

Obligatorio 6 - Concepto de Estado y aplicación a los movimientos del “Bicho”

IIE - Facultad de Ingeniería - Universidad de la República

Tallerine Biónico 2025

El objetivo de este obligatorio es comprender el concepto de estado y aplicarlo a los movimientos del Bicho. De esta forma cada movimiento se dividirá en un conjunto de estados los cuales permitirán dar dinamismo al “bicho” cuando se cambia de un movimiento a otro. Además, esto permitirá solucionar de forma sencilla los problemas de equilibrio.

1. Concepto de estado

Dado un movimiento cualquiera del “bicho” (ej. avanzar), un Estado queda determinado por la posición de las patas y hacia dónde se están moviendo. En general va a ser suficiente solo con la posición de las patas, pero puede suceder que por sí sola, la posición no determine un Estado. Veamos el siguiente ejemplo para aclarar.

Ejemplo:

Si se considera el movimiento de vaivén de una hamaca, este movimiento se puede dividir en la siguiente sucesión de posiciones:

1. Al centro, abajo, yendo hacia adelante
2. Adelante arriba
3. Al centro, abajo, yendo hacia atrás
4. Atrás arriba

1. Al centro, abajo, yendo hacia adelante
2. Adelante arriba
3.

Se puede representar el movimiento de la hamaca con el diagrama de la figura 1.

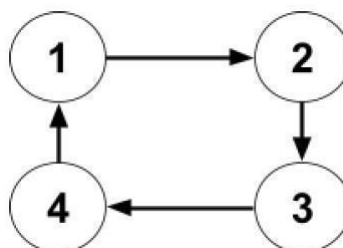


Figura 1: Diagrama de estados del movimiento de vaivén de la hamaca

Observar que la posición 1 y 3 es la misma, pero es necesario diferenciarlas pues no basta con saber que la hamaca se encuentra “al centro y abajo” para saber cual es la siguiente posición, sino que se requiere saber cuál es su posición anterior o, lo que es lo mismo, en que dirección se dirige la hamaca.

Lo anterior no sucede con las posiciones 2 y 4, ya que dada una de ellas, la siguiente posición, será “al centro y abajo”, pero cuidado!!! que no es lo mismo ir a 1 o a 3 aunque sean la misma posición.

Estas 4 posiciones serán los estados del movimiento de vaivén que está representado en el diagrama de la figura 1.

Se consideran ahora 2 movimientos de la hamaca

1. Vaivén
2. Detenida

Consideraciones:

- Siempre se comienza a hamacar hacia adelante.
- Para detenerse basta con pasar por el centro y la hamaca permanecerá allí.

Se muestra en la figura 2 un diagrama de estados para cada movimiento donde cada estado indica:

- 0) Al centro, abajo, detenida
- 1) Al centro, abajo, yendo hacia adelante
- 2) Adelante arriba
- 3) Al centro, abajo, yendo hacia atrás
- 4) Atrás arriba

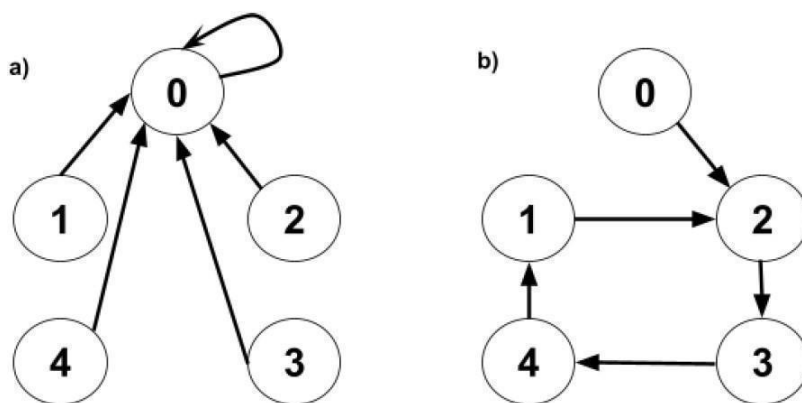


Figura 2: Diagrama de estados del movimiento: a) “Detenida” b) “Vaivén”

A partir de la figura 2 se observa que dado el estado actual y un movimiento, queda determinado el siguiente estado. Cada movimiento se resume en una sucesión de estados. Se podría implementar el movimiento de la hamaca con un servo.

Se considera para ello que las posiciones de la hamaca se corresponden con los siguientes ángulos del servo.

- Adelante arriba → 30°
- Al centro → 90°
- Atrás arriba → 150°

y se reciben los siguientes comandos vía bluetooth.

- Vaivén → "a"
- Detenido → "b"

El siguiente código pretende representar el movimiento de la hamaca utilizando los estados definidos anteriormente.

```
#include <stdint.h>
#include "udriver_pca9685_fing.h"
using namespace UDriver_PCA9685;

///Servos
const I2CAddress ADDR = 0x6A;
PCA9685ServoController servo_ctrl(ADDR);

#include <Adafruit_Microbit.h>

Adafruit_Microbit microbit; //defino objeto para el manejo de Bluetooth

// Variables globales
const pin_t PIN_HAMACA = SV1; // Pin para el control del servo
char comando;
int estado; // En esta variable se guarda el estado

void setup()
{
    Serial.begin(9600);

    // Inicializo servos
    servo_ctrl.begin();

    // se pueden agregar mas carateristicas personalizadas
    microbit.BTLESerial.setLocalName("microbit");
    microbit.BTLESerial.begin();

    estado = 0; // Se inicializa variable de estado
    comando = 'b'; // Se impone un comando inicial.
}

void loop(){
    char static dato;
    microbit.BTLESerial.poll();
    if (microbit.BTLESerial.available()) { //Hay un nuevo dato disponible
Bluetooth microbit
        dato=microbit.BTLESerial.read();
        if (dato!='\n'){comando = dato;}
    }
    /*Ejecuta la función correspondiente según la letra enviada desde el
control BT.*/
```

```

switch (comando) {
case 'a':
    Vaiven();
    break;

case 'b':
    Detenida();
    break;

default:
    delay(100);
    break;    // otro comando... la hamaca queda quieta.
}
}

void Detenida(){
    estado = 0; // siempre el siguiente estado será el "0"
    servo_ctrl.move_servo(PIN_HAMACA, 90); // hay un sólo estado. No se
    requiere switch
    delay(150);
}

void Vaiven(){
    switch (estado){ // Este switch es solo para el próximo estado
    case 0:
        estado = 2;
        break;
    case 1:
        estado = 2;
        break;
    case 2:
        estado = 3;
        break;
    case 3:
        estado = 4;
        break;
    case 4:
        estado = 1;
        break;
    default: // Si surgen otros estados no considerados....
        estado = 0;
        break;
    }
    switch (estado){ // Este switch es solo para la posición del servo
    case 0:
        servo_ctrl.move_servo(PIN_HAMACA, 90); //Hamaca al centro
        delay(150);
        break;
    case 1:
        servo_ctrl.move_servo(PIN_HAMACA, 90); //Hamaca al centro
        delay(150);
        break;
    case 2:
        servo_ctrl.move_servo(PIN_HAMACA, 30); //Hamaca arriba adelante
        delay(150);
        break;
    case 3:
        servo_ctrl.move_servo(PIN_HAMACA, 90); //Hamaca al centro
        delay(150);
        break;
    case 4:

```

```

servo_ctrl.move_servo(PIN_HAMACA, 150); //Hamaca arriba atrás
delay(150);
break;
default: // Si surgen otros estados no considerados....
servo_ctrl.move_servo(PIN_HAMACA, 90); //Hamaca al centro
delay(150);
break;
}
}

```

Sketch 1: Ejemplo Hamaca con estados

Importante: Al finalizar la rutina del movimiento (detenida o vaivén en este caso) la posición de las patas deben corresponderse con el estado. Para esto se debe primero realizar el **switch** que cambia el estado y segundo el **switch** que realiza el movimiento para ir a la posición del nuevo estado

2. Ejercicio

- Considerando el movimiento **Avanzar()** ya implementado, fraccionarlo en una sucesión movimientos más pequeños de forma que cada uno de ellos corresponde a:
 - Un único movimiento de uno o varios servos
 - Un único retardo.

Cada posición, luego de estos movimientos más pequeños corresponde a un estado.

Indicar en un diagrama de estados similar al de la figura 2 la representación de **Avanzar()**.

- A partir del ejemplo de la hamaca y del diagrama de estados obtenido en el ejercicio anterior, modificar el código de **Avanzar()** de forma de utilizar estados.
- Modificar **Retroceder()**, **GiroHorario()**, **GiroAntiHorario()** y **Saludar()** de forma de utilizar estados.

Aclaración: Asignar números de estado diferentes para cada tipo de movimiento; no se deben reutilizar estados. Por ejemplo, si un estado en *avanzar* presenta la misma posición de patas y muslos que un estado en *retroceder*, ambos deben tener números de estado distintos.