

Comunicaciones Inalámbricas  
Notas del Curso  
Facultad de Ingeniería  
Universidad de la República

Pablo Belzarena y Federico Larroca

4 de octubre de 2023

©2017 Pablo Belzarena (belza [at] fing [dot] com [dot] uy) y Federico ‘Larroca’ La Rocca (flarroca [at] fing [dot] com [dot] uy).

Esta obra es publicada bajo una licencia Creative Commons Atribución-CompartirIgual 4.0 Internacional. Para leer una copia de esta licencia, visite <http://creativecommons.org/licenses/by-sa/4.0/> o envíe una carta a Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.



# Índice general

<b>1. Sincronización temporal</b>	<b>5</b>
1.1. Introducción . . . . .	5
1.2. Diagrama general de un sincronizador temporal en tiempo discreto . .	11
1.3. Estimación del error . . . . .	12
1.3.1. Introducción, tiempo continuo . . . . .	12
1.3.2. Tiempo Discreto . . . . .	13
1.3.3. Detección de error: Estimador de máxima verosimilitud . . . .	14
1.3.4. Detección de errores: otros estimadores . . . . .	15
1.3.5. Cálculo del estimador del error de máxima verosimilitud . . . .	17
1.4. Interpolación . . . . .	19
1.4.1. Introducción . . . . .	19
1.4.2. Interpolador polinómico . . . . .	20
1.4.3. Interpolador polifásico . . . . .	21
1.5. Control del interpolador . . . . .	22
1.5.1. Contador módulo-1 . . . . .	22
1.5.2. Algoritmo recursivo . . . . .	25



# Capítulo 1

## Sincronización temporal

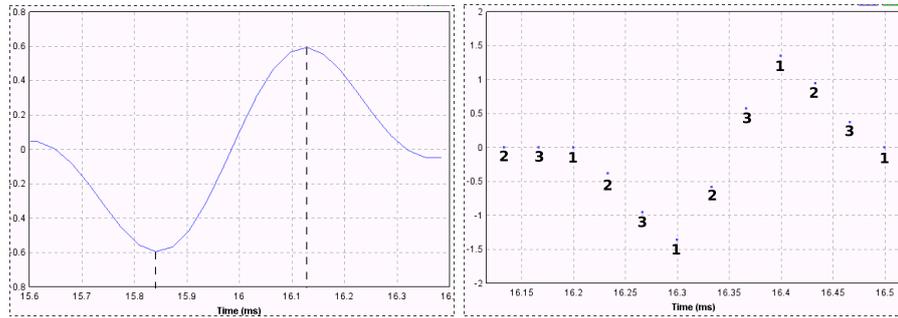
### 1.1. Introducción

La sincronización temporal consiste en alinear las escalas de tiempo de dos o más procesos que ocurren en puntos separados espacialmente. Esta es una de las funciones más críticas en un sistema de comunicación. El problema de sincronización del receptor digital es cómo obtener información precisa que le indique los instantes de muestreo óptimos de la señal recibida.

En los primeros sistemas de comunicación, la señal de información temporal necesaria para sincronizar transmisor y receptor era transmitida en un canal separado enviando una línea espectral a una frecuencia múltiplo de la señal temporal utilizada en los datos. Este método utilizado para sincronizar sistemas de comunicación, presenta diversas desventajas, entre otras, un uso ineficiente del ancho de banda y de la potencia del transmisor. En los sistemas de comunicación digital modernos la información temporal debe ser obtenida de la señal de datos propiamente dicha procurando minimizar alguna señal de error obtenida de la señal recibida y/o la señal muestreada.

El objetivo último del receptor es recuperar los símbolos enviados por el transmisor más allá del ruido, fading, errores diversos del receptor, etc.. Para llevar a cabo esta tarea, se sabe que se maximiza la relación señal a ruido si se obtiene una muestra del símbolo recibido donde se de el máximo del pulso a la salida del filtro apareado. En tiempo continuo esto implica muestrear el pulso a la salida del filtro apareado como se muestra en la figura 1.1a en el tiempo correspondiente a la línea punteada. Por lo tanto, en este caso, el problema es ubicar ese extremo relativo de la señal recibida y tomar la muestra en ese punto.

Sin embargo, en muchos receptores digitales modernos y particularmente en aquellos con una arquitectura SDR (Software Defined Radio), el muestreo de la señal en tiempo continuo se hace en lazo abierto y se obtienen una o más muestras por pulso como se muestra en la figura 1.1b. En este caso se tomaron tres muestras por símbolo y de acuerdo a la numeración de dicha figura, la muestra que debería utilizar el sistema es la muestra que tiene el número 1 de cada símbolo. Si se toma la muestra 1, se tendría para los pulsos muestreados de la figura referida cuatro valores. El primero correspondería a un 0, luego un -1, luego un 1 y luego nuevamente un 0. La primera



(a) Señal de tiempo continuo

(b) Señal en tiempo discreto

Figura 1.1: Señal en tiempo continuo y tiempo discreto

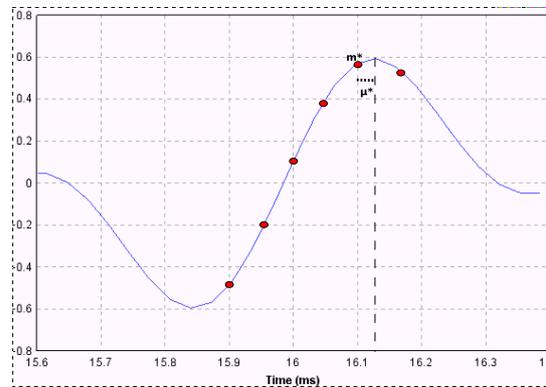


Figura 1.2:  $m^*$  y  $\mu^*$

pregunta de interés es la siguiente: cuál muestra debe tomarse de las tres de cada símbolo en este caso o de las  $N$  de cada símbolo en general?. Esa pregunta es una de las principales que se buscará contestar en este capítulo. Se notará con  $m^*$  a esa muestra.

Otro problema es que en general las muestras en el receptor no se encuentran en fase con las muestras en el transmisor. En la figura anterior sí lo estaban, ya que había una muestra en los máximos del pulso a la salida del filtro apareado. Por ejemplo en la figura 1.3a, se muestra la señal analógica y la señal muestreada en el receptor también con tres muestras por símbolo (la señal analógica y la muestreada tienen un pequeño desfase a los efectos de visualizarlas mejor). En este caso ninguna muestra se tomó exactamente en el máximo o mínimo de los pulsos por lo que para obtener el símbolo con la mayor inmunidad al ruido se debe interpolar. Por lo tanto, en este caso además de la pregunta anterior respecto a con cuál muestra quedarse surge otra interrogante: ¿A qué distancia está el máximo del pulso de la muestra elegida? Llamaremos a esta distancia  $\mu^*$ . Se muestran  $m^*$  y  $\mu^*$  en la figura 1.2.

No se debe olvidar que en general las señales con las que se trabaja son señales complejas. Por ejemplo, en la figura 1.3b se muestran la parte real e imaginaria de la señal que en las figuras anteriores se mostró solo la parte real. Se ha graficado uniendo

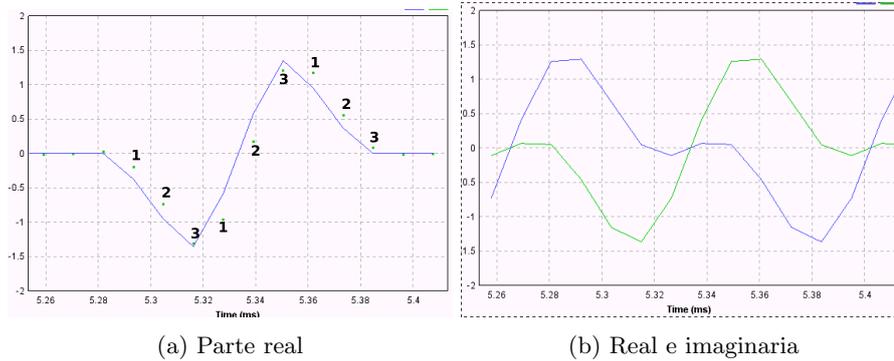


Figura 1.3: Señal discreta muestreada en el receptor

las muestras de cada señal con líneas a fin de que sea más fácil de visualizar.

Lo que importa en última instancia es recuperar los símbolos enviados. Se busca por tanto muestrear en el máximo del pulso a los efectos de tener mayor inmunidad al ruido.

Por ejemplo para la señal compleja de la figura 1.3b que tiene tres muestras por símbolo, dependiendo de qué muestra se utiliza para obtener los símbolos se obtienen las tres constelaciones que se muestran en las figuras 1.4a, 1.4b, 1.4c. En este caso se envió una constelación QPSK  $(-1, -j, 1, j)$ . La constelación que se podría decir que es la correcta más allá de un factor de ganancia es la que se muestra en la figura 1.4c. Sin embargo, si se observa con detalle esta constelación los puntos no se encuentran exactamente sobre los ejes porque no se está muestreando en los máximos del pulso conformador. Para optimizar el muestreo deberíamos realizar una interpolación y así obtener una constelación con menor error.

En las otras dos constelaciones claramente si existe ruido se puede rápidamente llegar a que no se pueda discriminar correctamente los símbolos.

En los ejemplos anteriores se envió una secuencia de símbolos constantes y que se repiten indefinidamente (en particular se envía 00,10,01,11 repetidamente). En las figuras 1.5a, 1.5b, 1.5c se muestra el mismo receptor tomando las mismas muestras para construir la constelación que en el caso anterior, pero ahora en lugar de enviar una secuencia fija de símbolos, los símbolos se envían aleatoriamente.

Como se puede ver en la figura 1.5b aún sin ruido (en estos casos estudiados el canal es totalmente ideal y sin ruido), el tomar la muestra incorrecta puede hacer que la decisión sobre el símbolo enviado pueda estar equivocada. Por otro lado, aún tomando la muestra con la que se obtiene la constelación más adecuada 1.5c existe una dispersión importante de puntos. Esto se puede mejorar si se interpola correctamente.

Un último problema que se debe abordar a los efectos de realizar una correcta sincronización es la diferencia entre las bases de tiempo. Es decir, el transmisor y el receptor tienen relojes diferentes y por tanto, la relación entre el tiempo de muestreo en el transmisor y el receptor en general no es un número entero. Eso lleva a que en el receptor la cantidad de muestras por símbolo no sea constante en todos los símbolos y en símbolos sucesivos no se muestree en los mismos puntos a cada símbolo. Este problema se puede observar en la figura 1.6b. En ella se puede ver que los símbolos no

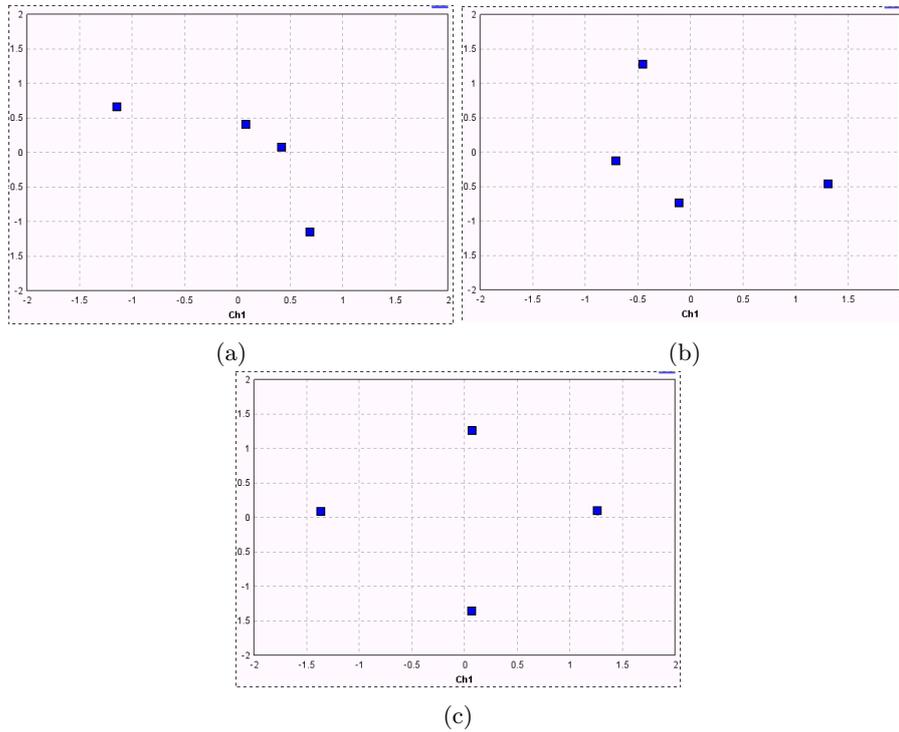


Figura 1.4: Constelaciones obtenidas utilizando diferentes muestras, secuencia fija de símbolos

se muestrean en los mismos puntos. Por ejemplo en los puntos indicados como A,B,C se puede apreciar que los puntos de muestreo se van desplazando debido a que el tiempo de muestreo no es un múltiplo entero del tiempo de símbolo. Esto lleva a que no solo se deba identificar la muestra a considerar,  $m^*$  y la distancia de esta muestra al máximo del pulso  $\mu^*$ . El problema en este caso es que esa distancia  $\mu^*$  varía con el tiempo y en algún momento la muestra a considerar  $m^*$  ya no es la adecuada porque se pasa del máximo y la adecuada pasa a ser la anterior.

Esta dinámica en la cual las muestras se van moviendo a lo largo del pulso se puede observar en la constelación que se obtiene si se toma cualquier muestra fija (1, 2, ó 3) para obtener el símbolo. Como esa muestra se mueve por los pulsos (tanto el correspondiente a la parte real como al de la parte imaginaria), el símbolo obtenido se va moviendo en el plano complejo. En la figura 1.6a se puede ver este movimiento. En este ejemplo se está enviando repetitivamente la misma secuencia de cuatro símbolos  $(-1, -j, j, 1)$  y por eso se observa ese patrón en el plano complejo.

**Ejercicio 1.1.** Se considera el diagrama en Gnuradio de la figura 1.7 cuyo código se encuentra en el material adjunto a este texto. Este diagrama incluye un transmisor, un receptor y un canal.

En el transmisor, se puede enviar desde una fuente de datos cuya secuencia se repite, utilizando un bloque Vector Source  $(0,1,2,3)$  o una secuencia de datos aleatoria

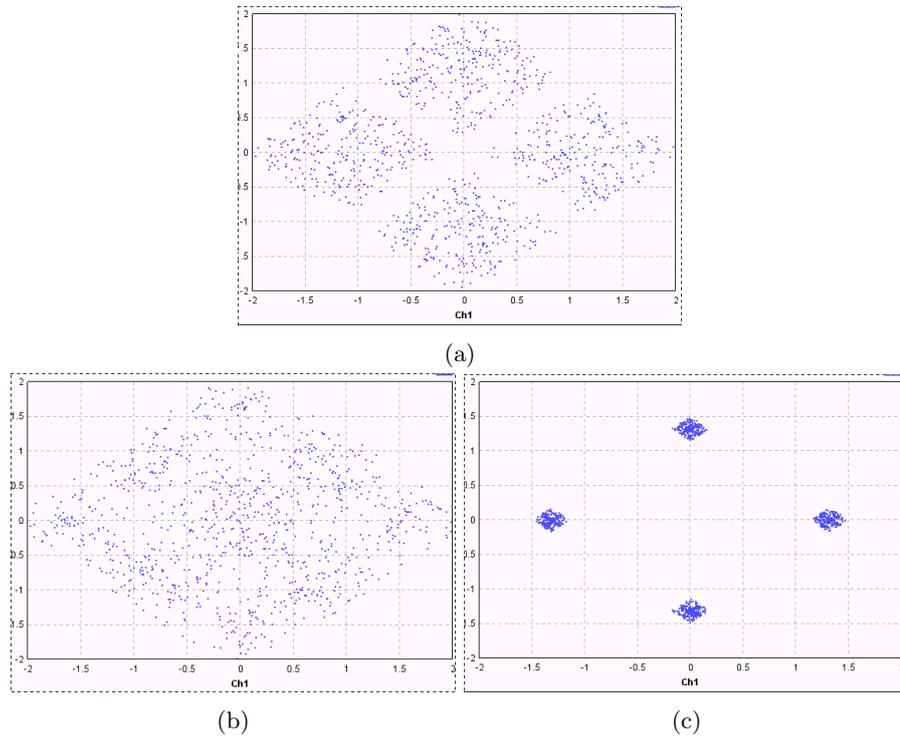


Figura 1.5: Constelaciones obtenidas utilizando diferentes muestras, secuencia aleatoria de símbolos

(bloque *Random Source*). Posteriormente se convierten estos datos a símbolos QPSK, se interpola para tener tres muestras por símbolo y se lo pasa por el pulso conformador RRC. Los dos principales parámetros que se modificarán en este ejercicio de dicho filtro RRC son: el exceso de ancho de banda:  $\alpha$  y la cantidad de símbolos por las que se extiende el pulso RRC ( $len\_sym\_src$ ).

El modelo de canal en este diagrama está compuesto por un bloque *Channel Model*, por un *Rational Resampler*, seguido por un bloque de retardo y un bloque "1 cada N". El *Rational Resampler*, está utilizado para generar 10 muestras interpoladas por cada muestra original. El bloque 1 cada N, vuelve a quedarse con la cantidad original de muestras y con el bloque de retardo se selecciona cuál de las muestras desfasadas de las muestras originales considerará el bloque 1 de N. Esto emula la conversión a tiempo continuo y el retardo desconocido del canal de una transmisión real. Del bloque *Channel Model* en este ejercicio el parámetro que se modificará será  $\epsilon$ . Este parámetro es el cociente entre las escalas de tiempo de transmisor y receptor. Si  $\epsilon$  es igual a uno, las escalas de tiempo están sincronizadas.

El receptor es muy simple, solo tiene el pulso apareado y luego permite seleccionar una de las tres muestras recibidas como "símbolo recibido". Para esto, se utiliza un bloque 1 de N. En las ventanas gráficas donde se muestra la señal recibida y la constelación correspondiente se puede cambiar el retardo del receptor para seleccionar una de las 3 muestras de cada símbolo. También se puede en la ventana gráfica cambiar

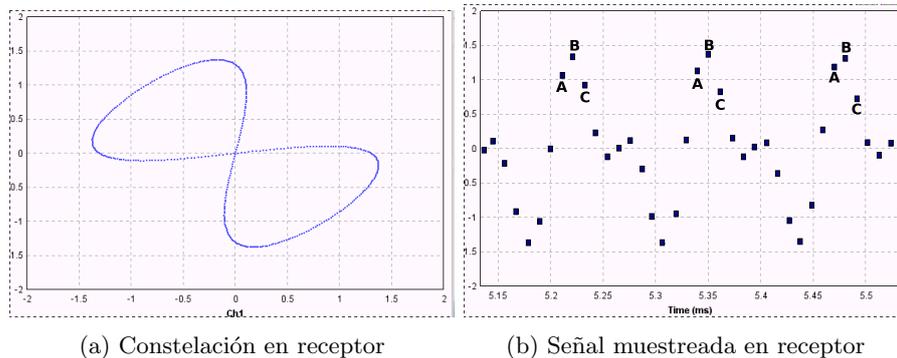


Figura 1.6: Bases de tiempo diferentes entre transmisor y receptor. Se envía secuencia repetitiva de símbolos.

el retardo del canal para emular diferentes retardos.

*Configuración inicial:* Inicialmente se trabajará con la siguiente configuración y luego se irán modificando algunos parámetros: Fuente de datos: Vector source. `samp_per_sym`: 3, `alfa`: 0.35, `len_sym_src`: 7. Los otros valores deben mantenerse como están configurados por defecto. El Channel Model inicialmente será de un canal ideal. El único parámetro que se modificará en la práctica será `epsilon`, que se tomará igual a 1 en la configuración inicial. Los demás parámetros del canal se mantendrán siempre en: sin ruido, `tap` = 1, sin offset de frecuencia.

a) Entienda bien el objetivo de cada bloque. Con la configuración inicial corra el programa. Manteniendo el retardo del canal en cero, observe las constelaciones que se obtienen si el retardo del receptor vale 0, 1 o 2. ¿por qué varía la constelación? ¿Hay alguna muestra que obtenga la constelación correcta? ¿por qué?.

Seleccione la muestra que considere se aproxima más a la constelación correcta. Ahora varíe el retardo del canal en 1,2,3, etc. Explique lo que observa. Indique la muestra y la parte fraccional que debería utilizar un interpolador para obtener la constelación correcta. ¿ qué sucede si el retardo del canal lo pone en 0 y 30? ¿por qué sucede esto?

b) Cambie ahora solamente de fuente de datos y utilice la Random Source. Vuelva a repetir la parte a) y explique por qué se dan las diferencias obtenidas. En particular explique por qué tomando la muestra y el retardo del canal óptimo no se obtiene un punto solamente como en caso a) sino un conjunto de puntos muy cercanos. Ahora observe que sucede si el parámetro `len_sym_src` se pone en 1, 3, 7 y 21. Explique lo que observa.

c) En las mismas condiciones que en b) repita para `len_sym_src` en 3 pero ahora además para cada caso cambie `alfa` en 0.15, 0.35 y 0.9. ¿ qué observa? explique lo que sucede. Recuerde una propiedad de la T. de Fourier que dice que cuanto más derivable es una función  $g(t)$  más rápido tiende a cero su T. de Fourier  $G(f)$  y lo mismo vale también en el otro sentido.

d) Como conclusiones de las partes anteriores qué debería hacer con `alfa` y el largo del pulso para minimizar los errores de sincronización temporal. Por otro lado, qué desventaja le ve a cambiar `alfa` y el largo del pulso en la dirección que es óptima para

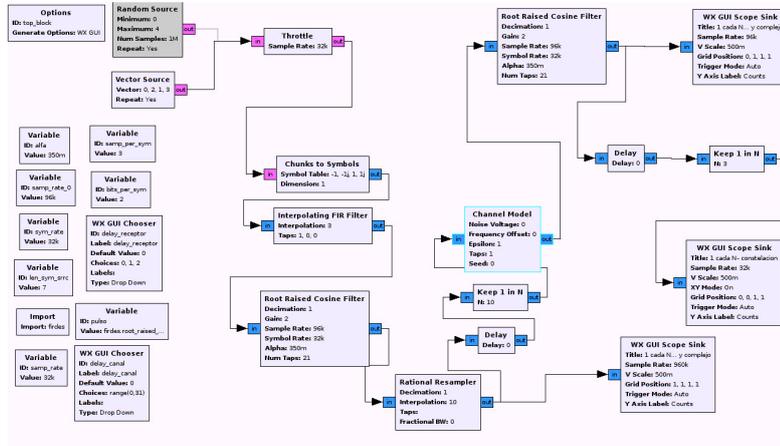


Figura 1.7: Ejercicio 1

la sincronización temporal.

e) Ahora se volverá a la configuración inicial y sólo se modificará el parámetro  $\epsilon$  del canal a 1.01. Observe como queda la constelación y explique por qué sucede esto. ¿se modifica la constelación si cambia el retardo del receptor? ¿por qué? Utilice ahora la fuente Random. Observe que sucede. ¿es posible sincronizar eligiendo alguna muestra? ¿qué habría que hacer?

## 1.2. Diagrama general de un sincronizador temporal en tiempo discreto

En esta sección se presentará el esquema general de un sistema de sincronización temporal. En las secciones siguientes se analizará cada uno de los bloques en particular.

El esquema general se muestra en la figura 1.8. Se ilustra este esquema con un receptor PAM en banda base. Más adelante se verán modificaciones necesarias para el caso QPSK. Hay que resaltar que en este capítulo se asume una sincronización perfecta en frecuencia y fase. Por tanto, el único problema a resolver es la sincronización en el tiempo.

En el diagrama 1.8 luego de pasar la señal por el filtro apareado, se debe seleccionar con qué muestra quedarse ( $m^*$ ) y cuál es el desplazamiento del máximo del pulso respecto de esta muestra ( $\mu^*$ ) a los efectos de interpolar correctamente. Para obtener estos dos datos habitualmente se utiliza un esquema realimentado tipo PLL. A partir de la muestra obtenida por el interpolador, se estima el error que se está cometiendo

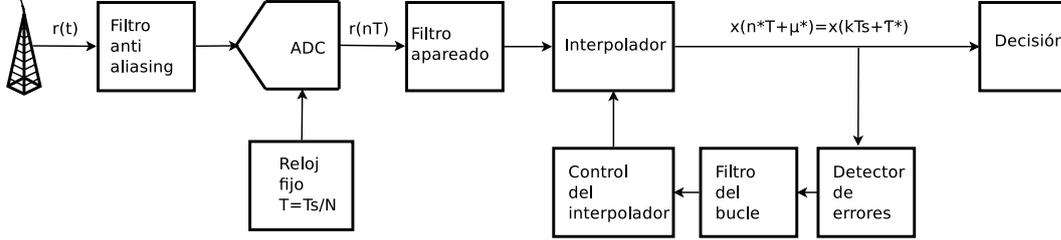


Figura 1.8: Diagrama de bloques sincronizador temporal

respecto de la muestra óptima y ese error se filtra y se introduce en un bloque de control que le informa al interpolador  $\hat{m}$  y  $\hat{\mu}$ , los estimadores de  $m^*$  y  $\mu^*$ .

### 1.3. Estimación del error

#### 1.3.1. Introducción, tiempo continuo

Para una mejor comprensión del problema en tiempo discreto, en esta introducción primero se formula el problema en tiempo continuo porque ayudará al lector a ver de forma más intuitiva lo que se busca. Se considera en este análisis el caso PAM bandabase y luego se extenderá el mismo para otros escenarios. La señal que se recibe en la antena del receptor se llamará  $r(t)$  y se puede escribir de la siguiente forma

$$r(t) = G_a \sum_m a[m]p(t - mT_s - \tau) + w(t)$$

donde  $G_a$  es el acumulado de ganancias y pérdidas del transmisor al receptor incluyendo antenas, medio de comunicación, etc..  $a[k]$  es el  $k$ -ésimo símbolo PAM enviado,  $T_s$  es el tiempo de símbolo,  $p(t)$  es el pulso conformador con soporte en  $-L_p T_s \leq t \leq L_p T_s$ ,  $\tau$  es el retardo variable y desconocido que existe entre transmisor y receptor y  $w(t)$  es ruido blanco aditivo y gaussiano.

La salida del filtro apareado se denominará  $x(t)$  y verifica la siguiente ecuación:

$$x(t) = G_a \sum_m a[m]r_p(t - mT_s - \tau) + \nu(t)$$

donde  $\nu(t) = w(t) * p(-t)$  y  $r_p(\cdot)$  es la convolución del pulso y el pulso apareado es decir:

$$r_p(u) = \int_{-L_p T_s}^{L_p T_s} p(t)p(t - u)dt \quad (1.1)$$

Por lo tanto, luego de muestrear la señal, se obtiene

$$x(kT_s + \hat{\tau}) = G_a a[k]r_p(\hat{\tau} - \tau) + G_a \sum_{m \neq k} a[m]r_p((k - m)T_s + \hat{\tau} - \tau)$$

$$x(kT_s + \hat{\tau}) = G_a a[k] r_p(-\tau_e) + G_a \sum_{m \neq k} a[m] r_p((k-m)T_s - \tau_e)$$

Donde  $\tau_e$  es el error de sincronización y en las últimas dos ecuaciones no se ha considerado la componente de ruido a los efectos de estudiar solamente el problema de sincronización. Observar que si el pulso conformador verifica la condición de Nyquist el segundo término es cero para  $\tau_e = 0$ .

Si  $\tau$  fuera estimado correctamente se sabe que  $kT_s + \tau$  debería coincidir con el punto donde el pulso conformador tiene un máximo. Es decir si  $\tau_e = 0$  entonces se estaría tomando las muestras en el receptor en el máximo del pulso (que coincide con el máximo de  $r_p$  por ser el pulso simétrico). Si se considera una señal PAM binaria si  $\tau_e = 0$  se estaría muestreando en un máximo o un mínimo de la señal de salida del filtro apareado como se muestra en la figura 1.1a y por lo tanto la derivada de la señal en el punto de muestreo sería cero. Si  $\tau_e$  no es cero entonces se estaría muestreando en un flanco de la señal y su derivada no sería cero. Se está sumiendo que hay un cambio de símbolo en este razonamiento. Si no hubiera cambio de símbolo la derivada podría ser cercana a cero aún cuando se estuviera muestreando en el lugar equivocado. Por este motivo, se asume que hay suficientes cambios de símbolos como para que el sincronizador pueda actuar. Muchas veces lo que se hace es “aleatorizar” la secuencia enviada para asegurar suficientes cambios de símbolos y evitar largas secuencias del mismo símbolo.

Por lo tanto, ya se tiene una estimación del error: la derivada de la señal  $x(t)$  en el punto de muestreo. Ahora bien, no solo interesa el error sino también saber si se está muestreando antes o después del máximo a los efectos de mover el muestreo en la dirección correcta. Se está adelantado si el símbolo es positivo y la derivada es negativa o si el símbolo es negativo y la derivada es positiva. Por el contrario se está atrasado si el símbolo es positivo y la derivada es positiva o si el símbolo es negativo y la derivada es negativa. De donde el error se puede escribir como:

$$e[k] = a[k] \dot{x}(kT_s + \hat{\tau}[k])$$

Como en general el verdadero  $a[k]$  no se conoce, se utiliza la estimación del símbolo  $\hat{a}[k]$ , es decir:

$$e[k] = \hat{a}[k] \dot{x}(kT_s + \hat{\tau}(k))$$

Para el caso de PAM binario;

$$e[k] = \text{signo}(x(kT_s + \hat{\tau}[k])) \dot{x}(kT_s + \hat{\tau}[k])$$

Se resalta nuevamente que este estimador del error asume la existencia de suficientes transiciones de símbolos.

### 1.3.2. Tiempo Discreto

En esta sección se seguirá considerando un sistema PAM y luego se extenderán los resultados a otros tipos de sistemas. Si se considera el sistema de la figura 1.8, y utilizando la misma nomenclatura que en la sección anterior, se tiene que si se muestrea la señal recibida cada tiempo  $T$  :

$$r(nT) = G_a \sum_m a[m]p(nT - mT_s - \tau) + w(nT)$$

Se asumirá también que los símbolos no están correlacionados en el siguiente sentido:

$$\mathbf{E}[a[k]a[m]] = E_{prom}\delta(m - k)$$

donde  $E_{prom}$  es la energía promedio de los símbolos.

Luego esta señal discreta se pasa por un filtro  $p(-nT)$  y la salida del mismo es

$$x(nT) = \frac{G_a}{T} \sum_m a[m]r_p(nT - mT_s - \tau) + \nu(nT)$$

donde  $r_p(\cdot)$  corresponde al muestreo de la señal en tiempo continuo de la ecuación 1.1, por lo cual aparece el  $\frac{1}{T}$  en esta ecuación al aproximar la integral por la sumatoria.

En general, en tiempo discreto el bloque que estima el error producirá una señal de error cada tiempo de símbolo. Si se asume un interpolador que se queda con una muestra por símbolo, la salida del interpolador se puede expresar como:

$$x(kT_s + \hat{\tau}) = \frac{G_a}{T} \sum_m a[m]r_p((k - m)T_s + \hat{\tau} - \tau) + \nu(kT_s + \hat{\tau}) \quad (1.2)$$

La salida del bloque detector de errores  $e[k]$ , será una función de  $\tau_e = \tau - \hat{\tau}$ . Estas muestras variarán en el tiempo dependiendo de los símbolos y de la interpolación obtenida, por lo tanto es razonable caracterizarla por lo que se denomina curva- $S$   $g(\tau_e) = \mathbf{E}[e[k]]$

### 1.3.3. Detección de error: Estimador de máxima verosimilitud

El estimador de máxima verosimilitud utiliza la pendiente de la curva de transiciones entre símbolos para estimar el error. En la siguiente sección se verá porqué este estimador es de máxima verosimilitud. Esta sección se centrará en estudiar la curva- $S$  de este estimador.

$$e[k] = a[k]\dot{x}(kT_s + \hat{\tau}[k]) \quad (1.3)$$

Como en general el verdadero  $a[k]$  no se conoce, se utiliza el símbolo que se estima  $\hat{a}[k]$ , es decir:

$$e[k] = \hat{a}[k]\dot{x}(kT_s + \hat{\tau}[k]) \quad (1.4)$$

Para el caso de PAM binario;

$$e[k] = \text{signo}(x(kT_s + \hat{\tau}[k]))\dot{x}(kT_s + \hat{\tau}[k])$$

Se verá en primer lugar cómo calcular la curva- $S$  de este estimador y en segundo lugar como calcular la derivada necesaria para este estimador en tiempo discreto.

$$\begin{aligned}
 g(\tau_e) &= \mathbf{E}[e[k]] \\
 &= \mathbf{E}[a[k]\dot{x}(kT_s + \hat{\tau})] \\
 &= \mathbf{E}\left[a[k]\frac{G_a}{T}\sum_m a[m]\dot{r}_p((k-m)T_s - \tau_e)\right]
 \end{aligned}$$

En esta ecuación,  $\dot{r}_p(\cdot)$  corresponde al muestreo de la derivada de la señal en tiempo continuo de la ecuación 1.1 y se tuvo en cuenta que el ruido no está correlacionado con la señal. Además si se utiliza la hipótesis sobre la no correlación de la secuencia de símbolos enviados, el único término de la sumatoria que no se anula es el término en  $k$  y por lo tanto:

$$g(\tau_e) = \frac{G_a}{T} E_{prom} \dot{r}_p(-\tau_e) \quad (1.5)$$

En la figura 1.9 se muestra la curva-S para este estimador de error. Esta curva fue obtenida mediante simulaciones con GNU-Radio. Se muestran dos curvas. La curva en azul corresponde al estimador cuando se utiliza el verdadero  $a[k]$  según la ecuación 1.3. La curva en rojo es utilizando un estimador de  $a[k]$  según la ecuación 1.4. Como se observa en la figura la curva que utiliza el estimador de  $a[k]$  (en rojo) cuando el error es importante se aparta de la curva azul. Esto se debe a que en esta zona comienzan a aparecer muchos errores y por tanto se estima mal el valor de  $a[k]$ .

La pendiente de la curva de la figura 1.9 en  $\tau_e = 0$  es la ganancia de este bloque en el sistema realimentado e influye en el desempeño del control de la interpolación. Esta pendiente depende de la amplitud de la señal recibida, de la energía media de los símbolos y de la pendiente de  $\frac{dr_p}{dt}$  en  $\tau_e = 0$  según la ecuación 1.5. La ganancia de este bloque depende del pulso  $p(t)$  y de la señal recibida y por lo tanto es difícil sintonizar el bloque realimentado. Por esta razón, se recomienda utilizar un bloque control de ganancia que fije la amplitud de la señal en un punto determinado.

Para realizar este estimador es necesario emplear un derivador según las ecuaciones vistas. Para este fin se pueden usar los diferentes filtros derivadores vistos en el capítulo de filtros del curso. El esquema sería como el indicado en la figura 1.10. Sin embargo, también es posible juntar estos dos bloques en uno solo según las siguientes ecuaciones:

$$\begin{aligned}
 \dot{x}(nT) &= x(nT) * d(nT) \\
 &= (r(nT) * p(-nT)) * d(nT) \\
 &= r(nT) * (p(-nT) * d(nT)) \\
 &= r(nT) * \dot{p}(-nT)
 \end{aligned}$$

Es decir que en lugar de colocar el filtro apareado y luego el bloque derivador se puede utilizar directamente un bloque con la derivada del pulso apareado.

#### 1.3.4. Detección de errores: otros estimadores

Existen varios estimadores de error en el siguiente ejercicio se verán dos de ellos de uso habitual.

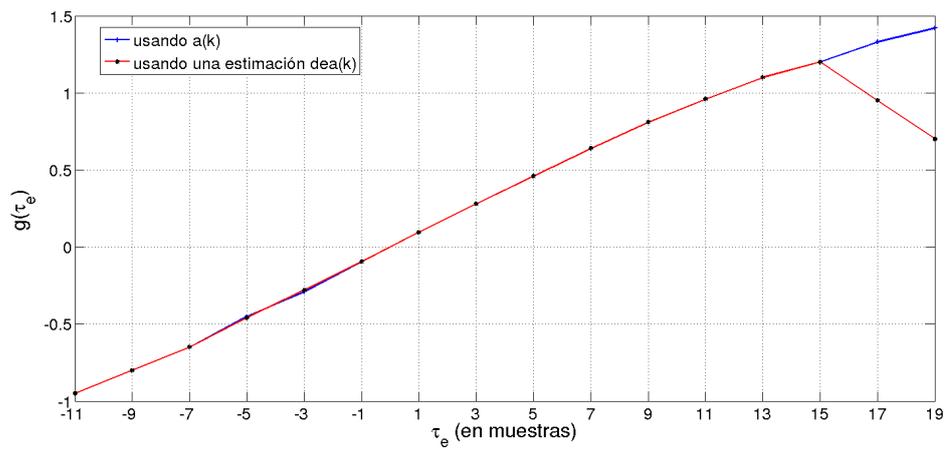


Figura 1.9: Cueva S estimador ML. Pulso SRRC 35% exceso de ancho de banda.  $N=20$  muestras por símbolo. La derivada se calcula como la diferencia entre la muestra y la anterior

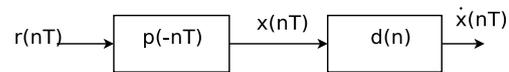


Figura 1.10: Derivador

**Ejercicio 1.2.** 1) El Detector de errores temporales “early-late” aproxima la derivada y por tanto el estimador de máxima verosimilitud y calcula el error según:

$$e[k] = a[k] (x(kT_s + \hat{\tau} + \Delta T_s) - x(kT_s + \hat{\tau} - \Delta T_s))$$

$\Delta$  depende de las muestras por símbolo que se tengan. Por ejemplo si se tuvieran dos muestras por símbolo  $\Delta = 0,5$ .

Calcular la curva  $S g(e) = \mathbb{E}\{e[k]\}$ .

2) El estimador de Muller & Mueller estima el error según la siguiente ecuación

$$e[k] = a[k-1]x(kT_s + \hat{\tau}) - a[k]x((k-1)T_s + \hat{\tau})$$

Calcular la curva  $S g(e) = \mathbb{E}\{e(k)\}$ . Se debería llegar a que es proporcional a  $r_p(T_s - \tau_e) - r_p(-T_s - \tau_e)$  Observar que  $r_p(t)$  es simétrica entorno a 0 ¿por qué?. Interpretar el resultado.

3) En el adjunto a este texto correspondiente al ejercicio 2 se entrega un estimador del error similar al early late. Explicar la funcionalidad de cada elemento del receptor, escribir las ecuaciones del estimador del diagrama entregado. Verificar que para la muestra en la cual se obtiene la constelación correcta se obtiene error cercano a cero. Graficar la curva  $S$  empíricamente. Utilizar la fuente vector y la aleatoria.

4) Opcional: Diseñar un estimador del error M&M y repetir lo solicitado en la parte 3 para este caso.

### 1.3.5. Cálculo del estimador del error de máxima verosimilitud

En las sección 1.3 se analizó como estimar el error a partir de un análisis cualitativo de la señal recibida. En esta sección veremos que ese análisis cualitativo brinda un estimador del error que tiene una interpretación formal cómo el estimador de máxima verosimilitud.

Para esto se retomará el análisis de los estimadores de máxima verosimilitud visto en capítulos precedentes. En esta sección se retomará el análisis del estimador de máxima verosimilitud en tiempo discreto. Se asumirá que la sincronización en frecuencia y fase es perfecta y que solo hay error de tiempo.

Retomando lo visto se tenía que el logaritmo del estimador de máxima verosimilitud se podía expresar a menos de una constante multiplicativa cómo:

$$\Lambda(r/\tau) = 2 \sum_i \text{Re} [r(iT)s^*(iT, \tau)] - \sum_i |s(iT, \tau)|^2$$

$$s(iT, \tau) = \sum_k a[k]p((iT - kT_s) - \tau)$$

donde se asumirá por el momento y a los efectos de simplificar los cálculos que se tiene un número infinito de muestras y que el pulso se extiende infinitamente. Luego haremos las consideraciones para acotar estas sumatorias.

De estas ecuaciones derivando respecto al parámetro y observando que  $s = a * p$  se llega a :

$$\frac{d\Lambda(r/\tau)}{d\tau} = \frac{d}{d\tau} \left\{ \sum_i 2\text{Re} \left[ r(iT) \sum_k a^*[k] p((iT - kT_s) - \tau) \right] - \sum_i \left| \sum_k a[k] p((iT - kT_s) - \tau) \right|^2 \right\}$$

Intercambiando las sumatorias,

$$\begin{aligned} \frac{d\Lambda(r/\tau)}{d\tau} &= \frac{d}{d\tau} \left\{ 2 \sum_k \text{Re} \left[ a^*[k] \sum_i r(iT) p((iT - kT_s) - \tau) \right] \right\} \\ &\quad - \frac{d}{d\tau} \left\{ \sum_i \sum_m \sum_n a[m] a^*[n] p((iT - mT_s) - \tau) p^*((iT - nT_s) - \tau) \right\} \\ &= \frac{d}{d\tau} \left\{ 2 \sum_k \text{Re} \left[ a^*[k] \sum_i r(iT) p((iT - kT_s) - \tau) \right] \right\} \\ &\quad - \frac{d}{d\tau} \left\{ \sum_m \sum_n a[m] a^*[n] \sum_i p((iT - mT_s) - \tau) p^*((iT - nT_s) - \tau) \right\} \end{aligned}$$

Observando la definición del pulso apareado y la salida de este ( $x = r * p_{\text{apareado}}$ ) y tomando la derivada se llega a:

$$\begin{aligned} \frac{d\Lambda(r/\tau)}{d\tau} &= 2 \sum_{k=0}^{L_0-1} \text{Re} [a^*[k] \dot{x}(kT_s + \tau)] \\ &\quad + 2 \sum_{m=0}^{L_0-1} \sum_{n=m-D}^{m+D} a[m] a^*[n] \dot{h}((n-m)T_s) \end{aligned} \quad (1.6)$$

donde  $x(\cdot)$  es la señal a la salida del filtro apareado e  $\dot{x}(\cdot)$  su derivada. Se define  $h(t) = p(t) * p(-t)$  y  $\dot{h}(t)$  es su derivada. En esta última ecuación se consideró también que se tiene un conjunto finito de muestras  $L_0$  y que el pulso es de duración finita.

Una consideración especial se debe hacer sobre el segundo término de la ecuación 1.6. Este término representa una suma de  $2D$  términos, donde se ha asumido que la respuesta del filtro es despreciable o nula a partir de la muestra número  $D$ . Este término habitualmente se desprecia en el estimador de máxima verosimilitud que se encuentra en la literatura sobre el tema. Algunos autores este término lo desprecian porque proviene de  $\sum_i |s(iT, \tau)|^2$  que es la energía de la señal enviada y si el período de observación es suficientemente largo tiene poca dependencia de  $\tau$ . Se observa también que  $\dot{h}(\cdot)$  verifica que  $\dot{h}(0) = 0$  y  $\dot{h}(t) = -\dot{h}(-t)$ . De esta propiedad se aprecia que despreciar este término es similar a considerar  $D = 0$ . Por otro lado, cuando se calcula curva-S de este estimador si los símbolos son no correlacionados y como  $\dot{h}(0) = 0$  al tomar valor esperado la suma de los  $2D$  términos da cero y por tanto no influyen en el valor medio del error.

Por lo tanto de aquí en adelante se considerará una aproximación al estimador de máxima verosimilitud que desprecia este término.

$$\frac{\partial \Lambda(r/\tau)}{\partial \tau} = \sum_{k=0}^{L_0-1} \text{Re} [a[k]^* \dot{x}((kT) + \tau)]$$

La suma en los términos anteriores deben ser iguales a cero para obtener el estimador de máxima verosimilitud. Observar que si la cantidad de muestras es suficientemente grande por la Ley de los grandes números  $\sum e[k] \approx \sum \mathbf{E}[e[k]]$

Para llevar a cero suma anterior se trata de llevar a cero en media el error en un tiempo  $k$  definido como:

$$e_\tau[k] = \text{Re} [a[k]^* \dot{x}((kT_s) + \tau[k])]$$

Se analizará ahora con más detalle el error temporal.

$$\begin{aligned} e_\tau[k] &= \text{Re} [a[k]^* \dot{x}((kT_s) + \tau[k])] \\ &= \text{Re}[a[k]] \text{Re} [\dot{x}((kT_s) + \tau[k])] + \text{Im}[a[k]] \text{Im} [\dot{x}((kT_s) + \tau[k])] \end{aligned} \quad (1.7)$$

De esta ecuación se puede observar que en el caso PAM, donde los símbolos son reales, se obtiene:

$$e_\tau[k] = a[k] \dot{x}((kT_s) + \tau[k])$$

Esta ecuación coincide con la que se había obtenido anteriormente de forma más intuitiva en la ecuación 1.3.

Para el caso de señales con símbolos complejos como QPSK por ejemplo, de la ecuación 1.7 se observa que en este caso se deben sumar los errores de la parte real y la parte imaginaria y por tanto un posible diagrama de bloques para un sincronizador temporal discreto en el caso QPSK sería el que se muestra en la figura 1.11

## 1.4. Interpolación

### 1.4.1. Introducción

En el receptor, se tienen muestras de la señal  $x(t)$  cada intervalos de tiempo  $T$ . La muestra deseada no tiene por qué coincidir con ninguna de las muestras que se tienen. Es decir, el extremo de la señal puede darse en un punto  $t = kT_I$  que no tiene porqué coincidir con ninguna de las muestras que están tomadas cada  $T$ . El punto óptimo, estará entre dos de las muestras que se tienen en  $nT$  y  $(n + 1)T$ . Lo que se busca es interpolar la señal  $x(t)$  entre estos dos puntos. A estos puntos se los denominará  $m(k)$  y  $m(k) + 1$  ya que corresponden a la muestra que obtendremos del  $k$ -ésimo pulso en  $kT_I$ . La distancia entre el punto  $m(k)T$  y  $kT_I$  es una fracción de  $T$  y la denominaremos  $\mu(k)T$  con  $0 \leq \mu(k) \leq 1$ . El problema del bloque de control del interpolador es suministrarle al interpolador  $m(k)$  y  $\mu(k)$ . El problema del interpolador es conocidos  $m(k)$  y  $\mu(k)$  obtener  $x(m(k)T + \mu(k)T)$  a partir de las muestras que se tienen de la señal.

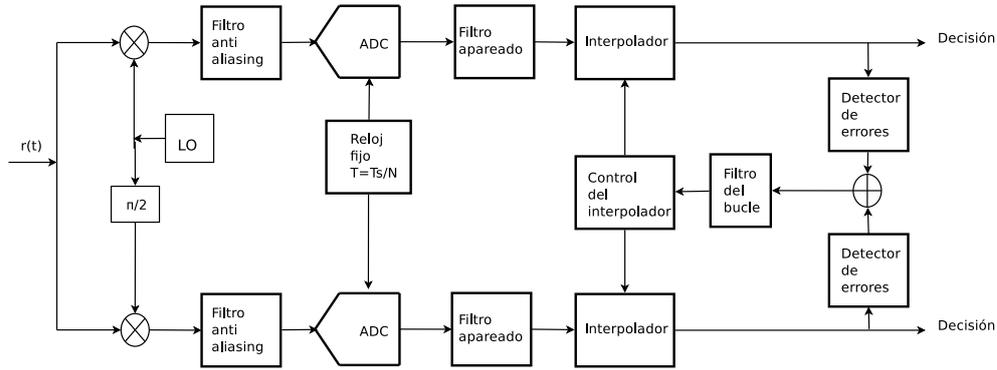


Figura 1.11: Esquema sincronizador QPSK

### 1.4.2. Interpolador polinómico

Este es el interpolador más simple y la idea es aproximar  $x(t)$  entre los puntos  $m(k)T$  y  $(m(k) + 1)T$  por un polinomio. En esta sección se verá el interpolador de primer orden o lineal y quedará como ejercicio ver alguno de los de mayor orden. El interpolador lineal aproximará  $x(t)$  por la recta  $a_1t + a_0$  entre las muestras correspondientes a  $m(k)$  y  $m(k) + 1$ . Por lo tanto se verifica que:

$$x(m(k)T) = a_1m(k)T + a_0 \quad (1.8)$$

$$x((m(k) + 1)T) = a_1(m(k) + 1)T + a_0 \quad (1.9)$$

$$x((m(k) + 1)T) - x(m(k)T) = a_1 \quad (1.10)$$

Además en el punto de interés se debe cumplir que:

$$x((m(k) + \mu(k))T) = a_1((m(k) + \mu(k))T) + a_0$$

A partir de esta última ecuación y utilizando las ecuaciones 1.8 y 1.10 para sustituir  $a_0$  y  $a_1$  y operando se obtiene:

$$x((m(k) + \mu(k))T) = x((m(k) + 1)T)\mu(k) + x(m(k)T)(1 - \mu(k))$$

Es decir que se puede obtener la estimación de la señal en el punto deseado a partir de las muestras en  $m(k)$  y  $m(k) + 1$  y del valor  $\mu(k)$  que suministre el bloque de control del interpolador.

**Ejercicio 1.3.** Observar que el interpolador anterior se puede escribir como un filtro FIR de la forma:

$$x((m(k) + \mu(k))T) = \sum_{i=-l_1}^{l_2} x((m(k) - i)T)h(i) \quad (1.11)$$

donde los  $h(i)$  dependen solo de  $\mu(k)$ . Utilizando esta expresión dibujar el filtro interpolador correspondiente para el interpolador lineal de primer orden.

### 1.4.3. Interpolador polifásico

Es común utilizar en sistemas de tiempo discreto un interpolador polifásico por su eficiencia y desempeño. En esta sección se analizará el fundamento de dicho interpolador.

El teorema de muestreo dice que la señal  $x(t)$  se puede representar:

$$x(t) = \sum_k x(kT) \operatorname{sinc}\left(\frac{t - kT}{T}\right) \quad (1.12)$$

Por lo tanto de 1.12

$$x(t + \tau) = \sum_k x(kT) \operatorname{sinc}\left(\frac{t - kT + \tau}{T}\right)$$

si ahora se muestrea esta señal cada  $nT$ ,

$$x(nT + \tau) = \sum_k x(kT) \operatorname{sinc}\left(\frac{(n - k)T + \tau}{T}\right)$$

Esta última ecuación dice que las muestras de la señal desplazadas un tiempo  $\tau$  se pueden recuperar de las muestras de la señal sin desplazar pero utilizando un filtro desplazado  $\tau$ . Esto por lo tanto permite con un banco de filtros obtener las muestras de una señal con el desplazamiento  $\tau$  que se desee respecto del muestreo original de la señal. Este esquema se muestra en la figura 1.12.

La idea anterior es la que permite construir un interpolador polifásico adecuado a las necesidades de la sincronización temporal como se verá en las ecuaciones siguientes.

Se considera la señal  $x(nT)$  a la salida del filtro apareado y de la que se desea recuperar muestras desplazadas una cantidad arbitraria del muestreo original. En lugar de utilizar un filtro ideal como en el teorema de muestreo se utilizará un filtro pasa bajo con respuesta al impulso  $h(t)$ . Las muestras a obtener no tendrán un desplazamiento arbitrario sino que el intervalo de muestreo del receptor  $T$  se dividirá en  $Q$  partes iguales. Se podrá obtener la señal interpolada a intervalos de duración  $\frac{q}{Q}$ . Se identifica con una variable  $q = 0, 1, \dots, Q - 1$  a cada una de esas partes y se puede escribir:

$$\begin{aligned} x(nT + \frac{q}{Q}T) &= \sum_k x(kT) h\left((n - k)T + \frac{q}{Q}T\right) \\ &= \sum_i x((n - i)T) h\left(iT + \frac{q}{Q}T\right) \end{aligned}$$

con  $q = 0, \dots, Q - 1$

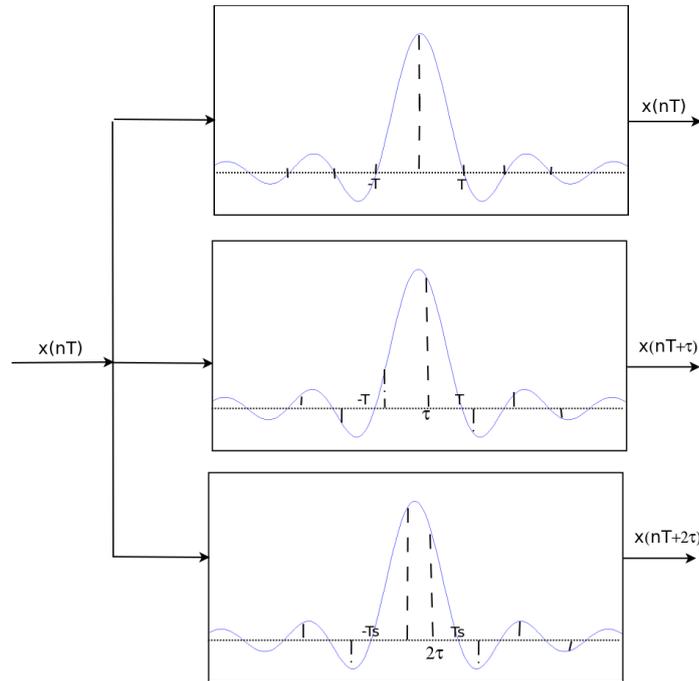


Figura 1.12: Filtro

Estas ecuaciones se pueden entonces representar por el diagrama de bloques de la figura 1.12, donde  $h_q(nT) = h(nT + \frac{q}{Q}T)$  con  $q = 0, \dots, Q - 1$ . Ahora bien, si se tiene en cuenta que el filtro apareado es también un filtro pasabajos, se puede utilizar este filtro en lugar del filtro  $h(\cdot)$  y resulta entonces el diagrama de bloques de la figura 1.14. El conmutador a la salida del banco de filtros elige de acuerdo al  $\mu$  que le indique el control del interpolador el filtro a utilizar para interpolar en el punto correcto.

## 1.5. Control del interpolador

En esta sección se analizará un algoritmo para definir cuál es la muestra del  $k$ -ésimo símbolo que se debe elegir del conjunto de muestras de  $x(nT)$ . Esta será la muestra denominada  $m(k)$  y el desplazamiento respecto de esta muestra para interpolar,  $\mu(k)$ .

### 1.5.1. Contador módulo-1

Existen diferentes formas de obtener los valores  $m(k)$  y  $\mu(k)$ . En primer lugar se analizará el esquema de la figura 1.15 donde el control es realizado por un contador descendente módulo-1.

Si no hay error, como se tienen  $N$  muestras por símbolo el contador módulo-1 dará underflow cada  $N$  muestras. El underflow se dará en realidad en la muestra  $m(k) + 1$  y el punto de cruce por cero de este oscilador determinará  $\mu(k)$ . Este oscilador juega el

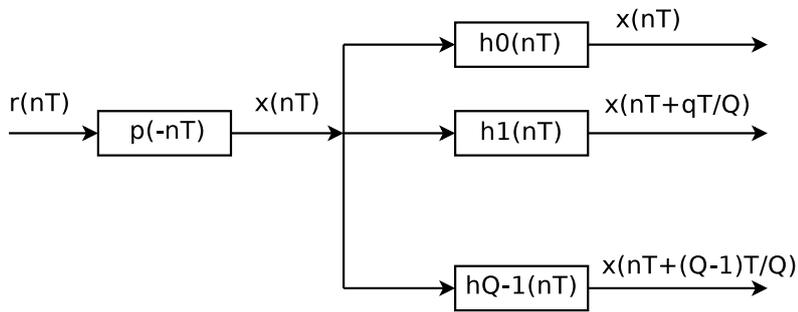


Figura 1.13: Filtro

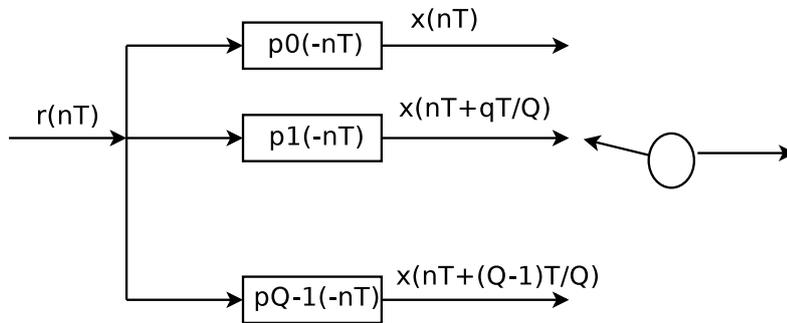


Figura 1.14: Implementación del Filtro Interpolador Polifásico

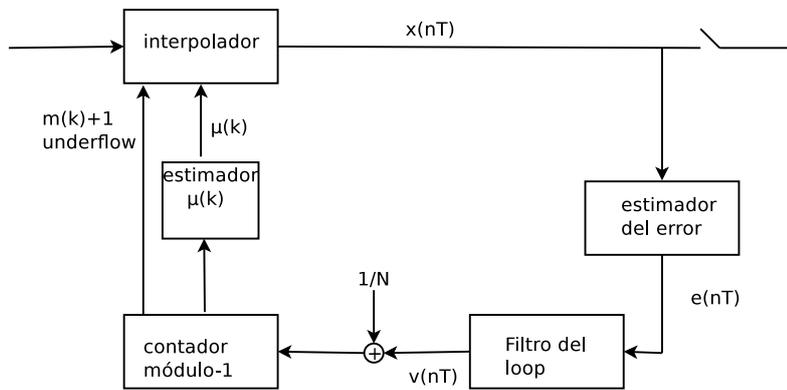


Figura 1.15: Implementación del control del interpolador contador módulo-1

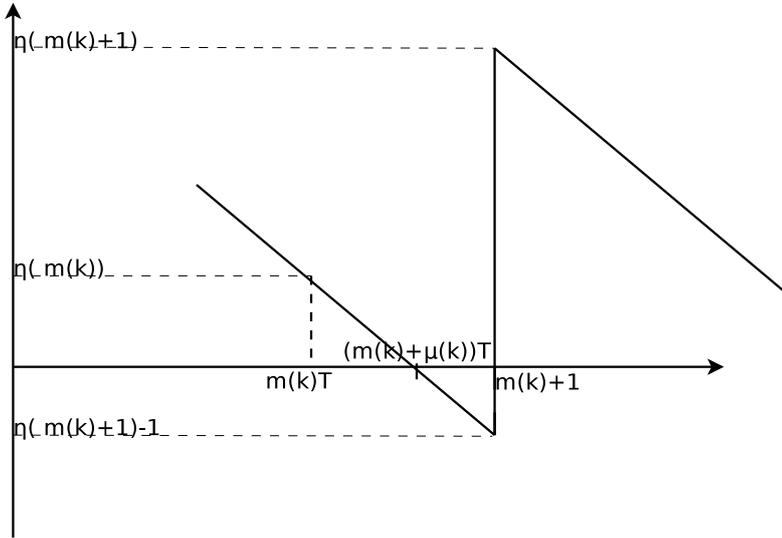


Figura 1.16: underflow contador módulo-1

rol del VCO (en los PLL de tiempo continuo), es decir será un oscilador de frecuencia media  $1/NT$  y cuya fase se modificará de acuerdo con el error filtrado  $v(nT)$ .

Como se muestra en la figura 1.16 donde se grafica la salida del contador en función del tiempo. El contador tendrá un underflow en  $(m(k) + 1)T$  donde se reseteará y le sumará uno a la salida para comenzar a decrecer nuevamente. Este underflow es la primera señal de control que recibirá el filtro interpolador para conocer el conjunto de muestras con las que debe interpolar. La segunda señal debe ser  $\mu(k)$  que se calcula de la siguiente forma.

Utilizando la nomenclatura del diagrama de la figura 1.15, se puede observar que la salida del contador módulo-1 ( $\eta$ ) verifica la siguiente ecuación:

$$\eta((n + 1)T) = \eta(nT) - \frac{1}{N} - v(nT)$$

Si el error es cero desciende con pendiente  $\frac{1}{NT}$ .

Un modelo linealizado como PLL de este sistema se puede observar en la figura 1.17.

Observar que en el símil con el PLL,  $K_0 = -1$  por ser un contador descendente y como vimos  $K_p = KE_{prom} > 0$ . En un lazo de PLL de segundo orden  $K_p K_0 K_1 = 2\xi w_n$  y  $K_p K_0 K_2 = w_n^2$ . De estas ecuaciones se puede deducir que  $K_1$  y  $K_2$  serán negativas. Por lo tanto si el error es positivo,  $v(nT)$  será negativo y la pendiente de descenso disminuye, atrasando así el instante de muestreo y si el error es negativo, la pendiente aumenta, adelantando así el instante de muestreo.

Si se denomina  $W(nT) = \frac{1}{N} + v(nT)$ , para los puntos  $m(k)T$  y  $(m(k) + 1)T$  se verifica:

$$\eta((m(k) + 1)T) = \eta(m(k)T) + 1 - W(m(k)T)$$

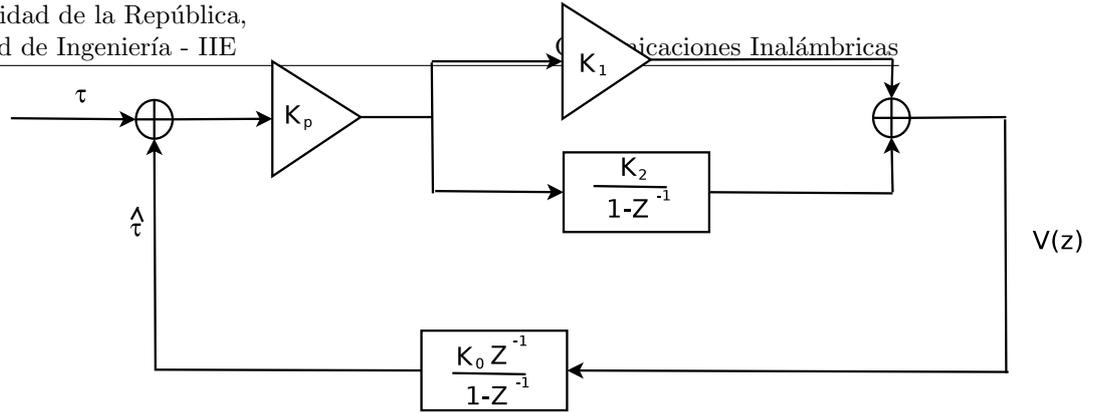


Figura 1.17: modelo de pll linealizado en tiempo discreto

Observar que  $W(nT)$  es una estimación de  $\frac{T}{T_I}$ , siendo  $T_I$  el tiempo entre dos máximos de dos símbolos consecutivos.

Según se puede observar en la figura 1.16 se verifica la siguiente relación entre los triángulos:

$$\frac{1 - \eta((m(k) + 1)T)}{\eta(m(k)T)} = \frac{1 - \mu(k)}{\mu(k)}$$

de donde,

$$\begin{aligned} \mu(m(k)) &= \frac{\eta(m(k)T)}{1 + \eta(m(k)T) - \eta((m(k) + 1)T)} \\ &= \frac{\eta(m(k)T)}{W(m(k)T)} \end{aligned}$$

Es decir que a partir del valor del contador en  $m(k)T$  y de  $W(m(k)T)$  es posible obtener el valor  $\mu(k)$  que necesita el filtro interpolador.

### 1.5.2. Algoritmo recursivo

Se verá a continuación otro método alternativo para controlar la interpolación. Si se denomina  $kT_I$  al tiempo en que se toma la muestra del  $k$ -ésimo símbolo, entonces se verifica que

$$\begin{aligned} kT_I &= (m(k) + \mu(k))T \\ (k + 1)T_I &= (m(k + 1) + \mu(k + 1))T \end{aligned}$$

de donde,

$$T_I = (m(k+1) + \mu(k+1))T - (m(k) + \mu(k))T \quad (1.13)$$

$$m(k+1) = m(k) + \frac{T_I}{T} + \mu(k) - \mu(k+1) \quad (1.14)$$

Como  $m(k)$  y  $m(k+1)$  son enteros los restantes términos deben tener parte fraccional nula y por lo tanto:

$$\mu(k+1) = \left( \frac{T_I}{T} + \mu(k) \right) \text{ módulo } 1$$

Esta ecuación nos permite calcular recursivamente el valor de  $\mu$  que se le debe dar al filtro interpolador. Por lo tanto  $m(k)$  se podrá calcular recursivamente utilizando la parte entera de los tres últimos términos de la ecuación 1.14

$$m(k+1) - m(k) = \left\lfloor \frac{T_I}{T} + \mu(k) - \mu(k+1) \right\rfloor$$

donde  $\frac{T_I}{T}$  se estima como  $W(kT) = \frac{1}{N} + v(kT)$ .

**Ejercicio 1.4.** Sean  $r(nT)$  la señal muestreada que ingresa al receptor digital,  $p(nT)$  la respuesta a impulso del filtro apareado y  $\dot{p}(nT)$  la de la derivada del filtro apareado. Escribir un diagrama completo de un sincronizador temporal que cumpla lo siguiente:

1. Utilice como interpolador el polifásico visto en clase.
2. Calcule el error usando alguno de los algoritmos vistos para tal fin.
3. Tenga un bloque proporcional e integral en la realimentación del error.
4. Calcule recursivamente  $m(k)$  y  $\mu(k)$ , es decir el momento donde debe muestrear.

**Ejercicio 1.5.** En este ejercicio se pide en primer lugar estudiar la documentación del sincronizador polifásico de gnuradio. Posteriormente implementar el transmisor QPSK y canal vistos en el ejercicio 1.1 y un receptor que utilice un sincronizador polifásico.

a) El sincronizador polifásico en gnuradio tiene 4 salidas. Explique qué es cada una de las 4 salidas e indique de dónde se obtienen en el diagrama de bloques que realizó en el ejercicio 1.4. Empiece por hacer funcionar el receptor cuando  $\epsilon = 1$  y un  $\text{retardo\_canal} = 0$ .

b) El parámetro crítico para sintonizar este bloque es el denominado: loop bandwidth. Explique qué representa y qué efecto tiene. En particular explique qué sucede si lo pone en un valor muy pequeño o muy grande. Le sugerimos comenzar con este parámetros en  $\pi/100$  y luego probar que sucede si lo cambia a  $10\pi$ ,  $\pi$  y  $\pi/100000$ . Pruebe cambiar el retardo del canal de 0 a 7 por ejemplo y observar la respuesta dinámica de la constelación y de la fase del bloque de sincronización.

Al cambiar el retardo del canal, observe que cambia en la salidas del bloque polifásico y la constelación. Explique lo que observe.

c) Para esta parte cambie la base de tiempo del receptor modificando  $\epsilon$  y colocándolo por ejemplo en 1.001. Observe los cambios en las 4 salidas del interpolador polifásico y explique lo que observa. Encuentre aproximadamente en qué rango de loop bandwidth logra sintonizar el receptor. Le sugerimos probar por ejemplo  $5\pi$ ,  $\pi$ ,  $\pi/10$ ,  $2\pi/50$ ,  $2\pi/100$ ,  $2\pi/1000$  y  $2\pi/2000$ .

d) Por último, cambie el parámetro  $\epsilon$  a 1.009 y analice si el sincronizador funciona correctamente o no en  $5\pi$ ,  $\pi$ ,  $\pi/10$ ,  $2\pi/20$  y  $2\pi/50$ .

Explique por qué a medida que la diferencia entre las bases de tiempo es mayor el rango de sincronización del PLL del sincronizador polifásico disminuye.



# Bibliografía

- [1] Digital Communication Barry, John R., Lee, Edward A., Messerschmitt, David G. Originally published under Leww,E.A.;Messerschmitt,D.G. 3rd ed. 2003, XVII, 838 p. In 2 volumes.
- [2] Proakis, J. and Salehi, M., Digital Communications, isbn=9780072957167, 2007, McGraw-Hill Education.
- [3] Digital Communications: A Discrete-Time Approach, Michael Rice, isbn=9780130304971, 2008, Pearson, Prentice Hall.
- [4] Signal Processing Techniques for Software Radios, Behrouz Farhang-Boroujeny, 2008, University of Utah.