

Simulación en MATLAB y Lenguaje de Programación C del Algoritmo de Gardner para el Sincronismo en el Tiempo de Señales Seriales en Telecomunicaciones

Simulation in MATLAB and C Programming Language Gardner Algorithm for the Time Sync Serial Signals in Telecommunications

Emmanuel Boutillon¹, Milton Ríos Julcapoma²; Franklin Rafael Cochachin Henostroza³

¹*Université de Bretagne – Sud, Francia*

^{2,3}*INICTEL - Universidad Nacional de Ingeniería, Lima, Perú*

Resumen— El presente trabajo realiza el estudio del Algoritmo de Gardner el cual es usado para sincronizar las señales en el dominio del tiempo y con esto poder recuperar correctamente la información. Su implementación es simple y sencilla debido a que requiere solo dos muestras de la señal por cada símbolo de datos. La simulación del algoritmo de Gardner se realizó en los lenguajes de programación MATLAB y C obteniéndose los resultados esperados, los cuales concuerdan con la teoría.

Abstract— This paper deals with the study of Gardner algorithm which is used to synchronize signals in the time domain and with this information can be successfully retrieved. Its implementation is simple and easy because it requires only two samples per data symbol. The simulation of Gardner algorithm was performed in both MATLAB and C language obtaining the expected results, which agree with the theory.

Palabras claves - Algoritmo Gardner, Convolución, Control Predictivo, PID, Laplace.

Key Words- Gardner algorithm, Convolution, Predictive Control, PID, Laplace.

I. INTRODUCCIÓN

En la sección II se expone los motivos que nos llevaron a la elección del presente proyecto, fundamentando y justificando el problema que se

quiere resolver y dando a conocer los alcances y limitaciones. Por otra parte en la sección III se plantean los objetivos que se quieren alcanzar al realizar el presente proyecto.

En la sección IV se presenta el marco teórico necesario para el estudio y comprensión del Algoritmo de Gardner, describiendo matemáticamente la obtención de este algoritmo.

En la sección V se da a conocer las consideraciones que se tomó para poder simular el Algoritmo de Gardner en los lenguajes de programación MATLAB y C; para el lenguaje programación C se usó el producto de software **Code::Blocks**. Así mismo, en la sección VI se presentan los resultados obtenidos y finalmente en la sección VII se da a conocer las conclusiones obtenidas.

Se debe saber que realizar operaciones matemáticas complejas como la convolución, son fáciles de efectuar en el lenguaje de programación MATLAB, debido a que MATLAB cuenta con funciones matemáticas ya implementadas, que se pueden llevar a cabo con simples comandos. Sin embargo, en el lenguaje de programación C, se deben crear los diferentes códigos, para que estos realicen las diferentes operaciones matemáticas.

II. PRESENTACIÓN DEL PROBLEMA

A. Descripción del problema

Necesidad de conocer los diversos algoritmos existentes para la sincronización de las señales

¹ Emmanuel Boutillon, e-mail: emmanuel@univ-ubs.fr

² Milton Ríos Julcapoma, e-mail: mríos@inictel-uni.edu.pe

³ Franklin R. Cochachin H., e-mail: fc Cochachin@inictel-uni.edu.pe

Recibido: Mayo 2014 / Aceptado: Junio 2014.

desfasadas en tiempo y en fase que llegan a un receptor.

B. Justificación del problema

El conocimiento adecuado de los diversos algoritmos para el sincronismo de las señales juega un papel importante en diversas aplicaciones como en la decodificación. La adecuada sincronización de las señales es crítica para recuperar de manera correcta la información original.

Es necesario conocer el funcionamiento de los algoritmos de sincronización existentes para poder plantear nuevos algoritmos de sincronización o usarlos en una aplicación.

En la comunicación satelital, la señal que llegue al receptor estará afectada por el ruido, el ruido hace que cambie la fase y amplitud de la señal, como consecuencia la señal no estará sincronizada a la entrada del receptor.

El estudio y desarrollo de nuevos decodificadores para el Standard DVB-S2 [1] para comunicación satelital hace uso de los algoritmos de sincronización de señales. El *LAB-STICC Y CNRS* [5], actualmente están haciendo estudios y pruebas para sincronizar las señales en el tiempo de forma paralela usando el Algoritmo de Gardner [2].

C. Alcances y delimitaciones

Con el adecuado conocimiento del Algoritmo de Gardner, se puede plantear una nueva arquitectura para el sincronismo de señales en el tiempo, donde la nueva arquitectura trabaje las señales de transmisión-recepción en forma paralela.

Este trabajo se enfoca en el estudio, codificación y simulación del Algoritmo de Gardner en los lenguajes de programación MATLAB y C; La señal a sincronizar es una señal que se transmite en forma serial y se recibe también de forma serial.

III. OBJETIVOS

Codificar y simular el algoritmo de Gardner en los lenguajes de programación MATLAB y C para el sincronismo en el tiempo de señales en una transmisión-recepción serial, así mismo dar a conocer algunos fundamentos básicos de este algoritmo aplicado a un sistema de comunicaciones propuesto por Gardner.

IV. MARCO TEÓRICO

Algoritmo de Gardner fue elaborado para el sincronismo de señales de banda base, señales BPSK y señales QPSK [2].

Para la recuperación del sincronismo de la señal, el Algoritmo de Gardner es bastante independiente de la fase de la portadora [2] (error de fase o de frecuencia que presenta la portadora); además para ser implementado requiere de solo dos muestras de la señal por cada símbolo de datos.

A. Modelo del receptor propuesto por Gardner

Para describir el funcionamiento del receptor propuesto por Gardner, mostrado en la Fig.1, debemos indicar primero que los símbolos son transmitidos de manera síncrona desde punto de transmisión al punto de recepción. Es decir existe un reloj del sistema que rige la comunicación, además cada símbolo ocupa un intervalo de tiempo T (*intervalo de símbolo*) dentro del tiempo que dura la transmisión.

La señal de entrada al receptor es demodulada en un par de señales de banda base (sin portadora). Estas señales de banda base están desfasadas en cuadratura, las cuales son llamadas señales I y Q, así mismo están presentes en los canales denominados canales I y Q, y la fase de la portadora local (*local carrier*) debe ser ajustada de acuerdo a la portadora que trae consigo la señal de entrada.

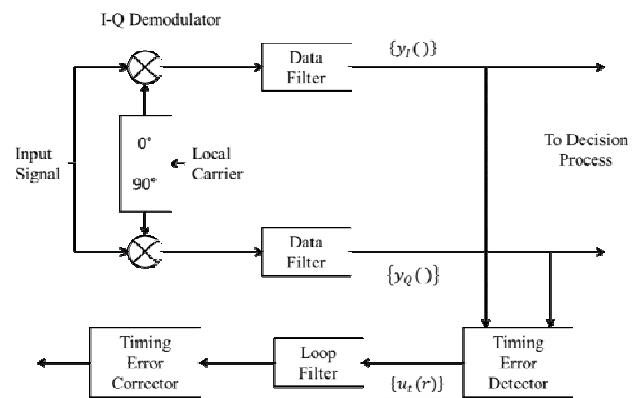


Fig.1. Diagrama de bloques simplificado de un modem típico [2].

A la salida de los mezcladores, las señales en los canales I y Q pasan por los filtros de datos (*data filter*) en donde se reduce al mínimo el ruido y se eliminan los productos no deseados que se generan en los mezcladores (productos de intermodulación); de esta manera se obtiene las señales reales $y_I(t)$ e $y_Q(t)$ para entrar al bloque Detector de Error de Temporización (*timing error detector*) en donde se verifica si la señal está sincronizada o no (en este bloque donde trabaja el algoritmo de Gardner), a la salida de este bloque se obtiene la señal de error de sincronización $u_e(t)$, esta secuencia de error pasa por un filtro de bucle (*Loop Filter*) que suaviza ésta secuencia de error llegando finalmente al bloque Corrector de Error de

Temporización (*timing error corrector*) en donde se corrige el error de sincronización ajustando el tiempo de muestreo.

En el algoritmo de Gardner, para cada señal (señal I y señal Q) se tomarán dos muestras por intervalo de símbolo y dichas muestras son coincidentes en el tiempo en cada secuencia I y Q (ver Fig.2). Una de las muestras se toma en el instante que se decide que hay inicio de un símbolo y la otra muestra se toma a la mitad del intervalo entre los inicios de dos símbolos consecutivos, tal como se muestra en la Fig.3. Con esto se toma información de tres puntos diferentes en dos símbolos consecutivos con el fin de producir un punto de error temporal (*one timing error point*).

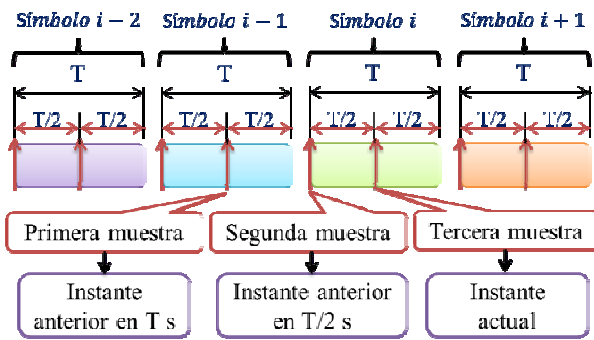


Fig.2. Símbolos y muestras tomadas en los instantes indicados que se usarán en el algoritmo de Gardner [2].

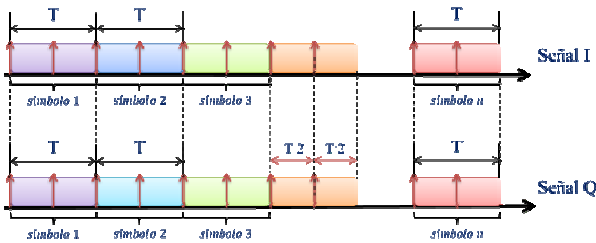


Fig.3. Secuencia de n símbolos de la señal I con la señal Q [2].

El algoritmo de Gardner calcula el error de sincronización $\{u_i(r)\}$ a partir de tres puntos de muestreo diferentes, éste error calculado puede tener el valor cero, lo cual indica que los datos que se reciben están sincronizados; pero si el error calculado no es cero, el algoritmo ajusta el tiempo de toma de los tres puntos, esto quiere decir que tomará tres nuevos puntos atrasando o adelantando el tiempo en T/2 para la toma de nuevas muestras, teniendo un nuevo inicio, en este nuevo inicio se toma otra vez los tres puntos y se calcula nuevamente el error de sincronización (este nuevo error calculado no es necesariamente igual al error de sincronización hallado anteriormente); el atraso o adelanto en T/2 del tiempo se dará hasta que el

error de sincronización calculado sea cero, teniendo en ese momento los datos sincronizados.

El algoritmo de Gardner al necesitar la información de tres puntos de muestreo diferentes (una muestra en el instante actual tomada, luego dos muestras tomadas en los instantes T/2 s y T s antes respectivamente) necesita almacenar estos datos en una memoria, por lo que decimos que el algoritmo posee memoria. Estas características se deben tomar en cuenta en el análisis del algoritmo y en su implementación.

B. Descripción del Algoritmo

Los símbolos se transmiten de manera síncrona, cada símbolo ocupa un intervalo de tiempo T (*intervalo de símbolo*). Por cada símbolo se tomará dos muestras y las muestras tienen que ser coincidentes en el tiempo en cada señal I y Q.

Se tiene a $x(t)$ como la señal (I o Q) en el instante t. Luego, haciendo la diferencia de la señal en los instantes t y $t - T/2$, se tiene:

$$x_d(t) = x(t) - x(t - T/2) \tag{1}$$

$$x_d^2(t) = x^2(t) + x^2(t - T/2) - 2x(t)x(t - T/2) \tag{2}$$

Las muestras son $E(r)$ y $L(r)$, el índice r se usa para designar el número de símbolo.

Tras un muestreo en $t = rT + \tau$ se tiene:

$$E(r) = x_d^2(\tau + rT) \tag{3}$$

$$E(r) = x^2(\tau + rT) + x^2(\tau + (r - 1/2)T) - 2x(\tau + rT)x(\tau + (r - 1/2)T) \tag{4}$$

$$L(r - 1) = x^2(\tau + (r - 1/2)T) + x^2(\tau + (r - 1)T) - 2x(\tau + (r - 1/2)T)x(\tau + (r - 1)T) \tag{5}$$

Se define $u_i(r)$ como el error.

$$u_i(r) = E(r) - L(r - 1) \tag{6}$$

Si a la ecuación anterior se le cambia de signo, el cambio de signo no tiene significado en las manipulaciones formales o en la carga computacional del procesador, pero nos asegura una pendiente negativa en el punto de seguimiento de la salida del detector [2], por lo tanto se puede reescribir como:

$$u_i(r) = L(r-1) - E(r) \quad (7)$$

Reemplazando los valores de $L(r-1)$ y $E(r)$ tenemos:

$$u(r) = x^2(\tau + (r-1)T) - x^2(\tau + rT) + 2x(\tau + (r-1/2)T) \left\{ \begin{array}{l} x(\tau + rT) - \\ x(\tau + (r-1)T) \end{array} \right\} \quad (8)$$

Ahora vemos la potencia útil del algoritmo, esta potencia es la media de muchas muestras, no es solamente el valor de una muestra aislada [2].

Sea $Avg\{u_i(\tau + (r-1)T)\}$ como la media de muchas muestras, luego:

$$U_i(\tau) = Avg\{x^2(\tau + (r-1)T)\} - Avg\{x^2(\tau + rT)\} + 2Avg\left\{x(\tau + (r-1/2)T) \left[\begin{array}{l} x(\tau + rT) - \\ x(\tau + (r-1)T) \end{array} \right] \right\} \quad (9)$$

De la ecuación anterior, los dos primeros términos son iguales, la media del conjunto no puede depender del índice r porque la señal subyacente es de ciclo estacionario. La presencia del término $x^2(\tau)$ no contribuye al promedio útil de salida, debemos sospechar de tales términos como fuentes potenciales de ruido propio [2]. Cancelando dichos términos nos queda la siguiente ecuación:

$$u_i(r) = x(\tau + (r-1/2)T) \left\{ \begin{array}{l} x(\tau + rT) - \\ x(\tau + (r-1)T) \end{array} \right\} \quad (10)$$

$$u_i(r) = x(r-1/2) \{x(r) - x(r-1)\} \quad (11)$$

Este es el algoritmo de detección de tiempo (*timing-detector algorithm*) para señales reales de banda base. De la misma manera, si hacemos dos cálculos, uno para el canal I y otro para el canal Q, añadimos al procesador una señal demodulada, teniendo lo siguiente en forma general:

$$u_i(r) = y_i(r-1/2) \{y_i(r) - y_i(r-1)\} + y_Q(r-1/2) \{y_Q(r) - y_Q(r-1)\} \quad (12)$$

El índice r se utiliza para designar el número de símbolo. Se denota el r -ésimo símbolo como $y_I(r)$ y $y_Q(r)$, los valores $y_I(r-1/2)$ e $y_Q(r-1/2)$ se

toman a la mitad del camino entre el $(r-1)$ -ésimo símbolo y el r -ésimo símbolo.

V. DESCRIPCIÓN DE LA SOLUCIÓN

A. Simulación del Algoritmo de Gardner en Código Matlab

Para la simulación se usó el producto de software **MatLab R2011a**, se han utilizado únicamente líneas de comando en donde por simplicidad se ha evitado el uso de herramientas como el Simulink.

B. Consideraciones tomadas para la simulación del Algoritmo de Gardner en Código Matlab

De acuerdo a lo mostrado en la Fig.4, en el receptor para recuperar la señal que fue enviada por el transmisor y afectada por el ruido del canal, la señal es interpolada (en adelante) con un filtro de interpolación que depende del desfase temporal, luego de la interpolación se sincronizará la señal con el algoritmo de Gardner.

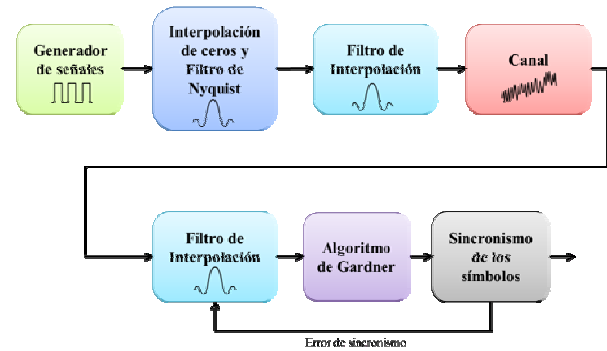


Fig.

4. Diagrama en bloques para la implementación en simulación del Algoritmo de Gardner.

Para la simulación del Algoritmo de Gardner se tomaron las siguientes consideraciones:

- **Transmisor:**
 - Como entrada al transmisor se simuló una señal BPSK que luego fue sobremuestreada al doble.
 - Se simuló el transmisor mediante un filtro de Nyquist y un filtro de interpolación.
 - El filtro de Nyquist sirve para generar un señal sobre muestreada a partir de un procedimiento de interpolación de "ceros"
 - El filtro de Interpolación sirve para retardar la señal sobremuestreada.
- **Canal de Comunicación**
 - Se simuló un canal de comunicación con comportamiento gaussiano el cual afecta a la señal cambiando su magnitud y su fase.
- **Receptor**

- Se simuló un receptor mediante un filtro de interpolación, un bloque que implementa el algoritmo de Gardner y un bloque de sincronismo.
- El filtro de interpolación sirve para adelantar cada muestra de la señal recibida del canal.
- El Bloque Algoritmo de Gardner tendrá como función sincronizar la señal para poder recuperar correctamente los datos enviados por el transmisor.
- El Bloque Sincronismo de los símbolos estima el tiempo en el cual se debe tomar las muestras y corregir su desfasaje temporal.

C. Simulación del Algoritmo de Gardner en C.

El lenguaje C se usa para la simulación de diversos algoritmos, debido a que es más rápido en tiempo de compilación comparado al Matlab. Para la simulación se usó el Software **Code::Blocks - 12.11** en donde se han utilizado líneas de comando.

D. Consideraciones tomadas para la simulación del Algoritmo de Gardner en Lenguaje C.

En el receptor, para recuperar la señal que fue enviada por el transmisor y afectada por el ruido del canal, la señal es interpolada (en adelante) con un filtro de interpolación que depende del desfase temporal, luego de la interpolación se sincronizará la señal con el algoritmo de Gardner.

Para la simulación del Algoritmo de Gardner se tomaron las siguientes consideraciones:

- **Transmisor:**
 - El número de datos que tendrá la señal de entrada se hace a través de un archivo de texto.
 - La señal de entrada se ingresa por un archivo de texto, la señal tiene que ser una señal BPSK
 - Se simuló el transmisor mediante un filtro de Nyquist y un filtro de interpolación.
 - El filtro de Nyquist sirve para generar un señal sobre muestreada a partir de un procedimiento de interpolación de “ceros”
 - El filtro de Interpolación sirve para retardar la señal sobremuestreada.
- **Canal de Comunicación**
 - Se simuló un canal de comunicación en donde se le añadirá ruido a la señal cambiando su magnitud.
- **Receptor**
 - Se simuló un receptor mediante un filtro de interpolación, un bloque que implementa el algoritmo de Gardner y un bloque de sincronismo.
 - El filtro de interpolación sirve para adelantar cada muestra de la señal recibida del canal.
 - El Bloque Algoritmo de Gardner tendrá como función sincronizar la señal para poder recuperar correctamente los datos enviados por el transmisor.

- El Bloque Sincronismo de los símbolos estima el tiempo en el cual se debe tomar las muestras y corregir su desfasaje temporal.
- Los datos de la señal sincronizada son escritos en un archivo de texto.

VI. RESULTADOS

En las figuras siguientes se muestran los resultados obtenidos en la simulación del Algoritmo de Gardner en los lenguajes de programación Matlab y C. Los gráficos obtenidos corresponden a una señal simulada 'x' de longitud igual a 16, se escogen pocos valores para poder tener una visualización grafica de los resultados.

La señal x: es una señal BPSK, para la simulación se tomó el valor $x = [1 \ 1 \ 1 \ -1 \ -1 \ -1 \ -1 \ 1 \ 1 \ 1 \ 1 \ -1 \ -1 \ 1 \ 1 \ 1]$, con esta señal se trabajó para obtener las presentes figuras; **la señal x** se presenta en la Fig.5.

La señal a la entrada del Receptor: es una señal afectada por el ruido del canal (ruido gaussiano), está desfasada tanto en tiempo como en frecuencia, además tiene una amplitud variable debido al ruido del canal; ésta señal se presenta en la Fig.6.

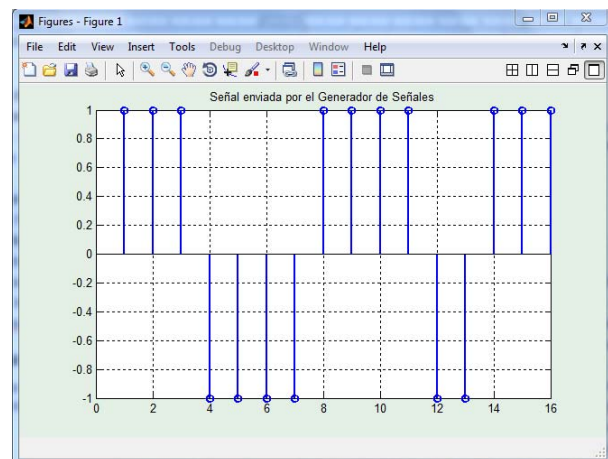


Fig.5. Señal x, señal enviada por el Generador de Señales.

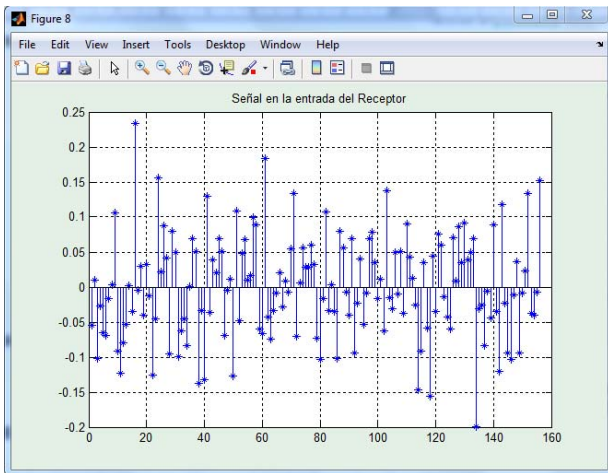


Fig.6. Señal a la entrada del Receptor.

La señal recuperada con el Algoritmo de Gardner: señal que se obtiene a partir de **la señal a la entrada del Receptor**. Esta señal debe ser idéntica a **la señal x** para poder afirmar que el algoritmo cumplió con el trabajo de sincronizar la señal en el tiempo, de la simulación realizada se puede comprobar y afirmar que efectivamente se puede recuperar la señal original, el resultado se presenta en la Fig.7.

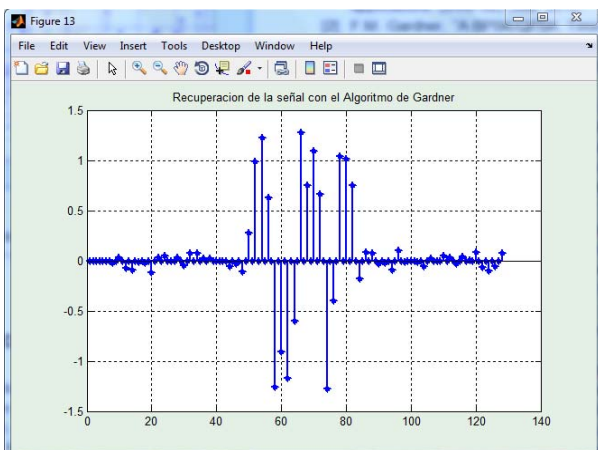


Fig.7. Señal recuperada con el Algoritmo de Gardner.

Finalmente se presenta la Fig.8 en donde se muestra **la señal y** la cual es una señal presente en el **Transmisor** y que se encuentra sincronizada, también muestra **la señal yresult** la que es sincronizada con el Algoritmo de Gardner y se encuentra en el **Receptor**. **La señal yresult** es idéntica a **la señal y**, en otras palabras la curva que representa la señal sincronizada con el Algoritmo de Gardner (**La señal yresult**) sigue a la curva que representa la señal sincronizada originada en el Transmisor (**señal y**). En la simulación que se realizó se comprobó que efectivamente las 2 señales mencionadas anteriormente son muy semejantes y se puede afirmar que el algoritmo de Gardner cumplió su cometido de sincronizar la señal en el tiempo.

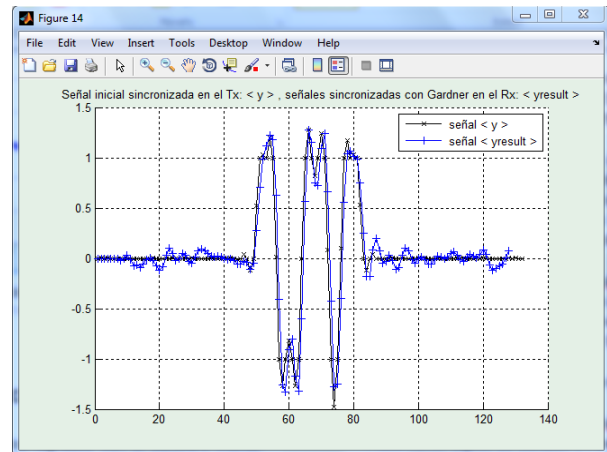


Fig.8. Comparación entre las señales **y** e **yresult**.

La señal y es la señal filtrada por el filtro de Nyquist, esta señal no tiene desfase temporal, es la señal que siempre se debe obtener para la recuperación correcta de la información. **La señal y** en la Fig.6.4 es de color negro.

La señal yresult es la señal interpolada con el filtro de interpolación y es corregida con el Algoritmo de Gardner. **La señal yresult** en la Fig.8 es de color azul.

VII. RESULTADOS

Los resultados logrados con el producto de software Matlab corresponden a los esperados en cuanto a la recuperación del sincronismo de la señal transmitida. Se puede también a partir de este trabajo hacer nuevas simulaciones con canales de comunicación de comportamiento diferente al gaussiano tales como: Raleigh, Rician, etc.

La obtención de los valores que corresponden a los diferentes procesos que se siguen en la simulación del Algoritmo de Gardner nos proporciona una gran facilidad para ver la evolución del algoritmo.

En el lenguaje de programación C, la escritura de los valores en los diversos procesos por donde pasa la señal, en los archivos de texto, ayuda y facilita la tarea de verificación del funcionamiento del Algoritmo de Gardner.

Para la simulación en el lenguaje C, el ingreso de datos por un archivo de texto facilita y hace cómodo el ingreso de la señal; y los valores impresos en los archivos de texto pueden ser usados para hacer gráficos y hacer un análisis estadístico de las señales usando el Matlab.

RECONOCIMIENTOS

Agradezco al Ing. Daniel Díaz Ataucuri y al Ing. Modesto Palma García representantes del INICTEL-UNI por el apoyo y la amistad brindada. Los consejos

y experiencias brindadas que me guían en el camino apasionante de la investigación científica.

REFERENCIAS

- [1] ETSI EN 302 307 v1.3.1, “Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2)”, *ETSI*, Mar 2013.
- [2] F.M. Gardner, “A BPSK/QPSK Timing-Error Detector for Sampled Receivers”, *IEEE Trans. Commun*, Vol. COM-34, pp. 423-429, May 1986.
- [3] J. Bermejo Parra, J. García Lledó y A. Espín Moreno, “Implementación del algoritmo de sincronismo de Gardner en dispositivos FPGA”, *Actas del XXIII Simposium Nacional de la Unión Científica Internacional de Radio (URSI 2008)*, Septiembre 2008.
- [4] P. Savvopoulos and T. Antonakopoulos, “Comparative Performance Analysis of Symbol Timing Recovery for DVB-S2 Receivers”, *University of Patras, Department of Electrical and Computer Engineering*, 2009.
- [5] John G. Proakis y Dimitris G. Manolakis, “Tratamiento digital de señales”, *Pearson Educación S.A.*, 4.^a Edición, 2007.
- [6] Juan R. Clavijo Mendoza, “Diseño y simulación de sistemas microcontrolados en lenguaje C”, ISBN 978-958-44-8619-6, Hecho e impreso en Colombia, 1.^a Edición, Colombia, 2011.
- [7] Javier García de Jalón, José Ignacio Rodríguez, Rufino Goñi, Alfonso Brazález, Patxi Fuentes y Rubén Rodríguez, “Aprenda lenguaje ANSI C como si estuviera en Primero”, Universidad de Navarra, San Sebastián, España, 1998.