



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño



***Implementación de software para la  
sincronización de fase en frecuencia fija  
para sistemas electrónicos monofásicos y  
trifásicos mediante la plataforma  
LAUNCHXL-F28027 (C2000) y el entorno de  
desarrollo Code Composer Studio.***

**MEMORIA PRESENTADA POR:**

*Carlos Crespo Martínez.*

**Máster Universitario en Ingeniería Mecatrónica.**

**DIRECTOR:**

*Francisco Gimeno Sales.*

**CODIRECTOR:**

*Salvador Orts Grau.*

*Valencia-España, julio 2016.*

## Índice.

Capítulo 1. ....	1
Lazo de seguimiento de fase básico. ....	1
1.1    Introducción. ....	1
1.2    Funcionamiento de un PLL. ....	3
1.2.1    PLL operando en seguimiento. ....	3
1.2.2    PLL intentando lograr el seguimiento. ....	3
1.3    Linealización del PLL. ....	4
1.3.1    Rango de seguimiento de un PLL. ....	5
1.3.2    Rango de captura de un PLL. ....	6
1.3.3    Diseño del filtro. ....	6
1.4    Software PLL (SPLL). ....	6
1.4.1    Compensación entre sistema SPLL y sistema PLL por hardware. ....	6
1.4.2    Ejemplo SPLL: Un LPLL como SPLL. ....	7
Capítulo 2. ....	9
Aplicación del PLL. ....	9
2.1    Introducción. ....	9
2.2    Filtrado y reconstrucción de señales con alto niveles de ruido. ....	9
2.3    Sintetizadores de frecuencia. ....	10
2.4    Detección y separación de componentes del espectro de la señal de entrada contenidos en el margen de captura. ....	10
2.5    Control de velocidad de motores. ....	11
2.6    PLL en el control de un motor síncrono de imanes permanentes de 750 kW. ....	13
2.7    PLL en convertidores electrónicos de potencia para máquinas síncronas. ....	14
2.8    Sistemas de sincronización con la red eléctrica utilizando un inversor monofásico como interfaz. ....	15
2.9    Sistemas de sincronización con la red eléctrica utilizando un convertidor monofásico como interfaz. ....	16
Capítulo 3. ....	18
PLL, sincronización con la red eléctrica. ....	18
3.1    Introducción. ....	18
3.2    PLL Monofásico. ....	18
3.2.1    Estructura básica del PLL. ....	18
3.2.2    PLL monofásico digital. ....	18
3.2.3    PLL en cuadratura de la señal de entrada desplazada 90 grados. ....	19

3.2.4 SPLL para sistemas monofásicos de la librería SolarLib C28x. ....	20
3.3 PLL Trifásico.....	26
3.3.1 SRF-PLL: Marco de referencia síncrono – PLL. ....	26
3.3.2 DSRF-PLL: Doble marco de referencia síncrono – PLL.....	27
3.3.3 DSOGI-PLL: Doble Integrador Generalizado de Segundo Orden – PLL.....	27
3.3.4 EPLL: Bucle de seguimiento de fase mejorado.....	28
3.3.5 SPLL-SRF para sistemas trifásicos de la librería SolarLib C28x. ....	28
Capítulo 4. ....	32
Procesador digital de señales.....	32
4.1 Introducción. ....	32
4.2 Controlador digital de señales. ....	34
4.3 Arquitectura del DSC TMS320F28027 de Texas Instruments. ....	34
4.4 Software para DSC TMS320F28027de Texas Instruments.....	36
4.5 Configuración para DSC TMS320F28027.....	36
4.5.1 Mapa de memoria. ....	36
4.5.2 Archivo enlazador de comandos. ....	38
4.5.3 Archivos de cabecera de los registros de periféricos.....	38
4.5.4 Inicialización del procesador. ....	38
4.5.5 Inicialización del sistema. ....	39
Capítulo 5. ....	40
Implementación de software PLL para señales sinusoidales monofásicas y trifásicas como entrada usando el entorno de desarrollo Code Composer Studio y la plataforma LAUNCHXL-F28027 (C2000). ....	40
5.1 Introducción. ....	40
5.2 Implementación de PLL monofásico en cuadratura con señal de ingreso generada con software. ....	40
5.3 Implementación de PLL monofásico en cuadratura con señal de ingreso generada por generador de funciones. ....	42
5.4 Implementación de SPLL para sistemas monofásicos de la librería SolarLib C28x con señal de ingreso generada con software.....	44
5.5 Implementación de SPLL para sistemas monofásicos de la librería SolarLib C28x con señal de ingreso formada por el generador de funciones. ....	48
5.6 Implementación SPLL_3ph_SRF para sistemas trifásicos con la librería SolarLib con señales de ingreso generadas por software. ....	51
5.7 Implementación SPLL_3ph_SRF para sistemas trifásicos con la librería SolarLib con señales de ingreso generada con LaunchPadXL TMS320F28069M. ....	54

5.8 Implementación SPLL\_3ph\_SRF con filtro pasa bajo para sistemas trifásicos utilizando la librería SolarLib con señales de ingreso generada con LaunchPadXL TMS320F28069M. .... 57

## **Objetivo General.**

Implementar, analizar y evaluar códigos propios y propuestos por la librería SolarLib C28x de lazos de seguimiento de fase con la plataforma LAUNCHXL-F28027 (C2000) y el entorno de desarrollo Code Composer Studio proporcionados por Texas Instruments para señales sinusoidales monofásicas y trifásicas de entrada que consideren los tiempos de ejecución y variables necesarias de los algoritmos.

## **Objetivos Específicos.**

- Introducción en el funcionamiento y aplicaciones del lazo de seguimiento de fase o PLL por sus siglas en inglés Phase-Locked Loop.
- Analizar los sistemas PLL's utilizados para la sincronización con la red eléctrica monofásica y trifásica.
- Introducir al procesador digital de señales de la Texas Instruments TMS320F28027 tanto en su arquitectura, software y configuración.
- Implementar códigos de lazo de seguimiento de fase en cuadratura de señales sinusoidales monofásicas como entrada para evaluar el tiempo de ejecución y variables necesarias del algoritmo con el entorno de desarrollo Code Composer Studio y la plataforma LAUNCHXL-F28027 (C2000) proporcionados por Texas Instruments.
- Utilizar la librería SolarLib C28x de Texas Instruments en Code Composer Studio y la plataforma LAUNCHXL-F28027 (C2000) para evaluar y analizar software de lazo de seguimiento de fase para señales sinusoidales monofásicas y trifásicas como entrada que evalúen los tiempos de ejecución y variables necesarias de los algoritmos.

## **Relevancia y aportación del tema.**

Cada día la microelectrónica avanza a pasos agigantados en el transcurso del tiempo, se ha pasado desde el transistor bipolar inventado en la década de los cuarenta hasta diminutos chips programables en la actualidad, sin estos pequeños dispositivos no existirían la gran variedad de sistemas empotrados que facilitan el buen vivir de los seres humanos. Entre estos minúsculos microchips con software incluido están los procesadores digitales de señales que a su vez se sub clasifican en controladores digitales de señales o llamados abreviadamente DSC's. Los DSC's tienen potentes estructuras en Hardware y Software que permiten hacer varias aplicaciones de control, entre estas aplicaciones de control están los lazos de seguimiento de fase. Los lazos de seguimiento de fase son sistemas que son utilizados en redes monofásicas y trifásicas para conocer el ángulo de fase y la frecuencia de la red eléctrica. Antes de inyectar energía eléctrica a la red eléctrica desde un sistema de generación eléctrica es necesario conocer la frecuencia y la fase de la red eléctrica y posteriormente enganchar esta frecuencia y este ángulo a la frecuencia y fase de la señal generada por el sistema de generación eléctrica. Si se desprecia el sincronismo la red eléctrica, las cargas conectadas a la red eléctrica y el sistema de generación de energía se ponen en riesgo. Este enganche entre la señal de la red eléctrica y la señal generada por el método de generación de energía puede realizarse con el lazo de seguimiento de fase.

Este trabajo académico aporta metodologías de aplicación de software PLL sobre la plataforma LAUNCHXL-F28027 (C2000), en estas técnicas se evalúan tiempos de ejecución y variables necesarias de algoritmos tanto para sistemas monofásicos y trifásicos con la ayuda de la librería solar provista por Texas Instruments, a más se valora el espacio ocupado por las memorias Flash y Ram del procesador digital de señales TMS320F28027.

### **Estructura del trabajo académico.**

Este trabajo está estructurado en cinco capítulos que empiezan describiendo teóricamente conceptos hasta su finalización con la parte práctica.

El primer capítulo introduce al lector en la historia, concepto, estructura y funcionamiento del PLL. El segundo capítulo hace una breve descripción del alcance en las aplicaciones de estos dispositivos. La tercera sección da una visión general de los diferentes sistemas de lazo de seguimiento de fase utilizados para la sincronización con la red eléctrica tanto monofásica como trifásica. El cuarto apartado detalla al procesador digital de señales TMS320F28027 de Texas Instruments y al Code Composer Studio como plataformas para la creación de sistemas software de lazo de seguimiento de fase, entre los temas narrados del TMS320F28027 constan su arquitectura y configuración. Finalmente, el capítulo 5 que es la práctica en sí, se refieren los cálculos de los parámetros utilizados, los diagramas de flujos y los esquemas gráficos de las señales al ingreso y a la salida de los códigos de lazo de seguimiento de fase implementados.

## Capítulo 1.

### Lazo de seguimiento de fase básico.

#### 1.1 Introducción.

Históricamente la primera noticia acerca del lazo de seguimiento de fase que se menciona en [1] corresponde al documento: "E. V. Appleton, Automatic synchronization of triode oscillators, *Proc. Cambridge Phil. Soc.*, 21(Part III):231 (1922-1923)", que trata sobre la sincronización automática de dos osciladores, luego en 1932 el lazo enganchado en fase fue referenciado por investigadores que lo usaron para la recepción síncrona de señales de radio, después en 1943 los televisores empezaron a llevar lazos de seguimiento de fase para sincronizar los pulsos de sincronismo de la estación emisora con el barrido de las líneas horizontales y por el año 1950 se aplicó el PLL en la recuperación de la sub portadora del color en receptores de televisión. Su uso fue concreto por su complejidad y alto costo en las áreas de la investigación y militar, hasta la década de los 60 cuando toda la circuitería se logró compactar en un chip, actualmente un chip popular es el LM565. En la década de los setenta se logró el primer lazo de seguimiento de fase híbrido denominado DPLL por sus siglas en lengua inglesa: Digital Phase-Locked Loop, usado para sincronizar las comunicaciones espaciales, su detector de fase fue construido con circuitos lógicos mientras que su oscilador controlado por voltaje y su filtro permanecían analógicos, así se logra mitigar en algo los inconvenientes de los sistemas analógicos como el envejecimiento, variaciones de funcionamiento por cambios de temperatura, sensibilidad a tolerancia y condiciones de operación a los componentes analógicos. La NASA llamada así por sus siglas en inglés National Aeronautics and Space Administration los manipulo para compensar variaciones de frecuencias que toleraban las transmisiones de sus satélites a causa del efecto Doppler e inestabilidad de sus componentes, el efecto Doppler es el aparente cambio de frecuencia de una onda procedente del movimiento relativo del origen respecto al observador.

Los primeros circuitos integrados PLL's fueron construidos con tecnología bipolar, poco tiempo después aparecieron los circuitos lógicos transistor a transistor o TTL (Transistor-transistor logic) y más tarde surgieron los circuitos integrados semiconductor complementario de óxido metálico o CMOS (Complementary metal-oxide-semiconductor). Hay circuitos integrados que contienen un completo sistema PLL y también existen circuitos integrados que contienen parte de un sistema PLL, como ejemplo se puede conseguir circuitos integrados que incluyen uno o dos detectores de fase, además varios sistemas sintetizadores de frecuencia PLL están disponibles en un solo circuito integrado. En la actualidad los chips DPLL de propósito general se basan en el antiguo chip 4046 que es miembro de la familia original CMOS 4000, el 4046 contiene dos detectores de fase: detector fase-frecuencia o PFD (Phase-Frequency detector) y EXOR (OR Exclusiva).



A más del campo de las comunicaciones las técnicas PLL son utilizadas en diversos campos industriales como en el control de motores, procesadores de señales y recientemente las técnicas PLL se usan en la sincronización con la red eléctrica pública con un convertidor o inversor como interfaz.

El lazo de seguimiento de fase también llamado como oscilador enganchado es un sistema electrónico que engancha dos formas de onda con la misma frecuencia, pero desplazada en fase. El fundamental uso de un PLL es comparar las frecuencias de dos formas de onda e igualar la frecuencia de la forma de onda del lazo a la frecuencia de la forma de onda de la señal de entrada. Como lo citan Boylestad y Nashelsky en [2], el PLL está formado por un comparador de fase, un filtro pasa bajo y un oscilador controlado por voltaje o VCO por sus siglas en inglés Voltage-Controlled Oscillator. Referirse a la figura 1.1.

Un comparador de fase es un circuito multiplicador analógico o mezclador de frecuencias el cual genera una diferencia de tensión en representación de una diferencia de fase. Una opción para hacer un comparador de fase puede ser la compuerta X-OR.

Un filtro pasa bajo es un circuito electrónico que deja pasar frecuencias bajas y atenúa las frecuencias altas. Se lo puede implementar con amplificadores operacionales.

Un oscilador controlado por voltaje es un circuito que proporciona una señal variable de salida triangular o cuadrada cuya frecuencia puede ajustarse a lo largo de un rango controlado por una tensión de corriente continua. Una unidad conocida es el circuito integrado 566.

La función del amplificador es aumentar la tensión de la señal de ingreso obteniendo la señal incrementada a la salida.

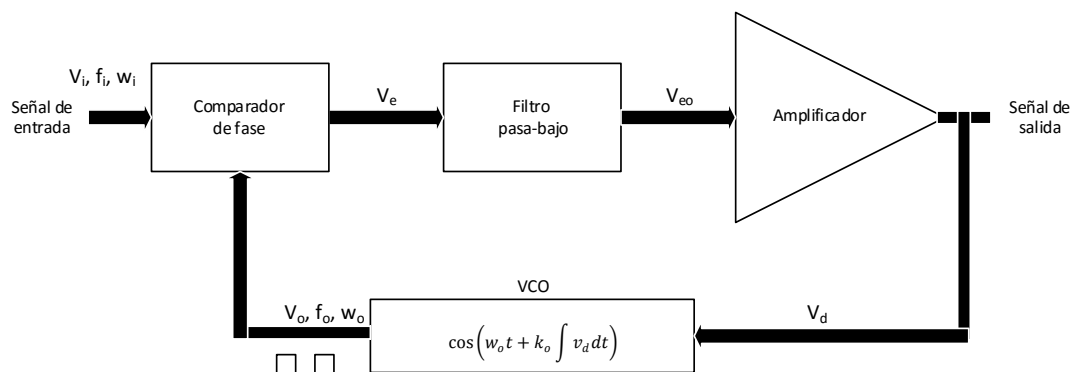


Figura 1.1 Diagrama de bloques del lazo de seguimiento de fase

Se obtiene la señal de salida del comparador de fase  $V_e$  que representa la diferencia de fase entre las señales de entrada  $V_i$  y la señal procedente del VCO  $V_o$ . Posteriormente  $V_e$  es llevado a un filtro pasa-bajos que proporciona un voltaje de salida que puede ser amplificado si se requiere, este voltaje de salida puede ser la señal de salida del PLL y este mismo voltaje  $V_d$  es internamente utilizado para modular la frecuencia  $f_o$  del VCO. El lazo cerrado conserva fija la frecuencia  $f_o$  del VCO respecto a la frecuencia  $f_i$  de la señal de entrada.

Un PLL es un sistema de realimentación con señal de error de realimentación de fase, no hay señal de error de voltaje o corriente como en la mayor parte de los sistemas convencionales.

## 1.2 Funcionamiento de un PLL.

Refiriéndose a la figura 1.1 se explica la operación básica de un PLL cuando el lazo opera en seguimiento y cuando el lazo intenta lograr el seguimiento.

### 1.2.1 PLL operando en seguimiento.

EL PLL opera en seguimiento cuando la frecuencia de la señal de entrada y la frecuencia proveniente del VCO son iguales. Cuando estas dos frecuencias son las mismas el voltaje  $V_d$  es el necesario para mantener en seguimiento al VCO con la señal de entrada. Luego el VCO generará una señal cuadrada con amplitud fija igual a la frecuencia de entrada. Cuando el lazo de seguimiento de fase opera en seguimiento las dos frecuencias que ingresan al comparador de fase serán iguales, aunque sus fases pueden ser diferentes. Una diferencia constante de fase de las señales que ingresan al comparador proporciona un voltaje fijo en continua que va hacia el VCO. Entonces los cambios de frecuencia de la señal de ingreso se traducen como cambios en la tensión en continua que va hacia el VCO. En un rango de frecuencia de seguimiento y captura el voltaje en continua lograra que la frecuencia del VCO coincida con la frecuencia de la señal de ingreso.

### 1.2.2 PLL intentando lograr el seguimiento.

Si se tienen como señales de entradas al comparador de fase:

$$V_i = v_i \sin w_i t$$

y

$$V_o = v_o \cos(w_o t + \varphi(t))$$

Y asumiendo que el comprador es un multiplicador, a la salida del comparador se obtiene la multiplicación de las dos señales:

$$V_e = v_i \sin w_i t \times v_o \cos(w_o t + \varphi(t))$$

Aplicando funciones trigonométricas se tiene:

$$V_e = \frac{v_i v_o}{2} (\sin((w_i - w_o)t - \varphi(t)) + \sin((w_i + w_o)t + \varphi(t)))$$

La señal  $V_e$  tiene componentes de frecuencia con la suma y diferencia de las señales comparadas.

Ahora al pasar  $V_e$  por un filtro pasa-bajo y si las frecuencias  $w_i$  y  $w_o$  son cercanas se obtiene a la salida del filtro la siguiente ecuación:

$$V_{eo} = \frac{v_i v_o}{2} (\sin((w_i - w_o)t - \varphi(t)))$$

Se nota que en  $V_{eo}$  solo se tiene la componente de baja frecuencia de la señal  $V_e$

La señal  $V_{eo}$  está en función de la diferencia instantánea de fase entre las dos señales de entrada al comparador:

$$(w_i t - (w_o t + \varphi(t)))$$

Ahora se considera que la señal a la salida del filtro  $V_{eo}$  tiene  $\varphi(t) = 0$ :

$$V_{eo} = \frac{v_i v_o}{2} (\sin((w_i - w_o)t))$$

y

$$w_i > w_o$$

Para un pequeño tiempo esto resultara un valor positivo para  $V_{eo}$ . Este valor positivo aumenta la frecuencia de salida del VCO haciendo que se acerquen más  $w_i$  y  $w_o$ . Al ser  $V_{eo}$  una señal sinusoidal, después su valor de fase superara los 180 grados ( $(w_i - w_o)t > \pi$ ) y se invertirá el funcionamiento, ahora  $V_{eo}$  será negativo pero la diferencia de las dos frecuencias es tan pequeña que estas no se distanciarán demasiado entre sí como cuando se aproximaron en el tiempo en que la fase era menor a 180 grados. Este ciclo se repite y en cada ciclo las dos frecuencias se aproximan cada vez más hasta que finalmente la señal de entrada y la señal proveniente del VCO obtienen el seguimiento. Un ciclo simétrico iniciaría con un valor negativo de  $V_{eo}$ , cuando  $w_i < w_o$  las dos frecuencias juntas se atraerían.

### 1.3 Linealización del PLL.

La linealización del PLL se hace para facilitar el diseño del filtro. Se linealiza el PLL cerca de la frecuencia de sincronización cuando la tensión a la entrada del VCO es cero. Esta frecuencia es conocida como la frecuencia natural del VCO, entonces el modelo lineal solo funciona con cambios de frecuencias cerca a la frecuencia natural.

La figura 1.2 muestra el diagrama de bloque del PLL linealizado donde  $F(s)$  es un filtro pasa bajo para obtener una señal proporcional ( $K_D$ ) a la diferencia en frecuencia y fase de las dos formas de onda. La señal de salida del filtro  $V_{eo}$  es una señal igual al seno de la diferencia de fase entre las dos señales de entrada, así que la aproximación lineal se cumple cuando la diferencia de fase es pequeña.

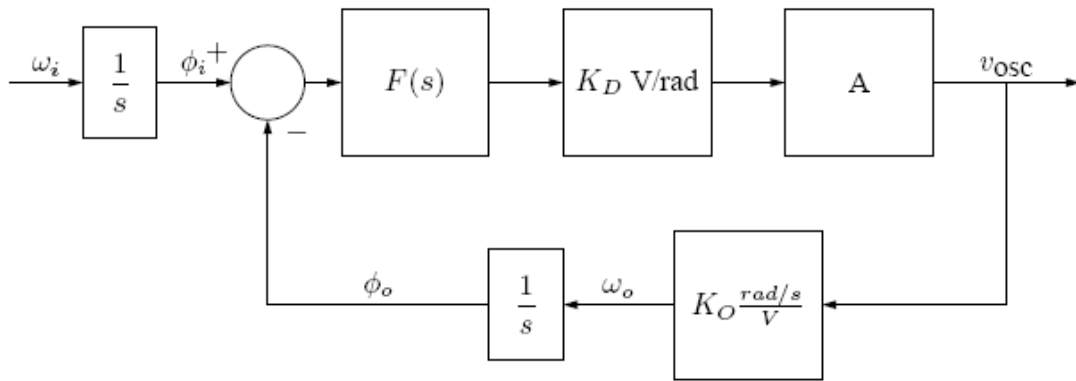


Figura 1.2

H R Pota [3]

La salida del VCO es  $\omega_o$  para un voltaje de entrada igual a cero.

La función de transferencia en el dominio de la frecuencia aplicando la transformada de Laplace es:

$$\frac{V_{osc}(s)}{\Omega_i(s)} = \frac{AF(s)}{s + K_D AF(s)K_o}$$

Reemplazando  $F(s)$  por la función de transferencia de un filtro pasa bajo de primer orden:

$$F(s) = K_D \frac{1}{1 + \frac{s}{\omega_c}}$$

Se obtiene:

$$\frac{V_{osc}(s)}{\Omega_i(s)} = \frac{K_D A}{s \left(1 + \frac{s}{\omega_c}\right) + K_D A K_o} \quad (\text{Ecuación 1.1})$$

### 1.3.1 Rango de seguimiento de un PLL.

El rango de seguimiento del PLL es el rango de frecuencia de la señal de entrada para el cual la salida se enganchará. Este rango es obtenido cuando el PLL está en sincronismo, para conocer el rango de seguimiento del PLL se debe saber la diferencia de fase para la que trabaja el comparador de fase, algunos comparadores trabajan para un rango de 90 grados y otros que trabajan para un rango total de 360 grados. Para un comparador de fase digital que trabaja para un rango de fase de 90 grados el rango de seguimiento del PLL es:

$$\frac{\pi}{2} K_D A K_o$$

Este es rango máximo de variación de frecuencia que se puede tener dentro del bucle.

### 1.3.2 Rango de captura de un PLL.

Es el rango de frecuencia centrado a través de la frecuencia  $f_0$  del VCO. En este rango de frecuencia el lazo que inicialmente no está en seguimiento puede adquirir el seguimiento con la señal de entrada.

Rango de seguimiento de un PLL > Rango de captura de un PLL

### 1.3.3 Diseño del filtro.

Si

$$K_D AK_o$$

es grande y

$$W_c$$

es pequeña entonces la ecuación 1.1 indicaría que el sistema de lazo cerrado de segundo orden tiene polos sub amortiguados, un gráfico del lugar de las raíces lo explicaría. Se necesita que

$$K_D AK_o$$

sea grande para tener un gran rango de seguimiento del PLL, sin embargo, hace que el sistema sea sub amortiguado, una posible solución es tener un cero en el filtro  $F(s)$  que amortigüe el sistema.

## 1.4 Software PLL (SPLL).

### 1.4.1 Compensación entre sistema SPLL y sistema PLL por hardware.

En esta época de los microcontroladores y procesadores digitales de señales es buena idea implementar sistemas PLL por software en donde las funciones del PLL son hechas por un programa de computadora, así como las partes de hardware se van incrementando cuando el sistema PLL va aumentando el nivel de sofisticación el número de instrucciones de computador aumenta con el desarrollo de complejidad que los algoritmos PLL requieren.

SPLL puede competir con soluciones de hardware solo si los algoritmos son ejecutados lo suficientemente rápido en la plataforma de hardware usada para correr el programa. SPLL es ventajoso respecto a soluciones con hardware cuando el poder computacional ya está disponible. Otra gran ventaja es la factibilidad, un diseño SPLL ofrece más grados de libertad disponibles ya que un solo diseño SPLL puede ser adaptado para ser ejecutado a un LPLL (PLL lineal) o a un DPLL (PLL digital) o ejecutar una función que no es capaz de hacer una solución por hardware.

### 1.4.2 Ejemplo SPLL: Un LPLL como SPLL.

El diagrama de flujo de señales de un algoritmo SPLL actuando de forma similar a un sistema hardware LPLL se muestra en la figura 1.3.

La señal de entrada  $u_1$  es una señal analógica que es periódicamente muestreada a la frecuencia  $f_s = 1 / T$  por un ADC (Convertidor Analógico-Digital) donde  $T$  es el tiempo de muestreo.  $u_1(n)$  es una expresión simplificada de la señal de entrada muestreada en el tiempo, las otras señales del SPLL son señales muestreadas también. En consecuencia, todos los bloques del diagrama de flujo de señales trabajan sincrónicamente con el reloj del ADC. En el diagrama de flujo de señales hay tres bloques: un multiplicador digital (MUL), un filtro digital (Digital Filter) y un DCO (Oscilador Controlado Digital). La salida del multiplicador digital que trabaja como un detector de fase es  $u_d(n)$ . El filtro digital funciona como un filtro bucle y su salida es  $u_f(n)$  y finalmente el DCO genera una señal de salida de onda cuadrada muestreada  $u_2(n)$ , en donde se calcula  $u_2(n)$  de la fase  $\phi_2(n)$  del DCO.

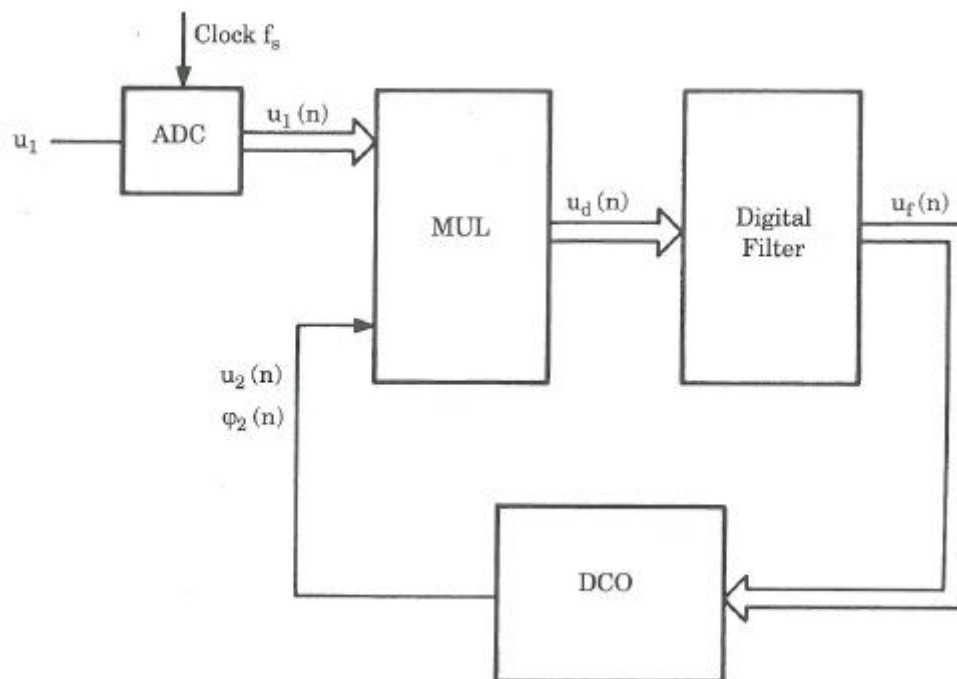


Figura 1.3 Diagrama de bloque de un SPLL actuando como LPLL

ROLAND E. BEST [4]

Ahora la figura 1.4 representa el algoritmo SPLL en un diagrama de estructura. Al iniciar el algoritmo se asigna valores iniciales a las variables importantes, posteriormente el programa entra en un ciclo interminable hasta que el sistema es detenido, esta detención es asumida por la señal de reloj que periódicamente genera interrupciones en el microcontrolador o procesador digital de señales, estas interrupciones son hechas en los instantes de muestreo. Tan pronto como la interrupción es reconocida el algoritmo SPLL es ejecutado. La ejecución del algoritmo empieza con la adquisición de una muestra de la señal de entrada  $u_1(t)$ , en las

próximas tres sentencias del diagrama de estructura se calcula el valor actual de las señales  $u_d(n)$  y  $u_f(n)$  y el valor actual más uno de  $\phi_2(n)$  y  $u_2(n)$ . Finalmente se desplaza en el tiempo todas las variables del SPLL.

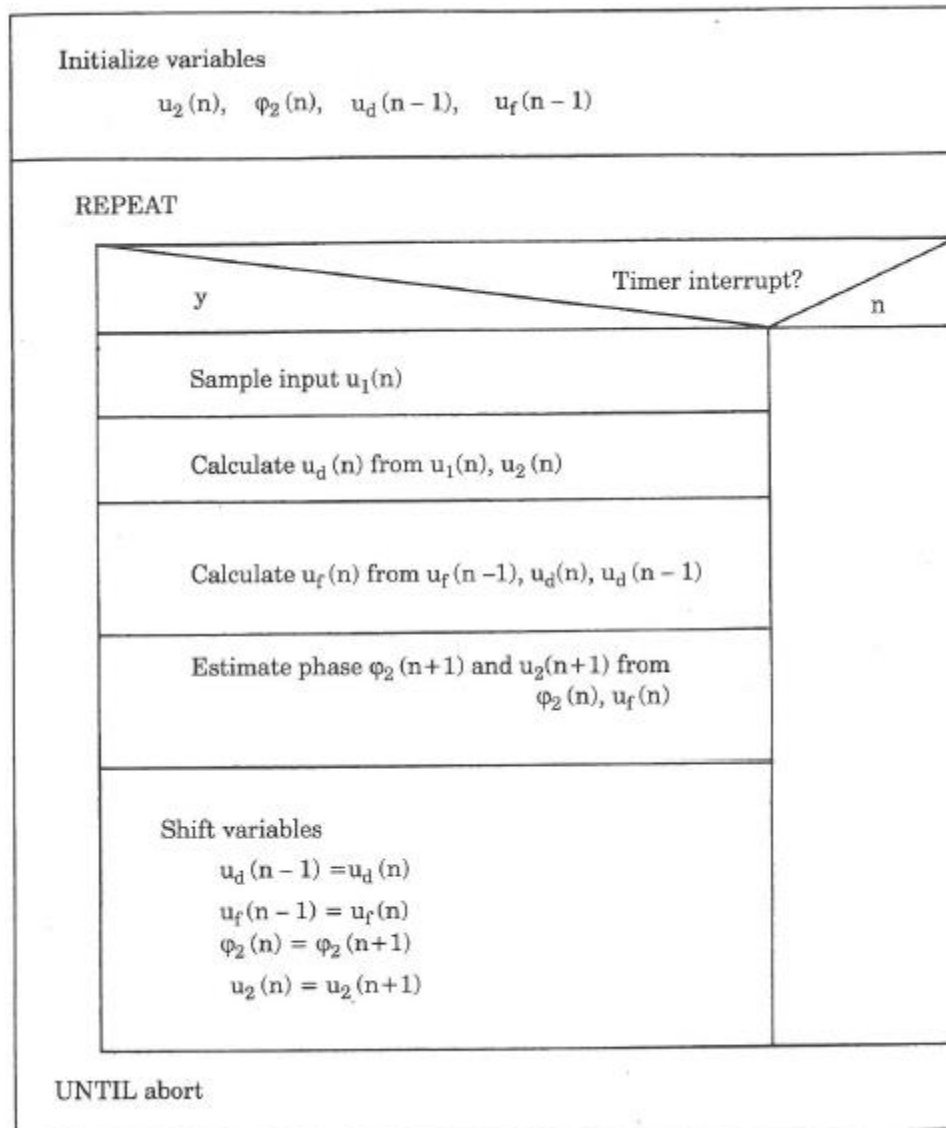


Figura 1.4 Diagrama de estructura definiendo la operación del SPLL

## Capítulo 2.

### Aplicación del PLL.

#### 2.1 Introducción.

Los televisores actuales aplican hasta dos lazos de seguimiento de fase, uno que engancha en fase y frecuencia a la etapa de salida horizontal (HSync) y otro para la sincronía de la generación de la señal del barrido de líneas (VSync), otras aplicaciones del PLL entran en los campos de la demodulación FM, multiplicación y división de frecuencia. FM se refiere a Frecuencia Modulada. Estos sistemas también son utilizados en la sincronización con la red eléctrica tanto para sistemas monofásicos como trifásicos, los sistemas de alimentación conectados a la red eléctrica necesitan saber el ángulo de fase y frecuencia del suministro de red, entre estos sistemas de alimentación están la fuente de alimentación ininterrumpida o UPS por sus siglas en inglés: Uninterruptible Power Supplies, inversores trifásicos y monofásicos, los sistemas flexibles de transmisión de corriente alterna o FACTS por sus siglas en inglés: Flexible AC Transmission Systems, entre otros más.

A continuación, se nombran algunas aplicaciones de los lazos de seguimiento de fases.

#### 2.2 Filtrado y reconstrucción de señales con alto niveles de ruido.

La figura 2.1 muestra la respuesta del circuito de una señal contaminada con una frecuencia de 1040 Hz. Como la frecuencia fundamental es la única dentro del rango de captura se produce una sintonía a su frecuencia.

$V_i$ : Señal contaminada.

$V_o$ : Señal del PLL reconstruida.



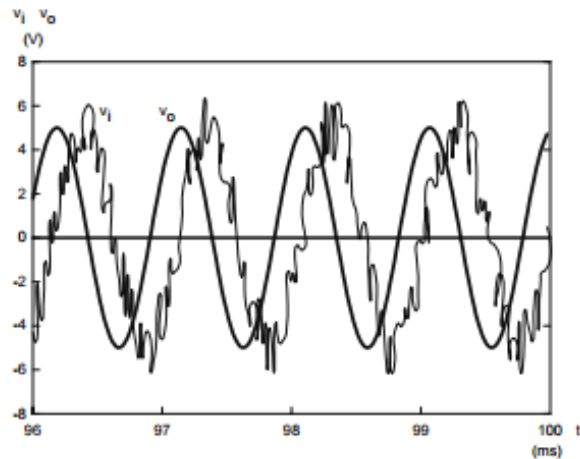


Figura 2.1

R. Pindado [5]

### 2.3 Sintetizadores de frecuencia.

El diagrama de bloque de un sintetizador de frecuencia de la figura 2.2 consiste en realimentar el circuito de un PLL mediante un divisor de frecuencia con relación  $N$  consiguiendo así la sintonía con relación  $\bar{\omega}_o = N\bar{\omega}_i$ .

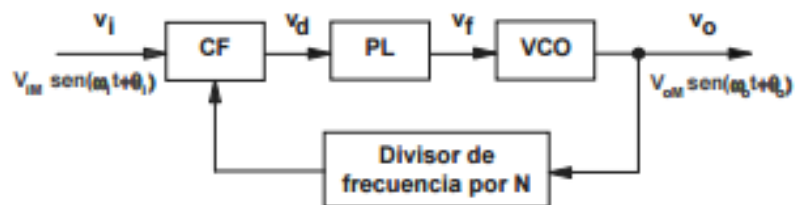


Figura 2.2

R. Pindado [5]

### 2.4 Detección y separación de componentes del espectro de la señal de entrada contenidos en el margen de captura.

La figura 2.3 muestra la sintonía de una señal cuadrada de entrada con una amplitud de 5 V y 140 Hz de frecuencia con su séptimo armónico con 980 Hz y 910mV que cae dentro del margen de captura controlando la frecuencia de la señal de salida.

$V_i$ : Señal cuadrada de entrada.

$V_o$ : Señal del séptimo armónico de  $V_i$ .

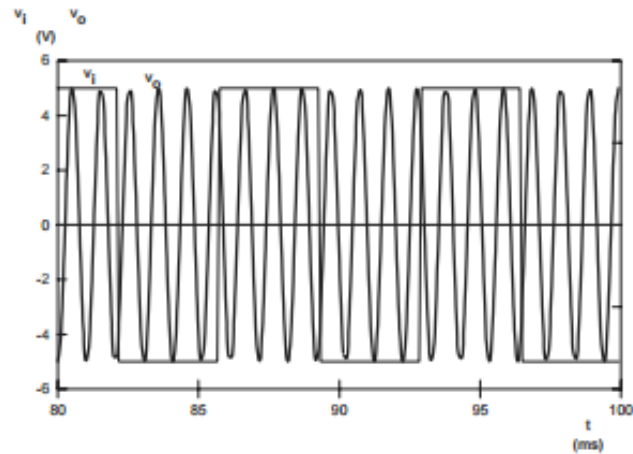


Figura 2.3

R. Pindado [5]

### 2.5 Control de velocidad de motores.

Controles de velocidad muy precisos a bajo costo son posibles con el PLL. Para evidenciar las ventajas del control de velocidad del motor con el PLL se compara con un control de velocidad convencional de un motor.

La figura 2.4 muestra el esquema convencional del control de velocidad de un motor.

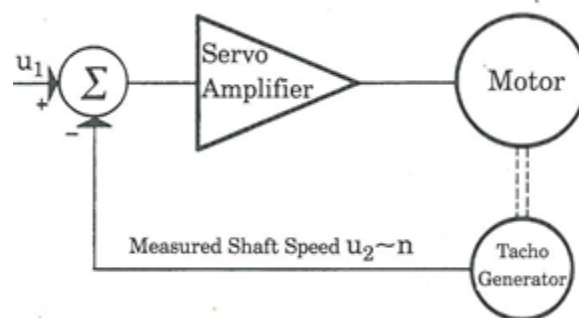


Figura 2.4 Diagrama de bloque convencional del control de velocidad de un motor

ROLAND E. BEST [4]

$U_1$  es la velocidad de referencia del eje del motor,  $U_2$  es la señal de velocidad del eje del motor medida con un tacómetro y es proporcional a  $n$ ,  $n$  es la velocidad del motor.

El servoamplificador amplifica la diferencia entre la velocidad real y la velocidad de referencia, la salida del amplificador controla el motor. La salida del servoamplificador es alta pero finita y para controlar el motor un error diferente de cero debe existir.

La no linealidad del tacómetro es una fuente de error y su alto costo es otro inconveniente.

Ahora la figura 2.5 muestra el diagrama de bloque de un control de velocidad para un motor basado en PLL.

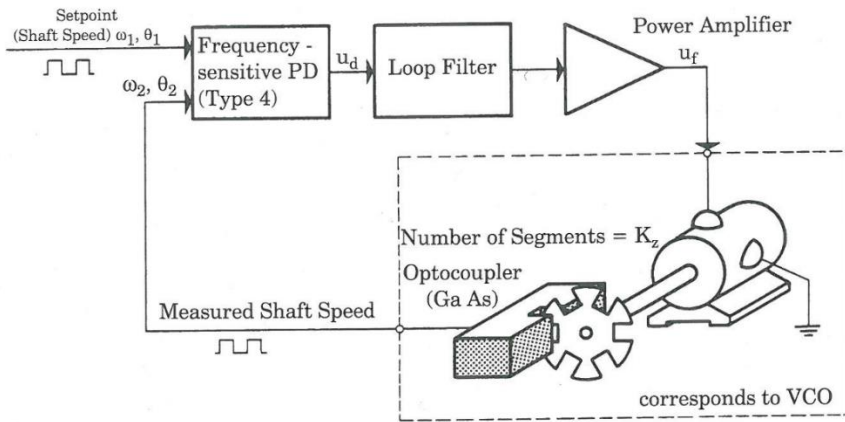


Figura 2.5 Diagrama de bloque de un control de velocidad de un motor basado en PLL

ROLAND E. BEST [4]

El sistema de control es un PLL en el cual el VCO es remplazado por la combinación de un motor y un tacómetro óptico. La señal medida del eje del motor es generada por un auto acoplador en el cual el haz de luz es cortado por un sector de disco. La figura 2.6 detalla el circuito del auto acoplador, el auto acoplador es fabricado de un diodo emisor de luz y un fototransistor, para lograr una señal cuadrada a la salida a la etapa del fototransistor le sigue un disparador Schmitt CMOS tipo 74HC/HCT14.

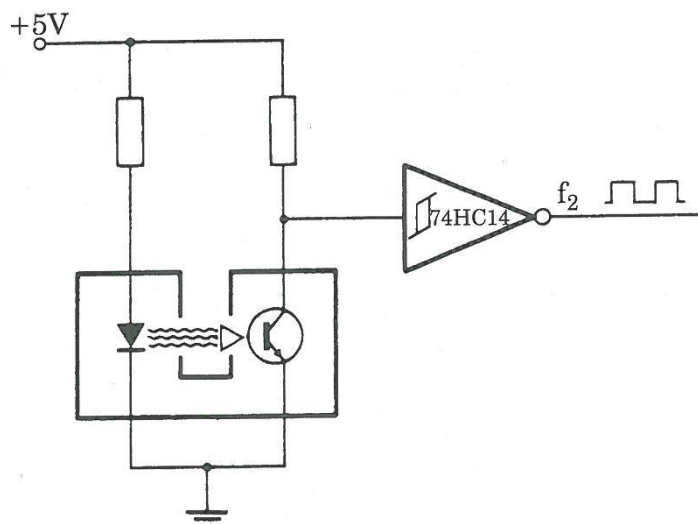


Figura 2.6 Circuito del tacómetro óptico

ROLAND E. BEST [4]

La señal generada por el auto acoplador tiene una frecuencia proporcional a la velocidad del motor ya que el detector de fase compara tanto las frecuencias como las fases entre la señal de referencia y la señal de velocidad medida, el sistema se estabiliza a error de velocidad cero.

## **2.6 PLL en el control de un motor síncrono de imanes permanentes de 750 kW.**

El esquema 2.7 muestra el sistema de control de un generador síncrono de imanes permanentes de 750 kW. El bloque bc/qd convierte la corriente trifásica a corrientes de ejes qd. El bloque regulador de corriente (Current regulation) regula independientemente las corrientes del eje q y eje d. El bloque modulación por ancho de pulso o PWM por sus siglas en inglés Pulse-Width Modulation es un bloque de modulación por ancho de pulso de espacio vectorial que convierte las señales de tensión del eje q y eje d a el tiempo de encendido y apagado de cada transistor bipolar de puerta aislada o IGBT por sus abreviaturas en inglés Insulated Gate Bipolar Transistor. El bloque de captura de giro y detección del ángulo inicial (Catch spin & initial angle detection) detecta la velocidad del generador y el ángulo inicial durante el arranque.

En funcionamiento normal el comando del eje d de corriente ( $I_d \text{ cmd}$ ) es puesto en cero y el comando del eje q de corriente ( $I_q \text{ cmd}$ ) es controlado por la salida del bloque de regulación de voltaje (Vdc regulation). Si el bus de corriente directa (Vdc) es mayor que la tensión de ajuste de 700 voltios la salida del bloque de regulación de voltaje se hace positivo y si el bus de corriente directa es menor que 700 voltios el comando del eje q de corriente se hace negativo. Al tener un  $I_q \text{ cmd}$  positivo hace que automáticamente el motor convierta la potencia eléctrica de entrada en energía cinética y un  $I_q \text{ cmd}$  negativo hace que el motor genere energía.

La medición del flujo magnético es clave para el control del motor sin sensor. El rendimiento de la medida del flujo afecta directamente al rendimiento del sistema de control del motor. El bloque de la medición del flujo y PLL (Flux estimation & PLL) es utilizado para obtener las medidas de posición y velocidad del motor.

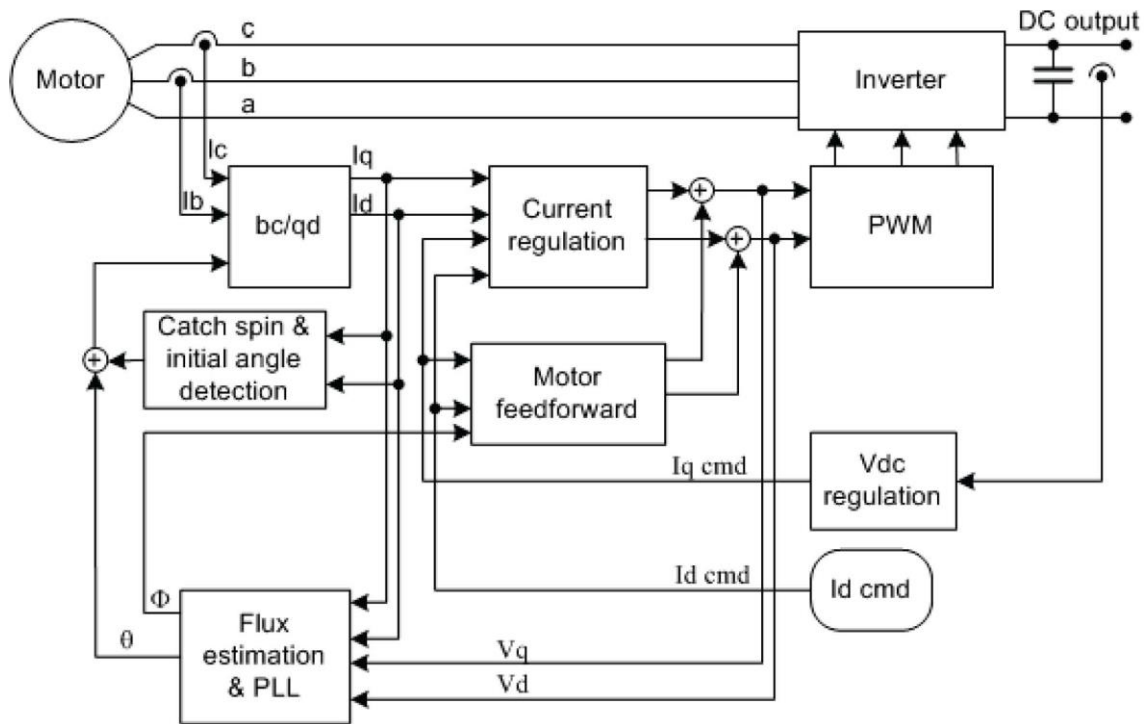


Figura 2.7 Esquema del sistema de control

Liping Zheng, Dong Le [6]

### 2.7 PLL en convertidores electrónicos de potencia para máquinas síncronas.

Para conocer el ángulo de fase del voltaje en inversores de auto-conmutación se vale en implementar un PLL en el circuito de control. La figura 2.8 integra un regulador PLL al modelo dinámico de un inversor en donde  $U_{DC}^*$  es la tensión de referencia del bus de corriente continua del inversor,  $U_{DC}$  es la tensión actual del bus de corriente continua del inversor,  $H_{(VDC)}(s)$  es el regulador de tensión del bus de corriente continua del inversor,  $I^*$  es la corriente de referencia del inversor,  $I_{inv}$  es la corriente actual del inversor,  $H_i(s)$  es el modelo del controlador de corriente,  $C$  es la capacitancia del bus de corriente directa del inversor,  $U_{grid}$  representa la tensión y corriente de red,  $\delta_{inv}$  es la referencia del ángulo de fase del inversor,  $U_{pll}$  es el voltaje de salida del regulador PLL y  $H_{pll}$  es el modelo del regulador PLL.

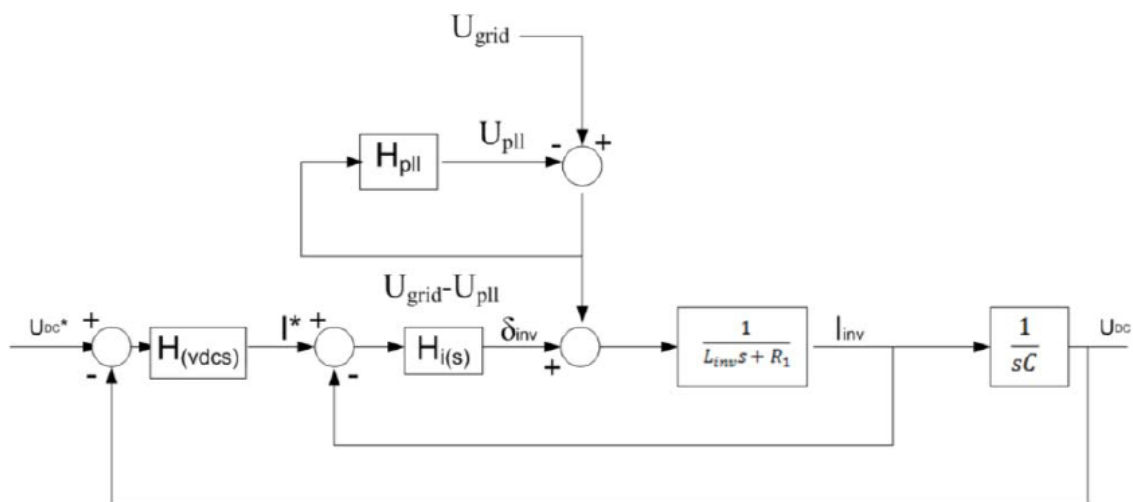


Figura 2.8 Modelo dinámico de un inversor con el regulador integrado PLL

Colin Debruyne\*, Stijn Derammelaere\*, Jan Desmet\*, Lieven Vandeveldt [7]

**2.8 Sistemas de sincronización con la red eléctrica utilizando un inversor monofásico como interfaz.**

El PLL sincroniza el ángulo de fase instantáneo de un inversor de tensión con el ángulo de fase de la red eléctrica para lograr un factor de potencia cercano a la unidad.

La figura 2.9 muestra el diagrama de bloque de un típico inversor de tensión conectado a la red eléctrica empleando PLL como método de sincronización.

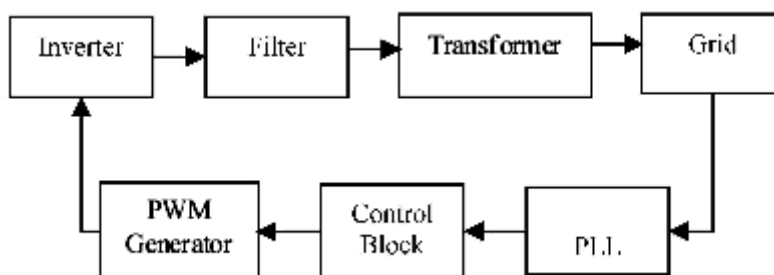


Figura 2.9

K.M.S.Y Konara, M.L. Kolhe, W.G.C.A. Sankalpa [8]

Refiriéndose a la figura 2.10 la señal de referencia de la red eléctrica entra a un sistema de sincronización llamado por ejemplo SOGI-PLL por sus siglas en inglés Second Order Generalized Integrator - Phase-Locked Loop y a la salida se tiene una señal sinusoidal normalizada en amplitud y sincronizada en fase y frecuencia que es enviada hacia el Control del inversor CD/CA generando así la modulación de ancho de pulso o PWM. Finalmente, la señal generada

por la energía renovable y la señal de red tendrán la misma fase y frecuencia y se podrá inyectar a la red sin dificultades.

El inversor CD/CA convierte la corriente directa en corriente alterna, CD se entiende por Corriente Directa y CA por Corriente Alterna. PWM se encarga de generar una señal sinusoidal y la etapa de control se encarga de la conmutación de los dispositivos semiconductores para definir la salida sinusoidal que se desea.

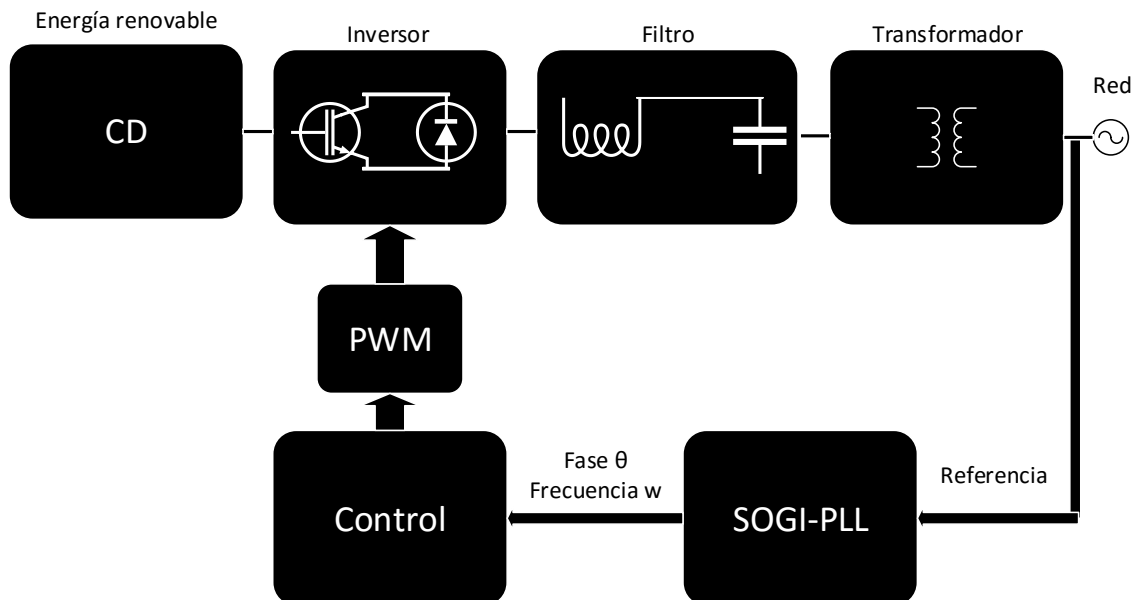


Figura 2.10 Sistema general de un generador de energía eléctrica conectado a la red eléctrica

Villalobos, R.-López, H.-Vázquez, N.-Hernández, C. [9]

### 2.9 Sistemas de sincronización con la red eléctrica utilizando un convertor monofásico como interfaz.

La figura 2.11 hace referencia a un subsistema de control basado en PLL que genera las variables necesarias como la frecuencia y el ángulo de fase de la tensión de la red para regular y sincronizar adecuadamente un convertidor con la red eléctrica monofásica. Pese a las distorsiones y transitorios de red tales como armónicos, cambios de tensión y frecuencia, el PLL debe seguir con precisión y rapidez al mismo tiempo que proporcione una salida de baja distorsión hacia el sistema de control bajo cualquier condición imaginable de la red, caso contrario se podrían presentar problemas tales como disparos falsos de los componentes electrónicos en el convertidor o incluso dificultades de estabilidad del sistema.

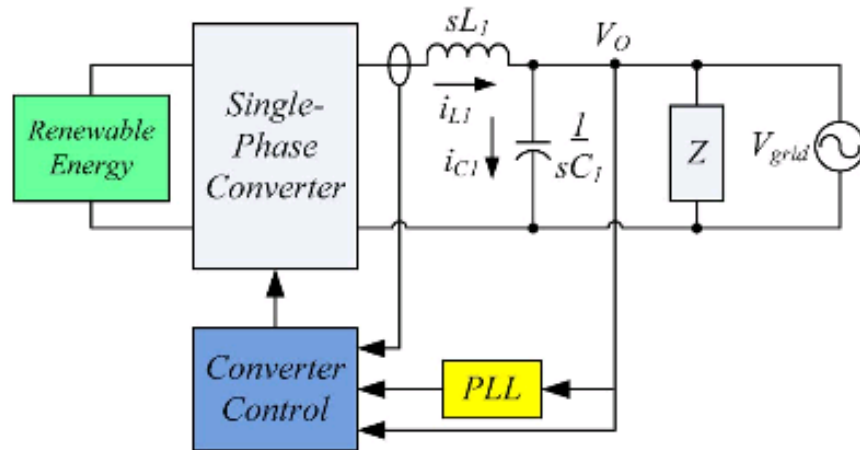


Figura 2.11

Timothy Thacker, Dushan Boroyevich, Rolando Burgos, Fei Wang [10]

El puente H del convertidor monofásico junto al control, conexión a la red eléctrica y diagrama PLL es mostrado en la figura 2.12. La corriente de referencia  $I^*$  se selecciona para proporcionar plena carga a la demanda de tal manera que no se inyecte corriente de la red.

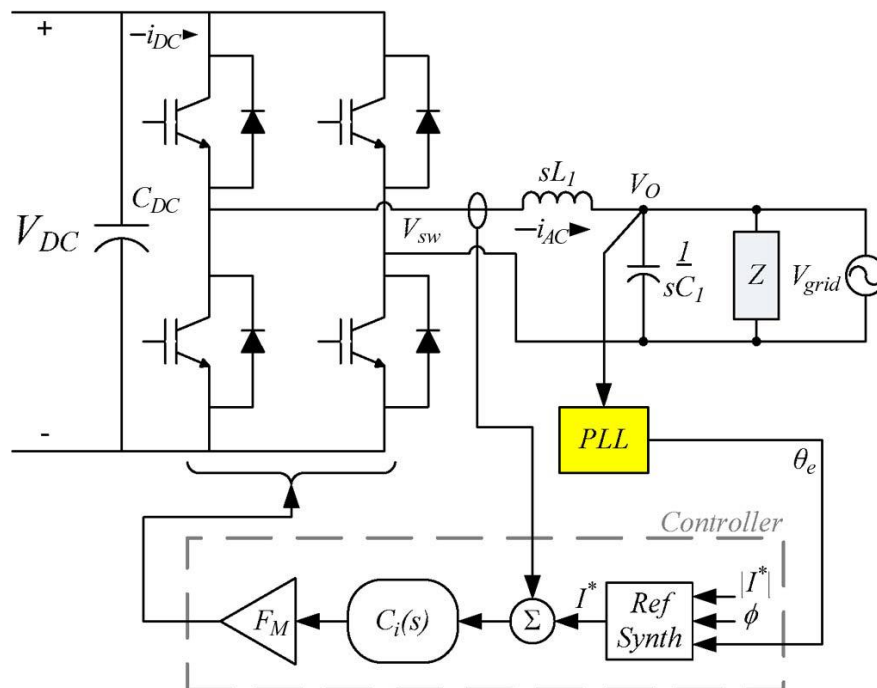


Figura 2.12 Convertidor monofásico con lazo de corriente interno y control PLL

Timothy Thacker, Dushan Boroyevich, Rolando Burgos, Fei Wang [10]

Este trabajo académico está aplicado a observar el funcionamiento de software PLL cuando se tiene como ingreso una señal monofásica o trifásica sinusoidal.



## Capítulo 3.

### PLL, sincronización con la red eléctrica.

#### 3.1 Introducción.

Desde hace poco las técnicas PLL se utilizan para la sincronización con la red eléctrica de servicio público. Un PLL ideal proporciona una rápida y precisa información de sincronización con alto grado de inmunidad e insensibilidad a todo tipo de distorsiones de la señal de entrada y perturbaciones. Este capítulo muestra un estudio de las diferentes técnicas de sincronización del PLL en redes monofásicas y trifásicas.

#### 3.2 PLL Monofásico.

Entre los sistemas PLL monofásicos sincronizados con la red eléctrica utilizados y tratados en esta memoria se mencionan:

- Estructura básica del PLL.
- PLL monofásico digital.
- PLL en cuadratura de la señal de entrada desplazada 90 grados.
- SPLL para sistemas monofásicos de la librería SolarLib C28x.

##### 3.2.1 Estructura básica del PLL.

La solución de sincronización más aceptada para una señal variante en el tiempo es descrita en la figura 3.1 que resume lo que cuenta el capítulo 1, en donde la diferencia entre el ángulo de fase de la señal de entrada ( $V_{in}$ ) y el ángulo de fase a la salida ( $\hat{\theta}$ ) es medida por un detector de fase (phase detection) e ingresada a un filtro (loop filter). El filtro controla al VCO que genera la señal de salida la cual sigue a la señal de entrada.

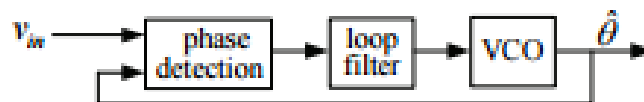


Figura 3.1 Estructura básica del PLL

Xiao-Qiang GUO, Wei-Yang WU, He-Rong GU [11]

##### 3.2.2 PLL monofásico digital.

Este sistema es común en convertidores de potencia, la figura 3.2 muestra un PLL digital monofásico con multiplicador lineal como detector de fase.

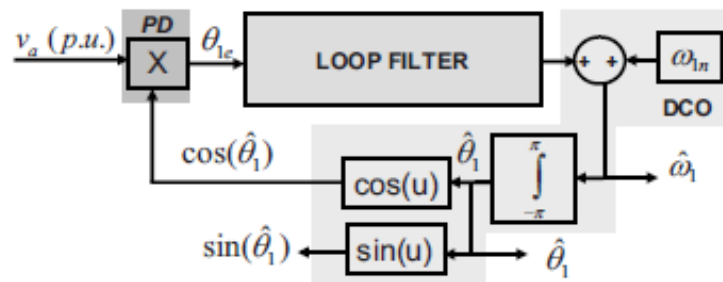


Figura 3.2 PLL digital monofásico

Francisco D. Freijedo, Jesus Doval-Gandoy, Oscar Lopez, Carlos Martinez-Peñalver, Alejandro G. Yepes, Pablo Fernandez-Comesaña, Jano Malvar, Andres Nogueiras, Jorge Marcos y Alfonso Lago [12]

El término VCO ha sido remplazado por el DCO que se entiende como Oscilador controlado digitalmente.

El PLL monofásico digital es el sistema más lineal comparado con los otros PLL's monofásicos, sin embargo, debido a las no linealidades del detector de fase todos los PLL's monofásicos necesitan un ancho de banda reducido en aplicaciones con la red eléctrica. Por tanto, el PLL monofásico conectado a la red eléctrica es lento de hecho sus tiempos de establecimientos son más grande que un ciclo de la frecuencia natural  $\omega_{1n}$ .

### 3.2.3 PLL en cuadratura de la señal de entrada desplazada 90 grados.

Este método consiste en usar la cuadratura de la señal de entrada  $\pi/2$  radianes creando un sistema de dos fases ficticias. Basta con desplazar la señal de entrada monofásica de la red un ángulo  $\pi/2$  radianes hacia atrás. Esta técnica es usada para la detección de fase. Este método trabaja bien para frecuencia fija, por ejemplo 50 Hz. Si la frecuencia cambia entonces el PLL obtendrá un considerable error.

Las señales ortogonales pueden producir errores de sincronización si la señal de entrada contiene armónicos, debido a que cada armónico causa un retardo con respecto a la componente fundamental.

El controlador PI (Proporcional-Integral) es un filtro pasa bajas que atenúa las altas frecuencias que puedan producirse en  $i'_\alpha$  y  $i'_\beta$  siendo al final un controlador de error en régimen permanente (ver figura 3.3).

La salida de un controlador PI es igual a la variación proporcional de la magnitud del error más la variación proporcional del tiempo en que ha permanecido el error. La acción del compensador PI es definida como:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt$$

Donde  $u(t)$  es la acción de control,  $e(t)$  el error,  $K_p$  es la constante proporcional y  $T_i$  denominado tiempo integral. El tiempo integral regula la velocidad de acción de control, mientras que una variación de la constante de proporcionalidad afecta a la parte integral y a la parte proporcional de la acción de control.

La función de transferencia del sistema de control es:

$$\frac{U(s)}{E(s)} = K_p \left( 1 + \frac{1}{T_i s} \right) = K_p + \frac{K_i}{s}$$

Donde  $K_i$  se llama constante integral y se define como:

$$K_i = \frac{K_p}{T_i}$$

La figura 3.3 muestra la estructura del PLL

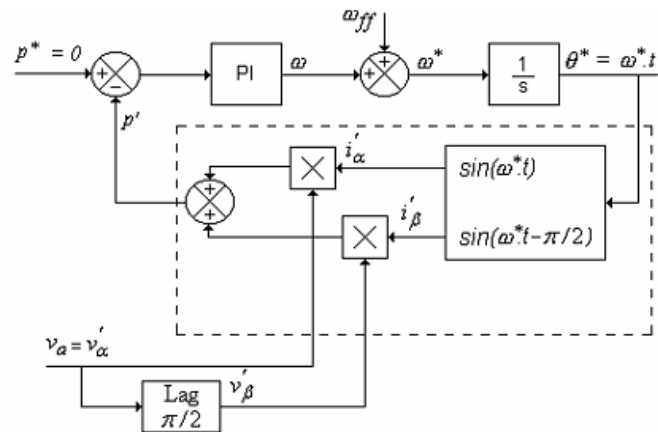


Figura 3.3

TI E2E Community [13]

### 3.2.4 SPLL para sistemas monofásicos de la librería SolarLib C28x.

Esté módulo de software implementa un SPLL para calcular la fase instantánea de una red monofásica y los valores seno y coseno de la red monofásica usadas en los lazos de control.

La figura 3.4 indica que la entrada en este módulo es  $.V_{in}$ ; sus salidas son:  $.\sin(\theta)$ ,  $.\cos(\theta)$ ,  $.\theta$  y su parámetro interno es  $wn$ .

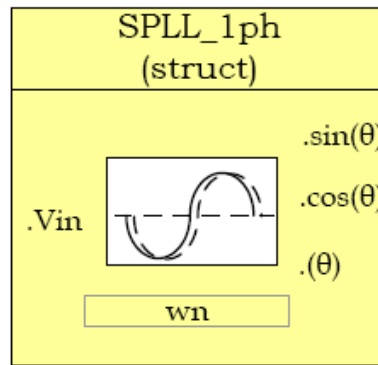


Figura 3.4

Texas Instruments. C28x Solar Library [14]

La figura 3.5 muestra el diagrama de un PLL formado por un detector de fase (PD), un filtro pasa bajos (LPF) y un Oscilador controlado por voltaje (VCO).

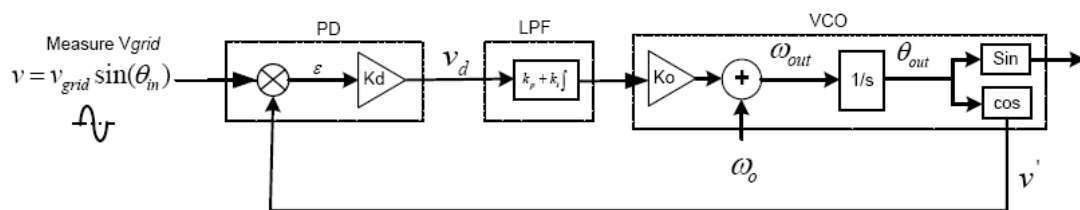


Figura 3.5

Texas Instruments. C28x Solar Library [14]

Una medida de señal monofásica proveniente de la red eléctrica se representa matemáticamente como:

$$v = v_{grid} \sin(\theta_{in})$$

Grid se traduce como red eléctrica.

Al hacer:

$$\theta_{in} = \omega_{grid}t + \theta_{grid}$$

Se tiene:

$$v = v_{grid} \sin(\omega_{grid}t + \theta_{grid})$$

Por otra parte, la salida del VCO es:

$$v' = \cos(\theta_{out})$$

Al hacer:

$$\theta_{out} = w_{PLL}t + \theta_{PLL}$$

Se obtiene:

$$v' = \cos(w_{PLL}t + \theta_{PLL})$$

Al multiplicar la salida del VCO por la señal monofásica medida de la red eléctrica y por la ganancia del PD ( $K_d$ ) se calcula la salida del PD:

$$v_d = v * v' * K_d$$

$$v_d = \frac{K_d v_{grid}}{2} \left[ \sin((w_{grid} - w_{PLL})t + (\theta_{grid} - \theta_{PLL})) \right. \\ \left. + \sin((w_{grid} + w_{PLL})t + (\theta_{grid} + \theta_{PLL})) \right]$$

$v_d$  proporciona información de la diferencia de fase entre la fase de red y la fase del PLL, aunque esta información contiene una componente de alta frecuencia. Para solucionar esta componente de alta frecuencia se añade el controlador PI (LPF) que actúa como filtro pasa bajo y elimina las componentes de alta frecuencia. A la salida del LPF se logra:

$$\bar{v}_d = \frac{K_d v_{grid}}{2} \sin((w_{grid} - w_{PLL})t + (\theta_{grid} - \theta_{PLL}))$$

Funcionando el PLL establemente o lo que es lo mismo cuando las frecuencias son iguales, se hace nulo:

$$w_{grid} - w_{PLL}$$

Y al linealizar el funcionamiento del PLL:

$$\sin(\theta) = \theta$$

Se logra:

$$\bar{v}_d = \frac{K_d v_{grid}}{2} (\theta_{grid} - \theta_{PLL})$$

Por otro lado, la figura 3.5 muestra que:

$$\bar{v}_d = err * K_d$$

Entonces al combinar las dos anteriores ecuaciones se consigue la ecuación del error:

$$err = \frac{v_{grid}(\theta_{grid} - \theta_{PLL})}{2}$$

La función de transferencia en bucle cerrado viene dada por:

$$\text{Bucle cerrado TF} = \text{Bucle abierto TF} / (1 + \text{Bucle abierto TF})$$

Entonces la función de transferencia en lazo cerrado de la fase del PLL puede escribirse como:

$$H_o(s) = \frac{\theta_{out}(s)}{\theta_{in}(s)} = \frac{LF(s)}{s+LF(s)} = \frac{v_{grid}\left(k_p s + \frac{k_p}{T_i}\right)}{s^2 + v_{grid}k_p s + v_{grid}\frac{k_p}{T_i}}$$

Esta función de transferencia es de orden 2 porque tiene dos polos y representa las características de un filtro pasa bajo que ayuda a atenuar los armónicos de orden superior.

Y la función de transferencia en lazo cerrado del error del PLL se puede escribir como:

$$E_o(s) = \frac{V_d(s)}{\theta_{in}(s)} = 1 - H_o(s) = \frac{s}{s + LF(s)} = \frac{s^2}{s^2 + k_p s + \frac{k_p}{T_i}}$$

Esta función de transferencia indica que hay dos polos en el origen lo que significa que es posible rastrear una rampa de pendiente constante en el ángulo de fase de entrada sin error en estado estacionario.

Mientras que la función de transferencia genérica de segundo orden es:

$$H(s) = \frac{2\zeta w_n s + w_n^2}{s^2 + 2\zeta w_n s + w_n^2}$$

Al comparar  $H_o(s)$  con  $H(s)$  se tiene la frecuencia natural y el coeficiente de amortiguación:

$$w_n = \sqrt{\frac{v_{grid}k_p}{T_i}}$$

$$\zeta = \sqrt{\frac{v_{grid}T_i k_p}{4}}$$

El PI en el PLL de la figura 3.5 sirve para:

- Filtrar altas frecuencias que son el doble de la portadora de red (señal que sale del VCO, esta señal se denomina portadora porque varía la frecuencia según la señal moduladora que es la señal de la red eléctrica monofásica).
- Control de respuestas del PLL a cambios de las condiciones de la red como cambios de fase o sobretensiones.

El LPF soluciona las altas frecuencias que podría tener la portadora de la red sin embargo el LPF no atenúa satisfactoriamente las pequeñas frecuencias provenientes de la red eléctrica produciendo a la salida del LPF componentes de frecuencias altas que afectan al rendimiento del PLL. Para atenuar la frecuencia proveniente de la red se utiliza un filtro pasa banda también llamado NotchFilter a la salida del PD. La figura 3.6 muestra el sistema total del PLL que maneja la librería SolarLib C28x.

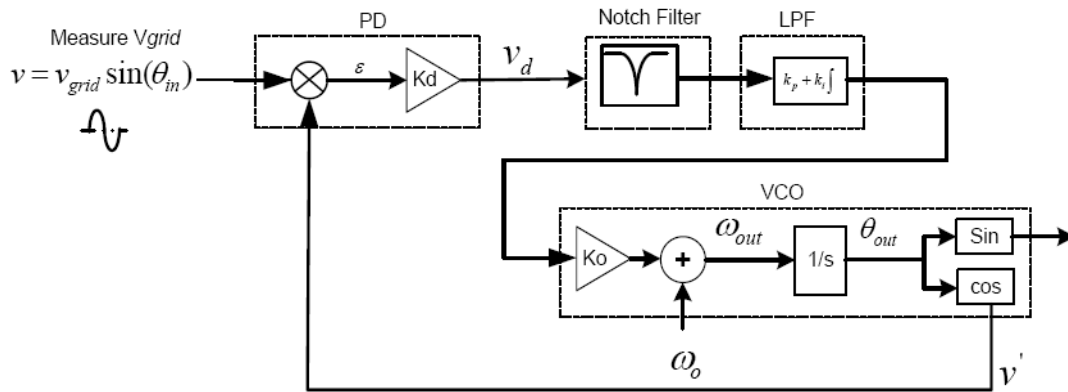


Figura 3.6

Texas Instruments. C28x Solar Library [14]

En este caso la puesta en funcionamiento del PI puede hacerse basándose en la respuesta dinámica del PLL sin preocuparse por las características del LPF.

Con la siguiente ecuación de control digital se implementa el PI:

$$y_{lf}[n] = y_{lf}[n - 1] * A1 + y_{notch}[n] * B0 + y_{notch}[n - 1] * B1$$

Al pasar del dominio del tiempo discreto a la frecuencia con la transformada Z la ecuación de control digital se escribe como:

$$\frac{y_{lf}(z)}{y_{notch}(z)} = \frac{B0 + B1 * Z^{-1}}{1 - Z^{-1}}$$

Ahora la función de transferencia del controlador PI es:

$$\frac{y_{lf}(s)}{y_{notch}(s)} = K_p + \frac{K_i}{s}$$

Y al utilizar la transformada bilineal:

$$s = \frac{2}{T} \left( \frac{z - 1}{z + 1} \right)$$

Se obtiene:

$$\frac{y_{lf}(z)}{y_{notch}(z)} = \frac{\left( \frac{2 * K_p + K_i * T}{2} \right) - \left( \frac{2 * K_p - K_i * T}{2} \right) z^{-1}}{1 - z^{-1}}$$

Siendo T el tiempo de muestreo.

Por otra parte, en esta ecuación general de segundo orden:

$$H(s) = \frac{w_n^2}{s^2 + 2\zeta w_n s + w_n^2}$$

La respuesta al paso es:

$$y(t) = 1 - ce^{\sigma t} \sin(w_d t + \phi)$$

Si se ignora el cero del semiplano izquierdo. Ahora el tiempo de establecimiento ( $t_s$ ) viene dado por el tiempo que toma la respuesta en estabilizarse entre una banda de error, implantando este error como:  $\partial$  se tiene:

$$1 - \partial = 1 - ce^{\sigma t_s}$$

$$\partial = ce^{\sigma t_s}$$

$$t_s = \frac{1}{\sigma} * \ln\left(\frac{c}{\partial}\right)$$

Donde:

$$\sigma = \zeta w_n$$

$$c = \frac{w_n}{w_d}$$

$$w_d = \sqrt{1 - \zeta^2} w_n$$

El diseño del filtro pasa banda al discretizar la ecuación del filtro pasa banda en el dominio de s:

$$H_{nf}(s) = \frac{s^2 + 2\zeta_2 w_n s + w_n^2}{s^2 + 2\zeta_1 w_n s + w_n^2}$$

Donde:

$$\zeta_2 \ll \zeta_1$$

Usando orden cero:

$$s = \frac{(z - 1)}{T}$$

Se obtiene:

$$H_{nf}(z) = \frac{z^2 + (2\zeta_2 w_n T - 2)z + (-2\zeta_2 w_n T + w_n^2 T^2 + 1)}{z^2 + (2\zeta_1 w_n T - 2)z + (-2\zeta_1 w_n T + w_n^2 T^2 + 1)} = \frac{B_0 + B_1 z^{-1} + B_2 z^{-2}}{A_0 + A_1 z^{-1} + A_2 z^{-2}}$$



Los coeficientes del filtro pasa banda se pueden cambiar de forma adaptativa al variar la frecuencia de red. Los coeficientes se eligen de manera que atenúen al doble la frecuencia de red sin afectar a las otras frecuencias.

### 3.3 PLL Trifásico.

Entre las técnicas de sincronización con la red eléctrica trifásica utilizadas y tratadas en esta memoria se mencionan:

- SRF-PLL: Marco de referencia síncrono – PLL.
- DSRF-PLL: Doble marco de referencia síncrono – PLL.
- DSOGI-PLL: Doble Integrador Generalizado de Segundo Orden – PLL.
- EPLL: Bucle de seguimiento de fase mejorado.
- SPLL-SRF para sistemas trifásicos de la librería SolarLib C28x.

#### 3.3.1 SRF-PLL: Marco de referencia síncrono – PLL.

La correcta operación de convertidores electrónicos de potencia conectados a la red necesita una técnica de sincronización para medir la fase de tensión de la red eléctrica. El rendimiento de esta técnica de sincronización está relacionado con la calidad de energía consumida o entregada. Una técnica de sincronización robusta y simple es la SRF-PLL.

Esta técnica de sincronización con la red eléctrica es usada en sistemas de tres fases, está basada en la proyección de las tensiones de la red eléctrica en un marco de referencia síncrono dq como ilustra la figura 3.7. El lazo de control incluye un regulador PI y un oscilador controlado por voltaje representado por un integrador que permitirá alcanzar en régimen permanente la referencia.

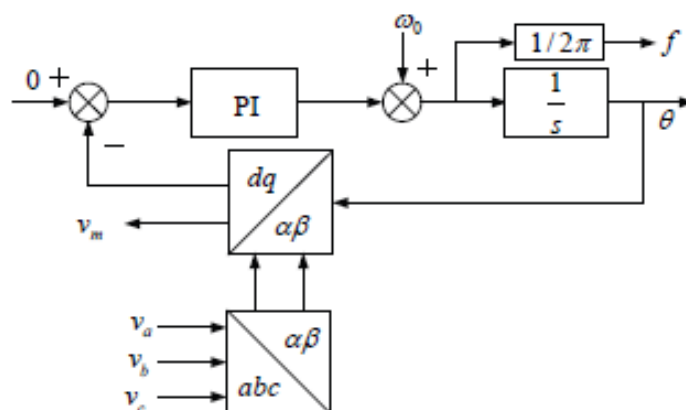


Figura 3.7 Diagrama de bloques de un SRF-PLL

Xiao-Qiang GUO, Wei-Yang WU, He-Rong GU [11]

### 3.3.2 DSRF-PLL: Doble marco de referencia síncrono – PLL.

Esta técnica de sincronización para los convertidores conectados a la red utiliza un desacoplamiento para separar las componentes positivas y negativas de frecuencia fundamental de la secuencia del vector de voltaje para lograr un mejor rendimiento bajo condiciones de desequilibrio. La figura 3.8 muestra su diagrama de bloque.

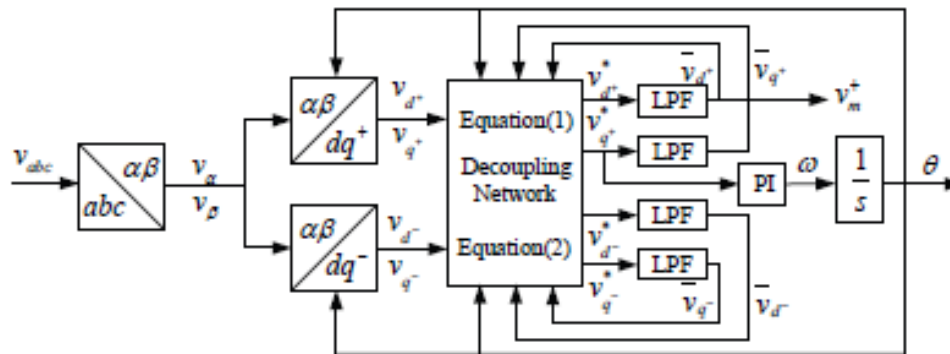


Figura 3.8 Diagrama de bloques de un DSRF-PLL

Xiao-Qiang GUO, Wei-Yang WU, He-Rong GU [11]

### 3.3.3 DSOGI-PLL: Doble Integrador Generalizado de Segundo Orden – PLL.

Esta técnica extrae la secuencia positiva fundamental para el SRF-PLL, pero utiliza un doble integrador generalizado de segundo orden para crear el generador de señales en cuadratura, esta técnica de sincronización se usa en los convertidores conectados a la red eléctrica.

A continuación, la figura 3.9 muestra el diagrama de bloque:

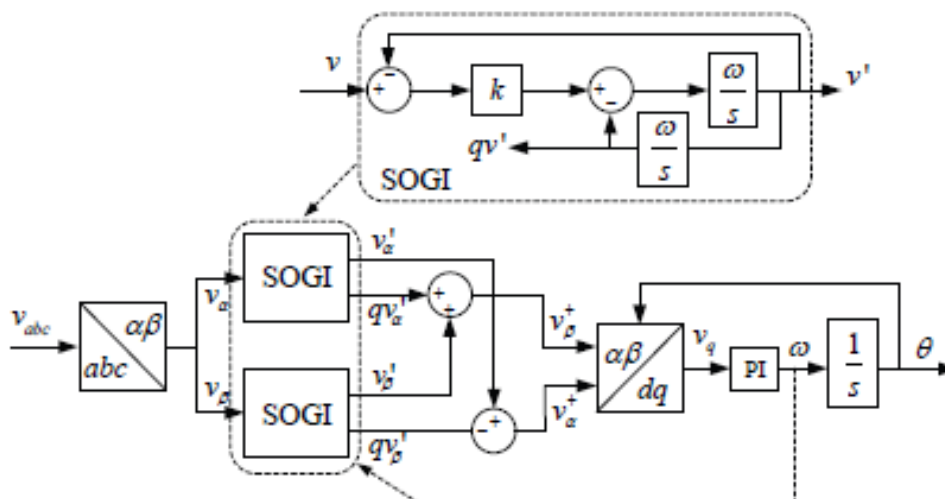


Figura 3.9 Diagrama de bloques de un DSOGI-PLL

Xiao-Qiang GUO, Wei-Yang WU, He-Rong GU [11]

### 3.3.4 EPLL: Bucle de seguimiento de fase mejorado.

Técnica de sincronización usada en los convertidores conectados a la red eléctrica, es un método de sincronización no lineal de frecuencia adaptativa. Su mejora con respecto al PLL convencional es que su detector de fase proporciona más información tal como la amplitud y ángulo de fase. La figura 3.10 y 3.11 muestran el diagrama de bloque monofásico y trifásico del EPLL.

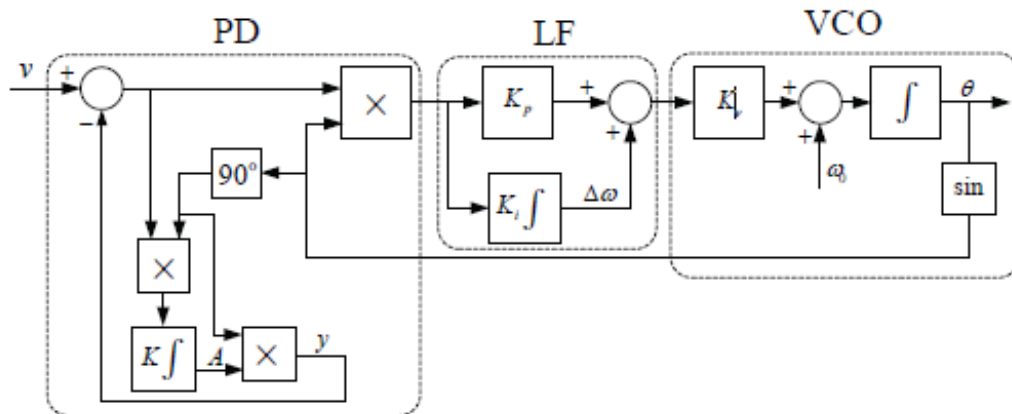


Figura 3.10 Diagrama de bloque de EPLL monofásico

Xiao-Qiang GUO, Wei-Yang WU, He-Rong GU [11]

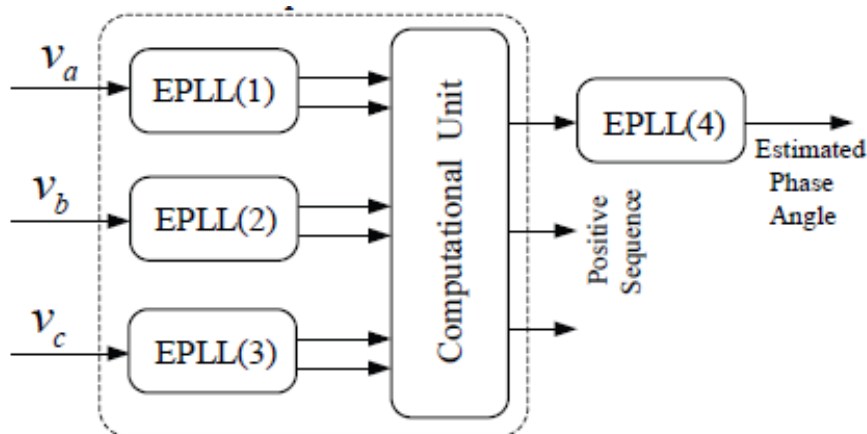


Figura 3.11 Diagrama de bloque de EPLL trifásico

Xiao-Qiang GUO, Wei-Yang WU, He-Rong GU [11]

### 3.3.5 SPLL-SRF para sistemas trifásicos de la librería SolarLib C28x.

Este módulo implementa un SPLL-SRF basado en un sistema de referencia síncrono para una red trifásica.

La figura 3.12 indica que el ingreso al módulo es  $.V_q$ ; sus salidas son:  $.\sin(\theta)$ ,  $.\cos(\theta)$  y  $(\theta)$ ; un parámetro interno del módulo es  $f_n$ .

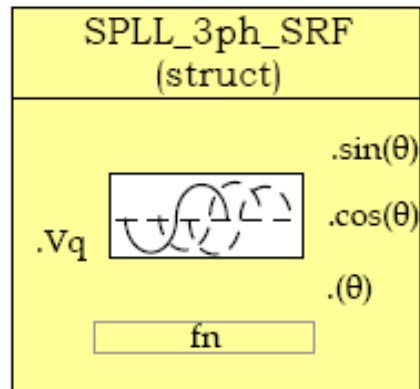


Figura 3.12

Texas Instruments. C28x Solar Library [14]

Este módulo transforma un sistema trifásico de alterna variante en el tiempo a otro sistema de corriente continua en un marco de referencia de rotación con la ayuda de transformadas.

Asumiendo que la siguiente ecuación de tres fases y frecuencia  $w$  tiene la secuencia:  $V_a \rightarrow V_b \rightarrow V_c$ :

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = V \begin{bmatrix} \cos(wt) \\ \cos\left(wt - \frac{2\pi}{3}\right) \\ \cos\left(wt - \frac{4\pi}{3}\right) \end{bmatrix}$$

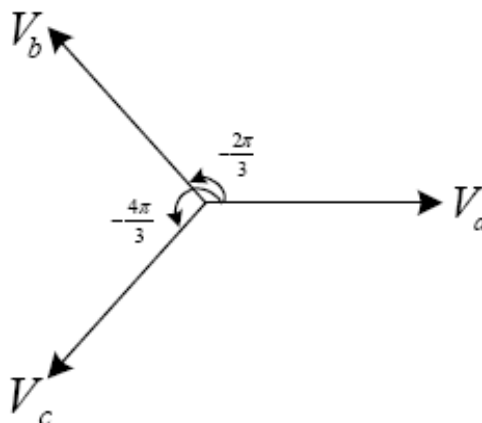


Figura 3.13 Secuencia de señal de tres fases y frecuencia  $w$

Texas Instruments. C28x Solar Library [14]

La primera transformación es la denominada transformada de Clark la cual reduce las tres magnitudes del sistema trifásico a un sistema de componente ortogonal también denominado marco de referencia estacionario:

$$V_{\alpha\beta 0} = T_{abc \rightarrow \alpha\beta 0} V_{abc}$$

$$\begin{bmatrix} V_{\alpha} \\ V_{\beta} \\ V_0 \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & \cos(2\pi/3) & \cos(4\pi/3) \\ 0 & \sin(2\pi/3) & \sin(4\pi/3) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \times \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\omega t) \\ \sin(\omega t) \\ 0 \end{bmatrix}$$

Ahora se asume que el vector de tensión de la red  $V_{net}$  está a un ángulo  $\theta$  con el marco de referencia ortogonal y rotando a frecuencia  $\omega$ . Entonces la segunda transformación es proyectar las componentes ortogonales en el marco de referencia giratorio logrando pasar a un sistema de corriente continua:

$$V_{dq0} = T_{\alpha\beta 0 \rightarrow dq0} V_{\alpha\beta 0}$$

$$\begin{bmatrix} V_d \\ V_q \\ V_0 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} V_{\alpha} \\ V_{\beta} \\ V_0 \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\omega t) \\ \sin(\omega t) \\ 0 \end{bmatrix}$$

$$= \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta) \cos(\omega t) + \sin(\theta) \sin(\omega t) \\ -\sin(\theta) \cos(\omega t) + \cos(\theta) \sin(\omega t) \\ 0 \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\omega t - \theta) \\ \sin(\omega t - \theta) \\ 0 \end{bmatrix}$$

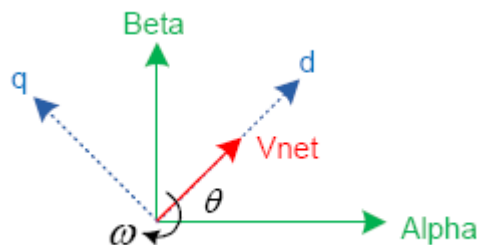


Figura 3.14 Proyección de las componentes ortogonales en el marco de referencia giratorio

Texas Instruments. C28x Solar Library [14]

Para que el PLL funcione correctamente o sea  $\theta$  se igual a  $\omega t$  la componente en cuadratura se linealiza de la siguiente forma:

$$V_q \approx (\omega t - \theta)$$

El análisis anterior indica que la componente en  $q$  es cero para un sistema trifásico equilibrado, entonces la componente  $q$  se utiliza como señal de error para el PLL para realizar el seguimiento a la fase apropiada.

En resumen, las tres fases se transforman en un marco de referencia giratorio y la componente  $q$  es utilizada para detectar la fase. Un filtro pasa bajo es necesario para eliminar el error en

estado estacionario y la salida de este filtro se alimenta a un VCO que genera el ángulo y las cantidades del seno.

La figura 3.15 muestra el diagrama funcional del PLL formado por un detector de fase aprovechando la transformada de Park, un filtro pasa bajo LPF que se encarga de eliminar el error en estado estacionario y un VCO que calcula el valor del ángulo y los valores del seno.

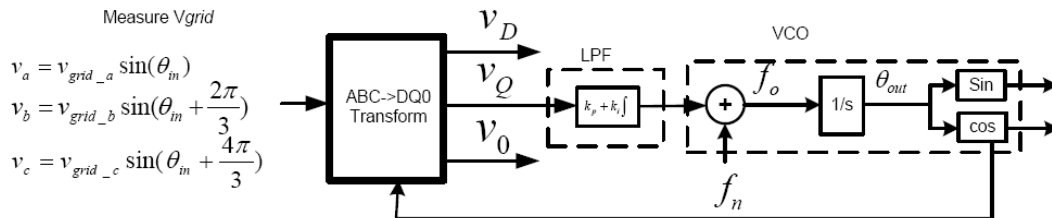


Figura 3.15

Texas Instruments. C28x Solar Library [14]

En el capítulo 5 se implementa el SPLL en cuadratura de la señal de entrada desplazada 90 grados, el SPLL monofásico de la librería SolarLib C28x y el SPLL-SRF trifásico de la librería SolarLib C28x

## Capítulo 4.

### Procesador digital de señales.

#### 4.1 Introducción.

Para procesar señales digitalmente son utilizados los procesadores digitales de señales o DSP por sus siglas en inglés Digital Signal Processor, el procesamiento digital de señales es la manipulación matemática de una señal de información para optimizarla o cambiarla en algún sentido.

La figura 4.1 muestra los módulos de un DSP:

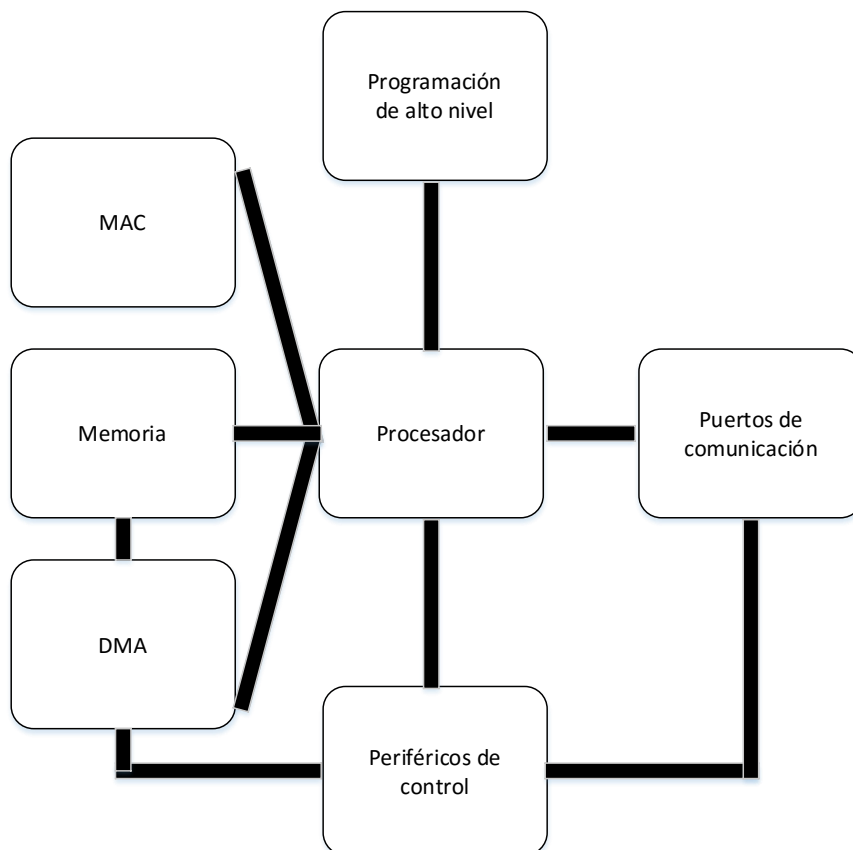


Figura 4.1 Módulos de un DSP

Un DSP tiene programación de alto nivel porque es un dispositivo programable por instrucciones a diferencia que un FPGA por sus siglas en inglés Field Programmable Gate Array que es programable a bajo nivel ya que es un dispositivo programable por circuito.

La gran diferencia que existe entre los DSP's con otros dispositivos como los microcontroladores o FPGA's son los módulos MAC y DMA. Estos dos módulos aceleran el procesamiento de señales.

El modulo multiplicador y acumulador o módulo MAC que en inglés es Multiplier-Accumuladores es una unidad dedicada a implementar por Hardware la multiplicación y acumulación de los algoritmos de un DSP, el modulo acceso directo a memoria o módulo DMA por sus iniciales en inglés Direct Memory Access lleva la información de entrada a la memoria RAM para que el procesador trabaje en los algoritmos, una vez finalizado el trabajo del procesador el DMA toma los resultados del trabajo y los envía a los periféricos de salida, con esto el DMA independiza el trabajo del movimiento de datos tanto de ingreso como de salida al procesador.

Los algoritmos en que trabaja un DSP se puede observar en la figura 4.2

Algoritmo	Ecuación
Filtro de respuesta al impulso finita	$y(n) = \sum_{k=0}^M a_k x(n-k)$
Filtro de respuesta al impulso infinita	$y(n) = \sum_{k=0}^M a_k x(n-k) + \sum_{k=1}^N b_k y(n-k)$
Convolución	$y(n) = \sum_{k=0}^N x(k)h(n-k)$
Transformada discreta de Fourier	$X(k) = \sum_{n=0}^{N-1} x(n) \exp[-j \left(\frac{2\pi}{N}\right) nk]$
Transformada coseno discreta	$F(u) = \sum_{x=0}^{N-1} c(u) \cdot f(x) \cdot \cos\left[\frac{\pi}{2N} u(2x+1)\right]$

Figura 4.2 Algoritmos para un DSP

Francisco Gimeno, Salvador Orts, Salvador Seguí [15]

Los DSP's pueden estar orientados a control, potencia, video, comunicaciones entre otros.

Texas Instruments divide a sus DSP's en tres series: Serie C2000, serie C5000 y serie C6000.



La serie C2000 está orientada a control por esto a estos dispositivos se los suele llamar DSC que significa Controlador Digital de Señales, estos dispositivos tienen baja velocidad, pero tienen una gran cantidad de periféricos.

La serie C5000 está orientada a bajo consumo de potencia, pero es más veloz, esta serie puede ser utilizada en audio.

La serie C6000 es la más potente en cuanto a procesamiento y velocidades, aunque tiene pocos periféricos, generalmente se dedica a recibir la señal por un puerto, procesarla y entregar sus resultados por otro puerto lo más rápido posible.

#### **4.2 Controlador digital de señales.**

Los controladores digitales de señales DSC por sus siglas en inglés son de gran utilidad en sistemas empujados a su vez los sistemas empujados son sistemas que son controlados por un computador empujado en el mismo medio.

El C2000 de Texas Instruments es un dispositivo con gran cantidad de periféricos de alto desempeño integrados para aplicaciones de control en tiempo real. Su núcleo optimizado para hacer cálculos matemáticos ayuda a los diseñadores para mejorar la flexibilidad, confianza y eficiencia del sistema.

Se estudia el TMS320F28027 de la familia C2000 de Texas Instruments

#### **4.3 Arquitectura del DSC TMS320F28027 de Texas Instruments.**

La figura 4.3 muestra el diagrama de bloques del procesador.

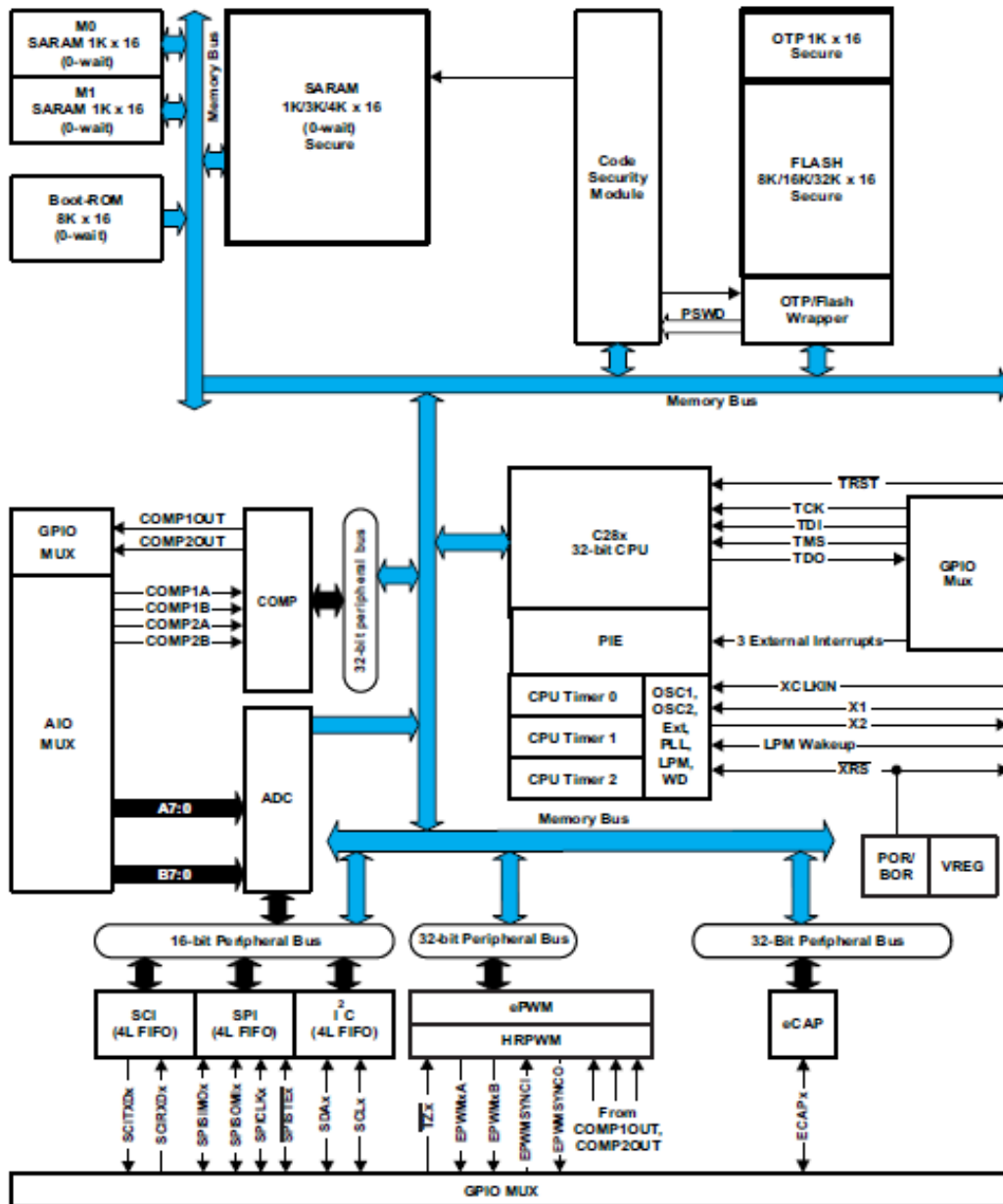


Figura 4.3 Diagrama de bloques TMS320F28027

Texas Instruments [16]

Básicamente el diagrama de bloques contiene:

Entre los periféricos del procesador se tienen: ePWM (modulación de ancho de pulso), eCAP (módulo de captura), ADC, WD (perro guardián), protocolo I2C, puerto SCI y el estándar SPI.

Una CPU con registros de 32 bits.

A más el procesador contiene 3 temporizadores, la memoria ROM encargada de mantener la información cuando se desenergiza el DSP, una memoria RAM y una memoria flash más lenta, pero con mayor capacidad respecto a la memoria RAM.

#### **4.4 Software para DSC TMS320F28027 de Texas Instruments.**

Texas Instruments proporciona el Code Composer Studio como software para este controlador digital de señales, este software de programación integrado cuenta con todas las herramientas para generar código hasta obtener el punto ejecutable en un DSP.

El Code Composer Studio cuenta con un editor de texto pasando luego a un compilador que revisa la sintaxis y el resultado es llevado al ensamblador que es el encargado de pasarlo a lenguaje de máquina y finalmente a un enlazador que enlaza todos los archivos para obtener el archivo .exe que es el equivalente del archivo ejecutable en Windows, también se cuenta con herramientas de gráficos y librerías.

Además, Code Composer Studio cuenta con un simulador de código utilizado en casos en que no se tenga una tarjeta de aprendizaje, herramientas de aprendizajes denominadas eZdsp y un emulador externo.

#### **4.5 Configuración para DSC TMS320F28027.**

Para configurar el dispositivo se utiliza:

- Mapa de memoria.
- Archivo de comando enlazador.
- Archivos de cabecera de los registros de periféricos.
- Inicialización del procesador.
- Inicialización del sistema.

##### **4.5.1 Mapa de memoria.**

La figura 4.4 indica las diferentes direcciones de los elementos del procesador.

	Data Space	Prog Space
0x00 0000	<i>M0 Vector RAM (Enabled if VMAP = 0)</i>	
0x00 0040	M0 SARAM (1K x 16, 0-Wait)	
0x00 0400	M1 SARAM (1K x 16, 0-Wait)	
0x00 0800	Peripheral Frame 0	Reserved
0x00 0D00	PIE Vector - RAM (256 x 16) (Enabled if VMAP = 1, ENPIE = 1)	
0x00 0E00	Peripheral Frame 0	
0x00 2000	Reserved	
0x00 6000	Peripheral Frame 1 (4K x 16, Protected)	Reserved
0x00 7000	Peripheral Frame 2 (4K x 16, Protected)	
0x00 8000	L0 SARAM (4K x 16) (0-Wait, Secure Zone + ECSL, Dual Mapped)	
0x00 9000	Reserved	
0x3D 7800	User OTP (1K x 16, Secure Zone + ECSL)	
0x3D 7C00	Reserved	
0x3D 7C80	Calibration Data	
0x3D 7CC0	Get_mode function	
0x3D 7CE0	Reserved	
0x3D 7E80	Calibration Data	
0x3D 7EB0	Reserved	
0x3D 7FFF	PARTID	
0x3D 8000	Reserved	
0x3F 0000	FLASH (32K x 16, 4 Sectors, Secure Zone + ECSL)	
0x3F 7FF8	128-Bit Password	
0x3F 8000	L0 SARAM (4K x 16) (0-Wait, Secure Zone + ECSL, Dual Mapped)	
0x3F 9000	Reserved	
0x3F E000	Boot ROM (8K x 16, 0-Wait)	
0x3F FFC0	<i>Vector (32 Vectors, Enabled if VMAP = 1)</i>	

Figura 4.4

Texas Instruments [16]

Se describe brevemente el mapa de memoria con respecto a la figura 4.4 aclarando que las direcciones se encuentran en numeración hexadecimal:

M0SARAM y M1SARAM se ubican en las 2048 primeras direcciones y se utilizan tanto para programas y para datos.

Los vectores de interrupción o PIE Vectors van de la dirección 0x000D00 a la dirección 0x000E00, en estas direcciones se guardan las funciones que atienden la interrupción.

La trama de periférico 0 o PeripheralFrame0 en inglés se encuentran entre las direcciones 0x000E00 y 0x002000. En estas direcciones se escriben los registros de control de los periféricos.

El bloque de la zona LO SARAM es utilizado para programa y para datos. La figura también muestra un espejo de este bloque entre las direcciones 0x3F8000 y 0x3F9000

Se tiene varios conjuntos de direcciones reservadas, uno de ellos va desde la dirección 0x009000 hasta la 0x3D7800

Memoria User OTP de 1000 localidades.

La memoria Flash cuenta con 32000 localidades.

El código de seguridad o PASSWORDS que es un grupo de localidades donde se escribe una contraseña para acceder al código.

Finalmente se tiene las direcciones 0x3FE000 hasta 0x3FFFC0 de las subrutinas de inicialización que están en la memoria de reinicio ROM y los vectores que apuntan estas subrutinas en las direcciones siguientes a la 0x3FFFC0

#### **4.5.2 Archivo enlazador de comandos.**

El archivo enlazador de comandos describe el nombre, la localización y el tamaño del mapa de memoria, el Code Composer Studio en un proyecto genera automáticamente un archivo .cmd que es el archivo enlazador de comandos.

#### **4.5.3 Archivos de cabecera de los registros de periféricos.**

Estos archivos son proveídos por Texas Instruments para poder desarrollar código en C, es una forma fácil de desarrollar código ya que se tiene unas estructuras que contiene todos los elementos y todos los registros de control pudiendo acceder a través de estas estructuras a los campos de bit de los registros.

#### **4.5.4 Inicialización del procesador.**

##### **Reset.**

Este dispositivo tiene dos fuentes de reinicio el perro guardián y una terminal física. En la terminal física hay reinicio simplemente al pasar de un estado de alto a bajo mientras que el perro guardián es un temporizador que al desbordarse reinicia el dispositivo.

## Interrupciones.

El CPU debe atender las interrupciones asociadas a eventos de los periféricos ePWM, eCAP, entre otros y el total de interrupciones que se pueden generar por los periféricos y los temporizadores son algo más de 96 sin embargo el CPU consta solamente de 16 interrupciones  $\overline{XRS}$ ,  $NMI$ ,  $INT1$ ,  $INT2$ , ...,  $INT14$ , entonces ¿cómo se puede mapear más de 96 interrupciones al procesador que contiene solamente 16 interrupciones?, para solucionar ese inconveniente se necesita un multiplexor llamado PIE por sus siglas en inglés Peripheral Interrupt Expansion como indica la figura 4.5

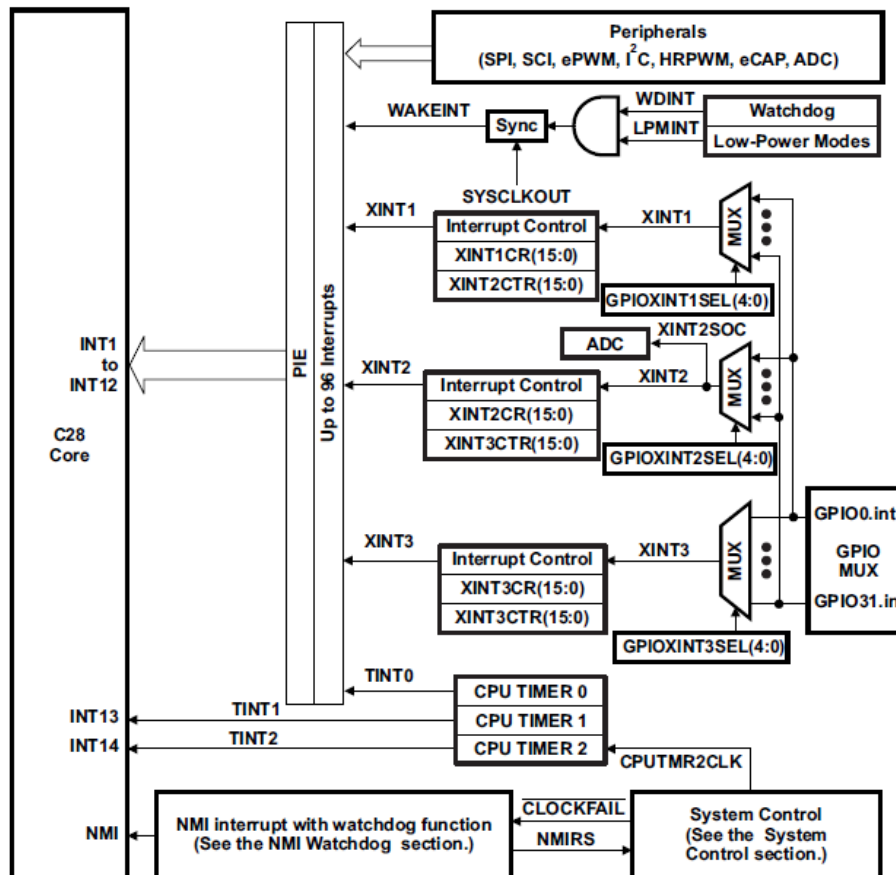


Figura 4.5 Fuentes de interrupción

Texas Instruments [16]

### 4.5.5 Inicialización del sistema.

Generalmente la inicialización del sistema indica la frecuencia del reloj, el estado de las terminales al iniciar el dispositivo, los estados de las interrupciones y la habilitación o deshabilitación del perro guardián.

## Capítulo 5.

### Implementación de software PLL para señales sinusoidales monofásicas y trifásicas como entrada usando el entorno de desarrollo Code Composer Studio y la plataforma LAUNCHXL-F28027 (C2000).

#### 5.1 Introducción.

En este capítulo se muestran los datos experimentales con sus flujogramas de los distintos códigos implementados.

#### 5.2 Implementación de PLL monofásico en cuadratura con señal de ingreso generada con software.

Esta prueba experimental es implementada en LaunchPad C2000 con el TMS320F28027 de Texas Instruments. El archivo se llama `pll_dq_funcionseno_monofasico.c`, en el buffer nombrado `buffer_entrada` del archivo se tiene como entrada una señal sinusoidal de amplitud -1 a 1 voltio a 50 Hz que es generada con la función `seno` de la librería `IQmath` y el desbordamiento del `timer2` a 78,125 us como periodo de muestreo. Por lo tanto, la frecuencia de muestreo es 12800 Hz y se construye cada onda sinusoidal con 256 muestras.

Para producir el desplazamiento  $\pi/2$  radianes basta con desplazar la señal sinusoidal 256/4 o 64 posiciones y finalmente se calcula el valor pico, el compensador PI e integrador.

Para el cálculo del valor pico de una señal muestreada se almacena el valor máximo en cada ciclo y se compara con las muestras del siguiente ciclo.

Para calcular el compensador PI se tiene presente lo que se dijo en el inciso 3.2.3:

Salida del controlador PI = variación proporcional de la magnitud del error + variación proporcional del tiempo en que ha permanecido el error.

Entonces en el archivo `pireg.c` que se ejecuta cada 78,125 us se tiene:

$$\text{Out} = (K_p)(\text{error}) + \text{integral}$$

Donde:

$$\text{integral} = \text{integral} + (K_i \cdot t_s) (\text{error})$$

De lo anterior se deduce fácilmente que `Out` es la salida del controlador PI,  $(K_p)(\text{error})$  es la variación proporcional de la magnitud del error y  $(K_i \cdot t_s) (\text{error})$  es la variación proporcional del tiempo en que ha permanecido el error.

Para que  $(K_i \cdot t_s) (\text{error})$  sea la variación proporcional del tiempo que permanece el error `Ki_ts` debe ser el tiempo de muestreo, o sea

$$K_i \cdot t_s = 7.8125E-5$$

La variable `integral` actualiza la variación proporcional del tiempo en que permanece el error.

El valor de `Kp` deducido experimentalmente es 2.

Ahora para el cómputo del integrador en el archivo `pll_dq_funcionseno_monofasico.c` se posee:

$$\text{Integrador} += (\text{Out} + \text{wff}) (7.8125\text{E-}5)$$

Al relacionar la ecuación anterior con la figura 3.3 se concluye que la salida del controlador PI es sumada con el máximo valor de la frecuencia angular que puede salir del regulador PI y este resultado es multiplicado por el tiempo de muestreo para obtener la fase de salida del PLL, la variable integrador se actualiza en cada desbordamiento del `timer2`.

La figura 5.1 muestra el diagrama de flujo del código implementado.

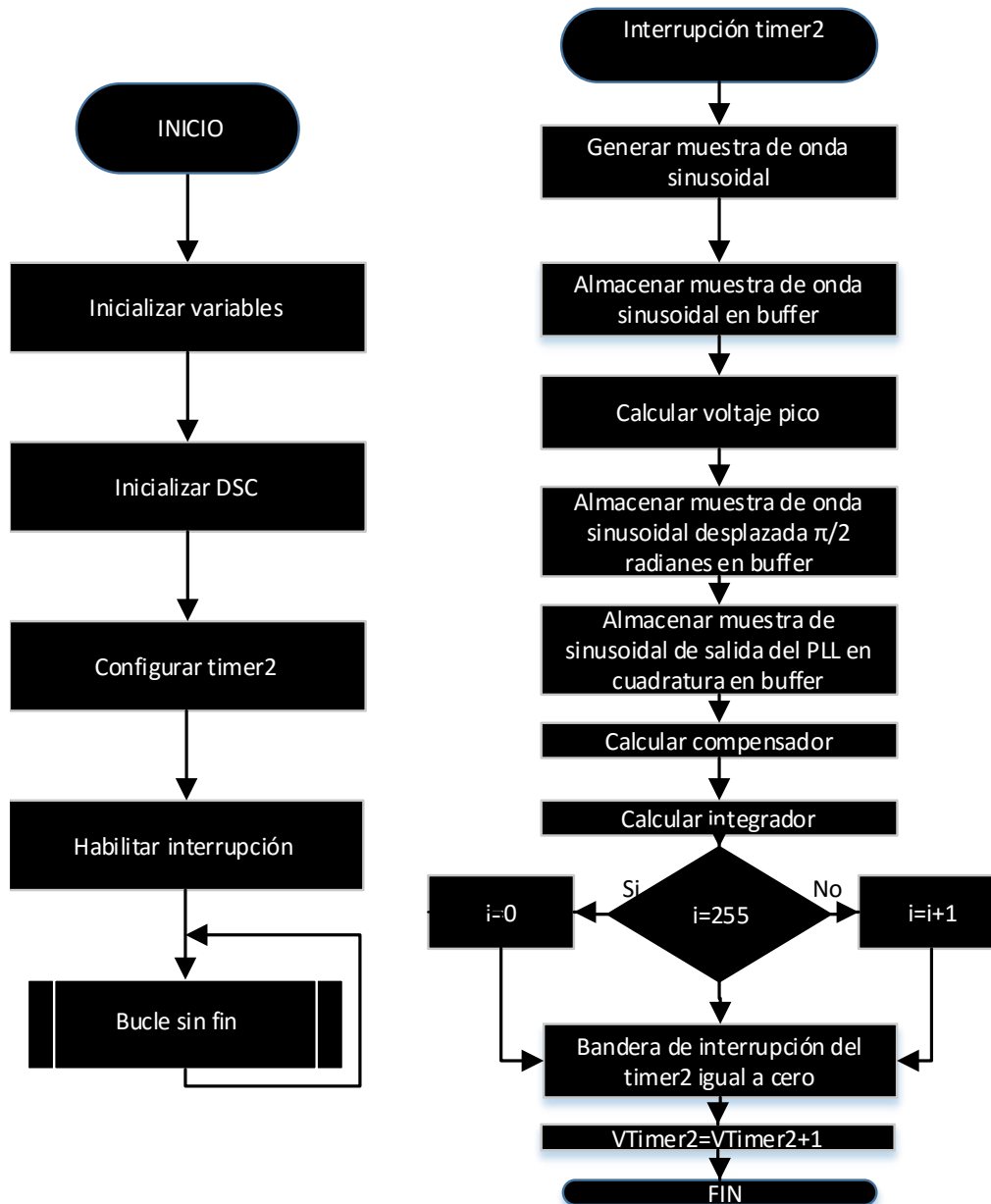


Figura 5.1

El buffer denominado `buffer_salida` muestra la señal que sigue a la señal de entrada en la figura 5.2. Tanto `buffer_entrada` como `buffer_salida` están en sistema de numeración IQ22.



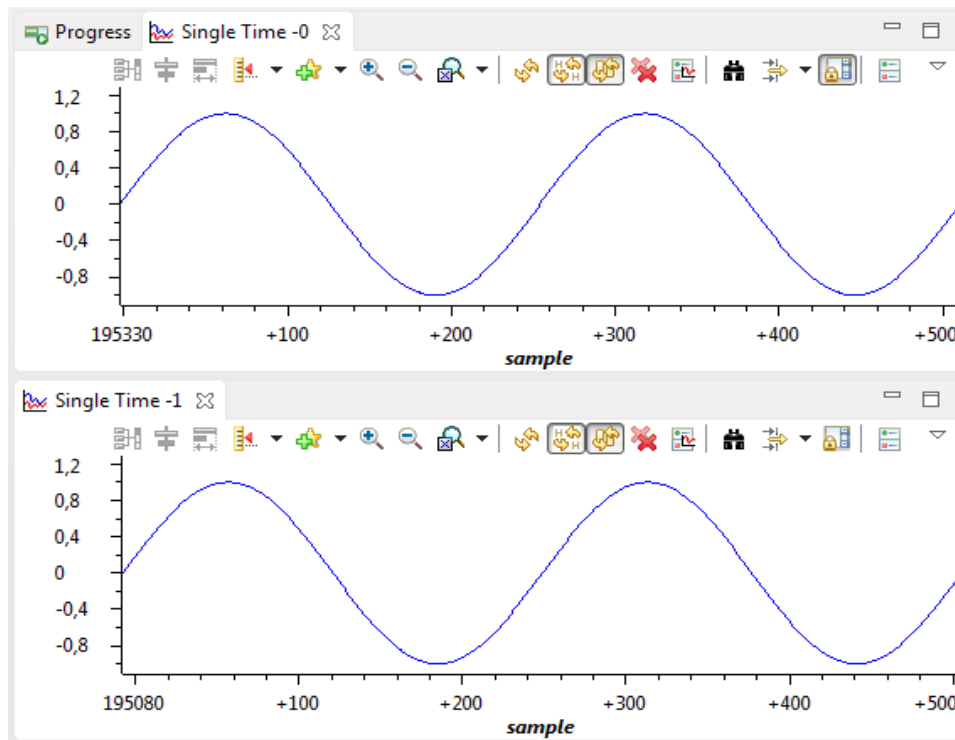


Figura 5.2 buffer\_entrada versus buffer\_salida

### 5.3 Implementación de PLL monofásico en cuadratura con señal de ingreso generada por generador de funciones.

El archivo se denomina pll\_dq\_generador\_monofasico y se implementa en LaunchPad C2000 con el TMS320F28027 de Texas Instruments. La entrada es una señal sinusoidal de 50 Hz con una amplitud pico a pico de 0 a 2.93 voltios creados por un generador de funciones y medida por el ADCINA0. La imagen capturada del osciloscopio de la señal de entrada es la figura 5.3:

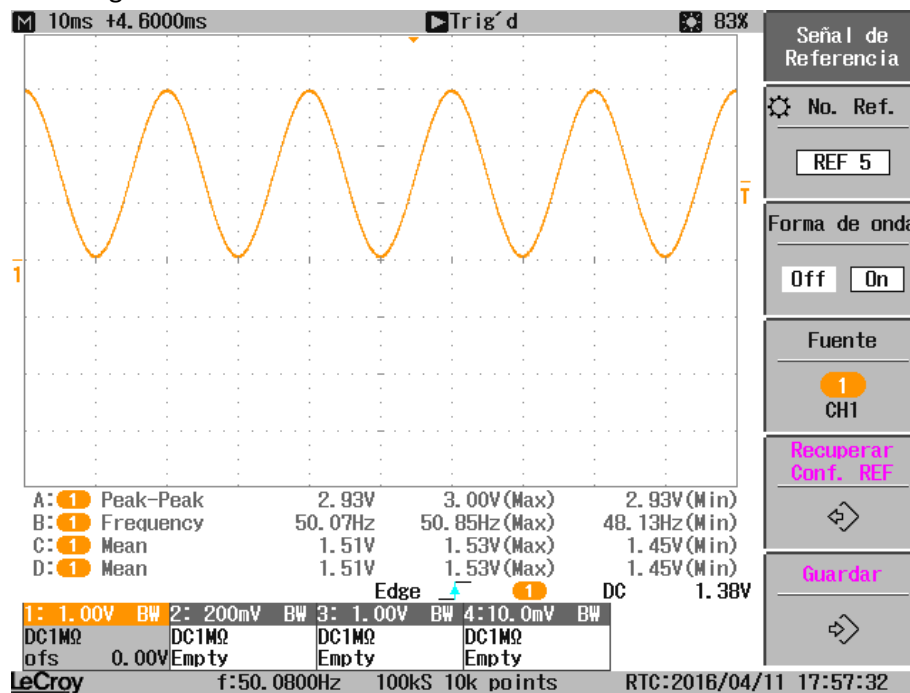


Figura 5.3

El TMS320F28027 soporta una señal de ingreso de 0 a 3.3 voltios sin embargo para evitar la saturación del ADC (convertidor analógico digital) se ingresa una tensión de 0 a 2.93 voltios.

Cada ciclo de señal sinusoidal tiene 256 muestras, por tanto, la frecuencia de muestreo de la señal es  $256 \cdot 50\text{Hz} = 12800\text{Hz}$ , se vale del desbordamiento del timer2 para generar este muestreo y este timer también es responsable del disparo del ADC que está configurado en forma continua. Para producir el desplazamiento  $\pi/2$  basta con desplazar la señal sinusoidal 256/4 o 64 posiciones y finalmente se calcula el valor pico, el compensador e integrador.

El cálculo del valor pico, compensador PI e integrador se especificó en el apartado precedente.

La figura 5.4 muestra el diagrama de flujo del código:

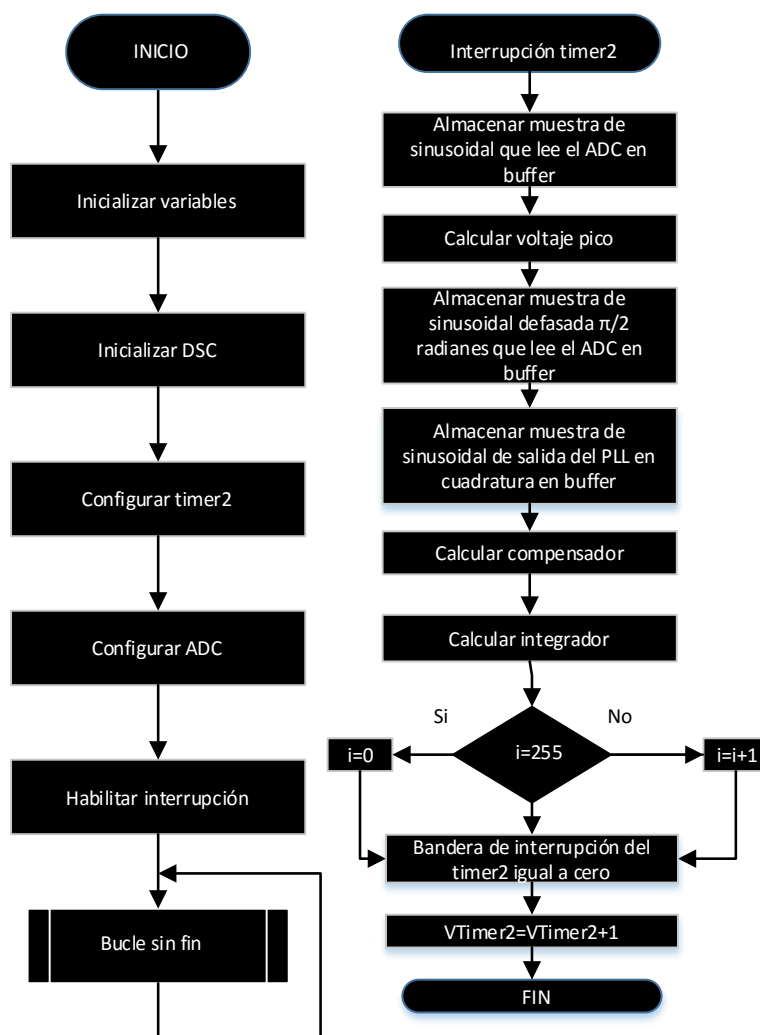


Figura 5.4

La señal de entrada del generador se muestra en el buffer llamado buffer y la señal de salida se visualiza en el buffer nombrado buffer\_salida de la figura 5.5. Los datos de buffer y buffer\_salida son en sistema de numeración IQ22.

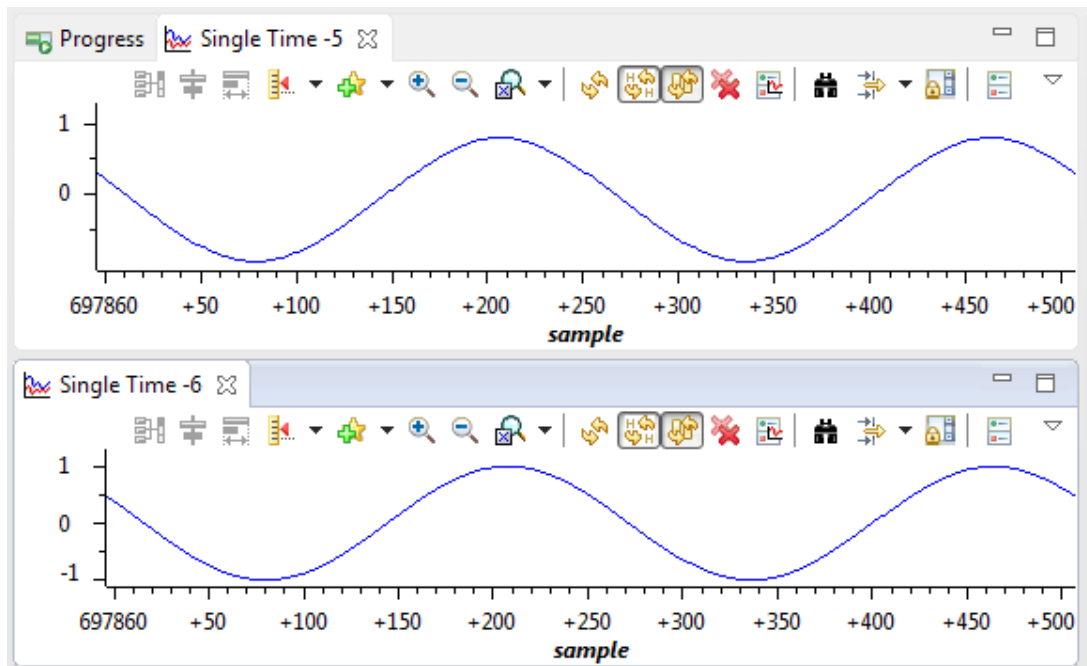


Figura 5.5 buffer\_entrada versus buffer\_salida.

#### 5.4 Implementación de SPLL para sistemas monofásicos de la librería SolarLib C28x con señal de ingreso generada con software.

El archivo ejecutado en el TMS320F28027 se denomina `spll_1ph_solar_funcionseno.c`. Se crea una señal monofásica de entrada a 50 Hz y con amplitud pico a pico de -1 a 1 voltio generada con la función seno de la librería `IQmath` y el desbordamiento del `timer2` a 50 us, esto es con frecuencia de muestreo de 20000Hz o sea que una señal sinusoidal se construye con 400 muestras, sin embargo por la pequeña memoria que posee el TMS320F28027 al operar, solo se visualizan muestras múltiplos de 4 de la frecuencia de muestreo en el entorno gráfico del Code Composer Studio, lo que traduce que la señal sinusoidal se visualiza con 100 muestras.

Realmente este código es una simulación ya que el `timer2` no desborda a  $1/20000$  segundos (50 us) si no lo hace a 10000 us, se lo hace a 10000 us porque el código se ejecuta en memoria FLASH ya que hay más capacidad de memoria que en RAM para almacenar en buffers las señales de ingreso y salida, sin embargo, tiene el inconveniente de que FLASH es más lenta y no da tiempo a ejecutarse la rutina comandada en el interior del `timer 2`.

La figura 5.6 muestra el diagrama de flujo del código.

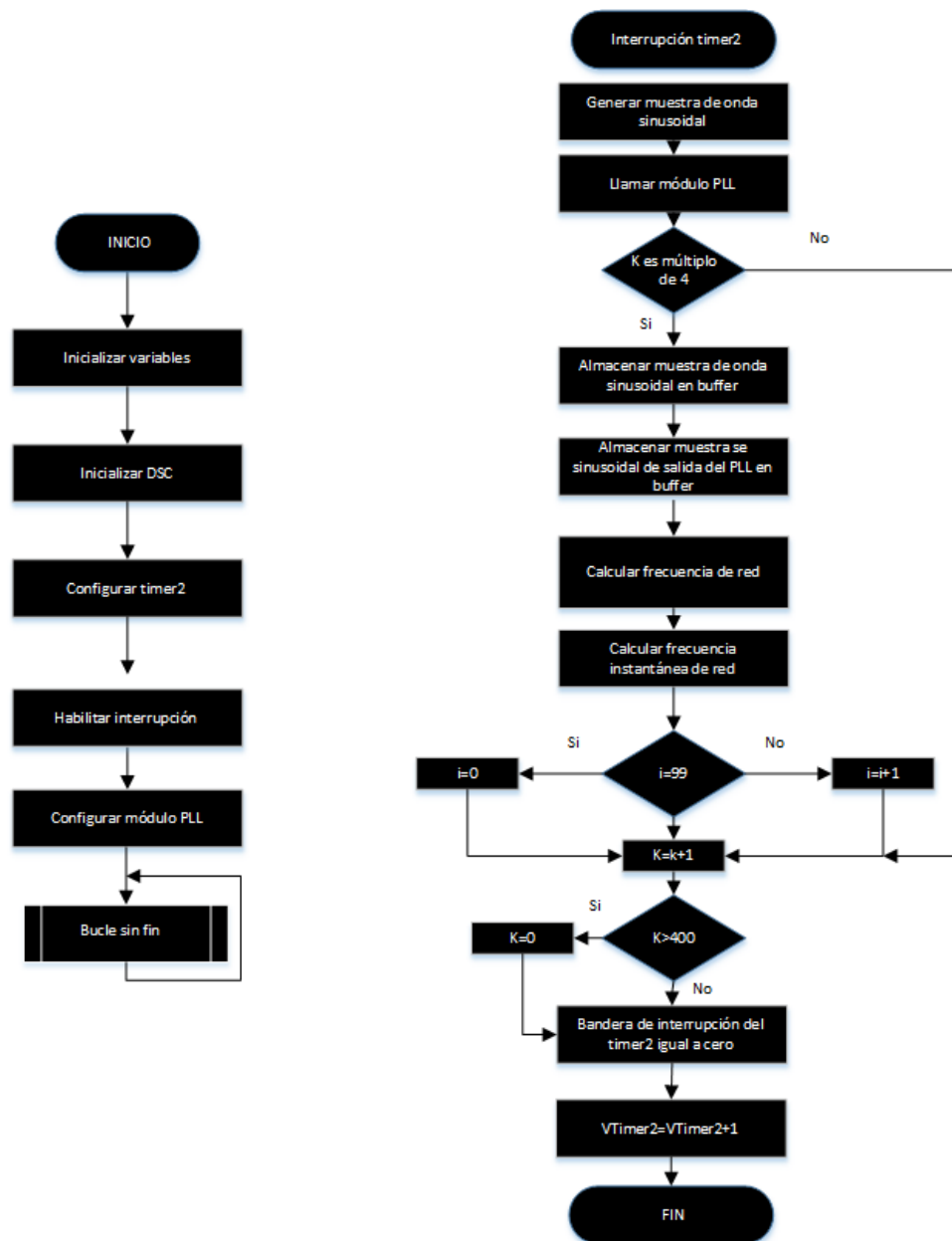


Figura 5.6

El proceso configurar módulo PLL del diagrama de flujo de la figura 5.6 se resume en dos pasos:

- Paso 1: Se incluye la librería "Solar\_IQ.h" en el proyecto creado en el Code Composer Studio.
- Paso 2: Se crea y se adhiere las estructuras `spll1` y `spll_notch_coef1` al módulo PLL en el archivo `spll_1ph_solar_funcionseno.c` del proyecto creado en el Code Composer Studio.

El proceso llamar módulo PLL del diagrama de flujo de la figura 5.6 se detalla en los siguientes pasos:

- Paso 1: Se inicializa el módulo PLL en el archivo `spll_1ph_solar_funcionseno.c` donde se especifica la entrada de frecuencia de red que en este caso es 50 Hz, la frecuencia de muestreo que se lo hace a 20000 Hz, la dirección del objeto PLL, se inicializa todas las variables y buffers de datos internos y se establece los coeficientes del filtro pasa banda como 0,000000001 y 1 de acuerdo a la frecuencia de red.
- Paso 2: La medida de red que en esta implementación es una señal sinusoidal generada por software es convertida de IQ24 a IQ21, y finalmente se llama a la función SPLL.

A continuación se realizan los cálculos del SPLL para un tiempo de establecimiento ( $t_s$ ) de 30 milisegundos, el coeficiente de amortiguación ( $\zeta$ ) de 0,7 y el error de banda ( $\delta$ ) a 0,115 siguiendo el apartado 3.2.4.

Se calcula c:

$$c = \frac{1}{\sqrt{1 - \zeta^2}} = 1,40028008$$

Ahora se encuentra la atenuación ( $\sigma$ ):

$$\sigma = \frac{\text{Ln} \frac{c}{\delta}}{t_s} = 83,3165142$$

Entonces la frecuencia natural ( $w_n$ ) es:

$$w_n = \frac{\sigma}{\zeta} = 119,023592$$

Con las ecuaciones:

$$w_n = \sqrt{\frac{v_{grid} k_p}{T_i}}$$

$$\zeta = \sqrt{\frac{v_{grid} T_i k_p}{4}}$$

y asumiendo que  $v_{grid}$  es 1, encontramos la constante proporcional ( $k_p$ ) y el tiempo integral ( $T_i$ )

$$T_i = 0,01176237$$

$$k_p = 166,633028$$

Luego:

$$k_i = \frac{k_p}{T_i} = 14166,6154$$

Para una frecuencia de muestreo de 20000 Hz (periodo de muestreo (T) igual a 50 us) obtenemos los coeficientes  $B_0$  y  $B_1$  del filtro del bucle o LPF por sus siglas en inglés: Loop Filter

$$B_0 = \frac{2 * K_p + K_i * T}{2} = 166,987194$$

$$B_1 = \frac{2 * K_p - K_i * T}{2} = -166,27886$$

Finalmente, los coeficientes del filtro pasa banda obtenidos experimentalmente que hacen un mejor seguimiento de la señal de salida respecto a la señal de entrada son:

$$\zeta_2 = 0,000000001$$

Y

$$\zeta_1 = 1$$

La figura 5.7 muestra la señal de ingreso y la señal de seguimiento del SPLL indicada en los registros `buffer_q24` y `buffer_salida_q21` del archivo `spll_1ph_solar_funcionseno.c` respectivamente. Los datos de la señal de ingreso están en sistema de numeración IQ24 mientras de los datos de la señal de salida son en IQ21.

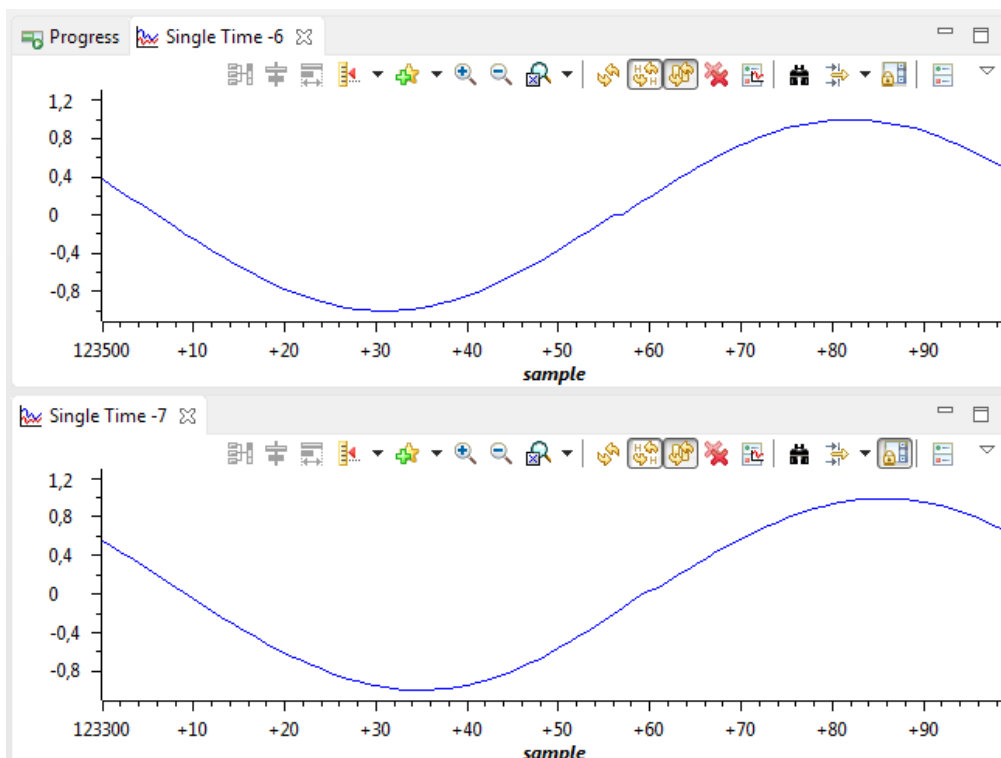


Figura 5.7 `buffer_q24` versus `buffer_salida_q21`

Ahora se visualiza en la figura 5.8 los registros `frecu_red` y `frecu_instan_red` en el Code Composer Studio que son la frecuencia de la sinusoidal generada por software y la frecuencia instantánea de la sinusoidal generada por software respectivamente en unidades de hertzios del archivo `spll_1ph_solar_funcionseno.c`

(*)- frecu_red	long	49.97467041 (Q-Value(21))	0x00008D2A@Data
(*)- frecu_instan_red	long	50.46025085 (Q-Value(21))	0x00008D2C@Data

Figura 5.8

En la simulación del código `frecu_instan_red` sigue a `frecu_red`.

### 5.5 Implementación de SPLL para sistemas monofásicos de la librería SolarLib C28x con señal de ingreso formada por el generador de funciones.

A la entrada se tiene una señal monofásica de 50 Hz con amplitud de 0 a 2.93 voltios creados por un generador de funciones. La señal de entrada capturada en el osciloscopio es la figura 5.9

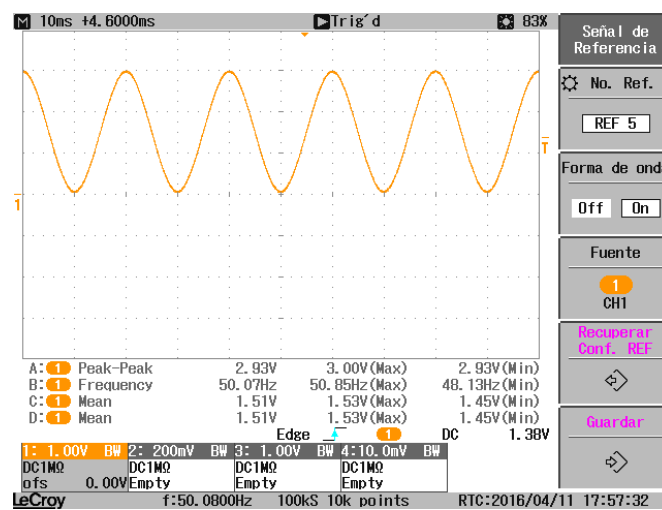


Figura 5.9

El TMS320F28027 soporta una señal de ingreso de 0 a 3.3 voltios sin embargo para evitar la saturación del ADC se ingresa una tensión de 0 a 2.93 voltios.

Con el timer2 la señal de ingreso se muestrea a 20000 Hz en el código, este timer también dispara el ADC que está configurado en forma continua. Después de configurar todos los parámetros que indica la librería SolarLib y ejecutar el código en el TMS320F28027 la figura 5.10 visualiza en el entorno gráfico del CodeComposer Studio la señal de ingreso y de salida del PLL monofásico almacenadas en los buffers llamados `buffer_q24_E` y `buffer_q21_S` respectivamente, los datos de la señal de ingreso están en sistema de numeración IQ24 mientras que los datos de la señal de salida son IQ21.

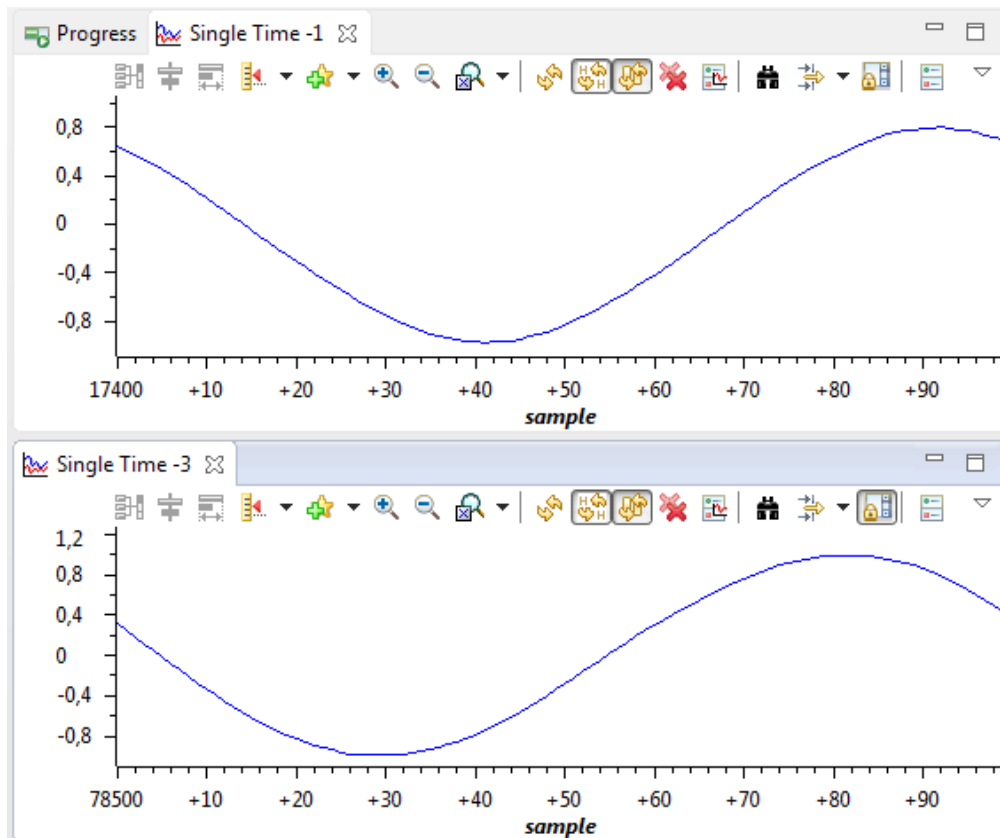


Figura 5.10 buffer\_q24\_E versus buffer\_q21\_S del archivo spll\_1ph\_solar\_generador.c

Muestreando a 20000 Hz una señal sinusoidal debe reconstruirse en 400 muestras (20000 muestras / 50 señales), sin embargo, la figura visualiza una señal reconstruida en 100 muestras, esto se debe a que en el software por cada cuatro muestras de la frecuencia de muestreo se visualiza una muestra en el Code Composer Studio, esto ayuda a suplir la poca memoria que posee el TMS320F28027 operando en memoria RAM.

La figura 5.11 adjunta el diagrama de flujo del código.



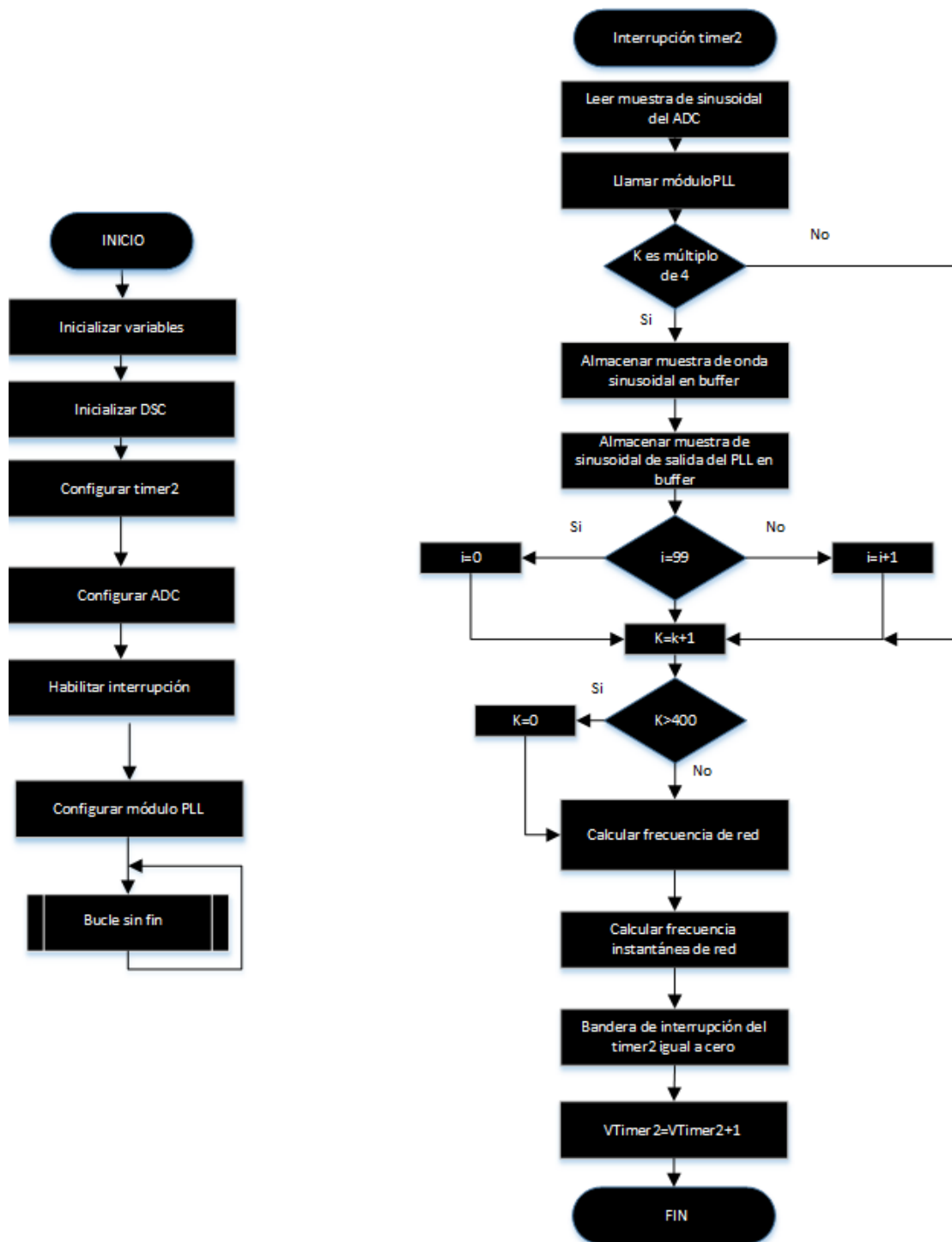


Figura 5.11

El proceso configurar módulo PLL del diagrama de flujo de la figura 5.11 se resume en dos pasos:

- Paso 1: Se incluye la librería "Solar\_IQ.h" en el proyecto creado en el Code Composer Studio.

- Paso 2: Se crea y se adhiere las estructuras `spll1` y `spll_notch_coef1` al módulo PLL en el archivo `spll_1ph_solar_generador.c` del proyecto creado en el Code Composer Studio.

El proceso llamar módulo PLL del diagrama de flujo de la figura 5.11 se detalla en los siguientes pasos:

- Paso 1: Se inicializa el módulo PLL en el archivo `spll_1ph_solar_generador.c` donde se especifica la entrada de frecuencia de red que en este caso es 50 Hz, la frecuencia de muestreo que se lo hace a 20000 Hz, la dirección del objeto PLL, se inicializa todas las variables y buffers de datos internos y se establece los coeficientes del filtro pasa banda como 0,000000001 y 1 de acuerdo a la frecuencia de red.
- Paso 2: La medida de red que en está implementación es una señal sinusoidal monofásica tomada de un generador de funciones es pasada de IQ24 a IQ21, y finalmente se llama a la función SPLL.

La figura 5.12 muestra una tabla con los parámetros del SPLL establecidos y calculados detallados en el apartado anterior:

	Parámetros	Valores
Parámetros establecidos	Tiempo de establecimiento ( $t_s$ )	30 ms
	Coficiente de amortiguación ( $\zeta$ )	0,7
	Error de banda ( $\theta$ )	0,115
Parámetros calculados	c	1,40028008
	Atenuación ( $\sigma$ )	83,3165142
	Frecuencia natural ( $w_n$ )	119,023592
	Amplitud de tensión de señal de entrada ( $v_{grid}$ )	1
	Tiempo integral ( $T_i$ )	0,01176237 s
	Constante proporcional ( $k_p$ )	166,633028
	Constante integral ( $k_i$ )	14166,6154
Parámetros establecidos	Frecuencia de muestreo	20000 Hz
	Periodo de muestreo (T)	50 us
Parámetros calculados	Coficientes del LPF	$B_0 = 166,987194$ $B_1 = -166,27886$
Parámetros establecidos	Coficientes del filtro pasa banda	$\zeta_2 = 0,000000001$ $\zeta_1 = 1$

Figura 5.12

## 5.6 Implementación SPLL\_3ph\_SRF para sistemas trifásicos con la librería SolarLib con señales de ingreso generadas por software.

El archivo `SRF_PLL_trifasico_funcionseno.c` crea tres señales trifásicas a 50 Hz con amplitud de +1 a -1 voltio desfasadas 120 grados.

Una señal sinusoidal se construye con la librería IQmath en 399 muestras que al multiplicar por 50 señales se obtiene una frecuencia de muestreo de 19950 Hz. Las otras dos señales sinusoidales desplazadas  $2\pi/3$  se construyen simplemente desplazando 133 y 266 posiciones.

Realmente este código es una simulación ya que el timer2 no desborda a  $1/19950$  segundos (50.1253 us) si no lo hace a 10000 us, se lo hace a 10000 us porque el código se ejecuta en memoria FLASH ya que hay más capacidad de memoria para generar las tres señales sinusoidales que en memoria RAM, sin embargo, tiene el inconveniente de que FLASH es más lenta y no da tiempo a ejecutarse la rutina comandada en el interior del timer2.

La figura 5.13 indica en una tabla los parámetros necesarios para implementar este código que previamente se han detallado en el apartado 5.4. Se omiten los coeficientes del filtro pasa banda porque este software no los hace necesarios.

	<b>Parámetros</b>	<b>Valores</b>
<b>Parámetros establecidos</b>	Tiempo de establecimiento ( $t_s$ )	30 ms
	Coeficiente de amortiguación ( $\zeta$ )	0,7
	Error de banda ( $\partial$ )	0,115
<b>Parámetros calculados</b>	c	1,40028008
	Atenuación ( $\sigma$ )	83,3165142
	Frecuencia natural ( $w_n$ )	119,023592
	Amplitud de tensión de señal de entrada ( $v_{grid}$ )	1
	Tiempo integral ( $T_i$ )	0,01176237 s
	Constante proporcional ( $k_p$ )	166,633028
	Constante integral ( $k_i$ )	14166,6154
<b>Parámetros establecidos</b>	Frecuencia de muestreo	19950 Hz
	Periodo de muestreo (T)	50,125 us
<b>Parámetros calculados</b>	Coeficientes del LPF	$B_0 = 166,988082$ $B_1 = -166,27798$

Figura 5.13

La figura 5.14 muestra el diagrama de flujo del código implementado.

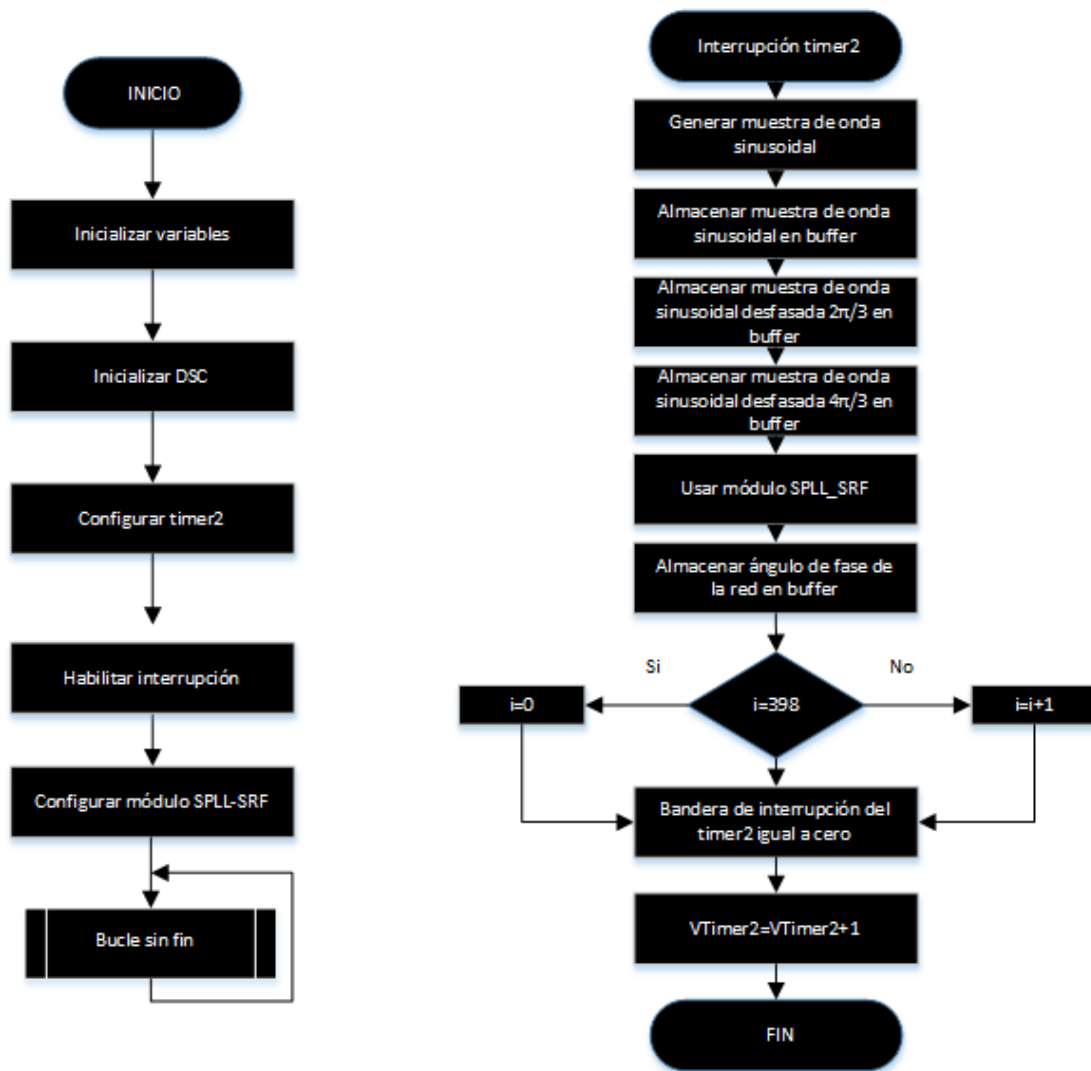


Figura 5.14

El proceso configurar módulo SPLL-SRF del diagrama de flujo de la figura 5.14 se resume en dos pasos:

- Paso 1: Se incluye la librería "Solar\_IQ.h" en el proyecto creado en el Code Composer Studio.
- Paso 2: Se crea y se adhiere la estructura `spll1` al módulo SPLL-SRF en el archivo `SRF_PLL_trifasico_funcionseno.c` del proyecto creado en el Code Composer Studio.

El proceso usar módulo SPLL\_SRF del diagrama de flujo de la figura 5.14 se detalla en los siguientes pasos:

- Paso 1: Se inicializa el módulo SPLL-SRF en el archivo `SRF_PLL_trifasico_funcionseno.c` donde se especifica la entrada de frecuencia de red que en este caso es 50 Hz, la frecuencia de muestreo que se la hace a 19950 Hz y la dirección del objeto PLL.
- Paso 2: Se transforma de un sistema trifásico de alterna variante en el tiempo que en esta implementación es una señal sinusoidal trifásica generada por software a otro

sistema de corriente continua en un marco de referencia de rotación, luego esta transformación es convertida de IQ24 a IQ21 y finalmente se llama a la función SPLL.

La figura 5.15 indica la gráfica del buffer denominado `buffer_angu_fase_red` que corresponde al ángulo de fase de la red trifásica generada por software que va de 0 a  $2\pi$  en un ciclo completo de la sinusoidal. Los datos de este buffer son en numeración IQ21.

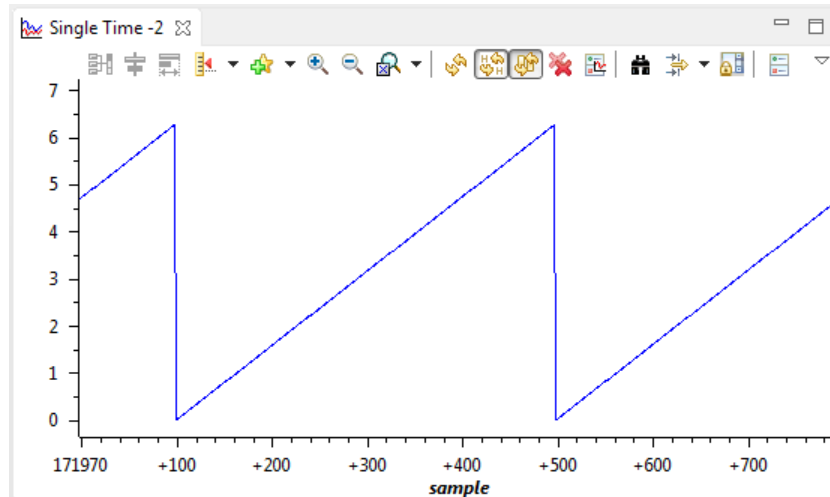


Figura 5.15

También se visualiza en la figura 5.16 el registro de la frecuencia nominal de red trifásica generada por software en Hz  $f_n$  y el registro de la frecuencia instantánea de red trifásica generada por software en Hz  $f_o$ :

(*)= fo	long	50.06658602 (Q-Value(21))	0x0000889C@Data
(*)= fn	long	50.0 (Q-Value(21))	0x0000889E@Data

Figura 5.16

En la simulación se observa como  $f_o$  es estable y se engancha a  $f_n$ .

### 5.7 Implementación SPLL\_3ph\_SRF para sistemas trifásicos con la librería SolarLib con señales de ingreso generada con LaunchPadXL TMS320F28069M.

Con el LaunchPadXL TMS320F28069M se genera 3 señales trifásicas a 51.46 Hz desfasadas 120 grados con tensión pico a pico de 0 a 3.3 Voltios como muestra la figura 5.17 descargada del osciloscopio:

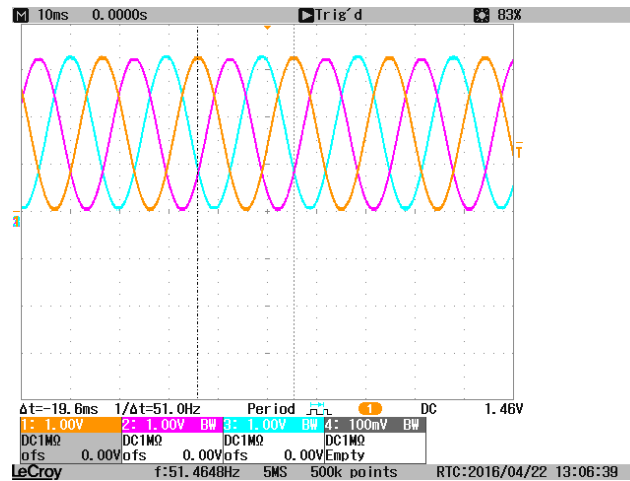


Figura 5.17

Estas tres señales son ingresadas al LaunchPad XL TMS320F28027 que implementa el SPLL\_3ph\_SRF de la librería SolarLib que son muestreadas a 20000 Hz por el timer2.

La figura 5.18 muestra el diagrama de flujo del código.

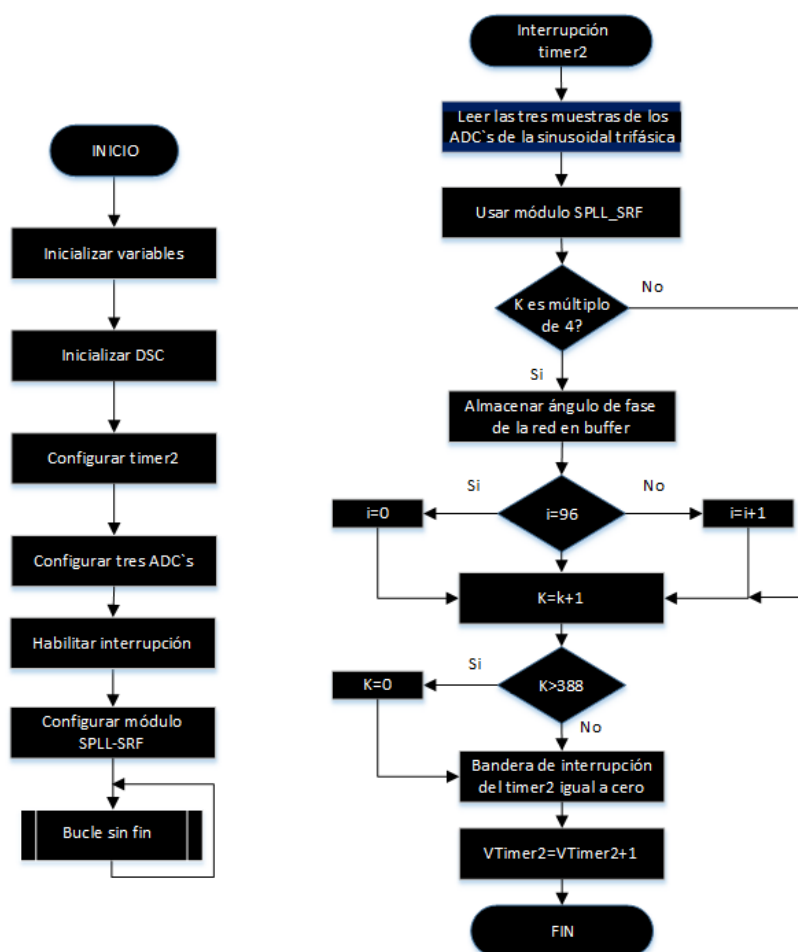


Figura 5.18

El proceso configurar módulo SPLL-SRF del diagrama de flujo de la figura 5.18 se resume en dos pasos:

- Paso 1: Se incluye la librería “Solar\_IQ.h” en el proyecto creado en el Code Composer Studio.
- Paso 2: Se crea y se adhiere la estructura spll1 al módulo SPLL-SRF en el archivo SRF\_PLL\_trifasico\_TMS320F28069M.c del proyecto creado en el Code Composer Studio.

El proceso usar módulo SPLL\_SRF del diagrama de flujo de la figura 5.18 se detalla en los siguientes pasos:

- Paso 1: Se inicializa el módulo SPLL-SRF en el archivo SRF\_PLL\_trifasico\_TMS320F28069M.c donde se especifica la entrada de frecuencia de red que en este caso es 51.46 Hz, la frecuencia de muestreo que se la hace a 20000 Hz y la dirección del objeto PLL.
- Paso 2: Se transforma de un sistema trifásico de alterna variante en el tiempo que en está implementación es una señal sinusoidal trifásica obtenida del procesador digital de señales TMS320F28069M a otro sistema de corriente continua en un marco de referencia de rotación, luego está transformación es convertida de IQ24 a IQ21 y finalmente se llama a la función SPLL.

El buffer denominado buffer\_angu\_fase\_red del archivo SRF\_PLL\_trifasico\_TMS320F28069M.c que indica el ángulo de fase de la red trifásica se muestra en la figura 5.19. Este ángulo debe variar entre 0 a  $2\pi$  y los datos del buffer son en sistema de numeración IQ21

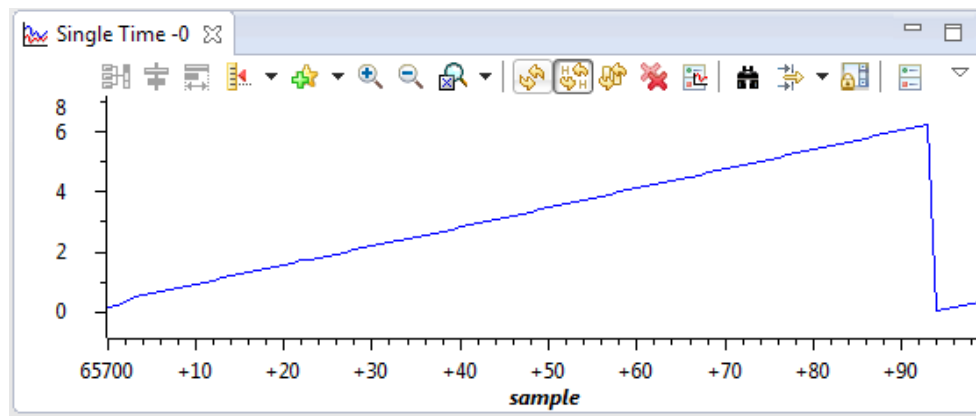


Figura 5.19

Por la poca capacidad de memoria RAM del TMS320F28027 para graficar el ángulo de fase de red en el Code Composer Studio se toman muestras múltiplos de cuatro de la frecuencia muestreada por el timer2, de esta forma una señal sinusoidal se construye en 97 muestras ( $388/4=97$ ).

Finalmente, para ver los parámetros establecidos y calculados previamente obtenidos del apartado 5.4 referirse a la tabla de la figura 5.20. Este software no hace necesario los coeficientes del filtro pasa banda.

	<b>Parámetros</b>	<b>Valores</b>
<b>Parámetros establecidos</b>	Tiempo de establecimiento ( $t_s$ )	30 ms
	Coficiente de amortiguación ( $\zeta$ )	0,7
	Error de banda ( $\vartheta$ )	0,115
<b>Parámetros calculados</b>	c	1,40028008
	Atenuación ( $\sigma$ )	83,3165142
	Frecuencia natural ( $w_n$ )	119,023592
	Amplitud de tensión de señal de entrada ( $v_{grid}$ )	1
	Tiempo integral ( $T_i$ )	0,01176237 s
	Constante proporcional ( $k_p$ )	166,633028
	Constante integral ( $k_i$ )	14166,6154
<b>Parámetros establecidos</b>	Frecuencia de muestreo	20000 Hz
	Periodo de muestreo (T)	50 us
<b>Parámetros calculados</b>	Coficientes del LPF	$B_0 = 166,987194$ $B_1 = -166,27886$

Figura 5.20

### 5.8 Implementación SPLL\_3ph\_SRF con filtro pasa bajo para sistemas trifásicos utilizando la librería SolarLib con señales de ingreso generada con LaunchPadXL TMS320F28069M.

Entre la señal trifásica que sale del LaunchPadXL TMS320F28069M y la que entra al TMS320F28027 se añade un filtro pasa bajo por cada fase formado por 1 resistencias de 12 Kohms y 1 condensadores de 0.1 uf para mejorar la señal de entrada al PLL. El resultado fue similar al obtenido en el apartado 5.7.



## Conclusiones.

Este trabajo concluye en la importancia de investigar y evaluar lazos de seguimiento de fase. Estos lazos pueden usarse en sistemas de sincronización con la red eléctrica para desarrollarlos en la inyección de energía a la red eléctrica proveniente de la generación de energía eléctrica de fuentes de renovables, entonces, los lazos de seguimiento de fase igualan la fase y la frecuencia de la señal de la red eléctrica con la señal que es inyectada a la red eléctrica logrando un óptimo traspaso de energía. Otro campo para investigar y evaluar PLL's es en el control de máquinas eléctricas, un PLL puede suplir el control de velocidad de un motor, un PLL junto con un estimador de flujo magnético acoplados a un sistema de control puede indicar la velocidad y/o posición de una máquina síncrona, un PLL introducido en un sistema de control para máquina síncrona proporciona el ángulo de fase del voltaje en inversores auto conmutados.

Los controladores digitales de señales son herramientas útiles y sencillas para lograr el funcionamiento de los PLL's, en la evaluación del software PLL se ha utilizado la plataforma LAUNCHXL-F28027 (C2000) con el Code Composer Studio como entorno de desarrollo, a continuación, se enumeran las aportaciones del abajo académico:

- 1) Metodología en la aplicación de PLL's sobre la plataforma LAUNCHXL-F28027 (C2000).
- 2) Evaluación de espacio ocupado, tiempo de ejecución y variables necesarias de SPLL sobre la plataforma LAUNCHXL-F28027 (C2000) en sistemas monofásicos:

Algoritmos		Espacio ocupado en FLASH	Espacio ocupado en RAM	Tiempo de ejecución	Variables imprescindibles
SPLL monofásico en cuadratura	Señal de ingreso generada por software (ejecutado en FLASH)	2,2 K (Kilo-bytes) de 32 K	2,5 K de 6 K	271,11 us (microsegundos)	<b>18 variables:</b> i, i_90, Va, Vb, Vsin, Vsin_90, integrador, Ref, Fdb, Err, Out, integral, wff, pi_2, kp, ki_ts, OutMax, OutMin y 7.8125E-5
	Señal de ingreso generada por generador de funciones (ejecutado en FLASH)	2,6 K de 32 K	2 K de 6 K	15743 us	<b>18 variables:</b> i, i_90, Va, Vb, Vsin, Vsin_90, integrador, Ref, Fdb, Err, Out, integral, wff, pi_2, kp, ki_ts, OutMax, OutMin y 7.8125E-5

3) Evaluación de espacio ocupado, tiempo de ejecución y variables necesarias de SPLL utilizando la librería solar de Texas Instruments sobre la plataforma LAUNCHXL-F28027 (C2000) en sistemas monofásicos y trifásicos.

Algoritmos		Espacio ocupado en FLASH	Espacio ocupado en RAM	Tiempo de ejecución	Variables imprescindibles
SPLL monofásico	Señal de ingreso generada por software (ejecutado en FLASH)	5,9 K de 32 K	1,5 K de 6 K	354,26 us	<b>8 variables externas al módulo:</b> Va, i, k, 50, 20000, 0.000000001, 1, 0.01570796327 <b>26 variables internas al módulo:</b> AC_input, theta [0], theta [1], cos [0], cos [1], sin [0], sin [1], wo, wn, B2_notch, B1notch, B0notch, A2notch, A1notch, B1_if, B0_if, A1_if, Upd [0], Upd [1], Upd [2], ynotch [0], ynotch [1], ynotch [2], ylf [0], ylf [1], delta_t.
	Señal de ingreso generada por generador de funciones (ejecutado en RAM)	0 K de 32 K	4,5 K de 6 K	98,61 us	<b>9 variables externas al módulo:</b> Va, i, k, 50, 20000, 0.000000001, 1, 0.000488281, 2048. <b>26 variables internas al módulo:</b> AC_input, theta [0], theta [1], cos [0], cos [1], sin [0], sin [1], wo, wn, B2_notch, B1notch, B0notch, A2notch, A1notch, B1_if, B0_if, A1_if, Upd [0], Upd [1], Upd [2], ynotch [0], ynotch [1], ynotch [2], ylf [0], ylf [1], delta_t.
SPLL-SRF trifásico	Señal de ingreso generada por software (ejecutado en FLASH)	5,9 K de 32 K	2,6 K de 6 K	10220,95 us	<b>13 variables externas al módulo:</b> Va, Vb, Vc, i, i_20, i_240, 50, 19950, 0.0157473316, 266, 398, 399 y 133. <b>12 variables internas al módulo:</b> v_q [0], v_q [1], ylf [0], ylf [1], fo, fn, theta [0], theta [1], delta_T, B1_if, B0_if, A1_if
	Señal de ingreso generada por el TMS320F28069M (ejecutado en RAM)	0 K de 32 K	4,2 K de 6 K	91,51 us	<b>9 variables externas al módulo:</b> Va, Vb, Vc, i, k, 50, 20000, 0.000488281, 2048. <b>12 variables internas al módulo:</b> v_q [0], v_q [1], ylf [0], ylf [1], fo, fn, theta [0], theta [1], delta_T, B1_if, B0_if, A1_if

Para trabajos futuros se pueden evaluar los algoritmos de software PLL en otras plataformas y en condiciones reales.

## Plan de trabajo

Los espacios en negro de la tabla describen por mes las actividades del trabajo realizado:

Actividades	Enero 2016	Febrero 2016	Marzo 2016	Abril 2016	Mayo 2016	Junio 2016	Julio 2016
Elección del tema							
Recolección de información							
Selección de información							
Memoria							
Revisión							
Defensa							

## Bibliografía.

- [1] WIKIPEDIA. Lazo de seguimiento de fase. [Documento en línea]. Disponible desde Internet en: <[https://es.wikipedia.org/wiki/Lazo\\_de\\_seguimiento\\_de\\_fase](https://es.wikipedia.org/wiki/Lazo_de_seguimiento_de_fase)> [con acceso en enero del 2016]
- [2] Boylestad-Nashelsky. Electrónica: Teoría de circuitos y dispositivos electrónicos. Traducido por Carlos Mendoza. 8va edición. México: PEARSON EDUCACIÓN, 2003, 1040 páginas. ISBN:970-26-0436-2. [con acceso en enero del 2016]
- [3] H R Pota. 06/06/2005. Phase-Locked Loop. [Documento en línea]. Disponible desde Internet en: <[http://enhanceedu.iiit.ac.in/wiki/images/PLL\\_3.pdf](http://enhanceedu.iiit.ac.in/wiki/images/PLL_3.pdf)> [con acceso en enero del 2016]
- [4] ROLAND E. BEST. Phase-Locked Loops DESIGN, SIMULATION, AND APPLICATIONS. Quinta edición. Estados Unidos de América: McGraw-Hill, 2003, ISBN 0-07-141201-8. [con acceso en mayo del 2016]
- [5] Universitat Politècnica de Catalunya. R. Pindado. Phase Locked-Loop (PLL): Fundamento y aplicaciones. [Documento en línea]. Disponible desde Internet en: <<http://www.jcee.upc.es/JCEE2001/PDFs2001/pindado.pdf>> [con acceso en enero del 2016]
- [6] Liping Zheng and Dong Le. 2014. Control of a 750kW Permanent Magnet Synchronous Motor. [Documento en línea]. Disponible desde Internet en: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6869834>> [con acceso en mayo del 2016]
- [7] Colin Debruyne\*†, Stijn Derammelaere\*†, Jan Desmet\*† and Lieven Vandeveldet†. 2012. Using General Synchronous Machine Theory to integrate PLL Controller Dynamics into a Static Power Electronic Converter Model. [Documento en línea]. Disponible desde Internet en: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6374092>> [con acceso en mayo del 2016]
- [8] K.M.S.Y Konara, M.L. Kolhe, W.G.C.A. Sankalpa. 12/2015. Grid Synchronization of DC Energy Storage Using Voltage Source Inverter with ZCD and PLL Techniques. [Documento en línea]. Disponible desde Internet en: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7399055>> [con acceso en mayo del 2016]
- [9] Villalobos, R.; López, H.; Vázquez, N.; Hernández, C. Octubre 2012. Estructuras de control SOGI para sincronización con la red eléctrica. [Documento en línea]. Disponible desde Internet en: <<http://pistaseducativas.itc.mx/wp-content/uploads/2012/10/1-VILLALOBOS-PE-99-6-21.pdf>> [con acceso en enero del 2016]
- [10] Timothy Thacker, Member, IEEE, Dushan Boroyevich, Fellow, IEEE, Rolando Burgos, Member, IEEE, and Fei Wang, Fellow, IEEE. 06/2011. [Documento en línea]. Disponible desde Internet en: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5556002>> [con acceso en mayo del 2016].

[11] Xiao-Qiang GUO, Wei-Yang WU, He-Rong GU. 04/2011. Phase locked loop and synchronization methods for grid interfaced converters: a review. [Documento en línea]. Disponible desde internet en: <<http://www.red.pe.org.pl/articles/2011/4/48.pdf>> [con acceso en febrero del 2016] Texas Instruments. 10/21013. Piccolo™ Microcontrollers. [Documento en línea]. Disponible desde Internet en:<<http://www.ti.com/lit/ds/sprs523j/sprs523j.pdf>> [con acceso en marzo del 2016]

[12] Department of Electronic Technology, University of Vigo. Francisco D. Freijedo, Jesus Doval-Gandoy, Oscar Lopez, Carlos Martinez-Peñalver, Alejandro G. Yepes, Pablo Fernandez-Comesaña, Jano Malvar, Andres Nogueiras, Jorge Marcos y Alfonso Lago. Grid-Synchronization Methods for Power Converters. [Documento en línea]. Disponible desde Internet en:<<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5414976>> Formato pdf. [con acceso en febrero del 2016]

[13] TI E2E Community. Single-phase PLL in C2000. [Documento en línea]. Disponible desde Internet en: <<http://e2e.ti.com/group/microcontrollerprojects/m/c2000microcontrollerprojects/665489.aspx>> [con acceso en febrero del 2016]

[14] Texas Instruments. 01/2014. C28x Solar Library. Formato pdf. [con acceso en febrero del 2016]

[15] Universidad Politécnica de Valencia, Departamento de Ingeniería Electrónica. Francisco Gimeno, Salvador Orts, Salvador Seguí. DEA\_Introduccion\_C2000.pdf. [con acceso en marzo del 2016]

[16] Texas Instruments, SPR523J [con acceso en marzo del 2016]