de normas DIN Alemanas, establece en tanto que:

Definición 1.1. *Sistema*

Un Sistema es una disposición delimitada de entidades interactuantes.

En esta última definición podemos destacar las siguientes palabras:

Delimitación: Los sistemas tienen una *frontera* (espacial o conceptual) que los separa del resto del Universo. Esto no quiere decir que no pueda haber interacción entre el sistema y el resto, sino que los elementos que están fuera del sistema y que actúan sobre este se reemplazan por sus acciones. Estas acciones son vistas como *entradas* externas por el sistema.

Entidades Interactuantes: Son los *Componentes* del sistema, que pueden ser elementos, procesos o sistemas más simples (sub-sistemas).

Disposición (de las Entidades): Los sistemas tienen una *Estructura* de manera que el funcionamiento de los mismos no depende sólo de los componentes que tiene sino también de la manera en que dichos componentes están dispuestos y conectados.

De este análisis, podemos concluir de manera equivalente que un *Sistema* es una entidad formada por un conjunto de componentes y una estructura.

1.1.2. Clases de Sistemas

Según los procesos que involucran, los sistemas suelen clasificarse como *Físicos* (o *Concretos*) y *Abstractos* (o *Conceptuales*). Los Sistemas Físicos se caracterizan por su existencia en la realidad física espacio-temporal y en los mismos la interacción entre los componentes involucra el intercambio de materia, y/o energía, y/o información.

Los sistemas físicos, en tanto, se pueden clasificar como *dinámicos* o *estáticos*, siendo los primeros los que tienen capacidad de almacenar energía, materia o información.

Este curso está dedicado al estudio de los *Sistemas Físicos Dinámicos*, si bien en el transcurso del mismo trabajaremos eventualmente con sistemas más generales.

1.2. Modelos Matemáticos

En las disciplinas científicas y técnicas se trabaja con sistemas sobre los que habitualmente debemos aplicar métodos experimentales. Esto es, debemos realizar ensayos para observar la reacción del sistema a los mismos y establecer leyes (generalmente matemáticas) que expliquen el comportamiento de dichos sistemas.

Estos métodos experimentales, sin embargo, no son siempre viables. Hay distintos factores por los cuáles muchas veces no es conveniente (o es imposible) experimentar directamente sobre un sistema. Entre estos factores, encontramos:

Costos: Los experimentos sobre un sistema pueden tener costos que impidan su realización. Por ejemplo, realizar un experimento en una planta industrial podría significar detener un proceso de producción durante varias horas, con las consiguientes pérdidas económicas.

Riesgos: Las consecuencias del experimento pueden ser inadmisibles como por ejemplo un experimento sobre una planta nuclear para estudiar el escape de vapor radioactivo a la atmósfera.

Experimentos Irrealizables: En ocasiones un experimento no puede ser realizado porque las acciones involucradas son físicamente imposibles (por ejemplo, si nos quisiéramos preguntar cuál sería el efecto de un desplazamiento del eje magnético de la Tierra) o porque el sistema aún no existe (en la etapa de diseño de dicho sistema, típicamente). Este último caso es muy habitual en aplicaciones de Ingeniería.

Frente a la inconveniencia o a la imposibilidad de realizar experimentos directamente sobre los sistemas, se recurre a la experimentación sobre *modelos* de dichos sistemas.

1.2.1. El Concepto de Modelo

Definición 1.2. Modelo

Un modelo de un sistema es una herramienta que permite responder interrogantes sobre este último sin tener que recurrir a la experimentación sobre el mismo.

En la práctica, un modelo es una representación simplificada del sistema que representa donde la simplificación se realiza en función de la o las preguntas que se quieren responder. De esta manera, un modelo siempre involucra un sistema y uno o más interrogantes.

1.2.2. Clases de Modelos

Más allá de las disciplinas científicas y técnicas, los seres humanos utilizamos cotidianamente modelos (es decir, simplificaciones) de la realidad para poder comprender lo que nos rodea y ocurre. Estos modelos pueden ser de distinto tipo:

Modelos mentales: Hay incontables situaciones cotidianas en las cuales utilizamos modelos mentales para predecir lo que ocurriría ante potenciales acciones que se realicen. Por ejemplo, si sostenemos un objeto y lo soltamos desde cierta altura, no es necesario conocer las leyes de la Física para saber que va a caer al piso.

Modelos verbales/textuales: También utilizamos cotidianamente modelos muy simplificados que expresan de manera verbal o textual el comportamiento de ciertos sistemas ante diversas situaciones. Por ejemplo, la frase “Viento del este, agua como peste”.

Modelos físicos: Son objetos físicos que reproducen algunas propiedades de sistemas reales para ayudarnos a responder preguntas sobre los mismos. Por ejemplo, un túnel de viento para el estudio de fenómenos aerodinámicos, una reproducción a escala del lecho de un río para estudios hidrológicos, etc.

Modelos matemáticos: Son modelos que describen mediante expresiones matemáticas (ecuaciones, desigualdades, expresiones lógicas) las relaciones entre las distintas magnitudes que caracterizan un sistema.

1.2.3. Clasificación de Modelos Matemáticos

Hay diversas formas de clasificar los modelos matemáticos, entre ellas encontramos:

Tiempo Continuo vs. Tiempo Discreto: La manera en la que se representa la evolución del tiempo en un modelo establece una pauta para su clasificación. Decimos que un modelo es de tiempo continuo cuando las relaciones entre las variables están definidas para todo instante de tiempo (dentro del intervalo de validez del modelo). Por el contrario, diremos que un modelo es de tiempo discreto cuando las relaciones entre las variables estén definidas sólo en determinados instantes discretos de tiempo.

Estáticos vs. Dinámicos: Un sistema se dice *dinámico* cuando hay relaciones entre variables que no dependen únicamente de sus valores instantáneos sino también de valores pasados de las mismas. Es decir, un sistema es dinámico cuando hay *memoria*. Por el contrario, en un sistema *estático* todas las relaciones entre variables dependen exclusivamente de los valores instantáneos de dichas variables. A modo de ejemplo, el modelo de un circuito eléctrico que contenga algún capacitor y/o inductor resultará dinámico, mientras que el modelo de un circuito que contenga únicamente elementos resistivos será estático.

Determinísticos vs. Estocásticos: Un modelo es determinístico si expresa matemáticamente sin incertidumbre las relaciones entre las variables. El modelo determinístico asigna de esta forma unívocamente valores y/o funciones ciertas y determinadas a la información que procesa. Por el contrario, un modelo es estocástico si expresa las relaciones con incertidumbre entre las variables mediante conceptos probabilísticos usando variables aleatorias. Dichas relaciones son descriptas usando variables o procesos estocásticos.

Parámetros Distribuidos vs. Parámetros Concentrados: Las magnitudes que caracterizan a los fenómenos físicos toman valores en el tiempo y en el espacio. Si un modelo matemático conserva la dependencia espacio-temporal en la representación de las magnitudes, se dice que el mismo es de *parámetros distribuidos*, ya que en general los coeficientes o parámetros del sistema están distribuidos en el espacio (por ejemplo, la densidad del fluido compresible en un gasoducto o la resistividad, inductancia y capacidad por unidad de longitud en una línea de transmisión). Los modelos a parámetros distribuidos son, típicamente, ecuaciones en derivadas parciales ya que involucran variación en el tiempo y espacio que resultan en derivadas respecto a la variable temporal y respecto a las coordenadas espaciales.

Cuando las variables de un modelo no dependen de las coordenadas espaciales o bien cuando se reemplaza la dependencia espacial de las variables por su promedio en la región del espacio donde están definidas se dice que el modelo es de *parámetros concentrados*. Estos modelos dinámicos son, típicamente, ecuaciones diferenciales ordinarias o más generalmente ecuaciones algebraico diferenciales.

Paramétricos vs. No Paramétricos: Un modelo *paramétrico* está caracterizado por un número finito y determinado de parámetros (por ejemplo, una función transferencia o un sistema de ecuaciones

diferenciales). Los modelos no paramétricos, en cambio, la caracterización se basa típicamente en series de datos cuya longitud no es fija (por ejemplo, la curva de la respuesta a un escalón de un sistema dinámico, la curva de respuesta en frecuencia de un amplificador).

Lineales vs. No Lineales: Se dice que un modelo es *lineal* cuando vale el principio de superposición, es decir que la acción de causas superpuestas provoca un efecto equivalente a la superposición de los correspondientes efectos.

Estacionarios vs. Inestacionarios: Un modelo se dice *estacionario* si estando en idénticas condiciones en dos instantes diferentes, responde de igual manera ante idénticas acciones.

Causales vs. Acausales: Un modelo se dice *causal* cuando además de describir relaciones matemáticas entre las magnitudes establece relaciones de causa-efecto entre las mismas. Los Diagramas de Bloques son la herramienta típica de modelado causal.

1.2.4. Elementos de los Modelos Matemáticos

Como ya mencionamos, los modelos matemáticos formulan mediante expresiones matemáticas (ecuaciones principalmente) las relaciones entre las magnitudes que caracterizan al mismo. Dependiendo de la naturaleza del modelo en cuestión, estas expresiones matemáticas pueden estar dadas de diversas maneras. Sin embargo, independientemente de la representación que utilicemos, las expresiones matemáticas involucradas siempre relacionarán magnitudes que se clasifican de la siguiente manera:

Variables Fundamentales: Son las variables que representan el tiempo y las coordenadas espaciales.

Variables Descriptivas: Son las variables que representan las magnitudes físicas asociadas al sistema. Entre las mismas encontramos:

Parámetros: Son magnitudes constantes generalmente o pueden ser variables pero con una ley predeterminada independiente de los procesos que puedan ocurrir en el sistema.

Entradas (Variables Independientes): Son variables descriptivas cuyas señales son independientes de otras señales en el sistema, y no están prefijadas. Representan acciones externas del resto del universo sobre el sistema. En aplicaciones de control se suelen dividir en dos categorías:

Entradas manipuladas: Entradas cuya trayectoria se puede manipular.

Perturbaciones: Entradas que son el resultado de fenómenos que están fuera del modelo y por lo tanto no se pueden ni manipular ni conocer a priori la evolución de sus trayectorias.

Variables Dependientes: Son variables descriptivas cuyas señales dependen de la evolución del sistema. Dentro de estas encontramos a las *salidas*, que son simplemente variables dependientes de interés.

Un concepto fundamental asociado a las variables que forman parte del modelo matemático de un sistema dinámico es el de *estado*, que puede definirse como sigue:

Definición 1.3. *Vector de Estados*

Un vector de estados es un conjunto de variables dependientes cuyo conocimiento en un instante de tiempo permite calcular el valor de cualquier otra variable dependiente en dicho instante, asumiendo conocidos los valores de las entradas en ese instante de tiempo.

Por ejemplo, en un circuito RLC (serie o paralelo) conociendo la tensión del capacitor y la corriente del inductor (además del valor de alguna entrada, si la hubiera) podemos calcular cualquier otra corriente o tensión. De esta forma, el vector formado por ambas variables sería un vector de estados. Alternativamente, podríamos elegir la carga en el capacitor y el flujo en el inductor y también podríamos calcular cualquier otra corriente o tensión del circuito, lo que muestra que hay múltiples maneras de elegir el vector de estados.

Más aún, de acuerdo a la definición anterior, podríamos elegir como vector de estados las cuatro variables (tensión y carga del capacitor, corriente y flujo del inductor), lo que conduce a un vector con

estados redundantes (ya que conociendo la tensión del capacitor podemos saber la carga y viceversa, y lo mismo ocurre con el inductor). Efectivamente, el conjunto de cuatro variables es un vector de estados pero el mismo no es un *vector de estados minimal*:

Definición 1.4. *Vector de Estados Minimal*

Un vector de estados minimal es un vector de estados que contiene la mínima cantidad posible de variables dependientes.

El número de variables que constituyen un vector de estados minimal determina así el *orden* del modelo matemático, definido como sigue:

Definición 1.5. *Orden de un Modelo Matemático* Es el número mínimo de variables dependientes con cuyo valor quedan estáticamente determinadas todas las otras variables dependientes del modelo.

Comparando las últimas dos definiciones, queda claro que el orden de un modelo es la cardinalidad de cualquier vector de estados minimal.

1.3. Representación de Modelos Matemáticos

Hay diversas representaciones para los modelos matemáticos, las que resultan más o menos adecuadas según la naturaleza del sistema a representar y según el problema que se quiera resolver.

En este curso trabajaremos en gran medida con modelos de parámetros concentrados de sistemas físicos dinámicos, los que constituirán desde el punto de vista matemático sistemas de ecuaciones diferenciales ordinarias (ODEs, por sus siglas en inglés) o, más generalmente, sistemas de ecuaciones algebraico diferenciales (DAEs). Para las ODEs utilizaremos principalmente su representación en forma de *Ecuaciones de Estado*.

1.3.1. Ecuaciones de Estado

Dado el circuito de la Figura 1.1, podemos plantear las siguientes ecuaciones:

$$u_R(t) - R i_R(t) = 0 \quad (1.1a)$$

$$\dot{q}(t) - i_C(t) = 0 \quad (1.1b)$$

$$q(t) - C u_C(t) = 0 \quad (1.1c)$$

$$\dot{\phi}(t) - u_L(t) = 0 \quad (1.1d)$$

$$\phi(t) - L i_L(t) = 0 \quad (1.1e)$$

$$u_S(t) - v(t) = 0 \quad (1.1f)$$

$$u_L(t) + u_R(t) + u_C(t) - u_S(t) = 0 \quad (1.1g)$$

$$i_L(t) - i_R(t) = 0 \quad (1.1h)$$

$$i_C(t) - i_L(t) = 0 \quad (1.1i)$$

$$i_S(t) - i_R(t) = 0 \quad (1.1j)$$

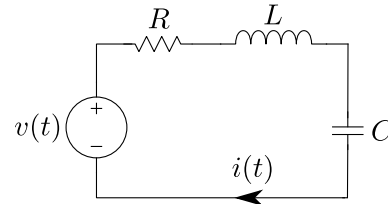


Figura 1.1: Circuito RLC Serie

Si trabajamos algebraicamente sobre el sistema de ecuaciones (1.1) podemos llegar fácilmente a la siguiente representación compacta

$$\begin{aligned} \dot{q}(t) &= \frac{1}{L} \phi(t) \\ \dot{\phi}(t) &= -\frac{1}{C} q(t) - \frac{R}{L} \phi(t) + u_S(t), \end{aligned} \quad (1.2)$$

que tiene la forma de un sistema de *Ecuaciones de Estado*, en el cual las *variables de estado* son la carga $q(t)$ y el flujo $\phi(t)$ y la variable de entrada es la tensión de la fuente $u_S(t)$.

Puede verse fácilmente que, en concordancia con el concepto de *vector de estados* dado por la Definición 1.3, conociendo $q(t)$ y $\phi(t)$ además de la entrada $u_S(t)$, podemos calcular cualquier otra variable del modelo. Más aún, repasando la Definición 1.4, el vector $[q(t), \phi(t)]$ constituye un *Vector de Estados Minimal* por lo que el modelo matemático es de segundo orden.

Las Ecuaciones de Estado suelen complementarse con ecuaciones de salida estáticas que calculan ciertas variables dependientes de interés. Por ejemplo, si necesitáramos conocer las tensiones sobre la resistencia y sobre la inductancia ($u_R(t)$ y $u_L(t)$), podemos complementar el sistema de la Ec.(1.2) con las siguientes ecuaciones de salida

$$\begin{aligned} u_R(t) &= \frac{R}{L} \phi(t) \\ u_L(t) &= -\frac{1}{C} q(t) - \frac{R}{L} \phi(t) + u_S(t). \end{aligned} \quad (1.3)$$

En casos generales, la representación en ecuaciones de estado tendrá la forma

$$\begin{aligned} \dot{x}_1(t) &= f_1(x_1(t), \dots, x_n(t), u_1(t), \dots, u_m(t), t) \\ \dot{x}_2(t) &= f_2(x_1(t), \dots, x_n(t), u_1(t), \dots, u_m(t), t) \\ &\vdots \\ \dot{x}_n(t) &= f_n(x_1(t), \dots, x_n(t), u_1(t), \dots, u_m(t), t) \end{aligned} \quad (1.4)$$

donde $x_i(t)$ serán las *variables de estado* y las variables $u_i(t)$ serán las *variables de entrada*. Observar que esta notación permite describir relaciones *no lineales* a través de las funciones $f_i(\cdot)$ y representar también sistemas *inestacionarios* a través de la dependencia directa de las f_i con el tiempo t .

La Ecuación (1.4) suele utilizarse de manera vectorial, definiendo los vectores de estado $\mathbf{x} \triangleq [x_1, \dots, x_n]^T$ y de entradas $\mathbf{u} \triangleq [u_1, \dots, u_m]^T$ y reescribiendo entonces

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t). \quad (1.5)$$

Ocasionalmente complementaremos las Ecuaciones de Estados con Ecuaciones de Salida como la de la Eq.(1.3), que en el caso general tomarán la siguiente forma:

$$\begin{aligned} y_1(t) &= g_1(x_1(t), \dots, x_n(t), u_1(t), \dots, u_m(t), t) \\ y_2(t) &= g_2(x_1(t), \dots, x_n(t), u_1(t), \dots, u_m(t), t) \\ &\vdots \\ y_n(t) &= g_n(x_1(t), \dots, x_n(t), u_1(t), \dots, u_m(t), t) \end{aligned} \quad (1.6)$$

Las Ecuaciones de salida también admiten la representación vectorial

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t), \quad (1.7)$$

donde $\mathbf{y} \triangleq [y_1, \dots, y_p]^T$ es el vector de salidas.

Un caso particular de las Ecuaciones de Estado ocurre cuando la función \mathbf{f} es *lineal* y por lo tanto las Ecs.(1.5) y (1.7) pueden reescribirse como

$$\begin{aligned} \dot{\mathbf{x}}(t) &= A(t) \mathbf{x}(t) + B(t) \mathbf{u}(t) \\ \mathbf{y}(t) &= C(t) \mathbf{x}(t) + D(t) \mathbf{u}(t) \end{aligned} \quad (1.8)$$

para ciertas matrices $A(t)$, $B(t)$, $C(t)$ y $D(t)$. En tal caso, diremos que el modelo matemático es *lineal*. Si además estas matrices fueran constantes (es decir, no dependieran de t), el modelo resultaría *lineal y estacionario*.

1.3.2. Ecuaciones Diferenciales Algebraicas

Las ecuaciones diferenciales algebraicas, como lo indica su nombre, son una combinación de ecuaciones diferenciales y ecuaciones algebraicas y pueden verse como una generalización de las ODEs. Este tipo de modelo matemático surge naturalmente al listar las relaciones matemáticas que aparecen entre las distintas variables de un sistema y por lo tanto suele ser la representación más simple de obtener en el proceso de modelado.

Por ejemplo, el sistema dado por las Ecs.(1.1a)-(1.1j) constituye un sistema de ecuaciones diferenciales algebraicas. En el mismo, podemos distinguir dos ecuaciones diferenciales, la Ec.(1.1b) y la Ec.(1.1d), y ocho ecuaciones puramente algebraicas.

Podemos ver también que el modelo contiene relaciones que son intrínsecas a los elementos del circuito y otras que se deben a la manera en la que dichos elementos están conectados. Específicamente, las Ecuaciones (1.1a)-(1.1f) se corresponden a las leyes asociadas a una resistencia, a un capacitor, a un inductor y a una fuente de tensión. Por el contrario, las Ecuaciones (1.1g)-(1.1j) se corresponden a la manera en que dichos elementos se encuentran conectados.

Las relaciones de un modelo matemático que son intrínsecas a los elementos se denominan *Relaciones Constitutivas*, mientras que las que expresan la manera en que los elementos se conectan se denominan *Relaciones Estructurales*. Como veremos más adelante, distinguir claramente estas relaciones va a ser fundamental a la hora de construir un modelo matemático.

De hecho, si a partir de un esquema gráfico como el de la Figura 1.1 quedaran unívocamente definidas cuáles son las relaciones constitutivas y estructurales del modelo matemático, podríamos utilizar directamente el esquema de dicha figura como representación equivalente al sistema de la DAE (1.1). En realidad, esto es lo que haremos más adelante cuando veamos los principios del *Modelado Orientado a Objetos*.

En un caso general las DAEs tendrán la forma:

$$\begin{aligned} f_1(x_1(t), \dots, x_n(t), \dot{x}_1(t), \dots, \dot{x}_n(t), a_1(t), \dots, a_r(t), u_1(t), \dots, u_m(t), t) &= 0 \\ f_2(x_1(t), \dots, x_n(t), \dot{x}_1(t), \dots, \dot{x}_n(t), a_1(t), \dots, a_r(t), u_1(t), \dots, u_m(t), t) &= 0 \\ &\vdots \\ f_{n+r}(x_1(t), \dots, x_n(t), \dot{x}_1(t), \dots, \dot{x}_n(t), a_1(t), \dots, a_r(t), u_1(t), \dots, u_m(t), t) &= 0 \end{aligned} \quad (1.9)$$

donde las variables $x_i(t)$ serán en principio los *estados* (más adelante veremos que algunas de estas componentes pueden no constituir verdaderamente estados). Las variables $a_i(t)$ serán las *variables algebraicas* y las variables $u_i(t)$ serán las *variables de entrada*. Observar que, al igual que en el caso de las Ecuaciones de Estado, esta notación permite describir relaciones *no lineales* a través de las funciones $f_i(\cdot)$ y representar también sistemas *inestacionarios* a través de la dependencia directa de las f_i con el tiempo t .

Al igual que las ODEs, la Ecuación (1.9) suele presentarse de manera vectorial, definiendo los vectores de estado $\mathbf{x} \triangleq [x_1, \dots, x_n]^T$, de entradas $\mathbf{u} \triangleq [u_1, \dots, u_m]^T$ y de variables algebraicas $\mathbf{a} \triangleq [a_1, \dots, a_r]^T$ y reescribiendo entonces

$$\mathbf{f}(\mathbf{x}(t), \dot{\mathbf{x}}(t), \mathbf{a}(t), \mathbf{u}(t), t) = 0. \quad (1.10)$$

Como ya mencionamos, las DAEs son generalmente la representación más simple de obtener cuando buscamos construir un modelo matemático. De hecho, basta con listar las relaciones constitutivas y estructurales de un modelo para tener una DAE sin ser necesario ningún tipo de trabajo algebraico. Sin embargo, muchas veces no resulta simple simular o realizar análisis de propiedades directamente sobre las DAEs. Para estos propósitos será conveniente convertirlas (cuando sea posible) en sistemas de ecuaciones diferenciales ordinarias que generalmente representaremos en su forma de *Ecuaciones de Estado*.

Este proceso de conversión de DAEs a ODEs se suele denominar *causalización* y tiene la ventaja de ser completamente algoritmizable y, por lo tanto, lo puede resolver una computadora. Más adelante en el curso veremos como es este proceso de causalización y qué herramientas informáticas pueden utilizarse para el mismo.

1.4. Utilización de Modelos Matemáticos

Cuando definimos el concepto de Modelo, dijimos que los mismos servían para responder ciertas preguntas sobre un sistema evitando realizar experimentos sobre dicho sistema. En el caso más específico de los modelos matemáticos, estas preguntas generalmente implican predecir el comportamiento del sistema ante distintas situaciones, ya sea a través de resultados cuantitativos o bien estableciendo ciertas propiedades cualitativas.

Pueden entonces establecerse a grandes rasgos dos maneras de utilizar un modelo matemático:

Análisis Teórico: consistente en el estudio analítico de los modelos, determinando propiedades tanto de carácter cualitativo (estabilidad, controlabilidad, observabilidad, etc.) como cuantitativo (a través de la resolución analítica de las ecuaciones de estado, por ejemplo).

Análisis Experimental: consistente en el estudio mediante *simulaciones* de los modelos para también establecer propiedades cualitativas y cuantitativas.

1.4.1. Análisis Teórico de los Modelos Matemáticos

Supongamos que queremos determinar si el transitorio que se produce al alimentar el circuito de la Figura 1.1 con un escalón de tensión en la fuente tiene o no oscilaciones en función de los parámetros R , L y C . Una manera relativamente simple de hacerlo sería obteniendo un sistema de Ecuaciones de Estados como la Ec.(1.8) (donde las matrices resultarán constantes) y calculando analíticamente los autovalores de la matriz A en función de los valores de los parámetros (ver Capítulo 7). Un procedimiento equivalente sería obtener la Función Transferencia entre la entrada y una salida y calcular los polos.

En los sistemas lineales y estacionarios (como lo es el caso del circuito de la Figura 1.1) suele ser relativamente sencillo analizar propiedades mediante análisis teórico, si bien para esto en general será necesario transformar primero el modelo dado por una DAE como la de la Ec. (1.1) en una ODE como la Ec.(1.2). Más adelante veremos como se puede realizar esta transformación de manera sistemática y luego, en el Capítulo 7, repasaremos algunas de las principales técnicas para realizar análisis sobre los modelos lineales y estacionarios.

En los sistemas no lineales, en cambio, es bastante más complejo establecer propiedades de carácter teórico. De hecho, las herramientas que vimos en cursos anteriores para analizar modelos matemáticos están restringidas a los modelos lineales y estacionarios. Más adelante, en el Capítulo 8, veremos sin embargo que en ocasiones podremos aproximar un modelo no lineal como el de la Ec.(1.5) por uno lineal como el de la Ec.(1.8) y obtener ciertas propiedades del modelo no lineal (como la estabilidad y la forma de las trayectorias) a partir de las propiedades del modelo *linealizado*.

Sin embargo, en sistemas más complejos o ante preguntas de carácter más cuantitativo las técnicas analíticas dejan de ser convenientes y la simulación digital resulta la herramienta más adecuada.

1.4.2. Simulación

El camino más simple y directo para estudiar la evolución de las variables de un modelo matemático ante condiciones iniciales y entradas determinadas es la *simulación digital*, que consiste en resolver las ecuaciones del modelo utilizando algún tipo de aproximación numérica (en el caso de los sistemas de tiempo continuo). Al contrario que en los métodos teóricos, veremos que podemos simular directamente sobre las DAEs ya que todo el proceso posterior a la formulación de las ecuaciones del modelo es totalmente algoritmizable.

La desventaja de la simulación frente al estudio analítico es que los resultados que se obtienen son particulares para un determinado juego de parámetros, para una determinada trayectoria de entradas y para un determinado conjunto de condiciones iniciales. Sin embargo, dado que muchas veces las simulaciones son muy rápidas, es posible realizarlas *barriendo* juegos de parámetros, entradas y condiciones iniciales, pudiendo en muchos casos establecerse resultados cualitativos a partir de múltiples experimentos.

En el Capítulo 2 veremos las principales técnicas para simulación de sistemas continuos.

1.4.3. Limitaciones al usar Modelos Matemáticos

Cuando formulamos un modelo matemático a partir de un sistema, introducimos ciertas simplificaciones acordes al problema que queremos resolver y a las preguntas que queremos responder con dicho modelo. Sin embargo, una vez obtenido el modelo matemático, las hipótesis que se utilizaron para realizar las simplificaciones pueden no quedar explicitadas.

Por ejemplo, en el esquema del circuito de la Figura 1.1 no aparecen las dimensiones físicas del mismo. Por lo tanto, no podemos determinar cuál será la máxima frecuencia a la cual las leyes de la teoría de circuitos constituyen una aproximación válida. Si alguien quisiera simular la respuesta de este circuito (o analizar las propiedades teóricas) a una entrada senoidal de muy alta frecuencia, el modelo matemático en principio lo permite. Sin embargo, dependiendo de las dimensiones físicas del sistema real, los resultados obtenidos podrán tener o no sentido para dicho sistema.

Es un error muy común analizar o simular el modelo de un sistema sin tener en cuenta que las trayectorias obtenidas estén dentro del rango en el cual la aproximación que brinda dicho modelo tiene sentido. Por eso es fundamental en todo momento relacionar los resultados obtenidos al usar un modelo con su correspondencia en el sistema real.

1.5. Construcción de Modelos Matemáticos

El proceso que se realiza para obtener un modelo matemático que permita responder ciertas preguntas sobre un sistema se denomina *Modelado*. Este proceso suele realizarse de diversas maneras, habiendo dos grandes grupos de técnicas conceptualmente diferentes pero de uso complementario en la práctica, como luego veremos:

Modelado por Primeros Principios : También denominado modelado analítico, físico, o de *caja blanca* se basa en la obtención de las ecuaciones del modelo a partir de leyes conocidas que rigen el comportamiento de las componentes del sistema y de la forma en que se relacionan las mismas.

Modelado por Identificación : Llamado también modelado por experimentación o modelado de *caja negra*, se basa en proponer un modelo matemático con una estructura determinada y luego ajustar sus parámetros en base a datos obtenidos mediante diversos experimentos.

Más allá de la diferencia conceptual entre ambos procedimientos, como ya dijimos, ambos son complementarios. En el modelado por primeros principios se suele recurrir a la experimentación para determinar los valores de ciertos parámetros. Asimismo, en el modelado por identificación la estructura del modelo propuesto se suele basar en cierto conocimiento de los primeros principios que rigen la dinámica del sistema en cuestión.

1.5.1. Modelado por Primeros Principios

El proceso de construcción de un modelo matemático por primeros principios puede dividirse generalmente en las siguientes etapas:

Formulación del problema : El primer paso para construir un modelo matemático es definir claramente el problema que se quiere resolver con dicho modelo. Recordemos que un modelo se define siempre a partir de un sistema y de un problema a resolver.

Determinación de los fenómenos relevantes : En función del problema a resolver se realizan diversas *hipótesis simplificadoras*, se definen las magnitudes que cuantifican los fenómenos relevantes y se delimita de esta manera el sistema físico.

Si el sistema en cuestión es un circuito eléctrico, por ejemplo, y el problema es estudiar el tiempo de respuesta de los transitorios, muy posiblemente podremos utilizar la teoría de circuitos clásica. Si en cambio el problema es el diseño de un disipador para dicho circuito, deberemos incluir fenómenos térmicos además. Si el problema en cambio fuera estudiar las emisiones electromagnéticas del circuito, tendremos que incluir en el modelo las ecuaciones de Maxwell y la teoría de líneas de transmisión.

Descomposición en estructura y componentes : Para simplificar la obtención del modelo matemático, habitualmente buscaremos reconocer distintos subsistemas simples (componentes) determinando a su vez la manera en que los mismos interactúan (estructura). El resultado de esta etapa es generalmente un esquema funcional del estilo de la Figura 1.1.

Obtención de las Relaciones Matemáticas : Una vez reconocidas las componentes y la estructura, se deberán formular las *Relaciones Constitutivas* y las *Relaciones Estructurales*. Para este paso bastará con hacer un esquema con las componentes y sus conexiones si se utiliza un lenguaje de modelado orientado a objetos como Modelica. En cualquier caso, el resultado de esta etapa será un sistema de ecuaciones diferenciales algebraicas como el de la Ec.(1.1) y a partir de este momento podemos decir que contamos con un Modelo Matemático. Sin embargo, es posible que el mismo no esté en la forma que lo necesitamos para resolver el problema en cuestión o incluso que aún no conozcamos los valores de algunos parámetros y sea necesario continuar con el proceso de modelado.

Manipulación del Modelo : Hay diversos procedimientos para convertir un sistema de DAEs en otro tipo de representación (ecuaciones de estado, función transferencia, diagramas de bloques, etc.). Estos procedimientos son completamente algoritmizables y pueden realizarse aprovechando diversas herramientas de software.

Validación y Ajuste de Parámetros : Una vez construido el modelo matemático, se suele corroborar mediante simulaciones que el mismo efectivamente replique el comportamiento del sistema original bajo ciertas condiciones en las que es posible observar lo que hace dicho sistema. En esta etapa, también se suelen obtener y ajustar los valores de ciertos parámetros del modelo para lograr que el comportamiento del mismo sea lo más similar posible al del sistema original. Hay ocasiones en las cuales esta etapa falla y es necesario revisar las hipótesis simplificadoras retomando el procedimiento de modelado desde dicha etapa.

1.5.2. Modelado por Identificación

El modelado por identificación se suele utilizar cuando no se conocen bien la estructura y los componentes que conforman el sistema real o bien cuando estos son demasiado complejos y por lo tanto el modelo matemático resultante sería excesivamente complicado para el problema que se quiere resolver.

Al igual que en el caso del modelado por primeros principios, el proceso de modelado por identificación puede plantearse a través de una secuencia de etapas que en este caso puede resumirse como sigue:

Obtención de Datos : En función del problema a resolver y de las características del sistema, se suelen realizar diversos experimentos que muestren el comportamiento del sistema ante distintas condiciones y/o secuencia de entradas. En ciertas aplicaciones no es posible experimentar, pero se cuenta con series de datos históricos permiten realizar este paso. Estos datos suelen dividirse en dos subconjuntos: los *datos de ajuste* y los *datos de validación*.

Elección de la Estructura del Modelo : De acuerdo al conocimiento previo sobre el sistema, a lo observado en los datos, y siempre en concordancia con el problema a resolver, se elige un conjunto de modelos que generalmente comparten cierta estructura pero no los valores de los parámetros y que podrían representar al sistema. Por ejemplo, este conjunto podría consistir en todas las funciones transferencias de segundo orden con respuesta subamortiguada.

Selección del Modelo : Del conjunto de modelos elegido en el punto anterior, se selecciona el que mejor replica los datos de ajuste. Este punto suele involucrar un proceso de *optimización* para la obtención de los parámetros del modelo.

Verificación del Modelo : Finalmente, se suele verificar que el modelo seleccionado en el paso anterior tenga un ajuste aceptable respecto de los datos de validación. Si esto falla, es probable que haya que modificar el conjunto de modelos y/o ampliar el conjunto de datos de ajuste realizando eventualmente más experimentos.

1.5.3. Modelado de Caja Gris

En la práctica generalmente el modelado no se realiza siguiendo estrictamente los primeros principios ni siguiendo estrictamente las técnicas de identificación, sino mediante una combinación de ambos procedimientos. Es muy común comenzar el modelado aplicando hasta donde sea posible los primeros principios pero dejando como incógnita los valores de ciertos parámetros. Luego, se suelen diseñar experimentos para obtener datos (o se pueden utilizar series de datos existentes) y a partir de estos datos, se aplican técnicas de identificación que permiten ajustar los valores de los parámetros desconocidos.

Esta forma de proceder que combina el modelado por primeros principios con la identificación se denomina *Modelado de Caja Gris*.

1.6. Herramientas de Software para Modelado y Simulación

El modelado de sistemas de cierta complejidad no puede concebirse hoy en día sin el apoyo de herramientas informáticas específicas, que simplifican enormemente el trabajo de construcción de los modelos matemáticos y su posterior manipulación para obtener resultados de simulación y analizar diversas propiedades de los mismos.

Hay diversas herramientas para modelar sistemas específicos de cada disciplina (como el PSpice y sus derivados para los circuitos electrónicos, por ejemplo) y también herramientas para sistemas generales como Simulink.

Dada la diversidad de herramientas y la falta de compatibilidad entre las mismas, a fines del siglo pasado la comunidad de Modelado y Simulación decidió desarrollar un lenguaje abierto para describir modelos que sea independiente de la disciplina y que tenga una definición abierta que permita su utilización en cualquier herramienta de software que desee implementarlo. Este lenguaje, llamado *Modelica*, es actualmente el estándar más aceptado para representar modelos matemáticos de sistemas dinámicos.

Por ejemplo, el sistema de DAEs presentado en la Ec.(1.1) para el caso con entrada $v(t) = \sin(t)$ puede representarse en Modelica utilizando el Código 1.1. La traducción es totalmente directa y el modelo puede simularse utilizando cualquier herramienta de software que utilice el lenguaje Modelica.

Código 1.1: Modelo de la Ec.(1.1) en Modelica

```

model Circuit
  Real uR, iR, q, iC, uC, phi, uL, iL, uS, iS;
  parameter Real R=1, C=1, L=1;
equation
  uR - R * iR = 0;
  der(q) - iC = 0;
  q - C * uC = 0;
  der(phi) - uL = 0;
  phi - L * iL = 0;
  uS - sin(time) = 0;
  uL + uR + uC - uS = 0;
  iL - iR = 0;
  iC - iL = 0;
  iS - iR = 0;
end Circuit;

```

En particular, en este curso utilizaremos *OpenModelica*, una herramienta de código abierto que tiene un gran grado de desarrollo. Hay además herramientas comerciales como Dymola (la más eficiente y desarrollada) y Wolfram SystemModeler, entre otras.

Más adelante veremos que utilizando el lenguaje Modelica no sólo podremos describir los modelos en forma de DAEs, sino también directamente a partir de sus componentes y estructuras brindando una representación textual o directamente gráfica.

1.7. Bibliografía Complementaria

En este capítulo presentamos sintéticamente varios temas, cuyas fuentes bibliográficas principales son el libro de Peter Fritzson [Fri15] y un apunte anterior de la cátedra [Jun15]. Quienes quieran profundizar sobre temas relacionados a la Teoría de Sistemas pueden recurrir a libros dedicados al tema como [Sky05] mientras que sobre Teoría de Modelado y Simulación se puede recomendar el libro *Theory of Modeling and Simulation* [ZMK18].

En este curso luego profundizaremos sobre otros temas (simulación de sistemas continuos, técnicas de modelado y técnicas de análisis). Respecto al modelado, trabajaremos casi exclusivamente utilizando primeros principios y en los capítulos posteriores profundizaremos al respecto. Quien quiera profundizar sobre modelado por identificación, la referencia bibliográfica más clásica es sin dudas el libro de Ljung *System identification* [Lju98].

1.8. Problemas Propuestos

[P1.1] Clasificación de Modelos Matemáticos

Dados los siguientes modelos matemáticos, clasificarlos según sean (a) Estáticos o Dinámicos. (b) Parámetros Distribuidos o Concentrados. (c) Lineales o No Lineales. (d) Estacionarios o Inestacionarios.

1. El modelo de la Ec.(1.1) suponiendo que $v(t)$ es una entrada externa.
2. El modelo de la Ec.(1.1) suponiendo que $v(t) = \sin(t)$ no es una entrada externa, sino que es parte del modelo.
3. El modelo de un péndulo dado por las EE/ES:

$$\begin{aligned}\dot{\theta}(t) &= \omega(t) \\ \dot{\omega}(t) &= -\frac{g}{l} \sin(\theta(t)) \\ x(t) &= l \cos(\theta(t)) \\ y(t) &= l \sin(\theta(t))\end{aligned}\tag{P1.1a}$$

donde $\theta(t)$ y $\omega(t)$ son la posición y velocidad angular del péndulo, g es la aceleración de la gravedad y l es la longitud del cable.

4. El modelo del calentamiento de una barra cilíndrica de cobre debido a la circulación de corriente eléctrica, dado por la ecuación:

$$\frac{\partial T(r, t)}{\partial t} = \sigma \left(\frac{\partial^2 T(r, t)}{\partial r^2} + \frac{1}{r} \cdot \frac{\partial T(r, t)}{\partial r} + \frac{P_e(t)}{\lambda \cdot V} \right)\tag{P1.1b}$$

donde $T(r, t)$ es la temperatura en el tiempo t a una distancia r del centro de la barra, $P_e(t)$ es la potencia eléctrica (entrada externa) que se disipa en la barra, mientras que σ (coeficiente de difusión), λ (conductividad) y V (volumen de la barra) son parámetros constantes.

El modelo se completa con las *condiciones de borde*:

$$\begin{aligned}\frac{\partial T(0, t)}{\partial r} &= 0.0 \\ \frac{\partial T(R, t)}{\partial r} &= -k_1 (T(R, t)^4 - T_r(t)^4) - k_2 (T(R, t) - T_r(t))\end{aligned}\tag{P1.1c}$$

donde R es el radio de la barra, $T_r(t)$ es la temperatura ambiente y los coeficientes k_1 y k_2 son constantes.

5. El modelo de las Ecs.(P1.1b)–(P1.1c) suponiendo ahora que $k_1 = 0$.
6. El modelo dado por la Ecuación en Derivadas Parciales

$$\frac{\partial^2 \phi(x, y)}{\partial x^2} + \frac{\partial^2 \phi(x, y)}{\partial y^2} = \rho(x, y) \quad (\text{P1.1d})$$

que representa el potencial eléctrico en un espacio de dos dimensiones (x, y) y donde $\rho(x, y)$ es la densidad de carga (que no cambia con el tiempo).

[P1.2] Orden de Modelos Matemáticos

Determinar el orden de los modelos matemáticos dinámicos del Problema P1.1. En cada caso, además, proponer un vector de estados minimal.

[P1.3] Orden de Modelos Matemáticos II

Determinar el orden de los siguientes modelos matemáticos, proponiendo además en cada caso un vector de estados minimal.

1. El modelo del circuito de La Fig.1.2, dado por la Ec.(P1.3a):

$$\begin{aligned} C_1 \dot{u}_{C_1}(t) - i_{C_1}(t) &= 0 \\ C_2 \dot{u}_{C_2}(t) - i_{C_2}(t) &= 0 \\ u_R(t) - R i_R(t) &= 0 \\ i_R(t) + i_{C_1}(t) + i_{C_2}(t) &= 0 \\ u_{C_1}(t) - u_R(t) &= 0 \\ u_{C_1}(t) - u_{C_2}(t) &= 0 \end{aligned} \quad (\text{P1.3a})$$

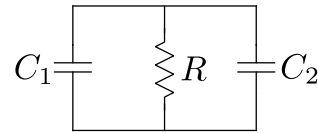


Figura 1.2: Circuito RC Paralelo

2. El modelo del péndulo de la Fig.1.3, dado por la Ec.(P1.3b).

$$\begin{aligned} \dot{x}(t) - v_x(t) &= 0 \\ \dot{y}(t) - v_y(t) &= 0 \\ m \dot{v}_x(t) + \frac{x(t)}{L} F(t) &= 0 \\ m \dot{v}_y(t) + \frac{y(t)}{L} F(t) - m g &= 0 \\ x(t)^2 + y(t)^2 - L^2 &= 0 \end{aligned} \quad (\text{P1.3b})$$

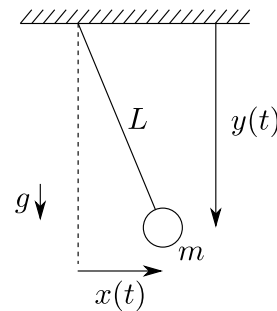


Figura 1.3: Péndulo

[P1.4] Relaciones Constitutivas y Estructurales

1. Clasificar las ecuaciones del modelo matemático de la Ec.(P1.3a) según sean constitutivas o estructurales.
2. Repetir el punto anterior para el modelo de la Ec.(P1.3b).

Capítulo 2

Simulación de Sistemas Continuos

En el Capítulo 1 mencionamos que una de las principales razones por las que se utilizan modelos matemáticos es por la posibilidad de experimentar sobre los mismos sin recurrir al sistema real. Estos experimentos que se realizan sobre los modelos matemáticos se denominan *simulaciones* y cuando los modelos son de tiempo continuo y toman la forma de ecuaciones diferenciales, dichas simulaciones involucran distintas estrategias de aproximación numérica ya que, en general, las ecuaciones diferenciales no pueden resolverse de manera analítica.

Este capítulo, basado principalmente en el libro *Continuous System Simulation* [CK06], tiene el objetivo de introducir de manera breve las bases de la integración numérica de ecuaciones diferenciales ordinarias y su uso en el contexto de la simulación de sistemas de tiempo continuo.

2.1. Principios de la Integración Numérica de ODEs

Consideremos a modo de ejemplo un sistema-masa-resorte-amortiguamiento muy simple como el de la Figura 2.1, que puede representarse por el siguiente sistema de ecuaciones:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= \frac{1}{m}[-k x_1(t) - b x_2(t) + F(t)] \end{aligned} \quad (2.1)$$

donde la variable $x_1(t)$ representa la posición, $x_2(t)$ representa la velocidad y $F(t)$ es la fuerza de entrada aplicada al sistema. Los parámetros son la masa m , el coeficiente de roce b y la constante del resorte k .

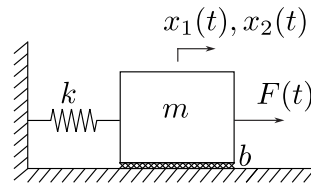


Figura 2.1: Sistema Masa Resorte

Si queremos determinar como evolucionarán las distintas variables de este sistema a medida que pasa el tiempo, deberemos resolver la Ecuación (2.1). En este caso, por ejemplo, tomando parámetros $m = b = k = 1$, suponiendo que la entrada es constante $F(t) = 1$, y asumiendo que $x_1(0) = x_2(0) = 0$, obtenemos la siguiente solución:

$$\begin{aligned} x_1(t) &= 1 - \frac{\sqrt{3}}{3} e^{-t/2} \sin \frac{\sqrt{3}}{2} t - e^{-t/2} \cos \frac{\sqrt{3}}{2} t \\ x_2(t) &= \frac{\sqrt{12}}{3} e^{-t/2} \sin \frac{\sqrt{3}}{2} t \end{aligned} \quad (2.2)$$

para todo $t \geq 0$. La gráfica de esta solución puede verse en la Figura 2.2

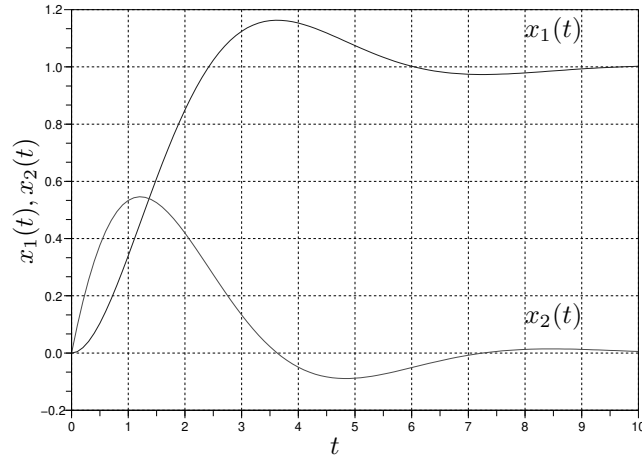


Figura 2.2: Solución de la Ecuación (2.1) del Sistema Masa Resorte

Si bien en este caso fue posible resolver de manera *analítica* la ecuación diferencial y obtener así la evolución de las variables en el tiempo, en general esto no será posible. En presencia de no linealidades, excepto casos muy simples y triviales, no es posible obtener una expresión analítica para la solución de una Ecuación Diferencial Ordinaria.

En los casos lineales, aún siendo posible resolver las ecuaciones (usando la Transformada de Laplace, por ejemplo), es muchas veces engorroso y poco práctico debido a la complejidad de los cálculos y de las expresiones resultantes.

Por todos estos motivos, generalmente se recurre a distintos métodos que bajo ciertas condiciones permiten obtener soluciones aproximadas, normalmente mediante el uso de computadoras. Estos algoritmos, denominados *métodos de integración numérica* para ecuaciones diferenciales ordinarias constituyen la herramienta básica fundamental para la *Simulación de Sistemas Continuos*.

En general, nos interesará obtener las trayectorias de un sistema de ecuaciones de la forma

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2, \dots, x_n, u_1, \dots, u_m, t) \\ \dot{x}_2 &= f_2(x_1, x_2, \dots, x_n, u_1, \dots, u_m, t) \\ &\vdots \\ \dot{x}_n &= f_n(x_1, x_2, \dots, x_n, u_1, \dots, u_m, t) \end{aligned}$$

donde las x_i son las variables de estado y $u_i(t)$ son las entradas. Este sistema de ecuaciones puede reescribirse en forma vectorial como sigue:

$$\dot{\mathbf{x}}(t) = \hat{\mathbf{f}}(\mathbf{x}(t), \mathbf{u}(t), t)$$

donde definimos $\mathbf{x}(t) \triangleq [x_1, x_2, \dots, x_n]^T$, $\mathbf{u}(t) \triangleq [u_1, \dots, u_m]^T$ y $\hat{\mathbf{f}}(\cdot) \triangleq [f_1, \dots, f_n]^T$.

Para poder calcular una solución numérica, será necesario siempre conocer completamente el vector de trayectorias de entrada $\mathbf{u}(t)$. Asumiendo entonces que este vector es conocido, no tiene sentido expresarlo como una variable. Por este motivo, en la literatura sobre métodos de integración numérica se define

$$\mathbf{f}(\mathbf{x}(t), t) \triangleq \hat{\mathbf{f}}(\mathbf{x}(t), \mathbf{u}(t), t)$$

de manera que el efecto de las entradas queda contenido en la propia definición de la función \mathbf{f} .

En virtud de esto, el objetivo de los métodos de integración numérica será encontrar una solución aproximada del sistema de ecuaciones diferenciales:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t) \tag{2.3}$$

a partir de la condición inicial conocida

$$\mathbf{x}(t_0) = \mathbf{x}_0 \quad (2.4)$$

2.1.1. Métodos de Euler

El método más sencillo para resolver la Ec.(2.3) fue propuesto por Leonhard Euler en 1768. Consiste básicamente en aproximar la derivada por el cociente incremental en la ecuación mencionada:

$$\dot{\mathbf{x}}(t) \approx \frac{\mathbf{x}(t+h) - \mathbf{x}(t)}{h} \approx \mathbf{f}(\mathbf{x}(t), t)$$

de donde despejamos

$$\mathbf{x}(t+h) \approx \mathbf{x}(t) + h \mathbf{f}(\mathbf{x}(t), t)$$

Llamando entonces $t_k = t_0 + k h$ para $k = 0, 1, \dots$ y definiendo $\mathbf{x}_k \triangleq \mathbf{x}(t_k)$, resulta la fórmula de *Forward Euler*:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h \mathbf{f}(\mathbf{x}_k, t_k) \quad (2.5)$$

El parámetro h , que define la distancia entre dos instantes de tiempo donde calculamos la solución, se denomina *paso de integración*. Más adelante veremos que h podrá ser constante o variable según el caso, y estudiaremos formas de elegir su valor adecuado.

El algoritmo de Forward Euler es entonces trivial: dado \mathbf{x}_0 y elegido un valor de h , hacemos $k = 0$ y utilizamos la fórmula anterior para calcular \mathbf{x}_1 . Luego, a partir de este valor de \mathbf{x}_1 calculamos \mathbf{x}_2 y así sucesivamente.

El algoritmo terminará cuando llegemos al paso en que $t_k = t_f$, donde el parámetro t_f se denomina *tiempo final* de la simulación. También veremos más adelante como elegir el tiempo final.

Para el sistema masa resorte de la Ec.(2.1), el método de Forward Euler con paso $h = 0.1$ comenzaría con los siguientes cálculos:

$$\begin{aligned} \mathbf{x}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} &\Rightarrow \mathbf{f}(\mathbf{x}_0, t_0) = \begin{bmatrix} x_2 = 0 \\ 1 - x_1 - x_2 = 1 \end{bmatrix} \Rightarrow \mathbf{x}_1 = \mathbf{x}_0 + h \mathbf{f}(\mathbf{x}_0, t_0) = \begin{bmatrix} 0 + 0.1 \cdot 0 \\ 0 + 0.1 \cdot 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} \\ \mathbf{x}_1 = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} &\Rightarrow \mathbf{f}(\mathbf{x}_1, t_1) = \begin{bmatrix} x_2 = 0.1 \\ 1 - x_1 - x_2 = 1 - 0 - 0.1 = 0.9 \end{bmatrix} \Rightarrow \mathbf{x}_2 = \mathbf{x}_1 + h \mathbf{f}(\mathbf{x}_1, t_1) = \begin{bmatrix} 0 + 0.1 \cdot 0.1 \\ 0.1 + 0.1 \cdot 0.9 \end{bmatrix} = \\ &= \begin{bmatrix} 0.01 \\ 0.19 \end{bmatrix} \end{aligned}$$

Esto es, de acuerdo al método de Forward Euler, $\mathbf{x}(t_1 = 0.1) \approx [0 \ 0.1]^T$ y $\mathbf{x}(t_2 = 0.2) \approx [0.01 \ 0.19]^T$.

La Figura 2.3 muestra el resultado de continuar calculando la solución de la Ec.(2.1) con este método numérico usando un paso de integración $h = 0.1$ y compara dicha solución con la analítica.

Como bien puede apreciarse en la gráfica, la solución numérica brindada por el método de Euler se aproxima bastante a la solución analítica. Más adelante analizaremos las causas del error y como se puede mejorar.

Otro método de integración numérica muy conocido, inspirado en el anterior y que recibe el nombre de *Método de Backward Euler*, reemplaza la fórmula de la Ec.(2.5) por la siguiente:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h \mathbf{f}(\mathbf{x}_{k+1}, t_k) \quad (2.6)$$

El método de Backward Euler tiene un pequeño inconveniente: de acuerdo a la Ec.(2.6) para calcular \mathbf{x}_{k+1} necesitamos conocer \mathbf{x}_{k+1} .

Naturalmente, conociendo \mathbf{x}_k podemos resolver de alguna forma la Ec.(2.6) y obtener de allí el valor de \mathbf{x}_{k+1} . Por este motivo, se dice que el método de Backward Euler es un *Método Implícito*, ya que para encontrar cada valor debemos resolver una ecuación. En contraposición, el método de Forward Euler es un *Método Explícito* ya que cada valor de la solución numérica queda determinado sin necesidad de resolver ecuaciones algebraicas.

Cuando la función \mathbf{f} es lineal, la resolución de la ecuación implícita (2.6) es muy sencilla (si bien requiere invertir una matriz). Por el contrario, en el caso no lineal, en general necesitaremos utilizar algún

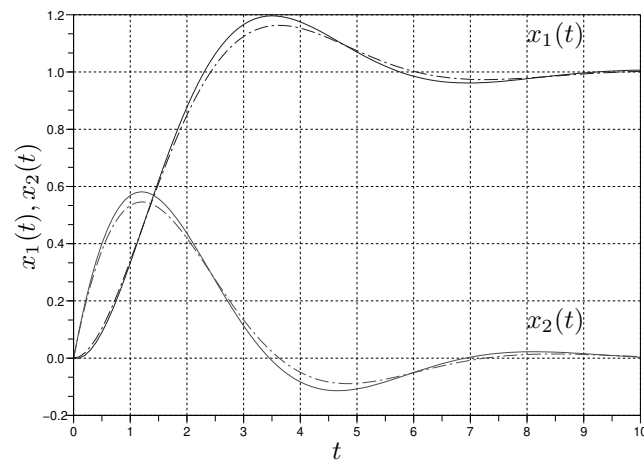


Figura 2.3: Solución de la Ecuación (2.1) del Sistema Masa Resorte con Forward Euler (sólida) con $h = 0.1$ y Solución Analítica (punteada)

algoritmo iterativo que encuentre la solución para cada instante de tiempo. Normalmente, se utiliza la *iteración de Newton*.

Por lo tanto, la implementación práctica del método de Backward Euler es mucho más compleja y computacionalmente costosa que la de Forward Euler. Más adelante, veremos que ventajas traerá que harán que valga la pena su uso.

La Figura 2.4 muestra el resultado de calcular la solución con Backward Euler usando un paso de integración $h = 0.1$ y compara dicha solución con la analítica. Como puede observarse, no hay una diferencia perceptible en cuanto a la precisión si se compara con lo que obtuvimos usando Forward Euler.

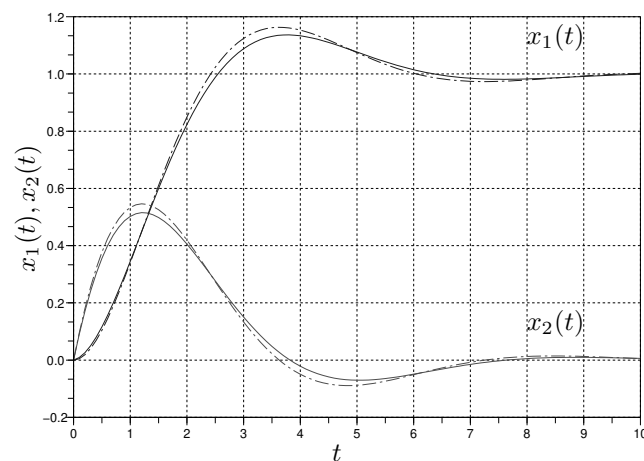


Figura 2.4: Solución de la Ecuación (2.1) del Sistema Masa Resorte con Backward Euler (sólida) con $h = 0.1$ y Solución Analítica (punteada)

2.1.2. La Iteración de Newton

Si queremos implementar de manera práctica el método de Backward Euler para simular sistemas no lineales, deberemos programar algún algoritmo que pueda *despejar* \mathbf{x}_{k+1} de la Ec.(2.6). El más utilizado para este propósito, tanto para Backward Euler como para cualquier otro método *implícito* es la iteración de Newton.

Supongamos que tenemos una ecuación algebraica

$$\mathbf{F}(\mathbf{x}) = 0 \quad (2.7)$$

y queremos encontrar una raíz de la misma, es decir, un valor del vector \mathbf{x} que hace que dicha ecuación se cumpla. La iteración de Newton brinda un procedimiento que, partiendo de un valor inicial \mathbf{x}^0 , permite obtener una sucesión de vectores $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N$, que bajo ciertas condiciones se aproxima cada vez más a la solución \mathbf{x} . La iteración está dada por la fórmula:

$$\mathbf{x}^{l+1} = \mathbf{x}^l - \left(\frac{\partial \mathbf{F}}{\partial \mathbf{x}} \Big|_{\mathbf{x}^l} \right)^{-1} \mathbf{F}(\mathbf{x}^l) \quad (2.8)$$

En la práctica, las iteraciones se realizan hasta que la diferencia entre \mathbf{x}^{l+1} y \mathbf{x}^l se torna más pequeña que cierta tolerancia pre-establecida.

Para el caso particular de Backward Euler, la Ec.(2.6) puede reescribirse como

$$\mathbf{x}_{k+1} - \mathbf{x}_k - h \mathbf{f}(\mathbf{x}_{k+1}, t_k) = \mathbf{F}(\mathbf{x}_{k+1}) = 0 \quad (2.9)$$

donde la incógnita es \mathbf{x}_{k+1} por lo que la iteración resulta

$$\mathbf{x}_{k+1}^{l+1} = \mathbf{x}_{k+1}^l - \left(I - h \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_{\mathbf{x}_{k+1}^l} \right)^{-1} (\mathbf{x}_{k+1}^l - \mathbf{x}_k - h \mathbf{f}(\mathbf{x}_{k+1}^l, t_k)) \quad (2.10)$$

Sobre este procedimiento, valen los siguientes comentarios:

- Para que esta iteración converja, es importante que la misma comience desde un punto inicial \mathbf{x}_{k+1}^0 no muy lejano a la solución. Por esto suele utilizarse $\mathbf{x}_{k+1}^0 = \mathbf{x}_k$.
- En cada iteración debemos calcular la matriz *Jacobiana* $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ evaluada en \mathbf{x}_{k+1}^l y luego invertir la matriz $(I - h \frac{\partial \mathbf{f}}{\partial \mathbf{x}})$. Esto suele tener un costo computacional muy elevado, sobre todo cuando el sistema es de orden alto.
- Para evitar este costo, se suele calcular la matriz Jacobiana sólo en la primer iteración.
- Para las precisiones que se utilizan habitualmente en aplicaciones de ingeniería, el procedimiento suele converger tras 4 o 5 iteraciones.
- Si el sistema fuera lineal, puede verse que la iteración converge al valor exacto tras una sola iteración.

2.1.3. Precisión de las Aproximaciones

La Figura 2.5 muestra el resultado de calcular la solución del sistema Masa Resorte con distintos pasos de integración usando el método de Forward Euler. Como puede observarse, el error de la aproximación crece a medida que aumentamos el paso de integración h .

Trataremos entonces de justificar teóricamente esta observación.

Conocida la solución analítica en un punto $\mathbf{x}(t_k)$, podemos calcular la solución analítica tras un paso de integración de valor h haciendo uso de la expansión en serie de Taylor en torno a t_k :

$$\begin{aligned} \mathbf{x}(t_{k+1}) &= \mathbf{x}(t_k) + h \frac{d\mathbf{x}}{dt}(t_k) + \frac{h^2}{2!} \frac{d^2\mathbf{x}}{dt^2}(t_k) + \frac{h^3}{3!} \frac{d^3\mathbf{x}}{dt^3}(t_k) + \dots \\ &= \mathbf{x}(t_k) + h \mathbf{f}(\mathbf{x}(t_k), t_k) + \frac{h^2}{2!} \frac{d^2\mathbf{x}}{dt^2}(t_k) + \frac{h^3}{3!} \frac{d^3\mathbf{x}}{dt^3}(t_k) + \dots \\ &= \mathbf{x}(t_k) + h \mathbf{f}(\mathbf{x}(t_k), t_k) + o(h^2) \end{aligned}$$

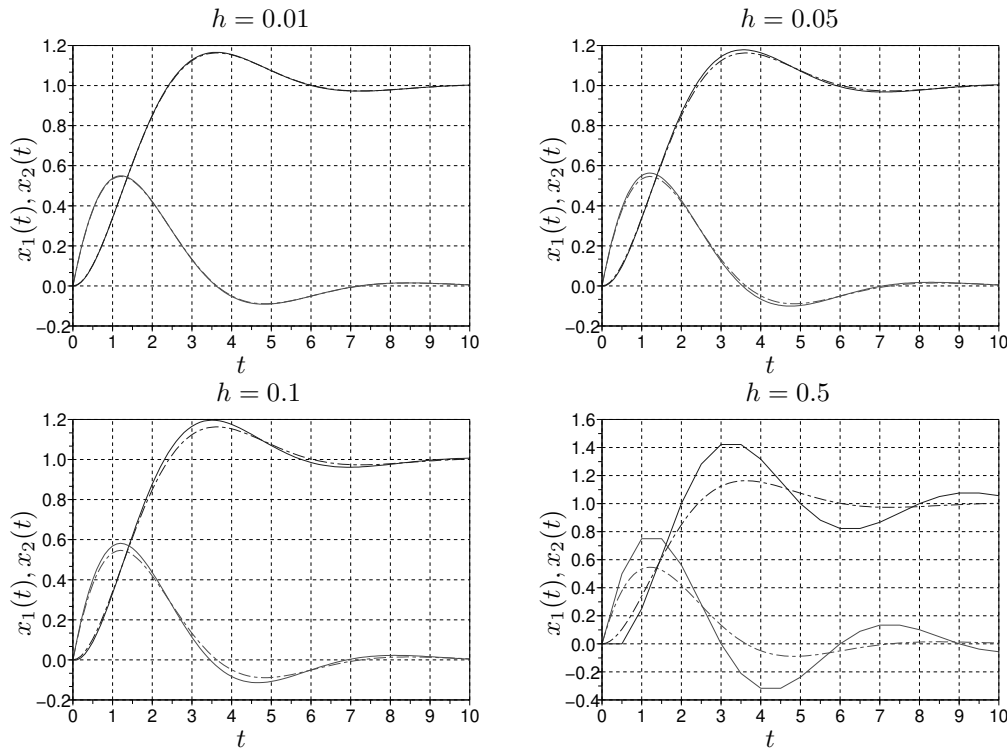


Figura 2.5: Solución de la Ecuación (2.1) del Sistema Masa Resorte con Forward Euler (sólida) para distintos pasos de integración y Solución Analítica (punteada)

Como podemos ver en esta expresión, el método de Forward Euler utiliza la expansión de Taylor sólo hasta el término de h , ignorando los términos de h^2 , h^3 , etc. Dado que el método iguala la serie hasta el término de h^1 , se dice que es un algoritmo de primer orden.

El error que comete el método de Euler tras el primer paso es del orden de h^2 . Esto es, asumiendo que se conoce la solución analítica en t_k el método de Euler cometerá un error proporcional a h^2 en t_{k+1} , es decir, en un paso del algoritmo.

Este error (cometido tras un paso) se denomina *Error Local por Truncamiento*¹, y en general aumentará a medida que se aumente el paso de integración. Este error siempre será proporcional a h^{N+1} donde N es el orden del método de integración.

En realidad, salvo en el primer paso, no conoceremos el valor de la solución analítica en t_k . Por lo tanto, en t_{k+1} habrá un error adicional ya que en cada caso estaremos partiendo de un valor inexacto en t_k . Por lo tanto, la diferencia que observamos entre la solución analítica y la solución numérica en un instante arbitrario de tiempo no es el error local, sino lo que se denomina *Error Global*.

Puede demostrarse que el máximo valor que alcanza el error global tiene siempre un orden menos que el error local. Esto es, en un método de orden N el error local es proporcional a h^{N+1} mientras que el máximo error global es proporcional a h^N .

En los métodos de Forward y Backward Euler, al ser ambos de primer orden, el máximo error global resulta proporcional al paso h . Por lo tanto, si disminuimos 10000 veces el paso de integración el error máximo disminuirá aproximadamente 10000 veces. Esto es, para realizar una simulación 10000 veces más precisa necesitamos usar un paso 10000 veces más chico, lo que implica realizar 10000 veces más cálculos.

En un método de cuarto orden, en cambio, disminuir 10 veces el paso de integración implica disminuir 10000 veces el error. Entonces, para realizar una simulación 10000 veces más precisa con un método de cuarto orden necesitamos usar un paso 10 veces más chico, lo que sólo implica realizar 10 veces más cálculos.

¹Se habla de truncamiento ya que se puede pensar que el método *trunca* la serie de Taylor a partir de h^2 .

Esta última observación tiene una implicancia importante: cuando queremos simular con mucha precisión debemos necesariamente utilizar métodos de orden alto. De otra manera, estaremos obligados a disminuir mucho el paso de integración lo que implica realizar muchos pasos y por lo tanto muchos cálculos. Si bien como veremos más adelante los métodos de orden mayor tienen mayor costo computacional por paso, este costo se compensa al utilizar pasos más largos obteniendo mejor precisión.

Si bien los errores de truncamiento son generalmente los más importantes, hay aplicaciones en las que no se pueden ignorar los *Errores de Redondeo*. Estos errores se deben a la representación numérica con un número finito de bits en las computadoras.

Para evitar estos errores siempre se intenta trabajar con representaciones de punto flotante de 64 bits (doble precisión), lo que permite en general despreocuparse de los efectos del redondeo. De todas maneras, en algunas aplicaciones que requieren gran precisión se utilizan métodos de orden muy alto (para dinámica de cuerpos celestes, por ejemplo) y los efectos del redondeo no pueden ignorarse por completo.

Una característica interesante de los errores de redondeo es que, a diferencia de los errores por truncamiento, tienden a aumentar a medida que el paso de integración disminuye. Una explicación simplificada de este fenómeno puede deducirse de observar que ocurre en la Fórmula de Euler (2.5) al disminuir h . Cuando sumamos en punto flotante un número grande con uno muy pequeño, debido al redondeo, obtenemos nuevamente el número grande. Si repetimos indefinidamente la operación, seguiremos por lo tanto obteniendo siempre el primer número grande como resultado.

2.1.4. Estabilidad Numérica

Una pregunta que surge naturalmente tras analizar el error global de un método es si el mismo se sigue acumulando indefinidamente o no a medida que avanza el tiempo de la simulación. Observando nuevamente la Figura 2.5, vemos que a medida que la solución analítica se acerca a su valor final, la solución numérica tiende al mismo valor.

En otras palabras, en la Figura 2.5 vemos que la solución numérica preserva la estabilidad y el punto de equilibrio de la solución analítica.

Ahora bien, ¿ocurre esto para cualquier paso de integración?. La Figura 2.6 muestra que ocurre si seguimos aumentando el paso h .

Evidentemente no siempre se preserva la estabilidad. Para $h = 1$ perdemos la *estabilidad asintótica* y para $h = 1.5$ la solución numérica es claramente inestable.

Veamos que ocurre si, en cambio, utilizamos el método de Backward Euler. La Figura 2.7 muestra los resultados de este método implícito y como puede observarse, la estabilidad se preserva en todos los casos.

¿Como podemos justificar teóricamente estos resultados?

Si aplicamos el método de Forward Euler a un sistema lineal y estacionario,

$$\dot{\mathbf{x}}(t) = A \mathbf{x}(t) + B \mathbf{u}(t) \quad (2.11)$$

resulta,

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h (A \mathbf{x}_k + B \mathbf{u}_k) = (I + h A) \mathbf{x}_k + h B \mathbf{u}_k$$

Es decir, la ecuación en diferencias resultante tiene matriz de evolución

$$A_d = (I + h A)$$

Los autovalores λ_d de A_d serán las soluciones de

$$\det(\lambda_d I - A_d) = \det(\lambda_d I - I - h A) = 0$$

Si el determinante de una matriz es nulo, también lo es el determinante del producto por h^{-1} , por lo que,

$$\det(h^{-1} (\lambda_d - 1) I - A) = 0$$

de donde resulta que los autovalores de A_d y los de A se relacionan según:

$$h^{-1} (\lambda_d - 1) = \lambda$$

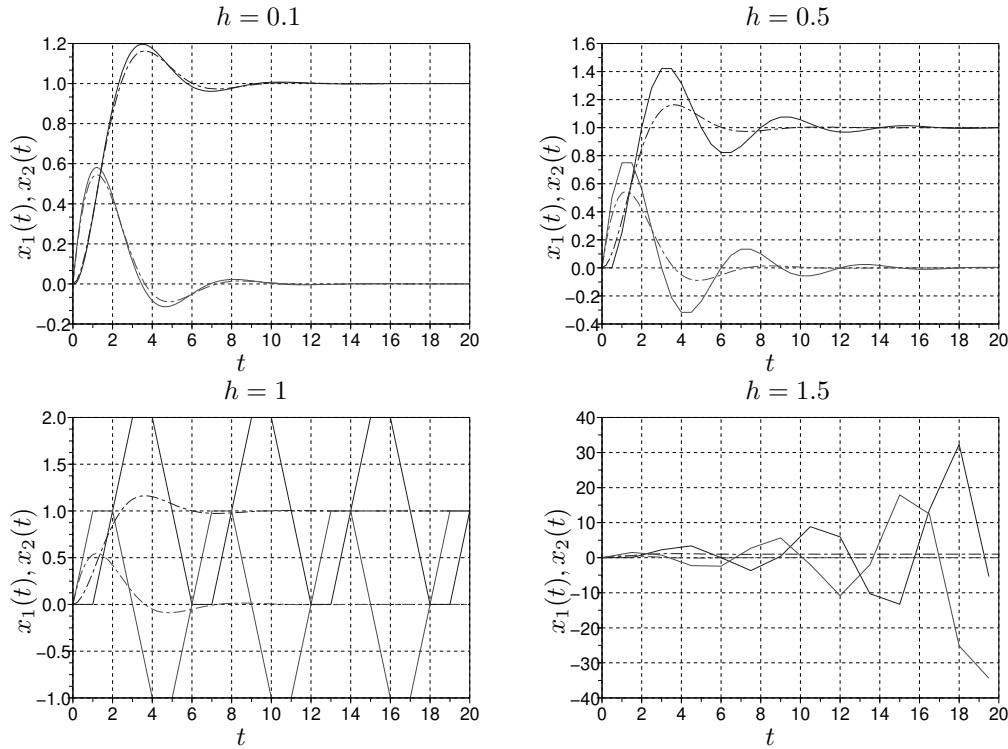


Figura 2.6: Solución de la Ecuación (2.1) del Sistema Masa Resorte con Forward Euler (sólida) para distintos pasos de integración y Solución Analítica (punteada)

de donde,

$$\lambda_d = \lambda h + 1 \quad (2.12)$$

Es decir, los autovalores de la matriz del sistema discreto aproximado pueden calcularse directamente en función de los autovalores del sistema continuo original.

Si aproximamos la solución de un sistema estable, será deseable que la aproximación resulte también estable. Para esto, deberá cumplirse para todos los autovalores de A que

$$|\lambda h + 1| < 1 \quad (2.13)$$

lo que sólo ocurrirá para valores pequeños de h . Esta condición de estabilidad del método deberá cumplirse siempre y es fundamental a la hora de elegir el paso de integración.

El sistema masa resorte del ejemplo tiene autovalores $\lambda_{1,2} = -0.5 \pm j\frac{\sqrt{3}}{2}$. Puede verse entonces que la condición de estabilidad de la Ec. (2.13) se cumple sólo para $h < 1$, lo que corrobora los resultados de la Fig.2.6.

Si aplicamos en cambio Backward Euler al sistema lineal y estacionario de la Ec.(2.11), resulta

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k + h (A \mathbf{x}_{k+1} + B \mathbf{u}_{k+1}) \Rightarrow \\ (I - h A) \mathbf{x}_{k+1} &= \mathbf{x}_k + h B \mathbf{u}_{k+1} \\ \mathbf{x}_{k+1} &= (I - h A)^{-1} [\mathbf{x}_k + h B \mathbf{u}_{k+1}] \end{aligned}$$

de donde la ecuación en diferencias resultante tiene matriz de evolución

$$A_d = (I - h A)^{-1}$$

Los autovalores λ_d de A_d serán las soluciones de

$$\det(\lambda_d I - A_d) = \det(\lambda_d I - (I - h A)^{-1}) = 0$$

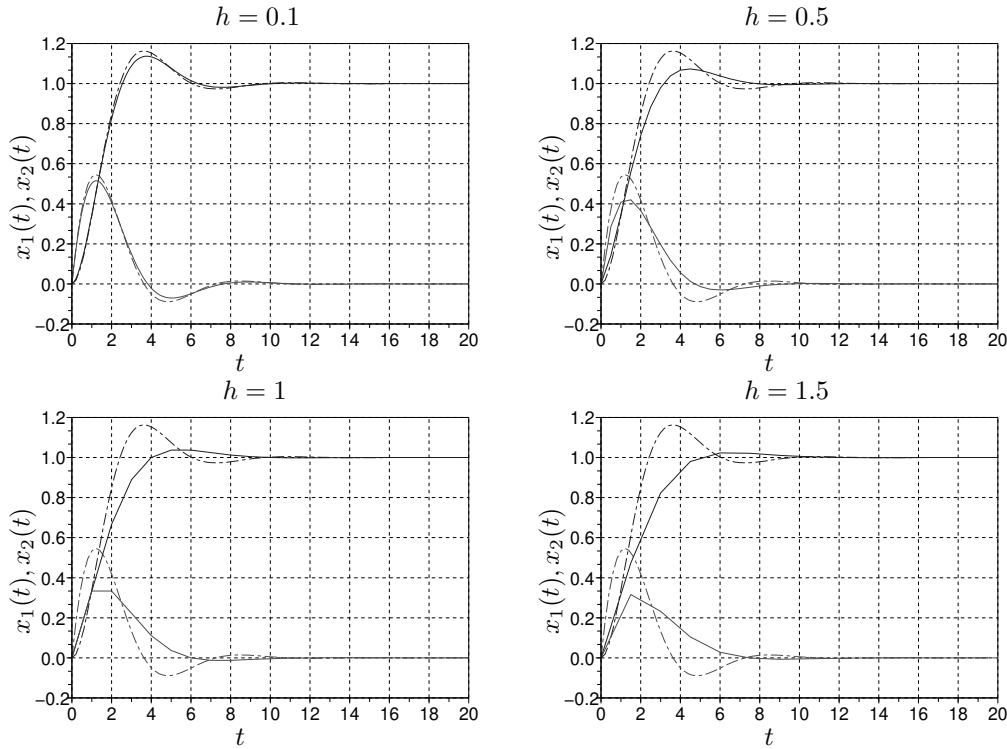


Figura 2.7: Solución de la Ecuación (2.1) del Sistema Masa Resorte con Backward Euler (sólida) para distintos pasos de integración y Solución Analítica (punteada)

Si el determinante de una matriz es nulo, también lo es el determinante del producto por $\frac{(I-hA)}{h\lambda_d}$ de donde,

$$\det\left(\lambda_d \frac{(I-hA)}{h\lambda_d} - \frac{I}{h\lambda_d}\right) = \det\left(I \left(\frac{1}{h} - \frac{1}{h\lambda_d}\right) - A\right) = 0$$

de donde resulta que los autovalores de A_d y los de A se relacionan según:

$$\frac{1}{h} - \frac{1}{h\lambda_d} = \lambda$$

o bien

$$\lambda_d = \frac{1}{1 - \lambda h}$$

Por lo tanto, la condición para que el resultado numérico sea estable es que

$$\left|\frac{1}{1 - \lambda h}\right| < 1 \quad (2.14)$$

Puede verse fácilmente que cuando $\text{Re}(\lambda) < 0$, la condición se cumplirá siempre. Esto es, al simular un sistema estable con el método de Backward Euler, la solución numérica siempre resultará estable (hecho que corrobora lo observado en la Fig.2.7).

Esta propiedad de preservar la estabilidad numérica es la principal ventaja de Backward Euler sobre Forward Euler. Más adelante veremos que esta característica será imprescindible para poder simular adecuadamente ciertos sistemas.

Entre tanto, una desventaja del algoritmo de Backward Euler es que para autovalores inestables $\text{Re}(\lambda) > 0$ puede resultar (si h es muy grande) $|\lambda_d| < 1$ (estable). Es decir, la simulación de un sistema inestable puede resultar estable, lo cual es indeseado.

Por ejemplo, el sistema inestable

$$\dot{x}(t) = x(t)$$

tiene un autovalor $\lambda = 1$. Por lo tanto, usando cualquier paso $h > 2$, vemos que la condición de la Ec.(2.14) se cumple lo que indica que la solución numérica será estable.

2.2. Métodos Monopaso

Como mencionamos antes, los métodos de Forward y Backward Euler que vimos hasta aquí realizan sólo aproximaciones de primer orden. Debido a esto, si se quiere obtener una buena precisión en la simulación, se debe reducir excesivamente el paso de integración lo que implica una cantidad de pasos y de cálculos en general inaceptable.

Para obtener aproximaciones de orden mayor, será necesario utilizar más de una evaluación de la función $\mathbf{f}(\mathbf{x}, t)$ en cada paso. Cuando dichas evaluaciones se realizan de manera tal que para calcular \mathbf{x}_{k+1} sólo se utiliza el valor de \mathbf{x}_k , se dice que el algoritmo es *monopaso*. Por el contrario, cuando se utilizan además valores anteriores de la solución (\mathbf{x}_{k-1} , \mathbf{x}_{k-2} , etc.), se dice que el algoritmo es *multipaso*.

Los métodos monopaso se suelen denominar también *Métodos de Runge-Kutta*, ya que el primero de estos métodos de orden alto fue formulado por Runge y Kutta a finales del siglo XIX.

2.2.1. Métodos Explícitos de Runge Kutta

Los métodos explícitos de Runge-Kutta realizan varias evaluaciones de la función $\mathbf{f}(\mathbf{x}, t)$ en cercanías del punto (\mathbf{x}_k, t_k) y luego calculan \mathbf{x}_{k+1} realizando una suma pesada de dichas evaluaciones.

Uno de los algoritmos más simple es el denominado *Método de Heun*, inventado por Heun en el año 1900. Dicho método utiliza la fórmula

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{h}{2} (\mathbf{k}_1 + \mathbf{k}_2) \quad (2.15)$$

donde

$$\begin{aligned} \mathbf{k}_1 &= \mathbf{f}(\mathbf{x}_k, t_k) \\ \mathbf{k}_2 &= \mathbf{f}(\mathbf{x}_k + h \mathbf{k}_1, t_k + h) \end{aligned} \quad (2.16)$$

Comparado con Euler, este método realiza una evaluación adicional de la función \mathbf{f} para calcular \mathbf{k}_2 . El beneficio que se obtiene es que el método de Heun realiza una aproximación de segundo orden y resulta mucho más preciso que Euler.

Para el sistema masa resorte de la Ec.(2.1), el método de Heun con paso $h = 0.1$ comenzaría con los siguientes cálculos:

$$\begin{aligned} \mathbf{x}_0 &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow \\ \mathbf{k}_1 &= \mathbf{f}(\mathbf{x}_0, t_0) = \begin{bmatrix} x_2 \\ 1 - x_1 - x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \mathbf{x}_0 + h \mathbf{k}_1 &= \begin{bmatrix} 0 + 0.1 \cdot 0 \\ 0 + 0.1 \cdot 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} \Rightarrow \\ \mathbf{k}_2 &= \mathbf{f}(\mathbf{x}_0 + h \mathbf{k}_1, t_0 + h) = \begin{bmatrix} 0.1 \\ 1 - 0 - 0.1 \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix} \\ \mathbf{x}_1 &= \mathbf{x}_0 + \frac{h}{2} (\mathbf{k}_1 + \mathbf{k}_2) = \begin{bmatrix} 0 + 0.05 \cdot (0 + 0.1) \\ 0 + 0.05 \cdot (1 + 0.9) \end{bmatrix} = \begin{bmatrix} 0.005 \\ 0.095 \end{bmatrix} \end{aligned}$$

Otro método, uno de los más utilizados, es el de Runge-Kutta de orden 4 (RK4), desarrollado por Runge y Kutta en 1895:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{h}{6} (\mathbf{k}_1 + 2 \mathbf{k}_2 + 2 \mathbf{k}_3 + \mathbf{k}_4)$$

donde

$$\begin{aligned} \mathbf{k}_1 &= \mathbf{f}(\mathbf{x}_k, t_k) \\ \mathbf{k}_2 &= \mathbf{f}\left(\mathbf{x}_k + h \frac{\mathbf{k}_1}{2}, t_k + \frac{h}{2}\right) \\ \mathbf{k}_3 &= \mathbf{f}\left(\mathbf{x}_k + h \frac{\mathbf{k}_2}{2}, t_k + \frac{h}{2}\right) \\ \mathbf{k}_4 &= \mathbf{f}(\mathbf{x}_k + h \mathbf{k}_3, t_k + h) \end{aligned}$$

Este algoritmo, como vemos, utiliza cuatro evaluaciones de la función \mathbf{f} en cada paso. En este caso, el beneficio es que resulta una aproximación de orden 4.

La Tabla 2.1 resume los resultados de simular el sistema Masa Resorte de la Ec.(2.1) con los métodos de Euler, Heun y Runge–Kutta de orden 4. En cada caso, se reporta el máximo error obtenido en toda la simulación (esto es, el máximo error global). Puede notarse como en Euler el error disminuye linealmente con el paso, mientras que en Heun lo hace de forma cuadrática y en RK4 lo hace con la cuarta potencia del paso.

Paso	Error Euler	Error Heun	Error RK4
$h = 0.5$	0.298	0.0406	4.8×10^{-4}
$h = 0.1$	0.042	1.47×10^{-3}	6.72×10^{-7}
$h = 0.05$	0.0203	3.6×10^{-4}	4.14×10^{-8}
$h = 0.01$	3.94×10^{-3}	1.42×10^{-5}	6.54×10^{-11}

Tabla 2.1: Máximo Error Cometido por los Métodos de Euler, Heun y RK4 para Distintos Pasos de Integración con el Sistema Masa Resorte de le Ec.(2.1)

Hay infinidad de métodos de Runge Kutta explícitos para diversos órdenes (incluyendo métodos de órdenes mayores a 10). De todas maneras, los de orden 1 a 5 son los más utilizados en la práctica.

En lo referente a estabilidad, los métodos de RK tienen características similares a las de Forward Euler. Esto es, preservan la estabilidad para valores no muy grandes del paso de integración h . Por lo tanto, aunque la precisión sea buena, no podremos utilizar pasos muy grandes debido a los límites que impone la estabilidad.

Por ejemplo, para el método de Heun, podemos analizar la estabilidad sobre un sistema escalar de primer orden:

$$\dot{x} = \lambda x(t)$$

cuya solución analítica será estable si $\lambda < 0$. Aplicando Heun, tenemos,

$$\begin{aligned} x_{k+1} &= x_k + \frac{h}{2}(k_1 + k_2) \\ &= x_k + \frac{h}{2}(\lambda x_k + \lambda (x_k + h \lambda x_k)) \\ &= x_k + h \lambda x_k + \frac{h^2 \lambda^2}{2} x_k \\ &= \left(1 + h \lambda + \frac{h^2 \lambda^2}{2}\right) x_k \\ &= \lambda_d x_k \end{aligned}$$

es decir, el autovalor discreto resulta $\lambda_d = 1 + h \lambda + \frac{h^2 \lambda^2}{2}$, y la condición de estabilidad numérica es entonces

$$\left|1 + h \lambda + \frac{h^2 \lambda^2}{2}\right| < 1$$

que se traduce en dos desigualdades:

$$\begin{cases} 1 + h \lambda + \frac{h^2 \lambda^2}{2} < 1 \\ 1 + h \lambda + \frac{h^2 \lambda^2}{2} > -1 \end{cases}$$

La primera desigualdad conduce a la condición

$$h < \frac{2}{|\lambda|} \quad (2.17)$$

y la segunda se cumple siempre que $\lambda < 0$. En conclusión, para un autovalor real negativo, el método de Heun dará un resultado estable cuando el paso sea suficientemente chico de manera que se cumpla la Ec.(2.17). Este resultado es idéntico al de Forward Euler para autovalores reales negativos.

2.2.2. Métodos Monopaso Implícitos

Vimos que el método de Backward Euler era un algoritmo implícito cuya principal ventaja es preservar la estabilidad de la solución numérica para cualquier paso de integración. Sin embargo, al igual que Forward Euler, realiza sólo una aproximación de primer orden.

Hay diversos métodos implícitos monopaso de orden mayor que, al igual que Backward Euler, preservan la estabilidad.

Uno de los más utilizados es la *Regla Trapezoidal*, cuya definición es la siguiente:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{h}{2} [\mathbf{f}(\mathbf{x}_k, t_k) + \mathbf{f}(\mathbf{x}_{k+1}, t_{k+1})] \quad (2.18)$$

Este método implícito realiza una aproximación de segundo orden y tiene la propiedad (al menos en sistemas lineales y estacionarios) de que la solución numérica es estable si y sólo si la solución analítica es estable.

Por este motivo, la regla trapezoidal se suele utilizar a menudo para simular eficazmente sistemas muy oscilantes.

Si bien existen numerosos métodos implícitos monopaso, en la práctica no son tan utilizados ya que en general los métodos multipaso implícitos suelen ser más eficientes.

2.2.3. Algoritmos de Control de Paso

Hasta aquí consideramos siempre el paso h como un parámetro fijo que debe elegirse previo a la simulación. Sin embargo, en muchos casos es posible implementar algoritmos que cambien el paso de integración de forma automática a medida que avanza la simulación.

Los algoritmos de *control de paso* tienen por propósito mantener el error de simulación acotado para lo cual ajustan automáticamente el paso en función del error estimado. La idea es muy simple, en cada paso se hace lo siguiente:

1. Se da un paso con el método de integración elegido calculando \mathbf{x}_{k+1} y cierto paso h .
2. Se estima el error cometido.
3. Si el error es mayor que la tolerancia, se disminuye el paso de integración h y se recalcula \mathbf{x}_{k+1} volviendo al punto 1.
4. Si el error es menor que la tolerancia, se acepta el valor de \mathbf{x}_{k+1} calculado, se incrementa el paso h y se vuelve al punto 1 para calcular \mathbf{x}_{k+2} .

Si bien la idea es muy sencilla, de la descripción anterior surgen varias cuestiones:

- ¿Cómo estimamos el error?
- ¿Qué ley utilizamos para recalculamos el paso de integración h en cada caso?
- ¿Qué valor de paso de integración usamos inicialmente?

A continuación analizaremos en detalle dichos interrogantes.

Estima del Error

La estima del error de integración se hace utilizando dos métodos de orden diferente. Dado \mathbf{x}_k , calculamos, por ejemplo, \mathbf{x}_{k+1} usando un método de orden 4 y luego $\tilde{\mathbf{x}}_{k+1}$ con un método de orden 5. Dado que un método de orden 5 es mucho más preciso que uno de orden 4, podemos suponer que el error en el método de orden 4 es directamente la diferencia entre ambos.

Para ahorrar cálculos, se suelen buscar métodos que compartan evaluaciones comunes en las funciones, de manera que para evaluar $\tilde{\mathbf{x}}_{k+1}$ puedan reutilizarse algunos de los cálculos hechos para calcular \mathbf{x}_{k+1} .

Por ejemplo, las siguientes fórmulas calculan el método de Heun (2do orden) y un método de RK de tercer orden:

$$\begin{aligned} \mathbf{k}_1 &= \mathbf{f}(\mathbf{x}_k, t_k) \\ \mathbf{k}_2 &= \mathbf{f}(\mathbf{x}_k + h \mathbf{k}_1, t_k + h) \\ \mathbf{k}_3 &= \mathbf{f}\left(\mathbf{x}_k + \frac{h}{4} \mathbf{k}_1 + \frac{h}{4} \mathbf{k}_2, t_k + \frac{h}{2}\right) \end{aligned}$$

y luego

$$\mathbf{x}_{k+1} = x_k + \frac{h}{2}(\mathbf{k}_1 + \mathbf{k}_2) \tag{2.19}$$

$$\tilde{\mathbf{x}}_{k+1} = x_k + \frac{h}{6}(\mathbf{k}_1 + \mathbf{k}_2 + 4\mathbf{k}_3) \tag{2.20}$$

de donde puede estimarse el error como

$$\mathbf{e}_{k+1} \approx \mathbf{x}_{k+1} - \tilde{\mathbf{x}}_{k+1} = \frac{\mathbf{k}_1 + \mathbf{k}_2 - 2\mathbf{k}_3}{3}$$

Este método se denominará RK23, ya que se trata de un algoritmo de Runge–Kutta de orden 2 con estima de error de orden 3. Como puede observarse la Ec.(2.19) (Heun) y la Ec.(2.20) (RK3) comparten las etapas de cálculo de \mathbf{k}_1 y \mathbf{k}_2 , lo que ahorra bastante costo computacional.

Uno de los métodos más utilizados en la práctica es un RK45 denominado *Runge–Kutta–Fehlberg*, que utiliza 6 evaluaciones de la función \mathbf{f} para calcular dos soluciones, una de orden 4 y otra de orden 5.

Ajuste del Paso

Como vimos anteriormente, el error local que comete un método de orden N es proporcional a h^{N+1} , esto es,

$$err \approx c h^{N+1} \tag{2.21}$$

donde c es una constante desconocida. Si usamos cierto paso h y obtenemos un error err , la pregunta es, ¿que paso h_0 deberíamos usar para obtener un error tol , donde tol es la tolerancia de error que admitimos?

La respuesta a esta pregunta es lo que nos permitirá ajustar el paso tanto para disminuirlo como para aumentarlo.

Naturalmente, podemos escribir

$$tol \approx c h_0^{N+1}$$

Dividiendo miembro a miembro con la Ec.(2.21) resulta,

$$\frac{tol}{err} \approx \frac{h_0^{N+1}}{h^{N+1}}$$

de donde,

$$h_0 = h (tol/err)^{\frac{1}{N+1}}$$

que nos brinda la fórmula para ajustar el paso de acuerdo a la tolerancia admitida, al error estimado y al paso utilizado. Esta última ecuación generalmente se modifica de manera tal que

$$h_0 = 0.8 h (tol/err)^{\frac{1}{N+1}} \tag{2.22}$$

para asegurar que el paso utilizado no resulte demasiado cercano al límite teórico de precisión y evitar tener que recalcular pasos.

Si simulamos con esta idea el sistema masa resorte de la Ec.(2.1), utilizando el método de Runge–Kutta–Fehlberg (RK45) que mencionamos antes, obtenemos el resultado que se muestra en la Fig.2.8.

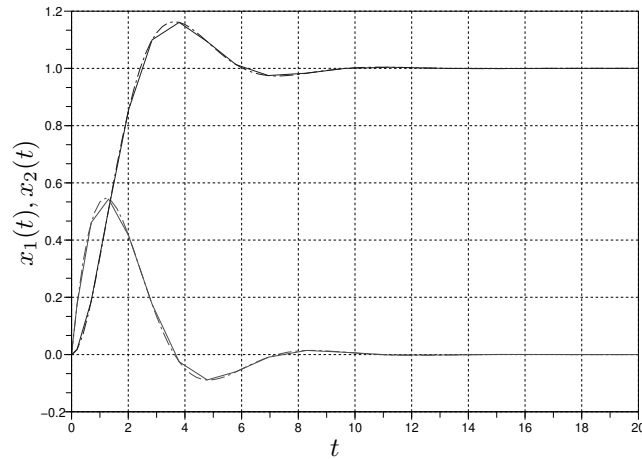


Figura 2.8: Solución de la Ecuación (2.1) del Sistema Masa Resorte con Runge–Kutta–Fehlberg (sólida) con $tol = 0.001$ y Solución Analítica (punteada)

El error es inapreciable aunque las curvas difieren ya que la solución numérica tiene muy pocos puntos (hay sólo 25 pasos) y al graficar se perciben las rectas que unen los puntos sucesivos.

La Figura 2.9 muestra la variación del paso de integración. A medida que el sistema se acerca al punto de equilibrio los pasos se hacen más grandes. Sin embargo, h no puede crecer demasiado debido a las limitaciones de estabilidad del método.

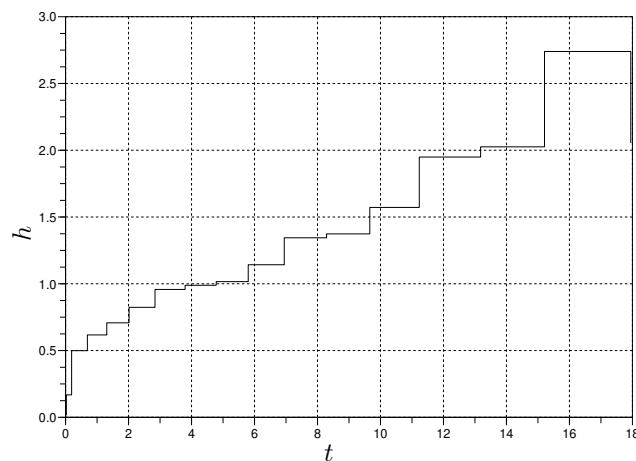


Figura 2.9: Variación del Paso de Integración en la Solución de la Fig.2.8

Algunos Parámetros de los Algoritmos de Control de Paso

Los parámetros que utilizan los algoritmos de control de paso son generalmente los siguientes:

Tolerancia relativa: Es el error que se admite, expresado como fracción de los valores de las variables. Por ejemplo, cuando se propone una tolerancia relativa de 0.001 (un valor típico), se está imponiendo que el error no debe superar el uno por mil del valor de la solución.

Tolerancia absoluta: Cuando las variables toman valores muy cercanos a cero, la tolerancia relativa impone un error demasiado pequeño que implicaría utilizar pasos exageradamente chicos. Por esto, se establece este parámetro adicional que es el mínimo valor de tolerancia a utilizar.

Máximo paso de integración: A veces, puede resultar que los dos métodos usados coincidan muy bien en el valor calculado y el error estimado sea cercano a cero. De acuerdo a la fórmula (2.22), en tal caso el valor de h resultará muy grande. Por eso, se suele limitar el paso a un valor máximo.

Mínimo paso de integración: En ciertas ocasiones, es posible que no se pueda alcanzar la tolerancia establecida sin utilizar pasos exageradamente pequeños. Esto traería como consecuencia que se trabe la simulación ya que deberían realizarse un número demasiado grande de pasos. Por esto, se suele limitar también por debajo el paso de integración. Este paso mínimo se suele también utilizar como valor inicial del paso.

Intervalo de Interpolación: Al utilizar control de paso, la solución se calcula a intervalos irregulares de tiempo. En muchas aplicaciones, es importante contar con la solución calculada en determinados instantes de tiempo generalmente equiespaciados. Para esto, se suelen utilizar algoritmos de interpolación que calculan la solución en los instantes deseados.

2.3. Métodos Multipaso

Los métodos monopaso obtienen aproximaciones de orden alto utilizando para esto varias evaluaciones de la función \mathbf{f} en cada paso. Para evitar este costo computacional adicional, se han formulado diversos algoritmos que, en lugar de evaluar repetidamente la función \mathbf{f} en cada paso, utilizan los valores evaluados en pasos anteriores.

Veremos a continuación algunos de estos algoritmos.

2.3.1. Métodos Multipaso Explícitos

Dentro de los métodos multipaso explícitos encontramos, entre los más utilizados, a los métodos de Adams–Bashforth (AB) y de Adams–Bashforth–Moulton (ABM).

El método de AB de 2do orden, por ejemplo, está dado por la fórmula:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{h}{2} (3\mathbf{f}_k - \mathbf{f}_{k-1}) \quad (2.23)$$

donde definimos

$$\mathbf{f}_k \triangleq \mathbf{f}(\mathbf{x}_k, t_k)$$

El método de AB de 4to orden, en tanto, es el que sigue:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{h}{24} (55\mathbf{f}_k - 59\mathbf{f}_{k-1} + 37\mathbf{f}_{k-2} - 9\mathbf{f}_{k-3}) \quad (2.24)$$

Como vemos, estos métodos requieren realizar una única evaluación de la función \mathbf{f} en cada paso. El orden alto de aproximación se logra utilizando la información de las evaluaciones anteriores de dicha función.

Un problema de los métodos multipaso es el arranque. La expresión de la Ec.(2.23) sólo puede utilizarse para calcular la solución a partir de $k = 2$. Para el primer pasos no está definido el valor de \mathbf{x}_{-1} requerido por la fórmula. Similarmente, el algoritmo de AB4 de la Ec.(2.24) sólo puede usarse a partir de $k = 4$ por la misma razón.

Por este motivo, los primeros pasos de los métodos multipaso deben realizarse con algún método monopaso de, al menos, el mismo orden de precisión. Por ejemplo, para el método de AB4 de la Ec.(2.24), deberían darse los tres primeros pasos con Runge–Kutta de orden 4.

Veamos por ejemplo, el resultado de utilizar AB2 para simular el sistema masa resorte del ejemplo. Usaremos como valores iniciales $\mathbf{x}_0 = [0 \ 0]$ y $\mathbf{x}_1 = [0.005 \ 0.095]$, este último calculado previamente con el método de Heun. Luego,

$$\begin{aligned}\mathbf{x}_0 &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow \mathbf{f}(\mathbf{x}_0, t_0) = \begin{bmatrix} x_2 \\ 1 - x_1 - x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \mathbf{x}_1 &= \begin{bmatrix} 0.005 \\ 0.095 \end{bmatrix} \Rightarrow \mathbf{f}(\mathbf{x}_1, t_1) = \begin{bmatrix} x_2 \\ 1 - x_1 - x_2 \end{bmatrix} = \begin{bmatrix} 0.095 \\ 0.9 \end{bmatrix} \\ \mathbf{x}_2 &= \mathbf{x}_1 + \frac{h}{2}(3 \mathbf{f}_1 - \mathbf{f}_0) = \begin{bmatrix} 0.005 \\ 0.095 \end{bmatrix} + 0.05 \left(3 \begin{bmatrix} 0.095 \\ 0.9 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) = \begin{bmatrix} 0.01925 \\ 0.18 \end{bmatrix}\end{aligned}$$

Otro problema de los métodos de Adams–Bashforth es que resultan estables sólo para valores muy pequeños del paso de integración. Esta característica se mejora utilizando los métodos de Adams–Bashforth–Moulton (ABM).

El método de ABM de tercer orden, por ejemplo, está dado por la siguiente fórmula

$$\begin{aligned}\mathbf{x}_{k+1}^P &= \mathbf{x}_k + \frac{h}{12}(23\mathbf{f}_k - 16\mathbf{f}_{k-1} + 5\mathbf{f}_{k-2}) \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \frac{h}{12}(5\mathbf{f}_{k+1}^P + 8\mathbf{f}_k - \mathbf{f}_{k-1})\end{aligned}$$

donde $\mathbf{f}_{k+1}^P = \mathbf{f}(x_{k+1}^P, t_{k+1})$.

Este método, denominado *predictor–corrector*, tiene mejor estabilidad que AB3 a costa de utilizar una evaluación extra de la función \mathbf{f} .

2.3.2. Métodos Multipaso Implícitos

Para obtener soluciones numéricas que preserven la estabilidad independientemente del paso de integración, al igual que en los métodos monopaso, hay que recurrir a los métodos implícitos.

Los métodos implícitos más utilizados en la práctica son los denominados *Backward Difference Formulae* (BDF), que son de tipo multipaso. Por ejemplo, el siguiente es el método de BDF de orden 3:

$$\mathbf{x}_{k+1} = \frac{18}{11}\mathbf{x}_k - \frac{9}{11}\mathbf{x}_{k-1} + \frac{2}{11}\mathbf{x}_{k-2} + \frac{6}{11} \cdot h \cdot \mathbf{f}_{k+1} \quad (2.25)$$

Este método tiene prácticamente el mismo costo computacional que Backward Euler, ya que la ecuación a resolver es muy similar. Sin embargo, BDF3 es de tercer orden.

Existen métodos de BDF de hasta orden 8, si bien los más utilizados son los de orden 1 (Backward Euler) hasta 5.

2.3.3. Control de Paso

En los métodos multipaso se puede también controlar el paso de integración de manera similar a la de los métodos monopaso. Sin embargo, las fórmulas de los métodos multipaso son sólo válidas asumiendo paso constante (ya que usan valores anteriores de la solución). Por lo tanto, para cambiar el paso en un método multipaso hay que interpolar los últimos valores de la solución a intervalos regulares correspondientes al nuevo paso de integración. En consecuencia, cambiar el paso tiene un costo adicional en este caso.

Por este motivo, los algoritmos de control de paso en métodos multipaso sólo cambian el paso cuando el cambio de paso es suficientemente grande.

2.4. Algunos Casos Problemáticos

Muchos sistemas dinámicos en la práctica tienen modelos en sistemas de ecuaciones diferenciales que resultan problemáticos para los distintos métodos de integración numérica.

Veremos a continuación distintos casos que nos obligarán a utilizar ciertas técnicas particulares.

2.4.1. Sistemas Stiff

Consideremos nuevamente el sistema masa resorte de la Ec.(2.1), pero ahora supongamos que el coeficiente de rozamiento es $b = 100$.

Con este nuevo valor para el parámetro, el sistema tiene matriz de evolución

$$A = \begin{bmatrix} 0 & 1 \\ -1 & -100 \end{bmatrix}$$

cuyos autovalores son $\lambda_1 \approx -0.01$ y $\lambda_2 \approx -100$.

La Figura 2.10 muestra la solución analítica (sólo se grafica x_2 en este caso). En la curva de la izquierda se ve la trayectoria completa, que se asemeja a una evolución de primer orden (correspondiente al autovalor lento $\lambda_1 \approx -0.01$). La curva de la derecha muestra en tanto el inicio de la trayectoria, que también se asemeja a una evolución de primer orden, pero mucho más rápida (correspondiente al autovalor $\lambda_2 \approx -100$).

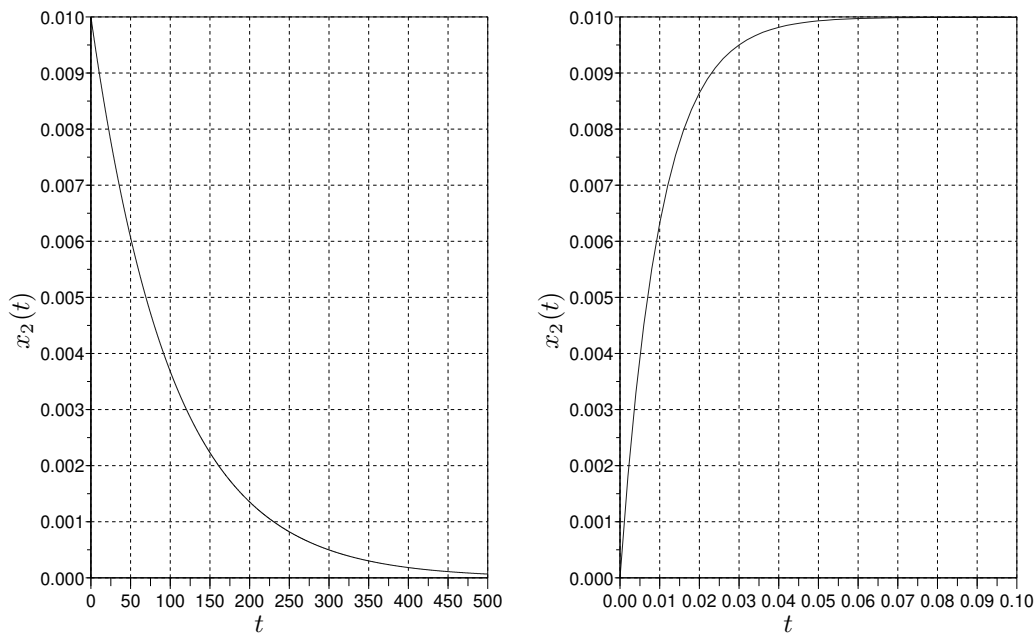


Figura 2.10: Solución Analítica para x_2 en el Sistema Masa Resorte con $b = 100$.

Evidentemente este sistema posee una dinámica lenta y una rápida. Los sistemas que tienen estas características se denominan *sistemas stiff* o rígidos. Como veremos, esto nos complicará a la hora de simularlo.

Es claro que si queremos simular este sistema usando algún método de integración, debemos hacerlo hasta un tiempo final de al menos $t_f = 500$. Además, si utilizamos el método de Euler, por ejemplo, deberemos usar un paso de integración $h < 0.02$ ya que de otra manera tendremos un resultado inestable según la Ec.(2.13). Esto implicará que haremos al menos $500/0.02 = 25000$ pasos para completar la simulación, lo que es absurdo dado lo simple del sistema.

La primer idea es entonces usar algún método de paso variable, como por ejemplo, el RK45 que vimos antes. El resultado en este caso no es mucho mejor, ya que el método necesita 13604 pasos para completar la simulación. La Figura 2.11 muestra la evolución del paso de integración en dicha simulación (se muestra sólo lo que ocurre hasta $t = 10$, luego sigue constante).

Aunque como puede verse en la solución analítica a la izquierda de la Fig.2.10 la dinámica rápida prácticamente desaparece a partir de $t = 0.1$, el método de RK45 nunca puede aumentar el paso de

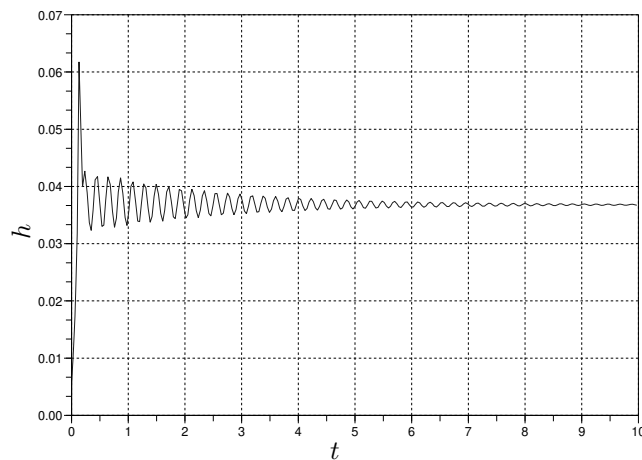


Figura 2.11: Paso de Integración h en la Simulación de un Sistema Stiff con RK45.

integración mucho más allá de $h \approx 0.04$. La explicación para esto es sencilla: debido al autovalor rápido $\lambda_2 \approx -100$ con pasos de integración mayores el método de RK4 se torna inestable por lo que el algoritmo obtiene un error grande que lo obliga a disminuir nuevamente el paso.

Este efecto sólo puede evitarse si usamos algoritmos que no se inestabilicen, esto es, si usamos alguno de los métodos implícitos que presentamos.

Utilizando por ejemplo, un método implícito de RK de cuarto orden con control de error de orden 5, con la misma tolerancia que en RK45 (0.001), la simulación se completa en sólo 33 pasos y se obtiene una precisión idéntica a la anterior. La Fig.2.12 muestra la evolución del paso de integración en este caso. El mismo está sólo limitado por la precisión y nunca por la estabilidad.

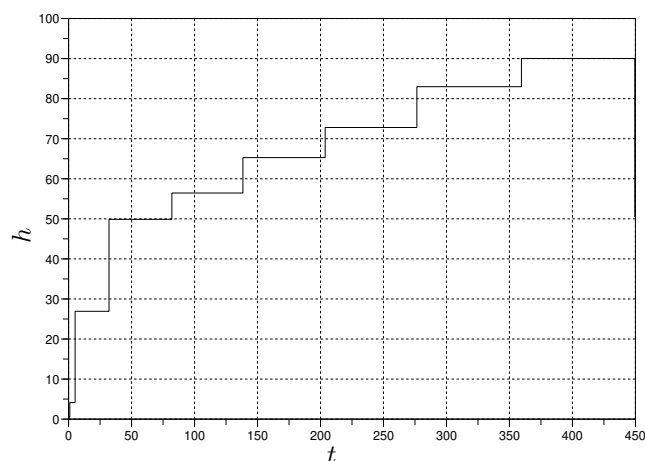


Figura 2.12: Paso de Integración h en la Simulación de un Sistema Stiff con un método implícito monopaso.

Si bien una computadora moderna puede realizar 13000 pasos en un sistema sencillo en un tiempo casi despreciable, en general los casos stiff son mucho más severos que lo visto aquí. En el simple ejemplo del masa resorte, si hubiéramos tenido $b = 10000$, RK45 necesitaría más de 13 millones de pasos para completar la simulación.

En la práctica, es casi imposible simular sistemas stiff sin recurrir a algoritmos implícitos.

2.4.2. Sistemas Marginalmente Estables

Muchos sistemas de la práctica tienen nulo o escaso amortiguamiento. En tal caso, las soluciones muestran oscilaciones sostenidas o bien que se amortiguan muy lentamente. Esta característica puede observarse en el ejemplo del sistema Masa Resorte haciendo $b = 0$ o tomando b muy pequeño.

Los sistemas con estas propiedades se denominan *marginalmente estables* y también revisten algunos problemas para la simulación numérica.

En el caso del sistema masa resorte con $b = 0$ tenemos un par de autovalores imaginarios puros conjugados $\lambda_{1,2} = \pm j$, por lo que las soluciones serán periódicas.

La Figura 2.13 muestra la solución analítica y el resultado de simular el sistema con Forward y Backward Euler usando $h = 0.1$. Como puede observarse en la misma, la solución numérica de Forward Euler es inestable, mientras que la de Backward Euler es asintóticamente estable.

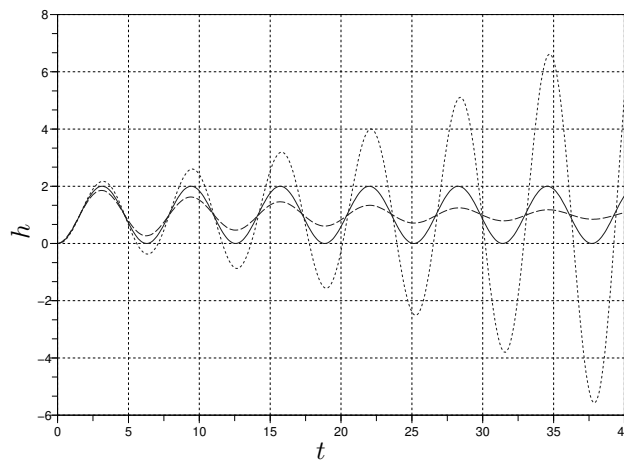


Figura 2.13: Sistema Masa Resorte con $b = 0$: Solución analítica (sólida), Forward Euler (punteada) y Backward Euler (línea discontinua).

Ninguna de las dos soluciones respeta la característica marginalmente estable original por lo que los resultados son cualitativamente incorrectos. Para obtener buenos resultados hay que utilizar métodos denominados *F-Estables* que preservan también la estabilidad marginal. Uno de estos métodos es la Regla Trapezoidal que presentamos en la Ec.(2.18).

2.4.3. Sistemas con Discontinuidades

La siguiente ecuación es un modelo muy simple de una pelotita que rebota contra el piso:

$$\begin{aligned} \dot{x}(t) &= v(t) \\ \dot{v}(t) &= -g - s_w(t) \cdot \frac{1}{m}(k \cdot x(t) + b \cdot v(t)) \end{aligned}$$

donde

$$s_w = \begin{cases} 0 & \text{si } x(t) > 0 \\ 1 & \text{en otro caso} \end{cases}$$

En este modelo, estamos considerando que cuando $x(t) > 0$ la pelotita está en el aire y responde a una ecuación de caída libre ($s_w = 0$). Cuando la pelotita entra en contacto con el piso, en cambio, sigue un modelo *masa-resorte-amortiguador*, lo que produce el rebote.

En este caso, consideramos parámetros $m = 1$, $b = 30$, $k = 1 \times 10^6$ y $g = 9.81$.

Decidimos simular este sistema utilizando el método de RK4, durante 5 segundos, a partir de la condición inicial $x(0) = 1$, $v(0) = 0$. Esta situación corresponde a soltar la pelotita desde 1 metro de distancia del piso. Comenzamos a tener resultados *decentes* con un paso de integración $h = 0.002$. En la Fig 2.14 se muestran los resultados de la simulación con pasos $h = 0.002$, $h = 0.001$ y $h = 0.0005$.

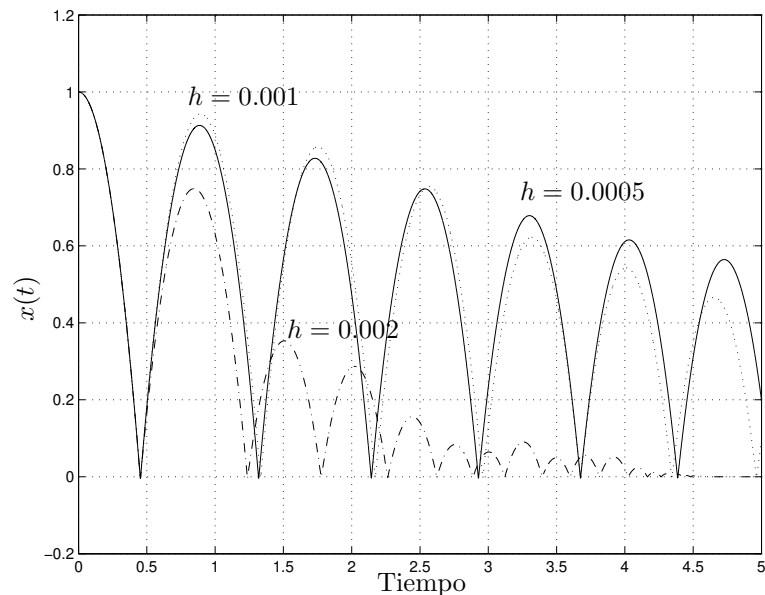


Figura 2.14: Simulación con RK4 de la pelotita rebotando.

Evidentemente, no podemos confiar en los resultados de esta simulación. La del paso más pequeño parecería más precisa, pero no hay manera de asegurarlo por el momento.

Mirando el comienzo de la simulación, no hay error apreciable hasta el primer pique. Evidentemente, el problema tiene que ver con la discontinuidad.

Veamos que ocurre en tanto si utilizamos un método de paso variable. En la Fig 2.15 se pueden apreciar los resultados con RK23 utilizando las tolerancias relativas 10^{-3} , 10^{-4} y 10^{-5} . De nuevo, los resultados son desalentadores. Si bien algunos piques se resuelven *bien* (al menos el método hace lo mismo con las tres tolerancias), en otros piques el error se torna inaceptable.

Tanto al usar paso fijo como al usar paso variable el problema tiene que ver con que se dieron pasos que atravesaron la discontinuidad. Es decir, en t_k teníamos $x(t_k) > 0$ y en t_{k+1} resultó $x(t_{k+1}) < 0$. Como la función \mathbf{f} resulta discontinua entre t_k y t_{k+1} , se produce un error muy grande.

La manera de corregir estos errores es evitando la situación anterior. Para esto, si nos encontramos con que $x(t_{k+1}) < 0$ hay que volver atrás y buscar el punto donde se produce la discontinuidad, esto es, el valor de \tilde{t} para el cual $x(\tilde{t}) = 0$. Una vez detectado este punto (para lo cual puede ser necesario iterar), se debe realizar un paso de valor $h = \tilde{t} - t_k$ (para llegar a dicho punto) y desde este nuevo punto reiniciar la simulación.

Los problemas de detección y tratamiento de discontinuidades son cruciales para simular correctamente sistemas de electrónica de potencia. En estos sistemas, generalmente, las discontinuidades ocurren a frecuencias muy grandes por lo que el uso de algoritmos eficientes para tratar con las discontinuidades es imprescindible.

2.5. Simulación de DAEs

Como vimos en el Capítulo 1, lo primero que obtenemos durante el proceso de modelado por primeros principios es un conjunto de relaciones constitutivas y estructurales que conforman un sistema de ecuaciones diferenciales algebraicas que en su expresión general tendrán la forma de la Ec.(1.10). Recordando que

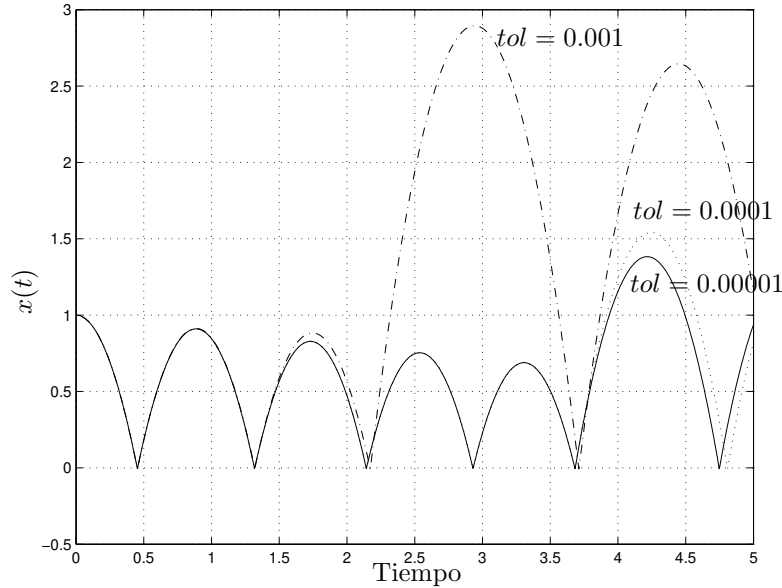


Figura 2.15: Simulación con RK23 de la pelotita rebotando.

para simular necesitamos conocer las trayectorias de entradas, esta ecuación tomará la siguiente forma:

$$\mathbf{f}(\mathbf{x}(t), \dot{\mathbf{x}}(t), \mathbf{a}(t), t) = 0. \quad (2.26)$$

Recordemos que $\mathbf{x} \triangleq [x_1, \dots, x_n]^T$ es el vector de estados y $\mathbf{a} \triangleq [a_1, \dots, a_r]^T$ es el vector de variables algebraicas.

Para poder simular un sistema como este utilizando alguno de los métodos numéricos que vimos, deberíamos implementar una iteración que a cada paso calcule el valor de $\dot{\mathbf{x}}(t)$, para lo que se deberá despejar de la Ec.(2.26) no sólo dicho vector sino también el vector de variables algebraicas $\mathbf{a}(t)$ (es un sistema de $n + r$ ecuaciones). Esto podría hacerse con la iteración de Newton.

Si además el método de integración a utilizar fuera implícito (como Backward Euler, por ejemplo) deberíamos agregar otra iteración de Newton por fuera para obtener \mathbf{x}_{k+1} . En definitiva, tendríamos un algoritmo de simulación del estilo

```

for  $k = 0 : N - 1$  do
  Inicializar  $\mathbf{x}_{k+1}^0$ 
  repeat
    Inicializar  $\dot{\mathbf{x}}_{k+1}^0$ 
    repeat
      Calcular  $\dot{\mathbf{x}}_{k+1}^{j+1}$ 
    until  $|\dot{\mathbf{x}}_{k+1}^{j+1} - \dot{\mathbf{x}}_{k+1}^j| < \text{tol}$ 
    Calcular  $\mathbf{x}_{k+1}^{l+1}$ 
  until  $|\mathbf{x}_{k+1}^{l+1} - \mathbf{x}_{k+1}^l| < \text{tol}$ 
end for
    
```

Esto es, tenemos tres iteraciones anidadas, una para avanzar el tiempo (for k), y dos iteraciones de Newton en el interior, cada una con sus respectivos cálculos de Jacobiano e inversión de la matriz resultante. Por lo tanto, el costo asociado a este algoritmo podría ser prohibitivo si el tamaño del problema fuera muy grande.

Si bien en el capítulo siguiente veremos que en muchos casos podremos convertir la DAE en una ODE simplificando notablemente el problema, hay otra solución que puede seguirse.

De la fórmula de Backward Euler de la Ec.(2.6), podemos reescribir

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h \mathbf{f}(\mathbf{x}_{k+1}, t_k) = \mathbf{x}_k + h \dot{\mathbf{x}}_{k+1}$$

y despejar

$$\dot{\mathbf{x}}_{k+1} = \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{h}$$

y finalmente reemplazar en la DAE (2.26):

$$\mathbf{f}(\mathbf{x}_k, \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{h}, \mathbf{a}_k, t_k) = 0. \quad (2.27)$$

En esta última ecuación la incógnita es directamente \mathbf{x}_{k+1} , que puede calcularse en cada paso con la iteración de Newton sin necesidad de tener dos iteraciones anidadas.

Esta misma idea se puede utilizar con otros métodos implícitos (particularmente con los métodos de BDF), y es una alternativa muy utilizada en la práctica.

2.6. Implementación de los Algoritmos en Software

Hay muchas herramientas de software que implementan los distintos métodos de integración numérica. Hay además versiones de los algoritmos ya implementadas y disponibles en distintos lenguajes de programación (sobre todo en lenguaje C), incluyendo métodos de todo tipo: explícitos, implícitos, con control de paso, para DAEs, etc. Entre los códigos abiertos disponibles más populares que implementan algoritmos de integración numérica de paso variable se encuentran DOPRI (métodos explícitos de Runge Kutta), DASSL (implícitos multipaso, para DAEs), y los CVODE, que incluyen métodos implícitos (BDF) y explícitos (AM) multipaso. Estos códigos se denominan *solvers* y son utilizados por diversas herramientas de modelado y simulación.

Pueden además implementarse fácilmente los distintos algoritmos que vimos en este capítulo, sobre todo los explícitos, ya que los métodos implícitos requieren implementar además la iteración de Newton que implica una dificultad extra. Por ejemplo, utilizando el lenguaje de Matlab u Octave, el Código 2.1 implementa el método de Forward Euler.

Código 2.1: Función de Matlab/Octave que implementa método de Forward Euler

```
function [t,x]=feuler(f,x0,h,t0,tf)
    t=[t0:h:tf];
    x=zeros(length(x0),length(t));
    x(:,1)=x0;
    for k=1:length(t)-1
        x(:,k+1)=x(:,k)+h*f(x(:,k),t(k));
    end
end
```

Notar que este algoritmo usa como entradas la función \mathbf{f} que calcula la derivada del estado, la condición inicial \mathbf{x}_0 , el paso de integración h , el tiempo inicial t_0 y el tiempo final de simulación t_f . Como resultado, brinda un vector fila t con los instantes de tiempo donde se realizaron los pasos y una matriz x que contiene en cada columna el valor de la solución en el instante t_k .

Si queremos simular utilizando este algoritmo un modelo Masa-Resorte como el de Ec.(2.1), deberemos construir la función $\mathbf{f}(\mathbf{x}, t)$ que calcula la derivada del estado $\dot{\mathbf{x}}$ según dicho modelo como se muestra en el Código 2.2.

Código 2.2: Función de Matlab/Octave con el modelo de la Ec.(2.1)

```
function dx=masares(x,t)
    m=1; b=1; k=1; %parametros
    F=1;           %entrada
    der_x1=x(2);
    der_x2=-k/m*x(1)-b/m*x(2)+F;
    dx=[der_x1; der_x2]; %vector de derivadas
end
```

Luego, podemos realizar la simulación y graficar los resultados invocando el Código 2.3.

Código 2.3: Comandos de Matlab/Octave para simular el modelo de la Ec.(2.1)

```
[t,x]=feuler(@masares,[0;0],0.1,0,10);
plot(t,x)
```

Matlab y Octave tienen también herramientas que resuelven ecuaciones algebraicas basadas en la iteración de Newton. Particularmente, el comando `fsolve` permite resolver un problema de tipo $\mathbf{F}(\mathbf{z}) = 0$. Utilizando este comando entonces, un método implícito como el de Backward Euler se puede implementar de manera trivial usando el Código 2.4:

Código 2.4: Función de Matlab/Octave que implementa método de Backward Euler

```
function [t,x]=beuler(f,x0,h,t0,tf)
t=[t0:h:tf];
x=zeros(length(x0),length(t));
x(:,1)=x0;
for k=1:length(t)-1
    F=@(z)(z-x(:,k)-h*f(z,t(k))); %funcion que debe ser 0
    x(:,k+1)=fsolve(F,x(:,k));
end
end
```

Además, tanto Matlab como Octave tienen implementados distintos métodos numéricos de paso variable, incluyendo `ode23` y `ode45` (métodos explícitos de Runge-Kutta) y `ode15s` que es un método multipaso implícito similar a los algoritmos de BDF. En Octave también hay una implementación de DASSL que permite simular directamente DAEs.

Las herramientas de modelado y simulación basadas en Modelica, como Dymola, Wolfram System-Modeler y OpenModelica, cuentan con una amplia variedad de solvers y tienen además la ventaja de que realizan un preproceso sobre las ecuaciones (veremos parte de ellas en el Capítulo 3) de manera que, particularmente en las DAEs, los modelos se pueden escribir de manera más sencillas y las simulaciones resultan más eficientes.

2.7. Bibliografía Complementaria

Los contenidos de este apunte están principalmente basados en las notas de clases de la materia de doctorado *Simulación de Sistemas Continuos* [Kof06], que a su vez están basadas principalmente en el libro [CK06] (disponible en la biblioteca de la Facultad y de la Escuela de Ingeniería Electrónica).

Por otro lado, los libros [HNW00] y [HW91] son referencia obligada en la disciplina para quienes tengan interés en las propiedades más matemáticas de los algoritmos.

Finalmente, una colección muy importante de algoritmos y programas para implementar diversos métodos numéricos (no sólo para ecuaciones diferenciales, sino para problemas más generales) puede encontrarse en el libro clásico [PTVF07].

2.8. Problemas Propuestos

[P2.1] Forward y Backward Euler

Dada la Ecuación Diferencial Ordinaria

$$\dot{x}(t) = -3x(t) + 3 \quad (\text{P2.1a})$$

con condición inicial $x(0) = 0$, se pide,

1. Obtener y graficar la solución analítica.
2. Resolverla mediante el método de Forward Euler utilizando pasos $h = 0.1$ y $h = 1$. (Calcular los 5 primeros valores en cada caso).
3. Resolverla ahora mediante Backward Euler utilizando los mismos pasos que en el punto anterior.

4. Comparar el error cometido tras el primer paso de integración utilizando los dos valores de h en cada método. Explicar los resultados.
5. Analizar la estabilidad numérica de las soluciones obtenidas con ambos valores de h y con ambos métodos.

[P2.2] Método de Heun

Utilizando el Método de Heun, definido en las Ecs.(2.15)–(2.16), se pide

1. Resolver la Ec.(P2.1a) a partir de la condición inicial $x(0) = 0$ utilizando pasos $h = 0.1$ y $h = 1$. (Calcular los 5 primeros valores en cada caso).
2. Comparar el error cometido tras el primer paso de integración utilizando los dos valores de h . Explicar las diferencias o coincidencias con los métodos de Euler del Problema P2.1.
3. Analizar la estabilidad numérica con ambos valores de h y comparar con lo que se obtiene usando Forward y Backward Euler.
4. Programar en Octave o Matlab una función que implemente este método, similar a la del Código 2.1 y verificar los resultados obtenidos en los puntos 1 y 3.

[P2.3] Regla del Punto Medio

La *Regla del Punto Medio* es un método de integración de segundo orden, que tiene versiones implícitas y explícitas. Una versión de este método aplicada al sistema

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t) \quad (\text{P2.3a})$$

tiene la siguiente definición:

$$\begin{aligned} \mathbf{k}_1 &= \mathbf{f}(\mathbf{x}_k, t_k) \\ \mathbf{k}_2 &= \mathbf{f}\left(\mathbf{x}_k + \frac{h}{2} \mathbf{k}_1, t_k + \frac{h}{2}\right) \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + h \mathbf{k}_2 \end{aligned} \quad (\text{P2.3b})$$

Se pide entonces

1. Clasificar este método según sea
 - a) Implícito/Explícito
 - b) Monopaso/Multipaso
2. Utilizando este método con un paso $h = 0.1$, obtener la solución hasta $t_f = 0.5$ del sistema de la Ec.(P2.1a) a partir de la condición inicial $x(0) = 0$.
3. Determinar para que valores del paso de integración h la solución numérica del ejemplo de la Ec.(P2.1a) resulta estable.
4. Programar en Octave o Matlab una función que implemente este método, similar a la del Código 2.1 y verificar los resultados obtenidos en los puntos anteriores.

[P2.4] Regla Trapezoidal

Repetir el Problema P2.3 utilizando la Regla Trapezoidal, dada por la Ec.(2.18), en lugar de la Regla del Punto Medio.

[P2.5] Método de Adams-Bashforth 3

Dado el sistema de la Ec.(P2.3a), el método de AB3 lo aproxima con la fórmula

$$\mathbf{x}_{k+1} = x_k + \frac{h}{12} (23 \mathbf{f}(\mathbf{x}_k, t_k) - 16 \mathbf{f}(\mathbf{x}_{k-1}, t_{k-1}) + 5 \mathbf{f}(\mathbf{x}_{k-2}, t_{k-2})) \quad (\text{P2.5a})$$

Se pide entonces

1. Clasificar este método según sea

- a) Implícito/Explícito
- b) Monopaso/Multipaso

2. Dado el sistema

$$\dot{x}(t) = -2x(t) + 4 \quad (\text{P2.5b})$$

con la condición inicial $x(0) = 0$, se pide

- a) Obtener la solución numérica con el método de AB3 hasta $t_f = 0.6$ con un paso $h = 0.2$. Si necesita condiciones iniciales adicionales, utilice la solución analítica:

$$x(t) = 2 \cdot (1 - e^{-2 \cdot t})$$

- b) Comparar la solución numérica y la analítica.
- c) Obtener la solución numérica con el método de AB3 hasta $t_f = 0.06$ con un paso $h = 0.02$. Si necesita condiciones iniciales adicionales, utilice nuevamente la solución analítica.
- d) Comparar los errores cometidos tras el primer paso con los dos pasos de integración y concluya sobre el orden del método.

Capítulo 3

Causalización de Sistemas de DAEs

En el Capítulo 1 vimos que las Ecuaciones Diferenciales Algebraicas son la forma más simple en la que se puede obtener un modelo matemático. Sin embargo, este tipo de representación no es la más adecuada para simular porque en cada paso de integración debemos resolver un sistema de ecuaciones algebraicas que puede tener gran tamaño. Tampoco es la más adecuada para realizar análisis de propiedades, porque para esto necesitamos que el sistema esté en su forma de Ecuaciones de Estado o en forma de Función Transferencia si fuera lineal y quisiéramos analizar la respuesta en frecuencia.

En este capítulo, que está basado principalmente en el Capítulo 7 del libro *Continuous System Simulation* [CK06], vamos a ver versiones simplificadas de los procedimientos que permiten en ciertos casos transformar sistemas de Ecuaciones Diferenciales Algebraicas en sistemas de Ecuaciones Diferenciales Ordinarias y en otros casos permiten al menos reducir el número de variables que intervienen en las iteraciones para calcular la derivada del vector de estados $\dot{\mathbf{x}}(t)$.

Estos procedimientos constituyen una parte esencial de los *compiladores* de los lenguajes de modelado como Modelica.

3.1. Ordenamiento y Causalización de Ecuaciones

3.1.1. Modelos Implícitos y Explícitos

Consideremos nuevamente el circuito RLC del Capítulo 1, con el sistema de DAEs que definen sus relaciones constitutivas y estructurales, donde $v(t)$ representa una función del tiempo conocida.

$$u_R(t) - R i_R(t) = 0 \quad (3.1a)$$

$$\dot{q}(t) - i_C(t) = 0 \quad (3.1b)$$

$$q(t) - C u_C(t) = 0 \quad (3.1c)$$

$$\dot{\phi}(t) - u_L(t) = 0 \quad (3.1d)$$

$$\phi(t) - L i_L(t) = 0 \quad (3.1e)$$

$$u_S(t) - v(t) = 0 \quad (3.1f)$$

$$u_L(t) + u_R(t) + u_C(t) - u_S(t) = 0 \quad (3.1g)$$

$$i_L(t) - i_R(t) = 0 \quad (3.1h)$$

$$i_C(t) - i_L(t) = 0 \quad (3.1i)$$

$$i_S(t) - i_R(t) = 0 \quad (3.1j)$$

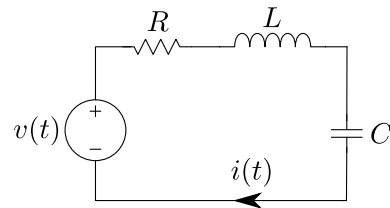


Figura 3.1: Circuito RLC Serie

Si quisiéramos simular este sistema utilizando el método de Forward Euler sin realizar ningún tipo de transformación sobre el sistema de DAEs, tendríamos que implementar un algoritmo iterativo que sea capaz de *despejar* las derivadas de los estados (y las variables algebraicas) de la Ec.(3.1), lo que constituye

un sistema de 10 ecuaciones con 10 incógnitas. Esto además debería realizarse cada vez que queremos obtener el valor de dichas derivadas (una vez en cada paso si usamos Forward Euler).

El Código 3.1 contiene dos funciones de Matlab/Octave que brindan una manera de realizar esto:

Código 3.1: Funciones de Matlab/Octave con el modelo de la Ec.(3.1)

```
function dx=circuit_ode(x,t)
    F = @(z) circuit_dae(x,z(1:2),z(3:10),t);
    z0=zeros(10,1);
    z=fsolve(F,z0);
    dx=z(1:2);
end

function res=circuit_dae(x,dx,a,t)
    R=1; C=1; L =1; %parametros
    q=x(1); phi=x(2); %estados
    der_q=dx(1); der_phi=dx(2); %derivadas de estados
    uR=a(1); iR=a(2); iC=a(3); uC=a(4); uL=a(5); iL=a(6); uS=a(7); iS=a(8);
    res=zeros(10,1); %residuo (debera ser cero)
    res(1) = uR - R * iR;
    res(2) = der_q - iC;
    res(3) = q - C * uC;
    res(4) = der_phi - uL;
    res(5) = phi - L * iL;
    res(6) = uS - sin(t);
    res(7) = uL + uR + uC - uS;
    res(8) = iL - iR;
    res(9) = iC - iL;
    res(10) = iS - iR;
end
```

La función `circuit_dae` calcula el lado izquierdo de las ecuaciones diferenciales algebraicas, es decir, las expresiones que deben ser nulas. Estas expresiones dan como resultado un residuo `res`. Esta función es utilizada por la función `circuit_ode` que calcula la derivada del estado de manera que efectivamente dicho residuo sea cero. Esto último lo hace utilizando la función `fzero` que calcula las raíces de una ecuación algebraica.

Podemos entonces simular esta DAE directamente utilizando el método de Forward Euler del Código 2.1 invocando

Código 3.2: Comando de Matlab/Octave para simular la Ec.(3.1)

```
[t,x]=feuler(@circuit_ode,[0;0],0.1,0,10);
plot (t,x)
```

Si bien esto funciona, es extremadamente costoso desde el punto de vista computacional. Nuestro objetivo entonces será transformar este sistema de DAEs en algo equivalente al sistema de Ecuaciones de Estado dado por

$$\begin{aligned} \dot{q}(t) &= \frac{1}{L} \phi(t) \\ \dot{\phi}(t) &= -\frac{1}{C} q(t) - \frac{R}{L} \phi(t) + v(t), \end{aligned} \quad (3.2)$$

es decir, buscaremos tener expresiones explícitas para las derivadas de los estados en función de los propios estados y de las entradas.

Una opción para obtener expresiones explícitas está dada por el siguiente sistema de ecuaciones

$$u_C(t) := \frac{q(t)}{C} \quad (3.3a)$$

$$i_L(t) := \frac{\phi(t)}{L} \quad (3.3b)$$

$$u_S(t) := v(t) \quad (3.3c)$$

$$i_R(t) := i_L(t) \quad (3.3d)$$

$$i_C(t) := i_L(t) \quad (3.3e)$$

$$i_S(t) := i_R(t) \quad (3.3f)$$

$$u_R(t) := R i_R(t) \quad (3.3g)$$

$$u_L(t) := u_S(t) - u_R(t) - u_C(t) \quad (3.3h)$$

$$\dot{q}(t) := i_C(t) \quad (3.3i)$$

$$\dot{\phi}(t) := u_L(t) \quad (3.3j)$$

donde el símbolo $:=$ indica una *asignación* (no una igualdad). Puede verse fácilmente que la secuencia de asignaciones dada por las Ecs.(3.3a)-(3.3j) tiene las siguientes propiedades:

- En el lado izquierdo de cada asignación se calcula una variable que no fue calculada previamente.
- En el lado derecho de cada asignación se utilizan estados o bien variables algebraicas que ya fueron calculadas previamente.

Esto implica que la secuencia de asignaciones se puede programar en cualquier lenguaje de programación de manera muy simple y conduce a un código que permite evaluar efectivamente las derivadas de los estados en función de los estados y las entradas como en la Ec.(3.2). De hecho, si utilizamos Matlab u Octave, podemos traducir la secuencia de asignaciones en la siguiente función:

Código 3.3: Función de Matlab/Octave con el modelo de la Ec.(3.3)

```
function dx=circuit(x,t)
  q=x(1); phi=x(2); %estados
  R=1; L=1; C=1; %parametros
  v=sin(t); %entrada
  uC=q/c;
  iL=phi/L;
  uS=v;
  iR=iL;
  iC=iL;
  iS=iR;
  uR=R*iR;
  uL=uS-uR-uC;
  der_q=iC;
  der_phi=uL;
  dx=[der_q; der_phi]; %vector de derivadas
end
```

Esta función puede utilizarse junto con el método de Forward Euler del Código 2.1 para simular el sistema, y puede verse que es equivalente a la función que obtendríamos si implementamos directamente el código a partir del modelo en Ecuaciones de Estado dado por la Ec.(3.2):

Código 3.4: Función de Matlab/Octave con el modelo de la Ec.(3.2)

```
function dx=circuit2(x,t)
  q=x(1); phi=x(2); %estados
  R=1; L=1; C=1; %parametros
  v=sin(t); %entrada
  der_q=phi/L;
```



```

der_phi=-q/C-R/L*phi+v;
dx=[der_q; der_phi]; %vector de derivadas
end

```

Estos modelos explícitos son muchísimo más eficientes que el del Código 3.1 ya que no involucran iteraciones para resolver los sistemas de ecuaciones.

El pasaje de la Ec.(3.1) a la Ec.(3.3) implica ordenar las ecuaciones horizontalmente (para decidir qué variable se despeja de cada ecuación) y verticalmente (para decidir en qué orden se realizan las asignaciones). Veremos entonces un procedimiento que, en muchos casos, permite resolver este problema de ordenamiento causalizando un sistema de DAEs y convirtiéndolo en una secuencia de asignaciones que equivale a un sistema de Ecuaciones de Estado.

3.1.2. Algoritmo de Ordenamiento Horizontal y Vertical

En un caso general, tendremos un sistema de ecuaciones diferenciales algebraicas de la forma:

$$\begin{aligned}
 f_1(x_1(t), \dots, x_n(t), \dot{x}_1(t), \dots, \dot{x}_n(t), a_1(t), \dots, a_r(t), v_1(t), \dots, v_m(t), t) &= 0 \\
 f_2(x_1(t), \dots, x_n(t), \dot{x}_1(t), \dots, \dot{x}_n(t), a_1(t), \dots, a_r(t), v_1(t), \dots, v_m(t), t) &= 0 \\
 &\vdots \\
 f_{n+r}(x_1(t), \dots, x_n(t), \dot{x}_1(t), \dots, \dot{x}_n(t), a_1(t), \dots, a_r(t), v_1(t), \dots, v_m(t), t) &= 0
 \end{aligned} \tag{3.4}$$

donde las variables x_i serán (en principio) los estados, las a_i son variables algebraicas y las entradas v_i se asumen conocidas.

El objetivo será entonces ordenar tanto vertical como horizontalmente estas ecuaciones de manera de poder calcular las derivadas de los estados \dot{x}_i . Para esto, tomaremos en cuenta lo siguiente:

- Tendremos un sistema de $n + r$ ecuaciones y $n + r$ incógnitas.
- Las incógnitas son las n derivadas de los estados y las r variables algebraicas desconocidas.

El algoritmo que utilizaremos se basará en las siguientes reglas:

Regla 1 Si una ecuación (E_i) contiene una *única* incógnita u_j (que puede ser una derivada o una variable algebraica), entonces:

- u_j debe despejarse de (E_i) (ordenamiento *horizontal*). Esto debe ser así porque, de lo contrario, no podríamos utilizar dicha ecuación para otra cosa.
- La ecuación (E_i) despejada debe colocarse al principio (ordenamiento *vertical*). Esto puede hacerse así, ya que esta ecuación no necesita que ninguna otra incógnita se calcule previamente.

Regla 2 Si una incógnita u_j aparece en una *única* ecuación (E_i), entonces:

- u_j debe despejarse de (E_i) (ordenamiento *horizontal*). Esto debe ser así porque no hay otra ecuación para calcular u_j .
- La ecuación (E_i) despejada debe colocarse al final (ordenamiento *vertical*). Esto puede hacerse así ya que la incógnita u_j no será necesaria en ninguna otra ecuación.

Cada vez que apliquemos una de estas reglas, la incógnita que despejamos dejará de ser considerada incógnita y la ecuación utilizada se removerá del sistema. De esta manera, si todo funciona bien (luego veremos que esto no será siempre así), tras $n + r$ iteraciones del algoritmo tendríamos nuestro sistema ordenado tanto vertical como horizontalmente.

Si aplicamos entonces este procedimiento sobre el modelo del circuito dado por el sistema de DAEs de la Ec.(3.1), en el primer paso encontramos que podemos aplicar la Regla 1 a la tercer ecuación, ya que esta contiene una única incógnita ($u_C(t)$) que deberemos despejar de la misma y colocar arriba de todo. Colocamos entonces a la derecha las ecuaciones causalizadas.

$$\begin{aligned}
 \mathbf{u}_R(t) - R \mathbf{i}_R(t) &= 0 & u_C(t) &:= \frac{q(t)}{C} \\
 \dot{q}(t) - i_C(t) &= 0 \\
 \boxed{q(t) - C u_C(t) = 0} \\
 \dot{\phi}(t) - u_L(t) &= 0 \\
 \phi(t) - L i_L(t) &= 0 \\
 \mathbf{u}_S(t) - v(t) &= 0 \\
 \mathbf{u}_L(t) + \mathbf{u}_R(t) + u_C(t) - \mathbf{u}_S(t) &= 0 \\
 i_L(t) - i_R(t) &= 0 \\
 i_C(t) - i_L(t) &= 0 \\
 i_S(t) - i_R(t) &= 0
 \end{aligned}$$

A partir de este punto, u_C deja de ser una incógnita (y no la representamos más en negritas) removiendo además la ecuación correspondiente del sistema de DAEs.

En el siguiente paso, vemos que la incógnita $\dot{\phi}(t)$ sólo aparece en la cuarta ecuación y podemos por lo tanto aplicar la Regla 2, tras lo cual colocamos la ecuación despejada al final.

$$\begin{aligned}
 \mathbf{u}_R(t) - R \mathbf{i}_R(t) &= 0 & u_C(t) &:= \frac{q(t)}{C} \\
 \dot{q}(t) - i_C(t) &= 0 \\
 q(t) - C u_C(t) &= 0 \\
 \boxed{\dot{\phi}(t) - u_L(t) = 0} \\
 \phi(t) - L i_L(t) &= 0 \\
 \mathbf{u}_S(t) - v(t) &= 0 \\
 \mathbf{u}_L(t) + \mathbf{u}_R(t) + u_C(t) - \mathbf{u}_S(t) &= 0 \\
 i_L(t) - i_R(t) &= 0 \\
 i_C(t) - i_L(t) &= 0 \\
 i_S(t) - i_R(t) &= 0 & \dot{\phi}(t) &:= u_L(t)
 \end{aligned}$$

Tras eliminar la ecuación anterior, al avanzar a la ecuación siguiente vemos que la misma contiene una única incógnita y podemos entonces aplicar la Regla 1.

$$\begin{aligned}
 \mathbf{u}_R(t) - R \mathbf{i}_R(t) &= 0 & u_C(t) &:= \frac{q(t)}{C} \\
 \dot{\mathbf{q}}(t) - \mathbf{i}_C(t) &= 0 & i_L(t) &:= \frac{\phi(t)}{L} \\
 q(t) - C u_C(t) &= 0 \\
 \dot{\phi}(t) - u_L(t) &= 0 \\
 \boxed{\phi(t) - L i_L(t) = 0} \\
 \mathbf{u}_S(t) - v(t) &= 0 \\
 \mathbf{u}_L(t) + \mathbf{u}_R(t) + u_C(t) - \mathbf{u}_S(t) &= 0 \\
 \mathbf{i}_L(t) - \mathbf{i}_R(t) &= 0 \\
 \mathbf{i}_C(t) - \mathbf{i}_L(t) &= 0 \\
 \mathbf{i}_S(t) - \mathbf{i}_R(t) &= 0 & \dot{\phi}(t) &:= u_L(t)
 \end{aligned}$$

La siguiente ecuación también contiene una única incógnita (u_S), por lo que aplicamos nuevamente la Regla 1.

$$\begin{aligned}
 \mathbf{u}_R(t) - R \mathbf{i}_R(t) &= 0 & u_C(t) &:= \frac{q(t)}{C} \\
 \dot{\mathbf{q}}(t) - \mathbf{i}_C(t) &= 0 & i_L(t) &:= \frac{\phi(t)}{L} \\
 q(t) - C u_C(t) &= 0 & u_S(t) &:= v(t) \\
 \dot{\phi}(t) - u_L(t) &= 0 \\
 \phi(t) - L i_L(t) &= 0 \\
 \boxed{\mathbf{u}_S(t) - v(t) = 0} \\
 \mathbf{u}_L(t) + \mathbf{u}_R(t) + u_C(t) - \mathbf{u}_S(t) &= 0 \\
 i_L(t) - \mathbf{i}_R(t) &= 0 \\
 \mathbf{i}_C(t) - i_L(t) &= 0 \\
 \mathbf{i}_S(t) - \mathbf{i}_R(t) &= 0 & \dot{\phi}(t) &:= u_L(t)
 \end{aligned}$$

Luego de este paso, podemos ver por ejemplo que \dot{q} aparece sólo en la segunda ecuación y aplicar la Regla 2.

$$\begin{aligned}
 \mathbf{u}_R(t) - R \mathbf{i}_R(t) &= 0 & u_C(t) &:= \frac{q(t)}{C} \\
 \boxed{\dot{\mathbf{q}}(t) - \mathbf{i}_C(t) = 0} & & i_L(t) &:= \frac{\phi(t)}{L} \\
 q(t) - C u_C(t) &= 0 & u_S(t) &:= v(t) \\
 \dot{\phi}(t) - u_L(t) &= 0 \\
 \phi(t) - L i_L(t) &= 0 \\
 \mathbf{u}_S(t) - v(t) &= 0 \\
 \mathbf{u}_L(t) + \mathbf{u}_R(t) + u_C(t) - \mathbf{u}_S(t) &= 0 \\
 i_L(t) - \mathbf{i}_R(t) &= 0 \\
 \mathbf{i}_C(t) - i_L(t) &= 0 & \dot{q}(t) &:= i_C(t) \\
 \mathbf{i}_S(t) - \mathbf{i}_R(t) &= 0 & \dot{\phi}(t) &:= u_L(t)
 \end{aligned}$$

Continuando con este procedimiento, tras cinco pasos más, llegamos finalmente al sistema de la Ec.(3.3) que contiene todas las ecuaciones ordenadas horizontal y verticalmente y que, como vimos, resulta equivalente a un sistema de Ecuaciones de Estado.

Las ecuaciones ordenadas de la Ec.(3.3) pueden utilizarse tanto para simular un sistema utilizando una función como la del Código 3.3 como para obtener las Ecuaciones de Estado de manera sistemática. Esto es útil para luego realizar análisis sobre el modelo, utilizando las EE o la Función Transferencia (asumiendo que el sistema es lineal) que se puede obtener directamente de las EE.

Dado que en cada paso del algoritmo ordenamos una ecuación, en un caso genérico como el de la Ec.(3.4) el procedimiento finalizará tras $n+r$ pasos. Tanto para acelerar el proceso de causalización como para generar un código más corto y eficiente para evaluar las derivadas del estado, se suelen eliminar las ecuaciones *triviales* del modelo, eliminando también las correspondientes variables algebraicas.

3.1.3. Eliminación de Ecuaciones Triviales

Las ecuaciones triviales son las que tienen la forma $a_i - a_j = 0$ (o sea, $a_i = a_j$) y en tal caso se dice que las variables algebraicas a_i y a_j son *alias* y se puede eliminar una de ellas. En la Ec.(3.1) podríamos eliminar las últimas tres ecuaciones que son triviales reemplazando las corrientes por una única corriente (i_L por ejemplo), lo que reduciría de 8 a 5 el número de variables algebraicas.

$$\begin{aligned} u_R(t) - R i_L(t) &= 0 \\ \dot{q}(t) - \dot{i}_L(t) &= 0 \\ q(t) - C u_C(t) &= 0 \\ \dot{\phi}(t) - u_L(t) &= 0 \\ \phi(t) - L i_L(t) &= 0 \\ u_S(t) - v(t) &= 0 \\ u_L(t) + u_R(t) + u_C(t) - u_S(t) &= 0 \end{aligned}$$

En realidad, es posible también eliminar i_L y u_L ya que ambas son iguales a dos derivadas de los estados (podemos tratar las derivadas de los estados como variables algebraicas, pero que no se pueden eliminar). En tal caso, el sistema se reduce a

$$u_R(t) - R \dot{q}(t) = 0 \quad (3.6a)$$

$$q(t) - C u_C(t) = 0 \quad (3.6b)$$

$$\phi(t) - L \dot{q}(t) = 0 \quad (3.6c)$$

$$u_S(t) - v(t) = 0 \quad (3.6d)$$

$$\dot{\phi}(t) + u_R(t) + u_C(t) - u_S(t) = 0 \quad (3.6e)$$

que sólo contiene tres variables algebraicas.

3.1.4. Obtención de las Ecuaciones de Estado

Una vez ordenado horizontal y verticalmente un sistema de DAEs, la obtención de las ecuaciones de estado es directa. Se puede directamente comenzar desde la primera ecuación del sistema ordenado e ir reemplazando las variables algebraicas por sus expresiones en función de los estados.

Por ejemplo, si usamos el algoritmo de causalización en el sistema reducido (sin ecuaciones triviales) de la Ec.(3.6), obtenemos el siguiente sistema ordenado:

$$u_C(t) := \frac{q(t)}{C} \quad (3.7a)$$

$$\dot{q}(t) := \frac{\phi(t)}{L} \quad (3.7b)$$

$$u_S(t) := v(t) \quad (3.7c)$$

$$u_R(t) := R \dot{q}(t) \quad (3.7d)$$

$$\dot{\phi}(t) := u_S(t) - u_R(t) - u_C(t) \quad (3.7e)$$

y podemos proceder como sigue:

$$u_C(t) := \frac{q(t)}{C} \quad (\text{está bien porque } q(t) \text{ es estado})$$

$$\dot{q}(t) := \frac{\phi(t)}{L} \quad (\text{está bien porque } q(t) \text{ es estado})$$

$$u_S(t) := v(t) \quad (\text{está bien porque } v(t) \text{ es entrada})$$

$$u_R(t) := R \dot{q}(t) = R \frac{\phi(t)}{L} \quad (\text{usamos la 2da ecuación para reemplazar } \dot{q}(t))$$

$$\dot{\phi}(t) := u_S(t) - u_R(t) - u_C(t) = v(t) - R \frac{\phi(t)}{L} - \frac{q(t)}{C}$$

donde en el último paso usamos la primera, la tercera y la cuarta ecuación para reemplazar las variables algebraicas en función de los estados. Luego, las EE son:

$$\dot{q}(t) = \frac{\phi(t)}{L}$$

$$\dot{\phi}(t) = v(t) - R \frac{\phi(t)}{L} - \frac{q(t)}{C}$$

3.2. Lazos Algebraicos

3.2.1. Ejemplo Introductorio

En el ejemplo de la Ec.(3.1) el algoritmo de causalización funcionó muy bien y permitió obtener un modelo explícito. La pregunta es ¿funcionará siempre?.

Veamos ahora en la Figura 3.2 otro circuito, en principio más simple que el anterior, junto al sistema de DAEs que definen sus relaciones constitutivas y estructurales.

$$u_{R_1}(t) - R_1 i_{R_1}(t) = 0 \quad (3.8a)$$

$$u_{R_2}(t) - R_2 i_{R_2}(t) = 0 \quad (3.8b)$$

$$\dot{\phi}(t) - u_L(t) = 0 \quad (3.8c)$$

$$\phi(t) - L i_L(t) = 0 \quad (3.8d)$$

$$i_L(t) + i_{R_1}(t) + i_{R_2}(t) = 0 \quad (3.8e)$$

$$u_{R_1}(t) - u_{R_2}(t) = 0 \quad (3.8f)$$

$$u_L(t) - u_{R_2}(t) = 0 \quad (3.8g)$$

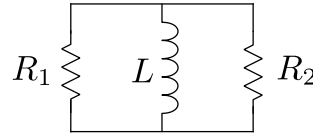


Figura 3.2: Circuito RL Paralelo

Este sistema de ecuaciones diferenciales algebraicas tiene, en principio, una variable de estado (ϕ) y 6 variables algebraicas (tensiones y corrientes en cada elemento) lo que constituye un sistema de 7 ecuaciones con 7 incógnitas (recordando que una de las incógnitas es la derivada del estado $\dot{\phi}$).

Veamos entonces qué sucede al utilizar el procedimiento de ordenamiento que vimos anteriormente. Para esto, comenzaremos eliminando las ecuaciones triviales (las últimas dos) y las variables algebraicas correspondientes (u_{R_1} y u_{R_2}). Luego, observando que la derivada $\dot{\phi}$ aparece en una única ecuación, podemos aplicar la Regla 2:

$$\begin{aligned}
 u_L(t) - R_1 i_{R_1}(t) &= 0 \\
 u_L(t) - R_2 i_{R_2}(t) &= 0 \\
 \boxed{\dot{\phi}(t) - u_L(t) = 0} \\
 \phi(t) - L i_L(t) &= 0 \\
 i_L(t) + i_{R_1}(t) + i_{R_2}(t) &= 0
 \end{aligned}
 \qquad \dot{\phi}(t) := u_L(t)$$

La ecuación siguiente tiene sólo una incógnita (i_L), por lo que podemos aplicar ahora la Regla 1.

$$\begin{aligned}
 u_L(t) - R_1 i_{R_1}(t) &= 0 \\
 u_L(t) - R_2 i_{R_2}(t) &= 0 \\
 \dot{\phi}(t) - u_L(t) &= 0 \\
 \boxed{\phi(t) - L i_L(t) = 0} \\
 i_L(t) + i_{R_1}(t) + i_{R_2}(t) &= 0
 \end{aligned}
 \qquad i_L(t) := \frac{\phi(t)}{L}$$

$$\qquad \dot{\phi}(t) := u_L(t)$$

Tras este segundo paso, no podemos aplicar ninguna de las reglas ya que todas las ecuaciones tienen dos incógnitas y todas las incógnitas aparecen en al menos dos ecuaciones. En nuestro procedimiento nos quedaron entonces un par de ecuaciones ordenadas (la primera y la última) y tres ecuaciones sin ordenar:

$$\begin{aligned}
 u_L(t) - R_1 i_{R_1}(t) &= 0 \\
 u_L(t) - R_2 i_{R_2}(t) &= 0 \\
 \dot{\phi}(t) - u_L(t) &= 0 \\
 \phi(t) - L i_L(t) &= 0 \\
 i_L(t) + i_{R_1}(t) + i_{R_2}(t) &= 0
 \end{aligned}
 \qquad i_L(t) := \frac{\phi(t)}{L}$$

$$\begin{aligned}
 u_L(t) - R_1 i_{R_1}(t) &= 0 \\
 u_L(t) - R_2 i_{R_2}(t) &= 0 \\
 i_L(t) + i_{R_1}(t) + i_{R_2}(t) &= 0 \\
 \dot{\phi}(t) &:= u_L(t)
 \end{aligned}$$

Observando el resultado, es claro que nos queda un sistema de tres ecuaciones con tres incógnitas (i_{R_1} , i_{R_2} y u_L) que deben resolverse simultáneamente. Estas ecuaciones, en este caso, constituyen lo que se denomina un *lazo algebraico*.

Cuando el procedimiento de ordenamiento se encuentra en una situación en la cuál no se puede aplicar ninguna de las reglas porque todas las ecuaciones tienen al menos dos incógnitas y todas las incógnitas aparecen en al menos dos ecuaciones, puede verse que hay uno o más lazos algebraicos.

3.2.2. Algoritmo de Rasgado

A priori, podríamos resolver el problema implementando algo similar a lo que hicimos con el Código 3.1. De hecho, si hacemos esto directamente sobre la Ec.(3.8) tendríamos un sistema de 7 ecuaciones y 7 incógnitas, que podría reducirse a 5 ecuaciones e incógnitas tras eliminar las ecuaciones triviales. Utilizando sin embargo la última expresión, podríamos reducir aún más esto a un problema de 3 ecuaciones con 3 incógnitas.

Código 3.5: Función de Matlab/Octave con el modelo de la Ec.(3.8)

```
function dx=rlcircuit(x,t)
```

```

R1=1;R2=2;L=1; %parametros
phi=x; %estado
iL=phi/L;
%la siguiente funcion debe ser 0
F = @(z) rlcircuit_loop(z(1),z(2),z(3),iL);
z0=zeros(3,1);
z=fsolve(F,z0);
uL=z(1);
iR1=z(2);
iR2=z(3);
der_phi=uL;
dx=der_phi;
end

function res=rlcircuit_loop(uL,iR1,iR2,iL)
R1=1;R2=2;L=1; %parametros
%la siguiente funcion debe ser 0
res=zeros(3,1);
res(1)=uL-R1*iR1;
res(2)=uL-R2*iR2;
res(3)=iL+iR1+iR2;
end

```

Si bien esto es más eficiente que resolver un sistema de 5 ecuaciones como el que resultaría al colocar directamente la DAE, todavía podemos hacer algo más.

Cuando en el procedimiento de ordenamiento nos encontramos con un lazo algebraico, un truco es suponer que alguna incógnita del lazo es conocida (aunque no lo sea), guardar una ecuación donde aparezca dicha incógnita y continuar el proceso de ordenamiento, colocando al final del mismo la ecuación que nos guardamos.

Continuando con el ejemplo anterior, si suponemos conocida la variable algebraica u_L podríamos *guardar* la primera ecuación y ordenar la segunda que ahora tendría una única incógnita lo que permitiría utilizar la Regla 1.

$$\begin{aligned}
 u_L(t) - R_1 i_{R_1}(t) &= 0 & i_L(t) &:= \frac{\phi(t)}{L} \\
 \boxed{u_L(t) - R_2 i_{R_2}(t) = 0} & & i_{R_2}(t) &:= \frac{u_L(t)}{R_2} \\
 \dot{\phi}(t) - u_L(t) &= 0 & & \\
 \phi(t) - L i_L(t) &= 0 & & \\
 i_L(t) + i_{R_1}(t) + i_{R_2}(t) &= 0 & \dot{\phi}(t) &:= u_L(t)
 \end{aligned}$$

Tras este paso, la última ecuación tiene una única incógnita y podemos aplicar nuevamente la Regla 1:

$$\begin{aligned}
 u_L(t) - R_1 i_{R_1}(t) &= 0 & i_L(t) &:= \frac{\phi(t)}{L} \\
 u_L(t) - R_2 i_{R_2}(t) &= 0 & i_{R_2}(t) &:= \frac{u_L(t)}{R_2} \\
 \dot{\phi}(t) - u_L(t) &= 0 & i_{R_1}(t) &:= -i_{R_2}(t) - i_L(t) \\
 \phi(t) - L i_L(t) &= 0 & & \\
 \boxed{i_L(t) + i_{R_1}(t) + i_{R_2}(t) = 0} & & \dot{\phi}(t) &:= u_L(t)
 \end{aligned}$$

En este punto, podemos ver que la ecuación que reservamos en rojo ya no tiene incógnitas, excepto

que en realidad no conocemos $u_L(t)$. En definitiva, el sistema ordenado sería:

$$i_L(t) := \frac{\phi(t)}{L} \quad (3.9a)$$

$$i_{R_2}(t) := \frac{u_L(t)}{R_2} \quad (3.9b)$$

$$i_{R_1}(t) := -i_{R_2}(t) - i_L(t) \quad (3.9c)$$

$$u_L(t) - R_1 i_{R_1}(t) = 0 \quad (3.9d)$$

$$\dot{\phi}(t) := u_L(t) \quad (3.9e)$$

que tiene una única variable no explícita (u_L). De esta forma, podemos implementar una iteración sobre una única variable que resuelva el sistema de ecuaciones algebraicas que define la derivada del estado.

La función de Matlab/Octave del Código 3.6 brinda una implementación de esta idea y resulta mucho más eficiente que la del Código 3.5 .

Código 3.6: Función de Matlab/Octave con el modelo de la Ec.(3.9)

```
function dx=rlcircuit2(x,t)
    R1=1;R2=2;L=1; %parametros
    phi=x; %estado
    iL=phi/L;
    %la siguiente funcion debe ser 0
    F = @(uL) rlcircuit_loop2(uL,iL);
    uL=fsolve(F,0);
    der_phi=uL;
    dx=der_phi;
end

function res=rlcircuit_loop2(uL,iL)
    R1=1;R2=2;L=1; %parametros
    %la siguiente funcion debe ser 0
    iR2=uL/R2;
    iR1=-iR2-iL;
    res=uL-R1*iR1;
end
```

En este procedimiento de *rasgado* o *ruptura* del lazo (*tearing* en inglés) se denomina *variable de rasgado* a la variable que suponemos conocida (u_L en el ejemplo). En ocasiones se debe elegir más de una variable de rasgado y en modelos grandes es común que exista más de un lazo algebraico.

3.2.3. Obtención de las Ecuaciones de Estado

Podemos avanzar incluso un paso más y reemplazar en el sistema de ecuaciones del lazo utilizando las variables de rasgado y luego despejarlas de manera algebraica cuando esto sea posible, lo que transforma finalmente el sistema de DAEs en un sistema de ecuaciones de estado totalmente explícito.

Más precisamente, lo que haremos será reemplazar las variables algebraicas en el orden en que aparecen en función de las variables de estado y de rasgado para finalmente despejar estas últimas. En el caso del

ejemplo, podemos proceder como sigue a partir de la Ec.(3.9):

$$\begin{aligned}
 i_L(t) &:= \frac{\phi(t)}{L} \quad (\text{está bien porque } \phi(t) \text{ es estado}) \\
 i_{R_2}(t) &:= \frac{\mathbf{u}_L(t)}{R_2} \quad (\text{está bien porque } u_L(t) \text{ es variable de rasgado}) \\
 i_{R_1}(t) &:= -i_{R_2}(t) - i_L(t) = -\frac{\mathbf{u}_L(t)}{R_2} - \frac{\phi(t)}{L} \quad (\text{reemplazando de la 1ra. y 2da Ec.}) \\
 \mathbf{u}_L(t) - R_1 i_{R_1}(t) &= \mathbf{u}_L(t) - R_1 \left(-\frac{\mathbf{u}_L(t)}{R_2} - \frac{\phi(t)}{L}\right) = 0 \quad (\text{reemplazando } i_{R_1} \text{ de la 3ra Ec.}) \\
 \dot{\phi}(t) &:= u_L(t)
 \end{aligned} \tag{3.10}$$

Luego, la ecuación que habíamos guardado depende sólo de la variable de rasgado y de ninguna otra algebraica,

$$u_L(t) - R_1 \left(-\frac{u_L(t)}{R_2} - \frac{\phi(t)}{L}\right) = 0, \tag{3.11}$$

de donde podemos despejar $u_L(t)$ en función del estado $\phi(t)$:

$$u_L(t) = -\frac{R_1}{1 + \frac{R_1}{R_2}} \frac{\phi(t)}{L}. \tag{3.12}$$

Utilizando esta expresión podemos reemplazar la variable de rasgado en la Ec.(3.10) de donde

$$\dot{\phi}(t) = -\frac{R_1}{1 + \frac{R_1}{R_2}} \frac{\phi(t)}{L}. \tag{3.13}$$

El procedimiento de causalización que seguimos puede aplicarse con cualquier sistema, independientemente de que sea lineal o no lineal. La diferencia que encontraremos es que en casos no lineales es poco probable que podamos despejar la expresión de las variables de rasgado a partir de una ecuación como la Ec.(3.11) para llegar a la Ec.(3.12). Por lo tanto, en tales casos no será posible en general obtener un sistema de Ecuaciones de Estado explícitas. Más aún, algunos sistemas no lineales contienen expresiones de las cuales no es posible despejar analíticamente la variable que corresponde al ordenamiento horizontal según el algoritmo básico de causalización y deben implementarse iteraciones aunque no haya un lazo algebraico.

3.3. Sistemas de Índice Alto

3.3.1. Ejemplo Introductorio

Consideremos ahora el circuito de la Figura 3.3 con sus relaciones constitutivas y estructurales dadas por la Ec.(3.14).

$$C_1 \dot{u}_{C_1}(t) - i_{C_1}(t) = 0 \tag{3.14a}$$

$$C_2 \dot{u}_{C_2}(t) - i_{C_2}(t) = 0 \tag{3.14b}$$

$$u_R(t) - R i_R(t) = 0 \tag{3.14c}$$

$$i_R(t) + i_{C_1}(t) + i_{C_2}(t) = 0 \tag{3.14d}$$

$$u_{C_1}(t) - u_R(t) = 0 \tag{3.14e}$$

$$u_{C_1}(t) - u_{C_2}(t) = 0 \tag{3.14f}$$

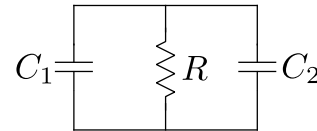


Figura 3.3: Circuito RC Paralelo

En principio este sistema tiene dos variables de estado (u_{C_1} y u_{C_2}) y cuatro variables algebraicas (i_{C_1} , i_{C_2} , i_R y u_R). Dado que tiene seis ecuaciones y seis incógnitas, el modelo estaría, a priori, bien planteado.

Sin embargo, si intentamos aplicar el procedimiento de causalización, podemos ver inmediatamente que Ec.(3.14f) no tiene incógnitas (recordemos que los estados se asumen conocidos) y por lo tanto no puede utilizarse.

La causa del problema es bastante evidente: estamos intentando utilizar dos variables de estado (u_{C_1} y u_{C_2}) en un modelo que es de primer orden ya que hay una ecuación, la Ec.(3.14f), que dice que ambas variables deben ser iguales y por lo tanto no podemos elegir independientemente sus condiciones iniciales. En general, cuando una ecuación se queda sin incógnitas durante el proceso de causalización se dice que la misma constituye una *Ecuación de Restricción* y esto se debe a que el modelo tiene un orden menor al número de variables que supusimos que eran estados porque aparecían sus derivadas.

Cuando ocurre esto, decimos que el sistema de DAEs es *estructuralmente singular*. En un caso general, podemos detectar la presencia de singularidades estructurales cuando al aplicar el procedimiento de causalización, en algún momento una ecuación se queda sin incógnitas.

Dado que no podemos causalizar este sistema, una primera opción podría ser intentar nuevamente implementar una solución de *fuerza bruta* y tratar de simular el sistema iterando sobre la DAE para obtener la derivada del estado \dot{x} en cada paso. Esta idea puede llevarse a cabo con el Código 3.7.

Código 3.7: Función de Matlab/Octave con el modelo de la Ec.(3.14)

```
function dx=rccircuit_ode(x,t)
    F = @(z) rccircuit_dae(x,z(1:2),z(3:6),t);
    z0=zeros(6,1);
    z=fsolve(F,z0);
    dx=z(1:2);
end

function res=rc_circuit_dae(x,dx,a,t)
    C1=1;C2=2;R=1; %parametros
    uC1=x(1); uC2=x(2); %estados
    der_uC1=dx(1); der_uC2=dx(2); %derivadas de estados
    uR=a(1); iR=a(2); iC1=a(3); iC2=a(4); %variables algebraicas
    res=zeros(6,1); %residuo (debe ser cero)
    res(1) = C1*der_uC1-iC1;
    res(2) = C2*der_uC2-iC2;
    res(3) = uR-R*iR;
    res(4) = iR+iC1+iC2;
    res(5) = uC1-uR;
    res(6) = uC1-uC2;
end
```

Desafortunadamente, la idea no va a funcionar. Un primer problema es que tenemos que dar dos condiciones iniciales (el estado tiene dos componentes). Poniendo de todas formas ambas condiciones iguales para cumplir con la última ecuación, al ejecutar

```
[t,x]=feuler(@rccircuit_ode,[1;1],0.1,0,1);
```

la función que resuelve las ecuaciones algebraicas de la DAE (`fsolve`) se encuentra con que la matriz que debe invertir para la iteración de Newton es singular y por lo tanto no puede resolver correctamente el sistema.

3.3.2. Algoritmo de Pantelides

¿Cómo se puede solucionar esto?. La idea básica, debida a Constantino Pantelides [Pan88], es derivar respecto al tiempo las ecuaciones de restricción. Si $u_{C_1}(t) - u_{C_2}(t) = 0$, luego debe cumplirse también que $\dot{u}_{C_1}(t) - \dot{u}_{C_2}(t) = 0$ por lo que agregaremos esta última ecuación al sistema. Dado que ahora tenemos una ecuación más, necesitamos también una incógnita más, lo que lograremos decidiendo que una de las variables de estado de la Ecuación de Restricción no sea más variable de estado.

Suponiendo entonces que u_{C_2} no es más un estado, por ejemplo, el nuevo sistema de ecuaciones queda:

$$C_1 \dot{u}_{C_1}(t) - i_{C_1}(t) = 0 \quad (3.15a)$$

$$C_2 du_{C_2}(t) - i_{C_2}(t) = 0 \quad (3.15b)$$

$$u_R(t) - R i_R(t) = 0 \quad (3.15c)$$

$$i_R(t) + i_{C_1}(t) + i_{C_2}(t) = 0 \quad (3.15d)$$

$$u_{C_1}(t) - u_R(t) = 0 \quad (3.15e)$$

$$u_{C_1}(t) - u_{C_2}(t) = 0 \quad (3.15f)$$

$$\dot{u}_{C_1}(t) - du_{C_2}(t) = 0 \quad (3.15g)$$

En este nuevo sistema tenemos siete ecuaciones y siete incógnitas (ahora u_{C_2} es una variable algebraica). Además, lo que antes llamábamos \dot{u}_{C_2} ahora es una nueva variable algebraica du_{C_2} que representa lo que se llama una derivada *ficticia* (*dummy derivative* en inglés). Si bien proviene de lo que era la derivada de u_{C_2} , para el nuevo sistema es sólo una nueva variable algebraica.

Veamos entonces el resultado de aplicar el algoritmo de causalización sobre este nuevo sistema. Podemos ver fácilmente que es posible aplicar la Regla 1 a las Ecs.(3.15e) y (3.15f).

$$\begin{array}{l} C_1 \dot{u}_{C_1}(t) - i_{C_1}(t) = 0 \\ C_2 du_{C_2}(t) - i_{C_2}(t) = 0 \\ u_R(t) - R i_R(t) = 0 \\ i_R(t) + i_{C_1}(t) + i_{C_2}(t) = 0 \\ u_{C_1}(t) - u_R(t) = 0 \\ u_{C_1}(t) - u_{C_2}(t) = 0 \\ \dot{u}_{C_1}(t) - du_{C_2}(t) = 0 \end{array} \quad \begin{array}{l} u_R(t) := u_{C_1}(t) \\ u_{C_2}(t) := u_{C_1}(t) \end{array}$$

Tras estos dos primeros pasos, la tercera ecuación tiene una única incógnita y puede aplicarse nuevamente la Regla 1 para calcular $i_R(t)$.

$$\begin{array}{l} C_1 \dot{u}_{C_1}(t) - i_{C_1}(t) = 0 \\ C_2 du_{C_2}(t) - i_{C_2}(t) = 0 \\ u_R(t) - R i_R(t) = 0 \\ i_R(t) + i_{C_1}(t) + i_{C_2}(t) = 0 \\ u_{C_1}(t) - u_R(t) = 0 \\ u_{C_1}(t) - u_{C_2}(t) = 0 \\ \dot{u}_{C_1}(t) - du_{C_2}(t) = 0 \end{array} \quad \begin{array}{l} u_R(t) := u_{C_1}(t) \\ u_{C_2}(t) := u_{C_1}(t) \\ i_R(t) := \frac{u_R(t)}{R} \end{array}$$

Tras este paso, nos encontramos con un sistema de cuatro ecuaciones y cuatro incógnitas, donde todas las incógnitas aparecen en al menos dos ecuaciones y donde todas las ecuaciones contienen al menos dos incógnitas. Es decir, tenemos un *lazo algebraico* en las variables \dot{u}_{C_1} , i_{C_1} , i_{C_2} y du_{C_2} .

Para continuar, utilizaremos como variable de rasgado a \dot{u}_{C_1} reservando la primera ecuación. Asumiendo entonces conocida \dot{u}_{C_1} se puede despejar du_{C_2} de la última ecuación, luego i_{C_2} de la segunda, tras esto i_{C_1} se obtiene de la suma de corrientes y esto deja a la primera ecuación sin otra incógnita que la variable de rasgado completando la ruptura del lazo.

Esto se traduce en el siguiente sistema de ecuaciones ordenadas que contienen sólo una incógnita:

$$u_R(t) := u_{C_1}(t) \quad (3.16a)$$

$$u_{C_2}(t) := u_{C_1}(t) \quad (3.16b)$$

$$i_R(t) := \frac{u_R(t)}{R} \quad (3.16c)$$

$$du_{C_2}(t) := \dot{u}_{C_1}(t) \quad (3.16d)$$

$$i_{C_2}(t) := C_2 du_{C_2} \quad (3.16e)$$

$$i_{C_1}(t) := -i_R(t) - i_{C_2}(t) \quad (3.16f)$$

$$C_1 \dot{u}_{C_1}(t) - i_{C_1}(t) = 0 \quad (3.16g)$$

La función de Matlab/Octave del Código 3.8 brinda una implementación a partir de las ecuaciones obtenidas tras el ordenamiento y el rasgado del lazo algebraico.

Código 3.8: Función de Matlab/Octave con el modelo de la Ec.(3.16)

```
function dx=rccircuit2(x,t)
    C1=1;C2=2;R=1; %parametros
    uC1=x; %estado
    uR=uC1;
    uC2=uC1;
    iR=uR/R;
    %la siguiente funcion debe ser 0
    F = @(dot_uC1) rccircuit_loop2(dot_uC1,iR);
    dot_uC1=fsolve(F,0);
    dx=dot_uC1;
end

function res=rccircuit_loop2(dot_uC1,iR)
    C1=1;C2=2;R=1; %parametros
    duC2=dot_uC1;
    iC2=C2*duC2;
    iC1=-iR-iC2;
    %la siguiente funcion debe ser 0
    res=C1*dot_uC1-iC1;
end
```

Podemos también obtener de manera explícita la ecuación de estados reemplazando las variables algebraicas en función de los estados y de la variable de rasgado \dot{u}_{C_1} en la Ec.(3.16):

$$u_R(t) := u_{C_1}(t)$$

$$u_{C_2}(t) := u_{C_1}(t)$$

$$i_R(t) := \frac{u_R(t)}{R} = \frac{u_{C_1}(t)}{R}$$

$$du_{C_2}(t) := \dot{u}_{C_1}(t)$$

$$i_{C_2}(t) := C_2 du_{C_2} = C_2 \dot{u}_{C_1}(t)$$

$$i_{C_1}(t) := -i_R(t) - i_{C_2}(t) = -\frac{u_{C_1}(t)}{R} - C_2 \dot{u}_{C_1}(t)$$

$$C_1 \dot{u}_{C_1}(t) - i_{C_1}(t) = C_1 \dot{u}_{C_1}(t) + \frac{u_{C_1}(t)}{R} + C_2 \dot{u}_{C_1}(t) = 0$$

y finalmente podemos despejar de la última ecuación la expresión de la variable de rasgado

$$\dot{u}_{C_1}(t) = -\frac{u_{C_1}(t)}{R(C_1 + C_2)} \quad (3.17)$$

que en este caso es directamente la ecuación de estados.

En definitiva, utilizando el Algoritmo de Pantelides transformamos un sistema que no puede causalizarse en uno que contiene un lazo algebraico.

El algoritmo de Pantelides (en una versión simplificada) se puede resumir mediante los siguientes pasos:

- Reemplazar las variables algebraicas por sus expresiones en función de los estados en las ecuaciones de restricción (esto no fue necesario en el ejemplo).
- Derivar respecto al tiempo las ecuaciones de restricción, agregarlas al sistema y transformar una variable de estado en algebraica por cada restricción.
- Al derivar, llamar $\dot{x}_i(t)$ a las derivadas de los estados $x_i(t)$ y $dx_j(t)$ a las derivadas de las variables algebraicas $x_j(t)$ que dejaron de ser estados.
- Verificar que el nuevo sistema sea causalizable. Si no lo es, se debe aplicar nuevamente el procedimiento.

3.3.3. El índice de perturbación de una DAE

Los sistemas de ecuaciones diferenciales algebraicas que no contienen lazos algebraicos y que pueden causalizarse completamente utilizando las Reglas 1 y 2 se denominan *DAEs de índice 0*. En estos casos, el sistema puede generalmente transformarse en una ODE explícita reordenando las ecuaciones.

Cuando un sistema contiene lazos algebraicos pero puede eventualmente obtenerse la derivada del vector de estados \dot{x} sin modificar el sistema se dice que es una *DAE de índice 1*. Cuando el sistema no se puede causalizar sin modificarlo, si el sistema está bien formulado, se dice que es una *DAE de índice alto*.

El circuito RC de la Ec.(3.14) constituye una DAE de índice 2 ya que fue necesario aplicar una única vez el algoritmo de Pantelides para que nos quede un sistema de índice 1. Hay sistemas en los cuales es necesario aplicar Pantelides más de una vez. Si en un sistema debemos aplicar dos veces el algoritmo para llegar a una DAE de índice 1, entonces diremos que el sistema tiene índice 3, y así sucesivamente para los índices superiores.

Veamos entonces un ejemplo algo más complejo que el anterior, consistente en el modelo del péndulo de la Fig.3.4 dado por las ecuaciones diferenciales algebraicas de la Ec.(3.18), donde x e y representan las coordenadas de la masa, v_x y v_y las velocidades horizontal y vertical y $F(t)$ es la fuerza que realiza el cable en el sentido longitudinal.

$$\dot{x}(t) - v_x(t) = 0 \quad (3.18a)$$

$$\dot{y}(t) - v_y(t) = 0 \quad (3.18b)$$

$$m \dot{v}_x(t) + \frac{x(t)}{L} F(t) = 0 \quad (3.18c)$$

$$m \dot{v}_y(t) + \frac{y(t)}{L} F(t) - m g = 0 \quad (3.18d)$$

$$x(t)^2 + y(t)^2 - L^2 = 0 \quad (3.18e)$$

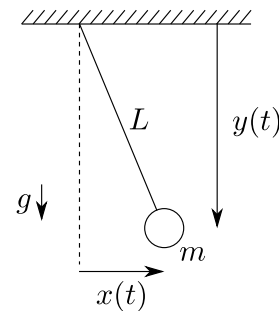


Figura 3.4: Péndulo

Tenemos en este caso 5 ecuaciones y 5 incógnitas, pero al igual que en el ejemplo anterior, la última ecuación no contiene a priori ninguna incógnita. Esto es nuevamente esperable ya que el sistema es claramente de segundo orden y lo estamos modelando con 4 potenciales variables de estado.

Procederemos entonces con el algoritmo de Pantelides derivando la ecuación de restricción, de donde se obtiene

$$\dot{\mathbf{x}}(t) - v_x(t) = 0 \quad (3.19a)$$

$$d\mathbf{y}(t) - v_y(t) = 0 \quad (3.19b)$$

$$m \dot{v}_x(t) + \frac{x(t)}{L} \mathbf{F}(t) = 0 \quad (3.19c)$$

$$m \dot{v}_y(t) + \frac{y(t)}{L} \mathbf{F}(t) - m g = 0 \quad (3.19d)$$

$$x(t)^2 + y(t)^2 - L^2 = 0 \quad (3.19e)$$

$$2 x(t) \dot{\mathbf{x}}(t) + 2 y(t) d\mathbf{y}(t) = 0 \quad (3.19f)$$

Al derivar la ecuación de restricción, decidimos además que $y(t)$ no sea más variable de estado por lo que $dy(t)$ es una *derivada ficticia*.

La antigua restricción de la Ec.(3.19e) contiene ahora una incógnita $y(t)$, que según la Regla 1 del algoritmo de causalización deberemos despejar usando dicha ecuación. De manera similar, la Ec.(3.19a) contiene sólo la incógnita $\dot{x}(t)$, por lo que la deberemos despejar de dicha ecuación. Esto dejará a la nueva Ec.(3.19f) con una única incógnita $dy(t)$ que deberemos despejar desde allí. Luego, tras los tres primeros pasos del algoritmo de causalización tendremos lo siguiente:

$$\begin{aligned} \dot{\mathbf{x}}(t) - v_x(t) &:= 0 & y(t) &:= \sqrt{L^2 - x(t)^2} \\ dy(t) - v_y(t) &= 0 & \dot{\mathbf{x}}(t) &:= v_x(t) \\ m \dot{v}_x(t) + \frac{x(t)}{L} \mathbf{F}(t) &= 0 & dy(t) &:= -\frac{x(t)}{y(t)} \dot{\mathbf{x}}(t) \\ m \dot{v}_y(t) + \frac{y(t)}{L} \mathbf{F}(t) - m g &= 0 \\ x(t)^2 + y(t)^2 - L^2 &= 0 \\ 2 x(t) \dot{\mathbf{x}}(t) + 2 y(t) d\mathbf{y}(t) &= 0 \end{aligned}$$

Esta vez la ecuación que vincula $dy(t)$ con $v_y(t)$ no tiene más incógnitas, lo que constituye una ecuación de restricción en el nuevo sistema.

Aplicaremos nuevamente el algoritmo de Pantelides, reemplazando en primer lugar las variables algebraicas de la ecuación de restricción en función de las variables de estado. Para esto aprovecharemos las ecuaciones ya causalizadas como sigue:

$$\begin{aligned} y(t) &:= \sqrt{L^2 - x(t)^2} \quad (\text{está bien porque } x(t) \text{ es estado}). \\ \dot{\mathbf{x}}(t) &:= v_x(t) \quad (\text{está bien porque } v_x(t) \text{ es estado}). \\ dy(t) &:= -\frac{x(t)}{y(t)} \dot{\mathbf{x}}(t) = -\frac{x(t)}{\sqrt{L^2 - x(t)^2}} v_x(t) \quad (\text{reemplazando con las dos ecs. anteriores}) \end{aligned}$$

con lo que la ecuación de restricción queda

$$dy(t) - v_y(t) = -\frac{x(t)}{\sqrt{L^2 - x(t)^2}} v_x(t) - v_y(t) = 0 \quad (3.20)$$

que depende sólo de las variables de estado. Tras este paso, eliminamos v_y de la lista de variables de estado y su derivada que era \dot{v}_y será ahora la derivada ficticia dv_y .

Luego, derivando la ecuación de restricción (3.20) obtenemos

$$\frac{d\left(-\frac{x(t)}{\sqrt{L^2 - x(t)^2}} v_x(t) - v_y(t)\right)}{dt} = -h(x(t), \dot{\mathbf{x}}(t), v_x(t), \dot{v}_x(t)) - dv_y(t) = 0 \quad (3.21)$$

donde $h(x(t), \dot{x}(t), v_x(t), \dot{v}_x(t))$ es la expresión de la derivada del primer término, cuya obtención queda como ejercicio.

Agregando entonces la Ec.(3.21) al sistema de ecuaciones de la Ec.(3.19), tenemos

$$\dot{\mathbf{x}}(t) - v_x(t) = 0 \quad (3.22a)$$

$$d\mathbf{y}(t) - v_y(t) = 0 \quad (3.22b)$$

$$m \dot{v}_x(t) + \frac{x(t)}{L} F(t) = 0 \quad (3.22c)$$

$$m dv_y(t) + \frac{y(t)}{L} F(t) - m g = 0 \quad (3.22d)$$

$$x(t)^2 + y(t)^2 - L^2 = 0 \quad (3.22e)$$

$$2 x(t) \dot{\mathbf{x}}(t) + 2 y(t) d\mathbf{y}(t) = 0 \quad (3.22f)$$

$$-h(x(t), \dot{\mathbf{x}}(t), v_x(t), \dot{v}_x(t)) - dv_y(t) = 0 \quad (3.22g)$$

Si se aplica el procedimiento de causalización a este sistema, podrá verse que el mismo encontrará un lazo algebraico en 3 variables ($\dot{v}_x(t)$, $F(t)$, $dv_y(t)$) que puede causalizarse usando como variable de rasgado $\dot{v}_x(t)$, por ejemplo, de donde se obtiene:

$$\dot{x}(t) := v_x(t) \quad (3.23a)$$

$$y(t) := \pm \sqrt{L^2 - x^2} \quad (3.23b)$$

$$dy(t) := \frac{x(t)}{y(t)} \dot{x}(t) \quad (3.23c)$$

$$v_y(t) := dy(t) \quad (3.23d)$$

$$dv_y(t) := -h(x(t), \dot{x}(t), v_x(t), \dot{v}_x(t)) \quad (3.23e)$$

$$F(t) := \frac{m L}{y(t)} (g - dv_y(t)) \quad (3.23f)$$

$$m \dot{v}_x(t) + \frac{x(t)}{L} F(t) = 0 \quad (3.23g)$$

Dado que tuvimos que utilizar dos veces el algoritmo de Pantelides, este problema es de índice 3, algo que es muy habitual en los sistemas mecánicos con restricciones.

3.3.4. Elección de las Variables de Estado

Si observamos el sistema de la Ec.(3.23) podemos ver un par de problemas. En primer lugar, el signo de $y(t)$ no queda bien definido. En segundo lugar, $y(t)$ aparece dividiendo en varias ecuaciones. Mientras asumamos que el péndulo se puede mover sin alcanzar un ángulo de 90 grados respecto de la posición vertical, no habrá problemas ya que podemos tomar siempre la solución positiva para $y(t)$ y nunca ocurrirá que $y(t) = 0$.

Ahora bien, si hubiésemos elegido $y(t)$ como variable de estado en lugar de $x(t)$ tendríamos un problema más grave. En tal caso, el signo de $x(t)$ no estaría definido y eso sí es un problema cuando el péndulo se mueve cerca de la posición de reposo. Más aún, en lugar de divisiones por $y(t)$, tendríamos divisiones por $x(t)$ lo que aparejaría divisiones por cero cada vez que el péndulo queda en posición vertical.

Esto puede verse directamente en el modelo original de la Figura 3.4 donde es evidente que conocer el valor de $x(t)$ nos permite conocer $y(t)$ mientras que lo contrario no es cierto ya que hay dos soluciones. Además, con el péndulo en posición vertical no es posible calcular la aceleración horizontal $\dot{v}_x(t)$ a partir de conocer la aceleración vertical $\dot{v}_y(t)$ en el modelo original de la Ec.(3.18).

Todo esto delata que nuestra elección de $x(t)$ y de $v_x(t)$ como estados fue acertada. De otra forma, hubiéramos llegado a un modelo *correcto* desde el punto de vista del ordenamiento pero incapaz de ser simulado.

En este caso, una mejor opción hubiera sido incorporar el ángulo $\theta(t)$ a partir de la relación $x(t) = L \cos(\theta(t))$ y elegirlo como variable de estado dejando $x(t)$ e $y(t)$ como algebraicas. De esta manera desaparecerían los problemas de indeterminación entre $x(t)$ e $y(t)$

3.4. DAEs y Diagramas de Bloques

3.4.1. Diagramas de Bloques y Relaciones Causales

Los Diagramas de Bloques constituyen una herramienta para expresar relaciones matemáticas de manera gráfica. Una diferencia primordial entre las DAEs y los DBs es que los últimos, además de expresar relaciones matemáticas entre variables, asumen relaciones causales entre las mismas. Es decir, dada una relación

$$K a_1(t) - a_2(t) = 0$$

en un Diagrama de Bloques tenemos que asumir que $a_2(t) := K a_1(t)$ o bien que $a_1(t) := \frac{a_2(t)}{K}$. Estas opciones se traducen en los dos posibles Diagrama de Bloques de la Figura 3.5.

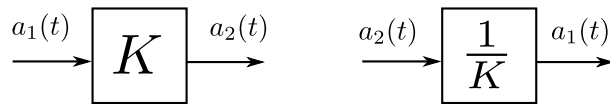


Figura 3.5: Dos posibles Diagramas de Bloques para la relación $K a_1(t) - a_2(t) = 0$.

Esta falta de unicidad en la representación de las relaciones matemáticas hace que los DBs no sean una herramienta muy adecuada para construir modelos. De hecho, asumir una determinada causalidad entre variables de un modelo puede provocar que luego no podamos *conectar* dicho modelo con otro. Por este motivo, las DAEs y los lenguajes acausales en general son las representaciones más adecuadas para las tareas de modelado.

Sin embargo, los DBs son una herramienta útil en muchos casos ya que permiten analizar de manera simple algunas propiedades importantes de los modelos que representan. Veremos entonces como construir de manera simple Diagramas de Bloques a partir de las DAEs y como en ese proceso de construcción podremos vislumbrar algunas características estructurales (como el índice de perturbación) del modelo.

3.4.2. Procedimiento para obtener un DB desde una DAE

El procedimiento para obtener un Diagrama de Bloques desde una DAE es muy simple y puede resumirse en los siguientes pasos:

1. Eliminar las ecuaciones triviales y los correspondientes alias (no necesario este paso, es sólo para simplificar).
2. Elegir una variable de la DAE para comenzar y colocar una flecha en el DB con dicha variable. Preferentemente comenzar por una variable de estado.
3. Tomar una variable del DB que no esté calculada por ningún bloque.
4. Buscar una ecuación de la DAE donde aparezca dicha variable teniendo en cuenta lo siguiente:
 - Si la variable es un estado $x_i(t)$ usar la ecuación $x_i(t) = \int \dot{x}_i(t)$.
 - Si la variable es la derivada de un posible estado $\dot{x}_i(t)$ y no quedara ninguna ecuación, usar la ecuación $\dot{x}_i(t) = \frac{d}{dt}x_i(t)$.
5. Agregar un bloque que calcule la variable en cuestión utilizando la ecuación elegida. Si se usó la ecuación $x_i(t) = \int \dot{x}_i(t)$ el bloque será un integrador mientras que si se usó $\dot{x}_i(t) = \frac{d}{dt}x_i(t)$ el bloque será un derivador.
6. Eliminar la ecuación utilizada de la DAE.

7. Si las variables de entrada al bloque agregado ya están calculadas en el DB como salida de otro bloque, conectarlas.
8. Si queda alguna variable en el DB que aún no es salida de ningún bloque, volver al paso 3.

Veremos entonces estos pasos sobre el modelo del circuito RLC serie de la Ec.(3.1), simplificado tras eliminar las ecuaciones triviales.

$$u_R(t) - R \dot{q}(t) = 0 \tag{3.24a}$$

$$q(t) - C u_C(t) = 0 \tag{3.24b}$$

$$\phi(t) - L \dot{q}(t) = 0 \tag{3.24c}$$

$$u_S(t) - v(t) = 0 \tag{3.24d}$$

$$\dot{\phi}(t) + u_R(t) + u_C(t) - u_S(t) = 0 \tag{3.24e}$$

Tomando entonces el estado $q(t)$ para comenzar y colocando un integrador nos queda $\dot{q}(t)$ como entrada del bloque. Utilizando la ecuación $\phi(t) - L \dot{q}(t) = 0$ para calcular $\dot{q}(t)$, nos queda $\phi(t)$ como entrada. El procedimiento completo puede verse en la Figura 3.6.

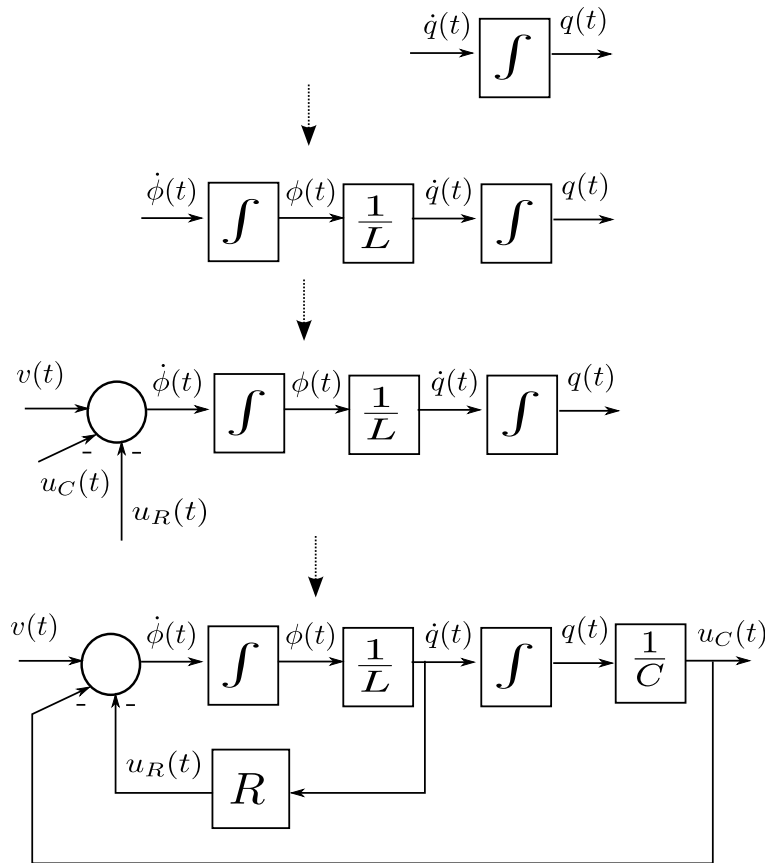


Figura 3.6: Construcción del DB a partir del sistema de DAEs de la Ec.(3.24).

Para que el DB tenga esta forma tuvimos que elegir la ecuación $\phi(t) - L \dot{q}(t) = 0$ para calcular $\dot{q}(t)$. Si hubiésemos elegido la ecuación $u_R(t) - R \dot{q}(t) = 0$ nos habría quedado un derivador en lugar de un integrador en este caso. Esto no sería un problema a priori, pero muchas de las propiedades que pueden analizarse sobre los DBs requieren que el mismo contenga sólo integradores.

Una forma más sencilla de construir el DB es partiendo del sistema ya ordenado, como la Ec.(3.3), colocar un integrador por cada estado y luego seguir una por una las asignaciones obteniendo las restantes señales hacia adelante.

3.4.3. Diagramas de Bloques y Lazos Algebraicos

Consideremos nuevamente el ejemplo del circuito RL que contenía un lazo algebraico, dado por la Ec.(3.8). Eliminando las ecuaciones triviales y alias obtenemos:

$$\dot{\phi}(t) - R_1 i_{R_1}(t) = 0 \tag{3.25a}$$

$$\dot{\phi}(t) - R_2 i_{R_2}(t) = 0 \tag{3.25b}$$

$$\phi(t) - L i_L(t) = 0 \tag{3.25c}$$

$$i_L(t) + i_{R_1}(t) + i_{R_2}(t) = 0 \tag{3.25d}$$

Si seguimos el procedimiento para obtener el DB a partir de estas ecuaciones, podemos construir el Diagrama de Bloques de la Figura 3.7. En el mismo puede verse que hay un camino cerrado entre las variables $i_{R_1}(t)$, $i_{R_2}(t)$ y $\dot{\phi}(t)$ que no pasa por ningún integrador. Dicho camino refleja el lazo algebraico que habíamos detectado en la DAE correspondiente.

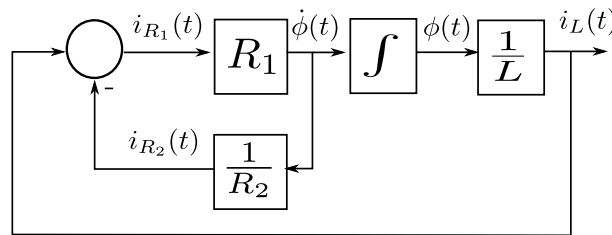


Figura 3.7: Diagrama de Bloques a partir de la Ec.(3.25).

En casos generales, no sólo en este ejemplo, los lazos algebraicos de un sistema de DAEs se reflejan en la presencia de un camino cerrado a través de bloques estáticos en el Diagrama de Bloques.

3.4.4. Diagramas de Bloques y Sistemas de Índice Alto

Si construimos ahora un Diagrama de Bloques para el modelo del péndulo dado por la Ec.(3.18), podemos obtener el que se muestra en la Figura 3.8 donde los bloques con doble recuadro indican operaciones no lineales. Estos bloques calculan las señales:

$$y(t) := \sqrt{L^2 - x(t)^2}$$

$$F(t) := \frac{L}{y(t)} (m g - m \dot{v}_y(t))$$

$$\dot{v}_x(t) := -\frac{x(t) F(t)}{m L}$$

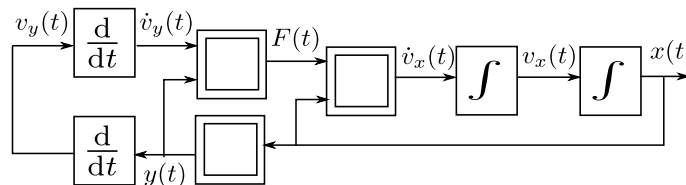


Figura 3.8: Diagrama de Bloques a partir de la Ec.(3.18).

En este caso, independientemente de las ecuaciones que elijamos para calcular cada variable, siempre necesitaremos bloques derivadores.

En las épocas de la computación analógica, la presencia de un derivador en un DB traía aparejados problemas de ruido, ya que las simulaciones se realizaban utilizando circuitos analógicos que emulaban los elementos de los diagramas de bloques. Si bien ese problema ya no existe, la presencia de derivadores

hará difícil el análisis de ciertas propiedades. De hecho, la obtención de las Ecuaciones de Estado a partir de un DB como el de la Fig.3.8 es tan complicada como la de obtener dichas ecuaciones a partir de la DAE correspondiente de índice alto (ya sabemos que hay que usar un método de reducción de índice como el de Pantelides).

Más allá del ejemplo analizado, vale en general que las DAEs de índice alto implican que el DB correspondiente contendrá derivadores.

3.5. Herramientas de Software para Causalización de DAEs

Distintas variantes de los algoritmos que vimos en este capítulo se encuentran implementados en las distintas herramientas de software de modelado y simulación. Particularmente, estos algoritmos constituyen un pilar fundamental de las herramientas que utilizan el lenguaje Modelica.

Gracias a estos procedimientos podemos utilizar directamente los modelos de DAEs sin preocuparnos demasiado por la presencia de lazos algebraicos o singularidades estructurales.

Los compiladores de Modelica como Dymola, WoframSystemModeler y OpenModelica comienzan el procesamiento de los modelos con un procedimiento de *aplanado* en caso que el modelo utilice conexiones y otras características orientadas a objetos (que veremos más adelante en el Capítulo 5) obteniendo un conjunto de DAEs.

Luego, se eliminan las ecuaciones triviales y las variables algebraicas (*alias*) correspondientes. En el siguiente paso se realiza la causalización, que sigue en parte las ideas que vimos en este capítulo. Finalmente produce el código en lenguaje C con las ecuaciones del modelo causalizadas.

Este código puede ser utilizado por distintos solvers (DOPRI, DASSL, etc) de manera muy similar a lo que hicimos nosotros cuando invocábamos desde Octave o Matlab al método de Forward Euler.

Como vimos, un problema con los sistemas de índice alto es decidir qué variables se utilizarán como estados. Para esto, los compiladores de Modelica generan código con distintas opciones y luego, durante la simulación, deciden cuál es la mejor (evitando divisiones por cero, por ejemplo). Esto se denomina *selección dinámica de estados*. El lenguaje Modelica además brinda la posibilidad de decidir que una variable sea tratada como estado de manera obligatoria o preferencial, o lo contrario, que nunca sea considerada como estado.

Un modelo para el péndulo que refleja directamente la Ec.(3.18) en Modelica y que aclara que preferimos usar x como variable de estado cuando sea posible se muestra en el Código 3.9.

Código 3.9: Modelo de la Ec.(3.18) en Modelica

```

model Pendulum
  Real x(start=0.1, stateSelect=StateSelect.prefer), vx, y, vy, F;
  parameter Real L=1, m=1, g=9.8;
equation
  der(x)-vx=0;
  der(y)-vy=0;
  m*der(vx)+x/L*F=0;
  m*der(vy)+y/L*F-m*g=0;
  x^2+y^2-L^2=0;
end Pendulum;

```

Utilizando OpenModelica, a partir de este modelo el compilador genera un código con asignaciones casi idénticas a las de la Ec.(3.23) y la simulación funciona correctamente. Si cambiamos la opción `stateSelect=StateSelect.prefer` a la variable y , en cambio, aparecen distintos errores durante la simulación.

En OpenModelica pueden verse las ecuaciones causalizadas generadas mediante el *Transformational Debugger*.

3.6. Bibliografía Complementaria

Este capítulo, como mencionamos al comienzo del mismo, está basado en gran medida en el Capítulo 7 del libro *Continuous System Simulation* [CK06]. Una descripción mucho más detallada de los algoritmos y su implementación en Modelica puede encontrarse en el libro de Peter Fritzson *Principles of object-oriented modeling and simulation with Modelica 3.3: a cyber-physical approach* [Fri14].

Los algoritmos que vimos en el capítulo son en realidad versiones simplificadas y adaptadas de los originales, que se basan en *Teoría de Grafos*. En el procedimiento de causalización que se usa en las herramientas de software se comienza resolviendo un problema de *máximo matching* entre ecuaciones e incógnitas lo que permite detectar singularidades estructurales. Si las hubiera, se aplica el algoritmo de Pantelides en su versión original que utiliza análisis sobre el grafo de estructura [Pan88]. Una vez reducido el índice, se deben separar los distintos lazos algebraicos, lo que se logra detectando los *componentes fuertemente conexos* del grafo para lo que se utiliza generalmente el *Algoritmo de Tarjan* [Tar72] que resuelve a la vez el ordenamiento vertical de los sistemas de ecuaciones.

3.7. Problemas Propuestos

[P3.1] Rectificador de Media Onda

El circuito de la Figura 3.9 puede modelarse mediante el sistema de DAEs de la Ec.(P3.1a).

$$\begin{aligned}
 \dot{\phi}(t) - u_L(t) &= 0 \\
 L i_L(t) - \phi(t) &= 0 \\
 u_R(t) - R i_R(t) &= 0 \\
 u_D(t) - g(i_D(t)) &= 0 \\
 u_S(t) - U \sin(\omega t) &= 0 \quad (\text{P3.1a}) \\
 i_L(t) - i_S(t) &= 0 \\
 i_L(t) - i_R(t) &= 0 \\
 i_L(t) - i_D(t) &= 0 \\
 u_S(t) - u_R(t) - u_D(t) - u_L(t) &= 0
 \end{aligned}$$

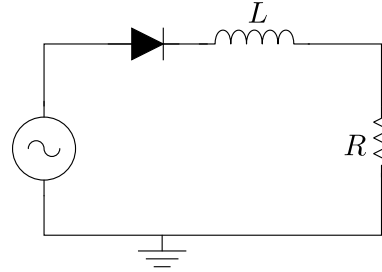


Figura 3.9: Rectificador de Media Onda con carga RL serie.

En las ecuaciones, la función $g(i_D(t))$ representa relación Volt-Ampère (no lineal) del diodo. Se pide:

1. Enumerar las variables de estado y las variables algebraicas del modelo.
2. Eliminar todas las ecuaciones triviales posibles y las correspondientes variables algebraicas.
3. Aplicar el algoritmo de causalización, detectando la posible presencia de lazos algebraicos o singularidades estructurales.
4. Suponer que los parámetros son $R = 100\Omega$, $L = 0.001\text{Hy}$, $\omega = 2\pi 50$, $U = 220\sqrt{2}\text{V}$ y que la función $g(i_D)$ está definida como

$$g(i_D) = \begin{cases} R_{\text{on}} i_D & \text{si } i_D > 0 \\ R_{\text{off}} i_D & \text{en otro caso} \end{cases} \quad (\text{P3.1b})$$

donde $R_{\text{on}} = 10^{-6}\Omega$ y $R_{\text{off}} = 10^6\Omega$. Con estos valores,

- a) Escribir una función de Octave o Matlab que represente las ecuaciones y simular el circuito usando Forward Euler y Backward Euler con un paso de integración $h = 0.001$ hasta un tiempo final $t_f = 0.1$.
- b) Implementar la simulación con OpenModelica y comparar los resultados. Una ley discontinua como la Ec.(P3.1b) puede implementarse como

$$\text{u_D} = \text{if } i_D > 0 \text{ then } R_{\text{on}} * i_D \text{ else } R_{\text{off}} * i_D;$$

5. Escribir el modelo en Ecuaciones de Estado.

[P3.2] Rectificador de Media Onda II

El circuito rectificador de la Figura 3.10 es similar al del Problema P3.1 salvo que ahora tiene una resistencia en paralelo con el inductor. El mismo puede modelarse mediante las relaciones constitutivas y estructurales de la Ec.(P3.2a).

$$\begin{aligned}
 \dot{\phi}(t) - u_L(t) &= 0 \\
 L i_L(t) - \phi(t) &= 0 \\
 u_R(t) - R i_R(t) &= 0 \\
 u_D(t) - g(i_D(t)) &= 0 \\
 u_S(t) - U \sin(\omega t) &= 0 \\
 u_{R_2}(t) - R_2 i_{R_2}(t) &= 0 \\
 i_{R_2}(t) - i_L(t) - i_R(t) &= 0 \\
 i_S(t) - i_D(t) &= 0 \\
 i_D(t) - i_{R_2}(t) &= 0 \\
 u_S(t) - u_L(t) - u_D(t) - u_{R_2}(t) &= 0 \\
 u_R(t) - u_L(t) &= 0
 \end{aligned} \tag{P3.2a}$$

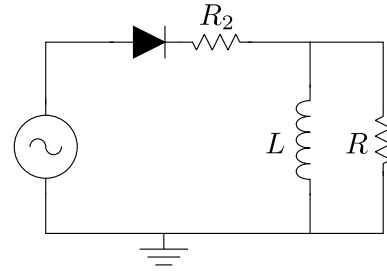


Figura 3.10: Rectificador de Media Onda con carga RL en paralelo.

1. Repetir los primeros tres puntos del Problema P3.1. Si hubiera un lazo algebraico, aplicar el procedimiento de *rasgado* tomando como variable de rasgado la tensión en el diodo $i_D(t)$ y reservando la ecuación constitutiva del diodo.
2. Repetir el punto anterior utilizando $u_D(t)$ en lugar de $i_D(t)$. ¿Funciona el algoritmo con una sola variable de rasgado? (si no funciona no es necesario seguir).
3. Repetir usando $u_D(t)$ como variable de rasgado, pero reservando la ecuación $u_S(t) - u_L(t) - u_D(t) - u_{R_2}(t) = 0$. Analizar qué ocurre en este caso si en la Ec.(P3.1b) se usa el parámetro $R_{on} = 0$.

[P3.3] Polea Simple

El esquema de la Figura 3.11 representa una masa suspendida mediante un resorte a través de una polea simple con momento de inercia J . El modelo matemático correspondiente puede caracterizarse por las Ecs.(P3.3a). Se pide entonces:

$$\begin{aligned}
 m \dot{v}_{y_m}(t) - F_m(t) &= 0 \\
 \dot{y}_m(t) - v_{y_m}(t) &= 0 \\
 k \Delta x_r(t) - F_r(t) &= 0 \\
 J \dot{\omega}(t) - \tau(t) &= 0 \\
 \dot{\theta}(t) - \omega(t) &= 0 \\
 r F_1(t) - r F_2(t) - \tau(t) &= 0 \\
 F_1(t) - F_r(t) &= 0 \\
 F_2(t) - m g - F_m(t) &= 0 \\
 r \theta(t) + \Delta x_r(t) &= 0 \\
 \Delta x_r(t) + y_m(t) &= 0
 \end{aligned} \tag{P3.3a}$$

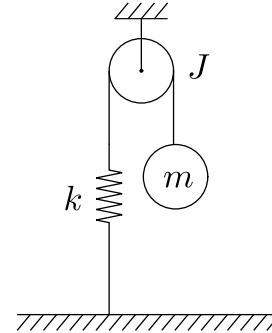


Figura 3.11: Polea Simple.

1. Repetir los tres primeros puntos del Problema P3.1 para este nuevo sistema y explicar la causa del índice alto en la DAE.
2. Aplicar el algoritmo de Pantelides hasta obtener un sistema de índice 1 y determinar el índice de perturbación de la DAE original.
3. Aplicar el procedimiento de rasgado y obtener un sistema explícito de Ecuaciones de Estado.

Capítulo 4

Técnicas Básicas de Modelado

En este capítulo trabajaremos sobre la construcción de modelos matemáticos provenientes de sistemas de distintos dominios, utilizando para esto técnicas básicas de modelado *por primeros principios*. Como ya adelantamos en el Capítulo 1, la clave estará en la descomposición del sistema en componentes simples y en el planteo de las relaciones constitutivas de los componentes y de las relaciones estructurales de las conexiones entre componentes.

Veremos entonces cuáles son los componentes típicos que constituyen los sistemas de los distintos dominios de la física y cómo se pueden plantear las relaciones constitutivas y estructurales correspondientes. Llegaremos de esta manera a plantear sistemas de DAEs que conformarán el modelo matemático buscado.

En algunos casos construiremos también Diagramas de Bloques siguiendo el procedimiento del Capítulo 3.4, lo que nos permitirá analizar ciertas características de los modelos.

4.1. Circuitos Eléctricos

Suponiendo que las frecuencias de las señales son suficientemente bajas para que las longitudes de las ondas electromagnéticas sean significativamente más largas que las dimensiones físicas de los circuitos, los modelos matemáticos pueden basarse en la Teoría de Circuitos. De otra manera, deberíamos utilizar directamente las Ecuaciones de Maxwell y recurrir a la Teoría de Líneas de Transmisión, obteniendo *ecuaciones en derivadas parciales* (modelos de parámetros distribuidos).

Haciendo entonces esta salvedad y focalizando en los modelos de parámetros concentrados de la Teoría de Circuitos, está claro que los componentes serán básicamente fuentes, resistores, inductores y capacitores con sus correspondientes relaciones constitutivas y que las relaciones estructurales estarán dadas por las Leyes de Kirchhoff.

4.1.1. Componentes Básicos

Los circuitos eléctricos utilizan como variables descriptivas básicas voltajes y corrientes, por lo que las relaciones constitutivas de los componentes vincularán dichas magnitudes. La Figura 4.1 muestra los componentes básicos.

Dipolos Pasivos

Entre los dipolos pasivos básicos encontraremos los resistores, inductores y capacitores:

Resistores: Son elementos que, desde el punto de vista energético, son puramente disipativos. Su ley constitutiva es una relación *estática* entre la corriente y el voltaje, que en el caso lineal toma la forma

$$v(t) - R i(t) = 0. \quad (4.1)$$

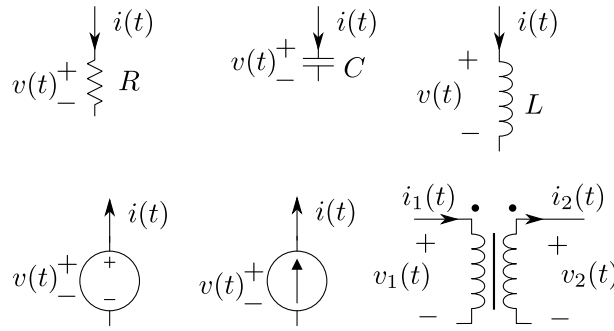


Figura 4.1: Componentes básicos de circuitos eléctricos.

En casos más generales encontraremos leyes no lineales de la forma

$$g(v(t), i(t)) = 0, \quad (4.2)$$

donde $g(\cdot)$ será una función limitada al primer y tercer cuadrante que por lo tanto cumplirá $g(0, 0) = 0$ (de otra manera no se trataría de un elemento pasivo).

Teniendo en cuenta esto, podemos clasificar dentro de esta categoría también a los diodos.

Capacitores: Son elementos que almacenan carga eléctrica. Su ley constitutiva es una relación estática entre carga y voltaje (que implica una relación dinámica entre corriente y voltaje). En el caso lineal esta relación toma la forma

$$\begin{aligned} C u(t) - q(t) &= 0 \\ \dot{q}(t) - i(t) &= 0 \end{aligned} \quad (4.3)$$

de donde, si no necesitamos incluir la carga en el modelo, obtenemos

$$C \dot{u}(t) - i(t) = 0. \quad (4.4)$$

En el caso no lineal, en tanto, tendremos

$$\begin{aligned} g(u(t), q(t)) &= 0 \\ \dot{q}(t) - i(t) &= 0 \end{aligned} \quad (4.5)$$

donde, al igual que en el caso del resistor, $g(\cdot)$ será una función limitada al primer y tercer cuadrante que cumplirá $g(0, 0) = 0$.

Inductores: Son elementos que almacenan energía en el campo magnético. Su ley constitutiva es una relación estática entre flujo magnético y corriente (que implica una relación dinámica entre corriente y voltaje). En el caso lineal esta relación toma la forma

$$\begin{aligned} L i(t) - \phi(t) &= 0 \\ \dot{\phi}(t) - u(t) &= 0 \end{aligned} \quad (4.6)$$

de donde, si no necesitamos incluir el flujo en el modelo, obtenemos

$$L \frac{di(t)}{dt} - u(t) = 0. \quad (4.7)$$

En el caso no lineal, en tanto, tendremos

$$\begin{aligned} g(i(t), \phi(t)) &= 0 \\ \dot{\phi}(t) - u(t) &= 0 \end{aligned} \quad (4.8)$$

donde, al igual que en los casos del resistor y el capacitor, $g(\cdot)$ será una función limitada al primer y tercer cuadrante que cumplirá $g(0, 0) = 0$.

Fuentes Ideales

Las fuentes ideales son dipolos activos que imponen tensiones o corrientes al resto del circuito

Fuente de Voltaje: Es un dipolo cuya tensión sigue una ley independiente de la corriente que circula en el mismo.

Fuente de Corriente: Es un dipolo cuya corriente tiene una ley independiente del voltaje en el mismo.

Transformadores

Son cuadripolos que transmiten energía entre dos partes del sistema vinculando las tensiones y corrientes.

Transformador Ideal: Un transformador ideal transmite energía entre dos subsistemas sin almacenar ni disipar y está caracterizado por las ecuaciones:

$$\begin{aligned} v_1(t) - n v_2(t) &= 0 \\ n i_1(t) - i_2(t) &= 0 \end{aligned} \quad (4.9)$$

Transformadores Reales: Los modelos más realistas de transformadores incluyen diversos fenómenos de pérdidas e inductancias. Por ejemplo, si consideramos la inductancia en el primario y secundario de un transformador, la Ec.(4.9) se convierte en

$$\begin{aligned} v_1(t) - L_1 \frac{di_1(t)}{dt} - M \frac{di_2(t)}{dt} &= 0 \\ v_2(t) - M \frac{di_1(t)}{dt} - L_2 \frac{di_2(t)}{dt} &= 0 \end{aligned} \quad (4.10)$$

4.1.2. Relaciones Estructurales

Como mencionamos, pueden utilizarse las leyes de Kirchoff como herramientas básicas para plantear las relaciones estructurales de los modelos.

Por ejemplo, para el circuito de la Figura 4.2, podemos plantear las relaciones constitutivas (4.11a)–(4.11f) y las estructurales (4.11g)–(4.11k).

$$\begin{aligned} \dot{\phi}(t) - v_L(t) &= 0 & (4.11a) \\ L i_L(t) - \phi(t) &= 0 & (4.11b) \\ v_R(t) - R i_R(t) &= 0 & (4.11c) \\ g(v_D(t), i_D(t)) &= 0 & (4.11d) \\ v_S(t) - U \sin(\omega t) &= 0 & (4.11e) \\ v_{R_2}(t) - R_2 i_{R_2}(t) &= 0 & (4.11f) \\ i_{R_2}(t) - i_L(t) - i_R(t) &= 0 & (4.11g) \\ i_S(t) - i_D(t) &= 0 & (4.11h) \\ i_D(t) - i_{R_2}(t) &= 0 & (4.11i) \\ v_S(t) - v_L(t) - v_D(t) - v_{R_2}(t) &= 0 & (4.11j) \\ v_R(t) - v_L(t) &= 0 & (4.11k) \end{aligned}$$

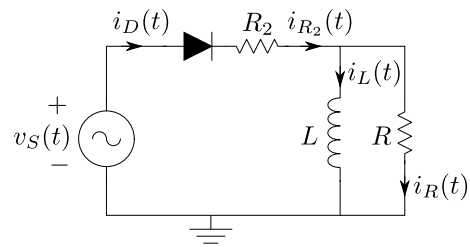


Figura 4.2: Rectificador de Media Onda con carga RL en paralelo.

Podemos notar que el sistema de DAEs resultante contiene 11 ecuaciones y 11 variables (entre estados y algebraicas). Si tuviéramos un número distinto de ecuaciones que de variables, el modelo estaría mal planteado.

Otra manera de plantear las relaciones estructurales es considerando directamente los potenciales eléctricos en los nodos. Esto permite resumir todas las relaciones estructurales mediante dos reglas:

1. Todos los puntos conectados a un nodo tienen el mismo potencial.
2. La suma de todas las corrientes entrantes a un nodo es cero.

Al trabajar con potenciales, deberemos además considerar explícitamente la presencia de la conexión a tierra. Este elemento tiene una relación constitutiva:

$$U_G(t) = 0 \tag{4.12}$$

donde $U_G(t)$ es el potencial eléctrico.

Con esto, a las relaciones constitutivas dadas por las Ecs.(4.11a)–(4.11f)y (4.12), se le agregan las siguientes ecuaciones estructurales:

$$\begin{aligned} i_S(t) - i_D(t) &= 0 & (4.13a) \\ i_D(t) - i_{R_2}(t) &= 0 & (4.13b) \\ i_{R_2}(t) - i_L(t) - i_R(t) &= 0 & (4.13c) \\ i_L(t) + i_R(t) - i_G(t) - i_S(t) &= 0 & (4.13d) \\ v_S(t) - (U_1(t) - U_4(t)) &= 0 & (4.13e) \\ v_D(t) - (U_1(t) - U_2(t)) &= 0 & (4.13f) \\ v_{R_2}(t) - (U_2(t) - U_3(t)) &= 0 & (4.13g) \\ v_L(t) - (U_3(t) - U_4(t)) &= 0 & (4.13h) \\ v_R(t) - (U_3(t) - U_4(t)) &= 0 & (4.13i) \\ U_G(t) - U_4(t) &= 0 & (4.13j) \end{aligned}$$

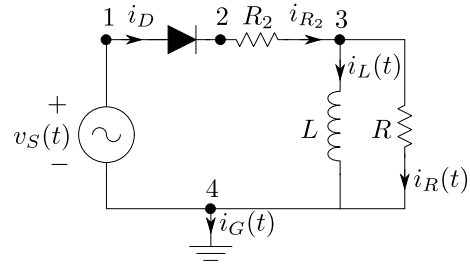


Figura 4.3: Circuito rectificador de media onda con carga RL con nodos de potencial eléctrico.

Si bien el modelo que utiliza los potenciales tiene más variables algebraicas (y más ecuaciones, por lo tanto), veremos más adelante que su uso será conveniente en el contexto de modelado orientado a objetos, ya que sus leyes estructurales son mucho más simples y no requieren de conceptos topológicos como *serie* o *paralelo*.

4.1.3. Elementos de Conmutación

En muchas aplicaciones electrónicas se utilizan elementos que ante ciertas situaciones experimentan cambios abruptos entre estados de corte o de conducción. La Figura 4.4 muestra un diodo y una llave que podría representar un transistor trabajando entre corte y saturación.

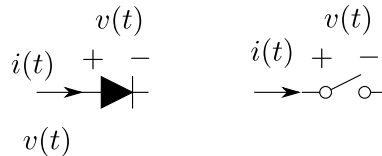


Figura 4.4: Elementos básicos de conmutación.

Si consideramos que en el estado de corte estos componentes no conducen corriente y que en estado de conducción no tienen diferencia de potencial, es probable que tengamos problemas para causalizar las DAEs y obtener un sistema de Ecuaciones de Estado donde podamos realizar análisis o simular. Para evitar estos problemas, se suele considerar que dichos elementos tienen una resistencia muy pequeña en estado de conducción y una muy grande en estado de corte.

Teniendo en cuenta esto, pueden considerarse las siguientes relaciones constitutivas básicas:

Diodo Ideal:

$$v_D(t) - R_D(v_D(t)) i_D(t) = 0 \tag{4.14a}$$

donde

$$R_D(v_D) = \begin{cases} R_{\text{on}} & \text{si } v_D > v_\gamma \\ R_{\text{off}} & \text{en otro caso} \end{cases} \quad (4.14b)$$

donde v_γ es un parámetro (un valor típico podría ser 0.6V). Este modelo se corresponde al de un resistor no lineal como el de la Ec.(4.2) donde la función $g(\cdot)$ es discontinua.

Llave Ideal:

$$v_S(t) - R_S(s(t)) i_S(t) = 0 \quad (4.15a)$$

donde

$$R_S(s) = \begin{cases} R_{\text{on}} & \text{si } s = 1 \\ R_{\text{off}} & \text{si } s = 0 \end{cases} \quad (4.15b)$$

donde $s(t)$ es una señal de entrada que puede valer 0 (llave cerrada) o 1 (llave abierta).

La llave, definida de esta manera, se comporta como una *resistencia modulada*, ya que la relación entre las variables $i_S(t)$ y $v_S(t)$ se encuentra modulada por una señal de entrada $s(t)$.

Un circuito muy utilizado en muchas aplicaciones es el convertidor *boost* o elevador, representado en La Fig.4.5, cuyas relaciones constitutivas y estructurales están dadas por la Ec.(4.16).

$$C \frac{dv_C(t)}{dt} - i_C(t) = 0 \quad (4.16a)$$

$$L \frac{di_L(t)}{dt} - v_L(t) = 0 \quad (4.16b)$$

$$R i_R(t) - v_R(t) = 0 \quad (4.16c)$$

$$R_D(v_D(t)) i_D(t) - v_D(t) = 0 \quad (4.16d)$$

$$R_S(s(t)) i_S(t) - v_S(t) = 0 \quad (4.16e)$$

$$v_F(t) - U = 0 \quad (4.16f)$$

$$i_L(t) - i_F(t) = 0 \quad (4.16g)$$

$$i_L(t) - i_S(t) - i_D(t) = 0 \quad (4.16h)$$

$$i_D(t) - i_C(t) - i_R(t) = 0 \quad (4.16i)$$

$$v_F(t) - v_L(t) - v_S(t) = 0 \quad (4.16j)$$

$$v_S(t) - v_D(t) - v_C(t) = 0 \quad (4.16k)$$

$$v_C(t) - v_R(t) = 0 \quad (4.16l)$$

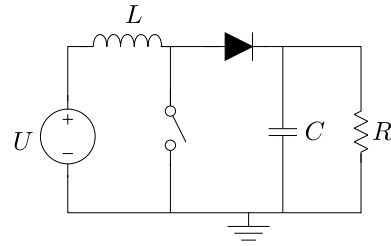


Figura 4.5: Circuito Boost o Elevador.

A partir de la DAE formada por la Ec.(4.16) se puede construir directamente un Diagrama de Bloques como el de la Figura 4.6 siguiendo el procedimiento del Capítulo 3.4. El hecho que en el DB todos los bloques tengan bien definida su entrada y salida indica que tenemos igual número de variables (señales) que de ecuaciones (bloques). El DB además muestra la presencia de un lazo algebraico entre las corrientes y voltajes de la llave y del diodo.

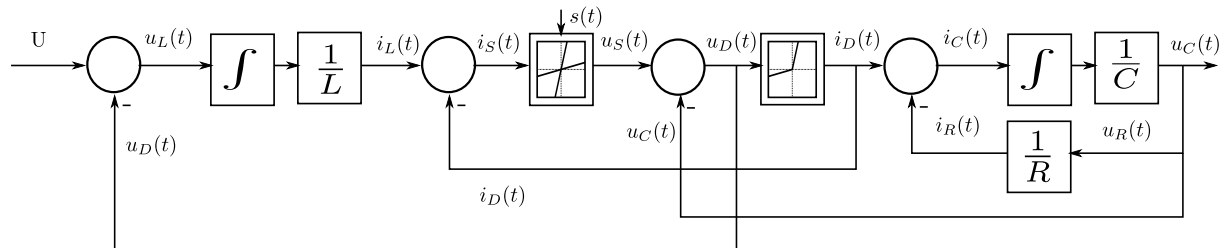


Figura 4.6: Diagrama de Bloques de la Ec.(4.16).

En el Diagrama de Bloques puede verse que si consideramos que el diodo no tiene resistencia de conducción, no podríamos determinar la corriente $i_D(t)$ a partir de la tensión del diodo $u_D(t)$, ya que en estado de conducción el voltaje es nulo para cualquier corriente. De manera similar, si la llave tuviera resistencia de corte infinita, no podríamos calcular el voltaje $u_S(t)$ a partir $i_S(t)$ cuando $s = 0$.

Si en lugar de construir el Diagrama de Bloques hubiéramos aplicado sobre la DAE directamente el procedimiento de causalización que vimos en el Capítulo 3, hubiéramos encontrado el lazo y tras aplicar el algoritmo de rasgado tendríamos asignaciones equivalentes a las que realiza el DB. Por lo tanto, tendríamos las mismas dificultades al considerar ideales los elementos de conmutación.

El modelo que utilizamos para el diodo al considerar R_{on} y R_{off} se denomina *lineal a tramos* ya que su característica Volt-Ampère está formada por dos rectas. Un modelo más realista utiliza una exponencial sin discontinuidades:

$$i_D - I_0 \left(e^{\frac{v_D}{\eta V_T}} - 1 \right) = 0 \quad (4.17)$$

para ciertos parámetros I_0 , η y V_T , lo que constituye también un resistor no lineal como el de la Ec.(4.2).

Si bien este modelo no tiene las discontinuidades de la Ec.(4.14), suele ser computacionalmente más costoso simular con una relación no lineal tan abrupta como la que genera la Ec.(4.17).

En cualquier caso, siempre debe tenerse en cuenta la pregunta a responder para elegir el modelo más adecuado.

4.1.4. Consideraciones sobre el orden

A priori, el orden del modelo de un circuito sería igual al número de componentes almacenadores (inductores y capacitores) ya que en los mismos aparece una variable derivada que se puede tomar como estado. Sin embargo, como vimos en el Capítulo 3.3, en ocasiones los supuestos estados pueden estar vinculados estáticamente por ecuaciones de restricción y en tal caso el orden se ve reducido.

Estas ecuaciones de restricción van a aparecer cuando tengamos capacitores en paralelo, inductancias en serie, o casos un poco más complejos que implican que a dos o más almacenadores no podemos asignarles condiciones iniciales independientes. Desde el punto de vista de las DAEs resultantes, ya vimos que la presencia de restricciones implica que el sistema será de *índice alto*.

En definitiva, el orden será igual al número de componentes almacenadores menos el número de ecuaciones de restricción.

4.2. Sistemas Mecánicos

Los modelos matemáticos de los sistemas mecánicos se basan principalmente en la segunda ley de Newton:

$$\sum F_i(t) = m a(t) \quad (4.18)$$

y en leyes asociadas a distintos fenómenos que habitualmente ocurren en estos sistemas (rozamiento, por ejemplo).

Veremos entonces como plantear las relaciones constitutivas y estructurales correspondientes.

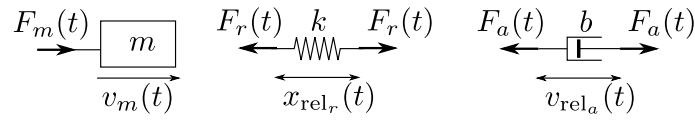
4.2.1. Modelos Mecánicos Traslacionales

Analizaremos a continuación los modelos matemáticos de sistemas mecánicos que se mueven en una dimensión, denominados sistemas mecánicos traslacionales.

Componentes y Relaciones Constitutivas

La Figura 4.7 muestra los componentes elementales de la mecánica traslacional.

Describiremos a continuación los componentes brindando sus relaciones constitutivas:


Figura 4.7: Componentes básicos de la mecánica traslacional

Masa: asociado a la masa de parámetro m podemos plantear las siguientes ecuaciones:

$$\begin{aligned} \dot{x}_m(t) - v_m(t) &= 0 \\ m \dot{v}_m(t) - F_m(t) &= 0 \end{aligned} \quad (4.19)$$

donde $x_m(t)$ es la posición, $v_m(t)$ la velocidad y $F_m(t)$ es la fuerza neta aplicada sobre la masa.

Notar que estas relaciones constitutivas valen para cualquier masa que se mueva en una dimensión.

Resorte: asociado al resorte de parámetro k podemos plantear la ecuación:

$$k x_{\text{rel}_r}(t) - F_r(t) = 0 \quad (4.20)$$

donde $x_{\text{rel}_r}(t)$ es la deformación del resorte y $F_r(t)$ es la fuerza ejercida sobre los extremos del mismo. Notar que según el esquema de fuerzas de la Figura 4.7 estamos considerando positiva la deformación cuando el resorte se estira.

La Ecuación(4.20) se corresponde a un resorte lineal. En casos más generales, un resorte tendrá asociada una relación entre deformación y fuerza del tipo:

$$g(x_{\text{rel}_r}(t), F_r(t)) = 0. \quad (4.21)$$

Amortiguador: respecto al amortiguador de parámetro b tenemos la relación constitutiva:

$$b v_{\text{rel}_a}(t) - F_a(t) = 0 \quad (4.22)$$

que se corresponde a un amortiguador lineal y donde $v_{\text{rel}_a}(t)$ es la velocidad relativa entre los extremos del amortiguador y $F_a(t)$ es la fuerza que se ejerce entre dichos extremos. En este caso, según el esquema de la Figura 4.7, estamos considerando positiva la velocidad relativa cuando el amortiguador se estira.

En casos más generales tendremos

$$g(v_{\text{rel}_a}(t), F_a(t)) = 0, \quad (4.23)$$

relación que permite modelar casos no lineales.

Un amortiguador puede modelar también fenómenos de fricción, que pueden depender de la fuerza normal (lo que convertiría al amortiguador en un componente *modulado*).

Relaciones Estructurales

Al igual que en los circuitos, en los modelos mecánicos encontraremos dos clases de relaciones estructurales. Por un lado habrá ecuaciones que vinculen posiciones y/o velocidades entre componentes (similar a lo que ocurría con los potenciales en el caso de los circuitos). Estas ecuaciones estarán asociadas a cuestiones esencialmente geométricas.

Por otro lado habrá ecuaciones que vinculen fuerzas basadas en el hecho que todas las fuerzas concurrentes en un punto sin masa deben sumar cero. Esto es de alguna manera análogo a lo que ocurría con las corrientes en los circuitos.

Veremos a continuación estas relaciones sobre un ejemplo muy simple dado por el modelo masa-resorte-amortiguador de la Figura 4.8.

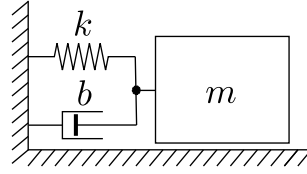


Figura 4.8: Modelo Masa-Resorte-Amortiguador

Para vincular entonces en primer lugar las posiciones y velocidades que aparecen en los distintos componentes, podemos plantear las siguientes relaciones estructurales:

$$\begin{aligned} v_m(t) - v_{\text{rel}_a}(t) &= 0 \\ x_{\text{rel}_r}(t) + L_0 - x_m(t) &= 0 \end{aligned} \quad (4.24)$$

donde L_0 es la *longitud en reposo* del resorte (no confundir con la posición inicial). De esta manera, la segunda ecuación estructural dice que cuando la posición de la masa es igual a la longitud en reposo, el mismo no está deformado.

Por otro lado, la ecuación que vincula las fuerzas es:

$$F_m(t) + F_r(t) + F_a(t) = 0 \quad (4.25)$$

Luego, juntando las relaciones constitutivas de las Ecs.(4.19), (4.20) y (4.22) y las estructurales de las Ecs.(4.24) y (4.25), obtenemos el sistema completo de DAEs:

$$\begin{aligned} \dot{x}_m(t) - v_m(t) &= 0 \\ m \dot{v}_m(t) - F_m(t) &= 0 \\ k x_{\text{rel}_r}(t) - F_r(t) &= 0 \\ b v_{\text{rel}_a}(t) - F_a(t) &= 0 \\ v_m(t) - v_{\text{rel}_a}(t) &= 0 \\ x_{\text{rel}_r}(t) + L_0 - x_m(t) &= 0 \\ F_m(t) + F_r(t) + F_a(t) &= 0 \end{aligned}$$

que tiene 7 ecuaciones en 7 variables.

El procedimiento que seguimos para plantear el modelo en este caso puede resumirse en los siguientes pasos:

1. Descomponer el sistema en componentes.
2. Plantear las relaciones constitutivas.
3. Plantear las relaciones estructurales que vinculan posiciones y/o velocidades.
4. Plantear las relaciones estructurales para las fuerzas.

Sin embargo, en el tercer paso no fuimos muy sistemáticos, ya que en un caso vinculamos posiciones (resorte y masa) y en el otro velocidades (amortiguador y masa) por lo que las relaciones estructurales no se tradujeron siempre de la misma manera, sino que las elegimos en función de las variables que tenía cada componente.

Sistematización de las Relaciones Estructurales

Lo más sistemático sería vincular siempre las posiciones ya que la relación $x_1(t) = x_2(t)$ implica $v_1(t) = v_2(t)$ pero no al revés (por las condiciones iniciales).

Esto implicaría que deberían aparecer las posiciones (o la deformación) del amortiguador, lo que implicará que las relaciones constitutivas del amortiguador deberían ser:

$$\begin{aligned} b v_{\text{rel}_a}(t) - F_a(t) &= 0 \\ \dot{x}_{\text{rel}_a}(t) - v_{\text{rel}_a}(t) &= 0 \end{aligned} \quad (4.26)$$

Procediendo entonces de esta forma más sistemática, el sistema de DAEs quedaría como sigue:

$$\begin{aligned} \dot{x}_m(t) - v_m(t) &= 0 \\ m \dot{v}_m(t) - F_m(t) &= 0 \\ k x_{\text{rel}_r}(t) - F_r(t) &= 0 \\ b v_{\text{rel}_a}(t) - F_a(t) &= 0 \\ \dot{x}_{\text{rel}_a}(t) - v_{\text{rel}_a}(t) &= 0 \\ x_m(t) - x_{\text{rel}_a}(t) &= 0 \\ x_{\text{rel}_r}(t) + L_0 - x_m(t) &= 0 \\ F_m(t) + F_r(t) + F_a(t) &= 0 \end{aligned}$$

contando con 8 ecuaciones e incógnitas.

En este nuevo modelo se puede ver inmediatamente que hay una ecuación de restricción $x_m(t) - x_{\text{rel}_a}(t) = 0$ ya que la misma vincula dos supuestos estados (y es claro además que el sistema es de orden 2 y no puede haber 3 estados). Esto es un problema de índice alto que puede reducirse directamente a índice 0 aplicando una vez el algoritmo de Pantelides.

En definitiva, el precio que pagamos por utilizar relaciones estructurales más simples que vinculan directamente las posiciones es tener problemas de índice alto que requieren algún trabajo posterior (que lo puede hacer un algoritmo de forma totalmente sistemática).

Ejemplo con Serie Mecánica y Fuente

Con esta forma de plantear las relaciones estructurales, veamos un nuevo ejemplo dado por el sistema de la Figura 4.9. El mismo puede representarse directamente mediante el sistema de DAEs de la Ec. (4.27).

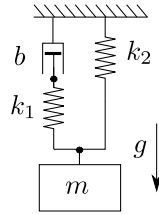


Figura 4.9: Sistema Masa-Resorte-Amortiguador con conexión serie mecánica.

$$\dot{x}_m(t) - v_m(t) = 0 \quad (4.27a)$$

$$m \dot{v}_m(t) - F_m(t) = 0 \quad (4.27b)$$

$$k_1 x_{\text{rel}_1}(t) - F_{r_1}(t) = 0 \quad (4.27c)$$

$$k_2 x_{\text{rel}_2}(t) - F_{r_2}(t) = 0 \quad (4.27d)$$

$$b v_{\text{rel}_a}(t) - F_a(t) = 0 \quad (4.27e)$$

$$\dot{x}_{\text{rel}_a}(t) - v_{\text{rel}_a}(t) = 0 \quad (4.27f)$$

$$F_g(t) - m g = 0 \quad (4.27g)$$

$$x_{\text{rel}_1}(t) + L_{0_1} + x_{\text{rel}_a}(t) + L_{0_a} - x_m(t) = 0 \quad (4.27h)$$

$$x_{\text{rel}_2}(t) + L_{0_2} - x_m(t) = 0 \quad (4.27i)$$

$$F_m(t) + F_{r_1}(t) + F_{r_2}(t) + F_g(t) = 0 \quad (4.27j)$$

$$F_{r_1}(t) + F_a(t) = 0 \quad (4.27k)$$

En este sistema de DAEs puede verse que no hay singularidades estructurales y que efectivamente se trata de un modelo de tercer orden.

Apareció además aquí un nuevo componente que representa una fuente ideal debida a la fuerza de la gravedad.

Conexiones con Posiciones Absolutas

Si bien en este último ejemplo las relaciones estructurales relacionadas con las posiciones fueron más simples de obtener, el procedimiento sigue sin ser completamente sistemático respecto a dichas relaciones ya que en las mismas mezclamos posiciones absolutas y relativas y el planteo de dichas relaciones requiere analizar la geometría del esquema más que las conexiones entre los componentes.

Una manera más clara es definir directamente las posiciones absolutas de los puntos de conexión de cada componente (lo que agregará variables y ecuaciones a las relaciones constitutivas del resorte y el amortiguador). Luego, una vez definidos esas posiciones, las ecuaciones de cada conexión consistirán simplemente en igualar las posiciones de los puntos conectados.

Siguiendo esta idea, el modelo del amortiguador debería ser

$$\begin{aligned} b v_{\text{rel}_a}(t) - F_a(t) &= 0 \\ \dot{x}_{\text{rel}_a}(t) - v_{\text{rel}_a}(t) &= 0 \\ x_{\text{rel}_a}(t) - x_{a_1}(t) + x_{a_2}(t) &= 0 \end{aligned} \quad (4.28)$$

donde $x_{a_1}(t)$ y $x_{a_2}(t)$ son las posiciones de los extremos del amortiguador. Similarmente, el modelo del resorte será

$$\begin{aligned} k x_{\text{rel}_r}(t) - F_r(t) &= 0 \\ x_{\text{rel}_r}(t) + L_0 - x_{r_1}(t) + x_{r_2}(t) &= 0 \end{aligned} \quad (4.29)$$

donde $x_{r_1}(t)$ y $x_{r_2}(t)$ son las posiciones de los extremos del resorte y L_0 es la longitud en reposo del resorte. Luego, al conectar ambas componentes la ecuación estructural que vincula las posiciones será simplemente:

$$x_{a_1}(t) - x_{r_1}(t) = 0$$

y algo similar valdrá para cualquier conexión.

Al trabajar con posiciones absolutas, tendremos además que fijar una posición de referencia (similar a lo que hicimos en los circuitos cuando trabajamos con potenciales). Esto se puede hacer con un componente que denominaremos *punto fijo* que podrá modelar una pared, piso, o en general cualquier posición fija respecto al eje de coordenadas.

Esto es análogo a lo que hicimos al utilizar potenciales absolutos en los circuitos. El precio que pagaremos por esta simplificación será que tendremos muchas variables algebraicas (con muchos alias y ecuaciones triviales). Sin embargo, esto no será un problema cuando trabajemos con lenguajes de modelado orientados a objetos como Modelica.

4.2.2. Modelos Mecánicos Rotacionales

Los modelos rotacionales siguen esencialmente las mismas leyes que los traslacionales cambiando las posiciones y velocidades por ángulos y velocidades angulares y la masa por el momento de inercia.

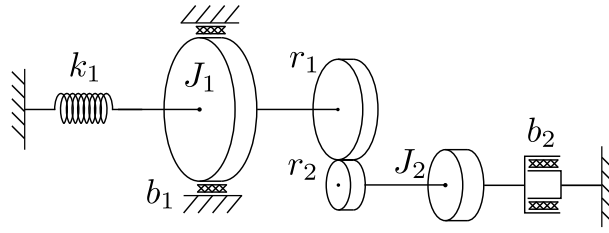
Componentes y Relaciones Constitutivas

El esquema de la Fig.4.10 por ejemplo muestra un sistema mecánico rotacional que tiene dos inercias con fricción contra un marco fijo, una caja de reducción y un resorte de torsión.

Caracterizaremos entonces los componentes con sus relaciones constitutivas:

Inercia: Análoga a la masa traslacional, puede modelarse mediante la segunda ley de Newton en variables angulares:

$$\begin{aligned} \dot{\theta}(t) &= \omega(t) \\ J \dot{\omega}(t) &= \tau_J(t) \end{aligned} \quad (4.30)$$


Figura 4.10: Modelo mecánico rotacional

donde $\theta(t)$ y $\omega(t)$ representan la posición y velocidad angular, τ_J es el torque neto aplicado y J es el parámetro que representa el momento de inercia.

Resorte de torsión: Análogo al resorte de la mecánica traslacional, se puede caracterizar por la siguiente relación constitutiva:

$$k \theta_{\text{rel},r}(t) - \tau_r(t) = 0 \quad (4.31)$$

donde $\theta_{\text{rel},r}(t)$ es la deformación angular entre los extremos del resorte y $\tau_r(t)$ es el torque que se ejerce sobre dichos extremos.

La Ec.(4.31) se corresponde a un caso lineal. En un caso general tendremos una relación estática genérica entre torque y ángulo:

$$g(\theta_{\text{rel},r}(t), \tau_r(t)) = 0 \quad (4.32)$$

Amortiguador Rotacional: Similar al amortiguador traslacional, está caracterizado por la relación constitutiva:

$$b \omega_b(t) - \tau_b(t) = 0 \quad (4.33)$$

donde $\omega_b(t)$ es la velocidad angular a la que se produce la fricción y $\tau_b(t)$ es el torque correspondiente.

En un caso no lineal, la Ec.(4.33) toma la forma

$$g(\omega_b(t), \tau_b(t)) = 0 \quad (4.34)$$

Al igual que lo que ocurría en el caso traslacional, podemos explicitar el ángulo para que las relaciones estructurales vinculen los mismos agregando la ecuación

$$\dot{\theta}_b(t) - \omega_b(t) = 0 \quad (4.35)$$

Caja Reductora: La caja reductora es un elemento de tipo *transformador* que vincula velocidades (o posiciones) y torques idealmente sin almacenar ni consumir energía. Es análoga al transformador eléctrico que vimos antes y su relación constitutiva puede plantearse como

$$\begin{aligned} \frac{r_1}{r_2} \omega_1(t) - \omega_2(t) &= 0 \\ \tau_1(t) - \frac{r_1}{r_2} \tau_2(t) &= 0 \end{aligned} \quad (4.36)$$

donde $\omega_1(t)$ y $\omega_2(t)$ son las velocidades angulares a ambos lados de la caja siendo $\tau_1(t)$ y $\tau_2(t)$ los torques correspondientes. El cociente $\frac{r_1}{r_2}$ es la relación de transformación y el hecho que sea inversa en el torque respecto de la velocidad implica que la potencia a ambos lados de la caja es idéntica.

Si decidimos utilizar posiciones para las relaciones estructurales, se puede reemplazar las velocidades angulares $\omega_i(t)$ por los ángulos $\theta_i(t)$ en la Ec.(4.36).

En cuanto a las relaciones estructurales, valen las mismas consideraciones que en la mecánica traslacional:

- Se igualan las posiciones o velocidades al conectar dos componentes.
- La suma de todos los torques conectados a un mismo punto es cero.

Con estas consideraciones, podemos plantear el siguiente sistema de DAEs para el sistema de la Figura 4.10:

$$\dot{\theta}_1(t) - \omega_1(t) = 0 \quad (4.37a)$$

$$J_1 \dot{\omega}_1(t) - \tau_{J_1}(t) = 0 \quad (4.37b)$$

$$\dot{\theta}_2(t) - \omega_2(t) = 0 \quad (4.37c)$$

$$J_2 \dot{\omega}_2(t) - \tau_{J_2}(t) = 0 \quad (4.37d)$$

$$k \theta_{\text{rel}_r}(t) - \tau_r(t) = 0 \quad (4.37e)$$

$$b_1 \omega_{b_1}(t) - \tau_{b_1}(t) = 0 \quad (4.37f)$$

$$b_2 \omega_{b_2}(t) - \tau_{b_2}(t) = 0 \quad (4.37g)$$

$$\frac{r_1}{r_2} \omega_{r_1}(t) - \omega_{r_2}(t) = 0 \quad (4.37h)$$

$$\tau_{r_1}(t) - \frac{r_1}{r_2} \tau_{r_2}(t) = 0 \quad (4.37i)$$

$$\theta_1(t) - \theta_{\text{rel}_r}(t) = 0 \quad (4.37j)$$

$$\omega_1(t) - \omega_{b_1}(t) = 0 \quad (4.37k)$$

$$\omega_1(t) - \omega_{r_1}(t) = 0 \quad (4.37l)$$

$$\omega_2(t) - \omega_{r_2}(t) = 0 \quad (4.37m)$$

$$\omega_2(t) - \omega_{b_2}(t) = 0 \quad (4.37n)$$

$$\tau_r(t) + \tau_{b_1}(t) + \tau_{r_1}(t) + \tau_{J_1}(t) = 0 \quad (4.37ñ)$$

$$\tau_{b_2}(t) + \tau_{r_2}(t) + \tau_{J_2}(t) = 0 \quad (4.37o)$$

Este sistema de DAEs tendrá una singularidad estructural ya que aparentemente hay 4 estados, pero las velocidades $\omega_1(t)$ y $\omega_2(t)$ están relacionadas por la caja reductora. Si aplicamos Pantelides, por lo tanto, podemos ver que llegaremos a un modelo de orden 3 con variables de estado $\omega_1(t)$, $\theta_1(t)$ y $\theta_2(t)$.

En rigor de verdad, el sistema es de segundo orden. La inconsistencia se debe a que ninguna relación estructural vincula $\theta_1(t)$ con $\theta_2(t)$ (sólo están vinculadas sus derivadas) y podremos asignarles a los ángulos condiciones iniciales arbitrarias.

Por este motivo, como ya señalamos en el caso traslacional, es conveniente utilizar los ángulos para las relaciones estructurales.

4.2.3. Modelos Roto-Traslacionales

Como indica su nombre, estos modelos combinan componentes traslacionales y rotacionales. Los componentes novedosos serán los que ejerzan de interfaz entre ambos dominios que serán de tipo *transformador*, ya que no almacenarán ni consumirán energía.

Piñón-Cremallera

La Fig.4.11 muestra un sistema mecánico roto-traslacional.

Además de los componentes que ya vimos, encontramos un componente que ejerce de interfaz entre ambos dominios consistente en un sistema *piñón-cremallera* que puede caracterizarse por la siguiente ley:

$$\begin{aligned} r \omega(t) - v(t) &= 0 \\ \tau(t) - r F(t) &= 0 \end{aligned} \quad (4.38)$$

y reemplazando en la Ec.(4.41) se obtiene

$$\tau(t) (x(t) - r \cos(\theta)) + F(t) (r x(t) \sin(\theta)) = 0 \quad (4.42)$$

que junto a la Ec.(4.40) constituyen las relaciones estructurales del componente.

Una diferencia importante entre el sistema piñón-cremallera y el biela-manivela es que el primero establece relaciones lineales entre las magnitudes, mientras que en el segundo las relaciones son no lineales. De esta manera, la presencia de un biela-manivela resultará en un modelo no lineal.

4.2.4. Mecánica Planar

Cuando analizamos los componentes mecánico-traslacionales, lo hicimos asumiendo que el movimiento se realizaba en una única dimensión.

En la mecánica planar asumimos que los objetos se pueden desplazar en dos dimensiones, pudiendo eventualmente rotar sobre sus ejes (ejes perpendiculares al plano de movimiento).

En este caso, las leyes serán esencialmente las mismas de la mecánica traslacional, pero para ambas dimensiones.

Componentes básicos y relaciones constitutivas

La Fig.4.13 muestra el modelo de un péndulo, que consiste en una masa puntual m y una barra rígida sin masa de longitud L .

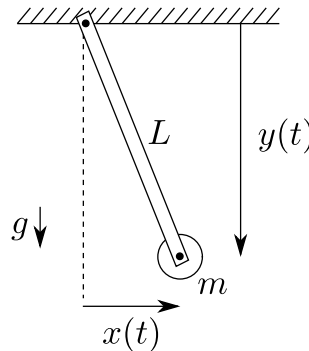


Figura 4.13: Péndulo simple

En dicho modelo podemos distinguir tres componentes:

Masa Puntual: se puede caracterizar por las siguientes relaciones constitutivas:

$$\begin{aligned} \dot{x}(t) - v_x(t) &= 0 \\ \dot{y}(t) - v_y(t) &= 0 \\ m \dot{v}_x(t) - F_x(t) &= 0 \\ m \dot{v}_y(t) - F_y(t) &= 0 \end{aligned} \quad (4.43)$$

que son una extensión directa a dos dimensiones del caso de la masa que se desplaza en una dimensión.

Fuente de Fuerza: modelando la fuerza de la gravedad, podemos caracterizarla por las relaciones constitutivas:

$$\begin{aligned} F_{x_g}(t) &= 0 \\ F_{y_g}(t) + m g &= 0 \end{aligned} \quad (4.44)$$

Barra rígida sin masa: Se puede representar mediante las siguientes ecuaciones:

$$\begin{aligned}
 L \sin(\theta) - (x_2(t) - x_1(t)) &= 0 \\
 L \cos(\theta) - (y_2(t) - y_1(t)) &= 0 \\
 F_{x_1}(t) - F(t) \sin(\theta) &= 0 \\
 F_{y_1}(t) - F(t) \cos(\theta) &= 0 \\
 F_{x_1}(t) + F_{x_2}(t) &= 0 \\
 F_{y_1}(t) + F_{y_2}(t) &= 0 \\
 \dot{\theta}(t) &= \omega
 \end{aligned} \tag{4.45}$$

donde $x_1(t)$ y $x_2(t)$ son las posiciones horizontales de cada extremo mientras que $y_1(t)$ e $y_2(t)$ son las correspondientes posiciones verticales. El ángulo $\theta(t)$ es el que forma la barra con la dirección vertical. La fuerza $F(t)$ es la que transmite la barra en sentido longitudinal, las $F_{x_1}(t)$ y $F_{x_2}(t)$ son las fuerzas que transmiten los extremos en dirección horizontal y las $F_{y_1}(t)$ y $F_{y_2}(t)$ son las que transmiten en dirección vertical.

La inclusión de la velocidad angular no es estrictamente necesaria, pero permitirá utilizar θ como variable de estado, lo que será conveniente.

Relaciones estructurales

El modelo del péndulo puede completarse con las siguientes relaciones estructurales:

$$\begin{aligned}
 x_1(t) &= 0 \\
 y_1(t) &= 0 \\
 x_2(t) - x(t) &= 0 \\
 y_2(t) - y(t) &= 0 \\
 F_x(t) + F_{x_2}(t) + F_{x_g}(t) &= 0 \\
 F_y(t) + F_{y_2}(t) + F_{y_g}(t) &= 0
 \end{aligned} \tag{4.46}$$

ya que agregándola a las Ecs.(4.43), (4.44) y (4.45) se obtiene un sistema de 19 ecuaciones e incógnitas.

Naturalmente puede plantearse un modelo mucho más simple como el de la Ec.(3.18) que utilizamos en el Capítulo 3. Sin embargo, planteando relaciones constitutivas y estructurales es posible construir modelos mucho más complejos ya que las relaciones constitutivas son *reutilizables* en cualquier contexto.

Por ejemplo, el esquema de la Figura 4.14 representa un puente grúa y se puede modelar utilizando las mismas relaciones constitutivas del péndulo, más la de la masa de la Ec.(4.19) y la del amortiguador de la Ec.(4.26).

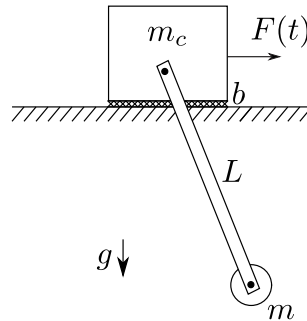


Figura 4.14: Puente grúa

Este modelo puede completarse con las relaciones estructurales

$$\begin{aligned}
 x_1(t) - x_m(t) &= 0 \\
 y_1(t) &= 0 \\
 x_2(t) - x(t) &= 0 \\
 y_2(t) - y(t) &= 0 \\
 F_x(t) + F_{x_2}(t) + F_{x_g}(t) &= 0 \\
 F_y(t) + F_{y_2}(t) + F_{y_g}(t) &= 0 \\
 x_m(t) - x_{rel_a}(t) &= 0 \\
 F_m(t) + F_{x_1}(t) + F_a(t) &= 0
 \end{aligned} \tag{4.47}$$

En el Capítulo 5, en el contexto del *modelado orientado a objetos*, veremos una forma más sistemática de reaprovechar los modelos reutilizando las relaciones constitutivas y veremos también una forma mucho más sistemática de plantear las relaciones estructurales.

En estos ejemplos de mecánica en el plano no consideramos la eventual presencia de cuerpos sólidos con momento de inercia (sólo modelamos una masa puntual). En un cuerpo con momento de inercia, además de las posiciones horizontales y verticales, aparece el ángulo de rotación respecto al centro de masa como tercera variable de posición y aparece también la transmisión de torque en las relaciones estructurales.

Estos conceptos pueden también extenderse al caso de mecánica en tres dimensiones (llamada también mecánica multicuerpos). En estos casos tendremos tres coordenadas espaciales y tres ángulos respecto al centro de masa que describirán la posición de un cuerpo en el espacio.

4.2.5. Consideraciones sobre el orden

Al igual que en cualquier sistema dinámico, el orden será igual al número de condiciones iniciales que podamos asignar arbitrariamente. Asociado a cada posición que podamos asignar independientemente en el modelo, tendríamos además la velocidad. Por esto, en general en un sistema mecánico el orden es el doble que los grados de libertad en el sistema.

Esto, sin embargo, vale siempre y cuando asociemos cada posición con una masa. En un punto de unión de dos o más elementos sin masa, sin embargo, podemos asignar el valor a una sola variable (posición o velocidad) en el mejor de los casos ya que habrá al menos una restricción entre ambas. Por ejemplo, en un serie mecánico como el del modelo de la Fig.4.9, en el punto de unión entre el resorte y el amortiguador podríamos fijar la posición inicial. Tras esto, quedará determinada la fuerza del resorte que será igual a la fuerza del amortiguador. Por lo tanto, quedará fijada la velocidad del amortiguador lo que nos quita la posibilidad de fijar arbitrariamente su condición inicial.

4.3. Sistemas Hidráulicos

Los modelos matemáticos de los sistemas hidráulicos vinculan principalmente magnitudes correspondientes a presiones $P(t)$ y caudales $Q(t)$, apareciendo también volúmenes $V(t)$ y momentos de presión $\Gamma(t)$, donde valen las relaciones:

$$\begin{aligned}
 \dot{V}(t) &= Q(t) \\
 \dot{\Gamma}(t) &= P(t)
 \end{aligned}$$

Bajo ciertas hipótesis simplificadoras, estos sistemas se pueden analizar como *circuitos hidráulicos* análogos a los circuitos eléctricos. En estos, el caudal juega el rol de la corriente, la presión es análoga al potencial eléctrico, el volumen se corresponde a la carga eléctrica y el momento de presión se corresponde con el flujo magnético.

Una hipótesis que utilizaremos en todos los modelos (salvo que se aclare lo contrario) es que trabajaremos con *presiones manométricas*, es decir, presiones medidas con respecto a la presión atmosférica.

4.3.1. Componentes Básicos y Relaciones Constitutivas

La Figura 4.15 muestra un sistema hidráulico simple, donde una bomba ideal lleva un caudal $Q_1(t)$ a un tanque que se vacía a través de una válvula R_H .

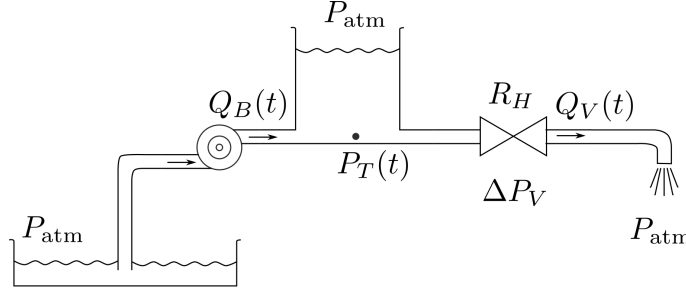


Figura 4.15: Sistema hidráulico simple

En el esquema podemos reconocer los siguientes componentes:

Válvula: Representa el fenómeno de *Resistencia Hidráulica* y si suponemos que el componente es lineal tendremos la relación constitutiva

$$\Delta P_V(t) - R_H Q_V(t) = 0. \quad (4.48)$$

donde $\Delta P_V(t)$ es la diferencia de presión entre ambos lados de la válvula, $Q_V(t)$ es el caudal y R_H un parámetro que modela la resistencia hidráulica. Notar la analogía de este componente con un resistor eléctrico.

Generalmente las válvulas hidráulicas se modelan como elementos no lineales, donde la relación entre la diferencia de presión y caudal está dada por

$$Q_V(t) - C_v \sqrt{|\Delta P_V|} = 0,$$

o, para considerar la posibilidad de caudal en ambos sentidos,

$$Q_V(t) - C_v \sqrt{|\Delta P_V|} \text{sign}(\Delta P_V) = 0. \quad (4.49)$$

Un caso más general aún está dado por

$$g(\Delta P_V(t), Q_V(t)) = 0. \quad (4.50)$$

En muchos casos también las válvulas suelen abrirse y cerrarse de manera controlada, lo que se modela por una variable de apertura $z(t)$ que puede representar la posición de un vástago. En tal caso, una relación constitutiva como la Ec.(4.49) se transforma en

$$Q_V(t) - C_v(z) \sqrt{|\Delta P_V|} \text{sign}(\Delta P_V) = 0, \quad (4.51)$$

que toma la forma que se muestra en la Figura 4.16.

En el caso general, una válvula controlada obedece a una ley del tipo

$$g(\Delta P_V, Q_V(t), z(t)) = 0. \quad (4.52)$$

Tanque: Considerando que trabajamos con presiones manométricas (lo que equivale a considerar nula la presión atmosférica), podemos calcular la presión en el fondo del tanque como

$$P_T(t) = \rho g h_T(t) \quad (4.53)$$

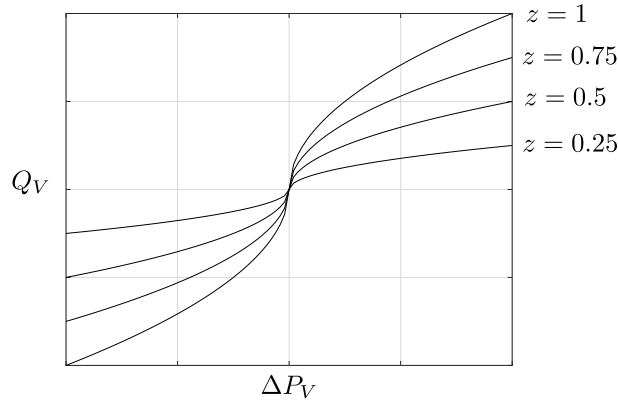


Figura 4.16: Característica de una Válvula Controlada

donde $h_T(t)$ es la altura de la columna de agua, g es la aceleración de la gravedad y ρ es la densidad del agua (del orden de 1000Kg/m^3).

La altura de la columna de agua, asumiendo que el fluido es *incompresible*, la podemos calcular como

$$h_T(t) = \frac{V_T(t)}{A} \quad (4.54)$$

donde $V_T(t)$ es el volumen del tanque y A es el área de la sección del tanque. Reemplazando $h_T(t)$ con esta expresión en la Ec.(4.53) y recordando que la derivada del volumen es el caudal neto que entra al tanque $Q_T(t)$ obtenemos :

$$\begin{aligned} \dot{V}_T(t) - Q_T(t) &= 0 \\ P_T(t) - \frac{\rho g}{A} V_T(t) &= 0 \end{aligned} \quad (4.55)$$

que conforma la relación constitutiva del tanque. Notar la analogía con un capacitor eléctrico, donde la presión cumple el rol del voltaje y el volumen cumple el rol de la carga.

En este caso el componente es lineal porque la presión depende linealmente de la altura en la Ec.(4.53) y la altura depende linealmente del volumen en la Ec.(4.54). Si se considera la compresibilidad del fluido o si el área del tanque no es constante con la altura la relación resulta no lineal.

En un caso general, por lo tanto, tendremos

$$\begin{aligned} \dot{V}_T(t) - Q_T(t) &= 0 \\ g(P_T(t), V_T(t)) &= 0. \end{aligned} \quad (4.56)$$

Bomba de Caudal: Una bomba ideal de caudal impone un caudal $Q_B(t)$ independiente de la diferencia de presión entre la entrada y la salida. De esta forma, la relación constitutiva de la bomba ideal es

$$Q_B(t) - v(t) = 0 \quad (4.57)$$

donde $v(t)$ es una función conocida.

4.3.2. Relaciones Estructurales

De manera similar a lo que ocurre en los circuitos eléctricos, podemos obtener las relaciones estructurales con las siguientes reglas:

1. Todo lo que está conectado a un mismo punto tiene la misma presión.

2. La suma de los caudales entrantes a un punto es cero.

Con esto, para el sistema hidráulico de la Figura 4.15 podemos plantear las siguientes relaciones estructurales:

$$\begin{aligned} Q_B(t) - Q_V(t) - Q_T(t) &= 0 \\ P_T - \Delta P_V &= 0 \end{aligned} \quad (4.58)$$

de manera que el sistema completo de DAEs, formado por las Ecs.(4.48), (4.55), (4.57) y (4.58), resulta

$$\begin{aligned} \Delta P_V(t) - R_H Q_V(t) &= 0 \\ \dot{V}_T(t) - Q_T(t) &= 0 \\ P_T(t) - \frac{\rho g}{A} V_T(t) &= 0 \\ Q_B(t) - v(t) &= 0 \\ Q_B(t) - Q_V(t) - Q_T(t) &= 0 \\ P_T - \Delta P_V &= 0 \end{aligned} \quad (4.59)$$

lo que constituye un sistema de 6 ecuaciones en 6 variables (5 algebraicas y una de estado).

4.3.3. Otros Componentes

La Figura 4.17 muestra otro sistema hidráulico donde, además de los componentes ya vistos, podemos encontrar los siguientes:

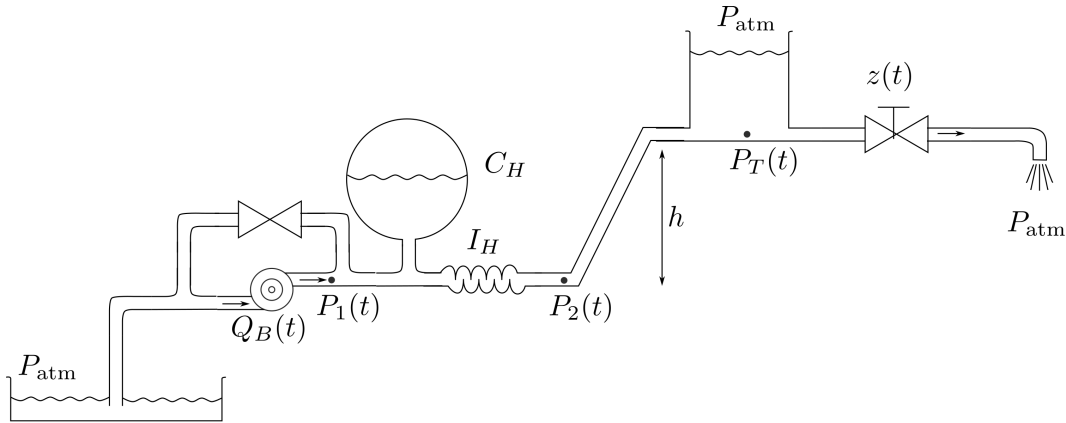


Figura 4.17: Sistema hidráulico con inercia y columna de agua.

Columna de Líquido: Representa la diferencia de presión entre dos puntos debida al peso del líquido. Asumiendo que el fluido es incompresible, la diferencia de presión entre la parte inferior y la superior de la columna está dada por

$$P_{inf}(t) - P_{sup}(t) = \rho g h \quad (4.60)$$

lo que puede verse como una relación análoga a la de una fuente de tensión constante.

Inercia: Modela la inercia de la masa de líquido. Su relación constitutiva es

$$\begin{aligned} \dot{\Gamma}_I(t) - \Delta P_I(t) &= 0 \\ \Gamma_I(t) - I_H Q_I(t) &= 0 \end{aligned} \quad (4.61)$$

donde $\Delta P_I(t)$ es la diferencia de presión a ambos lados del componente, $Q_I(t)$ es el caudal, $\Gamma_I(t)$ es el momento de presión y el parámetro I_H se denomina *inercia*. Este componente puede pensarse como análogo a un inductor eléctrico.

Acumulador Hidráulico: Modela algo similar a un tanque, excepto que el fluido se acumula aumentando la presión por la propia compresibilidad del fluido o por la de un gas que se encuentra confinado dentro del contenedor. Las relaciones constitutivas son

$$\begin{aligned} \dot{V}_C(t) - Q_C(t) &= 0 \\ C_H P_C(t) - V_C(t) &= 0 \end{aligned} \quad (4.62)$$

para el caso lineal, lo que resulta análogo a un capacitor.

4.3.4. Consideraciones sobre el Orden

Al igual que en los circuitos eléctricos, el orden de los modelos matemáticos de sistemas hidráulicos coincide en principio con el número de almacenadores de energía: tanques, inertancias y acumuladores hidráulicos. Sin embargo, al igual que en los circuitos, suele haber restricciones que reducen el orden.

4.4. Sistemas Térmicos

Los modelos matemáticos de los sistemas térmicos suelen ser de parámetros distribuidos, ya que en muchos casos no se puede ignorar la dependencia espacial de los fenómenos involucrados, en especial la *difusión* y la *convección*.

El fenómeno de conducción o difusión de temperatura en una dimensión, por ejemplo, se modela a través de la *Ecuación del Calor*:

$$\frac{\partial T(x, t)}{\partial t} = \sigma \frac{\partial^2 T(x, t)}{\partial x^2} \quad (4.63)$$

que dice que la temperatura en un punto varía en el tiempo de forma proporcional a la derivada segunda respecto al espacio (en casos de dos y tres dimensiones, la fórmula es idéntica pero utilizando el *Laplaciano* de T). La Ecuación (4.63) podría modelar, por ejemplo, la dinámica de la temperatura a lo largo de un cable.

Por otro lado, el transporte por *convección*, es decir, por el movimiento de un fluido que transporta su calor a lo largo de un camino (un caño, por ejemplo), se modela mediante la ecuación:

$$\frac{\partial T(x, t)}{\partial t} = -v \frac{\partial T(x, t)}{\partial x} \quad (4.64)$$

donde v representa la velocidad a la que se mueve el fluido (o algo proporcional a ella).

Sin embargo, aunque ambos fenómenos ocurran distribuidos en el espacio, bajo ciertas hipótesis simplificadoras se pueden concentrar en uno o más puntos. Con este enfoque de parámetros concentrados veremos a continuación cómo modelar sistemas térmicos relativamente simples caracterizados por fenómenos de difusión, convección y radiación.

En los modelos utilizaremos como variables descriptivas de las magnitudes del sistema la temperatura $T(t)$ (su unidad estará dada en grados Kelvin), el calor $E(t)$ (en Joules) y el flujo de calor $q(t)$ (en Watts, es decir, calor por unidad de tiempo).

4.4.1. Componentes y Relaciones Constitutivas Básicas

La Figura 4.18 contiene un esquema muy simple de un colector solar.

Dentro del colector circula un fluido con un caudal $Q_f(t)$ ingresando a una temperatura $T_{f_i}(t)$. El colector propiamente dicho está a una temperatura $T_c(t)$ recibiendo un flujo de calor $q_s(t)$ debido a la radiación solar. En este modelo podemos plantear los siguientes componentes asociados a los distintos fenómenos físicos:

Capacidad Térmica: Cuando concentramos parámetros podemos dividir cada objeto en uno o más *recintos* con capacidad de almacenar calor, asumiendo que en el interior de cada recinto la temperatura es uniforme. En el caso del esquema de la Figura 4.18, por ejemplo, hay delimitados dos

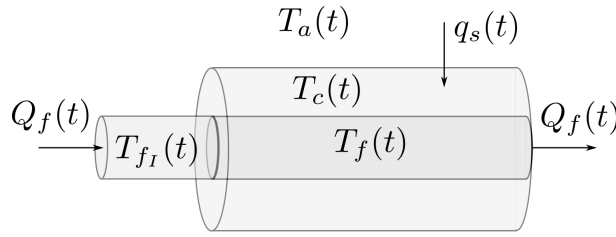


Figura 4.18: Esquema simplificado de un Colector Solar.

recintos: uno que contiene el fluido a temperatura $T_f(t)$ y el otro correspondiente al colector que se encuentra a temperatura $T_c(t)$.

Asociado a cada recinto tendremos una relación constitutiva

$$C \dot{T}(t) - q(t) = 0 \quad (4.65)$$

donde C es la *capacidad térmica*, $T(t)$ es la temperatura del recinto y $q(t)$ es el flujo neto de calor entrante al recinto. Notar la clara analogía entre la capacidad térmica y la capacidad eléctrica.

La capacidad térmica de un recinto puede calcularse a partir de la masa m del recinto y de la *capacidad calorífica específica* del material c según

$$C = m c \quad (4.66)$$

En el caso del recinto que contiene el fluido a temperatura $T_f(t)$ en la Fig.4.18, suponiendo que el fluido tiene densidad ρ_f y capacidad calorífica específica c_f , y que el recinto tiene un volumen V_f , podemos calcular la capacidad térmica según:

$$C_f = \rho_f V_f c_f \quad (4.67)$$

Cuando se considera relevante la dependencia de la capacidad térmica con la temperatura, la ley lineal de la Ec.(4.65) se torna no lineal. Además, en ocasiones se modela también la dependencia de la capacidad térmica con la presión.

Conducción: Al considerar parámetros concentrados, el fenómeno de conducción o difusión de calor se reduce a un flujo de temperatura entre recintos (o entre recintos y el exterior) proporcional a la diferencia de temperatura entre los mismos. Por analogía con el caso eléctrico, dicho componente se denomina *Resistor Térmico* y está caracterizado por la ley

$$\Delta T(t) - R q(t) = 0 \quad (4.68)$$

donde $\Delta T(t)$ es la diferencia de temperatura, $q(t)$ es el flujo de calor y R es la *Resistencia Térmica*.

En el colector solar de la Figura 4.18 pueden plantearse dos resistores térmicos: uno entre el fluido y el colector y otro entre el colector y el ambiente.

La resistencia térmica depende de la conductividad de los materiales y de la geometría de los recintos y su interfaz. Respecto a la geometría, la resistencia resulta proporcional al espesor e inversamente proporcional a la superficie. Suponiendo que la superficie es constante a lo largo del espesor del medio aislante entre dos recintos, la fórmula de la resistencia térmica resulta:

$$R_T = \frac{e}{\lambda A} \quad (4.69)$$

donde e es el espesor, A es el área y λ es la conductividad del material.

Convección: El transporte de materia entre recintos, o desde el exterior hacia los recintos o desde los recintos hacia el exterior se corresponde a un fenómeno de convección. El intercambio de energía ocurre porque dicha materia ingresa o egresa transportando su calor.

En el caso de un fluido que ingresa a un recinto con un caudal $Q(t)$ a una temperatura $T(t)$, se puede ver que el flujo de calor asociado será

$$q(t) = \rho c T(t) Q(t) \quad (4.70)$$

donde ρ es la densidad y c la capacidad calorífica específica del fluido. Esta ecuación puede deducirse a partir de la expresión de la capacidad térmica y del hecho que el caudal es la derivada del volumen.

Para el esquema del colector de la Figura 4.18, hay dos componentes de convección: uno correspondiente al caudal de fluido entrante al recinto de temperatura $T_f(t)$ y el otro correspondiente al caudal saliente de dicho recinto.

En modelos más detallados los caños se suelen dividir en secciones conformando varios recintos consecutivos, de manera que el flujo de calor saliente de un recinto es el que entra al siguiente.

Radiación: El fenómeno de radiación se corresponde a la emisión y absorción de energía radiante por parte de los átomos y moléculas de los cuerpos sólidos, líquidos y gases. La radiación está gobernada por la ley de Stephan-Boltzmann, que establece que la energía emitida por una superficie es proporcional a la cuarta potencia de la temperatura absoluta:

$$q(t) = \sigma \varepsilon A T(t)^4 \quad (4.71)$$

donde σ es la constante de Stephan-Boltzmann, ε es el coeficiente de emisión de la superficie, A es el área de la superficie y $T(t)$ es la temperatura absoluta en Kelvin.

En el caso del colector solar, podemos modelar entonces el intercambio de calor por radiación entre el ambiente y el colector mediante una ecuación

$$q_{ca}(t) = \alpha (T_c(t)^4 - T_a(t)^4) \quad (4.72)$$

para cierta constante α .

Fuente Ideal de Temperatura: En el modelo del colector, podemos considerar la temperatura ambiente $T_a(t)$ como una fuente ideal, ya que se puede asumir que su evolución es independiente de lo que ocurre en el colector. Lo mismo respecto de la temperatura de entrada del fluido $T_{f_i}(t)$. Esto sería análogo a una fuente de tensión en un circuito eléctrico.

Fuente Ideal de Flujo de Calor: El flujo de calor que se recibe del sol puede considerarse una fuente ideal, ya que el mismo es independiente de lo que ocurre en el colector.

4.4.2. Relaciones Estructurales

Las relaciones estructurales en los sistemas térmicos a parámetros concentrados se pueden pensar como análogas a las de los circuitos eléctricos, considerando la analogía entre corriente y flujo de calor y la analogía entre potencial eléctrico y temperatura.

Teniendo en cuenta esto, el modelo completo para el caso del colector estaría dado por el siguiente

sistema de DAEs con 14 ecuaciones y variables:

$$\begin{aligned}
 C_f \dot{T}_f(t) - q_f(t) &= 0 \\
 C_c \dot{T}_c(t) - q_c(t) &= 0 \\
 \Delta T_{cf}(t) - R_{cf} q_{cf}(t) &= 0 \\
 \Delta T_{ca}(t) - R_{ca} q_{ca}(t) &= 0 \\
 q_i(t) - \rho_f c_f T_{f_i}(t) Q_f(t) &= 0 \\
 q_o(t) - \rho_f c_f T_f(t) Q_f(t) &= 0 \\
 T_a(t) - v_1(t) &= 0 \\
 T_{F_i}(t) - v_2(t) &= 0 \\
 q_s(t) - v_3(t) &= 0 \\
 Q_f(t) - v_4(t) &= 0 \\
 \Delta T_{cf}(t) - (T_c(t) - T_f(t)) &= 0 \\
 \Delta T_{ca}(t) - (T_c(t) - T_a(t)) &= 0 \\
 q_f(t) - (q_{cf}(t) + q_i(t) - q_o(t)) &= 0 \\
 q_c(t) - (q_s(t) - q_{cf}(t) - q_{ca}(t)) &= 0
 \end{aligned} \tag{4.73}$$

donde las señales $v_i(t)$ son funciones conocidas del tiempo. Notar que $Q_f(t)$ proviene de una fuente ideal de caudal en este modelo, lo que evidencia que la convección es un fenómeno *multidominio*.

Discretización Espacial

Como ya mencionamos, en muchas ocasiones el problema que queremos resolver usando el modelo hace que no podamos ignorar la distribución espacial de los fenómenos térmicos. En esos casos, en principio, deberíamos recurrir a ecuaciones en derivadas parciales como la Ec.(4.63) o la Ec.(4.64).

Para simular ecuaciones en derivadas parciales, al igual que en el caso de las ordinarias, hay que discretizar. Sin embargo, no sólo hay que discretizar el tiempo sino también el espacio. Una técnica que se utiliza a menudo es la denominada *Método de Líneas* que discretiza el espacio transformando las ecuaciones en derivadas parciales en sistemas de ecuaciones diferenciales ordinarias.

Si utilizamos esta idea en un sistema térmico, el resultado será equivalente a dividir cada dominio espacial en muchos recintos donde podemos utilizar lo que vimos para sistemas de parámetros concentrados.

Consideremos por ejemplo nuevamente el sistema de la Fig.4.18 pero asumiendo ahora que la temperatura del fluido no es uniforme, sino que se va calentando a medida que circula por el colector (para simplificar supondremos que la temperatura del colector sí es uniforme). Para esto, podemos proseguir como sigue:

- Dividimos el caño en N recintos iguales con temperaturas $T_{f_1}(t), T_{f_2}(t), \dots, T_{f_N}(t)$.
- La capacidad térmica de cada recinto, recordando que es proporcional a la masa, será C_f/N .
- La resistencia térmica entre cada recinto y el colector será $N R_{cf}$ ya que la misma es inversamente proporcional a la superficie de contacto entre los recintos.
- Habrá convección entre recintos consecutivos, de manera que el flujo de calor entrante en un recinto $q_{i_j}(t)$ será el saliente del anterior $q_{o_{j-1}}(t)$.

Con estas consideraciones, las relaciones constitutivas y estructurales asociadas al j -ésimo recinto serán:

$$\begin{aligned}
 \frac{C_f}{N} \dot{T}_{f_j}(t) - q_{f_j}(t) &= 0 \\
 \Delta T_{cf_j}(t) - N R_{cf} q_{cf_j}(t) &= 0 \\
 q_{o_j}(t) - \rho_f c_f T_{f_j}(t) Q_f(t) &= 0 \\
 \Delta T_{cf_j}(t) - (T_c(t) - T_{f_j}(t)) &= 0 \\
 q_{f_j}(t) - (q_{cf_j}(t) + q_{i_j}(t) - q_{o_j}(t)) &= 0 \\
 q_{i_j}(t) - q_{o_{j-1}}(t) &= 0
 \end{aligned} \tag{4.74}$$

con la salvedad que la última ecuación sólo vale para j entre 2 y N , y para $j = 1$ tenemos directamente el flujo de convección de entrada:

$$q_{i_1}(t) - \rho_f c_f T_{f_1}(t) Q_f(t) = 0. \tag{4.75}$$

El sistema de las Ecs.(4.74)–(4.75) contiene $6N$ ecuaciones. Reemplazando estas en la Ec.(4.73) se obtiene un sistema de $6N + 8$ ecuaciones e incógnitas.

Está claro que cuanto más dividamos el dominio para obtener más precisión y más detalle en la representación espacial de la temperatura, más grande será el sistema de DAEs y el orden del modelo.

4.4.3. Consideraciones Sobre el Orden y Analogía con otros Dominios

En los modelos matemáticos a parámetros concentrados de sistemas térmicos, el orden resulta igual al número de recintos con capacidad de almacenar calor, es decir, es igual al número de *capacitores* térmicos.

Si bien en teoría podría haber restricciones que hagan que el orden resulte menor, dichas restricciones estarían asociadas a dos recintos que tienen forzada la misma temperatura. Esto implicaría una resistencia térmica nula entre ambos lo que haría que carezca de sentido considerar ambos recintos como diferentes ya que no habría ningún fenómeno que separe a los mismos.

Una singularidad de los sistemas térmicos es que carecen de fenómenos análogos a la inductancia ya que hay una única forma de almacenar energía (debido a que la variable que se almacena es justamente la energía). En todos los otros dominios encontramos dos maneras de almacenar energía: campo eléctrico y campo magnético o energía potencial y energía cinética (en sus variables traslacionales, rotacionales o en sus magnitudes hidráulicas correspondientes).

Una consecuencia de que no haya fenómenos análogos a la inductancia es que en los modelos matemáticos puramente térmicos no puede haber modos oscilantes. Es decir, el comportamiento será siempre sobreamortiguado.

4.5. Sistemas Multidominio

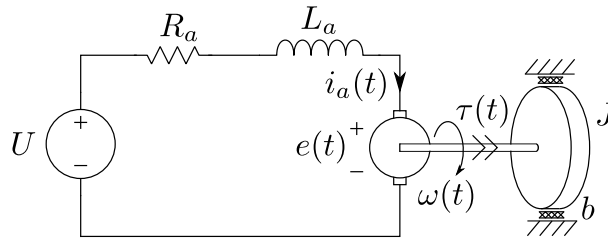
Hasta aquí vimos por separado modelos matemáticos que representan fenómenos en el dominio eléctrico, en el mecánicos, en el hidráulico y en el térmicos. En la práctica, muchos de los sistemas de interés involucran fenómenos que se corresponden a más de un dominio. Estos fenómenos multidominio resultan en nuevos componentes, con sus correspondientes relaciones constitutivas, que ejercen de *interfaz* entre sub-modelos de cada dominio. Veremos entonces a continuación algunos de estos componentes.

4.5.1. Sistemas Electro-Mecánicos

La Figura 4.19 muestra el esquema de un motor de corriente continua. El mismo tiene elementos que ya conocemos y que representan la resistencia e inductancia de armadura en el lado eléctrico y el momento de inercia y el rozamiento en el lado mecánico. Hay además un nuevo componente que representa la conversión electromagnética-mecánica de energía.

Este nuevo componente, bajo ciertas hipótesis, puede modelarse mediante las siguientes relaciones constitutivas:

$$\begin{aligned}
 e(t) - K_m \omega(t) &= 0 \\
 \tau(t) - K_m i_a(t) &= 0
 \end{aligned} \tag{4.76}$$


Figura 4.19: Motor de Corriente Continua

donde $e(t)$ es la fuerza contraelectromotriz, $i_a(t)$ es la corriente de armadura, $\omega(t)$ es la velocidad angular, $\tau(t)$ es el torque electromotriz y K_m es una constante.

Notar que este componente transmite la potencia sin que se pierda o almacene energía, ya que

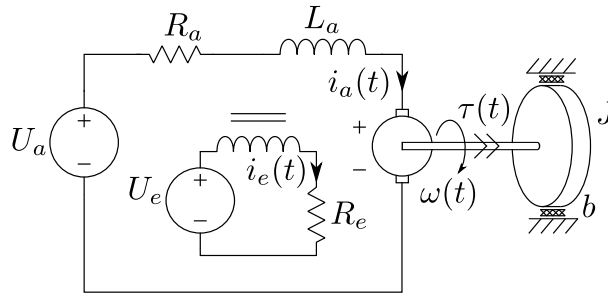
$$P_{\text{elec}}(t) = e(t) i_a(t) = \tau(t) \omega(t) = P_{\text{mec}}(t) \quad (4.77)$$

Este modelo tiene validez asumiendo que el flujo de excitación del motor es constante. Caso contrario, un modelo más detallado está dado por

$$\begin{aligned} e(t) - K \phi_e(t) \omega(t) &= 0 \\ \tau(t) - K \phi_e(t) i_a(t) &= 0 \end{aligned} \quad (4.78)$$

donde $\phi_e(t)$ es el flujo de excitación.

La Figura 4.20 muestra el modelo de un motor de corriente con excitación independiente, en el cual puede verse el circuito de excitación que contiene un inductor no lineal representando la característica magnética del arrollamiento de excitación.


Figura 4.20: Motor de Corriente Continua con Excitación Independiente

En este caso, la relación constitutiva de la Ec.(4.78) se complementa con la del inductor no lineal:

$$\begin{aligned} \dot{\phi}_e(t) - u_e(t) &= 0 \\ g(i_e(t)) - \phi_e(t) &= 0 \end{aligned} \quad (4.79)$$

donde $u_e(t)$ es la tensión en bornes del inductor, $i_e(t)$ es la corriente de excitación y $g(\cdot)$ es una función no lineal que vincula la corriente de excitación con el flujo magnético.

4.5.2. Sistemas Hidráulico-Mecánicos

La Figura 4.21 muestra un sistema que contiene componentes mecánicos e hidráulicos.

En la misma podemos distinguir dos instancias de un nuevo componente (pistón) que vincula ambos dominios y está caracterizado por las siguientes relaciones constitutivas:

$$\begin{aligned} Q(t) - A v(t) &= 0 \\ F(t) - A P(t) &= 0 \end{aligned} \quad (4.80)$$

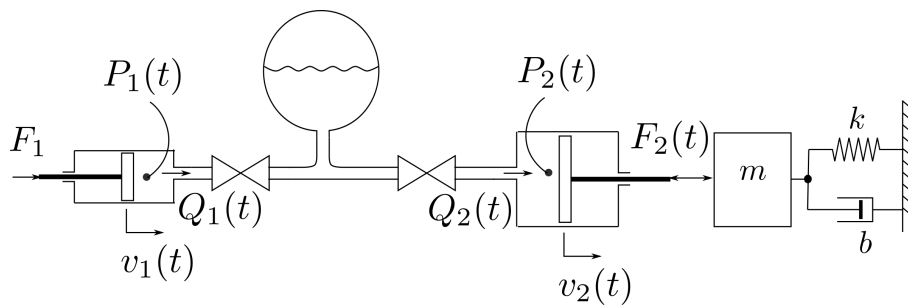


Figura 4.21: Sistema Hidráulico-Mecánico

donde $P(t)$ y $Q(t)$ son la presión y el caudal del lado hidráulico mientras que $F(t)$ y $v(t)$ son la fuerza y la velocidad del lado mecánico y el parámetro A es el área del pistón. Al igual que en los demás elementos transformadores, puede verse que la potencia es la misma a ambos lados del pistón.

Dado que en el dominio mecánico conviene utilizar las posiciones para plantear las relaciones estructurales, generalmente complementaremos la Ecuación (4.80) con la relación $\dot{x}(t) - v(t) = 0$.

En cualquier caso, las relaciones planteadas en términos de velocidad y caudal establecen una relación entre posición y volumen. Una implicancia de esto en el modelo de la Figura 4.21 es que la posición de ambos pistones y el volumen de líquido en el acumulador hidráulico están vinculados, por lo que conocido el volumen de líquido en el acumulador y la posición de uno de los pistones, quedará determinada la posición del restante (asumiendo que conocemos la cantidad total de líquido en el sistema).

4.5.3. Componentes Modulados

En los modelos electro-mecánicos e hidráulico-mecánicos encontramos componentes que ejercen de interfaz entre los distintos dominios al estilo de los transformadores. Estos elementos vinculan magnitudes a ambos lados de manera que se produce un intercambio de energía entre los dominios. De esa forma, por ejemplo, la potencia eléctrica que se le entrega a un motor se transforma en energía mecánica rotacional. En este tipo de conexión entre los dominios hay una interacción clara. Lo que hagamos de cada lado del componente de interfaz va a afectar al otro lado.

Otra manera de interacción se da cuando un componente es afectado por otro (en principio de otro dominio, pero podría ser también del mismo). Por ejemplo, en una válvula controlada la acción mecánica sobre la posición del vástago afecta su relación entre presión y caudal, pero no implica intercambio de energía entre los dominios. En otras palabras, lo que ocurra en el dominio hidráulico no *carga* a la parte mecánica. En tal caso, decimos que la válvula está *modulada* por el sistema mecánico.

De manera similar, el flujo de calor que disipa una resistencia eléctrica puede considerarse como entrada de un sistema térmico. Sin embargo, el sistema térmico no le transfiere energía al eléctrico sino que tendremos una fuente de flujo de calor modulada del lado eléctrico.

A modo de ejemplo, la Figura 4.22 muestra un sistema que contiene componentes hidráulicos, térmicos y eléctricos que modela de manera muy simplificada un sistema de calentamiento eléctrico de agua (podría ser un termotanque eléctrico).

Encontramos aquí dos tipos de componentes multidominio que pueden pensarse como fuentes de flujo de calor moduladas:

Potencia Eléctrica: La disipación de potencia de la resistencia R actúa como fuente de flujo de calor en el sistema térmico según la ley constitutiva:

$$q_R(t) - u_R(t) i_R(t) = 0 \tag{4.81}$$

donde $q_R(t)$ es el flujo de calor, y $u_R(t)$ e $i_R(t)$ son la tensión y corriente en el resistor.

Convección: Siguiendo la ley de la Ec.(4.70), tenemos dos fenómenos de convección debidos al ingreso y egreso de líquido en el recinto en los cuales el flujo de calor depende del caudal.

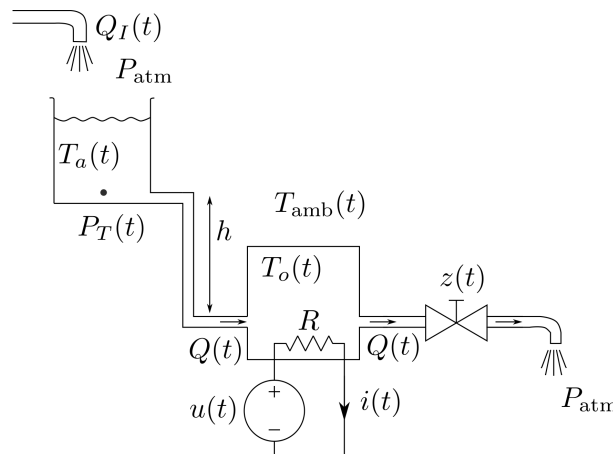


Figura 4.22: Sistema Hidráulico-Térmico-Eléctrico

Podemos considerar además que el sistema tiene algún control sobre la potencia eléctrica en función de la temperatura, de manera que la tensión $u(t)$ depende de $T_o(t)$, lo que constituye una fuente de tensión modulada. Si además incluyéramos en el modelo la parte mecánica que mueve el vástago de la válvula con posición $z(t)$, tendríamos que dicho sistema mecánico modula la válvula de salida.

4.6. Modelos no Físicos

Hasta aquí estudiamos como plantear las relaciones matemáticas correspondientes a modelos de sistemas provenientes de distintos dominios de la física. Hay otros dominios, no relacionados directamente con la física, donde se pueden plantear modelos matemáticos similares (ecuaciones diferenciales algebraicas) que permiten estudiar la evolución de las distintas magnitudes mediante simulaciones o distintas herramientas de análisis. Entre los mencionados dominios no físicos encontramos la epidemiología, la dinámica de poblaciones y la economía, entre otros. Veremos entonces como plantear modelos matemáticos (simples) en dichos dominios.

4.6.1. Modelos Epidemiológicos

Una de las ramas de la ciencia en la que se utilizan modelos de ecuaciones diferenciales es la epidemiología. Dentro de los modelos que se usan en esta disciplina encontramos por ejemplo los *Modelos Compartimentales*, en los cuales se divide una población en distintas categorías (compartimentos) y se describe la evolución del número de individuos correspondientes a cada categoría.

Uno de los modelos epidemiológicos compartimentales más simples que permite representar la propagación de una enfermedad contagiosa en una población surge de dividir dicha población en tres categorías: individuos susceptibles (S), individuos infectados (I) e individuos recuperados (R). En este modelo, denominado SIR, la población total es obviamente la suma del número individuos en cada categoría: $N(t) = S(t) + I(t) + R(t)$ y asumiendo que en el período en cuestión no varía la población total, se puede considerar que $N(t)$ es constante. En la mayor parte de los casos, se toma $N = 1$ de manera que $S(t)$, $I(t)$ y $R(t)$ representan la fracción de población susceptible, infectada y recuperada respectivamente.

Las hipótesis más simples para la construcción del modelo SIR son las siguientes:

- Sólo los individuos susceptibles pueden infectarse. Se supone que una vez recuperados, no hay posibilidad de reinfección (esto es válido generalmente al considerar escalas de tiempo no muy largas).
- El número de individuos que se infecta en cada instante de tiempo es proporcional al número de individuos infectados y al número de individuos susceptibles.

- El número de individuos que se recupera en cada instante es proporcional al número de individuos infectados.

Teniendo en cuenta esto, el modelo es muy simple:

$$\begin{aligned} \dot{S}(t) &= -\beta S(t) I(t) \\ \dot{I}(t) &= \beta S(t) I(t) - \gamma I(t) \\ \dot{R}(t) &= \gamma I(t) \end{aligned} \tag{4.82}$$

donde β y γ son dos parámetros. γ modela la inversa del tiempo medio que una persona infectada permanece como tal. El primer parámetro, en tanto, puede escribirse como $\beta \triangleq R_0 \gamma$ donde R_0 es el *número reproductivo básico* que representa cuantos individuos se exponen al agente infeccioso debido a cada persona infectada.

El parámetro γ depende exclusivamente de la enfermedad (durante cuanto tiempo en promedio las personas pueden contagiar), mientras que el parámetro R_0 depende tanto de la enfermedad (que tan contagiosa es) como del contacto social entre los individuos (a mayor contacto, mayor valor de R_0). La condición necesaria para que una epidemia se propague es que el parámetro R_0 sea mayor que 1. Además, cuanto mayor es R_0 puede verse que la epidemia se expande más rápidamente y termina alcanzando a un porcentaje más alto de la población. Esto explica los esfuerzos que se hacen para reducir el contacto social cuando aparecen epidemias que tienen consecuencias graves para la salud de la población.

En la práctica el parámetro R_0 suele ser desconocido y se debe ajustar a partir de los datos de la evolución de casos detectados. Para el Covid-19 en la ciudad de Rosario, por ejemplo, la primera ola de casos ocurrió entre los meses de Julio y Diciembre de 2020. Simulando el modelo de la Ec.(4.82) para parámetros $\gamma = 0.1177$ y $R_0 = 1.65$, escalando los resultados de acuerdo a la población de la ciudad expuesta al virus y a estimaciones de los niveles de sub-detección de casos, se obtienen los resultados de la Figura 4.23.

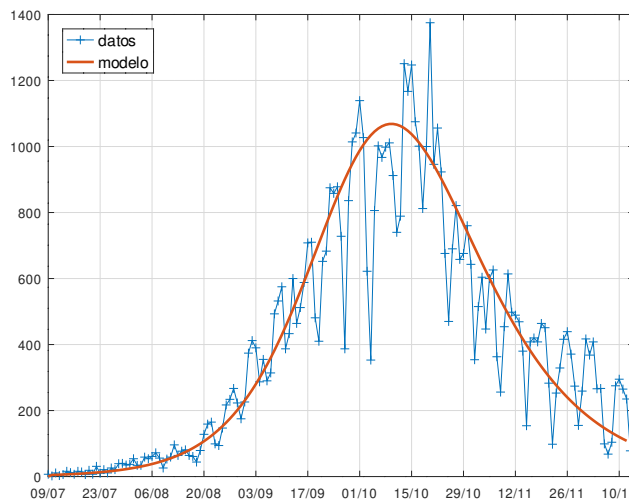


Figura 4.23: Evolución de Casos y Resultados de Simulación

Como puede verse, los resultados de simulación con el modelo se ajustan muy bien a los casos observados. De hecho, el valor de R_0 utilizado se puede ajustar muy bien utilizando los datos de las dos primeras semanas de la ola de casos y lo que sigue se puede considerar una proyección que coincide de manera muy razonable con lo que ocurrió.

Hay muchas enfermedades, el Covid-19 entre ellas, en las cuales pasan varios días desde que una persona se expone al agente patógeno hasta que comienza a contagiar. Esto se puede modelar con un

cuarto compartimento de población *expuesta* (E), lo que da lugar al modelo SEIR que tiene 4 variables de estado.

Para realizar simulaciones más largas, se puede considerar también que los individuos recuperados pueden volver a ser susceptibles tras cierto intervalo de tiempo, lo que da lugar a modelos de tipo SIRS o SEIRS.

Una limitación del modelo SIR de la Ec.(4.82) es que considera una población homogénea y cerrada que no recibe individuos infectados (casos importados). Esto último puede corregirse sumando un término de entrada en la ecuación de $\dot{I}(t)$ que modele el flujo de individuos infectados que arriban a la población. El uso de entradas permite además conectar distintos modelos SIR que modelan la evolución de la epidemia en distintas poblaciones (ciudades, provincias, etc.). En dichos casos, las conexiones entre los modelos SIR de las poblaciones representan el flujo de personas infectadas entre las mismas, que depende del número de individuos infectados en dichas poblaciones.

Con estas consideraciones, el modelo para cada población queda:

$$\begin{aligned}\dot{S}_j(t) &= -\beta_j S_j(t) I_j(t) \\ \dot{I}_j(t) &= \beta_j S_j(t) I_j(t) - \gamma I_j(t) + \sum_{i \neq j} r_{i,j} I_i \\ \dot{R}_j(t) &= \gamma I_j(t)\end{aligned}\tag{4.83}$$

donde $r_{i,j}$ es un parámetro que representa la *movilidad* de personas desde la población i hacia la población j .

4.6.2. Modelos Ecológicos y Poblacionales

Otra rama de la ciencia donde se suelen utilizar modelos de ecuaciones diferenciales es la Ecología, y más precisamente, la *Dinámica de Poblaciones*. En estos modelos, las magnitudes fundamentales son generalmente los tamaños de las poblaciones de una o distintas especies y los modelos expresan las variaciones de dichos tamaños debido a las tasas de natalidad y mortalidad y a la interacción entre las especies si hubiera más de una.

El modelo más simple que puede plantearse es el de *crecimiento logístico*, que establece que en ausencia de limitaciones de recursos y de depredadores, el número de individuos de una especie crece de manera proporcional a la propia población, dando lugar a la ecuación:

$$\dot{x}(t) = a x(t)\tag{4.84}$$

donde el parámetro a representa la tasa de crecimiento (podría interpretarse como la diferencia entre la tasa de natalidad y la de mortalidad). La solución a esta ecuación es una exponencial creciente y, bajo las hipótesis de recursos ilimitados y ausencia de predadores, suele representar bastante bien el crecimiento de una población.

Un modelo clásico muy difundido en la dinámica de poblaciones es el que desarrollaron independientemente Lotka y Volterra a mediados de la década de 1920 para explicar la evolución de la población de dos especies que compiten: presas y depredadores. Este modelo, denominado *Lotka-Volterra*, en su versión más simple asume que las presas se reproducen sin límites en ausencia de depredadores y que los depredadores mueren y se extinguen en ausencia de presas. Cuando ambas especies están presentes, hay un fenómeno de depredación que tiende a aumentar la población de depredadores y a reducir la población de presas. Este fenómeno involucra un término proporcional a la cantidad de individuos de ambas especies.

El modelo toma por lo tanto la siguiente forma:

$$\begin{aligned}\dot{x}_1(t) &= a x_1(t) - b x_1(t) x_2(t) \\ \dot{x}_2(t) &= c x_1(t) x_2(t) - d x_2(t)\end{aligned}\tag{4.85}$$

donde x_1 representa el número de presas y x_2 el número de depredadores. En cuanto a los parámetros, a es la tasa de reproducción de las presas, b es la tasa de depredación, c es la tasa de reproducción de

los predadores en presencia de presas y d es la tasa de mortalidad de los depredadores. Notar que en ausencia de depredadores, el modelo es equivalente al de crecimiento logístico dado por la Ec.(4.84).

El modelo de Lotka-Volterra permite explicar la evolución de la población de diversas especies en la naturaleza. Por ejemplo, la Figura 4.24 muestra la evolución de las poblaciones de liebres (presas) y lince (depredadores) en Canadá desde fines del siglo XIX hasta mediados del siglo XX. Estos datos se corresponden en realidad a la cantidad de pieles de liebres y lince recolectados por la *Hudson Bay Company* en Canadá. En la misma gráfica se pueden ver también los resultados de simulación del modelo de la Ec.(4.85) con parámetros $a = 0.835$, $b = 0.0311$, $c = 0.582$, y $d = 0.0156$ a partir de condiciones iniciales $x_1(0) = 19.87$, $x_2(0) = 14.82$.

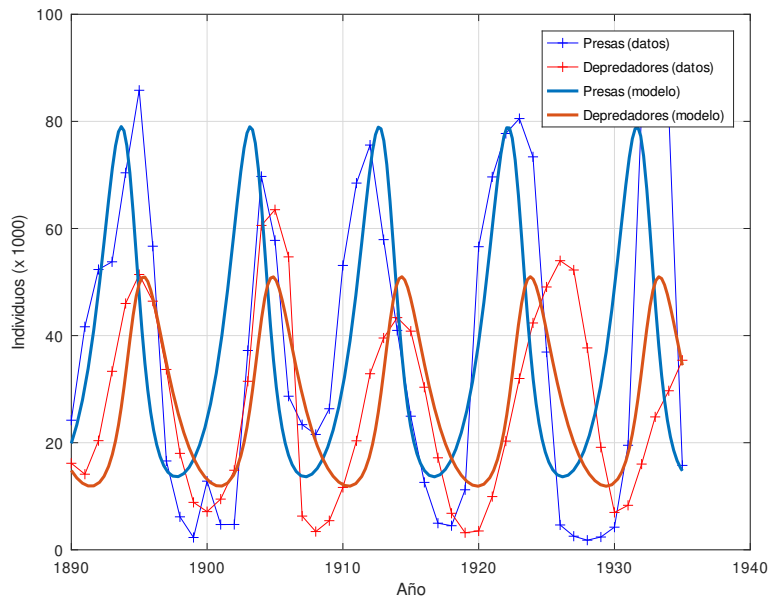


Figura 4.24: Población de Lince y Liebres.

Como puede observarse, si bien el ajuste dista de ser perfecto, el comportamiento cualitativo del modelo se corresponde razonablemente con los datos observados. Una característica del modelo de Lotka-Volterra es que las soluciones son siempre oscilantes salvo que haya una única especie inicialmente o que se comience desde el punto de equilibrio no trivial (hay un punto de equilibrio en el origen y otro para cierto número de presas y depredadores).

Los modelos poblacionales suelen además presentar las siguientes variantes:

Limitación de recursos: En el modelo de Lotka Volterra se supone que en ausencia de depredadores la población de presas crece sin límites. Para representar una limitación al crecimiento se suele agregar un término cuadrático a la primera ecuación, de manera que el sistema resulta

$$\begin{aligned} \dot{x}_1(t) &= a x_1(t) - b x_1(t) x_2(t) - e x_1^2(t) \\ \dot{x}_2(t) &= c x_1(t) x_2(t) - d x_2(t) \end{aligned}$$

y el número de presas sólo puede crecer hasta el valor $x_1^{\text{máx}} = a/e$. La misma idea se puede aplicar al modelo de crecimiento logístico de la Ec.(4.84).

Competencia y Cooperación: El modelo de la Ec.(4.85) asume que las especies se comportan como presa y depredador. Hay otros modelos donde se asume que las especies compiten por recursos, y por lo tanto se limitan mutuamente el crecimiento. Un modelo de competencia entre especies, también desarrollado independientemente por Lotka y Volterra, está dado por las siguientes ecuaciones:

$$\begin{aligned} \dot{x}_1(t) &= a x_1(t) - b x_1^2(t) - c x_1(t) x_2(t) \\ \dot{x}_2(t) &= e x_2(t) - f x_2^2(t) - g x_1(t) x_2(t) \end{aligned}$$

De manera análoga, hay modelos de cooperación donde los términos de interacción cambian de signo.

Múltiples especies: Se suelen plantear modelos de múltiples especies ejerciendo roles de presas y depredadores y/o de competidores. Por ejemplo, el siguiente modelo representa 3 especies, donde la segunda (x_2) depreda a la primera (x_1) y la tercera (x_3) depreda a la segunda:

$$\begin{aligned}\dot{x}_1(t) &= a x_1(t) - b x_1(t) x_2(t) \\ \dot{x}_2(t) &= c x_1(t) x_2(t) - d x_2(t) - e x_2(t) x_3(t) \\ \dot{x}_3(t) &= f x_2(t) x_3(t) - g x_3(t)\end{aligned}$$

Migración: Los modelos poblacionales suelen dividirse en regiones para incorporar los efectos de migración de especies. Un modelo de migración muy simple establece que la tasa de migración entre dos regiones contiguas A y B es proporcional a la diferencia de población entre ambas, con lo que un modelo presa depredador en dos regiones contiguas tomaría la siguiente forma:

$$\begin{aligned}\dot{x}_{1,A}(t) &= a x_{1,A}(t) - b x_{1,A}(t) x_{2,A}(t) + m_1 (x_{1,B}(t) - x_{1,A}(t)) \\ \dot{x}_{2,A}(t) &= c x_{1,A}(t) x_{2,A}(t) - d x_{2,A}(t) + m_2 (x_{2,B}(t) - x_{2,A}(t)) \\ \dot{x}_{1,B}(t) &= a x_{1,B}(t) - b x_{1,B}(t) x_{2,B}(t) - m_1 (x_{1,B}(t) - x_{1,A}(t)) \\ \dot{x}_{2,B}(t) &= c x_{1,B}(t) x_{2,B}(t) - d x_{2,B}(t) - m_2 (x_{2,B}(t) - x_{2,A}(t))\end{aligned}$$

Los modelos más detallados de migración consideran directamente la distribución espacial de la población y dan lugar a ecuaciones en derivadas parciales.

4.6.3. Modelos Económicos

En economía también se suelen utilizar modelos de ecuaciones diferenciales para explicar la evolución de distintas variables en el tiempo. Si bien los procesos económicos son muy complejos para explicarlos con modelos simples, en ocasiones estos modelos ayudan a visualizar propiedades cualitativas de la evolución de ciertas magnitudes. En los modelos económicos las variables generalmente representan magnitudes como capital y producción y las ecuaciones expresan cómo cambian estas magnitudes con el tiempo a partir de procesos de inversión, de negociación, etc.

Un ejemplo clásico relativamente simple es el *Modelo de Goodwin* que parte de la siguiente observación: Al aumentar la ganancia de las empresas de un país, estas invierten más dinero, lo que permite incrementar la producción. Para poder incrementar la producción, es necesario contratar más trabajadores, lo que aumenta la tasa de empleo. Ahora, cuando aumenta la tasa de empleo (es decir, cuando baja la desocupación), los trabajadores pueden exigir mejores salarios (ya que la producción los necesita). Esto hace que la ganancia de las empresas baje, con lo que baja la inversión, la producción y la tasa de empleo y al haber más desempleo las empresas puedan bajar los salarios ya que habrá desocupados dispuestos a trabajar por menos plata.

Es evidente que existe cierto punto de equilibrio entre salarios, ganancia de las empresas y tasa de empleo, y seguramente podrán imaginarse gente diciendo que basta con dejar sin regulación la economía para que este proceso alcance dicho equilibrio. Sin embargo, cualquier persona con conocimientos de dinámica de sistemas sabe que la existencia de un punto de equilibrio no implica que dicho punto sea estable. Más aún, el proceso que explicamos en el párrafo anterior se parece mucho a la descripción de un oscilador.

En el año 1967 un economista norteamericano llamado Richard Goodwin propuso un modelo matemático para representar este proceso que ya venía siendo observado desde un siglo antes en datos económicos que mostraban oscilaciones sostenidas en la tasa de empleo, en los salarios y la producción.

Veamos entonces como se construye este modelo aplicado a la economía de un país, en una versión simplificada donde consideraremos que la población (y por ende la fuerza laboral disponible) se mantiene constante y que no cambia la productividad con el tiempo.

Llamaremos $K(t)$ al capital total invertido en un país con fines productivos y llamaremos $q(t)$ a la producción que genera dicho capital. Asumiremos como primera hipótesis que la producción es proporcional

al capital invertido, esto es,

$$\frac{K(t)}{q(t)} = \sigma \quad (4.86a)$$

donde σ es un parámetro que expresa el capital que hay que invertir para producir una unidad de producto.

La segunda hipótesis que asumiremos es que la producción $q(t)$ es proporcional al número de trabajadores empleados $l(t)$ (en realidad $l(t)$ se denomina *fuerza laboral*). Esto es,

$$\frac{q(t)}{l(t)} = a \quad (4.86b)$$

donde a es un parámetro que expresa la productividad (cuanto produce en promedio cada trabajador empleado).

Llamaremos $w(t)$ al salario real promedio recibido por cada trabajador, expresado en unidades de producción (no es cuanto dinero gana un trabajador, sino cuanta producción se puede comprar con dicho salario). Luego, el salario total recibido por el conjunto de trabajadores será el producto de $w(t)$ por la fuerza laboral $l(t)$. De aquí resulta que la fracción $u(t)$ de la producción total $q(t)$ que reciben como pago los trabajadores en su conjunto es

$$u(t) = \frac{w(t) l(t)}{q(t)}. \quad (4.86c)$$

La ganancia total $p(t)$ obtenida por las empresas es igual a la producción total menos lo que se le paga a los trabajadores, esto es,

$$p(t) = q(t) - w(t) l(t) \quad (4.86d)$$

y una tercera hipótesis es que parte de la ganancia obtenida se reinvierte en forma de capital y el resto se destina a ahorro, de donde

$$\dot{K}(t) = p(t) - \lambda q(t) \quad (4.86e)$$

siendo λ un parámetro que expresa la fracción de la producción que se ahorra.

Finalmente la última hipótesis es que los trabajadores negocian su salario en función del nivel de empleo. Cuando el nivel de empleo es cercano a 1 pueden hacer que su salario crezca mucho, mientras que cuando el nivel de empleo es bajo pierden capacidad de negociación. Un modelo muy utilizado para expresar esto lo brinda la ecuación:

$$\dot{w}(t) = \left(-\gamma + \frac{\rho}{1-v(t)}\right) w(t) \quad (4.86f)$$

donde γ y ρ son dos parámetros y $v(t)$ es el nivel de empleo, que se puede calcular como

$$v(t) = \frac{l(t)}{n} \quad (4.86g)$$

donde n es la fuerza laboral disponible.

Las Ecs.(4.86a)–(4.86g) son un sistema de DAEs que representan el modelo de Goodwin. En dicho modelo se observan generalmente las variables $u(t)$ y $v(t)$, es decir, la fracción de la producción que obtienen los trabajadores y el nivel de empleo.

Puede verse que, bajo ciertas condiciones en los parámetros, hay un punto de equilibrio para ciertos valores de u y v . Sin embargo, ese punto no es asintóticamente estable y las trayectorias son oscilaciones similares a las del modelo de Lotka-Volterra.

A modo ilustrativo, la Figura 4.25 muestra resultados de simulación y datos para Argentina en el período 2004–2020. Si bien las variaciones del nivel de empleo y de salarios puede atribuirse a muchos factores (a partir de 2015 hubo un proceso de liberalización que implicó endeudamiento y desempleo) puede verse que el modelo aproxima razonablemente la evolución.

Los parámetros utilizados en la simulación de la Figura 4.25 fueron $\gamma = 0.2066$, $\rho = 0.01876$, $\sigma = 11.633$ y $\lambda = 0.4891$.

Hay muchos modelos que se utilizan en economía y la mayor parte de ellos se comportan como osciladores que explican las fases cíclicas que se observan en casi todas las economías (explosión, crisis, contracción, estancamiento, expansión y recuperación), y más fuertemente en las economías desreguladas.

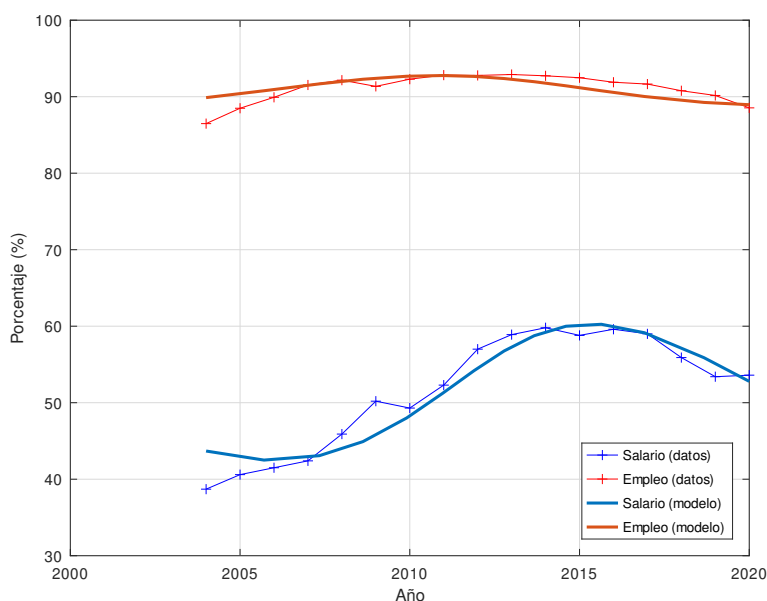


Figura 4.25: Participación de los trabajadores en el producto y nivel de empleo en Argentina (2004-2020).

A modo de ejemplo, cuando al modelo de Goodwin que tiene dos actores (capitalistas y trabajadores) se le agrega como tercer actor el sector financiero, resulta un sistema de tercer orden (modelo de Keen) que puede explicar bastante bien las crisis financieras que ocurren periódicamente tanto a nivel local de los países como a nivel global. De hecho, utilizando estos modelos varios economistas predijeron la crisis financiera mundial de 2008.

4.6.4. Parametrización de los Modelos

A diferencia de los modelos físicos donde la mayor parte de los parámetros pueden calcularse a partir de magnitudes conocidas, en los modelos no físicos los parámetros suelen ser no medibles. El nivel de contacto social en un modelo epidemiológico o el factor de depredación en un modelo poblacional son ejemplos de magnitudes que no se pueden medir de forma directa. La manera de estimar estos parámetros es observando datos reales y luego ajustando dichos parámetros para que los resultados de simular el modelo coincidan lo mejor posible con los datos reales.

Esta forma de construir los modelos se suele denominar *Modelado de Caja Gris*, ya que construimos el modelo por primeros principios y luego utilizamos técnicas de *Identificación* de sistemas (Modelado de Caja Negra) para calcular los parámetros no medibles.

4.7. Construcción Directa de Diagramas de Bloques

Hay sistemas muy simples en los cuales no es estrictamente necesario proceder de manera sistemática planteando las relaciones constitutivas y estructurales. En sistemas como el de la Figura 4.8, por ejemplo, se puede construir directamente un Diagrama de Bloques sin listar previamente todas las ecuaciones.

Para esto, alcanzará con tomar una *salida* (o cualquier variable de interés) e ir construyendo hacia atrás con bloques de manera recursiva. Una vez que el Diagrama de Bloques esté *cerrado*, es decir, una vez que cada variable del modelo esté calculada por algún bloque (y suponiendo obviamente que no utilizamos dos veces la misma ecuación) puede verse que el modelo resultante tendrá tantas variables como ecuaciones. Este procedimiento no es ni más ni menos que el que ya utilizamos en el Capítulo 3 para pasar de la DAE al DB, sólo que con un poco de práctica se puede realizar directamente sin escribir previamente la DAE.

A modo de ejemplo, dado el circuito RLC de la Figura 4.26, el procedimiento de construcción (ilustrado en la Figura 4.27) seguiría los siguientes pasos:

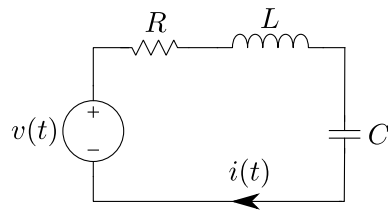


Figura 4.26: Circuito RLC Serie

1. Tomamos como salida la tensión del capacitor $u_C(t)$ y utilizamos las leyes del capacitor para calcularla a partir de la corriente $i(t)$.
2. Calculamos ahora $i(t)$ usando las leyes del inductor, a partir de la tensión $u_L(t)$.
3. Calculamos $u_L(t)$ como suma de tensiones a partir de $u_R(t)$, $u_C(t)$ y la entrada $v(t)$.
4. Dado que $u_C(t)$ ya está calculada y que $v(t)$ es entrada, resta calcular $u_R(t)$ a partir de la ley de la resistencia.

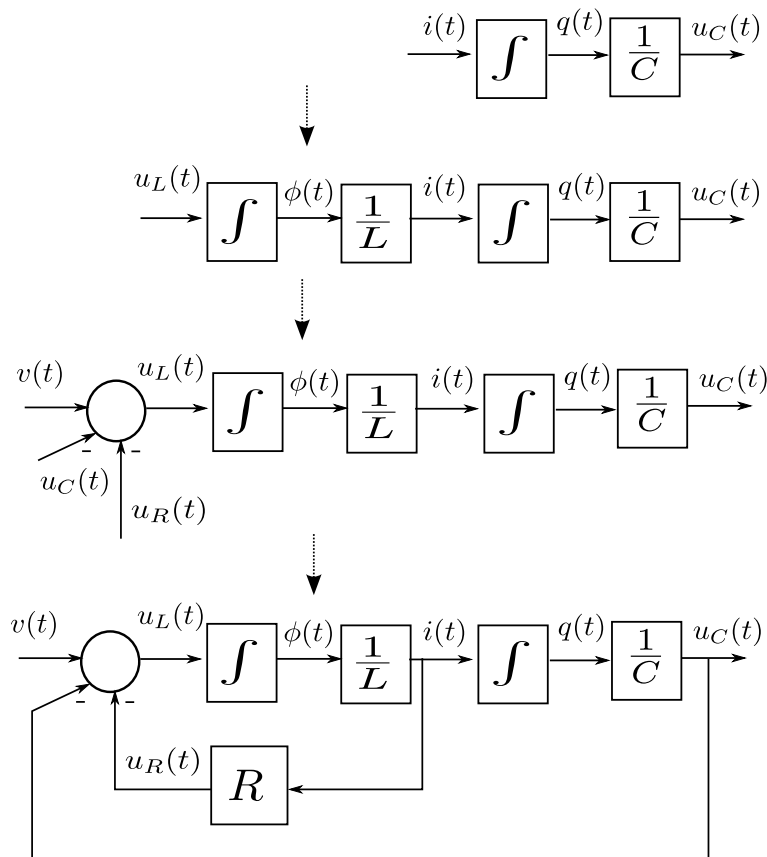


Figura 4.27: Construcción del DB a partir del circuito de la Fig.4.26.

Este procedimiento es generalmente más rápido que listar todas las relaciones constitutivas y estructurales y en sistemas muy simples permite obtener un modelo en Diagramas de Bloques de forma casi inmediata. Sin embargo, cuando los sistemas se tornan más complejos la idea generalmente ya no funciona.

4.8. Bibliografía Complementaria

En este capítulo utilizamos como fuentes principales el libro *Continuous System Modeling* de François Cellier [Cel91] y el Capítulo 14 del libro de Peter Fritzson [Fri14].

En cuanto a modelos epidemiológicos, como lectura de profundización podemos recomendar el libro de Brauer, Castillo-Chavez y Feng [BCCF19]. Respecto a los modelos ecológicos de dinámica de poblaciones como el de Lotka-Volterra, hay una gran cantidad de libros y artículos al respecto. Entre ellos, el libro [Kot01] presenta diversos modelos con sus propiedades matemáticas y el de Tiller [Til12] desarrolla un ejemplo con migraciones utilizando el lenguaje Modelica.

Con respecto a modelos económicos, un libro que tiene un enfoque desde las ecuaciones diferenciales es el de Gandolfo [Gan10], y brinda varios modelos ilustrativos (incluyendo el de Goodwin que presentamos en este capítulo).

4.9. Problemas Propuestos

[P4.1] Circuito Boost

Considerar nuevamente el circuito elevador (boost) de la Figura 4.5.

1. Escribir todas las relaciones constitutivas, incluyendo la de la puesta a tierra, y plantear las relaciones estructurales utilizando potenciales de manera análoga a lo hecho con el circuito rectificador de la Figura 4.3.
2. Listar las variables algebraicas y de estado y verificar que se correspondan el número de ecuaciones e incógnitas.
3. Suponer que no se utiliza el componente de puesta a tierra y replantear las ecuaciones correspondientes. Analizar si es posible causalizar el modelo resultante.

[P4.2] Modelo Mecánico Traslacional

Para el modelo mecánico de la Figura 4.9, se pide

1. Plantear la relación constitutiva de un componente *punto de anclaje fijo* que se corresponda a una pared, piso o techo.
2. Utilizando posiciones absolutas, reescribir las relaciones constitutivas y estructurales del sistema.
3. Eliminar las ecuaciones triviales del sistema completo de DAEs. Analizar si el número aparente de variables de estado se corresponde con el orden del modelo.
4. Causalizar el modelo obteniendo las Ecuaciones de Estado.
5. Obtener un Diagrama de Bloques directamente del sistema de DAEs.

[P4.3] Modelo Mecánico Traslacional con Dos Masas

La Figura 4.28 muestra un esquema mecánico traslacional con dos masas.

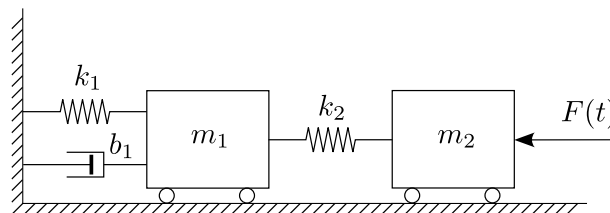


Figura 4.28: Sistema con dos Masas.

1. Analizar el orden del modelo y proponer un posible vector de estados minimal.
2. Plantear las relaciones constitutivas de los componentes de manera que en las mismas aparezcan las posiciones absolutas. Incluir el elemento correspondiente al punto de anclaje fijo.
3. Completar el sistema de DAEs con las relaciones estructurales utilizando posiciones absolutas.
4. Analizar si hay correspondencia entre el orden analizado en el primer punto y el número de variables que aparecen derivadas.
5. Eliminar ecuaciones triviales y alias y causalizar el modelo.
6. Obtener un Diagrama de Bloques.

[P4.4] Modelo Mecánico Traslacional con Palanca

El esquema de la Figura 4.29 representa un sistema mecánico con una palanca. En el mismo, se asume que el ángulo que forma la palanca con la horizontal es pequeño, de manera que puede considerarse fija la posición horizontal de los extremos de la misma.

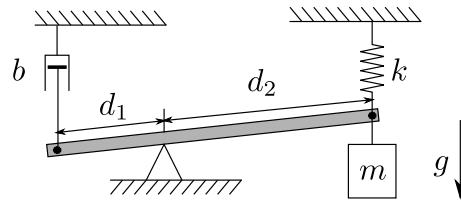


Figura 4.29: Sistema con Palanca.

1. Plantear las relaciones constitutivas de la palanca vinculando las siguientes variables:
 - Posición vertical del primer extremo: $y_1(t)$.
 - Posición vertical del segundo extremo: $y_2(t)$.
 - Posición vertical del pivot: $y_p(t)$.
 - Fuerza ejercida sobre la palanca en el primer extremo: $F_1(t)$.
 - Fuerza ejercida sobre la palanca en el segundo extremo: $F_2(t)$.
2. Plantear las restantes relaciones constitutivas y las relaciones estructurales correspondientes.
3. Analizar el orden del modelo y comparar con el número de estados que aparecen derivados.
4. Construir un Diagrama de Bloques a partir del sistema de DAEs.

[P4.5] Modelo Mecánico Rotacional

Considerar el modelo rotacional de la Figura 4.10.

1. Eliminar las ecuaciones triviales y los alias del modelo dado por la DAE (4.37) y obtener el Diagrama de Bloques correspondiente.
2. Reducir el índice mediante Pantelides derivando la restricción correspondiente a $\omega_1(t) = \omega_2(t)$ y obtener el nuevo Diagrama de Bloques.
3. Replantear el modelo de la DAE (4.37) reemplazando la restricción entre las velocidades angulares por una entre los ángulos y obtener el Diagrama de Bloques correspondiente.

[P4.6] Modelo Roto-Traslacional

Dado el sistema de la Figura 4.11,

1. Obtener un sistema de Ecuaciones Diferenciales Algebraicas que lo represente.
2. Determinar el orden del modelo.
3. Analizar si el sistema de DAEs tendrá índice alto.
4. Obtener un Diagrama de Bloques y analizar la eventual presencia de derivadores o lazos algebraicos.

[P4.7] Sistema Biela-Manivela

Dado el sistema de la Figura 4.12,

1. Obtener un modelo en Ecuaciones Diferenciales Algebraicas y determinar el orden del modelo.
2. Modificar el modelo anterior suponiendo ahora que la manivela tiene un momento de inercia J .
3. Explicar por qué el sistema de DAEs resultantes tendrá índice alto.
4. Eliminar las ecuaciones triviales y aplicar el método de reducción de índice analizando las ventajas y desventajas en las posibilidades de elección de las variables de estado.

[P4.8] Puente Grúa

Considerar nuevamente el Puente Grúa de la Figura 4.14.

1. Analizar el orden del modelo matemático y proponer un juego de variables de estado que permita caracterizar correctamente el sistema en cualquier posición.
2. Suponer ahora que el cable que sostiene la carga es elástico, de manera que la fuerza $F(t)$ que transmite es proporcional a la deformación $\Delta L(t)$ de la misma.
 - a) Modificar las relaciones constitutivas de la barra rígida dadas por la Ec.(4.45) para considerar este fenómeno.
 - b) Analizar el orden del modelo matemático resultante para el puente grúa con el cable elástico.

[P4.9] Doble Péndulo

La Figura 4.30 muestra el esquema de un doble péndulo.

1. Analizar el orden del modelo y proponer un vector de estados minimal.
2. Obtener un modelo de DAEs aprovechando las relaciones constitutivas utilizadas para el modelo del péndulo simple de la Figura 4.13 y planteando las ecuaciones estructurales correspondientes.

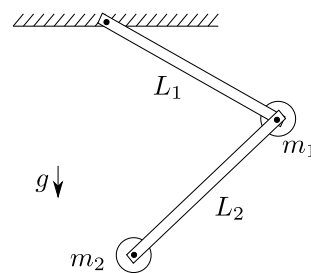


Figura 4.30: Péndulo doble

[P4.10] Sistema Hidráulico

Considerar nuevamente el sistema de la Figura 4.17.

1. Escribir un sistema completo de DAEs, incluyendo las relaciones constitutivas y estructurales del modelo.

2. Verificar que haya correspondencia entre el número de ecuaciones y de variables.
3. Listar las variables algebraicas y las variables de estado.

[P4.11] Sistema Térmico

Considerar nuevamente el esquema de la Figura 4.22.

1. Escribir un sistema de DAEs para la parte térmica del modelo, asumiendo que la temperatura $T_a(t)$ y las variables que provienen de los demás dominios (caudales, corrientes, tensiones) son entradas.
2. Causalizar el sistema.
3. Obtener un Diagrama de Bloques.

[P4.12] Intercambiador de Calor

El esquema de la Figura 4.31 representa un *intercambiador de calor* donde un líquido a se mueve por un caño interior intercambiando calor con un líquido b que se mueve por el caño exterior en dirección contraria.

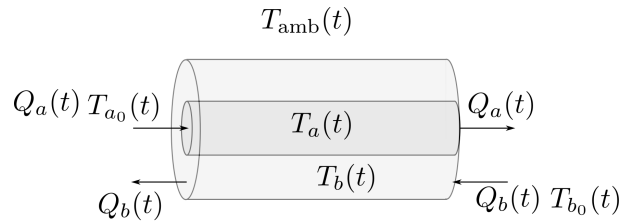


Figura 4.31: Intercambiador de Calor

Asumiendo que la temperatura de cada líquido es uniforme en el interior de cada caño, y suponiendo que tanto las temperaturas de ingreso $T_{a_0}(t)$ y $T_{b_0}(t)$ como los caudales $Q_a(t)$, $Q_b(t)$ son entradas conocidas, se pide:

1. Obtener las relaciones constitutivas asociadas a los fenómenos relevantes del sistema:
 - Almacenamiento de calor en ambos recintos. Suponer que la capacidad específica de los líquidos a y b es c_a y c_b , respectivamente, que su densidad es ρ_a y ρ_b y que el volumen total de cada caño es V_a y V_b .
 - Convección (ingreso y egreso de líquido).
 - Conducción entre ambos líquidos, suponiendo una resistencia térmica R_{ab} .
 - Conducción desde el líquido b hacia el ambiente, suponiendo una resistencia térmica R_{amb} .
2. Completar el sistema de ecuaciones diferenciales algebraicas utilizando las relaciones estructurales. Verificar que el número de variables y ecuaciones se correspondan.
3. Analizar el orden del modelo matemático.

[P4.13] Intercambiador de Calor Distribuido

Considerar nuevamente el Problema P4.12 pero suponiendo ahora que la temperatura no es uniforme en el interior de cada caño. Asumir en cambio que cada caño está discretizado espacialmente en N recintos en los cuales la temperatura sí es uniforme.

1. Recalcular los valores de resistencia y capacidad en función de las nuevas dimensiones.
2. Plantear de manera genérica las relaciones constitutivas asociadas al i -ésimo recinto de cada caño.
3. Plantear las relaciones estructurales. Tener en cuenta que al circular en sentido contrario los líquidos, la conducción se da entre la sección i del caño a y la $N - i + 1$ del caño b .

4. Verificar que el sistema de DAEs tenga igual número de ecuaciones que de variables y determinar el orden del modelo.

[P4.14] Sistemas Electromecánico El esquema de la Figura 4.32 representa un motor de corriente continua cuyo circuito de excitación está alimentado a través de un divisor resistivo para debilitar el campo resultante.

1. Analizar el orden del modelo matemático y proponer un vector de estados minimal.
2. Plantear las relaciones constitutivas de todos los componentes del modelo. Suponer que el inductor no lineal tiene una ley $g(i_e) - \phi_e = 0$.
3. Completar el sistema de DAEs con las relaciones estructurales.
4. Obtener un Diagrama de Bloques del modelo a partir del sistema de DAEs y determinar si hay lazos algebraicos o singularidades estructurales.

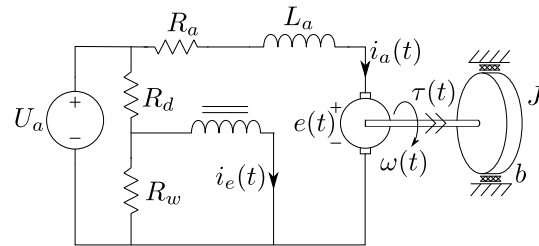


Figura 4.32: Sistema Electro-Mecánico: Motor con Debilitamiento de Campo

[P4.15] Sistema Hidráulico-Mecánico

Considerar nuevamente el esquema de la Figura 4.21.

1. Analizar el orden del modelo y proponer un vector de estados minimal.
2. Obtener un modelo de DAEs en base a las relaciones constitutivas y estructurales.
3. Realizar un Diagrama de Bloques y analizar la eventual presencia de lazos algebraicos y/o singularidades estructurales.

[P4.16] Bomba Aspirante-Impelente

El sistema de la Figura 4.33 representa parte de una bomba aspirante-impelente que llena un tanque, consistente en un pistón que avanza y retrocede conectado en el cilindro a dos válvulas que sólo permiten la circulación de fluido en un sentido.

1. Plantear las relaciones constitutivas de todos los componentes del sistema y las correspondientes relaciones estructurales. Para el caso de las válvulas utilizar relaciones constitutivas análogas a las de los diodos.
2. Proponer un juego de parámetros para todos los elementos que tengan sentido físico y simular el sistema utilizando OpenModelica.

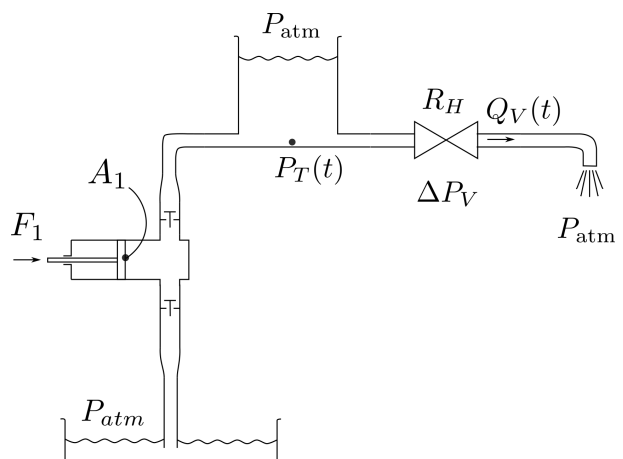


Figura 4.33: Bomba aspirante-impelente

[P4.17] **Plataforma Hidráulica**

El sistema de la Figura 4.34 representa una plataforma hidráulica, que permite elevar una carga de masa m .

1. Plantear las relaciones constitutivas de todos los componentes del sistema y las correspondientes relaciones estructurales. Tener en cuenta que como la parte superior del tanque no está a la presión atmosférica, sino que sostiene la plataforma (lo que se corresponde a un pistón), sus relaciones constitutivas serán

$$\begin{aligned} \dot{V}_T(t) - Q_T(t) &= 0 \\ P_{\text{rel}}(t) - \frac{\rho g}{A} V_T(t) &= 0 \end{aligned} \quad (\text{P4.17a})$$

donde $P_{\text{rel}}(t)$ es la diferencia de presión entre la parte inferior y la superior del tanque y $Q_T(t)$ sigue siendo el caudal neto en el tanque.

2. Verificar que el número de variables algebraicas y estados (o supuestos estados) se corres-

ponde con el número de ecuaciones en el sistema de DAEs resultante.

3. Analizar el orden del modelo y compararlo con el número de supuestos estados.

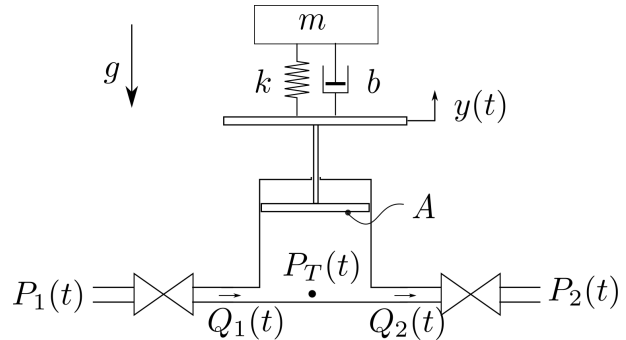


Figura 4.34: Plataforma Hidráulica

[P4.18] **Termotanque**

Considerar nuevamente el sistema de la Figura 4.22. Obtener un sistema de DAEs para el sistema completo asumiendo que las variables $Q_I(t)$, $u(t)$, $T_a(t)$, $z(t)$ y $T_{\text{amb}}(t)$ son entradas conocidas.

Capítulo 5

Modelado Orientado a Objetos

En el Capítulo 4 estudiamos como obtener modelos matemáticos de sistemas de diferentes dominios (eléctricos, mecánicos, hidráulicos y térmicos). En todos los casos, dichos modelos eran un conjunto de relaciones constitutivas y estructurales que formaban un sistema de ecuaciones diferenciales algebraicas.

Al haber componentes repetidos en un mismo modelo o entre diferentes modelos, encontrábamos obviamente las mismas ecuaciones constitutivas en los mismos excepto por algún eventual cambio de valor de los parámetros. Por otro lado, en las relaciones estructurales, siempre había variables que se igualaban (potenciales eléctricos, posiciones, presiones o temperaturas, según el dominio) y otras variables cuya suma daba cero (corrientes, velocidades, caudales o flujos de calor).

La presencia de leyes y ecuaciones repetidas en todos los modelos lleva naturalmente a las preguntas: ¿es realmente necesario plantear todas las relaciones constitutivas cada vez que agregamos un componente, o alcanza con decir cuál es el componente agregado?, ¿es necesario colocar todas las relaciones estructurales o alcanza con decir cómo están conectados los componentes?.

La respuesta casi obvia es que no es realmente necesario brindar todas las ecuaciones. Si podemos definir clara y unívocamente las relaciones constitutivas ligadas a cada componente y las relaciones estructurales correspondientes a cada conjunto de conexiones, entonces bastará con listar los componentes y sus conexiones para que quede determinado de manera única el sistema de ecuaciones diferenciales algebraicas.

Esta idea es la base del modelado orientado a objetos en general y del lenguaje Modelica en particular. En este capítulo, basado en parte en el libro de Peter Fritzson [Fri14], veremos entonces las herramientas que nos brinda el lenguaje Modelica para describir modelos matemáticos utilizando los principios del modelado orientado a objetos.

5.1. Modelica y Modelos Orientados a Objetos

Comenzaremos con un ejemplo, consistente en el circuito de la Figura 5.1 que contiene dos resistores, un capacitor y un inductor. Agregando como variables los potenciales de los conectores positivos y negativos de cada componente, podemos plantear las relaciones constitutivas de la Ec.(5.1)

$$\begin{aligned}
 C \frac{dv_C}{dt}(t) - i_C(t) &= 0 & (5.1a) \\
 V_{C_p}(t) - V_{C_n}(t) - v_C(t) &= 0 & (5.1b) \\
 L \frac{di_L}{dt}(t) - v_L(t) &= 0 & (5.1c) \\
 V_{L_p}(t) - V_{L_n}(t) - v_L(t) &= 0 & (5.1d) \\
 R i_R(t) - v_R(t) &= 0 & (5.1e) \\
 V_{R_p}(t) - V_{R_n}(t) - v_R(t) &= 0 & (5.1f) \\
 R_2 i_{R_2}(t) - v_{R_2}(t) &= 0 & (5.1g) \\
 V_{R_{2_p}}(t) - V_{R_{2_n}}(t) - v_{R_2}(t) &= 0 & (5.1h) \\
 V_{G_p}(t) &= 0 & (5.1i)
 \end{aligned}$$

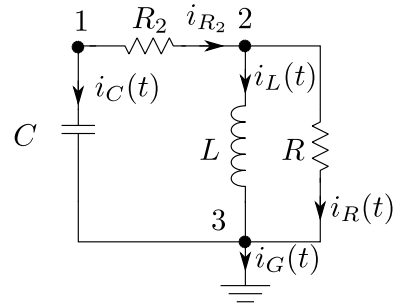


Figura 5.1: Circuito RLC

Para completar el modelo podemos plantear además las siguientes relaciones estructurales:

$$\begin{aligned}
 V_{C_p}(t) - V_{R_{2_p}}(t) &= 0 & (5.2a) \\
 -i_C(t) - i_{R_2}(t) &= 0 & (5.2b) \\
 V_{R_{2_n}}(t) - V_{L_p}(t) &= 0 & (5.2c) \\
 V_{R_{2_n}}(t) - V_{R_p}(t) &= 0 & (5.2d) \\
 i_{R_2}(t) - i_L(t) - i_R(t) &= 0 & (5.2e) \\
 V_{C_n}(t) - V_{G_p}(t) &= 0 & (5.2f) \\
 V_{R_n}(t) - V_{G_p}(t) &= 0 & (5.2g) \\
 V_{L_n}(t) - V_{G_p}(t) &= 0 & (5.2h) \\
 i_R(t) + i_L(t) + i_C(t) - i_G(t) &= 0 & (5.2i)
 \end{aligned}$$

lo que completa un sistema con 18 ecuaciones y variables. Este modelo, puede directamente implementarse en el lenguaje Modelica con el siguiente código:

Código 5.1: Modelo de las Ecs.(5.1)–(5.2) en Modelica

```

model RLC
  Real vC(start=1), VCp, VCn, iC;
  Real vL, VLp, VLn, iL;
  Real vR, VRp, VRn, iR;
  Real vR2, VR2p, VR2n, iR2;
  Real VGp, iG;
  parameter Real C=1, L=1, R=1, R2=2;
equation
  //capacitor
  C * der(vC) - iC = 0;
  VCp - VCn - vC = 0;
  //inductor
  L * der(iL) - vL = 0;
  VLp - VLn - vL = 0;
  //resistor
  R * iR - vR = 0;
  VRp - VRn - vR = 0;
  //resistor2
  R2 * iR2 - vR2 = 0;
  VR2p - VR2n - vR2 = 0;
  // ground
  VGp = 0;
  // structure: Node 1
    
```



```

VCp - VR2p = 0;
-iC - iR2 = 0;
// structure: Node 2
VR2n - VLp = 0;
VR2n - VRp = 0;
iR2 - iL - iR = 0;
// structure: Node 3
VCn - VGp = 0;
VRn - VGp = 0;
VLn - VGp = 0;
iR + iL + iC - iG = 0;
end RLC;

```

Veremos a continuación estrategias para representar el mismo modelo de manera más simple y estructurada aprovechando las distintas funcionalidades que provee un lenguaje de modelado orientado a objetos.

5.1.1. Modelado por Componentes

Una de las primeras observaciones en el modelo del Código 5.1 es que las ecuaciones del resistor aparecen repetidas. Además, el código es en sí muy largo, con todas las declaraciones y ecuaciones en un mismo nivel jerárquico. Esto sería equivalente a escribir un programa largo en una única sección de código plano.

Lo primero que haremos entonces para simplificar y estructurar el modelo es dividirlo en distintos objetos pertenecientes a distintas clases. Para esto, crearemos en primer lugar modelos elementales correspondientes a cada componente:

Código 5.2: Clases con los Modelos de Componentes Elementales

```

model Capacitor
  Real v, i, Vp, Vn;
  parameter Real C=1;
equation
  C * der(v) - i = 0;
  Vp - Vn - v = 0;
end Capacitor;

model Inductor
  Real v, i, Vp, Vn;
  parameter Real L=1;
equation
  L * der(i) - v = 0;
  Vp - Vn - v = 0;
end Inductor;

model Resistor
  Real v, i, Vp, Vn;
  parameter Real R=1;
equation
  R * i - v = 0;
  Vp - Vn - v = 0;
end Resistor;

model Ground
  Real i, Vp;
equation
  Vp = 0;
end Ground;

```

Cada modelo constituye una clase. Es decir, tenemos una clase `Capacitor`, una clase `Inductor`, una clase `Resistor` y una clase `Ground`. Cada clase tiene como *atributos* ciertos parámetros (C , L , R) y ciertas variables (representando corrientes, potenciales y diferencias de potencial). Estas variables y parámetros son en realidad objetos de la clase `Real` y tienen a su vez sus atributos (valor inicial, por ejemplo).

Podemos entonces construir el modelo del circuito RLC instanciando objetos de las clases `Capacitor`, `Inductor`, `Resistor` y `Ground`, y luego añadiendo las relaciones estructurales como sigue:

Código 5.3: Modelo del Circuito RLC Utilizando Clases Elementales

```

model RLC2
  Capacitor cap(v.start=1);
  Inductor ind;
  Resistor res, res2(R=2);
  Ground gr;
equation
  //Node 1
  cap.Vp - res2.Vp = 0;
  -cap.i - res2.i = 0;
  //Node 2
  res2.Vn-ind.Vp = 0;
  res2.Vn - res.Vp = 0;
  res2.i - ind.i - res.i = 0;
  //Node 3
  cap.Vn - gr.Vp = 0;
  res.Vn - gr.Vp = 0;
  ind.Vn - gr.Vp = 0;
  res.i + ind.i + cap.i - gr.i = 0;
end RLC2;

```

La primera sentencia del modelo es `Capacitor cap(v.start=1)`, que crea el objeto `cap` perteneciente a la clase `Capacitor`. Además, al instanciar este objeto, le modifica uno de sus atributos (el valor inicial de v). De manera similar, la sentencia `Resistor res, res2(R=2)` crea dos objetos de la clase `Resistor`, llamados `res` y `res2`. El segundo de ellos tiene además modificado su atributo R .

En la sección de ecuaciones, en tanto, encontramos expresiones como `cap.Vp - res2.Vp = 0`. Puede verse que la forma de acceder a los atributos de cada objeto es mediante la expresión `objeto.atributo`. Dado que los atributos son objetos, pueden también utilizarse expresiones como `cap.v.start` para acceder al atributo `start` del objeto `v` en el objeto `cap`.

El modelo RLC2 del Código 5.3 es equivalente al modelo RLC del Código 5.1, pero brinda una descripción mucho más concisa y clara del modelo. En la práctica, para poder utilizar el modelo del Código 5.3 necesitaremos que todas las clases estén definidas dentro de un mismo *paquete* de Modelica. Esto permitirá al compilador (OpenModelica, por ejemplo) encontrar la definición de cada clase utilizada en el modelo en cuestión. Para esto existe la clase especializada `package` que funciona como contenedor de otras clases (incluidos otros paquetes).

Una parte de un paquete con componentes de circuitos elementales y el ejemplo del circuito RLC podría ser el siguiente:

Código 5.4: Parte de un Paquete con Modelos Elementales de Circuitos

```

package Circuits1

  model Capacitor
    Real v, i, Vp, Vn;
    parameter Real C=1;
  equation
    C * der(v) - i = 0;
  ...

package Examples

```

```

model RLC2
  Capacitor cap(v.start=1);
  Inductor ind;
  Resistor res, res2(R=2);
  Ground gr;
equation
  //Node 1
  cap.Vp - res2.Vp = 0;
  -cap.i - res2.i = 0;
  //Node 2
  res2.Vn-ind.Vp = 0;
  res2.Vn - res.Vp = 0;
  res2.i - ind.i - res.i = 0;
  //Node 3
  cap.Vn - gr.Vp = 0;
  res.Vn - gr.Vp = 0;
  ind.Vn - gr.Vp = 0;
  res.i + ind.i + cap.i - gr.i = 0;
end RLC2;

end Examples;

end Circuits1;

```

Si bien esta forma de utilizar componentes nos ahorra de escribir cada vez las relaciones constitutivas, el modelo planteado de esta manera tiene todavía el problema de que las relaciones estructurales están listadas *a mano*.

5.1.2. Conectores y Conexiones

Como mencionamos antes, las relaciones estructurales pueden tener dos formas:

1. Igualación de variables (para magnitudes como potenciales, posiciones, etc.).
2. Suma de variables igual a cero (para magnitudes como corrientes, fuerzas, etc.)

En el contexto del lenguaje Modelica, las variables que se suman se denominan variables de *flujo*.

Para especificar conexiones aprovechando estas características el lenguaje Modelica provee la clase especializada `connector`, que permite describir qué variables se vinculan en una conexión y si las mismas son o no son variables de flujo.

Por ejemplo, para el caso de los circuitos eléctricos, podemos utilizar el siguiente conector:

Código 5.5: Definición de un Conector Eléctrico en Modelica

```

connector Pin
  Real v;
  flow Real i;
end Pin;

```

El Código 5.5 define la clase `Pin` que tiene dos atributos: el potencial `v` y la corriente `i` que es una variable de flujo. Luego, podemos utilizar objetos de esta clase para crear el modelo de un capacitor por ejemplo:

Código 5.6: Modelo de un Capacitor Utilizando Conectores

```

model Capacitor
  Pin p, n;
  Real v, i;
  parameter Real C=1;
equation
  C * der(v) - i = 0;

```

```

v = p.v - n.v;
p.i = i;
n.i + p.i = 0;
end Capacitor;

```

Notar que al definir este componente, estamos tomando como positiva la corriente que entra al conector llamado *p* (que representa obviamente el positivo del capacitor), considerando positivas las corrientes entrantes a los conectores.

De manera similar podemos definir los restantes componentes que habíamos definido en el Código 5.2 como sigue:

Código 5.7: Clases con los Modelos de Componentes Elementales usando Conectores

```

model Inductor
  Pin p, n;
  Real v, i;
  parameter Real L=1;
equation
  L * der(i) - v = 0;
  v = p.v - n.v;
  p.i = i;
  n.i + p.i = 0;
end Inductor;

model Resistor
  Pin p, n;
  Real v, i;
  parameter Real R=1;
equation
  R * i - v = 0;
  v = p.v - n.v;
  p.i = i;
  n.i + p.i = 0;
end Resistor;

model Ground
  Pin p;
equation
  p.v = 0;
end Ground;

```

Utilizando estos componentes, podemos entonces de manera muy simple armar el modelo equivalente al del circuito RLC de la Figura 5.1 como sigue:

Código 5.8: Modelo del Circuito RLC Utilizando Clases Elementales

```

model RLC3
  Capacitor cap(v.start=1);
  Inductor ind;
  Resistor res, res2(R=2);
  Ground gr;
equation
  connect(cap.p,res2.p);
  connect(res2.n,ind.p);
  connect(res2.n,res.p);
  connect(cap.n,gr.p);
  connect(ind.n,gr.p);
  connect(res.n,gr.p);
end RLC3;

```

El Código 5.8 refleja de manera casi gráfica el circuito donde cada ecuación define una conexión. Al declarar `connect(cap.p,res2.p)` estamos diciendo que los conectores positivos del capacitor `cap` y del resistor `res2` están conectados. Esto implicará dos cosas:

1. que hay una ecuación implícita $cap.p.v=res2.p.v$.
2. que las variables de flujo `cap.p.i` y `res2.p.i` participarán de una suma igual a cero junto a las variables de flujo de los demás conectores que estén conectados al mismo punto.

En este caso, como no hay otro conector conectado a dicho nodo, resultará directamente $cap.p.i+res2.p.i=0$.

De esta manera, podemos definir directamente las relaciones estructurales brindando las conexiones, sin necesidad de enumerar ninguna ecuación de manera explícita.

Puede verse que el modelo dado por el Código 5.8 es equivalente al del Código 5.3. Sin embargo, el modelo contendrá más variables y ecuaciones triviales dado que al definir los conectores, tuvimos que incorporar en los modelos de los componentes las ecuaciones para las corrientes por dichos conectores.

5.1.3. Herencia

Observando los componentes de los Códigos 5.6 y 5.7 es evidente que hay muchos atributos y ecuaciones repetidos en cada componente. A excepción del modelo de la tierra, el resto de los componentes tienen exactamente los mismos atributos (salvo el parámetro) y las mismas ecuaciones (salvo la primera, que define que se trata de un capacitor, inductor o resistor).

Estas coincidencias se deben a que son en realidad dipolos pasivos. Cualquier dipolo pasivo, de hecho, tendrá dos conectores, una diferencia de potencial entre ambos y una corriente definida como positiva cuando ingresa por el conector positivo.

Estas coincidencias pueden explotarse utilizando el concepto de *herencia* en los lenguajes orientados a objetos. Para esto, podemos definir una clase correspondiente al modelo de un dipolo pasivo¹ como sigue:

Código 5.9: Modelo de un Dipolo Pasivo

```
partial model OnePort
  Pin p, n;
  Real v, i;
equation
  v = p.v - n.v;
  p.i = i;
  n.i + p.i = 0;
end OnePort;
```

Notar que utilizamos el modificador `partial` para definir la clase. Esto implica que la clase no se puede instanciar. Sólo puede utilizarse para *heredar* sus atributos y ecuaciones. Aprovechando entonces esta característica de los lenguajes orientados a objetos como Modelica, podemos reformular de manera más simple las clases de los Códigos 5.6 y 5.7 como se muestra a continuación:

Código 5.10: Clases con los Modelos de Dipolos Pasivos usando Herencia

```
model Capacitor
  extends OnePort;
  parameter Real C=1;
equation
  C * der(v) - i = 0;
end Capacitor;

model Inductor
  extends OnePort;
```

¹En inglés los dipolos se denominan *one-port devices*, entendiéndose que un puerto tiene asociados dos conectores.

```

parameter Real L=1;
equation
  L * der(i) - v = 0;
end Inductor;

model Resistor
  extends OnePort;
  parameter Real R=1;
equation
  R * i - v = 0;
end Resistor;

```

El comando `extends` permite heredar todos los atributos y ecuaciones de una clase parcial al crear una nueva clase. Esto facilita la creación de nuevos componentes aprovechando las características comunes de los mismos.

5.1.4. Modelado Jerárquico

La Figura 5.2 muestra el circuito equivalente de una celda solar que está conectada a un circuito elevador tipo *Boost* como el que vimos en el Capítulo 4 (página 70). El esquema a su vez alimenta una resistencia de carga R .

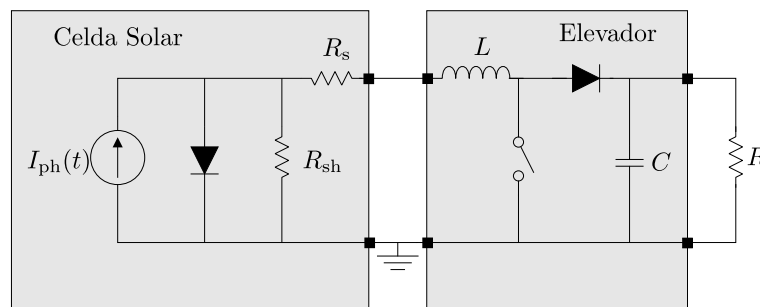


Figura 5.2: Circuito equivalente de una celda solar con un convertidor *Boost*.

Para modelar este circuito podemos simplemente definir los modelos de la fuente de corriente, de los diodos y de la llave (los restantes componentes ya los conocemos) y luego conectar todo como hicimos en el Código 5.8 para el circuito de la Figura 5.1.

Los modelos de los nuevos componentes serán:

Código 5.11: Modelos de Componentes Eléctricos Adicionales usando Herencia

```

model Diode
  extends OnePort;
  parameter Real Ron=1e-5,Roff=1e5,Vknee=0.6;
equation
  i= if v>Vknee then (v-Vknee)/Ron+Vknee/Roff else v/Roff;
end Diode;

model Switch
  extends OnePort;
  parameter Real Ron=1e-5,Roff=1e5;
  discrete Real s;
equation
  v= if s>0.5 then i*Ron else i*Roff;
end Switch;

model ConstCurr
  extends OnePort;

```

```

parameter Real I=1;
equation
  i= -I;
end ConstCurr;

```

de donde el modelo completo nos quedará:

Código 5.12: Modelo de una Celda Solar con circuito elevador

```

model PanelBoost
  Capacitor cap(C=1e-4);
  Inductor ind(L=1e-4);
  Ground gr;
  Switch sw;
  Diode di;
  Resistor rs(R=0.35),rsh(R=200),rl(R=10);
  Diode dio;
  ConstCurr cc(I=2);
  parameter Real dc=0.75;
  parameter Real T=1e-4;
equation
  connect(cc.p,dio.p);
  connect(cc.p,rsh.p);
  connect(cc.p,rs.p);
  connect(cc.n,dio.n);
  connect(cc.n,rsh.n);
  connect(cc.n,gr.p);
  connect(rs.n,ind.p);
  connect(ind.n,sw.p);
  connect(ind.n,di.p);
  connect(di.n,cap.p);
  connect(cap.p,rl.p);
  connect(cap.n,rl.n);
  connect(cap.n,sw.n);
  connect(cap.n,gr.p);

algorithm
  when sample(0,T) then
    sw.s:=1;
  end when;
  when sample(dc*T,T) then
    sw.s:=0;
  end when;
end PanelBoost;

```

Si bien este modelo funcionará, el mismo tiene muchos componentes en un mismo nivel *jerárquico*. Esto en general complica la construcción, la visualización y la verificación de los modelos. Aunque no es tan grande aún el tamaño del modelo y es manejable en la forma dada por el Código 5.12, es preferible dividir el modelo en submodelos más simples.

Para esto, podemos construir dos subsistemas, uno correspondiente al modelo de una celda solar y el otro al del circuito Boost, tal cual aparecen en la Figura 5.2. Estos subsistemas pueden definirse según el siguiente Código de Modelica:

Código 5.13: Subsistemas Celda Solar y Circuito Boost

```

model SolarCell
  Pin p,n;
  Resistor rs(R=0.35),rsh(R=200);
  Diode dio;
  ConstCurr cc(I=2);

```

```

equation
  connect(cc.p,dio.p);
  connect(cc.p,rsh.p);
  connect(cc.p,rs.p);
  connect(rs.n,p);
  connect(cc.n,dio.n);
  connect(cc.n,rsh.n);
  connect(cc.n,n);
end SolarCell;

model Boost
  Pin p1,p2,n;
  Capacitor cap(C=1e-4);
  Inductor ind(L=1e-4);
  Switch sw;
  Diode di;
  parameter Real dc=0.5;
  parameter Real T=1e-4;
equation
  connect(p1,ind.p);
  connect(ind.n,sw.p);
  connect(ind.n,di.p);
  connect(di.n,cap.p);
  connect(cap.p,p2);
  connect(cap.n,n);
  connect(cap.n,sw.n);
algorithm
  when sample(0,T) then
    sw.s:=1;
  end when;
  when sample(dc*T,T) then
    sw.s:=0;
  end when;
end Boost;

```

Conectando el modelo de la celda solar a un resistor de carga y a una tierra, podemos probar su funcionamiento por separado. De manera similar, conectando el modelo del Boost a una fuente de tensión (entre el pin p1 y el pin n), a un resistor de carga (entre el pin p2 y el pin n) y a una tierra, podemos también simular su funcionamiento independiente. Esto facilita enormemente la depuración de errores en los modelos.

Usando estas nuevas clases, entonces, podemos construir de manera mucho más simple el modelo del panel con el circuito elevador:

Código 5.14: Modelo de una Celda Solar con circuito elevador usando subsistemas

```

model PanelBoost
  SolarCell cell;
  Boost boost;
  Ground gr;
  Resistor rl(R=10);
equation
  connect(cell.p,boost.p1);
  connect(boost.p2,rl.p);
  connect(rl.n,boost.n);
  connect(boost.n,cell.n);
  connect(boost.n,gr.p);
end PanelBoost;

```

Este modelo, en el cual utilizamos el concepto de *modelado jerárquico*, es mucho más sencillo. Además,

las nuevas clases `SolarCell` y `Boost` pueden reutilizarse en otros modelos o para crear modelos complejos con varias instancias de los mismos.

Por ejemplo, podemos formar un conjunto de celdas consistente en N filas, donde cada fila tiene M paneles conectados en paralelo y las filas están conectadas en serie entre sí (así es como se utilizan habitualmente las celdas para construir paneles solares). Este circuito, representado en la Figura 5.3, se puede representar por el Código 5.15.

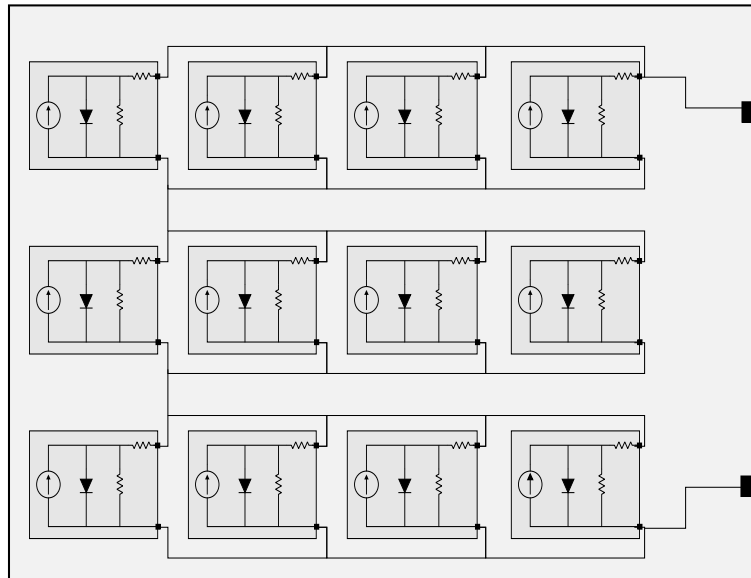


Figura 5.3: Esquema de un Panel Solar

Código 5.15: Panel Solar

```

model PanelArray
  parameter Integer N=3,M=4;
  SolarCell[N,M] cell;
  Pin p,n;
equation
  for i in 1:N loop
    for j in 1:M-1 loop
      connect(cell[i,j].p,cell[i,j+1].p);
      connect(cell[i,j].n,cell[i,j+1].n);
    end for;
  end for;
  for i in 1:N-1 loop
    connect(cell[i,1].n,cell[i+1,1].p);
  end for;
  connect(cell[1,1].p,p);
  connect(cell[N,1].n,n);
end PanelArray;

```

Luego, podemos utilizar esta clase en lugar de la celda solar en el modelo con el circuito Boost y la carga resistiva:

Código 5.16: Modelo de un Panel Solar con circuito elevador usando subsistemas

```

model PanelArrayBoost
  PanelArray panel(N=4,M=3);
  Boost boost;
  Ground gr;

```

```

Resistor r1(R=10);
equation
  connect(panel.p,boost.p1);
  connect(boost.p2,r1.p);
  connect(r1.n,boost.n);
  connect(boost.n,panel.n);
  connect(boost.n,gr.p);
end PanelArrayBoost;

```

Este modelo, pese a lo simple de su descripción, cuenta con 379 variables y ecuaciones (de las cuales 251 son triviales). Cambiando solamente los valores de N y M se pueden formar de manera elemental sistemas mucho más grandes. Si no hubiéramos recurrido al modelado jerárquico, hubiera sido casi imposible representar este último sistema.

5.2. Construcción de Librerías de Distintos Dominios

Los conceptos de modelado orientado a objetos que vimos hasta aquí permiten construir de manera simple componentes, conectarlos de forma elemental para armar modelos y reutilizar luego tanto los componentes como los modelos. Los distintos elementos que se van utilizando en el proceso (conectores, modelos parciales, modelos, etc.) se pueden organizar para facilitar su uso posterior en *librerías* (usaremos este término por constumbre, aunque la palabra más correcta sería *bibliotecas*).

Utilizando además herramientas de edición gráfica de Modelica (tanto OpenModelica como Wolfram SystemModeler y Dymola cuentan con las mismas) es posible brindar atributos gráficos a los componentes. Estos atributos gráficos se agregan como *anotaciones* a las distintas clases que no tienen ningún efecto sobre el modelo en sí, pero que indican como debe visualizarse cada componente. Las anotaciones están además estandarizadas en Modelica, lo que permite que los modelos construidos usando distintas herramientas no sólo funcionen en las otras herramientas, sino que además se vean igual en su capa gráfica.

Veremos entonces como se pueden construir y utilizar librerías elementales de los distintos dominios, teniendo en cuenta no sólo los aspectos referentes a la definición matemática de los modelos sino también a su visualización.

5.2.1. Circuitos Eléctricos

En la sección anterior ya introdujimos las principales ideas de cómo hacer modelos orientados a objetos de circuitos eléctricos. Vamos a extender ahora estas nociones aprovechando algunas características adicionales del lenguaje Modelica y de las herramientas de edición gráfica.

Tipos de Datos y Unidades

Modelica permite asignar unidades a las magnitudes que representan las variables. Si bien las unidades funcionan generalmente como un *adorno* (ya que en general no se utilizan para realizar operaciones) las mismas ayudan a visualizar mejor los modelos y los resultados de simulación.

Para utilizar unidades, la clase `Real` tiene un atributo denominado `unit` cuyo valor es una cadena de caracteres. Podríamos entonces en principio instanciar una variable real que representa un voltaje diciendo que su unidad está dada en voltios con un comando del estilo `Real v(unit="V");`.

Otra alternativa es crear una nueva clase derivada de la clase `Real` que ya tenga como unidad el voltio. Esto puede hacerse mediante el comando `type` como se muestra en el siguiente código, que además define una clase para las corrientes:

Código 5.17: Clases con unidades para circuitos eléctricos

```

type Voltage = Real(unit="V");
type Current = Real(unit="A");

```

Utilizando estas clases, podemos redefinir entonces el conector eléctrico Pin que ya habíamos utilizado en el Código 5.5 de manera que quede más claro :

Código 5.18: Definición de un Conector Eléctrico en Modelica

```
connector Pin
  Voltage v;
  flow Current i;
end Pin;
```

Vistas de ícono

Podemos además agregarle una vista gráfica al conector editando en la capa de íconos. Si ponemos por ejemplo un cuadrado relleno de color azul la herramienta modifica automáticamente el código anterior agregando una anotación:

Código 5.19: Definición de un Conector Eléctrico en Modelica con Anotación Gráfica

```
connector Pin
  Voltage v;
  flow Current i;
  annotation(Icon(graphics = {Rectangle(fillColor = {0, 0, 255}, fillPattern =
    FillPattern.Solid, extent = {{-80, 80}, {80, -80}})}));
end Pin;
```

Utilizando ahora este conector, podemos crear el modelo Ground de una puesta a tierra, como el que definimos en el Código 5.7. Si queremos que en el mismo se vea el conector con su capa gráfica, en lugar de escribir en la vista de texto la línea Pin p, deberemos arrastrar el conector a la vista de diagrama. Luego podemos escribir en la sección equation (en la vista de texto) la ecuación del componente $p.v=0$ y finalmente podemos dibujar en la vista de ícono el dibujo correspondiente a la puesta a tierra.

Dibujando por ejemplo el ícono de la Figura 5.4 (el dibujo del conector aparece automáticamente en el ícono al arrastrarlo inicialmente), la herramienta automáticamente agrega las anotaciones que aparecen en el Código 5.20. El texto %name que agregamos en el ícono permite que las interfaces gráficas muestren el nombre del componente al instanciarlos en un modelo.

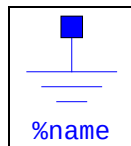


Figura 5.4: Vista de ícono del modelo de la puesta a tierra

Código 5.20: Puesta a Tierra en Modelica con Anotaciones Gráficas

```
model Ground
  Pin p annotation(Placement(visible = true, transformation(origin = {-102, 0}, extent =
    {{-10, -10}, {10, 10}}, rotation = 0), iconTransformation(origin = {0, 80}, extent =
    {{-18, -18}, {18, 18}}, rotation = 0)));
equation
  p.v = 0;
  annotation(
    Icon(coordinateSystem(preserveAspectRatio = true, extent = {{-100, -100}, {100, 100}}),
      graphics = {Line(origin = {0, -30}, points = {{-60, 50}, {60, 50}}, color = {0, 0, 255}),
        Line(origin = {0, -30}, points = {{-40, 30}, {40, 30}}, color = {0, 0, 255}), Line(
          origin = {0, -30}, points = {{-20, 10}, {20, 10}}, color = {0, 0, 255}), Line(origin =
          {0, -30}, points = {{0, 96}, {0, 50}}, color = {0, 0, 255}), Text(origin = {0, -30},
          textColor = {0, 0, 255}, extent = {{-150, -10}, {150, -50}}, textString = "%name")}));
end Ground;
```

De manera similar, podemos crear la clase parcial `OnePort` que habíamos presentado en el Código 5.9, pero ahora declarando las variables `i` y `v` con su tipo correspondiente y arrastrando dos conectores de la clase `Pin` a la vista de diagrama, de manera que la herramienta genere las anotaciones gráficas como muestra el Código 5.21.

Código 5.21: Modelo parcial para un dipolo con anotaciones gráficas

```
partial model OnePort
  Pin p annotation(...);
  Pin n annotation(...);
  Voltage v;
  Current i;
equation
  v = p.v - n.v;
  p.i = i;
  n.i + p.i = 0;
end OnePort;
```

Luego, a partir de esta clase podemos crear elementos como un resistor, capacitor e inductor, agregando a la vista de ícono de cada modelo el dibujo correspondiente (los conectores ya estarán dibujados al heredarlos del modelo parcial). En la Figura 5.5 se muestran los íconos del resistor, capacitor e inductor. En el texto, el símbolo `%R` implica que dicho texto se va a reemplazar en los diagramas que construyamos por el valor del parámetro `R`. Como antes, el símbolo `%name` se reemplazará por el nombre del componente en el modelo.

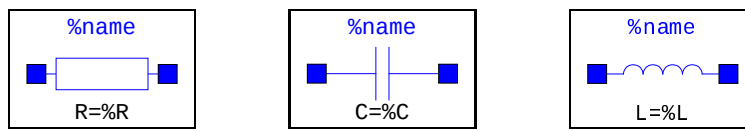


Figura 5.5: Vista de ícono de los dipolos pasivos.

Construcción Gráfica de Modelos

Los componentes creados con íconos y vista de sus conectores se pueden instanciar y acoplar de manera gráfica. Para eso, bastará con crear un nuevo modelo y arrastrar los componentes a la vista de diagrama y luego conectarlos haciendo click y arrastrando sobre el propio esquema. Una vez hecho esto, es posible modificar los parámetros u otros atributos de los objetos (como el valor inicial) haciendo doble click sobre el objeto, o bien yendo a la vista de texto y procediendo como antes.

La Figura 5.6 muestra el esquema conectado de un circuito RLC tal como se ve en la interfaz gráfica de OpenModelica. El modelo correspondiente, tal como se ve en la vista de texto, se puede ver en el Código 5.22. Notar que la interfaz gráfica muestra además las unidades de los parámetros en caso que al definir las clases hayamos puesto unidades a los mismos.

Código 5.22: Vista de texto del modelo RLC de la Figura 5.6

```
model RLC
  Resistor res annotation(...);
  Ground ground annotation(...);
  Capacitor cap(C=100e-6, v(start = 1)) annotation(...);
  Inductor ind(L=1e-3) annotation(...);
equation
  connect(ind.n, res.p) annotation(...);
  connect(res.n, cap.p) annotation(...);
  connect(cap.n, gr.p) annotation(...);
  connect(ind.p, gr.p) annotation(...);
end RLC;
```

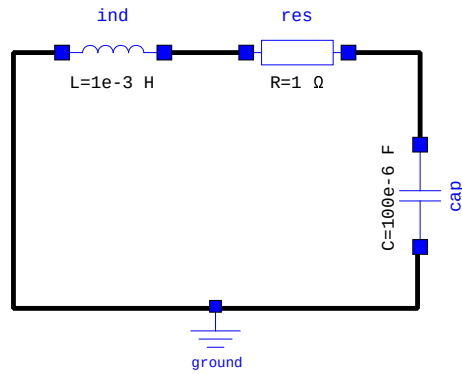


Figura 5.6: Vista de Diagrama de un Circuito RLC en Modelica

Modelado Jerárquico

También podemos construir de manera gráfica modelos jerárquicos, arrastrando componentes y conectores para armar subsistemas y luego creando íconos para dichos subsistemas de manera que puedan utilizarse como si fueran nuevos componentes.

Por ejemplo, la Figura 5.7 muestra el circuito equivalente de una celda solar, al que le agregamos dos conectores de tipo Pin para poder conectarlo con otros componentes, con su correspondiente vista de Ícono.

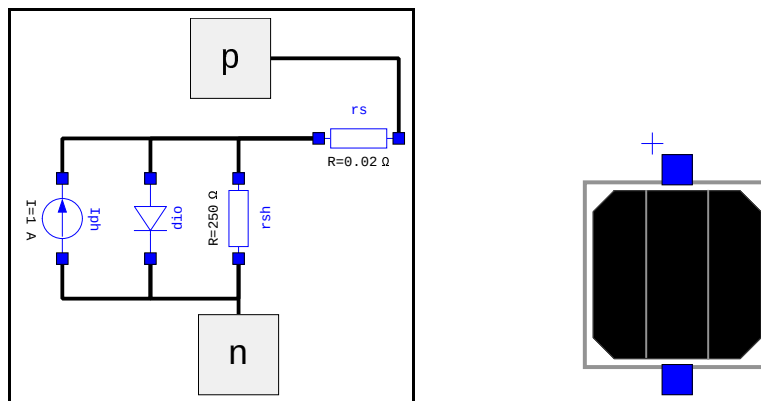


Figura 5.7: Vistas de Diagrama y de Ícono del Circuito Equivalente de una Celda Solar

Luego, usando este nuevo componente jerárquico, podemos crear modelos como el de la Figura 5.8

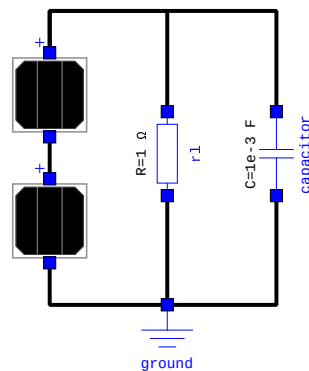


Figura 5.8: Vistas de Diagrama de un modelo con dos celdas solares

Organización de la Librería

Dado que al construir una librería se definen muchas clases (tipos, conectores, modelos parciales, componentes, etc.) conviene organizar la librería usando paquetes anidados. Por ejemplo, el Código 5.23 contiene parte de una librería denominada `Circuits` y la Figura 5.9 muestra como se ve la misma en el explorador de librerías de OpenModelica. Este paquete contiene dentro cuatro paquetes: uno con las unidades (`Units`), otro con elementos auxiliares para construir componentes (`Interfaces`), otro con los componentes (`Components`) y el último con los ejemplos (`Examples`).

Dentro de este último paquete, se encuentra otro paquete denominado `Panels` en el que está el ejemplo de los dos paneles solares, junto con el modelo del componente jerárquico de un panel solar. Cuando se construyen ejemplos complejos que cuentan con componentes creados de manera jerárquica (o simplemente con componentes creados especialmente para dicho modelo), conviene hacer un paquete para todo el modelo, incluyendo los nuevos componentes creados.

Código 5.23: Librería con paquetes anidados

```

package Circuits

  package Units
    type Voltage = Real(unit="V");
    type Current = Real(unit="A");
  end Units;

  package Interfaces
    import Circuits.Units.*;
    connector Pin
    ...
    partial model OnePort
    ...
  end Interfaces;

  package Components
    import Circuits.Interfaces.*;
    model Ground
    ...
    model Resistor
    ...
    model Capacitor
    ...
    model Inductor
    ...
  end Components;

  package Examples
    import Circuits.Components.*;
    model RLC
    ...
    package Panels
      model Panel
      ...
    end Panels;
  end Examples;
end Circuits;

```

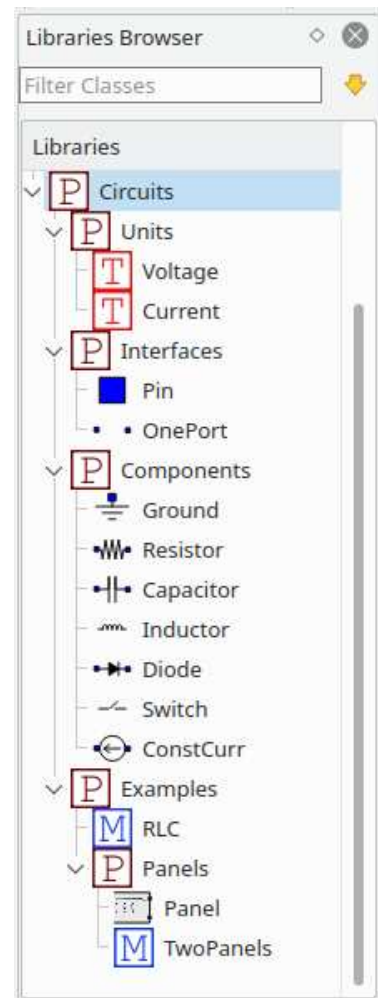


Figura 5.9: Librería con Paquetes Anidados

En el paquete `Interfaces` se utiliza el comando `import Circuits.Unit.*`. Esto permite que todas

las clases definidas en el paquete `Circuits.Unit` estén disponibles en `Interfaces`. Esto se repite luego en los siguientes paquetes. De no utilizar el comando `import`, habría que utilizar los caminos completos al instanciar cada clase. Por ejemplo, para instanciar un objeto `res` de la clase `Resistor` deberíamos escribir `Circuits.Components.Resistor res;`.

5.2.2. Sistemas Mecánicos

Siguiendo exactamente las mismas ideas que para los circuitos eléctricos, podemos construir librerías de sistemas mecánicos traslacionales, rotacionales y de mecánica en el plano o incluso mecánica multicuerpos (en tres dimensiones).

Comenzaremos entonces con los sistemas mecánicos traslacionales.

Mecánica Traslacional

Así como en los circuitos las variables que intervenían eran los potenciales (que se igualaban en las conexiones) y las corrientes (que sumaban cero en las conexiones), en los sistemas mecánicos dicho rol lo podrán cumplir las posiciones (que se igualarán en las conexiones) y las fuerzas (que sumarán cero en una conexión).

El conector entonces, que llamaremos `Flange`, tendrá dos variables: la posición y la fuerza, siendo esta último de tipo `flow`. Usando luego estos conectores, de manera análoga a lo que hicimos con el modelo parcial para definir un dipolo, podemos definir el modelo parcial que represente un objeto deformable (que nos permitirá definir un resorte o un amortiguador).

Con estos componentes, podemos comenzar a construir la librería como muestra el Código 5.24.

Código 5.24: Unidades, conectores y modelos parciales de una librería de mecánica traslacional

```
package Translational
package Units
  type Position = Real(unit="m");
  type Force = Real(unit="N");
end Units;

package Interfaces
import Translational.Units.*;
connector Flange
  Position s;
  flow Force f;
annotation(...);
end Flange;

partial model Compliant
  Flange flange_b annotation(...);
  Flange flange_a annotation(...);
  Position s_rel;
  Force f;
equation
  s_rel = flange_b.s - flange_a.s;
  flange_b.f = f;
  flange_a.f = -f;
end Compliant;
end Interfaces;
...
```

En el modelo parcial `Compliant` la deformación `s_rel` es positiva cuando la posición del conector `flange_b` es mayor a la de `flange_a`. Podemos interpretar que el `flange_b` es el del lado derecho mientras el `flange_a` es el izquierdo. La fuerza `f`, en tal caso, es la que se ejerce sobre el componente considerándola positiva hacia la derecha.

En base a estas clases, podemos crear entonces componentes elementales de la misma forma que hicimos antes, incluyendo sus íconos para poder construir los modelos desde la interfaz gráfica.

Código 5.25: Modelos elementales de una librería de mecánica traslacional

```

package Components
import Translational.Units.*;
import Translational.Interfaces.*;
model Fixed
  Flange flange annotation(...);
  parameter Position s_0=0;
  equation
    flange.s=s_0;
  annotation(...);
end Fixed;

model Mass
  Flange flange annotation(...);
  parameter Real m(unit="Kg")=1;
  Position s;
  Real v(unit="m/s");
  Force f;
  equation
    s=flange.s;
    f=flange.f;
    m*der(v)-f=0;
    der(s)-v=0;
  annotation(...);
end Mass;

model Spring
  extends Compliant;
  parameter Real k(unit="N/m")=1;
  parameter Position s_rel0=0;
  equation
    f=k*(s_rel-s_rel0);
  annotation(...);
  end Spring;

model Damper
  extends Compliant;
  parameter Real b(unit="N.s/m")=1;
  equation
    f=b*der(s_rel);
  annotation(...);
end Damper;

model ConstForce
  Flange flange annotation(...);
  parameter Force F = 1;
  equation
    flange.f = -F;
  annotation(...);
end ConstForce;

end Components;

```

Utilizando estos componentes podemos entonces armar fácilmente un sistema Masa-Resorte-Amortiguador conectando directamente en la interfaz gráfica como se muestra en la Figura 5.10, lo que a su vez resulta en el modelo de la vista de texto del Código 5.26.

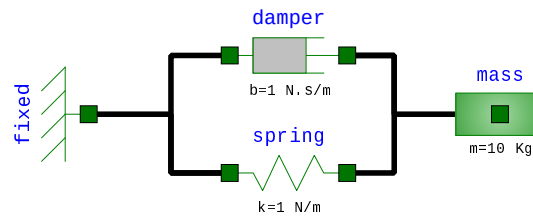


Figura 5.10: Vista de Diagrama de un sistema Masa-Resorte-Amortiguador en Modelica

Código 5.26: Vista de texto del modelo Masa-Resorte-Amortiguador de la Figura 5.10

```

model SpringDamperMass
  Fixed fixed annotation(...);
  Mass mass(m = 10, s(start = 1)) annotation(...);
  Spring spring annotation(...);
  Damper damper annotation(...);
equation
  connect(fixed.flange, spring.flange_a) annotation(...);
  connect(spring.flange_b, mass.flange) annotation(...);
  connect(damper.flange_b, spring.flange_b) annotation(...);
  connect(damper.flange_a, spring.flange_a) annotation(...);
end SpringDamperMass;

```

Mecánica Rotacional

Siguiendo prácticamente la misma idea de la mecánica traslacional, podemos construir una librería y modelos de mecánica rotacional. Reemplazando las posiciones por ángulos y las fuerzas por torques, podemos crear las clases básicas que definen unidades y conectores como se muestra en el Código 5.27 y con estos, los componentes elementales que se muestran en el Código 5.28.

Código 5.27: Unidades, conectores y modelos parciales de una librería de mecánica rotacional

```

package Rotational
  package Units
    type Angle=Real(unit="rad");
    type Torque=Real(unit="N.m");
  end Units;

  package Interfaces
    import Rotational.Units.*;
    connector Flange
      Angle phi;
      flow Torque tau;
    end Flange;

    partial model Compliant
      Flange flange_b annotation(...);
      Flange flange_a annotation(...);
      Angle phi_rel;
      Torque tau;
    equation
      phi_rel = flange_b.phi - flange_a.phi;
      flange_b.tau = tau;
      flange_a.tau = -tau;
    end Compliant;
  end Interfaces;
...

```

Código 5.28: Modelos elementales de una librería de mecánica rotacional

```

package Components
import Rotational.Units.*;
import Rotational.Interfaces.*;
model Fixed
  parameter Angle phi_0=0;
  Flange flange annotation(...);
  equation
    flange.phi=phi_0;
  annotation(...);
end Fixed;

model Inertia
  Rotational.Interfaces.Flange flange annotation(...);
  parameter Real J(unit="Kg.m2")=1;
  Angle phi;
  Real w(unit="rad/s");
  Torque tau;
  equation
    phi=flange.phi;
    tau=flange.tau;
    J*der(w)-tau=0;
    der(phi)-w=0;
  annotation(...);
end Inertia;

model Spring
  extends Compliant;
  parameter Real k(unit="N.m/rad")=1;
  parameter Angle phi_rel0=0;
  equation
    tau=k*(phi_rel-phi_rel0);
  annotation(...);
end Spring;

model Damper
  extends Compliant;
  parameter Real b(unit="N.m.s/rad")=1;
  equation
    tau=b*der(phi_rel);
  annotation(...);
end Damper;
end Components;

```

Utilizando estos componentes, podemos por ejemplo construir un modelo Inercia-Resorte-Amortiguador como el que se muestra en la Figura 5.11.

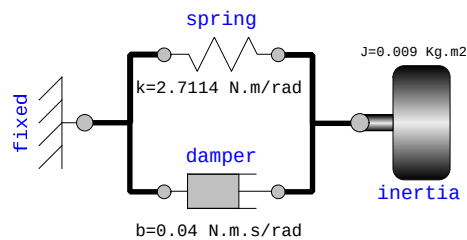


Figura 5.11: Vista de Diagrama de un sistema rotacional Inercia-Resorte-Amortiguador en Modelica

Sistemas Roto-traslacionales

Los modelos roto-traslacionales tendrán componentes que funcionen como interfaz entre las partes traslacionales y las rotacionales. Dichos componentes, por lo tanto, tendrán necesariamente conectores de ambos tipos.

Por ejemplo, podemos crear el modelo de un componente *piñón-cremallera* arrastrando un conector *Flange* traslacional y uno rotacional, y vinculando sus variables de posición y ángulo y las variables de fuerza y torque por otro. De esta forma, el modelo resultante es el del Código 5.29.

Código 5.29: Modelo de un Sistema Piñón-Cremallera

```

model RackPinion
  Mechanical.Translational.Interfaces.Flange flangeT annotation(...);
  Mechanical.Rotational.Interfaces.Flange flangeR annotation(...);
  parameter Real r(unit="m")=1 "Pinion radius";

equation
  flangeT.s - r*flangeR.phi = 0;
  r*flangeT.f + flangeR.tau = 0;

  annotation(...);
end RackPinion;

```

Notar que en los conectores siempre asumimos que la fuerza y el torque (o la corriente en el caso eléctrico) son positivos cuando ingresan al componente. Por lo tanto, en un componente de tipo transformador dichos flujos deben tener signo opuesto, lo que explica el signo de la ecuación $r \cdot \text{flangeT.f} + \text{flangeR.tau} = 0$.

En este caso además no utilizamos el comando `import`, sino que directamente instanciamos los conectores con todo el camino hacia el paquete donde está la clase correspondiente. Aquí, de hecho, no hubiéramos podido importar las clases ya que ambos tipos de conectores tienen el mismo nombre (*Flange*).

Utilizando este componente, podemos construir muy fácilmente el modelo correspondiente al sistema de la Figura 4.11. Dicho modelo, cuya definición textual está en el Código 5.30, se ve gráficamente como muestra la Figura 5.12.

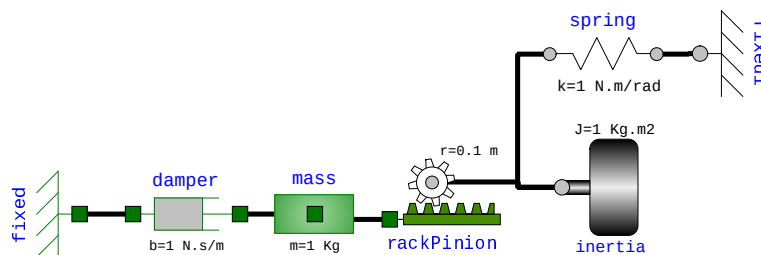


Figura 5.12: Modelo Roto-Traslacional en la vista de Diagrama

Código 5.30: Modelo Roto-Traslacional en la vista de Texto

```

model RotTrans
  Mechanical.RotoTranslational.Components.RackPinion rackPinion annotation(...);
  Mechanical.Translational.Components.Fixed fixed annotation(...);
  Rotational.Components.Inertia inertia(phi(start = 1)) annotation(...);
  Mechanical.Rotational.Components.Fixed fixed1 annotation(...);
  Mechanical.Rotational.Components.Spring spring annotation(...);
  Mechanical.Translational.Components.Damper damper annotation(...);
  Mechanical.Translational.Components.Mass mass annotation(...);

equation
  connect(rackPinion.flangeR, inertia.flange) annotation(...);
  connect(inertia.flange, spring.flange_a) annotation(...);

```

```

connect(spring.flange_b, fixed1.flange) annotation(...);
connect(damper.flange_a, fixed.flange) annotation(...);
connect(mass.flange, rackPinion.flangeT) annotation(...);
connect(damper.flange_b, mass.flange) annotation(...);
end RotTrans;

```

Mecánica Planar

Extenderemos ahora las ideas que vimos hasta acá para construir una librería de mecánica en el plano, basándonos principalmente en la librería descrita en [Zim12].

A diferencia del Capítulo 4 donde nos limitamos a la presencia de masas puntuales, aprovecharemos las ventajas del modelado orientado a objetos para considerar cuerpos rígidos con momento de inercia. Por este motivo, además de posiciones y fuerzas, los conectores deberán vincular torque y ángulo.

Comenzaremos entonces definiendo las unidades, conectores y modelos parciales al igual que en los otros casos:

Código 5.31: Unidades, conectores y modelos parciales de una librería de mecánica planar simplificada

```

package Planar
package Units
  type Position = Mechanical.Translational.Units.Position[2];
  type Force = Mechanical.Translational.Units.Force[2];
end Units;

package Interfaces
import Mechanical.Planar.Units.*;
connector Frame
  Position r;
  flow Force f;
  Mechanical.Rotational.Units.Angle phi;
  flow Mechanical.Rotational.Units.Torque tau;
  annotation(...);
end Frame;

partial model Compliant
  Mechanical.Planar.Interfaces.Frame frame_a annotation(...);
  Mechanical.Planar.Interfaces.Frame frame_b annotation(...);
  Real l(unit = "m");
  Real phi(unit = "rad");
  Real w(unit = "rad/s");
  Mechanical.Translational.Units.Force f;
equation
  frame_b.r[1] - frame_a.r[1] = l * cos(phi);
  frame_b.r[2] - frame_a.r[2] = l * sin(phi);
  der(phi) = w;
  phi=frame_a.phi;
  phi=frame_b.phi;
  frame_a.f = -frame_b.f;
  frame_b.f[1]*cos(phi)+frame_b.f[2]*sin(phi)=f;
  frame_a.tau+frame_b.tau+l * cos(phi) *frame_b.f[2]-l * sin(phi)*frame_b.f[1]=0;
end Compliant;
end Interfaces;

...

```

Notar que utilizamos magnitudes traslacionales y rotacionales donde las magnitudes de posición y fuerza tienen dos dimensiones. Los conectores `Frame` vinculan ahora estas magnitudes vectoriales además de las escalares correspondientes a las variables angulares.

El modelo parcial `Compliant` cumple un rol análogo al del traslacional, representando un objeto sin masa y dos conectores que pueden desplazarse sólo en sentido longitudinal. El mismo tiene ahora una variable `phi` que representa el ángulo que forma el componente con la dirección horizontal. En cuanto a las ecuaciones, además de las relaciones geométricas obvias entre posiciones y ángulos, aparecen las que hacen el balance de fuerzas ($\text{frame_a.f} = -\text{frame_b.f}$), la transmisión de fuerza longitudinal ($\text{frame_b.f}[1]*\cos(\text{phi})+\text{frame_b.f}[2]*\sin(\text{phi})=f$), y la última que realiza el balance de torques.

A partir de estas clases básicas, se pueden definir los siguientes componentes elementales:

Código 5.32: Modelos elementales de una librería de mecánica planar

```

package Components
import Mechanical.Planar.Units.*;
import Mechanical.Planar.Interfaces.*;

model Fixed
  parameter Position r = {0, 0};
  parameter Angle phi = 0;
  Frame frame annotation(...);
equation
  frame.r = r;
  frame.phi = phi;
  annotation(...);
end Fixed;

model pointMass
  Frame frame annotation(...);
  parameter Real m(unit = "Kg") = 1 "Mass of the body";
  parameter Real[2] g(each unit = "m/s2") = {0, -9.81} "Gravity acceleration";
  Position r;
  Real[2] v;
equation
  r = frame.r;
  v = der(r);
  m*der(v) = frame.f + m*g;
  frame.tau=0;
  annotation(...);
end pointMass;

model Body
  DSFLib.Mechanical.Planar.Interfaces.Frame frame annotation(...);
  parameter Real m(unit = "Kg") = 1 "Mass of the body";
  parameter Real J(unit = "Kg.m2") = 0.1 "Inertia of the body";
  parameter Real[2] g(each unit = "m/s2") = {0, -9.81} "Gravity acceleration";
  Position r;
  Real[2] v;
  Angle phi;
  Real w;
equation
  r = frame.r;
  phi = frame.phi;
  v = der(r);
  w = der(phi);
  m*der(v) = frame.f + m*g;
  J*der(w) = frame.tau;
  annotation(...);
end Body;

model RigidBar
  extends Compliant;

```

```

    parameter Real L(unit = "m") = 1;
equation
  l = L;
  annotation(...);
end RigidBar;

model Prismatic
  extends Compliant;
  Mechanical.Translational.Interfaces.Flange support annotation(...);
  Mechanical.Translational.Interfaces.Flange flange_a annotation(...);
equation
  f=-flange_a.f;
  support.s=0;
  flange_a.s=1;
  annotation(...);
end Prismatic;

model Revolute
  Frame frame_a annotation(...);
  Frame frame_b annotation(...);
  Mechanical.Rotational.Interfaces.Flange support annotation(...);
  Mechanical.Rotational.Interfaces.Flange flange_a annotation(...);
equation
  frame_b.r = frame_a.r;
  frame_a.f = -frame_b.f;
  frame_a.tau ==-frame_b.tau;
  frame_b.tau=-flange_a.tau;
  support.phi=0;
  flange_a.phi=frame_b.phi-frame_a.phi;
  annotation(...);
end Revolute;

end Components;

```

Esta librería tiene algunos elementos novedosos, como la definición de ciertos parámetros (fuerzas y posiciones) con valores vectoriales. En Modelica los vectores se expresan entre llaves $\{x_1, x_2, x_3, \dots\}$.

Los componentes **Fixed**, **PointMass** y **Body** tienen definiciones completamente análogas a las de sus contrapartes traslacionales y rotacionales, sólo que involucran magnitudes en dos dimensiones traslacionales y en una dimensión rotacional. Además, en el caso de las masas, incorporamos por omisión la aceleración de la gravedad en el sentido vertical (se puede suprimir cambiando adecuadamente el valor del parámetro vectorial **g**). Por otro lado, el componente **RigidBar** es una extensión trivial de la clase **Compliant** fijando una longitud constante.

Los restantes componentes son dos articulaciones esenciales para modelar cualquier sistema. La articulación prismática (**Prismatic**) brinda una restricción al movimiento entre los dos extremos, permitiendo sólo el desplazamiento en sentido longitudinal. Se puede por lo tanto derivar directamente de la clase **Compliant**, y en principio bastaría con igualar a cero las fuerzas en sentido longitudinal (la articulación no transmite fuerza en el sentido en el que se mueve). En tal caso la ecuación sería simplemente $f=0$.

Sin embargo, la articulación tiene además dos conectores traslacionales (tipo **Flange**), cada uno solidario a un extremo del objeto, que transmiten fuerza a los extremos en sentido longitudinal. Estos conectores pueden usarse para agregar componentes entre los mismos (resortes, amortiguadores, fuentes de fuerza) con la condición de que dichos componentes traslacionales formen un esquema que actúe entre ambos. El subsistema traslacional entre ambos conectores tendrá como origen la posición del que se denomina **support**. No pueden utilizarse masas traslacionales en dichos subsistemas ya que el conector de origen podría estar en un marco no inercial y además al haber masa involucrada no se cumplirá en general que el subsistema ejerza la misma fuerza hacia ambos conectores traslacionales.

La otra articulación es la de revolución, que funciona como un rodamiento, permitiendo el giro entre dos conectores pero no el desplazamiento entre ambos. Análogo al caso de la articulación prismática, en

principio este componente podría definirse igualando posiciones, anulando la suma de fuerzas y anulando también el torque transmitido entre los extremos. Sin embargo, se agregan aquí también dos conectores rotacionales (uno solidario a cada extremo) de manera que el torque transmitido es el del subsistema rotacional conectado a los mismos. Valen también las mismas consideraciones respecto a los componentes rotacionales que se pueden conectar al mismo (no se deben incluir inercias).

Usando estos componentes podemos crear de manera muy simple modelos cuya formulación directa a partir de DAEs sería muy complicada. Por ejemplo, la Figura 5.13 muestra el modelo de un doble péndulo construido utilizando estos componentes.

```

model DoublePendulum
  Fixed fixed(phi = -1.57079);
  Revolute revolute;
  RigidBar rigidBar(phi.start=-1);
  pointMass pointMass(m = 0.5);
  pointMass pointMass1;
  RigidBar rigidBar1;
  Revolute revolute1;
equation
  connect(fixed.frame, revolute.frame_a);
  connect(revolute.frame_b, rigidBar.frame_a);
  connect(pointMass.frame, rigidBar.frame_b);
  connect(pointMass1.frame, rigidBar1.frame_b);
  connect(pointMass.frame, revolute1.frame_a);
  connect(revolute1.frame_b, rigidBar1.frame_a);
end DoublePendulum;
    
```

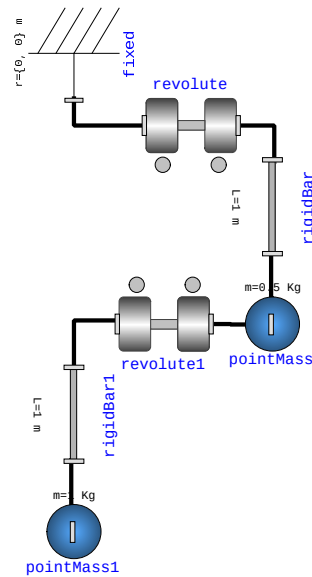


Figura 5.13: Vista de Diagrama y de Texto de un doble péndulo en Modelica

Aprovechando también los conectores rotacionales y traslacionales podemos agregar también componentes de fricción, resortes, fuerzas y torques de entrada, etc. Por ejemplo, el sistema biela-manivela de la Figura 4.12 de la página 78 puede modelarse de manera mucho más simple usando esta librería, tal como se muestra en la Figura 5.14 y el Código 5.33.

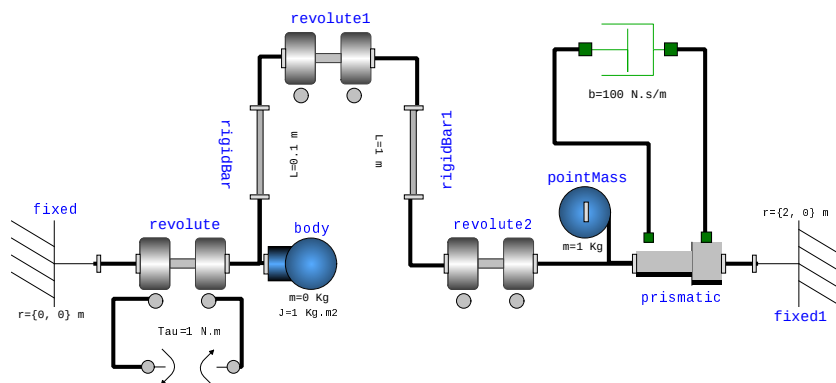


Figura 5.14: Sistema Biela-Manivela en la vista de Diagrama

Código 5.33: Sistema Biela-Manivela en la vista de Texto

```

model SliderCrank
  Fixed fixed;
  Revolute revolute;
    
```

```

RigidBar rigidBar(L = 0.1);
Revolute revolute1;
RigidBar rigidBar1(L = 1);
Revolute revolute2;
Prismatic prismatic;
Fixed fixed1(phi = 3.14159, r = {2, 0});
pointMass pointMass(g = {0, 0});
Body body(J = 1, g = {0, 0}, m = 0);
Mechanical.Translational.Components.Damper damper(b = 100);
Mechanical.Rotational.Components.ConstTorque constTorque;
equation
connect(revolute.frame_a, fixed.frame);
connect(revolute.frame_b, rigidBar.frame_a);
connect(rigidBar.frame_b, revolute1.frame_a);
connect(revolute1.frame_b, rigidBar1.frame_a);
connect(rigidBar1.frame_b, revolute2.frame_a);
connect(revolute2.frame_b, prismatic.frame_b);
connect(prismatic.frame_a, fixed1.frame);
connect(pointMass.frame, prismatic.frame_b);
connect(body.frame, rigidBar.frame_a);
connect(damper.flange_a, prismatic.support);
connect(prismatic.flange_a, damper.flange_b);
connect(constTorque.flange_a, revolute.support);
connect(constTorque.flange_b, revolute.flange_a);
end SliderCrank;

```

5.2.3. Sistemas Hidráulicos

Teniendo en cuenta las analogías que hay entre los sistemas hidráulicos y los eléctricos, podemos también de manera muy simple construir una librería de componentes elementales. En este caso, asumiremos que los sistemas pueden modelarse mediante *circuitos hidráulicos* en los cuales circula un fluido incompresible.

Bajo estas hipótesis, reemplazando los potenciales por presiones y las corrientes por caudales, podemos crear las clases básicas que definen unidades y conectores como se muestra en el Código 5.34 y con estos, los componentes elementales que se muestran en el Código 5.35.

Código 5.34: Unidades, conectores y modelos parciales de una librería de circuitos hidráulicos

```

package Hydraulics
package Units
type Pressure = Real(unit = "Pa");
type VolumeFlow = Real(unit = "m3/s");
end Units;

package Interfaces
import Hydraulics.Units.*;

connector FluidPort
Pressure p;
flow VolumeFlow q;
annotation(...);
end FluidPort;

partial model TwoPort
FluidPort fluidPort_b annotation(...);
FluidPort fluidPort_a annotation(...);
Pressure p;
VolumeFlow q;

```



```

equation
  p = fluidPort_b.p - fluidPort_a.p;
  q = fluidPort_b.q;
  q = -fluidPort_a.q;
end TwoPort;
end Interfaces;
...

```

Código 5.35: Modelos elementales de una librería de circuitos hidráulicos

```

package Components
import Hydraulics.Interfaces.*;
import Hydraulics.Units.*;

model Valve
  extends TwoPort;
  parameter Real RH(unit = "Pa.s/m3") = 1000;
equation
  p - RH * q = 0;
  annotation(...);
end Valve;

model Tank
  parameter Real A(unit = "m2") = 1 "Tank area";
  parameter Real g(unit = "m/s2") = 9.8;
  parameter Real rho(unit = "Kg/m3") = 1000 "Fluid density";
  FluidPort fluidPort annotation(...);
  VolumeFlow q;
  Pressure p;
  Pressure dp;
equation
  q = der(v);
  p = v * rho * g / A;
  dp = if v>0 or q>0 then 0 else 1e10*q;
  fluidPort.q = q;
  fluidPort.p = p + dp;
  annotation(...);
end Tank;

model Inertance
  extends TwoPort;
  parameter Real I(unit = "Pa.s2/m3") = 1;
equation
  I * der(q) = p;
  annotation(...);
end Inertance;

model Column
  extends TwoPort;
  parameter Real H(unit = "m") = 1 "Column height";
  parameter Real g(unit = "m/s2") = 9.8;
  parameter Real rho(unit = "Kg/m3") = 1000 "Fluid density";
equation
  p = rho * g * H;
  annotation(...);
end Column;

model ConstPress
  Pressure P = 0;

```

```

    FluidPort fluidPort annotation(...);
  equation
    fluidPort.p = P;
    annotation(...);
  end ConstPress;

  model IdealPump
    extends TwoPort;
    parameter Real Q(unit = "m3/s") = 1e-3;
  equation
    q = Q;
    annotation(...);
  end IdealPump;
end Components;

```

Estos modelos se corresponden directamente con las relaciones constitutivas vistas en el Capítulo 4 excepto en el caso del tanque, que tiene una modificación para evitar que el volumen en el mismo se torne negativo. Dicha modificación es equivalente al agregado de una válvula en el ingreso con mucha resistencia hidráulica cuando el volumen se hace negativo y nula resistencia cuando el volumen es positivo.

Utilizando entonces estos componentes podemos de manera muy simple construir ejemplos, como el del Código 5.36 que se muestra en la Figura 5.15.

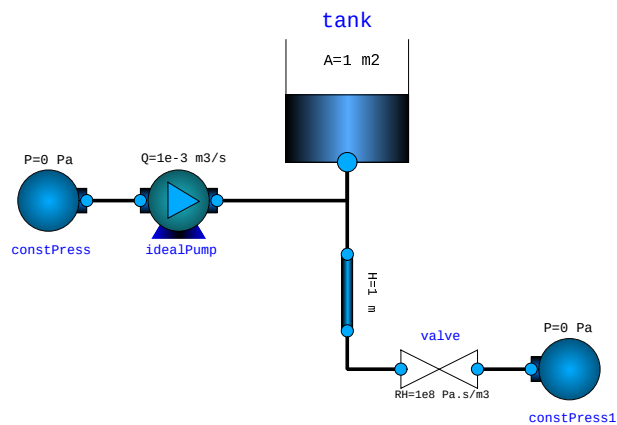


Figura 5.15: Modelo Hidráulico en la vista de Diagrama

Código 5.36: Modelo Hidráulico en la vista de Texto

```

model TankValve
  Tank tank(v(start = 1));
  Valve valve(RH = 1e8);
  ConstPress constPress;
  Column column;
  IdealPump idealPump;
equation
  connect(constPress.fluidPort, valve.fluidPort_a);
  connect(tank.fluidPort, column.fluidPort_a);
  connect(column.fluidPort_b, valve.fluidPort_b);
  connect(idealPump.fluidPort_b, constPress.fluidPort);
  connect(idealPump.fluidPort_a, tank.fluidPort);
end TankValve;

```

5.2.4. Sistemas Térmicos

Procediendo de nuevamente de forma análoga, podemos construir una librería de modelos térmicos simples, considerando exclusivamente fenómenos de transmisión de calor de parámetros concentrados.

Bajo estas hipótesis, reemplazando los potenciales por temperaturas y las corrientes por flujos de calor, podemos crear las clases básicas que definen unidades y conectores como se muestra en el Código 5.37 y con estos, los componentes elementales que se muestran en el Código 5.38.

Código 5.37: Unidades, conectores y modelos parciales de una librería de sistemas térmicos

```
package Thermal

  package Units
    type Temperature = Real(unit="K",start=293);
    type HeatFlow = Real(unit="W");
  end Units;

  package Interfaces
    import Thermal.Units.*;
    connector HeatPort
      Temperature T;
      flow HeatFlow q;
      annotation(...);
    end HeatPort;

    partial model TwoPort
      HeatPort heatPort_b annotation(...);
      HeatPort heatPort_a annotation(...);
      Temperature T_rel;
      HeatFlow q;
    equation
      T_rel=heatPort_b.T-heatPort_a.T;
      q=heatPort_b.q;
      q=-heatPort_a.q;
    end TwoPort;
  end Interfaces;

  ...
```

Notar que para el tipo `Temperature` definimos como valor por omisión 293°K , lo que se corresponde a cero grados Celsius.

Código 5.38: Modelos elementales de una librería de sistemas térmicos

```
package Components
  import Thermal.Units.*;
  import Thermal.Interfaces.*;

  model HeatCapacitor
    HeatPort heatPort annotation(...);
    parameter Real C(unit="J/K")=1;
    Temperature T;
    HeatFlow q;
  equation
    C*der(T)=q;
    q=heatPort.q;
    T=heatPort.T;
    annotation(...);
  end HeatCapacitor;

  model ThermalResistor
    extends TwoPort;
```

```

    parameter Real R(unit="K/W")=1;
  equation
    R * q=T_rel;
    annotation(...);
  end ThermalResistor;

  model HeatFlowSource
    HeatPort heatPort annotation(...);
    parameter HeatFlow Q =1;
  equation
    heatPort.q=-Q;
    annotation(...);
  end HeatFlowSource;

  model ConstTemp
    HeatPort heatPort annotation(...);
    parameter Temperature T =273.15;
  equation
    heatPort.T=T;
    annotation(...);
  end ConstTemp;
end Components;

```

Entre los componentes no incluimos el correspondiente al fenómeno de convección, ya que el mismo involucra una variable hidráulica (caudal) por lo que lo definiremos más adelante como componente multidominio.

Utilizando entonces estos componentes elementales podemos de manera muy simple construir ejemplos, como el del Código 5.39 que se muestra en la Figura 5.16, que representa la transferencia de calor entre dos recintos, uno de los cuales recibe un flujo constante mientras que el otro disipa calor al ambiente.

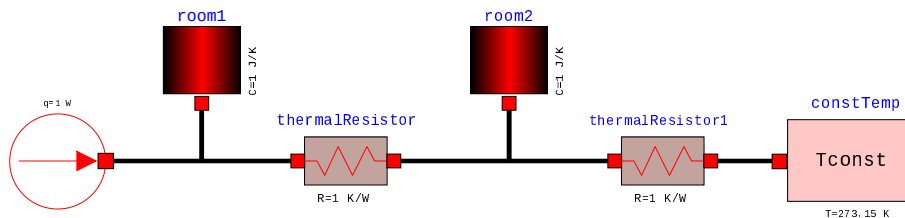


Figura 5.16: Modelo Térmico en la vista de Diagrama.

Código 5.39: Modelo Térmico en la vista de Texto

```

model TwoCompartments
  HeatCapacitor room1;
  HeatCapacitor room2;
  ThermalResistor thermalResistor;
  HeatFlowSource heatFlowSource;
  ThermalResistor thermalResistor1;
  ConstTemp constTemp(T = 293);
equation
  connect(room1.heatPort, thermalResistor.heatPort_b);
  connect(thermalResistor.heatPort_a, room2.heatPort);
  connect(thermalResistor1.heatPort_b, room2.heatPort);
  connect(thermalResistor1.heatPort_a, constTemp.heatPort);
  connect(heatFlowSource.heatPort, room1.heatPort);
end TwoCompartments;

```

5.2.5. Sistemas Multidominio

Cuando construimos modelos mecánicos que combinaban modelos traslacionales, rotacionales y planares, lo que hicimos fue crear componentes de interfaz que contenían conectores de los distintos tipos y vinculaban adecuadamente sus variables. Para crear modelos de distintos dominios procederemos de la misma manera.

Para poder utilizar además los componentes definidos en las distintas librerías pondremos todos los paquetes dentro de uno mayor que llamaremos `DSFLib`. Esta librería, que contiene los distintos componentes y ejemplos desarrollados en este capítulo, se puede descargar desde <https://www.fceia.unr.edu.ar/dsf/DSFLib.zip>.

Sistemas Electromecánicos

Dentro de los sistemas electromecánicos más utilizados encontramos los que realizan la conversión de energía electromagnética en mecánica, tal como lo hace el motor de corriente continua. Construiremos entonces primero un componente que representa esta conversión suponiendo que el campo de excitación es constante.

Código 5.40: Componentes electromecánico básico, con flujo de excitación constante

```

package ElectroMechanical
package Components
import DSFLib.Circuits.Interfaces.*;
import DSFLib.Circuits.Units.*;
import DSFLib.Circuits.Components.*;
import DSFLib.Mechanical.Rotational.Interfaces.*;
import DSFLib.Mechanical.Rotational.Units.*;

model EMF
  Pin p annotation(...);
  Pin n annotation(...);
  Flange flange annotation(...);
  Voltage e;
  Current i;
  Torque tau;
  Real w(unit="rad/s");
  parameter Real K(unit="V.s/rad")=1;

  equation
    e = p.v - n.v;
    p.i = i;
    n.i + p.i = 0;
    e=K*w;
    tau=K*i;
    tau=-flange.tau;
    w=der(flange.phi);
    annotation(...);
end EMF;

```

Utilizando este componente podemos fácilmente construir el modelo de un motor de corriente continua incluyendo la resistencia e inductancia de armadura en el lado eléctrico y el momento de inercia en el lado mecánico, como se muestra en la Figura 5.17.

Podemos también definir un componente para el caso de un motor con excitación independiente. Para esto utilizaremos dos conectores para la armadura (como antes) y dos conectores para la excitación. Además, agregaremos al modelo un inductor no lineal que represente la característica no lineal del circuito de excitación.

Código 5.41: Componente electromecánico con flujo de excitación independiente

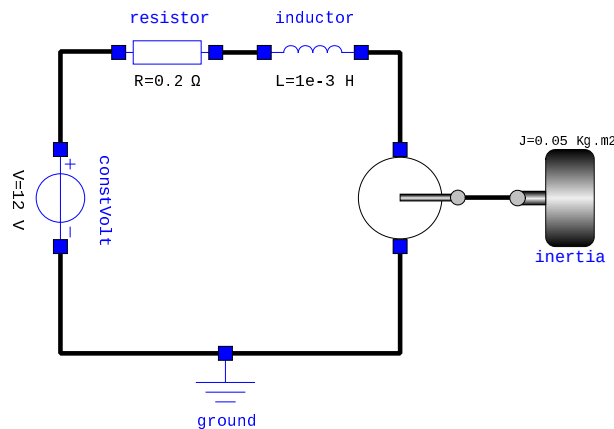


Figura 5.17: Modelo de un Motor de Corriente Continua en la vista de Diagrama.

```

model SepExcDCM
  parameter Real K(unit="1/rad")=1;
  parameter Real[:] currTable={-2,-1,0,1,2};
  parameter Real[:] fluxTable={-2,-1,0,1,2};
  NLInductor inductor(currTable=currTable,fluxTable=fluxTable) annotation(...);
  Pin p_ex annotation(...);
  Pin n_ex annotation(...);
  Pin p annotation(...);
  Pin n annotation(...);
  Flange flange annotation(...);
  Voltage e;
  Current i;
  Torque tau;
  Real w(unit="rad/s");

equation
  e = p.v - n.v;
  p.i = i;
  n.i + p.i = 0;
  e=K*inductor.phi*w;
  tau=K*inductor.phi*i;
  tau=-flange.tau;
  w=der(flange.phi);
  connect(p_ex, inductor.p) annotation(...);
  connect(inductor.n, n_ex) annotation(...);
  annotation(...);
end SepExcDCM;

```

En este componente, combinamos conexiones para el inductor no lineal y ecuaciones. Dicho inductor no lineal calcula la relación entre el flujo y la corriente interpolando o extrapolando linealmente sobre una tabla de valores de corriente y flujo. Notar que dicha tabla está declarada como parámetro para el modelo completo, y luego se pasa como parámetro al inductor no lineal. Esto se suele hacer a menudo al usar modelado jerárquico, definiendo ciertos parámetros globales para el sistema grandes, que luego se utilizan en algunos de sus componentes.

El modelo de este inductor no lineal está en el Código 5.42 y es muy similar al del inductor lineal excepto que reemplaza el parámetro L por una tabla de valores de corriente y flujo.

Código 5.42: Modelo de un inductor no lineal

```

model NLInductor
  extends OnePort;

```

```

parameter Real[:] currTable = {-2, -1, 0, 1, 2};
parameter Real[:] fluxTable = {-2, -1, 0, 1, 2};
Real phi(unit = "V.s");
equation
  der(phi) = v;
  phi = DSFLib.Utilities.Functions.LookUpTable(i, currTable, fluxTable);
  annotation(...);
end NLIInductor;

```

La interpolación o extrapolación sobre la tabla de corriente y flujo se realiza con una función llamada `LookUpTable` y cuyo código se muestra a continuación:

Código 5.43: Función de Interpolación o Extrapolación sobre una Tabla

```

function LookUpTable
  input Real x;
  input Real[:] xdata;
  input Real[:] ydata;
  output Real y;
protected
  Integer k;
algorithm
  k := 1;
  while x > xdata[k + 1] and k < size(xdata, 1) - 1 loop
    k := k + 1;
  end while;
  y := (ydata[k + 1] - ydata[k]) / (xdata[k + 1] - xdata[k]) * (x - xdata[k]) + ydata[k];
end LookUpTable;

```

El lenguaje Modelica permite definir funciones arbitrarias con distintas variables de entrada y salida, que se calculan siguiendo un algoritmo. Notar que en el mismo, no hay ecuaciones sino asignaciones a través del símbolo `:=`.

Volviendo entonces al nuevo componente que definimos en el Código 5.41, podemos crear el modelo de un motor de corriente continua con excitación en paralelo como se muestra en la Figura 5.18. En dicho diagrama de Modelica, el inductor no lineal y el motor son parte del ícono del componente correspondiente.

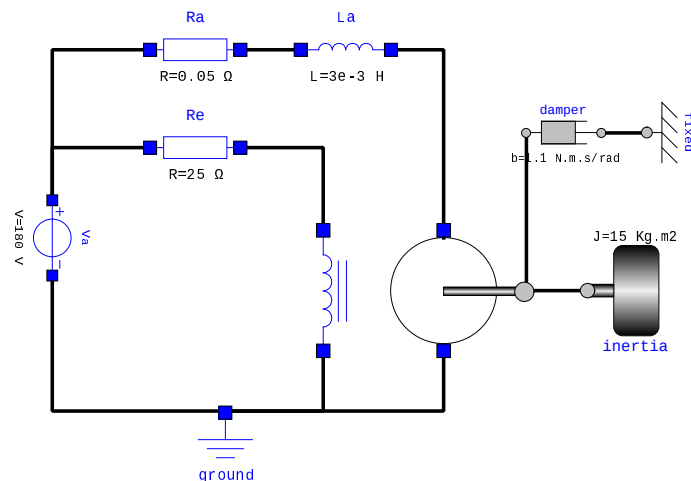


Figura 5.18: Modelo de un Motor de Corriente Continua con Excitación en Paralelo en la vista de Diagrama.

Sistemas Hidráulico-Mecánicos

El componente básico hidráulico mecánico que vimos en el Capítulo 4 es el pistón. En este caso, consideraremos un par cilindro-pistón que considera la posibilidad que el cilindro se mueva, por lo que el

caudal será proporcional a la velocidad relativa entre el pistón y el cilindro.

Código 5.44: Modelo de un Pistón-Cilindro

```

model PistonCylinder
  extends DSFLib.Mechanical.Translational.Interfaces.Compliant;
  DSFLib.Hydraulics.Interfaces.FluidPort fluidPort annotation(...);
  parameter Real A(unit="m^2")=1;
equation
  fluidPort.q=der(s_rel)*A;
  fluidPort.p*A=-f;
  annotation(...);
end PistonCylinder;

```

Notar que desde el punto de vista mecánico, utilizamos como base el modelo `Compliant` lo que nos permite ahorrar definiciones comunes.

Utilizando entonces este nuevo componente podemos crear el modelo de un gato hidráulico como se muestra en la Figura 5.19. En el mismo, ignoramos el peso del líquido dentro de los cilindros.

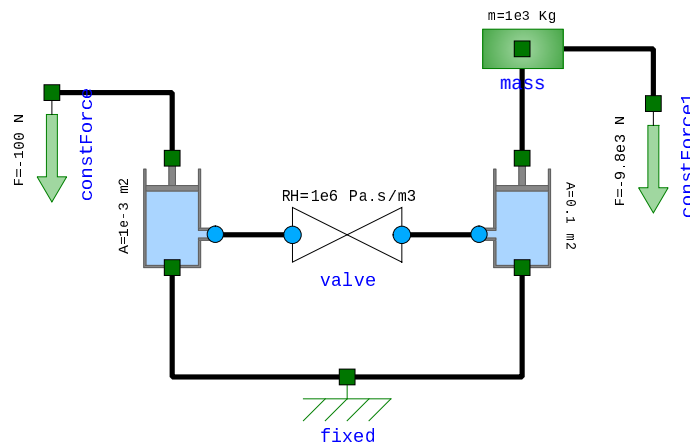


Figura 5.19: Modelo de un Gato Hidráulico en la vista de Diagrama.

Sistemas Térmo-Hidráulicos

Si bien se pueden plantear muchos componentes termo-hidráulicos (hay muchos fenómenos que ocurren involucrando ambos dominios en la termodinámica), veremos aquí sólo el componente correspondiente a la convección.

En dicho componente, supondremos que hay un caudal q circulando por un caño entre dos conectores hidráulicos y que el fluido ingresa y egresa con una temperatura T de manera que el flujo de calor entrante y saliente es $q_{\text{heat}} = \rho \cdot c \cdot q \cdot T$.

Con esta idea, podemos usar el siguiente modelo

Código 5.45: Modelo que representa el fenómeno de convección

```

model Convection
  extends DSFLib.Hydraulics.Interfaces.TwoPort;
  HeatPort heatPort_a annotation(...);
  HeatPort heatPort_b annotation(...);
  Temperature T;
  HeatFlow q_heat;
  parameter Real rho(unit = "Kg/m3") = 1000 "Fluid density";
  parameter Real c(unit = "J/K/Kg") = 4184 "Fluid specific heat capacity";
equation
  p = 0;

```

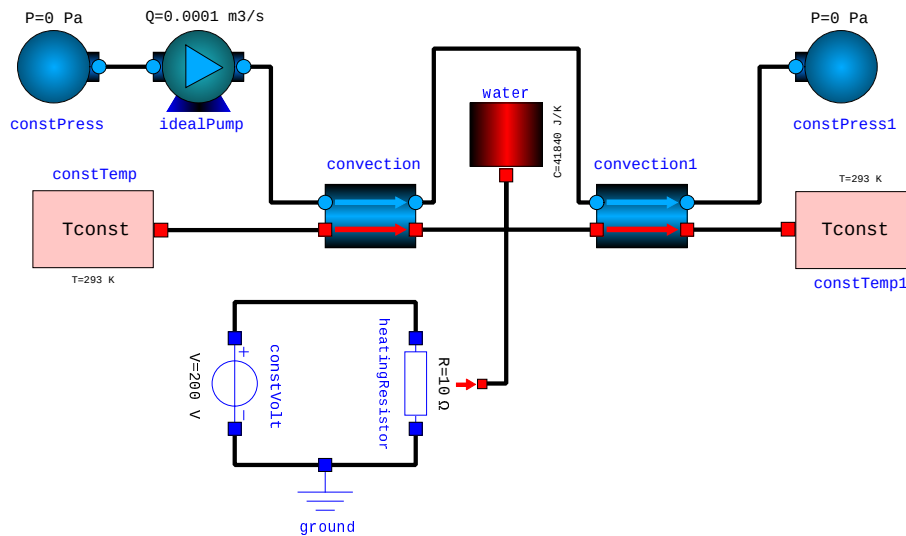



Figura 5.21: Modelo Simplificado de un Termostato Eléctrico en la vista de Diagrama.

5.3. Sistemas de Control y Modelos Causales

Hasta aquí trabajamos exclusivamente con modelos acausales, en los cuales las ecuaciones establecían relaciones entre las variables sin que aparezcan restricciones de causa-efecto. Esto, en sistemas puramente físicos, es lo más conveniente ya que nos permite acoplar componentes y subsistemas de manera muy sencilla, dejando el problema de causalización (necesario para simular o realizar análisis) a los algoritmos posteriores.

Sin embargo, en los sistemas artificiales las señales suelen involucrar un sentido causal. Por ejemplo, en un sistema de control, suelen medirse ciertas variables que funcionan como *entradas* para el sistema de control. A partir de estas mediciones, se suele calcular una acción de control (salidas del controlador) que son enviadas a los actuadores correspondientes. Por este motivo, es natural utilizar Diagramas de Bloques para representar estos sistemas.

El lenguaje Modelica cuenta con un tipo de clase específica (**block**) que permite crear bloques incorporando información causal y manteniendo todas las características que ya vimos para los modelos. Veremos entonces como formular Diagramas de Bloques luego veremos como combinar los mismos con modelos de sistemas físicos de los distintos dominios mediante la interfaz de sensores y actuadores.

5.3.1. Diagramas de Bloques

El lenguaje Modelica permite utilizar el prefijo **input** para indicar que una variable es una *entrada*. Esto quiere decir que dicha variable debe calcularse de forma externa al modelo donde está definida. De manera recíproca, puede utilizarse el prefijo **output** para indicar que una variable se calcula internamente en un modelo.

Podemos entonces definir conectores con estos prefijos y utilizarlos como entradas y salidas para construir diagramas de bloques. Cuando un conector tiene el prefijo **input**, el mismo puede sólo ser conectado con un conector que contenga el prefijo **output** y viceversa.

En base a estos principios, podemos comenzar a definir las clases básicas para una librería de diagramas de bloques como sigue:

Código 5.47: Interfaces Básicas de Diagramas de Bloques

```
package Blocks
  package Interfaces
    connector RealInput = input Real annotation(...);
    connector RealOutput = output Real annotation(...);
```

```

partial block SIS0
  DSFLib.ControlSystems.Blocks.Interfaces.RealInput u annotation(...);
  DSFLib.ControlSystems.Blocks.Interfaces.RealOutput y annotation(...);
  annotation(...);
end SIS0;
end Interfaces;
...

```

Como puede verse, al definir el modelo parcial SIS0 en lugar de utilizar `model` usamos `block`. Podríamos haber utilizado `model` en su lugar pero la clase `block` es más específica y sólo permite utilizar conectores de entrada y salida.

Usando luego estas clases básicas, podemos crear algunos componentes elementales para los Diagramas de Bloques como sigue:

Código 5.48: Componentes Básicos de Diagramas de Bloques

```

package Components
  block Integrator
    extends DSFLib.ControlSystems.Blocks.Interfaces.SIS0(y.start = y_0);
    parameter Real y_0 = 0;
  equation
    der(y) = u;
    annotation(...);
  end Integrator;

  block Gain
    extends DSFLib.ControlSystems.Blocks.Interfaces.SIS0;
    parameter Real K=1 "Gain";
  equation
    y = K * u;
    annotation(...);
  end Gain;

  block Add
    DSFLib.ControlSystems.Blocks.Interfaces.RealInput u1 annotation(...);
    DSFLib.ControlSystems.Blocks.Interfaces.RealInput u2 annotation(...);
    DSFLib.ControlSystems.Blocks.Interfaces.RealOutput y annotation(...);
    parameter Real k1=1 "Gain on first input";
    parameter Real k2=1 "Gain on second input";
  equation
    y = k1 * u1 + k2 * u2;
    annotation(...);
  end Add;

  block StepSource
    DSFLib.ControlSystems.Blocks.Interfaces.RealOutput y annotation(...);
    parameter Real U = 1 "Step amplitude";
    parameter Real Tstep = 0;
  equation
    y = if time < Tstep then 0 else U;
    annotation(...);
  end StepSource;

end Components;

```

Podemos también plantear algunos bloques más complejos, aprovechando la posibilidad de Modelica de utilizar matrices, como el siguiente que representa un sistema en ecuaciones de estados con una entrada y una salida.

Código 5.49: Bloque que modela un sistema en ecuaciones de estado

```

block StateSpace
  extends DSFLib.ControlSystems.Blocks.Interfaces.SISO;
  parameter Real A[:, size(A, 1)] = [0, 1; -1, -1];
  parameter Real B[size(A, 1), 1] = [0; 1];
  parameter Real C[1, size(A, 1)] = [1, 0];
  parameter Real D[1, 1] = [0];
  parameter Real x_0[size(A, 1), 1] = [0; 0];
  Real x[size(A, 1), 1](start = x_0);
protected
  Real ymat[1, 1];
equation
  der(x) = A * x + B * u;
  ymat = C * x + D * u;
  y = ymat[1, 1];
  annotation(...);
end StateSpace;

```

En este caso, los parámetros son matrices y la condición inicial. Un detalle es que en Modelica un objeto de la clase `Real` no es equivalente a un objeto de la clase `Real [1,1]`. Por eso, tuvimos que definir la salida *matricial* `ymat` y luego asignar su elemento a la salida `y`. El comando `protected` implica que todo lo que se declara a continuación es privado de la clase y no puede accederse (es decir, no podremos ver el atributo `ymat` desde otras clases).

Utilizando estas mismas ideas, podemos hacer un bloque de función transferencia, en el cual primero convertimos el modelo a ecuaciones de estado.

Código 5.50: Bloque que modela una Función Transferencia

```

block TransferFunction
  extends DSFLib.ControlSystems.Blocks.Interfaces.SISO;
  parameter Real num[:] = {1} "Numerator coefficients";
  parameter Real den[:] = {1, 1} "Denominator coefficients";
protected
  parameter Real A[size(den, 1)-1, size(den, 1)-1](each fixed=false);
  parameter Real B[size(den, 1)-1, 1](each fixed=false);
  parameter Real C[1, size(den, 1)-1](each fixed=false);
  parameter Real D[1, 1](each fixed=false);
  Real x[size(den, 1)-1, 1];
  Real ymat[1, 1];
initial algorithm
  (A,B,C,D):=DSFLib.Utilities.Functions.TF2SS(num,den);
equation
  der(x) = A * x + B * u;
  ymat = C * x + D * u;
  y = ymat[1, 1];
  annotation(...);
end TransferFunction;

```

En este caso, hay ciertos parámetros privados (por el comando `protected`) que no se asignan al declararse, sino que se calculan en una sección llamada `initial algorithm`. Para poder hacer esto, hay que avisar que dichos parámetros no tienen un valor fijo.

Como puede notarse, el algoritmo inicial convierte la función transferencia a ecuaciones de estado a través de una función (TF2SS) que devuelve las matrices *A*, *B*, *C* y *D* y luego, las ecuaciones son idénticas a las del bloque de ecuaciones de estado (Código 5.49).

Utilizando estos componentes, podemos fácilmente crear modelos arbitrarios como el que se muestra en la Figura 5.22 que puede representar una planta (en espacio de estados) controlada con un controlador proporcional integral (ganancia e integrador) que procesa el error entre la salida de la planta medida con un sensor (función transferencia) y la referencia (entrada escalón).

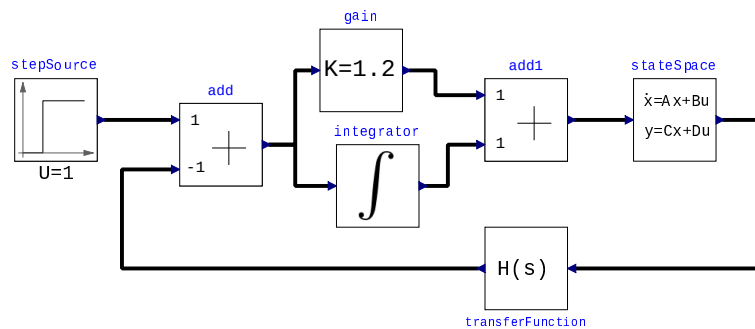


Figura 5.22: Sistema de Control en Diagrama de Bloques

5.3.2. Sistemas Muestreados

Muy frecuentemente los controladores se implementan mediante dispositivos digitales (computadoras, microcontroladores, etc.) para lo cual se muestrean ciertas señales continuas de la planta, se calcula la salida del controlador y luego se actúa sobre las entradas de la planta reteniendo los valores discretos del controlador entre intervalos de muestreo.

Esto puede fácilmente incorporarse en modelos de Modelica. El lenguaje cuenta con la posibilidad de definir variables *discretas* que sólo cambian cuando ocurre cierta condición. Además, cuenta con una función específica para que esta condición ocurra cada cierto intervalo de tiempo regular. Con esto, podemos por ejemplo constuir el modelo de un bloque de función de transferencia discreta como sigue:

Código 5.51: Bloque que modela una Función Transferencia Discreta

```

block DiscreteTransferFunction
  extends DSFLib.ControlSystems.Blocks.Interfaces.SISO;
  parameter Real num[:] = {1, -0.95} "Numerator coefficients";
  parameter Real den[:] = {1, -1} "Denominator coefficients";
  parameter Real T = 0.1 "Sample Period";
protected
  parameter Integer n=size(den,1);
  parameter Integer m=size(num,1);
  discrete Real ylast[n](each start=0);
  discrete Real ulast[n](each start=0);
equation
  y = ylast[1];
algorithm
  when sample(0, T) then
    ylast[2:n]:=ylast[1:n-1];
    ulast[2:n]:=ulast[1:n-1];
    ulast[1]:=u;
    ylast[1]:=-ylast[2:n]*den[2:n]/den[1]+ulast[n-m+1:n]*num[1:m]/den[1];
  end when;
  annotation(...);
end DiscreteTransferFunction;

```

Este modelo tiene dos vectores de variables discretas: uno que guarda las últimas entradas `ulast[:]` y otro que guarda las últimas salidas `ylast[:]`. Cada T segundos se torna verdadera la función `sample(0, T)` y en dichos momentos se actualizan estas variables discretas según la ecuación en diferencias que define la función transferencia discreta.

Notar que tanto la entrada u como la salida y del bloque son continuas, por lo que externamente este bloque es continuo (es un sistema muestreado). Por esto, podemos utilizarlo sin problemas en el contexto de modelos continuos como se muestra en el modelo de Modelica de la Figura 5.23.

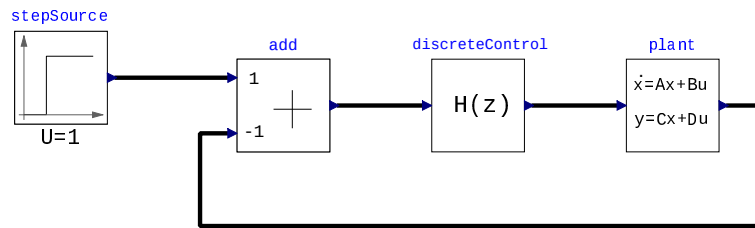


Figura 5.23: Sistema de Control Discreto en Diagrama de Bloques (vista de diagrama)

5.3.3. Sistemas con Eventos

Los sistemas muestreados pueden pensarse como un caso particular de sistemas en los cuales hay acciones asociadas a la ocurrencia de cierta condición. En ese caso, la condición es que haya transcurrido cierto intervalo de tiempo y entonces la acción es tomar una muestra y actualizar el control.

En casos más generales, la ocurrencia de eventos que desencadenan una acción de control (o un cambio en ciertas variables del modelo) se pueden representar fácilmente en Modelica utilizando el comando `when` en la sección `algorithm`.

Un ejemplo de este uso se puede ver en el Código 5.52 que implementa el bloque `Hysteresis` que cambia el valor de la salida cuando la entrada se torna mayor que cierto parámetro `u_on` y la vuelve a cambiar cuando se torna menor que otro parámetro `u_off`.

Código 5.52: Bloque que modela una función con histéresis

```

block Hysteresis
  extends DSFLib.ControlSystems.Blocks.Interfaces.SISO;
  discrete Boolean on;
  parameter Real u_on=1;
  parameter Real u_off=-1;
  parameter Real y_on=1;
  parameter Real y_off=-1;
  initial algorithm
    if u>u_on then
      on:=true;
    end if;
  equation
    y=if on then y_on else y_off;
  algorithm
    when u>u_on then
      on:=true;
    end when;
    when u<u_off then
      on:=false;
    end when;
  end Hysteresis

```

En dicho bloque, utilizamos una variable booleana (`on`) para representar el estado del bloque, y en función de dicho estado calculamos la salida. Los eventos se disparan cada vez que alguna condición (`u>u_on` o `u<u_off`) se torna verdadera. El algoritmo inicial es necesario además para asegurar que la variable `on` comience en el estado correcto ya que por omisión las variables booleanas toman el valor de `false`).

Usando este bloque, se puede modelar un control *on-off* con histéresis que es muy utilizado en la práctica (los acondicionadores de aire, por ejemplo, regulan la temperatura de esta manera) como se muestra en la Figura 5.24.

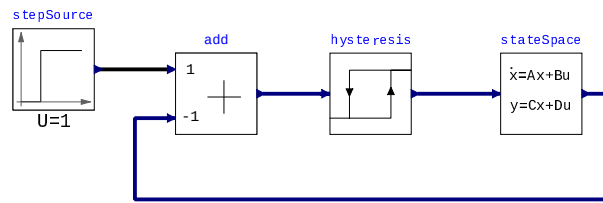


Figura 5.24: Sistema de Control Discreto en Diagrama de Bloques (vista de diagrama)

5.3.4. Sensores y Actuadores

Si bien utilizar diagramas de bloques simplifica en buena medida la representación de los sistemas de control, este tipo de modelos no es lo más apropiado para representar los subsistemas físicos que forman parte de los mismos. De hecho, construir un diagrama de bloques de sistemas multidominio con cierta complejidad puede llevar muchísimo tiempo y no tiene casi ningún beneficio. Usar en cambio modelado orientado a objetos es muchísimo más sencillo.

Lo más adecuado entonces es mantener la representación orientada a objetos de las partes físicas del modelo y utilizar diagramas de bloques para representar los controladores. Es claro que para esto necesitaremos componentes que hagan de interfaz entre los subsistemas correspondientes a los dominios físicos y los subsistemas donde utilizaremos diagramas de bloques.

Desde los modelos físicos hacia los diagramas de bloques necesitaremos medir ciertas magnitudes a través de componentes que llamaremos *sensores*. Estos sensores, que tendrán conectores físicos y conectores de salida para los bloques, serán ideales y no deberán alterar el funcionamiento del sistema físico (podremos igualmente representar sensores no ideales combinando sensores ideales con otros componentes).

Siguiendo estos principios, podemos entonces plantear los siguientes modelos de sensores para el dominio eléctrico:

Código 5.53: Modelos de Sensores para Circuitos Eléctricos

```

model VoltageSensor
  extends DSFLib.Circuits.Interfaces.OnePort;
  DSFLib.ControlSystems.Blocks.Interfaces.RealOutput y annotation(...);
equation
  y = v;
  i = 0;
  annotation(...);
end VoltageSensor;

model CurrentSensor
  extends DSFLib.Circuits.Interfaces.OnePort;
  DSFLib.ControlSystems.Blocks.Interfaces.RealOutput y annotation(...);
equation
  y = i;
  v = 0;
  annotation(...);
end CurrentSensor;

```

Estos sensores tienen todos dos conectores eléctricos (heredados de la clase `OnePort`) y miden la tensión entre ambos o la corriente.

De manera similar podemos construir modelos de sensores para los demás dominios. Por ejemplo, para los sistemas mecánicos rotacionales tenemos:

Código 5.54: Modelos de Sensores para Sistemas Mecánicos Rotacionales

```

model TorqueSensor
  extends DSFLib.Mechanical.Rotational.Interfaces.Compliant;
  DSFLib.ControlSystems.Blocks.Interfaces.RealOutput y annotation(...);

```

```

equation
  y = tau;
  phi_rel = 0;
  annotation(...);
end TorqueSensor;

model AngleSensor
  extends DSFLib.Mechanical.Rotational.Interfaces.Compliant;
  DSFLib.ControlSystems.Blocks.Interfaces.RealOutput y annotation(...);
equation
  y = phi_rel;
  tau = 0;
  annotation(...);
end AngleSensor;

model AngSpeedSensor
  extends DSFLib.Mechanical.Rotational.Interfaces.Compliant;
  DSFLib.ControlSystems.Blocks.Interfaces.RealOutput y annotation(...);
equation
  y = der(phi_rel);
  tau = 0;
  annotation(...);
end AngSpeedSensor;

```

Al igual que en el caso de los circuitos, estos sensores miden entre dos conectores mecánicos el torque o la velocidad relativa (o el ángulo entre los conectores).

Siguiendo los mismos principios se pueden construir sensores para los demás dominios mecánicos, hidráulicos y térmicos.

En cuanto a los actuadores, también los consideraremos ideales (si no lo fueran, los construiremos combinando componentes). Por lo tanto, serán fuentes ideales controladas en todos los dominios y podrán modelarse de igual manera que las fuentes ideales pero reemplazando el parámetro (o la función) que fija el valor por una señal de entrada que proviene de un conector de bloques.

Con esta idea, por ejemplo, los actuadores eléctricos serán fuentes de tensión y corriente controladas que pueden describirse como sigue:

Código 5.55: Modelos de Actuadores para Circuitos Eléctricos

```

import DSFLib.Circuits.Interfaces.*;

model ControlledVoltage
  extends OnePort;
  Blocks.Interfaces.RealInput u annotation(...);
equation
  v = u;
  annotation(...);
end ControlledVoltage;

model ControlledCurrent
  extends OnePort;
  DSFLib.ControlSystems.Blocks.Interfaces.RealInput u annotation(...);
equation
  i = -u;
  annotation(...);
end ControlledCurrent;

```

De manera similar pueden plantearse fuentes controladas para los demás dominios. Con estos componentes, es simple modelar sistemas de control donde un modelo físico interactúa con un sistema de control en Diagrama de Bloques. Por ejemplo, la Figura 5.25 muestra el modelo (en vista de Diagrama) de un motor de corriente continua con excitación independiente, con una señal de entrada que comanda la

tensión de armadura y una señal de salida que mide la velocidad angular. El bloque resultante, haciendo uso del modelado jerárquico, se coloca en el lazo de realimentación como se muestra en la Figura 5.26.

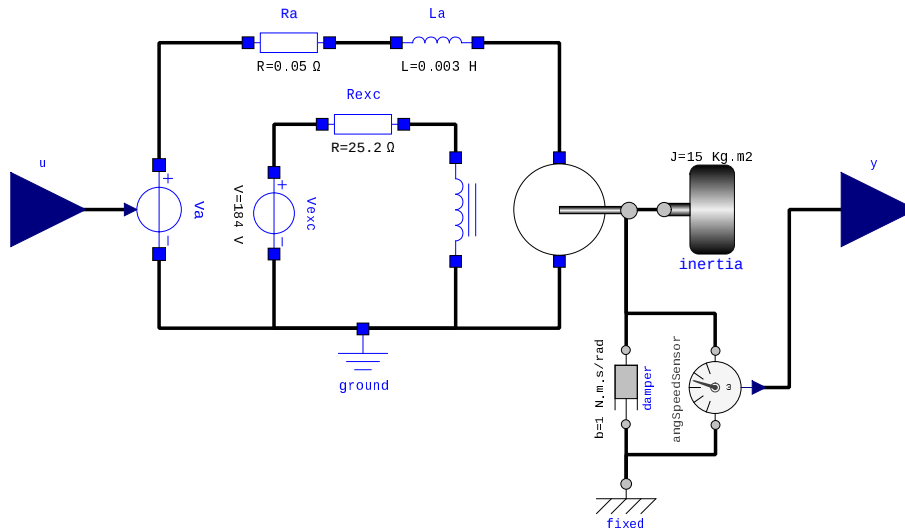


Figura 5.25: Motor de Corriente Continua con Entrada y Salida

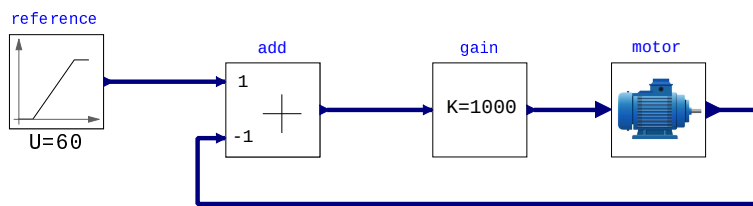


Figura 5.26: Modelo del Control de un Motor de Corriente Continua

5.4. DAE Definida por un Modelo Orientado a Objetos

El modelo orientado a objetos permite simplificar mucho la construcción del sistema de DAEs que representa el modelo. En primer lugar, al agregar un componente a un modelo, estamos en realidad agregando las ecuaciones algebraicas y diferenciales que conforman las relaciones constitutivas del mismo. Por otro lado, al conectar dos componentes mediante el comando `connect` estamos indirectamente escribiendo una serie de relaciones estructurales.

Por lo tanto, un modelo orientado a objetos no es más que una representación simplificada de un conjunto de ecuaciones diferenciales algebraicas. Dado entonces un modelo orientado a objetos podemos de manera simple encontrar el sistema de DAEs que representa:

- Las relaciones constitutivas serán las de cada componente. En las variables que intervienen en dichas ecuaciones deberemos agregar un prefijo que identifique el componente para evitar que dos variables de objetos distintos tengan el mismo nombre.
- Las relaciones estructurales se podrán obtener de las conexiones. Por cada conexión tendremos ecuaciones de igualdad asociadas a las variables que no son de flujo. Además, tendremos que detectar todas las conexiones que convergen en un mismo punto para agregar las ecuaciones de suma de flujos correspondientes.

Veamos entonces por ejemplo el modelo masa-resorte de la Figura 5.27.

Comenzaremos listando las relaciones constitutivas de los componentes que se encuentran en el Código 5.25:

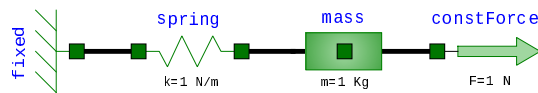


Figura 5.27: Sistema Masa Resorte en la vista de Diagrama.

- Para el punto fijo tenemos:

```
fixed.flange.s=fixed.s_0;
```

donde agregamos el prefijo `fixed` a la variable `flange.s` y al parámetro `s_0` para indicar que se trata de las variables y parámetros del objeto llamado `fixed`.

- Para la masa, en tanto, tenemos:

```
mass.s=mass.flange.s;
mass.f=mass.flange.f;
mass.m*der(mass.v)-mass.f=0;
der(mass.s)-mass.v=0;
```

donde al igual que antes, agregamos el prefijo con el nombre del objeto a las variables y parámetros.

- Para la fuerza constante las relaciones constitutivas son:

```
constForce.flange.f = -constForce.F;
```

- Para el resorte, por último, tenemos las relaciones constitutivas propias:

```
spring.f=spring.k*(spring.s_rel-spring.s_rel0);
```

a la que se deben agregar las relaciones constitutivas de la clase `Compliant` de la cual deriva el modelo del resorte:

```
spring.s_rel = spring.flange_b.s - spring.flange_a.s;
spring.flange_b.f = spring.f;
spring.flange_a.f = -spring.f;
```

En cuanto a las relaciones estructurales, las mismas surgen de las conexiones del modelo:

```
connect(fixed.flange, spring.flange_a);
connect(spring.flange_b, mass.flange);
connect(constForce.flange, mass.flange);
```

de donde habrá tres ecuaciones que igualan posiciones:

```
fixed.flange.s=spring.flange_a.s;
spring.flange_b.s=mass.flange.s;
constForce.flange.s=mass.flange.s;
```

y como hay dos nodos de conexión (uno asociado a cada lado del resorte) tendremos dos ecuaciones que sumen fuerzas:

```
fixed.flange.f+spring.flange_a.f=0;
spring.flange_b.f+mass.flange.f+constForce.flange.f=0;
```

Finalmente, colocando todas las relaciones constitutivas y estructurales juntas tenemos el modelo completo en forma de sistema de DAEs:

Código 5.56: Sistema de DAEs para el sistema masa-resorte de la Figura 5.27

```
fixed.flange.s=fixed.s_0;

mass.s=mass.flange.s;
mass.f=mass.flange.f;
```

```

mass.m*der(mass.v)-mass.f=0;
der(mass.s)-mass.v=0;

constForce.flange.f = -constForce.F;

spring.f=spring.k*(spring.s_rel-spring.s_rel0);
spring.s_rel = spring.flange_b.s - spring.flange_a.s;
spring.flange_b.f = spring.f;
spring.flange_a.f = -spring.f;

fixed.flange.s=spring.flange_a.s;
spring.flange_b.s=mass.flange.s;
constForce.flange.s=mass.flange.s;

fixed.flange.f+spring.flange_a.f=0;
spring.flange_b+mass.flange.f+constForce.flange.f=0;

```

Como puede verse, la obtención de la DAE a partir del modelo orientado a objetos es totalmente sistemática y, particularmente, las relaciones estructurales son triviales (lo que era un problema cuando intentábamos directamente obtener la DAE en el Capítulo 4).

Utilizando OpenModelica se puede obtener directamente el listado de DAEs con el comando `Instantiate Model`, que brinda tanto las DAEs como la definición de los parámetros y variables en la forma de un modelo *plano* (es decir, sin clases ni objetos). Para el caso de este sistema masa-resorte, dicho comando nos brinda el siguiente resultado:

Código 5.57: Salida del comando `Instantiate Model` para el sistema masa-resorte de la Figura 5.27

```

class DSFLib.Mechanical.Translational.Examples.SpringMassForce
  Real fixed.flange.s(unit = "m");
  Real fixed.flange.f(unit = "N");
  parameter Real fixed.s_0(unit = "m") = 0.0;
  Real mass.flange.s(unit = "m");
  Real mass.flange.f(unit = "N");
  parameter Real mass.m(unit = "Kg") = 1.0;
  Real mass.s(unit = "m");
  Real mass.v(unit = "m/s");
  Real mass.f(unit = "N");
  Real spring.flange_b.s(unit = "m");
  Real spring.flange_b.f(unit = "N");
  Real spring.flange_a.s(unit = "m");
  Real spring.flange_a.f(unit = "N");
  Real spring.s_rel(unit = "m");
  Real spring.f(unit = "N");
  parameter Real spring.k(unit = "N/m") = 1.0;
  parameter Real spring.s_rel0(unit = "m") = 0.0;
  Real constForce.flange.s(unit = "m");
  Real constForce.flange.f(unit = "N");
  parameter Real constForce.F(unit = "N") = 1.0;
equation
  fixed.flange.s = fixed.s_0;
  mass.s = mass.flange.s;
  mass.f = mass.flange.f;
  mass.m * der(mass.v) - mass.f = 0.0;
  der(mass.s) - mass.v = 0.0;
  spring.f = spring.k * (spring.s_rel - spring.s_rel0);
  spring.s_rel = spring.flange_b.s - spring.flange_a.s;
  spring.flange_b.f = spring.f;
  spring.flange_a.f = -spring.f;
  constForce.flange.f = -constForce.F;

```

```

fixed.flange.f + spring.flange_a.f = 0.0;
mass.flange.f + spring.flange_b.f + constForce.flange.f = 0.0;
fixed.flange.s = spring.flange_a.s;
constForce.flange.s = mass.flange.s;
constForce.flange.s = spring.flange_b.s;
end DSFLib.Mechanical.Translational.Examples.SpringMassForce;

```

5.5. La Librería Estándar de Modelica

Los modelos que fuimos construyendo en este capítulo son versiones muy simplificadas de componentes que forman parte de la librería estándar de Modelica (MSL, por *Modelica Standard Library*). La MSL es una librería abierta y gratuita que contiene paquetes de componentes de diversos dominios, y que es desarrollada y mantenida por la Modelica Association, un consorcio de universidades y de empresas que está a cargo no sólo de esto, sino de la definición del propio lenguaje. Además de los componentes, la librería cuenta con una capa de documentación (la documentación en Modelica tiene una representación estandarizada) y cuenta con numerosos ejemplos de sistemas de diversa complejidad.

Entre los principales paquetes con los que cuenta la librería se pueden mencionar:

Sistemas Mecánicos: Incluye componentes traslacionales, rotacionales y multicuerpos (mecánica 3D).

En el último caso, los componentes cuentan además con una capa de animación lo que permite visualizar los resultados en animaciones 3D. Esta capa de animación se puede en realidad utilizar en cualquier modelo o librería que uno construya (la animación también tiene anotaciones estandarizadas en Modelica).

Sistemas Eléctricos: Incluye componentes analógicos, digitales, máquinas (motores principalmente), elementos polifásicos y convertidores de potencia. Hay además una implementación de los componentes de Spice.

Dinámica de Fluidos: Tiene componentes que vinculan el movimiento de los fluidos con sus propiedades termodinámicas, de manera que permite representar tanto fluidos incompresibles como gases y fluidos compresibles en general.

Sistemas Térmicos: Contiene componentes que modelan la transferencia de calor (similar a la librería de sistemas térmicos que desarrollamos en la Sección 5.2.4) y componentes más complejos que tienen en cuenta la termodinámica en interacción con fluidos.

Diagramas de Bloques: Incorpora bloques similares a los que vimos en la Sección 5.3.1, pero con mucho más componentes y además incluye bloques que utilizan señales discretas, booleanas, etc.

Sistemas Magnéticos: Tiene componentes que modelan fenómenos que involucran campos magnéticos y su interacción con dominios eléctricos y mecánicos (solenoides, elementos de bobina móvil, máquinas asíncronas, etc.)

State Graphs: Permiten modelar *Sistemas de Eventos Discretos* utilizando este formalismo, similar al de las Redes de Petri.

Además de estos paquetes, la librería contiene paquetes con funciones matemáticas, unidades de medida, y diversas utilidades que permiten leer y escribir archivos desde los modelos, entre otras cosas.

Además de la MSL existen diversas librerías muchas de ellas gratuitas y otras comerciales, desarrolladas por la comunidad de modelado y simulación (o en algunos casos por empresas de software) y que contienen componentes para modelar desde automóviles hasta sistemas químicos pasando por modelos de edificios, aviones, y una vasta variedad de aplicaciones en los diversos dominios de la ciencia y la ingeniería. Las principales librerías disponibles están listadas en la página <https://modelica.org/libraries.html>.

5.6. Bibliografía Complementaria

Las principales referencias sobre el lenguaje Modelica y las técnicas de modelado orientado a objetos son los libros de Peter Fitzson, particularmente los libros [Fri14] y [Fri15], este último en castellano y disponible gratuitamente para descargar desde <http://www.diva-portal.org/smash/get/diva2:853769/fulltext01.pdf>.

Anualmente se realiza además la conferencia de Modelica en la cual todos los trabajos se publican y se encuentran disponibles para descargar gratuitamente desde la página <https://modelica.org/publications.html>. Desde dicha página es posible también descargar numerosas tesis de maestría y doctorales vinculadas con Modelica y hay un listado de los principales libros que tratan sobre distintos temas vinculados al lenguaje.

5.7. Problemas Propuestos

[P5.1] Convertidor Buck

La Figura 5.28 muestra el circuito correspondiente a un convertidor de continua en continua denominado *Buck* o *reductor*.

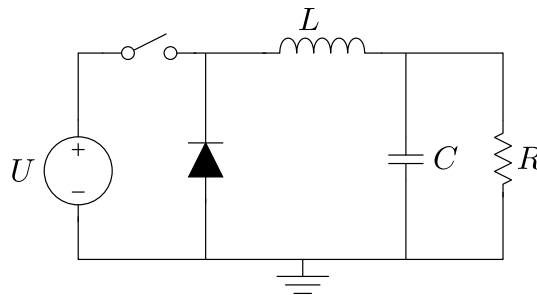


Figura 5.28: Convertidor Buck.

1. Utilizando los componentes de la librería `Circuits` desarrollada en la Sección 5.2.1, escribir el modelo orientado a objetos correspondiente.
2. Simular el modelo en OpenModelica. Suponer que los parámetros son los siguientes: $L = 10^{-4}\text{Hy}$, $C = 10^{-4}\text{F}$, $U = 12\text{V}$, $R = 10\Omega$. Para la llave, utilizar un período de $T = 10^{-4}$ segundos y un *duty cycle* del 50 %.
3. Escribir las relaciones estructurales correspondientes al nodo donde está la puesta a tierra.

[P5.2] Rectificador de Media Onda

El circuito de la Figura 5.29 es un rectificador de media onda que alimenta una carga resistiva. Para poder modelarlo, se pide:

1. Desarrollar el modelo en Modelica de la fuente de voltaje sinusoidal, usando como base el modelo parcial de un dipolo del Código 5.9. Este componente deberá tener dos parámetros: amplitud (en Volts) y frecuencia (en Hz.).
2. Desarrollar el modelo en Modelica del transformador, de manera que tenga 4 conectores (dos positivos y dos negativos) y que satisfaga las siguientes ecuaciones:

$$\begin{aligned} v_1(t) &= L_1 \frac{di_1(t)}{dt} + M \frac{di_2(t)}{dt} \\ v_2(t) &= M \frac{di_1(t)}{dt} + L_2 \frac{di_2(t)}{dt} \end{aligned} \tag{P5.2a}$$

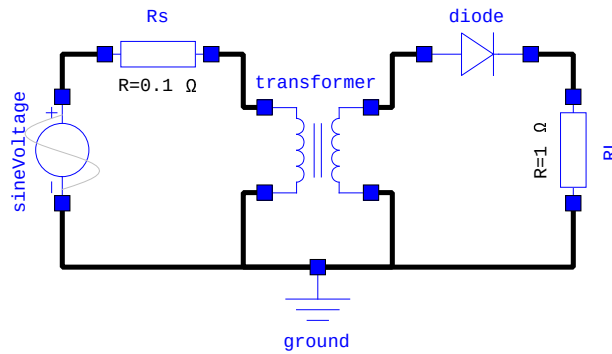


Figura 5.29: Rectificador de Media Onda.

donde $v_1(t)$ y $v_2(t)$ son los voltajes en el primario y secundario, $i_1(t)$ e $i_2(t)$ son las corrientes respectivas y los parámetros L_1 y L_2 son la inductancia primaria y secundaria mientras que M es la inductancia mutua.

3. Construir con estos elementos el modelo orientado a objetos del circuito, describiendo los componentes y las conexiones.
4. Simular el circuito en OpenModelica utilizando parámetros $L_1 = 0.02\text{Hy}$, $L_2 = 0.1\text{Hy}$, y $M = 0.04\text{Hy}$ para el transformador. Además, usar $V = 311\text{V}$ y $f = 50\text{Hz}$ para la fuente.
5. Observar que ocurre si se toman parámetros $L_1 = L_2 = M = 0.1\text{Hy}$. Explicar lo observado a partir de la Ecuación (P5.2a).

[P5.3] Sistema con Palanca

La Figura 5.30 representa un sistema mecánico traslacional que contiene un componente que modela una palanca.

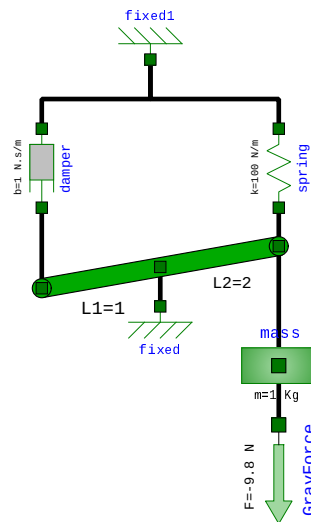


Figura 5.30: Sistema con Palanca.

1. Analizar el orden del modelo y proponer un conjunto minimal de variables de estado.
2. Construir el modelo de la palanca teniendo en cuenta lo siguiente:

- El componente tiene tres conectores **Flange** a distancias L_1 y L_2 entre sí (el largo total es $L_1 + L_2$). Las distancias L_1 y L_2 serán los parámetros del sistema.
- Se consideran sólo pequeños movimientos, por lo que puede considerarse que

$$L_1 (s_b(t) - s_c(t)) = L_2 (s_c(t) - s_a(t)) \tag{P5.3a}$$

siendo s_a , s_b y s_c la posición de los tres conectores (s_c es la del conector del medio).

- La palanca no tiene masa, por lo que las fuerzas de los tres conectores deberán sumar cero.
- Tampoco hay momento de inercia, por lo que la suma de torques respecto al conector central deberá ser cero. Esto es,

$$L_2 f_a(t) - L_1 f_b(t) = 0; \tag{P5.3b}$$

ya que dos fuerzas con igual sentido producen torques con sentidos opuestos respecto al centro.

3. Usando este modelo, construir el modelo del sistema completo, describiendo los componentes y sus conexiones.
4. Utilizando OpenModelica simular el sistema con los parámetros de la Figura 5.30, considerando además que la posición del punto fijo que representa el techo es de 2 mts y la del piso es 0. Suponer que la longitud en reposo del resorte es también de 2 mts.

[P5.4] Sistema con Polea .

El esquema roto-traslacional de la Figura 5.31 contiene una polea simple.

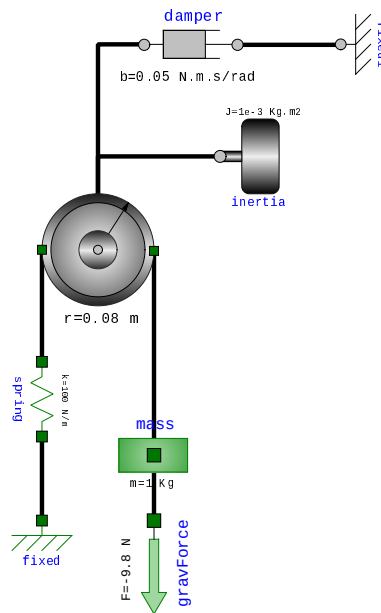


Figura 5.31: Sistema con Polea

1. Analizar el orden del modelo y proponer un conjunto minimal de variables de estado.
2. Construir el modelo de la polea teniendo en cuenta lo siguiente:
 - El componente tiene dos conectores traslacionales y uno rotacional.
 - Hay dos parámetros: r (radio de la polea) y s_0 (posición media de los conectores traslacionales), por lo que se cumplirá

$$s_0 = (s_b + s_a)/2 \tag{P5.4a}$$

siendo s_a y s_b la posición de los conectores izquierdo y derecho respectivamente.

- Cuando el ángulo del conector rotacional es cero, asumimos que la posición de ambos conectores traslacionales es igual, por lo tanto:

$$r \phi = (s_b - s_a)/2 \tag{P5.4b}$$

- El componente no tiene momento de inercia, por lo que la suma de torques será nula:

$$r f_a - r f_b = \tau \tag{P5.4c}$$

siendo f_a y f_b la fuerza en los conectores izquierdo y derecho y τ el torque en el conector rotacional.

3. Utilizando este componente, plantear el modelo completo mencionando sus restantes componentes y las conexiones.
4. Simular el sistema en OpenModelica.

[P5.5] Sistema Biela-Manivela

El esquema roto-traslacional de la Figura 5.32 contiene un componente Biela-Manivela.

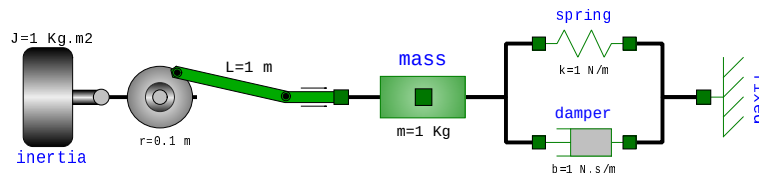


Figura 5.32: Sistema Biela-Manivela

1. Analizar el orden del modelo y proponer un conjunto minimal de variables de estado.
2. Construir el modelo del componente Biela-Manivela teniendo en cuenta las siguientes consideraciones:
 - El componente tiene un conector traslacional y uno rotacional.
 - Hay tres parámetros: la longitud de la biela L , el radio de la manivela r y la posición del centro de la manivela s_0 .
 - Para relacionar la posición y el ángulo de los conectores se puede usar la siguiente ecuación:

$$s = r \cos(\phi) + L \sqrt{1 - \left(\frac{r}{L} \sin(\phi)\right)^2} \tag{P5.5a}$$

tal que $s + s_0$ es la posición del conector traslacional y ϕ es el ángulo del conector rotacional.

- Para relacionar la fuerza y el torque de los conectores, se puede usar:

$$\tau (s - r \cos(\phi)) = f (r s \sin(\phi)) \tag{P5.5b}$$

donde τ es el torque del conector rotacional y f la fuerza del translacional.

3. Utilizando este componente, plantear el modelo completo mencionando sus restantes componentes y las conexiones.
4. Simular utilizando OpenModelica con los parámetros indicados en la figura suponiendo además que el punto fijo de la derecha está en la posición $s_0 = 1m$ y que la condición inicial de velocidad en la inercia es de 1 rad./seg.

De haber problemas de causalización, agregar como variable la velocidad angular al sistema biela manivela mediante una ecuación `der(phi)=w` declarando `Real phi(stateSelect=StateSelect.prefer)` para utilizar preferentemente esta variable como estado.

[P5.6] Vibraciones de una Carga Desbalanceada

El modelo de la Figura 5.33 está pensado para estudiar el efecto de una carga rotacional desbalanceada acoplada al eje de un motor. En este caso, el motor está modelado como una fuente de torque con su momento de inercia J , masa total M y rozamiento en el eje. El eje del motor se puede desplazar verticalmente, y la estructura está apoyada sobre una base, cuyo vínculo se representa a través de un resorte y del rozamiento. La carga desbalanceada es una masa m que se mueve en dos dimensiones rotando a una distancia r del eje.

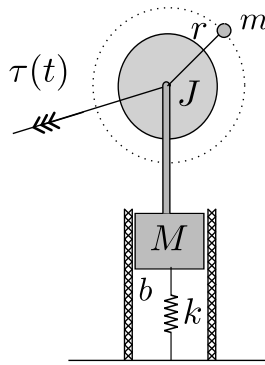


Figura 5.33: Sistema con Carga Desbalanceada

1. Analizar el orden del modelo y proponer un conjunto minimal de variables de estado.
2. Construir el modelo de la fuente de torque constante, utilizando como base el modelo parcial Compliant rotacional de la librería DSFLib.
3. Construir el modelo completo utilizando los parámetros de la figura suponiendo que la masa que representa el desbalance gira a 0.1m del eje.
4. Analizar como varía la amplitud de las oscilaciones en la base (deformación del resorte) cuando cambia el radio de giro de la masa.

[P5.7] Puente Grúa

El esquema roto-traslacional-planar de la Figura 5.34 contiene un modelo de un puente grúa con el cable retráctil. Dicho cable tiene un componente rotacional donde se enrolla de manera que puede alargarse o retraerse.

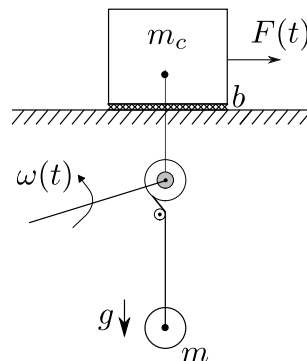


Figura 5.34: Puente Grúa con Cable Retráctil

1. Analizar el orden del modelo y proponer un conjunto minimal de variables de estado.

2. Construir el modelo del cable retráctil teniendo en cuenta las siguientes consideraciones:
 - Puede utilizarse una articulación prismática como base para representar el movimiento longitudinal del mismo.
 - Entre los conectores traslacionales de esta articulación deberá haber un elemento rototraslacional con dos conectores traslacionales y uno rotacional que represente el carrete donde se enrolla el cable.
 - Dicho elemento rototraslacional, que puede extenderse de la clase `Compliant`, tendrá dos parámetros: la longitud del cable L_0 cuando el ángulo del conector rotacional es 0 y el radio r del carrete.
 - Para relacionar la longitud del cable y el ángulo del conector se puede usar la ecuación $l = L_0 - r \cdot \text{flange_r} \cdot \text{phi}$.
 - Para relacionar la fuerza que transmite el cable en sentido longitudinal y el torque del conector rotacional se puede usar $\text{flange_r} \cdot \text{tau} = f \cdot r$.
3. Construir el modelo de una fuente ideal de velocidad angular.
4. Utilizando estos componentes, plantear el modelo completo mencionando sus restantes componentes y las conexiones.
5. Simular utilizando OpenModelica con los parámetros $m_c = 250\text{Kg}$, $b = 1\text{Ns/m}$, $m = 1000\text{Kg}$, $F = 1000\text{N}$ y suponiendo además que la longitud del cable con ángulo de retracción cero es $L_0 = 1\text{mts}$ y que el radio es $r = 0.01\text{m}$.

[P5.8] Sistema Hidráulico

Considerar nuevamente el modelo de la Figura 5.15. En el mismo reemplazar la válvula lineal por otra no lineal cuya relación constitutiva esté dada por la ley:

$$Q(t) - G \sqrt{|\Delta P(t)|} \text{sign}(\Delta P(t)) = 0. \tag{P5.8a}$$

para cierto parámetro G . Definir el nuevo componente y simular el sistema.

[P5.9] Sistema Térmico con Radiación

Considerar el modelo de la Figura 5.35.

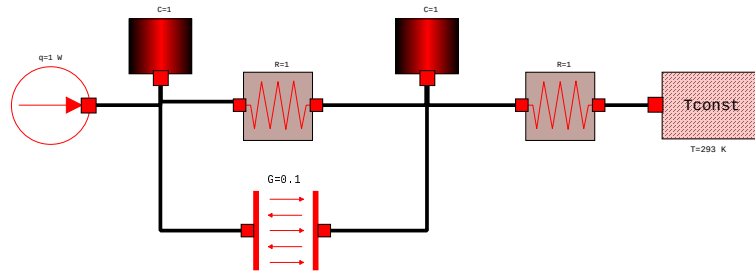


Figura 5.35: Modelo Térmico con Radiación.

1. Analizar el orden y proponer un juego de variables de estado minimal.
2. Construir el modelo correspondiente a la radiación entre dos superficies, teniendo en cuenta lo siguiente:
 - El flujo de calor por radiación entre dos superficies (desde la superficie b hacia la a), bajo ciertas hipótesis resulta:

$$q_{ba} = G \sigma (T_b^4 - T_a^4) \tag{P5.9a}$$
 donde G es un parámetro (dependiente de la geometría y de los materiales de las superficies), $\sigma = 5.67 \cdot 10^8$ es la constante de Stefan–Boltzmann.

- Puede extenderse el modelo parcial térmico TwoPort para calcular el flujo según la ecuación anterior.
3. Plantear el modelo completo del sistema y simular para distintos valores del parámetro G .

[P5.10] Modelo de un Montacargas

La Figura 5.36 muestra el esquema de un montacargas impulsado por un motor de corriente continua con excitación independiente.

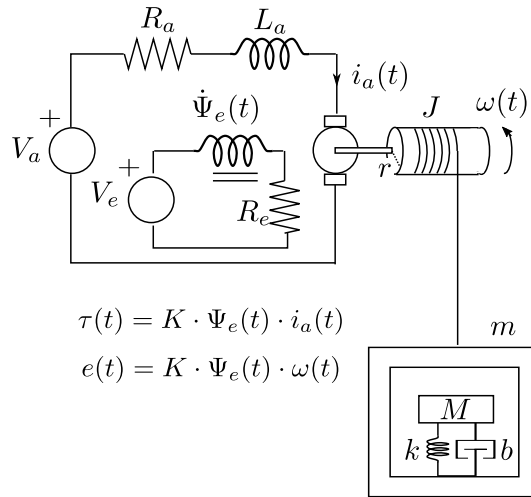


Figura 5.36: Esquema de un Montacargas

1. Escribir un modelo orientado a objetos que represente el subsistema del motor y que tenga un conector mecánico rotacional para que pueda conectarse con el resto del sistema.
2. Escribir un modelo orientado a objetos que represente el subsistema mecánico traslacional del montacargas (sin el cable) que tenga un conector mecánico traslacional para poder conectarlo con el resto.
3. Usando los dos modelos anteriores y un modelo rototraslacional apropiado formar el modelo completo.

[P5.11] Bomba Aspirante-Impelente

Considerar el sistema de la Figura 5.37, donde la posición del pistón $x(t)$ sigue una evolución sinusoidal conocida.

1. Construir una clase de Modelica que represente una válvula de una vía que tenga las siguientes relaciones constitutivas:

$$\Delta P(t) = \begin{cases} R_{\text{on}} q(t) & \text{si } q(t) > 0 \\ R_{\text{off}} q(t) & \text{si } q(t) \leq 0 \end{cases} \quad (\text{P5.11a})$$

con $R_{\text{on}} = 10^{-6}$ y $R_{\text{off}} = 10^{12}$.

2. Construir una clase que represente una fuente de posición sinusoidal que tenga como parámetros la amplitud y la frecuencia.
3. Utilizando estos nuevos componentes, construir un modelo del sistema completo.

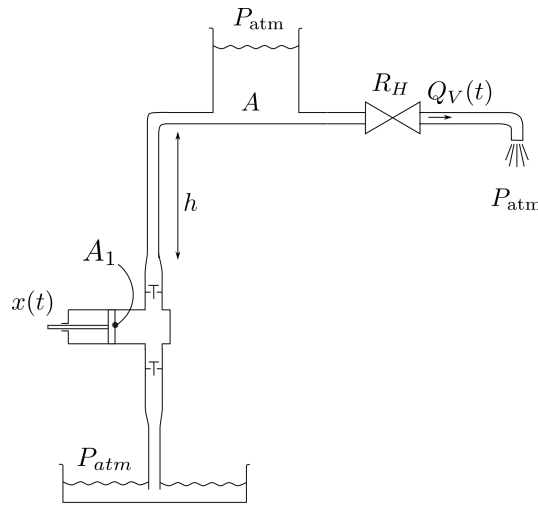


Figura 5.37: Sistema con una Bomba Aspirante Impelente

4. Simular el sistema en OpenModelica suponiendo que el área del pistón es $A_1 = 0.01m^2$, la resistencia hidráulica $R_H = 10^6 \text{ Pa s/m}^3$ y la posición del pistón sigue la ley $x(t) = 0.01 \sin(2 \pi f t)$ con $f = 10\text{Hz}$. Usar el resto de los componentes con los valores por defecto de la DSFLib.

[P5.12] Bomba Aspirante-Impelente II

El esquema de la Figura 5.38 muestra una versión más completa del sistema del Problema P5.11, donde el movimiento del pistón se realiza a través de un biela-manivela impulsado por un motor de corriente continua.

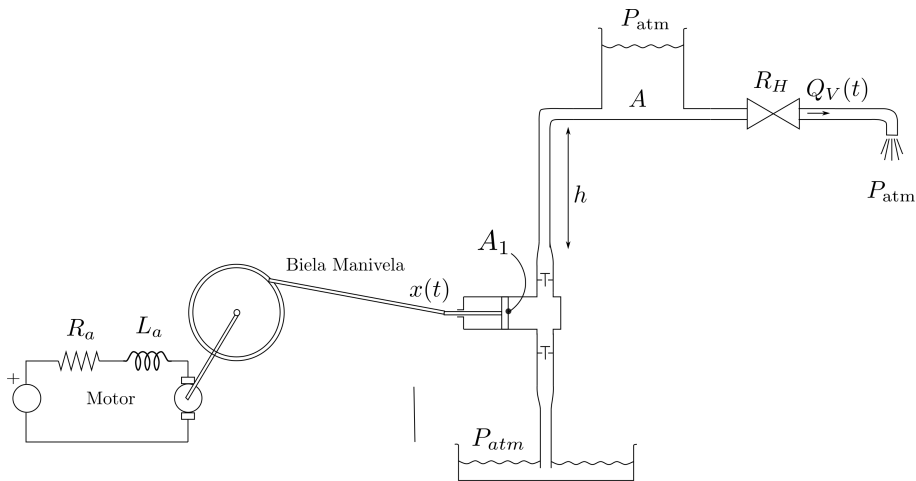


Figura 5.38: Sistema con una Bomba Aspirante Impelente

1. Construir una clase de Modelica correspondiente al motor, que tenga como interfaz un único conector rotacional asociado al eje del motor. El motor deberá contener, además de lo que se ve en la figura, una inercia.
2. Construir una clase de Modelica correspondiente a la bomba, que incluya el sistema biela-manivela, el pistón-cilindro y las dos válvulas de una vía. Dicho modelo deberá tener como interfaces un conector rotacional (asociada al eje de la manivela) y dos conectores hidráulicos asociados a la

entrada de la válvula de entrada y a la salida de las válvulas de salida. Utilizar para el sistema biela manivela el componente desarrollado en el Problema P5.5

3. Utilizando estas dos clases, construir un modelo del esquema completo.
4. Simular el sistema utilizando los mismos parámetros del Problema P5.11 y además suponiendo que la longitud de la biela es $L = 0.1\text{m}$, que el radio de la manivela es $r = 0.01\text{m}$ y que los parámetros del motor son $J = 1 \text{ Kg/m}^2$, $L_a = 10^{-3}\text{Hy}$, $R_a = 0.1\Omega$, $K = 1 \text{ V s/rad}$ y $U_a = 120\text{V}$.

[P5.13] Colector Solar

Considerar el modelo del colector solar de la Figura 5.20.

1. Reescribir el modelo suponiendo ahora que el caño donde circula el líquido está dividido en dos secciones con temperaturas distintas (formando dos recintos), pero que el colector es un único recinto.
2. Proponer un subsistema para cada sección de caño que tenga como interfaces dos puertos hidráulicos y dos puertos térmicos.
3. Reescribir ahora el modelo suponiendo que tanto el caño como el colector están divididos en 2 secciones.
4. Proponer un subsistema que agrupe una sección de caño y de colector, de manera que cada sección tenga como interfaces dos puertos hidráulicos y tres puertos térmicos (agregando el correspondiente al recinto del colector).

[P5.14] Control de Tensión en una Fuente Buck

La Figura 5.39 muestra el esquema de un control de tensión para un convertidor Buck. En el mismo, la tensión de referencia de 6V se compara con la salida y se implementa un control Proporcional-Integral (PI) que calcula el ciclo de trabajo (*duty cycle*) del convertidor. El ciclo de trabajo es la entrada de un bloque que genera una señal PWM (*Pulse Width Modulation*) que controla la llave del convertidor.

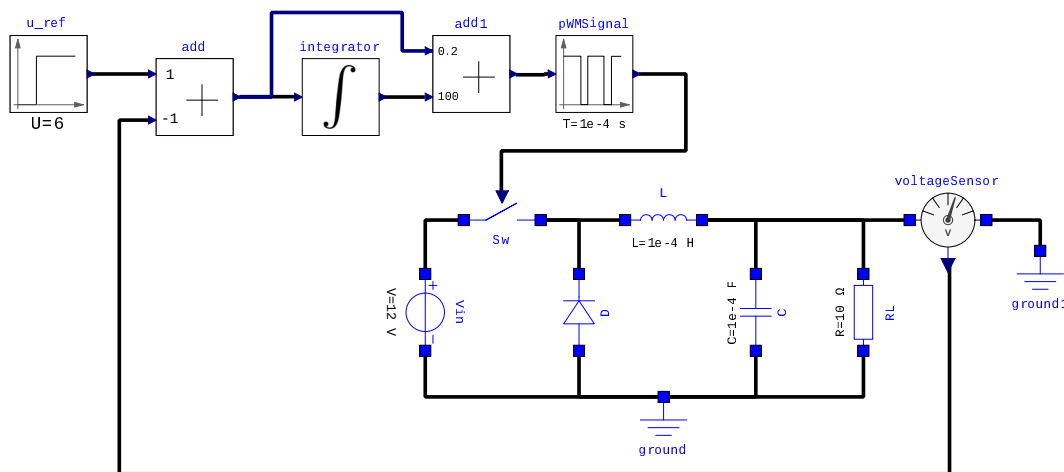


Figura 5.39: Control de Tensión en un Convertidor Buck

1. Modificar el modelo de la llave para que la señal que controla la posición de la misma sea una entrada proveniente de un Diagrama de Bloques.
2. Construir un bloque que genere la señal PWM a partir de una entrada, utilizando el siguiente código:

Código 5.58: Bloque que Genera una Señal PWM

```

block PWMSignal
  extends DSFLib.ControlSystems.Blocks.Interfaces.SISO;
  parameter Real T(unit = "s") = 1 "Period" ;
  discrete Real yd;
  discrete Real dc;
  discrete Real tlast;
equation
  y = yd;

algorithm
  when sample(0, T) then
    yd := 1;
    dc := min(max(u,1e-5),0.99999);
    tlast := time;
  end when;

  when time > tlast + dc * T then
    yd := 0;
  end when;
end PWMSignal;

```

3. Construir el modelo completo.
4. Agrupar el modelo en tres bloques, uno para la referencia, otro para el controlador y otro para la planta.
5. Simular usando OpenModelica con los parámetros indicados en la figura hasta un tiempo final $t_f = 0.01$. Observar que ocurre al variar las ganancias del control PI.

[P5.15] Control de Nivel de un Tanque

El esquema de la Figura 5.40 representa un sistema de control de nivel en un tanque. En el mismo, un control proporcional compara el nivel del tanque con el de una referencia (de un metro). La diferencia entre ambos niveles regula la apertura de la válvula por la que entra el agua al tanque. El sistema tiene además una válvula de salida que representa el consumo del usuario.

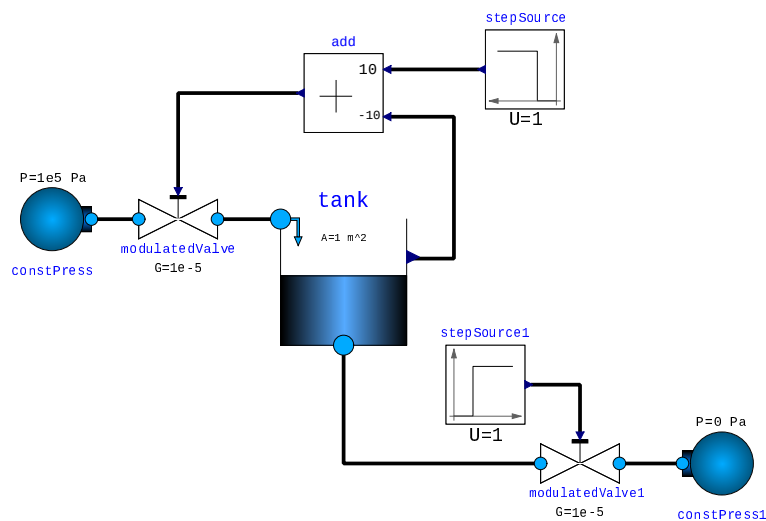


Figura 5.40: Control de Nivel de un Tanque

1. Modificar el modelo del tanque agregando un conector hidráulico que represente una entrada independiente de agua a presión atmosférica (que puede considerarse nula). Agregar además al componente una salida de Diagrama de Bloques que calcule el nivel de agua en el tanque $h(t)$.
2. Modificar el modelo de la válvula agregando una señal de entrada $u(t)$ proveniente de un Diagrama de Bloques que modele la apertura de la válvula. Utilizar la siguiente ley para el componente:

$$q(t) = \min(\max(u(t), 0), 1) G \sqrt{|p(t)|} \text{sign}(p(t)) \quad (\text{P5.15a})$$

de manera que el valor de apertura saturate entre 0 y 1 y donde G es un parámetro.

3. Construir el modelo completo utilizando los nuevos componentes.
4. Simular en OpenModelica con los parámetros indicados en la figura, suponiendo que el escalón de referencia de nivel se da en tiempo $t = 0$ y que el escalón de apertura de la válvula de salida se da en tiempo $t = 500\text{s}$.

[P5.16] Refrigeración de un Motor

La Figura 5.41 esquematiza el sistema de refrigeración de un motor. En este sistema circula agua impulsada por una bomba cuyo caudal se regula mediante una válvula controlada. El agua circula entre el motor (al que enfría) y el radiador.

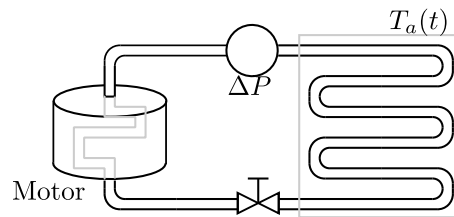


Figura 5.41: Sistema de Refrigeración de un Motor

Consideraremos las siguientes hipótesis:

- La bomba impone una diferencia de presión ΔP constante.
- La válvula sigue la ley de la Ecuación (P5.15a).
- El motor tiene temperatura uniforme y una capacidad térmica C_m .
- En el motor se genera un flujo de calor q_m constante.
- El agua tiene temperatura uniforme dentro del motor. La capacidad térmica de la masa de agua en el motor es C_{wm} .
- El agua tiene temperatura uniforme dentro del radiador. La capacidad térmica de la masa de agua en el radiador es C_{wr} .
- Hay una resistencia térmica R_{mw} entre el motor y el agua.
- Hay una resistencia térmica R_{wa} entre el agua y el ambiente a través del radiador.
- Consideramos despreciable la capacidad térmica del radiador.
- La apertura de la válvula $u(t)$ es proporcional a la temperatura medida en el motor (en grados centígrados). A 0°C la válvula está totalmente cerrada y a 100°C está totalmente abierta.

Se pide entonces

1. Analizar el orden del modelo y proponer un vector minimal de variables de estado.

2. Construir un modelo orientado a objetos del sistema completo.
3. Simular el sistema en OpenModelica considerando los siguientes parámetros:
 - Capacidades térmicas: $C_{wr} = C_{wm} = 10^4$, $C_m = 5 \cdot 10^4$.
 - Resistencias térmicas: $R_{mw} = 0.001$, $R_{wa} = 0.004$.
 - Flujo de calor entrante al motor: $q_m = 10^4 \text{W}$.
 - Presión de la bomba: $\Delta P = 1000 \text{Pa}$.
 - Parámetro de la válvula controlada: $G = 10^{-5}$

[P5.17] Relaciones Estructurales

En relación a los problemas anteriores, se pide:

1. Escribir las relaciones estructurales asociadas al conector de la masa del modelo de la Figura 5.30 del Problema P5.3.
2. Repetir para el conector de la inercia de la Figura 5.31 del Problema P5.4.
3. Ídem para el conector de la masa de la Figura 5.33 del Problema P5.6.

Capítulo 6

Modelado Mediante Formalismos Energéticos

En este capítulo veremos dos técnicas de modelado que se basan en principios energéticos. Estas técnicas permiten describir las relaciones constitutivas y estructurales de un sistema a partir de los fenómenos energéticos (almacenamiento, disipación, conversión y generación) que ocurren en el mismo.

Veremos en primer lugar la técnica de los *Bond Graphs* (o Diagramas de Enlace), que brinda un lenguaje gráfico unificado para distintos dominios con símbolos que representan los distintos fenómenos energéticos y las relaciones estructurales entre las magnitudes que intervienen en los mismos.

Luego veremos las *Ecuaciones de Euler-Lagrange* que constituyen una herramienta que sistematiza y simplifica la obtención de sistemas de ecuaciones diferenciales. Si bien este enfoque puede usarse en distintos dominios, en este curso nos enfocaremos específicamente para el caso particular de los sistemas mecánicos.

6.1. Introducción a los Bond Graphs

Los Bond Graphs o Diagramas de Enlace son un lenguaje gráfico para representar modelos de sistemas físicos de diferentes dominios a partir de los fenómenos energéticos que ocurren en los mismos. Este formalismo fue introducido por Henry Paynter en la década de 1960 y permite formular de manera gráfica y acausal las relaciones constitutivas y estructurales de un sistema físico.

6.1.1. Componentes Elementales

El Bond o Enlace

El componente más elemental de un Bond Graph es el *Bond* o Enlace (se suele llamar también Power Bond o Enlace de Potencia), que representa la transmisión de potencia entre dos partes de un sistema. La transmisión de energía se realiza a través de dos variables (llamadas variables de potencia) denominadas esfuerzo $e(t)$ y flujo $f(t)$ cuyo producto da por resultado potencia $p = e f$. El Bond se representa mediante un arpón dirigido en el sentido positivo de la transmisión de potencia, como se muestra en la Figura 6.1, donde se representa la transmisión de potencia desde el Sistema A hacia el Sistema B (esto es, cuando e y f son positivas la potencia se transmite en dicha dirección).

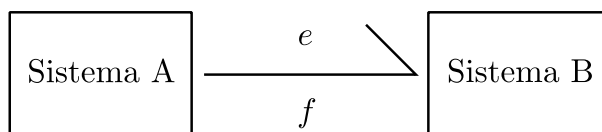


Figura 6.1: Un Enlace o *Bond* con sus variables de potencia e (esfuerzo) y f (flujo).

En el dominio eléctrico, por ejemplo, se representa como variable de esfuerzo la tensión y como variable de flujo la corriente. Luego veremos la generalización a otros dominios.

Elementos Estructurales

Hay dos componentes estructurales básicos, denominados Vínculo 0 y Vínculo 1 (0-Junction y 1-Junction) que representan el flujo de potencia entre subsistemas.

El Vínculo 0 representa la distribución de potencia con el esfuerzo como variable común a todos los subsistemas conectados. La Figura 6.2, por ejemplo, muestra un Vínculo 0 que transmite potencia entre cuatro subsistemas.

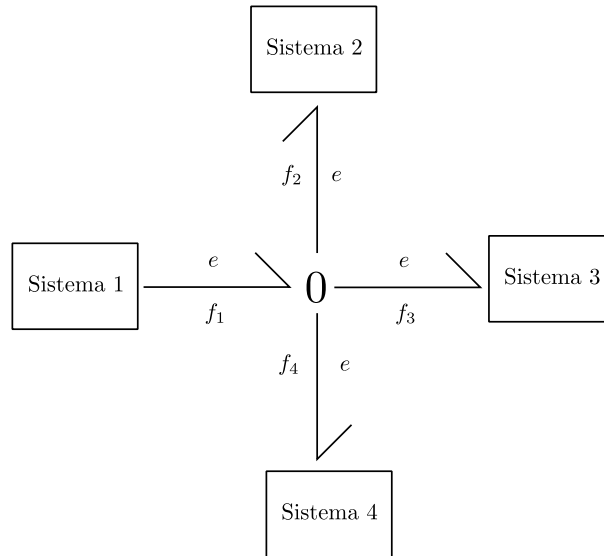


Figura 6.2: Un Vínculo 0 conectando cuatro subsistemas.

Dado que se trata de un componente estructural, se verificará que la suma de las potencias es cero, y por lo tanto:

$$\sum_{i=1}^m \pm p_i(t) = \sum_{i=1}^m \pm e_i(t) f_i(t) = e(t) \sum_{i=1}^m \pm f_i(t) = 0 \tag{6.1}$$

donde el signo \pm dependerá del sentido entrante o saliente de los arpones. Dado que la ecuación anterior deberá verificarse para cualquier trayectoria $e(t)$, deberá cumplirse también que

$$\sum_{i=1}^m \pm f_i(t) = 0 \tag{6.2}$$

que junto al esfuerzo común entre todos los enlaces conforma la ley constitutiva del Vínculo 0. En el caso particular del Vínculo 0 de la Figura 6.2, con un enlace entrante y tres salientes, la Ec.(6.2) toma la expresión:

$$f_1(t) - f_2(t) - f_3(t) - f_4(t) = 0 \tag{6.3}$$

En los circuitos eléctricos, por ejemplo, el Vínculo 0 representará la conexión en paralelo de subsistemas de modo que la Ecuación (6.2) se corresponde con la Ley de Kirchhoff de corrientes.

El componente dual es el Vínculo 1, que representa la transmisión de potencia entre subsistemas con el flujo como variable común, como se muestra en la Figura 6.3.

Al igual que en el Vínculo 0, deberá verificarse que la suma de las potencias es nula, de donde resultará

$$\sum_{i=1}^m \pm e_i(t) = 0 \tag{6.4}$$

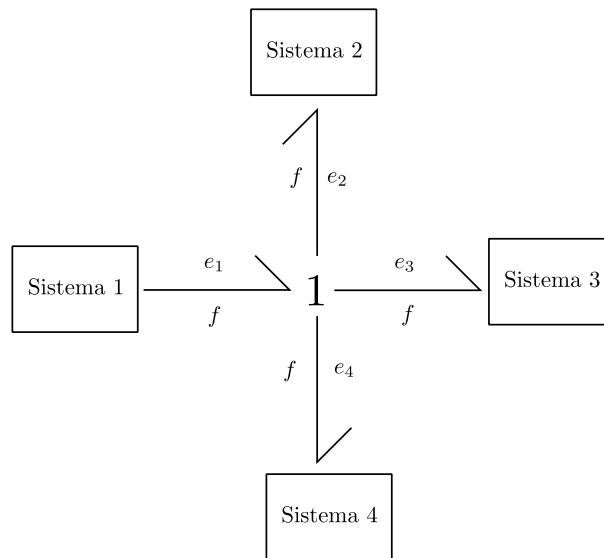


Figura 6.3: Un Vínculo 1 conectando cuatro subsistemas.

lo que junto a las ecuaciones del flujo común establece la ley constitutiva del componente.

En el caso de los circuitos eléctricos, el Vínculo 0 representará la conexión en serie de subsistemas.

Elementos Pasivos

Hay tres elementos pasivos básicos, uno que modela el fenómeno de disipación llamado Resistor (que se representa con el símbolo **R**), y dos que modelan fenómenos de almacenamiento: el Capacitor (que se representa con el símbolo **C**) y la Inercia (que se representa con el símbolo **I**).

La Figura 6.4 muestra un Resistor conectado a un sistema. En el Bond correspondiente, están representadas las variables de flujo y de esfuerzo.

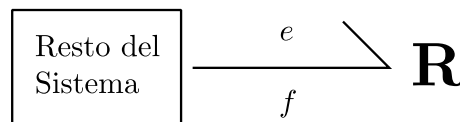


Figura 6.4: Un Resistor conectado a un sistema.

El Resistor tiene una ley constitutiva que vincula estáticamente las variables de flujo y esfuerzo. En el caso general esto se traduce en una ecuación algebraica del tipo

$$g(e, f) = 0 \tag{6.5}$$

donde $g(\cdot, \cdot)$ es una función que debe cumplir $g(0, 0) = 0$ (esto implica que esfuerzo y flujo igual a cero es un punto de la curva) y que debe ser tal que la relación entre e y f esté restringida al primer y tercer cuadrante). Esta restricción al primer y tercer cuadrante implica que el componente siempre disipa energía.

En el caso lineal, la relación entre esfuerzo y flujo queda establecida por un parámetro R según la ley:

$$e - R f = 0 \tag{6.6}$$

Para los circuitos eléctricos, es evidente que el elemento **R** representará la resistencia eléctrica.

La Figura 6.5 muestra un Capacitor que, como mencionamos, es uno de los elementos que representa el almacenamiento de energía.

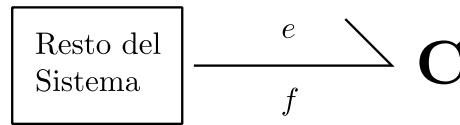


Figura 6.5: Un Capacitor conectado a un sistema.

El capacitor, además de las variables de potencia (e y f) vincula una *variable de energía* $q(t)$ que representa el *desplazamiento generalizado*, según las leyes constitutivas:

$$\begin{aligned} \dot{q}(t) - f(t) &= 0 \\ g(q(t), e(t)) &= 0 \end{aligned} \tag{6.7}$$

donde de manera similar que en el resistor, la función $g(\cdot, \cdot)$ establece una relación estática entre $q(t)$ y $e(t)$ restringida al primer y tercer cuadrante.

En el caso lineal, la relación estática entre esfuerzo y desplazamiento la establece un parámetro C , de manera que la Ec.(6.7) ahora queda

$$\begin{aligned} \dot{q}(t) - f(t) &= 0 \\ q(t) - C e(t) &= 0 \end{aligned} \tag{6.8}$$

Es claro también que en los circuitos eléctricos el elemento **C** permitirá representar capacitores y que la variable de energía $q(t)$ representará la carga en el capacitor.

El último componente pasivo elemental es la Inercia, que se muestra en la Figura 6.6 y que también modela almacenamiento de energía.

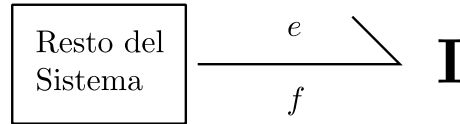


Figura 6.6: Una Inercia conectada a un sistema.

La inercia, además de las variables de potencia (e y f) vincula otra variable de energía $p(t)$ que representa el *momento generalizado*, según las leyes constitutivas:

$$\begin{aligned} \dot{p}(t) - e(t) &= 0 \\ g(p(t), f(t)) &= 0 \end{aligned} \tag{6.9}$$

donde de manera análoga al resistor y al capacitor, la función $g(\cdot, \cdot)$ establece una relación estática entre $p(t)$ y $f(t)$ restringida al primer y tercer cuadrante.

En el caso lineal, la relación estática entre flujo y momento la establece un parámetro I , de manera que la Ec.(6.9) ahora queda

$$\begin{aligned} \dot{p}(t) - e(t) &= 0 \\ p(t) - I f(t) &= 0 \end{aligned} \tag{6.10}$$

En los circuitos eléctricos el elemento **I** permitirá representar inductores donde la variable de energía $p(t)$ representará el flujo magnético.

Con los elementos pasivos y los vínculos estructurales podemos construir modelos simples, como el correspondiente a un circuito RLC serie que se muestra en la Figura 6.7. En la misma, junto a los componentes aparecen los parámetros C_1 , L_1 y R_1 que ayudan a visualizar de qué componente se trata.

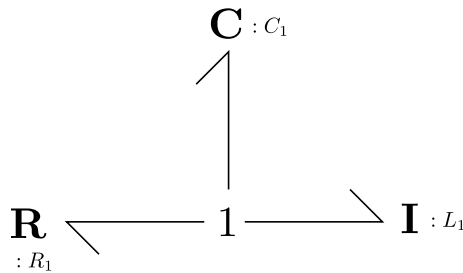


Figura 6.7: Representación en Bond Graph de un Circuito RLC Serie.



Figura 6.8: Fuentes Ideales de Esfuerzo y Flujo en Bond Graphs.

Fuentes Ideales

Hay dos tipos de fuentes ideales en Bond Graphs. La fuente ideal de esfuerzo (**Se**) y la de flujo (**Sf**), que se muestran en la Figura 6.8.

La fuente de esfuerzo se caracteriza por una ley constitutiva

$$e(t) = E(t) \tag{6.11}$$

donde $E(t)$ es una trayectoria dada. En el caso de los circuitos eléctricos este componente representaría una fuente ideal de tensión.

La fuente ideal de flujo, en tanto, se caracteriza por la ley dual

$$f(t) = F(t) \tag{6.12}$$

donde $F(t)$ es una trayectoria brindada.

A modo de ejemplo, la Figura 6.9 muestra el modelo en BG de un circuito RLC serie con una fuente de tensión ideal. En el mismo, además de los parámetros se indica la corriente común que representa el Vínculo 1.

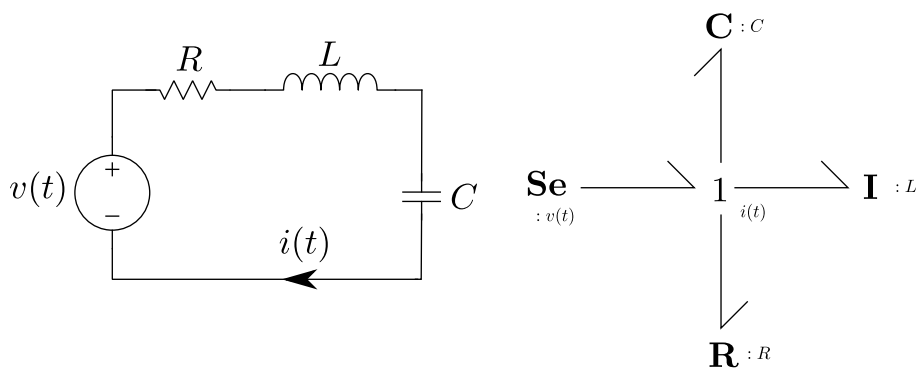


Figura 6.9: Circuito RLC Serie con Fuente Ideal.

Transformadores y Giradores

Además de los vínculos 0 y 1, hay otros dos elementos que representan la transferencia de energía entre subsistemas, denominados Transformador (**TF**) y Girador (**GY**).

El Transformador, mostrado en la Figura 6.10, representa la transferencia de energía entre dos subsistemas a través de una relación que vincula los esfuerzos de ambos lados entre sí y los flujos de ambos lados entre sí.



Figura 6.10: Un Transformador Conectando dos Subsistemas.

Las ecuaciones constitutivas de este componente son

$$\begin{aligned} e_1 - K e_2 &= 0 \\ K f_1 - f_2 &= 0 \end{aligned} \tag{6.13}$$

donde K es el parámetro del transformador. Notar que e_1 y f_1 en la Ec.(6.13) son el esfuerzo y flujo del lado *primario*, es decir, el que aporta energía, mientras que e_2 y f_2 son las variables de potencia del secundario. Por lo tanto K representaría la ganancia en el flujo desde el primario hacia el secundario. Notar además que el componente transmite potencia de manera ideal ya que

$$p_1 = e_1 f_1 = e_2 f_2 = p_2 \tag{6.14}$$

En el dominio mecánico rotacional este componente puede representar una caja reductora, por ejemplo. Como luego veremos con más detalle, en mecánica las variables de flujo serán las velocidades. La Figura 6.11 muestra un sistema rotacional con el modelo en Bond Graphs correspondiente que incluye un transformador.

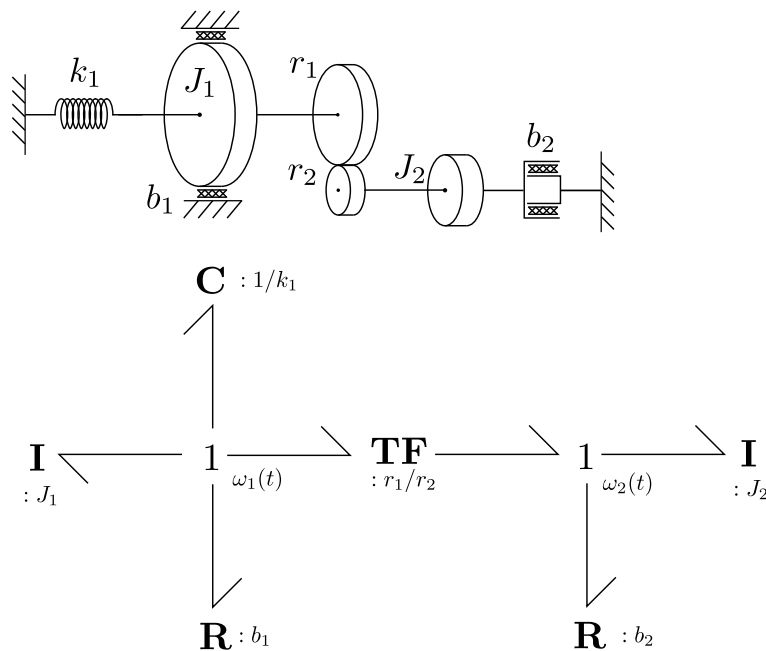


Figura 6.11: Modelo Mecánico-Rotacional con un Transformador.

El Girador, en tanto, mostrado en la Figura 6.12, representa la transferencia de energía entre dos subsistemas a través de una relación que vincula de manera cruzada esfuerzos y flujos de ambos lados.

Las ecuaciones constitutivas por lo tanto resultan:

$$\begin{aligned} e_1 - K f_2 &= 0 \\ K f_1 - e_2 &= 0 \end{aligned} \tag{6.15}$$



Figura 6.12: Un Girador Conectando dos Subsistemas.

Al igual que el transformador, este componente transmite potencia de manera ideal verificando la Ecuación (6.14). El fenómeno que típicamente se representa mediante este elemento es la conversión electromagnética-mecánica de energía que ocurre por ejemplo en un motor de corriente continua.

La Figura 6.13 muestra el modelo de un motor de corriente continua con excitación constante en su representación en Bond Graph, donde aparece un Girador.

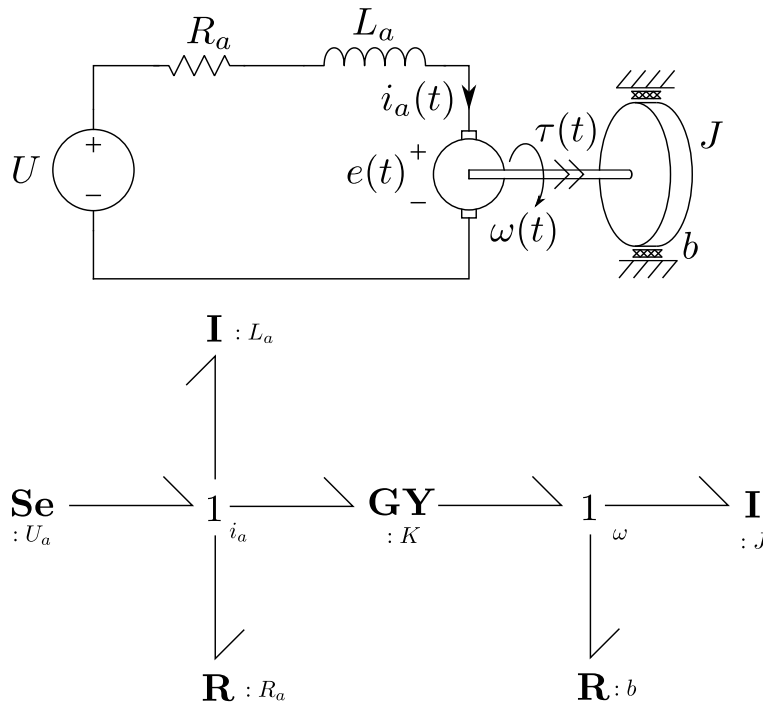


Figura 6.13: Modelo Electromecánico con un Girador.

6.1.2. Sensores y Componentes Modulados

Los componente básicos que vimos permiten representar los distintos fenómenos de interacción energética entre subsistemas. Sin embargo, como vimos en capítulos anteriores, en muchas ocasiones hay interacciones entre sistemas que no involucran intercambio de energía. Por ejemplo, cuando movemos la posición del vástago de una válvula cambiamos su relación presión-caudal, pero no estamos intercambiando energía entre la parte mecánica que mueve el vástago y la parte hidráulica. Lo que ocurre es que hay una *interacción en señal* que es muy diferente a una *interacción en potencia*.

Los Bond Graphs *puros* que utilizan los elementos básicos que vimos antes, sólo pueden representar interacción en potencia ya que un enlace que vincula dos subsistemas representa la potencia que intercambian. Para poder representar interacciones en señal necesitaremos utilizar *componentes modulados*, esto es, componentes que cambien sus características en función de una señal de entrada que provendrá de un Diagrama de Bloques. Esto permitirá por ejemplo representar sistemas de control, para lo cual además necesitaremos *sensores* que transformen los esfuerzos y flujos de un Bond Graph en las correspondientes señales en los Diagramas de Bloques.

La Figura 6.14, por ejemplo, muestra el modelo en Bond Graph de un motor de corriente continua de excitación independiente donde aparece un *Girador Modulado (MGY)*. En dicho modelo hay interacción

Bond Graphs	Esfuerzo e	Flujo f	Momento p	Desplazamiento q
Circuitos Eléctricos	Tensión (V)	Corriente (A)	Flujo Magnético (Wb)	Carga (C)
Mecánica Traslacional	Fuerza (N)	Velocidad (m/s)	Impulso (N s)	Desplazamiento (m)
Mecánica Rotacional	Torque (N m)	Velocidad Angular (m/s)	Momento Angular (N m s)	Ángulo (rad)
Fluidodinámica	Presión (Pa)	Caudal (m ³ /s)	Impulso del fluido (N s)	Volumen (m ³)
Termodinámica	Temperatura (K)	Flujo de entropía (W/K)	–	Entropía (J/K)
Química	Potencial Químico (J/mol)	Flujo molar (mol/s)	–	Número de Moles (mol)

Tabla 6.1: Variables Generalizadas en los Distintos Dominios Físicos

Para el caso lineal, recordando que $f_I(p_I) = p_I/I$, resulta

$$E_I(p_I) = \frac{1}{2I} p_I^2 = \frac{1}{2} I f_I^2 \tag{6.17}$$

Siguiendo igual razonamiento, para un elemento C resultará:

$$E_C(q_C) = \int_0^{q_C} e_C(q) dq \tag{6.18}$$

Para el caso lineal, recordando que $e_C(q_C) = q_C/C$, resulta

$$E_C(q_C) = \frac{1}{2C} q_C^2 = \frac{1}{2} C e_C^2 \tag{6.19}$$

6.2. Modelado con Bond Graphs

Veremos a continuación algunas técnicas elementales para plantear modelos en Bond Graphs de distintos dominios. En todos los casos, el procedimiento de modelado consistirá en colocar primero los vínculos 1 y 0 que representan la estructura, luego los transformadores y giradores que completan las relaciones entre distintas variables de flujo y de esfuerzo y finalmente los elementos que representan el almacenamiento, la generación y la disipación de energía asociada a cada variable de potencia.

6.2.1. Circuitos Eléctricos

Para modelar circuitos eléctricos la idea básica es utilizar un vínculo 0 para cada potencial (excepto la masa) y vínculos 1 para formar las diferencias de potencial a las que ocurra algún fenómeno. Luego, la idea es colocar en cada vínculo 0 los elementos correspondientes a dicho voltaje.

Consideremos por ejemplo el circuito de la Figura 6.15.

En el mismo podemos distinguir dos potenciales (v_1 y v_2) y una diferencia de potencial ($v_1 - v_2$) a la que ocurre algún fenómeno. Entonces, podemos comenzar a formar la estructura con tres vínculos 0 como se muestra en el Paso 1 de la Figura 6.16. Luego, para completar la estructura debemos formar las diferencias de potencial uniendo los vínculos 0 correspondientes mediante vínculos 1 como se muestra en el Paso 2 (notar que podríamos haber usado ahí todos los enlaces en sentido contrario). Finalmente, asociado a cada voltaje, agregamos los componentes que representan los fenómenos energéticos como se muestra en el Paso 3. Podríamos además eventualmente simplificar este Bond Graph eliminando el vínculo 0 que representa el voltaje $v_1 - v_2$ ya que es *pasante* (tiene sólo dos enlaces con idéntico flujo y esfuerzo). En tal caso, el Resistor R_2 quedaría directamente conectado al vínculo 1.

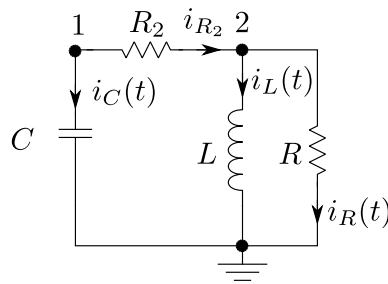


Figura 6.15: Circuito RLC

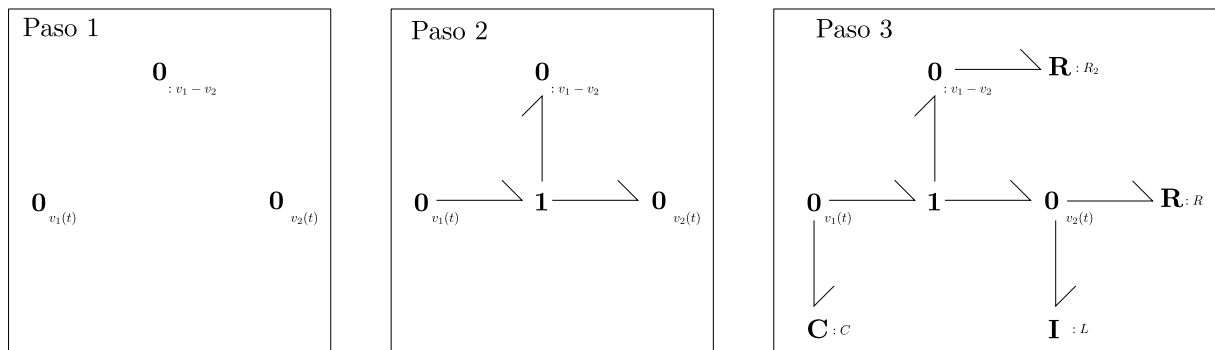


Figura 6.16: Construcción del Bond Graph de un Circuito RLC

Notar que sólo tuvimos en cuenta las relaciones entre los voltajes sin mirar que pasaba con las corrientes. Esto sin embargo funciona ya que al construir un modelo en Bond Graph estamos además representando lo que ocurre con la potencia. Dado que la potencia es el producto de la corriente y la tensión, si están bien establecidas las relaciones entre los distintos voltajes y entre las distintas potencias, implícitamente quedan bien establecidas las relaciones entre las corrientes.

6.2.2. Sistemas Mecánicos

El procedimiento para hacer modelos de sistemas mecánicos es muy similar al caso eléctrico. La idea es asociar un vínculo 1 a cada velocidad absoluta o relativa del sistema a la que ocurra algún fenómeno. Luego, continuar utilizando vínculos 0 para conectar las velocidades absolutas con las relativas. Si hubiera algún transformador, también deberá utilizarse para vincular las velocidades correspondientes y por último, una vez completada la estructura, se agregarán los componentes que representan los fenómenos energéticos.

La Figura 6.17 muestra un sistema roto-traslacional con el procedimiento de construcción del Bond Graph correspondiente. En el Paso 1 colocamos un vínculo 1 asociado a cada velocidad. Luego, en el Paso 2 con un vínculo 0 conectamos las velocidades absolutas y relativas. Además, mediante un transformador vinculamos la velocidad traslacional v_2 con la rotacional ω . Finalmente, agregamos los componentes en el Paso 3.

Análogo al caso eléctrico, construimos el modelo sin tener en cuenta las fuerzas o torques. Sólo miramos que ocurriría con las velocidades. Dado que en Bond Graphs usamos también las relaciones entre potencias, las relaciones entre las fuerzas y/o torques quedan implícitamente establecidas.

Cabe hacer algunas aclaraciones sobre el modelo obtenido, que serán generales para los casos mecánicos:

- Las inercias se representan mediante el componente **I** que será lineal. Comparando la Ec.(6.10) con la expresión de la cantidad de movimiento $p(t) = m v(t)$, queda claro que para los parámetros se cumplirá $I = m$. Análogamente, en el caso rotacional tendremos $I = J$.

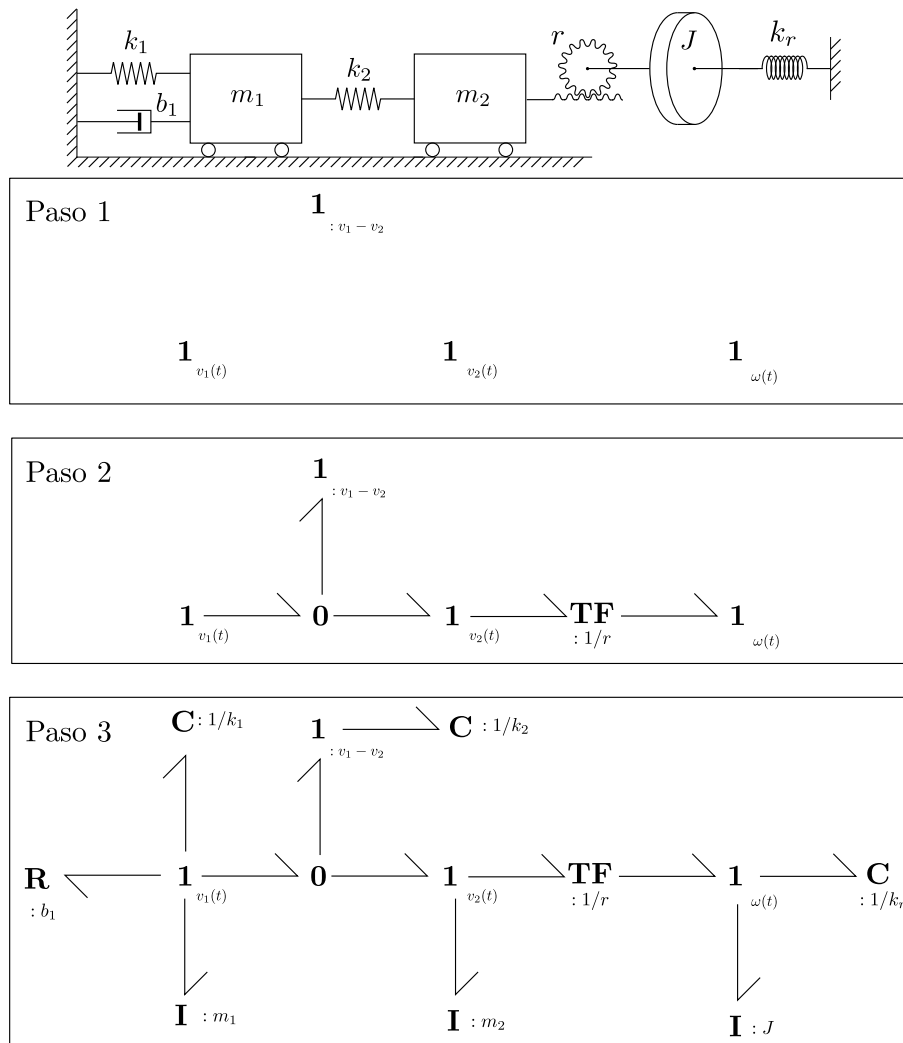


Figura 6.17: Construcción del Bond Graph de un Sistema Roto-Traslacional

- El rozamiento y los amortiguadores se representa mediante el componente **R**. En el caso lineal se cumplirá la ley $F(t) = b v(t)$ y comparando esta expresión con la Eq.(6.6) es claro que los parámetros cumplirán $R = b$. En el caso rotacional se puede deducir la misma expresión.
- Los resortes se representan mediante el componente **C**. En el caso lineal se cumplirá la ley $F(t) = k \Delta x(t)$, que comparada con la expresión $q(t) = C e(t)$ de la Eq.(6.8) conduce a la relación $C = 1/k$. Lo mismo ocurrirá en el caso rotacional.
- En el modelo no aparecen las posiciones absolutas de las masas. En los Bond Graphs no aparecerán posiciones, sólo aparecerán las velocidades, las fuerzas, las cantidades de movimiento y las deformaciones de los resortes. El motivo es que las posiciones absolutas no juegan ningún rol en el intercambio de energía.
- Para obtener posiciones absolutas podemos tomar señales de velocidad e integrarlas (mediante un bloque integrador) armando un *Diagrama Mixto* como el de la Figura 6.14.

6.2.3. Sistemas Hidráulicos

Para los sistemas hidráulicos puede plantearse un procedimiento idéntico al de los circuitos eléctricos, colocando un vínculo 0 para cada presión absoluta (excepto la atmosférica) y para cada presión relativa

a la que ocurra algún fenómeno.

La Figura 6.18 muestra un sistema hidráulico roto-traslacional y el procedimiento de construcción del Bond Graph. En el Paso 1 colocamos un vínculo 0 asociado a cada presión absoluta y relativa. Luego, en el Paso 2 con dos vínculos 1 conectamos las presiones absolutas y relativas. Finalmente, agregamos los componentes en el Paso 3.

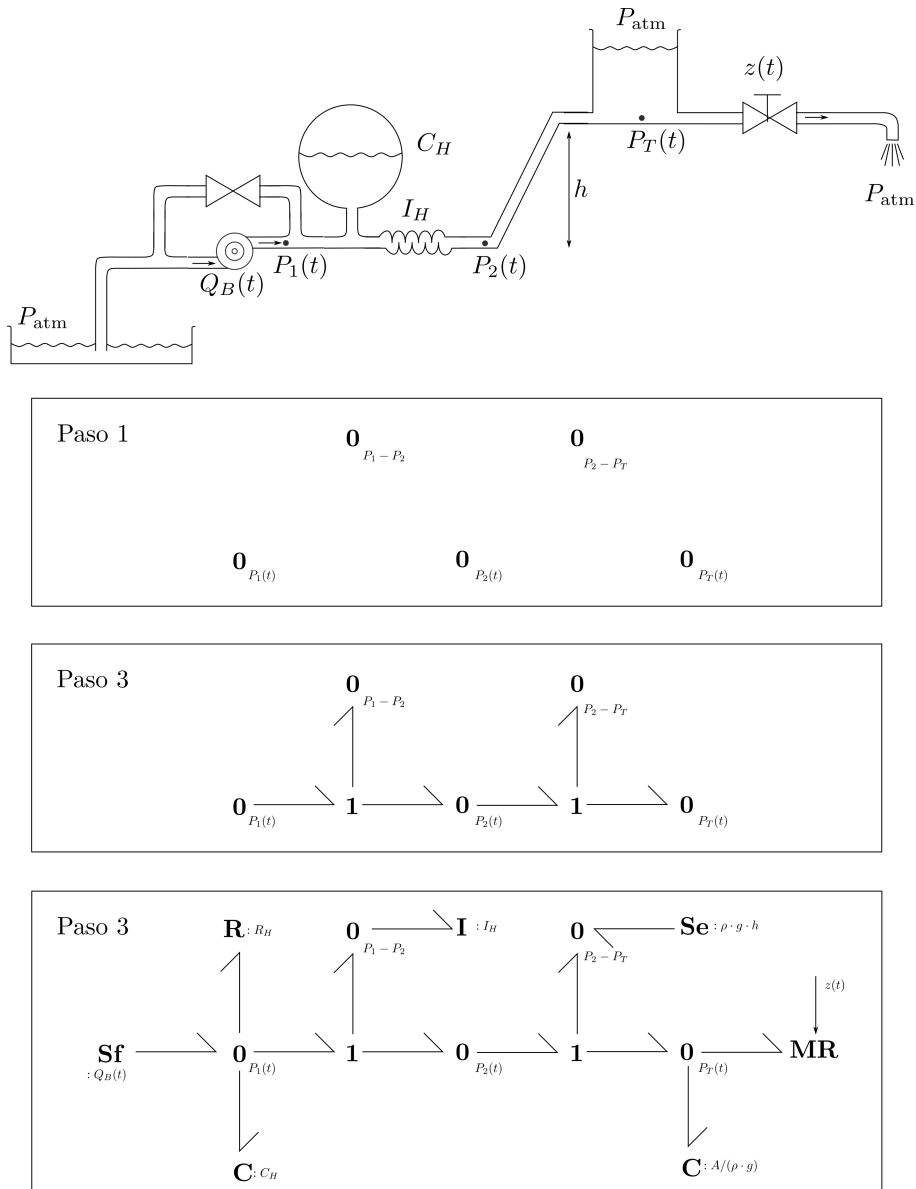


Figura 6.18: Construcción del Bond Graph de un Sistema Hidráulico

Al igual que en los casos eléctricos y mecánicos, las correspondencias entre componentes hidráulicos y de Bond Graphs son directas. En el caso del tanque, cabe recordar que la relación constitutiva del mismo es

$$\begin{aligned}
 P(t) &= \rho g h(t) = \frac{\rho g}{A} V(t) \\
 \dot{V}(t) &= Q(t)
 \end{aligned}
 \tag{6.20}$$

y comparando esta expresión con la Ec.(6.8), resulta

$$C = \frac{A}{\rho g} \tag{6.21}$$

En el caso de la columna de agua, notar que la asociamos a una fuente de esfuerzo ideal. En cuanto al sentido de potencia del enlace que la conecta, podríamos haber usado uno en sentido contrario obteniendo un modelo equivalente salvo que el flujo en dicho enlace (que no influye en el modelo) hubiera resultado de signo contrario. En general, al conectar una fuente de esfuerzo a un vínculo 0 no importa el sentido del enlace. Lo mismo ocurre al conectar una fuente de flujo a un vínculo 1. El sentido en tales casos sólo define en que dirección consideramos que es positivo el aporte de potencia. Podemos por lo tanto poner siempre el sentido desde la fuente hacia el resto del sistema.

Si en cambio conectamos una fuente de esfuerzo con un vínculo 1, el sentido del enlace es importante ya que el esfuerzo de la fuente forma parte de la sumatoria de esfuerzos asociada a dicho vínculo. En este caso igualmente podemos usar fuentes que aportan energía al sistema (con el enlace en dirección al resto del sistema) colocando el signo adecuado al esfuerzo de la fuente. Vale lo mismo para una fuente de flujo en un vínculo 0.

6.2.4. Sistemas Multidominio

Para construir modelos de Bond Graph de sistemas multidominio se puede seguir los procedimientos correspondientes a cada dominio y completar el diagrama con los elementos que representen la interfaz entre los distintos dominios. Estos componentes de interfaz, cuando involucren intercambio de energía entre los subsistemas, resultarán de tipo transformador o girador. Por el contrario, cuando lo que haya sea una interacción en señal, la conexión entre submodelos se dará a través de elementos modulados.

Respecto a los modelos electromecánicos, ya presentamos el modelo de un motor de corriente continua y vimos que la interfaz la realizaba un girador. Para sistemas hidráulico-mecánicos, en tanto, el componente de interfaz será generalmente un pistón (o algo similar) que podrá representarse mediante un transformador ya que el mismo vincula entre sí las variables de flujo (velocidad y caudal) y las variables de esfuerzo (fuerza y presión).

La Figura 6.19 muestra el Bond Graph correspondiente a un sistema hidráulico-mecánico. El mismo tiene dos transformadores que representan los pistones de área A_1 y A_2 .

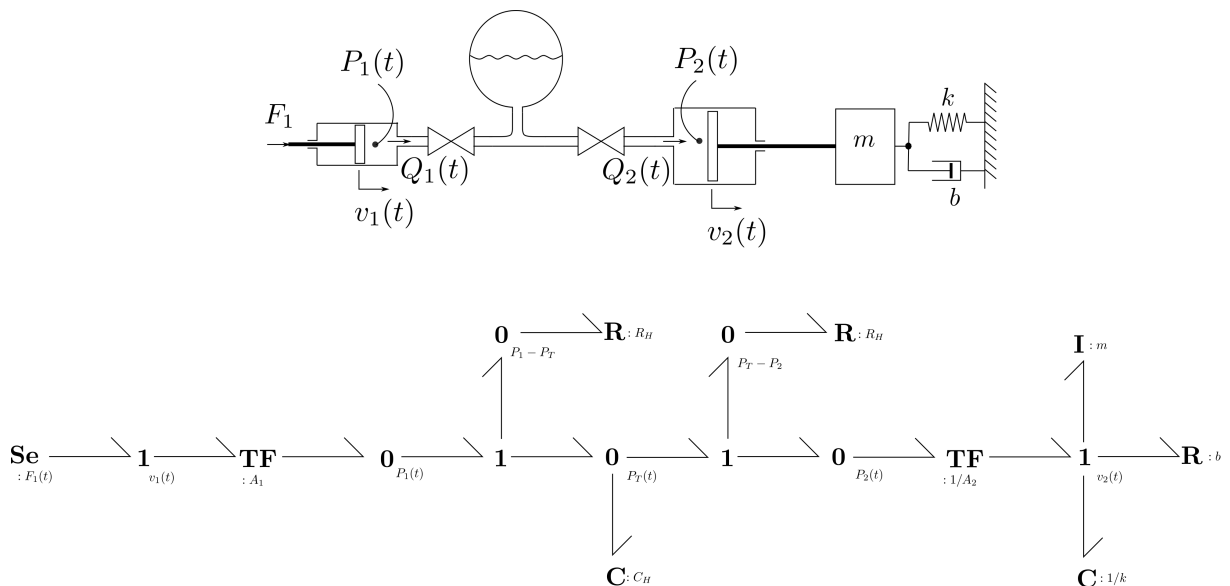


Figura 6.19: Bond Graph de un Sistema Hidráulico-Mecánico

En cuanto a los módulos de los transformadores, recordemos que según la Ec.(6.13) vale la relación $f_2(t) = K f_1(t)$, de donde K es la ganancia en el flujo desde el primario hacia el secundario. Por lo tanto,

en el transformador de la izquierda tenemos

$$f_2(t) = Q(t) = A_1 v_1(t) = K f_1(t) \tag{6.22}$$

de donde $A_1 = K$. En el segundo transformador el primario está en la parte hidráulica, por lo tanto la relación queda al revés.

Notar además que hay cuatro vínculos 0 pasantes en el Bond Graph de la Figura 6.19, por lo que el diagrama puede reducirse bastante.

6.3. Bond Graphs y DAEs

Al igual que lo que ocurría con los modelos orientados a objetos, al construir un modelo en Bond Graph estamos implícitamente definiendo un sistema de DAEs, en el cual pueden distinguirse relaciones estructurales asociadas a los vínculos 0 y 1 y relaciones constitutivas asociadas a los restantes elementos. Como veremos, la obtención de las DAEs es algo más simple que en los modelos orientados a objetos y hasta podremos determinar de forma simple si un modelo contendrá lazos algebraicos o índice alto.

6.3.1. Obtención de las Relaciones Constitutivas y Estructurales

Para obtener las relaciones constitutivas y estructurales que conforman el sistema de DAEs correspondiente a un Bond Graph, seguiremos el siguiente procedimiento:

1. Numeraremos los enlaces utilizando los números 1, 2, etc.
2. Denominaremos e_i y f_i a las variables de esfuerzo y flujo asociadas al enlace i .
3. Escribiremos en función de dichas variables generalizadas las ecuaciones de los componentes, tanto de los elementos **R**, **C**, etc. como de los vínculos estructurales **0** y **1**.
4. En el caso de los almacenadores, aparecerán las variables de energía q_i o p_i según el tipo.

Veamos por ejemplo el modelo de la Figura 6.20, del que ya habíamos obtenido el Bond Graph (aunque en este caso eliminamos un vínculo pasante). En el mismo siguiendo el primer paso del procedimiento enumeramos los enlaces.

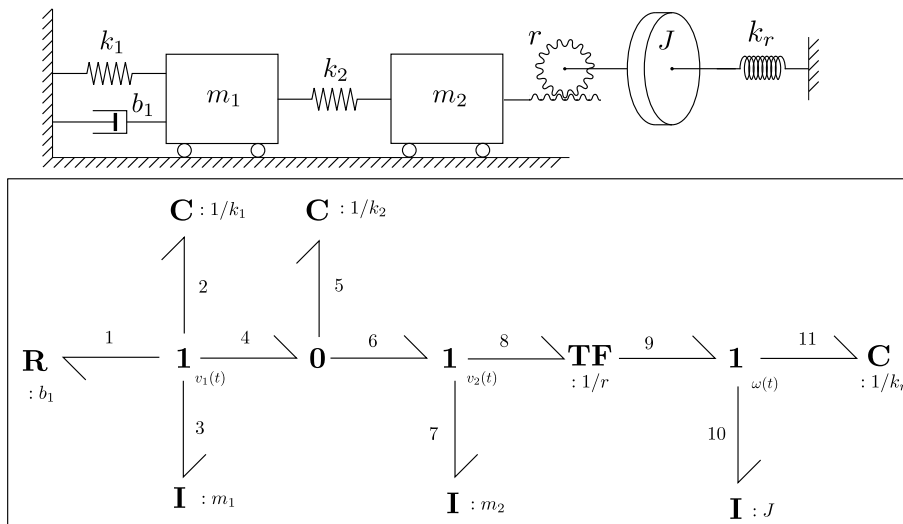


Figura 6.20: Bond Graph para Obtener el Sistema de DAEs.

Podemos entonces plantear el sistema de DAEs de las Ecs.(6.23)–(6.24). Como puede verse, el procedimiento es simple y sistemático. El sistema tiene 28 ecuaciones y 28 variables. Hay 22 variables de potencia

Relaciones constitutivas:

$$\begin{aligned}
 e_1(t) - b_1 f_1(t) &= 0 \\
 \dot{q}_2(t) - f_2(t) &= 0 \\
 q_2(t) - \frac{1}{k_1} e_2(t) &= 0 \\
 \dot{p}_3(t) - e_3(t) &= 0 \\
 p_3(t) - m_1 f_2(t) &= 0 \\
 \dot{q}_5(t) - f_5(t) &= 0 \\
 q_5(t) - \frac{1}{k_2} e_5(t) &= 0 \\
 \dot{p}_7(t) - e_7(t) &= 0 \\
 p_7(t) - m_2 f_7(t) &= 0 \\
 e_8(t) - \frac{1}{r} e_9(t) &= 0 \\
 \frac{1}{r} f_8(t) - f_9(t) &= 0 \\
 \dot{p}_{10}(t) - e_{10}(t) &= 0 \\
 p_{10}(t) - J f_{10}(t) &= 0 \\
 \dot{q}_{11}(t) - f_{11}(t) &= 0 \\
 q_{11}(t) - \frac{1}{k_r} e_{11}(t) &= 0
 \end{aligned} \tag{6.23}$$

Relaciones estructurales:

$$\begin{aligned}
 f_1(t) - f_2(t) &= 0 \\
 f_1(t) - f_3(t) &= 0 \\
 f_1(t) - f_4(t) &= 0 \\
 -e_1(t) - e_2(t) - e_3(t) - e_4(t) &= 0 \\
 e_4(t) - e_5(t) &= 0 \\
 e_4(t) - e_6(t) &= 0 \\
 f_4(t) - f_5(t) - f_6(t) &= 0 \\
 f_6(t) - f_7(t) &= 0 \\
 f_6(t) - f_8(t) &= 0 \\
 e_6(t) - e_7(t) - e_8(t) &= 0 \\
 f_9(t) - f_{10}(t) &= 0 \\
 f_9(t) - f_{11}(t) &= 0 \\
 e_9(t) - e_{10}(t) - e_{11}(t) &= 0
 \end{aligned} \tag{6.24}$$

(correspondientes a los flujos y esfuerzos en los 11 enlaces) y 6 variables de energía correspondientes a los 6 almacenadores.

Es importante en todo momento no perder de vista la equivalencia entre las variables generalizadas y las originales para poder interpretar el sentido físico de las ecuaciones. En este caso, por ejemplo, si quisiéramos saber la fuerza que ejerce el piñón cremallera sobre la masa m_2 es claro que la misma está representada por el esfuerzo en el enlace 8, es decir, dicha fuerza es el esfuerzo e_8 . De manera similar, el torque neto sobre la inercia será el esfuerzo e_{10} y la deformación del resorte de torsión será el desplazamiento q_{11} .

6.3.2. Causalización

Si bien los Bond Graphs son intrínsecamente acausales, es posible incorporar información a los mismos respecto de la causalidad en las ecuaciones involucradas, lo que puede utilizarse principalmente para analizar propiedades del sistema.

Como sabemos, cada enlace transmite potencia entre dos subsistemas A y B a través de dos variables, el esfuerzo y el flujo. Desde el punto de vista causal, hay dos posibilidades: que el sistema A calcule el flujo y el B calcule el esfuerzo o lo contrario. La Figura 6.21 expresa estas dos situaciones y muestra como se indica en un Bond Graph la situación causal.

Como vemos, la codificación es simple. Se agrega una barra transversal al enlace del lado del sistema que calcula el flujo. Dicho de otra manera, la barra de causalidad se coloca del lado hacia el cual se impone el esfuerzo.

Utilizando esta representación podemos decidir las relaciones causales entre todas las variables de un Bond Graph, para lo cual tendremos que tener en cuenta las siguientes restricciones:

- Una fuente de esfuerzo necesariamente debe imponer esfuerzo al resto del sistema. Por lo tanto la barra de causalidad deberá estar del lado del otro subsistema.
- Una fuente de flujo necesariamente debe imponer flujo al resto del sistema. Por lo tanto la barra de causalidad deberá estar del lado de la fuente.

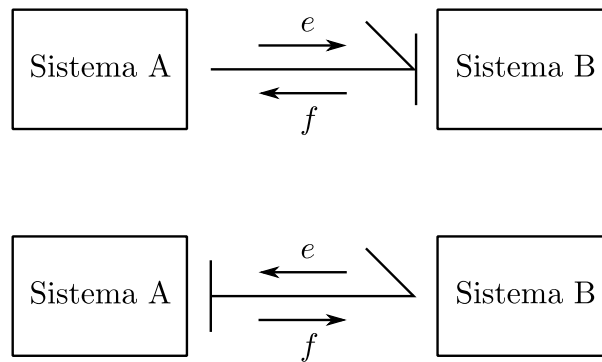


Figura 6.21: Posibles Situaciones Causales en un Enlace

- Un vínculo 0 tiene esfuerzo en común, por lo tanto un único enlace debe imponer el esfuerzo. Luego, debe haber un único enlace con la barra de causalidad del lado del vínculo 0.
- Un vínculo 1 tiene flujo en común, por lo tanto un único enlace debe imponer el flujo. Luego, todos los enlaces salvo uno debe la barra de causalidad del lado del vínculo 1.
- Los transformadores establecen una relación entre dos esfuerzos y otra relación entre dos flujos. Luego, si uno de los subsistemas conectados a un transformador calcula el esfuerzo, quedará determinado el esfuerzo del otro subsistema. Lo mismo respecto a los flujos. Por lo tanto, deberá haber una única barra de causalidad del lado del transformador.
- En los giradores ocurre lo contrario que en los transformadores y por lo tanto las dos barras deberán estar del lado del girador o las dos del otro lado.
- En los capacitores podríamos usar ambas opciones, pero preferiremos que impongan esfuerzo ya que eso implicará utilizar *causalidad integral*.
- En los inductores podríamos usar ambas opciones, pero preferiremos que impongan flujo.
- Para los resistores la causalidad será indistinta.

Teniendo en cuenta estas reglas, podemos proponer el siguiente procedimiento para causalizar un Bond Graph:

1. Si no quedan fuentes sin causalizar, avanzamos al punto 5.
2. Causalizamos una fuente.
3. Propagamos la causalidad a través de los vínculos estructurales, transformadores y giradores.
4. Volvemos al punto 1.
5. Si no quedan almacenadores sin causalizar, avanzamos al punto 9.
6. Causalizamos un almacenador de forma integral.
7. Propagamos la causalidad a través de los vínculos estructurales, transformadores y giradores.
8. Volvemos al punto 5.
9. Si no quedan resistores sin causalizar termina el procedimiento.
10. Causalizamos un resistor de forma arbitraria.
11. Propagamos la causalidad a través de los vínculos estructurales, transformadores y giradores.

12. Volvemos al punto 9.

La propagación de la causalidad en tanto, se corresponde al siguiente procedimiento:

1. Si un vínculo 0 tiene un enlace que le impone esfuerzo, causalizar los restantes enlaces imponiendo esfuerzo al resto del sistema y propagar.
2. Si un vínculo 0 tiene todos los enlaces que le imponen flujo salvo uno, causalizar ese enlace imponiendo flujo al resto del sistema y propagar.
3. Si un vínculo 1 tiene un enlace que le impone flujo, causalizar los restantes enlaces imponiendo flujo al resto del sistema y propagar.
4. Si un vínculo 1 tiene todos los enlaces que le imponen esfuerzo salvo uno, causalizar ese enlace imponiendo esfuerzo al resto del sistema y propagar.
5. Si un transformador tiene un enlace que le impone esfuerzo, causalizar el otro enlace imponiendo esfuerzo al resto del sistema y propagar.
6. Si un transformador tiene un enlace que le impone flujo, causalizar el otro enlace imponiendo flujo al resto del sistema y propagar.
7. Si un girador tiene un enlace que le impone esfuerzo, causalizar el otro enlace imponiendo flujo al resto del sistema y propagar.
8. Si un girador tiene un enlace que le impone flujo, causalizar el otro enlace imponiendo esfuerzo al resto del sistema y propagar.

A modo de ejemplo, la Figura 6.22 ilustra los primeros pasos del procedimiento de causalización de un Bond Graph, correspondiente al modelo hidráulico-mecánico de la Figura 6.19 en el cual eliminamos los enlaces pasantes. En primer lugar causalizamos la fuente y luego propagamos a través del transformador. Luego causalizamos de forma integral el capacitor hidráulico y propagamos a través del vínculo 0 y luego del vínculo 1 de la izquierda.

Prosiguiendo luego con el proceso de causalización se llega finalmente a un Bond Graphs con todos sus enlaces causalizados como se muestra en la Figura 6.23. En el mismo puede verse que todos los elementos almacenadores preservan la causalidad integral.

6.3.3. Singularidades Estructurales y Lazos Algebraicos

Singularidades Estructurales

Consideremos ahora nuevamente el modelo roto-traslacional que habíamos introducido anteriormente. En la Figura 6.24 puede verse el resultado del proceso de causalización donde la inercia que representa la masa m_2 está en causalidad derivativa. Si hubiéramos causalizado dicha inercia con causalidad integral, hubiera quedado la inercia correspondiente a J en causalidad derivativa.

Es evidente el motivo. La masa m_2 y la inercia J están acopladas rígidamente y por lo tanto sus velocidades no son independientes. Desde el punto de vista de lo que vimos en el Capítulo 3, este problema es estructuralmente singular y el sistema de DAEs tiene por lo tanto índice alto.

Cada vez que el procedimiento de causalización nos conduzca a la aparición de causalidades derivativas sabremos que el sistema de DAEs resultante tendrá índice alto.

Lazos Algebraicos

Consideremos por último nuevamente el circuito eléctrico de la Figura 6.25 con el Bond Graph correspondiente. Siguiendo el procedimiento de causalización se llega a la situación que se muestra a la izquierda de la figura donde hay dos opciones para causalizar los resistores.

Cualquiera de las dos opciones nos permitirá completar la causalización. De todas formas, puede verse que esta situación en la cual hay más de una opción para completar la causalización se debe a la presencia

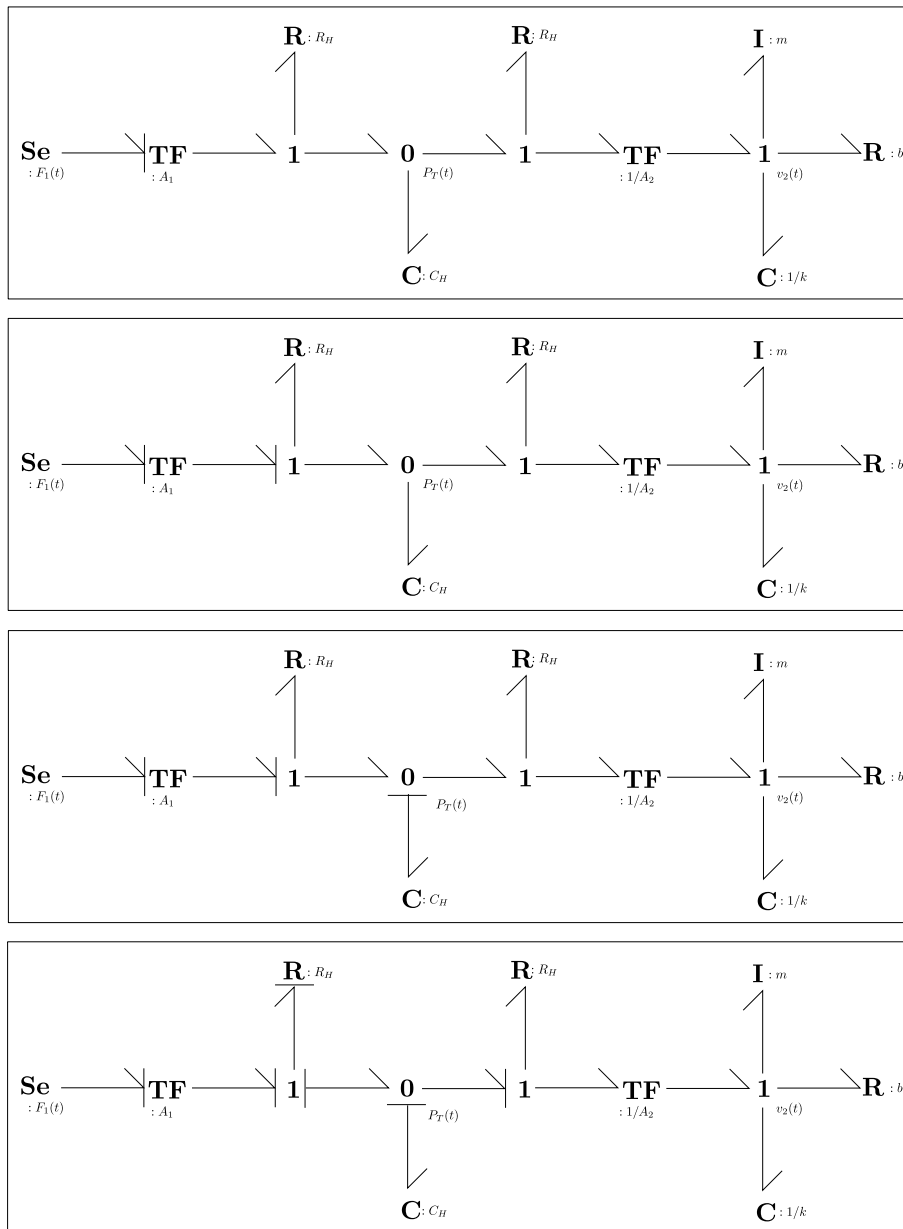


Figura 6.22: Proceso de Causalización de un Bond Graph

de un lazo algebraico. Esto se debe a que conociendo los valores de los estados y las entradas (es decir, conociendo las variables de esfuerzo o flujo que imponen almacenadores y fuentes) no queda establecido el valor del esfuerzo o del flujo en todos los resistores. Esto implicará que haya dependencia entre el esfuerzo y flujo de dos o más resistores cuyas ecuaciones deberán resolverse de forma simultánea.

6.3.4. Orden de la DAE

El orden de la DAE que resulta de un Bond Graph será igual al número de almacenadores en causalidad integral. De esta manera, el orden del Bond Graph de la Figura 6.25 será 2 (lo que se corresponde con el circuito).

Si miramos sin embargo el orden del Bond Graph de la Figura 6.24 vemos que hay 5 almacenadores en causalidad integral aunque el orden del modelo mecánico sea 4 (hay dos grados de libertad). En realidad

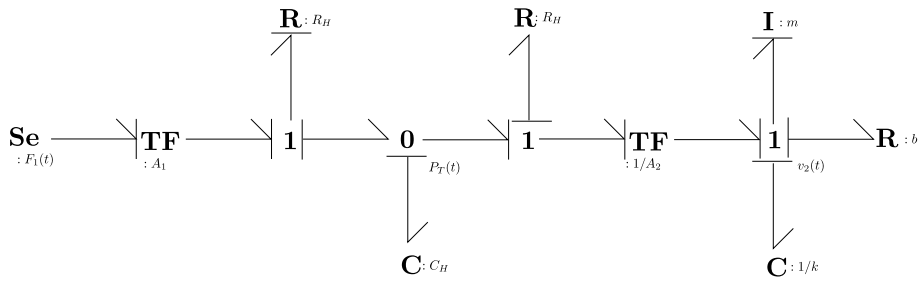


Figura 6.23: Bond Graph Causalizado

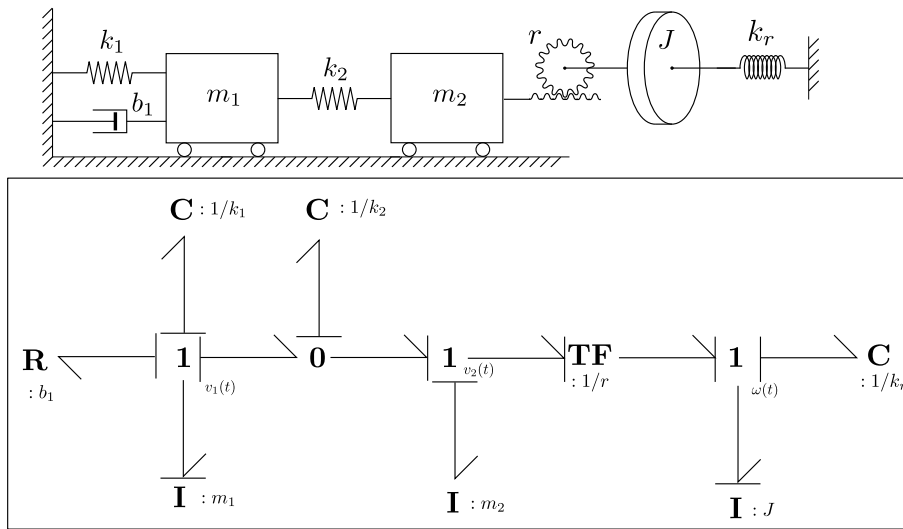


Figura 6.24: Bond Graph con Causalidad Derivativa.

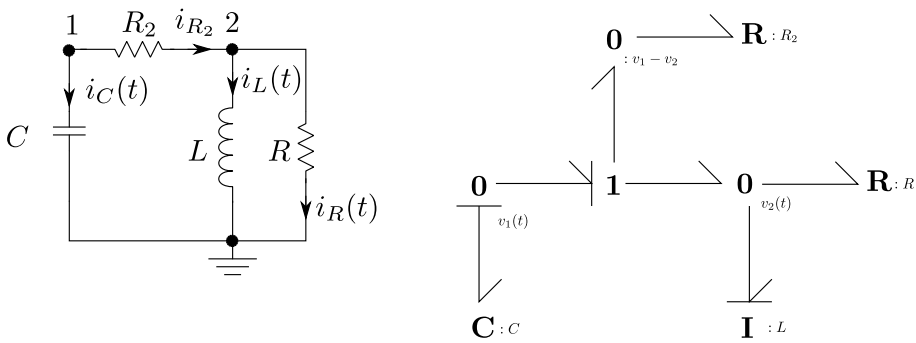


Figura 6.25: Bond Graph con Lazo Algebraico.

el orden del sistema de DAEs que produce el Bond Graph es efectivamente 5 y eso se debe a que las relaciones estructurales se establecen en función de las velocidades y no de las posiciones (ya habíamos visto este problema en el Capítulo 4). En consecuencia, no aparece la vinculación entre las posiciones (sólo aparece entre las velocidades) y en principio uno podría poner condiciones iniciales arbitrarias a la deformación de los distintos resortes.

6.4. Bond Graphs y Modelica

Como es de esperar, los Bond Graphs pueden representarse usando el lenguaje Modelica. Hay de hecho una librería denominada BondLib [CN05] que contiene una implementación de los distintos componentes.

En este curso utilizaremos una librería basada en las ideas de la BondLib, pero simplificada para poder construir modelos de forma más inmediata. Un problema de la BondLib es que cada enlace es un objeto, y hay que hacer dos conexiones, uno entre el enlace y el componente de un lado y otro entre el enlace y el componente del otro lado. En nuestra simplificación, utilizaremos componentes que ya contienen el enlace.

Los componentes de Bond Graph que veremos forman parte de la librería DSFLib.

6.4.1. Componentes Básicos

Para poder conectar un enlace con un componente necesitaremos un conector apropiado que vincule las variables del enlace (e y f) con las correspondientes del componente. Un problema que tendremos es que si el componente es un vínculo 1, todas las variables de flujo deberán ser iguales y todas las de esfuerzo se sumarán.

Esto implica que en realidad el concepto de variable de flujo de Modelica difiere del de Bond Graph. En Modelica los flujos suman cero en una conexión independientemente del contexto. En Bond Graph, por el contrario, los flujos pueden igualarse y los esfuerzos pueden sumar cero (como ocurre en el vínculo 1).

Por tal motivo, no podremos definir un conector con variables de flujo, sino que estaremos forzados a que las variables e y f se igualen a ambos lados del conector. Esto a su vez nos traerá un problema con los signos: si todas las variables se igualan a ambos lados, no tendremos forma a priori de saber en un vínculo 1 si hay que sumar o restar los esfuerzos (lo mismo ocurrirá con los flujos). Esto lo resolveremos utilizando una tercer variable en un conector que indica la dirección del enlace, de manera que la definición del conector resulta la brindada en el Código 6.1.

Código 6.1: Definición de un Conector de Bond Graph en Modelica

```
connector BondCon
  Real e;
  Real f;
  Integer d;
  annotation(...);
end BondCon;
```

La variable d entonces tomará los valores 1 o -1 según el sentido del enlace correspondiente sea entrante o saliente. Luego, un enlace dirigido de un conector `bondCon1` a un conector `bondCon2` puede representarse como sigue:

Código 6.2: Definición de un Enlace de Bond Graph en Modelica

```
model Bond
  DSFLib.BondGraph.Interfaces.BondCon bondCon1 annotation(...);
  DSFLib.BondGraph.Interfaces.BondCon bondCon2 annotation(...);
equation
  bondCon1.e = bondCon2.e;
  bondCon1.f = bondCon2.f;
  bondCon1.d = -1;
  bondCon2.d = 1;
  annotation(...);
end Bond;
```

Utilizando un enlace podemos construir el modelo de un componente pasivo que contenga un enlace dirigido hacia el componente:

Código 6.3: Definición de un Componente Pasivo con Enlace en Modelica

```

partial model PassiveOnePort
  DSFLib.BondGraph.Interfaces.BondCon bondCon annotation(...);
  Real e, f;
equation
  e = bondCon.e;
  f = bondCon.f;
  bondCon.d = -1;
  annotation(...);
end PassiveOnePort;

```

y luego, a partir de este componente podemos heredar componentes como el resistor, capacitor e inductor:

Código 6.4: Componentes Pasivos de Bond Graph en Modelica

```

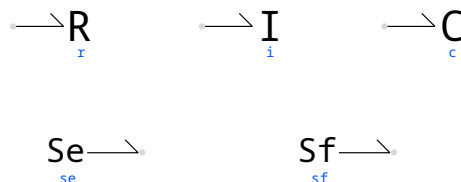
model R
  extends DSFLib.BondGraph.Interfaces.PassiveOnePort;
  parameter Real R = 1;
equation
  e = R*f;
  annotation(...);
end R;

model C
  extends DSFLib.BondGraph.Interfaces.PassiveOnePort;
  parameter Real C = 1;
  parameter Real q_0 = 0;
  Real q(start = q_0);
equation
  der(q) = f;
  q = C*e;
  annotation(...);
end C;

model I
  extends DSFLib.BondGraph.Interfaces.PassiveOnePort;
  parameter Real I = 1;
  parameter Real p_0 = 0;
  Real p(start = p_0);
equation
  der(p) = e;
  p = I*f;
  annotation(...);
end I;

```

Con ideas muy similares puede crearse modelos activos correspondientes a las fuentes de flujo y esfuerzo. La Figura 6.26 muestra los componentes pasivos y activos correspondientes en su vista gráfica.

**Figura 6.26:** Componentes Pasivos y Activos de Bond Graph en Modelica.

6.4.2. Vínculos Estructurales, Transformadores y Giradores

Los vínculos estructurales 0 y 1 tienen un número variable de enlaces a los que pueden conectarse. Si bien esto se puede representar en el lenguaje Modelica, debido a limitaciones del entorno gráfico, es preferente utilizar un número fijo de conectores donde eventualmente quede algún conector sin enlace. El siguiente es el modelo de un vínculo 0 con cuatro conectores donde al final se verifica si algún enlace no tiene conexiones y en tal caso se establecen las ecuaciones de flujo igual a cero y se da un valor a la dirección (para que no quede indefinida).

Código 6.5: Modelo de un Vínculo 0 con 4 Conectores en Modelica

```

model J0
  DSFLib.BondGraph.Interfaces.BondCon bondCon1 annotation(...);
  DSFLib.BondGraph.Interfaces.BondCon bondCon2 annotation(...);
  DSFLib.BondGraph.Interfaces.BondCon bondCon3 annotation(...);
  DSFLib.BondGraph.Interfaces.BondCon bondCon4 annotation(...);
equation
  bondCon1.e = bondCon2.e;
  bondCon1.e = bondCon3.e;
  bondCon1.e = bondCon4.e;
  bondCon1.f*bondCon1.d + bondCon2.f*bondCon2.d + bondCon3.f*bondCon3.d + bondCon4.f*
    bondCon4.d = 0;

  if cardinality(bondCon1) == 0 then
    bondCon1.f = 0;
    bondCon1.d = 1;
  end if;
  if cardinality(bondCon2) == 0 then
    bondCon2.f = 0;
    bondCon2.d = 1;
  end if;
  if cardinality(bondCon3) == 0 then
    bondCon3.f = 0;
    bondCon3.d = 1;
  end if;
  if cardinality(bondCon4) == 0 then
    bondCon4.f = 0;
    bondCon4.d = 1;
  end if;
  annotation(...);
end J0;

```

En este modelo, el operador `cardinality` devuelve cuantos componentes están conectados a un conector. Si fuera necesario tener más de 4 enlaces en un vínculo 0, se pueden conectar dos vínculos 0 entre sí mediante un enlace.

Siguiendo la misma idea, se puede construir el modelo de un vínculo 1.

En cuanto a los transformadores y giradores, los modelos son también similares sólo que tendrán dos conectores y las leyes que relacionan las variables de esfuerzo y flujo serán diferentes.

La Figura 6.27 muestra los componentes multi-puerta correspondientes en su vista gráfica.

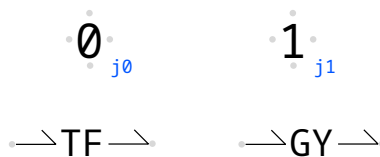


Figura 6.27: Componentes Multi-Puerta de Bond Graph en Modelica.

Utilizando estos componentes es simple construir modelos como por ejemplo el del Circuito de la

Figura 6.15, cuyo modelo en Modelica se muestra en la Figura 6.28. En el mismo se pueden ver los componentes y sus conexiones.

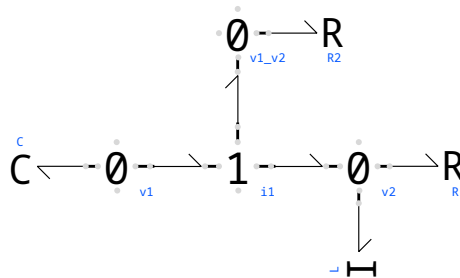


Figura 6.28: Bond Graph de un Circuito Eléctrico en Modelica.

6.4.3. Componentes Modulados

Incorporando un conector de entrada tipo `RealInput` es posible reemplazar el parámetro de un componente por una señal construyendo así un componente modulado. Por ejemplo, el siguiente código muestra el modelo de un resistor modulado, construido a partir de un modelo parcial pasivo modulado:

Código 6.6: Resistor Modulado de Bond Graph en Modelica

```

model ModPassiveOnePort
  extends DSFLib.BondGraph.Interfaces.PassiveOnePort;
  DSFLib.ControlSystems.Blocks.Interfaces.RealInput u annotation(...);
end ModPassiveOnePort;

model MR
  extends DSFLib.BondGraph.Interfaces.ModPassiveOnePort;
equation
  e = u*f;
  annotation(...);
end MR;
    
```

Utilizando este componente podemos construir el modelo Bond Graph correspondiente al sistema hidráulico de la Fig.6.18, según se muestra en la Figura 6.29.

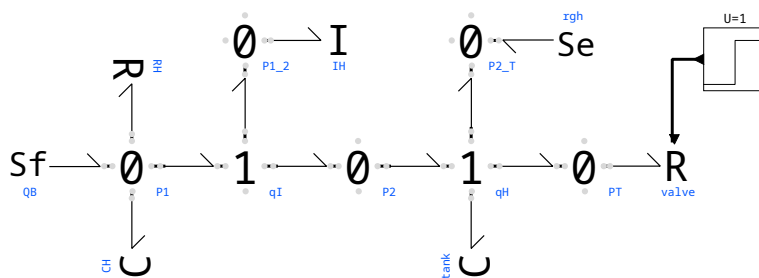


Figura 6.29: Bond Graph de un Sistema Hidráulico en Modelica.

6.4.4. Sensores

Para poder observar en señal las variables que forman parte de un Bond Graph se puede incluir algún tipo de modelo de sensor. Por ejemplo, la Figura 6.30 muestra el Bond Graph en Modelica correspondiente al modelo de un motor de corriente continua con excitación independiente que vimos en la Fig.6.14.

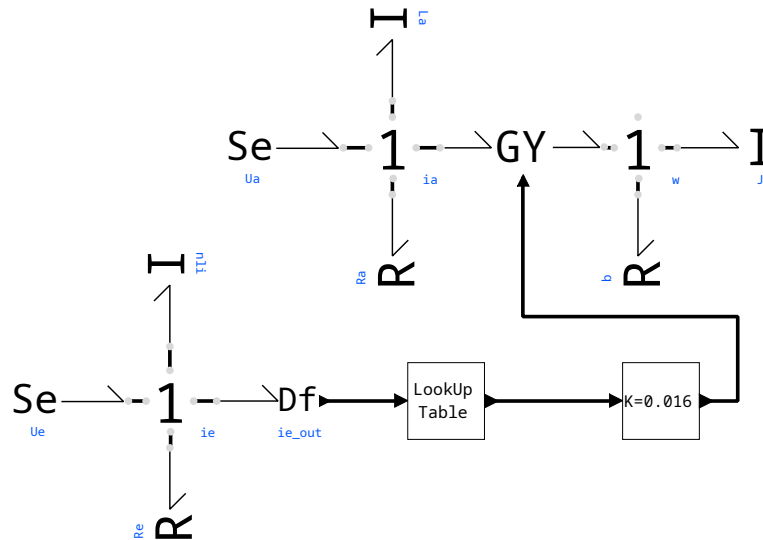


Figura 6.30: Bond Graph de un Sistema Electromecánico con Sensor de Flujo.

En este modelo vemos un *sensor de flujo* Df que se comporta como una fuente de esfuerzo nulo (para no afectar el vínculo 1) y cuenta con una salida tipo *RealOutput* donde sale la señal de flujo. Hay también un sensor de esfuerzo denominado De con un comportamiento análogo.

Un problema de este modelo es que el flujo magnético se calcula en dos lugares: en la inercia no lineal que representa el arrollamiento de excitación y en el bloque *LookUp Table* que genera la señal de flujo. Una opción más simple es agregar a la inercia no lineal un conector *RealOutput* por donde salga directamente la señal de flujo. En tal caso, el modelo resultante puede simplificarse notablemente como se muestra en la Figura 6.31.

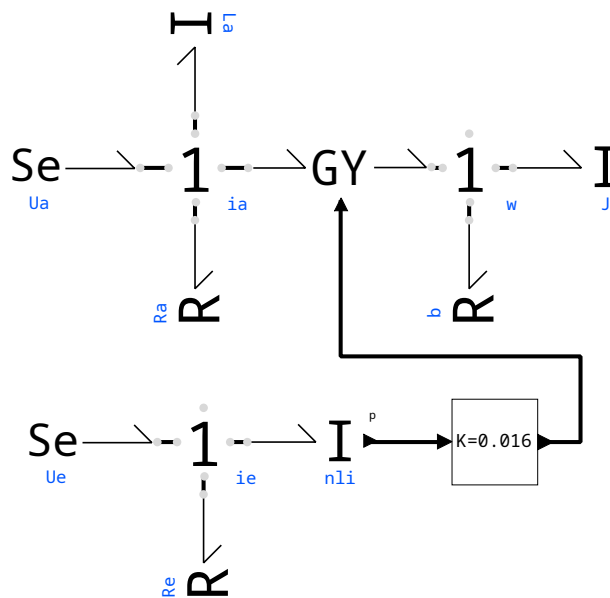


Figura 6.31: Bond Graph de un Sistema Electromecánico con Sensado del Momento.

6.4.5. Interfaz con Otras Librerías

Es posible también combinar modelos en Bond Graph con modelos desarrollados en base a la librería estándar de Modelica o a la DSFLib. Para esto bastará con crear componentes de interfaz que contengan conectores de ambos dominios con las ecuaciones adecuadas.

6.5. Ecuaciones de Euler-Lagrange

A lo largo del curso, siempre que trabajamos con sistemas mecánicos utilizamos como base directa o indirectamente la segunda ley de Newton: $\sum F_i(t) = m a(t)$. Incluso al trabajar con Bond Graphs a partir de principios energéticos, incluimos el elemento *Inercia* que en el caso mecánico representa esta ecuación en su relación constitutiva.

Veremos ahora un enfoque diferente que, también basado en principios energéticos, permite simplificar los problemas asociados al planteo de esta ecuación (particularmente las dificultades para determinar las fuerzas que actúan en sistemas en dos o tres dimensiones). Esta metodología, que da lugar a las ecuaciones e Euler-Lagrange, es muy utilizada en la práctica para obtener modelos de ecuaciones diferenciales de sistemas mecánicos complejos y es de gran utilidad, por ejemplo, para modelar manipuladores robóticos.

La metodología también sirve para modelar sistemas de otros dominios (electromecánicos, por ejemplo), pero en este curso nos limitaremos a los casos puramente mecánicos.

Comenzaremos introduciendo ejemplos para comprender en qué consiste el enfoque y luego justificaremos por qué este método funciona.

6.5.1. Ejemplos Introdutorios

Consideremos un sistema masa resorte (sin amortiguador) como el de la Figura 6.32, donde $q(t)$ indica la posición medida desde el equilibrio y $\dot{q}(t)$ la velocidad de la masa.

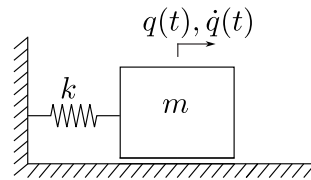


Figura 6.32: Sistema Masa Resorte.

Vamos a definir una función que denominaremos *Lagrangiano* del sistema, que calculará la diferencia entre la energía cinética y la potencial, esto es:

$$L(q, \dot{q}) \triangleq T(q, \dot{q}) - V(q, \dot{q}) \quad (6.25)$$

En este caso particular, dado que la energía cinética es $T = \frac{1}{2}m \dot{q}^2$ y la potencial elástica es $V = \frac{1}{2}k q^2$, el Lagrangiano resulta

$$L = \frac{1}{2}m \dot{q}^2 - \frac{1}{2}k q^2 \quad (6.26)$$

Una vez calculado el Lagrangiano, vamos a proponer la siguiente ecuación denominada *Ecuación de Euler Lagrange* y que más adelante analizaremos de dónde procede:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) = \frac{\partial L}{\partial q} \quad (6.27)$$

Particularizando entonces para el Lagrangiano de la Ec.(6.26), y teniendo en cuenta que $\frac{\partial L}{\partial \dot{q}} = m \dot{q}$ y que $\frac{\partial L}{\partial q} = -k q$, la Ec.(6.27) toma la siguiente forma:

$$\frac{d(m \dot{q}(t))}{dt} = -k q(t)$$

o equivalentemente,

$$m \ddot{q}(t) + k q(t) = 0$$

que puede verificarse fácilmente que es la ecuación que rige la dinámica del sistema masa resorte de la Figura 6.32.

El caso analizado es muy simple y lineal, y aparentemente (al menos en este caso) la ecuación de Euler-Lagrange parece representar correctamente el sistema. Veamos algunos casos más complejos entonces.

Consideremos ahora el péndulo de la Figura 6.33, en el cual q representa el ángulo con la vertical.

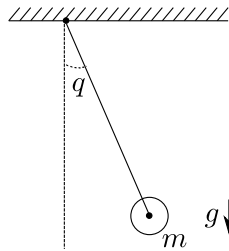


Figura 6.33: Péndulo Simple.

En este caso, la energía cinética la podemos calcular como $T = \frac{1}{2} J \dot{q}^2 = \frac{1}{2} m l^2 \dot{q}^2$, mientras que la energía potencial resulta $V = m g h = -m g l \cos(q)$. Luego, el Lagrangiano para este sistema resulta:

$$L = T - V = \frac{1}{2} m l^2 \dot{q}^2 + m g l \cos(q) \tag{6.28}$$

Teniendo en cuenta ahora que $\frac{\partial L}{\partial \dot{q}} = m l^2 \dot{q}$ y que $\frac{\partial L}{\partial q} = -m g l \sin(q)$, la Ec.(6.27) toma la siguiente forma:

$$\frac{d(m l^2 \dot{q})}{dt} = -m g l \sin(q) \tag{6.29}$$

o equivalentemente,

$$\ddot{q} + \frac{g}{l} \sin(q) = 0$$

que es la ecuación del péndulo simple que ya conocemos. Evidentemente la Ecuación de Euler-Lagrange sigue brindando un modelo correcto en este caso no lineal.

Veamos ahora que pasa si agregamos complejidad (y un grado más de libertad) al sistema, incluyendo elasticidad al cable como se muestra en Figura 6.34. En este nuevo esquema, la variable $q_2(t)$ representa la deformación del cable y la variable $q_1(t)$ el ángulo de dicho cable con la vertical. El cable sin deformar tiene longitud l .

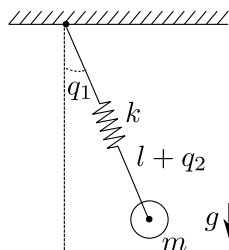


Figura 6.34: Péndulo con Cable Elástico.

En este caso podemos descomponer la energía cinética como la suma de la energía debida al movimiento radial y la debida al movimiento angular:

$$T = \frac{1}{2} m \dot{q}_2^2 + \frac{1}{2} J \dot{q}_1^2 = \frac{1}{2} m \dot{q}_2^2 + \frac{1}{2} m (l + q_2)^2 \dot{q}_1^2$$

Hay por otro lado energía potencial gravitatoria y elástica:

$$V = -m g (l + q_2) \cos(q_1) + \frac{1}{2} k q_2^2$$

de manera que el Lagrangiano toma la forma:

$$L = \frac{1}{2} m \dot{q}_2^2 + \frac{1}{2} m (l + q_2)^2 \dot{q}_1^2 + m g (l + q_2) \cos(q_1) - \frac{1}{2} k q_2^2 \quad (6.30)$$

Dado que hay dos *coordenadas* (q_1 y q_2), plantearemos una Ecuación de Euler-Lagrange como la Ec.(6.27) para cada una de ellas, esto es:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) = \frac{\partial L}{\partial q_i} \quad (6.31)$$

Para q_1 (el ángulo) tenemos

$$\frac{\partial L}{\partial \dot{q}_1} = m (l + q_2)^2 \dot{q}_1; \quad \frac{\partial L}{\partial q_1} = -m g (l + q_2) \sin(q_1)$$

y tras derivar la primera expresión respecto al tiempo, la Ec.(6.31) resulta

$$2 m (l + q_2) \dot{q}_2 \dot{q}_1 + m (l + q_2)^2 \ddot{q}_1 = -m g (l + q_2) \sin(q_1)$$

de donde,

$$2 \dot{q}_2 \dot{q}_1 + (l + q_2) \ddot{q}_1 + g \sin(q_1) = 0 \quad (6.32)$$

Para q_2 (deformación del resorte) resulta

$$\frac{\partial L}{\partial \dot{q}_2} = m \dot{q}_2; \quad \frac{\partial L}{\partial q_2} = m (l + q_2) \dot{q}_1^2 + m g \cos(q_1) - k q_2$$

Tras derivar la primera expresión respecto al tiempo, la Ecuación (6.31) para q_2 queda:

$$m \ddot{q}_2 = m (l + q_2) \dot{q}_1^2 + m g \cos(q_1) - k q_2 \quad (6.33)$$

que junto a la Ec.(6.32) completa el modelo del sistema. Puede verse también que estas ecuaciones son correctas. Por ejemplo, en la Ec.(6.33) que modela el movimiento en sentido radial de la masa, aparece la fuerza del resorte ($k q_2$), la fuerza que ejerce la gravedad en función del ángulo $m g \cos(q_1)$ y la fuerza centrípeta dependiente de la velocidad angular $m (l + q_2) \dot{q}_1^2$ (la coordenada radial es no inercial, y por lo tanto aparecen fuerzas virtuales).

6.5.2. El Principio de Acción Estacionaria

En los tres ejemplos que vimos hasta acá la Ecuación de Euler-Lagrange brindó un modelo correcto de la dinámica. Vamos a analizar ahora por qué.

Para esto, vamos a definir en primer lugar la siguiente magnitud, que llamaremos la *acción*:

$$S[q, t_1, t_2] \triangleq \int_{t_1}^{t_2} L(q, \dot{q}, t) dt \quad (6.34)$$

donde $L(q, \dot{q}, t)$ es el Lagrangiano del sistema. Notar que la acción S es un *funcional* ya que depende de la función $q(t)$ en el intervalo $[t_1, t_2]$.

Supongamos que sabemos que el sistema modelado por el Lagrangiano L en tiempo t_1 está en la posición $q(t_1)$ mientras que en t_2 está en la posición $q(t_2)$. Nos hacemos entonces la siguiente pregunta: ¿Cuál trayectoria de todas las que empiezan en $q(t_1)$ y terminan en $q(t_2)$ es la que sigue $q(t)$?

La respuesta a esta pregunta la da el *Principio de Acción Estacionaria*, que constituye uno de los principios fundamentales de la Física y que, para una partícula sujeta a *fuerzas conservativas*, establece lo siguiente:

Definición 6.1. *Principio de Acción Estacionaria*

La trayectoria que sigue una partícula produce un valor estacionario de la acción.

Para comprender este principio, tenemos en primer lugar que analizar el concepto de *valor estacionario* en un funcional. Para una función, un valor estacionario es el que ocurre en un mínimo local, en un máximo local o en un punto de ensilladura, es decir, en un punto en el cual si nos movemos infinitesimalmente, el valor de la función no cambia (la función tiene derivada nula).

En el caso de un funcional como la acción de la Ec.(6.34), un valor estacionario ocurre cuando al variar infinitesimalmente cierta trayectoria $q(t)$, la acción S permanece constante.

Supongamos entonces que $q_0(t)$ es la trayectoria que produce un valor estacionario de la acción S y definamos la siguiente trayectoria:

$$q_a(t) \triangleq q_0(t) + a \beta(t) \quad (6.35)$$

donde a es un número real y $\beta(t)$ es una función diferenciable cualquiera que cumple $\beta(t_1) = \beta(t_2) = 0$. De esta manera, la función $q_a(t)$, para cualquier valor de a , cumple con las condiciones $q_a(t_1) = q_0(t_1)$ y $q_a(t_2) = q_0(t_2)$. Además, si $a = 0$ resulta $q_a(t) = q_0(t)$.

Veamos entonces que condición debe cumplir L para que efectivamente $q_0(t)$ sea la trayectoria que produce un valor estacionario. Para esto, observemos que deberá cumplirse que

$$\frac{\partial}{\partial a} S[q_a(t), t_1, t_2] = 0 \quad \text{si } a = 0.$$

Evaluemos entonces la derivada anterior a partir de la definición de la acción dada por la Ec.(6.34).

$$\frac{\partial}{\partial a} S = \frac{\partial}{\partial a} \int_{t_1}^{t_2} L(q_a, \dot{q}_a, t) dt = \int_{t_1}^{t_2} \frac{\partial}{\partial a} L(q_a, \dot{q}_a, t) dt = \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q} \frac{\partial q_a}{\partial a} + \frac{\partial L}{\partial \dot{q}} \frac{\partial \dot{q}_a}{\partial a} \right) dt$$

De la Ec.(6.35) resulta $\frac{\partial q_a}{\partial a} = \beta(t)$. Además, $\dot{q}_a(t) = \dot{q}_0(t) + a \dot{\beta}(t)$ de donde $\frac{\partial \dot{q}_a}{\partial a} = \dot{\beta}(t)$. De esta forma,

$$\frac{\partial}{\partial a} S = \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q} \beta(t) + \frac{\partial L}{\partial \dot{q}} \dot{\beta}(t) \right) dt = \int_{t_1}^{t_2} \frac{\partial L}{\partial q} \beta(t) dt + \int_{t_1}^{t_2} \frac{\partial L}{\partial \dot{q}} \dot{\beta}(t) dt$$

Integrando por partes el último término, se obtiene

$$\int_{t_1}^{t_2} \frac{\partial L}{\partial \dot{q}} \dot{\beta}(t) dt = \frac{\partial L}{\partial \dot{q}} \beta(t) \Big|_{t_1}^{t_2} - \int_{t_1}^{t_2} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) \beta(t) dt = - \int_{t_1}^{t_2} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) \beta(t) dt$$

y reemplazando en la expresión de $\frac{\partial S}{\partial a} = 0$ se obtiene

$$\frac{\partial S}{\partial a} = \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q} \beta(t) - \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) \beta(t) \right) dt = \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q} - \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) \right) \beta(t) dt = 0$$

La única manera que esta ecuación se cumpla para cualquier función $\beta(t)$ arbitraria es que se cumpla:

$$\frac{\partial L}{\partial q} - \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) = 0$$

Esta expresión es equivalente a la Ecuación de Euler-Lagrange que habíamos definido en (6.27).

En otras palabras, la ecuación de Euler-Lagrange puede verse como el resultado de aplicar el *Principio de Acción Estacionaria* (a veces llamado Principio de Mínima Acción).

Si bien el análisis que presentamos aquí fue hecho sobre una partícula, el mismo se puede extender a sistemas de múltiples partículas y por lo tanto a cuerpos y sistemas mecánicos en general, dando lugar a la Ecuación (6.31) para cada coordenada de movimiento.

Un aspecto importante a tener en cuenta es que las Ecuaciones de Euler-Lagrange que se obtienen a partir del Lagrangiano de la Ec.(6.25) son válidas suponiendo que todas las fuerzas son conservativas (típicamente la gravedad y las fuerzas elásticas en sistemas mecánicos puros).

6.5.3. Coordenadas Generalizadas

En base a lo que ya sabemos respecto al orden de los sistemas mecánicos (dos veces los grados de libertad), al utilizar Euler-Lagrange es claro que debemos utilizar tantas coordenadas q_i como grados de libertad para describir el Lagrangiano. Por este motivo, en el primer ejemplo de la Figura 6.32 hay un sólo grado de libertad y adoptamos la coordenada q igual al desplazamiento de la masa (medida desde la posición de reposo). De manera similar, en el ejemplo del péndulo simple de la Figura 6.33 usamos el ángulo como única coordenada y en el péndulo con elasticidad de la Figura 6.34 utilizamos tanto el ángulo como la elongación del resorte ya que había dos grados de libertad.

Sin embargo, en todos los casos podríamos haber utilizado alternativamente otras coordenadas generalizadas y las ecuaciones seguirían aún siendo válidas. Por ejemplo, en el péndulo con elasticidad podríamos tomar directamente q_1 y q_2 como las posiciones horizontal y vertical. De esta manera, la energía cinética resulta

$$T = \frac{1}{2}m \dot{q}_1^2 + \frac{1}{2}m \dot{q}_2^2$$

mientras que la energía potencial es

$$V = m g q_2 + \frac{1}{2}k (\sqrt{q_1^2 + q_2^2} - l)^2$$

de manera que el Lagrangiano queda

$$L = T - V = \frac{1}{2}m \dot{q}_1^2 + \frac{1}{2}m \dot{q}_2^2 - m g q_2 - \frac{1}{2}k (\sqrt{q_1^2 + q_2^2} - l)^2$$

Para el movimiento horizontal, las derivadas parciales resultan

$$\frac{\partial L}{\partial q_1} = -k q_1 \frac{\sqrt{q_1^2 + q_2^2} - l}{\sqrt{q_1^2 + q_2^2}}; \quad \frac{\partial L}{\partial \dot{q}_1} = m \dot{q}_1$$

por lo que la Ecuación de Euler-Lagrange (6.31) resulta

$$m \ddot{q}_1 = -k (\sqrt{q_1^2 + q_2^2} - l) \frac{q_1}{\sqrt{q_1^2 + q_2^2}} \quad (6.36)$$

donde puede verse fácilmente que el lado derecho es la fuerza del resorte $k (\sqrt{q_1^2 + q_2^2} - l)$ proyectada en la dirección horizontal.

Repetiendo el procedimiento para el movimiento horizontal resulta

$$\frac{\partial L}{\partial q_2} = -m g - k q_2 \frac{\sqrt{q_1^2 + q_2^2} - l}{\sqrt{q_1^2 + q_2^2}}; \quad \frac{\partial L}{\partial \dot{q}_2} = m \dot{q}_2$$

y la Ecuación de Euler Lagrange correspondiente es

$$m \ddot{q}_2 = -m g - k (\sqrt{q_1^2 + q_2^2} - l) \frac{q_2}{\sqrt{q_1^2 + q_2^2}} \quad (6.37)$$

donde aparece ahora la gravedad y la proyección vertical de la fuerza del resorte.

En general, puede demostrarse fácilmente que al cambiar coordenadas mediante una transformación $\tilde{q} = R(q)$, y escribir el Lagrangiano en las nuevas coordenadas, las ecuaciones de Euler-Lagrange que se obtienen son las mismas que se hubieran obtenido haciendo el cambio de variables directamente sobre las ecuaciones de Euler-Lagrange originales.

6.5.4. Fuerzas no Conservativas

La presencia de fuerzas no conservativas puede incorporarse a las ecuaciones de Euler-Lagrange mediante agregando los términos de las *fuerzas generalizadas* Q_i que actúan en la dirección de la coordenada q_i . De esta manera, en presencia de dichas fuerzas las ecuaciones de Euler-Lagrange resultan:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = Q_i \quad (6.38)$$

Por ejemplo, si queremos considerar la presencia de una fuerza de fricción en el eje del péndulo simple $Q = b \dot{q}$, la Ecuación (6.29) ahora resulta

$$\frac{d(m l^2 \dot{q})}{dt} + m g l \sin(q) = -b \dot{q} \quad (6.39)$$

De manera similar podemos agregar fricción no lineal o fuerzas de entrada.

Veamos por ejemplo el puente grúa de la Figura 6.35. En el mismo, llamaremos q_1 a la posición del carro de masa m_c y q_2 al ángulo que forma la cuerda con la vertical.

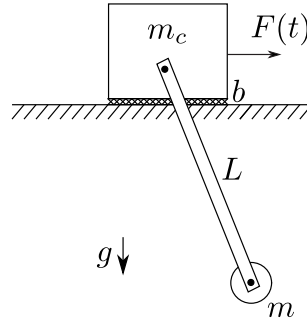


Figura 6.35: Puente Grúa con Fuerza Externa y Fricción.

La energía cinética la podemos escribir como

$$T = \frac{1}{2} m_c \dot{q}_1^2 + \frac{1}{2} m v_t^2$$

donde v_t es la velocidad tangencial de la carga, que puede descomponerse en las velocidades horizontal y vertical $v_t^2 = v_x^2 + v_y^2$. Para escribir v_x y v_y en función de las coordenadas q_1 y q_2 tendremos en cuenta que:

$$x = q_1 + L \sin(q_2) \implies v_x = \dot{x} = \dot{q}_1 + L \cos(q_2) \dot{q}_2$$

y

$$y = -L \cos(q_2) \implies v_y = \dot{y} = L \sin(q_2) \dot{q}_2$$

por lo que la energía cinética en función de las coordenadas generalizadas resulta

$$\begin{aligned} T &= \frac{1}{2} m_c \dot{q}_1^2 + \frac{1}{2} m (\dot{q}_1 + L \cos(q_2) \dot{q}_2)^2 + \frac{1}{2} m (L \sin(q_2) \dot{q}_2)^2 \\ &= \frac{1}{2} (m_c + m) \dot{q}_1^2 + m L \cos(q_2) \dot{q}_1 \dot{q}_2 + \frac{1}{2} m L^2 \dot{q}_2^2 \end{aligned}$$

La energía potencial, en tanto, es

$$V = m g y = -m g L \cos(q_2)$$

y el Lagrangiano queda

$$L = \frac{1}{2} (m_c + m) \dot{q}_1^2 + m L \cos(q_2) \dot{q}_1 \dot{q}_2 + \frac{1}{2} m L^2 \dot{q}_2^2 + m g L \cos(q_2)$$

Para la coordenada q_1 las derivadas parciales del Lagrangiano son

$$\frac{\partial L}{\partial q_1} = 0; \quad \frac{\partial L}{\partial \dot{q}_1} = (m_c + m) \dot{q}_1 + m L \cos(q_2) \dot{q}_2$$

En la dirección de la coordenada q_1 actúan las fuerzas $F(t)$ (entrada) y la fricción $F_f = -b \dot{q}_1$. Luego, la ecuación de Euler-Lagrange (6.38) queda

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_1} \right) - \frac{\partial L}{\partial q_1} = (m_c + m) \ddot{q}_1 + m L \cos(q_2) \ddot{q}_2 - m L \sin(q_2) \dot{q}_2^2 = F(t) - b \dot{q}_1 \quad (6.40)$$

Para la coordenada q_2 las derivadas parciales del Lagrangiano son

$$\frac{\partial L}{\partial q_2} = -(m L \dot{q}_1 \dot{q}_2 + m g L) \sin(q_2); \quad \frac{\partial L}{\partial \dot{q}_2} = m L \cos(q_2) \dot{q}_1 + m L^2 \dot{q}_2$$

En la dirección de q_2 no actúan fuerzas no conservativas, por lo que resulta

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_2} \right) - \frac{\partial L}{\partial q_2} = m L \cos(q_2) \ddot{q}_1 - m L \sin(q_2) \dot{q}_1 \dot{q}_2 + m L^2 \ddot{q}_2 + (m L \dot{q}_1 \dot{q}_2 + m g L) \sin(q_2) = 0 \quad (6.41)$$

De esta manera, queda un sistema de 4to. orden dado por las Ecs.(6.40)–(6.41).

6.5.5. Ecuaciones de Euler-Lagrange y DAEs

A partir de las Ecuaciones de Euler-Lagrange es muy simple obtener un sistema de DAEs. Para esto bastará con definir las variables de velocidad correspondientes a cada coordenada generalizada $v_1 \triangleq \dot{q}_1$, $v_2 \triangleq \dot{q}_2$, etc. y luego reemplazar con las mismas en las ecuaciones de Euler-Lagrange cada vez que aparezcan derivadas temporales. Esto es, deberemos reemplazar \dot{q}_i por v_i y \ddot{q}_i por \dot{v}_i (en rigor de verdad, alcanza con reemplazar las derivadas segundas). Al sistema de DAEs hay que agregarle también las definiciones de dichas velocidades.

Por ejemplo, para el caso de las Ecs.(6.40)–(6.41) se obtiene:

$$\begin{aligned} (m_c + m) \dot{v}_1 + m L \cos(q_2) \dot{v}_2 - m L \sin(q_2) v_2^2 &= F(t) - b v_1 \\ m L \cos(q_2) \dot{v}_1 - m L \sin(q_2) v_1 v_2 + m L^2 \dot{v}_2 + (m L v_1 v_2 + m g L) \sin(q_2) &= 0 \\ \dot{q}_1 &= v_1 \\ \dot{q}_2 &= v_2 \end{aligned} \quad (6.42)$$

En muchas ocasiones, con este procedimiento se puede despejar directamente las ecuaciones de estado. En este caso en particular esto no es posible, porque aparecen las derivadas de v_1 y v_2 en las dos primeras ecuaciones, lo que implica que hay un lazo algebraico. Por lo tanto, para obtener las Ecuaciones de Estado aquí deberíamos usar lo que vimos en el Capítulo 3.

6.5.6. Extensión a Otros Dominios

Si bien las ecuaciones de Euler-Lagrange surgieron de la mecánica clásica, las mismas pueden extenderse a otros dominios que van desde la mecánica relativista hasta los sistemas electromecánicos y electromagnético-mecánicos. En todos los casos basta con definir apropiadamente las coordenadas generalizadas y la energía cinética y potencial.

Una salvedad es que en todos los demás dominios en lugar de utilizar la energía cinética T se utiliza la coenergía cinética T^* . La energía cinética puede interpretarse como el área debajo de la curva p (cantidad de movimiento) vs. v (velocidad). Es decir, la integral $\int p(v)dv$. La coenergía cinética, en cambio, es el área a la izquierda de dicha curva, o sea la integral $\int v(p)dp$. Cuando hay una dependencia lineal entre la cantidad de movimiento y la velocidad (como ocurre en la mecánica clásica al suponer que la masa es constante con la velocidad) la energía y la coenergía cinética resultan equivalentes. Sin embargo, cuando la relación es no lineal (como ocurre en la mecánica relativista o ante la presencia de inductancias no lineales en sistemas electromecánicos) debe utilizarse la coenergía T^* para definir el Lagrangiano.

En este curso, de todas maneras, sólo trabajaremos con sistemas mecánicos bajo hipótesis no relativistas, por lo que el comentario anterior es sólo a título informativo.

6.6. Bibliografía Complementaria

Hay varios libros en la literatura que permiten profundizar sobre los temas abarcados en este capítulo. En cuanto a Bond Graphs, un libro clásico que aborda la dinámica de sistemas utilizando este formalismo de modelado es el de Karnopp, Margolis y Rosenberg [KMR12]. Otro libro clásico sobre modelado con

Bond Graphs es el de Thoma [Tho75]. Con resultados y extensiones más avanzadas, es también recomendable el libro de Borutzky [Bor10]. Por último, si bien no está dedicado exclusivamente a Bond Graphs, el libro de Cellier [Cel91] es muy didáctico.

Con respecto a Euler-Lagrange, lo desarrollado aquí sigue inicialmente el enfoque del libro de Morin [Mor08] (en el orden de presentación del método) pero también se utilizan conceptos (como la incorporación de fuerzas no conservativas) provenientes de otras fuentes. Hay libros con contenido más avanzado y formal donde, por ejemplo, se deducen las ecuaciones de Euler-Lagrange a partir de la mecánica clásica [GPS01]. Finalmente, el libro de Ortega [OPNSR13] trata con el uso de Euler-Lagrange en sistemas multidominio para aplicaciones de control y es muy recomendable como lectura de profundización.

6.7. Problemas Propuestos

[P6.1] Sistema con Palanca

Considerar nuevamente el sistema con palanca de la Figura 4.29 del Problema P5.3.

1. Obtener un modelo Bond Graph del sistema.
2. A partir del Bond Graph, obtener un sistema de DAEs.
3. Causalizar el Bond Graph y analizar el orden y la eventual presencia de lazos algebraicos y/o singularidades estructurales.
4. Construir un modelo en Modelica del sistema y simular utilizando distintos juegos de parámetros para verificar que el modelo sea correcto.

[P6.2] Sistema con Polea

Repetir el Problema P6.1 para el sistema de la Figura 5.31.

[P6.3] Convertidor Buck

Repetir el Problema P6.1 para el sistema del Problema P5.1 (Figura 5.28). Para simular, utilizar un resistor modulado y un bloque que genere la onda cuadrada con los valores R_{on} y R_{off} .

[P6.4] Modelo de un Montacargas

Considerar nuevamente el esquema de la Figura 5.36 del Problema P5.10. Se pide:

1. Construir un Bond Graph de la parte mecánica traslacional suponiendo que la polea actúa como una fuente ideal de fuerza sobre el cable.
2. Modificar el modelo anterior reemplazando la fuente de esfuerzo por el modelo Bond Graph del motor y la polea.
3. Analizar el orden del sistema físico y el orden del Bond Graph (para esto deberá causalizar).

[P6.5] Biela-Manivela

Considerar el sistema Biela-Manivela de la Figura 5.32. Se pide:

1. Obtener la expresión del módulo $K(\phi)$ de un transformador de Bond Graph que represente el mecanismo Biela-Manivela donde ϕ es el ángulo de la manivela. Dicho módulo deberá calcular la relación entre la velocidad angular de la manivela y la velocidad lineal del extremo de la biela. Para esto, se puede derivar respecto al tiempo la relación entre el ángulo y la posición del extremo de la biela dada por el Teorema del Coseno:

$$s(t)^2 + r^2 - 2 r s(t) \cos(\phi(t)) = L^2 \quad (\text{P6.5a})$$

donde L es el largo de la biela y r es el radio de la manivela.

2. Dibujar el Bond Graph del modelo completo utilizando un transformador asumiendo que internamente el componente tiene la expresión que calcula el módulo.
3. Construir en Modelica el modelo de dicho transformador de Bond Graph, que puede heredarse de la clase `TwoPort`. Internamente deberá calcular el ángulo $\phi(t)$ y en base a este la posición $s(t)$ y el módulo $K = \dot{s}(t)/\dot{\phi}(t)$. Este transformador tendrá como parámetros L y r .
4. Construir una solución alternativa utilizando un transformador modulado y calculando la señal $K(\phi)$ mediante un sensor de flujo, un integrador y un nuevo bloque que calcule la expresión de $K(\phi)$ en función de los parámetros L y r .

[P6.6] Bomba Aspirante-Impelente

Considerar nuevamente el sistema de la Figura 5.37 del Problema P5.11. Se pide,

1. Dibujar un Bond Graph del modelo suponiendo que la velocidad del pistón está determinada por una fuente ideal.
2. Construir en Modelica el modelo de un Resistor de Bond Graph que se corresponda con la válvula de una vía. Dicho componente tendrá como parámetros las resistencias R_{on} y R_{off} .
3. Usando el componente nuevo, construir el modelo completo del Bond Graph en Modelica donde la velocidad del pistón siga una trayectoria sinusoidal. Verificar el correcto funcionamiento. Es posible que en una de las válvulas deba invertir las resistencias R_{on} y R_{off} .

[P6.7] Plataforma Hidráulica

Para el sistema de la Figura 4.34 del Problema P4.17 se pide:

1. Construir un modelo Bond Graph. Tener en cuenta en el mismo que el elemento que representa el tanque trabaja a la diferencia de presión entre la parte superior y la inferior. Asumir además que las presiones $P_1(t)$ y $P_2(t)$ están impuestas externamente (o sea, son fuentes ideales).
2. Obtener el sistema de DAEs a partir del Bond Graph.
3. Analizar el orden del sistema físico y el del Bond Graph.

[P6.8] Ecuaciones de Euler-Lagrange de un Sistema con Dos Masas

Considerar el sistema de la Figura 6.36.

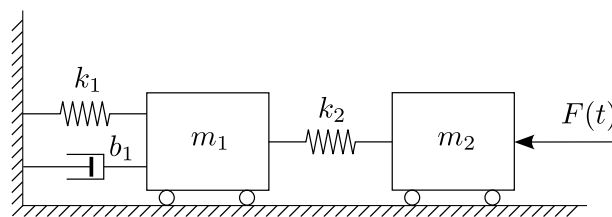


Figura 6.36: Sistema Mecánico con Dos Masas

1. Analizar el orden del sistema y proponer un juego de coordenadas generalizadas.
2. Escribir el Lagrangiano en función de las coordenadas generalizadas.
3. Ignorando la presencia del amortiguador y de la fuerza de entrada F , obtener las ecuaciones de Euler Lagrange.

4. Modificar las ecuaciones obtenidas en el punto anterior incorporando el amortiguador y la fuerza de entrada.
5. Obtener un modelo en forma de Ecuaciones de Estado.

[P6.9] Sistema con Restricción Cinemática

Considerar ahora el sistema de la Figura 6.17 en la página 174 y repetir el Problema P6.8 para el mismo. De ser posible, utilizar las mismas variables generalizadas.

[P6.10] Doble Péndulo

La Figura 6.37 muestra un péndulo doble. Se pide:

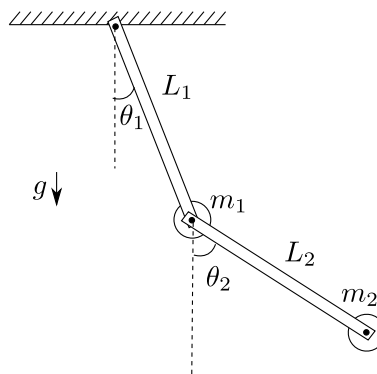


Figura 6.37: Péndulo doble

1. Calcular la posición absoluta (horizontal y vertical) de las masas m_1 y m_2 en función de los ángulos θ_1 y θ_2 .
2. Obtener una expresión para la energía potencial gravitatoria asociada a cada masa en función de los ángulos mencionados.
3. Calcular la energía cinética asociada a cada masa en función de dichos ángulos y sus derivadas.
4. Obtener las Ecuaciones de Euler Lagrange.
5. Obtener un sistema de Ecuaciones Diferenciales Algebraicas.
6. Obtener las Ecuaciones de Estado.

[P6.11] Sistema Biela-Manivela

Considerar nuevamente el sistema biela-manivela de la Figura 5.32 en la página 5.32. Se pide:

1. Calcular la posición y la velocidad de la masa en función del ángulo y la velocidad angular de la manivela.
2. Obtener la expresión del Lagrangiano usando como coordenada generalizada el ángulo de la manivela.
3. Obtener las Ecuaciones de Euler-Lagrange ignorando la presencia del amortiguador.
4. Modificar las ecuaciones considerando ahora la presencia del amortiguador.

Capítulo 7

Herramientas de Análisis de Sistemas Lineales

Como hemos mencionado en el Capítulo 1, los modelos matemáticos pueden utilizarse para realizar análisis experimental (simulaciones) y teórico. Cuando el modelo matemático del sistemas es lineal, existen importantes herramientas de análisis teórico que se pueden aplicar. En el Capítulo 8 se verá además que las herramientas de análisis de sistemas lineales pueden ser útiles también para sistemas no lineales, dado que el comportamiento de estos últimos puede aproximarse por el de un sistema lineal bajo ciertas restricciones.

En este capítulo nos dedicaremos a repasar algunas herramientas de análisis de sistemas lineales e invariantes en el tiempo (LTI según su sigla en inglés: *Linear Time-Invariant*), tanto para modelos internos (ecuaciones de estado, EE) como para modelos externos, es decir, entrada-salida (funciones transferencia, FT). Iremos en orden de complejidad creciente comenzando por los sistemas de orden 1, luego los de orden 2, para luego comentar las posibilidades de análisis de sistemas de orden superior.

7.1. Sistemas de Orden 1

7.1.1. Ecuaciones de Estado

Un sistema libre, es decir sin entradas, es un sistema donde la única causa que puede generar una evolución es una condición inicial. Las EE de un sistema lineal, invariante en el tiempo, libre y de orden 1 constan de una única ecuación diferencial escalar, de la forma

$$\dot{x}_1 = \lambda_1 x_1. \quad (7.1)$$

Numerosos sistemas de interés práctico (mecánicos, eléctricos, hidráulicos, térmicos) pueden ser modelados en la forma (7.1), como se puede ver en el Ejemplo de la Figura 7.1. La variable de estado para cada uno de estos sistemas se indica a través de su dependencia temporal en cada diagrama. Si se consideran relaciones constitutivas lineales, de acuerdo a lo visto en el Capítulo 4, las ecuaciones para cada sistema resultan:

$$\begin{aligned} \dot{i} &= -\frac{1}{RC}i, & \dot{v} &= -\frac{b}{m}v, & \dot{\omega} &= -\frac{b}{J}\omega, \\ \dot{h} &= -\frac{1}{R_H C}h, & \dot{\theta} &= -\frac{1}{RC}\theta, \end{aligned} \quad (7.2)$$

donde para el sistema hidráulico y térmico se definen $C := \frac{A}{\rho g}$ y $C := mc$, respectivamente, con A sección transversal del tanque, ρ densidad del líquido, g aceleración gravitatoria, m masa y c calor específico. Notar que cada una de las ecuaciones diferenciales en (7.2) tiene la forma (7.1) y que es sencillo determinar el valor del parámetro λ_1 (¡Ejercicio!), resultando $\lambda_1 < 0$ en cada caso.

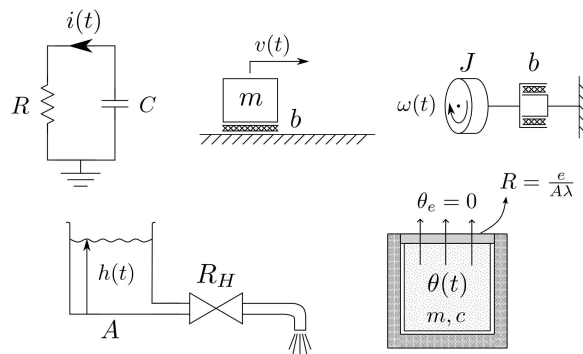


Figura 7.1: Ejemplos de sistemas de primer orden de diversos dominios físicos: eléctrico, mecánico traslacional y rotacional, hidráulico y térmico.

El espacio de estados de un sistema de orden 1 no es más que el eje real, \mathbb{R} . La EE (7.1), al ser escalar, se puede resolver fácilmente dando el resultado

$$x_1(t) = e^{\lambda_1 t} x_1(0), \quad \text{para todo } t.$$

Como se esperaba, la trayectoria $x_1(\cdot)$ depende de cuál sea la condición inicial $x_1(0)$. La forma cualitativa de la trayectoria, mejor dicho de cualquier trayectoria, depende del valor del coeficiente λ_1 :

- $\lambda_1 < 0$: Exponencial convergente, $\lim_{t \rightarrow \infty} x_1(t) = 0$;
- $\lambda_1 = 0$: Constante, $x_1(t) = x_1(0)$ para todo t ;
- $\lambda_1 > 0$: Exponencial divergente, $\lim_{t \rightarrow \infty} x_1(t) = \infty$.

Podemos dar una idea general de la evolución temporal para cada caso mediante el gráfico de la Figura 7.2. Este gráfico muestra cada caso de evolución desde la condición inicial $x_1(0) = 1$. Para cada uno de los

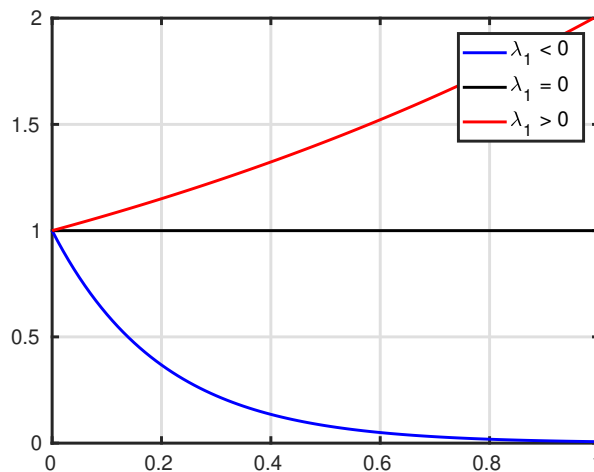


Figura 7.2: Evoluciones temporales de sistemas de orden 1 (el eje horizontal representa el tiempo).

tres casos (convergente, constante, divergente) podemos también dar una idea cualitativa de la evolución desde cualquier condición inicial graficando en el espacio de estados en lugar de temporalmente, como muestra la Figura 7.3 para el caso $\lambda_1 < 0$. Este gráfico indica que cualquier condición inicial positiva dará origen a una trayectoria de valor decreciente hasta llegar asintóticamente al origen; cualquier condición inicial negativa a una creciente hacia el origen y la condición inicial nula quedará constante en el origen. Similarmente podemos graficar los casos divergente (Figura 7.4) y constante (Figura 7.5).

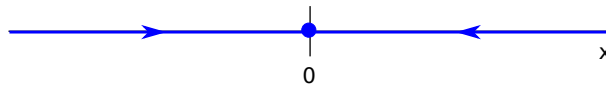


Figura 7.3: Evolución en el espacio de estado de un sistema de orden 1. Caso convergente.

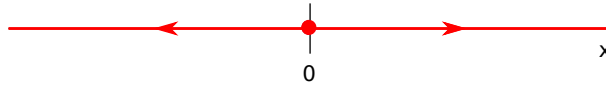


Figura 7.4: Evolución en el espacio de estado de un sistema de orden 1. Caso divergente.

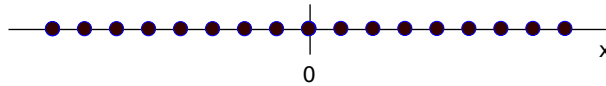


Figura 7.5: Evolución en el espacio de estado de un sistema de orden 1. Caso constante.

7.1.2. Función Transferencia

Si el sistema de orden 1 posee una entrada, el modelo EE (7.1) cambia por

$$\dot{x}_1 = \lambda_1 x_1 + b_{11} u_1 \quad (7.3)$$

y si, además, se identifica en el mismo una variable de salida, la EE (7.3) se complementa con

$$y_1 = c_{11} x_1 + d_{11} u_1. \quad (7.4)$$

Debido a la linealidad e invariancia temporal de estas ecuaciones, recordando que $\mathcal{L}\{\dot{f}(t)\} = s\mathcal{L}\{f(t)\} - f(0) = sF(s) - f(0)$, podemos transformar Laplace y obtener:

$$\begin{aligned} s X_1(s) - x_1(0) &= \lambda_1 X_1(s) + b_{11} U_1(s), \\ Y_1(s) &= c_{11} X_1(s) + d_{11} U_1(s). \end{aligned}$$

Despejando $X_1(s)$ en la primera ecuación, reemplazando en la segunda y operando podemos obtener fácilmente

$$\begin{aligned} Y_1(s) &= \frac{c_{11}}{s - \lambda_1} x_1(0) + \left[\frac{c_{11} b_{11}}{s - \lambda_1} + d_{11} \right] U_1(s) \\ &= \frac{c_{11}}{s - \lambda_1} x_1(0) + \frac{d_{11} s + (c_{11} b_{11} - \lambda_1 d_{11})}{s - \lambda_1} U_1(s) \end{aligned}$$

donde se observa claramente que la transformada de la salida contiene dos términos, uno que se anula si la condición inicial es nula y otro que se anula si la entrada es nula. El primer término corresponde entonces a la transformada de Laplace de la respuesta libre observada en la salida, mientras que el segundo a la de la respuesta forzada de la salida. **Siempre y cuando la condición inicial sea nula**, la relación entre la entrada y la salida puede escribirse en la forma

$$\begin{aligned} Y_1(s) &= \frac{d_{11} s + (c_{11} b_{11} - \lambda_1 d_{11})}{s - \lambda_1} U_1(s) \\ &= \frac{b_1 s + b_0}{s + a_0} U_1(s) =: G(s) U_1(s), \end{aligned} \quad (7.5)$$

donde $G(s)$ recibe el nombre de función transferencia (entre la entrada u_1 y la salida y_1), y vemos que para un sistema de orden 1 tiene la forma general

$$G(s) = \frac{b_1 s + b_0}{s + a_0}. \quad (7.6)$$

En el caso particular desarrollado, donde esta FT de orden 1 representa la relación entrada-salida del modelo (7.3)–(7.4), tenemos

$$b_1 = d_{11}, \quad b_0 = c_{11} b_{11} - \lambda_1 d_{11}, \quad a_0 = -\lambda_1.$$

Respuesta al escalón

Queremos evaluar ahora la evolución de la salida de un sistema LTI cuya FT es de la forma (7.6) cuando su entrada es un escalón unitario y las condiciones iniciales son nulas. De aquí en adelante, llamaremos $h(t)$ a esta evolución, es decir, a la “respuesta al escalón unitario” del sistema y $H(s)$ a su transformada de Laplace. Sabemos que si $G(s)$ es la FT de un sistema, entonces

$$H(s) = \underbrace{G(s)}_{\text{FT}} \cdot \underbrace{\frac{1}{s}}_{\text{Transf. del escalón}} \tag{7.7}$$

Para hallar la evolución temporal $h(t)$ podemos utilizar sin duda las herramientas ya conocidas sobre cómo antitransformar Laplace. Proponemos aquí un procedimiento cualitativo más ligado a nuestro conocimiento de sistemas: utilizar los teoremas de valor inicial (T.V.I.) y del valor final (T.V.F.), más el conocimiento del tipo de evolución temporal (por ejemplo, exponencial convergente). Planteamos entonces lo siguiente.

$$\begin{aligned} \text{(T.V.I.) } h(0^+) &= \lim_{s \rightarrow \infty} s \cdot H(s) = \lim_{s \rightarrow \infty} s \cdot G(s) \cdot \frac{1}{s} = G(\infty) \\ \text{(T.V.F.) } h(\infty) &= \lim_{s \rightarrow 0} s \cdot H(s) = G(0), \end{aligned}$$

donde hemos definido la notación

$$h(0^+) := \lim_{t \rightarrow 0^+} h(t), \quad h(\infty) := \lim_{t \rightarrow \infty} h(t), \quad G(\infty) := \lim_{s \rightarrow \infty} G(s).$$

Hay que tener en cuenta que la aplicación del T.V.F. sólo es válida si el límite temporal existe, de lo contrario el valor $G(0)$ puede existir aunque no exista $h(\infty)$.

Para el caso en que $G(s)$ tiene la forma general de primer orden (7.6), sabemos que la relación entrada-salida será BIBO-estable (es decir, que a toda entrada acotada corresponderá una salida acotada) cuando $a_0 > 0$. Esto corresponde a decir que el único polo de la FT es negativo, $\lambda_1 < 0$. En este caso, la evolución exponencial de la salida no diverge sino que evolucionará desde el valor inicial exponencialmente hacia el valor final. En este caso tenemos, utilizando (7.6),

$$h(\infty) = G(0) = \frac{b_0}{a_0}, \quad h(0^+) = G(\infty) = b_1,$$

y la constante de tiempo de la evolución exponencial es $\tau = 1/a_0$. Podemos ya graficar la evolución sin más cómputo, como se muestra en la Figura 7.6. El valor de τ para una evolución de primer orden puede obtenerse gráficamente mediante el denominado método de la tangente. El mismo se realiza trazando la tangente en algún punto de la evolución y computando la diferencia temporal entre el punto tangente y el punto donde corta la recta horizontal trazada en el valor final $h(\infty)$. Para la evolución de la Figura 7.6 se ha utilizado este hecho trazando la tangente en el punto en que la evolución vale $h(0^+)$ como ayuda para graficar la curva. Notar que el hecho de que las condiciones iniciales sean nulas implica que la salida es nula para tiempos negativos, pero si $b_1 \neq 0$, la salida dejará de ser nula para tiempos positivos arbitrariamente pequeños; eso es lo que indica $h(0^+) = b_1$. Mediante nuestro conocimiento de las técnicas para antitransformar Laplace (expansión en fracciones simples) podemos verificar que la respuesta al escalón unitario del sistema cuya FT es (7.6) es la siguiente:

$$h(t) = \underbrace{\frac{b_0}{a_0}}_{G(0)} + \left(\underbrace{b_1}_{h(0^+)} - \frac{b_0}{a_0} \right) \underbrace{e^{-a_0 t}}_{e^{\lambda_1 t} = e^{-t/\tau}}, \quad \text{para todo } t > 0 \text{ si } a_0 \neq 0.$$

Esta expresión es válida para cualesquiera valores reales de b_0 y b_1 , y siempre que $a_0 \neq 0$. Se puede verificar que si $a_0 < 0$, entonces $h(\infty) \neq G(0)$ (no se puede aplicar el T.V.F.). En el caso particular $a_0 = 0$, el polo de la FT (7.6) se ubica en el origen, resultando

$$G(s) = \frac{b_1 s + b_0}{s} = b_1 + \frac{b_0}{s}.$$

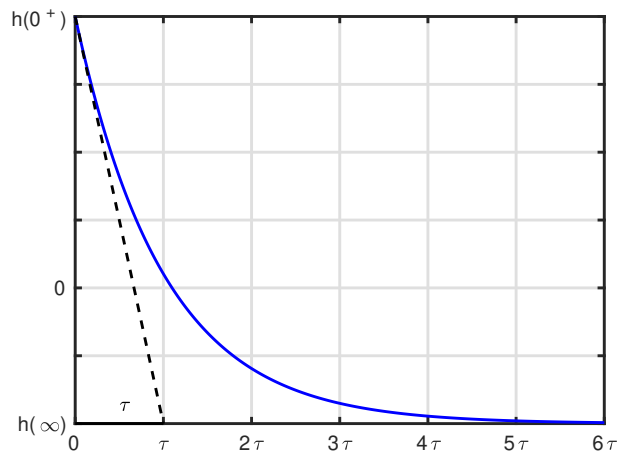


Figura 7.6: Ejemplo de respuesta al escalón estable de orden 1. En un intervalo correspondiente a una constante de tiempo τ , la respuesta recorre el 63% de la diferencia entre el valor inicial $h(0^+)$ y el valor final $h(\infty)$. Se ilustra el caso $h(0^+) > 0 > h(\infty)$.

Esta FT posee un término constante b_1 sumado a un término de acción integral $\frac{b_0}{s}$, de modo que la respuesta al escalón unitario será

$$h(t) = b_1 + b_0 t, \quad \text{para todo } t > 0 \text{ si } a_0 = 0.$$

Ejemplo 7.1. Dada la FT siguiente, graficar su respuesta al escalón unitario.

$$G(s) = \frac{-s + 1}{s + 1}.$$

En primer lugar, observamos que al ser el polo $\lambda_1 = -1$ negativo, la respuesta será convergente. Calculamos valor inicial, valor final y constante de tiempo:

$$h(0^+) = G(\infty) = -1, \quad h(\infty) = G(0) = 1, \quad \tau = 1/a_0 = 1$$

y en base a estos valores podemos graficar inmediatamente la respuesta, como muestra la Figura 7.7.

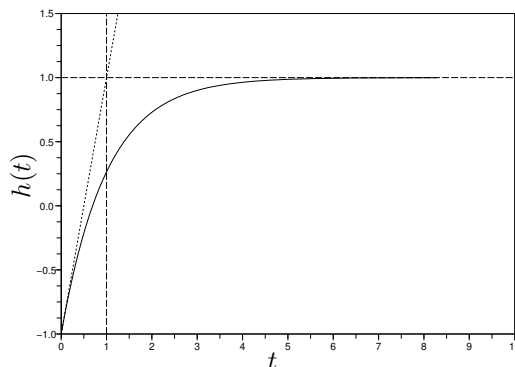


Figura 7.7: Respuesta al escalón unitario.

7.2. Sistemas de Orden 2

7.2.1. Ecuaciones de Estado

Un modelo de EE de un sistema LTI libre de orden 2 es el siguiente

$$\begin{aligned}\dot{x}_1 &= a_{11}x_1 + a_{12}x_2 \\ \dot{x}_2 &= a_{21}x_1 + a_{22}x_2\end{aligned}\tag{7.8}$$

que en su forma matricial equivale a

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x},\tag{7.9}$$

con

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.\tag{7.10}$$

A diferencia del caso escalar (orden 1) ya desarrollado, la solución de las EE (7.8) no se obtiene de forma tan directa sino que requiere algunos cálculos y operaciones adicionales. Sin embargo, podemos comenzar viendo algunos casos particulares de fácil solución, para luego ver cómo usar éstos en la solución del caso general.

Caso desacoplado (matriz \mathbf{A} diagonal)

Cuando la matriz \mathbf{A} es diagonal, es decir cuando tenemos

$$\begin{aligned}\dot{x}_1 &= \lambda_1 x_1, \\ \dot{x}_2 &= \lambda_2 x_2,\end{aligned}\quad \mathbf{A} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix},$$

podemos reconocer fácilmente que el sistema de orden 2 está formado en realidad por dos subsistemas de orden 1 que no interactúan entre sí. Esto se evidencia en que la ecuación de \dot{x}_1 depende únicamente de x_1 (y no de x_2) y lo análogo ocurre con \dot{x}_2 . En este caso tan particular, que recibe el nombre de caso desacoplado, podemos obtener la solución de forma directa:

$$\begin{aligned}x_1(t) &= e^{\lambda_1 t} x_1(0), \\ x_2(t) &= e^{\lambda_2 t} x_2(0),\end{aligned}\quad \underbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}}_{\mathbf{x}(t)} = \underbrace{\begin{bmatrix} e^{\lambda_1 t} & 0 \\ 0 & e^{\lambda_2 t} \end{bmatrix}}_{\Phi(t)} \underbrace{\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix}}_{\mathbf{x}(0)}.$$

Aquí arriba a la izquierda hemos escrito la solución componente por componente mientras que a la derecha indicamos la forma matricial de escribir la solución:

$$\mathbf{x}(t) = \Phi(t)\mathbf{x}(0).$$

La matriz variante en el tiempo $\Phi(t)$ recibe el nombre de *Matriz de Transición* del sistema. En la Figura 7.8 puede verse un ejemplo eléctrico de un sistema LTI de orden 2. El mismo consiste en dos mallas desacopladas que comparten una misma masa. Las EE para este sistema resultan

$$\begin{aligned}\dot{v}_C &= -\frac{1}{R_1 C} v_C, \\ \dot{i}_L &= -\frac{R_2}{L} i_L,\end{aligned}\quad \mathbf{A} = \begin{bmatrix} -\frac{1}{R_1 C} & 0 \\ 0 & -\frac{R_2}{L} \end{bmatrix}.$$

Similarmente a como procedimos en el caso escalar, podemos graficar la evolución temporal de la solución. Para cada componente tendremos una evolución cualitativamente distinta dependiendo del signo del coeficiente λ_i correspondiente ($i = 1$ o $i = 2$). La variedad de comportamientos cualitativos que puede exhibir el sistema de orden 2 como tal se puede observar con facilidad si graficamos la trayectoria no temporalmente sino en el espacio de estados, cuyos ejes representan los valores de las variables de estado, x_1 y x_2 . Un gráfico en el espacio de estados que ilustra trayectorias desde distintas condiciones iniciales y da idea de este comportamiento cualitativo recibe el nombre de **Retrato de Fase** del sistema.

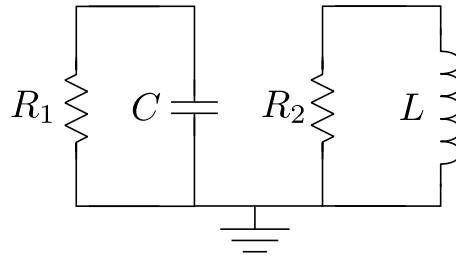


Figura 7.8: Sistema LTI de orden 2 desacoplado: dos mallas serie, una RC y otra RL , comparten la misma masa.

Retratos de Fase de un sistema libre de orden 2 desacoplado

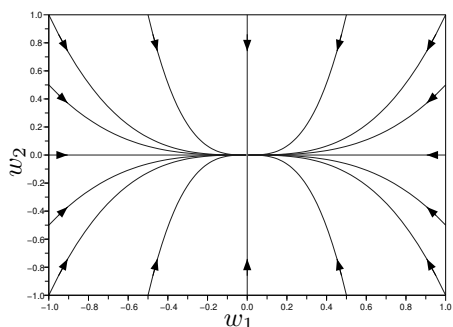
La Figura 7.9 muestra los distintos retratos de fase que puede exhibir un sistema de ecuaciones de estado de orden 2 desacoplado, de ecuaciones

$$\begin{aligned} \dot{w}_1 &= \lambda_1 w_1, \\ \dot{w}_2 &= \lambda_2 w_2, \end{aligned} \quad A = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix},$$

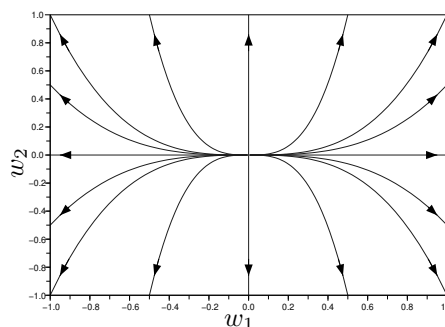
Las formas de estas evoluciones, es decir los tipos de retratos de fase, reciben los nombres siguientes:

- **Nodo Estable:** $\lambda_1 < 0$ y $\lambda_2 < 0$ con $\lambda_1 \neq \lambda_2$.
- **Nodo Inestable:** $\lambda_1 > 0$ y $\lambda_2 > 0$ con $\lambda_1 \neq \lambda_2$.
- **Ensilladura:** $\lambda_1 \lambda_2 < 0$ (uno positivo y el otro negativo).
- Estrella estable: $\lambda_1 = \lambda_2 < 0$.
- Estrella inestable: $\lambda_1 = \lambda_2 > 0$.
- Nodo estable degenerado: uno nulo y el otro negativo.
- Nodo inestable degenerado: uno nulo y el otro positivo.
- Constante: $\lambda_1 = \lambda_2 = 0$ (la matriz A es nula).

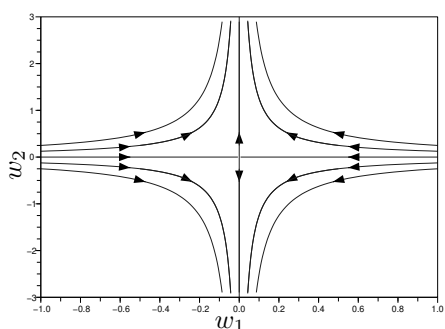
En esta lista se indican en **negrita** los más importantes por ser los más usuales.



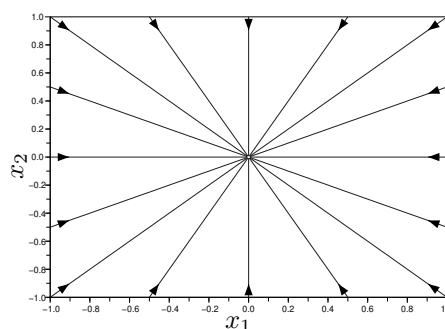
(a) **Nodo estable.** Se ilustra $\lambda_2 < \lambda_1 < 0$.



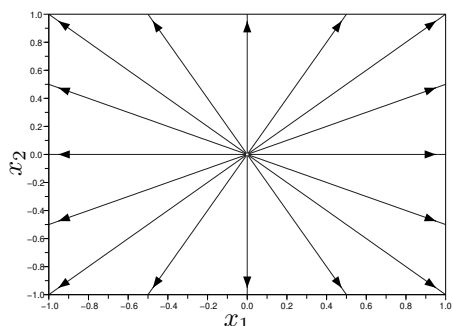
(b) **Nodo inestable.** Se ilustra $\lambda_2 > \lambda_1 > 0$.



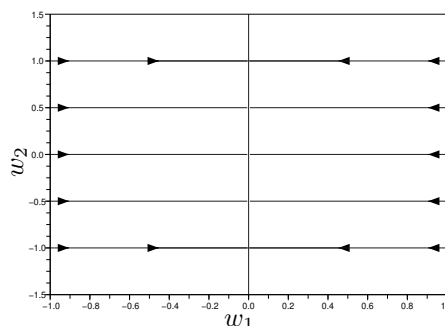
(c) **Enselladura.** Se ilustra $\lambda_1 < 0, \lambda_2 > 0$.



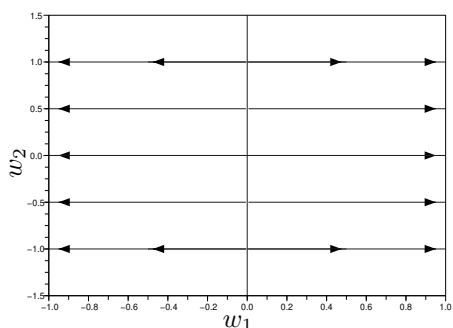
(d) **Estrella estable,** $\lambda_1 = \lambda_2 < 0$.



(e) **Estrella inestable,** $\lambda_1 = \lambda_2 > 0$.



(f) **Nodo estable degenerado** con $\lambda_1 < 0, \lambda_2 = 0$.



(g) **Nodo inestable degenerado** con $\lambda_1 > 0, \lambda_2 = 0$.

(h) **Constante,** $\lambda_1 = \lambda_2 = 0$.

Figura 7.9: Retratos de Fases de sistemas de orden 2 desacoplados.

Caso de oscilaciones

Además de los comportamientos ya vistos, los que pueden entenderse fácilmente interpretando al sistema de orden 2 como la combinación de subsistemas de orden 1, existen comportamientos propios de los sistemas de orden 2 que no poseen una descripción sencilla como superposición de sistemas de orden 1. Estos comportamientos corresponden al caso cuando las variables de estado presentan oscilaciones, cuya amplitud puede ser decreciente (oscilación amortiguada), constante (oscilación sostenida) o creciente (oscilación divergente). La forma más sencilla que pueden tener las EE en el caso oscilatorio es la siguiente

$$\begin{aligned} \dot{w}_1 &= \alpha w_1 - \omega w_2 \\ \dot{w}_2 &= \omega w_1 + \alpha w_2 \end{aligned} \quad \omega \neq 0, \quad (7.11)$$

que en forma matricial resulta

$$\dot{\mathbf{w}} = A \mathbf{w}, \quad \text{con} \quad A = \begin{bmatrix} \alpha & -\omega \\ \omega & \alpha \end{bmatrix}. \quad (7.12)$$

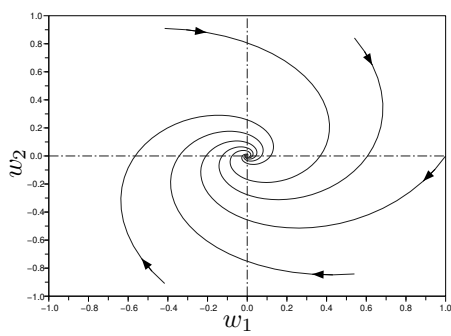
Pasando a coordenadas polares (ρ, θ) , donde ρ representa el módulo del vector \mathbf{w} y θ su ángulo con respecto a la horizontal, puede demostrarse que las EE (7.11) se transforman en

$$\begin{aligned} \dot{\rho} &= \alpha \rho, \\ \dot{\theta} &= \omega. \end{aligned}$$

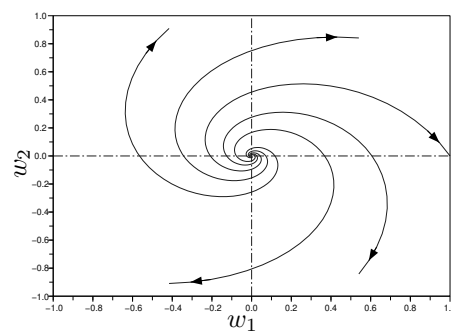
Estas ecuaciones pueden resolverse de manera directa, resultando

$$\begin{aligned} \rho(t) &= e^{\alpha t} \rho(0), \\ \theta(t) &= \theta(0) + \omega t. \end{aligned}$$

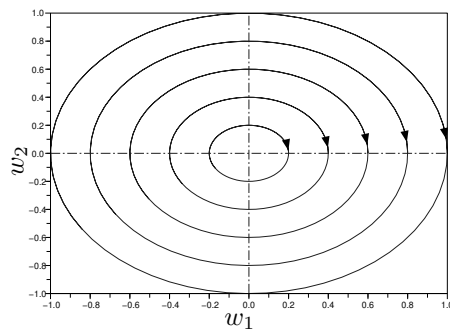
La primera ecuación indica que el módulo del vector de estado evoluciona exponencialmente en alguna de las formas antes vistas, dependiendo del signo del coeficiente α : $\alpha < 0$ para exponencial convergente, $\alpha = 0$ para evolución constante, $\alpha > 0$ para exponencial divergente. La segunda ecuación indica que el vector de estado rota a velocidad angular constante ω . La evolución del vector de estado combina estas dos situaciones: su módulo evoluciona exponencialmente mientras rota a velocidad constante. Los Retratos de Fase correspondientes se ilustran en la Figura 7.10.



(a) Foco estable.



(b) Foco inestable.



(c) Centro.

Figura 7.10: Retratos de Fases (modales) de sistemas de orden 2 oscilatorios.

Caso acoplado general

En general, la matriz A obtenida al modelar un sistema LTI de orden 2 no será de ninguna de las formas sencillas ya vistas. Baste como ejemplo un simple circuito RLC serie con valores arbitrarios de resistencia R , inductancia L y capacitancia C (¡Ejercicio!). La pregunta natural es cómo proceder para interpretar la evolución (es decir, la solución) del estado que satisface unas EE dadas en la forma general (7.8). La solución (luego mostraremos el desarrollo correspondiente) es la siguiente:

Se deben hallar los autovalores λ_1 y λ_2 de la matriz A :

- Si λ_1 y λ_2 son reales y distintos, el comportamiento corresponderá a alguno de los casos desacoplados en correspondencia con los signos de λ_1 y λ_2 ;
- Si λ_1 y λ_2 son complejos conjugados, el comportamiento corresponderá a alguno de los casos oscilatorios con $\alpha = \Re\{\lambda_1\}$ y $\omega = \pm \Im\{\lambda_1\}$;
- Si $\lambda_1 = \lambda_2$ y la matriz A es diagonal, se trata de un caso desacoplado (estrella).
- Si $\lambda_1 = \lambda_2$ y la matriz A no es diagonal, el comportamiento se verá más adelante (casos Jordan).

Recordemos que los autovalores de una matriz A son las soluciones de la ecuación:

$$\det(\lambda I - A) = 0$$

En la mayoría de los casos entonces, la forma cualitativa de la evolución en el espacio de estados de un sistema LTI libre de orden 2 está determinada por los autovalores λ_1 y λ_2 de la matriz A . La diferencia con respecto al caso desacoplado se observará en que las evoluciones aparecerán como ‘torcidas’ o, mejor dicho, ‘linealmente transformadas’ con respecto a como se ven las evoluciones del caso desacoplado. La Figura 7.11 ilustra estos retratos de fase en el caso de autovalores reales; la Figura 7.12 lo hace para el caso de autovalores complejos conjugados no reales. En cada uno de los Retratos de Fase de la Figura 7.11 existen algunas trayectorias contenidas en rectas que pasan por el origen. Estas rectas reciben el nombre de **direcciones invariantes bajo la dinámica del sistema** ya que una condición inicial sobre esa recta da origen a una trayectoria íntegramente contenida en la misma recta. Estas direcciones se corresponden con las direcciones de los **autovectores** de la matriz A . Por el contrario, los Retratos de Fase de la Figura 7.12 no poseen direcciones invariantes. Las direcciones indicadas con líneas de trazo se corresponden en cambio a los vectores parte real y parte imaginaria de los autovectores de A , ya que en este caso de autovalores complejos, los autovectores correspondientes son vectores de números complejos. En la Sección 7.3 daremos el desarrollo teórico y justificación correspondiente.

7.2.2. Función Transferencia

Una FT general de orden 2 es de la forma

$$G(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^2 + a_1 s + a_0}. \quad (7.13)$$

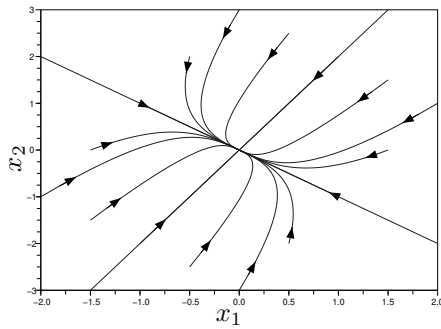
Recordemos que una FT modela la relación entre una entrada y una salida de un sistema LTI y se puede pensar como el cociente entre las transformadas de Laplace de la salida y la entrada cuando las condiciones iniciales son nulas. Nos interesa aquí relacionar la FT (7.13) con la respuesta $h(t)$ al escalón unitario. Ya hemos mencionado que esto se puede hacer mediante las herramientas y los métodos conocidos sobre cómo antitransformar Laplace, pero lo que buscamos aquí es combinar conocimiento cualitativo sobre la forma de la evolución, junto con algunos cálculos cuantitativos para entender más a fondo las características del sistema.

Comenzamos calculando los polos p_1 y p_2 de la FT, los cuales son las raíces de

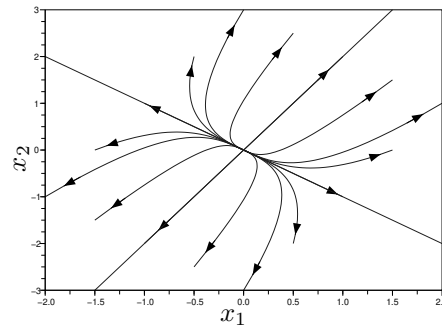
$$s^2 + a_1 s + a_0 = 0,$$

de modo que podemos escribir

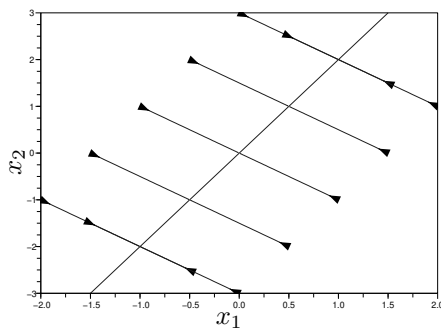
$$s^2 + a_1 s + a_0 = (s - p_1)(s - p_2) = s^2 - (p_1 + p_2)s + p_1 p_2. \quad (7.14)$$



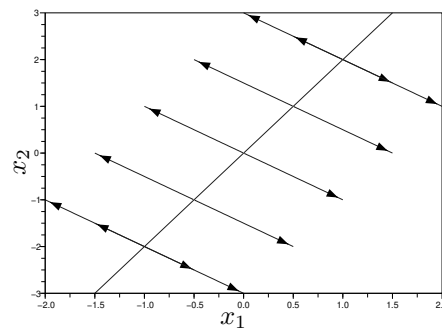
(a) Nodo estable.



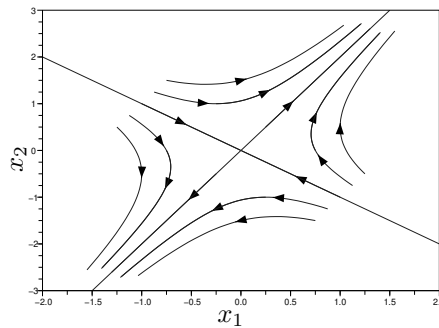
(b) Nodo inestable.



(c) Nodo estable degenerado.



(d) Nodo inestable degenerado.



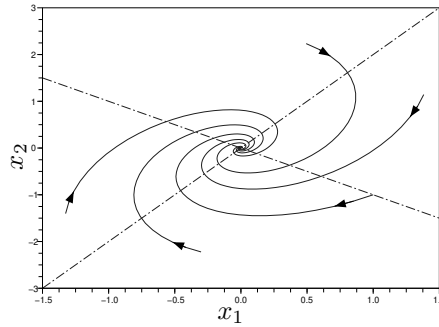
(e) Ensilladura.

Figura 7.11: Retratos de Fases de sistemas que se pueden desacoplar, con autovalores reales.

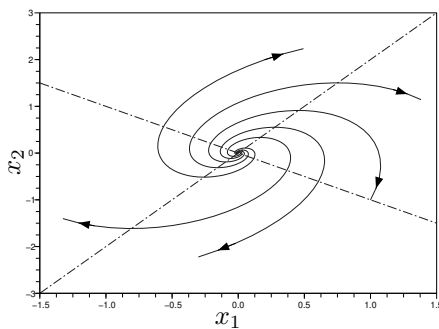
Sabemos que p_1 y p_2 caracterizan cualitativamente la forma de la evolución temporal, de manera similar a como los autovalores de la matriz A lo hacen en el caso libre. Por lo tanto, el caso más interesante a tratar aquí es el que se puede dar en sistemas de orden 2 pero que no posee una interpretación sencilla como superposición de evoluciones de orden 1. Esto corresponde al caso en que p_1 y p_2 son complejos conjugados no reales.

Caso oscilatorio amortiguado

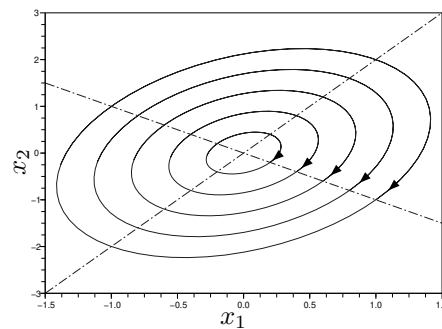
El caso de polos complejos conjugados no reales, $p_1, p_2 = \alpha \pm j\omega$ con $\omega \neq 0$, da lugar a una respuesta temporal oscilatoria, con amplitud de evolución exponencial relacionada con α y frecuencia de oscilación



(a) Foco estable.



(b) Foco inestable.



(c) Centro.

Figura 7.12: Retratos de Fases de sistemas con autovalores complejos conjugados no reales.

ω . Más precisamente, la respuesta al escalón en este caso es de la forma

$$h(t) = H_\infty + Ke^{\alpha t} \sin(\omega t + \phi), \quad t > 0, \tag{7.15}$$

con H_∞ , K y ϕ ciertos coeficientes reales. El caso de oscilaciones amortiguadas es el caso BIBO-estable, y se da para $\alpha < 0$. Notar que los polos determinan: (i) la velocidad o tasa exponencial a la que se extinguen (o crecen) las oscilaciones mediante su parte real α y (ii) la frecuencia de la oscilación amortiguada mediante su parte imaginaria ω . Como los polos dependen únicamente del denominador de la FT (7.13) y están determinados por los coeficientes a_0 y a_1 [recordar (7.14)], entonces **los parámetros b_0 , b_1 y b_2 del numerador no afectan la forma cualitativa de la oscilación amortiguada** en cuanto a su velocidad de amortiguamiento y frecuencia de oscilación, ya que **esta forma está determinada exclusivamente por el denominador de la FT a través de sus polos.**

Determinación de los polos mediante la respuesta al escalón

Supongamos tener una respuesta al escalón unitario de una FT de orden 2 cuya respuesta tiene forma cualitativa de oscilación amortiguada. Se desea obtener los polos de la FT cuya respuesta se observa. Con este fin, identificamos primero en la respuesta el valor final $h(\infty)$, que de acuerdo a (7.15) coincide con el coeficiente H_∞ porque $\alpha < 0$ hace que la exponencial se extinga para $t \rightarrow \infty$. Luego, observamos que la respuesta, al ser oscilatoria, cruza varias veces este valor, en distintos instantes. De (7.15) los instantes t_1, t_2 , etc. en que $h(t_i)$ satisface $h(t_i) = H_\infty$ se dan únicamente cuando $\sin(\omega t_i + \phi) = 0$. Siendo t_i y t_{i+1} dos instantes consecutivos en los que esto sucede, debe ser cierto que

$$\sin(\omega t_i + \phi + \pi) = \sin(\omega t_{i+1} + \phi) = 0,$$

de modo que $t_{i+1} = t_i + \frac{\pi}{\omega}$. La distancia temporal entre t_i y t_{i+1} , dada por $t_{i+1} - t_i$, corresponde a un semiperíodo de la oscilación. Podemos determinar fácilmente la frecuencia de oscilación amortiguada ω

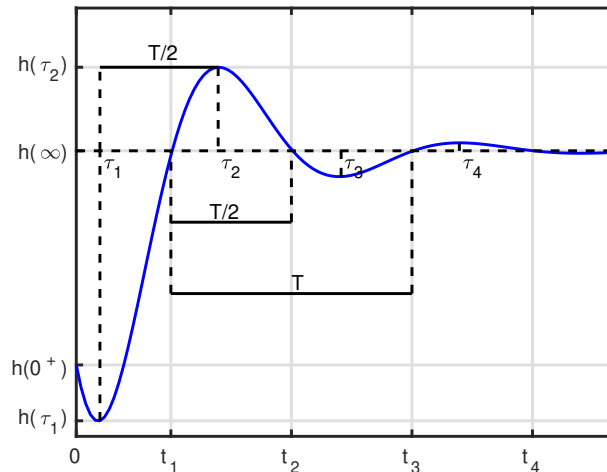


Figura 7.13: Respuesta al escalón de una FT BIBO-estable de orden 2 con polos complejos conjugados. Se indican tiempos t_1, t_2, t_3 y t_4 de cruce por el valor final $h(\infty)$, así como tiempos $\tau_1, \tau_2, \tau_3, \tau_4$ de desviaciones máximas con respecto al valor final. Se observa que $t_{i+1} - t_i = \tau_{i+1} - \tau_i = T/2$, para $i = 1, 2, 3$.

de la siguiente forma:

$$\omega = \frac{\pi}{t_{i+1} - t_i} = \frac{\pi}{T/2} = \frac{2\pi}{T},$$

donde hemos definido T como el (pseudo) período de la oscilación observada.

Se puede demostrar que no sólo la distancia entre cruces sucesivos del valor final es igual a un semiperíodo de oscilación, sino también la distancia entre desviaciones máximas sucesivas con respecto al valor final [simplemente analizando la derivada de $h(t)$]. En la respuesta, podemos identificar fácilmente también las desviaciones máximas con respecto al valor final, las que se dan cuando el término $Ke^{\alpha t} \sin(\omega t + \phi)$ en (7.15) llega a un máximo (positivo) o a un mínimo (negativo). Sean τ_i y $\tau_{i+1} = \tau_i + \pi/\omega$ instantes arbitrarios, separados un semiperíodo, pero en los cuales no se cruce justo el valor final. Suele ser conveniente que estos instantes correspondan a desviaciones máximas consecutivas (una por encima y otra por debajo) del valor final. Utilizando (7.15) tendremos entonces

$$\begin{aligned} h(\tau_{i+1}) - H_\infty &= Ke^{\alpha\tau_{i+1}} \sin(\omega\tau_{i+1} + \phi) = Ke^{\alpha\tau_i} e^{\alpha\pi/\omega} \sin(\omega\tau_i + \pi + \phi) \\ &= e^{\alpha\pi/\omega} [-Ke^{\alpha\tau_i} \sin(\omega\tau_i + \phi)] = -e^{\alpha\pi/\omega} [h(\tau_i) - H_\infty]. \end{aligned}$$

Tomando valor absoluto y dividiendo, obtenemos

$$\frac{|h(\tau_{i+1}) - H_\infty|}{|h(\tau_i) - H_\infty|} = e^{\alpha\pi/\omega} \tag{7.16}$$

de donde ya conociendo ω podemos hallar α . La expresión a la izquierda del igual representa cuánto se amortigua la oscilación a lo largo de un semiperíodo. La Figura 7.13 ilustra los instantes y valores característicos mencionados.

Caso oscilatorio, forma usual del denominador

Cuando los polos son complejos conjugados, suele expresarse el denominador de la FT de la siguiente forma

$$s^2 + a_1s + a_0 = s^2 + 2\xi\omega_n s + \omega_n^2, \tag{7.17}$$

donde $a_0 > 0$ necesariamente para tener polos complejos conjugados (¿Por qué?). La constante ω_n recibe el nombre de *frecuencia de oscilación natural o no amortiguada*, mientras que ξ el de *coeficiente de amortiguamiento*. Siendo $p_{1,2} = \alpha \pm j\omega$ los polos (con $\omega > 0$ sin pérdida de generalidad), recordando (7.14)

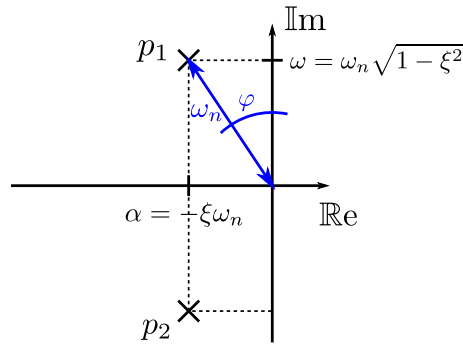


Figura 7.14: Polos complejos conjugados $p_{1,2} = \alpha \pm j\omega$. Se indican las relaciones con la frecuencia no amortiguada ω_n y el coeficiente de amortiguamiento ξ . La frecuencia ω_n puede verse como el módulo de cada polo, ya que $|p_1|^2 = |p_2|^2 = \alpha^2 + \omega^2 = \omega_n^2$. El ángulo φ depende únicamente de ξ , ya que $\tan \varphi = |\alpha|/\omega = \xi/\sqrt{1-\xi^2}$.

y operando obtenemos la relación entre a_0 , a_1 , ξ , ω_n y la parte real α e imaginaria ω de los polos:

$$\begin{aligned} a_1 &= 2\xi\omega_n = -2\alpha, \\ a_0 &= \omega_n^2 = \alpha^2 + \omega^2, \end{aligned}$$

o también, poniendo parte real e imaginaria en función de ξ y ω_n :

$$\alpha = -\xi\omega_n, \quad (7.18a)$$

$$\omega = \sqrt{\omega_n^2 - \alpha^2} = \omega_n\sqrt{1-\xi^2}. \quad (7.18b)$$

De esto último podemos ver que debe ser $\xi^2 < 1$ para que los polos sean efectivamente complejos conjugados (es decir, para que ω resulte real, como fue planteado desde el inicio). También de la ecuación de α , tenemos los casos: $\alpha < 0$ para $0 < \xi < 1$ (caso BIBO-estable), $\alpha = 0$ para $\xi = 0$ (polos imaginarios puros), y $\alpha > 0$ para $-1 < \xi < 0$ (oscilación divergente). Asimismo, podemos ver que el cociente entre la parte real e imaginaria de los polos da como resultado

$$\frac{\alpha}{\omega} = \frac{-\xi}{\sqrt{1-\xi^2}}, \quad (7.19)$$

expresión que depende únicamente de ξ . Reemplazando (7.19) en (7.16), se obtiene que el amortiguamiento en un semiperíodo es

$$\frac{|h(\tau_{i+1}) - H_\infty|}{|h(\tau_i) - H_\infty|} = e^{\alpha\pi/\omega} = e^{-\pi\xi/\sqrt{1-\xi^2}}. \quad (7.20)$$

Esto justifica el nombre **coeficiente de amortiguamiento** ξ . De (7.18) vemos también que $\omega_n = \omega$ cuando $\xi = 0$, es decir que ω_n representa la frecuencia de oscilación observada únicamente en el caso de inexistencia de amortiguamiento. Esto justifica el nombre **frecuencia de oscilación no amortiguada** ω_n . Las relaciones entre los polos y los valores característicos ω_n y ξ se ilustran en la Figura 7.14.

Determinación del numerador de la FT

Sean cuales sean los polos, podemos aplicar los teoremas de valor final y valor inicial a la FT (7.13), como lo hicimos para orden 1:

$$h(0^+) = G(\infty) = b_2, \quad y \quad (7.21)$$

$$\text{si existe } h(\infty), \text{ entonces } h(\infty) = G(0) = \frac{b_0}{a_0}. \quad (7.22)$$

Recordemos también que si p_1 y p_2 poseen parte real α negativa, entonces la FT es BIBO-estable y el límite $h(\infty)$ existirá. La ecuación (7.21) indica que el coeficiente b_2 del numerador representa la amplitud

del salto o discontinuidad que posee la respuesta al escalón $h(t)$ en $t = 0$: si $b_2 \neq 0$, entonces $h(0^-) = 0$ (condiciones iniciales nulas) será distinto que $h(0^+) = b_2$. También observamos que el valor final $h(\infty)$ está relacionado con b_0 . Habiendo determinado a_0 en el análisis del denominador, de (7.22) podemos despejar b_0 .

Resta relacionar el valor de b_1 . Una forma es aplicar el T.V.I. a la derivada de la respuesta al escalón, de la siguiente forma. Si $H(s)$ representa la transformada de Laplace de $h(t)$, recordando (7.7), (7.21) y las propiedades de la transformada de la derivada, podemos hallar

$$\mathcal{L}\{\dot{h}(t)\} = sH(s) - h(0^+) = G(s) - b_2 \quad (7.23)$$

$$\begin{aligned} \dot{h}(0^+) &= \lim_{s \rightarrow \infty} s\mathcal{L}\{\dot{h}(t)\} = \lim_{s \rightarrow \infty} s[G(s) - b_2] = \lim_{s \rightarrow \infty} \frac{s[b_2s^2 + b_1s + b_0 - b_2(\cancel{s^2} + a_1s + a_0)]}{s^2 + a_1s + a_0} \\ &= \lim_{s \rightarrow \infty} \frac{(b_1 - b_2a_1)s^2 + (b_0 - b_2a_0)s}{s^2 + a_1s + a_0} = b_1 - b_2a_1. \end{aligned} \quad (7.24)$$

Habiendo determinado a_1 al analizar el denominador y b_2 analizando el salto en 0 como indica (7.21), podemos medir la derivada por derecha en 0 y obtener b_1 despejando de la última ecuación.

Ejemplo 7.2. La Figura 7.15 ilustra la respuesta al escalón unitario de un sistema LTI de orden 2. Se desea hallar la FT de este sistema.

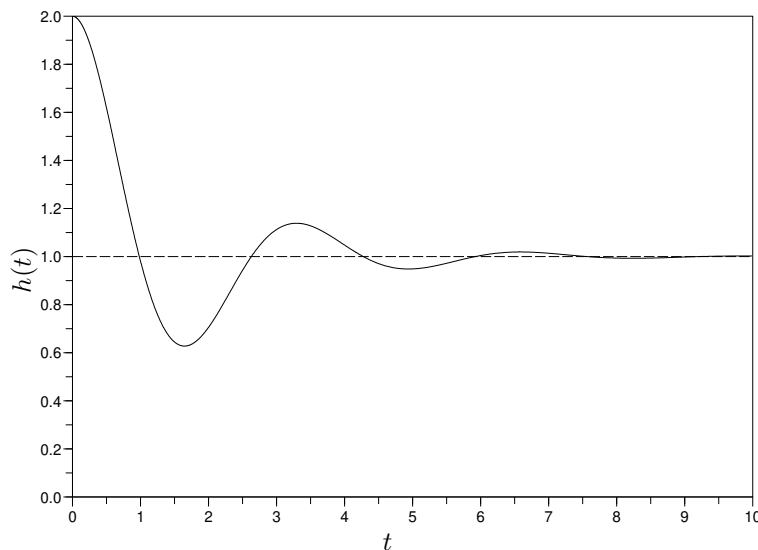


Figura 7.15: Respuesta al Escalón Unitario de un sistema LTI de orden 2.

Solución: Sabemos que una FT genérica de orden 2 es de la forma (7.13). Más aún, como la evolución de la Figura 7.15 es oscilante amortiguada, sabemos que los polos de esta FT deben ser complejos conjugados con parte real negativa. De la gráfica relevamos muy fácilmente lo siguiente: $h(0^+) = 2$, $h(\infty) = 1$. Determinamos en primer lugar el denominador de la FT, es decir, los coeficientes $a_0 = \omega_n^2$ y $a_1 = 2\xi\omega_n$, donde hemos recordado (7.17). Sobre la figura podemos determinar el período de la (pseudo)oscilación evaluando tiempos sucesivos en que se cruza el valor final, o bien en que la desviación con respecto al valor final es un extremo. Obtenemos de esta forma un período $T = 3.3$ seg, aproximadamente. De esto podemos calcular inmediatamente la parte imaginaria de los polos, que está dada por la frecuencia de esta oscilación en rad/seg:

$$\omega = \frac{2\pi}{T} \approx 1.9.$$

A continuación, determinamos el amortiguamiento a lo largo de un semiperíodo. Para esto conviene, como se dijo anteriormente, determinar instantes en que la desviación con respecto al valor final es un

extremo (máxima o mínima). Estos instantes corresponden a extremos de $h(t)$, es decir a puntos donde $\dot{h}(t) = 0$. Observando la figura, observamos (medimos bien, en caso de disponer de los datos numéricos) que la derivada por derecha en tiempo 0 satisface $\dot{h}(0^+) = 0$, lo que implica que $t = \tau_1 = 0$ será uno de los tiempos en los cuales el desvío con respecto al valor final será un máximo o un mínimo (un máximo en este caso) y que el instante inmediato siguiente donde esto sucede es $\tau_2 = T/2$. Usando la fórmula (7.20) con $i = 1$, calculamos

$$\frac{|h(\tau_2) - H_\infty|}{|h(\tau_1) - H_\infty|} = \frac{|0.63 - 1|}{|2 - 1|} = 0.37 = e^{-\pi\xi/\sqrt{1-\xi^2}}$$

de donde puede despejarse que $\xi \approx 0.3$. Con estos valores, podemos hallar

$$\omega_n = \frac{\omega}{\sqrt{1-\xi^2}} \approx 2.$$

Sabemos ahora entonces que el denominador de la FT es $s^2 + 2 \cdot 0.3 \cdot 2 s + 2^2 = s^2 + 1.2 s + 4$.

Pasamos ahora a la determinación del numerador. Recordando (7.22) y (7.21),

$$\begin{aligned} 1 = h(\infty) = G(0) &= \frac{b_0}{\omega_n^2}, & \rightarrow & \quad b_0 = \omega_n^2 = 4, \\ 2 = h(0^+) = G(\infty) &= b_2. \end{aligned}$$

Finalmente, para determinar b_1 recordamos (7.24) y medimos $\dot{h}(0^+)$, cosa que ya hemos hecho en este caso, obteniendo

$$0 = \dot{h}(0^+) = b_1 - b_2 a_1 = b_1 - 2 \cdot 1.2 \quad \rightarrow \quad b_1 = 2.4.$$

En definitiva, la FT para la cual la respuesta al escalón unitario es la de la Figura 7.15 será

$$G(s) = \frac{2 s^2 + 2.4 s + 4}{s^2 + 1.2 s + 4}.$$

Un caso particular: el P_{T2}

Un caso particular pero de mucha utilidad es el caso en que $b_2 = b_1 = 0$, lo que implica que la FT de orden 2 es de la forma

$$G(s) = \frac{b_0}{s^2 + a_1 s + a_0}. \tag{7.25}$$

En este caso, según (7.21) y (7.24), resultará $h(0^+) = \dot{h}(0^+) = 0$. Es decir, la respuesta al escalón de esta FT no salta en el origen, indicado por $h(0^-) = h(0^+) = 0$, ni tampoco su derivada, $\dot{h}(0^-) = \dot{h}(0^+) = 0$.

En el caso de polos complejos conjugados, como esta respuesta posee derivada nula al comienzo, entonces el instante de inicio $t = 0$ corresponderá efectivamente a un máximo o mínimo de la oscilación y se puede considerar a este instante como el instante τ_i en (7.20), resultando τ_{i+1} el instante del siguiente extremo de la oscilación. Resultará entonces que el primer extremo de la oscilación, posterior al inicio, se dará a un semiperíodo de distancia del inicio, similarmente a lo que ocurre en la Figura 7.15 del Ejemplo 7.2, pero con la diferencia de que la respuesta no salta en el origen, es decir $h(0^-) = h(0^+) = h(0) = 0$. La Figura 7.16 ilustra la respuesta al escalón de un PT_2 .

7.3. Sistemas de Orden 2 Generales y de Orden Arbitrario

7.3.1. Retratos de Fase

Como hemos comentado en la Sección 7.2.1, en general las ecuaciones de estado de un sistema libre no resultarán desacopladas. En tal caso, y como ya se ha mencionado, son los autovalores de la matriz A los que principalmente determinan el tipo de retrato de fase. A continuación veremos la justificación de esto

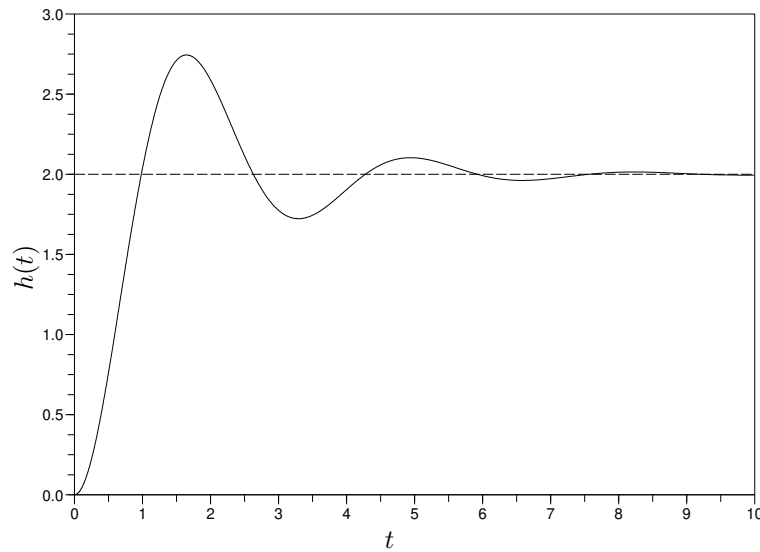


Figura 7.16: Respuesta al Escalón Unitario de un PT2. La característica es que $h(0) = \dot{h}(0) = 0$.

y el procedimiento general a seguir para entender la forma que poseen los retratos de fase de sistemas de orden 2 generales, así como de orden superior.

Dado un sistema LTI libre de orden n mediante sus ecuaciones de estado

$$\dot{\mathbf{x}}(t) = A \mathbf{x}(t), \tag{7.26}$$

proponemos el cambio de variables

$$\mathbf{x}(t) = V \mathbf{w}(t) \tag{7.27}$$

donde V es una matriz invertible de dimensión $n \times n$. La ecuación matricial (7.27) indica que cada componente x_i del vector \mathbf{x} es igual a una combinación lineal de los componentes w_1, \dots, w_n del vector \mathbf{w} , con coeficientes tomados de las entradas de la matriz V . Reemplazando (7.27) en (7.26), resulta

$$V \dot{\mathbf{w}}(t) = A V \mathbf{w}(t).$$

Multiplicando a la izquierda por V^{-1} (recordar que el producto matricial no es conmutativo y por lo tanto se debe preservar el orden de las multiplicaciones) se obtiene

$$\dot{\mathbf{w}}(t) = V^{-1} A V \mathbf{w}(t) = \Lambda \mathbf{w}(t), \tag{7.28}$$

donde se ha definido la nueva matriz

$$\Lambda := V^{-1} A V. \tag{7.29}$$

El cambio de variables realizado, de \mathbf{x} a \mathbf{w} puede interpretarse como un cambio de coordenadas en el espacio de estados. Cabe entonces la pregunta: ¿existirá un cambio de coordenadas tal que el sistema se vea desacoplado en las nuevas coordenadas? Y si existe, ¿cómo se obtiene tal cambio de coordenadas? En las nuevas coordenadas, las EE están dadas por (7.28), caracterizadas por la matriz de evolución Λ . Para que estas EE estén desacopladas, la matriz Λ debe ser diagonal.

Recordemos que si λ_i es un autovalor de A y V_i un autovector asociado a λ_i , entonces $V_i \neq 0$ y se satisface

$$A V_i = \lambda_i V_i, \tag{7.30}$$

es decir, la multiplicación de la matriz A por uno de sus autovectores da como resultado un múltiplo escalar del autovector, donde el factor escalar que multiplica es el autovalor. La ecuación (7.30) es equivalente a

$$\lambda_i V_i - A V_i = \lambda_i I V_i - A V_i = (\lambda_i I - A) V_i,$$

donde hemos tenido la astucia de introducir la matriz identidad para poder extraer por derecha el factor común V_i . Si la matriz A tiene n autovectores linealmente independientes y tomamos V como una matriz cuyas columnas son estos n autovectores de A ,

$$V = [V_1 \quad V_2 \quad \cdots \quad V_n], \tag{7.31}$$

entonces resultará V una matriz invertible y sucederá:

$$\begin{aligned} A V &= A [V_1 \quad V_2 \quad \cdots \quad V_n] = [AV_1 \quad AV_2 \quad \cdots \quad AV_n] = [\lambda_1 V_1 \quad \lambda_2 V_2 \quad \cdots \quad \lambda_n V_n] \\ &= [V_1 \lambda_1 \quad V_2 \lambda_2 \quad \cdots \quad V_n \lambda_n] = [V_1 \quad V_2 \quad \cdots \quad V_n] \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix} = V \Lambda, \end{aligned}$$

ecuación que equivale a (7.29) siendo además Λ una matriz diagonal.

En resumen, si la matriz A tiene n autovectores linealmente independientes, realizando el cambio de coordenadas (7.27) dado por la matriz V cuyas columnas son los n autovectores, entonces en las nuevas coordenadas \mathbf{w} las EE estarán desacopladas y el comportamiento de cada componente w_i estará caracterizado por el correspondiente autovalor λ_i . Un caso en el que la existencia de n autovectores linealmente independientes está asegurada es cuando la matriz A tiene sus n autovalores distintos, ya que autovectores correspondientes a distintos autovalores son necesariamente linealmente independientes.

Los autovalores de una matriz pueden ser números reales o complejos. Al ser A una matriz de números reales, si uno de sus autovalores, digamos λ_i , resulta no ser real entonces necesariamente otro de sus autovalores, digamos λ_{i+1} , deberá ser el complejo conjugado del primero: $\lambda_{i+1} = \bar{\lambda}_i$. Notar que autovalores complejos conjugados son distintos, y entonces poseen autovectores asociados que son linealmente independientes. Esto significa que en el desarrollo de arriba algunos de los λ_i podrían no ser reales, así como algunos de los autovectores correspondientes V_i . En tal caso, si bien podríamos obtener una forma desacoplada (matriz Λ diagonal), el sistema desacoplado resultante no sería de ninguna de las formas vistas anteriormente en la Sección 7.2.1. A continuación ejemplificaremos cómo proceder en tal caso para un sistema de orden 2.

Orden 2, autovalores no reales

Tomemos el caso de un modelo de orden $n = 2$, donde los autovalores de la matriz A (la que será de dimensión 2×2) son $\lambda_1 = \lambda_2 \notin \mathbb{R}$. Escribamos $\lambda_1 = \alpha + j\omega$ y $\lambda_2 = \alpha - j\omega$, con α y ω reales, este último no nulo. Tomemos autovectores correspondientes, de modo que

$$\begin{aligned} V_1 &= V_{\mathbb{R}} + jV_{\mathbb{I}} & V_{\mathbb{R}} &= \begin{bmatrix} v_{r1} \\ v_{r2} \end{bmatrix} = \text{Re}\{V_1\} & V_{\mathbb{I}} &= \begin{bmatrix} v_{i1} \\ v_{i2} \end{bmatrix} = \text{Im}\{V_1\} \\ V_2 &= V_{\mathbb{R}} - jV_{\mathbb{I}} \end{aligned}$$

La matriz V de (7.31) resulta con entradas complejas (no reales), de modo que aunque el cambio de coordenadas (7.27) produce una forma desacoplada, esta última no resultará real. Si se toma en cambio la siguiente matriz V para el cambio de coordenadas (7.27):

$$V = [V_{\mathbb{R}} \quad V_{\mathbb{I}}], \tag{7.32}$$

resultará

$$\Lambda = V^{-1}AV = \begin{bmatrix} \alpha & -\omega \\ \omega & \alpha \end{bmatrix} \tag{7.33}$$

y en las nuevas coordenadas el sistema se verá como alguno de los casos oscilatorios ya vistos.

En resumen, en el caso en que A tiene autovalores complejos conjugados (no reales), el cambio de variables o transformación de coordenadas que lleva a la forma más sencilla analizada está dado por la matriz V cuyas columnas son el vector parte real y vector parte imaginaria de un autovalor de A . En la Figura 7.12 se pueden observar los retratos de fase correspondientes al caso de orden 2 con autovalores no reales. Las rectas en línea de trazos son las direcciones de los vectores $V_{\mathbb{R}}$ y $V_{\mathbb{I}}$.

Orden 2, autovalores repetidos, matriz A no diagonal

Queda por analizar el caso en que al calcular los autovalores de A se obtienen autovalores repetidos, pero la matriz A no viene dada en forma diagonal. En estos casos, no se podrán encontrar n autovectores linealmente independientes que puedan usarse para cambiar de coordenadas y transformar las EE a una forma desacoplada. El caso más sencillo en el que esto sucede es el siguiente

$$\begin{aligned} \dot{w}_1 &= \lambda w_1 + w_2 & \dot{\mathbf{w}} &= J\mathbf{w} & J &= \begin{bmatrix} \lambda & 1 \\ 0 & \lambda \end{bmatrix} \\ \dot{w}_2 &= \lambda w_2 \end{aligned} \quad (7.34)$$

donde la matriz J tiene el autovalor repetido λ y usamos variables w_1 y w_2 (en lugar de x_1 y x_2) para enfatizar que en general las EE no van a estar en esta forma sencilla. Notar que si bien estas EE no están desacopladas, sí lo está la ecuación para \dot{w}_2 , ya que depende únicamente de w_2 . Podemos entonces resolver inmediatamente y obtener

$$w_2(t) = e^{\lambda t} w_2(0).$$

Al disponer ahora de la expresión para la evolución de w_2 , podemos reemplazarla en la primera ecuación:

$$\dot{w}_1(t) = \lambda w_1(t) + e^{\lambda t} w_2(0).$$

Esta es una ecuación lineal que también puede resolverse fácilmente, obteniendo como resultado

$$w_1(t) = e^{\lambda t} w_1(0) + t e^{\lambda t} w_2(0).$$

Combinando las soluciones obtenidas en forma vectorial, podemos escribir

$$\mathbf{w}(t) = \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix} = \begin{bmatrix} e^{\lambda t} & t e^{\lambda t} \\ 0 & e^{\lambda t} \end{bmatrix} \begin{bmatrix} w_1(0) \\ w_2(0) \end{bmatrix} = \Phi(t) \mathbf{w}(0). \quad (7.35)$$

El retrato de fase correspondiente recibe el nombre de caso Jordan, debido a que la matriz J en (7.34) es un *bloque de Jordan de orden 2* correspondiente al autovalor λ . Dependiendo del signo del autovalor λ tendremos los casos estable ($\lambda < 0$), degenerado ($\lambda = 0$) e inestable ($\lambda > 0$), los que se ilustran en la Figura 7.17. Notar que el eje w_1 es una dirección invariante bajo la dinámica del sistema.

Supongamos ahora el sistema

$$\dot{\mathbf{x}} = A\mathbf{x}$$

donde la matriz A posee un autovalor doble λ , pero la cual no es diagonal ni de la forma (7.34). En este caso, no existen 2 autovectores linealmente independientes y no podremos hallar tampoco un cambio de coordenadas $\mathbf{x} = V\mathbf{w}$ de modo que el sistema se vea desacoplado en las nuevas variables \mathbf{w} . Nos preguntamos entonces si podemos hallar un cambio de coordenadas que lleve al sistema a la forma (7.34). La respuesta es que sí y que tal cambio de coordenadas está dado por

$$V = [V_1 \quad V_{p1}], \quad (7.36)$$

con V_1 autovector de A , de modo que satisface la ecuación

$$(\lambda I - A)V_1 = 0,$$

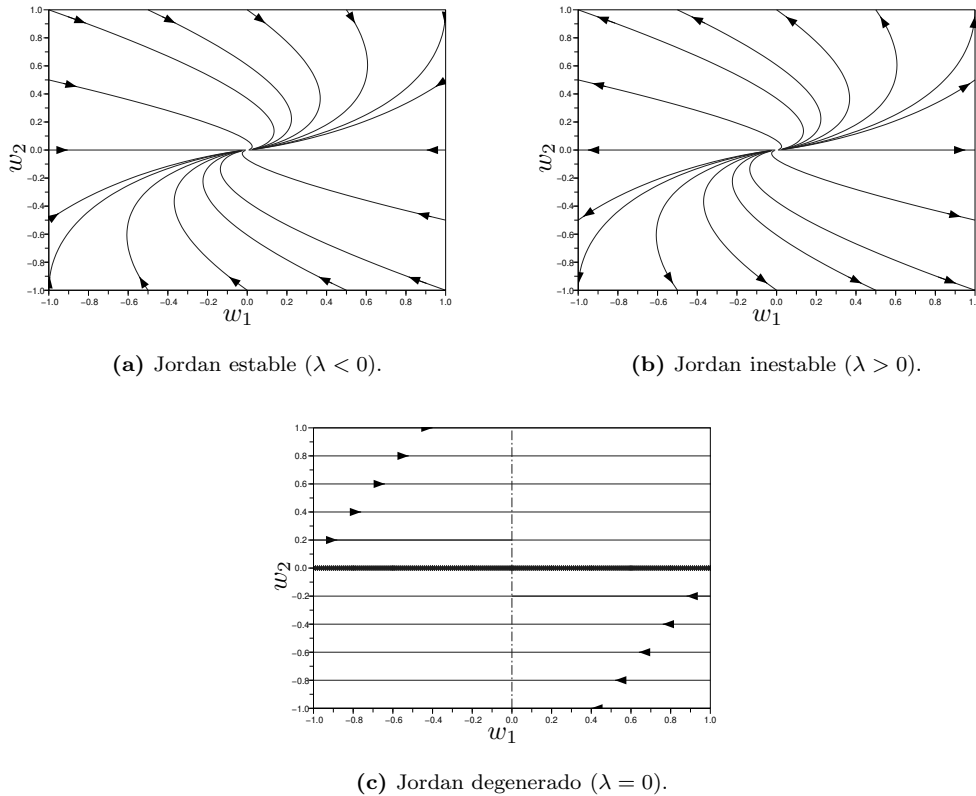


Figura 7.17: Retratos de Fases de un sistema de la forma (7.34) (casos Jordan).

y donde V_{p1} está relacionado tanto con λ como con V_1 , siendo solución de la ecuación

$$(\lambda I - A)V_{p1} = -V_1.$$

Realizando el cambio de coordenadas $\mathbf{x} = V\mathbf{w}$ resultará en las nuevas coordenadas

$$\dot{\mathbf{w}} = V^{-1} A V \mathbf{w} = J \mathbf{w}, \tag{7.37}$$

con J como en (7.34). El vector V_{p1} recibe el nombre de vector principal asociado a V_1 . Los retratos de fases generales para este caso se ilustran en la Figura 7.18. La dirección invariante observada es la dirección del autovector V_1 .

Sistemas de orden superior

En el caso de sistemas LTI libres de orden mayor a 2, modelados por medio de las EE

$$\dot{\mathbf{x}} = A \mathbf{x},$$

la idea principal para analizar la dinámica es la misma que para los casos de orden 2 generales: hallar una transformación de coordenadas de modo que el sistema se vea más simple en las nuevas coordenadas. Esta transformación estará dada por una matriz V cuyas columnas sean autovectores linealmente independientes de A , sus vectores parte real e imaginaria en el caso de autovalores complejos, o los vectores principales asociados en el caso que no existan suficientes autovectores linealmente independientes. No ahondaremos aquí en esto, salvo mencionando que la matriz $\Lambda = V^{-1}AV$ resultante recibe el nombre de *forma canónica real de Jordan* de A .

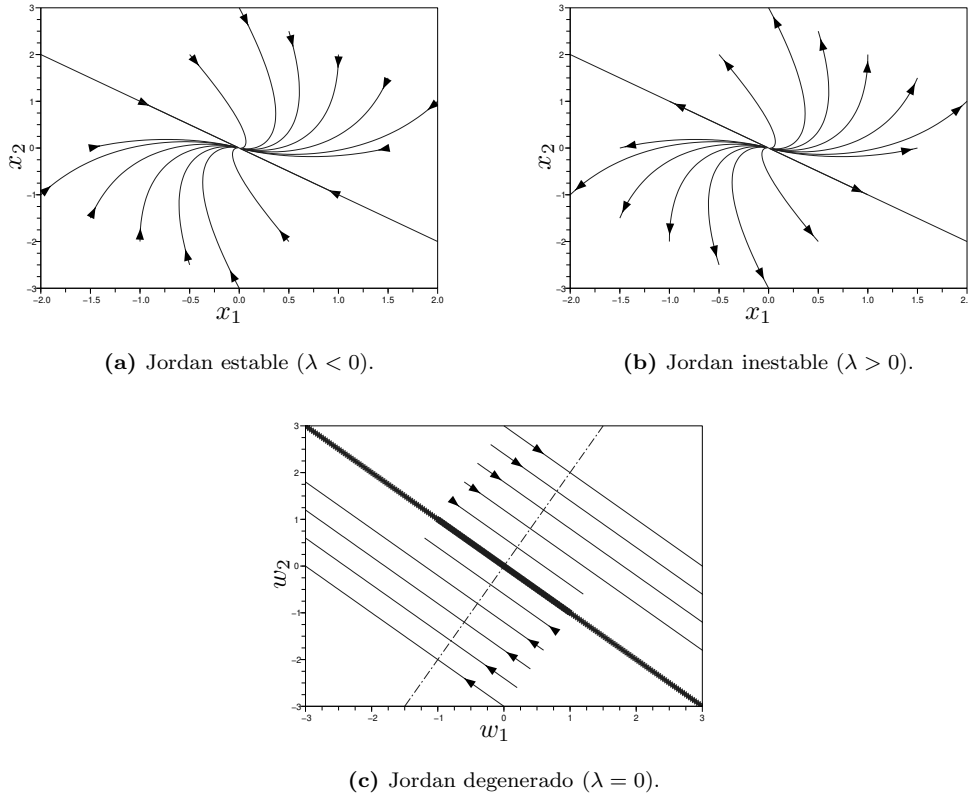


Figura 7.18: Retratos de Fases del tipo Jordan.

7.3.2. Solución General de las Ecuaciones de Estado

Consideremos un sistema de EE dado en su forma matricial:

$$\dot{\mathbf{x}}(t) = A \mathbf{x}(t) + B \mathbf{u}(t)$$

y supongamos que la condición inicial está dada por

$$\mathbf{x}(0) = \mathbf{x}_0$$

Para calcular la solución $\mathbf{x}(t)$ del sistema, podemos recurrir a la Transformada de Laplace:

$$s \mathbf{X}(s) - \mathbf{x}_0 = A \mathbf{X}(s) + B \mathbf{U}(s)$$

de donde,

$$(s I - A)\mathbf{X}(s) = \mathbf{x}_0 + B \mathbf{U}(s) \Rightarrow \mathbf{X}(s) = (s I - A)^{-1} \mathbf{x}_0 + (s I - A)^{-1} B \mathbf{U}(s)$$

Notar que lo desarrollado en la Sección 7.1.2 corresponde a este mismo desarrollo para el caso en que $n = 1$. Antitransformando resulta

$$\mathbf{x}(t) = \Phi(t) \mathbf{x}_0 + \int_0^t \Phi(\tau) B \mathbf{u}(t - \tau) d\tau \tag{7.38}$$

donde

$$\Phi(t) \triangleq \mathcal{L}^{-1} \{ (s I - A)^{-1} \} \tag{7.39}$$

y el término con la integral es el producto de convolución entre $\Phi(t)$ y $B\mathbf{U}(t)$. Como puede verse, la solución de las EE dada por la Ec.(7.38) tiene dos términos. Uno debido a las condiciones iniciales \mathbf{x}_0

y otro debido a las entradas $\mathbf{u}(t)$. El primer término se denomina *respuesta libre* y el segundo *respuesta forzada*.

La matriz $\Phi(t)$ que caracteriza la solución se denomina *Matriz de Transición* del sistema. Teniendo en cuenta la analogía de la Ec.(7.39) con la expresión de la transformada de una función exponencial, se define la *exponencial matricial*:

$$e^{A t} \triangleq \Phi(t) = \mathcal{L}^{-1} \{ (s I - A)^{-1} \} \quad (7.40)$$

que en el caso que A sea escalar coincide con la exponencial normal. La matriz de transición $\Phi(t) = e^{A t}$ tiene varias propiedades que coinciden con las de la exponencial escalar:

$$\begin{aligned} \Phi(0) &= e^{A \cdot 0} = I \\ \Phi(t_1 + t_2) &= e^{A(t_1+t_2)} = e^{A t_1} e^{A t_2} = \Phi(t_1) \Phi(t_2) \\ \Phi(-t) &= e^{-A t} = (e^{A t})^{-1} = \Phi(t)^{-1} \end{aligned}$$

Los elementos de la matriz de transición $\Phi(t)$ tienen una forma bastante particular, como veremos a continuación. Recordando que la inversa de una matriz puede escribirse como la matriz de cofactores dividida por el determinante, resulta

$$e^{A t} = \mathcal{L}^{-1} \{ (s I - A)^{-1} \} = \mathcal{L}^{-1} \left\{ \frac{\text{adj}^T(s I - A)}{\det(s I - A)} \right\}$$

Esto muestra que cada elemento de la matriz $\Phi(t)$ tiene por transformada un cociente de polinomios. Esto es,

$$\Phi_{i,j}(t) = \mathcal{L}^{-1} \left\{ \frac{N_{i,j}(s)}{\det(s I - A)} \right\}$$

con $N_{i,j}$ el elemento en fila i y columna j de la matriz de cofactores $\text{adj}^T(s I - A)$. El polinomio $\det(s I - A)$ tiene raíces en los valores de s que satisfacen $\det(s I - A) = 0$, es decir cuando s coincide con un *autovalor* de la matriz A . Por lo tanto, llamando $\lambda_1, \lambda_2, \dots, \lambda_n$ a los n autovalores de la matriz A , expandiendo cada cociente de polinomios $N_{i,j}(s)/\det(s I - A)$ en fracciones simples y antitransformando, resulta que (en muchos casos, no en todos¹) cada componente de la matriz de transición podrá escribirse como:

$$\Phi_{i,j}(t) = c_1 e^{\lambda_1 t} + \dots + c_n e^{\lambda_n t}$$

para ciertas constantes c_1, c_2 , etc. Hay que tener en cuenta que si un autovalor $\lambda_i = \alpha + j\omega$ no es real, entonces su complejo conjugado $\lambda_{i+1} = \bar{\lambda}_i = \alpha - j\omega$ también es autovalor, y los coeficientes correspondientes satisfacen $c_i = d_R + jd_I$, $c_{i+1} = \bar{c}_i = d_R - jd_I$. Por lo tanto, dentro de la suma anterior tendremos

$$\begin{aligned} c_i e^{\lambda_i t} + c_{i+1} e^{\lambda_{i+1} t} &= c_i e^{\lambda_i t} + \bar{c}_i e^{\bar{\lambda}_i t} = 2\text{Re}\{c_i e^{\lambda_i t}\} \\ &= 2\text{Re}\{c_i e^{\alpha t} e^{j\omega t}\} = 2e^{\alpha t} [d_R \cos(\omega t) + d_I \sin(\omega t)] \end{aligned}$$

dando lugar a comportamiento oscilatorio. Si consideramos entonces la respuesta libre de un sistema lineal y estacionario, haciendo $\mathbf{u} = 0$ en la Ec.(7.38), tendremos

$$\mathbf{x}(t) = \Phi(t) \mathbf{x}_0 = \begin{bmatrix} \alpha_{1,1} e^{\lambda_1 t} + \dots + \alpha_{1,n} e^{\lambda_n t} \\ \vdots \\ \alpha_{n,1} e^{\lambda_1 t} + \dots + \alpha_{n,n} e^{\lambda_n t} \end{bmatrix}$$

para ciertas constantes $\alpha_{i,j}$ que dependerán de la condición inicial \mathbf{x}_0 y de la matriz A , de modo que autovalores no reales aparecen en pares complejos conjugados y las constantes $\alpha_{i,j}$ serán tales que cada componente de $\mathbf{x}(t)$ resulte real.

¹Estamos asumiendo que no hay autovalores repetidos. Si los hubiera, la suma poseerá términos de la forma $t^k e^{\lambda_j t}$.

7.3.3. Puntos de Equilibrio y Estabilidad Interna

Los modelos que involucran al estado, como las EE, se utilizan a menudo para representar sistemas no lineales. Algunos métodos para analizar EE no lineales se verán en el Capítulo 8, mientras que en este capítulo nos concentramos exclusivamente en los sistemas lineales.

Consideremos unas EE lineales e invariantes en el tiempo de orden $n \in \mathbb{N}$ arbitrario con $m \in \mathbb{N}$ entradas

$$\dot{\mathbf{x}}(t) = A \mathbf{x}(t) + B \mathbf{u}(t) \quad (7.41)$$

donde la entrada es constante e igual al valor $\bar{\mathbf{u}} \in \mathbb{R}^m$, es decir, $\mathbf{u}(t) = \bar{\mathbf{u}}$ para todo t . Reemplazando en las EE, obtenemos

$$\dot{\mathbf{x}}(t) = A \mathbf{x}(t) + B \bar{\mathbf{u}}. \quad (7.42)$$

Un **Punto de Equilibrio** del sistema (7.42) es un valor constante $\bar{\mathbf{x}}$ del vector de estado \mathbf{x} tal que si la condición inicial es $\mathbf{x}(0) = \bar{\mathbf{x}}$, la trayectoria resultante, es decir la solución de (7.42) que parte de esa condición inicial, es $\mathbf{x}(t) = \bar{\mathbf{x}}$ para todo t . En otras palabras, un punto de equilibrio es un punto (en el espacio de estados) tal que si el sistema comienza en ese estado, se queda en ese estado. Para que un estado constante sea solución de (7.42), su derivada debe ser nula. Por lo tanto, los puntos de equilibrio del sistema son las soluciones de

$$\mathbf{0} = A \bar{\mathbf{x}} + B \bar{\mathbf{u}}. \quad (7.43)$$

Si la matriz A es invertible (o sea, no tiene autovalores nulos), existirá un único punto de equilibrio, el que se puede calcular multiplicando a la izquierda por la inversa de la matriz A :

$$\mathbf{0} = A \bar{\mathbf{x}} + B \bar{\mathbf{u}} \quad \Rightarrow \quad \bar{\mathbf{x}} = -A^{-1} B \bar{\mathbf{u}}.$$

En caso que la matriz A no sea invertible, hay dos posibilidades: infinitos puntos de equilibrio o ningún punto de equilibrio.

Ahora bien, si el estado inicial no coincide con el punto de equilibrio, ¿cómo podemos determinar si la solución se acerca al equilibrio o no? La respuesta a esta pregunta está relacionada con el concepto de *Estabilidad Interna*.

Para un sistema LTI diremos que un punto de equilibrio es **estable** cuando toda solución que parte suficientemente cerca del mismo no se aleja indefinidamente². Diremos además que un punto de equilibrio es inestable cuando no es estable. Por último, diremos que un punto de equilibrio es asintóticamente estable cuando además de ser estable, toda solución tiende asintóticamente al mismo.

Modelo Incremental

Así como hemos visto que se puede cambiar de coordenadas para ver las EE en una forma más sencilla que las originales, también se puede realizar una traslación de modo que un punto de equilibrio se vea como el origen en las nuevas variables trasladadas. Por ejemplo, si tenemos un punto de equilibrio $\bar{\mathbf{x}}$ en correspondencia con la entrada constante $\bar{\mathbf{u}}$, podemos definir el *estado incremental* $\Delta \mathbf{x}(t) = \mathbf{x}(t) - \bar{\mathbf{x}}$, de modo que

$$\begin{aligned} \Delta \dot{\mathbf{x}}(t) &= \dot{\mathbf{x}}(t) - \underbrace{\dot{\bar{\mathbf{x}}}}_{=\mathbf{0}} = A \mathbf{x}(t) + B \bar{\mathbf{u}} = A \Delta \mathbf{x}(t) + \underbrace{A \bar{\mathbf{x}} + B \bar{\mathbf{u}}}_{=\mathbf{0}} \\ &= A \Delta \mathbf{x}(t). \end{aligned}$$

En las *variables incrementales* $\Delta \mathbf{x}$, el sistema se ve como un sistema libre. Por lo tanto, podemos analizar directamente la estabilidad del origen del sistema sin entrada caracterizado por la matriz A ,

$$\dot{\mathbf{x}}(t) = A \mathbf{x}(t),$$

²En el caso no lineal es necesaria una definición más precisa, que veremos en el Capítulo 8.

y concluir sobre cualquier punto de equilibrio que resulte de aplicar entradas constantes.

Como vimos anteriormente, la solución de la ecuación libre tiene la forma

$$\mathbf{x}(t) = \Phi(t) \mathbf{x}_0 = \begin{bmatrix} \alpha_{1,1} e^{\lambda_1 t} + \cdots + \alpha_{1,n} e^{\lambda_n t} \\ \vdots \\ \alpha_{n,1} e^{\lambda_1 t} + \cdots + \alpha_{n,n} e^{\lambda_n t} \end{bmatrix}$$

siendo λ_i los autovalores de la matriz A . De aquí, es evidente que:

El punto de equilibrio de un sistema LTI es asintóticamente estable si y sólo si todos los autovalores de la matriz A tienen parte real negativa.

En este caso hablamos de un único punto de equilibrio ya que la condición establece que no hay autovalores nulos y por lo tanto A es invertible. Además, es claro que,

Si uno o más autovalores de la matriz A de un sistema LTI tienen parte real positiva, sus puntos de equilibrio serán inestables.

Por otro lado, cuando los autovalores tengan parte real nula, no se aproximarán al equilibrio y tampoco se alejarán mucho, salvo que se trate de autovalores múltiples, ya que en ese caso aparecerán términos no amortiguados multiplicados por t , t^2 , etc. Luego,

Si un sistema LTI tiene todos sus autovalores con parte real menor o igual que cero y no tiene autovalores con parte real nula repetidos, entonces sus puntos de equilibrio serán estables.

Esto último brinda una condición suficiente (pero no necesaria) para asegurar estabilidad interna. Existen además sistemas donde la matriz A tiene autovalores con parte real nula repetidos cuyos puntos de equilibrio son estables, como el caso de la matriz nula $A = \mathbf{0}$.

Como vemos, la posición de los autovalores de la matriz A en un modelo de EE juega un rol análogo al de la posición de los polos de la función transferencia en la determinación de la estabilidad.

Capítulo 8

Herramientas de Análisis de Sistemas No Lineales

Gran parte de los modelos que vimos y construimos en los distintos capítulos tenían componentes con leyes no lineales. Sin embargo, las distintas herramientas de análisis que nos permiten establecer propiedades de estabilidad o de la respuesta temporal que estudiamos hasta acá están limitadas a sistemas lineales ya que se basan esencialmente en mirar los polos de la función transferencia $G(s)$ o los autovalores de la matriz de evolución A .

En este capítulo presentaremos algunas herramientas básicas para analizar propiedades en sistemas no lineales. El capítulo está basado en parte en los tres primeros capítulos del libro *Nonlinear Control* de Hassan Khalil [Kha15] y en una adaptación del mismo en castellano que constituyen las Notas de Clase de la materia de doctorado Sistemas No Lineales de nuestra facultad [SB00]. Respecto al Teorema de Hartman-Grobman, una referencia de profundización es el libro *Differential Equations and Dynamical Systems* de Lawrence Perko [Per13].

8.1. Trayectorias, Puntos de Equilibrio y Estabilidad

Las trayectorias en el espacio de estados de los sistemas lineales dependían principalmente de la posición de los autovalores de la matriz de evolución A . De acuerdo a los mismos, podíamos determinar tanto la estabilidad de los puntos de equilibrio como la forma cualitativa del retrato de fases. En los sistemas no lineales veremos que las trayectorias pueden ser significativamente más complejas y que puede haber múltiples puntos de equilibrio con características de estabilidad muy distintas.

8.1.1. Trayectorias en Sistemas No Lineales

Comenzaremos presentando algunos ejemplos de sistemas no lineales relativamente simples y observando las trayectorias de los mismos.

Péndulo con Fricción

Consideraremos el modelo del péndulo simple que vimos en capítulos anteriores con el agregado de fricción lineal y representando el movimiento en variables angulares. Llamando $x_1(t) = \phi(t)$ al ángulo medido respecto de la vertical y $x_2(t) = \omega(t)$ a la velocidad angular correspondiente, las ecuaciones resultan las siguientes:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= -\frac{g}{L} \sin(x_1(t)) - \frac{b}{J} x_2(t) \end{aligned} \tag{8.1}$$

donde g es la aceleración de la gravedad, L es la longitud del cable, b es el coeficiente de fricción y $J = m L^2$ es el momento de inercia. Suponiendo que $b = m = L = 1$ y que $m = 9.81$, la Figura 8.1 muestra trayectorias a partir de distintas condiciones iniciales.

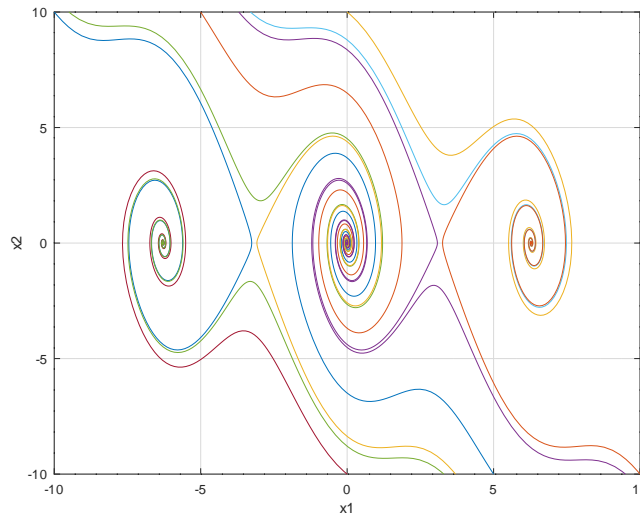


Figura 8.1: Trayectorias en el Espacio de Estados del modelo del péndulo de la Ecuación (8.1)

En el retrato de fases podemos observar lo siguiente:

- Hay varios puntos de equilibrio que luego veremos que se corresponden a los ángulos $x_1 = 0, \pm\pi, \pm 2\pi$, etc. y a la velocidad angular $x_2 = 0$.
- Cerca de ciertos puntos de equilibrio las trayectorias se asemejan a las de un foco estable.
- Cerca de otros puntos de equilibrio las trayectorias se asemejan a las de una ensilladura.
- Lejos de los puntos de equilibrio (para velocidades angulares grandes) las trayectorias no se asemejan a nada conocido.

Para comprender mejor estas trayectorias, la Figura 8.2 muestra la evolución en el tiempo de ambos estados a partir de cierta condición inicial. En dicha evolución se observa que el ángulo comienza cercano a π (o sea, en posición casi vertical de péndulo invertido) y con una gran velocidad angular negativa. Luego de pasar por la posición $\phi = 0$, debido a la gran velocidad inicial, el péndulo sigue girando y completa una vuelta más tras lo cual se termina deteniendo en torno a la posición $\phi = -2\pi$.

Más allá de esta interpretación física de las trayectorias, está claro que el retrato de fases muestra cosas que no habíamos visto en los sistemas lineales. La presencia de múltiples puntos de equilibrio es algo novedoso (teníamos siempre un punto, o cuando la matriz A era singular había infinitos cubriendo una recta o ninguno). También es novedoso que los distintos puntos de equilibrio muestren distintas características de estabilidad.

Circuito con Diodo Túnel

La Figura 8.3 muestra un circuito oscilador con un diodo túnel y la característica Volt-Ampere de dicho componente. En dicha característica se observa una zona donde la resistencia tiene un comportamiento negativo, que es lo que se aprovecha para generar oscilaciones sostenidas. Las ecuaciones del circuito son las siguientes:

$$\begin{aligned} \dot{x}_1(t) &= \frac{x_2(t)}{C} - \frac{i_D(t)}{C} - \frac{x_1(t)}{RC} \\ \dot{x}_2(t) &= \frac{E}{L} - \frac{x_1(t)}{L} \end{aligned} \quad (8.2)$$

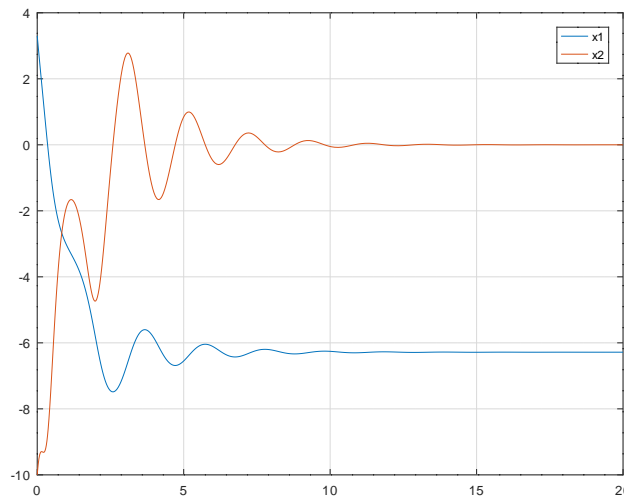


Figura 8.2: Trayectoria en el Tiempo del modelo del péndulo a partir de una condición inicial.

donde $x_1(t)$ es la tensión del capacitor y $x_2(t)$ es la corriente en la inductancia: La corriente del diodo $i_D(t)$ está dada por la relación Volt-Ampere del mismo:

$$i_D(t) = 83.72 x_1(t)^5 - 226.31 x_1(t)^4 + 229.62 x_1(t)^3 - 103.79 x_1(t)^2 + 17.76 x_1(t) \quad (8.3)$$

lo que introduce un término no lineal al circuito.

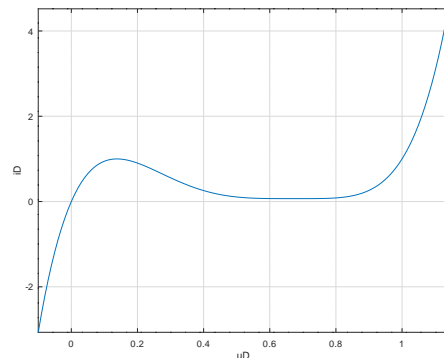
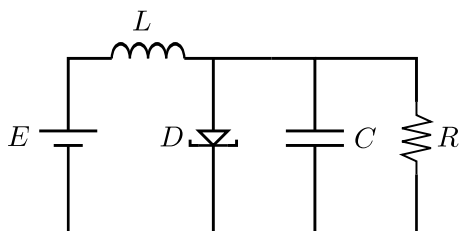


Figura 8.3: Circuito Oscilador con Diodo Túnel y Característica Volt-Ampere del Diodo.

La Figura 8.4 muestra dos trayectorias en el espacio de estados, obtenidas utilizando parámetros $E = 0.5$, $R = 10$, $L = 1$ y $C = 25$ (si tomamos como unidad de tiempo el microsegundo, las unidades de la inductancia y capacitor serían micro-Henry micro-Coulomb). Una trayectoria (la que parte del punto x_b) se asemeja a un foco inestable que parte cerca de un equilibrio, pero luego parece converger a una curva cerrada con una oscilación sostenida. La otra, parece la de un foco estable, pero en lugar de converger al equilibrio, lo hace hacia la misma curva cerrada.

Esta curva cerrada hacia la que convergen las distintas trayectorias se denomina *ciclo límite* y en este caso puede verse que desde cualquier condición inicial las trayectorias terminan en el mismo. Esto implica que, tras un transitorio, las variables realizan una oscilación sostenida cuya amplitud y frecuencia son independientes de las condiciones iniciales. La Figura 8.5 muestra la evolución temporal de la trayectoria que se inicia dentro del ciclo límite, y se ve como la misma converge a una señal periódica.

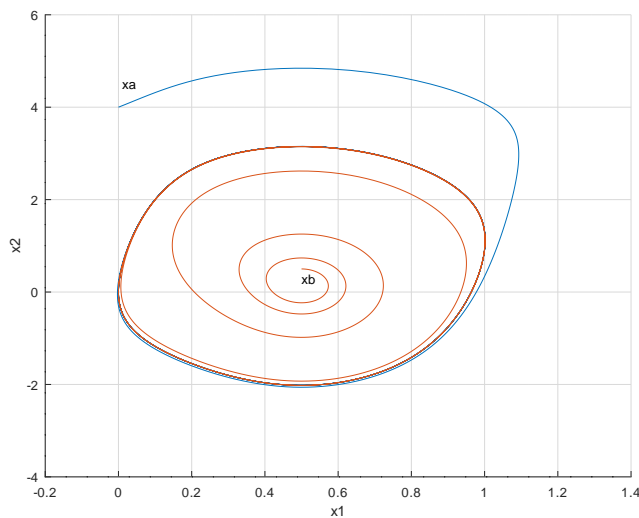


Figura 8.4: Trayectorias en el Espacio de Estados del modelo del Circuito de la Ecuación (8.2)

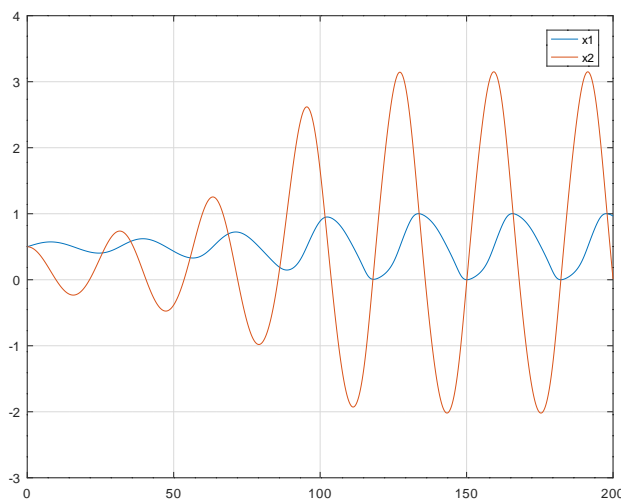


Figura 8.5: Evolución temporal de las variables del oscilador de la Ecuación (8.2)

Los ciclos límites son trayectorias que se producen exclusivamente en sistemas no lineales.

Oscilador de Tercer Orden

Consideremos ahora el siguiente sistema de tercer orden:

$$\begin{aligned}
 \dot{x}_1(t) &= x_2(t) \\
 \dot{x}_2(t) &= x_1(t) - x_1(t)^3 + b x_2(t) - k x_3(t) \\
 \dot{x}_3(t) &= w_f (x_2(t) - x_3(t))
 \end{aligned}
 \tag{8.4}$$

donde b , k y w_f son parámetros. Adoptando $b = 0.35$, $k = 1.6$ y $w_f = 0.5$, la trayectoria desde una condición inicial $x(0) = [0.1, 0, 0]^T$ resulta la que se muestra en las Figuras 8.6–8.7.

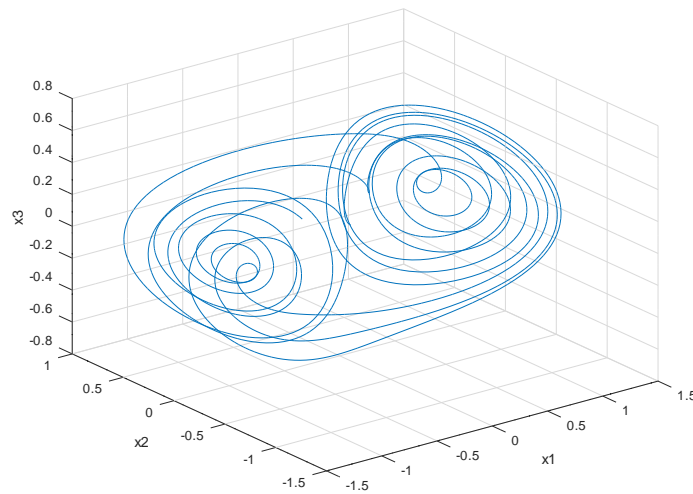


Figura 8.6: Trayectorias en el Espacio de Estados del sistema de la Ecuación (8.4).

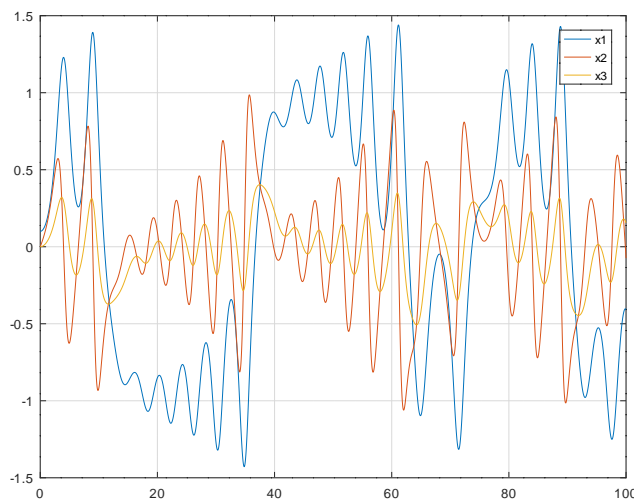


Figura 8.7: Evolución temporal de las variables del sistema de la Ecuación (8.4).

En estas trayectorias se puede observar un comportamiento oscilatorio no periódico, donde ni siquiera hay un patrón claro. En realidad, habíamos visto ya un sistema con un comportamiento similar cuando simulamos el doble péndulo de la Figura 4.30. Una característica de estas evoluciones es que dos trayectorias que se inician en puntos arbitrariamente cercanos, tras un tiempo divergen y siguen caminos muy diferentes. Por este motivo, cuando hay incertidumbre de parámetros o ruido, el comportamiento se torna muy impredecible.

Este fenómeno se denomina *caos* y es también exclusivo de los sistemas no lineales. En los sistemas continuos autónomos sólo puede aparecer cuando el orden es 3 o más. En un sistema de segundo orden, al no poder cortarse entre sí las trayectorias, una solución oscilatoria que a largo plazo no gana ni pierde amplitud termina conformando una curva cerrada y por lo tanto una evolución periódica.

Si bien en este curso no estudiaremos herramientas para determinar la presencia de caos en un sistema, es importante poder reconocer el fenómeno y tener presente que dicho comportamiento es bastante

frecuente en sistemas no lineales.

8.1.2. Determinación de los Puntos de Equilibrio

Como veremos más adelante, las principales propiedades que podremos analizar en los sistemas no lineales estarán relacionadas con los *puntos de equilibrio* de los mismos. Recordemos que un punto de equilibrio de un sistema es un valor del estado $x(t) = \bar{x}$ en el cual se cumple que la derivada respecto al tiempo es nula y por lo tanto la solución permanece en dicho valor constante.

Vamos a tratar en este curso con sistemas no lineales invariantes respecto al tiempo, esto es, que puedan expresarse como

$$\dot{x}(t) = f(x(t), u(t)) \quad (8.5)$$

donde $x(t) \in \mathbb{R}^n$ y $u(t) \in \mathbb{R}^m$. Los puntos de equilibrio de este sistema estarán por lo tanto asociados a la presencia de una entrada constante $u(t) = \bar{u}$ y serán las soluciones \bar{x} de la ecuación

$$\dot{x}(t) = f(\bar{x}, \bar{u}) = 0. \quad (8.6)$$

Si tomamos por ejemplo el sistema del péndulo de la Ec.(8.1) (donde no hay entrada), resulta

$$\begin{aligned} \dot{x}_1(t) &= \bar{x}_2 = 0 \\ \dot{x}_2(t) &= -\frac{g}{L} \sin(\bar{x}_1) - \frac{b}{J} \bar{x}_2 = 0 \end{aligned}$$

y de la primera ecuación resulta $\bar{x}_2 = 0$, mientras que de la segunda resulta $\sin(\bar{x}_1) = 0$ de donde $\bar{x}_1 = k \pi$ para todo $k \in \mathbb{Z}$. Esto es, hay infinitos puntos de equilibrio de la forma $[k \pi, 0]^T$ como ya habíamos observado en la Figura 8.1. Desde el punto de vista del sistema físico, estos puntos de equilibrio se corresponden a dos posiciones del péndulo. Cuando k es par, el mismo se encuentra en posición vertical hacia abajo (posición estable) mientras que cuando k es impar se encuentra en posición vertical hacia arriba (péndulo invertido).

Tomando ahora el modelo del circuito con diodo túnel de la Ec.(8.2), podemos calcular el o los puntos de equilibrio haciendo:

$$\begin{aligned} \dot{x}_1(t) &= \frac{\bar{x}_2}{C} - \frac{\bar{i}_D}{C} - \frac{\bar{x}_1}{R C} = 0 \\ \dot{x}_2(t) &= \frac{E}{L} - \frac{\bar{x}_1}{L} = 0 \end{aligned}$$

y de la segunda ecuación resulta $\bar{x}_1 = E$ mientras que de la primera ecuación se obtiene

$$\bar{x}_2 = \bar{i}_D + \frac{\bar{x}_1}{R} = 83.72 \bar{x}_1^5 - 226.31 \bar{x}_1^4 + 229.62 \bar{x}_1^3 - 103.79 \bar{x}_1^2 + 17.76 \bar{x}_1 + \frac{\bar{x}_1}{R}$$

que para los parámetros $E = 0.5$ y $R = 10$ resulta $\bar{x} = [0.5, 0.15688]^T$ (hay un único punto de equilibrio).

Por último, en el modelo del oscilador de tercer orden de la Ec.(8.4), podemos calcular los puntos de equilibrio como sigue:

$$\begin{aligned} \dot{x}_1(t) &= \bar{x}_2 = 0 \\ \dot{x}_2(t) &= \bar{x}_1 - \bar{x}_1^3 + b \bar{x}_2 - k \bar{x}_3 = 0 \\ \dot{x}_3(t) &= w_f (\bar{x}_2 - \bar{x}_3) = 0 \end{aligned}$$

y de la primera ecuación se obtiene $\bar{x}_2 = 0$, mientras que de la tercera resulta $\bar{x}_3 = \bar{x}_2 = 0$ y reemplazando en la segunda queda $\bar{x}_1 - \bar{x}_1^3 = 0$ lo que tiene tres soluciones: $\bar{x}_1 = 0$ y $\bar{x}_1 = \pm 1$. Por lo tanto, hay tres puntos de equilibrio: $\bar{x}_a = [0, 0, 0]^T$, $\bar{x}_b = [1, 0, 0]^T$ y $\bar{x}_c = [-1, 0, 0]^T$.

8.1.3. Modelo en Torno al Equilibrio

En muchas ocasiones es muy útil considerar que el punto de equilibrio está en el origen ya que esto facilita el análisis de estabilidad y de la respuesta a condiciones iniciales del sistema. Para esto, cuando un punto de equilibrio no se corresponda con el origen lo que haremos será desplazar el eje de coordenadas mediante un simple cambio de variables de manera que en las nuevas variables el origen sea efectivamente el punto de equilibrio.

Una característica de los sistemas lineales y estacionarios era que para una entrada constante se desplazaba el punto de equilibrio pero no cambiaba la dinámica en torno al mismo. Por lo tanto, al hacer un cambio de variables tal que el punto de equilibrio en las nuevas coordenadas sea el origen la matriz A quedaba idéntica. Como veremos, en los sistemas no lineales esto no será así y dependiendo de la entrada podrá en general cambiar la dinámica y por lo tanto la forma de la función f que determina el lado derecho de la ecuación.

Veamos entonces como realizar el cambio de variables. Dado el sistema de la Ec.(8.5) y un punto de equilibrio \bar{x} para la entrada constante $u(t) = \bar{u}$, definimos las nuevas variables $\Delta x(t) = x(t) - \bar{x}$, y $\Delta u(t) = u(t) - \bar{u}$ luego:

$$\Delta \dot{x}(t) = \dot{x}(t) = f(x(t), u(t)) = f(\Delta x(t) + \bar{x}, \Delta u(t) + \bar{u})$$

Definiendo entonces $\tilde{f}(\Delta x(t), \Delta u(t)) \triangleq f(\Delta x(t) + \bar{x}, \Delta u(t) + \bar{u})$, resulta

$$\Delta \dot{x}(t) = \tilde{f}(\Delta x(t), \Delta u(t)) \quad (8.7)$$

lo que constituye el modelo en torno al equilibrio.

Considerando por ejemplo el modelo del péndulo dado por la Ec.(8.1) y el punto de equilibrio $\bar{x} = [\pi, 0]^T$, el modelo en torno a este equilibrio resulta

$$\begin{aligned} \Delta \dot{x}_1(t) &= \Delta x_2(t) \\ \Delta \dot{x}_2(t) &= -\frac{g}{L} \sin(\Delta x_1(t) + \pi) - \frac{b}{J} \Delta x_2(t) = \frac{g}{L} \sin(\Delta x_1(t)) - \frac{b}{J} \Delta x_2(t) \end{aligned} \quad (8.8)$$

8.2. Linealización

Los fenómenos físicos que ocurren en los sistemas son generalmente de naturaleza no lineal. Sin embargo, en muchas ocasiones se tiene en cuenta sólo la parte lineal de las relaciones entre las magnitudes ya que esto facilita el análisis de propiedades temporales y frecuenciales, el diseño de sistemas de control, etc. Además, mientras las trayectorias no se alejen mucho de algún punto de equilibrio, las hipótesis de linealidad suelen implicar aproximaciones más que razonables.

Veremos entonces cómo obtener aproximaciones lineales a partir de modelos no lineales. Más adelante estudiaremos bajo qué condiciones ciertas propiedades del sistema linealizado podrán extenderse al sistema no lineal.

8.2.1. Linealización en Torno al Equilibrio

Tomemos nuevamente el sistema de la Ec.(8.5) y consideremos que el mismo tiene cierto punto de equilibrio \bar{x} para cierta entrada constante \bar{u} y veamos como es la expansión en serie de Taylor de la función $f(x(t), u(t))$ en torno a \bar{x} y \bar{u} :

$$\dot{x}(t) = f(x(t), u(t)) = f(\bar{x}, \bar{u}) + \left. \frac{\partial f}{\partial x} \right|_{\bar{x}, \bar{u}} (x(t) - \bar{x}) + \left. \frac{\partial f}{\partial u} \right|_{\bar{x}, \bar{u}} (u(t) - \bar{u}) + O[(x(t) - \bar{x})^2, (u(t) - \bar{u})^2]$$

donde $O[\cdot]$ representa el resto de la serie que tiene sólo términos de orden superior. Recordando que $f(\bar{x}, \bar{u}) = 0$ al tratarse de un punto de equilibrio, y definiendo como antes

$$\Delta x(t) = x(t) - \bar{x}; \quad \Delta u(t) = u(t) - \bar{u} \quad (8.9)$$

a las variables en torno a dicho equilibrio, obtenemos el siguiente modelo en torno al equilibrio

$$\Delta \dot{x}(t) = \left. \frac{\partial f}{\partial x} \right|_{\bar{x}, \bar{u}} \Delta x(t) + \left. \frac{\partial f}{\partial u} \right|_{\bar{x}, \bar{u}} \Delta u(t) + O[\Delta x(t)^2, \Delta u(t)^2]$$

Despreciando entonces los términos de orden superior resulta el siguiente modelo lineal y estacionario:

$$\Delta \dot{x}(t) = A \Delta x(t) + B \Delta u(t) \quad (8.10)$$

donde

$$A \triangleq \left. \frac{\partial f}{\partial x} \right|_{\bar{x}, \bar{u}} \quad B \triangleq \left. \frac{\partial f}{\partial u} \right|_{\bar{x}, \bar{u}}. \quad (8.11)$$

Notar que A es la matriz Jacobiana de f evaluada en el punto de equilibrio. El sistema dado por la Ec.(8.10) es lo que llamamos modelo linealizado.

Supongamos entonces que queremos obtener un modelo linealizado del péndulo en torno a un punto de equilibrio genérico \bar{x} . Tenemos:

$$f(x(t), u(t)) = \begin{bmatrix} x_2(t) \\ -\frac{g}{L} \sin(x_1(t)) - \frac{b}{J} x_2(t) \end{bmatrix}$$

Por lo tanto, la matriz Jacobiana es:

$$\left. \frac{\partial f}{\partial x} \right|_{\bar{x}} = \begin{bmatrix} \left. \frac{\partial f_1}{\partial x_1} \right|_{\bar{x}} & \left. \frac{\partial f_1}{\partial x_2} \right|_{\bar{x}} \\ \left. \frac{\partial f_2}{\partial x_1} \right|_{\bar{x}} & \left. \frac{\partial f_2}{\partial x_2} \right|_{\bar{x}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{L} \cos(\bar{x}_1) & -\frac{b}{J} \end{bmatrix}$$

Luego, si particularizamos para un punto de equilibrio $\bar{x} = [k \pi, 0]^T$ con k par resulta:

$$A_{\text{par}} = \left. \frac{\partial f}{\partial x} \right|_{\bar{x}} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{L} & -\frac{b}{J} \end{bmatrix} \quad (8.12)$$

mientras que para k impar resulta

$$A_{\text{impar}} = \left. \frac{\partial f}{\partial x} \right|_{\bar{x}} = \begin{bmatrix} 0 & 1 \\ \frac{g}{L} & -\frac{b}{J} \end{bmatrix} \quad (8.13)$$

Claramente tendremos un modelo linealizado diferente según linealicemos en torno a un punto de equilibrio de abajo o de arriba (péndulo invertido).

8.2.2. Linealización en un Sistema de DAEs

Cuando construimos un modelo orientado a objetos o simplemente usamos las relaciones constitutivas y estructurales no obtenemos directamente un sistema de ecuaciones de estado como el de la Ec.(8.5), sino un sistema de ecuaciones diferenciales algebraicas que siempre podrá tomar la siguiente forma *semi-implícita*:

$$\begin{aligned} \dot{x}(t) &= f(x(t), a(t), u(t)) \\ 0 &= g(x(t), a(t), u(t)) \end{aligned} \quad (8.14)$$

donde $a(t)$ es un vector de variables algebraicas que puede despejarse de la segunda ecuación (que es implícita).

En este caso, los puntos de equilibrio para una entrada constante \bar{u} son las soluciones a la ecuación

$$\begin{aligned} f(\bar{x}, \bar{a}, \bar{u}) &= 0 \\ g(\bar{x}, \bar{a}, \bar{u}) &= 0 \end{aligned} \quad (8.15)$$

y la linealización en torno a los mismos toma la forma

$$\begin{aligned}\Delta\dot{x}(t) &= F_x \Delta x(t) + F_a \Delta a(t) + F_u \Delta u(t) \\ 0 &= G_x \Delta x(t) + G_a \Delta a(t) + G_u \Delta u(t)\end{aligned}\quad (8.16)$$

donde

$$F_x \triangleq \left. \frac{\partial f}{\partial x} \right|_{\bar{x}, \bar{a}, \bar{u}}; \quad F_a \triangleq \left. \frac{\partial f}{\partial a} \right|_{\bar{x}, \bar{a}, \bar{u}}; \quad F_u \triangleq \left. \frac{\partial f}{\partial u} \right|_{\bar{x}, \bar{a}, \bar{u}} \quad (8.17)$$

y de manera similar

$$G_x \triangleq \left. \frac{\partial g}{\partial x} \right|_{\bar{x}, \bar{a}, \bar{u}}; \quad G_a \triangleq \left. \frac{\partial g}{\partial a} \right|_{\bar{x}, \bar{a}, \bar{u}}; \quad G_u \triangleq \left. \frac{\partial g}{\partial u} \right|_{\bar{x}, \bar{a}, \bar{u}} \quad (8.18)$$

De la segunda ecuación en la Ec.(8.16), asumiendo que la matriz G_a no es singular, se puede despejar $\Delta a(t)$:

$$\Delta a(t) = -G_a^{-1} (G_x \Delta x(t) + G_u \Delta u(t))$$

y reemplazando en la primera se obtiene:

$$\Delta\dot{x}(t) = F_x \Delta x(t) - F_a G_a^{-1} (G_x \Delta x(t) + G_u \Delta u(t)) + F_u \Delta u(t)$$

y finalmente

$$\Delta\dot{x}(t) = (F_x - F_a G_a^{-1} G_x) \Delta x(t) + (F_u - F_a G_a^{-1} G_u) \Delta u(t)$$

y comparando esta expresión con la de la Ec.(8.10) se obtiene que las matrices de evolución y de entrada son

$$A = F_x - F_a G_a^{-1} G_x; \quad B = F_u - F_a G_a^{-1} G_u \quad (8.19)$$

Para que este procedimiento funcione necesitamos que la matriz G_a no sea singular. Puede mostrarse que esta matriz es singular cuando el problema tiene índice alto, por lo que la linealización en tal caso requiere primero reducir el índice utilizando Pantelides. Una vez hecho esto, si fuera necesario, a partir de la DAE podemos calcular directamente el modelo linealizado en ecuaciones de estado explícitas sin necesidad de obtener las ecuaciones de estado del sistema no lineal.

8.2.3. Linealización en OpenModelica

OpenModelica tiene una herramienta para linealizar directamente un modelo en torno a un punto independientemente de cuál sea su formulación original, esto es, sin importar si es un modelo orientado a objetos o si es simplemente un sistema plano de ecuaciones diferenciales algebraicas.

La linealización funciona de la siguiente forma: OpenModelica genera el sistema de ecuaciones de estado utilizando el procedimiento que vimos en el Capítulo 3 y simula el mismo hasta cierto instante de tiempo (seleccionado por el usuario). Luego, para ese tiempo final, calcula las derivadas parciales correspondientes en torno al valor del estado en dicho instante de tiempo.

Para ilustrar como se puede utilizar esta herramienta, tomamos por ejemplo el modelo del péndulo de la Ec.(8.1) y lo escribimos en lenguaje Modelica:

Código 8.1: Modelo del Péndulo de la Ec.(8.1) en Modelica

```
model Pendulum
  Real phi,w;
  parameter Real b=1,J=1,g=9.81,L=1;
equation
  der(phi)=w;
  der(w)=-g/L*sin(phi)-b/J*w;
end Pendulum;
```

Si bien sería posible en principio hacer la linealización invocando directamente al compilador `omc` con los parámetros adecuados, es mucho más simple realizarlo desde la herramienta `OMShell`, que es una terminal que permite utilizar interactivamente dicho compilador. Desde `OMShell` se puede entonces abrir el archivo `Pendulum.mo` que contiene el modelo del Código 8.1 e invocar el comando `linearize(Pendulum,stopTime=0)` que realiza la linealización en torno a la condición inicial como se muestra a continuación:

Código 8.2: Terminal de OMShell

```

OMShell 1.1 Copyright Open Source Modelica Consortium (OSMC) 2002-2023
Distributed under OSMC-PL and GPL, see www.openmodelica.org

Connected to OpenModelica 1.21.0
To get help on using OMSshell and OpenModelica, type "help()" and press enter.

>> loadFile("/data/Modelica/Pendulum.mo")
true

>> linearize(Pendulum,stopTime=0)
record SimulationResult
"...
stdout          | info      | Linear model is created at /tmp/OpenModelica/linearized_model.mo
..."

```

En el mensaje de salida (se muestra sólo una parte) el compilador informa que se creó el modelo linealizado en una carpeta temporal. Dicho modelo, que se muestra a continuación, coincide con el que habíamos obtenido para k par en la Ec.(8.12).

Código 8.3: Modelo Linealizado del Péndulo en Modelica

```

model linearized_model "Pendulum"
  parameter Integer n = 2 "number of states";
  parameter Integer m = 0 "number of inputs";
  parameter Integer p = 0 "number of outputs";
  parameter Real x0[n] = {0, 0};
  parameter Real u0[m] = zeros(0);
  parameter Real A[n, n] =
    [0, 1;
     -9.809999999999993, -1];
  parameter Real B[n, m] = zeros(n, m);
  parameter Real C[p, n] = zeros(p, n);
  parameter Real D[p, m] = zeros(p, m);
  Real x[n] (start=x0);
  input Real u[m];
  output Real y[p];
  Real 'x_phi' = x[1];
  Real 'x_w' = x[2];
equation
  der(x) = A * x + B * u;
  y = C * x + D * u;
end linearized_model;

```

Si quisiéramos en cambio linealizar en torno al punto de equilibrio $\bar{x} = [\pi, 0]^T$, deberíamos modificar la condición inicial utilizando dichos valores y luego proceder de igual manera.

Como puede verse en el modelo lineal resultante, el mismo no sólo calcula la matriz A , sino también las matrices B , C y D (que en este caso son nulas ya que no hay entradas ni salidas). Para especificar que una variable es entrada debemos definirla con el prefijo `input` mientras que para especificar es salida debemos utilizar el prefijo `output`.

Consideremos entonces por ejemplo un péndulo invertido, que tiene una entrada $\tau(t)$ que aplica un torque en el eje (que podría servir para controlar el péndulo en dicha posición que sabemos que es inestable). El modelo correspondiente en Modelica sería el siguiente

Código 8.4: Modelo del Péndulo de la Ec.(8.1) con entrada

```

model InvPendulum
  output Real phi(start=3.141509);
  Real w;
  input Real tau=0;
  parameter Real b=1,J=1,g=9.81,L=1;
equation
  der(phi)=w;
  der(w)=-g/L*sin(phi)-b/J*w+tau/J;
end InvPendulum;

```

Notar que pusimos la condición inicial $\phi_0 = \pi$ y además la entrada $\tau = 0$, lo que nos permitirá linealizar en torno a dicho punto de equilibrio. Procediendo igual que antes, se obtiene:

Código 8.5: Modelo Linealizado del Péndulo Invertido en Modelica

```

model linearized_model "InvPendulum"
  parameter Integer n = 2 "number of states";
  parameter Integer m = 1 "number of inputs";
  parameter Integer p = 1 "number of outputs";
  parameter Real x0[n] = {3.141509, 0};
  parameter Real u0[m] = {0};
  parameter Real A[n, n] =
    [0, 1;
     9.809999943513269, -1.000000000000577];
  parameter Real B[n, m] =
    [0;
     1.000000000000577];
  parameter Real C[p, n] = [0.9999999977371976, 0];
  parameter Real D[p, m] = [0];
  Real x[n] (start=x0);
  input Real u[m] (start=u0);
  output Real y[p];
  Real 'x_phi' = x[1];
  Real 'x_w' = x[2];
  Real 'u_tau' = u[1];
  Real 'y_phi' = y[1];
equation
  der(x) = A * x + B * u;
  y = C * x + D * u;
end linearized_model;

```

Notar que la matriz A también coincide con la que calculamos en la Ec.(8.13).

Veamos finalmente como usar esta herramienta para linealizar un modelo orientado a objetos más complejo. La Figura 8.8 muestra el modelo orientado a objetos de un motor con excitación independiente que utiliza puertos de entrada y salida de señal construido con componentes de la DSFLib.

En dicho modelo usamos los parámetros $R_a = 0.05$, $L_a = 0.003$, $R_e = 25.2$, $J = 15$, $b = 1$, $V_e = 184$, la constante del motor $K = 0.016$ y la característica magnética del circuito de excitación dada por los valores $[0, 3, 6, 9, 10, 12, 15]$ para la corriente y $[0, 200, 375, 475, 500, 525, 545]$ para el flujo.

Consideraremos que queremos linealizar el modelo en torno al punto de equilibrio que se obtiene con una entrada constante $U_a = 460$. Para esto, colocamos el modificador `start=460` en el puerto de entrada.

El modelo está dentro de un paquete denominado `MotorControl` guardado en un archivo `MotorControl.mo`. Para poder linealizar el modelo, entonces, debemos entonces abrir la DSFLib y el paquete `MotorControl` dentro de OMSHELL y luego invocar la linealización:

Código 8.6: Terminal de OMSHELL

```

>> loadFile("/data/Modelica/DSFLib.mo")
true

```

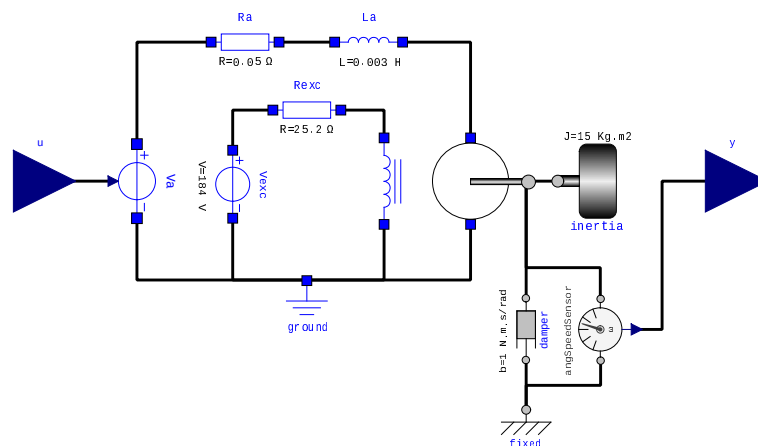


Figura 8.8: Modelo de un motor con excitación independiente y puertos de entrada y salida de señal

```
"Notification: Automatically loaded package Modelica 3.2.3 due to uses annotation from DSFLib.
Notification: Automatically loaded package Complex 3.2.3 due to uses annotation from Modelica.
Notification: Automatically loaded package ModelicaServices 3.2.3 due to uses annotation from
  Modelica.
"
>> loadFile("/data/Modelica/MotorControl.mo")
true

>> linearize(MotorControl.Motor,stopTime=20)
```

Para evitarnos calcular el punto de equilibrio, simplemente pusimos un valor grande al tiempo final para que la linealización se realice en torno al estado en el cuál quede el sistema transcurrido ese tiempo. El resultado es el siguiente:

Código 8.7: Modelo Linealizado del Motor

```
model linearized_model "MotorControl_Motor"
  parameter Integer n = 4 "number of states";
  parameter Integer m = 1 "number of inputs";
  parameter Integer p = 1 "number of outputs";
  parameter Real x0[n] = {10.25359225568756, 68.63986823113224, 1584.517302642607,
    418.3859651148796};
  parameter Real u0[m] = {460};
  parameter Real A[n, n] =
    [-16.66668579026573, -2231.391860543775, 0, -366.0794381134812;
    0.4462783744718695, -0.06666666730613563, 0, 0.01093716889065744;
    0, 1.000000009592135, 0, 0;
    0, 0, 0, -0.7559998921758215];
  parameter Real B[n, m] =
    [333.3333332161497;
    0;
    0;
    0];
  parameter Real C[p, n] = [0, 1.000000009592135, 0, 0];
  parameter Real D[p, m] = [0];
  Real x[n] (start=x0);
  input Real u[m] (start=u0);
  output Real y[p];

  Real 'x_inductor.i' = x[1];
  Real 'x_inertia.w' = x[2];
```

```

Real 'x_sepExcDCM.flange.phi' = x[3];
Real 'x_sepExcDCM.inductor.phi' = x[4];
Real 'u_u' = u[1];
Real 'y_y' = y[1];
equation
  der(x) = A * x + B * u;
  y = C * x + D * u;
end linearized_model;

```

8.3. Estabilidad de Lyapunov

En los sistemas lineales, las propiedades de estabilidad de los puntos de equilibrio se determinaban fácilmente a partir de la posición de los autovalores de la matriz A . Esto evidentemente no se puede extender a los sistemas no lineales. Más aún, las definiciones de estabilidad que habíamos usado en los sistemas lineales no son correctas al aplicarlas a los sistemas no lineales.

Veremos entonces como definir formalmente el concepto de estabilidad de los puntos de equilibrio y algunas herramientas para determinar las características de estabilidad de los mismos.

Tanto en las definiciones como en las herramientas consideraremos que el punto de equilibrio es el origen. Esto no implica una pérdida de generalidad, ya que si el punto de equilibrio fuera otro, podemos utilizar el modelo en torno a dicho punto desplazando de tal manera el punto de equilibrio al origen.

8.3.1. Definición de Estabilidad

Para los modelos lineales habíamos hecho una definición no muy precisa respecto a la estabilidad de los puntos de equilibrio. Habíamos dicho que un punto de equilibrio era estable si las trayectorias no se alejaban indefinidamente del mismo y que era inestable en caso contrario. Esto es correcto para los sistemas lineales pero no lo es en casos no lineales como los que vimos para el péndulo, por ejemplo.

De hecho, si en el caso del péndulo consideramos una condición inicial cercana al punto de equilibrio $\bar{x} = [\pi, 0]$ las trayectorias no se alejan indefinidamente, sino que tienden a otro punto de equilibrio estable cercano como puede verse en la Figura 8.1. Sin embargo, es claro que el punto de equilibrio es inestable.

Veremos entonces como definir formalmente los conceptos de estabilidad. Dado que siempre es posible hacer un cambio de coordenadas que lleve el origen a un punto de equilibrio arbitrario, daremos todas las definiciones como si el punto de equilibrio fuera directamente el origen.

Definición 8.1 (Estabilidad de Lyapunov). *El origen del sistema estacionario*

$$\dot{x}(t) = f(x) \tag{8.20}$$

es estable en el sentido de Lyapunov sí y solo si dado un valor de $\varepsilon > 0$ arbitrario existe $\delta(\varepsilon) > 0$ tal que la condición $\|x(t_0)\| < \delta$ implica $\|x(t)\| < \varepsilon$ para todo $t > t_0$.

La Figura 8.9 ilustra el concepto de la Estabilidad de Lyapunov del origen. Dada una bola de radio ε centrada en el origen, siempre existe una bola (dentro de la anterior) de radio δ , tal que para cualquier condición inicial dentro de la bola más chica, la trayectoria resultante nunca sale de la bola más grande.

Esta definición implica que el sistema puede permanecer siempre arbitrariamente cerca del punto de equilibrio. Se trata además de una definición *local* que sólo exige la existencia de bolas suficientemente chicas en torno al origen desde las cuales las trayectorias no se alejan mucho del mismo, pero no pide nada con respecto a lo que puede ocurrir cuando las condiciones iniciales están más alejadas.

La definición de estabilidad además no pide que las trayectorias se acerquen al equilibrio. Para esto último está la definición de *estabilidad asintótica*:

Definición 8.2 (Estabilidad Asintótica). *El origen del sistema (8.20) es asintóticamente estable si es estable en el sentido de Lyapunov y si puede elegirse $\delta > 0$ tal que la condición $\|x(t_0)\| < \delta$ implica $x(t) \rightarrow 0$ para $t \rightarrow \infty$.*

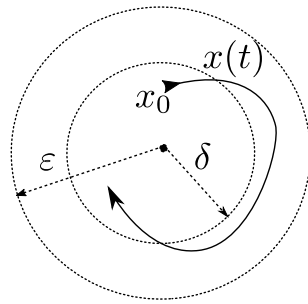


Figura 8.9: Estabilidad de Lyapunov

Notar que la condición de estabilidad asintótica también es local y no dice nada sobre lo que ocurre con condiciones iniciales alejadas. Existe igualmente el concepto de estabilidad asintótica global, que garantiza la convergencia al origen desde cualquier condición inicial.

Obviamente, el origen nunca podrá ser asintóticamente estable de manera global si existen otros puntos de equilibrio.

8.3.2. Método Directo de Lyapunov

Una vez que definimos los conceptos de estabilidad, nos resta buscar herramientas que permitan analizar si los puntos de equilibrio de un sistema no lineal serán o no estables. Comenzaremos entonces analizando un par de ejemplos simples.

Consideremos el siguiente sistema de primer orden:

$$\dot{x}(t) = -g(x(t)) \quad (8.21)$$

donde $g(x)$ es una función definida en el primer y tercer cuadrante que verifica $g(0) = 0$. Este sistema podría representar un circuito RC con un capacitor $C = 1$ y un resistor no lineal arbitrario con característica volt-ampere $i = g(v)$.

Dado que es un circuito pasivo, debería ser estable. Si además el resistor siempre permite el paso de corriente cuando hay tensión (o sea que $g(v) = 0$ sólo ocurre cuando $i = 0$) entonces todo el tiempo habrá disipación y en cierto punto debería desaparecer la energía del circuito.

En este caso, podemos observar lo siguiente: si en el instante t tenemos $x(t) > 0$ la Ec.(8.21) dice que $\dot{x}(t) < 0$. Esto implica que la tensión siempre decrecerá y eventualmente tendremos que $x(t \rightarrow \infty) \rightarrow 0$. Algo análogo ocurre si $x(t) < 0$, ya que en tal caso resulta $\dot{x}(t) > 0$ y la conclusión es la misma. Evidentemente el origen de este sistema es estable (más aún, es global y asintóticamente estable ya que estamos suponiendo que $g(v) = 0$ sólo ocurre para $v = 0$).

Esto lo pudimos analizar de manera simple porque el sistema es de primer orden y es muy sencillo ver las formas de las trayectorias. En general, este tipo de análisis no es viable en sistemas de orden mayor.

Veamos entonces otra forma de hacer el análisis en base a la observación de que el resistor debe disipar energía. Sabemos que la energía almacenada en el capacitor es $V = 0.5 x^2$, y por lo tanto la derivada de la energía respecto al tiempo es

$$\dot{V}(x) = \frac{\partial V}{\partial x} \dot{x}(t) = -x(t) g(x(t)) \quad (8.22)$$

Como $g(x)$ es una función del primer y tercer cuadrante, resulta $\dot{V}(x) < 0$ para todo x , salvo para $x = 0$ donde vale $\dot{V}(0) = 0$. Esto quiere decir que la energía disminuye para cualquier valor del estado x (salvo en el origen), y que por lo tanto tendremos que $V(x(t)) \rightarrow 0$ si $t \rightarrow \infty$ y dado que $V(x) = 0$ implica $x = 0$ resulta que todas las trayectorias tienden al origen, que por lo tanto es global y asintóticamente estable.

Veamos si podemos hacer ahora un análisis similar con un sistema de segundo orden, como el correspondiente al circuito de la Figura 8.10, cuyo modelo está dado por el sistema de la Ec.(8.23).

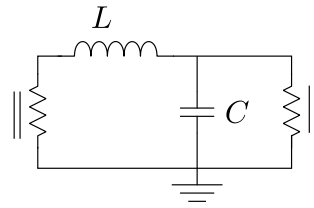


Figura 8.10: Circuito RLC con resistores no lineales.

$$\begin{aligned} \dot{x}_1(t) &= \frac{-g_1(x_1(t)) + x_2(t)}{C} \\ \dot{x}_2(t) &= \frac{-x_1(t) - g_2(x_2(t))}{L} \end{aligned} \tag{8.23}$$

Aquí, $g_1(\cdot)$ y $g_2(\cdot)$ son dos funciones de primer y tercer cuadrante con $g_k(0) = 0$ que representan las características de dos resistores no lineales. Notar que el origen es un punto de equilibrio de este sistema.

La energía total del circuito es

$$V(x) = 0.5 C x_1^2 + 0.5 L x_2^2 \tag{8.24}$$

cuya derivada resulta

$$\begin{aligned} \dot{V}(x) &= C x_1(t) \dot{x}_1(t) + L x_2(t) \dot{x}_2(t) \\ &= C x_1(t) \frac{-g_1(x_1(t)) + x_2(t)}{C} + L x_2(t) \frac{-x_1(t) - g_2(x_2(t))}{L} \\ &= x_1(t) (-g_1(x_1(t)) + x_2(t)) + x_2(t) (-x_1(t) - g_2(x_2(t))) \\ &= -x_1(t) g_1(x_1(t)) - x_2(t) g_2(x_2(t)) \end{aligned}$$

Dado que g_1 y g_2 son funciones de primer y tercer cuadrante, es claro que $\dot{V}(x) < 0$ para todo $x \neq [0, 0]^T$ y luego tendremos que $V(x(t)) \rightarrow 0$ si $t \rightarrow \infty$ y dado que $V(x) = 0$ implica $x = [0, 0]^T$ resulta que todas las trayectorias tienden al origen, que por lo tanto es global y asintóticamente estable.

Esta idea es la base del *Método Directo de Lyapunov*, en la cual se utiliza una función $V(x)$ positiva (que no será necesariamente la energía) y se estudia que ocurre con su derivada temporal. El siguiente teorema conforma la base de este método:

Teorema 8.1 (Lyapunov). *Sea el origen $x = 0$ un punto de equilibrio del sistema $\dot{x}(t) = f(x(t))$ y sea $E \subseteq \mathbb{R}^n$ una región del espacio de estados que contiene al origen. Sea $V : E \rightarrow \mathbb{R}$ una función continuamente diferenciable tal que:*

$$V(x) = 0 \text{ si } x = 0 \text{ y } V(x) > 0 \text{ si } x \in E - \{0\}$$

luego,

- Si la derivada satisface $\dot{V}(x) \leq 0$ para todo $x \in E - \{0\}$ entonces el origen es estable.
- Si la derivada satisface $\dot{V}(x) < 0$ para todo $x \in E - \{0\}$ entonces el origen es asintóticamente estable.

Este teorema garantiza que si una función $V(x)$ es *definida positiva* en una región E que contiene al origen y su derivada es *definida negativa* en dicha región, entonces el origen (que debe ser un punto de equilibrio) es asintóticamente estable. Si la derivada es *semidefinida negativa* entonces sólo garantiza que el origen es estable.

Una función se dice *definida positiva* cuando es siempre positiva excepto en el origen (donde es nula) y se dice que es *definida negativa* cuando es siempre negativa (excepto en el origen, donde es nula). De

manera similar, una función es semidefinida positiva cuando es siempre positiva o nula, y en el origen es nula (y existe el concepto análogo de semidefinida negativa).

Una función definida positiva con derivada definida o semidefinida negativa se denomina *Función de Lyapunov* y, de acuerdo al Teorema 8.1 la sola existencia de dicha función permite garantizar la estabilidad del origen. La función de energía del circuito RLC dada por la Ec.(8.24) es por lo tanto una función de Lyapunov y entonces el origen del sistema de la Ec.(8.23) es estable. Más aún, dado que $\dot{V}(x) < 0$ es asintóticamente estable.

Para el caso del péndulo de la Ec.(8.1) podemos también tomar como candidata a función de Lyapunov a la energía total, sumando la potencial gravitatoria y la energía cinética:

$$V(x) = m g L (1 - \cos(x_1)) + 0.5 J x_2^2 \quad (8.25)$$

Notar que para que esta función sea definida positiva debemos limitar la región E , ya que si $x_1 = 2\pi$ por ejemplo y $x_2 = 0$ resulta $V(x) = 0$. Podemos considerar por lo tanto que $-\pi < x_1 < \pi$. La derivada entonces resulta:

$$\begin{aligned} \dot{V}(x) &= m g L \sin(x_1) \dot{x}_1 + J x_2 \dot{x}_2 \\ &= m g L \sin(x_1) x_2 + J x_2 \left(-\frac{g}{L} \sin(x_1) - \frac{b}{J} x_2\right) \\ &= \left(m g L - J \frac{g}{L}\right) \sin(x_1) x_2 - b x_2^2 \\ &= -b x_2^2 \end{aligned}$$

donde en el último paso usamos que $J = m L^2$.

Observar que la condición $\dot{V}(x) = -b x_2^2$ implica que la derivada es semidefinida negativa, ya que cuando $x_2 = 0$ resulta $\dot{V}(x) = 0$ para cualquier valor del estado x_1 , incluso fuera del origen. Es claro entonces que según el Teorema 8.1 podemos garantizar que el sistema es estable, pero no podemos garantizar aún que sea asintóticamente estable.

Sin embargo, dado que hay disipación en el péndulo, parece bastante evidente que debe haber estabilidad asintótica. Veremos a continuación entonces una extensión al Teorema de Lyapunov que permite garantizar estabilidad asintótica en casos como este, donde la derivada de la función de Lyapunov es semidefinida negativa.

8.3.3. Principio de Invariancia de LaSalle

Siguiendo con el ejemplo del péndulo, lo que ocurre es que $\dot{V}(x) = -b x_2^2$ es negativa en todos lados, excepto cuando $x_2 = 0$ (es decir en el eje x_1). Esto quiere decir que el sistema pierde energía en todos lados menos cuando la trayectoria cruza el eje x_1 .

La pregunta que surge entonces es: ¿existe la posibilidad de que el estado permanezca para siempre en una zona distinta del origen donde $\dot{V}(x) = 0$ y por lo tanto la energía no decrezca hasta llegar a cero?. Veamos en este caso si esto es posible:

- Para que $\dot{V}(x) = 0$ necesitamos que $x_2 = 0$ (esto ocurre en el eje x_1).
- Si se cumple la condición $x_2 = 0$, luego de acuerdo a la Ec.(8.1) la derivada de x_2 verifica:

$$\dot{x}_2(t) = -\frac{g}{L} \sin(x_1(t))$$

y esto puede ser 0 sólo si $x_1 = k\pi$.

- Por lo tanto, si $|x_1| < \pi$ y $x_1 \neq 0$, la condición $x_2 = 0$ implica $\dot{x}_2 \neq 0$.
- Esto último implica que la trayectoria no puede permanecer en el eje, sino que pasará por el eje y lo cruzará.

Ahora que es claro que el estado no puede permanecer para siempre en una zona donde $\dot{V}(x) = 0$ (salvo que sea el origen), es evidente que la energía seguirá decreciendo y eventualmente llegará a cero y por lo tanto el origen será asintóticamente estable.

Este razonamiento se formaliza en el *Principio de Invariancia de LaSalle*, que establece que cuando $\dot{V}(x)$ es semidefinida negativa, las trayectorias que se inician suficientemente cerca del origen convergen al *máximo invariante* contenido en la zona donde $\dot{V}(x) = 0$.

Un conjunto invariante es una zona del espacio de estados en la cual toda trayectoria que se inicia en la misma, permanece en dicha zona para siempre. De acuerdo a esto, un punto de equilibrio es un invariante. También, en un sistema lineal la dirección de un autovector es invariante (ya que cuando las trayectorias se inician en un autovector, permanecen en el mismo para siempre).

En la práctica, para demostrar la estabilidad asintótica del origen utilizando el principio de invariancia de LaSalle bastará probar que el origen es el único conjunto invariante contenido en la zona donde $\dot{V}(x) = 0$ ya que las trayectorias convergerán a dicho punto.

En el caso del péndulo, mostramos de hecho que la zona donde $\dot{V}(x) = 0$ es el eje x_1 , y que dentro de dicha zona, el único conjunto invariante es el origen ya que en cualquier otro punto las trayectorias no pueden permanecer en el eje. Esto, según LaSalle, implica que el origen es asintóticamente estable.

Si consideramos nuevamente el circuito No Lineal de la Ec.(8.23) pero ahora con $g_2(\cdot) = 0$ (esto equivale a eliminar el resistor que está en serie con el inductor en la Fig.8.10), la derivada de la función de energía de la Ec.(8.24) queda

$$\dot{V}(x) = -x_1(t) g_1(x_1(t)) - x_2(t) g_2(x_2(t)) = -x_1(t) g_1(x_1(t)) \quad (8.26)$$

que es semidefinida negativa (se anula en el eje x_2). Si tomamos la ecuación de \dot{x}_1 para $x_1 = 0$, resulta

$$\dot{x}_1(t) = \frac{-g_1(x_1(t)) + x_2(t)}{C} = \frac{x_2(t)}{C} \quad (8.27)$$

lo que indica que para que se preserve la situación $x_1(t) = 0$ y la trayectoria pueda mantenerse sobre el eje x_2 (o sea, en la zona donde $\dot{V}(x) = 0$) debe ocurrir que $x_2 = 0$, lo que sólo ocurre en el origen. Por lo tanto el origen es el máximo invariante y según LaSalle las trayectorias convergerán al mismo. Es decir, el origen será asintóticamente estable.

8.4. Análisis Mediante Aproximaciones Lineales

En los sistemas lineales el análisis de estabilidad y la determinación de la forma de las trayectorias es muy simple. Dado que ya vimos como linealizar un sistema no lineal, cabe preguntarnos si no podemos usar estas aproximaciones para estudiar las propiedades del sistema no lineal original. Veremos a continuación entonces cuando estas aproximaciones son válidas para analizar tanto la estabilidad como la forma de las trayectorias.

8.4.1. Método Indirecto de Lyapunov

La gran dificultad asociada al uso del método directo de Lyapunov es la necesidad de contar con una candidata de Lyapunov $V(x)$ apropiada. Si bien en ciertos casos podremos usar la energía, en muchos otros esto no será posible ya que muy frecuentemente los modelos tienen partes que no provienen directamente de la física o donde la física se encuentra simplificada y aproximada mediante magnitudes a las cuales no se les puede asignar una interpretación energética.

Sin embargo, cuando un sistema es lineal es muy simple encontrar una función de Lyapunov. Por supuesto, en un caso lineal parece no tener sentido utilizar Lyapunov dado que podemos establecer directamente la estabilidad de los puntos de equilibrio a partir de los autovalores de la matriz A . Sin embargo, vimos que en cercanías de un punto de equilibrio podemos linealizar un sistema no lineal y en tal caso, saber que existe una función de Lyapunov para el sistema linealizado nos ayudará a determinar las propiedades de estabilidad del punto de equilibrio del sistema no lineal.

De hecho, se puede mostrar que cuando todos los autovalores de la matriz de evolución A de un sistema lineal y estacionario tienen parte real negativa, entonces se puede obtener una función de Lyapunov $V(x)$ cuadrática definida positiva con su derivada $\dot{V}(x)$ definida negativa y también cuadrática.

Dado que si nos acercamos suficientemente al punto de equilibrio los términos lineales son más grandes que los no lineales, el efecto de los términos no lineales no influye en el signo de $\dot{V}(x)$ lo que garantiza la estabilidad asintótica.

Veamos esto con un ejemplo. Consideremos el sistema de primer orden:

$$\dot{x}(t) = -x(t) + x(t)^3 \quad (8.28)$$

Si linealizamos en torno al origen obtenemos

$$\dot{x}(t) = -x(t) \quad (8.29)$$

Proponemos entonces la candidata de Lyapunov $V(x) = 0.5 x^2$ de donde cuya derivada resulta

$$\dot{V}(x) = x \dot{x} = x(-x) = -x^2 \quad (8.30)$$

que es definida negativa y por lo tanto el origen del sistema linealizado resulta asintóticamente estable.

Si intentamos ahora usar la misma candidata de Lyapunov para el sistema no lineal, obtenemos:

$$\dot{V}(x) = x \dot{x} = x(-x + x^3) = -x^2(1 - x^2) \quad (8.31)$$

que resulta negativa para $|x| < 1$. Luego, existe una región que contiene al origen en la cual $\dot{V}(x)$ es definida negativa lo que implica que el origen es asintóticamente estable.

Lo que ocurre en este ejemplo se puede generalizar. Cada vez que $\dot{V}(x)$ sea definida negativa para el sistema linealizado, será definida negativa para el no lineal al restringirnos a una región suficientemente cercana al origen lo que implicará a su vez que el origen es un punto de equilibrio asintóticamente estable del sistema no lineal. Dado que si todos los autovalores de A tienen parte real negativa siempre podremos encontrar una función de Lyapunov con derivada definida negativa para el sistema linealizado, resulta que si todos los autovalores de la matriz A del sistema linealizado en un punto de equilibrio tienen parte real negativa, podemos garantizar que el punto de equilibrio será asintóticamente estable.

Este razonamiento se formaliza con el Método Indirecto de Lyapunov, que además brinda condiciones para la inestabilidad como se establece en el siguiente teorema:

Teorema 8.2 (Método Indirecto de Lyapunov). *Sea \bar{x} un punto de equilibrio del sistema $\dot{x}(t) = f(x(t))$ donde f es continuamente diferenciable en un entorno de dicho punto. Sea*

$$A = \left. \frac{\partial f(x)}{\partial x} \right|_{x=\bar{x}} \quad (8.32)$$

Entonces

1. *El punto de equilibrio \bar{x} es asintóticamente estable si todos los autovalores de A tienen parte real negativa.*
2. *El punto de equilibrio \bar{x} es inestable si uno o más autovalores de A tiene parte real positiva.*

Notar que cuando hay autovalores con parte real nula y ningún autovalor con parte real positiva el teorema no brinda ninguna conclusión sobre la estabilidad. Los autovalores con parte real nula implican que hay dinámica lineal que no se acerca ni se aleja del equilibrio (podrían ser centros, por ejemplo). En tales casos, es la parte no lineal la que determina si las soluciones se acercan o no al equilibrio y esa información desaparece en la linealización.

Consideremos entonces nuevamente el ejemplo del péndulo. Habíamos visto al linealizar que para los puntos de equilibrio de la forma $[k\pi, 0]^T$ con k par la matriz A tomaba la expresión de la Ec.(8.12), cuyos autovalores pueden calcularse de la ecuación característica:

$$\lambda \left(\lambda + \frac{b}{J} \right) + \frac{g}{L} = 0$$

que al ser un polinomio de segundo orden sin cambios de signos tendrá todas sus raíces con parte real negativa. Por lo tanto, estos puntos de equilibrio serán asintóticamente estables.

Si tomamos en cambio los puntos de equilibrio con k impar, la matriz A toma la expresión de la Ec.(8.13), cuyos autovalores pueden calcularse ahora de la ecuación:

$$\lambda \left(\lambda + \frac{b}{J} \right) - \frac{g}{L} = 0$$

que tiene un cambio de signo y por lo tanto habrá al menos un autovalor con parte real positiva lo que según el Teorema 8.2 indicará que dichos puntos de equilibrio serán inestables.

Como vemos, el método indirecto de Lyapunov se puede aplicar de manera mucho más simple en general que el método directo, ya que sólo requiere linealizar y calcular los autovalores.

Como contrapartida, hay casos en los cuales puede no funcionar. Tomemos por ejemplo el sistema del circuito de la Fig.8.10 dado por la Ec.(8.23) y supongamos que

$$g_1(x_1) = a_1 x_1 + b_1 x_1^3; \quad g_2(x_2) = a_2 x_2 + b_2 x_2^3$$

Si linealizamos este sistema en torno al origen, obtenemos

$$A = \left. \frac{\partial f}{\partial x} \right|_{x=0} = \begin{bmatrix} \frac{a_1 + 3 b_1 \bar{x}_1^2}{C} & \frac{1}{C} \\ -\frac{1}{L} & \frac{a_2 + 3 b_2 \bar{x}_2^2}{L} \end{bmatrix} = \begin{bmatrix} \frac{a_1}{C} & \frac{1}{C} \\ -\frac{1}{L} & \frac{a_2}{L} \end{bmatrix} \quad (8.33)$$

que para el caso $a_1 = a_2 = 0$ tiene autovalores con parte real nula. Por lo tanto el Método Indirecto de Lyapunov no dice nada sobre la estabilidad del origen. Esto se debe a que no hay disipación lineal, y al linealizar desaparece el efecto de los resistores. Sin embargo, el Método Directo sí nos garantiza que el origen es asintóticamente estable.

Si consideramos nuevamente el caso del péndulo pero ahora con $b = 0$ (sin rozamiento) y linealizamos, obtenemos también en el origen una matriz A con autovalores imaginarios puros y nuevamente el Método Indirecto no nos dice nada.

Para dicho caso, usando como candidata de Lyapunov a la función $V(x)$ dada por la Ec.(8.25) se obtiene la derivada $\dot{V}(x) = 0$ para todo x . De acuerdo al método directo de Lyapunov el origen es estable ya que $\dot{V}(x)$ es semidefinida negativa (es nula, lo que es un caso particular de semidefinida negativa). Obviamente no se puede probar que sea asintóticamente estable (de hecho no lo es, ya que no pierde energía).

8.4.2. Teorema de Hartman–Grobman

Como ya habíamos mencionado, en el retrato de fases del péndulo de la Fig.8.1 se puede observar que en las cercanías de los puntos de equilibrio las trayectorias se asemejan a las de un foco estable o a las de una ensilladura.

La linealización del modelo del péndulo en torno a los puntos de equilibrio estables nos daba la matriz de evolución A de la Ec.(8.12), que para los parámetros $m = b = L = J = 1$ y $g = 9.81$ toma la forma

$$A = \begin{bmatrix} 0 & 1 \\ -9.81 & -1 \end{bmatrix}$$

cuyos autovalores son $\lambda_{1,2} = -0.5 \pm j 3.092$ y efectivamente se corresponden a un foco estable. La linealización en torno a los puntos de equilibrio inestables, en cambio, nos dan la matriz Jacobiana de la Ec.(8.13), que para los mismos parámetros resulta

$$A = \begin{bmatrix} 0 & 1 \\ 9.81 & -1 \end{bmatrix}$$

y cuyos autovalores son $\lambda_1 = -3.672$ y $\lambda_2 = 2.672$, lo que se corresponde a una ensilladura.

Todo esto parece sugerir que las trayectorias del sistema linealizado en torno a un punto de equilibrio se aproximan a las del sistema no lineal original en las cercanías de dicho punto. De hecho, la Figura 8.11 muestra las trayectorias del sistema no lineal y del linealizado para el caso del péndulo y las mismas son muy similares.

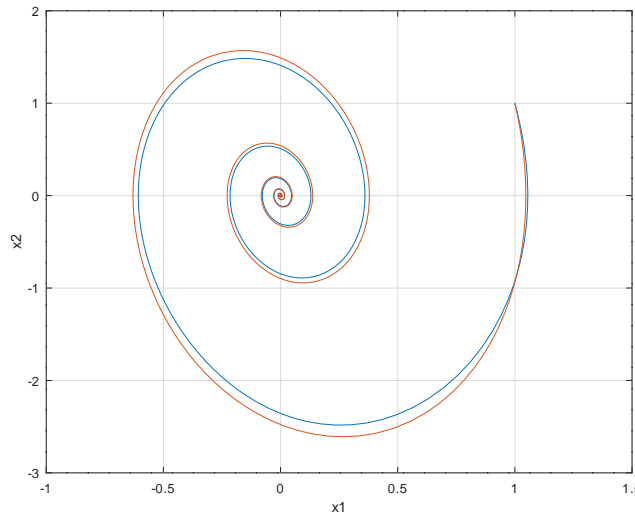


Figura 8.11: Trayectorias en el Espacio de Estados del modelo del péndulo y de su linealización

En realidad, esto es esperable. Sabemos que cuanto más nos acercamos al equilibrio los términos no lineales se tornan más pequeños en relación a los lineales. Por lo tanto, parece lógico que las trayectorias del sistema no lineal se parezcan a las que se obtienen despreciando dichos términos. Esta idea se formaliza mediante el Teorema de Hartman-Grobman.

Teorema 8.3. *Hartman-Grobman* Sea el origen $x = 0$ un punto de equilibrio del sistema $\dot{x}(t) = f(x(t))$ donde f es continuamente diferenciable en torno a dicho punto y donde

$$A = \left. \frac{\partial f(x)}{\partial x} \right|_{x=0}$$

no tiene autovalores con parte real nula. Entonces, existen dos entornos del origen U y V entre los cuales existe un homomorfismo $H(\cdot)$ que verifica

$$H(x(t)) = e^{A t} H(x(0))$$

para las soluciones $x(t) \in U$ de la ecuación diferencial.

En la práctica, esto se traduce en que las trayectorias del sistema no lineal tienen una forma similar a la del sistema linealizado con la condición de que la matriz A de este último no tenga autovalores con parte real nula.

Para entender esta última condición recordemos que la presencia de autovalores con parte real nula podría implicar trayectorias cerradas tipo centros, por ejemplo. En tales casos, la parte no lineal podría hacer que las trayectorias converjan o diverjan asemejándose a focos estables o inestables y por lo tanto es esperable que la linealización no preserve la forma cualitativa de las trayectorias.

En el ejemplo del péndulo vimos que la linealización en torno a los puntos de equilibrio arrojaba matrices jacobianas con parte real distinta de cero, por lo que en todos los puntos de equilibrio vale el Teorema de Hartman-Grobman. Esto garantiza que las trayectorias del sistema linealizado (focos estables y ensilladuras) se asemejan a las del sistema no lineal.

Para el circuito con diodo túnel dado por la Ec.(8.2), en tanto, si se linealiza el modelo en torno al equilibrio $\bar{x} = [0.5, 0.15688]^T$ resulta la matriz Jacobiana

$$A = \begin{bmatrix} 0.0283 & 0.04 \\ -1 & 0 \end{bmatrix}$$

cuyos autovalores son $\lambda_{1,2} = 0.014 \pm j 0.2$ y efectivamente se corresponden a un foco inestable.

8.5. Bibliografía Complementaria

Como ya mencionamos, las principales fuentes bibliográficas que se utilizaron para elaborar este capítulo son el libro de Khalil [Kha15] y las notas de clase en castellano de la materia de doctorado *Control de Sistemas No Lineales* [SB00]. Las referencias mencionadas por supuesto ahondan mucho más en los temas y utilizan una formalización matemática mucho más rigurosa de lo que presentamos aquí.

Para profundizar sobre el Teorema de Hartman-Grobman, en tanto, se puede consultar el libro de Perko [Per13] que, al igual que las otras referencias, tiene un tratamiento riguroso de la teoría de sistemas no lineales (sin incluir control).

8.6. Problemas Propuestos

[P8.1] Oscilador de Van der Pol

El oscilador de Van der Pol es un sistema que responde a las siguientes ecuaciones:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= -\omega^2 x_1(t) + \epsilon (1 - x_1^2(t)) x_2(t) \end{aligned} \quad (\text{P8.1a})$$

donde $\omega > 0$ y $\epsilon > 0$ son parámetros. Se pide:

1. Demostrar que el origen es el único punto de equilibrio del sistema.
2. Linealizar el sistema en torno al origen y analizar la estabilidad de dicho punto.
3. Determinar, si fuera posible, la forma de las trayectorias en las cercanías del origen.
4. Simular el sistema desde una condición inicial cercana al origen para los parámetros $\omega = \epsilon = 1$ y observar la forma del retrato de fases.

[P8.2] Sistema Masa-Resorte

La Figura 8.12 muestra un sistema masa-resorte con fricción no lineal cuadrática, que puede modelarse mediante el siguiente sistema de ecuaciones de estado:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= -\frac{k}{m} x_1(t) - \frac{b}{m} x_2(t) |x_2(t)| + \frac{F(t)}{m} \end{aligned} \quad (\text{P8.2a})$$

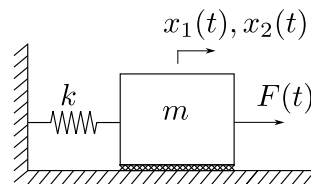


Figura 8.12: Sistema Masa-Resorte

Se pide:

1. Calcular el punto de equilibrio para una entrada constante $F(t) = \bar{F}$.
2. Obtener el modelo linealizado en torno a dicho punto de equilibrio.
3. Determinar, en base al método indirecto de Lyapunov, si es posible sacar conclusiones sobre la estabilidad del sistema no lineal.

4. Demostrar que el punto de equilibrio es estable usando el método directo de Lyapunov. Para esto, obtener el modelo en torno al equilibrio y usar la energía como candidata a función de Lyapunov. Alternativamente se puede usar la función $V(\Delta x_1, \Delta x_2) = 0.5 c x_1^2 + 0.5 x_2^2$ para una constante genérica c que podrá luego ajustarse para que la derivada resulte semidefinida negativa.
5. Demostrar la estabilidad asintótica usando el Principio de Invariancia de LaSalle.

[P8.3] Modelo de Lotka-Volterra

El siguiente sistema es un modelo clásico que representa la dinámica de la población de dos especies, presa y depredador:

$$\begin{aligned}\dot{x}_1(t) &= a x_1(t) - b x_1(t) x_2(t) \\ \dot{x}_2(t) &= c x_1(t) x_2(t) - d x_2(t)\end{aligned}\tag{P8.3a}$$

donde $x_1(t)$ es el número de presas, $x_2(t)$ es el número de depredadores y los parámetros a , b , c y d son todos positivos. Se pide entonces

1. Calcular todos los puntos de equilibrio del sistema.
2. Obtener el modelo linealizado en torno a cada punto de equilibrio.
3. A partir de los modelos linealizados, determinar cuando sea posible la estabilidad de los puntos de equilibrio.
4. Determinar, cuando sea posible, la forma cualitativa de las trayectorias en torno a los puntos de equilibrio.
5. Simular el sistema a partir de distintas condiciones iniciales suponiendo que todos los parámetros valen 1.

[P8.4] Análisis de Estabilidad Global En el siguiente sistema se puede demostrar que el único punto de equilibrio es el origen.

$$\begin{aligned}\dot{x}_1(t) &= (1 - x_1^2 - x_2^2) x_1 + x_2 \\ \dot{x}_2(t) &= -x_1 + (1 - x_1^2 - x_2^2) x_2\end{aligned}\tag{P8.4a}$$

Se pide:

1. Analizar mediante linealización la estabilidad y la forma de las trayectorias en torno al origen. Indicar si las conclusiones son válidas para puntos alejados del origen.
2. Utilizando como candidata de Lyapunov la función $V(x) = 0.5 (x_1^2 + x_2^2)$, analizar el signo de su derivada $\dot{V}(x)$ en función del módulo del estado $r^2 = x_1^2 + x_2^2$.
3. Simular desde distintas condiciones iniciales y comparar las trayectorias obtenidas con el análisis del signo de $\dot{V}(x)$.

[P8.5] Estabilidad y Trayectorias de un Sistema No Lineal Dado el siguiente sistema:

$$\begin{aligned}\dot{x}_1(t) &= 6 x_1 - x_1 x_2 - x_1^2 \\ \dot{x}_2(t) &= -4 x_2 + x_1 x_2\end{aligned}\tag{P8.5a}$$

se pide

1. Calcular todos los puntos de equilibrio.

2. Analizar la estabilidad de cada punto de equilibrio.
3. Determinar, cada vez que sea posible, la forma de las trayectorias en torno a los puntos de equilibrio.

[P8.6] Modelo de FitzHugh-Nagumo

El modelo de FitzHugh-Nagumo representa de manera simplificada la dinámica de excitación de una neurona y está caracterizado por las siguientes Ecuaciones de Estado:

$$\begin{aligned} \dot{x}_1 &= \frac{1}{C} \left(x_1 - \frac{x_1^3}{3} - x_2 + I(t) \right) \\ \dot{x}_2 &= \frac{1}{\tau} (x_1 + a - b x_2) \end{aligned} \tag{P8.6a}$$

donde $x_1(t)$ es el potencial de la membrana de la neurona y la entrada $I(t)$ representa la corriente de excitación externa. En este problema consideraremos los parámetros $a = 1$, $b = 1$, $C = 1$ y $\tau = 10$.

1. Suponiendo una entrada constante $I(t) = 1$, calcular el punto de equilibrio. **Ayuda:** Observe que al ser $b = 1$ puede despejar de la segunda ecuación el valor de equilibrio de $x_1 - x_2$ y luego utilizar dicho valor en la primera ecuación.
2. Obtener un modelo linealizado en torno al punto de equilibrio calculado.
3. Calcular los autovalores de la matriz de evolución A del modelo linealizado y analizar la estabilidad del sistema lineal. Dibujar además el retrato de fase del sistema linealizado (sólo en el plano modal).
4. Determinar la estabilidad del sistema no lineal en torno al punto de equilibrio. Justificar la respuesta.
5. Analizar si es posible dibujar el retrato de fases aproximado del sistema linealizado en las cercanías del punto de equilibrio.
6. Repetir todo el problema para la entrada constante $I(t) = \frac{2}{3}$.
7. Simular el sistema y verificar los resultados.

[P8.7] Sistema con elasticidad y fricción generalizadas El sistema

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -g_1(x_1) - g_2(x_2) + u(t) \end{aligned} \tag{P8.7a}$$

representa un sistema masa-resorte-amortiguador con características generalizadas de elasticidad y fricción, dadas por las funciones arbitrarias $g_1(\cdot)$ y $g_2(\cdot)$ que están definidas en el primer y tercer cuadrante y que verifican $g_i(x_i) = 0 \iff x_i = 0$.

Se pide:

1. Para la entrada $u(t) = 0$, mostrar que el origen es el único punto de equilibrio.
2. Linealizar el sistema en torno al origen. Escribir los parámetros en función de las derivadas parciales $\partial g_i / \partial x_i$ evaluadas en $x_i = 0$.
3. Determinar que condiciones deben tener las derivadas parciales $\partial g_i / \partial x_i$ para que se pueda garantizar estabilidad en base al método indirecto de Lyapunov.
4. Demostrar que el origen es estable independientemente de la condición anterior. Puede utilizarse como candidata de Lyapunov la función $V(x) = G_1(x_1) + 0.5 x_2^2$, donde

$$G_1(x_1) = \int_0^{x_1} g_1(z) dz$$

5. Demostrar además que el origen es asintóticamente estable.
6. Suponer ahora se cambia la entrada por una con valor constante $u(t) = \bar{u}$.
 - a) ¿Se puede garantizar que haya un único punto de equilibrio?
 - b) ¿Es posible garantizar la estabilidad del o de los puntos de equilibrio con la información brindada sobre las funciones $g_i(\cdot)$?
 - c) Proponga un par de condiciones adicionales sobre dichas funciones para garantizar la estabilidad asintótica del o de los puntos de equilibrio.

[P8.8] Motor con Excitación Independiente

Consideraremos un motor de corriente continua con excitación independiente como el de la Figura 8.13, en el cual los parámetros son los dados por la Tabla 8.1 mientras que la característica magnética de la excitación es la de la Figura 8.14

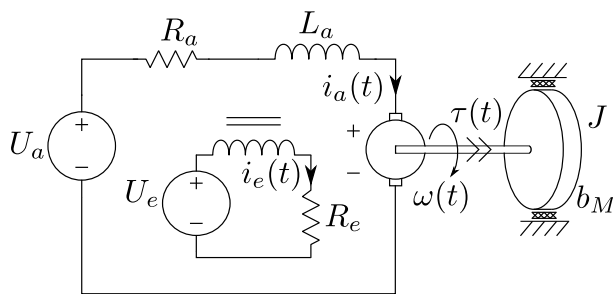


Figura 8.13: Motor de Corriente Continua con Excitación Independiente.

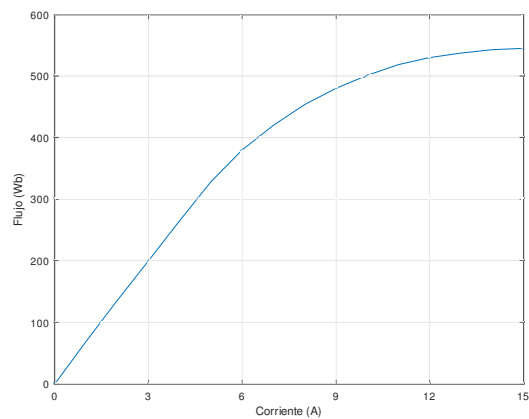


Figura 8.14: Característica Magnética de Excitación.

Parám.	Descripción	Valor	Unidad
J	Momento de Inercia	15	Kg m^2
b_M	Coef. Roce Eje	1.1	N m s
R_a	Resis. de Armadura	0.05	Ω
L_a	Induc. de Armadura	0.003	H
R_e	Resis. de Excitación	25.2	Ω
K_M	Constante Motor	0.016	V s / rad
U_{aN}	Tension Armadura Nom.	460	V
U_{eN}	Tension Excit. Nom.	184	V

Tabla 8.1: Parámetros del Motor de Corriente Continua

El modelo del motor se puede escribir mediante las siguientes ecuaciones de estado:

$$\begin{aligned}
 \dot{x}_1 &= (U_a(t) - R_a x_1 - K_M x_2 x_3)/L_a \\
 \dot{x}_2 &= U_e(t) - R_e g^{-1}(x_2) \\
 \dot{x}_3 &= (K_M x_2 x_1 - b_M x_3)/J
 \end{aligned}
 \tag{P8.8a}$$

donde la función $g^{-1}(\cdot)$ representa la inversa de la característica de campo de la Figura 8.14.

Se pide:

1. Interpretar físicamente que representan las variables de estado x_1 , x_2 y x_3 .
2. Calcular el punto de equilibrio para entradas $U_a(t) = U_{aN}$ y $U_e(t) = U_{eN}$.
3. Obtener un modelo linealizado en torno al punto de equilibrio encontrado en el punto anterior. Para estimar numéricamente el valor de $\partial g^{-1}(x_2)/\partial x_2|_{\bar{x}_2}$ puede aprovecharse la curva de la Figura 8.14.
4. De ser posible, estudiar la estabilidad del punto de equilibrio y la forma de las trayectorias en torno al mismo.
5. Suponiendo ahora entradas nulas, analizar la estabilidad del origen mediante el método directo de Lyapunov. Puede usarse como candidata de Lyapunov la energía total. En el caso del inductor no lineal la energía puede escribirse como

$$E(x_2) = \int g^{-1}(x_2(t)) \dot{x}_2(t) dt = \int g^{-1}(x_2) dx_2 = \Gamma(x_2)$$

es decir, la energía es la primitiva de $x_2(t)$.

6. ¿Para que rango de condiciones iniciales vale lo analizado en el ítem 5? ¿y lo analizado en el ítem 4?.
7. Este modelo no tiene en cuenta el ángulo del eje del motor. ¿Qué cambiaría si se lo considera?.

Bibliografía

- [Ack71] Ackoff, Russell L: *Towards a system of systems concepts*. Management science, 17(11):661–671, 1971.
- [BCCF19] Brauer, Fred, Carlos Castillo-Chavez y Zhilan Feng: *Mathematical models in epidemiology*. Springer, 2019.
- [Bor10] Borutzky, Wolfgang: *Bond graph methodology: development and analysis of multidisciplinary dynamic system models*. Springer Science & Business Media, 2010.
- [Cel91] Cellier, François E: *Continuous System Modeling*. Springer New York, 1991.
- [CK06] Cellier, François y Ernesto Kofman: *Continuous System Simulation*. Springer, New York, 2006.
- [CN05] Cellier, François E y Ángela Nebot: *The Modelica bond graph library*. En *Proc. 4th International Modelica Conference, Hamburg Germany*, 2005.
- [DB08] Dorf, Richard C. y Robert H. Bishop: *Modern Control Systems*. Pearson Education, Inc., 11th edición, 2008.
- [FAZ15] Fatoorehchi, Hooman, Hossein Abolghasemi y Reza Zarghami: *Analytical approximate solutions for a general nonlinear resistor–nonlinear capacitor circuit model*. Applied Mathematical Modelling, 39(19):6021–6031, 2015.
- [Fri14] Fritzson, Peter: *Principles of object-oriented modeling and simulation with Modelica 3.3: a cyber-physical approach*. John Wiley & Sons, 2014.
- [Fri15] Fritzson, Peter: *Introducción al modelado y simulación de sistemas técnicos y físicos con modelica*. Linköping University Electronic Press, 2015.
- [Gan10] Gandolfo, Giancarlo: *Economic dynamics: study edition*. Springer, 2010.
- [GPS01] Goldstein, Herbert, Charles Poole y John Safko: *Classical mechanics. Third Edition*, 2001.
- [HNW00] Hairer, Ernst, Syvert Nørsett y Gerhard Wanner: *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer, Berlin, Germany, 2nd edición, 2000.
- [HW91] Hairer, Ernst y Gerhard Wanner: *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*. Springer, Berlin, 1991.
- [JS09] Jeltsema, Dimitri y Jacquelin MA Scherpen: *Multidomain modeling of nonlinear networks and systems*. IEEE Control Systems Magazine, 29(4):28–59, 2009.
- [Jun15] Junco, Sergio: *Sistemas Dinámicos y Modelos Matemáticos*, 2015.
- [Jun20] Junco, Sergio: *Métodos Energéticos de la Mecánica Analítica en el Modelado de Sistemas Físicos en Ingeniería: Ecuaciones de Euler-Lagrange Ecuaciones de Hamilton y pHS*, 2020.
- [Kha15] Khalil, Hassan K: *Nonlinear control*, volumen 406. Pearson New York, 2015.

- [KMR12] Karnopp, Dean C, Donald L Margolis y Ronald C Rosenberg: *System dynamics: modeling, simulation, and control of mechatronic systems*. John Wiley & Sons, 2012.
- [Kof06] Kofman, Ernesto: *Simulación de Sistemas Continuos. Notas de Clase*. Departamento de Control, FCEIA, UNR. Disponible online en http://www.fceia.unr.edu.ar/control/ssc/notas_ssc.pdf, 2006.
- [Kot01] Kot, Mark: *Elements of mathematical ecology*. Cambridge University Press, 2001.
- [Kuo96] Kuo, Benjamin: *Sistemas de Control Automático. Séptima Edición*. Prentice Hall Hispanoamericana, México, 1996.
- [Lju98] Ljung, Lennart: *System identification*. Springer, 1998.
- [LTM⁺09] Lindberg, Erik, Elena Tamaseviciute, Gytis Mykolaitis, Skaidra Bumeliene, Tatjana Pyragiene, Arunas Tamasevicius y Raimundas Kirvaitis: *Autonomous third-order duffing-holmes type chaotic oscillator*. En *2009 European Conference on Circuit Theory and Design*, páginas 663–666. IEEE, 2009.
- [Mor08] Morin, David: *Introduction to classical mechanics: with problems and solutions*. Cambridge University Press, 2008.
- [OPNSR13] Ortega, Romeo, Julio Antonio Loría Perez, Per Johan Nicklasson y Hebertt J Sira-Ramirez: *Passivity-based control of Euler-Lagrange systems: mechanical, electrical and electromechanical applications*. Springer Science & Business Media, 2013.
- [Pan88] Pantelides, Constantinos C: *The consistent initialization of differential-algebraic systems*. *SIAM Journal on Scientific and Statistical Computing*, 9(2):213–231, 1988.
- [Per13] Perko, Lawrence: *Differential equations and dynamical systems*, volumen 7. Springer Science & Business Media, 2013.
- [PTVF07] Press, William H., Saul A. Teukolsky, William T. Vetterling y Brian P. Flannery: *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, New York, 3rd edición, 2007.
- [SB00] Seron, Maria Marta y Julio H Braslavsky: *Sistemas No lineales. Notas de Clase*. Departamento de Electrónica, Universidad Nacional de Rosario, 2000. <https://www.fceia.unr.edu.ar/control/snl/Apunte.pdf>.
- [Sky05] Skyttner, Lars: *General systems theory: Problems, perspectives, practice*. World scientific, 2005.
- [Tar72] Tarjan, Robert: *Depth-first search and linear graph algorithms*. *SIAM journal on computing*, 1(2):146–160, 1972.
- [Tho75] Thoma, Jean U: *Introduction to bond graphs and their applications*. Pergamon Press, 1975.
- [Til12] Tiller, Michael: *Introduction to physical modeling with Modelica*, volumen 615. Springer Science & Business Media, 2012.
- [Zim12] Zimmer, Dirk: *A planar mechanical library for teaching modelica*. En *Proceedings of the 9th International MODELICA Conference*, páginas 681–690. Linköping University Press, 2012.
- [ZMK18] Zeigler, Bernard P, Alexandre Muzy y Ernesto Kofman: *Theory of modeling and simulation: discrete event & iterative system computational foundations*. Academic press, 2018.