University of Nottingham
UK | CHINA | MALAYSIA

A Fast Remotely Operable Digital Twin of a Generic Electric Powertrain for Geographically Distributed Hardware-in-the-Loop Simulation Testbed

Surojit Sen, MSc.

Submitted to the University of Nottingham for the degree of Doctor of Philosophy,

January 2022

i. Abstract

The automotive industry today is seeing far-reaching and portentous changes that will change the face of it in the foreseeable future. Digitalisation and Electrification are two of the key megatrends that is changing the way vehicles are developed and produced. A recent development in R&D process is the Hardware-in-the-Loop (HIL) method that uses a hybrid approach of testing a physical prototype immersed in a virtual environment, which is nowadays being creatively re-applied towards geographically separated multi-centre testing strategies, that suits the horizontally integrated and supply-chain driven industry very well. Geographical separation entails the deployment of a "Digital Twin" in remote centre(s) participating in multi-centre testing. This PhD aims to produce a highly robust, efficient, and rapidly computable Digital Twin of a generic electric powertrain using the multi-frequency averaging (MFA) technique that has been extended for variable frequency operation. This PhD also aims to commission a local HIL simulation testbed for a generic electric power inverter testing. The greater goal is to co-simulate the local HIL centre testing a prototype inverter, and its Digital Twin in a different location "twinning" the prototype inverter as best as possible.

A novel approach for the Digital Twin has been proposed that employs Dynamic Phasors to solve the system in the frequency domain. An original method of multiplication of two signals in the frequency domain has been proposed. The resultant model has been verified against an equivalent time domain switching model and shown to outperform appreciably. A distinctive advantage the MFA Digital Twin offers is the "fidelity customisability"; based on application, the Twin can be set to compute a low (or high)-fi model at different computational cost. Finally, a novel method of communicating high-speed motor shaft position information using a low-speed processing system has been developed and validated. This has been applied to run real-life HIL simulation cycles on a test inverter and effects studied.

The two ends of a multi-HIL testbed, i.e., local HIL environment for an inverter, and its Digital Twin, has been developed and validated. The last piece of the puzzle, i.e., employing a State Convergence algorithm to ensure the Digital Twin is accurate duplicating the performance of its "master", is required to close the loop. Several ideas and process plans have been proposed to do the same.

ii. Acknowledgements

First and foremost, I wish to express heartfelt gratitude for my parents, Susmita and Abhijit, and my brother, Sanket, for having faith in me and providing unending encouragement and emotional support. My family has been my support system over these years, and I cannot imagine finishing this PhD without them. I must also thank Shreeja for offering much needed support in the last year and cheering me on to the finish line.

I would like to thank my two supervisors, Dr Paul L Evans for his continued academic support and help with day-to-day challenges, and Prof C Mark Johnson for granting me the opportunity to study this topic and giving a strong direction to this PhD. I would also like to thank my moderation supervisor Dr Serhiy Bozhko for keeping my progress in check periodically and offering an outside perspective to various problems during the study. I also appreciate the technical support provided by the department's technician and safety team (Mr Colin Blackburn, Mr Chris Varley, Mr Dan Bathgate, and others). Lastly, I also appreciate the customer support provided by the Triphase (now a part of National Instruments) technical team in ironing out the bugs in the simulator rig.

iii. Contents

iv. List of Figures

v. List of Tables

vi. List of Abbreviations

ABS	Anti-lock Braking Systems
AC	Alternating Current
AD	Automatic Differentiation
AFE	Active Front-end
APC	Advanced Propulsion Centre
BEV	Battery Electric Vehicle
C	Capacitor-only filter
CAN	Controller Area Network
CASE	Connected, Autonomous, Shared, and Electrified
CB	Converter Bridge
CCS	Controller Current Source
C-HIL	Controller Hardware-in-the-Loop
CPU	Central Processing Unit
CS	Current Source
CVS	Controller Voltage Source
DC	Direct Current
DT	Digital Twin
DIM	Damping Impedance Method
DORC	Degree of Remote Coupling
DP	Dynamic Phasor
DSP	Digital Signal Processor
DSP	Digital Signal Processor
DTM	Direct Torque Modulation
DUT	Device Under Test
EMaT	Electromagnetic Transient
EMeT	Electromechanical Transient
EMF	Electromotive Force
EMTP	Electromagnetic Transient Program
EV	Electric Vehicle
FFT	Fast Fourier Transform

FPGA	Field Programmable Gate Array
FS	Fourier Series
GAM	Generalised Average Modelling
GD-HIL	Geographically Distributed Hardware-in-the-Loop
GHG	Greenhouse Gases
GPU	Graphics Processing Unit
GSSA	Generalised State Space Averaging
HIL	Hardware-in-the-Loop
HVDC	High Voltage Direct Current
IA	Interface Algorithm
ICE	Internal Combustion Engine
ID-HIL	Internet Distributed Hardware-in-the-Loop
IEEE	Institute of Electrical and Electronics Engineers
IET	Institution of Engineering and Technology
IGBT	Insulated Gate Bipolar Transistor
ITM	Ideal Transformer Method
IUT	Inverter Under Test
L	Inductor-only filter
LC	Inductor-Capacitor filter
LCC	Inductor-Capacitor-Capacitor filter
LCL	Inductor-Capacitor-Inductor filter
LCLC	Inductor-Capacitor-Inductor-Capacitor filter
LTI	Linear Time-Invariant
MAAS	Mobility-as-a-Service
MFA	Multi-Frequency Averaging
MOSFET	Metal-Oxide-Semiconductor Field Effect Transistor
MSc	Master of Science
NASA	National Aeronautics and Space Administration
NP	Noise Perturbation
ODE	Ordinary Differential Equations
OEM	Original Equipment Manufacturer
OS	Operating System
P2T	Phase-to-Time

PC	Personal Computer
PCD	Partial Circuit Duplication
PCI	Peripheral Component Interconnect
PCM	Powertrain Control Module
PEC	Power Electronic Converter
PhD	Doctor in Philosophy
P-HIL	Power Hardware-in-the-Loop
PMSM	Permanent Magnet Synchronous Machine
PV	Photovoltaic
PWM	Pulse Width Modulation
RCF	Resistive Companion Form
RCP	Rapid Controller Prototyping
RP	Rapid Prototyping
RPM	Revolutions Per Minute
RS232	Recommended Standard 232
RT	Real-Time
RTDS	Real-Time Digital Simulator
RTOS	Real-Time Operating System
RTT	Real-Time Target
RTT	Round-trip-time
SC	State Convergence
SPWM	Sinusoidal Pulse Width Modulation
SRC	Signal-extension Resistive Companion
SSA	State Space Averaging
SVM	Space Vector Modulation
SVPWM	Space Vector Pulse Width Modulation
TCP	Transmission Control Protocol
TCS	Traction Control System
TFA	Time-variant First Order Approximation
TFP	Transfer Function Perturbation
THI	Third Harmonic Injection
TLM	Transmission Line Method
TNA	Transient Network Analyser

TSB	Time Stamp Bridges
UDP	User Datagram Protocol
V2G	Vehicle-to-Grid
VCHV	Virtually Connected Hybrid Vehicle
VPPC	Vehicular Power and Propulsion Conference
VS	Voltage Source
VSC	Voltage Source Converter
VSS	Virtual System Simulation
VTB	Virtual Testbed
VTB-RT	Real-Time Virtual Testbed
WTHD	Weighted Total Harmonic Distortion
WTHD0	Weighted Total Harmonic Distortion referenced to modulation wave
XSG	Xilinx System Generator

CHAPTER 1 Introduction

1.1 Automotive industry – Past, Present and Future

The automotive industry comprises of all the businesses involved in the production of motor vehicles, that includes the powertrain (engine, transmission, and more recently traction batteries and drives), the body (chassis and panels) and the electronics (display, lights, computing, and software). A little more than a century old, this industry has today grown into a significant economic driving force for most industrial nations around the world and accounts for more than 5 trillion dollars in revenue [1]. Not just economically, the industry has also been on the forefront of technological development and is currently undergoing arguably its grandest and most disruptive changes in its history [1][2][3]; **connected**, **autonomous**, **shared** and **electrified** (CASE) car of the near future. *This thesis is the outcome of the work conducted for the degree of doctorate in philosophy, which facilitates significant reduction of time and effort spent in developing a modern automotive powertrain (hybrid/electric).* Therefore, it is prudent to introduce the reader briefly about the origin and evolution, and how is the future getting shaped.

Fig 1-1 Ford Model T Touring Car from 1925 [4]

The first reciprocating engine-driven “automobile” was developed in late the 19th century as a “horseless carriage” in Europe, although steam and electric powered modes of transport were in use for niche applications well before that [5]. By the turn of the twentieth century, numerous other European countries had joined in but the real

revolution that cemented the automotive “industry” firmly was Ford Motor Company in America with its assembly line production (Fig 1-1). It is worth noting here that the popular gasoline-powered engine was not the first choice during the inception era of the car as it faced stiff competition between electric and steam power. Electricity mainly because it was minimal maintenance, easy to operate, and noise-free; whereas the steam-powered powertrain was a more serious rival since it was a more mature technology and aided by general adoption by the populace. There are several reasons given in [6][7] as to why electric vehicles never succeeded in garnering market share against the fuel-powered cars during the 1920s and 1930s. Mainly, the problems faced by electric propulsion were extremely limited battery capacity and negligible charging infrastructure. The main problem for steam propulsion was expensive construction of boilers light enough for road cars. Thus, began the era of internal-combustion engine (ICE) automobile as we see today.

Over the decades the technology has evolved gradually to account for passenger and pedestrian safety standards, driving experience and creature comfort, while the business model of private and fleet ownership remained largely the same. Improvement of fuel economy was initially a slow burner among areas of research, but became a mainstay after the global oil crisis in the 1970s. As a result, the latter half of the 20th century saw rapid advancement of the internal combustion engine in terms of efficiency and there was a marked shift towards the more efficient diesel engines in Europe. Moreover, there was a marked rise in government interest to fund research in electrified powertrains for road transport as well.

Climate change has been another crucial factor towards the desire for electrification in the automotive industry. Electrified powertrains are advantageous because they have zero tailpipe emissions that helps improve local air quality, which had started to become a major health concern in big cities in the 1990s. Another big reason is reduction in carbon footprint per mile driven for an electrified vehicle. While it may be argued that there is significant carbon impact to generate electricity if the national energy grid is mostly reliant on fossil-fuel based sources, the lifecycle emissions of an electrified vehicle (EV) is still better than ICE vehicles [8]. Most progressive nations in modern

times have a sizeable portion of their energy generation from renewable sources which further helps the case. Fig 1-2 below shows that lifetime greenhouse gas emission of an EV varies with the national energy grid quality but is still mostly better than IC vehicles. Moreover, the IC engines after decades of incremental improvements have now reached their peak technological maturity and hence electrification is arguably the best way forward. *As a result,* ***electrification*** *is one of the key megatrends that is shaping the future of the industry.*

The future of the automotive industry and market will see dramatic changes as an exponential rise in digitalisation and Internet of Things (IoT) have continued to shift the cost balance from steel to silicon in vehicle engineering [2]. Big tech giants from the Silicon Valley (Apple, Google, Tesla) have entered the field with vast investment towards **connected** and **autonomous** services for a vehicle. Secondly, the business model of private ownership is gradually giving way to community ownership in the form of “Mobility-as-a-Service” (MAAS) industry as opposed to the traditional automotive industry; research has found a privately owned vehicle spends more than 90% of its lifetime simply parked, increasing the utilisation fraction of a vehicle through vehicle **sharing** holds the potential to drastically reduce the lifecycle cost. These four megatrends of the industry, popularly referred to by the acronym CASE (connected, autonomous, shared, electrified), are complexly interlinked and have forced legacy manufacturers to submit to the post-modern automotive world of a cheap, feature-rich, and highly connected car.

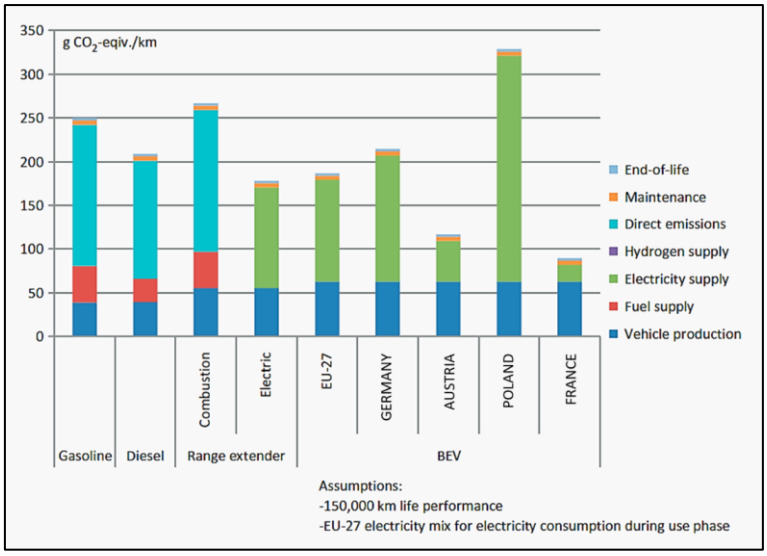


Fig 1-2 Lifetime greenhouse gas (GHG) emissions of types of vehicles and sub-divided by countries for Battery Electric Vehicles (BEV) [8]

1.2 Key Technological Enablers of Electrification

Santini [5] notes that the history of EVs can be divided into three phases. The first is the pre-IC era when a small minority of road transportation was done using EVs until the IC engine car completely took over by the 1920s due to reasons discussed earlier. The second era began in the 1970s with the oil crisis when various countries in America and Europe began research in vehicle electrification. The momentum was soon lost within the decade due to drop in oil prices. One of the major reasons that EVs did not take off in the market is that the technology was not mature enough. This changed by the turn of 21st century when governmental push was complemented by major strides in technological advancements for widespread uptake in the consumer market. There are three primary technology enablers that have helped make a strong case towards automotive electrification [5]: **energy storage**, **energy conversion** and **energy control**.

First, the development of the Lithium-Ion battery [9][10] *that started in late 90s.* Having a light compact source of energy has always been the bottleneck for widespread adoption for EV and Li-Ion chemistries today offer the best prospects in this regard. Li-Ion cells have been under research for more than two decades now and started off as energy source for portable appliance like mobile phones and laptops. While they are

the best prospects today for automotive applications, there is still a long way to go to compete with the energy and power densities offered by ICE powertrains. They are not very safe to operate and have a sizeable environmental impact during mining and manufacturing.

Second, the development of modern power electronics [11] *that started in the 80s.* Emergence of solid-state switching devices for power applications (thyristors, MOSFETs, IGBTs) significantly accelerated the pace of improvements of drive systems towards smaller, lighter, and more efficient units. This is especially valuable for automotive applications that require low power but high-density units that can sustain a wide range of temperature swing and mechanical vibrations and harshness.

Third, the development of electronics and software that have facilitated lightning-fast computing in small and light package. The Moore's Law states that the density of transistors in an IC package doubles every two years [12], something which has been observed for many decades now (Fig 1-3). The automotive industry has benefited its fair share from this advancement as well and modern cars have exceedingly complex software behind the optimisation of various systems in the powertrain like ignition timing, transmission control and vehicle dynamics related safety features like ABS (**Anti-lock Braking System**, prevents braked wheels from locking up and losing traction) and EBD (**Electric Brake force Distribution**, cleverly manipulating the braking force on the inside and outside wheels to aid the driver in keeping the vehicle pointing in the intended direction). Modern electric drive control schemes are complex and are only possible due to the advent of electronics.

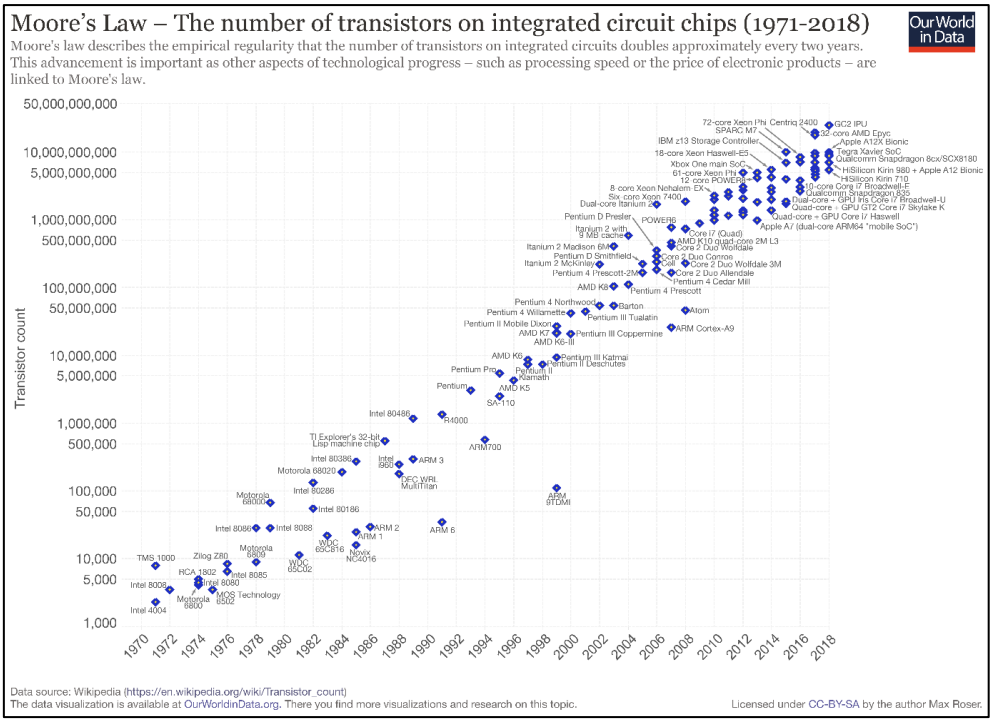


Fig 1-3 Moore's Law - The number of transistors on integrated circuit chips (1971-2018) [12]

These three enabling technologies, mostly the latter two (fast power electronics and digitalisation) have increasingly trickled into R&D methods and practices in the form of ***Virtual Prototyping***. Modern computers are required in every step of development in the form of CAD (**Computer Aided Design**, software packages that are used to draw shapes and assemblies of product) and CAE (**Computer Aided Engineering**, an umbrella term used to indicate all engineering analysis tasks like **Finite Element Analysis**, **Fluid Dynamics,** and **Circuit Simulation** for electrical and electronics). While this evolution has increased the complexity by bringing together new disciplines to solve common goals, it has also opened many avenues to reduce costs by making the development process more automated and simulation intense. While the need for a real prototype is still unavoidable (since digital models can only reach limited levels of accuracy), this step can be pushed further back in the development cycle, dropping the overall cost of designing.

Another significant opportunity that has emerged in the last two decades is ***hardware-in-the-loop*** (HIL) simulation for testing and validation. A HIL platform allows real prototype units to be connected in a digital closed-loop virtual system for real-time

simulation. This brings the best of both worlds together; highly accurate behaviour of components that cannot be modelled to high fidelity may be incorporated in the testing process. As the rest of the system is in a virtual environment, multiple test iterations are guaranteed to be repeatable.

1.3 Key Challenges in Engineering a Modern Car

Electrification was identified as one of the biggest disruptive changes taking place in the automotive industry due to several reasons but primarily air quality and climate change. Since the associated technologies are still about a decade away from reaching price and weight parity, an EV is still significantly expensive and heavy when compared with an equivalent IC engine car. As a result, engineers are trying to wring out every drop of efficiency that can be achieved in an electrified powertrain through system optimisation and ensuring every component is purpose-built to work hand in hand with the other components. This requires rigorous testing of components at a system level which may get very expensive if real hardware is used.

Virtual Prototyping is a major enabler of repeated testing and validation through reasonably accurate models of various components, as it avoids building a physical prototype. But there are two areas where Virtual Prototyping does not apply and falls short. First, there are numerous components that cannot be modelled to a high degree of fidelity. Consequently, a physical prototype must be built that must be proved in a virtual environment. Second, at system level simulation the overall accuracy is lower than individual components since the error margins of each component tend to compound together.

This expands the remit of Virtual Prototyping into a “hybrid” of both Real and Virtual elements in the simulation (Fig 1-4). HIL simulation is a specialised field of Virtual Prototyping that addresses that by providing for an interface between the real and virtual worlds. A real prototype may be connected in this HIL testbed or rig and operated as if it was fully immersed in the virtual closed-loop system, where the hardware would appear as a black box that reacts exactly as the hardware does. HIL simulation has

emerged as a mainstay since the early 2000s which is especially crucial in developing electrical systems in the car.

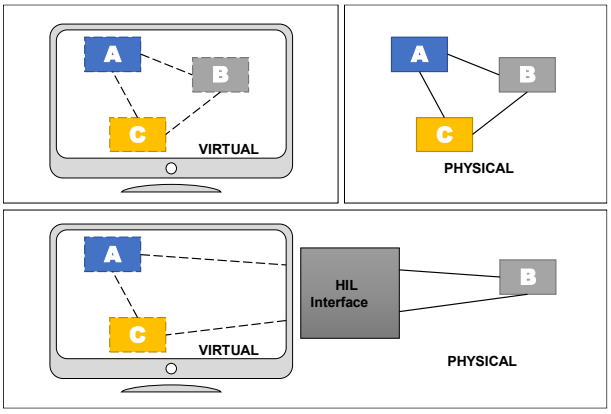


Fig 1-4 Underlying concept of Hardware-in-the-Loop (HIL) - Hybrid of real and virtual systems operating in tandem

In automotive powertrains, especially hybrid electric powertrains, there are several components that are very complexly integrated together. While conducting system testing, there may be a case where more than one component would require HIL testing. It is straightforward to combine multiple prototypes together and form a single hardware if they are linked together in the closed-loop system, but any other scenario would require multiple HIL rigs to immerse these physical components in the virtual system. Apart from the obvious cost concern to commission and maintain multiple HIL rigs, a technical issue is not necessarily there with this setup.

The real problem of multiple HIL setup arises when different HIL prototypes are not co-located. This is common in the automotive industry as it is more horizontally integrated as opposed to vertically. The end-product is hugely dependent on an elaborate network of supplier companies. As an example [13], the popular car Opel/Vauxhall Insignia has a multitude of independent suppliers (Fig 1-5). Different companies have their respective intellectual property protection measures in place which adds hindrance to system prototype development with “reserved” technologies. Hence, there is also a logistic barrier to bring all component prototypes together at one place since research centres for these components are usually dispersed geographically. So, having a multi-HIL simulation testbed virtually connected would benefit the R&D process in the industry

by allowing system level HIL simulations without having to co-locate various prototypes to enable unparalleled freedom of experiments.

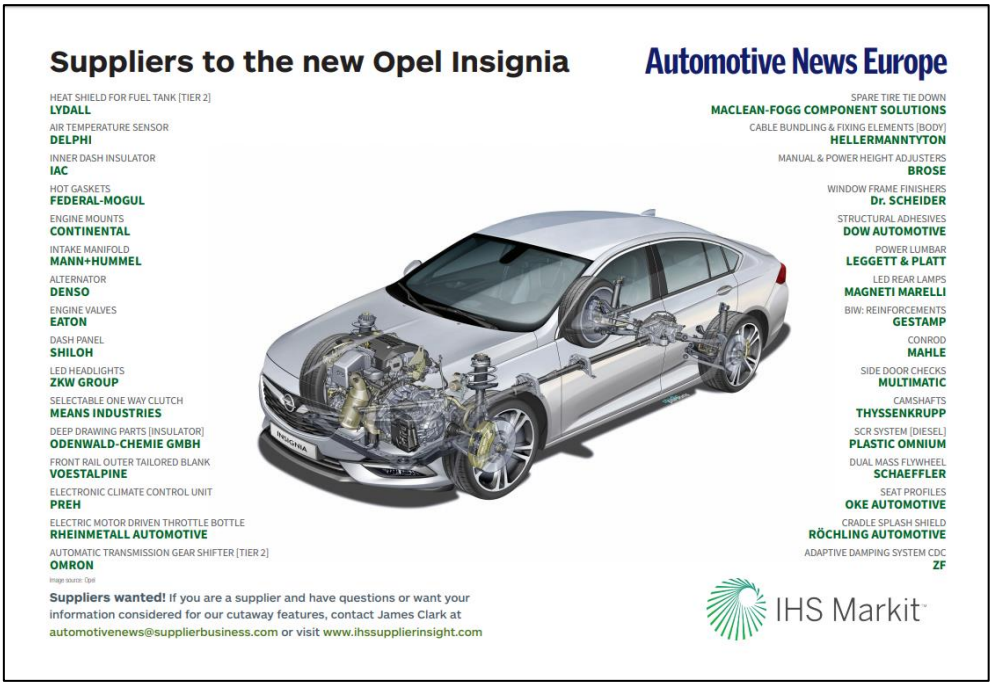


Fig 1-5 Suppliers to the Opel/Vauxhall Insignia, demonstrating the hugely scattered supply chain network in the automotive industry [13]

Delving deeper in imagining this concept, there are several foreseeable challenges in realising such a virtual simulation testbed. An apparent concern is setting up a reliable and fast communication “tunnel” between various locations for stable and real-time co-simulation. This challenge arises due to the requirement for breaking the physical interface between real-components and replacing it with a virtual communication medium (like the internet) that is inherently flawed due to finite delay and *jitter*. Jitter refers to the quality of communication to reorder bits of information upon receipt at the destination.

1.4 Virtually Connected Hybrid Vehicle

Considering the increasing pressure from the governments and emerging technological opportunities, the industry has come together with the aid of public funding to reduce developmental expenditure by exploiting digitalisation and virtualisation. Virtually Connected Hybrid Vehicle (VCHV) is one such project. A virtual simulation platform to

undertake geographically distributed multi-HIL simulation (Fig 1-6) of a generic automotive powertrain was proposed 5 years ago with a vision to reduce development costs and time-to-market of new powertrains. VCHV is one of the many projects funded by the Advanced Propulsion Centre (APC) [14][15], which is the UK government's initiative to propel the country to become the global hub of research, development and production of low carbon propulsion technologies.

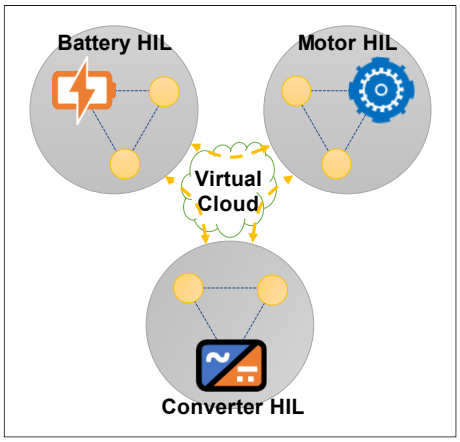


Fig 1-6 Underlying concept of Geographically Distributed Hardware-in-the-Loop (GD-HIL) - HIL centres in separate locations operating in tandem

APC operates through its six centres [16] that are technological centres of excellence in their respective fields pertaining to hybrid propulsion system (battery, motor, engine etc.). VCHV project proposes to host a distributed simulation of the entire powertrain with each of these six centres behave as a HIL centre for its respective powertrain component (Fig 1-7). Since something like this has never been done before, the project would start with a feasibility study and arrive at the best opportunity for implementation of a working proof of concept.

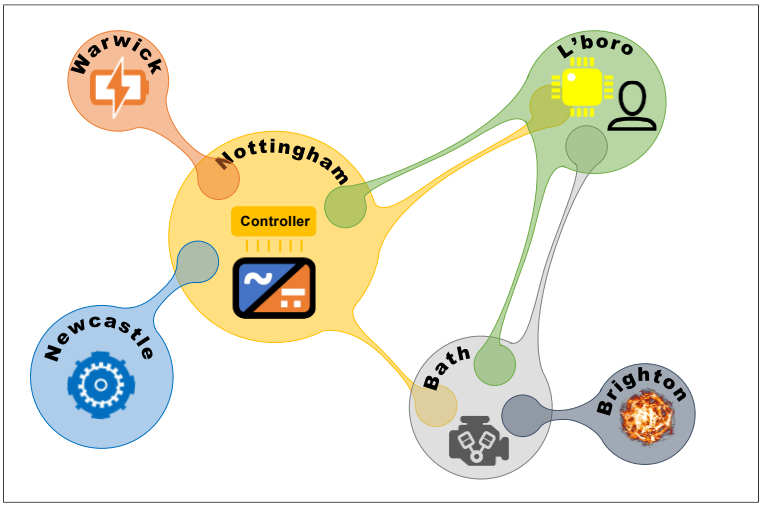


Fig 1-7 Proposed multi-centre GD-HIL virtual simulation testbed

University of Nottingham is the **power electronics centre** and strives to deliver the enabling technology to bring a generic automotive power converter "online" in this distributed HIL simulation platform. This PhD aims to explore the state of the art in geographically distributed HIL simulation for automotive hybrid electric powertrains and investigate and identify suitable architecture for the VCHV HIL system. Being the power converter HIL centre, it is aimed to "project" a replica model of a generic electric powertrain to other HIL centres like the **engine centre** and **vehicle controller centre** based on real-time operation of a physical power converter locally and the same for motor and battery remotely.

1.5 Intent and Novelty of this PhD Project

As with any fresh project, a scoping exercise was done to investigate the state of the art in technology and what could be achieved in terms of distributed simulation with the current technology in computing and communications. It may be reasonably inferred that it would be impossible to virtually connect two "halves" of a system and run a simulation cycle to study certain effects of interest that have time constants comparable to the delaying effects of the virtual connection. A tight degree of coupling may warrant the need of a local "surrogate" model that is able to replicate the immediate reactions of the hardware in the short-term and rely on periodic realignment information form the hardware over long-term to keep itself from diverging. Fig 1-8 illustrates this proposition

with a hypothetical scenario. The simulation time constant is 1ms which is analogous to the short term. The virtual connection time constant is 10ms which is the long term. This surrogate model (henceforth referred to as the “Digital Twin”) reacts to the simulation in centre B every 1ms. The HIL simulator in centre A sends a realignment packet to its Digital Twin every 10ms hence keeping it in check.

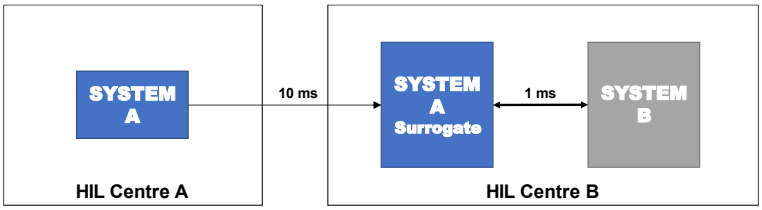


Fig 1-8 Concept of local "surrogate" model dealing with instantaneous responses locally, periodically realigned by actual model over remote communication

Especially true for electrical systems, the waveforms of interest have very short time constants generally in the order of microseconds. This observation made it distinctly clear that a Digital Twin is necessary for electrical systems in this application and much of the PhD was spent towards finding and implementing the best possible Digital Twin. This model of the real hardware shall be deployed at every remote location that would directly interact with the hardware at that remote location (via the HIL simulator) in real-time. Any divergence between the Digital Twin and its “master” hardware shall be periodically realigned through virtual communication between them via the internet.

Thinking about its applications, it is also beneficial for a product like this to be as generic as possible. In multi-HIL scenarios (more than two HIL locations), it may be required for a DUT to be replicated in various levels of precision in different locations. For example (Fig 1-9), a power converter of an electric drive needs to be replicated to a high detail at the motor simulator, since performance of a motor vastly depends on the harmonic content of the converter output. Contrarily, the battery management system simulator in a third location does not concern itself with detailed harmonic content of the converter power input and is content with a slow-averaged value.

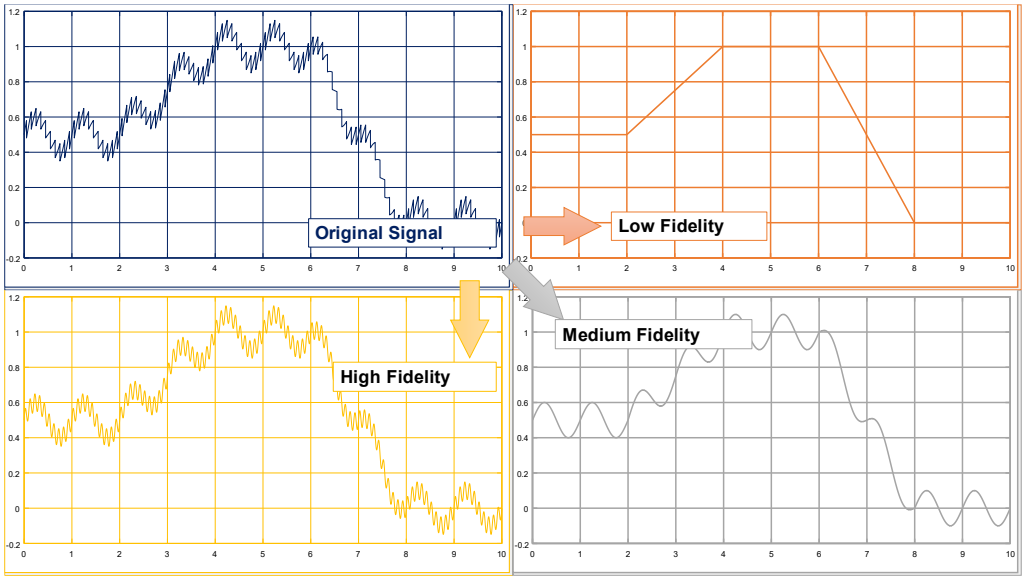


Fig 1-9 Different levels of fidelity of original signal may be required at different centres for their respective application

So, the Digital Twin is required to do the following:

A. Computationally efficient for real-time operation with the simulator
B. Particularly good model accuracy for instantaneous response, i.e., highly accurate model for a narrow operating region
C. Capability to react to external realignment information to shift its operating region when necessary
D. Deployable in varying precision configuration basis the coupling requirement with the respective HIL hardware

The best solution for an ideal Digital Twin was sought that conforms with these four key requirements as stated above. Various popular modeling strategies were investigated from the time v frequency domain spectrum and the best methodology was adapted to suit the special requirements for this application.

Following on from the introduction into the project above, outlined below are the outcomes intended for this PhD project:

- Develop a robust Digital Twin for a generic electric automotive powertrain that can be deployed in multiple locations and kept converged with the master DUT/hardware over various qualities of the virtual communication like the internet. This Digital Twin

must be efficient and quick for real-time operation and consume the minimum computing power as possible by having a customisable “fidelity setting” dependent on the requirement of the local simulator.

- Commission an electric power converter HIL simulator rig to enable HIL testing of an automotive power converter and demonstrate drivecycle runs on the simulator. This would be used as a benchmark to validate the performance of the Digital Twin baseline behaviour, and as the real-time DUT realigning the remotely running Digital Twin, effectively being the second half of the proof of concept for this PhD.
- Undertake a feasibility study of the geographically distributed hardware-in-loop and its viability given the state of the art in power converter modeling and simulation, HIL technology and virtual communications. This is planned to be a collaborative effort with the other partner universities which have similar PhD programs (for distributed HIL virtual testbed) but for other components of a hybrid powertrain, like the engine or battery. Some publications from partner universities are [17][18][19][20][21].

There are **two primary novelty aspects of the Digital Twin** that have been explored and presented in this PhD project – variable frequency operation of a Digital Twin based on the Multi-Frequency Averaging (MFA) technique, and novel approach to multiplication of two waveforms in the frequency domain. As will be shown in the following chapters, MFA is potentially an excellent technique for Digital Twin implementation but all previously published works on MFA-based models explored were found to be only for a fixed frequency operation. Automotive application necessarily requires dynamic speed of operation, and the underlying mathematics was further developed and successfully tested to facilitate variable frequency of operation.

A **third novel contribution** in this PhD is the development and validation of a proxy encoder solution for motor emulation. Control of a physical synchronous motor (primarily used in automotive traction applications) requires highly accurate and precise information of the motor shaft position. Testing an inverter requires this high-speed signal to be generated using low-speed CPU based processor in the simulator. A novel software-based solution was proposed and developed in this work.

A virtual HIL simulation testbed for multiple prototypes operating from different locations for an electric automotive powertrain application has never been envisioned or attempted before. This PhD project has endeavoured to do this in the last couple of years by conceptualising this distributed multi-HIL framework for automotive powertrain testing. This also forms the foundation for the subsequent works done on the development and testing of the electric powertrain Digital Twin.

Intermediate results and progress have been presented to the wider community through the following publications:

- Conference paper in VPPC, Chicago, USA [22]
- Journal paper in IET EST [23]

1.6 Thesis Structure Guideline

CHAPTER 1 provides a broad introduction to the automotive industry and the current disruptive megatrends that will forge the modern electrified car of the future. Key technological challenges are discussed which are needed to be overcome to achieve this. Geographically distributed Hardware-in-the-Loop (GD-HIL) simulation is identified as a key piece of the puzzle and how this PhD project is adding towards the common goal. There is a need for a global system architecture which shall be elaborated in the next two chapters. Key technical elements identified to implement GD-HIL for Power Electronic Converters (PEC) are:

- local emulation of remote hardware (i.e., battery and machine) and
- remote emulation of local hardware (i.e., converter/electric powertrain emulation in the engine location).

CHAPTER 2 provides an extensive literature review of various HIL types and specific technological challenges and enablers. Every type is supported by multiple ongoing projects in the world as case studies. Specific learning outcomes from each case study are identified that are incorporated in the geographically distributed HIL simulation testbed of a generic electrified propulsion system. An overall global system architecture has been proposed towards a reliable and robust “virtually connected hybrid vehicle”.

CHAPTER 3 provides an extensive literature review of different approaches to virtually connect between different HIL centres. Different methods of information exchange between centres are investigated and the Master-Follower approach is found to be the best option for electrified powertrains. An MFA-based Digital Twin is further identified as the ideal method that satisfies the four pre-requisites stated as A, B, C, D in the previous sub-chapter (1.5). Following on, the thesis has been split into two parts: Part 1 – Remote Digital Twin (CHAPTER 4) and Part 2 – Local HIL implementation (CHAPTER 5 and CHAPTER 6).

Especially for electrified propulsion systems, Multi-Frequency Averaging (MFA) methodology has been identified to be the best for real-time computability, remote realignment of parameters and configurable fidelity of output. Variable frequency operation of MFA and a practical implementation of the above three features have been identified as the novelty in this PhD (preliminary work published earlier in IET and IEEE publications [22][23]). **CHAPTER 4** provides the mathematical foundation of the MFA model and builds on it to produce the variable frequency operation framework. The implementation of the same through a MATLAB/Simulink™ model has been discussed. A benchmarking study has been done to verify this model against detailed simulation models of an equivalent system.

CHAPTER 5 provides the methodology for doing a HIL experiment on a generic power electronic converter. Motor emulation is discussed in detail and its implementation on the Triphase simulator rig is explained.

CHAPTER 6 provides the results and findings after drivecycle experiments using the motor emulation strategy developed in the previous chapter were performed on the Triphase simulator.

Chapter 7 concludes the thesis by wrapping all technical findings from the previous chapters together. All shortcomings in the project duration are identified and future work has been proposed.

Each chapter has **Introduction** and **Summary** sub-chapters that gives an executive overview without delving into minute details.

CHAPTER 2 Geographically Distributed Power Hardware-In-The-Loop Simulation

2.1 Introduction

Geographically distributed power hardware-in-the-loop (GD-PHIL) is a complex term loaded with several important distinguishing features, which requires to be deconstructed to understand properly.

Simulation in general, refers to the *imitation* of a real operation or event of a physical system, but without using it. In digital simulation, the imitation occurs in a computer where the physical system in question is replicated in the digital environment using various “digital models” put together in a closed loop. Broadly, there are two categories of models, *analytical* and *empirical*. Analytical models replicate the physics involved in the real system in the form of mathematical equations which are solved at defined periodic intervals or *time-steps*. Empirical or Predictive models use real observations rather than mathematically describable relationships of the system to replicate it. This often requires *training* the model using the real system on a set of operating regions to build a *lookup table*, which is then referred to during simulation. In majority of modern models (thanks to fast computers), some form of hybridisation between the two categories is used.

HIL (hardware-in-the-loop) refers to a simulation scheme where at least one component of the simulated system is a real hardware that is operated alongside and in conjunction with the simulator running a software model. This form of “hybrid” system has some sub-system(s) in physical form and the rest in virtual form, interfaced together by a “HIL simulator” that runs the software sub-system models in real-time and facilitates the interaction between real and virtual by means of sensors and actuators. [24] provides a comprehensive review of HIL technologies and lists various definitions used in the research community.

P-HIL (power-HIL) refers to the specialised branch of HIL simulation wherein the real hardware is an electrical system (as opposed to an electronic system) where significant power flows in and out of it, that is sensed and actuated at the interface to the digital environment. In order to make suitable distinction, the electronic-only HIL scheme is often referred to as **C-HIL** (controller-HIL). Exchange of power may also be in mechanical form, like in the case of an engine-in-the-loop simulator, where a physical engine is operated through a duty cycle with the road load being emulated by an electric dynamometer (dyno) dictated by the virtual road and vehicle dynamics model (this would also fall under P-HIL category).

GD-HIL (geographically distributed HIL) refers to, as the name suggests, a simulation scheme wherein multiple HIL simulators separated by large distances are virtually connected to complete the closed-loop system.

After this brief definition of the various concepts in HIL simulation, they are dissected into greater detail in the following sub-chapters wherein the utility of each type is discussed in a historical perspective and then followed by select case studies.

2.2 Hardware-in-the-Loop Simulation

2.2.1 Introduction

Development of any novel product from conception to completion goes through three well-defined stages which form the widely used “V-model” (Fig 2-1). These stages can be broadly classified into **Definition** (requirements capture and system design), **Implementation** (initial prototype build, real or virtual) and **Validation** (test for functionality fulfilment). The last two stages hold critical importance in the success of a product and is potentially where the most amount of time and money is spent in the cycle. Hence it stands to strong reason that substantial effort is made to reduce expenditure in these stages. Various works describe the use of the V-model in specific applications (controller design [25], generic product development [24], control software development [26]).

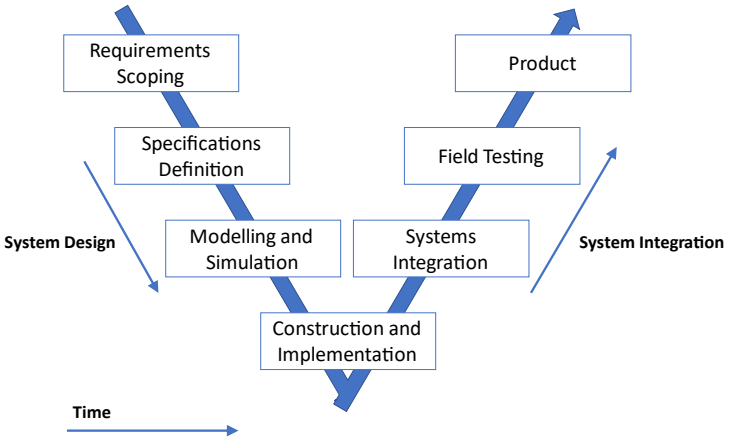


Fig 2-1 System-V approach to development process from concept to product

After the advent of fast digital computers since the 1970s and 1980s, preliminary stages of *prototyping* and *Verification and Validation* (V&V) became increasingly in the virtual domain using software tools. Software simulation is a remarkably effective paradigm where a prototype may be built and operated at practically zero cost which meant the design could be progressively refined after multiple iterations. But it has certain limitations. A virtual prototype of a component is effectively a model and is only as accurate and precise as our knowledge about the associated physics is. There may be some components which are highly *modellable*, but this does not hold true universally. Even if the associated physics is well known it may just be too complex to solve efficiently and in real-time.

Physical prototype development for V&V is essential even today (with modern computers faster than ever before), especially at the later stages of development when minimal design changes are expected, and a proof of concept is valuable. *Reliability and Robustness* (R&R) tests like stress tests and weather durability tests hold significance with physical prototypes only. But this step is associated with significant cost in time and money. Fig 2-2 depicts the trade-off between physical and virtual V&V and shows why a hybrid approach may be immensely beneficial. This gap is filled with the HIL paradigm wherein some components of the virtual system may be replicated in physical form and simulated together, “immersed” in the virtual environment.

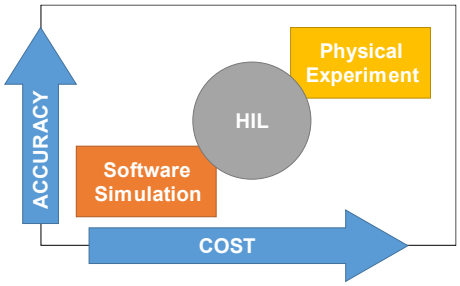


Fig 2-2 Experimental Cost v Accuracy depiction of prototype validation using software, HIL, and physical experimentation

HIL technology has emerged as a creative and powerful tool in the last two decades for V&V with the use of both real and virtual prototypes simulated together in a closed loop virtual environment. There are two ways in which this may work. First, a real prototype *controller* may be subjected to a virtual environment (or *plant*) to evaluate if the controller is functioning properly without having to subject it to the real world. There are several reasons why this is useful. For example, the real application of a prototype may be destructive in nature (e.g., missile guidance system). Another example, a virtual prototype of a novel controller must be evaluated in real world conditions (plant) as an accurate plant model is not feasible. This methodology is often segregated from HIL to be referred to as *Rapid Controller Prototyping* (RCP). [27] is among other works that makes this distinction (Fig 2-3). In this PhD, HIL shall be discussed in its umbrella-term sense.

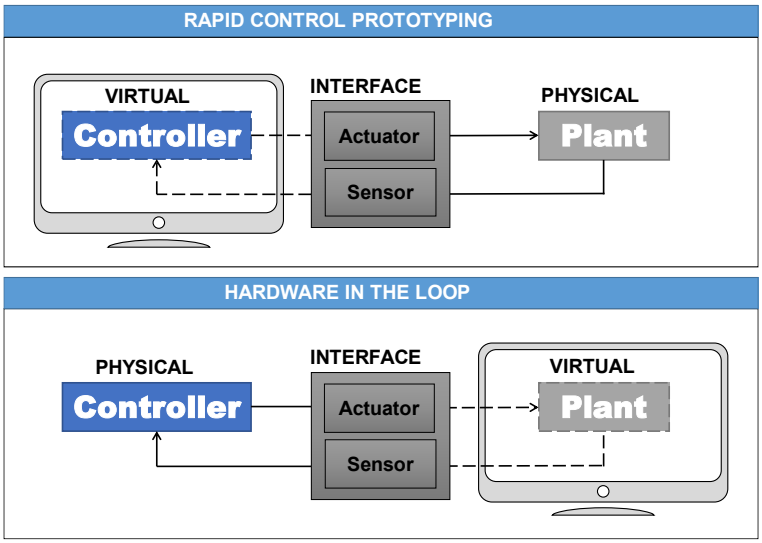


Fig 2-3 Conceptual distinction between Rapid Prototyping (RP) and Hardware-in-the-Loop (HIL)

HIL has grown to become a mainstay in a diverse spectrum of industries like the automotive, aerospace, railway, marine and robotics. It has been argued in the research community that although this new tool is indispensable in the industry, it is not very well standardised [28] and the terminology is not always formalised among various works of research, partly because every project has its own characteristic interpretation of various terms and definitions, and owing to the fact that it's still a relatively new field. *While this PhD does not intend to propose a standard set of definitions, a thorough review of the literature has been done and concepts are presented hereon in a condensed and structured manner.*

2.2.2 A Brief History of HIL Simulation

A form of "flight simulator" with a real pilot sitting inside a gimbal-joint mounted prototype airplane in 1910 [29], also known as the "Sanders Teacher" is believed to be the first instance when HIL was conceptualised in a functional form. A pilot trained on this setup with full control over the plane's control surfaces to regulate the virtual flight against wind. This setup was not effective as the wind was unpredictable. Around the same time a more sophisticated "Antoinette's Barrel" was developed where the plane mock-up (made from a barrel cut in half, see Fig 2-4) was moved by human instructors instead of the wind to allow the pilot to feel the effects of various controls. Analog computers came in the picture during the 1940s followed by digital equivalents in the 1960s. [30][31] list more examples of historic flight simulators (like the "Linktrainer" of 1936 that was a fixed cockpit with all instruments replicated to familiarise the pilot) and their challenges back in the 1960s. Baarspul [29] provides a detailed review of the evolution in this application as newer technologies led to the likes of pneumatic- and hydraulic-actuated flight simulators and digital flight simulators available today. Defence and aerospace were the pioneering industries to make use of HIL in flight and missile control [32][33] like the Sidewinder [34] project in 1972 and NASA's HiMAT [35] in 1984.

Fig 2-4 “Antoinette’s Barrel”, one of the earliest HIL simulator concepts from 1910s [36]

As the technology evolved the costs dropped making these HIL technologies trickle down to the automotive industry from aerospace, although it can be reasoned that the life-critical need of driving simulators in road vehicles is much less compared to aircrafts thus avoiding the necessity until later. The first driving simulators came in 1980s with Daimler-Benz [37] developing a complete in-vehicle simulator (Fig 2-5). A stationery car is housed inside a dome with projections on the walls that emulate real-life traffic conditions. The vehicle was fully instrumented to feed into the virtual environment that alters the 3D visual scenario accordingly. The housing dome also had hydraulic actuators to effect a 6-DOF motion to produce inertial response for the driver. Another notable driving simulator also emerged from Japan along similar lines [38].

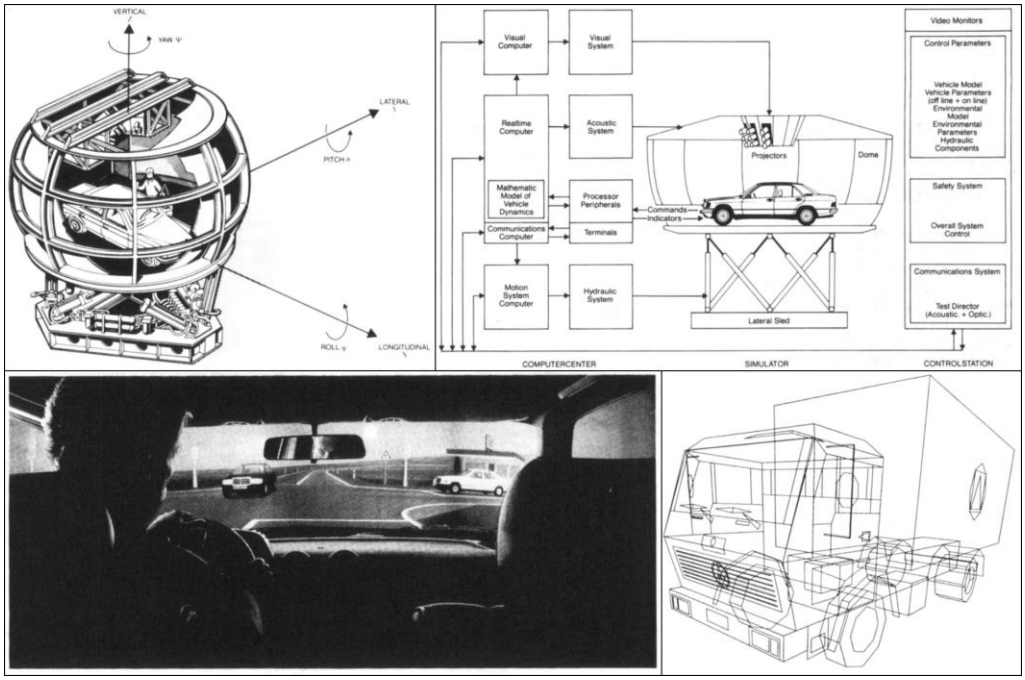


Fig 2-5 One of the first vehicle cabin simulator from the 80s developed by Daimler Benz [37]

Other than driving simulators, HIL also began in various vehicle sub-systems like the brake systems, suspension hydraulics and general powertrain control and vehicle dynamics systems. Anti-lock Braking Systems (ABS) and Traction Control System (TCS) were tested on HIL systems in 1980s through industrial partners like Robert Bosch GmbH[39][40]. The first implementations of the “Engine Dyno” were also seen around this time [41]. This was called the “Dynamic Motor Teststands” wherein a real engine was connected to a generic electric motor controlled by digital model of the transmission system and road load.

The 1990s saw dramatic advancement of microprocessor-based electronic controllers for ABS/TCS, automatic transmission and engine throttle and ignition control. These controllers saw significant development and testing through some form of HIL equipment. Real-time simulation took a significant jump in capabilities with the use of parallel computing (in the form of transputers) and digital signal processors (DSP) in the 1990s which enabled much more complex simulations of larger system like Multibody Systems [42] and Combustion Engines [43]. A later sub-chapter provides a review of computing technologies for HIL technology and much more detail is covered there.

Real-time simulation of electrical and power systems also gained traction in this period albeit for a completely different application: energy grids. The need for these simulations was to increase reliability, efficiency and quality of power generation and transmission. Simulators started off as analog miniaturised replicas of the real power system (called analog Transient Network Analysers, or TNA) which eventually became digital simulators. Use in automotive application is a very new concept and shall be discussed further in a later sub-chapter on Power-HIL. Bouscayrol gives a good review of HIL applications in electric drives for both controller-only and power (electrical and mechanical) types [44].

By the late 1990s, commercially available HIL equipment gained popularity when companies like dSPACE™ GmbH and ETAS (subsidiary of Robert Bosch GmbH) started producing generic and customised tools for controller HIL testing [45]. Opal-RT also emerged during this time with cost-effective solutions using PC-based computing and distributed computing for electrical applications [46]. Before this most examples were in-house developed by companies and hence were only partially published or not at all. This is partly why there is not a lot of historical review papers in the community but [24][27][47] have a comprehensive coverage of most applications that have happened historically.

2.2.3 Deconstruction and Requirements

A HIL simulator may be divided into three key areas (Fig 2-6):

- **Virtual Environment** operates the models and conducts the simulation in real-time.
- **Physical Prototype** are the physical components that close the simulation loop.
- **Interface** comprises of the sensors and actuators to transfer signals between the real and virtual domains.

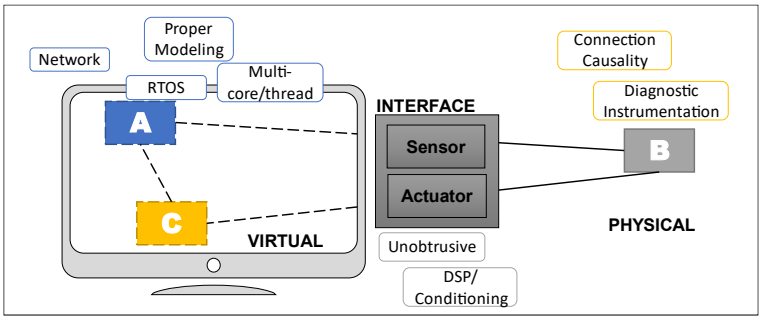


Fig 2-6 Three key areas of HIL simulation – Virtual Environment, Physical Prototype, and Interface [47]

[47] provides a comprehensive breakdown of HIL essentials and the various technology enablers that are required for effective operation of the system.

2.2.3.1 Virtual Environment

- Since physical hardware cannot be made to operate slower or faster than reality, the simulation must keep up with the pace and hence is required to run on a real-time. Not only real-time, but the simulation must also exchange signals at fixed time frames with the hardware owing to the bidirectional nature of the system. This requires a **real-time operating system** (RTOS) that executes the integration at fixed time-steps using clock-driven interrupts. Being limited to **fixed time-stepping** solvers introduces a new problem of discrete/continuous mismatch since a physical device is always continuously operating and may trigger an event in between two time-steps of the simulation. This means a small time-step should be sufficiently small, but not so small that the RTOS is unable solve the *model step* before every *real time step*, and lose the real-time operation.
- The system of virtual models that operate may suffer from a concept called *stiffness*, i.e., components having vastly disparate time constants. This is markedly true for electromechanical systems (EV powertrains) wherein the electrical parts are significantly faster than the mechanical parts. In such case, it is useful for the different components to be solved at different rates using either different *execution streams* of the same processor (**multithreading**) or different cores (**multicore** processing). Modern CPUs have multiple cores, and each core supports multithreading to solve this issue. A significant challenge with parallel operation

within CPU is to synchronise between them and unify the task-level outputs into model system output. Use of multiple processors and advantages of multi-rate solving have been discussed in detail in [48].

- The models operating in the virtual environment often struggle to balance between two opposing forces: fidelity and simplicity. The model needs to be accurate enough to fulfil the design requirements of the experiment and not confuse the physical system by throwing highly erratic and unrealistic conditions and responses. On the other hand, it must be simple enough for the simulator to sustain a real-time operation. Significant work has been done to determine suitable HIL models in automotive mechanical system like the hydraulic clutch [49] but more work is needed for electrical systems.

2.2.3.2 Physical Prototype

- When a HIL simulation is planned in the development process of a system, it is essential to identify which part(s) of the complete system would benefit most from a physical prototype(s) and what should be an ideal *coupling point* between real and virtual to facilitate a synergy between them. Partitioning is a term used in the literature to objectively define the best point to couple the real and virtual [47]. Partitioning refers to identifying the ideal place to split the system that allows direct experimental insight into the part that deserves it the most.
- The physical prototype, like any experiment not just HIL, must be adequately and intelligently instrumented to allow the virtual environment to gain access into various *waveforms of interest* inside the prototype. It is essential that the properties of this **diagnostic instrumentation** such as bandwidth and tolerance are appropriately rated high enough to not hamper the real simulation.

2.2.3.3 Interface

- With every simulation it is endeavoured for it to be as close to real life as possible, and the real part should *feel* like it is really operating in its real-life environment, not a virtual mock-up. Towards that end, the physical/virtual interface composed of sensors and actuators must be of high fidelity, bandwidth and *unobtrusive* by not

altering the dynamics of the system-under-test. It is argued in [50] that a HIL simulator is essentially a *controller* where the virtual environment dictates the hardware to track a hypothetical system reference. It has been well established that unobtrusiveness is key towards efficacy of the simulation results and has been proved in [51].

- Sensor signals picked at the interface often require significant conditioning and digital **signal processing** before it may be fed into the virtual simulation, these tasks may include anti-aliasing, analog to digital conversion and noise attenuation. These secondary tasks are usually done separate from the main HIL processor.

2.2.4 Necessity and New Opportunities

Necessity of HIL-based applications was briefly introduced earlier through a historical perspective. Broadly, advantages of HIL has been condensed into the following [26][47]:

- *First*, when the system prototype (controller + plant) is either destructive, extremely expensive, or imposes risk of loss of human life in the V&V steps. This is especially true for aerospace and defence applications, where prototype aircrafts are expensive to build and require a human pilot to operate, or navigation/guidance system of a missile or rocket.
- *Second*, the presence of highly non-linear and non-modellable components in the developed system. Usually, sensors and actuators fall under this category as they are difficult to accurately model and economical to procure off-the-shelf units. Moreover, these peripherals often have stages of analog/digital communication protocols and interfaces with the main processor, of which there are no straightforward ways to model or simulate. Even if a complex physical process like combustion and emission formation in engines can be modelled, they would run much slower than real-time and often using the real hardware (like the engine on dyno) is an easier option.
- *Third*, this route is often cost effective. HIL is cheaper than a full system prototype and is more true-to-life than pure virtual simulation in performance. An opportunity to move the “process field” into the laboratory comes with great monetary benefits.

As a result, Rapid Prototyping use this technology to quickly build and iteratively test controller prototypes (called Rapid Controller Prototype, or RCP) without a significant cost penalty.

- *Fourth*, the opportunity to have the plant in virtual environment means highly repeatable test cycles can be run as many times as needed. This is especially needed for reliability testing of controller prototypes wherein the device is continuously operated several times in the same cycle until it fails. Unless cycle iterations were not identical test conditions, the exact failure mode is impossible to confidently identify.
- *Fifth*, a much broader range of test conditions may be emulated in the virtual domain for comprehensiveness. Highly improbable yet critical for safety scenarios in real life should be easy to replicate at whim in the virtual world that allows significant freedom of experiments.

By the turn of the 21st century, a gradual shift of focus into systems engineering and optimisation has been observed. This is due to two reasons; individual units have gained technological maturity and companies are relying more on tuning various units of a system to work together symbiotically; and the fact that modern applications are getting more multidisciplinary and complex in nature (electrified powertrains is a classic example). Concurrent Systems Engineering [47] is where HIL outshines and really holds its ground in utility; HIL facility allows different engineering teams or departments to develop various parts of the system without losing sight of integration issues and testing the system at every stage of the development process, thereby “baking-in” the system optimisation objective.

Two new opportunities have emerged with HIL technology very recently – Power HIL (P-HIL) and Geographically Distributed HIL (GD-HIL). Fathy et al have discussed how HIL metamorphosed from “control validation tool” to “system synthesis paradigm” [47]. The idea is to exploit the new capability of exchanging high-power signals across the real-virtual interface, and not just controller-level signals. An entire component like the power converter (controller & plant) may be simulated with the rest of the system.

GD-HIL initially started as multiple HIL systems in the same location connected with a direct communication link. The motivation was to have a scalable architecture to add more computing units to increase the processing power with time. This opened the idea of having multiple HIL systems in distinct locations connected over the internet. These two sub-branches of HIL have been discussed in detail in two subsequent sub-chapters ahead.

2.2.5 Case Studies

[24][26][44] provide a useful reference for a wide spectrum of case studies in the HIL world, not just automotive but aerospace, railways, marine, defence, power systems etc. Two relevant examples shall be discussed here.

2.2.5.1 Case Study 1: Foundational understanding of HIL layout

[52] discusses the development for Powertrain Control Module (PCM – an electronic controller that controls the fuel injection into engine among other secondary and auxiliary tasks) of a car. The prototype controller is connected to a HIL simulator that models the sensing/actuating and the rest of the vehicle (i.e., plant) in virtual. A step-by-step guidance of configuring this HIL application is briefly explained, and two specific issues are examined in detail: virtual models for real-time "HIL-capable utility" and proper partitioning of interface signals.

First, authors identified these HIL-capable virtual models must have:

- Defined for minimal operative dynamics only, as simple, and quick as possible. Each model should be simple algebraic of ordinary differential equations (ODE) strictly avoiding of algebraic loops. Single dimensional lookups may replace analytical models that are too complex to solve in real-time.
- Fixed step size solvers used only, like Euler, Trapezoidal, or Range-Kutta methods.
- Unit and scaling conversion between adjacent blocks and/or models strictly avoided to reduce computational head.
- Overall modeling environment flexibility a key design philosophy to quickly modify the system virtually when needed.

Second issue deals with appropriate partitioning of interface signals between real and virtual. Any "HIL interface" relies on accuracy in two things: conversion of a control signal into physical actuation (e.g., a good "actuator" should produce a steady voltage irrespective of loading), and sensing of physical signal and feeding the information into the model. Defining the real/virtual interface at the appropriate stage is essential to minimise the inaccuracy and delays introduced. There are three primary areas that need to be considered:

- **Transformation of signal** – A control signal from the prototype controller may be communicated through a PWM (pulse width modulated) signal, wherein a slow-changing information is communicated via a high frequency signal. For example, a magnetic solenoid coil (fuel injector) is controlled by a PWM drive signal, where the duty cycle of the PWM dictates the total opening of the valve (Fig 2-7). While the speed at which the solenoid must be controlled is slow (ms), it still requires a much quicker PWM signal to operate with variable valve opening (10s of μs in this example). In order to replace this real solenoid, the HIL simulator must be able to read the controller output signal and translate it to the "equivalent effect" in the virtual system. This transformation calculation is usually best dealt with a separate microprocessor core or standalone DSP unit, just to avoid the main process from getting overloaded.

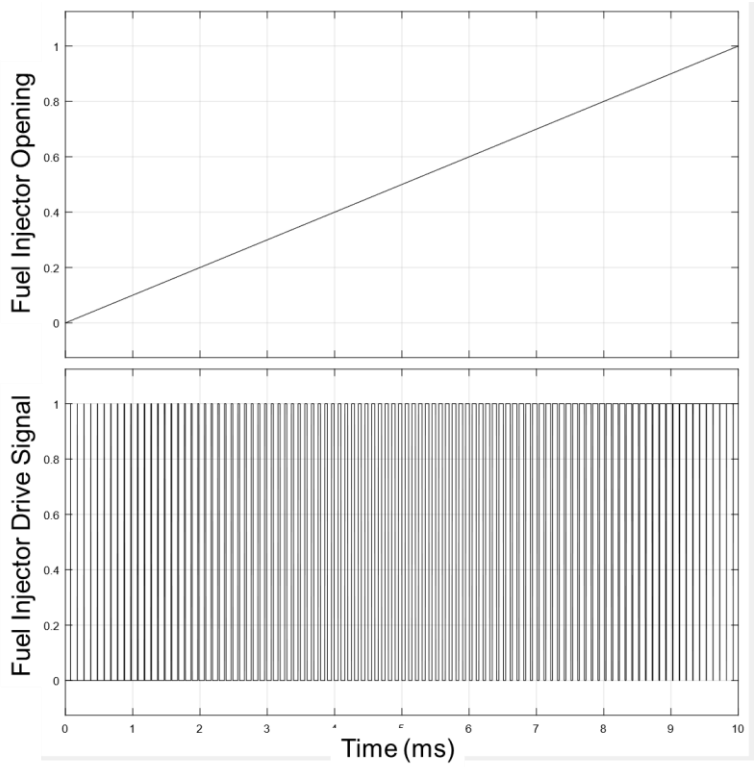


Fig 2-7 PWM-actuated fuel injector solenoid

- **Physical Signal Interfacing** – A real controller unit may have hundreds of analog and/or digital I/O pins that must be provided for by a valid interface by the HIL simulator. For example, a PCM monitors various things like fuel pressure, air intake manifold pressure, vehicle speed, oxygen concentration in air intake, various emissions metrics, engine health metrics etc. All these signals do not necessarily are required for the HIL tests, but they still need to be duplicated (by the HIL itself or an independent peripheral) so that the PCM believes it is in a real system.
- **Fault Mode Simulation** – A real controller unit would have several fault detection protocols in place that may need to be evaluated as a part of the V&V. This experiment may require access to a separate set of signals with different bandwidth requirements, hence it is essential to know what experiments are required before designing a HIL platform.

This work formed the foundation philosophy in laying out a proper HIL setup throughout the project.

2.2.5.2 Case Study 2: Treatment of high-speed signals using low-speed CPU

[25] discusses the challenges and potential solutions of undertaking HIL simulations for power electronics and electric drives. Two of these are:

- Controller triggers PWM gate pulses at an extremely high switching frequency and the HIL simulator solver time-stepping must be significantly smaller (a hundred times smaller advised) to achieve near-trueness. Fig 2-8 shows an apparent issue in sampling. Not just actuating, the same issue is present in sensing as well. Quadrature Encoder to feedback shaft position also is extremely high speed and standard real-time simulation may not cope with it. A feasible solution is to have an intermediary FPGA-based I/O module for all the rapid interfaces that can read or generate pulses at say 10ns steps and compress into information sent or received at a slower rate for the real-time simulator. OPAL-RT™ Timestamp Bridges have been found to be very useful in this application [53].

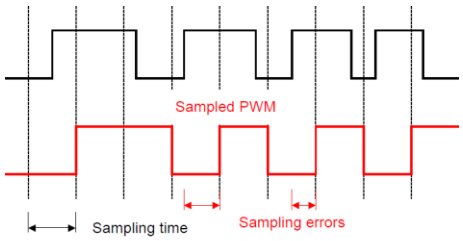


Fig 2-8 Digital sampling of physical signal at a limited/restricted bandwidth introducing sampling error [25]

- Real-time simulation for hybrid powertrains schemes is difficult for the solver because of numerical stiffness of the system. While the mechanical aspects of the powertrain (IC engine, motor shaft) have a time constant of $10\mu s$, the electrical and power-electronics aspects are of $10ns$ scale. [54] compares the fastest time-steps capability of various real-time simulators as shown in Fig 2-9 (although the technology has improved and modern state-of-the-art simulators can simulate faster, modern power electronics have also become much faster with the advent of MOSFETs in wide-bandgap devices). There is a huge gap which poses a problem for the solver operating in fixed-step real-time operation. This novel issue for HIL

applications is best solved by rethinking the strategies for processor core allocation and parallel computing.

Motor model	PMSM-dq (CPU)	PMSM-phase (CPU)	PMSM-JMAG (CPU)	PMSM-phase (FPGA)	PMSM-JMAG (FPGA)
Phase-domain or DQ?	DQ	Phase	Phase	Phase	Phase
Implementation target	CPU	CPU	CPU	FPGA	FPGA
Neutral point voltage	no	yes	no	yes	yes
Can simulate open-phase condition	no	yes	no	yes	yes
Cogging torque and torque ripple	no	no	yes	no	yes
Model update and latency					
Inductance update rate	10 us	13 us	25 us	10 ns	40 us (interpolated on FPGA)
Back-EMF update rate	10 us	13 us	25 us	10 ns	10 ns
Total latency (from switch gate pulse to current output)	20-30 us	26-39 us	50-75 us	1.5 us	1.5 us
Switch pulse capture resolution	10 ns	10ns	10ns	10ns	10ns
Max electric speed (Hz)	400 Hz	400 Hz	100 Hz	400 Hz	400 Hz
Max rotor speed (for 4 pole machine)	12000 RPM	12000 RPM	3000 RPM	12000 RPM	12000 RPM
Max PWM frequency	10-17 kHz	8-13 kHz	4-7 kHz	100 kHz	100 kHz

Fig 2-9 OPAL-RT™ motor drive simulator types and characteristics [54]

There was significant learning from this case study in application to this PhD. Quadrature Encoder input resolution issue was solved in an approach like the OPAL-RT™ Timestamped Bridge approach (although independently thought of and implemented). *Numerical stiffness* was a known concern from the beginning and the modeling approach was modified accordingly.

2.3 Power-HIL Simulation for Electric Powertrain Applications

2.3.1 Introduction

Benefits of HIL as a concept for hybrid simulation of real and virtual elements in a closed loop in V&V process of product development are abundant and have been discussed extensively in the previous sub-chapter. In a closed loop hybrid simulation, it is only natural to progress to physical elements that exchange significant amounts of power across the real-virtual boundary instead of just having control signals. The motivation arises from the fact that many components involving hydraulics and friction dynamics (tire and brakes) are easier to prototype in hardware compared to virtual simulation, as

these virtual models can be very stiff and complex with high error margins. The sensing/actuating and diagnostic instrumentation technologies have improved to a level that bidirectional integration of full-sized real and virtual system (not just the controller) is possible to a significant extent today [47]. This development took place in the past couple of decades and has led to the research community clearly demarcating the “traditional” HIL as Controller-HIL (C-HIL) and this type as Power-HIL (P-HIL).

Other than vehicle dynamics, the need for P-HIL has strongly emerged in electric drive applications for hybrid vehicles. Maximum electric drive applications today have power conversion stage(s) composed on high-speed switching devices like IGBT (Insulated Gate Bipolar Transistor) and MOSFET (Metal Oxide Semiconductor Field Effect Transistor) that are non-linear in nature and fast switching between states. This means, the devices have two completely different sets of properties in two operating conditions, on and off, and they switch between the two states very quickly (tens of kHz for IGBT to tens of MHz for MOSFET). Efficiency and heat loss in these devices depend significantly on the switching dynamics (e.g., frequency of switching, state transitions) and vary with external factors like lifetime, temperature, and vibration. The state of the art in real time simulation cannot cope with modelling of these conversion stages at a switching level which makes a compelling case to have the entire converter in physical form in-loop with the rest of the system, i.e., the powertrain, implemented as a virtual model.

While there is a clear case for undertaking P-HIL experiments for power converters, it is by no means easy and inexpensive (as was the case with vehicle dynamics components mentioned earlier), reasons of which are discussed in the third section (section 2.3.3). This is followed by a review of the P-HIL equipment available today for such applications (OPAL-RT™, dSPACE™, triphase) that includes the rig been used for this PhD study. A set of case studies from the research and industrial community are dissected in the last section with relevance to this PhD. These technical discussions follow from a brief history of the P-HIL scheme in the next section.

2.3.2 A Brief History of Power-HIL Simulation

A brief history into HIL in general cross-industry was given in the previous sub-chapter. In this section, electric power simulation is discussed for various applications. The earliest implementations are from the 1930s when a miniaturised equivalent electric power circuit was developed in physical form and evaluated. These were called Transient Network Analysers (TNA) and were primarily developed to save time which would have otherwise been spent to conduct laborious calculations using the analytical approach [55]. A general TNA comprised of a matrix of various power components like sources, passive elements (resistor, inductor, capacitor), transmission lines, and transformers, which could be wired to replicate a miniature version of the real-world electric system in question, a national-level power grid for example. As these simulators were analog, they were inherently linear in nature and hence could be easily interfaced with prototype controllers to conduct HIL simulations.

Analog simulations have its share of disadvantages [56]. They are expensive to build and maintain (contactor cleaning, component calibration etc.). Every new test setup requires re-wiring the matrix which is associated with appropriate checks to ensure safety in high voltage connections. Moreover, there is a significant probability to warrant the need to upgrade the hardware to fulfil new simulation requirements. All these problems led to H. W. Dommel build the Electromagnetic Transient Program (EMTP) in 1969 that uses a computer to simulate power networks virtually using analytical models in the time domain. This quickly took popularity due to its high flexibility and low cost. As it was all virtual, new components could be simulated by simply writing a new code. With the development of computing technologies, parallel processing gained popularity and real-time simulations were possible. In 1993, Manitoba HVDC Research Centre in Canada developed the first Real-Time Digital Simulator and named it RTDS Technologies™ [56][57]. A large electric grid network is split into small nodes, and each being computed in a dedicated processor, the overall system being fully scalable to append more processor nodes to simulate larger networks.

RTDS gave rise to digital simulation in the power system V&V process since the 90s. P-HIL simulation could be implemented using RTDS by having an appropriate power interface. P-HIL is different from HV component testing in the way that this is bi-directional, a closed-loop system whereas HV component testing equipment like “short circuit generators” to assess circuit breakers and protection relays are relatively simpler implementations and conduct only open-loop tests.

2.3.3 Deconstruction and Requirements

HIL equipment was deconstructed in the previous sub-chapter and this section shall build upon that classification to add further elements to it specific to P-HIL applications.

2.3.3.1 Virtual Environment

The virtual environment is identical for P-HIL applications as it requires a real-time operating system (RTOS) with fixed time-stepping solver. Owing to fixed time-stepping, the problem of numerical stiffness (various components of the virtual system running vastly different time constants) comes into the picture that is solved by proper modelling in software and using multi-core processors with multithreading capabilities in hardware.

2.3.3.2 Physical Prototype

Two concepts or “enablers” were discussed in the previous sub-chapter: System Partitioning and Diagnostic Instrumentation.

The decision of where exactly to break the physical link (**Partitioning**) in the real system to replace with a real-virtual interface is ever more critical for P-HIL applications and requires many considerations to achieve synergy between the prototype hardware and virtual model. In-depth knowledge of the available power sensors and actuators (in the HIL simulator) in terms of their control bandwidth is essential to stay within the technological limits. For example, in Fig 2-10 below (right side), the mechanism of emulating an impedance Z is by measuring the voltage at the input (sensor stage), processing it to find the current that the impedance would have passed through it (computation stage), and then setting the set point for the current source in the emulator

power stage (actuator stage). This three-step process completes the control loop and has an inherent delay which must be considered to find the maximum bandwidth this HIL simulator can be operated at. In case there are dynamics (introduced by the “Real Battery”) faster than this control-loop delay there is a significant risk of having an uncontrolled oscillation which de-stabilises the simulation.

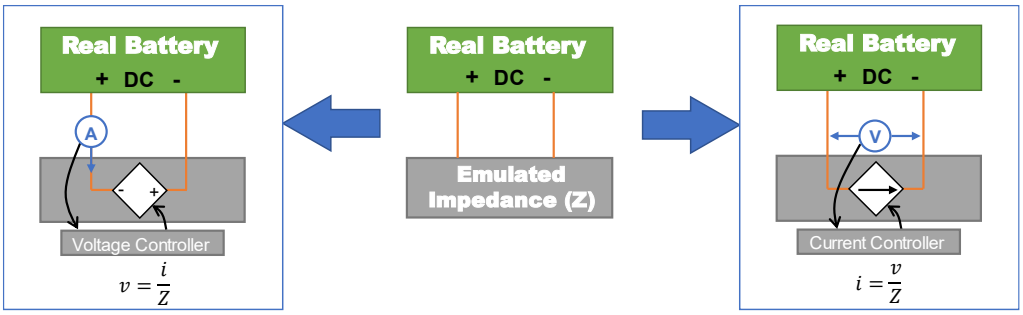


Fig 2-10 Two ways of DC load emulation – voltage-controlled v current-controlled

Another consideration is **System Causality**, i.e., defining the cause-effect relationship at the interface. In Fig 2-10 above, the impedance can be emulated in the HIL via two mechanisms: current source (CS) or voltage source (VS). In case of CS, the voltage across its terminals is measured through high bandwidth sensor and used to calculate the current set point the emulator should target for a value of impedance Z being emulated. Similarly, for VS the current is measured, and an appropriate voltage set point is calculated. This mechanism is explained more elaborately in the next section. This decision is often down to the output filter stage of the emulator: in case of an inductor filter (L or LCL), it is easier to control the current and hence a CS is chosen, for a capacitor filter (C or LC) a VS type is chosen.

2.3.3.3 Interface

The concept of **Unobtrusiveness** was discussed as an enabler for interfacing at the real-virtual boundary, and it is even more critical for P-HIL applications. **Signal Conditioning/Processing** is the second enabler discussed in the previous sub-chapter which also applies for P-HIL applications.

Let us visualise a Battery driving power through a Physical Impedance (Fig 2-10 above). The battery can be visualised as a pressurised tank of electrons which are just waiting

to be released, while the impedance can be visualised as a thin pipe that restricts the flow of electrons as a function of the *electron pressure*. When they are connected, the current starts flowing at the speed of travel of electrons in a conducting material at nearly the speed of light, which for all practical reasons is infinite. That is the physical mechanism of flow of current in an electrical circuit.

In order to "partition" across this physical connection and replace the impedance with an "emulator", the emulator must be able to perform as close as possible to the real impedance. Another important requirement for "digital loads" like this is to have a dynamically variable impedance value, i.e., the impedance value $Z(t)$ could be input as a function of time. This is usually implemented by a VSI (Voltage Source Inverter) in modern applications, which can generate a voltage across itself at a specified bandwidth. The steps of operation of this emulator are as follows [58]:

i. The emulator voltage is set at a default level not too far away from the measured voltage of the Battery (to prevent large in-rush current. Another way is via a pre-charge resistor circuit on the Battery).
ii. The electrical connection relay/contactor is closed, the circuit is completed.
iii. The current through the emulator is measured, with an error margin and measurement bandwidth.
iv. The measured value is communicated to the real-time target (RTT) PC through an I/O peripheral, with an error and delay.
v. The RTT computes the voltage target for the VSI (power stage of emulator) based on measured current and requested impedance. This step has some error and latency.
vi. The target set point is communicated to the VSI through an I/O peripheral, again with an error and delay.
vii. The VSI applies this desired voltage against the battery voltage with an error and bandwidth.

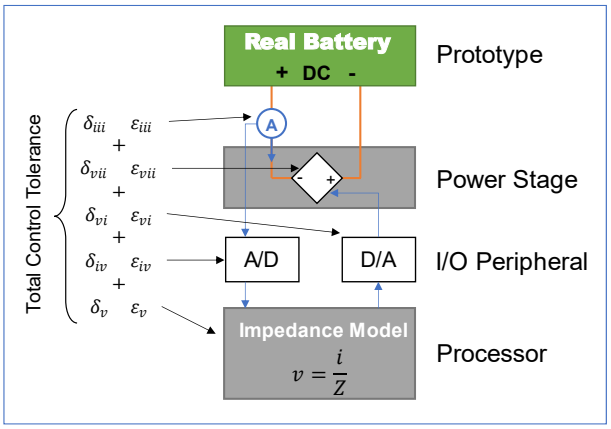


Fig 2-11 Total round-trip-time in load emualtion control breakdown

In this process (Fig 2-11 above), steps iii (steps i and ii are preparatory turn-on process of equipment only) and onwards have some error and delay which accumulate to an overall error and maximum bandwidth of control (in the physical system, this error is zero and bandwidth is infinite). Upon close observation, it becomes clear that a P-HIL operation is simply a control system chasing a set point, and like any control system has a maximum bandwidth and tolerance value! Dropping down to a limited bandwidth of control eventually means the waveforms of interest (i.e., the measured current, desired impedance, requested voltage) at the emulator must be well below this restriction to prevent instabilities.

In a Controller-HIL application, this control-loop bandwidth is already understood and accepted in the actual physical system that the HIL is trying to replicate. Certainly, there is an additional layer of delay and error in the HIL application, but this can simply be left as margin-of-error headroom in the real controller bandwidth availability. In case of P-HIL applications, the real system (physical power link between battery and impedance, for example) operates at infinite bandwidth. [59] provides a very handy guide on various interfacing issues for HIL and P-HIL real-time simulators.

2.3.3.4 Requirements for P-HIL Simulation Environment

OPAL-RT™ is a leading provider of HIL and P-HIL equipment for commercial use and have produced a rule-of-thumb list of considerations in designing a P-HIL rig [60]. A similar guideline document is provided by RTDS Technologies™ (the earliest in commercial HIL simulators business) that covers more technical concepts with

foundational mathematics for the interested reader [61]. The following should be account for:

- Amplifier Absolute Maximum Ratings (Voltage, Current, Power) – These ratings must be set based on the DUT that needs to be simulated. During turn on/off procedures of the DUT, there are voltage/current transient spikes that are seen which must be considered. Often the peak ratings of the simulator amplifier/sensor are significantly above the nominal rating, allowing the simulator to absorb the transients, but it is best to consider a comfortable margin.
- 2 Quadrant v 4 Quadrant Operation – As above, this decision depends on the application and substantially alters the price point of the solution. Motor emulation for automotive applications invariably require sourcing/sinking capability as regeneration of the traction motor is an integral part of powertrain.
- Amplifier and Overall Control Bandwidth – Depending on the application, this decision may significantly alter the total cost as high bandwidth amplifiers may be twice as expensive. Motor emulation to study thermal capacity of inverter may require emulating simple RMS motor model at fundamental frequency. On the other hand, studies relating to inverter controller capability to suppress torque ripples (that may be caused by cogging torque) may require injection of various harmonics over the fundamental. It is imperative to remember a fast-acting amplifier must be supported by a high-bandwidth eco-system like the sensors and RTDS processor. The entire control loop bandwidth must be considered (sensing and I/O latencies and simulation time-step) to avoid an unstable control. Sensors and I/O peripherals have slew-rate and latency that needs to be factored in appropriately.
- Harmonic Addition by Amplifier – Most medium to high power P-HIL applications use amplifiers operating in a switching fashion (on/off quickly on a PWM drive signal) as opposed to linear operation. While switching amplifiers greatly improve power efficiency but they add unwanted harmonics to the output. Adequate filter stages are imperative that may absorb a lot of the harmonics, but some would still remain; these lingering harmonics need to be accounted for mathematically when designing the HIL experiment.

- Sensor Signal Conditioning and Filtering – The sensor stream is prone to noise from various causes beyond control which necessitates the need for applying conditioning stages. These stages add on to the overall delay of the control loop. The best trade-off needs to be attained between measurement accuracy and bandwidth.
- Protection Systems and Instrumentation – Even if the emulation requirement is only for slower systems like low-speed hydroelectricity motors or thermal transients, the converter DUT would invariably have several transients that must be protected against by active and automatic control systems in place. Usually an FPGA-based protection board is a part of the simulator power stage that can quickly cut or de-rate the output and runs in parallel and independently from the main emulator program.

2.3.4 Available Equipment and Solutions

All available P-HIL simulator equipment today invariably have (Fig 2-12):

- **Power Stage** – This is a VSI that is powered off a DC bus (this DC bus may be directly taken from the infrastructure or have a Rectification Stage after direct grid supply). Depending on the power level and highest switching frequencies required for the design application, the switching device may be IGBT or MOSFET. This VSI has its own controller that is controlled by the Real Time Target (RTT).
- **Filter Stage** – As the inverters today invariable operate on PWM operation, the output of the VSI is not a smooth and stable voltage curve, it is a quickly changing voltage between V_{DC+}, V_{DC-} and 0. A filter stage is required to stabilise the output. Different kinds of filters may be used, and most equipment have the option to switch between different options on the fly using a relay network. With an inductive filter (L or LCL) it is easier to control the current output of the complete power emulator and with a capacitative filter (C or LC) it is easier to control the voltage output.
- **Sensors** – There are voltage and current sensors at the outputs of the power stage and filter stage.
- **Real Time Target** – This is the system/computer that operates the entire simulator using a Real-Time Operating System (RTOS). Usually, it uses a fibre optic cable to communicate with the VSI controllers.

- **Workstation** – This is a standard PC that acts as the correspondence unit for the entire simulator. The simulator models are visually built and test (offline) in this machine and uploaded to the RTT after appropriate code generation. This machine also provides visual feedback of operation of the simulator and various output sensors.

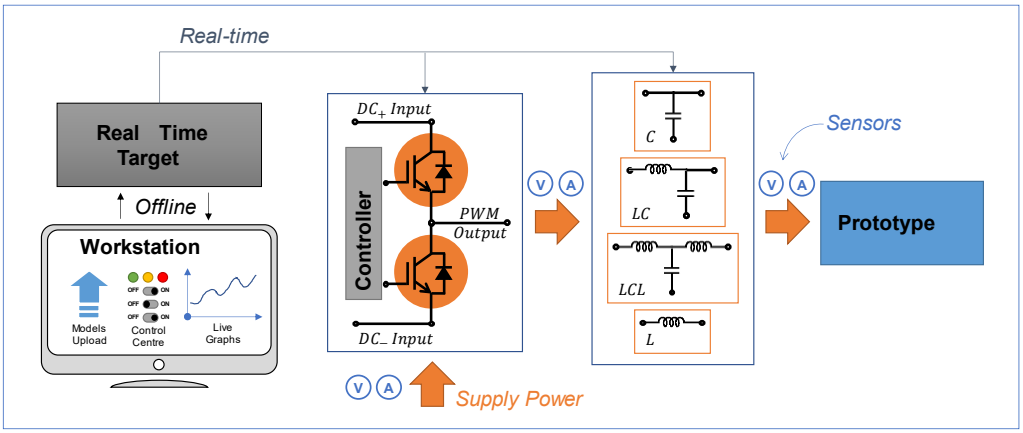


Fig 2-12 Overall layout of a generic power-HIL simulator rig

[62] provides a helpful comparison of most Real-Time Digital Simulators (RTDS) available commercially today and includes a brief history of evolution to what it is today. Various processors technologies (DSP, FPGA, CPU) and their evolution since the past three decades is discussed. Most modern commercial solutions today are based on general-purpose PC-based processors since they are cost-effective and can be parallelised indefinitely to increase processing power to solve more complex models. Often these simulators make use of FPGA-based cards that work alongside the main CPU (real-time step size is in the order of µs for CPU) to perform high-speed auxiliary functions that require time constants of the order of few ns, like encoder emulation and high-speed I/O.

Three popular commercial solutions (OPAL-RT™, dSPACE™, triphase™) are discussed in further detail below.

2.3.4.1 OPAL-RT™

OPAL-RT™ is one of the pioneers of the RTDS technology for HIL and P-HIL applications and an early adopter of the hybrid processor-FPGA simulator hardware.

Their latest offering for power electronics emulation for automotive applications is called the eMEGASIM™ (Fig 2-13 from [62]) that uses a general-purpose CPU in conjunction with FPGA module to tackle different parts of the system model [63].

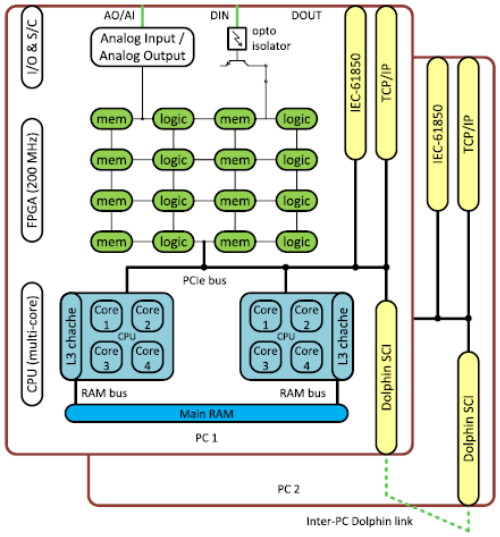


Fig 2-13 OPAL-RT™ power-HIL simulator rig control overview [62]

This configuration is especially useful for hybrid automotive applications since these systems are especially very “stiff” (a wide range of time constants ranging from ns for power electronic switching devices to ms for mechanical and thermal sub-systems) and benefit significantly from isolating the sub-systems and solving them independently using dedicated hardware. A power electronic converter switching at 20kHz (50 µs) would ideally require a time step of 500 ns (1% of switching period) to account for accurate ripple reproduction during steady-state and harmonic analysis [63]. The software package offered is called RT-LAB which uses two support software: Simulink/Simscape™ by MathWorks to programs models for the CPU cores [64] and Xilinx System Generator (XSG) to program the FPGA cores [65].

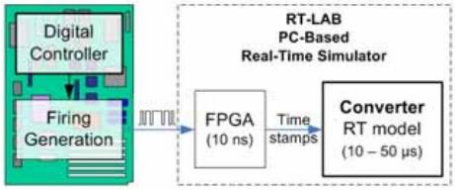


Fig 2-14 Example use of FPGA "expansion" board in HIL simulator setup to transform between high-bandwidth and low bandwidth signals [25]

[66] gives a comprehensive review of CPU v HIL hardware configurations for simulations. CPU is great for flexible modelling using graphical interface software packages like MATLAB/Simulink™ which supports automatic code generation, but suffers from slow time steps (well above 1 µs) due to significant transaction time between the processor core and the PCI information exchange bus (all I/O peripherals communicate with the processor core via the PCI bus). This is where FPGA excels as all I/O peripherals are directly connected with the FPGA core allowing rapid simulation with high bandwidth of the order of 100-500 ns. The downside of FPGA is that it is difficult to program and often requires simplistic solvers for efficient operation.

The use of FPGA expansion board for auxiliary tasks is also an option with OPAL-RT™ systems. The use of Time Stamp Bridges (TSB) has been explained in [25] wherein an FPGA captures PWM signals (at 10 ns intervals) and sends time-stamped information to the CPU-based core at an appropriate slower rate (Fig 2-14).

2.3.4.2 dSPACE™

dSPACE™ developed in Germany around the same time as OPAL-RT™ in Canada (mid 90s) and offers CPU-based RTDS hardware, although their first HIL units were DSP based. Their current HIL and P-HIL solution is called the SCALEXIO™ that offer 25 µs simulation time stepping capability using their Simulink™-compatible solver software although there is the added option to add FPGA add-on cards to simulate faster transients at 400 ns [67]. The most powerful processors from the Intel Xeon family supporting up to 8 cores in a single CPU are used for state-of-the-art simulation speeds and bandwidths. The real strength of dSPACE™ solutions come from the scalability to tie multiple SCLAEXIO units together using their in-house developed IOCNET™ local networking framework that allows up to one hundred nodes spaced 100 m apart (Fig 2-15).

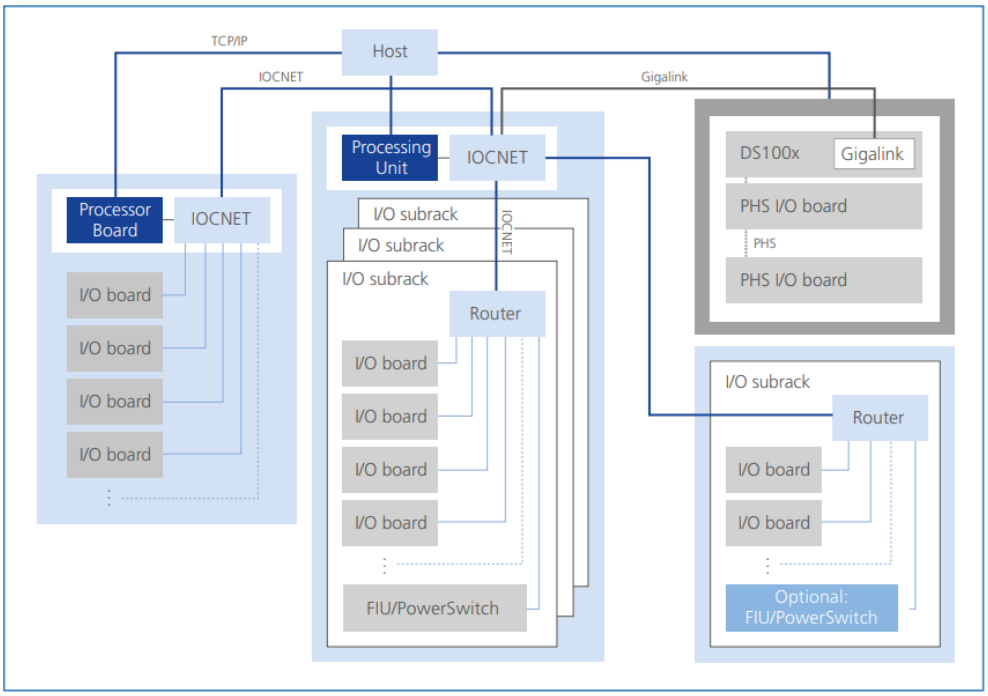


Fig 2-15 Scalability achievable using the SCALEXIO™ architecture available from dSPACE™ [67]

For HIL applications, there are portable hardware blocks like the AutoBox™ (for in-vehicle applications for track activity) and LabBox™ (for bench-top applications) in addition to customisable and scalable rack-mount options. dSPACE™ specialises in Controller-HIL applications primarily and offers application specific integration of the RTDS hardware with power amplifiers and sensors to build upon a P-HIL equipment. Nonetheless, they offer controllable power inverters (800V and 75A nominal) that can be custom-rack mounted and multiple of them could be put together for a common P-HIL rig (Fig 2-16) [68]. There are many use cases published on the company website: Full sized battery emulation at cell level for BMS V&V [69], power converter testing for traction motors [68], HIL tests for DC/DC converters [70].

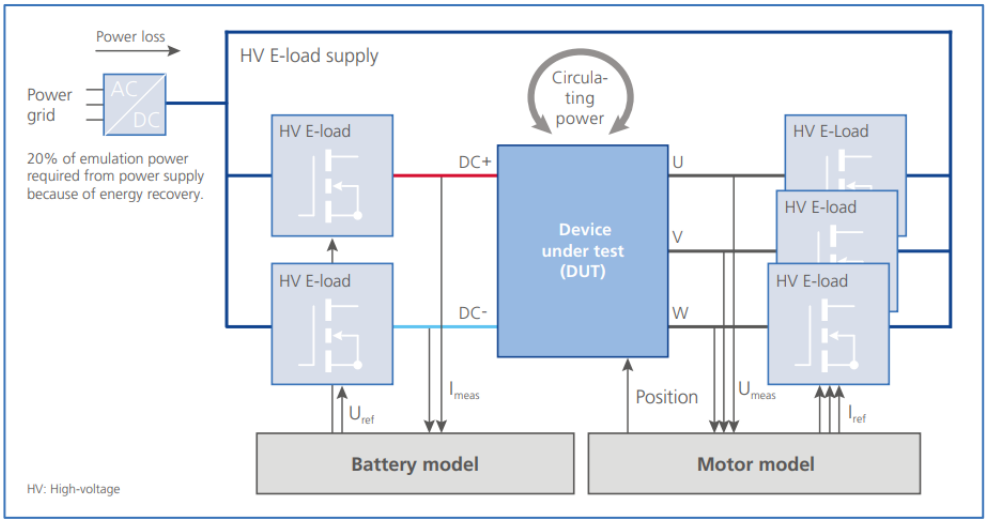


Fig 2-16 Typical power-HIL simulator layout from dSPACE™ [68]

A useful technical paper written by dSPACE™ authors [71] demonstrates the capabilities for conducting all three levels of HIL simulation: signal level (Controller-HIL), power level (Power-HIL) and mechanical level (wherein P-HIL goes one step further to control a mechanical actuator system like a physical motor). This paper provides numerous practical things to consider when designing a P-HIL simulator system.

2.3.4.3 Triphase™ (Subsidiary of National Instruments)

Triphase specialises in high power HIL applications and provides a self-contained unit for conducting multiple P-HIL simulations simultaneously. They provide two power level categories: 15kVA and 90kVA units named the PM15 and PM90. A commissioned rig connects directly to 3-phase grid supply via an active-front-end (AFE) called the A30, optionally via a transformer (I30) to provide galvanic isolation from the grid. The AFE charges up the common DC-Link that is shared between all the PM modules in the system. The back end may be composed of several PM modules like the F03, F30, M30 (see Fig 2-17 below) that uses the energy stored in the DC-Link to convert to the desired electrical energy form, i.e., unipolar/bipolar DC voltage or current (F03), single-phase or 3-phase voltage or current (F30/M30).

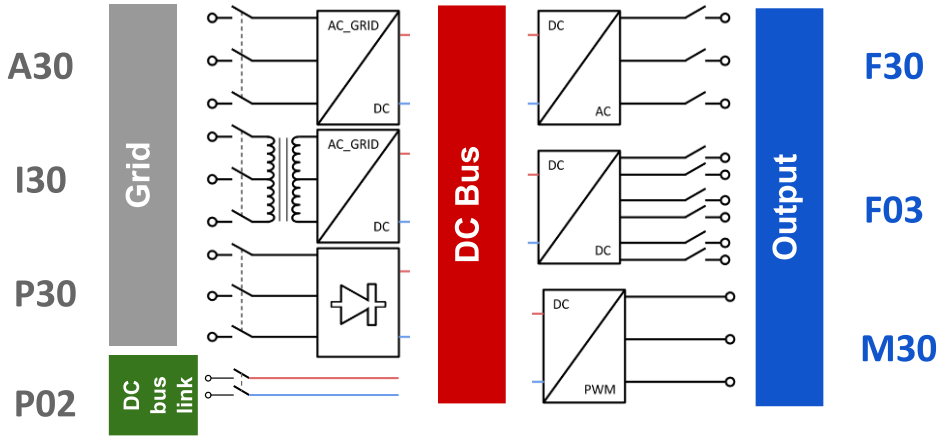


Fig 2-17 Generic block-style "power stages" offered by Triphase to “build-your-own” power-HIL simulator rig

The specific advantage that triphase systems have over competitors is the open data and software architecture provided via fully open-source MATLAB/Simulink™ models. The user has the option to control with as little (via the web-based user interface) or as much detail (via the MATLAB/Simulink™ models). Extra layers of control and data manipulation can be added into the model, which is subsequently built into C-code (via the Simulink™ Builder) and uploaded into the RTT (real time target) which is a multi-core CPU-based PC running an RTOS (real-time Operating System).

The triphase system offers unparalleled control and customisation over the hardware and software equally. The triphase philosophy relies on the fact that at the end, every PM unit is essentially a voltage source converter (VSC) followed by a filter stage; so, by allowing the user to modify the hardware (and corresponding modifications in the software), the back-end PM units could be reconfigured to work as a voltage or current source, DC, or AC. Owing to these advantages, this was the chosen equipment for developing this PhD work.

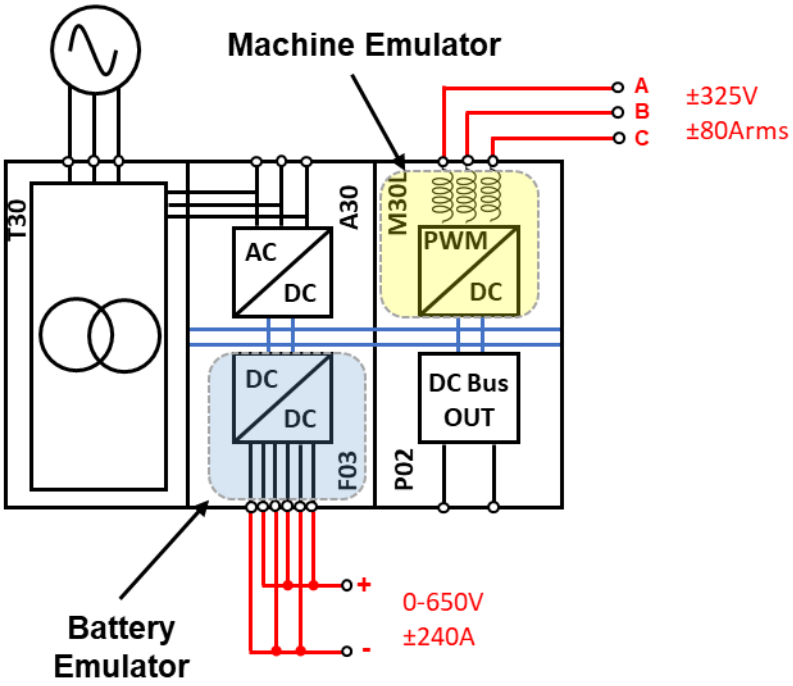


Fig 2-18 Triphase power-HIL simulator rig as configured and procured from Triphase for this PhD

A three-cabinet equipment was procured from triphase as shown in Fig 2-18 above. First cabinet on the left houses the isolation transformer (T30), second cabinet houses two PM units, A30 (AFE) and F03 (unipolar DC voltage source, but can be optionally reconfigured to F30, 3-phase AC voltage source), third cabinet houses M30L (3-phase current source with inductor only filter, to emulate a motor) and P02 (direct access to DC-Link).

2.3.5 Case Studies

2.3.5.1 Case Study 1: Appropriate Interface Algorithm

[58] presents pioneering work in formalising various interfacing algorithms at the simulator-DUT power interface based on two earlier works [72][73]. A simple linear electric circuit is chosen (Fig 2-19) that is required to be replicated in part-simulator and part-hardware. Two implementations are possible using voltage or current amplification in the hardware process.

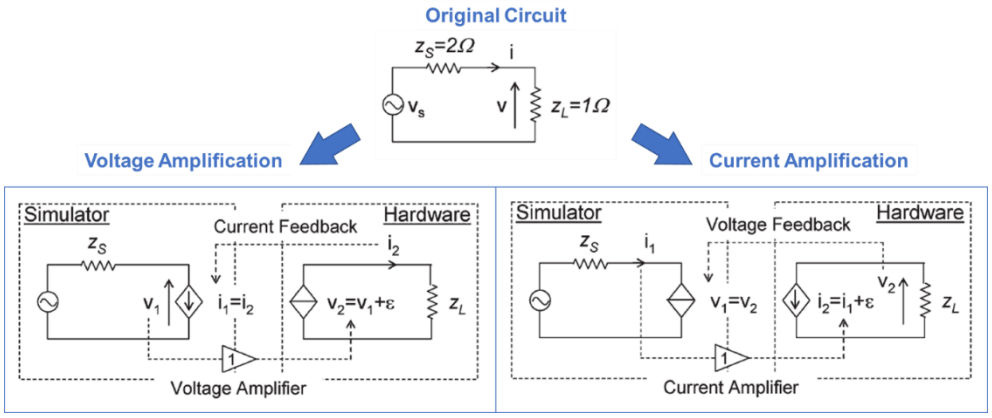


Fig 2-19 Replication of simple linear electric circuit in power-HIL simulator setup using the Ideal Transformer Method (ITM) [58]

In the case of voltage amplification, it is reasoned that the voltage emulation would have certain error to the amount of ε. As a result, the current measured would have an error of ε/Z_L that in turn causes an error of $-\left(Z_S/Z_L\right)\varepsilon$ in the next computed value of emulated voltage. By sending this new value to be amplified, the error also gets amplified, and the control loop enters an unstable oscillation if $\left(Z_S/Z_L\right) > 1$. On the flipside, if current amplification is opted for, the condition for unstable oscillation is $\left(Z_S/Z_L\right) < 1$.

The above elaborates the necessity to choose the appropriate interface algorithm (IA). Five IAs have been elucidated in this paper [58] which are:

- **Ideal Transformer Model** (ITM) – The above implementation is the ITM type and is the simplest solution.
- **Time-Variant First Order Approximation** (TFA) – This is an older method assuming that the circuit can be approximated to a first order linear equivalent, i.e., a simple RC or RL circuit. There are many drawbacks of this IA (mathematically unstable being the primary one) and is not a popular approach.
- **Transmission Line Model** (TLM) – This IA (Fig 2-20) is used when the two halves of the original circuit have a long transmission line in the middle in the form of an inductance or capacitance, which is usually the case in power grid applications. This IA is highly stable but suffers from the fact that the linking inductance/capacitance is

replaced by a resistor that consumes power. Depending on the value of the equivalent resistor, significant power may be parasitically lost from the capability of the emulator.

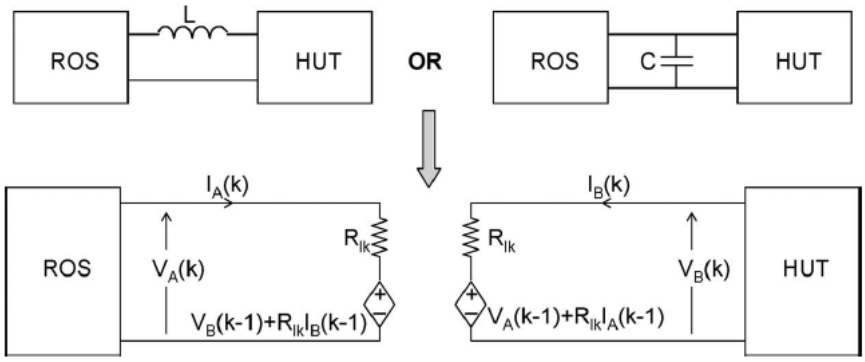


Fig 2-20 Power-HIL simulation approach using the Transmission Line Model (TLM) [58]

- **Partial Circuit Duplication** (PCD) – The total source ($Z_S = Z_a + Z_{ab}$) and load ($Z_L = Z_b + Z_{ab}$) impedance is split and the common impedance Z_{ab} is repeated in the virtual and real worlds (Fig 2-21). For high values of Z_{ab} compared with either Z_a or Z_b the control loop is very stable with a low overall error. The same problem remains as TLM in that significant power is lost in Z_{ab}.

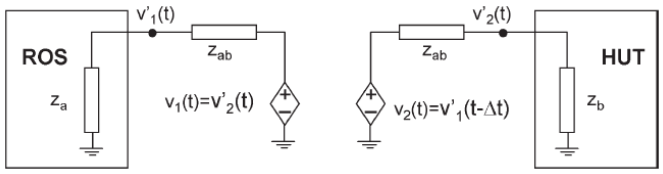


Fig 2-21 Power-HIL simulation approach using the Partial Circuit Duplication (PCD) [58]

- **Damping Impedance Method** (DIM) – This is an amalgamation of the PCD and ITM IAs and tries to achieve the best of both worlds (Fig 2-22).

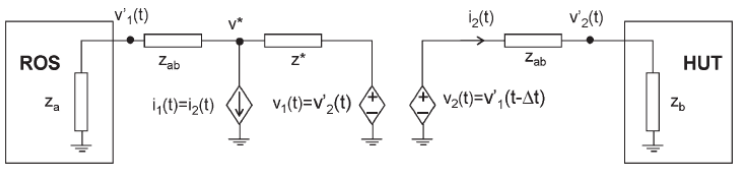


Fig 2-22 Power-HIL simulation approach using the Damping Impedance Method (DIM) [58]

This paper has demonstrated the Open-Loop Gains and Accuracy for each of these IAs mathematically. In a subsequent chapter on the PhD methodology a similar exercise has been done for the power interface for our particular use case for automotive HIL simulation.

2.3.5.2 Case Study 2: P-HIL Interface Accuracy Evaluation

Wei Ren in their PhD thesis [56] has presented detailed work on characterising the "transparency" of a P-HIL interface in any generic experimental setup. All sources of errors (leading to drop in "interface transparency") have been lumped into two categories (Fig 2-23): **Transfer Function Perturbation** (TFP) and **Noise Perturbation** (NP).

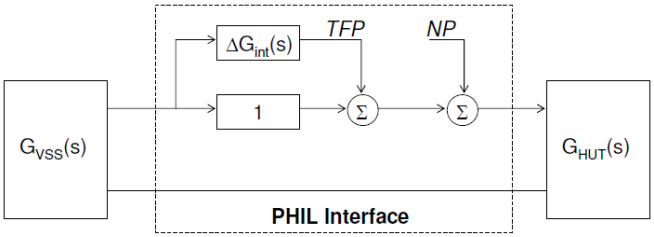


Fig 2-23 P-HIL interface "transparency" characterisation as observed in [56]

TFP refers to all the inaccuracies caused by deviations from unity gain of the power amplifier. A non-unity transfer function adds noise proportional to the system responses of the Virtual System Simulation (VSS) and examples of this includes signal transmission delays or the low-pass filter effect of the interface. NP refers to the external noise elements induced into the simulation control loop and that includes sensor noise and PWM switching noise injected by the amplifier. This noise is independent of the system response.

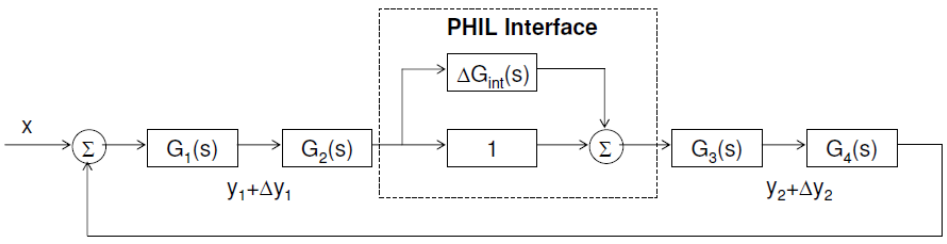


Fig 2-24 P-HIL interface "transparency" characterisation experimental setup in [56]

An idealised generic P-HIL system was drawn (Fig 2-24) and the TFP error was derived for two outputs of interest: output of $G_1(s)$ and $G_3(s)$ blocks. Here the two blocks on the left side denote the virtual system and the two blocks on the right denote the real system, separated by the interface. An ideal interface has a transfer function of 1 (as shown) but an extra "perturbation" component $\Delta G_{int}(s)$ has been added to depict non-ideality. y_1 and y_2 are system outputs and Δy_1 and Δy_2 are the perturbations. The TFP error index has been defined as the ratio of perturbation and ideal signal. A weighting function W_0 may be multiplied with this ratio to treat errors in different frequency regions differently since the different accuracy levels may be required for these frequency bands.

$$E_{TFP\_y_1} = W_0 \frac{\Delta y_1}{y_1} = \left| W_0 \frac{G_1 G_2 G_3 G_4 \Delta G_{int}}{1 - G_1 G_2 G_3 G_4 (1 + \Delta G_{int})} \right|$$

$$E_{TFP\_y_2} = W_0 \frac{\Delta y_2}{y_2} = \left| W_0 \frac{\Delta G_{int}}{1 - G_1 G_2 G_3 G_4 (1 + \Delta G_{int})} \right|$$

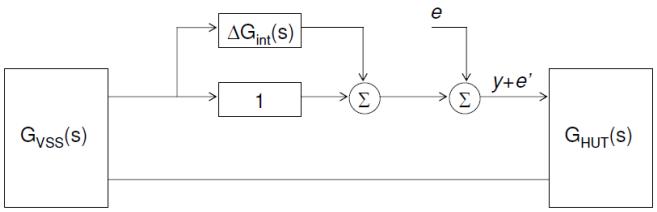


Fig 2-25 P-HIL interface "transparency" characterisation "noise" injection in [56]

NP was calculated by adding an extra noise signal to the interface path (Fig 2-25). The added external noise e lead to an additional e' in the system output. The NP error index was defined as the ratio of e' and e. This ratio may be multiplied by the weighting function W_0 like previously. Additionally, a scaling function W_I must be multiplied to normalise the input error in proportion to the system output y. For example, most sensors specify a maximum sensor noise as a percentage of the nominal rating, which can be used to define W_I.

$$E_{NP} = \left| W_O \frac{W_I}{1 - G_1 G_2 G_3 G_4 (1_{\Delta G_{int}})} \right|$$

Following the mathematical derivations, a useful guide is provided to characterise any P-HIL simulator experiment based on device characteristics provided in the manufacturer datasheets of the amplifiers and sensors being used in the interface. Useful examples for linear and non-linear P-HIL cases were examined. These definitions were used to characterise the 5 IAs explained in the previous case study and compare them in terms of accuracy, stability, and ease of implementation.

2.3.5.3 Case Study 3: Motor Load Emulation

[74][75] from 1998 is one of the earliest pioneering work on development of a “virtual machine” intended for testing a prototype inverter by allowing it to drive this “virtual machine” independently. The virtual machine is considered as an active impedance capable of bidirectional power flow that is implemented through two back-to-back inverters connected to the grid supply with a DC Link in between the inverters (Fig 2-26, CB = Converter Bridge, MM = Motor Model).

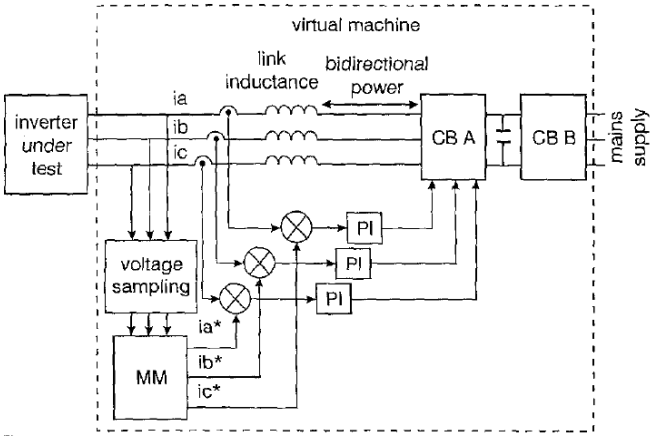


Fig 2-26 Overview layout of motor emulation using two back-to-back converter bridges (CB) [74][75]

The objective of this virtual model is simple, read input voltages and draw appropriate currents. While this is easy in theory, practical implementation gets tricky mainly because high speed PWM voltage signal must be read and processed quickly to produce the current draw setpoints for the current controller.

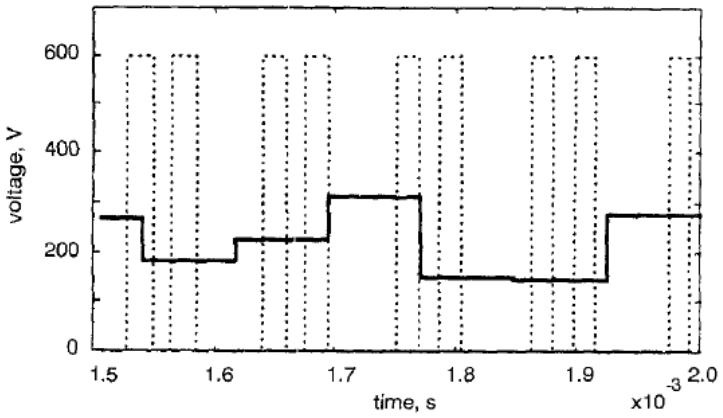


Fig 2-27 Novel method to read high-frequency voltage PWM waveforms using low-frequency processor [74][75]

In this study, the test inverter has a switching frequency of 9 kHz (111 μs) and the "virtual machine" control speed must be at least as fast and hence is set at 13 kHz (77 μs). The voltage signals are read via an "integrate and sample" mechanism wherein the voltage values are constantly integrated and reset every 77 μs (Fig 2-27). The integrated voltage values are then used in the subsequent time step to compute the desired current draw. Then input current is read every step (no integrate and sample mechanism necessary since the current waveform is smooth unlike PWM) and compared with this target current and the error value is fed into a PI controller that controls the "virtual machine" inverter.

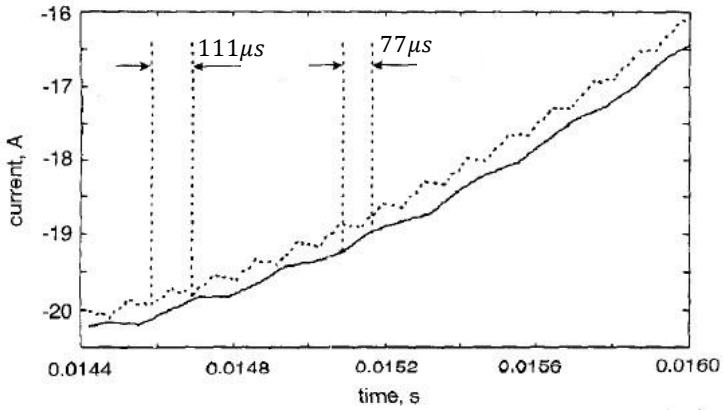


Fig 2-28 Current waveform Effect of using the "integrate-and-sample" method for measuring high-frequency waveforms using low-frequency processor [74][75]

Two things are observed as deviations from the actual physical system (Fig 2-28): the "integrate and sample" mechanism has an inherent low-pass filter action that

dramatically smoothens the application voltage, and due to the inherent nature of processing system, there is a single time-step of delay between the read voltage value and applied current value. Both these problems may be mitigated significantly by increasing the computation speeds dramatically. The RTDS systems are much faster today with FPGA processors in place.

The choice of the link inductor in the “virtual machine” is also crucial. The sizing has two contradictory driving forces: a small value is preferable to have faster control of the transients generated by the machine emulator inverter. On the other hand, a large value although makes the control sluggish, it significantly removes the PWM ripples generated by the machine emulator inverter because after all, the machine emulator is a standard inverter with PWM voltage output whereas a real machine must have no ripples (there would be ripples generated by the stator as back-emf by means of how the rotor is designed and its saliency, but the frequency is much slower). A good balance in the middle should be struck that is close to the actual stator inductance and any deviation may be rectified by having a virtual inductor in the motor model.

This experiment emulates an induction machine which is inherently an open-loop control system, which makes things significantly less complex. For synchronous machine emulation, highly accurate shaft position feedback is required that needs to be emulated as well.

2.4 HIL Simulation Across Geography

2.4.1 Introduction

Distributed HIL simulation refers to all HIL simulation schemes that involves two (or more) HIL simulators locations tied together via relevant communications network engaging in a single closed-loop system simulation in real-time. Depending on the distance between locations, they may be tied together by a dedicated network like local LAN for different buildings of the same enterprise, or general internet connection for locations spread out over larger geography, also known as *Internet-Distributed HIL* (ID-HIL).

[76] mentions four key benefits of parallel/distributed simulation:

- *Reduced execution time* – Dividing the overall process into multiple parallel processes can be computed simultaneously across multiple computers. We have touched upon this topic earlier under multi-core CPUs.
- *Integrating simulators from different manufacturers* – Simulators operating on legacy platforms may be tied together under a central supervisory computer for a cost-effective solution compared to porting the programs onto a common platform.
- *Fault tolerance* – Having the option of a fallback processor in case the primary processor fails adds redundancy to the simulation. This is particularly useful when premature termination of simulation may result in expensive hardware failure.
- *Geographical distribution* – Physical co-location may be avoided by creating a virtual environment for multiple simulators to engage simultaneously. This is the bulk of our discussion hereon.

There are numerous benefits of GD-HIL that have driven tremendous progress in the past decade. R&D is often conducted in different research centres that may be in separate buildings, cities, or even continents! Considering the electric automotive industry which is very horizontally integrated, and supply-chain oriented, different interdependent components of a single system (e.g., battery, motor, engine) are usually developed and manufactured by different suppliers and technology providers. Without GD-HIL, various R&D centres would develop their respective products in an islanded fashion with only offline collaboration, eventually bringing the prototypes together physically for online system HIL simulation. This may need to be an iterative process for fine tuning that proves to be very costly due to the coordination and planning involved.

Two recent megatrends in the automotive industry further support the utility of GD-HIL. *Powertrain electrification and digitalisation* has brought together various technologies from different areas together tightly interdependent in the system. This was not the case a few decades ago; a standard IC engine was wholly developed under a common roof (usually by the OEM) with the rest of the drivetrain (transmission and gearbox) very

closely developed, various control units (fuel injection, firing, gear shifts) were simple. This is complemented by *concurrent engineering* wherein different components of the system (e.g., motor, battery, control) and different processes (e.g., design, testing, manufacturing) are undertaken simultaneously and everything is optimised to work together from the very start of the product development.

The extra layer of networked communications certainly comes with technical challenges. GD-HIL requires online simulation, in the sense that in addition to running all the HIL centres in real-time, they also need to be synchronised with each other. The HIL centres must be initialised in a steady-state t=0 condition and should start simultaneously. During run time, all local simulations must be synchronously tied together via a global network that also keeps divergence in check and exchanges the interface variables without missing/dropping packets. While this is easy in deterministic local networks (like dedicated enterprise LAN in between buildings of the same facility), communicating over the internet is vastly more difficult since internet is non-deterministic and is riddled with issues like latency (a fixed average delay between packet sent and received in destination) and jitter (a stochastic error component in the fixed delay amount). Series of packets sent over the internet are delayed, re-ordered, and dropped upon reaching the destination that need to be carefully and smartly accounted for in real-time with minimal computation overhead.

This sub-chapter only looks at three key case studies that gives a holistic view into the state-of-the-art in this relatively new concept. The “Deconstruction and Requirements” sections in the previous sub-chapters have covered all the essential concepts that are also applicable to geographically distributed HIL applications.

2.4.2 Case Studies

2.4.2.1 Case Study 1: Soldier/Hardware-in-the-Loop

One of the earliest attempts at real GD-HIL was undertaken during 2005-2007 by the US Army TARDEC (Tank Automotive Research, Development and Engineering Centre) as a military exercise to produce real-life duty cycles of a military combat vehicle towards developing electric hybrid propulsion technologies for the next generation

“Humvee” [77]. A “soldier-in-the-loop” 6-DOF crewstation (housing a driver and gun operator) conducted immersive synthetic battlefield simulations mimicking a realistic mission with several miles of drive in off-terrain conditions combating enemy ambushes along the way [78]. Three experiments were conducted of which the latter two involved a remote online connection (called the *Long Haul* or *Remote Link* in this work) with the powertrain simulator located 2450 miles away [79][80]. This is a true application of GD-HIL wherein the *real driver* and *vehicle/terrain simulation* (location 1) and the real powertrain HIL rig (location 2) are engaging in real-time co-simulation (Fig 2-29 and Fig 2-30).

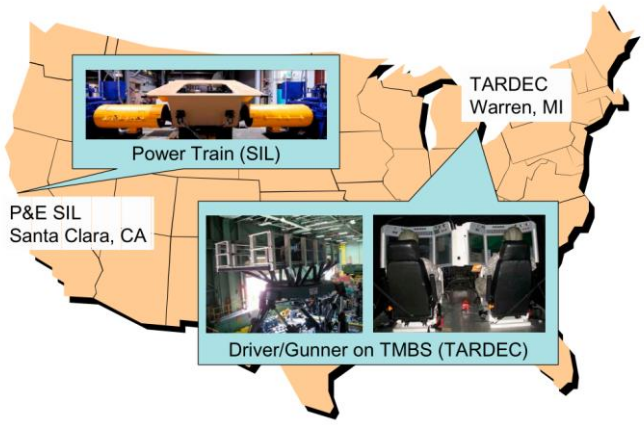


Fig 2-29 Geographically distributed simulation application in the US military with two locations co-simulated [79]-[81]

[79] provides a set of goals and constraints for the *Remote Link*:

- Realistic driving/gunning experience without abrupt jerky motion
- Realistic power system response closely following the powertrain HIL current state
- Driver/gunner generated loads reflected on real powertrain state
- Safety cannot be compromised in the manned driving simulator
- Closed-loop stability of dynamical systems in both HIL centres must not be compromised
- Geographical distance creates unavoidable latency, jitter, and data loss
- Physical constraints of both HIL equipment dictate overall simulation limits

First decision to be made was the choice of communication channel. A dedicated 56 kbps modem was compared with non-dedicated connection over internet; internet channel proved better in all aspects (loss rate and round-trip-time or RTT). Transport protocols for internet communication (UDP v TCP) were compared; while TCP provides data reliability (dropped packets are re-transmitted automatically), UDP was much faster. Since the simulation transmission rate (~30ms) was much smaller than RTT (~90ms), by the time a dropped packet is re-transmitted by TCP, it would be too late already and a new packet needs transmitting, hence making the case for UDP.

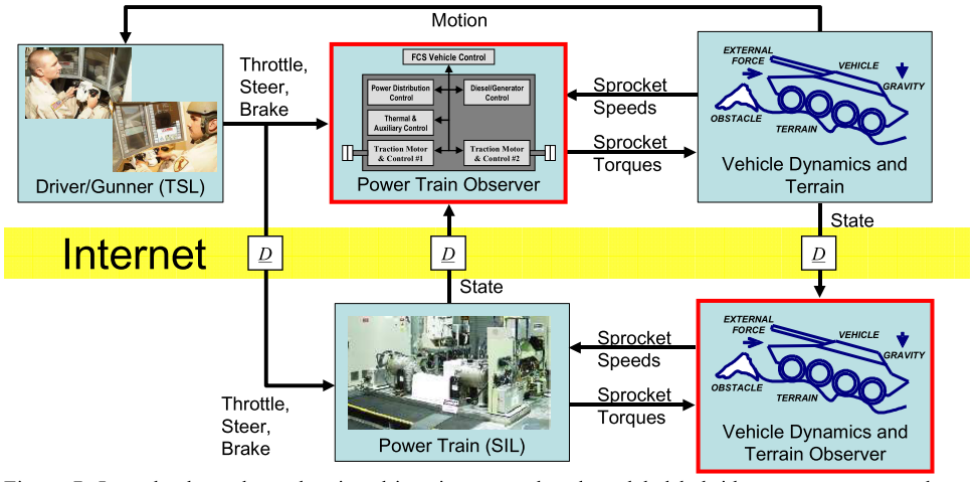


Fig 2-30 Geographically distributed simulation application in the US military system architecture [79]-[81]

To keep the local HIL closed-loop systems stable and completely isolated from internet-related issues, it was clear that a local *observer* model of the remote hardware must only be interfaced with the local hardware so that all local reactions are instantaneous and bounded. These local observer models must follow the real hardware as tightly as allowed by the internet connection, which is called *state convergence*. A good state convergence algorithm endeavours to keep the selected state variables in the observer model close to the real system. There are two observer models whose states need to be converged: *Powertrain* and *Dynamics* models (Fig 2-31).

State Convergence of Observer Models

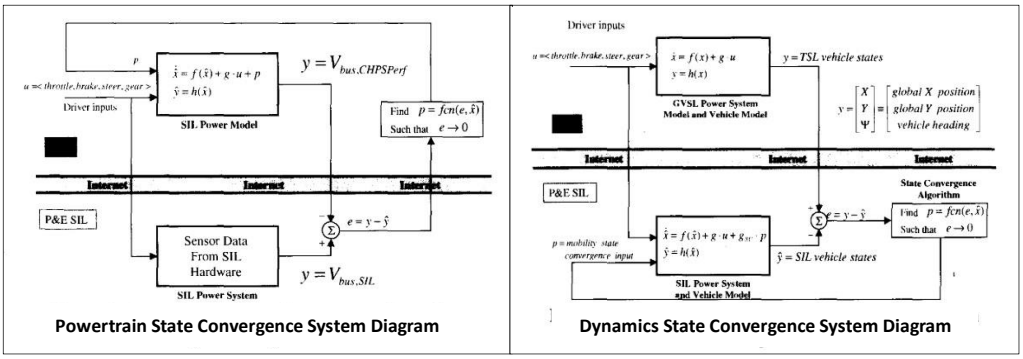


Fig 2-31 State Convergence Model for the Powertrain and Dynamics control systems in US Military GD-HIL testbed [79]-[81]

Both observer models were represented as standard state space models with an additional state convergence control input p that was computed locally to minimize the error e. The choice of p for the observer models depended on how quickly and smoothly the control action is required. For the *dynamics* observer, a skyhook lateral acceleration, a skyhook yaw angular acceleration, and driver's augmented throttle action were chosen to converge the observer outputs global XY coordinates and yaw angle. For the *powertrain* observer, an artificial current drive was injected into the DC Bus to converge its value. A metric called *Leakage Energy* was calculated post-simulation by integrating the external control power supplied into the plant models (product of mass, skyhook accelerations and velocities for the *Dynamics* model and product of external current and DC Bus voltage for the *Powertrain* model) over time. This value after normalisation was <2% for both observers which indicates that the models themselves were mostly accurate, and only needed external control amounting to <2% over the complete simulation run for state convergence.

Effect of Internet Delay

Error value e was required to be calculated for both observers that was minimised to produce the convergence vector p. In case of Dynamics observer, the user input $u_{dynamics}$ (throttle, brake, steering, gear), desired system output $\hat{y}_{dynamics}$ (global XY coordinates and yaw angle) reached it after a single-trip delay of Δ. This means the observer and the real powertrain closed-loop was running behind the soldier-in-the-loop

simulator by Δ. As a result, $e_{dynamics}$ was also calculated in this delayed timeline and hence was independent of Δ.

This is not the case for *Powertrain* observer. The model output $y_{powertrain}$ (DC Bus voltage) was sent via the same UDP packet from soldier-in-the-loop simulator location (as the driver inputs and *Dynamics* model output) to the real powertrain location. $e_{powertrain}$ was calculated as the difference between $y_{powertrain}$ and actual DC Bus Voltage measured, but in the delayed timeline at the powertrain location (delayed by Δ). This error value is sent back to soldier-in-the-loop simulator location that suffers another Δ delay, thus effecting a total chronological mismatch of 2Δ. Hence, the state convergence control is affected by internet delay.

This case study was the cornerstone work that was referred to when developing the GD-HIL architecture for this PhD. A lot of subsequent work was done in the 2010s with this as its foundation to characterise the effect of internet delays [82], effect of coupling point selection [83] and various state convergence control strategies [84][85][81]; they were vital in pinpointing the exact strategy needed for this PhD.

2.4.2.2 Case Study 2: Virtual Test Bed (VTB) and its Real Time (-RT) Extension

Most of the early HIL activities from the early 2000s were either undertaken behind locked doors of industry or were based off of commercial platforms that were unaffordable and/or inflexible [86]. To address these concerns, a low-cost platform/language-agnostic specific to power electronics control applications (C-HIL) was developed in the University of South Carolina in 2001 called the Virtual Test Bed (VTB) [87]. A real-time extension was also developed subsequently that offered a hard real-time extension called the VTB-RT [88].

The system (Fig 2-32) was based on the Linux OS due to its open-source availability, good performance and real-time support (Microsoft Windows hardware abstraction layer makes it inefficient to achieve hard real-time operation, although the GUI features of VTB, like schematic editor and data visualiser were implemented in the Windows environment). An add-on freeware package (RTAI – Real-Time Application Interface, now called *Xenomai*) was installed to enable hard real-time capability. Data Acquisition

drivers were installed via another freeware called Comedi that allowed a list of off-shelf DAQ interface “cards” to be installed in the OS, for controller signal IO with the physical system. Finally, the VTB-RT application was compiled and downloaded into the PC along with the plant model [88]. One of the novel aspects of VTB-RT was the solver used; the Signal extension Resistive Companion (SRC) solver denotes every component in the circuit with an *across* and a *through* variable (i.e., voltage and current). This allows “natural” coupling which means that the wire connection between any two components is bi-directional and allows power transfer to take place. This is imperative in a HIL system where actual power transfer takes place at the interface [86].

After several successful C-HIL experiments, the VTB-RT was also subjected to a P-HIL simulation [89]. The signal exchange interface was appended with A/D sensors and D/A actuators to perform as the power stage for real system. Custom FPGA modules programmed using MATLAB/Simulink™ Xilinx System Generator (XSG) tool were used to read/generate fast physical signals like PWM and motor speed encoder. FPGA modules greatly increased the bandwidth capability of the VTB-RT simulator.

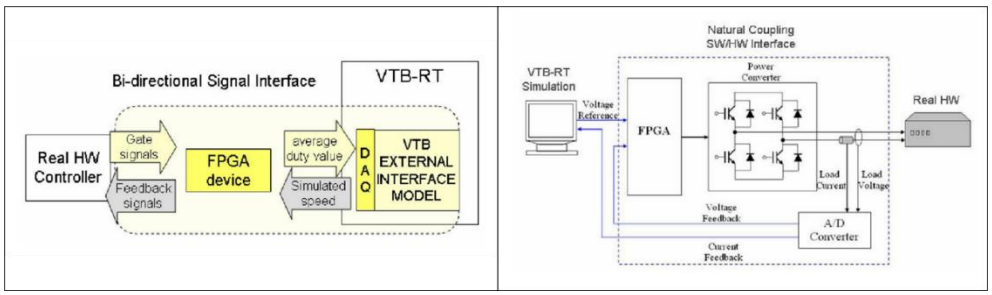


Fig 2-32 Virtual Test Bed - Real Time (VTB-RT) GD-HIL architecture [87]- [90]

VTB-RT was eventually expanded into GD-HIL capability by incorporating a GPS clock synchroniser FPGA module that produced an external clock signal for the real-time simulator based on the GPS clock pulse [91]. A simple proof-of-concept experiment (Fig 2-33) with a buck/boost converter (simulator-1) driving a DC motor (simulator-2) was conducted with two separate VTB-RT units synchronised with GPS clocks [90].

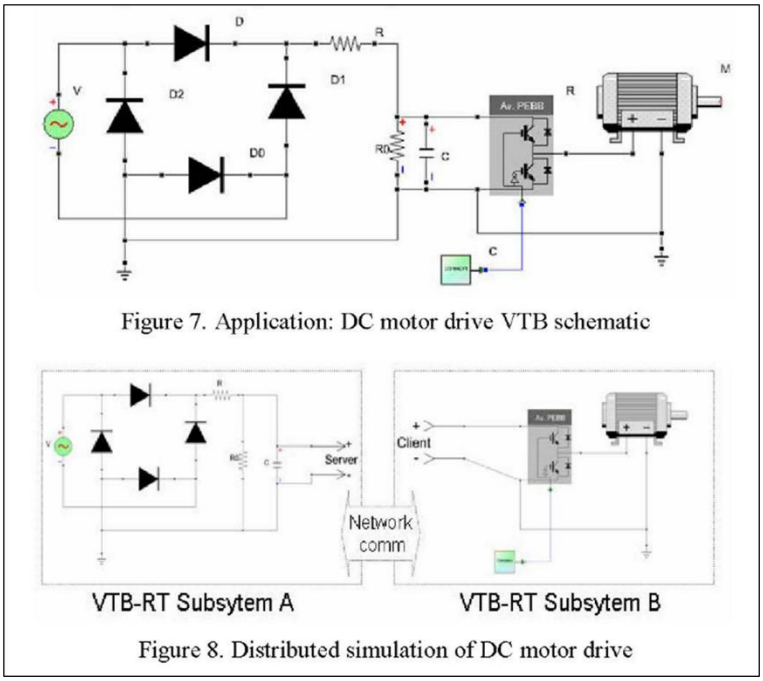


Fig 2-33 Virtual Test Bed - Real Time (VTB-RT) GD-HIL typical example experiment [87]- [90]

This project was the first of its kind in the way that the original motivation was to make this technology open-access. As a result, the literature covered several technical aspects of this project that has helped guide this PhD project.

2.4.2.3 Case Study 3: Global RT Superlab

This project started in the mid-2010s to virtually connect power systems simulation labs across vast distances to engage in meaningful real-time co-simulations for smart energy grid applications [92]. Modern and future energy grids have significant renewable energy penetration with presence of regional microgrids and households shifting from consumer-only to “prosumers” with PV panels and electric vehicles enabled with vehicle-to-grid (V2G). As a result of this highly integrated and distributed “smart” control in the energy grid, there is significant motivation for a project like this.

The main destabilising factor in a GD-P-HIL simulation is the variable indeterministic delay offered by the internet. The best method identified to mitigate this problem was to exchange *Dynamic Phasor* (DP) representation of the interface variables, instead of the direct time-domain variables (Fig 2-34). DPs are nothing but the time-varying Fourier components of the signal. This shift to time-frequency domain from purely time-domain for interface variables by dropping the bandwidth requirements of internet since a high

frequency signal is broken down into several slow-varying amplitudes of fundamental and harmonic frequencies. Moreover, various locations could use different simulation time-steps as required and do not need to account for re-sampling errors (as would have been the case for direct time-domain signals).

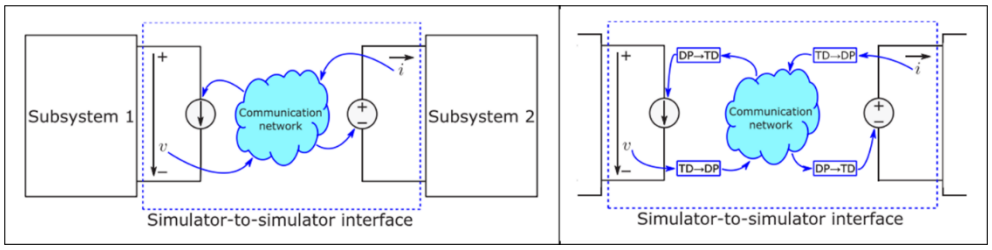


Fig 2-34 Use of frequency domain representations of interfacing variables in the Global Real-Time Superlab project [92]- [93]

The project experimented with decoupling two AC systems at the HVDC transmission link coupling point, and connected two simulators in Germany and USA [92]. The Ideal Transformer Method (ITM) is used to replace the de-coupled joint as shown in Fig 2-35 below. Various artificial time delays were injected to observe its effect on the convergence of the interface variable at the two locations. It was observed that time-domain and DP interface cases showed comparable results when there was negligible delay, with time-domain case deteriorating enormously with increasing delaying effects closer to realistic values (~130 ms RTT for Germany-USA).

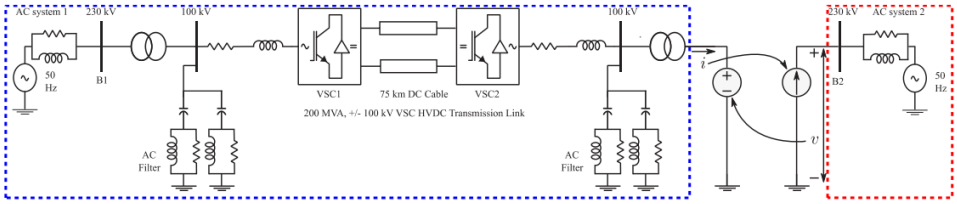


Fig 2-35 ITM coupling method used in the Global RT Superlab project [92]- [93]

Subsequent works tested similar use cases by linking Germany-Norway (~54 ms RTT) [94], Germany-Italy [95], Germany-Italy-Netherlands [96] using DPs, all showing promising results. In 2018, the project conducted experiments with 8 sites all engaged in a co-simulation [93]. The simulated energy grid included PV-emulating node, wind

turbine emulating node, multiple distribution grids and a transatlantic HVDC transmission link (Fig 2-36).

Fig 2-36 Globally distributed HIL experimental testbed offered in the Global RT Superlab project [92]- [93]

This project was the first to use frequency-domain representation of the interfacing variables (Dynamic Phasors – this concept is discussed in the subsequent chapters) to reduce the effect of indeterministic nature of an unstable communication medium like the internet. This fact has helped formulate the project plan for this PhD.

2.5 Summary

This chapter covered three kinds of HIL approaches in depth. **Controller-HIL (C-HIL)** refers to the simplest and oldest form where a physical prototype of an electronic controller-under-test is interfaced with a HIL equipment that simulates the rest of the system. There is exchange of only low power signals across the physical-virtual boundary. **Power-HIL (P-HIL)** refers to the next logical step from C-HIL wherein physical power devices are prototyped and simulated in a virtual environment. The P-HIL equipment allows for exchange of power signals across the physical-virtual boundary, and the rest of the system is simulated in a computer environment. **Geographically Distributed HIL (GD-HIL)** is the most recent type of HIL that is seeing widespread adoption with improving quality of telecommunications across geography. This could be C-HIL or P-HIL, but the distinguishing feature is that multiple physical prototypes are not co-located but undergo co-simulation in real-time.

HIL was deconstructed into smaller technical elements and classed under three categories: virtual environment, physical prototype, and interface. Specific requirements of successful C-HIL, P-HIL and GD-HIL setup were summarized under these three categories for better understanding.

Various case studies were discussed under each type of HIL. An example of Powertrain Control Module (PCM) for fuel-injection was discussed. Various challenges for conducting a C-HIL experiment like transformation of physical/virtual signal at interface, and fault mode simulation were discussed. A second example specific to PECs followed wherein the challenges of emulating extremely high speed PWM signals by microcontroller-based simulators was discussed and how an intermediate FPGA-based emulator is suitable. Numerical stiffness in simulating the rest of the system was also discussed which is a problem for PEC systems due to the wide variation in sub-system time constants.

Main components of P-HIL simulator and four popular P-HIL simulator suppliers (OPAL-RT™, RTDS™, dSPACE™ and triphase) and their offerings were briefly discussed. Four case studies related to P-HIL were discussed. Five different Interface Algorithm (IA) at the physical/virtual boundary and their specific applications were discussed. Another case investigated the accuracy of the IA. Third case investigated motor load emulation and the creation of the *virtual machine*. Lastly, a case for synchronous machine was investigated which is different from induction machine (in the third case study) in the way that it requires accurate shaft position information fed back to the inverter-under-test.

One of the earliest examples of GD-HIL was conducted by the US Military in the mid-2000s and was discussed as a case study. TCP and UDP were discussed as contenders for remote link and various State Convergence algorithms were investigated. Effects of delay in internet and selection of coupling point were also discussed. The next case study was the VTB and its real-time extension from University of South Carolina, wherein a DC Motor drive was progressively C-HIL, P-HIL and GD-HIL simulated. Last case study is an ongoing research project based in Europe called

the Global RT Superlab which endeavours to develop a GD-HIL test facility for grid applications.

This chapter provided a comprehensive vision into the world of diverse types of HIL applications. Technical takeaways from this review chapter that helped build the foundation for this PhD work are:

- A suitable global architecture was identified that satisfies the requirements for a geographically distributed multiple P-HIL electrified automotive powertrain testbed.
- Various challenges of HIL-simulating PWM signals in PECs and the necessity to have a form of FPGA-based intermediate stage in the physical/virtual interface.
- Numerical stiffness in full powertrain system simulation and how to overcome the challenge.
- Various advantages and disadvantages in P-HIL rigs offered by different suppliers.
- Various advantages and disadvantages of different Interface Algorithms specifically for P-HIL simulation of PECs.
- Motor emulation for inverter-under-test and synchronous machine shaft position emulation.
- Internet communication via TCP v UDP investigated and UDP emerges as the best alternative especially for real-time applications.

CHAPTER 3 Power Electronics and Drives Modelling Techniques

3.1 Introduction

In the previous chapter, various forms of Hardware-in-the-Loop (HIL) simulation, namely, Controller-HIL (C-HIL), Power-HIL (P-HIL), Geographically-Distributed-HIL (GD-HIL) were discussed in detail. Several technical limitations for conducting GD-P-HIL experiments were uncovered after a thorough review of the state-of-the-art and takeaways from various case studies were used as steppingstones towards formulating a solid structure for this work. This chapter delves deeper and investigates different techniques for keeping the HIL simulators (separated by a significant distance) converged by minimising the delaying effects of internet. It was noted that a standalone HIL system is essentially a closed-loop *dynamical* control system split into the virtual and physical worlds separated by an interface boundary, which has its own characteristic delaying effects. It is well known that any dynamical system loses stability when delays are introduced. Eliminating a steady delay amount is simpler than an indeterministic delay amount, characteristic of the internet. This chapter discusses various strategies to eliminate these effects and propose the best methodology to do so.

3.2 Different Approaches for Inter-Location Communication

3.2.1 Introduction

Consider a simple two-part electrical system that must be GD-P-HIL simulated: *a voltage source driving an impedance*. Prototypes of the two power components are not co-located but must be co-simulated over a virtual environment. There are two stages

of abstraction: **local P-HIL interface** and **global network interface**. The end goal is for the two prototypes to feel as if they are co-located.

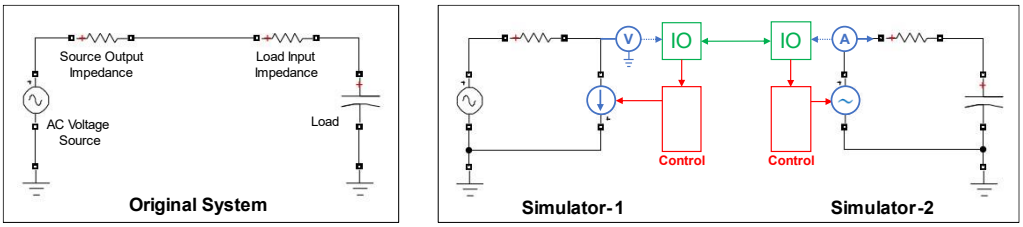


Fig 3-1 Generic electrical circuit split into two sub-systems using the ITM interface algorithm

In the emulated system (Fig 3-1), the **blue components** are the P-HIL actuators and sensors which have a certain delay and bandwidth. The controller (**red components**) also has a limited bandwidth owing to various numerical solver restrictions. They constitute the local sources of non-ideality. The **green components** are for external communications and constitute the global source of non-ideality. The Ideal Transformer Method (ITM) Interface Algorithm (IA) has been demonstrated here, which was explored in brevity in the previous chapter.

3.2.2 Direct Exchange of Interface Variables

The simplest mode of implementation would be to directly exchange the V-I values between the simulators. This would involve the following steps (text is colour-coded according to Fig 3-1 above):

- AC **Voltage** Source (VS) energises the circuit by applying voltage across Controlled Current Source (CCS)
- Sensor measures voltage applied by the VS in Simulator-1
- IO block sends value to simulator-2 over internet/other network
- Simulator-2 controller requests Controlled VS (CVS) to emulate
- Simulator-2 power stage emulates VS in Simulator-1
- This energises the circuit which generates a current
- Sensor measures current through CVS in Simulator-2
- IO block sends value to simulator-1 over internet/other network
- Simulator-1 controller requests CCS to emulate

- Simulator-1 power stage emulates CS in Simulator-2
- Emulated CS reacts to voltage applied by VS thus closing the loop

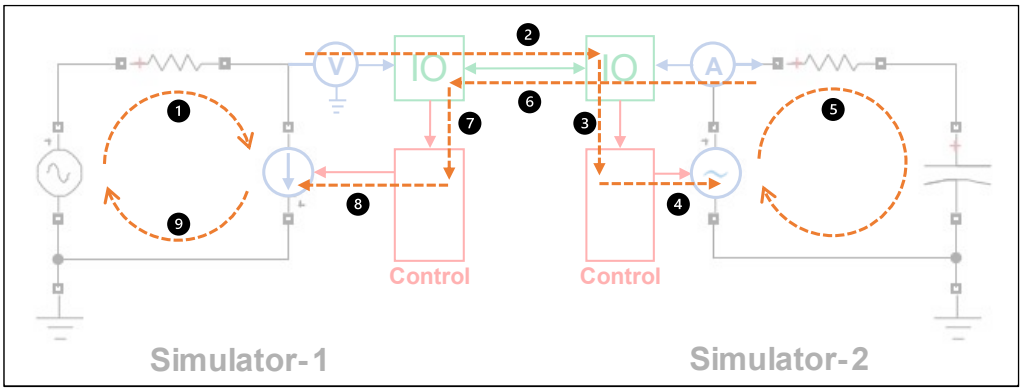


Fig 3-2 Operation of split-system model by direct transfer of interface variables (global control loop)

It is clear (Fig 3-2) that the round-trip time is long in this method and would reliably work for system frequencies much lower than the control bandwidth. Slight system jitters would de-stabilise the loop rendering the simulation impractical unless slow averaging experiments are the objective.

3.2.3 Exchanging Dynamic Phasors

Shifting the exchange process in the frequency domain is highly beneficial especially for electrical systems where the *electromechanical transients* (EMeT) are much slower than *electromagnetic transients* (EMaT), see Fig 3-3 below.

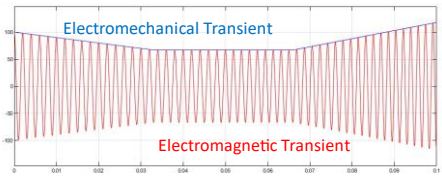


Fig 3-3 Difference in Electromagnetic (EMaT) and Electromechanical (EMeT) transient signals

In such scenario, exchanging the time-varying amplitudes of sinusoidal waveforms (dynamic phasors, or DP, discussed in further detail in the next chapter) significantly drops the bandwidth required from the control loop, which remains the same as explained in the previous section. Intermediate conversion blocks (time/frequency

domain) are added in both controllers as seen in Fig 3-4 below. This is the same method used in the RT Superlab case study discussed in the previous chapter [93].

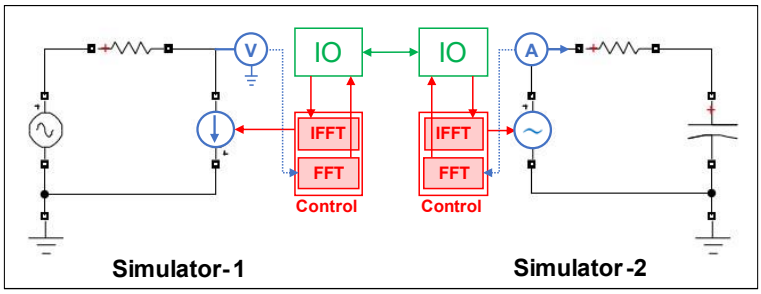


Fig 3-4 Operation of split-system model by exchanging interface variables in the frequency domain

As mentioned earlier in this section, electrical systems are characterised by fast EMaT and slow EMeT. This furnishes the opportunity of exchanging the amplitude information (instead of the complete waveform) at a much slower rate without compromising the fidelity of the reproduced waveform. This is where DPs are useful, which are time-varying Fourier Series (FS) coefficients of any waveform over a particular moving window. Broadly, the steps followed are as follows:

- At Simulator-1, waveform is dissociated into its constituent DPs over a particular time window. The time window "length" depends on the speed of change of the EMaT, i.e., if the amplitude of a fast sinusoid (10kHz, i.e., 100µs time period) can be approximated to remain constant over 10ms, the time window for FS computation can be chosen as 10ms.
- Specific DPs that are of interest for the experiment/simulation are communicated over to Simulator-2.
- Simulator-2 converts the DPs back to sinusoidal waveform using the amplitude and frequency information

It is interesting to note that conversion to frequency domain (FFT in Fig 3-4 above) produces amplitude, frequency, and phase information; and hence, all three components (of a particular interface variable) can be exchanged between two locations to reproduce the exact waveform. This, however, would require the same communication bandwidth as the previous approach (i.e., directly exchange time-

domain waveform) and defeats the purpose of exchanging DPs. The primary reason for exchanging DPs in a band-limited communication medium (like the internet, with indeterminate delays) is to slow down the communication speed.

3.2.4 Local Digital Twin

In some cases, the available bandwidth is not enough for the experiment as faster transients must be simulated, e.g., fast-varying frequency and/or amplitudes of the VS in the above experiment. In such cases, a proxy model must be deployed in both the simulators to directly deal with the physical system locally, deputising for the physical system with the other simulator (Fig 3-5). Both “follower” models must be kept converged to their respective “master” systems via standalone control over the network. This is the same method used in the soldier-in-the-loop case study discussed in the previous chapter [79].

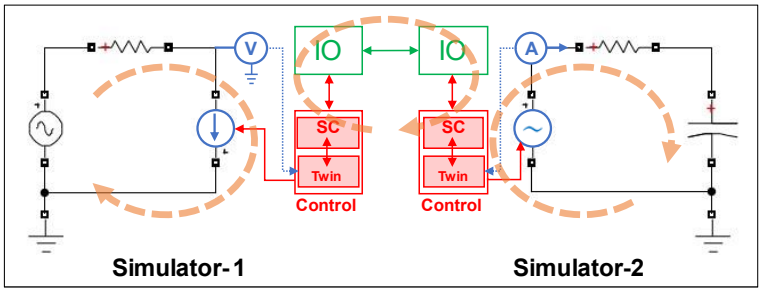


Fig 3-5 Operation of split-system model by employing a local Digital Twin

It must be noted here that this scheme has the least amount of interdependence among the local and global control loops; the local P-HIL simulator power stage (sensors and actuators) is instantaneously controlled with the locally residing *Digital Twin* only, action/reaction happens locally. There is a lot more design control to the user and this is much safer to operate. The *state convergence* (SC) loop is responsible for ensuring the *Digital Twin* is behaving as closely as possible to the physical system.

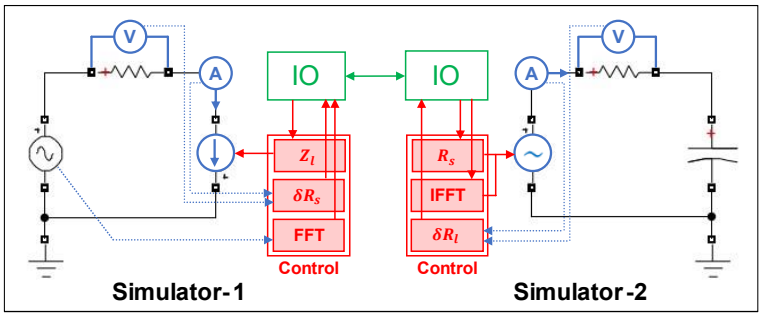


Fig 3-6 Operation of split-system model by employing a local Digital Twin operating on Dynamic Phasors

In the above example case (Fig 3-6), the RL load can be easily modelled analytically in Simulator-1. Let us assume that the load resistor R_l has a temperature-dependent δR_l additional component that cannot be modelled and hence must be transmitted over network. This component of information shall unavoidably be bandwidth-restricted. Similarly, for the other side, the δR_s and the VS DP information may be sent over to simulator-2 for modelling. *Dynamical stability* of both the local closed loops is kept isolated from the network.

3.3 Digital Twin Approaches

3.3.1 Introduction to Digital Twin

The concept of Digital Twin (DT) is fairly new and has gained interest in industry and academia alike in the past decade. [97] provides a comprehensive review on DT and provides a “thematic” collation of all publications that involves this concept. [98] offers a different take on reviewing this literature and offers a cross-industry viewpoint (most other review work is focussed on manufacturing or aircraft industry). The origin of this terminology can be attributed to Michael Grieves and their work with NASA in 2003 [99], however, the concept has existed since the 1960s [98] (NASA in its Apollo program built two replica space vehicle – one that stayed on Earth was used to replicate real operating condition to simulate behaviour in real time). A DT is described as consisting of three parts: a physical product, a digital representation of this product, and bi-directional data connections. The feature for the digital model to adapt to changing

circumstances in the physical product under real-time operation is what differentiates a **Twin** from a mere **model**.

A majority of work on DT have focussed on manufacturing and industry 4.0 [98]. Some relevant works [100] related to electric vehicles and electric power systems have been identified here. [101] discusses the process of developing a DT for an electric propulsion drive system for an autonomous vehicle, [102][103] discusses implication of DT in the development and lifecycle monitoring of EV batteries, [104][105] discusses DT for electric motor, [106] discusses a neural network model to develop DT for a generic power converter, [107] discusses a controller-embeddable real-time DT for power converter diagnostics.

3.3.2 Modelling strategy

Direct exchange of interface variables between connected HIL centres is not always feasible for tightly coupled distributed simulations systems as discussed in the previous sub-chapter, necessitating the presence of a remote DT that directly interfaces with the power stage. This scheme is composed of two parts (Fig 3-6 above): a DT that is fairly accurate for short duration, until it is "corrected" by the state convergence (SC) algorithm over a slower period based on the internet/network link.

A recent review paper [108] offers a comprehensive sweep of the state of the art in real-time modeling and simulation strategies of power electronics systems. *Modeling* is the mathematical representation of a physical system, something that can be interpreted by a computer, whereas *Simulation* is the numerical solving of this model on a computer over small time steps [109]. Different authors categorise models differently, but the overall theme remains the same: *detailed physical model* and *simple behavioural model*.

Physical Model, as the name suggests, is an accurate representation resembling the physical system as closely as possible. Power Electronic devices like IGBTs and diodes are represented by their non-linear v-i transfer functions and everything is modelled at the granular level. This results in the highest level of accuracy of output as well, but at the cost of complexity and slow simulation speed. These are also in the time-domain

and time steps in the order of ns, hence require an FPGA platform. [110] provides a clear description of developing a detailed model for a generic electric powertrain composed of a voltage source converter (VSC) and an induction machine. These aspects are further discussed ahead in this sub-chapter.

Behavioural Model is essentially a simplified version of the above aimed at faster and simpler computation most often by system linearisation and behaviour approximation. [111] divides behavioural models in power electronics systems into three categories:

- **Ideal Model Approach** – Non-linear devices like IGBT/MOSFET and diodes are approximated by on/off modes only and are replaced by an equivalent on/off resistor R_{on} or R_{off} (Fig 3-7). Stray inductance and capacitance are neglected, and all switch transients (rise and fall times and activation delay) are ignored.

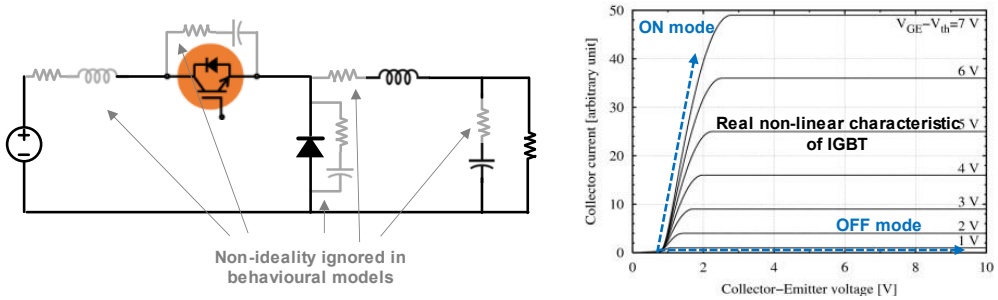


Fig 3-7 Ideal Model approach for modelling electrical systems

- **Switching Function Approach** – In above approach, since the non-linear devices, although linearised, are still approximated as two different devices in their two modes of conduction, result in different circuit topologies in different modes. If there are several of these devices, the total number of topologies accounting for all on/off combinations would be many and the simulator may struggle to toggle between them. The next level of abstraction is to replace the switching circuit with an equivalent ideal controlled current source and voltage source depending on the external behaviour (see Fig 3-8 below). In this approach, the switching action is implemented by switching functions f(t) and g(t) that is derived from the turn on/off directive of the IGBT, instead of constantly switching between rigid topologies.

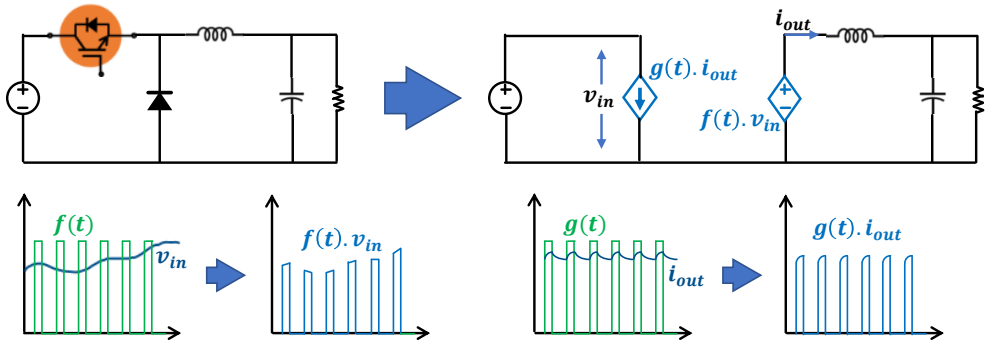


Fig 3-8 Switching Function approach for modelling electrical systems

- **Average Model Approach** – A further level of simplification can be achieved by completely eradicating the high frequency switching function. For example, in Fig 3-8 above, the switching function f(t) has a duty cycle of 40%. This simplification results in replacing the voltage source with a 40% pre-multiplied voltage source. The LC filter vanishes (slow varying load current and source voltage), and we are left with a duty cycle dependent voltage source and load resistor. This method is also referred to as the *State-Space Averaging* (SSA) technique that transforms the circuit into a linear time-invariant (LTI) system, making it fast. This topic is discussed further in a later sub-chapter.

The rest of the sub-chapter discusses various concepts introduced above in further detail. Various aspects of choosing the hardware platform based on project requirements is followed by an introduction to non-linear modeling. The Resistive Companion Form (RCF) model is a popular time-domain that is discussed next, followed by frequency-domain Multi-Frequency Averaging (MFA) that covers the foundation of SSA and builds from it.

3.3.3 Hardware Platform (CPU v FPGA v GPU)

There are three main platforms that are used today for power electronics real-time simulation:

- **CPU**-based solutions, either multi-core single CPUs or multiple CPUs tied together over local area network,
- **FPGA** (field programmable gate array), and

- **GPU** (graphics processing unit).

CPU or Central Processing Unit is what we see in any computer. They are purpose-designed to be generic processor which can perform a varied list of tasks or instructions serially. Modern CPUs have multiple cores on the same chip that allow some degree of parallel computing which increases computation speed for systems that can be intuitively split in parts [112]. A simulation time-step is composed of tasks other than just the single-step numerical solver process. This includes exchanging data with external IO channels and inter-core communication which is relatively slow for CPUs and necessitates dedicated time per simulation step. A PC-cluster is a similar way of parallelisation but instead of multiple cores in the single CPU chip, there are multiple of these CPUs in their dedicated machines tied together by a local network [113]. A PC-cluster is highly scalable.

FPGA is a specialised hardware designed for highly flexible high-speed parallel operations, as opposed to serial processing of tasks in a CPU [114][115]. An FPGA consists of a multitude of programmable *logic* and *IO blocks*, all connected in a mesh of *interconnection network* that is also programmable. The FPGA programmer can choose to connect any IO block to the input or output of any logic block granting unparalleled freedom to divide the complete system into single equations and assign them each to a logic block. Moreover, this grants direct external access to any variable in this system, as opposed to CPU where external IO must go through a serial interface card like the PCI. As a result, simulation time steps in the order ns are achievable that is immensely beneficial for “hard” real-time simulation. [109] covers a wide range of various FPGA projects for electrical systems real-time simulation which reports simulation speed gains of up to 6000x in comparison to equivalent CPU setups. On the downside, FPGA is costlier than CPU which hinders easy scalability, although some projects have implemented an “FPGA-cluster” [116] for large power system simulation. Also, the programming of an FPGA takes longer than software-based programming (for CPU applications) that makes iterative model refreshes difficult in rapid prototyping. Altera and Xilinx (now a subsidiary of Intel and AMD, respectively) are the two main producers of FPGA.

GPU is like a CPU but instead has thousands of cores granting enormous potential to perform parallel tasks. A common set of instructions is performed on different datasets in each core which is its speciality. This allows applications like detailed FEA model simulation (wherein the same task needs to be performed on the hundreds of mesh elements) to be performed very quickly without needing a pricey and difficult FPGA. The downside, as with CPU, is the inherent latency due to data IO only possible via an interface card (unlike direct access in an FPGA). A GPU itself is not sufficient and hence real-world applications are always in conjunction with a CPU that performs the supervisory role.

3.3.4 Non-Linear Modeling

Hardly any physical system that naturally exists is linear, there is always an element of non-linearity. But non-linear systems are difficult to solve mathematically which makes a case for linearising existing systems at the expense of model accuracy and robustness. But this is not always feasible if the system is highly non-linear over the operation range.

[117] provides a comprehensive review of various modelling techniques in power electronics systems in microgrids applications. Non-linear modelling methods are usually more complex computationally and hence do not see real-time utility. As a result, this chapter (and subsequent thesis chapters) does not investigate this any further.

3.3.5 Resistive Companion Form (RCF) Model

This is a popular technique for time-domain simulation of electrical power systems since coupling between adjacent components is *natural*, i.e., electrical nodes are power-conserved (Fig 3-9).

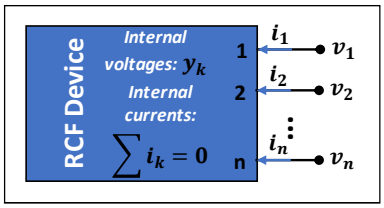


Fig 3-9 Resistive Companion Form (RCF) model

The SRC (Signal-extension Resistive Companion) model that was briefly discussed under the VTB case study section in previous chapter, is an extension of the RCF. An RCF component is represented by a through and across (current and voltage) variable for each terminal, and is represented by the following mathematical form [118]:

$$\begin{bmatrix} \boldsymbol{i} \\ \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} f_1(\dot{\boldsymbol{v}}, \dot{\boldsymbol{y}}, \int \boldsymbol{v}, \int \boldsymbol{y}, \boldsymbol{v}, \boldsymbol{y}, \boldsymbol{u}, t) \\ f_2(\dot{\boldsymbol{v}}, \dot{\boldsymbol{y}}, \int \boldsymbol{v}, \int \boldsymbol{y}, \boldsymbol{v}, \boldsymbol{y}, \boldsymbol{u}, t) \end{bmatrix} \quad 3.1$$

$\boldsymbol{v}$ is vector of external node voltages, $\boldsymbol{y}$ is vector of internal state variables, $\boldsymbol{i}$ is vector of external node currents, $\boldsymbol{u}$ is independent control vector and t is time. Since some of internal currents must be zero for conservation, the second term in the left-hand side is zero. Applying the trapezoidal rule of integration over single simulation time-step would resolve this equation to the following form:

$$\begin{bmatrix} \boldsymbol{i}(\boldsymbol{t}) \\ \boldsymbol{0} \end{bmatrix} = \begin{matrix} \boldsymbol{G}(\boldsymbol{v}(t), \boldsymbol{v}(t-1), \boldsymbol{y}(t), \boldsymbol{y}(t-1), t) \begin{bmatrix} \boldsymbol{v}(t) \\ \boldsymbol{y}(t) \end{bmatrix} \\ - \begin{bmatrix} b_1(\boldsymbol{v}(t), \boldsymbol{v}(t-1), \boldsymbol{y}(t-1), \boldsymbol{y}(t-1), \boldsymbol{i}(t-1), t \\ b_2(\boldsymbol{v}(t), \boldsymbol{v}(t-1), \boldsymbol{y}(t-1), \boldsymbol{y}(t-1), \boldsymbol{i}(t-1), t \end{bmatrix} \end{matrix} \quad 3.2$$

This resolves to the following for linear devices ($\boldsymbol{G}$ being static and $b_{1/2}$ being purely historic):

$$\begin{bmatrix} \boldsymbol{i}(\boldsymbol{t}) \\ \boldsymbol{0} \end{bmatrix} = \boldsymbol{G} \begin{bmatrix} \boldsymbol{v}(t) \\ \boldsymbol{y}(t) \end{bmatrix} - \begin{bmatrix} b_1(\boldsymbol{v}(t-1), \boldsymbol{y}(t-1), \boldsymbol{i}(t-1)) \\ b_2(\boldsymbol{v}(t-1), \boldsymbol{y}(t-1), \boldsymbol{i}(t-1)) \end{bmatrix} \quad 3.3$$

$\boldsymbol{G}$ is the Jacobian matrix of the V-I relationship of this device (in the $y = mx + c$ form, $\boldsymbol{G}$ can be interpreted as the gradient and $b_{1/2}$ as the y-intercept). $\boldsymbol{G}$ can also be interpreted as the conductance and $b_{1/2}$ as a reversed current source (see derivation of an LC filter below).

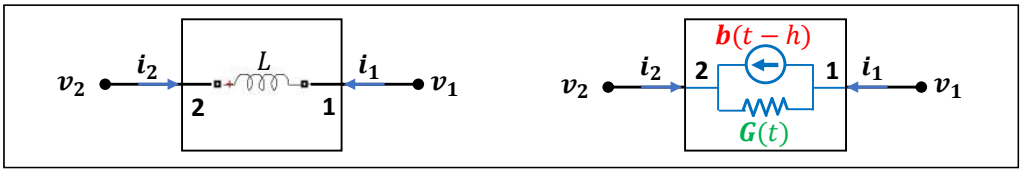


Fig 3-10 RCF model of an inductor

In the above example (Fig 3-10) of a pure inductor filter:

$$i_1 = -i_2 \tag{3.4}$$

$$\dot{i_2} = \frac{1}{L}v_2 - \frac{1}{L}v_1 \tag{3.5}$$

Equations 3.4 and 3.5 can be rearranged to simplify as follows:

$$i_2(t) = \frac{h}{2L}v_2(t) - \frac{h}{2L}v_1(t) + \frac{h}{2L}v_2(t-h) - \frac{h}{2L}v_1(t-h) + i_2(t-h)$$

$$\begin{bmatrix} i_1(t) \\ i_2(t) \end{bmatrix} = \boldsymbol{i}(t) = \begin{bmatrix} \frac{h}{2L} & \frac{-h}{2L} \\ \frac{-h}{2L} & \frac{h}{2L} \end{bmatrix} \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} + \begin{bmatrix} \frac{-h}{2L}v_2(t-h) + \frac{h}{2L}v_1(t-h) - i_2(t-h) \\ \frac{h}{2L}v_2(t-h) - \frac{h}{2L}v_1(t-h) + i_2(t-h) \end{bmatrix}$$

$$\begin{bmatrix} i_1(t) \\ i_2(t) \end{bmatrix} = \boldsymbol{i}(t) = \boldsymbol{G}(t)\boldsymbol{v}(t) + \boldsymbol{b}(t-h) \tag{3.6}$$

Equation 3.6 represents the inductor as an equivalent impedance-current source unit as shown in Fig 3-10.

[118] demonstrates an application of RCF in the VTB platform. Physical model of an induction machine was automatically translated to the RCF form using automatic differentiation (AD) and symbolic algebra. This work shows the vast utility of VTB and RCF in automation and rapid prototyping in the simulation framework. An earlier work from the 90s [119] discusses the origin mathematical foundation to the RCF (a similar

form called the Companion Link model) and expands it into a multiple de-coupled power systems (for grid transmission) application.

3.3.6 State-Space Averaging (SSA) Model

Rise of Power Electronics Converters (PECs) was discussed in the first chapter. Exponential technological development of faster switching devices and faster computing units have helped the rise of PECs in various fields like more electric aircraft, bi-directional grids, energy producer-cum-consumer at granular level (called “prosumers”), electrified transport including ships, underwater robots, and road vehicles. Modeling of PECs have introduced a new challenge: non-linearity, i.e., switches operate in one of two modes, conducting (ON) and blocking (OFF). *In simple terms, the same device behaves uniquely in the two modes, as if it were a different device altogether.*

One way to model non-linear systems is to use complex non-linear models (discussed earlier in 3.3.4), but they are computationally expensive and difficult for real-time applications. Moreover, often such high levels of detail offered by non-linear models is not required for the purpose. Averaging techniques to produce “linearised” systems offer much simpler and quicker models that have an acceptable level of approximation. State Space Averaging (SSA) is a popular method that is used by engineers for several decades now.

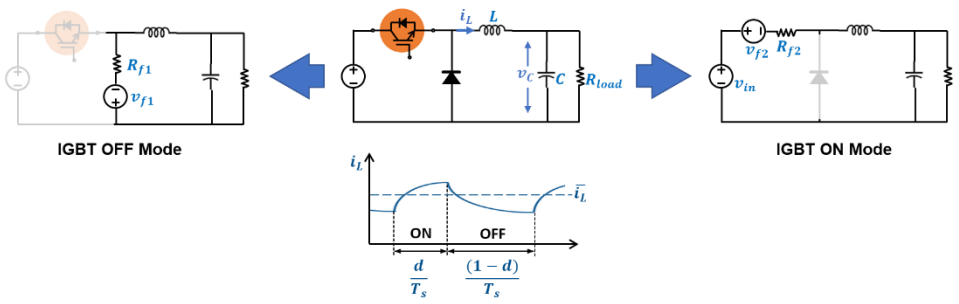


Fig 3-11 State Space Averaging (SSA) method for modelling electrical systems

The idea with SSA is to individually treat each switched configuration of the system as state space models, and then combine them into a single state space model using weighted averaging based on the duty cycle. Continuing with the DC buck converter

example (Fig 3-11 above) from above, there are two switching modes and two state variables in the system. Writing the state equations:

ON Mode:

$L\frac{di_L}{dt} = v_{in} - i_L R_{f2} - v_{f2} - v_C$ and $C\frac{dv_C}{dt} = i_L - \frac{v_C}{R_{load}}$ to produce the state function:

$$\begin{bmatrix} \dot{i_L} \\ \dot{v_C} \end{bmatrix} = \dot{\boldsymbol{X}} = \begin{bmatrix} -R_{f2}/L & -1/L \\ 1/C & -1/R_{load} \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} v_{in} - v_{f2}/L \\ 0 \end{bmatrix} = \boldsymbol{A_{ON}X + B_{ON}} \quad 3.7$$

OFF Mode:

$L\frac{di_L}{dt} = -i_L R_{f1} - v_{f1} - v_C$ and $C\frac{dv_C}{dt} = i_L - \frac{v_C}{R_{load}}$ to produce the state function:

$$\begin{bmatrix} \dot{i_L} \\ \dot{v_C} \end{bmatrix} = \dot{\boldsymbol{X}} = \begin{bmatrix} -R_{f1}/L & -1/L \\ 1/C & -1/R_{load} \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} -v_{f1}/L \\ 0 \end{bmatrix} = \boldsymbol{A_{OFF}X + B_{OFF}} \quad 3.8$$

Assuming a duty cycle of d we can combine the two state equations into one as following:

$$\dot{\boldsymbol{X}} = (d\boldsymbol{A_{ON}} + (1-d)\boldsymbol{A_{OFF}})\bar{\boldsymbol{X}} + d\boldsymbol{B_{ON}} + (1-d)\boldsymbol{B_{OFF}} \quad 3.9$$

Where $\bar{\boldsymbol{X}} = \begin{bmatrix} \overline{i_L} \\ \overline{v_C} \end{bmatrix}$

The state variable $\boldsymbol{X}$ is now the average value over the switching period indicated by $\bar{\boldsymbol{X}}$. This way, at the cost of losing the granularity within the averaging period (which is the switching period), the system is being transformed into an LTI system which is fast and simple. The problem is that all high frequency components introduced by the switching action is lost and only the slow-varying behaviour is retained. The switching ripple is crucial in many applications like calculating switching losses in the converter and electric machine. This is where the more generic form of averaging called Multi-Frequency Averaging (MFA) or Generalised Average Modelling (GAM) or Generalised SSA (GSSA) comes handy.

3.3.7 Multi-Frequency Averaging (MFA) Model

Advantages of treating state waveforms of an electromechanical system in the frequency-domain (dynamic phasors, DP) were discussed earlier in this chapter. Since electromagnetic transients (EMaT) are much faster and more predictable than electromechanical transients (EMeT) that are prone to unforeseen variations due to various mechanical aberrations in the powertrain, there is a strong advantage to transform time-domain EMaT waveforms into a slower-varying EMeT waveforms through means of DPs. Exchanging these DPs through slow and indeterministic internet for GD-HIL applications have proved successful (section 3.2.3). Advantage of having a local DT was also discussed in applications where the degree of remote coupling (DoRC) required faster exchange of interface signals than is feasibly possible in the underlying network (section 3.2.4).

It can be reasonably inferred that having the DT in the frequency-domain as well would help with computation requirements due to two reasons. Time-domain exchange waveforms do not need to be converted to/from frequency-domain for sent and received packets. Moreover, the core control model of the simulator can be operated at a slower time-step to take advantage of the inherently slow nature of EMeT. Motor control is most popularly done with DQ control which is anyway in the frequency-domain. There is a disadvantage too, since HIL power actuation and power sense physical signals are in time-domain, they need to translate between domains to interface with this core control model.

This method was first reported in 1991 [120]. Since then, it has been used in various engineering applications: electrical machines [121][122][123], DC/DC converters [124], flexible AC transmission systems [125], power system dynamics [126], sub-synchronous resonance [127], HVDC systems [128]. This strategy has been used adopted for the remote DT in this PhD. The mathematics have been elaborated and further developed in the next chapter.

3.4 Summary

This chapter delved into the technical details of implementing geographically distributed P-HIL simulation. An example of two P-HIL simulators separated by distance and connected only by a suitable telecommunication medium such as the internet was investigated. Three main methods of information exchange between the simulators were discussed: direct exchange of interface variables, direct exchange of dynamic phasors (interface variables but in frequency domain) and stationing local Digital Twins at each simulator. It was concluded that each method is applicable but depends on the Degree of Remote Coupling (DORC) that is required to be effected across the interface.

Next it was concluded that owing to the fast time constants in Power Electronic Converters (PECs), these HIL simulators are always very tightly coupled to the rest of the system, and hence require a Digital Twin approach. Various approaches that have been used in other applications for a Digital Twin (also referred to as Observer Model or Slave Model in other works) were discussed. The resistive Companion Form (RCF) Model is a popular approach that is used in the Virtual Testbed (VTB) project from South Carolina. This uses interface variables being exchanged in the time-domain form. The State Space Averaging (SSA) was next discussed which is arguably the most popular choice for reduced order modeling of PECs. This exploits the fact that PECs use switching devices that are always working in either of two binary states: On or Off, and every switching cycle can be averaged by assuming the system is in fact operating in a grey zone (instead of two black and white zones). This increases the speed of computation rapidly and produces superior results if the switching frequency is significantly higher than any waveform of interest.

An extension of the SSA is the Multi-Frequency Averaging (MFA) approach. This approach uses the FS expansion to dissociate the system variables into an infinite sum of sinusoids, which in case of PECs is only a limited number of sinusoids (usually the fundamental frequency, switching frequency and few harmonics and sidebands to the carrier). The state space system is then solved not just for the average value (as in the

case of SSA) but for a selected number of sinusoids that has been pre-chosen by the operator.

The MFA Digital Twin has been chosen for this PhD project for several reasons. First, it is a significant jump in waveform fidelity at the marginal increase in computation overhead. Second, the model can be pre-configured for the desired level of fidelity by choosing the frequency components one wants to compute in each simulation step, which is particularly handy when this DT shall be deployed in multiple remote HIL sites that may require distinct levels of fidelity requirements due to different degrees of coupling. Third, when information is exchanged between HIL sites is in form of Dynamic Phasors, there is no need for time/frequency-domain conversion since the MFA model also operates on the frequency domain.

The term “State Convergence” has been used in this chapter that is responsible for ensuring that the remote DT and the local model (or physical system) it is “following” are kept always converged, ensuring global stability of the simulation. This aspect of the project, although originally planned to be investigated, was unrealistic to achieve faithfully within the given time and personnel-power constraints. This topic should be further studied in the future to truly unlock the complete potential of the GD-HIL method.

CHAPTER 4 Digital Twin of an Electric Powertrain Emulation using Multi-Frequency Averaging Technique

4.1 Introduction

After a broad introduction chapter, the previous two chapters covered the state of the art in technologies related to HIL simulation and extended towards power-HIL (P-HIL) and geographically distributed HIL (GD-HIL). The thesis splits into two major focus areas of work of this PhD hereon. This chapter (PART 1) discuss the MFA model of a generic electric powertrain deployable in any number of remote HIL locations. Chapters 5 and 6 (PART 2) discuss the implementation of motor emulation and how it interacts with a physical inverter-under-test.

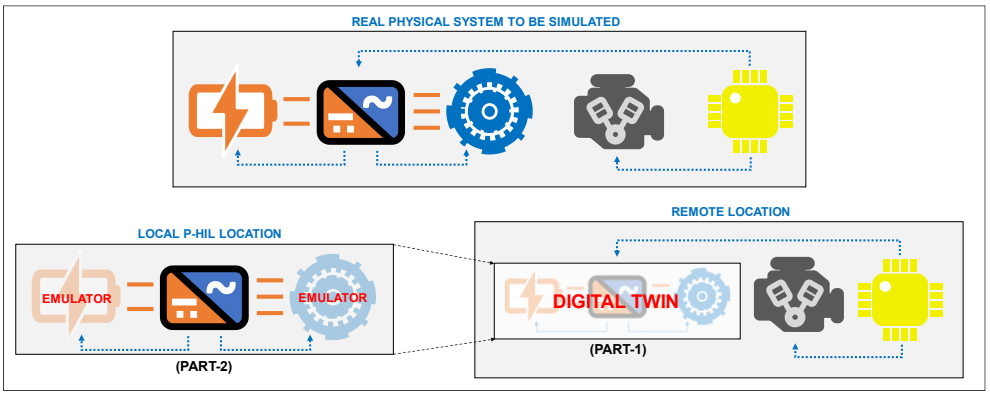


Fig 4-1 Real automotive hybrid powertrain system split into local P-HIL location (at Nottingham) and remote location

It was concluded in the previous chapters that Multifrequency Averaging (MFA) approach offers the best in terms of speed, fidelity, and configurability for a DT of a generic electric powertrain. This chapter builds the mathematical framework of the MFA model and extends it further to introduce the novel improvement of continuously variable frequency operation (section 4.2.3) required for realistic automotive drivecycle simulation, and novel method of time-domain multiplication operation in the frequency domain (“modified convolution” discussed in section 4.2.4).

This chapter breaks down the MFA portion into three parts. First (sections 4.2 and 4.3), the theory of Fourier Series (FS) expansion is discussed, and the MFA model mathematics is developed. Next (section 4.4), the above is used to build a MATLAB/Simulink™ model and various features of this is discussed. Last (sections 4.5 and 4.6), the model is simulated for various configurations and the performance (speed and accuracy) is benchmarked against an equivalent detailed switching model.

4.2 Theory of MFA Digital Twin

The MFA model uses the Fourier Series (FS) expansion of periodic time-domain to convert them into frequency-domain phasors. Frequency domain phasors (or Dynamic Phasors, or DP) are fixed-frequency sinusoids with a time-varying amplitude. It has been discussed earlier why MFA is a good compromise between computing speed and fidelity of waveforms produced. The specific advantage to PEC systems is that all non-sinusoidal waveforms produced are inherently composed of well-defined limited number of sinusoids which can be treated individually to get significant computing advantage without losing fidelity.

MFA has been deeply discussed in the research community over the past decades. This has however not been practically implemented for industry applications to their full potential. Only lately with the advent of cheap and powerful computers has this started albeit for fixed fundamental frequency applications. This PhD focusses on deploying this powerful tool towards practical industry application of automotive propulsion. *Automotive propulsion deals with constantly varying fundamental frequency and an extension to MFA that can account for this has been developed throughout this work and is the primary novelty element of this PhD.*

4.2.1 Dynamic Phasors

Fourier Series (FS) expansion allows decomposition of a periodic waveform into its constituent sinusoids over a specific time period $T = \frac{1}{\omega}$. This can be written in its mathematical form as follows:

$$x(t) = \sum_{k=-\infty}^{\infty} X_k e^{jk\omega t} \quad 4.1$$

Where X_k is the amplitude of the k^{th} harmonic, i.e., k^{th} FS coefficient. X_k can be calculated as follows:

$$X_k = \frac{1}{T}\int_0^T x(\tau)e^{-jk\omega\tau}d\tau \quad 4.2$$

Periodic signals however rarely exist in real life, and the concept of FS must be extended to accommodate dynamically changing waveform $x(t)$. This can be done by assuming $x(t)$ is a quasi-periodic waveform over a moving "time window", i.e., the waveform is periodic within a specific time window, that sequentially moves as real time progresses. This is better explained in Fig 4-2 below.

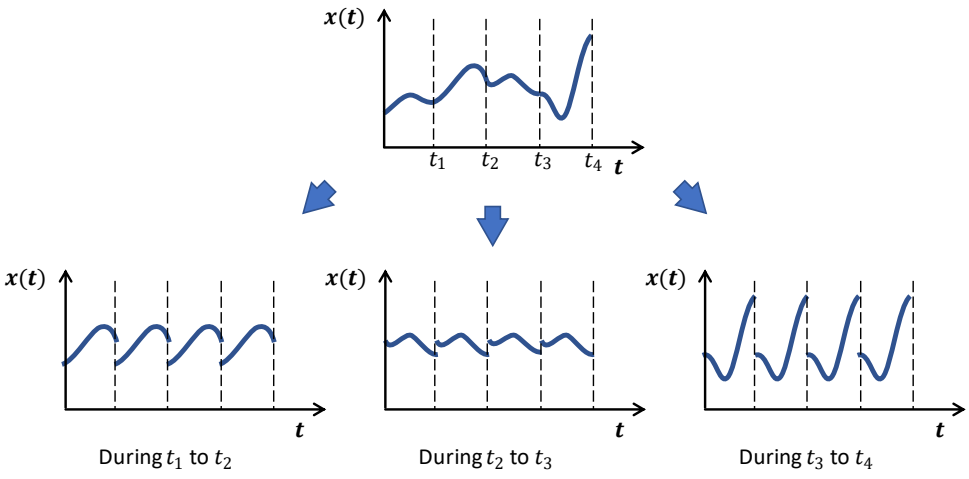


Fig 4-2 Non-periodic signal being processed as quasi-periodic signal for Dynamic Phasors

Given the above foundation, DP can be now defined as follows. The two changes from classic FS definition are: t is replaced with $\tau \in (t-T, t]$ that indicates quasi-periodic waveform in a fixed time window; and X_k is replaced by $X_k(t)$ that indicates the coefficient (or DP) is a function of time t.

$$x(\tau) = \sum_{k=-\infty}^{\infty} X_k(t) e^{jk\omega t} \quad 4.3$$

Where index-k refers to the k^{th} harmonic content. $X_k(t)$ (or $\langle \boldsymbol{x} \rangle_{\boldsymbol{k}}$) is the DP that can be produced as follows.

$$X_k(t) = \frac{1}{T}\int_{t-T}^{t} x(\tau)e^{-jk\omega\tau}d\tau = \langle \boldsymbol{x} \rangle_{\boldsymbol{k}} \quad 4.4$$

Equations 4.3 and 4.4 above represent DP in the complex exponential form, but this can be rewritten in the sine-cosine form as well (Equations 4.5-4.8 below). Converting to trigonometric form means the period T is now equal to 2π.

$$x(\tau) = \frac{1}{2}X_0(t) + \sum_{k=1}^{\infty} X_{ck}(t)\cos k\omega t + \sum_{k=1}^{\infty} X_{sk}(t)\sin k\omega t \quad 4.5$$

Where the three coefficients are defined as:

$$X_0(t) = \frac{1}{\pi}\int_{t-2\pi}^{t} x(\tau)d\tau \quad 4.6$$

$$X_{ck}(t) = \frac{1}{\pi}\int_{t-2\pi}^{t} x(\tau)\cos k\omega\tau\, d\tau \quad 4.7$$

$$X_{sk}(t) = \frac{1}{\pi}\int_{t-2\pi}^{t} x(\tau)\sin k\omega\tau\, d\tau \quad 4.8$$

The interconversion between trigonometric and complex exponential forms are defined by 4.9 to 4.11 below.

$$e^{i\theta} = \cos\theta + i\sin\theta \quad 4.9$$

$$\cos\theta = \frac{e^{i\theta} + e^{-i\theta}}{2} \quad 4.10$$

$$\sin\theta = \frac{e^{i\theta} - e^{-i\theta}}{2i} \quad 4.11$$

A key factor to consider when solving state systems using DPs is the treatment of **differentiation of a waveform with respect to time**, and **product of two waveforms**. This has been explained in [129] and the derivation shall not be discussed here. The results are as follows in Equations 4.12-4.13.

$$\left\langle \frac{dx}{dt} \right\rangle_k = \frac{d\langle x \rangle_k}{dt} + ik\omega\langle x \rangle_k \quad 4.12$$

$$\langle x.y\rangle_k = \sum_{n=-\infty}^{\infty} \langle x\rangle_{k-n}\langle y\rangle_n \quad 4.13$$

These properties are derived in the next section in the mathematical framework of the MFA DT that has been developed for this work.

4.2.2 Mathematical Framework for MFA Model

Efficacy of the MFA model is based on the rationale that solving a set of equations is less processor-intensive when solved in the frequency domain (in comparison to time domain) to achieve equivalent levels of fidelity. As the system of equations of any physical system (in this case, a generic electric powertrain) is originally in the time-domain, it needs to be converted to the frequency domain first.

Let us have a general waveform $x(t)$ as below using only DC component and two sinusoids (ω_m and ω_c for modulation and carrier frequencies respectively) for simplicity of understanding (as below). It must be noted that ω_m and ω_c do not have any physical significance at this stage of development, reader should simply assume that there are two unique frequency components in the waveform $x(t)$ and it is represented as following.

$$x(t) \approx \begin{bmatrix}\sin\omega_c t & \sin\omega_m t & 1 & \cos\omega_m t & \cos\omega_c t\end{bmatrix}.\begin{bmatrix}X_{s(1,0)}\\ X_{s(0,1)}\\ X_{(0,0)}\\ X_{c(0,1)}\\ X_{c(1,0)}\end{bmatrix}$$

$$x(t) \approx \boldsymbol{C}(t).\boldsymbol{X}(t) \quad 4.14$$

In the above equation 4.14 in comparison to 4.5, note that the ≈ symbol has been used (instead of =). This is to signify that only a limited number of sinusoids are being used to approximate the true value of the waveform $x(t)$ using the Fourier Series Expansion. Another difference is the **Fourier coefficient indexing method**, i.e., using two indices (e.g., $X_{sp,q}$) instead of 1 (e.g., X_{sk}). The first character in the subscript (either s or c) signifies the coefficient is for the sine or cosine function in the expansion. The next two characters are the two indices p and q which signify the multiplicative factor of ω_c and

ω_m respectively in the final frequency value (e.g., $X_{s1,2}$ is the coefficient of $\sin(\omega_c + 2\omega_m)t$). Lastly, any expression that is in **bold** format, signifies a matrix.

There are two operators that are used in the time-domain, i.e., **differentiation** with respect to time, and **multiplication**. Equivalent operations in the frequency-domain are conducted as follows.

Differentiation with respect to time in frequency-domain

Equation 4.14 can be differentiated with respect to time as following (the property of Product Rule or Leibniz Rule being employed).

$$\dot{x}(t) \approx \quad \dot{\boldsymbol{C}}.\boldsymbol{X} + \boldsymbol{C}.\dot{\boldsymbol{X}}$$

$$\dot{x}(t) \approx \quad [\omega_c \cos\omega_c t \quad \omega_m \cos\omega_m t \quad 0 \quad -\omega_m \sin\omega_m t \quad -\omega_c \sin\omega_c t] \begin{bmatrix} X_{s(1,0)} \\ X_{s(0,1)} \\ X_{(0,0)} \\ X_{c(0,1)} \\ X_{c(1,0)} \end{bmatrix} + \boldsymbol{C}. \begin{bmatrix} \dot{X}_{s(1,0)} \\ \dot{X}_{s(0,1)} \\ \dot{X}_{(0,0)} \\ \dot{X}_{c(0,1)} \\ \dot{X}_{c(1,0)} \end{bmatrix}$$

$$\dot{x}(t) \approx \quad [\sin\omega_c t \quad \sin\omega_m t \quad 1 \quad \cos\omega_m t \quad \cos\omega_c t]. \begin{bmatrix} 0 & 0 & 0 & 0 & -\omega_c \\ 0 & 0 & 0 & -\omega_m & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \omega_m & 0 & 0 & 0 \\ \omega_c & 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} X_{s(1,0)} \\ X_{s(0,1)} \\ X_{(0,0)} \\ X_{c(0,1)} \\ X_{c(1,0)} \end{bmatrix}$$

$$+ \boldsymbol{C}. \begin{bmatrix} \dot{X}_{s(1,0)} \\ \dot{X}_{s(0,1)} \\ \dot{X}_{(0,0)} \\ \dot{X}_{c(0,1)} \\ \dot{X}_{c(1,0)} \end{bmatrix}$$

$$\dot{x}(t) \approx \quad \boldsymbol{C}.(\boldsymbol{T}.\boldsymbol{X} + \dot{\boldsymbol{X}}) \qquad 4.15$$

Multiplication equivalent in frequency-domain

Multiplication in the time-domain is equivalently done in the frequency-domain using the convolution operator (see Equation 4.13). The exact derivation of the formula is complex and is beyond the scope of this work. However, it is interesting to note that the total number of frequency components (or sinusoids) in the resultant waveform (after multiplication in time-domain, or convolution in frequency domain) are more than the

number in the multiplying waveforms. These new sinusoids emerge in the form of sums and differences of each pair of frequency values. E.g., take two waveforms $x(t)$ and $y(t)$ each with DC component (0 Hz) and two AC components (e.g., 500 Hz and 20 kHz). $z(t) = x(t).y(t)$ would have the following frequency components:

- DC component (0 Hz)
- Original AC components (500 Hz and 20 kHz)
- New AC components, i.e., sums (1000 Hz, 40 kHz, 20.5 kHz)
- New AC components, i.e., differences (19.5 kHz)

4.2.3 Variable Frequency Operation

In equations 4.14 and 4.15 above, the MFA model framework was laid down. The inherent advantage comes from the fact that the instantaneous variation of a signal waveform (e.g., current, voltage of motor or inverter) is caused by two contributors: constituent sinusoids repeating over a fixed period (very fast time constant, as a function of the frequency value), and amplitudes of each sinusoid (much slower time constants, as a function of electrical filter time constants and/or mechanical properties). With the MFA model, the first kind of instantaneous variability (i.e., sinusoids) is not calculated, only the second kind (the amplitudes of sinusoids, or the dynamic phasors) are part of the simulation model. This allows for the time-step to be much larger and simulation speeds much faster.

This advantage is realised by defining the complete set of analytical equations (i.e., the state system) for the model in the frequency-domain only, and solving these equations at every time-step. The conversion of these dynamic phasors back into a time-varying waveform is done using Equation 4.14 by using the $\boldsymbol{C}(t)$ matrix. The underlying assumption in the $\boldsymbol{C}(t)$ matrix is that frequency ω is constant. With $\omega(t)$ being a function of t (i.e., variable frequency operation), this assumption does not hold true. In this case, the sinusoid should be a function of θ.

$$\theta = \int_{t-\tau}^{t} \omega . d\tau \qquad 4.16$$

Applying these changes in equations 4.14 and 4.15:

$$x(t) \approx \quad [\sin\theta_c(t) \quad \sin\theta_m(t) \quad 1 \quad \cos\theta_m(t) \quad \cos\theta_c(t)].\begin{bmatrix} X_{s(1,0)} \\ X_{s(0,1)} \\ X_{(0,0)} \\ X_{c(0,1)} \\ X_{c(1,0)} \end{bmatrix}$$

$$x(t) \approx \quad \boldsymbol{C}(\theta(t)).\boldsymbol{X}(t) \tag{4.17}$$

Similar to the operation in Equation 4.15,

$$\dot{x}(t) \approx \quad \dot{\boldsymbol{C}}(\theta).\boldsymbol{X} + \boldsymbol{C}(\theta).\dot{\boldsymbol{X}}$$

$$\dot{x}(t) \approx \quad [\dot{\theta_c}\cos\theta_c \quad \dot{\theta_m}\cos\theta_m \quad 0 \quad -\dot{\theta_m}\sin\theta_m \quad -\dot{\theta_c}\sin\theta_c].\begin{bmatrix} X_{s(1,0)} \\ X_{s(0,1)} \\ X_{(0,0)} \\ X_{c(0,1)} \\ X_{c(1,0)} \end{bmatrix} + \boldsymbol{C}.\begin{bmatrix} \dot{X}_{s(1,0)} \\ \dot{X}_{s(0,1)} \\ \dot{X}_{(0,0)} \\ \dot{X}_{c(0,1)} \\ \dot{X}_{c(1,0)} \end{bmatrix}$$

$$\dot{x}(t) \approx \quad [\omega_c\cos\theta_c \quad \omega_m\cos\theta_m \quad 0 \quad -\omega_m\sin\theta_m \quad -\omega_c\sin\theta_c].\begin{bmatrix} X_{s(1,0)} \\ X_{s(0,1)} \\ X_{(0,0)} \\ X_{c(0,1)} \\ X_{c(1,0)} \end{bmatrix} + \boldsymbol{C}.\begin{bmatrix} \dot{X}_{s(1,0)} \\ \dot{X}_{s(0,1)} \\ \dot{X}_{(0,0)} \\ \dot{X}_{c(0,1)} \\ \dot{X}_{c(1,0)} \end{bmatrix}$$

$$\dot{x}(t) \approx \quad [\sin\theta_c \quad \sin\theta_m \quad 1 \quad \cos\theta_m \quad \cos\theta_c].\begin{bmatrix} 0 & 0 & 0 & 0 & -\omega_c \\ 0 & 0 & 0 & -\omega_m & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \omega_m & 0 & 0 & 0 \\ \omega_c & 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} X_{s(1,0)} \\ X_{s(0,1)} \\ X_{(0,0)} \\ X_{c(0,1)} \\ X_{c(1,0)} \end{bmatrix}$$

$$+ \boldsymbol{C}.\begin{bmatrix} \dot{X}_{s(1,0)} \\ \dot{X}_{s(0,1)} \\ \dot{X}_{(0,0)} \\ \dot{X}_{c(0,1)} \\ \dot{X}_{c(1,0)} \end{bmatrix}$$

$$\dot{x}(t) \approx \quad \boldsymbol{C}(\theta).\left(\boldsymbol{T}(t).\boldsymbol{X} + \dot{\boldsymbol{X}}\right) \tag{4.18}$$

The model operation at every simulation time-step remains unchanged apart from ensuring the $\boldsymbol{T}$ matrix is also a function of time. The $\boldsymbol{C}$ matrix changes form and should be treated as a function of θ instead of t (when transforming into time-domain).

This result of differentiation for variable-frequency DPs is identical to the one shown in [129], the only difference is the mathematical format. In this work, the MFA model is built on the matrix form.

4.2.4 "Modified Convolution" for Multiplication-Equivalent in Frequency Domain

One of the key issues with translating a system of differential equations into frequency domain is the extra complexity it brings when a simple "multiplication of two waveforms" operation is required. In the frequency domain, this operation is converted to a convolution operation which is far more computationally expensive. The advantage of solving a physical system in the frequency domain, gets reversed if there are enough multiplication operations involved in the system. The next subchapter (4.3) shows that a simple and generic EV powertrain system has at least four multiplication operations.

Computational cost is of paramount importance when developing a real-time and remote-operable model, and hence, a novel method of multiplying two time-domain signals using their dynamic phasors in the frequency domain was developed. In the MFA modelling environment, the solver is computing dynamic phasors of only a fixed set of sinusoids that is pre-defined. This new method developed takes advantage of this fact and methodically produces the resultant vector of dynamic phasors for only the specific set of sinusoids. This method is superior to a standard convolution operation since convolution produces all the sinusoids (sums, differences, and multiples, of input sinusoid frequencies) that are not required in the MFA modelling environment. Working of an example case is demonstrated below.

Let us assume we want to solve the MFA model on three frequencies, i.e., ω_c, ω_t, and DC. Two variables x_1 and x_2 are multiplied to produce y. The equivalent variables in frequency domain will be as follows.

$$x_1(t) \approx \begin{bmatrix} \sin\omega_c t & \sin\omega_m t & 1 & \cos\omega_m t & \cos\omega_c t \end{bmatrix} . \begin{bmatrix} X1_{s(1,0)} \\ X1_{s(0,1)} \\ X1_{(0,0)} \\ X1_{c(0,1)} \\ X1_{c(1,0)} \end{bmatrix}$$

$$x_1(t) \approx \boldsymbol{C}(t) . \boldsymbol{X1}(t) \qquad 4.19$$

And,

$$x_2(t) \approx \quad \boldsymbol{C}(t).\boldsymbol{X2}(t) \tag{4.20}$$

If the two equations 4.19 and 4.20 are multiplied directly, it is complex since two multiplying sinusoids of different frequencies requires the use of trigonometric identities. A straightforward way to tackle this issue is by converting this to complex exponential representation (from trigonometric representation as described above in equations 4.9 to 4.11). This is done as follows for a general variable x:

$$x(t) \approx \quad [e^{-\omega_c t} \quad e^{-\omega_m t} \quad 1 \quad e^{\omega_m t} \quad e^{\omega_c t}].\begin{bmatrix} X_{-e(1,0)} \\ X_{-e(0,1)} \\ X_{(0,0)} \\ X_{+e(0,1)} \\ X_{+e(1,0)} \end{bmatrix}$$

$$x(t) \approx \quad \boldsymbol{C_e}(t).\boldsymbol{X_e}(t) \tag{4.21}$$

Where,

$$X_{-e(p,q)} = \quad \frac{X_{c(p,q)}}{2} - \frac{X_{s(p,q)}}{2i} \tag{4.22}$$

$$X_{+e(p,q)} = \quad \frac{X_{c(p,q)}}{2} + \frac{X_{s(p,q)}}{2i} \tag{4.23}$$

With x_1 and x_2 in the exponential form, resultant set of frequencies after multiplication is simple with addition of the frequency values (since they are now in the exponent of e). When multiplying the two signals (each comprised of five unique frequencies including DC component and negative frequencies, in this example), there would be 13 components produced in the product with 7 unique frequencies (again, including DC component and negative frequencies) as follows:

- $\pm 2\omega_c$ Twin carrier frequency (New)
- $\pm(\omega_c + \omega_m)$ Higher Sideband (New)
- $\pm\omega_c$ Carrier frequency
- $\pm(\omega_c - \omega_m)$ Lower sideband (New)
- $\pm 2\omega_m$ Twin modulation frequency (New)

- $\pm\omega_m$ Modulation frequency
- DC component

Knowing that we are interested in the same set of frequencies (as the original list) in the resultant variable y after multiplying x_1 and x_2, the exact mathematical steps required to produce each resultant dynamic phasor of y (sum of a particular set of product pairs of dynamic phasors of x_1 and x_2) can be pre-calculated in the initialisation stage, hence dramatically simplifying the real-time operation and reducing the computational overhead significantly.

An example of the above is described here. Say we want to pre-compute the sum of product pairs required for the DC component of the resultant wave y (x_1 and x_2 defined above in equations 4.19 and 4.20). The general formula would be given as:

$$\begin{aligned} Y_{+e(0,0)} = \quad & X1_{-e(1,0)}X2_{+e(1,0)} + \\ & X1_{-e(0,1)}X2_{+e(0,1)} + \\ & X1_{-e(0,0)}X2_{+e(0,0)} + \\ & X1_{+e(0,1)}X2_{-e(0,1)} + \\ & X1_{+e(1,0)}X2_{-e(1,0)} \end{aligned} \qquad 4.24$$

Irrespective of the exact values of the phasors, a general formula for every phasor resulting from a multiplication operation (in time domain) can be pre-computed and simply solved in real-time. The exact implementation of this block has been described in a later sub-chapter (section 4.4.3).

It should be noted here that the exact choice of frequencies to be solved in the MFA Modelling Environment is up to the end-user. E.g., a motor modeller may want to select the modulating frequency and the sidebands to switching harmonics as they are the harmonics of interest. The choice of frequencies in the above example is simply for ease of understanding.

4.3 System of State

A generic EV powertrain system has been depicted in Fig 4-3. Battery is emulated by a controlled DC voltage source and internal resistance. The PEC is modelled by an input LC filter and standard full-bridge converter. The motor is modelled by a balanced three phase RL impedance and a three-phase voltage source emulation back EMF.

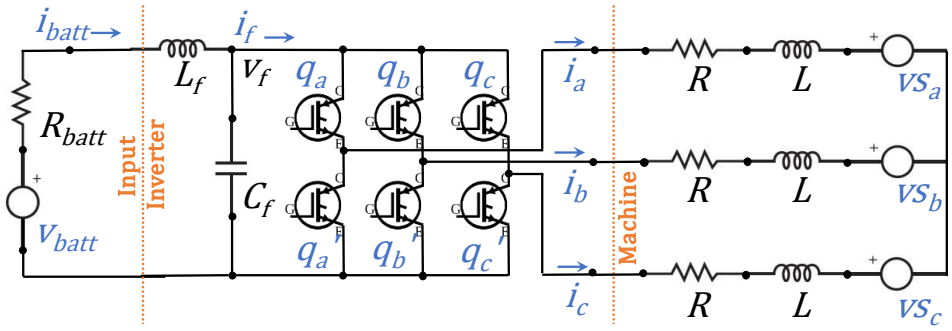


Fig 4-3 Generic electric powertrain system

The state equations are as follows:

$$\frac{d}{dt}i_{batt} = \frac{1}{L_f}v_{batt} - \frac{R_{batt}}{L_f}i_{batt} - \frac{1}{L_f}v_f \quad 4.25$$

$$\frac{d}{dt}v_f = \frac{1}{C_f}i_{batt} - \frac{1}{C_f}i_f \quad 4.26$$

$$i_f = q_a i_a + q_b i_b + q_c i_c \quad 4.27$$

$$v_{a,b,c} = v_f\left(\frac{2}{3}q_{a,b,c} - \frac{1}{3}q_{b,a,c} - \frac{1}{3}q_{c,a,b}\right) \quad 4.28$$

$$\frac{d}{dt}i_{a,b,c} = \frac{1}{L}v_{a,b,c} - \frac{R}{L}i_{a,b,c} - \frac{1}{L}vs_{a,b,c} \quad 4.29$$

The above system of equations in the time-domain can now be converted into the frequency-domain using the mathematics developed in the previous section (equations 10-11):

$$\frac{d}{dt}\boldsymbol{I_{batt}} = \frac{1}{L_f}\boldsymbol{V_{batt}} - \left(\frac{R_{batt}}{L_f} + T\right)\boldsymbol{I_{batt}} - \frac{1}{L_f}\boldsymbol{V_f} \quad 4.30$$

$$\frac{d}{dt}\boldsymbol{V_f} = \frac{1}{C_f}\boldsymbol{I_{batt}} - \frac{1}{C_f}\boldsymbol{I_f} - \boldsymbol{TV_f} \quad (4.31)$$

$$\boldsymbol{I_f} = \boldsymbol{Q_a} * \boldsymbol{I_a} + \boldsymbol{Q_b} * \boldsymbol{I_b} + \boldsymbol{Q_c} * \boldsymbol{I_c} \quad (4.32)$$

$$\boldsymbol{V_{a,b,c}} = \boldsymbol{V_{in}} * \left(\frac{2}{3}\boldsymbol{Q_{a,b,c}} - \frac{1}{3}\boldsymbol{Q_{b,a,c}} - \frac{1}{3}\boldsymbol{Q_{c,a,b}}\right) \quad (4.33)$$

$$\frac{d}{dt}\boldsymbol{I_{a,b,c}} = \frac{1}{L}\boldsymbol{V_{a,b,c}} - \left(\frac{R}{L} + T\right)\boldsymbol{I_{a,b,c}} - \frac{1}{L}\boldsymbol{Vs_{a,b,c}} \quad (4.34)$$

Each Dynamic Phasor (i.e., capitalised, and emboldened format) is a matrix of all the sinusoids amplitude values instantaneously. It is defined as:

$$\boldsymbol{X} = \begin{bmatrix} \vdots \\ X_{sp,q} \\ \vdots \\ X_{0,0} \\ \vdots \\ X_{cp,q} \\ \vdots \end{bmatrix} \quad (4.35)$$

Where,

$$\boldsymbol{X} \in \{\boldsymbol{I_{batt}}, \boldsymbol{V_{batt}}, \boldsymbol{V_{in}}, \boldsymbol{I_{in}}, \boldsymbol{Q_a}, \boldsymbol{Q_b}, \boldsymbol{Q_c}, \boldsymbol{I_a}, \boldsymbol{I_b}, \boldsymbol{I_c}, \boldsymbol{Vs_a}, \boldsymbol{Vs_b}, \boldsymbol{Vs_c}\}$$

4.3.1 Different PWM schemes for PECs

Practically all modern switched inverters operate on the principle of pulse width modulation (PWM). PWM inverters use fast switching devices to apply the positive (or negative) end of the DC input voltage periodically with a controlled duty cycle to reproduce any output voltage between 0% and 100% of the DC input voltage. The exact way the duty cycle signal (modulating signal) is converted into the switching states is a highly active area of research and novel PWM schemes are being proposed and embodied in commercial inverter controllers every year.

There are three main types of PWM schemes used in inverters:

- Sinusoidal PWM (SPWM)
- SPWM with Third Harmonic Injection (THI)
- Direct Torque Modulation (DTM)

- Space Vector Modulation (SVM or SVPWM)

2.1.1.1. Sinusoidal PWM (SPWM)

One of the most common and oldest PWM schemes is the Sinusoidal PWM (SPWM). In this method, a fast “carrier” signal, usually a sawtooth or triangular wave, is continuously compared with the modulating wave, and the resultant switch state (DC Bus Positive or DC Bus Negative) is dependent on whether the modulating wave is higher than the carrier wave. There are several alternative forms of SPWM [130]:

1. **Naturally Sampled –** Instantaneous values of the modulating and carrier wave are compared to identify the switch state. There could be three kinds of carrier waves used, leading edge sawtooth, trailing edge sawtooth, and triangular (illustration shown in Fig 4-4 below).
2. **Regularly Sampled –** As modern inverter controllers are DSP-based (digital signal processing), it significantly helps with computation expense if the modulating signal is sampled periodically and compared with an instantaneous carrier wave. Like in naturally sampled, there could two kinds of sawtooth carriers and triangular carrier wave (illustration shown in Fig 4-5 below). There are two ways of sampling in triangular, asymmetric, wherein modulating signal is sampled only once during a single wavelength of the carrier wave, usually when the peak of the carrier wave occurs. The other kind is symmetric, wherein modulating signal is sampled twice during a single wavelength of the carrier wave, when the carrier signal hits either the maximum or minimum value. This offers better resolution of the modulating signal but comes at more computation overhead. This method would be better suited to applications where modulating signal frequency is comparable to the carrier frequency.

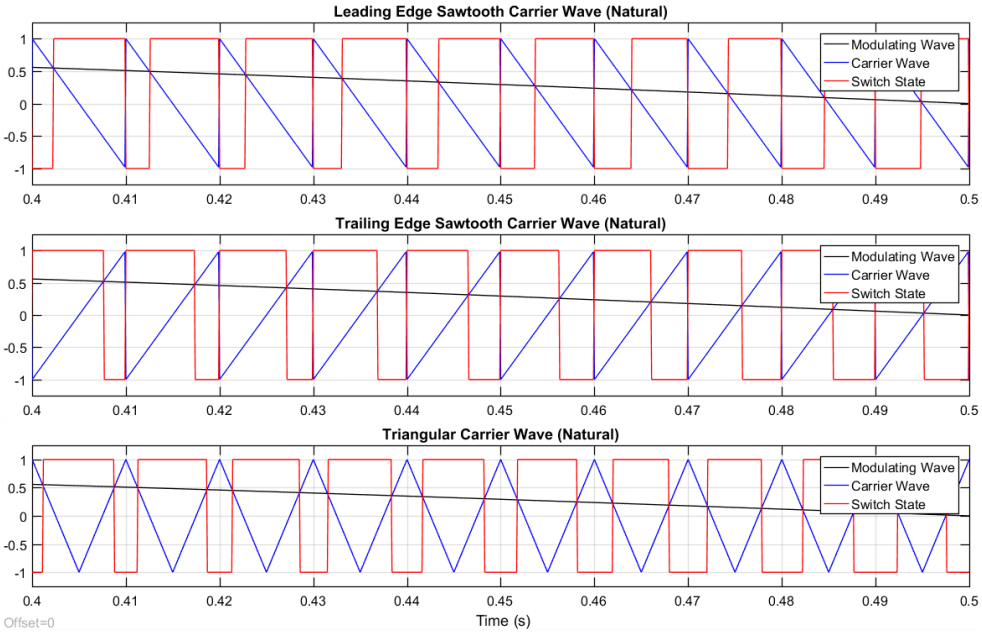


Fig 4-4 Pulse Width Modulation using Natural Sampling technique

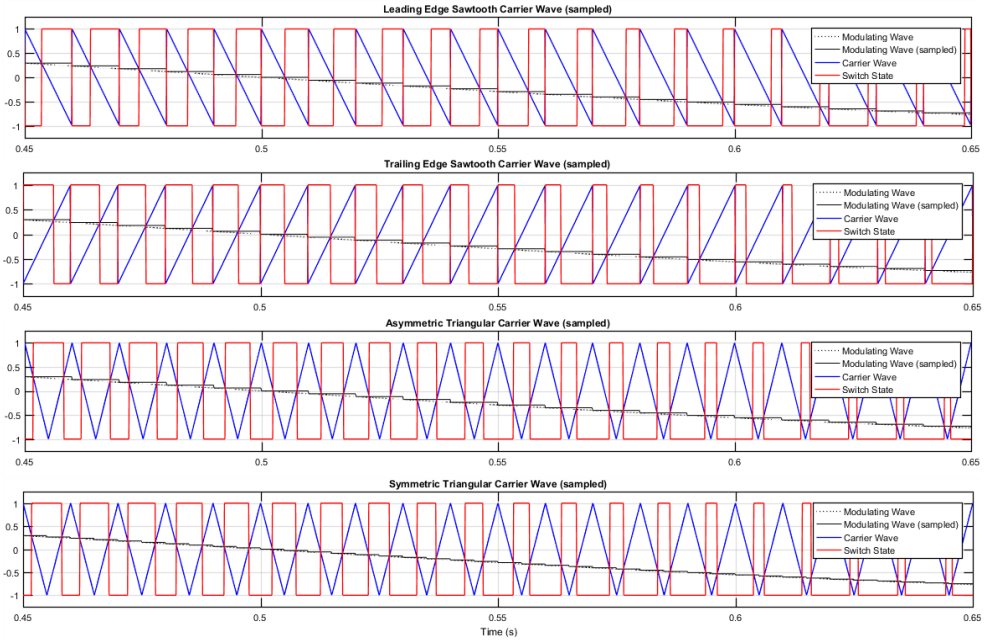


Fig 4-5 Pulse Width Modulation using Regular Sampling technique

2.1.1.2. SPWM with Third Harmonic Injection (THI)

It has been shown [131] that using the target sinusoid (the effective waveform required to be achieved by the inverter) as the only component in the modulating wave only helps achieve peak AC voltage that is only 86% ($\frac{\sqrt{3}}{2}$) of the DC bus voltage with a modulation index of 1 (maximum achievable without overmodulating). This limitation has significant

implications as the input DC side (capacitor and voltage supply) must be overrated to achieve a certain amount of AC voltage.

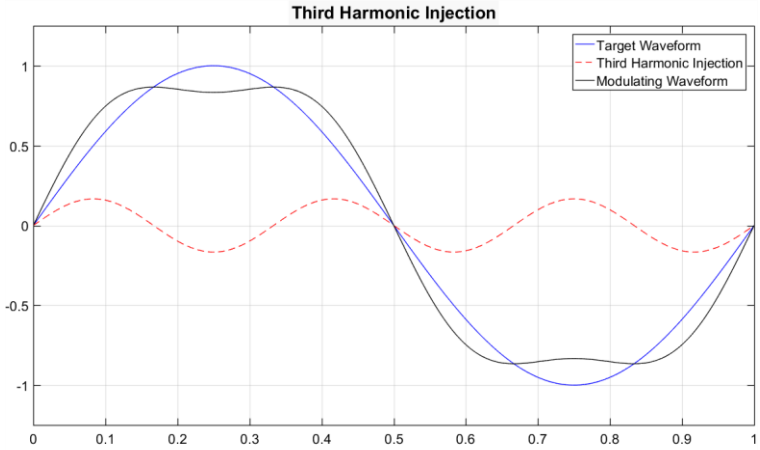


Fig 4-6 Pulse Width Modulation using Third Harmonic Injection

Applying a 3rd harmonic (Fig 4-6) modifies the original target sinusoid by softly clipping its peaks, allowing for higher than unity modulation index. Applying $\frac{1}{6}$ 3rd harmonic injection (i.e., a sixth of target sinusoid amplitude is injected as its additional third harmonic) is shown to produce maximum overmodulation capability (15% higher than 1, i.e., 1.155) which allows for the peak AC voltage waveform to be very close to DC bus voltage. Other THI ratios have been investigated in the research community [131]. For example, injection of 25% produces better harmonic performance at the cost of slight reduction of maximum modulation index (12% instead of 15%) [132].

2.1.1.3. Direct Torque Modulation (DTM)

This is an unconventional method of PWM switching wherein the switching state is activated for each carrier wavelength that achieves the same volt-second average as the original modulating waveform. This method is not practical to implement using regular DSP-based controllers as this requires numerical integration of the modulating waveform over each carrier wavelength that is computationally expensive.

2.1.1.4. Space Vector Modulation (SVM or SVPWM)

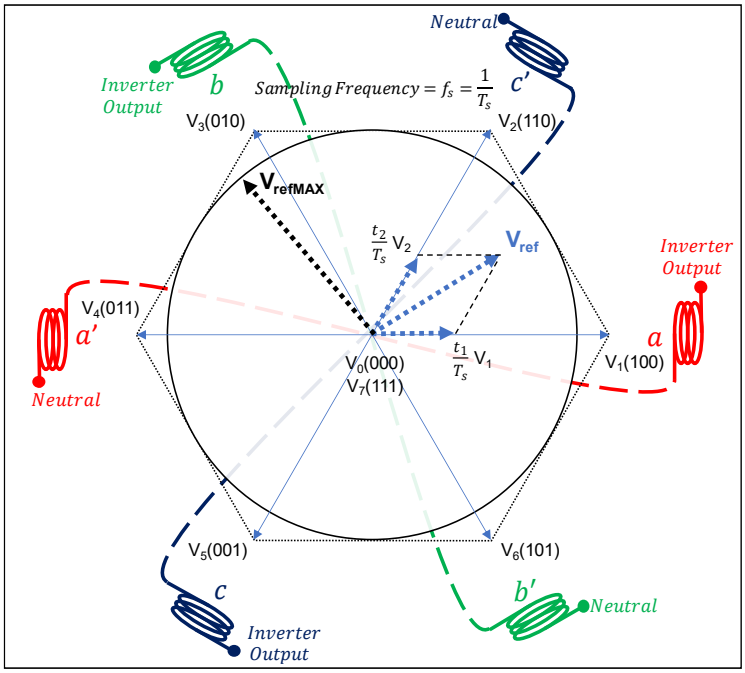
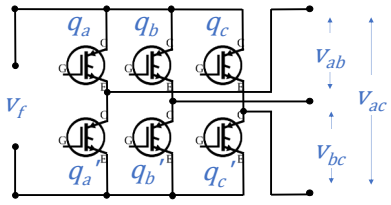


Fig 4-7 Concept of Space Vector Modulation (SVM)

This is the most common and modern form of PWM strategy that offers the designer unparalleled freedom to control the switches in a certain way to tune different performance parameters (like maximum modulation index, harmonic distortion etc.) according to application requirements. [133] provides a good discussion on the origin and basic concept of space vector plane.



Vector	q_a	q_b	q_c	q_a'	q_b'	q_c'	v_{AB}	v_{BC}	v_{CA}	
V_0 = {000}	OFF	OFF	OFF	ON	ON	ON	0	0	0	zero vector
V_1 = {100}	ON	OFF	OFF	OFF	ON	ON	$+V_{in}$	0	$-V_{in}$	active vector
V_2 = {110}	ON	ON	OFF	OFF	OFF	ON	0	$+V_{in}$	$-V_{in}$	active vector
V_3 = {010}	OFF	ON	OFF	ON	OFF	ON	$-V_{in}$	$+V_{in}$	0	active vector
V_4 = {011}	OFF	ON	ON	ON	OFF	OFF	$-V_{in}$	0	$+V_{in}$	active vector
V_5 = {001}	OFF	OFF	ON	ON	ON	OFF	0	$-V_{in}$	$+V_{in}$	active vector
V_6 = {101}	ON	OFF	ON	OFF	ON	OFF	$+V_{in}$	$-V_{in}$	0	active vector
V_7 = {111}	ON	ON	ON	OFF	OFF	OFF	0	0	0	zero vector

Fig 4-8 Different switch configurations employable in Space Vector Modulation (SVM)

To understand the concept of space vector, it is important to visualise the span of a single electrical revolution (not mechanical revolution) in the 2D plane (Fig 4-7). This plane is divided into six imaginary sectors spaced 60° apart, there being a stator coil (one of two ends of a phase) at the edge of adjacent sectors. At any given moment, the six gate switches can be energised (either ON or OFF) in exactly one of eight possible

combinations (Fig 4-8) that activates a maximum sized voltage vector along the six sector edges (two combinations produce zero vectors), generating a virtual dipole.

4.3.2 Switching Function (Double Fourier Series Representation)

Following the mathematical theory of the MFA model, and an introduction into various PWM switching techniques, the analytical solutions of the different switching functions could be developed. Equations 3-6 developed earlier in this chapter is the Fourier Series expansion for a single variable. To solve for the switching waveforms, this can be extended for two independent variables, which is required as the switching waveforms are composed of two independent variables, i.e., modulating wave and carrier wave. The derivation of this equation has been covered in [134] and shall not be covered in this work.

Assuming two independent variables x and y and their multiplying coefficients p and q, the following general equation can be defined.

$$
\begin{aligned}
f(x,y) = {} & \frac{1}{2}A_{00} && \text{(DC component)} \\
& + \sum_{\substack{q=1 \\ (p=0)}}^{\infty} \{A_{0q}\cos(qy) + B_{0p}\sin(qy)\} && \text{(Baseband and harmonics)} \\
& + \sum_{\substack{p=1 \\ (q=0)}}^{\infty} \{A_{p0}\cos(px) + B_{p0}\sin(px)\} && \text{(Carrier and harmonics)} \\
& + \sum_{p=1}^{\infty}\sum_{\substack{q=-\infty \\ (q\neq 0)}}^{\infty} \{A_{pq}\cos(px+qy) + B_{pq}\sin(px+qy)\} && \text{(Sidebands to carrier harmonics)}
\end{aligned} \tag{4.36}
$$

Where,

$$A_{pq} = \frac{1}{2\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} f(x,y).\cos(px+qy).dx.dy \tag{4.37}$$

$$B_{pq} = \frac{1}{2\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} f(x,y).\sin(px+qy).dx.dy \tag{4.38}$$

The above equations 4.36-4.38 could be used to provide an analytical solution for any application that has two independent variables. This form has been adapted to suit the application of PWM with these definitions:

- The carrier wave phase angle is the first variable:
  - $x = \theta_c = \int \omega_c . dt + \emptyset_c$
  - $\frac{dx}{dt} = \frac{d\theta_c}{dt} = \omega_c$
  - $\emptyset_c =$ Initial offset angle for carrier wave (fixed)
- The modulating wave phase angle is the second variable:
  - $y = \theta_m = \int \omega_m . dt + \emptyset_c$
  - $\frac{dy}{dt} = \frac{d\theta_m}{dt} = \omega_m$
  - $\emptyset_m =$ Offset phase angle for modulating wave (controlled)
- Index for harmonic number for the two variables is done with p and q. The following variables use this indexing method:
  - $\theta_{pq} = p\theta_c + q\theta_m = p(\int \omega_c . dt + \emptyset_c) + q(\int \omega_m . dt + \emptyset_c)$
  - A_{pq} and B_{pq} – Fourier coefficients for sinusoid of (p, q) index

Using above, equations 23-25 can be re-written as:

$$
\begin{aligned}
f(\theta_c, \theta_m) = \quad & \frac{1}{2} A_{00} && \text{(DC component)} \\
& + \sum_{\substack{q=1 \\ (p=0)}}^{\infty} \{A_{0q} \cos(q\theta_m) + B_{0q} \sin(q\theta_m)\} && \text{(Baseband and harmonics)} \\
& + \sum_{\substack{p=1 \\ (q=0)}}^{\infty} \{A_{p0} \cos(p\theta_c) + B_{p0} \sin(p\theta_c)\} && \text{(Carrier and harmonics)} \\
& + \sum_{p=1}^{\infty} \sum_{\substack{q=-\infty \\ (q\neq 0)}}^{\infty} \{A_{pq} \cos(p\theta_c + q\theta_m) + B_{pq} \sin(p\theta_c + q\theta_m)\} && \text{(Sidebands to carrier harmonics)}
\end{aligned}
\qquad 4.39
$$

Where,

$$A_{pq} = \frac{1}{2\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} f(\theta_c, \theta_m) . \cos(p\theta_c + q\theta_m) . d\theta_c . d\theta_m \qquad 4.40$$

$$B_{pq} = \frac{1}{2\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} f(\theta_c, \theta_m) . \sin(p\theta_c + q\theta_m) . d\theta_c . d\theta_m \qquad 4.41$$

These equations 4.39-4.41 form the foundation of producing the closed-form analytical solution of the switching variables $q_{a/b/c}$ (in equations 12-16) for different PWM schemes as discussed in the previous section. This can also be written in complex exponential form using the simple translation ($e^{j\theta} = \cos\theta + j\sin\theta$):

$$
\begin{aligned}
f(\theta_c, \theta_m) = \quad & \frac{1}{2}A_{00} && \text{(DC component)} \\
& + \frac{1}{2}\sum_{\substack{q=1 \\ (p=0)}}^{\infty} \left\{ \left(A_{0q} - B_{0q}\right) e^{jq\theta_m} + \left(A_{0q} + B_{0q}\right) e^{-jq\theta_m} \right\} && \text{(Baseband and harmonics)} \\
& + \frac{1}{2}\sum_{\substack{p=1 \\ (q=0)}}^{\infty} \left\{ \left(A_{p0} - B_{p0}\right) e^{jp\theta_c} + \left(A_{p0} + B_{p0}\right) e^{-jp\theta_c} \right\} && \text{(Carrier and harmonics)} \\
& + \frac{1}{2}\sum_{p=1}^{\infty}\sum_{\substack{q=-\infty \\ (q\neq 0)}}^{\infty} \left\{ \left(A_{pq} - \mathrm{B_{pq}}\right) \mathrm{e}^{p\theta_c + q\theta_m} \right. \\
& \qquad \left. + \left(A_{pq} + \mathrm{B_{pq}}\right) \mathrm{e}^{-(p\theta_c + q\theta_m)} \right\} && \text{(Sidebands to carrier harmonics)}
\end{aligned} \tag{4.42}
$$

Where,

$$
C_{pq} = \quad A_{pq} \pm jB_{pq} = \frac{1}{2\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} f(\theta_c, \theta_m).\, \mathrm{e}^{\pm(p\theta_c + q\theta_m)}.\, d\theta_c.\, d\theta_m \tag{4.43}
$$

The exact derivation has been covered in the relevant chapters of the book “Pulse Width Modulation for Power Converters” in detail and not covered in this work. The following (equations 4.3.2.1-4.3.2.11) define the frequency domain coefficients (or Dynamic Phasors, or $\boldsymbol{Q_{a/b/c}}$ in equations 4.30-4.34) for the following PWM schemes:

1. Sinusoidal PWM (SPWM)
    a. Trailing Edge Sawtooth Natural Sampled
    b. Triangular Natural Sampled
    c. Trailing Edge Sawtooth Regular Sampled
    d. Symmetric Triangular Regular Sampled
    e. Asymmetric Triangular Regular Sampled

2. Sinusoidal PWM (SPWM) with Third Harmonic Injection (THI)
    a. Triangular Natural Sampled
    b. Symmetric Triangular Regular Sampled
    c. Asymmetric Triangular Regular Sampled
3. Space Vector PWM (SVPWM)
    a. Triangular Natural Sampled
    b. Symmetric Triangular Regular Sampled
    c. Asymmetric Triangular Regular Sampled

In the following definitions, the entire equation has been presented in a divided manner for ease of understanding and translating to a computer program. Each term has been divided into the dynamic phasor (i.e., the time-varying amplitude of a sinusoid) and the corresponding sinusoid. Each phasor is multiplied by the sinusoid for every pair of indices being calculated in the engine, and added together to produce the time waveform. M is the modulating signal that is varying with time.

Specific parts of the text in the phasors are orange coloured to indicate that they do not require computation during run-time. This segregation is done to keep the run-time computation overhead low, i.e., the part of the phasor that can be pre-calculated (fixed "gain") in the initialisation stage is passed to the simulation program as a lookup table.

$J_{index}(parameter)$ signifies the Bessel function of first kind. This has been depicted for various values of index value in Fig 4-9 below. It can be visually inferred that the Bessel function applies a scaling factor of up to unity and as the value of $(index, parameter)$ increases, the function output rolls off.

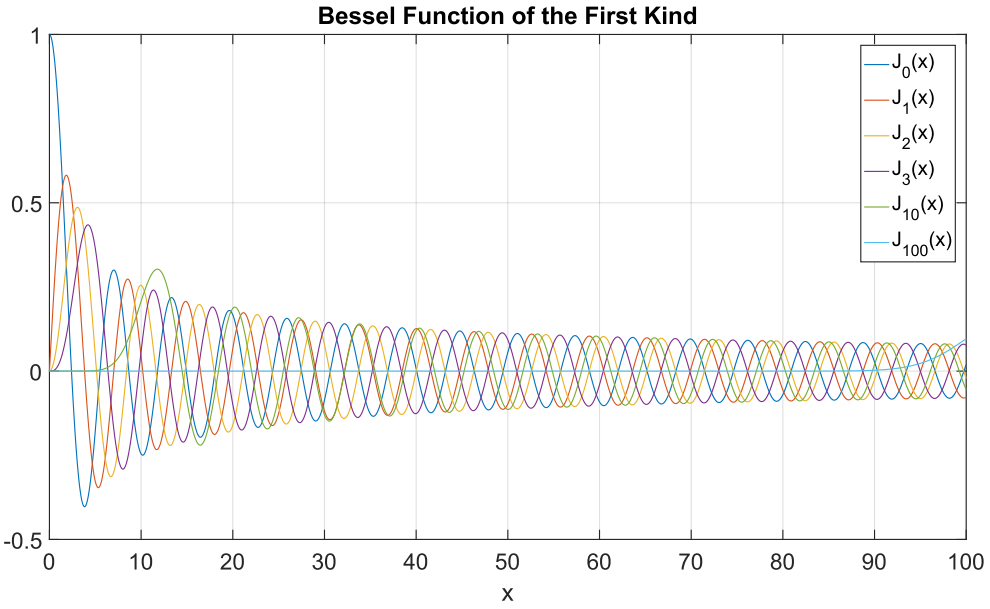


Fig 4-9 Example of Bessel function output for various index and parameter values

4.3.2.1 SPWM | Sawtooth Carrier | Natural Sampling

Dynamic Phasor (a)	Sinusoid (b)	Index	Conditions
0.5	1	$(0,0)$	
$0.5M$	$\cos\theta_m$	$(0,1)$	
$\frac{1}{p\pi}[\cos(p\pi) - J_0(p\pi M)]$	$\sin\theta_{p,0}$	$(p,0)$	$\sum_{\substack{p=1 \\ (q=0)}}^{\infty} a.b$
$\frac{1}{p\pi}\sin\left(q\frac{\pi}{2}\right)J_q(p\pi M)$	$\cos\theta_{p,q}$	(p,q)	$\sum_{p=1}^{\infty}\sum_{\substack{q=-\infty \\ q\neq 0}}^{\infty} a.b$
$\frac{-1}{p\pi}\cos\left(q\frac{\pi}{2}\right)J_q(p\pi M)$	$\sin\theta_{p,q}$	(p,q)	

4.3.2.2 SPWM | Triangular Carrier | Natural Sampling

Dynamic Phasor (a)	Sinusoid (b)	Index	Conditions
0.5	1	$(0,0)$	
$0.5M$	$\cos\theta_m$	$(0,1)$	
$\frac{2}{p\pi}\sin\left(p\frac{\pi}{2}\right)J_0\left(p\frac{\pi}{2}M\right)$	$\cos\theta_{p,0}$	$(p,0)$	$\sum_{\substack{p=1 \\ (q=0)}}^{\infty} a.b$
$\frac{2}{p\pi}\sin\left([p+q]\frac{\pi}{2}\right)J_q\left(p\frac{\pi}{2}M\right)$	$\cos\theta_{p,q}$	(p,q)	$\sum_{p=1}^{\infty}\sum_{\substack{q=-\infty \\ q\neq 0}}^{\infty} a.b$

4.3.2.3 SPWM | Sawtooth Carrier | Regular Sampling

A new term is introduced in this equation, $\omega_r = \frac{\omega_m}{\omega_c}$ is the ratio of modulating and carrier frequency. This term is required in all regular sampled modulation strategies.

$[p + q\omega_r] = T_r$ has been defined for ease of writing.

Dynamic Phasor (a)	Sinusoid (b)	Index	Conditions
0.5	1	$(0,0)$	
$\frac{1}{q\pi}\sin\left(q\frac{\pi}{2}\right)\frac{1}{\omega_r}J_q(q\omega_r\pi M)$	$\cos\theta_{0,q}$	$(0,q)$	$\sum_{\substack{q=1\\(p=0)}}^{\infty} a,b$
$\frac{-1}{q\pi}\cos\left(q\frac{\pi}{2}\right)\frac{1}{\omega_r}J_q(q\omega_r\pi M)$	$\sin\theta_{0,q}$	$(0,q)$	
$\frac{1}{p\pi}[\cos(p\pi) - J_0(p\pi M)]$	$\sin\theta_{p,0}$	$(p,0)$	$\sum_{\substack{p=1\\(q=0)}}^{\infty} a.b$
$\frac{1}{\pi}\sin\left(q\frac{\pi}{2}\right)\frac{1}{T_r}J_q(T_r\pi M)$	$\cos\theta_{p,q}$	(p,q)	$\sum_{p=1}^{\infty}\sum_{\substack{q=-\infty\\q\neq0}}^{\infty} a.b$
$\frac{-1}{\pi}\cos\left(q\frac{\pi}{2}\right)\frac{1}{T_r}J_q(T_r\pi M)$	$\sin\theta_{p,q}$	(p,q)	

4.3.2.4 SPWM | Symmetric Triangular Carrier | Regular Sampling

Dynamic Phasor (a)	Sinusoid (b)	Index	Conditions
0.5	1	$(0,0)$	
$\frac{2}{q\pi}\frac{1}{\omega_r}\sin\left(q[1+\omega_r]\frac{\pi}{2}\right)J_q\left(q\omega_r\frac{\pi}{2}M\right)$	$\cos\theta_{0,q}$	$(0,q)$	$\sum_{\substack{q=1\\(p=0)}}^{\infty} a,b$
$\frac{2}{p\pi}\sin\left(p\frac{\pi}{2}\right)J_0\left(p\frac{\pi}{2}M\right)$	$\cos\theta_{p,0}$	$(p,0)$	$\sum_{\substack{p=1\\(q=0)}}^{\infty} a.b$
$\frac{2}{\pi}\frac{1}{T_r}\sin\left([T_r+q]\frac{\pi}{2}\right)J_q\left(T_r\frac{\pi}{2}M\right)$	$\cos\theta_{p,q}$	(p,q)	$\sum_{p=1}^{\infty}\sum_{\substack{q=-\infty\\q\neq0}}^{\infty} a.b$

4.3.2.5 SPWM | Asymmetric Triangular Carrier | Regular Sampling

Dynamic Phasor (a)	Sinusoid (b)	Index	Conditions
0.5	1	$(0,0)$	

$\frac{2}{q\pi}\sin\left(q\frac{\pi}{2}\right)\frac{1}{\omega_r}J_q\left(q\omega_r\frac{\pi}{2}M\right)$	$\cos\theta_{0,q}$	$(0,q)$	$\sum_{\substack{q=1\\(p=0)}}^{\infty} a,b$
$\frac{2}{p\pi}\sin\left(p\frac{\pi}{2}\right)J_0\left(p\frac{\pi}{2}M\right)$	$\cos\theta_{p,0}$	$(p,0)$	$\sum_{\substack{p=1\\(q=0)}}^{\infty} a.b$
$\frac{2}{\pi}\sin\left([p+q]\frac{\pi}{2}\right)\frac{1}{T_r}J_q\left(T_r\frac{\pi}{2}M\right)$	$\cos\theta_{p,q}$	(p,q)	$\sum_{p=1}^{\infty}\sum_{\substack{q=-\infty\\q\neq0}}^{\infty} a.b$

4.3.2.6 THI+SPWM | Triangular Carrier | Natural Sampling

Dynamic Phasor (a)	Sinusoid (b)	Index	Conditions
0.5	1	$(0,0)$	
$0.5M$	$\cos\theta_m$	$(0,1)$	
$0.5M_3$	$\cos 3\theta_m$	$(0,3)$	
$\frac{2}{p\pi}\left[\begin{array}{c} J_0\left(p\frac{\pi}{2}M_3\right)J_k\left(p\frac{\pi}{2}M\right)\sin\left([p+k]\frac{\pi}{2}\right)\Big\vert_{k=\lvert q\rvert} \\ +J_0\left(p\frac{\pi}{2}M\right)J_h\left(p\frac{\pi}{2}M_3\right)\sin\left([p+h]\frac{\pi}{2}\right)\Big\vert_{3h=\lvert q\rvert} \\ +\sum J_k\left(p\frac{\pi}{2}M\right)J_h\left(p\frac{\pi}{2}M_3\right)\sin\left([p+k+h]\frac{\pi}{2}\right)\Big\vert_{k+3h=\lvert q\rvert} \\ +\sum J_k\left(p\frac{\pi}{2}M\right)J_h\left(p\frac{\pi}{2}M_3\right)\sin\left([p+k+h]\frac{\pi}{2}\right)\Big\vert_{k-3h=\lvert q\rvert} \\ +\sum J_k\left(p\frac{\pi}{2}M\right)J_h\left(p\frac{\pi}{2}M_3\right)\sin([p+k+h]\frac{\pi}{2})\Big\vert_{3h-k=\lvert q\rvert} \end{array}\right]$	$\cos\theta_{p,q}$	(p,q)	$\sum_{p=1}^{\infty}\sum_{\substack{q=-\infty\\q\neq0}}^{\infty} a.b$

In the above equation (and in all THI equations below), $1 \leq k < \infty$ and $1 \leq h < \infty$ with k and h being integers. In principle, the last three terms in the phasors are infinite summation which is common in closed-form analytical solutions of complex modulation strategies [131]. However, it has been observed in practice that only the first few terms need to be calculated due to the rapid roll off of the Bessel function magnitudes (see Fig 4-9 above). This is an important distinction in the THI and SV equations compared to standard SPWM modulation. This leads to significantly higher computational overhead.

4.3.2.7 THI+SPWM | Symmetric Triangular Carrier | Regular Sampling

Dynamic Phasor (a)	Sinusoid (b)	Index	Conditions

0.5	1	(0,0)	
$\frac{2}{\pi}\frac{1}{T_r}\left[\begin{array}{c} J_0\left(T_r\frac{\pi}{2}M_3\right)J_k\left(T_r\frac{\pi}{2}M\right)\sin\left([T_r+k]\frac{\pi}{2}\right)\Big\vert_{k=\vert q\vert} \\ +J_0\left(T_r\frac{\pi}{2}M\right)J_h\left(T_r\frac{\pi}{2}M_3\right)\sin\left([T_r+h]\frac{\pi}{2}\right)\Big\vert_{3h=\vert q\vert} \\ +\sum J_k\left(T_r\frac{\pi}{2}M\right)J_h\left(T_r\frac{\pi}{2}M_3\right)\sin\left([T_r+k+h]\frac{\pi}{2}\right)\Big\vert_{k+3h=\vert q\vert} \\ +\sum J_k\left(T_r\frac{\pi}{2}M\right)J_h\left(T_r\frac{\pi}{2}M_3\right)\sin\left([T_r+k+h]\frac{\pi}{2}\right)\Big\vert_{k-3h=\vert q\vert} \\ +\sum J_k\left(T_r\frac{\pi}{2}M\right)J_h\left(T_r\frac{\pi}{2}M_3\right)\sin([T_r+k+h]\frac{\pi}{2})\Big\vert_{3h-k=\vert q\vert} \end{array}\right]$	$\cos\theta_{p,q}$	(p,q)	$\sum_{p=1}^{\infty}\sum_{\substack{q=-\infty \\ q\neq 0}}^{\infty} a.b$

4.3.2.8 THI+SPWM | Asymmetric Triangular Carrier | Regular Sampling

Dynamic Phasor (a)	Sinusoid (b)	Index	Conditions
0.5	1	(0,0)	
$\frac{2}{\pi}\frac{1}{T_r}\left[\begin{array}{c} J_0\left(T_r\frac{\pi}{2}M_3\right)J_k\left(T_r\frac{\pi}{2}M\right)\sin\left([p+k]\frac{\pi}{2}\right)\Big\vert_{k=\vert q\vert} \\ +J_0\left(T_r\frac{\pi}{2}M\right)J_h\left(T_r\frac{\pi}{2}M_3\right)\sin\left([p+h]\frac{\pi}{2}\right)\Big\vert_{3h=\vert q\vert} \\ +\sum J_k\left(T_r\frac{\pi}{2}M\right)J_h\left(T_r\frac{\pi}{2}M_3\right)\sin\left([p+k+h]\frac{\pi}{2}\right)\Big\vert_{k+3h=\vert q\vert} \\ +\sum J_k\left(T_r\frac{\pi}{2}M\right)J_h\left(T_r\frac{\pi}{2}M_3\right)\sin\left([p+k+h]\frac{\pi}{2}\right)\Big\vert_{k-3h=\vert q\vert} \\ +\sum J_k\left(T_r\frac{\pi}{2}M\right)J_h\left(T_r\frac{\pi}{2}M_3\right)\sin([p+k+h]\frac{\pi}{2})\Big\vert_{3h-k=\vert q\vert} \end{array}\right]$	$\cos\theta_{p,q}$	(p,q)	$\sum_{p=1}^{\infty}\sum_{\substack{q=-\infty \\ q\neq 0}}^{\infty} a.b$

4.3.2.9 SVPWM | Triangular | Natural Sampling

Dynamic Phasor	Sinusoid	Index
0.5	1	(0,0)
$0.5M$	$\cos\theta_m$	(0,1)
$\frac{3\sqrt{3}}{2\pi(q^2-1)}\sin\left(q\frac{\pi}{6}\right)\sin\left(q\frac{\pi}{2}\right)M \quad \forall\, q\in\{3,9,15,21\ldots\}$	$\cos q\theta_m$	$(0,q)$
$\frac{4}{p\pi^2}\left[\begin{array}{c} \frac{\pi}{6}\sin\left([p+q]\frac{\pi}{2}\right)\left\{J_q\left(p\frac{3\pi}{4}M\right)+2\cos q\frac{\pi}{6}J_q\left(p\frac{\sqrt{3}\pi}{4}M\right)\right\} \\ +\frac{1}{q}\sin p\frac{\pi}{2}\cos q\frac{\pi}{2}\sin q\frac{\pi}{6}\left\{J_0\left(p\frac{3\pi}{4}M\right)-J_0\left(p\frac{\sqrt{3}\pi}{4}M\right)\right\}\Big\vert_{q\neq 0} \\ +\sum_{\substack{k=1 \\ (k\neq -q)}}^{\infty}\left[\begin{array}{c}\frac{1}{[q+k]}\sin\left([p+k]\frac{\pi}{2}\right)\cos\left([q+k]\frac{\pi}{2}\right)\sin\left([q+k]\frac{\pi}{6}\right) \\ \times\left\{J_k\left(p\frac{3\pi}{4}M\right)+2\cos\left([2q+3k]\frac{\pi}{6}\right)J_k\left(p\frac{\sqrt{3}\pi}{4}M\right)\right\}\end{array}\right] \\ +\sum_{\substack{k=1 \\ (k\neq q)}}^{\infty}\left[\begin{array}{c}\frac{1}{[q-k]}\sin\left([p+k]\frac{\pi}{2}\right)\cos\left([q-k]\frac{\pi}{2}\right)\sin\left([q-k]\frac{\pi}{6}\right) \\ \times\left\{J_k\left(p\frac{3\pi}{4}M\right)+2\cos\left([2q-3k]\frac{\pi}{6}\right)J_k\left(p\frac{\sqrt{3}\pi}{4}M\right)\right\}\end{array}\right] \end{array}\right]$	$\cos\theta_{p,q}$	(p,q)

4.3.2.10 SVPWM | Symmetric Triangular | Regular Sampling

Dynamic Phasor	Sinusoid	Index
0.5	1	$(0,0)$
$\frac{4}{\pi^2}\frac{1}{T_r}\left[\begin{array}{l}\frac{\pi}{6}\sin\left([T_r+q]\frac{\pi}{2}\right)\left\{J_q\left(T_r\frac{3\pi}{4}M\right)+2\cos q\frac{\pi}{6}J_q\left(T_r\frac{\sqrt{3}\pi}{4}M\right)\right\}\\+\frac{1}{q}\sin T_r\frac{\pi}{2}\cos q\frac{\pi}{2}\sin q\frac{\pi}{6}\left\{J_0\left(T_r\frac{3\pi}{4}M\right)-J_0\left(T_r\frac{\sqrt{3}\pi}{4}M\right)\right\}\Big\|_{q\neq 0}\\+\sum_{\substack{k=1\\(k\neq -q)}}^{\infty}\left[\begin{array}{l}\frac{1}{[q+k]}\sin\left([T_r+k]\frac{\pi}{2}\right)\cos\left([q+k]\frac{\pi}{2}\right)\sin\left([q+k]\frac{\pi}{6}\right)\\\times\left\{J_k\left(T_r\frac{3\pi}{4}M\right)+2\cos\left([2q+3k]\frac{\pi}{6}\right)J_k\left(T_r\frac{\sqrt{3}\pi}{4}M\right)\right\}\end{array}\right]\\+\sum_{\substack{k=1\\(k\neq q)}}^{\infty}\left[\begin{array}{l}\frac{1}{[q-k]}\sin\left([T_r+k]\frac{\pi}{2}\right)\cos\left([q-k]\frac{\pi}{2}\right)\sin\left([q-k]\frac{\pi}{6}\right)\\\times\left\{J_k\left(T_r\frac{3\pi}{4}M\right)+2\cos\left([2q-3k]\frac{\pi}{6}\right)J_k\left(T_r\frac{\sqrt{3}\pi}{4}M\right)\right\}\end{array}\right]\end{array}\right]$	$\cos\theta_{p,q}$	(p,q)

4.3.2.11 SVPWM | Asymmetric Triangular | Regular Sampling

Dynamic Phasor	Sinusoid	Index
0.5	1	$(0,0)$
$\frac{4}{\pi^2}\frac{1}{T_r}\left[\begin{array}{l}\frac{\pi}{6}\sin\left([p+q]\frac{\pi}{2}\right)\left\{J_q\left(T_r\frac{3\pi}{4}M\right)+2\cos n\frac{\pi}{6}J_q\left(T_r\frac{\sqrt{3}\pi}{4}M\right)\right\}\\+\frac{1}{n}\sin p\frac{\pi}{2}\cos q\frac{\pi}{2}\sin q\frac{\pi}{6}\left\{J_0\left(T_r\frac{3\pi}{4}M\right)-J_0\left(T_r\frac{\sqrt{3}\pi}{4}M\right)\right\}\Big\|_{q\neq 0}\\+\sum_{\substack{k=1\\(k\neq -q)}}^{\infty}\left[\begin{array}{l}\frac{1}{[q+k]}\sin\left([p+k]\frac{\pi}{2}\right)\cos\left([q+k]\frac{\pi}{2}\right)\sin\left([q+k]\frac{\pi}{6}\right)\\\times\left\{J_k\left(T_r\frac{3\pi}{4}M\right)+2\cos\left([2q+3k]\frac{\pi}{6}\right)J_k\left(T_r\frac{\sqrt{3}\pi}{4}M\right)\right\}\end{array}\right]\\+\sum_{\substack{k=1\\(k\neq q)}}^{\infty}\left[\begin{array}{l}\frac{1}{[q-k]}\sin\left([p+k]\frac{\pi}{2}\right)\cos\left([q-k]\frac{\pi}{2}\right)\sin\left([q-k]\frac{\pi}{6}\right)\\\times\left\{J_k\left(T_r\frac{3\pi}{4}M\right)+2\cos\left([2q-3k]\frac{\pi}{6}\right)J_k\left(T_r\frac{\sqrt{3}\pi}{4}M\right)\right\}\end{array}\right]\end{array}\right]$	$\cos\theta_{p,q}$	(p,q)

4.3.2.12 Summary of Switching Functions in MFA Model

Switching strategies for PWM-based Voltage-Source-Inverters is an active area of research, novel strategies are being proposed in the literature every year [135][136][137][138]. There are two reasons for the continued growth in the literature. *First*, new devices are increasingly being used in mainstream applications (IGBT, MOSFET, Silicon-Carbide, Gallium Nitride etc.) that require unique methods of treating the switching strategy. *Second*, there is no ideal strategy – different applications require different strategies.

The Weighted Total Harmonic Distortion with reference to modulation wave (WTHD0) has been summarised in Table 4-1 and Table 4-2 below [130][131][139]:

Strategy	Sinusoidal PWM				
Carrier Wave	Sawtooth		Triangular		
Sampling	Natural	Regular	Natural	Regular	
				Symmetric	Asymmetric
WTHD0	4.03%	4.16%	2.04%	2.05%	2.02%

Table 4-1 Weighted Total Harmonic Distortion (WTDH0) with reference to modulation wave for Sinusoidal PWM switching technique (various sampling strategies) [130][131][139]

Strategy	Third Harmonic Injection			Space Vector Modulation		
Carrier Wave	Triangular			Triangular		
Sampling	Natural	Regular		Natural	Regular	
		Symmetric	Asymmetric		Symmetric	Asymmetric
WTHD0	1.77%	1.78%	1.75%	1.75%	1.76%	1.72%

Table 4-2 Weighted Total Harmonic Distortion (WTDH0) with reference to modulation wave for Third Harmonic Injection and Space Vector Modulation switching technique (various sampling strategies) [130][131][139]

Space Vector Modulation (SVM) is by far the most popular strategy being employed in modern devices owing to the flexibility it offers with customised placement of the zero-vector in a switching wavelength. This flexibility can be used to optimise different performance metrics like computation load, switching losses, harmonic distortion etc. However, the standard form of SVM, wherein the active vectors are centred in each half carrier period, and the remainder time is equally split between the two zero space vectors, is identical to Sinusoidal PWM with Third Harmonic Injection [139]. In this work, the MFA model has been implemented with all eleven switching strategies, however, the last 3 on SVM do not have provision for customising placement of the zero-vector (the sole reason for SVM use) as this is out of scope for this PhD. The development in SVM in this PhD work is a good foundation to be continued in the future.

4.4 MATLAB/Simulink™ Implementation of MFA Digital Twin

A novel MATLAB/Simulink™ modeling environment based on the MFA methodology described in the previous two sub-chapters has been developed towards this PhD. This

sub-chapter describes the framework and highlights the interesting and novel aspects of this model.

The complete MFA Modelling Suite has been developed on MATLAB™ R2016b since the P-HIL simulator rig (Triphase) software is based on this version. The complete suite (Fig 4-10) is divided into three models:

- V3A.slx – This is the main “MFA” model that solves the system of state in the frequency domain using dynamic phasors. This model runs at a large time step only to capture the envelope function of fast sinusoids in the system, i.e., the dynamic phasors.
- V3B.slx – This is the frequency-to-time conversion model and can be used to visualise the model output from V3A in a human-interpretable time-domain format. This is used for validation purpose only.
- V3C.slx – This is the benchmark model based on conventional time-domain switching strategies. The Simulink™ Simscape Specialised Power Systems™ toolbox has been used. This is solving the same system of state but in the time-domain and at a much smaller time-step. This is also used for validation purpose only.

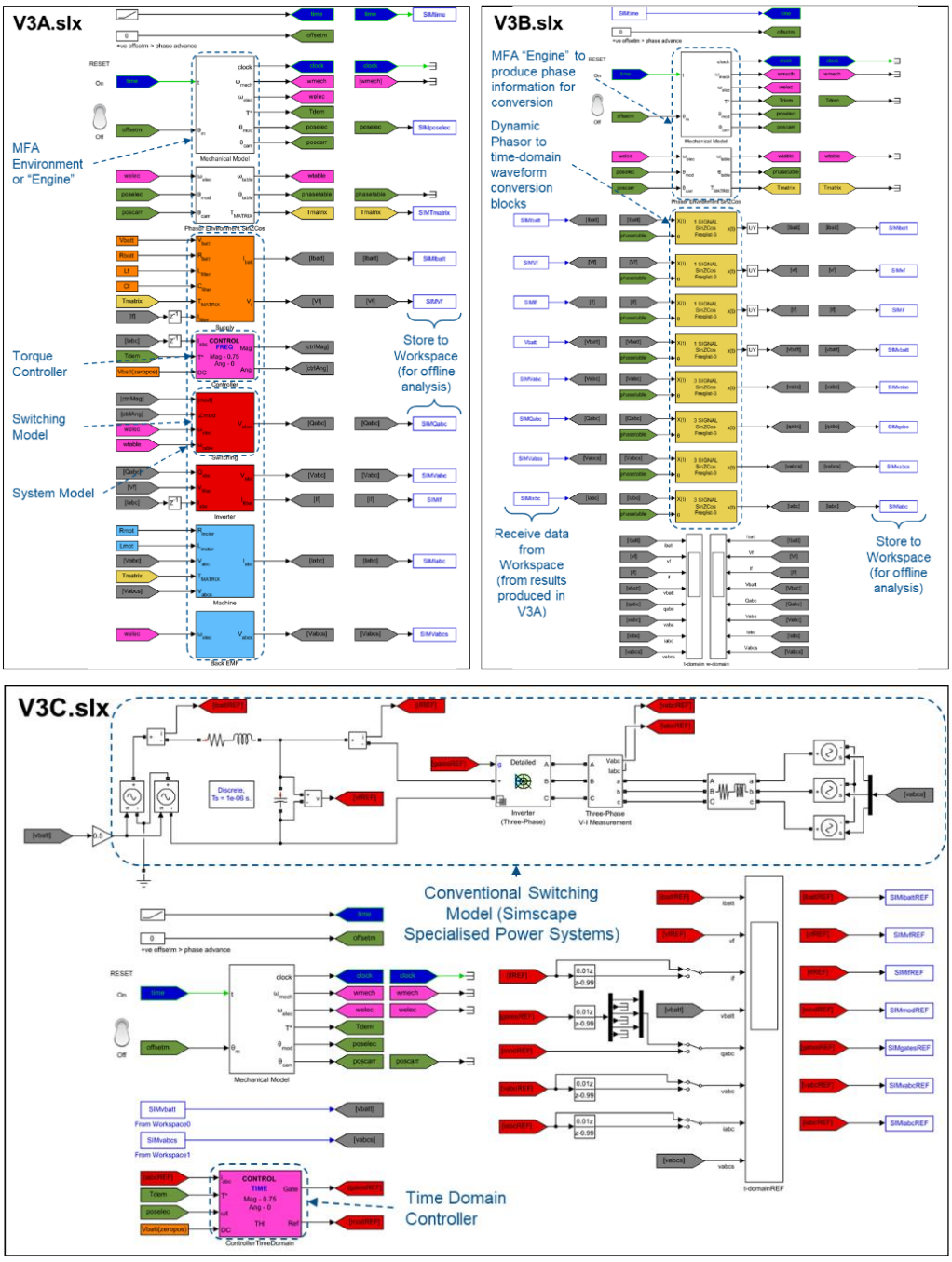


Fig 4-10 Top-level view of MFA models developed

4.4.1 MFA Environment or "Engine"

"Mechanical Model" block (Fig 4-11) is responsible for producing the frequency (mechanical and electrical), torque demand value for every simulation time step based on the lookup object that is programmed during initialisation. Based on the frequency,

a resetting integration block translates it to phase angle values for modulation and carrier waves.

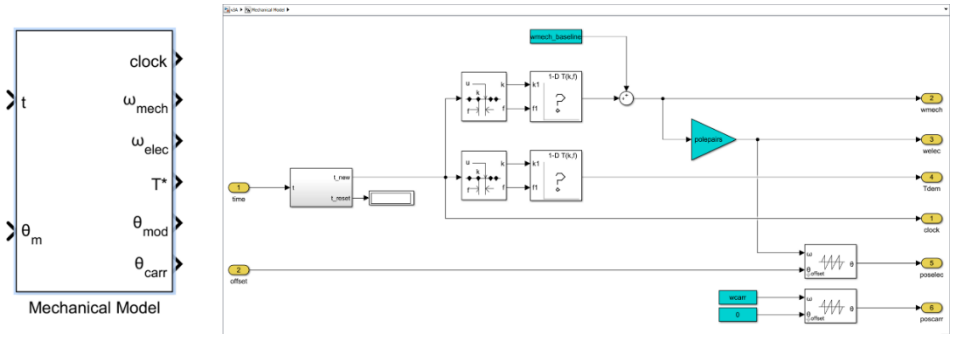


Fig 4-11 "Mechanical Model" block detailed view

“Phasor Environment SinZCos” block is responsible for converting the phase angle values (modulation and carrier waves) into a table of all the phasor angles. For example, assuming modulation and carrier frequencies of 50 Hz and 20 kHz, and MFA frequency list (or “freqlist”, in the Fig 4-12) of $\{(0\ 1), (0\ 3), (1\ -1), (1\ 1)\}$, frequency-table (or “w table” in the Fig 4-12 above) would be a vector of $\{50, 150, 19950, 20050\}$ values. Similarly, “phase table” produced is the dynamic phase angle value $(-\pi, \pi)$ for the same list of phasors, or sinusoids. This block also produces the “T-Matrix” that is an $n \times n$ square matrix as has been defined earlier in 0 and 4.2.3.

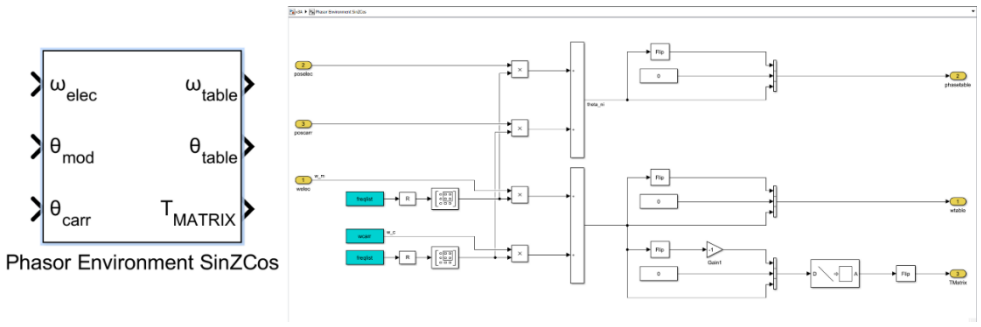


Fig 4-12 "Phasor Environment SinZCos" block detailed view

4.4.2 System Model

The MFA model is run in these blocks (Fig 4-13). The system model equations in the frequency-domain (defined above in equations 4.30-4.34) are coded within these blocks.

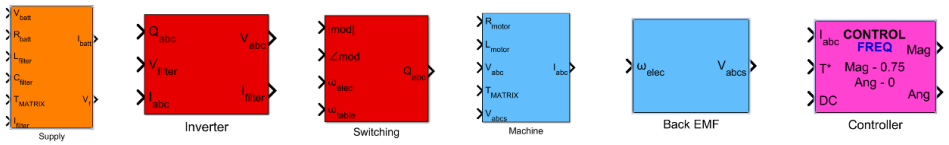


Fig 4-13 All System Model blocks overview

Supply

The “Supply” block is responsible for computing I_{batt} and V_f dynamic phasors using the equations 4.30 and 4.31. As both variables require a differentiation operation, T_{MATRIX} is an input. V_f is required in the computation of I_{batt} and this is being supplied internal to the block.

Inverter

The “Inverter” block is responsible for computing V_{abc} and I_f dynamic phasors using the equations 4.32-4.33. As these variables do not require a differentiation operation, hence T_{MATRIX} is missing as an input. There is a multiplication operation involved that is discussed later in section 4.4.3.

Switching

The “Switching” block (Fig 4-14) is responsible for producing the switching state dynamic phasors. These are Fourier approximations of the switching state square waves (1 or 0 values) as per equations 4.3.2.1-4.3.2.11. The model is equipped for any of the eleven switching strategies discussed in section 4.3.2. The switching state phasor is computed for one phase and are phase-shifted by -120° and 120° for phases B and C, respectively.

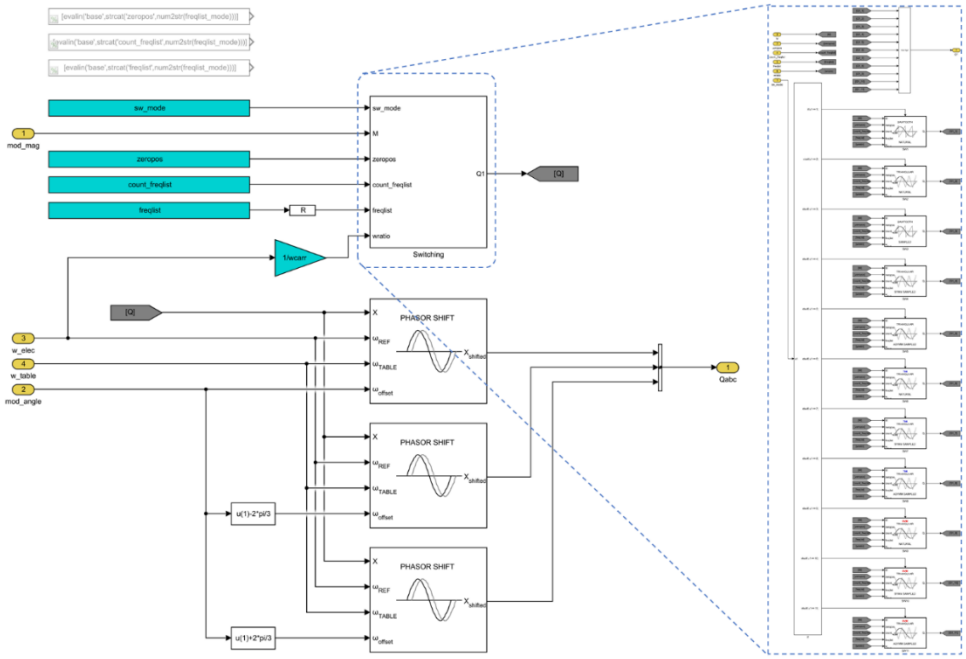


Fig 4-14 "Switching" block detailed view

Machine

The “Machine” block is responsible for computing I_{abc} dynamic phasors using the equation 4.34. As these variables do not require a differentiation operation, hence T_{MATRIX} is missing as an input. There is a multiplication operation involved that is discussed later in section 4.4.3.

Back EMF

The “Back EMF” block is responsible for computing V_{abcs} dynamic phasors using frequency of operation and back EMF constant (which is currently proportional to the frequency, but non-linearity can very easily be implemented in this block using lookup table).

Controller

The “Controller” block (Fig 4-15) is responsible for producing the demand ${I_{abc}}^*$ that is used to drive the “Switching” block. It is a standard PID controller in the d and q axes with the ${I_d}^*$ target set to 0 and ${I_q}^*$ target a proportion of the Torque demand (appropriate lookup table can be easily implemented here to address non-linear relation between quadrature current and torque of a motor). In this PhD, the controller operation

has not been investigated in detail and hence, the no feedback control has been applied, a fixed magnitude of 0.75 modulation index and 0 phase angle has been used for these experiments.

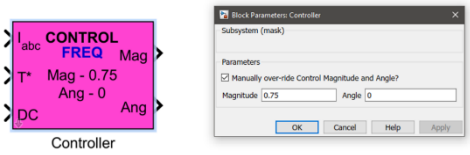


Fig 4-15 "Controller" block detailed view

4.4.3 "Modified Convolution"

The "Modified Convolution" block (Fig 4-16) is responsible for multiplying two waveforms using a novel method developed in this PhD (this was explained earlier in section 4.2.4). This is done in the following steps:

- **Initialisation** – As all frequencies are defined ahead of run-time, the exact formula (using pairs of multiplying signal phasors) for calculating the resultant phasors after multiplication is computed in the initialisation step.
- **Conversion to Exponential Form** (as defined in equations 4.9-4.11)
- **Sum of product pairs**, to produce the resultant phasors
- **Conversion back to the trigonometric form** (*SinZCos*, as defined in equations 4.9-4.11)

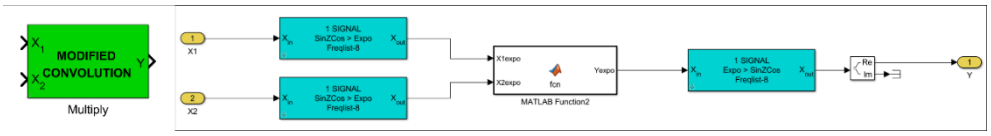


Fig 4-16 "Modified Convolution" block detailed view

One inherent issue in the MFA method of system solving, is that all the new frequencies generated as a result of multiplication operation, get lost. The solution is to include more frequencies in the original list. The initialisation step has a feature to pre-mark the "essential" frequencies and ensure that the larger set of frequencies include the new frequencies that are generated as a result of multiplication. A warning message (as shown below in Fig 4-17) is flashed if any frequency is missed.

```
!!!HARMLOST - Useful phasor (p=4, q=6) lost!!!
```

Fig 4-17 Warning message flashed for "lost harmonics" in initialisation stage

4.4.4 Phase to Time Conversion (V3B.slx)

The main component of the V3B.slx model is to convert all dynamic phasors vectors (frequency domain system variables) to time domain waveforms. This is done with the "P2T" block (shown above in Fig 4-18). This block takes the following inputs:

- Vector of dynamic phasors of a variable
- Vector of phase angles of each frequency component that is generated by the "Phase Environment SinZCos" block (described earlier in section 4.4.1).

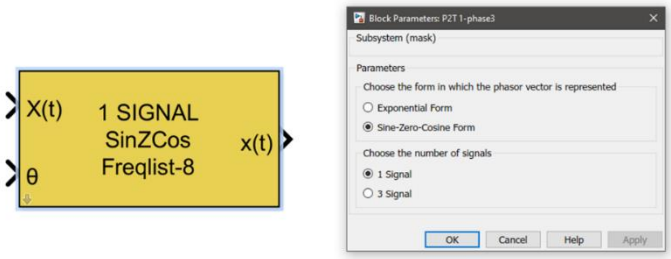


Fig 4-18 "Phase-to-Time" block detailed view

This block has been built to input phasors in both trigonometric ("Sine-Zero-Cosine") and complex exponential form.

4.5 Simulation Results (Steady State)

The MFA Modelling Suite described in the previous sub-chapter has been put in application initially for benchmarking in steady state (this sub-chapter) and eventually in variable-frequency operation (next sub-chapter). The purpose of this virtual experiment is to validate the computation time advantage that MFA promises and quantify it. This experiment also demonstrates that different fidelity levels can be achieved by including more sinusoids but at the expense of higher computation time. It shall be seen in the results that computation time and fidelity are not linearly correlated, and choosing the correct frequencies smartly is critical to reap the benefits of the MFA method.

Benchmarking simulation results have been categorised into three parts:

- Total simulation run-time for comparison against conventional time-domain switching model V3C.slx (Simscape Specialised Power Systems™). This is described in 4.5.2.
- Time domain waveforms of all physical variables produced by MFA model V3A.slx has been compared with waveforms produced from V3C.slx. The dynamic phasors (frequency domain equivalents of time-domain signals) produced by MFA model has been converted to time-domain waveforms with the V3B.slx model. This has been described in 4.5.3.
- Dynamic Phasors produced by MFA model has been compared with the equivalent waveforms from V3C.slx model. The time-domain waveforms from V3C.slx have been converted to phasors using the Fast Fourier Transform (FFT) tool in MATLAB™ offline. This is described in 4.5.4.

4.5.1 Simulation Setup

The physical parameters in the model (resistance, inductance etc.) have been set as follows:

- DC Side and Input Filter:
    - Battery Voltage – 100V
    - Battery Internal Resistance – 10Ω
    - Input filter Inductance – 10mH
    - Input filter Capacitance (DC Bus) – 500μF
- AC side (motor):
    - Phase Resistance – 10Ω
    - Phase Inductance – 10mH
    - Back EMF Constant – 10V/kRPM
- All initialisation voltage and current from 0.
- Carrier Frequency – 20kHz
- Fixed Fundamental Frequency (steady state) – 50Hz
- Simulation time-step:

- V3A.slx – 0.5ms (2kHz)
- V3B.slx and V3C.slx – 0.5µs (2MHz)

Switching Strategies

The Sine PWM and Third Harmonic Injection switching strategies have been investigated in this discussion. Only asymmetric regular sampling was chosen to be depicted in this chapter due to space constraints, although all three sampling methods for SPWM and THI strategies showed comparable performance. SVM has not been investigated due to reason mentioned earlier (section 4.3.2.12).

- Sine PWM (SPM)
  - Natural Sampling
  - Asymmetric Regular Sampling
- Sine PWM with Third Harmonic Injection (THI)
  - Natural Sampling
  - Asymmetric Regular Sampling

Frequencies

The MFA model has been iterated with four sets of computed frequencies, each with an increased number and consequently increased fidelity and computational time/complexity:

- MFA1 – Only fundamental frequency (and third harmonic for THI and SVM).
- MFA2 – MFA1 and carrier frequency.
- MFA3 – MFA2, second carrier frequency, and some top sidebands.
- MFA4 – MFA3 and all additional frequencies produced due to multiplication operation.

The above have been summarised in the Table 4-1 below:

p	***q***	**MFA1**	**MFA2**	**MFA3**	**MFA4**	***Remark***
0	0	✓	✓	✓	✓	DC
0	1	✓	✓	✓	✓	Fundamental
0	2				✓	

0	3	✓	✓	✓	✓	Third Harmonic
0	4				✓	
0	6				✓	
1	0		✓	✓	✓	First Carrier
1	1				✓	
1	2			✓	✓	
1	3				✓	
1	4			✓	✓	
1	5				✓	
1	7				✓	
2	0				✓	Second Carrier
2	1			✓	✓	
2	2				✓	
2	3			✓	✓	
2	4				✓	
2	6				✓	
2	8				✓	
3	0				✓	Third Carrier
3	1				✓	
3	2				✓	
3	3				✓	
3	4				✓	
3	5				✓	
3	6				✓	
4	0				✓	Fourth Carrier
4	1				✓	
4	2				✓	
4	3				✓	
4	4				✓	
4	5				✓	

Table 4-3 List of all harmonics computed in the four MFA models

4.5.2 Simulation Run-Time Comparison

Simulation run-time has been computed by running each model ten times in succession and finding the average value (Fig 4-19). The values are summarised in Table 4-4.

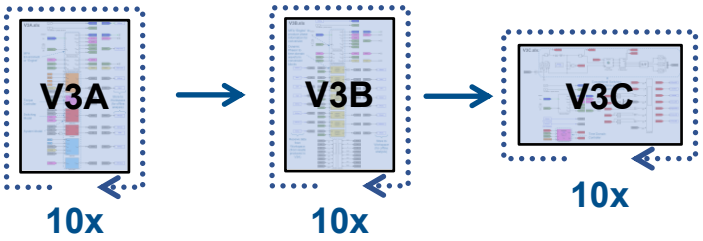


Fig 4-19 Simulation run-time comparison experimental plan

	Strategy	Sinusoidal PWM			Third Harmonic Injection		
	Carrier Wave	Triangular			Triangular		
	Sampling	Natural	Regular		Natural	Regular	
			Symmetric	Asymmetric		Symmetric	Asymmetric
		(s)	(s)	(s)	(s)	(s)	(s)
V3A.slx	*MFA1*	0.20	0.20	0.21	0.20	0.21	0.20
	MFA2	0.23	0.24	0.23	0.25	0.24	0.26
	MFA3	1.11	1.07	1.16	1.17	1.14	1.16
	MFA4	26.55	27.26	26.35	26.17	26.52	26.73
V3B.slx	*MFA1*	9.14	9.61	9.45	9.50	9.50	9.57
	MFA2	9.92	10.03	9.82	9.61	9.73	9.82
	MFA3	12.49	12.42	12.52	12.23	12.14	12.25
	MFA4	24.54	24.42	24.57	24.49	24.21	24.00
V3C.slx	*Benchmark*	18.51	18.45	18.43	18.40	18.26	18.26

Table 4-4 Simulation execution time of MFA models and equivalent time-domain switching model

It is clear that MFA1, MFA2, and MFA3 are computed significantly quicker than the equivalent benchmark model (MFA4 is a special case with many extra harmonics to produce accurate switching waveform). The MFA1 and MFA2 (fundamental waveforms and carrier frequency) is less than 1% computationally expensive of the benchmark model and MFA3 is about 6% of its equivalent model (summary in Table 4-5 below).

	Strategy	Sinusoidal PWM			Third Harmonic Injection		
	Carrier Wave	Triangular			Triangular		
	Sampling	Natural	Regular		Natural	Regular	
			Symmetric	Asymmetric		Symmetric	Asymmetric
		(%)	(%)	(%)	(%)	(%)	(%)
V3A.slx	*MFA1*	1%	1%	1%	1%	1%	1%
	MFA2	1%	1%	1%	1%	1%	1%
	MFA3	6%	6%	6%	6%	6%	6%
	MFA4	143%	148%	143%	142%	145%	146%

Table 4-5 Simulation execution time comparison of MFA models with equivalent time-domain switching model

4.5.3 Waveforms Comparison (Time domain)

MFA model results have been compared against the time-domain switching model for the following waveforms:

- Switching State (q_{abc})
- Phase Current (i_{abc})
- Phase Voltage (v_{abc})
- Battery Current (i_{batt})
- DC Bus Current (i_f)
- DC Bus Voltage (v_f)

Switching State (Inverter Gates)

Simulation results for the switching state waveform in time-domain (i.e., q_{abc}) are shown below in Fig 4-20 (Sine PWM) and Fig 4-21 (Third Harmonic Injection). The structure is as follows:

- Two columns of graphs, (a)-(d) and (e)-(h), depict Natural and Asymmetric Regular Sampling, respectively.
- Top row of graphs, (a) and (e), are the overview spanning the entire duration (0.08s to 0.1s). Simulation up to 0.08s was allowed to run for the system to attain steady state.
- Bottom graphs (separately for three phases), (b)-(d) and (f)-(h), are zoomed-in views of the waveforms at 0.0865s to 0.0867s.

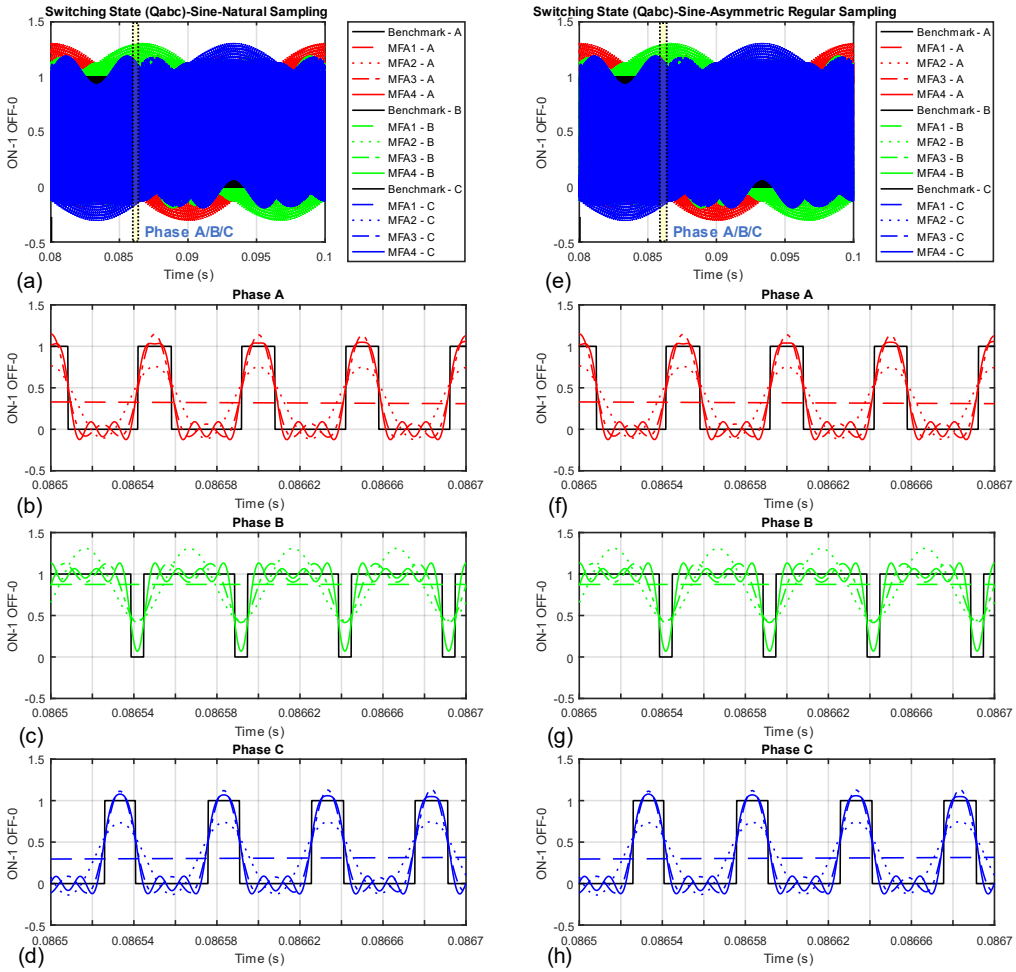


Fig 4-20 Time-domain waveform comparison (q_{abc}) for Natural and Asymmetric Regular Sampling (Sine PWM switching strategy)

Fig 4-20 above and Fig 4-21 below show remarkable reconciliation between the benchmark (black square waves) and different MFA model outputs (different dashed lines in red, blue, and green). It can be clearly observed that the higher MFA models are following the square form of the benchmark more closely which is as per expectation.

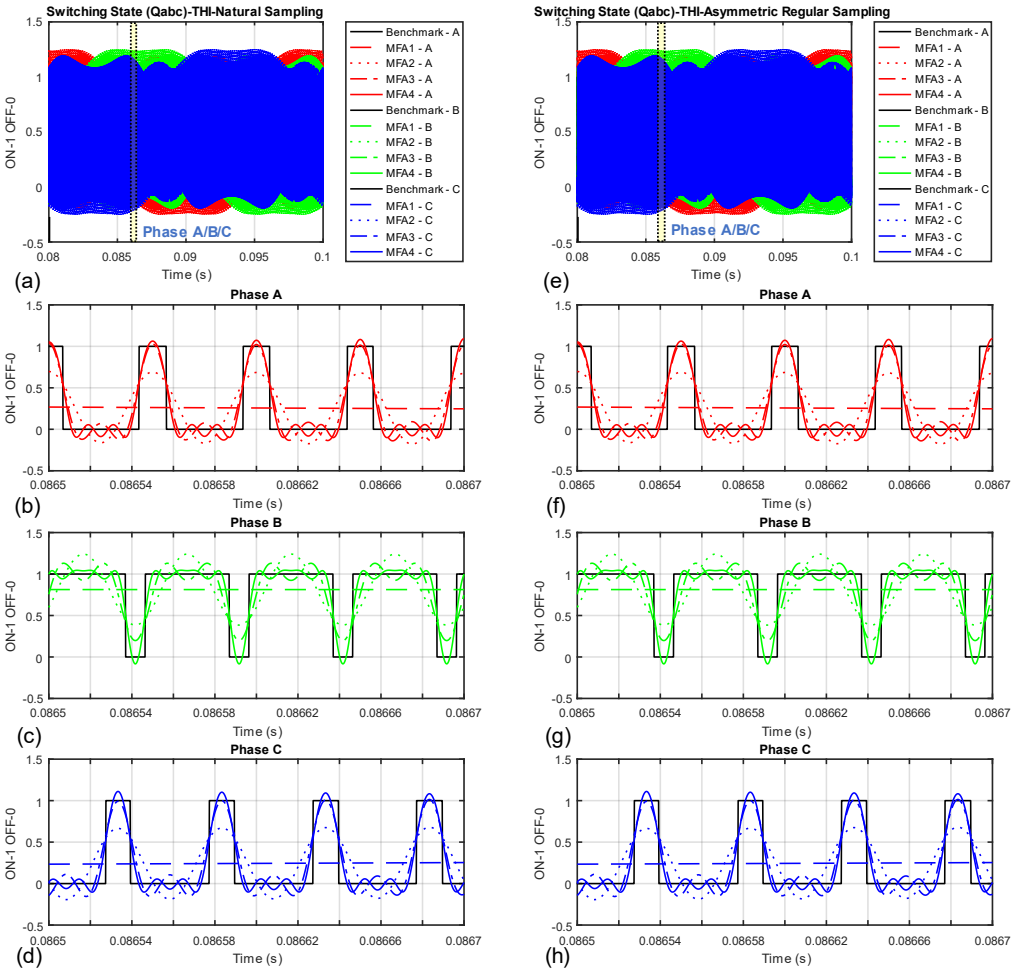


Fig 4-21 Time-domain waveform comparison (q_{abc}) for Natural and Asymmetric Regular Sampling (third harmonic injection switching strategy)

Phase Current

Simulation results for the phase current waveform in time-domain (i.e., i_{abc}) are shown below in Fig 4-22 (Sine PWM) and Fig 4-23 (Third Harmonic Injection). The structure followed is identical to the previous waveform.

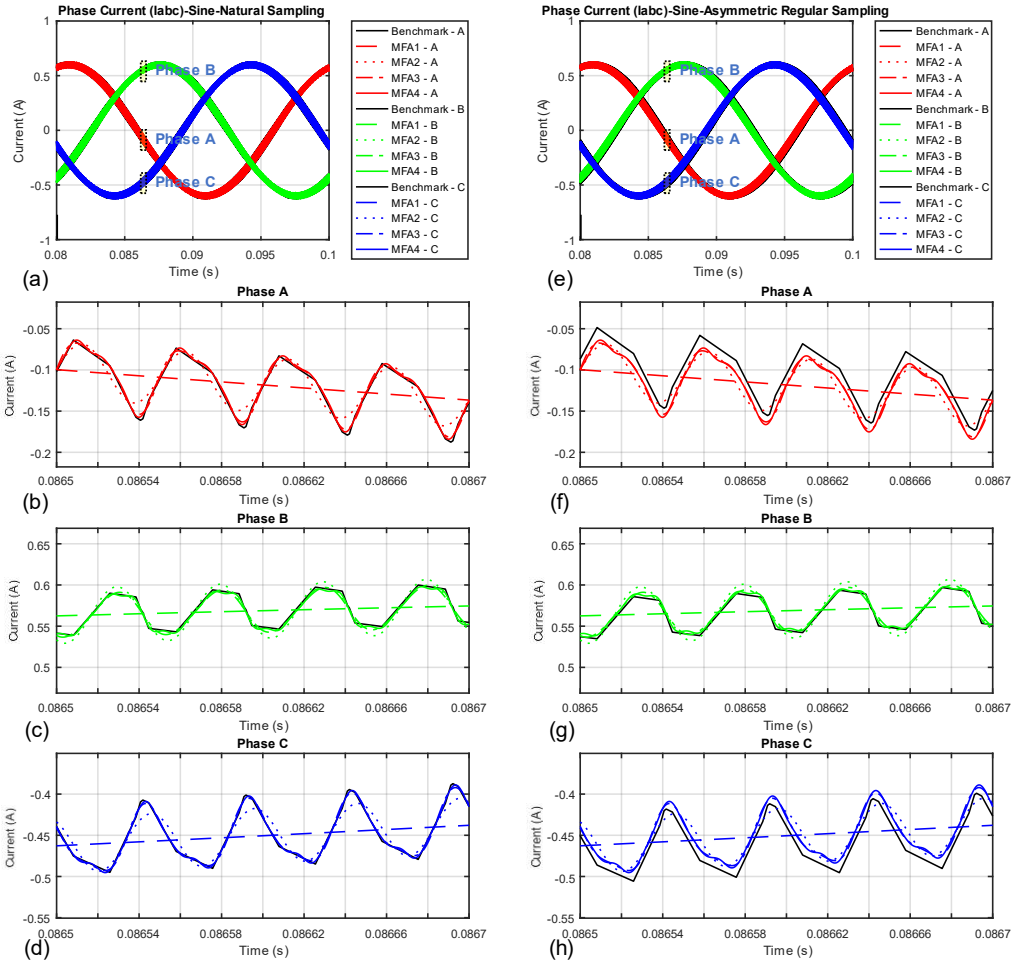


Fig 4-22 Time-domain waveform comparison (i_{abc}) for Natural and Asymmetric Regular Sampling (Sine PWM switching strategy)

The MFA model results are closely following the benchmark model waveforms for all four switching scenarios. The fours levels of MFA approximation (i.e., MFA1 to MFA4) show increasingly closer reconciliation with the benchmark waveform. MFA1 is the average behaviour with no switching components observed.

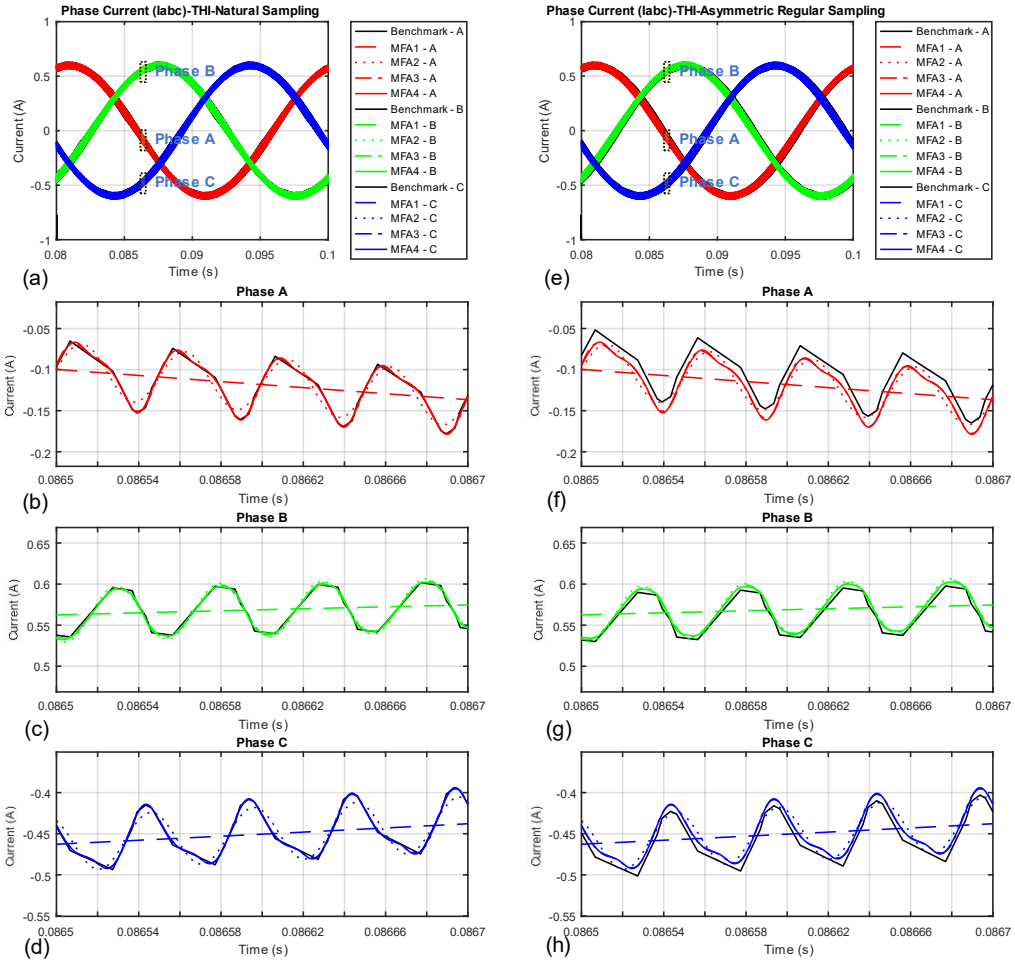


Fig 4-23 Time-domain waveform comparison (i_{abc}) for Natural and Asymmetric Regular Sampling (third harmonic injection switching strategy)

One feature that sticks out is that for the regular sampled experiment graphs, there is a slight mismatch in the average current amplitude. This is more noticeable in phases A and C, less so in phase B, which indicates that this deviation is a function of the phase angle of the wave. This deviation is minimal as can be seen in the summary view (part (e)) and hence has not been investigated further.

Phase Voltage

Simulation results for the phase voltage waveform in time-domain (i.e., v_{abc}) are shown below in Fig 4-24 (Sine PWM) and Fig 4-25 (Third Harmonic Injection). The structure followed is identical to the previous waveform.

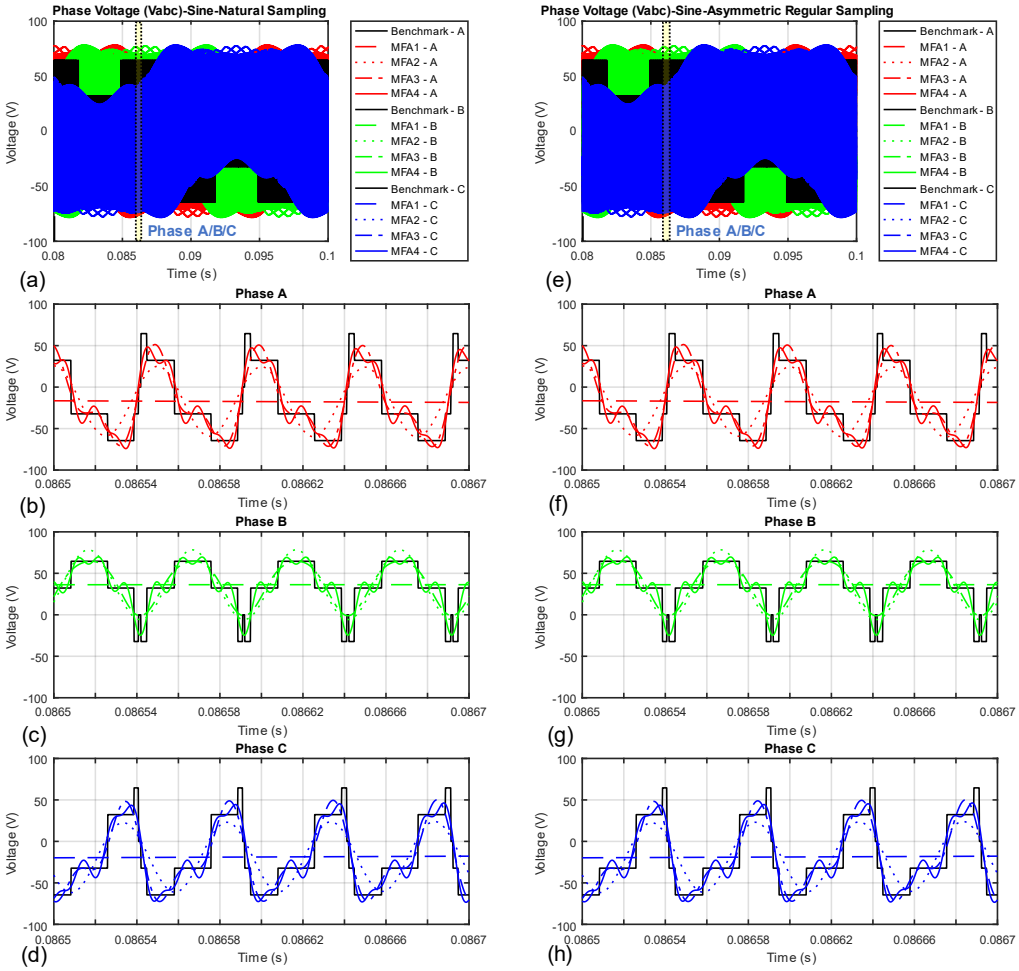


Fig 4-24 Time-domain waveform comparison (v_{abc}) for Natural and Asymmetric Regular Sampling (Sine PWM switching strategy)

The phase voltage outputs for the three legs can be observed here. As expected, the different levels of MFA model output reconcile with the benchmark waveform in increasing degrees of closeness.

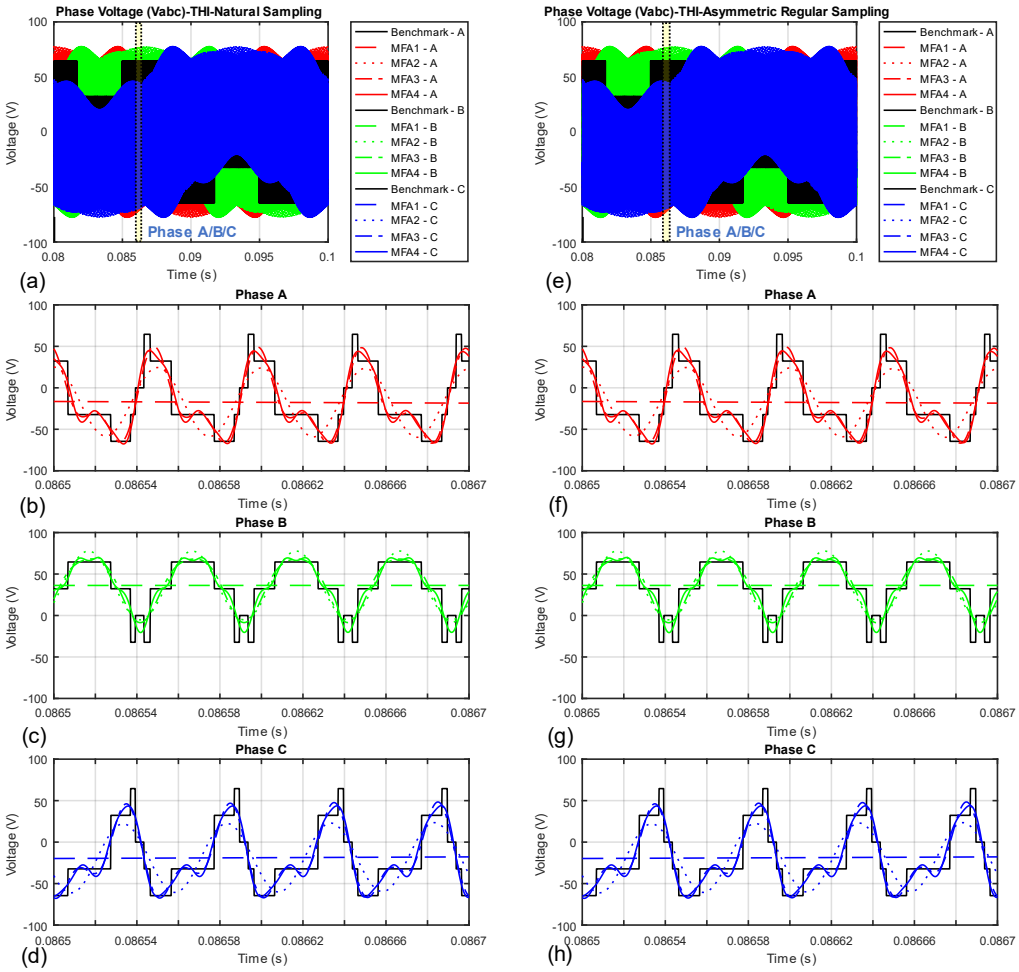


Fig 4-25 Time-domain waveform comparison (v_{abc}) for Natural and Asymmetric Regular Sampling (third harmonic injection switching strategy)

Battery Current

Simulation results for the battery current waveform in time-domain (i.e., i_{batt}) are shown below in Fig 4-26 (Sine PWM) and Fig 4-27 (Third Harmonic Injection). The structure is as follows:

- Two columns of graphs, (a)-(b) and (c)-(d), depict Natural and Asymmetric Regular Sampling, respectively.
- Top row of graphs, (a) and (c), are the overview spanning the entire duration (0.08s to 0.1s). Simulation up to 0.08s was allowed to run for the system to attain steady state.
- Bottom row of graphs, (b) and (d), are zoomed-in views of the waveforms at 0.0865s to 0.0867s.

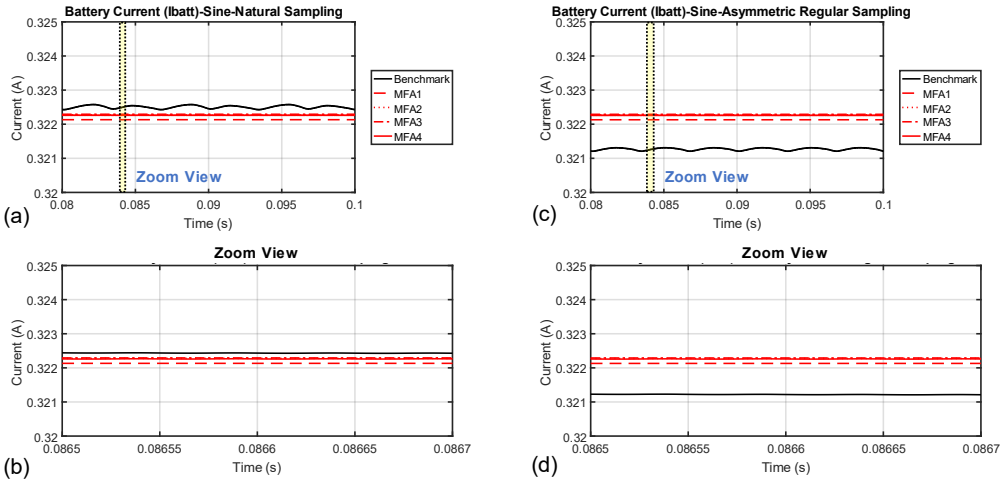


Fig 4-26 Time-domain waveform comparison (i_{batt}) for Natural and Asymmetric Regular Sampling (Sine PWM switching strategy)

The battery current waveforms produced by MFA model are closely following the benchmark model waveforms. A small DC value offset is observed in all four graphs. Interestingly, the deviation is higher for the Naturally Sampled simulations in the case of third harmonic injection, although the phase current deviations were close to benchmark. There is also a small but observable amount of low frequency AC “hum” that the MFA model is unable to catch.

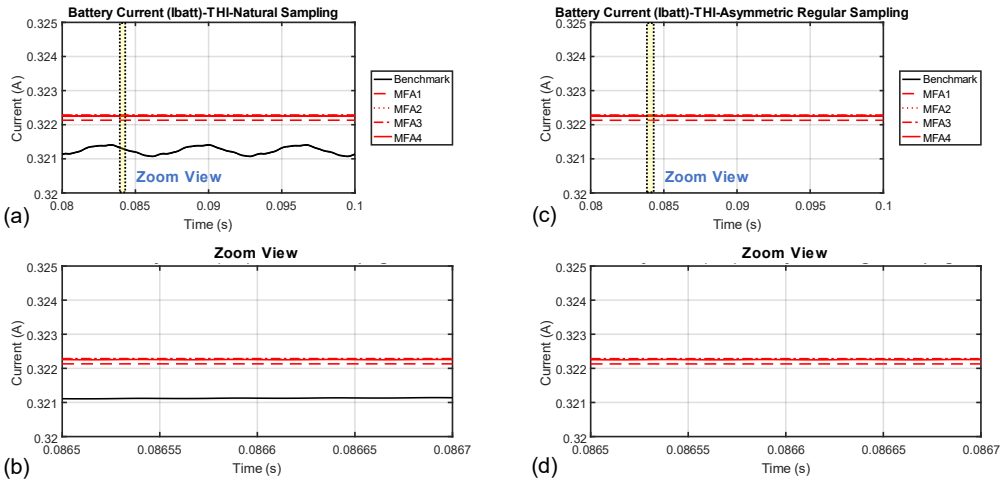


Fig 4-27 Time-domain waveform comparison (i_{batt}) for Natural and Asymmetric Regular Sampling (third harmonic injection switching strategy)

DC Bus Current (Input filter Current)

Simulation results for the DC bus current waveform in time-domain (i.e., i_f) are shown below in Fig 4-28 (Sine PWM) and Fig 4-29 (Third Harmonic Injection). The structure followed is identical to the previous waveform.

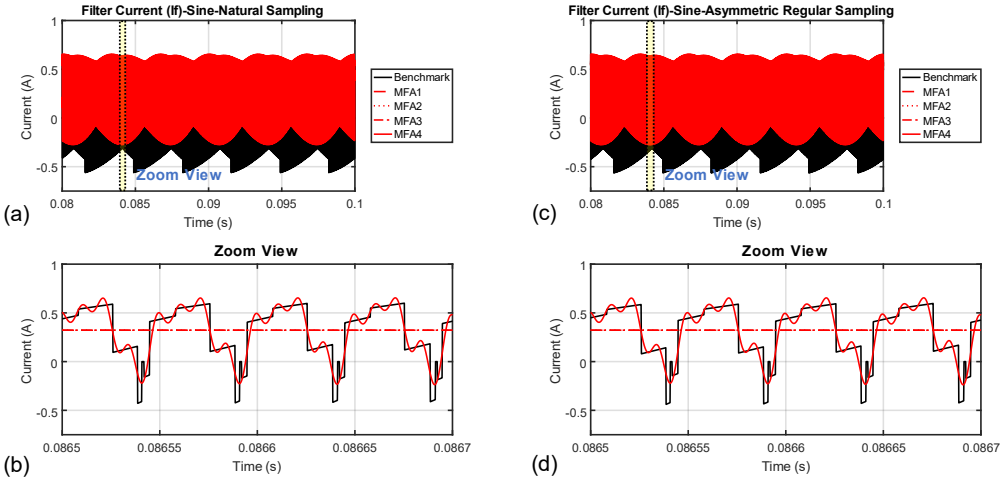


Fig 4-28 Time-domain waveform comparison (i_f) for Natural and Asymmetric Regular Sampling (Sine PWM switching strategy)

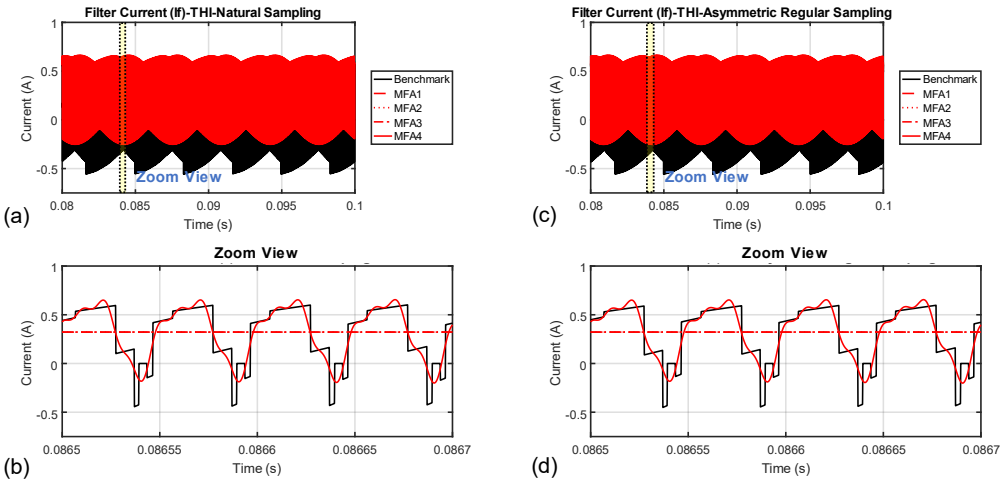


Fig 4-29 Time-domain waveform comparison (i_f) for Natural and Asymmetric Regular Sampling (third harmonic injection switching strategy)

The filter current graph shows an interesting feature. The first three MFA levels only demonstrate the DC component (or average value) of the benchmark waveform, and only MFA4 can capture the switching behaviour. This is a result of the multiplication block that was used in the initialisation stage (detailed explanation given earlier in 4.2.4 and 4.4.3) to identify which harmonics would be produced as a result of multiplying the switching harmonics and sidebands.

DC Bus Voltage (Input filter Voltage)

Simulation results for the DC bus voltage waveform in time-domain (i.e., v_f) are shown below in Fig 4-30 (Sine PWM) and Fig 4-31 (Third Harmonic Injection). The structure followed is identical to the previous waveform.

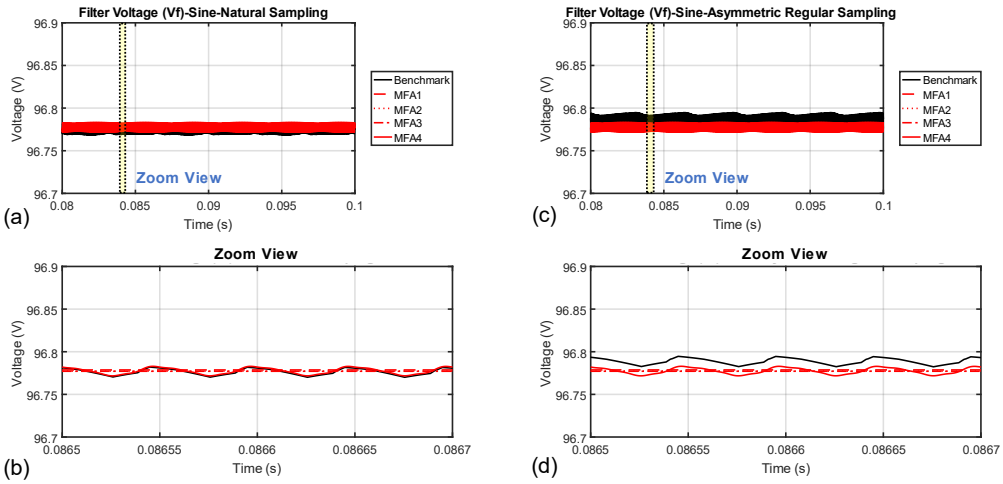


Fig 4-30 Time-domain waveform comparison (v_f) for Natural and Asymmetric Regular Sampling (Sine PWM switching strategy)

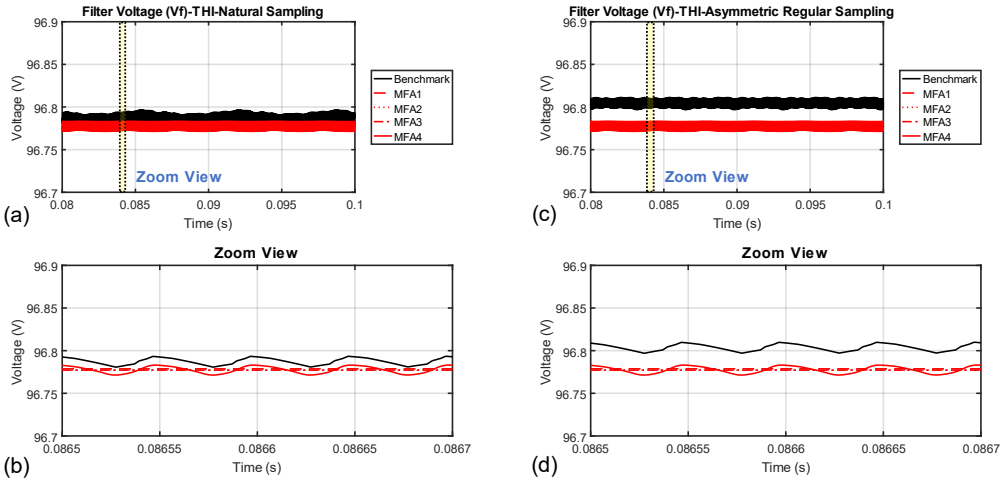


Fig 4-31 Time-domain waveform comparison (v_f) for Natural and Asymmetric Regular Sampling (third harmonic injection switching strategy)

The filter voltage graph shows comparable results to battery current. There is a small but distinct low frequency “hum” in the benchmark waveform that is not getting picked by the MFA model. There is a DC offset error observed in all the simulations except

Naturally Sampled Sine PWM switching, but the amount is miniscule (<0.5%) and hence not of significant concern.

4.5.4 Waveforms Comparison (Frequency Domain)

Identically to previous section, the MFA model results have been compared against the time-domain switching model for the following waveforms, in the frequency domain. The Fast Fourier Transform (FFT) method was used to produce phasor quantities for the benchmark results.

- Switching State (Q_{abc})
- Phase Current (I_{abc})
- Phase Voltage (V_{abc})
- Battery Current (I_{batt})
- DC Bus Current (I_f)
- DC Bus Voltage (V_f)

Switching State (Inverter Gates)

Simulation results for the switching state frequency spectrum (i.e., Q_{abc}) are shown below in figures Fig 4-32 (Sine PWM) and Fig 4-33 (Third Harmonic Injection). The structure is as follows:

- Two columns of graphs, (a)-(f) and (g)-(l), depict Natural and Asymmetric Regular Sampling, respectively.
- Top row of graphs, (a) and (g), are the overview spanning the entire spectrum (DC to 100 kHz) that includes up to 5 harmonics of the carrier frequency.
- Bottom graphs, (b)-(f) and (g)-(l), are zoomed-in views of the spectrum at:
    - DC region ((b) and (g), 0-400 Hz),
    - 1st carrier harmonic ((b) and (g), 20 kHz ±400 Hz),
    - 2nd carrier harmonic ((b) and (g), 40 kHz ±400 Hz),
    - 3rd carrier harmonic ((b) and (g), 60 kHz ±400 Hz),
    - 4th carrier harmonic ((b) and (g), 80 kHz ±400 Hz).

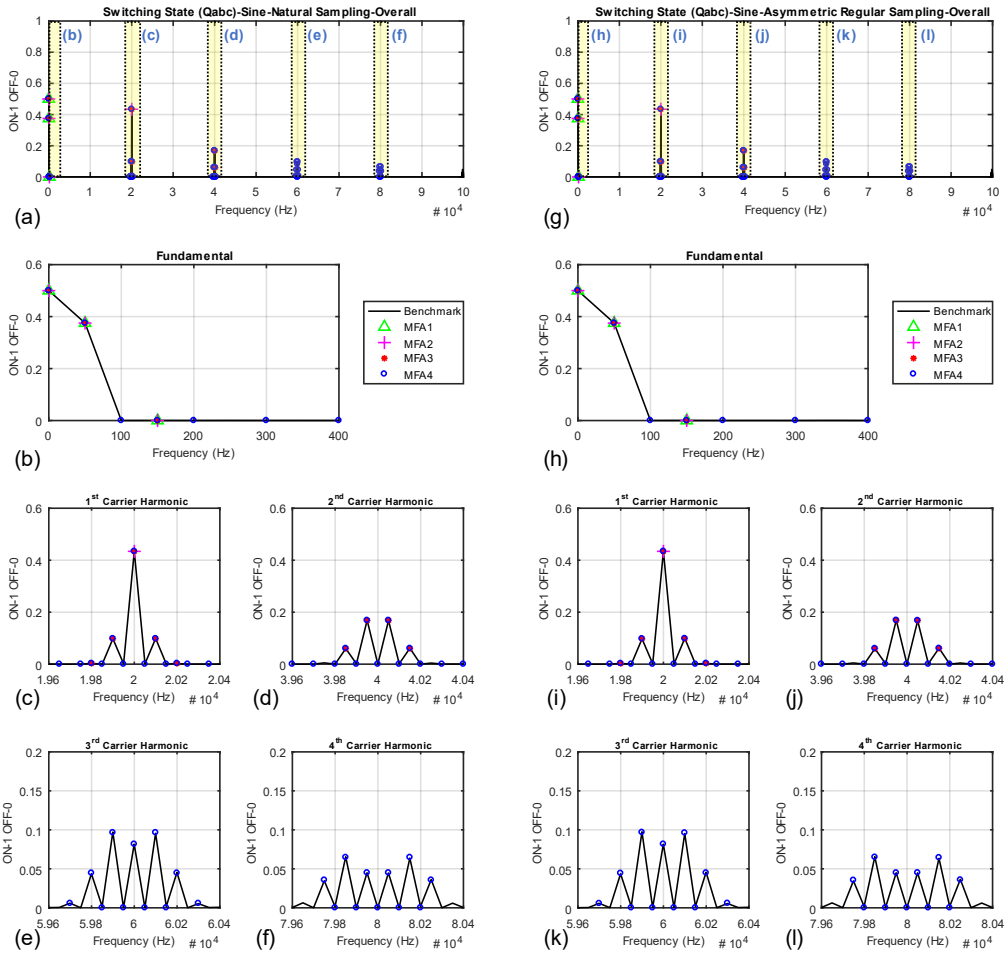


Fig 4-32 Frequency spectrum comparison (Q_{abc}) for Natural and Asymmetric Regular Sampling (sine PWM switching strategy)

The graph above (Sine PWM switching) shows good reconciliation between different MFA outputs and the FFT of benchmark waveforms. Fig 4-33 below (third harmonic injection) shows that the 3rd harmonic of the carrier frequency is underpredicted by approximately 20%. The frequency domain equations of the switching state waveforms were referred directly from verified literature (section 4.3.2), the efficacy of these equations has been proposed as future work after this PhD.

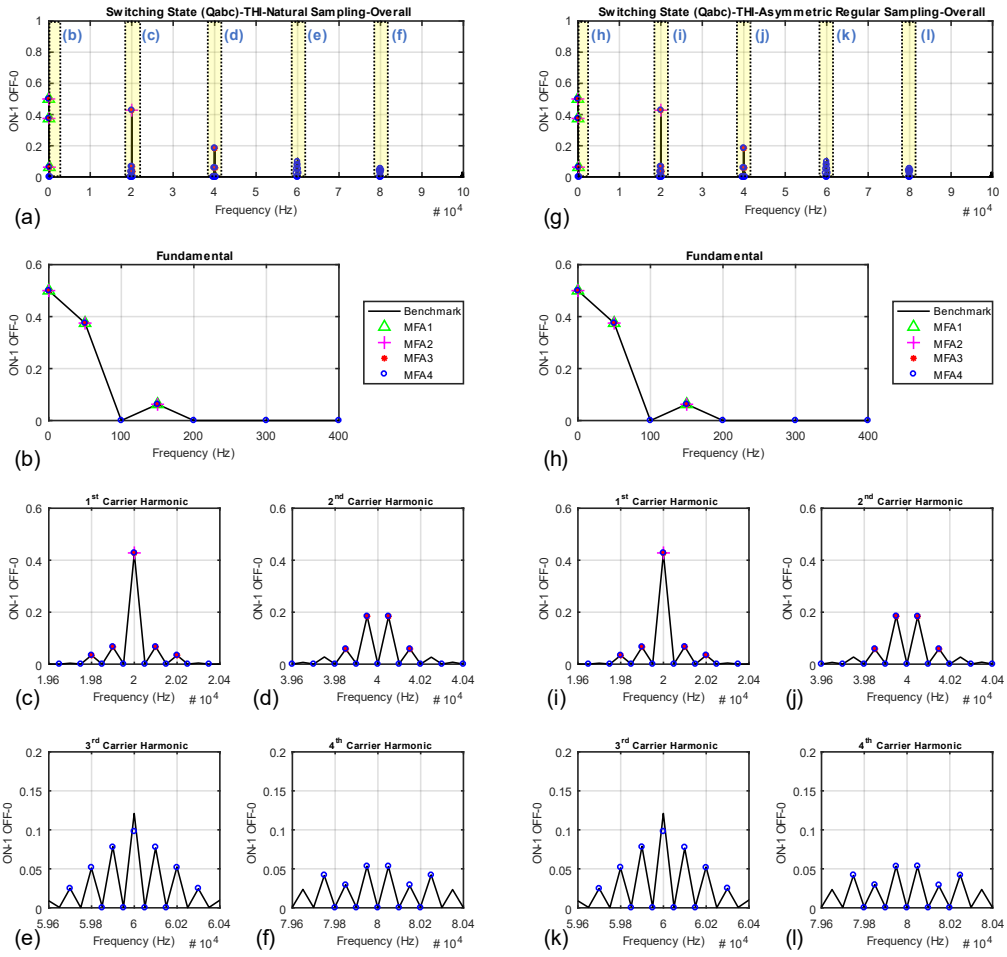


Fig 4-33 Frequency spectrum comparison (Q_{abc}) for Natural and Asymmetric Regular Sampling (third harmonic injection switching strategy)

Phase Current

Simulation results for the phase current frequency spectrum (i.e., I_{abc}) are shown below in figures Fig 4-34 (Sine PWM) and Fig 4-35 (Third Harmonic Injection). The structure followed is identical to the previous waveform.

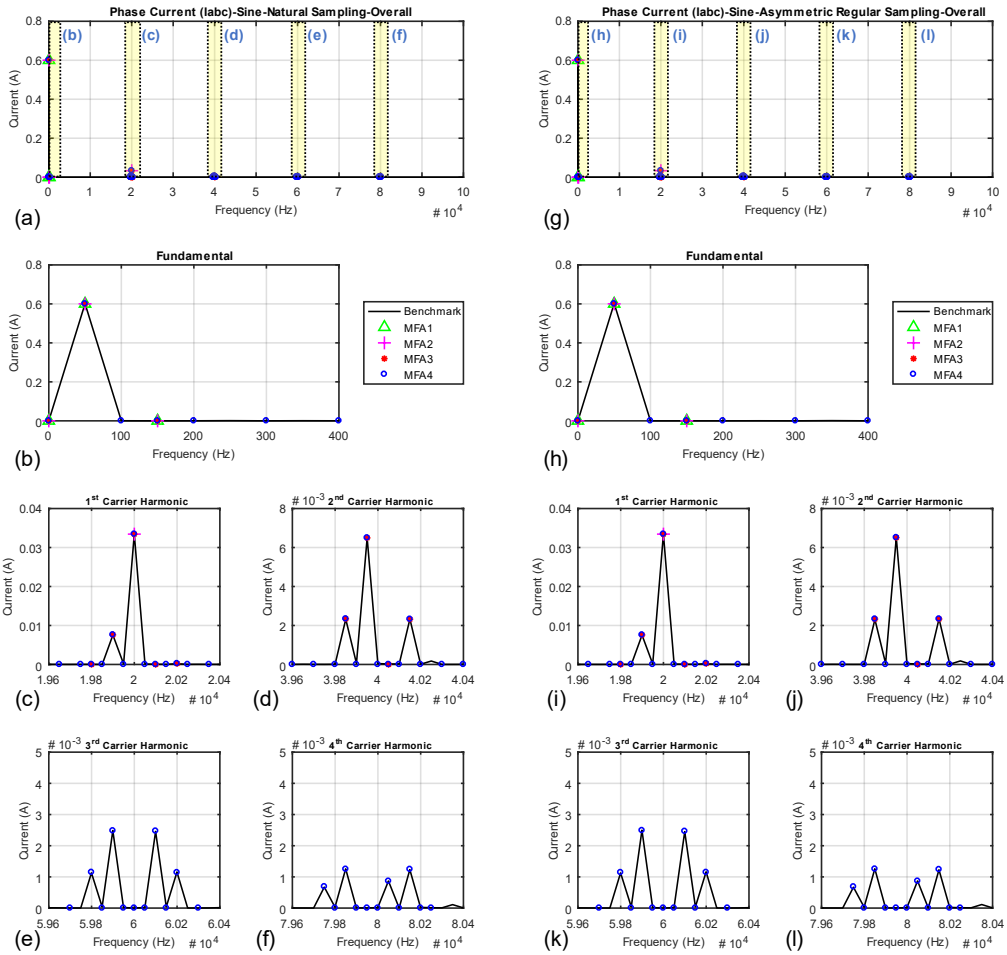


Fig 4-34 Frequency spectrum comparison (I_{abc}) for Natural and Asymmetric Regular Sampling (sine PWM switching strategy)

The two graphs above and below show good reconciliation between different MFA outputs and the FFT of benchmark waveforms. It is observed that some sidebands have been missed (e.g., 6th sideband to 2nd switching harmonic, or 40.3 kHz). This is an inherent weakness of the MFA model, unfortunately, i.e., there would always be some frequencies that get missed.

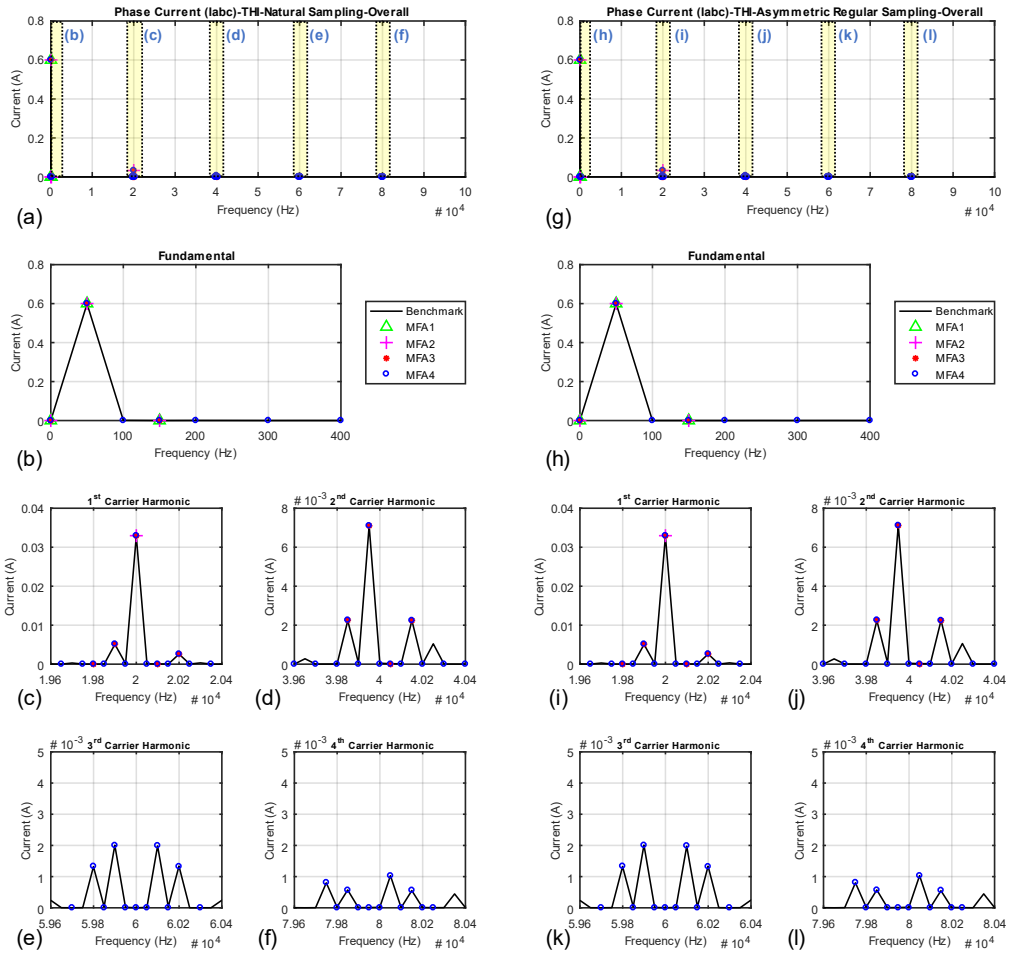


Fig 4-35 Frequency spectrum comparison (I_{abc}) for Natural and Asymmetric Regular Sampling (third harmonic injection switching strategy)

Phase Voltage

Simulation results for the phase voltage frequency spectrum (i.e., V_{abc}) are shown below in figures Fig 4-36 (Sine PWM) and Fig 4-37 (Third Harmonic Injection). The structure followed is identical to the previous waveform.

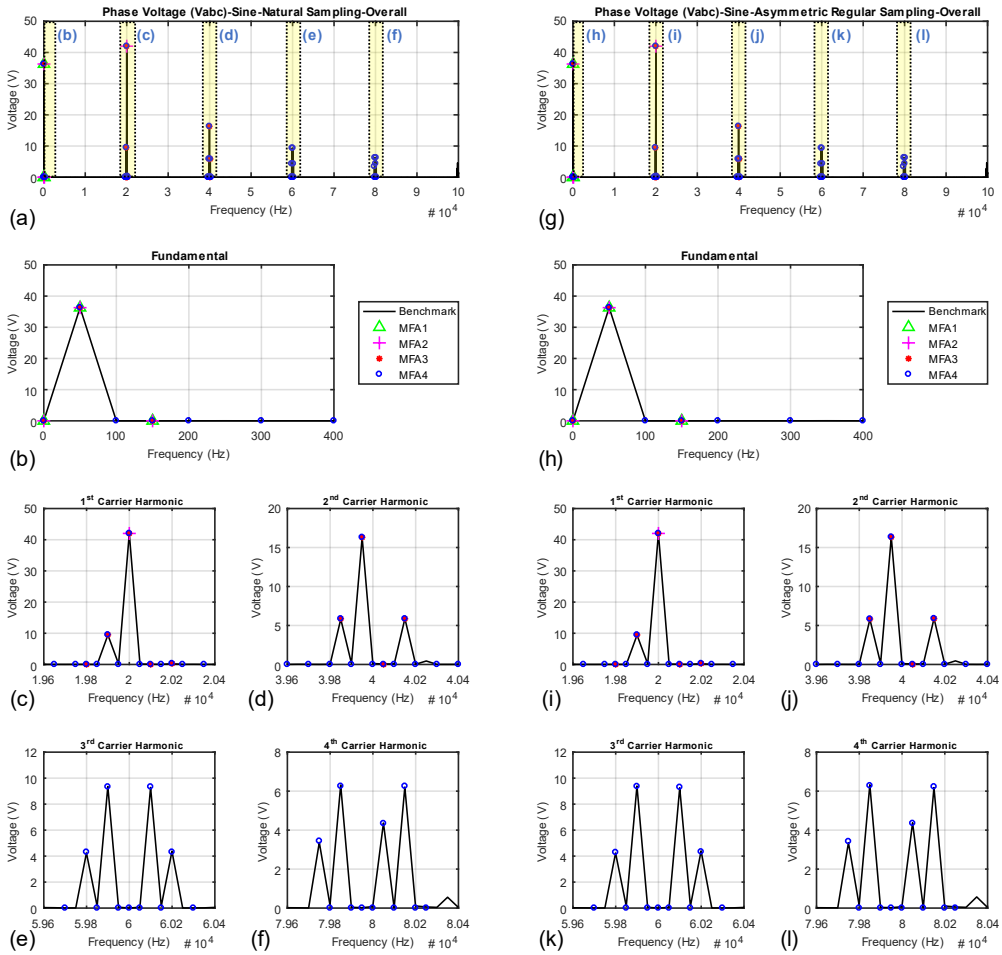


Fig 4-36 Frequency spectrum comparison (V_{abc}) for Natural and Asymmetric Regular Sampling (sine PWM switching strategy)

The two graphs above and below show good reconciliation between different MFA outputs and the FFT of benchmark waveforms. Most of the carrier harmonics and sidebands show minute overprediction (<1%) which is deemed inconsequential at this stage of development.

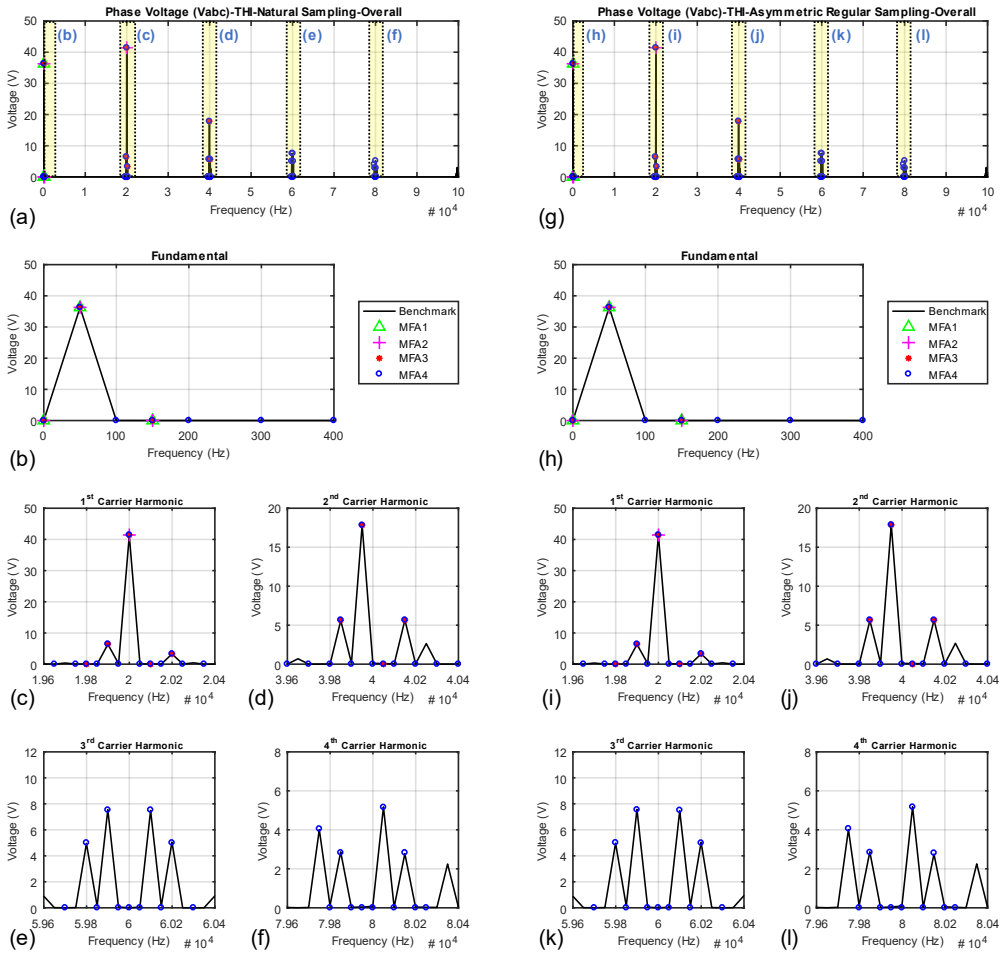


Fig 4-37 Frequency spectrum comparison (V_{abc}) for Natural and Asymmetric Regular Sampling (third harmonic injection switching strategy)

Battery Current

Simulation results for the battery current frequency spectrum (i.e., I_{batt}) are shown below in figures Fig 4-38 (Sine PWM) and Fig 4-39 (Third Harmonic Injection). The structure followed is identical to the previous waveform.

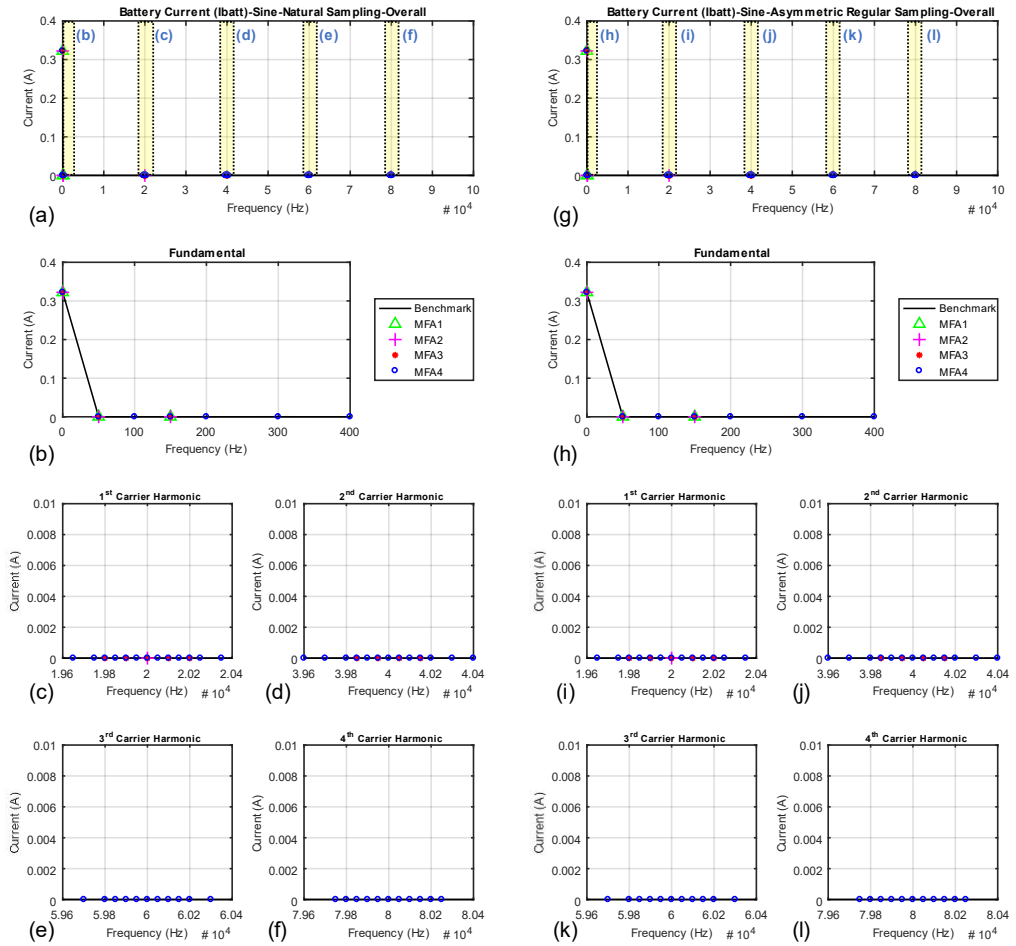


Fig 4-38 Frequency spectrum comparison (I_{batt}) for Natural and Asymmetric Regular Sampling (sine PWM switching strategy)

The two graphs above and below show good reconciliation between different MFA outputs and the FFT of benchmark waveforms. It is expected to have no AC values at all, and only DC component is visible.

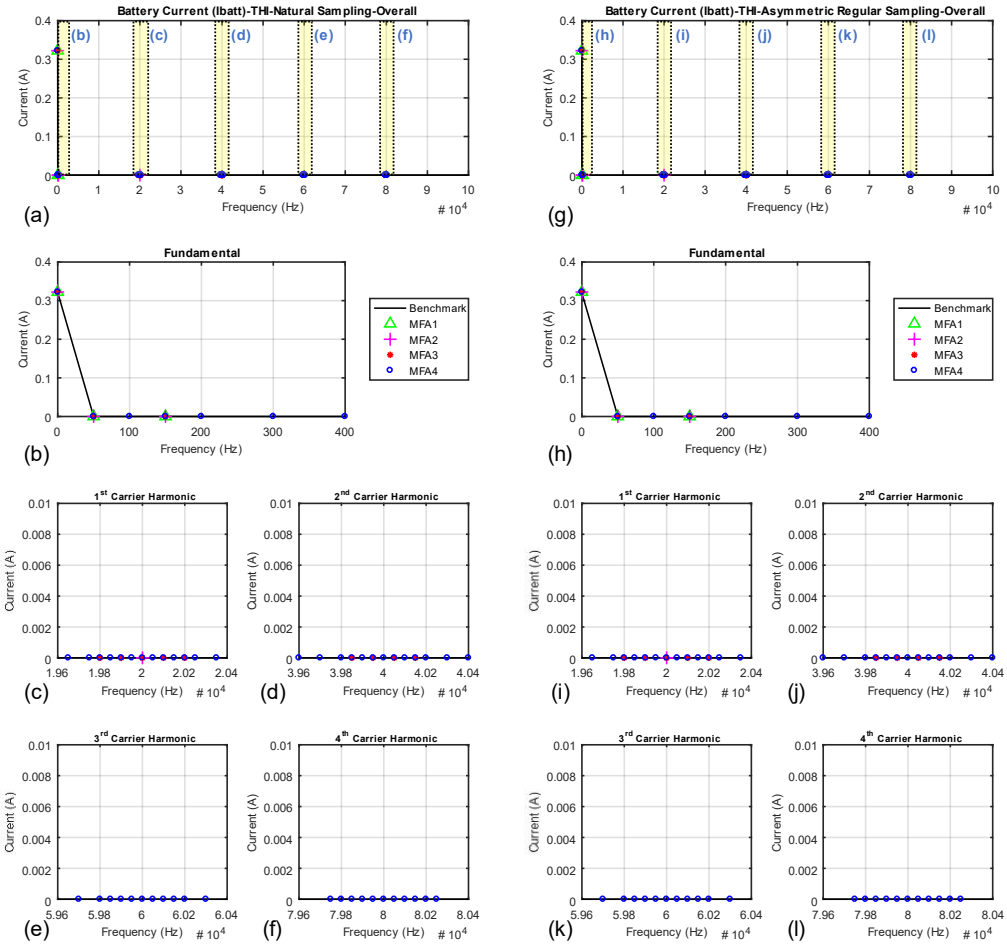


Fig 4-39 Frequency spectrum comparison (I_{batt}) for Natural and Asymmetric Regular Sampling (third harmonic injection switching strategy)

DC Bus Current (Input filter Current)

Simulation results for the DC bus current frequency spectrum (i.e., I_f) are shown below in figures Fig 4-40 (Sine PWM) and Fig 4-41 (Third Harmonic Injection). The structure followed is identical to the previous waveform.

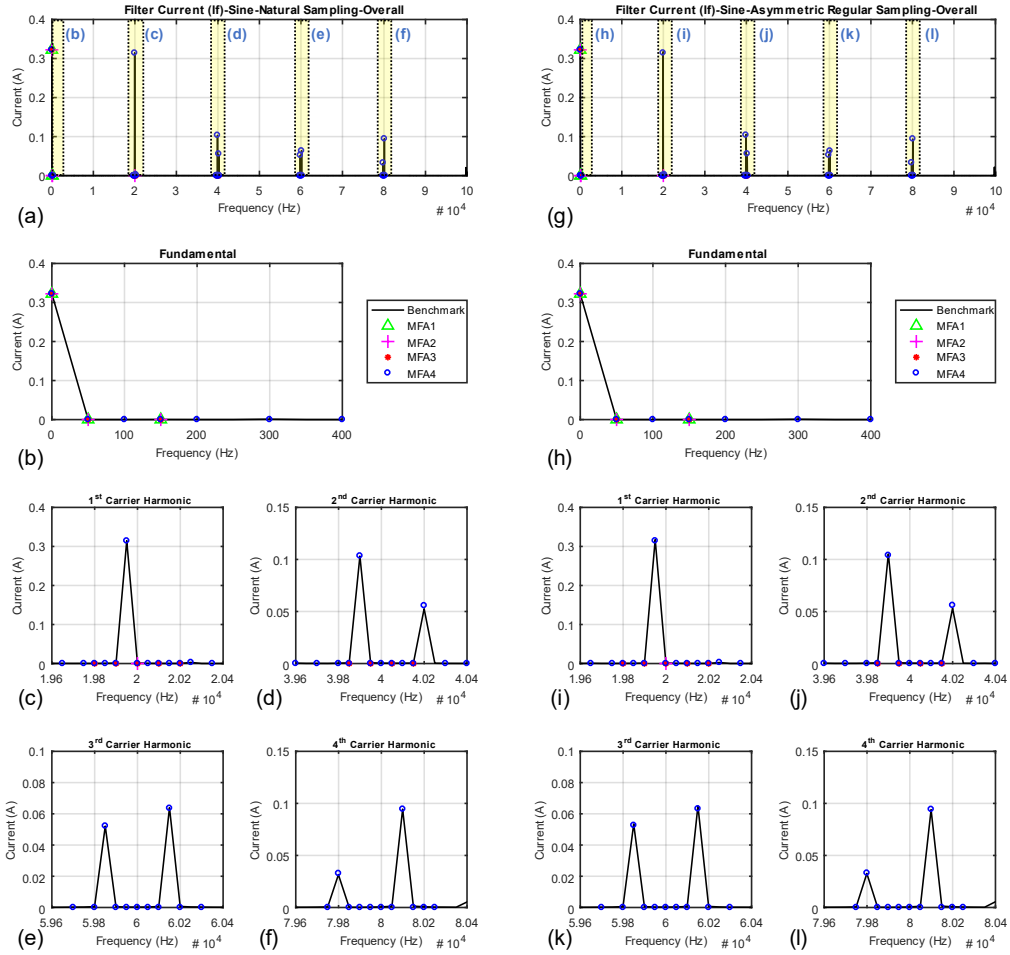


Fig 4-40 Frequency spectrum comparison (I_f) for Natural and Asymmetric Regular Sampling (sine PWM switching strategy)

The two graphs above and below show good reconciliation between different MFA outputs and the FFT of benchmark waveforms. The 4th sideband to second carrier harmonic shows slight overprediction for all four simulation experiments, but is particularly pronounced for the third harmonic injection experiments (approximately 25%).

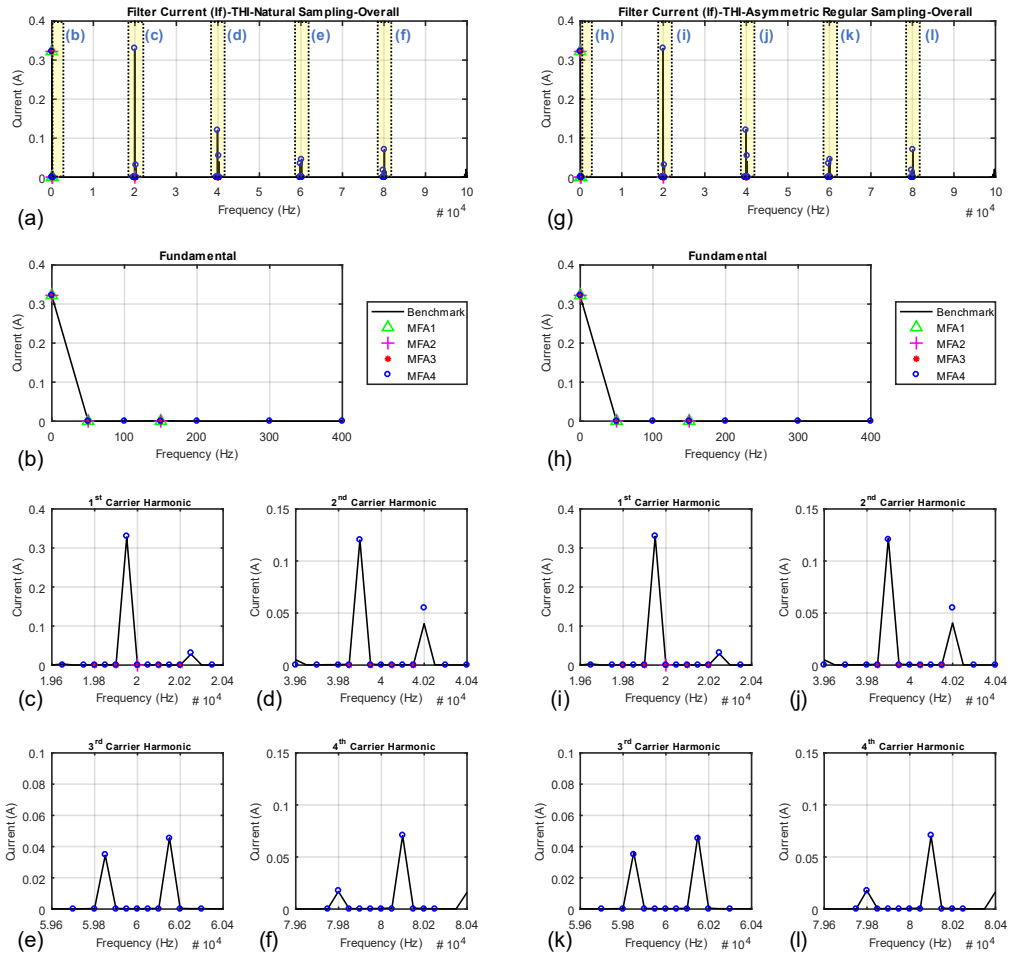


Fig 4-41 Frequency spectrum comparison (I_f) for Natural and Asymmetric Regular Sampling (third harmonic injection switching strategy)

DC Bus Voltage (Input filter Voltage)

Simulation results for the DC bus voltage frequency spectrum (i.e., V_f) are shown below in figures Fig 4-42 (Sine PWM) and Fig 4-43 (Third Harmonic Injection). The structure followed is identical to the previous waveform.

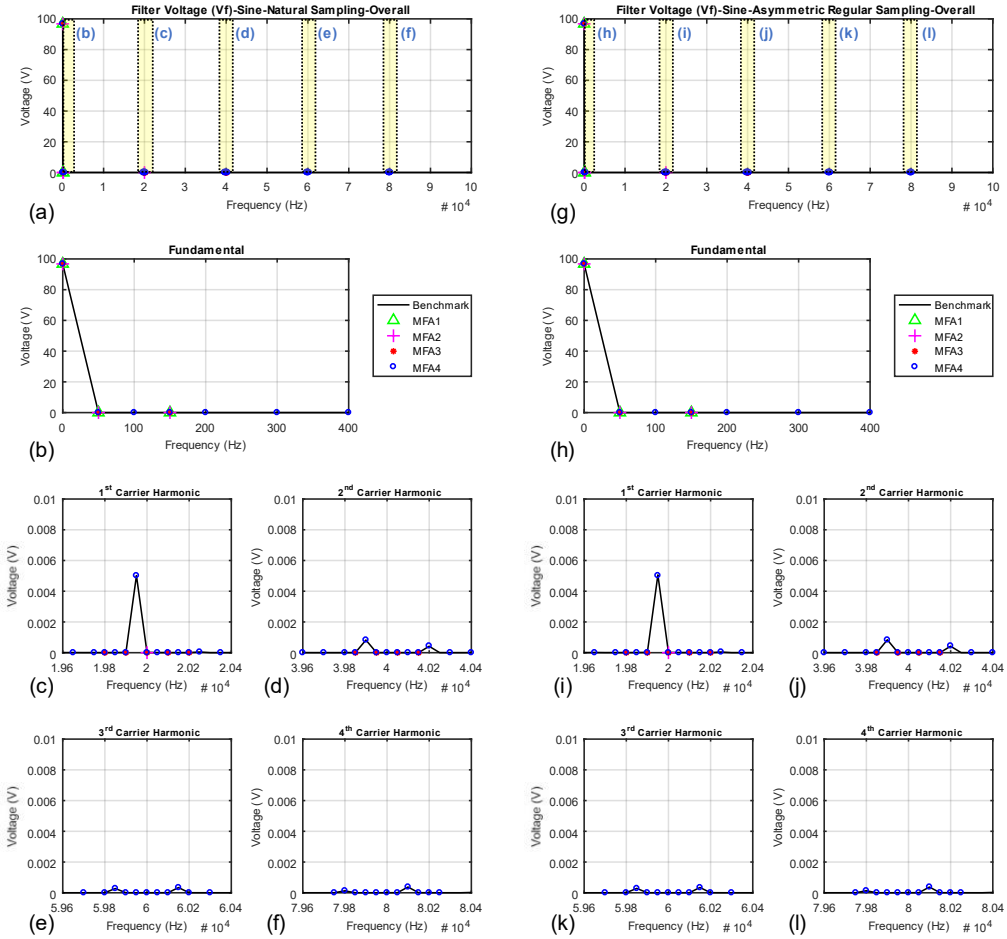


Fig 4-42 Frequency spectrum comparison (V_f) for Natural and Asymmetric Regular Sampling (sine PWM switching strategy)

The two graphs above and below show good reconciliation between different MFA outputs and the FFT of benchmark waveforms. This is in line with the results observed for the time domain comparisons.

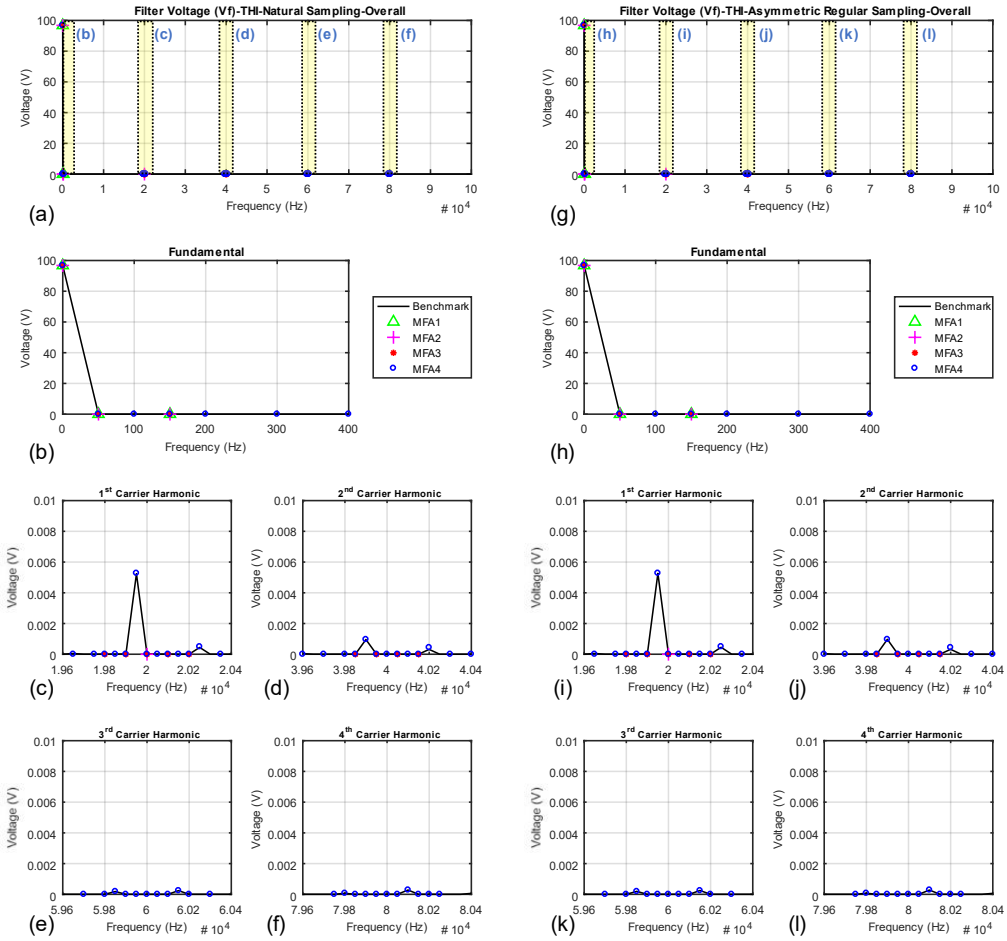


Fig 4-43 Frequency spectrum comparison (V_f) for Natural and Asymmetric Regular Sampling (third harmonic injection switching strategy)

4.6 Simulation Results (Variable Frequency)

Following the benchmarking exercise of the MFA Modelling Suite in the previous sub-chapter, this has been applied on a variable frequency application. The total simulation time duration has been kept the same as before, i.e., 0.1s, and the following frequency profile has been applied (Fig 4-44):

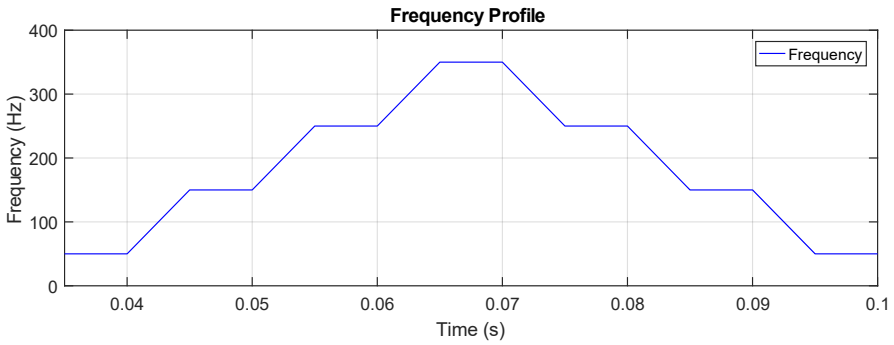


Fig 4-44 Frequency profile for variable frequency application of MFA model

4.6.1 Simulation Setup

The model parameters (resistance, inductance etc.) have been identical to the previous setup (steady state). Simulation time-step values have also been kept consistent, i.e., 0.5ms for the MFA model and 0.5µs for the time-domain models (MFA conversion and Benchmark models).

Sine PWM and third harmonic injection switching strategies using the natural sampling method have been shown below. Regular sampling methods were compared with the natural sampling results in the steady state experiments (sub-chapter 4.5) and conclusions drawn. The results under variable frequency operation were very similar as earlier, and hence have not been depicted here. All four frequencies (i.e., MFA1, MF2, MFA3, and MFA4) have been run for this simulation run.

4.6.2 Waveforms Comparison (Time domain)

Simulation results for the **phase current** waveform in time-domain (i.e., i_{abc}) are shown below in figure Fig 4-45. The structure is as follows:

- Two columns of graphs, (a)-(d) and (e)-(h), depict Sine PWM and third harmonic injection strategies, respectively.
- Top row of graphs, (a) and (e), are the overview spanning the entire duration (0.035s to 0.1s). Simulation up to 0.035s was allowed to run for the system to attain steady state.
- Second row of graphs, (b) and (f), are zoomed-in views of the waveforms at 0.04s to 0.05s.

- Third row of graphs, (c) and (g), are zoomed-in views of the waveforms at 0.07s to 0.08s.
- Fourth row of graphs, (d) and (h), are zoomed-in views of the waveforms at 0.085s to 0.1s.

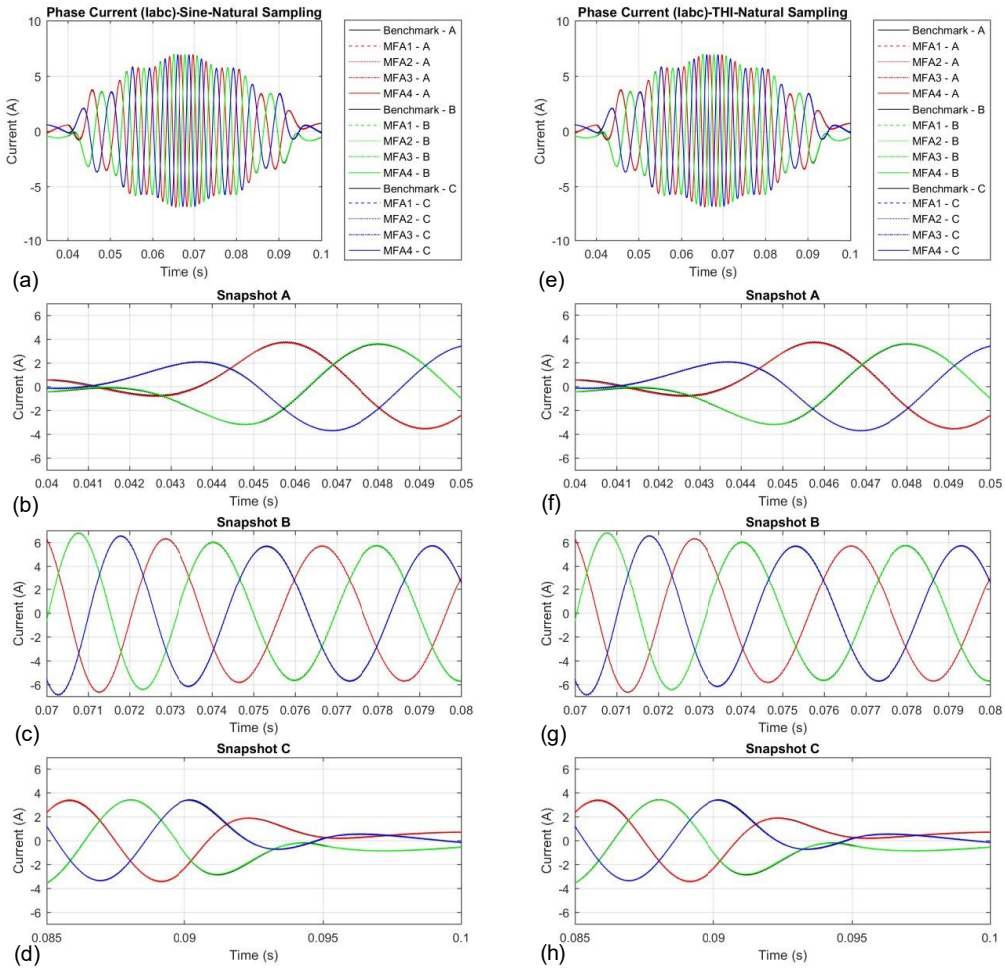


Fig 4-45 Time-domain waveform comparison (i_{abc}) for naturally sampled Sine PWM switching strategy

Simulation results for the **battery current** waveform in time-domain (i.e., i_{batt}) are shown below in Fig 4-46. Parts (a) and (b) depicts Sine PWM and third harmonic injection strategies, respectively.

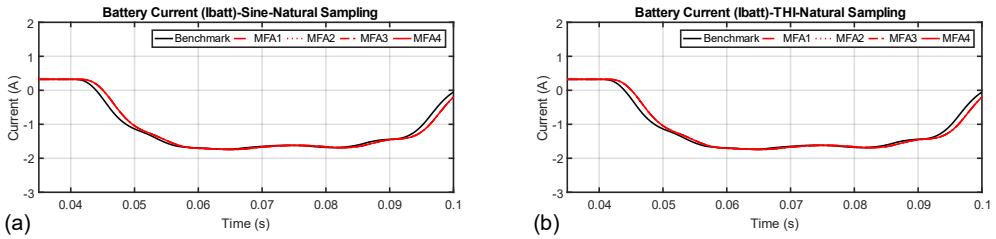


Fig 4-46 Time-domain waveform comparison (i_{batt}) for naturally sampled Sine PWM switching strategy

Simulation results for the **DC bus voltage** waveform in time-domain (i.e., v_f) are shown below in Fig 4-47. Parts (a) and (b) depicts Sine PWM and third harmonic injection strategies, respectively.

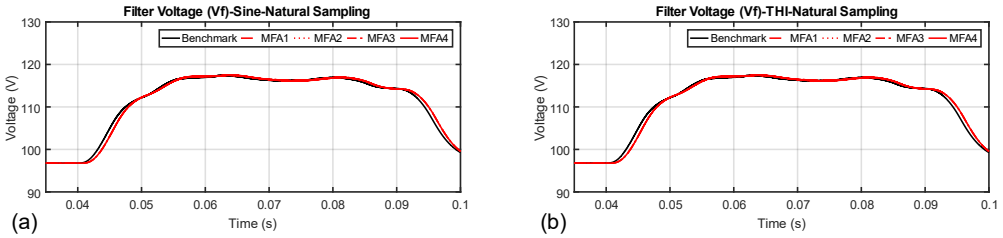


Fig 4-47 Time-domain waveform comparison (v_f) for naturally sampled Sine PWM switching strategy

The variable frequency operation results continue the promise of good reconciliation between the MFA results and the Benchmark. Phase current, battery current, and DC bus voltage figures are following the benchmark at the overview level very closely. The results at the minute level (i.e., where the carrier frequency components would be visible) have not be depicted in this section as the focus of this section is to examine the matching at an overview level only.

4.7 Summary

This chapter continued the efforts that began in the third chapter introducing the multi-frequency averaging (MFA) technique as a concept and its advantages for the DT application. First part of the chapter focussed solely on developing the mathematical theory from ground up to equip the reader with a solid foundation of this technique. The MFA DT framework was built upon this foundation. The two novelty elements of this

PhD were introduced, i.e., dynamically variable frequency operation and a fast method of multiplication of two waveforms in the frequency domain (an alternative to the standard method of convolution).

The next section described the complete implementation of the MFA model in the MATLAB/Simulink™ environment. This section discussed specific elements in the model and how they have been coded.

Lastly, this newly proposed DT was put through a set of rigorous validation experiments via simulation and comparison against a benchmark time domain switching model. The validation was done in two ways, steady-state at a fixed frequency and transient at variable frequency drivecycle. The steady-state results were compared in both time domain (waveforms) and frequency domain (dynamic phasors), both categories showed remarkable reconciliation with the benchmark. The variable frequency simulation experiments also showed that the MFA model was following the benchmark very closely.

Different switching strategies (sinusoidal, third harmonic injection, and space vector modulation) and sampling methods (natural, symmetric regular, and asymmetric regular) were introduced in their MFA-applicable forms. The results showed minor discrepancies in some of the simulation experiments using regular sampled switching methods. Complete validation of the model using all combinations of switching technique and sampling method should be conducted as a future exercise.

CHAPTER 5 Local Motor Emulation

5.1 Introduction

The previous chapter covered the first focus area of this PhD, i.e., DT of a generic electric powertrain deployable in a remote HIL location under varying conditions (PART 1). The second focus area (PART 2) answers the question of how to physically interface a test automotive inverter to a P-HIL rig and operate a closed-loop simulation. This chapter covers the technology development and implementation of the local HIL setup, and the next chapter covers the verification and validation using experimental results.

This chapter covers the following topics:

- P-HIL equipment details (for motor emulation, using Triphase hardware)
- Fundamentals of permanent magnet synchronous machine emulation
- Challenges of motor shaft position information delivery to real inverter
    - Low bandwidth of real time simulator and physical signal generation of encoder emulation
    - Phase delay compensation for encoder emulation

5.2 Triphase Real-Time P-HIL Simulator

A 90 kW P-HIL rig (Fig 5-1) was customised and commissioned by Triphase (now part of National Instruments) in 2017 for this PhD project. It is composed of three cabinets:

- **PM90T30**: 1x 90 kW isolation transformer (**T30**)
- **PM90A30F03**: AC/DC & DC/DC converters
    - 1x 90 kW active-front-end (**A30**)
    - 1x 90 kW DC/DC converter (**F03**)
- **PM90M30LP02**: Output stage
    - 1x 90 kW DC/AC PWM converter (**M30**)
    - 1x 90 kW direct DC bus out with fuse protection (**P02**)

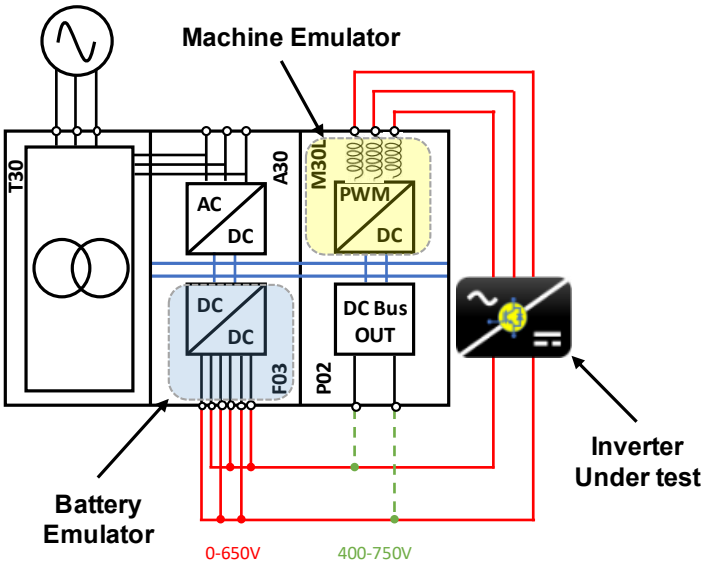


Fig 5-1 Triphase power-HIL simulator rig as configured and procured from Triphase for this PhD

The left cabinet is a simple isolation transformer (T30) intended to galvanically isolate the grid. Other than HV isolation with the grid, the transformer also allows common mode voltage swing on the secondary side which may be a by-product of switching action of three inverters that are downstream.

The middle cabinet consists of two inverter-and-filter units, one in the front and other in the back. The first inverter unit (A30) is called the Active Front End (AFE) that is tasked with keeping the DC Bus at a target voltage level. The AFE is 4-quadrant capable. When there is power being drawn downstream, the AFE is working in reverse direction (i.e., converting isolated grid AC to DC voltage, rectification). Conversely, when power is being generated, the AFE is equally capable of pumping power back into the grid. When the rig is started up, there is an independent pre-charge circuit (P30) that uses a single-phase diode-rectification stage with a pre-charge resistor to charge the DC bus first, before the AFE kicks-in enabling active rectification.

The second inverter in the middle cabinet (F03) is a fully controllable DC/AC voltage source. Three pairs of positive/negative pairs can be shunted together to produce a variable voltage from 0V to 650V with a continuous amperage of 240A. The reason for having three legs is that this is still a generic three-phase inverter controlled to produce a fixed DC voltage output, i.e., three independent buck converters. In this application,

the F03 power stage is designed to emulate a battery pack of an EV but can be equally deployed as a three-phase motor back-emf voltage generator.

The third cabinet has only one power stage (M30L) that is a fully controllable DC/AC current source. This also has three legs (same reasoning as above) and can produce a single-phase (hard wiring legs in parallel) and three-phase (independent legs) current output.

The major difference in the above two power stages (i.e., F03 and M30L) is the output filter stage. The F03 has a configurable LC/LCC/LCLC filter stage that enables the controller to produce a voltage output downstream of the filter. The M30L has a L filter stage enabling it to produce a current output downstream of the filter. In software, the power stage can be directed to emulate an extra impedance downstream of the filter as well, e.g., virtual resistance to dampen the power. These two features are discussed in further detail in the next sections.

5.2.1 Filter Stage

Any power electric converter (PEC) can produce a variable voltage/current output provided it has a downstream filter stage. A PEC works via PWM, i.e., switching each output leg between either the DC bus positive or negative rail very quickly which is averaged by the filter stage to produce a smooth curve. Without the filter, the output legs would only produce a variable duty cycle PWM voltage output.

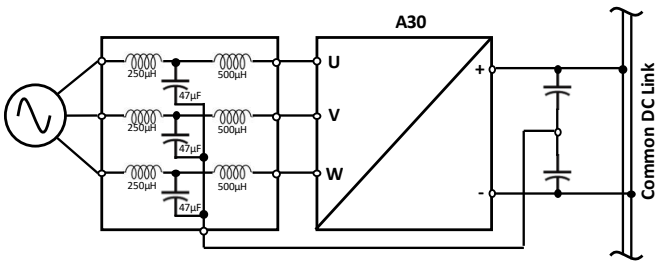


Fig 5-2 Detailed schematic of Triphase power-HIL simulator rig (A30 unit)

A30 (AFE, shown above in Fig 5-2) has an LCL filter which allows the simulator to control the current output to the grid (it can be positive and negative). An LCL filter is chosen for the AFE because the output voltage is grid-locked phase and voltage wise.

The capacitor offers voltage stability on the output and the controller can control the in-line current in the inductor. As a rule of thumb, it is easier to control current over an inductor and voltage across a capacitor. The reason is that inductors inherently resist quick change in current (similarly, capacitors resist change in voltage) and hence offers a stable controlled variable.

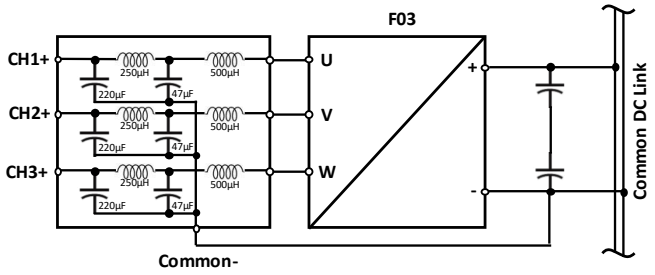


Fig 5-3 Detailed schematic of Triphase power-HIL simulator rig (F03 unit)

F03 (Fig 5-3) has a configurable LC/LCLC/LCC filter that can be pre-selected in software before powering the simulator on. The controller controls the voltage across the primary capacitor which are all referenced to the negative rail of the DC bus. This allows F03 to produce a voltage range of 0V to 650V. This is called Unipolar Voltage Source. Another alternative is Bipolar Voltage Source, wherein the capacitors are referenced to the centre-tap of the DC bus. In this case, the F03 can produce +/- half of DC bus voltage.

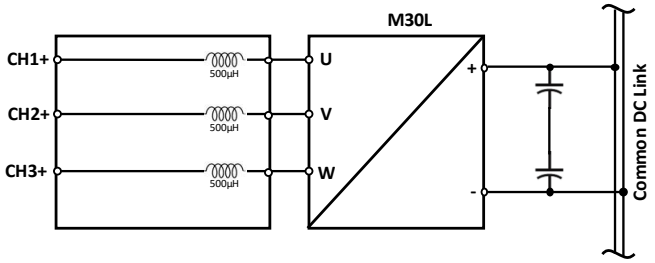


Fig 5-4 Detailed schematic of Triphase power-HIL simulator rig (M30L unit)

M30L (Fig 5-4) has a single stage static L filter only. This is designed to emulate a motor drive which is essentially a three-phase source/sink of current.

5.2.2 Virtual Circuit Emulation

Except the M30L (motor emulator), the power stages have filters which are at least second order, i.e., consists of an inductor and capacitor. Inductors and capacitors are energy storing devices and ideally are lossless, i.e., whatever energy they absorb, they can release the equal amount. In LC circuits, there is a real danger of resonance wherein at a particular resonant frequency, the stored energy swings back-and-forth between the two devices and can lead to uncontrolled oscillation. To prevent this from happening, a resistor is usually placed in series to dampen any resonant current. But a resistor also dissipates energy (to dampen resonant currents) which takes away from useful energy in non-resonant conditions.

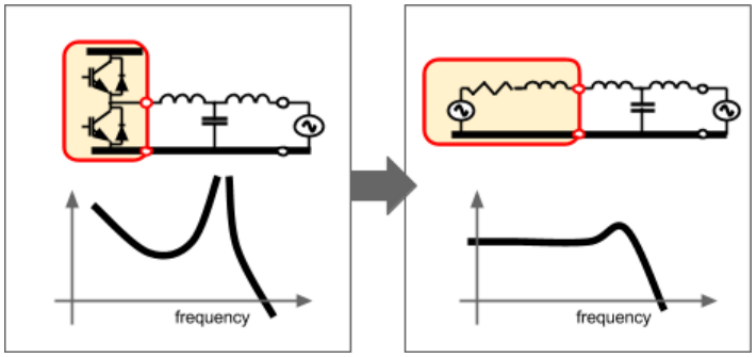


Fig 5-5 Virtual Circuit Emulation as implemented in Triphase simulator

To counter this problem, virtual circuit emulation (Fig 5-5) has been implemented in the Triphase controller. The power stage is implemented as a voltage source with a virtual resistance and inductance in series upstream of the physical filter stage. The emulated circuit is implemented using the control law as shown below.

$$v_{inv} = v_{command} = \frac{L_1}{L_1 + L_{emulated}}(v_{emulated} - R_c i_{inv}) + \frac{L_{emulated}}{L_1 + L_{emulated}} v_C$$

This provides the user with the additional capability to emulate extra impedance if needed for HIL experiments. For example, in the motor emulator M30L, there is a physical 500 µH inductor but if the emulated motor has a higher stator winding impedance, this can be virtually emulated.

5.2.3 Real-Time-Target (RTT) PC and Underlying Software

Fig 5-6 Triphase power-HIL simulator rig and associated equipment and computers

The control is provided by a CPU-based real-time computer running on Linux OS. This is called the Real-Time-Target (RTT). The RTT is programmed and operated via an engineering PC which is a standard engineering machine running Windows OS and MATLAB/Simulink™ as the base software to develop the application program. Once the application software is set, it is built using the Simulink™ Coder plug-in to produce a C++ code that is uploaded to the RTT via LAN connection.

The RTT connects to the PM90 modules via an optical fibre connection. The PM90 controllers are FPGA-based that control the PM inverters and applies several hardware limits on current and voltage. The presence of FPGA-based limits “via hardware” offers extra layer of protection.

5.2.4 EtherCAT Module

The EL2008 manufactured by Beckhoff is used in conjunction with the whole Triphase P-HIL setup to generate digital signals (Fig 5-7). This communicates with the RTT via

EtherCAT technology that is an ethernet-based fieldbus protocol specifically developed to apply real-time computing and low jitter.

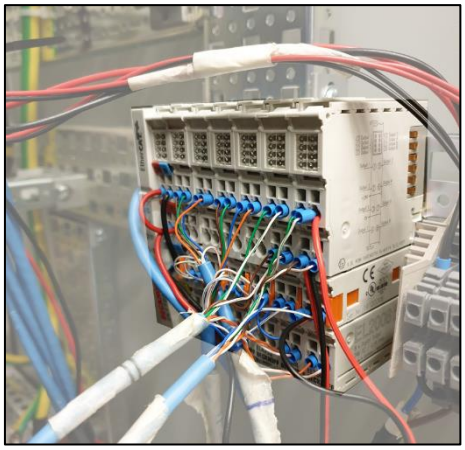


Fig 5-7 Beckhoff EtherCAT module for digital and analog I/O interfacing

5.3 Permanent Magnet Synchronous Machine Control Theory

Before delving into the experimental methodology and results for permanent magnet synchronous machine (PMSM) emulation using the triphase P-HIL simulator rig, it is imperative to briefly revise the theory of PMSM control (Fig 5-8).

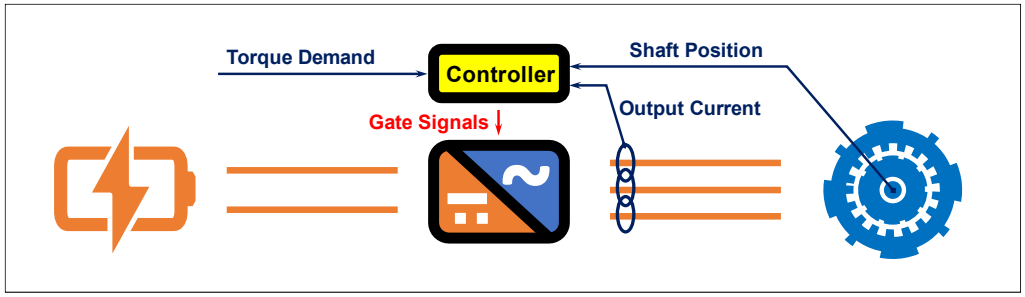


Fig 5-8 Brief overview diagram of generic permanent magnet synchronous motor drive

In synchronous mode, what the controller is essentially doing is:

- Continually track the spatial orientation of the polarity of the permanent magnet dipoles in the rotor
- Applying a voltage on the stator externally that induces a current in the windings that eventually produce “virtual” magnetic dipoles in sync with the rotor “physical” dipoles

- Depending on the torque demand (positive for “motor” action, negative for “generator” action), the “virtual” dipoles are phase advanced by a certain angle so that the “physical” dipoles are always chasing, producing torque. For negative torque demand, the “virtual” dipoles are phase delayed, so retarding the “physical” dipoles.

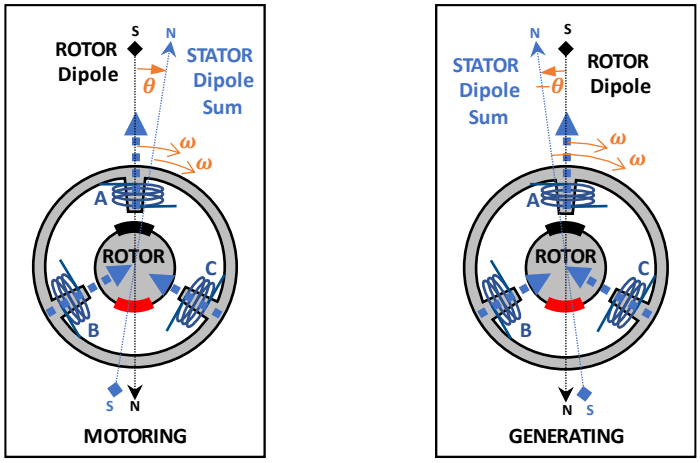


Fig 5-9 Motoring and Generating action conceptualised for a synchronous machine

In Fig 5-9 above, a single rotor dipole is represented for illustration. Usually, there are multiple dipoles, or “pole pairs” spatially distributed in the rotor. When there are multiple pole pairs, a single mechanical revolution is divided into multiple electrical revolutions, exactly as many pole pairs are present. For example, if there are six pole pairs, for a single mechanical revolution of the motor, there would be six electrical revolutions as seen by the controller.

The “virtual” dipoles are a direct effect of the current flowing through the stator windings. Therefore, controlling the current controls the dipoles. The torque demand directly translates to the phase advance (or delay) of the stator dipole, but implementing this requires controlling three independent variables (i.e., phase currents) to a sinusoidal reference, or “requested” value. This poses a challenge simply because it would require complex computation and electronics.

5.3.1 Clarke and Park Transforms

If the three phase currents are transformed from fixed reference frame (i.e., 3 stator phases) to a rotating reference frame (i.e., 2 quadrature-shifted axes fixed to the rotor),

the control simplifies significantly due to two reasons. First, only two variables are now being controlled (instead of three). Second, the reference is not sinusoidal anymore, it is DC in the rotating frame of reference. This mechanism is implemented using two mathematical transforms, Clarke and Park Transforms. Clarke Transform is used to convert measured current in its fixed abc frame (3-phase with 120° gap between phases) to fixed $\alpha\beta$ frame (quadrature axes system), and Park Transform is used to convert this to the dq frame (rotating with the rotor).

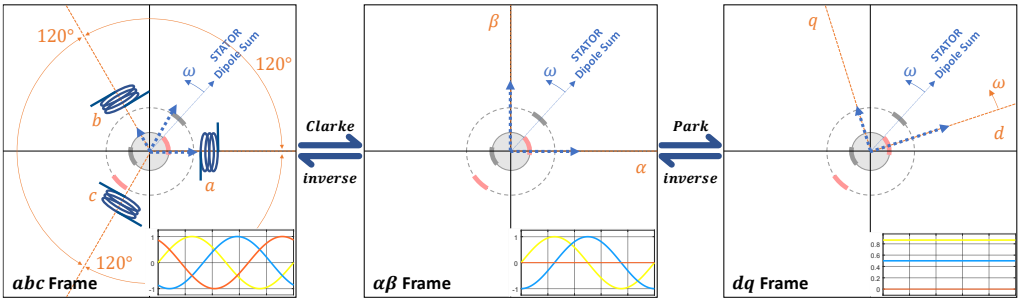


Fig 5-10 Conceptual difference between frames of reference for three-phase machine control: abc ∨ $\alpha\beta$ ∨ dq

The dq frame is especially useful and simplifies AC control to equivalent DC control (Fig 5-10). Applying field in the q axis directly induces torque in the rotor and applying field in the d axis add/removes field in the permanent magnets. If this is observed intuitively, the “virtual” dipole in the stator q axis is attracting/repelling the “physical” dipole in the rotor (positive q axis pulls the rotor in direction of the rotation, hence motoring action; negative q axis pushes the rotor away from the direction of rotation, hence generating action). The “virtual” dipole in the stator d axis is reinforcing field generated by the permanent magnets, but since they resist change to the flux density through themselves (i.e., offer reluctance), the d axis stator field reduces the flux produced from the permanent magnets, in turn reducing the effect of q axis field. While this may seem unintuitive, it helps reach speeds higher than the rated value (defined by the maximum applied stator voltage) at the cost of reduced torque (this is called Field Weakening and shall not be investigated further).

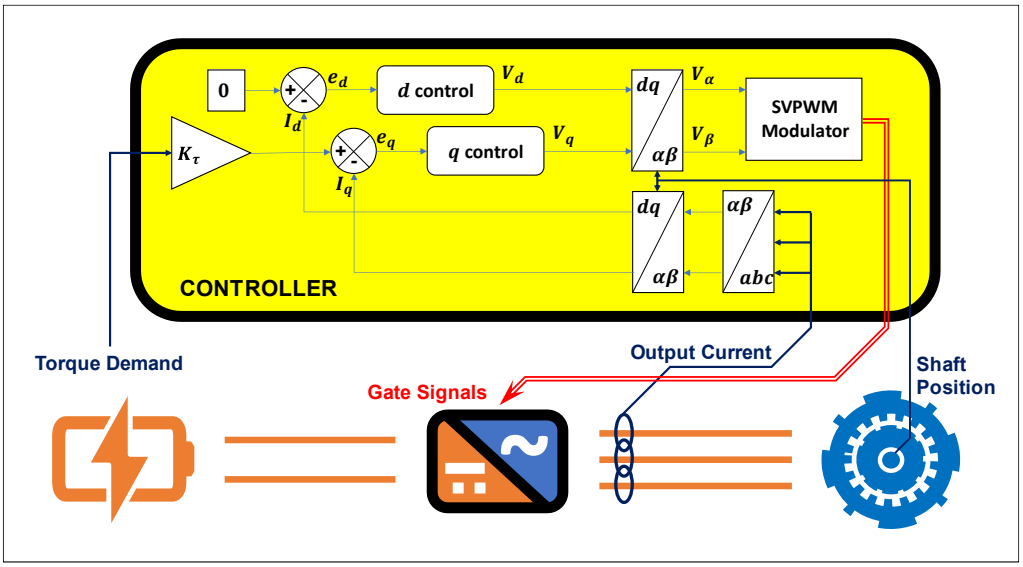


Fig 5-11 Detailed overview diagram of generic permanent magnet synchronous motor control

With a basic introduction to the foundation of AC machine control, the components of the “controller” block (Fig 5-11) can be explored in further detail.

5.3.2 Control System

After converting the 3-phase currents to the dq frame, the control system simplifies as it is de-coupled DC control now. The torque demand from -100% to +100% is linearly transformed to q-axis current setpoint -100% to +100% by multiplying with the Torque Constant (amperes per RPM). d-axis current setpoint is set to zero for rated operation and can be introduced to implement field-weakening. Two separate controllers are used to convert the error values to voltage setpoints. These controllers could be simple PI based or something complex like non-linear or fuzzy logic-based control. The voltage setpoints are again converted to the fixed reference frame (either $\alpha\beta$ or abc) and passed on to the Modulator that converts it into the six gate pulses.

5.3.3 Modulator

Modulation is the part where analog 3-phase voltage signals are converted to digital on/off signal for the six gate pulses in the power stage of the inverter. There are several modulation strategies like Sinusoidal PWM, Sinusoidal PWM with Third Harmonic Injection (THI), Delta Modulation, Direct Torque Modulation, and Space Vector Modulation (SVM) that were discussed in the previous chapter (section 4.3.1).

5.4 Encoder Emulation – ABZ Type

5.4.1 Theory of Encoder

It is well understood that PMSM control requires accurate motor shaft position information. In physical systems, this is done via a device attached to the motor shaft which send a physical signal representing high resolution shaft position information to the inverter/controller. This device is most commonly of two types: **analog resolver** or **digital encoder**.

Analog resolver requires a high frequency “exciter” sinusoid generated by a driving circuit (often in inverter/controller itself) that is converted to a pair of quadrature-shifted sinusoids with an amplitude envelope representative of the shaft position. This signal is read by the inverter/controller to produce phase voltages.

Digital encoder (most commonly the ABZ type, shown in Fig 5-12 below) produces digital pulses incrementally as the shaft moves through a revolution and an index pulse every time it makes a revolution (this kind is also called the Incremental Encoder and the analogue resolver is called the Absolute Encoder). A and B form a quadrature-shifted pulse train, total pulses per revolution varies for different encoders but is often in the 2^{10} to 2^{12} range. Z is the index pulse which occurs once per revolution. The inverter/controller reads the three signals and decodes accurate shaft position information by incrementally adding a counter from every consecutive edge of the A/B pulses and resetting the counter at every Z pulse. The direction of the motor shaft is interpreted by the sequence of the A/B phase just like in the analog resolver.

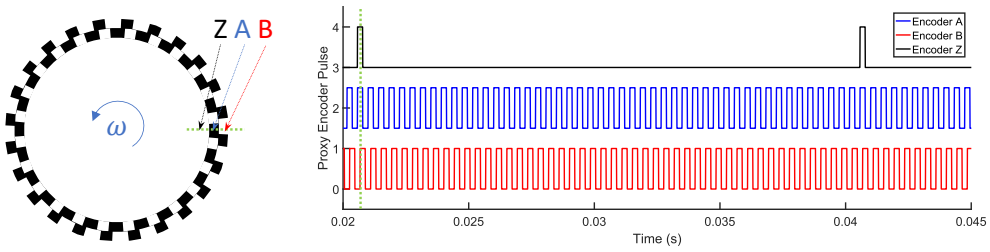


Fig 5-12 ABZ type digital encoder operation principle

There are other kinds of encoders as well (e.g., UVW type) which are also used in some applications but shall not be discussed in this work.

5.4.2 Encoder Emulation in Digital HIL Simulator

Due to the absence of a real physical motor being driven by the test inverter, there is no physical encoder present in loop. This means an integral part of motor emulation is to emulate an encoder as well. This section hereon presents the emulated encoder signal and accuracy comparison from the true value.

The Triphase simulator is a CPU-based real-time engine which makes it robust but also limits its time-stepping capabilities. The real-time model operates at 16 kHz frequency which means in ideal conditions, it can generate a pulse of up to 8 kHz frequency (one pulse must contain at least a high bit and a low bit in a cycle). This is the theoretical limiting frequency of requested pulse train. Next, the physical signal generation hardware limits also must be considered.

It was discussed earlier (section 5.2.4) that an EtherCAT based digital signal generator is used in the simulator. The rise and fall times are typically 60µs and 300µs respectively. If minimum 20µs of stay-on time and 20µs of stay-off time is included (conservatively), a full period is 400µs, which limits the maximum frequency of an accurately generated pulse train of 2.5 kHz.

If a digital encoder signal of $n_{encoder}$ pulse-per-revolution and a mechanical rotation of f_{motor} Hz is to be emulated,

$$f_{motor} \leq \frac{2500}{n_{encoder}} \qquad 5.1$$

A good encoder would have a minimum of 1024 pulse-per-revolution. That gives a maximum motor frequency of ~2.5 Hz which is unrealistic for any practical utility. A solution for this was discussed in chapter 3 wherein an intermediate FPGA stage converts a low-resolution shaft position information to a high-resolution signal through interpolation. A similar solution has been developed for this work. A microcontroller based intermediate stage reads a low-resolution proxy encoder output and converts it into detailed shaft position information that is fed into the inverter/controller. The best

low-resolution proxy encoder pulse-per-revolution needs to be arrived at, so that there are adequate “hard values” of information interspersed in a complete electric revolution and the gaps are filled with “soft values” via interpolation. Best interpolation strategies are discussed later.

Most modern inverters/controllers use Space Vector Modulation (SVM). In SVM, a single electric revolution is divided into six equal sectors. It was decided to use two hard values per sector. The microcontroller intermediate has the capability to detect both rising and falling edge of a digital signal. Using both edges of A and B pulses as position-stamps, each pulse produces four distinct position-stamps, hence the total position resolution of the encoder is four times of its pulse-per-revolution value. Therefore, a total of at least three pulse-per-revolution is needed in the proxy encoder. This gives us a limited maximum motor frequency possible of 833 Hz (e.g., for six pole pairs, this translates to 8333 RPM).

Methodology

A series of tests were conducted to quantify the error in the encoder emulation approach adopted for this PhD. The steps followed are as follows:

- Triphase simulator produced (via Beckhoff EtherCAT module) proxy encoder A/B/Z signals at different fixed frequencies (A sweep from 50 Hz to 500 Hz in steps of 50 Hz was conducted).
- The proxy encoder signals physically produced by the EtherCAT module were scoped using high resolution data capture instrument.
- The recorded experimental data was compared with the demanded data (by the simulator) in two ways to quantify the error:
    - **PART A** - The recorded encoder A/B pulse train was compared with the “true” demanded value. The time deviation was converted to the phase angle deviation (time in s multiplied with angular frequency in rad/s to give rad, which was eventually converted to ° by multiplying with $\frac{360}{2\pi}$) and plotted on a graph.

- **PART B** - The recorded A/B/Z pulse train was converted to the shaft position information exactly as would be done in the Proxy Encoder microcontroller hardware. This position information was compared with “true” value and various error metrics were computed.

Results – PART A

Recorded A/B/Z pulses were compared with the demanded “true” pulses to establish the extent of delays in the real-time output hardware and EtherCAT module. The average and standard deviation in phase error (in degrees) was calculated and compared. A/B/Z pulses are in blue, red, and black respectively, with experimental and true data in solid and dashed lines respectively (Fig 5-13).

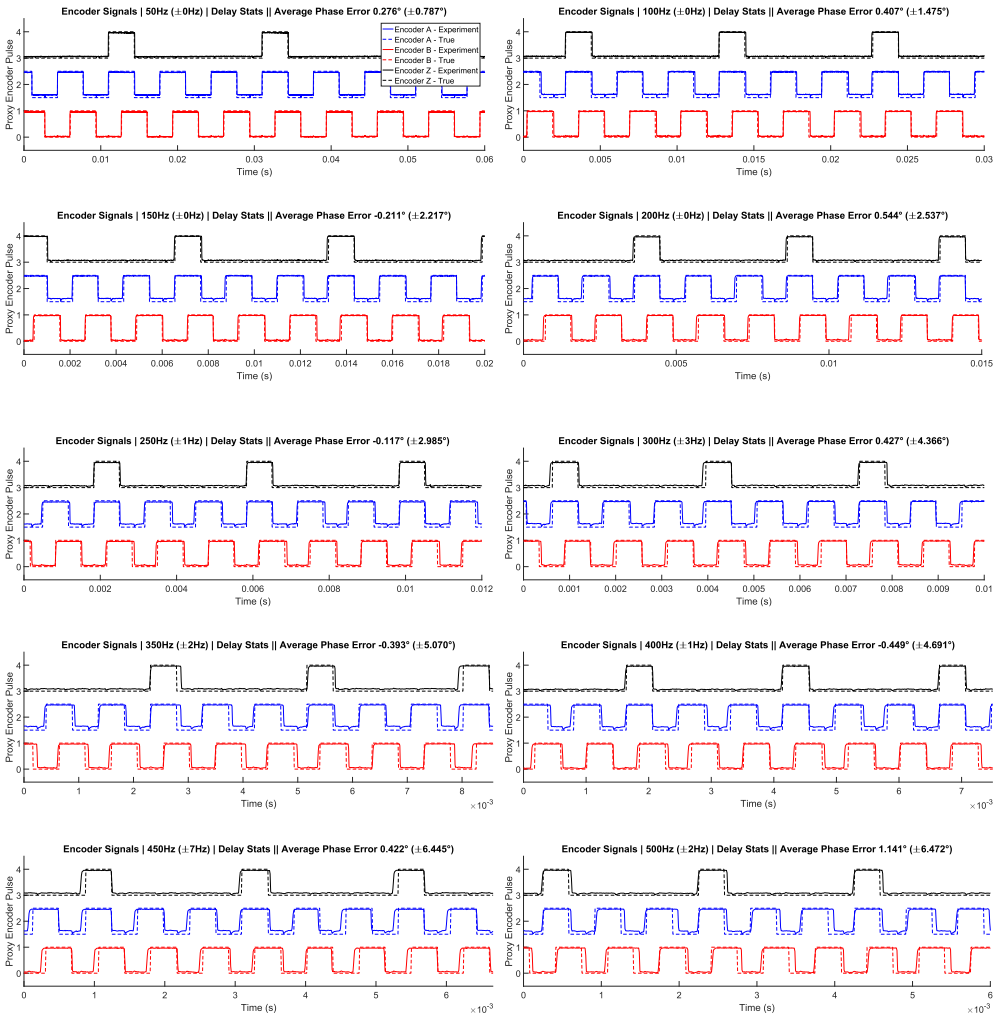


Fig 5-13 PART A Results – Measured encoder pulse signals compared with requested signals

Mean error and standard deviation was computed in the following manner. The oscilloscope captured a snapshot of at least 20 full revolutions of the A/B/Z pulses physically reproduced by the EtherCAT module. These were then aligned with the digital “requested” signals on the simulator PC. Exact time information of all the A/B pulse edges (rising and falling) and Z pulse edges (rising only) was compared and the mean error and standard deviation was produced.

Frequency	Encoder A		Encoder B		Encoder Z		Combined	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
Hz	°	°	°	°	°	°	°	°
50	0.276	*0.665*	0.255	*0.695*	0.304	*1.025*	**0.276**	***0.787***
100	0.573	*1.361*	0.470	*1.424*	0.189	*1.684*	**0.407**	***1.475***
150	-0.173	*2.027*	-0.284	*2.156*	-0.182	*2.535*	**-0.211**	***2.217***
200	0.803	*2.694*	0.636	*2.819*	0.209	*2.176*	**0.544**	***2.537***
250	-0.094	*3.348*	-0.204	*3.564*	-0.056	*2.133*	**-0.117**	***2.985***
300	0.652	*4.058*	0.484	*4.322*	0.158	*4.850*	**0.427**	***4.366***
350	-0.277	*4.778*	-0.459	*5.062*	-0.454	*5.523*	**-0.393**	***5.070***
400	-0.365	*5.420*	-0.568	*5.658*	-0.428	*3.136*	**-0.449**	***4.691***
450	0.558	*6.080*	0.293	*6.387*	0.428	*7.063*	**0.422**	***6.445***
500	1.874	*6.636*	1.741	*7.313*	-0.157	*5.662*	**1.141**	***6.472***

Table 5-1 PART A Results - Error statistics between measured and requested encoder signals for A, B, and Z pulses

The results from these tests are summarised. It is interesting to note that the mean error is insignificant, but the standard deviation is significant and rises with the increased frequency. This makes sense due to two reasons:

- Due to the alignment of “experimental” and “true” signal sets, the average delay or latency information is lost. So, the mean error figure is inconsequential.
- This experiment intends to shine a light on the jitter. It is expected to see the jitter value rise with the frequency of operation, and is reflected in the table below.
- This jitter is caused by inherent indeterministic nature of the EtherCAT device. The jitter amount is comparable across all frequency range when looking at it in terms of time (i.e., seconds), but the real effect to application is measured in angular error (i.e., degrees).

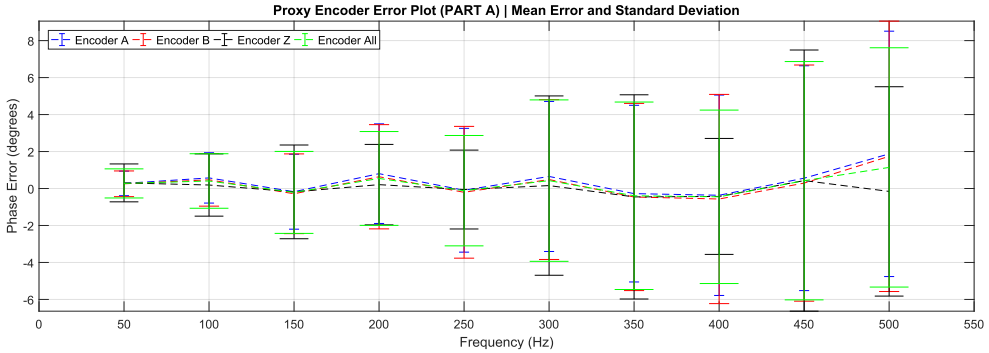
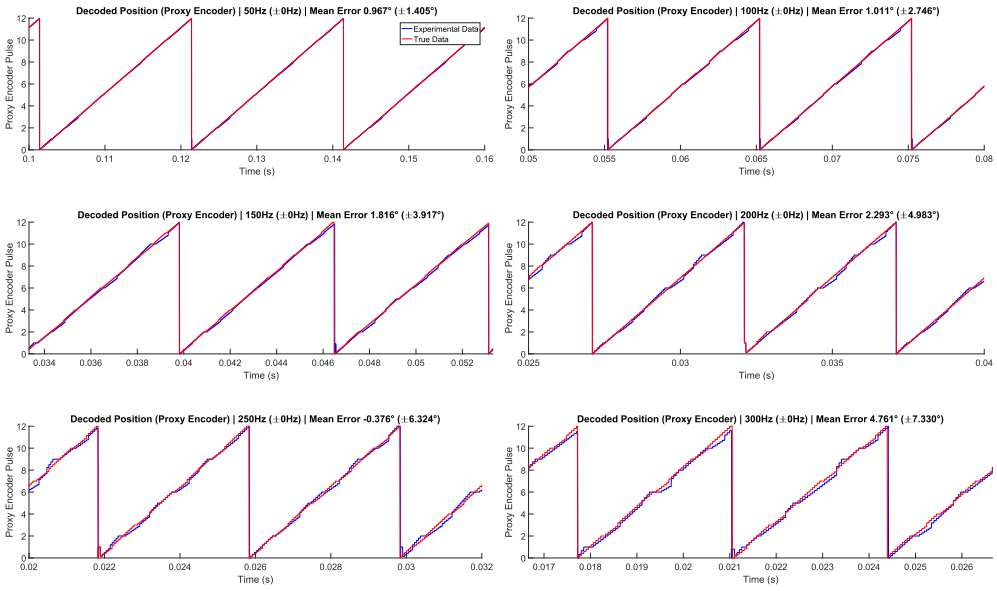


Fig 5-14 PART A Results - Error graph between measured and requested encoder signals for A, B, and Z pulses

Results – PART B

The “experimental” and “true” encoder pulses were used to recreate the detailed shaft position information exactly as it would be done in the “proxy” encoder intermediate stage (coded in the inverter-under-test). These were then compared and are shown below (Experimental/True position information in blue and red solid lines respectively). Three different interpolation strategies were tried:

- **TYPE 1** – linear trend between the previous two AB pulse edges (or “hard” values) was used to extrapolate the “soft” values until the next A or B pulse is detected.



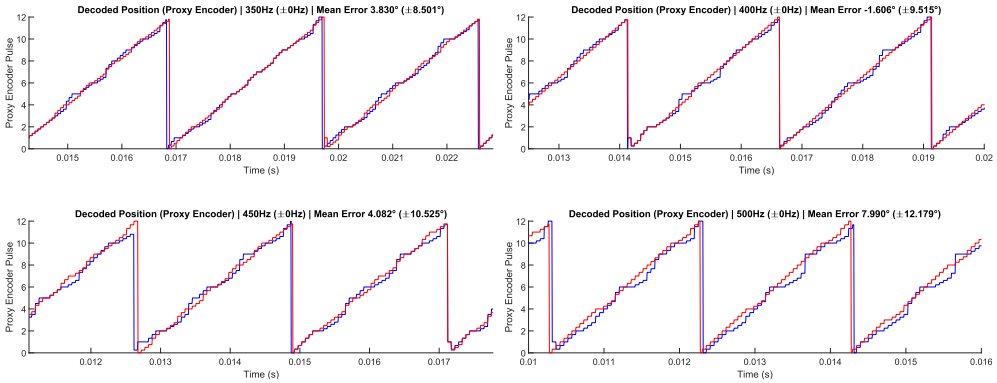


Fig 5-15 PART B Results – Measured phase angle compared with requested angle (TYPE 1)

- **TYPE 2** – linear trend using previous five AB pulse edges (or “hard” values) was used to extrapolate the “soft” values until the next A or B pulse is detected.

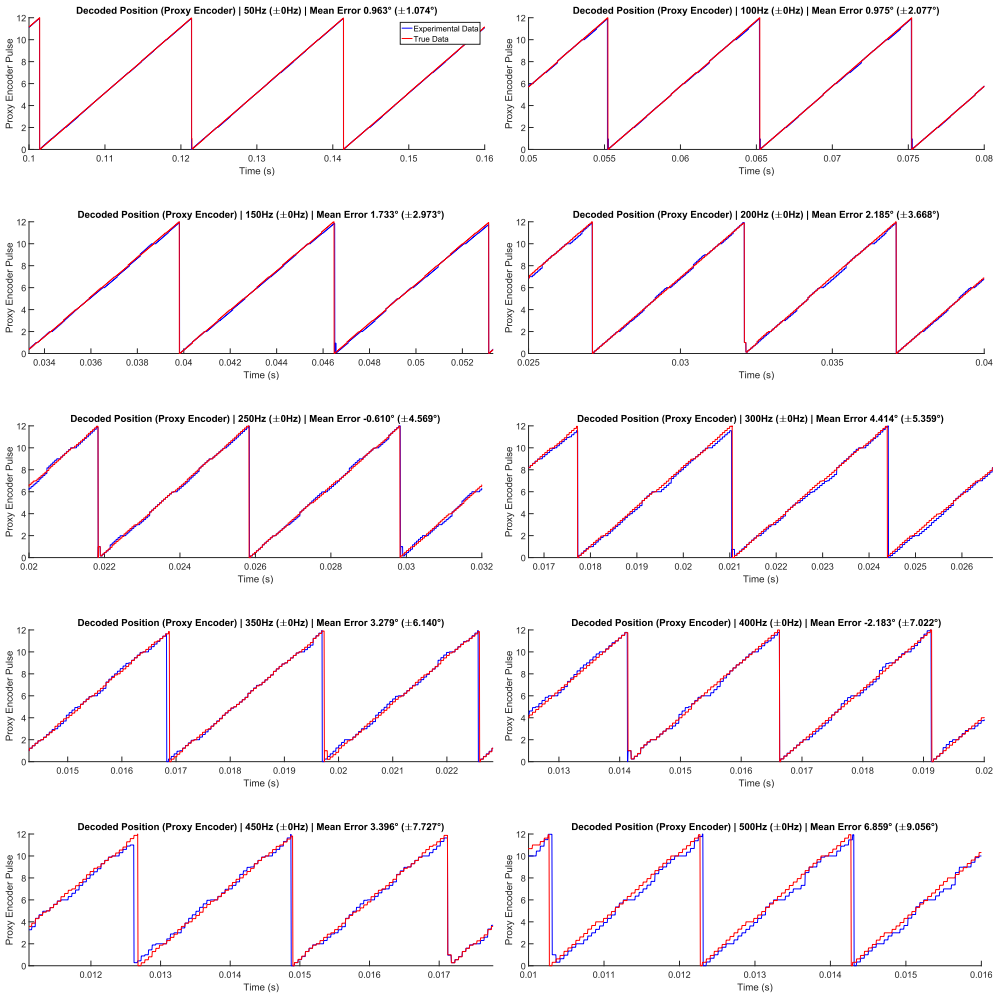


Fig 5-16 PART B Results – Measured phase angle compared with requested angle (TYPE 2)

- **TYPE 3** – Only the rising edge of Z pulse was used to calculate an average gradient of position values. This reduces the number of “hard values” per revolution from 12 to 1.

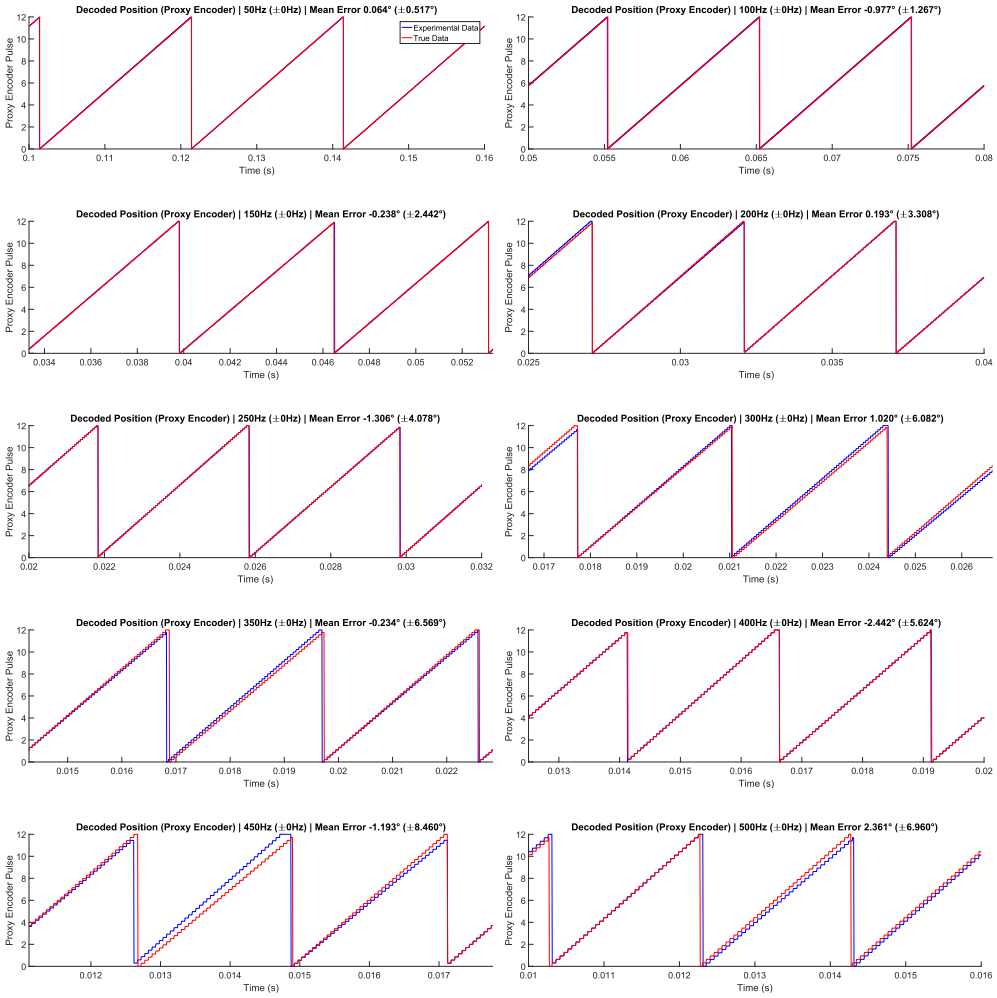


Fig 5-17 PART B Results – Measured phase angle compared with requested angle (TYPE 3)

The same datasets for experimental and true encoder pulses were used to do this study, hence at least 20 full revolutions were compared for each frequency value. The mean error and standard deviation were computed and has been summarised below.

Frequency	Type 1		Type 2		Type 3	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
Hz	°	°	°	°	°	°
50	0.967	1.405	0.963	1.074	0.064	0.517
100	1.011	2.746	0.975	2.077	-0.977	1.267
150	1.816	3.917	1.733	2.973	-0.238	2.442
200	2.293	4.983	2.185	3.668	0.193	3.308
250	-0.376	6.324	-0.610	4.569	-1.306	4.078
300	4.761	7.330	4.414	5.359	1.020	6.082
350	3.830	8.501	3.279	6.140	-0.234	6.569
400	-1.606	9.515	-2.183	7.022	-2.442	5.624
450	4.082	10.525	3.396	7.727	-1.193	8.460
500	7.990	12.179	6.859	9.056	2.361	6.960

Table 5-2 PART B Results - Error statistics between measured and requested phase angle

TYPE 2 technique is visibly superior to TYPE 1 technique wherein the average of previous four gradient values of shaft position change was used to interpolate position information until next AB edge is detected. Keeping this in mind, TYPE 2 technique has been selected for the low frequency regions, i.e., up to 250 Hz. Observing Part A graph, the overall signal standard deviation is within 3° for signals less than 250 Hz. The position error and standard deviation increases beyond 250 Hz for Type-2 interpolation technique where TYPE 3 technique shows more promise. The primary concern with TYPE 3 technique is that although the accuracy is higher, the “hard” value resolution drops from 12 to 1 (there are 12 AB edges for every Z pulse rising edge). While this is not a problem for fixed frequency operation (as the position changes at a fixed rate), different rate of change of shaft position within a single revolution is averaged causing a step change at every revolution.

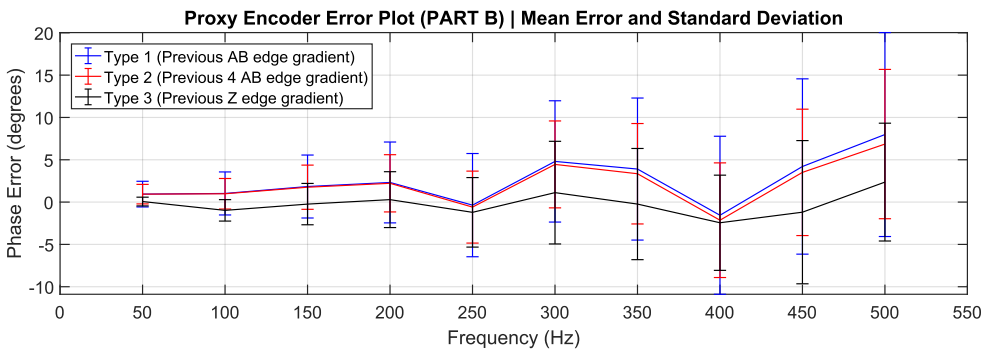


Fig 5-18 PART B Results – Error graph between measured and requested phase angle

It is interesting to note why only three methods of extrapolating the position information for generating “soft” points from previous “hard” points were shown. TYPE 2 extrapolation used the previous five “hard” points to linearly project the next position “soft” points. Other techniques were also used, i.e.:

- Linear trend with previous three, four, six, and seven points was also tried. But the one with previous five points was found to produce the best results.
- Other forms of curve fitting like quadratic, cubic, polynomial and exponential were also tried. They did not offer any advantages. This makes sense because the only significant mode of error introduced is due to the jitter in the reproduced encoder pulses. Having a simple averaging method (in linear curve fitting) eliminates the jitter to the best possible extent without adding computational overhead.

5.4.3 Phase Delay Compensation in Proxy Encoder Signal Generator

Other than accurately sending the motor shaft position information to the test inverter/controller to conduct a successful P-HIL simulation, the timing also needs to be very accurate, i.e., overall latency needs to be minimised (not just jitter). The previous section 5.4.2 focussed on characterising the jitter produced in the proxy encoder solution (hardware and software) at various fixed frequency operation points. This section aims to characterise the latency (average delay between simulator sending the signal and signal being physically reproduced) and eliminate it via phase compensation in the simulator itself.

Like the previous sub-section, a high resolution datalogger was used to record the encoder signals and the produced back EMF under different fixed frequencies ranging between 5 Hz and 500 Hz. The time delay between encoder Z pulse rising edge and the peak of the back EMF produced in phase A was measured. Like previous experiments done in this chapter, multiple revolutions (at least 20 for each operating region) were recorded and the mean error and standard deviation are visualised below.

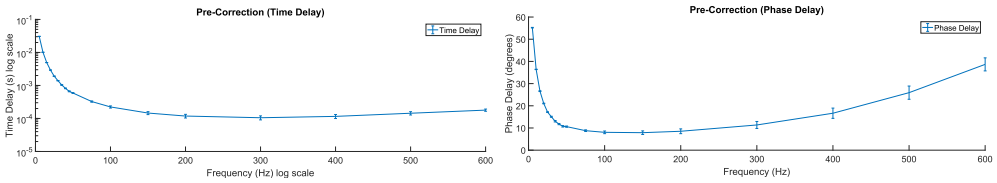


Fig 5-19 Time and Phase delay between measured and requested phase angle (pre-correction)

A spline was fitted on the phase delay and applied as a correction in the simulator application program. The experiment was repeated at fixed frequencies, results are reproduced below.

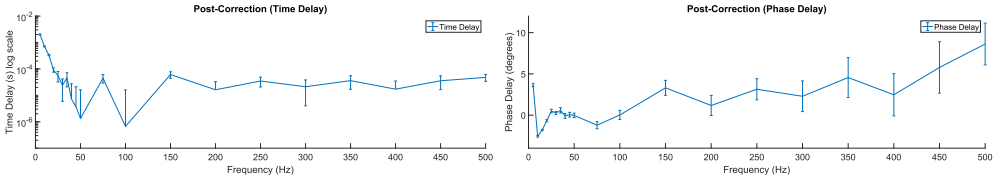


Fig 5-20 Time and Phase delay between measured and requested phase angle (post-correction)

The two results (pre and post correction) are reproduced in the same graph below.

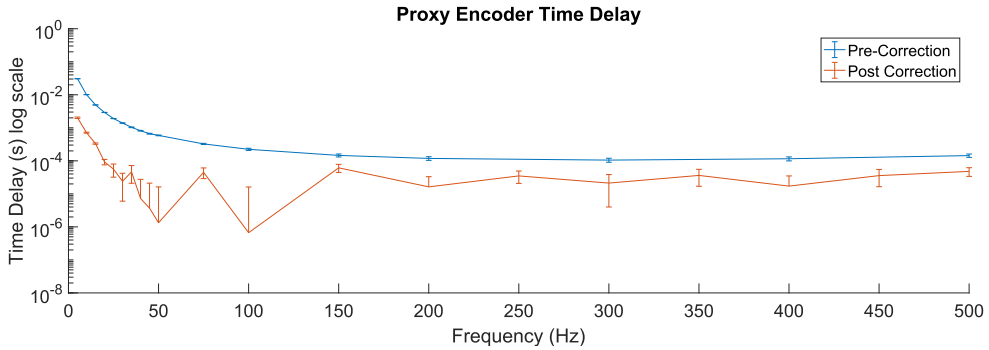


Fig 5-21 Time delay between measured and requested phase angle (comparison between pre and post correction)

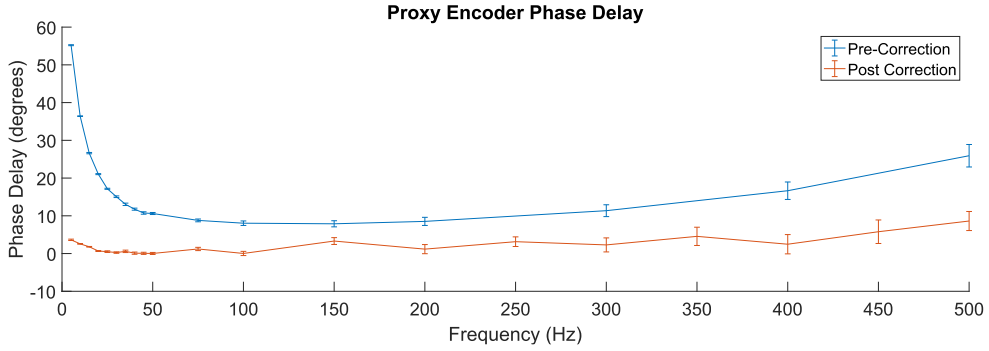


Fig 5-22 Phase delay between measured and requested phase angle (comparison between pre and post correction)

It can be clearly observed that the total phase deviation has been significantly reduced after the compensation exercise (Fig 5-22). Total phase deviation is below 5±2.5 degrees up to 400Hz.

Frequency	Pre-Correction				Post-Correction			
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
Hz	ms	ms	°	°	ms	ms	°	°
5	30.672	0.070	**55.208**	*0.127*	2.020	0.100	**3.636**	*0.180*
10	10.112	0.021	**36.402**	*0.077*	-0.713	0.024	**-2.568**	*0.088*
15	4.928	0.032	**26.611**	*0.173*	-0.333	0.015	**-1.797**	*0.083*
20	2.923	0.019	**21.043**	*0.137*	-0.093	0.017	**-0.668**	*0.124*
25	1.907	0.016	**17.159**	*0.140*	0.056	0.024	**0.504**	*0.216*
30	1.394	0.020	**15.055**	*0.219*	0.024	0.018	**0.259**	*0.194*
35	1.035	0.025	**13.043**	*0.317*	0.046	0.025	**0.585**	*0.321*
40	0.817	0.019	**11.760**	*0.272*	-0.007	0.020	**-0.105**	*0.291*
45	0.663	0.020	**10.747**	*0.321*	0.004	0.017	**0.060**	*0.282*
50	0.590	0.015	**10.612**	*0.269*	-0.001	0.015	**-0.024**	*0.268*
75	0.325	0.014	**8.784**	*0.368*	-0.045	0.016	**-1.214**	*0.431*
100	0.223	0.017	**8.042**	*0.596*	0.001	0.015	**0.024**	*0.556*
150	0.146	0.015	**7.884**	*0.810*	0.061	0.017	**3.312**	*0.905*
200	0.118	0.015	**8.521**	*1.077*	0.016	0.017	**1.172**	*1.212*
250	*	*	*	*	0.035	0.014	**3.144**	*1.276*
300	0.105	0.015	**11.359**	*1.570*	0.021	0.017	**2.290**	*1.858*
350	*	*	*	*	0.036	0.019	**4.556**	*2.419*
400	0.116	0.016	**16.649**	*2.312*	0.017	0.018	**2.472**	*2.555*
450	*	*	*	*	0.036	0.019	**5.782**	*3.117*
500	0.144	0.017	**25.920**	*2.979*	0.048	0.014	**8.620**	*2.526*
600	0.179	0.014	**38.665**	*2.938*	*	*	*	*

Table 5-3 Time and Phase delay statistics for pre and post correction

5.5 Discussion

Several hardware limitations of the P-HIL setup were discussed, i.e., low frequency time-stepping offered by triphase (CPU-based) real-time simulator, non-real-time implementation of proxy encoder signal and large rise/fall time of physical signal generated by the EtherCAT block. Owing to these limitations, fidelity of the generated shaft position information is not at par with the equivalent physical system. The degree of error was investigated to high detail and characterised. Eventually, the error was eliminated to the best possible extent offered by the software and hardware limitations of the system.

The requirement of 12 “hard” position-stamps per revolution depends on the maximum rate of change of frequency that is expected, equivalent to the acceleration rate of the

motor. For example, if the frequency changes by 100 Hz every second starting from an initial frequency of 1 Hz, the second pulse edge would appear earlier than what is expected as per the interpolation technique from previous pulse edge. On the other hand, if the initial frequency were 100 Hz, the error would still be there but would be less significant. the mathematical theory has been developed below. Using this, the minimum operable frequency using AB pulse edges (or TYPE 1 or TYPE 2 of PART B) and the maximum operable frequency beyond which using Z pulse only (TYPE 3 of PART B) makes more sense, can be arrived at.

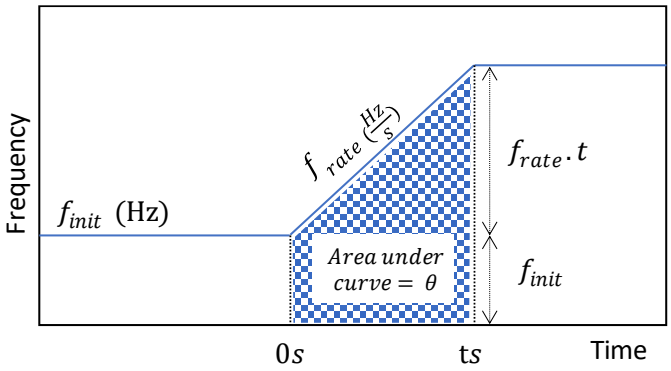


Fig 5-23 Mathematical concept underlying interpolation technique for phase angle computation

Let's say the initial frequency is f_{init} Hz and at 0 s, it starts to ramp up at f_{rate} Hz/s. In the illustration above, the area under the curve is the number of revolutions spanned by the motor shaft. Let's say it takes t s to complete one revolution (see Fig 5-23).

$$1 = f_{init}.t + \frac{1}{2}.t.f_{rate}.t \qquad 5.2$$

In case of type-3 interpolation technique, the error in position would be the triangular area (as the proxy encoder would think the frequency is still f_{init} based on previous revolution). It can be logically inferred that if f_{init} is high, the rectangular portion of the total area under curve would be more significant, consequently reducing the error overall.

In case of type-2 interpolation technique, only $\frac{1}{12}$ (or $\frac{1}{4*ppr}$) of the whole revolution is considered. Total triangular area under curve would automatically be smaller since t is

smaller to span only $\frac{1}{12}$ of a revolution. After every AB pulse edge, the f_{init} would change to the latest value. See illustration below in Fig 5-24.

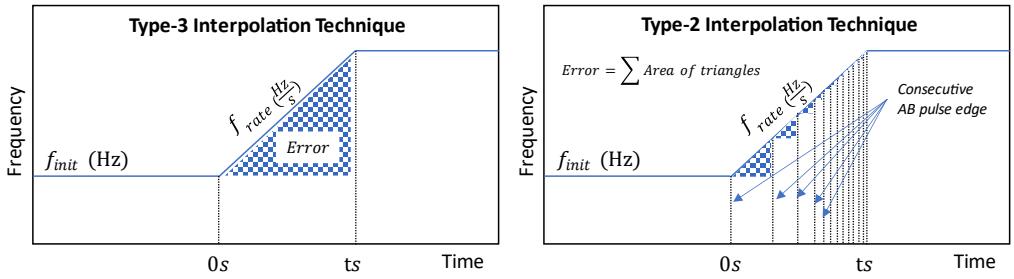


Fig 5-24 Mathematical concept underlying interpolation technique for phase angle computation (comparison of TYPE 2 and TYPE 3 techniques)

An effective way of characterising this is to calculate the error as a fraction of the total area under the curve for a single revolution. This can be done theoretically by solving the quadratic equation for time. Alternatively, a simulation model to compare results and calculate the error under various f_{init} and f_{rate} conditions was run to produce useful graphs.

An example of the simulation result for $f_{init} = 10\ Hz$ and $f_{rate} = 500\frac{Hz}{s}$ is shown below in Fig 5-25 and Fig 5-26. The first revolution is under fixed initial frequency condition and the next five revolutions are under ramping frequency condition. The position error can be clearly observed and the difference between TYPE 2 and TYPE 3 is clear.

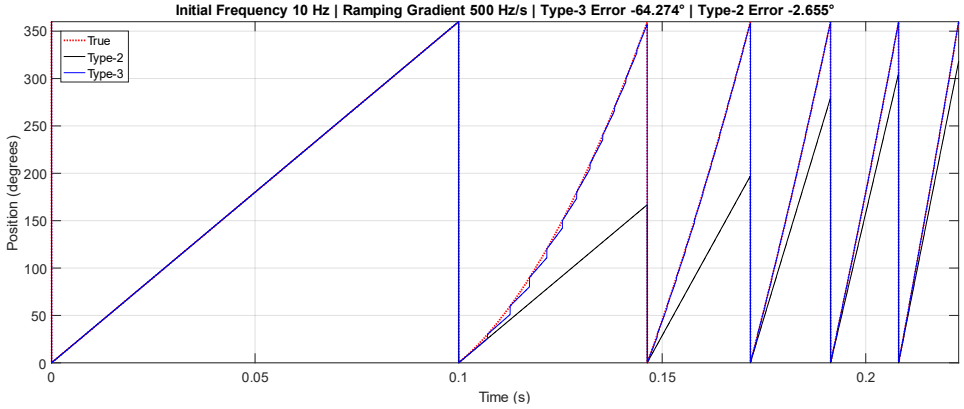


Fig 5-25 Measured phase angle value compared with requested for TYPE 2 and TYPE 3 techniques with rising frequency

The error values for different conditions have been reproduced below.

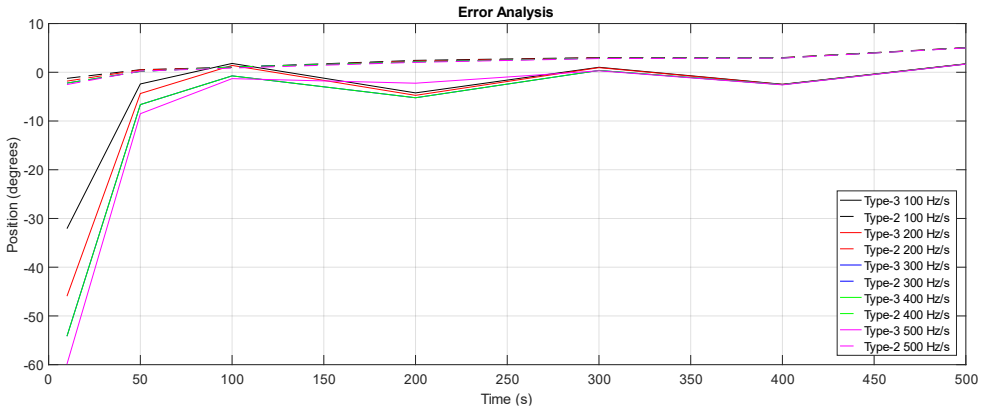


Fig 5-26 Error graph of TYPE 2 and TYPE 3 phase angle computation techniques at different frequency rise rates

Comparing the error values in position value interpretation by the proxy encoder should be compared with the error introduced by the hardware. If Type-2 graph is extrapolated to near 0 Hz, it is observed that error is <10 Hz for the lowest ramping rate (100 Hz/s), which indicates with this P-HIL set up, the standstill start up must be significantly slower (error introduced by hardware was observed to diminish near low frequencies). Exact values for boundary conditions will not studied further in this work, but a foundation has been built to easily conduct this work in the future.

5.6 Summary

This chapter focussed on the techniques and challenges associated with emulating a synchronous motor using a voltage-source-converter (VSC) based power-HIL emulator. Following on from the introduction into general power-HIL simulator hardware in CHAPTER 2 (2.3.4), the Triphase P-HIL simulator hardware was examined in detail and different aspects and their advantages were discussed. The Virtual Circuit Emulation (5.2.2) was discussed as a concept and how this has been implemented in the Triphase simulator was shown.

Above introduction to the available hardware was followed by a brief introduction into theory of control of permanent magnet synchronous machines. Clarke and Park transforms was discussed at a concept level and the advantage of having a DC control

target parameter (d and q current) as opposed to sinusoidal three-phase targets was discussed. The primary reason for requirement of accurate and highly precise motor shaft position information for PMSM control was uncovered as result of this discussion.

The main technical element explored in this chapter was how to faithfully convey the "emulated motor" shaft position information to the test inverter (IUT) using a novel "proxy encoder" solution. The inspiration behind this approach was earlier discussed in CHAPTER 2 (HIL specific case study on how to use FPGA interface to treat high-speed signals using low-speed processing hardware, section 2.2.5.2). The "proxy encoder" was developed in software that sits between the low-speed Triphase HIL real-time PC (or RTT, Real Time Target) and the test inverter (IUT) on a microcontroller-based custom developed interface circuit board. The rest of the chapter focussed on devising a comprehensive experimental plan to identify the weaknesses and limitations of this method. These experiments were performed, and results were analysed.

There were two kinds of errors investigated: **jitter** and **latency**. Jitter was characterised via two types, PART A (fidelity in timing of the A/B/Z pulses as produced by the available hardware) and PART B (fidelity of the reproduced position information as produced by software). Latency was characterised by identifying the average delay between the physical back EMF voltage produced by the Triphase power stage, and the "back EMF voltage phasor information" conveyed by the proxy encoder signal. This average delay was adjusted via phase delay compensation.

CHAPTER 6 Power-HIL Simulation with Real Inverter-Under-Test

6.1 Introduction

The previous chapter covered the technical foundation of the series of experiments done to demonstrate a proof of concept for a real inverter-under-test power-hardware-in-the-loop simulation. Challenges of emulating a real motor using the power stage (controllable generic inverter and configurable filters) were discussed. Communication of the motor shaft position information to the inverter-under-test in real-time using non-hard-real-time communication devices (Beckhoff EtherCAT module) was primarily investigated and issues with delay were eliminated to the best potential using available hardware and associated restrictions.

In this chapter, a prototype development inverter from an unrelated project was used as a test inverter. The reason for using a development inverter was the freedom of software modifications on offer and an exact hardware configuration and inverter topology availability, which would not have been possible with a commercial unit. Full access to the software allowed the proxy encoder software (developed in the previous chapter) to be coded into the inverter very easily. It also allowed knowledge of exact modulation strategy being used (private companies have this as a carefully guarded secret). There were downsides to this approach too. As this inverter was a work in progress, there were bugs and issues in the software and the dq controller was not tuned properly. This is explored in further detail in the discussion sub-chapter later.

6.2 Experimental Setup

The original idea for P-HIL setup with the triphase was to use two power stages in the triphase, i.e., F03 as DC Voltage Source and M30L as 3-phase AC Current Source. Due to reliability issues during commissioning the M30L as a motor emulator, a decision was taken to only use the F03 power stage as a 3-phase AC voltage source. A benchtop

programmable power supply was used to charge the DC bus of the Inverter-under-Test (IUT). This modification does not affect the project plan as motor emulation and proxy encoder development does not change.

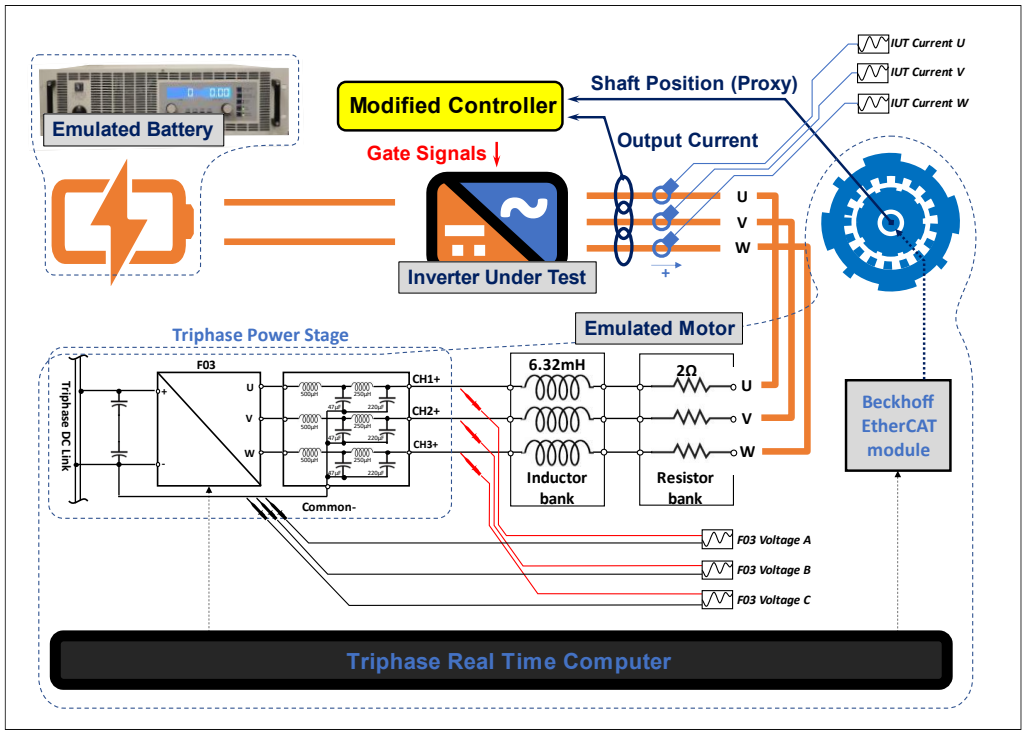


Fig 6-1 Detailed experimental setup for P-HIL experiments

The three main power units in the test setup were as follows:

- DC side of the Inverter-under-Test (IUT) is supplied by a bench power supply.
    - Make and Model – Elektro Automatik EA-PS 81500 30 3U
    - Ratings – 15 kW 1500V 30A
    - Remote Capability – Optional RS232/CAN interface
- IUT & Controller is a single unit (see Fig 6-2 below). It is composed of:
    - Infineon HybridPACK1™ with gate driver circuit which is a commercially available product.
    - Interface Board – Custom PCB with current sensors to detect phase current.
    - Texas Instruments Microcontroller Evaluation board (TM4c1294XL) is the controller which reads the phase currents and proxy encoder information and applies the gate pulses.

- Motor Emulation (Fig 6-3)
    - Triphase F03 Power Stage produces the back EMF phase voltages.
    - Inductor bank of 6.32mH per phase emulates the motor stator inductance.
    - Resistor bank of 2Ω per phase emulates the motor stator resistance.
    - Beckhoff EtherCAT module produces the proxy encoder signal that is read by the IUT modified controller.

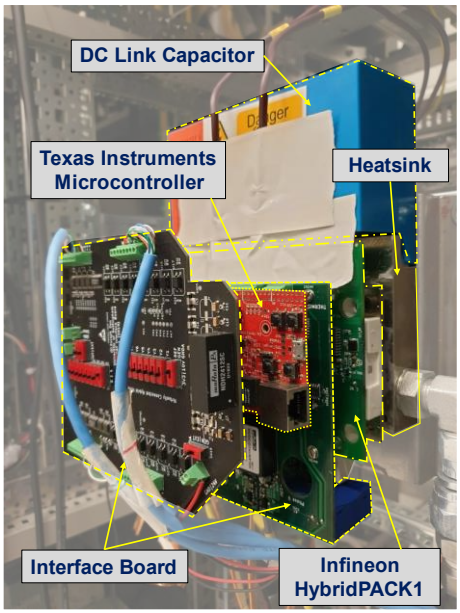


Fig 6-2 IUT – Infineon inverter setup

Two things of import, regarding some assumptions made in these experiments, must be noted here.

First, the test inverter (IUT) controller was an externally provided prototype. The quality of d and q current control was not validated to be of any level of fidelity as this was beyond the scope of the PhD. The results, however, demonstrate that the controller is of an acceptable quality and thus fulfils the purpose of this experiment.

Second, the passive resistor and inductor bank are not representative of a real motor. An automotive traction motor may have an inductance of up to 0.5mH and resistance of 0.1Ω per phase. This assumption does not affect the validity of the experiment as well.

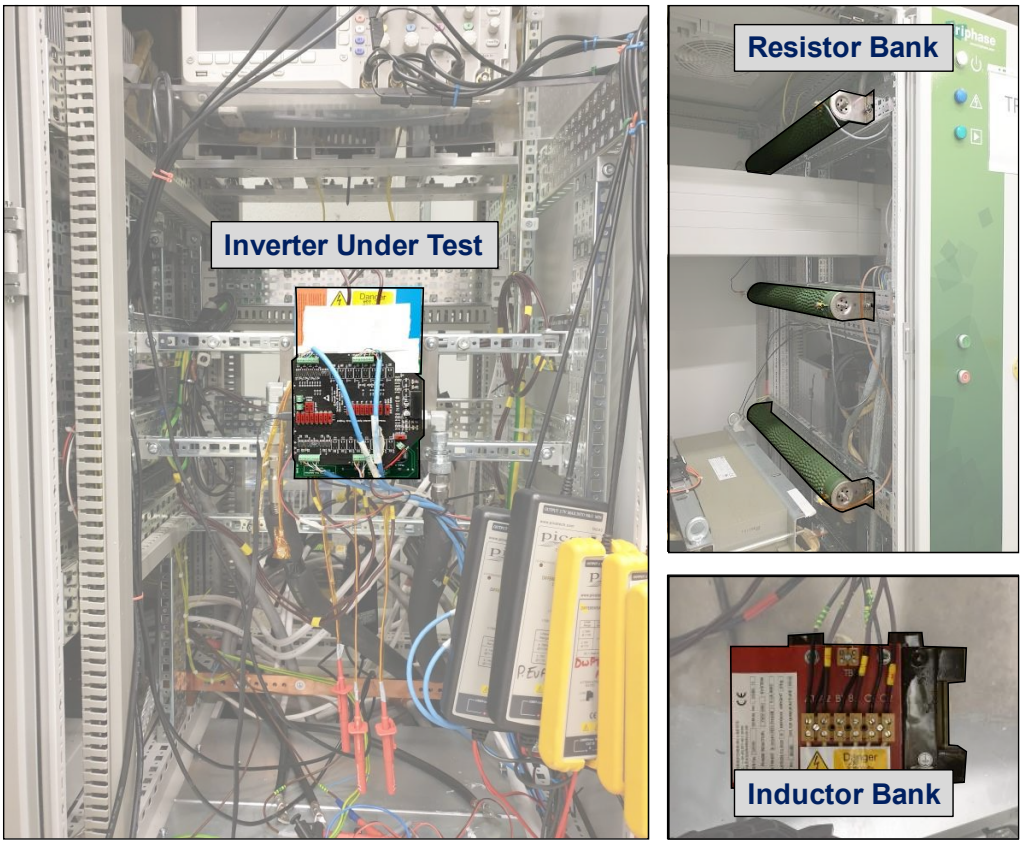


Fig 6-3 Experimental setup pictures

External datalogging (Fig 6-4 and Fig 6-5) is set up to measure currents and voltages. There are two oscilloscopes (Tektronix DPO 2024) which measure the F03 back EMF voltages (phases A, B, C), IUT output currents (phases U, V, W) and Proxy Encoder Z Pulse (duplicated on both scopes to align them together in post-processing). The transducers used are:

- Pico Technology TAO041 – 25Mhz ±70V/700V Differential Probe
- ProSys CP305 – 10kHz ±30A/300A (DC to 20kHz) Hall Effect True RMS transducer

Fig 6-4 External current and voltage measurement instruments for datalogging

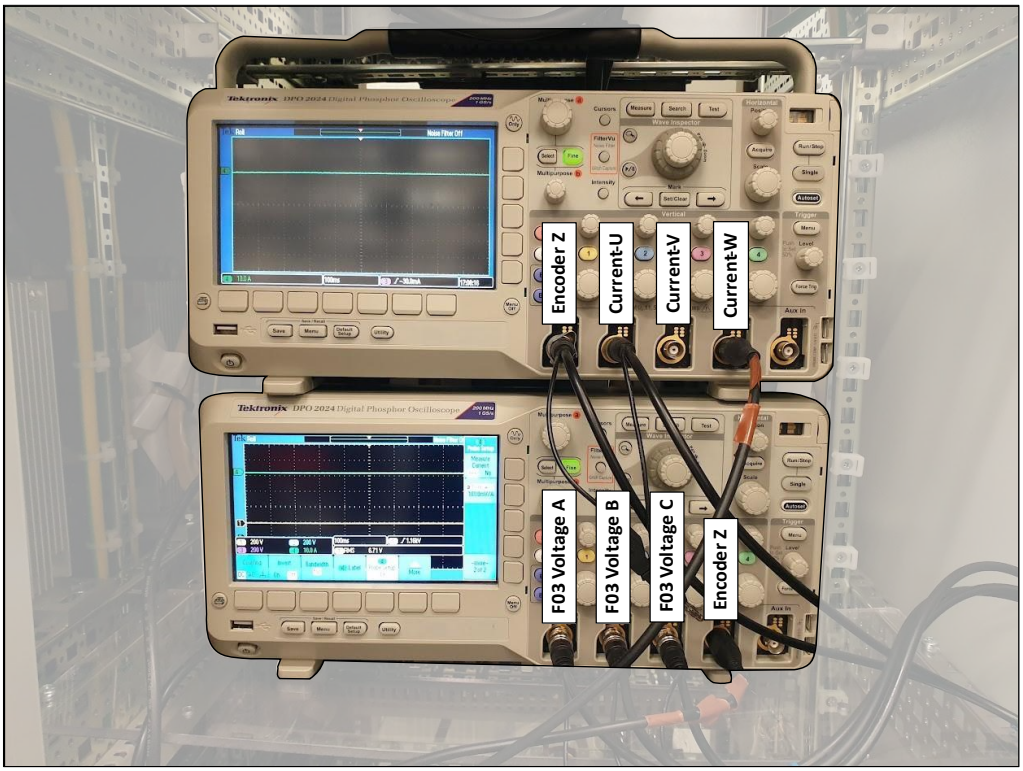


Fig 6-5 Oscilloscopes for reading and logging measurement voltage and current waveforms

6.3 Fixed Frequency Operation

Different fixed frequency steady-state conditions were replicated in the first instance. The IUT was requested to supply $I_d = 0A$ and $I_q = 5A$, i.e., the current supplied by the IUT (or current absorbed by the Motor Emulator) should be in-phase with the back EMF. A fixed voltage ($V_{rms} = 25V$) back EMF was applied (not proportional to frequency as would be in a real motor). This was done to focus on the alignment of the two phasors in this exercise. As the previous chapter, frequencies from 50Hz to 500Hz in steps of

50Hz was applied and the phase offset between back EMF voltage and phase current (supplied by the IUT) was measured for at least 20 revolutions (mean error and standard deviation was calculated). The resultant graphs have been reproduced in Fig 6-6 below (back EMF voltage and IUT phase current in blue and red solid lines, respectively).

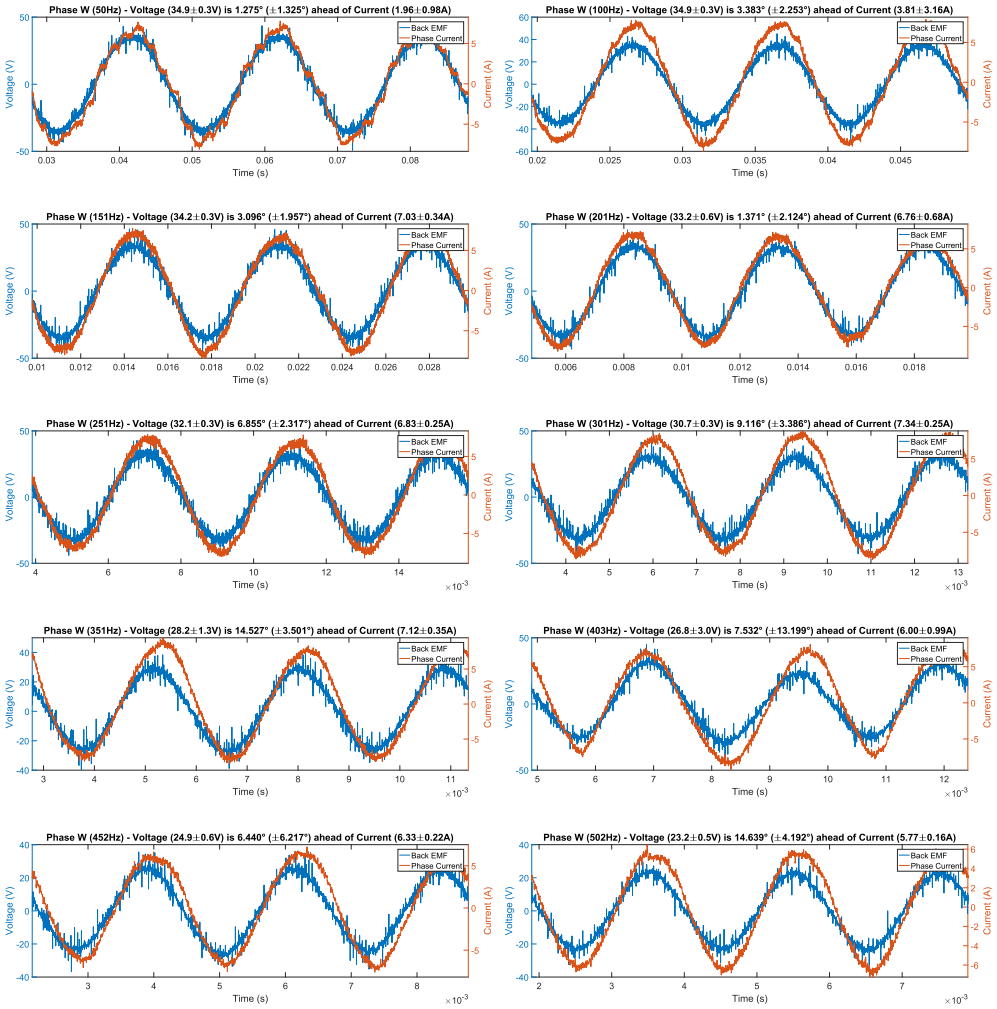


Fig 6-6 Back EMF voltage (Triphase) and phase current (IUT) waveforms at different frequencies

A summary table is produced below (Table 6-1). The mean error is below 10° for up to 300Hz and goes up to 15° for higher frequencies. This shows that motor emulation is working correctly within the error limits introduced by the hardware (EtherCAT module) and software (proxy encoder software) as discussed in the previous chapter.

Frequency	Back EMF		Phase Current		Phase Offset	
	Mean	Std.Dev.	Mean	Std.Dev.	Mean	Std.Dev.
Hz	V	V	A	A	°	°
50	34.63	0.25	6.69	0.21	1.275	1.325
100	34.61	0.44	7.08	0.25	3.383	2.260
150	34.11	0.27	7.24	0.13	3.101	1.963
200	33.16	0.64	7.21	0.37	1.371	2.125
250	31.61	0.21	7.36	0.30	6.860	2.324
300	30.37	0.40	7.87	0.20	9.116	3.376
350	27.95	0.63	7.75	0.27	14.527	3.507
400	26.74	2.63	7.18	0.53	7.540	13.220
450	25.35	0.45	6.58	0.19	6.447	6.231
500	23.56	0.46	5.83	0.13	14.632	4.196

Table 6-1 Phase offset between back EMF voltage (Triphase) and phase current (IUT) at different frequencies

6.4 Variable Frequency Operation

The next step in the verification process is to run variable frequency drivecycles. The back EMF setting was changed back to frequency-dependent operation and a back EMF Constant of $k_{bemf} = \frac{V_{bemf}}{freq} = 0.088$ was set. Three different frequency profiles (Fig 6-7) were devised as follows:

- **RAMP 1**
  - Steady state frequency of 250 Hz
  - Linearly drop to 50 Hz over 1s, i.e., ramp rate of -200 Hz/s
  - Linearly rise to 250 Hz over 1s, i.e., ramp rate of 200 Hz/s
  - Linearly drop to 50 Hz over 0.5s, i.e., ramp rate of -400 Hz/s
  - Linearly rise to 250 Hz over 0.5s, i.e., ramp rate of 400 Hz/s
  - Linearly drop to 50 Hz over 0.25s, i.e., ramp rate of -800 Hz/s
  - Linearly rise to 250 Hz over 0.25s, i.e., ramp rate of 800 Hz/s
  - Linearly drop to 50 Hz over 0.125s, i.e., ramp rate of -1600 Hz/s
  - Linearly rise to 250 Hz over 0.125s, i.e., ramp rate of 1600 Hz/s
- **RAMP 2** (milder version of RAMP 1)
  - Steady state frequency of 250 Hz
  - Linearly drop to 150 Hz over 1s, i.e., ramp rate of -100 Hz/s

- Linearly rise to 250 Hz over 1s, i.e., ramp rate of 100 Hz/s
- Linearly drop to 150 Hz over 0.5s, i.e., ramp rate of -200 Hz/s
- Linearly rise to 250 Hz over 0.5s, i.e., ramp rate of 200 Hz/s
- Linearly drop to 150 Hz over 0.25s, i.e., ramp rate of -400 Hz/s
- Linearly rise to 250 Hz over 0.25s, i.e., ramp rate of 400 Hz/s
- Linearly drop to 150 Hz over 0.125s, i.e., ramp rate of -800 Hz/s
- Linearly rise to 250 Hz over 0.125s, i.e., ramp rate of 800 Hz/s

- **STAIRCASE**
  - Steady state frequency of 50 Hz
  - Staircase steps of 100 Hz, 150 Hz, 200 Hz, 250 Hz
  - Each step steady state duration of 0.45s
  - Each step jump duration of 0.05s, i.e., ramp rate of ±1000 Hz/s

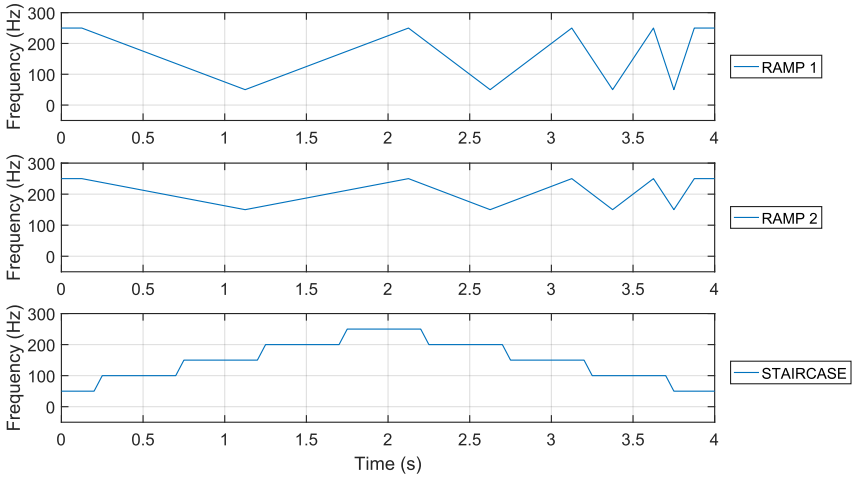


Fig 6-7 Variable frequency profiles applied (RAMP 1, RAMP2, and STAIRCASE)

The drivecycles conducted were as follows:

- **EXPERIMENT 1** – IUT turned off and RAMP 1 frequency profile
- **EXPERIMENT 2** – IUT turned on and RAMP 1 frequency profile
- **EXPERIMENT 3** – IUT turned on and RAMP 2 frequency profile
- **EXPERIMENT 4** – IUT turned on and STAIRCASE frequency profile

In order to provide a real-world reference to the ramping rates tested in these experiments, an example powertrain with the following vehicle details has been used:

- 6 pole pairs
- 10:1 final gear ratio
- 245/45/R19 tires (characteristic of Tesla Model S 2021 model) gives rolling radius of 593 mm and circumference of 3725 mm.

0 to 60 mph uniform acceleration in 5 s gives 12 mph/s average acceleration. Using the circumference of tire, this translates to 1.44 mechanical rotations/second increase every second. This translates to 86 Hz/s of electrical frequency ramping rate. The experiments have been conducted for up to 1600 Hz/s and have been demonstrated (in the following sections) to show negligible error in phase offset for up to 800 Hz/s (RAMP 2).

EXPERIMENT 1

The first experiment (200V DC bus voltage, RAMP 1 drivecycle) was conducted without turning the IUT on. This was used as the control data for the machine emulation performance and to verify that the data recording and processing was working as expected. The complete 4 s of drivecycle has been divided into 4 sections of interest and zoomed snapshots are also shown as described below.

In the following Fig 6-8, part (a) shows the overall view (4s total experimental duration) for:

- Back EMF on top (blue solid line),
- Phase current in middle (red solid line),
- and measured frequency of operation (blue dots for back EMF voltage, red dots for phase current, and black dashed line for the target frequency request) in bottom. The right-side Y-Axis depicts the phase delay (in degrees, black dots) between back EMF voltage and phase current. In EXPERIMENT 1, there is nothing shown as the IUT is turned off, hence no phase current produced. In ideal case scenario, there should be negligible phase delay produced indicating

the IUT is accurately tracking the back EMF voltage via the “proxy encoder” developed in the previous chapter.

Part (b) zooms into the duration from 0.625s to 1.625s, part (c) zooms into the duration from 1.875s to 2.375s, part (d) zooms into the duration from 3.125s to 3.625s, and part (e) zooms into the duration from 3.625s to 3.875s. In parts (b)-(e), back EMF (blue solid line) and phase current (red solid line) have been combined in the same graph (top). The bottom graph (for measured frequency and phase delay between voltage and current) follows the same terminology as part (a).

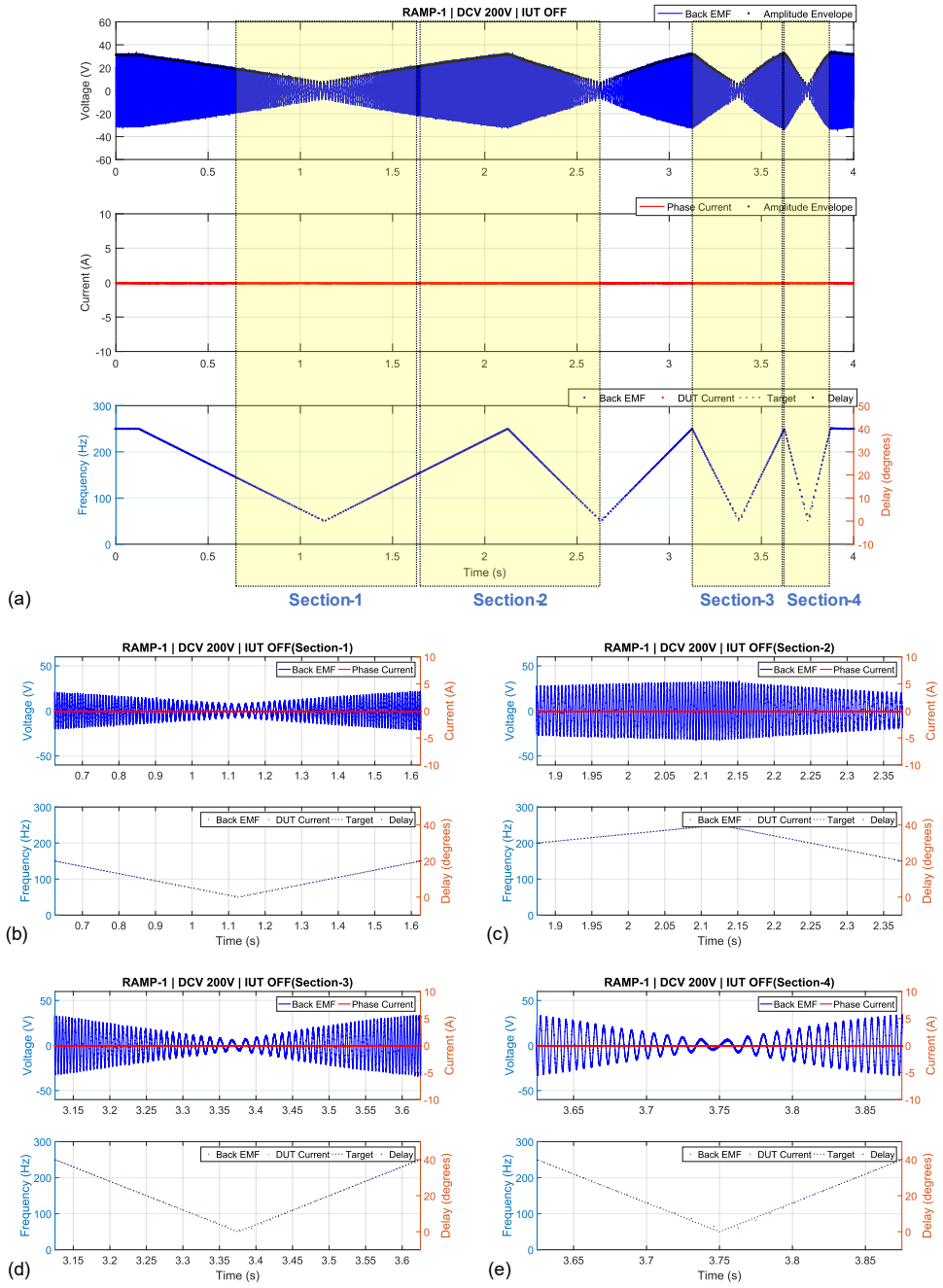


Fig 6-8 EXPERIMENT 1 Results – Back EMF voltage (Triphase) and phase current (IUT)

The back EMF voltage frequency was reproduced very accurately (total error of 0.01% with standard deviation of 1.5%, discussion in next sub-chapter). It is important to note that the mean error is not representative as the deviations occur on both sides of the true value. The standard deviation is a much better metric to indicate the inaccuracy.

The back EMF voltage magnitude produced was reproduced with an accuracy of 4.2% mean error and standard deviation of 4.8%. these values indicate that the voltage

control in the Triphase F03 power stage is not very accurate, especially knowing the fact that in this experiment, there is no load applied (i.e., IUT turned off and no phase current sinking to the Triphase). These results are also discussed in the next sub-chapter.

The next three experiments were conducted with the IUT turned on. Fidelity of the motor emulation results have been investigated in three ways:

- Phase delay between back EMF and phase current.
- Accuracy of frequency profile reproduced (discussed in next sub-chapter).
- Accuracy of voltage profile reproduced (discussed in next sub-chapter).

Fidelity of phase current waveform has not been investigated as this depends on the quality of the inverter-under-test, and falls out of scope of machine emulation.

EXPERIMENT 2

Experiment 2 was conducted at 200V DC Bus Voltage with the RAMP 1 drivecycle and IUT turned on. The structure of the results shown in below in Fig 6-9 is as described in the previous section, and has been repeated below in brevity.

- Part (a) shows the overall view (4s total experimental duration) for:
    - Back EMF on top (blue solid line)
    - Phase current in middle (red solid line)
    - Measured frequency of operation (blue dots for back EMF voltage, red dots for phase current, and black dashed line for the target frequency request) in bottom. The right-side Y-Axis depicts the phase delay (in degrees, black dots) between back EMF voltage and phase current.
- Part (b) zooms into the duration from 0.625s to 1.625s
- Part (c) zooms into the duration from 1.875s to 2.375s
- Part (d) zooms into the duration from 3.125s to 3.625s
- Part (e) zooms into the duration from 3.625s to 3.875s

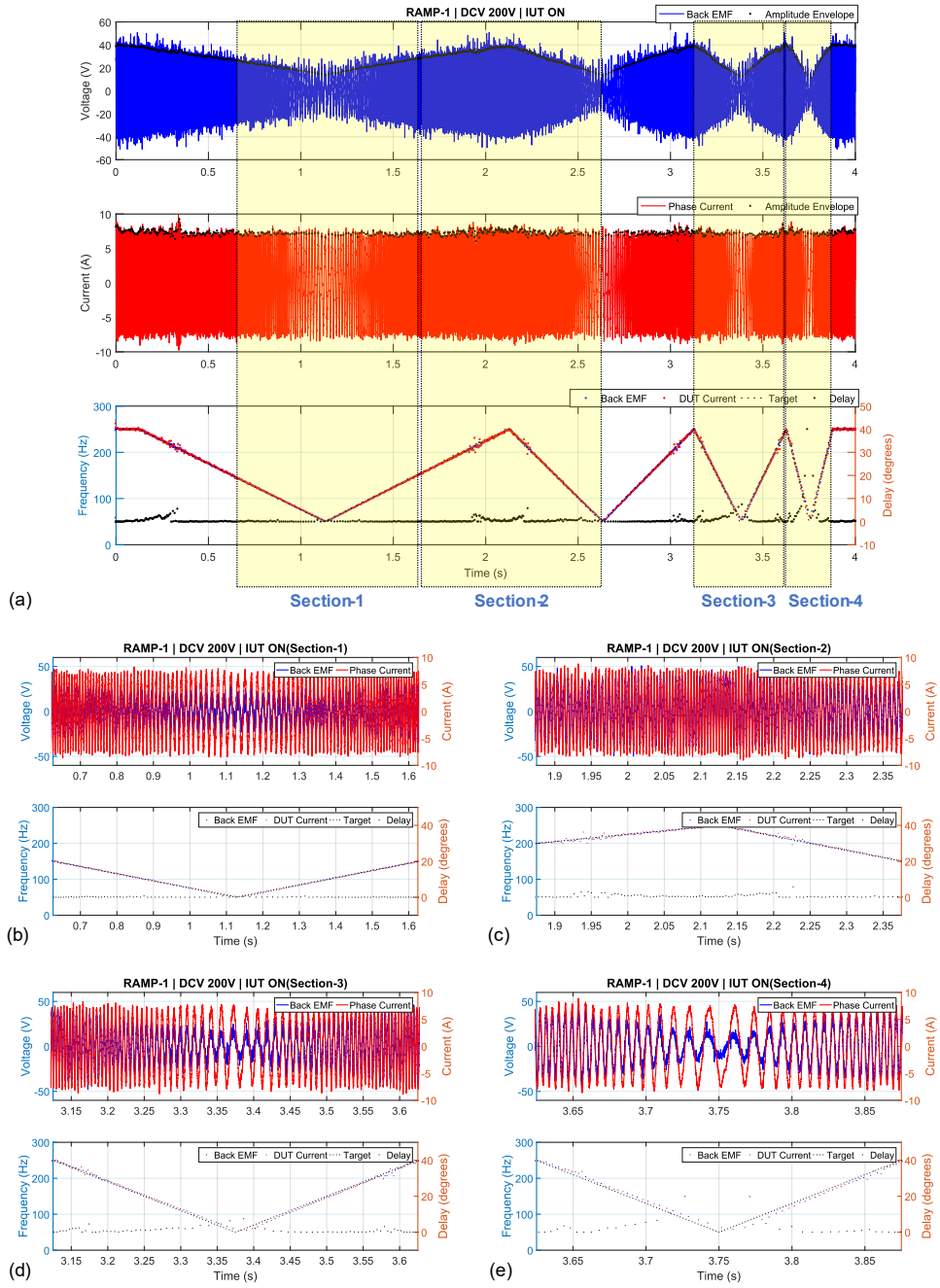


Fig 6-9 EXPERIMENT 2 Results – Back EMF voltage (Triphase) and phase current (IUT)

As observed in the Fig 6-9 above, phase deviation between back EMF and phase current (bottom graphs in parts (a)-(e)) is within 1° for majority of the drivecycle, with occasionally going close to 8° when the ramping switches direction in the first three sections (softer ramping rates). Section 4 (ramping rate of 1600 Hz/s) shows significant error in the region of 20° and only once touching 40° at the point of inflection.

Looking at the zoomed snapshot closely, it is clear the two sinusoids are very well aligned, and the anomaly looks like a post-processing calculation approximation issue. The way the phase offset was calculated was to identify the middle of each wavelength (single wave between consecutive zero crossings) and compare the time difference. This approximation holds reasonably accurate in most cases but when the rate of change of frequency is high enough to have definite change in the waveform within each wavelength, the approximation does not hold true.

This large error occurs only a few times (once ~40° at the point of inflection and twice ~20° in the two adjacent points) at and near the point of inflection during the highest ramping rate. 1600 Hz/s is realistically applicable for very small and high-power motors which are not seen in standard automotive applications. This anomaly is hence not of consequence for this application and shall not be analysed further.

EXPERIMENT 3

Experiment 3 was conducted at 200V DC Bus Voltage with the RAMP 2 (less aggressive version of Ramp-1, ramping rates are halved) drivecycle and IUT turned on. The structure of the results shown in below in Fig 6-10 is as described in the previous section, and has been repeated below in brevity.

- Part (a) shows the overall view (4s total experimental duration) for:
    - Back EMF on top (blue solid line)
    - Phase current in middle (red solid line)
    - Measured frequency of operation (blue dots for back EMF voltage, red dots for phase current, and black dashed line for the target frequency request) in bottom. The right-side Y-Axis depicts the phase delay (in degrees, black dots) between back EMF voltage and phase current.
- Part (b) zooms into the duration from 0.625s to 1.625s
- Part (c) zooms into the duration from 1.875s to 2.375s
- Part (d) zooms into the duration from 3.125s to 3.625s

- Part (e) zooms into the duration from 3.625s to 3.875s

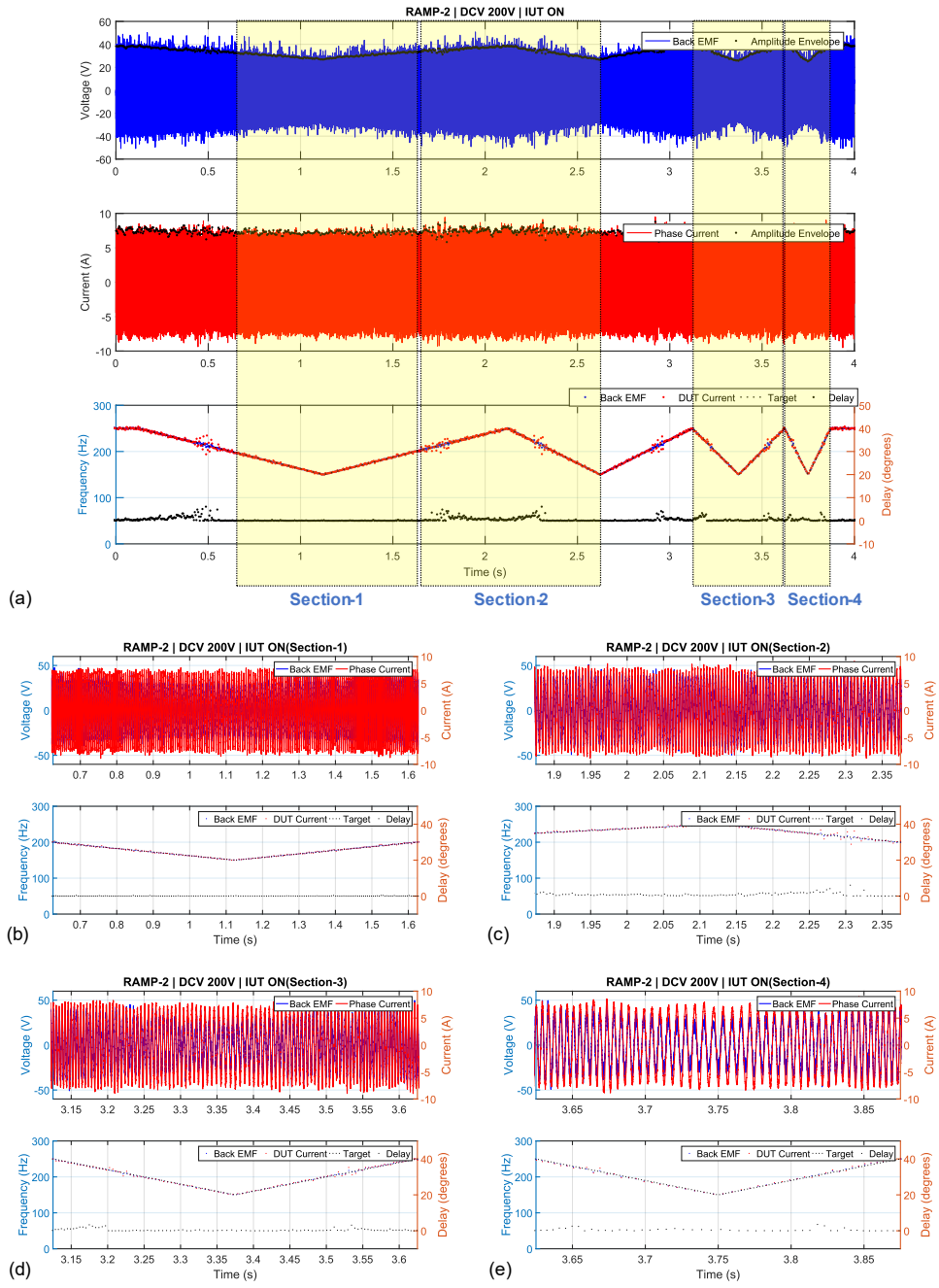


Fig 6-10 EXPERIMENT 3 Results – Back EMF voltage (Triphase) and phase current (IUT)

As observed in the Fig 6-10 above, phase deviation between back EMF and phase current (bottom graphs in parts (a)-(e)) is within 1° for most of the region as expected. The maximum deviation is within 5° for all the ramping rates which is expected from the previous experiment.

EXPERIMENT 4

Experiment 4 was conducted at 200V DC Bus Voltage with the STAIRCASE drivecycle and IUT turned on. This was done to investigate how can the system cope with sharp ramping rates followed by period of fixed frequency. the steps are jumps of 50 Hz in a matter of 0.05 s that translates to 1000 Hz/s. The structure of the results shown in below in Fig 6-11 is as described in the previous section, and has been repeated below in brevity.

- Part (a) shows the overall view (4s total experimental duration) for:
    - Back EMF on top (blue solid line)
    - Phase current in middle (red solid line)
    - Measured frequency of operation (blue dots for back EMF voltage, red dots for phase current, and black dashed line for the target frequency request) in bottom. The right-side Y-Axis depicts the phase delay (in degrees, black dots) between back EMF voltage and phase current.
- Part (b) zooms into the duration from 0.625s to 1.625s
- Part (c) zooms into the duration from 1.875s to 2.375s
- Part (d) zooms into the duration from 3.125s to 3.625s
- Part (e) zooms into the duration from 3.625s to 3.875s

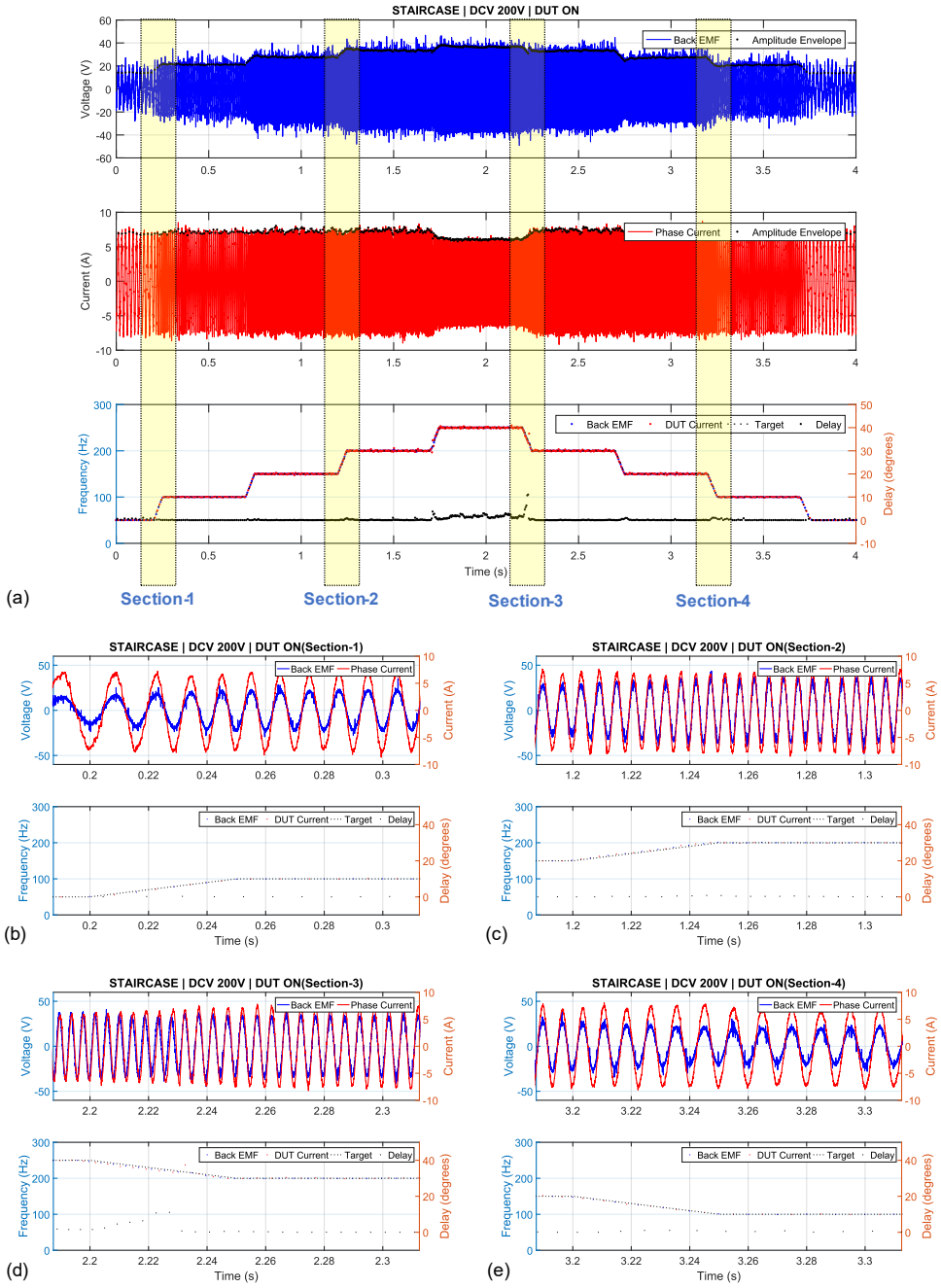


Fig 6-11 EXPERIMENT 4 Results – Back EMF voltage (Triphase) and phase current (IUT)

As observed in the Fig 6-11 above, phase deviation between back EMF and phase current (bottom graphs in parts (a)-(e)) is within 5° for majority of the drivecycle. The delay rises to around 12° at the ramping up and ramping down regions at the highest frequency step, i.e., 250 Hz.

6.5 Discussion

In the previous sub-chapter, the phase delay between the back EMF voltage (produced by the machine emulator) and phase current (produced by the IUT in response to the proxy encoder information generated by the machine emulator) was measured. This sub-chapter will focus on the findings on:

- How closely the machine emulator can follow the demanded frequency profile
- How closely the machine emulator can follow the demanded voltage profile

Before investigating the above, it would be useful to identify the maximum frequency that would be usually observed in standard automotive application. Using the same vehicle specification as earlier (Tesla Model S 2021 model), a top speed of 120 mph would require an electrical frequency of 864 Hz. The previous chapter showed that the proxy encoder system is reasonably accurate up to 350 Hz and requires switching over to a different strategy to reach 500 Hz. It was also seen that with the Triphase CPU-based real-time computer, an ideal maximum frequency of 833 Hz can be achieved (realistic maximum would be much less due to practical considerations).

This shows that the current hardware is restricting the maximum frequency of real-world applications to about 50 mph. There are alternative solutions available currently (FPGA-based real-time computer and encoder signal generation) that should be investigated in future work. This PhD project has limitations in terms of hardware and hence, the best possible experimental conditions have been conducted in this work.

6.5.1 Frequency Profile Fidelity

Experiment 1 conducted was with RAMP 1 and IUT turned off. The frequency profile of back EMF voltage generated by Triphase has been depicted in Fig 6-12 below (black dots) against a backdrop of requested frequency profile (dotted black line). The instantaneous error has also been depicted (blue dots) and can be seen to be rising at the points of inflection only during the low frequency regions, i.e., when ramping rate changes from negative to positive (error is close to 0% at the inflection points when ramping rate changes from positive to negative). This is an interesting feature of the

Triphase and is caused by the internal voltage controller. Since the error is within acceptable range and the internal controller of Triphase is not accessible, this finding shall not be investigated further.

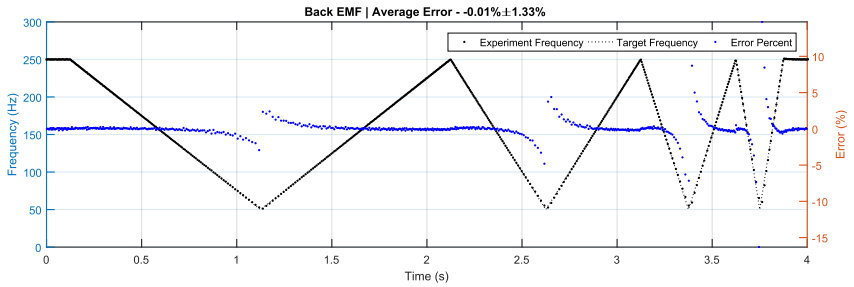


Fig 6-12 EXPERIMENT 1 – Error in reproduced frequency of back EMF voltage (Triphase)

Experiment 2 is identical to Experiment 1 but with the IUT turned on. The frequency profile of both back EMF voltage and phase current (large red dots) has been depicted in Fig 6-13 below against the true frequency (black dotted line) with the instantaneous error percent values (large blue dots for Voltage and large red dots for Current). The standard deviation of frequency for voltage waveform has risen from 1.33% to 2.52% purely as a result of current being sinked in the Triphase power stage. The standard deviation of frequency for current waveform is 2.71% and is similar to the voltage waveform as expected. As discussed in the previous sub-chapter, the average error percent value is of less significance as the error can be seen to be distributed equally around the true value, hence positive and negative values counter each other leading to a small average value.

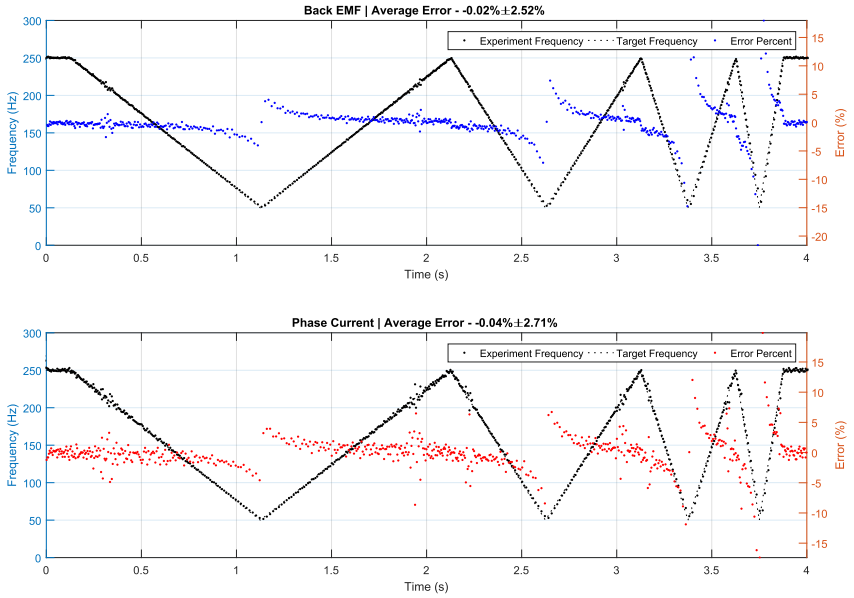


Fig 6-13 EXPERIMENT 2 – Error in reproduced frequency of back EMF voltage (Triphase) and phase current (IUT)

Experiment 3 uses a less steep version of Experiment 1 frequency profile (Fig 6-14 below). The instantaneous frequency error for back EMF voltage is constrained within 3% and the standard deviation is 0.53%. Instantaneous frequency error for phase current is mostly limited to 4% but rising to 8% at the 220 Hz mark. The standard deviation of phase current error is 1.56%.

It is interesting to note that there is a small but significant blip in instantaneous error value at around 220 Hz. This is observed in all three experiments with the IUT turned on. As this is only observed in the three experiments with the IUT turned on, it is reasonable to believe this is caused by something in the control implementation in the IUT. As the error is small and the IUT controller is out of scope for this PhD, this issue will not be investigated further.

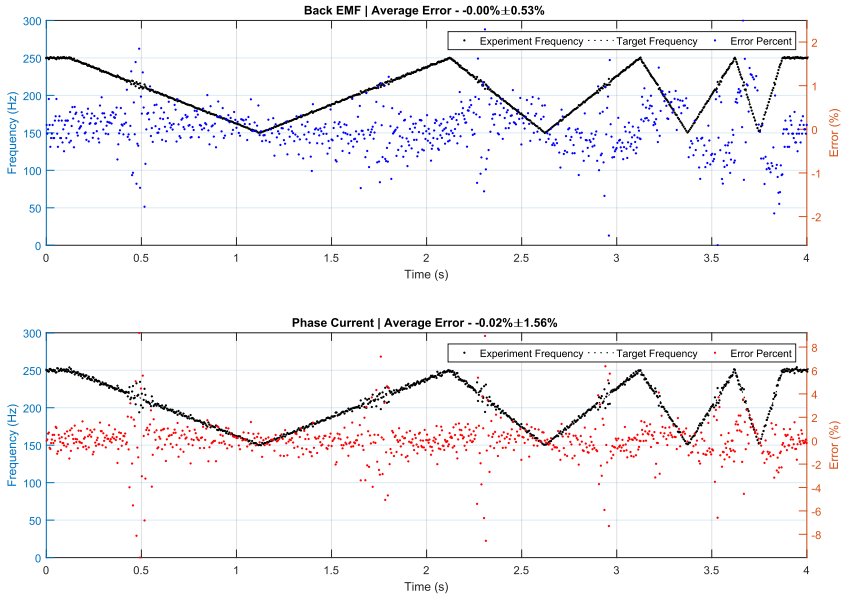


Fig 6-14 EXPERIMENT 3 – Error in reproduced frequency of back EMF voltage (Triphase) and phase current (IUT)

Experiment 4 is the last experiment conducted in this chapter. This uses a staircase frequency profile (Fig 6-15 below). The instantaneous frequency error for back EMF voltage is seen to be within 2% with a standard deviation of 0.58%. The instantaneous frequency error for phase current is seen to be within 4% with a standard deviation of 1%.

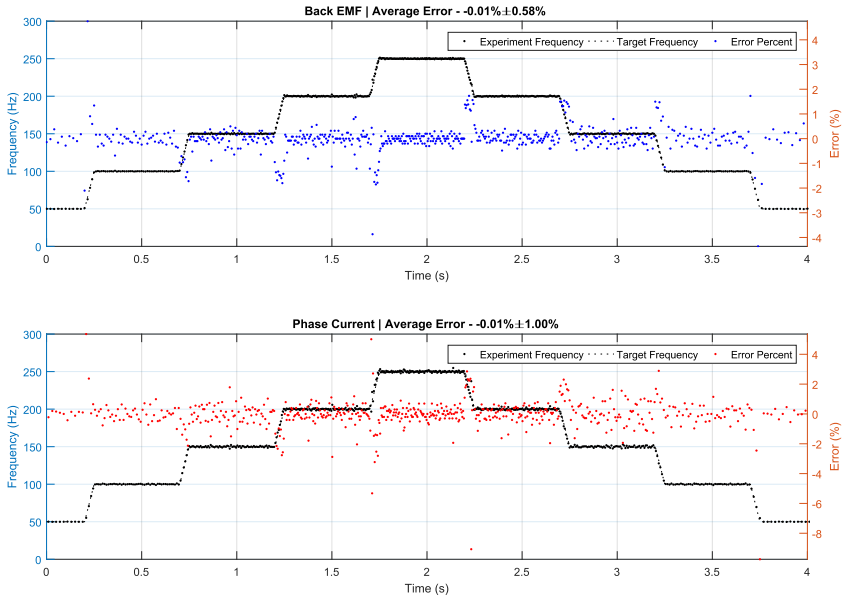


Fig 6-15 EXPERIMENT 4 – Error in reproduced frequency of back EMF voltage (Triphase) and phase current (IUT)

6.5.2 Back EMF Voltage Fidelity

Back EMF Voltage, in ideal conditions, is a fixed voltage value proportional to the frequency of operation. In a generic permanent magnet synchronous machine (PMSM), as the rotor shaft rotates faster, the back EMF voltage generated proportionally rises as the frequency rises. The exact relation between voltage and frequency is not as straightforward as a proportionality constant and there are non-linear factors at play, but for simplicity of experiments, this has been adopted, i.e., back EMF constant of 0.088 ($k_{bEMF} = 0.088$). In real application, nonlinear effects can be equally implemented as complex equations or look-up tables, but as a first step, it is important to investigate the fidelity of the voltage set point controller in the machine emulator power stage.

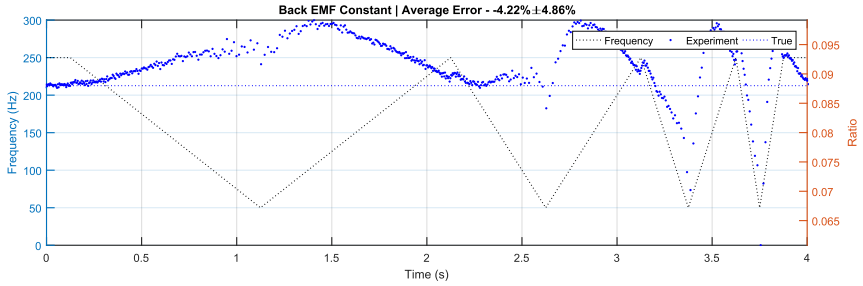


Fig 6-16 EXPERIMENT 1 – Error in back EMF voltage (Triphase) magnitude

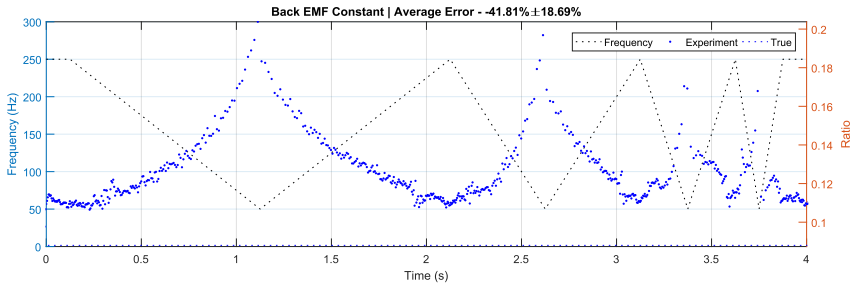


Fig 6-17 EXPERIMENT 2 – Error in back EMF voltage (Triphase) magnitude

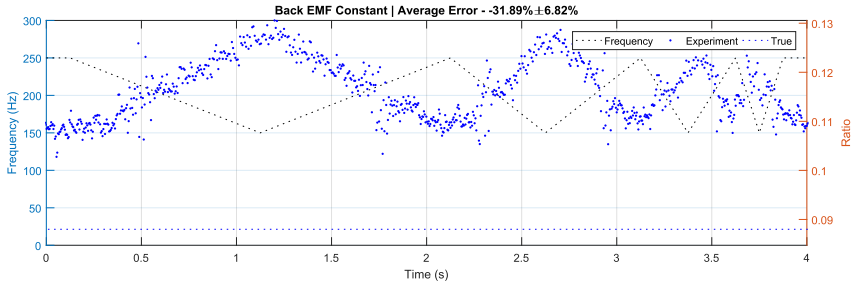


Fig 6-18 EXPERIMENT 3 – Error in back EMF voltage (Triphase) magnitude

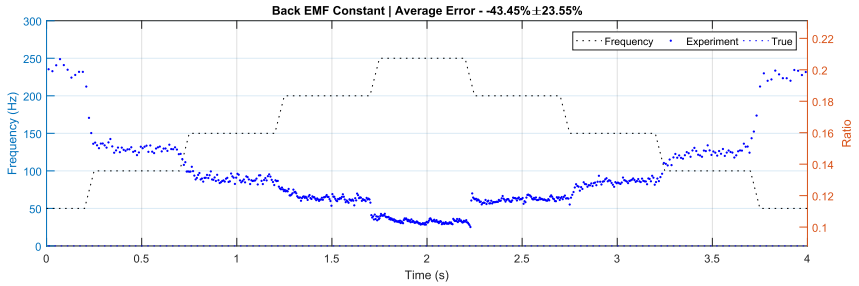


Fig 6-19 EXPERIMENT 4 – Error in back EMF voltage (Triphase) magnitude

The four experiments are depicted in Fig 6-16-Fig 6-19 above (target frequency in black dotted line, measured back EMF constant ratio in blue dots, target back EMF constant in blue dotted line). The ideal back EMF Constant value of 0.088 is (dotted blue line) compared against reproduced value. It can be clearly observed that there is a large error in the way the voltage phasor is reproduced by the Triphase power stage.

In Experiment 1 (Fig 6-16), without the IUT turned on, the constant veers away from its true value (i.e., 0.088) by up to 10% in the regions where the frequency changes sharply from high positive ramping rate to high negative ramping rate. The standard deviation is ~5%. In Experiment 2 (Fig 6-17), the k_{bEMF} instantaneous value rises by more than 125% (from 0.088 to ~0.2) in the lowest frequency regions. The standard deviation is 18.69%. Mean error value of 42% indicates that the experimental value is heavily biased on positive side of the true value, i.e., the reproduced voltage is significantly higher than what is demanded. In Experiment 3 (Fig 6-18), the k_{bEMF} instantaneous value rises by 47% (from 0.088 to ~0.13) in the lowest frequency regions. The standard deviation is 6.82% and the mean error is 32%. In Experiment 4 (Fig 6-19), the k_{bEMF} instantaneous

value rises by 125% (like in Experiment 2) at the lowest frequency regions and gets closest to the true value (~0.1) at the middle of the profile, i.e., when the frequency is at its highest value.

Triphase simulator is unable to control the AC voltage waveform accurately and is significantly varying with the frequency and the total current being sinked. On initial thoughts, the controller does not appear to be very well tuned and optimised for AC operation. This is the key area of improvement that is required in the Triphase P-HIL rig as it stands. The voltage controller software is inaccessible and hence has not been investigated in this PhD.

6.6 Summary

This is the final technical chapter of the thesis and puts the novel proxy encoder solution in action using a test inverter (IUT) connected to the Triphase P-HIL emulator rig. The Triphase was used to produce a controlled three-phase sinusoidal voltage waveform emulating the back EMF voltage of an electric motor. This was mated to the test inverter (IUT) via a three-phase RL passive filter circuit, to emulate the motor winding inductance and resistance. The IUT was sent the “motor” shaft position information using the method developed in the previous chapter, i.e., low-speed physical signal sent by the Triphase real-time PC using the Beckhoff EtherCAT (section 5.2.4) IO interface and subsequently converted to an equivalent high-speed position information.

Experimental validation was conducted under fixed-frequency (**steady-state**, section 6.3) and variable-frequency (**transient**, section 6.4) conditions with a fixed q-axis current and zero d-axis current.

The **steady-state** results showed good reconciliation between the phase angles of back EMF voltage (produced by Triphase) and phase current (produced by IUT) up to 300 Hz, results are summarised in Table 6-1 above. There are two main learnings from these studies:

- First, the waveforms accuracy (measured by how close to zero is phase angle offset between back EMF voltage and IUT phase current, since d-axis current demanded

is zero) is limited by the accuracy offered by the proxy encoder. It was seen in the previous chapter that at frequencies higher than 250 Hz, the efficacy drops significantly. For real-life automotive applications, much higher frequency of operation is required. The solution proposed is to investigate alternate ways to transfer the low-speed position information (perhaps via digital communication medium like CAN) directly to intermediary block (that converts this to accurate high speed encoder signals).

- Second, more rigorous tests with non-zero d-axis current target values are required to properly validate this power-HIL setup. These new experiments would produce more fruitful insights only after the above recommendation has been implemented to an extent.

The **transient** experiments were performed for three different frequency profiles that adequately covered different frequency rise/fall rates that is possible in standard automotive applications. The discussion section that followed clearly demonstrated that the frequency profile of back EMF voltage and IUT phase current waveforms are accurate within hardware limitations. These experiments however, exposed a significant limitation in the Triphase HIL simulator, i.e., the voltage controller efficacy. It was seen that the AC voltage control was significantly below expectations with changing frequency and sinked current (from the IUT). As modifying the controller for this commercial product was not feasible (within the scope of this PhD), this aspect should be investigated in the future. As accurate AC voltage control under various frequency transients and current loading conditions has been achieved up to very high standards for several years (example product from ETPS in [140]), this hurdle should not pose a significant problem to overcome.

CHAPTER 7 Conclusions

7.1 Introduction

This PhD work was conducted as part of the UK national drive towards Net-Zero and was partly funded by the Advanced Propulsion Centre [14]. The parent project (called Virtually Connected Hybrid Vehicle, or VCHV [15]) was originally intended to demonstrate a proof-of-concept testbed with multiple geographically distributed HIL centres for a hybrid automotive powertrain. A total of 8 PhDs were recruited from the following UK universities:

- University of Nottingham (Power Electronics centre)
- University of Warwick (Batteries centre)
- University of Newcastle (Electric Machines centre)
- University of Bath (Engines and Turbochargers centre)
- University of Loughborough (Vehicle Controller centre)

It was recognized in the first year (2017) that the overall aim was very ambitious and required significantly more funding towards post-doctoral researchers and dedicated technicians, and most importantly, much closer collaboration among the universities towards a common goal. Unfortunately, this was not realised to its full potential and the PhD recruitment from other universities also took up to 2 years to be completed. Owing to all these difficulties, the original ambition was not realised completely during the parent project duration.

However, this PhD has indisputably demonstrated that there is immense potential in a remote, multi-centre, real-time, co-simulation testbed for automotive powertrain prototype testing and validation. CHAPTER 1 discussed why and how this technological capability would benefit the automotive industry and significantly contribute towards reduction of time and cost for powertrain development process. The industry is very horizontally integrated and tightly connected to a vast network of suppliers, not just for

manufacturing, but also for research and development; this makes this capability even more relevant towards building sustainable and collaborative industry.

The motor inverter is the key component in an electric powertrain as it acts as the control centre for torque production (motive and regenerative) using PWM “switching” of DC power to produce controlled and well-defined three-phase AC power (battery and electrical machine are simply responsible for sourcing/sinking the power). This PhD work at Nottingham was focussed on developing the remote testbed capability for the motor inverter in a generic hybrid powertrain. This was done in two parts.

First, development of the remotely operable and controllable “digital twin” of a generic electric powertrain, that could be deployed in other HIL centres to faithfully replicate the physical hardware (battery, motor, and motor inverter/controller) performance given the limitations of communication medium between centres. CHAPTER 4 covers the work conducted in this area.

Second, development of the local power-HIL centre that can be used to simply “drop” a generic motor inverter prototype in the simulator, and be tested on various drivecycles, with a battery and motor “emulator” sourcing/sinking real power to this test inverter. CHAPTER 5 and CHAPTER 6 cover the work conducted in this area.

The work done in these three technical chapters follow a comprehensive review of the literature. This is split into two review chapters.

CHAPTER 2 focusses on the hardware-in-the-loop (HIL) and systematically breaks down this multi-faceted and multi-disciplinary concept into smaller technical elements, and then builds it back up into the three main kinds of HIL, i.e., controller HIL (or C-HIL), power HIL (or P-HIL), and geographically distributed HIL (or GD-HIL). Each type has been supported by several case studies of real-life research projects from around the world, and specific technical learning outcomes have been taken away from each and used to formulate the plan for this PhD.

CHAPTER 3 focusses on the real implementation of the “digital twin” of a generic electric powertrain. The need for deploying a digital twin at remote HIL centres

(especially for electric system replication) has been identified. The best method of implementing this has also been identified to be the Multi-Frequency Averaging (MFA) method that solves the remote system in the frequency domain (using Dynamic Phasors) instead of the traditional time domain.

The remainder of this chapter has been split into two parts. First, the novelty elements of this PhD have been summarised in section 7.3 that discusses some qualitative and quantitative results from the three technical chapters. Second, as with any real-life projects, several weaknesses were uncovered in the duration of the project, either due to technological limitations or restrictions imposed by the available hardware. This has been summarised in section 7.4 followed by proposal for additional work required to deliver a fully-function turnkey solution testbed for direct commercial research applications.

7.2 Discussion

It is critical to appreciate that this PhD work required understanding of a wide array of engineering topics that are not always related to each other. As a result, there was a risk of getting side-tracked and losing sight of the overall ambition of the project. However, this resulted in a good outcome of exploring several areas of interest in significant detail.

7.2.1 Hardware-in-the-Loop

HIL is a wide area of research and was broken down into distinct kinds of application – local HIL and power-HIL (P-HIL) applications, wherein a prototype hardware is tested in a virtual environment, and geographically-distributed-HIL (GD-HIL), wherein multiple HIL/power-HIL centres are tied together over the internet for real-time co-simulation. the latter kind offers distinct set of challenges – understanding networking and communications where Internet of Things (IoT) and different network protocols like TCP and UDP were investigated, and state convergence of multiple dynamical systems to ensure the global simulation stays converged.

In order to build a GD-HIL testbed, it was found logical to divide the project into two parts. *First*, local P-HIL test centre to test a generic automotive power converter that involved emulating an electric machine and communicating the emulated motor shaft position information to the power converter being tested. *Second*, developing a Digital Twin of the complete local P-HIL real-and-virtual test environment that can be deployed in a remote location and periodically realigned (this part is further discussed in the next sub-section).

In the *first* part, a machine emulator was implemented with the Triphase P-HIL rig that was available for this project. This “motor” was run with the test inverter up to 300 Hz in fixed and variable frequency drivecycles. The phase angle deviation between “motor” back EMF and test inverter phase angle was requested to be zero degrees and was found within 4% error margin.

There were two main issues identified. First, back EMF voltage magnitude showed very high error and dependency on inverter current being sinked/sourced. This issue was identified to be trivial and would require liaising with the equipment manufacturer to debug the voltage controller issue. Second problem identified was that the maximum frequency of operation was found to be much lower (300 Hz) than what would be required to undertake real-world automotive drivecycles (up to 864 Hz). This issue was an equipment limitation.

7.2.2 Multifrequency Averaging and Dynamic Phasors

Fourier Series expansion, a widely known and practiced mathematical concept to dissociate any waveform into its constituent sinusoids, can be used on dynamically variable waveforms as well that are not strictly periodic, rather, they are treated as quasi-periodic over a moving time window. This is called Dynamic Phasor concept and has been practiced in the academic community for several decades. The same, for variable frequency operation, is a relatively new concept, which has been investigated in detail.

Based on the learnings from the above, a novel MFA-based model was developed that can be operated on variable frequencies as well. Two important properties, i.e.,

differentiation with respect to time, and multiplication of two signals, were also discussed. A unique method of multiplying two signals (second property) that helps in significantly reducing the computation time, was developed, and assessed in this work.

MFA-based models were shown to take about 6% of the total computation time than its equivalent time-domain switching model. Various levels of fidelity (MFA model was set to distinct levels) were explored and demonstrated in this work, for fixed and variable frequency operation.

7.2.3 Digital Twin

This concept has recently gained exponential popularity due to the recent rise of Internet of Things and data-based product development. A standard Digital Twin was found to be composed of three primary parts: physical product, digital replica of the physical product, and bi-directional communication system between the two.

The MFA-based model was developed to become the Digital Twin of the local P-HIL real/virtual environment being tested. This model was developed in a way so that it can be configured to become the digital replica to varying levels of fidelity as needed in different instances and applications. The bi-directional communication is something that has not been fully developed in this work (also called State Convergence in this chapter).

7.3 Novel Contributions to Research

The novelty in this PhD can be summarised into three main elements:

- Multi-Frequency Averaging (MFA) methodology has been applied to produce a highly robust and customisable Digital Twin of a generic electric powertrain applicable to dynamically variable frequency operation. The distinctive advantages of this model are:
    - High speed of computation that is optimised for real-time operation
    - Quickly selectable “output fidelity” by choosing appropriate frequency elements as required for specific application.

- An original method of multiplying two waveforms in the frequency domain as an alternative to convolution operation. This method is purpose-designed to work in the proposed MFA-based Digital Twin and is an essential contributor to its high-speed performance.
- An original method of conveying highly precise motor shaft position information (required for synchronous motor control) using lower speed CPU-based processing system.

7.4 Future Work

This PhD has produced a solid foundation for developing a multi-HIL co-simulation testbed that potentially helps reduce the total development expense and time significantly. This work has shown that a fully functioning testbed for commercial application is possible. However, some more work is needed to realise this to its full potential.

Digital Twin

- Further validation experiments of MFA model are required to fully ratify this as a reliable and robust Digital Twin. Space Vector Modulation switching technique was introduced in CHAPTER 4, but not completed as customisable zero-vector placement requires further mathematical development and experimental validation. Finally, the Twin should be validated against physical data. This step is ideally completed in conjunction with another research project particularly focussed on inverter development. This allows unfettered access to different switching and control strategies which was missing in this PhD project.
- Closed-loop control in the frequency domain was not investigated and validated in this PhD work. Controller design is a key element that prototype inverters would require to be evaluated on this HIL testbed, and this is something that should be explored in detail.

Local Motor Emulation

- The proxy encoder solution developed in CHAPTER 5 is a viable solution, but not the best. A simple alternative is to use digital communication protocol like CAN or Serial to transmit real-time position information to a microcontroller- or FPGA-based intermediate stage that reads this signal and produces a realistic digital encoder signal.
- This can be taken a step further to build a complete suite of options for testing prototypes at sequential stages of development. For example, a client could come with an inverter only, or an inverter + gate driver, or inverter + gate driver + controller. This is explained in Fig 7-1. What is ideally needed, to produce a compact microcontroller or FPGA-based hardware that is tied with the Triphase real-time PC and can produce:
    - Realistic encoder signals (A/B/Z signals) or analog resolver signals (discussed in section 5.4.1) that caters to all kinds of interfaces in the market today.
    - A gate driver stage that can be optionally used if the prototype consists of the inverter only.
    - A controller stage that can be optionally used if the prototype consists of the inverter and gate driver, but not controller. This internal controller must be fully programmable so the client can test under various conditions and switching strategies to find the best configuration.

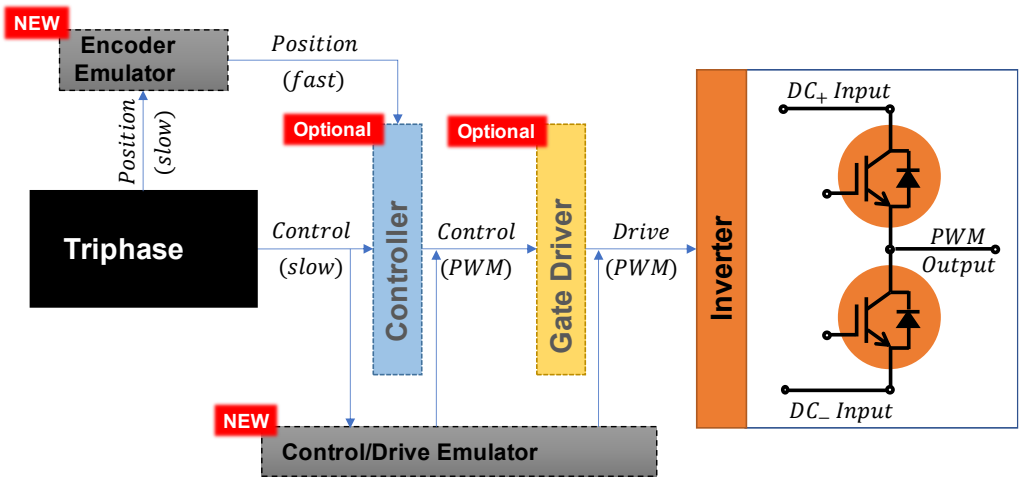


Fig 7-1 End goal for local Power-HIL testbed for prototype inverter testing at various stages of development (inverter only, or inverter and gate driver, or controller included too)

- AC voltage control in the Triphase simulator had several identifiable weaknesses, which need to be ironed out. Once there is good confidence in the simulator power stage in terms of its voltage and frequency following capability, the same experiments done in CHAPTER 6 should be conducted at different values of I_d and I_q to check field-weakening regions as well. In this PhD work, $I_d = 0$ condition has been experimentally validated which meant the back EMF voltage and IUT phase current were required to be in-phase that is only a special condition, not comprehensive.

State Convergence

- After the two ends of geographically distributed HIL setup (i.e., local motor emulation and Digital Twin) is fully fleshed out and validated, the next step is to develop a strategy to keep the Twin aligned with the IUT behaviour continuously. The MFA technique was chosen with this as a requirement, and the Digital Twin is designed such that different parameters can be modified on-the-fly via realignment packets being sent from the IUT simulator.
- A GUI modelling interface should be developed to provide a rounded experience to the client. Currently, the back-end is well developed that needs a professional looking front-end.

vii. References

[1] pwc and Bertrandt, “Challenges and Trends in the Automotive Industry 2019,” *strategy&*, p. 15, 2019.

[2] M. A. Rahim, M. A. Rahman, M. M. Rahman, A. T. Asyhari, M. Z. A. Bhuiyan, and D. Ramasamy, “Evolution of IoT-enabled connectivity and applications in automotive industry: A review,” *Veh. Commun.*, vol. 27, p. 100285, 2021, doi: 10.1016/j.vehcom.2020.100285.

[3] A. Mayyas, A. Qattawi, M. Omar, and D. Shan, “Design for sustainability in automotive industry: A comprehensive review,” *Renew. Sustain. Energy Rev.*, vol. 16, no. 4, pp. 1845–1862, 2012, doi: 10.1016/j.rser.2012.01.012.

[4] “Ford Model T.” https://en.wikipedia.org/wiki/Ford_Model_T.

[5] D. J., “Electric Vehicle Waves of History: Lessons Learned about Market Deployment of Electric Vehicles,” *IntechOpen - Electr. Veh. - Benefits Barriers*, 2011, doi: 10.5772/22411.

[6] D. A. Kirsch, *The Electric Vehicle and the Burden of History*. University of Nottingham, 2000.

[7] G. Mom, *The Electric Vehicle: Technology and Expectations in the Automobile Age*, 3rd ed. The John Hopkins University Press, 2004.

[8] H. Helms, C. Kämper, and U. Lambrecht, *Carbon dioxide and consumption reduction through electric vehicles*. Elsevier Ltd., 2015.

[9] A. Perner and J. Vetter, *Lithium-ion batteries for hybrid electric vehicles and battery electric vehicles*. Elsevier Ltd., 2015.

[10] B. Scrosati and J. Garche, “Lithium batteries: Status, prospects and future,” *J. Power Sources*, vol. 195, no. 9, pp. 2419–2430, 2010, doi: 10.1016/j.jpowsour.2009.11.048.

[11] T. M. Jahns and V. Blasko, “Recent Advances in Power Electronics Technology

for Industrial and Traction Machine Drives," *Proc. IEEE*, vol. 89, no. 6, pp. 963–975, 2001, doi: 10.1109/5.931496.

[12] "Moore's Law," [Online]. Available: https://en.wikipedia.org/wiki/Moore%27s_law.

[13] J. Clark, "Suppliers to the new Opel/Vauxhall Insignia," *Automotive News Europe*, 2018. https://europe.autonews.com/article/20180412/CUTAWAY01/304069998/suppliers-to-the-new-opel-vauxhall-insignia.

[14] "Advanced Propulsion Centre," *Advanced Propulsion Centre*. https://www.apcuk.co.uk/ (accessed Jan. 27, 2022).

[15] "VCHV – Virtually Connected Hybrid Vehicle," *Institute of Digital Engineering*. https://www.ideuk.org/project-vchv (accessed Jan. 27, 2022).

[16] "APC National Network." https://www.apcuk.co.uk/networks/ (accessed Jan. 27, 2022).

[17] X. Hu, Y. Gong, D. Zhao, and W. Gu, "Structurally Optimized Neural Fuzzy Modelling for Model Predictive Control," *IEEE Trans. Ind. Informatics*, no. December, pp. 1–1, 2021, doi: 10.1109/tii.2021.3133893.

[18] X. Hu, Y. Gong, D. Zhao, and W. Gu, "Local linear model tree with optimized structure," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1163–1168, 2020, doi: 10.1016/j.ifacol.2020.12.1326.

[19] W. Gu, D. Zhao, and B. Mason, "Real-time modelling and parallel optimisation of a gasoline direct injection engine," *Proc. Am. Control Conf.*, vol. 2019-July, pp. 5544–5549, 2019, doi: 10.23919/acc.2019.8814359.

[20] W. Gu, D. Zhao, and B. Mason, "A Review of Intelligent Road Preview Methods for Energy Management of Hybrid Vehicles," *IFAC-PapersOnLine*, vol. 52, no. 5, pp. 654–660, 2019, doi: 10.1016/j.ifacol.2019.09.104.

[21] C. Fan, T. R. B. Grandjean, K. O'Regan, E. Kendrick, and W. D. Widanage,

"Understanding non-linearity in electrochemical systems using multisine-based non-linear characterization," *Trans. Inst. Meas. Control*, 2021, doi: 10.1177/01423312211045991.

[22] S. Sen, P. L. Evans, and C. M. Johnson, "Multi-frequency averaging (MFA) model of electric-hybrid powertrain suitable for variable frequency operation applied in geographically-distributed power hardware-in-the-loop (GD-PHiL) simulation," *2018 IEEE Veh. Power Propuls. Conf. VPPC 2018 - Proc.*, 2019, doi: 10.1109/VPPC.2018.8604967.

[23] S. Sen, P. L. Evans, and C. M. Johnson, "Multi-frequency averaging (MFA) model of a generic electric vehicle powertrain suitable under variable frequency of averaging developed for remote operability," *IET Electr. Syst. Transp.*, vol. 10, no. 3, pp. 268–274, 2020, doi: 10.1049/iet-est.2019.0043.

[24] N. Brayanov and A. Stoynova, "Review of hardware-in-the-loop - a hundred years progress in the pseudo-real testing," *Electrotech. Electron.*, vol. 54, no. 3–4, pp. 70–84, 2019.

[25] C. Dufour, J. Bélanger, and S. Abourida, "Using real-time simulation in hybrid electric drive and power electronics development: Process, problems and solutions," *SAE Tech. Pap.*, no. 724, 2006, doi: 10.4271/2006-01-0114.

[26] P. Sarhadi and S. Yousefpour, "State of the art: hardware in the loop modeling and simulation with its applications in design, development and implementation of system and control software," *Int. J. Dyn. Control*, vol. 3, no. 4, pp. 470–479, 2015, doi: 10.1007/s40435-014-0108-3.

[27] 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, doi: 10.1016/S0967-0661(98)00205-6.

[28] M. Bacic, "On hardware-in-the-loop simulation," in *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 3194–3198, doi: 10.1109/CDC.2005.1582653.

[29] M. Baarspul, “A review of flight simulation techniques,” *Prog. Aerosp. Sci.*, vol. 27, no. 1, pp. 1–120, 1990, doi: https://doi.org/10.1016/0376-0421(90)90006-6.

[30] N.N., “The redifon comet flight simulator,” *De haveland gazette*, pp. 139–141, 1953.

[31] O. K. Anderson, “A Design Tool (Research and Development) Simulator Survey Report,” Jan. 1962, doi: https://doi.org/10.4271/620310.

[32] J. S. C. Jr. and A. C. Jolly, “Hardware-in-the-loop simulation at the U.S. Army Missile Command,” in *Technologies for Synthetic Environments: Hardware-in-the-Loop Testing*, 1996, vol. 2741, pp. 14–19, doi: 10.1117/12.241103.

[33] M. E. Sisle and E. D. McCarthy, “Hardware-in-the-loop simulation for an active missile,” *Simulation*, vol. 39, no. 5, pp. 159–167, 1982, doi: 10.1177/003754978203900503.

[34] M. Bailey and J. Doerr, “Contributions of hardware-in-the-loop simulations to Navy test and evaluation,” in *Technologies for Synthetic Environments: Hardware-in-the-Loop Testing*, 1996, vol. 2741, pp. 33–43, doi: 10.1117/12.241122.

[35] M. B. Evans and L. J. Schilling, “The Role of Simulation in the Development and Flight Test of the HiMAT Vehicle,” 1984.

[36] N. Brewer, “Your Engineering Heritage: Antoinette and Early Flight Simulation,” *IEEE Insights USA*, 2019. https://insight.ieeeusa.org/articles/history-early-flight-simulation/.

[37] J. Drosdol and F. Panik, “The daimler-benz driving simulator a tool for vehicle development,” *SAE Tech. Pap.*, vol. 94, no. 1985, pp. 981–997, 1985, doi: 10.4271/850334.

[38] T. Suetomi, A. Horiguchi, Y. Okamoto, and S. Hata, “The driving simulator with large amplitude motion system,” *SAE Tech. Pap.*, vol. 100, no. 1991, pp. 94–101, 1991, doi: 10.4271/910113.

[39] D. J. Kempf, L. S. Bonderson, and L. I. Slafer, “Real Time Simulation for Application to ABS Development,” Feb. 1987, doi: https://doi.org/10.4271/870336.

[40] W. Huber, W.-D. Jonner, and H. Demel, “Simulation, Performance and Quality Evaluation of ABS and ASR,” Feb. 1988, doi: https://doi.org/10.4271/880323.

[41] “Dynamischer Verbrennungsmotorenprufstand mit Echtzeitsimulation des Kraftfahrzeug-Antriebsstranges,” *Automobiltechnische Zeitschrift*, 1987.

[42] W. K. DLR, “Review of Multibody Computer, Codes for Vehicle System Dynamics,” *Veh. Syst. Dyn.*, vol. 22, no. sup1, pp. 3–31, 1993, doi: 10.1080/00423119308969463.

[43] C. Savaglio, “Hardware-in-the-Loop Simulation - An Engine Controller Implementation,” Mar. 1993, doi: https://doi.org/10.4271/930204.

[44] A. Bouscayrol, “Different types of Hardware-In-the-Loop simulation for electric drives,” *IEEE Int. Symp. Ind. Electron.*, pp. 2146–2151, 2008, doi: 10.1109/ISIE.2008.4677304.

[45] H. Hanselmann, “Hardware-in-the Loop Simulation as a Standard Approach for Development, Customization, and Production Test of ECU’s,” Nov. 1993, doi: https://doi.org/10.4271/931953.

[46] P. Baracos, G. Murere, C. A. Rabbath, and W. Jin, “Enabling PC-based HIL simulation for automotive applications,” *IEMDC 2001 - IEEE Int. Electr. Mach. Drives Conf.*, pp. 721–729, 2001, doi: 10.1109/IEMDC.2001.939394.

[47] H. K. Fathy, Z. S. Filipi, J. Hagena, and J. L. Stein, “Review of hardware-in-the-loop simulation and its prospects in the automotive area,” *Model. Simul. Mil. Appl.*, vol. 6228, p. 62280E, 2006, doi: 10.1117/12.667794.

[48] R. M. Howe, “Real-time multirate synchronous simulation with single and multiple processors,” in *Enabling Technology for Simulation Science II*, 1998, vol. 3369, pp. 331–342, doi: 10.1117/12.319349.

[49] T. Kozaki, H. Moro, H. K. Fathy, and S. Gopalswamy, “Balancing the speed and fidelity of automotive powertrain models through surrogation,” *Am. Soc. Mech. Eng. Dyn. Syst. Control Div. DSC*, vol. 73, no. 1 PART A, pp. 249–258, 2004, doi: 10.1115/IMECE2004-61293.

[50] C. Frangos, “Control system analysis of a hardware-in-the-loop simulation,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 26, no. 4, pp. 666–669, Jul. 1990, doi: 10.1109/7.55560.

[51] R. Zhang and A. G. Alleyne, “Dynamic Emulation Using an Indirect Control Input,” *J. Dyn. Syst. Meas. Control*, vol. 127, no. 1, pp. 114–124, Jun. 2004, doi: 10.1115/1.1876496.

[52] S. Raman, N. Sivashankar, W. Milam, W. Stuart, and S. Nabi, “Design and Implementation of HIL Simulators for Powertrain Control System Software Development,” in *Proceedings Of The American Control Conference*, 1999, no. 403, pp. 642–646.

[53] C. Dufour, S. Abourida, and J. Bélanger, “Real-Time Simulation of Electrical Vehicle Motor Drives on a PC Cluster,” *Proc. 10th Eur. Conf. Power Electron. Appl. (EPE 2003), Toulouse*, no. September 2017, pp. 2–4, 2003.

[54] C. Dufour, T. Ishikawa, A. Simon, and J. Belanger, “Modern hardware-in-the-loop simulation technology for fuel cell hybrid electric vehicles,” *VPPC 2007 - Proc. 2007 IEEE Veh. Power Propuls. Conf.*, pp. 432–439, 2007, doi: 10.1109/VPPC.2007.4544165.

[55] M. Hirakami and W. Neugebauer, “Transient Network Analyzer Operation with Digital Computer Control and Analysis,” *IEEE Trans. Power Appar. Syst.*, vol. PAS-100, no. 4, pp. 1597–1607, Apr. 1981, doi: 10.1109/TPAS.1981.316548.

[56] W. Ren, “Accuracy Evaluation of Power Hardware-in-the-Loop (PHIL) Simulation,” 2007.

[57] “About RTDS Technologies,” *RTDS Technologies*. https://www.rtds.com/about-rtds-technologies/.

[58] W. Ren, M. Steurer, and T. L. Baldwin, “Improve the stability and the accuracy of power hardware-in-the-loop simulation by selecting appropriate interface algorithms,” *IEEE Trans. Ind. Appl.*, vol. 44, no. 4, pp. 1286–1294, 2008, doi: 10.1109/TIA.2008.926240.

[59] W. Ren *et al.*, “Interfacing issues in real-time digital simulators,” *IEEE Trans. Power Deliv.*, vol. 26, no. 2, pp. 1221–1230, 2011, doi: 10.1109/TPWRD.2010.2072792.

[60] OPAL-RT, “Key considerations when selection amplifiers for your Power Hardware in the loop (PHIL) Testbed,” *OPAL-RT Technologies*, p. 18, 2019.

[61] “RTDS Power Hardware-in-the-Loop Report 2.1,” *RTDS Technologies*. https://www.rtds.com/applications/power-hardware-in-the-loop/.

[62] M. O. Faruque *et al.*, “Real-Time Simulation Technologies for Power Systems Design, Testing, and Analysis,” *IEEE Power Energy Technol. Syst. J.*, vol. 2, no. 2, pp. 63–73, 2015, doi: 10.1109/JPETS.2015.2427370.

[63] M. Lemaire, P. Sicard, and J. Belanger, “Prototyping and Testing Power Electronics Systems Using Controller Hardware-In-the-Loop (HIL) and Power Hardware-In-the-Loop (PHIL) Simulations,” *2015 IEEE Veh. Power Propuls. Conf. VPPC 2015 - Proc.*, pp. 1–6, 2015, doi: 10.1109/VPPC.2015.7353000.

[64] C. Dufour, J.-N. Paquin, V. Lapointe, J. Bélanger, and L. Schoen, “PC-Cluster-Based Real-Time Simulation of an 8-synchronous machine network with HVDC link using RT-LAB and TestDrive,” *7th Int. Conf. Power Syst. Transients (IPST ’07), Lyon, Fr.*, 2007, [Online]. Available: http://www.ipst.org/techpapers/2007/ipst_2007/papers_IPST2007/Session24/65.pdf%5Cnhttp://ipstconf.org/papers/Proc_IPST2007/07IPST029.pdf.

[65] C. Dufour, S. Abourida, J. Belanger, and V. Lapointe, “Real-Time Simulation of Permanent Magnet Motor Drive on FPGA Chip for High-Bandwidth Controller Tests and Validation,” in *IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*, Nov. 2006, pp. 4581–4586, doi:

10.1109/IECON.2006.347676.

[66] C. Dufour, S. Cense, V. Jalili-Marandi, and J. Belanger, “Review of state-of-the-art solver solutions for HIL simulation of power systems, power electronic and motor drives,” *2013 15th Eur. Conf. Power Electron. Appl. EPE 2013*, pp. 1–12, 2013, doi: 10.1109/EPE.2013.6632001.

[67] “SCALEXIO,” *dSPACE GmbH*. https://www.dspace.com/shared/data/pdf/2020/dSPACE_SCALEXIO_Product-information_02_2020_English.pdf.

[68] “High-Voltage Electronic Load - For power hardware-in-the-loop simulation up to 800 V,” *dSPACE GmbH*. https://www.dspace.com/shared/data/pdf/2019/dSPACE_High-Voltage_Electronic_Load_Product_information_01_2019_English.pdf.

[69] “HIL Tests for Battery Management Systems - Emulating battery cell voltages,” *dSPACE GmbH*. https://www.dspace.com/en/inc/home/applicationfields/stories/electricaldrive/hil_bms.cfm.

[70] “HIL Tests for DC/DC Converters - Testing power electronics control units,” *dSPACE GmbH*. https://www.dspace.com/en/inc/home/applicationfields/stories/electricaldrive/hil_dcdc-controller.cfm#145_32990.

[71] A. Wagener, T. Schulte, P. Waeltermann, and H. Schuette, “Hardware-in-the-loop test systems for electric motors in advanced powertrain applications,” *SAE Tech. Pap.*, vol. 2007, no. 724, pp. 776–790, 2007, doi: 10.4271/2007-01-0498.

[72] X. Wu, S. Lentijo, A. Deshmuk, A. Monti, and F. Ponci, “Design and implementation of a power-hardware-in-the-loop interface: a nonlinear load case study,” in *Twentieth Annual IEEE Applied Power Electronics Conference and Exposition, 2005. APEC 2005.*, Mar. 2005, vol. 2, pp. 1332-1338 Vol. 2, doi: 10.1109/APEC.2005.1453182.

[73] R. Kuffel *et al.*, “Expanding an Analogue HVDC Simulator’s Modelling Capability Using a Real-Time Digital Simulator (RTDS),” in *ICDS ’95. First International Conference on Digital Power System Simulators*, Apr. 1995, pp. 199-, doi: 10.1109/ICDS.1995.492830.

[74] D. J. Atkinson and H. J. Slater, “Real-time emulation for power equipment development. Part 1: Real-time simulation,” *IEEE Proc.*, vol. 14, no. 2, pp. 1–6, 1998.

[75] H. J. Slater, D. J. Atkinson, and A. G. Jack, “Real-time emulation for power equipment development. Part 2: The virtual machine,” *IEE Proc. Electr. Power Appl.*, vol. 145, no. 3, pp. 153–158, 1998, doi: 10.1049/ip-epa:19981849.

[76] R. M. Fujimoto, *Parallel and Distributed Simulation Systems*. John Wiley & Sons Inc, 2000.

[77] M. Brudnak *et al.*, “Virtual Combat Vehicle Experimentation for Duty Cycle Measurement,” *SAE Int. J. Commer. Veh.*, vol. 1, no. 1, pp. 54–70, 2008.

[78] V. Paul, “Human-in-the-loop Simulation-based Combat Vehicle Duty Cycle Mearurement : Duty Cycle Experiment 1 (06S-SIW-080),” *US Army TARDEC-GVSL*. pp. 1–26, 2006.

[79] M. J. Brudnak *et al.*, “Soldier/hardware-in-the-loop simulation-based combat vehicle duty cycle measurement: Duty cycle experiment,” *Simul. Interoperability Stand. Organ. - Spring Simul. Interoperability Work. 2007, 07 Spring SIW*, pp. 107–122, 2006.

[80] M. Compere, J. Goodell, M. Simon, W. Smith, and M. Brudnak, “Robust Control Techniques Enabling Duty Cycle Experiments Utilizing a 6-DOF Crewstation Motion Base , a Full Scale Combat Hybrid Electric Power System , and Long Distance Internet Communications Robust Control Techniques Enabling Duty Cycle Experiments Ut,” *Embry-Riddle Aeronaut. Univ. - Sch. Commons*, 2006.

[81] T. Ersal, M. J. Brudnak, A. Salvi, Y. Kim, J. B. Siegel, and J. L. Stein, “An iterative learning control approach to improving fidelity in internet-distributed hardwarein-

the-loop simulation," *J. Dyn. Syst. Meas. Control. Trans. ASME*, vol. 136, no. 6, pp. 1–8, 2014, doi: 10.1115/1.4027868.

[82] T. Ersal, M. J. Brudnak, J. L. Stein, and H. K. Fathy, "Statistical Transparency Analysis in Internet-Distributed Hardware-in-the-Loop Simulation," *IEEE/ASME Trans. Mechatronics*, vol. 17, no. 2, pp. 228–238, 2012.

[83] T. Ersal, R. B. Gillespie, M. J. Brudnak, J. L. Stein, and H. K. Fathy, "Effect of coupling point selection on distortion in internet-distributed hardware-in-The-loop simulation," *Int. J. Veh. Des.*, vol. 61, no. 1–4, pp. 67–85, 2013, doi: 10.1504/IJVD.2013.050840.

[84] A. Tandon, M. J. Brudnak, J. L. Stein, and T. Ersal, "An Observer Based Framework to Improve Fidelity in Internet-Distributed Hardware-in-the-Loop Simulations," *Proc. ASME 2013 Dyn. Syst. Control Conf.*, pp. 1–9, 2013.

[85] X. Ge, M. J. Brudnak, J. L. Stein, and T. Ersal, "A Norm Optimal Iterative Learning Control framework towards Internet-Distributed Hardware-In-The-Loop simulation," *Proc. Am. Control Conf.*, no. IIc, pp. 3802–3807, 2014, doi: 10.1109/ACC.2014.6859243.

[86] B. Lu, A. Monti, and R. A. Dougal, "Real-Time Hardware-In-the-Loop Testing During Design of Power Electronics Controls," in *IECON'03. 29th Annual Conference of the IEEE Industrial Electronics Society*, 2003, pp. 1840–1845, doi: 10.1109/IECON.2003.1280340.

[87] T. Lovett, A. Monti, E. Santi, and R. A. Dougal, "A Multilanguage Environment For Interactive Simulation And Development Of Controls For Power Electronics," in *IEEE 32nd Annual Power Electronics Specialists Conference*, 2001, pp. 1725–1729, doi: 10.1109/PESC.2001.954368.

[88] B. Lu, X. Wu, and A. Monti, "Implementation of a Low-Cost Real-Time Virtue Test Bed for Hardware-In-The-Loop Testing," in *31st Annual Conference of IEEE Industrial Electronics Society*, 2005, pp. 239–244, doi: 10.1109/IECON.2005.1568910.

[89] A. Monti, X. Wu, and S. Lentijo, “A novel interface for power-hardware-in-the-loop simulation,” in *IEEE Workshop on Computers in Power Electronics*, 2004, pp. 178–182, doi: 10.1109/CIPE.2004.1428147.

[90] H. P. Figueroa, J. L. Bastos, A. Monti, and R. Dougal, “A modular real-time simulation platform based on the virtual test bed,” *IEEE Int. Symp. Ind. Electron.*, vol. 2, pp. 1537–1541, 2006, doi: 10.1109/ISIE.2006.295700.

[91] Y. Work, Y. Shi, A. Monti, A. Carta, and C. Muscas, “GPS synchronized measurements for real-time hardware-in-the-loop experiments,” *Conf. Rec. - IEEE Instrum. Meas. Technol. Conf.*, pp. 854–858, 2008, doi: 10.1109/IMTC.2008.4547156.

[92] M. Stevic, A. Monti, and A. Benigni, “Development of a simulator-to-simulator interface for geographically distributed simulation of power systems in real time,” *IECON 2015 - 41st Annu. Conf. IEEE Ind. Electron. Soc.*, pp. 5020–5025, 2015, doi: 10.1109/IECON.2015.7392888.

[93] A. Monti *et al.*, “A Global Real-Time Superlab: Enabling High Penetration of Power Electronics in the Electric Grid,” *IEEE Power Electron. Mag.*, vol. 5, no. 3, pp. 35–44, 2018, doi: 10.1109/MPEL.2018.2850698.

[94] M. Stevic, S. Vogel, A. Monti, and S. D’Arco, “Feasibility of geographically distributed real-time simulation of HVDC system interconnected with AC networks,” *2015 IEEE Eindhoven PowerTech, PowerTech 2015*, 2015, doi: 10.1109/PTC.2015.7232700.

[95] M. Stevic *et al.*, “Virtual integration of laboratories over long distance for real-time co-simulation of power systems,” *IECON Proc. (Industrial Electron. Conf.*, pp. 6717–6721, 2016, doi: 10.1109/IECON.2016.7793422.

[96] M. Stevic *et al.*, “Multi-site European framework for real-time co-simulation of power systems,” *IET Gener. Transm. Distrib.*, vol. 11, no. 17, pp. 4126–4135, 2017, doi: 10.1049/iet-gtd.2016.1576.

[97] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, “Characterising the Digital

Twin: A systematic literature review," *CIRP J. Manuf. Sci. Technol.*, vol. 29, pp. 36–52, 2020, doi: 10.1016/j.cirpj.2020.02.002.

[98] M. R. Enders and N. Hoßbach, "Dimensions of digital twin applications - A literature review," *25th Am. Conf. Inf. Syst. AMCIS 2019*, no. August 2019, 2019.

[99] M. Grieves, "Digital twin: manufacturing excellence through virtual factory replication," *White Pap.*, vol. 1, pp. 1–7, 2014.

[100] G. Bhatti, H. Mohan, and R. Raja Singh, "Towards the future of smart electric vehicles: Digital twin technology," *Renew. Sustain. Energy Rev.*, vol. 141, no. August 2020, p. 110801, 2021, doi: 10.1016/j.rser.2021.110801.

[101] A. Rassolkin, T. Vaimann, A. Kallaste, and V. Kuts, "Digital twin for propulsion drive of autonomous electric vehicle," *2019 IEEE 60th Annu. Int. Sci. Conf. Power Electr. Eng. Riga Tech. Univ. RTUCON 2019 - Proc.*, 2019, doi: 10.1109/RTUCON48111.2019.8982326.

[102] N. G. Panwar, S. Singh, A. Garg, A. K. Gupta, and L. Gao, "Recent Advancements in Battery Management System for Li-Ion Batteries of Electric Vehicles: Future Role of Digital Twin, Cyber-Physical Systems, Battery Swapping Technology, and Nondestructive Testing," *Energy Technol.*, vol. 9, no. 8, pp. 1–17, 2021, doi: 10.1002/ente.202000984.

[103] Vandana, A. Garg, and B. K. Panigrahi, " Multi-dimensional digital twin of energy storage system for electric vehicles: A brief review ," *Energy Storage*, vol. 3, no. 6, pp. 1–14, 2021, doi: 10.1002/est2.242.

[104] S. Venkatesan, K. Manickavasagam, N. Tengenkai, and N. Vijayalakshmi, "Health monitoring and prognosis of electric vehicle motor using intelligent-digital twin," *IET Electr. Power Appl.*, vol. 13, no. 9, pp. 1328–1335, 2019, doi: 10.1049/iet-epa.2018.5732.

[105] A. Rassolkin, V. Rjabtsikov, T. Vaimann, A. Kallaste, V. Kuts, and A. Partyshev, "Digital Twin of an Electrical Motor Based on Empirical Performance Model," *2020 11th Int. Conf. Electr. Power Drive Syst. ICEPDS 2020 - Proc.*, pp. 2020–

2023, 2020, doi: 10.1109/ICEPDS47235.2020.9249366.

[106] A. Wunderlich and E. Santi, “Digital twin models of power electronic converters using dynamic neural networks,” *Conf. Proc. - IEEE Appl. Power Electron. Conf. Expo. - APEC*, pp. 2369–2376, 2021, doi: 10.1109/APEC42165.2021.9487201.

[107] M. Milton, C. O. De La, H. L. Ginn, and A. Benigni, “Controller-Embeddable Probabilistic Real-Time Digital Twins for Power Electronic Converter Diagnostics,” *IEEE Trans. Power Electron.*, vol. 35, no. 9, pp. 9852–9866, 2020, doi: 10.1109/TPEL.2020.2971775.

[108] F. Li *et al.*, “Review of Real-time Simulation of Power Electronics,” *J. Mod. Power Syst. Clean Energy*, vol. 8, no. 4, pp. 796–808, 2020, doi: 10.35833/MPCE.2018.000560.

[109] S. Mojlish, N. Erdogan, D. Levine, and A. Davoudi, “Review of Hardware Platforms for Real-Time Simulation of Electric Machines,” *IEEE Trans. Transp. Electrif.*, vol. 3, no. 1, pp. 130–146, 2017, doi: 10.1109/TTE.2017.2656141.

[110] G. Parma and V. Dinavahi, “Real-time digital hardware simulation of power electronics and drives,” *2007 IEEE Power Eng. Soc. Gen. Meet. PES*, vol. 22, no. 2, pp. 1235–1246, 2007, doi: 10.1109/PES.2007.385592.

[111] H. Jin, “Behavior-mode simulation of power electronic circuits,” *IEEE Trans. Power Electron.*, vol. 12, no. 3, pp. 443–452, 1997, doi: 10.1109/63.575672.

[112] L. Spracklen and S. G. Abraham, “Chip multithreading: Opportunities and challenges,” *Proc. - Int. Symp. High-Performance Comput. Archit.*, pp. 248–252, 2005, doi: 10.1109/HPCA.2005.10.

[113] M. O. Faruque and V. Dinavahi, “An advanced PC-Cluster based real-time simulator for power electronics and drives,” *IEEE Int. Symp. Ind. Electron.*, vol. 3, pp. 2579–2584, 2006, doi: 10.1109/ISIE.2006.295978.

[114] A. Sangiovanni-Vincentelli, “Architecture of Field-Programmable Gate Arrays,” *Proc. IEEE*, vol. 81, no. 7, pp. 1013–1029, 1993, doi: 10.1109/5.231340.

[115] E. Monmasson and M. N. Cirstea, “FPGA design methodology for industrial control systems - A review,” *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1824–1842, 2007, doi: 10.1109/TIE.2007.898281.

[116] Y. Chen and V. Dinavahi, “Hardware emulation building blocks for real-time simulation of large-scale power grids,” *IEEE Trans. Ind. Informatics*, vol. 10, no. 1, pp. 373–381, 2014, doi: 10.1109/TII.2013.2243742.

[117] A. Frances, R. Asensi, O. Garcia, R. Prieto, and J. Uceda, “Modeling electronic power converters in smart dc microgrids - An overview,” *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 6274–6287, 2018, doi: 10.1109/TSG.2017.2707345.

[118] W. Gao, S. Member, E. V Solodovnik, R. A. Dougal, and S. Member, “Symbolically Aided Model Development for an Induction Machine in Virtual Test Bed,” *EEE Trans. Energy Convers.*, vol. 19, no. 1, pp. 125–135, 2004, doi: 10.1109/TEC.2003.822316.

[119] S. Y. R. R. Hui, S. Member, and K. K. Fung, “Fast Decoupled Simulation of Large Power Electronic Systems Using New Two-Port Companion Link Models,” *IEEE Trans. Power Electron.*, vol. 12, no. 3, pp. 462–473, 1997, doi: 10.1109/63.575674.

[120] S. R. Sanders, J. M. Noworolski, X. Z. Liu, and G. C. Verghese, “Generalized Averaging Method for Power Conversion Circuits,” *IEEE Trans. Power Electron.*, vol. 6, no. 2, pp. 251–259, 1991, doi: 10.1109/63.76811.

[121] A. M. Stanković, S. R. Sanders, and T. Aydin, “Dynamic phasors in modeling and analysis of unbalanced polyphase AC machines,” *IEEE Trans. Energy Convers.*, vol. 17, no. 1, pp. 107–113, 2002, doi: 10.1109/60.986446.

[122] A. M. Stanković and T. Aydin, “Analysis of asymmetrical faults in power systems using dynamic phasors,” *Proc. IEEE Power Eng. Soc. Transm. Distrib. Conf.*, vol. 3, no. 3, p. 1957, 2000, doi: 10.1109/pess.2000.868834.

[123] T. Demiray, F. Milano, and G. Andersson, “Dynamic phasor modeling of the doubly-fed induction generator under unbalanced conditions,” *2007 IEEE*

Lausanne POWERTECH, Proc., no. 2, pp. 1049–1054, 2007, doi: 10.1109/PCT.2007.4538460.

[124] V. A. Caliskan, G. C. Verghese, and A. M. Stanković, “Multifrequency averaging of DC/DC converters,” *IEEE Trans. Power Electron.*, vol. 14, no. 1, pp. 124–133, 1999, doi: 10.1109/63.737600.

[125] P. Mattavelli, “Phasor dynamics of thyristor-controlled series capacitor systems,” *IEEE Trans. Power Syst.*, vol. 12, no. 3, pp. 1259–1267, 1997, doi: 10.1109/59.630469.

[126] T. Demiray, “Simulation of Power System Dynamics using Dynamic Phasor Models,” Swiss Federal Institute of Technology Zurich, 2008.

[127] M. C. Chudasama and A. M. Kulkarni, “Dynamic phasor analysis of SSR mitigation schemes based on passive phase imbalance,” *IEEE Trans. Power Syst.*, vol. 26, no. 3, pp. 1668–1676, 2011, doi: 10.1109/TPWRS.2010.2072793.

[128] H. Zhu, Z. Cai, H. Liu, Q. Qi, and Y. Ni, “Hybrid-model transient stability simulation using dynamic phasors based HVDC system model,” *Electr. Power Syst. Res.*, vol. 76, no. 6–7, pp. 582–591, 2006, doi: 10.1016/j.epsr.2005.09.017.

[129] T. Yang, S. Bozhko, J. M. Le-Peuvedic, G. M. Asher, and C. I. Hill, “Dynamic phasor modeling of multi-generator variable frequency electrical power systems,” *IEEE Trans. Power Syst.*, vol. 31, no. 1, pp. 563–571, 2016, doi: 10.1109/TPWRS.2015.2399371.

[130] D. G. Holmes and T. A. Lipo, “03 - Modulation of One Inverter Phase Leg,” in *Pulse Width Modulation for Power Converters*, 2010, pp. 95–153.

[131] D. G. Holmes and T. A. Lipo, “05 - Modulation of Three-Phase Voltage Source Inverters,” in *Pulse Width Modulation for Power Converters*, 2010, pp. 215–258.

[132] A. Bowes, SR and Midoun, “Suboptimal switching strategies for microprocessor-controlled PWM inverter drives,” *IEE Proc. B (Electric Power Appl.*, vol. 132, no.

3, pp. 133–148, 1985.

[133] D. G. Holmes and T. A. Lipo, “01 - Introduction to Power Electronic Converters,” in *Pulse Width Modulation for Power Converters*, 2010, pp. 1–56.

[134] D. G. Holmes and T. A. Lipo, “15 - Appendix 1: Fourier Series Representation of a Double Variable Controlled Waveform,” in *Pulse Width Modulation for Power Converters*, vol. 00, 2010, pp. 623–628.

[135] R. Gamoudi, D. E. Chariag, and L. Sbita, “A Review of Spread-Spectrum-Based PWM Techniques - A Novel Fast Digital Implementation,” *IEEE Trans. Power Electron.*, vol. 33, no. 12, pp. 10292–10307, 2018, doi: 10.1109/TPEL.2018.2808038.

[136] H. Chen and H. Zhao, “Review on pulse-width modulation strategies for common-mode voltage reduction in three-phase voltage-source inverters,” *IET Power Electron.*, vol. 9, no. 14, pp. 2611–2620, 2016, doi: 10.1049/iet-pel.2015.1019.

[137] A. M. Trzynadlowski *et al.*, “Random Pulse Width Modulation Techniques for Converter-Fed Drive Systems - A Review,” *IEEE Trans. Ind. Appl.*, vol. 30, no. 5, pp. 1166–1175, 1994.

[138] N. Rai and S. Chakravorty, “Generalized Formulations and Solving Techniques for Selective Harmonic Elimination PWM Strategy: A Review,” *J. Inst. Eng. Ser. B*, vol. 100, no. 6, pp. 649–664, 2019, doi: 10.1007/s40031-019-00411-1.

[139] D. G. Holmes and T. A. Lipo, “06 - Zero Space Vector Space.pdf,” in *Pulse Width Modulation for Power Converters*, 2010, pp. 259–336.

[140] ETPS, “Datasheet for ETPS EAC-ACS-4Q grid-tied 4 quadrant power supplies.” https://etps.co.uk/site/assets/files/1774/eac-acs-4q.pdf (accessed Jan. 27, 2022).