TUTORIAL
Implementation and Design of PLL and Enhanced PLL Blocks
April 2020
Phase-locked loop (PLL) has been widely used in many engineering applications. The primary function of the PLL is to generate a clean, unitary signal which is in synchronous with a noisy signal where the amplitude and frequency can change with time. This is required in almost all systems that are interfaced with the ac power grid and are intended to interact with it in an active and controlled manner. This includes grid-connected inverters, rectifier loads, motor drives, and battery chargers. PLL has also been extended to perform secondary tasks such as estimation of signal attributes including the magnitude, frequency, harmonics, sequence components, etc.

PLL is highly efficient and is very popular. The trade-off, however, is that parameter design is not always easy especially for those who do not have sufficient experience. This is due to the fact that PLL is inherently nonlinear, and no well-established and easily accessible rules for tuning the parameters has been developed.

This tutorial describes several conventional PLL blocks as well as enhanced PLL (ePLL) blocks implemented in PSIM for single-phase and three-phase applications. Simple and straightforward design guidelines to adjust the parameters of each PLL are presented. A simple test circuit is used to show the basic waveforms of the PLL block. Then a grid-connected inverter is used to illustrate how the PLL block is used in a practical application.

Four types of PLL blocks are presented in this tutorial:

- **PLL**: Conventional single-phase PLL blocks
- **ePLL**: Enhanced PLL blocks
- **3-phase PLL**: Conventional 3-phase PLL blocks
- **3-phase ePLL**: Enhanced 3-phase PLL blocks

Each block comes in two version: continuous time domain and discrete time domain.

1. **Conventional Single-Phase PLL**

Single-phase PLL has a very simple yet robust structure. However, due to the presence of large double-frequency ripples, it is not suitable for applications that require high level of accuracy. This section presents the PLL and introduces an effective way of tuning its parameters.

The block diagram of the PLL, with the definitions of signals, is shown below.

![Block Diagram of PLL](image)

where

- \( u(t) \) input signal to the PLL
- \( \omega_0 \) rated frequency of the input
- \( y(t) \) unitary output signal synchronous with the input
- \( y'(t) \) unitary output signal 90-degree delayed version of \( y(t) \)
Implementation and Design of PLL and Enhanced PLL Blocks

\[ \Phi \] estimated angle
\[ \omega \] estimated frequency in rad/sec.
\[ k_p \] proportional gain
\[ k_i \] integrating gain

Remarks

- Commonly, the point marked by \( x \) in the diagram above is considered as the estimated frequency. However, in this tutorial we consider the point marked by \( \omega \) as the estimated frequency. This point offers a more accurate estimate of the frequency with smoother dynamics.
- Information of the rated input magnitude \( A_o \) is used to tune the gains.

Design Guideline

1) Choose the natural frequency \( \omega_n \) as a small fraction of \( \omega_o \) where \( \omega_o \) is the rated frequency of the input. A typical choice can be:
\[ \omega_n = 0.1 \omega_o \]

2) Choose a reasonable value of the damping ratio \( \zeta \) where \( 0 \leq \zeta \leq 1 \). For example, \( \zeta = 0.5 \).

The selected values above make a good trade-off between the transient time and steady state ripples.

3) Calculate the parameters \( k_p \) and \( k_i \):

Use the previously selected values and the rated amplitude (peak value) of the input \( A_o \) to calculate the parameters.
\[ k_p = \frac{4\zeta \omega_n}{A_o} \]
\[ k_i = \frac{2\omega_n^2}{A_o} \]

Transient Time

The poles of the linearized PLL loop are at \(-\zeta \omega_n \pm j\omega_n\sqrt{1-\zeta^2}\). Therefore, for a good choice of \( \zeta \) (e.g. around 0.7), the time-constant of the response is around \( \tau = \frac{1}{\zeta \omega_n} \). For example, for \( \zeta = 0.5 \) and \( \omega_n = 0.1 \omega_o = 0.1 \times 377 = 37.7 \text{ rad/sec.} \), the time constant is around \( \tau = \frac{1}{0.5 \times 37.7} \) which is around 50 ms.

Double-Frequency Ripples

The major drawback of the conventional PLL is the presence of the double-frequency ripples at the output, as explained below. At the point denoted by \( z \) in the diagram in Page 2, a double-frequency component with the magnitude of \( A_o/2 \) exists due to the multiplication block.

The double-frequency ripples at the point \( \omega \) will be:
\[
\omega_{2f} = \frac{A_o k_i}{4 \omega_o} = 0.5 \omega_o \left(\frac{\omega_n}{\omega_o}\right)^2
\]

For instance, for a choice of \(\frac{\omega_n}{\omega_o} = 0.1\), the error is 0.5\% of \(\omega_o\) (or 0.005 pu). For a 60-Hz system, this is equal to 300 mHz (600 mHz peak-to-peak).

The double-frequency ripples at the point \(\Phi\) will be:

\[
\phi_{2f} = \frac{A_o}{4 \omega_o} \sqrt{k_p^2 + \frac{k_i^2}{4 \omega_o^2}} \approx 60 \zeta \left(\frac{\omega_n}{\omega_o}\right)
\]

For instance, for a choice of \(\frac{\omega_n}{\omega_o} = 0.1\), the error is equal to 6\(\zeta\) deg. For \(\zeta = 0.5\), the error of the phase angle is around 3\(^\circ\) (6\(^\circ\) peak-to-peak).

Analysis above shows that the transient time is inversely proportional to \(\zeta \omega_n\) where the angle ripple is directly proportional to \(\zeta \omega_n\). For the recommended values of \(\omega_n = 0.1 \omega_o\) and \(\zeta = 0.5\), the time constant is around 50 ms and the angle ripple is around 6\(^\circ\) peak-to-peak.

**PLL Block Implementation in PSIM**

Two blocks for conventional PLL, one in continuous time domain and the other in discrete time domain, are provided in PSIM as below:

<table>
<thead>
<tr>
<th>Continuous Domain</th>
<th>Discrete Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PLL</strong></td>
<td><strong>PLL</strong></td>
</tr>
<tr>
<td>(y)</td>
<td>(y)</td>
</tr>
<tr>
<td>(y')</td>
<td>(y')</td>
</tr>
<tr>
<td>(f_{eq})</td>
<td>(f_{eq})</td>
</tr>
<tr>
<td>(\phi) (in deg.)</td>
<td>(\phi) (in deg.)</td>
</tr>
</tbody>
</table>

The PLL block in the discrete domain (right) has the same image and terminal definition as the block in the continuous domain (left), except there is a character “\(z\)” at the upper right corner.

The PLL block parameters are:

- **Proportional Gain** \(k_p\)  Proportional gain in the PLL circuit
- **Integral Gain** \(k_i\)  Integral gain in the PLL circuit
- **Rated Input Frequency**  Rated frequency of the input, in Hz
- **Sampling Frequency**  Sampling frequency, in Hz (for the discrete domain block only)
Basic Test

A simple circuit is built to test the basic operations of the PLL block as shown below.

The input signal $u(t)$ is generated with various disturbances: phase disturbance at 0.2 sec., frequency change at 0.8 sec., and amplitude change at 0.8 sec. The waveforms are shown below.

The top panel shows the input signal $u(t)$, the unitary synchronous output $y(t)$, and the unitary quadrature output $y'(t)$. The lower panel shows the input signal frequency and the estimated frequency by the PLL.

Waveforms around 0.2 sec. are zoomed in for closer inspection, as shown below.
The following observations can be made from these waveforms:

- There is a steady double-frequency ripple with the peak-to-peak magnitude of about 600 mHz on the frequency and about 6° on the phase angle (as expected for the selection of $\zeta = 0.5$ and $\omega_n = 0.1\omega_o$).
- As expected, it takes about 50 ms for the system to approach the new operating point.
- When the magnitude of the input signal is increased, the peak-to-peak frequency error increases almost proportionally.

This simulation clearly confirms why the conventional PLL is not a suitable choice for applications which require fast responses (often below one cycle of 60 Hz or 50 Hz, i.e. 20 ms) and small errors (e.g. below half a degree in phase error and below 10 mHz in frequency error).

**Inverter Example**

A single-phase grid-connected inverter shown below is presented here to illustrate how PLL is used in a practical application. Files of this example can be found in “examples\PLL Blocks\single-phase”.
A proportional-resonant (PR) controller is used to control the inverter current. The transfer function of the PR controller is:

\[ G_{PR}(s) = k + \frac{k_r s}{s^2 + \omega_0^2} \]

In a control loop with ac commands and disturbances, such as this example, the PR controller has the advantage of removing the steady-state error completely. The feedforward term \( V_g \cdot y_{pll} \) cancels the grid voltage and provides a soft start of the inverter. Here, we assume the grid voltage amplitude \( V_g \) is known. In the enhanced PLL, this assumption is no longer needed.

The inverter provides a sinusoidal current at a desired magnitude in phase with the grid voltage \( v_c(t) \). The PR gain may be designed using root-locus approach, optimal control theory, or any other approaches (see the section below). The PLL design is provided in the parameter file. This is the same design that was discussed in the previous section: \( \zeta = 0.5, \omega_n = 0.1\omega_o \).

Waveforms of the inverter circuit are shown below.
The current reference magnitude changes from 10 A to 20 A at t=0.075 s, and then changes to 5 A at t=0.15 s. The top panel shows the current reference (red), the inverter current (green), and the reference magnitude (blue). The bottom panel shows the grid voltage.

For the sake of comparison, the PLL block is replaced with an ideal synchronization signal generator (which is a unit-magnitude voltage source in synchronous with the grid voltage), and two results are compared as shown below.

It is clearly observed that the inverter responses with the PLL are more distorted. The distortion is more pronounced at the low current magnitude. The distortion may be reduced at the expense of response speed to grid voltage variations.

Similar test circuit and single-phase inverter example are provided for the discrete domain PLL block, and the PSIM files can be found in “examples\Digital Control\PLL Blocks\single-phase”.
2. Single-Phase Enhanced PLL (ePLL)

Single-phase enhanced PLL (ePLL) was originally proposed in [1, 2]. An extensive coverage on ePLL can be found in the book [3].

The originality of ePLL is that, without changing the original and widely accepted structure of the conventional PLL, it adds a few simple blocks to remove its main drawback: double-frequency ripple. As a result of this improvement, ePLL also provides an estimate of the signal amplitude, fundamental component, and total distortions.

This section presents ePLL and introduces an effective way of tuning its parameters. Its implementation in PSIM is discussed. A simple test example and an inverter example are presented to illustrate the use of ePLL.

Block Diagram

The Block diagram of the EPLL implemented in the continuous time domain is shown below. Definition of the signals are given as follows.

\[
\begin{align*}
    u(t) & \quad \text{input signal to the PLL} \\
    \omega_0 & \quad \text{rated frequency of the input} \\
    y(t) & \quad \text{unitary output signal synchronous with the input} \\
    y'(t) & \quad \text{unitary output signal 90-degree delayed version of } y(t) \\
    A_0 & \quad \text{rated amplitude (peak value) of the input} \\
    A & \quad \text{estimated amplitude of the input} \\
    \Phi & \quad \text{estimated angle} \\
    \omega & \quad \text{estimated frequency in rad/sec.} \\
    y_f & \quad \text{estimated fundamental component} \\
    k_p & \quad \text{proportional gain} \\
    k_i & \quad \text{integrating gain} \\
    k_a & \quad \text{amplitude gain}
\end{align*}
\]

where

The division block does not exist in conventional structures. As a result, the setting of parameters depends on the amplitude of the input signal. Here, the division block is added to allow tuning of parameters independent from the input signal magnitude. The information of rated magnitude $A_0$...
is used only to avoid any possible divide-by-zero in the division block. The constant $E$ is suggested to be 0.001.

**Design Guideline**

1) Choose the first damping ratio $\zeta_1$ which determines the filtering strength of the ePLL. A smaller $\zeta_1$ means stronger filtering but slower dynamic response. We suggest a value range below for typical power applications:

$$0.25 \leq \zeta_1 \leq 0.75$$

2) Choose the second damping ratio $\zeta_2$ which corresponds to the damping ratio of the frequency loop. Since very fast frequency variation does not happen typically, can be set to a large value to avoid unnecessary oscillations. The recommended value range is:

$$1 \leq \zeta_2 \leq 2$$

3) Choose the constant $\lambda$. For typical applications, the suggested range of the constant $\lambda$ is:

$$0 \leq \lambda \leq 20$$

This will give a desired trade-off. A larger value of $\lambda$ means stronger suppression of the frequency swing. The default value is set to 10. If $\lambda=0$, this feature is disabled.

4) Set the parameter $k_p$ and $k_a$ as:

$$k_p = k_a = 2\zeta_1 \omega_0$$

5) Determine the integrating gain $k_i$ as:

$$k_i = \frac{k_p^2}{8\zeta_2^2}$$

The integrating gain $k_i$ may be made adaptive to further limit the range of frequency oscillations caused by phase angle jump or during the starting process [4]. With this approach, the $k_i$ equation is changed to:

$$k_i = \frac{k_p^2}{8\zeta_2^2} \times \frac{1}{1 + \frac{|e|}{\lambda A + \epsilon A_0}}$$

If $e \neq 0$ then use $\epsilon=0.001$ and the rest of the parameters to calculate the integration constant.

This will limit the spurious frequency estimation swings caused by phase jump at the cost of reducing the actual frequency estimation speed.
ePLL Block Implementation in PSIM

Two blocks for enhanced PLL, one in continuous time domain and the other in discrete time domain, are provided in PSIM as below:

The PLL block in the discrete domain (right) has the same image and terminal definition as the block in the continuous domain (left), except there is a character “z” at the upper right corner.

The ePLL block parameters are:

- **Proportional Gain** $k_p$  
  Proportional gain in the PLL circuit
- **Integral Gain** $k_i$  
  Integral gain in the PLL circuit
- **Rated Input Frequency** Rated frequency of the input, in Hz
- **Coefficient lambda** λ  
  Coefficient $\lambda$ as defined in the section above
- **Sampling Frequency** Sampling frequency, in Hz (for the discrete domain block only)

**Basic Test**

A simulation model is built to test the ePLL block in the basic operating conditions as shown below.

The input signal $u(t)$ is generated with various disturbances: phase disturbance at 0.2 sec., frequency change at 0.8 sec., and amplitude change at 0.8 sec. The waveforms are shown below.
The top panel shows the input signal $u(t)$, the unitary synchronous output $y(t)$, the unitary quadrature output $y'(t)$, and the fundamental output $y_f(t)$. The middle panel shows the input signal frequency and the estimated frequency. The lower panel shows the input signal amplitude and the estimated amplitude by the PLL block.

The following observations can be made from these waveforms:

- There is no double-frequency ripple.
- There is no steady state error or offset in any variables. All signal variations are tracked.
- The transient response is fast.

The absence of the double frequency ripple is the biggest improvement of the enhanced PLL block as compared to the conventional PLL block.
Inverter Example

A single-phase grid-connected inverter shown below is presented here to illustrate how the ePLL block is used in a practical application. PSIM files of this example can be found in “examples\PLL Blocks\single-phase”.

A proportional-resonant (PR) controller is used to control the inverter current. The PR controller has the advantage of removing the steady-state error completely. The feedforward term is the fundamental output $y_f(t)$, and it cancels the grid voltage and provides a soft start to the inverter. Note that in this case, the grid voltage amplitude information is no longer needed.

The inverter provides a sinusoidal current at a desired magnitude in phase with the grid voltage $v_c(t)$. The PLL design is provided in the parameter file. This is the same design that was discussed in the previous section: $\zeta_1 = 0.5$, $\zeta_2 = 1$, $\lambda = 10$.

Waveforms of the inverter circuit are shown below.
The current reference magnitude changes from 10 A to 20 A at t=0.075 s, and then changes to 5 A at t=0.15 s. The top panel shows the current reference (red), the inverter current (green), and the reference magnitude (blue). The bottom panel shows the grid voltage.

For comparison, the PLL block is replaced with an ideal synchronization signal generator (which is a unit-magnitude voltage source in synchronous with the grid voltage), and its result is compared with the result of the enhanced PLL and conventional PLL, as shown below.

The top panel shows the inverter current with the ideal synchronization signal and with ePLL. The bottom panel shows the inverter current with the ideal synchronization signal and with the conventional PLL. It is clearly observed that ePLL offers very similar result as the ideal synchronization signal with no distortion in the steady state. On the other hand, the waveform of the conventional PLL block is more distorted. The distortion is more pronounced at a low current magnitude. The distortions may be reduced at the expense of response speed to grid voltage variations.

Similar test circuit and single-phase inverter example are provided for the discrete domain PLL block, and the PSIM files can be found in “examples\Digital Control\PLL Blocks\single-phase”.
3. Conventional 3-Phase PLL

The conventional 3-phase PLL does not have the problem of the double-frequency ripple due to the fact that the ripples of three phases cancel each other. However, when the 3-phase inputs are not balanced, the double-frequency ripple will be present.

The block diagram of the conventional 3-phase PLL, with the definitions of signals, is shown below.

where

- $u(t)$: input signal to the PLL
- $\omega_o$: rated frequency of the input
- $y(t)$: unitary output signal synchronous with the input
- $y'(t)$: unitary output signal 90-degree delayed version of $y(t)$
- $\Phi$: estimated angle
- $\omega$: estimated frequency in rad/sec.
- $u_d$: d-axis component of the input
- $u_q$: q-axis component of the input
- $u_o$: zero-axis component of the input
- $k_p$: proportional gain
- $k_i$: integrating gain

The abc/dqo transformation is defined as:

$$
\begin{bmatrix}
u_d \\
u_q \\
u_o
\end{bmatrix} = \frac{2}{3} \begin{bmatrix}
cos(\phi) & cos(\phi - \frac{2\pi}{3}) & cos(\phi + \frac{2\pi}{3}) \\
sin(\phi) & sin(\phi - \frac{2\pi}{3}) & sin(\phi + \frac{2\pi}{3}) \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2}
\end{bmatrix} \begin{bmatrix}
u_a \\
u_b \\
u_c
\end{bmatrix}
$$

Remarks

- Commonly, the point marked by $x$ in the diagram above is considered as the estimated frequency. However, in this tutorial we consider the point marked by $\omega$ as the estimated frequency. This point offers a more accurate estimate of the frequency with smoother dynamics.

- The division block does not exist in conventional structures. As a result, the setting of parameters depends on the amplitude of the input signal. Here, we have added the division block to allow tuning of parameters independent from the input signal magnitude. The information of rated magnitude $A_o$ is used only to avoid any possible divide-by-zero in the division block. The constant $\epsilon$ is suggested to be 0.001.
**Design Guideline**

1) Choose the natural frequency $\omega_n$ as a small fraction of $\omega_o$:

$$0.1 \leq \frac{\omega_n}{\omega_o} \leq 0.5$$

2) Choose a reasonable value of the damping ratio:

$$0.5 \leq \zeta \leq 1$$

These values make a good trade-off between the transient time and filtering strength as explained below.

3) Calculate the parameters $k_p$ and $k_i$:

$$k_p = 2\zeta\omega_n$$

$$k_i = \frac{\omega_n^2}{\omega_o}$$

**Transient Time**

The poles of the linearized PLL loop are at $-\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$. Therefore, for a good choice of $\zeta$ (e.g. around 0.7), the time-constant of the response is around $\tau = \frac{1}{\zeta\omega_n}$. For example, for $\zeta = 0.7$ and $\omega_n = 0.25\omega_o = 0.25 \times 377 = 94.25$ rad/sec., the time constant is around $\tau = \frac{1}{0.7 \times 94.25}$ which is around 16 ms or one full cycle of 60 Hz.

**Double-Frequency Ripples**

The 3-phase PLL block does not have the double-frequency ripple problem when the three-phase input is balanced. However, when the input is unbalanced, the double-frequency ripple will be present.

Assume that the amplitudes of the positive-sequence and the negative sequence components of the input are $U^+$ and $U^-$ respectively. At the point $z$ in the 3-phase PLL diagram, the double-frequency component amplitude is $\frac{U^-}{U^+}$.

The double-frequency ripples at the point $\omega$ will be:

$$\omega_{2f} = \frac{k_i U^-}{2\omega_o U^+} = \frac{\omega_n^2 U^-}{2\omega_o U^+} = 0.5\omega_o\left(\frac{\omega_n}{\omega_o}\right)^2 \frac{U^-}{U^+}$$

For instance, for a choice of $\frac{\omega_n}{\omega_o} = 0.25$, and the imbalance factor of $IF = \frac{U^-}{U^+} = 0.04$, this corresponds to 75 mHz (150 mHz) error in a 60-Hz system.

The double-frequency ripples at the point $\Phi$ will be:

$$\phi_{2f} = \frac{IF}{2\omega_o} \sqrt{k_p^2 + \frac{k_i^2}{4\omega_o^2}} \approx 60\zeta\left(\frac{\omega_n}{\omega_o}\right)^2 * IF \quad \text{(deg.)}$$

For instance, for a choice of $\frac{\omega_n}{\omega_o} = 0.25$, and the imbalance factor of $IF = 0.04$, the error is equal to 6$\zeta$ deg. For $\zeta = 0.5$, the error of the phase angle is around 0.3° (0.6° peak-to-peak).
Analysis above shows that the transient time is inversely proportional to $\zeta \omega_n$ where the angle ripple is directly proportional to $\zeta \omega_n$. For the recommended values of $\omega_n = 0.25 \omega_o$ and $\zeta = 0.5$, the time constant is around 21 ms (just over one cycle of the 60-Hz system) and the angle ripple is around 0.6° peak-to-peak.

The integral gain $k_i$ may be made adaptive to further limit the range of frequency oscillations caused by phase angle jump or during the starting process. With this approach, the $k_i$ equation is changed to:

$$k_i = \omega_n^2 \frac{1}{1 + \lambda \frac{|u_d|}{|u_q| + \epsilon A_o}}$$

This will limit the spurious frequency estimation swings caused by phase jump at the cost of reducing the actual frequency estimation speed. For typical applications, the suggested range of the constant $\lambda$ is:

$$0 \leq \lambda \leq 20$$

This will give a desired trade-off. A larger value of $\lambda$ means stronger suppression of the frequency swing. The default value is set to 10. If $\lambda=0$, this feature is disabled.

**PLL Block Implementation in PSIM**

Two blocks for enhanced PLL, one in continuous time domain and the other in discrete time domain, are provided in PSIM as below:

The PLL block in the discrete domain (right) has the same image and terminal definition as the block in the continuous domain (left), except there is a character “z” at the upper right corner.

The PLL block parameters are:

- **Proportional Gain** $k_p$ Proportional gain in the PLL circuit
- **Integral Gain** $k_i$ Integral gain in the PLL circuit
- **Rated Input Frequency** Rated frequency of the input, in Hz
- **Rated Input Amplitude (peak)** Rated input amplitude $A_o$ (phase peak value)
- **Coefficient lambda** Coefficient $\lambda$
- **Sampling Frequency** Sampling frequency, in Hz (for the discrete domain block only)
Basic Test

A simple circuit is built to test the basic operations of the 3-phase PLL block as shown below.

The 3-phase input signals consist of positive, negative, and zero sequence components generated by three subcircuits. The negative and zero sequence circuits use the phase angle from the positive sequence circuit to ensure the same frequency, and they also have their own initial phase shifts.

The input signals have the following changes and disturbances:

- Phase disturbance at 0.1 sec.
- Frequency change at 0.3 sec.
- Amplitude change at 0.4 sec.
- Negative sequence component is applied at 0.5 sec.
- Zero sequence component is applied at 0.6 sec.

The waveforms are shown below.
The top panel shows the input signal $u_{abc}(t)$. The upper middle panel shows the input $u_a(t)$ and the unitary synchronous output $y(t)$. The lower middle panel shows the input frequency and the estimated frequency. The bottom panel shows estimated d-axis, q-axis, and zero-axis components of the input.

The following observations can be made from these waveforms:

- When the input signal is balanced, there is no double-frequency ripple.
- When the input signal contains the negative sequence component, there is a double-frequency ripple with the peak-to-peak magnitude of about 120 mHz on the frequency and about 0.6° on the phase angle (as expected for the selection of $\zeta = 0.5$ and $\omega_n = 0.25\omega_o$).
- It takes over a full cycle for the estimated output to reach the new steady state.

The 3-phase PLL has a simple and robust structure, and can track the frequency and angle changes smoothly. It does have error and double-frequency ripple under unbalanced conditions.
Inverter Example

A 3-phase grid-connected inverter shown below is presented here to illustrate how PLL is used in a practical application. Files of this example can be found in “examples\PLL Blocks\3-phase”.

The control is based on the dq frame. The 3-phase PLL block is used to generate the angle of the grid voltage. This angle is then used to convert the currents from the abc frame to the dq frame. PI controllers are used to control the currents. The system parameters and the design of the PI controllers as well as the PLL block are presented in the parameter file.

In this system, the operating conditions are changed as below:

- At t = 0.02 sec., the Iq reference is changed from 0 to 12.
- At t = 0.06 sec., the Id reference is changed from 0 to 8.
- At t = 0.1 sec., Phase a and b are shorted.

Waveforms of the inverter system are shown below.
The top panel shows the inverter currents. The middle panel shows the grid voltages. The bottom panel shows the currents $I_d$ and $I_q$ and their reference values. It can be observed that fast and smooth response is achieved. Also, after 0.1 sec. when there is a short circuit, there is a significant amount of double-frequency ripple.

Similar test circuit and 3-phase inverter example are provided for the discrete domain 3-phase PLL block, and the PSIM files can be found in “examples\Digital Control\PLL Blocks\3-phase”.

4. Enhanced 3-Phase PLL

The 3-phase enhanced PLL block has the capability to handle general 3-phase signals, regardless if they are balanced or not. It is able to decompose the signals into positive, negative, and zero sequence components. Furthermore, it provides all the information regarding the input signals, including frequency, amplitude, and phase angle.

The enhanced PLL block does not cause any error in the presence of negative and zero sequence components. This feature makes it ideally suited for applications where various devices are connected to a distribution system causing unbalance conditions to occur.

A detailed discussion on the technical background of the enhanced 3-phase PLL block can be found in [3].
PLL Block Implementation in PSIM

Two blocks for enhanced PLL, one in continuous time domain and the other in discrete time domain, are provided in PSIM as below:

The PLL block in the discrete domain (right) has the same image and terminal definition as the block in the continuous domain (left), except there is a character “z” at the upper right corner.

The PLL block parameters are:

- **Proportional Gain** $k_p$  
  Proportional gain in the PLL circuit

- **Integral Gain** $k_i$  
  Integral gain in the PLL circuit

- **Rated Input Frequency**  
  Rated frequency of the input, in Hz

- **Rated Input Amplitude**  
  Rated input amplitude $A_0$ (phase peak value)

- **Coefficient lambda**  
  Coefficient $\lambda$

- **Sampling Frequency**  
  Sampling frequency, in Hz (for the discrete domain block only)

**Design Guideline**

1. **Choose the first damping ratio** $\zeta_1$.
   
The first damping ratio $\zeta_1$ determines the filtering strength of the ePLL. A smaller $\zeta_1$ means stronger filtering but slower dynamic response. We suggest a value range below for typical power applications:
   $\quad 0.25 \leq \zeta_1 \leq 0.75$

2. **Choose the second damping ratio** $\zeta_2$.
   
The second damping ratio corresponds to the damping ratio of the frequency loop. Since very fast frequency variation does not happen typically, can be set to a large value to avoid unnecessary oscillations. The recommended value range is:
   $\quad 1 \leq \zeta_2 \leq 2$

3. **Calculate the parameters** $k_p$ and $k_i$ as:
   
   $k_p = \zeta_1 \omega_o$
4) Calculate the integrating gain $k_i$.

The integrating gain $k_i$ may be made adaptive to further limit the range of frequency oscillations caused by phase angle jump or during the starting process [4]. With this approach, the $k_i$ equation is changed to:

$$ k_i = \frac{k_p^2}{4\xi^2} \cdot \frac{1}{1 + \lambda \frac{|u_d|}{|A_p| + \epsilon A_0}} $$

This will limit the spurious frequency estimation swings caused by phase jump at the cost of reducing the actual frequency estimation speed. For typical applications, the suggested range of the constant $\lambda$ is:

$$ 0 \leq \lambda \leq 20 $$

This will give a desired trade-off. A larger value of $\lambda$ means stronger suppression of the frequency swing. The default value is set to 10. If $\lambda=0$, this feature is disabled.

**Basic Test**

A simple circuit is built to test the basic operations of the 3-phase PLL block as shown below.

The 3-phase input signals consist of positive, negative, and zero sequence components generated by three subcircuits. The negative and zero sequence circuits use the phase angle from the positive sequence circuit to ensure the same frequency, and they also have their own initial phase shifts.

The input signals have the following changes and disturbances:

- Phase disturbance at $t = 0.1$ sec.
- Frequency change at \( t = 0.3 \) sec.
- Amplitude change at \( t = 0.4 \) sec.
- Negative sequence component is applied at \( t = 0.5 \) sec.
- Zero sequence component is applied at \( t = 0.6 \) sec.

The waveforms are shown below.

The top panel shows the input signal \( u_{abc}(t) \). The upper middle panel shows the input \( u_{ad}(t) \) and the unitary synchronous output \( y(t) \). The lower middle panel shows the input frequency and the estimated frequency. The bottom panel shows estimated positive, negative, and zero sequence component amplitudes.

The following observations can be made from these waveforms:

- When the input signal contains the negative sequence or zero sequence components, there is no double-frequency ripple.
- The dynamic response is fast.

The result confirms that the enhanced PLL block not only removes the double-frequency error under unbalanced conditions, but it also provides additional information such as the magnitudes and phase angles of positive, negative, and zero sequence components.
Inverter Example

A 3-phase grid-connected inverter shown below is presented here to illustrate how PLL is used in a practical application. Files of this example can be found in “examples\PLL Blocks\3-phase”.

The control is based on the dq frame. The 3-phase PLL block is used to generate the angle of the grid voltage. This angle is then used to convert the currents from the abc frame to the dq frame. PI controllers are used to control the currents. The system parameters and the design of the PI controllers as well as the PLL block are presented in the parameter file.

In this system, the operating conditions are changed as below:

- At $t = 0.02\text{ sec.}$, the $I_q$ reference is changed from 0 to 12.
- At $t = 0.06\text{ sec.}$, the $I_d$ reference is changed from 0 to 8.
- At $t = 0.1\text{ sec.}$, Phase a and b are shorted.

Waveforms of the inverter system are shown below.
The top panel shows the inverter currents. The middle panel shows the grid voltages. The bottom panel shows the currents $I_d$ and $I_q$ and their reference values. It can be observed that fast and smooth response is achieved. Also, with the enhanced PLL, the inverter can continue to supply clean and balanced currents to the grid even under unbalanced conditions.

Similar test circuit and 3-phase inverter example are provided for the discrete domain 3-phase PLL block, and the PSIM files can be found in “examples\Digital Control\PLL Blocks\3-phase”.
**Applications**

1) Induction Heating

Induction furnace based on AC/DC and DC/AC conversion stages. The rectifier pulse generator contains a PLL to produce equidistant pulses for all six thyristors by means of single-phase synchronism. The PLL of the inverter generates the pulses to the IGBT’s according to the signal generated by the circuit or resonant load (RLC).
2) Control structure for three-phase inverter connected to the grid

The figure shown is a general structure for the natural reference frame control strategy, where energy from a renewable energy source (DC Bus) can be delivered to the grid. The DC-link voltage controller is used for balancing the power flow in the system. Phase angle of the grid voltages is provided by a PLL. This estimate is required to create the three current references. Each value is compared with the corresponding current measurement. The generated error is the input to each hysteresis controller. The output of these blocks is the switching state for the devices in the power inverter. The inverter controller can be used to control the active and reactive energy generated to the grid, the dc-link voltage regulation, to ensure the high quality of the injected energy, and the grid synchronization.

References


