University of Alberta

Adaptive Phase Synchronization Techniques for Unbalanced and Distorted Three-Phase Voltage System

by

Alexis Woinowsky-Krieger

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Power Engineering and Power electronics

Department of Electrical and Computer Engineering

Fall 2010

Edmonton, Alberta

Examining Committee

Dr. John Salmon, Electrical and Computer Engineering

Dr. Venkata Dinavahi, Electrical and Computer Engineering

Dr. Yunwei Li, Electrical and Computer Engineering

Dr. Stevan Dubljevic, Chemical and Materials Engineering

Dr. Ed Nowicki, Electrical and Computer Engineering, University of Calgary

ABSTRACT

Interfacing and operating AC power electronic systems requires rapid and accurate estimation of the phase angle of the power source, and specifically of the positive sequence of the three-phase utility grid voltage. This is needed to ensure reliable operation of the power control devices and of the resulting power flow. However, the quality of this information is undermined by various distortions and unbalanced conditions of the three-phase grid voltage. Phase estimation and power control can both be performed in real time by a DSP, but a DSP typically has limited computational resources, especially in regards to speed and memory, which motivates the search for computationally efficient algorithms to accomplish these tasks. In contrast to conventional PLL techniques, recent approaches have used adaptive amplitude estimation to enhance the acquisition of the phase information, resulting in faster response and improved performance.

This thesis presents a novel technique to estimate the phase of the positive sequence of a three-phase voltage in the presence of frequency variations and unbalanced conditions, referred to as hybrid negative sequence adaptive synchronous amplitude estimation with PLL, or H-NSASAE-PLL. The key feature consists of a feedback structure which embeds a positive sequence PLL and an adaptive synchronous negative sequence estimator to enhance the performance of the PLL. The resulting benefits include faster estimation of the phase of the positive sequence under unbalanced conditions with zero steady state error, simplified tuning of PLL parameters to address a wide range of application requirements, robust performance with respect to distortions and PLL parameters, a structure of minimal dynamical order (fifth) to estimate the main signal parameters of interest, simplified discretization, and reduced computational costs, making the proposed technique suitable for real time execution on a DSP.

The H-NSASAE-PLL is developed in the Matlab/Simulink environment, and a specialized test signal generator is developed to evaluate it's performance. The overall system is executed, and experimental results are produced, in real time, on a dSPACE DS1104 controller board.

ACKNOWLEDGEMENTS

An endeavor such as this cannot be accomplished without a lot of support from some key people, and I owe much gratitude to many for helping make this possible.

First of all, I wish to express my deepest gratitude to my supervisor, Dr. John Salmon, for his ongoing support and encouragement from the very beginning. Truly, I could not have completed this work if not for his patience, enthusiasm, insight, and assistance in regards to many technical issues and difficulties that inevitably arise in such a pursuit.

I must also thank my dear parents, who have also been a most important source of moral support and encouragement in this personal journey towards a higher standard of excellence than I might have otherwise been inclined to settle for.

I am also very grateful to Albert Terheide for his technical assistance and support to ensure the reliable operation of the equipment on which the work for this thesis has depended.

Last, and most important of all, I am immensely indebted to my dear wife, Lilia, for her enduring and supportive patience and love, without which I would no doubt have given up long ago.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AFC	Adaptive feed-forward control
ANF	Adaptive notch filter
APF	Active power filter, or all-pass filtering
ASAE	Adaptive synchronous amplitude estimation
ASAE-NSF	ASAE-based negative sequence filter (PSF)
ASAE-NSGI	ASAE-based negative sequence generalized integrator
ASAE-PNSF	ASAE-based positive-negative sequence filter (PNSF)
ASAE-PSF	ASAE-based positive sequence filter (PSF)
ASAE-PSGI	ASAE-based positive sequence generalized integrator
ASAE-QSG	ASAE-based quadrature signal generator (QSG)
ASC	Adaptive signal cancellation
BPF	Band -pass filter
DFT	Discrete Fourier transform
DSP	Digital signal processor (or processing)
EPLL Enhanced PLL	
FFT	Fast Fourier transform
FIR	Finite Impulse Response
FLL	Frequency-locked loop
FPGAField programmable gate array	
FRF	Fixed reference frame
H-NSASAE-PLL	Hybrid negative sequence adaptive sequence amplitude estimation with PLL
H-NSF-PLL	Hybrid negative sequence filter with PLL
IMP	Internal model principle
LPF	Low-pass filter
LTI	Linear time-invariant
LTV	Linear time-varying
M/PLL	Magnitude/phase-locked loop
NSC	Negative sequence calculation
NSF	Negative sequence filter
NSGI	Negative sequence generalized integrator
PD	Phase detector
PI	Proportional plus integral controller
PLL	Phase-locked Loop
PNSC	Positive-negative sequence calculation
PNSF	Positive-negative sequence filter
PSC	Positive sequence calculation
PSF-PLL	Positive sequence filter (PSF) based on a PLL
PSGI	Positive sequence generalized integrator
PSGI-PLL	PSGI-based PLL
PSF	Positive sequence filter
QASAE	Quadrature-signal ASAE
QSG	Quadrature signal generator
SOGI	Second order generalized integrator
SRF	Synchronous reference frame
VCO	Voltage controlled oscillator

LIST OF SYMBOLS

A	Signal peak amplitude
$\hat{A}$	Estimated amplitude
ε	Error signal
ε_α, ε_β	Estimate error of α and β signals
f	Signal frequency, Hz
$\hat{f}$	Estimated frequency
H_{AP}	All-pass filter transfer function
H_{LPF}	Low-pass filter transfer function
H_{BPF}	Band-pass filter transfer function
$I(t)$	VCO or filter I output, nominally in-phase with input signal
K	Band-pass filter K factor; $K = 2\zeta$
K_A	Gain for adaptive synchronous amplitude estimation algorithm
K_I	Integral gain for P-I controller of PLL
K_n	K factor for negative sequence filter
K_o	VCO constant, Hz per input unit
K_p	K factor for PSF, or proportional gain of PI controller
K_S	Speed gain for modified P-I controller of PLL
θ_e	Phase error
θ_i	Input signal phase
θ_o	Output signal phase
$\hat{\theta}_p$	Estimated phase angle of positive sequence
Q	VCO or filter Q output, nominally in-quadrature with input signal
$\mathbf{v}_{abc}$	three-phase voltage vector signal
$v_{\alpha p}$, $v_{\beta p}$	α and β components of positive$\alpha\beta$ sequence signal
$v_{\alpha n}$, $v_{\beta n}$	α and β components of negative $\alpha\beta$ sequence signal
$\hat{v}_\alpha$, $\hat{v}_\beta$	estimated α and β components of $\alpha\beta$ signal
$\hat{v}_{\alpha\beta}$	estimated $\alpha\beta$ vector signal
v_d, v_q	Park transformation d and q output signals
ω	Signal frequency, rad./sec.
$\hat{\omega}$	Estimated frequency
ω_n	Natural un-damped frequency of second order transfer function
ω_o	nominal frequency of signal or PLL
ζ	Zeta: damping factor of second order transfer function

1. INTRODUCTION

Interfacing and operating AC power electronic systems requires rapid and accurate estimation of the phase angle of the power source, and specifically of the positive sequence of the three-phase utility grid voltage. This is needed to ensure reliable operation of the power control devices and of the resulting power flow. For a three-phase AC power source, this phase angle is often estimated with a conventional two-phase PLL (after first converting the three-phase signal vector to a two-phase equivalent using a Clarke transformation). However, various distortions and unbalanced conditions of the line voltage adversely influence the quality of the PLL phase estimate, which motivates ongoing research to minimize the impact of such distortions.

In the last decade or so, so-called adaptive techniques have been proposed to improve the quality of frequency and phase estimation. These novel techniques can be generally characterized by some algorithm which explicitly generates an estimate of the fundamental component of interest in the distorted input signal. Using this estimate, an error is calculated, which is then used to adaptively drive the algorithm to minimize the error. In the specific reference case of an un-distorted sinusoidal input, this error signal can be driven to exactly zero, and quite fast, nominally in under two to three cycles.

This thesis proposes to further extend these adaptive techniques to enhance the performance under unbalanced conditions. A novel PLL algorithm is presented, referred to as a hybrid PLL with negative sequence adaptive synchronous amplitude estimation (H-NSASAE-PLL), which offers a number of important benefits. To put the proposed algorithm in perspective, this chapter presents a brief overview of some representative applications, the fundamental objective of phase estimation, common sources of distortion, existing alternative approaches to this estimation task, and how this background leads to current and to the proposed adaptive techniques to achieve this objective. The chapter

concludes with an overview of the key features of the proposed H-NSASAE-PLL, it's principal benefits, and of the main contributions of this thesis.

1.1 THE BASIC TASK OF PHASE ESTIMATION

The need for phase estimation and synchronization using PLL techniques arises in a very wide range of applications in power electronics, including:

a) triac and thyristor based power converters [1];
b) real time signal analysis, including harmonic detection, reactive power estimation, zero crossing detection, and for power systems protection, [2];
c) power conditioning, reactive power compensation, power factor correction, PWM voltage converters, [3];
d) active power filtering and harmonic distortion compensation in small rating stand-alone power grids, such as on ships and oil rigs, [4];
e) fault detection and phase tracking through voltage sags and unbalances due to line faults, [5];
f) voltage sag characterization, [6-8];
g) distributed power generation, including wind generation and photovoltaic power generation systems, [9-10].

The basic objective of phase estimation may be described as in Fig. 1.1. A sinusoidal voltage at a fundamental frequency, e.g. 60 Hz, may be distorted by various phenomena, and it is required to estimate the phase angle of the fundamental sinusoidal component. Thus, the required signal processing may be represented by the functional block diagram in Fig. 1.2. Active power is delivered from the AC source to the load by the fundamental sinusoidal component, and it therefore defines the reference frame, in the form of the estimated phase angle output $\hat{\theta}(t)$, for all pertinent information and timing of switching control devices. The internal structure of the estimation process depends on various factors, requirements, and constraints. The voltage frequency

may be very stable, thus allowing a slower estimation process, while achieving greater estimation accuracy. If the frequency is anticipated to vary considerably, then a faster response may be required, thus making the estimation more susceptible to various noise and distortion components. In power electronic applications, phase estimation must be accomplished in real time, which further limits the amount of computational effort available, and the resulting quality of the estimated phase signal.

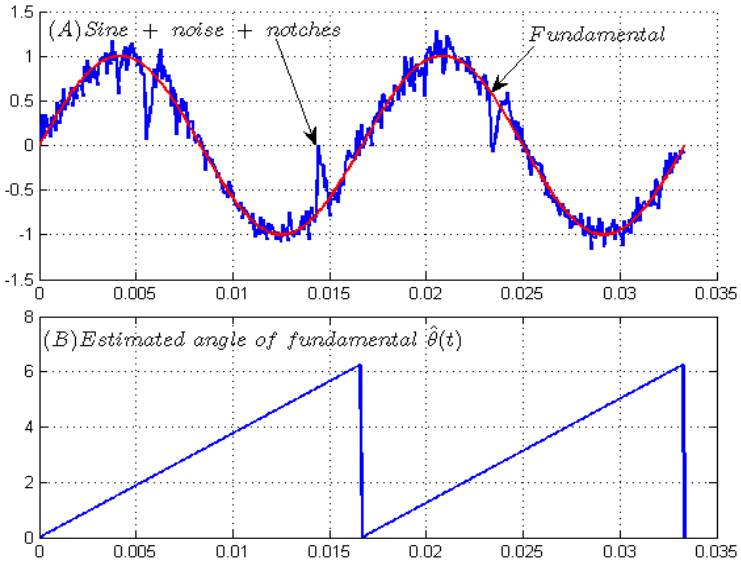


Figure 1.1 – (A) Distorted sinusoid and (B) estimated phase angle

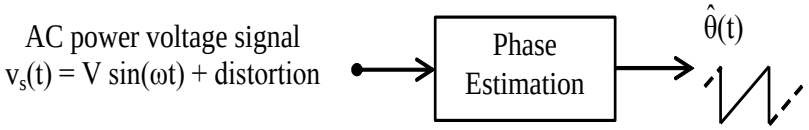


Fig. 1.2 – Phase estimation process

1.1.1 Signal Distortions and Filtering

The phase estimation process is typically challenged by various distortions, including:

a) switching transients due to switching power electronic devices, which may result in deep, narrow notches in the signal, [1, 2];

b) harmonics in the line voltage, due to non-sinusoidal currents resulting from non-linear loads, such as rectifiers, which cause distorted voltage drops across the line impedance, [3];
c) phase jumps, due to loads being connected or dis-connected to or from, the line;
d) frequency fluctuations, especially with isolated power grids, [4];
e) voltage sags, for the same reason as in (c), [5-9];
f) unbalanced condition of the three-phase grid voltage, due to unbalanced loads or fault conditions, [7-8].

Thus, the essential challenge of phase estimation is to produce an estimate of the phase of the fundamental sinusoidal component of the distorted input signal, as fast as possible, while simultaneously minimizing the estimate error. This is not unlike the general filtering problem, and is likewise faced with the same trade-off between speed of response vs. attenuation of distortion and estimation accuracy [10]. In general, the solution to attenuating the impact of various distortions involves some sort of filtering, which may take one or more various forms, depending on the specific kind of distortion being addressed, such as:

a) low-pass filtering (LPF): to attenuate all signal components above some cut-off frequency, chosen according to the required speed of response, [9];
b) band-pass filtering (BPF): also referred to as Space Vector Filtering (SVF) [9, 10] to select one particular component of interest, e.g. the fundamental, while retaining it's phase, and attenuating all other components;
c) all-pass filtering (APF): to provide specific phase shifting at a specified frequency; e.g. quadrature signals can be shifted by +/-90^{o} to subsequently extract balanced positive and negative sequence signals, [10-13];
d) notch filtering: to reject a component at a specific frequency, such as a double-frequency component from a multiplier-type phase detector, [14];
e) comb filtering: to reject all harmonic components, up to half the sampling frequency;

f) adaptive FIR filtering: which adjusts the filter coefficients to make it's frequency-selective response track the fundamental frequency, [15-16].

Any type of filtering in general results in slowing down the transient dynamic response, but with the desired benefit that the filter is designed for, of reduced error in steady state operation in response to the distortion being targeted by the chosen filter.

1.1.2 Performance Evaluation of the Phase Estimation Process

To assess the quality of the phase estimation process, standard tests to evaluate it's performance include [2, 9, 17, 19-23]:

a) subjecting it to a sinusoidal signal with a phase step, e.g. 25^{o}, and measuring the resulting phase error and transient behaviour;
b) changing the signal frequency in a step fashion, such as from 60 to 65 Hz, and measuring the estimated frequency response, phase deviation, and overall transient dynamic behaviour;
c) changing the signal amplitude, e.g. by +0.5 pu, and measuring the impact on the estimated frequency, phase, and amplitude estimates, as required;
d) adding some harmonic distortion, such as 20% of the fifth for three-phase test, and measuring the impact on the estimated frequency and phase, and on the estimated positive and negative sequence components and amplitudes; the fifth harmonic is chosen for three-phase evaluation, for being the most significant, and therefore the most indicative;
e) introducing unbalance in the three-phase signal, and measuring the estimated frequency and phase deviations, and the estimated negative sequence component and amplitude.

1.2 CLASSIFICATION OF PHASE ESTIMATION AND PLL ALGORITHMS

To put the proposed algorithm in perspective, a brief overview is presented of various phase estimation techniques in current usage. These may be categorized according to various factors, including:

a) computational requirements,
b) real time or non-real time implementation,
c) sampling rate control,
d) adaptivity to variations in the signal fundamental frequency.

The last factor serves to distinguish a phase estimation technique as either open-loop or closed-loop [10], according to whether it attempts to adaptively estimate the signal phase by negative feedback, and is thus able to track the signal frequency. A new category of phase estimation algorithms is introduced, referred to as adaptive, which extracts more information from the input signal, and thus further improves the overall performance over the open-loop and closed loop categories.

1.2.1 Open Loop Phase Estimation

Open loop phase estimation is characterized by the absence of any feedback corrective mechanism in the estimation process, as exemplified in Fig. 1.3.

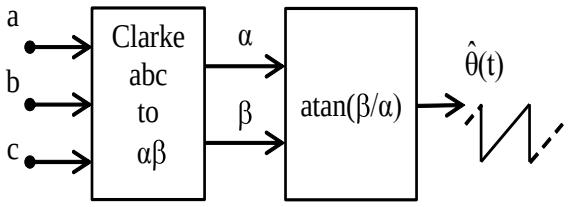


Fig. 1.3 – Open loop phase estimation

The three-phase abc vector signal is transformed into a quadrature or two-phase αβ signal, and then an elementary arc-tangent calculation is performed to produce the phase angle estimate of the vector signal. The principal objection to this

approach is that any noise or distortion in the input signal seriously degrades the estimated output angle signal, as demonstrated in Fig. 1.4, traces (A) and (B). Most notably, near the zero angle, the estimate is extremely sensitive to noisy zero crossings of the β signal, which significantly reduces the value of the estimated phase signal. Another important objection is the implication in the noisy angle signal, that the phase may at times reverse direction, which is inconsistent with the reality of the voltage source, regardless of the level of noise.

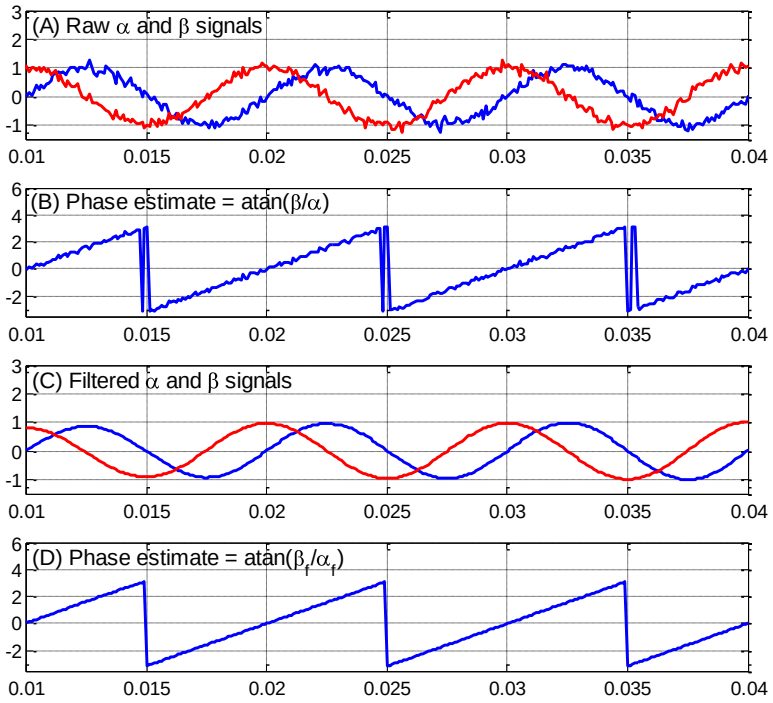


Fig. 1.4 – Open loop phase estimation; (A) Noisy quadrature αβ signals (B) Phase estimate of noisy αβ signals (C) Filtered α_f-β_f signals (D) Phase estimate from $\alpha_f\beta_f$ signals

An obvious solution to this problem is to filter the signal, as suggested in Fig. 1.5, which results in signals in Fig. 1.4, traces (C) and (D). The filter bandwidth can be reduced, although this also slows down the filter response. However, the same objections arise as before, even if to a lower degree, due to the random character of the noise.

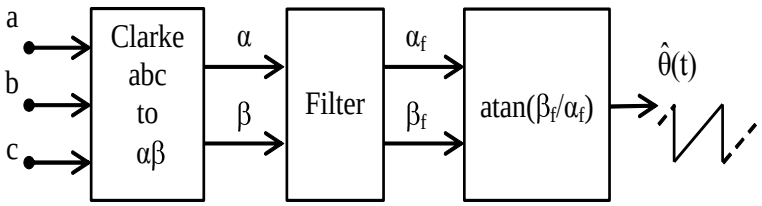


Fig. 1.5 – Open loop phase estimation with pre-filtering

Signal filtering can be easily extended to estimate the phase angle of a single-phase signal, as in Fig. 1.6, where a band-pass filter can be used to generate in-phase and quadrature, signals, suitably filtered to attenuate the noise and distortion components, [24-27]. However, for such a filter with a nominal fixed center frequency, the output responses will vary in phase as the input signal frequency varies, and result in a faulty phase estimate signal.

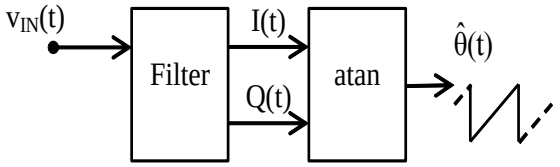


Fig. 1.6 – Phase estimation for a single-phase signal

1.2.2 Zero Crossing Detection Based Phase Estimation

The zero-crossing detection method can be represented as in Fig. 1.7. The phase angle is represented by a discrete value from a counter, which is clocked by a high frequency digital signal. The input signal $v_{IN}(t)$ is applied to a comparator, which produces a two-state signal, according to the polarity of $v_{IN}(t)$, and this signal is subsequently used to reset the counter and count N at every positive-going zero crossing of $v_{IN}(t)$. The estimated phase can be expressed in terms of the count N as: $\hat{\theta}(t) = K_N \times N$, where the constant K_N depends on the clock frequency and fundamental frequency of $v_{IN}(t)$. Although this method is relatively simple, there are a few problems, including [1, 15-16, 28-29]:

a) noise in the input signal will affect the timing of the zero crossing and the reset time of the counter, thus corrupting the resulting phase estimate;

b) the phase estimate relies on the constancy of the input signal frequency, and does not accommodate any variation in this frequency;

c) the information about the input signal phase is really only available at it's zero crossings, thus at most twice per cycle, which makes it impossible for this scheme to respond to any sudden phase jumps within a cycle [9-10].

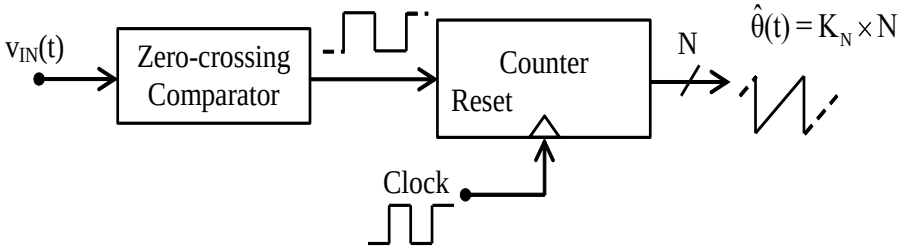


Fig. 1.7 – Zero-crossing detection method of phase estimation

1.2.3 DFT, Least Squares, and Kalman Filtering

The discrete Fourier transform (DFT), it's fast Fourier transform (FFT) variation, the least squares estimation method, and Kalman filtering approaches are significantly more computationally demanding, [10], and not realistically well suited for real time implementation on digital signal processors typically designed for real time power control, such as Texas Instruments' TMS320 family of DSP's, which must typically perform other control oriented computations, all at significantly high sampling rates, such as 10 KHz or higher. The DFT is further restricted by the requirement of an integer number of samples per fundamental period of the input signal, and thus assumes a constant input signal frequency, [31-33]. Although the sampling rate could be varied to satisfy this requirement, [32-33], this practice is not so common, as it also affects other computations that are typically performed by a DSP and which assume a constant sampling rate. The DFT also requires a significant amount of memory to hold the input signal samples, namely N samples per cycle, which depends on the sampling rate.

Although this is not at all un-realistic, it does contribute to the overall computational cost. For these principal reasons of computational cost and real time implementation on representative power control-oriented DSP's, these methods are not considered for the purpose of real time phase estimation.

1.2.4 Closed Loop Phase Estimation

In contrast to open-loop phase estimation, the closed-loop approach is structured such as to able to accommodate variations in input signal frequency by explicitly estimating it's phase. A generic block diagram of a closed loop phase estimation process is represented in Fig. 1.8.

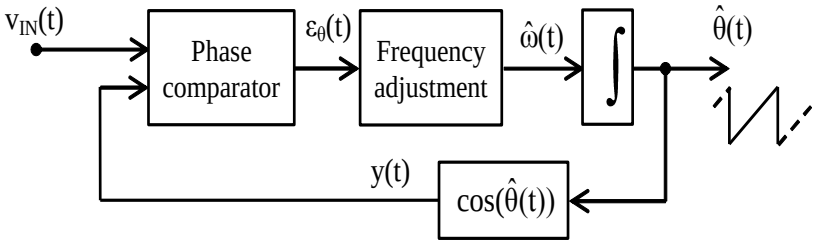


Fig. 1.8 – Closed loop phase estimation

This approach is characterized by the generation of a sinusoidal function, y(t), based on the estimated phase angle $\hat{\theta}(t)$, and the comparison of this signal with the input signal $v_{IN}(t)$, by some means, which produces a phase error signal $\varepsilon_\theta(t)$. This then results in a corrective action on the phase estimate to make it's frequency track the input signal frequency [2, 10, 35-38]. The input signal could be single-phase or multi-phase, and depending on the phase comparison technique used, $\varepsilon_\theta(t)$ could also contain additional harmonic components, due to distorted and unbalanced input signal conditions. Some means of eliminating or attenuating some, or all, of these harmonic components may become necessary, as exemplified in the decoupled double synchronous reference frame PLL [22-23].

1.2.5 Adaptive Phase Estimation Techniques

An important question to consider in the overall estimation process concerns what information is available in the input signal, or can be also estimated, in order to improve the overall phase estimation process. Thus, in contrast to the open-loop and closed-loop techniques of phase estimation, which effectively address only the phase and frequency of the input signal, adaptive techniques further consider the amplitude of the input component of interest. Thus, adaptive techniques are characterized by an explicit generation of an estimate of the fundamental component of interest in the distorted input signal, as indicated in Fig. 1.9.

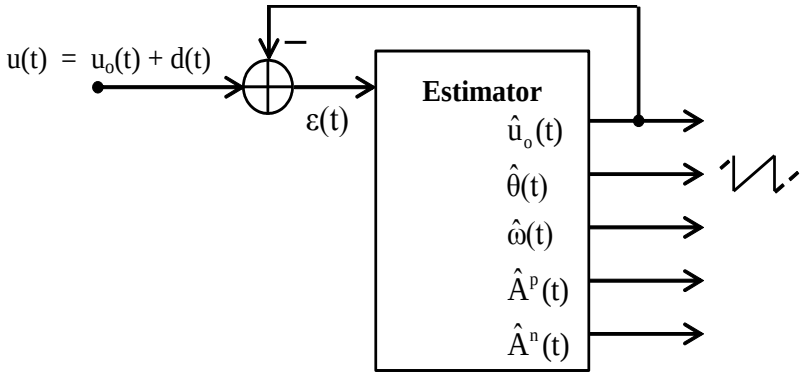


Fig. 1.9 – General adaptive phase estimation

The input signal u(t) is defined in terms of a fundamental un-distorted component $u_o(t)$, and a component of noise and distortion d(t):

$$u(t) = u_o(t) + d(t) \qquad u_o(t) = V\cos(\omega t)$$

and the overall objective is to estimate the component $u_o(t)$, which is accomplished by indirectly estimating the phase of the input fundamental component. The estimated signal is then subtracted from the input signal to produce an estimate error ε(t). This error signal subsequently drives the algorithm to minimize this error. In the specific reference case of an un-distorted signal u(t) = $u_o(t) = V\cos(\omega t)$, this error signal can be driven to exactly zero in steady state operation, and relatively fast, on the order of a few cycles, and even under two cycles, depending on overall design criteria and signal conditions. Such an adaptive approach necessarily entails estimating the fundamental component amplitude, which provides the additional information about the input, with which to effectively speed up the overall estimation process. Although Fig. 1.9 suggests

an estimator for a single-phase input signal, the approach can be extended for a two- or three-phase input signal, such that the signal u(t) can in general be considered a vector signal **u**(t).

1.3 PROPOSED H-NSASAE-PLL TECHNIQUE

In view of current techniques of estimating the phase angle of the positive sequence of an unbalanced grid voltage, a novel technique is proposed, referred to as hybrid PLL with negative sequence adaptive synchronous amplitude estimation (H-NSASAE-PLL), with the following features:

a) a negative feedback structure with two embedded estimating filters, one each to estimate the positive and negative sequence components of an unbalanced quadrature signal;
b) the overall operation can be described in terms of the well-known internal model principle (IMP);
c) the positive sequence estimator consists of a conventional two-phase PLL, with additional synchronous estimation of the positive sequence amplitude;
d) the negative sequence estimator uses the estimated, unit amplitude and balanced, quadrature signals from the above PLL, to synchronously estimate the negative sequence component of an unbalanced quadrature signal.

This novel structure results in the following important benefits:

a) the negative sequence estimator serves to cancel the negative sequence component in an unbalanced two-phase input signal, resulting in faster and more accurate estimation of the positive sequence, with zero steady state error with respect to an unbalanced input;
b) minimal dynamical order required to estimate all the fundamental quantities, namely the positive sequence fundamental frequency, phase, and amplitude, and of the negative sequence quadrature components;

c) the estimation processes, implemented in synchronous reference frame for both positive and negative sequence components, can be easily discretized using lowest order Euler integrators, with one integrator for each estimated quantity;
d) such simplified discretization, in addition to the minimal dynamical order for estimation, reduces overall computational demands, and thus makes it well suited for real time execution on a typical DSP with limited computational resources;
e) the proposed structure simplifies tuning the operating parameters according to the requirements for any given application;
f) the overall performance is robust with respect to operating parameters and distorted and unbalanced signal conditions.

1.4 CONTRIBUTIONS OF THE THESIS

In addition to the H-NSASAE-PLL, a few additional novel estimation techniques are presented, which serve to establish a necessary frame of reference for the proposed H-NSASAE-PLL:

a) Positive-negative sequence filter (PNSF)

   This establishes a fundamental underlying principle of operation of, and a mathematical foundation for, the H-NSASAE-PLL, whereby a positive and negative sequence estimators can be combined within a common feedback structure, according to the internal model principle. The mathematical foundation for the PNSF is established in terms of transfer functions in the familiar s-domain.

b) PLL-based positive sequence filter (PSF-PLL)

   This consists of a conventional positive sequence filter (PSF), implemented in synchronous reference frame, and centered on a conventional two-phase PLL. The result is a frequency-adaptive PSF with minimal dynamical order. This also establishes a useful mathematical basis for taking a simple PSF,

normally described in the Laplace s-domain, in stationary (or fixed) reference frame (FRF), and implementing it in synchronous reference frame (SRF). The very significant benefit is the ability to realize the equivalent functionality at lowest computational cost, in real time, on a typical DSP, with robust performance.

c) Hybrid PLL with negative sequence filter (H-NSF-PLL)

This consists of a PSF-PLL and an additional resonator, in SRF, to estimate the negative sequence, and effectively implements a frequency-adaptive PNSF. If implemented with a higher order differential equation solver, such as during the simulation phase of development, this clearly validates the principle of operation. However, this is computationally more demanding, and possibly not realizable for real time execution on a DSP, and thus motivates the H-NSASAE-PLL, which implements the negative sequence estimation in SRF.

Finally, a comprehensive test signal generator is also developed for evaluating the performances of the proposed H-NSASAE-PLL and other techniques, which allows for convenient analysis and comparison. The test program, consisting of the signal synthesizer and the PLL's under evaluation, executed in real time, is presented in chapter 4. The results, all observed in real time, and presented in chapter 5, show the superior performance of the proposed H-NSASAE-PLL. The overall test program in developed using the Matlab/Simulink platform, and executed in real time on the dSPACE DS1104 control board.

2. FREQUENCY AND PHASE ESTIMATION TECHNIQUES

This chapter presents underlying principles pertinent to conventional and adaptive methods of synchronization in current usage, as discussed in recent literature. These principles lead to the novel approaches described in chapter 3, and specifically to the proposed H-NSASAE-PLL technique to address unbalanced signal conditions. Table 2.1 outlines the subjects addressed in this chapter, with each one contributing a specific feature in the final H-NSASAE-PLL technique.

Table 2.1 – Underlying principles

	Subject	Key features
1.	conventional single-phase PLL	➢ fundamental core to many PLL's ➢ establishes small signal model and stability ➢ suffers from double-frequency ripple
2.	Park transformation	➢ converts two-phase signal between fixed (stationary) and synchronous reference frames
3.	conventional two-phase PLL	➢ uses Park transformation to eliminate double-frequency ripple with balanced input ➢ establishes small signal model and stability
4.	quadrature signal generation	➢ to generate $\pi/2$-shifted sinusoidal signals, suitable for Park transformation
5.	adaptive notch filter	➢ frequency estimation and tracking
6.	Clarke transformation	➢ To convert from three-phase to two-phase signal representation
7.	positive and negative sequences	➢ useful de-composition of unbalanced three-phase and two-phase signals
8.	internal model principle	➢ zero steady state error when estimating a specific signal
9.	adaptive synchronous amplitude estimation	➢ useful for robust discretized implementation of H-NSASAE-PLL technique

Table 2.1 – Underlying principles, continued

10.	EPLL	➢ Single-phase Magnitude/Enhanced PLL, based on single-phase PLL and ASAE ➢ establishes stability and fast convergence of adaptive phase and amplitude estimation
11.	ASAE and IMP equivalence	➢ equivalence of ASAE to band-pass filter (BPF) ➢ dynamic properties of BPF applicable to ASAE

2.1 CONVENTIONAL SINGLE-PHASE PLL OPERATION

A generic structure of a PLL is introduced, and then the specific case of the conventional single-phase PLL is discussed, which serves as a starting point for subsequent enhancements reflected in current adaptive frequency and phase estimation techniques. The conventional linearized model is presented, it's idealized dynamic characteristics, and it's behaviour is emphasized with regards to the inevitable double frequency ripple. The latter defines the perspective for the quadrature-phase conventional PLL and it's improved performance over the single-phase PLL.

2.1.1 PLL Block Diagram and Fundamental Theory

In the overall perspective of estimating the phase of a periodic signal, the problem can be described as depicted in Fig. 2.1

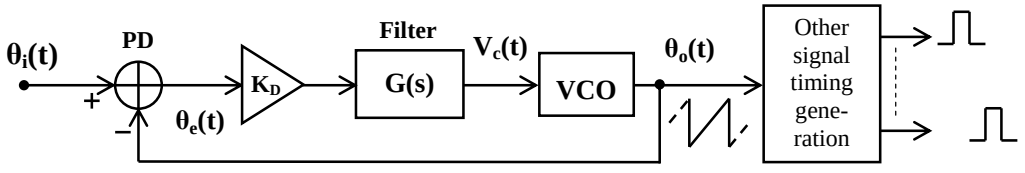


Fig. 2.1 – Basic PLL structure

Fig. 2.1 emphasizes the following principal signals and required functional blocks [10, 22-25]:

a) the main signal of interest consists of the phase $\theta_i(t)$, which is somehow represented by the actual, nominally periodic, input signal, such as a sinusoid, as encountered in electrical power systems.

b) in contrast to the actual input signal, which is nominally sinusoidal, the phase signal of interest is a periodic ramp, as represented by $\theta_o(t)$, which will have to be synthesized on the basis of the input signal, by the VCO. The term "VCO" stands for "Voltage Controlled Oscillator", which refers to electric circuit realizations of this functionality, whereby the signal frequency can be controlled by an input control voltage $V_c(t)$; in the simplest case, the frequency is determined by a linear relationship:

$$f = K_o \times V_c \qquad \text{e.g. } K_o \text{ in Hz/Volt}$$

In a digital realization, such as with a DSP, it would be more appropriately called "Controlled Frequency Oscillator", or Numerically Controlled Oscillator (NCO), but it is common in the literature to refer to it as VCO. The generated ramp signal $\theta_o(t)$ would be physically represented, such as in a DSP, by a numerical count, produced by a digital counter, and thus representing the phase angle from 0 to 2π, and subsequently used to produce triggering pulse signals to power control devices.

c) Functionally, the estimated phase signal $\theta_o(t)$ is compared to the input signal phase $\theta_i(t)$, by the phase detector, PD, which produces a measure of the phase discrepancy $\theta_e(t) = \theta_i(t) - \theta_o(t)$. There are various possible phase detection mechanisms, with different characteristics, but which can be usefully represented by a gain K_D, valid for some range, e.g. for $| \theta_e | < 30^o$, or more, depending on the technique used. Often, for "small" θ_e, a linearized characterization of K_D usually serves well for initial analysis and design, subject to further tuning to ensure robust operation in the face of actual operational circumstances.

d) Depending on the phase detection mechanism used, the PD output signal may be corrupted by various kinds of noise, either due to distortion of the input sinusoid, as is very common in power electronics applications, or

inevitably resulting from the intrinsic properties of the PD itself. This typically requires some low-pass filtering to attenuate the impact of this noise, and thus ensure an acceptable level of quality of the output phase estimate.

2.1.2 Conventional Single-Phase PLL

Referring to Fig. 2.1, the phase detection mechanism used depends primarily on the form and characteristics of the available input signal and of the desired output signal, [23-25]. For a nominally sinusoidal input signal $u(t) = A\cos(\omega t)$, corrupted by noise and harmonic distortion, a common and simple approach to measuring phase discrepancy between the incoming signal and the VCO synthesized signal, is to produce a fixed amplitude sinusoid $Q(t) = \sin(\omega t-\varphi)$, nominally lagging u(t) by 90^o plus a discrepancy $\varphi(t)$, and to multiply it by u(t), as in Fig. 2.2.

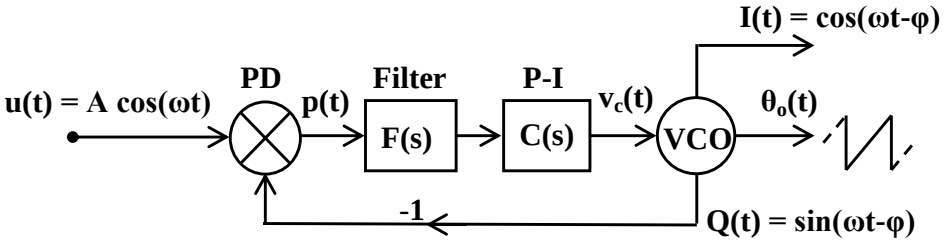


Fig. 2.2 – Conventional single-phase PLL

This results in:

$$p(t) = A\cos(\omega t) \times \sin(\theta_o(t)) , \qquad \theta_o(t) = \omega t-\varphi$$
$$= A/2 \times (\sin(2\omega t-\varphi) - \sin(\varphi))$$
$$= A/2 \times (-\sin(\varphi) + \sin(2\omega t-\varphi)) \qquad (2.1)$$

The above signal consists of two components:

a) $\varepsilon(\varphi) = A/2 \times \sin(\varphi)$: a useful measure of the phase discrepancy φ;

b) $A/2 \times \sin(2\omega t-\varphi)$: an oscillation at twice the incoming signal frequency.

The objective of the feedback loop is thus to correct the estimated angle $\theta_o(t)$ such as to drive the discrepancy $\varphi(t)$ to zero. To achieve this objective, $\theta_o(t)$ must

therefore be continually adjusted, according to the error signal $\varepsilon(t)$, by applying a suitable corrective control signal $v_c(t)$ to the VCO. Because this error signal is effectively corrupted by the double frequency component, some filtering is required, represented by F(s). Furthermore, to ensure zero steady state average error $\varepsilon(t)$ with slowly varying frequency of the input signal, a proportional-plus-integral (P-I) controller is commonly used, represented by C(s) in Fig. 2.2. Because C(s) incorporates an integrating operation, which has an intrinsically low-pass frequency response characteristic, and will thus attenuate the effect of this oscillation, a given application may not necessarily require an additional F(s).

2.1.3 PLL Closed Loop Transfer Function

Based on the previous qualitative description of the PLL operation, a more exact mathematical description of it's dynamic behaviour is now derived, which will serve as a useful frame of reference for subsequent variations and enhancements. It should be noted that the phase detector output, $\varepsilon(\varphi) = A/2 \times \sin(\varphi)$, is a non-linear function of the phase discrepancy $\varphi = \theta_i - \theta_o$; however, for an initial assessment of the overall behaviour, and as is typically justified for relatively small φ, such as $|\varphi| < 30^o$, the approximation is made that $\sin(\varphi) \approx \varphi$, which significantly simplifies the subsequent analysis. Thus, referring to Fig. 2.1 and 2.2, the principal signals and operations may be summarized as follows:

a) $u(t)$: actual input signal, nominally sinusoidal
b) $\theta_i(t)$: phase of input signal
c) $\theta_o(t)$: output phase signal
d) $I(t) : \cos(\theta_o(t))$
e) $Q(t) : \sin(\theta_o(t))$
f) $\varepsilon(t) = A/2 \times \sin(\theta_i(t) - \theta_o(t)) = A/2 \times \sin(\varphi(t)) \approx A/2 \times \varphi(t)$
   Thus, the phase detector can be characterized by a gain of A/2.
g) F(s) : in the simplest case, it can be set to a constant, e.g. F(s) = 2, to simplify subsequent derivations;
h) $C(s) = K_p + K_i/s$: a standard P-I controller

i) VCO: $\theta_o(t) = \int \omega_o(t)\,dt = 2\pi\int K_o \times V_c(t)\,dt$, K_o in Hz/input unit

Thus, the Laplace s-domain transfer function for the VCO can be characterized as: $\dfrac{\Theta_O(s)}{V_C(s)} = \dfrac{2\pi K_O}{s}$

With reference to an input signal at constant frequency ω, the dynamic performance of the overall negative feedback loop is typically characterized in terms of the phase deviation at the output, $\Delta\theta_o$, in response to phase deviations at the input, $\Delta\theta_i$, such as occurs with phase jumps due to sudden load changes. Thus, the overall transfer function from $\Delta\theta_i(t)$ to $\Delta\theta_o(t)$ can be expressed as:

$$\frac{\Delta\Theta_o(s)}{\Delta\Theta_i(s)} = \frac{\dfrac{A}{2}\times 2\times\left(K_p + \dfrac{K_i s}{s}\right)\times\dfrac{2\pi K_O}{s}}{1+\dfrac{A}{2}\times 2\times\left(K_p + \dfrac{K_i s}{s}\right)\times\dfrac{2\pi K_O}{s}}$$

$$= \frac{2\pi A K_O \times\ K_p s + K_i}{s^2 + 2\pi A K_O \times\ K_p s + K_i}$$

$$= \frac{2\pi A K_O K_p s + 2\pi A K_O K_i}{s^2 + 2\pi A K_O K_p s + 2\pi A K_O K_i} \qquad (2.2)$$

This second order transfer function can be related to the standard form, in terms of the natural un-damped frequency ω_n and the damping factor ζ:

$$= \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{2\zeta\ s/\omega_n + 1}{s^2 + 2\zeta\ s/\omega_n + 1} \qquad (2.3)$$

$$\omega_n = \sqrt{2\pi A K_O K_i} \qquad \zeta = \frac{K_p}{2}\sqrt{\frac{2\pi A K_O}{K_i}} \qquad (2.4)$$

As should be expected, the DC gain, at s = 0, is unity, meaning that for slow variations in the input signal frequency and phase, the output phase exactly tracks the input phase. The dynamic performance can also be seen to depend on the input signal amplitude, and it is possible to estimate this amplitude in order to introduce a normalizing division operation into the feedback loop to reduce the PLL operation sensitivity to the signal amplitude [39-42].

2.1.4 Frequency Normalized P-I Control

The transfer function previously obtained reveals an inter-dependence between design parameters K_p and K_i, such that, for example, if it is desired to increase the closed loop bandwidth to increase the speed of response, by increasing K_i, then it is also necessary to adjust K_p accordingly, to maintain the same damping factor. In order to separate these two factors, in terms of their influence on the closed loop dynamics, and thus make their adjustments more convenient, the following approach is adopted:

a) the VCO constant is fixed to K_o = 1 Hz per input unit, or 1 Hz/iu = 2π rad/sec./iu

b) the P-I controller structure is slightly modified, resulting in the overall closed loop structure in Fig. 2.3, where f_o is a constant parameter, and sets the nominal operating frequency, and is normally set equal to the nominal anticipated input signal frequency, in Hz;

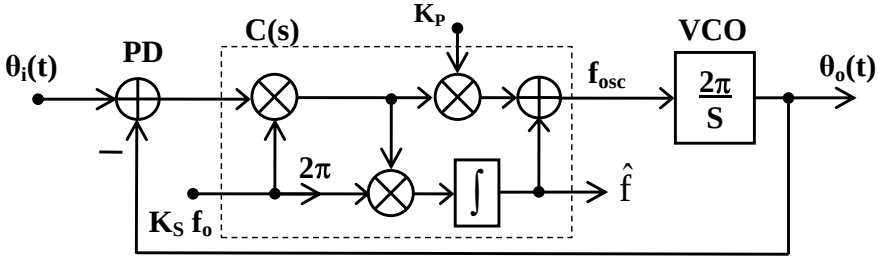


Fig. 2.3 – PLL with modified P-I controller

c) the P-I controller provides two outputs:

   a. f_{osc}: constitutes the actuating, or control, signal to the VCO;

   b. $\hat{f}$: provides a filtered version of f_{osc}, a smoother estimate of the input signal frequency, and is not subject to the higher frequency components in the f_{osc} signal due to the proportional action in C(s).

This results in the following transfer function from θ_i to θ_o, with $\omega_o = 2\pi f_o$:

$$\frac{\Delta\Theta_o(s)}{\Delta\Theta_i(s)} = \frac{\frac{K_S\omega_O}{s} \times \left(K_p s + K_S\omega_O\right) \times \frac{1}{s}}{1 + \frac{K_S\omega_O}{s} \times \left(K_p s + K_S\omega_O\right) \times \frac{1}{s}} = \frac{K_p K_S\omega_O s + K_S^2\omega_O^2}{s^2 + K_p K_S\omega_O s + K_S^2\omega_O^2} \qquad (2.5)$$

In terms of the standard form, ω_n and ζ are then simply determined by:

$$\omega_n = K_s\omega_o \quad \text{and} \quad \zeta = K_p/2 \tag{2.6}$$

The "speed" parameter K_s and the damping parameter K_p can be seen to independently determine the closed loop bandwidth, which is closely related to ω_n, and the damping factor ζ, respectively, which will significantly simplify their adjustments according to specific application requirements. Thus:

a) K_p will be typically set to approximately 1.7, corresponding to a damping factor $\zeta = 0.85$, for a damped and nominally optimal frequency step response, with no observable overshoot, as will be demonstrated later;

b) K_s can initially be set to a value in the vicinity of 1, corresponding to a closed loop bandwidth approximately equal to ω_o. However, depending on noise and distortion, it can be comfortably reduced to 0.1 or less, which will reduce the closed loop bandwidth and slow down the response time, while maintaining a constantly damped response.

This re-structuring and resulting independence between K_s and K_p will also help make more meaningful comparisons between various PLL algorithms.

2.1.5 Frequency Step Response

A common characterization of PLL operation is in terms of it's response to changes in input signal frequency, [2, 9, 17, 19-23], and it's ability to track such variations, as specifically evaluated with step frequency changes. Referring to the P-I controller in Fig. 2.3, and the estimated frequency output $\hat{f}$, and considering that the input signal phase $\theta_i(t)$ is related to it's frequency as: $\theta_i(t) = \int\omega_i(t)dt$, the transfer function from input frequency f_i, to $\hat{f}$, both in Hz, can be determined to be:

$$\frac{\Delta\hat{f}(s)}{\Delta f_i(s)} = \frac{K_S^2\omega_O^2}{s^2 + K_pK_S\omega_O s + K_S^2\omega_O^2} = \frac{\omega_n^2}{s^2 + K_p\omega_n s + \omega_n^2} \tag{2.7}$$

which also shows a unity gain at low frequency; that is, for slow changes in input signal frequency, the PLL can accurately track such variations, and can thus

produce an accurate estimate $\hat{f}$. Such a frequency step response is demonstrated in Fig. 2.4

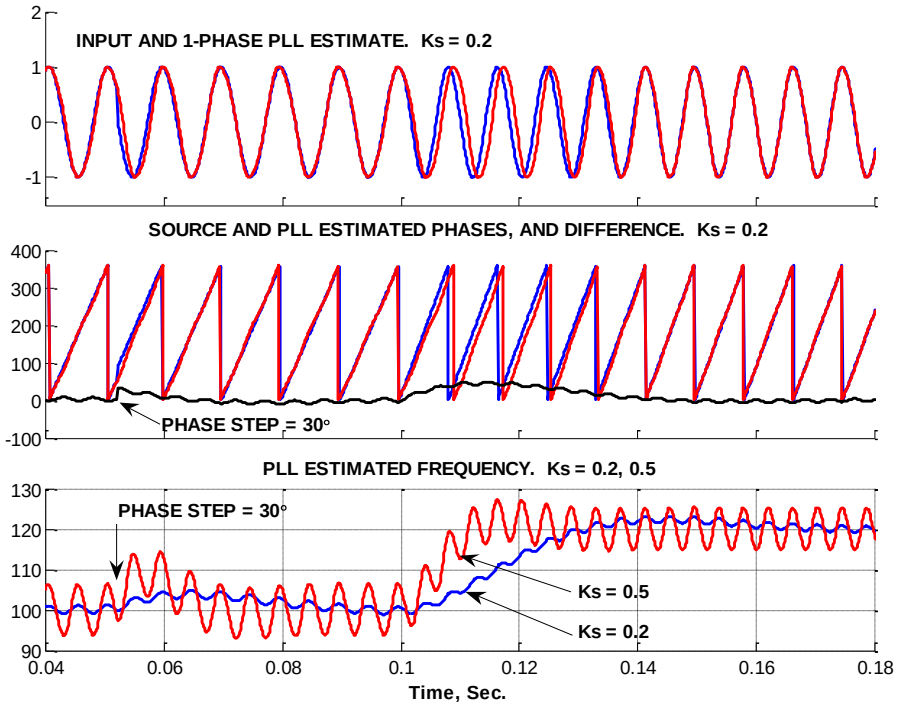


Fig. 2.4 – Conventional single-phase PLL performance

Fig. 2.4 reveals several important points:

a) the top trace displays a unit amplitude input signal, with a 30^o angle step at t = 0.05 sec., and a 20 Hz frequency step at t = 0.1 sec.;

b) the middle trace displays the phase angle at the source and estimated by the PLL, as well as their difference, which is seen to converge to zero after both step changes; a value of $K_s = 0.2$ is used;

c) the bottom trace displays the estimated frequency $\hat{f}$, for K_s 0.2 and 0.5;

d) a very salient feature of the bottom trace is the double-frequency ripple in the $\hat{f}$ signal, which is lower with a lower value of K_s;

e) if not for the ripple, the average response dynamics reflects the idealized, linearized, characterization of the transfer function in (2.7), whereas the presence of ripple reflects the actual fact of the non-ideal phase detecting multiplier operation. More importantly, for lower values of K_s and

consequently slower transient response, the average response validates the idealization of the phase detecting multiplier as linearly measuring the phase discrepancy, and presents a useful frame of reference for further analysis and investigation into improving the performance.

Point (d) reflects the familiar fact of feedback control systems, that to reduce the sensitivity to various disturbances, such as double-frequency ripple, requires that the overall closed-loop bandwidth be reduced, such as by lowering the gain K_s, thus also reducing the system's speed of response, a typical design trade-off between these two important factors of response speed and disturbance rejection.

2.2 PARK TRANSFORMATION

The double-frequency ripple has motivated much research to eliminate, or at least attenuate, it's impact on the PLL performance, [44-50]. The Park transformation is essentially an extension of the single-multiplier phase detector, and is commonly used for multi-phase sinusoidal signals; for example, for two- and three-phase signals, as encountered in power systems. For two-phase signals, given two-dimensional (vector) signals $\mathbf{v}_1(t)$ and $\mathbf{v}_2(t)$, with phase angles $\theta(t)$ and $\varphi(t)$, respectively:

$$\mathbf{v}_1(t) = V_1\begin{pmatrix}\cos(\theta(t))\\ \sin(\theta(t))\end{pmatrix} = V_1 e^{j\theta(t)} = V_1(\cos(\theta(t)) + j\sin(\theta(t))) \tag{2.8a}$$

$$\mathbf{v}_2(t) = V_2\begin{pmatrix}\cos(\varphi(t))\\ \sin(\varphi(t))\end{pmatrix} = V_2 e^{j\varphi(t)} = V_2(\cos(\varphi(t)) + j\sin(\varphi(t))) \tag{2.8b}$$

the Park transformation simply consists of a product of two complex numbers (with reference to time variable t omitted, for simplicity):

$$\begin{aligned}\mathbf{v}_1 \times \mathbf{v}_2 &= V_1\mathbf{e}^{j\theta} \times V_2\mathbf{e}^{j\varphi} =: V_1\begin{pmatrix}\cos(\theta)\\ \sin(\theta)\end{pmatrix} \times V_2\begin{pmatrix}\cos(\varphi)\\ \sin(\varphi)\end{pmatrix}\\ &= V_1 \times V_2 \times \begin{bmatrix}(\cos(\theta)\cos(\varphi) - \sin(\theta)\sin(\varphi))\\ +j(\cos(\theta)\sin(\varphi) + \sin(\theta)\cos(\varphi))\end{bmatrix}\\ &= v_d + jv_q\end{aligned}$$

In view of occasional discrepancies in the literature with regards to the Park transformation's output d and q signals, the above definition is adopted here; d is a measure of the product of sinusoidal components nominally directly in-phase, whereas q is a measure of the angular discrepancy of signals, via the product of signals in quadrature (e.g. $\cos(\theta) \times \sin(\varphi)$). Expressed in vector-matrix form:

$$\begin{aligned}\begin{pmatrix} v_d \\ v_q \end{pmatrix} &= V_1 V_2 \times \begin{pmatrix} \cos(\theta)\cos(\varphi) - \sin(\theta)\sin(\varphi) \\ \cos(\theta)\sin(\varphi) + \sin(\theta)\cos(\varphi) \end{pmatrix} \\ &= V_2 \times \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix} \times V_1 \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} \\ &= V_2 \times \boldsymbol{R}(\varphi) \times \mathbf{v}_1 \end{aligned} \tag{2.9}$$

(2.9) emphasizes that an incoming signal $\mathbf{v}_1$ is operated on with a gain of V_2, and rotated by an angle φ. If, as is usually the case when nominally synchronized with a PLL, $\mathbf{v}_1$ and $\mathbf{v}_2$ are at nearly the same frequency ω, but with $\mathbf{v}_2$ lagging incoming signal $\mathbf{v}_1$ with a phase difference $\delta = \theta - \varphi$, then multiplying $\mathbf{v}_1$ by the complex conjugate of $\mathbf{v}_2$ results in:

$$\begin{aligned}\begin{pmatrix} v_d \\ v_q \end{pmatrix} &= v_1 \times v_2^* = V_1 V_2 \times \begin{pmatrix} \cos(\omega t - \delta) & \sin(\omega t - \delta) \\ -\sin(\omega t - \delta) & \cos(\omega t - \delta) \end{pmatrix} \times \begin{pmatrix} \cos(\omega t) \\ \sin(\omega t) \end{pmatrix} \\ &= V_1 V_2 \times \begin{pmatrix} \cos(\omega t)\cos(\omega t - \delta) + \sin(\omega t)\sin(\omega t - \delta) \\ -\cos(\omega t)\sin(\omega t - \delta) + \sin(\omega t)\cos(\omega t - \delta) \end{pmatrix} \\ &= V_1 V_2 \times \begin{pmatrix} \cos(\delta) \\ \sin(\delta) \end{pmatrix} \\ &= V_2 \times R(-\omega t + \delta) \times v_1 \end{aligned} \tag{2.10}$$

Thus, when nominally synchronized, and with $\mathbf{v}_1$ of unit magnitude, the Park transformation produces a signal $v_q(t)$ which is at a frequency near 0 Hz, and which is a measure of the phase discrepancy δ. Again, when nearly phase synchronized, i.e. with small $\delta \approx 0$, then $\sin(\delta) \approx \delta$ is a useful linearization. The Park transformation also produces the signal $v_d(t) = V_2\cos(\delta)$, which is likewise at a frequency near 0 Hz, and when nearly phase synchronized, $\delta \approx 0$, then $\cos(\delta) \approx 1$, and $v_d(t)$ is a useful measure of the input signal amplitude. Thus, for purposes of phase detection in a PLL, the defining feature of the Park transformation is that

the availability of 2-dimensional quadrature signals, in both the input and the tracking generated signals, makes possible the perfect cancellation of any double-frequency ripple, which potentially allows the complete elimination of any low-pass filtering, except for any possible distortion components resulting from distortion of the input signals, resulting in potentially much faster response.

2.3 CONVENTIONAL TWO-PHASE PLL OPERATION

Considering phase synchronization only, the Park transformation's q output consists of a simple extension to the single-multiplier phase detector, with a second multiplier to cancel out the double-frequency ripple from the first multiplier, which leads to the conventional two-phase PLL, as in Fig. 2.5.

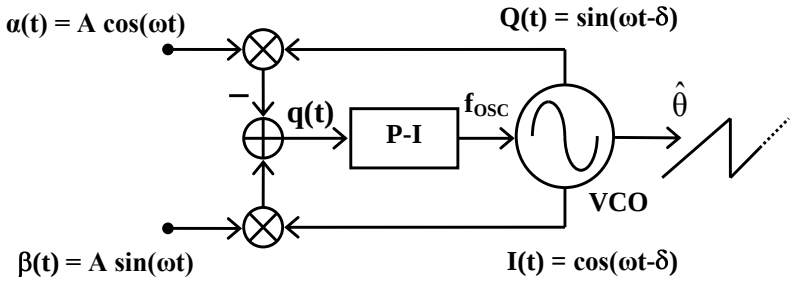


Fig. 2.5 – Conventional two-phase PLL

This results in the same relationships for the input to output phase and frequency transfer functions in (2.5) and (2.7), respectively.

$$\frac{\Delta\Theta_o(s)}{\Delta\Theta_i(s)} = \frac{K_p K_S \omega_O s + K_S^2 \omega_O^2}{s^2 + K_p K_S \omega_O s + K_S^2 \omega_O^2} = \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{2.5}$$

$$\frac{\Delta\hat{f}(s)}{\Delta f_i(s)} = \frac{K_S^2 \omega_O^2}{s^2 + K_p K_S \omega_O s + K_S^2 \omega_O^2} = \frac{\omega_n^2}{s^2 + K_p \omega_n s + \omega_n^2} \tag{2.7}$$

The Park transformation's d output provides a measure of the input signal's amplitude, and can be used to normalize the q signal, such as to maintain the PLL transient dynamics overshoot independent of the signal amplitude, as suggested in Fig. 2.6, [40-45].

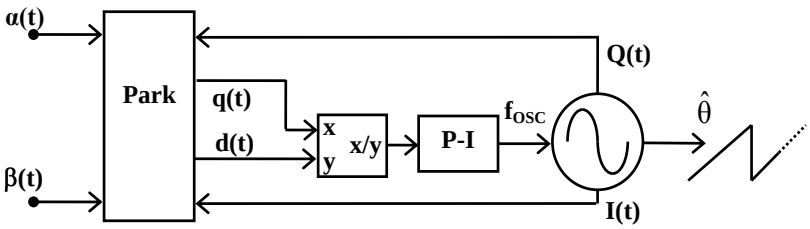


Fig. 2.6 – Amplitude normalized conventional two-phase PLL

The improvement in performance and potentially faster response of the conventional two-phase PLL is demonstrated in Fig. 2.7.

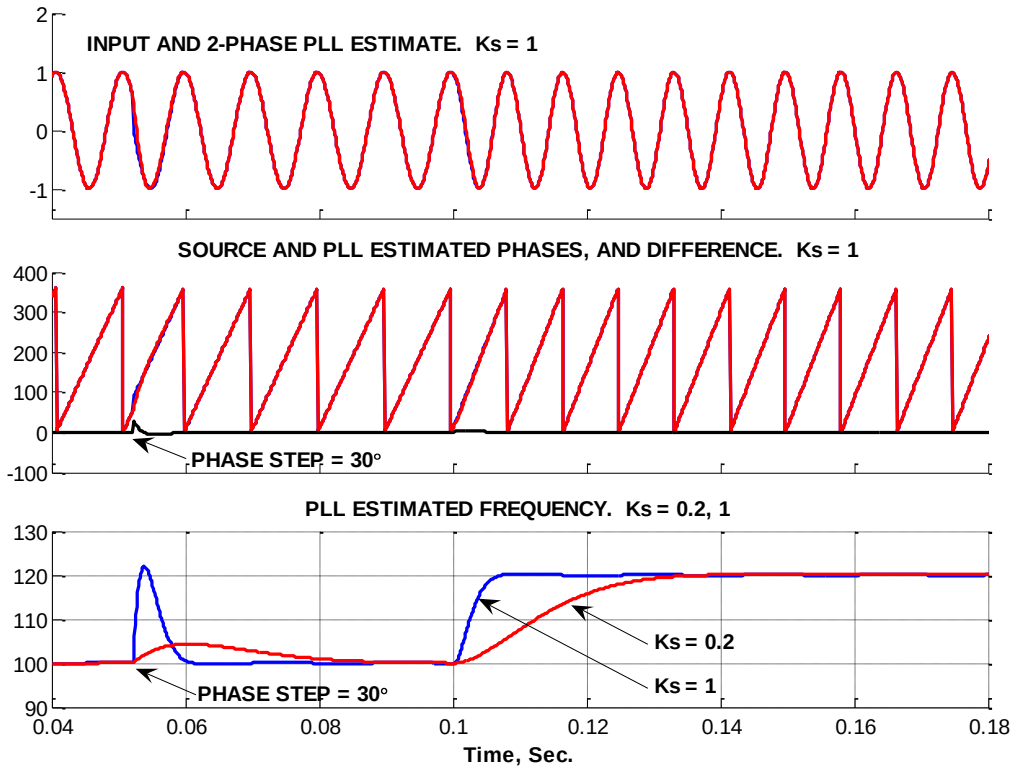


Fig. 2.7 – Conventional two-phase PLL performance

Fig. 2.7 emphasizes the following:

a) the top trace shows the input α signal, with a 30^o phase step at t = 0.05 and a 20 Hz frequency step at t = 0.1 sec., respectively, and the PLL's I output signals, with $K_s = 1$;

b) the middle trace shows the input signal phase angle, and the PLL's estimated phase output signal, and their difference, also with $K_s = 1$;

c) the bottom trace shows the PLL frequency estimate output, with $K_s = 0.2$ and 1, and, most importantly, reveals a complete absence of double-frequency ripple; this makes possible to tune the gain K_s for mush faster response;

d) as well, K_s can be adjusted over a considerably wider range, while maintaining the same relative transient response, i.e. with respect to overshoot in the frequency estimate step response, making it possible to easily tune the PLL's bandwidth and sensitivity to noise and distortion, according to specific application requirements.

2.4 QUADRATURE SIGNAL GENERATION

Given a single-phase input signal, if the benefit of the Park transformation is to be exploited to eliminate the double-frequency ripple, then a quadrature version of the input signal must be somehow synthesized. A number of techniques have been proposed, including:

a) a quarter-cycle delay line, consisting of N/4 memory storage elements, to delay a signal by a quarter-cycle, such as in the delayed signal cancellation (DSC) technique [35, 45], where N is an integer number of samples per cycle of the input signal. Although very simple to implement and with guaranteed stability, this approach assumes that the sampling frequency $f_s = N \times f_o$ exactly, to obtain the desired 90^o phase shift from a N/4 delay. Depending on how much the signal frequency may vary and on acceptable error, it may be required to frequency lock f_s to $N \times f_o$, which presents an additional challenge to the overall synchronization task. Another disadvantage of this approach is that, because of the constant amplitude all-pass frequency response characteristic of a delay line, all distortion components are passed through un-attenuated.

b) a first or second order all-pass phase-shifting filter, [10-13], of the form:

a. $H_{AP\text{-}1}(s) = \dfrac{1 - s/\omega_O}{1 + s/\omega_O}$: produces a unity gain, 90^o phase shift at ω_o

b. $H_{AP\text{-}2}(s) = \dfrac{s^2 - 2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = 1\angle -2\,\text{atan}\left(\dfrac{2\zeta}{\frac{\omega_n}{\omega} - \frac{\omega}{\omega_n}}\right)$

The desired phase shift and specified frequency are thus adjustable with ζ and $\omega_{n.}$

The all-pass filter likewise does not attenuate any distortion component, and requires adaptivity to track variations in input signal frequency to maintain the desired 90^o phase shift over the anticipated range of frequencies.

c) a second order low-pass filter, [24-26]:

$$H_{LPF}(s) = \frac{2\zeta\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad = 1\angle 90^\circ \text{ at } s = j\omega_n$$

This also may require real time tunability to maintain the desired 90^o phase shift as the signal frequency varies.

d) a Hilbert transform FIR filter, to produce an approximate 90^o phase shift over some specified range of frequencies around the nominal input signal frequency ω_o, [46]. Because this produces a constant phase shift over some frequency range, this does not require any frequency adaptivity. However, this also has a constant amplitude frequency response, which therefore also does not provide any attenuation of distortion components.

e) so-called Adaptive Signal Cancellation, [52], which adaptively estimates the amplitudes of the quadrature components of the signal, and then performs an elementary complex rotation by $\pi/2$, similar to the Park transformation.

f) a second order sinusoidal resonator-based band-pass/low-pass filter, also referred to as sinusoidal integrator, [53-56], or second order generalized integrator (SOGI), [24-26]. This approach is adopted for several reasons:

   a. it provides both an in-phase band-pass and a quadrature-phase low-pass filtered output signals, both with unity gain at ω_o:

$$H_{BPF}(s) = \frac{2\zeta\omega_n s}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad = 1\angle 0^\circ \text{ at } s = j\omega_n$$

$$H_{LPF}(s) = \frac{2\zeta\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad = 1\angle -90^\circ \text{ at } s = j\omega_n$$

b. useful filtering operation is intrinsic to the operation, and thus serves to attenuate distortion components;
c. the filter bandwidth and rate of attenuation of distortion components can easily be adjusted with the damping factor ζ;
d. it can be operated at fixed sampling frequency, and it's center frequency can easily be adjusted for the purpose of tracking input signal variations, making it very suitable to implement a so-called (frequency) adaptive notch filter (ANF);
e. the simultaneous availability of equal amplitude, in-phase and quadrature-phase, filtered, outputs, can serve to individually filter and phase-shift α and β signals, and subsequently extract positive and negative sequence components of an unbalanced input vector signal;
f. for all it's benefits, it is the most computationally economical structure, in terms of overall number of memory storage elements and arithmetic operations required, making it well suited for real time implementation on DSP or FPGA;
g. this structure can be conveniently realized in a synchronous reference frame, again making it very well suited for discrete implementation on a DSP or FPGA, and results also in improved performance at higher center frequencies and narrower bandwidths, as will be discussed later.

2.5 ADAPTIVE NOTCH FILTER FOR FREQUENCY TRACKING

Frequency variations of the input signal make it necessary to provide for frequency adaptivity of filtering units, such as the resonator-based quadrature signal generator introduced in the previous section. This section describes the operation of this device, from the perspective of augmenting it with a simple

frequency-tracking mechanism, together commonly referred to as an adaptive notch filter (ANF), [56-64] (due to the notch filtering at the core of the frequency-tracking functionality), or also frequency-locked loop (FLL), [24]. Such a device has been proposed for phase estimation and synchronization purposes, [2], as well as for extracting positive and negative sequences of an unbalanced signal, [62-65], for a few significant reasons:

a) the basic ANF structure is a third order, non-linear, dynamical system, consisting of a second order resonator, and a simple integrating mechanism. However, based on averaging principles, [56-57, 64, 67-69], it's overall operation, for the intended purposes of frequency tracking, can be reduced to a first order system, conveniently simplifying it's analysis and design;
b) owing to the above, it can easily serve as a front end pre-filter followed by a simple two-phase PLL, while the overall dynamics can be treated as just a first order sub-system followed by a second order PLL sub-system, with no further dynamic interaction between them, and thus again simplifying the overall analysis and design;
c) the afore-mentioned frequency-tracking mechanism in the ANF (a simple integrator) can easily be replaced by a two-phase PLL, [70], to provide simultaneous frequency-tracking and phase estimation, although this does then become a fourth order, non-linear, dynamical system, requiring more careful analysis and design.

2.5.1 The Second Order Quadrature Signal Generator

At the core of the frequency-adaptive notch filter is a two-integrator resonator, also referred to as a generalized integrator, [53-55], or second order generalized integrator (SOGI), [24-26, 71], shown in Fig. 2.8. With zero input and non-zero initial conditions, this produces a constant amplitude oscillation at frequency ω_o. With a sinusoidal input v(t) at the same frequency ω_o, the outputs x(t) and y(t) consist of sinusoids, 90^o apart, with linearly increasing amplitude, reflecting the integrating behaviour from which it gets it's name; for example, with $v(t) = \cos(\omega_o t)$:

$$x(t) = \omega_o/2 \times \cos(\omega_o t) \times t \; + \; \omega_o/2 \times \sin(\omega_o t)$$

The constant amplitude component will later be seen to vanish, once this structure is embedded in a feedback loop, such that only the first component above will be relevant.

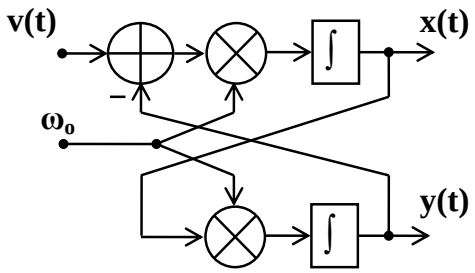


Fig. 2.8 – Second order generalized integrator

The transfer function from input v(t) to output x(t) can be expressed as in (2.8).

$$T_{XV}(s) = \frac{X(s)}{V(s)} = \frac{\omega_o / s}{1 + \omega_o^2 / s^2} = \frac{\omega_o s}{s^2 + \omega_o^2} \tag{2.8}$$

The above resonator can be embedded within a feedback loop, as in Fig. 2.9, which results in three useful functions at the indicated outputs:

a) I: band-pass filtered, in-phase with input, and with unity gain at ω_o:

$$T_{IU}(s) = \frac{I(s)}{U(s)} = \frac{K\omega_o s}{s^2 + K\omega_o s + \omega_o^2} = \frac{2\zeta\omega_n s}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{2.9}$$

b) Q: low-pass filtered, in quadrature and with unity gain at ω_o;

$$T_{QU}(s) = \frac{Q(s)}{U(s)} = \frac{K\omega_o^2}{s^2 + K\omega_o s + \omega_o^2} = \frac{2\zeta\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{2.10}$$

c) ε: notch-filtered, zero output response at ω_o:

$$T_{EU}(s) = \frac{E(s)}{U(s)} = \frac{s^2 + \omega_o^2}{s^2 + K\omega_o s + \omega_o^2} = \frac{s^2 + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{2.11-a}$$

$$T_{EU}(s = j\omega \approx j\omega_o) \approx \frac{\omega_n^2 - \omega^2}{j2\zeta\omega\omega_n} = \frac{(\omega_n - \omega)(\omega_n + \omega)}{j2\zeta\omega_n^2}$$

$$\approx \frac{(\omega_n - \omega)}{j\zeta\omega_n} = \frac{2}{jK}\frac{\Delta\omega}{\omega_n} \tag{2.11-b}$$

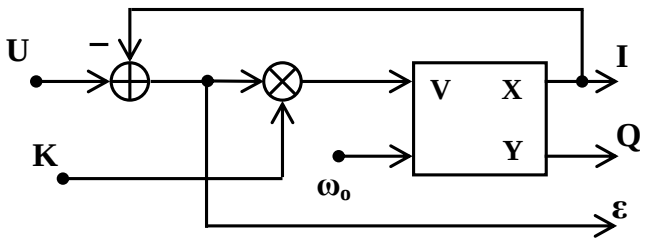


Fig. 2.9 – SOGI-based band-pass/low-pass filter

The above transfer functions are exemplified in Fig. 2.10, with K = 0.5, at a center frequency of 60 Hz. The BPF and LPF responses can be seen to intersect at ω_o with unity gain.

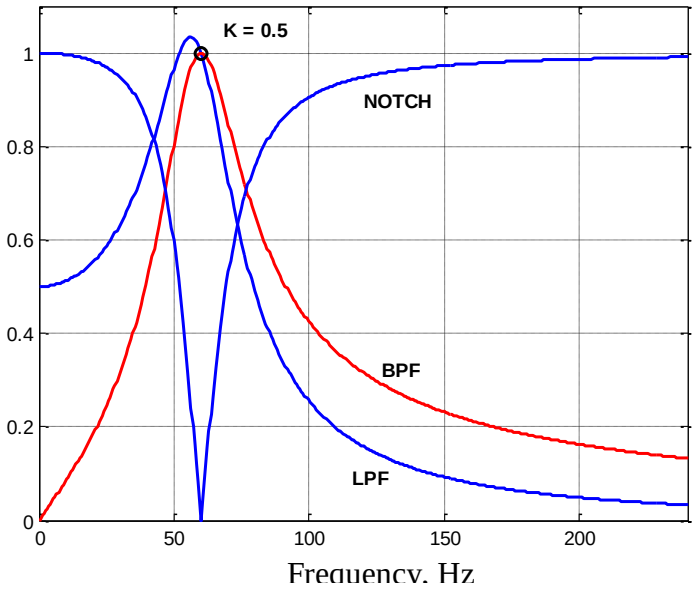


Fig. 2.10 – SOGI-based BPF, LPF, and notch filter frequency responses

2.5.2 Frequency-Adaptive Notch Filter

Close examination of the low-pass and notch filter transfer functions (2.10) and (2.11-a) reveals a very useful fact. The relationship between the two respective output signals is that they are either in phase for $\omega < \omega_o$, or exactly 180° out of phase for $\omega > \omega_o$. Indeed, the ratio of their transfer functions is purely real:

$$T_{EQ}(s = j\omega) = \frac{T_{EU}}{T_{QU}} = \frac{s^2 + \omega_o^2}{K\omega_o^2} = \frac{\omega_o^2 - \omega^2}{K\omega_o^2} = \frac{(\omega_o - \omega)(\omega_o + \omega)}{K\omega_o^2}$$

$$T_{EQ}(\omega \approx \omega_o) \approx \frac{2(\omega_o - \omega)}{K\omega_o} = \frac{\omega_o - \omega}{\zeta\omega_o} = \frac{\Delta\omega_i}{\zeta\omega_o} \tag{2.12}$$

Thus, T_{EQ} gives a useful indication of the direction of ω relative to ω_o, and suggests that we can form a correlation product $Q \times \varepsilon = Q(j\omega) \times \varepsilon(j\omega)$, [66], which then consists of a sinusoid squared with an average non-zero value $A^2 \times T_{QU}(j\omega) \times T_{EU}(j\omega)/2$, where A is the input signal amplitude, and this average signal is either positive or negative according to whether ω is less than, or greater than, ω_o. This product can then be used as an error signal to adjust the filter center frequency to converge toward the input signal frequency. Such an adaptive mechanism can easily be realized with an integrator, [24, 58], as in Fig. 2.11, whose output consists of an integration of this error signal:

$$\Delta\hat{\omega}(t) = K_A \int Q(t) \times \varepsilon(t) dt \tag{2.13}$$

The parameter K_a can be adjusted to set the rate of convergence of this frequency adaptation process, and speed of response to changes in input signal frequency.

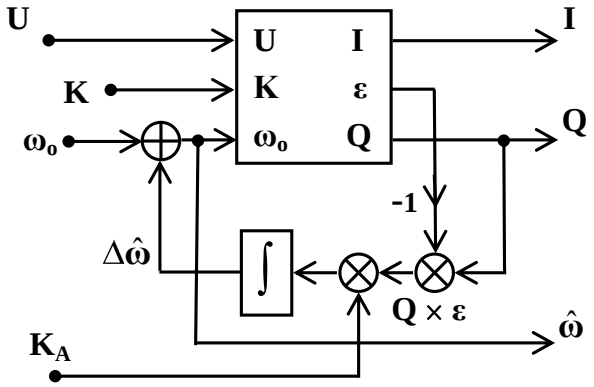


Fig. 2.11 – SOGI-based frequency-adaptive notch filter

Fig. 2.12 and 2.13 demonstrate two examples in response to an input signal frequency step from 100 to 110 Hz, one with slow convergence, with K = 0.5 and K_A = 0.05, and one with fast convergence, with K = 1.5 and K_A = 0.5. The case of slow convergence serves to clearly show the averaging principle in operation, as well as to reveal the approximate character of this averaging mechanism, made evident in the ripple in the estimated frequency signal.

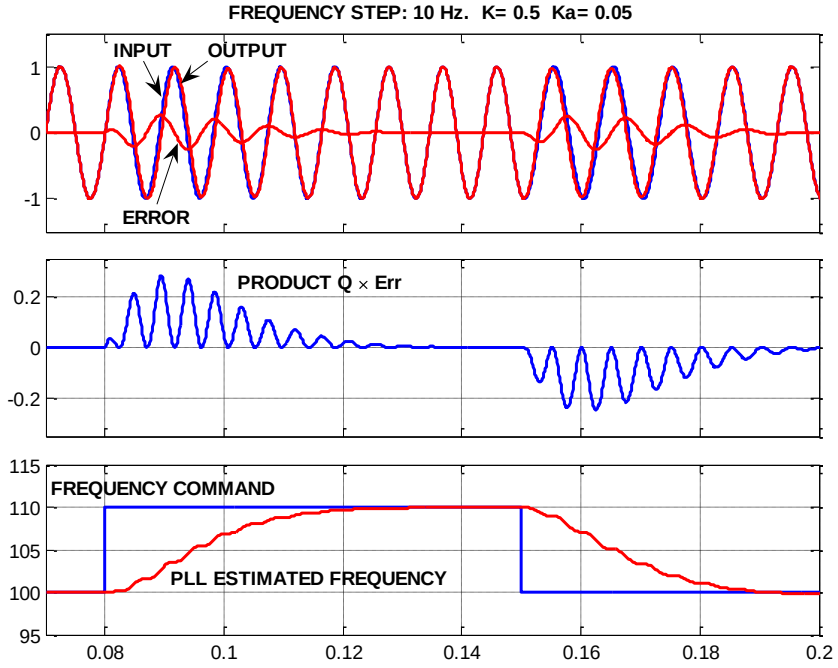


Figure 2.12 – ANF frequency step response, slow convergence

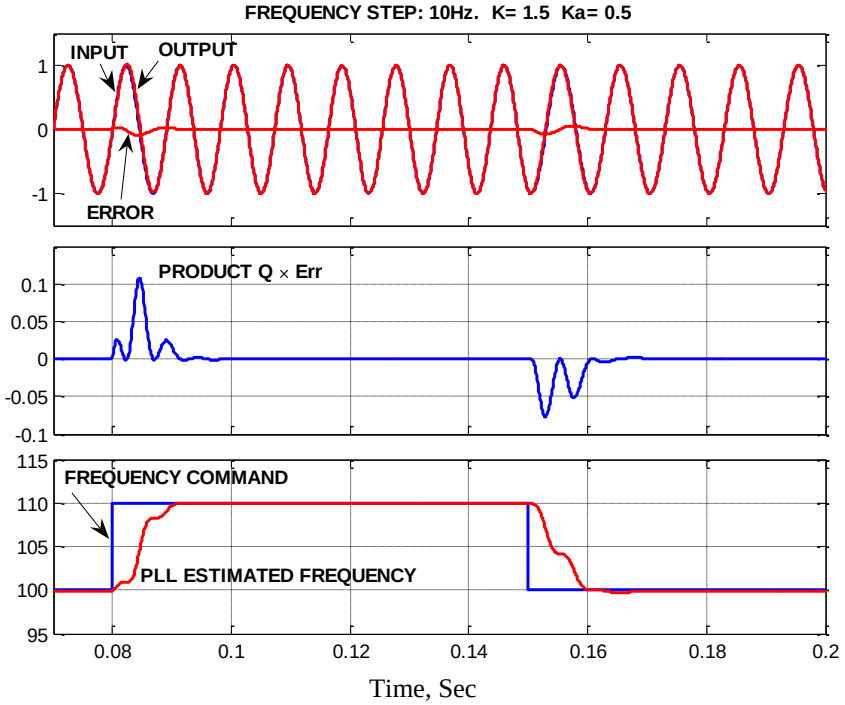


Figure 2.13 – ANF frequency step response, fast convergence

The case of faster convergence shows that during the transient response, the product signal Qε does very slightly change polarity, and there is significant ripple in the estimated frequency signal, but on average Qε is clearly of consistent polarity, and therefore useful for the intended purpose of frequency adaptivity. As a consequence, this averaging mechanism can be seen to simplify the overall dynamics, such that for lower values of K_A, and for a signal frequency close to the filter center frequency, the system can be usefully described by a first order system, as represented in Fig. 2.14, with a time constant set by the adaptation gain K_A.

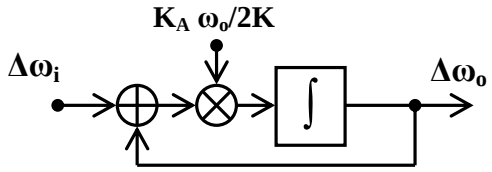


Fig. 2.14 – First order equivalent of adaptive notch filter

2.6 THREE-PHASE SIGNALS AND CLARKE TRANSFORMATION

Whereas power is most economically delivered over a three-wire system, by three sinusoidal voltages spaced 120^o apart, the signal processing involved in many applications can be achieved more efficiently after first transforming the three-phase signal vector, say $\mathbf{v}_{abc}(t)$, using a Clarke transformation, into a two-phase vector $\mathbf{v}_{\alpha\beta}(t)$ in the so-called αβ fixed (also referred to as stationary) reference frame (FRF), where the components $v_\alpha(t)$ and $v_\beta(t)$ are orthogonal to each other:

$$v_{\alpha\beta}(t) = \begin{pmatrix} v_\alpha(t) \\ v_\beta(t) \end{pmatrix} = \frac{2}{3}\begin{pmatrix} 1 & \cos(\gamma) & \cos(2\gamma) \\ 0 & \sin(\gamma) & -\sin(2\gamma) \end{pmatrix} \times \begin{pmatrix} v_a(t) \\ v_b(t) \\ v_c(t) \end{pmatrix}, \quad \gamma = 2\pi/3 \quad (2.14)$$

$$v_{\alpha\beta}(t) := T_{Clark} \times v_{abc}(t)$$

Thus, with $\mathbf{v}_{abc}(t)$ defined as:

$$\mathbf{v}_{abc}(t) = \begin{pmatrix} v_a(t) \\ v_b(t) \\ v_c(t) \end{pmatrix} = V \begin{pmatrix} \cos(\omega t) \\ \cos(\omega t - 2\pi/3) \\ \cos(\omega t - 4\pi/3) \end{pmatrix} \tag{2.15}$$

with peak magnitude V, the Clarke transformation produces:

$$\mathbf{v}_{\alpha\beta}(t) = \begin{pmatrix} v_\alpha(t) \\ v_\beta(t) \end{pmatrix} = V \begin{pmatrix} \cos(\omega t) \\ \sin(\omega t) \end{pmatrix} \tag{2.16}$$

It is noteworthy that the Clarke transformation, as described above, aims to produce signals $\mathbf{v}_{\alpha\beta}(t)$ with $v_\alpha(t) = v_a(t)$, i.e. aligned with $v_a(t)$, and with the same magnitude V as $\mathbf{v}_{abc}(t)$. Thus, the required transformation is not unique, and depending on other factors, for example trying to reduce the number of sensors at the three-phase system, the Clarke transformation could be alternately accomplished. For example, for a three-wire system, $v_a(t) + v_b(t) + v_c(t) = 0$, such that only two signals suffice to obtain the $\mathbf{v}_{\alpha\beta}(t)$ signals, such as b and c, as follows:

$$\mathbf{v}_{\alpha\beta}(t) = \begin{pmatrix} v_\alpha(t) \\ v_\beta(t) \end{pmatrix} = \begin{pmatrix} -1 & -1 \\ 1/\sqrt{3} & -1/\sqrt{3} \end{pmatrix} \times \begin{pmatrix} v_b(t) \\ v_c(t) \end{pmatrix} \tag{2.17}$$

2.7 POSITIVE AND NEGATIVE SEQUENCES, AND DE-COMPOSITION

With reference to the Park transformation, the phase discrepancy between two quadrature signals, measured by the q output, is free of any 2ω component, but only when the 2ω components of the two products cancel out exactly. This exact cancellation, however, depends on both components of the input quadrature signals to have the same magnitude and to be exactly 90^o apart, i.e. that they be balanced. However, since the input quadrature signal is obtained from the three-phase signal via a Clarke transformation, this two-phase balance depends on the three-phase balance of $\mathbf{v}_{abc}$. Various line and load conditions can adversely affect this three-phase balance, and it is then necessary to characterize such three-phase unbalanced conditions to properly compensate for them.

2.7.1 Three-Phase Symmetrical Components

Three-phase voltage (or current) signals, although nominally balanced, can suffer unbalanced conditions, due to asymmetrical loading or line faults, resulting in sub-optimal overall operation and power losses. To deal with such conditions, it is useful to represent a three-phase signal in terms of three symmetrical, or balanced, components, also referred to as positive, negative, and zero sequences, as in (2.18). Superscripts p, n, and z denote positive, negative, and zero, respectively, and the corresponding symmetrical components are defined as in (2.19).

$$\begin{pmatrix} v_a(t) \\ v_b(t) \\ v_c(t) \end{pmatrix} = \begin{pmatrix} 1 \\ \mathbf{e}^{-j2\pi/3} \\ \mathbf{e}^{+j2\pi/3} \end{pmatrix} v_a^p(t) + \begin{pmatrix} 1 \\ \mathbf{e}^{+j2\pi/3} \\ \mathbf{e}^{-j2\pi/3} \end{pmatrix} v_a^n(t) + \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} v_a^z(t)$$

$$\mathbf{v}_{abc}(t) = \begin{pmatrix} v_a(t) \\ v_b(t) \\ v_c(t) \end{pmatrix} = v_{abc}^p(t) + v_{abc}^n(t) + v_{abc}^z(t)$$

$$= \begin{pmatrix} v_a^p(t) \\ v_b^p(t) \\ v_c^p(t) \end{pmatrix} + \begin{pmatrix} v_a^n(t) \\ v_b^n(t) \\ v_c^n(t) \end{pmatrix} + \begin{pmatrix} v_a^z(t) \\ v_b^z(t) \\ v_c^z(t) \end{pmatrix} \tag{2.18}$$

$$= \begin{pmatrix} 1 & 1 & 1 \\ \mathbf{e}^{-j2\pi/3} & \mathbf{e}^{+j2\pi/3} & 1 \\ \mathbf{e}^{+j2\pi/3} & \mathbf{e}^{-j2\pi/3} & 1 \end{pmatrix} \times \begin{pmatrix} v_a^p(t) \\ v_a^n(t) \\ v_a^z(t) \end{pmatrix} \tag{2.19}$$

Thus, each symmetrical three-phase component is referred to the phase a signal:

$$\begin{pmatrix} v_a^p(t) \\ v_a^n(t) \\ v_a^z(t) \end{pmatrix} = \begin{pmatrix} V^p\cos(\omega t) \\ V^n\cos(\omega t - \varphi^n) \\ V^z\cos(\omega t - \varphi^z) \end{pmatrix} \tag{2.20}$$

In other words:

a) in the positive sequence vector, phases b and c lag behind phase a by 120^o and 240^o, respectively;

b) in the negative sequence vector, phases b and c lag behind phase a by 240^o and 120^o, respectively;

c) in the zero sequence vector, all three phases have the same sinusoidal voltage.

The zero sequence component, being common to all three phases, is then the instantaneous average of the three, and requires a common voltage reference point, or return path for the common current, and therefore requires a fourth neutral wire to exist. As a consequence, in the case of a three-wire system, the zero sequence cannot exist, and there are therefore only two independent voltages to speak of; for example, $v_{a-b}(t)$ and $v_{c-b}(t)$, both referred to phase b. Figures 2.15 and 2.16 depict, for example, phasor diagrams for a balanced and an unbalanced conditions. In both cases, v_α can be seen to be aligned with v_a. In Fig. 2.16, the negative sequence v^n_{abc} can be of any arbitrary phase relative to the positive sequence v^p_{abc}.

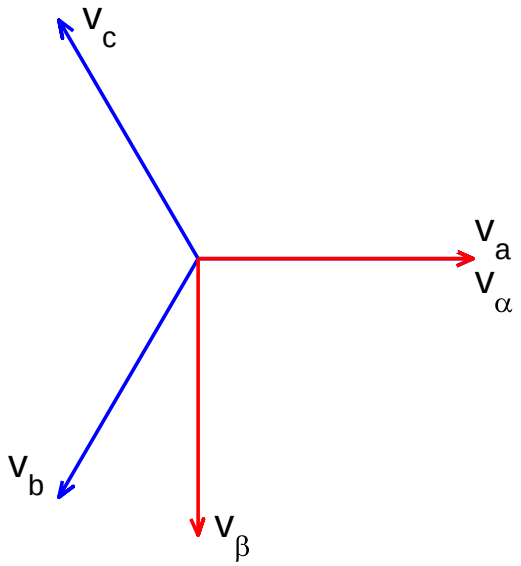


Fig. 2.15 - Three-phase and 2-phase phasors, balanced

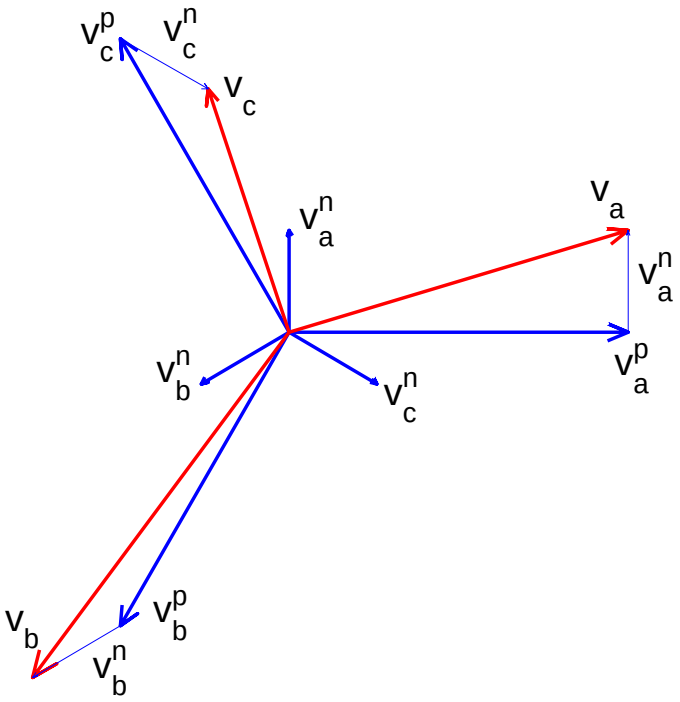


Fig. 2.16 - Three -phase and 2-phase phasors, unbalanced

2.7.2 Positive And Negative Sequence Extraction

The optimal condition for power delivery in a three-wire system consists of balanced three-phase voltage and current, characterized by a symmetrical positive sequence, which defines the reference for power transfer. Thus, when the three-phase voltage is unbalanced, any corrective action, such as by an active power filter, must be referred to, i.e. synchronized to, the positive sequence voltage. Therefore, given an actually measured three-phase voltage $\mathbf{v}_{abc}(t)$, it becomes necessary to estimate from it the positive sequence component, which subsequently defines the proper reference for all voltage, current, and phase information signals. One approach consists of instantaneous symmetrical component calculation, [21, 72], which requires phase shifting operations on each of the three individual phase signals to extract the positive, and possibly also the negative, sequence voltage vectors, depending on the application.

An alternative approach, not too different from the above in regards to some of the underlying operations, but which also serves the additional purpose of phase estimation and synchronization, is to transform the three-phase voltage $\mathbf{v}_{abc}(t)$ into the αβ reference frame, using a Clark transformation. For an unbalanced condition, this results in a two-phase voltage vector $\mathbf{v}_{\alpha\beta}(t)$, which can similarly be expressed in terms of two components, with positive and negative sequences, respectively. These can also be more conveniently expressed by two vectors rotating in opposite directions:

$$\begin{aligned} v_{\alpha\beta}(t) &= v^{p}_{\alpha\beta}(t) + v^{n}_{\alpha\beta}(t) = V^{p}e^{j\omega t} + V^{n}e^{-j\omega t-\varphi} \\ &= \begin{pmatrix} v^{p}_{\alpha}(t) \\ v^{p}_{\beta}(t) \end{pmatrix} + \begin{pmatrix} v^{n}_{\alpha}(t) \\ v^{n}_{\beta}(t) \end{pmatrix} \\ &= V^{p}\begin{pmatrix} \cos(\omega t) \\ \sin(\omega t) \end{pmatrix} + V^{n}\begin{pmatrix} \cos(\omega t+\varphi) \\ -\sin(\omega t+\varphi) \end{pmatrix} \\ &= V^{p}\left(\begin{pmatrix} \cos(\omega t) \\ \sin(\omega t) \end{pmatrix} + u^{n}\begin{pmatrix} \cos(\omega t+\varphi) \\ -\sin(\omega t+\varphi) \end{pmatrix}\right) \end{aligned} \qquad (2.21)$$

(2.21) emphasizes a few useful aspects:

a) the nominal voltage in terms of the positive sequence peak voltage V^p,
b) the relative degree of unbalance, u^n, which is a complex value,
c) the phase relationships between the α and β components for the positive and negative sequence, namely: v^p_β lags v^p_α by 90^o, whereas v^n_β leads v^n_α by 90^o

The last point is significant, as it defines a useful perspective for the subsequent signal processing required to estimate, or extract, the positive and negative sequence signals from the unbalanced signal $v_{\alpha\beta}(t)$. Consider, for example, the unbalanced phasor diagram in Fig. 2.17:

a) unbalanced three-phase phasors v_a, v_b, and v_c
b) two-phase v_α and v_β, obtained via a Clark transformation; note that $v_\alpha = v_a$

c) there is a useful symmetry in the effect of v_α^n on v_α^p, as opposed to the effect of v_β^n on v_β^p.

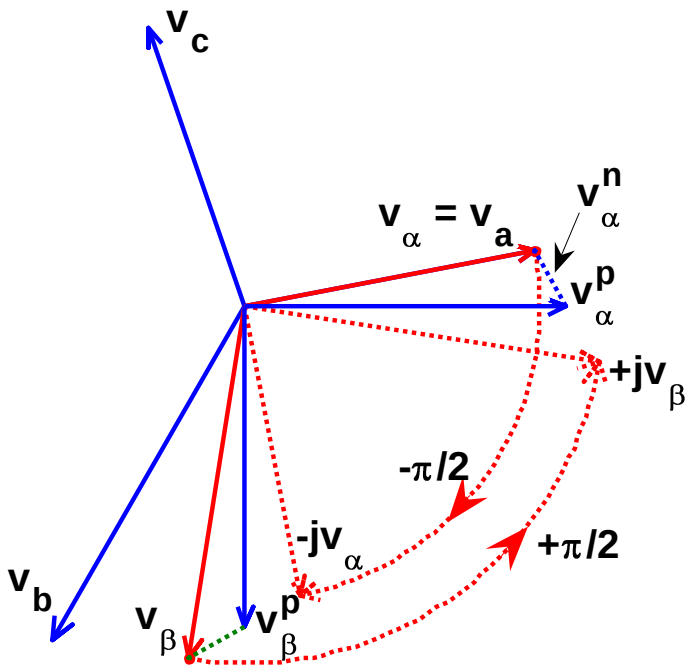


Fig 2.17 – Three-phase unbalanced phasors

The above observations suggest the following operations to separately extract the positive and negative sequence components from the measured signal $\mathbf{v}_{\alpha\beta}(t)$:

a) perform a phase shift operation of -90° and +90° on signals v_α and v_β, to produce signals $-jv_\alpha$ and $+jv_\beta$ respectively;

b) the desired positive sequence component v_α^p is then mid-way between v_α and $+jv_\beta$, and likewise v_β^p is mid-way between v_β and $-jv_\alpha$

These operations are usefully expressed as follows:

$$v_\alpha^p = v_\alpha + jv_\beta \,/2$$
$$v_\beta^p = v_\beta - jv_\alpha \,/2 \qquad (2.22\text{-a})$$

$$v_\alpha^n = v_\alpha - jv_\beta \,/2$$
$$v_\beta^n = v_\beta + jv_\alpha \,/2 \qquad (2.22\text{-b})$$

or, in matrix form:

$$\begin{pmatrix} v_\alpha^p \\ v_\beta^p \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & j \\ -j & 1 \end{pmatrix} \times \begin{pmatrix} v_\alpha \\ v_\beta \end{pmatrix}$$

$$= \mathbf{T^p} \times \begin{pmatrix} v_\alpha \\ v_\beta \end{pmatrix} \tag{2.23}$$

Similarly, the negative sequence components can be calculated with:

$$\begin{pmatrix} v_\alpha^n \\ v_\beta^n \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & -j \\ j & 1 \end{pmatrix} \times \begin{pmatrix} v_\alpha \\ v_\beta \end{pmatrix}$$

$$= \mathbf{T^n} \times \begin{pmatrix} v_\alpha \\ v_\beta \end{pmatrix} \tag{2.24}$$

It is noteworthy to observe the effect of each of the above operations on the measured signal $\mathbf{v}_{\alpha\beta}$ with regards to the components of opposite sequence. For example, keeping in mind the quadrature relationships between the α and β components of each of the positive and negative sequences, expressed by the +j and –j factors, respectively:

$$\mathbf{T^p} \times \begin{pmatrix} v_\alpha \\ v_\beta \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & j \\ -j & 1 \end{pmatrix} \times \begin{pmatrix} v_\alpha \\ v_\beta \end{pmatrix}$$

$$= \frac{1}{2}\begin{pmatrix} 1 & j \\ -j & 1 \end{pmatrix} \times \left(\begin{pmatrix} v_\alpha^p \\ v_\beta^p \end{pmatrix} + \begin{pmatrix} v_\alpha^n \\ v_\beta^n \end{pmatrix} \right)$$

$$= \frac{1}{2}\begin{pmatrix} 1 & j \\ -j & 1 \end{pmatrix} \times \left(\begin{pmatrix} 1 \\ -j \end{pmatrix} v_\alpha^p + \begin{pmatrix} 1 \\ j \end{pmatrix} v_\alpha^n \right)$$

$$= \frac{1}{2}\left(\begin{pmatrix} 1 - j^2 \\ -j - j \end{pmatrix} v_\alpha^p + \begin{pmatrix} 1 + j^2 \\ -j + j \end{pmatrix} v_\alpha^n \right)$$

$$= \frac{1}{2}\left(\begin{pmatrix} 2 \\ -2j \end{pmatrix} v_\alpha^p + \begin{pmatrix} 0 \\ 0 \end{pmatrix} v_\alpha^n \right)$$

$$= \begin{pmatrix} v_\alpha^p \\ v_\beta^p \end{pmatrix}$$

Thus, the $\mathbf{T^p}$ operation on $\mathbf{v}_{\alpha\beta}$ effectively cancels the negative sequence component out of the desired positive sequence result, and likewise $\mathbf{T^n}$ cancels the

positive sequence component out of the desired negative sequence result. These operations can be depicted as in Fig. 2.18, where PSC designates positive sequence calculation, and NSC and PSNC would likewise designate negative, and positive-negative, sequence calculation, respectively, combining $\mathbf{T^p}$ and $\mathbf{T^n}$, to produce four components, two positive and two negative sequence components.

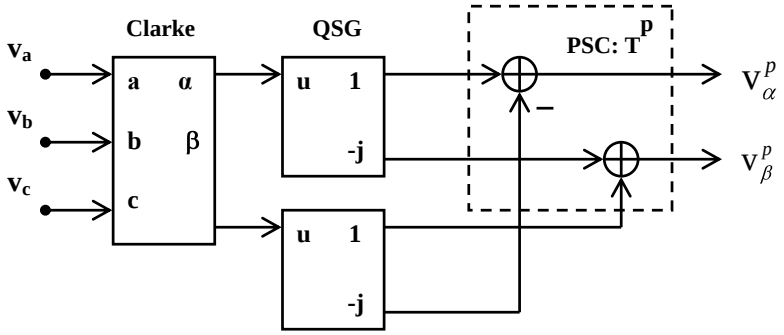


Fig. 2.18 – Positive Sequence Calculation

The blocks designated QSG refer to quadrature signal generation, for producing -90^o phase shifted signals, at outputs labeled "-j", from their respective inputs u. There are various techniques to perform this phase shifting operation, as discussed earlier in section 2.4.

2.8 INTERNAL MODEL PRINCIPLE

A very wide range of filtering techniques and design methodologies can be said to generally aim at extracting information from a physical signal. For example, a television receiver extracts, in several stages, the video signal for a selected channel, for presentation on a screen. In power electronics and control applications, the aim is to produce a usable and reliable output phase information signal, such as for the purpose of accurate timing of power control switching devices, on the basis of the physically available signal, which may be a distorted version of the information-bearing signal of interest. Thus, generally intrinsic to many filters is the aspect of eliminating, or attenuating, the impact of unwanted

components in a given physical signal, such as adjacent radio channels, distortion, and noise. An alternate approach to this process is to estimate the signal of interest in the presence of all additional unwanted components. The output result of this estimation process is then an approximation of the signal of interest, and the quality of this estimate depends on various design parameters of the filtering process.

A relevant implication of this estimation process is that the resulting filtered signal is effectively generated by a sub-component within the filtering mechanism, which itself is equivalent to the source which produces the signal of interest, to which distortion is subsequently added by intermediate physical phenomena. Thus, the filtering algorithm aims to minimize the error between the original signal of interest and the generated estimate, and effectively embeds an equivalent model of the signal source. This constitutes the so-called internal model principle [73], which is used in the proposed PLL techniques, and which is presented in this section, as follows:

a) the IMP is introduced for a simple first order low-pass filter, to establish the underlying mechanism to be exploited;
b) the principle is further demonstrated for single-phase sinusoidal signals, in the form of a band-pass filter, previously introduced, and as a basis for two-phase signal estimation;
c) the above sinusoidal estimator is extended to process a quadrature input vector signal, in the form of a positive sequence filter (PSF), to filter quadrature signals in the αβ reference frame.

2.8.1 First Order Low-Pass Filter

In the process of estimating a signal of interest, such as with a filter, the resulting signal estimate is expected, at least in the absence of any added distortion, to converge exactly to the signal of interest being estimated. For example, the estimation of a temperature transducer signal, consisting of a slowly varying

voltage, filtered by a simple first order low-pass filter, may be characterized by the following differential equation:

$$\tau\, v_o'(t) + v_o(t) = v_s(t) \qquad (2.25)$$

Such a filter can be depicted by the block diagram in Fig. 2.19.

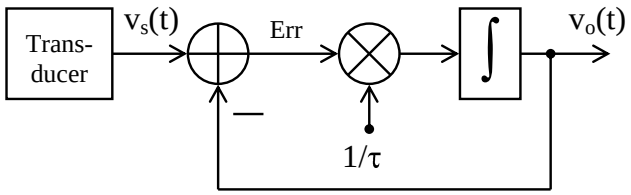


Fig. 2.19 – First order low-pass filter

The signal $v_s(t)$ is assumed to vary slowly, relative to the filter response time, such that, for the present purposes, it can be considered to be constant. Then, according to (2.25), it is expected that $v_o(t)$ will converge to $v_s(t)$, and therefore that the error signal will converge to zero. For constant input $v_s(t) = V_s$, and initial condition $v_o(0) = v_{oo}$:

$$v_o(t) = v_{oo} + (V_s - v_{oo})(1 - e^{-t/\tau})$$

$$Err(t) = (v_s - v_{oo})\, e^{-t/\tau}$$

as also exemplified in Fig. 2.20.

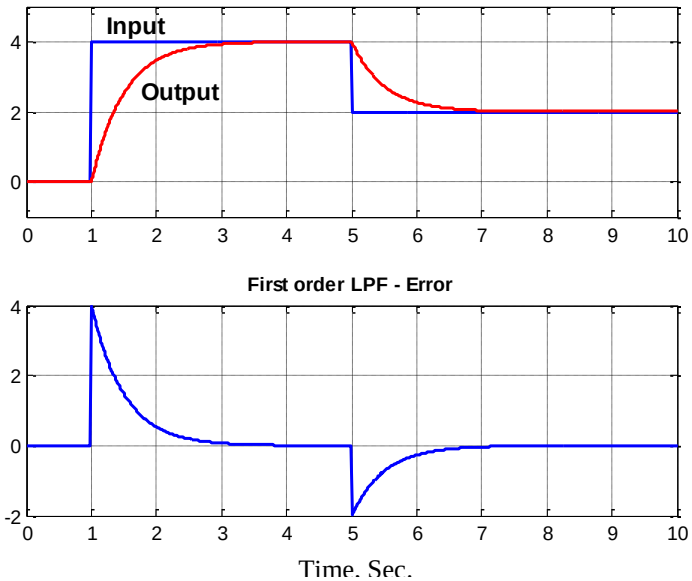


Fig. 2.20 – First order LPF step response

For the purpose of establishing the salient features of the internal model principle, it is necessary to distinguish two aspects in the above scenario. Whereas the actual signal may be produced by a physical device, the transducer, it's equivalent model within the filter, which progressively drives the error to zero, as claimed by the IMP, is characterized by dynamic properties, described by a differential equation in the time domain, or by an equivalent transfer function in terms of the Laplace s variable. In the s domain, the transfer function description for the LPF error signal is:

$$T_E(s) = \frac{Err}{V_s}(s) = \frac{1}{1 + 1/\tau s} \tag{2.26}$$

In the Laplace s domain, a constant signal in the time domain has a zero frequency, that is, s = 0. It is thus already apparent in the above that the filter steady state error response at zero frequency will be zero, due to the infinite value of the denominator. Indeed, the error response to a constant input $v_s(t) = A$, is:

$$Err(s) = V_s(s) \times T_E(s) = \frac{A}{s} \times \frac{1}{1 + 1/\tau s} = \frac{A/s}{1 + 1/\tau s} \tag{2.27}$$

The above expresses that the input constant signal $V_s(s) = A/s$, in the numerator, is matched by a corresponding integration 1/s in the denominator, and the time constant τ determines the filter's rate of convergence, or bandwidth. The steady state error response to such a constant input can be determined using Laplace's final value theorem:

$$\begin{aligned} Err(t \to \infty) &= \lim_{s \to 0} s \times \frac{A/s}{1 + 1/\tau s} = \lim_{s \to 0} s \times \frac{A\tau}{\tau s + 1} \\ &= 0 \end{aligned} \tag{2.28}$$

as expected. Thus, from the perspective of driving the filtering estimation error to zero, the practical significance of the above discussion is that the equivalent dynamic model of the transducer with constant output voltage is an integrator, as suggested in Fig. 2.21, which is characterized by it's step response to the elementary Dirac impulse function $\delta(t)$.

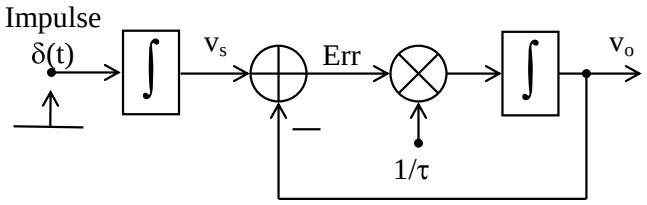


Fig. 2.21 – Equivalent model of signal source

One final and most important point in the above examination, and which will serve subsequent motives, is that at the characteristic frequency of the signal source, namely zero, the corresponding integration block in the filtering operation has a gain of infinity, which is the principal mechanism which results in the zero steady state error, that is, at that same characteristic frequency. This zero steady state error can be seen to be achieved by the fact that any non-zero error signal is continuously integrated, until the output reaches the input, that is, until the error reaches zero.

2.8.2 Second-Order Generalized Integrator And Band-Pass Filter

In view of the proposed PLL technique to be presented later on, the internal model principle is now demonstrated for the case of a sinusoidal signal. To estimate a single-phase sinusoidal signal, such as in the presence of distortion, a band-pass filter structure can be used. To set the framework for subsequent developments, this structure will be reviewed from the perspective of the internal model principle. The core of such a band-pass filter consists of a second order generalized integrator (an un-damped resonator), with center frequency ω_o matching the frequency of the input sinusoidal signal of interest v(t), as previously shown in Fig. 2.8. The transfer function from V to X is:

$$T_{XV}(s) = \frac{X}{V}(s) = \frac{\omega_o / s}{1 + \omega_o^2 / s^2} = \frac{\omega_o s}{s^2 + \omega_o^2} \tag{2.29}$$

which is characterized by an infinite gain at it's center frequency ω_o, that is:

$$\lim_{s \to j\omega_o} s^2 + \omega_o^2 \to 0$$

such that $|T_{XV}(j\omega_o)| \to \infty$. The response to an impulse input $v(t) = \delta(t)$, which establishes an initial condition $x(0) = \omega_o$, is:

$$x(0) = \int_{0-}^{0+} \omega_o V(t)dt = \int_{0-}^{0+} \omega_o \delta(t)dt = \omega_o \tag{2.30}$$

such that x(t) is simply an un-damped oscillation:

$$x(t) = \omega_o \cos(\omega_o t) := h_x(t)$$

where $h_x(t)$ designates the impulse response at x(t). For a sinusoidal input $v(t) = \cos(\omega_o t)$, the response is the convolution of v(t) and $h_x(t)$, which, in Laplace terms, is:

$$\begin{aligned} X(s) = V(s) \times H_X(s) &= \frac{s}{s^2 + \omega_o^2} \times \frac{\omega_o s}{s^2 + \omega_o^2} \\ &= \frac{\omega_o s^2}{s^2 + \omega_o^{2\ 2}} \end{aligned} \tag{2.31}$$

In the time domain, y(t) is then the inverse Laplace transform of the above:

$$y(t) = \frac{1}{2}\ \sin(\omega_o t) + \omega_o t \cos(\omega_o t) \tag{2.32}$$

which consists of two components:

a) $\omega_o \times t \times \cos(\omega_o t)$, increasing linearly with time, and which exhibits the integrating behaviour of interest, in response to a sinusoidal input at the same frequency;

b) $\sin(\omega_o t)$, with constant amplitude; once embedded in a negative feedback filter structure, such as a band-pass filter, this component simply vanishes.

Just as with the first order low-pass filter discussed earlier, it can be seen how the resonator structure has the same denominator in the Laplace transform description as the sinusoidal input signal, just as the resonator impulse response has the same time domain description as the input signal, which results in the response growing toward infinity. This infinite gain at ω_o, embedded in a band-pass filter, results in a unity gain tracking performance at ω_o.

The band-pass filter structure of Fig. 2.9 produces outputs characterized by transfer functions (2.9–2.11), with the following significant features of immediate interest:

a) T_{IU} describes a band-pass filter (BPF), with peak unit magnitude and zero phase shift response at it's center frequency ω_o; thus, at ω_o, the estimation error is 0, as also confirmed by T_{EU}, presented below;

b) T_{QU} describes a low-pass filter (LPF), with low-frequency gain K, and unit magnitude response with 90^o lagging phase relationship relative to output I, at ω_o. That is, $T_{QU}(s = j\omega_o) = \omega_o/s = \omega_o/j\ \omega_o = 1/j = -j = e^{-j\pi/2}$.

Fig. 2.22 shows the band-pass and notch filter responses for K = 0.2, 0.5, and 0.8, which determines the filter bandwidth.

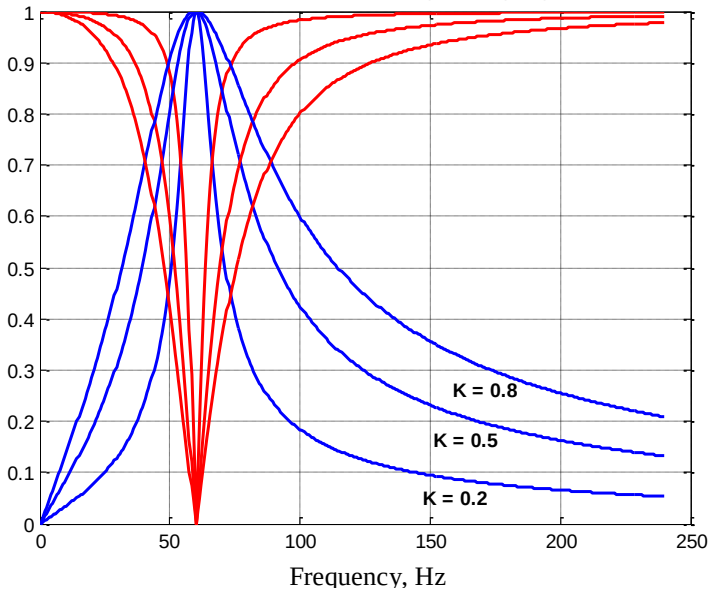


Fig. 2.22 – BPF and notch filter magnitude responses, K = 0.2, 0.5, 0.8

The time domain response of the filter to a sinusoidal step input at frequency ω_o is exemplified in Fig. 2.23, and shows how the filter estimate error eventually settles to zero. The top two traces are for K = 0.3, whereas the bottom trace shows only the filter state vector magnitude, defined as $\|\mathbf{X}\| = \sqrt{x_1^2 + x_2^2}$, for different values of K; higher values of K result in faster response, i.e. faster estimation convergence.

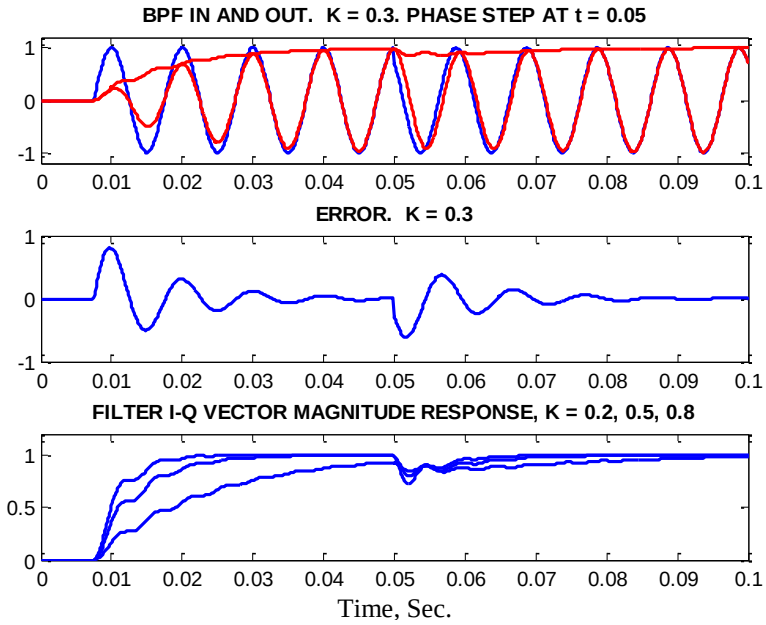


Fig. 2.23 – Filter response to sinusoidal step input

2.8.3 Positive Sequence Filter

The single-input second order generalized integrator can be extended with a second input for the purpose of estimating a two-input quadrature signal, in the form of a positive sequence generalized integrator (PSGI), shown in Fig. 2.24.

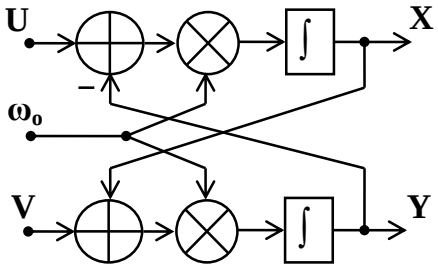


Fig. 2.24 – Positive sequence generalized integrator

With two inputs and two outputs, the PSGI can be described by the following transfer functions:

$$T_{XU}(s) = \frac{X(s)}{U(s)} = \frac{\omega_o / s}{1 + \left(\omega_o / s\right)^2} = \frac{\omega_o s}{s^2 + \omega_o^2} \qquad (2.33\text{-a})$$

$$T_{YU}(s) = \frac{Y(s)}{U(s)} = T_{XU}(s) \times \frac{\omega_o}{s} = \frac{\omega_o^2}{s^2 + \omega_o^2} \qquad (2.33\text{-b})$$

$$T_{YV}(s) = \frac{Y(s)}{V(s)} = T_{XU} = \frac{\omega_o s}{s^2 + \omega_o^2} \tag{2.33-c}$$

$$T_{XV}(s) = \frac{X(s)}{V(s)} = T_{YV}(s) \times \frac{-\omega_o}{s} = \frac{-\omega_o^2}{s^2 + \omega_o^2} \tag{2.33-d}$$

Furthermore, since the PSGI operation is intended specifically to operate on a (complex) balanced two-input quadrature signal, to evaluate it's performance in response to a positive and negative sequences, the two input signals must be considered together, which is more conveniently addressed if the above transfer functions are expressed in matrix form:

$$\begin{pmatrix} X(s) \\ Y(s) \end{pmatrix} = \begin{pmatrix} T_{XU}(s) & T_{XV}(s) \\ T_{YU}(s) & T_{YV}(s) \end{pmatrix} \times \begin{pmatrix} U(s) \\ V(s) \end{pmatrix}$$

$$= \frac{\omega_o}{s^2 + \omega_o^2} \times \begin{pmatrix} s & -\omega_o \\ \omega_o & s \end{pmatrix} \times \begin{pmatrix} U(s) \\ V(s) \end{pmatrix} \tag{2.34}$$

Thus, with a positive input sequence of unit magnitude:

$$u(t) = \cos(\omega_o t) = \mathrm{re}(e^{j\omega_o t})$$

$$v(t) = \sin(\omega_o t) = \mathrm{re}(-j\, e^{j\omega_o t})$$

such that, with $s = j\omega_o$:

$$\begin{pmatrix} X(j\omega_o) \\ Y(j\omega_o) \end{pmatrix} = \mathrm{re}\left(\frac{\omega_o}{s^2 + \omega_o^2} \times \begin{bmatrix} j\omega_o & -\omega_o \\ \omega_o & j\omega_o \end{bmatrix} \times \begin{bmatrix} 1 \\ -j \end{bmatrix} \times e^{j\omega_o t} \right)$$

$$= \mathrm{re}\left(\frac{\omega_o^2}{s^2 + \omega_o^2} \times \begin{bmatrix} j & -1 \\ 1 & j \end{bmatrix} \times \begin{bmatrix} 1 \\ -j \end{bmatrix} \times e^{j\omega_o t} \right)$$

$$= \mathrm{re}\left(\frac{2\omega_o^2}{s^2 + \omega_o^2} \times \begin{bmatrix} j \\ 1 \end{bmatrix} \times e^{j\omega_o t} \right) \tag{2.35}$$

Two important observations in the above are that:

a) the denominator $s^2 + \omega_o^2$ has an infinite value for an input signal frequency ω_o, which corresponds to the integrating behaviour of the filter in response to an input sequence at ω_o, just as with a single-phase input;

b) the factor $\begin{bmatrix} j \\ 1 \end{bmatrix}$ expresses the quadrature phase relationship of the outputs relative to each other and with respect to the respective input signals.

However, with a negative input sequence:

$$u(t) = \cos(\omega_o t) = \mathrm{re}(e^{j\omega_o t})$$

$$v(t) = -\sin(\omega_o t) = \mathrm{re}(j\,e^{j\omega_o t})$$

then:

$$\begin{pmatrix} X(j\omega_o) \\ Y(j\omega_o) \end{pmatrix} = \mathrm{re}\left(\frac{\omega_o^2}{s^2+\omega_o^2} \times \begin{bmatrix} j & -1 \\ 1 & j \end{bmatrix} \times \begin{bmatrix} 1 \\ j \end{bmatrix} \times e^{j\omega_o t} \right)$$

$$= \mathrm{re}\left(\frac{\omega_o^2}{s^2+\omega_o^2} \times \begin{bmatrix} 0 \\ 0 \end{bmatrix} \times e^{j\omega_o t} \right) \qquad (2.36)$$

Thus, this PSGI has the very important and defining characteristic of a zero response to a negative input sequence, and an infinite response to a positive input sequence. Note that although the same denominator $s^2+\omega_o^2$ also has an infinite value at ω_o, the internal structure of the filter effectively cancels this infinite factor, as expressed in (2.36), and therefore does not result in an infinite response, which will in fact be seen to vanish in the proposed overall operation.

Just as with the single-phase band-pass filter discussed previously, the PSGI can be embedded in a similar negative feedback structure, as shown in Fig. 2.25,

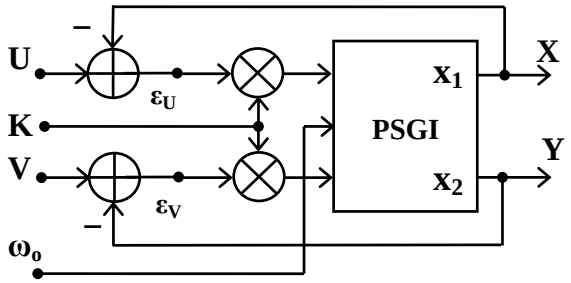


Fig. 2.25 – Positive-Negative Sequence Filter

with the following state variable description:

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \omega_o \begin{pmatrix} -K & -1 \\ 1 & -K \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + K\omega_o \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u(t) \\ v(t) \end{pmatrix} \qquad (2.37)$$

which results in the following transfer function matrix:

$$T_{PSF}(s)=\begin{bmatrix} T_{XU}(s) & T_{XV}(s) \\ T_{YU}(s) & T_{YV}(s) \end{bmatrix}=K^2\omega_o^2\frac{\begin{bmatrix} s/K\omega_o+1 & -1/K \\ 1/K & s/K\omega_o+1 \end{bmatrix}}{s^2+2K\omega_o s+\left(1+K^2\right)\omega_o^2} \tag{2.38}$$

For a positive input sequence at $s = j\omega_o$, the above results in:

$$T_{PSF}(j\omega_o)=K^2\omega_o^2\times\frac{\begin{bmatrix} j/K+1 & -1/K \\ 1/K & j/K+1 \end{bmatrix}}{-\omega_o^2+j2K\omega_o^2+\left(1+K^2\right)\omega_o^2}\times\begin{bmatrix}1\\-j\end{bmatrix} \tag{2.39}$$

$$=K^2\omega_o^2\times\frac{\begin{bmatrix} j/K+1 & -1/K \\ 1/K & j/K+1 \end{bmatrix}}{K^2\omega_o^2(j2/K+1)}\times\begin{bmatrix}1\\-j\end{bmatrix}$$

$$=\frac{\begin{bmatrix} 1+j2/K \\ -j\ \ 1+j2/K \end{bmatrix}}{1+j2/K}=\begin{bmatrix}1\\-j\end{bmatrix}$$

In other words, the internal resonator produces outputs x(t) and y(t) exactly matching inputs u(t) and v(t), respectively, resulting in zero error signals $\varepsilon_U(t)$ and $\varepsilon_V(t)$, leaving the resonator effectively un-coupled form the inputs. On the other hand, a negative input sequence at the same frequency ω_o produces a response:

$$K^2\omega_o^2\times\frac{\begin{bmatrix} j/K+1 & -1/K \\ 1/K & j/K+1 \end{bmatrix}}{-\omega_o^2+j2K\omega_o^2+\left(1+K^2\right)\omega_o^2}\times\begin{bmatrix}1\\j\end{bmatrix}$$

$$=\frac{\begin{bmatrix}1\\j\end{bmatrix}}{1+j2/K}=\frac{1\angle-\tan^{-1}(4/K^2)}{\sqrt{1+4/K^2}}\begin{bmatrix}1\\j\end{bmatrix} \tag{2.40}$$

which consists of a negative sequence, $\begin{bmatrix}1\\j\end{bmatrix}$, phase shifted by an angle of $-\tan^{-1}(4/K^2)$, with a non-zero magnitude of $1/\sqrt{1+4/K^2}$. In other words, the above positive sequence filter only attenuates, and does not completely eliminate any negative sequence at the output. Furthermore, greater attenuation requires a smaller value of K, which also slows down the filter response time. This incomplete elimination of the opposite sequence by the filter is in fact the

motivation for part of the proposed structure, referred to as Positive-Negative Sequence Filter (PNSF), presented in chapter 3.

2.9 ADAPTIVE SYNCHRONOUS AMPLITUDE ESTIMATION

A sinusoidal signal is defined by three parameters, namely, amplitude, phase, and frequency. If better knowledge of one parameter can assist in the estimation of another, then there is an incentive to estimate all parameters together, resulting in potentially faster estimation of all. Thus, in contrast to more conventional single-phase PLL synchronization techniques, where only the phase is estimated, and implicitly also it's frequency, a relatively novel introduction in the last few years has been to also incorporate an estimation of the signal's amplitude [74-78], resulting primarily in (a) the elimination of any steady state double frequency error, (b) the potential elimination of any low-pass filtering to attenuate this error signal, and (c) faster overall response.

This section presents:

a) the basic mechanism of adaptive synchronous amplitude estimation (ASAE), for estimating the amplitude of a sinusoidal signal with known frequency and phase;
b) a simple extension of the ASAE to estimate the amplitude of a sinusoidal signal with known frequency and unknown phase, also referred to, in some applications, as adaptive feed-forward cancellation (AFC), [75-76].

These features will provide the necessary basis for further subsequent enhancements for two-phase and unbalanced signals.

2.9.1 Single-Phase ASAE with Known Phase

The objective of estimating the amplitude of a sinusoidal signal is not unlike the principle of synchronous rectification in power electronic systems, or of amplitude de-modulation for an amplitude-modulated radio carrier signal [36].

However, in the case of these two familiar applications, the rectification process is followed by a low-pass filtering operation aiming to attenuate the ripple due to the sinusoidal or periodic "carrier" signal, which normally implies a long time constant, relative to the carrier period, and consequently slow dynamic response, which is precisely the basis for the desired attenuation. In the case of phase estimation and synchronization, however, a faster response is desired, on the order of one to two sinusoidal cycles, thus requiring a shorter time constant in the filtering process. With this objective in mind, the so-called adaptive synchronous amplitude estimation (ASAE) algorithm proceeds as follows:

a) produce an estimate $\hat{A}(t)$ of the signal amplitude of interest at the output of an integrator; this amplitude will be subsequently adjusted, by applying a suitable error signal to the integrator input, progressively correcting this estimate until, ideally, the estimate error converges to zero;

b) assume that the input signal phase is known; this phase information can be established with any suitable PLL, and this assumption will be validated by the subsequent development. Thus, if the input signal is: $u(t) = A\cos(\omega t)$, then we have access to a locally generated and synchronized, unit magnitude, reference signal: $r(t) = \cos(\omega t)$;

c) produce an estimate $\hat{u}(t)$ of the input signal, based on the amplitude estimate and the reference signal:

$$y(t) = \hat{u}(t) = \hat{A}(t) \times \cos(\omega t)$$

d) compare this estimate with the input u(t), and produce an estimate error:

$$\begin{aligned} \varepsilon(t) &= u(t) - \hat{u}(t) = A\cos(\omega t) - \hat{A}(t) \times \cos(\omega t) \\ &= (A - \hat{A}(t)) \times \cos(\omega t) \end{aligned}$$

e) synchronously de-modulate this signal, by multiplying it with the reference signal, resulting in:

$$\begin{aligned} \varepsilon(t) \times \cos(\omega t) &= (A - \hat{A}(t)) \times \cos(\omega t) \times \cos(\omega t) \\ &= (A - \hat{A}(t)) \times \cos^2(\omega t) \\ &= (A - \hat{A}(t)) \times \tfrac{1}{2}\ 1 + \cos(2\omega t) \end{aligned}$$

f) The term (1+cos(2ωt)) is of positive polarity, such that the above error signal has a polarity consistent with that of the estimate error, and can thus be used to correct the amplitude estimate $\hat{A}(t)$, as suggested in Fig. 2.26. User designed gain K_A allows to adjust the rate of convergence of the estimation process.

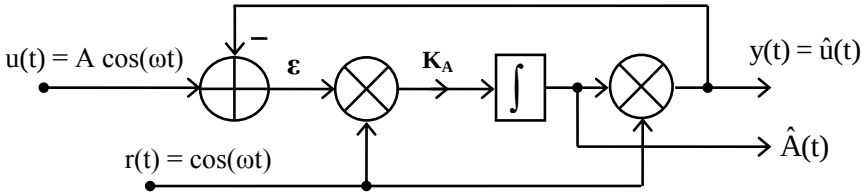


Fig 2.26 – Adaptive Synchronous Amplitude Estimation

The most important feature of the above mechanism is that as the amplitude estimate $\hat{A}(t)$ converges toward the actual input signal amplitude A, the error is driven exactly toward zero, resulting in zero ripple in the error and output signals, and therefore, the estimation adaptation gain K_A can be selected arbitrarily high, resulting in much faster response time than with a conventional low-pass filtering operation. This process can be described by the following non-linear dynamic system:

$$\begin{aligned}\dot{\hat{A}} &= K_A \times \cos(\omega t) \times (u(t) - \hat{A}(t) \times \cos(\omega t)) \\ &= K_A \times \cos(\omega t) \times (A\cos(\omega t)) - \hat{A}(t) \times \cos(\omega t)) \\ &= K_A \times \cos^2(\omega t) \times A - \hat{A}(t) \\ &\approx K_A/2 \times A - \hat{A}(t)\end{aligned} \tag{2.41}$$

which can be represented by the simplified first order system in Fig. 2.27.

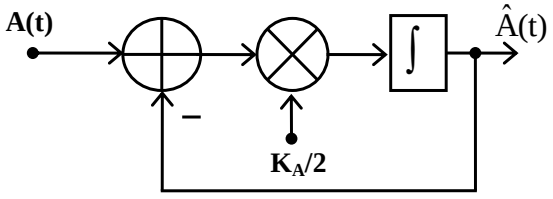


Fig 2.27 – ASAE first order simplified model

The approach implied in the preceding analysis is clearly based on the idea of averaging the $\cos^2(\omega t)$ function, as was also used earlier in the analysis of the adaptive notch filter, and is a common tool for analyzing non-linear dynamical systems, as may be justified, [66-69]. Thus, the approximation indicated above really applies only for smaller values of K_A, resulting in slower convergence, and serves more to get a better sense of the estimation operation. However, being a first order system, and since the derivative of the estimated amplitude can be seen to be always in the same direction as the amplitude estimate error $A - \hat{A}(t)$, as assured by the non-negative factor $\cos^2(\omega t)$, it can be seen that the algorithm will always converge. As a result, the factor K_A can be made arbitrarily high, or small, depending on specific application requirements. Fig. 2.28 shows, for example, (a) in the top trace, the amplitude estimation process for K = 0.1, 0.2, and 1, and (b) amplitude estimation correction signal $\cos^2(\omega t) \times A - \hat{A}(t)$, with K = 0.1, in the lower trace.

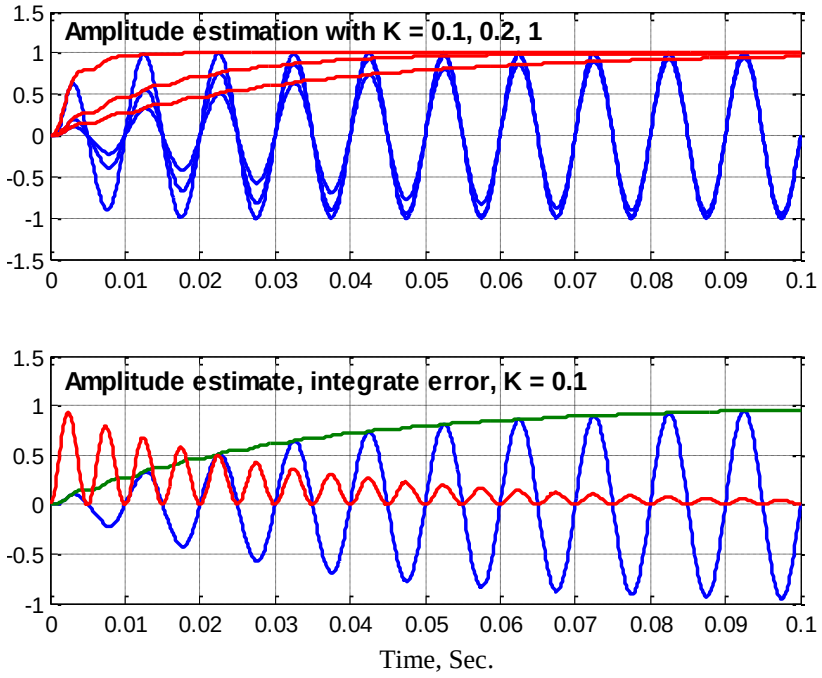


Fig 2.28 – ASAE process and convergence

2.9.2 Single-Phase ASAE with Unknown Phase

The previous ASAE algorithm, which assumed a priori knowledge of the input signal phase, can be easily extended to the case where only the frequency is known, but not the phase, as shown in Fig. 2.29. The input signal must now be represented in terms of two quadrature components:

$$\begin{aligned} u(t) &= A\cos(\omega t-\varphi) \\ &= A_c\cos(\omega t) + A_s\sin(\omega t) \\ &= A\cos(\varphi)\cos(\omega t) + A\sin(\varphi)\sin(\omega t) \end{aligned} \tag{2.42}$$

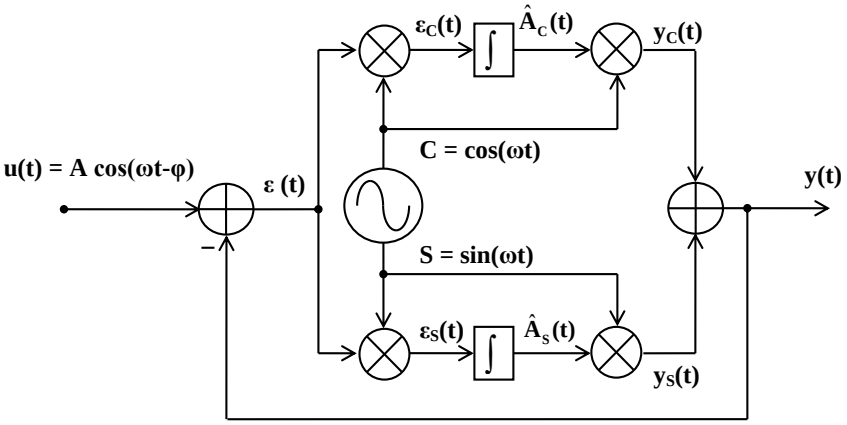


Fig. 2.29 – Single-phase ASAE, for unknown phase

Similarly to the previous development, this estimation algorithm can be described in terms of two separate amplitude estimation processes, one for amplitude A_c and another for amplitude A_s, which fundamentally depend on the orthogonality of the cos and sin trigonometric functions. The amplitude estimation process can thus be described by the following non-linear dynamic equations:

$$\begin{aligned} y(t) &= y_C(t) + y_S(t) \\ &= \hat{A}_C(t)\cos(\omega t) + \hat{A}_S(t)\sin(\omega t) \end{aligned} \tag{2.43}$$

$$\begin{aligned} \dot{\hat{A}}_C(t) &= K_A(u(t) - y(t)) \times \cos(\omega t) \\ &= K_A\left(\begin{array}{l} A_C\cos(\omega t) + A_S\sin(\omega t) \\ - \quad \hat{A}_C(t)\cos(\omega t) + \hat{A}_S(t)\sin(\omega t) \end{array} \right) \times \cos(\omega t) \\ &= K_A \quad A_C - \hat{A}_C(t) \;\; \cos(\omega t) + \;\; A_S - \hat{A}_S(t) \;\; \sin(\omega t) \;\; \times \cos(\omega t) \end{aligned}$$

$$= K_A \;\; A_C - \hat{A}_C(t) \;\; \cos^2(\omega t) + K_A \;\; A_S - \hat{A}_S(t) \;\; \sin(\omega t)\cos(\omega t)$$

$$\approx K_A \;\; A_C - \hat{A}_C(t) \;\; \cos^2(\omega t)$$

Likewise:

$$\dot{\hat{A}}_S(t) = K_A(u(t) - y(t)) \times \sin(\omega t)$$

$$= K_A \;\; A_C - \hat{A}_C(t) \;\; \sin(\omega t)\cos(\omega t) + K_A \;\; A_S - \hat{A}_S(t) \;\; \sin^2(\omega t)$$

$$\approx K_A \;\; A_S - \hat{A}_S(t) \;\; \sin^2(\omega t)$$

As indicated, the above approximations are validated for smaller values of K_A, in which case, the factor $\cos(\omega t)\times\sin(\omega t)$ contributes only to a small ripple, and is effectively averaged out, and the two estimation processes can be seen to be very conveniently represented by two independent first order dynamic processes, as exemplified in Fig. 2.30, for $K_A = 0.2$. The top trace also shows the transient amplitude estimation error, as the amplitude dips after phase changes in the signal.

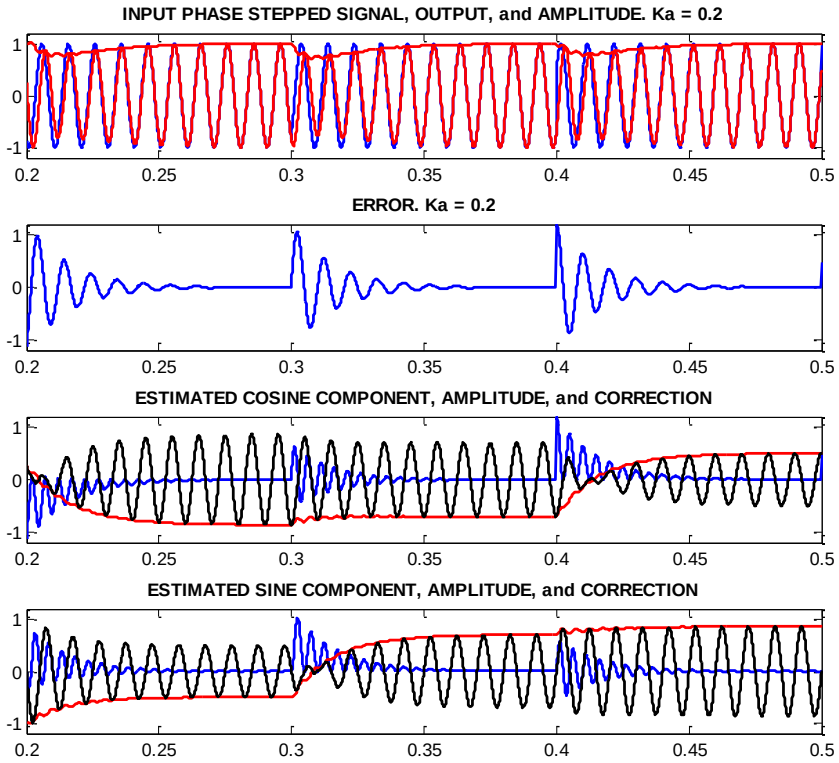


Fig. 2.30 – ASAE estimation of quadrature components, $K_A = 0.2$

With a higher value of K_A = 1, the process is also observed to converge more rapidly, as shown in Fig. 2.31. Strictly speaking, due to the non-linear character of this dynamic system, a more exacting stability analysis would be required for faster rates of convergence with higher values of K_A, such as with Lyapunov techniques.

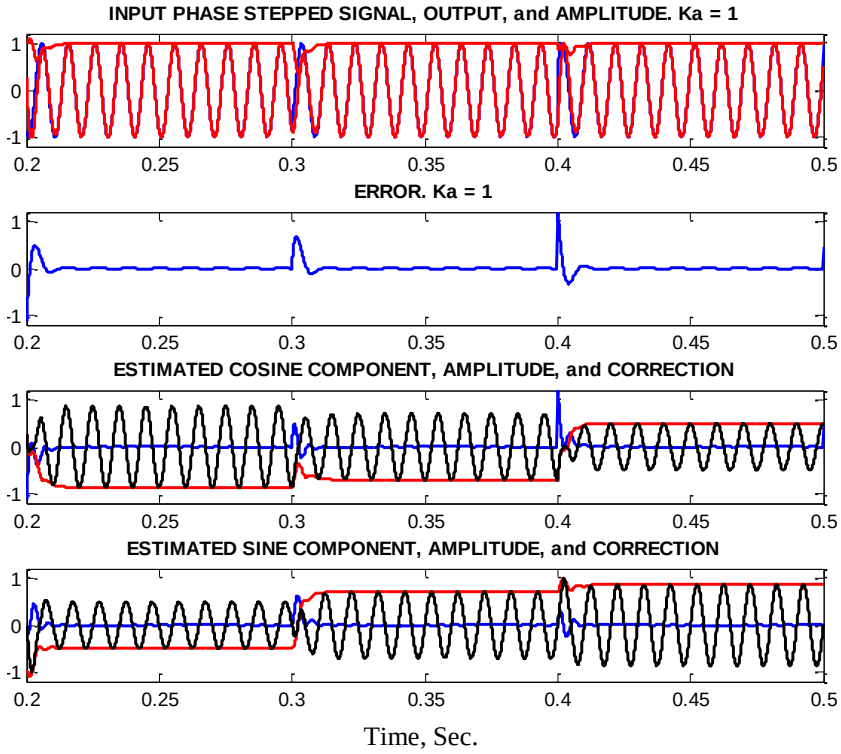


Fig 2.31 – ASAE estimation of quadrature components. Ka = 1

However, this particular algorithm, with it's internal products by sinusoidal signals, can be conveniently analyzed using Laplace transforms, and shown to be equivalent in behaviour, from an input-output point of view, to a simple LTI band-pass filter, as will be discussed later.

2.10 SINGLE-PHASE MAGNITUDE/ENHANCED PLL

The ASAE algorithm for known phase, discussed in section 2.9.1, can be described as in Fig. 2.32, where the synchronous signal r(t) is estimated

independently by a single-phase PLL. However, using a conventional single-phase PLL requires that it's bandwidth be reduced, due to the filtering necessary to attenuate the impact of the phase detector output ripple, which inevitably results in slower response.

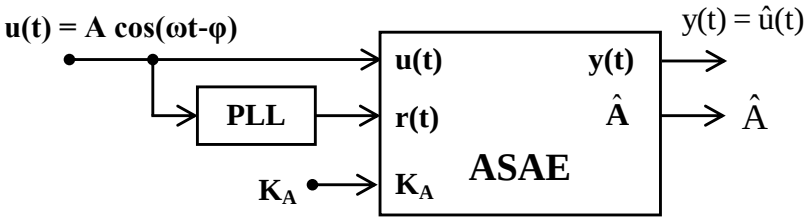


Fig. 2.32 – ASAE with single-phase PLL

A significant, yet simple, innovation was contributed in recent years to enhance the performance of the conventional single-phase PLL, referred to as a Magnitude/PLL, [79-80], or also Enhanced PLL (EPLL), [2, 81], shown in Fig. 2.33.

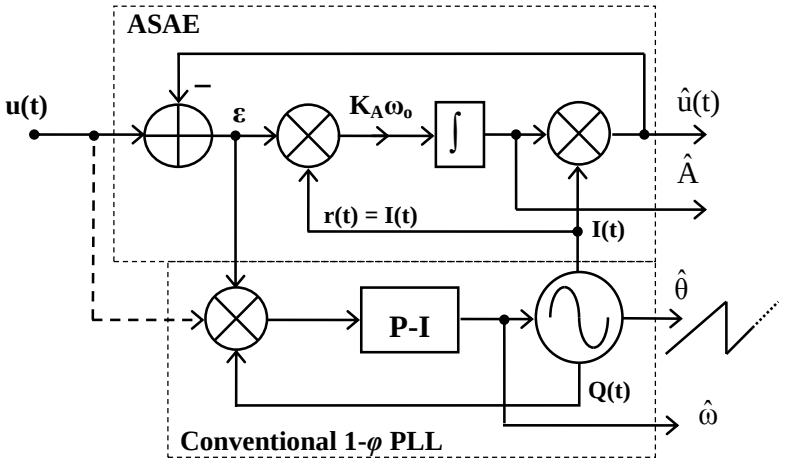


Fig. 2.33 – Magnitude/Enhanced PLL

The enhancement consists essentially in driving the PLL not from the input signal u(t) itself, as indicated in Fig. 2.32 and by the dashed line in Fig. 2.33, but from the error signal in the ASAE process. Although the PLL operation depends on an error signal ε, which itself depends on proper synchronization expected from the PLL via signal I(t), the overall dynamical system can be designed to converge towards a stable operation, as exemplified in Fig. 2.34, and discussed further

below. A detailed analysis of the internal dynamics and stability of the EPLL is discussed in [82]. A very useful conclusion of this analysis is that the linearized model reveals two un-coupled dynamic sub-systems, related to the ASAE process and to the PLL, respectively. For the sake of more meaningful analysis and comparison in the present discussion, the P-I controller internal structure used here is the same as adopted earlier, as in Fig. 2.3.

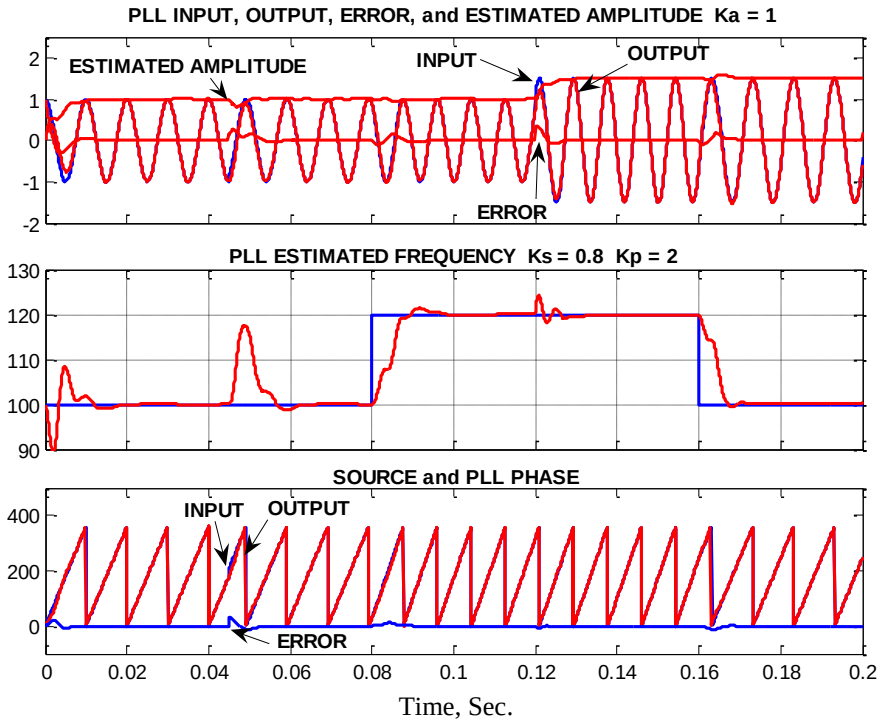


Fig. 2.34 – M/EPLL operation

Fig. 2.34 shows an example of the M/EPLL operation, and emphasizes a few significant features:

a) the test signal consists of an un-distorted sinusoid, with (a) a 30^o phase step at t = 0.045 sec., (b) a +20 Hz frequency jump at t = 0.08 sec., (c) a +50% amplitude jump at t = 0.12 sec., and (d) a -20 Hz frequency jump at t = 0.16 sec.;

b) the PLL parameters are chosen to be: $K_a = 1$, $K_s = 0.8$, and $K_p = 2$, for a relatively fast response, on the order of one cycle of the input signal at the nominal frequency, here 100 Hz; the error signal can be seen to settle in under one cycle, and so does the estimated amplitude settle about as fast;

c) most importantly, the steady state error can be seen to settle to zero, and there is therefore no ripple signal inside the PLL closed loop;
d) this ripple-free performance is also seen in the estimated frequency signal, in the middle trace, which settles to a constant value;
e) the estimated amplitude suffers a modest transient following a jump in phase and frequency; this is inevitable, since the same error signal is used to correct both the PLL frequency and phase, as well as the amplitude estimation;
f) the bottom trace displays the source signal phase and the PLL estimated phase, and their difference, and it also can be seen to settle in under one cycle;
g) although not shown, the gains can be also adjusted for slower response, resulting in narrower bandwidth and more robust performance in the face of noise and distortion.

This M/EPLL integrates well the salient features of the adaptive synchronous amplitude estimation and conventional PLL mechanisms, and can be seen to be capable of fast convergence, yet without any steady state ripple. A final, noteworthy observation is that the one estimate error signal ε is used for both amplitude and frequency/phase correction, by effectively correlating it, that is, multiplying it, with I and Q signals, respectively. This mechanism will prove useful when extended in the proposed techniques in chapter 3.

2.11 EQUIVALENCE OF ASAE TO LTI IMP

Although the ASAE algorithm is characterized by non-linear dynamical equations, and is therefore not easily mathematically tractable, for the particular case where input and intermediate signals are multiplied by sinusoidal signals, the Laplace transform's frequency shifting property can be used to express the input-output relationship in terms of linear time-invariant dynamics. This will greatly simplify subsequent mathematical manipulations, since more traditional and well

established tools can then be applied to obtain required results. Furthermore, for lower values of K_A, which result in slower convergence and equivalently narrower bandwidth, the ASAE algorithm can be usefully approximated by two independent first order dynamic systems, in contrast to a single second order resonator. From the perspective of discrete implementation, such as on a DSP, this can prove very beneficial, as the ASAE algorithm can be seen to perform significantly better than a discrete implementation of a continuous time resonator, especially for lower values of K_A, i.e. for narrower filter bandwidths, as would be required for greater attenuation of distortion components. Thus, this section serves to establish the equivalence between the ASAE algorithm and a LTI band-pass filter.

2.11.1 Laplace Transform Approach

To establish the overall input-output relationship of the ASAE algorithm for unknown phase of an input signal at ω_o, it is necessary to first determine the intermediate relationship from e(t) to y(t) of the forward block, shown in fig 2.35. The overall relationship from e(t) to y(t) can then be determined by considering time domain signals and operations, and their equivalents in the Laplace transform domain, as shown on the next page.

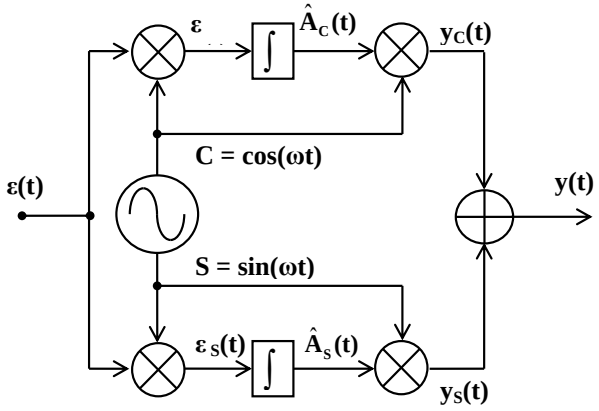


Fig 2.35 – ASAE forward block

Equivalence between time domain and Laplace s domain operations of ASAE:

TIME DOMAIN	**LAPLACE S DOMAIN**
$e(t) \times e^{j\omega t}$	$E(s-j\omega)$
$e_c(t) = e(t) \times \cos(\omega t)$ $= e(t)\,(e^{j\omega t}+e^{-j\omega t})/2$	$E_C(s) = \dfrac{E(s-j\omega)+E(s+j\omega)}{2}$
$e_s(t) = e(t) \times \sin(\omega t)$ $= e(t)\,(e^{j\omega t}-e^{-j\omega t})/2j$	$E_S(s) = \dfrac{E(s-j\omega)-E(s+j\omega)}{2j}$
$A_c(t) = \int e_c(t)dt$	$A_C(s) = \dfrac{E_C(s)}{s} = \dfrac{E(s-j\omega)+E(s+j\omega)}{2s}$
$A_s(t) = \int e_s(t)dt$	$A_S(s) = \dfrac{E_S(s)}{s} = \dfrac{E(s-j\omega)-E(s+j\omega)}{2js}$
$y_c(t) = A_c(t) \times \cos(\omega t)$ $= A_c(t)\,(e^{j\omega t}+e^{-j\omega t})/2$	$Y_C(s) = \dfrac{A_C(s-j\omega)+A_C(s+j\omega)}{2}$ $= \dfrac{1}{2}\left[\dfrac{E(s-2j\omega)+E(s)}{2(s-j\omega)}+\dfrac{E(s)+E(s+2j\omega)}{2(s+j\omega)}\right]$ $= \dfrac{E(s)\ (s+j\omega)+(s-j\omega)}{4(s-j\omega)(s+j\omega)}$ $+\dfrac{E(s-2j\omega)}{4(s-j\omega)}+\dfrac{E(s+2j\omega)}{4(s+j\omega)}$ $= \dfrac{sE(s)}{2(s^2+\omega^2)}+\dfrac{E(s-2j\omega)}{4(s-j\omega)}+\dfrac{E(s+2j\omega)}{4(s+j\omega)}$
$y_s(t) = A_s(t) \times \sin(\omega t)$ $= A_s(t)\,(e^{j\omega t}-e^{-j\omega t})/2j$	$Y_S(s) = \dfrac{A_S(s-j\omega)-A_S(s+j\omega)}{2j}$ $= \dfrac{1}{2j}\left[\dfrac{E(s-2j\omega)-E(s)}{2j(s-j\omega)}-\dfrac{E(s)-E(s+2j\omega)}{2j(s+j\omega)}\right]$ $= \dfrac{E(s-2j\omega)-E(s)}{4j^2(s-j\omega)}-\dfrac{E(s)-E(s+2j\omega)}{4j^2(s+j\omega)}$ $= \dfrac{E(s)-E(s-2j\omega)}{4(s-j\omega)}-\dfrac{E(s+2j\omega)-E(s)}{4(s+j\omega)}$

$$= \frac{E(s)\ (s + j\omega) + (s - j\omega)}{4(s - j\omega)(s + j\omega)} - \frac{E(s - 2j\omega)}{4(s - j\omega)} - \frac{E(s + 2j\omega)}{4(s + j\omega)}$$

$$= \frac{sE(s)}{2(s^2 + \omega^2)} - \frac{E(s - 2j\omega)}{4(s - j\omega)} - \frac{E(s + 2j\omega)}{4(s + j\omega)}$$

$y(t) = y_c(t) + y_s(t)$ $\qquad Y(s) = Y_C(s) + Y_S(s) = \frac{s}{(s^2 + \omega^2)} \times E(s)$

This last expression reveals precisely the operation of a sinusoidal, or second order generalized, integrator (SOGI), described earlier, on the input signal e(t). With the equivalence established, this mechanism can subsequently be used, as is suitable, instead of an actual resonator, to achieve the same functionality, with the following benefits:

a) improved numerical performance when discretized for digital implementation on a DSP, especially for narrower bandwidths and greater attenuation of distortion harmonic components;
b) the availability of the amplitudes of each of the quadrature components, making possible to determine the phase angle of the input signal relative to the reference cos and sin signals;
c) the above feature thus makes it possible to naturally incorporate this mechanism into a frequency-adaptive phase-locked feedback loop, with easily adjustable bandwidth. As well, a second such device, synchronized to the first one, can be used to detect the negative sequence component of an unbalanced input signal, it's phase and amplitude, and therefore to detect the type of fault that caused the unbalanced condition.

2.11.2 Impulse Response Approach

In order to subsequently extend this principle of equivalence to a positive sequence filter with two inputs and two outputs, it is useful to examine the internal operation of the above mechanism in response to an impulse input $e(t) = \delta(t-t_o)$ applied at an arbitrary time t_o. The internal signals then evolve as follows:

$$e_c(t) = e(t) \times \cos(\omega t) \quad \text{and:} \quad e_s(t) = e(t) \times \sin(\omega t)$$
$$= \delta(t - t_o) \times \cos(\omega t) \qquad = \delta(t - t_o) \times \sin(\omega t)$$
$$A_c(t) = \text{Step}(t - t_o) \times \cos(\omega t_o) \qquad A_s(t) = \text{Step}(t - t_o) \times \sin(\omega t_o)$$
$$:= C_o \times \text{Step}(t - t_o) \qquad := S_o \times \text{Step}(t - t_o)$$

and:

$$\begin{aligned} y(t) &= A_c(t) \times \cos(\omega t) + A_s(t) \times \sin(\omega t) \\ &= C_o \times \cos(\omega t) + S_o \times \sin(\omega t_o) \times \text{Step}(t - t_o) \\ &= \sqrt{C_o^2 + S_o^2} \times \cos(\omega t - \tan^{-1}(S_o / C_o)) \times \text{Step}(t - t_o) \\ &= \cos(\omega t - \tan^{-1}(\cos(\omega t_o) / \sin(\omega t_o)) \times \text{Step}(t - t_o) \\ &= \cos(\omega t - \tan^{-1}(\tan(\omega t_o)) \times \text{Step}(t - t_o) \\ &= \cos(\omega t - \omega t_o) \times \text{Step}(t - t_o) \\ &= \cos(\omega(t - t_o)) \times \text{Step}(t - t_o) \end{aligned}$$

Thus, as expected, the output response is time shift-invariant, identical, to that of the un-damped resonator.

2.11.3 Equivalence of Time-Varying ASAE to LTV IMP

Whereas the previous section applied the internal model principle to demonstrate the equivalence of the ASAE algorithm to a linear time-invariant band-pass filter, this same principle has more recently been also extended to demonstrate a similar equivalence for linear time variant estimator, more appropriately described in terms of state variables, [83]. This will prove more appropriate for the purpose of frequency-adaptive ASAE-based PLL, as proposed in the next chapter.

2.12 DECOUPLED DOUBLE SYNCHRONOUS REFERENCE FRAME PLL

With respect to the H-NSASAE-PLL proposed in chapter 3, which directly estimates the positive and negative sequence components, the DDSRF-PLL, [22-23], is noteworthy in that it performs a similar operation, but indirectly through additional transformations and low-pass filters, at greater computational cost. It's mechanism is also fundamentally different from the adaptive techniques adopted for the proposed H-NSASAE-PLL. Further details are addressed in [22, 23].

3. PROPOSED PHASE ESTIMATION TECHNIQUES

Based on underlying principles discussed previously, three novel positive sequence phase estimation PLL techniques are presented in this chapter, based on the novel IMP-based positive-negative sequence filter, which is presented first. Additional minor variations on existing techniques are also presented to help clarify the development of the proposed PLL techniques. These are all outlined in the table below, with the novel techniques identified by an asterisk.

Table 3.1 – Development of novel estimation techniques

	Subject	Key features
1.	* IMP-based positive-negative sequence filter	➢ zero steady state error with unbalanced two-phase sinusoidal signal ➢ minimal filter dynamical order
2.	Discretization of ASAE	➢ robust operation of discrete realization of BPF
3.	ASAE-based QSG	➢ minor addition to above for quadrature output
4.	Quadrature signal ASAE, known phase	➢ extension of ASAE for balanced quadrature signal with known phase
5.	ASAE-based PSF	➢ further extension with unknown phase
6.	* ASAE-based PNSF	➢ extension of above for unbalanced signal
7.	* PLL-based positive sequence filter	➢ extension of ASAE-based PSF with a PLL, for frequency-adaptive positive sequence filter ➢ user-adjustable bandwidth
8.	* Hybrid PLL-NSF	➢ frequency-adaptive PNSF, embedding PLL / ASAE-based PSF with resonator-based NSF ➢ validates proposed frequency-adaptive PNSF ➢ minimal order of overall PLL-based PNSF
9.	* Hybrid PLL with negative sequence adaptive synchronous amplitude estimation	➢ frequency-adaptive PNSF, embedding PLL / ASAE-based PSF with ASAE-based NSF ➢ most robust operation of discretized version of Hybrid PLL-NSF

The first eight serve mainly to establish operating principles necessary for the last one, which uniquely exhibits all the benefits most suitable for real time execution on a DSP. The other structures, which have not been encountered in the literature as such, serve as useful intermediate stages to help clarify the overall development toward the H-NSASAE-PLL, at which point all it's key features and benefits will be most evident. Altogether, the PLL phase estimation techniques can be classified in three categories, according to which parameters of a sinusoidal or quadrature signal are to be estimated:

a) amplitude of sinusoidal input with known frequency and phase,
b) estimating amplitude and phase, given known frequency,
c) estimating frequency, phase, and amplitude of a sinusoid.

3.1 IMP–BASED POSITIVE-NEGATIVE SEQUENCE FILTER

As discussed in the previous chapter, a quadrature signal with positive sequence can be estimated using the internal model principle, with a two-input two-output positive sequence filter (PSF), with two feedback signals, one each for the α and β components of the two-input signal. An unbalanced two-phase signal was also shown to be usefully represented as a sum of two balanced signals of opposite sequence. It is thus proposed to apply the IMP to combine a PSF and a negative sequence filter (NSF), in parallel, to separately estimate the positive and negative sequence components of an unbalanced two-phase $\alpha\beta$ signal, resulting in a positive-negative sequence filter (PNSF). The resulting balanced positive sequence can then be further de-modulated by a conventional two-phase PLL. Alternately, this structure also allows to actually replace the PSF component with a PLL-based PSF, resulting in a minimal order, frequency-adaptive, PNSF.

3.1.1 Proposed IMP-Based PNSF

As discussed earlier, based on a two-phase positive sequence generalized integrator (PSGI) as a building block, the internal model principle has been applied to create a two-input positive sequence filter, shown in Fig. 3.1.

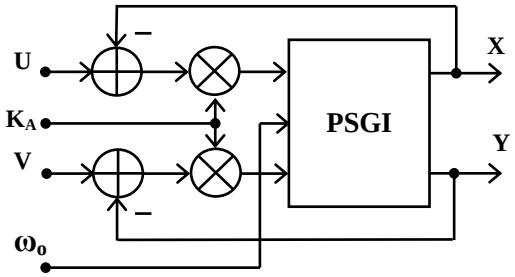


Fig. 3.1 – Positive sequence filter (PSF)

To apply the internal model principle to the case of an unbalanced two-phase αβ signal, it is useful to represent such a signal in terms of it's positive and negative sequence components:

$$\mathbf{v}_{\alpha\beta}(t) = v^{p}_{\alpha\beta}(t) + v^{n}_{\alpha\beta}(t) = V^{p}e^{j\omega t} + V^{n}e^{-j\ \omega t+\varphi}$$

$$= \begin{pmatrix} v^{p}_{\alpha}(t) \\ v^{p}_{\beta}(t) \end{pmatrix} + \begin{pmatrix} v^{n}_{\alpha}(t) \\ v^{n}_{\beta}(t) \end{pmatrix} \tag{3.1}$$

$$= V^{p}_{\alpha\beta}\begin{pmatrix} \cos(\omega t) \\ -\sin(\omega t) \end{pmatrix} + V^{n}_{\alpha\beta}\begin{pmatrix} \cos(\omega t+\varphi) \\ \sin(\omega t+\varphi) \end{pmatrix} \tag{3.2}$$

as depicted in Fig. 3.2. Thus, to estimate the positive and negative sequence components from the physical signals $v_{\alpha}(t)$ and $v_{\beta}(t)$, it is proposed to extend the previous two-phase positive sequence filter by adding a negative sequence generalized integrator (NSGI) within the same feedback loop, in parallel with the positive sequence generalized integrator (PSGI), both driven by the same error signals. The PSGI's and NSGI's respective α and β outputs are to be summed together, to produce the respective estimated α and β signals, which can then be subtracted from the corresponding physical α and β input signals, to produce the error signals to be applied to the two filters, as shown in Fig. 3.3. This parallel combination of the α and β output signals thus mirrors the structure of the source of the unbalanced αβ signal. Fig. 3.3 further emphasizes the option to separately select the K factors for each of the positive and negative sequence filtering components.

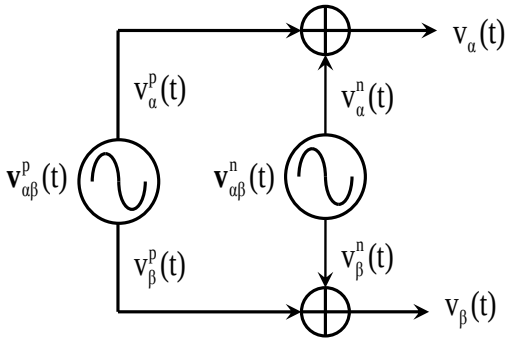


Fig. 3.2 – Unbalanced two-phase signal

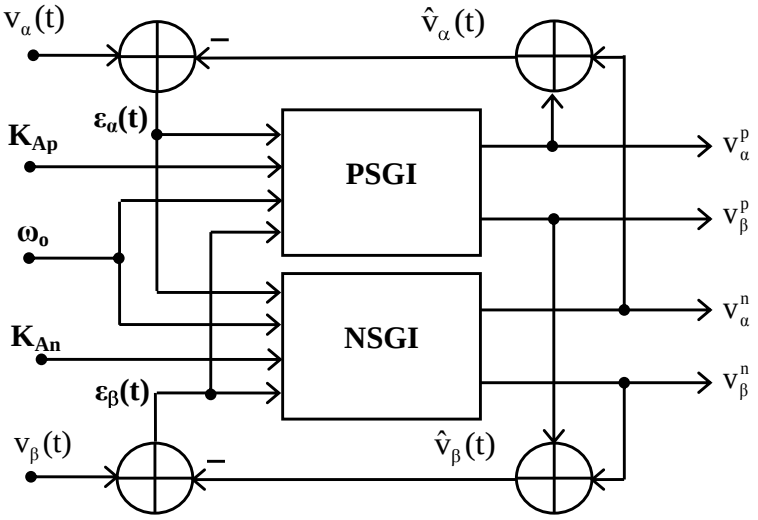


Fig. 3.3 – Positive-Negative sequence filter (PNSF)

A useful mathematical model of the PNSF can be derived from a state space description, as detailed in Fig. 3.4.

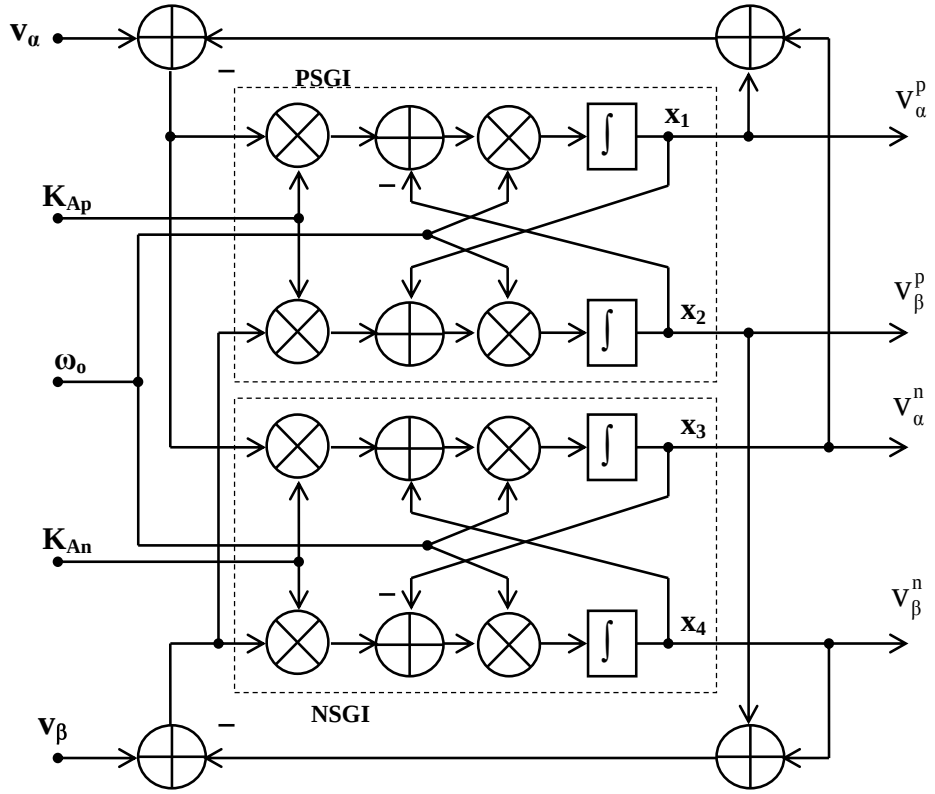


Fig. 3.4 – State space description of the PNSF

In order to simplify subsequent design and to separate bandwidth design from the nominal operating frequency ω_o, this factor ω_o is applied to each integrator. The PNSF can then be described in standard state space form as follows:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{aligned} \tag{3.3}$$

$$\mathbf{u}(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} = \begin{pmatrix} v_\alpha(t) \\ v_\beta(t) \end{pmatrix} \tag{3.4}$$

$$\mathbf{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \\ y_4(t) \end{bmatrix} = \mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} = \begin{bmatrix} \hat{v}_\alpha^p(t) \\ \hat{v}_\beta^p(t) \\ \hat{v}_\alpha^n(t) \\ \hat{v}_\beta^n(t) \end{bmatrix} \tag{3.5}$$

$$A = \omega_O \begin{bmatrix} -K_{Ap} & -1 & -K_{Ap} & 0 \\ 1 & -K_{Ap} & 0 & -K_{Ap} \\ -K_{An} & 0 & -K_n & 1 \\ 0 & -K_{An} & -1 & -K_{An} \end{bmatrix} \qquad B = \omega_O \begin{bmatrix} K_{Ap} & 0 \\ 0 & K_{Ap} \\ K_{An} & 0 \\ 0 & K_{An} \end{bmatrix} \tag{3.6}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

$$D = 0$$

3.1.2 Eigenvalue Characterization of PNSF

Based on the state space model of the proposed PNSF, the eigenvalues, or poles, of the transfer functions from v_α and v_β to each of the four outputs, are determined by (3.8). For positive values of K_p and K_n, the poles can be seen to be stable, as also confirmed in Fig. 3.5. For values of K_p and K_n ranging from 0.1 to 2, normalized to $\omega_o = 1$, these poles follow the loci shown, where the circles identify those locations where $K_p = K_n$. Arrows across lines indicate how the loci change as K_p increases, whereas arrows along lines indicate how K_n increases for any given K_p value. The corresponding time domain response and magnitude transfer functions are discussed next.

$$\begin{aligned}
p_1 &= \omega_o \left(-\frac{K_{Ap} + K_{An}}{2} - \frac{\sqrt{K_{Ap} + K_{An}}^{2} - 4 + i \times 4 \; + K_{Ap} - K_{An}}{2} \right) \\
p_2 &= \omega_o \left(-\frac{K_{Ap} + K_{An}}{2} + \frac{\sqrt{K_{Ap} + K_{An}}^{2} - 4 + i \times 4 \; + K_{Ap} - K_{An}}{2} \right) \\
p_3 &= \omega_o \left(-\frac{K_{Ap} + K_{An}}{2} - \frac{\sqrt{K_{Ap} + K_{An}}^{2} - 4 + i \times 4 \; - K_{Ap} + K_{An}}{2} \right) \\
p_4 &= \omega_o \left(-\frac{K_{Ap} + K_{An}}{2} + \frac{\sqrt{K_{Ap} + K_{An}}^{2} - 4 + i \times 4 \; - K_{Ap} + K_{An}}{2} \right)
\end{aligned} \tag{3.8}$$

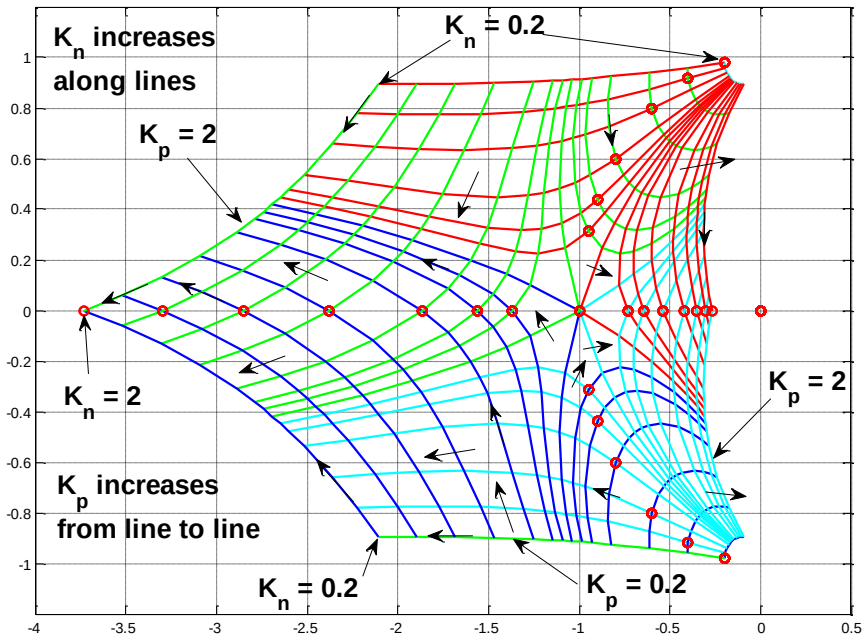


Fig. 3.5 – PNSF pole loci vs. K_p and K_n

3.1.3 PNSF Time Domain Response to Sequence Flip

The sequence selective LTI performance of the PNSF can be readily observed by first applying a balanced input signal with positive sequence, and then switching to a negative sequence, as suggested in Fig. 3.6, where $u(t-t_o)$ indicates a step function at time $t = t_o$.

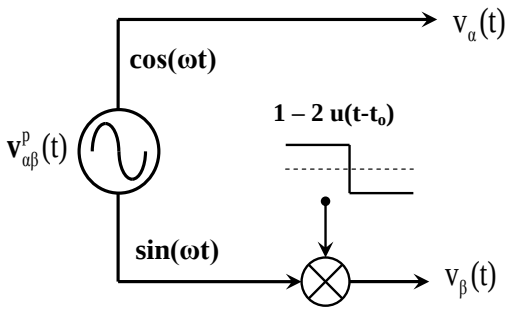


Fig. 3.6 – Change of sequence of balanced αβ signal

Such a signal can be described by:

$$\mathbf{V}_{\alpha\beta} = \begin{pmatrix} \cos(\omega t) \\ \sin(\omega t) \end{pmatrix} \quad \text{for } t < t_O$$
$$= \begin{pmatrix} \cos(\omega t) \\ -\sin(\omega t) \end{pmatrix} \quad \text{for } t \geq t_O \qquad (3.9)$$

An example response of a PNSF to such a signal, with $K_{Ap} = K_{An}$ 0.5 is presented in Fig. 3.7, which shows:

a) an αβ input signal with positive sequence, up to t = 0.03;

b) an αβ input signal with negative sequence, after t = 0.03;

c) the response of just a positive sequence filter (PSF), with K = 0.5, in the top trace, which shows a full response to the positive sequence input, but only partial attenuation of the response to a negative sequence;

d) the positive sequence output of the PNSF, in the middle trace, and magnitude of the PSGI state vector, showing zero steady state response to the negative sequence, after t = 0.03;

e) the negative sequence output of the PNSF, in the bottom trace, and magnitude of the PSGI state vector, showing zero steady state response to the positive sequence, up to t = 0.03.

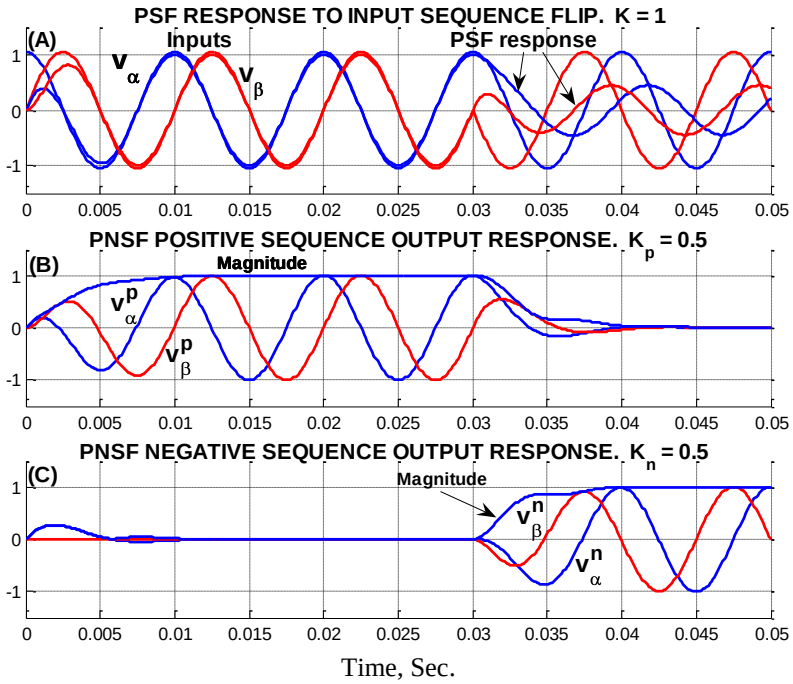


Fig. 3.7 – PNSF time domain response to sequence flip

Thus, the two internal PSGI and NSGI can be seen to correctly and fully reject input signals of opposite sequence, respectively. This will also be evident in the frequency domain magnitude response, discussed in the next section.

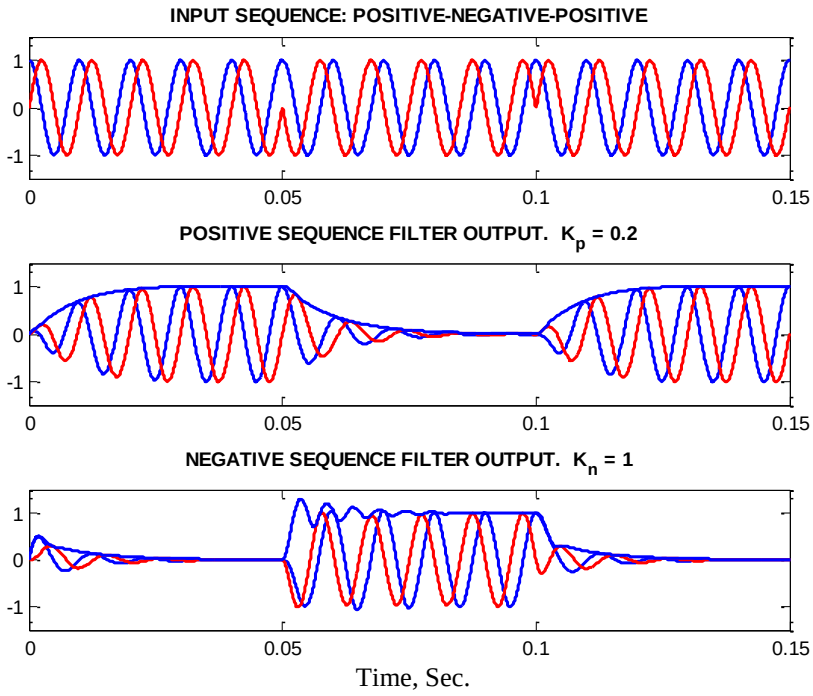


Fig. 3.8 – PNSF response to sequence flip, with different K_{Ap} and K_{An}

Fig. 3.8 presents again the time domain response of a PNSF to an input sequence flip, with different values for the positive and negative sequence filters:

a) the input signal with alternating sequences in the top trace;

b) the positive filter output response in the middle trace, and also the filter state vector magnitude as a useful indicator of response time. With K_p = 0.2, the positive sequence filter has a slower response, and correspondingly narrower bandwidth, as would be desirable for greater distortion attenuation before applying the output signal to a PLL.

c) the negative filter response in the bottom trace, also accompanied by the filter state vector magnitude, also for a useful indication of response time. With K_n = 1, the negative sequence filter has a faster response, allowing

for faster detection of a negative sequence component, as might result from an unbalanced condition, such as due to a line fault.

In conclusion, the proposed PNSF clearly shows how the PSF and NSF sub-filters each completely reject an input signal of the opposite sequence, and the ability for independent user-selected bandwidth and time domain response.

3.1.4 PNSF Magnitude Frequency Response

In the frequency domain, the positive and negative sequences are effectively distinguished by their frequencies, being positive and negative, identified by $s = j\omega$ and $s = -j\omega$, respectively. Fig. 3.9 shows the frequency responses for the positive and negative sequence filter outputs, with different values of K_p (or K_{Ap}) and K_n (or K_{An}). It can be observed that the two internal filter bandwidths can be separately selected, although there is some slight interaction between the two filters, as is evident by the presence of several overlapping but slightly different positive sequence filter responses for a single value of K_{Ap}.

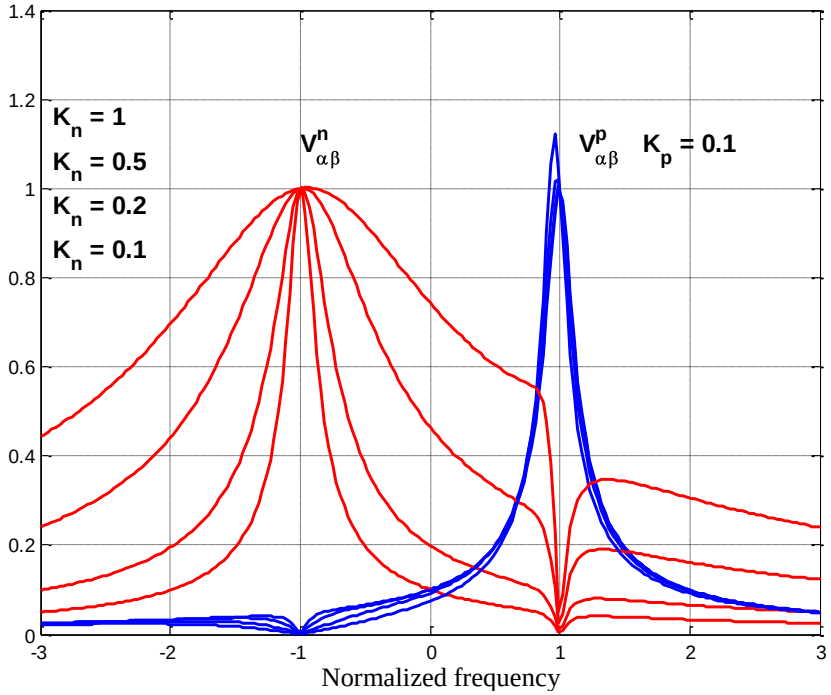


Fig. 3.9 – PNSF magnitude frequency responses

3.2 DISCRETIZATION OF ASAE

As indicated earlier, as the ASAE algorithm simultaneously estimates the amplitudes of the two quadrature components of a sinusoidal input signal, it can be seen to effectively operate as two independent, un-coupled, first order dynamic processes. As will be explained in this section, this proves to be very beneficial from the perspective of discrete implementation on a DSP in real time, especially for narrower bandwidths, as required for greater distortion attenuation, and therefore a very desirable aspect of the overall PLL implementation. To establish this benefit, this section first discusses the idealized band-pass filter for filtering and estimating a distorted sinusoid, and presents it's major drawback when discretized for digital implementation, which will serve to more clearly reveal the benefits of the ASAE algorithm.

3.2.1 Discretization of Continuous Time Resonator

Given the basic two-integrator resonator at the core of the continuous time model for the band-pass filter, it's simplest and computationally most economical discretization consists of simply replacing each integrator with a first order Euler type discrete approximation:

$$y(nT_S) = y((n-1)T_S) + \int_{(n-1)T_S}^{nT_S} v(t)dt \approx y((n-1)T_S) + v((n-1)T_S) \times T_S$$
$$y((n+1)T_S) \approx y(nT_S) + v(nT_S) \times T_S \qquad (3.10)$$

The second expression indicates that the current value of the input signal $v(nT_s)$ will serve to compute the next value of the output $y((n+1)T_s)$, which is a more realistic realization of the discrete implementation. This discretized version of the above integrator can be represented as in Fig. 3.10, where Z^{-1} denotes the Z transform of the single-sample delay operator, and the discretized version of the two-integrator resonator becomes as shown in Fig. 3.11. Fig. 3.11 also emphasizes the separation of the sampling time T_s and the nominal operating frequency ω_o, which serves to set the "time scale" for each integrator, and subsequently of any dynamic system, as a whole.

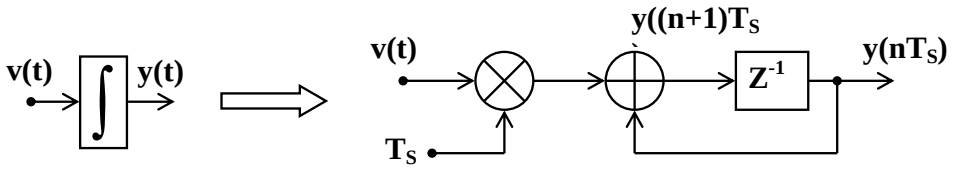


Fig. 3.10 – Euler discretization of integrator

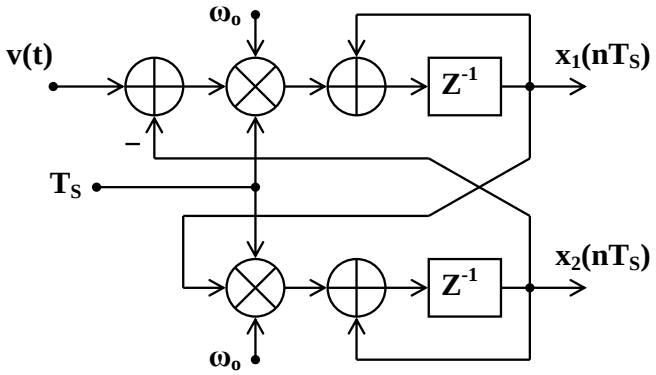


Fig. 3.11 – Discretized resonator

In state variable form, the system of Fig. 3.11 can be expressed as:

$$\begin{pmatrix} x_1(n+1) \\ x_2(n+1) \end{pmatrix} = \begin{pmatrix} 1 & -\omega_o T_S \\ \omega_o T_S & 1 \end{pmatrix} \begin{pmatrix} x_1(n) \\ x_2(n) \end{pmatrix} + \begin{pmatrix} \omega_o T_S \\ 0 \end{pmatrix} v(n) \tag{3.11}$$

with characteristic equation:

$$\det\left(ZI - \begin{pmatrix} 1 & -\omega_o T_S \\ \omega_o T_S & 1 \end{pmatrix} \right) = \det \begin{pmatrix} Z-1 & \omega_o T_S \\ -\omega_o T_S & Z-1 \end{pmatrix}$$

$$= Z-1^{\,2} + \omega_o T_S^{\,2}$$

$$= Z^2 - 2Z + 1 + \omega_o^2 T_S^2 \tag{3.12}$$

and eigenvalues:

$$Z_{1,2} = 1 \pm j\omega_o T_s$$

The eigenvalues for this discrete system can be seen to have a magnitude greater than one, such that this simplest discretization of a resonator is intrinsically unstable. Thus, it requires a minimum, frequency dependent, positive factor K to ensure stability, such as shown in the discretized band-pass filter, in Fig. 3.12.

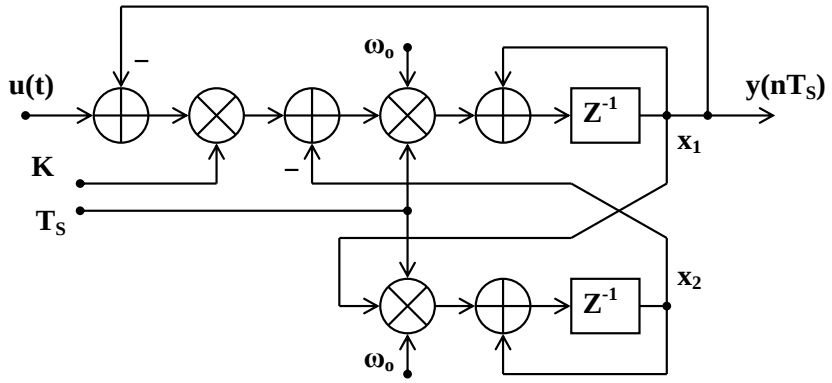


Fig. 3.12 – Discretized band-pass filter

The above can be described by the following state space realization:

$$\begin{pmatrix} x_1(n+1) \\ x_2(n+1) \end{pmatrix} = \begin{pmatrix} 1-K\omega_o T_S & -\omega_o T_S \\ \omega_o T_S & 1 \end{pmatrix} \begin{pmatrix} x_1(n) \\ x_2(n) \end{pmatrix} + \begin{pmatrix} K\omega_o T_S \\ 0 \end{pmatrix} u(n) \tag{3.13}$$

with the following eigenvalues:

$$Z_{1,2} = \text{eig}\begin{pmatrix} 1-K\omega_o T_S & -\omega_o T_S \\ \omega_o T_S & 1 \end{pmatrix} = \left(1 - \frac{K\omega_o T_S}{2} \pm \frac{\omega_o T_S \sqrt{K^2-4}}{2} \right) \tag{3.14}$$

with magnitude:

$$\left| Z_{1,2} \right| = 1 - K\omega_o T_S + \omega_o^2 T_S^2 \tag{3.15}$$

Thus, to ensure stability, this requires a minimum value of:

$K > \omega_o T_s$

which prevents the possibility of arbitrarily small filter bandwidths, as would be required for higher rate of harmonic attenuation.

3.2.2 Discretization of the ASAE Algorithm

In contrast to the discretized resonator, as indicated earlier, for the discretized version of the ASAE algorithm, implemented as shown in Fig. 3.13, it can usefully be treated and analyzed as two independent first order dynamic systems, one for each of the quadrature components of a sinusoidal signal at ω_o, as in Fig. 3.14. Thus, for example, the dynamic system for the estimation of the amplitude

$\hat{A}_C$ of the cosine component of $u(nT_s)$, represented by state $x_1(n)$, can be described by (3.16).

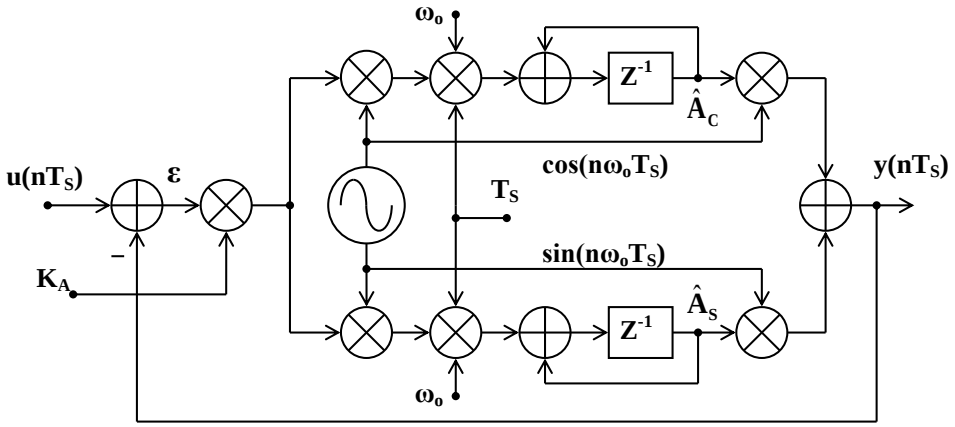


Fig. 3.13 – Discretized ASAE

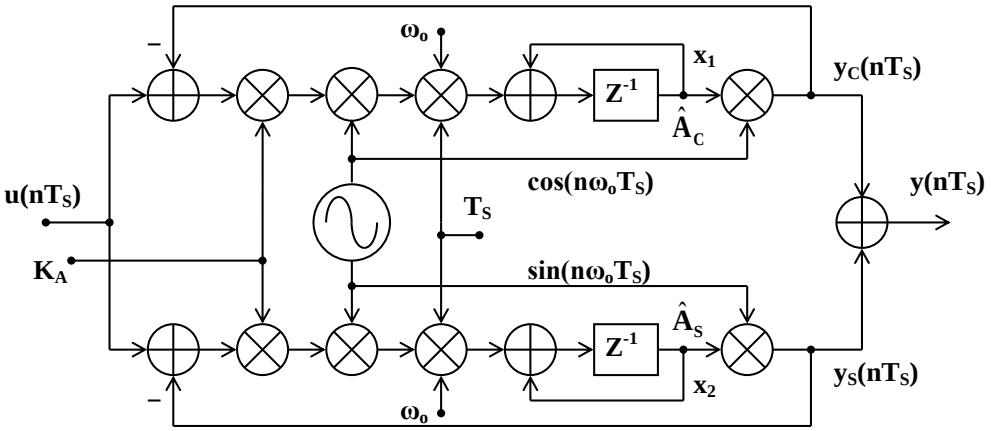


Fig. 3.14 – Two independent first-order processes of discretized ASAE

$$\begin{aligned} x_1(n+1) &= x_1(n) + K_A\omega_o T_S \times \cos(n\omega_o T_S) \times \; u(n) - x_1(n) \times \cos(n\omega_o T_S) \\ &= x_1(n) \times \; 1 - K_A\omega_o T_S \times \cos^2(n\omega_o T_S) \\ &\quad + K_A\omega_o T_S \times \cos(n\omega_o T_S) \times u(n) \end{aligned} \qquad 3.16)$$

The stability of the dynamic system, with $u(nT_s) = 0$, can be characterized by the feedback factor r, from x(n) to x(n+1), which would normally be referred to as the eigenvalue in the case of a constant value, and which is seen to be:

$$r = 1 - K_A\omega_o T_s \times \cos^2(n\omega_o T_s) \qquad (3.17)$$

For a discrete system, r can be positive or negative, but it's magnitude must be less than 1, in which case, the sequence x(n), with zero input, would progressively converge to zero.

3.2.3 Comparison of Discretized BPF and ASAE Algorithms

To appreciate the benefit of the ASAE algorithm over the BPF, it is necessary to emphasize some of the more pertinent aspects and constraints that arise in a typical realization:

a) greater savings in development time of the various algorithms can be achieved with graphical design tools, such as The Mathworks' Simulink, which makes possible block diagram design, for rapid development and simulation of solutions to be explored, and which motivates the usage of such tools.

b) initial exploration and validation of any proposed algorithm typically entails simulation with the highest order differential equation solver, such as Simulink's ode-5, which can produce near idealized results, and which can serve to initially validate a proposed principle under study. For example, a two-integrator resonator, resonating at a frequency of 100 Hz, with initial state $(x_1, x_2) = (1, 0)$, simulated with a step time of 100 μSec. over an interval of 10 sec., can be seen to maintain a constant amplitude to the end of this simulation interval. In contrast, as discussed earlier, using ode-1 solver, which uses a simple Euler first order discrete integrator to discretize the continuous time integrator in the original model, is intrinsically unstable and therefore fundamentally mis-represents the theoretically expected results in continuous time. However, the computational demands of the ode-5 solver are significantly greater than for the ode-1 solver, and this therefore motivates the usage of a suitable alternative equivalent model.

c) an actual discrete implementation in real time is inherently constrained by the computational resources of the target processor, and efficiency of the chosen compiler. For Simulink models, The Mathworks provides the Real

Time Workshop (RTW) tool to compile the models into standard C code, which is subsequently further compiled for the target platform, such as dSPACE's DS1104, adopted for this PLL project. Therefore, if possible, it is desirable to implement the algorithm with the simplest ode solver, and the ode-1 solver serves as a useful criterion by which to assess the overall value of the proposed solution and real time implementation.

The proposed ASAE algorithm is compared with a band-pass filter, by simulation with Simulink, first with ode-5 solver to establish idealized performance for reference, and then with the ode-1 solver; this is further validated with experimental results. An example is presented in Fig. 3.15.

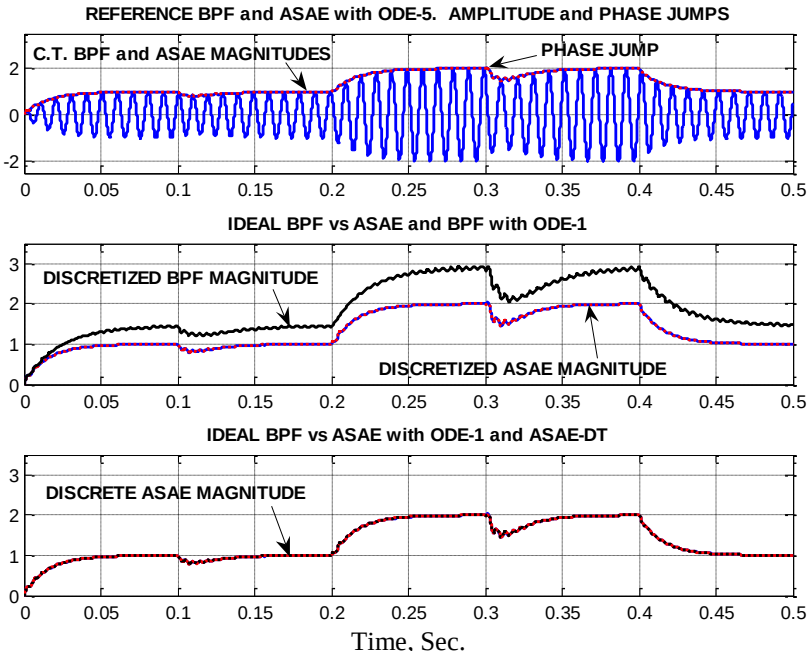


Fig. 3.15 – BPF and ASAE performances, idealized vs. discretized

The two algorithms, tuned to a center frequency ω_o = 100 Hz, using K = 0.2, are subjected to an amplitude and phase step modulated sinusoid at 100 Hz:

a) simulation with ode-5 serves to establish idealized continuous time performance for reference to compare with, and shows the consistency of performance between the two, as expected from their equivalency;

b) the top trace shows the filtered output and it's amplitude, which will better serve to subsequently compare the performances of the two algorithms;

c) simulation with ode-1 clearly shows the consistency of the ASAE algorithm with idealized results obtained with ode-5;

d) simulation with ode-1 also clearly shows how the resonator-based BPF performs significantly differently, resulting in a larger amplitude, as expected, since it's un-damped eigenvalues are unstable. In effect, the discretized BPF performance deviates too much from the idealized case to be acceptable.

e) the bottom trace shows the performance of the idealized BPF and of a discrete ASAE algorithm, as opposed to using continuous time integrators which are discretized by Simulink. The discrete ASAE can be seen to produce an identical result to the idealized BPF, making it clearly best suited for real time implementation.

Fig. 3.16 shows the response of the discrete ASAE algorithm to a step in amplitude of the input signal, and compares it to the idealized BPF, for different values of K, from 0.1 to 0.9. Again, the ASAE can be seen to perform as well as the idealized BPF.

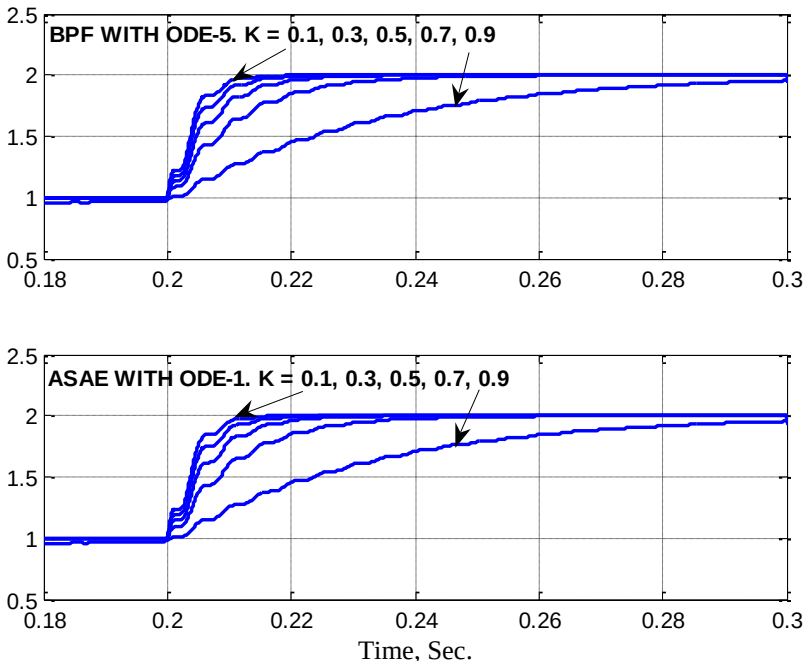


Fig. 3.16 – Idealized BPF and discrete ASAE with ode-1, with various K

3.2.4 Discrete Integration by Bilinear Transformation

Whereas the Euler discretization treats the integrated variable as constant over a sampling interval, a trapezoidal approximation, [84], produces a better approximation of the integration. However, coupling two such integrators, as in Fig. 3.11, results in a mutual computational dependence at any instant n. Such a mutual dependence is referred to, for example, as an algebraic loop in a Matlab/Simulink model, which typically requires some re-structuring of the model, such as by introducing a memory delay stage in the loop. However, such an addition is precisely to be avoided, in order to keep a minimal dynamical order.

In conclusion, the proposed discrete ASAE algorithm shows a performance far superior to the discretized BPF, and specifically with the least computationally intensive implementation of the integrator operations, making it very attractive for real time implementation on a DSP.

3.3 ASAE-BASED QUADRATURE SIGNAL GENERATOR

Building on the ASAE algorithm, this section presents an extension to generate a quadrature output signal, referred to as ASAE-QG, shown in Fig. 3.17.

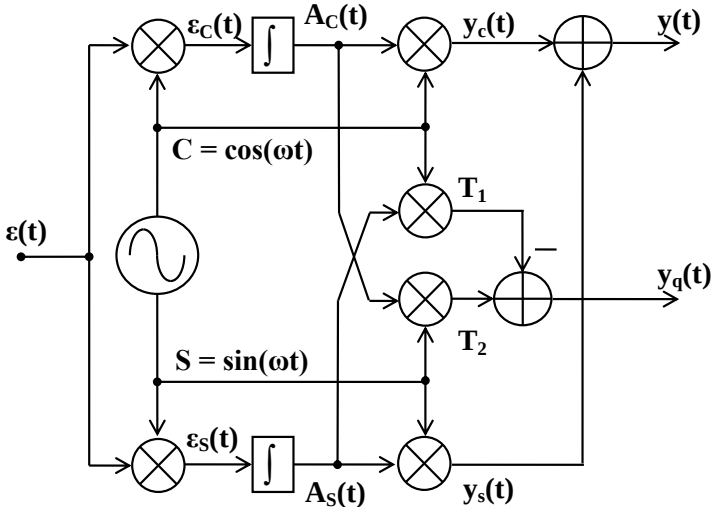


Fig. 3.17 – ASAE-based quadrature signal generator

Thus, given the earlier output signal from the resonator equivalent system:

$$y(t) = A_c(t) \times \cos(\omega_o t) + A_s(t) \times \sin(\omega_o t)$$

a quadrature version can be produced according to:

$$y_q(t) = A_c(t) \times \sin(\omega_o t) - A_s(t) \times \cos(\omega_o t)$$

Following a similar development as presented earlier, and building on it, the transfer function from E(s) to $Y_q(s)$ can be shown to be the same as for a LTI resonator:

IN TIME DOMAIN	**IN LAPLACE S DOMAIN**
$A_c(t) = \int e_c(t)dt$	$A_C(s) = \frac{E_C(s)}{s} = \frac{E(s - j\omega) + E(s + j\omega)}{2s}$
$A_s(t) = \int e_s(t)dt$	$A_S(s) = \frac{E_S(s)}{s} = \frac{E(s - j\omega) - E(s + j\omega)}{2js}$
$T_1(t) = A_s(t) \times \cos(\omega t)$ $= A_s(t)\,(e^{j\omega t}+e^{-j\omega t})/2$	$T_1(s) = \frac{A_S(s - j\omega) + A_S(s + j\omega)}{2}$ $= \frac{1}{2}\left[\frac{E(s - 2j\omega) - E(s)}{2j(s - j\omega)} + \frac{E(s) - E(s + 2j\omega)}{2j(s + j\omega)}\right]$ $= \frac{E(s)\ (s - j\omega) - (s + j\omega)}{4j(s - j\omega)(s + j\omega)} + \frac{E(s - 2j\omega)}{4j(s - j\omega)} - \frac{E(s + 2j\omega)}{4j(s + j\omega)}$ $= \frac{-2j\omega E(s)}{4j(s^2 + \omega^2)} + \frac{E(s - 2j\omega)}{4j(s - j\omega)} - \frac{E(s + 2j\omega)}{4j(s + j\omega)}$ $= \frac{-\omega E(s)}{2(s^2 + \omega^2)} + \frac{E(s - 2j\omega)}{4j(s - j\omega)} - \frac{E(s + 2j\omega)}{4j(s + j\omega)}$
$T_2(t) = A_c(t) \times \sin(\omega t)$ $= A_c(t)\,(e^{j\omega t}-e^{-j\omega t})/2j$	$T_2(s) = \frac{A_C(s - j\omega) - A_C(s + j\omega)}{2j}$ $= \frac{1}{2j}\left[\frac{E(s - 2j\omega) + E(s)}{2(s - j\omega)} - \frac{E(s) + E(s + 2j\omega)}{2(s + j\omega)}\right]$ $= \frac{E(s - 2j\omega) + E(s)}{4j(s - j\omega)} - \frac{E(s) + E(s + 2j\omega)}{4j(s + j\omega)}$ $= \frac{E(s)}{4j(s - j\omega)} - \frac{E(s)}{4j(s + j\omega)} + \frac{E(s - 2j\omega)}{4j(s - j\omega)} - \frac{E(s + 2j\omega)}{4j(s + j\omega)}$

$$= \frac{E(s)\ (s + j\omega) - (s - j\omega)}{4j(s - j\omega)(s + j\omega)} + \frac{E(s - 2j\omega)}{4j(s - j\omega)} - \frac{E(s + 2j\omega)}{4j(s + j\omega)}$$

$$= \frac{\omega E(s)}{2(s^2 + \omega^2)} + \frac{E(s - 2j\omega)}{4j(s - j\omega)} - \frac{E(s + 2j\omega)}{4j(s + j\omega)}$$

$y_q(t) = T_2(t) - T_1(t)$ $\qquad Y(s) = T_2(s) - T_1(s) = \frac{\omega}{(s^2 + \omega^2)} \times E(s)$

In conclusion, this extension to the ASAE-based band-pass filter can be used, equivalently to the band-pass filter, to generate quadrature signals for subsequent processing by a conventional 2-phase PLL.

3.4 QUADRATURE SIGNAL ASAE WITH KNOWN PHASE

The present section proposes an extension to the first ASAE algorithm for a single-phase input signal with known phase, to estimate the amplitude of a balanced quadrature signal with known phase, described by:

$$\mathbf{v}_{\alpha\beta}(t) = \begin{pmatrix} v_\alpha(t) \\ v_\beta(t) \end{pmatrix} = A \begin{pmatrix} \cos(\omega t) \\ \sin(\omega t) \end{pmatrix}$$

Thus, the proposed ASAE algorithm aims to generate an estimate of the above, by indirectly estimating the common amplitude of the two quadrature components:

$$\hat{\mathbf{v}}_{\alpha\beta}(t) = \begin{pmatrix} \hat{v}_\alpha(t) \\ \hat{v}_\beta(t) \end{pmatrix} = \hat{A}(t) \begin{pmatrix} \cos(\omega t) \\ \sin(\omega t) \end{pmatrix}$$

and then feed back these estimates for comparison with the actual input signals, to produce the estimate error signals ε_α and ε_β, as shown in Fig. 3.18. The resulting dynamic system is thus of first order, since it entails only one dynamic variable, and can be described as in (3.18), where $p_\alpha(t)$ and $p_\beta(t)$ are the intermediate products indicated in Fig. 3.18. Each of these products can be seen to behave in a manner very similar to the ASAE process with known phase, as described in (3.19).

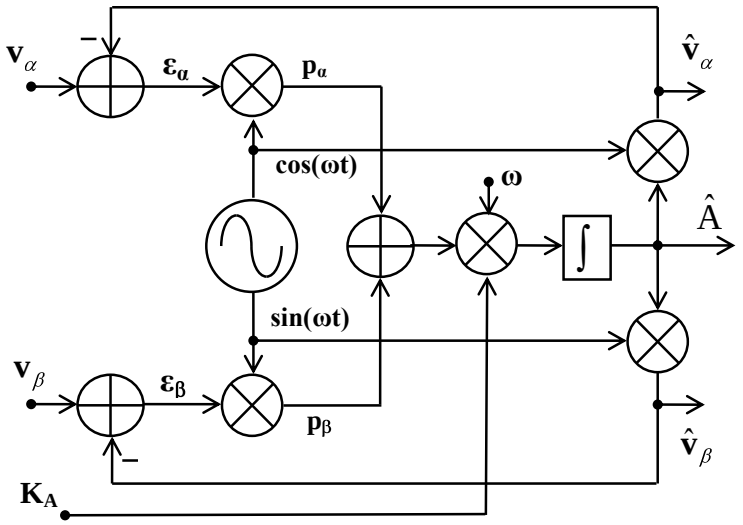


Fig. 3.18 – Quadrature Adaptive Synchronous Amplitude Estimation

$$\dot{\hat{A}}(t) = K_A \times \omega \times (p_\alpha(t) + p_\beta(t)) \tag{3.18}$$

$$\begin{aligned} p_\alpha(t) &= A\cos(\omega t) - \hat{A}(t) \times \cos(\omega t) \times \cos(\omega t) \\ &= A - \hat{A}(t) \times \cos^2(\omega t) \end{aligned} \tag{3.19}$$

Similarly:

$$p_\beta(t) = A - \hat{A}(t) \times \sin^2(\omega t)$$

The overall dynamic process for the estimation of the common amplitude can thus be described by the first order dynamic system in (3.20), as shown in Fig. 3.19.

$$\begin{aligned} \dot{\hat{A}}(t) &= K_A \omega \times A - \hat{A}(t) \times \cos^2(\omega t) + \sin^2(\omega t) \\ &= K_A \omega \times A - \hat{A}(t) \end{aligned} \tag{3.20}$$

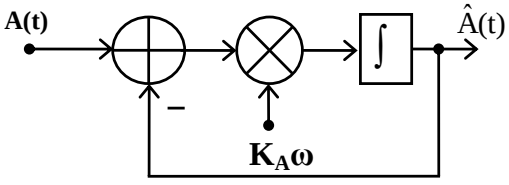


Fig. 3.19 – First order dynamic equivalent of Quadrature ASAE

In contrast to the single-phase ASAE with known phase, this quadrature ASAE process, or QASAE, with effectively twice the signal energy, has half the estimation time constant, and is also free of any ripple, due to the complementarity of the $\sin^2(\omega t)$ and $\cos^2(\omega t)$ components.

3.5 ASAE-BASED POSITIVE SEQUENCE FILTER

Based on the variations of the ASAE algorithm developed thus far, the principle can be further extended to a quadrature signal of unknown phase, as shown in Fig. 3.20, which also emphasizes a distinction between an inner ASAE-based positive sequence generalized integrator (PSGI) equivalent, and an outer feedback loop, which will serve a further subsequent development. This structure, referred to as ASAE-PSF, is thus equivalent to a resonator-based positive sequence filter. Extending the previous equations, this ASAE-PSF is governed by the following equations for the estimated amplitude state variables $\hat{A}_I$ and $\hat{A}_Q$:

$$\hat{v}_\alpha(t) = \hat{A}_I(t) \times \cos(\omega t) + \hat{A}_Q(t) \times \sin(\omega t) \tag{3.21a}$$

$$\hat{v}_\beta(t) = \hat{A}_I(t) \times \sin(\omega t) - \hat{A}_Q(t) \times \cos(\omega t) \tag{3.21b}$$

$$\dot{\hat{A}}_I(t) = K_A \omega \times \cos(\omega t) \times \left[v_\alpha(t) - \hat{v}_\alpha(t) \right] + \sin(\omega t) \times \left[v_\beta(t) - \hat{v}_\beta(t) \right] \tag{3.21c}$$

$$\dot{\hat{A}}_Q(t) = K_A \omega \times \sin(\omega t) \times \left[v_\alpha(t) - \hat{v}_\alpha(t) \right] - \cos(\omega t) \times \left[v_\beta(t) - \hat{v}_\beta(t) \right] \tag{3.21d}$$

This dynamic system is essentially the same as before, but also reflects the fact that there is now effectively twice the energy about the same phase information, available from two quadrature components, of equal magnitude, but 90^o apart. The I and Q subscripts serve to generalize from the previous C and S subscripts. In contrast to the single-phase ASAE, which estimates the cos and sin components of input sinusoid, which are nominally in-phase and in quadrature with the input signal, Fig. 3.21 emphasizes that the ASAE-PSF estimates the corresponding in-phase (I) and quadrature (Q) components of each component of the input quadrature signal $V_{\alpha\beta}$.

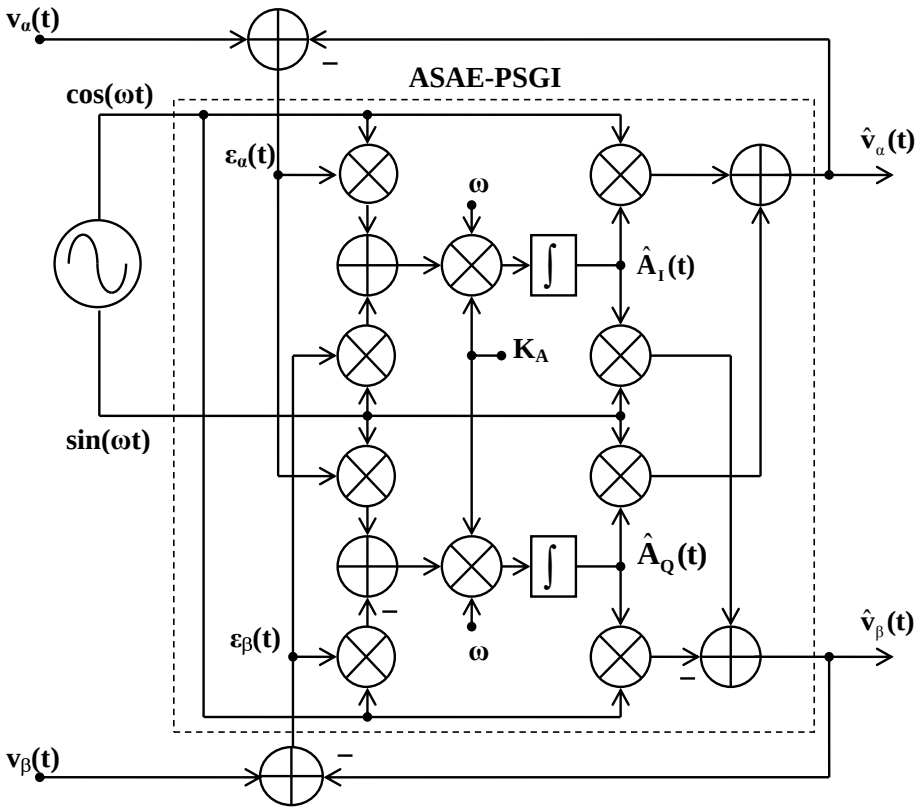


Fig. 3.20 – ASAE-based Positive Sequence Filter

Fig. 3.21 also reveals the underlying mechanism expressed in (3.21):

a) (3.21c and d) express how the estimates are defined from the amplitude estimates of the in-phase and quadrature components

b) (3.21a and b) express how the amplitude estimates are adjusted by correlating, i.e. multiplying, the estimate errors ε_α and ε_β by the cos and sin components, and adding appropriately.

The response of the ASAE-PSF to a quadrature input signal with various phase and amplitude steps is exemplified in Fig. 3.22, which shows:

a) phase steps from an initial value of $+\pi/3$ to $-\pi/3$, and by $+\pi/2$ at t = 0.05 and 0.15 sec., respectively, and an amplitude step of +100% at t = 0.1 sec.;

b) the top trace shows just the α component of the input and output response signals, and the internal state vector magnitude, which displays a transient droop at phase jumps;

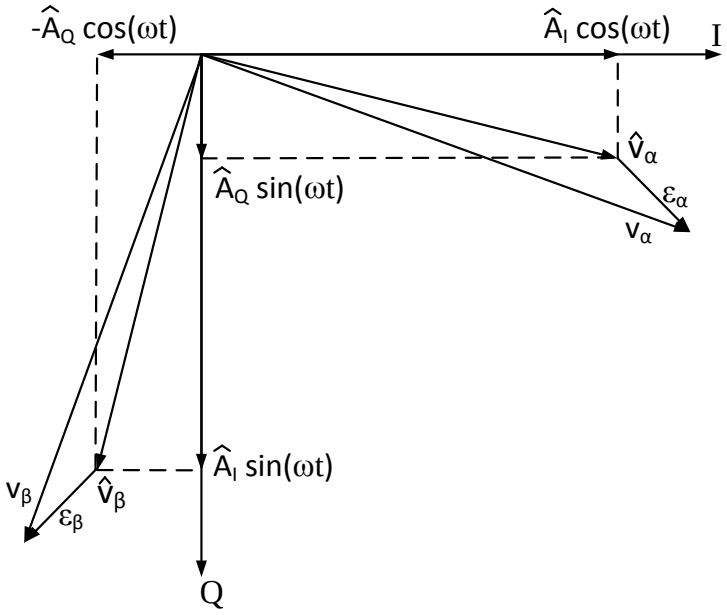


Fig. 3.21 – ASAE-PSF phasor diagram

c) the bottom trace shows the quadrature components of the output α signal and their respective magnitudes.

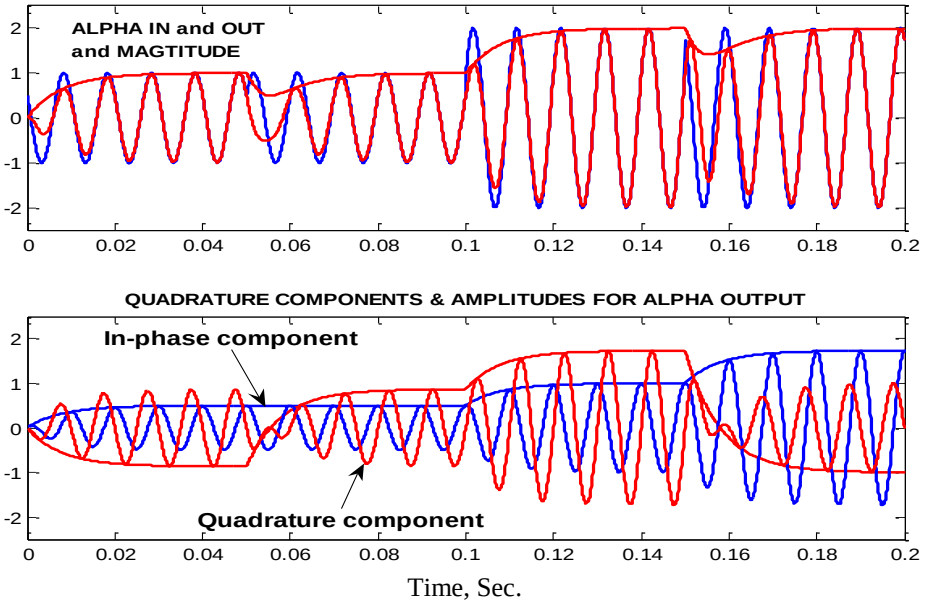


Fig. 3.22 – ASAE-PSF response to amplitude and phase steps; K = 0.2

The exponential responses of the quadrature components' magnitudes, free of any ripple, and without any transient oscillations, reveal a robust performance of the

ASAE-based positive sequence filter. A comparison of this device's α and β output signals, as well as the state vector magnitude with those of a resonator based PSF, with first order Euler discretized integrators, can be shown to exhibit the same inadequacies as discussed earlier, and again confirms the merits of the ASAE-based realization of the PSF function.

3.6 ASAE-BASED POSITIVE-NEGATIVE SEQUENCE FILTER

Having established the equivalence of the ASAE algorithm for single-phase and quadrature-phase signals to the resonator based BPF and PSF, respectively, two ASAE-based generalized integrators, a ASAE-PSGI and a ASAE-NSGI can be combined within a negative feedback structure in a manner similar to Fig. 3.3, to handle positive and negative sequences, respectively, to implement a ASAE-PNSF, as in Fig. 3.23. Referring to Fig. 3.20 and equations 3.21, the dynamics of the ASAE-PNSF can similarly be described in terms of four dynamic state variables for the magnitudes of the in-phase and quadrature components for both the positive and negative sequences, $\hat{A}_{Ip}$, $\hat{A}_{Qp}$, $\hat{A}_{In}$, and $\hat{A}_{Qn}$. The α and β signal estimates, and estimate errors can be expressed as:

$$\boldsymbol{\varepsilon}_{\alpha\beta}(t) = \mathbf{v}_{\alpha\beta}(t) - \hat{\mathbf{v}}_{\alpha\beta}(t) = \begin{pmatrix} \varepsilon_{\alpha}(t) \\ \varepsilon_{\beta}(t) \end{pmatrix} = \begin{pmatrix} v_{\alpha}(t) - \hat{v}_{\alpha}(t) \\ v_{\beta}(t) - \hat{v}_{\beta}(t) \end{pmatrix} \tag{3.22a}$$

$$\hat{\mathbf{v}}_{\alpha\beta}(t) = \begin{pmatrix} \hat{v}_{\alpha}(t) \\ \hat{v}_{\beta}(t) \end{pmatrix} = \hat{\mathbf{v}}_{\alpha\beta p}(t) + \hat{\mathbf{v}}_{\alpha\beta n}(t) = \begin{pmatrix} \hat{v}_{\alpha p}(t) + \hat{v}_{\alpha n}(t) \\ \hat{v}_{\beta p}(t) + \hat{v}_{\beta n}(t) \end{pmatrix} \tag{3.22b}$$

$$\hat{\mathbf{v}}_{\alpha\beta p}(t) = \begin{pmatrix} \hat{v}_{\alpha p}(t) \\ \hat{v}_{\beta p}(t) \end{pmatrix} = \begin{pmatrix} \hat{A}_{Ip}(t) \times \cos(\omega t) + \hat{A}_{Qp}(t) \times \sin(\omega t) \\ \hat{A}_{Ip}(t) \times \sin(\omega t) - \hat{A}_{Qp}(t) \times \cos(\omega t) \end{pmatrix}$$

$$= \begin{pmatrix} \cos(\omega t) & \sin(\omega t) \\ \sin(\omega t) & -\cos(\omega t) \end{pmatrix} \begin{pmatrix} \hat{A}_{Ip}(t) \\ \hat{A}_{Qp}(t) \end{pmatrix} \tag{3.22c}$$

$$\hat{\mathbf{v}}_{\alpha\beta n}(t) = \begin{pmatrix} \hat{v}_{\alpha n}(t) \\ \hat{v}_{\beta n}(t) \end{pmatrix} = \begin{pmatrix} \cos(\omega t) & \sin(\omega t) \\ -\sin(\omega t) & \cos(\omega t) \end{pmatrix} \begin{pmatrix} \hat{A}_{In}(t) \\ \hat{A}_{Qn}(t) \end{pmatrix} \tag{3.22d}$$

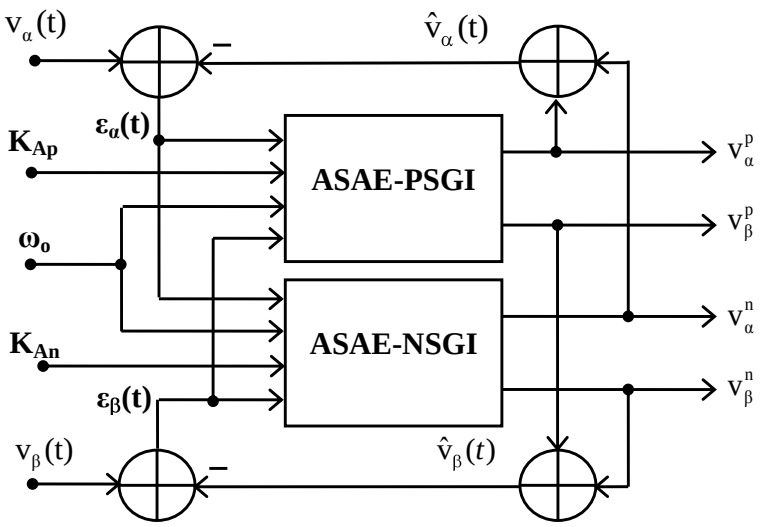


Fig. 3.23 – ASAE-based positive-negative sequence filter

The resulting state derivative equations are:

$$\dot{\hat{A}}_{Ip}(t) = K_{Ap}\omega_o \times \varepsilon_\alpha \times \cos(\omega t) + \varepsilon_\beta \times \sin(\omega t) \tag{3.22e}$$

$$\dot{\hat{A}}_{Qp}(t) = K_{Ap}\omega_o \times \varepsilon_\alpha \times \sin(\omega t) - \varepsilon_\beta \times \cos(\omega t) \tag{3.22f}$$

$$\dot{\hat{A}}_{In}(t) = K_{Ap}\omega_o \times \varepsilon_\alpha \times \cos(\omega t) - \varepsilon_\beta \times \sin(\omega t) \tag{3.22g}$$

$$\dot{\hat{A}}_{Qn}(t) = K_{Ap}\omega_o \times \varepsilon_\alpha \times \sin(\omega t) + \varepsilon_\beta \times \cos(\omega t) \tag{3.22f}$$

The performance of the ASAE-PNSF is exemplified in Fig. 3.24:

a) the top trace displays a two-phase signal, initially balanced, up to t = 0.05, and subsequently unbalanced by a negative sequence component with a magnitude of 0.5, and with phase jumps of $-\pi/3$ and $+\pi/2$ at t = 0.08 and t = 0.11 sec., respectively; the positive sequence amplitude steps up by 50% at t = 0.15;

b) the ASAE-PNSF is operated with $K_{Ap} = 0.2$ and $K_n = 1$;

c) the second trace (B) displays the positive sequence output and magnitude; a slow response can is observed, due to a low K_{Ap} value;

d) the third trace (C) displays the negative sequence β component at the source and output, and magnitude; the source and output signals are barely distinguishable, due to the low K_n value and fast response;

e) the negative sequence output magnitude shows a response time of approximately ¼ cycle, after the negative sequence appears at t = 0.05; slight transients can be observed after phase steps at t = 0.08 and t = 0.11; an oscillatory transient is observed also when the positive sequence component amplitude jumps, at t = 0.15 Sec., which reflects the internal coupling between the two sub-filters, as well as the wider bandwidth of the negative sequence sub-filter;

f) the bottom trace (D) displays the quadrature components of just the α component of the negative sequence output, and their respective magnitudes.

The ASAE-PNSF can be seen to clearly extract the positive and negative sequence components of an unbalanced two-phase input, and to allow useful and independent adjustment of the respective detection response times.

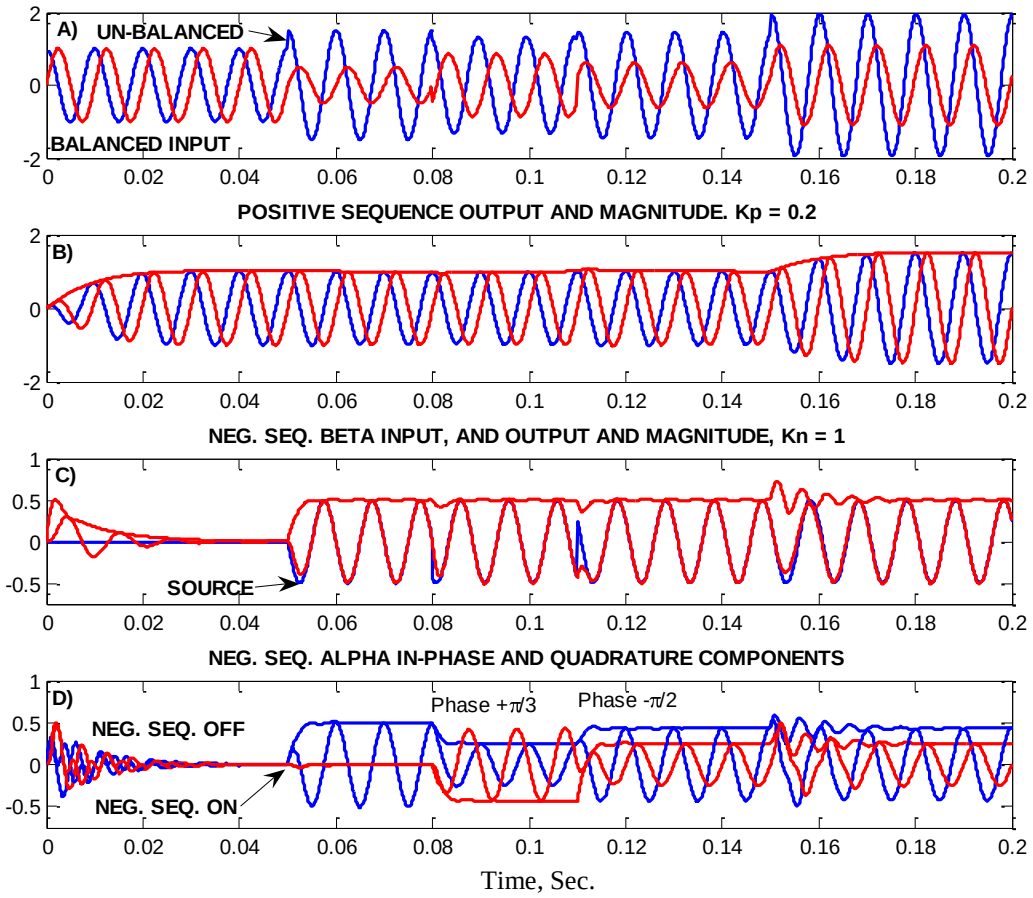


Fig 3.24 – ASAE-PNSF response to negative sequence input

3.7 QASAE-BASED POSITIVE SEQUENCE FILTER PLL

Whereas the ASAE-based filters discussed so far assumed a known and fixed input signal frequency, the present and subsequent PLL systems aim to further track an input signal's varying frequency. For this purpose, they include a phase detection mechanism, based on the commonly used Park transformation, to measure phase discrepancy between the input sinusoidal signal and the generated cos and sin signals, and control the tracking frequency accordingly, for the proper operation of the PNSF and ASAE sub-components. Thus, building on the QASAE from Fig. 3.18, the PLL-based frequency-tracking positive sequence filter is proposed, shown in Fig. 3.25, and is referred to as PSF-PLL.

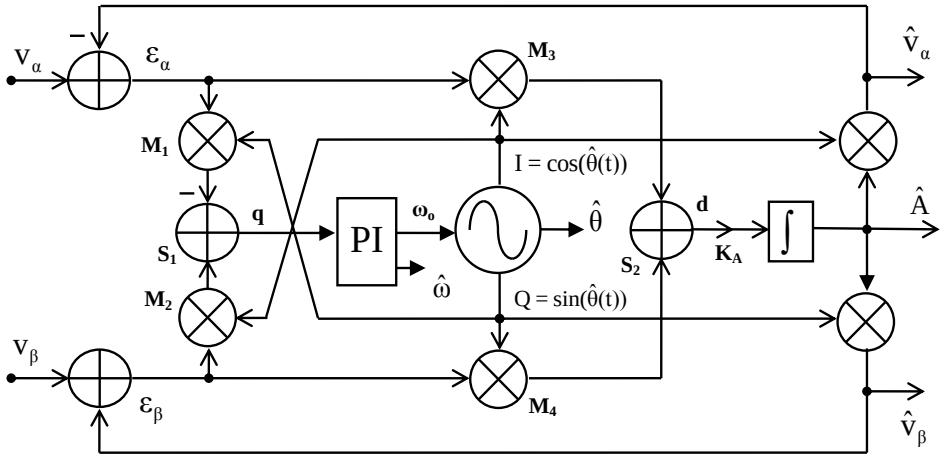


Fig. 3.25 – PLL-based frequency-tracking positive sequence filter

The PSF-PLL operates as follows:

a) phase detection is implemented by multipliers M_1 and M_2 and summer S_1, which generates the Park transformation's q(t) output signal, based on the input error signals ε_α and ε_β and generated I and Q signals;

b) this q(t) signal is applied to a regular PI controller, which produces (i) a frequency estimate signal ω_o to control the signal generator frequency, and (ii) a low-pass filtered frequency estimate $\hat{\omega}$, which is simply taken from the PI controller's integrator output, and is thus a smoother and better

quality estimate of the signal frequency, with higher frequency components attenuated relative to the ω_o signal;

c) multipliers M_3 and M_4 and summer S_2 produce the Park transformation's d(t) output signal. It is important to note, however, that due to the negative feedback of the estimated α and β signals, under sinusoidal and balanced conditions, the steady state error vector $(\varepsilon_\alpha(t), \varepsilon_\beta(t))^T$ converges to zero, which will also drive d(t) to zero, in contrast to the operation of the conventional two-phase PLL. Thus, in the PSF-PLL, as d(t) converges to zero, the estimated amplitude $\hat{A}(t)$ output converges to the actual positive sequence amplitude.

From the perspective of implementation, multipliers M_1 to M_4, together with summers S_1 and S_2 in fact constitute a Park transformation, such that the PSGI equivalent can be seen in terms of the familiar conventional two-phase PLL, as suggested in Fig. 3.26, referred to as a PSGI-PLL. Such a standard PLL building block is commonly available in programming libraries, typically optimized for real time execution, which makes it convenient for implementing this PSF-PLL. The performance of the PSF-PLL is exemplified in Fig. 3.27 and 3.28. Fig. 3.27 displays:

a) in the top trace, the α component of the input signal, and the PSF-PLL output α component, with $K_A = 0.1$ for a slow response, equivalent to a narrow bandwidth for greater attenuation of distortion;

b) in the middle trace, the source and PSF-PLL phase signals, and their difference, all normalized to an amplitude of 1, with $K_s = 1$ for a fast response to phase and frequency changes;

c) in the bottom trace, the PSF-PLL estimated frequency output, with $K_p = 1.7$ for minimal overshoot; the response is also compared to a standard 2-phase PLL, and a slight transient can be observed in response to an input amplitude step of 50%.

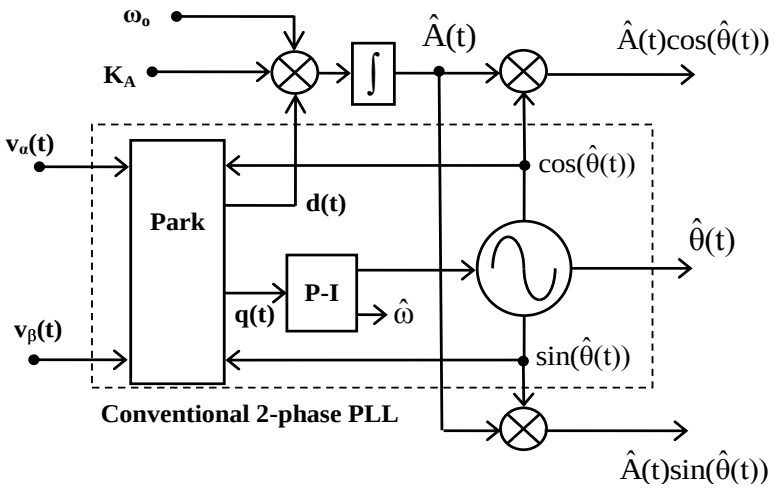


Fig. 3.26 - PSGI-PLL based on conventional two-phase PLL

Fig. 3.28 displays the PSF-PLL response to frequency steps with different values of K_s, from 0.1 to 1, with K_p fixed at 1.7 for the same overshoot requirement.

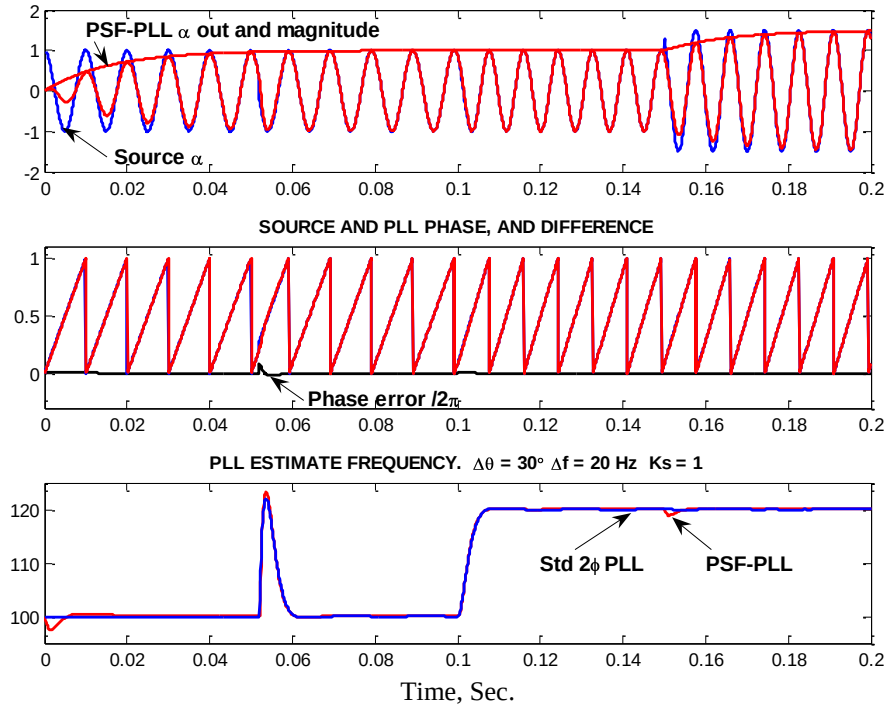


Fig. 3.27 – PSF-PLL response to phase, frequency, and amplitude steps

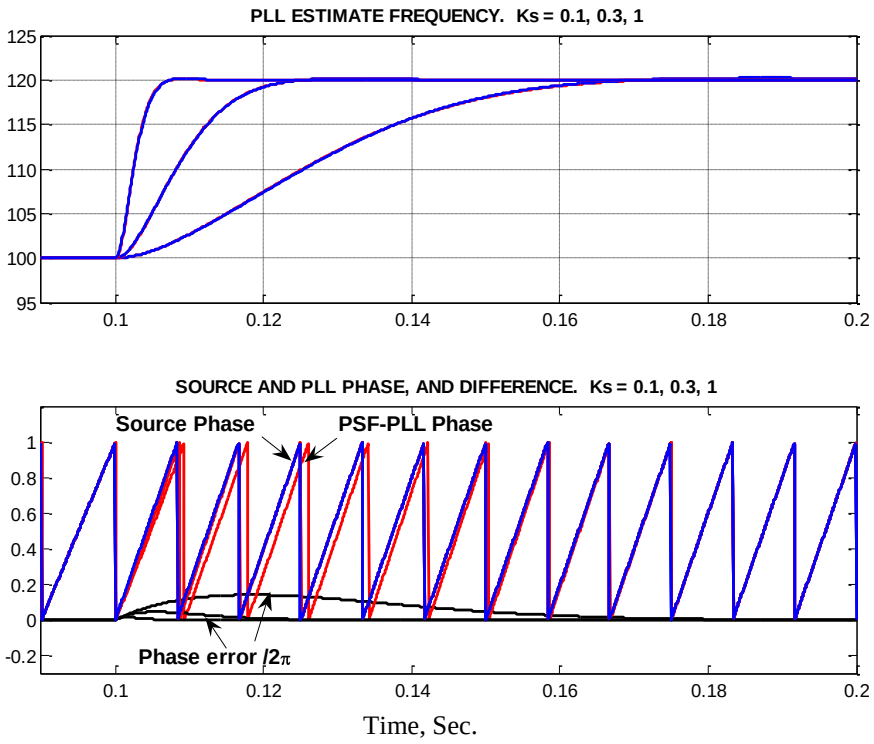


Fig. 3.28 – PSF-PLL response to frequency steps, with K_s = 0.1, 0.3, 1

Although not emphasized in the above two figures, the PSF-PLL design can be observed to exhibit conveniently independent adjustments of the different parameters:

a) K_A sets the rate of amplitude estimation, equivalently to a band-pass filter's bandwidth;
b) Kp sets the damping factor and overshoot of the phase/frequency feedback loop, and can be set once, according to allowable transient dynamic performance, such as keeping frequency estimate overshoot to under 5%;
c) Ks sets the rate of frequency estimation of the PLL, and thus the frequency tracking bandwidth and sensitivity to noise and distortion in the input signal.

3.8 HYBRID NSF-PLL

Whereas the PSF-PLL previously discussed can track frequency variations of a positive sequence signal, it implements only a positive sequence filter, and can thus only attenuate, and not completely cancel, any negative sequence component

of an unbalanced signal, as discussed earlier. To address this shortcoming, the PNSF principle, proposed in Fig. 3.3, can be modified by replacing the fixed frequency PSGI with the PSGI-PLL, as in Fig. 3.29. The proposed structure thus consists of two sub-components: (i) one, the PSGI-PLL operating in synchronous reference frame (SRF), and (ii) the other, the NSGI, operating in stationary, or fixed, reference frame (FRF), resulting in a hybrid combination, referred to hybrid NSGI-PLL, or H-NSGI-PLL. Fig. 3.29 emphasizes:

a) the PSGI-PLL, within a negative feedback structure, produces an estimate of an unbalanced input signal's positive sequence component, and thus, by itself, constitutes a frequency-tracking PSF, or PSF-PLL;

b) the NSGI serves to estimate the negative sequence of an unbalanced input signal, and by the negative feedback, it aims to cancel this negative sequence component from the input to the above PSF-PLL;

c) the estimated frequency is used to tune the NSGI, enabling it to ensure optimal estimation of the negative sequence component, in the face of frequency variations;

d) independent adjustments of the rates of estimations for the positive and negative sequence, via K_{ap} and K_{an}, respectively.

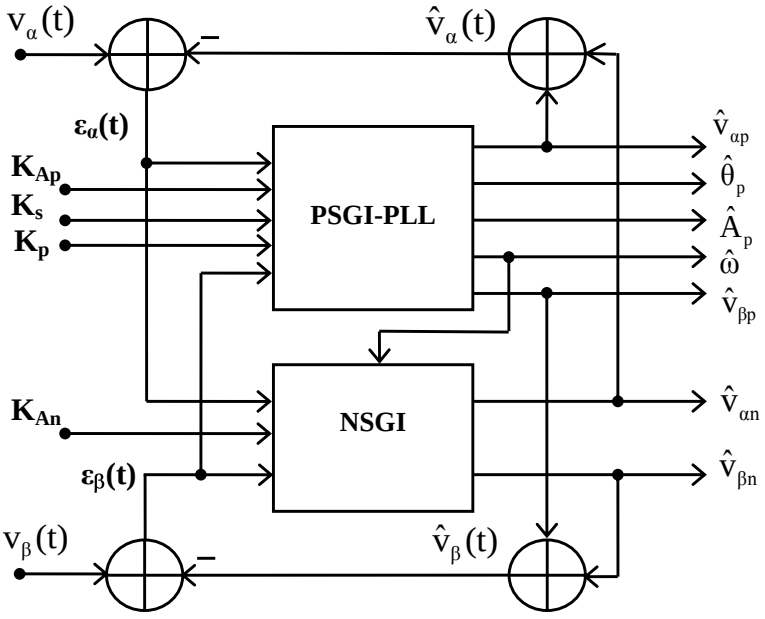


Fig. 3.29 – Hybrid NSGI with PSF-PLL

3.8.1 Phase and Frequency Step Response

Fig. 3.30 demonstrates the performance of the H-NSGI-PLL to a phase step of $\pi/12$ at t = 0.05, and a 20 Hz step in input signal frequency at t = 0.1, with K_s = 0.2, 0.5, and 1, while K_p = 1.7, K_{Ap} = 0.2, and K_{An} = 1; ode-5 is used; thus:

a) the top trace compares the frequency tracking response of the proposed H-NSF-PLL and of a conventional 2-phase PLL, for a balanced input signal. The performance is seen to be very robust over a wide range of Ks values.

b) the middle trace displays the phase of the source signal and as estimated by the H-NSF-PLL. With K_s = 1, the H-NSF-PLL is seen to catch up to the source phase within one cycle, and over 4 cycles with the slower value of K_s = 0.2.

c) the bottom trace displays the negatve sequence filter β output and state vector magnitude. The transient response is directly dependent on the time it takes for the PLL to catch up in phase to the input signal.

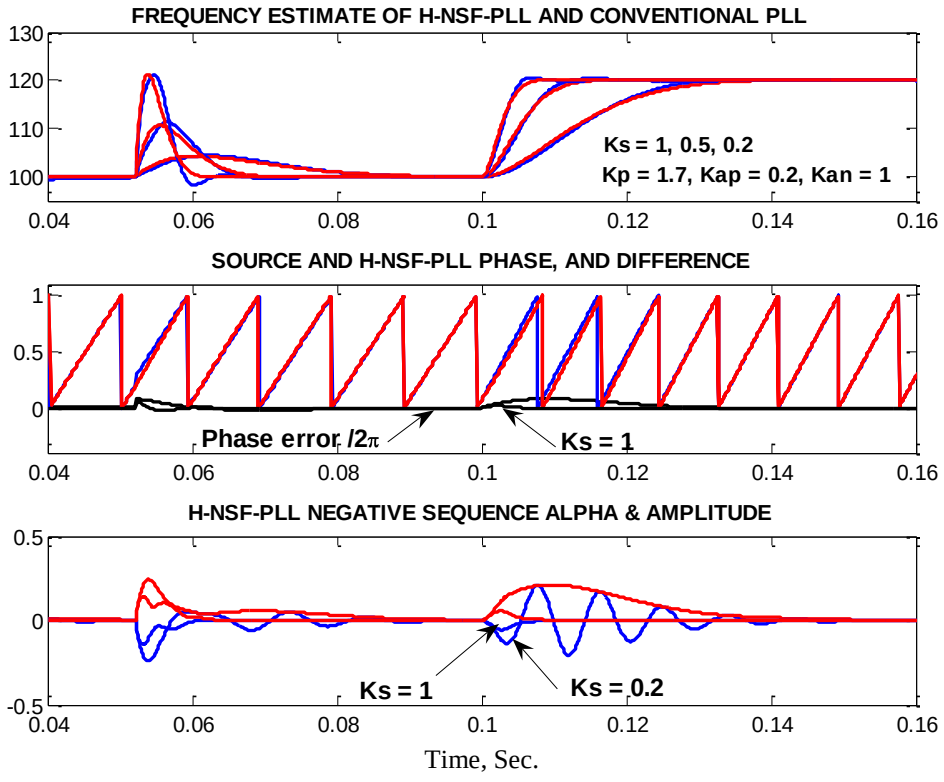


Fig. 3.30 – H-NSGI-PLL response to phase and frequency variations

3.8.2 H-NSF-PLL Response To Unbalanced Input

Fig. 3.31 demonstrates the performance of the H-NSGI-PLL to an unbalanced input condition, characterized by 0.5 pu of negative sequence, starting at t = 0.15, with $K_s = 1$, $K_p = 1.7$, $K_{Ap} = 0.2$, and $K_{An} = 1$; ode-5 is again used.

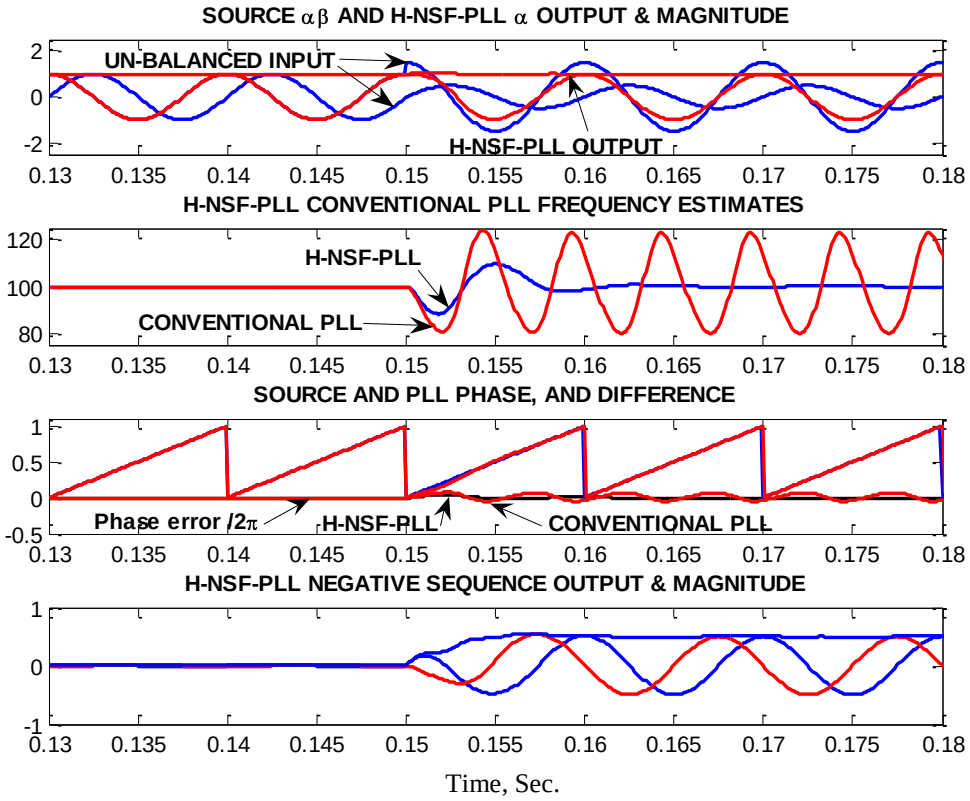


Fig. 3.31 – H-NSF-PLL response to unbalanced input

Fig. 3.31 presents:

a) in the top trace, the positive sequence α output component of the H-NSF-PLL shows a constant amplitude, with a barely noticeable transient after the step in negative sequence.

b) the second trace compares the estimated frequency output of the H-NSF-PLL and of a conventional PLL; after a brief transient, the H-NSF-PLL frequency estimate settles, while the conventional PLL produces a steady state ripple, as it has no built-in mechanism to compensate for the unbalanced input.

c) the third trace shows the phase angles at the sources and estimated by the H-NSF-PLL, and compares the phase difference for the H-NSF-PLL and a conventional PLL; again, the conventional PLL shows a steady state ripple, whereas the H-NSF-PLL quickly settles to a zero steady state error.

d) the forth trace shows the rapid response of the negative sequence output of the H-NSF-PLL, settling within a half-cycle.

3.8.3 H-NSF-PLL Response To Input Amplitude Step

Fig. 3.32 demonstrates the performance of the H-NSGI-PLL to a +50% step in amplitude of the input signal, starting at t = 0.15, with K_s = 1, K_p = 1.7, K_{An} = 1, and K_{Ap} = 0.2, 0.5, and 1; again, ode-5 is used.

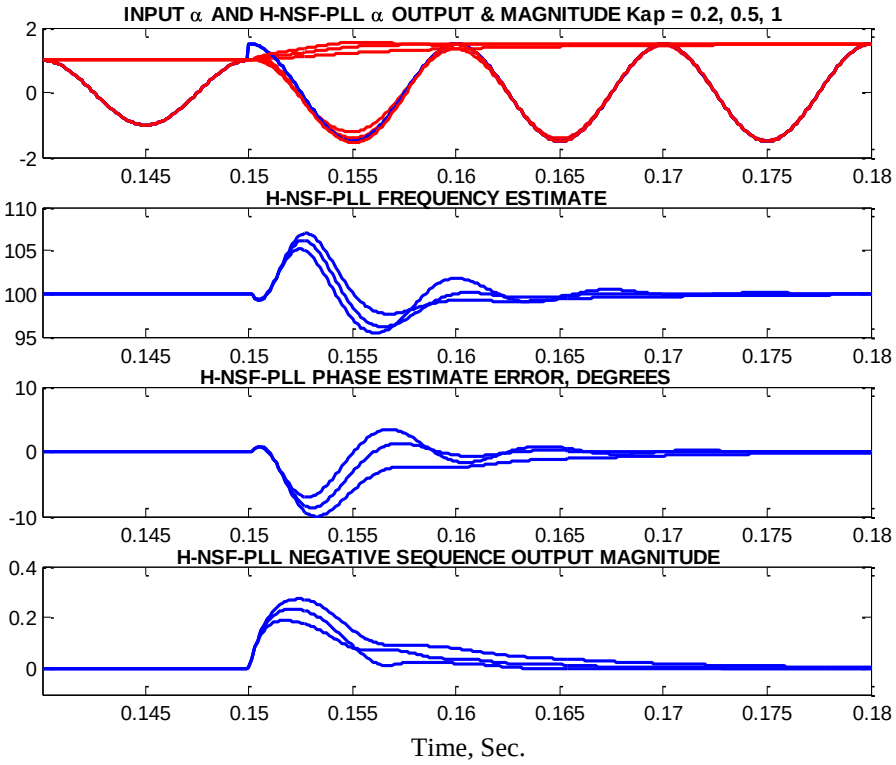


Fig. 3.32 – H-NSF-PLL response to amplitude step

Fig. 3.32 shows:

a) the top trace shows the positive sequence α output and amplitude;

b) the second and third traces show the estimated frequency and phase outputs; the settling time is dependent on the time constant for the amplitude estimation, and is shorter with higher values of K_{Ap};

c) the bottom trace shows how the negative sequence output undergoes a transient response to the positive sequence filter operation.

3.9 HYBRID NSASAE-PLL

As indicated in the previous case of the H-NSF-PLL, the idealized performance was demonstrated using ode-5, to ensure optimal discretization of the resonator-based NSGI for the purpose of verifying the proper operation of the H-NSF-PLL. However, although this can be implemented in real time, ode-5 discretization would require significantly higher computational resources, namely time. Thus, it is proposed to replace the resonator-based NSGI by a ASAE-based equivalent, discussed earlier, which makes possible to implement it with the simplest, and computationally most effective, first order discretization. This results in the structure in Fig. 3.33, referred to as a hybrid negative sequence ASAE with PLL, or H-NSASAE-PLL.

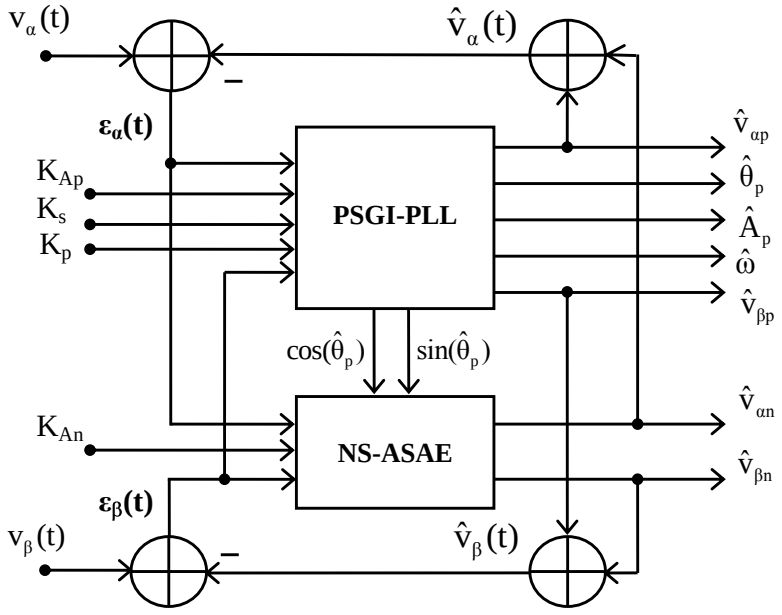


Fig. 3.33 – Hybrid NS-ASAE with PSF-PLL

The principal and only distinction from the previous H-NSF-PLL is that instead of feeding the estimated frequency from the PLL to the negative sequence estimator, the unit magnitude quadrature signals are communicated to it. The performance of the H-NSASAE-PLL is exemplified in Fig. 3.34 to 3.36, which show essentially the same performance as for the H-NSF-PLL:

a) Fig. 3.34 shows the response to phase and frequency steps with K_s = 0.2, 0.5, and 1.0. The estimated frequency shows a robust response over a wide range of K_s, allowing convenient adjustment according to application requirements;

b) Fig. 3.35 shows the response to unbalanced input: the positive sequence and estimated phase angle output shows almost no reaction, and the negative sequence is detected in under a half-cycle;

c) Fig. 3.36 shows the response to a +50% step in input amplitude, and the angle estimate remains within 5^o with the fastest setting of K_{Ap} = 1.

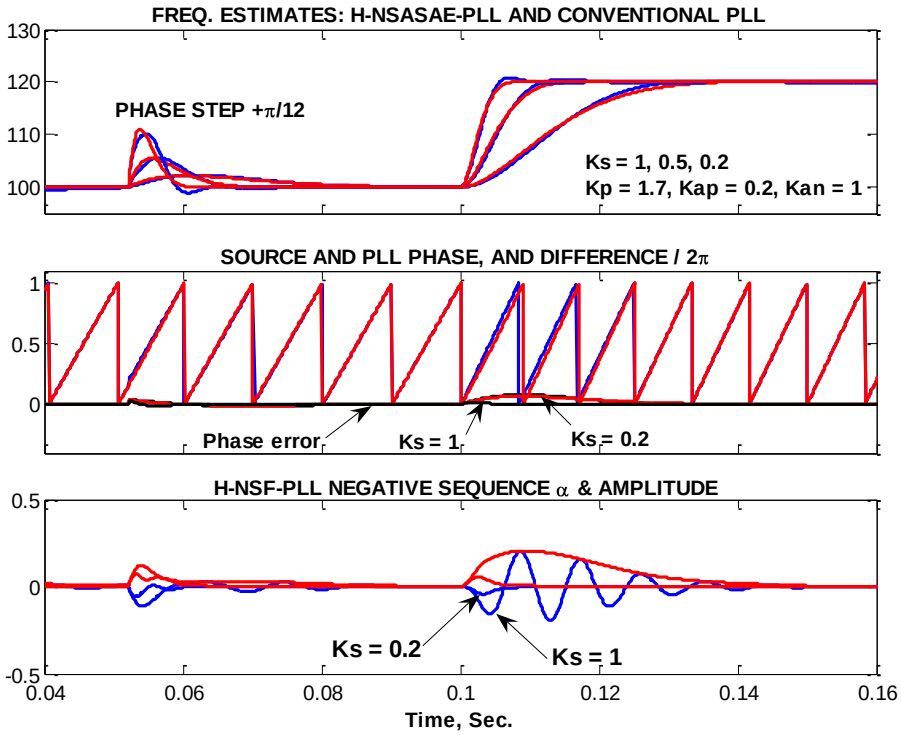


Fig. 3.34 – H-NSASAE-PLL response to phase and frequency variations

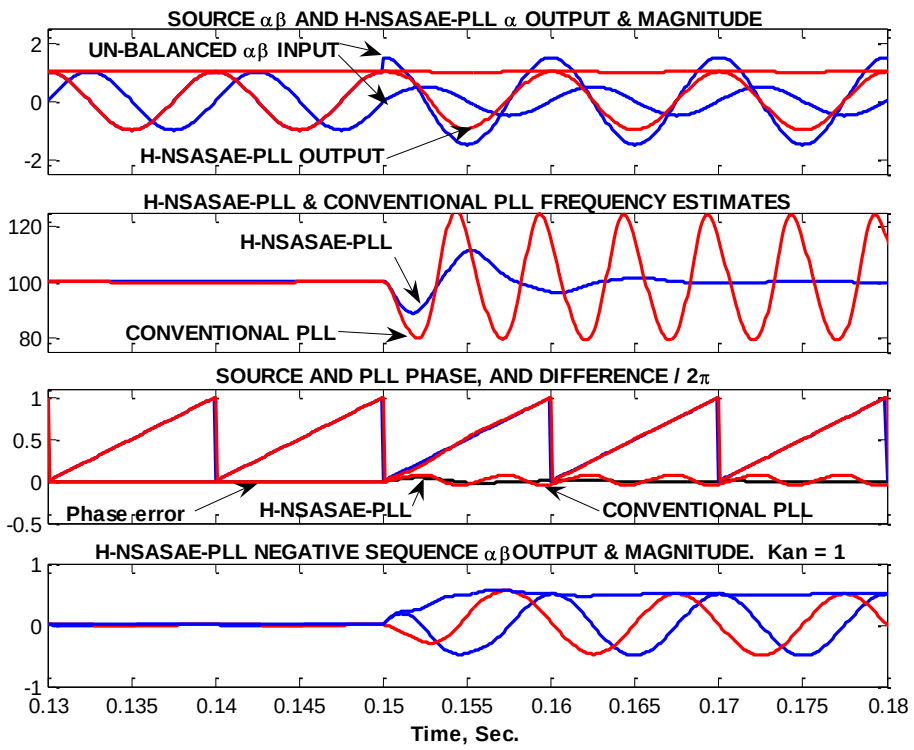


Fig. 3.35 – H-NSASAE-PLL response to unbalanced input

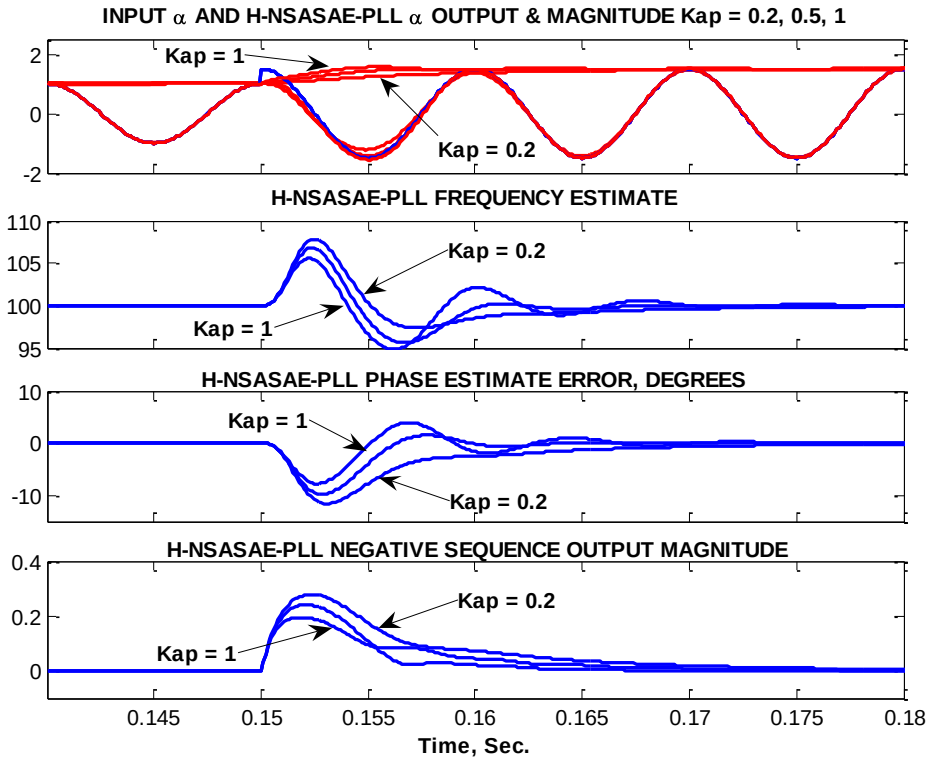


Fig. 3.36 – H-NSASAE-PLL response to amplitude variations

As indicated earlier, due to the intrinsic coupling between the positive and negative sequence filters within the same loop, the negative and positive sequence outputs are not entirely independent, and the severity of the negative sequence output transient is seen to last longer and with greater intensity with a lower PLL frequency tracking bandwidth, attributed to lower K_s.

3.10 CHAPTER SUMMARY

This chapter presented a number of novel techniques for filtering and extracting positive and negative sequence signals from a quadrature input signal, on the basis of the internal model principle and adaptive synchronous amplitude estimation. The IMP served as a basis for a parallel combination of two generalized (resonating) integrators, one each for extracting the positive and negative sequences. The ASAE served to synchronously estimate the amplitudes of the in-phase and quadrature components of the positive and negative sequence components of an unbalanced, quadrature, signal. Finally, these two principles were combined with a conventional two-phase PLL, within a phase and frequency-tracking negative feedback loop, to estimate the phase of the positive sequence component of an unbalanced two-phase signal. The primary benefit of the ASAE mechanism is that it can be implemented with a first order Euler discretization, resulting in lower computational burden, thus making it more attractive for real time applications, while performing as well as it's computationally more intensive resonator-based filter equivalent.

4. DEVELOPMENT AND EXPERIMENTAL SETUP

This chapter discusses the software and hardware platform which serve for the experimental setup for evaluating and comparing the performance of various PLL techniques. The overall development and evaluation framework can be described in terms of 4 principal components:

a) the Matlab/Simulink graphical design platform, by The Mathworks, in which two main sub-systems are developed:
   i. the various PLL algorithms of interest to evaluate,
   ii. test signal and pattern generation and output to DAC (digital to analog converters) for observation and measurement on an oscilloscope, as well as output to disk file for subsequent analysis and plotting;
b) the dSPACE DS1104 R&D Controller board, on which the Simulink test signal generator and PLL algorithms are executed in real time;
c) the dSPACE ControlDesk platform, with which a control panel can be designed to interface and interact with the above program in real time.

Specifically, this chapter presents the following:

a) a brief overview of the DS1104 R&D Controller Board, and the primary features of specific interest for studying the proposed PLL techniques;
b) the Simulink test program, PLL_tests.mdl, which has been developed for evaluating the proposed PLL techniques;
c) the dSPACE ControlDesk program for operating the DS1104, and more specifically, the user operated experiment, PLL_tests_x2.cdx, which presents a convenient control panel for the user to interact with the above PLL test program, as it executes in real time.

It is noteworthy that, although much of the programming for both the test signal generation and the PLL algorithms could be written in such a widely used language as C, it is much more time consuming than with a graphical design tool such as Simulink, and especially costly at the development stage. The graphical design approach allows a much higher degree of flexibility and time-efficiency to

explore, refine, and compare system variations. Once an algorithm has been evaluated and ascertained for robust performance, it can subsequently be re-written in C, or even in assembly language, with a much clearer focus, higher productivity, and with far fewer errors, saving much troubleshooting and debugging time. The graphical approach also makes for much clearer self-documentation and maintainability, further lowering overall development costs.

4.1 THE DSPACE DS1104 R&D CONTROLLER BOARD

The dSPACE DS1104 R&D Controller board, which serves to execute the compiled Simulink model program, is shown in Fig. 4.1. It consists of a PCI card, installed in a PC as a peripheral device, and provides an I/O connector through which to carry analog and digital signals to and from external devices. The board contains one main floating point CPU, and a separate DSP for additional DSP-specific computational power and peripherals.

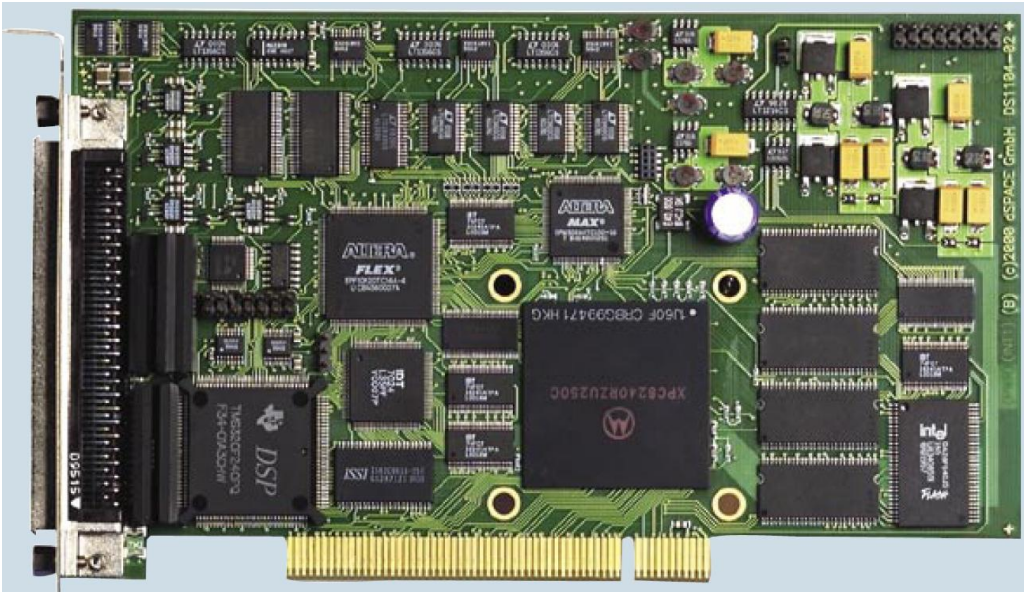


Figure 4.1 – DS1104 R&D Controller board

The DS1104 I/O connector interfaces to a CP1104 connector panel, shown in Fig. 4.2, which provides separate connections for signals including:

a) 8 analog input and 8 analog output signals

b) 20 digital I/O signals

c) 3 PWM channels

d) timer inputs and outputs

e) RS-232 asynchronous serial communication

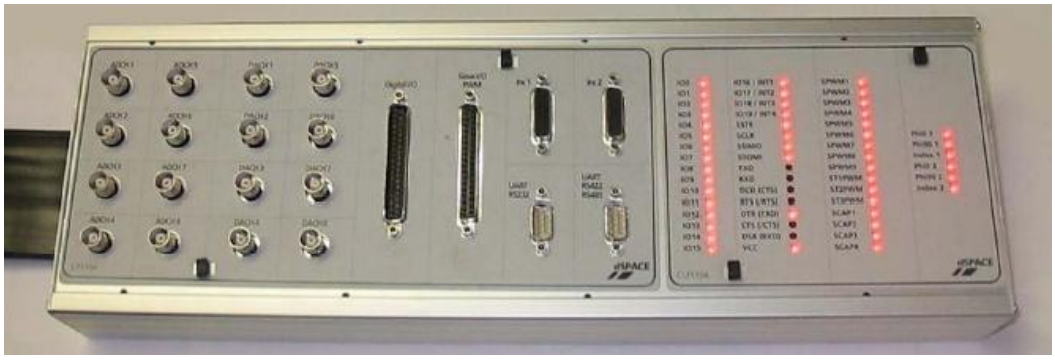


Fig. 4.2 – dSPACE CP1104 connector panel

The DS1104 functionality is represented by the block diagram in Fig. 4.3

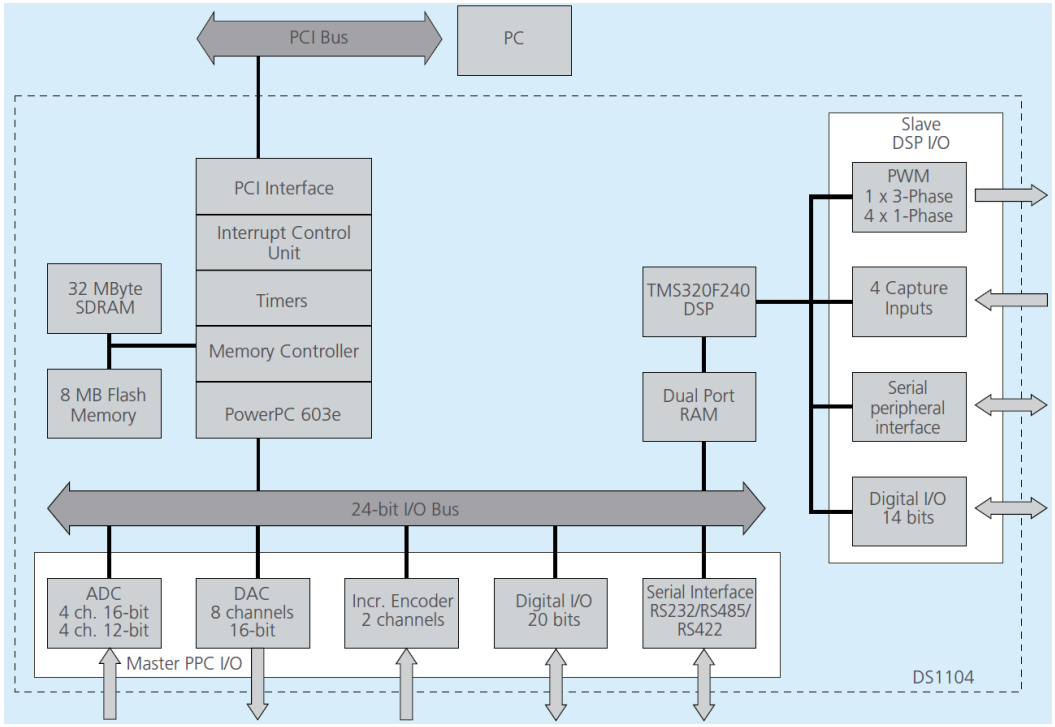


Figure 4.3 – DS1104 block diagram (source: www.dspaceinc.com)

The DS1104 offers the following main features of interest:

- Freescale MPC8240 32-bit floating point processor with PPC603e core and on-chip peripherals, with 250 MHz CPU clock;
- four general purpose 32-bit timers;
- 4-channel 12-bit parallel and 4-channel 16-bit sequential A-D converters, with +/- 10 V input voltage range, with conversion time under 2 uSec.;
- 8-channel 16-bit D-A converter, with +/- 10 V output voltage range;
- 20-bit parallel, TTL compatible, digital I/O;
- Texas Instruments 16-bit fixed-point TMS320F240 DSP, with 20 MHz CPU clock, 10 PWM channels, three 16-bit general purpose timers, 16-channel 12-bit A-D converter

The DS1104 platform is also supported by a library of Simulink blocks to easily access it's various peripherals.

4.2 THE SIMULINK PLL TESTS MODEL PROGRAM

The overall Simulink PLL test program is represented in Fig. 4.4, and consists of 2 principal sub-components:

a) test signal generator;
b) PLL algorithms under evaluation, collected in one block; one PLL algorithm can be selected, or enabled, at any time, for evaluation;

and also the following:

c) a simple switching unit, to connect the test signals either (a) directly to the PLL, such as for simulation, or (b) via DAC and back into ADC, such as for real-time operation, or for evaluating the PLL operation with test signals from another external source;
d) DAC output unit; all the principal signals of interest are directed to several DAC channels, suitably scaled, for observation on an oscilloscope. Due to the large number of signals, and limited number of DAC channels and typically few oscilloscope channels, all signals are directed to a switch, by which the user can select which signals to send out to the DAC's.

Furthermore, a standard TTL level digital signal is output for external synchronization of the oscilloscope, for convenient and steady display.

e) output to a Simulink scope block, which is useful for non-real time simulation, and which can be deleted before compiling the program for real-time execution on the DS1104;

f) a signal sampling block, to sample signals in steady state after transients have settled; signals may be sampled on the rising or falling edge of the signal generator's modulation state signal.

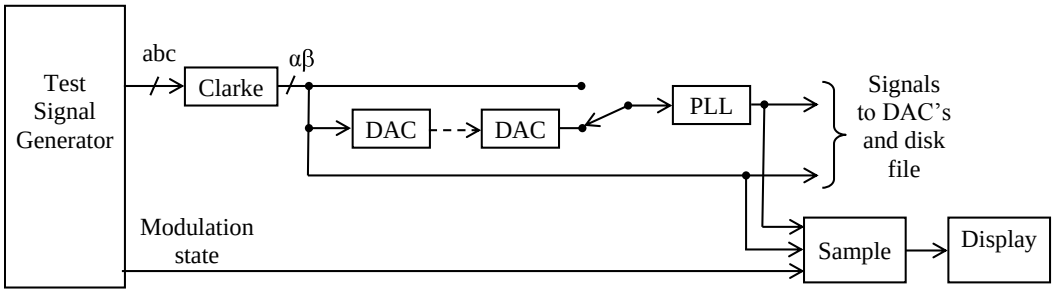


Figure 4.4 – Simulink PLL Tests program block diagram

4.2.1 Test Signal Generator

Phase estimation and synchronization is subject to various signal degradations of the input signal from an ideal constant sinusoidal, such as phase jumps, frequency variations, harmonic distortions, unbalanced and others conditions. Thus, representative test signals need to be synthesized to evaluate proposed PLL algorithms, a task which can also be accomplished with Simulink on the DS1104. The test signal generator operation can be described in terms of two principal components, represented by the block diagrams in Fig. 4.5 and 4.6, respectively:

a) a modulated signal source, which can synthesize:
   i. pure un-distorted two-phase, balanced, sinusoidal signal
   ii. square frequency modulation, e.g. jumping between 60 and 65 Hz
   iii. square phase modulation, e.g. jumping between 0 and 30^{o}
   iv. square amplitude modulation, e.g. jumping between 1 and 1.5 pu

v. square unbalance modulation, e.g. type C_A between 0 and 0.5 pu

vi. square distortion modulation, e.g. between 0 and 20% THD of 5^{th} harmonic

The various modulation values, such as frequency jumps, phase jumps, etc., can be set by the user at run time.

b) a counter, which toggles it's output signal between two states, which is used to determine the state of modulation of the above signal generator. The user can specify the number of whole signal cycles for each modulation state, which makes for convenient observation of transient performance after jumps in phase, frequency, etc. The user can also set the phase angle within a cycle at which the modulation state changes. This allows evaluation of how PLL transient dynamic performance may vary as a function of the instant of modulation state change, e.g. when the phase jumps at the zero crossing of the signal, vs. at the instant of its' peak value.

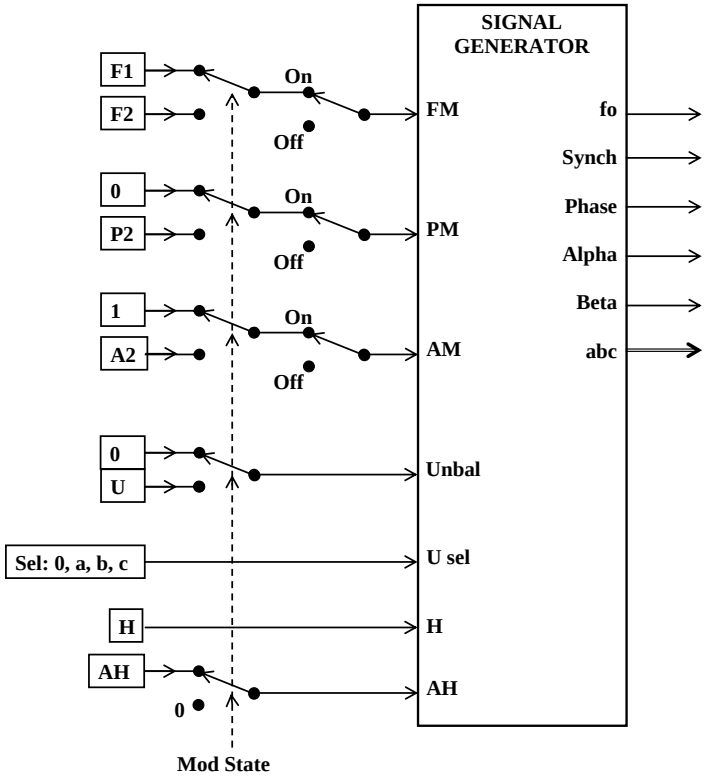


Fig. 4.5 – Modulated test signal generator

The following signal source parameters can be set by the user:

a) F1 and F2: signal frequency, e.g. 60 and 65 Hz
b) P2: phase jump; e.g. 30^{o}
c) A2: amplitude jump; e.g. 0.5 pu
d) U: degree and type of unbalance; e.g. 0.3 pu of type C_A
e) H, and AH: index of harmonic component to add, e.g. 5^{th}, and amount AH, e.g. 0.25 pu (25% THD). Due to limited CPU resources, only one harmonic can be selected at any time.

The modulation state counter operation is determined by three parameters:

- N1: number of signal cycles in state 1 (e.g. at F1) or un-modulated condition (e.g. 1.0 pu amplitude);
- N2: number of signal cycles in state 2 (e.g. at F2) or modulated condition (e.g. 1.5 pu amplitude);
- the phase, within a cycle, at which the modulation state changes.

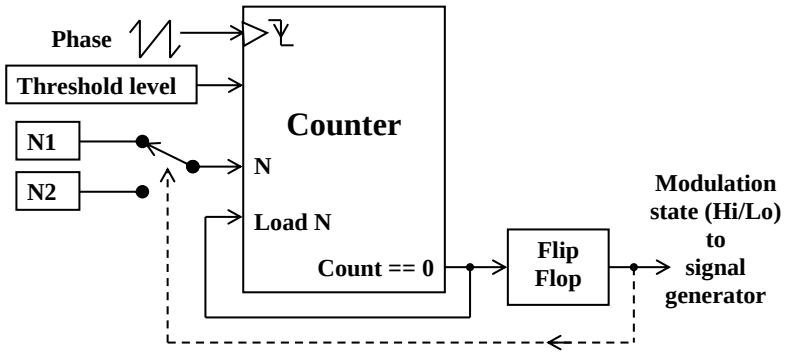


Fig. 4.6 – Modulation state generator

4.2.2 PLL and Output Signal Selection

Simulink provides a convenient non-switched mechanism of selecting one of several signals, referred to as signal merging, as shown in Fig. 4.7. Thus, the user can specify a number to identify one PLL to be enabled, and it's output signals are correctly directed to the single, common, group of output signals for examination.

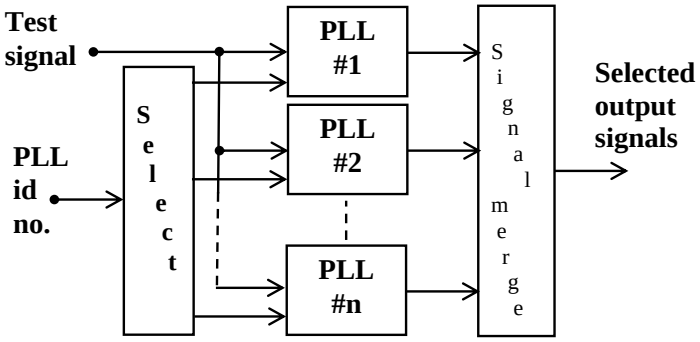


Fig. 4.7 – PLL selection for evaluation

4.2.3 PLL Test Program Operation

The overall operation of the PLL test program can be summarized as follows:

a) the rate of execution of the model, or sampling rate, is set to 10 KHz, being representative of PWM and sampling frequencies commonly used;

b) the test signal generator produces a 3-phase sinusoidal signal, which can be modulated in steps between two states, such as between two frequencies, phases, or amplitudes, etc.; the signal is transformed to two-phase αβ for processing by the selected PLL;

c) the signal can be not modulated at all, for initial verification, or the modulation can be activated for N1 cycles, and de-activated for N2 cycles; N1 and N2 are user-selectable integer values, and serve mainly to define an interval during which the PLL transient response can be conveniently observed on an oscilloscope, for example when the amplitude is stepped up by 0.5 pu, and back down by 0.5 pu;

d) the instants when the signal modulation is activated and de-activated is reflected in a digital synchronization signal on the connector panel, at connector CP17, on pin 20: DIO0, and is intended for external trigger of the oscilloscope. Furthermore, the modulating signal changes state and remains in either state for an integer number of signal cycles, thus making for a steady signal to be observed on the oscilloscope; as well, the triggering signal for the oscilloscope can be selected at either change of modulation;

e) the phase, between 0 and 360^o, within a cycle of the signal, at which this synchronization signal changes state, can be adjusted by the user, in order to examine how the PLL response may vary as a function of the timing of the change of modulation. Thus, for example, the signal generator may produce a signal at 60 Hz, starting at a phase of 45^o, for a duration of exactly 10 cycles, and then switch to 65 Hz, when the phase crosses 45^o, and vice versa;

f) the user may select one, or several, or all, of the modulations discussed earlier (phase, frequency, amplitude, etc.), and the amount of each (e.g. phase step of +/- 45^o, frequency step between 60 and 70 Hz, etc.);

g) because of the relative complexity of the overall Simulink model, the operational features are limited, such that, by design, only one harmonic frequency can be selected at a time; however, power electronic systems can entail harmonics of quite high frequency, such that a sampling rate of 10 KHz has been deemed an adequate minimum;

h) the 3-phase generated signal is transformed to a 2-phase αβ signal for the PLL; although 3-phase input PLL algorithms could be studied, it is most common in the literature that 3-phase signals are first transformed to 2-phase αβ reference frame; thus, because of the linear transformation and equivalence between three-phase and two-phase signals, for the sake of clarity, all relevant signals are examined in the two-dimensional αβ reference frame;

i) the user can select to connect the αβ signal directly to the PLL, or out to DAC, and then back in via ADC and then to the PLL; this allows the possibility of applying some other external signal, such as from a 3-phase power source, to the PLL for synchronization;

j) the various PLL parameters can be adjusted by the user;

k) when simulated in the Simulink environment, the PLL output signals can be directed to the Matlab workspace, for subsequent plotting and analysis;

l) for real time operation on the DS1104 platform, the PLL output signals are directed to DAC output, for examination on an oscilloscope. Due to the

limited number of DAC channels, some PLL signals are permanently output to some dedicated DAC channels, and other signals can be selected for output, one at a time, to DAC channel 6, 7, or 8. Thus, the user can conveniently examine any of all PLL signals, up to 4 at a time on a 4-channel oscilloscope;

m) some signals can be sampled on either the rising or falling edge of the modulation synchronization signal, for output on a numeric display on the ControlDesk control panel, to examine their steady state values after settling of the PLL transient response; these signals include: (a) the PLL estimated frequency, (b) positive sequence amplitude, (c) negative sequence amplitude, (d) the amplitude of the selected harmonic;

n) up to 15 signals of interest can be recorded to Microsoft Excel compatible disk file. Typically, this can be set up to occur at some threshold time after user button click, for a user specified number of samples. The date can be subsequently further analyzed and plotted.

4.3 dSPACE CONTROLDESK REAL-TIME OPERATION PLATFORM

The dSPACE DS1104 operation is supported by the dSPACE ControlDesk platform, with which so-called experiments can be designed. Such an experiment consists mainly of a control panel designed with user operated controls, indicators, and signal display devices, and provides the ability to record signals to disk files. Thus, the Simulink PLL test program is operated by the PLL_test_x2.cdx experiment, which appears as in Fig. 4.8, and consists of the sub-components described below.

4.3.1 Documentary and Operational Information

Brief program information is provided in the following:

a) the title box, at the top left, identifies the principal source files involved in the compilation of this entire operation;

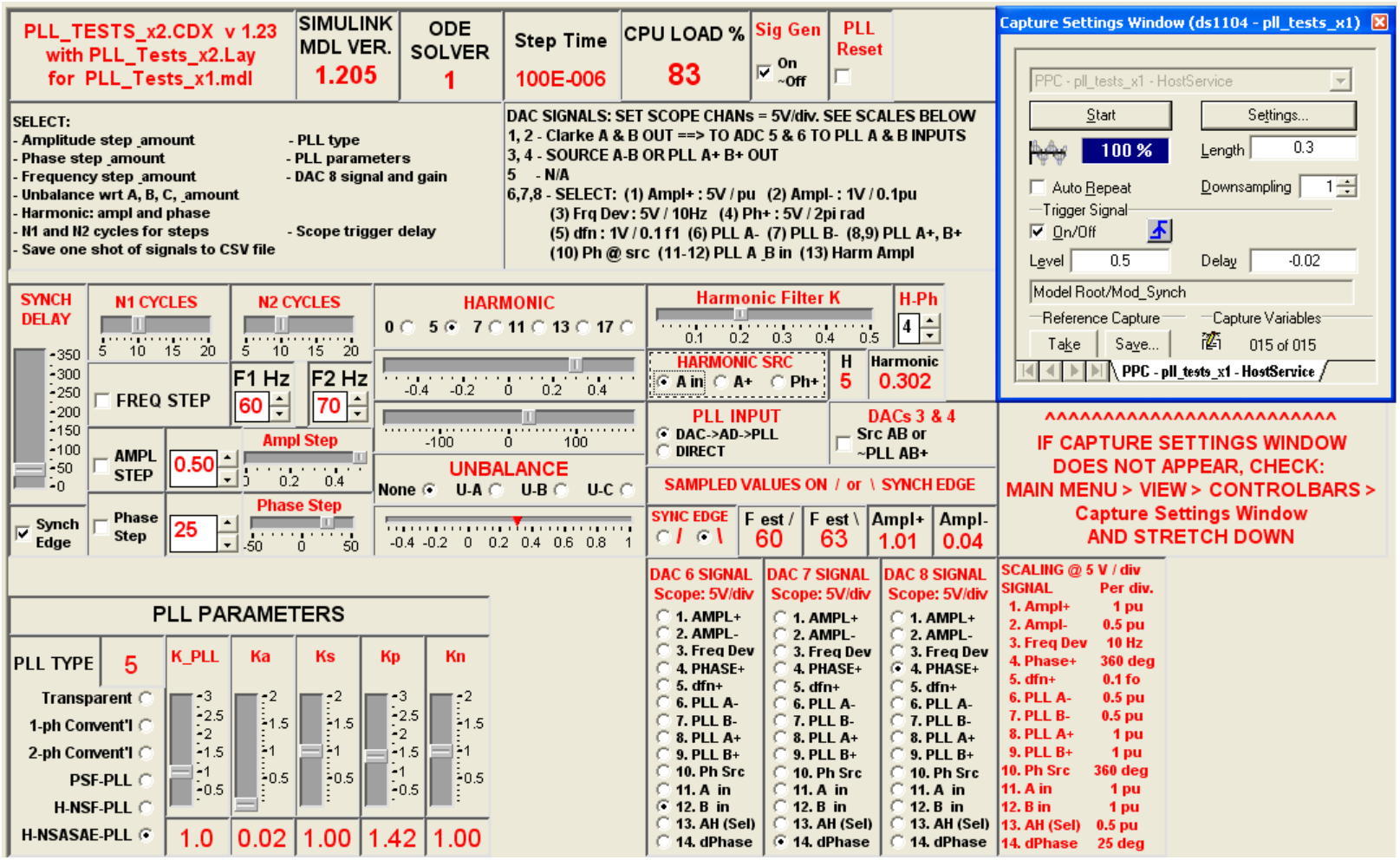


Figure 4.8 – dSPACE ControlDesk PLL Tests experiment control panel

b) the version number for the Simulink PLL Tests model file, which is typically updated in the Simulink model whenever it is edited, before compilation, as a means of confirmation that the program running under ControlDesk is actually the latest edited version;
c) CPU load % – this indicates the fraction of time, per sampling interval, assuming 10 KHz sampling rate, that the CPU is busy executing the model. This factor depends on the model complexity, and also on the solver selected in the Simulink model before compilation, and is a useful indicator for purposes of optimizing the model design;
d) Sig Gen check box – this is provided to enable or disable the signal generation function, and serves mainly to monitor it's impact on the CPU loading, as well as the loading of the selected PLL algorithm;
e) very brief summary of operating instructions;
f) summary of the 8 DAC output signals;

4.3.2 Signal Generation

Test signal synthesis is controlled by the following user set parameters:

a) SYNCH DELAY – this allows the user to adjust the phase, within a cycle of the generated sinusoidal signal, at which the modulating command signal will change state; on an oscilloscope display, this will cause the display to shift left or right as this slider is adjusted;
b) N1 CYCLES – this selects an integer number of signal cycles in one state of the modulation pattern;
c) N2 CYCLES – this selects an integer number of signal cycles in the other state of the modulation pattern;
d) FREQ STEP check box, F1 Hz, and F2 Hz – these select one and another frequency of the signal, when the FREQ STEP is checked on, producing a frequency step modulation pattern;
e) AMPL STEP check box, and Ampl Step slider – this enables step amplitude modulation, and the amount, in pu;

f) Phase Step check box, and Phase Step slider – this enables phase modulation, and the amount, in degrees;
g) Harmonic selection, and amplitude slider – this enables harmonic modulation, on and off, and the amount, in pu;
h) Unbalance selection, and slider – this enables 3-phase unbalance modulation, on and off, and the amount of the negative sequence, in pu. One of the three selections can be made, of U-A, U-B, or U-C, which correspond to unbalance types C_A, C_B, and C_C, respectively;

4.3.3 PLL Parameters

PLL operation is determined by the following user settings:

a) PLL TYPE selection – this allows to select one of several pre-defined PLL algorithms, including:
    1. straight through connection of αβ inputs to AB^+ and AB^- with no PLL at all; this is provided just for testing and verification purposes;
    2. conventional single-phase PLL, for reference;
    3. conventional quadrature-phase PLL, alsofor reference;
    4. PSF-PLL
    5. Hybrid NSF-PLL – a novel proposed algorithm
    6. Hybrid NSASAE-PLL – another novel proposed algorithm
b) K_PLL slider – this is normally set to 1, but allows to adjust all dynamic PLL parameters, i.e. K_A, K_S, and K_{RES}, together, to change the time scale and response time of the PLL operation;
c) K_A – this adjusts the rate of ASAE in the Hybrid NSF-PLL and Hybrid NSASAE-PLL algorithms, and effectively sets the bandwidth for amplitude response, and attenuation of harmonics in the extracted positive sequence output signals;
d) K_S – this sets the speed of response of the PLL phase detection loop;
e) K_P – this sets the proportional gain in the PLL phase detection loop;
f) K_NSF – this sets the bandwidth for the negative sequence filter;

g) PLL INPUT check box – select: DIRECT connection of signal generator to PLL, or via DAC to ADC
h) Harmonic Detection Rate (bandwidth) slider – this sets the bandwidth for the harmonic detection;

4.3.4 DAC Signal Output

To accommodate all signals to be output for observation through a limited number of DAC channels, the signals can be selected for output, as follows:

a) DACs 3 & 4 check box: select signal source α and β, or PLL positive sequence α+ and β+ output;
b) DAC 6, 7, and 8 signals – select one of the signals indicated for output, e.g. for observation on an oscilloscope

4.3.5 Steady State Value Display

Dynamic system performance is characterized, among other things, by steady state quantities, after the transient response has settled. A mechanism is provided to sample various values of interest, such as estimated frequency, on the transition of the modulation state signal, thus sampling designated signals in their steady state condition, provided that N1 and/or N2 are adequately selected to ensure sufficient settling time. These values are identified in the panel entitled "SAMPLED VALUES ON / or \ SYNCH EDGE":

a) SYNCH EDGE check box – select the edge of the modulation comnand signal on which to trigger the oscilloscope;
b) F est / and F est \ – this indicates the sampled value of the PLL output estimated frequency at either the rising or falling edge of the above synchronization signal;
c) $Ampl^{+}$ and $Ampl^{-}$ – this indicates the sampled value of the PLL output estimated amplitude of the positive and negative sequences, at either the rising or falling edge of the above synchronization signal;
d) Harmonic – this indicates the detected harmonic amplitude level at the selected edge of the synchronizaton signal;

e) HARM DIST check box – select source for the harmonic estimation unit, ether at the α input to the PLL, or at the PLL A^+ output; this allows a subsequent measure of the harmonic attenuation.

4.3.6 Data Output to Disk File

Although it appears as part of the control panel design, the Capture Settings Window is actually a floating, un-docked, panel that can be moved around, but has been placed in the upper right corner for visual convenience. Further operation details are provided in the ControlDesk reference documentation.

4.4 OVERALL PERFORMANCE

In view of all aspects of development and real time operation of the test program, the overall performance can be characterized by such measures as:

a) development effort and time: at one extreme, graphical design tools, such as Matlab/Simulink, are extremely productive, for the high level at which a programming task can be formulated, and meaningful results can be obtained very quickly. In contrast, text based programming, such as in C, is intrinsically at a lower level, and thus requires more detail to be specified.
b) efficiency of compiled code: generally, a program developed at a higher level will result in significantly more executable code, and greater execution time than code produced from a text based program. The overall program development task can be partitioned between the two, but this is very much a matter of programmer experience.
c) hardware platform speed: this ultimately defines a real limit to the amount of computations that can be performed, and may therefore strongly dictate how an overall programming task is to be partitioned between higher level graphical design and lower level text based coding.

The end goal thus dictates various constraints and costs, and determines the emphasis to be attributed to the respective factors identified above.

5. EXPERIMENTAL RESULTS

This chapter presents experimental results of performance tests, which compare the proposed PSF-PLL, H-NSF-PLL, H-NSASAE-PLL algorithms with the conventional two-phase PLL. The test signal generator and PLL algorithms execute together as part of a single Simulink model, running in real time on a dSPACE DS1104 platform, at a sampling rate of 10 KHz. The tests are categorized as follows, with the respective objectives as indicated, aiming ultimately to emphasize the superior performance of the adaptive techniques, and especially of the H-NSASAE-PLL under unbalanced input conditions:

a) Frequency step and phase step tests

   These standard tests show that the performance of the proposed adaptive techniques does not suffer at all as a consequence of the additional complexity over the conventional PLL.

b) Amplitude and distortion tests

   These standard tests show the superior performance of the adaptive techniques over the conventional PLL, in their frequency selective filtering ability. Thus, they are able to attenuate distortion in the estimated output signals, with a rate of attenuation determined by the gain Ka.

c) Unbalance tests

   The unbalanced input tests show the ability of the H-NSASAE-PLL to completely compensate for unbalanced input conditions, resulting in zero steady state error in under one cycle, even with an extreme unbalanced condition, as might result from a fault on the utility grid.

Thus, these experiments aim mainly to verify the primary targeted benefit of the negative sequence detection and compensation of the proposed H-NSASAE-PLL, and it's superior performance with the simplest order of discretization. Finally, it should be noted that for first order discretization of the Simulink model, the synchronous amplitude estimation in the H-NSASAE-PLL results in markedly superior performance over the H-NSF-PLL. As a result, for some tests, the H-NSF-PLL is not considered.

5.1 FREQUENCY STEP TESTS

Two frequency step tests, executed in real time, are presented to verify that:

a) the performance of the proposed H-NSF-PLL and H-NSASAE-PLL is comparable to that of the conventional PLL, with little or no adverse impact due to the additional negative sequence detection mechanism;

b) the performance remains robust over a wide range of the parameter Ks, which thus allows tuning of the PLL operation according to application requirements.

5.1.1 Frequency Step – Fast Response

The first frequency step test is defined by Ks = 1, for a fast frequency estimation response of the PLL. The negative sequence detection filter response is adjusted with Kn = 0.5, to dampen oscillations due to it's dynamic interaction with the positive sequence detection achieved by the PLL with adaptive positive sequence amplitude estimation. The response is shown in Fig. 5.2, for an input frequency step from 60 to 70 Hz. The estimated frequency signals can be seen to be almost identical for all four PLL types. The source and PLL estimated phase signals for the H-NSASAE-PLL are shown in trace (B). The major distinction appears in the estimated phase deviation signal, in trace (C); the H-NSF-PLL and H-NSASAE-PLL show slightly greater peak phase deviation, but with slightly faster settling.

5.1.2 Frequency Step Response with Varying Ks

The second frequency step test is defined by Ks = 1, 0.33, and 0.1, for different speeds of frequency estimation response of the PLL. The responses are shown in Fig. 5.2. The negative sequence detection filter response is adjusted with Kn = 1, which results in slight overshoot in, and slightly faster, frequency response for the H-NSF-PLL and H-NSASAE-PLL. Except for the fastest setting of Ks, the frequency responses can be seen to be near identical for all PLL types.

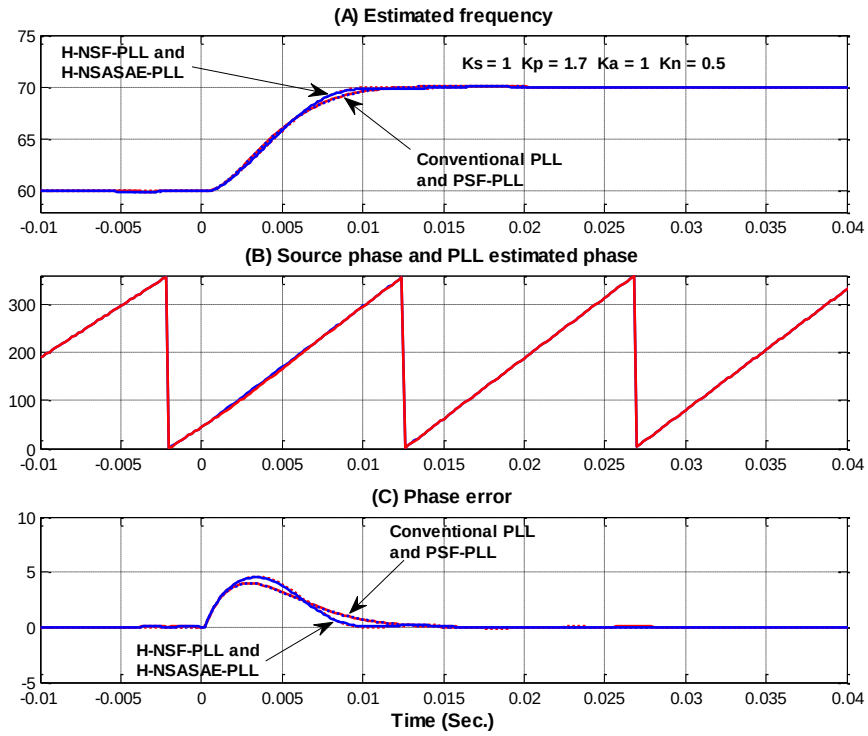


Figure 5.1 – Frequency step responses, Ks = 1.0. Experimental results.

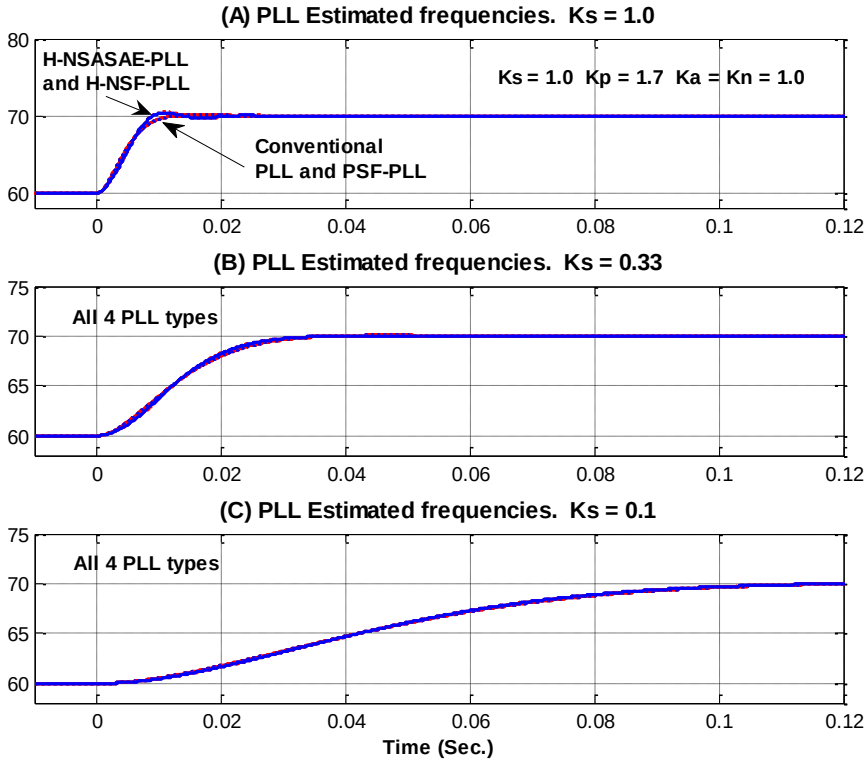


Figure 5.2 – Frequency step responses, with Ks = 1.0, 0.34, 0.1. Experimental results.

5.2 PHASE STEP TESTS

In view of the primary purpose of a PLL to provide an estimate of the input signal phase, phase jumps, such as due to loads being suddenly connected or dis-connected, or line faults, probably constitute the most severe disturbances that a PLL can be subjected to. Phase disturbances, for purposes of evaluating PLL performance, can be usefully categorized as:

a) due to loads being connected or dis-connected, resulting in the same phase jump on all three phases of the source voltage, and nominally constant voltage amplitude;

b) due to unbalanced line conditions, such as due to an unbalanced load, or due to line faults, which result in unequal phase and amplitude jumps on all three phases. This case is dealt with in the unbalanced input test.

Fig. 5.3 shows the PLL estimated frequency and phase in response to a phase jump of 25^{o}, with Ks = 0.8, Kp = 1.7, Ka = 1, Kn = 0.4.

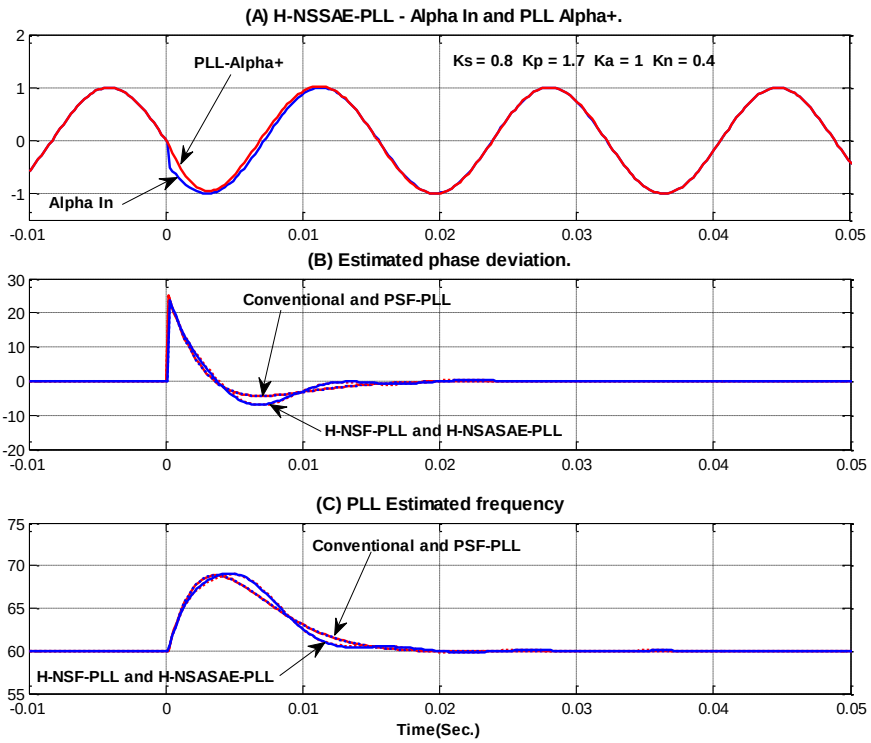


Figure 5.3 – Phase step response, Ks = 0.8, Kp = 1.7, Ka = 1, Kn = 0.4. Experimental results.

The parameters, and especially Kn, are set to minimize the H-NSF-PLL and H-NSASAE-PLL transient oscillations due to the negative sequence estimation filter. The top trace (A) shows the phase jump of the input α signal, and the estimated α output signal from the H-NSASAE-PLL. The estimated phase and frequency signals from all four PLL types are shown in traces (B) and (C), respectively, and can be seen to be very similar. As observed before, the H-NSF-PLL and H-NSASAE-PLL exhibit slightly less damped transient oscillations, due to the negative sequence filter component. However, given the same Ks and Kp parameter values for all PLL types, Kn can be adjusted for the H-NSF-PLL and H-NSASAE-PLL to attenuate these transient deviations to acceptable levels.

Fig. 5.4 shows the PLL phase responses to a 25^{o} phase step, with Ks = 0.8, 0.45, and 0.1, and emphasizes the robust performance of all PLL types with respect to the speed parameter. Kn is again set for minimal transient oscillations in the H-NSF-PLL and H-NSASAE-PLL types. Except for a small difference in response for the fastest value of Ks, all PLL types show almost the same performance, characterized by the same transient phase undershoot of approximately 4^{o}.

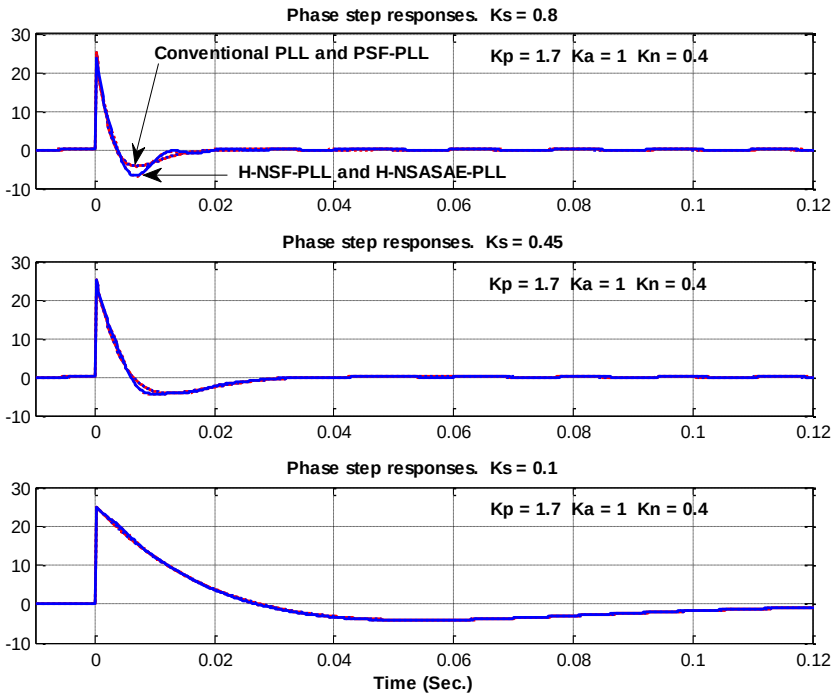


Figure 5.4 – Phase step responses, Ks = 0.8, 0.45, 0.1. Experimental results.

5.3 AMPLITUDE STEP TESTS

Fig. 5.5 shows the various PLL estimated amplitude responses to a +50% amplitude step, for amplitude adaptation gain Ka values of 1, 0.3, and 0.1. The instantaneous response of conventional PLL stands out because of the absence of any filtering; it's amplitude signal comes from the Park transformation unit, which embeds no dynamics at all, and thus produces an instantaneous result. It's amplitude output signal is thus the fastest, but as the distortion test will also show, it is the most sensitive to amplitude distortions. In contrast, all the other PLL's embed an amplitude estimation process, with user adjustable rate of convergence, which effectively accomplishes a filtering action which will prove beneficial to attenuate the impact of harmonic distortion in the input signal.

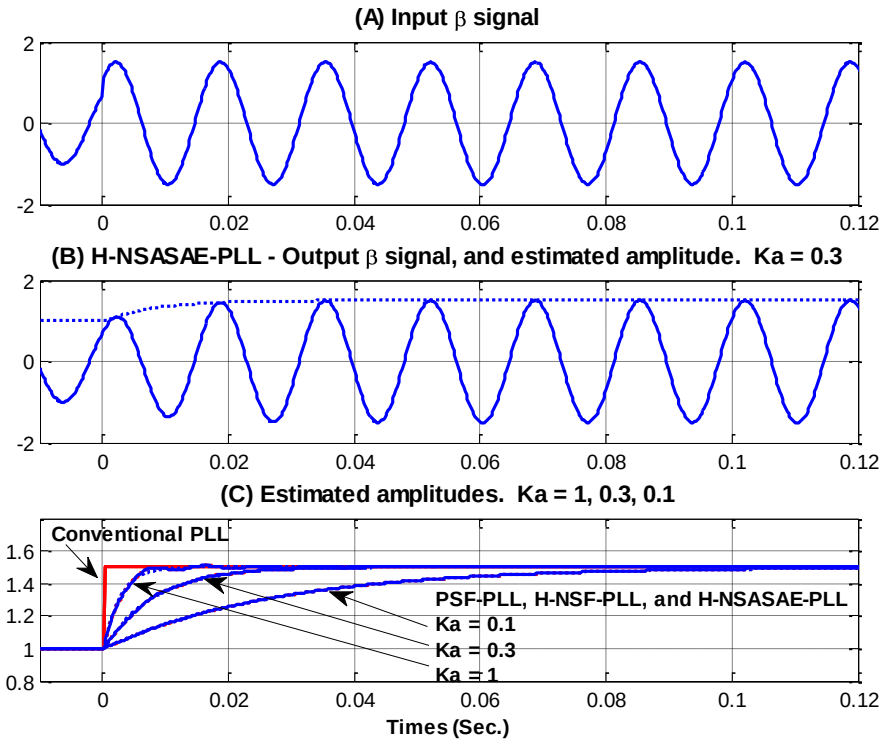


Figure 5.5 – Amplitude step response, Ka = 1, 0.3, 0.1. Experimental results.

For these three PLL algorithms, since the amplitude estimation process is the same, by design, their amplitude estimates have almost the same response. Close examination reveals a slight transient oscillation for the H-NSF-PLL and H-NSASAE-PLL, again, due to the dynamic interaction between the positive and

negative sequence estimation sub-filters. However, except for this small oscillation, the rate of convergence closely agrees with the PSF-PLL.

Fig. 5.6 indicates the sensitivity of the phase and negative sequence estimation dynamics to an amplitude step, and specifically to parameter Ka. For a +50% amplitude step, the peak phase deviation is under 3^o at the slowest setting of Ka = 0.1, and under 2^o at the fastest setting, Ka = 1. Similarly, the negative sequence peak amplitude is under 0.1 pu for Ka = 0.1, and under 0.05 pu for Ka = 1. These transients reflect the intrinsic dynamic interaction between the positive and negative sequence filters, and the response speed of the negative sequence estimation. Depending on requirements for a specific application, the negative sequence filter response can be further improved with suitable adjustment of Kn.

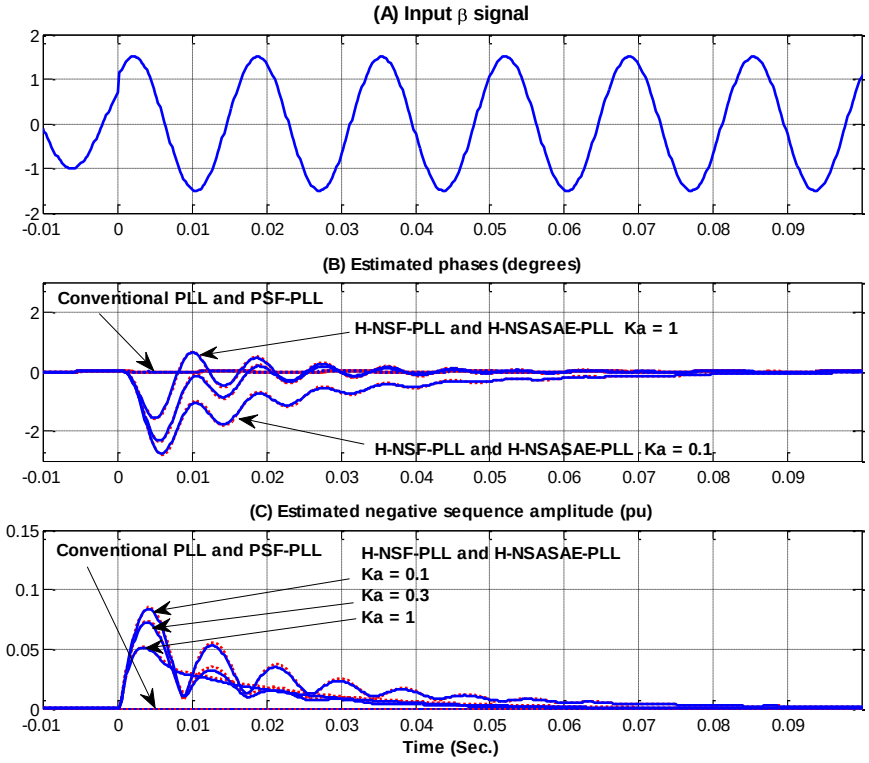


Figure 5.6 – Phase and negative sequence responses to amplitude step. Experimental results.

5.4 HARMONIC DISTORTION TESTS

Due to the overall complexity of the Simulink test model, and limited computing resources on the DS1104 platform for real time execution and data recording, harmonic distortion tests are performed with a single harmonic component, namely the fifth, since it is typically the dominant component in three-phase systems. Because of the relative independence of the amplitude estimation dynamics from the phase-frequency estimation dynamics, the impact of input harmonic distortion can be controlled quite independently on the αβ output signals and on the phase and frequency signals. Thus, the level of distortion in the αβ output signals is mainly determined by Ka, which effectively sets the bandwidth of the PSF-PLL, H-NSF-PLL, and H-NSASAE-PLL as frequency-tracking band-pass filters. In contrast, Ks determines the frequency estimation bandwidth, and thus the impact of the input distortion on the estimated phase.

Fig. 5.7 shows the input α signal in trace (A), and the α signal from a conventional PLL in trace (B), and from the H-NSASAE-PLL, with Ks = 1 and 0.1, in traces (C) and (D), respectively. The conventional PLL exhibits little variation in output distortion as Ks is varied from 1 to 0.1, which reflects the direct coupling of input distortion through the Park transformation, to the output. In contrast, the H-NSASAE-PLL exhibits a level of distortion in the output α and β which directly depends on Ka. Furthermore, since the cos and sin functions in the α and β signals depend on the estimated phase angle, any ripple in that quantity is reflected in the α and β outputs, which thus further depends on Ks. Trace (D) shows, for Ka = Ks = 0.1, a level of distortion of 0.8%, which represents an attenuation of 31 times, or 30 dB.

Fig. 5.8 indicates the independence of the conventional PLL output distortion from either Ka or Ks. In contrast, the PSF-PLL, H-NSF-PLL, and H-NSASAE-PLL αβ outputs are determined by the adaptive amplitude estimation, which is

determined by Ka, which allows for the level of distortion to be controlled by the user.

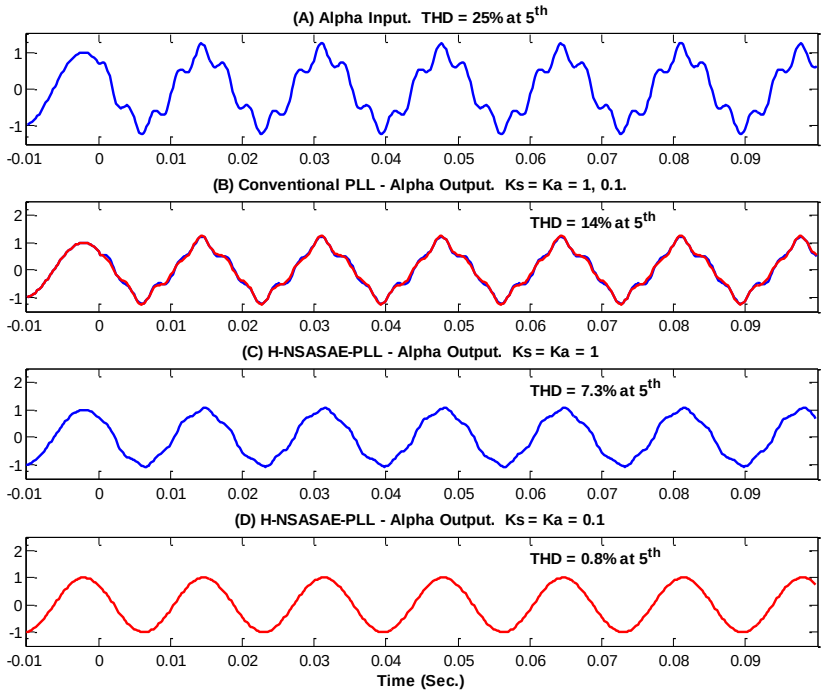


Figure 5.7 – Alpha input and output signals. Experimental results.

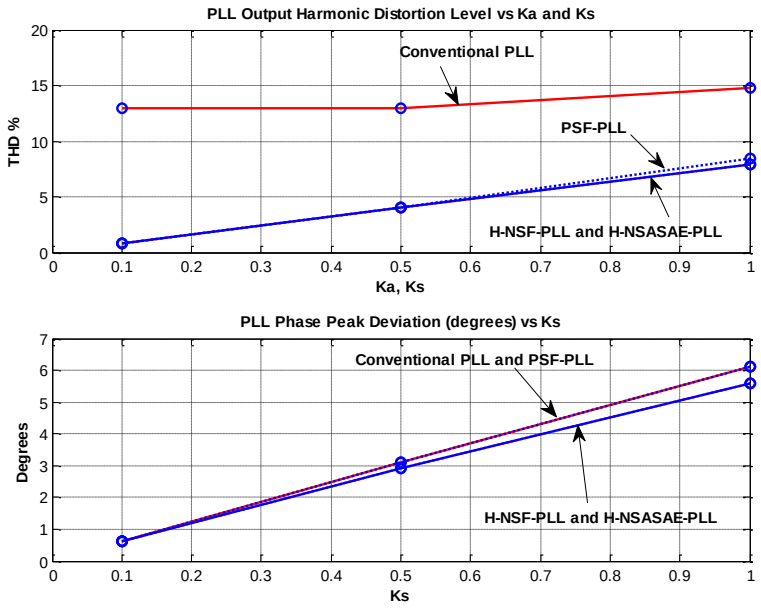


Figure 5.8 – PLL α output distortion, and peak phase deviation. Experimental results.

Fig. 5.9 shows the specific case of the dependence of the H-NSASAE-PLL estimated phase and peak deviation on Ks, and clearly shows the direct nearly proportional relationship between these two quantities.

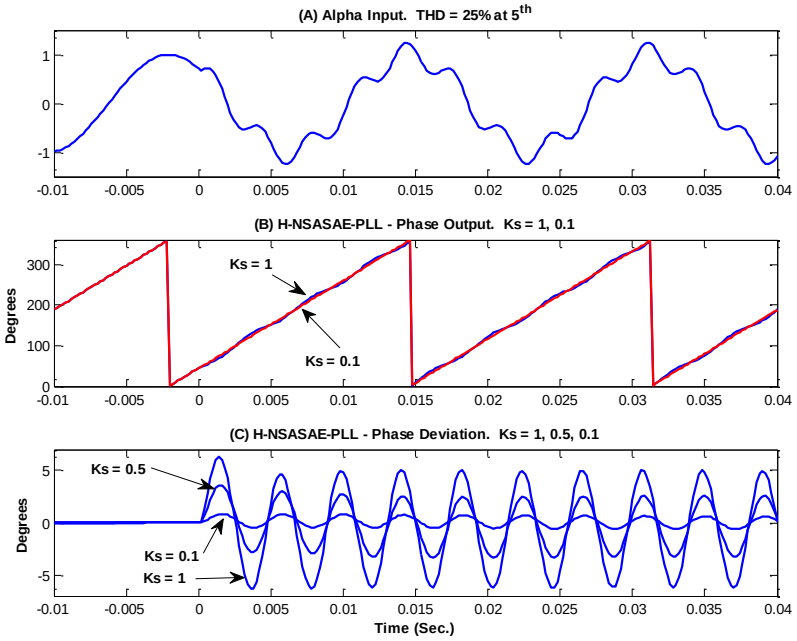


Figure 5.9 – H-NSASAE-PLL estimated phase and deviation. Experimental results.

5.5 UNBALANCE TESTS

The unbalanced input tests show the superior performance of the H-NSASAE-PLL over the other PLL algorithms being examined. The following aspects become evident in the results presented in this section:

a) the conventional 2-phase PLL provides no compensation for unbalanced inputs, which is most clearly revealed in the estimated phase and phase deviation signals;

b) the PSF-PLL can only attenuate the impact of the negative sequence component, according to the Ka parameter, but not completely compensate for it;

c) the H-NSF-PLL exhibits a small steady state oscillation in the phase deviation, most likely on account of the discretization of the continuous

time model; this aspect has not been examined in any further detail in the present scope of study;

d) the H-NSASAE-PLL clearly compensates for the unbalanced input, as evidenced in the zero steady state phase deviation. Furthermore:
   i. the peak phase deviation is directly controlled by the Ks parameter;
   ii. the rate of estimation of the negative sequence amplitude is controlled by Kn;
   iii. some internal oscillatory interaction is evident, between the positive and negative sequence detection filters, which can be controlled and minimized by lowering parameters Ka and Kn;
   iv. a higher setting of Kn results in faster negative sequence amplitude estimation and also in higher peak phase deviation, but this can be reduced with a lower value of Ks.

For the sake of clarity in the results presented, unbalances comparable to type C_A, [7-8], are examined, which is found to be representative of other types of unbalance. Such unbalance is characterized by (A) a larger magnitude of phase X, relative to the magnitude of the other two phases, and (B) a symmetrical change of phase in these other two phases, as impacted by a symmetrical three-phase negative sequence component. After a Park transformation, the resulting α and β components are orthogonal, and therefore more convenient to examine.

5.5.1 Estimated Phase Response to Unbalanced Input

Fig. 5.10 displays the following:

(A) an unbalanced 2-phase input, with the α and β signals increased and decreased by 50%, respectively,

(B) estimated phase responses from a conventional 2-phase PLL; phase angle contains a steady state ripple at 2ω ;

(C) estimated phase responses from a PSF-PLL; similarly, phase angle contains a steady state ripple at 2ω ;

(D) estimated phase responses from a H-NSASAE-PLL.

The H-NSASAE-PLL shows the clearest phase response in steady state, as is further confirmed in Fig. 5.11, which shows the phase deviation.

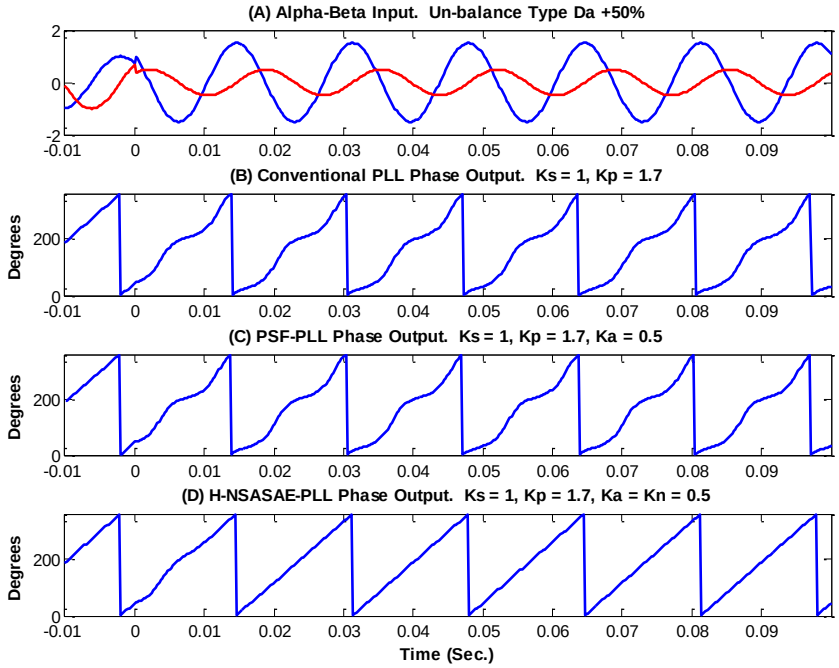


Figure 5.10 – Estimated phase response to unbalanced input. Experimental results.

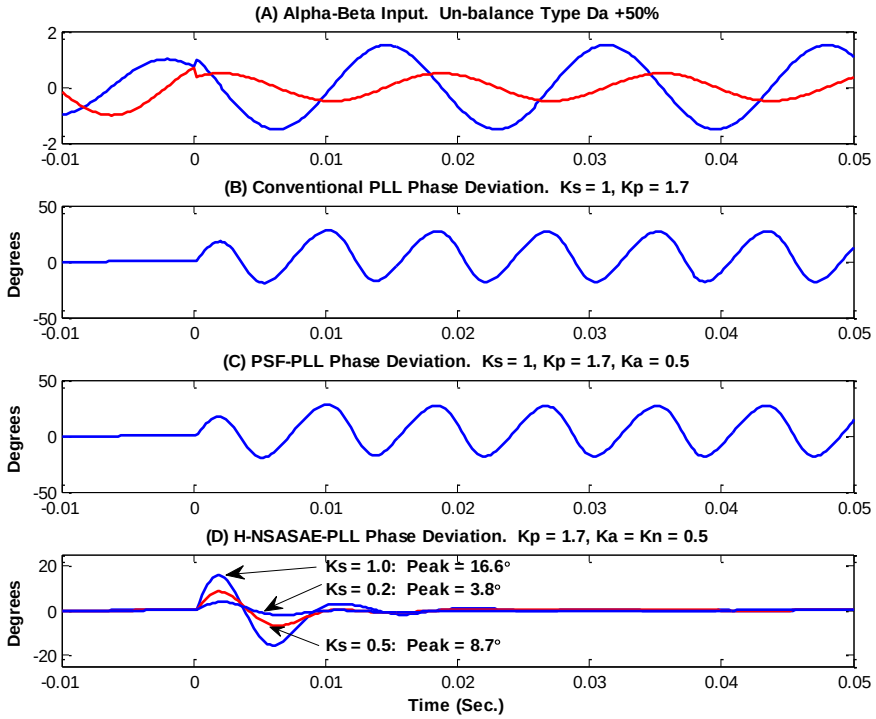


Figure 5.11 – Estimated phase deviation response to unbalanced input. Experimental results.

Trace (D) in Fig. 5.11 also shows how the peak phase deviation, which reflects the phase-frequency estimation feedback loop, is controlled by the Ks parameter:

- with Ks = 1.0: peak phase deviation = 16.6^{o}
- with Ks = 0.5: peak phase deviation = 8.7^{o}
- with Ks = 0.2: peak phase deviation = 3.8^{o}

The H-NSASAE-PLL clearly and completely compensates for the negative sequence component in the unbalanced input signal.

5.5.2 Positive and Negative Sequence Estimated Amplitudes

Fig. 5.12 shows the estimated positive and negative sequences and amplitudes, in response to an unbalanced input. The positive sequence amplitude can be seen to be barely affected, and the negative sequence estimated amplitude settles within approximately a ½ cycle.

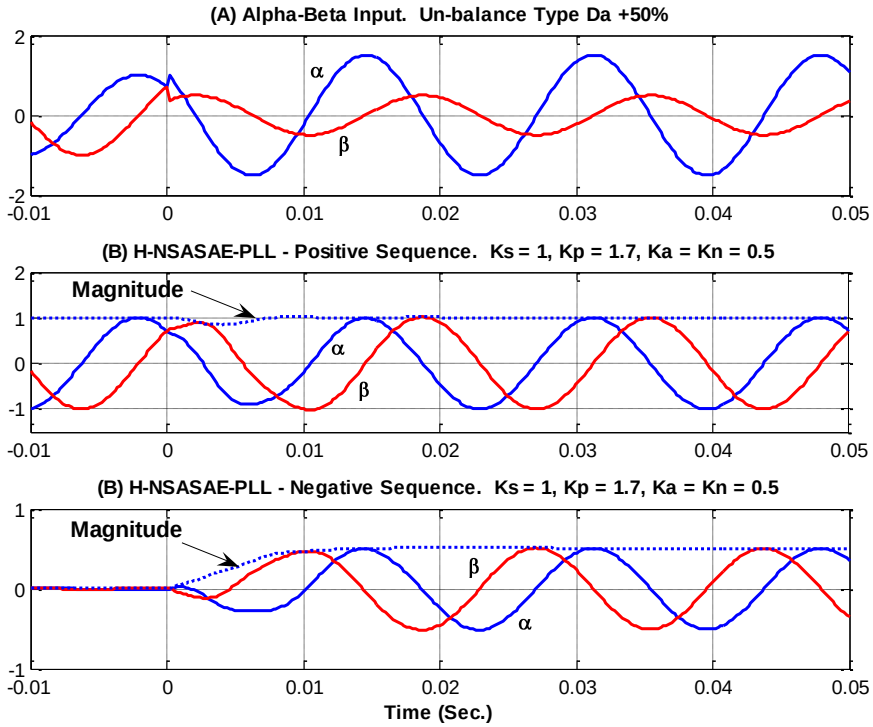


Figure 5.12 – Positive and negative sequence estimated amplitudes. Experimental results.

In Fig. 5.13, parameters Ka and Kn are increased from their previous values for a faster response of the estimated amplitudes, and Ks is varied from 0.2 to 1.0. Although not shown, the amount of transient oscillation in the negative sequence amplitude depends on the instant within a cycle at which the unbalance occurs, and the worst case is displayed in Fig. 5.13. As trace (B) shows, the amount of oscillation in the negative sequence amplitude can be controlled by the Ks parameter, which also affects the peak estimated phase deviation, shown in trace (C). With Ks = 0.5, there is small overshoot, and the amplitude is detected in under a ½ cycle.

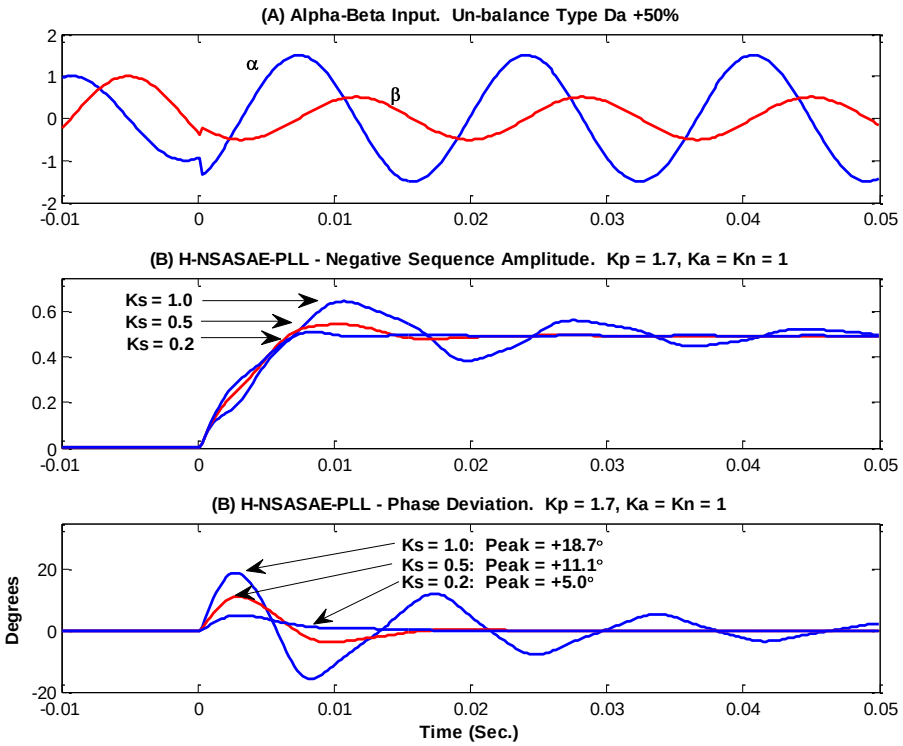


Figure 5.13 – Faster negative sequence estimation. Experimental results.

5.5.3 Extreme Unbalanced Condition

Fig. 5.14 shows the H-NSASAE-PLL positive and negative sequences and respective amplitudes, and phase deviation responses to an unbalanced condition with the α and β signals at 2 pu and 0 pu, respectively, which exemplifies a line fault condition. With Ka = Kn = 0.5, the response is not as fast as with Ka = Kn =

1, but a robust performance is assured in response to such an extreme unbalanced condition. The negative sequence amplitude is seen to settle in under one cycle.

- with Ks = 1.0: peak phase deviation = 47^{o}
- with Ks = 0.5: peak phase deviation = 21.5^{o}
- with Ks = 0.2: peak phase deviation = 7.9^{o}

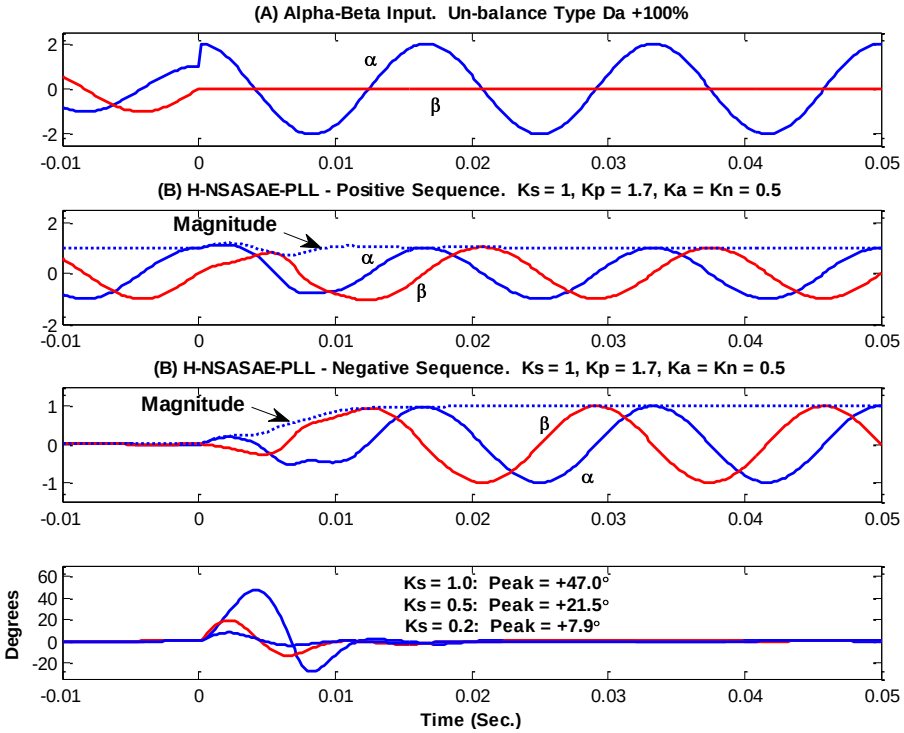


Figure 5.14 – Extreme unbalanced input condition. Experimental results.

5.6 EXECUTION TIMES

Although a graphical tool such as Simulink typically leads to very significant savings in development time, there is intrinsically a cost in another aspect of the overall system design. The underlying mechanism of such a tool, which makes it easy to use, and is completely transparent to the designer, typically results in compiled code, which is actually executed in real time by the target processor, which is considerably more complex than if it were developed by an experienced text-oriented programmer, such as in C. Indeed, programming in a well established language for such purposes, such as C, at a lower level of detail,

closer to the target platform, that is, in terms more specific to the target platform, typically results in much less actual executable code, which is thus less demanding of CPU execution time. It is thus of some interest to consider the execution times for the various PLL types examined in this chapter, to get a sense of the implication of coding the equivalent functionality in C, especially from the perspective of an experienced programmer.

Table 5.1 serves to compare the PLL algorithms examined in this chapter, executed on the dSPACE DS1104 platform, with a main 32-bit floating point processor running at a 250 MHz clock speed. It is noteworthy that the version of Matlab/Simulink used for this development, namely version 2008a, supports only 64-bit floating point integrators, which results in additional computational burden on a 32-bit floating point processor. One can thus immediately anticipate a significant gain in execution speed, if some, or all, of the equivalent functionality were programmed in C using 32-bit float type variables. Whereas the CPU load accounts for the execution of the entire program, including the test signal generator, the PLL load figure measures the difference in CPU load relative to the first case of no PLL, and thus indicates the figure of interest for each respective PLL. The execution time is calculated on the basis of the 250 MHz CPU clock speed, and a program sampling rate of 10 KHz. All indicated figures are approximate within about 1-2 units, as the CPU load value is sampled by the ControlDesk platform, and is somewhat jittery.

Table 5.1 – PLL execution times

	PLL Type	CPU Load %	PLL Load %	Execution time μSec.
i.	None	73	N/A	N/A
ii.	Conventional 1-phase	78	5	5
iii.	Conventional 2-phase	80	7	7
iv.	PSF-PLL	80	7	7
v.	H-NSF-PLL	84	11	11
vi.	H-NSASAE-PLL	80	7	7

5.7 SUMMARY

The main conclusions of interest indicated by the experiments presented in this section can be summarized as follows:

a) for un-distorted sinusoidal inputs, for the same values of Ks and Kp, the frequency and phase step responses are almost identical. This basically reflects the fact that the estimated amplitude dynamics are independent of the other two phase and frequency estimate states, which are closely coupled. As a direct consequence, phase and frequency deviation input stimulus excites the dynamic sub-system defined by the latter two states, and not the amplitude estimate state.

b) Amplitude variations, while the signal itself remains at constant fundamental frequency, excite the amplitude estimate loop dynamics, and the impact will be reflected in the output signal amplitude, and will thus be determined by the amplitude estimate adaptation gain Ka. Ka thus effectively controls the response time, such that the input-output relationship reflects a band-pass filter with bandwidth set by Ka. In contrast, the conventional PLL produces an instantaneous output amplitude signal directly from the Park transformation, without any filtering, and will thus reflect the same level of amplitude variation as in the input signal.

c) Harmonic distortion of the input signal impacts both the amplitude estimation and the phase-frequency estimation dynamics. The net estimated positive sequence α and β output signal distortion is then due to:

    i. the estimated amplitude dynamics, which is determined by Ka; a lower Ka slows down the estimation dynamics, and thus the bandwidth, and the level of output distortion;

    ii. the estimated phase and frequency dynamics, which is mainly determined by Ks; similarly, a lower Ks values reduces the level of output distortion. Thus, any ripple in the estimated angle is reflected in the instantaneous output α and β signals.

In contrast to the techniques with adaptive amplitude estimation, the conventional PLL provides no inherent mechanism to attenuate the impact of harmonic distortion, which thus directly impacts the PLL estimated phase and frequency.

d) For an unbalanced input, only the H-NSASAE-PLL provides the necessary mechanism to counteract it's negative sequence component and thus eliminate it's effect on the estimated amplitude, phase and frequency in steady state operation.

e) The parameters can be conveniently tuned to meet the needs of a given application, to ensure performance within specified tolerances.

6. CONCLUSION

This thesis presents a novel PLL algorithm to estimate the phase of the positive sequence of an unbalanced and distorted three-phase grid voltage. The proposed technique consists of two estimators operating in parallel, in synchronous reference frame, within a quadrature signal feedback structure, one each to estimate the positive and negative sequence of an unbalanced two-phase signal. The positive sequence estimator consists of a conventional two-phase PLL, augmented with synchronously adaptive estimation of the positive sequence amplitude. The negative sequence estimator performs a similar adaptive synchronous estimation of the negative sequence, using the quadrature signals, with unit magnitude, from the PLL. The two operate together, according to the internal model principle, such that the estimated negative sequence cancels the actual negative sequence of the input signal, resulting in improved performance of the positive sequence PLL. The phase of the positive sequence can be estimated rapidly, and with zero steady state error with respect to the unbalanced condition.

6.1 CONTRIBUTIONS AND BENEFITS

The principal contributions of this thesis, and respective benefits, can be summarized as follows:

a) H-NSASAE-PLL algorithm

- The internal structure of the H-NSASAE-PLL is of minimal dynamical order, namely 5, required to estimate the positive sequence phase, frequency, and amplitude, and of the negative sequence quadrature components.
- The synchronous nature of the estimation algorithm makes it straightforward to discretize it, using the simplest Euler type integrations for all estimated quantities.

- Such discretization makes it easily realizable for real time execution on a DSP, at minimal computational cost.

- The proposed structure allows for simple and convenient adjustment of the PLL parameters to accommodate a wide range of application requirements.

- The performance is robust with respect to PLL parameters and input signal conditions.

b) H-NSF-PLL algorithm

- This establishes the validity of the frequency-tracking ability of the H-NSASAE-PLL. In contrast to the H-NSASAE-PLL, the negative sequence is estimated by a second order generalized integrator, which performs well with higher solver order, as offered by the Matlab/Simulink platform, but which is computationally more demanding and less suited for real time execution on a typical DSP.

- For real time implementation with the lowest order of integration, this algorithm motivates the synchronous approach in the H-NSASAE-PLL to estimate the negative sequence.

c) PNSF

- This establishes the validity of the fundamental underlying principle at work in the above two, namely the parallel estimation of the positive and negative sequences of an unbalanced signal within a common feedback structure, which makes possible the rapid estimation of the phase of the positive sequence with zero steady state error.

- The parallel structure of two separate second order generalized integrators, one for each sequence, makes possible to adjust each independently for different bandwidths, as dictated by requirements for a given application.

d) PSF-PLL algorithm

- This constitutes a frequency-adaptive positive sequence filter, with adjustable bandwidth.
- This serves primarily to establish the mathematical foundation and practical usefulness of the equivalency between the second order generalized integrator and the adaptive synchronous amplitude estimation processes, for a time-varying signal frequency.

e) Test signal generator program

- This generator, developed in the Matlab/Simulink environment, produces a variety of standard test signals, with easily adjustable signal and PLL parameters, to evaluate any selected PLL algorithm.
- This generator executes in real time within the same program as the selected PLL under evaluation, and enables rapid and convenient comparison of PLL performances.
- Signals can be easily recorded to Microsoft Excel-compatible disk file, for subsequent analysis and plotting.

6.2 FUTURE WORK

The H-NSASAE-PLL algorithm presented in this thesis is a starting point, with many opportunities for further work. Whereas the development of the H-NSASAE-PLL was accomplished in the convenient graphical language of the Matlab/Simulink platform, with enormous savings in development time, the resulting compiled code is not efficient, as indicated by the level of CPU usage to execute the code in real time on the experimental DS1104 platform. In order to maximize the utilization of a DSP's various resources in real time, especially if some control operation is also required in the application, algorithms are often written in the well established C language, which has been widely adopted for

embedded DSP applications, such as in power electronics applications. Thus, any consideration to adopt the H-NSASAE-PLL in any power electronic application should first address this aspect of implementation.

Furthermore, the theoretical stability of the H-NSASAE-PLL was essentially inferred on the basis of the established stability of the various underlying principles, namely the conventional two-phase PLL, the ASAE, and the PNSF. The stability of the combination of a PLL with ASAE was likewise established for the single-phase EPLL, which revealed the independence of the phase-frequency and of the amplitude estimation dynamics, and this served as the starting point for the proposed H-NSASAE-PLL. Although experimental results were in agreement with the simulation results, and clearly demonstrated the stability of the H-NSASAE-PLL, it should nevertheless prove worthwhile to establish it's theoretical stability, for example, on the basis of Lyapunov stability theory.

Finally, this new H-NSASAE-PLL approach essentially addresses a simple fundamental question: what information is available in the input signal that can be used to improve this estimation process. Based on the adaptive notch filter principle presented in chapter 2, it is evident that additional frequency deviation information is also available, that can potentially be used, for example, in a feed-forward manner to accelerate the phase estimation process. In the same view, in the case of three-phase, four-wire, grid voltage, there are three independent signals, and not just two, as implied by the conventional three-phase to two-phase Clarke transformation, as typically used before application to a two-phase PLL. Thus, this also presents the availability of additional information with which to estimate the phase of interest. Admittedly, such a frequency deviation signal is equivalent to a derivative action on the phase deviation signal, and thus also risks introducing more noise. However, this is a matter to be carefully weighed against the benefit of the additional information provided by the frequency deviation signal.

REFERENCES

[1] O. Vainio, S. J. Ovaska. Digital Filtering for Robust 50/60 Hz Zero-Crossing Detectors. IEEE Transactions on Instrumentations and Measurement, vol. 45, no. 2, p. 426, April 1996.

[2] M. Karimi-Ghartemani and M. R. Iravani. A Nonlinear Adaptive Filter for Online Signal Analysis in Power Systems: Applications. IEEE Transactions on Power Delivery, vol. 17, no. 2, p. 617, April 2002.

[3] H. Akagi, E. H. Watanabe, and M. Aredes, "Instantaneous Power Theory and Applications to Power Conditioning," J. Wiley and Sons, 2007.

[4] Z. Fang, W. Longhui, C. Zhe, W. Xianwei, and W. Zhaoan, "Study on soft phase locked method to solving the synchronization problem of active power filter in stand-alone power grid," International Power Electronics and Motion Control Conference, IPEMC 2009, p. 1168-1174.

[5] G. Yalcinkaya, M. H. J. Bollen, and P. A. Crossley. Characterization of Voltage Sags in Industrial Distribution Systems. IEEE Transactions on Industry Applications, vol. 34, no. 4, p. 682, July/August 1998.

[6] J. Arrillaga, M. H. J. Bollen, and N. R. Watson. Power Quality Following Deregulation. Proceedings of the IEEE, vol. 88, no. 2, p. 246, February 2000.

[7] M.H.J. Bollen and E. Styvaktakis, "Characterization of Three-phase Unbalanced Dips (as easy as one-two-three ?),", Proceedings of the Power Engineering Society Meeting, vol. 2, p. 899, July 2000.

[8] H. H. J. Bollen. Algorithms for Characterizing Measured Three-Phase Unbalanced Voltage Dips. IEEE Transactions on Power Delivery, vol. 18 no. 3, July 2003, p. 937.

[9] J. Svensson. Synchronization Methods for Grid-Connected Voltage Source Converters. IEE Proceedings on General Transmission and Distribution, vol. 148, no. 3, p. 229, May 2001.

[10] M. Karimi-Ghartemani and M. R. Iravani. A Method for Synchronization of Power Electronic Converters in Polluted and Variable-Frequency

Environments. IEEE Transactions on Power Systems, vol. 19, no. 3, p. 1263, August 2004.

[11] S.-M. Kim and S.-K. Sul. Control of Rubber Tyred Gantry Crane with Energy Storage Based on Supercapacitor Bank. Proceedings of the Power Electronics Specialists Conference, PESC 2005, p. 262.

[12] J.-W. Choi, Y.-K. Kim, and H.-G. Kim. "Digital PLL control for single-phase photovoltaic system," IEE Proceedings of Electric Power Applications, vol. 153, no.1, p. 40-46, Jan. 2006.

[13] S.-J. Lee, J.-K. Kang, and S.-K. Sul. A New Phase Detecting Method for Power Conversion Systems. Proceedings of the Industry Applications Society conference, IAS 1999.

[14] Freijedo, F.D.; Doval-Gandoy, J.; Lopez, O.; Cabaleiro, J. Robust phase locked loops optimized for DSP implementation in power quality applications. 34th Annual Conference of IEEE Industrial Electronics, 2008. IECON 2008, p. 3052.

[15] O. Vainio, S. J. Ovaska, and M. Pöllä, "Adaptive Filtering Using Multiplicative General Parameters for Zero-Crossing Detection," IEEE Transactions on Industrial Electronics, vol. 50, no. 6, p. 1340, Dec. 2003.

[16] O. Vainio, S. J. Ovaska. Noise Reduction in Zero Crossing Detection by Predictive Filtering. IEEE Transactions on Industrial Electronics, vol. 42, no. 1, p. 58, February 1995.

[17] D. R.Costa, L.G. B. Rolim, and M. Aredes, "Analysis and Software Implementation of a Robust Synchronizing Circuit PLL Circuit," IEEE International Symposium on Industrial Electronics, ISIE 2003, vol. 1, p. 292-297.

[18] S. K. Chung, "A Phase Tracking System for Three Phase Utility Interface Inverters," IEEE Transactions on Power Electronics, vol. 15, no. 3, p. 431, May 2000.

[19] A. K. Ziarani and A. Konrad. A Nonlinear Adaptive Method of Elimination of Power Line Interference in ECG Signals. IEEE Transactions on Biomedical Engineering, vol. 49, no. 6, p. 540, June 2002.

[20] H. Karimi, M. Karimi-Ghartemani, and M. R. Irvani. Estimation of Frequency and its Rate of Change for Applications in Power Systems. IEEE Transactions on Power Delivery, vol. 19 no. 2, p. 472, April 2004.

[21] M. Karimi-Ghartemani and M. R. Iravani. A Signal Processing Module for Power System Applications. IEEE Transactions on Power Delivery, vol. 18 no. 4, p. 1118, October 2003.

[22] P. Rodriguez, J. Pou, J. Bergas, I. Candela, R. Burgos, and D. Boroyevich. Double Synchronous Reference Frame for Power Converters Control. PESC 2005, p. 1415.

[23] P. Rodríguez, J. Pou, J. Bergas, J. I. Candela, R. P. Burgos, and D. Boroyevich. Decoupled Double Synchronous Reference Frame PLL for Power Converters Control. IEEE Transactions on Power Electronics, vol. 22, no. 2, p. 584, March 2007.

[24] P. Rodríguez, A. Luna, M. Ciobotaru, R. Teodorescu, and F. Blabbjerg. Advanced Grid Synchronization System for Power Converters under Unbalanced and Distorted Operating Conditions. IECON 2006, p. 5173.

[25] P. Rodriguez, R. Teodorescu, I. Candela, A. V. Timbus, M. Liserre, and F. Blabbjerg. New Positive-Sequence Voltage Detector for Grid Synchronization of Power Converters under Faulty Grid Conditions. PESC 2006.

[26] P. Rodriguez, A. V. Timbus, R. Teodorescu, M. Liserre, and F. Blaabjerg, "Flexible Active Power Control of Distributed Power Generation Systems During Grid Faults," IEEE Transactions on Industrial Electronics, vol. 54, no. 5, Oct. 2007, p.2583-2592.

[27] M. Ciobotaru, R. Teodorescu, and V. Agelidis, "Offset rejection for PLL based synchronization in grid-connected converters," Applied Power Electronics Conference and Exposition, APEC 2008, p. 1611-1617.

[28] R. W. Wall. Simple Methods for Detecting Zero Crossing. Industrial Electronics Conference, IECON 2003, p. 2477.

[29] J. Sun. On the Zero-Crossing Distortion in Single-Phase PFC Converters. IEEE Transactions on Power Electronics, vol. 19, no. 3, May 2004, p. 685.

[30] E. Jacobsen and R. Lyons. The Sliding DFT. IEEE Signal Processing Magazine, vol. 20 no. 2, p. 74, March 2003.

[31] M. Karimi-Ghartemani and A. K. Ziarini. A Nonlinear Time-Frequency Analysis Method. IEEE Transactions on Signal Processing, vol. 52 no. 6, June 2004, p. 1585.

[32] P. Mattavelli, and F. P. Marafão, "Repetitive-Based Control for Selective Harmonic Compensation in Active Power Filters," IEEE Transactions on Industrial Electronics, vol. 51, no. 5, p. 1118, October 2004.

[33] B. P. McGrath, D. G. Holmes, and J. J. H. Galloway, "Power Converter Line Synchronization Using a Discrete Fourier Transform (DFT) Based on a Variable Sample Rate," IEEE Transactions on Power Electronics, vol. 20, n. 4, p. 877, July 2005.

[34] V. M. Moreno, M. Liserre, A. Pigazo, and A. Dell'Aquila, "A comparative analysis of real-time algorithms for power signal decomposition in multiple synchronous reference frames," IEEE Trans. Power Electron., vol. 22, no. 4, pp. 1280–1289, Jul. 2007.

[35] G.-C. Hsieh and J. C. Hung. Phase-Locked Loop Techniques - A Survey. IEEE Transaction on Industrial Electronics, vol. 43, no. 6, Dec. 1996.

[36] J. Smith. Modern Communication Circuits. 2nd ed. McGraw-Hill. 1998.

[37] M. Gardner. Phaselock Techniques. 3rd ed. J. Wiley & Sons. 2005.

[38] G. Nash. Phase-Locked Loop Design Fundamentals. Motorola Application Note AN535, 1995.

[39] R. Teodorescu and F. Blaabjerg, "Flexible Control of Small Wind Turbines With Grid Failure Detection Operating in Stand-Alone and Grid-Connected Mode," IEEE Transactions on Power Electronics, vol. 19 no. 5, p. 1323, Sept. 2004.

[40] H. E. P. de Souza, F. Bradaschia, F. A. S. Neves, M. C. Cavalcanti, G. M. S. Azevedo, and J. P. de Arruda, "A Method for Extracting the Fundamental-Frequency Positive-Sequence Voltage Vector Based on Simple Mathematical Transformations," IEEE Transactions on Industrial Electronics, vol. 56, no. 5, May 2009, p. 1539.

[41] H. Awad, J. Svensson, and M. J. Bollen, "Tuning Software Phase-Locked Loop for Series-Connected Converters," IEEE Transactions on Power Delivery, vol. 20, no. 1, p. 300, Jan. 2005.

[42] C. A. Busada, H. G. Chiacchiarini, and J. C. Balda, "Synthesis of Sinusoidal Waveform References Synchronized With Periodic Signals," IEEE Transactions on Power Electronics, vol. 23, no. 2, March 2008.

[43] G. Zhou, X. Shi, C. Fu, and Y. Wang, "Operation of a Three-phase Soft Phase Locked Loop Under Distorted Voltage Conditions Using Intelligent PI Controller," TENCON 2006, November 2006.

[44] V. Kaura and V. Blasko, "Operation of a Phase Locked Loop System Under Distorted Utility Conditions," IEEE Transactions on Industry Applications, vol. 33, no. 1, January/February 1997.

[45] L.N. Arruda, S. M. Silva, and B. J. C. Filho, "PLL Structures for Utility Connected Systems," 36th Industry Applications Society Annual Meeting, IAS 2001, vol.4, p. 2655-2660.

[46] S. M. Silva, B. M. Lopes, B. J. C. Filho, R. P. Campana, and W. Boaventura, "Performance Evaluation of PLL Algorithms for Single-Phase Grid-connected Systems," Proceedings of the Industrial Applications Society Conference 2004, vol. 4, p. 2259.

[47] L. G. B. Rolim, D. R. da Costa, and M. Aredes, "Analysis and Software Implementation of a Robust Synchronizing PLL Circuit Based on the pq Theory," IEEE Transactions on Industrial Electronics, vol. 53, no. 6, p. 1919, December 2006.

[48] R. M. S. Filho, P. F. Seixas, P. C. Cortizo, L. A. B. Torres, A. F. Souza, "Comparison of three single-phase PLL algorithms for UPS applications," IEEE Transactions on Industrial Electronics, p. 2923, August 2008.

[49] M. A. Perez, J. R. Espinoza, L. A. Moran, M. A. Torres, and E. A. Araya, "A Static-Power Converters With Polluted AC Systems," IEEE Transactions on Industrial Electronics, vol. 55, no. 5, p. 2185, May 2008.

[50] F. Blaabjerg, R. Teodorescu, M. Liserre, and A. V. Timbus, "Overview of Control and Grid Synchronization for Distributed Power Generation

Systems," IEEE Transactions on Industrial Electronics, vol. 53, no. 5, p. 1398, October 2006.

[51] E. J. Bueno, F. J. Rodriguez, F. Espinosa, S. Cobreces, "SPLL design to flux oriented of a VSC interface for wind power applications," IEEE Industrial Electronics Society Conference, IECON 2005, p. 2451.

[52] T. Ostrem, W. Sulkowski, L. Norum, W. Yadong "Phase-Locked Loop with Adaptive Signal Cancellation for Three-phase Network Side Voltage Source Inverter". Power Electronics and Applications, 2007 European Conference, p. 1.

[53] X. Yuan, J. Allmeling W. Merk, and H. Stemmler. "Stationary Frame Generalized Integrators for Current Control of Active Power Filters with Zero Steady State Error for Current Harmonics of Concern under Unbalanced and Distorted Operation Conditions," Proceedings of the Industrial Applications Society Conference 2000, vol. 4, p. 2143.

[54] X. Yuan, W. Merk, H. Stemmler, and J. Allmeling. Stationary Frame Generalized Integrators for Current Control of Active Power Filters with Zero Steady State Error for Current Harmonics Under Balanced and Distorted Operating Conditions. IEEE Transactions on Industrial Applications, vol. 38 no. 2, p. 523-532, March/April 2002.

[55] J. Allmeling, "A Control Structure for Fast Harmonics Compensation in Active Filters," IEEE Transactions On Power Electronics, vol. 19, no. 2, p. 508, March 2004.

[56] P. A. Regalia. An Improved Lattice-Based Adaptive IIR Notch Filter. IEEE Transactions on Signal Processing, vol. 39 no. 9, September 1991.

[57] L. Hsu, R. Ortega, and G. Damm. A Globally Convergent Frequency Estimator. IEEE Trans. on Automatic Control, vol. 44, no. 4, April 1999.

[58] M. Mojiri and A. R. Bakhshai, "An Adaptive Notch Filter for Frequency Estimation of a Periodic Signal," IEEE Transactions on Automatic Control, vol. 49, no. 2, p. 314, February 2004.

[59] M. Karimi-Ghartemani, M. Mojiri and A. R. Bakhshai. A Technique for Extracting Time-Varying Harmonics Based on an Adaptive Notch Filter.

Proceedings of the 2005 IEEE Conference on Control Applications, p. 624-628, August 2005.

[60] M. Mojiri, M. Karimi-Ghartemani, and A. Bakhshai, "Time-Domain Signal Analysis Using Adaptive Notch Filter," IEEE Transactions on Signal Processing, vol. 55, no. 1, p. 85, Jan. 2007.

[61] M. Mojiri and A. R. Bakhshai, "Estimation of n Frequencies Using Adaptive Notch Filter," IEEE Transactions on Circuits and Systems—II: Express Briefs, vol. 54, no. 4, p. 338, April 2007.

[62] D. Yazdani, A. Bakhshai, G. Joos, and M Mojiri. A Nonlinear Adaptive Synchronization Technique for Grid-Connected Distributed Energy Sources. Letters, IEEE Transactions on Power Electronics, vol. 23 no. 4, p. 2181, July 2008.

[63] D. Yazdani, M. Mojiri, A. Bakhshai, G. Joos. A Fast and Accurate Synchronization Technique for Extraction of Symmetrical Components. IEEE Transactions on Power Electronics, vol. 24 no. 3, March 2009.

[64] D. Yazdani, A.Bakhshai, and P. K. Jain, "A Three-Phase Adaptive Notch Filter Based Approach to Harmonic/Reactive Current Extraction and Harmonic Decomposition," accepted for publication in IEEE Transactions on Power Electronics 2010.

[65] D. Yazdani, A.Bakhshai, and P. K. Jain, "A Three-Phase Adaptive Notch Filter Based Approach to Harmonic/Reactive Current Extraction and Harmonic Decomposition," accepted for publication in IEEE Transactions on Power Electronics 2010.

[66] S. Sastry and M. Bodson. Adaptive Control, Stability – Convergence and Robustness. Prentice-Hall 1989, at: www.ece.utah.edu/~bodson/acsrcr.

[67] J.-J. E. Slotine and and W. Li. Applied Nonlinear Control. Prentice-Hall, 1989.

[68] K. Astrom and B. Wittenmark, Adaptive Control, Addison-Wesley, 1995.

[69] H. K. Khalil. Nonlinear Systems, 3rd ed. Prentice-Hall, 2002.

[70] P. Rodriguez, R. Teodorescu, I. Candela, A. V. Timbus, M. Liserre, and F. Blabbjerg. New Positive-Sequence Voltage Detector for Grid

Synchronization of Power Converters under Faulty Grid Conditions. PESC 2006.

[71] M. Ciobotaru, R. Teodorescu, and F. Blaabjerg. A New Single-Phase PLL Structure Based on Second Order Generalized Integrator. PESC 2006.

[72] M. R. Iravani and M. Karimi-Ghartemani. Online Estimation of Steady State and Instantaneous Symmetrical Components. IEE Proceedings – Generation, Transmission, and Distribution, vol. 150, no. 5, Sept. 2003.

[73] B. A. Francis and W. M. Wonham, "The Internal Model Principle of Control Theory," Automatica, vol. 12, no. 5, p. 457-465, 1976.

[74] S. Luo and Z. Hou. An adaptive Detecting Method for Harmonic and Reactive Currents. IEEE Transaction on Industrial Electronics, vol. 42, no. 1, p. 85, February 1995.

[75] M. Bodson, A. Sacks, and P. Khosla, "Harmonic generation in adaptive feedforward cancellation schemes," IEEE Transactions on Automatic Control, vol. 39, no. 9 p. 1939-1944, Sept. 1994.

[76] A. Sacks, M. Bodson, and P. Khosla, "Experimental Results of Adaptive Periodic Disturbance Cancellation in a High Performance Magnetic Disk Drive," Journal of Dynamics Systems, Measurement, and Control, Transactions of the ASME, vol. 118, p. 416-424, Sept. 1996.

[77] H. Karimi, M. Karimi-Ghartemani, M. R. Iravani, and A. R. Bakhshai, "An Adaptive Filter for Synchronous Extraction of Harmonics and Distortions," IEEE Trans. on Power Delivery, vol. 18 no. 4, Oct. 2003.

[78] S. Pigg and M. Bodson. Adaptive Rejection of Sinusoidal Disturbances of Known Frequency Acting on Unknown Systems. Proceedings of the 2006 American Control Conference, p. 4777.

[79] M. Bodson and S. C. Douglas, "Adaptive Algorithms for the Rejection of Sinusoidal Disturbances with Unknown Frequency," Automatica, vol. 33, no. 12, p. 2213-2221, December 1997.

[80] B. Wu and M. Bodson. A Magnitude/Phase Locked-Loop Approach to Parameter Estimation of Periodic Signals. IEEE Transactions on Automatic Control, vol. 48, no. 4, April 2003, p. 612.

[81] A. K. Ziarani and M. Karimi-Ghartemani, “On the Equivalence of Three Independently Developed Phase-Locked Loops,” IEEE Transactions On Automatic Control, vol. 50, no. 12, p. 2021, December 2005.

[82] M. Karimi-Ghartemani and A. K. Ziarani. Periodic orbit analysis of two dynamical systems for electrical engineering applications. Journal of Engineering Mathematics, 2003, p. 135–154.

[83] X. Guo and Marc Bodson, “Equivalence between adaptive feedforward cancellation and disturbance rejection using the internal model principle,” Int. J. Adapt. Control Signal Process. 2010, p. 211–218, April 2009.

[84] A. Antoniou, “Digital Filters – Analysis, Design, and Applications”, 2nd ed., McGraw-Hill, 1993.

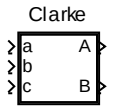
APPENDIX A – SIMULINK BLOCK LIBRARY

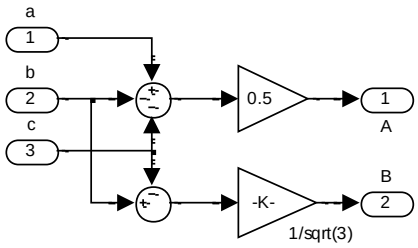
The following appendix lists Simulink building blocks defined for various PLL development and signal generation functions.

A.1 SIGNAL TRANSFORMATIONS

A.1.1 Three-Phase abc to Two-Phase αβ Clarke Transformation

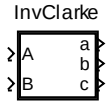
Name: Clarke			
Function:	Linear transformation of 3-phase abc to 2-phase αβ signals		
Inputs:		Outputs:	
a	3-phase a component	A	Equivalent to α
b	3-phase b component	B	Equivalent to β
c	3-phase c component		
Block symbol:	Internal schematic:		

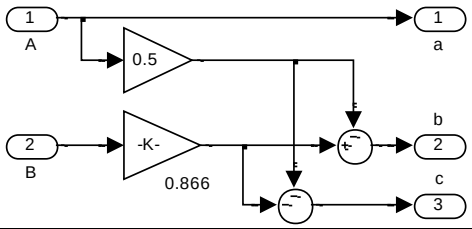




A.1.2 2-Phase to 3-Phase Inverse Clarke transformation

Name: InvClarke			
Function:	Linear transformation of 2-phase αβ to 3-phase abc signals		
Inputs:		Outputs:	
A	α component of quadrature signal	a	3-phase a component
B	β component of quadrature signal	b	3-phase b component
		c	3-phase c component
Block symbol:	Internal schematic:		

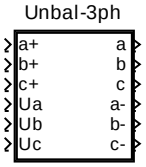




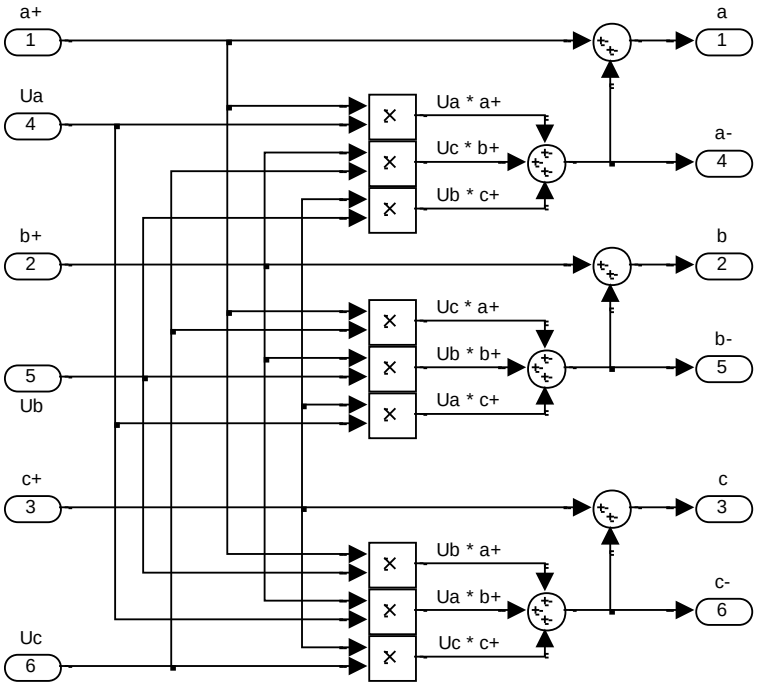
A.1.3 3-Phase Un-balance

Name: Unbal_3ph			
Function:	Generates unbalanced 3-phase signal, from balanced abc signal		
Inputs:		Outputs:	
a+, b+, c+	balanced 3-phase signals	a, b, c	un-balanced abc signals
Ua	un-balance factor, in pu, for type C_A	a-, b-, c-	negative sequence abc signals
Ub, Ub	Same, for type C_B and C_C		

Block symbol:

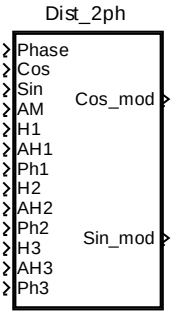


Internal schematic:

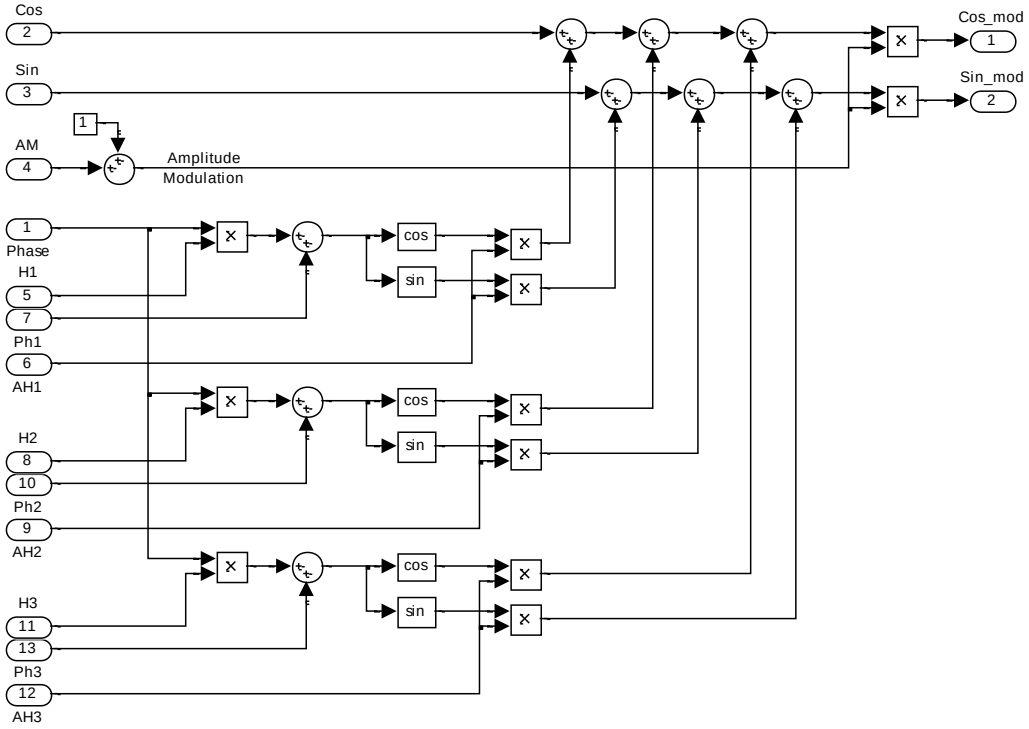


A.1.4 Two-phase distortion adder

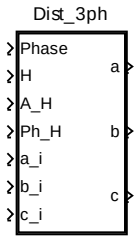
Name: Dist_2ph			
Function:	Adds up to three harmonic components to 2-phase signal		
Inputs:		Outputs:	
Phase	Phase of fundamental signal	Cos_mod	Fund. cos plus harmonics
Cos	Reference cos signal	Sin_mod	Fund. sin plus harmonics
Sin	Reference sin signal		
AM	Amplitude modulation control		
H1	Index of harmonic to add to above reference signals	Block symbol:	
AH1	Amplitude, in pu, of harmonic to add to above reference signals		
Ph1	Phase of harmonic to add to above reference signals		
H2, AH2, Ph2	Same as above, for a second harmonic component		
H3, AH3, Ph3	Same as above, for a third harmonic component		



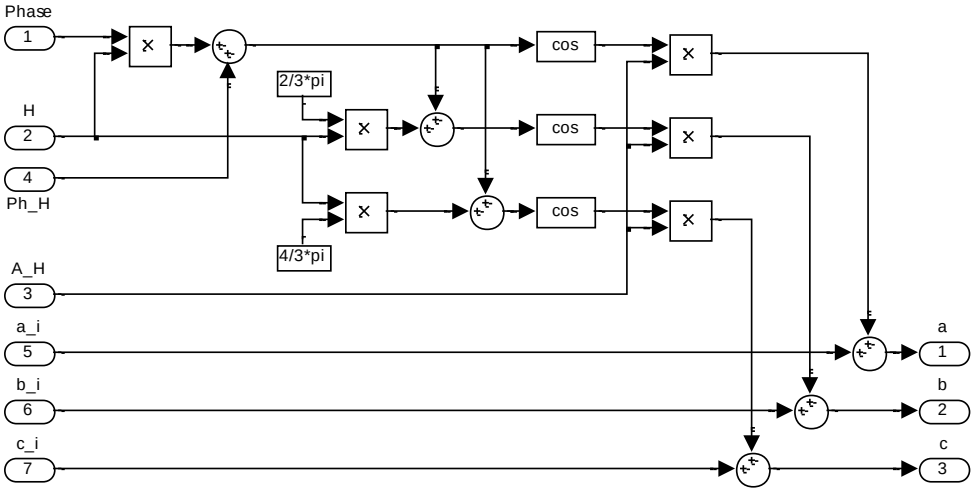
Internal schematic:



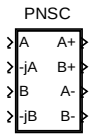
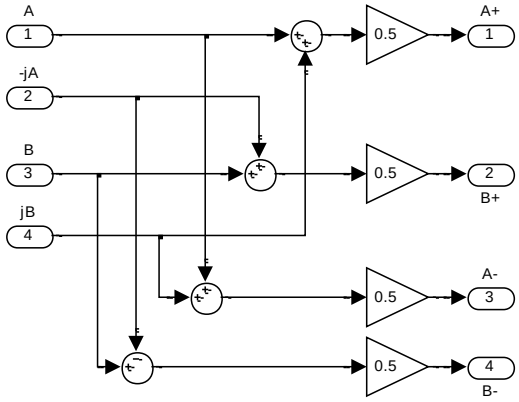
A.1.5 Three-phase distortion adder

Name: Dist_3ph			
Function:	Adds one harmonic component to 3-phase signal		
Inputs:		Outputs:	
Phase	Phase of fundamental	a	= a_i + harmonic
H	Index of harmonic to be added	b	= b_i + harmonic
A_H	Harmonic amplitude	c	= c_i + harmonic
Ph_H	Phase of harmonic		
a_i	3-phase a input		
b_i	3-phase b input		
c_i	3-phase c input		
Block symbol:			

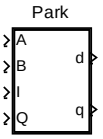
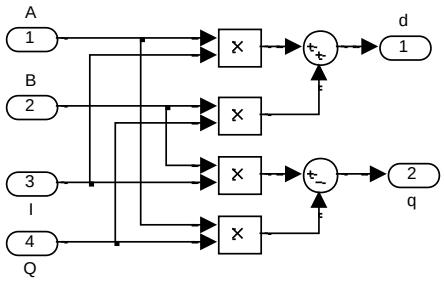
Internal schematic:



A.1.6 Positive Negative Sequence Calculation (PNSC)

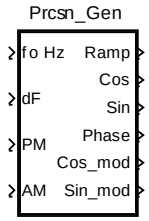
Name: PNSC			
Function:	Calculates positive and negative αβ sequence signals from un-balanced α and β signals and their phase shifted versions –jα and –jβ		
Inputs:		Outputs:	
A	α component of un-balanced αβ signal	A+	Positive sequence α signal
–jα	Same, shifted -90^o	B+	Positive sequence β signal
B	β component of un-balanced αβ signal	A-	Negative sequence α signal
–jβ	Same, shifted +90^o	B-	Negative sequence β signal
Block symbol: 	Internal schematic: 		

A.1.7 Park Transformation

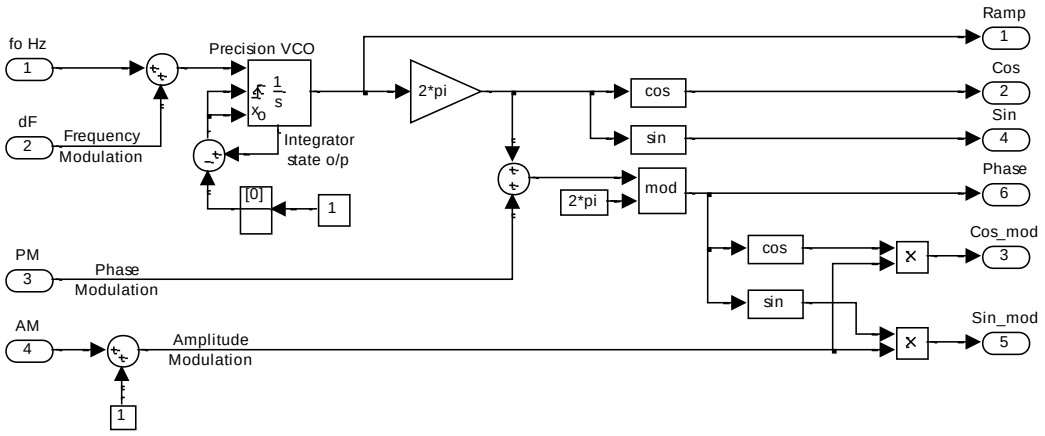
Name: Park			
Function:	Generates d and q signals from input signal $\mathbf{v}_{AB}$ and VCO IQ signals		
Inputs:		Outputs:	
A	α component of αβ signal	d	= V_{AB} cos(ΔAngle)
B	β component of αβ signal	q	= V_{AB} sin(ΔAngle)
I	I (cos) VCO signal		
Q	Q (sin) VCO signal		
Block symbol: 	Internal schematic: 		

A.2 SIGNAL GENERATION

A.2.1 Elementary precision frequency signal generator: ramp/cos/sin, with FM, PM, and AM

Name: Prcsn_Gen			
Function:	Generates frequency controlled signals, including unit magnitude phase ramp, quadrature sinusoids, and phase and amplitude modulated phase and quadrature sinusoidal signals		
Inputs:		Outputs:	
fo Hz	frequency, in Hz	Ramp	unit amplitude ramp signal, only frequency modulated
dF	Freq. modulation, deviation from fo	Cos	reference signal, only frequency modulated
PM	Phase Modulation	Sin	reference signal, only frequency modulated
AM	Ampl. modulation; output amplitude = 1 + this value	Cos_mod	modulated Cos signal
		Sin_mod	modulated Sin signal
		Phase	Phase modulated ramp signal
Block symbol:			

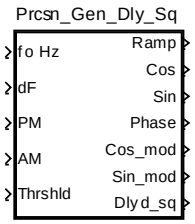
Internal schematic:



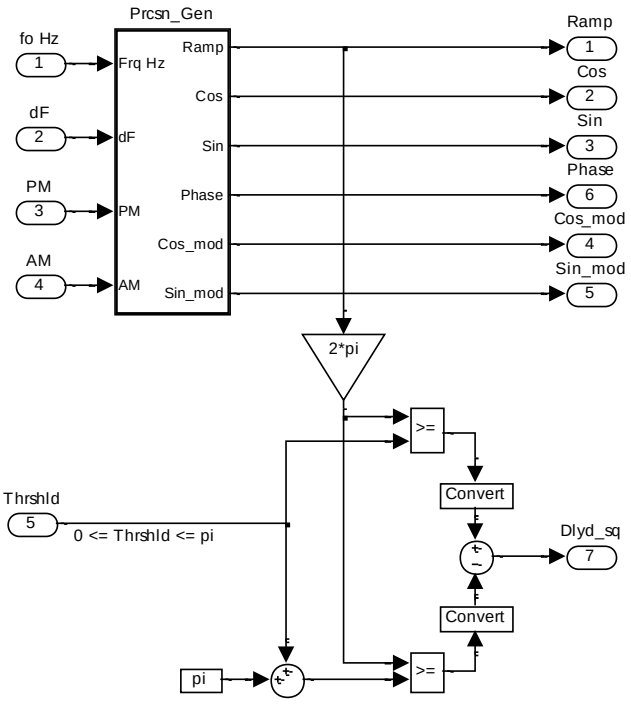
A.2.2 Precision frequency signal generation with delayed square wave

Name: Prcsn_Gen_Dly_Sq			
Function:	Same as Prcsn_Gen, plus square signal with controlled phase		
Inputs:		Outputs:	
Frq Hz	frequency, in Hz	Ramp	unit amplitude ramp signal, only frequency modulated
dF	Freq. modulation, deviation from fo	Cos	reference signal, only frequency modulated
PM	Phase Modulation	Sin	reference signal, only frequency modulated
AM	Ampl. modulation; output amplitude = 1 + this value	Cos_mod	modulated Cos signal
Thrshld	Threshold phase for square output toggle, in radians	Sin_mod	modulated Sin signal
		Phase	Phase modulated ramp signal

Block symbol:



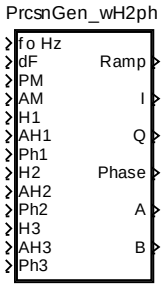
Internal schematic:



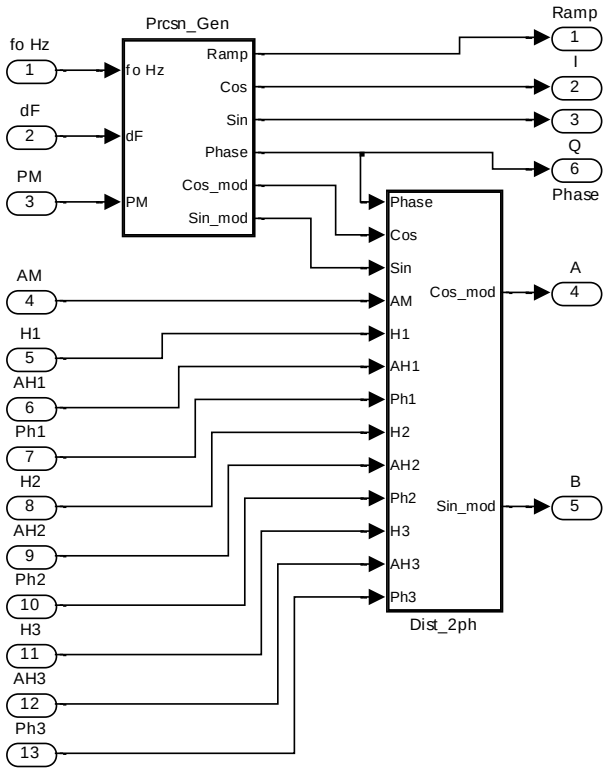
A.2.3 Precision frequency generator with harmonic distortion

Name: PrcsnGen_wH2ph-II			
Function:	Same as Prcsn_Gen, plus square signal with controlled phase		
Inputs:		Outputs:	
fo Hz	frequency, in Hz	Ramp	Reference phase (un-modulated) ramp signal
dF	Freq. modulation, deviation from fo	I	Reference cos signal, only frequency modulated
PM	Phase Modulation	Q	Reference sin signal, only frequency modulated
AM	Ampl. modulation; output amplitude = 1 + this value	Phase	Phase modulated phase signal
H1, H2, H3	Indices of 3 harmonics to add to fundamental	A	Output α signal
AH1-AH3	Amplitudes of each harmonic added	B	Output β signal
Ph1-Ph3	Phases of each harmonic added		

Block symbol:



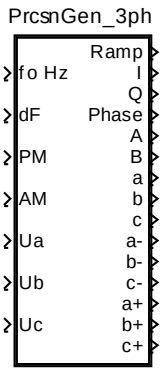
Internal schematic:



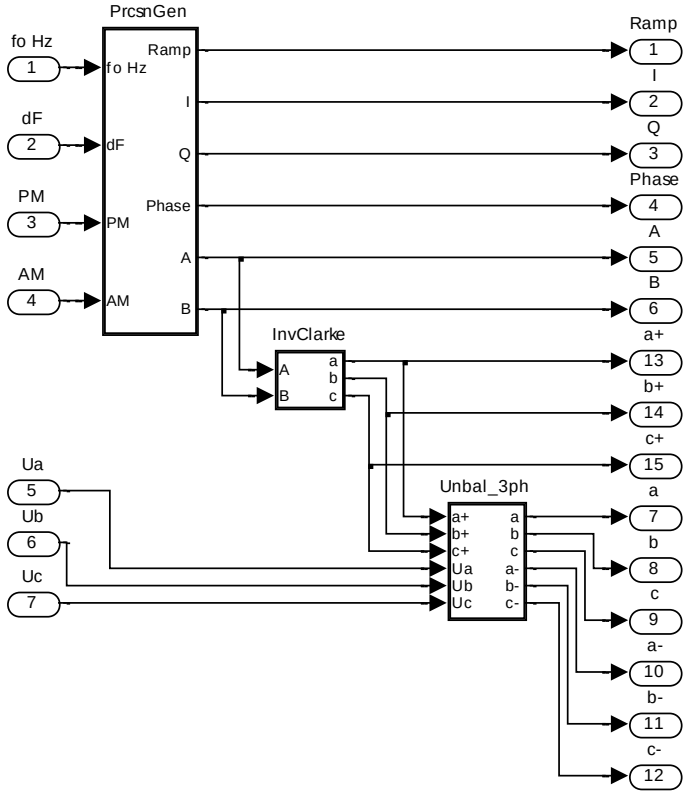
A.2.4 Precision frequency un-balanced 3-phase generator

Name: PrcsnGen_3ph			
Function:	Three-phase un-balanced signal generator		
Inputs:		Outputs:	
fo Hz	frequency, in Hz	Ramp	Reference phase signal, only frequency modulated
dF	Freq. modulation, deviation from fo	I	Reference cos signal, only frequency modulated
PM	Phase Modulation	Q	Reference sin signal, only frequency modulated
AM	Ampl. modulation; output amplitude = 1 + this value	Phase	Phase modulated phase signal
Ua-Uc	Un-balance factors, in pu, for types C_A, C_B, and C_C	A and B	Modulated α and β signals
		a+ to c+	Positive seq. 3-phase signal
		a- to c-	Negative seq. 3-ph. signal
		a to c	Un-balanced 3-phase signal

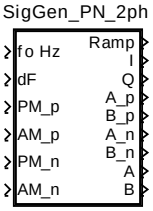
Block symbol:



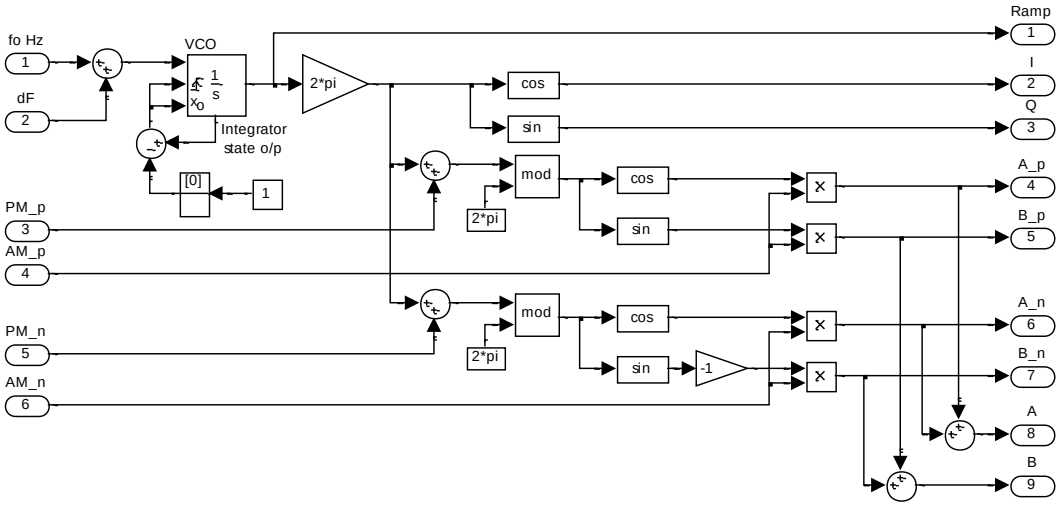
Internal schematic:



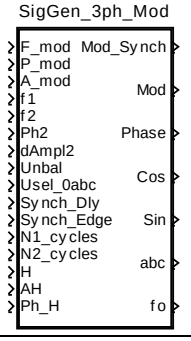
A.2.5 Precision frequency 2-phase generator, with separate positive and negative sequence amplitude and phase control

Name: SigGen_PN_2ph			
Function:	Controlled frequency quadrature signal generator with separate positive and negative sequence amplitude and phase control		
Inputs:		Outputs:	
fo Hz	Frequency, in Hz	Ramp	Un-modulated phase ref., only frequency modulated
dF	Deviation from fo	I	Un-modulated cos ref.
PM_p	Phase modulation for positive sequence	Q	Un-modulated sin ref.
AM_p	Amplitude modulation for positive sequence	A_p	Positive sequence α output
PM_n	Phase mod. for neg. seq.	B_p	Positive sequence β output
AM_n	Ampl. mod. for neg. seq.	A_n	Negative sequence α output
		B_n	Negative sequence β output
		A	Un-balanced α output
		B	Un-balanced β output
Block symbol:			

Internal schematic:

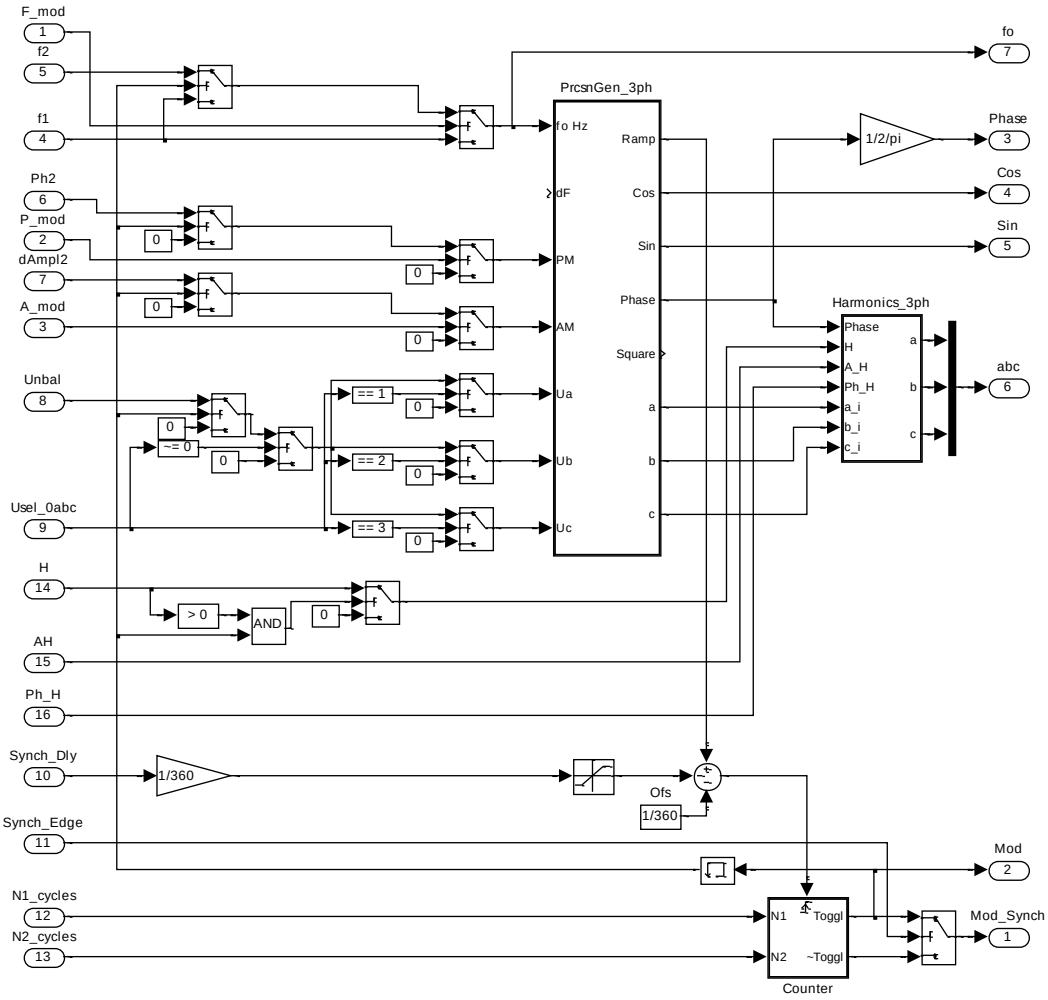


A.2.6 Precision frequency 3-phase generator, with built-in auto-modulation

Name: SigGen_3ph_Mod			
Function:	Generates 3-phase signal, with FM, PM, AM, un-balance, and distortion modulation, with square stepping modulation		
Inputs:		Outputs:	
F_mod	1 = Enables FM	Mod_synch	Mod edge for external scope triggering
P_mod	1 = Enables PM	Mod	Modulation state
A_mod	1 = Enables AM	Phase	Modulated signal phase
f1	Signal frequency when Mod = 0	Cos	Frequency, Phase, and Amplitude modulated cos reference
f2	Signal frequency when Mod = 1	Sin	Frequency, Phase, and Amplitude modulated sin reference
Ph2	Signal phase when Mod = 0	NOTE: the above Cos and Sin reference signals are balanced	
dAmpl2	Signal ampl. increment when Mod = 0	abc	3-phase vector signal
Unbal	Amount of unbalance, pu	fo	Signal frequency
Usel_abc	Select no un-balance, or type C_A, C_B, or C_C		
Synch_Dly	Delay threshold for modulation toggle		
Synch_Edge	Select Mod edge to toggle between signal modulation states		
N1_cycles, N2_cycles	Number of cycles when Mod = 0 / 1, respectively		
H	Index of harmonic to add		
AH, Ph_H	Amplitude and phase of harmonic component		
Block symbol:			

Internal schematic: next page

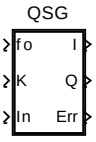
SigGen_3ph_Mod – Internal schematic:

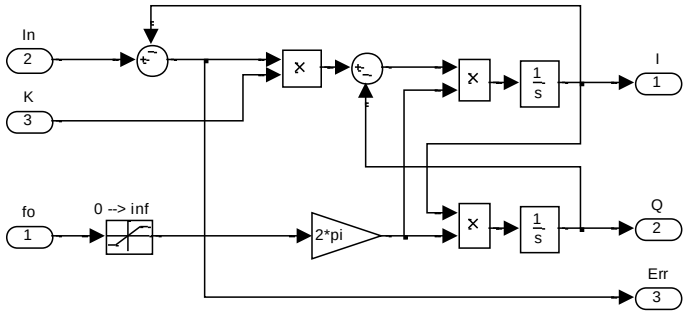


A.3 **FILTERS**

A.3.1 Controlled frequency single-phase quadrature signal generator/band/low-pass filter

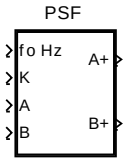
Name: QSG			
Function:	Implements resonator-based quadrature signal generator / low-pass / band-pass / notch filter		
Inputs:		Outputs:	
fo	Filter center frequency	I	In-phase, band-pass output
K	Filter K factor = 2ζ	Q	Low-pass output, in quadrature with input
In	Single-phase input signal	Err	Notch filter output
Block symbol:	Internal schematic:		

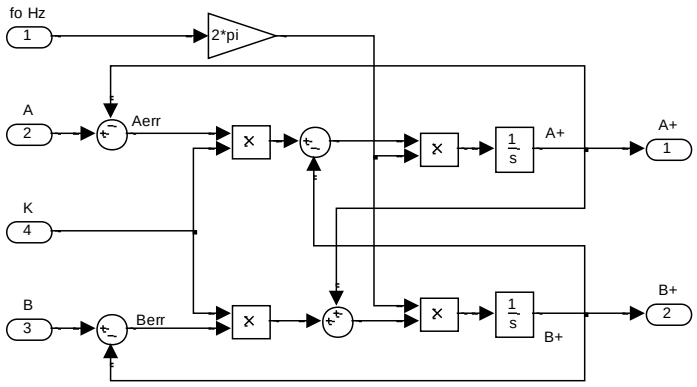




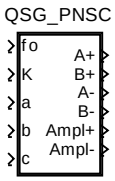
A.3.2 Positive sequence filter (PSF)

Name: PSF			
Function:	Implements resonator-based positive sequence filter		
Inputs:		Outputs:	
fo	Filter center frequency	A+, B+	Filtered αβ pos. seq. output
K	Filter K factor		
A, B	Quadrature-phase αβ input signal		
Block symbol:	Internal schematic:		

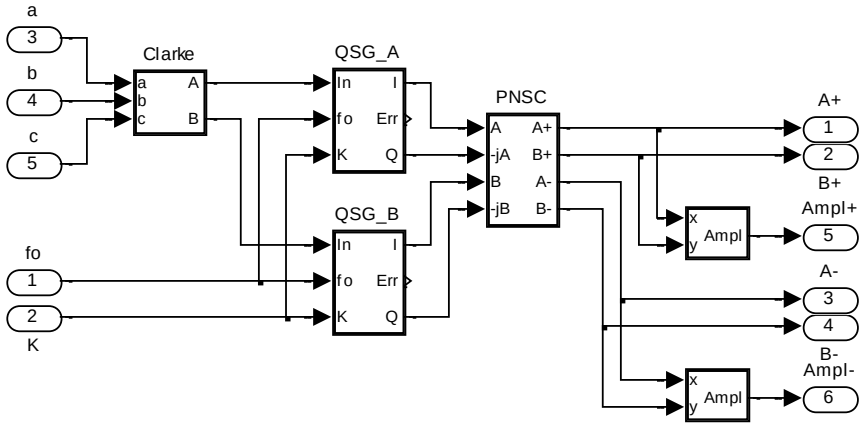




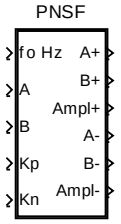
A.3.3 QSG-based Positive-Negative Sequence Calculation

Name: QSG-PNSC			
Function:	Implements QSG-based positive-negative sequence calculation		
Inputs:		Outputs:	
fo	Filter center frequency	A+, B+	Positive seq. αβ signal
K	Filter K factor	Ampl+	Positive seq. amplitude
a, b, c	Three-phase signals, un-balanced	A-, B-	Negative seq. αβ signal
		Ampl-	Negative seq. amplitude
Block symbol: 			

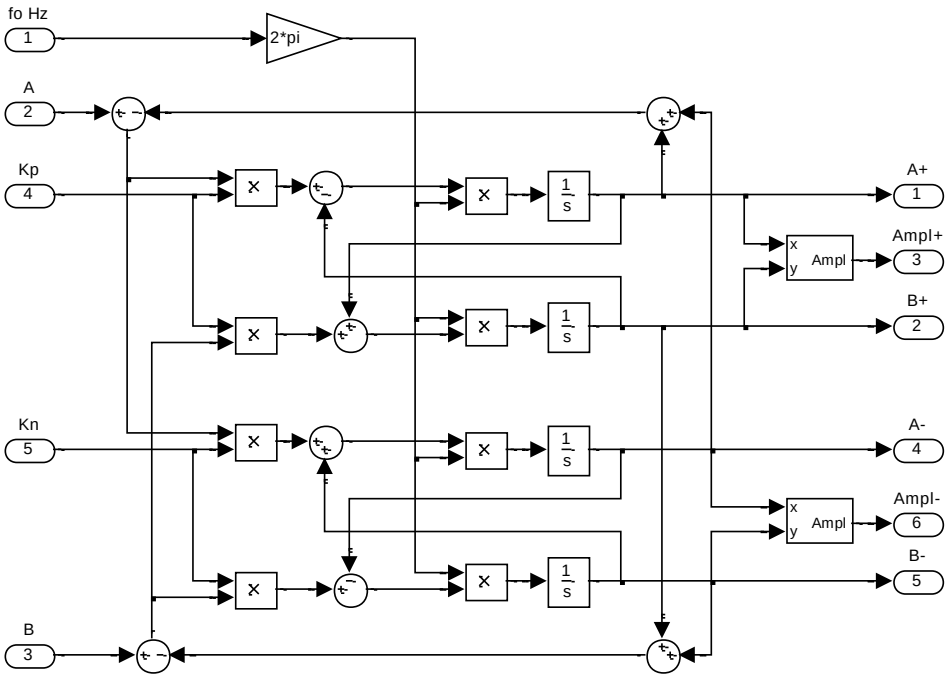
Internal schematic:



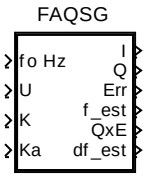
A.3.4 Positive-Negative Sequence Filter (PNSF)

Name: PNSF			
Function:	Performs positive-negative sequence extraction		
Inputs:		Outputs:	
fo	Filter center frequency	A+, B+	Positive seq. αβ signal
A, B	Three-phase signals, un-balanced	Ampl+	Positive seq. amplitude
Kp	Pos. seq. filter K factor	A-, B-	Negative seq. αβ signal
Kn	Neg. seq. filter K factor	Ampl-	Negative seq. amplitude
Block symbol:			

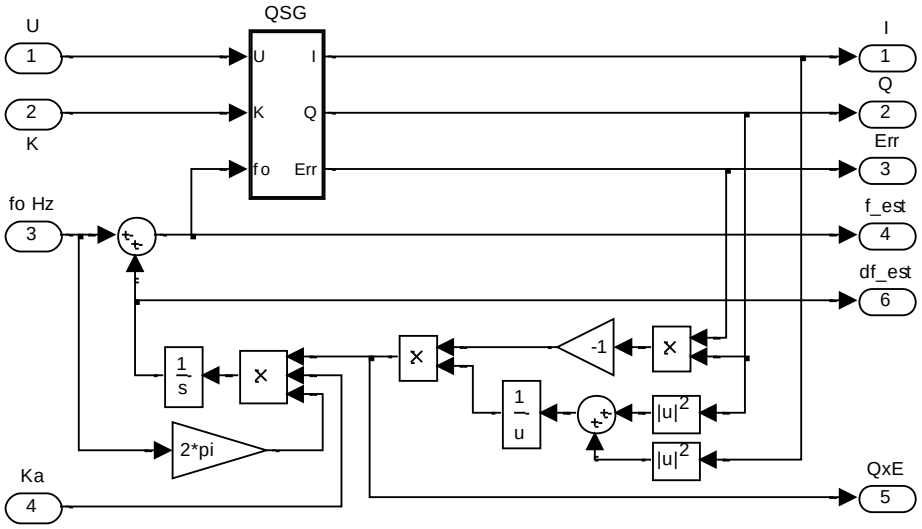
Internal schematic:



A.3.5 Frequency Adaptive Quadrature Signal Generator

Name: FAQSG			
Function:	Performs quadrature signal generation, with low-pass, band-pass, and notch filtering, while tracking input signal frequency		
Inputs:		Outputs:	
fo Hz	Filter center frequency	I, Q	In-phase and quadrature-phase outputs
U	Single-phase input	Err	= U – I
K	Filter K factor	f_est	Estimated input signal frequency
Ka	Frequency estimation adaptation gain	df_est	Deviation of f_est from fo
		QxE	Product of –Q and Err = correction for freq. estimate
Block symbol:			

Internal schematic:

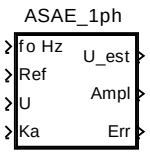


A.4 ADAPTIVE SYNCHRONOUS AMPLITUDE ESTIMATION

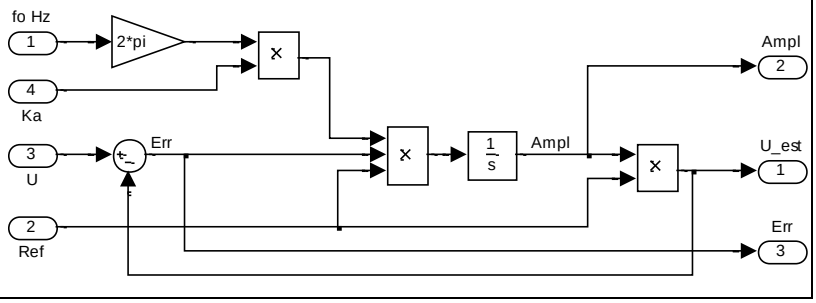
A.4.1 Single-Phase ASAE for Known Phase

NAME: ASAE_1ph			
Function:	Estimates amplitude of single-phase, phase-synchronized, sinusoid		
Inputs		Outputs	
fo Hz	freq. in Hz; serves to set time scale for adaptation	U_est	Estimate of input signal U
Ref	Ref. sinusoid, phase-synch'd to input U	Ampl	Estimated amplitude
U	input sinusoid with unknown amplitude	Err	Estimate error
Ka	Amplitude estimation adaptation gain		

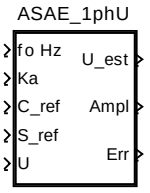
Block symbol:



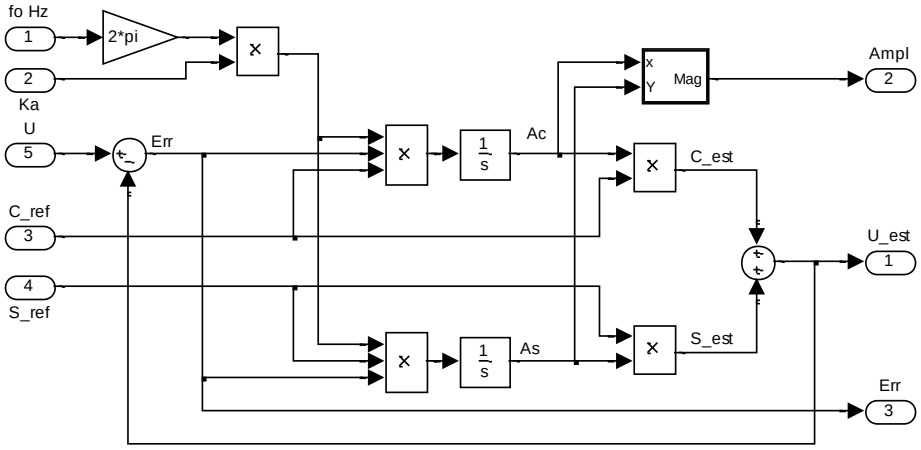
Internal schematic:



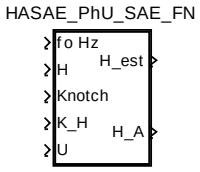
A.4.2 Single-Phase ASAE for Unknown Phase

NAME: ASAE_1phU			
Function:	Estimates amplitude of single-phase, frequency-synchronized, sinusoid		
Inputs		Outputs	
fo Hz	freq. in Hz; sets time scale for adaptation	U_est	Estimate of input signal U
Ka	Adaptation gain	Ampl	Estimated amplitude
C_ref	Ref. cos, frequency-synch'd to input U	Err	Estimate error
S_ref	Ref. sin, frequency-synch'd to input U	Block symbol: 	
U	Input sinusoid with unknown amplitude and phase		

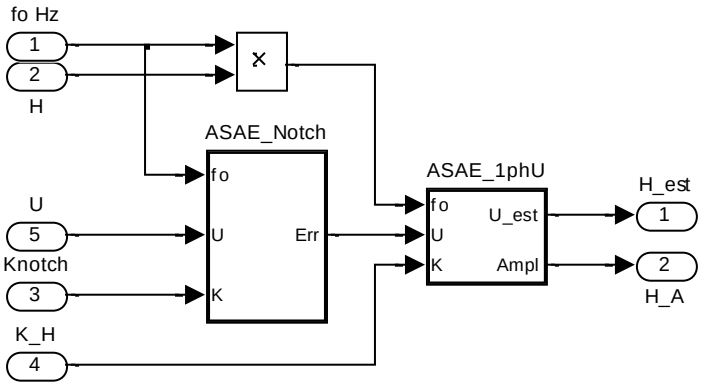
Internal schematic:



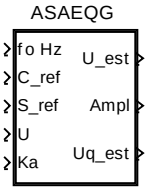
A.4.3 ASAE-based Harmonic Filter

NAME: HASAE_PhU_SAE_FN			
Function:	ASAE-based harmonic estimation, with ASAE-based fundamental notch pre-filtering		
Inputs		Outputs	
fo Hz	freq. in Hz; sets time scale for adaptation	U_est	Estimate of input signal U
K_notch	K for notch filter	H_est	
H	Harmonic index	Ampl	Estimated amplitude
K_H	K for harmonic filter	Block symbol:	
U	Input		

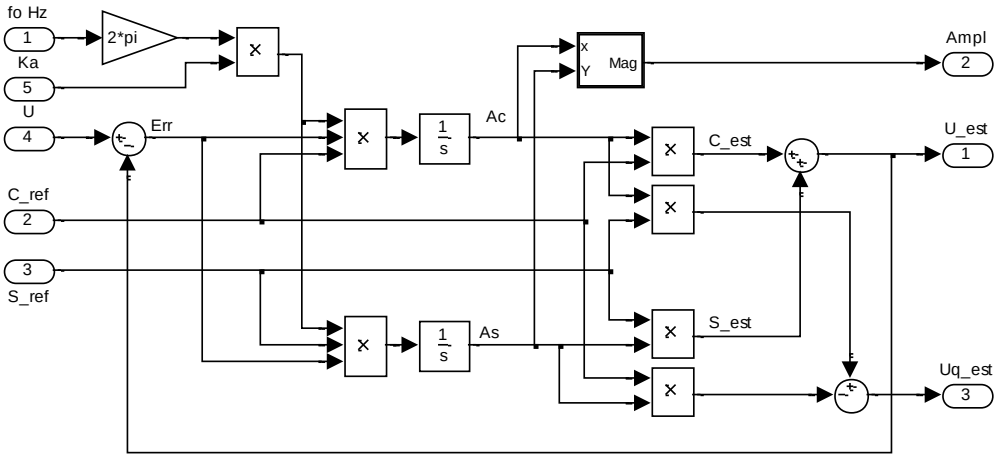
Internal schematic:



A.4.4 Single-Phase ASAE-based Quadrature Signal Generator

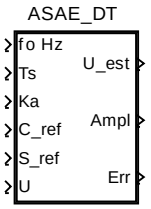
NAME: ASAEQG			
Function:	Performs quadrature signal generation of single-phase input using ASAE technique		
Inputs		Outputs	
fo Hz	freq. in Hz; sets time scale for adaptation	U_est	Estimate of input signal U
Ka	Adaptation gain	Uq_est	-π/2 shifted version of U_est
C_ref	Ref. cos, frequency-synch'd to input U	Ampl	Estimated amplitude
S_ref	Ref. sin, frequency-synch'd to input U	Block symbol: 	
U	input sinusoid with unknown amplitude		

Internal schematic:

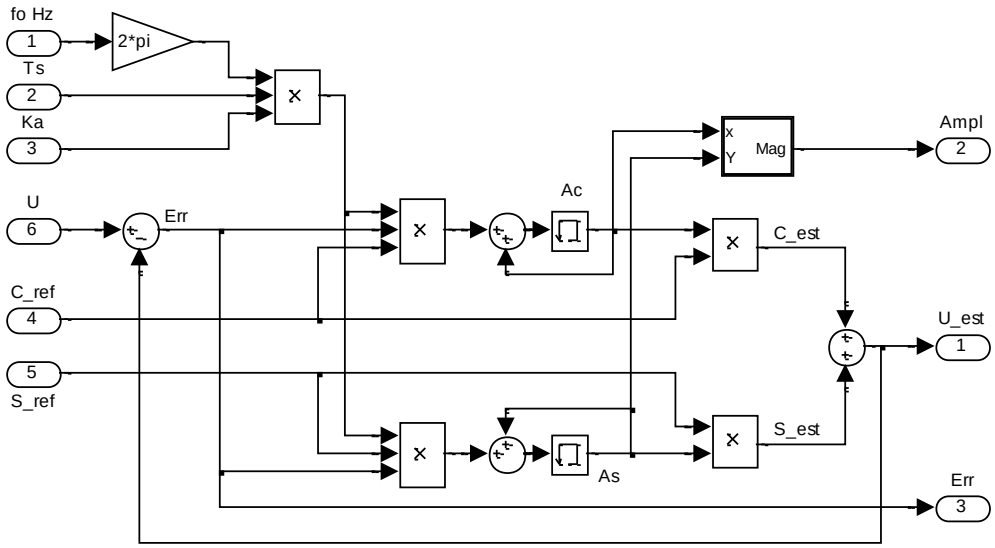


A.4.5 Discretized Single-Phase ASAE for Unknown Phase

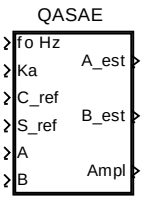
NAME: ASAE_DT			
Function:	Explicitly discretized version of ASAE_1phU		
Inputs		Outputs	
fo Hz	freq. in Hz; serves to set time scale for adaptation	U_est	Estimate of input signal U
Ts	Sampling time	Ampl	Estimated amplitude
Ka	Adaptation gain	Err	Estimate error
C_ref	Ref. cos, frequency-synch'd to input U	Block symbol:	
S_ref	Ref. sin, frequency-synch'd to input U		
U	input sinusoid with unknown amplitude		



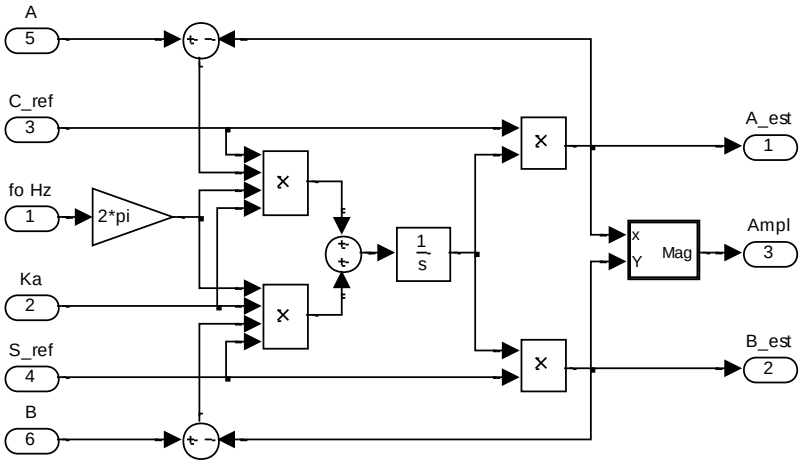
Internal schematic:



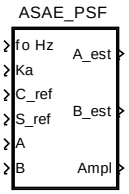
A.4.6 Quadrature-Phase ASAE for Known Phase

Name: QASAE			
Function:	ASAE-based quadrature signal amplitude estimation		
Inputs:		Outputs:	
fo Hz	freq. in Hz; serves to set time scale for adaptation	A_est	Estimate of α input signal
Ka	Adaptation gain	B_est	Estimate of β input signal
C_ref	Ref. cos, frequency-synch'd to αβ inputs	Ampl	Estimated amplitude
S_ref	Ref. sin, frequency-synch'd to αβ inputs	Block symbol: 	
A, B	α and β balanced quadrature components, with unknown amplitude		

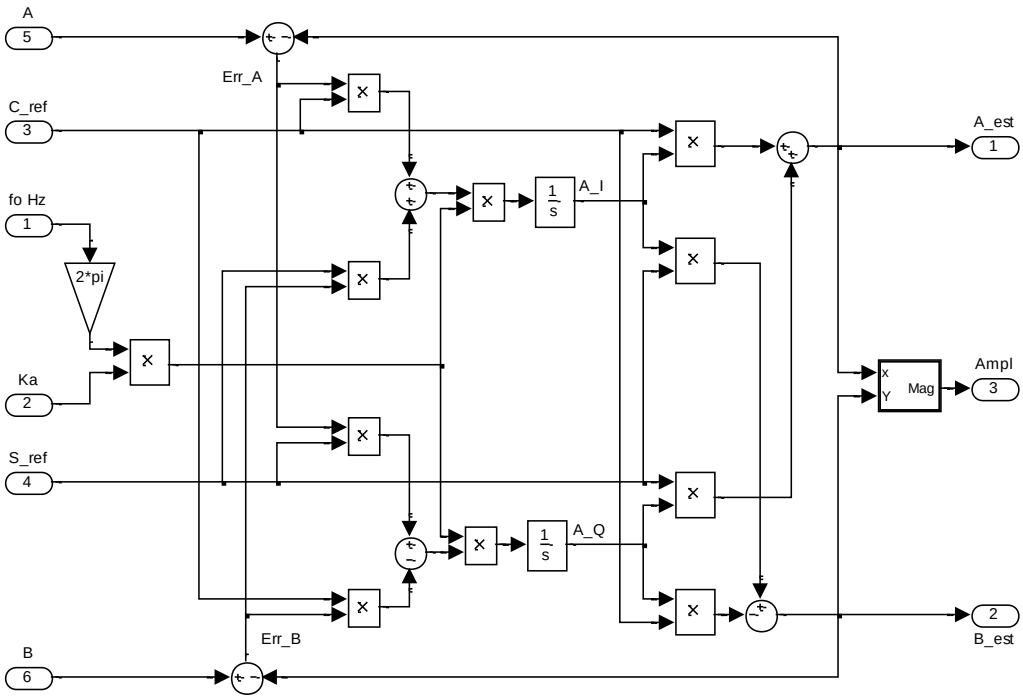
Internal schematic:



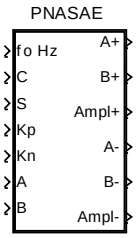
A.4.7 ASAE-based Positive Sequence Filter

Name: ASAE_PSF			
Function:	ASAE-based positive sequence filter		
Inputs		Outputs	
fo Hz	freq. in Hz; serves to set time scale for adaptation	A_est	Estimate of α input signal
Ka	Adaptation gain	B_est	Estimate of β input signal
C_ref	Ref. cos, frequency-synch'd to αβ inputs	Ampl	Estimated amplitude
S_ref	Ref. sin, frequency-synch'd to αβ inputs	Block symbol: 	
A, B	α and β balanced quadrature components, with unknown amplitude		

Internal schematic:

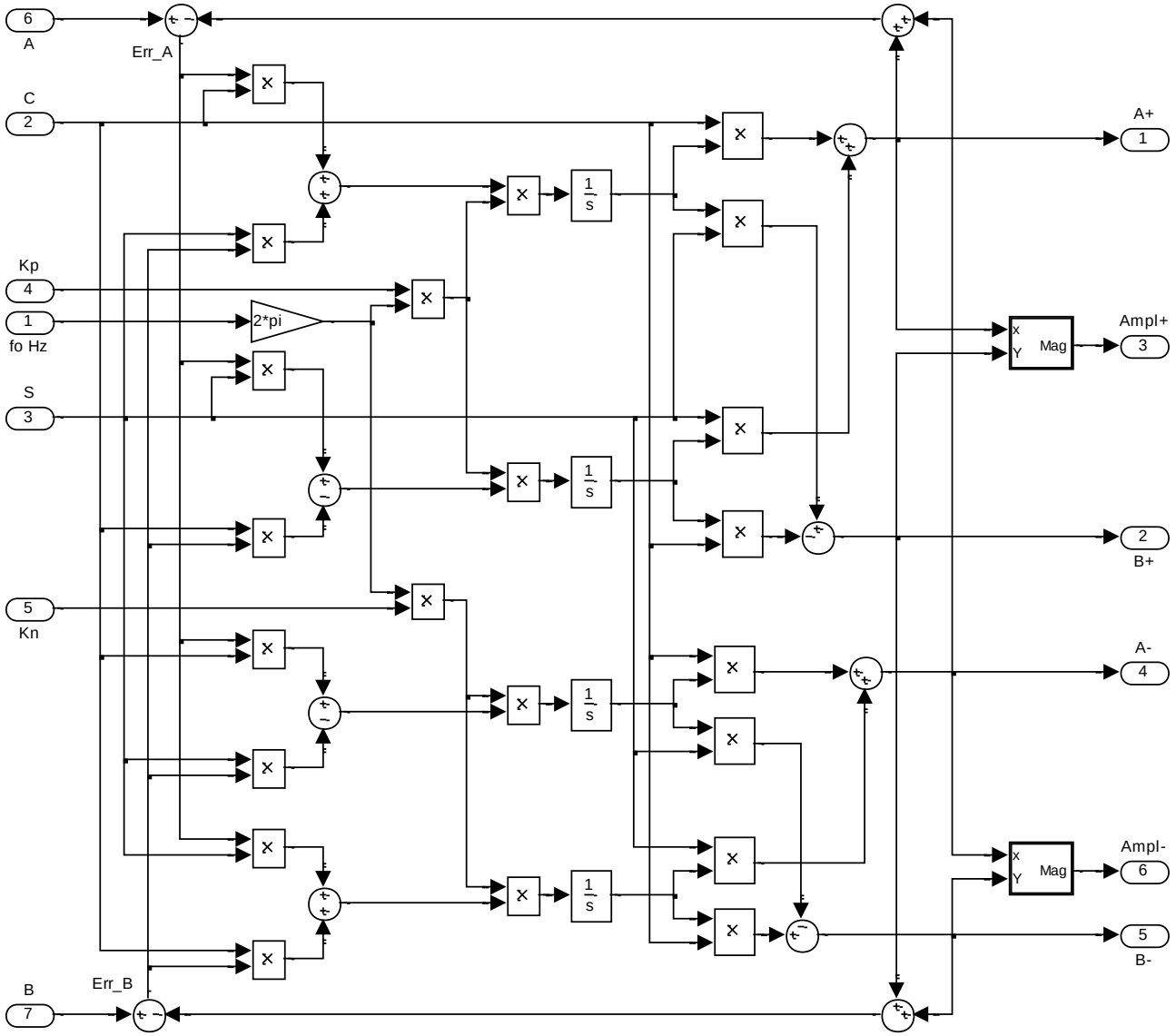


A.4.8 ASAE-based Positive-Negative Sequence Filter

Name: PNASAE			
Function:	ASAE-based positive-negative sequence filter		
Inputs		Outputs	
fo Hz	freq. in Hz; serves to set time scale for adaptation	A+, B+	Estimates of positive sequence αβ input
C	Ref. cos, frequency-synch'd to αβ inputs	Ampl+	Estimated positive sequence amplitude
S	Ref. sin, frequency-synch'd to αβ inputs	A-, B-	Estimates of negative sequence αβ input
Kp	Positive sequence adaptation gain	Ampl-	Estimated negative sequence amplitude
Kn	Negative sequence adaptation gain	Block symbol: 	
A, B	α and β balanced quadrature components, with unknown amplitude		

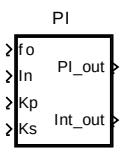
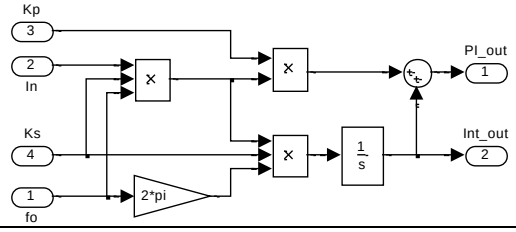
Internal schematic: next page.

PNASAE – Internal schematic:

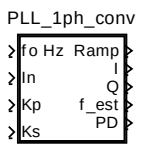
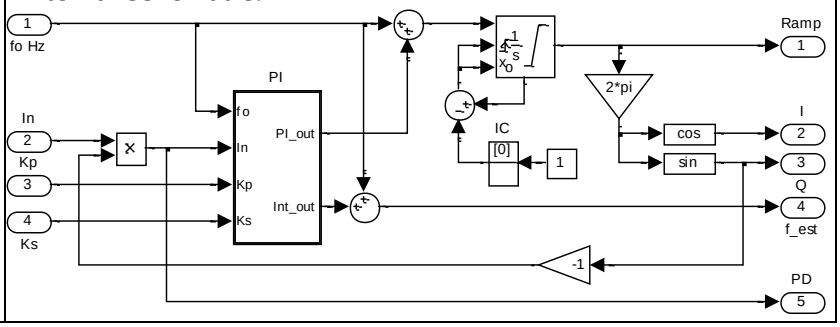


A.5 PHASE-LOCKED LOOPS

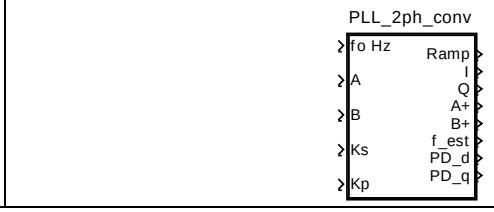
A.5.1 Frequency normalized P-I controller

Name: PI			
Function:	Frequency normalized P-I controller		
Inputs:		Outputs:	
fo Hz	Nominal frequency	PI_out	P-I controller output
In	Input	Int_out	P-I controller integrator output
Kp	Frequency normalized proportional gain		
Ks	Freq. norm'd integral gain		
Block symbol: 	Internal schematic: 		

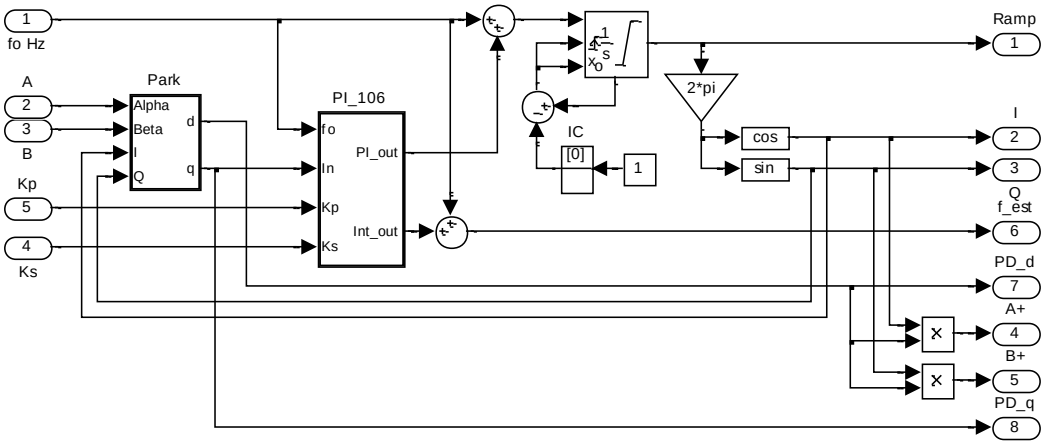
A.5.2 Conventional single-phase PLL with unit magnitude I-Q outputs

Name: PLL_1ph_conv			
Function:	Single-phase PLL		
Inputs:		Outputs:	
fo Hz	Nominal operating frequency	Ramp	Unit magnitude, estimated phase angle
In	Single-phase input signal	I, Q	Estimated In-phase and Quadrature, unit magnitude sinusoidal outputs
Kp	P-I controller frequency normalized proportional gain	f_est	Estimated frequency
Ks	P-I controller frequency normalized speed gain	PD	Phase detector output
Block symbol: 	Internal schematic: 		

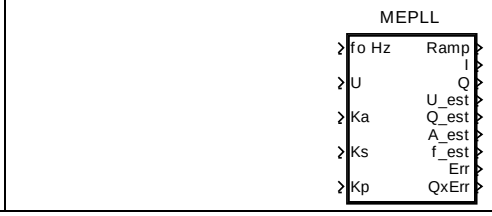
A.5.3 Conventional two-phase PLL with estimated positive sequence α+ and β+ outputs

Name: PLL_2ph_conv			
Function:	Estimates phase and positive sequence of input αβ signal		
Inputs:		Outputs:	
fo Hz	Nominal operating frequency	Ramp	Estimated phase
A, B	Quadrature-phase αβ input signal	I, Q	Unit magnitude quadrature signal
Ks	P-I controller frequency normalized speed gain	A+, B+	Estimated positive sequence
Kp	P-I controller frequency normalized proportional gain	f_est	Estimated frequency
		PD_d	Estimated positive sequence amplitude
		PD_q	Phase detector q output
Block symbol:			

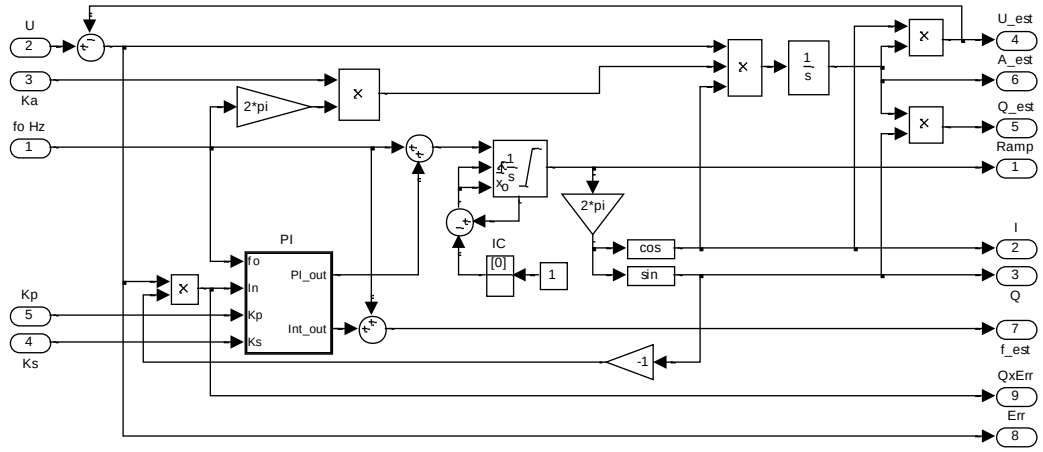
Internal schematic:



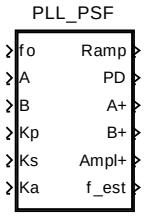
A.5.4 Single-phase Magnitude/Phase Locked Loop / Enhanced PLL

Name: MEPLL			
Function:	Estimates phase, frequency, and amplitude of single-phase sinusoid		
Inputs:		Outputs:	
fo Hz	Nominal operating frequency	Ramp	Estimated phase
U	Single-phase input	I, Q	Estimated unit magnitude quadrature signal
Ka	Amplitude estimation adaptation gain	U_est, Q_est	Estimated input, and -π/2 shifted version
Ks	PLL P-I controller proportional gain	A_est	Estimated amplitude
Kp	PLL P-I controller speed gain	f_est	Estimated frequency
		Err	Estimate error
		QxErr	Q times estimate error = phase detector output
Block symbol:			

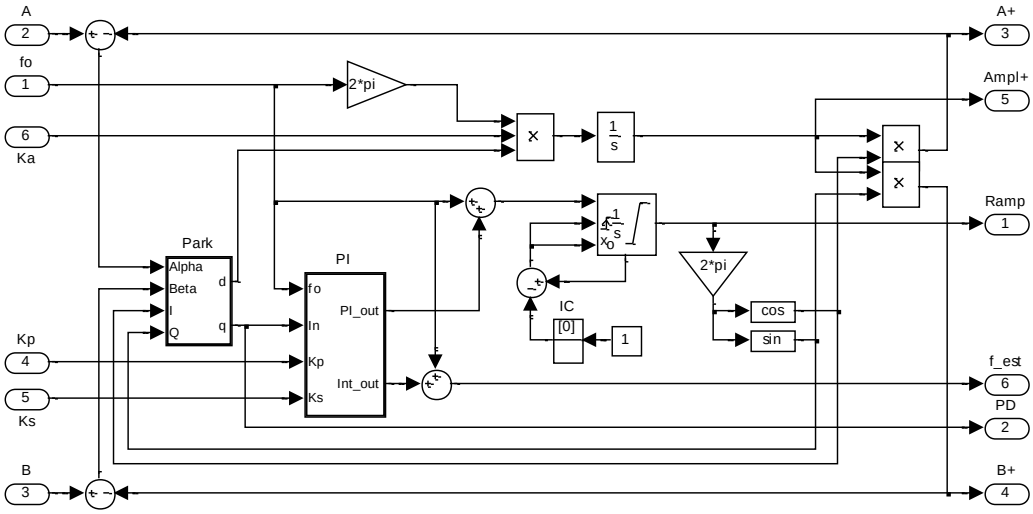
Internal schematic:



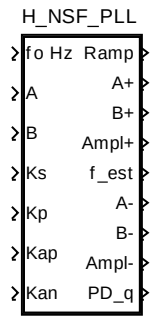
A.5.5 Conventional two-phase PLL-based Positive Sequence Filter

Name: PLL_PSF			
Function:	Implements a frequency-tracking positive sequence filter		
Inputs:		Outputs:	
fo Hz	Nominal operating frequency	Ramp	Unit amplitude estimated phase
A, B	Quadrature-phase signal	A+, B+	Estimated positive sequence
Ka	Amplitude estimation adaptation gain	Ampl+	Estimated positive sequence amplitude
Ks	PLL P-I controller speed gain	f_est	Estimated frequency
Kp	PLL P-I controller proportional gain	PD	Phase detector q output
Block symbol:			

Internal schematic:

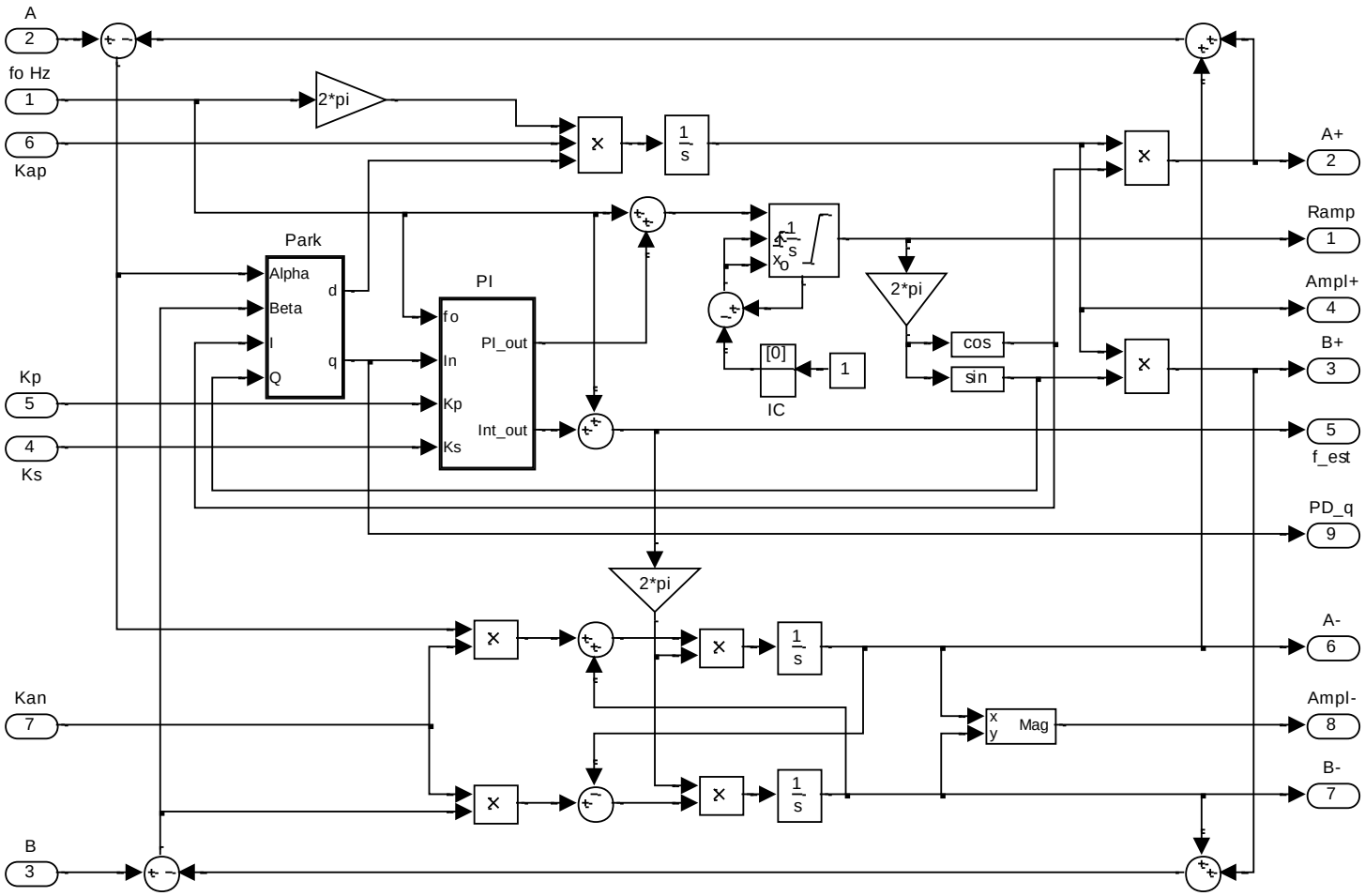


A.5.6 Hybrid Negative Sequence Filter with PLL (H-NSF-PLL)

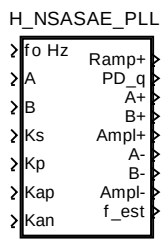
Name: H_NSF_PLL			
Function:	Estimates positive sequence and phase, and negative sequence, of two-phase un-balanced input		
Inputs:		Outputs:	
fo Hz	Nominal operating frequency	Ramp	Unit amplitude estimated phase
A, B	Quadrature-phase signal	A+, B+	Estimated positive sequence
Ks	PLL P-I controller speed gain	Ampl+	Estimated positive sequence amplitude
Kp	PLL P-I controller proportional gain	f_est	Estimated frequency
Kap	Amplitude estimation adaptation gain for positive sequence	A-, B-	Estimated negative sequence
Kan	Amp. estim. adaptation gain for neg. seq.	Ampl-	Estimated negative sequence amplitude
		PD_q	Phase detector q output
Block symbol:			

Internal schematic: next page

H-NSF-PLL – Internal schematic

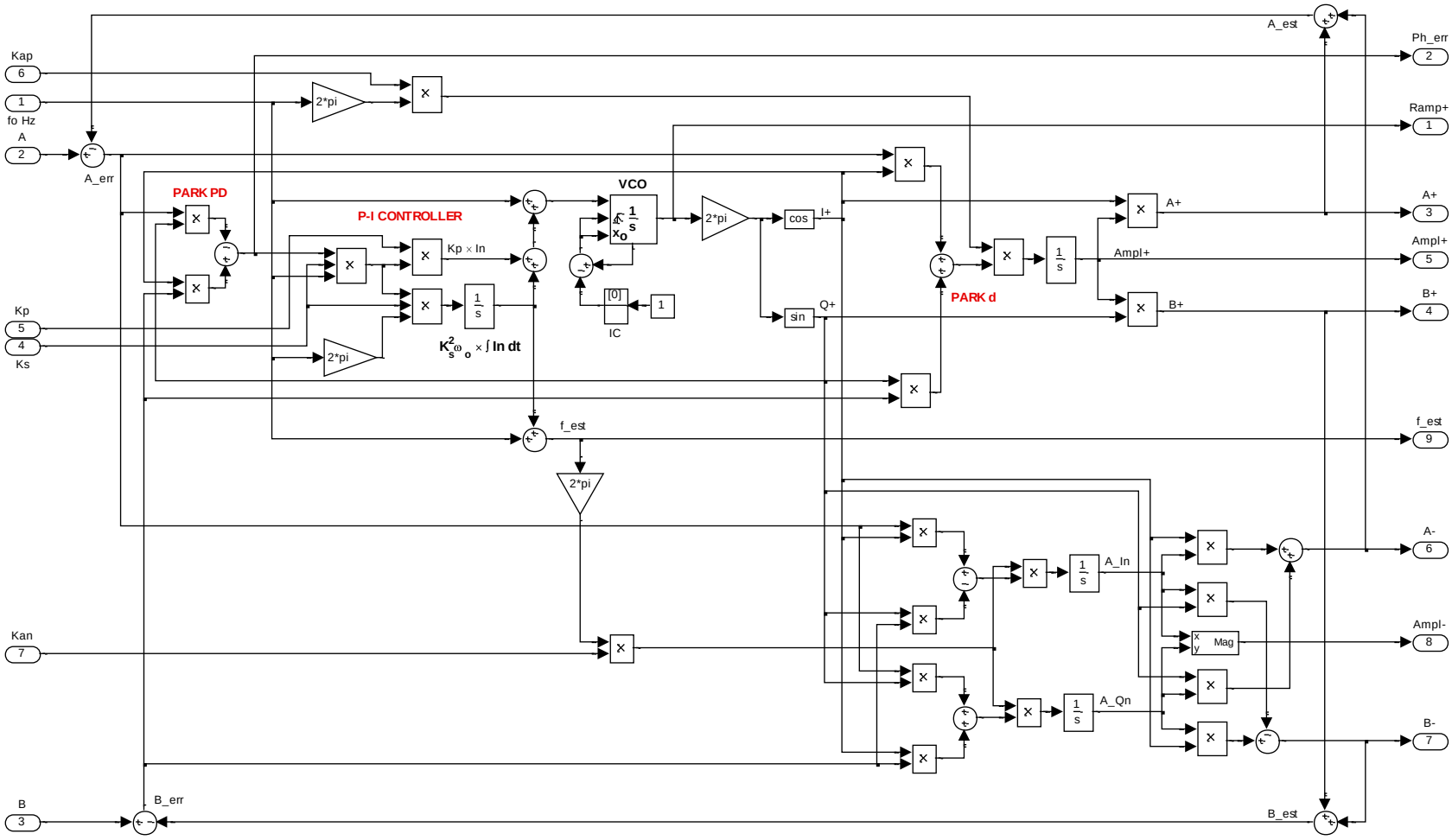


A.5.7 Hybrid Negative ASAE with PLL (H-NSASAE-PLL)

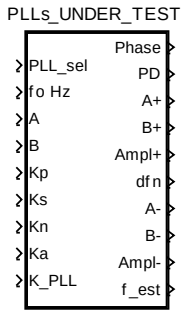
Name: H_NSASAE_PLL			
Function:	Estimates positive sequence and phase, and negative sequence, of two-phase un-balanced input, using ASAE technique		
Inputs:		Outputs:	
fo Hz	Nominal operating frequency	Ramp+	Unit amplitude estimated phase of positive sequence
A, B	Quadrature-phase signal	A+, B+	Estimated positive sequence
Ks	PLL P-I controller speed gain	Ampl+	Estimated positive sequence amplitude
Kp	PLL P-I controller proportional gain	f_est	Estimated frequency
Kap	Amplitude estimation adaptation gain for positive sequence	A-, B-	Estimated negative sequence
Kan	Amp. estim. adaptation gain for neg. seq.	Ampl-	Estimated negative sequence amplitude
		PD_q	Phase detector q output
Block symbol:			

Internal schematic: next page

H-NSASAE-PLL – Internal schematic

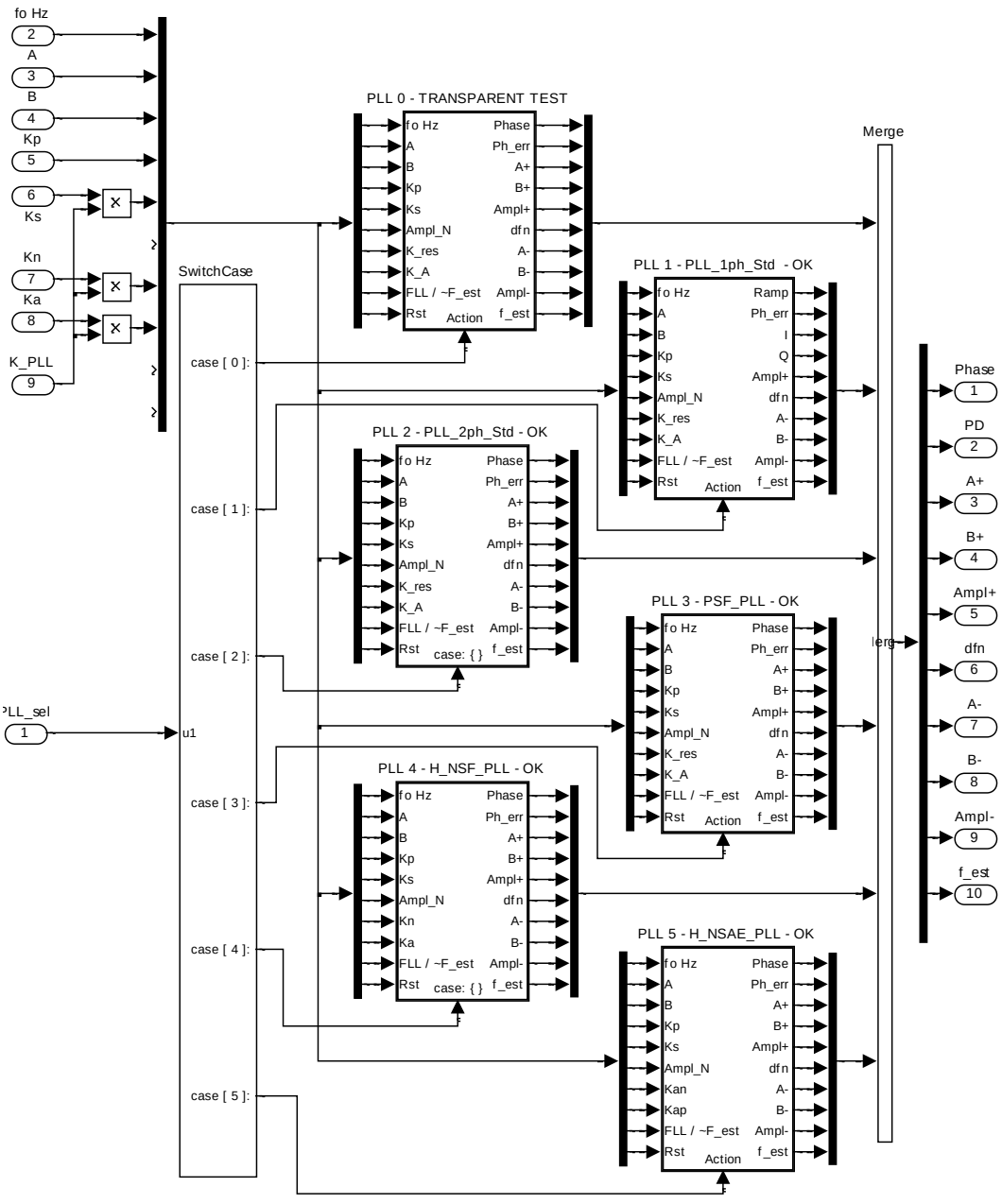


A.5.8 PLLs Under Test

Name: PLLs_UNDER_TEST			
Function:	This block collects together various PLL blocks for evaluation, for convenient run-time selection.		
Inputs:		Outputs:	
PLL_sel	Numerical value to select one of the PLL blocks		
fo Hz	Nominal operating frequency	Phase	Unit amplitude estimated phase of positive sequence
A, B	Quadrature-phase signal	PD	Phase detector q output
Kp	PLL P-I controller proportional gain	A+, B+	Estimated positive sequence
Ks	PLL P-I controller speed gain	Ampl+	Estimated positive sequence amplitude
Kn	Amplitude estimation adaptation gain for neg. seq. filter	dfn	Estimated frequency deviation from fo, as a fraction of fo
Ka	Amp. estim. adaptation gain for pos. seq.	A-, B-	Estimated negative sequence
K_PLL	Multiplicative factor for Ks, Kn, and Ka default = 1	Ampl-	Estimated negative sequence amplitude
		f_est	Estimated frequency
Block symbol:			

Internal schematic: next page

PLLs_UNDER_TEST – Internal schematic



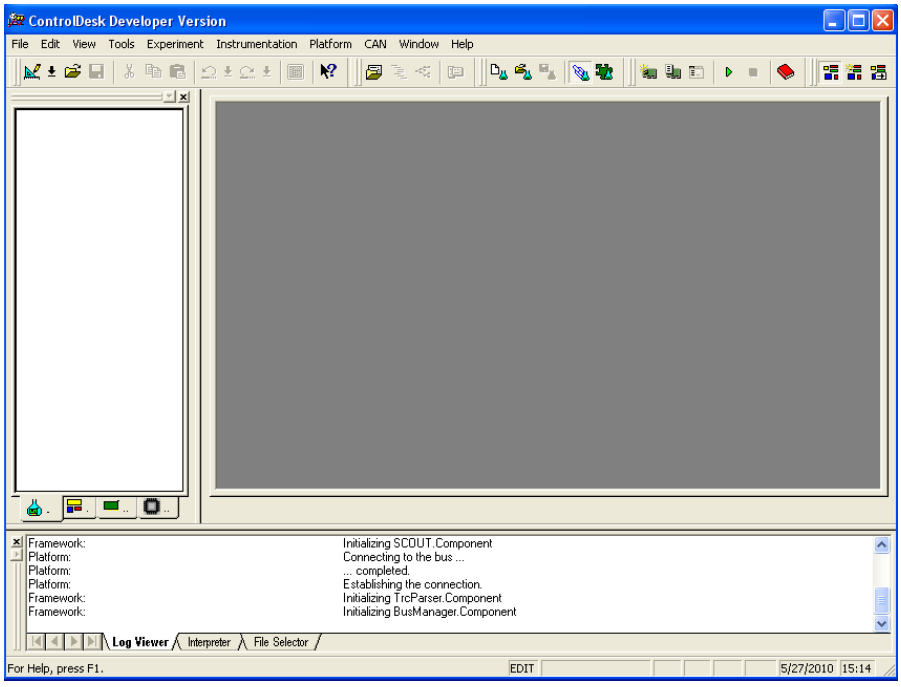
APPENDIX B – dSPACE CONTROLDESK OPERATION

This appendix presents a brief guideline to operating the Simulink PLL test program on the dSPACE DS1104 platform, under control of the ControlDesk program.

A) PROGRAM EXECUTION

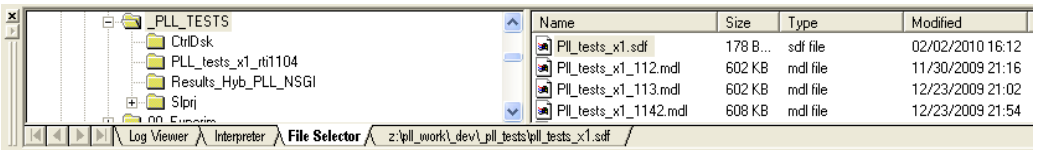
1) Start ControlDesk, either via it's icon (

) on the desk top, or from Windows' Start menu, in the dSPACE Tools folder, which should normally open up a window as shown below. If the Navigator pane is not visible to the left side, as seen below, then press ALT-1 (or select **View** on the top main menu bar, and select **Activate Navigator**), and likewise, if the Tool Window does not appear at the bottom, then press ALT-2 (or select **View** on the top main menu bar, and select **Activate Tool Window**).



2) At the bottom of the Navigator Pane are four tabs; select the Platform tab (third, green rectangle), which should present a list of available platforms, including the Simulink and DS1104 choices, as shown below. The platform on which the program is to execute will be selected next.

3) In the Tool Window at the bottom, select the File Selector tab, as shown below, and navigate to the drive and folder where the program sdf file is located, and locate the executable file PLL_Tests_x1.sdf.

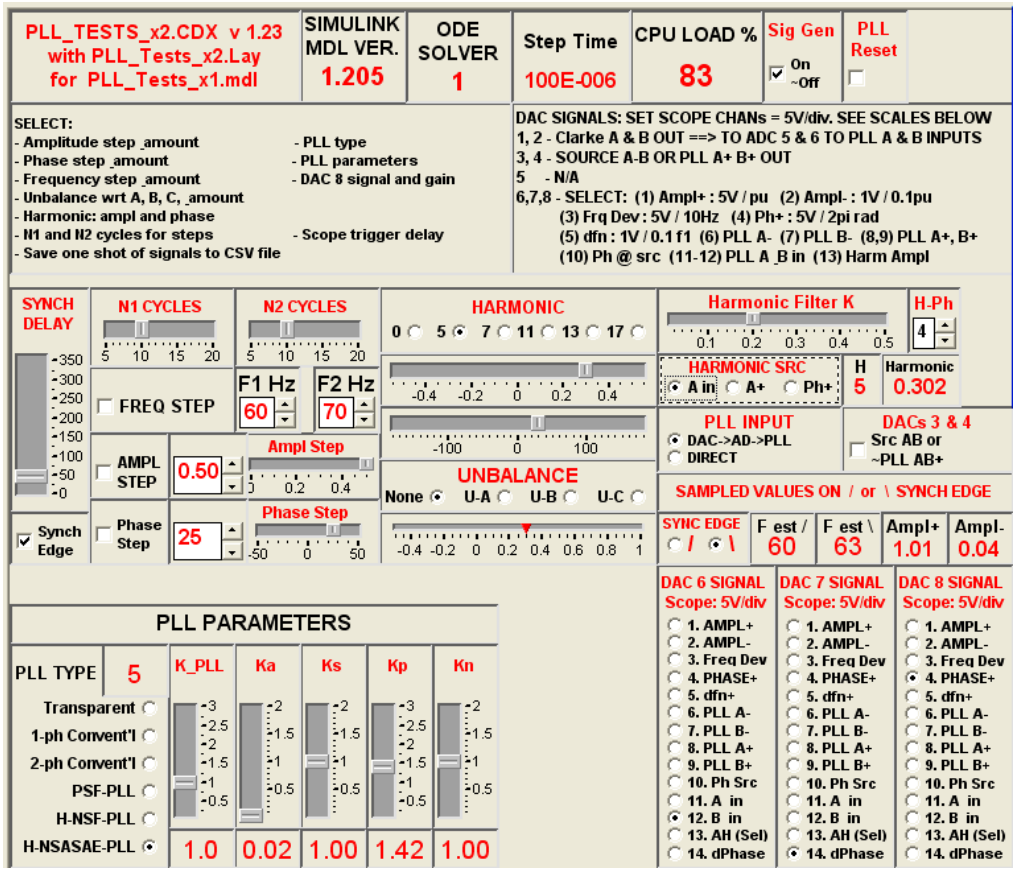


4) Click once on the PLL_Tests_x1.sdf file, and drag it to, and drop it on, the ds1104 platform in the Navigator pane, which effectively loads the program onto the DS1104 platform, and should start executing it.

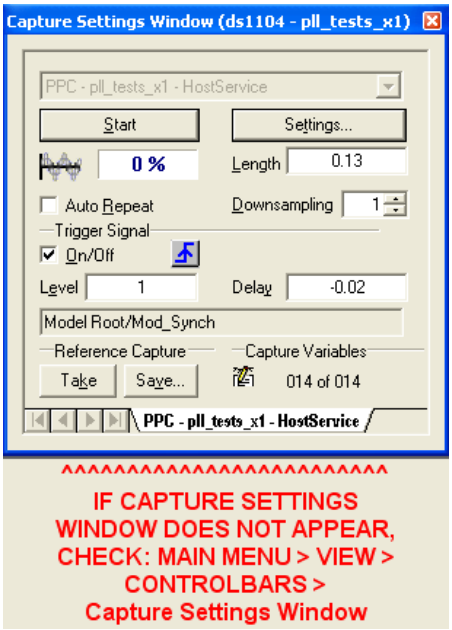
5) Close the Navigator and Tool Window panes, by clicking on the small close buttons ( ); if necessary, these can later be re-opened by pressing ALT-1 and ALT-2, respectively.

6) The program execution can be monitored and controlled in real time via a so-called experiment, which consists of a user-designed control panel, as shown below. Open the PLL tests experiment from the main top menu bar: select FILE, then Open Experiment (or press CTRL-SHIFT-O), and browse

as necessary to locate and select the file PLL_tests_x2.cdx; re-size the main ControlDesk window as necessary, to appear as shown in the partial display below. The experiment is not yet running, and must be switched to Animation Mode: from the main menu bar, select Instrumentation, then select Animation Mode (or press F5).



It may also be necessary to re-position the Capture Settings window, to appear within the frame of the main program panel, as suggested by the message "IF CAPTURE SETTINGS …", as shown below. Note that the Capture Settings window is in fact a separate entity, not docked to the main window, and depending on previous session operation, it may not appear. It can be opened from the main menu, via View, and then selecting Capture Settings Window.



B) PROGRAM OPERATION

The overall program layout and operation can be briefly summarized as follows. Refer to chapter 4 for further details.

7) The top left corner presents some general information, including:
   i. names of related ControlDesk experiment, ControlDesk layout, and Simulink files
   ii. Simulink program file version number
   iii. which Simulink solver was used to compile the program
   iv. run time relative CPU loading, which serves more as a guide as a program grows in complexity
   v. a very brief summary of user controls
   vi. a summary of signals output to the DS1104 DAC's for observation

8) The test signal generator operation can be controlled by the user via the following parameters:

i. N1 and N2: these select the duration, in cycles at F1 and F2, for each switching state of the selected signal modulations. For example, the frequency switching can be enabled, via FREQ STEP, below, to switch between 60 Hz for 10 cycles, and 70 Hz for 8 cycles;

ii. FREQ STEP: selects frequency switching on or off, to switch between F1 and F2 Hz;

iii. AMPL STEP: selects amplitude switching on or off, to switch between 1 and 1+Ampl2, for example between 1 and 1.5 pu;

iv. Phase Step: selects phase stepping on or off, to switch between 0 and the indicated value, in degrees;

v. HARMONIC: select which harmonic to switch on and off, and adjust the selected harmonic amplitude with the slider control; select 0 to disable harmonics;

vi. UNBALANCE: select the type of un-balance to introduce; for example U-A selects unbalance type C_A, which alternates between 0 and an amount corresponding to phase A amplitude = 1 + value selected with the slider; phases B and C change accordingly to maintain a zero sequence amplitude equal to 0;

vii. SYNCH DELAY: allows to adjust at which angle within a cycle that the selected parameters switching is to occur; for example, at a value of 0.2, the frequency will switch after N1 and N2 cycles when the phase crosses 0.2 cycle, i.e. 72^o;

viii. Synch Edge: selects the phase relation between the modulation switching and the trigger signal sent to the digital output pin 0 for scope triggering; this allows to select which transient response to observe on an oscilloscope.

9) The PLL parameters can be selected as follows:

i. select PLL type for testing

ii. the main PLL parameters consist of:

a. Ka: the positive sequence adaptive amplitude gain and bandwidth
b. Ks: the positive sequence PLL frequency tracking bandwidth
c. Kp: the positive sequence PLL proportional gain
d. Kn: the negative sequence adaptive amplitude gain and bandwidth
e. K_PLL: adjusts together parameters Ka, Ks, and Kn, to change the common speed of response, or "time scale", and bandwidth, of the PLL

10) Input and output signals: the actual PLL input signals can be selected from:
   i. ADC inputs 1 and 2, which can be connected either to a physical signal, or to DAC outputs 1 and 2, respectively, which produce the internal signal generator's α and β signals, or
   ii. the internal signal generator, directly without going through the DAC-ADC connection; this is useful for verifying that all is operating properly:
   iii. any three of various internal signals of interest can be selected for output via DAC outputs 6-8 for observation:

11) The steady state values of certain signals of interest (ex. PLL estimated frequency, or positive or negative sequence amplitude, or harmonic level) can be sampled just before the step in modulation, i.e. on the rising or falling edge of the modulation control signal. It is important allow a suitable duration, via N1 and N2, for the signals to settle after their transients, especially for lower values of Ka, Ks, and Kn, which result in slower convergence.

12) The selected harmonic component added to the generated signal for distortion:

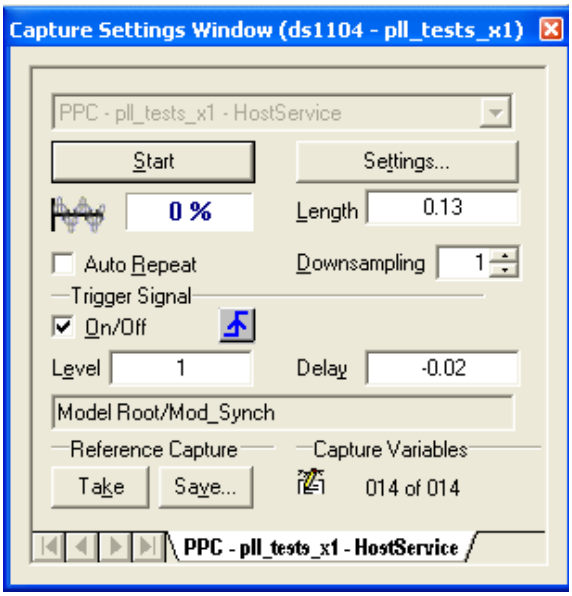
i. can be observed by it's amplitude, either at (a) the α input of the PLL, or (b) at it's α output; this can serve to measure the rate of attenuation of any harmonic component from input to output;

ii. can be sampled on either the rising or falling edge of the modulation control signal;

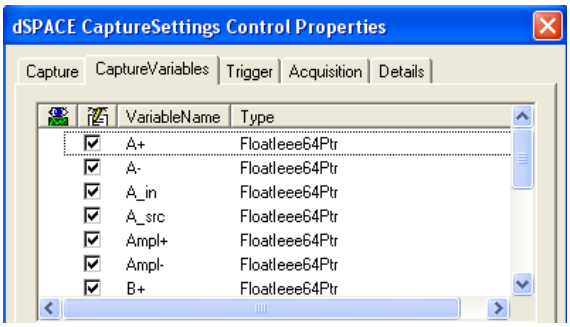
iii. is first processed by a band-pass filter with user-adjustable bandwidth, via it's K factor, adjustable from 0.01 to 0.2; for this reason, as mentioned above, N1 and N2 should be adjusted to allow adequate transient settling time, especially for low values of K.

C) DATA ACQISITION

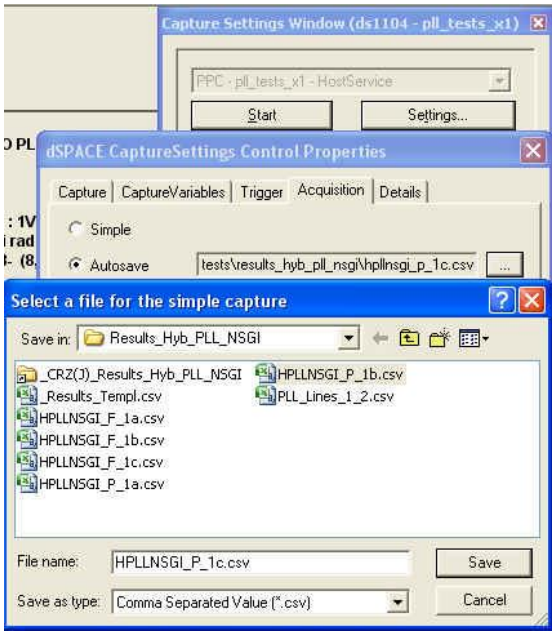
Because of constraints on CPU resources, not all analyses of interest can be implemented in real time. Thus, it may become necessary to record various signals of interest to disk for subsequent analysis, such as with Matlab. For this purpose, ControlDesk provides a capability for signal capture, which is configured via the Capture Settings Window, shown below.



1) Click on the Settings button, and then select the Capture Variables tab; a list of variables should appear as show below:

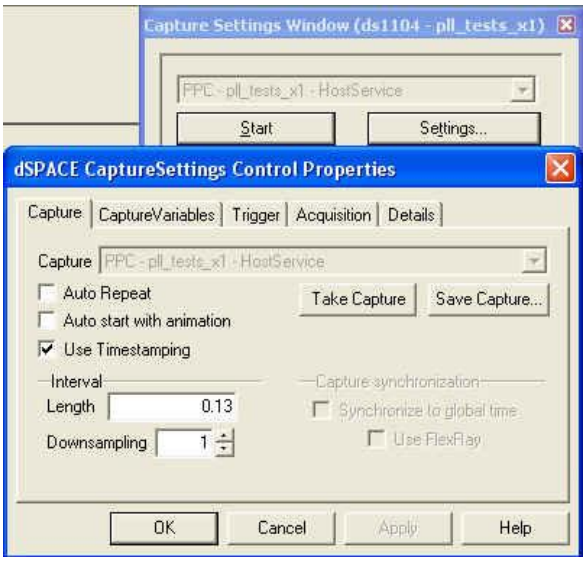


2) A file name must be specified in which to save the signals: select the Acquisition tab, and then check the Autosave selection, and then specify a file name, with extension .CSV (Comma Separated Variables), which is compatible with Excel, and suitable for input to Matlab.



3) The signals can be stored to the above-named file upon specified conditions, such as a trigger event, and for a limited record, as specified by selecting the Capture tab, as shown below:

4) Subsequently, as the program is running, at any time, the Start button can be pressed, to cause one data record to be saved to disk.



5) Finally, depending on the data resolution required, such as for subsequent plotting, it may be determined that not all data samples need be recorded to disk, such that the data may be downsampled. For example, whereas the model runs at 10 KHz, and thus generates data samples every 100 μSec., with the Downsampling parameter set to 2, only every second sample of every selected signal will be recorded to disk.