

PROYECTO DE GRADO



Universidad
Tecnológica
de Pereira

JORGE ALBERTO ARIAS ZULUAGA

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA
INGENIERÍA ELÉCTRICA
PEREIRA
JULIO - 2017**

ESTUDIO COMPARATIVO PARA PLL MONOFÁSICO

JORGE ALBERTO ARIAS ZULUAGA

CO-DIRECTOR

PHD. ING. ALEJANDRO GARCÉS RUIZ

DIRECTOR

ING. DANIEL FERNANDO PATIÑO IPUS

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA
INGENIERÍA ELÉCTRICA
PEREIRA
JULIO - 2017**

TABLA DE CONTENIDO

1. Introducción
 - 1.1. Planteamiento del problema
 - 1.2. Objetivo general
 - 1.3. Objetivos específicos
 - 1.4. Propuesta de solución
 - 1.5. Aportes del proyecto
 - 1.6. Estructura del documento
 2. Aspectos teóricos
 - 2.1. Antecedentes
 - 2.2. OSG-PLL
 - 2.3. TPFA-PLL
 - 2.4. VTD-PLL
 - 2.5. Adquisición de señales
 - 2.5.1. Acondicionamiento
 - 2.5.2. Adquisición
 3. Metodología propuesta
 - 3.1. Adquisición
 - 3.2. Implementación de PLLs en Arduino
 4. Aplicación de la metodología propuesta
 - 4.1. Resultados del PLL
 - 4.1.1. OSG-PLL
 - 4.1.2. TPFA-PLL
 - 4.1.3. VTD-PLL
 - 4.2. Medida de desempeño
 5. Conclusiones y trabajos a futuro
- Bibliografía

Agradecimientos

Le doy infinitas gracias a Dios por este maravilloso regalo. A mi Madre Lucía Zuluaga, a mi Padre José Eduval Arias, a mi Hermano Duán Andrés Arias Zuluaga, a mi Hermana Linda Palmera Arias.

Le agradezco a mi tía Angie Zuluaga, a mis Tíos Martín Zuluaga, Carlos Arturo Zuluaga, Diego Zuluaga, a mis tías Luz Marina y Nancy Zuluaga. A mi abuela Orfa Orozco que gracias a Dios pudo verme llegar a este logro tan importante. A Luz, Andrés y todos los que han sido maestros en el camino. Gracias a la vida que nos da bendiciones todos los días y que sea el comienzo de un camino de felicidad y proyectos maravillosos.

Capítulo 1

1. Introducción

1.1.Planteamiento del problema

La sincronización entre convertidores de potencia, con la red eléctrica, es un tema importante de investigación, debido a la integración masiva de fuentes renovables y de cargas basadas en componentes de electrónica de potencia (EDP), por lo cual, muchos investigadores se han dedicado a proponer y mejorar diversas técnicas que permiten vincular redes entre sí [1] [2].

Para lograr la coordinación de sistemas monofásicos es necesario determinar la fase, amplitud y frecuencia de la señal de la red. El manejo de la amplitud es sencillo ya que puede medirse con el valor pico de la señal o utilizando métodos de detección de voltaje rms, por el contrario, la detección de fase se hace crítica ya que una pequeña variación entre la señal entregada por los dispositivos basados en EDP y la señal generada por la red puede producir corrientes circulantes ocasionando daños en los elementos del sistema [3].

De los diferentes métodos de sincronización utilizados, las topologías de PLLs (Phase-Locked-Loop) son de las más aceptadas por su estructura simple y robustez frente a perturbaciones en la red [2]. En la literatura se encuentran múltiples PLLs monofásicos que son simulados y comparados entre sí. Sin embargo, en esta investigación se pretende seleccionar topologías para PLLs e implementarlas en Arduino, para posteriormente analizar los resultados experimentales y observar el funcionamiento de estos desde el punto de vista práctico.

1.2.Objetivo general

Implementar un sistema que permita medir parámetros de frecuencia y fase de una señal periódica monofásica, ante una variación en la frecuencia de la señal de entrada.

1.3.Objetivos específicos

- Diseñar distintas topologías de PLLs en MATLAB.
- Implementar las topologías seleccionadas para PLLs en Arduino.
- Analizar los resultados experimentales.
- Documentar resultados de la Investigación.

1.4.Propuesta de solución

Con el fin de alcanzar los objetivos propuestos, esta investigación se divide en tres etapas. En la primera etapa, se realiza una recopilación bibliográfica de los seguidores de fase en lazo cerrado más utilizados para sistemas monofásicos. Luego, se diseñan diferentes topologías de PLLs monofásicos por medio del software Matlab. Por último, se implementan diferentes topologías de PLLs monofásicos en el micro-controlador Arduino con el fin de medir una señal sinodal con una frecuencia conocida y realizar un análisis comparativo de los PLLs estudiados en esta investigación.

1.5.Aportes del proyecto

En este proyecto se implementan diferentes PLLs monofásicos en el micro-controlador Arduino, con el fin de analizar el comportamiento de estos seguidores de fase de manera que puedan ser usados en una etapa posterior de control.

Se presenta una metodología sencilla para el acondicionamiento y la adquisición de señales con el micro-controlador Arduino, además de presentan los algoritmos usados en el desarrollo del proyecto.

1.6.Estructura del documento

Este proyecto está dividido en 5 capítulos. El capítulo inicial contiene la introducción del proyecto, los objetivos, la propuesta de solución y los aportes del proyecto. En el segundo capítulo se encuentran los aspectos teóricos del proyecto, como antecedentes referentes a los PLLs monofásicos reportados en la literatura, sus topologías, adquisición de señales. El tercer capítulo corresponde a la metodología propuesta, aquí se presenta el acondicionamiento de la señal, adquisición de la señal e implementación de PLLs en Matlab y Arduino. En el cuarto capítulo se presenta la aplicación de la metodología propuesta, se presentan los escenarios de prueba, consideraciones previas y resultados. En el capítulo cinco se tienen las conclusiones y las indicaciones de trabajo a futuro. Al final se presentan las referencias citadas en este documento y se anexa el código del aplicativo

Capítulo 2

2. Aspectos teóricos

2.1.Antecedentes

En la literatura se proponen diferentes técnicas de sincronización de señales. Una técnica eficiente de estimación de parámetros de la señal distorsionada de tensión en la red [8], la cual se basa en el integrador generalizado de segundo orden (SOGI) y un generador de señales en cuadratura (QSG), los métodos para el análisis espectral son la Transformada Rápida de Fourier (TRF) y la transformada Discreta de Fourier (TDF) en ventana fija. Para determinar la robustez de esta técnica (QSG-SOGI) se compara con otras técnicas existentes en la bibliografía como, SOGI-FLL, SOGI-PLL y técnicas QPLL, encontrando que la técnica propuesta muestra gran capacidad para seguir la variación de los parámetros,

no hay necesidad de ajustar los parámetros del controlador para la estimación de la frecuencia, es más rápido, entre otras ventajas, sin embargo, cuando se va a implementar la TDF en ventana fija se requiere de un búfer de memoria aumentando la complejidad computacional, la cual se puede reducir mediante el uso de la forma recursiva de la TDF (RTDF).

También se propone en [9], una técnica de seguidor de frecuencia en lazo cerrado FLL basada en Filtro de Kalman Lineal LKF, capaz de: monitorear las perturbaciones de señal de tensión de la red y estimar con precisión todos los parámetros variables en el tiempo de esa señal, además, esta técnica reduce el tiempo de convergencia del LKF ya que la frecuencia fundamental de la red se extrae por separado usando la técnica de FLL y posteriormente esta frecuencia se utiliza para estimar las formas de onda ortogonales de la componente fundamental y los armónicos presentes en la forma de onda de tensión de la red distorsionada, sin embargo, esta técnica no es tan eficiente debido a que se necesita un pequeño tiempo de establecimiento para estimar el cambio de los parámetros.

Una comparación entre, el PLL basado en retardo (Delay-PLL), el PLL basado en la inversa de Park y el PLL basado en la transformada de Hilbert, se realiza en [10]. Estos son algoritmos de PLLs para sistemas monofásicos, fundamentados en un sistema de referencia síncrono el cual requiere un OSG. De los resultados obtenidos se concluye que Delay-PLL presenta el mejor rendimiento en la extracción de amplitud, frecuencia y ángulo de fase, frente a los otros dos PLLs estudiados.

Se presenta un OSG basado en un integrador generalizado de segundo orden (SOGI), con el cual es posible determinar valores de fase, amplitud y frecuencia de la tensión de red, parámetros fundamentales para generar las señales de referencia y asegurar una correcta operación de los sistemas de inversores conectados a la red [3]. Algunas de las ventajas de esta topología son: fácil de aplicar, la generación de las señales ortogonales es rápida, la estructura no se ve afectada por cambios de frecuencia.

En [11] se diseña y analiza el esquema de control de un inversor de seis interruptores para un sistema de generación de energía fotovoltaica conectado a la red monofásica. Para detectar los parámetros de la señal de tensión de la red y obtener la tensión de referencia a la salida del inversor se aplica la técnica de PLL basada en la transformada inversa de Park, que es uno de los algoritmos más utilizados para generar señales en cuadratura y obtener un buen rendimiento de sincronización con la red.

Un algoritmo Delay-PLL se presenta en [12], el cual utiliza filtros de Kalman para el seguimiento de una señal ascendente desde un GPS de la órbita terrestre baja, la cual proporciona información casi instantánea sobre las condiciones meteorológicas próximas a la superficie terrestre. El filtro de Kalman maneja los paquetes de estados optimizados como medidas, es decir, como entadas del sistema, además se utiliza un Delay-PLL como control en lazo cerrado para proporcionar las entradas y definir la región de los datos.

2.2.OSG-PLL

En la figura 1, se muestra la estructura del OSG-PLL, el cual utiliza la transformada de Park para obtener la tensión en eje directo y de cuadratura.

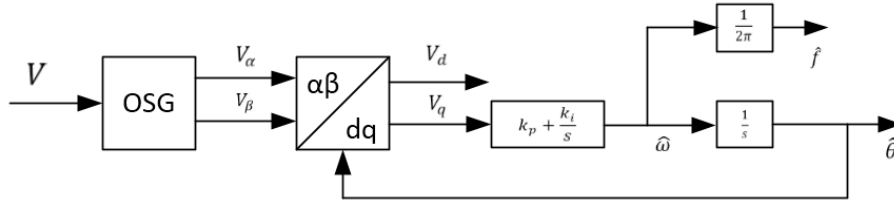


Figura 1. Diagrama de bloques para el OSG-PLL.

A continuación se describen las ecuaciones que representan el diagrama de bloques del OSG-PLL. Primero se obtienen los valores V_α y V_β , como se propone en (1).

$$V_\alpha = V$$

$$V_\beta = \begin{cases} \cos(\sin^{-1}(V_\alpha)) & \text{si: } dv/dt < 0 \\ -\cos(\sin^{-1}(V_\alpha)) & dv/dt > 0 \end{cases} \quad (1)$$

Luego se halla la tensión de cuadratura mediante la ecuación (2).

$$V_q = -V_\alpha \sin(\hat{\theta}) + V_\beta \cos(\hat{\theta}) \quad (2)$$

La tensión de cuadratura es usada para calcular el $\hat{\omega}$ de la red mediante un control PI como se muestra en (3).

$$\hat{\omega} = k_p V_q + k_i \int V_q(t) dt \quad (3)$$

Se integra el $\hat{\omega}$ y se obtiene el $\hat{\theta}$ de la red, el cual debe ser realimentado para la transformada de Park, y posteriormente se encuentra la frecuencia como se presenta en (4).

$$\hat{\theta} = \int \hat{\omega}(t) dt$$

$$\hat{f} = \frac{\hat{\omega}}{2\pi} \quad (4)$$

2.3.TPFA-PLL

En la figura 2, se muestra la estructura simplificada del TPFA-PLL, el cual utiliza la transformada de Park y la transformada de Park inversa para obtener el error en la tensión en eje directo y de cuadratura.

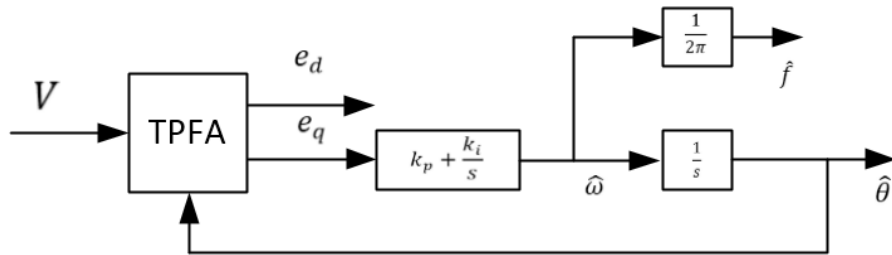


Figura 2. Diagrama de bloques para el TPFA-PLL.

A continuación se presentan las ecuaciones simplificadas para el diagrama de bloques del TPFA-PLL. Primero se obtienen los valores V_α y θ_g , como se propone en (5).

$$V_\alpha = V$$

$$\theta_g = \begin{cases} \sin^{-1}(V_\alpha) & \text{si: } dv/dt > 0 \\ -\sin^{-1}(V_\alpha) + \pi & dv/dt < 0 \end{cases} \quad (5)$$

Luego se halla el error en la tensión de cuadratura mediante la ecuación (6).

$$e_q = \sin(\theta_g - \hat{\theta}) \quad (6)$$

El error en la tensión de cuadratura es usada para calcular el $\hat{\omega}$ de la red mediante un control PI como se muestra en (7).

$$\hat{\omega} = k_p e_q + k_i \int e_q(t) dt \quad (7)$$

Las ecuaciones descritas en (4) son utilizadas para encontrar el $\hat{\theta}$ y la frecuencia de la red.

2.4.VTD-PLL

En la figura 3, se muestra la estructura simplificada del VTD-PLL, el cual reduce la carga computacional encontrando el error en la señal de manera que encuentre el $\hat{\theta}$ de la red.

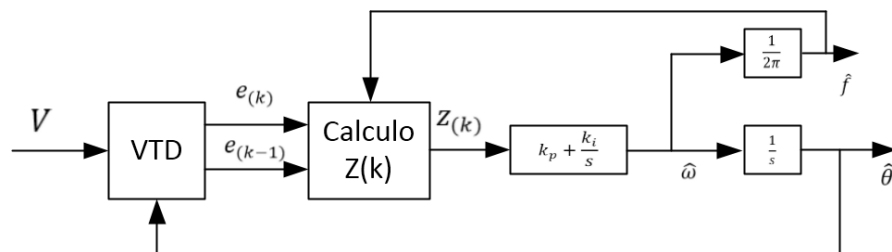


Figura 3. Diagrama de bloques para el VTD-PLL.

A continuación se presentan las ecuaciones simplificadas para el diagrama de bloques del VTD-PLL. Primero se obtienen los valores V_α y θ_g , como se propone en (5).

Luego se halla el error en el estado k , mediante la ecuación (8).

$$e_{(k)} = \sin(\theta_g - \hat{\theta}) \quad (8)$$

En la ecuación (9) se presenta la señal actualizada del error. Esta ecuación marca una diferencia con el método TPFA-PLL, debido a que en el VTD-PLL se tiene en cuenta la variación de la frecuencia.

$$z_{(k)} = e_{(k)}\hat{f} - e_{(k-1)} \quad (9)$$

La señal actualizada del error es usada para calcular el $\hat{\omega}$ de la red mediante un control PI como se muestra en (10).

$$\hat{\omega} = k_p z_{(k)} + k_i \int z_{(k)}(t) dt \quad (10)$$

Las ecuaciones descritas en (4) son utilizadas para encontrar el $\hat{\theta}$ y la frecuencia de la red.

2.5. Adquisición de señales

La etapa de adquisición de señales se divide en dos, primero se realiza un acondicionamiento de la señal AC, luego se adquieren los datos con el micro-controlador Arduino como se muestra a continuación.

2.5.1. Acondicionamiento

El acondicionamiento de señal es de suma importancia, debido a que el micro-controlador tiene un límite de tensión para la entrada analógica. Se debe tomar la señal de la red de distribución, luego se reduce el nivel del valor AC de la tensión de entrada mediante un transformador 110/6 V.

El Arduino Due tiene un límite para la entrada analógica de 0 a 3.3 voltios, por lo cual, la señal de entrada se debe atenuar, y adicionar un off-set de manera que el micro-controlador pueda observar la señal de la red sin ningún tipo de saturación. Para diseñar del acondicionamiento se implementa la ecuación $y = mx + b$, de manera que m es la ganancia que atenúa la señal, b corresponde al off-set, x es la señal de entrada, por último se tiene a y como la señal de salida del acondicionamiento, es decir, la entrada al micro-controlador.

En la figura 4 se muestra el diagrama de flujo del acondicionamiento.

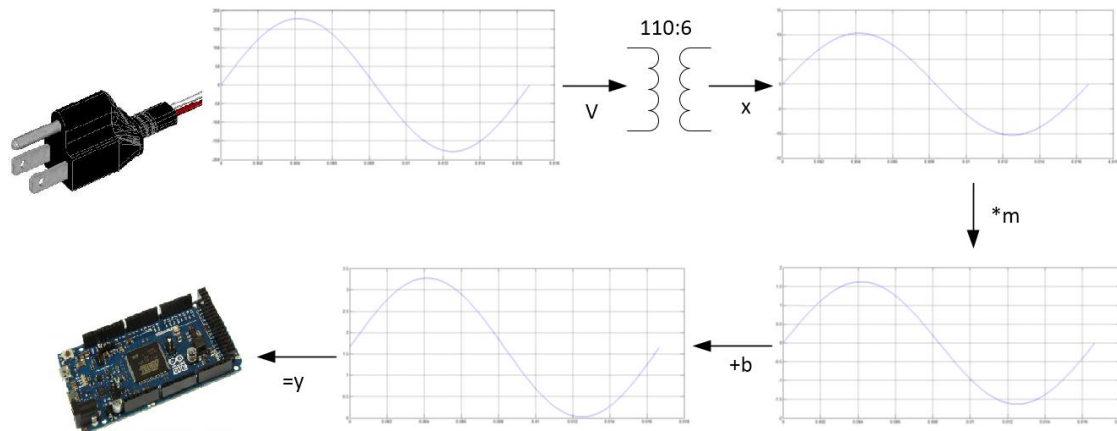


Figura 4. Diagrama de flujo para el acondicionamiento.

2.5.2. Adquisición

Arduino es una plataforma de prototipos de electrónica de código abierto (open-source) basado en hardware y software flexibles fáciles de usar. Las señales análogas de entrada para el Arduino Due, están representadas con un valor entero entre 0 y 1023, de manera que se debe escalar los datos de entrada para tener las señales en por unidad. Se utiliza la ecuación (11) para llevar la señal proveniente del acondicionamiento a valores en por unidad.

$$V = \text{analogRead}(A0) * 2/1023.0 - 1 \quad (11)$$

Capítulo 3

3. Metodología propuesta

La metodología propuesta se divide en 4 etapas, con el fin de realizar una comparación del comportamiento de 3 diferentes PLLs monofásicos, y sus posibles aplicaciones mediante la implementación de los métodos en un micro-controlador.

3.1. Adquisición

La etapa de adquisición se realiza mediante el micro-controlador Arduino Due, utilizando la ecuación (11) para obtener los valores de tensión en por unidad, provenientes del acondicionamiento. Luego de tener la señal de la red en el micro-controlador, es posible realizar la etapa de control para encontrar la frecuencia y la velocidad angular de la red de distribución., utilizando los algoritmos descritos en las secciones 2.2 al 2.4.

3.2. Implementación de PLLs en Arduino

Se implementan los diferentes PLLs estudiados en la plataforma Arduino, utilizando una entrada análoga que proviene de la etapa de acondicionamiento, con el fin de obtener las variables de frecuencia y velocidad angular de la red que se conecte el dispositivo desarrollado en este proyecto. El Arduino permite interactuar con el medio externo mediante adquisición de datos y programación basada en C, por lo cual se tiene una estructura estricta dividida en 3 partes. En la primera parte, se llaman las constantes, variables, vectores y demás elementos que se quieran para la aplicación, en la segunda parte se define la comunicación con la consola del PC, además se definen los pines de entrada y de salida del micro-controlador. La tercera parte contiene el cuerpo del código, es decir, el código que se tiene en esta sección se ejecuta continuamente.

A continuación se presentan los códigos desarrollados en la plataforma de Aduino, los cuales se ingresan en el micro-controlador de manera que permita obtener la velocidad angular, además el $\hat{\theta}$, requerido para realizar el acoplamiento entre sistema AC mediante diversas etapas de control. Para realizar las integrales requeridas en los PLLs, se debe tener el delta de tiempo con el cual se realizar el muestreo de la señal. La estrategia se muestra en los códigos implementados, la cual consiste en tomar el tiempo al momento de iniciar el cuerpo del código, luego se calcula el delta tiempo al finalizar la sección 3. La función `micros()`, es una función interna de Arduino que entrega una valor entero correspondiente a los microsegundos transcurridos en la simulación.

En la figura 5, se presenta el código para el OSG-PLL implementado en la plataforma Arduino, el cual estima los parámetros del sistema como la frecuencia, velocidad angular y el $\hat{\theta}$ de la red. Este código utiliza las ecuaciones mostradas en la sección **2.2**. Luego, en la figura 6 se muestra el código para el TPFA-PLL implementado en la plataforma de Arduino utilizando las ecuaciones mostradas en la sección **2.3**. Finalmente, en la figura 7 se muestra el código del TPFA-PLL implementado en la plataforma de Arduino haciendo uso de las ecuaciones descritas en la sección **2.4**.

```

int kp=100,ki=5000;
float wt=0,wtant=0;
float Valfa,Vant=0,Vbeta,intVqant=0;
float w,intVq,Frec;
float Vq,deltaT;
int tiempo=0,i=1;
float T=1e6/60.0;
void setup() {
  Serial.begin(9600);
}
void loop() {
  tiempo=micros();
  Valfa=analogRead(A0)*2/1023.0-1;
  if ((Valfa-Vant)/deltaT<0){
    Vbeta=cos(asin(Valfa));
  }else{Vbeta=-cos(asin(Valfa));}
  Vq=-Valfa*sin(wt)+Vbeta*cos(wt);
  intVq=deltaT*1.0e-6*Vq+intVqant;
  w=kp*Vq+ki*intVqant;//+377;
  if ((deltaT*1.0e-6*w+wtant)<2*3.1416){
    wt=deltaT*1.0e-6*w+wtant;}
  else{wt=deltaT*1.0e-6*w+wtant-2.0*3.1416;}
  intVqant=intVq;
  wtant=wt;
  Vant=Valfa;
  Frec=w/(2.0*3.1416);
  if (w<0){w=-1*w;
  Serial.println(w);
  deltaT=(micros()-tiempo);
}
}

```

Figura 5. OSG-PLL implementado en Arduino.

```

int kp=100,ki=5000;
float wt=0,wtant=0;
float Valfa,Vant=0,Vbeta,intVqant=0;
float w,intVq,Frec;
float Vq,deltaT;
int tiempo=0;
float T=1e6/60.0,The;
void setup() {
  Serial.begin(9600);
}
void loop() {
  tiempo=micros();
  Valfa=analogRead(A0)*2/1023.0-1;
  if ((Valfa-Vant)/deltaT>=0){
    The=asin(Valfa);
  }else{The=-asin(Valfa)+3.1416;}
  Vq=sin(The-wt);
  intVq=deltaT*1.0e-6*Vq+intVqant;
  w=kp*Vq+ki*intVqant;//+377;
  if ((deltaT*1.0e-6*w+wtant)<2*3.1416){
    wt=deltaT*1.0e-6*w+wtant;}
  else{wt=deltaT*1.0e-6*w+wtant-2*3.1416;}
  intVqant=intVq;
  wtant=wt;
  Vant=Valfa;
  Frec=w/(2.0*3.1416);
  Serial.println(w);
  deltaT=(micros()-tiempo);
}
}

```

Figura 6. TPFA-PLL implementado en Arduino.

```

float kp=0.1,ki=1;
float wg=2*3.1416*60.0,wff=400;
float z=377.0,zant=0,wtant=0,f_=0;
float ekant=0,wt=0,Vant=0;
float intz,w_,The,deltaw,deltawt,ek,V;
int tiempo=0,deltaT;
void setup() {
  Serial.begin(9600);
}
void loop() {
  tiempo=micros();
  V=analogRead(A0)*2/1023.0-1;
  if ((V-Vant)/deltaT>0){
    The=asin(V);
  }else{The=-asin(V)+3.1416;}
  intz=deltaT*1.0e-6*z+zant;
  w_=kp*z+ki*intz+wff;
  deltaw=wg-w_;
  deltawt=The-wt;
  ek=sin(deltawt);
  z=ek*f_-ekant;
  wt=deltaT*1.0e-6*w_+wtant;
  f_=wt/(2*3.1416*micros()*1.0e-6);
  ekant=ek;
  zant=z;
  wtant=wt;
  Vant=V;
  Serial.println(w_);
  deltaT=(micros()-tiempo);
}
}

```

Figura 7. VTD-PLL implementado en Arduino.

Capítulo 4

4. Aplicación de la metodología propuesta

La metodología propuesta en este proyecto utiliza la señal de tensión proveniente de la red de distribución. Esta señal pasa por la etapa del acondicionamiento e ingresa al micro-controlador Arduino para realizar la etapa de control y estimar los parámetros del sistema como la frecuencia, velocidad angular y el $\hat{\theta}$ de la red.

Se prueba la metodología propuesta bajo un escenario de prueba con variación en la frecuencia de la señal de entrada, para verificar el comportamiento de los PLLs ante variaciones en la frecuencia.

4.1. Resultados del PLL

A continuación, se presentan los resultados obtenidos para los PLLs. El micro-controlador Arduino utiliza el Serial Plotter para graficar cualquier variable del programa. En este caso, se imprime la frecuencia del sistema ante una variación en la frecuencia para la señal de entrada.

4.1.1. OSG-PLL

En la figura 8 se muestra el comportamiento del OSG-PLL para la frecuencia estimada con este PLL, ante variación en la frecuencia de la señal de entrada.

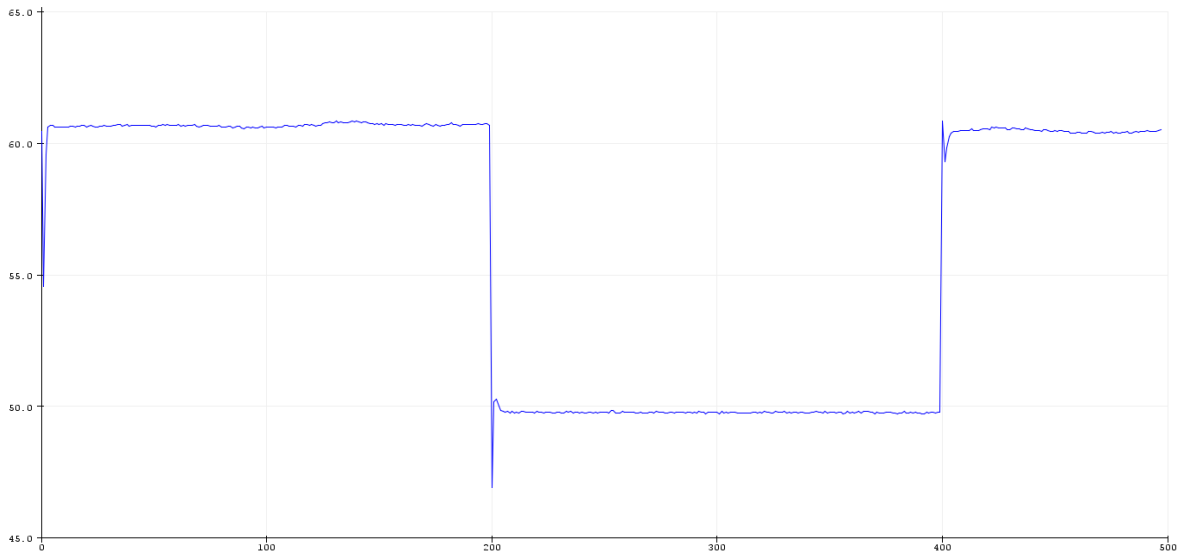


Figura 8. Velocidad angular de la red con el OSG-PLL.

4.1.2. TPFA-PLL

En la figura 9 se muestra el resultado del TPFA-PLL para la frecuencia de la red ante una variación en la frecuencia de la señal de entrada.

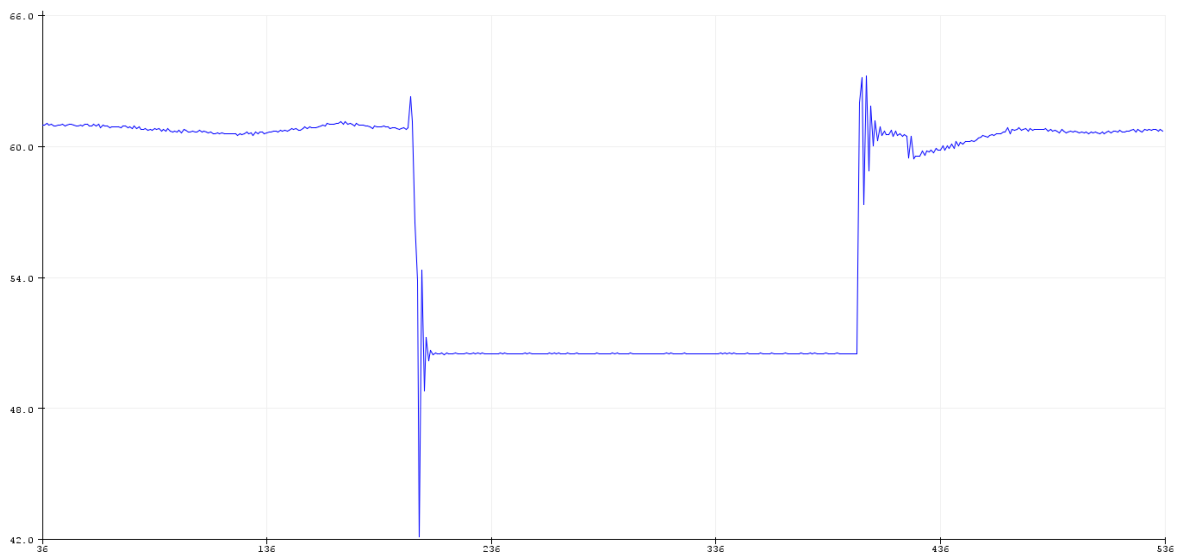


Figura 9. Velocidad angular de la red con el TPFA-PLL.

4.1.3. VTD-PLL

En la figura 10 se muestra el resultado del VTD-PLL para la frecuencia de la red ante una variación en la frecuencia de la señal de entrada.

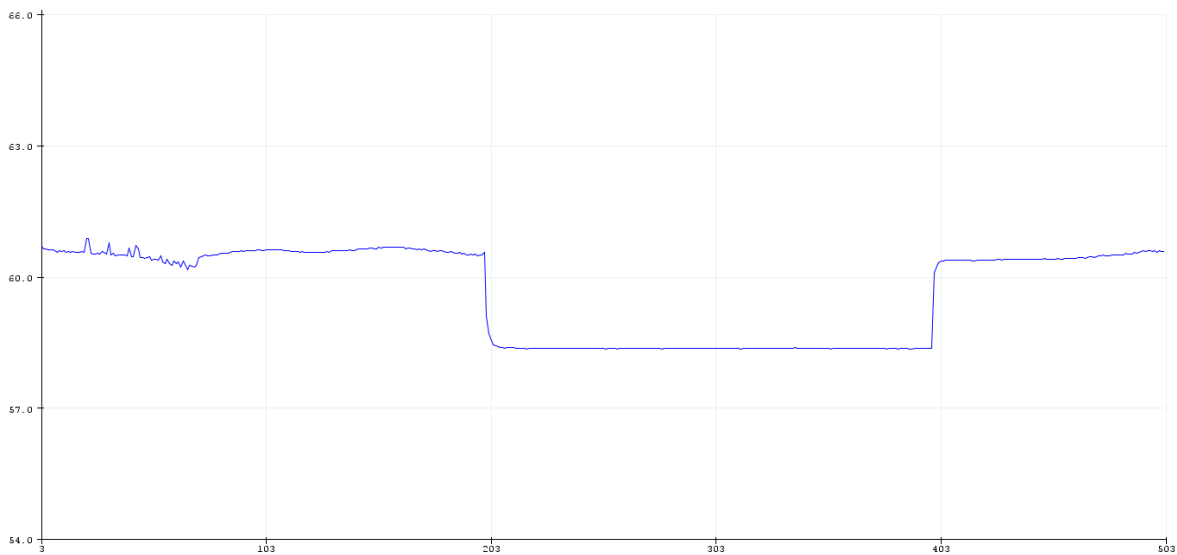


Figura 10. Velocidad angular de la red con el VTD-PLL.

4.2. Medida de desempeño

La medida de desempeño se obtiene con el rango de variación para la frecuencia del sistema obtenida con cada PLL estudiado.

PLL	Rango de frecuencia en estado estable
OSG-PLL	[60,40 – 60,60]; delta:
TPFA-PLL	[59,60 – 60,90]; delta:
VTD-PLL	[60,10 – 60,60]; delta:

Capítulo 5

5. Conclusiones y trabajos a futuro

5.1. Conclusiones

Se implementaron tres topologías diferentes de PLLs en el micro-controlador Arduino, en el OSG-PLL se obtuvo un mejor desempeño debido a que tiene un menor rango de variación para la frecuencia obtenida del sistema.

Es muy importante desarrollar un buen acondicionamiento, de manera que impida el mal funcionamiento del Arduino por sobretensión o polarización inversa, además, los PLLs implementados son sensibles a problemas de saturación de la señal.

5.2.Trabajos a futuro

Con el desarrollo presentado en este proyecto, se presentan como trabajos a futuro:

Diseñar e implementar un convertidor monofásico con la etapa de modulación y control, utilizando los PLLs desarrollados para realizar el acoplamiento y el control del convertidor con la red de distribución.

Implementar Diferentes topologías de PLLs trifásicos mediante el uso de micro-controladores como el Arduino Due, Beaglebone Black, entre otros.

Diseñar e implementar un convertidor trifásico con la etapa de modulación y control, utilizando las diferentes topologías de PLLs trifásicos para realizar el acoplamiento y el control del convertidor con la red de distribución.

Bibliografía

- [1] S. Chandrasekaran and K. Ragavan, "Phase-locked loop technique based on sliding DFT for single phase grid converter application," in Proc. IEEE Int. Conf. Power Elect. Drives Energy Syst., 2012, pp. 1–4.
- [2] S. Golestan, M. Monfared, F. D. Freijedo, and J. M. Guerrero "Dynamics assessment of advanced single-phase PLL structures," IEEE Trans. Ind. Electron., vol. 60, no. 6, pp. 2167–2177, Jun. 2013.
- [3] M. Ciobotaru, R. Teodorescu, and F. Blaabjerg, "A new single-phase PLL structure based on second order generalized integrator," in Proc. IEEE 37th Power Electron. Spec. Conf., Jun. 2006, pp. 1511–1516.
- [4] F. D. Freijedo, J. Doval-Gandoy, O. Lopez, P. Fernandez-Comesana, and C. Martinez-Penalver, "A signal-processing adaptive algorithm for selective current harmonic cancellation in active power filters," IEEE Trans. Ind. Electron., vol. 56, no. 8, pp. 2829–2840, Aug. 2009.
- [5] V. Soares, P. Verdelho, and G. D. Marqués, "An instantaneous active and reactive current component method for active filters," IEEE Trans. Power Electron., vol. 15, no. 4, pp. 660–669, Jul. 2000.
- [6] H. Yang, M. Luo, X. Zhao, J. M. Guerrero, and L. Xu "Comparative performance evaluation of orthogonal signal generators based single phase PLL algorithms—A Survey" IEEE Trans. Power Electron., vol. 31, no. 5, pp. 3932-3944, May. 2016.
- [7] M. S. Reza, M. Ciobotaru, and V. G. Agelidis, "Accurate estimation of single-phase grid voltage parameters under distorted conditions," IEEE Trans. Power Del., vol. 29, no. 3, pp. 1138–1146, Jun. 2014.
- [8] M. S. Reza, M. Ciobotaru, and V. G. Agelidis, "Tracking of timevarying grid voltage using DFT based second order generalized intergrator techniques," in Proc. IEEE Int. Conf. Power Syst. Technol., 2012, pp. 1–6.

- [9] M. S. Reza, M. Ciobotaru, and V. G. Agelidis, "Instantaneous power quality analysis using frequency adaptive Kalman filter technique," in Proc. IEEE Int. Power Elect. Motion Cont. Conf., 2012, pp. 81–87.
- [10] S. M. Silva, B. M. Lopes, J. C. Filho, R. P. Campana, and W. C. Bosventura, "Performance evaluation of PLL algorithms for single-phase grid-connected systems," in Proc. Ind. Appl. Conf., Oct. 2004, vol. 4, pp. 2259–2263.
- [11] Z. Wang, S. T. Fan, Y. Zheng, and M. Cheng, "Control of a six-switch inverter based single-phase grid-connected PV generation system with inverse Park transform PLL," in Proc. IEEE Int. Symp. Ind. Electron., May 28–31, 2012, pp. 258–263.
- [12] K. Q. Z. Chiang and M. L. Psiaki, "Kalman filter tracking of limb scan signal using a bank of correlators," IEEE Trans. Aerosp. Electron. Syst., vol. 49, no. 1, pp. 118–133, Jan. 2013.