University of Strathclyde Glasgow

INSTITUTE FOR ENERGY AND ENVIRONMENT

University of Strathclyde

Department of Electronic and Electrical Engineering

Measurement, control and protection of microgrids at low frame rates supporting security of supply

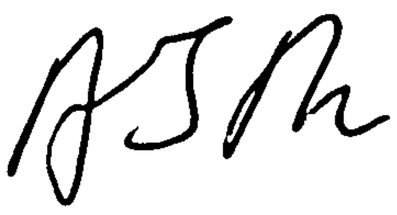
Volume 2

Andrew Roscoe

A thesis presented in fulfillment of the requirements for the degree of Doctor of Philosophy

February 2009

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has lead to the award of a degree.

Signed :  **Date :** 27th March 2009

Acknowledgements:

The author gratefully acknowledges the guidance given by Prof. Graeme Burt at the University of Strathclyde, the funding provided by Rolls-Royce, and the support of Dr. Nina Roscoe.

5 Frequency measurement

Within microgrids, as already described in section 2.1, system frequency is a much more dynamic parameter than it is in large national networks. This is due to the lower per-unit inertia of the generators and spinning loads, in combination with the larger sizes of commonly occurring load steps, relative to the system capacity and inertia. Thus, a measurement of frequency must respond fast enough to accurately follow the system frequency variations. The requirements for measurement accuracy, tracking and latencies were previously discussed section 2 and culminated in Table 2-14 and Table 2-15. This specifies a measurement latency of only 5 cycles (100ms) and a required accuracy of ±0.025Hz (±0.0005pu), with only 10 samples per cycle, and other interfering quantities such as harmonic content and flicker. The measurement of frequency to this accuracy with such a short latency is a specialised problem, to which "normal" techniques cannot be applied. In other fields of engineering, frequency measurement can generally take place over many cycles which allows much more conventional and accurate techniques to be used (Roscoe, 2005).

In addition to the requirement for a fast response time, tough requirements for "influence qualities" such as high harmonic content, power-line communications, flicker, and unbalance have all been laid down in section 2.7. The expected level of these interfering parameters is much higher within microgrids than it is for "normal" grid-connected power systems. This will be especially true for a battlefield and disaster-relief type scenarios when small power system networks might be operated outside the normal design constraints (and legal requirements) which would be applied for a distribution network. In such cases the risks of the following conditions are all increased: lower diversity of loads potentially injecting common harmonic currents, lower fault levels (higher impedances) causing higher levels of voltage harmonics due to harmonic currents, higher unbalance due to the difficulty of balancing single phase loads accurately across three phases, and large relative load steps relative to the size of the network leading to flicker and frequency deviations.

In addition to all these constraints, it is also desired to measure frequency using cheap equipment and processors, at relatively low sample rates so that the processor can also be used for other control/relaying tasks simultaneously (see section 2.9).

This chapter presents the design and performance of a new algorithm for frequency measurement. It can be called a "Clarke-FLL hybrid", being a combination of two entirely

separate frequency measurement algorithms based upon (firstly) a Clarke transformation and (secondly) a set of 3 single-phase frequency locked loops (FLLs). Such a hybrid is created because the two different measurement techniques have different strengths and weaknesses. The hybridisation enables the overall algorithm to exploit the strengths of both (and minimising the weaknesses) by selectively using one or the other depending upon dynamic circumstances. The algorithm provides measurements of frequency, amplitude and phase of a 3-phase signal set. The pre-existing work which directly contributes to the algorithm is the well-established mathematical Clarke transformation itself (Clarke, 1943), and some of the exact-time averaging and Fourier measurement blocks within the MATLAB SimPowerSystems blockset which were significantly enhanced in sections 3.2 and 3.5. The single biggest influence on the core of the final measurement design is the finding that the use of the optimised exact-time averaging techniques (particularly the novel extra ½-cycle averaging introduced in section 3.9) enable the Clarke transformation and Fourier algorithms to give significantly better results than other authors have previously achieved, even at very low sample rates. The work by Jovcic (2003) was of significant interest, both because he is the only other author to date to recognise the benefit of exact-time averaging, and also due to his use of 2nd harmonic cancellation techniques (although these ultimately did not contribute to the final algorithm, as described by section 3.10).

The final Clarke-FLL hybrid contains a large amount of Simulink code. This consists of many carefully placed uses of the exact-time averaging algorithm (optimised in section 3.2), the novel ripple remover (designed in section 4.3), and a significant quantity of code which implements:

- fault ride-through
- fast settling when a new or fast-changing signal is applied, including seeding of the FLL with data from the Clarke's measurement when appropriate

These two functions contain large amounts of code to implement logic, buffering, and hysteresis delays. All this code was created from scratch during this work.

In the section 5.7, the performance of the Clarke-FLL hybrid is trialled against 7 other candidate frequency measurement systems. It is shown to perform the best, and is thus the solution of choice. Benchmarking analysis is also performed to show that the computational load presented by the Clarke-FLL hybrid is competitive compared to the other solutions, despite is apparent complexity.

5.1 Established and published methods

Within a nationwide power system, adequate and accurate frequency measurement has traditionally been made with relatively simple techniques such as zero-crossing detection. These techniques have been adequate due to a combination of factors:-

- Smaller numbers of frequency measurement points required means that equipment can be relatively expensive
    - fast sample rates, sometimes into the 20 kSa/s range
    - precision analogue hardware front-ends
    - precision ADCs (analogue-digital converters)
- averaging over many cycles, which has been acceptable because the expected rate of change of frequency (ROCOF) has been low
- reliance upon the sinusoidal nature of the voltage waveforms
- reliance upon balanced operation, so that only one phase needs to be measured

In the following sub-sections, a summary review of established and proposed methods for frequency measurement within AC power systems is presented. Some of the proposed methods use novel techniques. However none (in their presented or raw implementations) are found to meet the requirements of chapter 2.

5.1.1 Frequency measurement by zero crossings

Traditional commercial relays have tended to use zero crossing algorithms to measure frequency. By taking samples, it is possible to find the pairs of samples which contain the zero crossing. Linear interpolation between these two samples can then be used to calculate the time at which the zero crossing is believed to have occurred. It is interesting to perform an error analysis for this frequency measurement method, using a purely sinusoidal waveform input, as a benchmark against which to analyse other frequency measurement methods and scenarios.

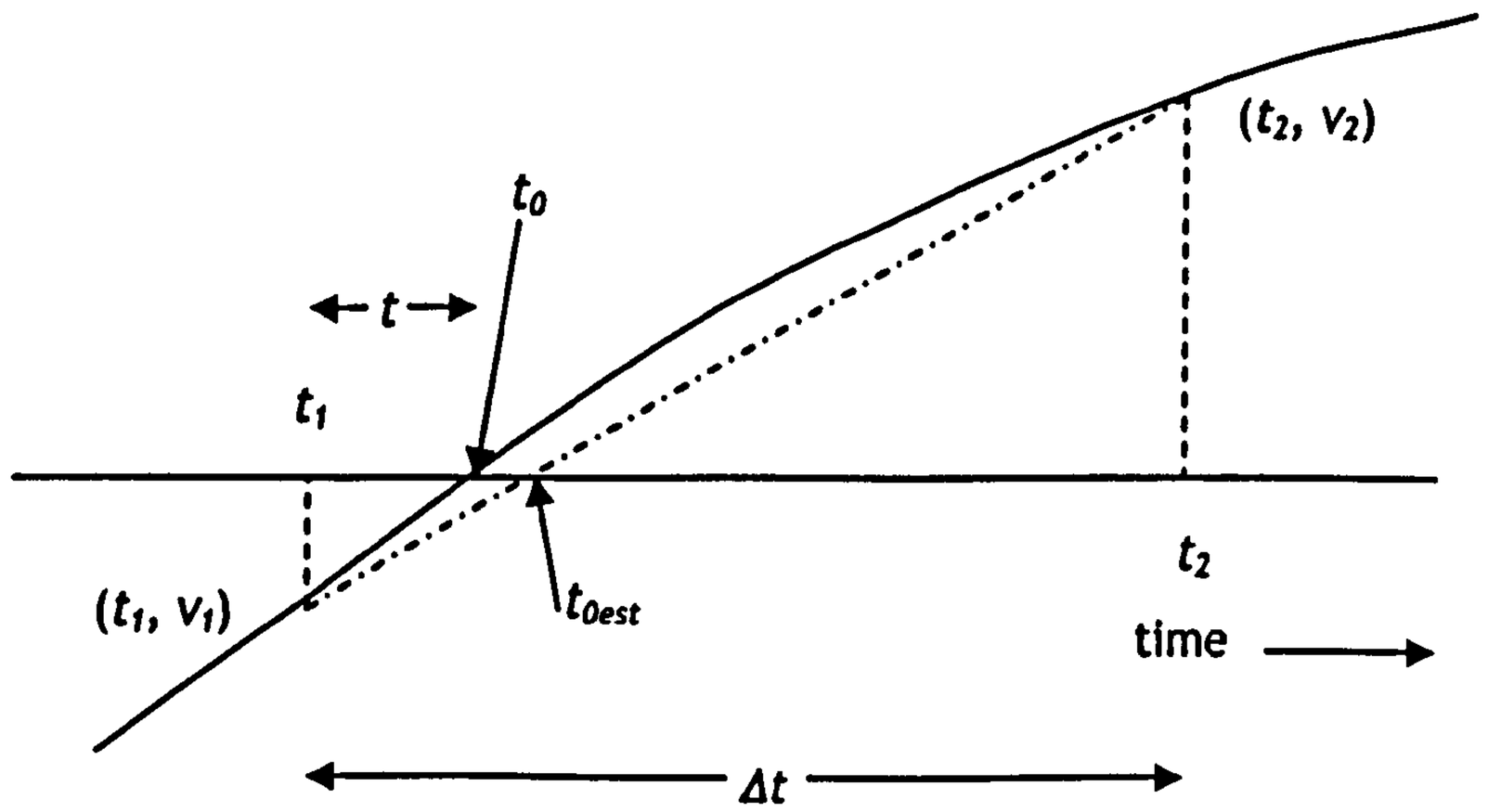


Figure 5-1 : Linear interpolation to estimate the time of a zero crossing

At a zero crossing point, the actual zero crossing occurs at time t_0. The sample interval is Δt_s. The first sample v_1 occurs at $t_1=t_0-t$, and the second sample v_2 occurs at $t_2=t_0-t+\Delta t_s$. If one of the sample values v_1 or v_2 is 0, then by examination the actual position of the zero crossing time is known to be at t_1 or t_2 respectively. Normally, however, this is not the case, and linear interpolation is carried out to determine an estimate of t_0, given by t_{0est}.

The estimate of t_0 can be made using the following equation:

$$t_{0est} = t_1 - v_1\left(\frac{t_2 - t_1}{v_2 - v_1}\right)$$ where v_1 and v_2 are the signed voltages (v_1 negative and v_2 positive in Figure 5-1)

(5.1)

The effect that this single zero-crossing measurement error will have on a frequency measurement at 50Hz is given by:-

$$F_{error} = \frac{1}{T} - \left(\frac{1}{T + (t_{0est} - t_0)}\right)$$ where T is the actual period of the sinusoid.

This expression simplifies, assuming that $t_{0est}-t_0 \ll T$, to:

$$F_{error} = \frac{(t_{0est} - t_0)}{T^2} = (t_{0est} - t_0)F^2$$ where F is the actual frequency

(5.2)

It is relatively trivial to now analyse the frequency error induced by the error $t_{0est}-t_0$ for different values of sample interval Δt_s and different values of the time offset t between the first sample and the zero crossing. At 50Hz, and using a 1 per-unit voltage magnitude,

v_1=sin(2π50(t_1-t_0)) and v_2=sin(2π50(t_2-t_0)). These values for a pure sinusoidal waveform can be substituted into (5.1) for different values of Δt_s and t to evaluate t_{0est}-t_0. For simplification, t_0 may arbitrarily be set to 0 for this evaluation.

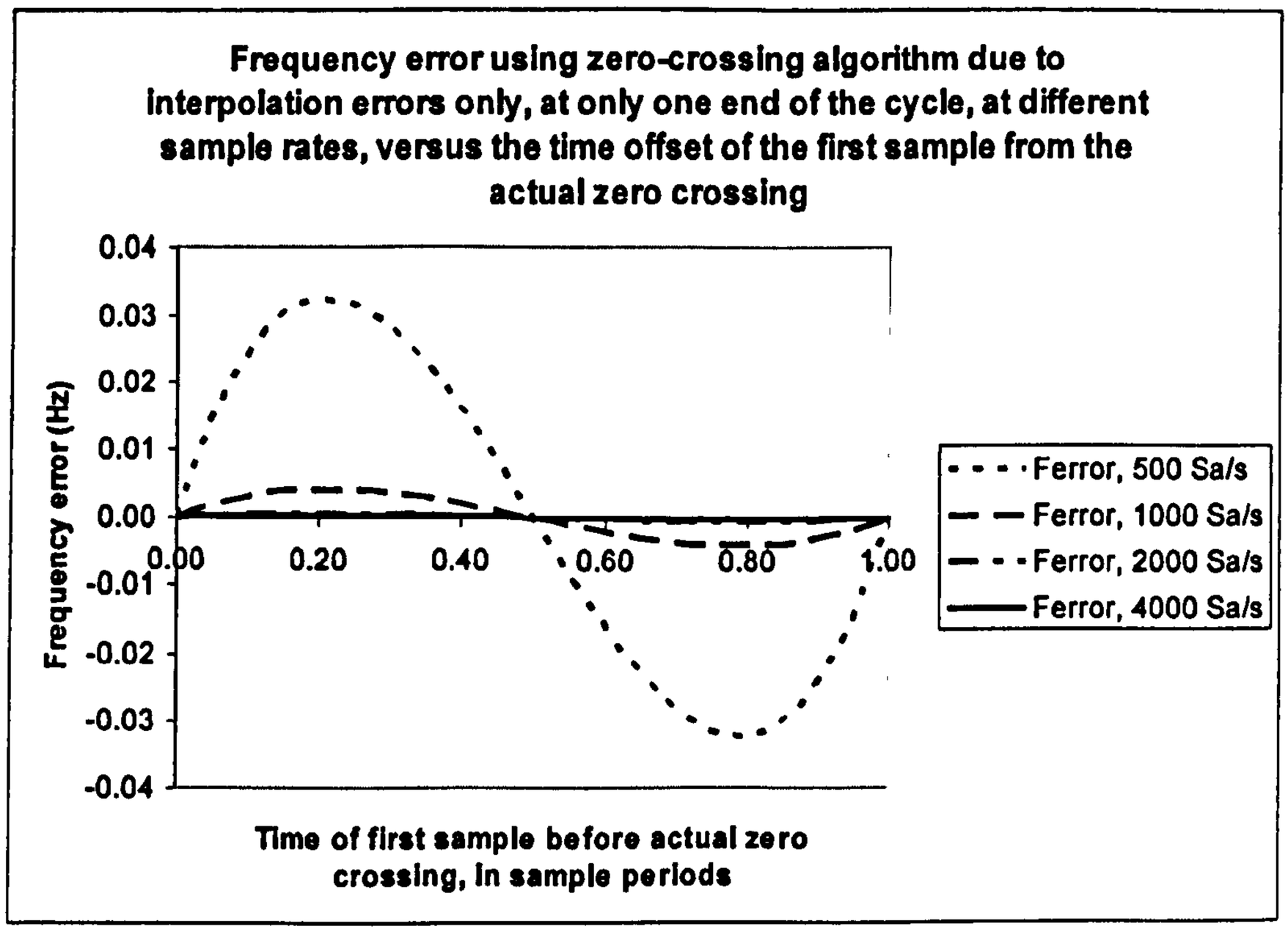


Figure 5-2 : Frequency error of zero crossing algorithm using pure 50Hz sinusoid inputs

Clearly, from Figure 5-2, the zero crossing algorithm doesn't work well at sample rates below 1000 Sa/s (20 Sa/cycle), when the sample instants do not coincide with the actual instants of the zero crossings. This is the reason that most traditional digital frequency measurement devices use one of the two following approaches:-

a) a very high sample rate, above 4 kHz and reportedly as high as 24 kHz (480 Sa/cycle), and/or

b) front-end sampling hardware which is clocked at a variable rate by a phase-locking system which targets samples at the exact instants of zero crossings. In this case, the performance of the hardware phase-locking PLL system will also affect the accuracy and dynamic response of the measurement.

Aghazadeh (2005) presents a method of reducing the errors, compared to the analysis performed above, for a zero-crossing measurement. However, the resulting frequency errors are ±0.05Hz when representative noise and harmonics (commensurate with the levels applied in this thesis) are included on the input signal. This is an unacceptable error level.

5.1.2 Methods based upon the Clarke transformation

An appealing method for measurement of system frequency in a 3-phase system is the use of the Clarke transformation (Clarke, 1943) to transform the three-phase voltage measurements into a rotating vector in 2-dimensions, by the standard equation:-

$$\begin{bmatrix} A \\ B \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & \frac{-1}{3} & \frac{-1}{3} \\ 0 & \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{3}} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} * \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \tag{5.3}$$

The angle of the vector $\begin{bmatrix} A \\ B \end{bmatrix}$ rotates at the system frequency. Thus, system frequency at any instant can be defined mathematically as

$$f = \frac{\partial}{\partial t}(\arctan 2(B, A)) \tag{5.4}$$

Where arctan2(B,A) is the well-known version of arctan that returns a correct answer over the full 4 quadrants of the plane, and also avoids division by zero errors. Also, a subtlety is that when arctan2(B,A) crosses the boundaries at ±π, 0 or 2π, an algorithm must correctly "unwrap" the phase to give a correct answer for f. Several authors make use of this technique to measure frequency.

Cantelli (2006) uses a version of the Clarke transform method, specifically because it can ride "seamlessly" through dips on single phases, whereas a single-phase frequency measurement is vulnerable to dips on that phase. The problems of phase unwrapping are avoided because the method does not use the atan() function but instead estimates the phase change from the Cartesian coordinates. There is a small time saving by avoiding the atan() function, although the evaluation does require a sqrt() function instead (see Fig. G-1 for relevant benchmarking experiments). The method is also vulnerable when sample rate is low, as the approximation to the atan() function then becomes less valid. Cantelli's method does not carry out exact-time averaging over ½ or one cycle, and so there are ripples at the output of his algorithm in the presence of harmonics or unbalance. This is reduced by a PID filter which of course introduces an IIR response characteristic. Cantelli (2006) ignores the effects of unbalance, which is a significant oversight, particularly as

unbalance can reach 100% during an unbalanced fault. No mention is made of the sample time used in any of the analysis. From the graphs presented in the paper it appears to be >>10 samples per cycle.

5.1.3 Methods based upon Fourier transforms

Aside from zero crossings, Fourier transforms are the most obviously suitable ways to measure the frequency of a sinusoidally shaped waveform, from a purely mathematically point of view. This is because they correlate the measured waveform with a synthesised sinusoidal waveform, in the real or complex domain. The transforms can take the form of an individual discrete Fourier transform (DFT), carried out for one exact frequency correlation (as used in this thesis), or as a repeated set of transforms at numerous correlation frequencies to provide an array of data across a frequency range. The FFT (Fast Fourier Transform) is a specialised form of this, which reduces the computational effort but imposes restrictions on the input and output data streams (regular sample times and fixed output frequencies). A single DFT calculation requires (many) sine and cosine evaluations, while an FFT calculation required many floating-point operations. Due to the computational effort required to calculate the DFT/FFT, such methods have taken some time to become acceptable within power systems applications. Instead, such methods as zero crossings and Walsh functions (Johns, 1995) have been used.

More recently, microcontroller CPU speeds (and particularly the floating-point capabilities) have gradually advanced to an extent that Fourier analysis can be carried out directly, so long as care is taken within the coding. Early digital implementations of Fourier analysis within power systems relays were simplified within the algorithms to use such things as sine/cosine lookup tables, or lookup tables for an entire sine/cosine correlation waveform at a fixed sample rate (Moore, 1996a). This type of implementation suffered from ±0.04Hz errors for input signals off-nominal frequency, due to errors resulting from limitations of lookup tables. An example of a very poorly coded Fourier analysis can be seen in Lin (2005). At the present time, analysis of Appendix F & Appendix G shows that sine and cosine operations can be calculated within modern microcontrollers almost as quickly as a lookup table can be accessed. This, particularly when combined with the realisation of exact-time averaging algorithms (see section 3.2) which can provide rolling-buffer operation, means that much more effective and accurate Fourier analysis algorithms can be coded in real-time using cheap microcontrollers.

5.1.4 Phase-locked loops (PLLs)

The MATLAB Simulink SimPowerSystems blockset provides code which implements both single and three-phase PLLs. These perform reasonably well, as they contain single-cycle

exact-time averaging. They are used for 2 of the 8 methods compared in trials presented later in this thesis.

Chung (2000) presents a quite conventional digital three-phase PLL using the dq frame, without any special techniques such as signal level normalisation or exact-time averaging. The resulting output suffers ripple in the presence of unbalance and harmonics. No mention is made of the sample time used in any of the analysis. From graphs it appears to be >>10 samples per cycle.

Awad (2005) describes a software PLL system. He introduces a useful normalisation of the [AB] vector to unit length before carrying out the phase detection, which in a PLL can be done by looking purely at the magnitude of the q component after transformation to the dq frame. This normalisation helps the PLL to retain the same dynamic performance during voltage dips. This normalisation, however, would not be required if the atan2(q,d) function was used to measure the actual phase rather than the estimated phase. Awad (2005) also uses a delay-signal cancellation technique to separate out the positive and negative sequences of the AB vector, so that the transformation to the dq frame contains only the positive sequence component. This is done, presumably, to minimise PLL ripple due to unbalance (but not harmonic content). No mention is made of the sample time used in any of the analysis. From graphs presented it appears to be >>10 samples per cycle. The delayed-signal cancellation is an approximation which will not be correct at all times, and assumes that system frequency is always at or very close to nominal. In a microgrid scenario, this is not necessarily true. The delayed-cancellation technique in Awad (2005) would not be required if an exact-time averaging algorithm over exactly ½ or 1 cycle was applied to the dq data. This would remove all ripples due to unbalance, and also harmonics.

Han (2006), presents a single-phase PLL, which adapts the sensitivity of the phase detector by using an ALC (normalisation) control loop so that during voltage dips the PLL retains the same gain. This concept is also used (but in a more comprehensive manner) in a PLL designed by Jovcic (2003).

Jovcic (2003), presents a PLL which does address the issue of low sample rate. This is done by the use of the SimPowerSystems block "Discrete Variable Frequency Mean value". Jovcic (2003) also introduces the concept of 2nd harmonic cancellation. These concepts have been analysed and extended in sections 3.2 and 3.8 of this thesis. Jovcic's PLL is a single-phase device, although the concepts extend readily to three-phase operation. This

PLL has been recreated in detail, exactly as described by the paper. During this work it was found by simulation that below 2000Sa/s the feedback loop inside the algorithm for voltage magnitude estimation can become unstable under high levels of harmonic contamination. At sample rates of 2000Sa/s and above, the PLL operates as published.

Jovcic's PLL therefore includes several useful techniques, and the performance of the phase-locked loop itself is very good. However, being a PLL, the author proposes that it is not the best way to measure frequency (see section 5.3). Also, the algorithm does not include any fault ride-through mechanisms (see requirements section 2.7.5). This thesis proposes (but does not explicitly prove) that if the frequency measurement algorithm described later in section 5.4.2 were used in combination with a PLL such as Jovcic's, the resulting algorithm would outperform the Jovcic PLL as published. This is because the frequency measurement algorithms proposed here can settle more quickly without overshoot, ride through faults better, and attenuate noise better, than the phase-tracking part of a PLL. The output of a better frequency measurement (not from a PLL) could in future be used to drive the exact-time averaging algorithms inside a PLL phase detector, to create a hybrid frequency measurement and PLL algorithm.

5.1.5 Kalman filters

Dash (2000) proposes using extended Kalman filters. This thesis does not propose to utilise Kalman filters due to their similarity to low-pass filters and the inferior rejection of noise and ripple offered by these compared to adaptive FIR filters (see section 3.3). The algorithms proposed by Dash have not been finalised and appear to be unstable under certain conditions. The graphs of results are also scaled in such a manner that the errors cannot be resolved to useful accuracy. No mention is made of the sample time used in any of the analysis. From graphs presented it appears to be >>10 samples per cycle.

5.1.6 Other methods

Choi (2006), attempts to use a phase-delaying technique on a single-phase voltage waveform. This is done to create two voltages from one, with, hopefully, one lagging 90 degrees behind the other. This can be processed in a similar way to the AB vector in Cantelli (2006). No mention is made of the sample time used in any of the analysis, and the diagrams suggest the system has actually been modelled in the "continuous" mode.

The results show ripple of the order of ±1Hz which is unacceptable. The delayed signal technique is far worse at removing unwanted effects of harmonics than exact-time averaging.

Salcic (2000) presents an extremely convoluted method which does not assume that the waveform is sinusoidal but instead simply looks for symmetry around the positive peaks, and then attempts to fix a timestamp at each peak. The time between timestamps is then used to deduce frequency, like a zero crossing algorithm. The sample rate requirements for this method are *Fmax.(Fmax/Fresolution)* = 53*53/0.1 = 28090Hz to achieve even a 0.1Hz resolution, which is unacceptable within the scope of this thesis. Waveforms with substantial THD (35%) are used as test waveforms. However, the resulting errors even with 28kSa/s sample rate, over 10 cycles, are still about 0.01Hz. This is poor performance for such a high sample rate. With a 1 cycle measurement, the errors are about 0.08 Hz. The algorithm also appears to have a quantised output; i.e. the output for the 1-cycle case is 49.925Hz, 50Hz or 50.075Hz, and cannot take values in between. Two fundamental problems within this paper are that the algorithm is looking for a maximum in a waveform, which is always numerically inaccurate compared to locating a zero, and that the algorithm appears unable to interpolate between samples so the sample rate required becomes unnecessarily large.

Lin (2005) describes an interesting algorithm (for both amplitude and frequency measurement) based upon adaptions of wavelet transforms. The sample rate used is also suitable, at 600Sa/s, and this method might be worth more analysis in the future. No noise or ADC quantisation is applied, however, and the scales on the graphs do not allow a good analysis of the performance. His proposed method is compared to a Fourier analysis (which is the proposed base measurement used in this thesis), but this appears to have been deliberately coded very poorly to give bad results for comparison.

Lopez (2008) describes a 9-sample technique (triplet of triple samples), which claims to be fast responding. However, similarly to the 2 and 3-sample amplitude measurement techniques (Johns, 1995), the method is very susceptible to noise and ripple due to harmonics. All the dynamic performance advantage of the technique is lost because the samples must be pre-processed by a bandpass filter which introduces a delay of at least 2 cycles to achieve even marginal performance (±0.02Hz accuracy with 40dB SNR at 128 samples per cycle). The technique has been coded in Simulink by the Author and improved with averaging techniques to create a 5-cycle measurement and compared to the proposed method of this thesis (subsequently to the text of this thesis being substantially complete). The Lopez (2008) method compares poorly. In particular, in the presence of harmonics the algorithm produces biased results with an over-estimate of frequency which cannot be removed by any amount of post-filtering. Thus, the algorithm pre-filters must be able to provide almost perfect sine waves - an almost impossible task without adding substantial group delay. In test, the method over-estimated frequency by up to 0.3 Hz for high

harmonic content (28% THD) signals, even with 2 cascaded 125 Hz low-pass pre-filters.

5.2 Summary from literature search for frequency measurement

- Most published papers discuss PLLs (Phase-Locked Loops), of which the primary function is to control power-electronic devices. The estimate of frequency is a by-product of this process.
- Few authors appear to fully prioritise the reduction of sample rates to enable the algorithms to be combined in multi-function systems. Exceptions are Lin (2005) who uses a sample rate down to 600 Sa/s, and Jovcic (2003), who addresses some of the problems due to lower sample rate, although the actual sample rates targeted or used is not described.
- Some authors apply harsh levels of harmonics, and some present unbalanced scenarios. However, in most cases the graphical results presented either show relatively poor errors with noise, ripple, or IIR settling characteristics, or show errors which have been hidden by graphs with wide scales.
- Most of the published algorithms can be either made redundant or significantly improved by use of an exact-time averaging algorithms. These are used by Jovcic (2003), and have been significantly extended/improved in this thesis (see sections 3.2 and Appendix G). Use of these blocks allows removal of much of the ripple due to unbalance and harmonics, and also removes noise more effectively than a low-pass filter (see chapter 3).

5.3 Why not to use a PLL for a measurement of frequency

During the course of this thesis, many different candidate methods for frequency measurement were investigated. Most were eventually rejected, and not all are presented here. Significant time was taken creating phased-locked-loop (PLL) solutions, not least because a 10kVA 3-phase grid-tie inverter with islanding capability was designed, built, coded and tested by the author. These phased-locked-loops were either single-phase units or three-phase dq analyses. Generally the algorithms, at the core, were very similar to the SimPowerSystems blockset algorithms "Discrete 1-phase PLL" and "Discrete 3-phase PLL". Variations on these algorithms which have been developed include:-

- Normalisation of the dq vector to unity magnitude, to standardise and stabilise the response during voltage dips/surges.
- Replacement of the phase detector (the measured value q from the dq vector) in the three-phase unit, with an atan2(q,d) result which allows faster locking or

re-locking when phase error is significant. This helps particularly when the dq vector crosses into the left quadrants of the dq plane, i.e. when d is negative and the PLL is significantly unlocked.

- Enhancement/improvement of the exact-time averaging algorithms as detailed in section 3.2.
- Experimentation with other post-PLL smoothing techniques for the estimation of frequency. (SimPowerSystems blocks use a slew-rate limit of ±12Hz/s combined with a 2nd-order low-pass filter set to 25Hz cut-off).
- Creation of PLLs with ½-cycle base correlations, and N-cycle base correlations. These allow faster response with more noise, or slower response with smoother outputs.
- Addition of ride-through capability during faults (particularly relevant for inverter drive PLLs)
- Additions which allow the PLLs to be seeded from an external estimate of frequency. The external estimate should be fast responding and robust, but can be noisy. This allows the PLL range to be extended over more than an octave which is unconventional. A seeding algorithm detects when the PLL frequency is substantially different to the robust seed frequency (with additional time qualifications), and when necessary resets the PLL to the seed frequency to re-lock it quickly. This allows extended frequency range since accidental locking to sub-harmonics or harmonics is no longer possible. It also significantly speeds up initial locking when a signal is first applied and the PLL has previously been free-running with a noise-only input.

Despite all the work on PLLs, in particular the seeding algorithms, this thesis proposes that they are not the best way to measure the parameter frequency. They lock to phase very well, which is what they are fundamentally designed to do. Because they lock to phase, however, transients cause an IIR response through the PI controller, and a step change in frequency always results in a response which overshoots. To demonstrate this, a simple Simulink simulation was created. This contains several PLLs:-

- SimPowerSystems 1-phase PLL
- The PLL designed by Jovcic (2003)
- the author's single-phase PLL based upon a single-cycle averaging of the q error, including the improved exact-time averaging algorithm. This has the same loop gains as the SimPowerSystems PLL.
- the author's single-phase PLL based upon a ½-cycle averaging system, with higher

loop gains and faster response.

- the author's single-phase PLL based upon a 1½-cycle averaging system, with lower loop gains and a smoother response.
- the author's three-phase PLL, based upon a 1-cycle averaging system of atan2(d,q). This has the equivalent loop gain of the SimPowerSystems 3-phase PLL block.

These PLLs were exposed to a waveform of a constant frequency 49.95Hz, at magnitude 1pu until t=1 second, when the frequency undergoes a step change to 50.05Hz. Figure 5-3 shows the response of the different PLLs.

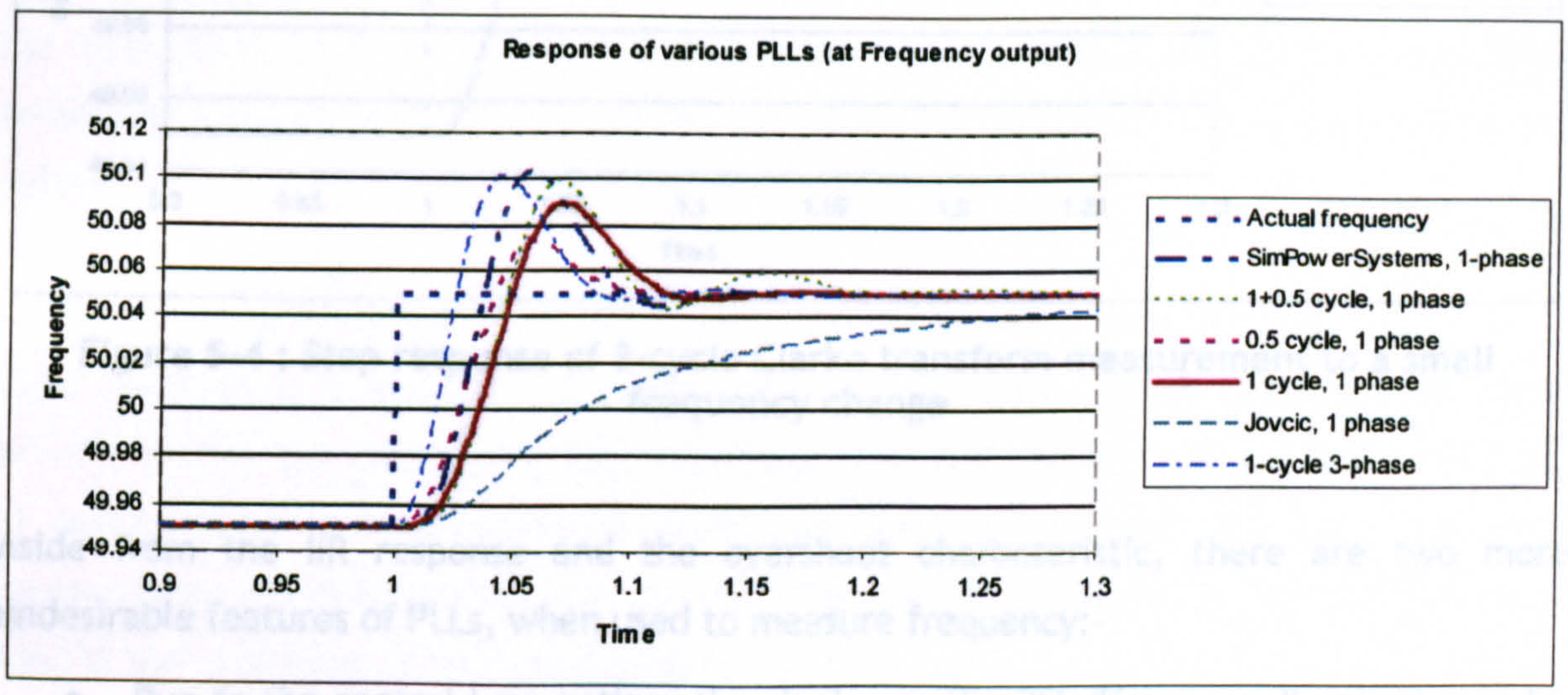


Figure 5-3 : Step response of various PLLs to a small frequency change

The responses of the PLLs exhibit two characteristics in Figure 5-3 which are undesirable. Firstly, the responses all overshoot, except for that of the Jovcic (2003) PLL, because it is so heavily filtered that its response is too slow. Secondly, the IIR response of the PI control loops means that the frequency measurement is still settling up to 200ms after the step change was applied. This is too slow to meet the requirements of 3-5 cycles (60-100ms) defined in section 2.1. It must be noted that a frequency seeding algorithm was included in the authors' PLLs for the above experiment, but since the frequency step is very small, the seeding algorithm quite rightly does not take action. This is because the frequency seeding can only seed the frequency and not the phase, so is only of benefit when the PLL has lost lock completely. This does not occur during the above simulation.

To contrast with Figure 5-3, Figure 5-4 shows the response to the same waveform of a simple measurement using the Clarke transform. In this case, the algorithm uses initial base averaging over exactly 1 cycle, cascaded into a second 2-cycle exact-time average

filter. This creates an excellent filter combination to reject noise, harmonics, unbalance and integration/interpolation errors due to low sample rates, as described in chapter 3. Clearly, the frequency measurement rises over 3 cycles (60ms) in a controlled fashion, with a closely defined FIR response and zero overshoot. This is preferable by far to any of the responses shown in Figure 5-3.

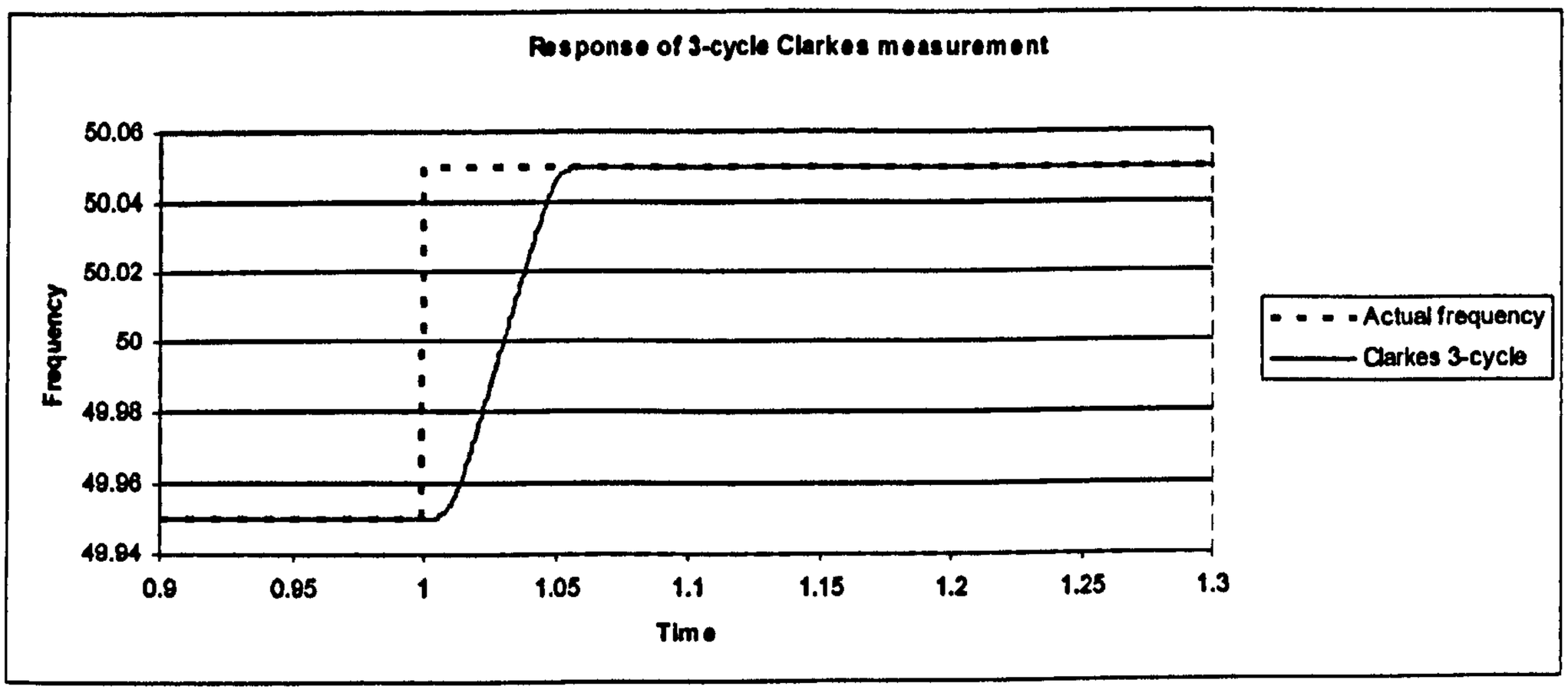


Figure 5-4 : Step response of 3-cycle Clarke transform measurement to a small frequency change

Aside from the IIR response and the overshoot characteristic, there are two more undesirable features of PLLs, when used to measure frequency:-

- Due to the control loop action, the ripples at the PLL "frequency" output may be at high frequency, but not at multiples of the fundamental frequency, even for situations where there are no aliased harmonics. This makes removal of the residual ripples much harder than for the FIR-based measurements. As shown in chapter 3, the FIR-based measurements using 1 and ½-cycle averaging create ripples at integer multiples of the fundamental for all the following cases, and these ripples can then be almost entirely removed by cascading a further exact-time average step. This almost completely removes the errors due to:-
    - Integration/interpolation
    - Un-aliased harmonics
    - Unbalance (for 3-phase AB and dq-based analyses)
    - Errors in the estimate of frequency used to dynamically set the parameters within the FIR exact-time averaging filters.
- During a phase jump, the response of a PLL is a necessary swing (undershoot followed by overshoot or vice-versa), to bring the PLL back into lock. The PLL frequency does not truly represent the waveform frequency during this time. This is the reason that PLLs such as the SimPowerSystems block and the Jovcic (2003)

implementation use slew-rate limiters and low-pass filters to massage the inner-loop frequency before using it as the "best estimate" of frequency. For a FIR filter which only measures frequency and is not locked to phase, the frequency measured will still contain a "blip" following a phase jump, but this "blip" is one-sided and does not contain the large overshoot-undershoot characteristic of a PLL. This is a useful feature, considering that switching large loads within microgrids can have exactly this effect due to the changing phase across a reactive component such as a transformer when real power flow changes significantly. For example, Table 2-15 explains how an almost instantaneous 10 degree phase jump could easily occur within a microgrid.

For all these reasons, the author does not believe that the use of a PLL is the best way of measuring frequency. This thesis instead proposes that techniques based upon FIR filters, without any form of PI control loop, are more appropriate.

5.4 Description of the eight methods trialled

Eight methods of estimating frequency are presented here. Six of these are described briefly, as they are used mainly for comparison. The best two methods are described first, in detail, since a combination of these two methods produces the best frequency measurement achieved to date, against the requirements of chapter 2. The test waveform used is that described in sections 2.10 & 2.11.

5.4.1 Method 1 : Clarke transformation with ride-through

The basic technique for frequency measurement using the Clarke transform was introduced in section 5.1.2. The algorithm shown below uses this technique with a number of enhancements. These allow the algorithm to switch between different modes and give fast initial settling, followed by more averaging during steady-state operation, combined with fault ride-through capability. Important considerations are pre-loading of filters during ride-through and settling, and also internal checks for validity of the final answer. These lower level details of the algorithms are all required to make this measurement as responsive as possible, and to optimise its potential for use as a seed algorithm for other frequency measurement methods. This algorithm is extremely robust and can cope with all frequency and amplitude scenarios, unless the signal amplitudes are below the noise, or there is a two-phase fault (line-line or line-line-ground). Frequency can be measured (hardware instrumentation permitting) from DC to almost the Nyquist frequency of the frame rate, without any danger of locking to sub-harmonics or harmonics. Only one "hard" maths function is required: an "atan2" function to determine the rate of change of angle of the AB vector.

Figure 5-5 (which is split over two pages) shows the highest level detail of this algorithm. Figure 5-6 through Figure 5-8 show lower-level details. Important features of the algorithm are now described:

Inside the measurement core (Figure 5-6), the Clarke transform is performed, and the rate-of-change of the AB vector angle gives the instantaneous frequency on a frame-by-frame basis. This contains noise due to all the influence qualities; in particular the effect of unbalance is to create a ripple in this measurement. Averaging over a single full cycle removes most of the ripple due to harmonics and unbalance. The effects due to Gaussian noise are still substantial, however, as the average over one cycle is the definite integral of the sample-by-sample derivative of atan2(B,A). The effect of this is that at any time, the output of the 1-cycle measurement is actually deduced from only 4 sample values, despite that fact that there may be significantly more samples than this taken during a single cycle.

The frequency can be further averaged over another 4 cycles to create a measurement with a 5-cycle response time. This can be done using a single FIR filter, or a combination of FIR filters. An analysis of candidate filter combinations was carried out, using z-domain filter analysis and bode plotting. The options considered were:-

- Single FIR average over 4 cycles
- Two cascaded averages over 2 cycles each
- ½-cycle averaging followed by 3½ cycle averaging

As previously found in sections 3.3.2 & 3.3.4, the cascaded filters provide better rejection of higher-frequency noise than the single FIR filter. In this case, the best choice is a cascaded pair of 2-cycle averages, which are seen implemented in Figure 5-6. The bode plots for the three considered filter options are shown in Figure 5-9 to Figure 5-11.

An important explanatory note regarding Figure 5-6 is the action of the "RideThrough" and "Standby" inputs. The "Standby" input is active when it appears that there is no valid set of inputs (measurements of voltages on phases A, B & C). In this case, the averaging filters are filled with the nominal frequency. This helps to minimise settling errors if a signal at nominal frequency is subsequently connected. Also, during ride-through, the filters are filled with the frequency which is assumed to be the correct frequency during the ride-through timeframe. This again helps the measurement settle if input signals re-appear at a similar frequency when a fault is cleared.

Measure the frequency of a 3-phase signal, via Clarkes with ride-through

Andrew Roscoe, 2007

This is a very robust algorithm:-
- capable of measuring down to DC (hardware input filters allowing)
- capable of measuring up to 1/(3*Tps)
- Never gets stuck on harmonics or subharmonics
- Rides through short faults
- Can ride through sustained single-phase faults but NOT two or three-phase faults (line-line or line-gnd)

Always provides about the right answer unless there is a 2-phase or 3-phase hard fault.
The fully processed output is averaged for smoothing, and also rides through voltage disturbances for the allowed time.
Other outputs have less averaging and can be used for noisier, quicker measurement.

```
Hard math function count:-
sin/cos/tan   : 0
asin/acos     : 0
atan,atan2    : 1
sqrt          : 0
x^y           : 0
e^x, 10^x     : 0
ln, log10     : 0
```

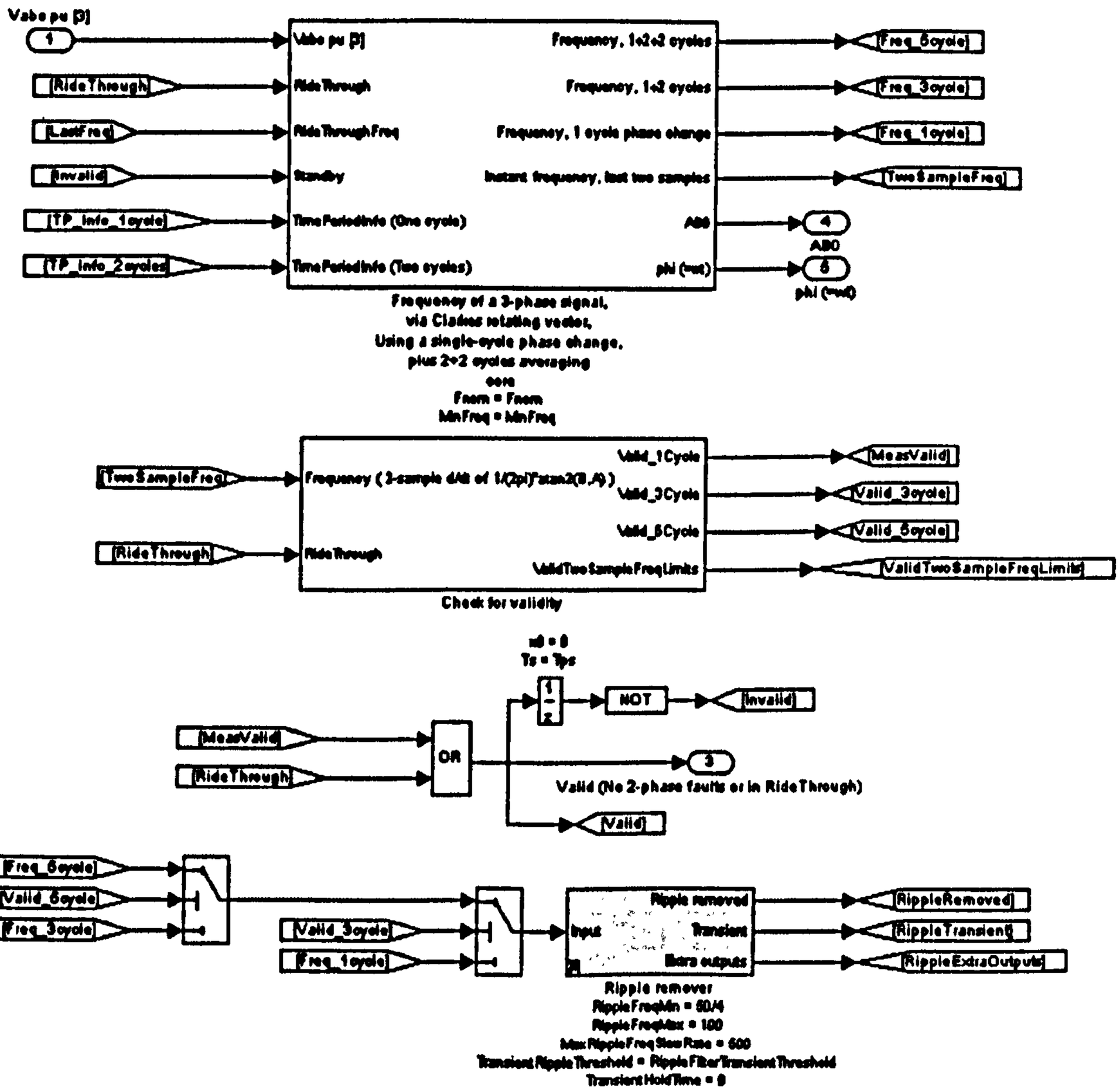
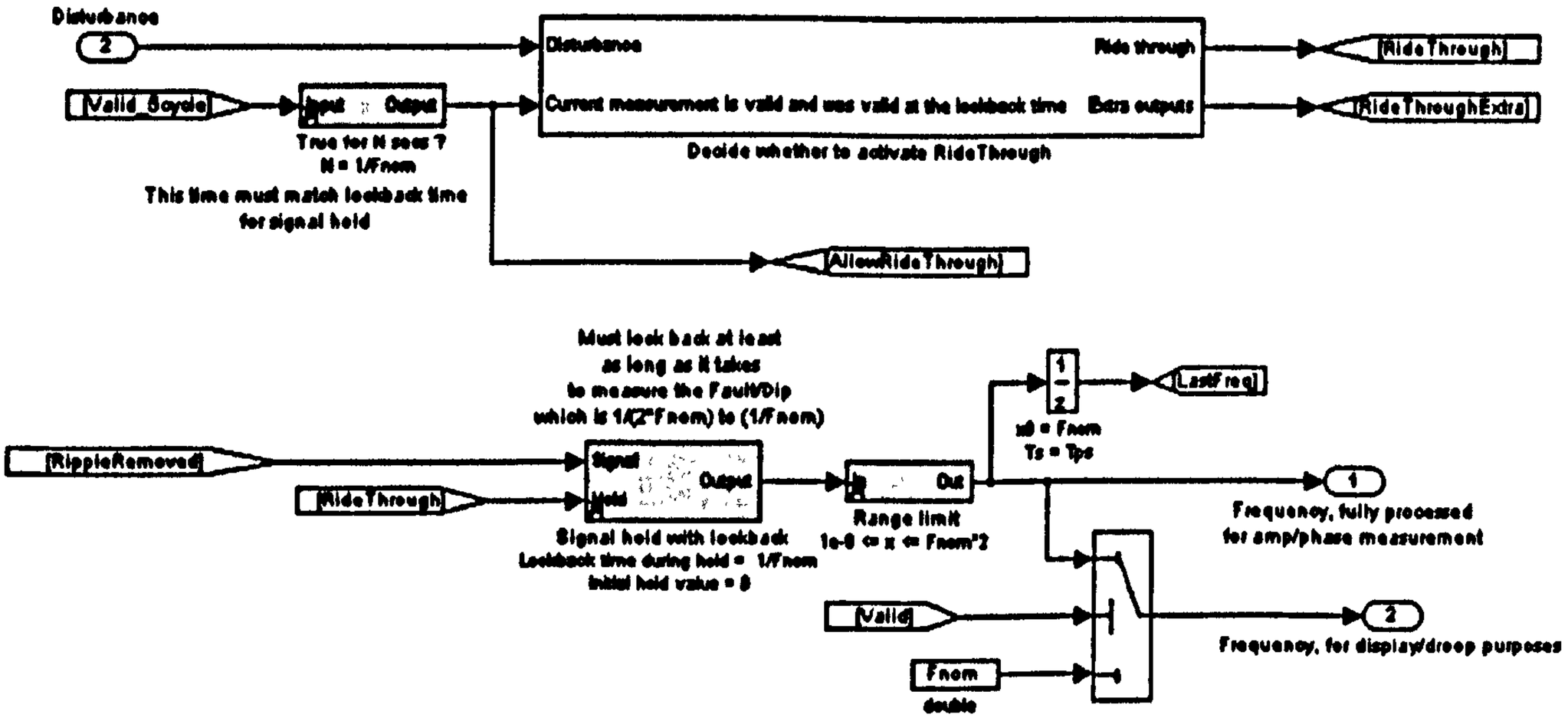
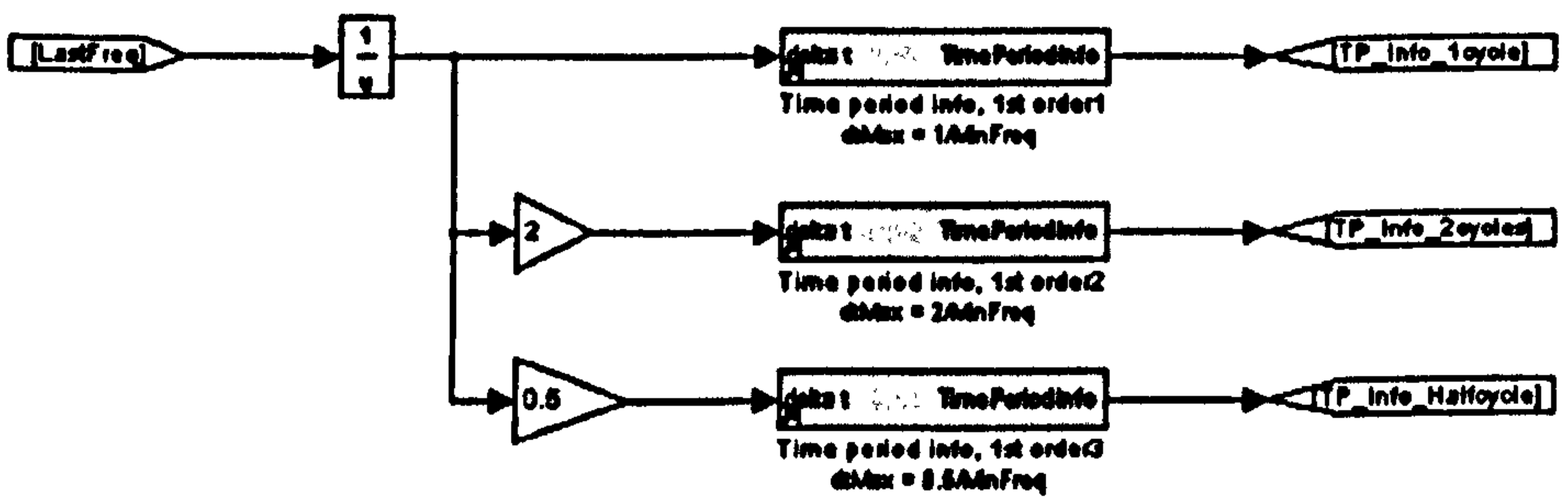


Figure 5-5 : Clarke transform frequency measurement algorithm - detail (1) [this and next page]



Here, the averaging time will be clipped at the upper end to 1/MinFreq.
Make MinFreq quite high, say Fnom*4/5. This keeps the response time small when frequency looks small.
This is okay, since the phase trajectory is nominally a straight line,
so the errors due to non-exact-cycle averaging for input frequencies less than Fnom*4/5 are small,
only due to gross unbalance etc.



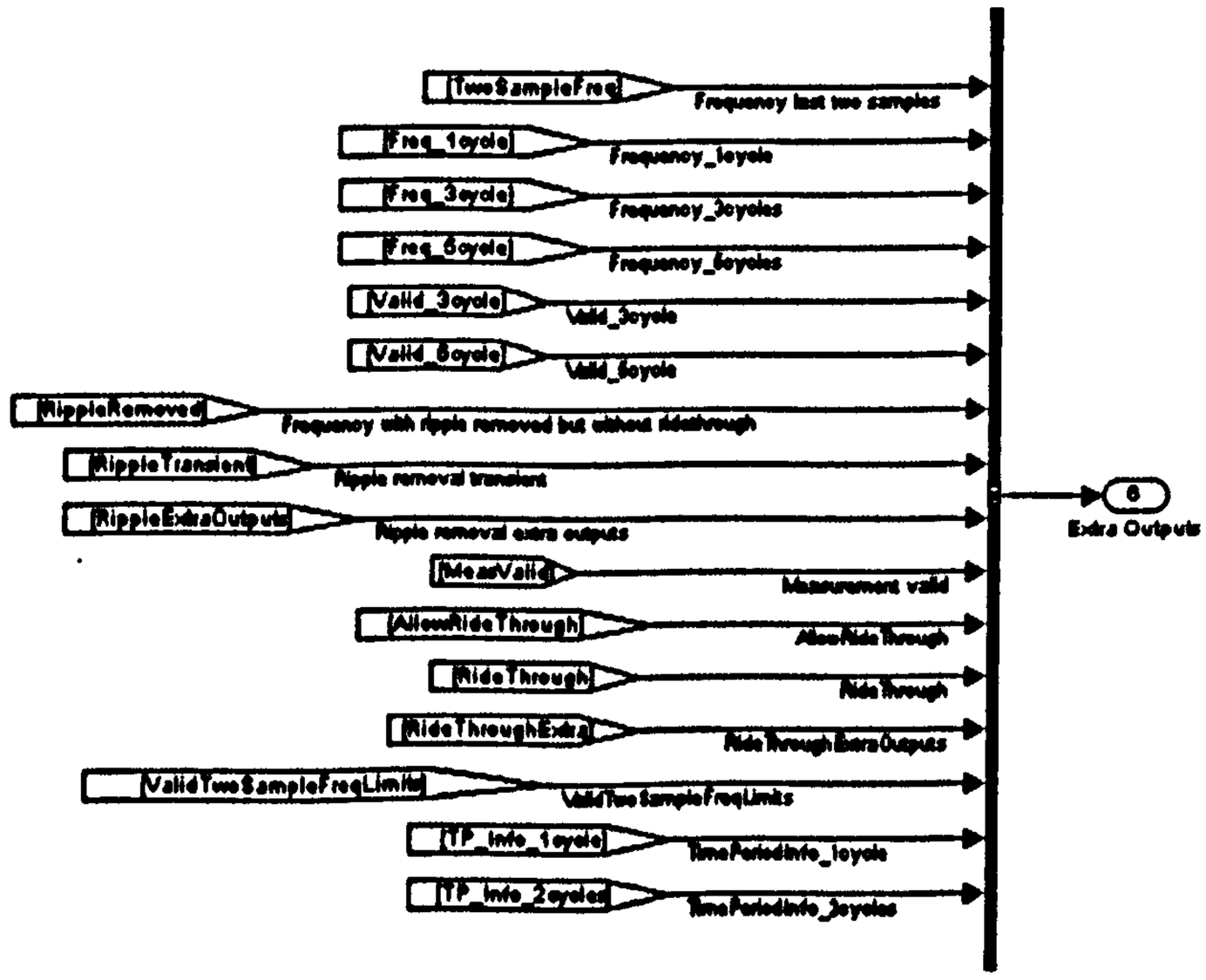


Figure 5-5 : Clarke transform frequency measurement algorithm - detail (1) [this and previous page]

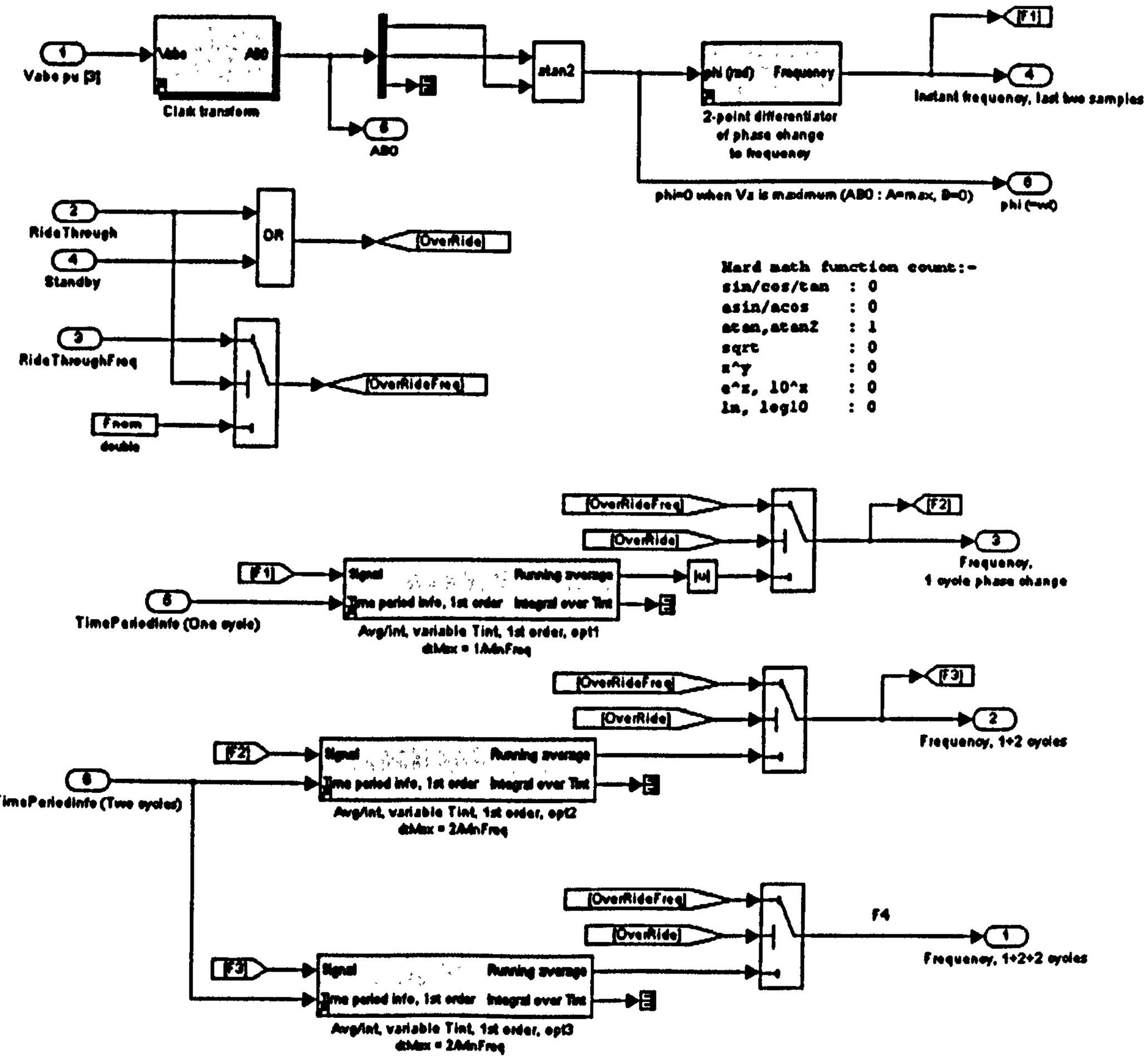


Figure 5-6 : Clarke transform frequency measurement algorithm - detail (2); core

The next stage (Figure 5-7) is to determine the validity of the measurement. It was considered to use a trajectory analysis of the [AB] vector, as derived by Ignatova (2005), for this. However, this analysis as presented takes at least a cycle to compute, which is too long. Instead, the analysis of validity is done by analysis of the instantaneous frequencies on a sample-by-sample basis, and checking these values against lower and upper limit lines which will not be exceeded when measurable 3-phase signals are applied which do not have 2-phase faults. The instantaneous frequencies are measured by the rate of change of the angle of the AB vector over a single sample frame. Either one of the following two problems will cause the limits to be exceeded:-

- The signal magnitudes are at a level approaching or lower than measurement noise.
- A two phase fault. This causes a collapse of the AB vector trajectory from a circle to a one-dimensional line which has a un-measurable rate of change of angle. Note that a single phase-ground fault results in a collapse of the AB vector trajectory from a circle to an ellipse of aspect ratio $1/3^{rd}$, with a maximum rate-of-change of angle per second of $3*2\pi*f$, where f is the frequency. This is shown in appendix B.1.4.

Therefore, sensible bounds on the two-sample frequency can be set at approximately:-

Lower limit: $-\text{atan}(y/0.05)/(2*\pi*T_s)$

Upper limit: $\text{atan}(y/0.05)/(2*\pi*T_s) + f_{nom}*1.2*3$

These limits allow for

- Declaration of invalidity when the [AB] vector magnitude falls below 0.05pu, with worst case measurement noise of y pu (set to typically 4 times the RMS measurement noise which includes instrumentation noise and ADC quantisation/noise errors)
- single-phase faults at up to $f_{nom}*1.2$, i.e. 60Hz for a 50Hz system.

Figure 5-7 shows this checking stage. Importantly, the resulting flag "ValidInstant" is then qualified by periods of 1.5 (nominal) cycle periods, 5 cycles, and 8 cycles. These qualified flags then describe the validity of the 1, 3 and 5-cycle averaged measurements. The additional qualification time allows for settling in the hardware low-pass filters, the internal averaging stages, and also transient settling within the power system itself. These flags are subsequently used (Figure 5-5) to select the appropriately averaged frequency result to use; the fastest during "startup", or the most averaged during steady-state operation.

An alternate novel method was investigated to detect validity and the presence of two-phase faults, in a manner similar to that of Ignatova (2005). This measured the area of the ellipse enclosed by the AB trajectory. This method proved to be more complex and less effective than the above method, and is not presented here.

The chosen result is then passed through the ripple removal filter, which was introduced in section 4.3.1. This can apply a maximum additional averaging period of 4 nominal cycle periods, but during fast frequency changes it will bypass itself to provide a faster result. Note that during large transient events, the ripple removal filter must remove itself to

meet the overall response specification of 5 cycles. This is because the previous averaging stages take 1+2+2=5 cycles already, and so an entirely new frequency input will take 5 cycles to filter through the 5 cycles of prior averaging and settle. However, during steady-state and slow frequency ramps, a total averaging of 1+2+2+4=9 cycles produces a result which represents the frequency at the midpoint of this averaging time period; i.e. 9/2=4.5 cycles prior to the output result. Thus, during steady state or slow frequency ramps, the ripple removal filter can switch in and the overall response time will be just within the specifications.

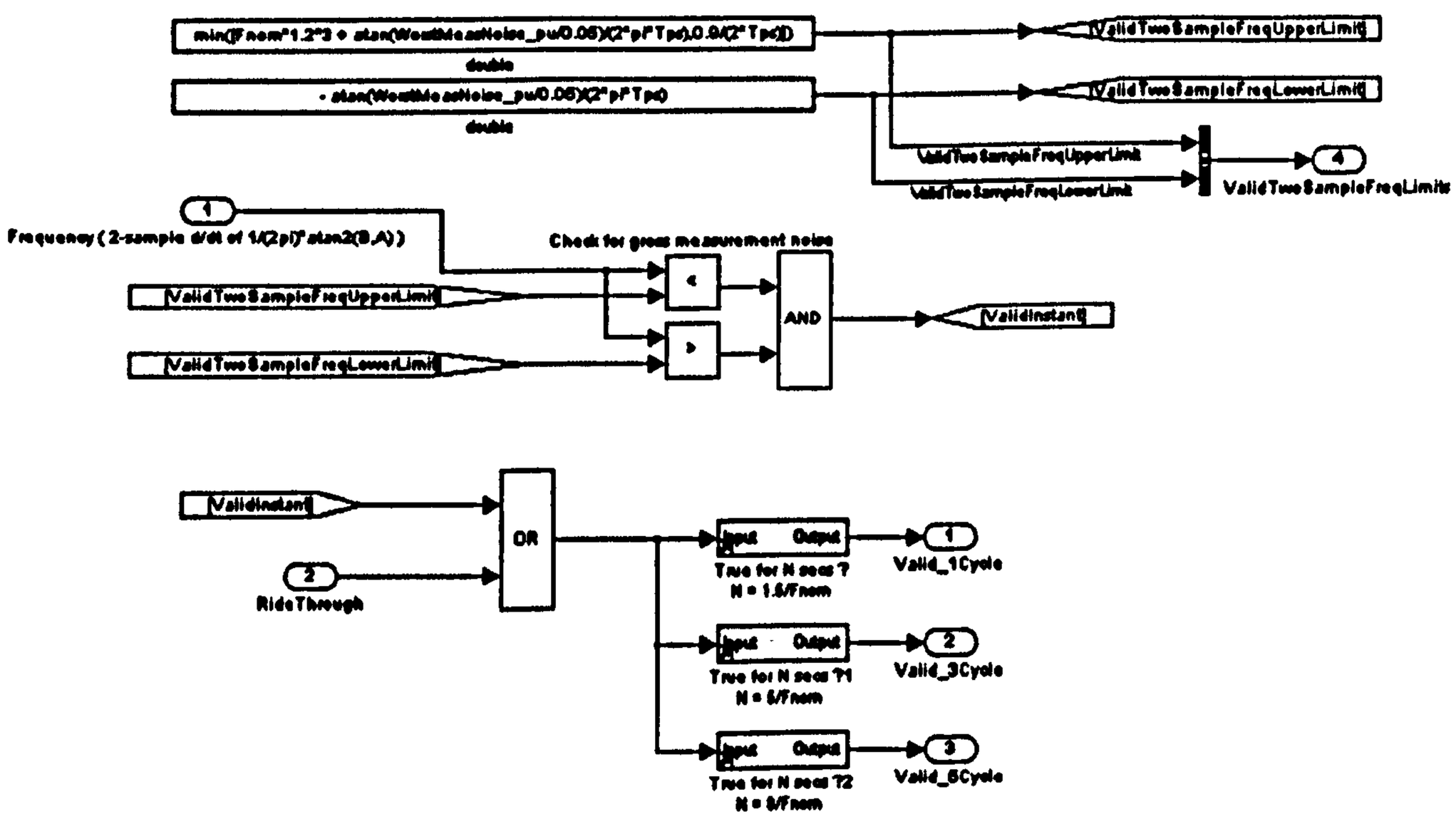


Figure 5-7 : Clarke transform frequency measurement algorithm - detail (3); validity tests

The next stage is to decide whether ride-through should be activated, due to a disturbance; i.e. sudden phase voltage dip (or surge), from a nominal operating condition. Such ride-through action is required as laid down in section 2.7.5. The algorithm for detecting such events is described later in section 5.4.2, since it requires knowledge of the phase magnitudes. The decision process within the Clarke transform frequency measurement is shown in Figure 5-8. If there is such a disturbance, AND the measurement was valid at the "lookback" time, then ride-through can be activated. This will trigger the overall algorithm (Figure 5-5) to hold its output at a constant value, which is taken from the measured frequency value at the "lookback" time. The "lookback" time is set at 1 cycle prior to the detection of the disturbance. This allows for the finite disturbance detection time, and so the frequency used for ride-through is taken from a time which is (hopefully) before any disturbance reached the actual frequency measurement. Ride-

through continues until:-

- a timer expires (set by *MaxRideThroughTime*)
- the phase magnitudes are usable, for a time long enough to make the measurement valid

As soon as either condition is violated, ride-through activity ceases.

Note that a subsequent ride-through action may not be initiated for a fixed period after a ride-through action begins.

Note also that here, and in other certain parts of the algorithms, single-sample delay states (z^{-1}) need to be inserted in some signal paths to avoid "algebraic loops", where these signals are passed back to earlier parts of the algorithm.

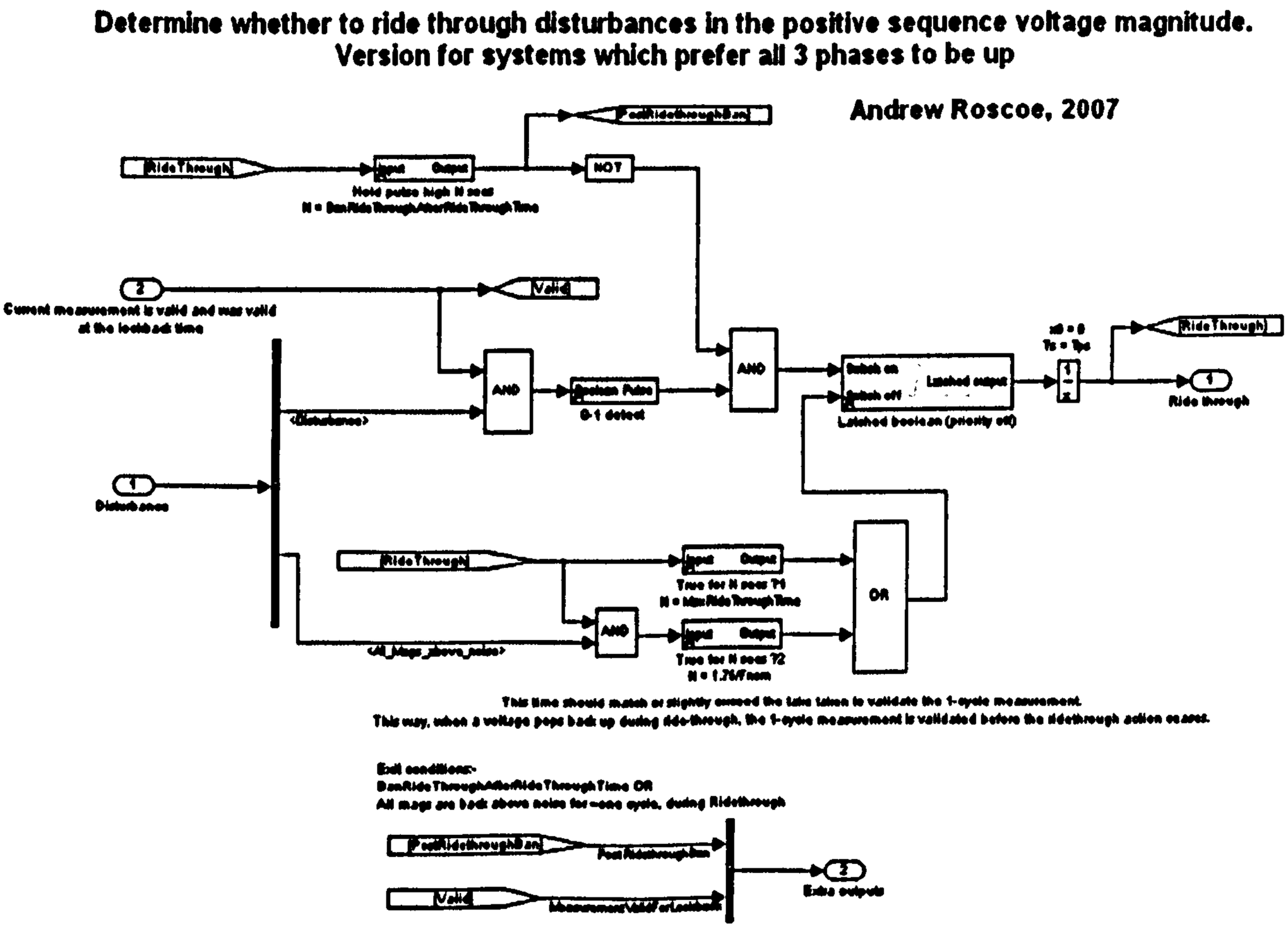


Figure 5-8 : Clarke transform frequency measurement algorithm - detail (4); ride-through decision

The final parts of the algorithm output the primary frequency result and the subsidiary outputs for debug/monitoring/logging (Figure 5-5). A subtle feature is that there are two outputs of frequency. The first is the best estimate, which should always be used to drive any subsequent amplitude/phase measurement blocks. In the case of no valid input signals (or 3-phase instrumentation failure), this output will essentially output noise. The alternate output is set to the nominal frequency if the measurement appears to be invalid.

This alternate output can be used to drive governors containing droop controls, since it will not cause any major primer-mover throttle ramps if there is some form of instrumentation failure. This last point is the reason why the nominal frequency was selected to be output under such conditions, rather than a default to 0Hz.

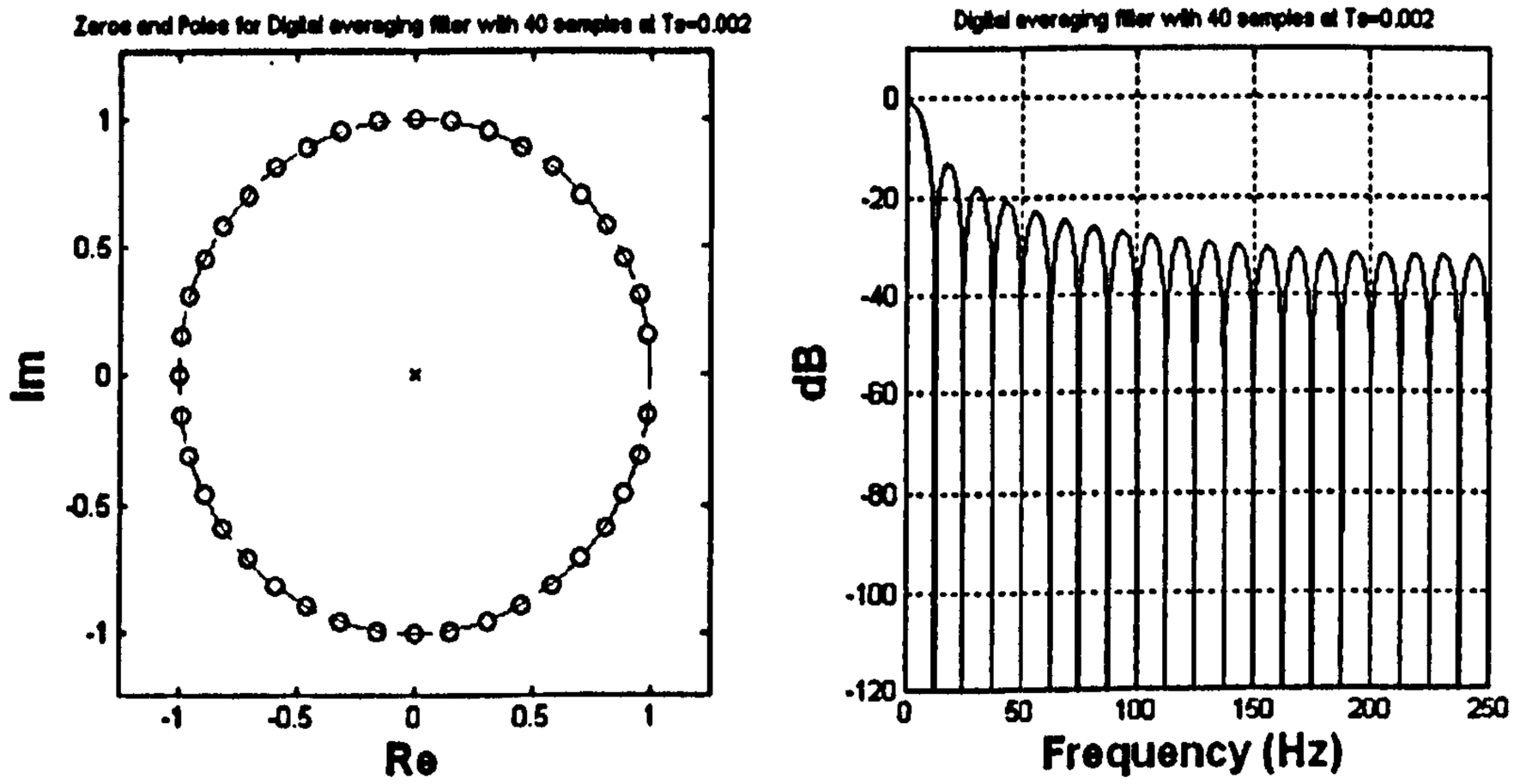


Figure 5-9 : Zeros, poles and Bode plot for a single averaging filter, averaging over 4 cycles

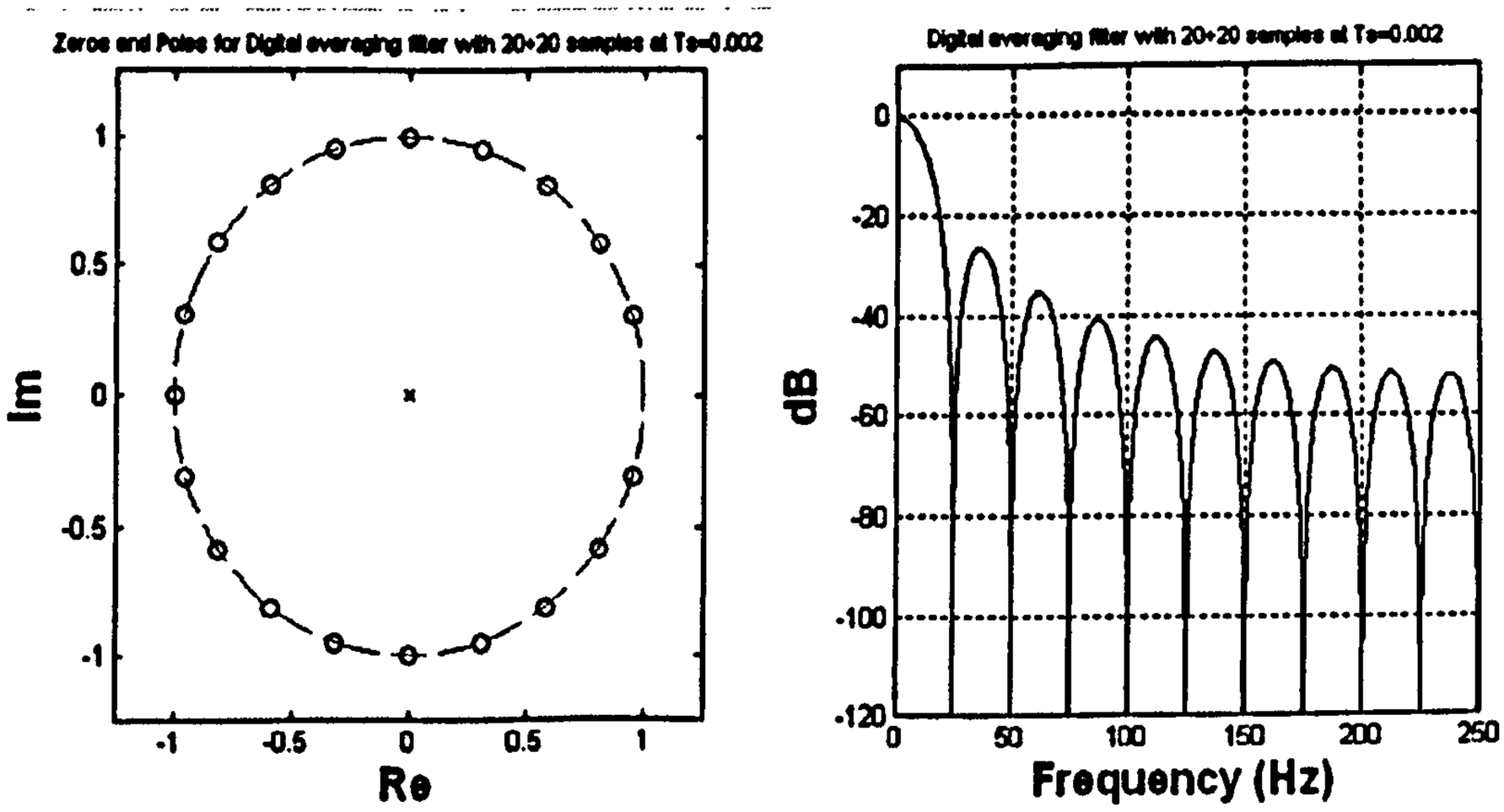


Figure 5-10 : Zeros, poles and Bode plot for a pair of cascaded averaging filters, averaging over 2+2 cycles

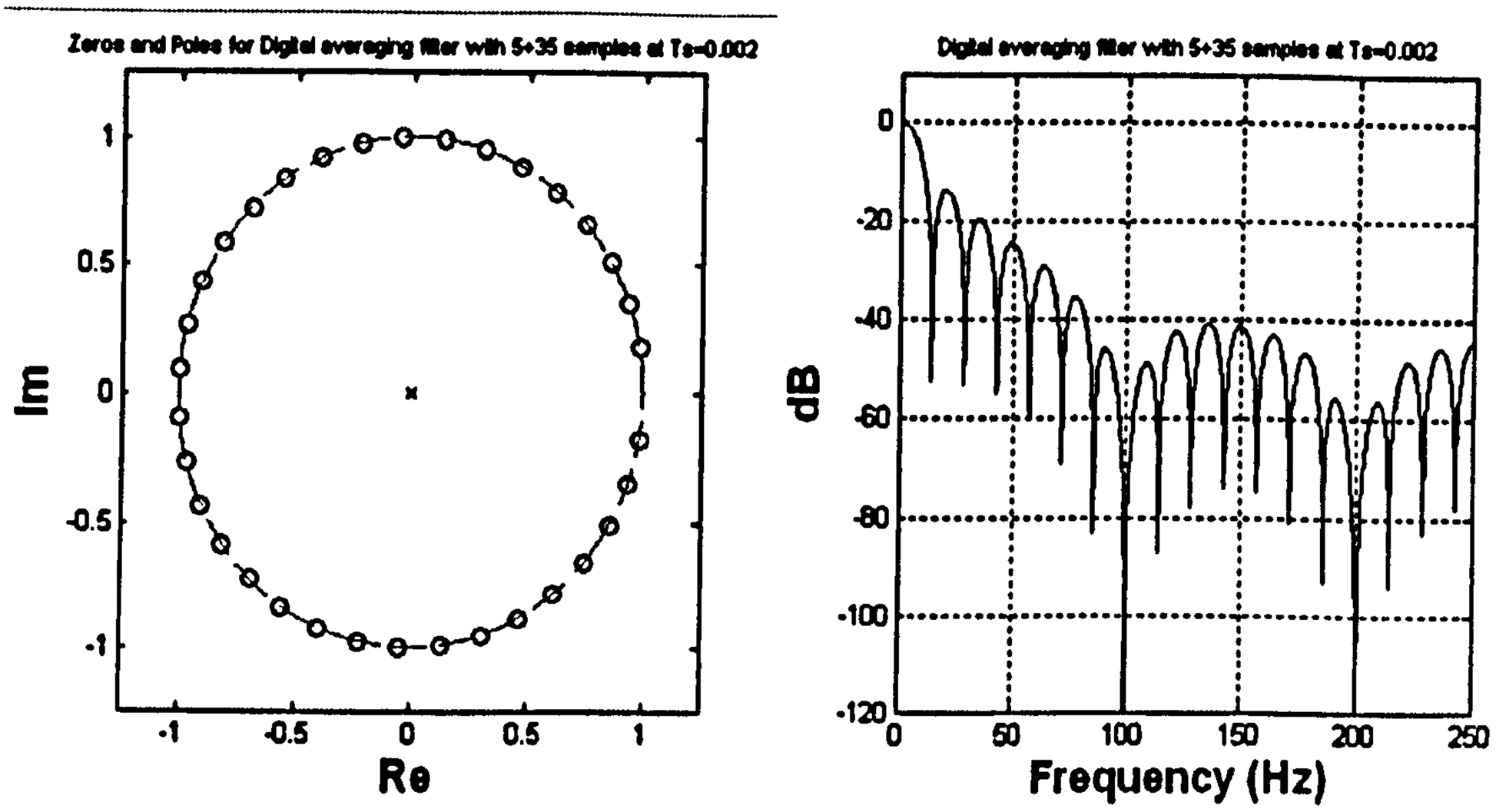


Figure 5-11 : Zeros, poles and Bode plot for a pair of cascaded averaging filters, averaging over ½ +3½ cycles

5.4.2 Method 2 : Clarke-FLL hybrid

The Clarke transform provides a fairly robust method of measuring frequency. However, it does not allow measurement during two-phase faults. To continue measuring frequency during such faults, a system of 3 PLLs with weighted averages to suitably prioritise the outputs of the active or faulted phase(s) could be used. However, as shown in section 5.3, PLLs do not exhibit a desired response where a frequency measurement is required. Instead, this thesis proposes that a FLL (frequency locked loop) is more appropriate. To be precise, a system of 3 single-phase FLLs with weighted average outputs is proposed. Such a set of FLLs aims to track the frequency of the 3 incoming signals, and not the phase. Thus, when signals with a step change in frequency are input, PLLs must exhibit significant overshoot at their frequency output, while the overshoot/ringing response of the FLLs, if correctly designed, is much smaller.

The problem with any PLL or FLL is the initial settling time when a new signal is input. To counter this problem, the FLL can be seeded with a value from another frequency measurement algorithm when appropriate. The seeding can be used to lock the FLL on to the new signal much faster than would otherwise occur. Given that a robust, fast-settling Clarke-transform based measurement has already been designed, the logical solution is to combine a FLL measurement with the Clarke measurement, use the Clarke measurement to seed the FLL when appropriate, and select the best output from either of the two algorithms for subsequent processing. As a bonus, the FLL measurement can be coded to provide not only the frequency measurement, but also the amplitude/phase measurements of the three phase voltages/currents. This reduces the computational burden since the amplitude/phases do not subsequently need to be measured.

The total algorithm can be called a "Clarke-FLL hybrid". The first half of the hybrid was presented in section 5.4.1, and can be used as a stand-alone frequency measurement system. The second part of the hybrid (the set of 3 FLLs) is presented in this section. The FLL set can be used without seeding, but is designed to operate using a seed value from the Clarke transform measurement. The FLL algorithm calls upon all the amplitude and phase measuring techniques of chapters 3 & 4. The combination of the two parts forms an extremely effective and robust algorithm. Its operation is described below and summarised on Figure 5-12.

This algorithm contains only FIR (finite impulse response) filters throughout. The exception is that the final measured frequency is fed back to the start of the algorithm, where it is used to help measure the amplitude and phase of each of the 3 input signals. Thus, if the input frequency has a step function disturbance, there is (without the action of seeding), the possibility of some FIR-type settling and ringing response within this algorithm. Note that there is no PI controller, and there is no attempt in this algorithm to track phase. The settling and ringing effects due to the FIR response can be mitigated by use of the seed frequency from the Clarke transform frequency measurement which was described in section 5.4.1. This seeding is extremely effective, because the main algorithm is a frequency locked loop and not a phase locked loop. Thus, seeding the algorithm can be done quickly and effectively. In a PLL this is not the case because seeding it with the correct frequency still requires the PLL to then hunt for the phase before it becomes locked.

The details of the Clarke's frequency measurement have already been described in section 5.4.1. The details of the FLL part of the Clarke-FLL hybrid are presented below in Figure 5-13, Figure 5-14, and Fig. A-1 to Fig. A-8 (see Appendix A). Many of the lower level blocks have been introduced in chapter 3 and require little further explanation. The algorithm is, however, necessarily large and comprehensive. Describing its operation to the lowest level of detail in words is not possible within a single chapter, and would merely repeat the information presented in the figures. The main points of note are summarised below, which, in conjunction with the figures, present the operation of the algorithm to an appropriate level of detail. The performance of this algorithm is demonstrated later in section 5.7.

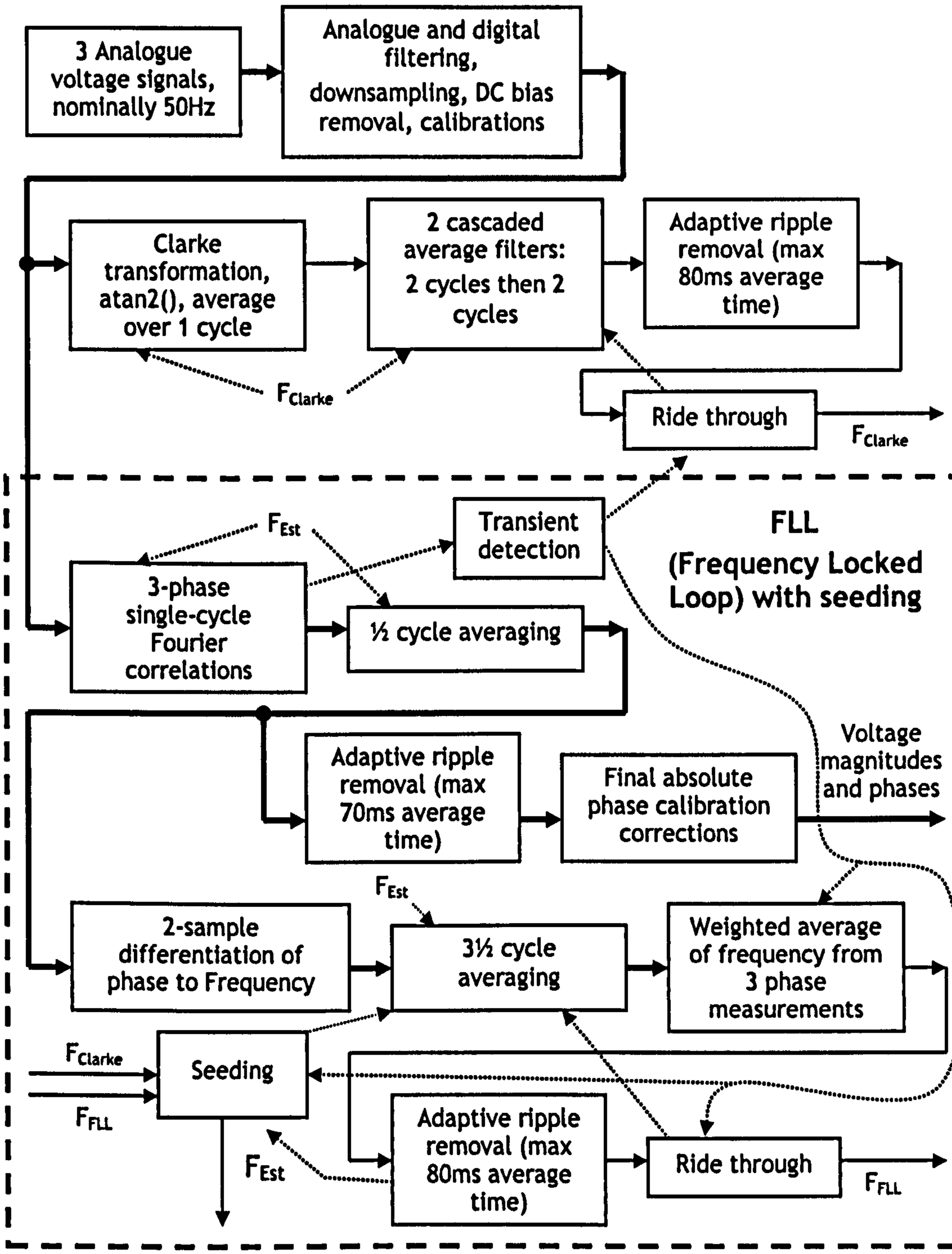


Figure 5-12 : Algorithm summary for Clarke-FLL hybrid

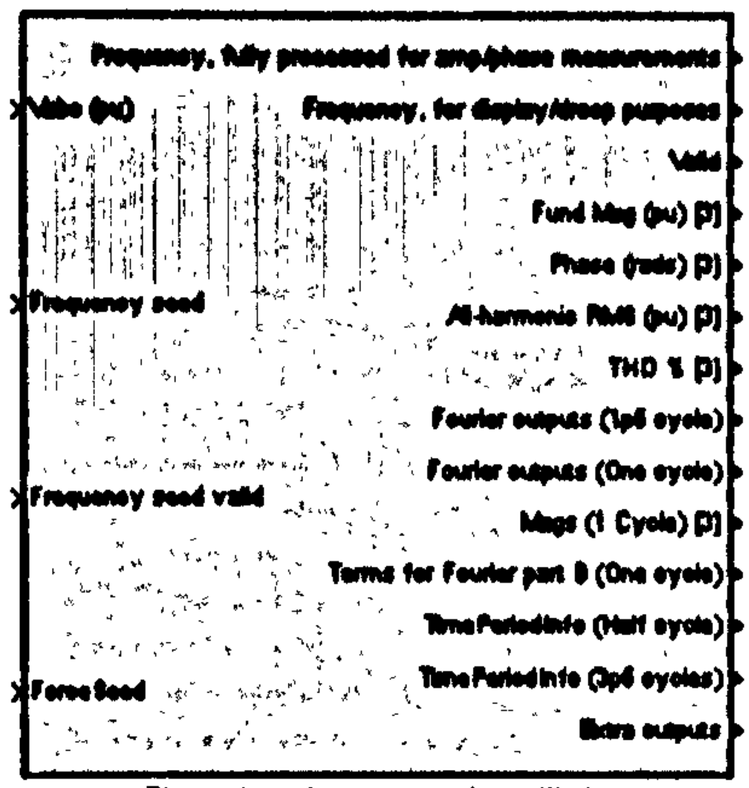


Figure 5-13 : FLL - overview

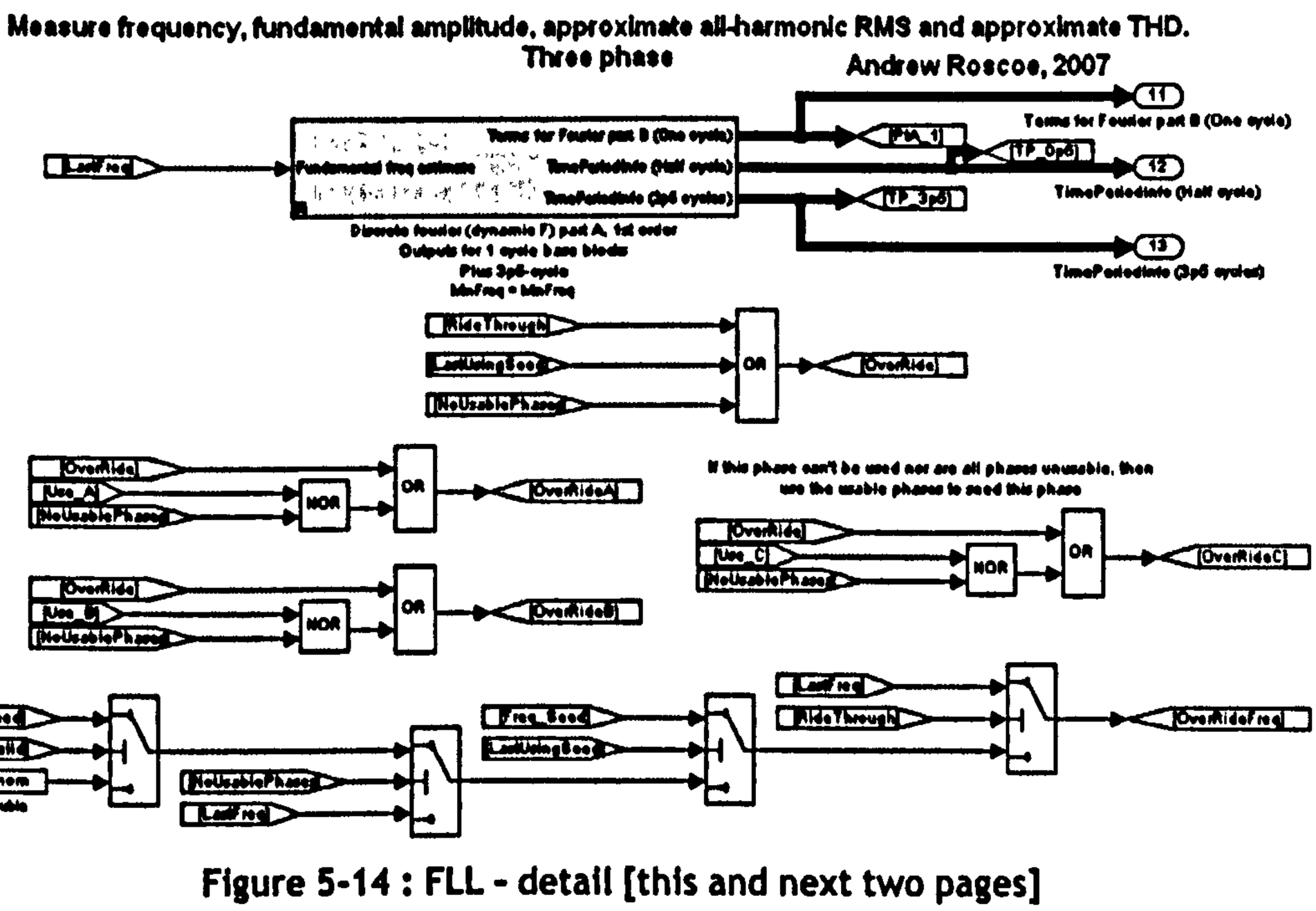


Figure 5-14 : FLL - detail [this and next two pages]

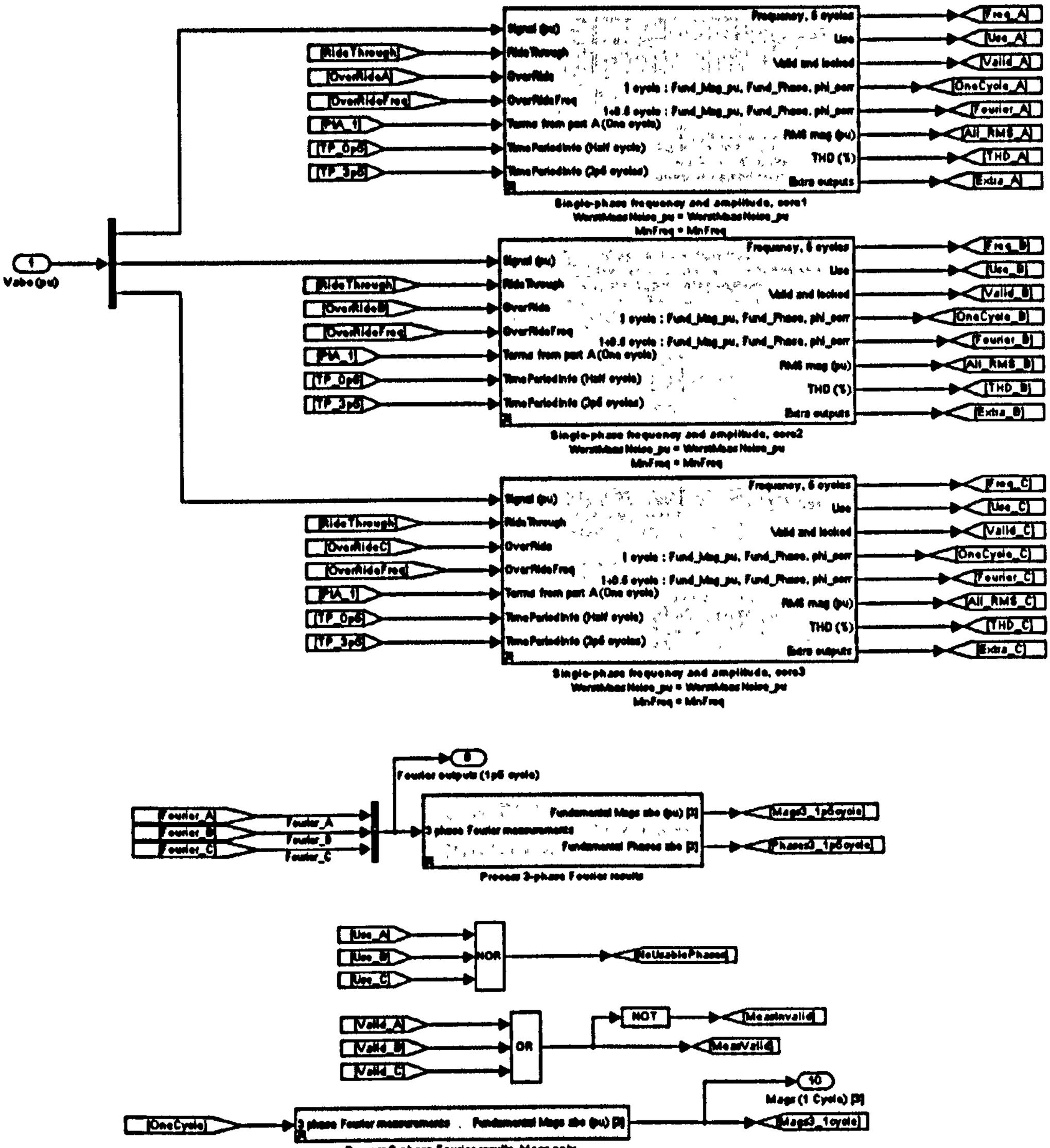


Figure 5-14 : FLL - detail [previous, this and next pages]

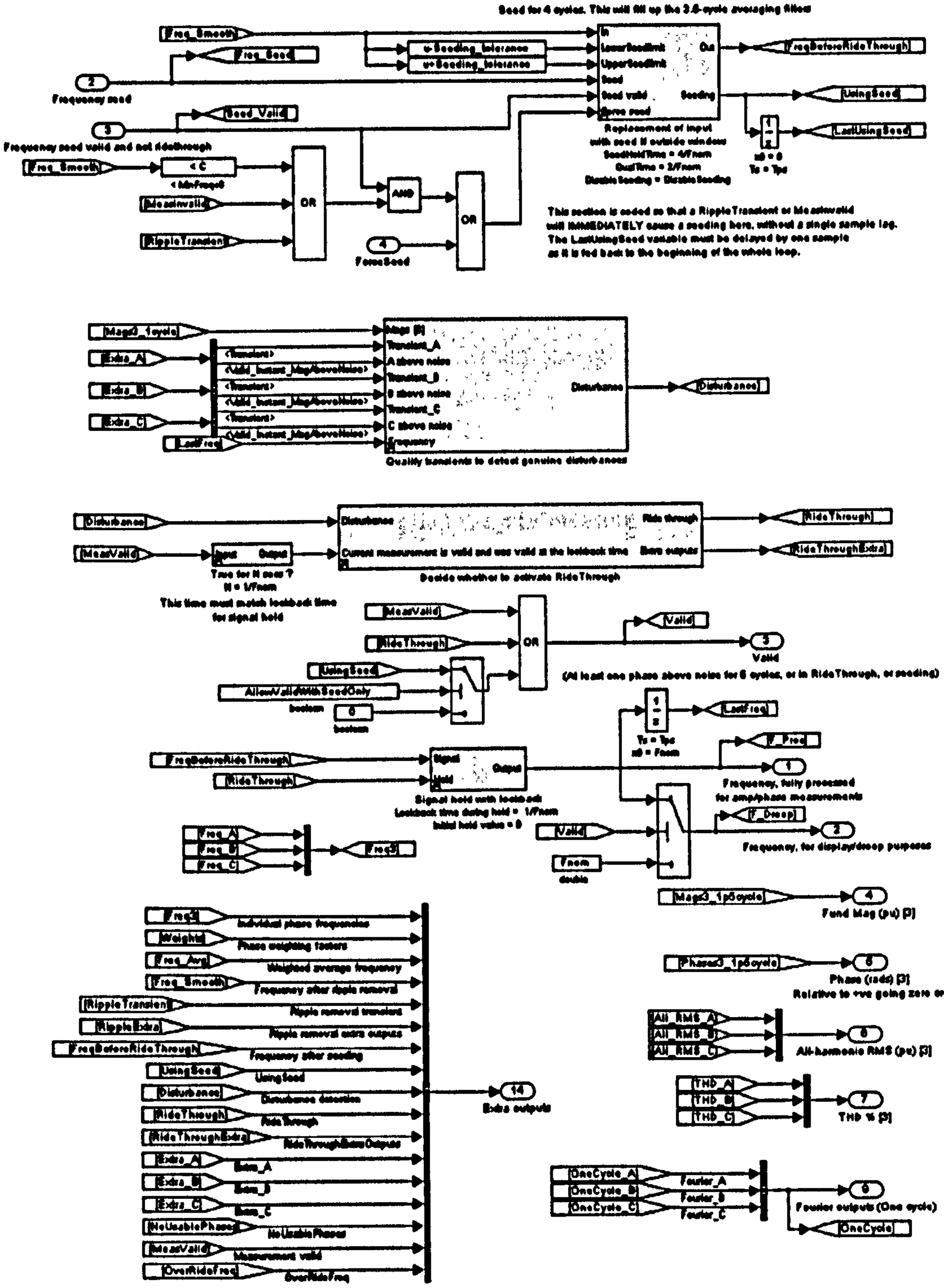


Figure 5-14 : FLL - detail [this and previous 2 pages]

The core of the algorithm is a set of 3 Fourier analysis blocks (Fig. A-1). These are "1(NC)+0.5" (see section 3.13) blocks using 1st order interpolation/integration for the 1-cycle and ½-cycle averaging blocks. This stage is just as described as in section 3.9. The outputs of the 3 blocks are the amplitudes and phases of the fundamental waveforms. These can be used directly for voltage measurement purposes, synchronisation checking

etc., and hence there is good optimisation of CPU use by making these blocks the core of the Clarke-FLL hybrid algorithm. All of the other frequency measurement methods presented required subsequent additional amplitude/phase measurement blocks to be executed in order to determine the amplitude (and disturbance) information for each phase.

The frequency of each phase is measured within the core blocks by differentiating the absolute phase (relative to a positive-going zero crossing) of the waveform on each phase, at the output of the "1(NC)+0.5" block. Then, as in the Clarke transform frequency measurement, an additional 3½ cycles averaging can always be applied to realise a measurement with ~5 cycle latency. In the case of the Clarke transform measurement, a 1+2+2 cycle averaging scheme was used which gives the best noise rejection (see section 5.4.1). Within the Clarke-FLL hybrid, a 1½-cycle measurement has already been carried out within the "1(NC)+0.5" block. A compromise between noise rejection and CPU loading is then to apply an additional 3½-cycle averaging filter to make up the averaging to 5 cycles, using a 1+½+3½ cycle measurement system. The decrease in noise rejection relative to the 1+2+2 cycle measurement system is relatively small, as shown by Figure 5-10 & Figure 5-11.

During certain circumstances, the 3½-cycle filters are filled (pre-loaded) with values other than the actual measured values. These circumstances are:-

- During fault ride-through
- When the seed frequency is being used
- When the phase voltage appears to be very low (in the noise), and at least one of the other phases appear to have a usable signal voltage present

This pre-load action helps to speed up settling of the averaging filters when a genuine signal (re-)appears.

When a voltage level above the level of noise appears on a phase, a sequence of flags gives an indication of the validity of the frequency measurement from that phase:-

- When the voltage has been significant for 2 cycles (40ms), the frequency measurement begins to be used within the algorithm as a whole. This is described by the flag "Use" and "Use_A", "Use_B" etc.. If this flag is low, the measurement of frequency from the phase is discarded, unless all phases have low voltages in which case all phase measurements are used in equal weights.
- When the voltage has been significant for 5 cycles (100ms), the frequency

measurement can be valid, but it must also be locked to be declared fully valid

- Lock, for each phase, is determined by analysis of the rate of change of phase of the input signal, relative to the correlating Fourier waveform at the measured frequency. This quantity will drop towards zero when the algorithm is at steady state with a steady frequency input. Lock is determined by analysing this signal through a low-pass filter and an "abs" function (see Fig. A-1). This provides the most reliable way of determining lock and unlock without using more CPU-intensive functions such as rolling buffers, and without incurring spurious unlock signals during transient voltage spikes.
- When any of the three phase measurements is valid (and locked), then the measurement as a whole is valid.

The logic in the above bullet points is important to get right, because being too conservative results in an algorithm which can never get started (it gets stuck in an "invalid" state), and being too lax allows incorrect answers to appear at the output with a "valid" status.

The answers from the 3 core frequency measurement blocks are combined together within a weighted averaging algorithm, shown in Fig. A-3 and Fig. A-4. This uses weightings determined from the phase magnitudes, applying higher weightings for signals with nominal 1pu magnitudes, and lower weightings for signals which are higher or lower than 1pu. In addition, logic of Figure 5-14 can set the weightings to 10^{-9} if the frequency measurements from each core block are invalid or unlocked, or if a single or two-phase disturbance is in progress (Fig. A-7). This allows clean, sustained ride-through of single and two-phase disturbances. If none of the 3 core blocks is valid or locked, then the logic does allow measurements to be used if the phase voltage magnitude has been significant for 2 cycles and the "Use_X" flag is set for that phase. This allows the algorithm to get going from a "cold start", even in then absence of a usable frequency seed.

A ripple filter is applied, as for the Clarke transform frequency measurement. Again, the maximum latency allowed is 4 cycles, for the reasons discussed in section 5.4.1. The ripple filter switches itself off during rapid frequency changes to keep the latency within specification.

The seeding algorithm is spread between Figure 5-14 and Fig. A-5. Seeding will occur if the Clarke transform frequency measurement is valid AND any of the following conditions are true:-

- The local frequency measurement is more than 1Hz different to the Clarke transform measurement
- The local frequency measurement is close to or less than the variable *MinFreq*, which sets the lowest frequency which can accurately be measured inside the Fourier measurement blocks due to the length of the averaging buffers. These must be predefined at compile-time as buffers with approximately X/(*MinFreq**Ts) samples stored in memory, where X is the number of cycles of averaging for each buffer. When the actual frequency drops below this value, the core Fourier blocks tend to unlock due to the ripple which appears. Note, this problem does not occur with the Clarke transform frequency measurement as the phase trajectory is a straight line, so an inaccurate averaging period does not produce such a rippling effect unless unbalance is large.
- If the ripple filter reports a transient event, seeding is a sensible option because it is likely that the Clarke transform frequency measurement will produce a better result due to its very fast settling.

To trigger ride-through, disturbances in the fundamental magnitudes of the three phase voltages must be detected quickly, and compared against thresholds of tolerance. This is done using the algorithm of Fig. A-7. The inputs to this algorithm are outputs from the 1-cycle latency Fourier correlations, namely the fundamental voltage magnitudes, and "transient" flags for each phase which are determined within the Fourier correlation block as shown in Figure 3-42. A valid disturbance is flagged when the voltages and frequency start at values close to nominal, and then at least one of the phase magnitudes deviates rapidly. This rapid deviation can occur due to a fault, disconnection, or surge transient. In the cycles that follow, the disturbed phase voltage may be immeasurably low, unsinusoidal, or contain a voltage spike. Any of these, or even a large step change in phase voltage, can cause inaccuracy in the frequency measurement taken from the disturbed phase. The disturbance flags generated by this algorithm will usually be within ≤½ a cycle of the event occurring, due to the settling characteristics of the 1-cycle Fourier correlations.

The next step is to decide whether a disturbance flag should trigger a ride-through action. This is done using the algorithm of Fig. A-6, which is very similar to the equivalent decision process for the Clarke transform frequency measurement algorithm shown in Figure 5-8. The difference in this case is that because the measurement can function during two-phase faults, the algorithm will exit from the ride-through condition if any of the three phase voltages re-appear to usable levels. The Clarke transform measurement equivalent

requires all three phase voltages to re-appear to usable levels to cause this same action.

The overall validity of the measurement is determined by any one of the following conditions being true:-

- The local measurement is valid
- Ride-through is active
- The seed is being used (this can only happen when the seed itself is valid)

The final step is to implement the ride-through, if it is active. This is done as shown in Fig. A-6 and Fig. A-8. When a ride-through action begins, the frequency is sampled and then held at a steady value for the duration of the ride-through state. The trick is to take the sample from a time prior to the beginning of the ride-through state. Here, a 1-cycle look-back is used. This time must be set to slightly longer than the disturbance detection might take, which might be ½ a cycle inside the algorithm plus the latency of the analogue & sampling hardware. This avoids taking the sampled value from a time after the disturbance began, when the frequency measurements may be already distorted.

This leads to the final output frequency. As for the Clarke transform measurement, there are two output ports. The first is unmodified, and should be used to feed any other algorithms which would prefer a "best estimate" answer even if the measurement is not valid. The second pin is for display/logging/droop actions, and reverts to an output at f_{nom} (nominal frequency) when the measurement is invalid. This avoids "random" frequencies appearing on displays etc. when no voltages are present and the instrumentation noise is the dominant signal. It also minimises the risk of droop controllers railing during a momentary or complete instrumentation failure.

5.4.3 Method 3 : Fixed-reference Fourier frequency measurement with seeding

This technique is in many ways similar to the FLL of section 5.4.2. In this case, however, the IIR settling response is eliminated by using a fixed 50Hz (f_{nom}) reference for the Fourier correlation waveforms. This means that the correlations do not make good representations of the actual magnitude and phase of the input signals, as for any frequencies off-nominal the ripple at the output of the 1-cycle correlation is large. However, when averaged using cascaded exact-time averaging filters this ripple can be mostly removed, by the processes described in section 4.1. Thus, a reasonable estimate of frequency can be made, on a phase-by-phase basis, which allows two-phase faults to be tolerated. As with the Clarke-FLL hybrid, the possibility exists of errors and locking to sub-harmonics/harmonics

if the input signal is sufficiently far below or above nominal. For this reason, and to benefit from the faster settling available from the Clarke transform measurement, this algorithm can elect to use the Clarke transform measurement as a seed when it makes sense. A drawback of this algorithm is that the amplitudes/phases must be subsequently re-measured using separate measurements, to avoid the small DC bias error at 0Hz which appears as described in section 4.1.

The coding details are presented in appendix E.1.

5.4.4 Method 4 : Three customised single-phase PLLs with weighted averaging and seeding

In order to compare the performance of a phase-locked-loop (PLL) with the above algorithms, a PLL-based solution has been coded, using the same concepts of fault ride-through and seeding. The code is shown in appendix E.2. It was chosen to implement a set of 3 single-phase PLLs rather than a single 3-phase dq-based PLL, because this solution allows operation during two-phase faults whereas the 3-phase PLL is vulnerable in this scenario. There is similarity to the previous two methods described, but it should be noted that some of the seeding details are different. This is because the single-phase PLLs contain an internal seeding algorithm. This is carefully coded because the seeding can only initialise the frequency part of the PLL; i.e. the PI controller. Once this is done, the seeding must be released and after this time the PLL must still hunt to find the correct phase lock. Thus, the seeding is more difficult to arrange, and less effective, than for the previous two algorithms.

5.4.5 Method 5 : Zero crossings

For comparison, the ride-through techniques and weighted averaging algorithms were also applied to a zero-crossings based technique. This would not be expected to perform so well, as only the pairs of samples nearest the zero crossings are used for the measurement, and thus the instrumentation noise is not so well attenuated. Also, subsequent exact-time averaging blocks do not attenuate ripple due to harmonics so well, as the resulting ripple frequency is not at exact multiples of the fundamental frequency. The algorithm requires no seeding as there is no FIR-type feedback loop. The primary measurements are made using the previous 5 cycles (10 zero crossings). The algorithm is robust against single and two-phase faults, as presents quite a low computational burden (see section 5.6). The code is presented in appendix E.3.

5.4.6 Method 6 : Three Jovcic's PLLs with weighted averaging

For comparison, a system of 3 PLLs to the Jovcic (2003) design is also implemented. The outputs of these are combined using a weighted averaging (as described in section 5.4.2) to optimise the performance during single and two-phase faults. The reader should refer to the original reference for the detailed PLL implementation details. The only differences between the implementation used in this thesis and the original Jovic code is that the 2nd harmonic cancellation algorithm has been simplified (see section 3.8) and a few checks have been added to avoid divisions by zero and overflow errors. There is no seeding and no detailed ride-through code added to this algorithm. It should be noted that Jovcic's PLL uses a low-pass filter with a cut-off frequency of only 1.3 Hz (ω=1/0.12=8.33 rad/s), which has a "5RC" settling time of about 0.6 seconds. This smoothes the output dramatically and causes substantial measurement latency.

5.4.7 Method 7 : SimPowerSystems 3-phase PLL

The SimPowerSystems 3-phase PLL is used without modification, for comparison with the other solutions. It should be noted that this (and the SimPowerSystems single-phase PLL), use a rate limiter set to ±12Hz/s combined with a 2nd-order low-pass filter with a cut-off frequency set to 25Hz, at the frequency output port. These two filters cause a large measurement latency during initial locking to a new signal at a new frequency. The algorithms are small compared to the proposed Clarke-FLL hybrid, but since the measurement of 3 phase amplitudes requires code additional to the SimPowerSystems PLL, the execution speed advantage over the Clarke-FLL hybrid is not great (see section 5.6).

5.4.8 Method 8 : Three SimPowerSystems single-phase PLLs with weighted averaging

Three SimPowerSystems 1-phase PLLs are used without modification. The outputs from the three PLLs are combined with a weighted average system to optimise the performance during single and two-phase faults. Again, the algorithms are small compared to the proposed Clarke-FLL hybrid, but since the measurement of 3 phase amplitudes requires code additional to the SimPowerSystems PLLs, the execution speed advantage over the Clarke-FLL hybrid is not great (see section 5.6).

5.5 Summary of considered techniques for frequency measurement

Table 5-1 shows a brief summary which compares the properties of the 8 trialled algorithms.

Algorithm	Three-phase measurement (3P) or 3x single-phase measurements using weighted average (3x1P W)	Robust during two-phase faults	Robust for wide frequency range? With (and without) seeding	Includes Ride-through coding
Clarke transform	3P	X	✓	✓
Clarke-FLL hybrid	3x1P W	✓	✓ (X)	✓
Fixed-ref Fourier with seeding	3x1P W	✓	✓ (X)	✓
3x 1-phase PLLs with seeding	3x1P W	✓	✓ (X)	✓
Zero crossings	3x1P W	✓	✓	✓
3x Jovcic's PLLs with weighting	3x1P W	✓	X	X
3x SPS PLLs with weighting	3x1P W	✓	X	X
SPS 3-phase PLL	3P	X	X	X

Table 5-1 : Summary of considered frequency measurement techniques

5.6 Benchmarking analysis

The 8 candidate methods for frequency measurement vary both in algorithmic complexity and also in the requirement for "hard" maths functions. The "hard" maths functions are those such as trigonometric functions and square root operators, as defined in section 3.13.1, Table 3-3. Such functions require significantly more time to execute than additions/multiplications or logical decisions (see Appendix G). An important consideration in the design of the Clarke-FLL hybrid measurement is that its overall computational burden and execution time should be competitive with alternative solutions. This is shown below. An important aspect of this analysis is that the Clarke-FLL hybrid algorithm provides measurements of both frequency and the amplitudes/phases of the 3 voltages/currents, in a carefully integrated manner to minimise CPU loading. The other 7 candidate frequency measurement algorithms require additional code to provide (in particular) accurate amplitude measurements. When the effect of this extra required code is accounted for, the total CPU burden of the Clarke-FLL hybrid is found to be competitive with that of the other algorithms.

The number of "hard" maths functions (see section 3.13.1 Table 3-3 for the definition of

"hard" functions) and delay buffer blocks (for the exact-time averaging) can be counted up for each algorithm combination, to provide a simple measure of CPU loading for each algorithm. These counts provide a reasonable but not comprehensive picture of CPU loading, since the many other more trivial but numerous calculations and operations will also contribute to the CPU load. Figure 5-15 below shows these counts for the 8 candidate algorithms, when used to provide a measurement of frequency and the fundamental amplitude/phase components of voltage, for a 3-phase voltage waveform set. The code to measure all-harmonic RMS and THD is not included in this analysis, since RMS and THD have been shown to be inaccurate measures at the low sample rates addressed in this thesis.

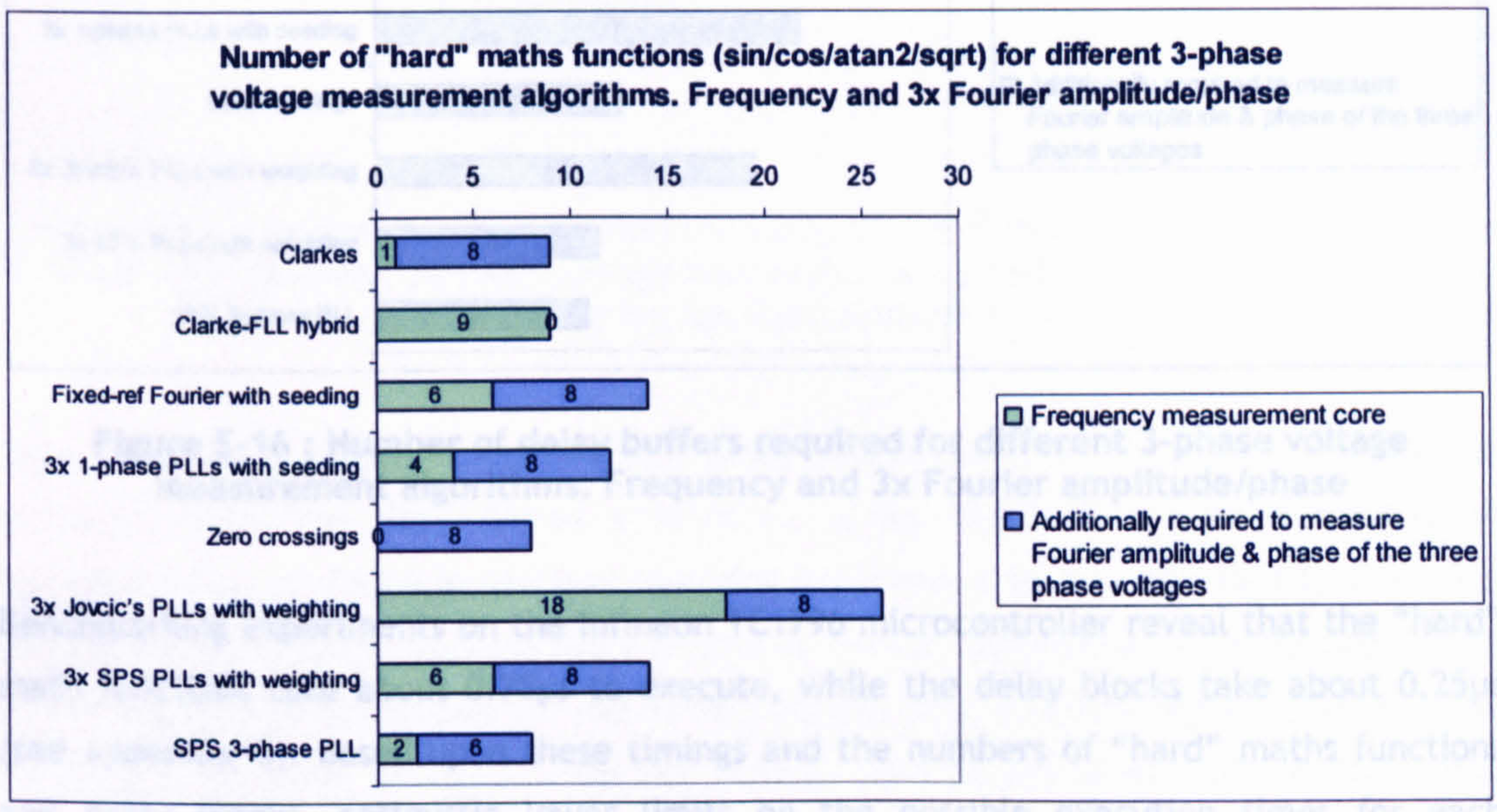


Figure 5-15 : Number of "hard" maths functions (sin/cos/atan2/sqrt) for different 3-phase voltage measurement algorithms. Frequency and 3x Fourier amplitude/phase

The Clarke-FLL hybrid compares well with the other candidate algorithms in Figure 5-15. The Zero crossings and SimPowerSystems 3-phase PLL algorithms require fewer "hard" functions.

The Clarke-FLL hybrid contains a lot of delay buffers (69), compared to most of the other candidate algorithms (see Figure 5-16)[1]. In particular, the zero crossings and

---

[1]3 delay buffer blocks are required for each signal averaging stage. Thus 36 buffer blocks are required to implement the "1(NC)+0.5" process on a set of 3 voltages. (3 blocks per average, 2 averages per phase for path A/B or Mag/Phase, 2 stages, 3 phases; 3*2*2*3=36).

SimPowerSystems PLL algorithms require fewer delay blocks, since their signal processing mechanisms are simpler.

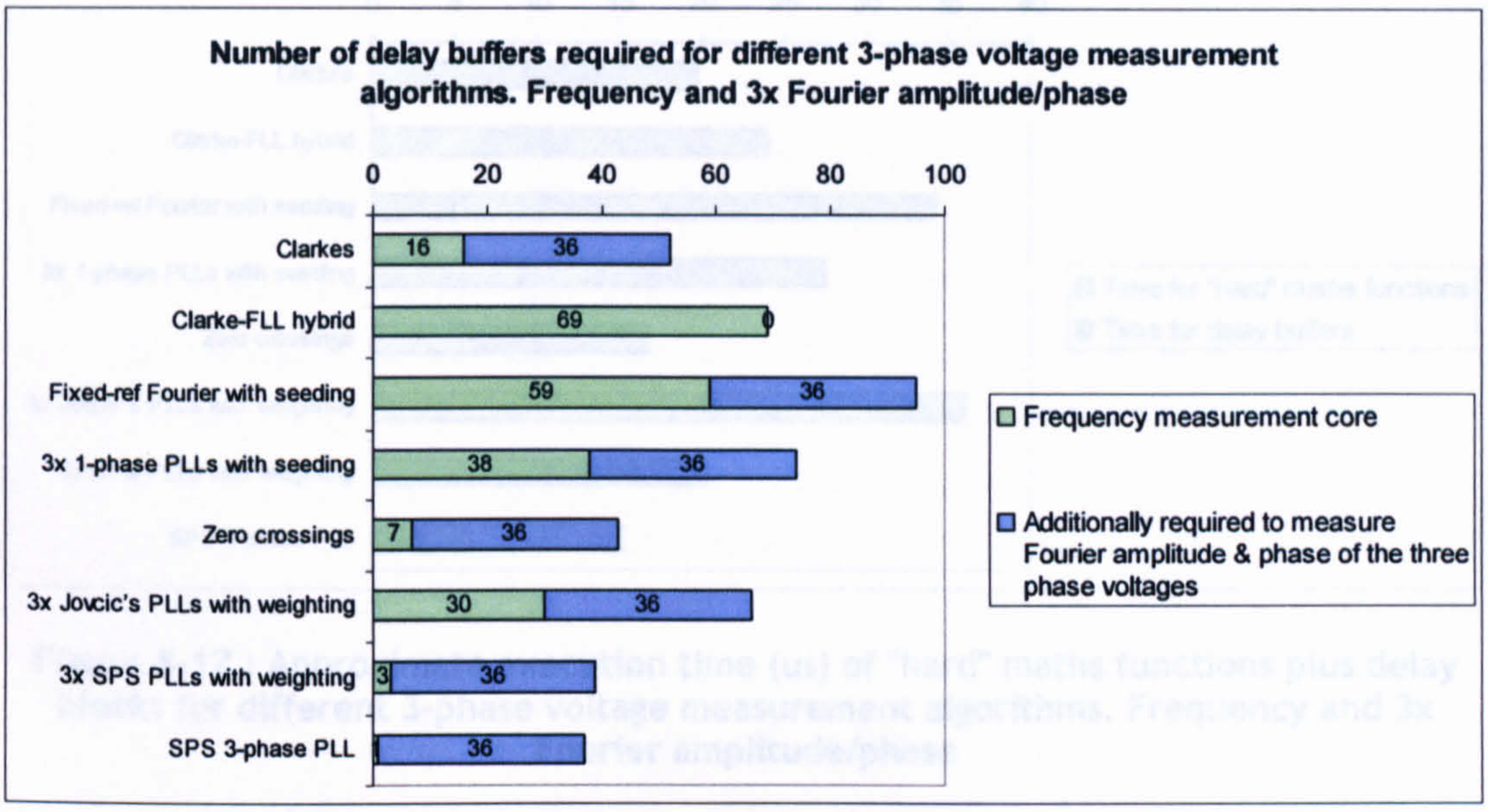


Figure 5-16 : Number of delay buffers required for different 3-phase voltage measurement algorithms. Frequency and 3x Fourier amplitude/phase

Benchmarking experiments on the Infineon TC1796 microcontroller reveal that the "hard" math functions take about 0.75µs to execute, while the delay blocks take about 0.25µs (see Appendix G). Based upon these timings and the numbers of "hard" maths functions and delay blocks, optimistic lower limits on the possible execution times for each algorithm can be simply estimated. These are shown in Figure 5-17. The Zero crossings and SimPowerSystems PLLs will have relatively low execution times due to their relative simplicity and the lack of any seeding functionality. Other algorithms will take longer. The Clarke-FLL hybrid compares reasonably (only 57% higher than the quickest solution, the SPS 3-phase PLL), and the prediction from this analysis is that the execution time should be somewhere in the excess of 24µs. A real benchmarking of the final algorithm is carried out in Appendix G, and the actual execution time for this algorithm is measured at 68µs.

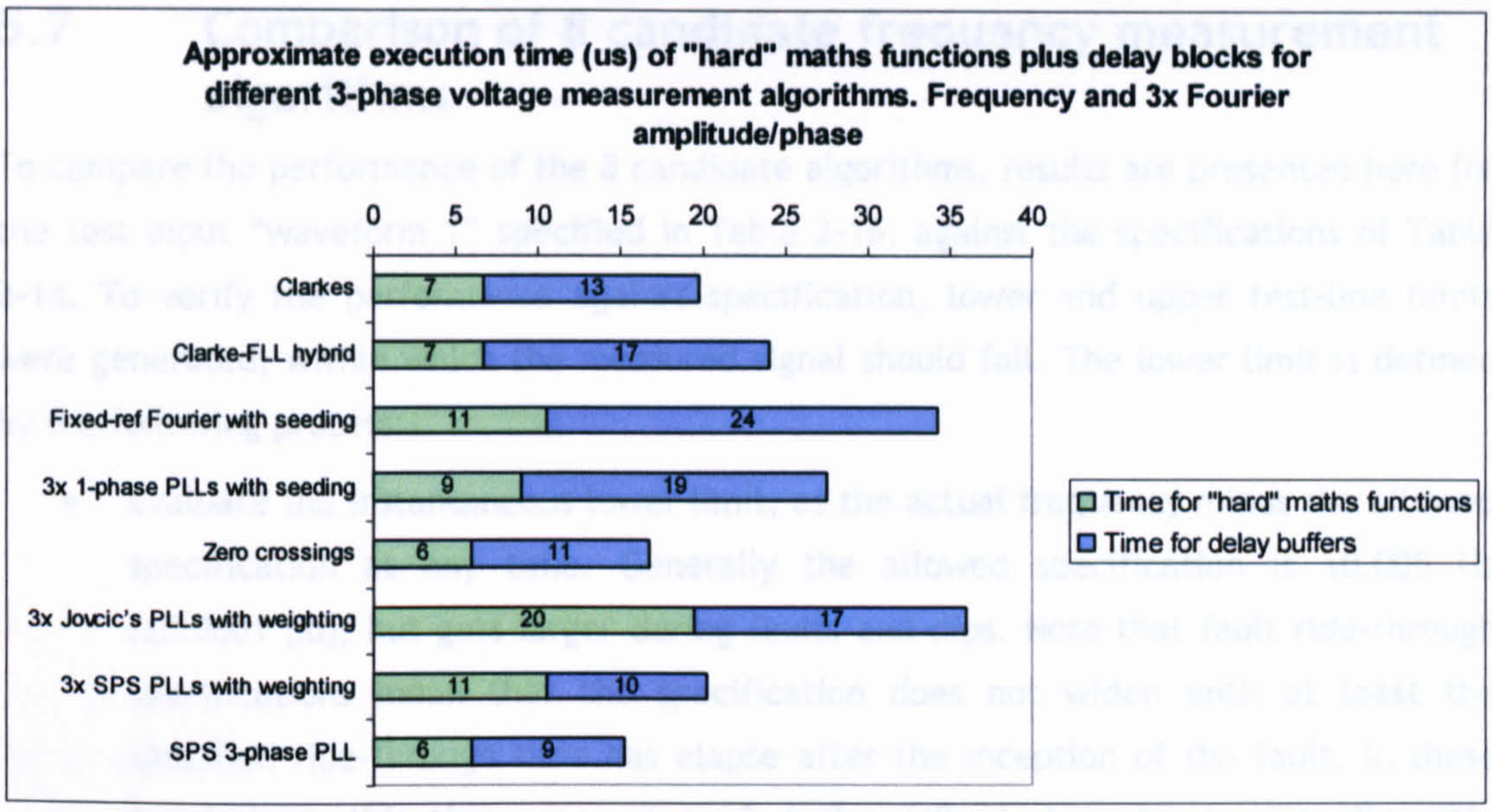


Figure 5-17 : Approximate execution time (us) of "hard" maths functions plus delay blocks for different 3-phase voltage measurement algorithms. Frequency and 3x Fourier amplitude/phase

Thus, the "hard" maths functions and delay blocks account for 35% of the overall execution time. The delay blocks account for about 17/68*100 = 25% of the overall execution time. As found in the benchmarking experiments in Appendix G, the execution time of the delay blocks appears to be limited by memory addressing and access time. This will be highly dependent upon the hardware platform used, and may be significantly faster (or slower) for other processors. On the TC1796, with an execution time of 68µs, at a frame rate of 10 frames per cycle (500 samples per second, a 2000µs frame rate), this leaves about 1932µs, 97% of the frame available for other tasks. These include analysis of 3-phase currents, positive/negative sequence analysis, analysis of further power system nodes, relaying actions, control algorithms, data logging, communication, etc. Some execution times for larger frequency/voltage/current and sequence/power-flow analysis algorithms have been measured and are shown in section Appendix G. In particular, the entire algorithm required to perform a full nodal measurement takes 156µs, less than 8% of the 2000µs frame time. This algorithm measures frequency, voltages and currents, with full sequence analysis and power flow calculation. All the amplitude and phase calibration functions are accounted for, including de-skewing of multiplexed ADC channels.

5.7 Comparison of 8 candidate frequency measurement algorithms

To compare the performance of the 8 candidate algorithms, results are presented here for the test input "waveform 1" specified in Table 2-16, against the specifications of Table 2-14. To verify the performance against specification, lower and upper test-line limits were generated, within which the measured signal should fall. The lower limit is defined by the following process:-

- Evaluate the instantaneous lower limit, as the actual frequency minus the allowed specification at any time. Generally the allowed specification is ±0.005 Hz (±0.0001 pu), but gets larger during faults and dips. Note that fault ride-through specifications mean that the specification does not widen until at least the specified ride-through time has elapse after the inception of the fault. In these simulations, this time was set to 2 cycles (40ms), but this is a configurable parameter for all the algorithms with fault ride-through code.
- Evaluate the lower limit by taking the minimum of all the instantaneous lower limits over the previous 5 cycles (100ms), since this is the required response time of the measurement as defined in Table 2-14.
- The upper limit is defined in a symmetric fashion compared to the lower limit.

The test waveform was synthesised at 1000 Sa/cycle (50 kSa/s) and the hardware low-pass filter was simulated at this same data rate, so as to be sure of accurately representing the attenuation of the higher-order harmonics (see appendix B.2.2).

The instrumentation noise was set to 0.005pu RMS (46dB SNR), with 2 bits RMS additional quantisation noise in the ADC, as specified by Table 2-15. The ADC and 6-tap FIR pre-filter of section 4.5 was implemented at 3000 Sa/s, a 6x over-sampling relative to the main algorithms, for the same reasons discussed in that section.

The main algorithms all execute at 500 Sa/s, 10 samples-per-cycle at nominal 50 Hz (t_s=1/500 s). All algorithms can operate at this low sample rate, although the Jovcic's PLL has been seen to be vulnerable to harmonics at this rate. Jovcic's PLL appears to be much more stable at 2 kSa/s or above.

At all points during the 60-second simulation, the result from each candidate algorithm is compared to the specification lines. An error score is allocated for each candidate algorithm if the algorithm result falls outside the allowed specifications. The error score at

each time step (of length t_s) is set to:-

$$Error = 0\{f_{Measured} > f_{LowerLimit}\}and\{f_{Measured} < f_{UpperLimit}\}$$

otherwise

$$Error = \left(\frac{t_s}{60}\right)\left[\frac{(f_{Measured} - f_{Actual})}{(f_{UpperLimit} - f_{Actual})} - 1\right]\{f_{Measured} > f_{Actual}\}$$

otherwise

$$Error = \left(\frac{t_s}{60}\right)\left[\frac{(f_{Actual} - f_{Measured})}{(f_{Actual} - f_{LowerLimit})} - 1\right]\{f_{Measured} < f_{Actual}\} \quad (5.5)$$

This error scoring system allocates error scores proportionally to the measurement error divided by the specification allowance. Thus, once the specification has been violated, the error score increases slowly for measurement only just outside the specification, but much more quickly for large violations.

The factor of $t_s/60$ in (5.5) scales the errors to an average figure over the 60 second test duration. The error at each time step is then clipped to a maximum value of 100. The cumulative error is taken by adding all the resulting error scores at each time step. Thus, the cumulative error represents the average amount by which the measurement is outside the specification window, on a proportionate basis to the window size. This provides a fair scoring system for comparison of the algorithm performances.

5.7.1 Results overview

It has been found possible to achieve the tightest target specification of ±0.005 Hz in all but a few transient events and at the worst frequencies, by using method 2, the Clarke-FLL hybrid. This is the algorithm of choice, both by the results of Table 5-2 and Table 5-1 . Even during the transient events and at the worst frequencies, the ±0.005 Hz specification is only narrowly missed, and a ±0.01 Hz error specification is almost always achieved.

The overall cumulative error scores for the 8 candidate methods tested, using a basic specification of ±0.005 Hz, and the Waveform 1 and 2 test inputs (Table 2-16 and section 2.11.2) are:-

Algorithm	Cumulative error scores, for waveform 1 test input (see Table 2-16) (average amount by which the measurement is outside the specification window, on a proportionate basis to the window size)	Cumulative error scores, for waveform 2 test input (see Table 2-17), steady state between 44 & 55 Hz (average amount by which the measurement is outside the specification window, on a proportionate basis to the window size)
Clarke transform	0.3343	0.00613
Clarke-FLL hybrid	**0.0094** (2.785 if seeding is disabled)	**0.000877**
Fixed-ref Fourier with seeding	0.0331	0.0124
3x 1-phase PLLs with seeding	0.0872	0.0819
Zero crossings	1.657	0.1876
3x Jovcic's PLLs with weighting	21.31	8.566
3x SPS PLLs with weighting	23.57	3.794
SPS 3-phase PLL	23.68	3.819

Table 5-2 : Summary of cumulative error scores for 8 candidate frequency measurement algorithms, using test Waveforms 1 (Table 2-16) and 2 (Table 2-17)

The results for the waveform 1 test input scenario are investigated in detail in the following sections. Figure 5-18 shows an overview of the frequency profile for the waveform 1 test profile, together with the results for the Clarke-FLL hybrid. The traces are identified as follows:-

- Black solid line : Measurement
- Red dash-dot : Actual synthesised signal frequency
- Blue dashes : Lower and Upper limit lines

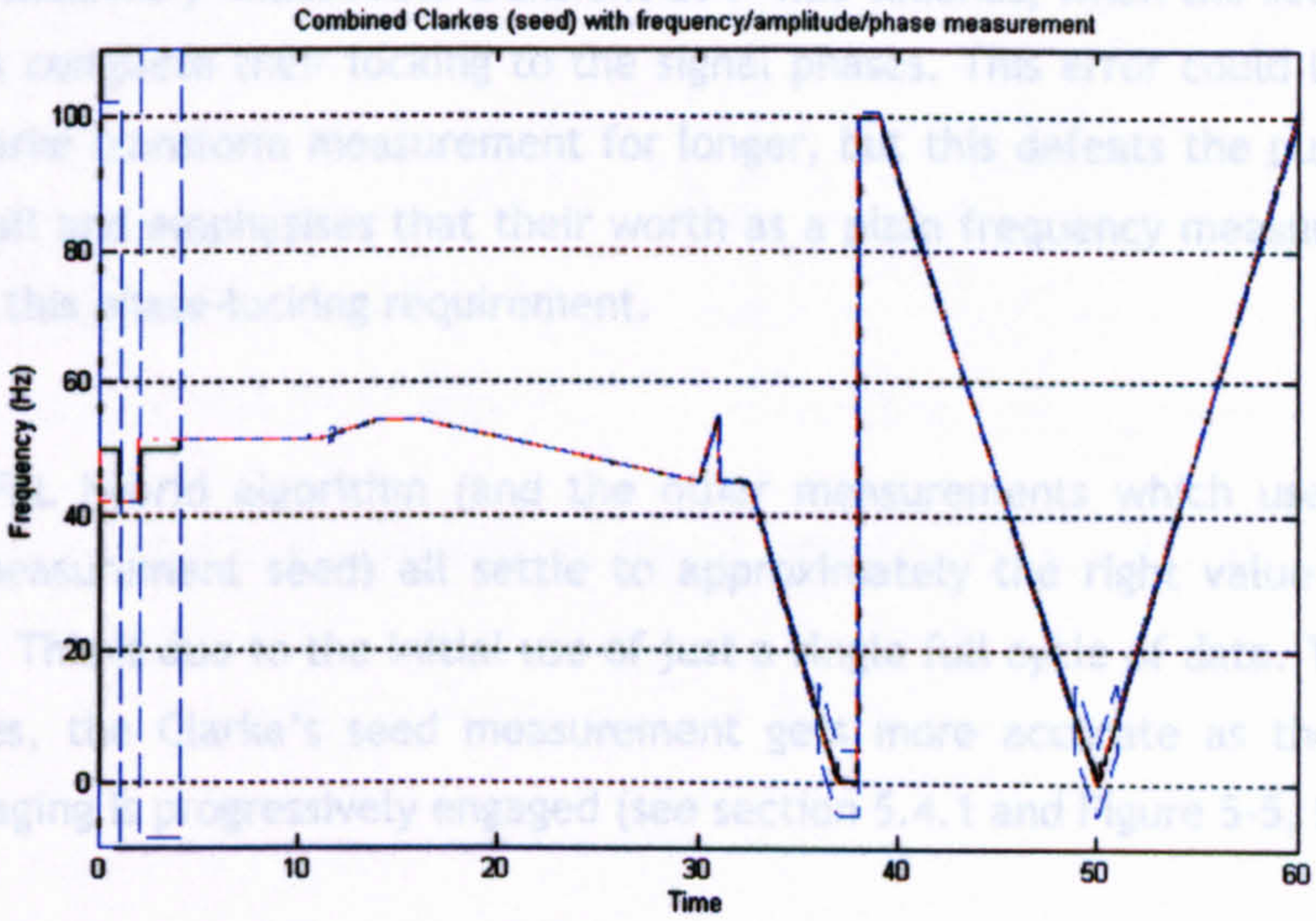


Figure 5-18 : Overview of frequency profile, limit lines & results for waveform 1 (Table 2-16)

5.7.2 Results in detail; initial settling

The simulation deliberately includes an initial period of three-phase signal at 42Hz, balanced 1pu voltage magnitudes between t=1 and t=2s. This has the effect of disturbing all algorithms from any preset values of nominal 50Hz. The first really interesting part of the simulation is thus the time at t=4s, when the signal at 51.282Hz is input. Figure 5-19 shows the results.

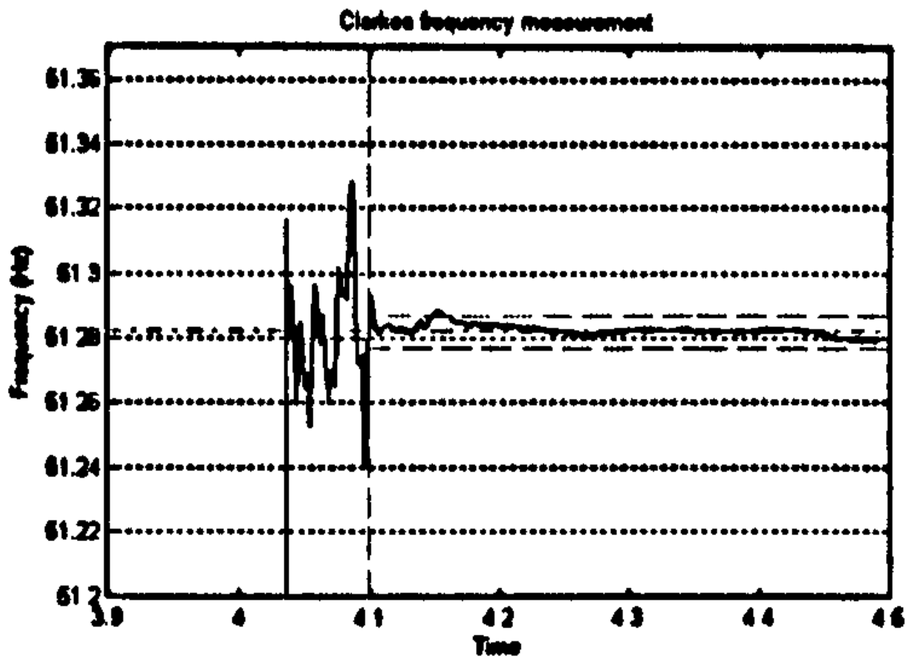
On all the graphs in subsections which follow, the traces are identified as follows:-

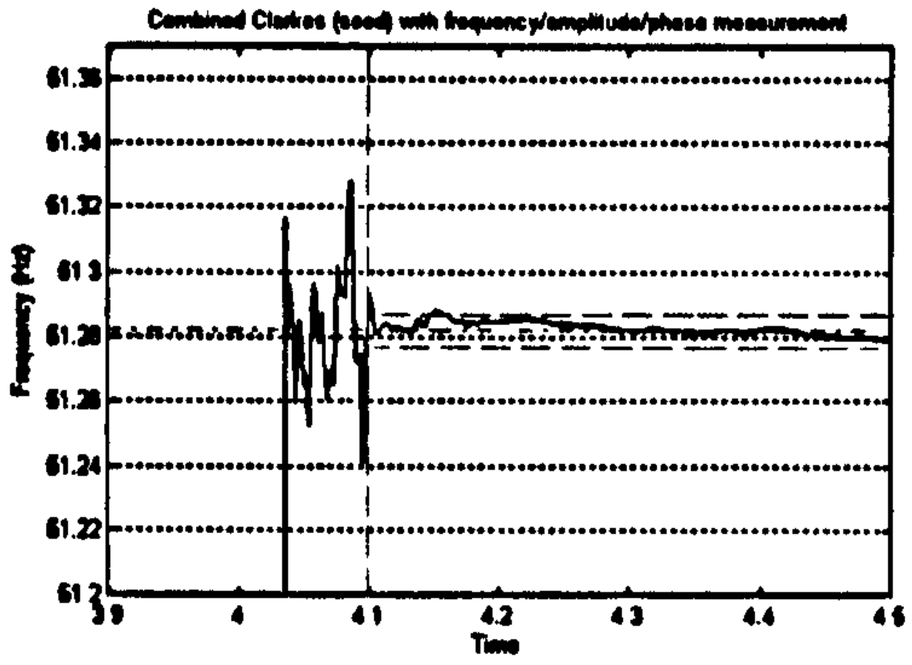
- Black solid line : Measurement
- Red dash-dot : Actual synthesised signal frequency
- Blue dashes : Lower and Upper limit lines

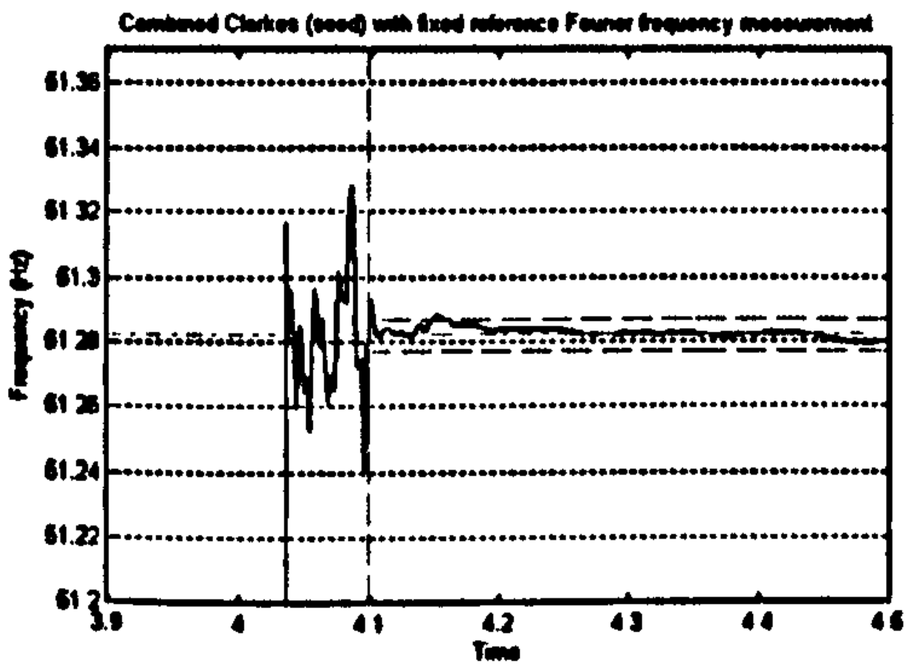
On Figure 5-19 and the following pages, the results of the Clarke-FLL hybrid algorithm (the proposed algorithm) is shown on the top-right plot, labelled "Combined Clarkes (seed) with frequency/amplitude/phase measurement"

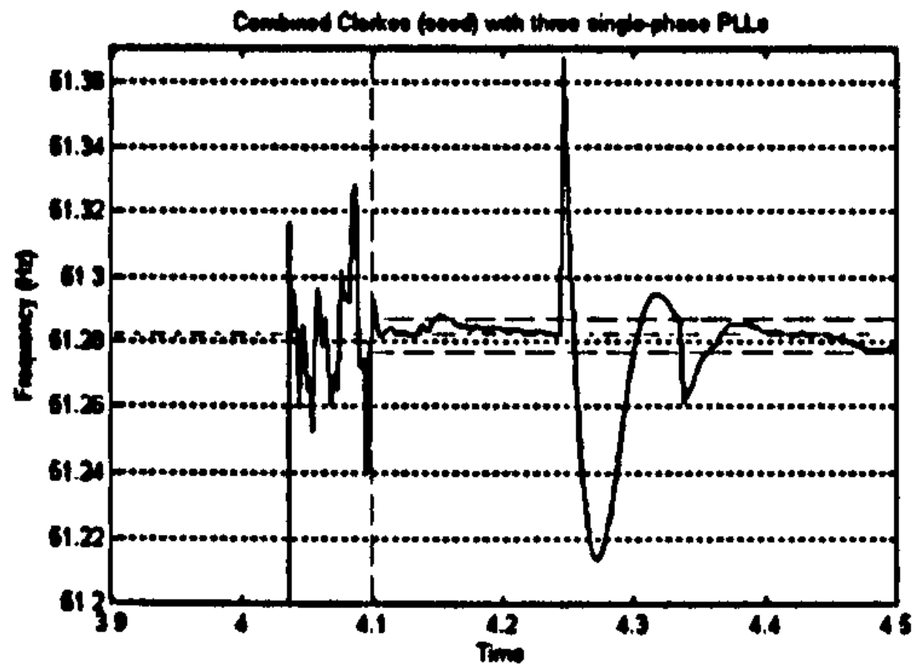
On Figure 5-19, the plots for the 2 SimPowerSystems PLLs and the Jovcic's PLLs need to be displayed on different scales as their settling is much, much slower than the other 5 candidate methods. The seeded methods all settle quickly and accurately, as they use the Clarke transform seed while the main algorithm is settling. The seeded 3x single-phase PLLs show a secondary undesirable transient at t=4.25 seconds, when the seeding turns off and the PLLs complete their locking to the signal phases. This error could be reduced by using the Clarke transform measurement for longer, but this defeats the purpose of using the PLLs at all and emphasises that their worth as a plain frequency measurement device is limited by this phase-locking requirement.

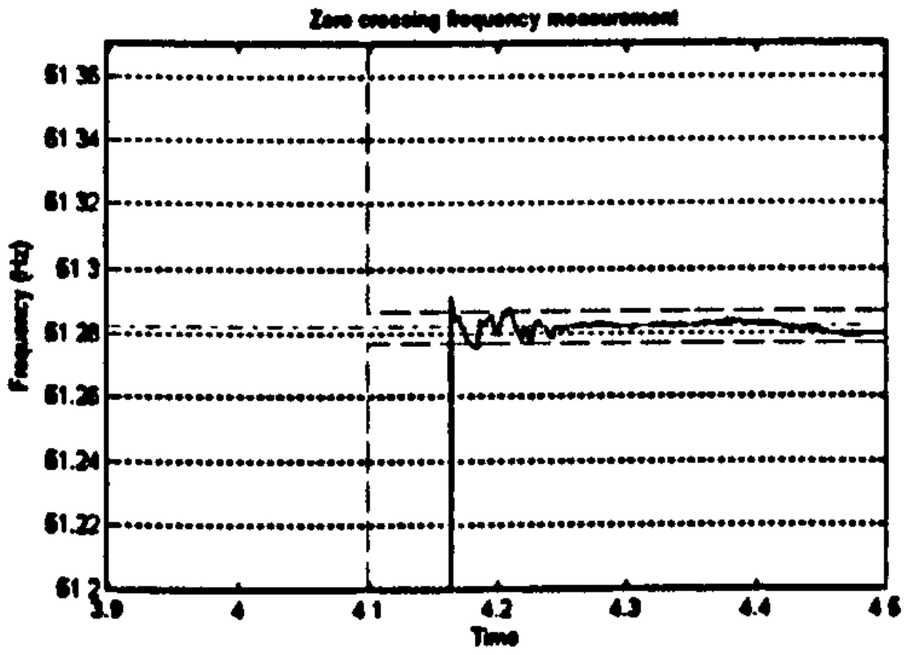
The Clarke-FLL hybrid algorithm (and the other measurements which use the Clarke's frequency measurement seed) all settle to approximately the right value very quickly, within 40ms. This is due to the initial use of just a single full cycle of data. Then, over the next 5 cycles, the Clarke's seed measurement gets more accurate as the 3-cycle and 5-cycle averaging is progressively engaged (see section 5.4.1 and Figure 5-5, Figure 5-6).

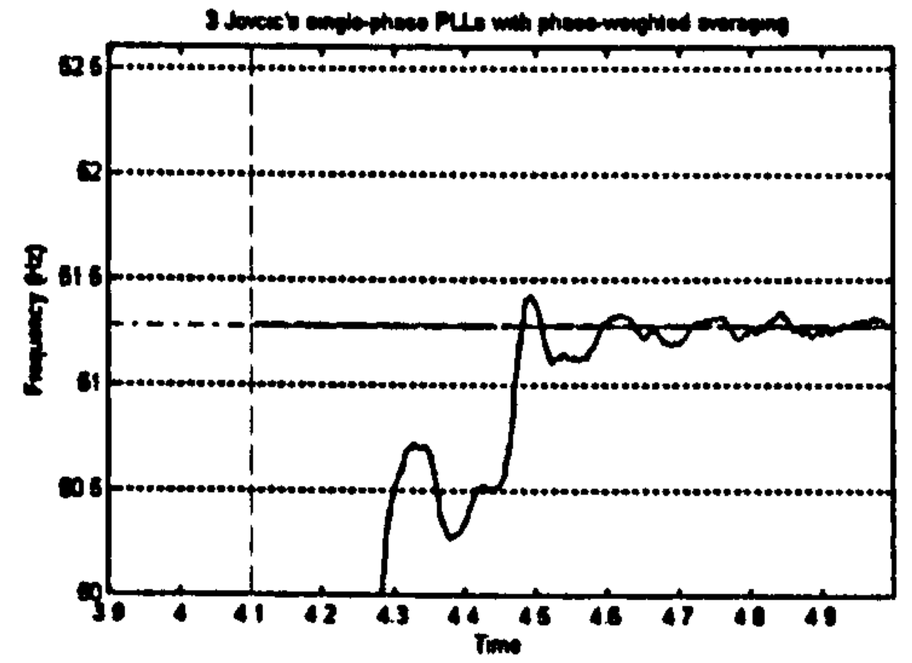


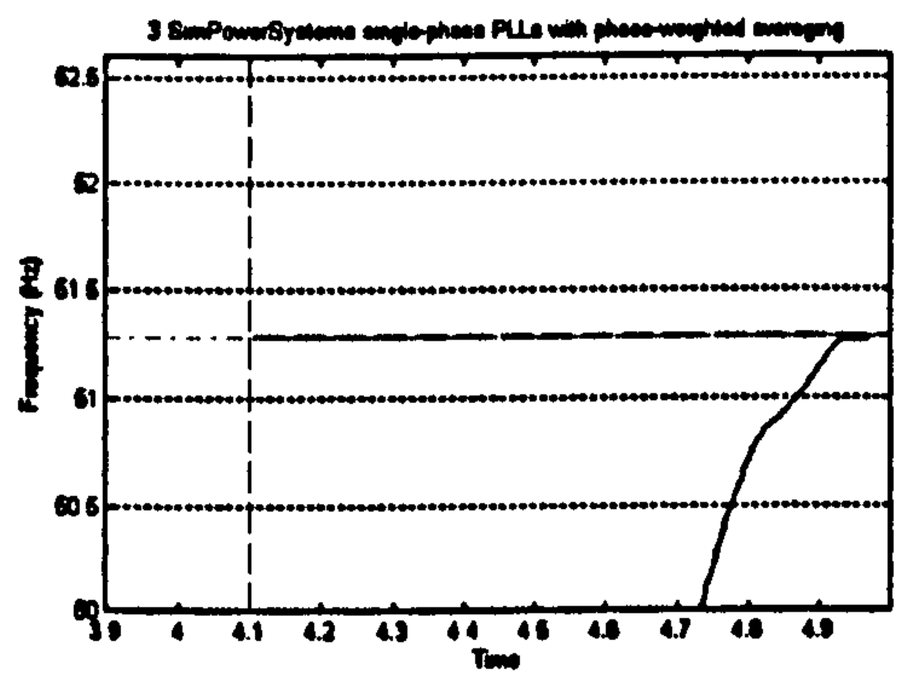












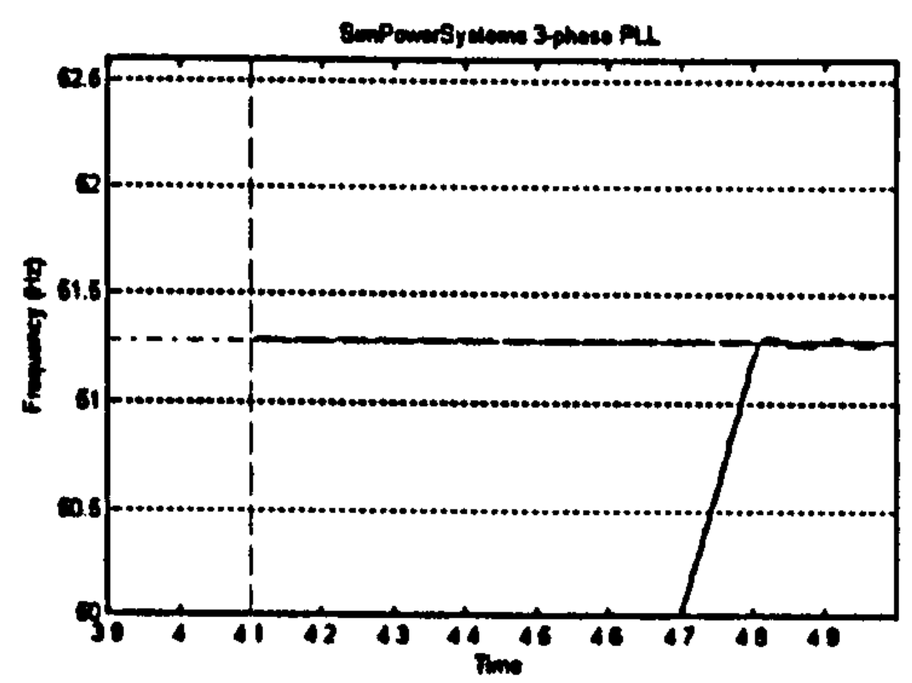


Figure 5-19 : 8 candidate methods; initial settling to 51.282 Hz

5.7.3 Results in detail; increasing influence qualities at 51.282 Hz

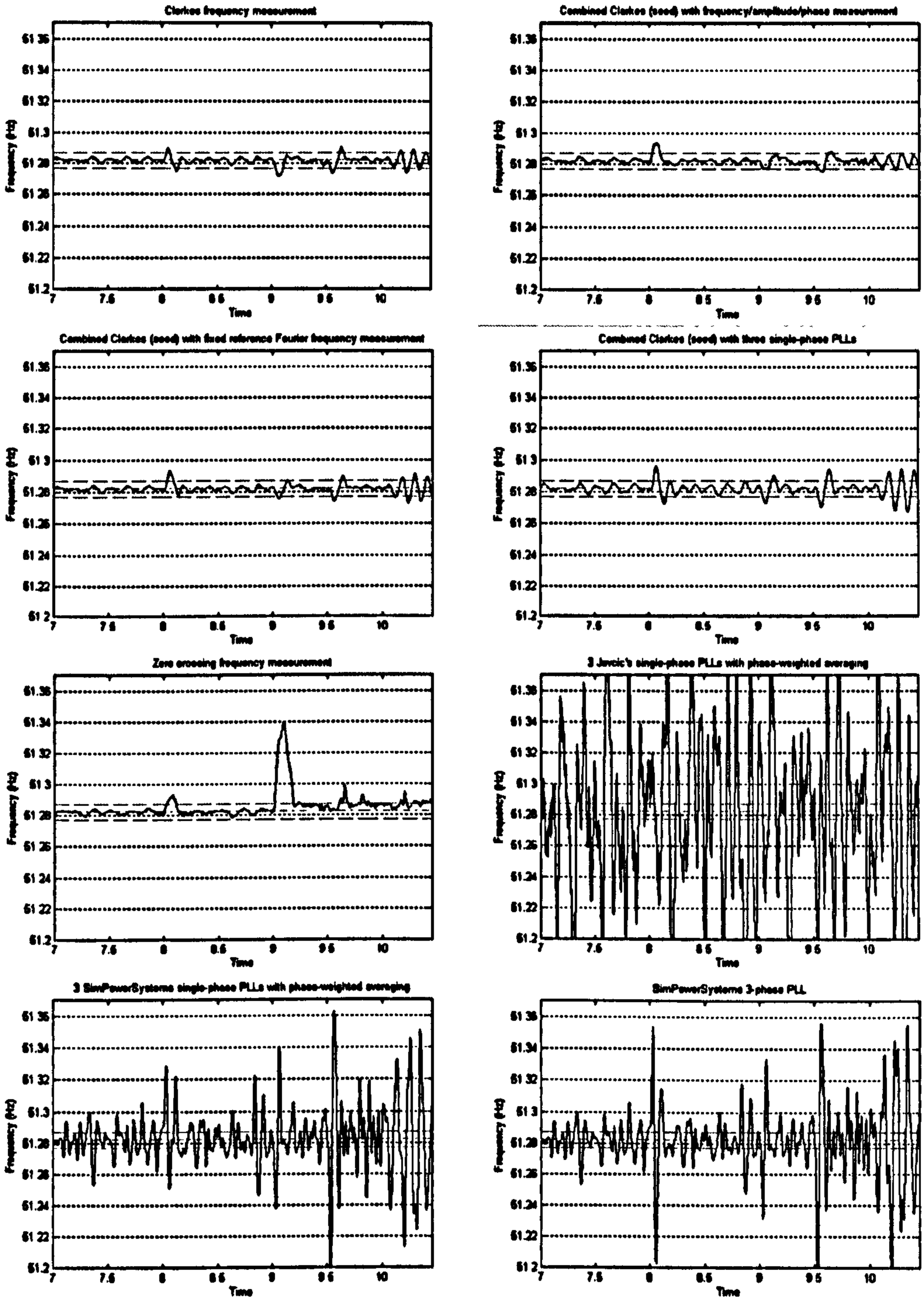


Figure 5-20 : 8 candidate methods; increasing influence qualities at 51.282 Hz

In these plots, 10% unbalance & 2% zero sequence is introduced at t=8s, harmonics (53% THD_V) is introduced at t=9s, and inter-harmonics are introduced at t=9.5s. 8% step flicker @ 13.5Hz is introduced at t=10s. The Clarke-FLL hybrid gives results which are not bettered by any other methods, but equalled by 3 others developed during this thesis. The zero crossing algorithm, Jovcic's PLL and SimPowerSystems PLLs give unacceptable results.

5.7.4 Results in detail; phase jump

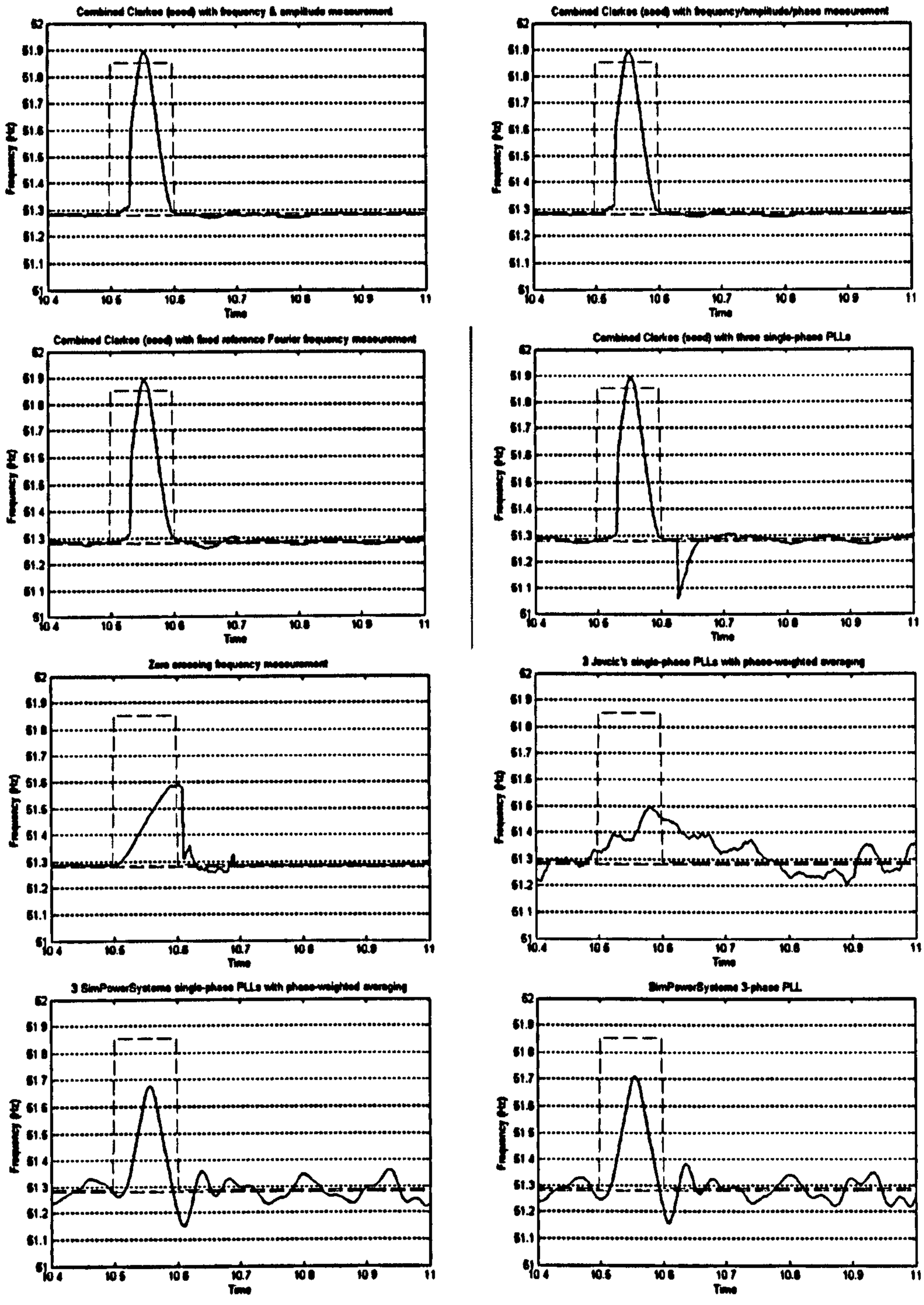


Figure 5-21 : 8 candidate methods; +10 degree phase jump at t=10.5s

Here, the Jovcic PLL shows the least response during the event (mainly because it has a very slow 1.32Hz low-pass filter at its output), but performs badly after the event. The response of the SPS PLLs to the step is limited due to 12Hz/s slew-rate filters which limits their dynamic response. Other algorithms perform adequately.

5.7.5 Results in detail; 40ms 100% three-phase dip to 0pu

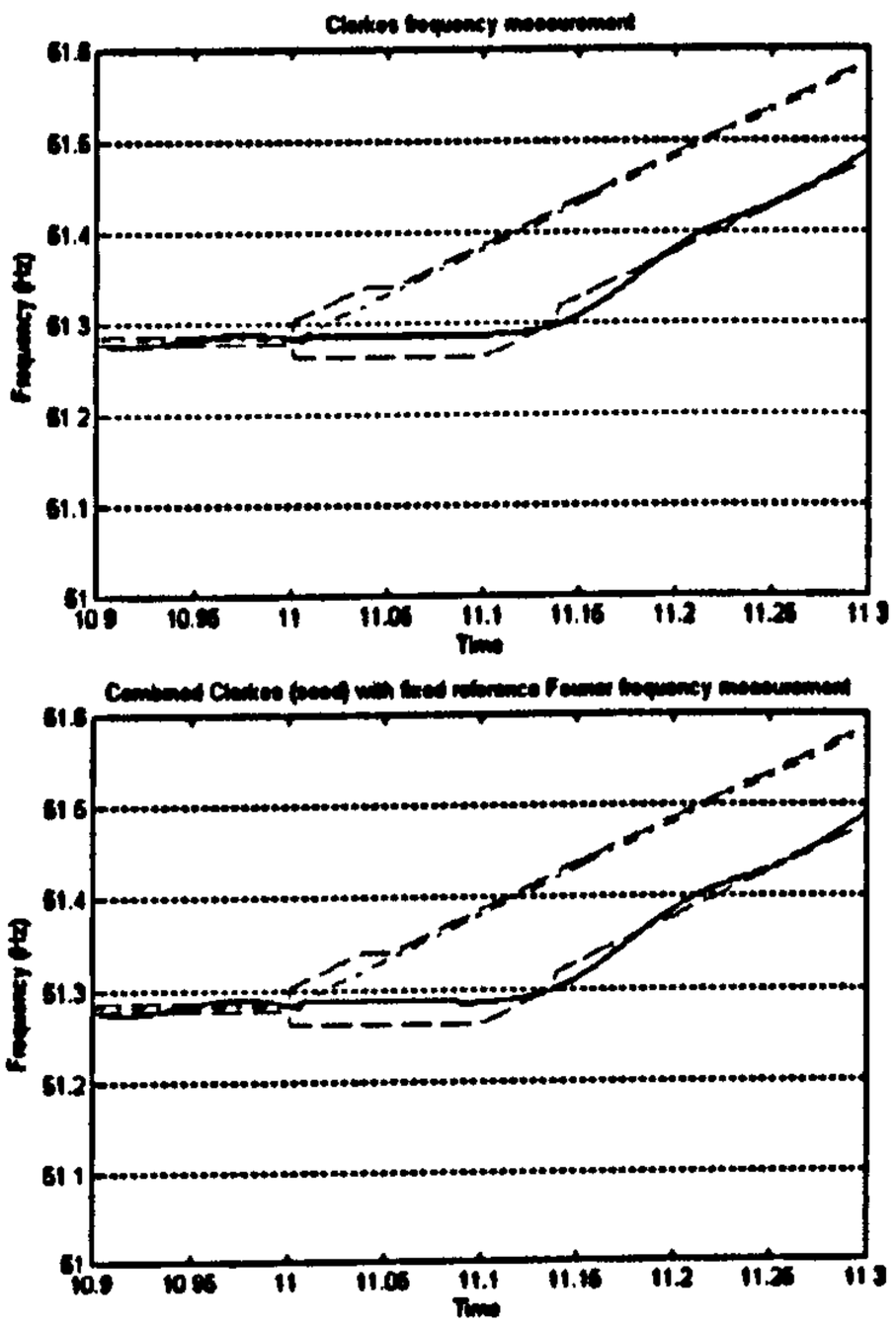


Figure 5-22 : 8 candidate methods; 40ms 100% three-phase dip to 0pu at t=11s

Frequency actually begins to rise during the dip, simulating generator overspeed, but as the dip is 100% deep, there is no way of measuring the signal. The first four algorithms ride through well with a fixed frequency output during the dip. The last four PLL algorithms do not have ride-through and free-run during the fault, which leads to much poorer results.

5.7.6 Results in detail; sustained 95% three-phase dip to 0.05pu and 60% three-phase dip to 0.4pu

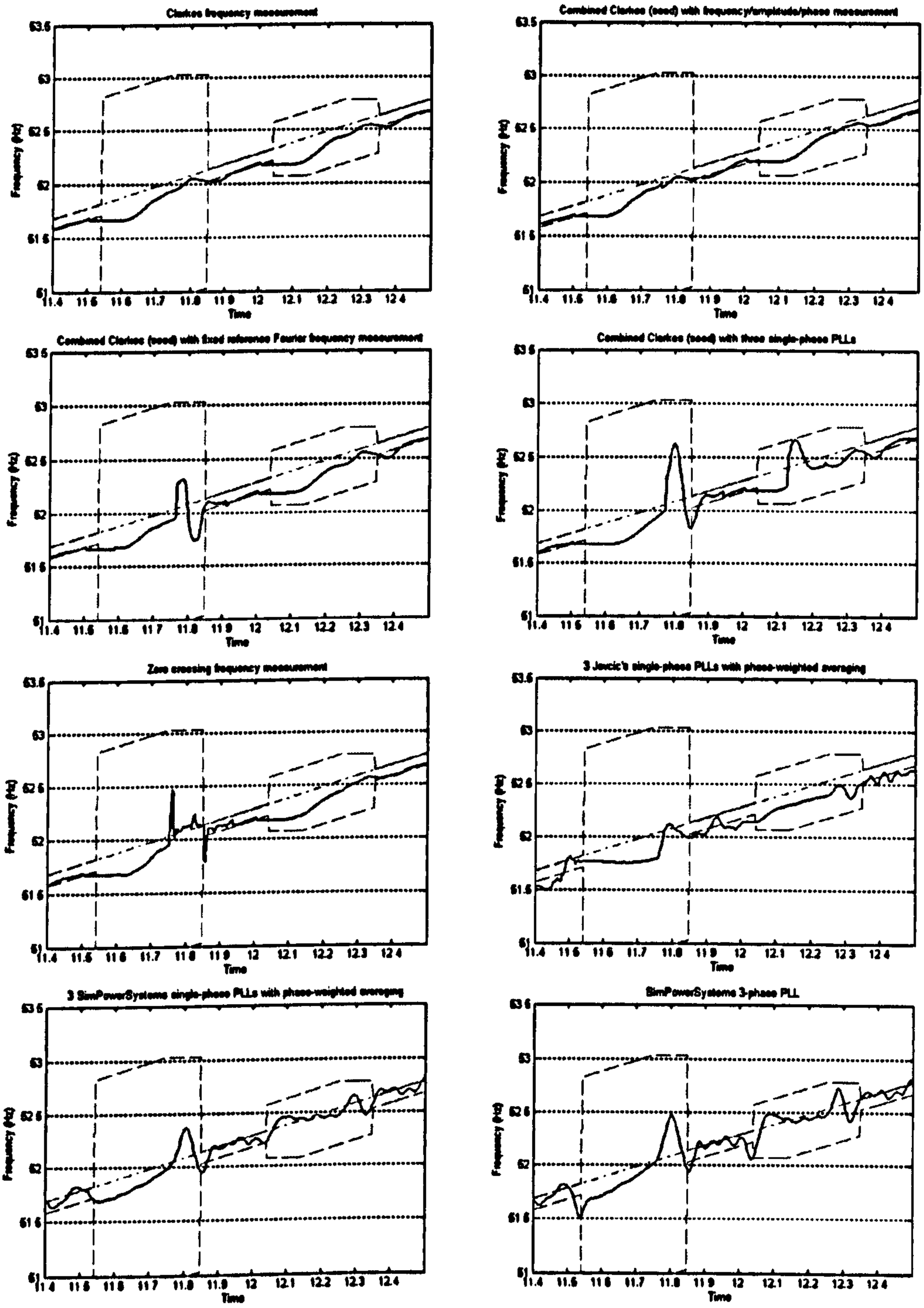


Figure 5-23 : 8 candidate methods; sustained 95% three-phase dip to 0.05pu @ t=11.5s and 60% three-phase dip to 0.4pu @ t=12s

All algorithms perform reasonably during this part of the test waveform, since all three phases are measurable (but the relative noise level is higher). The Clarke transform and the Clarke-FLL hybrid perform the best. Ride-through is briefly activated at the inceptions of the dips.

5.7.7 Results in detail; sustained single-phase fault and two-phase fault

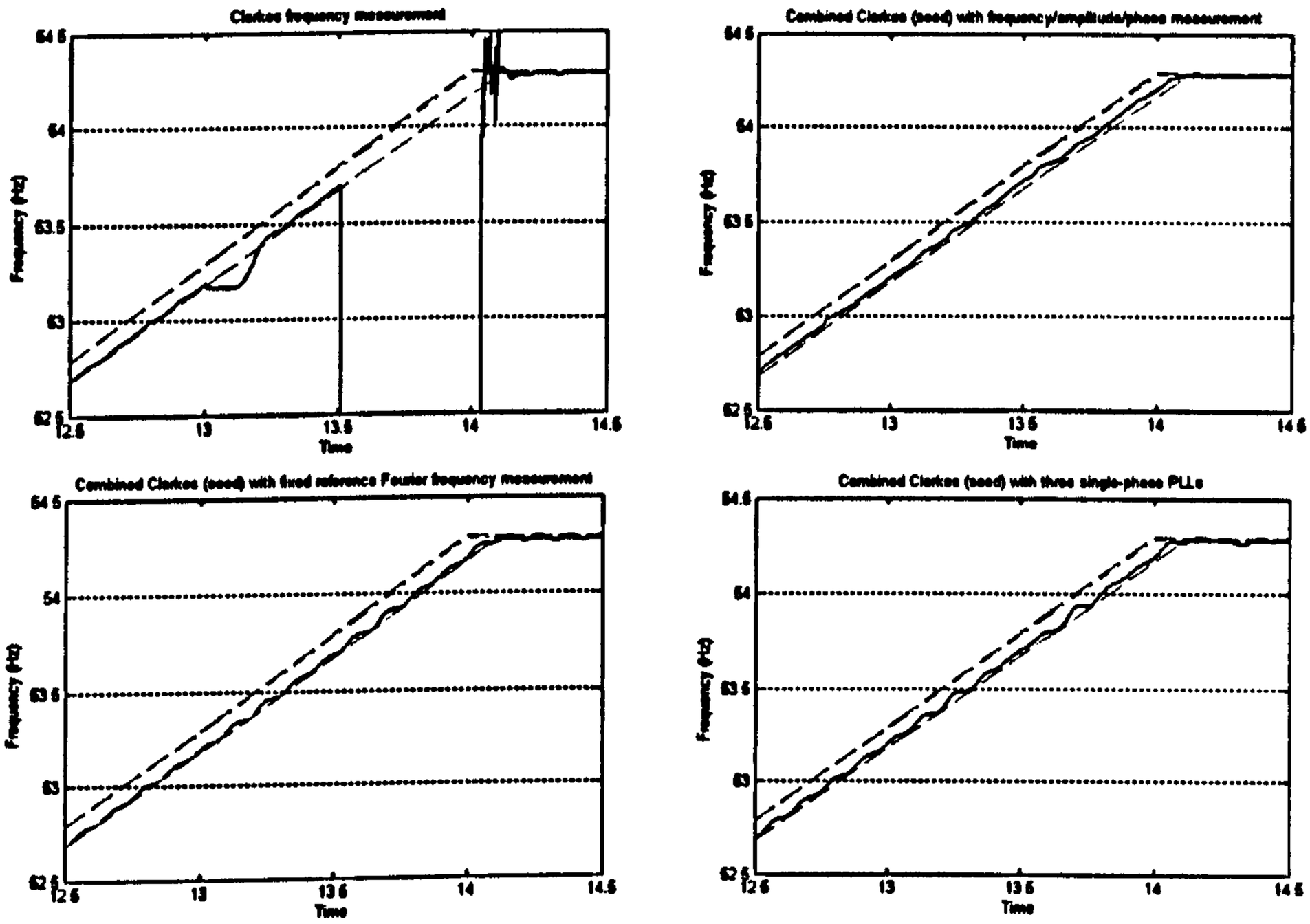


Figure 5-24 : 8 candidate methods; sustained single-phase fault (t=13-13.5s) and two-phase fault (t=13.5-14s)

The Clarke transform measurement becomes invalid during the two-phase fault and gives very poor results. The Clarke-FLL hybrid and the other 2 algorithms developed during this thesis perform well. The zero-crossing, Jovcic PLL, and SimPowerSystems PLLs all show poor performance.

5.7.8 Results in detail; steady state influence qualities at 54.282 Hz

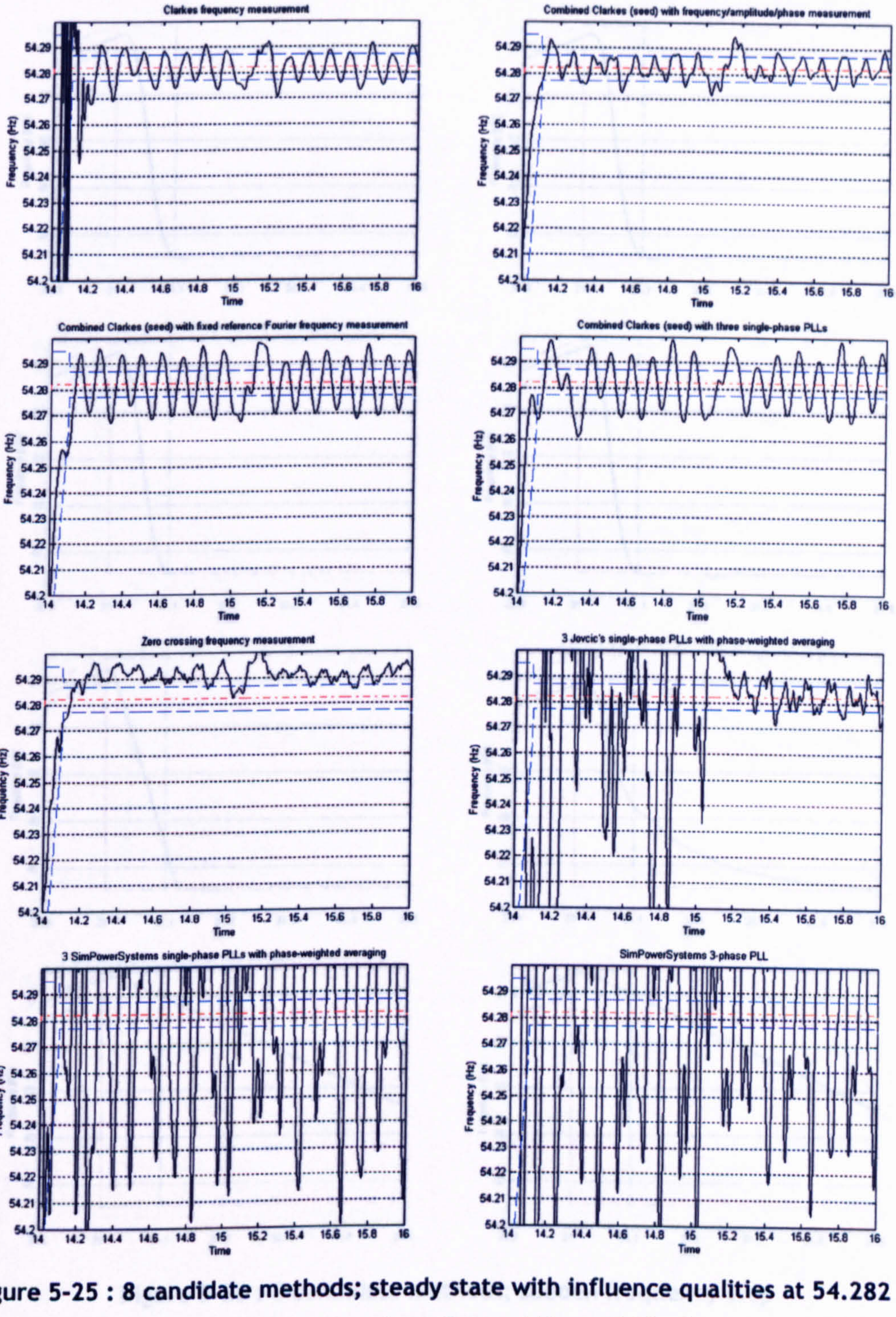


Figure 5-25 : 8 candidate methods; steady state with influence qualities at 54.282 Hz

The results of this scenario are worse than those of Figure 5-20, due to there being only 9.2 samples per cycle, and due to the effect of aliased harmonics. Clearly the Clarke transform and Clarke-FLL hybrid algorithms perform the best. The bias on the zero-crossing measurement is due to inter-harmonics and flicker.

5.7.9 Results in detail; frequency step

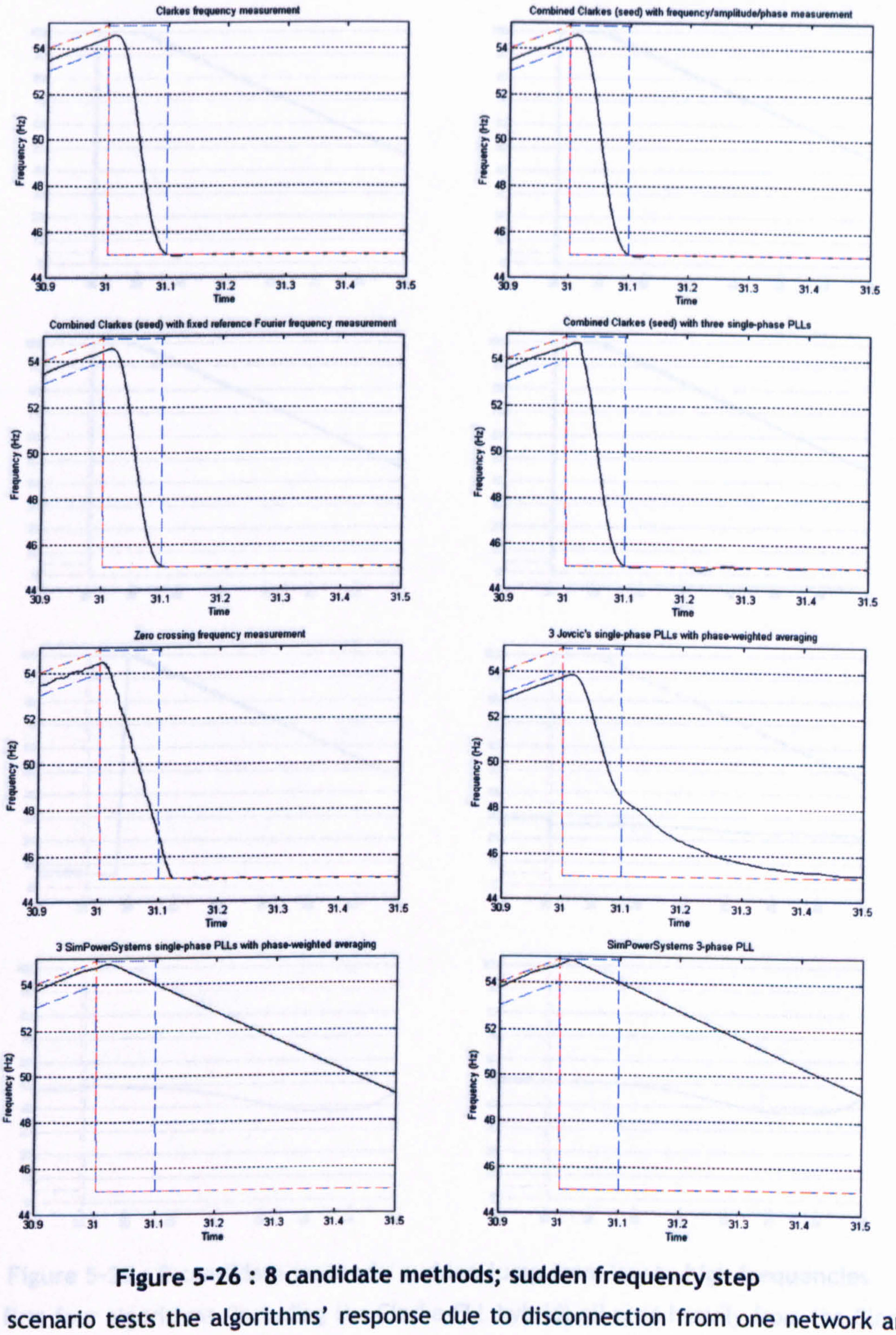


Figure 5-26 : 8 candidate methods; sudden frequency step

This scenario tests the algorithms' response due to disconnection from one network and reconnection to another within a short timeframe. The requirement is for a 5-cycle (100ms) settling time. Clearly the Jovcic's PLL and the SimPowerSystems PLLs apply filtering which does not allow such a step to be followed adequately.

5.7.10 Results in detail; sudden jump from low to high frequencies

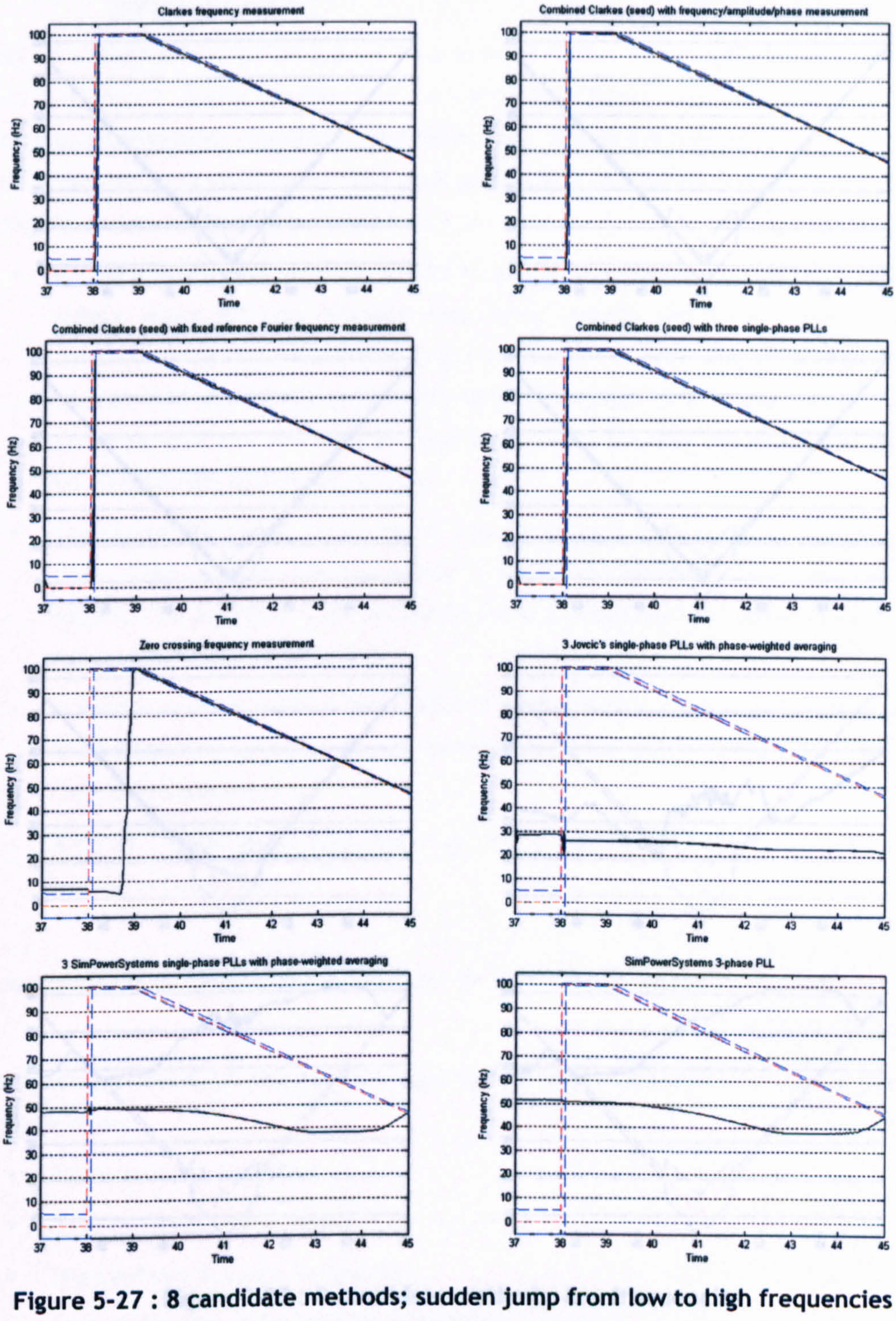


Figure 5-27 : 8 candidate methods; sudden jump from low to high frequencies

The first four algorithms (including the Clarke-FLL hybrid) all seed heavily from the Clarke transform during this scenario, enabling them to track the signal accurately. The Jovcic's PLL and SimPowerSystems PLLs lose lock (before t=37s due to low frequency) and, lacking a seeding algorithm, cannot re-attain it until frequency returns to near-nominal again. In this scenario the Jovcic's PLL does not manage to lock at all during the time period shown.

5.7.11 Results in detail; low frequencies

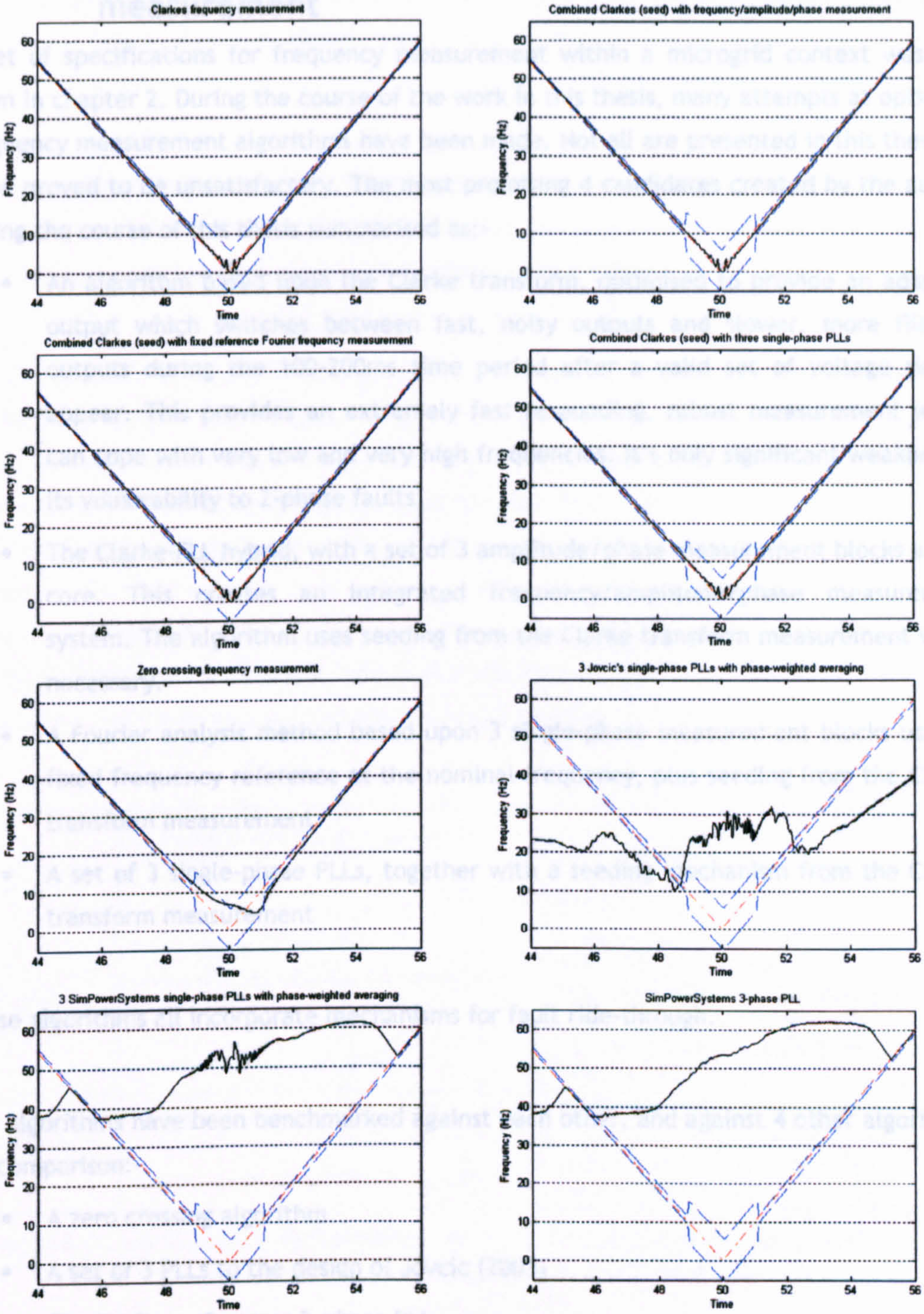


Figure 5-28 : 8 candidate methods; low frequencies

The first four algorithms (including the Clarke-FLL hybrid) all seed heavily from the Clarke transform when frequency drops to levels much less than $f_{nom}/2$, enabling them to track the signal accurately. The Jovcic's PLL and SimPowerSystems PLLs lose lock and, lacking a seeding algorithm, cannot re-attain it until frequency returns to near-nominal again.

5.8 Summary of findings with respect to frequency measurement

A set of specifications for frequency measurement within a microgrid context was laid down in chapter 2. During the course of the work in this thesis, many attempts at optimum frequency measurement algorithms have been made. Not all are presented in this thesis as some proved to be unsatisfactory. The most promising 4 candidates created by the author during the course of this thesis summarised as:-

- An algorithm based upon the Clarke transform, optimised to provide an adaptive output which switches between fast, noisy outputs and slower, more filtered outputs during the 100-200ms time period after a valid set of voltage signals appear. This provides an extremely fast responding, robust measurement which can cope with very low and very high frequencies. It's only significant weakness is its vulnerability to 2-phase faults.
- The Clarke-FLL hybrid, with a set of 3 amplitude/phase measurement blocks at the core. This creates an integrated frequency/amplitude/phase measurement system. The algorithm uses seeding from the Clarke transform measurement when necessary.
- A Fourier analysis method based upon 3 single-phase measurement blocks using a fixed frequency reference at the nominal frequency, plus seeding from the Clarke transform measurement
- A set of 3 single-phase PLLs, together with a seeding mechanism from the Clarke transform measurement

These algorithms all incorporate mechanisms for fault ride-through.

The algorithms have been benchmarked against each other, and against 4 other algorithms for comparison:-

- A zero crossing algorithm
- A set of 3 PLLs to the design of Jovcic (2003)
- The SimPowerSystems 3-phase PLL
- A set of 3 SimPowerSystems 1-phase PLLs

Other algorithms from research literature were considered but did not provide a suitable fit at the low sample rate (500 Sa/s) which was targeted.

The best performing algorithm is the Clarke-FLL hybrid. This algorithm achieves by far the lowest error score in the performance tests (see Table 5-2), and this also shows up in the traces on Figure 5-19 to Figure 5-28. This algorithm has a much better locking characteristic than a PLL when only frequency (and not phase) needs to be measured.

The optimal architecture for the Clarke-FLL system re-uses many of the concepts originally explored and implemented in chapters 3 & 4. This not only leads to an algorithm with very good performance, but also allows substantial code re-use.

For the worst-case microgrid voltage waveforms with 53% THD_V, plus unbalance, inter-harmonics, and flicker,(see section 2.7), it has been found possible to achieve the tightest target specification of ±0.005 Hz in all but a few transient events and at the worst frequencies, with an allowed response time of 100ms (nominally 5 cycles). Even during the transient events and at the worst frequencies, the ±0.005 Hz specification is only narrowly missed, and a ±0.01 Hz peak error specification is almost always achieved. This provides a significantly more robust and faster measurement than specified by BS EN 61000-4-30 (BSI, 2003) for a "Class A" instrument, with twice the accuracy, as shown on Table 5-3. This is despite the low frame rate used within the major algorithms.

	BS EN 61000-4-30 "Class A" performance (BSI, 2003)	Performance of architecture developed in this thesis, worst case scenario
Scenario	Harmonics up to 20% THD (twice times BS EN 61000-4-3 table 5, class 3) (BSI, 2002)	THD up to 53%. 2 cascaded low-pass anti-aliasing filters with cut-off frequency set to 125 Hz. Instrumentation noise level (before the ADC) at or below 0.005pu RMS (46dB SNR). A 12-bit ADC with a bit noise of no more than 2 bits RMS.
Frequency measurement	±0.01 Hz (±0.02%, ±0.0002pu) A measurement time of 500 cycles (10s)	±0.005 Hz (±0.01%, ±0.0001pu) A settling time of 5 cycles (100ms) during transients and a total averaging time of 9 cycles (180ms) normally

Table 5-3 : Performance of frequency measurement architecture proposed by this thesis, versus standard "Class A" performance

Applying this new algorithm to microgrid control systems and generator management systems may lead to the following benefits, relative to other frequency measurement methods:-

- Quicker settling time, hence more reliable information for decision-making algorithms and relays etc.

- Lower measurement latency. Hence, less likelihood of power system oscillations building up due to unintentionally lagged controls.
- Functionality at low sample rates, allowing the algorithm to be deployed on cheap microcontroller platforms in conjunction with other measurement/control algorithms.
- Lower measurement ripple in the presence of influence qualities such as harmonics, inter-harmonics, noise, unbalance and flicker. This leads to smaller power system oscillations due to less ripple being fed through to droop controllers etc...
- The new frequency measurement will provide a much better basis for a loss-of-mains relay based upon ROCOF, or an under/over-frequency relay, due to the combination of the above improvements. This is explored further in chapter 6.

5.9 References for chapter 5

Aghazadeh, R. et al. (2005). New technique for frequency and amplitude estimation of power system signals. IEE Proceedings on generation, transmission and distribution, vol. 152, no. 3, pp 435-440.

Awad, H., Svensson, J. & Bollen, M. (2005). Tuning software phase-locked loop for series-connected converters. IEEE transactions on power delivery, vol.20 issue 1. pp 300-8.

BSI, British Standards. (2000). Voltage characteristics of electricity supplies by public distribution systems. BS EN 50160:2000. ISBN 0 580 35925 5

BSI, British Standards. (2002). Electromagnetic compatibility (EMC) — Part 2-4: Environment — Compatibility levels in industrial plants for low-frequency conducted disturbances. BS EN 61000-2-4:2002. ISBN 0 580 40489 7

BSI, British Standards. (2003). Electromagnetic compatibility (EMC) - Part 4-30: Testing and measurement techniques - Power quality measurement methods. BS EN 61000-4-30. ISBN 0 580 42081 7

Canteli, M. et. al. (2006). Three-phase adaptive frequency measurement based on Clarke's transformation. IEEE transactions on power delivery vol.21 issue 3. pp 1101-5.

Choi, J., Kim, Y. & Kim, H. (2006). Digital PLL control for single-phase photovoltaic system. IEE proceedings on electric power applications, vol.153 issue 1. pp 40-6.

Chung, S. (2000). Phase-locked loop for grid-connected three-phase power conversion systems. IEE proceedings on electric power applications, vol. 147 issue 3. pp 213-19.

Clarke, E. (1943). Circuit analysis of a-c power systems. Wiley.

Dash, P. et. al. (2000). An extended complex Kalman filter for frequency measurement of distorted signals. IEEE transactions on instrumentation and measurement, vol.49 issue 4. pp 746-53.

Han, B. & Bae, B. (2006). Novel phase-locked loop using adaptive linear combiner. IEEE transactions on power delivery Han, vol.21 issue 1. pp 513-14.

Ignatova, V. et. al. (2005). Classification and characterisation of three-phase voltage dips by space vector methodology. International conference on future power systems. P 1600556.

Johns, A. & Salman, S. (1995). Digital protection for power systems. IEE Power Series 15, ISBN 086341303X.

Jovcic, D. (2003). Phase locked loop system for FACTS. IEEE transactions on power systems vol.18 issue 3. pp 1116-24.

Lin, T. & Domijan, A. (2005). On power quality indices and real time measurement. IEEE Transactions on power delivery, vol. 20 no. 4, pp 2552-2562.

Lopez, A. et al. (2008). Power system frequency measurement under nonstationary situations. IEEE Transactions on power delivery, vol. 22 no. 2, pp562-567.

Moore, P., Carranza, R. & Johns, A. (1996a). Model system tests on a new numeric method of power system frequency measurement. IEEE Transactions on power delivery, vol. 11 no. 2, pp 696-701.

Roscoe, A. (2005). Systems and methods for performing analysis of a multi-tone signal. US Patent number US 6,873,923. March 29th, 2005.

Salcic, Z., & Mikhael, R. (2000). A new method for instantaneous power system frequency measurement using reference points detection. Electric power systems research vol.55 issue 2. pp 97-102.

6 Loss-Of-Mains detection

Within this chapter, the frequency measurement algorithm developed in chapter 5 is integrated within a novel new relay for loss-of-mains (LOM) detection. Initially, this relay is investigated in simulation (see section 6.1), and then using real data from captured network events in section 6.2. Finally, in section 6.3, the LOM-detection relay is augmented by a novel microgrid management strategy. This strategy makes small adjustments to the reactive power flow between a microgrid and its parent network. This allows an exact active power match within the microgrid (generation to load), while still avoiding the non-detection zone (NDZ) of the LOM-detection relay. The combined measurement and management algorithms are embedded in a real-time microcontroller platform and tested in a real microgrid power system at the 2kVA-100kVA scale.

Previous work at the University of Strathclyde has analysed the performance of several commercially-available LOM (loss-of-mains) relays and their performance (Dysko, 2006, 2007, 2007b). A LOM event occurs when an upstream fault causes disconnection of a local power system from a parent network. The disconnection can be caused by a permanently or temporarily severed line or cable, or by a short circuit fault which causes a permanent or temporary circuit-breaker trip.

Detection of LOM is important for two main reasons, when generation sources are distributed within a power network:-

- To ensure safety against electric shock, LOM must be detected within a reasonable timeframe, such that generators within islanded networks will be disconnected or managed appropriately in a deliberate islanded state.
- To avoid damage to generators and distribution equipment, loss-of-mains should be detected such that subsequent reconnections to the parent network do not occur when the distributed generator(s) are out of synchronisation with the parent network.
- If LOM is not detected and a generator continues to operate in grid-connected mode, excursions to frequency and voltage may result which might violate power quality expectations and cause damage to loads.

During normal operation, distributed generators are conventionally operating in grid-connected mode, with real and reactive (P & Q) power output targets set at the generator, in combination with appropriate droop controls. When a loss-of-mains event occurs, the generators are no longer connected to the parent network. In this

configuration, the grid-connected control strategies will not generally keep the frequency and voltage of the power island at the same pre-disconnection values. The overall system may stabilise at some new (and acceptable) values of frequency and voltage, or the frequency and/or voltage may rise or fall outside acceptable limits, resulting in disconnection. The worst case generally presented for LOM detection is where the local real and reactive loads are, by chance, exactly (of very almost) equal to the local generation target outputs. In this case, in theory, the system could continue to operate at the same frequency and voltage, in which case the system is said to be within the "non detection zone" (NDZ) of the LOM relay. In most practical cases, however, there will be some load/generation imbalance in either P or Q, and any imbalance will cause both frequency and voltage excursions. The bigger the imbalance, the bigger the excursions.

The report by Dysko (2006) analyses different types of conventional LOM-detection relays under different load/generation imbalances and different non-LOM fault conditions, to assess their sensitivity (when local generation is closely matched to local loads) and discrimination (against non-LOM fault conditions, noise, harmonics, flicker, transients etc.). In this case, the non-LOM faults considered are limited to single, two and three-phase faults at various positions with a distribution network. The discrimination against non-LOM disturbances causing spurious tripping of LOM relays has been identified as a major current problem with many commercial and proposed LOM detection relays (Dysko, 2007).

In this section of the thesis, a new algorithm for detecting the LOM condition, based purely on local passive measurements of voltage, is proposed. Later, in section 6.2, a novel DG control strategy is also proposed and tested. The combination of the relay and the control strategy allows detection of LOM within 1-2 seconds during all scenarios, by deliberately just avoiding the non-detection-zone (NDZ) (Ye, 2004) of active and reactive power balance.

Generally the commercially available passive LOM-detection relays fall into two types: ROCOF (Rate of change of Frequency) and Vector Shift. Vector-shift relays are analysed in Dysko (2006) & Freitas (2005), and are found to be less suitable due to their relative inability to detect genuine LOM events. The ROCOF relays perform better, but when choosing the trip settings for commercial ROCOF relays, dilemmas are faced when trying to meet the demands of both sensitivity and discrimination. This is the reason that ROCOF relay settings in practice are set as low as 0.1 Hz/s, but up to 1.2 Hz/s (Vieira, 2006c), while the natural frequency variations of the grid to which these relays are connected are

much less than 1.2 Hz/s (see Table 2-1 and section 6.2). A review of other candidate passive LOM detection strategies is given in Dysko (2007), which highlights the fact that most passive LOM detection strategies can be made to be very sensitive, but few authors have paid attention to the issue of discrimination and spurious tripping. The new methods proposed in Dysko (2006) provide better results than commercially available ROCOF and vector-shift relays, but still a single setting has not be found for these algorithms which achieves both the sensitivity and discrimination required. (Vieira, 2006a) examined the use of under-frequency and over-frequency relays to detect loss-of-mains. Such a relay is good at avoiding spurious trips, but has the disadvantage of having to wait until the frequency excursion is quite large before detecting the loss-of-mains condition. This means that ride-through action (trying to switch to a stable and deliberate power island) after detection will be hard, as the frequency will be verging on allowed limits at the instant of changeover.

It should also be mentioned that there are many works concerning active loss-of-mains methods which are specific to inverter-connected generation. These techniques involve injection of current waveforms with deliberate harmonic content, either in pulsed or steady-state modes. Good examples of these are Huang (2001), Timbus (2004), and Sumner (2004a, 2004b). These methods require specific inverter hardware or current injection apparatus and are "active" forms of detection. Such methods are dependent upon the injection hardware functioning correctly and can therefore be regarded as less reliable than "passive" methods such as that proposed in this work. Where many such active devices are placed in the network, further risks arise such as degradation of power quality, and interference between the multiple current injection waveforms. The work in this thesis is targeted at strategies for "passive" detection of loss-of-mains which are independent upon generator type, and so these methods are not applicable.

6.1 Application to loss-of-mains detection (simulations)

6.1.1 Proposed new loss-of-mains detection algorithm - Phase Offset Relay (POR)

This thesis proposes a new algorithm, based most closely upon the principle of "Method 2" given by Dysko (2006). There are also some parallels (and many differences) to a method proposed by Wall (2004). Compared to the method proposed by Dysko (2006), the proposed algorithm contains the following major differences:-

- the robust integration algorithms of section 3.2 are used, avoiding integrator wind-up
- a triggering/resetting subsystem to avoid constant tripping, in place of the

high-pass filter used by Dysko(2006). This allows setting of the trigger threshold by a meaningful parameter, the expected ROCOF levels during normal system operation, instead of by a high-pass filter cutoff frequency which has little obvious physical significance.

- An allowance for variable phase offset trip thresholds during and immediately following faults. This significantly enhances the discrimination of the relay against faults and disturbances, without reverting to a (dangerous) complete blocking of the trip signal.

The new algorithm is called a Phase Offset Relay, or POR.

The relay operates on the principle that the phase offset of a system relative to a stable frequency can be estimated by

$$\phi = 2\pi \int \left(\int ROCOF \cdot dt \right) \cdot dt = \pi \cdot ROCOF \cdot t^2 \text{ where } \Phi \text{ is in radians} \qquad (6.1)$$

or

$$\phi = 360 \int \left(\int ROCOF \cdot dt \right) \cdot dt = 180 \cdot ROCOF \cdot t^2 \text{ where } \Phi \text{ is in degrees} \qquad (6.2)$$

Equation (6.2) can be inverted to reveal trip times for given (constant) ROCOF values and trip thresholds:

$$t = \sqrt{\frac{\phi}{180 \cdot ROCOF}} \text{ where } \Phi \text{ is in degrees} \qquad (6.3)$$

The advantages of this algorithm over a ROCOF algorithm are:-

1. Since the measurement of frequency is "hard" to make without noise on the measurement, the differentiation of frequency to obtain a value of ROCOF leads to an even noisier measurement. Using such a noise measure directly within a relay increases the risk of spurious trips (or non-detection of events). By taking the double-integral (equivalent to a double-averaging) of ROCOF, the noise is substantially reduced.
2. The double-integral stage also gives an answer with physical significance, i.e. the approximate phase deviation of the local voltage measurement from the steady-state value. This phase represents the risk of damage due to out-of-phase re-synchronisation.

3. The trip time for traditional ROCOF relays is generally a fixed time; i.e. >0.2Hz/s for 200ms must be measured for a trip. This neither allows tripping in less than 200ms when ROCOF is large, nor tripping in more than 200ms when ROCOF may be only slightly higher than 0.2Hz/s. It should be noted that a typical ROCOF relay setting of 0.2Hz/s for 0.2 seconds corresponds to a phase offset of only 1.4° which would be of no concern should an auto-reclose action occur. Given that up to 2 seconds is allowed under IEEE 1547 (IEEE, 2003)[1] for LOM detection, at a rate of 0.2Hz/s a trip time of 745ms would be more appropriate since this would be the time at which a 20° phase offset occurred between local generation and the parent network (by equation (6.3)). The phase offset relay does exactly this, using a fixed angle setting but a flexible trip time. This different approach allows much better discrimination between genuine LOM events and other disturbances such as normal load steps which cause brief frequency deviations.

The inner core of the algorithm is shown here mathematically. The real implementation is all coded digitally in discrete time steps.

1. Measure frequency f (by the algorithms of chapter 5)
2. Calculate $ROCOF = \frac{df}{dt}$. This can be done on a two-sample basis. Noise and/or spikes on this signal is acceptable as it will be averaged out by two cascaded integration steps.
3. Calculate frequency offset $\Delta f = \int_{t_t}^{t} ROCOF \cdot dt$ where t_t is the time at which the relay triggering starts. Note that Δf is equivalent to the measured value f, with the DC component removed. The differentiation and integration stages are a convenient way of accomplishing this, while also producing a value of ROCOF for indication (although it may be noisy and takes no direct part in the relay operation).

---

[1] The UK equivalent document, ER G59/1 (ENA, 1991), says only that "The setting of the relays should be agreed with the PES". ETR 113 (ENA, 1995) states than detection within 1 second may be required in some UK rural distribution networks due to potential auto-reclose times in feeders, but this is not a general requirement across the UK. ETR 113 also gives no indication of practical relay settings for DG installations with potential DG-to-load power matches of less than 0.1pu which enable such detection to be practically achieved using a ROCOF or similar relay.

4. Calculate phase offset $\phi = 360\int_{t_t}^{t} \Delta f \cdot dt$ where t_t is the time at which the relay triggering starts, and Φ is the phase offset in degrees.
5. Check the magnitude of the phase offset, abs(Φ) against the trip threshold Φ_{Trip}. If the trip threshold is exceeded, a trip signal is generated. No qualifying time is required since the double-integrated signal is very clean and free from noise, and the initial triggering algorithms (see below) remove spurious trips due to normal fluctuations of the network frequency due to load changes, generator despatching, switching etc..

A triggering algorithm is also required; otherwise the integrals above would lead to tripping due to the normal small fluctuations in frequency which occur in all power networks. Adding the triggering and implementing the entire algorithm requires a number of parameters and variables to be set and calculated. These are tabulated below for clarity.

Symbol	Meaning	Typical parameter value (or calculated variable)	Derivation
f	Frequency	Measured	typically 40-70Hz
ROCOF	Rate of Change of Frequency	Calculated	df/dt (two-sample basis)
Δf	Frequency variation	Calculated	$\Delta f = \int_{t_t}^{t} ROCOF \cdot dt$
t	Time now	Implicit	
t_t	Time when the relay is triggered	Implicit	
Φ	Phase offset (degrees)	Calculated	$\phi = 360\int_{t_t}^{t} \Delta f \cdot dt$
Φ_{Trip}	Phase offset trip setting	Typically 20 degrees	Set such that an accidental resynchronisation does not cause excessive currents. For a generator with leakage reactance 0.1pu, a 20 degree synchronisation results in currents of approximately sin(20°)/0.1=3.4 pu. Also, this setting determines trip times.
f_{Nom}	Nominal frequency	Typically 50Hz or 60Hz	

$R_{Trigger}$	Trigger ROCOF threshold	Typically 0.2 Hz/s	Set by the expected ROCOF during normal network operation (not accounting for close-in faults but allowing for distant faults)
$N_{TriggerCycles}$	Number of nominal cycles over which to evaluate triggering window	Typically 5	Set as high as possible, with the limit that the shortest LOM trip time will then be $t_{Trigger}$
$t_{Trigger}$	Trigger window time	Calculated (constant)	$N_{TriggerCycles}/f_{Nom}$
$\theta_{Trigger}$	Triggering phase angle threshold (degrees)	Calculated (constant)	$360.R_{Trigger} \cdot t_{Trigger}^{2}$
Δf_T	Frequency change during trigger window	Calculated	Rolling definite integral of ROCOF over last $t_{Trigger}$ seconds
θ	Phase change during trigger window	Calculated	360 times rolling definite integral of Δf_T over last $t_{Trigger}$ seconds

Table 6-1 : Parameters and variables for the Phase Offset Relay (POR)

The triggering threshold is set by an estimate of the expected upper level of expected ROCOF, $R_{Trigger}$, and a setting parameter $N_{TriggerCycles}$ that defines a time period $t_{Trigger}=N_{TriggerCycles}/f_{Nom}$ over which the triggering signal is evaluated. Over this timeframe, the steps 3-4 (above) are evaluated using rolling time windows of fixed time length $t_{Trigger}$, to calculate the frequency offset Δf_T and phase offset θ which has occurred over the last $t_{Trigger}$ seconds. This can then be checked against a phase offset trigger threshold $\theta_{Trigger}$ which would occur due to a constant ROCOF of $R_{Trigger}$.

At first glance, $\theta_{Trigger}$ would be derived directly from (6.2), given $t_{Trigger}$ and $R_{Trigger}$. but here the value of $\theta_{Trigger}$ is determined in a slightly different manner. This is because the triggering is evaluated over rolling windows of constant length $t_{Trigger}$ seconds, where $R_{Trigger}$ is also constant. Thus, evaluating

$$\theta_{trigger} = 360 \int_0^{t_{Trigger}} \left(\int_0^{t_{Trigger}} R_{Trigger} \cdot dt \right) dt = 360 \cdot R_{Trigger} \cdot t_{Trigger}^{2} \qquad (6.4)$$

where $\theta_{Trigger}$ is in degrees, i.e. double the value suggested by (6.2)

As well as triggering the main integrations within the trip detection algorithms, the triggering subsystem also has the power to reset and zero the trip detection algorithms. This means that sometimes the relay may be triggered (the main integrators of steps 3 and 4 above may begin to accumulate frequency and phase offset amounts), but will not trip. These events will occur when measured ROCOF rises above the trigger level $R_{Trigger}$ but the

subsequent frequency disturbance is not sufficient to cause an actual trip by violation of $\Phi>\Phi_{Trip}$ *or* $\Phi<-\Phi_{Trip}$. Such functionality is crucial to the discrimination function of the relay, to avoid spurious trips.

In the algorithm presented here, the triggering subsystem causes such a reset when the value θ evaluated over the rolling window of the triggering subsystem undergoes a zero crossing. Because the values of θ are evaluated using double integration of the ROCOF signal, these zero crossings are well defined and relatively noise-free. An alternative method considered was to reset the subsystem when the ROCOF value passed through a zero crossing, which in some ways is more desirable. (When ROCOF passes through 0, the frequency must be either flat, or at a maxima/minima, implying that frequency is under control and not crashing upwards or downwards). However, since the ROCOF measurement can be noisy, this method was not chosen. The other alternative considered was to detect zero-crossings of the Δf_T signal. In practice the times at which the resets would occur due to any of the 3 methods is found to be about the same, so the most noise-immune method is chosen.

The algorithm details are shown below in Figure 6-1. In addition, some external code (Figure 6-2) can be used to widen the phase offset trip threshold Φ_{Trip}, during and immediately subsequent to serious (close-in) balanced and unbalanced faults which do not lead to LOM events. This code widens the trip setting from the normal setting to a much wider setting during serious balanced and unbalanced faults. When the fault condition is lifted, the trip threshold slides back at a constant rate to the normal setting, over a period of time. This allows for significant power system oscillations within the immediate post-fault period, and provides a solution to the conflicting demands of sensitivity and discrimination which are identified (but not solved) in Dysko (2006 & 2007).

It should be noted that some LOM relays block the trip signal entirely (Dysko (2007), Freitas (2005) & Vieira (2006c)) when such fault conditions are detected, whereas the solution proposed here is only a widening of the angle trip threshold. Thus, with the algorithm below, the LOM relay will still throw a trip if the local frequency diverges consistent with a genuine LOM event, potentially triggered by a fault condition. The aim is to allow a certain amount of rotor swinging and power system oscillation (which can exceed the standard trip setting of 20°) after a fault without throwing a LOM trip, but still to have a wider trip setting in place which will be exceeded if (for example) pole slip or an actual LOM event occurs.

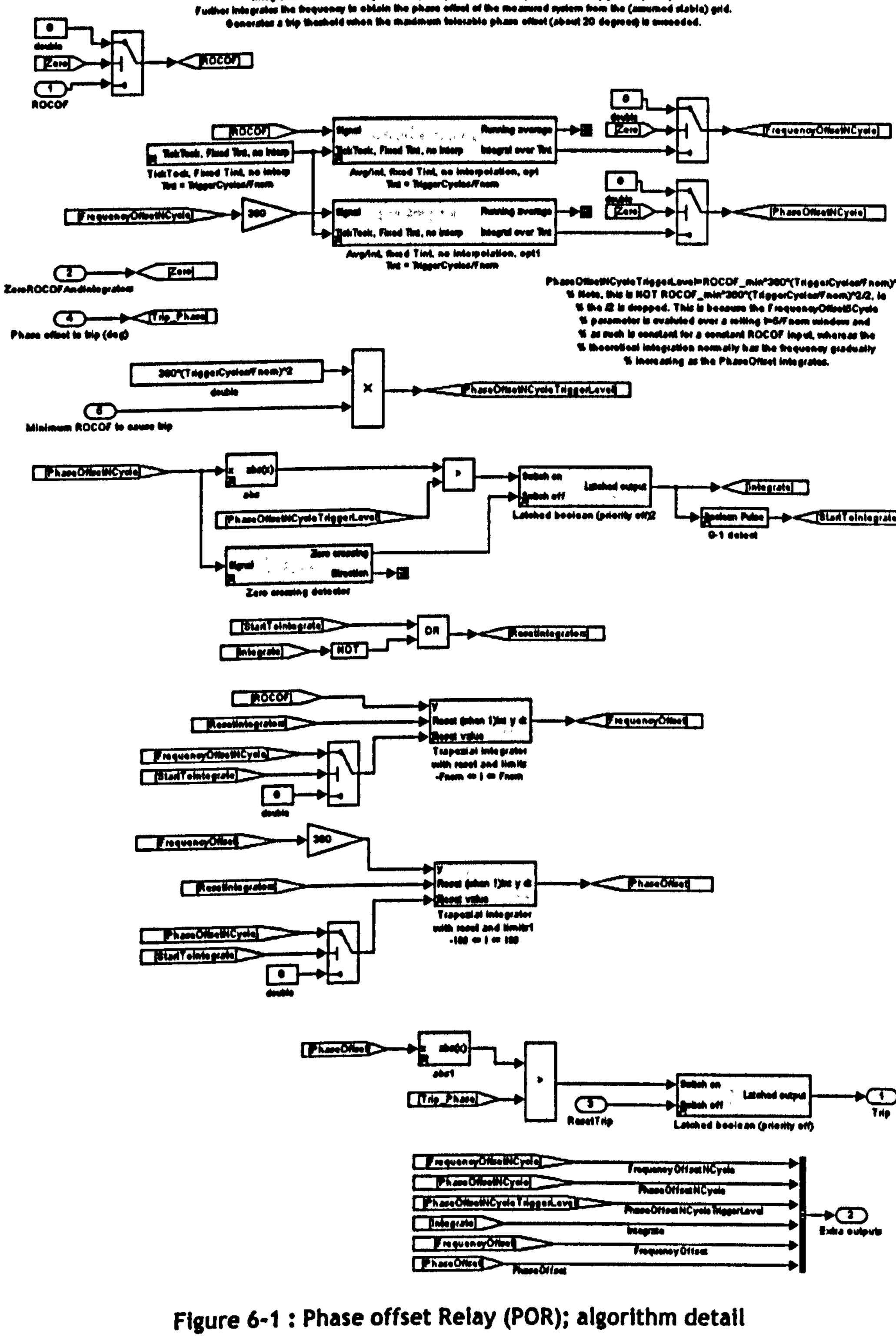


Figure 6-1 : Phase offset Relay (POR); algorithm detail

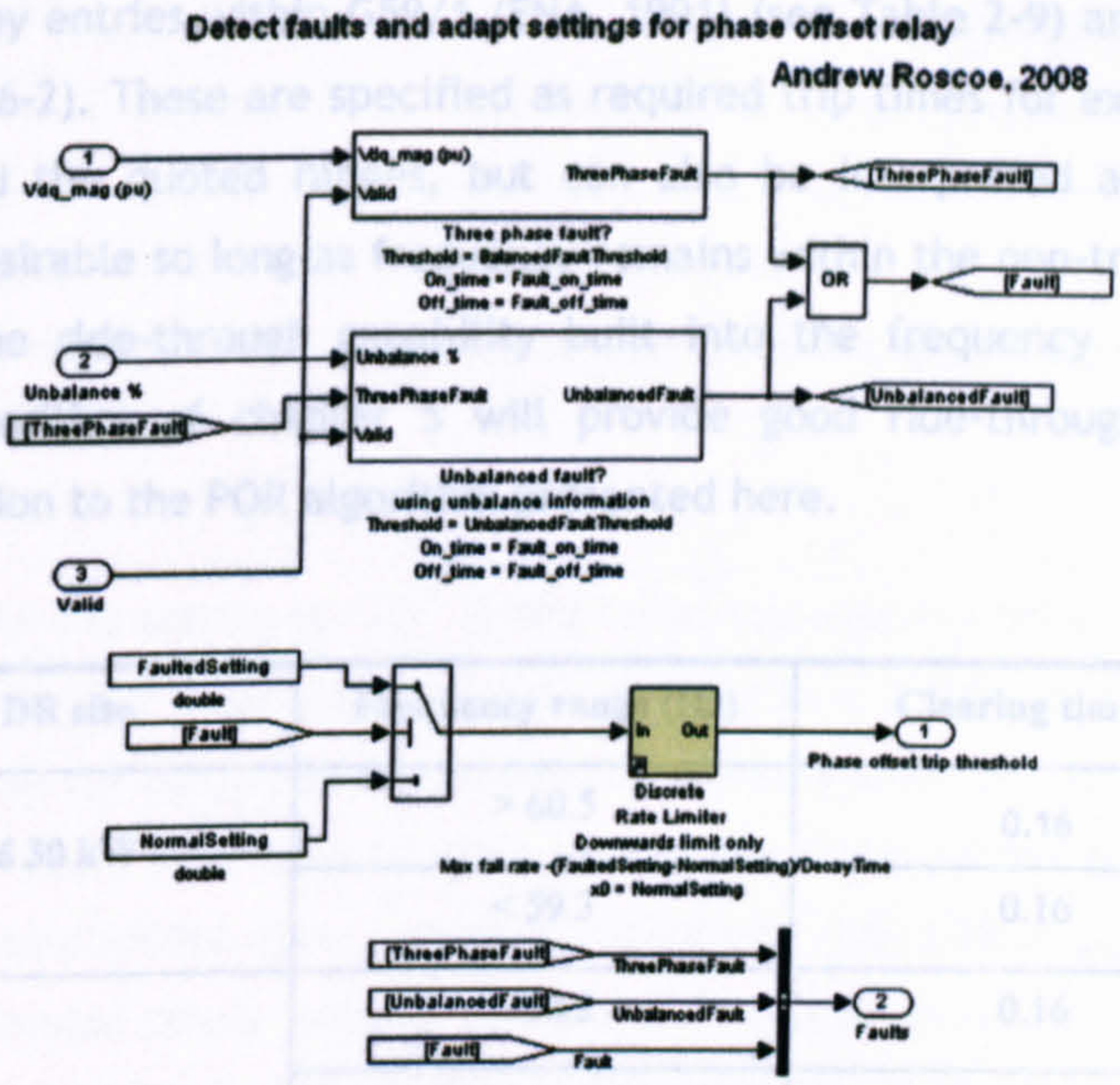


Figure 6-2 : POR; adaption of trip threshold immediately after faults

An example of the application of this algorithm is shown below in Figure 6-3, which is reproduced here from the simulation results of section 6.1.3.3 presented later. The blue line shows the apparent nodal phase variation due to power system oscillation immediately after fault clearance. The dashed red line shows how the trip setting can be automatically widened when the fault is detected (to 100° in this case), and then tapers back to the nominal setting (20° in this case) over a number of seconds after the fault is cleared.

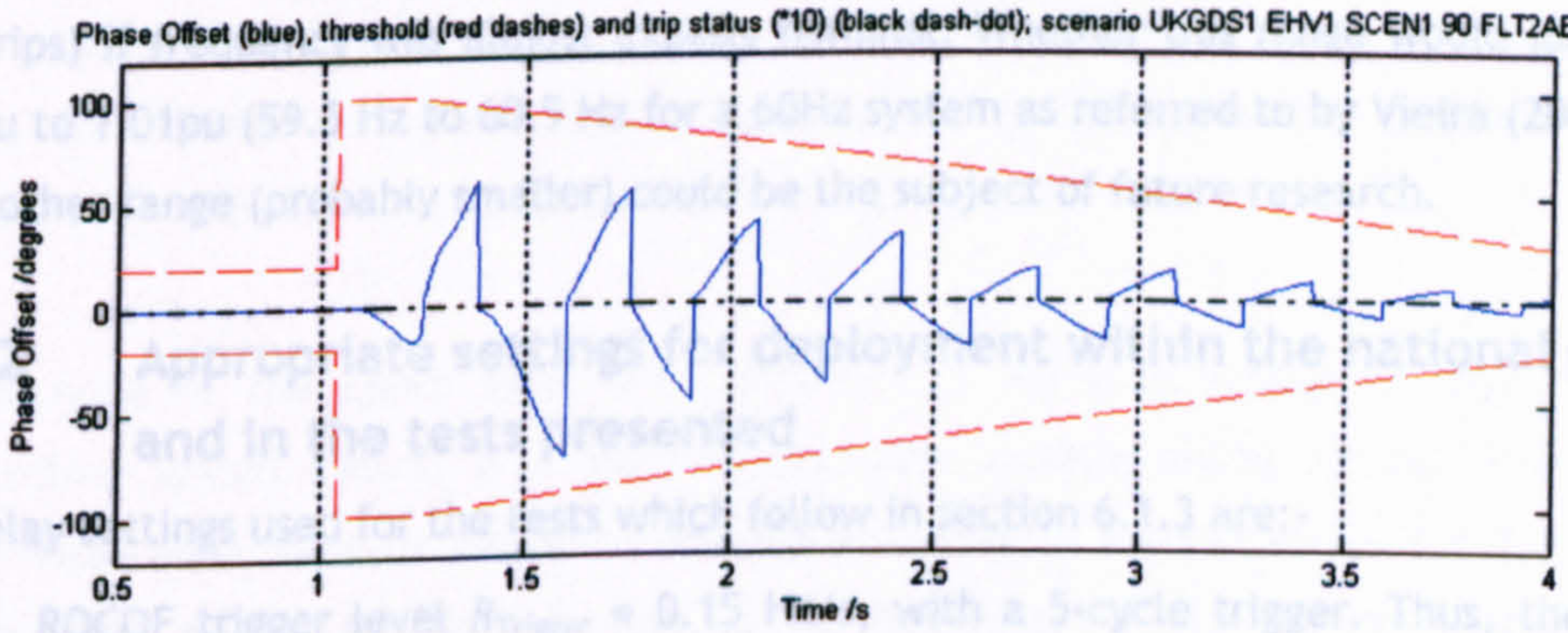


Figure 6-3 : POR; adaption of trip threshold immediately after faults

An additional algorithm to explicitly temporarily disable the trip setting could also be included in the future, although it is not included in this thesis. Veira (2006b), describes how there is a potential conflict between fault-ride through and loss-of-mains detection. The frequency range and time windows for fault ride-through during frequency excursions

can be deduced by entries within G59/1 (ENA, 1991) (see Table 2-9) and IEEE 1574 (IEEE, 2003) (see Table 6-2). These are specified as required trip times for excursions of system frequency beyond the quoted ranges, but can also be interpreted as times for which ride-through is desirable so long as frequency remains within the non-trip range. It should be noted that the ride-through capability built into the frequency (and thus ROCOF) measurement algorithm of chapter 5 will provide good ride-through capability even without modification to the POR algorithm presented here.

DR size	Frequency range (Hz)	Clearing time(s)
≤ 30 kW	> 60.5	0.16
	< 59.3	0.16
> 30 kW	> 60.5	0.16
	< {59.8 – 57.0} (adjustable set point)	
	< 57.0	0.16

Table 6-2 : Under/Overfrequency clearing times required under IEEE 1547 (2003)

Vieira (2006b) proposes that this conflict be resolved by modifying settings of existing ROCOF relay designs so that fault ride-through is generally achievable, as the ROCOF relays will tend not to trip due to normal loss-of-mains conditions until the local frequency goes outside the ranges of Table 6-2. A better, simpler alternative might be to insert a "no-trip" frequency range into the loss-of-mains relay which would reset the relay (and ban trips) if frequency was almost exactly nominal. Whether this range would ideally be 0.99pu to 1.01pu (59.5 Hz to 60.5 Hz for a 60Hz system as referred to by Vieira (2006b)) or some other range (probably smaller) could be the subject of future research.

6.1.2 Appropriate settings for deployment within the national grid, and in the tests presented

The relay settings used for the tests which follow in section 6.1.3 are:-

- ROCOF trigger level $R_{Trigger}$ = 0.15 Hz/s, with a 5-cycle trigger. Thus, the phase offset which must be exceeded over 5 cycles continuous rolling integration, is 0.54° by (6.4)
- A Phase Offset trip threshold of Φ_{Trip} = 20° during normal conditions. The normal setting of 20° is determined by the desire to trip off the generator before there is a risk of excessive re-synchronisation currents (see Table 6-1).
- A Phase Offset trip threshold of 100° during faults (balanced faults <0.8pu or

unbalanced faults with >10% unbalance), which is decreased back to the normal setting over 3 seconds after the fault is no longer perceived. This profile is determined by analysing the worst-case results obtained from the simulations of Dysko (2006). Note that time-hysteresis (via the parameters "On_time" and "Off_time" in Figure 6-2) is applied to stop transient spikes/dips causing erroneous detection of faults.

The widening of the trip setting to 100° during faults introduces a small (but finite) chance that a 100° out-of-phase might occur due to auto-reclose action. It should be borne in mind, however, that for this to occur, an entirely unlikely chain of events must occur. An upstream breaker would first have to open due to the fault current flowing. This would cause a genuine loss-of-mains event, coupled potentially with a fault persisting within the un-intentionally islanded power system. The protection schemes within the islanded power system would almost certainly trip quickly in this scenario. If the fault persisted locally, the trip would likely be under-voltage, over-frequency or unbalance. If the fault was cleared locally, then one of several possibilities exist:-

- That the voltage will remain nearly nominal or rise due to reactive power outputs of local generation exceeding that of local load demand. In this case the trip threshold for the POR would revert to 20° and the island should be detected by this relay, or by over-voltage or over/under-frequency alarms.
- That the voltage will fall. This might lead to the 100° trip threshold for the POR being retained for some time, as an under-voltage event would be perceived as a fault by the algorithm presented in this thesis. Note, however, that a sustained under-voltage for more than about 0.5 second will lead to the local power system (or at least generator) being tripped by the guidelines of G59 (Table 2-9). IEEE 1547 only allows 160ms before such a trip will occur (Table 2-8).

Only if none of the above local trips occurs, and the undervoltage persists, and an auto-reclose action occurs, is a 100° out-of-phase synchronisation risked. The reclose action would have to occur within 0.5 seconds of the initial fault inception for the relay to still be using a 100° trip setting, and this is much shorter than standard practice for re-close action (Areva T&D, 2007). The wider POR trip setting is gradually reduced over 3 seconds after a fault condition is cleared. During this time, should a reclose action occur, there is some risk of a less-than-100° but more than 20° out-of-phase re-synchronisation occurring. Assuming re-close happens no sooner than 2 seconds the relay setting will have diminished to 46°.

It should also be re-stressed that some existing ROCOF and vector-shift relays disable themselves completely during faults (Dysko (2007), Freitas (2005) & Vieira (2006c)), whereas the solution proposed here is only a widening of thresholds and is thus less likely to allow out-of-phase resynchronisation.

6.1.3 Analysis of relay performance

Within the report by Dysko (2006), simulations are performed which test both the sensitivity and discrimination of candidate LOM detection algorithms. Simulations were performed using the UKGDS EHV network 1 (SEDG, 2008), and the most challenging scenarios involved distributed generators of the synchronous variety. The schematic for UKGDS EHV network 1 is reproduced in Figure 6-4.

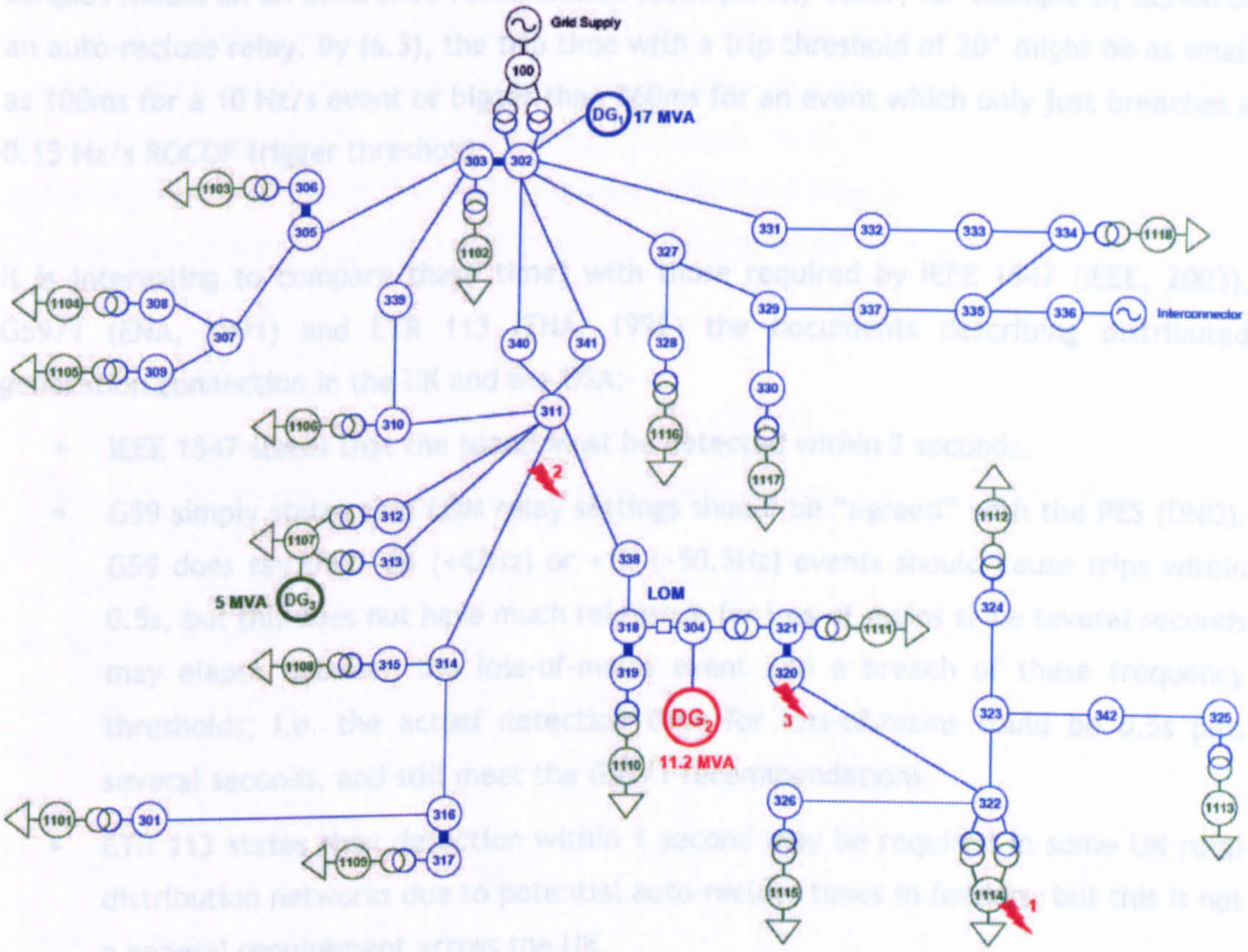


Figure 6-4 : UKGDS EHV network 1, reproduced from Dysko (2006)

From the results of Dysko (2006), the most challenging four scenarios are:-

- Sensitivity; detection of LOM with generator power output set 2.5% higher than local load power. (Scenario 3 in Dysko (2006)).
- Sensitivity; detection of LOM with generator power output set 2.5% lower than local load power. (Scenario 5 in Dysko (2006)).
- Discrimination; avoidance of trip during/following a 3-phase fault for 200ms at

position 2 (upstream of the distributed generator, within the distribution network). (Scenario 11 in Dysko (2006))

- Discrimination; avoidance of trip during/following a 3-phase fault for 200ms at position 3 (close downstream of the distributed generator). (Scenario 14 in Dysko (2006))

When detecting a genuine loss-of-mains (LOM) event, the detection will not be instant but will occur some time after the event begins. As described above, the purpose of the proposed POR is to create a trip signal when the local generator becomes out-of-phase with the parent network by an amount which would cause undesirable currents and torques should an un-controlled reconnection subsequently occur, for example by action of an auto-reclose relay. By (6.3), the trip time with a trip threshold of 20° might be as small as 100ms for a 10 Hz/s event or bigger than 860ms for an event which only just breaches a 0.15 Hz/s ROCOF trigger threshold.

It is interesting to compare these times with those required by IEEE 1547 (IEEE, 2003), G59/1 (ENA, 1991) and ETR 113 (ENA, 1995) the documents describing distributed generation connection in the UK and the USA:-

- IEEE 1547 states that the island must be detected within 2 seconds.
- G59 simply states that LOM relay settings should be "agreed" with the PES (DNO). G59 does say that -6% (<47Hz) or +1% (>50.5Hz) events should cause trips within 0.5s, but this does not have much relevance for loss-of-mains since several seconds may elapse between the loss-of-mains event and a breach of these frequency thresholds; i.e. the actual detection time for loss-of-mains could be 0.5s plus several seconds, and still meet the G59/1 recommendations.
- ETR 113 states than detection within 1 second may be required in some UK rural distribution networks due to potential auto-reclose times in feeders, but this is not a general requirement across the UK.

The approach taken by this thesis (and the relay proposed here) is that the tripping should take as long as it needs, so long as the risk to generator and distribution equipment is minimised, and that the trip does actually occur within the prescribed 1-2 seconds. The electrocution risk to personnel in touching parts of "presumed dead", but actually islanded, systems, is small within 1-2 seconds of any deliberate disconnection for maintenance purposes. For the tests performed in the subsequent sections, the 2-second detection requirement of IEEE 1547 (IEEE, 2003) has been used. The recommendation of ETR 113 is for a 1 second detection time. This can be achieved for the same non-detection

zone (NDZ) by tightening the trip threshold, or the trip threshold can be left the same but the NDZ will be larger.

To test the sensitivity and discrimination of the proposed new relay, the archived simulation results (3-phase voltage waveforms) for these 4 scenarios from Dysko (2006) are injected into simulations of the relay. The front-end of the relay is the frequency measurement algorithm described in chapter 5, and this feeds into the POR algorithm shown in Figure 6-1 and Figure 6-2. The sample rate for all the major algorithms is 500 Sa/s. An important characteristic of the relay is its rejection of interference due to noise and harmonics. However, the archived simulation results for the scenarios contains very pure waveforms of almost 0% THD (apart from during the fault inception/clearance), due to the limits of the simulation. To fully test the algorithms an intermediate pre-processing stage has been developed for this analysis. The archived simulation waveforms are analysed for frequency/magnitude/phase by the algorithms of chapters 3 and 5. Harmonic content (2^{nd} to 40^{th}) is then added to the original archived waveform with appropriate phases to synthesise the 2^{nd}-worst case microgrid waveform of section 2.7.2, Table 2-6 and Figure 2-5.

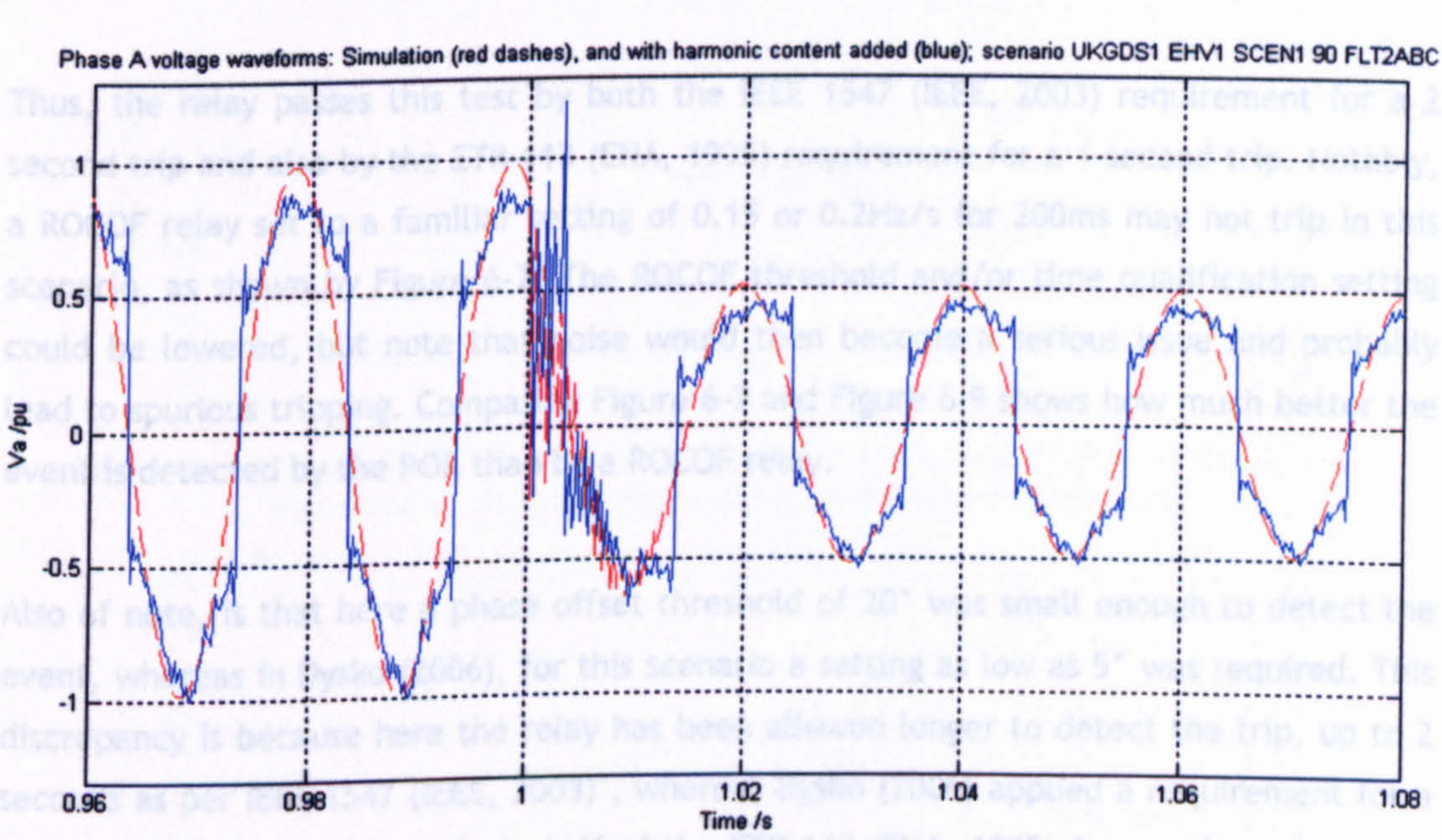


Figure 6-5 : Addition of harmonic content to the simulated waveforms; phase A voltage waveform from scenario simulation (red dashes), and with harmonic content added (blue solid line).

This produces a three-phase voltage waveform set with the same frequency, amplitudes

and phases as the simulation, but with a THD_V of approximately 28%. These corrupted signals are then used as the voltage inputs to the relay algorithms. An example of this process is shown in Figure 6-5, which is an excerpt from scenario 11 of Dysko (2006), with a 200ms fault beginning at t=1 second.

In addition to harmonic content, the analogue low-pass anti-aliasing filter, instrumentation noise and ADC quantisation/oversampling/downsampling are also simulated as described in chapter 3, section 3.4.

6.1.3.1 Sensitivity testing; detection of LOM with generator power output set 2.5% higher than local load power demand

This test uses scenario 3 from Dysko (2006). At t=1 second, the circuit breaker labelled "LOM" in Figure 6-4 is opened to create a loss-of-mains event. The relay should detect the LOM condition within 2 seconds. Figure 6-6 to Figure 6-9 show the results of this test. The relay successfully throws a trip signal, 910ms after the LOM event occurred. Triggering occurred 180ms after the LOM event, and at this time the 0.54 degrees which the 5-cycle phase offset trigger has accumulated is pre-loaded into the main phase offset integrator. This shows as a small initial brief rise in Figure 6-9 at t=1.18 seconds. The 20 degree threshold is exceeded 910ms after the LOM event, at t=1.91 seconds.

Thus, the relay passes this test by both the IEEE 1547 (IEEE, 2003) requirement for a 2 second trip and also by the ETR 113 (ENA, 1995) requirement for a 1-second trip. Notably, a ROCOF relay set to a familiar setting of 0.15 or 0.2Hz/s for 200ms may not trip in this scenario, as shown by Figure 6-7. The ROCOF threshold and/or time qualification setting could be lowered, but note that noise would then become a serious issue and probably lead to spurious tripping. Comparing Figure 6-7 and Figure 6-9 shows how much better the event is detected by the POR than by a ROCOF relay.

Also of note, is that here a phase offset threshold of 20° was small enough to detect the event, whereas in Dysko (2006), for this scenario a setting as low as 5° was required. This discrepancy is because here the relay has been allowed longer to detect the trip, up to 2 seconds as per IEEE 1547 (IEEE, 2003) , whereas Dysko (2006) applied a requirement for a trip within 500ms, conservatively half of the ETR 113 (ENA, 1995) 1-second requirement. For a trip within 500ms, Figure 6-9 indeed shows that a setting of about 5° would be required. This shows that the POR algorithms presented here are equal in sensitivity to the algorithm used in Dysko (2006).

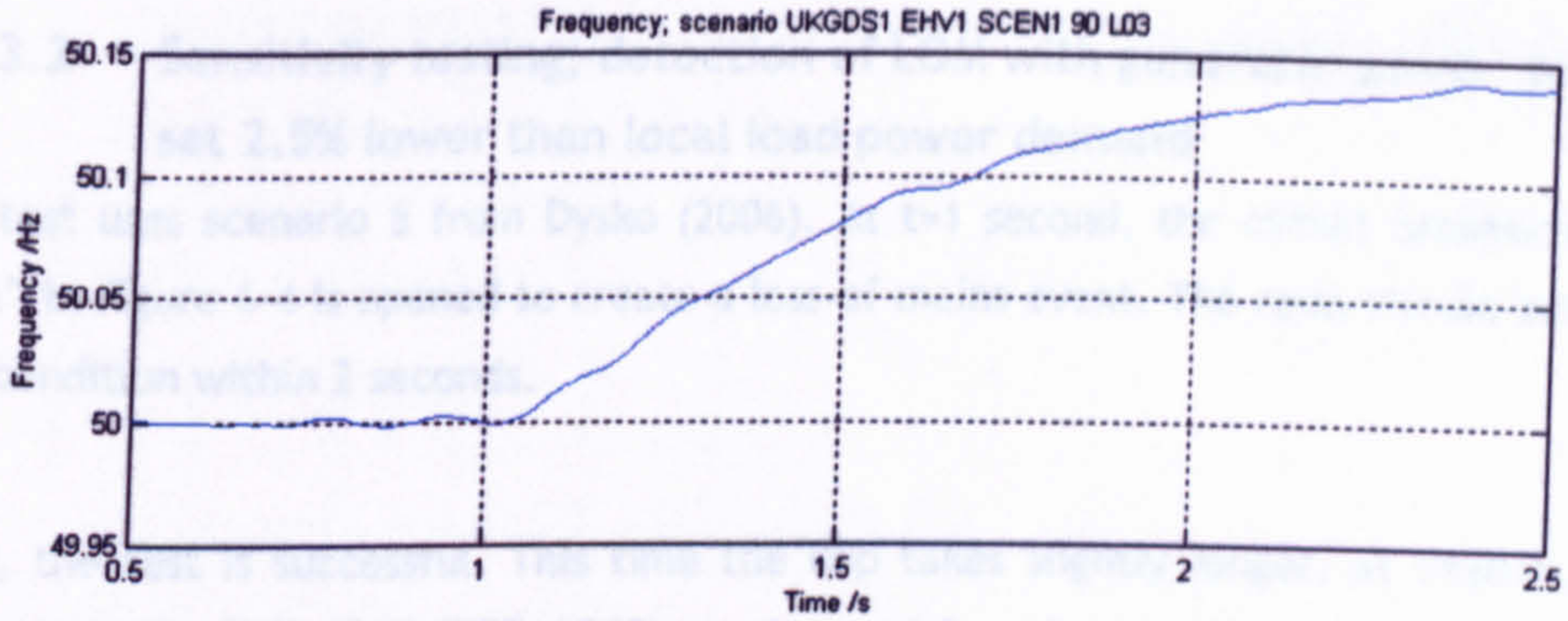


Figure 6-6 : Scenario 3; +2.5% power; Frequency excursion

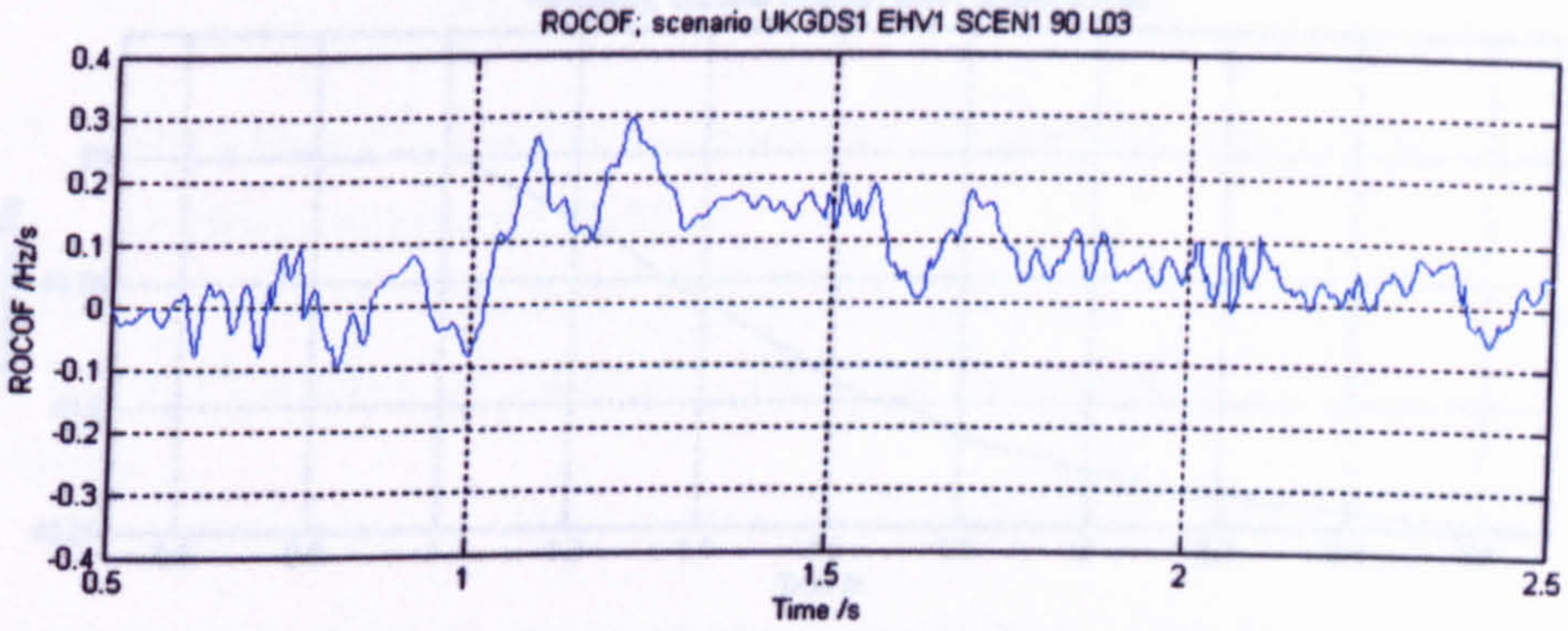


Figure 6-7 : Scenario 3; +2.5% power; ROCOF

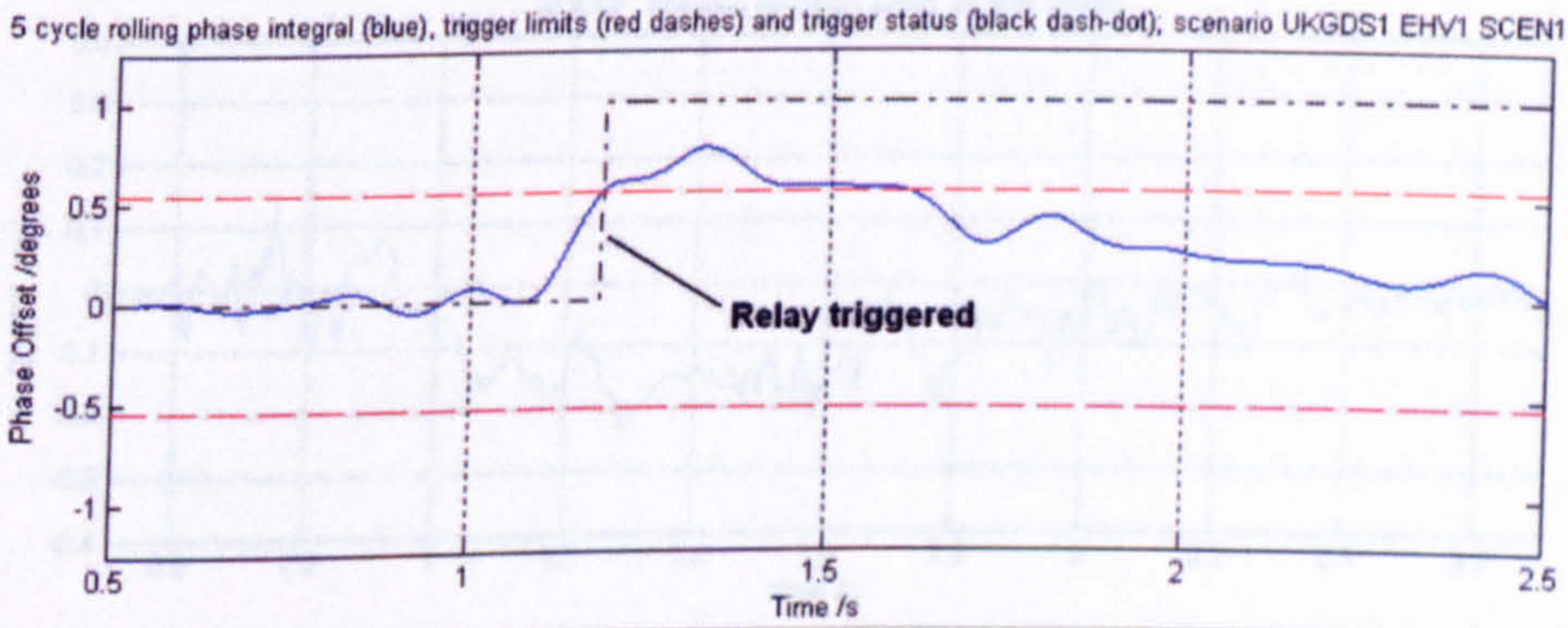


Figure 6-8 : Scenario 3; +2.5% power; Rolling 5-cycle phase integral (blue, with threshold in dashed red) and triggering (black dash-dot)

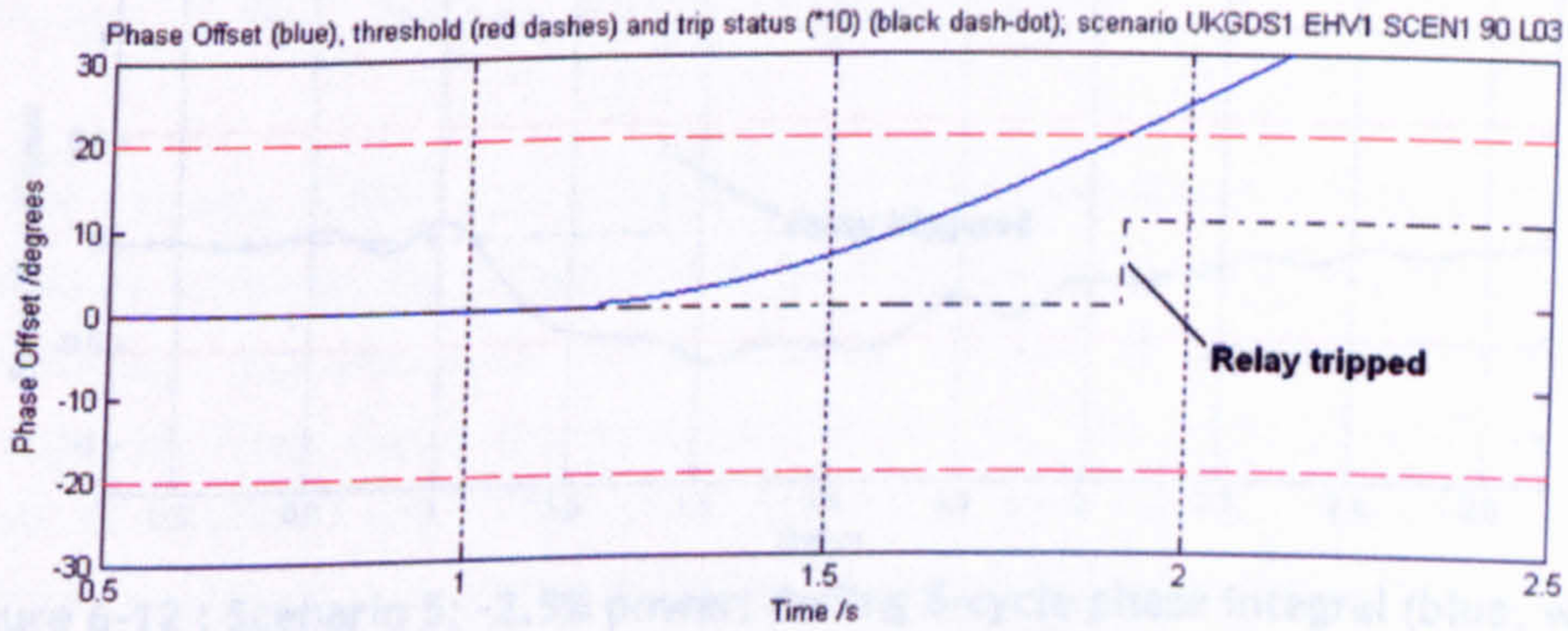


Figure 6-9 : Scenario 3; +2.5% power; Phase Offset (blue, with threshold in dashed red) and Trip signal (black dash-dot)

6.1.3.2 Sensitivity testing; detection of LOM with generator power output set 2.5% lower than local load power demand

This test uses scenario 5 from Dysko (2006). At t=1 second, the circuit breaker labelled "LOM" in Figure 6-4 is opened to create a loss-of-mains event. The relay should detect the LOM condition within 2 seconds.

Again, the test is successful. This time the trip takes slightly longer, at 1120ms, which again meets the IEEE 1547 (IEEE, 2003) requirement for a 2 second trip.

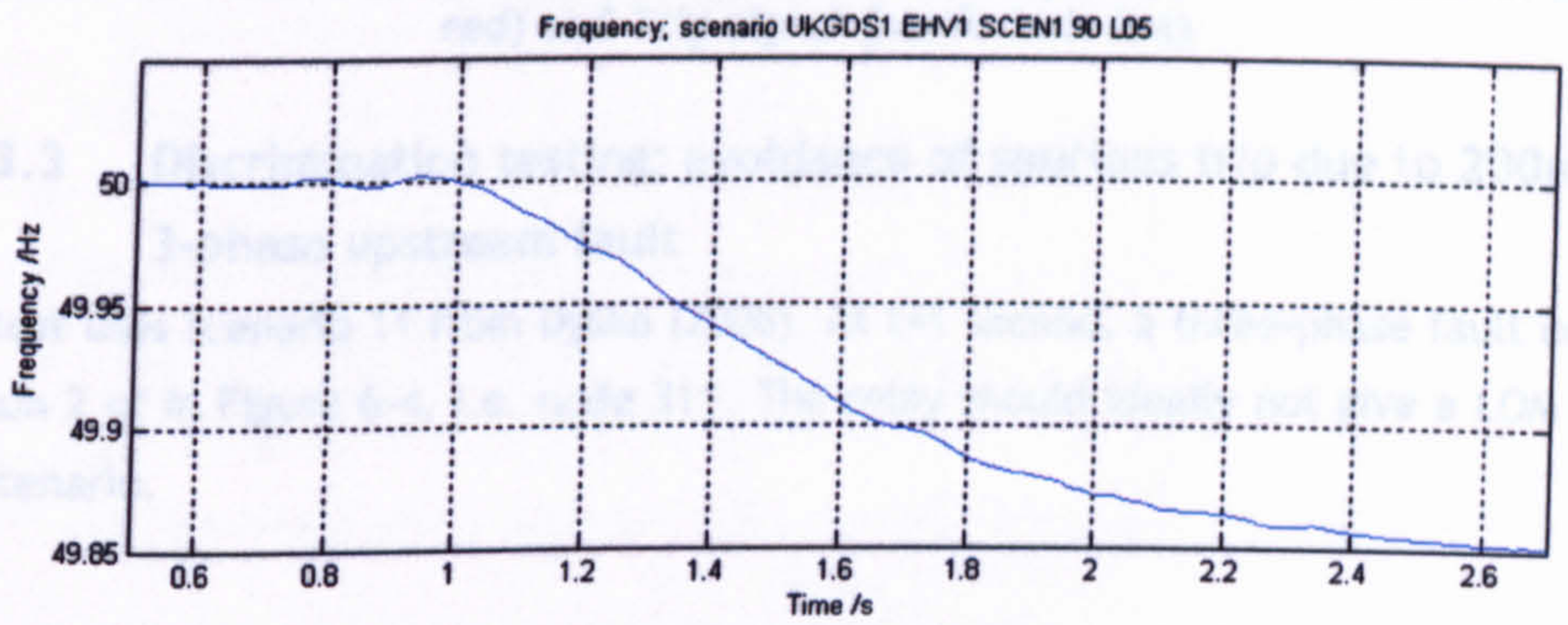


Figure 6-10 : Scenario 5; -2.5% power; Frequency excursion

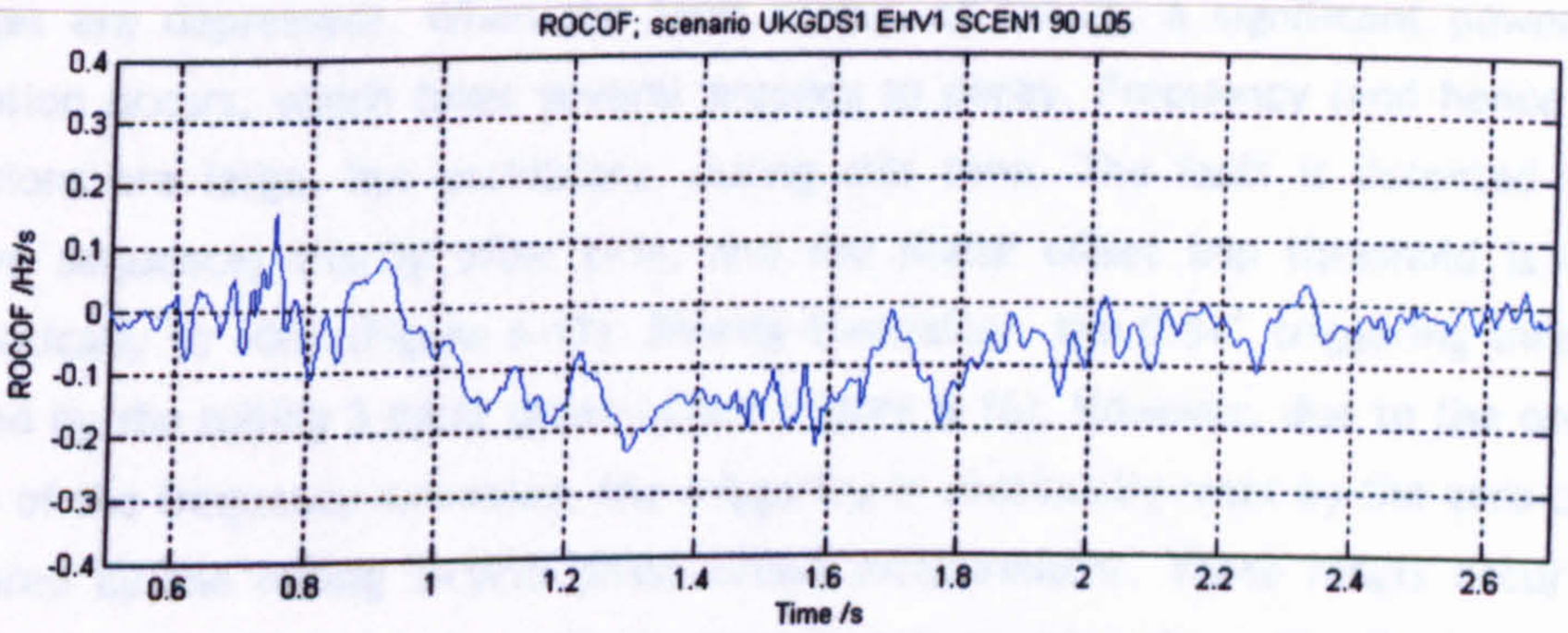


Figure 6-11 : Scenario 5; -2.5% power; ROCOF

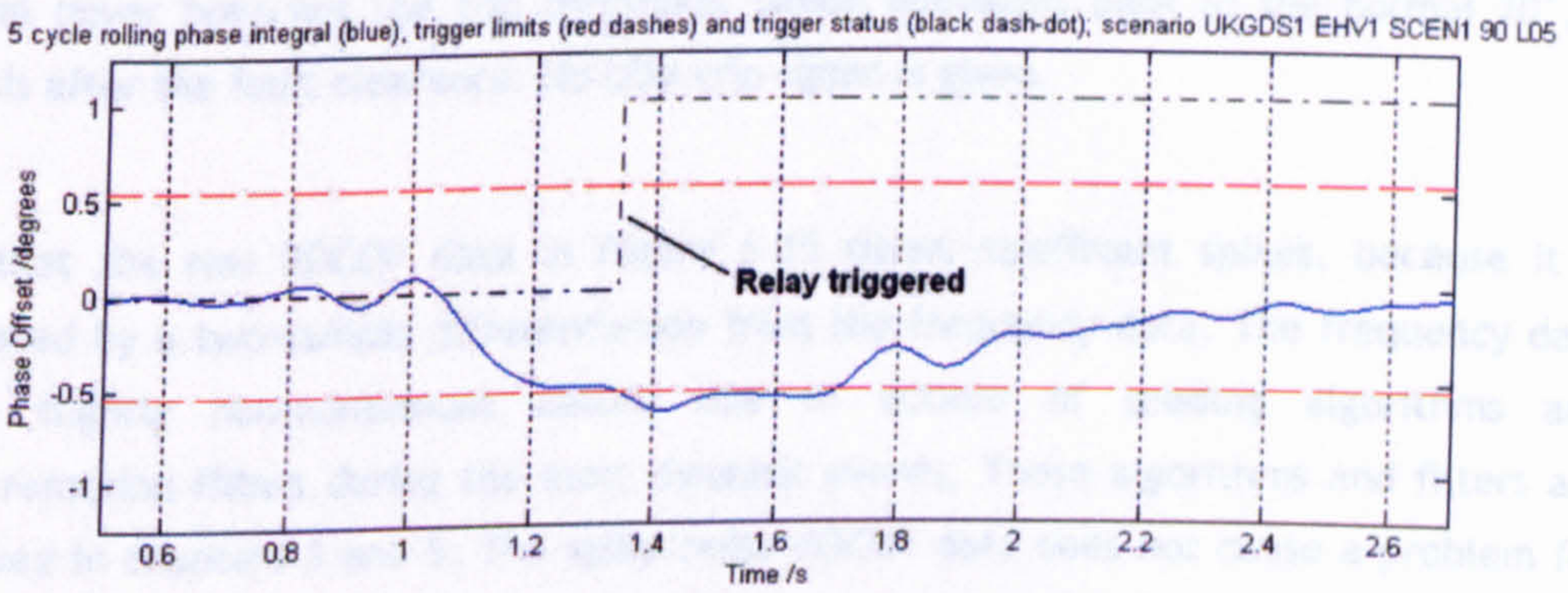


Figure 6-12 : Scenario 5; -2.5% power; Rolling 5-cycle phase integral (blue, with threshold in dashed red) and triggering (black dash-dot)

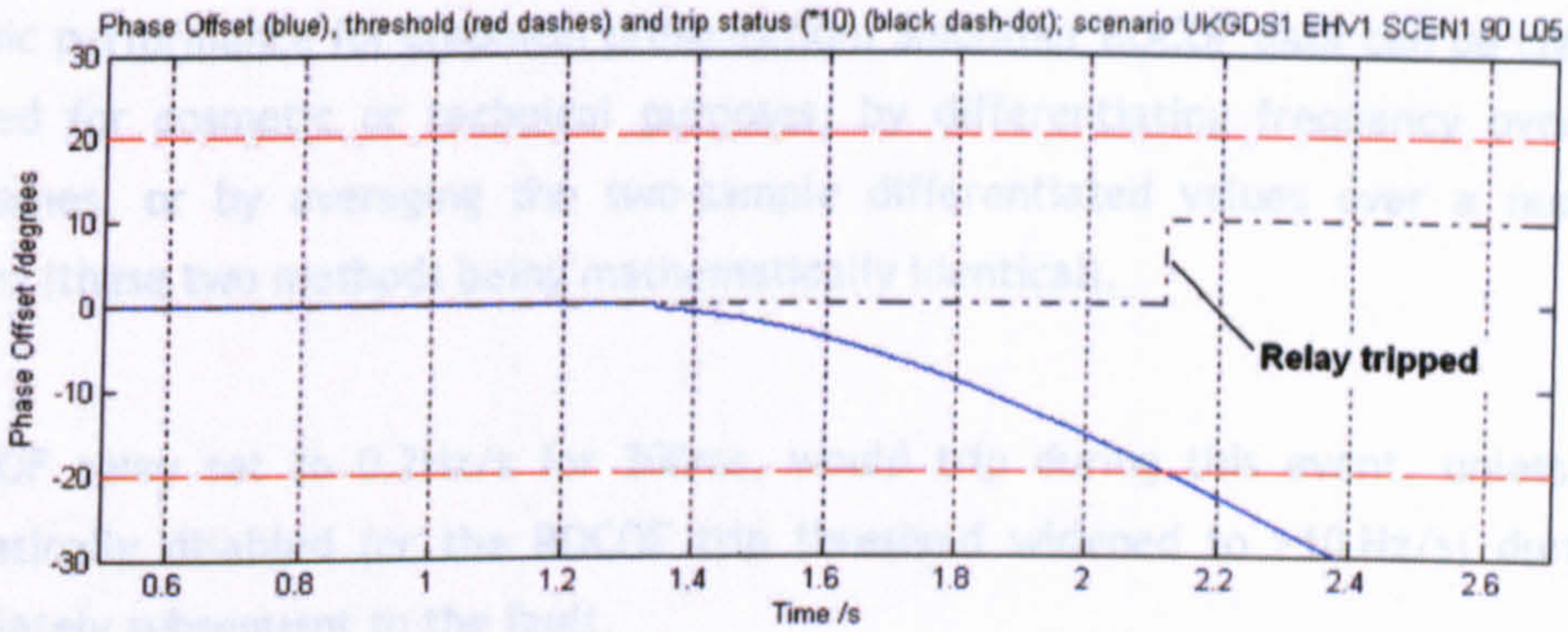


Figure 6-13 : Scenario 5; -2.5% power; Phase Offset (blue, with threshold in dashed red) and Trip signal (black dash-dot)

6.1.3.3 Discrimination testing; avoidance of spurious trip due to 200ms 3-phase upstream fault

This test uses scenario 11 from Dysko (2006). At t=1 second, a three-phase fault occurs at position 2 of in Figure 6-4, i.e. node 311. The relay should ideally not give a LOM trip for this scenario.

The fault begins at t=1s. During the fault itself, the frequency variation is small, (although voltages are depressed). When the fault clears, at t=1.2s, a significant power system oscillation occurs, which takes several seconds to decay. Frequency (and hence ROCOF) excursions are large, but oscillatory, during this time. The fault is detected (via low positive sequence) shortly after t=1s, and the phase offset trip threshold is widened automatically to 100° (Figure 6-17). Shortly thereafter, the 0.54° triggering threshold is violated by the rolling 5-cycle phase offset (Figure 6-16). However, due to the oscillatory nature of the frequency excursion, the triggering is continually reset by the zero crossings registered by the rolling 5-cycle phase offset measurement. These resets occur shortly after the peaks (positive and negative) of the frequency excursions. The final phase offset integral never breaches the trip threshold, which decreases back to the normal 20°, 3 seconds after the fault clearance. No LOM trip signal is given.

Note that the raw ROCOF data in Figure 6-15 shows significant spikes, because it is calculated by a two-sample differentiation from the frequency data. The frequency data has a slightly non-continuous nature due to actions of seeding algorithms and ripple-rejection filters during the most dynamic events. These algorithms and filters are described in chapters 3 and 5. The spiky/noisy ROCOF data does not cause a problem for the POR proposed here, which is specifically designed to cope with this kind of effect. The use of ROCOF data in this relay is merely as an intermediate stage to remove DC offset from the "Frequency Offset" result, and at the same time to provide an indication of

dynamic performance for graphical presentation. Smoother ROCOF data can be created, if required for cosmetic or technical purposes, by differentiating frequency over longer timeframes, or by averaging the two-sample differentiated values over a number of samples (these two methods being mathematically identical).

A ROCOF relay set to 0.2Hz/s for 200ms, would trip during this event, unless it was automatically disabled (or the ROCOF trip threshold widened to >10 Hz/s) during and immediately subsequent to the fault.

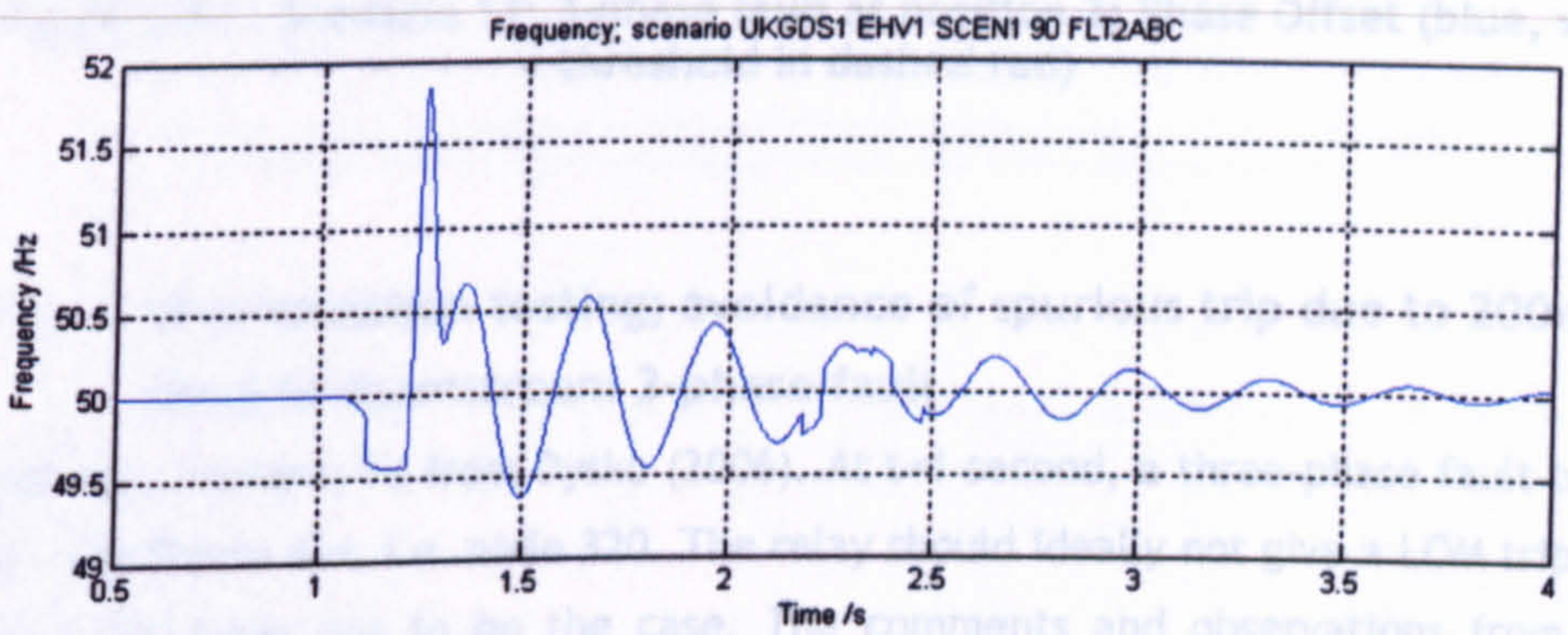


Figure 6-14 : Scenario 11; 3-phase fault at position 2; Frequency excursion

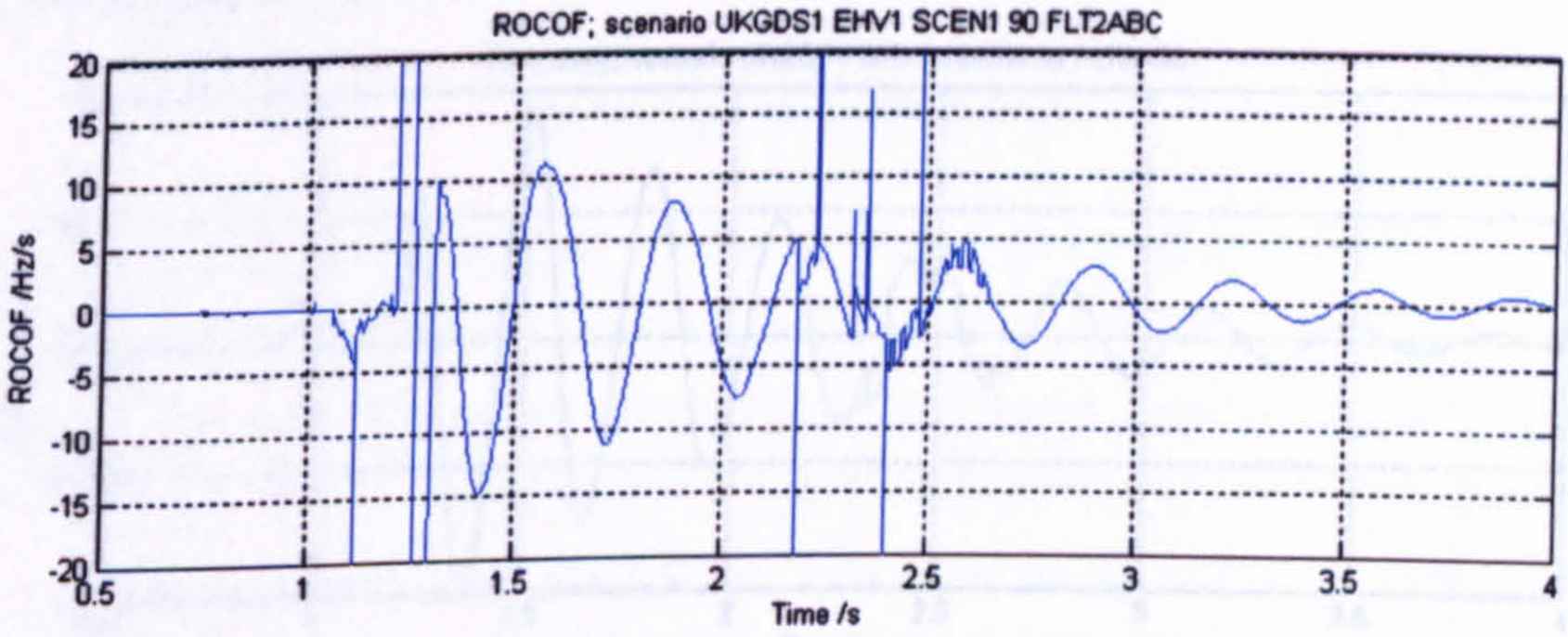


Figure 6-15 : Scenario 11; 3-phase fault at position 2; ROCOF

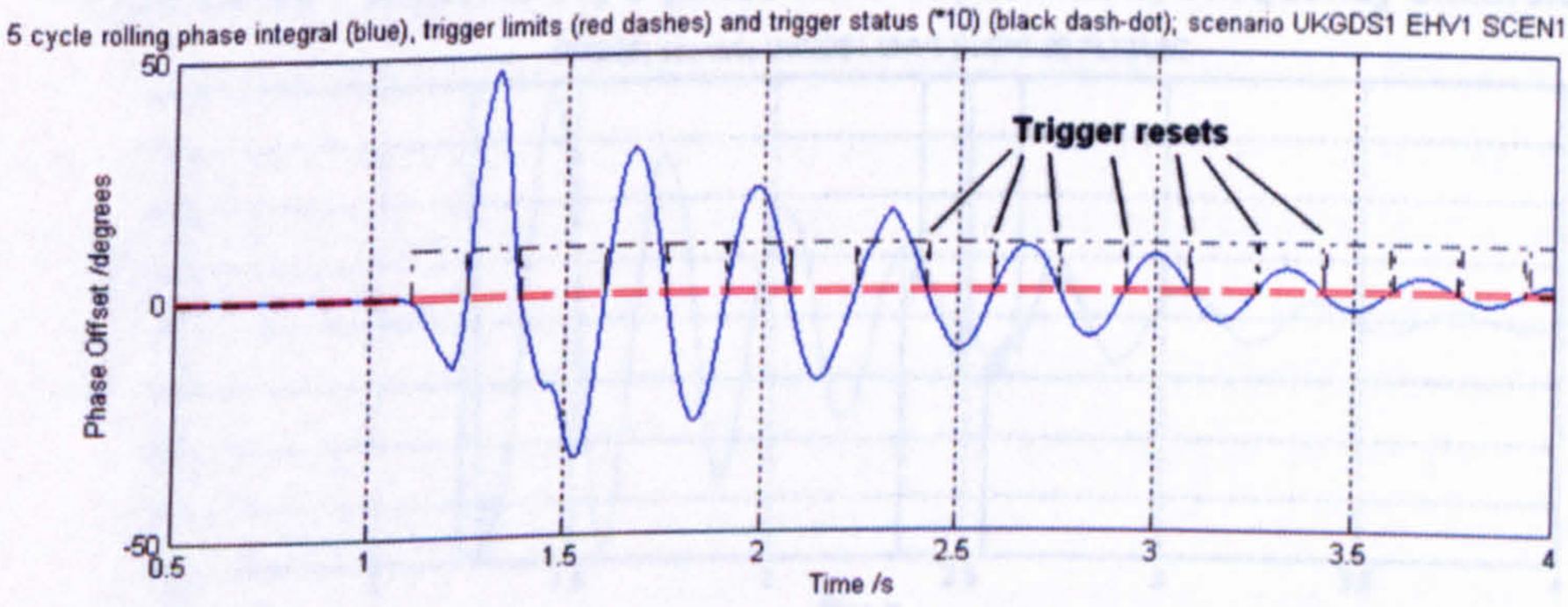


Figure 6-16 : Scenario 11; 3-phase fault at position 2; Rolling 5-cycle phase integral (blue, with threshold in dashed red) and triggering (black dash-dot)

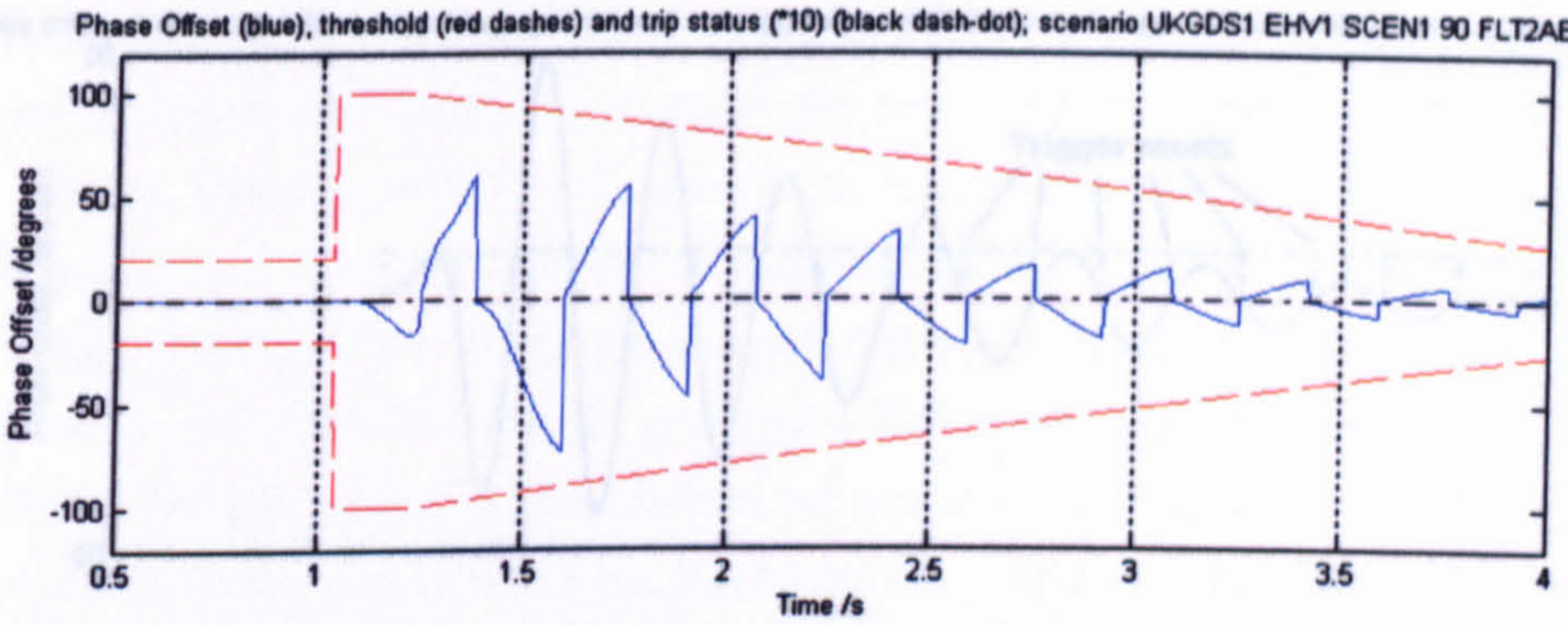


Figure 6-17 : Scenario 11; 3-phase fault at position 2; Phase Offset (blue, with threshold in dashed red)

6.1.3.4 Discrimination testing; avoidance of spurious trip due to 200ms close-in downstream 3-phase fault

This test uses scenario 14 from Dysko (2006). At t=1 second, a three-phase fault occurs at position 3 in Figure 6-4, i.e. node 320. The relay should ideally not give a LOM trip for this scenario. This turns out to be the case. The comments and observations from section 6.1.3.3 all apply equally to this scenario.

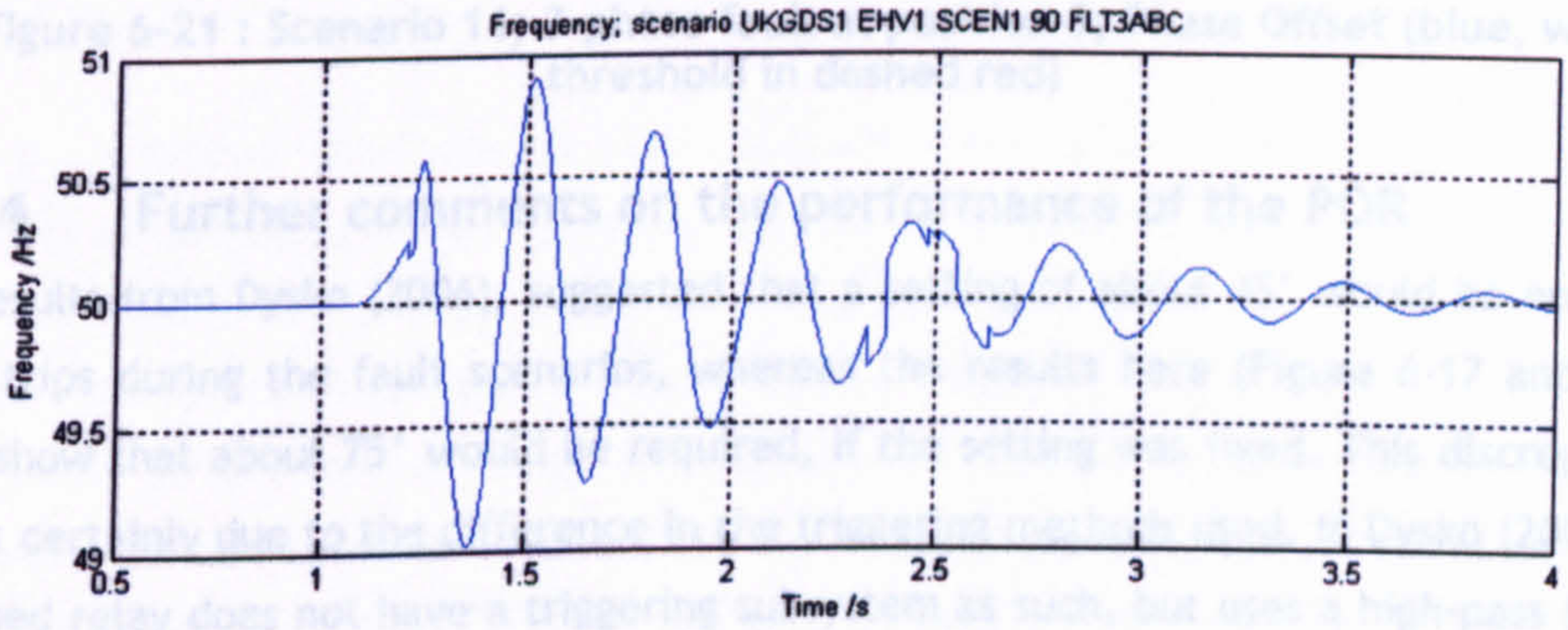


Figure 6-18 : Scenario 11; 3-phase fault at position 3; Frequency excursion

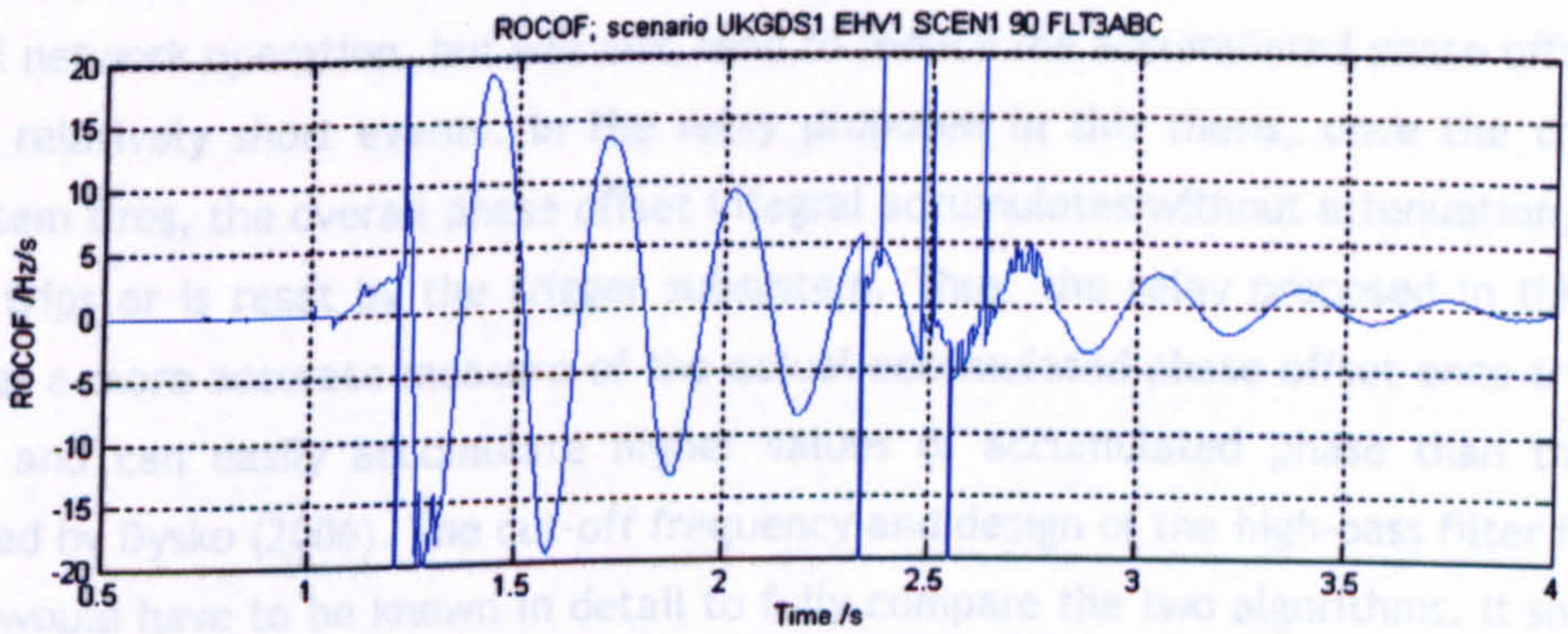


Figure 6-19 : Scenario 11; 3-phase fault at position 3; ROCOF

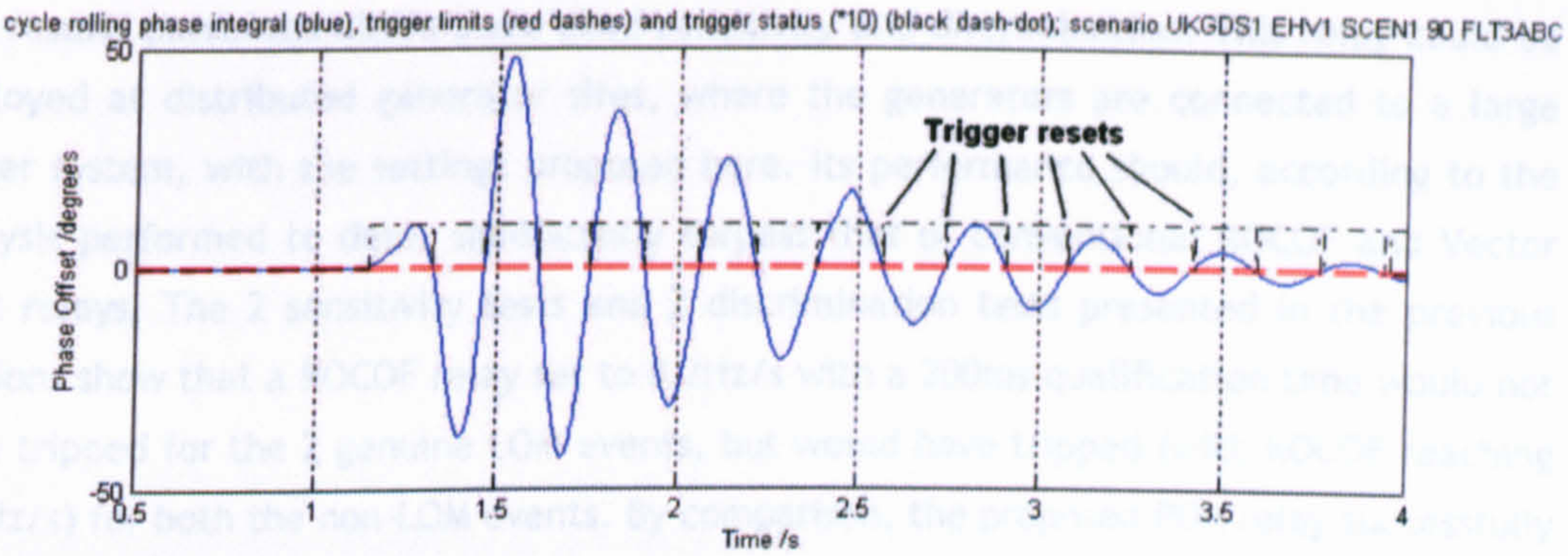


Figure 6-20 : Scenario 11; 3-phase fault at position 3; Rolling 5-cycle phase integral (blue, with threshold in dashed red) and triggering (black dash-dot)

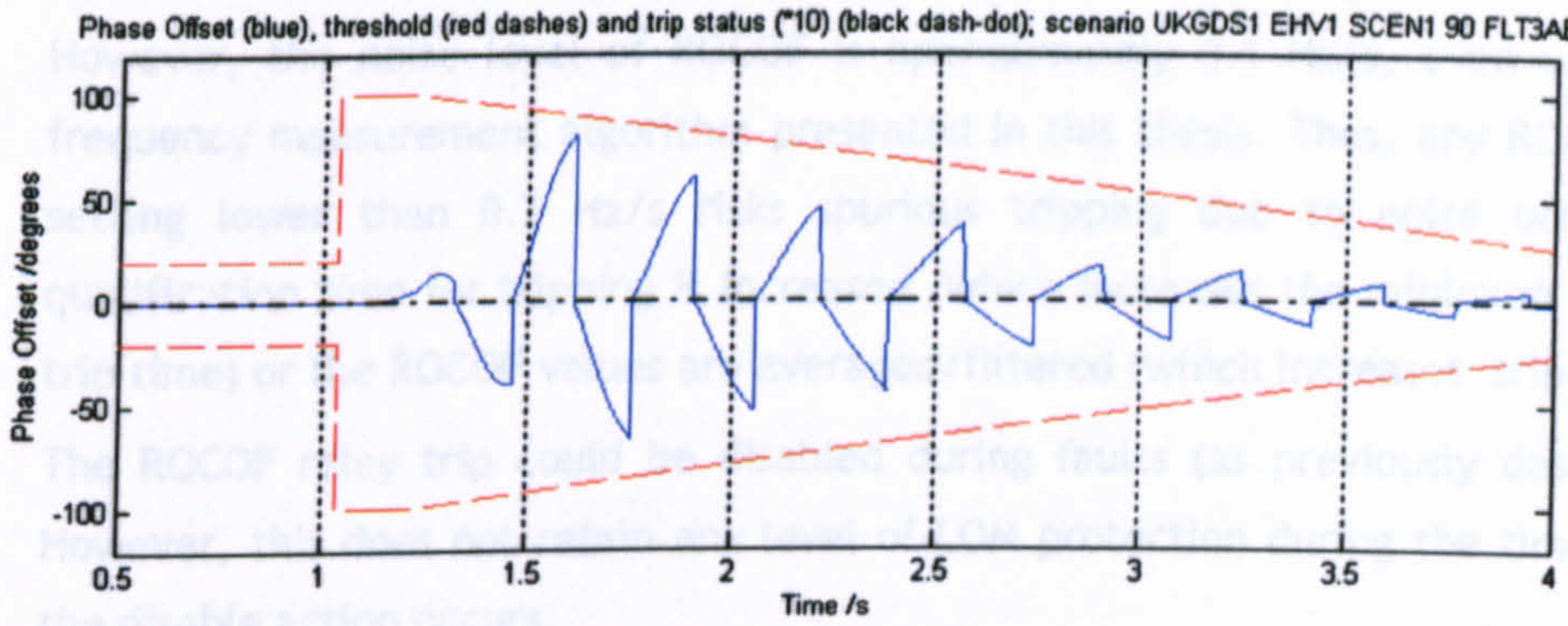


Figure 6-21 : Scenario 11; 3-phase fault at position 3; Phase Offset (blue, with threshold in dashed red)

6.1.4 Further comments on the performance of the POR

The results from Dysko (2006), suggested that a setting of about 45° would be enough to avoid trips during the fault scenarios, whereas the results here (Figure 6-17 and Figure 6-21) show that about 75° would be required, if the setting was fixed. This discrepancy is almost certainly due to the difference in the triggering methods used. In Dysko (2006), the proposed relay does not have a triggering subsystem as such, but uses a high-pass filter to continually attenuate the accumulated phase offsets. This stops the relay tripping during normal network operation, but will also tend to reduce the accumulated phase offset even during relatively short events. In the relay proposed in this thesis, once the triggering subsystem fires, the overall phase offset integral accumulates without attenuation, until it either trips or is reset by the trigger subsystem. Thus, the relay proposed in this thesis provides a more accurate measure of the actual accumulated phase offset once triggering starts, and can easily accumulate higher values of accumulated phase than the relay proposed by Dysko (2006). The cut-off frequency and design of the high-pass filter in Dysko (2006) would have to be known in detail to fully compare the two algorithms. It should be noted that the thresholds for the triggering subsystem (and for the Dysko relay the design of the high-pass filter), are equally important as the trip thresholds, in terms of setting the relay for optimum sensitivity and discrimination.

The results presented above show good sensitivity and discrimination. This relay could be deployed at distributed generator sites, where the generators are connected to a large power system, with the settings proposed here. Its performance should, according to the analysis performed to date, significantly surpass that of conventional ROCOF and Vector Shift relays. The 2 sensitivity tests and 2 discrimination tests presented in the previous sections show that a ROCOF relay set to 0.2Hz/s with a 200ms qualification time would not have tripped for the 2 genuine LOM events, but would have tripped (with ROCOF reaching >10Hz/s) for both the non-LOM events. By comparison, the proposed POR relay successfully detects and discriminates the relevant scenarios. It might be argued that:-

1. The ROCOF relay settings could be tightened, to enable greater sensitivity. However, the noise level of ROCOF is approximately 0.1 Hz/s, even using the frequency measurement algorithm presented in this thesis. Thus, any ROCOF trip setting lower than 0.2 Hz/s risks spurious tripping due to noise unless the qualification time for tripping is increased (which increases the minimum possible trip time) or the ROCOF values are averaged/filtered (which increases trip time).
2. The ROCOF relay trip could be disabled during faults (as previously described). However, this does not retain any level of LOM protection during the time which the disable action occurs.

However, a further challenge for LOM detection remains to be addressed. The natural frequency disturbances in microgrids are much larger than those of a power system such as the UK national grid. The expected rates of change of frequency were predicted in section 2.1, and summarised in Table 2-1. Notably, within a 100kVA microgrid, even switching on a single kettle at 3kW can be enough to cause a brief 0.4 Hz/s frequency deviation. Larger load steps can cause larger disturbances. The problem scenario is thus that a very small power system, say of 10kVA rating, is "grid-connected" to a parent network of only 100kVA. In this case, the 10kVA microgrid (generator plus its local load) will measure ROCOF events regularly greater than 0.4 Hz/s. Governor actions will act to mitigate these deviations within the constrains of the control system dynamics, but with a trigger setting of 0.15Hz/s, there is a significant risk that the relay (with the settings used above) would

1) trigger, as ROCOF surpasses 0.15Hz/s

2) trip, if the frequency did not stabilise to a new level quickly enough to avoid a 20° phase offset being accumulated. Note that if frequency stabilises, it causes ROCOF to become about 0 (with noise), which causes the 5-cycle phase offset rolling integral to also become about 0 (with noise), and hence cross zero to reset the trigger.

A potential solution to avoiding such spurious trips when the parent network is of limited size, without employing communication systems or active LOM-detection techniques, is to widen the ROCOF trigger to avoid spurious trips. This would be a better approach than widening the trip threshold of 20°, since this 20° figure relates to the worst uncontrolled synchronisation angle which would ideally be risked.

It might be possible to design an algorithm which continually monitors system frequency deviations and autonomously sets a suitable ROCOF trigger threshold. Or, knowledge of network configuration might be used to deliberately adjust the relay settings when the parent network itself joins (or leaves) a much larger power system which brings greater frequency stability. A combination of the above two possibilities might also be realised.

The impact of widening the trigger threshold would be two-fold:

- Larger system frequency deviations, due to larger load switching and generator despatching (as per-unit proportions of the total system inertia), could be accommodated without the relay triggering and risking a spurious trip.
- During genuine LOM events where active and reactive load powers are well matched to the generator(s) output power set-points, the risk of non-detection is raised. This can be seen by imagining a ROCOF trigger threshold of 0.4Hz/s in section 6.1.3.1 or 6.1.3.2 above. In those scenarios, with a ROCOF trigger threshold of 0.15Hz/s (converted to a 5-cycle phase offset of 0.54°), the relay only just triggered. If the trigger threshold was doubled, the relay would neither trigger nor trip.

Within a microgrid context, therefore, the relay may need to be de-sensitised to avoid constant spurious tripping. One option to reduce the risk of subsequent non-detection of genuine LOM events is to manage the generators and/or network topology such that there is a very low risk of any generator P/Q power output set-point being a close match for any local load P/Q combination. Only a small imbalance in either P (real power) or Q (reactive power), and not necessarily both, is required to cause imbalance in an unintentionally islanded power system which will allow detection of the LOM event. This effect, along with a novel algorithm to ensure such an imbalance, is presented in section 6.2 to follow.

6.1.5 Summary and further work opportunities arising from this section

- A novel algorithm for a Phase Offset Relay (POR), for the detection of loss-of-mains, has been designed and coded. The algorithm is fully robust for real-time

implementation and is significantly more immune to noise and interference effects than a ROCOF relay, due to its use of Phase as the key measurement parameter. Phase is calculated as the double-integral of ROCOF which is equivalent to two stages of averaging, thus giving good rejection of noise on a frequency/ROCOF measurement.

- The POR allows for variable trip times, graded so that the important parameter is the potential phase angle relative to the parent network which would result due to an uncontrolled out-of-phase resynchronisation.
- Both the triggering algorithm setting and the trip setting need to be appropriately set to provide the best balance of discrimination and sensitivity.
- Good discrimination can be provided for ride-through of faults by using a small algorithm which temporarily allows larger phase offset trip settings during faults (which can be detected using positive sequence and unbalance thresholds applied to 3-phase voltage measurements) The trip settings can be tapered from the wider settings back to normal (20°) over 3 seconds after a fault is cleared, to allow for network oscillations to die away.
- The network simulation results suggest that using a trigger setting of 0.15-0.2 Hz/s and a normal trip setting of 20° allows detection of loss-of-mains in less than 1.5 seconds; assuming that the unintentional power islands are more than 2.5% mismatched in terms of active power. These settings could be used sensibly for grid-connected systems attached to the UK national grid.
- When the parent network is smaller, the trigger setting of 0.2 Hz/s may be too small, allowing numerous spurious trips. In this case the trigger setting must be widened, which de-sensitises the relay. This is explored further in section 6.2.

Further work opportunities include:-

- Design and test an algorithm to automatically adapt the ROCOF triggering level $R_{Trigger}$ for the proposed relay, for different scenarios/size of parent network. The algorithm would need to monitor frequency deviations due to "normal" network behaviour.

6.2 Application to loss-of-mains detection; discrimination testing using real-world transient data

The most obvious case of spurious loss-of-mains detections is due to genuine power system frequency disturbances which cause ROCOF (rate of change of frequency) to exceed trigger or trip settings within loss-of-mains relays. Likely values of ROCOF within different sized microgrids, from 100kVA to 60GVA, were analysed in section 2.1 and Table 2-1. For distributed generators connected to a distribution system in the UK, the size of the entire

power system is approximately 60GVA. According to Table 2-1, ROCOF should be limited to about 0.01 Hz/s for normal operation, and about 0.1-0.2 Hz/s for worst-case power station tripping leading to a ~2GW step change in generation.

To verify that the estimated value of 0.01 Hz/s from Table 2-1 is correct, a data logging application was created. This executes on the ADI RTS (Real-Time-Station) (ADI, 2008) with a sample rate of 2000 Sa/s. The sampled values are three-phase voltages at the LV side of an 11kV/433V transformer, shown on in Figure 6-28, which is part of the laboratory infrastructure. The application software processes the data in real-time using the algorithms produced during this thesis. Parameters monitored include frequency, ROCOF, fundamental voltage amplitudes, unbalance, THD and possible spurious loss-of-mains detection events. Data is logged every second, with the most interesting event causing additional capture at the full 2000 Sa/s frame rate for a 60 second window (30 seconds pre-trigger plus 30 seconds post-trigger data).

Over 825 hours of monitoring (summer and winter), were logged during 2007. All the genuine ROCOF events captured were restricted to ROCOF rates of less than 0.025 Hz/s, on a second-by-second basis. This suggests that the analysis of Table 2-1 is correct (to within a power of 2). The term "genuine" is here used to restrict the analysis to events which appear as genuine frequency slides which occur over several seconds.

In contrast, 7 events were logged which transiently show much larger rates of change of frequency. These only show up on the 1-second logged data as spikes. The thresholds for detecting "interesting" events within the logging application can be set to very low levels, much tighter than a normal protection relay would be set. In this way, such interesting deviations from static conditions can be logged at 2000 Sa/s for further analysis.

Interestingly, of the 825 hours data which was logged, 5 of the 7 transient events which were not "genuine" rates-of-change of frequency, but were transient events, occurred within a single 6 hour period. The remaining 2 transient events occurred 4 days later within a 15 minute period. Analysis of the events would suggest that they are distant unbalanced faults, switching or tap-changing events. The faults do not cause voltage depressions sufficient to cause widening of the POR trip settings as described in section 6.1.

The worst of the 7 transient events contains only a small (0.02pu) voltage step. The transient would appear to be due to either a load step or a tap-change within the

distribution network. A small phase change also results, which leads to a brief perception of frequency change. It is this perceived frequency change which presents the risk of spurious LOM detection. The measurement of ROCOF is as high as 2 Hz/s (both in the negative and positive directions). This is a large amount (much larger than the "genuine" background 0.01-0.025 Hz/s ROCOF rates). This finding, when added to the work of section 6.1, indicates that spurious trips of loss-of-mains relays based upon frequency, phase or ROCOF detection are far more likely to be caused by faults and transient events than genuine frequency changes.

To test the discrimination of the proposed POR against such transient events, the captured event waveforms (sampled phase voltages) from the laboratory hardware, can be replayed into the proposed relay code. The logged data is sampled at 2000 Sa/s, so to test the main algorithm at a sample rate of 500 Sa/s (nominally 10 samples per cycle at 50 Hz) the data is FIR-filtered using a 3-zero/3-pole filter (to remove potentially aliased harmonics as described in 4.5) and down-sampled by a factor of 4, before being input to the main algorithms. The main algorithms are the amplitude/phase/frequency measurements from chapters 3 & 5 and the POR from section 6.1.1.

The graphs below show the performance of the algorithms and the POR due to the worst of the 7 transients recorded. The settings for the POR are a trigger threshold $R_{Trigger}$ = 0.15 Hz/s and a trip threshold Φ_{Trip} = 20°. These are the identical settings used for the testing in 6.1, and represent the tightest settings which are anticipated to be used in practice.

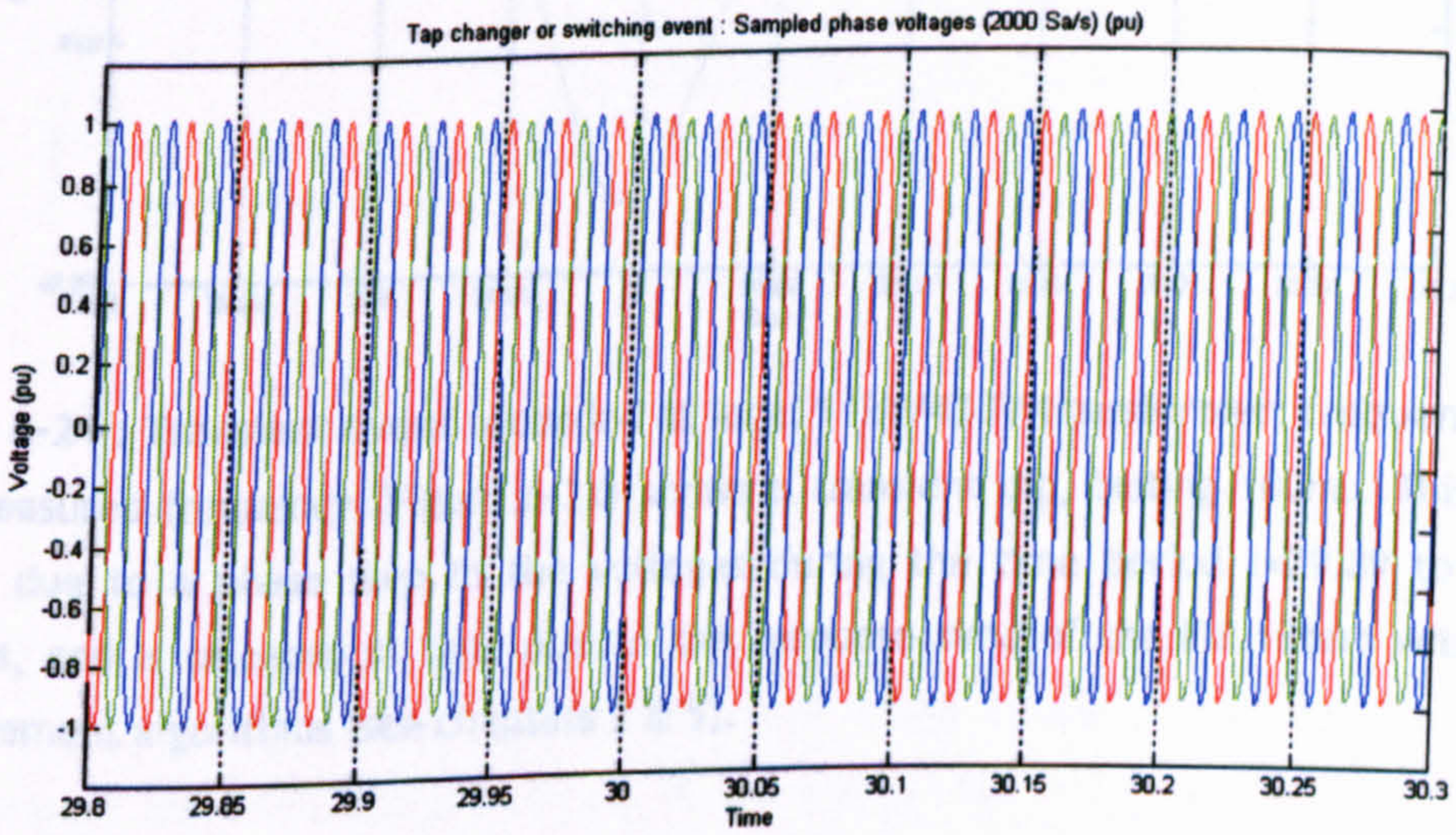


Figure 6-22 : Transient event recorded at local 11kV/433V transformer; sampled voltages (pu)

Clearly, from Figure 6-22, there is no significant voltage dip or surge on any phase.

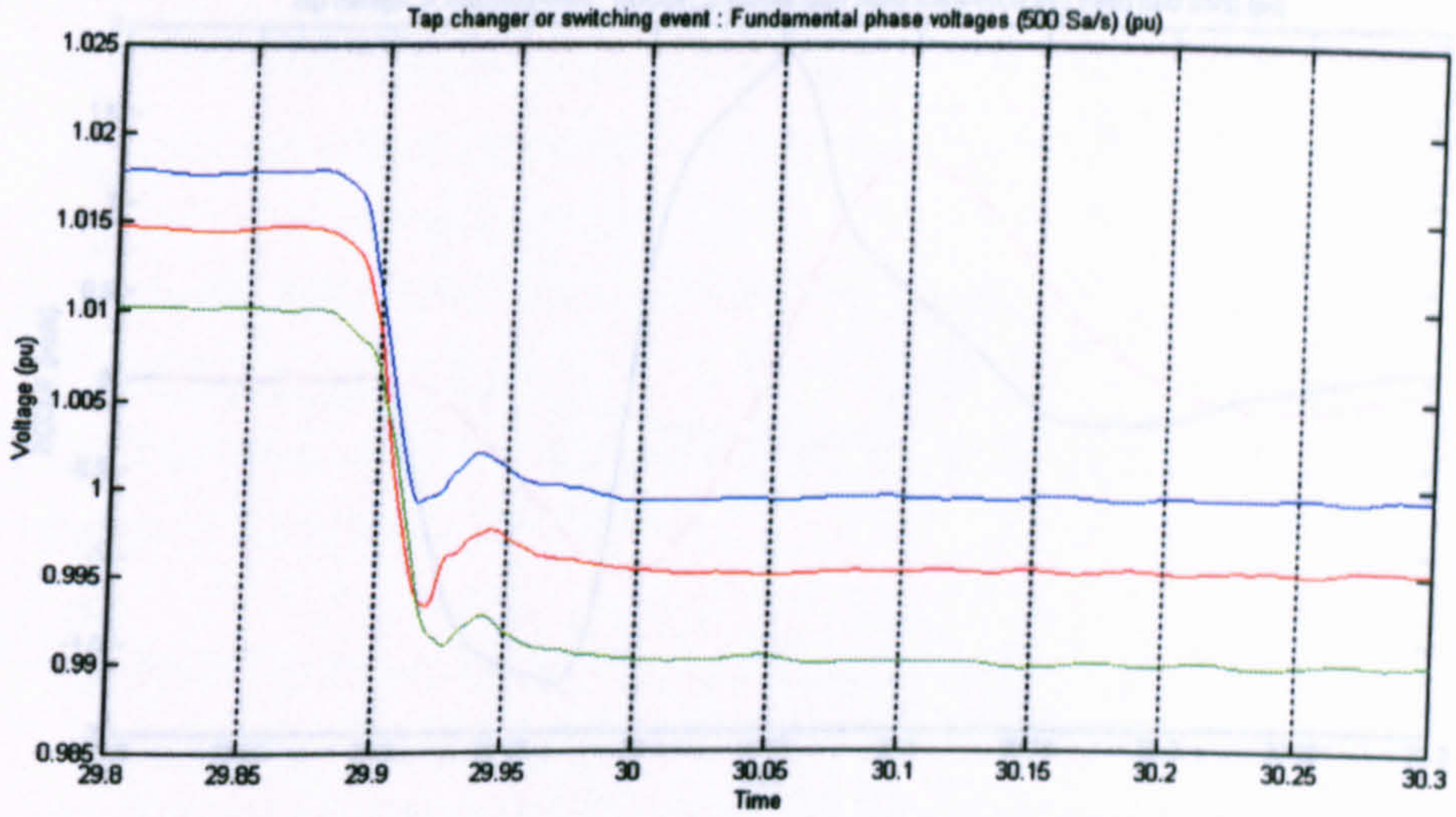


Figure 6-23 : Transient event recorded at local 11kV/433V transformer; fundamental voltages (pu)

Analysis of the fundamental voltage amplitudes shows that a 0.02pu voltage step event does in fact occur at t≈29.88s. This may be due to a tap changer or switching.

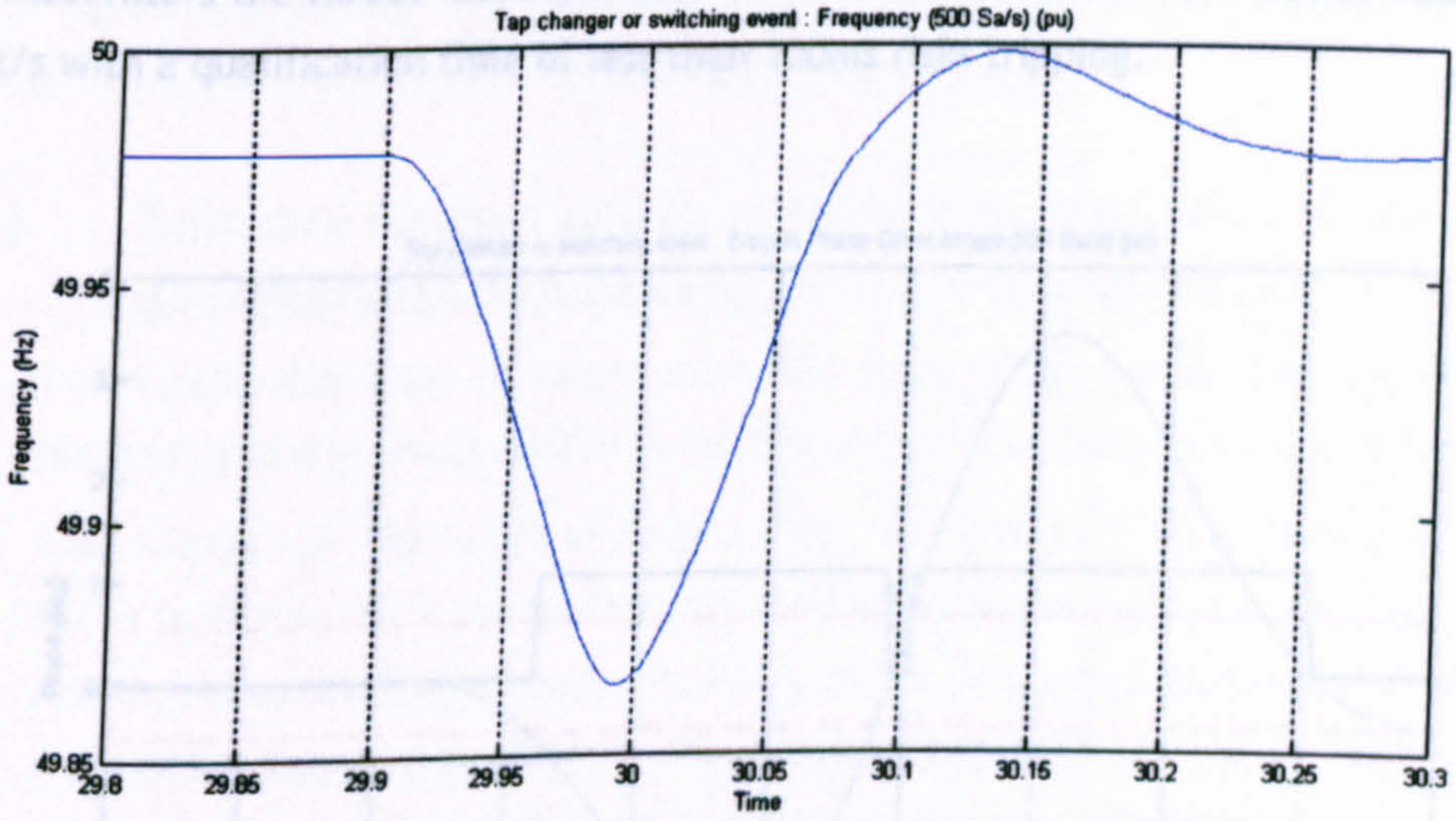


Figure 6-24 : Transient event recorded at local 11kV/433V transformer; frequency (Hz)

The measured frequency (Figure 6-24) shows a transient dip, lasting 100ms. This shape results due to a phase step in the voltages during the time period t=29.89 to t=29.9 seconds, and is smeared in time due to the response time of the FIR filters within the measurement algorithms (see chapters 3 & 5).

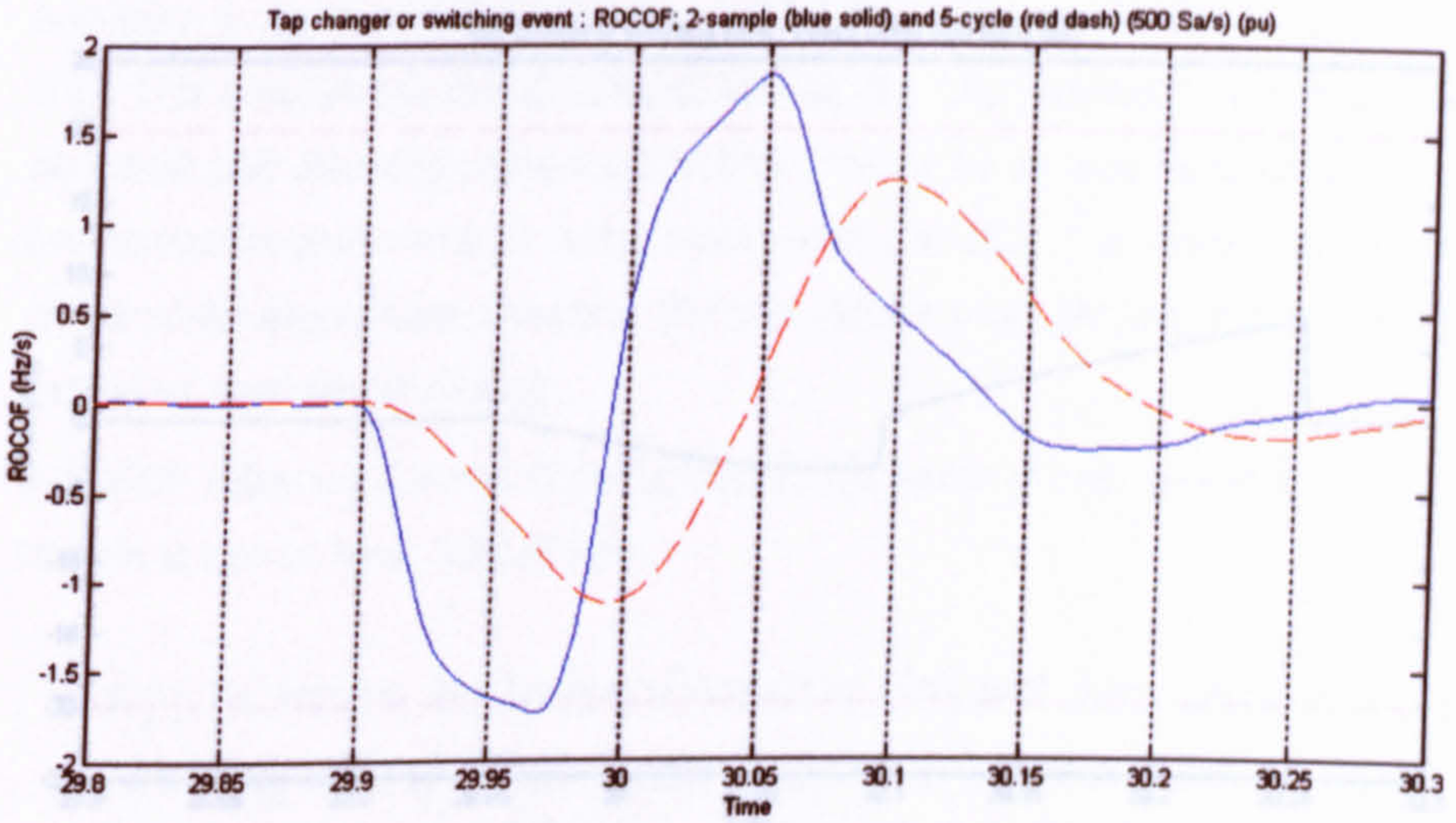


Figure 6-25 : Transient event recorded at local 11kV/433V transformer; ROCOF (Hz/s)

Figure 6-25 shows the calculated ROCOF. The POR uses the ROCOF value shown as a blue solid line, calculated by the 2-sample differentiation of the frequency shown in Figure 6-24. The red dashed line shows the same data averaged over a further 5 cycles (100ms); it is not used in the subsequent analysis but shows that even a ROCOF relay which smoothes/filters the ROCOF data will show a >1Hz/s deviation. Thus a ROCOF relay set to 0.5Hz/s with a qualification time of less than 100ms risks tripping.

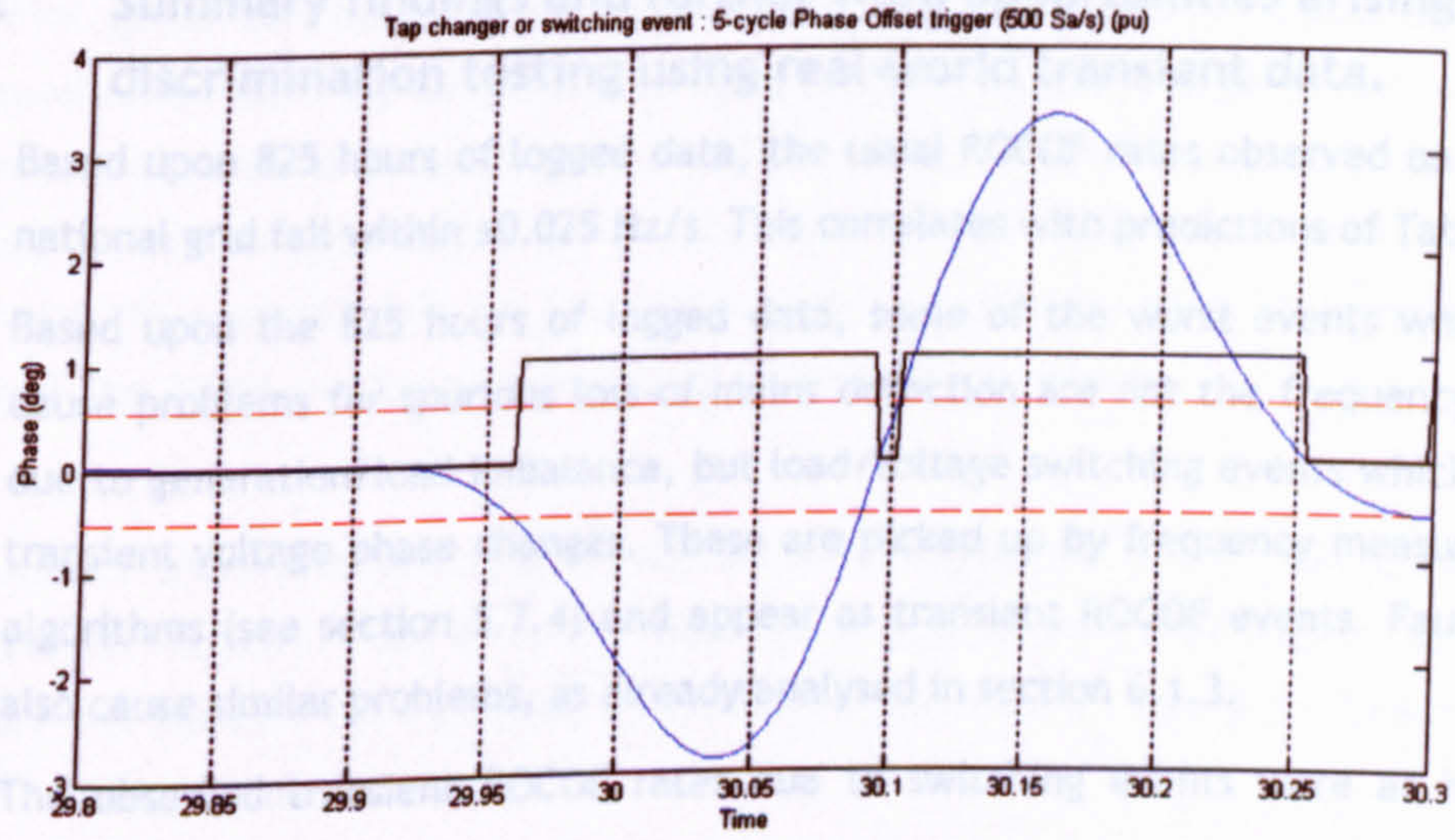


Figure 6-26 : Transient event recorded at local 11kV/433V transformer; POR triggering

The POR relay is triggered at t=29.97s when the rolling 5-cycle phase offset (blue solid line) exceeds the trip threshold (shown in red dashes) set by $R_{Trigger}$ = 0.15 Hz/s (Figure 6-26). The relay is reset at t=30.09s when the rolling 5-cycle phase offset crosses through zero, and is then triggered again at t=30.1s due to another violation of the threshold.

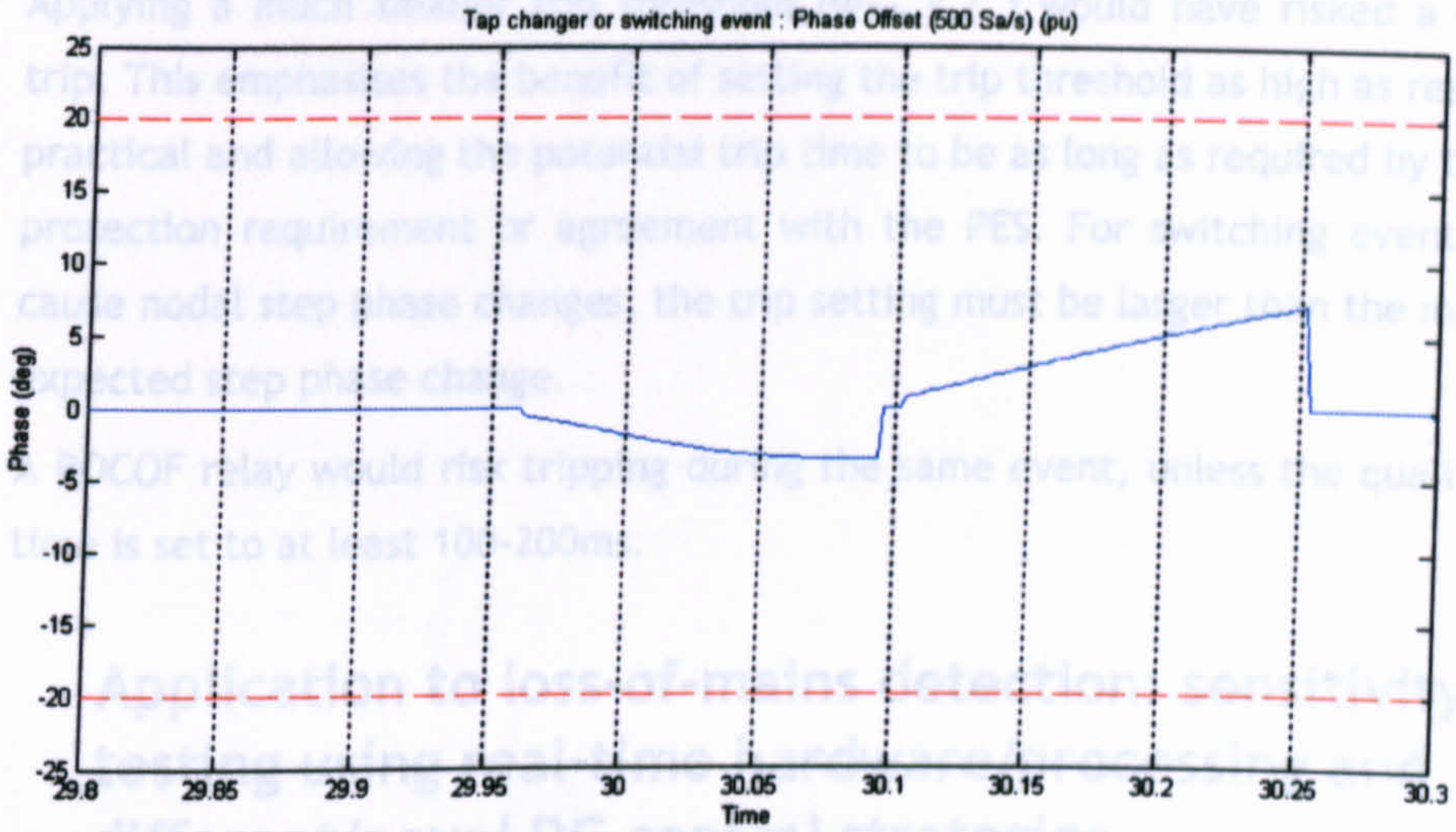


Figure 6-27 : Transient event recorded at local 11kV/433V transformer; phase offset

Although ROCOF is large (over 1Hz/s as shown in Figure 6-25), the duration for the event is short, and the ROCOF value reverses from a -ve to a +ve direction, causing the POR triggering to reset, as shown in Figure 6-26. The phase offset never reaches the trip threshold of $\Phi_{Trip} = 20°$. The largest angle reached is about 7.1°. This would have caused a trip if a tighter trip setting of 5° had been used (as used by Dysko(2006)) to achieve 500ms tripping with a 2.5% active power unbalance (see section 6.1.3.1).

6.2.1 Summary findings and further work opportunities arising from discrimination testing using real-world transient data.

- Based upon 825 hours of logged data, the usual ROCOF rates observed on the UK national grid fall within ±0.025 Hz/s. This correlates with predictions of Table 2-1.
- Based upon the 825 hours of logged data, some of the worst events which can cause problems for spurious loss-of-mains detection are not the frequency slides due to generation/load imbalance, but load/voltage switching events which cause transient voltage phase changes. These are picked up by frequency measurement algorithms (see section 5.7.4) and appear as transient ROCOF events. Faults will also cause similar problems, as already analysed in section 6.1.3.
- The observed transient ROCOF rates due to switching events were as high as ±1.8 Hz/s.
- For the worst event observed, the proposed POR does not trip, despite the trigger threshold ($R_{Trigger}$ = 0.15 Hz/s) being substantially less than the peak ROCOF value of 1.8 Hz/s. The relay successfully discriminates between the switching events and a genuine loss-of-mains event. This is due to the variable trip time allowed, as the transient event is smaller than the trip threshold $\Phi_{Trip} = 20°$.

- Applying a much smaller trip threshold ($\Phi_{Trip} < 7°$) would have risked a spurious trip. This emphasises the benefit of setting the trip threshold as high as reasonably practical and allowing the potential trip time to be as long as required by the local protection requirement or agreement with the PES. For switching events which cause nodal step phase changes, the trip setting must be larger than the maximum expected step phase change.
- A ROCOF relay would risk tripping during the same event, unless the qualification time is set to at least 100-200ms.

6.3 Application to loss-of-mains detection: sensitivity testing using real-time hardware/processing and different/novel DG control strategies

During the work leading to this thesis, a laboratory power systems network capable of performing experiments in microgrid control has been created by the author. A schematic is shown in Figure 6-28. The laboratory network contains a "Grid Supply Point" (GSP), which is the connection point to a "parent network". There are two options available for use as the parent network: the local 433V 3-phase mains supply can be used, or an 80kVA synchronous generator. The 80kVA synchronous generator is a more flexible device to use, since the frequency and voltage can be perturbed away from nominal values as desired (through sets of scenarios) to mimic problems within the parent network. The terminals of the 80kVA generator can also be synchronised with a real-time digital simulation (performed on an RTDS digital simulator, RTDS (2008)) of a much larger power network. Measurement of currents at the machine terminals can be fed back into the simulation (via simulated current sources) to provide a closed-loop simulation-hardware-simulation path. In this way, the remainder of the laboratory power network becomes "hardware-in-the-loop", with the 80kVA synchronous generator as the linking hardware.

For the loss-of-mains tests presented in this section, the national grid LV supply was used as the parent network. This is simply because it was quieter than using the 80kVA generator, and because the flexibility of the 80kVA generator as a parent network is not required for loss-of-mains testing.

The laboratory contains 2 separate microgrids containing distributed generation of both the synchronous and solid-state inverter style. Local loads can be configured on each microgrid to represent steady or fluctuating loads, both real and reactive. The microgrids are controlled by locally autonomous control algorithms, developed by the author and prototyped on the ADI RTS (Real-Time-Station) (ADI, 2008).

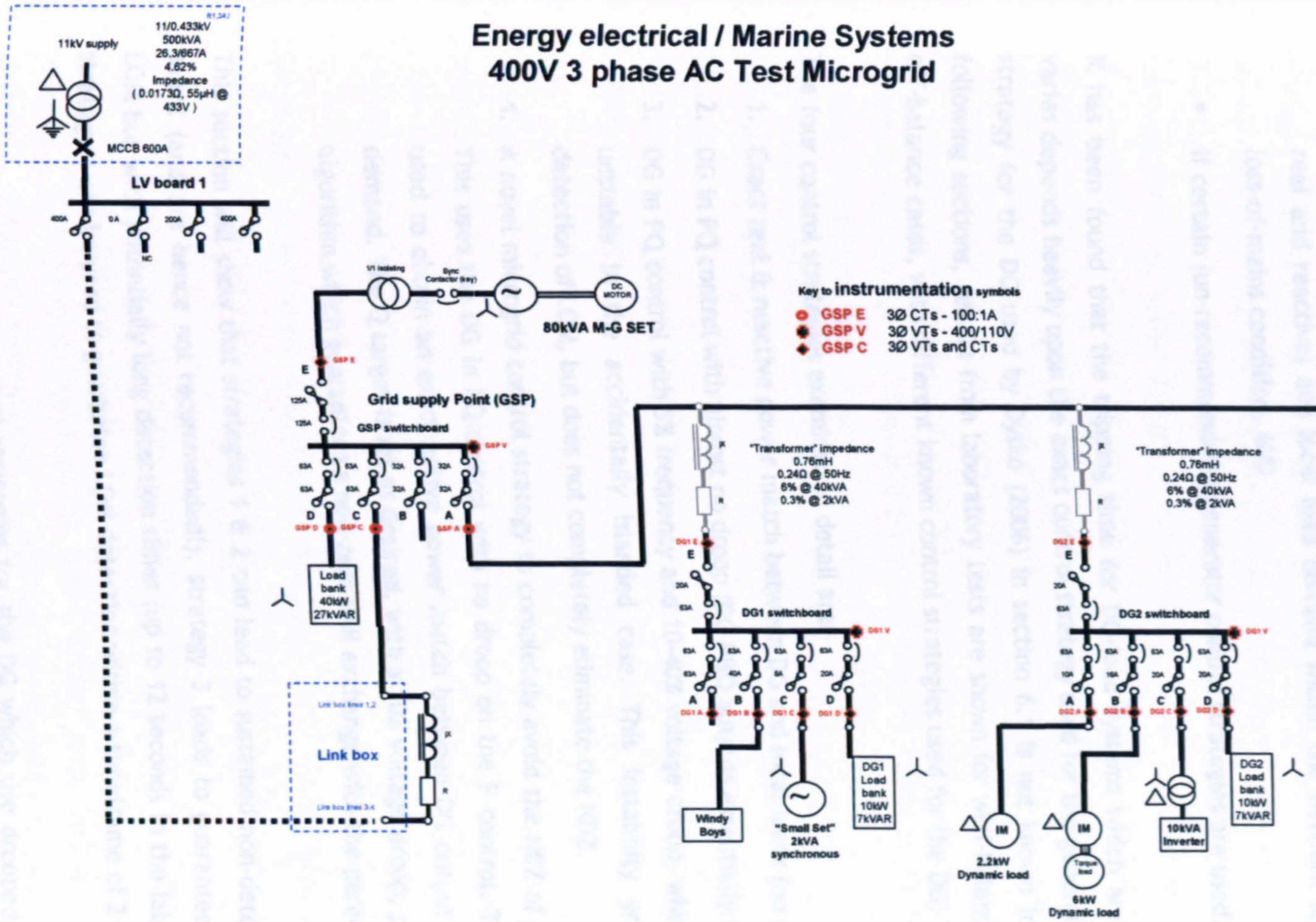


Figure 6-28 : 400V 3-phase microgrid laboratory; single-line diagram

6.3.1 Generator control strategies in grid-connected mode

Experience with the hardware of in Figure 6-28 confirms that it is relatively difficult to obtain a match between local generation set points and local load demands, in both P and Q, such that the LOM event cannot be detected within a timeframe of 2 seconds, when using the Phase Offset Relay (POR) of section 6.1. It is, however, possible to create such non-detection scenarios in the laboratory with synchronous generators:

- by deliberately creating an exact match between generator power output (both real and reactive) and local load demand within the network subjected to the loss-of-mains condition, AND
- if certain (un-recommended!) generator control strategies are used.

It has been found that the tripping time for DG-load systems which are well balanced varies depends heavily upon the exact control strategy used for the generator. The control strategy for the DG used by Dysko (2006) in section 6.1 is not known in detail. In the following sections, results from laboratory tests are shown for well-balanced and slightly off-balance cases, with different known control strategies used for the DG.

The four control strategies examined in detail are:-

1. Exact real & reactive power match between DG and local loads (no droops)
2. DG in PQ control with almost no droop (DG P&Q outputs essentially fixed)
3. DG in PQ control with 5% frequency and 10-40% voltage droop, which proves to be unstable in the accidentally islanded case. This instability greatly aids the detection of LOM, but does not completely eliminate the NDZ.
4. A novel microgrid control strategy to completely avoid the NDZ of LOM detection. This uses the DG in PQ control with no droop on the P control. The P control is used to obtain an exact active power match between DG output and local load demand. The Q target is set as desired, with a 10% voltage droop, plus an override algorithm which guarantees a non-zero VAR exchange with the parent network.

This section will show that strategies 1 & 2 can lead to sustained non-detection of a LOM event (and are hence not recommended!), strategy 3 leads to guaranteed detection of LOM but with potentially long detection times (up to 12 seconds in the laboratory), while strategy 4 can be used to guarantee LOM detection within a timeframe of 2 seconds.

It should be noted that control strategies for the DG which use drooped frequency and

voltage targets ("FV" type control) are not examined in detail here. It can be shown that such strategies can lead to very large LOM non-detection zones (sustained non-detection), since the controls are designed to be inherently stable in the islanded state. This type of DG control scheme is thus not appropriate when using local (communication-less) passive LOM detection relays.

Strategies 1 and 4 are of particular interest. There are times when deliberately obtaining a close match between DG output power and local load demand is desirable. The obvious example is during a planned change-over to an islanded state, which might be done to ensure security of supply or to improve power quality. Before changing to islanded mode, it is highly beneficial to pre-match DG real power output to the local load real power demand. When the actual change to islanded mode occurs, the frequency excursions are thus minimised as the prime mover throttle control is already at the correct set-point. A mismatch in real power leads to a sudden prime mover power output requirement upon islanding, and this may take several seconds to implement. This applies not only to rotating prime movers but also to static prime movers such as fuel cells etc. which may have inherent lag times due to fuel pumps and pressures. If the output power cannot be ramped quickly enough, the frequency will go outside allowable limits (or in the case of an inverter, the DC bus may collapse or over-volt). A similar pre-match in reactive power is also desirable, although the tolerance to mismatches is much higher as the effect upon islanding will be a brief voltage surge or depression. The tolerance for short-term voltage excursions is at least ±10%. Voltage can be also be adjusted relatively quickly (compared to throttle settings), via electronic field controls for synchronous generators or switching patterns for inverters.

Strategy 1 aims for a perfect local match of both real and reactive power, and thus provides no frequency or voltage support. The resulting microgrid is also deliberately placed within the likely non-detection zone of loss-of-mains relays. Strategy 4 is a novel scheme which deliberately matches real power locally but does not aim to match reactive power locally. Instead, it provides voltage support to the local and wider network by VAR exchange with the parent network, while ensuring a finite VAR exchange with the parent network to completely avoid the non-detection zone of the loss-of-mains relay.

6.3.1.1 DG control strategy 1 - Exact real & reactive power match between DG and local loads

Strategy 1 uses PQ control without any droop controls at all, but an active balancing algorithm. This simple strategy aims to source exactly the same amount of power from the DG as is required by the local load, both for real and reactive powers. This strategy was

implemented to automatically and deliberately create scenarios which might demonstrate the non-detection zone of loss-of-mains relays in the laboratory.

6.3.1.2 DG control strategy 2 - PQ control with almost no droop

Strategy 2 uses PQ control with 10000% frequency droop (for a 1pu change in real power output) and 10000% voltage droop (for a 1pu change in reactive power output). In this case, the DG outputs an almost constant amount of real and reactive power, independent of measured frequency and voltage. There is only a very small restorative effect towards nominal frequency and voltage, and the system is unstable when any significant perturbation arises.

6.3.1.3 DG control strategy 3 - PQ control with 5% frequency and 10-40% voltage droop

Strategy 3 uses PQ control with 5% frequency droop (for a 1pu change in real power output) and 10-40% voltage droop (for a 1pu change in reactive power output). In this case, the DG outputs real and reactive powers which tend to have a "restorative" effect on frequency and voltage towards nominal values via the droop controls. However, the droop controllers and generator/prime mover controls/response contain phase lags, which tend to push the system into an unstable state in islanded mode. Such a PQ control strategy has been used by the author in the laboratory at Strathclyde. It has proved to be an appropriate control strategy for a grid-connected DG unit.

To understand the instability of strategy 3 following a LOM event, the control strategy and system response can be approximately modelled. A simplified and approximate diagram of the control system and plant is shown below (Figure 6-29). The diagram is split into two parts: P control (throttle or real power) and Q control (field or reactive power). The controller is designed to operate in grid-connected mode controlling the export of real and reactive power with droop controls, with frequency and voltage set predominantly by the parent network. In the diagram below, the control system and plant is placed (accidentally) in islanded mode. Here, it has been assumed for simplicity that the active and reactive power requirement of the loads is fixed. The generator electrical output powers P_{Gen} and Q_{Gen} are therefore also fixed at the load real and reactive powers, since the generator and loads are joined together in an islanded power system. Also, the cross-couplings between the P and Q systems have been ignored in the stability analysis. Some of the additional linkages which would be required to model all the real effects are indicated by dotted lines and boxes. The following effects are thus ignored:-

- Load real powers proportional to voltage or voltage^2 (such as light bulbs), or frequency or frequency^2 (such as fans).

- Load powers dependent upon rates of change of frequency. This would occur due to loads with inertia, causing regeneration for example.
- Load reactive powers being dependent upon voltage, voltage^2, or frequency.
- The impact upon voltage of the generator frequency (due to the rotor field current creating more volts at the armature).

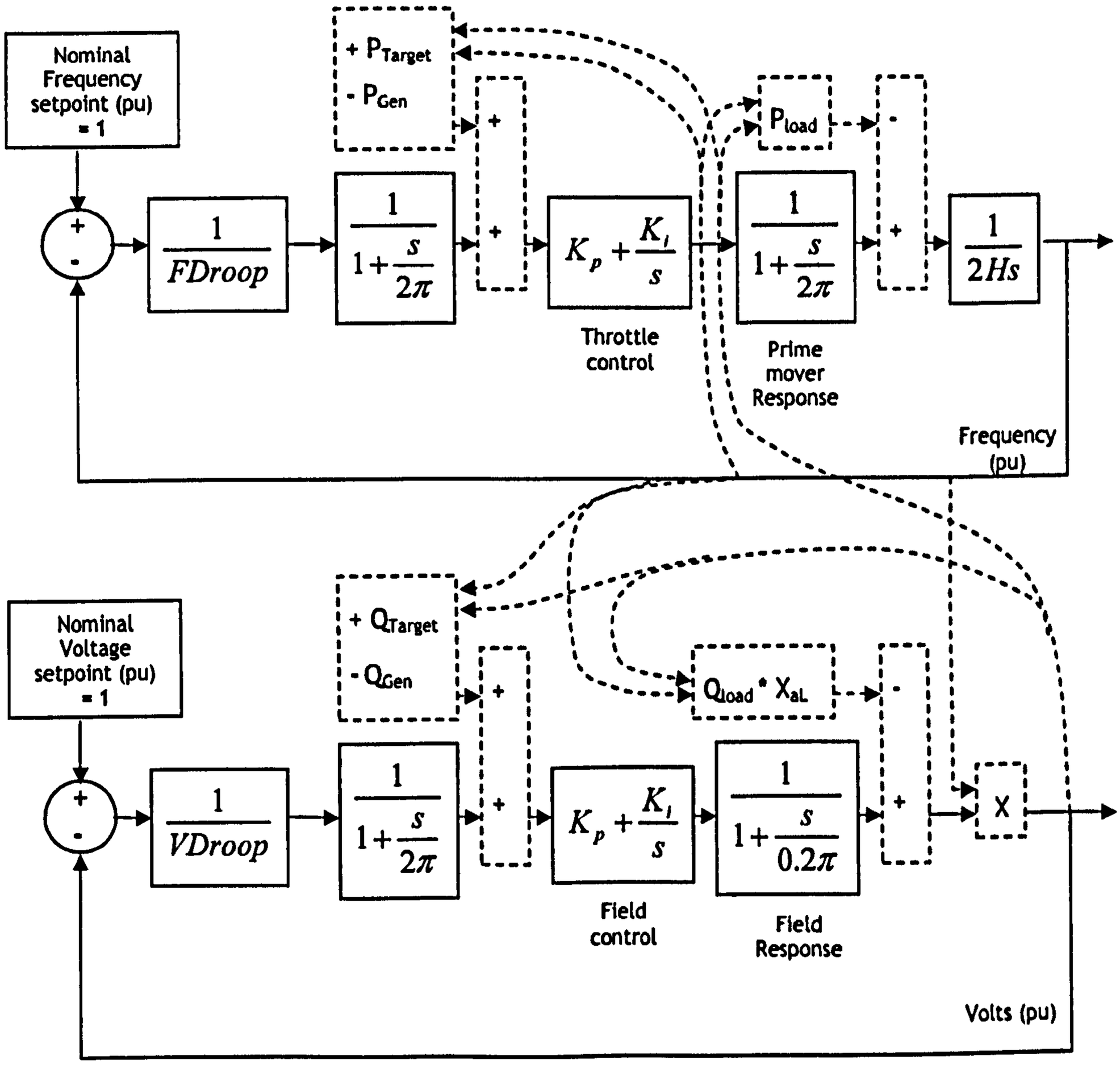


Figure 6-29 : Simplified diagram of control and plant during an unintentional island event. P-control and Q-control systems with droop.

A qualitative analysis of the simplified P control (real power) control loop shows that it consists of:-

- The inputs, which are the nominal frequency set-point (1pu) and the actual system frequency, against which the real power target is drooped.

- The drooped frequency setting *FDroop*, which is 0.05 (5%), giving a gain of 20.
- A low-pass filter implemented in software, set with a cut-off frequency of 1Hz. This smoothes noise and slugs the droop output. Gain is 1 at DC, and $1/\sqrt{2}$ at 1Hz, decreasing further with increasing frequency. A phase lag of 45° will be added for a 1 Hz signal, and up to 90° for higher frequencies.
- The throttle control. Here, K_p is set to 0, and all the control is integral, with K_I=1. This has a gain of infinity at DC, $1/2\pi$ at 1Hz, and further decreasing gain with frequency. Phase lag is fixed at 90° for all non-DC signals. This phase lag is an important component of the OLTF which introduces instability during islanded operation.
- The prime mover torque/power response time, estimated here by a low-pass filter with a cut-off frequency of 1 Hz. Gain is 1 at DC, and $1/\sqrt{2}$ at 1Hz, decreasing further with increasing frequency. A phase lag of 45° will be added for a 1 Hz signal, and up to 90° for higher frequencies. This represents a fast-responding prime mover and many large movers will have slower responses leading to higher instability in the following analysis.
- The prime mover (and hence system frequency) will speed up or slow down proportional to the power difference between primer mover power output and the load power. The rate of change is reduced by $1/2H$ where H is the prime mover / generator per-unit inertia. Here, H is estimated as 1, a suitable figure for a distributed generator (Mullane, 2005). This gives a gain of infinity at DC, $1/4\pi H$ at 1 Hz, and further decreasing gain with frequency.

The total P-control system open-loop transfer function thus has a gain of infinity at DC and $1/FDroop/16\pi^2 H$ at 1Hz, decreasing further with increasing frequency. This is a gain of 0.13 at 1Hz using the example values. The phase lag just above DC is 180°, rising to 270° at 1Hz, and increasing towards 360° at higher frequencies. This simple analysis is confirmed by the bode plot of the transfer function (Figure 6-30).

Referring to Figure 6-30, and using classical analysis of gain and phase margin, this system is unstable. The phase margin is non existent, because the phase lag of the OLTF (Open Loop Transfer Function) is equal to (and larger than) 180° over the range of frequency values from DC to 0.45Hz, where Gain is also >1. Similarly, there is no gain margin. This analysis predicts that the system will oscillate at some frequency below 0.45Hz. As will be seen in section 6.3.6, this is what is observed.

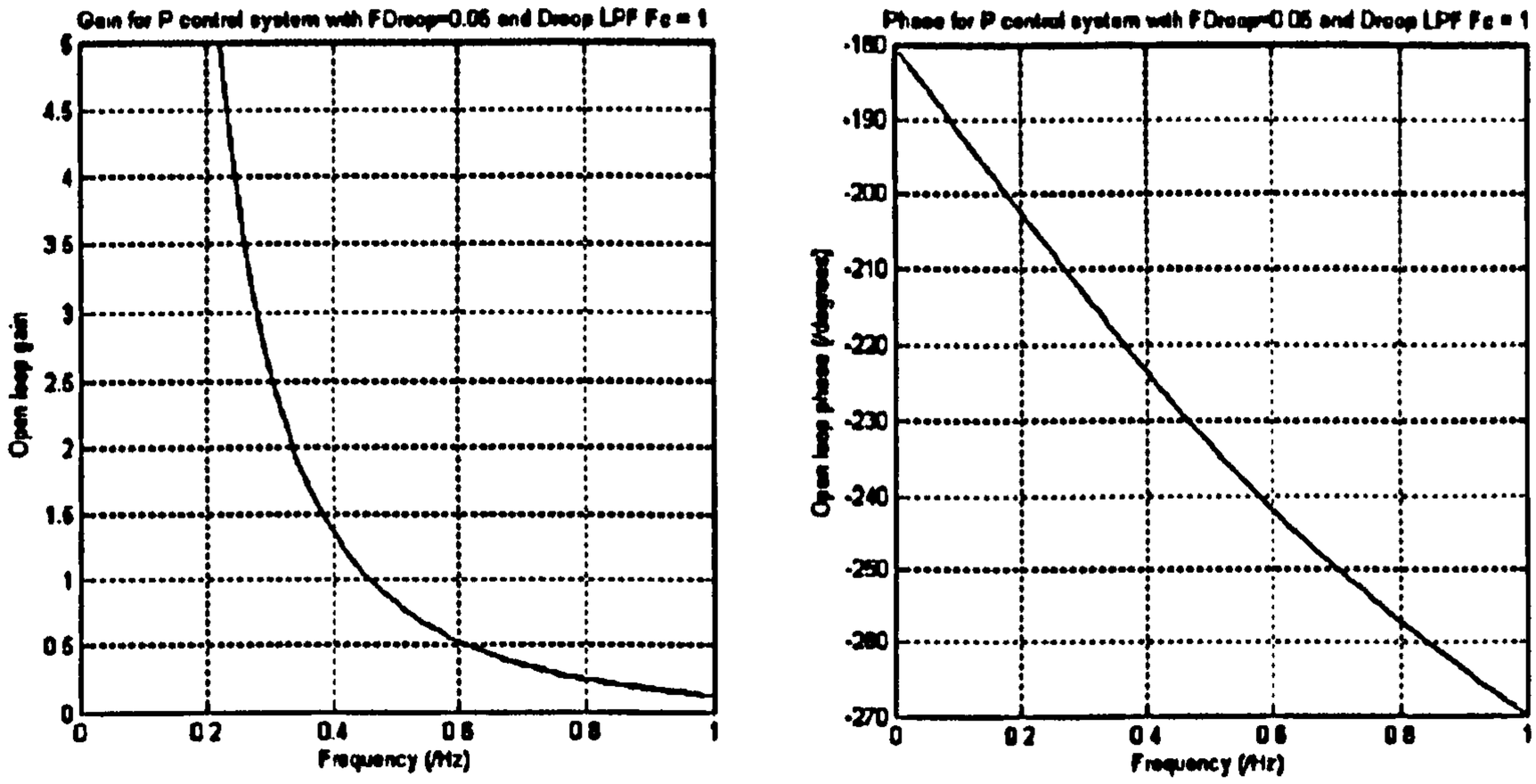


Figure 6-30 : Bode plot of simplified P (real power) grid-connected control loop during unintentional islanding, 5% frequency droop

A qualitative analysis of the simplified Q control (reactive power) control loop of Figure 6-29 shows that it consists of:-

- The inputs, which are the nominal voltage set-point (1pu) and the actual system voltage, against which the reactive power target is drooped.
- The drooped voltage setting *VDroop*, which is between 0.1 (10%) and 0.4 (40%), giving a gain of between 10 and 2.5 respectively.
- A low-pass filter implemented in software, set with a cut-off frequency of 1Hz. This smoothes noise and slugs the droop output. Gain is 1 at DC, and $1/\sqrt{2}$ at 1Hz, decreasing further with increasing frequency. A phase lag of 45° will be added for a 1 Hz signal, and up to 90° for higher frequencies.
- The field control. Here, K_p is set to 0, and all the control is integral, with K_i=1. This has a gain of infinity at DC, $1/2\pi$ at 1 Hz, and further decreasing gain with frequency. Phase lag is a fixed 90° for all non-DC signals. This phase lag is an important component of the OLTF which introduces instability during islanded operation.
- The synchronous generator field current drive response time, estimated here by a low-pass filter with a cut-off frequency of 0.1 Hz (the approximate response of an 80kVA synchronous generator in the Strathclyde laboratory). Gain is 1 at DC, $1/\sqrt{2}$ at 0.1Hz, and about 1/10 at 1 Hz, decreasing further with increasing frequency. A phase lag of 45° will be added for a 0.1 Hz signal, and up to 90° for higher frequencies.

The total Q-control system open-loop transfer function thus has a gain of infinity at DC and approximately 1/*VDroop*/(10/2*2π) at 1 Hz, decreasing further with increasing frequency. This represents a gain of 0.028 at 1Hz for the 40% droop case. The phase lag just above DC is 90°, rising to about 225° at 1 Hz, and increasing towards 270° at higher frequencies. This simple analysis is confirmed by the bode plot of the transfer function for the 40% droop slope case.

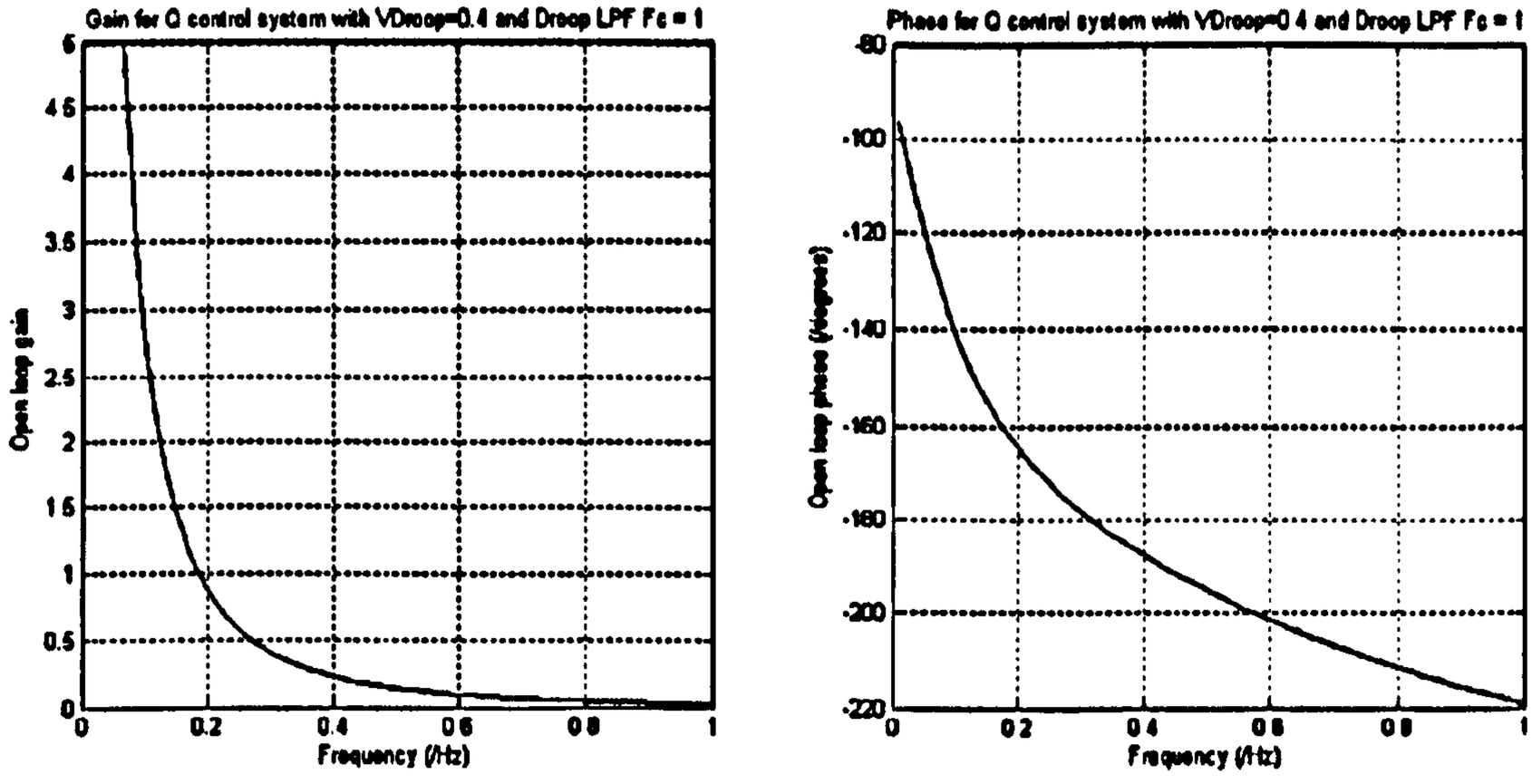


Figure 6-31 : Bode plot of simplified Q (real power) grid-connected control loop during unintentional islanding, 40% voltage droop

Referring to Figure 6-31, and using classical analysis of gain and phase margin, this system may just be stable. However the phase margin is only 20° and the gain margin is 9dB. The approximations made during the analysis are large enough that in reality the system could easily be unstable in isolation. This is especially true when the cross-couplings to the unstable P-control system are considered.

Changing the voltage droop from 40% to 10% results in a 4x (12dB) increase in the OLTF gain. The bode plot for the resulting system is shown in Figure 6-32. This system is now unstable in isolation, as was the P-control system. The phase margin is non existent, because the phase lag of the OLTF (Open Loop Transfer Function) is equal to (and larger than) 180° over the range of frequency values from 0.31Hz to 0.38Hz, where Gain is also >1. Similarly, there is no (-3dB) gain margin. This analysis predicts that the system will oscillate at some frequency between 0.31 and 0.38Hz.

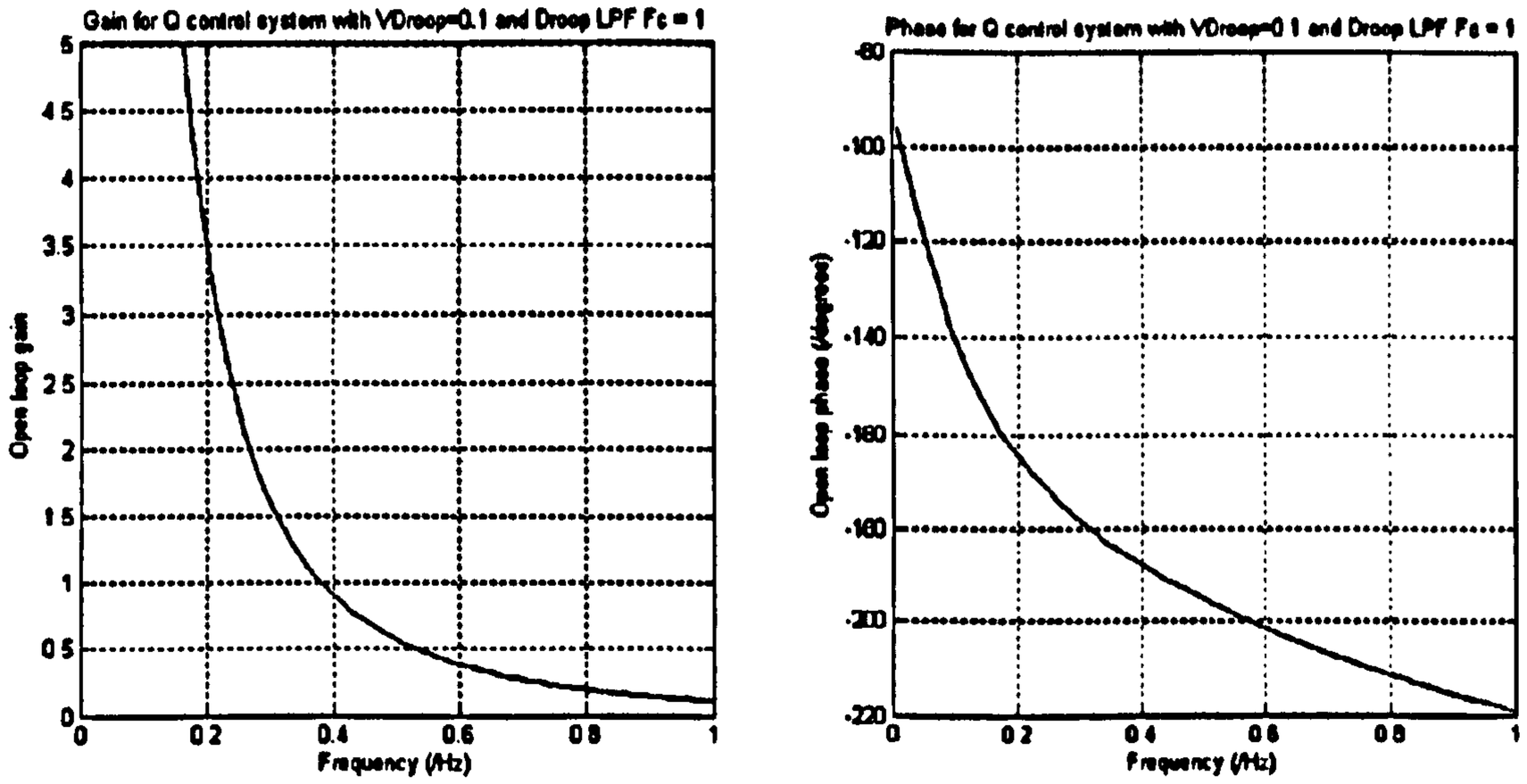


Figure 6-32 : Bode plot of simplified Q (real power) grid-connected control loop during unintentional islanding, 10% voltage droop

Works from several authors such as Du (2005) and Ye (2006) also advocate the use of grid-connected controls for DG which become fundamentally unstable when the power system becomes islanded. In these works, the controllers are adapted from frequency and voltage (FV) controllers (with P and Q offsets) using proportional or PI controls which are inherently stable in islanded mode. These can only be made unstable by adding positive feedback terms via bandpass filters. The advantage of strategy 3 (and 4) over such previously published works is that in strategy 3 & 4 the traditional droop control settings are retained intact, so that network support and power/VAR sharing functions are maintained, without the addition of extra feedback terms. This allows the DG unit to partake in frequency and voltage support, via modified P and Q exports dependent upon measured system frequency and voltage. The instability arises purely from the natural instability of this system (when unintentionally islanded) due to phase lags (particularly due to the integral-only controls) and droop slopes (loop gains).

6.3.1.4 DG control strategy 4 - Real power match and 10% voltage droop, guaranteed LOM NDZ avoidance using non-zero VAR exchange

Strategy 4 uses PQ control based on strategy 3 but with four significant modifications:-

- Generator real power output can be deliberately matched to the local load demand, by an automatic process without a droop control.
- The reactive power export target from the DG is set to 0.
- Voltage droop is set at 10%, to provide significant voltage support, and also to guarantee that the Q control system is independently unstable (by the analysis of section 6.3.1.3 and Figure 6-32).

- A simple over-ride algorithm always insists on a measurable VAR exchange (more than about ±0.05pu) with the parent network. This guarantees that exact PQ balance within the local microgrid is never obtained and the loss-of-mains event is detectable within a reasonable timeframe (Affonso, 2005), even when an exact active-power balance is achieved.

This method actively makes a balance of real power generation and demand, driving the real power exchange with the parent network to zero. The reactive power import/export will adjust itself, via the 10% droop control, in order to provide a stabilising effect on voltage towards the nominal voltage level (it provides significant voltage support, up to 1pu VAR export/import if the voltage varies from nominal by 10%). A zero reactive power exchange with the parent network is avoided via a strategy that adjusts the reactive power flow, and is simple in concept. Care has to be taken to add appropriate hysteresis within the algorithm, however, to stop it cycling around decision thresholds. In some respects the algorithm is similar to that proposed by Lokov (2005), however in that work the active power exchange with the parent network is deliberately moved from zero, while the reactive power is not considered. The algorithm proposed here adjusts reactive power flow away from zero, and hence allows much more freedom of generator set-point selection, particularly with respect to active power.

This novel control strategy involves an algorithm with several steps. This algorithm can be executed continuously in real time while a generator (and its local power system including loads) is grid-connected to a parent network. The algorithm is the subject of UK patent application 0810512.4 filed 10 June 2008 (Roscoe, 2008).

A diagram of the context is shown in Figure 6-33.

The algorithm ensures that either real power or reactive power exchange (P_{Net} and Q_{Net}) between the local power system and the parent network are above certain thresholds. These thresholds are small in per-unit terms. However, they are large enough that, upon unintentional islanding, the local power system control is always perturbed enough that the unstable controls of generator real and reactive power outputs result in oscillation and detectable loss-of-mains conditions within less than 2 seconds (Affonso, 2005).

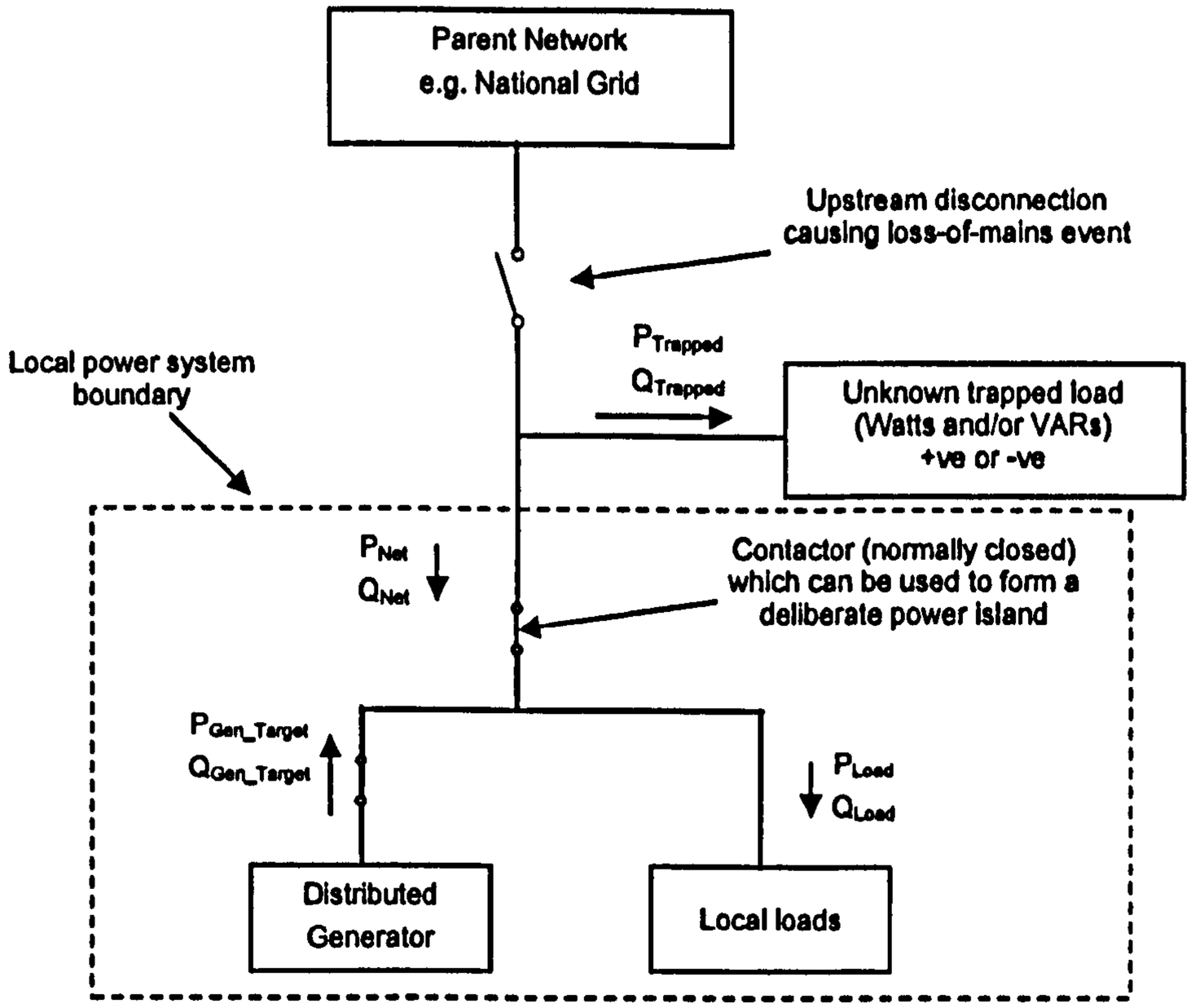


Figure 6-33 : Electrical single-line diagram of the context for control strategy 4

In simple terms, the algorithm is described below:-

1) Either: intentionally match the local active power generation target P_{Gen_Target} with local real power demand P_{Load}, or, determine the target P_{Gen_Target} appropriately using financial, power flow or operational drivers/constraints. This optional power matching can be an important precursor to deliberately islanded operation, because it minimises the frequency and prime mover transients when a change from grid-connected to islanded operation occurs. However, matching the active power target also means that the local generation tends to track any local load changes and this may result in prime mover wear or non-optimal efficiency due to the constant adjustment of throttle and power output levels. Some of this effect can be mitigated by suitable dead-band functions within the prime mover throttle control. The decision on whether to match the active power target must be taken by appropriate trade-offs between the risks of outage versus the costs of operation in the different modes. These inputs may change dynamically in real time, requiring a constant re-assessment of the optimal operational mode.
2) Monitor the real power exchange from the parent network to the local power system. Call this power flow P_{Net}.
3) If the absolute value of P_{Net} falls below a set threshold P_t, then there is a close

match of real power between local load demand and local generation output (excluding trapped load), and the flag P_{Match} is set to TRUE (otherwise P_{Match} is set to FALSE).

4) If P_{Match} is FALSE, then the algorithm is finished and starts again at step 1 in the next execution frame.

5) Otherwise, if P_{Match} is TRUE, continue ...

6) Monitor the reactive power flow from the parent network to the local power system. Call this power flow Q_{Net}. In practice, due to hysteresis considerations within the following control algorithm, Q_{Net} should not be measured directly, but instead deduced by subtracting the normal generator VAR output target Q_{Gen_Target} (after the set point Q target and droop controls have been applied) from the measured local load reactive power demand Q_{Load}.

7) If the absolute value of Q_{Net} falls below a threshold Q_t, then there is (or will be) a close match of real and reactive power between local load demand and local generation output (excluding trapped load). The flag Q_{Adjust} is set to TRUE.

8) If Q_{Adjust} is FALSE, then the algorithm is finished and starts again at step 1 in the next execution frame. If Q_{Adjust} is TRUE, then continue ....

9) The aim is now to adjust Q_{Gen_Target}, the target reactive power output from the generator, such that Q_{Net}, the reactive power flow from the parent network to the local power system, has an absolute value of plus or minus Q_t, the threshold below which we do not want abs(Q_{Net}) to fall. Decide which way to adjust Q_{Gen_Target} (up or down) in order to achieve abs(Q_{Net})> Q_t with the minimum of adjustment to the original generator reactive power target Q_{Gen_Target}. The decision can be made by using a flag Q_{Adjust_Up}=(Q_{Gen_Target}.Q_{Load}>0). This means that if Q_{Gen_Target}>Q_{Load}, Q_{Adjust_Up} will be TRUE, or FALSE otherwise. The idea here is that if Q_{Gen_Target}>Q_{Load}, reactive power is already flowing from the local power system back into the parent network (but the magnitude of the reactive power flow is less than Q_t as already determined). Thus, in this case, increasing the reactive power output power from the generator by less than Q_t will cause abs(Q_{Net}) to exceed Q_t which is the desired result. If the generator output was reduced, then it would have to be reduced by more than Q_t to achieve abs(Q_{Net})>Q_t.

10) The modified generator reactive power output target can now be calculated from $Q_{Gen\_Target\_New}$ = Q_{Load} + Q_t (if Q_{Adjust_Up} is TRUE), or $Q_{Gen\_Target\_New}$ = Q_{Load} - Q_t (if Q_{Adjust_Up} is FALSE)

11) A final check is that $Q_{Gen\_Target\_New}$ is within the acceptable control range of the generator. If not, then the setting of Q_{Adjust_Up} should be inverted and the value of $Q_{Gen\_Target\_New}$ recalculated.

An overview of this process (not including all the details) is summarised graphically in Figure 6-35. This process is repeated at a suitable frame rate within a microgrid controller.

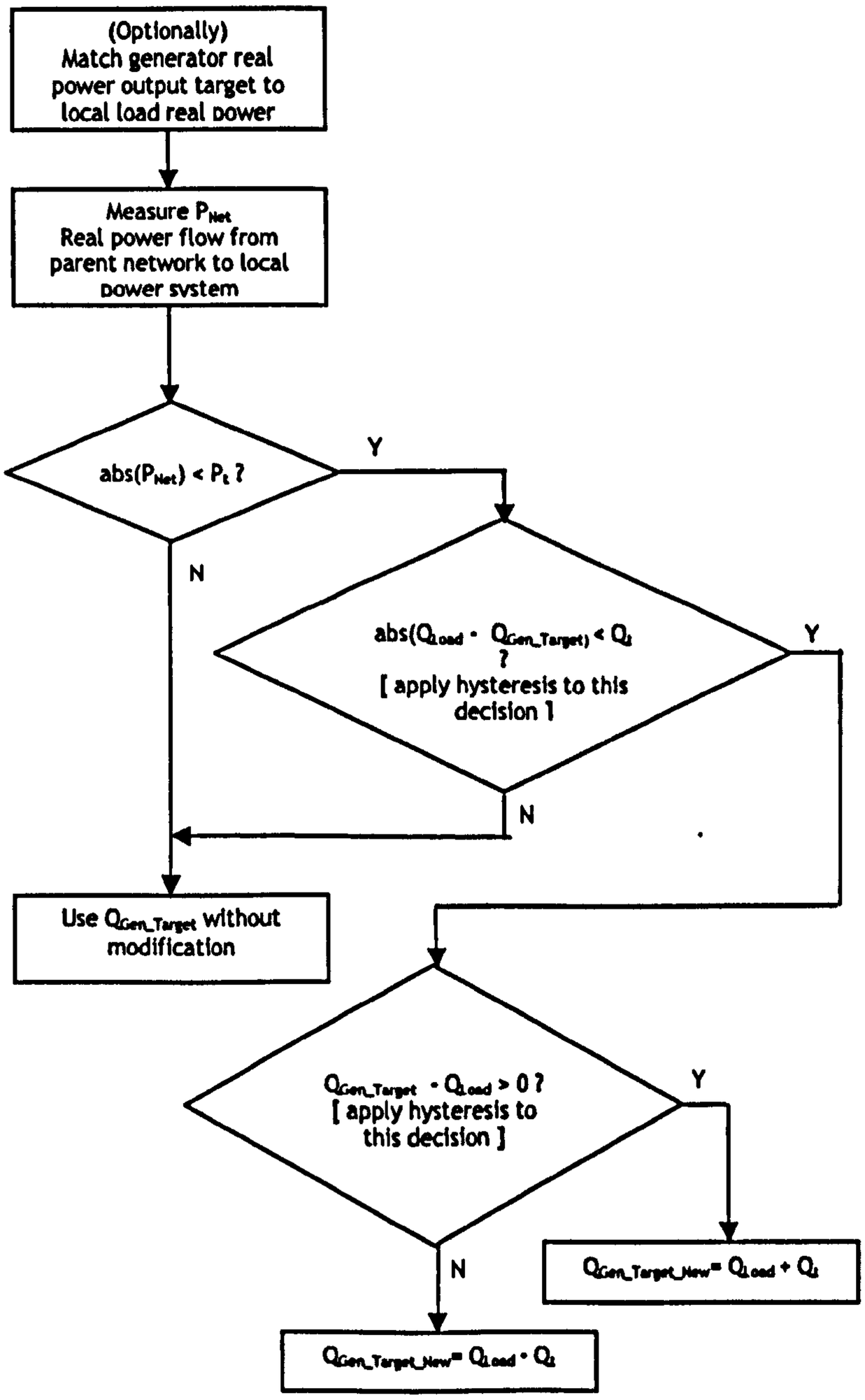


Figure 6-34 : Abbreviated summary of strategy 4 algorithm

It should be noted that in addition to the steps described above, appropriate hysteresis (involving time and/or decision thresholds) should be applied during the decision-making processes which set the Boolean values of

- P_{Match} (Boolean decision)
- Q_{Adjust} (Boolean decision)
- Q_{Adjust_Up} (Boolean decision)

This hysteresis avoids the controls changing regularly from 0 to 1 and vice versa when the

active and reactive power flows are hovering around the decision thresholds. There is also some value in converting the Boolean value Q_{Adjust_Up} (with value 0 or 1) into a floating point value $Q_{Adjust\_Direction}$ with a value of -1 or +1 respectively. This can then be passed through a slew-rate filter with appropriate slew rate limits to give $Q_{Adjust\_Direction\_Rate\_Limited}$. This slew-rate limiting simply smoothes out any step changes to generator reactive power targets which would otherwise occur. $Q_{Gen\_Target\_New}$ can then be calculated from

$$Q_{Gen\_Target\_New} = Q_{Load} + Q_t * Q_{Adjust\_Direction\_Rate\_Limited}$$

A final comment is that there may be trapped loads outside the boundary of the local power system, as shown on Figure 6-33. If a loss-of-mains event occurs, it may result in load (or generation) outside of the local power system becoming part of the unintentional power island.

The worst case would then be if the local active power P_{Gen} is not deliberately matched to P_{Load} and is in fact accidentally almost matched to (P_{Load} + $P_{Trapped}$), and also if a close reactive power match accidentally exists between Q_{Gen} and (Q_{Load} + $Q_{Trapped}$). This is an unlikely but potential scenario. Note that this scenario can be avoided by deliberately matching P_{Gen} to P_{Load}, as in the optional step 1) above. This means that if $P_{Trapped}$ is significantly non-zero, i.e. abs($P_{Trapped}$>P_t), then P_{Gen} will never be approximately equal to (P_{Load} + $P_{Trapped}$) since P_{Gen} = P_{Load}. If $P_{Trapped}$ is very close to zero, i.e. abs($P_{Trapped}$)≤P_t, then P_{Gen} will be very close to (P_{Load} + $P_{Trapped}$) but in this case the algorithmic steps 2) to 11) above will take place. For an accidental close match of reactive power Q_{Gen} = (Q_{Load} + $Q_{Trapped}$) to then also occur, abs($Q_{Trapped}$) would then have to be ≥Q_t, . This is unlikely if abs($P_{Trapped}$)≤P_t, unless the trapped load (or generator) has an extremely poor power factor. Thus is can be seen that, (counter-intuitively), operating the local power system with a deliberate match of local real power generation to local power demand can be used as a tool to avoid the non-detection zone of loss-of-mains, when the possibility of additional trapped loads exists.

Referring back to the system model for strategy 3 of section 6.3.1.3, in Figure 6-29, the control scheme for strategy 4 can be compared to it.

- the Q control scheme is identical except that the droop is changed from 40% to 10%, and the system is initially deliberately destabilised by the guaranteed VAR exchange.
- The P control scheme is different, due to the active power matching algorithm.

The first pass simplified control model for strategy 4 can be represented as shown below. This is adapted from Figure 6-29.

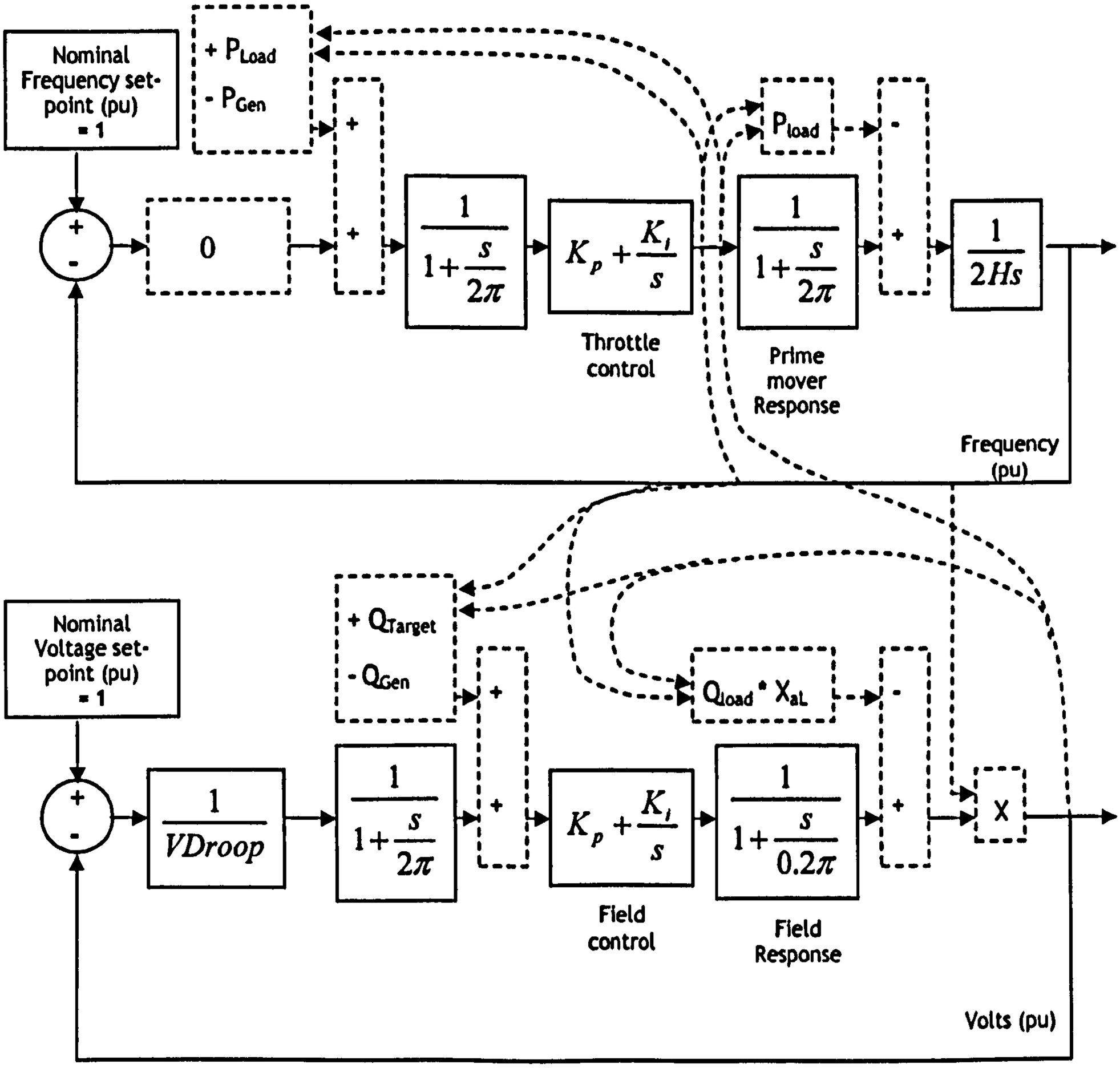


Figure 6-35 : Simplified diagram of control and plant during an unintentional island event. Active power match and drooped Q-control systems.

The bode plot and stability analysis for the Q droop system is identical to that shown in Figure 6-32. It is unstable and is likely to oscillate at 0.31 to 0.38Hz.

The P control loop, on the other hand, has been broken. The loop gain is now zero. Instead, the input to the PI controller for throttle is the difference between the measured load power and the measured generator output power. During normal grid-connected operation, the load power can be measured by adding the generator output power to the power fed by the parent network. Thus, the input to the PI controller is given by:-

$$PI\_Controller\_Input = P_{Load} - P_{Gen} = (P_{Gen} + P_{Net}) - P_{Gen} = P_{Net} \tag{6.5}$$

Therefore, during grid-connected operation, P_{Net}, the power flow from the parent network plus the trapped load adjustment, will be driven to zero by control action. However, when a loss-of-mains event occurs, P_{Net} becomes $-P_{Trapped}$, defined by the actual real power demand of any trapped load. Thus, the PI controller input will be $-P_{Trapped}$ by (6.5).

Similarly to control method 1, of section 6.3.1.1, the P control loop when analysed alone is not classically unstable when accidentally islanded due to LOM event. However, the loop gain is zero and thus there is no control at all. Any destabilising of the system (for example a small local load step, or $P_{Trapped} \neq 0$) will not be corrected by any restorative action, the input to the PI controller will be a constant non-zero number and power output will rise or fall in an approximately linear slope due to integral control action. When combined with an unstable Q control and the P-Q control cross-couplings, overall instability will almost certainly result. This causes frequency deviations allowing detection of the LOM event, and minimisation of the NDZ.

6.3.2 Laboratory testing of the POR with the 4 different DG control strategies

To assess the performance of both the POR and the different proposed DG control strategies, the laboratory microgrid of Figure 6-28 was used. The amplitude, phase, frequency, and POR algorithms developed in this thesis were integrated into a much larger set of algorithms which are designed to control an entire microgrid containing a single major despatchable DG unit and local loads (loads which may contain smaller despatchable or renewable DG units effectively contributing negative loads). This microgrid control application has been developed by the author, but to describe it in full is beyond the scope of this thesis. One of the major challenges within this application is the successful detection of loss-of-mains (LOM) within a microgrid scenario, particularly when the microgrid DG and local load powers are deliberately matched during a pre-islanding process as described in section 6.3.1. Since the microgrid control algorithm is designed to operate with either synchronous or inverter-connected generation, the LOM-detection algorithm is desired to be a passive method rather than an active method.

The microgrid control algorithm, and other related code to operate the entire network, is written in MATLAB Simulink (with some S-functions), auto-generated into "C" code, and then built into executables which run on a Real-Time-Station (RTS) (ADI, 2008). The main microgrid control application runs on a single CPU at a 500 Sa/s frame rate (nominally 10

samples per cycle). Other lab infrastructure, data logging etc. runs on parallel processors at the same frame rate. IO (Inputs/Outputs) is handled on a separate processor. This can be clocked at 1500Sa/s (30 samples per cycle, 3x oversampling) and FIR filters applied in a similar method to that described in section 4.5.

In the sections which follow, for each proposed DG control strategy, the following process (with small variations) was undertaken:

- Connect the microgrid loads to the national grid. The grid connection is directly via a 500kVA 11kV/433V transformer. The microgrid loads in the synchronous generator case were approximately 1300-1400W at a power factor of 0.9 lagging. This power level was limited by the synchronous generator rating. In the inverter case the loads were larger, due to the inverter having a larger capacity of 10kVA.
- Synchronise the generator to the microgrid (and the national grid). This a reliable automated process, built into the microgrid control application.
- Set droop controls as appropriate for the control strategy under test.
- Set P and Q set-points to achieve a very close balance in both real and reactive power, such that the P and Q exchange with the parent network (the national grid) become very small. Note that for methods 1 and 4 (see section 6.3.1) this balancing method is automatic.
- Begin data logging
- Instigate a deliberate loss-of-mains condition by opening a contactor ("GSP A" on in Figure 6-28) upstream of the microgrid
- Wait for LOM to be detected ...
- Stop data logging
- Repeat ...

The data logging captures all variables at a 250Sa/s frame rate (decimation 2 from the 500Sa/s main algorithm frame rate). Nodal voltages and currents can be captured at the full 500Sa/s frame rate if necessary, and subsequently re-analysed with new candidate measurement algorithms.

In all cases, the trip setting of the POR was 20°. The ROCOF trigger threshold was set to 0.2Hz/s for most tests, except for some of the tests of strategy 4 in section 6.3.7. The inertia of the synchronous motor-generator is approximately H=0.9 pu.

6.3.3 DG control strategy 1 - Exact real & reactive power match between DG and local loads.

As described in section 6.3.1.1, control strategy 1 deliberately aims for an exact balance between DG real & reactive power outputs and the local load demands. As such it deliberately places the local microgrid in an extremely vulnerable state as regards the potential for non-detection of the loss-of-mains condition. The likelihood of being within the non-detection zone and either having a no-trip or very long trip result is high.

In the laboratory, it was indeed shown that a sustainable power island was able to be formed, without detectability of loss-of-mains[1]. The experiment was repeatable. This clearly shows that control strategy 1 is not inherently unstable (although any change to local load demand will cause it to be). Strategy 1 is thus not recommended for DG installations where LOM must be detected using passive relays based upon voltage measurements.

Figure 6-36 to Figure 6-41 show the results of the test which shows sustained non-detection of LOM. At the start (t=520), the DG and local loads are grid-connected, but the DG is outputting 1000W in a drooped manner. At t=525 seconds, a "PQ" balance algorithm was engaged. The DG real and reactive powers then rise to meet the load powers (plus trapped load). The LOM event itself was instigated at t=534 seconds, and is not detected. After this, the undetected power island frequency and voltage stay almost constant for many seconds. Only after a deliberate 150W load change is made at t=594.75 seconds is the LOM detected (at t=595.3 s, a 550ms trip time).

Figure 6-36 : Sustained non-detection of LOM; Frequency (Hz)

[1] To achieve this result, it was necessary to account for 20W of parasitic trapped load upstream of the microgrid, which was still back-fed during the LOM condition. The 20W offset had to be artificially inserted into the control loop. This parasitic load was measured by experiment, and consists of neon power indicator lamps and voltage transformer loads. The same 20W control offset was subsequently included in all the tests of this section, to deliberately create the biggest risk of non-detection.

Figure 6-37 : Sustained non-detection of LOM; ROCOF (Hz/s) and instant of LOM inception (blue)

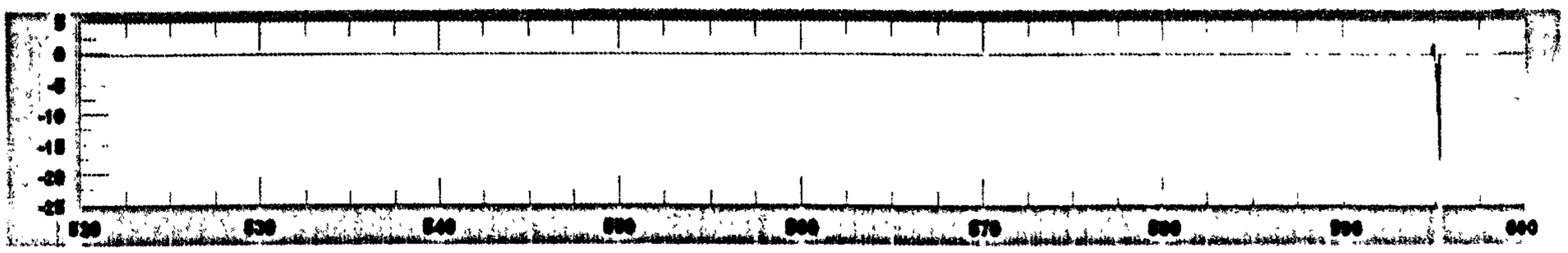


Figure 6-38 : Sustained non-detection of LOM; Phase Offset (degrees)

Figure 6-39 : Sustained non-detection of LOM; Voltage (+ve sequence fundamental, line-line)

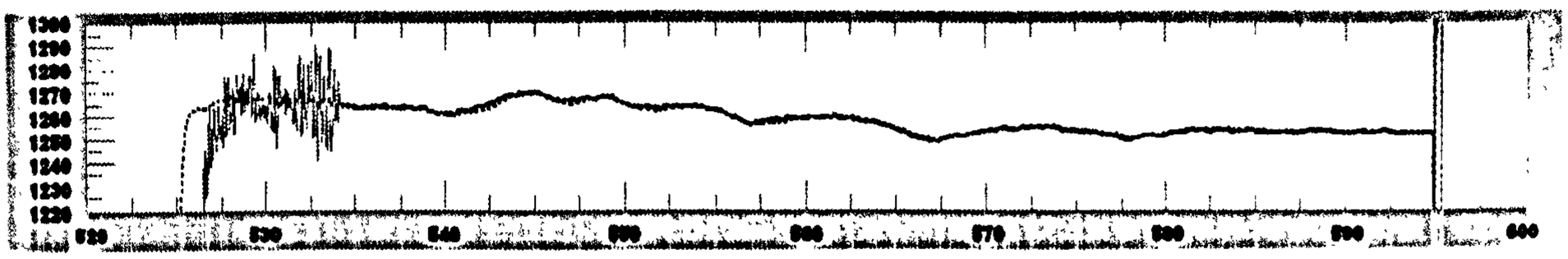


Figure 6-40 : Sustained non-detection of LOM; DG Real power output (solid) and target (dashed)

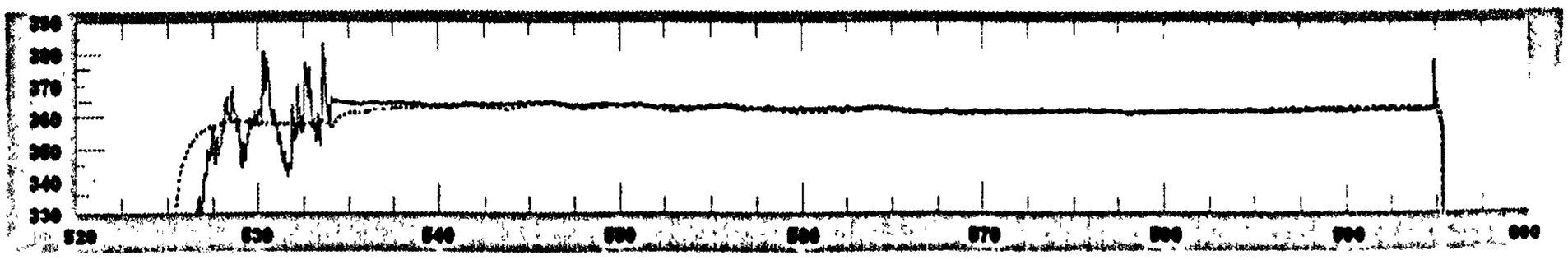


Figure 6-41 : Sustained non-detection of LOM; DG Reactive power output (solid) and target (dashed)

6.3.4 DG control strategy 1 - Exact real & reactive power match between DG and local loads, using inverter-connected generation

The above test method was repeated, but instead using a 10kVA inverter-connected generator and a suitable load magnitude. Despite the careful matching of both real and reactive powers[1], it was not possible to create a non-detection of loss-of-mains. This is due to the dynamics of the inverter control in grid-connected mode, and in particular the high bandwidth of the PLL which determines its effective inertia. This is extremely small, and hence the frequency excursion is much larger than for a synchronous generator when a LOM event occurs. The longest detection time captured was 1.4 seconds. This is shown below. The LOM event was instigated at t = 366.18 s and detected at t = 367.58 s.

This shows that (in this case), the use of inverter-connected generation can reduce the size of the LOM NDZ, due to the reduced "inertia" of the inverter hardware. This can be used to advantage in such installations. However, the inertia of a different inverter-connected system might be larger, either to different inverter software leading to a lower-bandwidth PLL (higher effective inertia), or due to high-inertia loads connected. Either of these two effects would significantly alter the results from those shown below, and potentially lead to sustained non-detection as shown in section 6.3.3.

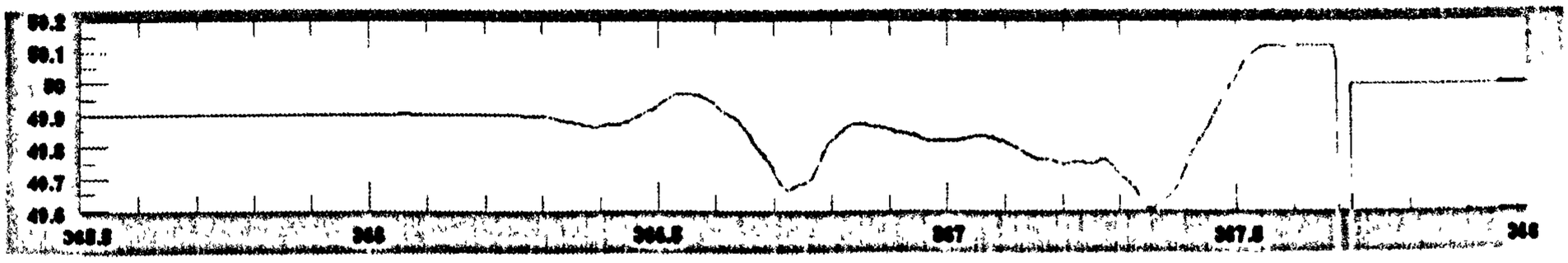


Figure 6-42 : Longest detection time using inverter-connected generation; Frequency (Hz)

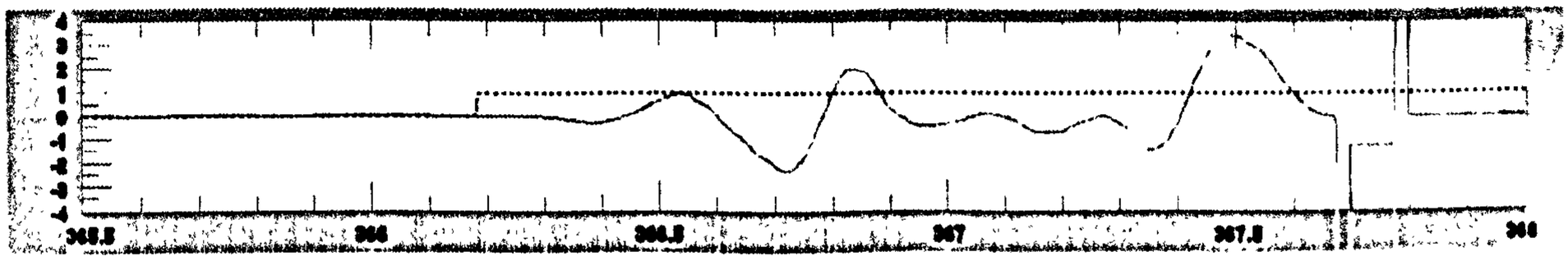


Figure 6-43 : detection time using inverter-connected generation; ROCOF (Hz/s) and instant of LOM inception (blue)

[1] A small amount of measured parasitic trapped load (8W in this case) was also accounted for

Figure 6-44 : detection time using inverter-connected generation; Phase Offset (degrees)

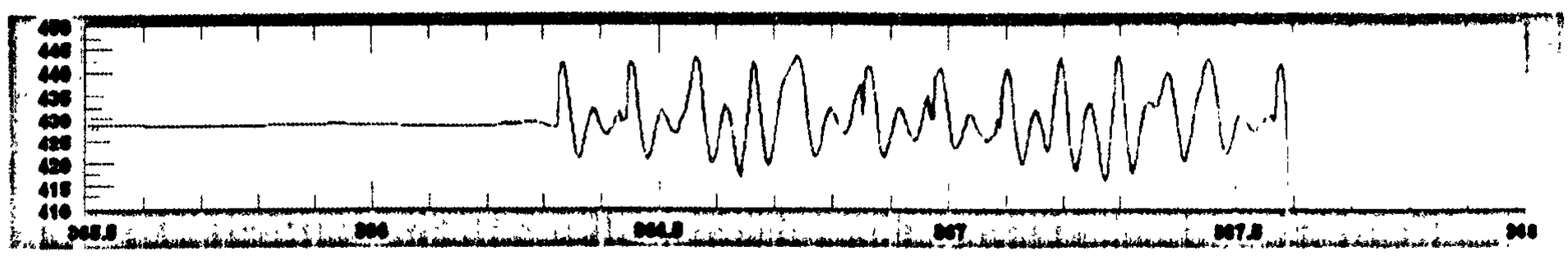


Figure 6-45 : detection time using inverter-connected generation; Voltage (+ve sequence fundamental, line-line)

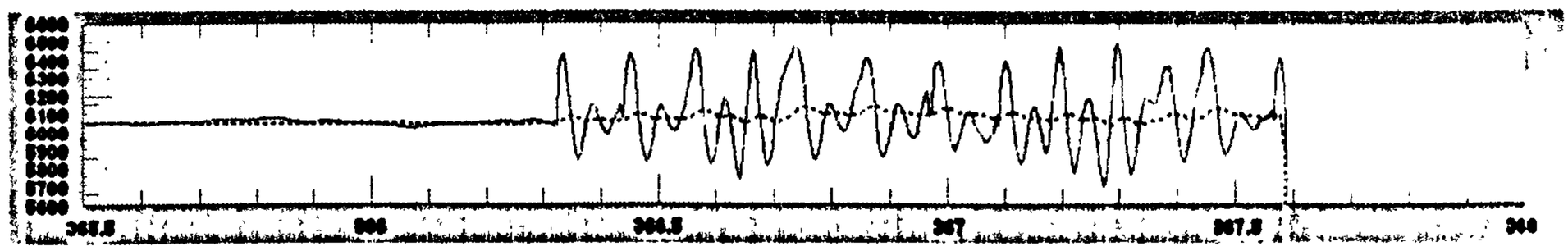


Figure 6-46 : detection time using inverter-connected generation; DG Real power output (solid) and target (dashed)

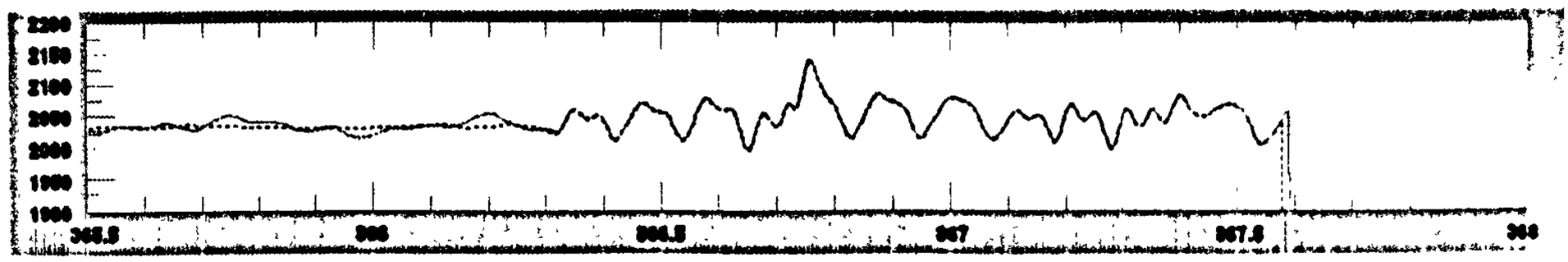


Figure 6-47 : detection time using inverter-connected generation; DG Reactive power output (solid) and target (dashed)

6.3.5 DG control strategy 2 - PQ control with almost no droop

These tests use the DG control strategy 2 of section 6.3.1.2. The P and Q set-point of the DG unit was manually set such that the real and reactive power exchange with the parent network was as close to zero as possible[1]. Since the target is a zero (or almost zero) power flow in the grid-connection branch, tuning the setup to achieve this is not heavily dependent upon exact calibration of any CTs, VTs, or sampling hardware. This is very helpful in achieving the balance accurately.

Using this control strategy, a case of sustained non-detection of LOM was demonstrated in the lab after only a few tries. The condition was sustained for >100 seconds until a small

---

[1] Actually, that the real power exchange was 20W to account for the expected trapped load, measured in section 6.3.3

load change was deliberately made. This caused very fast detection of LOM.

Strategy 2 is thus not recommended for DG installations where LOM must be detected using passive relays based upon voltage measurements. There are neither frequency/phase nor voltage excursions significant enough to enable the detection of LOM.

The graphs Figure 6-48 to Figure 6-53 below show logged data from the experiment. The loss-of-mains (LOM) event is instigated at t=236.5 seconds. Real power was matched within about 10W (<0.01pu) and reactive power was matched within about 10VAR (<0.01pu). At t=240 s, a LOM detection almost occurs, but the phase offset does not reach the trip level of 20° before the POR triggers are reset by a reversal of ROCOF. The undetected island, with the generator running at fixed P and Q outputs, was obviously not exactly balanced, because it settles to a new frequency of 49.2 Hz (down from 50.05 Hz) and voltage of 436V (down from 438V). The resulting state appeared to be perpetually stable in the laboratory, provided no changes to generator or load settings were made. A 150W load change was made at t=366.9 s. The POR tripped at t=367.4 s, 500ms after the small load change.

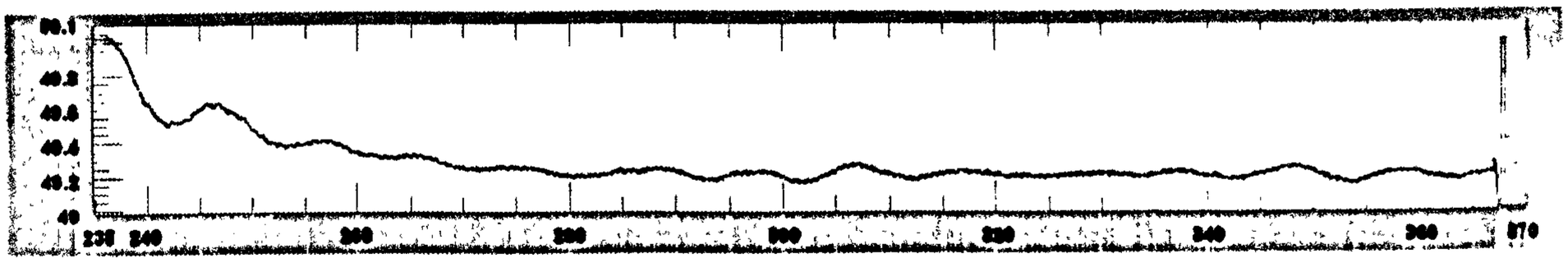


Figure 6-48 : Sustained non-detection of LOM; Frequency (Hz)

Figure 6-49 : Sustained non-detection of LOM; ROCOF (Hz/s) and instant of LOM inception (blue)

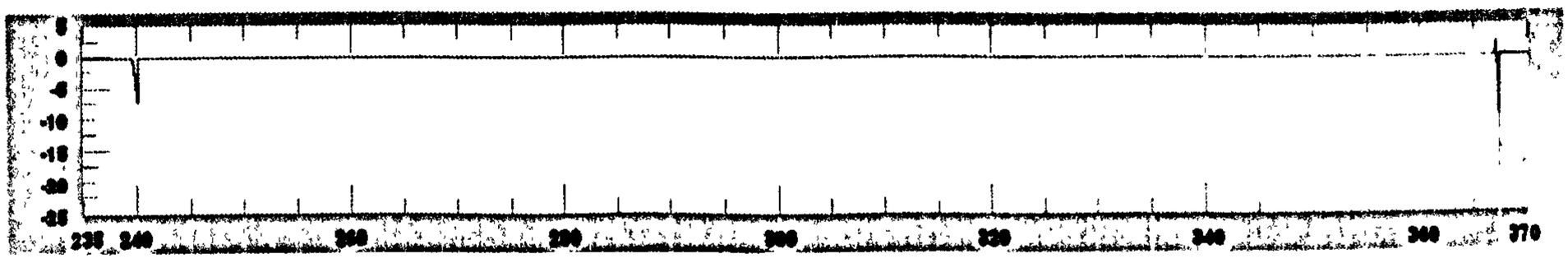


Figure 6-50 : Sustained non-detection of LOM; Phase Offset (degrees)

Figure 6-51 : Sustained non-detection of LOM; Voltage (+ve sequence fundamental, line-line)

Figure 6-52 : Sustained non-detection of LOM; DG Real power output (solid) and target (dashed)

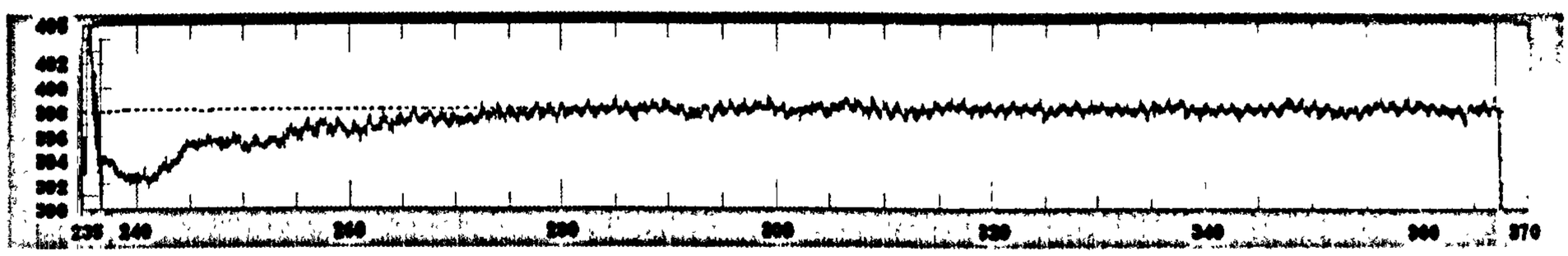


Figure 6-53 : Sustained non-detection of LOM; DG Reactive power output (solid) and target (dashed)

6.3.6 DG control strategy 3 - PQ control with 5% frequency and 40% voltage droop

These tests use the DG control strategy 3 of section 6.3.1.3. There is a 5% frequency droop and 40% voltage droop. The P and Q set-point of the DG unit was manually set such that the real and reactive power exchange with the parent network was as close to zero as possible[1]. This was quite hard to achieve in practice, because the parent network frequency and voltage were constantly changing, resulting in fluctuations to the generator output power (real and reactive) due to the action of the droop controls. During the tests performed, frequency often changed by several hundredths of a Hz during a few seconds. With the 5% frequency droop slope and a 1500W base power, a change of 0.03 Hz over 10 seconds results in a power adjustment of 0.012pu, or 18W. The voltage drifted by up to 1V (in 433) over 10 seconds, giving rise to reactive power adjustments of about 0.006 pu, or 9 VARs. Getting matches closer than this was difficult, but a number of attempts were made to get as perfect a match as possible.

21 attempts were made at achieving a perfect match. No sustained non-detections of LOM were noted. However, the longest trip time with the droop settings of 5% and 40% was 11.5 seconds although it will be seen in below that this event was a slight corner case, involving clipping of the generator output power which stopped the droop controls acting properly. The next highest trip times were 7.5 seconds and 5.5 seconds. The average trip time of the 20 trials was 3.7 seconds. The spread of trip times is shown in Figure 6-54.

---

[1] Accounting for the measured 20W parasitic trapped load

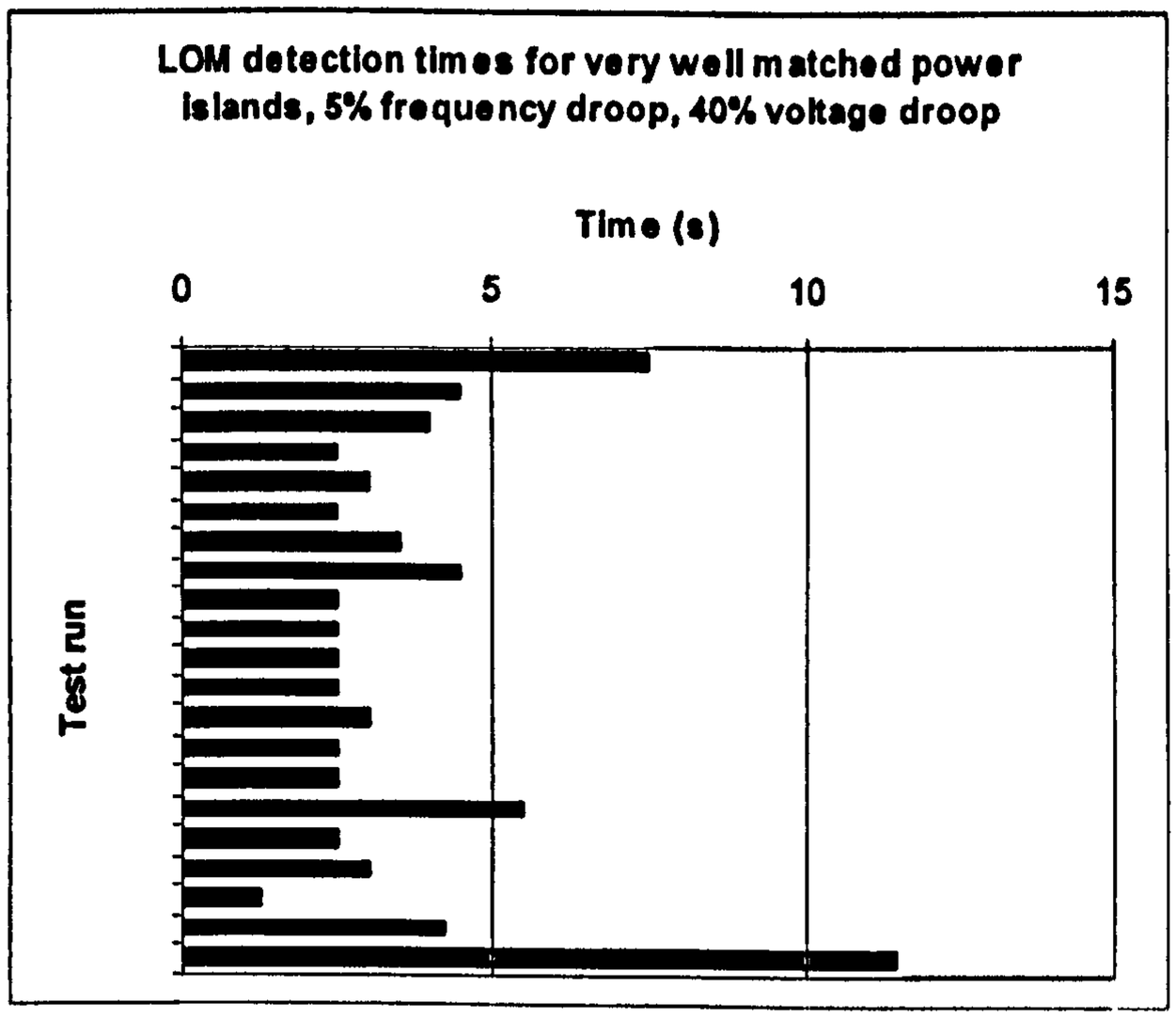


Figure 6-54 : LOM detection times for very well matched power islands, 5% frequency droop, 40% voltage droop

Strategy 3 is thus a significantly better control method than strategy 1 or 2, for DG installations where LOM must be detected using passive relays based upon voltage measurements. The LOM event will always be detected, but the detection time will be up to 12 seconds as suggested by laboratory testing. Although unlikely, the detection time might even be more than 12 seconds for extremely well balanced events, but only if the loads and DG set-points are extremely invariant.

An example of a slow detection with a trip time of 5.5 seconds is shown below. Real power was almost perfectly matched. Reactive power was matched to within 15 VARs (<0.01 pu). The LOM condition was instigated at t=919.7 s. Trip did not occur until t=925.2 s. The natural frequency of the system instability is approximately 0.31 Hz, shown by a half-period of about 1.6 seconds evident in Figure 6-59. This shows fair corroboration with the prediction made in section 6.3.1.3, which predicted instability with a natural frequency of between DC and 0.45Hz, based upon the P control system being unstable and the Q control system being marginally stable with a voltage droop of 40%. This provides evidence that the models produced in section 6.3.1.3 provide a fair means of analysing the system.

Figure 6-55 : 5.5s trip time with 5% frequency and 40% voltage droop; Frequency (Hz)

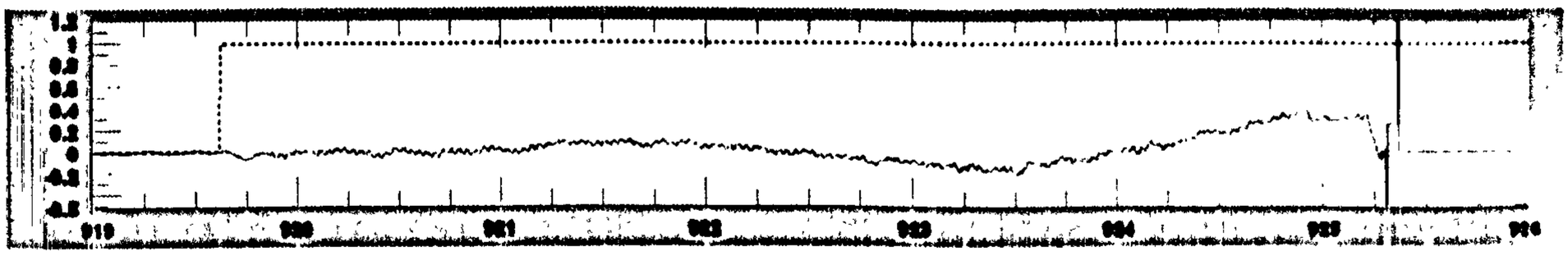


Figure 6-56 : 5.5s trip time with 5% frequency and 40% voltage droop; ROCOF (Hz/s) and instant of LOM inception (blue)

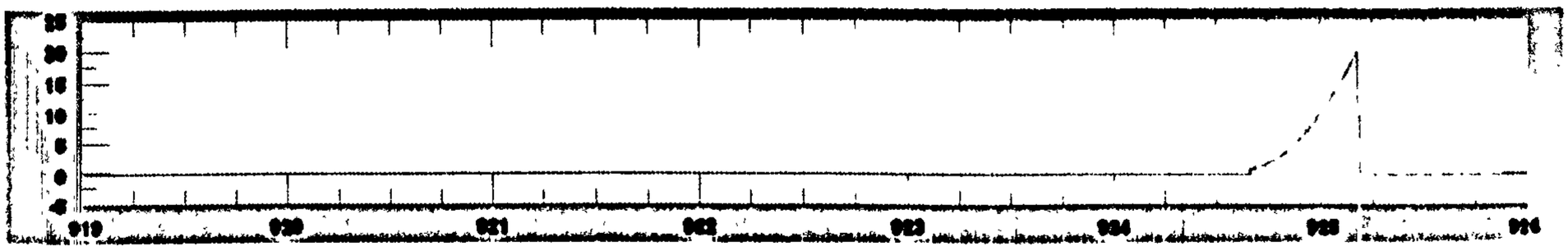


Figure 6-57 : 5.5s trip time with 5% frequency and 40% voltage droop; Phase Offset (degrees)

Figure 6-58 : 5.5s trip time with 5% frequency and 40% voltage droop; Voltage (+ve sequence fundamental, line-line)

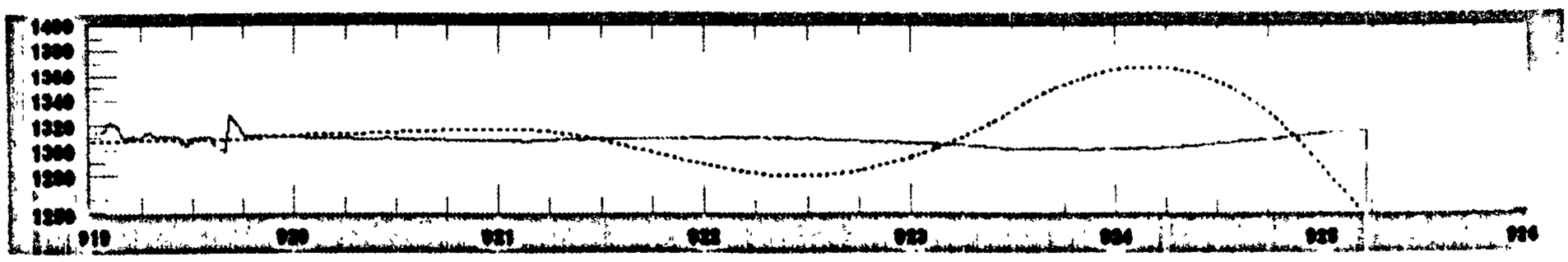


Figure 6-59 : 5.5s trip time with 5% frequency and 40% voltage droop; DG Real power output (solid) and target (dashed)

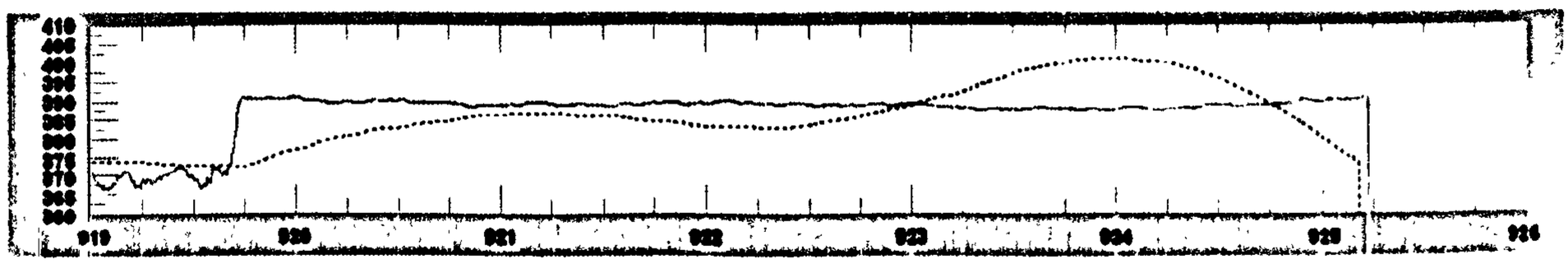


Figure 6-60 : 5.5s trip time with 5% frequency and 40% voltage droop; DG Reactive power output (solid) and target (dashed)

The example with a trip time of 11.4 seconds is shown below. Real power was almost perfectly balanced, while reactive power was balanced within about 10 VARs (<0.01pu). The LOM condition was instigated at t=743.3 s. Trip did not occur until t=754.7 s. This example is an interesting corner case. In this test, the local load power was extremely close to the rated output power of the generator (1500W). Figure 6-65 shows that the generator real power control (throttle) was clipped to 1500W (1pu). This diminished the action of the droop control feedback loop and consequently, the instability of the system was less and the trip time took longer that it would otherwise have done.

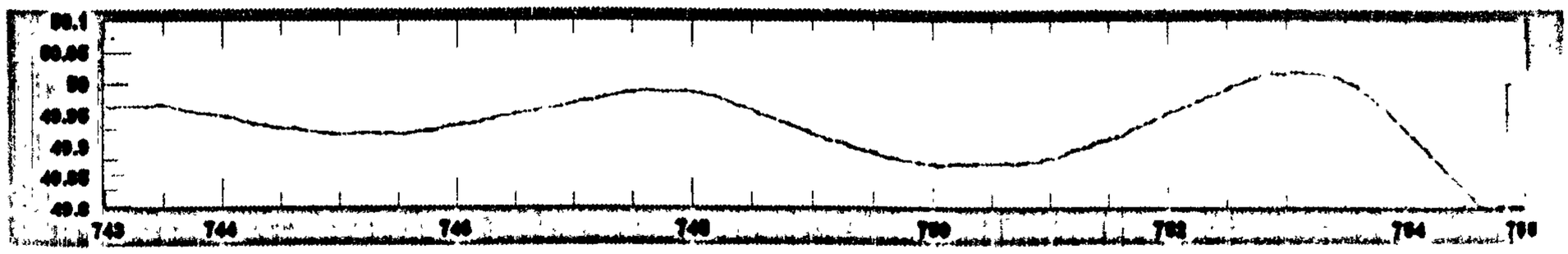


Figure 6-61 : 11.4s trip time with 5% frequency and 40% voltage droop; Frequency (Hz)

Figure 6-62 : 11.4s trip time with 5% frequency and 40% voltage droop; ROCOF (Hz/s) and instant of LOM inception (blue)

Figure 6-63 : 11.4s trip time with 5% frequency and 40% voltage droop; Phase Offset (degrees)

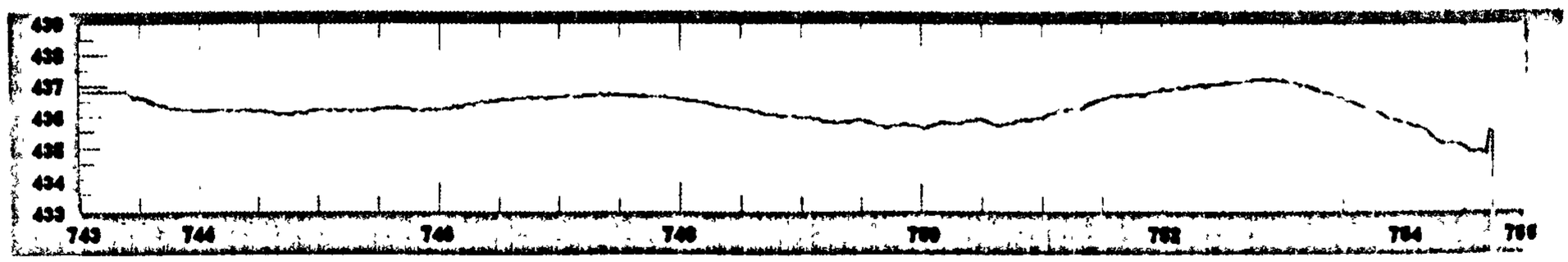


Figure 6-64 : 11.4s trip time with 5% frequency and 40% voltage droop; Voltage (+ve sequence fundamental, line-line)

Figure 6-65 : 11.4s trip time with 5% frequency and 40% voltage droop; DG Real power output (solid) and target (dashed)

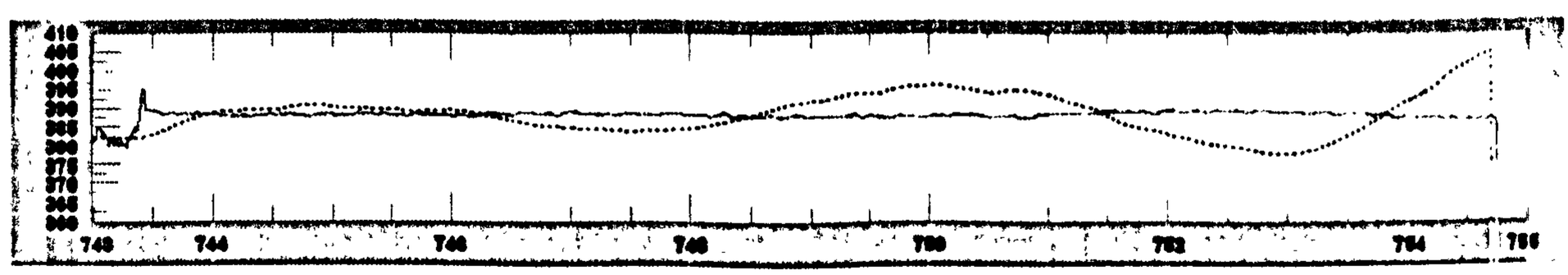


Figure 6-66 : 11.4s trip time with 5% frequency and 40% voltage droop; DG Reactive power output (solid) and target (dashed)

6.3.7 DG control strategy 4 - Real power match and 10% voltage droop, guaranteed LOM NDZ avoidance using non-zero VAR exchange

The performance of control strategy 4 (see section 6.3.1.4) in combination with the POR (designed in section 6.1.1) was tested many times, with some different combinations of settings relevant to traditional and microgrid applications.

These tests show that the combination of strategy 4 with the use of the POR is a good way of controlling the DG unit, and guaranteeing LOM detection within a prescribed timeframe. The target timeframe in this case is 2 seconds (IEEE 1547, 2003). To achieve the target timeframe, the ROCOF trigger setting and power flow thresholds (see section 6.3.1.4) must be set appropriately.

The first set of tests use a more conventional ROCOF trigger setting of 0.2 Hz/s for the POR (together with the trip setting of 20° which was used for all tests). In this case, the DG control strategy was set to avoid a VAR exchange with the parent network of smaller than ±0.05pu; with 1pu reactive power being 2000 VAR, this equates to ±100 VAR. This configuration represents a good solution for operating a microgrid in grid-connected or islanded mode when its parent network is the national grid, which has relatively good frequency stability.

The second and third sets of tests use ROCOF trigger settings of 1 Hz/s and 0.5 Hz/s. This significantly lowers the sensitivity of the POR. The reason that this step may need to be taken is to accommodate grid-connection to a parent network significantly smaller than the national grid. For example, as Table 2-1 showed, a 100kVA network will regularly achieve ROCOF rates of up to 0.4 Hz/s simply with 3kW load steps. Larger load steps would produce larger ROCOF rates, even though they may only be transitory events until prime movers respond. In the second and third sets of tests, to mitigate the de-sensitisation of the LOM triggering, the control strategy was adjusted to avoid a VAR exchange with the parent network of smaller than ±0.1pu (±200 VAR). This tends to initially de-stabilise the local power system twice as much as the ±0.05pu offset used in the first tests.

The procedure for these tests is slightly different than for the previous sections. The

active power balance is achieved automatically by the automatic control algorithm[1]. The AVR droop control imports or exports VARs in a manner which provides voltage support. During these tests, the parent network voltage was about 430V RMS line-line. The degree of reactive power balance was thus changed between the test runs by altering the voltage which the AVR regards as its nominal target. This alters the generator reactive power output via the 10% voltage droop slope. Only the novel non-detection-zone (NDZ) avoidance algorithm prevented a more exact balance being achieved, which would have led to some long trip times similar to those of Figure 6-54.

6.3.7.1 Detection times: 0.2 Hz/s ROCOF trigger setting, >±0.05pu VAR exchange

16 test runs were completed, with active power exactly balanced. As hoped, the detection times using these settings were all less than 2 seconds. Figure 6-67 shows the results, tabulated as detection times versus VAR import from the parent network. The VAR import/export from the parent network is never smaller than 0.05 pu due to the action of the LOM NDZ avoidance strategy. Clearly, detection time peaks at about 1.5 seconds for the most closely matched events, but decreases if the VAR exchange is bigger than 0.05 pu. The scatter of detection times between 1.1 and 1.5 seconds, for the same 0.05pu VAR import/export conditions, is due to tiny random fluctuations in the actual hardware test conditions (frequencies, power flows, throttle responses, phase angles, and measurements) which lead to relatively larger detection time variations due to the unstable nature of the system (see Figure 6-55 to Figure 6-60). This effect is similar to that of the difficulty in making accurate weather forecasts due to the way that small local variations can lead to large effects on wide areas over the following days.

---

[1] Taking into account the additional 20W of parasitic trapped load which will be acquired by the microgrid when the loss-of-mains (LOM) event occurs

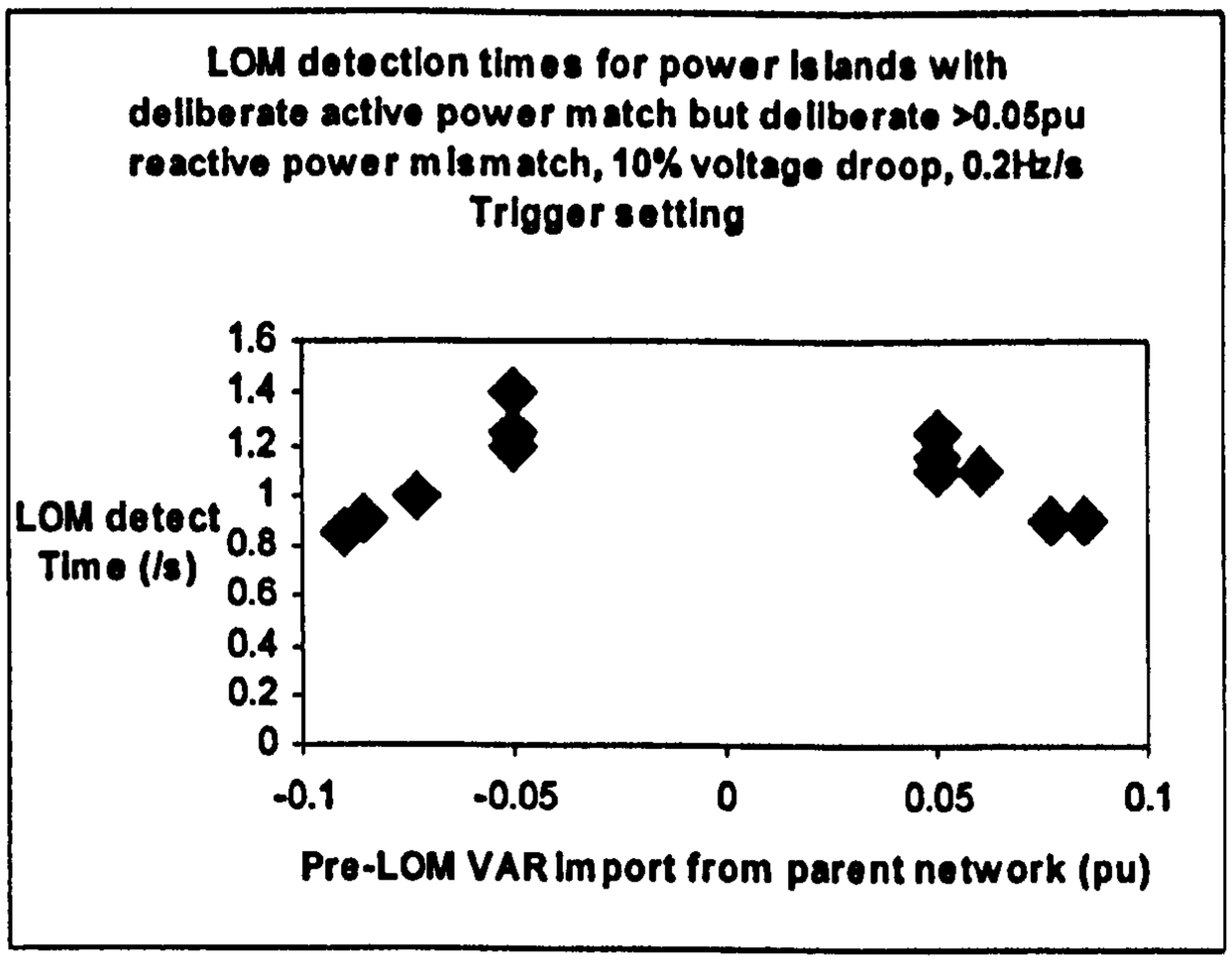


Figure 6-67 : LOM detection times for power islands with deliberate active power match but deliberate >0.05pu reactive power mismatch, 10% voltage droop, 0.2Hz/s Trigger setting

These detection times are all acceptable, being below the 2 second limit given by IEEE 1547 (IEEE, 2003) and quicker than an expected auto-reclose action. The maximum trip time using the proposed NDZ avoidance strategy (strategy 4) is about 1.5 seconds, compared to trip times up to 11 seconds which were possible without the NDZ avoidance strategy. The reduction in maximum trip time is achieved because the NDZ has been correctly avoided by strategy 4. This combination of control strategy and relay settings is therefore appropriate and robust for use within scenarios where the parent network is very large, and expected ROCOF rates due to normal network frequency deviations are less than 0.2 Hz/s, and the total local inertia (DG plus loads) is approximately H=1 pu.

6.3.7.2 Detection times: 1.0 Hz/s ROCOF trigger setting, >±0.1pu VAR exchange

In these tests, the ROCOF trigger setting was set very wide, at 1.0 Hz/s. This would allow, for example, a 6-7kW load step within a 100kVA parent network, to which an even smaller microgrid was grid-connected, without causing a LOM trip (see Table 2-1). 22 test runs were completed. The results are shown in Figure 6-68. The control strategy in this case is set to avoid VAR exchanges of less than 0.1 pu. The test runs which have VAR exchange magnitudes larger than 0.1 pu cause LOM to be detected in less than about 2 seconds. However, some of the test runs which only had a 0.1 pu VAR exchange before the LOM occurred, produced detection times of up to 8 seconds. In some of these runs, the POR was never triggered at the 1 Hz/s level, and therefore could not trip. In these cases the final trip was due to under-frequency or over-frequency at the 47 and 52 Hz levels, which

is not ideal. These trip times are unacceptably long, considering the risk of auto-reclose action and the potential stress damage to contactors or machines.

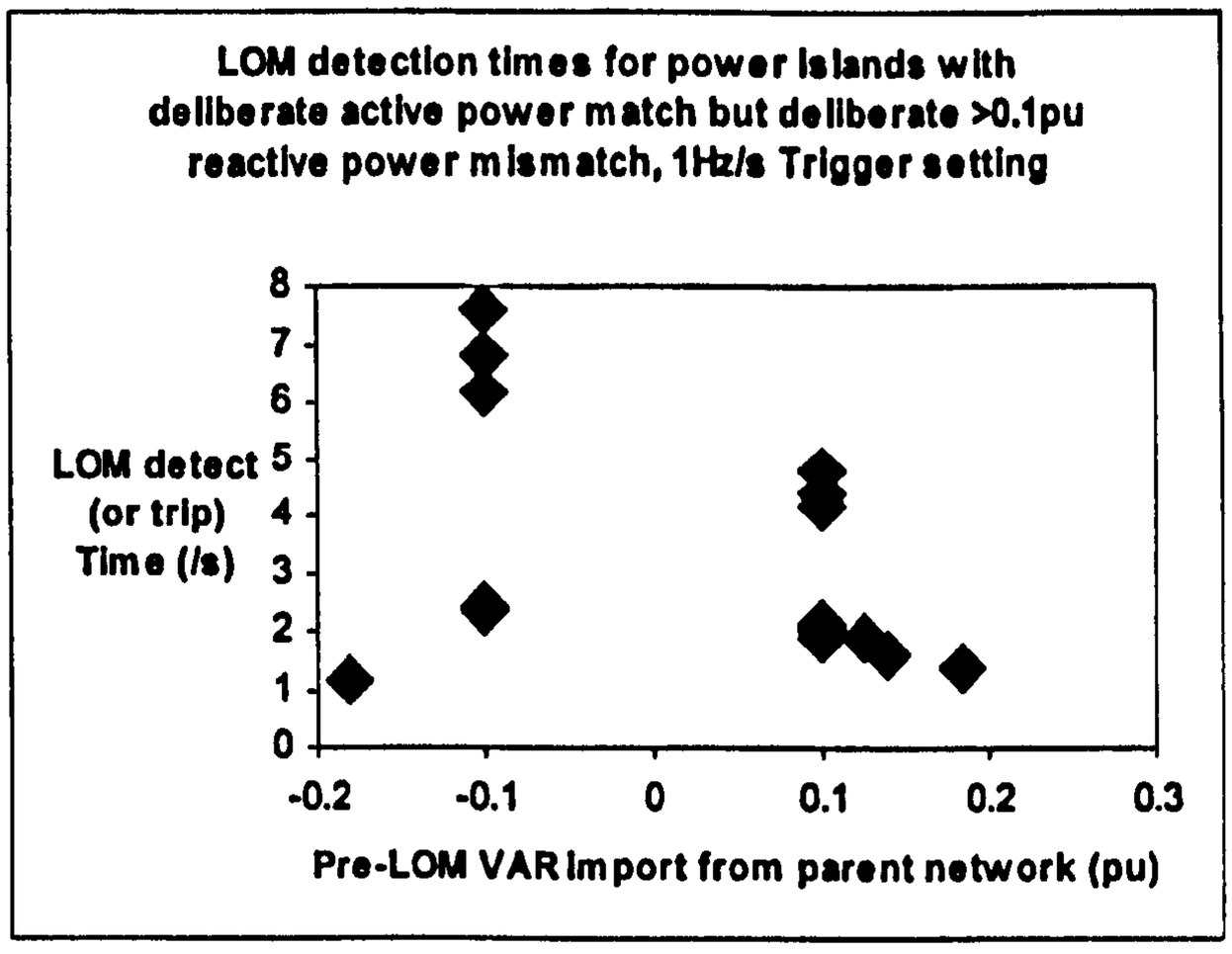


Figure 6-68 : LOM detection times for power islands with deliberate active power match but deliberate >0.1pu reactive power mismatch, 1Hz/s Trigger setting

This indicates, that at a trigger setting of 1 Hz/s, the guaranteed minimum VAR exchange with the parent network should be larger then 0.1 pu, in order to avoid the NDZ and guarantee tripping within 2 seconds.

6.3.7.3 Detection times: 0.5 Hz/s ROCOF trigger setting, >±0.1pu VAR exchange

Following on from the above test, the minimum VAR exchange with the parent network was held at ±0.1 pu, but the ROCOF trigger setting for the POR was reduced to 0.5Hz/s. This would allow, for example, a 7.5kW load step within a 200kVA parent network, to which an even smaller microgrid was grid-connected, without causing a LOM trip (extrapolated from the 100kVA entry in Table 2-1). If the parent network rating was at least 1MVA (with a rotating generator), then routine load steps of up to about 50kW could be made without tripping off smaller microgrids, by exceeding the 0.5 Hz/s ROCOF trigger level (again from Table 2-1). These maximum allowed load steps are 4-5% of the network capacity, with an assumed total inertia (generators and loads) of H=2pu.

So, assuming that genuine ROCOF levels within the parent network are almost always less than 0.5Hz/s, using a minimum VAR exchange of ±0.1 pu, the LOM detection times are as shown below.

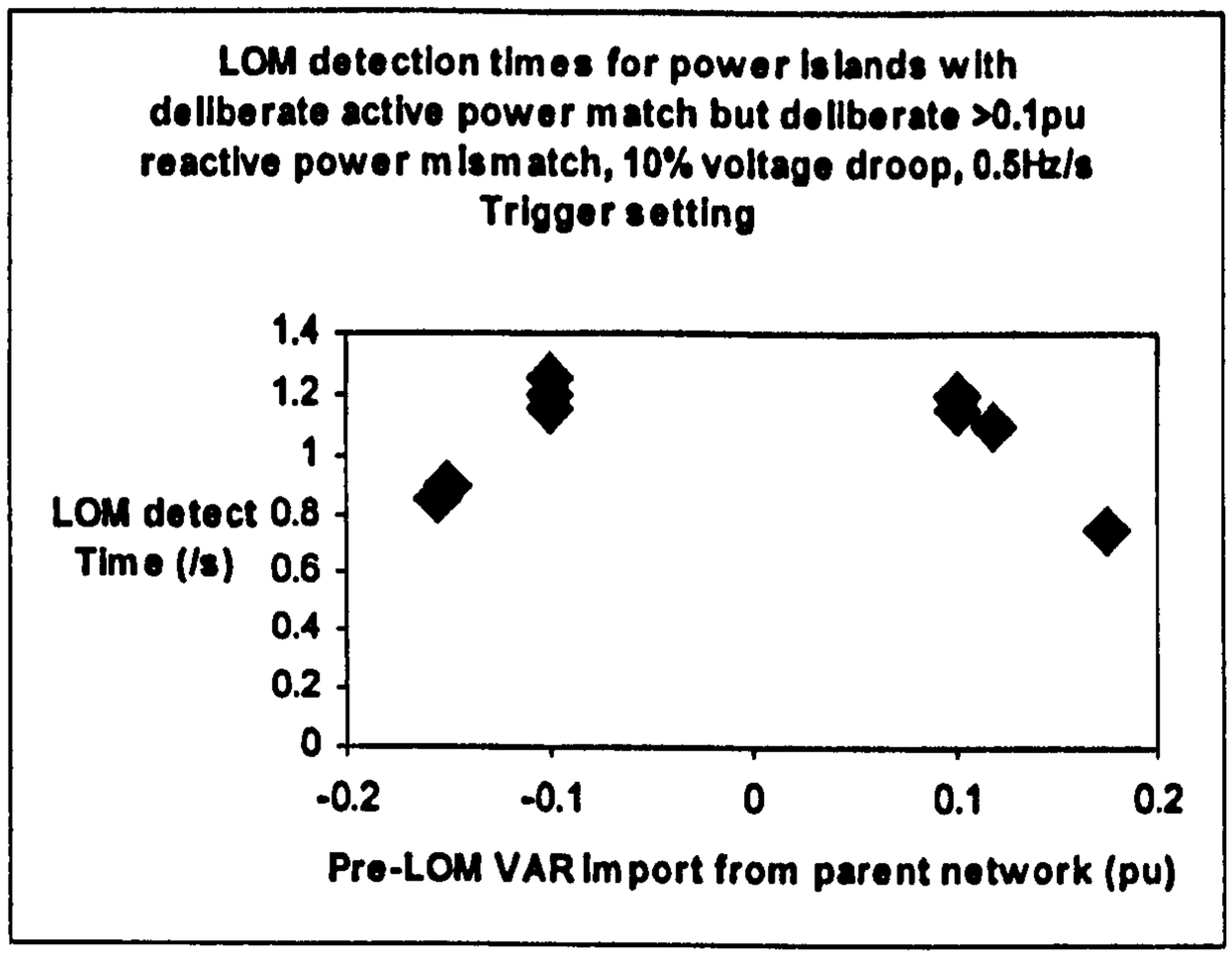


Figure 6-69 : LOM detection times for power islands with deliberate active power match but deliberate >0.1pu reactive power mismatch, 0.5Hz/s Trigger setting

15 test runs are shown in Figure 6-69, although some of the datapoints with trip times of about 1.2 seconds overlie each other, giving the impression that only 9 test runs were completed. The longest detection time was 1.25 seconds, an acceptable figure, below the 2 second limit given by IEEE 1547 (IEEE, 2003) and quicker than an expected auto-reclose action. The detection times where the VAR exchange was higher than 0.1 pu are shorter. This combination of control strategy and relay settings is therefore appropriate for use within microgrid scenarios where the parent network is significantly smaller than the national grid, but the maximum step load change is restricted to approximately 4-5% of the parent network capacity, the parent network system inertia is approximately 2pu, and the local system inertia (DG plus loads) is approximately H=1 pu..

It is also interesting to note that detection times of about 1.2 seconds in Figure 6-69 compare to similar detection times in Figure 6-67. This is after a 2.5x increase in the ROCOF trigger threshold, from 0.2 Hz/s to 0.5 Hz/s, and a 2x increase in the VAR exchange. This demonstrates that even small VAR exchanges help to significantly destabilise the grid-connected power network when it experiences a LOM event, and that the magnitude of the de-stabilising effect is at least linearly related to the VAR flow.

The logs from the longest detection event of Figure 6-69 are shown below. The effect of the deliberate Pre-LOM 0.1 pu reactive power exchange with the parent network can be seen in Figure 6-75, as the generator reactive power output drops by 0.1 pu (200 VAR) when the LOM event occurs. Subsequent to the LOM occurring, the voltage within the

power island then rises as the generator controls try (but fail) to re-attain the target VAR flow. The rising voltage causes an increase in real power demand from the resistive loads, which drags frequency downwards. Due to the real power matching algorithm of control strategy 4, which is used in this test, the generator real power target is also gradually increased. This is caused by integral control action of any actual power flow to the parent network (which in this case is just slightly off zero due unaccounted-for trapped parasitic load). This has a mitigating effect on the rate of the frequency excursion but the DG control response is lagged due to the control filters and integral action, hence the frequency excursion is still negative. There is no frequency restoration control or frequency droop applied, and so the resulting system is unstable both in terms of frequency and voltage.

Figure 6-70 : 1.25s trip time with automatic NDZ avoidance; Frequency (Hz)

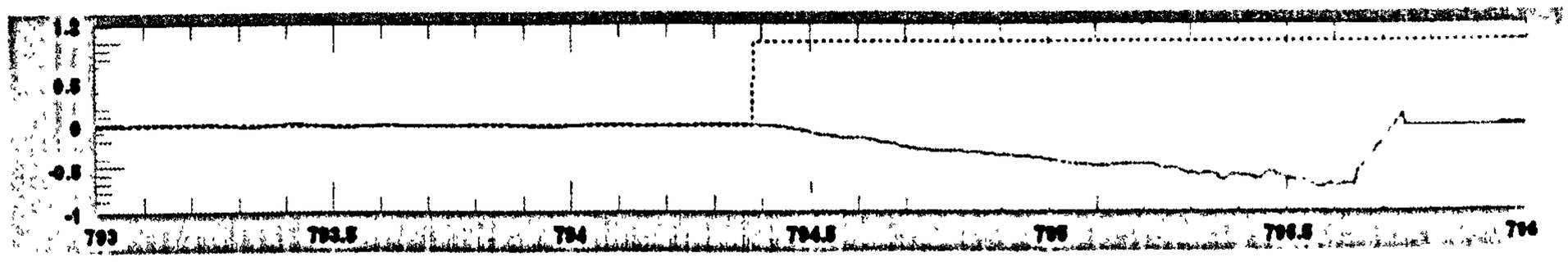


Figure 6-71 : 1.25s trip time with automatic NDZ avoidance; ROCOF (Hz/s) and instant of LOM inception (blue)

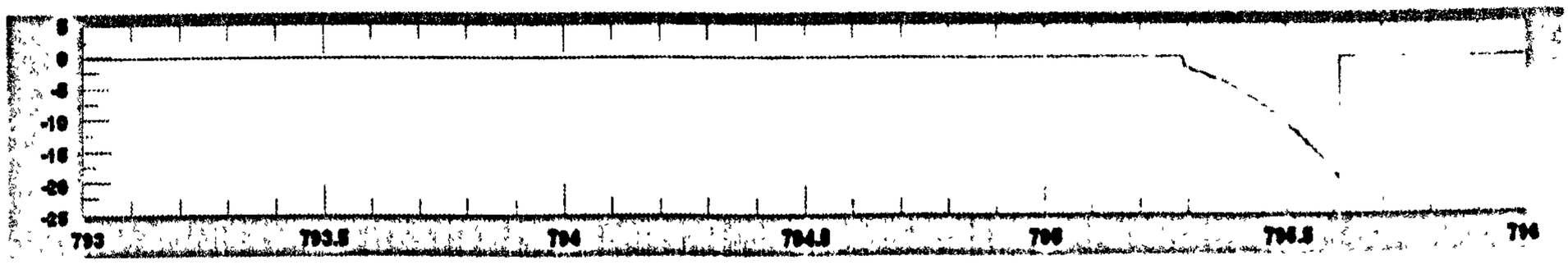


Figure 6-72 : 1.25s trip time with automatic NDZ avoidance; Phase Offset (degrees)

Figure 6-73 : 1.25s trip time with automatic NDZ avoidance; Voltage (+ve sequence fundamental, line-line)

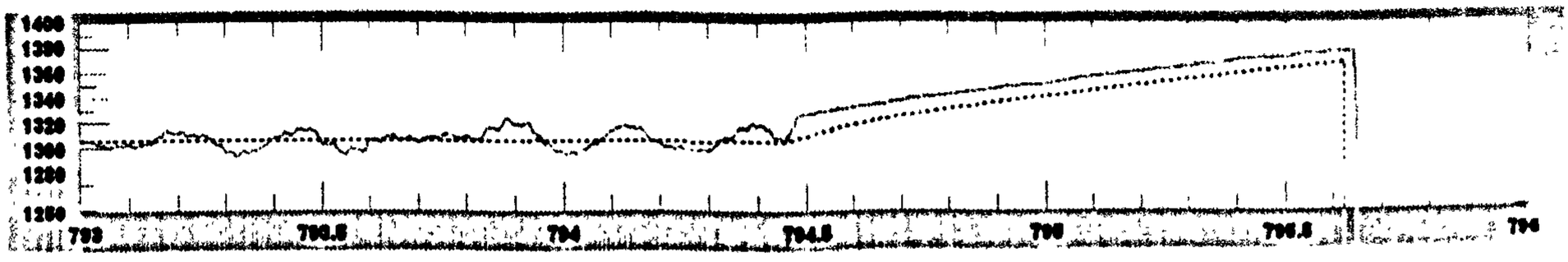


Figure 6-74 : 1.25s trip time with automatic NDZ avoidance; DG Real power output (solid) and target (dashed)

Figure 6-75 : 1.25s trip time with automatic NDZ avoidance; DG Reactive power output (solid) and target (dashed)

6.3.8 Summary findings and further work opportunities resulting from hardware testing

The primary conclusions from this section are:-

- 4 different strategies (with active and reactive power set-points) have been compared for grid-connected generator control, to see which ones resulted in the most effective detection of loss-of-mains.
- The two worst grid-connected control algorithms examined have fixed real and reactive power outputs (zero droop) or active matching of both real and reactive power within the local power system. Examples of sustained non-detection of loss-of-mains were created in the laboratory, using both these control methods.
- Adding 5% frequency droop and 10-40% (voltage) reactive power droop controls (containing appropriate low-pass filtering and integral control within the control algorithms, and lags within the prime mover hardware and generator field), creates an unstable system if the local power system is accidentally placed in islanded mode. However, detection times of between 5 and 12 seconds can still be demonstrated, by creating very close matches of DG output to local load demand (both real and reactive), and placing the local power system with the non-detection-zone (NDZ) of the loss-of-mains detection relay.
- Changing the active power control to an active balancing of real power between local loads and generation, in conjunction with a 10% (voltage) reactive power droop control, also creates an unstable system in islanded mode, although long detection times can still result.
- Further, a novel algorithm can be added which insists on a non-zero (at least ±0.05pu to ±0.1pu) reactive power exchange with the parent network, if the real power exchange is almost zero. This can be used to reduce the maximum detection time from >10 seconds to <2 seconds which is an acceptable figure even in systems with auto-reclose breakers. The active and reactive power thresholds can be set appropriately to meet either the 2 second (IEE 1547 (IEEE, 2003)) or 1 second (ETR 113 (ENA, 1995)) requirements for LOM detection.

- Suitable settings for the relay trigger threshold depend upon the size, and inertia, of the parent network to which a microgrid is connected, the maximum permitted step load within the parent network, and the inertia of the local DG/loads. Table 2-1 in conjunction with the results of sections 6.3.7.1 to 6.3.7.3 can be used as a guide for suitable trigger levels, scaled proportionately if necessary.
- If the parent network is a large network such as the national grid, and the local DG/loads have a total inertia of approximately H=1pu, then suitable settings for the relay are a 0.2 Hz/s trigger threshold and a 20° trip threshold (temporarily widening to 100° during balanced or unbalanced faults, as described in section 6.1). In this case either the real or reactive power exchange with the parent network should be kept at a magnitude >0.05pu to guarantee detection of LOM within 2 seconds.
- If the parent network is of the order of 1MVA capacity, with inertia of approximately H=2pu and a maximum load step of 50kW, and the local DG/loads have a total inertia of approximately H=1pu, then suitable settings for the relay are of the order of a 0.5 Hz/s trigger and a 20° trip signal (temporarily widening to 100° during balanced or unbalanced faults, as described in section 6.1). In this case either the real or reactive power exchange with the parent network should be kept at a magnitude >0.1pu to guarantee detection of LOM within 2 seconds. The trigger threshold can be widened further from 0.5 Hz/s if required, to avoid spurious tripping when routine load/generator switching causes ROCOF events larger than 0.5 Hz/s. In this case, the minimum real/reactive power imbalance should also be increased proportionately from 0.1pu, to guarantee LOM detection within 2 seconds.
- Detection times tend to be significantly reduced where inverter-connected distributed generators are installed, due to the low "inertia" of the PLL. However, this statement is dependent upon inverter software design and also any inertia of connected local loads. These are installation dependent. For example, inverters might in future contain artificial inertia within the PLLs to aid fault ride-through etc..

Further work opportunities include:-

- Enhancing an opportunity from section 6.1.5: design and test an algorithm to automatically adapt not only the ROCOF triggering level $R_{Trigger}$ for the POR, but also the minimum allowed real/reactive power imbalances P_t & Q_t within the control algorithm, for different scenarios/size of grid during islanded operations. The adaptive algorithm would need to monitor (at least) frequency deviations due

to "normal" network behaviour.

- To add significant spinning loads with inertia, to simulate an industrial environment, and verify whether the detection times are changed significantly, for the same control algorithm settings and relay settings.
- Du (2005) and Ye (2006) propose the addition of positive feedback to the P and Q controls (throttle and field) based upon differential filtered or bandpass filtered measurements of the measured system voltage and frequency. These systems as presented are undesirable since they reduce or remove the droop controls to enable network support. If the droop controls used in this thesis could be combined with the positive feedback differential terms from Du (2005) or Ye (2006), faster detection times might result for the same small P or Q imbalances.
- To examine more combinations of ROCOF trigger threshold, minimum power exchange thresholds, and DG/load inertias, to derive a theoretical or empirical equation tying these parameters together with a given maximum LOM detection time limit.

6.4 References for chapter 6

ADI. (2008). Applied Dynamics International. [Online]. [Accessed 14/03/2008]. < http://www.adi.com/>

Affonso, C. et al. (2005). Performance of ROCOF relays for embedded generation applications. IEE Proceedings on transmission and distribution, vol. 152 no.1, pp. 109-114.

Areva T&D. (2007). Network Protection and Automation Guide. [Online]. [Accessed 26/11/2007]. <http://www.areva-td.com/scripts/solutions/publigen/content/templates/show.asp?P=1228&L=US> Chapter 14

Dysko, A., Burt, G. & Bugdal, R. (2006). Novel Protection Methods for Active Distribution Networks with High Penetrations of Distributed Generation - Year 2 Report. DTI/CDGSEE/TR/2006-7. [Online]. [Accessed 14/03/2008]. <http://www.sedg.ac.uk/DGSEE Protection Project - Year 2 Report.pdf>

Dysko, A. et al. (2007). Reducing unnecessary disconnection of renewable generation from the power system. IET Renewable power generation, vol. 1 no. 1, pp 41-48.

Dysko, A. et al. (2007b). UK distribution system protection issues. IET Generation, transmission and distribution, vol. 1 no. 4, pp 679-687.

Du, P., Nelson, J. & Ye, Z. (2005). Active anti-islanding schemes for synchronous-machine-based distributed generators. IEE Proceedings on generation, transmission and distribution, vol. 152 no. 5, pp 597-606.

ENA. (1991). ER G59/1. Recommendations for the connection of embedded generating

plant to the public electricity suppliers' distribution system.

ENA (1995). ETR 113. Notes for guidance of the protection of embedded generation plant up to 5MW for operation in parallel with public electricity suppliers' distribution systems.

Freitas, W. et al. (2005). Comparative analysis between ROCOF and Vector Surge relays for distributed generation applications. IEEE Transactions on power delivery, vol. 20 no. 2, pp. 1315-1323.

Huang, S. & Pai, F. (2001). Design and operation of grid-connected photovoltaic system with power-factor control and active islanding detection. IEE Proceedings on transmission and distribution, vol. 148 no. 2, pp 243-250.

IEEE. (2003). IEEE 1547, IEEE Standard for interconnecting distributed resources with electric power systems.

Lokov, G. & Finstad, R. (2005). Active anti-islanding system for an electric power plant and his method. European Patent no. EP 1 574 672 A2.

Mullane, A., Bryans, G. & O'Malley, M. (2005). Kinetic energy and frequency response comparisons for renewable generation systems. Conference on future power systems. Paper 008-01.

Roscoe, A. (2008). Power flow management to avoid the non-detection of loss of mains. UK patent application 0810512.4 filed 10 June 2008.

RTDS. (2008). RTDS technologies. [Online]. [Accessed 14/03/2008]. <http://www.rtds.com/index.html>

SEDG. (2008). United Kingdom Generic Distribution System (UKGDS). [Online]. [Accessed 14/03/2008]. <http://www.sedg.ac.uk>

Sumner, M., Palethorpe, B. & Thomas, D. (2004a). Impedance measurement for improved power quality - Part 1: The measurement technique. IEEE Transactions on power delivery, vol. 19 issue 3 pp 1442-1448.

Sumner, M., Palethorpe, B. & Thomas, D. (2004b). Impedance measurement for improved power quality - Part 2: A new technique for stand-alone active shunt filter control. IEEE Transactions on power delivery, vol. 19 issue 3 pp 1457-1463.

Timbus, A. et al. (2004). Online grid measurement and ENS detection for PV inverter running on highly inductive grid. IEEE power electronics letters, vol. 2 no. 3, pp 77-82.

Vieira, C. et al. (2006a). Performance of frequency relays for distributed generation protection. IEEE Transactions of power delivery, vol. 21 no. 3, pp 1120-1127.

Vieira, C. et al. (2006b). Efficient coordination of ROCOF and frequency relays for distribution generation protection by using the application region. IEEE Transactions of power delivery, vol. 21 no. 4, pp 1878-1884.

Vieira, C. et al. (2006c). Formulas for predicting the dynamic performance of ROCOF relays for embedded generation applications. IEE Proceedings on generation transmission and distribution, vol. 153 no. 4, pp 399-406

Wall, S. (2004). Detection of islanded behaviour and anti-islanding protection of a generator in grid-connected mode. US Patent no. US 2004/0178641 A1.

Ye, Z. et al. (2004). Evaluation of anti-islanding schemes based on nondetection zone concept. IEEE Transactions on power electronics, vol. 19 no. 5. pp 1171-1176.

Ye, Z. et al. (2006). Anti-islanding protection systems for synchronous machine based distributed generators. US Patent no. US2006/0082936 A1

7 Conclusions

The work of this thesis enables an increase in the security of supply within microgrid scenarios using distributed generation. This is accomplished by addressing two identified gaps in established knowledge.

1) Accurate, timely measurement of amplitude/phase/frequency with low frame rates within power systems experiencing poor power quality. The achievement of this goal allows many simultaneous measurement and control functions to be integrated on a single cheap microcontroller at an appropriate frame rate, suitable for ubiquitous deployment with small-scale generation at low incremental cost. The resulting distributed control algorithms can be used to enhance security of supply by offering the possibility of network reconfiguration, frequency support, ancillary services provision, and/or deliberate islanding.
2) Reliable, timely detection of Loss-Of-Mains (LOM) when local real power generation is deliberately balanced to local real power demand, while avoiding spurious (nuisance) tripping due to switching, noise, harmonics, and network faults. The ability to detect LOM reliably, despite a match of active power within the microgrid, allows a microgrid controller to switch quickly to islanded mode with a minimal frequency and voltage excursion following a LOM event, which increases the security of supply at the local level.

To accomplish the main overall goals, a comprehensive system-level study of the requirements to be met and the potential scenarios has been carried out. The requirements to be met are driven by required control action times (latencies) and required measurement accuracies for control and relaying purposes. A large emphasis is placed upon measurement ripple magnitude, since measurement ripple can cause power system oscillations when fed back into the system via control action. The system requirement study brings together information from a number of sources. Some data is directly available from applicable standards, but much is calculated from potential worst-case scenarios in non-standard applications such as rural, islanded, battleground or disaster-relief scenarios.

The first conclusion of the requirement study is that rates of change of frequency within microgrid scenarios can regularly be much higher than seen normally within the UK power grid. This presents problems for many previously published frequency measurement algorithms. A second major conclusion is that harmonic distortion on voltages could in theory reach 53% THD in weak power systems containing predominantly lighting and

computer/IT loads. Operating any power system in such a condition is undesirable for any length of time, but this upper limit is used throughout this thesis as a test case for candidate measurement algorithms to ensure robust operation. The potential magnitudes of other "influence qualities" such as inter-harmonics, flicker, noise, unbalance etc. are also calculated and used to generate suitable test waveforms. These test waveforms are invaluable for verification of the performance of the measurement algorithms proposed in this thesis, and could also be used for other projects to test other measurement/control algorithms.

To measure AC system parameters, the use of exact-time averaging is proposed by this thesis as the single most powerful and applicable building block. Only one previous work concerning AC signal measurements applies such a technique, and in that work the technique is simply used as a block and not analysed. This thesis performs an in-depth analysis of the properties of such an algorithm. The practical difficulty lies in averaging a signal over an exact timeframe which may not be an integer multiple of the sample interval. A version of an algorithm to carry out such averaging is included within the MATLAB SimPowerSystems blockset, but several significant deficiencies have been noted and overcome. The improvements made during this thesis include speed increases, improvements to the latency and coherency of the algorithm output, extension to 2^{nd} order interpolation, and modification of the code to allow robust operation over long periods of time in embedded processing applications. Without such improvements, the MATLAB exact-time averaging algorithm was significantly limited in application scope. With these improvements, the algorithm can now be used as a robust building block for AC signal measurements in embedded target processors.

A major contribution of this thesis is the development and demonstration of simple but effective algorithms using repeated application of this exact-time averaging block. Two key properties of this exact-time averaging algorithm are identified within this thesis, which, now being fully understood, help to define optimum architectures for measurement algorithms.

The first property is that the output of such an averaging block (for a perfect sinusoidal input and over a time period of exactly 1 cycle) exhibits ripple due to interpolation error and the finite sample interval, but that the frequency of the ripple is predictable. Typically, when measuring a Fourier correlation the input signal is at frequency f, and the Fourier correlation products are therefore at frequency $2f$. Averaging over one cycle period then leads to interpolation error at a frequency of $2f$. This can be almost

completely removed by a further exact-time average over ½ a cycle period. Thus, a single cycle Fourier correlation followed by a ½ cycle average is shown in this thesis to be an extremely simple but powerful measurement architecture to allow low sample rate operation down to 10 samples per cycle, with performance surpassing that of more complex 2nd harmonic cancellation techniques.

The second property highlighted by this thesis is that cascaded average filters provide excellent attenuation of Gaussian noise, with performance considerably exceeding that of low-pass filters with equivalent latencies. The cascading of two 2-cycle averaging filters, via the convolution of the impulse responses and the pole-zero placements, also provides much better noise reduction than does a single 4-cycle averaging filter.

Combining the new knowledge of these two important properties, this thesis shows how the exact-time averaging block can be used to build a number of adapted and entirely novel stages, and that these stages can be cascaded in optimum novel configurations to create extremely effective measurements of AC signals. The measurement latency can be traded off against minimisation of ripple and noise, under adverse conditions of harmonics, inter-harmonics, noise, ADC quantisation, and other interfering signals. This thesis shows how to make the optimum measurements with latencies (within the digital computations) of ½, 1, 1½, and more than 1½ cycles, with increasing performance as allowed latency increases.

The largest single problem with measurements at low sample rates is identified in this thesis as that of aliased harmonics (predominantly the 9th and 11th) causing low frequency ripple at the measurement output. This is found to be a hurdle to making adequately ripple-free measurements at a main frame rate of 10 samples per cycle, in the cases of worst harmonic distortion. To comply with the system-level requirements, two solutions to this problem are proposed together. Firstly, a novel adaptive ripple-remover is designed and implemented. This measures the frequency of any sub-harmonic ripple and removes it as far as possible, using exact-time averaging over reasonably long timeframes as allowed within the measurement latency. An important feature of this algorithm is the ability to automatically bypass itself during transients, thus providing faster measurements at the expense of increased noise/ripple when appropriate. This is extremely valuable since it can be used within algorithms which are used both for control (where low ripple is of primary concern) and also for relaying (where response time is of primary concern). Secondly, to achieve the lowest ripple performance (±0.001pu ripple on voltage measurements with 28% THD), it is also necessary to oversample the ADCs at 3kSa/s (6x

oversampling) and apply very small 6-tap FIR filters at this higher frame rate, before downsampling to 500 Sa/s (10 samples per cycle) for the main measurement algorithm. However, such an oversampled measurement architecture does not add expense to measurement hardware because it can be realised on existing microcontroller platforms such as the Infineon TC1796.

A general measurement architecture proposed by this thesis is shown in Figure 4-42. This architecture, which is fully implemented and tested within this thesis, allows "Class A" measurement accuracies to be achieved but at much lower sample rates and measurement latencies than traditionally used.

To measure frequency effectively, the same architecture is used with embellishments. More than 8 candidate solutions were compared, from which the final solution has been developed and selected. The best solution is found to be a hybrid of a 3-phase Clarke-transform based measurement and a novel Frequency Locked Loop (FLL) algorithm which measures both frequency and voltage (amplitudes & phases) of a 3-phase voltage set. The hybrid combines the best properties of each algorithm. The Clarke transform measurement is very fast settling due to the nominally constant speed of rotation of the AB vector, but does not function well during two-phase faults due to collapse of the AB vector trajectory to a straight line. The FLL is slower settling but can tolerate two-phase faults and large levels of unbalance without detriment to measurement accuracy/ripple. Coupling the two algorithms together with the appropriate averaging stages and decision processes is a significant achievement and provides an ideal solution. It is shown to provide significantly better response than algorithms based upon zero crossings, phased locked loops (PLLs) or other previously published techniques. Although the proposed solution is not a PLL, many of the concepts and algorithmic blocks could be applied to PLL applications to good advantage.

The Clarke-FLL hybrid algorithm also includes code which provides the following additional features:-

- Self-checking for validity of frequency measurement output
- Initial fast-settling of frequency measurement within 2 cycles (40ms) following signal application.
- Ride-through of frequency measurement for configurable time periods during brief 100% three-phase voltage dips, to allow the maximum potential for riding through system faults without tripping generators or loads unnecessarily.

None of these features have been seen before within published algorithms.

To prove the robustness and applicability of the new algorithms, they have been implemented and benchmarked on two real-time hardware platforms: the Infineon TC1796 microcontroller and the Real-Time-Station from Applied Dynamics International. Use of the algorithms over extended periods of time has proved their computational robustness in the real-time environment, which requires much more carefully constructed software than required by simple simulation exercises. The algorithms have been incorporated into several applications used in the laboratory at Strathclyde, including a microgrid management control agent (running at 500 samples per second) and a power quality meter (running at 1500 samples per second).

Thorough benchmarking exercises were also conducted to measure the breakdown of the algorithm execution times; these are presented in Appendix G. Such data is rarely available and even more rarely published, but is extremely valuable as a tool for speed improvement and higher-level system design. The data was used during this thesis to incrementally improve the speed of several key algorithmic blocks and allowed an overall execution time reduction of 25 to 50% for the major algorithms. Re-use of calculations and minimisation of trigonometric function evaluation within the analysis blocks also contributes to the small execution time. The total execution time for measurement of 3-phase voltages and currents at a node, with full sequence analysis and power flow analysis, is 156μs on the TC1796 microcontroller, less than 8% of a 2000μs frame time at 500 samples per second. Importantly, this leaves the remaining 1844μs frame time available for other generator/microgrid measurement and control functions

In addition to the creation and testing of the fundamental measurement algorithms, an algorithm has also been realised for a new type of passive loss-of-mains detection relay, called a Phase Offset Relay (POR). This is the first implementation of such a relay in a robust version suitable for deployment on a real-time target. This relay is independent of generator type, and uses local voltage measurements only. The relay uses calculations of perceived phase offset, relative to a parent network.

It is shown that this relay can be used to successfully detect loss-of-mains with generation-load imbalances of only 2.5% (real or reactive) within the 2 seconds allowed by IEEE 1547, even for power systems containing synchronous generators, using a trigger setting of 0.15-0.2Hz/s and a trip setting of 20°. This is comparable with the best available ROCOF relays using ROCOF trip settings of 0.15-0.2Hz. However, the main aim of this new relay algorithm is to bring the following additional benefits:-

- Much lower risk of spurious tripping due to noise, due to the double-integration (averaging) stages involved in the conversion from the ROCOF figure to first frequency and then phase.
- Much lower risk of spurious tripping due to extended trip times (up to that allowed by the system operator) when ROCOF only just exceeds the trigger threshold but the phase offset accumulates slowly. This discriminates genuine LOM events against normal load steps much better than a ROCOF relay.
- Much lower risk of spurious tripping due to post-fault power system oscillations, by the design and implementation of a novel new algorithm which dynamically adjusts the trip setting during and immediately subsequent to close-in faults. The relay is shown not to trip during simulated scenarios which cause commercially available and other proposed relays to trip.

The final significant piece of work in this thesis is a new strategy for microgrid management which involves small reactive power flow adjustments. The aim of this is to guarantee detection of a LOM event even when active power is exactly balanced within a microgrid. This deliberate match of active power is desirable as a strategic pre-islanding measure, but without the new reactive power control algorithm, such a power match introduces a high risk of not detecting a LOM event. The new algorithm continually monitors reactive power exchange between a local power system and its parent network (grid), and makes small adjustments to the generator reactive power output if required, such that the non-detection-zone (NDZ) of the loss-of-mains detection relay is always avoided by enough to provide a detectable but not over-large frequency disturbance upon islanding. Allowance is also made for the possibility of trapped load. The algorithm is tested using a real microgrid containing a synchronous generator and real/reactive loads, and shown to be fully effective. A generator stability analysis also shows that standard droop controls can be used, whereas previous published works with similar aims require unconventional droop controls to allow detection of LOM. The use of conventional droop controls is important since it allows frequency and voltage support functions at the same time as avoiding the LOM NDZ. The application of this reactive power control strategy is shown in the laboratory to reduce the worst-case LOM detection time from >10 seconds to <2 seconds, in line with IEEE 1547.

All of the algorithms developed during this thesis have been created to meet specific needs of an integrated microgrid management control system, which executes on a single microcontroller platform. All of the algorithms have been successfully integrated within this control system at 500 Sa/s and perform as described in this thesis, enabling more

reliable operation of the microgrid than was possible prior to the development of these solutions. This has been proved for the frequency/amplitude/phase measurements by trialling several published and novel candidate solutions in the laboratory environment over a period of 4 years, before the algorithms presented in chapters 3 to 5 were finalised. Their measurement latency and noise/ripple behaviour of the Clarke-FLL hybrid surpasses any other method yet tried or published, given the constraints of sample rate laid down in this thesis. This leads to smaller control actions due to noise/ripple, and more stable operation due to the low latency. The effectiveness of the LOM NDZ avoidance strategy has also been proved in the laboratory environment.

The algorithms are coded in a combination of Simulink (MATLAB R14SP1 was used throughout) and C-code (Simulink "S-functions") and are fully robust for long-term real-time deployment. The Simulink "real-time-workshop" and "embedded coder" features have proved to be an effective way of writing error-free code which can be tested in simulation on a PC and then deployed (without code modification) to real-time targets. The combined algorithmic designs and results from this thesis now provide an excellent foundation upon which to build more advanced microgrid control and protection applications.

7.1 Further Work

The measurement algorithms of chapters 3-5 have been tested many times and no deficiencies are currently known. The point of note is that if sample rates higher than 500 Sa/s can be used, then the algorithm performance will be better than described in this thesis. A possible opportunity (depending upon microcontroller capability) is to split the algorithms and to move the initial 1-cycle Fourier correlations onto the peripheral control processor at the oversampled rate of 3 kSa/s (or more), while leaving the main bulk of the algorithms at the much slower sample rate of 500 Sa/s. The oversampled FIR notch filter would be removed. This would further improve the performance of the algorithms and potentially allow the anti-aliasing filter cut-off frequency to be raised, improving the RMS and THD measurements as well as the Fourier fundamental measurement.

Suitable settings for the POR are dependent upon the size and qualities of the parent network. Typical settings for connection to the UK national grid are a ROCOF trigger threshold of 0.15-0.2 Hz/s and a trip threshold of 20°, with an imbalance in real or reactive power of about 0.025 (2.5%) required to guarantee detection of LOM within 2 seconds. For smaller parent networks, the ROCOF trigger threshold needs to be raised to avoid spurious trips due to regularly occurring frequency deviations. A larger reactive power imbalance is then required to guarantee detection within 2 seconds. Exploring the (non) linearity of this relationship, and the practical ROCOF threshold required for different networks, is a significant opportunity for further work. It might be possible to create an algorithm which can continually monitor the magnitude of ROCOF on a power system and automatically set ROCOF trigger thresholds appropriately, to adapt to changing parent network parameters.

The LOM detection tests could be repeated with a significant proportion of high-inertia spinning loads, to verify that the LOM events can still be detected. During the testing of this thesis, only static R&L loads were used due to equipment availability.

Finally, as a modification to the reactive power control algorithm, Du (2005) and Ye (2006) propose the addition of positive feedback to the P and Q controls (throttle and field) based upon differential filtered or bandpass filtered measurements of the measured system voltage and frequency. These systems as presented are undesirable since they include no standard droop controls to enable network support. If the droop controls used in this thesis could be combined with the positive feedback differential terms, faster detection times might result for the same small P or Q imbalances.

8 About the author

Andrew Roscoe received the B.A. degree in Electrical and Information Sciences Tripos at Pembroke College, Cambridge, England in 1991. He was awarded the M.A. degree in 1994, and elected to chartered membership of the IEE in 1996. Andrew worked for GEC Marconi from 1991 to 1995. He was involved in antenna design and calibration, specialising in millimetre-wave systems and solid-state phased array radars. Andrew worked from 1995 to 2004 with Hewlett Packard and subsequently Agilent Technologies, in the field of microwave communication systems, specialising in the design of test and measurement systems for personal mobile and satellite communications. Andrew was awarded an MSc with distinction from the University of Strathclyde in 2004, in the field of "Energy systems and the Environment". Andrew is currently a research fellow in the Institute for Energy and the Environment, Department of Electronic and Electrical Engineering at Strathclyde University, working in the field of distributed generation and active network management. Recent projects include real-time pricing studies, the creation and deployment of microgrid control algorithms at the 100kVA scale, the design/build of a 10kVA 3-phase inverter, new algorithms for the measurement of dynamic system parameters using low sample rates, and loss-of-mains detection strategies.

Appendix A Lower-level FLL code details

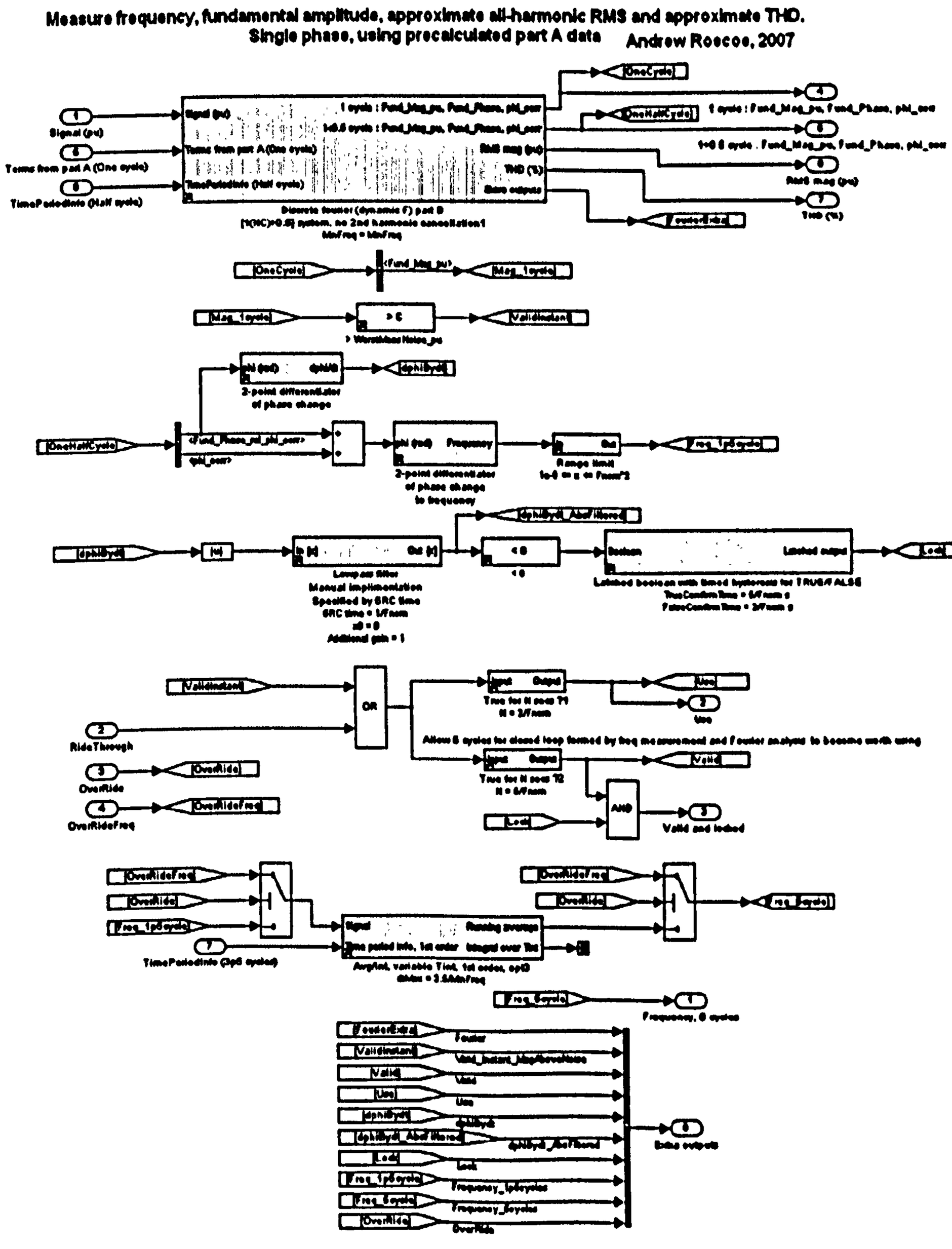


Fig. A-1 : Clarke-FLL hybrid - detail (2); core

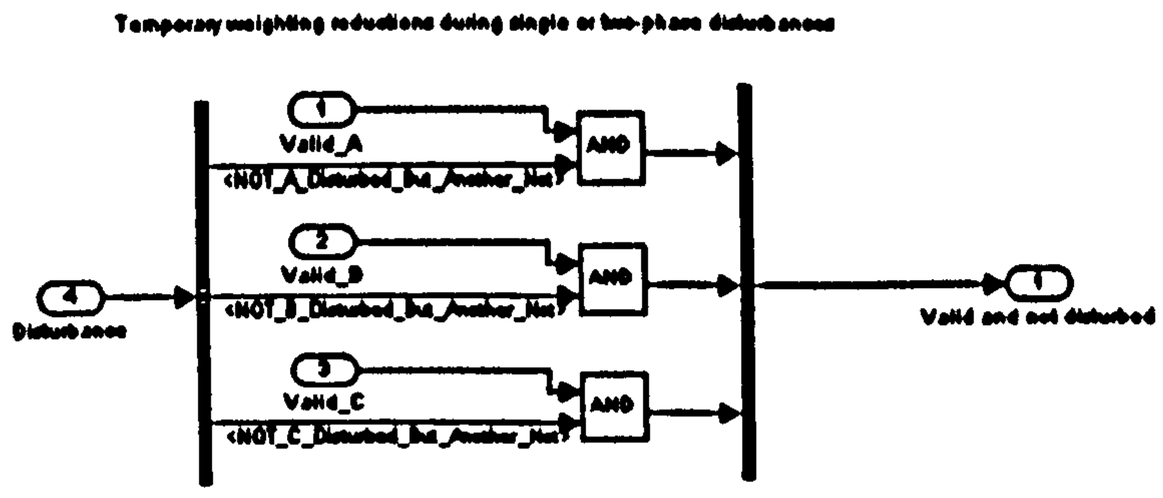


Fig. A-2 : Clarke-FLL hybrid - detail (3); single and two-phase fault ride-through

Fig. A-3 : Clarke-FLL hybrid - detail (4); weighted averaging of frequency

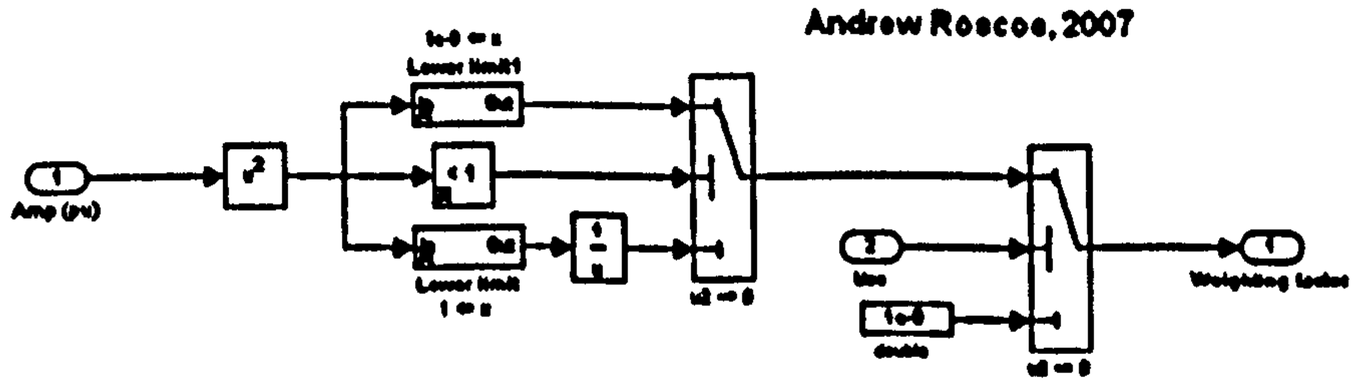


Fig. A-4 : Clarke-FLL hybrid - detail (5); weighted averaging of frequency subroutine

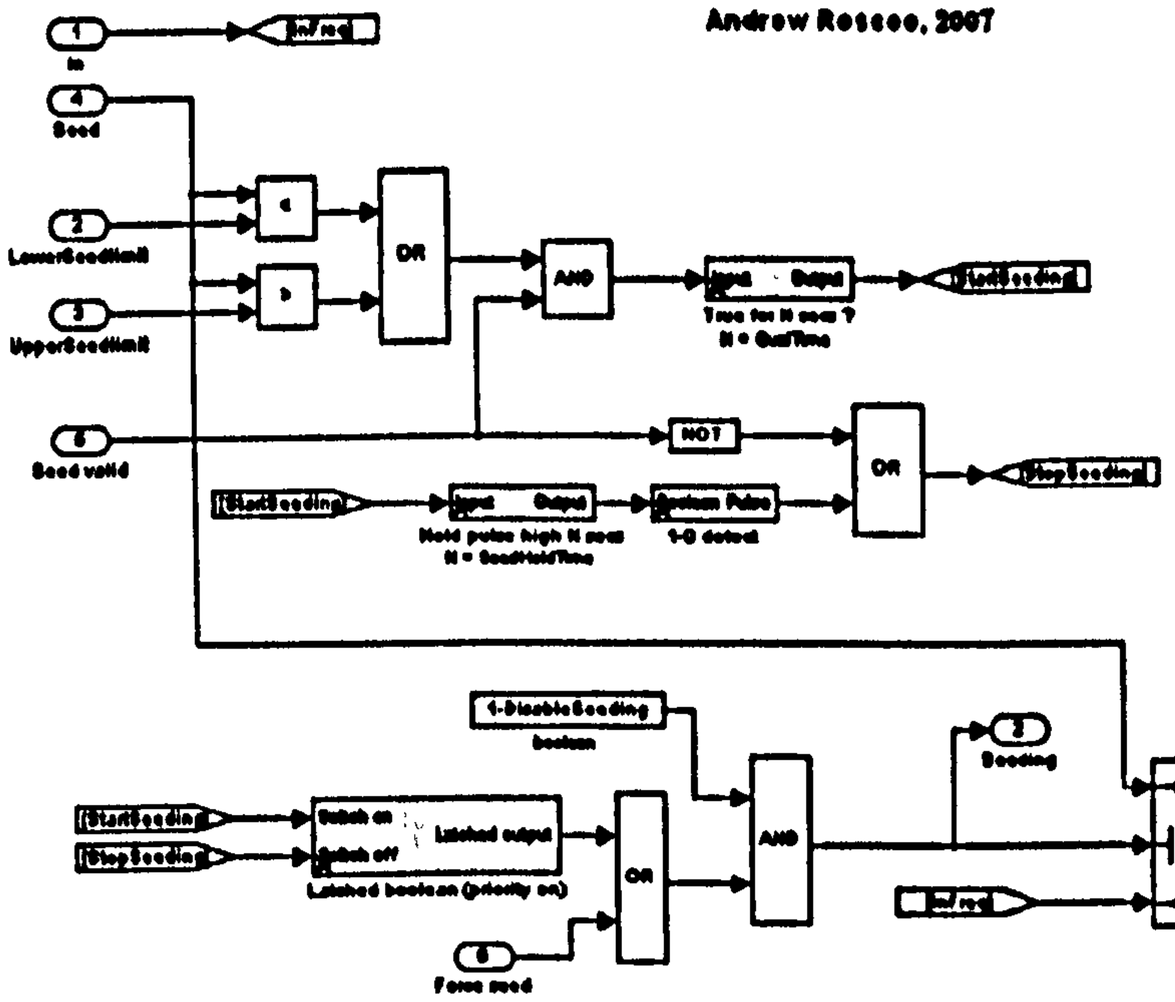
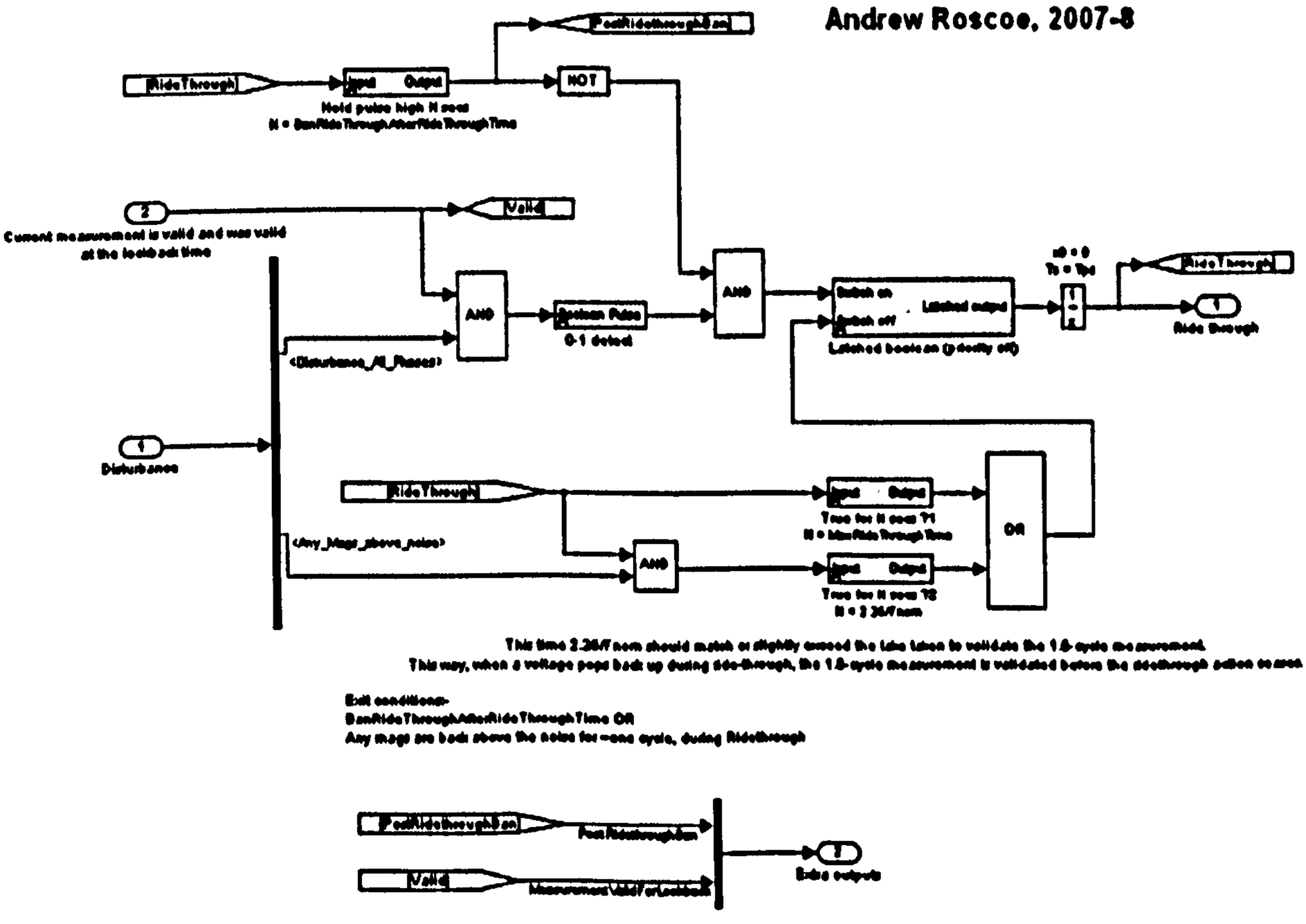


Fig. A-5 : Clarke-FLL hybrid - detail (6); seeding



This time 2.25/f nom should match or slightly exceed the time taken to validate the 1.5-cycle measurement.
This way, when a voltage pops back up during ride-through, the 1.5-cycle measurement is validated before the ridethrough action is taken.

Exit conditions:
BothRideThroughAfterRideThroughTime OR
Any mags are back above the noise for one cycle, during Ridethrough

Fig. A-6 : Clarke-FLL hybrid - detail (8); ride-through activation

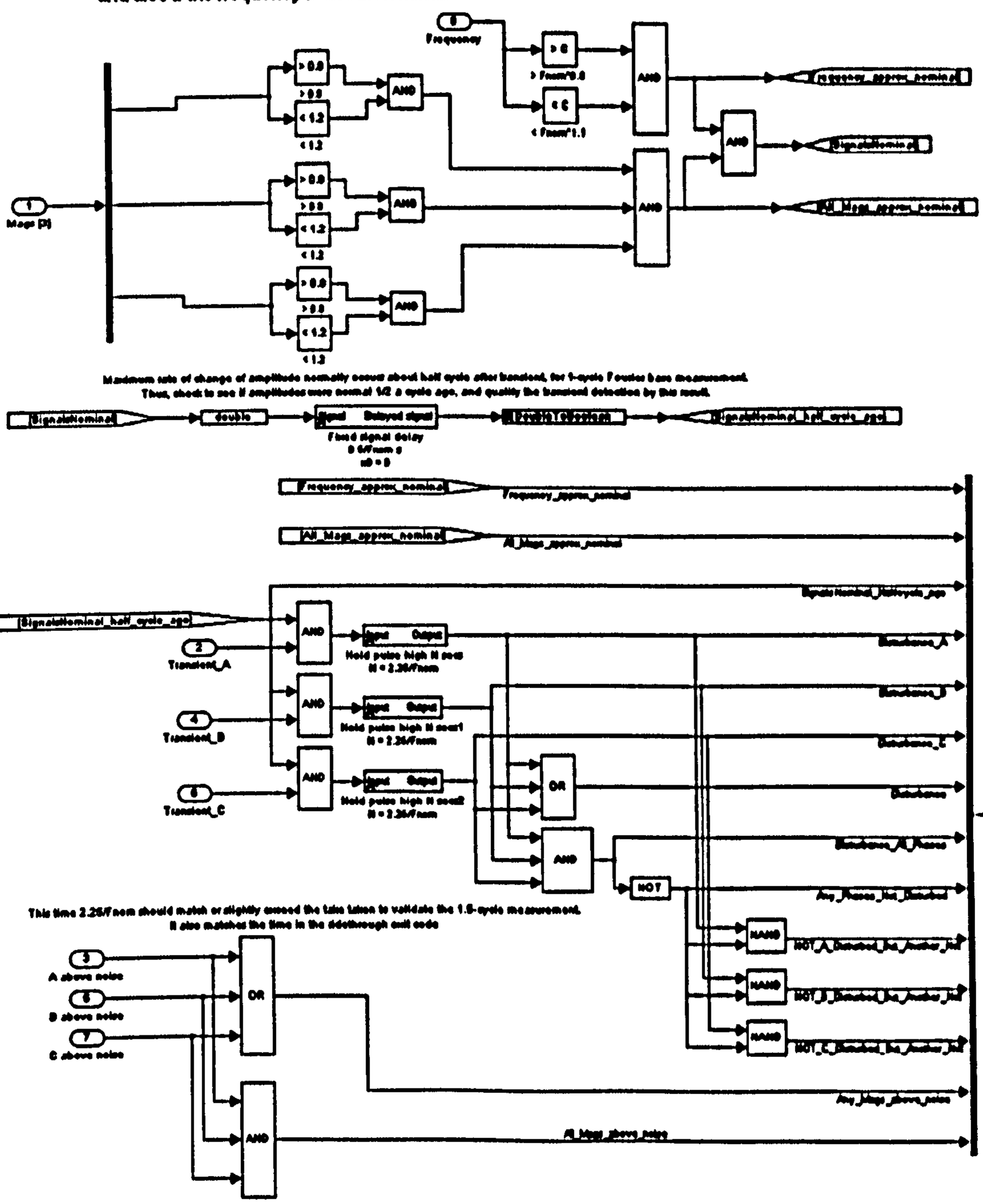


Fig. A-7 : Clarke-FLL hybrid - detail (7); disturbance detection

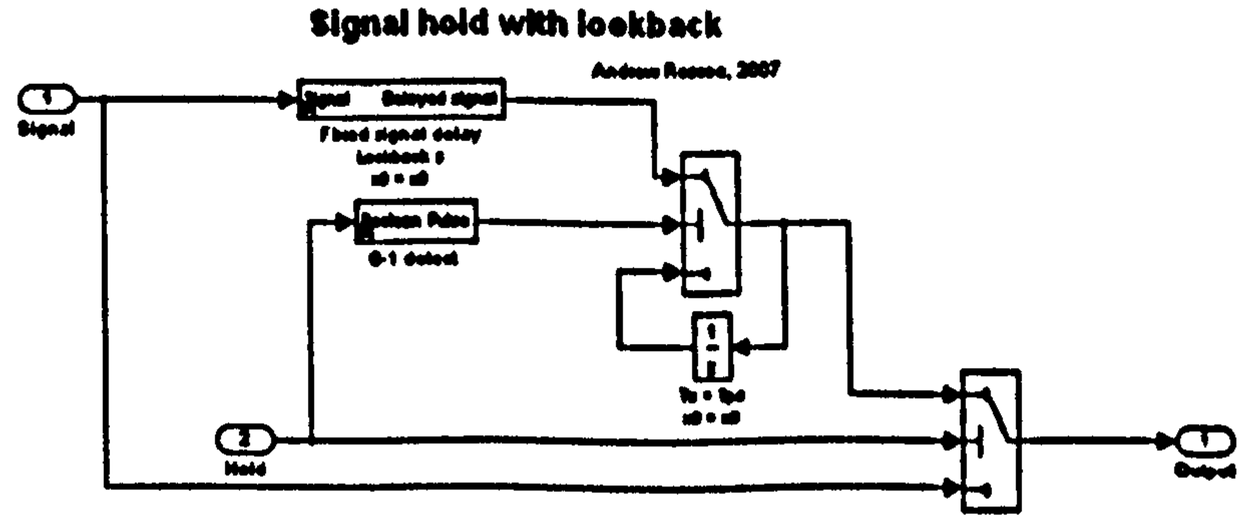


Fig. A-8 : Clarke-FLL hybrid - detail (9); ride-through look-back

Appendix B Useful formulae

B.1 Three-phase power flows

B.1.1 Power, current, voltage and resistance

$$I_p = \frac{P_3}{V_3\sqrt{3}} \qquad \{B.1\}$$

$$P_3 = \frac{V_3^2}{R_p} \qquad \{B.2\}$$

- I_p = RMS Phase current (A)
- P_3 = Three phase power flow (VA)
- V_3 = RMS line-line voltage (V)
- R_p = Resistance, phase to neutral in a balanced star system

B.1.2 Estimation of overhead line voltage required

An estimation of required 3-phase voltage for overhead line connections can be made by the formula:

$$E = k\sqrt{Pl} \qquad \{B.3\}$$

where:-

- E = 3-phase line voltage (kV)
- P = power to be transmitted (kW)
- l = line length (km)

and k is a coefficient dependent on the impedance (real and reactive) of the line. k varies from about 0.06 for a compensated line, to 0.1 for a compensated line with voltage regulation of 5%.

This formula derives from the limits of phase angle across the transmission line, which should never exceed 90° otherwise the network is at risk of breaking apart. The phase angle is determined by the real power flow and the line inductance, and by the reactive power flow and the line resistance. [Wildi, T (2002). Electrical machines, drives and power systems. Fifth edition. Prentice Hall. ISBN 0 13 098637 2.]

B.1.3 P and Q flows along a transmission line

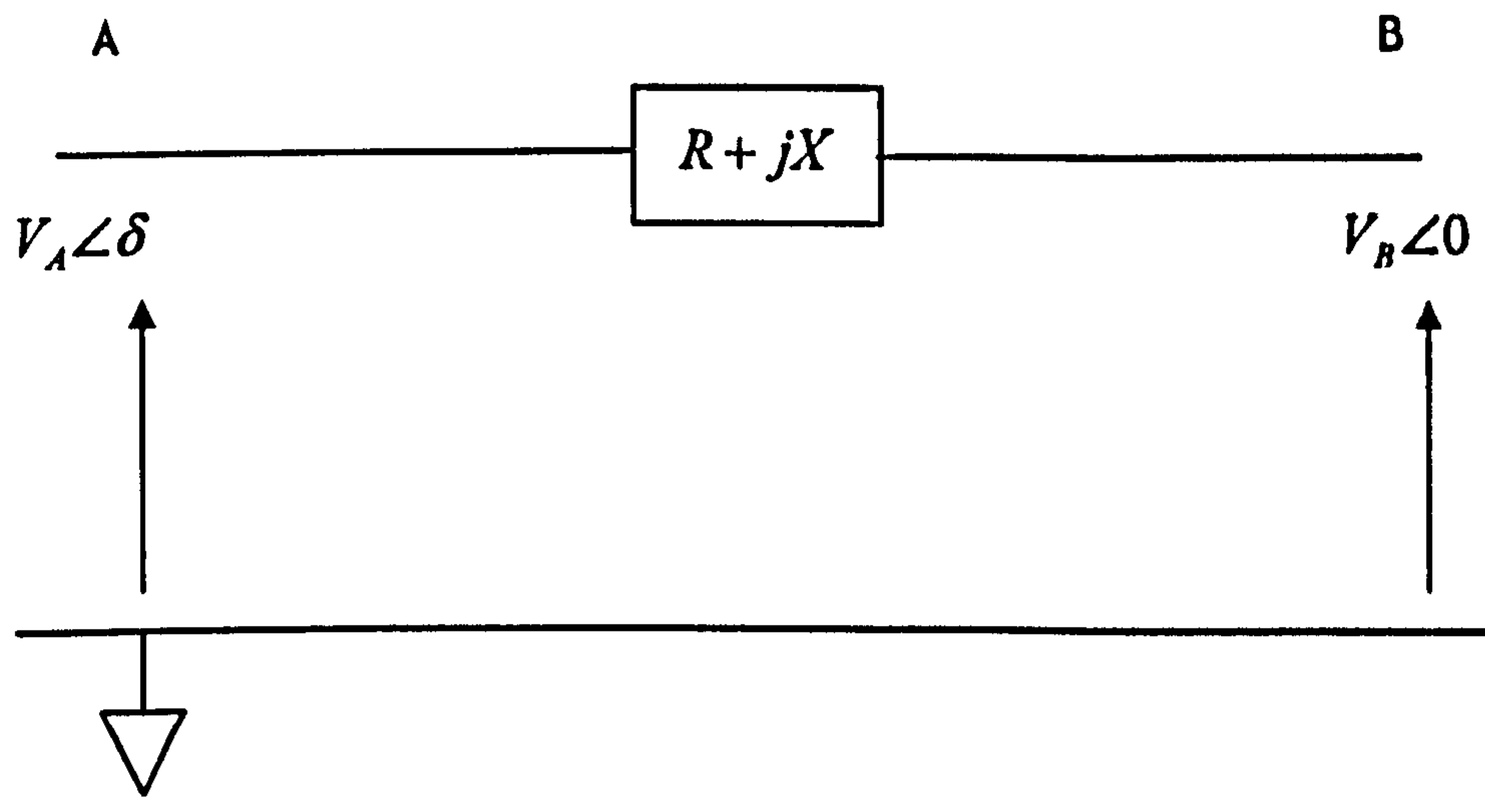


Fig. B-1 : P and Q flows along a transmission line

$$P = \frac{V_A}{R^2 + X^2}\left[XV_B \sin\delta + R(V_A - V_B \cos\delta)\right]$$

$$Q = \frac{V_A}{R^2 + X^2}\left[X(V_A - V_B \cos\delta) - RV_B \sin\delta\right]$$

B.1.4 AB vector trajectory under a single-phase fault

This is easiest analysed by dropping phase A

By (5.3)

$$\begin{bmatrix} A \\ B \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & \frac{-1}{3} & \frac{-1}{3} \\ 0 & \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{3}} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \times \begin{bmatrix} 0 \\ \cos\left(\omega t - \frac{2\pi}{3}\right) \\ \cos\left(\omega t + \frac{2\pi}{3}\right) \end{bmatrix} = \begin{bmatrix} \frac{1}{3}\cos(\omega t) \\ \sin(\omega t) \\ -\frac{1}{3}\cos(\omega t) \end{bmatrix}$$

Other single-phase faults will result in other elliptical traces with aspect ratios of 3:1

B.2 Filters

B.2.1 1st order low-pass filter

$\frac{\overline{V_{out}}}{\overline{V_{in}}} = \frac{1}{1 + j2\pi fRC}$ if the filter is made of resistor and capacitor components R & C

The cut-off frequency F_c is set by $\omega_c = 2\pi F_c = 1/RC$

This can also be expressed in the Laplace domain as $\frac{1}{1 + \frac{s}{\omega_c}}$

Hence,

$$\frac{\overline{V_{out}}}{\overline{V_{in}}} = \frac{1}{\sqrt{1 + \left(\frac{f}{F_c}\right)^2}} \angle -\arctan\left(\frac{f}{F_c}\right)$$

{B.4}

Which, for example, gives V_o/V_i as $1/\sqrt{2}$ at -45° if $f = F_c$.

Also, Given $2\pi F_c = 1/RC$, then if we want to define a filter by its "5RC" settling time, i.e. the time taken to settle to within 1% after a step change, then we can determine the appropriate F_c by:

$$F_c = \frac{5}{2\pi \cdot 5RC} = \frac{0.7958}{5RC}$$

{B.5}

B.2.2 Sample time required to accurately model an analogue low-pass filter

A low-pass filter can be approximated by a digital single-pole filter. This can be designed in Simulink, by the following generalised code segment (which can work with significantly more complex filters):-

```
z=sym('z');
% First get Num and Dec in terms of laplace
Num=[1];
Den=[1/(2*pi*LPF_Fc), 1];
H=tf(Num,Den);
Hd=c2d(H,Ts,'zoh');
```

```
[Num_z_cell,Den_z_cell]=tfdata(Hd);
Num_z_0=Num_z_cell{1,1}(2);
Den_z_1=Den_z_cell{1,1}(1);
Den_z_0=Den_z_cell{1,1}(2);

filter_num=Num_z_0
filter_den=z*Den_z_1+Den_z_0
Total_filter=filter_num/filter_den
```

The size of the time step, relative to the cut-off frequency and the actual waveforms input to the filter, determines how accurately the digital filter performance follows that of an analogue filter. This accuracy is important when simulations of analogue filtering hardware are being carried out. For example, in this thesis, a 125Hz low-pass analogue anti-aliasing filter is commonly modelled. Input signals of concern are those up to at least the 40th harmonic of 50Hz, or 2000Hz. To assess what time step is required to accurately represent a 125Hz analogue low-pass filter in the digital domain, a simple MATLAB script can be used.

This script shows that a time-step of approximately 40µs, 1/25000th of a second, or 500Sa/cycle at 50Hz, is required to accurately model the response in gain, to within 0.1dB (for input signals up to 2000Hz). The phase accuracy is only ±15°. A rule of thumb is therefore that the digital time-step needs to be of the order of 10 times smaller than the period of the highest frequency whose attenuation needs to be accurately modelled through the filter. The plots for 40µs and 100µs time-step filter implementations are shown in Fig. B-2 to Fig. B-5.

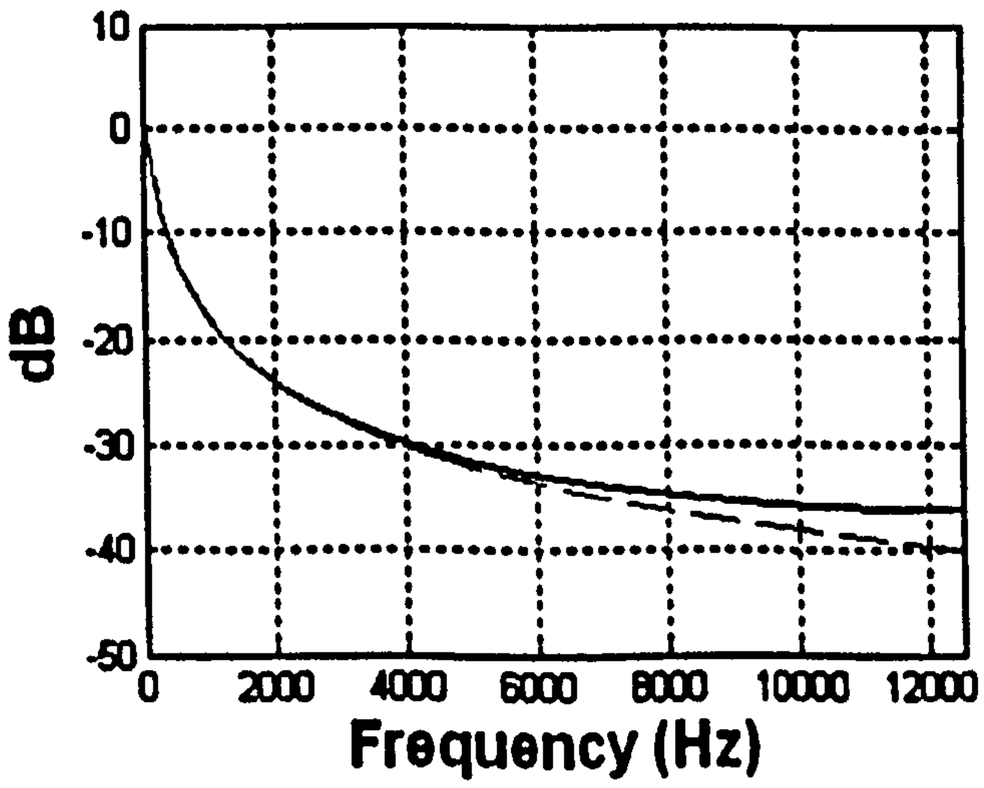


Fig. B-2 : Digital single-pole low-pass filter accuracy (gain), Fc=125Hz, Ts=40µs. Actual (solid) and theoretical (red dashes)

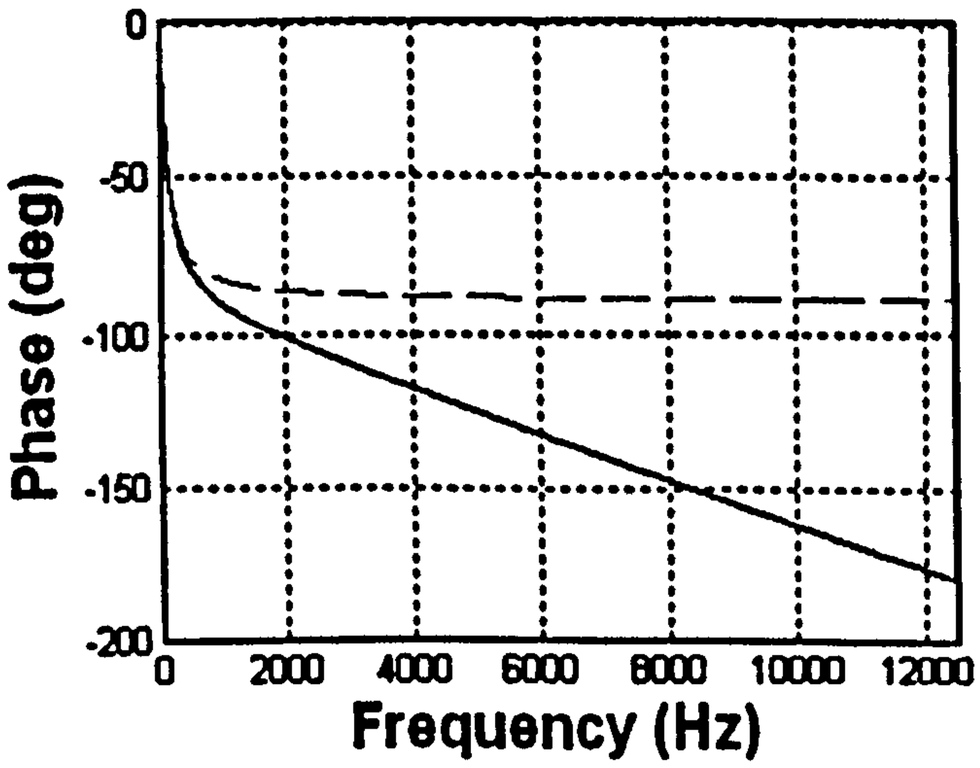


Fig. B-3 : Digital single-pole low-pass filter accuracy (phase), Fc=125Hz, Ts=40μs. Actual (solid) and theoretical (red dashes)

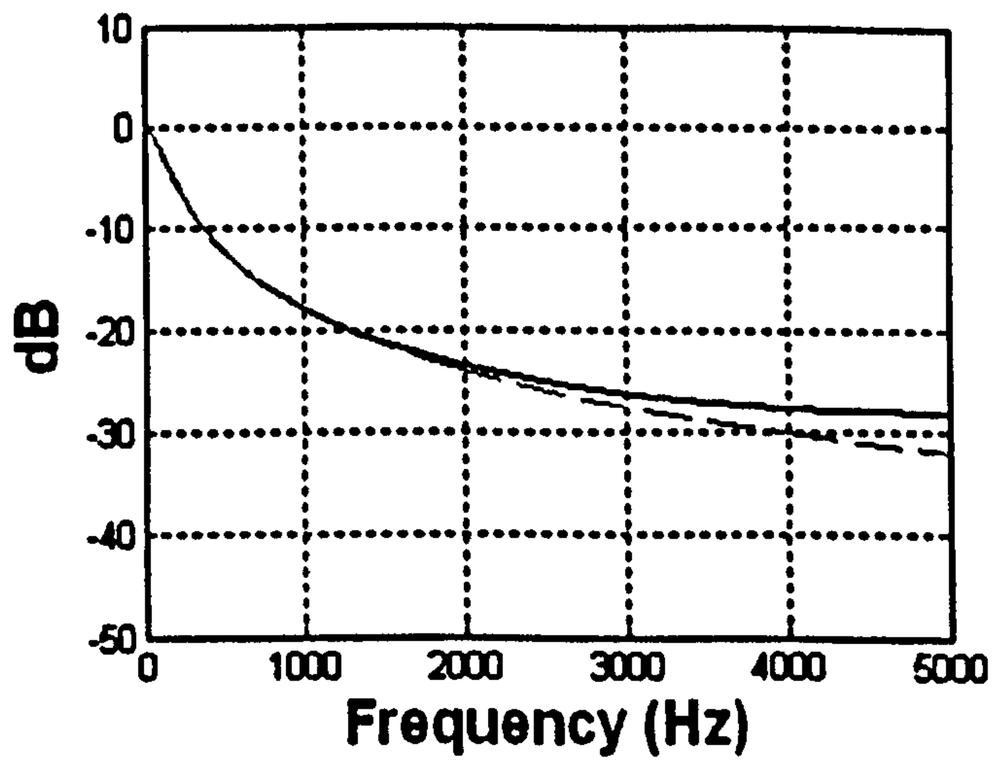


Fig. B-4 : Digital single-pole low-pass filter accuracy (gain), Fc=125Hz, Ts=100μs. Actual (solid) and theoretical (red dashes)

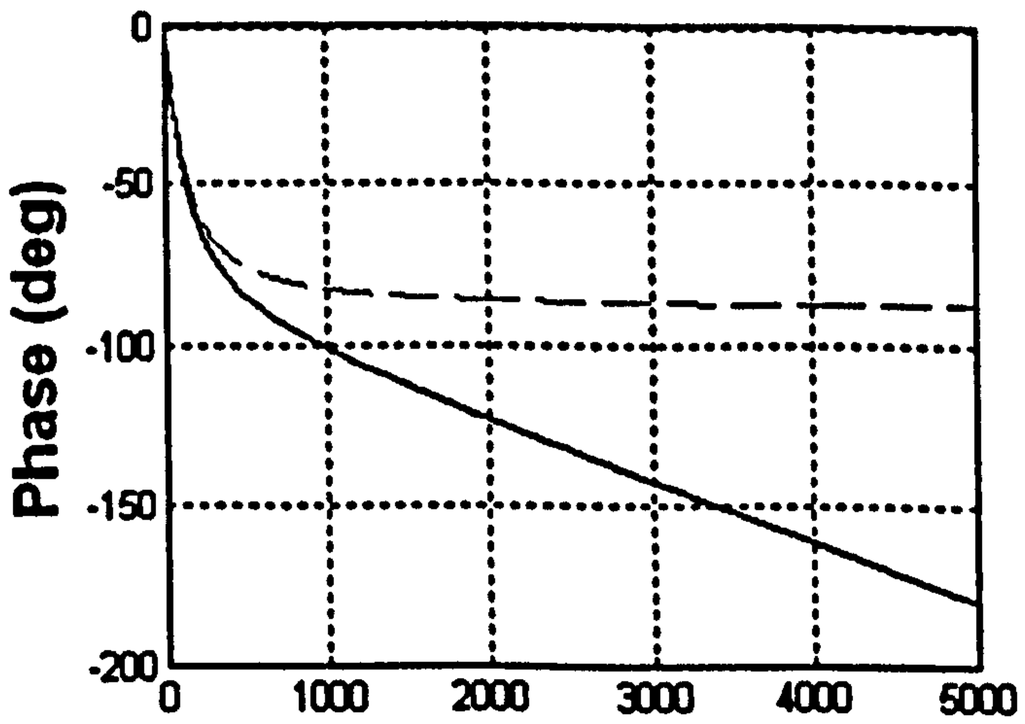


Fig. B-5 : Digital single-pole low-pass filter accuracy (phase), Fc=125Hz, Ts=100μs. Actual (solid) and theoretical (red dashes)

B.2.3 1st order high-pass filter

$\frac{\overline{V_{out}}}{\overline{V_{in}}} = \frac{1}{1+\frac{1}{j2\pi fRC}}$ if the filter is made of resistor and capacitor components R & C

The cut-off frequency F_c is set by $\omega_c = 2\pi F_c = 1/RC$

This can also be expressed in the Laplace domain as $\frac{1}{\frac{\omega_c}{s}+1}$ or $\frac{\frac{s}{\omega_c}}{1+\frac{s}{\omega_c}}$

Hence,

$$\frac{\overline{V_{out}}}{\overline{V_{in}}} = \frac{1}{\sqrt{1+\left(\frac{F_c}{f}\right)^2}} \angle \arctan\left(\frac{F_c}{f}\right)$$

{B.6}

Which, for example, gives V_o/V_i as $1/\sqrt{2}$ at +45° if $f=F_c$.

B.2.4 2st order low-pass filter

$\frac{\overline{V_{out}}}{\overline{V_{in}}} = \frac{1}{1-(2\pi f)^2 LC}$ if the filter is made of inductor and capacitor components L & C

The resonant frequency F_c is set by $2\pi F_c = 1/\sqrt{LC}$. Hence,

$$\frac{\overline{V_{out}}}{\overline{V_{in}}} = \frac{1}{1-\left(\frac{f}{F_c}\right)^2}$$

{B.7}

which has:-

- a phase of 0 if $f<F_c$
- a phase of 180° if $f>F_c$
- an infinite gain (resonance) when $f=F_c$

B.3 High-pass filter cutoff to give flat gain at nominal frequency, when combined with 2 cascaded low-pass anti-aliasing filters

By equations (3.3) & {B.6}, the gain of the total filter package (2 cascaded low-pass filters

plus high-pass filter) will be:

$$Gain = \frac{1}{\left(1+\left(\frac{f}{F_{cLPF}}\right)^2\right)\sqrt{1+\left(\frac{F_{cHPF}}{f}\right)^2}} \qquad \{B.8\}$$

where F_{cLPF} and F_{cHPF} are the cut-off frequencies for the low-pass and high-pass filters.

Equation {B.8} can be differentiated and solved to find the high-pass filter cut-off frequency required to achieve flat gain at nominal frequency (f), for a given low-pass filter.

$$Gain = \frac{1}{\sqrt{K}} \text{ where } K = \left(1+\left(\frac{f}{F_{cLPF}}\right)^2\right)^2\left(1+\left(\frac{F_{cHPF}}{f}\right)^2\right)$$

$$\Rightarrow \frac{dGain}{df} = -\frac{1}{2}\cdot\frac{1}{K^{\frac{3}{2}}}\cdot\frac{dK}{df}$$

Now, solve to find F_{cHPF} which makes $\frac{dGain}{df} = 0$. This can be done by solving for $\frac{dK}{df} = 0$

$$\frac{dK}{df} = 2\cdot\left(1+\left(\frac{f}{F_{cLPF}}\right)^2\right)\cdot\frac{2f}{F_{cLPF}^2}\left(1+\left(\frac{F_{cHPF}}{f}\right)^2\right)+\left(1+\left(\frac{f}{F_{cLPF}}\right)^2\right)^2\cdot\frac{-2F_{cHPF}^2}{f^3}$$

$$0 = \frac{2f}{F_{cLPF}^2}\left(1+\left(\frac{F_{cHPF}}{f}\right)^2\right)+\left(1+\left(\frac{f}{F_{cLPF}}\right)^2\right)\cdot\frac{-F_{cHPF}^2}{f^3}$$

$$0 = \frac{2f^4}{1}\left(1+\left(\frac{F_{cHPF}}{f}\right)^2\right)+\left(F_{cLPF}^2+f^2\right)\cdot\frac{-F_{cHPF}^2}{1}$$

$$0 = F_{cHPF}^2\left(2f^2-F_{cLPF}^2-f^2\right)+2f^4 = F_{cHPF}^2\left(f^2-F_{cLPF}^2\right)+2f^4$$

$$F_{cHPF} = \sqrt{\frac{2f^4}{\left(F_{cLPF}^2-f^2\right)}}$$

Which finally reduces to

$$F_{cHPF} = \frac{\sqrt{2}.f}{\sqrt{\frac{F_{cLPF}^2}{f^2}-1}} \qquad \{B.9\}$$

B.4 Peak impulse response of two cascaded averaging filters

For two digital averaging filters cascaded, the peak and shape of the impulse response to an input signal one sample long at magnitude 1 can be found via the following thought experiment (which is a shortcut to carrying out the convolution integral):

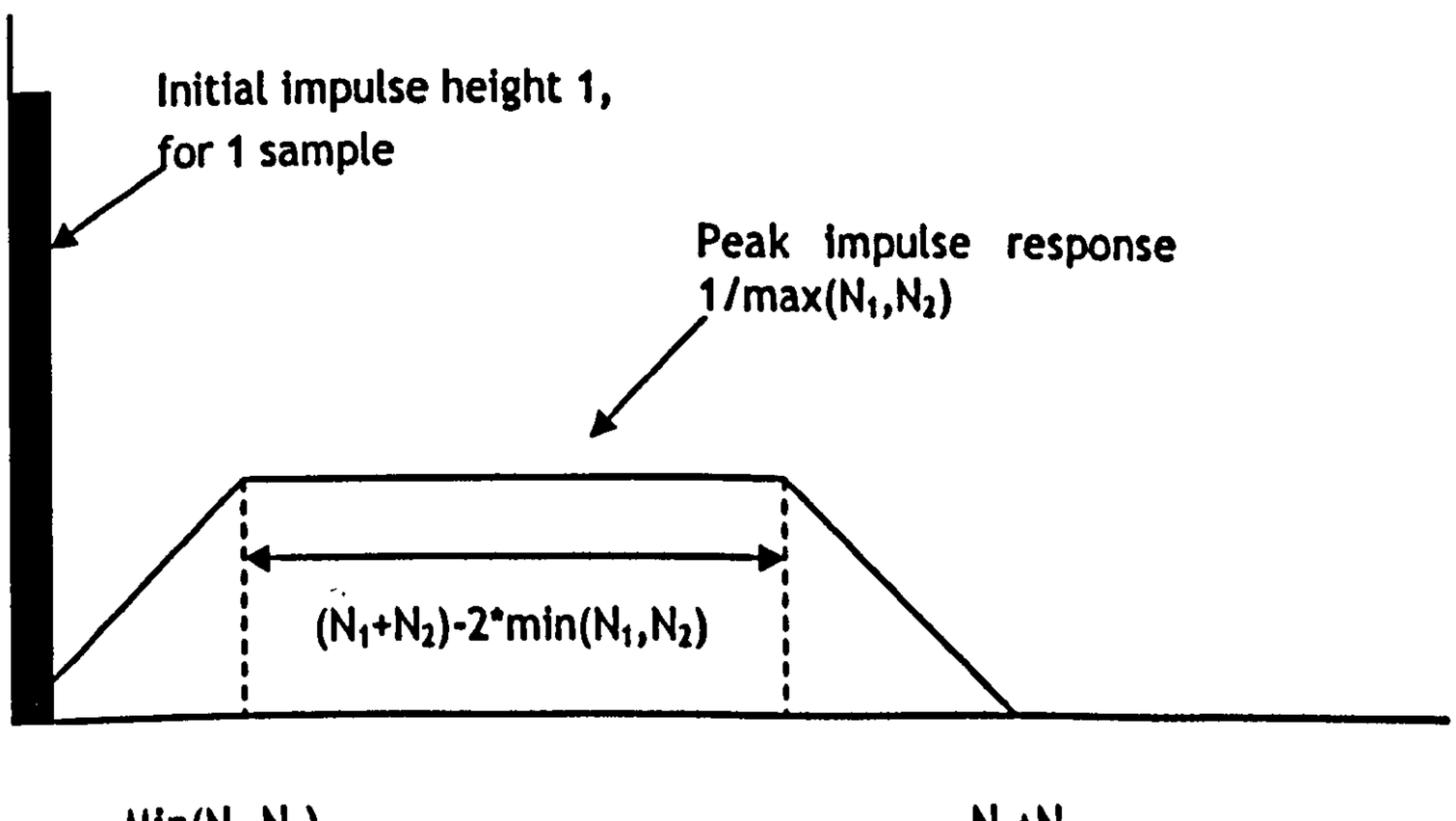
- The two cascaded filters average the signal for N_1 and N_2 samples respectively.
- When the impulse arrives, the output from the first filter becomes $1/N_1$ and will stay at this level for N_1 frames
- This signal immediately enters the second filter and this begins to rise at a rate of $1/N_2$ times the input signal level, which is $1/N_1$. The rise rate is thus $1/(N_1N_2)$ per frame.
- The rising slope of the output from the second filter will last for the shorter of N_1 or N_2 frames. When this time is passed, the output will plateau for $(N_1+N_2)-2*\min(N_1,N_2)$ frames, before falling in a symmetric fashion to 0, (N_1+N_2) frames after the impulse is applied.
- The plateau height will be $1/(N_1N_2)*\min(N_1,N_2) = 1/\max(N_1,N_2)$

The total "area" of the impulse response is thus $2*½*\min(N_1,N_2)*1/\max(N_1,N_2)+((N_1+N_2)-2*\min(N_1,N_2))*1/\max(N_1,N_2)$

$= 1/\max(N_1,N_2)*[\min(N_1,N_2)+((N_1+N_2)-2*\min(N_1,N_2))]$

$= 1/\max(N_1,N_2)*[(N_1+N_2)-\min(N_1,N_2)]$

$= 1/\max(N_1,N_2)*[\max(N_1,N_2)] = 1$



Appendix C Useful data

C.1 Typical parameters of overhead lines and underground cables

	System voltage	Phase conductor (Al/Fe), mm²	Resistance Ω/km (20°C)	Reactance Ω/km (50Hz)	Capacity
Overhead line	400V	25/0	1.06	0.3	150A
Overhead line	400V	50/0	0.64	0.28	250A
Overhead cable	400V	35/0	0.87	0.1	150A
Underground cable	400V	120/0	0.25	0.07	300A
Overhead line	11kV	50/0	0.64	-0.04 *	250A
Underground cable	11kV	185/0	0.16	0.08	380A
Overhead line	20kV	54/9	0.54	- 0.4 *	250A
Underground cable	20kV	120/0	0.25	0.11	300A
Overhead line	110kV	242/39	0.12	- 0.4*	650

* - value depends on spacing and cross-arm construction

Tab. C-1 : Typical parameters of overhead lines and underground cables

This data is sourced from the three sources:-

IEE (2003). Electricity distribution network design. IEE Power engineering series 21. 2nd edition. 2003. ISBN 0 86341 309 9. p28.

Wildi, T (2002). Electrical machines, drives and power systems. Fifth edition. Prentice Hall. ISBN 0 13 098637 2. p677.

Laughton, M. & Warne, D. (2003). Electrical engineers reference book. Sixteenth edition. Newnes. ISBN 0 7506 46373. p31/32.

C.2 ITI CBEMA Curve

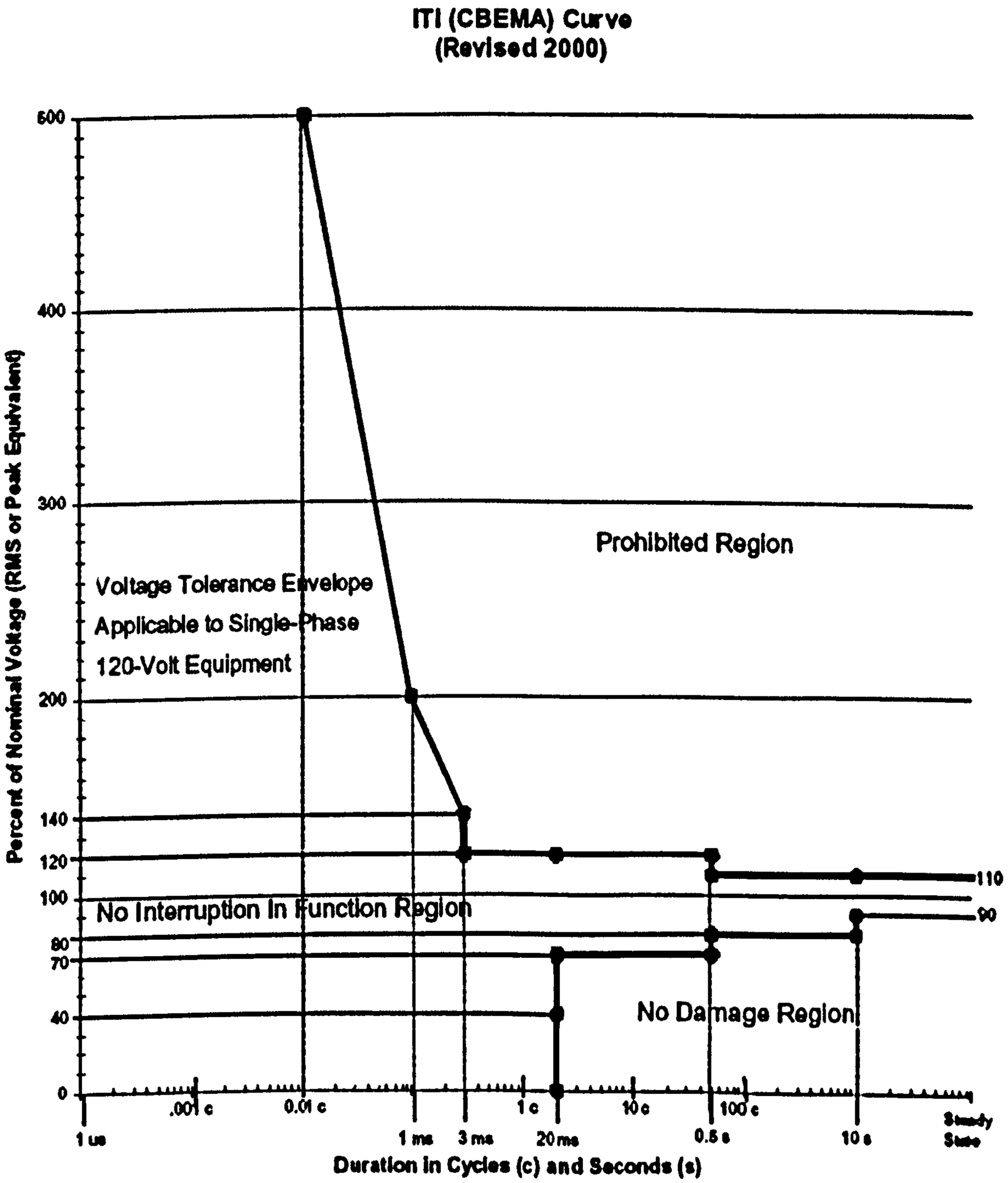


Fig. C-1 : The ITI (CBEMA) Curve

ITI. (2007). ITI (CBEMA) curve application note. [Online]. [Accessed 28/11/2007]. <http://www.itic.org/archives/iticurv.pdf>

Appendix D Pole-zero & Bode plots of various FIR averaging filter combinations

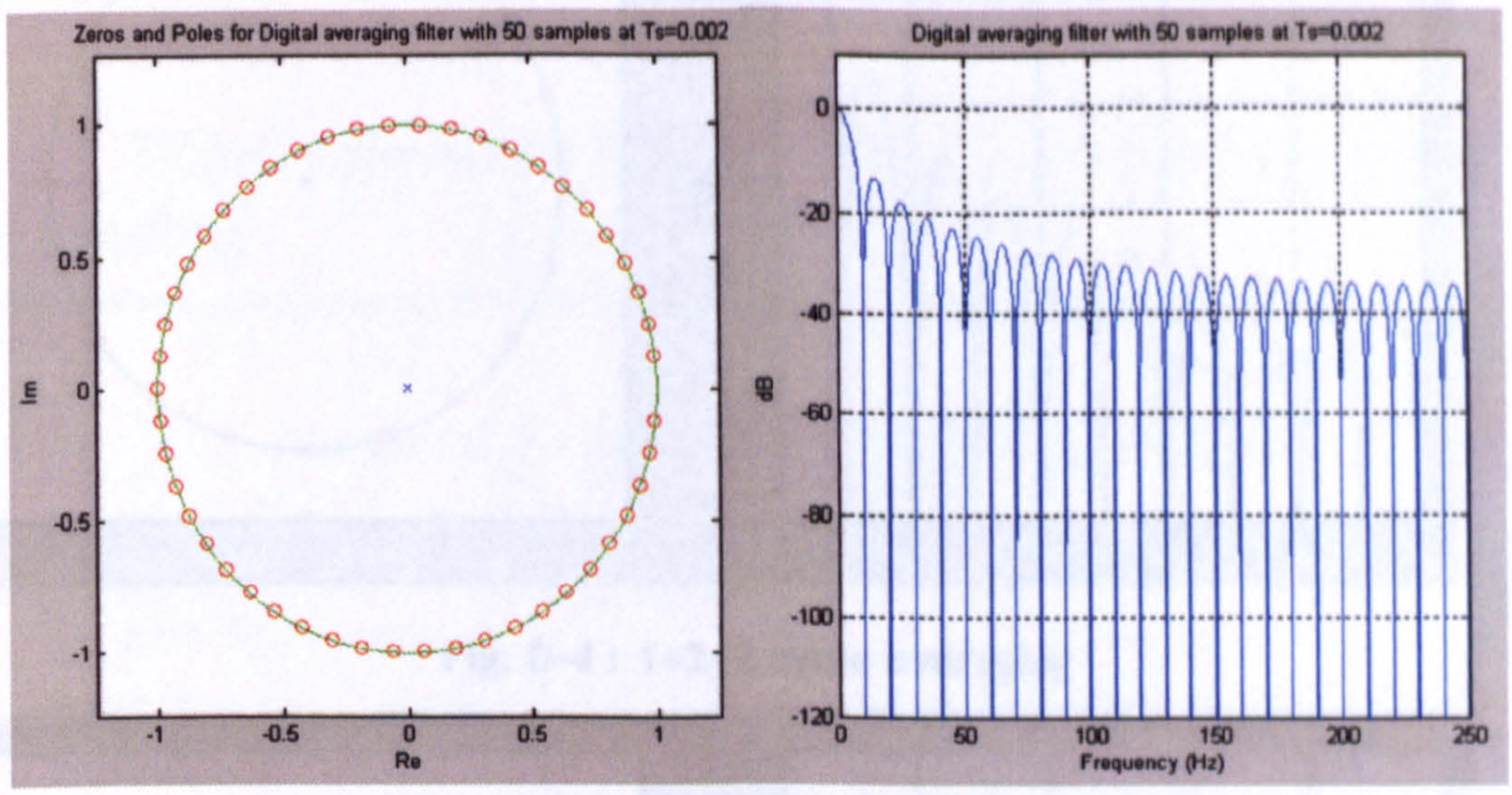


Fig. D-1 : 5 cycle averaging

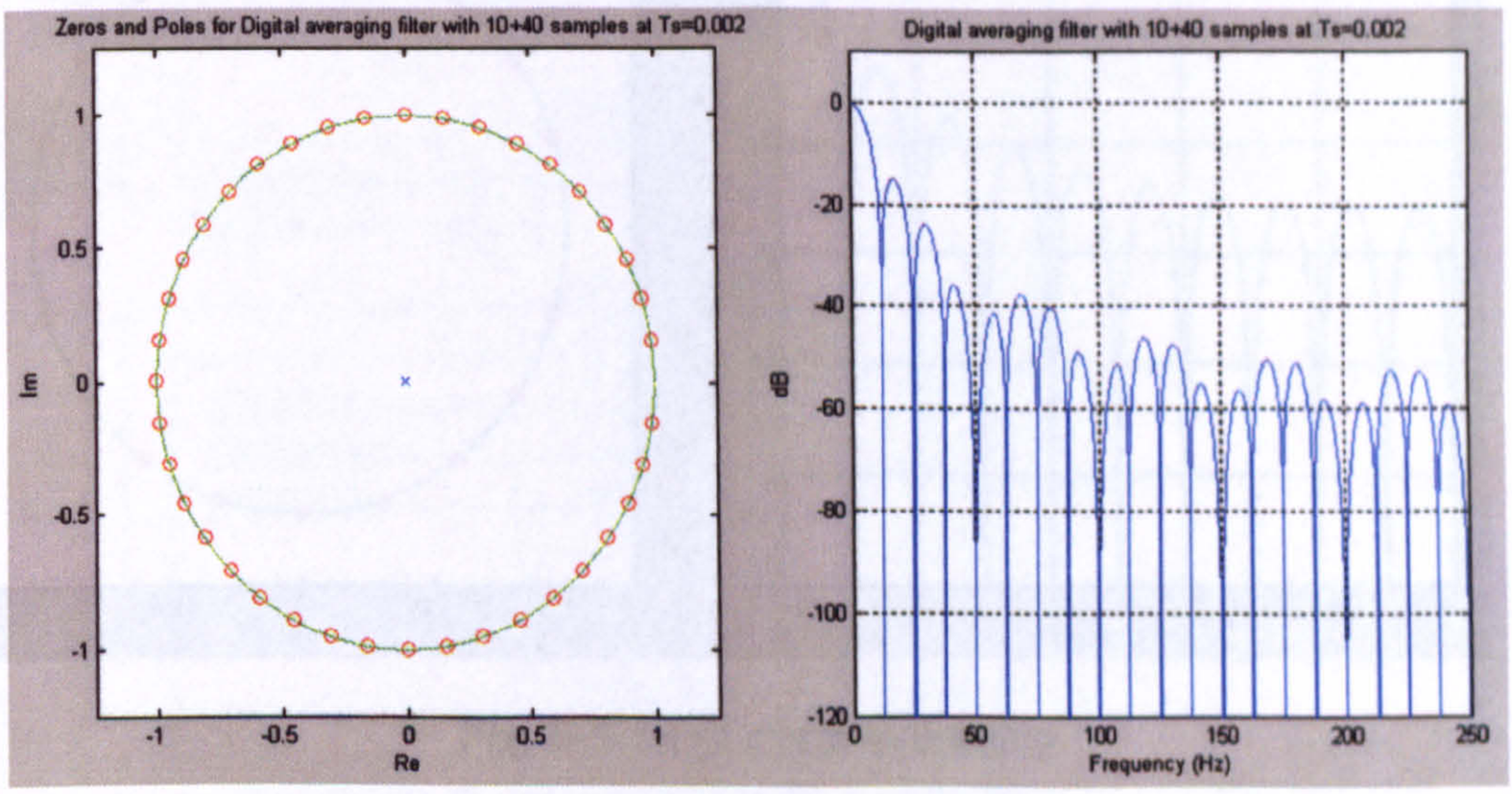


Fig. D-2 : 1+4 cycle averaging

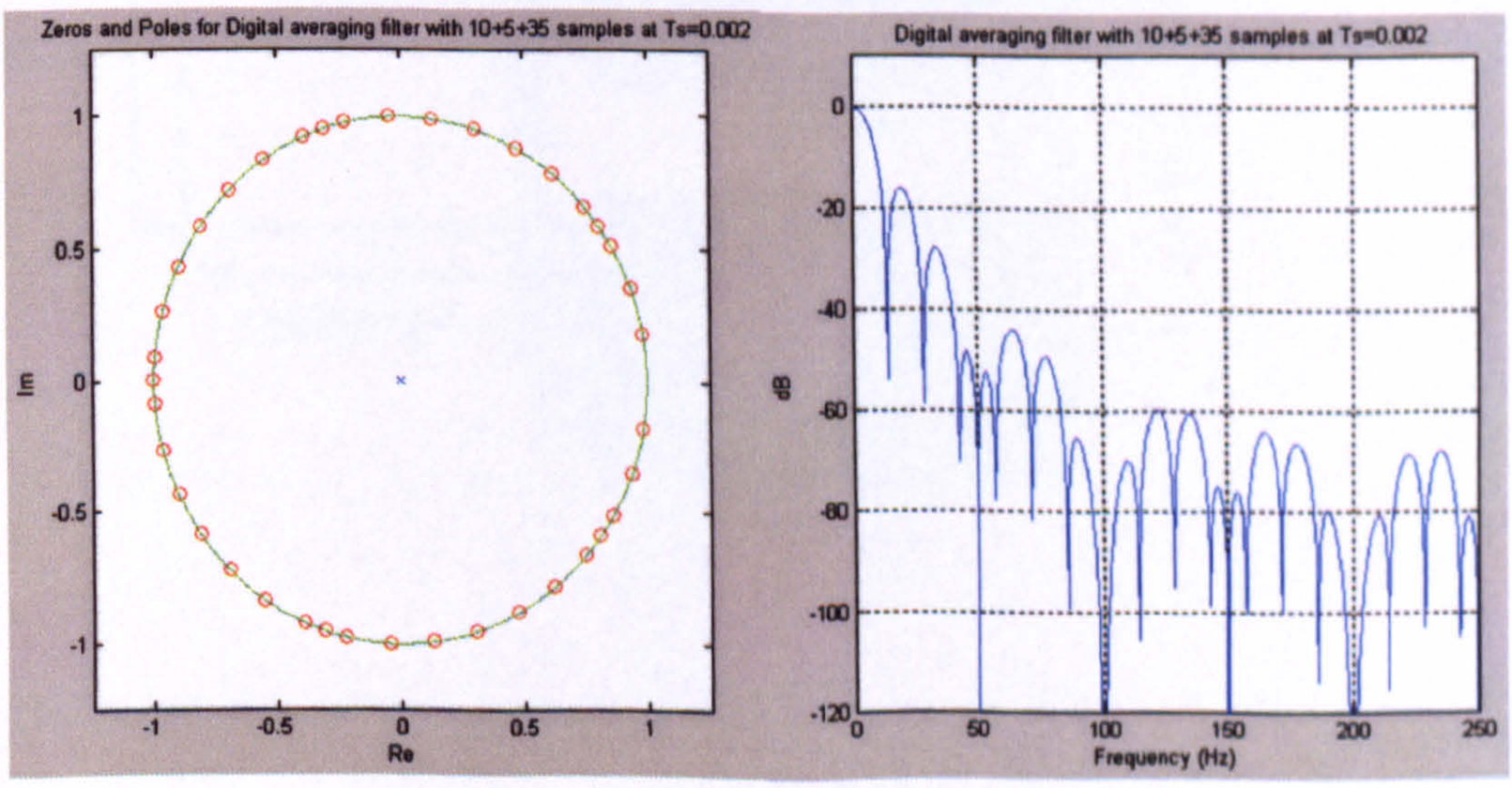


Fig. D-3 : 1+0.5+3.5 cycle averaging

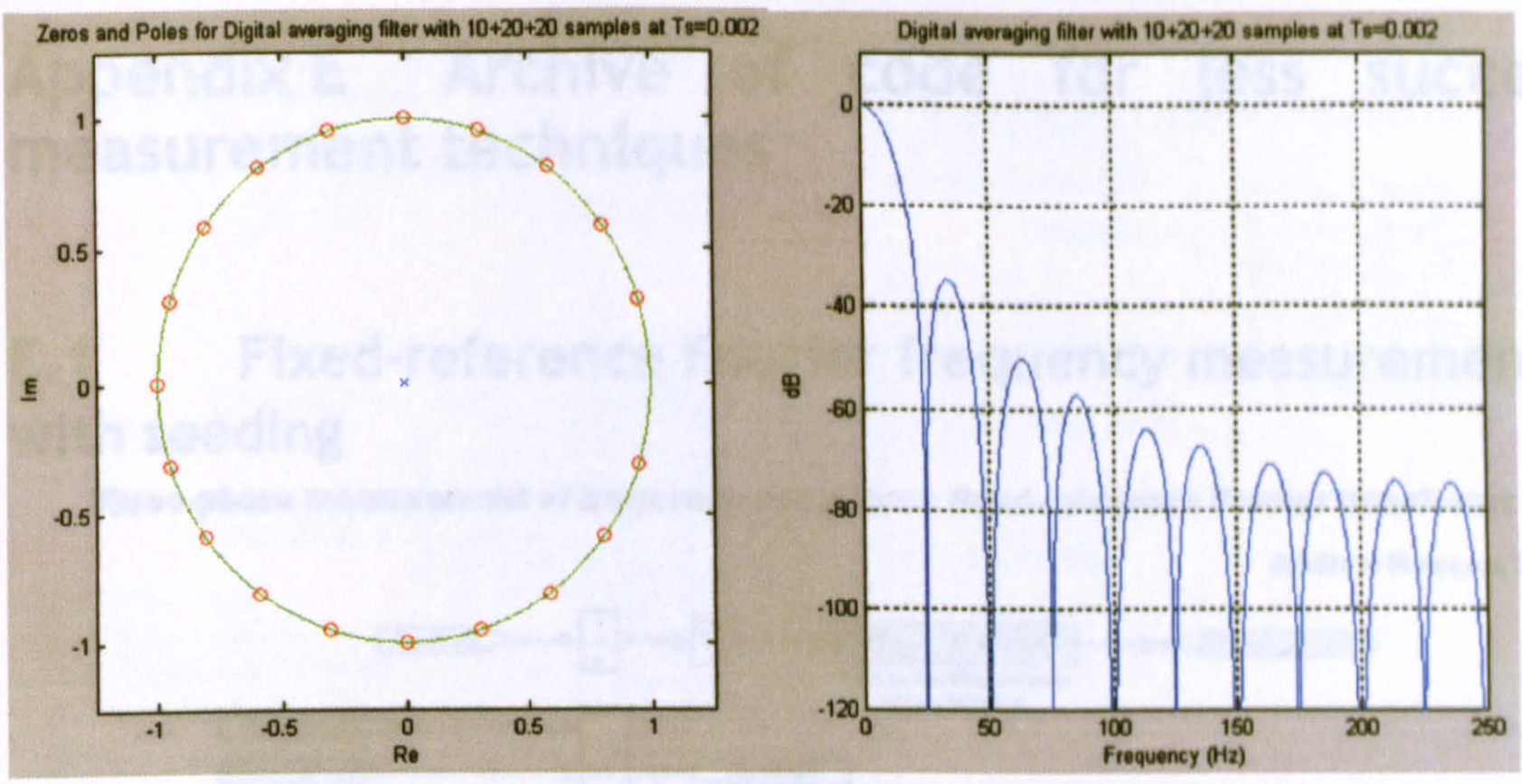


Fig. D-4 : 1+2+2 cycle averaging

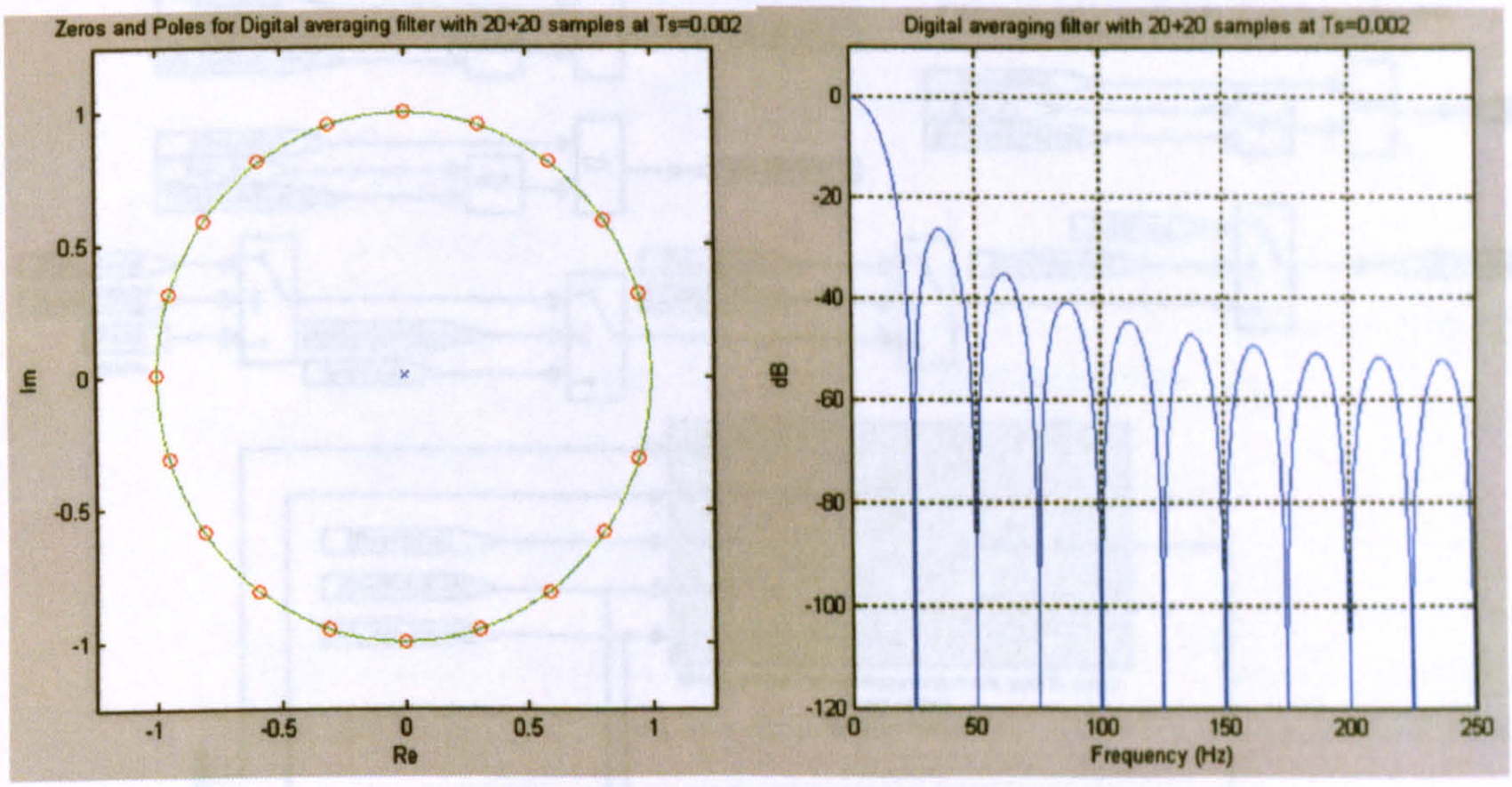
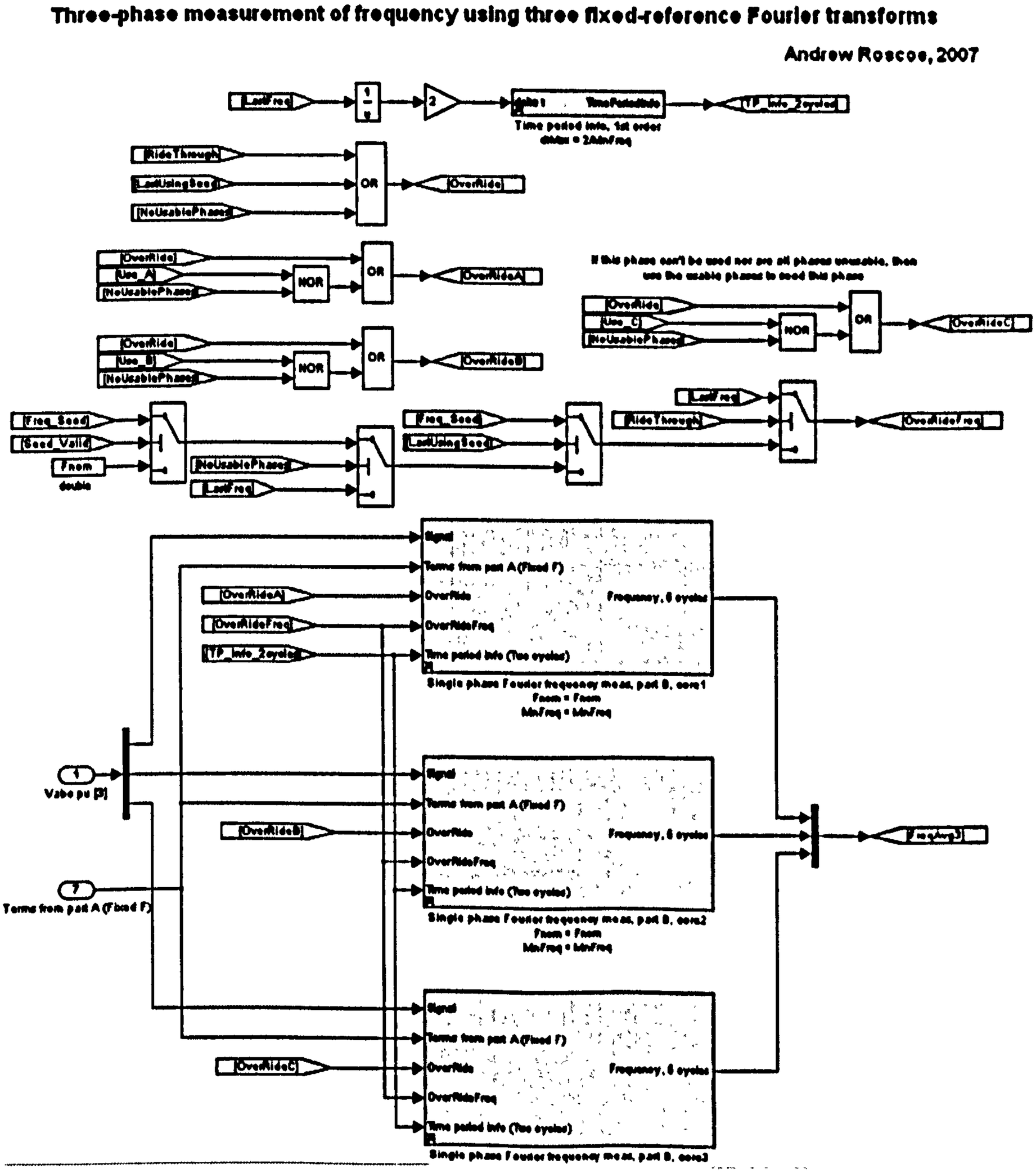
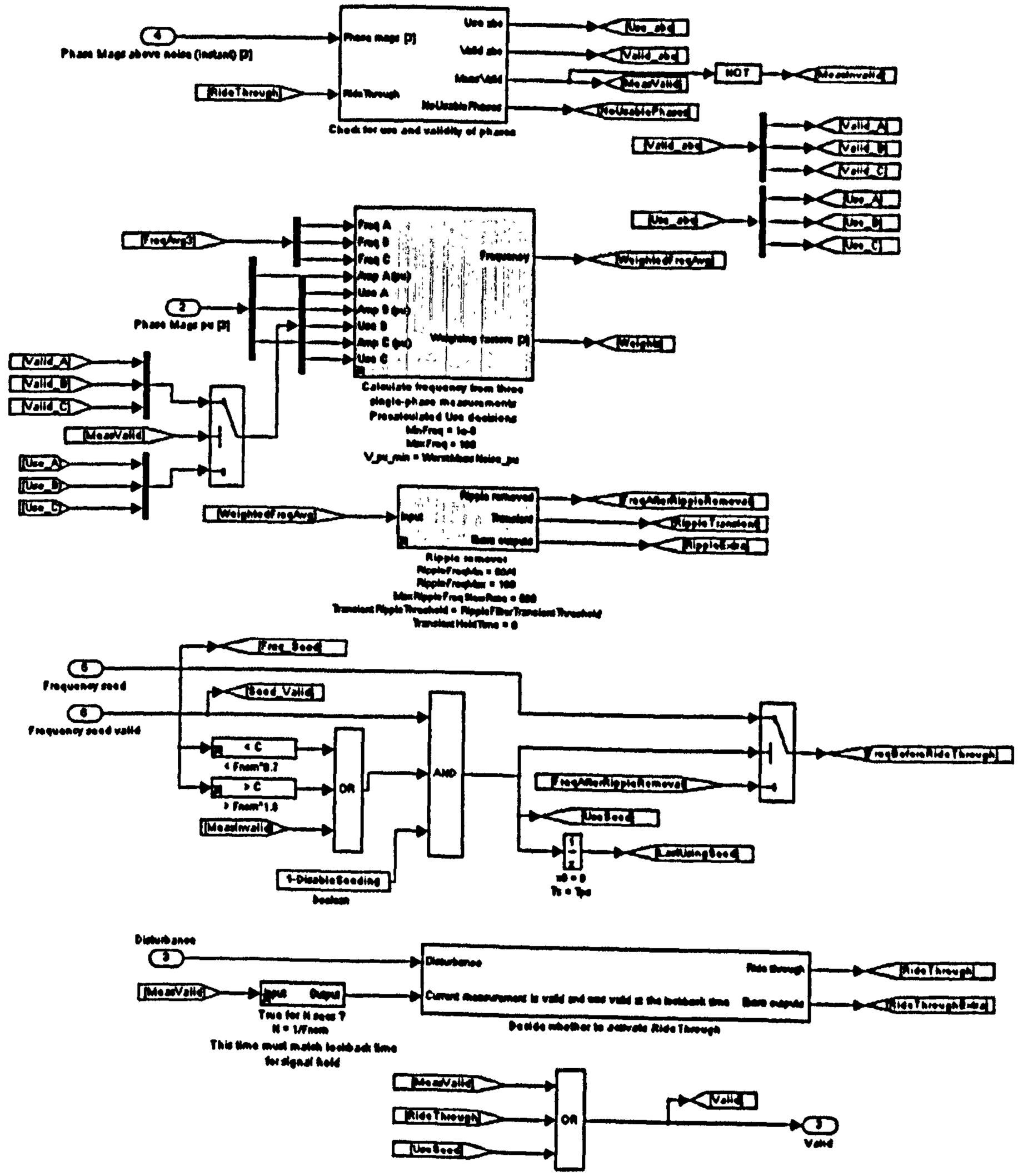


Fig. D-5 : 2+2 cycle averaging

Appendix E Archive of code for less successful measurement techniques

E.1 Fixed-reference Fourier frequency measurement with seeding




4
Phase Mags above noise (instant) [3]
Phase mags [3]
RideThrough
Use abc
Valid abc
MeasValid
NoUsablePhases
RideThrough
Check for use and validity of phases
NOT
Valid_abc
Valid_A
Valid_B
Valid_C
Use_abc
Use_A
Use_B
Use_C
FreqAvg3
2
Phase Mags pu [3]
Freq A
Freq B
Freq C
Amp A (pu)
Use A
Amp B (pu)
Use B
Amp C (pu)
Use C
Frequency
Weighting factors [3]
WeightedFreqAvg
Weights
Calculate frequency from three single-phase measurements
Precalculated Use decisions
MinFreq = 1e-8
MaxFreq = 100
V_pu_min = WorstMeasNoise_pu
Valid_A
Valid_B
Valid_C
MeasValid
Use_A
Use_B
Use_C
WeightedFreqAvg
Ripple removed
Input
Transient
Same outputs
FreqAfterRippleRemoved
RippleTransient
RippleEnbd
Ripple removed
RippleFreqMin = 60/1
RippleFreqMax = 100
Max Ripple Freq Slew Rate = 600
Transient Ripple Threshold = RippleFilterTransientThreshold
Transient Hold Time = 0
6
Frequency seed
6
Frequency seed valid
Freq_Seed
Seed_Valid
< C
< From"0.7
> C
> From"1.0
MeasInvalid
OR
AND
1-DisableSeeding
boolean
FreqAfterRippleRemoved
UseSeed
1
z
n0 = 0
Ts = Ts
FreqBeforeRideThrough
Disturbance
3
MeasValid
True for N secs ?
N = 1/Fnom
This time must match lookback time for signal hold
Disturbance
Current measurement is valid and was valid at the lookback time
Ride through
Same outputs
Decide whether to activate Ride Through
RideThrough
RideThroughEnbd
MeasValid
RideThrough
UseSeed
OR
Valid
3
Valid

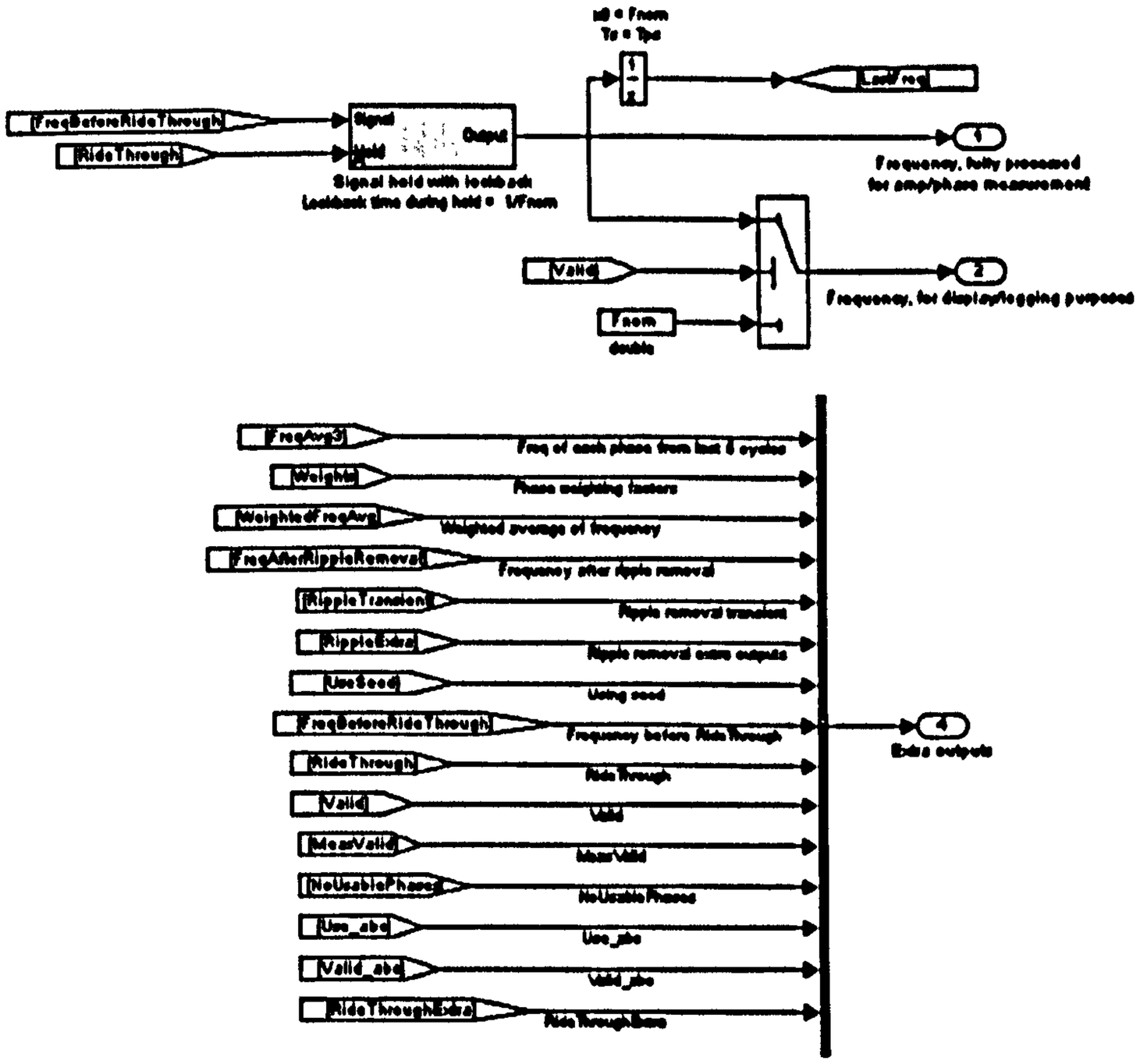


Fig. E-1 : Fixed-reference Fourier frequency measurement - detail (1) spread over this and previous 2 pages

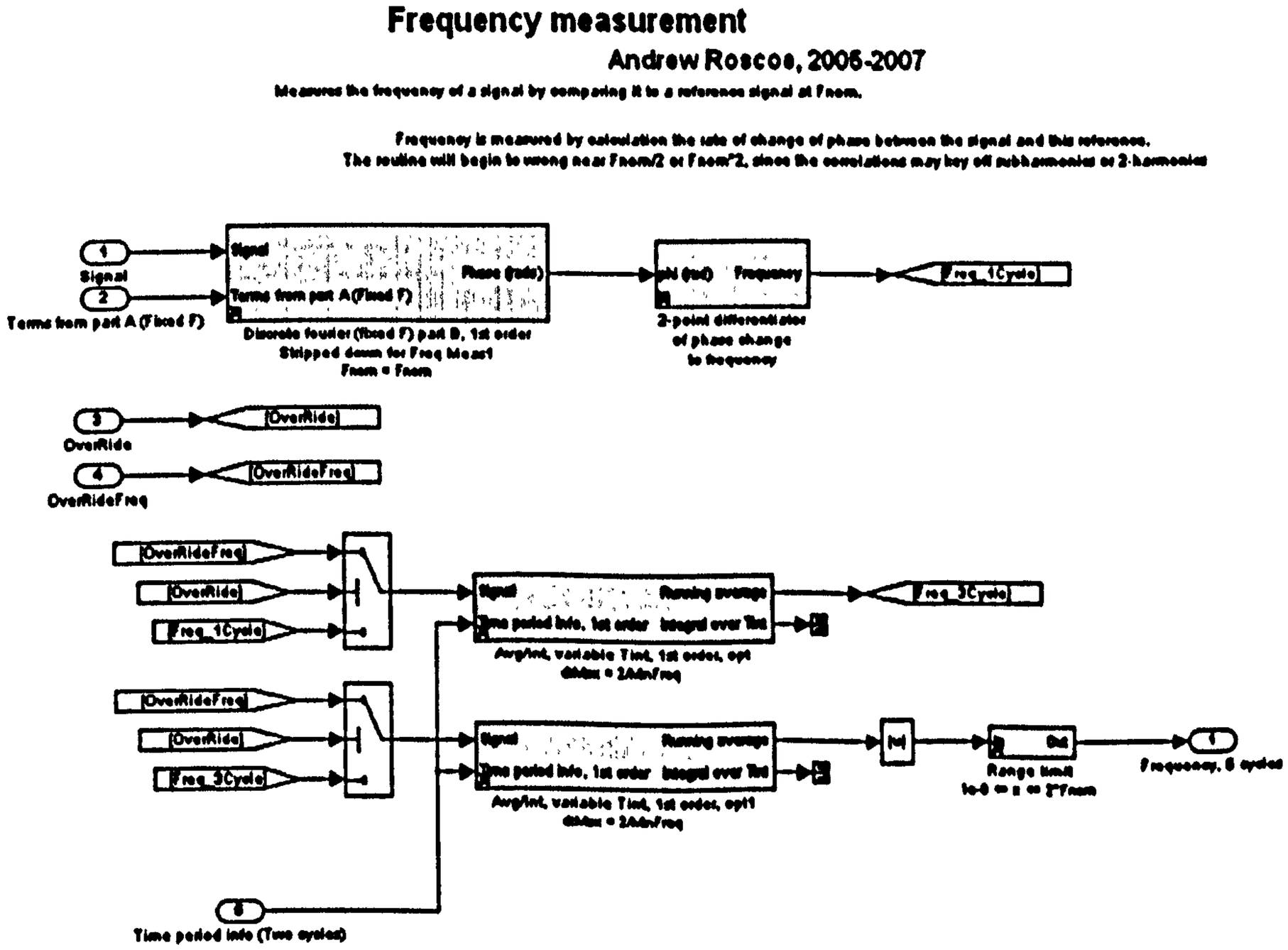
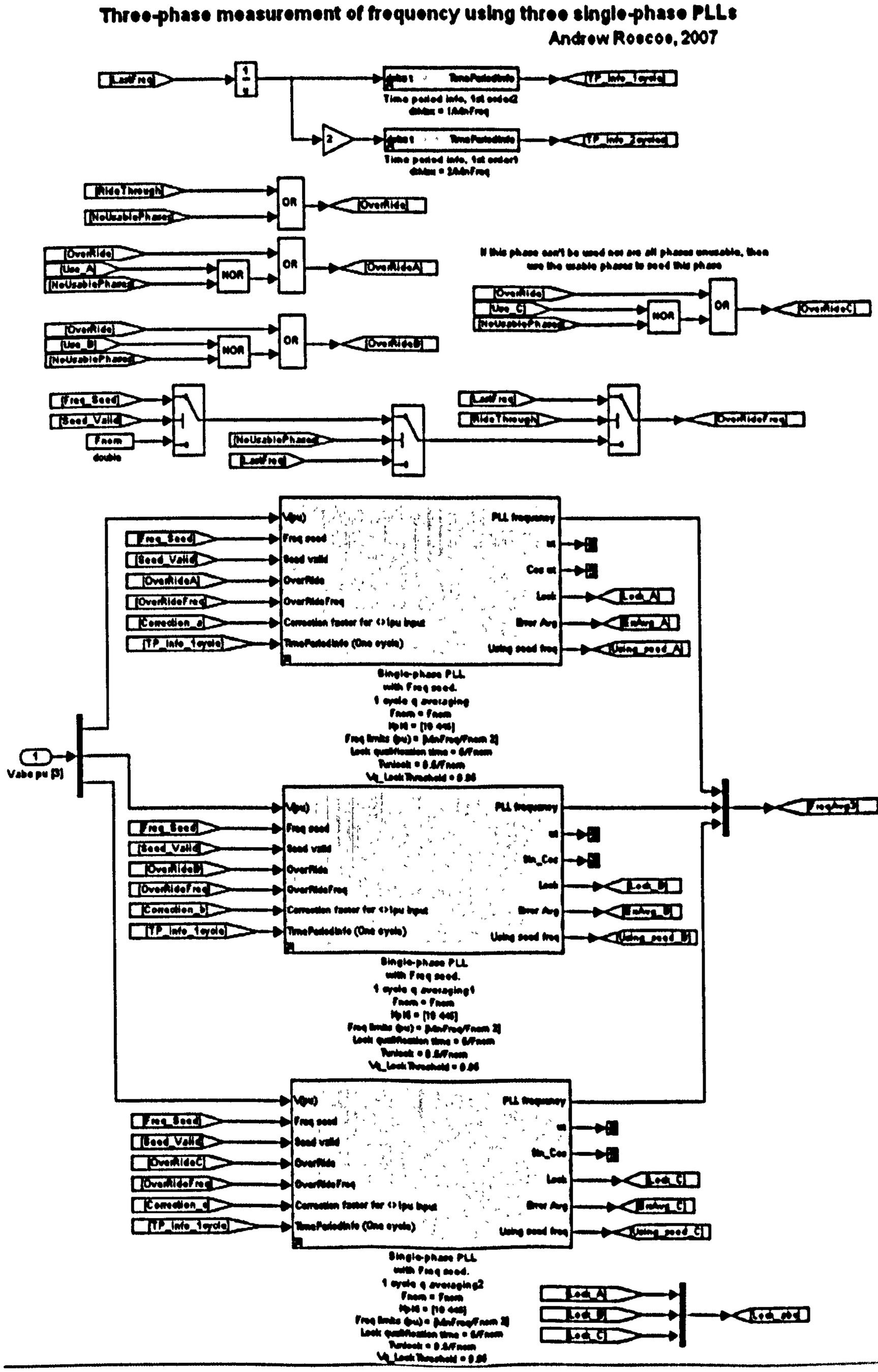
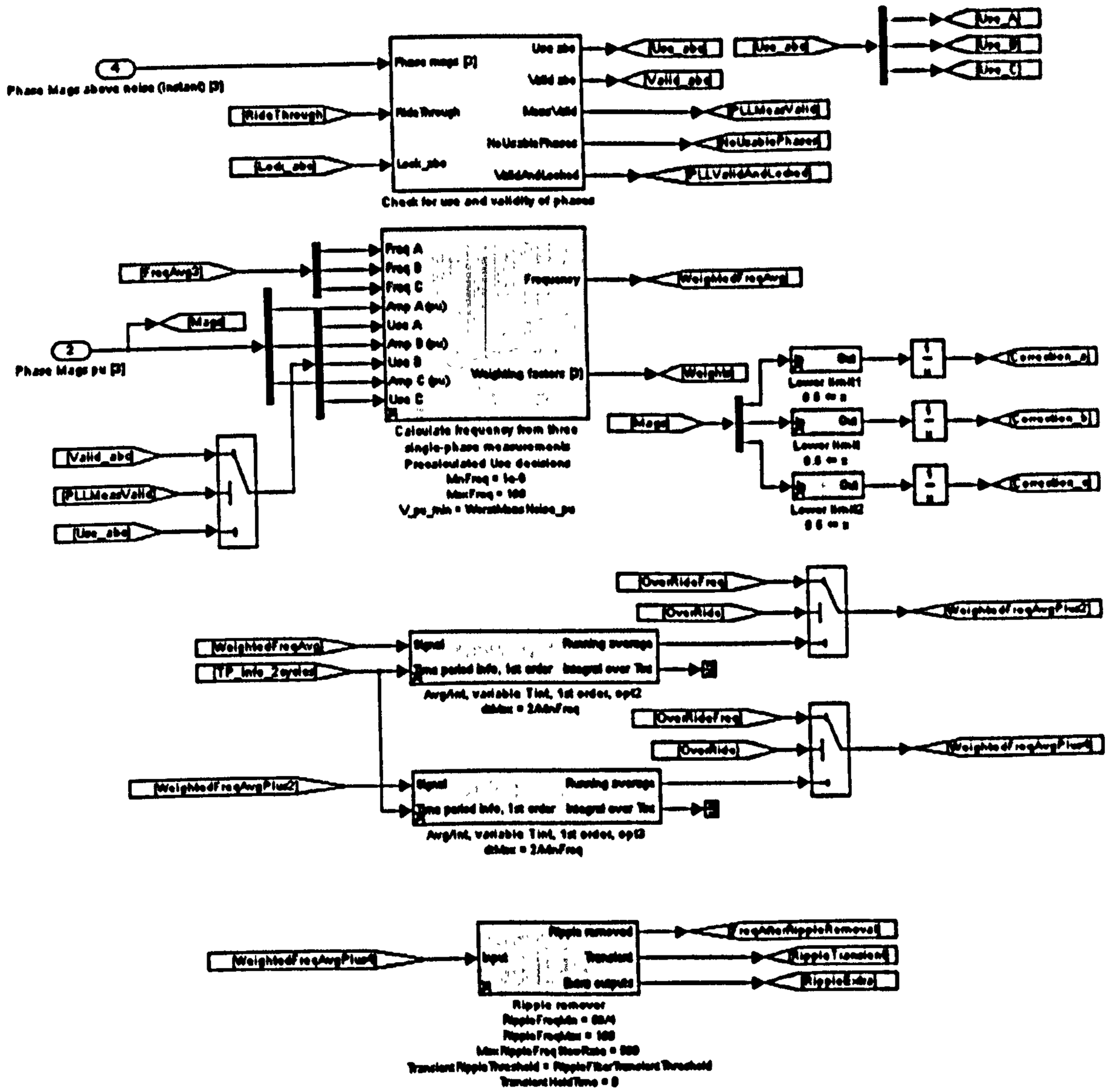


Fig. E-2 : Fixed-reference Fourier frequency measurement - detail (2); core

E.2 Three customised single-phase PLLs with weighted averaging and seeding




Phase Mags above noise (instant) [3]
RideThrough
Lock_abc
Check for use and validity of phases
Freq Avg3
Mags
Phase Mags pu [3]
Valid_abc
PLLMeasValid
Use_abc
Calculate frequency from three single-phase measurements
Precalculated Use decisions
MinFreq = 1e-6
MaxFreq = 100
V_pu_min = WorstMeasNoise_pu
Weighted Freq Avg
TF_Info_2cycles
OverRideFreq
OverRide
Running average
Time period info, 1st order Integral over Tbs
AvgInt, variable Tint, 1st order, opt2
dtMax = 2/MinFreq
Ripple remover
Ripple Freq Min = 60/4
Ripple Freq Max = 100
Max Ripple Freq SlewRate = 500
Transient Ripple Threshold = RippleFilterTransient Threshold
Transient HoldTime = 0

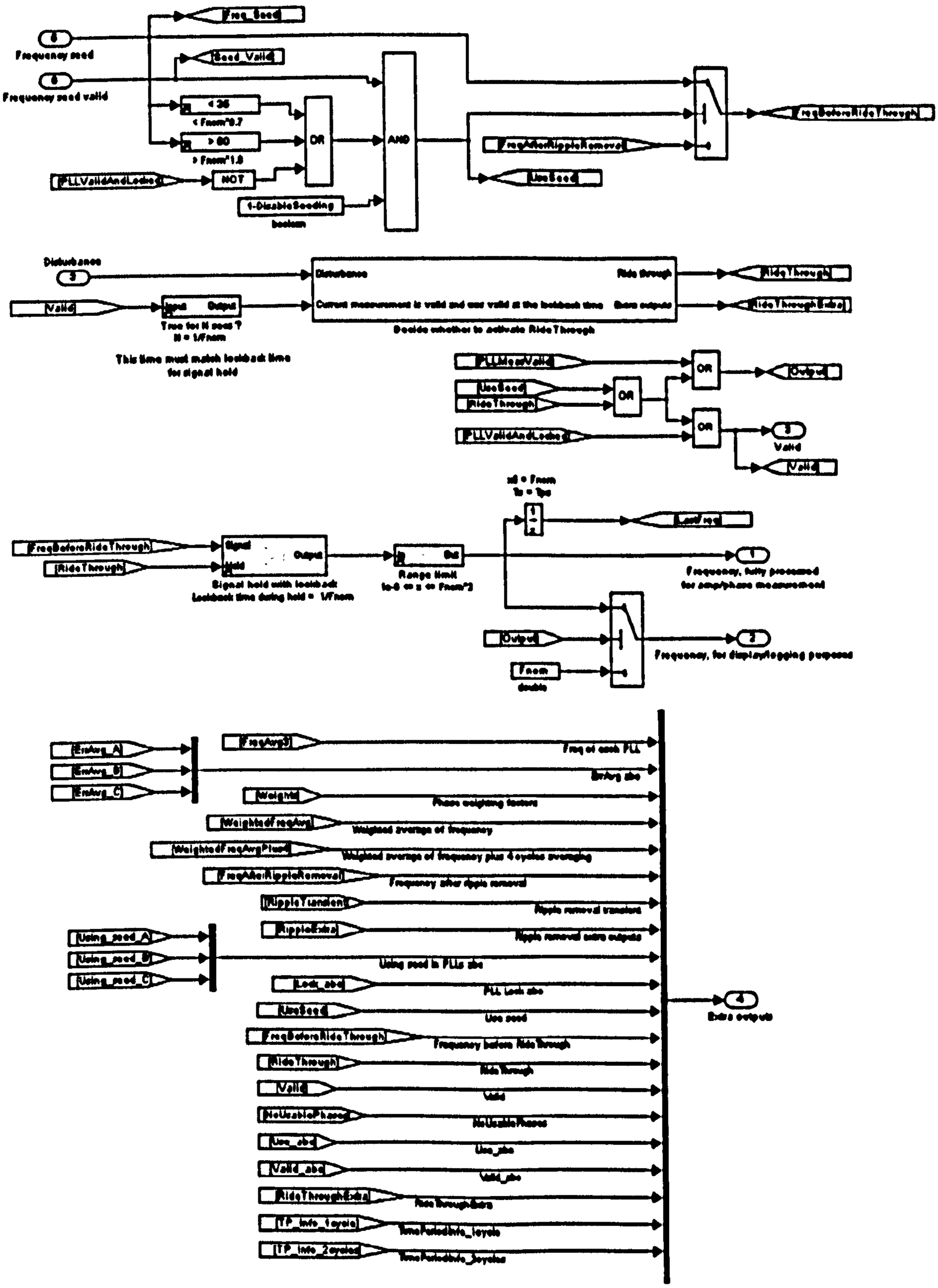


Fig. E-3 : Three single-phase PLLs with weighted averaging and seeding - detail (1) spread over this and previous 2 pages

1-phase PLL, 1 cycle measurement, with external frequency seed for large frequency steps

Note: sin(a) * cos(b)= 1/2 * sin(a+b) + 1/2 * sin(a-b)

Andrew Roscoe, 2007

The Frequency seed should come from a fast responding, robust frequency measurement algorithm such as Clarkes.
The seed validity should be low when appropriate (e.g. during two-phase faults if Clarkes is used as the seed)
To use without seed, set valid LOW at all times.

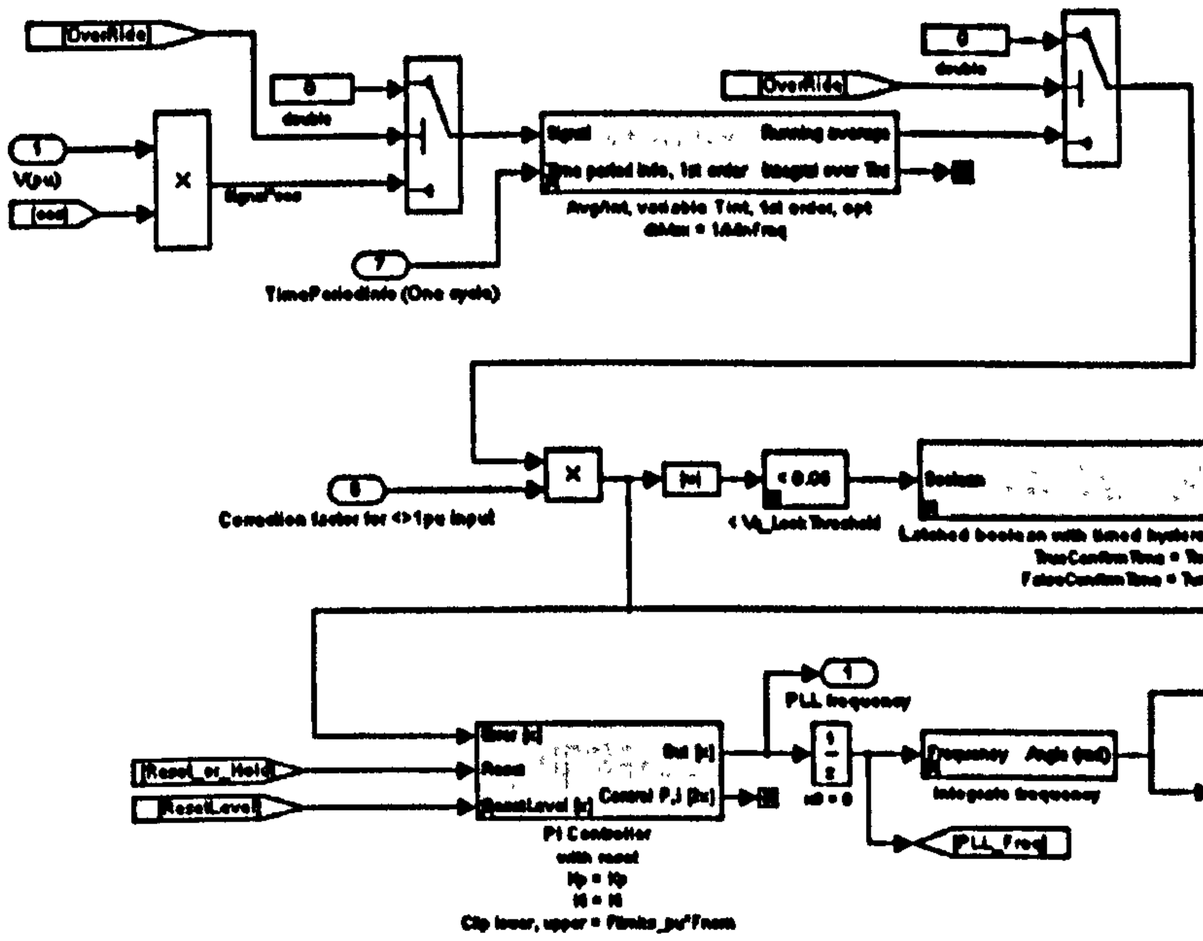


Fig. E-4 : Three single-phase PLLs - detail (2); core

Determine whether the PLL needs seeding (possible harmonic/subharmonic lock, or far from lock)
Ensure PI controller is loaded with the RideThrough frequency when fault ride-through is active

Determine whether the PLL needs reseeding:

If real frequency (the seed) is F0, and the PLL frequency is F, and we have N cycles averaging for the q axes, then:

Andrew Roscoe, 2007

Over the N cycles, the dq trajectory will move (F0-F).N/F revolutions over the N cycles. (averaging time N/F).
This is ((F0/F) - 1).N revolutions.
The absolute value of this must be less than 1/4 for us to sensibly track the frequency.
Hence, limits on F are:

F > F0/(1+1/4N)
F < F0/(1-1/4N)

In the algorithm above, N is set to 1
F > F0.4/5
F < F0.4/3

If it is not within these limits, then it should be set to F0

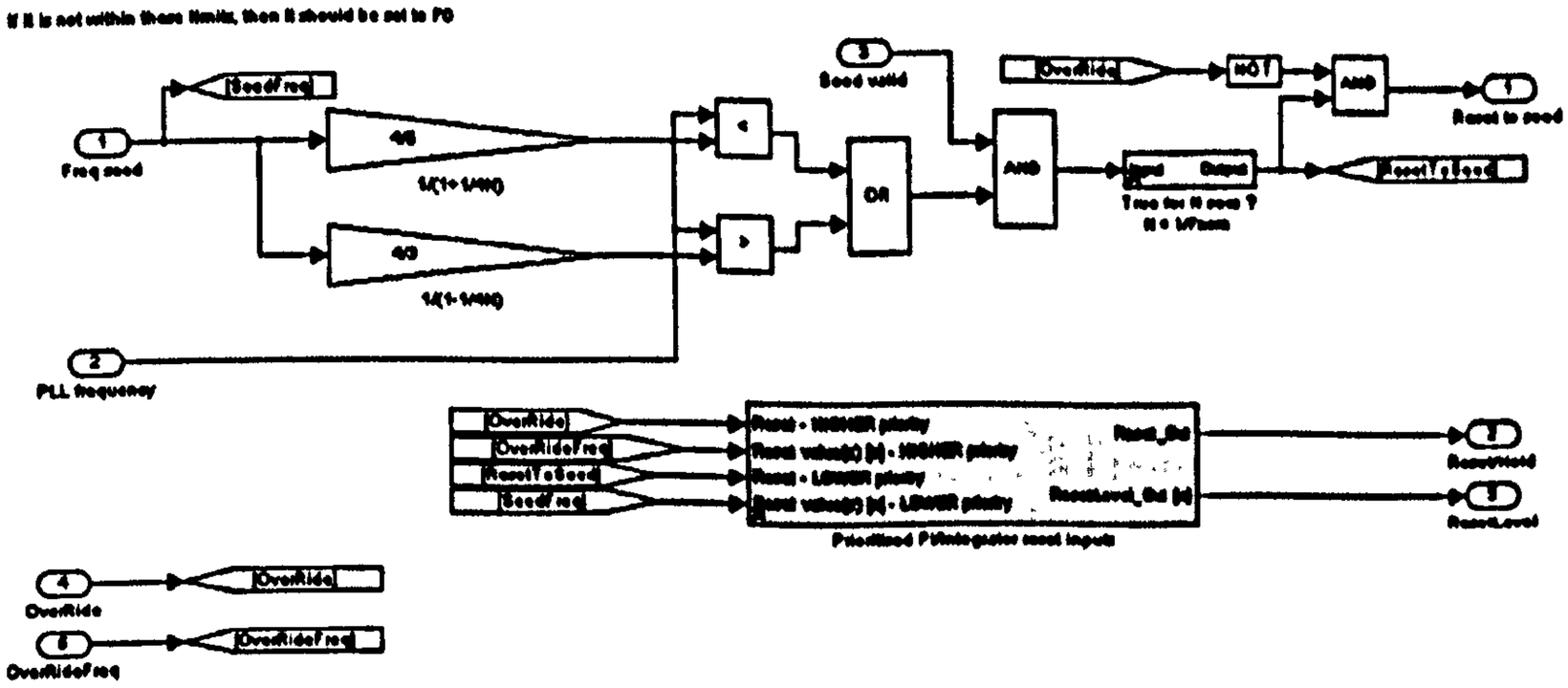
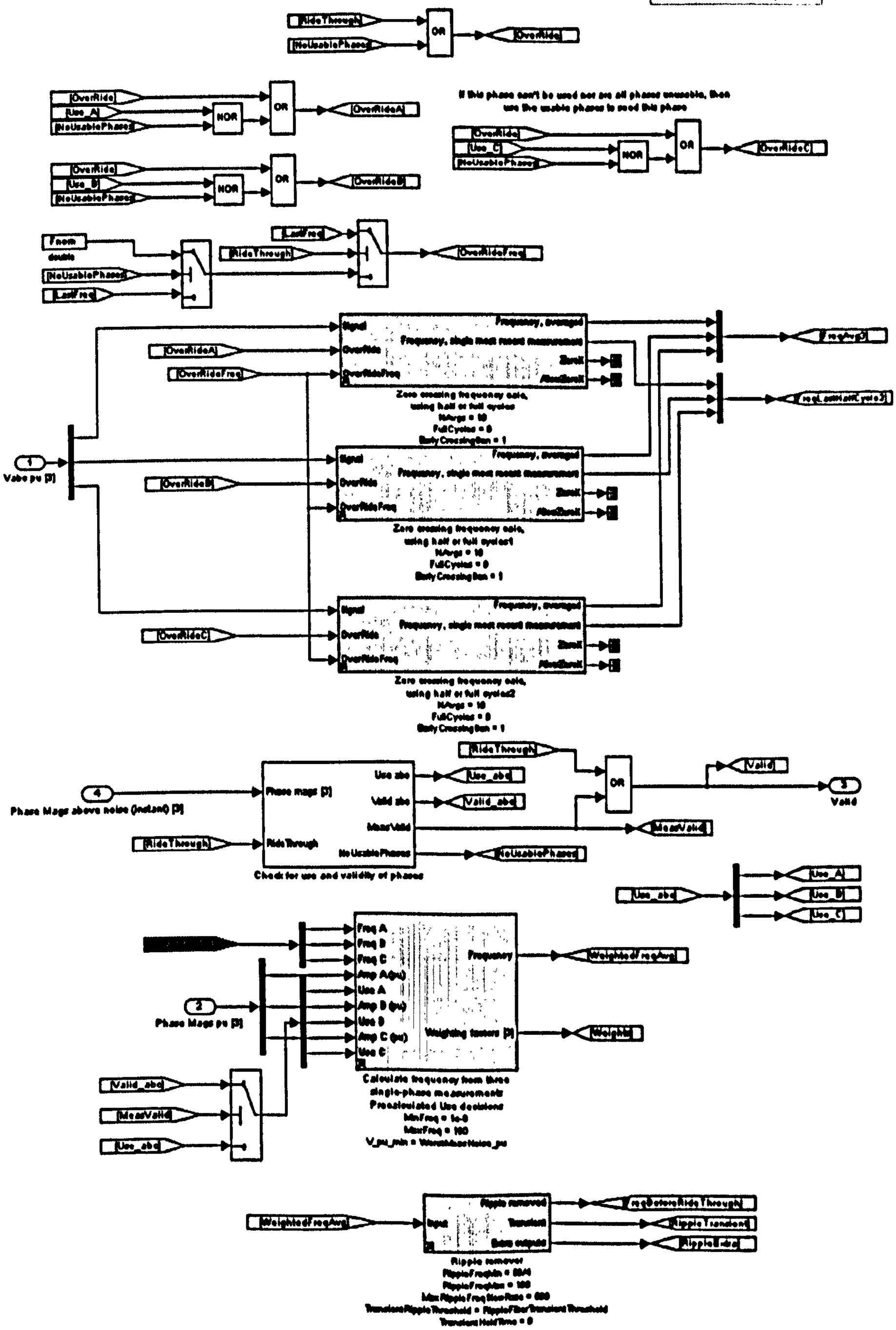


Fig. E-5 : Three single-phase PLLs - detail (3); core PLL seeding decision

E.3 Zero crossings

Three-phase measurement of frequency using zero crossings

Andrew Roscoe, 2007



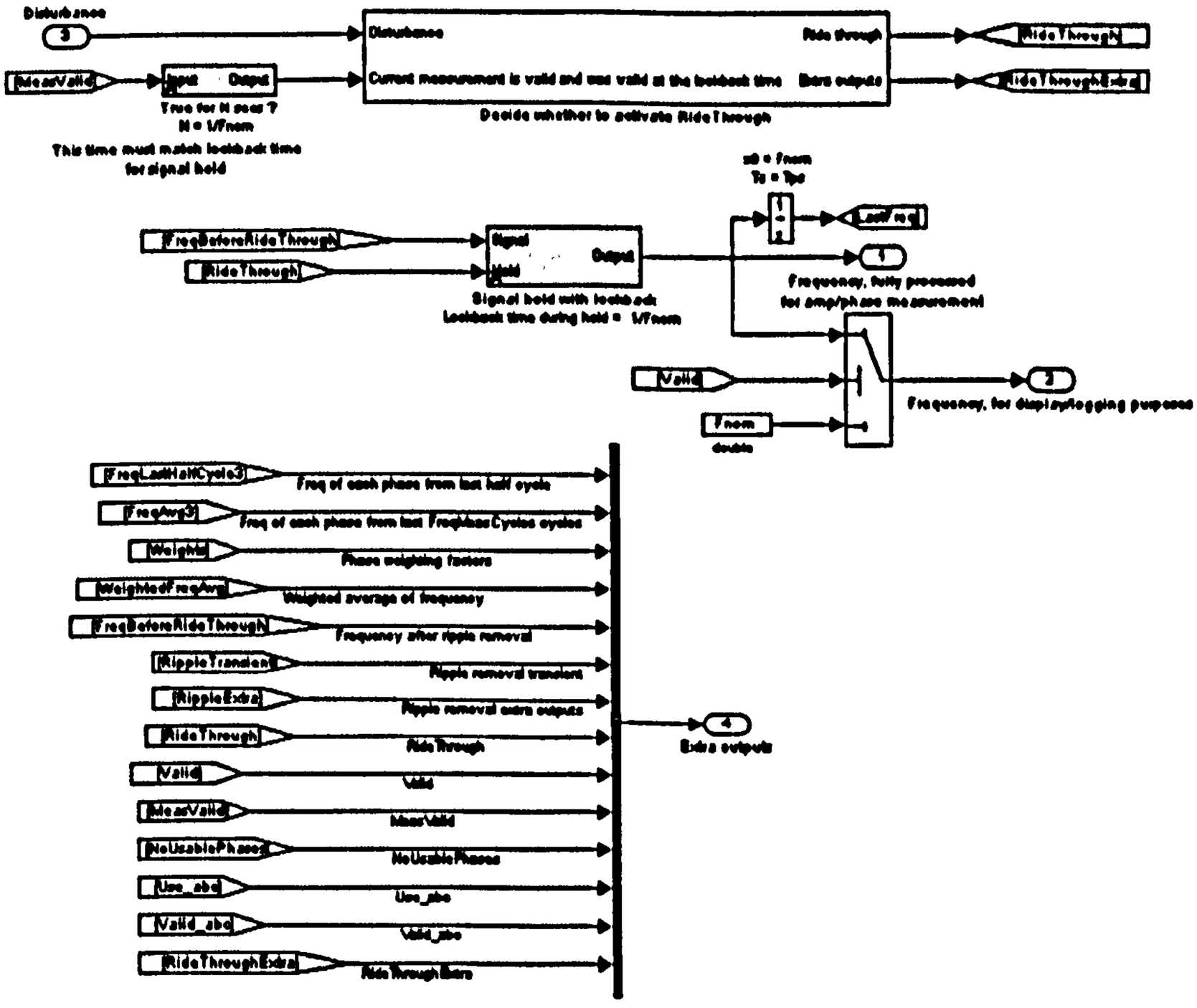


Fig. E-6 : Zero crossing measurement with weighted averaging - detail (1) spread over this and previous page

Frequency measurement by zero crossings over half-cycles or full-cycles

Andrew Roscoe, 2007

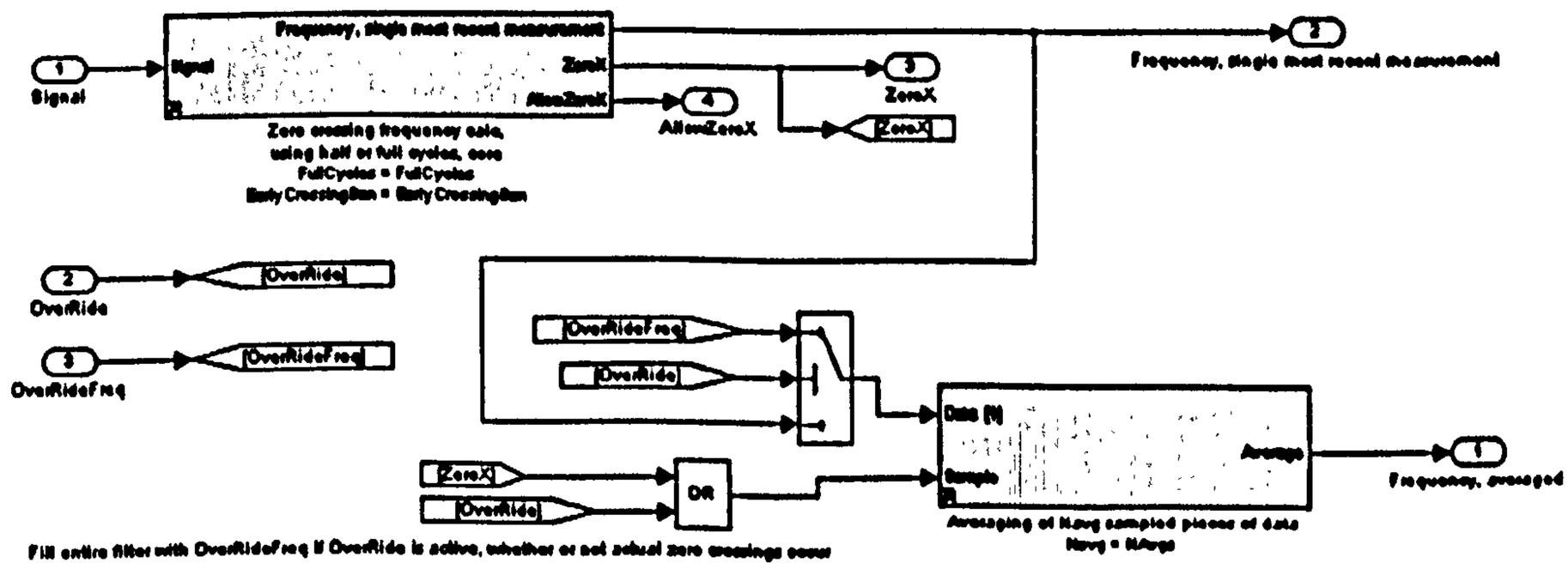


Fig. E-7 : Zero crossing measurement with weighted averaging - detail (2); core

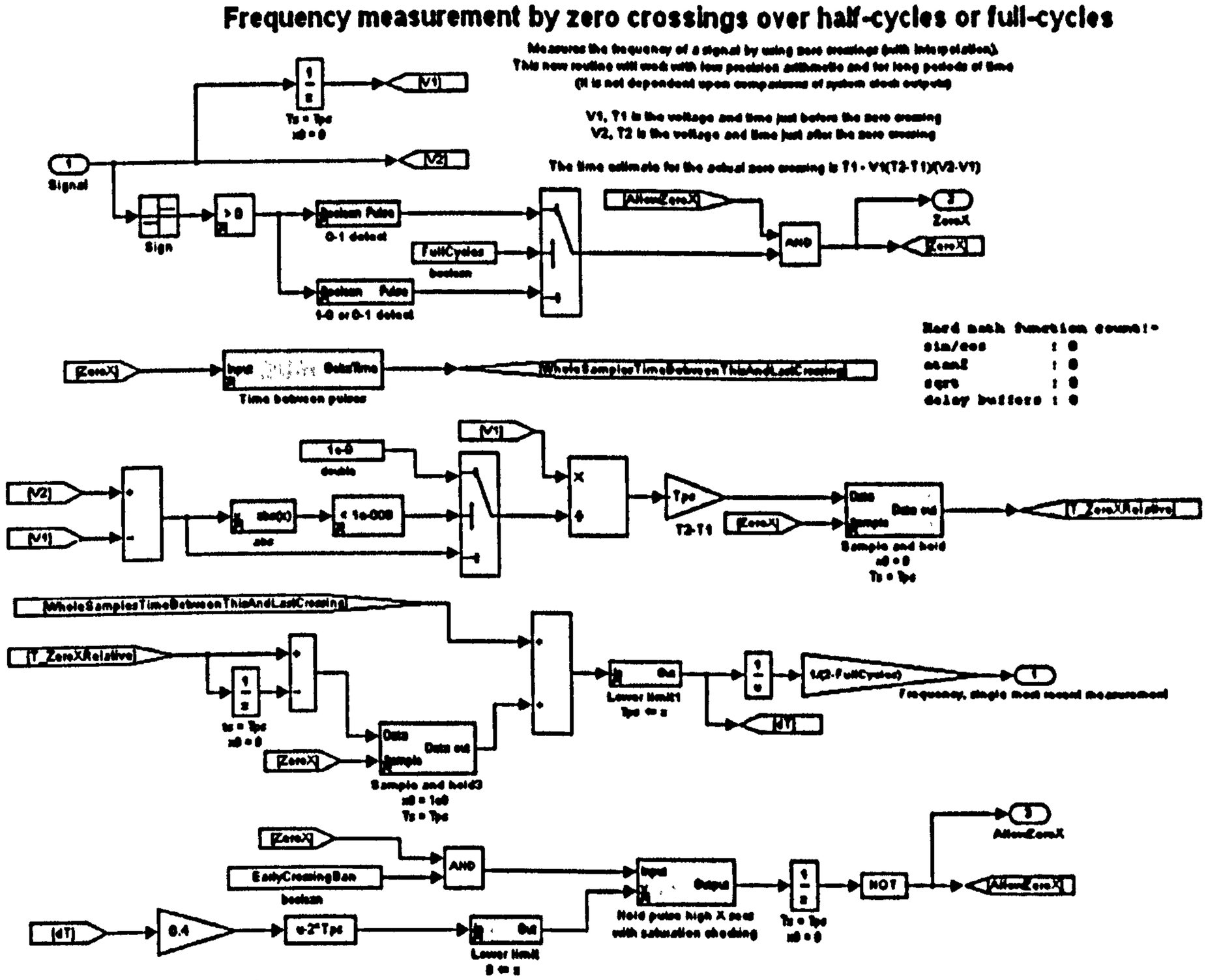


Fig. E-8 : Zero crossing measurement - detail (3); inner core

Appendix F Examination of the use of sine/cosine lookup tables

To decrease the CPU loading of Fourier calculations, one potential option is to use lookup tables for the evaluations of sine and cosine. This removes the need for the complex mathematical function, at the expense of fast-access RAM use, plus the code required to access the lookup table and interpolate.

To assess the viability of this option, two experiments were carried out:-

1. An analysis of the errors introduced to the 1-cycle and "One plus half" cycle Fourier routines from sections 3.6 and 3.9, by using lookup tables of different sizes.
2. An analysis of the times required for the sine/cosine operations, versus the lookup table options, using the Infineon TC1796 microcontroller.

F.1 Errors introduced due to Sin/Cos lookup tables

The lookup tables in this error analysis are implemented as single-quadrant tables. For example, if the lookup table size is 46, then the lookup table contains 46 pre-calculated values of sin(φ) where φ is [0,2,4,6 ... 86,88,90] degrees, with 90/(46-1)=2 degree steps. Results for both sine and cosine evaluations over the full 4 quadrants can be pulled from this table by careful coding, which includes linear interpolation between the most appropriate tabulated values.

The experiments of sections 3.7 and 3.9 were repeated, with the modification that lookup tables of various sizes were used, while sample rate was held at 10 samples per cycle. Only the 1st order Fourier algorithms were analysed.

The conclusion is that the magnitude outputs of the single-cycle Fourier calculation have a mostly DC error term added due to the use of the lookup tables. This can be seen because the ripple frequencies shown in Fig. F-2 are not multiples of the input frequency, but are 0. The effect this has on the overall RMS error magnitudes for the 1-cycle and "One plus Half" cycle measurements are shown in Fig. F-1 and Fig. F-3 . The proportionate effect on the 1-cycle measurements becomes very small once the lookup table size is more than 20 (the lower limit of error is limited by the interpolation/integration error at 10 samples/cycle which is shown in Figure 3-45), but note that this error cannot be removed by the extra half-cycle averaging since it is mostly a DC error term. Thus, to determine an appropriate table size, Fig. F-3 must also be examined. This suggests that a size of 46 (2

degrees per step) reduces the DC error term due to the lookup table to less than 0.0002 pu which is perfectly acceptable. A table of size 21 (4.5 degrees per step) would result in an error of only 0.001pu which is just about acceptable. The maximum errors when evaluating sin(φ) and cos(φ) using an interpolated table with steps of 4.5° are a maximum of $8e^{-4}$. If a non-interpolated table was used, the required step spacing to achieve the same magnitude of error can be calculated by knowing that sin(φ)≈φ when φ is small. Hence, the step spacing could be double the maximum error of $8e^{-4}$, i.e. $1.6e^{-3}$ radians, or 0.1°. By coincidence, this same step spacing would be the requirement for accuracy when calculating the positive and negative sequence values, to keep unbalance measurements accurate to about 0.1%. This is because a set of 3 phase voltages with identical magnitudes, but relative phases of 0°, -120.1° & -239.9° (i.e. with 0.1° phase errors - real or measured), results on a calculated unbalance of 0.1%.

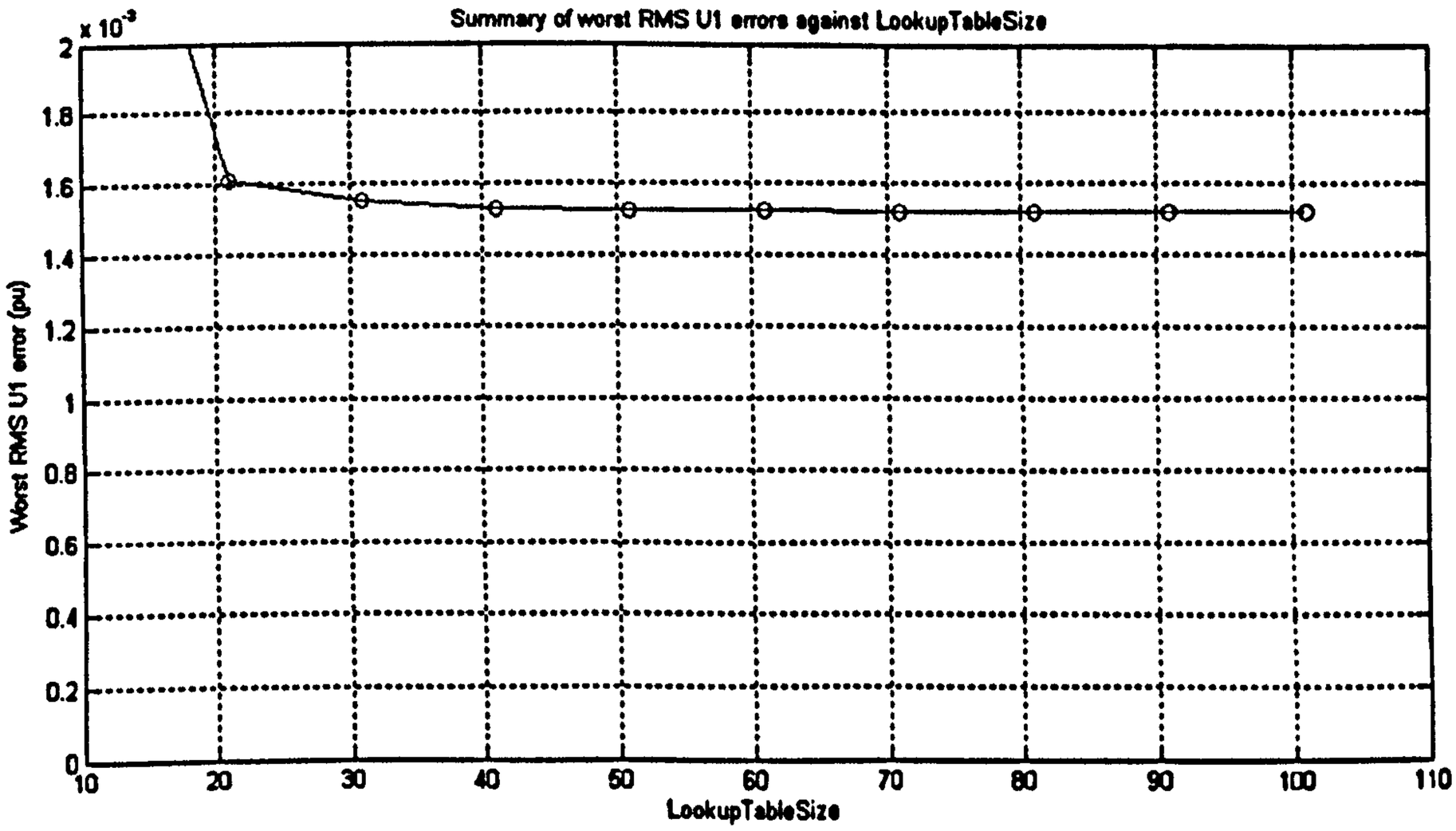


Fig. F-1 : Fourier analysis of fundamental. Largest RMS errors due to Sin/Cos lookup plus integration & interpolation over the 45-55Hz range. Single cycle Fourier analysis.10 samples/cycle, 1st order method.

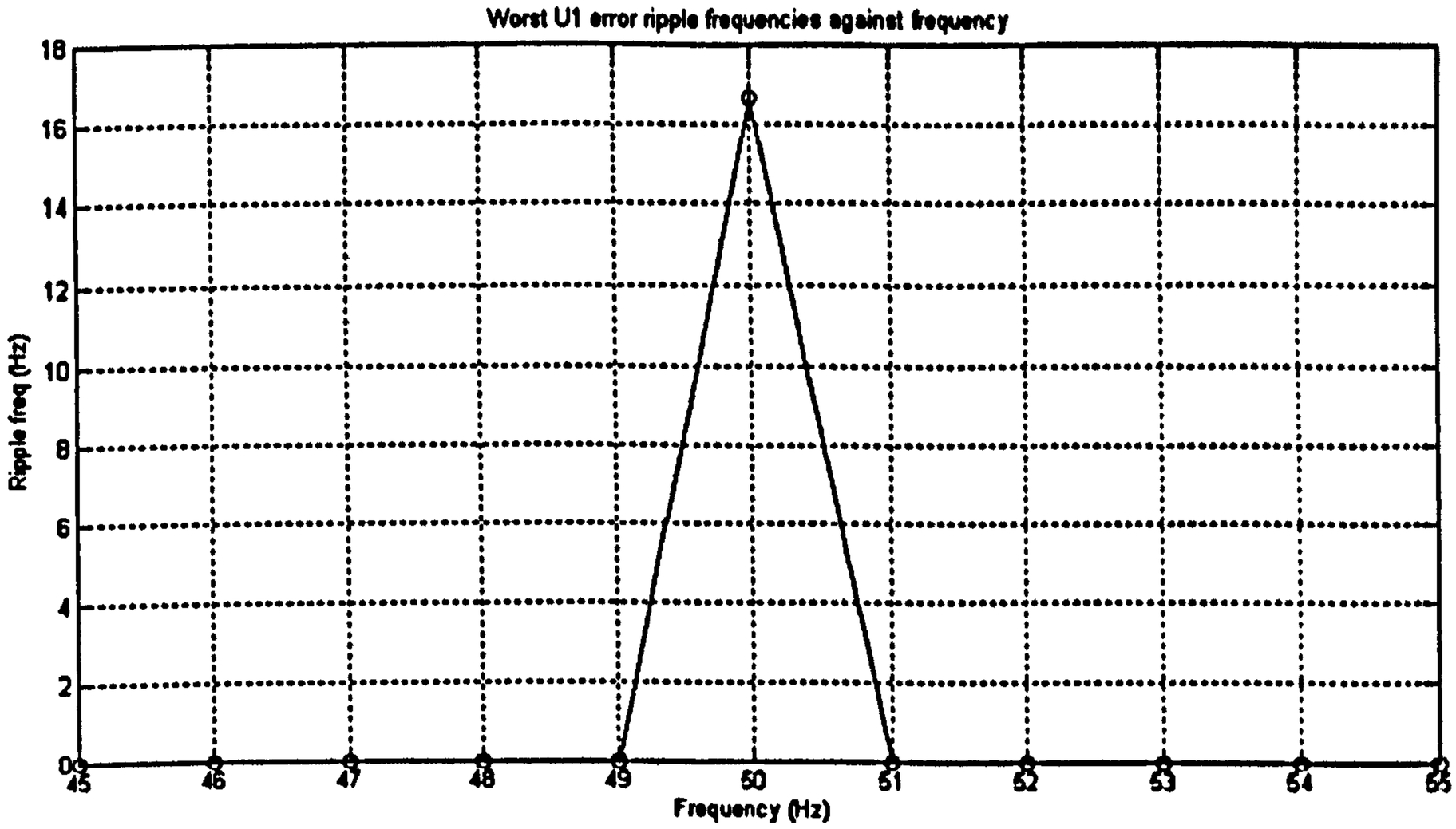


Fig. F-2 : Fourier analysis of fundamental. Ripple frequencies due to Sin/Cos lookup plus integration & interpolation. Lookup table size 11 (9 degree steps). Single cycle Fourier analysis.10 samples/cycle, 1st order method.

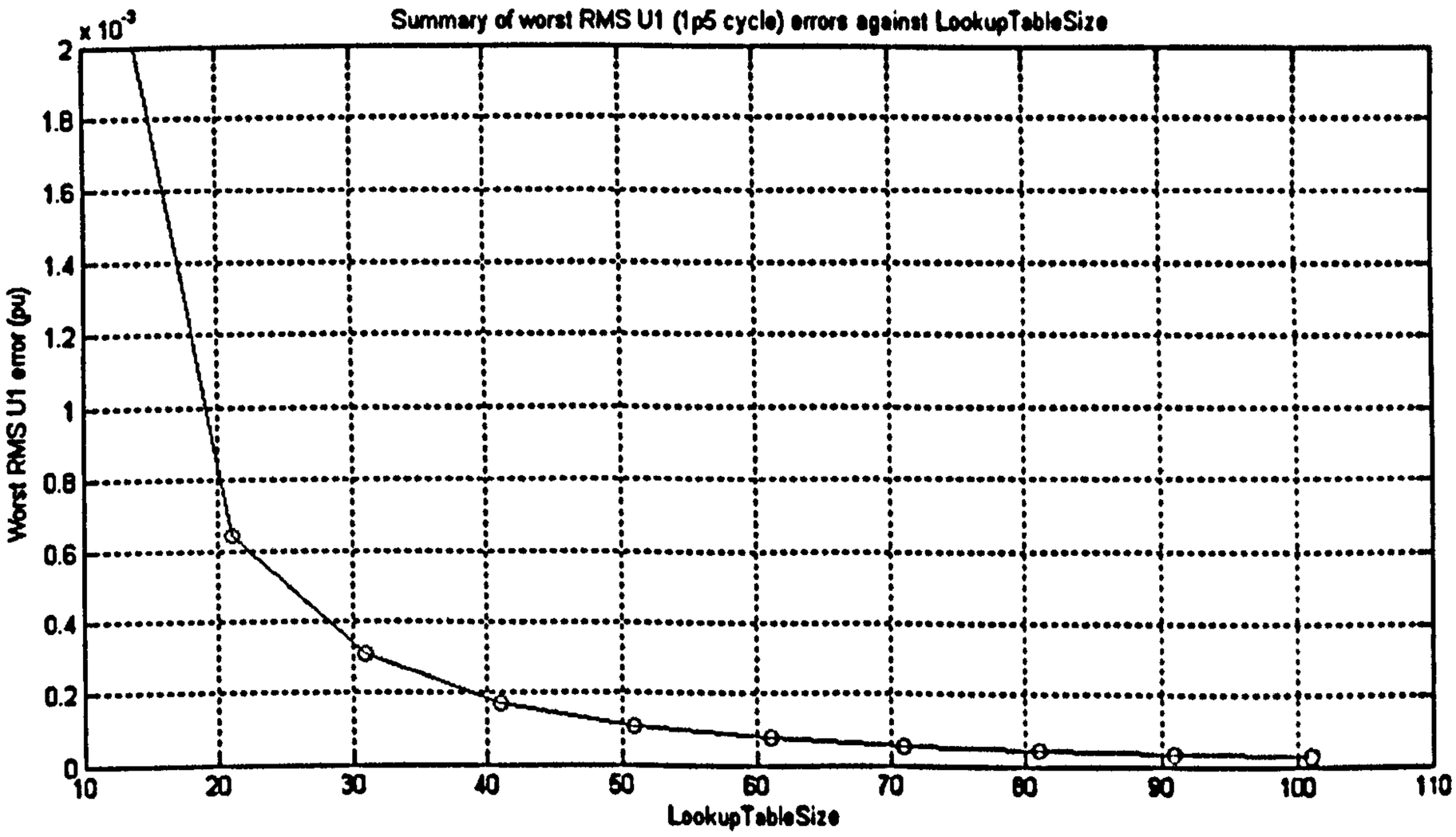


Fig. F-3 : Fourier analysis of fundamental. Largest RMS errors due to Sin/Cos lookup plus integration & interpolation over the 45-55Hz range. Single cycle Fourier analysis plus half cycle averaging.10 samples/cycle, 1st order method.

F.2 Analysis of relative execution times of Sin/Cos vs lookup tables on Infineon TC1796 microcontroller

This is addressed in section Appendix G.

Appendix G Speed benchmarking on the TC1796 microcontroller and ADI RTS

In Appendix F, the use of sine/cosine lookup tables was proposed as a potential method to reduce the computation time of the Fourier algorithms. To find out whether the improvement could be realised, and to measure the execution times of other pieces of code, a benchmarking framework was created. The benchmark framework can be applied to two types of target, relevant to the ongoing work at the University of Strathclyde:-

- An Infineon TC1796 microcontroller. The benchmark framework was created by stripping out an existing inverter control application to leave only the bare bones of an interrupt-driven code segment at a 4000Hz repetition rate (250μs frame time). This can be reduced to 1000Hz (1000μs frame time) to test very big blocks. Inside this code segment, different pieces of evaluation code can be placed, and repeated in a loop. The number of loop iterations is set so that the execution time approaches 250μs or 1000μs. This makes the execution time measurable using only a basic oscilloscope, coupled to an output pin which the microcontroller toggles at the beginning and end of the loop execution. A simple reference piece of code is used inside the loop the first time, to back the overhead of the looping code out of the measurement. The pieces of code under test are added to this setup in turn, to measure the incremental execution times.
- An ADI RTS real-time station platform. In this application, the execution times for large blocks can be measured by compiling the code segments on to the target processor (either the ce5100 or ce5500 variety - CPU clock speeds 500MHz and 1GHz respectively). The ADI RTS variable ADI_SUBSYS_CUR_TIME[1] can be used to evaluate the incremental execution time when adding code segments.

All the tests presented use code created in Simulink, and auto-generated into C code using the "real-time workshop" and "embedded coder" MATLAB plug-ins. This usually generates very efficient code, but some exceptions of interest are noted here. The results are tabulated in Fig. G-1 to Fig. G-3.

In the case of the Infineon TC1796 microcontroller, The CPU clock speed was 150MHz. The code was initially loaded and executed directly from the internal flash memory at A0000000, which is the normal configuration for a finished application. This causes the code (particularly lookup tables) to execute slower than if it is executed from the internal scratchpad RAM at D4000000. A boot loader application to load the program from flash to RAM at turn-on would improve execution speed, but this would limit the RAM available for

other tasks such as data logging etc. Also note that a power-systems application (plus variable/buffer space) may be too large to all fit inside internal RAM, even without data logging. The results from the initial TC1796 benchmarking are shown in Fig. G-1 and Fig. G-2.

Points of note are:-

- The sine/cosine evaluation from interpolated lookup tables can provide speed benefits over a floating-point calculations, which take about 1µs each. However, the lookup retrieval and interpolation code itself has a reasonable overhead. Interpolating from a memory-efficient 1-quadrant table (0 to 90 degrees only) requires almost as much time as the floating-point calculations. Interpolating from a 4-quadrant table (4 times as much memory required) takes about 0.5 µs for each, even for a pair of sin/cos answers for the same input angle. This would tend to suggest that memory access from the table may be a limiting factor here. A non-interpolated 4-quadrant table can be used to produce results in about 0.3 µs . Again, even when the data is retrieved in sin/cos pairs, for which most of the code is common, the time taken is almost 0.6us. This again suggests that addressing and memory access from the table is the limiting factor. In this case, because the program is loaded into the TC1796 internal flash memory, the lookup table will also be inside the flash memory. A 7x improvement might be realised by running the algorithm (or just storing the table) in internal RAM. In summary, on the TC1796 it is probably not worth using lookup tables when the application is resident in the flash memory, since the times for floating-point calculations are not much longer than the lookup times, and no numerical errors need to be accounted for
- All the rounding functions such as floor/ceil/round/fix provided by Simulink, when compiled on the TC1796, take surprisingly long to evaluate. Also, the floating-point to int32 conversions without saturation checking take longer than the conversions with saturation checking. This is not intuitive, but occurs because the Simulink without-saturation-checking algorithm uses the "fmod" floating-point modulo function, which takes more time to execute than the bounds checking which the saturation checking uses.
- The speed of the rounding functions can be significantly improved by instead using the native casting provided by a C code expression such as "I=(int)f", where I is an int32 and *f* is a floating-point variable. This can be achieved by using simple Simulink S functions. However, when using this approach, great care must be taken on three counts. The first is that the possibility of overflows/underflows

must be rigorously avoided or trapped. The second is that different target processors behave in different ways. For example, a PC running Simulink responds to the C code casting "i=(int)f" with a "fix" (towards zero) truncation, while the TC1796 processor responds with a "round" action. Within the code libraries used during this thesis, a flag must be set by the user which determines the extra small code segments which must be added to the C code casting to achieve the required rounding function "ceil", "floor", "round" or "fix". These code sections are different for the different target processor types. The final point is that any rounding function may make a mistake of 1, when the floating-point value is very close to a decision threshold. Subsequent code, such as array indexing and use in the "mod" function, should account for this possibility.

- The Simulink library function "mod" is very slow, since it uses the Simulink function "floor", and also because it involves detailed checks for numerical precision around the rounding decision boundaries. The mod function can easily be replaced manually in Simulink using the function mod(A,B)=A-B*floor(A/B), and by using a C cast to carry out the "floor" operation. This produces much faster C code. If required, a saturation/over-range check can be added to make sure that A/B is not too large, before carrying out the "floor" operation. A final tweak is an extra piece of code which checks whether the output of the mod function is actually within the bounds of 0<=mod<=B. This can occur due to numerical precision errors in the "floor" calculation. If the bounds are exceeded, then B can be added or subtracted from the answer to wrap it back inside the expected range.
- The int32 additions/subtractions are very fast, almost immeasurable, as would be expected. Note, however, that the Simulink increment/decrement library functions "V--", "Q--", "V++" & "Q++" did not translate into operations such as "i++" or "i--" in C code. Real-time-workshop instead turned these into "i+1" and "i-1" operations, looking identical to a "bias" of +1 or -1.
- The Simulink algorithms for gain and multiplications using int32 variables are very slow. It is assumed that this accounts for over-ranging etc. If over-ranging is avoided or trapped some other way, or made impossible, then simple single-line C code S-functions of the form k=i*j and k=i/j can be used to carry out multiplications and divisions of int32 variables. These are much faster than the Simulink implementations.
- Floating-point multiplications take no longer than floating-point additions/subtractions (≈0.1 μs), but divisions are 50% longer.
- The "two-taps" variable delay buffer S-function (see section 3.2.1) takes less time

to evaluate (0.25µs) than a floating-point sine or cosine. This is important as many of these blocks are used within the Fourier analysis, averaging, and filtering blocks proposed in this thesis. It should be noted that significant effort has been expended in reducing the execution time of the single-tap and two-tap delay buffers. The final versions take about 0.25µs (almost independent of buffer size), compared to the SimPowerSystems variant which takes 0.75µs (almost independent of buffer size), and the non-S-function method of Figure 3-3 takes massive times of 14.6µs for a 20-delay buffer and 114µs for a 160-delay buffer. This last method uses the "memcpy" instruction in C code which explains the lengthy execution times which are heavily dependent upon buffer length. Similarly to the results of the sin/cos lookup table analysis above, this result suggests that RAM memory access time is one of the major limiting factors of the execution speed on the TC1796, although in this case the memory used for buffers is the internal RAM, which should be 7x faster to access than the internal flash. In addition, the 0.25µs taken for the fastest buffer cannot be explained by simply adding up the execution times for the relatively simple (all int32) operations required for the bulk of the algorithm, so the deduction is that memory access speed is the constraint.

- The "sqrt" function does not take particularly long, at 0.6µs.
- The "atan2" function is relatively expensive, at 1.5µs
- The "abs" function provided by Simulink takes a massive amount of time (0.55µs) and can easily be replaced by Simulink code with a single "switch" and a "unary minus" block, to form an expression y=(a<0?-a:a). This takes only 0.04µs.
- The Simulink function "hypot" contains some obscure c code to avoid over-ranging. Generally, this isn't required if the inputs are reasonable values and a manually coded version is faster/better. hypot=sqrt(a*a+b*b).
- When the output of a Simulink code block is stubbed out with the "terminator" block, the Simulink real-time workshop process which generates C code is extremely efficient at removing all the previous code which is required to calculate that output, if the values are not needed for any other outputs (This feature can be disabled by un-checking the "Block Reduction" optimisation in the Simulink Real-Time-Workshop options). This is very useful to know for future development, since manual effort to cut-down complex blocks to simpler, faster blocks can often be bypassed. Instead, full blocks can be inserted in Simulink and the outputs simply stubbed out, safe in the knowledge that Simulink will cull all the un-needed C code. This can also be used as a "comment" mechanism inside Simulink. A block or section of code with all outputs stubbed with terminators will

generally not appear at all in the C code. Note: an exception is that the S-function delay blocks referred to extensively in this thesis can not be removed in this way. All code up to and including the delay blocks is included inside the c code, whether or not the output of the delay block is actually used. This is because the delay buffers constitute signal storage. Simulink knows there is signal storage within the buffers, and this violates one of the three conditions required for the "Block Reduction" optimisation to be applied. For this reason, some of the large analysis blocks used in this thesis must be manually stripped down if not all functions are required. A good example is that if the all-harmonic RMS and THD measurements are not needed, significant execution time can be saved by manually deleting the un-needed blocks of code in Simulink.

- Another useful observation is that a boolean switch can be inserted inside Simulink code, with the boolean switch value set to a constant. The two paths feeding the switch can be entirely different algorithms. When Simulink creates C code, it knows the value of the constant Boolean flag, and therefore does not create any C code for the un-needed code path. This is very useful, since different candidate algorithms can be switched between at the C code generation stage, without modifying any Simulink code libraries, simply by changing a MATLAB workspace value from a 1 to a 0 or vice versa. As an example, this approach makes the coding of the machine-dependent floating-point to integer casting algorithms easy to manage.

Due to the large number of delay buffer blocks used within some of the algorithms (see section 5.6), significant effort was expended in optimising the delay blocks. Starting from a baseline execution time of 1.5µs per block (the time taken for the SimPowerSystems "variable transport delay" block on the TC1796 without cache enabled), the time for the Author's blocks was reduced to 0.3µs, using the same target configuration. This is achieved by strict "in-lining" of the S-function code within the Simulink ".tlc" file, use of pointer arithmetic, removal of un-needed bounds checks, and careful C-code implementation. The final result is a less readable but faster algorithm with the identical functionality. The code for the two-tap delay output is archived in section G.1. The single-tap (and three-tap) versions are similar.

The execution time of the delay blocks drops to < 0.25µs on the TC1796 when CPU caching is enabled (see below).

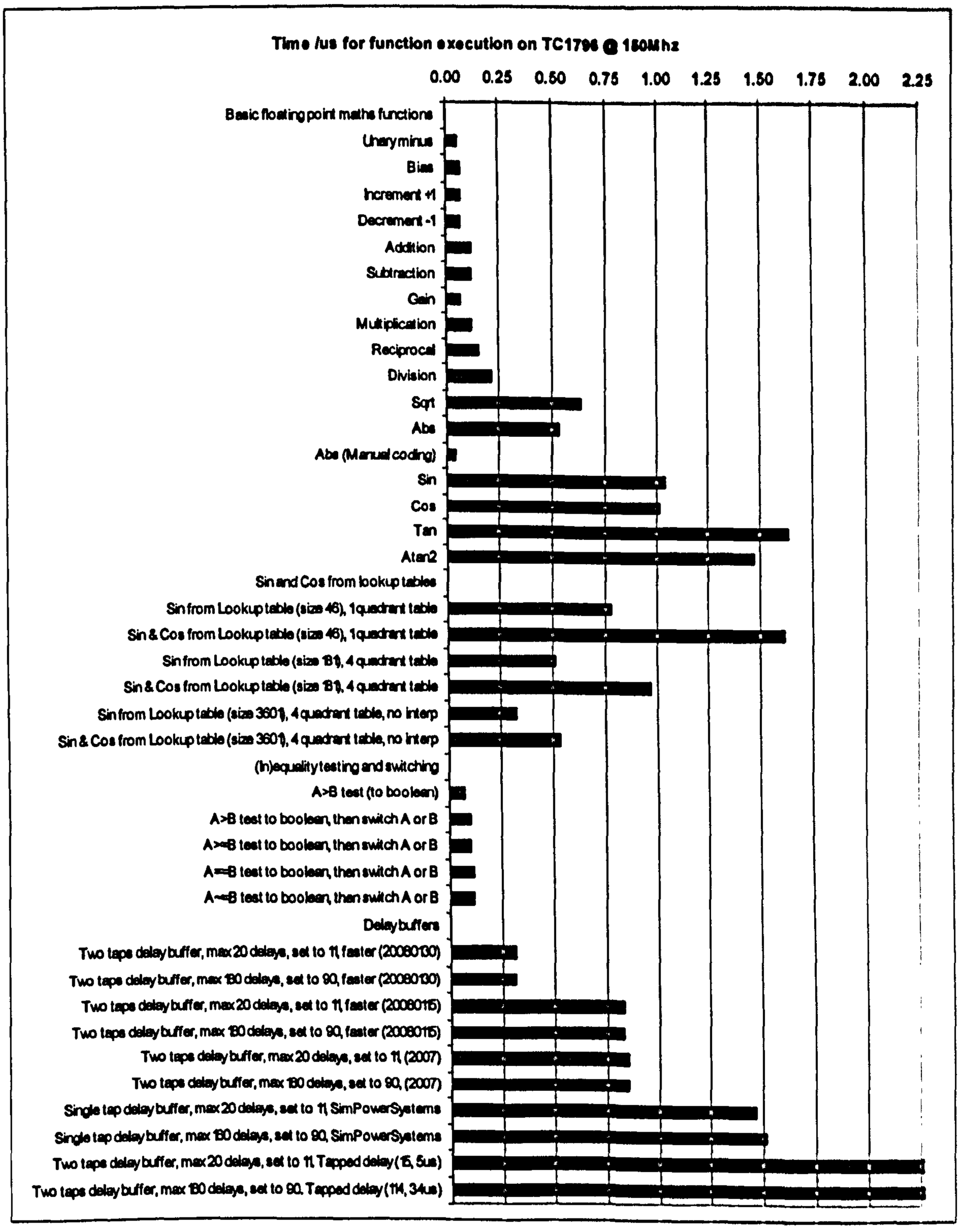


Fig. G-1 : Execution times (μs) for common function evaluations on the TC1796 microcontroller. CPU clock 150MHz

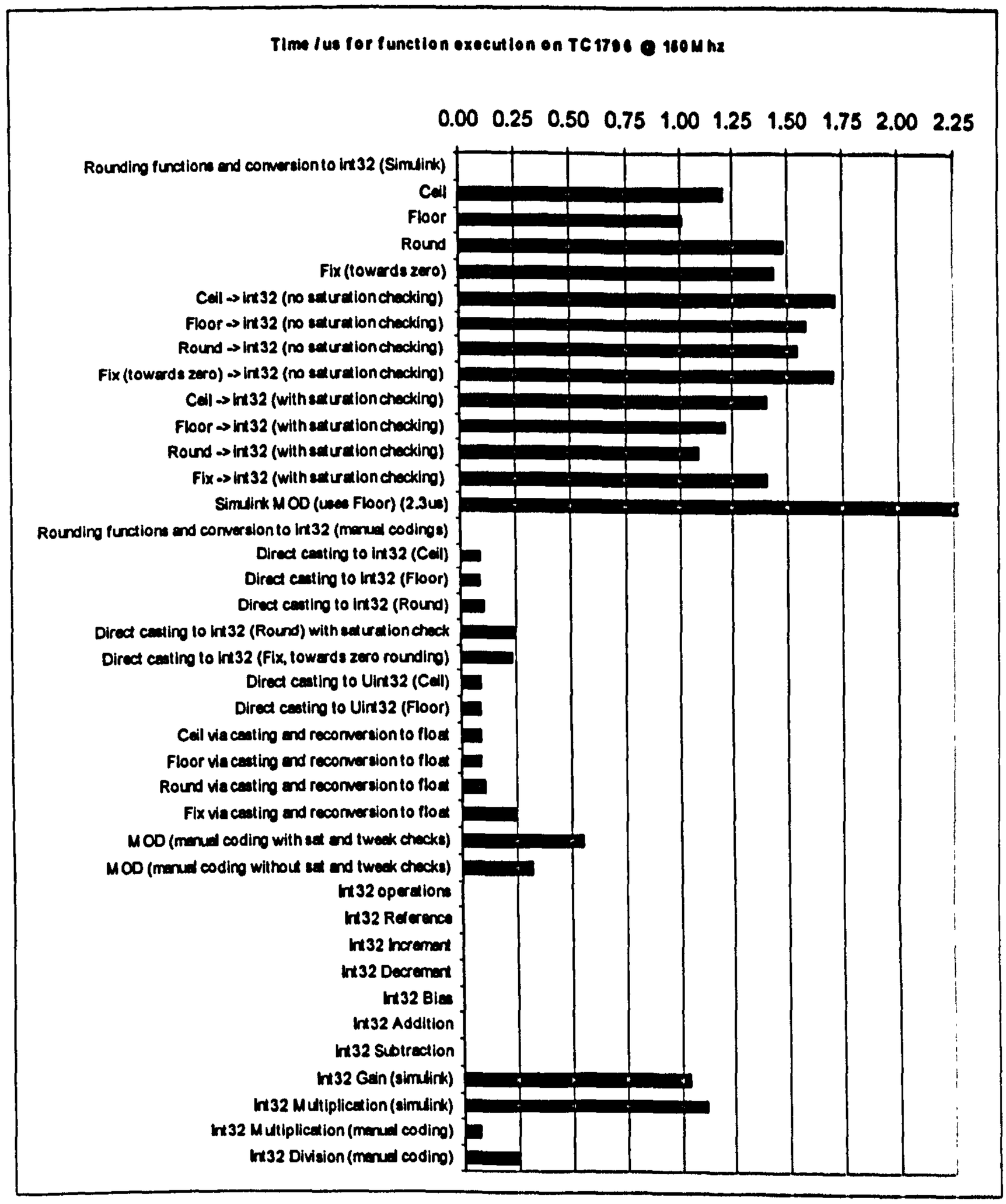


Fig. G-2 : Execution times (µs) for rounding and integer function evaluations on the TC1796 microcontroller. CPU clock 150MHz

A 30% speedup on the TC1796 can be achieved by enabling the CPU cache[1]. This allows

---

[1] To enable the CPU cache on the TC1796, link the application to 0x80000000 not 0xA0000000 in Tasking, and program the flash in HITOP via 0x80000000. Reset and run from 0x80000000. Inside the initialisation C code, insert the lines:-

```
MAIN_vResetENDINIT();
PMI_CON0=0;/* (Enable 16kB CPU caching) See systems units manual pages 2-24 to 2-27 */
MAIN_vSetENDINIT();
```

flash memory chunks in the 0xA0000000 flash memory area to be uploaded into the 16kB CPU cache as required, by using the memory mapped section at 0x80000000, to improve execution speed. The improvements affect most of the individual code segments by around 30% relative to the times shown in Fig. G-1 and Fig. G-2.

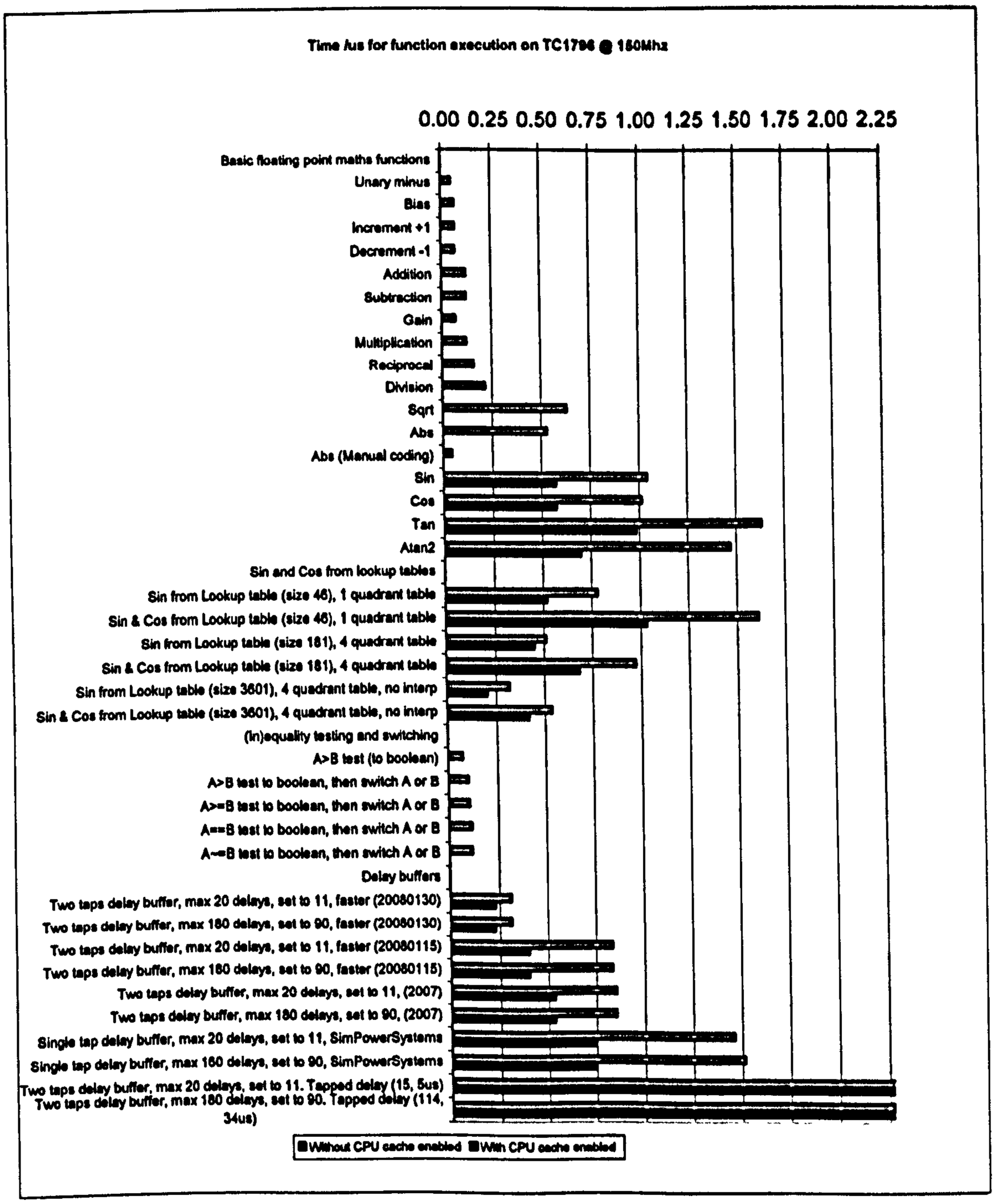


Fig. G-3 : Execution times (µs) for common function evaluations on the TC1796 microcontroller. CPU clock 150MHz. Some functions re-benchmarked with CPU cache enabled.

Fig. G-3 shows the data from Fig. G-2 repeated, again on the TC1796, but with a few of the functions re-benchmarked with the CPU cache enabled for comparison. The most important speed-up is that the delay block execution time can be dropped from 0.3µs to < 0.25µs.

In Fig. G-4, the times for some large algorithms are shown. On this chart, the same algorithms have been benchmarked on three different platforms:-

- Infineon TC1796 microcontroller, CPU clock 150MHz, program in 2MB internal flash memory at 0x80000000, 16kB CPU cache enabled, with the most optimised delay blocks (version 20080201)
- ADI RTS processor, ce5100 variety. Motorola PowerPC, CPU clock 500MHz, with the most optimised delay blocks (version 20080201)
- ADI RTS processor, ce5500 variety, with the most optimised delay blocks (version 20080201). *"CE5500 is ADI's next generation of compute power for the RTS real-time simulator. The heart of this compute engine is the 1GHz G4-based PowerPC processor. This CE includes 32KB on-chip L1 cache, 256KB on-chip L2 cache, 2MB L3 cache, and 512 MB of RAM... Performance improvements range from 3 to 5 times the computational power of the 500MHz CE5100"*

The fourth algorithm with times of 133/323/35 µs contains all the code required to evaluate voltage and power flow at a node of a three-phase power system. Frequency and the voltage magnitudes/phases are calculated as per section 5, with ripple removal filters applied to the 3 voltage magnitude results. The 3-phase current magnitudes and phases are also analysed by re-using the same measured frequency, as per section 3. In addition, a positive/negative sequence analysis is carried out and the unbalance calculated. P & Q flows, power factors and power angles are calculated for the three phases. The sequence analysis and power flow calculations are done without any extra sin/cos evaluations, by careful re-use of path averaging data (see section 3.9).

Finally, Fig. G-5 shows the breakdown of execution time on the TC1796 microcontroller, this time also showing the times required for the DC bias removal, ADC skew, and amplitude/phase corrections for LPF and FIR filter stages. The total times for 3-phase voltage/frequency measurement is 113µs. Extending this to a 6-phase voltage/current set with full sequence/balance analysis and power flow analysis increases the algorithm time to 156µs.

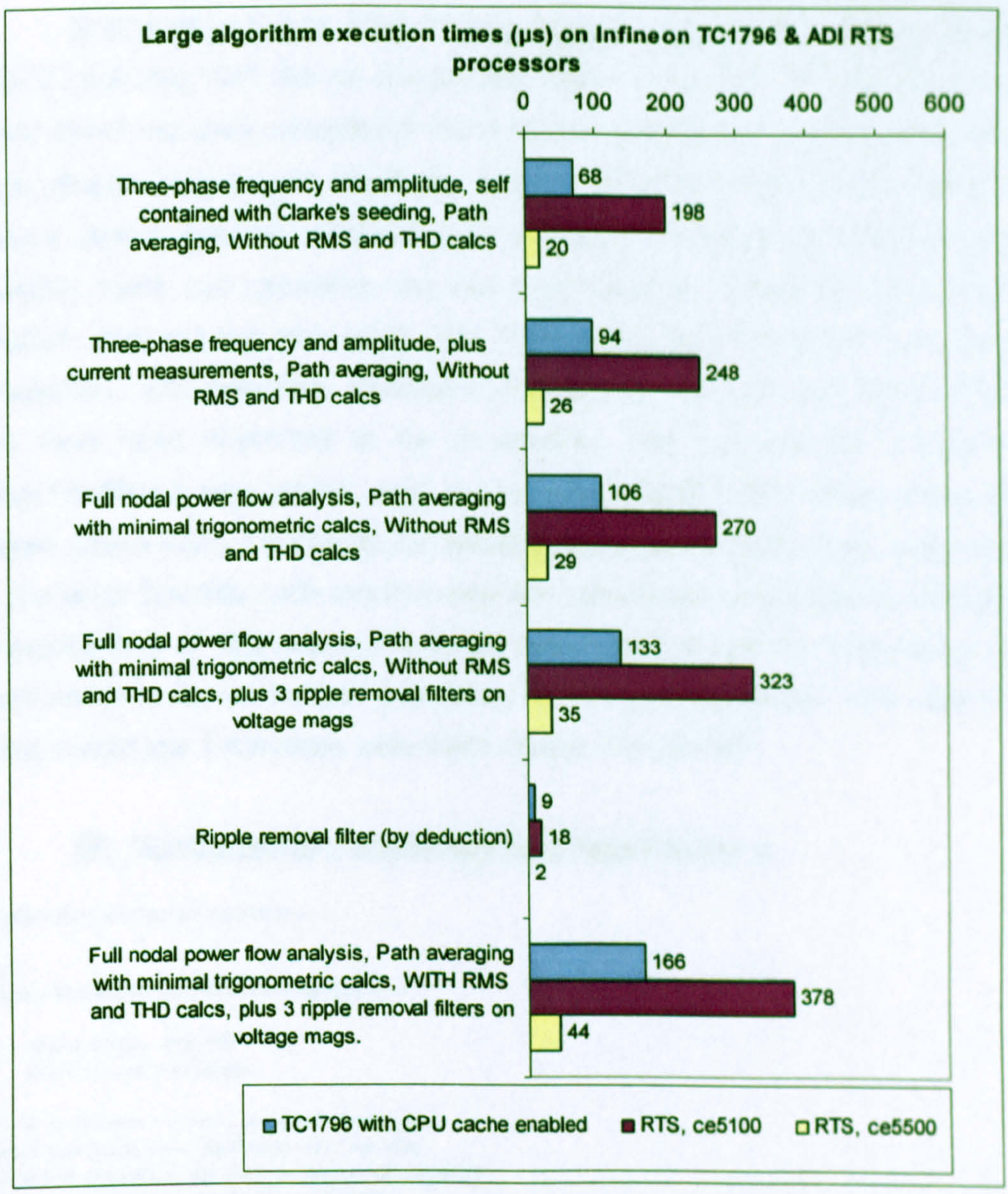


Fig. G-4 : Execution times (µs) for large 3-phase signal processing algorithms on the TC1796 microcontroller, and the ADI RTS processors (ce5100 & ce5500).

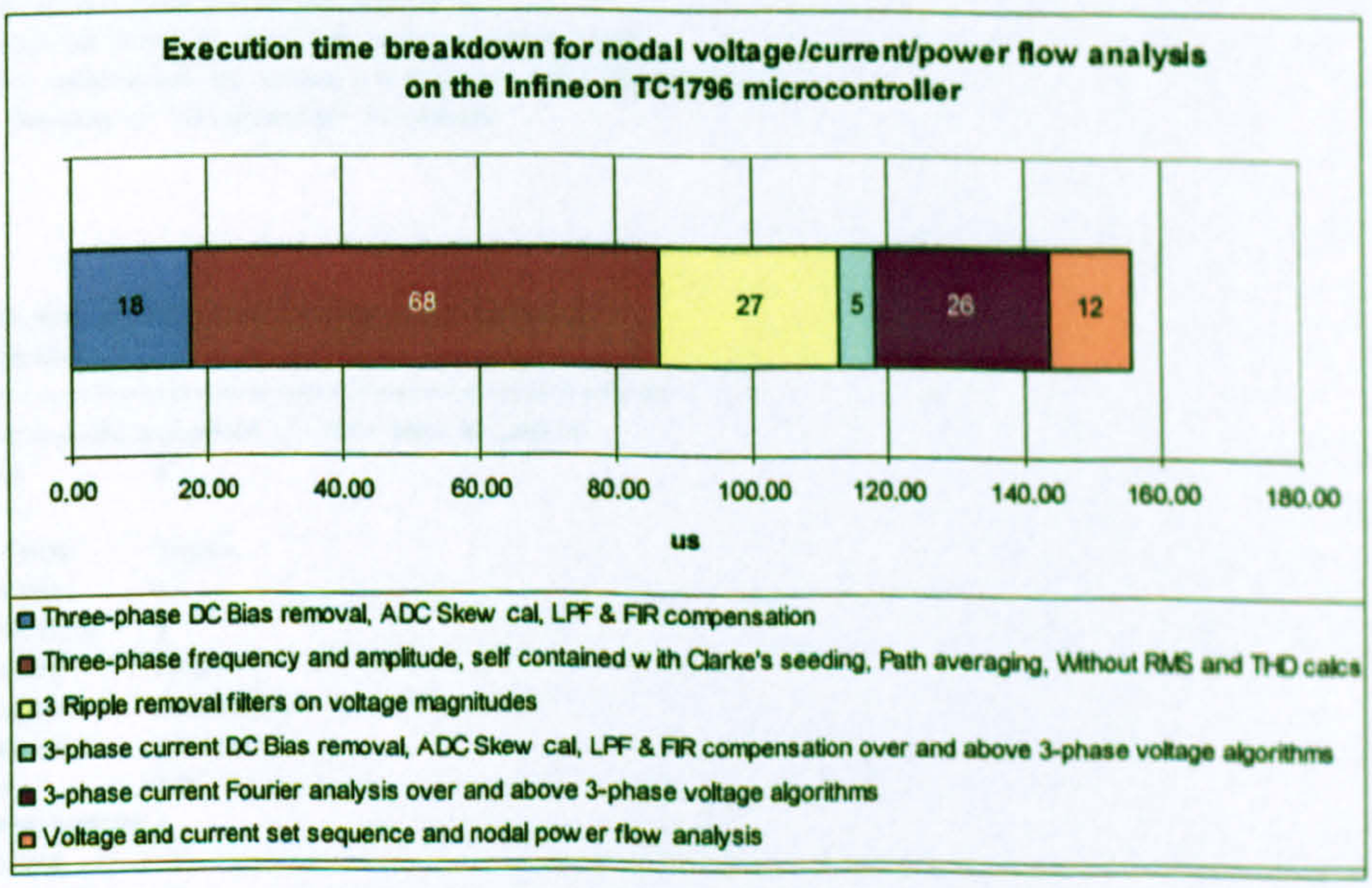


Fig. G-5 : Execution time breakdown for nodal voltage/current/power flow analysis algorithm on the TC1796 microcontroller

G.1 S-function for the "two-taps" variable-delay buffer

Here, the C-code and "tlc" file for the variable-delay "two-taps" delay buffer is archived. Significant effort has been expended in incremental optimisation and benchmarking of this algorithm, despite its apparent simplicity. Mainly, the effort involves optimising the "tlc" file which is used to generate actual C-code by Real-Time Workshop. The hard part is not the algorithm itself, but optimising the way that Simulink creates the interface between the Simulink code and the "tlc" code. The "tlc" file is fully "in-lined" (see the Simulink documentation), and resulting assembler instructions due to the Simulink-S-Function interface have been minimised as far as possible. The code for the "single-tap" and "three-tap" buffers is very similar, as is the code for a buffer with a fixed delay. Note that in the code shown here, the checks for sensible delay value inputs have been commented out, as the prior Simulink code ensures that the values sent are within acceptable limits. (In the applications of this thesis, the same delay value is sent to many delay blocks, so this check needs to be carried out only once, for many delay blocks). Less robust Simulink algorithms should use S-functions with these checks re-inserted!

G.1.1 SF_VariableDiscreteDelayTwoTapsFaster.c

```
/*
 * File: SF_VariableDiscreteDelayTwoTapsFaster.c
 *
 * Description:
 *   S-function "SF_VariableDiscreteDelayTwoTapsFaster.c".
 *
 *   Author : Andrew Roscoe, 2006-2008
 *            University of Strathclyde
 *
 * This version is the most in-lined, has direct feedthrough
 * (no "normal" S-function state implementation), and also
 * skips the bounds checking of the input parameter "DelaySamples".
 * "DelaySamples" is assumed to already be in the range
 * 1 <= DelaySamples <= MaxDelaySamples
 * If values outside this range are input, segmentation faults may occur.
 *
 * Slower versions of this code can be implemented :-
 * - with less in-lining (clearer, more easily maintainable code)
 * - without direct feedthrough (by adding a S-function state implementation)
 * - with bounds checking of "DelaySamples" re-enabled
 *
 */


#define S_FUNCTION_NAME SF_VariableDiscreteDelayTwoTapsFaster
#define S_FUNCTION_LEVEL 2
/*<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*/
/* %%%-SFUNWIZ_defines_Changes_BEGIN --- EDIT HERE TO _END */
#define NUM_INPUTS          2
/* Input Port  0 */
#define IN_PORT_0_NAME      Signal
#define INPUT_0_WIDTH       1
#define INPUT_DIMS_0_COL    1
#define INPUT_0_DTYPE       real_T
#define INPUT_0_COMPLEX     COMPLEX_NO
#define IN_0_FRAME_BASED    FRAME_NO
#define IN_0_DIMS           1-D
#define INPUT_0_FEEDTHROUGH 1
#define IN_0_ISSIGNED        0
#define IN_0_WORDLENGTH      8
#define IN_0_FIXPOINTSCALING 1
#define IN_0_FRACTIONLENGTH  9
#define IN_0_BIAS            0
```

```
#define IN_0_SLOPE            0.125
/* Input Port  1 */
#define IN_PORT_1_NAME        DelaySamples
#define INPUT_1_WIDTH         1
#define INPUT_DIMS_1_COL      1
#define INPUT_1_DTYPE         int32_T
#define INPUT_1_COMPLEX       COMPLEX_NO
#define IN_1_FRAME_BASED      FRAME_NO
#define IN_1_DIMS             1-D
#define INPUT_1_FEEDTHROUGH   1
#define IN_1_ISSIGNED          0
#define IN_1_WORDLENGTH        8
#define IN_1_FIXPOINTSCALING  1
#define IN_1_FRACTIONLENGTH    9
#define IN_1_BIAS              0
#define IN_1_SLOPE            0.125

#define NUM_OUTPUTS           2
/* Output Port  0 */
#define OUT_PORT_0_NAME       DelayedSignal
#define OUTPUT_0_WIDTH        1
#define OUTPUT_DIMS_0_COL     1
#define OUTPUT_0_DTYPE        real_T
#define OUTPUT_0_COMPLEX      COMPLEX_NO
#define OUT_0_FRAME_BASED     FRAME_NO
#define OUT_0_DIMS            1-D
#define OUT_0_ISSIGNED         1
#define OUT_0_WORDLENGTH       8
#define OUT_0_FIXPOINTSCALING 1
#define OUT_0_FRACTIONLENGTH   3
#define OUT_0_BIAS             0
#define OUT_0_SLOPE            0.125
/* Output Port  1 */
#define OUT_PORT_1_NAME       DelayedSignalTwo
#define OUTPUT_1_WIDTH        1
#define OUTPUT_DIMS_1_COL     1
#define OUTPUT_1_DTYPE        real_T
#define OUTPUT_1_COMPLEX      COMPLEX_NO
#define OUT_1_FRAME_BASED     FRAME_NO
#define OUT_1_DIMS            1-D
#define OUT_1_ISSIGNED         1
#define OUT_1_WORDLENGTH       8
#define OUT_1_FIXPOINTSCALING 1
#define OUT_1_FRACTIONLENGTH   3
#define OUT_1_BIAS             0
#define OUT_1_SLOPE            0.125

#define NPARAMS               1
/* Parameter  1 */
#define PARAMETER_0_NAME      MaxDelaySamples
#define PARAMETER_0_DTYPE     int32_T
#define PARAMETER_0_COMPLEX   COMPLEX_NO

#define SAMPLE_TIME_0         INHERITED_SAMPLE_TIME
#define NUM_DISC_STATES       0
#define DISC_STATES_IC        [0]
#define NUM_CONT_STATES       0
#define CONT_STATES_IC        [0]

#define SFUNWIZ_GENERATE_TLC 1
#define SOURCEFILES "__SFB__"
#define PANELINDEX            6
#define USE_SIMSTRUCT         0
#define SHOW_COMPILE_STEPS    0
#define CREATE_DEBUG_MEXFILE 0
#define SAVE_CODE_ONLY        0
#define SFUNWIZ_REVISION      3.0
/* %%%-SFUNWIZ_defines_Changes_END --- EDIT HERE TO _BEGIN */
/*<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*/
#include "simstruc.h"
#define PARAM_DEF0(S) ssGetSFcnParam(S, 0)

#define IS_PARAM_INT32(pVal) (mxIsNumeric(pVal) && !mxIsLogical(pVal) &&\
!mxIsEmpty(pVal) && !mxIsSparse(pVal) && !mxIsComplex(pVal) && mxIsInt32(pVal))

/*====================*
 * S-function methods *
```

```
 *====================*/
#define MDL_CHECK_PARAMETERS
 #if defined(MDL_CHECK_PARAMETERS) && defined(MATLAB_MEX_FILE)
   /* Function: mdlCheckParameters =============================================
    * Abstract:
    *    Validate our parameters to verify they are okay.
    */
   static void mdlCheckParameters(SimStruct *S)
   {
    #define PrmNumPos 46
    int paramIndex = 0;
    bool validParam = false;
    char paramVector[] ={'1'};
    static char parameterErrorMsg[] ="The data type and/or complexity of parameter    does not match the information *
    "specified in the S-function Builder dialog. For non-double parameters you will need to cast them using int8, int16,*
    "int32, uint8, uint16, uint32 or boolean.";

    /* All parameters must match the S-function Builder Dialog */
             {
              const mxArray *pVal0 = ssGetSFcnParam(S,0);
              if (!IS_PARAM_INT32(pVal0)) {
                validParam = true;
                paramIndex = 0;
                goto EXIT_POINT;
              }
             }
    EXIT_POINT:
     if (validParam) {
             parameterErrorMsg[PrmNumPos] = paramVector[paramIndex];
             ssSetErrorStatus(S,parameterErrorMsg);
     }
          return;
   }
 #endif /* MDL_CHECK_PARAMETERS */
/* Function: mdlInitializeSizes =============================================
 * Abstract:
 *   Setup sizes of the various vectors.
 */
static void mdlInitializeSizes(SimStruct *S)
{
   const int32_T   *MaxDelaySamples  = mxGetData(PARAM_DEF0(S));

   DECL_AND_INIT_DIMSINFO(inputDimsInfo);
   DECL_AND_INIT_DIMSINFO(outputDimsInfo);
   ssSetNumSFcnParams(S, NPARAMS);  /* Number of expected parameters */
     #if defined(MATLAB_MEX_FILE)
          if (ssGetNumSFcnParams(S) == ssGetSFcnParamsCount(S)) {
           mdlCheckParameters(S);
           if (ssGetErrorStatus(S) != NULL) {
             return;
           }
          } else {
           return; /* Parameter mismatch will be reported by Simulink */
          }
     #endif

   ssSetNumContStates(S, NUM_CONT_STATES);
   ssSetNumDiscStates(S, NUM_DISC_STATES);

   if (!ssSetNumInputPorts(S, NUM_INPUTS)) return;
   /*Input Port 0 */
   ssSetInputPortWidth(S,  0, INPUT_0_WIDTH); /* */
   ssSetInputPortDataType(S, 0, SS_DOUBLE);
   ssSetInputPortComplexSignal(S,  0, INPUT_0_COMPLEX);
   ssSetInputPortDirectFeedThrough(S, 0, INPUT_0_FEEDTHROUGH);
   ssSetInputPortRequiredContiguous(S, 0, 1); /*direct input signal access*/

   /*Input Port 1 */
   ssSetInputPortWidth(S,  1, INPUT_1_WIDTH); /* */
   ssSetInputPortDataType(S, 1, SS_INT32);
   ssSetInputPortComplexSignal(S,  1, INPUT_1_COMPLEX);
   ssSetInputPortDirectFeedThrough(S, 1, INPUT_1_FEEDTHROUGH);
   ssSetInputPortRequiredContiguous(S, 1, 1); /*direct input signal access*/
```

```
    if (!ssSetNumOutputPorts(S, NUM_OUTPUTS)) return;
    /* Output Port 0 */
    ssSetOutputPortWidth(S, 0, OUTPUT_0_WIDTH);
    ssSetOutputPortDataType(S, 0, SS_DOUBLE);
    ssSetOutputPortComplexSignal(S, 0, OUTPUT_0_COMPLEX);
    /* Output Port 1 */
    ssSetOutputPortWidth(S, 1, OUTPUT_1_WIDTH);
    ssSetOutputPortDataType(S, 1, SS_DOUBLE);
    ssSetOutputPortComplexSignal(S, 1, OUTPUT_1_COMPLEX);

    /**** Work Vector Modifications ***/
    ssSetNumSampleTimes(S, 1);
    ssSetNumRWork(S, *MaxDelaySamples+1);
    ssSetNumIWork(S, 1);
    ssSetNumPWork(S, 0);
    ssSetNumModes(S, 0);
    ssSetNumNonsampledZCs(S, 0);

    /* Take care when specifying exception free code - see sfuntmpl_doc.c */
    ssSetOptions(S, (SS_OPTION_EXCEPTION_FREE_CODE |
                     SS_OPTION_USE_TLC_WITH_ACCELERATOR |
                         SS_OPTION_WORKS_WITH_CODE_REUSE));
    }

    # define MDL_SET_INPUT_PORT_FRAME_DATA
    static void mdlSetInputPortFrameData(SimStruct  *S,
                                         int_T      port,
                                         Frame_T    frameData)
    {
        ssSetInputPortFrameData(S, port, frameData);
    }
    /* Function: mdlInitializeSampleTimes =========================================
     * Abstract:
     *    Specifiy  the sample time.
     */
    static void mdlInitializeSampleTimes(SimStruct *S)
    {
        ssSetSampleTime(S, 0, SAMPLE_TIME_0);
        ssSetOffsetTime(S, 0, 0.0);
    }

    #define MDL_START  /* Change to #undef to remove function */
    #if defined(MDL_START)
    /* Function: mdlStart =======================================================
     * Abstract:
     *    This function is called once at start of model execution. If you
     *    have states that should be initialized once, this is the place
     *    to do it.
     */
    static void mdlStart(SimStruct *S)
    {
        const int32_T  *MaxDelaySamples  = mxGetData(PARAM_DEF0(S));
        int32_T i;

        for (i = 0; i < (*MaxDelaySamples+1); i++)
            ssGetRWork(S)[i] = 0.0;
        ssGetIWork(S)[0] = 0;                                   /* This is the index of the buffer where we want the data to start going
    into */

        mexPrintf("Initialised a SF_VariableDiscreteDelayTwoTapsFaster buffer of %d samples.\n", *MaxDelaySamples+1);
    }
    #endif /*  MDL_START */

    #define MDL_SET_INPUT_PORT_DATA_TYPE
    static void mdlSetInputPortDataType(SimStruct *S, int port, DTypeId dType)
    {
        ssSetInputPortDataType( S, 0, dType);
    }
    #define MDL_SET_OUTPUT_PORT_DATA_TYPE
    static void mdlSetOutputPortDataType(SimStruct *S, int port, DTypeId dType)
    {
        ssSetOutputPortDataType(S, 0, dType);
    }

    #define MDL_SET_DEFAULT_PORT_DATA_TYPES
    static void mdlSetDefaultPortDataTypes(SimStruct *S)
    {
      ssSetInputPortDataType( S, 0, SS_DOUBLE);
```

```
 ssSetOutputPortDataType(S, 0, SS_DOUBLE);
}
/* Function: mdlOutputs =======================================================
 *
*/
static void mdlOutputs(SimStruct *S, int_T tid)
{
    const real_T   *Signal          = (const real_T*) ssGetInputPortSignal(S,0);
    const int32_T  *DelaySamples    = (const int32_T*) ssGetInputPortSignal(S,1);
    real_T         *DelayedSignal   = (real_T *)ssGetOutputPortRealSignal(S,0);
    real_T         *DelayedSignalTwo = (real_T *)ssGetOutputPortRealSignal(S,1);
    const int32_T  *MaxDelaySamples = mxGetData(PARAM_DEF0(S));

    real_T  *VDD_buffer = ssGetRWork(S);

/* VDD_in is a pointer to the index of the buffer where we want the new data to go */
/* *VDD_in is the index of the buffer where we want the new data to go */
    int_T   *VDD_in     = ssGetIWork(S);

    int_T   VDD_out;
    real_T  *Out_ptr;
/*    int_T   Offset  */

    VDD_buffer[*VDD_in] = *Signal;

/*    Offset = *DelaySamples;  */
/*    if (Offset > *MaxDelaySamples) Offset = *MaxDelaySamples;    Check omitted in this faster implementation */
/*    if (Offset < 1) Offset = 1;                                  Check omitted in this faster implementation */

    VDD_out = *VDD_in - *DelaySamples;
    if (VDD_out < 0) {
        VDD_out += *MaxDelaySamples;    /* VDD_out=VDD_out+*MaxDelaySamples+1 */
        VDD_out++;
    }

    Out_ptr=&VDD_buffer[VDD_out];
    *DelayedSignal = *Out_ptr;

    if (VDD_out == *MaxDelaySamples)
        *DelayedSignalTwo = VDD_buffer[0];
    else {
        Out_ptr++;
        *DelayedSignalTwo = *Out_ptr;
    }

    /* Leave VDD_in ready to take the next sample */
    (*VDD_in)++;
    if (*VDD_in > *MaxDelaySamples)
        *VDD_in = 0;
}


/* Function: mdlTerminate =====================================================
 * Abstract:
 *    In this function, you should perform any actions that are necessary
 *    at the termination of a simulation.  For example, if memory was
 *    allocated in mdlStart, this is the place to free it.
 */
static void mdlTerminate(SimStruct *S)
{
}

#ifdef  MATLAB_MEX_FILE    /* Is this file being compiled as a MEX-file? */
#include "simulink.c"      /* MEX-file interface mechanism */
#else
#include "cg_sfun.h"       /* Code generation registration function */
#endif
```

G.1.2 SF_VariableDiscreteDelayTwoTapsFaster.tlc

```
SF_VariableDiscreteDelayTwoTapsFaster.tlc
%% File : SF_VariableDiscreteDelayTwoTapsFaster.tlc
%%
%% Description:
%%   S-function "SF_VariableDiscreteDelayTwoTapsFaster.c".
```

```
%%
%%   Author : Andrew Roscoe, 2006-2008
%%            University of Strathclyde
%%
%% This version is the most in-lined, has direct feedthrough
%% (no "normal" S-function state implementation), and also
%% skips the bounds checking of the input parameter "DelaySamples".
%% "DelaySamples" is assumed to already be in the range
%% 1 <= DelaySamples <= MaxDelaySamples
%% If values outside this range are input, segmentation faults may occur.
%%
%% Slower versions of this code can be implemented :-
%% - with less in-lining (clearer, more easily maintainable code)
%% - without direct feedthrough (by adding a S-function state implementation)
%% - with bounds checking of "DelaySamples" re-enabled

%implements  SF_VariableDiscreteDelayTwoTapsFaster "C"


%% Function: Start ============================================
%%
%% Purpose:
%%      Initialise work vectors (global variables).
%%
%function Start(block, system) Output

  /* S-Function Initialise "SF_VariableDiscreteDelayTwoTapsFaster" Block: %<Name> */

  %assign nelements1 = LibBlockParameterSize(P1)
  %assign param_width1 = nelements1[0] * nelements1[1]
  %if (param_width1) > 1
   %assign pp1 = LibBlockMatrixParameterBase(P1)
  %else
   %assign pp1 = LibBlockParameter(P1, "", "", 0)
  %endif

  %assign vdd_buffer_RWork = LibBlockRWork(0, "", "", 0)
  %assign vdd_in_IWork = LibBlockIWork(0, "", "", 0)

  {
%%    int_T       MaxDelaySamples = %<pp1>;                 /* MaxDelaySamples parameter value */
    real_T      *VDD_buffer     = &%<vdd_buffer_RWork>;  /* VDD_buffer real work vector */
    int_T       *VDD_in         = &%<vdd_in_IWork>;      /* VDD_in integer work vector */
    int_T       i;

    for (i = 0; i < (%<pp1>+1); i++)
        VDD_buffer[i] = 0.0;
    *VDD_in = 0;       /* This is the index of the buffer where we want the data to start going into */
  }

%endfunction %% Start

%% Function: Outputs ==========================================
%%
%% Purpose:
%%      Code generation rules for mdlOutputs function.
%%
%function Outputs(block, system) Output
  /* S-Function Output "SF_VariableDiscreteDelayTwoTapsFaster" Block: %<Name> */

  %assign nelements1 = LibBlockParameterSize(P1)
  %assign param_width1 = nelements1[0] * nelements1[1]
  %if (param_width1) > 1
     %assign pp1 = LibBlockMatrixParameterBase(P1)
   %else
     %assign pp1 = LibBlockParameter(P1, "", "", 0)
   %endif

%assign pu0 = LibBlockInputSignal(0, "", "", 0)
%assign pu1 = LibBlockInputSignal(1, "", "", 0)
%assign py0 = LibBlockOutputSignal(0, "", "", 0)
%assign py1 = LibBlockOutputSignal(1, "", "", 0)

%assign vdd_buffer_RWork = LibBlockRWork(0, "", "", 0)
%assign vdd_in_IWork = LibBlockIWork(0, "", "", 0)

{
```

```
%%    int_T   MaxDelaySamples     = %<ppl>;              /* MaxDelaySamples parameter value */
%%    real_T  Signal              = %<pu0>;              /* Input signal */
%%    int_T   DelaySamples        = %<pul>;              /* Input signal - number of delay samples */
%%    real_T  DelayedSignal       = %<py0>;              /* Output signal */
%%    real_T  DelayedSignalTwo    = %<pyl>;              /* Output signal */
    real_T  *VDD_buffer         = &%<vdd_buffer_RWork>;  /* VDD_buffer real work vector */
    int_T   *VDD_in             = &%<vdd_in_IWork>;      /* VDD_in integer work vector */
    int_T   VDD_out;
    real_T  *Out_ptr;
/*    int_T   Offset;  */

/* DelayBufferCountTag    This comment allows a count of these blocks in the RTW c-code file */

    VDD_buffer[*VDD_in] = %<pu0>;

/*    Offset = %<pul>;  */
/*    if (Offset > %<ppl>) Offset = %<ppl>;    Check omitted in this faster implementation */
/*    if (Offset < 1) Offset = 1;              Check omitted in this faster implementation */

    VDD_out = *VDD_in - %<pul>;
    if (VDD_out < 0) {
        VDD_out += %<ppl>;     /* VDD_out=VDD_out+MaxDelaySamples+1 */
        VDD_out++;
    }

    Out_ptr=&VDD_buffer[VDD_out];
    %<py0> = *Out_ptr;

    if (VDD_out == %<ppl>)
        %<pyl> = VDD_buffer[0];
    else {
        Out_ptr++;
        %<pyl> = *Out_ptr;
    }

    /* Leave VDD_in ready to take the next sample */
    (*VDD_in)++;
    if (*VDD_in > %<ppl>)
        *VDD_in = 0;
 }

%endfunction %% Outputs

%% [EOF] SF_VariableDiscreteDelayTwoTapsFaster.tlc
```

Appendix H Logged domestic voltage and current waveforms

The figures below record a snapshot of single-phase domestic voltage and current waveforms on 21st March 2008 at the Author's home. The load is relatively light (~500W), and is made up mostly of "energy saver" light bulbs and audio-visual equipment. Of note is the extremely high harmonic content (52.9% THD) of the current drawn, which is mostly from the peak of the voltage waveform.

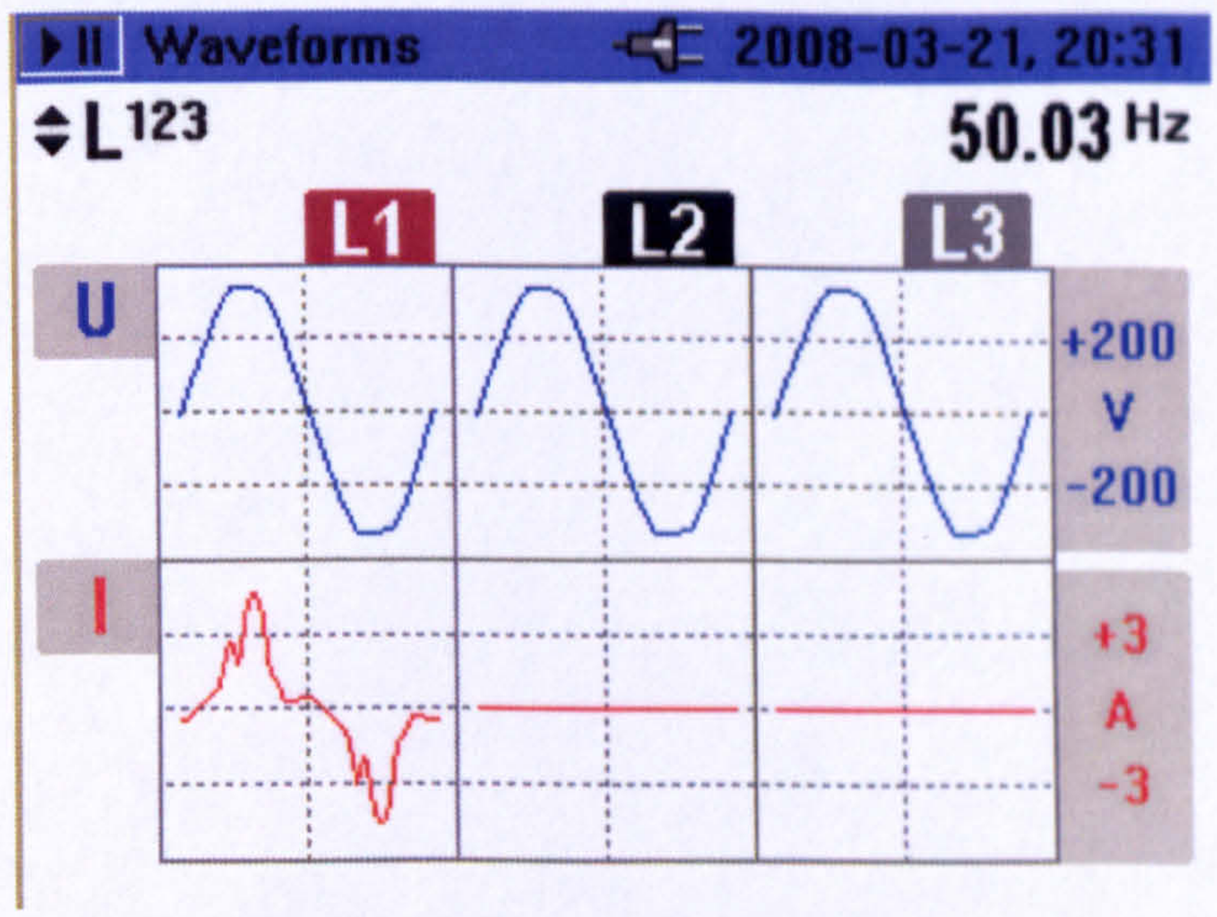


Fig. H-1 : Typical low-load domestic voltage and current waveforms

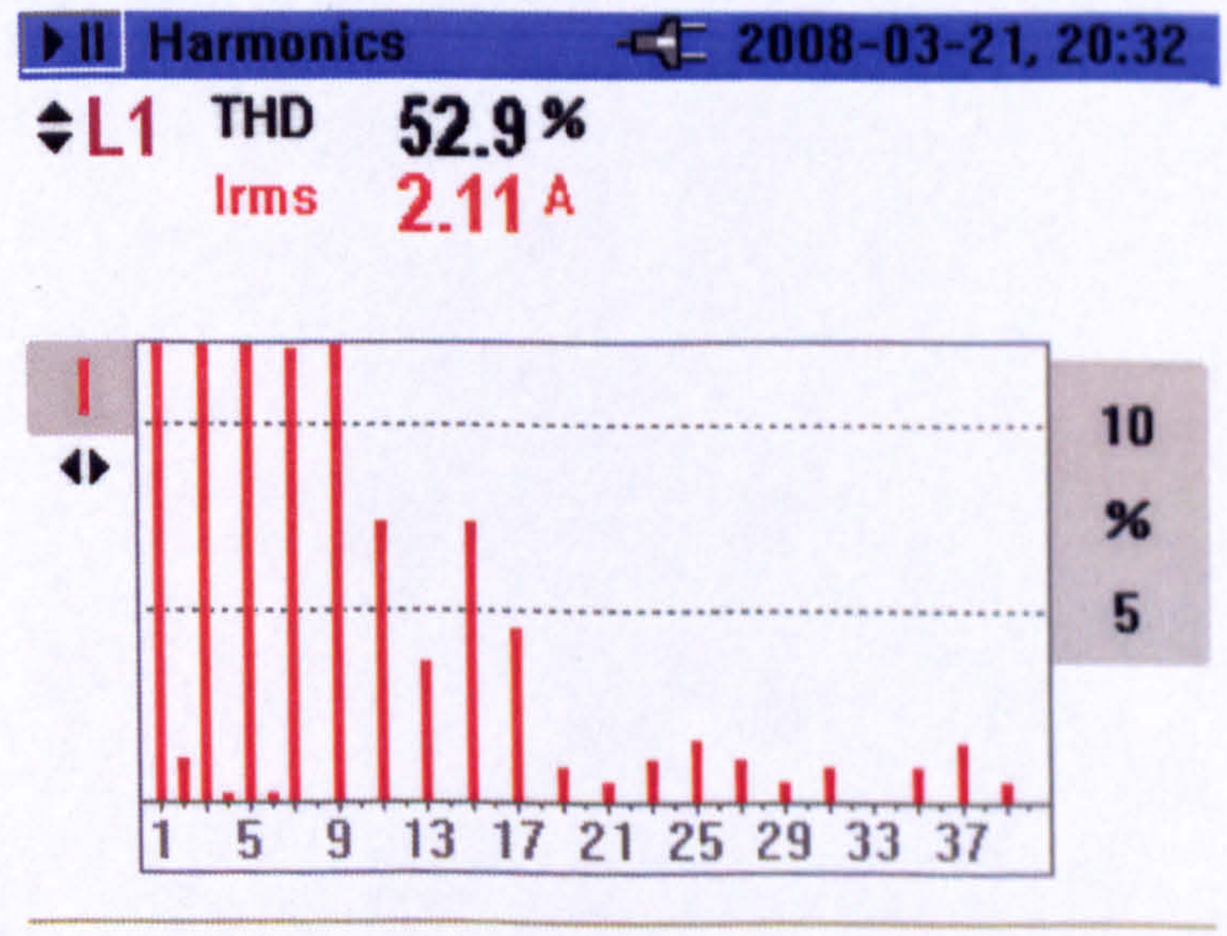


Fig. H-2 : Typical low-load domestic current harmonic distortion