CHAPTER 4

DEVELOPMENT OF MATLAB/LABVIEW MODEL OF PHASOR MEASUREMENT UNIT

4.1 INTRODUCTION

With the advent of Phasor Measurement Units (PMUs), it has become quite possible to monitor and control power system networks in real time. PMUs are able to provide time stamped measurements of voltage and current phasors using Global Positioning System (GPS) satellites, in microseconds. PMUs placed at the substation as well as distribution transformers may be efficiently utilized on real time estimation of voltage and current phasors, and frequency. Measured data sent to master controller at the main substation and local controllers at DTs may be processed by these controllers to determine available power at the main substation, current through feeders, real and reactive power loading of three phases of feeders, at frequent interval. Thus, distribution network may be monitored and controlled in real time at regular intervals.

Different phasor estimation techniques such as Fast Fourier transform (FFT), Discrete Fourier transform (DFT) may be used by PMUs to estimate phasors. In this chapter, DFT based phasor estimation in developed MATLAB and LABVIEW models of PMU has been presented. Developed MATLAB and LABVIEW models estimate phasors using non-recursive as well as recursive algorithms.

4.2 METHODOLOGY FOR PHASORS ESTIMATION

Present work estimates phasors using DFT that is a method in which fourier transform of a number of samples is calculated taken from an input signal x(t). The Fourier transform is calculated at discrete steps in the frequency domain.

Consider a pure sinusoidal quantity given by-

$x(t) = \sqrt{2}\, X\, \mathrm{Sin}\, (\omega t + \varphi)$ (4.1)

ω is the frequency of the signal in radian per second, and φ is the phase angle in radians. Considering (4.1) and assuming that x(t) is sampled N times per cycle,

$x(k) = \sqrt{2}\, X\, \mathrm{Sin}\, (\frac{2\pi}{N} k + \varphi)$ (4.2)

The DFT of x(k) contains the fundamental frequency component given by-

$X = \frac{2}{N} \sum_{k=0}^{N-1} x(k).\ e^{\frac{-j2\pi k}{N}} = X_c - jX_s$ (4.3)

where, $X_c = \frac{2}{N} \sum_{k=0}^{N-1} x(k).\mathrm{Cos}\, (\frac{2\pi k}{N})$ (4.4)

and, $X_s = \frac{2}{N} \sum_{k=0}^{N-1} x(k).\mathrm{Sin}\, (\frac{2\pi k}{N})$ (4.5)

Considering (N-1) sample as the last sample for the phase estimation, the phasor X^{N-1} is given by-

$X^{N-1} = X_c^{N-1} - j\, X_s^{N-1} = X\, [\mathrm{Cos}\varphi + j\, \mathrm{Sin}\varphi] = X\, e^{j\varphi}$ (4.6) In (4.6), the φ represents the angle between the time of the first sample and the peak of the input signal.

4.2.1 Non-Recursive phasor computation

Considering the data sample for the r^{th} window-

$X_c^{(r)} = \frac{2}{N} \sum_{k=0}^{N-1} x(k + r - 1).\mathrm{Cos}\, (\frac{2\pi k}{N})$ (4.7)

and, $X_s^{(r)} = \frac{2}{N} \sum_{k=0}^{N-1} x(k + r - 1).\mathrm{Sin}\, (\frac{2\pi k}{N})$ (4.8)

$X^{*\,(r)} = \frac{1}{\sqrt{2}} X_s^{(r)} + jX_c^{(r)} = X^{*\,(r-1)}.\ e^{\frac{j2\pi k}{N}}$ (4.9)

Phasor estimation using (4.9) represents non-recursive algorithm as it requires 2N multiplications and 2(N-1) additions. While progressing of one data to next window, only one sample x_0 is discarded and sample x_N is added to the data set. As the phasor calculations are performed fresh for each window without using any data from the earlier estimates, this algorithm is known as a non-recursive algorithm**.**

4.2.2 Recursive phasor computation

In recursive computation, an update is to be made in old data phasor that determines the value of new phasor. Superscripts 'new' signify computation from data window 2 and 'old' signify computation from data window 1. Consider the sinusoid,

$$x(t) = \sqrt{2}\ X\ \mathrm{Sin}\ (\omega t+ \varphi+\frac{2\pi}{N}) \tag{4.10}$$

where, N= number of samples taken in data window. With recursive algorithm, phasor using new data window 2 may be estimated using-

$$X^{*(new)} = Xe^{\,j(\varphi+\frac{2\pi}{N})} = X^{*\,(old)}e^{\,j(\frac{2\pi}{N})} \tag{4.11}$$

In general, the r^{th} phasor is computed from the $(r\text{-}1)^{th}$ phasor by-

$$X^{*\,(r)} = X^{*\,(r-1)} + j\frac{1}{\sqrt{2}}\cdot\frac{2}{N}\,(x_{N+r}-x_r).e^{\frac{-j2\pi}{N}(r-1)} \tag{4.12}$$

In the recursive algorithm, only two multiplications are needed to be performed at each new sample time. When the input signal is a pure sine wave of fundamental frequency i.e. $X_{N+r}=X_r$, $X^{*\,(r-1)}$ becomes $X^{*(r)}$ for all r. Therefore, when a recursive computation is used to calculate phasors, it leads to stationary phasors in the complex plane.

4.3 PHASOR MEASUREMENT UNIT MODEL IN MATLAB/SIMULINK

The developed MATLAB/SIMULINK model of PMU is shown in Fig. 4.1. Various blocks of MATLAB/ SIMULINK model of PMU are explained below.

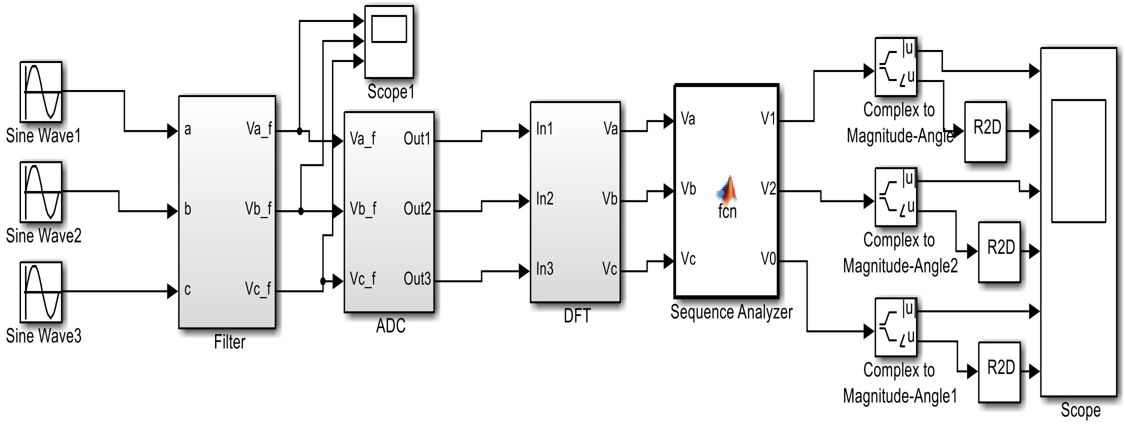

Fig. 4.1 MATLAB/SIMULINK model of PMU

4.3.1 Input Signal (Sine Wave)- The analog sinusoidal signal is provided as an input to the PMU in the form of sine wave. These analog inputs may be the output of current and voltage transformers.

4.3.2 Filter- An anti-aliasing filter is used before a signal sampling to satisfy the sampling theorem, and to approximately limit the bandwidth of a signal. The Butterworth Band Pass filter specifically have center frequency f_0=50 Hz, bandwidth Δf=10 Hz, and has flat response in pass-band as compared to other filter. The Butterworth filter of 2^{nd} order is chosen to obtain better results. The 2^{nd} order Butterworth band-pass filter has accurate filter response as compared to 3^{rd} order and 5^{th} order. The 2^{nd} order filter response for a signal x(t)=230(Cos $_1$ $_1$+Cos $_2$ $_2$+Cos $_3$ $_3$) having frequency $_1$=40 Hz for t=1s, $_2$=50 Hz for t=0.5s, $_3$=60 Hz for t=0.5s with center frequency f_0=50 *Hz* and bandwidth Δf=10 *Hz* has been considered in present work and is shown in Fig. 4.2.

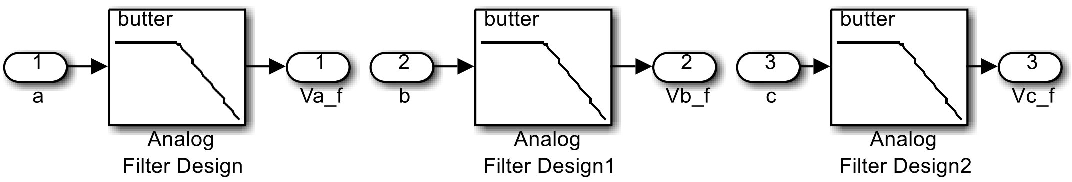

Fig. 4.2 Butterworth Band Pass filter of order 2

4.3.3 Analog-to-Digital Convertor (ADC)- ADC is a device used to convert a continuous signal to discrete form physical quantity i.e. voltage in terms of discrete form. Quantization of the input involves in ADC, which introduces a small amount of error. ADC consists of pulse generator, quantizer and sample and hold circuit as shown in Fig. 4.3. The sample & hold circuit gives the equivalent result of an ADC. Sample and hold circuit converts the input signal into digital signal.

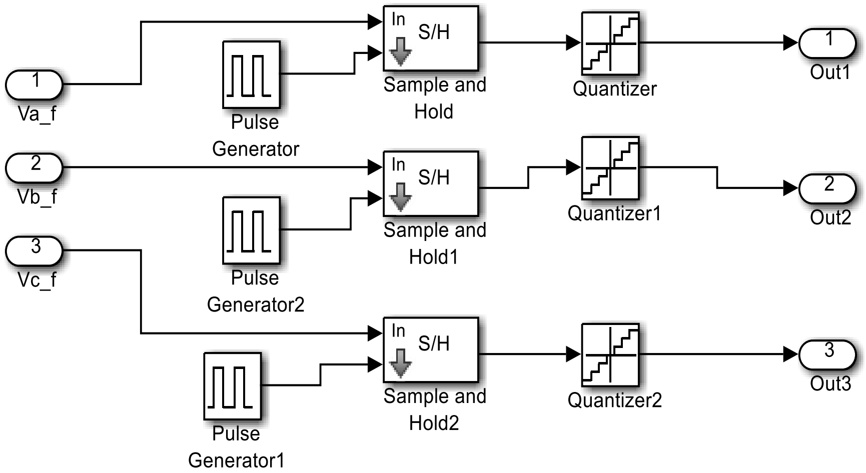

Fig. 4.3 ADC using Sample and Hold Circuit

In PMU, Global Positioning System (GPS) receiver is used for time stamping. In present work, pulse generator has been used that generates 1000 pulses for sampling rate of 20 samples/cycle in 1s as an input signal and has been sampled in accordance with output of the pulse generator. The quantizer is used to round off error and discretize the input at the interval of 0.5s. This represents Clock Synchronization Unit i.e. used to detect 1 Pulse Per Second (PPS) synchronizing signals from GPS source. A control system that generates an output signal whose phase is related to the phase of an input signal represents Phase Locked Loop (PLL).

4.3.4 Discrete Fourier Transform (DFT)- This block has been used to estimate phasors using DFT considering non-recursive and recursive algorithms.

4.3.5 Sequence Analyzer- The estimated voltage and current phasors are used as inputs for embedded MATLAB function known as sequence analyzer that helps in obtaining positive sequence phasor.

4.4 PHASOR MEASUREMENT UNIT MODEL IN LABVIEW

LABoratory Virtual Instrument Engineering Workbench (LABVIEW) uses programs i.e. represented by icons to create applications and these programs are called Virtual Instruments (VI). The Graphical Interface (GI) of LABVIEW consists of front and back panel window.

(i). The inputs are predefined and controlled from the front panel and output is reflected in front panel only;

(ii). Back panel is used to connect VI to construct the logical operations.

LABVIEW models have been constructed using standard library VI, and also using some user defined VI.

In the non-recursive algorithm, the input signal has been generated from library VI i.e. Simulate Signal VI. This analog input signal is then converted to discrete signal with A2D VI and stored in an array. Here, data window is considered for 12 samples. A user defined VI has to be prepared for calculating the Fourier coefficient of the data samples. Appropriate arithmetic operations are performed to estimate phasor for first data window. The complex term obtained after phasor calculation is converted into polar form and displayed as output. This algorithm is repeated for subsequent data samples. As newer estimate of phasor is performed, the phasor rotates anticlockwise by an angle Θ due to delay of each sample by one sampling angle. The developed non-recursive LABVIEW model is shown in Fig. 4.4.

For recursive algorithm, phasors are to be estimated over a data window. The non-recursive algorithm has been used to generate the first estimate of phasor in the present work. Once the phasor has been obtained for the first data window, recursive algorithm has been used. When phasor estimate is obtained for $(N+r-1)^{th}$ sample, $(N + r)^{th}$ sample is compared with r^{th} sample using suitable mathematical operands provided in the standard library. The developed recursive LABVIEW model for stage I and stage II is shown in Figs. 4.5(a) and 4.5(b).

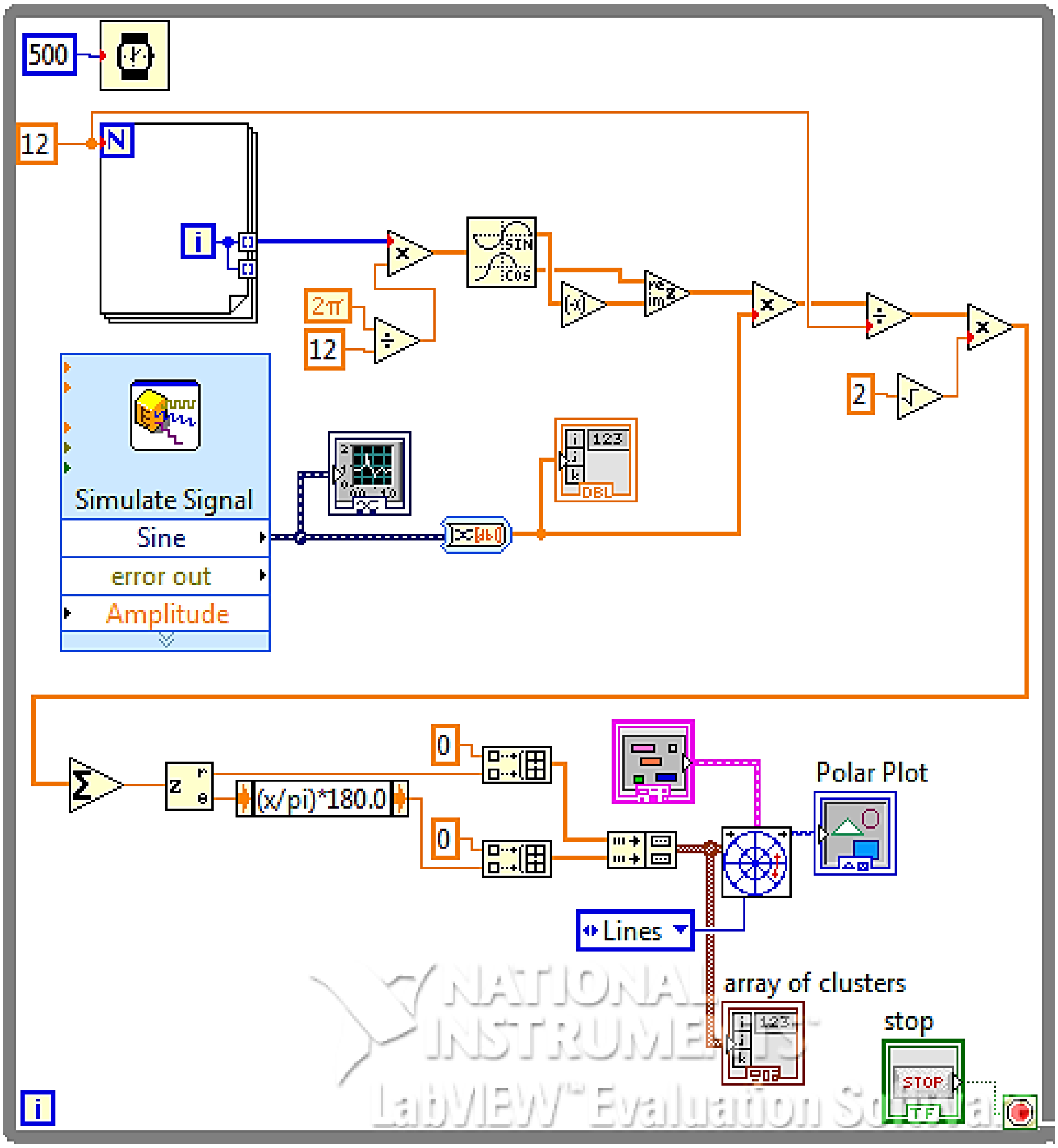

Fig. 4.4 The non-recursive LABVIEW model

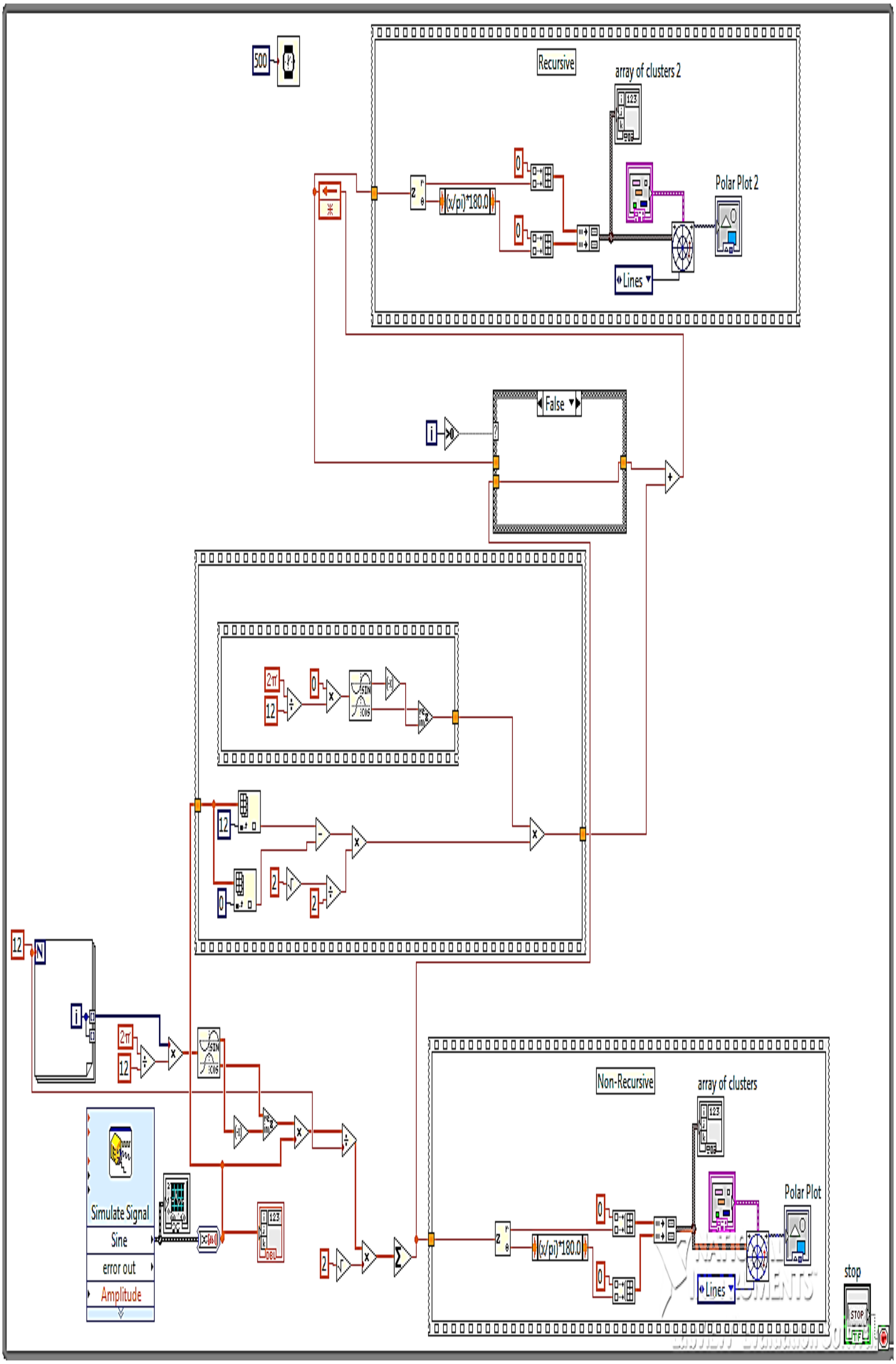

Fig. 4.5(a) The recursive LABVIEW model- Stage I (False Condition)

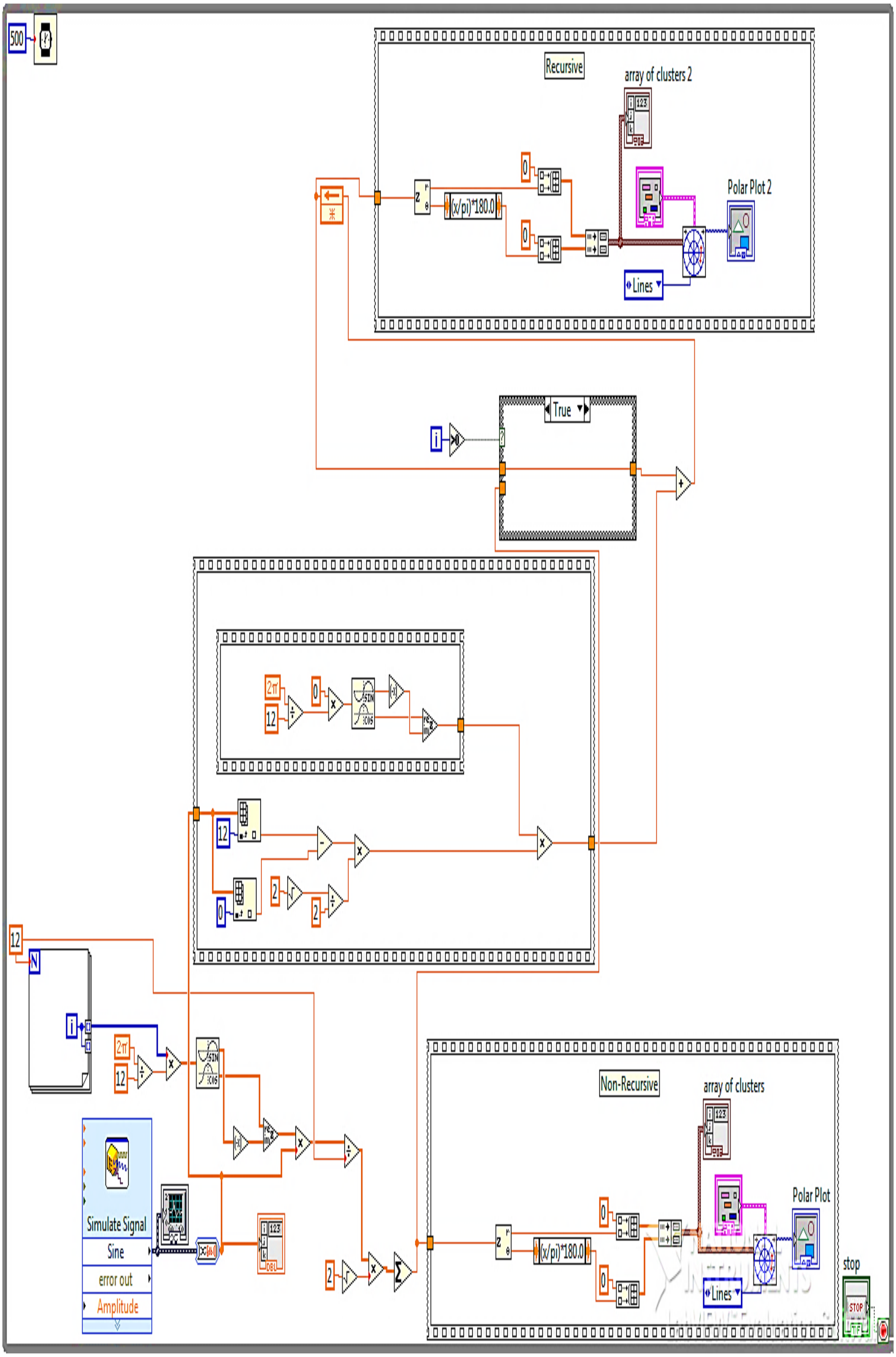

Fig. 4.5(b) The recursive LABVIEW model- Stage II (True Condition)

4.5 CASE STUDIES

MATLAB Simulation and LABVIEW model for PMU were developed and compared for recursive and non-recursive DFT algorithms to get the required phase and magnitude of the signal.

A test case have been considered for single phase 230V, $x(t)=230 \quad (100 \quad + \frac{}{4})$ at frequency 50Hz which is sampled at a sampling frequency of 600 Hz i.e. 12 samples per cycle. The phasor output in polar form for non-recursive and recursive estimates is shown in Figs. 4.6(a), 4.6(b), 4.6(c), 4.6(d), respectively. It is observed from the Figs. 4.6(a) & 4.6(b) that phasor estimates obtained with non-recursive algorithm maintains a constant magnitude. However, it is observed from Fig. 4.6(b) that the new phasor is advanced by 30^o from the phasor shown in Fig. 4.6(a). It is observed from Figs. 4.6(c) and 4.6(d) that with recursive phasor estimation, estimated magnitude and phase of the phasor remains constant over different data windows.

The phasor has been estimated using recursive and non-recursive algorithms. The estimated phasors are shown in Table 4.1. Developed MATLAB based model and LABVIEW based model resulted in same phasor estimation using recursive and non-recursive algorithms. However, computational times in phasor estimation by two models were different. Computational time for phasor estimation using non-recursive and recursive algorithm for the MATLAB based model and LABVIEW based model are shown in Table 4.1. It is observed from Table 4.1 that MATLAB based model requires less computational time in phasor estimation as compared to LABVIEW based model using non-recursive and recursive algorithms. It is also observed from Table 4.1 that recursive algorithm based phasor estimation takes less computational time compared to non-recursive algorithm for the MATLAB based models for most of the cases, whereas, for the LABVIEW based models, the two algorithms require same computational effort. First 20 samples have been considered for phasor estimation. Since phasor estimation is performed over a cycle, the first phasor is obtained after one complete cycle of the sinusoid i.e. after obtaining 12 samples. Due to this, first 12 columns of Table 4.1 of the recursive and non-recursive phasor updates are empty, at which time the first data window is completely filled.

TABLE 4.1 COMPUTATIONAL TIME FOR PHASOR ESTIMATION USING NON-RECURSIVE AND RECURSIVE ALGORITHM FOR THE MATLAB AND LABVIEW BASED MODEL

Sample No.	Computational Time				Samples (Xn)	Non-Recursive phasor estimates	Recursive phasor estimates
	In MATLAB		In LABVIEW				
	In Non-Recursive Estimation (Micro-secs.)	In Recursive Estimation (Micro-secs.)	In Non-Recursive Estimation (Secs.)	In Recursive Estimation (Secs.)			
0	19.89	16.68	0	0	162.635	--	--
1	6.41	5.13	0.0016	0.0016	59.5284	--	--
2	1.28	2.57	0.0033	0.0033	-59.5284	--	--
3	1.92	1.92	0.005	0.005	-162.635	--	--
4	1.92	2.57	0.0066	0.0066	-222.163	--	--
5	1.28	2.57	0.0083	0.0083	-222.163	--	--
6	1.92	3.21	0.01	0.01	-162.635	--	--
7	1.28	2.57	0.011	0.011	-59.5284	--	--
8	1.92	3.21	0.013	0.013	59.5284	--	--
9	1.28	3.21	0.015	0.015	162.635	--	--
10	1.28	3.21	0.016	0.016	222.163	--	--
11	1.92	2.57	0.018	0.018	222.163	--	--
12	45.55	3.85	0.02	0.02	162.635	162.635∠45°	162.635∠45°
13	39.77	3.85	0.021	0.021	59.5284	162.635∠75°	162.635∠45°
14	43.62	3.85	0.023	0.023	-59.5284	162.635∠105°	162.635∠45°
15	42.34	3.85	0.025	0.025	-162.635	162.635∠135°	162.635∠45°
16	39.77	3.85	0.026	0.026	-222.163	162.635∠165°	162.635∠45°
17	42.34	3.85	0.028	0.028	-222.163	162.635∠-165°	162.635∠45°
18	42.34	3.21	0.03	0.03	-162.635	162.635∠-135°	162.635∠45°
19	41.70	3.85	0.031	0.031	-59.5284	162.635∠-105°	162.635∠45°

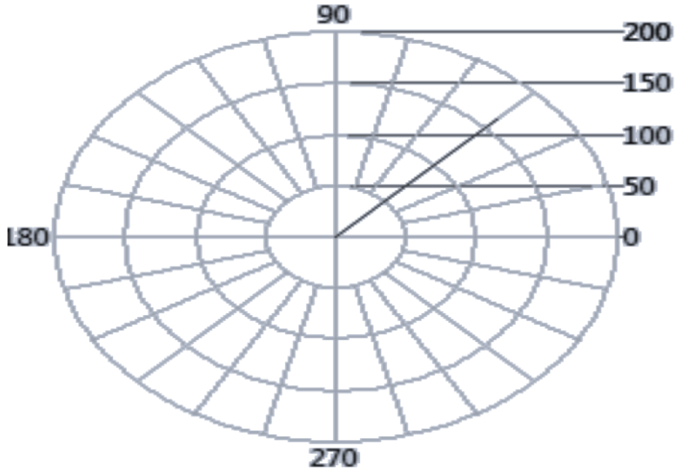

Fig. 4.6(a) Non-Recursive Phasor Estimation for first data window

(with fixed magnitude and phase angle=45°)

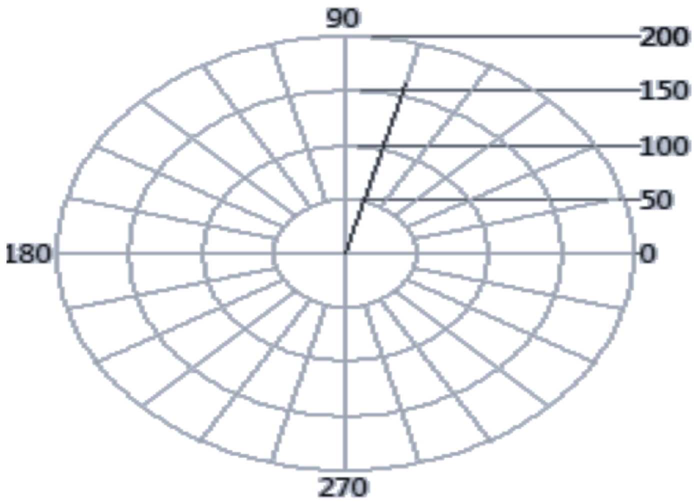

Fig. 4.6(b) Non-Recursive Phasor Estimation for second data window

(with fixed magnitude and phase angle=75°)

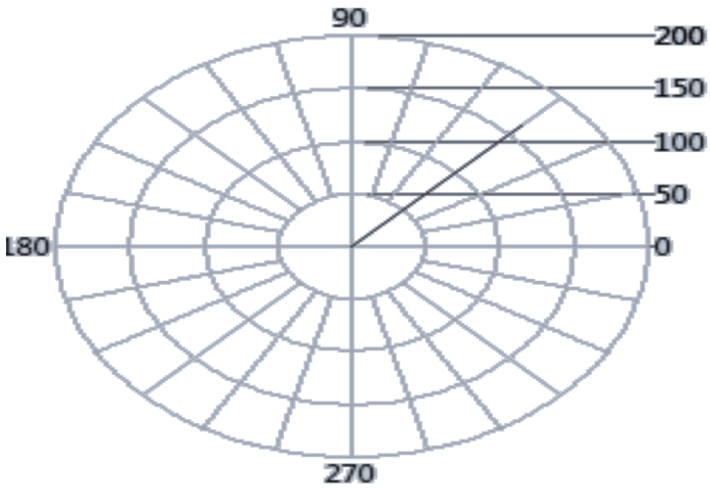

Fig. 4.6(c) Recursive Phasor estimation for first data window

(with fixed magnitude and fixed phase angle=45°)

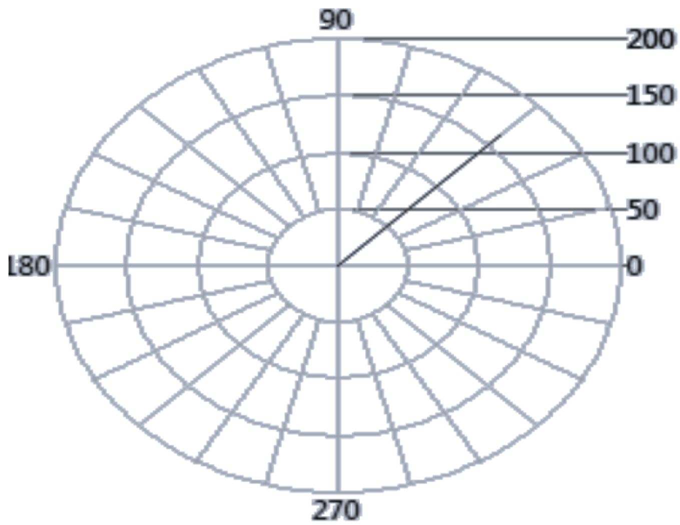

Fig. 4.6(d) Recursive Phasor Estimation for second data window

(with fixed magnitude and fixed phase angle=45°)

4.6 SUMMARY

Two models have been developed for estimation of phasors using PMUs i.e. (i) MATLAB based model and (ii) LABVIEW based model. Both the models consider DFT with non-recursive as well as recursive algorithms in phasor estimation. Out of the two models, MATLAB based model requires less computational effort compared to LABVIEW based model. Proposed models are quite simple to be developed, and may be effective for research purposes.