5-2017

Low Bandwidth Communication for Networked Power Hardware-In-The-Loop Simulation

Sean Borgsteede
University of South Carolina

Low Bandwidth Communication for Networked Power Hardware-in-the-Loop Simulation

by

Sean Borgsteede

Bachelor of Science
University of South Carolina, 2015

Submitted in Partial Fulfillment of the Requirements

For the Degree of Master of Science in

Electrical Engineering

College of Engineering and Computing

University of South Carolina

2017

Accepted by:

Herbert L. Ginn, Director of Thesis

Andrea Benigni, Reader

Cheryl L. Addy, Vice Provost and Dean of the Graduate School

Acknowledgements

I would like to express my sincere gratitude to my Thesis Supervisor and mentor, Dr. Herbert Luther Ginn, and my Committee Member and mentor, Dr. Andrea Benigni, for their guidance, time, and suggestions. Thank you for all the support and know that I am grateful. I could not be where I am today without the guidance of my professors.

Special thanks to the Navy for the funding of our research projects here at the University of South Carolina on the All-Electric Ship (AES) design project. In addition, I thank the community of graduate students here at USC for their comradery and support through late nights and deadlines.

I would also like to thank my friends and family for their encouragement and support during my studies at the university. To my Mother and Father, I thank you for my upbringing and caretaking over the years.

Abstract

Power-Hardware-In-the-Loop (PHIL) simulations allow the design and validation of power hardware components in virtual power system schemas with near real-time operation. This technique is increasingly used in the development cycle of many products to reduce design costs and increase design fidelity. The Hardware under Test (HuT) interfaces with a simulation of the user's choosing through a hardware interface (HI). The digitally simulated system (DSS) runs on the real-time simulator before sending a reference value to the hardware interface to enforce. In this virtual to real interface, closed-loop stability and the simulation accuracy are the two paramount criteria in regards to the operational safety and experimental reliability.

The stability of the PHIL simulation represents the highest challenge in the implementation of this digitally simulated power system. The interface between the HuT and the DSS ideally adds zero distortion and maintains a unity gain. In practice, PHIL methods each have stability and accuracy trade-offs that will be discussed.

Any delay present in the simulation may negatively damp the system into unstable conditions. Careful consideration and compensation of interface delays can alleviate these adverse conditions. By communicating via time-frequency communication and partitioning a Point of Common Coupling (PCC) into the Hardware Interface controller, complex linear and nonlinear circuits may be simulated in a safe, stable and accurate environment.

Analysis and verification of this thesis has been tested using a PHIL testbed at the University of South Carolina. The PHIL testbed is composed of a TI TMS320F28335 digital signal processor, Opal-RT real-time simulator, and Matlab/Simulink.

Table of Contents

List of Figures

List of Abbreviations

DIM ..... Damping Impedance Method

DSS ..... Digitally Simulated System

FDC ..... Frequency Domain Communication

HI ..... Hardware Interface

HIL ..... Hardware-In-the-Loop

HuT ..... Hardware under Test

IA ..... Interface Algorithm

IDIM ..... Current Type Damping Impedance Method

ITM ..... Ideal Transformer Method

PCD ..... Partial Circuit Duplication

PHIL ..... Power-Hardware-In-the-Loop

RDFT ..... Recursive Discrete Fourier Transform

TDC ..... Time Domain Communication

Chapter 1 : Introduction

1.1 Background

Prototyping of complex power electronic systems is costly, time-intensive, and sometimes risky. Simulation has been widely used as a prototyping tool across many such disciplines. A modern modeling technique known as Hardware-in-the-Loop (HIL) incorporates the realistic limitations of hardware with the flexibility of simulation. This technique evolved from the use of EMTP solvers as shown in [1]. The hardware of importance, known as hardware-under-test(HuT), can be validated in-lab before progressing to expensive field tests. A computer model running in Real-Time (RT) is connected to the HuT to mimic or replace any surrounding hardware that would be present under normal operating conditions. The virtual system and hardware are coupled via low power control and feedback signals. The HuT is analyzed in a virtual system capable of flexible yet realistic tests in a controlled environment. Using a HIL method can reduce the time, money, and hazard associated with testing expensive and dangerous equipment. Detecting potentially catastrophic errors caused by uncommon or extreme test conditions early in the development phase lends HIL to be a very popular tool to bridge the gap between simulation and field tests. Rapid prototyping of digital controls using HIL have both boosted the reliability of designs and shortened the control design cycle [2].

Some of the first implementations of HIL were military testing of missile guidance, aircraft technology, and flight simulation[3]. Continuous improvements in computing, modeling, and communication have allowed real-time simulation of mechanical systems such as: braking systems, combustion engines, and drivetrains[4]. Historically, testing relays was the primary use for Real-Time Simulation in electrical engineering[5]. In contrast, HIL has r4ecently been used to

validate large electrical converters, grid-balancing strategies, and control systems [6]. HIL is often synonymous with Controller Hardware-in-the-Loop (CHIL) where only the controller of a system is connected to a virtual representation of the system. This thesis focuses on the simulation technique known as Power Hardware-in-the-Loop (PHIL). PHIL builds on the benefits of HIL by exchanging actual power between the HI and HuT. This allows the validation of switches, passives, and relays in addition to the control system. A robust variety of tests can be accomplished due to the added versatility.

1.2 PHIL Limitations

PHIL simulations must interface with the real world as seamlessly as possible. A variety of problems involving sampling rate, delay, quantization, and component saturation must be considered. Specifically, the stability of the system is highly dependent on the sampling rate and delay present in the interface between the real and virtual systems. As PHIL is often used to test devices under non-ideal or disturbed conditions, a high bandwidth will be necessary for fast-acting transient conditions. Due to advances in current technology, the computing delay and latency are less of a concern[7]. In this case, the bandwidth of the hardware interface between simulation and hardware bottlenecks the bandwidth of the PHIL simulation.

Commonly, the simulation and hardware interface are directly connected via analog I/O to maximize bandwidth and minimize delay. Digital interfaces may also be considered and in some cases more desirable than analog coupling, however, communication latency inserts additional delay into the interface. With a lesser communication link, the delay and latency of the simulation is heavily increased. The challenge of ensuring simulation stability and conservation of energy between two subsystems becomes significant when utilizing the Ethernet and local networks as a communication link. Overcoming the limitations of this delay has been successfully demonstrated by a cross-Atlantic PHIL simulation between South Carolina and Aachen[8]. The goal of this research is to create a PHIL simulation method that reduces effects of delayed

communication between the simulator and the hardware interface system such that it is feasible for a range of PHIL applications. The following techniques are tested to accomplish this goal: using a current-based interface algorithm (IIA) other than ITM, implementing a hybrid communication consisting of instantaneous and harmonic phasor values, and most importantly, partitioning of the simulation within the HI to effectively avoid the delay's effects.

1.3 PHIL Interface Algorithms

In PHIL simulations, two systems interact by exchanging real power across a simulation/hardware interface. This interface enables coupling between the virtual system and the power hardware. The main difference between HIL and PHIL can be seen in Fig. 1.1.

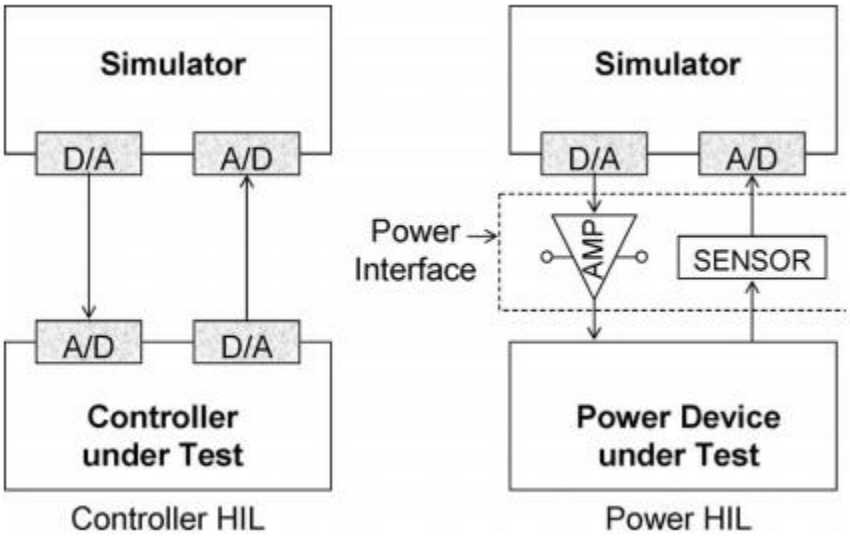

Figure 1.1 Comparison of Controller and Power HIL configurations

The HUT will exchange power with the HI, which will actively mimic the Simulated System by amplifying signals from the simulator. A plethora of literature has been published on HIL while PHIL has been more recent and many issues have not been investigated as thoroughly. The effects and influence of several interface algorithms on a PHIL simulation will be analyzed. Most of the comparison will rely on overall system stability margins. Additionally, the majority

of interface algorithms are developed for use in conjunction with voltage power amplifiers [7]. Only one current-amplifier based Interface Algorithm, Ideal Transformer Method (ITM) is widely recognized across PHIL literature. Some scenarios or applications may require or desire a current amplifier based Interface Algorithm.

An alternative choice for an IIA system is proposed and investigated in [9]. This will be accomplished by implementing this IIA to couple a RT simulation target with a three phase, 30KW rated, and IGBT-based current amplifier. A variety of IA methods exist that interface with the hardware via a voltage amplifier. Common voltage-based IA (VIA) will be briefly mentioned but ITM and an IB-DIM (Current based Damping Impedance Method) will be analytically compared to show the benefits to stability that IB-DIM can offer.

1.4 PHIL Partitioning & Communication

To counteract the delay, a partitioning of the model is proposed by shifting part of the simulated system into the HI as a set of state space equations. This partition would emulate common components at a Point of Common Coupling (PCC). These components could represent a common LC filter or a specified load designed to test the HuT. The load could be programmed to be a time-varying load with non-periodic, harmonic, and/or pulsed components. Any sudden shifts would be almost immediately followed according to the state-space estimations and then later be reflected within the Simulated System. Just as the human spine responds to immediate and important stimuli before the brain can react; the partition would react to disturbances immediately until the Simulated System can fully respond to the disturbance. Analysis of the stability benefits will be found using simulation and then the Hardware testbed.

1.5 Significance of Research

Currently, many universities, laboratories, and companies have been using HIL to accomplish tests and validate projects without access to a large systems or expensive products. As

mentioned earlier, HIL allows flexibility, safety, and cost-effectiveness. PHIL simulations between in-field hardware and in-house simulators are not possible under normal PHIL operating conditions. Companies such as NREL and INNL are cooperating to create new and interesting testbeds across hundreds of miles[10]. USC Columbia and RWTH Aachen have achieved simulator-to-simulator co-simulation across the Atlantic Ocean. On a smaller scale, labs around the campus of a University could interface with local utility's test bed and field equipment for hands-on learning and novel experiments. The need for easy-to-use, high-fidelity networked PHIL is present and growing.

Chapter 2 : PHIL Methods

2.1 Interface Algorithms

The ITM is composed of two ideal sources enforcing the conservation of energy. This is the most straightforward and conventional IA available. Depending on the signal being amplified, it is categorized as voltage type (Fig. 2.1) or current type (Fig. 2.2). Commonly, the voltage type is used for inductive HUTs while the current type is used for capacitive[11].

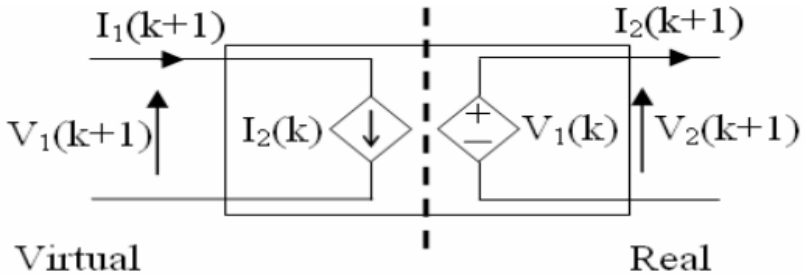

Figure 2.1: Voltage-Type ITM Method

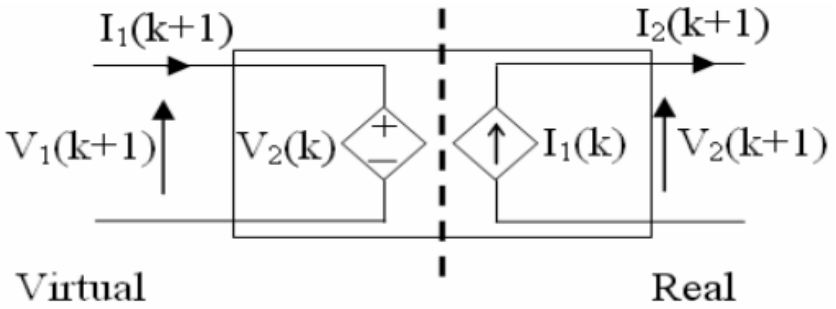

Figure 2.2: Current-Type ITM Method

The Transmission Line Model (TLM) technique model connects two circuits through a reactive component modeled as a transmission-line section. When the ΔV or ΔI per time step are small relative to the system, the two circuits can be considered decoupled as seen in (Fig 2.2). Another IA would be the Taganrog Interface which is named after the Taganrog State University of Radio Engineering. As shown below in Figure 2.3, current and voltage must be interchanged each time step. The resistors r_1 and r_2 are stabilization resistors meant to ensure a decoupled simulation. Unlike the linking resistor in TLM, r_1 and r_2 can be chosen regardless of the system parameters[11].

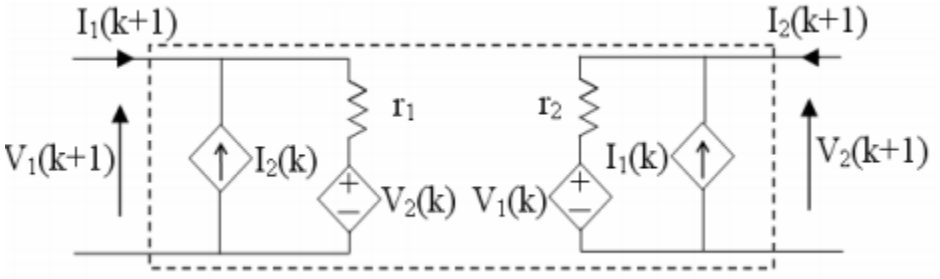

Figure 2.3: Taganrog Interface Algorithm

The Time-Variant First-Order Estimation(TFA) attempts to compensate the delay caused by the D/A, A/D, and computation by approximating the HUT with a time-variant first-order system. An example of the IA and the equation used to simulate a HUT based on a capacitive HUT can be seen below. As seen in the diagram below, v_2 is the voltage across the HUT and i_1 is the current supplying HUT.

$$\frac{dv_2}{dt} = a * i_1 + b * v_2 \quad (1)$$

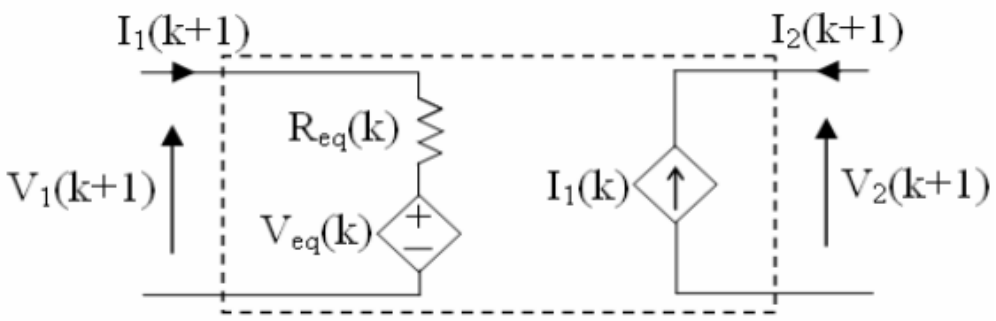

Figure 2.4: Time-Variant First-Order System

Partial Circuit Duplication (PCD) separates the original circuit into sub-circuits and solves them independently to decrease computation time. The name is due to the two partial circuits each containing a copy of the coupling resistance found in each other. The larger the coupling resistance, the faster the convergences to a solution using the Gauss-Seidel method iteratively as seen in [12]. PCD depends on a convergence factor that cannot be assured in one time-step and is subject to instability in this case. It is of interest to mention that PCD can be implemented as a Current-Type IA.

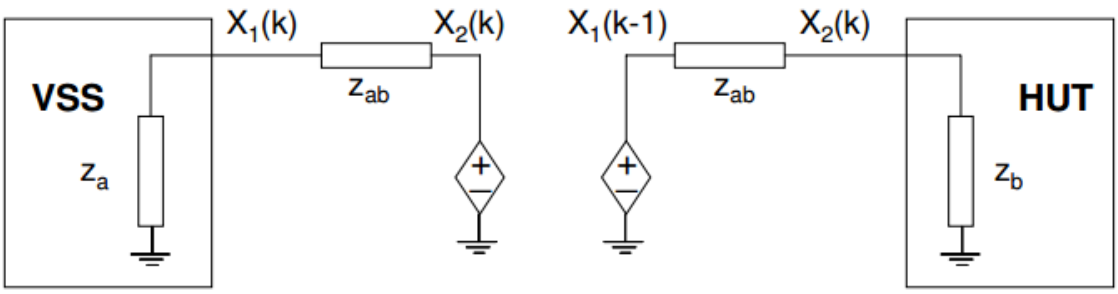

Figure 2.5: Partial Circuit Duplication

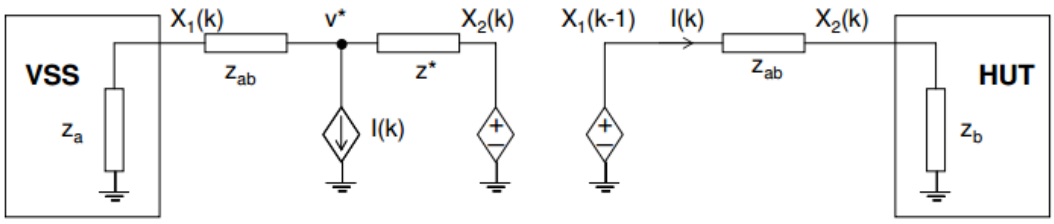

Figure 2.6: DIM Voltage-Type Interface

Damping Impedance Method (DIM) is a composite of the PCD and ITM method. The simplification to each of the two IA from DIM can be realized when z^* is equal to infinity and zero. At an infinite value, an open circuit is created to act as ITM. Setting z^* to zero creates a short circuit and the circuit mimics the PCD method. Comparing Figure 2.1, Figure 2.5, and Figure 2.6, the above assumptions can be seen. The loop transfer function of DIM is:

$$\boldsymbol{G_{LP}} = \frac{\boldsymbol{z_a}(\boldsymbol{z_b} - \boldsymbol{z^*})}{(\boldsymbol{z_b} + \boldsymbol{z_{ab}})(\boldsymbol{z_a} + \boldsymbol{z_{ab}} + \boldsymbol{z^*})} \quad \textbf{(2)}$$

When z^* is equal to Z_b, the magnitude of the function is zero. No errors will propagate between time-steps and the system is stable. In practice, an exact value for z^b would not be possible. Several techniques have been proposed for online calculation of z^* like the one found in [13].

There is a novel IA for current-type setups discussed in [9] that formulates a current-type DIM interface and discusses it merits and shortcomings. The derived IA can be see below in

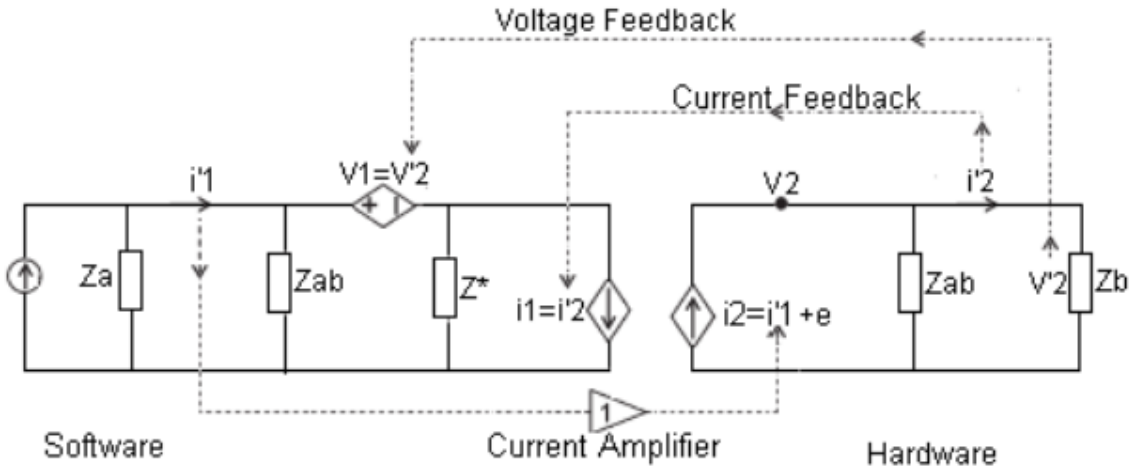

Figure 2.7: Current-Type Damping Impedance Method

Using this IA with $Zab \sim \infty$ and $Z^* = Zb$, creates an extremely accurate and stable interface that is ideal for sudden load steps and non-idealities. In practice, maintaining the relation $Z^* = Zb$ is impossible and would likely only restrict the difference between the two values to a very low magnitude.

2.2 Stability and Accuracy of PHIL Test

Due to the real-time nature of PHIL, the time step of the simulation must be maintained at a very small value to ensure fidelity and stability. The calculation and communication time needed for any particular setup may limit the bandwidth potential of a PHIL setup. Operating outside of these limitations will cause erroneous results or complete instability of the system.

Perturbations caused by the PHIL interface setup can be sensor noise, delay of signal, necessary stability-enhancing components, or the LPF effect due to the amplifier. Any noise perturbation (NP) refers to sensor noises and switching harmonics injected by the interface. The transfer function perturbation (TFP) refers to the transfer function of the interface and its non-

idealities. Any gain other than unity will introduce some simulation error and directly affect the response of the system[12].

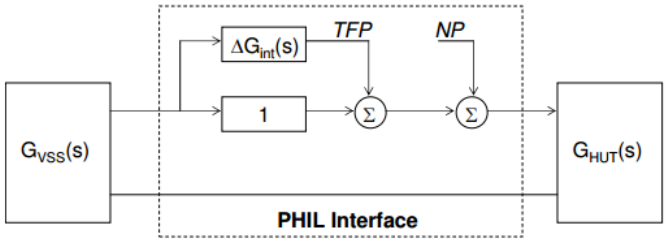

Figure 2.8: Schematic of TFP & NP errors

The ideal PHIL interface has a unity gain, zero time delay, and infinite bandwidth. Any variation from this idealized transfer function can be understood as $\Delta G_{int}(s)$. Based on Figure 2.9, the error can be represented by Equation 3 for any given frequency.

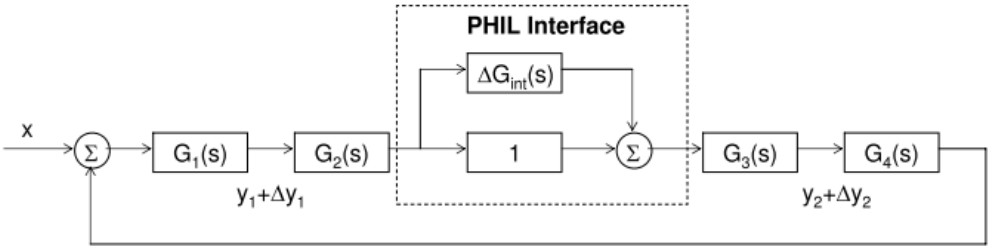

Figure 2.9: Close loop system with Transfer Function Perturbation

$$\boldsymbol{E_{TFP}} = \frac{|\boldsymbol{\Delta y}|}{|\boldsymbol{y}|} = \left|\frac{\boldsymbol{\Delta G(jw)}}{\boldsymbol{G(jw)}}\right| \tag{3}$$

$$\boldsymbol{E_{TFP\_y1}} = \left|\frac{\boldsymbol{G_{LP}\Delta G_{int}}}{\boldsymbol{1 - G_{LP}(1 + \Delta G_{int})}}\right| \quad \boldsymbol{E_{TFP\_y2}} = \left|\frac{\boldsymbol{\Delta G_{int}}}{\boldsymbol{1 - G_{LP}(1 + \Delta G_{int})}}\right| \tag{4}$$

Each of these equations refer to the open loop transfer function ($G_{LP} = G_1G_2G_3G_4$) and ΔG_{int}. The individual gains in the system are not independently correlated to the error function. To calculate total TFP error, both terms would have to be considered.

Time delays, sensor noise, and interface disturbances are classified as noise perturbation (NP) and usually consist of frequency components not present in the actual system. As seen in Figure 2.8, NP is separate from the TFP and must be normalized to the magnitude of the system response. The corresponding equation can be found below where W_I is the normalizing function.

$$\boldsymbol{E_{NP} = \left|\frac{e'}{e} W_I\right| = \left|\frac{W_I}{1 - G_{LP}(1 + \Delta G_{int})}\right|} \quad \textbf{(5)}$$

The author of [12] reduces the upper bound of the simulation error to a fairly simple set of equations using the following assumptions:

$$\boldsymbol{1 + \Delta G_{int} \approx 1} \quad \textbf{(6)}$$

$$\boldsymbol{|G_{LP}(1 + \Delta G_{int})| \approx |G_{LP}| < 1} \quad \textbf{(7)}$$

$$\boldsymbol{E_{TFP} = \max\left(E_{TFP_{y1,}} E_{TFP_{y2}}\right) = \frac{1}{1 - |G_{LP}| * |1 + \Delta G_{int}|}} \quad \textbf{(8)}$$

$$\boldsymbol{E_{NP} = \left|\frac{W_I}{1 - G_{LP}(1 + \Delta G_{int})}\right| \leq \frac{W_I}{1 - |G_{LP}| * |1 + \Delta G_{int}|}} \quad \textbf{(9)}$$

First, the interface transfer function is assumed to be small to assure simulation accuracy and secondly, the magnitude of the loop transfer function must be less than 1 for stable systems regardless of the delay. After calculating the error function of the system, interface, and weighing coefficient, the simulation can be conservatively ensured to not exceed the error function for any given frequency. An example of a comparison is shown below

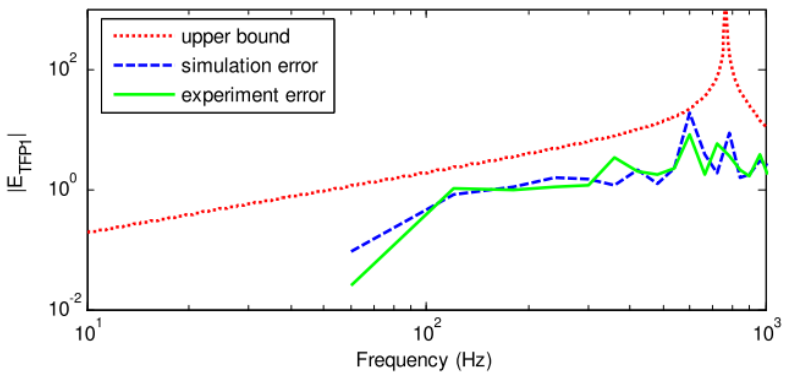

Figure 2.10: TFP error function vs. Frequency

Some IAs are better suited to delays and step changes than others. For example, the Taganrog method suffers heavily during any sudden changes in voltage or current measurement while the TFA method performs admirably. The TFA IA seeks to predict or compensate the error caused by delays. For well-established models with steady-state testing, TFA has many benefits over less complex models such as the ITM. In the case of non-linear HUTs, TFA and PCD are unable to create accurate simulation results. There are large tradeoffs between accuracy and stability in the choosing of an IA and many variables must be considered before choosing one. IA methods will be analyzed according to the scope of this work.

ITM	$G_{LP\_VITM} = -\frac{Z_S}{Z_L}$ and $G_{LP\_IITM} = -\frac{Z_L}{Z_S}$
TFA	$G_{LP\_TFA} = -\frac{Z_S}{Z_L} = -\frac{Z_S}{R+sL}(1-\frac{sT}{2})$
TLM	$G_{LP\_TLM} = \frac{Z_S - Z_{lk}}{Z_S + Z_{lk}} \frac{Z_L - Z_{lk}}{Z_L + Z_{lk}} e^{-2sT}$
PCD	$G_{LP\_PCD} = \frac{z_a z_b}{(z_a + z_{ab})(z_b + z_{ab})}$
DIM	$G_{LP\_DIM1} = \frac{z_a (z_b - z^*)}{(z_b + z_{ab})(z_a + z_{ab} + z^*)}$

Figure 2.11 Interface Algorithm Table for Stability

The above table[12] displays the transfer function of major IAs. From the transfer function and component values, the stability of a system can be quickly determined. An important point of the ITM would be the differing stability criteria depending on the source orientation. This will be discussed and explained in Chapter 3.4.

2.3 Communication Techniques

Historically, the simulator and hardware interface are connected via analog interfaces that exhibit almost no latency or bandwidth restrictions. PHIL interface communication between two sources over Ethernet or other delayed communication networks in the time domain is uncommon due to the simulation's sensitivity to time-shifts and delays. As referenced in[8], the problem of power conservation and the fidelity of the real-time simulation is very common yet poses an additional challenge when communicating over a lesser medium.

By compensating time-varying and delayed behavior of internet communication, geographically separated system could participate in a real-time simulation without the expected latency, stability, and accuracy issues. Reference [5] Proposes an approach based on time-frequency representation to overcome the communication & stability challenges. Using a discrete Fourier transform, the time-domain signal can be decomposed into the fundamental and its low-level harmonics for communication. This time-varying Fourier communication will be referred to as Frequency Domain Communication (FDC). FDC packages are sent, inversely transformed, and summated to recreate the original signal. Using this technique, alternate simulation time-steps are possible as well. Results from reference [8] show the effects of a delay in the loop. As seen in Figure 2.12, the integrated system and the recreated signal overlap completely.

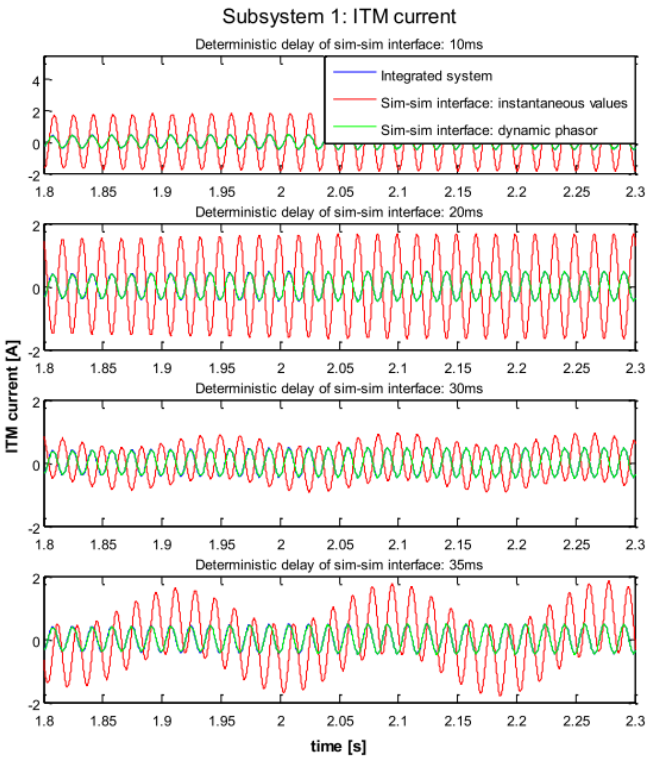

Figure 2.12: Comparison of Dynamic Phasor & Instantaneous Values under varying delays

Conversely, instantaneous value communication suffers dramatically from additional delays. Using a relative 2-norm error and inducing a voltage sag at 6s in Figure 2.13, it is seen that harmonic components aid in a more accurate simulation.

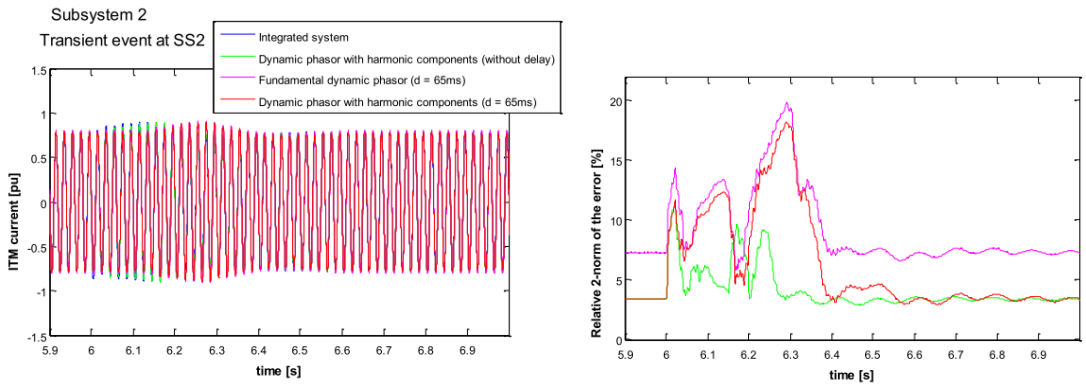

Figure 2.13: Error Reduction due to Harmonic Component Addition

Reference [8] speaks of adding up to the 11th harmonic component to increase the accuracy of the simulation. Ideally all harmonics within the bandwidth of a hardware interface would be included. This setup would be ill-suited to non-periodic and inter-harmonic disturbances as they would be effectively filtered out.

Using standard instantaneous value communication, an LPF can be used to suppress high frequency simulation artifacts from propagating throughout the loop. The LPF is not necessary but increases stability margin in all cases[13]. FDC would intuitively act as a LPF as only the chosen frequency components are sent. Filter design could be implemented using LPF by giving a weight to each of the harmonic components in a discrete LPF.

A novel approach to inter-cycle fluctuations is discussed in [14] to synchronize grid-connected power electronics in systems with high voltage or frequency variability. Most control signal references are time-domain-based to avoid the computation cost required for frequency-based reference signal generation. Overall, a variety of communication techniques exist to improve either the stability or accuracy of a PHIL simulation under a range of operating conditions. Later, these will be explored in conjunction with the objectives of this Thesis.

2.4 Partitioning & Co-simulation

Instead of multiple cores processing separate tasks simultaneously, multiple devices would be working together to accomplish tasks in tandem. Part of the simulation will be strategically off-loaded to the DSP. Ideally this part would be applicable to any high frequency filters present in the system to minimize the magnitude of high frequency communication between the Simulated System and the Partition.

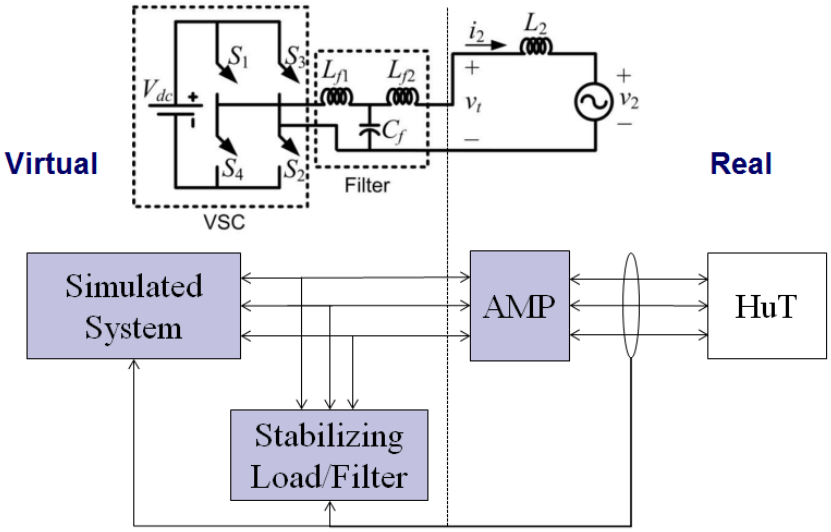

Figure 2.14 Breakdown of Simulation system groups

When breaking a simulation into two components, the point of interconnection must be carefully chosen. To maximize the stability margin of the system, the system will be separated at several locations to analyze the optimal position.

Existing ODE solving techniques will be references and used to guess, test, and choose the best possible solution. Existing large circuit models may use relaxation-based solvers to separate and validate correct simulation of the circuit while restricting the memory and computation requirements. Circuit modeling depends on convergence of a solution as quick as possible. To increase convergence, the large model must be separated correctly to ensure that tightly coupling components remain in the same lumped sub-circuit. If too many components are lumped together, the benefits of co-simulation will be lost. The proposed relaxation-strategy deployed in [15] discusses the coupling metrics loosely used to choose the test cases found below.

Because the two simulations are asynchronously connected by FDC, differing simulation time steps are possible. This allows independently chosen time-steps for each system.

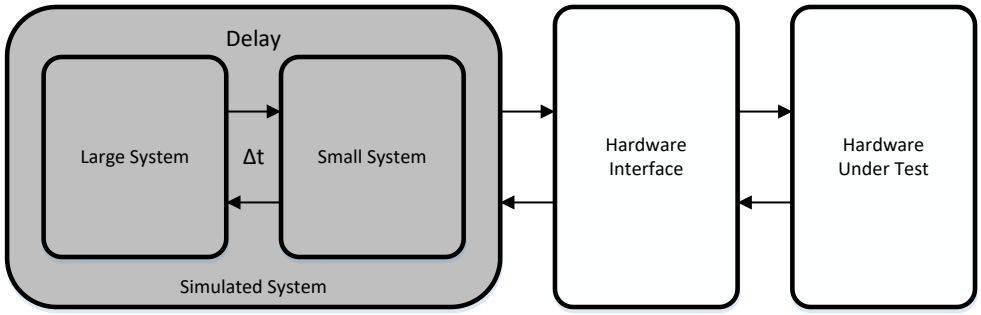

Figure 2.15 Partitioned Functional PHIL setup

In [13], a low pass filter is tested to increase stability margins. Modeling a first order delay using Pade's approximation, the following bode plots reflect a system with and without the low pass filter. The effectiveness of the LPF will be discussed further in Chapter 3.4. Below, the LPF stabilizes an otherwise unstable system and creates

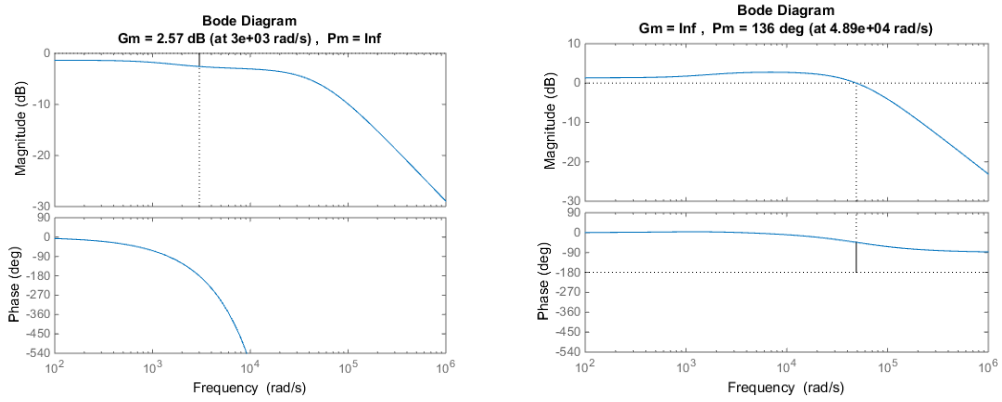

Figure 2.16 1ms delayed loop gain with LPF(right) and Loop gain without LPF(left)

Each of the above techniques can directly improve system stability and accuracy if correctly implemented. In Chapter 3, these techniques will be applied and analyzed to determine the best method available

Chapter 3 : Methodology & Simulation

3.1 Problem Clarification and Strategies

A PHIL setup between geographically separated hardware and simulators is currently very limited. To overcome this obstacle, several techniques will be tested at The University of South Carolina's Power and Energy Laboratory. The PHIL hardware interface is an American Superconductor's PM-1000 PEBB-based bi-directional VSC module. Using a TMS320F2833 floating-point digital signal processor (DSP), Opal-RT, and a Matlab/Simulink, the PHIL Simulation can be connected to a variety of HUTs for IEEE 1547 or UL 1741 Testing. Testbed applications include: power-factor, harmonic, and non-periodic compensation. Other control techniques such as voltage ride-through and inertia emulation may be tested as well. The communication link between the DSP and Opal-RT is an Ethernet link that limits the accuracy and stability of the simulation. Ensuring accuracy and stability during communication delays, load steps, and faults are paramount to this work.

The initial plan was to use FDC to increase the accuracy of the delayed PHIL system. Due to its zero steady error in simulation, non-transient conditions could be accurately simulated. After reading articles and journal papers aiming to increase the stability and accuracy of PHIL simulations, other promising methods were found for mitigating adverse effects of a slow communication link. Alternate interface algorithms such as the previously discussed current-based DIM and the TFA were analyzed. Partitioning of the model yielded positive results when applied to models with switching harmonics. These early results were gathered from MATLAB Simulink in addition to the FDC strategies.

Together the following three techniques were evaluated for their effectiveness at completing the objective. The DIM method increased the stability margin in average PHIL setup but due to the signal feedback needed, the accuracy suffers from the delay. The impedance of the HUT must be monitored to achieve the full potential of the DIM method. The TFA method approximates the operation of the system with a first order system and attempts to predict response of the system to counteract small delays. Approximating the system creates numerical instability when solving the system and requires an additional compensation current component. Due to the delay and system approximation, the TFA method's stability suffers greatly.

FDC alleviates many problems in simulation but the implementation of the technique between the DSP and OPAL proved extremely difficult. Due to the real-time TD systems on both ends of communication, ensuring synchronization and simulation accuracy was not feasible in the time . Partitioning the model to mitigate error and instabilities yielded positive results and increased the simulation's stability margin. Due to ease of implementation, benefits, and versatility, Partitioning was chosen as the main technique to pursue.

3.2 Interface Algorithms

Commonly used IAs discussed earlier all have their pros and cons that depend on the requirements of the simulation and the HuT characteristics. For the specific objective of mitigating the effects of communication delay, the most popular interface algorithms were tested and analyzed. Due to the restriction of I-type interface algorithms, ITM and DIM were chosen. First, the two were compared using MATLAB scripts to predict stability under predetermined conditions. Lastly, Simulink models simulated the physical components with a delayed communication structure in between.

To explore the stability of PHIL implementation, the two interface algorithms will be compared under similar load conditions. The open loop transfer function and block diagram of the ITM method can be seen below:

$$\boldsymbol{T_{OL} = T_{DAC}\, T_{ADC} T_{VSC} T_{LPF} \frac{Z_{HuT}}{Z_{Virtual}}} \quad \textbf{(10)}$$

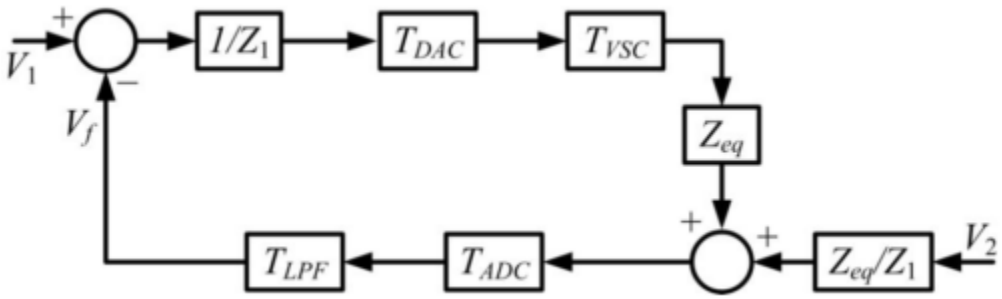

Figure 3.1 ITM Loop Gain Flowchart

Assuming all errors associated with the interface are negligible except the time delay, the time delays can be grouped as a time delay of t_d. The transfer function can be rewritten as

$$T_{OL} = -\frac{Z_{HuT}}{Z_{Virtual}} e^{-st_d} \quad \textbf{(11)}$$

The open loop result with similar assumptions yields Equation 7 for the I-DIM interface. The importance of this equation lies in the numerator. When $Z^* \approx Z_{HuT}$, Z^* damps the responses of the system while maintaining a near-unity gain for critical frequencies.

$$T_{OL} = \frac{Z^* - Z_{HUT}}{Z_{Virtual} + Z_{ab} + Z^*} e^{-st_d} \quad \textbf{(12)}$$

With a Z_{HuT} of $3 + 0.0005s$ and $Z_{virtual}$ equal to $7 + 0.0005$s, the following Nyquist plots were calculated:

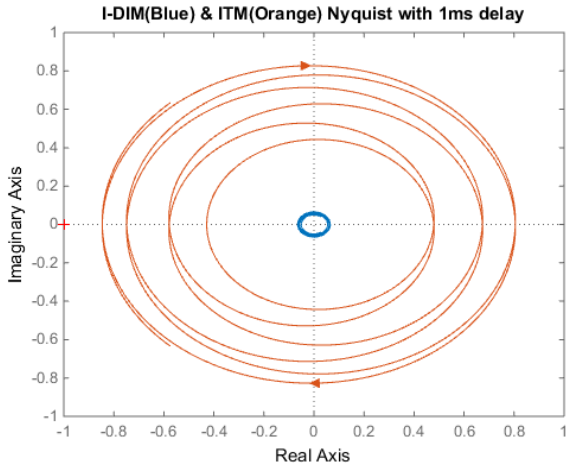

Figure 3.2 Nyquist Plot of IA stability when $\boldsymbol{Z_{virtual} > Z_{HuT}}$

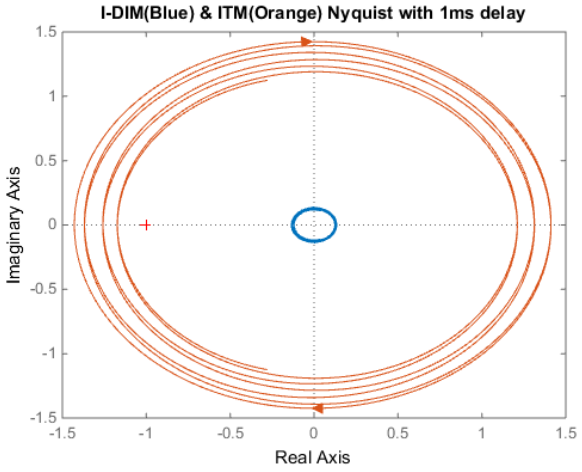

Figure 3.3 Nyquist Plot of IA stability when $\boldsymbol{Z_{virtual} < Z_{HuT}}$

When Z_{HuT} is greater than $Z_{virtual}$, the Nyquist plot of the I-ITM will always encircle (-1,0). Even with a mismatched impedance between Z^* and Z_{HuT}, I-DIM still ensures stability under a variety of load conditions. Altering the inductance of Z_{HuT} by increasing it to magnitude higher value gravely affects I-ITM while I-DIM is not adversely affected.

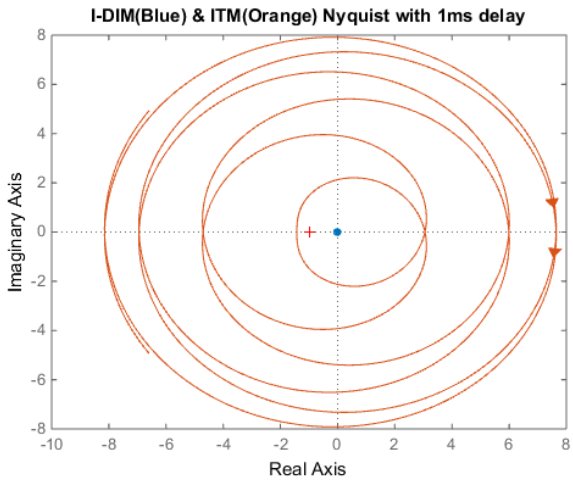

Figure 3.4 Nyquist Plot with $\boldsymbol{Z_{HuT}} = 5mH$

The DIM greatly increased stability of a system with conditions that would otherwise cripple ITM. The accuracy of the method must be ensured as well for a viable PHIL simulation. To test this, a Simulink model was created to assist this study. The following figure lists component values and placement for the simulation:

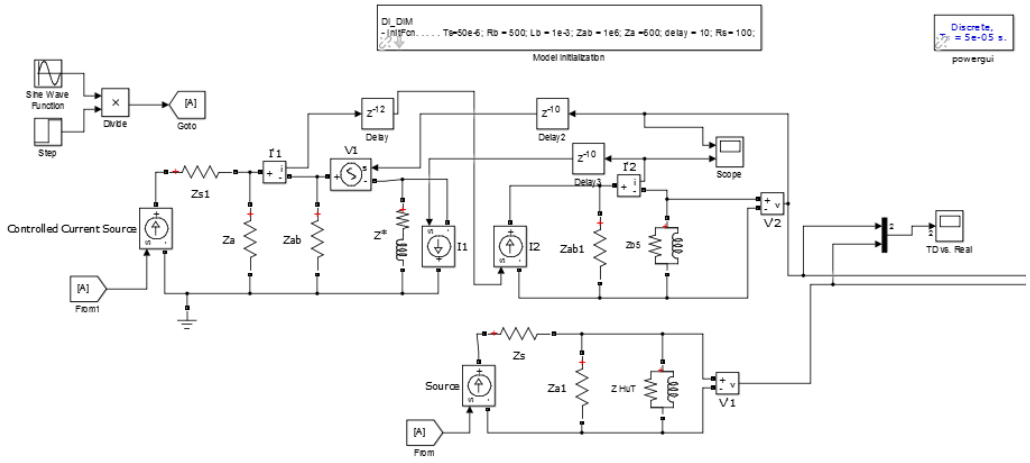

Figure 3.5 I-DIM simulation with delay of 1.67ms

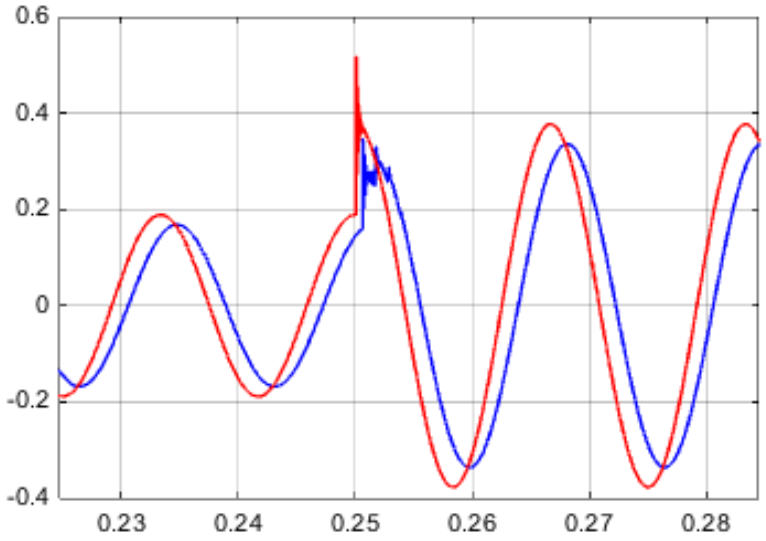

Figure 3.6 Output Current of Actual(Red) and DIM(Blue) with 500$\boldsymbol{\mu s}$ delay

The TFP error is caused by the delay as well as the non-unity gain of the interface. As seen above, the magnitude of the output does not accurately represent the system under normal conditions. Implementing DIM instead of ITM increases the accuracy, stability and complexity of the simulation. As explained earlier, I-DIM requires depends heavily on matching the impedance Z^* to the Hut impedance. When the two are identical, the IA remains stable under all conditions. The interface method still works admirably when the two values are kept quite similar. To achieve this in real-time, active tracking of the HuT's impedance must be implemented to account for step changes during operation.

Active tracking of the impedance can be expensive and prone to error. Possible strategies and shortcomings to accomplish this are discussed further in [9]. Beyond the active tracking of the impedance, Simulink does not have a controlled impedance block to reflect these constant changes and an improvised block would have to created and tested for accurate representation. Due to these shortcomings, ITM may still be the ideal IA to use for this simulation. This decision will be discussed in Chapter 4.

3.3 Frequency Domain Communication

Knowledge of the communication delay can be used to correctly synchronize with the local waveform. Any delay or time-shift between voltage and current in two subsystems will affect the fidelity of system, even when stable. Previous works have used this method to perform long distance simulator- to-simulator simulations from Aachen, Germany to Columbia, SC. As mentioned earlier, the technique sends the Laplace transform of the signal and inverts the transformation after receiving the data. Once synchronized, the two systems recreate the signal according to the local frame of reference.

To test the abilities of the FD Communication, a Simulink model was created to independently test the communication technique used in [8]. A simple 120Hz AC voltage source was sampled and input into the FFT block of Simulink, and inversely transformed. The received signal is then compared to the input signal as seen below

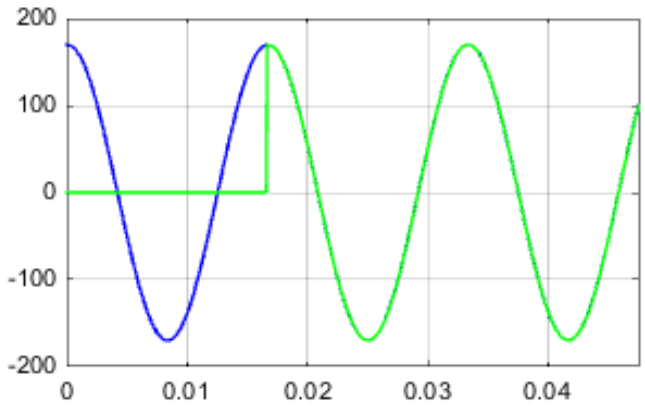

Figure 3.7 FFT of signal recombined and compared

With an applied delay of approximately a half cycle at 60Hz, 8.33ms, the signal is then compared to the original to show the effect of synchronization and FD communication. The received signal remains correctly positioned in relative phase to voltage or current. Only the magnitude of the signal is affected by the delay. Sudden steps in magnitude are not recorded until

one cycle has passed for calculation. This is due to the FFT's need of an entire cycle of the natural frequency to compute the output.

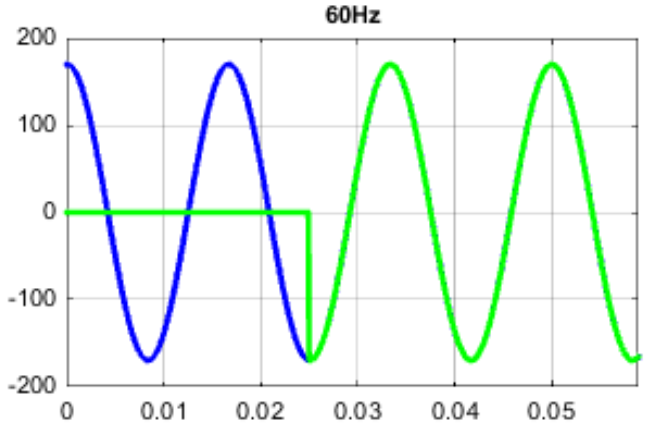

Figure 3.8: FD Communication delayed by 8.34ms

This additional delay only exacerbates the largest fault of basic FFT transformation. Transient conditions have a high chance of disrupting the conservation of energy across the interface. To mitigate this, a Recursive Discrete Fourier Transform (RDFT) found in [14] is utilized. Instead of requiring a cycle of sample points to transform and inversely transform, The RDFT maintains a moving window of sample points that update the output signal with each sample in an efficient manner. The block diagram of the RDFT algorithm is shown in Fig. 3.9.

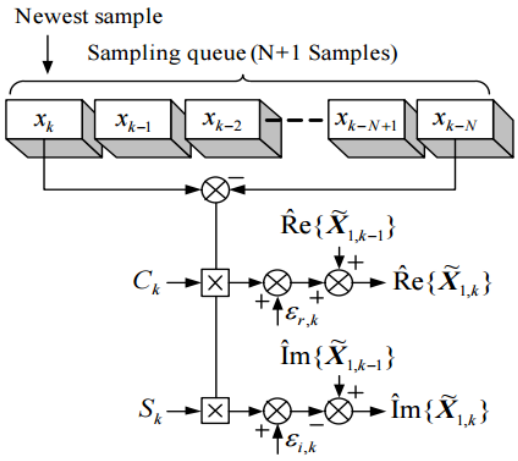

Figure 3.9 Recursive Discrete Fourier Transform architecture

The same input signal is input to the Custom RDFT block in Fig 3.10 and then recombined to the following output

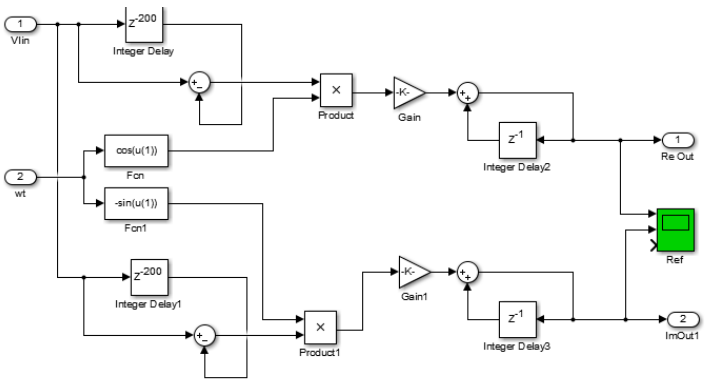

Figure 3.10 Simulink Custom RDFT Block Based on 200 Samples per cycle

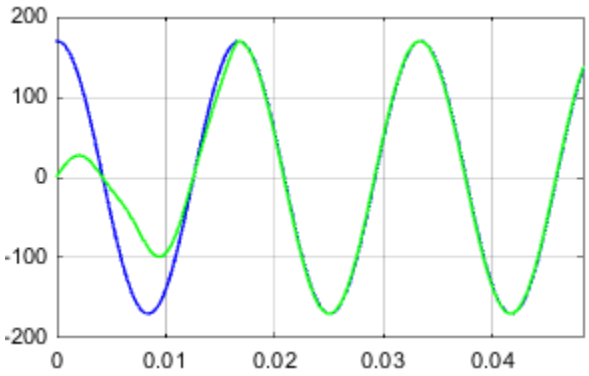

Figure 3.11 RDFT Output(Green) during from zero-to-full-step(Blue)

Not only does the RDFT start partially accounting for transient conditions immediately, but it does so while using the onboard memory and processing efficiently. The benefits of this will be explained in the implementation section below.

Overall, the FDC method would greatly mitigate steady-state error while minimizing the speed at which the packets would be sent. However, transient conditions would threaten this method just as much as basic TD communication. FDC may be beneficial but transient conditions must still be dealt with.

3.4 Model Partitioning & Low Pass Filtering

Ideally the partitioning of the model would nullify or cancel the adverse effects of the delay. As the delay affects the TFP, the partitioning of the model will inversely affect the TFP by containing most of the fast dynamics within the PCC. The effectiveness will be studied from a transfer function perspective as well as a simulated power system with a delayed ITM coupling.

Multi-rate Co-simulation has been performed in the past to fulfill particular objectives. Often certain sections of a system benefit from or require a small simulation time step. To reduce the computational stress or to run a larger system, different time steps are chosen according to the dynamics of the system. This allows a more efficient distribution of simulation resources and can increase fidelity or lower costs. A variation of the same has been accomplished by allowing the partition to operate independently from the rest of the system. By partitioning the PCC on the DSP and isolating high frequency components, the communication loop does not need to operate as frequently.

Contrary to popular literature on the subject, this work aims to break the model into separate systems to accommodate for the delay in communication. By moving part of the system to the other side of the delay, the high frequency dynamics can be managed and handled before entering the delayed communication loop. As a majority of high order harmonics in the system are heavily damped at a Point of Common Coupling (PCC), this technique uses this inherent effect to its advantage. In a more technical sense, the stability equation for ITM has been manually altered by increasing the impedance ratio across the delay.

The effect of partitioning is compared using a basic circuit's measurements before and after being torn and connected via a delayed ITM model.

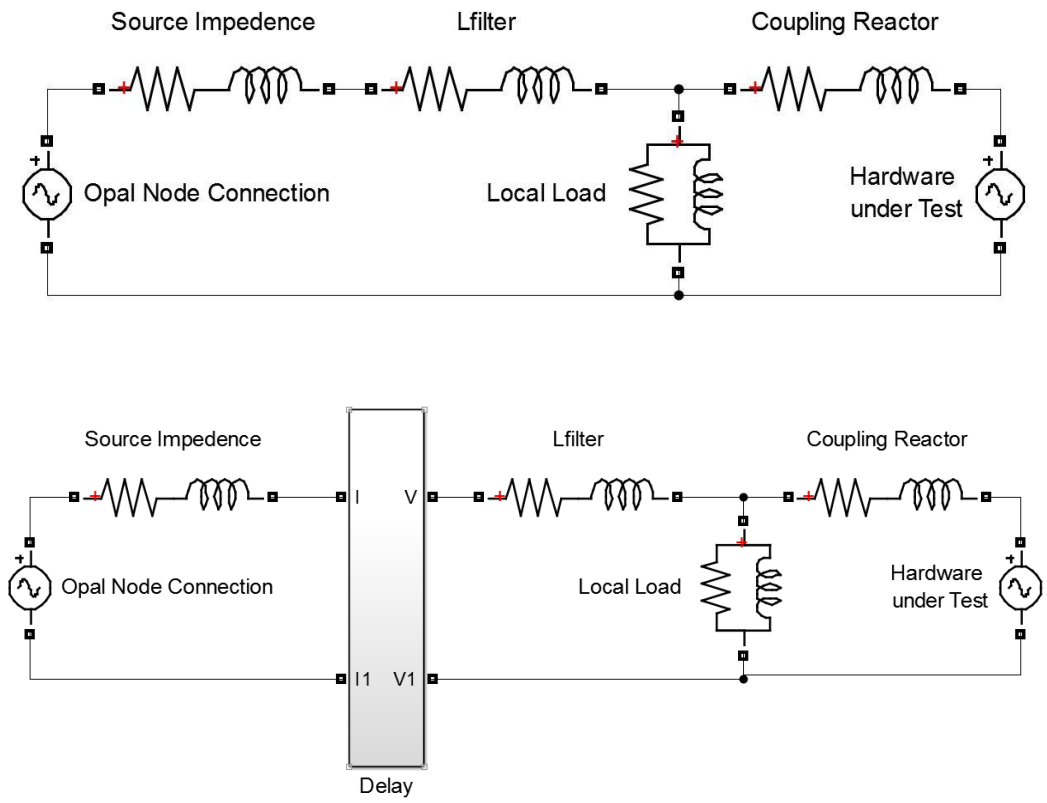

Figure 3.12: (Top) Test Case for Partitioning without delay and (Bottom) with delay

This setup should operate similarly to the first one except this simulation has active sources on both sides of the delay. Current measurements are used to check simulation fidelity under a variety of conditions.

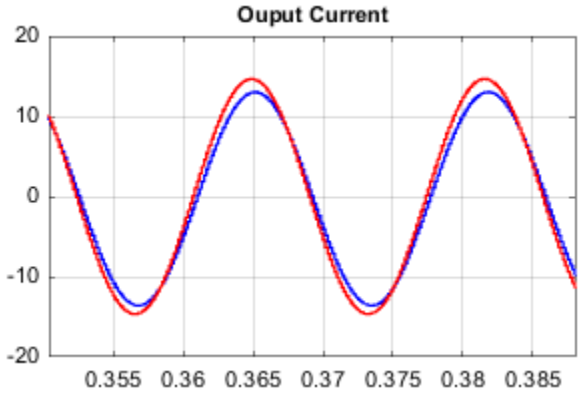

Figure 3.13 Output current comparison due to 166.7μs delay

Once the delay exceeds 1ms, the circuit with the delay becomes unstable due to the high frequencies harmonics in the Opal source. To stabilize the system and ensure high frequency noise does not amplify and destabilize the simulation, a LPF is tested with the above test circuit. The voltage feedback to Opal is filtered by a first order filter before transfer. This insures the loops transfer function remains below unity at high frequencies to protect from high frequency amplification in the loop. The effects of the filter can be shown by increasing the delay without destabilizing the circuit. The LPF is a simple filter with a cutoff frequency of 2000, which is the bandwidth of the current controller.

$$\frac{2000}{s+2000}e^{-st_d} \tag{13}$$

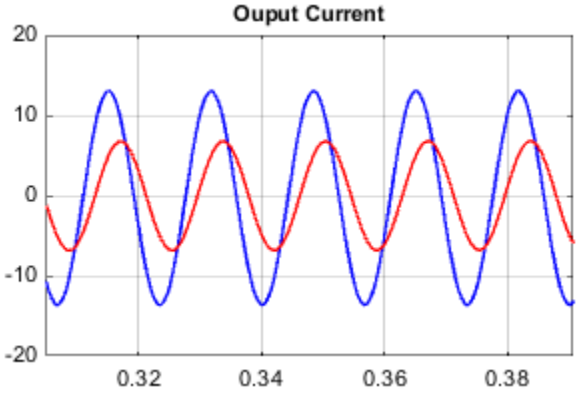

Figure 3.14 Extreme Delay of 15ms and Attenuation of Received Signal

Even though the system is stable, the accuracy of the interface and the TFP are greatly altered. In this way, the LPF must be used carefully as to not gain stability at the cost of accuracy. Partitioning combined with an LPF yields easily obtainable stability margins while allowing a high level of accuracy in most conditions.

Furthermore, manually stabilizing the system by increasing the impedance of the PCC alters the loop transfer function to achieve $Z_{PCC} > Z_{HuT}$. The PCC circuit can be modified to achieve a variety of goals depending on the PHIL simulation's purpose. Some frequency warping

of high-order harmonics may occur closer to the cutoff frequency, but the effect would be minimal. Consequentially, this method requires a monotonous setup as the PCC circuit must be discretely modeled and computed on the DSP. All of these characteristics will be discussed more in Chapter 4.

Chapter 4 : Implementation

4.1 Initial System Configuration

As seen in Ch. 3.4, initial simulation results of the aforementioned methods can create an accurate and stable simulation over the network. FDC allows for near zero steady-state error while partitioning of the system can stabilize the system and allow near instant responses to the transients on the hardware. The low pass filter can be used to alter the stability margin of the system at the cost of some high frequency attenuation. Lastly, the I-DIM can ensure stability and accuracy but severely increases the complexity and is sensitive to high delay. The feasibility of each of the methods will be discussed in detail below.

4.1.1 Opal Communication Strategy

Using the TMS320f2833 32 bit floating point DSP allows the user a lot of flexibility at the cost of ease of use. Alternatively, the OPAL-RT has a shallower learning curve for entry level projects and applications but requires in-depth analysis and reading to repurpose its more complicated functions. TCP/IP communication was chosen as the method of communication between the two devices to low overhead, message confirmation, and existing experience. For TCP/IP communication protocol to and from the OPAL-RT, a given data structure was supplied from RT-Lab. Any alteration of this structure for tailored needs caused a critical simulation error. Abiding by this structure, communication protocols were designed on the DSP to accommodate. Some complication was encountered because the W5300's communication is 16 bit and little Endian while the OPAL payload is organized in 64 bit big Endian. After receiving the packet, the 16 bit pieces received by the W5300 must be combined into the original number and then parsed to 32 bit floating points for all required calculations. Original troubleshooting between the two

devices proved to be difficult due to the setup required before each simulation. After an established method of basic data communication was established, the communication was integrated to the other functions of the DSP by replacing the original ADC input. After all necessary calculations, the resulting number is parsed to a 64 bit double and then converted back to a 32 bit float.

```
// Define the structure of the data that we send out. This is used in the
// SendToIPPort function
struct data_out
{
  short  dev_id;             // (2 bytes) Sender device ID
  int    msg_id;             // (4 bytes) Message ID
  short  msg_len;            // (2 bytes) Message length (data only)
  double data[MAXSENDSIZE];  // Up to MAXSENDSIZE doubles (8 bytes each)
};
//
// Define the structure of the data that we receive. This is used in the
// RecvFromIPPort function
struct data_in
{
  short  dev_id;             // (2 bytes) Sender device ID
  int    msg_id;             // (4 bytes) Message ID
  short  msg_len;            // (2 bytes) Message length (data only)
  double data[MAXRECVSIZE];  // Up to MAXRECVSIZE doubles (8 bytes each)
};
// ********************************************************************
```

Figure 4.1: TCP Package Structure within Opal

4.1.2 Frequency Domain Communication

To fully implement FDC, three things must be changed from the regular communication strategy. The number of signals between Opal and the DSP must be increased according to the bandwidth needed for simulation. All critical harmonics, such as the 3rd, 5th,7th, etc., would add 2 more signals each. If the controller has a bandwidth of 2 kHz for a 60 Hz system, the first 33 harmonics would be needed for complete periodic accuracy between the two systems. 66 total values would be sent every communication loop per measured signal. In a three phase system with dq transformation, 132 values would be required.

Secondly, all of the values would need to be stored in memory on both sides of the loop. Memory is not a large concern for Opal but the DSP has a limited amount of accessible onboard

memory. Even with the chips 256K by 16 Flash and 34K by 16 SARAM memory, the number of harmonics and sample points per cycle would need to limited greatly as calculations will be performed with IEEE 754 32 bit floating point notation.

Lastly, a Fast Fourier Transform must be performed to collect all necessary harmonics before sending. Once received, each of these waveforms must be combined into the resultant waveform. As the delay would result in a phase shift between the systems, the deterministic delay would be used to compensate and recreate the waveform correctly. Using a RDFT instead of a classic FFT minimizes the calculation window and number of operations but doubles the memory needed for the computation variables.

4.2 Final Configuration

4.2.1 Layout

The FD communication was initially chosen for this project to allow networked PHIL simulations due to its zero steady state error and compensation against delay. Although simulation results were successful, initial implementation proved unsuccessful as the synchronization of the two systems was not ensured as it had been in Simulink. When isolated in a controlled environment where synchronization was forced, transient conditions were tested with unwanted results. A step change on the HuT side resulted in a perturbation that impacted the simulation after a delay which then perturbed HuT after a similar delay. The implementation of the FDC lacks flexibility due to its constrained frequency components, computation, and memory requirements. A critical task of PHIL simulations is the study of transients and non-ideal conditions. Fidelity of the simulation in relation to real-world results would not be guaranteed under these conditions.

Once FDC was exhausted, model partitioning was chosen. The stabilizing effect of the non-delayed response from the state space model yields high flexibility with the delay. The

interconnect circuit that will interface the two circuits was chosen to be two RL line branches and one local RL load that can be used to replicate a local load. In the case of PV inverter test case, this state space model could represent local residential loads or a commercial building's consumption in a distribution-level simulation.

To validate initial findings and hypothesis, a test case was established. The following circuit ideally represents many of the components that would be present in a full system simulation.

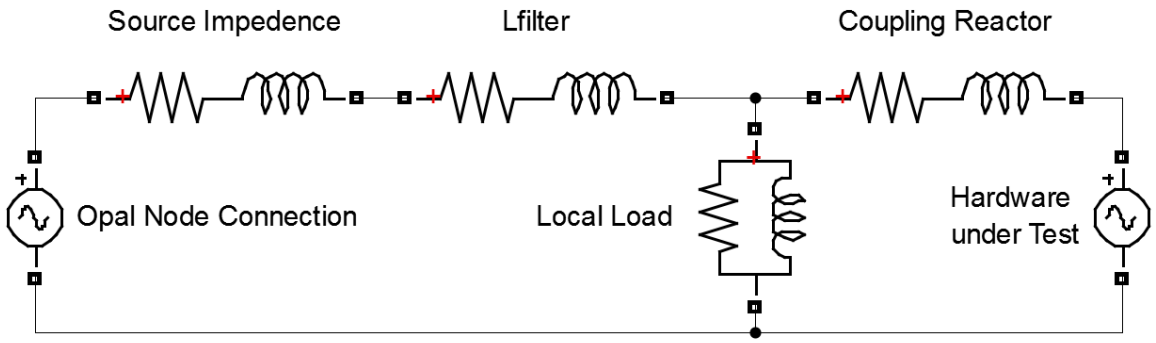

Figure 4.2 Test Circuit for System Validation.

Opal will be connected to the DSP through TCP/IP over Ethernet. The DSP will run a state space model as discussed earlier. The hardware side output of the SS model will feed directly to the current controller also running in the DSP. Due to the configuration, the only delay between the HuT and the state space model will be the insignificant two-cycle delay in the deadbeat controller. Delay length, inductor values, and step-changes will be chosen to analyze the robustness of this setup.

4.2.2 Partitioning

Creating a stable and accurate interface between the Opal simulation and the HuT with its own circuit model would allow a configurable interconnect between the two but would also introduce its own artifacts to the transfer function between systems. Further, the implementation

of the model within the DSP will have its own challenges. The modeled circuit would be created, tested in Simulink, then built as a continuous time state space model and discretized to run on the DSP. The DSP itself would have to solve for each time step of the model. The model is limited by the computational speed of the DSP because high-order matrix multiplication would not be completed within the allotted time step. The sampling and state space estimation method would use the same CPU-interrupt to ensure simplicity and consistent operation. Due to this, a trade-off must be established between the complexity of the partition and the sampling frequency.

Once the system was fully tested in Simulink and correct values were assigned to each component, the system must be discretized to implement on the DSP. A high value resistor was added in parallel to the current source to allow convergence of the state space system. The following state space equations were solved by hand.

$$\frac{dI_{L1}}{dt} = \frac{(I_g - I_{L1})R_X - (I_{L1} - I_{L2} - I_{L3})R_3 - I_{L1}R_1}{L1} \tag{14}$$

$$\frac{dI_{L2}}{dt} = \frac{(I_{L1} - I_{L2} - I_{L3})R_3 - I_{L2}R_2 - V_O}{L2} \tag{15}$$

$$\frac{dI_{L3}}{dt} = \frac{(I_{L1} - I_{L2} - I_{L3})R_3}{L3} \tag{16}$$

$$V_m = (I_g - I_1)R_x \;\&\; I_O = I_{L2} \tag{17}$$

Once established, the A, B, C, and D matrices were imported to MATLAB to verify the system's stability. The system was discretized using Backward Euler to ensure stability and simplicity. The discretized system can be found below as well. The system was ported into a

Simulink S-function and compared to the same circuit represented with Powerlib components. Figure 4.3 shows the layout of the entire simulation simplified.

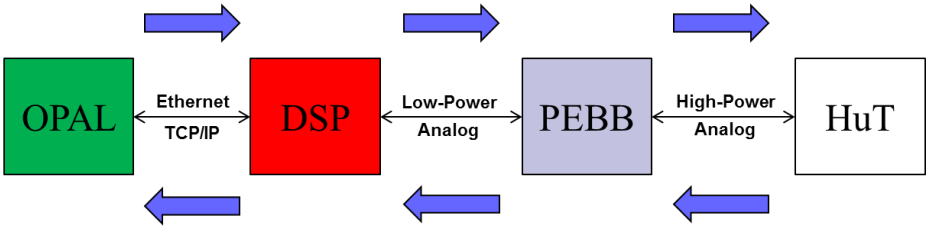

Figure 4.3 Overall Layout with connecting mediums

The model is discretized at the sample period of the DSP's control system. Discretizing the matrices in MATLAB and testing stability and response; before porting over to Code Composer Studio, aids in ensuring any changes to the model do not accidentally destabilize the model. Matrix multiplication was programmed into the SS_Calculation method to be completed once every sample period. Using the Direct Memory Access, the results will be saved in the SARAM for future calculations and communication. The structure of the DSP board allows for SS calculation and current control while intercepting data packets as well. The timing of these processes is carefully structured using two interrupts.

4.2.3 Sources of Error and Validation

To ensure correct operation, subsystems were tested individually before integrating the system together. The delay between Opal and the HuT would be inherently unstable due to the TFP, A filter model will act as a buffer between the two systems. This buffer will be used as a stabilizer to the overall simulation. As the stabilizing filter will consist of coupling inductors normally present in a PCC to the grid system, high frequency disturbances will be introduced to the grid in a controlled manner. The PCC can be configured to consist of an average model distributed load or a local load found in a Micro grid.

Validation on the DSP's ADC and value scaling was accomplished independently. Basic TCP/IP communication was tested and recorded in excel.

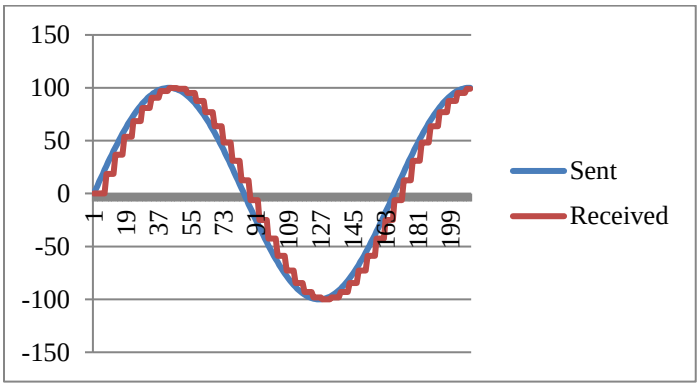

Figure 4.4 Using the DSP as an Echoserver

Other sources of error would include the ADC, DAC, variable component values and transient changes. Transient conditions suffer from the delay due to the delay of effect from the PCC circuit. The system was tested under the same conditions as discussed earlier.

Full system testing yields the reference signal from a non-partitioned circuit in Simulink to compare with the signal received after being transmitted to the DSP and the State Space Model before returning. As expected, the received signal is delayed and discrete.

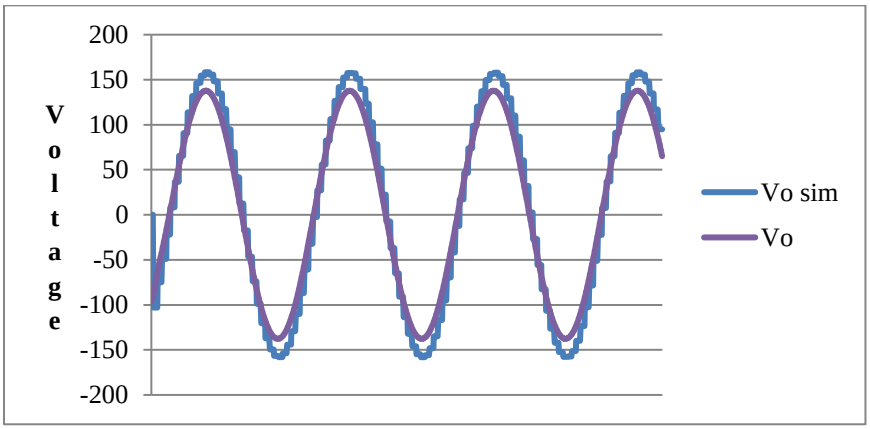

Fig 4.5: Fully Digital Simulation with Accurate recreation

As seen above, the voltage error at any given time can vary rapidly but the traditional methods are often not stable at all. As seen in Figure 2.12 in Ch.2, the errors caused by instability can contain large phase and magnitude errors. Creating an environment in which distance-inhibited co-simulations are increasingly more viable allows new areas of collaborative research. These results pave the way for future implementation of increasingly more complicated and detail oriented simulations in the realm of Networked PHIL simulation.

Chapter 5 : Conclusions

5.1 Summary

Throughout this work, the objective has been to create a delay-resistant PHIL interface between with the Delfino F28335 DSP and Opal RT simulator. Several techniques were explored in detail before partitioning a circuit model within the DSP. This decision was made with consideration to the complexity of setup, limitations of criteria, and simulation fidelity. PHIL simulations are much more prone to instability to the conservation of energy that must be maintained artificially. Due to complexities present in the I-DIM and FDC techniques, they must be tested, refined, and validated before being tested on hardware.

Partitioning in combination with a LPF yielded the best simulation results combined with ease of implementation. The use of a local load or filter should be beneficial to almost any PHIL simulation setup needed to test the system and the HuT. A stable and mostly accurate simulation has been allowed due to partitioning.

5.2 Recommendations and Future Work

An exhaustive list of test cases, limitations, and component values may be tested to fully validate these techniques. For the sake of this work, the aforementioned tests were used to ensure viability of the technique in a real-time real-world PHIL test. The experimental results of these tests were performed without the full power hardware equipment within the loop due to the lack of specific equipment parts and time. These tests ensure the stability and should be conducted using the PEBB amplifier.

Further TCP/IP communication structures could be optimized specifically for a PHIL loop with only the necessary resolution for each value instead of communicating with the 64 bit IEEE notation. Reducing the throughput needed per step, memory & clock cycles needed to fulfill the requirements. Using State Space Nodes (SSN) in Opal would allow a larger grid model to run efficiently as multiple co-simulated nodes on the RT simulator.

Developing I-DIM to allow real-time impedance tracking would create a very viable help that could supplement other techniques to increase stability and accuracy. To accomplish this, a monitoring device would likely be dedicated to the task of calculating the HuT's impedance at the fundamental frequency as well as important harmonics. The online calculation of the value would negatively impact the simulation's accuracy and stability if not properly matched or if not properly tuned to respond to large variations of the load's impedance.

Future work includes creating a RDFT communication loop with steady-state monitoring to detect which communication technique would be best suited to the current sample. If this algorithm was able to quickly and accurately utilize the best of both communication modes, simulation accuracy under a variation of transient and steady-state conditions would greatly increase.

Partitions should be further quantified and tested for validity under non-ideal circumstances. One such circumstance would involve non-periodic load compensation. If a nearby Opal node contains disturbances due to an arc furnace or thyristor bridge, an actual compensator in the lab could be tested under a slew of operational possibilities. The compensator would be connected to the PEBB which is controlled according to the PCC circuit output and control loop.

Implementing multiple techniques and optimizing currently proposed strategies would possibly allow the best qualities of each one to be gleaned and tested. Once setup, this PHIL

testbed can easily be altered to include other interesting parts and power electronics. The possible testing for renewable resources, distributed energy resources, and fault conditions are numerous and varied.

.

References

[1] J. Bélanger, L. a Snider, J. Paquin, C. Pirolli, and W. Li, "A Modern and Open Real-Time Digital Simulator of Contemporary Power Systems," *Design*, pp. 1–10, 2009.

[2] A. Monti, E. Santi, R. A. Dougal, and M. Riva, "Rapid prototyping of digital controls for power electronics," *IEEE Trans. Power Electron.*, vol. 18, no. 3, pp. 915–922, 2003.

[3] M. Bacic, "On hardware-in-the-loop simulation," *Proc. 44th IEEE Conf. Decis. Control*, pp. 3194–3198, 2005.

[4] R. Isermann, J. Schaffnit, and S. Sinsel, "Hardware-in-the-loop simulation for the design and testing of engine-control systems," *Control Eng. Pract.*, vol. 7, no. 5, pp. 643–653, 1999.

[5] A. Benigni and A. Monti, "A Parallel Approach to Real-Time Simulation of Power Electronics Systems," *Power Electron. IEEE Trans.*, vol. 30, no. 9, pp. 5192–5206, 2015.

[6] L. Gregoire, W. Li, J. Belanger, and L. Snider, "Validation of a 60-Level Modular Multilevel Converter Model-Overview of Offline and Real-Time HIL Testing and Results," *IPST 2011, Int. Power …*, pp. 1–6, 2011.

[7] X. Wu, S. Lentijo, and A. Monti, "A novel interface for Power-Hardware-In-the-Loop simulation," *Proc. IEEE Work. Comput. Power Electron. COMPEL*, no. 1, pp. 178–182, 2004.

[8] M. Stevic, A. Monti, and A. Benigni, "Development of a simulator-to-simulator interface for geographically distributed simulation of power systems in real time," *41st Annu. Conf. IEEE Ind. Electron. Soc. IECON*, pp. 5020–5025, 2015.

[9] R. M. Mersenski, *Evaluation of A New Power-Hardware-In-The- Loop (Phil) Interface Algorithm For Current Controlled Amplifiers*. 2011.

[10] B. Palmintier, B. Lundstrom, S. Chakraborty, T. Williams, K. Schneider, and D.

Chassin, “A Power Hardware-in-the-Loop Platform with Remote Distribution Circuit Cosimulation,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2236–2245, 2015.

[11] X. Wu and A. Monti, “Methods for partitioning the system and performance evaluation in powerhardware-in-the-loop simulations - Part II,” *IECON Proc. (Industrial Electron. Conf.*, vol. 2005, pp. 257–262, 2005.

[12] W. Ren, “Accuracy Evalaution of Power Hardware-in-,” 2007.

[13] M. Dargahikafshgarkolaei, “Stability Analysis and Implementation of Power-Hardware-in-the-Loop for Power System Testing,” 2015.

[14] H. L. Ginn and G. Chen, “Digital control method for grid-connected converters supplied with nonideal voltage,” *IEEE Trans. Ind. Informatics*, vol. 10, no. 1, pp. 127–136, 2014.

[15] V. B. Dmitriev-Zdorov, “Generalized coupling as a way to improve the convergence in relaxation-based solvers,” *Proc. EURO-DAC ’96. Eur. Des. Autom. Conf. with EURO-VHDL ’96 Exhib.*, pp. 15–20, 1996.