Fault Tolerant Distributed Drive for Multiphase Machines

Filippo Savi

January 17, 2022

To the memory of my grandfather Luigi,

without whom this journey would have never even started.

Acknowledgements

I would like to thank all the people that helped me during the course of my doctoral studies.

First and foremost I am deeply grateful to my supervisors, Prof. Chris Gerada, Prof. Pat Wheeler and Prof. Giampaolo Buticchi, for their invaluable advice and technical support. Without their guidance the achieved results would not have been possible.

I would like to offer my special thanks to Prof. Michele Degano and Dr. Alessandro Galassini, who have been instrumental in ensuring the success of my stay in Nottingham

My sincerest thanks go also to Prof. Davide Barater and Prof. Giovanni Franceschini from Università degli studi di Modena and Reggio Emilia, for their support, guidance, and for the free access they gave me to their labs.

I would also like to extend my sincere thanks to all my colleagues PhD students at the University of Nottingham, Ningbo, for the wonderful time spent together.

Last but not least I would like to thank my family, and friend who have always been there for me in the time of need and in my darkest hour.

List of symbols

ω_e	Electrical angular speed
ω_m	mechanical angular speed
V_{ph}	Phase Voltage
I_{ph}	Phase current
α_e	electrical angular acceleration
α_m	mechanical angular acceleration
R_{ph}	Per-phase winding resistance
L_{ph}	Per-phase winding inductance
T_E	Electrical Torque
T_A	Acceleration Torque
T_L	Load Torque
J	Rotor moment of inertia
K_e	Back-emf constant
n_p	Number of poles
η	Inverter efficiency
K_p	Proportional gain
K_i	Integral gain

K_r	Resonant gain
ζ	Resonant damping
L_n	Self inductance for winding number n
$M_{n,m}$	Mutual inductance between winding number n and m
$I_d(6)$	Six-by-six identity matrix
$J_{6,6}$	Six-by-six ones matrix
$0_{6,6}$	Six-by-six zeros matrix

List of Abbreviations

ADC	Analogue to Digital Converter
AMBA	Advanced Microcontroller Bus Architecture
APB	Advanced Peripheral Bus
AST	Abstract Syntax Tree
AXI	Advanced eXtensible Interface
BER	Bit Error Rate
BOM	Bill Of Material
BSC	Binary Symmetrical Channel
DMA	Direct Memory Access
DSP	Digital Signal Processing
DTC	Direct Torque Control
EBNF	Extended Backus-Naur form
EHA	Electro-Hydraulic Actuator
EMA	Electro-Mechanical Actuator
FOC	Field Oriented Control
FEC	Forward Error Correction
FIFO	First In, First Out
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
HDL	Hardware Description Language

HMI	Human Machine Interface
HTML	HyperText Markup Language
HVDC	High Voltage DC
ISA	Instruction Set Architecture
JSON	JavaScript Object Notation
JWT	JSON Web Token
LQR	Linear Quadratic Regulator
MCU	MicroController unit
MEA	More Electric Aircraft
MMC	Modular Multilevel Converter
MPU	MicroProcessor Unit
MTBF	Mean Time Between Failures
ODE	Ordinary Differential Equation
PID	Proportional Integral Derivative
PIR	Proportional Integral Resonant
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
RAM	Random Access Memory
REST	Representational State Transfer
ROM	Read Only Memory
RS	Reed-Solomon
SIMD	Single Instruction Multiple Data
SNR	Signal to Noise Ratio
SoC	System on Chip
SOGI	Second Order Generalized Integrator

SPI	Serial Peripheral Interface
SSV	Structured Singular Value
TLS	Transport Level Security
THD	Total Harmonic Distortion
URL	Uniform Resource Locator
VLIW	Very Long Instruction Word
VSD	Vector Space Decomposition
VSI	Voltage Source Inverter

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

It is widely accepted by both the global scientific community [1] and the biggest political bodies, such as the United Nations [2], that climate changes due to the impact of human activities in post-industrial societies constitutes the biggest problem facing mankind, with potentially dire and wide-ranging consequences on a global scale. The root cause of these issues can be traced to the uncontrolled emission of large amounts of greenhouse gasses, like carbon dioxide (CO_2), in the atmosphere. The primary process responsible for this can be easily identified in the combustion of fossil fuels like coal or oil derived products, used for heat and power generation. Progress is being made in energy generation, with up to 25% of electricity coming from renewable energy sources, as shown in Fig. 1.1.

Electrification is increasingly seen as a possible solution toward reduction of greenhouse gas emissions from human activities that traditionally rely on fossil fuels as their primary energy source [4]. However, there is still a varied group of sectors, denominated "hard-to-abate", for which no currently viable, carbon-neutral alternative is available. For some of them the fossil fuel is

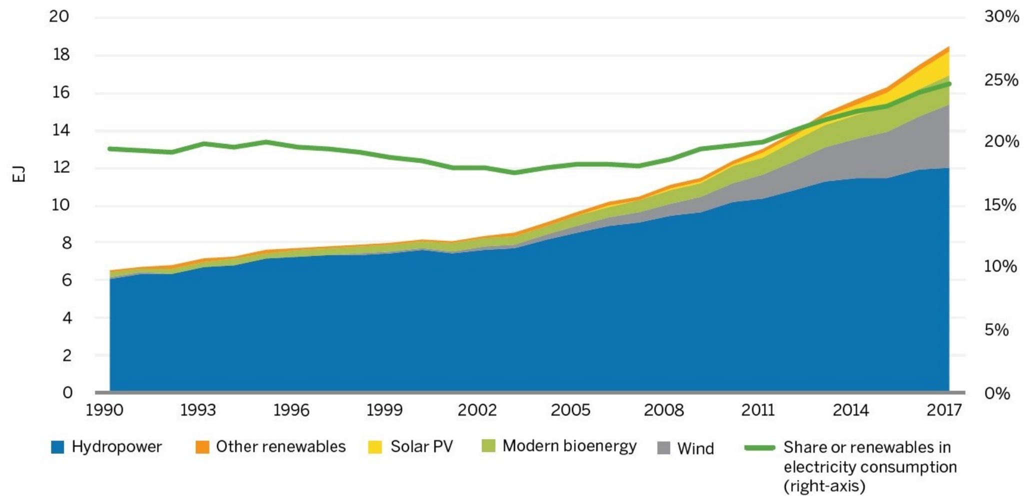


Figure 1.1: Renewable energy share in the overall energy mix [3]

not only a source of energy, but also performs a crucial role in the whole process, that can not be performed using only electricity [5]. This can be exemplified by the steelmaking process [6], where coal coke is crucial in the chemical reduction of the iron oxide containing mineral ore to pig iron. In other sectors direct connection to the power grid completely impossible, like for transportation [7]. The aviation sector in particular, whose emission have steadily increased due to its fast and continued growth, can be included in this last category. To tackle the decarbonization challenge in this field, roadmaps have been produced from industry-wide coalitions including airlines and aircraft manufacturers, or regulating bodies, in both the United Kingdom [8], European Union [8] and China [9].

Unfortunately, the low volumetric and gravimetric energy density of all currently available forms of energy storage make direct electrification impractical, if not outright impossible in the near future. It is for these reasons that, as a short term measure, the emissions form the aviation sector need to be minimized, by increasing the fuel efficiency of modern airliners as much as possible. The first avenue that is being thoroughly explored [10–12] is the transition to a More Electric Aircraft (MEA), where pneumatic and hydraulic subsystems in the aircraft are replaced with electric power distribution and actuation. These ideas have also started to find traction in the commercial aviation industry, with the Boeing 787 going for a no-bleed architecture, reducing fuel consumption and emissions by increasing the thermodynamic efficiency of the turbofan engines [13]. When looking at less conventional architectures, even larger efficiency gains are possible with the introduction of a hybrid electric approach [14] to the main propulsion engines. The use of a turbogenerator inside the plane fuselage and several electric motors in a concept defined Distributed Propulsion [15], this gives much more freedom to the aircraft designer in the optimal and independent positioning of both energy generation and propulsion subsystem in the aircraft enabling among other things more efficient aerodynamics, leading to a very large fuel burn reduction. A common trait between all the proposals is a considerable shift toward electrification of a number of mission critical systems, including main propulsion. Among the biggest issues blocking a wider adoption of electric infrastructure in aircraft are power density and fault tolerance, as discussed in [16, 17]. On the front of power density, wide-bandgap devices [18] with their high operating frequencies allow designers to reduce both size and weight of power converters, by reducing the size of passive filtering components. While on the reliability front, the adoption of fault-tolerant architectures is one of the more promising avenues of development, to bring electric systems' reliability up to levels acceptable in the aviation field. Distributed architectures in particular, as studied in this thesis, can not only be developed to be tolerant to a desired number of faults, but can also offer other benefits, like scalability and modularity, both desired properties with increasing power requirements.

1.2 Guiding principles and Contribution

The aim of this work is to explore the needs, in terms of power electronics and control of high power fault-tolerant machine drive when used in mission critical aerospace applications. Much work has already been done to develop high performance electrical machines and converter topologies, making large gains in these fields less likely. The architectural and control levels on the other hand have been much less explored, making an in depth, system level study of these topics worthwhile. The main focus throughout the whole design process has been total scalability of the designed architecture, achieved through vertically integrated co-design of power electronics hardware, firmware and software. In this thesis a distributed fault-tolerant drive architecture is presented, potentially able to scale effortlessly up to hundreds of phases, without significant redesign.

Few key technologies enabling high frequency distributed current control in a scalable system have been developed. A custom communication protocol has been introduced, designed from the ground up for distributed power electronics control applications, enabling sub microsecond communication latency on bandwidth limited channels, like common and cheap plastic optical fibres. Being tailored to the specific application it also avoids the common pitfalls related to the industrial automation origins of the most common protocols.

A second cornerstone of the architecture, introduced in this thesis, are a custom Instruction Set Architecture (ISA) and the related processing core, tailored to perform control system calculations with total determinism, as opposed to the commonly used Discrete DSP processors. Their implementation inside an FPGA combines the advantages of both systems, with scalability and determinism proper of custom logic, while retaining the ease of development typical of software systems.

Finally, a highly scalable synchronous reference frame current control strategy is adopted, in order to take full advantage of the designed distributed drive architecture. The avoidance of commonly used reference frame transformations completely decouples the implementation of current controllers between phases, enabling parallelization of the entire control system, that channels, in case of necessity, be partitioned across multiple computing nodes, without the strict need for high speed communication links between them.

1.3 Content of the Thesis

Chapter 1 This introductory chapter contextualizes the work presented in this thesis within the wider research sector, showing its importance for electrification and decarbonization of the aerospace sector. Here, the novelty of the research that has been carried out is also detailed and the structure of the thesis presented.

Chapter 2 This chapter gives an overview of the state-of-the-art as pertaining to the subject of this thesis. First the various possible tolerant actuation hardware configuration are revised, both from a high level system view and from drive component perspective. The second area subjected to the review is the possible multiphase current control methodologies, able to support fault-tolerant operation. Finally, an overview of the development in power distributed system architectures in power electronics is given.

Chapter 3 The chapter describes in detail the hardware design of the power cells used in the system architecture proposed by this thesis. An initial set of starting requirements is used to guide an initial multi-objective design optimization, that helped fix few initial key requirements like switching frequency, switching device model, number of devices in parallel, and others. Then the specific schematic design is presented, detailing all the major design choices.

Chapter 4 Here the FPGA based control platform architecture is described, consistent of three layers. First is the Human Machine Interface (HMI), that allows the users and researchers to easily operate the drive, modifying parameters or operating modes, while at the same time giving feedback on the whole system status through a real time plot of a number of selected data-points. The second layer, the management layer, is a software component that acts as the back-end operation of the HMI, saving user specific data like parameters, scripts, programs and so on, while also managing the low level interface with the control logic. Last is the Control layer, implemented directly in FPGA logic, that is responsible for all real-time control tasks.

Chapter 5 In this chapter the femtoCore embedded DSP is presented, a processor ISA and implementation specifically designed for feedback control system implementation. Thanks to the custom design and tight integration with the surrounding logic it allows to achieve the same guarantees in terms of determinism and jitter free operations typically achieved by FPGA systems while also enjoying the ease of programming commonly associated with pure software development, obtaining the best of both worlds. The interleaved Single Instruction Multiple Data (SIMD) execution model further simplifies the implementation of multiphase control systems.

Chapter 6 This chapter presents the Independent current control methodology, designed for seamless and scalable fault-tolerant operation. Starting from the development of a single phase machine model with two degrees of freedom (input voltage and operating frequency). Then the static reference frame current controller is presented, analysed and tuned with an H_∞ robust tuning approach. Finally, a state space stability analysis was performed to ensure stability of the control system in both the pre- and post-fault scenario.

Chapter 7 The chapter presents detailed system level simulations of the designed architecture. First the electric domain alone is simulated, verifying performance and stability, when taking into account higher order effects. Then a thermal model of the converter is introduced to verify its behaviour, especially with respect to possible thermal runaway scenarios. Finally, a sensitivity analysis is performed to evaluate the most influential parameters with respect to selected performance metrics.

Chapter 8 Here the proposed fault-tolerant drive architecture is experimentally validated and qualified. Single power cell testing is performed to evaluate the hardware design and ensure trouble free operation. Then the entire system is tested with a six-phase asymmetrical permanent magnet synchronous machine. Steady state testing demonstrates the output current quality. Load steps are performed to evaluate the control dynamics and finally the system behaviour in response to a single open-phase fault is evaluated, showing good pre- and post-reconfiguration behaviour.

Chapter 9 This last chapter draws the conclusions of the performed work, while also presenting few interesting future avenues of development.

Chapter 2

State of the Art

2.1 Reliability and Fault tolerance in electric drives

When approaching the design of a high value system, clear reliability goals should be specified during the initial phases, along with all other requirements, Mean Time between failures (MTBF) and failure rates being among the most commonly used. Only at this point the true design process can begin. Two philosophies can be followed to meet these targets, increasing the inherent reliability of the components to completely prevent failures, and fault tolerance, where faults are treated as inevitable, and designers aim to guarantee that the system remain functional despite them.

In a vast majority of cases design for reliability techniques are enough to achieve the stated objectives, however for truly mission critical equipment, neither of these two approaches alone is sufficient, as they largely complement each other. While avoiding failure by design is certainly preferable, it is not always possible, either for technical reasons such as cost, weight or volume restrictions or, more fundamentally, because the probability of a random unforeseen event leading to a design failure cannot be eliminated entirely (i.e.

cosmic rays or high energy gamma ray bursts hitting semiconductor devices). To address this possibility, a fault-tolerant approach must be followed, where the system is partitioned in smaller subsections, that can work independently of each other, to keep operating even in these circumstances. This strategy is also not viable when applied alone, since the use of insufficiently reliable components can lead to multiple and cascading failures that quickly defeat the available level of redundancy.

2.2 Design for Reliability

The design process for mission critical systems must include a comprehensive reliability analysis study that can, through physics of failure techniques [19], assess the most common modes of failure, quantifying their importance in terms of impact to the desired figures of merit. When performing these analyses it is important to consider the impact of all components in the design towards the global failure, including the manufacturing process as it also plays a vital role in the reliability of any manufactured good, as discussed in [20]. Thus detailed knowledge of the statistical distributions in material and process related parameters is fundamental, as any divergence with those assumed for reliability analysis will completely invalidate all results, leading to large discrepancies between real and forecasted and failure rate.

It is for these reasons, that the application of a complete reliability analysis, when working at a purely architectural level is rarely warranted, even when the realization of a laboratory scale demonstration system is required, as this will be inevitably very different from the series production units to be deployed. Resulting in potentially misleading results.

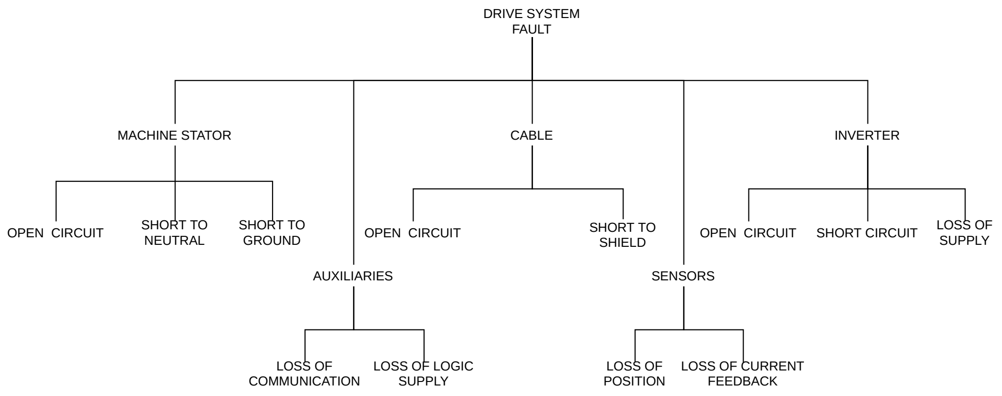


Figure 2.1: Possible fault types in a drive system

2.3 Hardware failure modes

Before evaluating a system ability it is critically important to examine the possible failure modes in a typical actuation system, their likelihood and the possible mitigations available at the design level. In Fig. 2.1 is shown a hierarchy of faults categorized on the base of the component they involve. Starting from the electrical machine itself, only stator faults have a direct impact on the design of a fault-tolerant drive system, as no mitigation or tolerance to them is even theoretically possible from the electrical domain, like in the case of a mechanically locked rotor. The simplest class of problems to deal with is open circuit faults in one or more phases; due to their stable and self insulating nature they only require control level changes. Another class of failure mechanisms is short circuit connections in the stator coils, of which two types are possible. Short to neutral (also known as inter-winding shorts), can be treated similarly to the previous class, with the affected phase drive being excluded, in order to avoid any current flow through the damaged components. The second, much more serious category of machine issues that can arise during operation comprises short circuits to grounded components of the machine, like chassis or slot walls. These, while being very similar to inter-winding shorts, must be treated differently, as it is not possible to fully isolate the affected phase. It should also be noted that even a small amount of current flow through a grounded component will likely pose electrocution and electromagnetic interference hazard. Making this type of issues very difficult to deal with even when segmented multi three-phase stators are used, as long as there is any significant magnetic coupling between them.

Moving further upstream, in traditional, non-integrated, drive systems the cable linking machine and inverter has several associated failure modes. The open circuit one, having the same effects of an open circuit winding fault, is dealt with in an identical manner. Other problems that can affect cables are short circuits, either between phases or toward the shielding conductors and braids. These two cases need to be treated separately, as in the first case, where the conductors feeding two separate phases are shorted, a control level response can be sufficient to keep the system running, as long as enough redundant phases are present in the system, being the two affected machine windings now effectively in parallel. The second scenario, where a phase conductor is shorted with the cable shield, depending on whether this is grounded or not, can either be unrecoverable, for reasons similar to the corresponding machine fault, or it might be addressed by shutting down the affected phase.

When considering a standard two level voltage source inverter (VSI), three separate classes of failure modes are possible. Open circuit faults arise when one or both the transistor groups for a phase can not be switched on, either because of an internal problem or due to a malfunction in the complex gate drive circuits. This fault represents a stable configuration that can again be dealt with through control level actions. Short circuit faults are also possible, where the device fail to turn off permanently connecting one of the machine terminals to either the DC side power supply. This second class of issues can be managed in the same way as the previous one, if external means to disconnect the offending phase, through contactors, thyristors or even fuses, are present. In a system without these provisions the transistor themselves can be made to act as fuses physically destroying them through a controlled overcurrent event.

Another issue that can arise preventing the inverter, and consequently the whole drive system to operate, is a loss of supply condition, where the incoming power feed is lost due to an external event. When a traditional power distribution architecture is employed, no mitigation against these type of problems is possible. However, with the advent of smart grids, even for vehicular applications, a multiple active bridge frontend [21] can allow the drive system to operate from multiple supplies, mitigating the risk of a complete loss of power. It should also be noted that as well as improving overall reliability, system efficiency and flexibility can also be increased, by adjusting the inverter DC-link voltage to the most appropriate one for the intended application.

Finally, faults in sensors and auxiliary systems, if not correctly managed during system design and integration can also bring down the entire system. With regard to rotor speed/position sensing, any issues in this subsystem can be solved by switching to a fallback sensorless current control algorithm. In case of a loss or degradation of current sensor data, depending on the degree of redundancy available in the system, the drive can either keep operating by reconstructing the missing current from the other ones, by taking advantage of the zero-sum nature of symmetric phase sets, or if not enough working sensors are remaining, the affected phase can be shut down, with the drive continuing operations with a reduced phase count. Other minor issues can present challenges if not considered, loss of communication, as an example, can be easily managed through the simple addition of fully redundant links, while in case of a loss of control logic supply, a simple

Component	Probability of fault
Machine Stator	$1.4 \times 10\times^{-8}$
Inverter & controller	$8.6 \times 10\times^{-5}$
Power supply	$5.4 \times 10\times^{-5}$

Table 2.1: Probabilities of fault broken down for each component of an actuation system

backup battery can keep these, typically low power systems, running for long enough avoid operational issues.

While obtaining a statistically significant description of the fault probability of each component, with information about each single failure mode and relative rate of occurrence is very challenging and well beyond the scope of this work. It is nonetheless important to have an estimate probability of failure for each component of the system, to avoid spending time and energy improving a non-critical part of the system. From the fault probabilities given in [22], and shown in Table 2.1, is clear that the reliability of both the inverter, with its controller and that of the power distribution system is far lower than the windings one, and as such these two parts of the actuation system need to be improved. Regarding the supply side much work is being done with the proposed integration of smart grids to vehicular power distribution networks [23], while the work presented in this thesis will focus on the second aspect, reliability of the inverter and its controller.

2.4 Hardware Fault Tolerance

Since the consequences associated with a catastrophic system failures, in mission critical applications can be extreme, the design process for electrical drives, when targeting them, cannot rely uniquely upon design for reliability techniques, a multi-tiered approach needs to be applied, to guarantee operation even in case of fault events. All these considerations point to the need for fault tolerance in these types of systems whereby redundancy is used to insure correct behaviour in all situations [24].

The first choice presented, when designing an electromechanical actuator system (EMA), from a fault tolerance point of view, is the type and quantity of faults that the system must be able to handle. Once this information is available one of several redundancy types, as delineated in [25], can be employed to reach the desired goal.

- **Complete subsystem redundancy**: The entire subsystem, comprising the load, is treated as a complete unit, and replicated, so that even in case of total failure, the overall desired functionality is not completely compromised. The main benefit of this approach is simplicity in terms of development, as regular non-Fault-tolerant components can be used. This solution is not always applicable or even desirable due to either the high cost, or the unreasonable volume/weight requirements.
- **Full actuator redundancy** Where multiple EMA are connected to the same mechanical load. In this case electrical machine, the electrical drive and any necessary controller are taken together as single component. Historically this solution has been the most adopted especially for small size machines. This solution is possible since the characteristics of the mechanical parts in a typical EMA have favourable and well understood reliability techniques
- **Machine drive redundancy** If the drive reliability is much lower with respect to the machine itself, as is typically the case, due to the large difference in complexity between the two, it can be worth replicating only the weakest component, as opposed to the whole system. This strategy can only be used in combination with electrical machines that can operate with both one or more three-phase winding sets.

- **Internal component redundancy** At this level redundancy is added within each component itself, like more phases or winding sets in order to keep a level of functionality in the component even after a fault, for this strategy to work the component must be able to not only keep working correctly after the fault but also to survive the fault transition.

Historically the first two solutions have been the most adopted, as they allowed to work with mostly standard components, avoiding the need for custom designs. The increase in complexity in both loads, electrical machines and especially drives electronics meant that the drawbacks of these approaches continued to grow with the rising cost, weight and volume of the individual components. Determining shift in both academic and industrial interest toward the last two type of redundancies. As with these, many non-critical components can be shared between the various redundant sections.

2.4.1 Complete subsystem redundancy

This type of redundancy architecture relies on the duplication of entire subsystems, so that in case of a failure the affected portion can be safely excluded, leaving the rest of the system unaffected. An example of this is the propulsive system of modern airliners, composed of multiple engines, that can be independently excluded in case of fault. While full replication can offer the best fault isolation ensuring excellent protection from cascading failures, it is only applicable in very specific scenarios, since such a large degree of replication is very expensive both in terms of performance and monetary cost. As a result it is typically only used when the replication is also beneficial from a performance perspective.

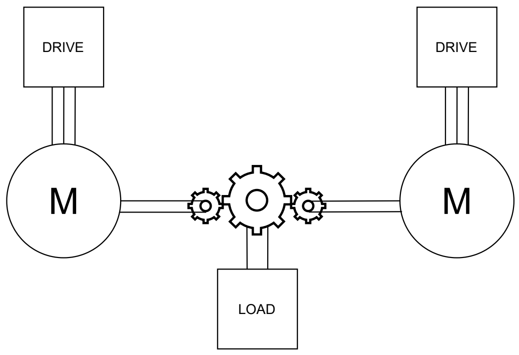


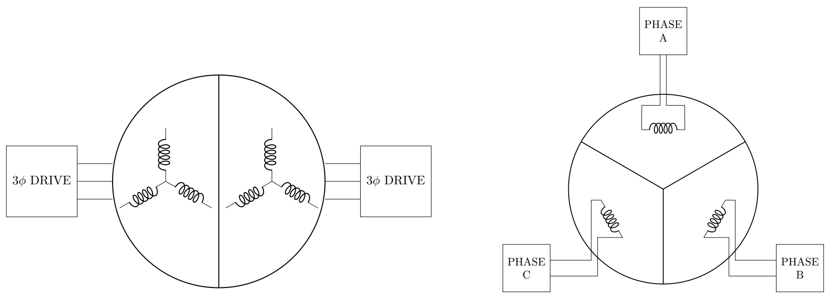
Figure 2.2: Full actuator redundancy diagram

2.4.2 Full Actuator Redundancy

This type of redundant actuation architecture, conceptually shown in Fig. 2.2 is the earliest one employed where full system redundancy is not possible, in [26] a dual machine solution is developed for potential usage on primary flight surface where machines are coupled through a common gearbox onto a single shaft. Another option is the use of axially segmented machines, as in [27] where two completely separate stators can act independently on a single rotor. In both cases the two electrical machines are completely unrelated to each other, with only a mechanical side connection. In case of fault in one of the two electrical systems, the other system can continue to work completely unaffected, when the faulty part of the system is disconnected. Another advantage of this solution is that large degree of separation between system partitions allows the use of standard components, potentially lowering total cost.

2.4.3 Machine drive redundancy

As widely attested in the scientific literature [34–36], the reliability of electrical machines is much higher than the one of machine drives, as the much higher complexity of these electronic systems, drive up failure rates. To avoid the duplication of an entire machine when its reliability is high



(a) Segmented multi three-phase machine [28–30] (b) Single phase drive redundancy [31–33]

Figure 2.3: Machine drive redundancy diagram

enough for the required application segmented machine designs can be used.

In these systems the machine is partitioned, as shown in Fig. 2.3a with several independent sections, carefully designed in order to minimize mutual couplings in case of failure. Each one of them is paired with its own machine drive, to guarantee fault-tolerance increasing overall reliability. Several examples of this strategy are studied in literature [28–30] with up to four independent three-phase sets. An additional benefit of this partitioning is the simplification of the drive design itself, lowering the peak current requirements.

Another possibility in this space, as shown in Fig. 2.3b is the use of separate per phase drive units, as proposed in [31–33]. This architecture while requiring a more complex and custom set of drive components, can be beneficial for small machines, where the per phase current is low enough to allow the use of a standard machine design, with less stringent requirements regarding the minimization of coupling between sections.

2.4.4 Internal component redundancy

The last possible redundant actuation architecture, involving the least amount of duplication, consists in the use of components that can sus-

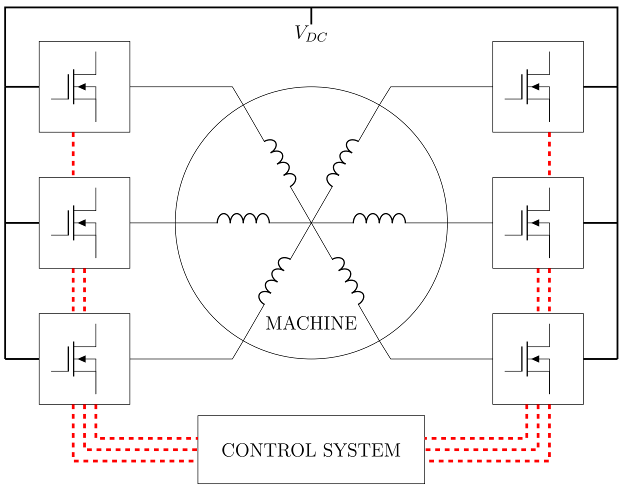


Figure 2.4: Distributed drive architecture

tain one or more faults without loss of functionality and with only minimal loss of performance. When this solution is applied correctly, it can achieve equivalent levels of protection as the previous ones, while potentially being more power dense and economical.

The components designed to be used for these applications must be explicitly partitioned into several independent subsystems, with the boundaries between them carefully considered in order to avoid faults cascading through multiple sections. From a machine design point of view multiphase machines, as shown in [37, 38], have been thoroughly researched, the use of a single neutral point allows a machine to sustain a higher number of faults with respect to a corresponding multi-three phase one, where the performance degradation when running in a faulty state is much more severe [39]. It should be noted that where both galvanic isolation between multiple supplies and high performance in a post-fault scenario are needed a hybrid approach can be followed, with machines composed by multiple polyphase sets.

On the drive side several classes of fault-tolerant architectures have been proposed over the years. The first is based on slightly modified regular three-phase drive systems, such as presented in [40–42], where some fault tolerance is achieved without meaningful system level changes, enabling these drives to be even used in retrofits. Another class of fault-tolerant drives, proposed in [43, 44] consist of simple integrated multiphase drive, where the structure of a conventional three-phase VSI inverter is extended to cover the additional phases. The simplicity of this approach makes it appealing, however the high degree of coupling between phases increases the risk of fault propagation. A more promising approach, which can be seen as a hybrid between this and the single phase drive units approach (previously shown in Fig. 2.3b), is the use of a distributed drive architecture, as shown in Fig. 2.4, where the machine neutral is still connected, but each phase is driven by its own power electronics. In these type of systems [31–33, 45], only a single DC link is used, as the single neutral point would break the galvanic isolation between separate power domains. In a system with physically separated phase drive modules, the parasitic inductance of the DC-link connections between phases limits the spread of faults, decreasing the chance of them cascading and bringing down the entire drive.

2.5 Fault-tolerant Control

Along with hardware physically capable of enduring partial faults without compromising the whole system operative status, it is also important for the current control strategy to be resilient to failures. The solutions proposed in scientific literature can be categorized in three groups, as shown in Fig. 2.5: Field Oriented Control (FOC) derived techniques, where the controllers act in a rotating frame of reference synchronized with the rotor; Fault-tolerant DTC, which extends traditional Direct Torque Control; and static reference frame control where the current controller acts directly on each phase current completely independently.

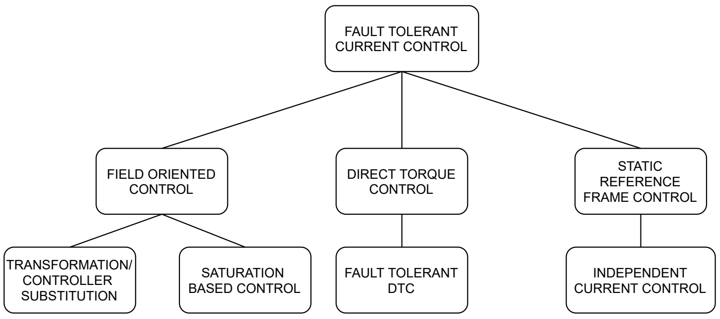

FAULT TOLERANT CURRENT CONTROL
FIELD ORIENTED CONTROL
DIRECT TORQUE CONTROL
STATIC REFERENCE FRAME CONTROL
TRANSFORMATION/ CONTROLLER SUBSTITUTION
SATURATION BASED CONTROL
FAULT TOLERANT DTC
INDEPENDENT CURRENT CONTROL

Figure 2.5: Fault-tolerant current control classification

2.5.1 Control of multiphase machines and Vector Space Decomposition

Given the sinusoidal nature of the problem and high degree of coupling present in a multiphase machine, the direct development of an effective current control strategy is not an easy task, consequently the use of reference frame transformations is very common, allowing both for a reduction, or even elimination, of the couplings while permitting the use of quasi-static DC values in the controllers. The most popular of these approaches makes use of the Vector Space Decomposition (VSD), introduced in [46], where the n-dimentinal space is partitioned in a number of 2-dimentional subspaces orthogonal with respect to one another. Moreover, the transformation is constructed explicitly to concentrate all fundamental frequency components of the currents in a single subspace, usually denoted $\alpha - \beta$, while all higher order harmonics are relegated to the others. This determines that, once neglected the second order effects, only vectors in the first sub-plane contribute to the electromechanical torque generation process. To further simplify control design, a rotation matrix (Park transformation) is used to go from static to a rotating frame of reference, allowing the use of traditional PI controllers that are not able to achieve zero steady-state error when tracking sinusoidal references. Several variations of this traditional approach have been proposed in [47–49].

The most notable downside of traditional VSD based control strategies is the lack of fault tolerance, since in case of failure, the controllers will start to produce set-points that are likely invalid or not reachable in the post fault scenario, leading to poor tracking and eventually instability. The root cause of the issue is the dependence of both direct and inverse VSD transformations on the state of health of the system. The most natural solution to the issue is the adoption of a fast fault detection algorithm that can, once identified type and location of the fault, trigger their substitution with different ones, specifically chosen depending on system current health state. This strategy, while conceptually very simple, has some serious downsides and complications that make a robust and general purpose implementation very difficult, especially for systems with a high phases count. The transformation swap, needs to be handled smoothly and carefully to avoid further destabilization of the system due to the controller internal state mismatch between pre- and post- changeover scenarios. Fast fault detection is also problematic as there is an inherent trade-off between robustness to sensor noise and bandwidth. The biggest problem for these strategies is however their limited scalability, as $2^{n_{ph}}$ system states are possible from a health perspective, each one associated to a different set of transformations, leading to an exponential growth in complexity with the number of phases.

A second approach to VSD based fault-tolerant operation, introduced in [50], relies on heavy saturation of the controllers available output for the non-torque generating subspaces to limit the influences of cross coupling and errors present in the post-fault scenario. This effectively solves all the problems of the regular FOC, at the price of loss of control authority and consequently tracking for the non $\alpha - \beta$ plane currents.

2.5.2 DTC and others

The use of direct torque control has also been extended for use in fault-tolerant multiphase drives, as proposed in [51, 52]. The common idea behind these methods consists in the fast detection of the fault coupled with the substitution of the switching states look-up table with the appropriate one for the post-fault health state of the system. This avoids the first issue that plagues FOC implementations, as no substitutions upstream of the controllers need to be made, allowing them to keep working as in the nominal case, however it still suffers from the same scalability issues, being the switching states table dependent on the specific fault configuration of the system. Fast fault detection is also needed as before, with similar considerations with respect to the bandwidth.

Few completely different approaches are, shown in [53] and [54], where the control is done on a per-phase basis without machine specific transformation steps, either through the use of resonant controllers, that can track sinusoidal references, or through PI controllers in a rotating reference frame synchronized with the desired output sinusoids. These techniques maintain one-to-one correspondence between phases and controllers, eliminating the need for health state specific transformations. This solves both the issues of poor scalability and stability as the remaining phases are only indirectly affected by the fault. The reconfiguration of the system to its post-fault configuration is also much easier requiring only a routine reference change, as opposed to a complete controller substitution.

2.6 Distributed System Architectures

2.6.1 Introduction to distributed systems

Distributed systems are of common use in many fields where the properties of scalability or fault tolerance are desired, from computing [55] to communications between remote nodes [56] and large scale control systems [57]. Even in power electronics, there are clear examples of large-scale use of distributed system to tackle the problems that arise from the massive needs in HVDC power transmission, with Modular Multilevel Converters (MMC) [58]. In the context of fault-tolerant drives a distributed structure can help increase overall reliability. The effective design of a distributed system requires addressing several challenges in the areas of communication, topology and synchronization. These will be highlighted in the following sections.

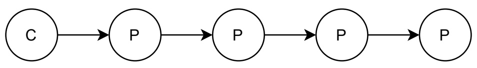
2.6.2 Communication

The foundation of any distributed system is the communication network that links the various nodes together and its choice is instrumental in guaranteeing both performance and fault tolerance. Due to the hard real time constraints encountered in a power electronics system, introduced latency is the key figure of merit that needs to be evaluated during the protocol choice phase. Several projects in the MMC space have made successful use of communication systems based on the EtherCAT protocol [59–61]. This was designed by Beckoff Automation for process control and industrial automation and, when used in combination with proprietary lower layers, as opposed to the standard Ethernet hardware used for less critical systems, can reach sub millisecond cycle times. The main benefit of the adoption of this protocol is the relative ease of implementation, due to the large number of standard compliant software IP available. The biggest drawback of this approach is the large software complexity needed to support a standard compliant higher layer implementation, that can interoperate seamlessly with other components in the ecosystem. The use of a ring topology also limits is usefulness in fault-tolerant systems.

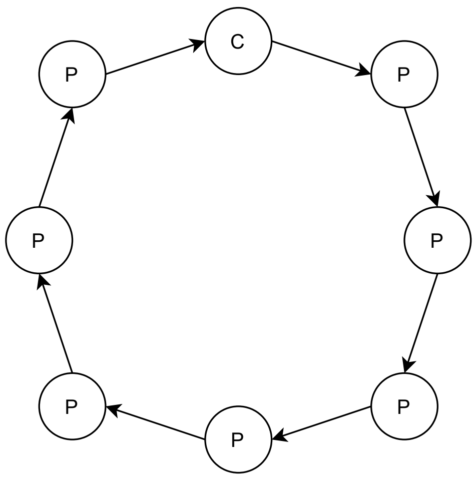
Several research teams have also proposed semi-custom communication protocols that leverage a standard data link layer for the second level of the ISO/OSI network stack, coupled with custom topologies and ad-hoc physical layers to address the challenges posed by the high voltage environment typically present in MMC converters. In [62], Ethernet is used with a custom daisy-chained node topology allowing the system to achieve single digit microsecond latency for small number of nodes. While this approach shows promise, it is still unsuitable for high availability mission critical applications since a cable or node failure can cut off a large part of the network from the main control node. In [63] A "Redundant Star" topology is introduced, allowing multiple central controllers to communicate with the power nodes, guaranteeing much better fault tolerance, as no single point of failure is present. Another semi-custom solution, presented in [64] uses the Xilinx Aurora chip to chip interconnect along with much higher frequency optical fibre components, as specified in the ARINC 802 [65] and following standards, in order to greatly reduce the communication latency. The downside to using these types of components is that only very few, fast enough, dedicated transceivers are present in modern FPGA that can support such a high data rate. Limiting the possible topology choices to ring or daisy chain, which are less suitable for fault-tolerant systems.

2.6.3 Topologies

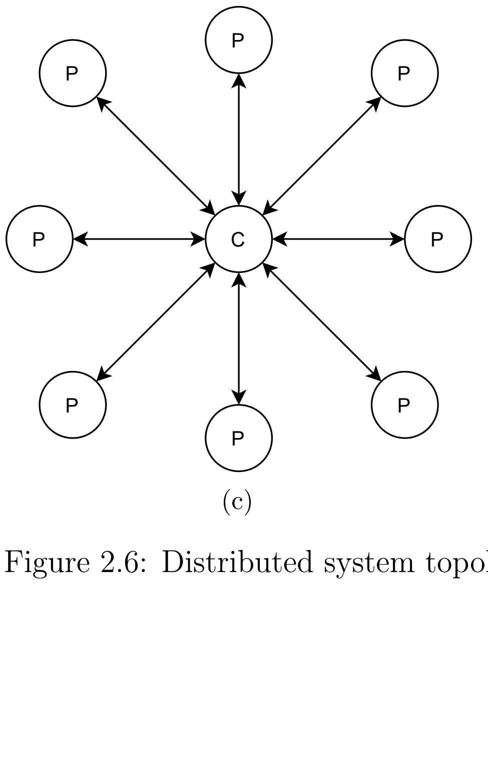
As already anticipated in the communication section, both the physical and logical network topology are extremely important as they largely define the



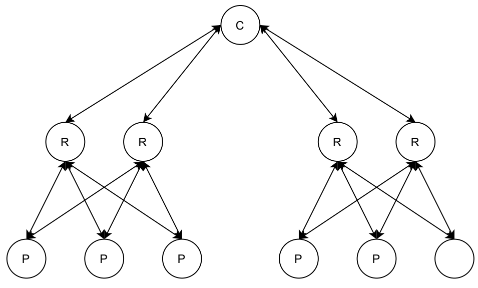
(a)



(b)



(c)



(d)

Figure 2.6: Distributed system topologies (a) line, (b) ring, (c) star, (d) tree

fault tolerance of the entire system against communication and controller failures. The line and ring topologies, shown in Fig. 2.6a and 2.6b, very popular in the MMC space due to their easy cabling, are undesirable for high availability mission critical applications as the failure of a single node can cut off large parts of the system from the main central controller. Redundant arrangements that use two parallel networks can mitigate some failure modes, making the system resilient to a single link failure, however they do not guard against node failure, as the physical layer of each node is required to receive and retransmit each packet. It should be noted that while the physical network processing layer where the packet retransmission happen is typically extremely reliable, much more common power supply issues can still render it inoperative. Another disadvantage of this topology is the poor scalability as even though the bandwidth required by each single node is small, all the links must be sized for the total cumulative system bandwidth. This not only increases cost but also limits the total amount of nodes that can be linked in a single network. Finally, the cycle latency with these topologies grows linearly with the number of nodes limiting the performance of large systems.

A different type of structure, shown in Fig. 2.6c is the star, where one or more controllers are connected directly with all the power nodes. This topology, while requiring a larger central controller, and more complex cable routing, has all around better performance. The end to end latency lower, drastically so in large systems, with respect to the previous two topologies, as the transmissions can happen in parallel. Each link also requires much less bandwidth as only the data for a single node need to be transmitted. These two advantages combined allow the use of much lower transmission frequencies, with the associated cost reductions, while attaining cycle times comparable with those reachable with the previous topologies. From a fault tolerance perspective this topology is also much better, as a failure in either a node or a link brings down only the affected component, instead of the entire system. Scalability to very large number of nodes is still an issue as the main controller becomes quickly IO limited, as the size and complexity of cabling quickly renders this topology impractical.

Finally, we have the tree layout, shown in Fig. 2.6d. This can be seen as an extension of the previous, as one or more intermediate routing/aggregation layer are interposed between the system controllers and the power nodes. Their addition reduces the growth in the number of links that any individual component must have from linear to logarithmic. Allowing the entire system to scale to potentially hundreds, if not thousands of nodes. The physical cabling of the system is also reduced as the routers can be scattered throughout the system keeping the majority of connections local. The main price to pay for this ability is the increase in both bandwidth needed for the router to controller links, that need now to serve multiple nodes, and the introduction of latency with each additional layer of routing between controller and end nodes. From a fault tolerance perspective a network with a tree topology can be less than ideal, as the nodes in the intermediate routing layer aggregate the traffic from many nodes. In case of a failure of one of them a large part of the network will be cut off from the root of the tree, which hosts the controller, as shown in Fig 2.7, where a single fault is able to bring down 50% of the network. Consequently, when this topology is used in a mission critical application, the use of redundant routers is necessary.

To evaluate the scalability of each network topology Table 2.2, shows the relationships between network size and several parameters of interest, such as maximum number of needed channels, maximum number of interfaces needed on a single node and worst case latency; with n being the number of

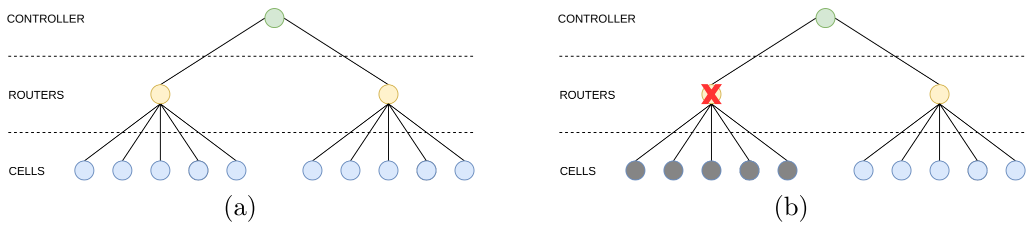


Figure 2.7: Tree network pre- (a) and post- (b) router fault.

nodes in the network, m the number of nodes supported by each router in a tree topology, τ_p a single hop transmission delay, r the number of routing layers in the tree and B the bandwidth required by a single node. The number of individual channels needed by each topology for a generic network, is comparable for all topologies, with a linear dependence from the number of nodes. When considering the maximum number of connectors needed on any node, large differences are present between the different topologies. For line and ring only 2 links are needed in the worst case scenario, regardless of the network size. The star topology, is the worst by this metric, as the central node needs as many connectors as there are nodes in the network; all other nodes however only need a single link to the central controller. This determines its relatively poor scalability. Finally, the tree network layout is the most balanced in this respect, as it allows the designer to choose how many nodes will be supported by a single router while allowing the network to grow up to very large number of nodes. Another important property to consider is the maximum bandwidth required by any single channel in the network, as it impacts the cost. The line and ring network layouts are the worst in this regard as the entire traffic of the whole network needs to be transmitted through the entire network. By contrast in the star topology all the links need only the bandwidth for a single node. When considering the tree topology, a distinction must be made between the vast majority of links, those between routers and the leafs of the tree that only require

Topol-ogy	Number of channels	Interfaces per node	Latency	Band-width
Line	$n-1$	2	$n\tau_p$	nB
Ring	n	2	$n\tau_p$	nB
Star	n	n	τ_p	B
Tree	$n+m$	m	$(r+1)\tau_p$	$B\|mB$

Table 2.2: Overviews of performance scaling with network size for each topology

enough bandwidth for a single node and the router/controller links, which need enough bandwidth for the m nodes serviced by that router.

2.6.4 Synchronization

Two different level of synchronization must be achieved, the first is at a physical layer, where the clock frequencies of all the nodes in the system need to be equal for communication to be even possible. Two possibilities for this are the distribution of a single clock signal throughout the system, and the use of clock and data recovery circuits, that can lock to and extract clock information from the transition of the communication protocol data lines.

Logical Synchronization, on the other hand, is a fundamental area of complexity in distributed systems, to allow them to act cohesively. In power electronics and especially machine drives, this is even more important, as timing errors between the nodes can result in either increased losses, or even damages to the system since a significant amount of stored energy can be released quickly and destructively. From this point of view a big distinction must be made between star/tree and line/ring network topologies. In the first case, since parallel transmissions are possible, consequently a simple periodic synchronization packet can be used in order to establish a timing baseline that all components of a system will follow. On the other hand, for line/ring topologies, the synchronization is much more problematic, as the inherent serial nature of the transmissions, coupled with the significant latency for the last node, means that it is borderline impossible to have a communication reach all the nodes at the same time. One way of solving the problem without resorting to additional hardware, is the measurement of the internode delays. A dynamic compensation process is consequently required as one time static calibration would be both too stringent, requiring a particular ordering of the modules in the chain, and vulnerable to drift. To do this several algorithms can be used [66, 67].

Topology	Scalability	Fault Tolerance	Cabling Complexity	synchronization	Applications
Line	$\sim$	$--$	$+$	$-$	Small to medium non mission critical networks
Ring	$\sim$	$-$	$+$	$-$	Small to medium non mission critical networks
Star	$-$	$+$	$--$	$+$	Small mission critical networks
Tree	$+$	$\sim$	$\sim$	$\sim$	Large mission critical network

Table 2.3: Overview of Network topologies properties and applications

A qualitative overview of various properties for each network is shown in Table 2.3, where $+$ denotes a strong performance area, $\sim$ indicates an average one, and $-$ a week point for each topology; along with the ideal applications the topology should be used in. In particular the line and ring topologies should be used, where low cabling complexity is desired, like in very space constrained applications; however the low fault tolerance and average scalability, with a latency that is strongly dependent on the network size, make ring and line networks a poor fit for mission critical and latency sensitive applications. The star layout by contrast shines where fault tolerance, low latency and ease of synchronization are valued over everything else, making this network type ideal, when coupled with a redundant central

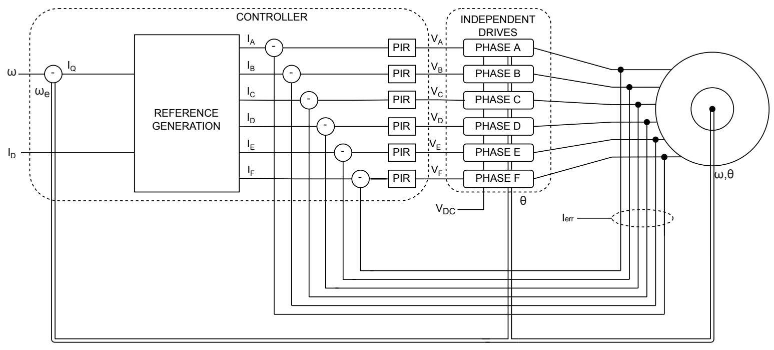


Figure 2.8: Diagram of the proposed drive system

node, for small networks where failures need to be contained as much as possible. Finally, the tree topology, has unmatched scalability, allowing an exponential growth of the network size with the addition of few routing layers. Its fault tolerance, while not as good as in star networks, is still acceptable for use in mission critical applications, especially when used in conjunction with a redundant routing layer.

2.7 Proposed fault-tolerant machine drive

Having considered all the issues and technical solutions discussed in the previous sections, the system configuration shown in Figure 2.8, was chosen to develop a scalable fault-tolerant machine drive architecture, able to scale up to a high number of phases while still being able to work with modern wide band-gap based high frequency capable power devices. To achieve these goals a distributed approach is followed with each phase driven by a single power cell. The control electronics is linked to each drive unit by a fully custom digital communication protocol that minimizes latency, while decoupling the power cell internal implementation details from the rest of the system, simplifying the integration phase. The use of a media independent serial link as the physical layer enhances system flexibility allowing both electrical and optical signalling. Proportional Integral Resonant (PIR) regulators are used to control the current in each phase while avoiding coordinate transformations, whose computational complexity increases polynomially with the number of phases.

The developed system, in its current configuration, is intended for mission critical intrinsically redundant applications, where a loss of input power while undesirable can be tolerated, with the Fault-tolerant characteristic of the drive providing graceful performance degradation instead of complete loss of functionality. These applications include:

- Electro-Mechanical Actuators (EMA)
- Electro-Hydraulic Actuators (EHA)
- Starter-generators
- Pumps

With respect to propulsion applications, where multiple insulated supplies are needed to be available to maintain a level of operation at all costs, the system shown in Figure 2.8 is not suitable, and it would need to be modified either by replicating the whole drive chain, or by splitting the current one into two isolated three-phase sets, and in both cases feeding the partitions from different supplies.

Chapter 3

Power cell Hardware

The hardware related work presented in this thesis is finalized to the development of an experimental platform to demonstrate the viability of distributed, networked architectures for use in high frequency mission critical machine drives. The developed and validated hardware component will also be able to serve as the foundation for further research on the topics of: large scale high frequency distributed control, tolerance to communication faults in networked converters and cyberphysical security in hard real time, low latency control networks. Due to the limited potential for innovation in the logic control hardware space, a pre-existing Field Programmable Gate Array (FPGA) based control platform was used, focusing the effort on the power electronics side and control logic, with associated firmware and software.

3.1 Drive Optimization

The design of the power cell hardware followed a two-step process, first the detailed operating parameters for the drive have been determined to fit the intended application, through an optimization process, then the detailed hardware design has been finalized. The initial design constraints, shown in Tab. 3.1, specify the basic system level requirements for the overall

Parameter	Value
Minimum DC-Link Voltage	540 V
number of phases	6
Rated output current	180 A
Maximum Heatsink Temperature	75 °C

Table 3.1: Initial Design Constraints.

drive, regarding minimum DC-link voltage, and output currents, defining the maximum output power available for the load, as well as number of output phases and maximum allowed heatsink temperature. These were the main inputs for both the optimization, and the following design process discussed in the following sections.

The first step in the design of the power hardware was an optimization intended to explore the possible design space, selecting a balanced converter setup and operating point upon which to base the design. The aim of this process is the selection of nominal values for the following parameters:

- Number of parallel switching devices in each cluster
- Model of switching devices
- Switching frequency
- DC-Link Capacitance

To guide the optimization algorithm towards the desired goal, a combination of the performance indicators in the following set have been considered for the cost function:

- Maximum Power Los
- Average Efficiency
- Volume

- Input Voltage ripple
- Cost
- Output current THD
- Heatsink temperature

Reliability parameters, like the mean time between failures (MTBF) have not been added to this optimization, due to their dependence on too factors unknown in this early stage of design. A scientifically accurate reliability analysis is only possible when manufacturing and assembly parameters are known. As an example the exact chemical composition of the solder paste or the reflow profile used for the power devices are needed to evaluate its mechanical [68] and metallurgical properties of the joint [69].

While many possible approaches to multi-objective optimization are available, the choice was done to use an a-priori linear scalarization technique to formulate the problem as single objective, this allows the use of much more mature tooling specially to deal with the mixed integer nature of the device choice, as it is only limited to the commercially available transistors.

$$T(X) = \sum_{n=1}^{7} f_i(x_i) w_i \tag{3.1}$$

The resulting target function is shown in Eq.(3.1), where $f_i(x_i)$ are the objectives and w_i is the scalarization weight associated with each one of them.

The weights are set up by the designer, also known as the Decision Maker (DM), to represent the relative importance of the various aspects of a design and should consider the variability range of each factor. Their choice, if done properly, can be used to steer the outcome in the desired direction,

Factor	Weight
Power Loss	2.2
Average efficiency	10
Volume	1
Input Voltage Ripple	0.5
Cost	0.1
Output Current THD	1
Heatsink temperature	0.6

Table 3.2: Scalarization weights

while still guaranteeing the achievement of an optimal solution. The values used for the design are shown in Table 3.2.

3.1.1 Power Loss

One of the most important factors in the cost function is the maximum power loss term, as it directly impacts the operating envelope of the drive in multiple areas, especially regarding the available output current. Several losses contributions have been considered to obtain the most accurate estimate possible. The first is from conduction losses, averaged on an output sinusoid cycle.

$$P_{Cond} = 2n_{phases}n_{cluster}\frac{2}{\pi}R_{DS(ON)}\left(\frac{I_{OUT}}{n_p\sqrt{2}}\right)^2 \tag{3.2}$$

These are only dependent on the magnitude of the output current, being MOSFETs fully bidirectional device, when the effects of dead-time are ignored, as it only constitutes a minimal part of the period. Where n_{phases} is the number of phases, $n_{cluster}$ is the number of parallel transistors in each cluster $R_{DS(ON)}$ is the transistor on state resistance and I_{OUT} is the peak output current.

To evaluate switching losses manufacturer provided data was used, since not enough parameters are available for analytical modelling. As a starting point for this process the energy losses per commutation values given in the data sheet are used to linearly interpolate exact parameter values in any point of the drive operating envelope leading to

$$P_{SW} = n_{phases} n_{cluster} (E_{ON} + E_{OFF}) f_{sw} \tag{3.3}$$

where E_{ON} and E_{OFF} are the per switching event energy losses and f_{sw} is the switching frequency

Both reverse recovery and gate driving contributions have been omitted as negligible with respect to the others. It should be noted that due to dead time presence only half of the commutations will count towards switching losses, as the diode will start conducting the load current, which must remain constant, and result in a zero voltage turn off transition.

3.1.2 Average Efficiency

In order not to skew the overall design effort too much toward the maximum output power area, where the highest amount of losses will naturally fall, the average efficiency has been evaluated, and added to the optimization cost function. The losses, over the whole speed and torque operating ranges, have been calculated following the methods described earlier and used to evaluate the drive efficiency. They are then used to calculate the global average efficiency. Since this value must be minimized, as opposed to most others, it will be subtracted from the cost function.

3.1.3 Volume

The volume used by a drive depends on a lot of factors, many of economic nature, massively complicating the process of finding a precise estimate, starting from the drive specifications alone. A figure of merit, that is able to acceptably capture the influence of the considered parameters on the global volume can be obtained by calculating the volume needed by the two biggest internal components, that are active devices and DC-link capacitors.

$$V_{DRIVE} = 6nV_{DEVICE} + k_{cv}C_{DC}V_{DC} \tag{3.4}$$

The first factor is obtained by multiplying the total number of MOSFETs present in the design by the volume required by each one. While for the film capacitors, used with this class of voltages, the volume is assumed proportional to capacitance C_{DC} and blocking voltage V_{DC}, through the constant k_{cv}.

3.1.4 Cost

A drive cost, similarly to the volume, is also difficult to estimate without a precise production plan in mind. This notwithstanding it is still a very important factor in the definition of the converter specification, for this reason the cost of the active devices

$$COST_{DRIVE} = 2n_{phases}n_{cluster}COST_{DEVICE} \tag{3.5}$$

has been added to the optimization target, as they constitute the single biggest cost item in the Bill of Material (BOM) of the whole drive, especially when using wide bandgap devices.

3.1.5 Output Current THD

Output current total harmonic distortion (THD) is another crucial figure of merit for a machine drive since, as shown in [70], it can greatly affect the performance and losses of driven machines. To take this into account the output current weighted THD (WTHD) was taken as a factor in the cost function. The calculation was done following the methods shown in [71].

3.1.6 Input voltage ripple

Following the method shown in [72], the input voltage ripple of the converter was added to the cost function. In order to correctly capture the relationship between switching frequency and input filtering capacitance volume.

3.1.7 Heatsink Temperature

The heatsink temperature was considered, to model the thermal behaviour of the system and guarantee that the losses generated during operation can be effectively dissipated.

$$T_h = T_J - P_{LOSS} R_{TH}; \tag{3.6}$$

This ensures that the junction temperature of the devices is maintained always within the limits of the device safe operating area. The inclusion of the heatsink temperature, in the optimization, is the reduction of thermal stresses on the transistors, potentially enhancing reliability. Where R_{TH} represents the overall thermal resistance of the chosen liquid cooling system and T_J the maximum acceptable junction temperature. As with the average efficiency, this factor needs to be maximized, so its contribution has been subtracted from the cost, instead of adding it.

3.1.8 Optimization Process and results

Genetic Algorithm

To solve the described optimization problem an evolutionary approach was used through a genetic algorithm. This class of methods tries to find a globally optimal solution by mimicking the biological process of natural selection and evolution. In these algorithms the inputs are treated as a chromosome, while each solution, defined by a specific set of chromosomes, represents an individual. At the start of the process the entire population, composed of few hundred individuals is selected randomly, then iteratively the fittest ones are selected using the defined cost function, and used to produce the next generation, through a mix of recombination and random mutations. This loop is then carried on until either a selected cost threshold is exceeded, or there are no more fitness improvements.

Results

To explore the results of the performed multi-objective optimization a statistical sample of the last generation input domain was chosen and used to evaluate all the separate factors composing the cost function, resulting in a set of points defining the valid group of solutions. The cloud of points in this multidimensional space is delineated by a boundary called Pareto frontier, the solutions falling on this surface can be considered optimal, while all the others are defined as dominated solutions. Unfortunately the direct visualization of this surface is not possible due to the high number of dimensions, consequently a series of two-dimensional slices are extracted showing, in figure 3.1 the most interesting views of the Pareto frontier with respect to the selected solution, whose associated optimized variables values are shown in Table 3.3, and denoted in the plot by the red marker. It should be noted that due to the projection operation inherent in the vector space dimension reduction, the chosen solution may be only close to the Pareto frontier in any single graph as the associated solution, as the associated solution could be dominated, and consequently undesirable in other dimensions not shown in the graph.

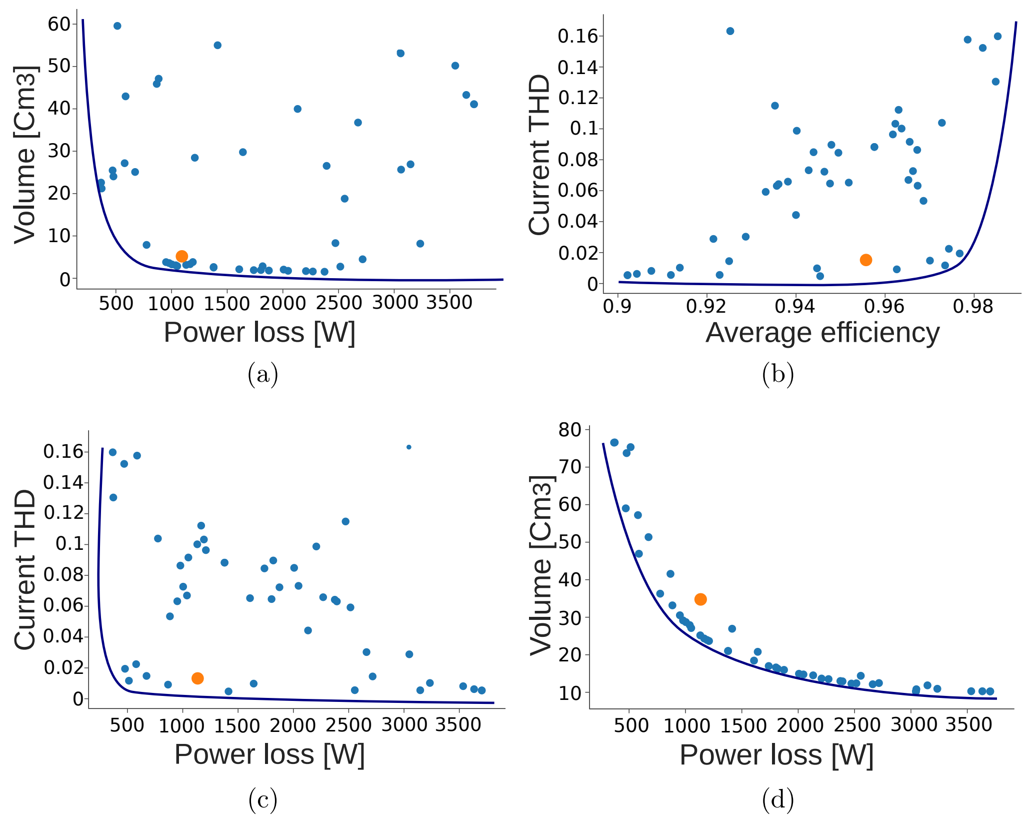


Figure 3.1: Selected Pareto set plots

Parameter	Value
Transistor model	C3M0065090J
Number of parallel devices	7
Switching Frequency	97 kHz
Minimum DC-Link Capacitance	3 µF

Table 3.3: Hardware Optimization Results

Sensitivity Analysis

Starting from the optimization process results, shown previously, a sensitivity analysis was performed. Allowing both for a better understanding of the various trade-offs involved in the selection of a correct starting point for the detailed hardware design, and also acting as a first layer consistency check to validate the optimization model. A one-at-a-time (OAT) approach was followed, where all performance indicators are repeatedly evaluated while sweeping the value of each input variable separately, with all other keeping their initial value, allowing to unambiguously discriminate the effect of each input on all the evaluated metrics. While it is not important at this early stage in the design process, it should be noted that the higher order effects caused by the variation of multiple parameters at the same time are not captured with this technique. Consequently, different sensitivity analysis methods should be used for late stage design characterization, where the mathematical models used capture correctly all the higher order effects.

The first parameter to be varied is the number of parallel devices used in each switching cluster. The results, shown in Figure 3.2, demonstrate, as expected, a hyperbolic dependency of the maximum power losses from the number of devices, with a comparable and compatible behaviour of both the average efficiency curve and the maximum allowed heatsink temperature. With respect to this last quantity, the necessity of sub-zero temperature cooling when few paralleled devices makes these theoretically valid configurations not practically usable due to the challenges of working with sub ambient temperature fluids, especially when close or below the dew point. Costs and volume linearly increase with the number of devices, while ripple and Current THD are not directly affected and are consequently not shown in the plots.

The next input variable to be examined is the switching frequency, whose

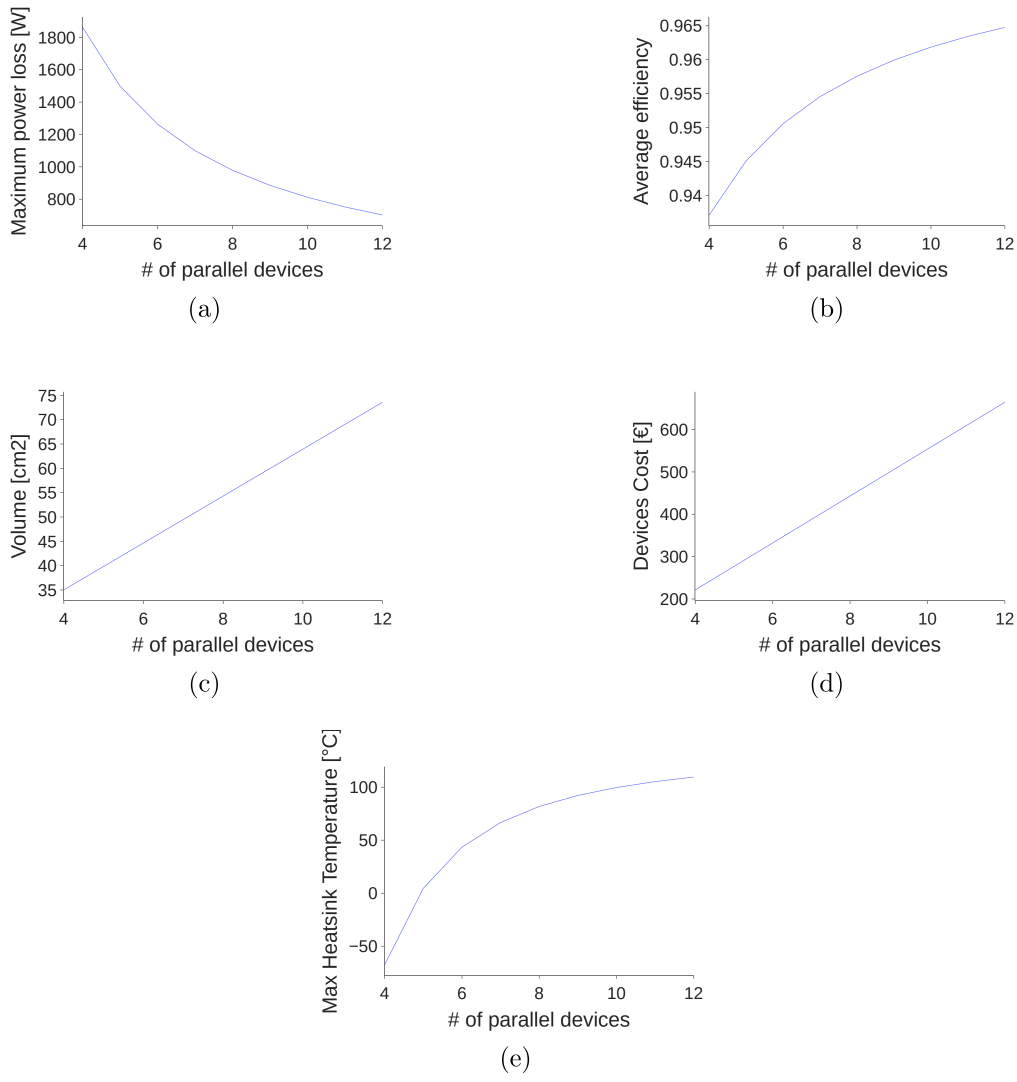
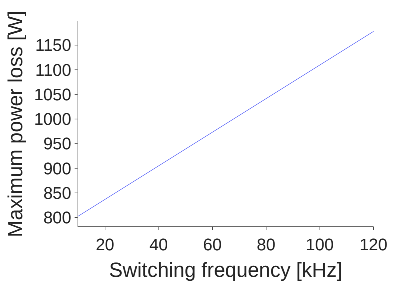
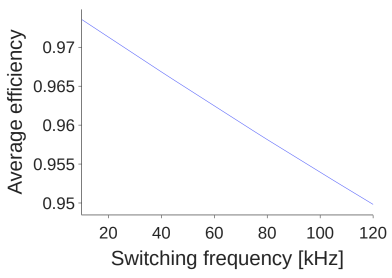


Figure 3.2: Effects of the variation of the number of devices.

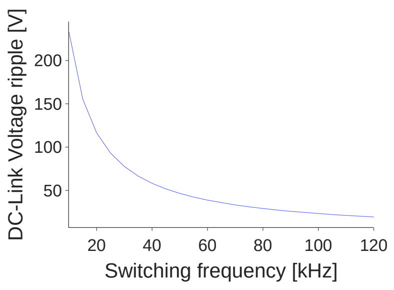
effects are shown in Figure 3.3. As expected an increase in this parameter has large impacts on both input voltage ripple and output current THD, with some diminishing-returns at higher frequencies. Maximum losses, and efficiency all show a linear increase proportional to frequency due to the growth of the switching losses component. The maximum allowable heatsink temperature to maintain an acceptable junction temperature on the other hand is inversely proportional to the switching frequency, as expected, since the temperature dropped across the multiple thermal interfaces between coolant and junction increases with the growing switching losses. Last but



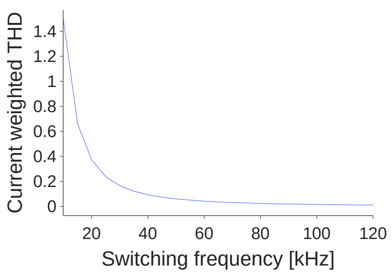
(a)



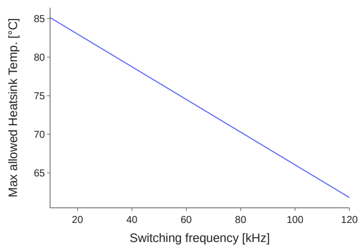
(b)



(c)



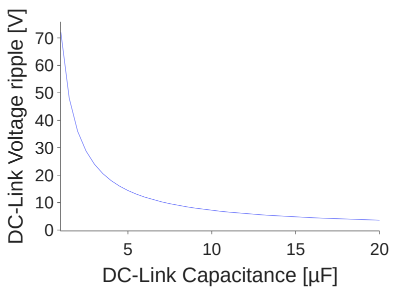
(d)



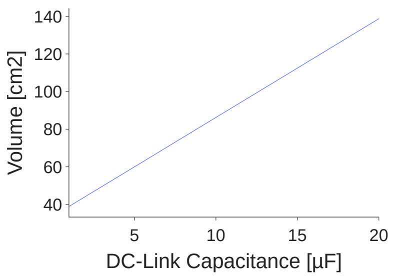
(e)

Figure 3.3: Effects of the variation of the switching frequency.

not the least cost and volume are not affected, given the choice of considering DC-Link capacitance a separate variable.



(a)



(b)

Figure 3.4: Effects of the variation of the DC-Link capacitance.

The amount of DC-Link capacitance used, has linear effects on both input voltage ripple and volume, as shown in Figure 3.4, while it does not affect the cost or losses related performance indicators. In regard to the choice of the transistor itself, among the potential candidates, the analysis has not shown any meaningful results, and it is consequently not added to this section. This is both due to the limited sample size of devices considered in the study, the arbitrariness of the device characteristics with respect to externally observable characteristics and the black box modelling approach that needs to be used given the lack of detail on the internal device structure and operation in the manufacturer published data-sheets.

3.2 Hardware design

The detailed design process starts with the results from the hardware optimization analysis, shown in Table 3.3. These not only define which MOSFET to use, but also the number of them needed in parallel to reach the desired power handling capability, as well as the target operating frequency and the minimum DC-link capacitance. The diagram in Fig. 3.5, shows a high level overview of the hardware structure with the main half bridge power section constituted by two switch clusters, each one served by a single high current gate driver. The control logic section oversees the communication to and from the central controller and of gate signal generation. All the voltages required for the power cell operation are generated from a single 15V DC input by two power supply sections, a general one and another isolated to serve the high side.

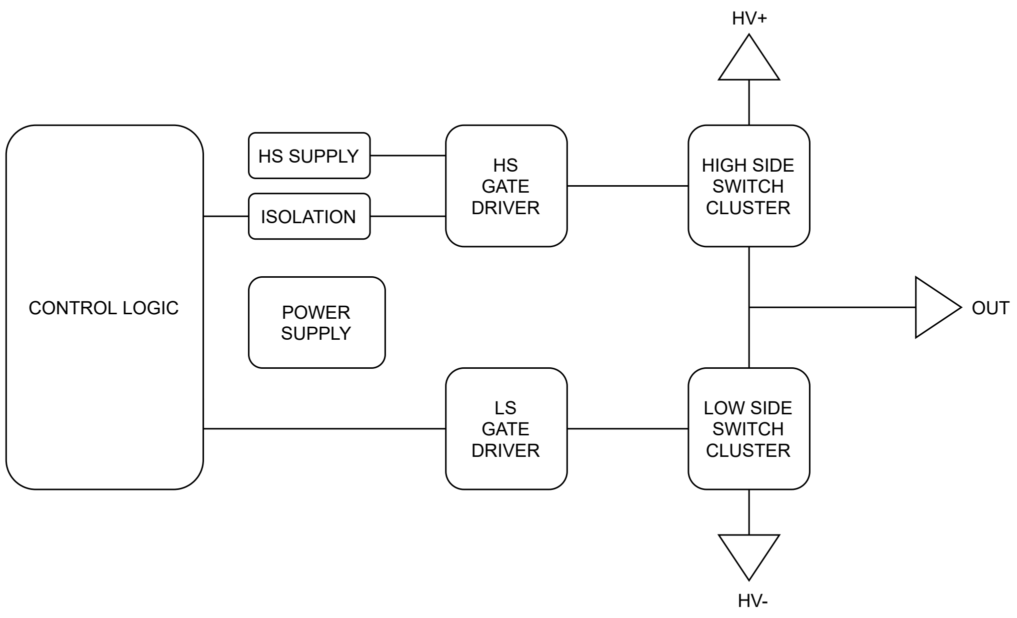


Figure 3.5: High Level Hardware Overview

3.2.1 Switch clusters

The switching clusters, whose schematic is shown in Figure 3.6, are composed of MOSFETs directly paralleled with each other. A dedicated gate resistor is used to enhance equal gate current sharing, helping to avoid situations where, due to the natural spread of parameters, the fastest devices in the group start conducting too far ahead of the others, inducing a current crowding effect that can lead to accelerated ageing and premature failure. Space has been allocated at design time to allow for the addition of per transistor RC snubbers and Zener diode overvoltage protection, if needed during the initial bring-up phase. Probing support was also considered at the design stage with the addition of an MMCX connector directly to the common gate signal, allowing for a minimal impedance connection with the probe head.

3.2.2 Gate Drivers

While the use of an integrated gate driver has been considered, the choice fell upon a discrete one, as shown in figure 3.7, to allow future research on active

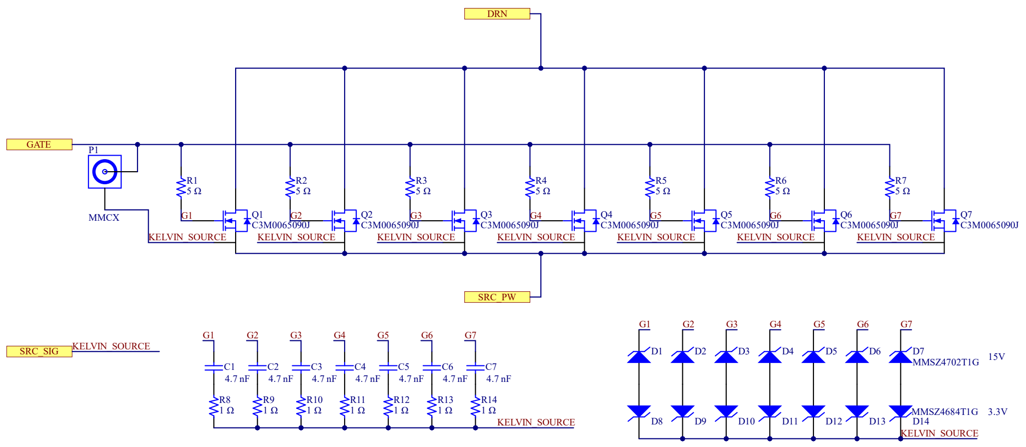


Figure 3.6: Switching Cluster Schematic

gate driving techniques. Working from the output backwards, the couple of BJTs Q12 and Q13 form a push-pull stage that boosts the current to be able to drive the whole cluster. These transistors, along with the relative bypass capacitors form the high frequency gate loop whose impedance should be minimized. The three level waveform required for active driving is generated through Q8, Q9 and Q15 with the last one acting as pull down and the former two as pull up to two different voltage levels, allowing the transistor to be either driven with two different current levels, while commutating, or even turning it on in different points in its characteristic. Bipolar devices have been used for the top spots as their lower capacitance makes them more suitable to this role. A two transistors level shifting arrangement formed by Q14 and Q16 is used to translate the logic level turn off signal, as the gate driver is operating with a slight negative bias, to be able to fully turn off the Silicon Carbide main devices.

3.2.3 Control Logic And Isolation

The control logic section consists of a single Field Programmable Gate Array (FPGA), mounted on a daughterboard it communicates with the main controller and generates the six signals needed to operate the power

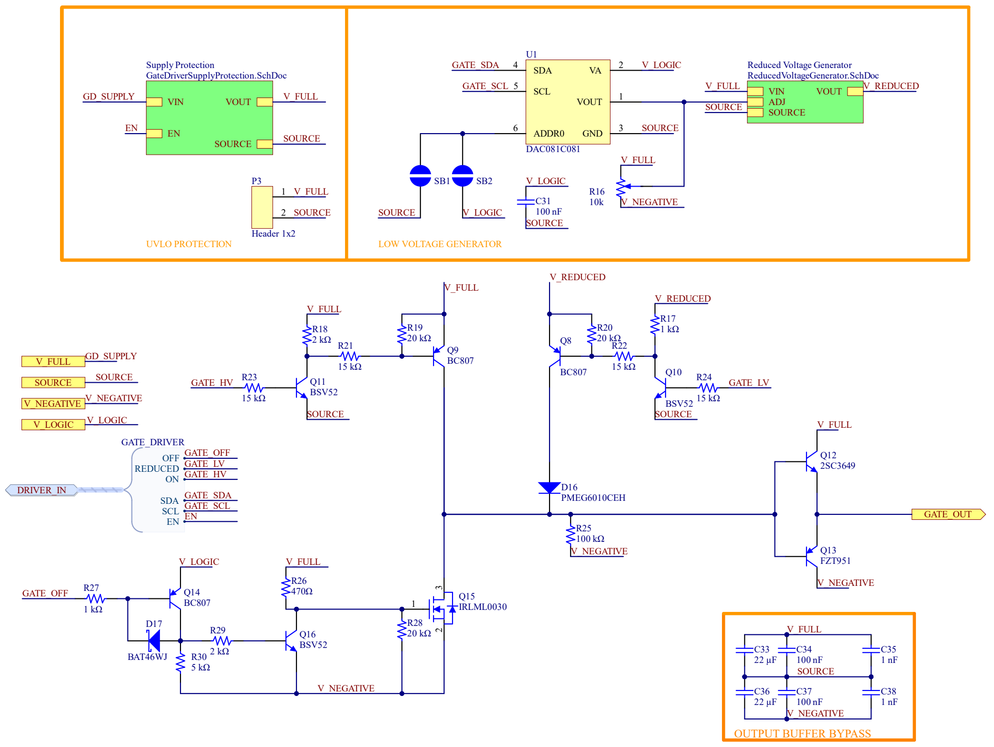


Figure 3.7: Gate Driver Schematic

section. The only external logic circuits present are buffers used to isolate the delicate FPGA I/O circuits from everything else. While the low side gate signals can be directly connected to the gate driver, this is not possible with the high side, whose gate signals must be referenced to the outputs. Optocouplers, shown in figure 3.8, are used to pass the signals from one domain to the other while keeping the two circuits isolated.

3.2.4 Power supplies

Finally, the power supply section, shown in figure 3.9, is constituted by multiple DC/DC converters that produce all voltage rails needed throughout the design. The main 15 V gate driving supply is connected directly to the input on the low side, while on the high side an DC/DC supply module is used to provide the required isolation. In addition to this a −2.5 V negative bias supply is needed to achieve complete turn-off in the main switching

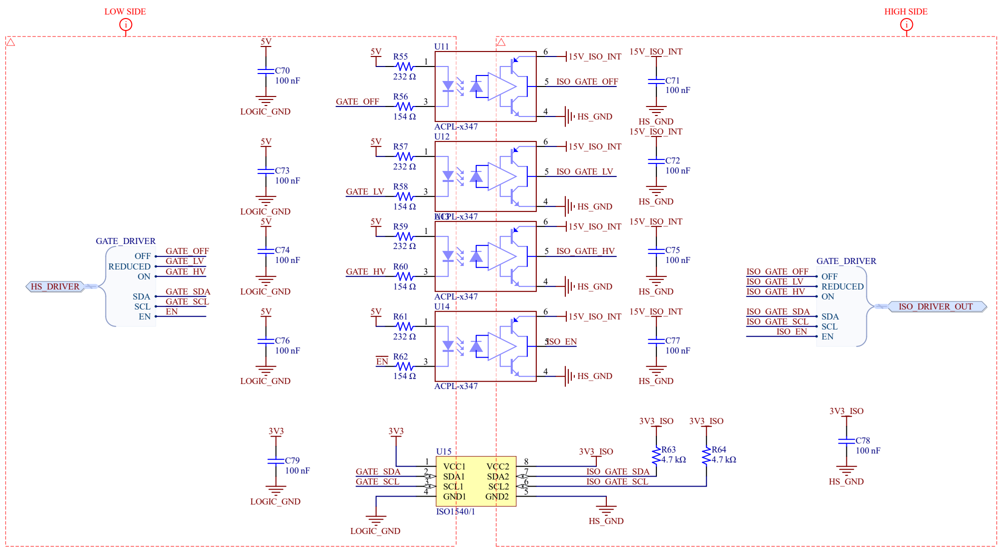


Figure 3.8: High Side Isolation Schematic

devices. This is derived from the main supply through an inverting buck-boost converter. Another minor rail required by the gate drivers is the adjustable voltage one required by the active gate driving features, which is also derived from the 15 V rail through a simple buck converter, controlled by injecting current directly on the high impedance feedback node by a DAC converter through a high value resistor. Lastly a 3.3 V logic supply is generated through another buck converter for the logic part.

3.2.5 Circuit Board layout

The physical circuit layout and routing, shown in Fig. 3.10 has been studied in detail, since for high frequency wide bandwidth converters is as important as the circuit design itself. The parasitics introduced in this stage, can easily compromise an otherwise working design, consequently a lot of care has been taken in their minimization. A C shape disposition of the seven paralleled devices in each cluster is adopted to ensure equidistance from the gate driver output, reducing the chance of asynchronous commutation between devices. The area of the main commutation loop was reduced as much as possible by

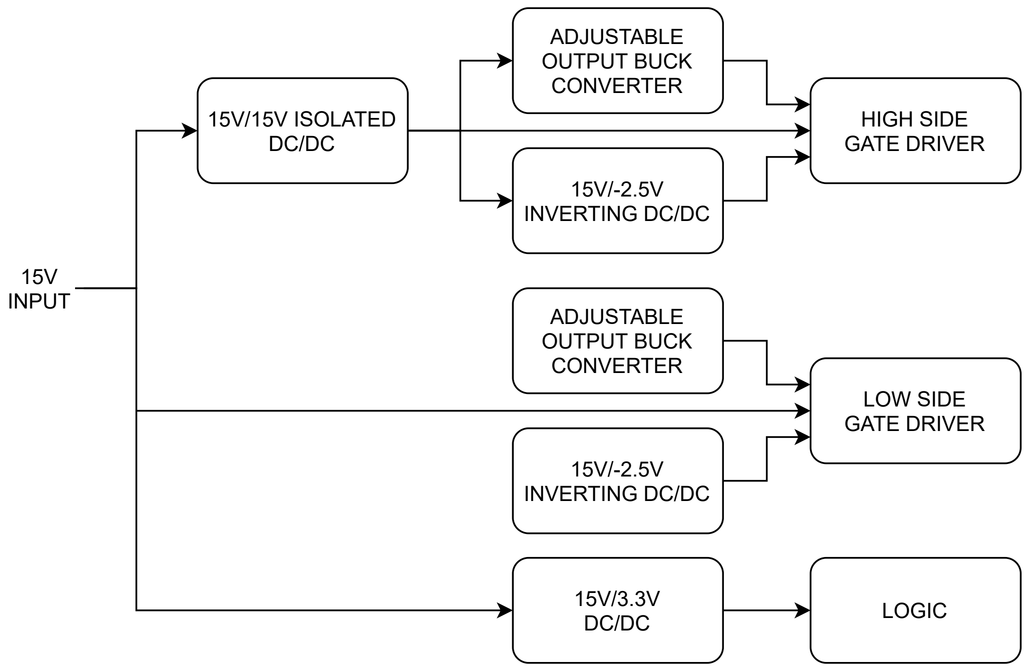
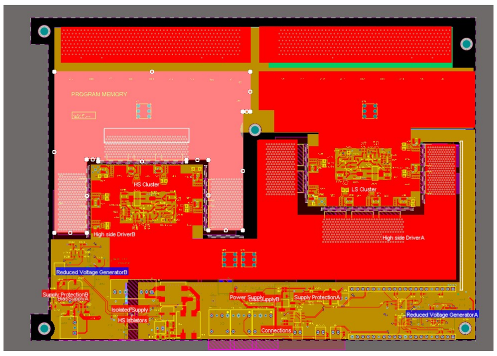
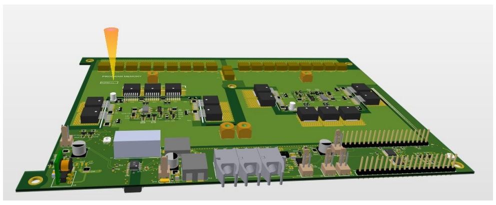

ADJUSTABLE OUTPUT BUCK CONVERTER
15V/15V ISOLATED DC/DC
HIGH SIDE GATE DRIVER
15V/-2.5V INVERTING DC/DC
15V INPUT
ADJUSTABLE OUTPUT BUCK CONVERTER
LOW SIDE GATE DRIVER
15V/-2.5V INVERTING DC/DC
15V/3.3V DC/DC
LOGIC

Figure 3.9: Power Supply system overview

the proximity of DC-Link and bypass capacitors with the devices terminals. Distributed capacitance is also introduced using interleaved negative and positive high voltage polygons on all six layers of the circuit board to limit as much as possible the conducted EMI emissions on the DC side.



(a)



(b)

Figure 3.10: PCB layout (a) and 3D render (b) of the per phase power hardware.

Chapter 4

FPGA control Platform Architecture

4.1 Introduction

Two classes of technologies are available when it comes to implementation of modern digital control systems: software based controls and FPGA based systems. In software based control, microcontrollers (MCU) or microprocessors (MPU) are used to calculate the required control laws, and manage the complete path from analogue input conversion to PWM output generation thanks to dedicated peripherals. In Field Programmable Gate Array based systems, the whole control system is composed of discrete synchronous logic circuits, that implement directly both arithmetical and Input/Output (IO) functions. Between the two, MCU based systems are usually preferred, due to the simpler software development process and more mature toolset with respect to the typical workflow of Hardware Description Languages (HDL) used with FPGA platform. The presence in microcontrollers of on-die, specifically designed, fixed function peripherals like PWM modulators, ADC converters, etc. further simplifies system integration. The purely software based approach however has also some notable downsides.

- **Limited scalability:** Monolithic MCU based systems are limited by the amount of available on-die resources (PWM channels, ADC channels, etc.), while networking multiple independent units significantly drives up complexity.
- **Limited performance:** The serial nature of software execution in processors, coupled with the strong determinism constraints, which preclude the use of many performance enhancing techniques, like caching, speculative and out of order execution, limit the maximum available computational resources.

FPGA based systems address these two limits of microcontroller based systems by providing a "Silicon blank slate", upon which the designer can build a fully custom infrastructure, that can be easily scaled up or down in a project-by-project basis. The fully parallel nature of these devices allows better performance scaling by allowing fully independent processing of parallel tasks. The last advantage that made FPGA an almost mandatory choice is their better suitability to the implementation of custom communication protocols, the design and performance of which will have a large impact on the overall system behaviour. It should also be noted that hardware logic based implementations, are preferred in mission critical systems, as both their behaviour and their fault conditions are simpler to evaluate and more predictable with respect to software ones.

The proposed system architecture is constituted by three separate layers, as shown in Fig. 4.1, each one responsible for a subset of functionalities, starting from the top one can find the HMI layer, that translates the user's commands to a form that can be easily acted upon by the underlying layers. The management layer offloads the most common tasks like configuring

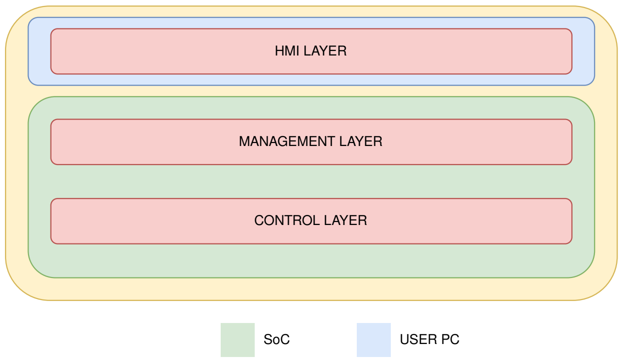


Figure 4.1: System Layers

the FPGA with the correct bitstream and programming the processing core, along with being in charge of storing all user and application specific information. The lowest layer, the control layer, is the one where all the hard real-time control functions are implemented.

The HMI is running directly on the user PC, while all other layers, that form the backend server to the Application, are implemented in a Xilinx Zynq (xc7z020) System on Chip (SoC) that is comprised by a programmable logic (PL) part, which contains an FPGA fabric, and a Dual core ARM cortex A9 processor system, with relative peripherals.

4.2 HMI layer

The HMI layer, whose structure is shown in Fig. 4.2, while seemingly unimportant, is critical for the smooth operation of any control system, as it allows the presentation of information to the user in a meaningful way, and acts as a first layer of translation between user inputs and machine parameters. It is important for this layer to be as flexible and dynamically adaptable as possible, to avoid the necessity of application specific changes to the lower layer, as the complexity of development sharply increases the closer we move to the real time control portion.

Another requirement for a well-designed HMI component is the ease

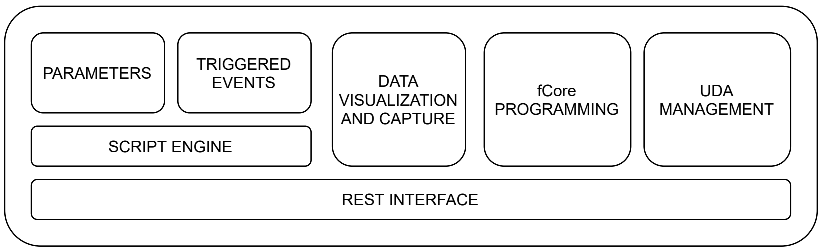
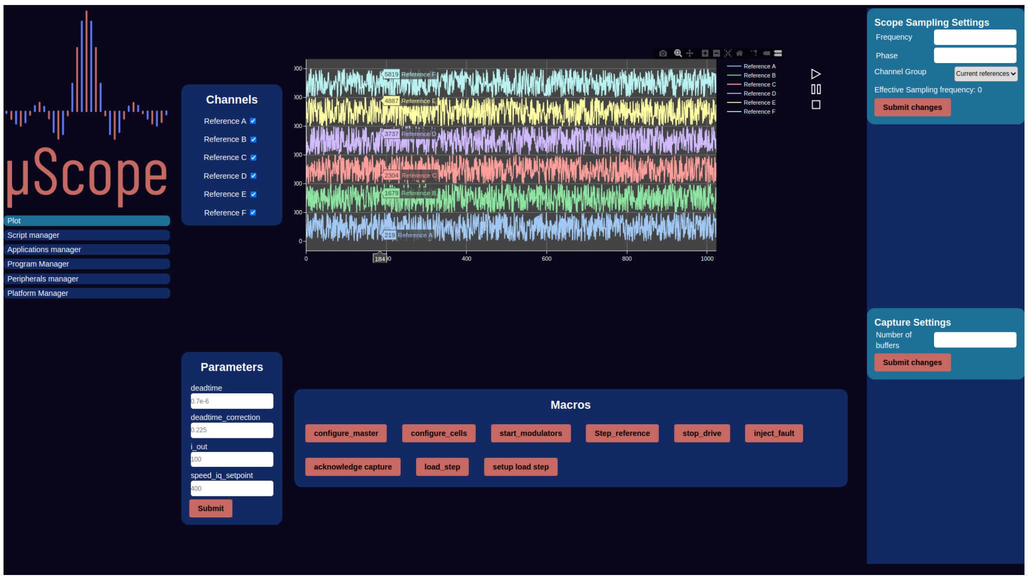


Figure 4.2: HMI Layer structure



Parameters

deadtime: 0.7e-6
deadtime_correction: 0.225
i_out: 100
speed_iq_setpoint: 400

Figure 4.3: Human Machine Interface example

of deployment especially in multi-user scenarios. For this reason, a web application, was chosen as the technological platform, leveraging TCP/IP connectivity to link it to the rest of the platform. The use of a standard network connection allows leveraging the OS native software stack, avoiding the development of specific device drivers, needed by all major operating systems for high performance and reliable operation.

The use of a standard web browser to run the application, shown in Fig. 4.3, allows plug and play operation without the need for system-wide software installation. Other advantages given by this system architecture are the possibility of completely wireless operation, through ubiquitous Wi-Fi connectivity for complete electrical safety, and possibly even remote management through an internet connection.

4.2.1 User Defined Applications

Given the extreme flexibility of the underlying platform, as peripherals and IP cores can be easily added and removed from the FPGA fabric dynamically a fixed function user interface is undesirable, as it would prove either too general, exposing too many low level details, or excessively specific with respect to the current system, and consequently incapable of adapting to different requirements.

To solve this issue the concept of user defined applications is introduced. The system designer, along with the HDL designers can specify a set of peripherals, register, data channels, events and parameters that make sense for the specific use case and associated FPGA bitstream, along with the scripts needed to translate them in a form that the lower layers can understand.

- **Peripherals:** Peripheral definitions stand at the heart of the HMI, they match the IP block present in the FPGA fabric and translate the list of registers they expose to more meaningful and user-friendly names that simplify script writing.

- **Registers:** These values represent static, application specific, register configurations needed for correct functionality. They are configured only once at startup thus are not directly user accessible.

- **Data channels:** These define which the data streams that are collected inside the FPGA fabric and can be exported through the integrated scope.

- **Channel groups:** These select which of the available data channel are combined to be visualized or captured

- **Parameters:** Parameters are user defined floating point values that can be modified while the system is running to affect its behaviour, they are translated to machine understandable values through scripts

- **Events:** The Events are user activated triggers to scripts. They can be used in conjunction with parameters when several of them must be modified in an atomic manner

- **Scripts:** Scripts are a powerful tool that allows the user to translate meaningful and user-friendly parameters to machine understandable register values. They must contain a single JavaScript function.

This information is compiled in a self-contained dictionary that is managed by the lower layers. Upon startup, the user will choose the desired application, allowing the UI to be configured to show the correct information. To allow the creation and management of user applications, along with all their components, several management pages are also provided that enable quick and easy creation, update and removal of all the aforementioned components. Extensive importing and exporting options are also available for easy migration of all data between platforms.

4.2.2 Parameters, user events and Scripting

These three features combine to allow the user to effectively control the behaviour of the system. Parameters and user events are used to trigger the execution of Scripts by the browser scripting engine. A context is passed into each script as argument and contains the value of all parameters, along with a workspace in which variables can be written and read from, be used to share information among different scripts. A list of writes to platform registers can be optionally output by the script to be performed by the management layer. To ease script development an integrated editor is included allowing their creation and editing directly in the application.

4.2.3 Data Visualization and capture

Another extremely important component of a functional HMI interface for power converters is a data visualization system that can show the user sensor reading, control variables and other interesting data points. This allows control system tuning, as well as catering to any other operating requirement. Due to a practical limitation in the lower layers only a selected number of data channels can be captured, and shown at once (six in the current implementation). Since many more data points are available in a typical application, the concept of channel group is introduced, allowing the user to quickly select which set of data points to assign to the data channels. From a technical point of view the data is autonomously captured by the control layer and saved to a buffer. The UI layer periodically polls this and then shows the data in a live refreshing real time chart plot. For smooth operation, its frame rate has been capped at 10 fps, as faster refresh results in a worse experience due to frame rate inconsistency.

To aid in the capture of fast transient events such as load steps, a hardware assisted data capture mechanism is implemented. In this mode the scope hardware components capture a programmable amount of pre-trigger data, and then outputs a trigger pulse that allows the rest of the logic to react accordingly. Once enough data is captured to completely fill the buffer, the sampling is suspended, while the resulting data is transferred to the HMI layer to be exported as CSV.

4.2.4 fCore Programming

Last function of this HMI layer is the support for programming of the fCore embedded DSP, to this effect a program management UI is provided, it allows the creation, editing, with the integrated editor and removal of programs. Their compilation is carried out by the management layer, whose status

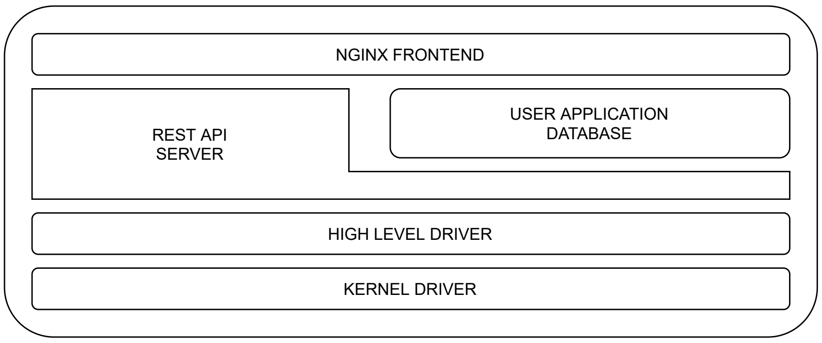


Figure 4.4: Management Layer structure

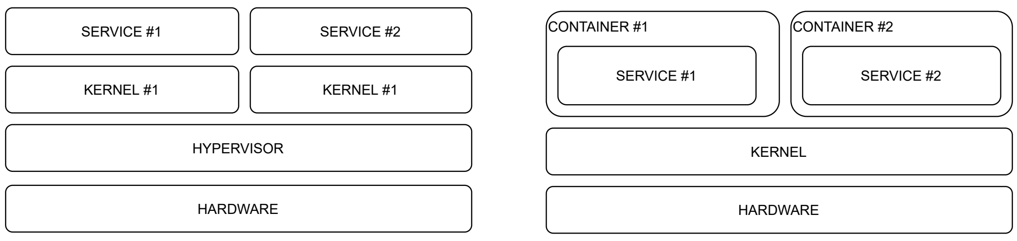
is directly reported to the user. Program loading can be performed both automatically at startup and manually at runtime.

4.3 Management layer

The main purpose of this layer is to abstract away the complexity of the hardware platform, providing a stable interface that the HMI layer can build upon. From an implementation perspective, this being a pure software component, a standard Linux system can deliver better security, thanks to a much more audited code base, ease of use, with a standard compliant, secure and fully featured networking stack and access to a range of other technologies.

4.3.1 Layer partitioning and containerization

To perform the large number of diverse and wide-ranging tasks assigned to this layer, it was partitioned into several independent sections. Advantages of this architecture are many from the better long term maintainability, with respect to a single monolithic software component, to the faster development time. The most important benefit deriving from this separation is the establishment of a clear interface between the internet connected, and consequently untrustworthy and the rest of the system. The presence



(a) (b)

Figure 4.5: Virtualization (a) and Containerization (b) diagrams

of clear boundaries simplifies the implementation of secure authentication mechanisms, firewalling and strong encryption. Enhancing security through a layered defence, where multiple independent protections need to be breached to gain control of the core system.

To take full advantage of the previously shown benefits logical separation of task is required but not sufficient, as enforcement of the established boundaries is crucial to mount an effective defence. The first method, involves, as shown in Fig. 4.5a running each service in a virtual machine, abstracted by the physical hardware, this provides the maximum possible security as neither memory nor processing is shared, but rather mediated by a hypervisor. This very strict division does impact performance, as the share-nothing approach forces communication between different services through the full network stack. Containerization, the chosen technique, is a complementary approach, that addresses the performance issue, while still retaining most of the isolation benefits available through virtualization. It leverages specific protection mechanisms offered by the operating system kernel that can be used to control the resources allocated and shared by different sets of processes, effectively partitioning the user space components in several isolated containers, as shown in Fig4.5b that communicate only through a well-defined and more performant local network.

In particular the management layer is constituted by four separate containers that communicate through standard TCP/IP protocols, it should be however noted that using a single kernel greatly decreases the amount of processing needed in the networking stack, decreasing IO latency. The services implemented by these are respectively:

- Frontend
- REST API server
- Database
- High Level driver

4.3.2 Frontend

This is the first Component in the management layer, it is composed by a single Nginx server that provides the full interface between this layer and the HMI running on the user device. It not only serves the static parts of the application like images, scripts and HTML files, but it also acts as a reverse proxy for the REST interface whose endpoints are implemented downstream. This allows the user application to access all the management layer functionality through a single URL, strictly complying with the same-origin policy enforced by all major browsers that allows only a single origin for all requests made by a web page. Another benefit that the use of this component brings is the possibility to use a simple and cheap single domain TLS certificate to enable the use of a secure network connection in the link with the client, encrypting all the traffic and preventing man in the middle attacks. This, while not needed for completely local and trusted network, it is mandatory for remote operation over the public internet. TLS termination at this level can be considered secure as all other components of the system are implemented on the same hardware and communicate only on a separate trusted virtual network.

4.3.3 REST API server and Database

This component communicates to the user through a REST interface, authenticated using JSON Web Tokens (JWT). It allows persistence and self-contained operation by managing the application database and passes the hardware access requests through to the driver. The server has been implemented as a custom python application, using the Flask micro-framework. Several classes allow the creation, retrieval, update and deletion of all components of the user defined applications. Compilation of femtoCore programs is also handled at this level, a dynamic library loaded through a custom python module, is used to compile and assemble the user specified program content. Once the operation is completed the resulting binary, an array of 32 bit words, can be either cached in the database for later use or directly passed to the lower layers. This component is also responsible for loading the correct bitstream file to the FPGA portion of the SoC through standard operating system interfaces.

4.3.4 High level and Kernel drivers

The driver is the component responsible for the low level management of the control layer. It is split into two parts, a kernel portion and a user-space one. The high level driver, implemented as a regular user space C++ application, running in its own docker container, communicates with the server through a TCP socket acting as a broker between the control layer and the rest of the stack. It reads and writes to memory mapped registers on the FPGA, and programs the femtoCore processors present in the fabric. At this level, the data captured by the scope is also sorted into several arrays, one for each channel, before sending it forward to the server ready for visualization.

A second part of the driver component has been implemented as a loadable Kernel module, it exposes the two general purpose AXI interfaces

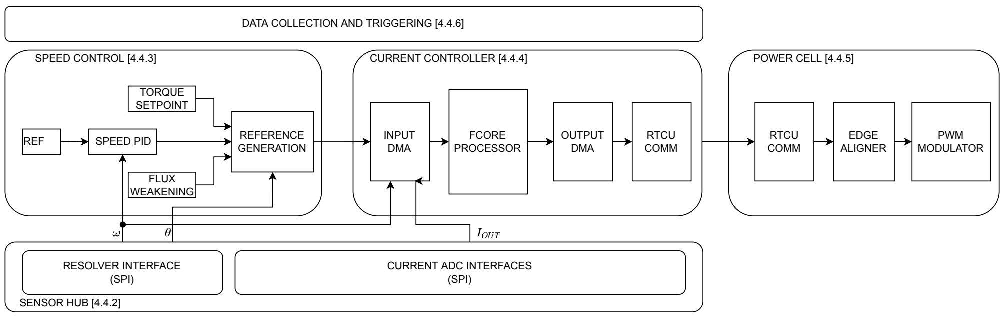


Figure 4.6: Control Layer Structure

to the FPGA, through which the kernel communicates, to the user-space components, and handles the low level scope captured data, moving them from the hardware managed Direct Memory Access (DMA) buffer to a separate holding one ready to be sent to the HMI.

4.4 Control layer

This is the lowest layer in the whole stack and is responsible for all the hard real-time aspects of the system, like control and protection related functions. From a high level perspective, the whole architecture can be broken down in five functional units, as shown in Fig. 4.6.

4.4.1 Control BUS structure

The Processing System (PS) and all the developed IP blocks in the Programmable Logic (PL) section of the system on chip, are linked together

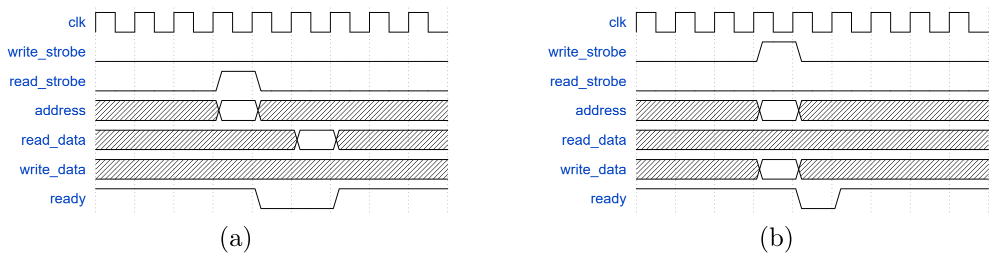


Figure 4.7: Timing diagram of a Read (a) and write (b) Control bus transaction

through an extensive bus structure, not shown in the previous diagram for clarity. It allows control and run time configuration of the whole control layer through general purpose software programming techniques, increasing flexibility of the complete system. The cited use cases determine the main set of requirements to be used in the bus protocol selection, which are the minimization of FPGA resource usage and implementation simplicity. These, coupled with the low bandwidth needed for configuration data, dictated the use of a simple shared bus design. Derived from Intel Avalon Memory Mapped bus, the utilized protocol uses separate address and data busses, with read and write strobe lines signalling both data validity and transmission direction, as shown in the timing diagrams in Fig. 4.7.

While data flow is bidirectional, only the master endpoints can initiate a bus transaction, an active low ready signal, can be used by busy slaves to delay further communications. To enable multi-master operation, fixed priority arbitration is implemented in multiport switches. The control bus is connected through to the PS AXI bus using AMBA APB as an intermediary protocol, with a custom combinatorial translation layer on one side and Xilinx IP on the other. The only peripheral directly connected to the processing system bus is the femtoCore processor instruction memory, which is directly mapped in the control software address space, both simplifying and speeding up upload of the programs to be run.

4.4.2 Sensor HUB

The sensor HUB contains logic needed to interface with external sensors and related signal processing. Both position and current sensors can be communicated to through a simple SPI interface. The communication strategies and operating protocols of the two designs however are different and should thus be treated differently. The resolver sensor is handled completely externally through the Analog Devices AD2S1210 resolver to digital IC. It takes care of both excitation signal generation and output signal decoding and digitization, through two hardware tracking loops that keep track of both phase and angular speed of the rotating shaft. To read both values a standard single channel SPI interface is used along with some simple logic to handle the resolution and signal selection. On the Fabric side an AXI stream interface is used to pass the data onwards without the need for further processing. For the currents, sensor signals are digitized through a bank of LTC2313-14 analogue to digital converters. A single multichannel SPI peripheral is used for all, since synchronous sampling is desired, as well as to reduce the logic usage. Here a common Finite State Machine (FSM) and set of registers control a bank of 6 parallel shift registers. Once the raw samples have been latched in, they are passed on through AXI stream interfaces to the ADC post-processing blocks that perform basic offset calibration and average the current signals down to the desired frequency to be fed to the controllers. Additionally, the raw samples are collected for capture and visualization.

The last function implemented here is the fault monitoring, a bank of comparators is used for detection of over-range on both the fast and decimated data stream. This allows gross overloads to be handled as quickly as possible, while allowing to monitor and more aggressively limit the cleaner controller input signal.

4.4.3 Speed control

This control section is responsible for the speed regulation of the machine when running in constant speed mode, as well as the sinusoidal reference generation. To balance versatility, logic utilization and complexity a fixed point integer implementation of a PID controller has been chosen. A shift and multiply approach enables a resource efficient implementation of fractional controller gains, at the price of discrete power of two denominator values. Even if not strictly necessary for this application, the implementation has been fully pipelined, to allow single clock cycle operation. Since the current control algorithm chosen needs sinusoidal references, a multichannel DDS signal generator has been used, allowing the generation of an arbitrary number of sinusoids. At the core of its implementation, a look-up table is used to hold the pre-computed values of the sine function in the first quadrant, through addressing and trigonometric identities the value of both sine and cosine in their whole domain can be computed effectively reducing the RAM required for the table by a factor 4.

4.4.4 Current control

The central function for any machine drive, be it traditional or distributed, is the current control section, which is responsible for the lowest level of operation of the system. Given the large ratio between hardware clock and switching frequency, a processor based implementation was chosen for this function, to lower resource usage. To insert and retrieve data from the core register space two distinct DMA engines are used for this task. A custom designed core is used to guarantee a completely deterministic execution. A more detailed presentation of its design and implementation will be discussed in chapter 5. Once the duty cycle for all power cells have been calculated by the core and extracted through DMA they are sent to each power cell through the custom RTCU protocol detailed in Section 4.5.

4.4.5 Power cell

In the Power cell, after reception of a duty cycle message, by the communication logic, the value is passed to the Edge aligner module where it is saturated to protect the power hardware, avoiding dangerous shoot through conditions, and then used to calculate the updated modulator register values. This module is also capable of inserting an independent amount of dead-time for both transitions, allowing the compensation for highly asymmetric gate driver components. The last component in the chain is the PWM modulator itself. This highly configurable PWM block can generate very complex PWM pattern thanks to multiple carriers, with programmable time shifts each of which can drive a parametrized number of output channels with two comparators each, automatic generation of complementary signals is also available if needed. Shadow loading is also implemented to allow easy and safe register update without the need for synchronization.

4.4.6 Data capture

To allow visibility in the control logic, needed to both tune, operate and monitor the system. Data capture points have been introduced, tapping off AXI stream interfaces throughout the whole design. These converge to a bank of multiplexers that select six of these channels. The data is then sampled at a user selectable frequency, channel tagged, and then held temporarily in an inline FIFO. Whose content, once full is transferred automatically to a buffer in main memory. An interrupt is used upon DMA transfer completion to trigger the higher layers processing. To easily allow the execution of fast transient measurement an output trigger facility is also provided, where a pulse is sent when a configurable FIFO fill level is reached. Upon completion

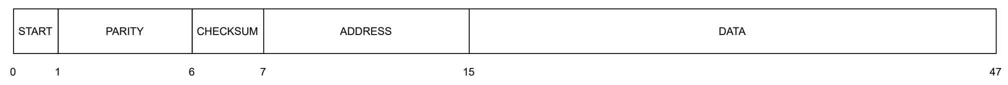


Figure 4.8: Developed protocol message

of the acquisition capture is halted temporarily to allow the higher level layers to download the data.

4.5 Communication Protocol

From a logical perspective each message in the developed protocol consists of a fixed structure packet, shown in Fig. 4.8, composed of a header containing a start bit and error correction information with both parity for the data and a global checksum. These are followed by a 40 bit payload field, that is conventionally partitioned in a single byte address denoting the message purpose and a 4 byte message data field.

The small packet size, while somewhat hurting efficiency keeps latency at a minimum, while still allowing a complete 32 bit word transfer with each message. To guarantee reception an optional acknowledge can be sent back to the sender, allowing the assessment of link health. A periodic automatic heartbeat message is sent by a watchdog timer to guarantee an upper bound in the fault detection time. To decrease its impact on channel capacity and transmission latency the counter is reset after a successful message exchange, making this aspect completely transparent during nominal use.

The overall structure of the transmission and reception chains are shown in Fi. 4.9. A transmission starts with multiplicative scrambling of the message payload, that is feed to the error correction encoder which adds parity and checksum information. A serializer is then used to push the data over the channel. Upon reception the packet will go through the inverse process, with a de-serializing, decoding and de-scrambling. Resulting in the

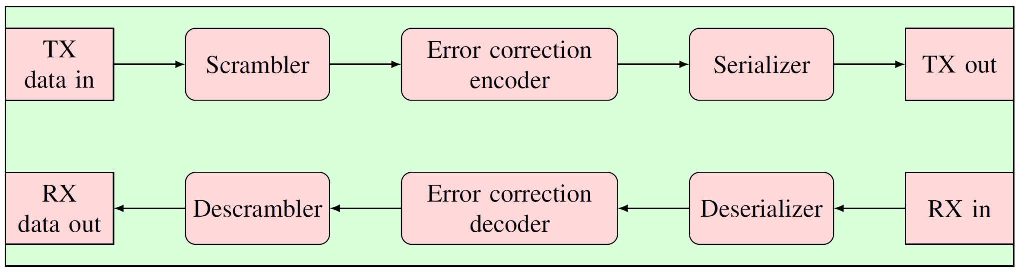


Figure 4.9: Diagram of the overall communication chain

original message being passed to the downstream logic.

4.5.1 Synchronization

Two separate solutions are available to achieve clock synchronization between nodes, the use of a clock distribution network and clock recovery. For small systems with few nodes, the first choice is used and preferred, as costs are usually comparable with the latter while allowing for a much lower complexity receiver. In this case a clock signal is generated in the central controller and passed either electrically or optically to all the other nodes in the system. The narrow pass-band characteristic of the signal, coupled with the known and stable frequency allows the use of very high quality factor filters, to reject EM interference even in very noisy environments. When this technique is chosen, the scrambling/descrambling steps can be bypassed as they are not strictly needed. As the size of the system grows, there will be a point where the cost and complexity of distributing a clock signal to dozens or even hundreds of nodes become too large. In these cases, the clock can be recovered directly from the incoming bit stream, provided a large enough density of transitions is present. In this case a hardware Phase Locked Loop (PLL) is locked to the incoming data stream, thus recovering the transmitter clock. The two techniques can even be present in a single system, to better adapt to the physical characteristics of each channel. For a hierarchical tree topology network, where the main controller is linked only with tier two routers, which are in turn connected to the power nodes, explicit clock distribution can be used for the small number of high speed links while clock recovery is used for the other role or vice versa.

4.5.2 Error Handling

Error handling is a very important area of a communication protocol design that can greatly impact system performance. The traditional approach of error detection and message retransmission can deal with numerous random errors while having minimal impact on the transmission efficiency during nominal operation. The downside of this process is the introduction of a large amount of jitter when a packet needs to be retransmitted, potentially leading to a deadline violation. An alternative strategy, used in the designed protocol, is the use of Forward Error Correction (FEC), where enough redundancy is added to the bit stream to enable the receiver to mathematically reconstruct the original bit pattern, completely avoiding the need for retransmission. The main downside of these techniques is the limited number of correctable bits, requiring the designer to evaluate the error probability on the expected channel, allowing the addition of enough redundancy to reduce the probability of an undetected and uncorrected error to acceptable levels. To address this weakness, two error correction techniques can be chosen for the developed protocol, depending on the expected bit error probability. A Hamming single error correction, double error detection (SECDED) code is used for reliable channels, where the probability of more than two errors is small. While a Reed-Solomon code can be used (RS(15,11)) capable of correcting up to 2 symbols, with each symbol containing 4 consecutive bits can be used when operating on a worse channel.

Chapter 5

femtoCore eDSP

5.1 Why femtoCore?

The implementation of control systems on FPGA can take two routes, fully custom logic, which can attain much higher operating frequencies, at the cost of a long and complex implementation and verification process, or serial, through processor or finite automata. This second choice allows the use of a much quicker software development model, as opposed to the HDL one, with the main drawbacks being longer cycle times and potentially a lack of determinism. While the frequency trade off does not impact the considered application as modern FPGA working frequencies are high enough to allow the calculations to take as many as few thousand cycles while still respecting the required deadlines. The issue of determinism is much more problematic for mission critical applications, as it brings about all the problems inherent in real-time low jitter safety critical software development, and the related certifications.

Upon close examination of the desired use case, it is apparent that the implementation of most control systems, can be reduced to the solution of one or more equations, and provided that all required data is made available by

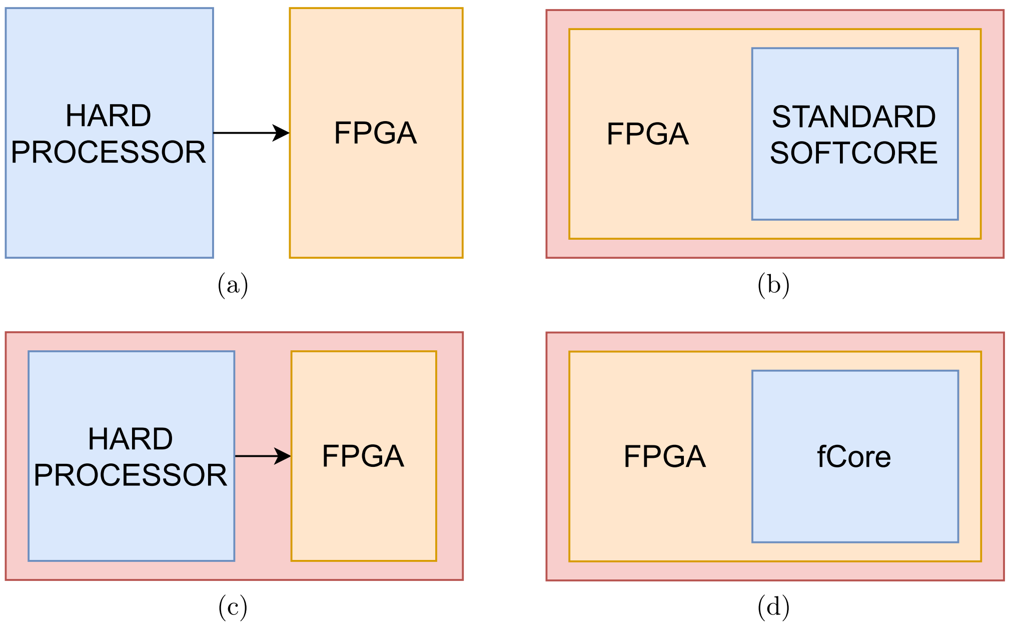


Figure 5.1: Architecture considered in the comparison: a) Separate FPGA and DSP, b) SoC with Hard processor and FPGA, c) FPGA with soft-core processor, d) architecture with propose core

the rest of the system, the processing core's only responsibility is to perform a series of arithmetic and logic operations in order to obtain the required control action. To take advantage of this characteristic of the application, a custom instruction set (ISA) and processor core has been specifically designed in order to allow the implementation of the control system calculations as software, while retaining a fully deterministic execution.

It is important to highlight the differences between the use of the proposed processing core and some of the more traditional processor-FPGA hybrid architectures, shown in Fig. 5.1. The first, in Fig. 5.1a, is the use of separate FPGA and processor, on the same circuit board connected together through the processor external bus interface. This solution is typically the least flexible and slowest of the four, the external interface severely limits the maximum frequency of the communication, the synchronization of the two elements can also be challenging when bidirectional information flow is required. Due to these limitations, when this solution is adopted, the programmable logic device is only used to implement the final modulation. Consequently, the control tasks are performed completely by the processor. As such the advantages and disadvantages for this solution are the same as for the traditional single MCU implementation. The second solution, in Fig. 5.1b, consists of integrating the processor directly inside the FPGA using a soft-core IP, these are available both from FPGA vendors (Xilinx Microblaze and Intel Nios II) or third parties (ARM cortex M1 and M3). The integration of both components on the same die allows much higher bandwidth between the two components and makes synchronization of the different parts of the design much simpler, allowing some degree of acceleration of computationally intensive tasks by the FPGA. The third architecture, in Fig. 5.1c, replaces the soft-core processor with a hard macro processor, realized in silicon on the same die of the FPGA, and connects them with a communication bus (typically AMBA AXI). This solution allows the use of a much more performing processor core, like the Cortex A9 running at several hundred megahertz on the Zynq. This change comes however with some stark disadvantages. While the soft-core processor can be configured to minimize the amount of execution time jitter, by removing caches, disabling branch prediction, where present and so on; the hard processor system on the SoCs is optimized for raw throughput, as opposed to latency or low jitter, forcing the adoption of large timing margins to compensate, negatively affecting the achievable performance. The last Architecture, using the femtoCore processor, shown in Fig. 5.1d is closely related to the second one, where the general purpose soft-core processor is replaced by the femtoCore processor. The main advantage of this arrangement is easy synchronization, since the core execution can be started with an external signal and the runtime is constant.

In table 5.1 it is shown a more detailed qualitative comparison of few

	A	B	C	D
Clock frequency	−	=	+	=
Determinism	−	−	−	+
Programming complexity	=	=	−	+
Logic Hardware complexity	+	+	−	−

Table 5.1: Comparison between FPGA-Processor solutions.

key features and metrics between the four solutions. The + symbol denotes a strength of the architecture, − signifies a weakness and = shows where the solution is average. In the clock frequency category the hard processor shows its definite advantage. Both soft-core and the proposed solution are average and the separate IC topology is considered weak, since while the individual components might be able separately to run at high frequencies, the communication bottleneck between them limits the overall system effectiveness. As shown in the previous section, only the proposed solution can claim a truly deterministic execution time, for the control algorithm that can be known in advance. In the category of programming complexity solution, C is weak, as the processors found in these type of systems are designed with compatibility with modern operating systems in mind, making them a lot more complex to use with respect to all other architectures that use microcontroller architectures specifically designed to be used in a bare metal context. The proposed solution on the other hand can be easily developed for since, for the targeted application, only a relatively short sequence of mathematical operations are required, as all other I/O and timing tasks are carried out by the custom logic outside the core. One downside shared by both solutions C and D is Logic hardware complexity, which is higher than the in the other two as solution C requires dealing with the data exchange between hard core and FPGA portion, while the proposed solution requires external logic to handle sensor sampling and data movement.

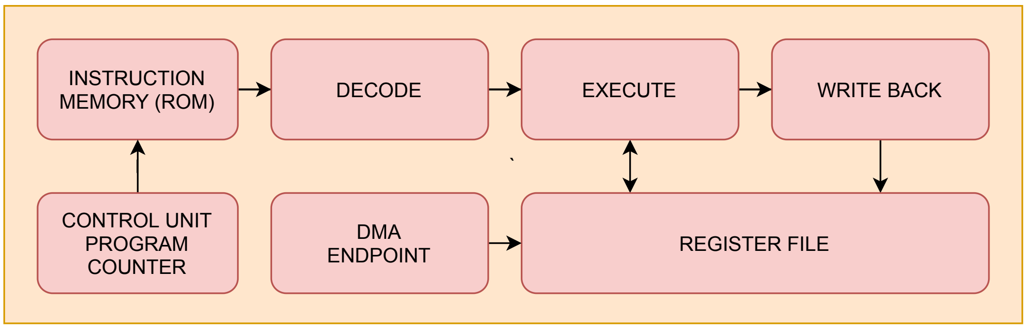


Figure 5.2: High level fCore architecture diagram

5.2 High Level Architecture

The high level architecture of the designed processor, shown in Fig. 5.2, it is very simple, yet somewhat unconventional, the base is a simple three stages pipeline with a decoder, an execution unit and finally result write-back, derived from the classic five stages RISC pipeline, eliminating the instruction fetch stage; all instructions have a fixed size, and the memory access, not needed in this architecture. A single pool of data memory is used, denominated Register file, with sixty-three 32-bit general purpose registers (r1 to r63) and a single zero register (r0), that holds the constant value of zero and is used internally to implement several virtual operations, such as register to register data movement and the no operation (also known as NOP). A control unit starts the execution upon reception of an external trigger signal and halts it when reaching either the stop instruction or the end of the program memory. This unit also contains the program counter that is advanced once for each execution cycle. A DMA endpoint is also present that allows the external logic to load the inputs directly in the register file, at the appropriate location, and to extract the results once the done signal is issued. Each one of the available operations, listed in Tab. 5.2, can be encoded in a single 32-bit wide instruction, except for the constant load which takes two. The most important feature of this core is the lack of any control flow instruction, such as conditional or unconditional jumps.

Table 5.2: Available operations

Arithmetic	Logic	Conversion	Saturation	miscellaneous
addition	and	From integer	positive	greater than
subtraction	or	To integer	negative	load constant
multiplication	not			less or equal to

This guarantees that the duration of each execution of a single program to be identical eliminating jitter by design, it allows also to easily evaluate ahead of time, during compilation, how long a specific program will take to complete execution, and as a consequence the maximum achievable operating frequency.

In order to reach a reasonable clock frequency, the execution of a single floating point operation is implemented as a five stage pipeline, that once filled, can process an instruction every clock cycle. It should be noted that in traditional design a long pipeline is undesirable since it might need to be flushed upon a jump or branch instruction, leading to uncertainty in the run time and decreasing throughput. In the femtoCore, the execution time is largely independent of the pipeline length, that, once full, will never be flushed, leading to a fully deterministic execution time.

The only minor downside from the multi cycle operation of the execution unit is the need for delay slots in the program to avoid data hazards when the next adjacent instructions are co-dependent. This addition can be performed either by the compiler/assembler, through a simple static analysis pass, or in hardware, with a small increase in front-end complexity.

When dealing with uncoupled multichannel systems, as in the intended application, the same program will need to be executed multiple times, once for each channel. To benefit this use case the core has support for automated SIMD execution. When this mode is enabled, the execution is interleaved, executing each instruction on every channel before advancing the program

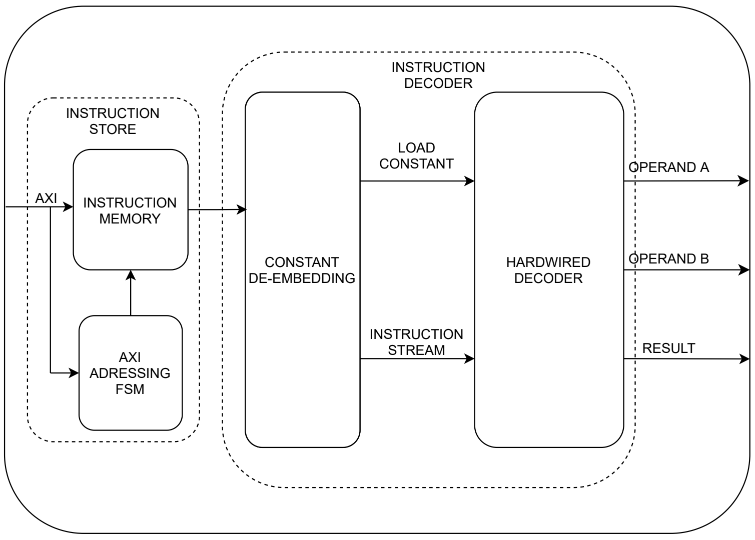


Figure 5.3: Frontend structure

counter, the Register file is also expanded to have a full complement of thirty-two registers for each channel. A channel counter helps to select the active partition of the full register file. When operating in this mode since no data dependency is present between channels, less delay slots are needed. When more than six channels are present, the pipeline latency is completely masked requiring no additional delay slots.

5.3 Front-end

The core front-end is composed of two sections, as shown in Fig 5.3, the instruction store and decoder.

5.3.1 Instruction Store

The instruction Store is the memory bank that holds the femtoCore program to be run. From the core perspective this can be seen as a Read Only Memory (ROM), as its contents are immutable and executed in place, without the need for caching. From a system perspective however it can be both read and written to, allowing the dynamic reconfiguration of the program, without the need for whole FPGA reconfiguration. The physical implementation the component uses a dual port block RAM primitive, that is directly connected to the processing system, through the AXI GP1 bus, as a slave device, mapping the core program memory directly in the processor address space allowing higher layers of the stack easy configuration.

The interface on the core side, is much simpler than the one used for programming, as it only needs to support a very simple sequential, read only access pattern, allowing the use of an AXI stream type interconnect without any performance limitation. Interlocking between the two is used to avoid potential conflicts, that could lead to execution of an only partially re-written program.

5.3.2 Instruction Decoder

The instruction set was designed with ease of decoding in mind, with most instruction being a single 32-bit word wide, the only notable exception to this principle, is the constant load instruction, which can load a single floating point to a specified register, that employs two words, where the first contains opcode and destination register, while the second one contains the constant to be loaded. An alternative view that can be adopted, for easier decoder design is to consider the raw instruction stream coming from the store as composed up of two components, the instruction stream proper, formed of single word units, and a set of constants that are interleaved with this last one.

Since no branching will be encountered, as guaranteed by ISA design, a two-step, pipelined, decoding process can be used without any performance penalty, where in the first operation extracts the constants from the raw stream while in the second step the instruction is split up in the composing fields and the appropriate control signal generated. A hardwired decoder im-

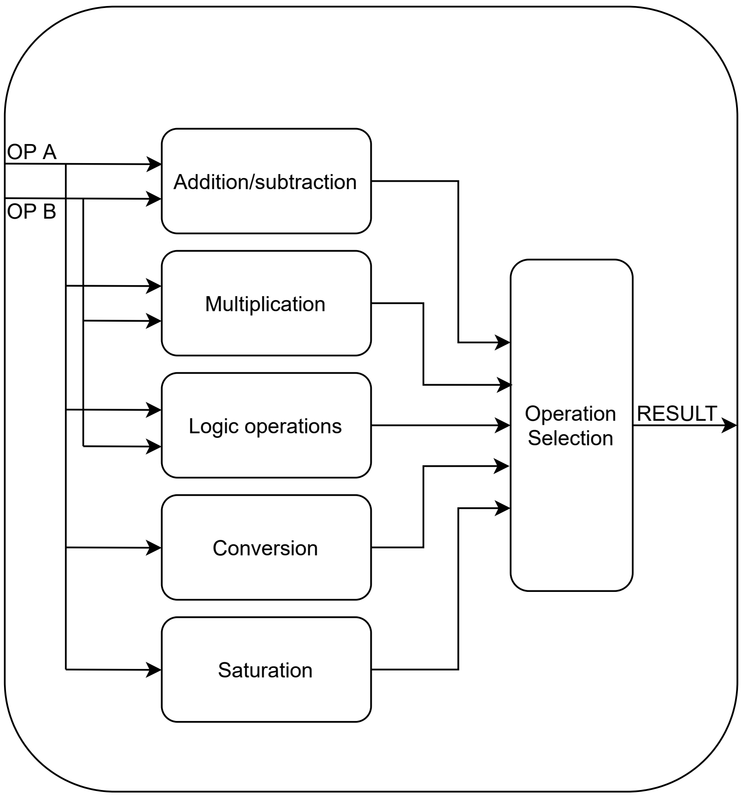


Figure 5.4: Execution unit structure

plementation is chosen, due to the simplicity and small size of the instruction set, making a microcoded design both cumbersome and not necessary.

5.4 Execution

At the heart of the core sits the execution unit, shown in Fig. 5.4, and it is responsible for actually executing the calculations specified by the instruction. Due to the narrow intended application scope for this core only floating point operations are supported, with no integer or memory addressing capability. This module is composed of several units used to implement additions/subtractions, multiplications, logic operations, conversions and saturations. As a deliberate choice, the core can only issue and retire only one of these operations in each clock cycle, even if the hardware could be made capable of executing all five at once with minimal changes. This limitation,

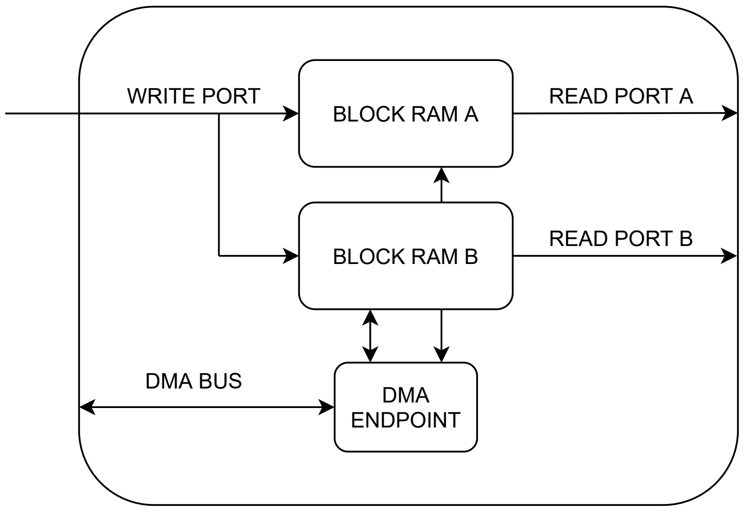


Figure 5.5: Register file structure

while hurting raw throughput limits decoder/compiler complexity, avoiding the need for either Very Long Instruction Word (VLIW) or superscalar execution techniques. It should also be noted that not all problems map well to very wide execution units, as data dependencies can severely limit the amount of instruction level parallelism available, as it is often the case for the type of arithmetic code used in control systems.

5.5 Register File

The resister file, as shown in Fig. 5.5, is the only pool of working memory that the core can access. Avoiding complex memory hierarchies allows for a simpler and more deterministic overall design, with the downside of limiting the total amount of available memory. To support single cycle operation this component has a single write and two separate read ports, as many operations work on two operands. On a hardware level a single memory block is sufficient for a functional implementation of this component, two dual port ram blocks are used with their write port tied together, to effectively map this module on the underlying fixed function RAM blocks present in the FPGA hardware, greatly decreasing logic usage with respect to a naive implementation. A DMA endpoint is also included in the module to allow IO operations from the external world in the femtoCore processor register space. Interlocking is used to prevent DMA access when the core is running, eliminating the need for strict synchronization between core and other components in logic, and avoiding the need for additional concurrent read and write ports to RAM blocks.

5.6 ISA

To achieve the goal of completely deterministic execution, the Instruction Set Architecture (ISA) design plays as much of a role as the processor architecture and implementation themselves. Careful choice of the allowed operations can ensure bounded execution time, while not impeding software development for the targeted application. The lack of ISA level support for branch and procedure call operation ensures a linear and predictable execution flow. The unified memory structure also eliminates the need for most data handling operations, as the whole memory pool can be directly accessed by the execution units, leaving the load constant as the only operation in this class.

From a physical perspective all instructions have a very similar structure with a 5 bit opcode followed by a series of optional 6 bit arguments representing the addresses of operands and destination. Four different structures, shown in Fig. 5.6, are used.

- **Independent instructions**: This structure is used for a varied class of instructions mainly needed to control the execution flow of the program The instructions are composed by the opcode only, with the remaining bits zeroed out for future expansion.

- **Load Constant**: This structure, used for the load constant instruction

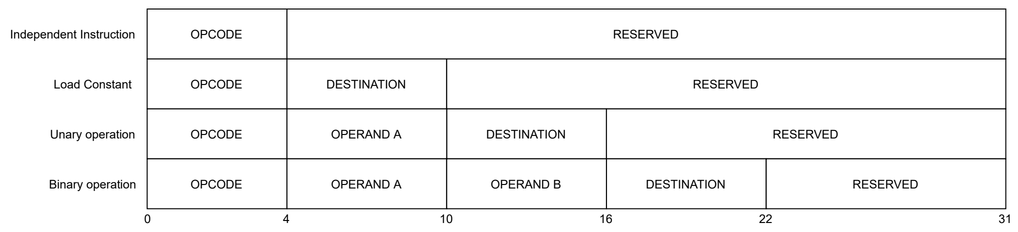


Figure 5.6: Physical instruction structures

only, here the opcode is followed by a destination address, with all other bits zeroed. The constant to load needs to be placed as a complete 32 bit word after this instruction, interleaving it with the instruction stream.

- **Unary instructions**: This structure is used for instruction that act on a single operand and is used for conversion between the floating point format used by the core and fixed integers inputs and output. For these the opcode is followed by operand and result destination addresses.

- **Binary instructions**: This structure is used for arithmetic, logic and comparison instructions that act on two operands and return a result. Here the opcode is followed by the two operand and destination addresses.

A complete list of all instructions included in the ISA is given in Appendix A.

5.7 Software development

When developing an application specific processing core, the software support is as important as the hardware itself. The extreme simplicity of the architecture, coupled with the arithmetic nature of the code typically required in control applications, and use of external DMA transfers for IO, makes this architecture the perfect candidate for direct high level assembly programming. To simplify the development experience, several concepts from higher level languages are supported by the assembler. Bounded loops are available through unrolling, even when the underlying architecture does not have branch instructions, at the expense of program size. Named variables are also supported, allowing the use of descriptive names instead of registers. The complete assembly grammar reference, in extended Backus–Naur form (eBNF) can be found in Appendix B.

5.8 femtoCore High Level Assembler

5.8.1 Frontend

The frontend is the first part of the assembler to execute, reading the input program, to produce the Abstract Syntax Tree (AST). This tree data structure, contains in its nodes and leaves all information needed to generate the output binary. They are widely used in most compiler toolchain implementations as they ease the process of compilation, breaking it down into a series of small transformations serially applied to each node of the tree, in a pattern called Visitor. This section of the Assembler also deals with the include files, which can contain constant and variable declarations, as they are separately parsed, and then merged in the main AST. A map of all the variables and constants used in the whole program is constructed at this stage, to help future optimization passes.

5.8.2 Transformation Passes

The core of the assembler functionality is implemented as a series of optimization passes. Each one of them comprises a function that gets called at each node of the three in a depth first recursive traversal, and potentially modifies either the target node or its children. They are run in the order listed below.

- Loop unrolling: This pass implements fixed size loops, by simply repeating the target instructions the required number of times.
- Pseudo operation substitution: The pass substitutes pseudo-operations, defined by the ISA specification, but not implemented in hardware, like the No Operation (NOP) or register to register move (MOV) instructions in terms of already implemented ones, this allows the programmers to use a more expressive and understandable set of instructions while minimizing hardware decoder complexity.
- Variable lifetime mapping: This pass performs a linear scan over the code, determining the lifetime of each, non IO involved variable, defined by their first and last use. This information is then stored in the variable map, constructed during parsing, to be used by subsequent passes
- Constant load interleaving: With this pass the constants specified by the "load constant" operation are interleaved in the instruction stream as specified by the ISA.
- Register allocation: This last pass implements a register allocation operation, assigning a specific physical register to each variable in the code. A linear scan approach has been used, as opposed to the more conventional graph colouring technique, as the lack of a larger pool of memory where to spill to in case of conflicts limits the number of variables usable in a program to the maximum number of registers, negating the advantages of the more complex algorithm.

5.8.3 Code Generation

This is the last component of the femtoCore assembler, and it is responsible for the conversion of the optimized AST to a stream of instruction to be fed to the hardware. Operatively, the tree is traversed one last time while the instruction contained in each tree are constructed, targeting a specified instruction format. To account for possible future expansion of the instruction set, along with the possible addition of more registers, the width of both opcode and register address fields is parametrized and easily modified.

Chapter 6

Independent current control

The current control is the final component needed for a complete machine drive system. From a high level perspective, it is as important as the hardware itself to overall performance. It hides the complex non-ideal characteristics of any real-world electrical machine from higher level controllers, simplifying their design. This control system, is also of fundamental importance for the fault tolerance characteristics of the entire system, as it also implements most of the counter-measures necessary to keep the system running.

The traditional coordinate transformation based field-oriented control techniques have been discarded, as even if they can deal with faults, they have less than ideal scaling characteristics. The direct and inverse transformations used as for the vector space change of base require knowledge of the complete state of the system. This factor severely limits horizontal scaling, as even in a distributed environment the network between the nodes can quickly become the performance bottleneck.

The chosen static reference frame based approach, on the other hand does not suffer from these problems, as the current for each phase is controlled locally to each phase in a completely independent manner. This allows to mantein a one-to-one relationship between sensors, controllers and power

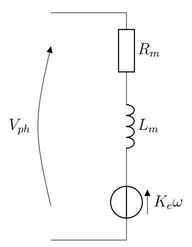


Figure 6.1: Machine phase equivalent circuit

cells, enabling the fault-tolerant control of a very high phase-count system by splitting it into multiple, potentially independent zones. Another benefit of the chosen control system architecture is its behaviour with respect to effects like mutual inductance between windings. Which is treated as a pure disturbance by the controllers and consequently rejected, without requiring additional compensating actions, usually needed for traditional transformation based techniques [73].

6.1 Machine modelling

Before moving to a more in detail analysis of the controllers themselves, the modelling for the rest of the system needs to be detailed. The standard machine modelling techniques, using the rotating reference frame with d and q axes, cannot be applied as the whole control is performed in the abc space. An additional challenge, not normally present with more traditional architectures, is the fact that the frequency of the current, and consequently the rotational speed, is not a constant but one of the controller inputs, leading to a Multiple Inputs Single Output characteristic.

The starting point to derive a plant model that is compatible with the constraints was the RLE equivalent circuit of a machine phase, shown in Fig. 6.1. This was coupled with mechanical inertia and torque-power relationships, forming the system of equations shown in (6.1) where T_E is the electrical torque, T_L is the mechanical load torque, T_A is the acceleration torque due to the rotor moment of inertia J and α_m is the angular acceleration of the rotor.

$$\begin{cases} V_{ph} = (R_{ph} + sL_{ph})I_{ph} + K_e\omega_e \\ \omega_e = \frac{P_M}{T_E} \approx \frac{V_{ph}I_{ph}}{T_E} \\ T_E = T_A + T_L = J\alpha_m + T_L = J\frac{d\omega_m}{dt} + T_L \end{cases} \tag{6.1}$$

A couple of simplifying assumptions are needed to extract a transfer matrix that can be used for control system design purposes. First and foremost, machine losses have been neglected, equating mechanical and electrical power. This is necessary to cleanly separate the contributions to phase current given by input voltage and electrical frequency. As the non LTI nature of the first equation makes extraction of transfer functions impossible. Another simplification that is made it that of unitary power factor. These factors, while decreasing the absolute accuracy of the mathematical model, do not fundamentally alter its behaviour from a control system design perspective, especially when concerning stability.

Even with these assumptions in place the Back-EMF term defined as:

$$V_{BEMF} = K_e\omega = \frac{V_{ph}I_{ph}}{J\alpha_m + T_L} \tag{6.2}$$

still needs some processing, as it is non-linear for the following two reasons: one of the inputs (angular speed) appears in the denominator, as well as the other input (phase voltage) being directly multiplied by the output (phase current). To deal with these challenges this factor has been linearized by using a first order Taylor series approximation, leading to equation (6.3).

$$\begin{aligned} V_{BEMF} = K_eV_{ph}I_{ph}G(x) &\longrightarrow K_e\nabla(V_{ph}I_{ph}G(x)) \\ &= K_e(\dot{V}_{ph}I_0G(0) + V_0\dot{I}_{ph}G(0) + V_0I_0\dot{G}(x)) \end{aligned} \tag{6.3}$$

where

$$G(x) = \frac{1}{J\alpha_e \frac{2}{n_p} + T_L} = \frac{1}{T_L} - \frac{J\frac{2}{n_p}\alpha_e}{T_L^2}$$

finally, combining everything together, leads to an equation in two variables associated to the MISO system that needs to be controlled. From this the extraction of a transfer matrix, shown in (6.4) is relatively straightforward.

$$I = \begin{bmatrix} \frac{1 - I_0 \frac{K_e \eta}{T_L}}{R_{ph} + sL_{ph} + V_0 \frac{K_e \eta}{T_L}} & \frac{J \frac{2}{np} s}{T_L (R_{ph} + sL_{ph} + \frac{V_0}{T_L})} \end{bmatrix} \begin{bmatrix} V_{IN} \\ \omega_e \end{bmatrix} \tag{6.4}$$

6.2 Current control structure

6.2.1 Overview

For fault-tolerant operation, the current control system must be able to position each current vector independently of the others. This decouples torque control, that is inherently linked to the machine structure, and health state of the system, from each single phase current control, which can be effectively performed by each phase power cell independently, if they form a balanced set, whose sum is zero, to respect Kirchhoff's current law at the isolated machine neutral point.

To achieve this goal, the current controllers themselves must be moved from the $d-q$ or VSD space, back to the regular *abc* space, as shown in the system diagram in Fig. 6.2, where there is a one to one correspondence between controller and physical machine phase.

In case of fault, the ensuing disturbance will affect all controllers equally and only if the references remain unchanged. As opposed in a more traditional architecture where the controllers are placed in a transformed vector space where not all controllers might be affected, leading to a much more

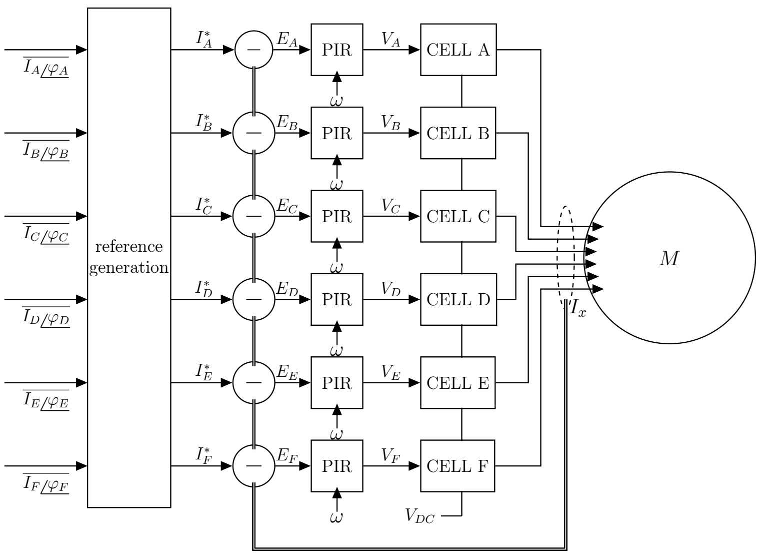


Figure 6.2: System level control architecture

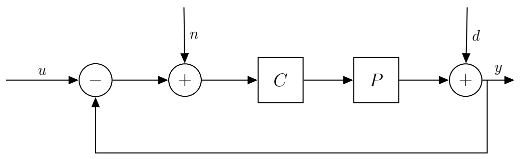


Figure 6.3: Generic Feedback control system diagram

problematic unbalance.

6.3 Sinusoidal reference tracking

In Fig. 6.3 is shown a generic feedback control system where n and d represent additive noise and disturbances, C and P are the controller and plant transfer functions, while u and y the input and output vectors. Its tracking performance is determined by the ability of the controller to reject disturbances, and consequently can be easily measured through the sensitivity function, defined as the transfer function between disturbance and output, given as:

$$S(s) = \frac{Y(s)}{D(s)} = \frac{1}{1 + C(s)P(s)} \tag{6.5}$$

To achieve no steady-state error this needs to reach zero, condition only possible when the loop gain $C(s)G(s)$ is infinite. For constant or quasi static references this means infinite steady state gain, whereas for other signal reference the gain peak must happen at the appropriate frequencies. Consequently, several of the most used control techniques such as PID or Linear-Quadratic Regulators (LQR), are unable to track sinusoidal signals with no steady state error, and for this reason, they can not be directly used for applications that rely on them.

6.4 Resonant control

Resonant control techniques are among the few, able to track a sinusoidal signal. These add a gain peak in the controller transfer function, through the insertion of a purely imaginary complex-conjugate pole pair at the desired operating frequency, allowing the elimination of steady state error. Very simple and well known implementations of this methodology are the PR and PIR controllers, where a resonant element is combined in the structure of the more conventional PI controller, either as a substitution or in addition to the integrator, depending on whether zero steady state error is desired. This results in the transfer function (6.6)

$$PIR(s) = K_p + K_r \frac{\omega s}{s^2 + \omega^2} + \frac{K_i}{s} \tag{6.6}$$

A frequent modification to the basic resonant element, is the addition of a damping term, reducing the infinite gain to a selected high value, improving control stability, resulting in transfer function (6.7)

$$PIR(s) = K_p + K_r \frac{\omega \zeta s}{s^2 + \omega \zeta s + \omega^2} + \frac{K_i}{s} \tag{6.7}$$

These techniques have already been proposed and demonstrated for traditional motor control applications [74, 75]. The implementation of the

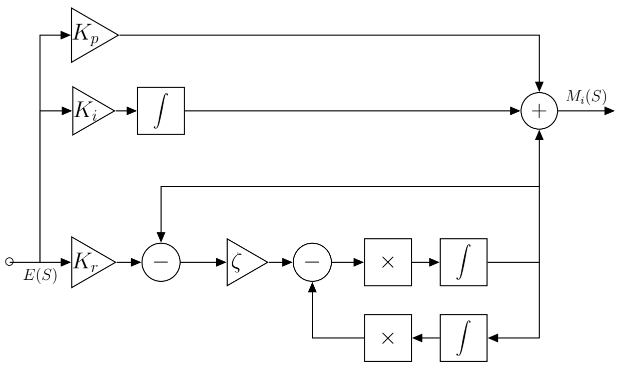


Figure 6.4: PIR controller diagram

PIR controllers is done through the structure shown in Fig. 6.4, where a Second Order Generalized Integrator (SOGI) is used to synthesize the resonant part of the desired transfer function.

6.4.1 Controller Tuning

As for the regular PI and PID controllers, the tuning of the various gain components in a PIR controller is what ultimately determines the shape of the implemented transfer function. Several methods have been used to this effect, from a modified version of the popular Ziegler–Nichols heuristic [76], to optimal Linear-Quadratic Regulators (LQR) based control techniques [77], to global optimizations based on genetic algorithms [78]. Most of these methods rely however on an extremely precise model of the plant, as it is crucial in the whole tuning process, and poorly handle disturbances, which can result in instability under non-ideal conditions. To address these issues the several techniques have been introduced in the field of robust controls, that aim to analyse and synthesize controllers in presence of disturbances of model uncertainty, to maximize stability and evaluate their impact on the closed loop system performance. The μ-synthesis [79] approach based on structured singular values (SSV) allows to tune a fixed structure controller through H_∞ methods to achieve a final set of parameters that can minimize the effects of disturbances, such as model uncertainty or external noise on the closed loop performance. The method employed to tune a fixed structure controller to minimize the impact of disturbances is called D-K iteration. In the first step of this iterative process, a regular H_∞ synthesis is used to tune the controller, minimizing the overall gain of the system. Then the robust performance of the system is evaluated, obtaining a scaled norm, this is the D step. Subsequently, in the K step, a new synthesis is performed minimizing this norm. These two steps are then repeated until a desired convergence criterion is met.

In order to guide the synthesis process toward the desired system behaviour, a set of weight functions are selected and applied to the inputs (W_i for the current set-point and W_w for the angular velocity), to the outputs (W_o) and to the control variables (W_v for the voltage); in a process called H_∞ loop shaping. Creating a set of virtual outputs, four in this case, only used during the controller synthesis process. The only constraints placed to the optimization process used for these steps, are upper and lower bounds for the tuned parameters, to speed up the computational process by limiting the search space and ensure a solution with reasonable values (neither too small nor too large) to avoid potential numerical problems.

A multiplicative uncertainty in a range of 30% around the mean value has been assumed for both the moment of inertia and the rotor magnet flux linkage, as the direct measurement of these quantities is often not practical. It should be noted that the type and amount of uncertainty that needs to be added to the model is highly variable on a case by case basis.

A result of the tuning process is shown in the Bode diagrams in Fig. 6.5, where multiple lines are drawn, to show the effects of the uncertainties on system performance and the robustness of the controller tuning. The exact controller gain values obtained through this process, used in the experimental

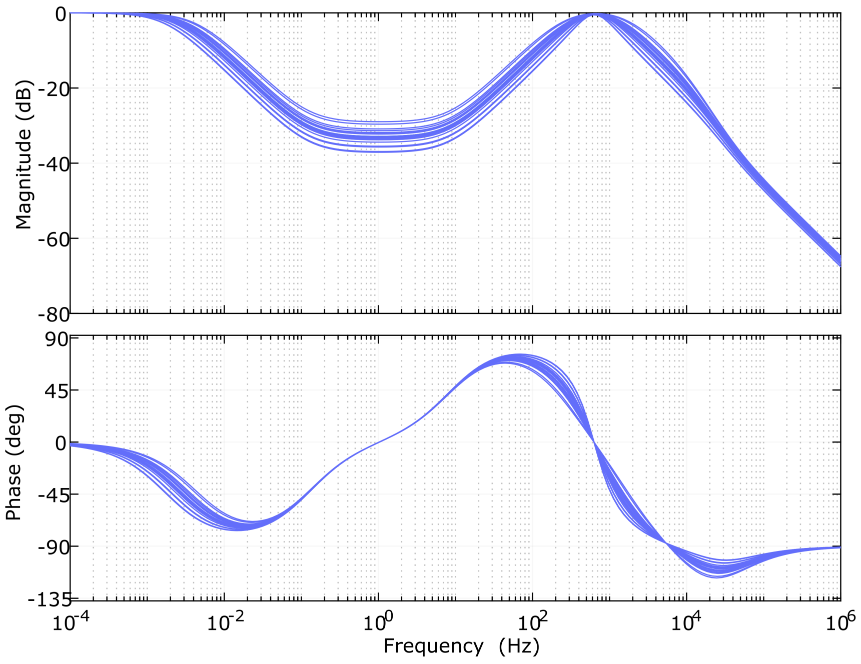


Figure 6.5: Closed loop transfer function from set-point to output current of the uncertain system

Table 6.1: Controller parameters

parameter	value
K_p	0.8
K_r	1000
K_i	0.5
ζ	0.0591

section are shown in Table 6.1.

6.5 Handled faults

Key for the successful implementation of a fault-tolerant machine drive system is a correct partition of responsibilities, between the different aspects and layers of a design. Not only with respect to the regular operating mode, but also with regard to fault handling, simplifying the overall design process. For this reason, some type of problems, such as loss of the converter power supply, have not been considered in the control system design, as a system level approach is required to achieve fault-tolerant operation, through component or even subsystem level redundancy. Other issues, loss of position feedback, can be effectively solved with lower level self-contained approaches, namely the addition of a speed/position estimator, as shown in [80, 81], whose outputs are used upon sensor fault detection, transitioning the drive to sensorless operation.

In Table 6.2 are specified the types of fault relevant to the current control system design, and how they are handled by the proposed architecture. Open circuit (OC) faults in all components in the current path: inverter, cables and machine stator, can be handled natively by the proposed architecture, with simple reconfiguration of the input references to maximize torque generation. Similarly, short circuit (SC) to neutral inside the machine stator coils can be resolved by reconfiguring the current references to operate the machine with a reduced number of phases, and shutting down the appropriate inverter portion to avoid unwanted fault currents. When instead the short circuit is to grounded components, like machine chassis or grounded shield conductors in the cables (if present), the options are usually much more limited, as the entire drive system will need to be shut down to avoid potentially dangerous ground currents. Finally, when the short circuit is in one of the drive phases, this needs to be excluded from the system before reconfiguration. This can be done through external contactors or fuses if present, or if sufficient fault isolation between adjacent phases is present through an intentional shoot through event.

6.5.1 Fault mode operation

While the proposed architecture can gracefully handle failure of one or more phases without immediate stability challenges, it is still desirable to perform

Component	Fault	Response
Armature	OC	Reconfiguration
	SC to neutral	Reconfiguration
	SC to ground	Safety shutdown
Cable	OC	Reconfiguration
	SC to ground	Safety shutdown
Inverter	OC	Reconfiguration
	SC	Exclusion of affected phase and reconfiguration

Table 6.2: Overview of relevant fault modes and how they can be handled

fault detection, which is now a delay insensitive process, and it can be done through one of the many approaches published in literature [82, 83]. In order to minimize performance loss, the drive operation should then be reconfigured to achieve optimal performance. In the case of the proposed architecture, this operation is extremely simple, and can be easily performed online. It only consists in a simple reference signal change, to produce the new set of sinusoidal references, thanks to the controller ability track arbitrary current phasors.

First and foremost, it is important to determine the type of fault experienced by the system. Without delving into excessive detail two large classes can be easily identified, open and short circuit faults. From a drive operating perspective, these two must be treated differently. To deal with short circuit faults, the affected part of the system needs to be isolated, in order to interrupt the flow of current, transitioning to a more easily dealt with open type fault. Several techniques have been proposed to do so [84, 85], and they fall broadly in two groups, active, where thyristors or relays are used to open the affected path, or passive, where fuses are added to the circuit and react to the current spike following the short. In case of an open circuit fault, either direct, or induced, the offending phase is completely cut off making it unable to generate torque, while the rest of the system is unaffected, making it possible, for multiphase machines, to carry on working.

To achieve optimal post fault behaviour the remaining phases current vectors should be reconfigured to still generate a smooth rotating MMF. The easiest way to calculate the phase angles needed by the reference signal generators in hardware, is by using a Vector Space Decomposition Approach [86, 87]. By using the transformation shown in equation (6.8) the phase current of the star-connected asymmetrical six phase machine, are moved to a new vector space, that can be subdivided in three sub-spaces where torque is generated only by current in the $\alpha - \beta$ plane while the others largely only contribute to losses, with only second order effects on the electromechanical energy conversion, in most machines.

$$\begin{bmatrix} I_\alpha \\ I_\beta \\ I_x \\ I_y \\ I_{0+} \\ I_{0-} \end{bmatrix} = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 & -1/2 & -1/2 & \sqrt{3}/2 & -\sqrt{3}/2 & 0 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 & 1/2 & 1/2 & -1 \\ 1 & -1/2 & -1/2 & -\sqrt{3}/2 & \sqrt{3}/2 & 0 \\ 0 & -\sqrt{3}/2 & \sqrt{3}/2 & 1/2 & 1/2 & -1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} I_a \\ I_b \\ I_c \\ I_d \\ I_e \\ I_f \end{bmatrix} \tag{6.8}$$

To guarantee smooth torque generation, it is sufficient that the circular trajectory described by the first two currents in their plane remains invaried, in both pre- and post-fault configurations. Other constraints that the set of currents needs to satisfy is to have zero current in the $x - y$ plane, to avoid any unnecessary losses, and finally that the current in the positive and negative zero sequence current balance out, due to the neutral point arrangement. All these are summarized in equations (6.9) to (6.11).

$$
\begin{aligned}
I_\alpha &= I_\beta \\
\varphi_\alpha &= \varphi_\beta - \frac{\pi}{2}
\end{aligned}
\tag{6.9}
$$

$$
\begin{aligned}
I_{\alpha}\underline{/\varphi_\alpha} = \sqrt{2}(I_{A}\underline{/\varphi_A} - \frac{1}{2}I_{B}\underline{/\varphi_B} - \frac{1}{2}I_{C}\underline{/\varphi_C} + \\
\frac{\sqrt{3}}{2}I_{D}\underline{/\varphi_D} - \frac{\sqrt{3}}{2}I_{E}\underline{/\varphi_E}) \\
I_{\beta}\underline{/\varphi_\beta} = \sqrt{2}(\frac{\sqrt{3}}{2}I_{B}\underline{/\varphi_B} - \frac{\sqrt{3}}{2}I_{C}\underline{/\varphi_C} + \frac{1}{2}I_{D}\underline{/\varphi_D} + \\
\frac{1}{2}I_{E}\underline{/\varphi_E} - I_{F}\underline{/\varphi_F})
\end{aligned}
\tag{6.10}
$$

$$
\begin{aligned}
I_n = 0 \; \forall n \in \{faulty\ phases\} \\
I_A + I_B + I_C + I_D + I_E + I_F = 0
\end{aligned}
\tag{6.11}
$$

When running in nominal conditions and all phases are working, only one solution to this set of constraints is possible, with even a single faulty phase, the problem becomes not fully constrained, leading to an infinite number of potential solutions. To choose the best one an optimization problem is set up, to find the solution that satisfies these constraints while minimizing the average conduction losses, using the following cost function:

$$
I_A^2 + I_B^2 + I_C^2 + I_D^2 + I_E^2 + I_F^2 \tag{6.12}
$$

From an implementation perspective, this approach does not suffer from the problems delineated in the introduction, as this expensive part can be performed offline, pre-computing the angles and relative magnitude the reference phasors in each scenario, storing them in a simple Look Up Table

(LUT) to be consulted at runtime.

6.6 Controller stability

In the field of control engineering there are several techniques that can be used to evaluate stability of a system, the simplest, especially when working with small linear systems, are the ones from classical control theory based on stability margins and the Nyquist plot. While these are very widely used in the analysis of the current control loops for power converters, they are not suitable for the designed system, as they become increasingly difficult to apply for systems with multiple inputs and outputs. For these reasons a state space approach was followed.

6.6.1 State-Space Representation

The state of a dynamical system is a set of variables containing all present and past information needed, along with the current inputs to fully describe it future behaviour. The state-space representation, is a mathematical model that correlates the state, inputs and outputs of a system through linear differential equations shown in (6.13) where x denotes the state vector, while y and u the system output and input vectors respectively with A, B, C, D being four matrices that represent the relationships between them.

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{6.13}$$

It can be shown that, for a system in this form, the eigenvectors of the A matrix, obtained by solving (6.14) with I being the identity matrix, correspond to the poles of the system. That can thus be calculated by simply evaluating the determinant of matrix A, this allows the application of the

Routh–Hurwitz criterion, for which a necessary and sufficient condition for system stability is the absence of poles with positive real part.

$$det(sI - A) = 0 \tag{6.14}$$

6.6.2 Stability analysis

To prove stability of the proposed architecture, in both pre- and post-fault configurations, stability analysis needs to be performed, to mathematically guarantee that the system is well-behaved in all situation. While several tools can be utilized to this effect, the choice was made to use a state-space approach due to its simplicity, especially for higher order models. As a first step in its derivation process the dynamics of the complete system need to be described with one or more differential equations. This can be done by equating the voltage at each machine terminal phase with the controller output, also considering the interactions between different phases given by mutual inductances, as shown in equation (6.15) where I_s is the current in the examined phase, while I_x is the one in the others, L and $M_{s,x}$ are respectively the winding self inductance and mutual inductance to the other windings, calculated with the same method as in [88] and V_B is the Back-EMF voltage. This last term will be processed similarly to when extracting the machine transfer function, with the only difference being that the angular speed can be considered constant, for this analysis that is aimed strictly at electrical domain phenomena.

$$\begin{split}(K_p + K_r \frac{\zeta\omega' s}{s^2 + \zeta\omega' s + \omega^2} + \frac{K_i}{s})(I_s^* - I_s) = \\ RI_s + sLI_s + sM_{s,x}I_x + V_B\end{split} \tag{6.15}$$

$$X = \begin{bmatrix} I_x \\ \dot{I}_x \\ \ddot{I}_x \\ \dddot{I}_x \\ \ddddot{I}_x \end{bmatrix} \quad U = \begin{bmatrix} I_x^* \\ \dot{I}_x^* \\ \ddot{I}_x^* \\ \dddot{I}_x^* \end{bmatrix} \tag{6.16}$$

After elimination of the denominator from the left-hand side and expansion/collection of all relevant terms, an Ordinary Differential Equation (ODE) with several fourth order terms. The state and input vector choice is shown in equation (6.16) where I_x represents a vector of the six phase currents and I_x^* is a vector of the six current setpoints. A state space model for the nominal system can now be extracted and is shown in equations (6.17) to (6.27) where the overlined quantities define the relevant 6×6 matrix (i.e. $\bar{R}$ for resistance, $\bar{L}$ for inductances, etc.)

$$\boldsymbol{A}_{\mathbf{(30\times30)}} = \begin{bmatrix} 0_{6,6} & I_{d,6} & 0_{6,6} & 0_{6,6} & 0_{6,6} \\ 0_{6,6} & 0_{6,6} & I_{d,6} & 0_{6,6} & 0_{6,6} \\ 0_{6,6} & 0_{6,6} & 0_{6,6} & I_{d,6} & 0_{6,6} \\ (K_i/\bar{L_n})I_{d,6} & A_2 & A_3 & A_4 & A_5 \\ 0_{6,6} & 0_{6,6} & 0_{6,6} & 0_{6,6} & 0_{6,6} \end{bmatrix} \tag{6.17}$$

$$A_2 = \left(\frac{K_p + \bar{R}}{\bar{L_n}}\omega^2 + \frac{K_i}{\bar{L_n}}\zeta\omega + \frac{K_e V_0 \omega^2}{\bar{L_n} T_L}\right) \cdot I_{d,6} \tag{6.18}$$

$$A_3 = \left(\frac{K_p + K_r + \bar{R}}{\bar{L_n}}\omega\zeta + \frac{K_i}{\bar{L_n}} + \omega^2 + \frac{K_e V_0 \omega\zeta}{T_L \bar{L_n}}\right) \cdot I_{d,6} + \left(-\frac{M_{n,m}}{\bar{L_n}}\omega^2\right) \cdot [J_{6,6} - I_{d,6}] \tag{6.19}$$

$$A_4 = \left(\frac{K_p + \bar{R}}{\bar{L_n}} + \omega\zeta + \frac{K_e V_0}{T_L \bar{L_n}}\right) \cdot I_{d,6} + \left(\frac{M_{n,m}}{\bar{L_n}}\omega\zeta\right) \cdot [J_{6,6} - I_{d,6}] \tag{6.20}$$

$$A_5 = \frac{\bar{M_{n,m}}}{\bar{L_n}}\omega\zeta[J_{6,6} - I_{d,6}] \tag{6.21}$$

$$\boldsymbol{B}_{\mathbf{(24\times30)}} = \begin{bmatrix} 0_{6,6} & 0_{6,6} & 0_{6,6} & 0_{6,6} \\ 0_{6,6} & 0_{6,6} & 0_{6,6} & 0_{6,6} \\ 0_{6,6} & 0_{6,6} & 0_{6,6} & 0_{6,6} \\ B_1 & B_2 & B_3 & B_4 \\ 0_{6,6} & 0_{6,6} & 0_{6,6} & 0_{6,6} \end{bmatrix} \tag{6.22}$$

$$B_1 = \frac{K_i}{\bar{L_n}}\omega^2 I_{d,6} \tag{6.23}$$

$$B_2 = \left(\frac{K_p}{\bar{L_n}}\omega^2 + \frac{K_i}{\bar{L_n}}\zeta\omega\right) I_{d,6} \tag{6.24}$$

$$B_3 = \left(\frac{K_p + K_r}{\bar{L_n}}\omega\zeta + \frac{K_i}{\bar{L_n}}\right) I_{d,6} \tag{6.25}$$

$$B_4 = \frac{K_p}{\bar{L_n}} I_{d,6} \tag{6.26}$$

$$\boldsymbol{C}_{\mathbf{(1\times30)}} = [1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad\; 0 \quad ... \quad 0] \tag{6.27}$$

$$\boldsymbol{D}_{\mathbf{(1\times24)}} = [0 \quad ... \quad 0] \tag{6.28}$$

The stability analysis, once the system behaviour is described with this representation can be easily guaranteed through analysis of the eigenvalues of the $\boldsymbol{A}$ matrix, that corresponding with the poles of the system, must have zero or negative real part. To then re-assess stability of the system in a post open circuit fault state, it is sufficient to zero out all the components of the $\boldsymbol{A}$ matrix related to the faulty phase, as the fault cause all current flow to stop. The eigenvalues of this new model can be inspected to prove stability even in this configuration. The results of both normal and faulty system stability analyses, shown in the pole-zero map at Fig. 6.6, demonstrate through linear analysis, and under the following assumptions:

- Negligible machine losses
- Unitary power factor
- Absence of saturation in both stator and rotor

That the control system is guaranteed to be stable, even during a single phase open circuit fault scenario. It should also be noted that the inclusion of the mutual induction contribution in the analysis verifies the viability of the control system design based on independent phases assumption.

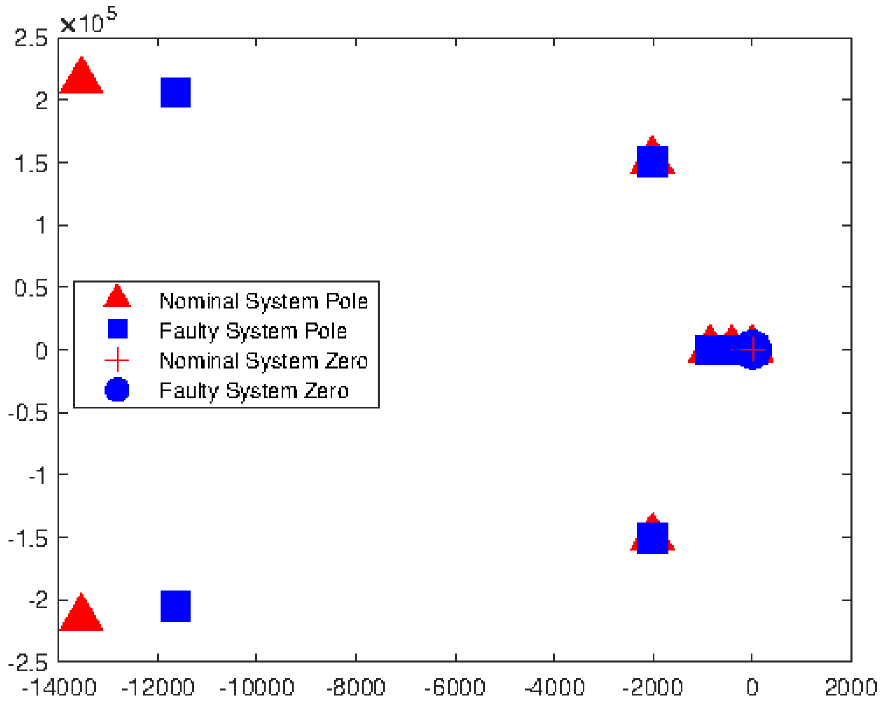


Figure 6.6: Pole zero map of the current control system in both pre (blue marks) and post (red marks) fault scenarios

Chapter 7

Simulations

7.1 Simulation goals

It has become standard procedure, in modern development and design flows to use Simulation as an intermediate step between regular analytical performance evaluation and experimental testing to further validate a designs performance and robustness. In particular, with the help of these techniques it is possible to evaluate the impact of second order effects, materials non-idealities and their various possible interactions. With this overarching goal in mind the design presented in the previous chapters has been simulated in order to achieve the following goals:

- Validate the control system stability, in a closer to reality setting with respect to the mathematical stability analysis.
- Evaluate the effects of temperature and ensure the absence of possible thermal runaway conditions.
- Evaluate sensitivity of the drive performance to parameter variations.

To achieve these objectives several simulations have been prepared. The first one, concerned with evaluating the purely electrical aspects of the

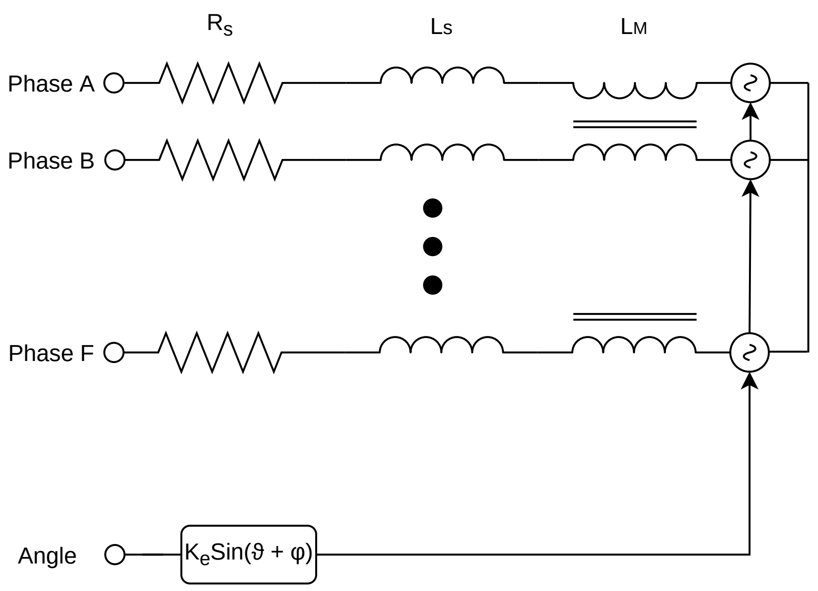


Figure 7.1: Simulation model of the electrical machine.

design, without taking temperature into account as a factor. Successively the thermal aspect of the design will be studied. Finally, a sensitivity analysis will be performed on a significant subset of components to identify the critical points in the design where most of the manufacturing attentions should be placed, and where on the other hand potential variations can be allowed without effects on the overall performance.

7.2 Electrical Simulation

7.2.1 Setup

Electrical Machine

Since the complete drive system is the focus of this thesis, rather than just the electrical machine itself, or its control, the decision has been made to use the simplest electrical machine model suitable for system level simulation, rather than seeking the absolute highest modelling accuracy possible. The first requirement for these type of models is the computational efficiency,

Parameter	Value
Phase resistance	$8.5\,\mathrm{m\Omega}$
Phase Inductance	$166\,\mathrm{\mu H}$
Mutual inductance	$66\,\mathrm{\mu H}$
pole pairs	2
Rotor moment of inertia	$0.0038\,\mathrm{Nms^2}$
Back EMF constant	$0.1846\,\mathrm{V\,Hz^{-1}}$
Rated Power	18 kW
Peak rated voltage	312 V
Rated speed	3000 rpm

Table 7.1: Machine parameters

as the long timescale needed to assess thermal and mechanical transients would extend excessively the required processing time for each individual simulation. It is for these reasons that the use of Finite Element Method (FEM) techniques has been excluded. High level look-up table based "black box" type models, while very efficient, are not suitable, as they are difficult to successfully and accurately interface with the circuital models used for the converter. The choice has consequently fallen on a Simple RLE model, shown in Fig. 7.1, where θ is the rotor electrical angle and φ is the phase angle. This model, on top of being very computationally efficient allows for a simple injection of one or more open phase faults, that can be modelled as a time dependent resistance value. The mechanical side of the machine was also modelled, keeping into account the rotor inertia, and a user controlled load torque, allowing simulated load step testing.

Converter

The first class of models that can be used to simulate electronic circuits is based on non-linear differential equations, that accurately describe the behaviour of all passive and active components, with all the associated higher order effects. This approach, routinely used for analogue and Integrated Circuit design, is not suitable for system level simulation of power electronics

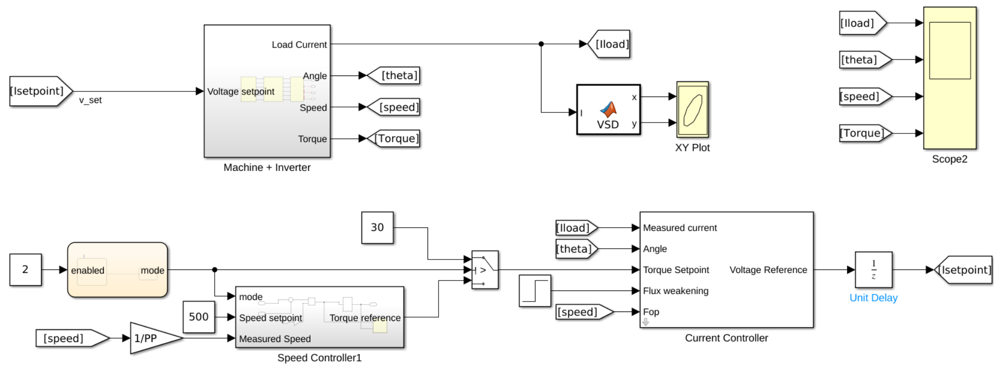


Figure 7.2: Complete simulation model.

converters, as the vastly different time-constants present in the overall model, ranging from sub-nanosecond, for device internal dynamics, to the seconds typical of mechanical and especially thermal sub-systems, mean that some parts of the model need to be evaluated many millions or even billions of time in each run, making the overall simulation extremely heavy.

A useful property of power electronic circuits in general, and machine drives in particular, is the fact that the behaviour of voltages and currents in the circuit are fully determined by passive components, especially inductances and capacitances, as the associated time constants are orders of magnitude slower with respect to the transient behaviour of transistor commutations. This means that as long as the transistors are strictly operated between the fully on and off states, they can be treated as ideal switches, commutating instantly between the two states, eventually keeping track of on state voltage loss through a small value time dependent resistor This piece-wise linear approach significantly reduces computational complexity making allowing even the longer time-frames to be simulated.

7.2.2 Results

The complete system, consisting of electrical machine, converter and control system, shown in Fig. 7.2, is implemented in a MATLAB/Simulink environ-

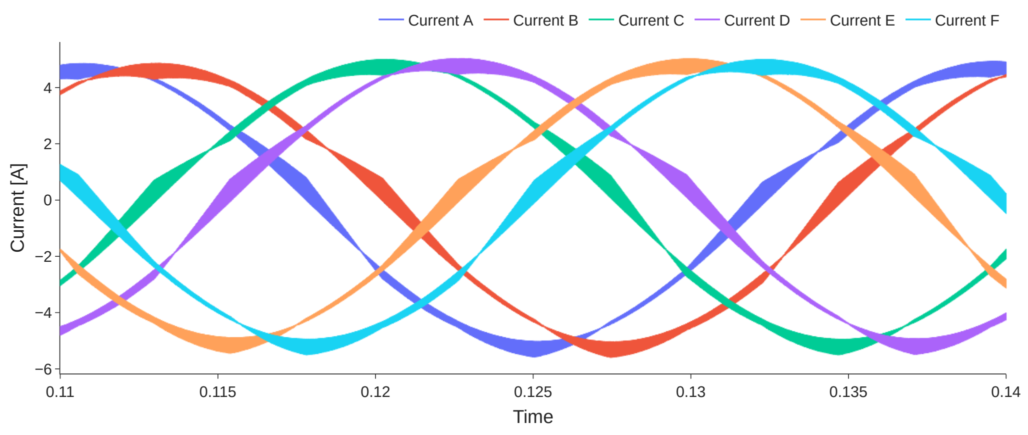


Figure 7.3: Steady state currents in simulation

ment where the electrical domain is managed through the use of the PLECS piece-wise linear simulator. For all simulations constant current control was used, with a current amplitude profile specific to one of the following test cases:

- Steady state test.
- Load step test.
- Single Open phase fault transient.

The first simulation is aimed at evaluating the steady state performance of the control system, so a constant fixed amplitude current is used with a constant torque load. The phase currents, shown in Fig. 7.3 are then saved after having waited for the control system to settle following the initial transient.

To evaluate the dynamic performance of the designed system a load step simulation is performed. For limitations in the experimental setup that will be used to validate these simulations, a step in the reference current amplitudes, with the machine connected to a simulated constant speed load was used, as opposed to a straight load torque step. The results, shown in Fig. 7.4 shows fast response without overshoots.

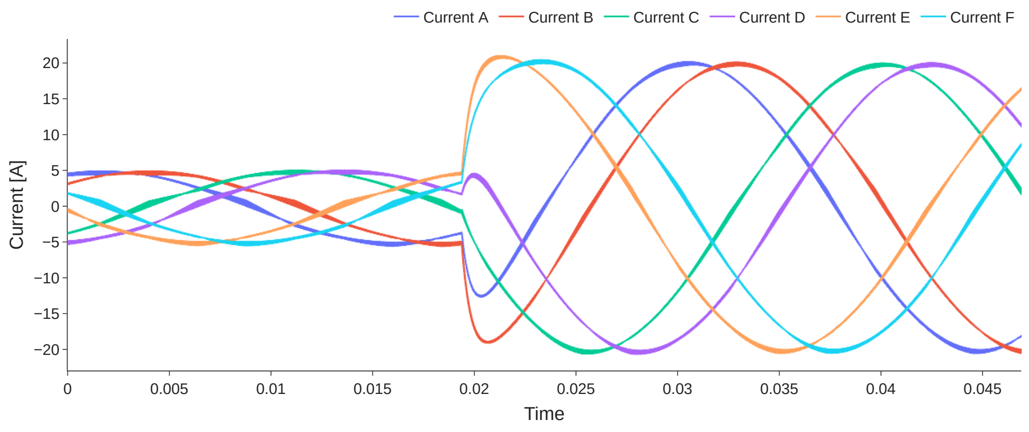


Figure 7.4: Load step currents in simulation

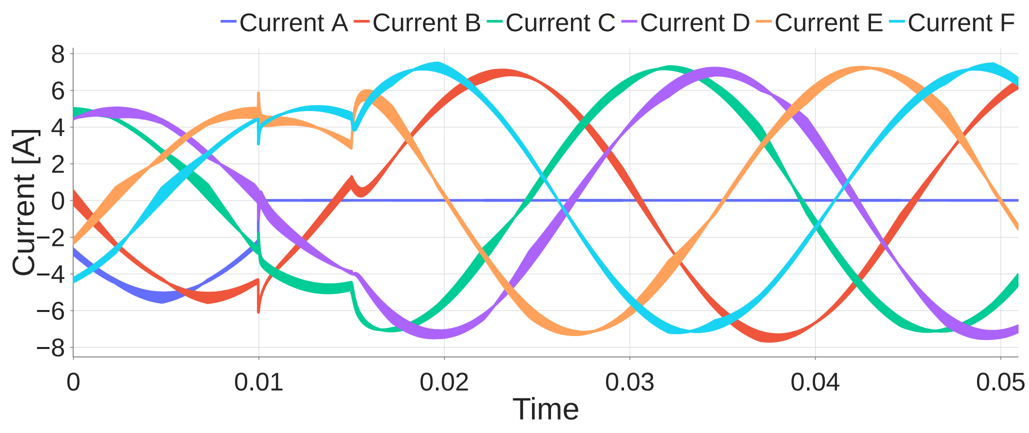


Figure 7.5: Fault transient currents in simulation

The last and most important test, simulates an open circuit fault transient on one of the six phases of the machine. To simulate this event, a high value resistor was placed in series with the affected phase, effectively reducing the current that can flow through it to negligible levels. After a 5 ms delay, emulating the time required for fault detection, the references are then reconfigured for optimal post-fault operation. The behaviour shown in the simulation, confirms the theoretical stability analysis with the control system keeping control of the current throughout the fault and resuming regular operation once completed reconfiguration.

Chapter 8

Experimental validation

8.1 Communication protocol Testing

Before starting the main phase of this experimental campaign, a series of tests has been performed to assess the performance of the developed communication protocol, in terms of latency introduced into the communication path, and resistance to random errors. For both these tests, the setup shown in Fig. 8.1 was used, where both the controller and the power cell logic have been implemented in the same FPGA, and linked together through a physical loopback connection, consisting of a short optical fibre section.

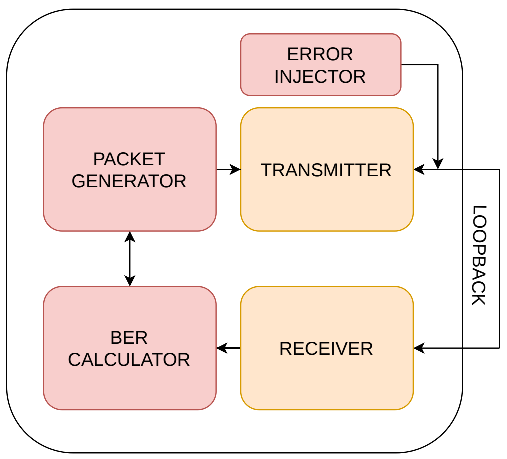


Figure 8.1: Communication Protocol testing setup

Additional infrastructure also implemented alongside the logic under test, where needed to support the measurements. All tests were run with an FPGA clock frequency of 100 MHz, and Optical transceivers with a 40 MHz bandwidth.

8.2 Latency testing

The measurement of the latency introduced by the designed communication protocol is performed on a physical loopback connection, where the transceiver input and output connections are tied together with an appropriate length of transmission medium. For the test, random traffic is generated and fed to the transmission pipeline, while the reception one is monitored, as the received data is passed onwards for Bit Error Rate (BER) calculation. Measurement of the end to end latency was performed both in the FPGA logic by starting a counter upon packet transmission and stopping after completion of the reception phase, as well as with an oscilloscope, monitoring the transmitter and receiver activity signals, that are made available on one of the FPGA IO pins for this test. It should be noted that due to the short cabling run typical of power electronics application both latency measurement techniques the latency in communication is dominated by the transmission time over a bandwidth limited channel, as opposed to propagation time, which is completely negligible in this application.

The results of the test are shown in Table 8.1, where the first column shows the utilized forward error correction (FEC) technique, while the subsequent ones show: the absolute amount of latency added by the transmission and reception steps and the total end to end latency. When forward error correction is not needed, namely when an optical physical medium is used, the total amount of latency added to a communication is 910 ns. If some error

FEC Technique		Absolute Latency	Added latency
None	TX	460 ns	910 ns
	RX	450 ns	
Hamming	TX	540 ns	1.11 µs
	RX	570 ns	
Reed Solomon	TX	750 ns	1.82 µs
	RX	1070 ns	

Table 8.1: Latency test results

correction is desired, like when part of the wiring run consists of electrical cabling, the Hamming Single Error Correction Double Error Detection (SECDED) method can be used with a minimal amount of added latency while still being able to detect two bits and correct single bit errors. When heavy interference is expected, the Reed-Solomon error correction algorithm can be employed, allowing the correction of a much longer error sequence, at the cost of a significantly higher added latency.

8.3 Bit Error Rate testing

To evaluate the robustness of studied error correction algorithms, the Bit Error Rate (BER) is calculated as in (8.1), where N_{errors} is the number of errors after error correction, and $N_{tx\ bits}$ is the number of total sent bits

$$BER = \frac{N_{errors}}{N_{tx\ bits}} \tag{8.1}$$

To this purpose a BSC has been used as it can model accurately the conditions that can be found in the real world for power electronics systems [89]. The very simple fully digital hardware receiver architecture is incapable of any kind of error detection, apart from a complete channel failure, and thus the only possible type of error are bit flips. Burst errors are also not considered, as the wide-band noise typically present in solid state energy conversion systems is unlikely to corrupt a significant number of consecutive bits, also

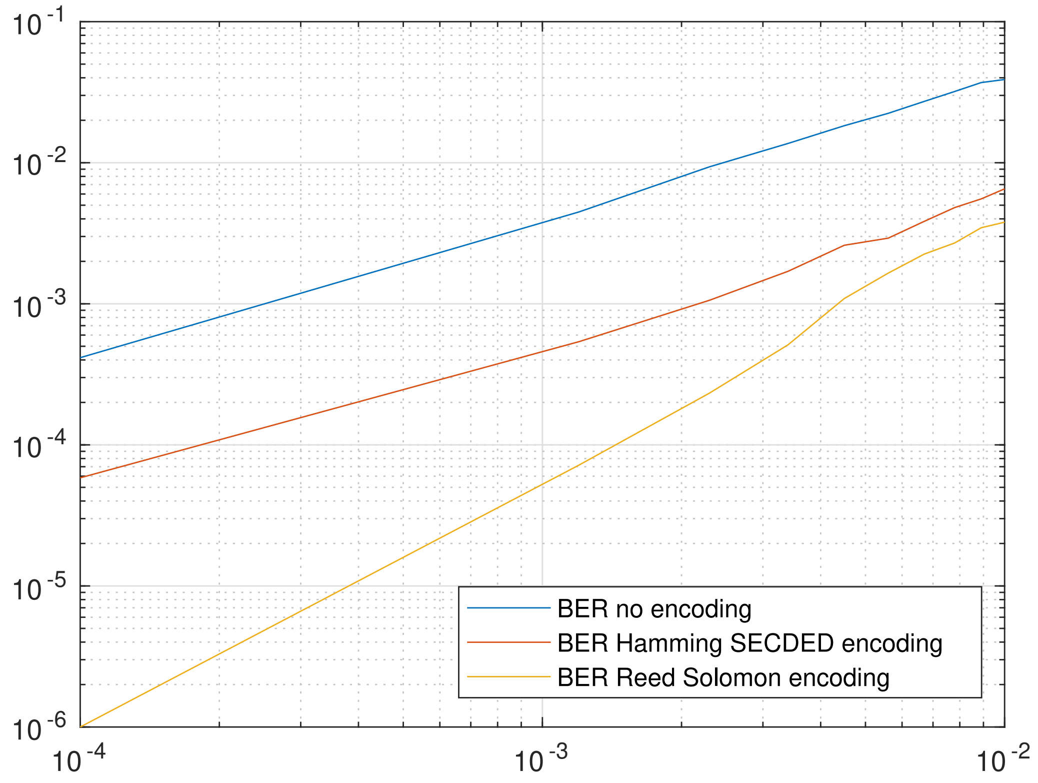


Figure 8.2: BER as a function of error probability for a BSC channel.

taking into account the fairly low frequencies involved.

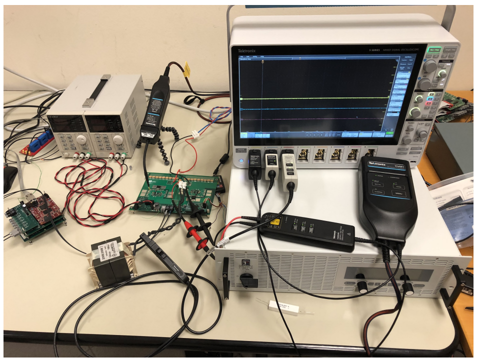
In Fig. 8.2 the measured BER is plotted against the error probability for a BSC channel. This test clearly shows the relative error correcting capabilities of the various error correcting codes. As with the previous test, the results are shown for the two studied FEC techniques, together with the BER for the unprotected transmission, when no error correction is used, as a baseline for the comparison. As expected, the RS coding has better performances across the whole range of channel conditions, however the gap between it and the simpler Hamming coding reduces significantly as the error probability increases indicating a low impact of the error correcting coding on the overall number of errors, in those situations.

It is worth noting that in both cases the BER with error correction is much lower than the one for the uncoded channel, proving the efficacy of FEC even when overwhelmed with more errors than it can handle. For channels with very high error probabilities (due to poor signal-to-noise ratio) the small gap between the two coding methods and relatively large latency of the stronger one, decrease the suitability of RS coding, requiring either a stronger coding, or a change in the physical channel design.

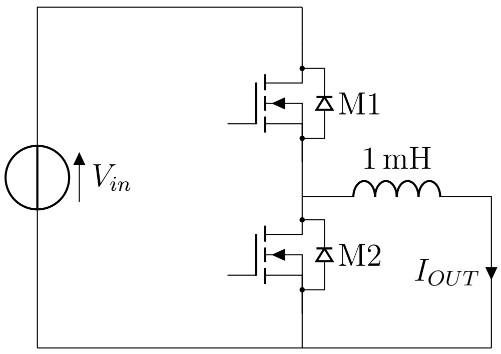
8.4 Power cell Testing

8.4.1 High voltage test

The first step in the experimental validation of the developed system is performance testing of the power cells themselves, establishing a baseline and ranges for operative parameters, like internal gate driver timings. This step is also vital in establishing an acceptance testing procedure that ensures functionality of all power cells upon integration in the final distributed architecture.



(a)



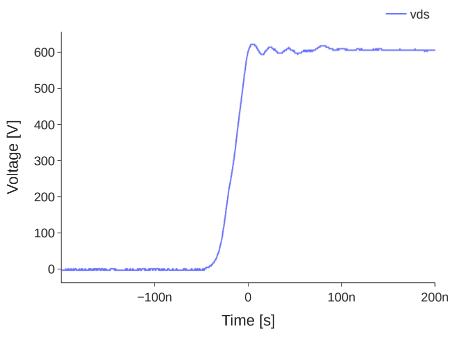
(b)

Figure 8.3: Power cell test experimental setup (a) schematic and (b) picture

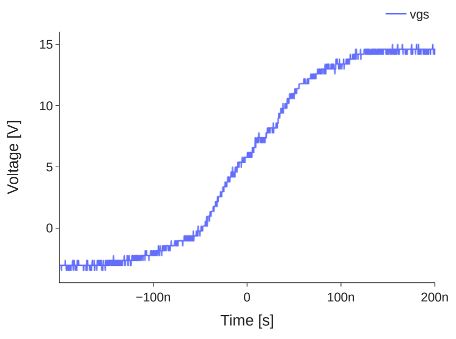
The chosen experimental setup is shown in Fig. 8.3, along with the relative electrical schematic. It contains a single power cell configured and operated as synchronous rectified constant current step-down DC/DC converter with a heavily inductive load. This allows the main switching transistors to operate with both nominal voltage and a significant current, without the need for a load that can dissipate high powers; as well as being representative of a motor winding at standstill, when the back EMF is not present. The first test, was designed to validate signal integrity of the cell, in particular with respect to dv/dt that the wide bandgap devices can generate when commutating, to this effect no additional load was added to the circuit and the maximum input DC voltage of 600 V was used. The high side gate-source voltage was captured with a Tektronix TIVP1 optically isolated high bandwidth differential probe and the low side drain-source voltage was captured with a THDP0200 high voltage differential probe. The results of the test are shown in Fig. 8.4 where, due to symmetry of the circuit operation, only a rinsing edge is displayed to enhance resolution. Both waveforms appear very clean with no overshoot for the high side gate voltage and only minimal one in the cell output, that keeps the maximum voltage stress on the devices well below the acceptable limits, even with a $20kV/\mu s$ output slew rate. It should also be noted that the apparent lack of synchronization between gate and drain voltage waveforms is a result of the high commutation speed achieved by wide bandgap semiconductor devices, which allow the output transition happening almost exclusively during the miller-plateau phase of the commutation.

8.4.2 Low voltage test

To assess the performance of the power cells in a scenario closer to the complete system a second test was performed with a heavy inductive load, and 270V input DC-link voltage subjecting the transistors to a more realistic operating point. To achieve this a current control loop was used, to regulate the output current to the desired level of 16 A. The results of the test, shown in Fig. 8.5, demonstrates a similar behaviour to the previous test, with only a slight increase in the drain-source overshoot, mainly due to the difference



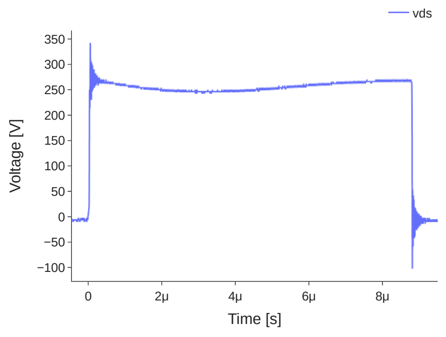
(a)



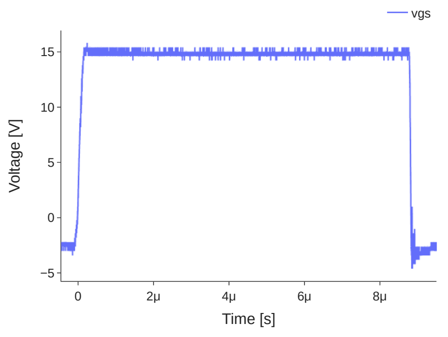
(b)

Figure 8.4: Drain-source (a) and gate-source (b) voltages captured during the 600 V test

in parasitic parameters between the two setups. The high side gate voltage still demonstrates a very clean waveform with no overshoot, important factor in the long term reliability of the main MOSFETs gate oxide layer.



(a)



(b)

Figure 8.5: Drain-source (a) and gate-source (b) voltages captured during the 270 V test

8.4.3 Thermal Test

During the previous test a thermal imaging camera has been used to capture the temperature distribution in the power cell, with the aim of verifying the correct static and dynamic current sharing between all the devices in each switching cluster. The resulting image, shown in Fig. 8.6, shows only a minimal temperature gradient, between devices that can be attributed to the

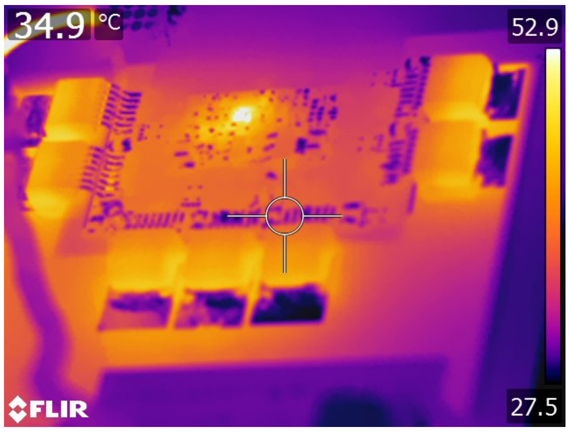


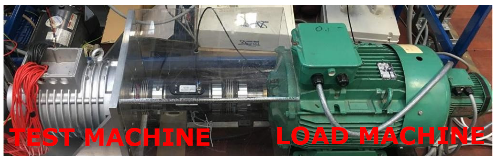
Figure 8.6: Thermal image of the power cell.

difference in heat dissipated through the drain tab, that can be considered normal, given the physical layout of the board and the lack of main heatsink.

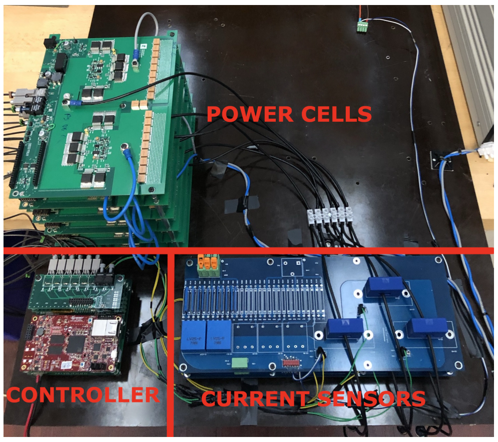
8.5 System level Testing

8.5.1 Experimental Setup

The proposed Drive Architecture is experimentally validated on a configurable multiphase test machine, set up as a permanent magnet synchronous motor, with an asymmetrical stator composed of two three-phase sets radially shifted by 30°. Each one of the six phases is controlled through a separate power cell, fed by a common 270V DC bus. A single centralized main controller, used for simplicity, handles current control, as well as sensor acquisition. A resolver, paired with a monolithic Resolver to Digital Chip (AD2S1210) samples both shaft angle, and true mechanical speed. The output currents are sensed for feedback through isolated closed loop hall effect sensors (LEM LA55P-SP1). An induction machine coupled to the test motor is operated as a variable load. Both machine and drive are shown in Fig. 8.7 and the related specifications in Table 8.2. It is important to note that some differences exist between the initial design parameters, shown in



(a)



(b)

Figure 8.7: Experimental Setup

chapter 3 and those used in the experimental setup. DC-Link voltage is limited to the chosen values by the motor testing rig power supply while rated current/torque was limited by both the test machine and testing rig. Finally, switching frequency is limited to 60 kHz due to limitations in the gate driving circuitry.

Parameter	value
stator inductance	0.125 mH
stator resistance	8.5 mΩ
flux linkage	0.0923 Wb
pole pairs	4
Rated speed (test rig limited)	1500 rpm
Rated current	50 A
Switching frequency	60 kHz
DC link voltage	270 V

Table 8.2: Experimental setup parameters

8.5.2 Data capture and processing

The six phase currents signals from the hall effect sensors are simultaneously sampled and digitized at 240 kSps by six LTC2313-14 14 bit analogue to digital converter. These raw samples are then directly saved to a file, along with the relative timebase to be used in plots and analyses presented later in this chapter. The control loop input signals are derived by these through a downsampling process that reduces data rate by a factor of 4, to arrive at the target sampling frequency.

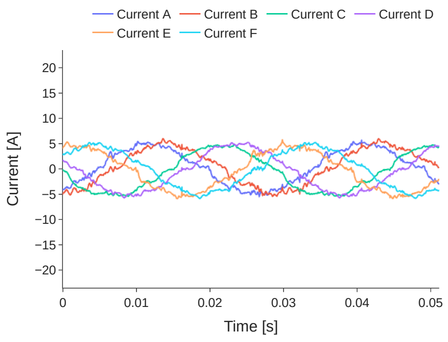
8.5.3 Steady State Test

The first set of tests has been performed with a completely healthy system, working nominally, with the aim of assessing the steady state behaviour of the drive. Throughout these tests a constant current set point was maintained, while varying speed and DC link voltage. The captured data is shown in Fig. 8.8 and Fig. 8.9. From a stability perspective, no signs of oscillations are present, confirming the results of the analysis presented in chapter 6. The envelopes of the multiphase sets demonstrate an excellent steady state tracking, with no visible error.

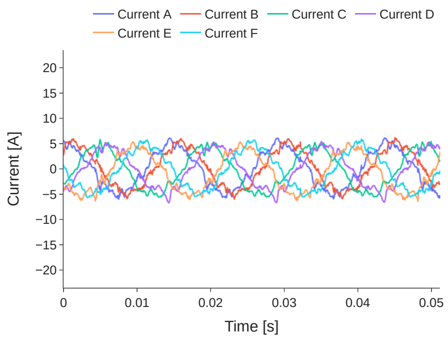
To assess the output power quality, the Total Harmonic Distortion (THD) of the phase current is calculated through a Fast Fourier Transform (FFT) analysis of an integer number of fundamental periods.

The Total Harmonic Distortion of the phase current is chosen as the figure of merit for power quality. For its calculation a Fast Fourier Transform of an integer number of fundamental periods is performed, obtaining frequency domain information. The indicator is then calculated from this equation:

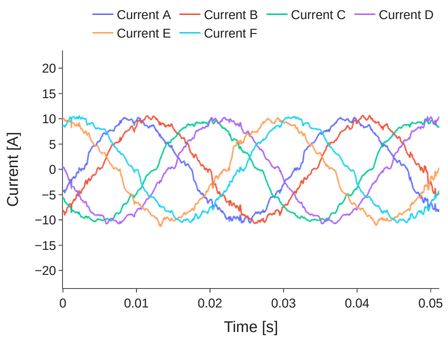
$$THD = \sqrt{\frac{I_3^2 + I_5^2 + ...}{I_f^2}} \tag{8.2}$$



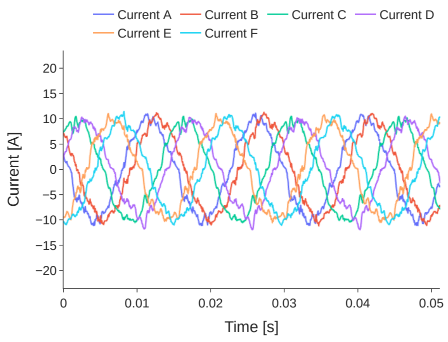
(a)



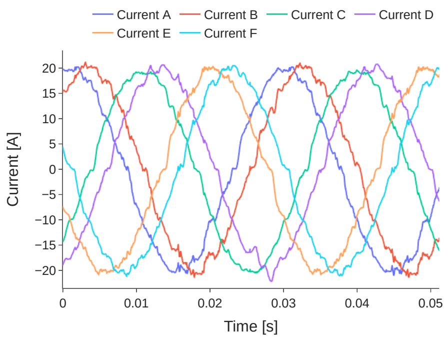
(b)



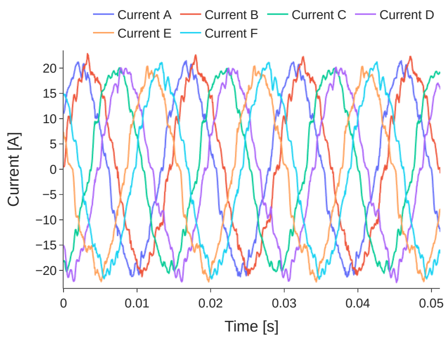
(c)



(d)

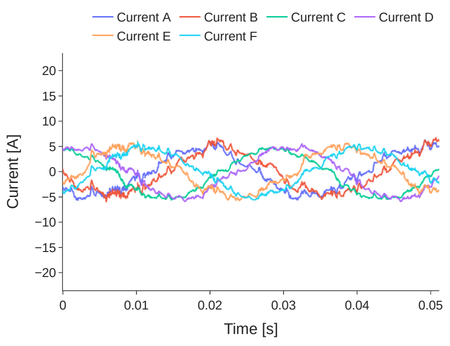


(e)

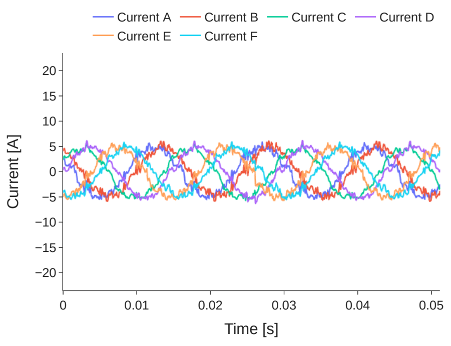


(f)

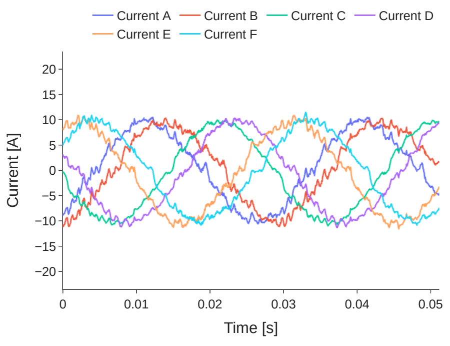
Figure 8.8: Steady state currents at 135V Input voltage and 35 (a, c, e) or 70% (b, d, f) of full speed



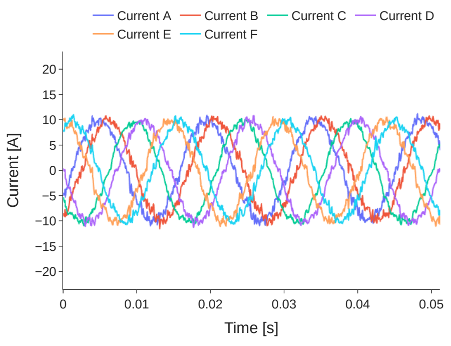
(a)



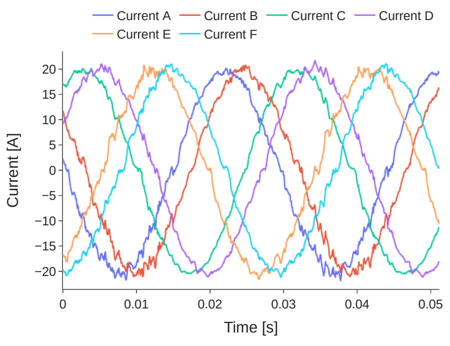
(b)



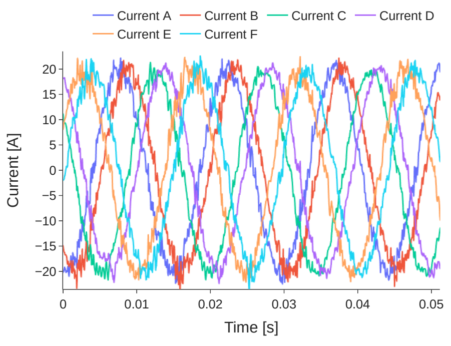
(c)



(d)



(e)



(f)

Figure 8.9: Steady state currents at 270V input voltage and 35 (a, c, e) or 70% (b, d, f) of full speed

	THD				SNR			
Voltage	135 V		270 V		135 V		270 V	
Speed	35%	70%	35%	70%	35%	70%	35%	70%
5 A	5.87%	6.61%	9.25%	5.35%	8.35	6.09	6.34	6.86
10 A	3.64%	4.17%	5.25%	3.18%	11.51	9.91	9.69	12.17
15 A	2.89%	3.37%	2.53%	2.56%	13.16	13.65	13.31	12.97

Table 8.3: THD and SNR at various operating points

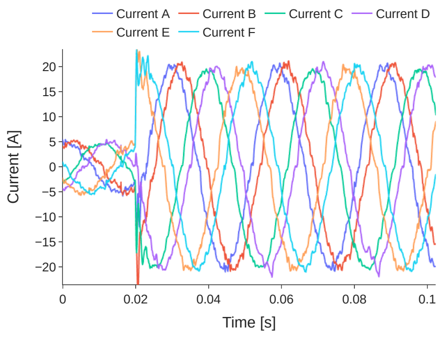
Where I_x^2 represents the amplitude of the x-th harmonic, while I_f^2 is the amplitude of the fundamental.

Another metric calculated to verify the reliability of the obtained THD figure is the Signal to noise ratio:

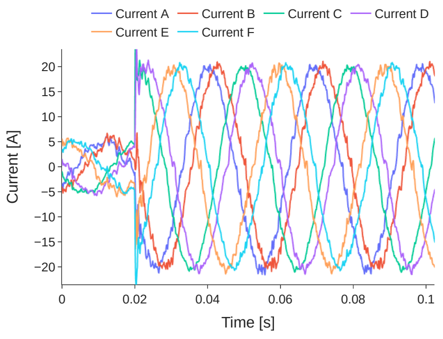
$$SNR = \frac{P_{signal}}{P_{noise}} \tag{8.3}$$

defined as the ratio between the power of the fundamental signal P_{signal} and that of the noise P_{noise}, constituted by all component of the spectrum at other frequencies

The average THD And SNR for all six phases are shown in Table 8.3. As expected the calculated distortion decreases as the current increases, this is due to the vulnerability of sensors, cabling and other circuitry to noise when close to the system resolution limit. A difference in the overall noise-floor is also the cause of the contradicting trends with regard to speed and voltage, where THD increases with speed at 135 V and decreases at 270 V. At the highest current set-point on the other hand, the signal-to-noise ratio for all captures is broadly similar, making the figures directly comparable. There is only a slight decrease in SNR at the highest voltage and speed, which while visible as increased noise on the plots does not impact on the THD figure.



(a)



(b)

Figure 8.10: Load steps at varying input voltage

8.5.4 Load Step Tests

A second key technique used to validate the performance of a drive is the load step, it allows the estimation of the current control bandwidth, and ensure stability even in challenging conditions. The preferred methodology for this test, consisting of driving the machine to a constant speed, while varying the load in a step wise manner could not be followed due to limitation to the external test rig infrastructure. Consequently, the drive under test was directly controlled to step its current set-point with the load machine spinning at a constant speed. To better capture the fast dynamics of the system the Data acquisition module triggering function is used to perform the reference change at a well-defined point in the capture windows allowing both pre- and post-trigger information to be displayed.

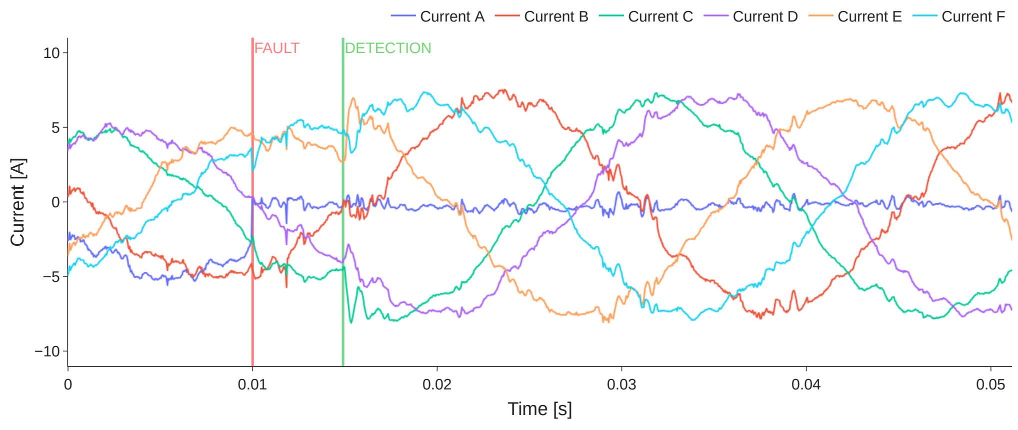
The test, whose results are shown in Fig. 8.10, have been performed at two different voltages with the same speed, going from 5 A to 20 A. In both cases the system reacted immediately to the response with minimal overshoot and no signs of ringing or oscillations. The rise time of the currents in response to the stimulus are in both cases are measured to be 200 µs, pointing to a bandwidth of 1.75 kHz when assuming a first order response.

8.5.5 Fault Transient Tests

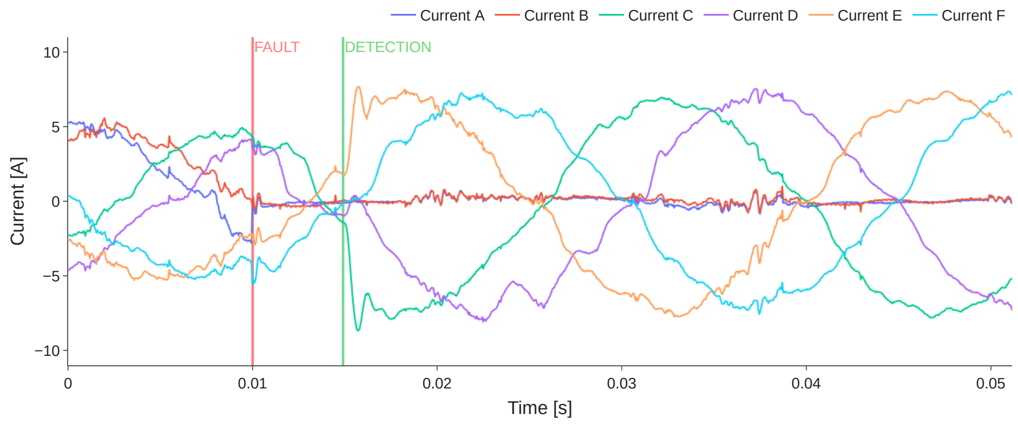
The last and most important test group is aimed at proving the fault tolerance of the proposed architecture. To test this aspect in a safe and scientifically repeatable fashion, an open circuit fault in one or two phases was emulated. To do so, the corresponding power cell was completely turned off, asynchronously with respect to the control, in order to cut the current to zero in the most realistic way possible. Then after a delay the references are reconfigured, with all other controller and test parameter remaining equal. The delay duration, representing, among other things the fault detection algorithm response time and run time, is highly variable, with both the system size and complexity of both system and algorithm, where the accurate determination of the fault type and position in a large highly coupled system can be difficult to perform. Consequently, the value of 5 ms, was chosen, as worst case scenario, highlighting the stability of the proposed control system, even in this situation.

As for the load step test, the fault is triggered through the data collection interface, in order to closely control the events positioning in the captured data buffer.

The results of these trials are shown in Fig. 8.11, at 10 ms into the capture the fault is triggered, immediately upon which the affected currents drop to zero, while the other phases, do not show any sign of instability, only being affected by the neutral point unbalance. After reconfiguration, the currents rapidly shift to follow the new set of references, with a transient behaviour that is very similar to the one observed for the load step.



(a)



(b)

Figure 8.11: Open circuit fault transient test.

8.5.6 Simulation Validation

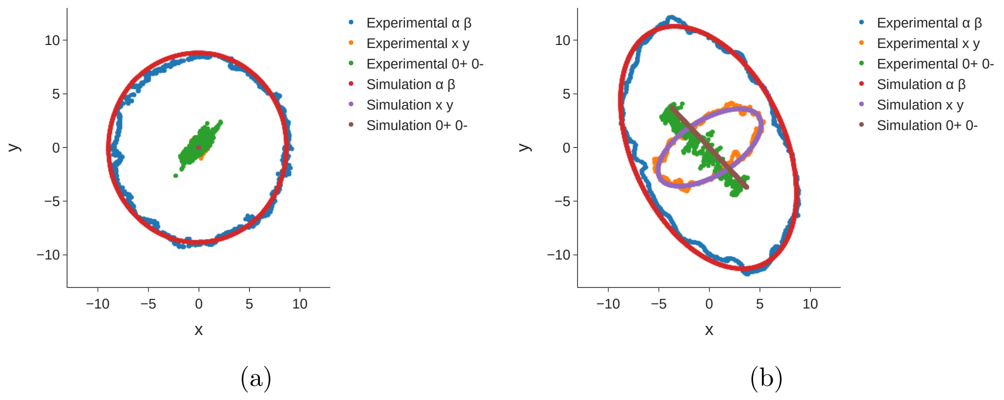


Figure 8.12: Steady state comparison of currents, in VSD space, between simulation (solid line) and experimental results (dashed line).

The data captured during experimental testing, is also used to validate the correctness of the simulations presented in chapter 7. First to validate both the pre- and post-fault steady state tracking performance the VSD components of both sets of currents are shown in Fig. 8.12, highlighting a near perfect matching in both cases.

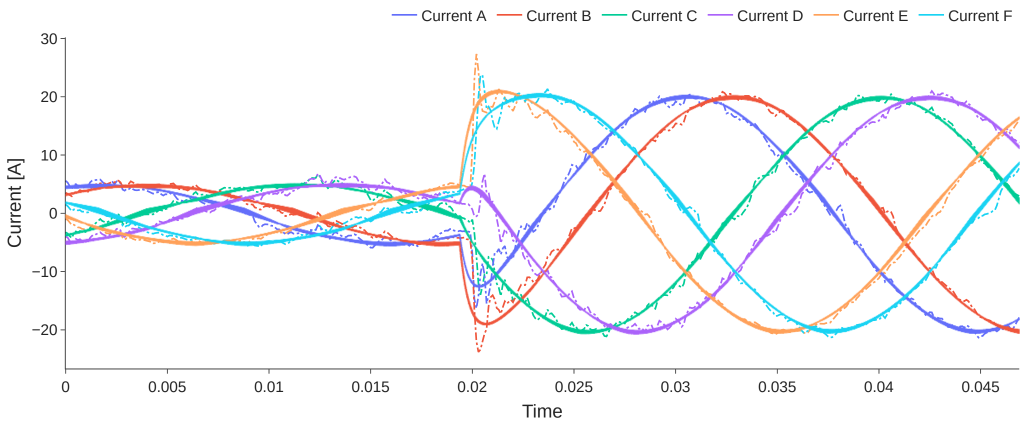


Figure 8.13: Comparison of load step currents between simulation (solid line) and experimental results (dashed line).

The second comparison, shown in Fig. 8.13, shows equally good matching between the experimentally measured currents, plotted with a dashed line, and the simulated currents, shown with a solid line, for the load step test.

This confirms that even the dynamic behaviour of the constructed simulation model is representative of the real system one.

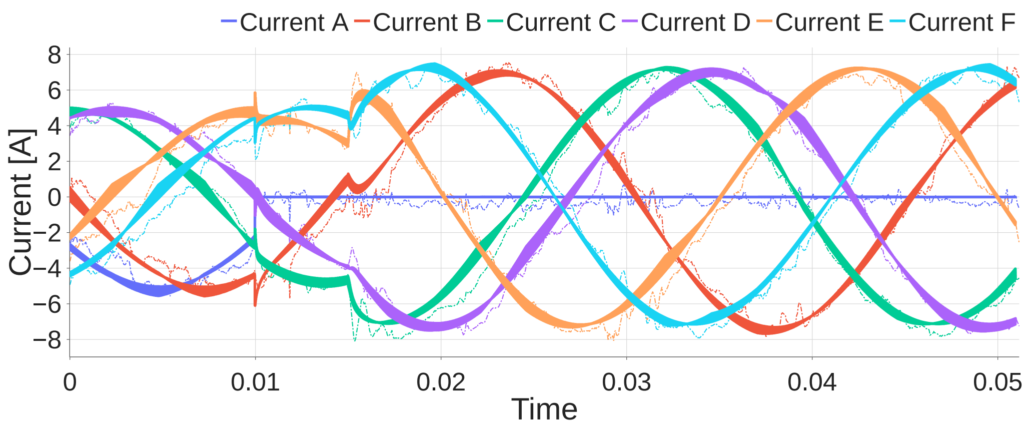


Figure 8.14: Comparison of currents during a fault transient between simulation (solid line) and experimental results (dashed line).

Finally, in Fig. 8.14, simulation (solid line) and experimental results (dashed line) are compared during a fault transient. Even in this tough scenario, the comparison shows excellent matching between simulation and experimental result demonstrating the dependability of the simulation results even for the study of the system response to faults.

The near perfect matching of the system behaviour between analysis, simulation and experimental results, validates all assumptions made during the control system design stage. First and foremost it has been demonstrated that the single phase design approach can be safely and successfully used, neglecting the mutual inductances, that will be dealt with by the robust control strategy as a disturbance, and consequently rejected. This allows a fully independent approach to the control of each phase, where only a locally sensed current information is used, completely removing the need for a single central controller, reducing the need for communications, potentially increasing reliability. The analytical approach to mutual inductance calculation has also been demonstrated as sufficient, not needing to rely

Test Name	Test target	Test goal
Power cell HV	Power cell hardware	Functional safety
Power cell LV	power cell hardware	Operating point output waveform quality
Steady state machine test	Healthy drive system	Functional verification and output distortion
Load step machine test	Healthy drive system	Current control loop bandwidth and stability
Open phase fault transient	faulty drive system	Stability, tracking and power quality during and post fault

Table 8.4: Summary of the conducted tests and relative goals

on FEM modelling. Last but not least the similarity between simulated and experimentally measured currents shows validity and correctness of the physical implementation of the system.

In conclusion Table 8.4 shows an overview of the experimental campaign, showing performed tests, the part of the system they targeted and the goal/key finding of the test.

Chapter 9

Conclusions and future work

This thesis presented a distributed fault-tolerant drive architecture for multiphase machine drives targeted at mission critical applications. To meet the challenges of such a complex environment, a vertically integrated development approach was followed. With clean slate implementation of all components in the system, from the hardware itself, to the software stack, to the control methodology. The holistic approach allows greater focus to be placed on the most challenging aspects of the system. This avoids expenditure of unnecessary efforts on non-critical components.

The first pillar of the developed architecture is the femtoCore embedded DSP processor, which enables software-like control implementation, without sacrificing flexibility and determinism, the two key advantages of FPGA based controls. Moreover, The relatively high clock frequency of 100 MHz and single cycle operation capability, allow this core to outperform a conventional microcontroller based control implementation [90]. All these characteristics make the femtoCore an ideal match for high performance mission critical control systems.

The second key technology that make high frequency distributed power electronics converter architectures possible is the custom communication protocol, co-designed with the rest of the system. The small packet size, coupled with the very low associated processing overhead allows the achievement of a microsecond level end-to-end message latency (1.1 µs), ensuring a very low impact on the control system performance and stability. Moreover, the use of forward error correction techniques significantly improves jitter over noisy channels with respect to the traditional detect and re-transmit strategy.

Finally, the use of static reference frame based PIR controllers, enables seamless horizontal scaling across multiple controllers, allowing the same base architecture to scale from the relatively small 6 phase drive presented in chapter 8 to very large systems with phase counts ranging in the hundreds, as typical of Megawatt scale drives. Another advantage of static reference frame control is the direct control of stator currents which simplifies the graceful handling of fault transients. In these circumstances a simple reference allows the retention of complete control over the remaining phases currents, regulating all components of the output current and guaranteeing optimal torque production even in a post-fault scenario.

The Fault tolerance of the developed architecture is proven through experimental testing on a 18 kW machine, with a hardware emulated drive failure, where the system has proven capable of handling the loss of both one and two phases without showing signs of instability in both the 5 ms fault detection window and ensuing reconfiguration transient. Demonstrating the ability of the designed architecture to handle complex fault scenarios with a hitless recovery.

9.1 Future Works

While the proposed system architecture provides a strong foundation for the development and deployment of high performance fault-tolerant machine drives, it still has some room for improvement in the following areas:

1. The current system implementation relies on the direct connection of analogue input sensing hardware to the main controller, this choice, while simplifying synchronization of the various phases, presents some scalability challenges, both in terms of EM noise susceptibility and cabling complexity. To avoid these issues migration to fully digital smart sensing modules should be considered, with their direct integration in the communication network, allowing hundreds of nodes to be managed by a single controller instance.

2. A two level voltage source based architecture was used for the power-cells in this thesis, this while striking a good balance between cost, complexity, on state losses and switching losses, does present some challenges when coupled with wide bandgap devices, due to the large voltage time gradients potentially degrading the machine stator insulation system. The development of alternatives based on Multi-level topologies can help solve this issue, and broaden the applicability of the developed architecture.

3. The current communication latency is dominated by the transmission time component, due to the limited available channel bandwidth. This could become a bottleneck in large networks where a tree topology is used where one or more routing layers are necessary. To alleviate these issues, offering the option of a higher bandwidth physical layer, could help mitigate this issue. This reduced latency channel could also improve communication latency in smaller systems by up to an

order of magnitude, bringing overall end-to-end delay in line with the monolithic architectures.

4. With the present incarnation of the system being targeted at fully air-gapped deployments, where no connection to untrusted component is possible, no encryption or authentication solution was integrated in the developed communication protocol, to avoid adding any unnecessary latency to the communication. Their addition, as optional features could broaden applicability of the proposed system to a larger class of problems.

List of publications

Conference Publications

- F. Savi, G. Buticchi, C. Gerada, P. Wheeler, D. Barater “Information Technologies for Distributed Machine Drives: An Overview” 2019 IEEE International Electric Machines & Drives Conference (IEMDC), San Diego, USA, 2019, pp. 1805-1809, doi:10.1109/IEMDC.2019.8785232.

- F. Savi, D. Barater, G. Buticchi, P. Wheeler and C. Gerada, “Multi-phase Fault Tolerant Distributed Control Techniques for Integrated Drives Based on Resonant Regulators” IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society, Singapore, 2020, pp. 2702-2707, doi: 10.1109/IECON43393.2020.9254985.

- F. Savi, G. Buticchi, C. Gerada, P. Wheeler and D. Barater, “Wide Bandgap Voltage Source Inverter Design for Automotive Electric Drivetrain” 2018 IEEE International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles & International Transportation Electrification Conference (ESARS-ITEC), Nottingham, UK, 2018, pp. 1-5, doi: 10.1109/ESARS-ITEC.2018.8607561.

- F. Savi, D. Barater, M. Di Nardo, M. Degano and C. Gerada, “A System Level Comparison of Drive Topologies for High Speed Electrical Machines” IECON 2017 - 43rd Annual Conference of the IEEE

Industrial Electronics Society, Beijing, China, 2017, pp. 4357-4362, doi: 10.1109/IECON.2017.8216750.

9.2 Journal Publications

- Filippo Savi, Davide Barater, Mauro Di Nardo, Michele Degano, Chris Gerada, Pat Wheeler, Giampaolo Buticchi, “High-Speed Electric Drives: A Step Towards System Design” in IEEE Open Journal of the Industrial Electronics Society, vol. 1, pp. 10-21, 2020, doi: 10.1109/OJIES.2020.2973883.

- Bianchi, V.; Savi, F.; De Munari, I.; Barater, D.; Buticchi, G.; Franceschini, G. Minimization of Network Induced Jitter Impact on FPGA-Based Control Systems for Power Electronics through Forward Error Correction. Electronics 2020, 9, 281.

- F. Savi, D. Barater, G. Buticchi, C. Gerada and P. Wheeler, "A Scalable System Architecture for High-Performance Fault Tolerant Machine Drives," in IEEE Open Journal of the Industrial Electronics Society, vol. 2, pp. 428-440, 2021, doi: 10.1109/OJIES.2021.3104977.

- F. Savi, J. Harikumaran, D. Barater, G. Buticchi, C. Gerada and P. Wheeler, "Femtocore: An Application Specific Processor for Vertically Integrated High Performance Real-Time Controls," in IEEE Open Journal of the Industrial Electronics Society, vol. 2, pp. 479-488, 2021, doi: 10.1109/OJIES.2021.3112124.

Appendix A

femtoCore ISA

Mnemonic	**Opcode**	**Format**	**Description**	**Operation**
NOP	0	Independent	No operation	$r0 \leftarrow r0 + r0$
ADD	1	Binary	Addition of two registers	$dest \leftarrow A + B$
SUB	2	Binary	Subtraction of two registers	$dest \leftarrow A - B$
MUL	3	Binary	Multiplication of two registers	$dest \leftarrow A - B$
ITF	4	Unary	Convert value from integer to FP	$dest \leftarrow float(operand)$
FTI	5	Unary	Convert value from FP to integer	$dest \leftarrow int(operand)$
LDC	6	Load	load FP constant to register	$dest \leftarrow constant$
CGT	8	Binary	test if A is greater than B	$dest \leftarrow A > B?1 : 0$
CLE	9	Binary	test if A is less than or equal to B	$dest \leftarrow A \leq B?1 : 0$
CEQ	10	Binary	test if A is equal to B	$dest \leftarrow A = B?1 : 0$
CNE	11	Binary	test if A is not equal to B	$dest \leftarrow A > B?1 : 0$
STOP	12	Independent	Conclude the core execution	stop
AND	13	Binary	Logical and between two registers	$dest \leftarrow A\&B$
OR	14	Binary	Logical or between two registers	$dest \leftarrow A|B$
NOT	15	Unary	Logical not between two registers	$dest \leftarrow A|B$
SATP	16	Unary	Saturate in a positive direction	$dest \leftarrow A < lim?A : lim$
SATN	17	Unary	Saturate in a negative direction	$dest \leftarrow A > lim?A : lim$

Appendix B

femtoCore Assembly Grammar

```
grammar fs_grammar;
program : code;
code
: ( instruction |for_block
  | pragma |declaration)+;

declaration
: (input_decl | variable_decl
  | constant_decl | output_decl)
;

instruction
: reg_instr | imm_instr | indep_instr
  | pseudo_instr | branch_instr
  | conv_instr | load_instr;

reg_instr
```

```
: reg_opcode  operand ',' operand ',' destination;
imm_instr : imm_opcode destination ',' immediate;
load_instr
: 'ldc' destination ',' FloatingPointLiteral;
branch_instr
: branch_opcode operand ',' operand ',' operand;
conv_instr : conv_opcode operand ',' operand;
indep_instr : 'stop' | 'nop';

pseudo_instr
: pseudo_opcode
operand ',' operand  (',' operand)*;

operand : Register | Identifier;



destination: Register | Identifier;
immediate :
Integer | Hexnum | Octalnum | Identifier;

float_const : FloatingPointLiteral;

reg_opcode
: 'add' | 'sub' | 'mul' | 'and'
  | 'or' | 'satp' | 'satn';
conv_opcode: 'itf' | 'fti' | 'not' ;
imm_opcode : 'ldr';
branch_opcode: 'ble' | 'bgt' | 'beq' | 'bne';
```

```
pseudo_opcode: 'mov';



for_block
: 'for('for_decl';'for_end';
  '(for_incr|for_dec)')' '{'code'}';
for_incr: Identifier ('++');
for_dec: Identifier ('--');
for_decl: Identifier '=' Integer;

for_end: Identifier for_end_comp_type Integer;
for_end_comp_type : ('<' | '>' | '<=' | '>=');
pragma: '#pragma ' Identifier;



variable_decl : 'let' (Register | Identifier);
constant_decl : 'const' (Register | Identifier);
input_decl : 'input' (Register | Identifier);
output_decl : 'output' (Register | Identifier);

Register: 'r' (Digit +);

Identifier
: Letter ('_' | Letter | Digit)*
;
```

Hexnum

```
: '0x' HexDigit +
;

Integer
: (Digit +)
;

Octalnum
: ('0' .. '7') + ('o' | 'O')
;

fragment HexDigit
: ('0' .. '9' | 'a' .. 'f' | 'A' .. 'F')
;

FloatingPointLiteral
: '-'? ('0' .. '9') + '.' ('0' .. '9')* Exponent?
  | '.' ('0' .. '9') + Exponent?
  | ('0' .. '9') + Exponent
;

fragment Exponent
: ('e' | 'E') ('+' | '-')? ('0' .. '9') +
;

String
: '\'' ('\\' . | ~ ('\\' | '\''))* '\''
;
```

```
fragment Letter
: ('a' .. 'z' | 'A' .. 'Z')
;

fragment Digit
: '0' .. '9'
;

Label
: Identifier (':')
;

WS
: (' ' | '\t' | '\n' | '\r') -> skip
;

BlockComment
:     '/*' .*? '*/'
;

LineComment
:     '//' ~[\r\n]*
;
```

Bibliography

[1] C. F. Keller. "Global warming: a review of this mostly settled issue". In: *Stochastic Environmental Research and Risk Assessment* 23.5 (2009), pp. 643–676. ISSN: 1436-3259. DOI: 10.1007/s00477-008-0253-3. URL: https://doi.org/10.1007/s00477-008-0253-3.

[2] "State of the global climate 2020". In: (2020), 38 p. :

[3] "2020. Tracking SDG 7: The Energy Progress Report". In: (2020).

[4] K. Keramidas et al. "Global Energy and Climate Outlook 2019: Electrification for the low-carbon transition The role of electrification in low-carbon pathways, with a global and regional focus on EU and China". In: (Mar. 2020). DOI: 10.13140/RG.2.2.23781.35043.

[5] C. Bataille et al. "A review of technology and policy deep decarbonization pathway options for making energy-intensive industry production consistent with the Paris Agreement". In: *Journal of Cleaner Production* 187 (2018), pp. 960–973. ISSN: 0959-6526. DOI: https://doi.org/10.1016/j.jclepro.2018.03.107. URL: https://www.sciencedirect.com/science/article/pii/S0959652618307686.

[6] D. Gielen et al. "Renewables-based decarbonization and relocation of iron and steel making: A case study". In: *Journal of Industrial Ecology* 24.5 (2020), pp. 1113–1125. DOI: https://doi.org/10.1111/jiec.12997.

[7] R. C. Pietzcker et al. "Long-term transport energy demand and climate policy: Alternative visions on transport decarbonization in energy-economy models". In: *Energy* 64 (2014), pp. 95–108. ISSN: 0360-5442. DOI: `https://doi.org/10.1016/j.energy.2013.08.059`.

[8] "DECARBONISATION ROAD-MAP: A PATH TO NET ZERO A plan to decarbonise UK aviation". In: (2020).

[9] "The China Civil Aviation (CCA) 14th 5-year Plan". In: (2021).

[10] B. Sarlioglu and C. T. Morris. "More Electric Aircraft: Review, Challenges, and Opportunities for Commercial Transport Aircraft". In: *IEEE Transactions on Transportation Electrification* 1.1 (2015), pp. 54–64. DOI: `10.1109/TTE.2015.2426499`.

[11] V. Madonna, P. Giangrande, and M. Galea. "Electrical Power Generation in Aircraft: Review, Challenges, and Opportunities". In: *IEEE Transactions on Transportation Electrification* 4.3 (2018), pp. 646–659. DOI: `10.1109/TTE.2018.2834142`.

[12] W. Cao et al. "Overview of Electric Motor Technologies Used for More Electric Aircraft (MEA)". In: *IEEE Transactions on Industrial Electronics* 59.9 (Sept. 2012), pp. 3523–3531. ISSN: 1557-9948. DOI: `10.1109/TIE.2011.2165453`.

[13] V. C. Cavalcanti and C. R. de Andrade. "A Trade-off Study of a Bleedless and Conventional Air Conditioning Systems". In: *2008 SAE Brasil Congress and Exhibit*. SAE International, Oct. 2008. DOI: `https://doi.org/10.4271/2008-36-0001`. URL: `https://doi.org/10.4271/2008-36-0001`.

[14] F. Berg et al. "HTS Electrical System for a Distributed Propulsion Aircraft". In: *IEEE Transactions on Applied Superconductivity* 25.3 (2015), pp. 1–5. DOI: `10.1109/TASC.2014.2384731`.

[15] J. L. Felder et al. “Turboelectric distributed propulsion in a hybrid wing body aircraft”. In: (2011).

[16] J. K. Nøland et al. “High-Power Machines and Starter-Generator Topologies for More Electric Aircraft: A Technology Outlook”. In: *IEEE Access* 8 (2020), pp. 130104–130123. DOI: 10.1109/ACCESS.2020.3007791.

[17] V. Raveendran, M. Andresen, and M. Liserre. “Improving Onboard Converter Reliability for More Electric Aircraft With Lifetime-Based Control”. In: *IEEE Transactions on Industrial Electronics* 66.7 (2019), pp. 5787–5796. DOI: 10.1109/TIE.2018.2889626.

[18] A. Raciti et al. “State of the art and emerging solid-state power devices in the perspective of more electric aircraft”. In: *2018 AEIT International Annual Conference*. 2018, pp. 1–6. DOI: 10.23919/AEIT.2018.8577345.

[19] H. Wang et al. “Transitioning to Physics-of-Failure as a Reliability Driver in Power Electronics”. In: *IEEE Journal of Emerging and Selected Topics in Power Electronics* 2.1 (2014), pp. 97–114. DOI: 10.1109/JESTPE.2013.2290282.

[20] M.J. Cushing et al. “Comparison of electronics-reliability assessment approaches”. In: *IEEE Transactions on Reliability* 42.4 (1993), pp. 542–546. DOI: 10.1109/24.273574.

[21] G. Buticchi, L. Costa, and M. Liserre. “Improving System Efficiency for the More Electric Aircraft: A Look at dcdc Converters for the Avionic Onboard dc Microgrid”. In: *IEEE Industrial Electronics Magazine* 11.3 (2017), pp. 26–36. DOI: 10.1109/MIE.2017.2723911.

[22] J. W. Bennett et al. “Safety-critical design of electromechanical actuation systems in commercial aircraft”. English. In: *IET Electric Power Applications* 5.1 (Jan. 2011), pp. 37–47.

[23] G. Buticchi et al. “On-Board Microgrids for the More Electric Aircraft—Technology Review”. In: *IEEE Transactions on Industrial Electronics* 66.7 (2019), pp. 5588–5599. DOI: 10.1109/TIE.2018.2881951.

[24] R.V. White and F.M. Miles. “Principles of fault tolerance”. In: *Proceedings of Applied Power Electronics Conference. APEC ’96*. Vol. 1. 1996, 18–25 vol.1. DOI: 10.1109/APEC.1996.500416.

[25] S. Zhu et al. “Design Considerations of Fault-Tolerant Electromechanical Actuator Systems for More Electric Aircraft (MEA)”. In: *2018 IEEE Energy Conversion Congress and Exposition (ECCE)*. 2018, pp. 4607–4613. DOI: 10.1109/ECCE.2018.8557426.

[26] P. Giangrande et al. “Considerations on the Development of an Electric Drive for a Secondary Flight Control Electromechanical Actuator”. In: *IEEE Transactions on Industry Applications* 55.4 (2019), pp. 3544–3554. DOI: 10.1109/TIA.2019.2907231.

[27] N. Ertugrul et al. “Fault tolerant motor drive system with redundancy for critical applications”. In: *2002 IEEE 33rd Annual IEEE Power Electronics Specialists Conference. Proceedings (Cat. No.02CH37289)*. Vol. 3. 2002, 1457–1462 vol.3. DOI: 10.1109/PSEC.2002.1022381.

[28] A. Galassini et al. “A Modular Speed-Drooped System for High Reliability Integrated Modular Motor Drives”. In: *IEEE Transactions on Industry Applications* 52.4 (2016), pp. 3124–3132. DOI: 10.1109/TIA.2016.2540608.

[29] S. Rubino et al. “Asymmetrical twelve-phase induction starter/generator for more electric engine in aircraft”. In: *2016 IEEE Energy*

Conversion Congress and Exposition (ECCE). 2016, pp. 1–8. DOI: 10.1109/ECCE.2016.7854889.

[30] W. Cao et al. "Overview of Electric Motor Technologies Used for More Electric Aircraft (MEA)". In: *IEEE Transactions on Industrial Electronics* 59.9 (2012), pp. 3523–3531. DOI: 10.1109/TIE.2011.2165453.

[31] L. de Lillo et al. "Multiphase Power Converter Drive for Fault-Tolerant Machine Development in Aerospace Applications". In: *IEEE Transactions on Industrial Electronics* 57.2 (2010), pp. 575–583. DOI: 10.1109/TIE.2009.2036026.

[32] T. M. Jahns. "Improved Reliability in Solid-State AC Drives by Means of Multiple Independent Phase Drive Units". In: *IEEE Transactions on Industry Applications* IA-16.3 (1980), pp. 321–331. DOI: 10.1109/TIA.1980.4503793.

[33] J.A. Haylock et al. "Operation of a fault tolerant PM drive for an aerospace fuel pump application". In: *1997 Eighth International Conference on Electrical Machines and Drives (Conf. Publ. No. 444)*. 1997, pp. 133–137. DOI: 10.1049/cp:19971053.

[34] J. W. Bennett et al. "Fault-Tolerant Design Considerations and Control Strategies for Aerospace Drives". In: *IEEE Transactions on Industrial Electronics* 59.5 (2012), pp. 2049–2058. DOI: 10.1109/TIE.2011.2159356.

[35] T. Sadeghi and A. Lyons. "Fault tolerant EHA architectures". In: *IEEE Aerospace and Electronic Systems Magazine* 7.3 (1992), pp. 32–42. DOI: 10.1109/62.141899.

[36] P. J. Tavner and J. P. Hasson. "Predicting the design life of high integrity rotating electrical machines". In: *1999 Ninth International*

Conference on Electrical Machines and Drives. 1999, pp. 286–290. DOI: 10.1049/cp:19991036.

[37] X. Xue et al. "Design of Five-Phase Modular Flux-Switching Permanent-Magnet Machines for High Reliability Applications". In: *IEEE Transactions on Magnetics* 49.7 (2013), pp. 3941–3944. DOI: 10.1109/TMAG.2013.2244201.

[38] G. Zhang et al. "Design and Comparison of Two Six-Phase Hybrid-Excited Flux-Switching Machines for EV/HEV Applications". In: *IEEE Transactions on Industrial Electronics* 63.1 (2016), pp. 481–493. DOI: 10.1109/TIE.2015.2447501.

[39] H.S. Che et al. "Postfault Operation of an Asymmetrical Six-Phase Induction Machine With Single and Two Isolated Neutral Points". In: *IEEE Transactions on Power Electronics* 29.10 (2014), pp. 5406–5416. DOI: 10.1109/TPEL.2013.2293195.

[40] N. M. A. Freire and A. J. M. Cardoso. "A Fault-Tolerant PMSG Drive for Wind Turbine Applications With Minimal Increase of the Hardware Requirements". In: *IEEE Transactions on Industry Applications* 50.3 (2014), pp. 2039–2049. DOI: 10.1109/TIA.2013.2282935.

[41] R. L. de Araujo Ribeiro et al. "Fault-tolerant voltage-fed PWM inverter AC motor drive systems". In: *IEEE Transactions on Industrial Electronics* 51.2 (2004), pp. 439–446. DOI: 10.1109/TIE.2004.825284.

[42] C. C. Yeh and N. A. O. Demerdash. "Induction Motor-Drive Systems with Fault Tolerant Inverter-Motor Capabilities". In: *2007 IEEE International Electric Machines Drives Conference*. Vol. 2. 2007, pp. 1451–1458. DOI: 10.1109/IEMDC.2007.383642.

[43] N. Bianchi, E. Fornasiero, and S. Bolognani. "Performance of Five-phase Motor Drive under Post-fault Operations". In: *Electric Power*

Components and Systems 39.12 (2011), pp. 1302–1314. DOI: 10.1080/15325008.2011.567221.

[44] H. Guzman, F. Barrero, and M. J. Duran. "IGBT-Gating Failure Effect on a Fault-Tolerant Predictive Current-Controlled Five-Phase Induction Motor Drive". In: *IEEE Transactions on Industrial Electronics* 62.1 (2015), pp. 15–20. DOI: 10.1109/TIE.2014.2331019.

[45] M. D. Hennen et al. "Development and Control of an Integrated and Distributed Inverter for a Fault Tolerant Five-Phase Switched Reluctance Traction Drive". In: *IEEE Transactions on Power Electronics* 27.2 (2012), pp. 547–554. DOI: 10.1109/TPEL.2011.2132763.

[46] Y. Zhao and T. A. Lipo. "Space vector PWM control of dual three-phase induction machine using vector space decomposition". In: *IEEE Transactions on Industry Applications* 31.5 (1995), pp. 1100–1109. DOI: 10.1109/28.464525.

[47] G. K. Singh, K. Nam, and S. K. Lim. "A simple indirect field-oriented control scheme for multiphase induction machine". In: *IEEE Transactions on Industrial Electronics* 52.4 (2005), pp. 1177–1184. DOI: 10.1109/TIE.2005.851593.

[48] E. Levi et al. "Modeling, Control, and Experimental Investigation of a Five-Phase Series-Connected Two-Motor Drive With Single Inverter Supply". In: *IEEE Transactions on Industrial Electronics* 54.3 (2007), pp. 1504–1516. DOI: 10.1109/TIE.2007.894694.

[49] H. S. Che et al. "Current Control Methods for an Asymmetrical Six-Phase Induction Motor Drive". In: *IEEE Transactions on Power Electronics* 29.1 (2014), pp. 407–417. DOI: 10.1109/TPEL.2013.2248170.

[50] I. G. Prieto et al. "Field-Oriented Control of Multiphase Drives With Passive Fault Tolerance". In: *IEEE Transactions on Industrial Electronics* 67.9 (Sept. 2020), pp. 7228–7238. ISSN: 1557-9948. DOI: 10.1109/TIE.2019.2944056.

[51] M. Bermudez et al. "Open-Phase Fault-Tolerant Direct Torque Control Technique for Five-Phase Induction Motor Drives". In: *IEEE Transactions on Industrial Electronics* 64.2 (2017), pp. 902–911. DOI: 10.1109/TIE.2016.2610941.

[52] I. Jlassi and A. J. M. Cardoso. "Fault-Tolerant Back-to-Back Converter for Direct-Drive PMSG Wind Turbines Using Direct Torque and Power Control Techniques". In: *IEEE Transactions on Power Electronics* 34.11 (2019), pp. 11215–11227. DOI: 10.1109/TPEL.2019.2897541.

[53] B. Sen and J. Wang. "Stationary Frame Fault-Tolerant Current Control of Polyphase Permanent-Magnet Machines Under Open-Circuit and Short-Circuit Faults". In: *IEEE Transactions on Power Electronics* 31.7 (2016), pp. 4684–4696. DOI: 10.1109/TPEL.2015.2478337.

[54] A. Tani et al. "Control of Multiphase Induction Motors With an Odd Number of Phases Under Open-Circuit Phase Faults". In: *IEEE Transactions on Power Electronics* 27.2 (2012), pp. 565–577. DOI: 10.1109/TPEL.2011.2140334.

[55] M. D. Assunção et al. "Big Data computing and clouds: Trends and future directions". In: *Journal of Parallel and Distributed Computing* 79-80 (2015). Special Issue on Scalable Systems for Big Data Management and Analytics, pp. 3–15. ISSN: 0743-7315. DOI: https://doi.org/10.1016/j.jpdc.2014.08.003.

[56] J. Jiang and Y. Qian. "Distributed Communication Architecture for Smart Grid Applications". In: *IEEE Communications Magazine* 54.12 (2016), pp. 60–67. DOI: 10.1109/MCOM.2016.1600321CM.

[57] V. Nasirian et al. "Distributed Cooperative Control of DC Microgrids". In: *IEEE Transactions on Power Electronics* 30.4 (2015), pp. 2288–2303. DOI: 10.1109/TPEL.2014.2324579.

[58] S. Debnath et al. "Operation, Control, and Applications of the Modular Multilevel Converter: A Review". In: *IEEE Transactions on Power Electronics* 30.1 (2015), pp. 37–53. DOI: 10.1109/TPEL.2014.2309937.

[59] H. Tu and S. Lukic. "Comparative study of PES Net and SyCCo bus: Communication protocols for modular multilevel converter". In: *2017 IEEE Energy Conversion Congress and Exposition (ECCE)*. 2017, pp. 1487–1492. DOI: 10.1109/ECCE.2017.8095966.

[60] C. L. Toh and L. E. Norum. "Implementation of high speed control network with fail-safe control and communication cable redundancy in modular multilevel converter". In: *2013 15th European Conference on Power Electronics and Applications (EPE)*. 2013, pp. 1–10. DOI: 10.1109/EPE.2013.6631825.

[61] J. Fey et al. "Development of a Modular Multilevel Converter Demonstrator with EtherCAT Communication". In: *2019 IEEE 13th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG)*. 2019, pp. 1–6. DOI: 10.1109/CPE.2019.8862424.

[62] S. Rietmann et al. "Field Bus for Data Exchange and Control of Modular Power Electronic Systems with High Synchronisation Accuracy". In: *2018 International Power Electronics Conference (IPEC-Niigata*

2018 -ECCE Asia). 2018, pp. 2301–2308. DOI: 10.23919/IPEC.2018.8507631.

[63] C. Ditmanson and S. Kolb. “A distributed and fault-tolerant control system for a new modular wind turbine converter”. In: *2014 16th European Conference on Power Electronics and Applications*. 2014, pp. 1–8. DOI: 10.1109/EPE.2014.6910973.

[64] B. Steinmann, G. Fernandez, and N. Cherix. “Tree-shaped networked control system for modular power converters with sub-µs latency and ns-scale synchronization accuracy”. In: *2019 IEEE Energy Conversion Congress and Exposition (ECCE)*. 2019, pp. 6775–6782. DOI: 10.1109/ECCE.2019.8912155.

[65] *FIBER OPTIC CONNECTORS*. Standard. Cedar Rapids, USA: Aeronautical Radio Incorporated, June 2013.

[66] R. T. Hofmeister et al. “Distributed slot synchronization (DSS): a network-wide slot synchronization technique for packet-switched optical networks”. In: vol. 16. 12. 1998, pp. 2109–2116. DOI: 10.1109/50.736579.

[67] V. Kamchevska et al. “Synchronization in a random length ring network for SND-controlled optical TDM switching”. In: vol. 9. 1. 2017, A26–A34. DOI: 10.1364/JOCN.9.000A26.

[68] L. Gao et al. “Effect of alloying elements on properties and microstructures of SnAgCu solders”. In: *Microelectronic Engineering* 87.11 (2010), pp. 2025–2034. ISSN: 0167-9317. DOI: https://doi.org/10.1016/j.mee.2010.04.007. URL: https://www.sciencedirect.com/science/article/pii/S0167931710001322.

[69] A.C.K. So and Y.C. Chan. “Reliability studies of surface mount solder joints - effect of Cu-Sn intermetallic compounds”. In: *IEEE Transac-*

tions on Components, Packaging, and Manufacturing Technology: Part B 19.3 (1996), pp. 661–668. DOI: 10.1109/96.533909.

[70] F. Savi et al. “High-Speed Electric Drives: A Step Towards System Design”. In: *IEEE Open Journal of the Industrial Electronics Society* 1 (2020), pp. 10–21. DOI: 10.1109/OJIES.2020.2973883.

[71] D. G. Holmes and T. A. Lipo. “Modulation of ThreePhase Voltage Source Inverters”. In: *Pulse Width Modulation for Power Converters: Principles and Practice*. 2003, pp. 215–258. DOI: 10.1109/9780470546284.ch5.

[72] P. A. Dahono, Y. Sato, and T. Kataoka. “Analysis and minimization of ripple components of input current and voltage of PWM inverters”. In: *IAS ’95. Conference Record of the 1995 IEEE Industry Applications Conference Thirtieth IAS Annual Meeting*. Vol. 3. 1995, 2444–2450 vol.3. DOI: 10.1109/IAS.1995.530614.

[73] V. Lešić et al. “Field-oriented control of an induction machine with winding asymmetries”. In: *2012 15th International Power Electronics and Motion Control Conference (EPE/PEMC)*. 2012, LS7b-1.2-1-LS7b-1.2–7. DOI: 10.1109/EPEPEMC.2012.6397506.

[74] R. Bojoi et al. “Dual three-phase induction motor drive with digital current control in the stationary reference frame”. In: *Power Engineer* 20.3 (2006), pp. 40–43. DOI: 10.1049/pe:20060308.

[75] H. S. Che et al. “Current Control Methods for an Asymmetrical Six-Phase Induction Motor Drive”. In: *IEEE Transactions on Power Electronics* 29.1 (2014), pp. 407–417. DOI: 10.1109/TPEL.2013.2248170.

[76] L. F. A. Pereira and A. S. Bazanella. “Tuning Rules for Proportional Resonant Controllers”. In: *IEEE Transactions on Control Systems*

Technology 23.5 (2015), pp. 2010–2017. DOI: 10.1109/TCST.2015.2389655.

[77] S. Silwal and M. Karimi-Ghartemani. "On the design of proportional resonant controllers for single-phase grid-connected inverters". In: *2016 12th IEEE International Conference on Control and Automation (ICCA)*. 2016, pp. 797–803. DOI: 10.1109/ICCA.2016.7505376.

[78] W. Lenwari, M. Sumner, and P. Zanchetta. "The Use of Genetic Algorithms for the Design of Resonant Compensators for Active Filters". In: *IEEE Transactions on Industrial Electronics* 56.8 (2009), pp. 2852–2861. DOI: 10.1109/TIE.2009.2018535.

[79] S. Sumsurooah, M. Odavic, and S. Bozhko. "μ Approach to Robust Stability Domains in the Space of Parametric Uncertainties for a Power System With Ideal CPL". In: *IEEE Transactions on Power Electronics* 33.1 (2018). DOI: 10.1109/TPEL.2017.2668900.

[80] D. Paulus, J. F. Stumper, and R. Kennel. "Sensorless Control of Synchronous Machines Based on Direct Speed and Position Estimation in Polar Stator-Current Coordinates". In: *IEEE Transactions on Power Electronics* 28.5 (2013), pp. 2503–2513. DOI: 10.1109/TPEL.2012.2211384.

[81] Y. Park and S. K. Sul. "Sensorless Control Method for PMSM Based on Frequency-Adaptive Disturbance Observer". In: *IEEE Journal of Emerging and Selected Topics in Power Electronics* 2.2 (2014), pp. 143–151. DOI: 10.1109/JESTPE.2013.2296596.

[82] M. J. Duran et al. "A Simple, Fast, and Robust Open-Phase Fault Detection Technique for Six-Phase Induction Motor Drives". In: *IEEE Transactions on Power Electronics* 33.1 (2018), pp. 547–557. DOI: 10.1109/TPEL.2017.2670924.

[83] M. Riera-Guasp, J. A. Antonino-Daviu, and G. Capolino. “Advances in Electrical Machine, Power Electronic, and Drive Condition Monitoring and Fault Detection: State of the Art”. In: *IEEE Transactions on Industrial Electronics* 62.3 (2015), pp. 1746–1759. DOI: 10.1109/TIE.2014.2375853.

[84] M. Naidu, S. Gopalakrishnan, and T. W. Nehl. “Fault-Tolerant Permanent Magnet Motor Drive Topologies for Automotive X-By-Wire Systems”. In: *IEEE Transactions on Industry Applications* 46.2 (2010), pp. 841–848. DOI: 10.1109/TIA.2009.2039982.

[85] B. A. Welchko et al. “Fault tolerant three-phase AC motor drive topologies: a comparison of features, cost, and limitations”. In: *IEEE Transactions on Power Electronics* 19.4 (2004), pp. 1108–1116. DOI: 10.1109/TPEL.2004.830074.

[86] A. Mohammadpour and L. Parsa. “Global Fault-Tolerant Control Technique for Multiphase Permanent-Magnet Machines”. In: *IEEE Transactions on Industry Applications* 51.1 (2015), pp. 178–186. DOI: 10.1109/TIA.2014.2326084.

[87] J. Fu and T. A. Lipo. “Disturbance free operation of a multiphase current regulated motor drive with an opened phase”. In: *Conference Record of the 1993 IEEE Industry Applications Conference Twenty-Eighth IAS Annual Meeting*. 1993, 637–644 vol.1. DOI: 10.1109/IAS.1993.298884.

[88] X. Wang et al. “Selective Torque Harmonic Elimination for Dual Three-Phase PMSMs Based on PWM Carrier Phase Shift”. In: *IEEE Transactions on Power Electronics* 35.12 (2020), pp. 13255–13269. DOI: 10.1109/TPEL.2020.2991264.

[89] T. C. Maxino and P. J. Koopman. “The Effectiveness of Checksums for Embedded Control Networks”. In: *IEEE Transactions on Dependable and Secure Computing* 6.1 (Jan. 2009), pp. 59–72. ISSN: 1545-5971. DOI: 10.1109/TDSC.2007.70216.

[90] Filippo Savi et al. “Femtocore: An Application Specific Processor for Vertically Integrated High Performance Real-Time Controls”. In: *IEEE Open Journal of the Industrial Electronics Society* 2 (2021), pp. 479–488. DOI: 10.1109/OJIES.2021.3112124.