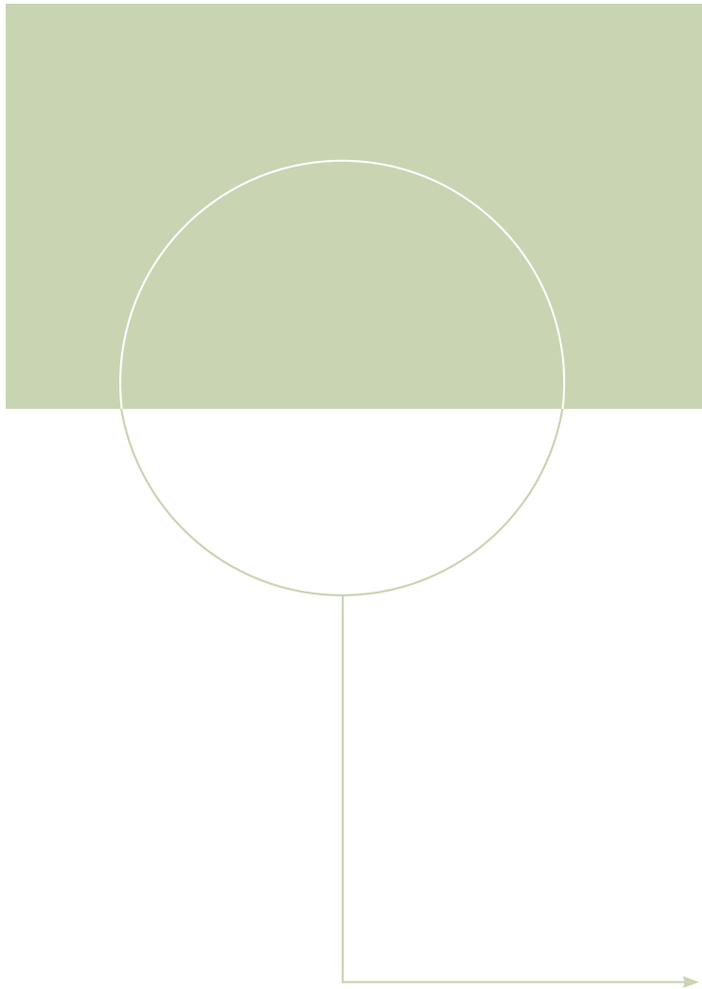


Master's thesis

2023

Alexander Abele

NTNU
Norwegian University of
Science and Technology
Faculty of Information Technology and Electrical
Engineering
Department of Engineering Cybernetics

Alexander Abele

Developing a digital twin for safety demonstration

A case study related to an all-eletrical safety valve

January 2023

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

NTNU

Norwegian University of
Science and Technology

Developing a digital twin for safety demonstration

A case study related to an all-eletrical safety valve

Alexander Abele

Mechatronic / Systems engineering

Submission date: January 2023

Supervisor: Prof. Dr. Mary Ann Lundteigen

Co-supervisor: Ludvig Björklund

Prof. Dr. Markus Glaser

Norwegian University of Science and Technology

Department of Engineering Cybernetics

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Developing a digital twin for safety demonstration

A case study related to an all-electric safety valve

handed in by

Alexander Abele
Student number: 584810

Main supervisor:	Prof. Dr. Mary Ann Lundteigen
Co-supervisor:	Ludvig Björklund
Co-supervisor:	Prof. Dr. Markus Glaser
processing time:	22.08.2022 – 15.01.2023

Preface

This is the master's thesis in Mechatronic / Systems Engineering from Aalen University. It was carried out in the autumn semester of 2022 at the cybernetics department at NTNU, which was possible through the ERASMUS+ program. My professor proposed the thesis at Aalen University as a way to write the thesis abroad and to contribute to a PhD research project in Norway. The thesis is also interesting for people working and researching the shift from electro-hydraulic actuators to all-electric subsea safety valves.

Trondheim, 2023-01-12

Alexander Abele

Acknowledgement

This thesis ends the three-semester master's degree in Mechatronic / Systems Engineering at Aalen University in germany.

Firstly, I would like to thank my family for always being with me during this time and in my studies. Thank you.

Secondly, I would like to thank my professor, Dr Markus Glaser, from Aalen University, who made it possible, so that I had the opportunity to write my master's thesis in Norway at NTNU and could see one of the most beautiful countries and cultures I have seen in my life. Thanks to this opportunity, I decided to continue living abroad, here in Norway. Thank you for this opportunity and the support you gave me.

Lastly, I also want to thank Professor Dr Mary Ann Lundteigen and Ludvig Björklund for supervising me in this thesis in Norway and for all the opportunities they offered me aside from the thesis. Thank you for these opportunities and the guidance throughout the thesis.

A.A

Executive summary

A different technology needs to be used to allow the oil and gas industry to produce more oil from deeper reservoirs below the seabed. The technology used in the last decades cannot operate in this depth. A possible technology which could replace the technology used right now is an all-electric approach. This thesis looks into a potential motor which could be used for this technology. And is also looking into whether a digital twin can replace decade-long experiments with prototypes and if it is as good as or even better at demonstrating safety.

The thesis was conducted in two main steps. In the first step, literature research was conducted to become familiar with the environment, the research in this field and the topics in general. In the second step, a Matlab Simulink model of the motor, the motor's control unit, and the failures were created.

The thesis showed that a potential for the technology as well as for the digital twin exists. Due to minor setbacks while creating the models and the unfamiliarity with the technology, the thesis could only cover the first rotation of creating a model, testing it and discussing improvements about it. Further work needs to be done to thoroughly verify if this attempt can provide enough information about the motor and using a digital twin in a safety demonstration.

Table of Contents

Preface I

Acknowledgement II

Executive summary III

Glossary X

1. Introduction 1
- 1.1. Background . . . 1
- 1.2. Objective . . . 2
- 1.3. Approach . . . 2
- 1.4. Delimitation . . . 3
- 1.5. Content of this thesis . . . 3

2. Digital twin - Safety demonstration - All-electric safety valve 5
- 2.1. Digital twin . . . 5
- 2.2. Safety demonstration . . . 7
  - 2.2.1. Safety demonstration process . . . 7
- 2.3. All-electric safety valve . . . 9
  - 2.3.1. Electro-hydraulic valve . . . 9
  - 2.3.2. All-electric principle . . . 12

3. The case study 14
- 3.1. The motor . . . 14
- 3.2. Direct-quadrature-zero transformation . . . 15
- 3.3. Alpha-beta transformation . . . 17
- 3.4. Motor parameter . . . 18
- 3.5. Motor equations . . . 20
  - 3.5.1. Electrical dependencies . . . 20
  - 3.5.2. Mechanical dependencies . . . 21
  - 3.5.3. Interfacing . . . 21

3.6. Space vector pulse width modulation . . . 22
3.7. Operation and control . . . 24
3.7.1. Role as an actuator . . . 24
3.7.2. System control functions . . . 25
3.8. System structure analysis and FMECA . . . 26
3.8.1. Time frames . . . 28
3.9. Failure modes . . . 29
3.10. Safety demonstration . . . 32
3.10.1. Test strategy . . . 33

4. Modelling **36**
4.1. Implementation in Simulink . . . 36
4.1.1. Alpha-beta equations . . . 36
4.1.2. Direct-quadrature-zero equations . . . 36
4.2. Motor equations . . . 37
4.3. System architecture . . . 38
4.4. Failure mode implementation . . . 41

5. Implementation of safety demonstration **45**
5.1. Test execution and failure modes . . . 47

6. Results **49**
6.1. Results under normal working conditions . . . 49
6.2. Results of failure modes . . . 51
6.2.1. Sensor drift . . . 51
6.2.2. Missing phase . . . 54
6.2.3. Phase unbalance . . . 56
6.2.4. Wear down . . . 58
6.2.5. Broken stem . . . 59
6.2.6. Overspeed . . . 60
6.2.7. Increased friction . . . 62

7. Discussion **63**
7.1. Results of the test plan . . . 63
7.2. Objective statements . . . 65
7.3. Protective functions . . . 66
7.4. Use in overall research project . . . 66

8. Conclusion **67**

9. Recommendations for further work **68**

Bibliography **69**

A. Appendix **71**

Appendix 71

A.1. Matlab Code 71

A.2. FMECA 76

List of Figures

2.1. Trend of digital twin on google 5
2.2. Christmas tree Aker Solutions 9
2.3. Schematic electro-hydraulic Christmas tree 11
2.4. Sketch hydraulic safety valve with spring-return 11
2.5. All-electric architecture approach 12

3.1. Comparison between the different reference frames 15
3.2. DQ0 reference frame in a motor 16
3.3. Interfaces 21
3.4. 3 phase inverter circuit 22
3.5. Space vector modulation 23
3.6. System structure analysis 27
3.7. Time frames 28

4.1. Implementation of alpha-beta transformation 36
4.2. Implementation of direct-quadrature-zero transformation 37
4.3. Implementation of inverse direct-quadrature-zero transformation 37
4.4. Implementation of the electrical dependencies 38
4.5. Implementation of the mechanical dependencies 38
4.6. Implementation of rotational speed generation 38
4.7. System architecture - motor 39
4.8. Matlab implementation - motor 39
4.9. System architecture - controller 40
4.10. Matlab implementation - controller 41
4.11. Implementation failure mode sensor drift 42
4.12. Implementation failure mode missing phase 42
4.13. Implementation failure mode unbalance 43
4.14. Implementation failure mode broken stem 43
4.15. Implementation failure mode overspeed 44
4.16. Implementation failure mode increased friction 44

5.1. Model 46

6.1. Run under normal condition 50
6.2. Results sensor drift first case 51
6.3. Results sensor drift second case 52
6.4. Results sensor drift third case 52
6.5. Results sensor drift fourth case 53
6.6. Current 53
6.7. Results missing phase first case 55
6.8. Voltage, results missing phase second case 56
6.9. Results phase unbalance 57
6.10. Results wear down 59
6.11. Results broken stem 60
6.12. Results overspeed 61
6.13. Results increased friction 62

A.1. FMECA Part 1 77
A.2. FMECA Part 2 78

List of Tables

3.1. PMSM preliminary data 18
3.2. Parameter 20
3.3. Table for switching vectors 23
3.4. Switching pattern 24
3.5. Control functions 25
3.6. Failure mode matrix 30

Glossary

DCV	directional control valve.
DHSV	downhole safety valve.
DQ0	direct-quadrature-zero.
FMECA	failure modes, effects, and criticality analysis.
FOC	field-oriented control.
HP SOV	high pressure solenoid-operated valve.
LP SOV	low pressure solenoid-operated valve.
PMSM	permanent magnet synchronous machine.
PMV	production master valve.
PWM	pulse width modulation.
PWV	production wing valve.
SVPWM	space vector pulse width modulation.

1. Introduction

This chapter gives an overview of the topics and tasks this thesis will discuss. This includes the introduction of the background, the objective, the approach, the delimitations and the content of this thesis.

1.1. Background

In the last years, the oil and gas industry is experiencing a paradigm shift towards increased digitalization and electrification of the system. This includes changing the old electro-hydraulic valve actuator of the subsea oil platforms with an all-electric version.

This thesis is contributing to the PhD research project of my supervisor Ludvig Björklund, with the overall title *Digital twin for safety demonstration - Modeling a cyber-physical system.* The research project discusses the possibility of using a digital twin as a testing platform throughout the life cycle of the real system. This approach allows using the digital twin to simulate the behaviour by injecting failures or adjusting physical parameters in the digital twin. With this, it is possible to evaluate how the system (including externally delivered controllers and diagnostics) behaves when for example, a sensor is drifting or the battery capacity degraded over time and is, therefore, wrongly estimated. This research is already considering the use of an all-electric approach for the valves. The research project is part of the *SUBPRO* research centre for research-based innovations within subsea production and processing.

The thesis which will be discussed in this thesis report is going to look into the motor used in the research project. The intended motor type used is going to be a PMSM. Additionally, as the research project looks into the system's behaviour when it is experiencing abnormal working conditions, i.e. injected failures, this thesis shall also be able to induce abnormal working conditions into the system.

1.2. Objective

The overall questions which will be handled in this thesis will be:

- What specific requirements are placed on the digital twin to be useful for safety demonstration?
- How to decide on what needs to be modelled?
- What are the constraints of using such a model for safety demonstration?

To reach this, the following objectives have to be accomplished:

1. Explain, with a basis in a literature study, the meaning of 'digital twin' and 'safety demonstration' and why it is of interest to apply these two in combination for an 'all-electric safety valve' and its control system.
2. Describe the study case (the motor), including its role in the actuation of the valve and the interfaces to other parts of the valve actuation system. Identify and describe in detail the motor control functions under normal and abnormal operating conditions and point out those critical for the valve actuation's safety.
3. Identify and provide rationales for selecting parameters of interest for the digital model of the motor and interfaces.
4. Create a Simulink model of the motor and the interface to the motor control system (referred here to as the digital twin). Create also a representation of the motor control system.
5. Prepare a test strategy and step-wise plan with a basis in the requirements following a safety demonstration.
6. Carry out the test plan and discuss the results. Identify and discuss uncertainties related to the trustworthiness of the results, considering the scope of modelling, assumptions and test strategy.
7. Propose ideas for further work.

1.3. Approach

The first approach of the thesis was a literature study in the topics *digital twin*, *subsea valve*, *all electrical safety valve*, *IEC 61508*, *Industry4.0*, *safety demonstration*, *permanent*

magnet synchronous machine, *offshore oil platform* and *failure modes for motors*. Information on each topic was collected in a OneNote notebook, and based on the information collected, the first draft of the project plan was created. The project plan contained the objectives and the tasks which were part of the thesis.

With the project plan created, the thesis was separated into the modelling and theoretical part. The modelling part contained the creation of the motor model, the controller model and the failure mode models. The theoretical part included the motor dependencies, the system structure analysis, FMECA and the conditions for testing under the topic of safety demonstration.

Additionally, to connect with people in similar areas and work fields and present the thesis, I participated in multiple networks, such as the PDS, SUBPRO and CyberRAMS forum.

1.4. Delimitation

In the context of the research project, this thesis is limited to the motor with its inputs and outputs and a representation of the control system for the motor. Information about the motor is limited to the fact that the motor is a permanent magnet synchronous machine with partly known parameters, but no information about the manufacturer or specific model. Additionally, regarding abnormal working conditions, the thesis only looks at a few conditions. These conditions will mainly include scenarios where the system is experiencing failures which have their source in degradation over time. Further delimitations include the closing of the valve. As there is no additional information about the valve actuation system besides the basic architecture available, the assumption was made that the motor needs to execute 30 rotations to close the valve. Another delimitation is that the thesis creates the basic models of the motor and the control unit, but the discussion about the results will only cover the first rotation, which means the results of the first test execution and which changes need to be made for the next test execution.

1.5. Content of this thesis

- **chapter 1, Introduction**: This chapter gives an overview of the topics and tasks this thesis will discuss. This includes the introduction of the background, the objective, the approach, the delimitations and the content of this thesis.
- **chapter 2, Digital twin - Safety demonstration - All-electric safety valve**: This chapter provides background information on the terms ‘digital twin’, ‘safety

demonstration' and 'all-electric safety valve', and how they are connected in this thesis.

- **chapter 3, The case study**: This chapter introduces the motor. Furthermore, it introduces used equations, parameters, the necessary transformation matrix for the use of the motor and the defined failures, which the digital twin will be able to simulate. It also provides basic information about how the system and its components work.
- **chapter 4, Modelling**: This chapter provides information about the modelling part of the thesis. How the motor and the control system representation is modelled in Matlab Simulink. Furthermore, it displays how the failure modes are implemented.
- **chapter 5, Implementation of safety demonstration**: This chapter provides information about how the model is tested under requirements following a safety demonstration. It explains how the digital twin and the control system representation are operated to generate the results.
- **chapter 6, Results**: This chapter provides the results of the test plan execution. It displays the results of each failure mode and mentions the first impression about the results.
- **chapter 7, Discussion**: This chapter discusses the results introduced in the previous chapter and further discusses how reliable these results are. Furthermore, it continues the discussion from the failure modes to the objective statements and the contribution to the project.
- **chapter 8, Conclusion**: This chapter provides an overview of the key points discussed in this thesis.
- **chapter 9, Recommendations for further work**: This chapter, as the last, provides ideas and topics which could be relevant for further work based on this thesis.

2. Digital twin - Safety demonstration - All-electric safety valve

This chapter provides background information on the terms 'digital twin', 'safety demonstration' and 'all-electric safety valve', and how they are connected in this thesis.

2.1. Digital twin

The term digital twin was introduced by Grieves [5] in 2003. At that time, the idea and the implementation of a digital representation of actual physical products were still relatively new and immature. Most of the information for the physical products was still paper-based and even manually collected [4]. This means that the thought of a digital twin was just that. A thought. It was physically impossible to use the concept of a digital twin with the information at hand.

Almost two decades later, the idea of digital twins started to get more attention again. As shown in figure 2.1, the number of search requests for the term digital twin increased significantly over the years. But what is a digital twin?

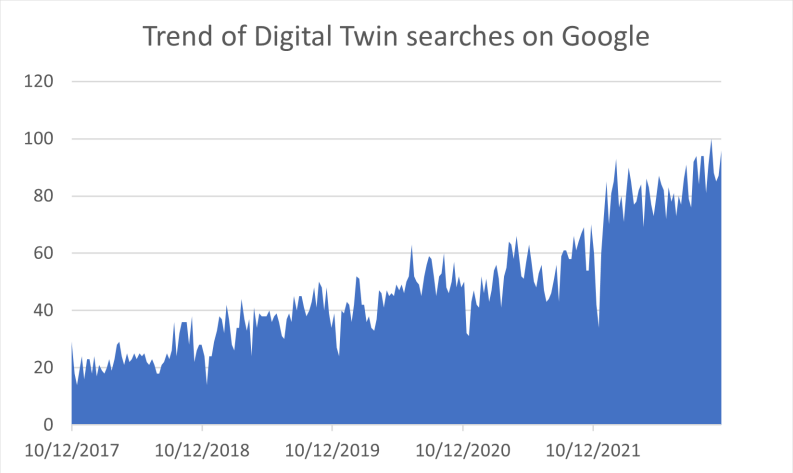


Figure 2.1.: Trend of digital twin on google (data from [2])

Grieves [4] introduced it in 2003 as an information mirroring model. This concept means the whole physical product will be mirrored in the virtual space. So, everything the real device experiences will also be experienced by the mirrored twin. Grieves and Vickers [6] reintroduced the digital twin in 2017 as a 'set of virtual information constructs that fully describes a potential or actual physical manufactured product from the micro atomic level to the macro geometrical level'. With this definition, any possible stimulation will be simulated. As this would need a lot of calculation power and time, which is not always available, there is the approach to only simulate the necessary information [4]. This 'light-weight models' [4] only simulate the necessary information, which is enough in most cases.

Grieves was not the only one defining a digital twin concept: Bolton et al. [1] described the digital twin as 'a dynamic virtual representation of a physical object or system across its lifecycle, using real-time data to enable understanding, learning and reasoning' and introduced the thought of using digital twins over the whole life cycle of the product. Purdy et al. [10] defined it similarly, without stating the word life cycle but with a similar thought in mind. He named digital twins 'a virtual replica of an object, being, or system that can be continuously updated with data from its physical counterpart'. Additionally to the term digital twin, two other words sometimes get mistaken for a digital twin, as their definition is quite similar. These terms are digital model and digital shadow.

According to WizataSA [18], the difference starts with how information is transmitted between the actual device and the digital copy. WizataSA [18] states a digital model as 'a virtual three-dimensional representation of an object that can be used for simulation and analysis'. Digital models are often used in construction and manufacturing as a three-dimensional representation in a CAD tool or as a simulation model in a manufacturing process. The difference to a digital twin is that the information for a digital model is inserted and extracted manually. There is no connection to the physical object. Digital shadows, on the other hand, are defined as 'virtual copies that we create to interact with other people and environments'. They differentiate themselves from digital models in the way how information is transmitted. Where a digital twin automatically sends and receives data from his actual counterpart, a digital shadow only receives information automatically. It is not able to share any information back with its natural counterpart.

As there is no unique definition of a digital twin, the thesis follows the definition of Purdy et al. [10] with the intent to create a virtual replica that can receive and transmit input and output signals in real-time to the physical counterpart. With this definition, the digital twin could simulate with real-world data. This definition was chosen not only to be able to simulate the behaviour of the system in the development phase but also to use the digital twin in the production phase to be able to recognise any misbehaviour of the system and use this information to run diagnostics without interrupting the production.

2.2. Safety demonstration

This section is based on the book *Demonstrating safety of software-dependent systems* from Meulen et al. [7]. The book is relevant because it describes a new framework that can be used to demonstrate novel solutions safely. Additionally, it uses examples from subsea electronic technology, which this thesis is also looking at.

As its name suggests, safety demonstration is the demonstration of safety, the demonstration that the product can meet the expectations and requirements for its safety criteria. This is easier for established industries and technologies, as, through years of conducting experiences and collecting knowledge, there are established ways to design, build and operate to meet the required safety standards. But this way of development is no longer feasible for cutting-edge, bleeding-edge, novel, and new technologies in general, as there are no years of experience and experiments. Therefore, there is a need to demonstrate the safety of these technologies without years of testing.

Another point is that for years the approach to safety was KISS, 'keep it simple, stupid', which means keep the system as simple as possible. It was feasible by its introduction in the US Navy in 1960, but now the world is no longer the same as 60 years ago. Customers want features that are often difficult, if not impossible, to implement without using software. But as soon as software is implemented, the system also becomes complex and does, therefore, no longer follow the KISS principle. The increased complexity allows the system to fail in way more ways than without software. And each form of failure needs to be accounted for and classified for its, among others, risk for safety and environmental hazard.

2.2.1. Safety demonstration process

This section is based on the chapter 2: 'Safety demonstration process' of the book *Demonstrating safety of software-dependent systems* from Meulen et al. [7].

As Andreas Falck and Andreas Hafver already pointed out in chapter 2: 'Safety demonstration process' in the book *Demonstrating safety of software-dependent systems* [7], safety demonstration is not one single activity done by one person. It is 'a range of interlinked and overlapping activities involving different disciplines and stakeholders and [is] often conducted by separate units or organisations'. According to them, it can be split into three subprocesses, each reflecting different perspectives. The guideline they propose in their chapter divides the safety demonstration process into three subprocesses, each reflecting on different perspectives. This should help determine where the safety demonstration effort should be focused.

The first of the three perspectives is '[a]ctivity perspective - assessing the overall risk of the activity', which focuses on the activity as a whole. It includes what risk can be expected from the activity as a whole to itself and the environment. This can lead to the development of safety barrier management processes, operational limitations and restrictions. Additionally, it can accept how much risk is tolerable and how much risk reduction is needed to bring the risk to an acceptable level. The definition of an adequate level is defined by the regulators' requirements and operators' internal risk acceptance criteria.

The second perspective is the '[s]trategy perspective - treating the risk'. This focuses on 'developing strategies to manage the risk associated with specific hazards, including strategies to handle uncertainties and potential surprises'. It is used to reduce the risk of a given risk picture to a tolerable level. This is done by introducing various strategies, such as design features for inherent safety or safety barriers, which might prevent the occurrence of the risk or mitigate the effects.

The third perspective is the '[t]echnology perspective - qualifying the technology'. This focuses on the technological aspect with 'technologies to be used for conducting activities, including technologies employed as part of risk treatment strategies'. This perspective examines which technology can be used for the given action and strategy. It may have been proven in the field if it is an already established technology. But if it is a novel technology, or any new technology, it must be qualified following §9 of the PSA's facilities regulation. This technology qualification is required 'to assure that the selected technologies can meet defined safety performance requirements - both technical and operational - without introducing new hazards or modifying the risk picture in unacceptable ways'. This is only necessary for novel technologies and new technologies, as established technologies should already be developed with these requirements.

To introduce novelties under the path of safety demonstration, it is firstly relevant to specify which part of the solution is novel and under which perspective this aspect can be included. Is it location-specific (first perspective), is it a different strategy to treat a hazard (second perspective), or is it a new technology (third perspective)? Depending on in which view the novelty is listed, the safety demonstration needs to be based on a different risk process.

The novel part in this thesis is in the technology perspective, therefore the third perspective. Furthermore, as the all-electrical approach of this thesis is an electronic safety-critical function, the international standard IEC61508 needs, among others, also to be considered. It covers requirements regarding process safety and the designs for electronic safety-critical functions.

2.3. All-electric safety valve

An all-electric safety valve is a term used to describe the use of safety valves in the oil and gas industry that only operate with electrical energy. This section, including its subsections, will explain the difference between this approach and the architecture used for the last 40 years.

2.3.1. Electro-hydraulic valve

The following section is based on chapter 17.1 'Architecture of electro-hydraulic Christmas trees' from Meulen et al. [7]. When companies drill for subsea oil, they need to prevent the oil they want to collect from leaking into the ocean. And if there is an emergency on the oil platform, they need to be able to stop the oil and gas from continuing to flow. For this, they use safety valves. When they drill into the reservoirs, they create a well coated in cement. On top of this well, they place a device called a subsea Christmas tree on the seafloor. Figure 2.2 shows an example of such a Christmas tree from the Company Aker Solutions.

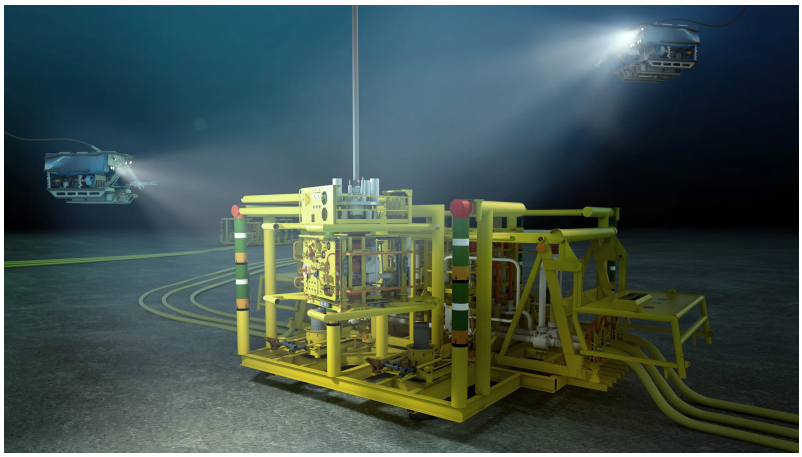


Figure 2.2.: Christmas tree Aker Solutions [14]

These Christmas trees contain valves that control the oil and gas flow. If an emergency makes it necessary that no oil or gas comes up from the reservoir, the signal will be given to close the valves.

Depending on the size of the oilfield, the Christmas tree can contain more than two valves. These additional valves can, for example, be used to inject chemicals. The two valves which are always inside a Christmas tree are the production wing valve (PWV) and the production master valve (PMV). These two valves are control valves and safety valves

at the same time. In a standard and in an emergency shutdown scenario, both valves will be actuated to close and prevent oil and gas from continuing to flow. A third safety valve, which is only operated as a safety valve, is the so-called downhole safety valve (DHSV). This valve is positioned inside the well at 500m below the seafloor. Due to being placed inside the well, and it working under a higher pressure then the other two safety valves, its internal architecture looks completely different compared to the PMV and PWV, but its purpose is the same. Close off the well and stop oil and gas from continuing to flow.

The standard architecture for these Christmas trees, which has been used for almost 40 years now, is an electro-hydraulic approach. Figure 2.3 from Meulen et al. [7], which was initially derived from figure A.13.1 of the 2018 version of the Norwegian guideline NOG 070, displays a simplified version of the design of electro-hydraulic Christmas trees. It displays the three safety valves and the choke valve, a control valve used to control the flow in the flowline. The flowline is the pipe system which connects the Christmas tree with the platform.

The already mentioned valves PMV, PWV and DHSV displayed in figure 2.3 are controlled by multiple 3/2 single solenoid valves, the *directional control valves*. Under normal operating conditions, the *EPU relay* switch is closed, which therefore actuates the *dumb directional control valve (DCV)*, the *low pressure solenoid-operated valve (LP SOV)* and the *high pressure solenoid-operated valve (HP SOV)*. With them being actuated, the hydraulic fluid used to actuate the Christmas tree's control and safety valves can reach the *directional control valves*. These valves control the opening and closing of the *PMV*, *PWV* and the *DHSV* under normal operating conditions.

In a controlled closing situation, the *directional control valves* will no longer be actuated, which leads to the hydraulic fluids being pushed back under the force of the spring into the hydraulic tank. Figure 2.4 shows a sketch of the electro-hydraulic valve in the open and closed positions. Open and close from the view of the valve itself. The left figure displays the valve in an open position, which means that no oil and gas is able to pass the valve. When the oil platform is going into production, hydraulic pressure is generated and closes the valve, which is displayed in the right figure. In the fully closed position the white areas of the stem and the valve align and oil and gas can flow. If the pressure is lost, the displayed spring is pushing the hydraulic fluid out of the valve chamber and is therefore opening the valve.

In case of an emergency closing, the *emergency shutdown node* is actuated, which opens the *LP SOV* and *HP SOV*, respectively. Additionally, it opens the switch *EPU relay*, which leads to the dis-actuation of the *dumb DCV*. In case the energy supply from the platform is cut, the *uninterruptible power supply* is lost, which in turn dis-actuates the *dumb DCV*

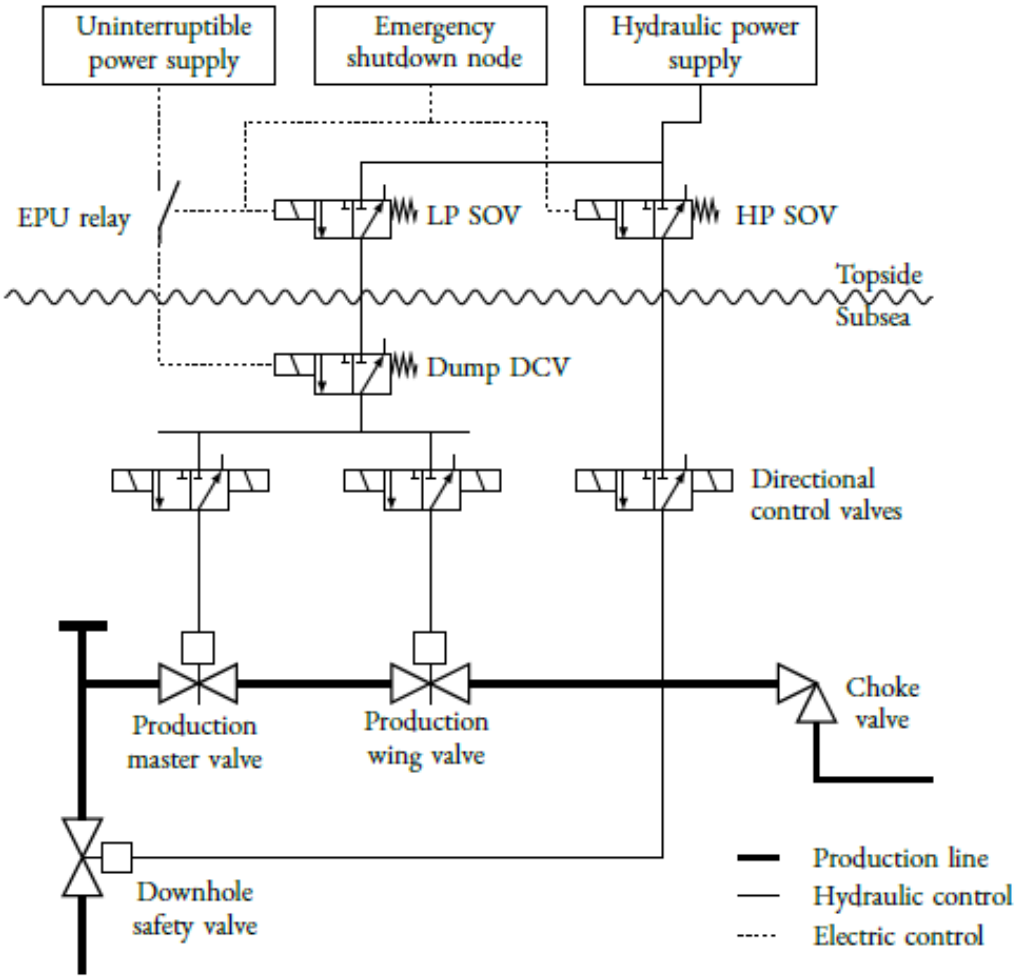


Figure 2.3.: Schematic electro-hydraulic Christmas tree[7]

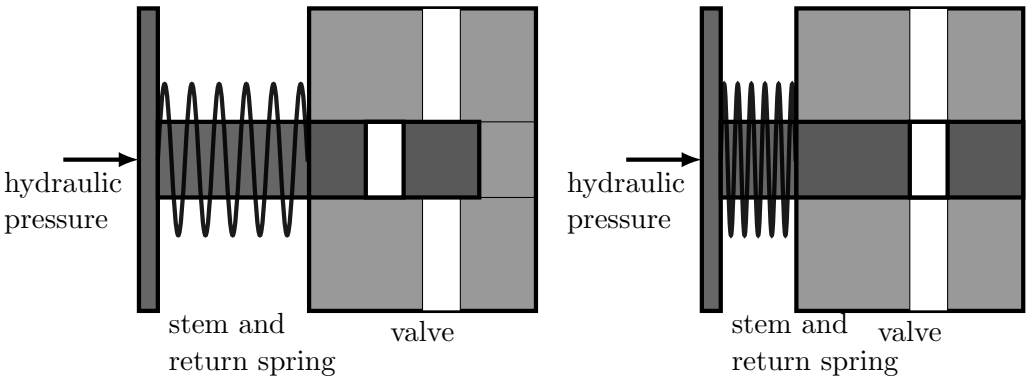


Figure 2.4.: Sketch hydraulic safety valve with spring-return

too. Even though the *directional control valves* are still actuated, as the *dumb DCV* is no longer actuated, the hydraulic fluid can flow back to the tank. With this, the *PMV*, *PWV*

and the *DHSV* are closing. This behaviour is also called *de-energize-to-safe*. The *to-safe* does not mean that the closing system itself is safe, but it means that everything behind the closing system and down the production line is secure, as no oil or gas can continue to flow.

Even though this architecture prevents leakage and hazard if there is no power supply, there are some downsides connected to this architecture. Firstly, there is no energy in place anymore. This, in turn, leads to the issue that it is not usable if reaching or maintaining a safe state requires more than just a simple power cut. Additionally, it is impossible to monitor the status of the actuator continuously. And the quality of the closing system can only be determined if the mechanism is released. As this prevents the oil and gas from flowing, it also leads to a stop in production. This test is usually only conducted once a year, so in the worst case, the spring can break one day after the functional test and the fact that the safety system is non-functional stays undetected for an entire year. Another disadvantage is that the hydraulic fluid will be suspended in the water under certain circumstances, which is an environmental hazard. The electro-hydraulic valves do not only have drawbacks, or else they wouldn't have been in use for the last forty years. First, it follows the KISS principle 'keep it simple, stupid', which means there is no complex control behaviour implemented. And secondly, in case of a power loss, it is still able to bring the actuator to a safe state and prevent risks.

2.3.2. All-electric principle

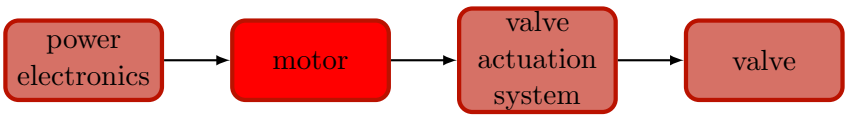


Figure 2.5.: All-electric architecture approach

Most of the disadvantages listed in the last section can be negated through an all-electric approach. All-electric in this context means the only energy used to actuate the valves is electrical energy. This approach leads to a different architectural approach (figure 2.5), which could allow the development of Christmas trees with less weight, smaller actuators and even place them in deeper depth than which is possible with an electro-hydraulic system. These can all be accomplished as there is no need for hydraulic fluid and the spring anymore. The figure also displays the way how the approach is going to work. The *power electronics* contain a battery system which is supplying the *motor* with energy and a control signal. The *motor*, which is the main part of this thesis, is delivering mechanical energy to the *valve actuation system*, which in turn is moving the *valve* to its necessary position.

There is a paradigm shift in the oil and gas industry happening which is going for increased digitalization and electrification of the systems. All-electric valves are used in the marine technology for years. They are used for example in the ballast water system of ships, the fuel system and the loading and unloading of fluids to and from ships. Even some non-critical subsea actuators are already implemented using an all-electrical approach. The reason why the safety valves of Christmas trees are not yet implemented with all-electrical valves is among others, that the regulations for the safety valves do not allow the implementation. To overcome this hurdle, new regulations need to be implemented. But each new regulations needs to prove that the way it suggests it at least on the same level or even better then the regulation it replaces. Researcher and academia, as well as operators, suppliers and the Petroleum Safety Authority such as the Safety 4.0 Project of DNV [3] with their book [7] focus on the development of a framework for standardized demonstration of safety 'to enable and accelerate the up-take of novel subsea solutions' [3]. The all-electric approach of subsea safety valves is one of these novel subsea technologies.

3. The case study

This chapter introduces the motor. Furthermore, it introduces used equations, parameters, the necessary transformation matrix for the use of the motor and the defined failures, which the digital twin will be able to simulate. It also provides basic information about how the system and its components work.

3.1. The motor

As stated in the background to this thesis, the motor type of choice for this thesis is a permanent magnet synchronous machine (PMSM).

A PMSM is a brushless motor with very high reliability and efficiency. Due to his name-giving condition, a permanent magnet rotor, a PMSM, has higher torque with a smaller frame size, which means it can generate more torque in the same space compared to other motors. Additionally, it can produce torque at zero speed but requires digitally controlled inverters.

A PMSM is, like any other electric motor, an electrical machine that can convert electrical energy into mechanical energy. To do so, the motor uses a three-phase AC input supply voltage to create a rotating magnetic field, which causes the permanent magnet rotor to rotate synchronously with the magnetic field. This rotation generates torque.

To control a PMSM, vector control techniques are used. These vector control techniques are usually also called field-oriented control (FOC). The basic idea behind vector control techniques is to split the stator current into a magnetic field part and a torque generating part. This split allows the control of both components separately and at the same time simplifies the controller [9]. This split is not particularly a split. It is more a transformation of the coordinate system and therefore allows to view and alter the relevant information more directly and easily.

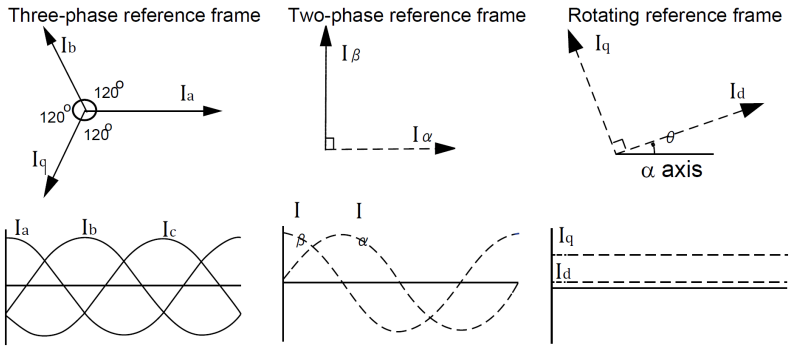


Figure 3.1.: Comparison between the different reference frames[9]

Figure 3.1 displays the different frames used to display the coordinate system transformations. The *three-phase reference frame* to the left in the figure is the regular frame with which the PMSM will be supplied. The *two-phase reference frame* in the middle will be introduced in the Section 3.3 on page 17. Based on this reference frame, it is possible to create the pulse width modulation (PWM) signal for the PMSM. Finally, the *rotating reference frame* to the right is the reference frame which simplifies the control of the PMSM. It will be introduced in the following Section 3.2.

3.2. Direct-quadrature-zero transformation

As mentioned in the previous section, the direct-quadrature-zero (DQ0)- transformation simplifies the control of the PMSM. The DQ0- transformation shifts the coordinate frame from a three-phase coordinate system into a stationary two-phase coordinate system, in which the signals are no longer sinusoidal but stationary. In figure 3.2 from *Permanent Magnet synchronous Motor Control* [9] are both reference frames (*three-phase reference frame* and *rotating reference frame*) displayed. The three red arrows labelled with *axis of phase A*, *axis of phase B*, and the unlabeled arrow in the lower left corner displays the three-phase reference frame. The two blue arrows display the rotating reference frame. If the rotor in the middle turns counterclockwise, as indicated, the three-phase reference frame would experience the three-phase signal. But the rotating reference frame would stay stationary, with the *d axis* in the direction of the north pole of the permanent magnet and the *q axis* perpendicular to it. This positioning is not picked randomly but allows to control the current in both axes. Induced current into the *d axis* would lower the magnetic field of the permanent magnet, which can be used for control systems like field weakening control, and induced current in the *q axis* can be used to create torque [9].

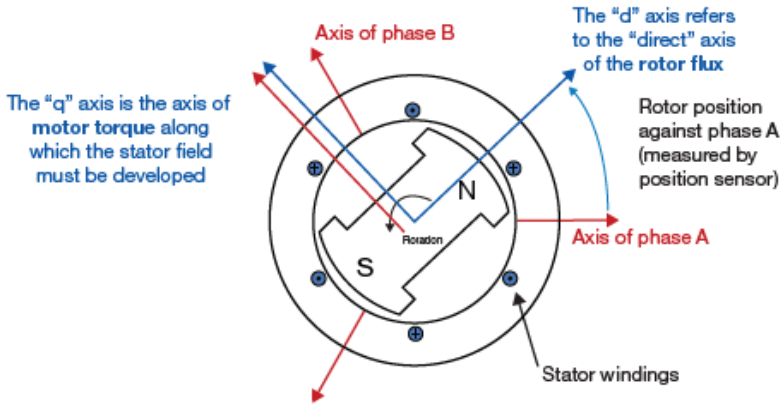


Figure 3.2.: DQ0 reference frame in a motor[9]

The following equations for the DQ0 -transformation are denoted with 'U', the international sign for voltage, but they also apply for the current 'I' and can be used interchangeably. The formulas (3.1) and its inverse (3.2) are derived from TheMathworks [16] and formulas (3.3), (3.4), (3.5), (3.7), (3.8) and (3.9) are derived from them respectively.

$$\begin{bmatrix} d \\ q \\ 0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(\theta) & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ -\sin(\theta) & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{3.1}$$

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 1 \\ \cos\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta - \frac{2\pi}{3}\right) & 1 \\ \cos\left(\theta + \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) & 1 \end{bmatrix} \begin{bmatrix} d \\ q \\ 0 \end{bmatrix} \tag{3.2}$$

$$U_d = \frac{2}{3}\left(\cos(\theta) * U_a + \cos\left(\theta - \frac{2\pi}{3}\right) * U_b + \cos\left(\theta + \frac{2\pi}{3}\right) * U_c\right) \tag{3.3}$$

$$U_q = \frac{2}{3}\left(-\sin(\theta) * U_a - \sin\left(\theta - \frac{2\pi}{3}\right) * U_b - \sin\left(\theta + \frac{2\pi}{3}\right) * U_c\right) \tag{3.4}$$

$$0 = \frac{2}{3}\left(\frac{1}{2} * U_a + \frac{1}{2} * U_b + \frac{1}{2} * U_c\right) \tag{3.5}$$

With formula (3.5), after multiplying and reducing it leads to:

$$0 = (U_a + U_b + U_c) \tag{3.6}$$

which supports the assumption that the system is balanced ($U_a + U_b + U_c = 0$).

To transform back into the original reference frame, it is possible to use the inverse formulas

derived from formula (3.2):

$$U_a = (\cos(\theta) * U_d - \sin(\theta) * U_q) \tag{3.7}$$

$$U_b = \left(\cos\left(\theta - \frac{2\pi}{3}\right) * U_d - \sin\left(\theta - \frac{2\pi}{3}\right) * U_q\right) \tag{3.8}$$

$$U_c = \left(\cos\left(\theta + \frac{2\pi}{3}\right) * U_d - \sin\left(\theta + \frac{2\pi}{3}\right) * U_q\right) \tag{3.9}$$

3.3. Alpha-beta transformation

As already mentioned in chapter 3 on page 14 in combination with figure 3.1, with the *two-phase reference frame* it is possible to create the PWM signal for the PMSM. The *two-phase reference frame* transforms the coordinate system from a three-phase sinusoidal signal into a two-phase sinusoidal signal, with the α-axis being perpendicular to the a-axis and the β-axis being the imaginary axis. The formulas (3.10) and (3.11) are derived from TheMathworks [15] and formulas (3.12), (3.13), (3.14), (3.15), (3.16) and (3.17) are derived from them respectively. The formulas are again denoted with 'U' but are interchangeable with the current 'I'.

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{3.10}$$

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \tag{3.11}$$

$$U_\alpha = \frac{2}{3}\left(1 * U_a - \frac{1}{2} * U_b - \frac{1}{2} * U_c\right) \tag{3.12}$$

$$U_\beta = \frac{2}{3}\left(\frac{\sqrt{3}}{2} * U_b - \frac{\sqrt{3}}{2} * U_c\right) \tag{3.13}$$

$$U_\gamma = \frac{1}{3}(U_a + U_b + U_c) \tag{3.14}$$

Under the assumption of a balanced system ($U_a + U_b + U_c = 0$), which leads to $U_\gamma = 0$ and can therefore be neglected in the implementation.

To transform back into the original reference frame it is possible to use the inverse formulas derived from formula (3.11) and knowing the fact that U_γ is zero in a balanced system it can be neglected in the derivation of the equations.

$$U_a = U_\alpha \tag{3.15}$$

$$U_b = -\frac{1}{2} * U_\alpha + \frac{\sqrt{3}}{2} * U_\beta \tag{3.16}$$

$$U_c = -\frac{1}{2} * U_\alpha - \frac{\sqrt{3}}{2} * U_\beta \tag{3.17}$$

3.4. Motor parameter

The following table 3.1 displays all the known parameters for the motor from the research project. This section aims to look through these parameters and discuss which one of these is relevant for modelling and which one of these is negligible.

Table 3.1.: PMSM preliminary data

Nominal torque	T_{NomOC}	Nm	47
Nominal current	I_{NomOC}	A_{rms}	20
Nominal speed	n_{NomOC}	rpm	399
Nominal power	P_{NomOC}	W	1964
Winding losses / Total losses	P_{DOC}	W	167 / 219
Holding torque	T_{HOC}	Nm	33.2
Holding current	I_{HOC}	A_{rms}	14.2
Torque constant	k_t	Nm/A_{rms}	2.291
Back EMF constant (Phase - Phase)	k_e	V_{rms}/rpm	0.14
Motor constant	km	$Nm/\sqrt{W}$	2.759
Idle speed	n_{idle}	rpm	503
max. speed (fieldweaking)	n_{max}	rpm	600
max. frequency (idle/fieldweaking)	f_{max}	Hz	126 / 150
DC bus voltage	U_{DC}	V_{DC}	100
∅ Resistance per phase (winding only)	R_{ph20}	Ω	0.112
∅ Inductance per phase (winding only)	L_{ph}	mH	1.714
electr. time constant $\tau = \mathrm{L/R}$	τ_{el}	ms	15.25
Number of pole pairs	n		15
Winding connection			Star
Rotor inertia	J	kgm^2	0.06
Inductance q axis	L_q	H	1.77e-3
Inductance d axis	L_d	H	1.31e-3

For this, there is first the question about *how detailed the model should be*? Because the

more details are needed in the model, the more parameters must be considered. Usually, the answer would be *as much as necessary, but as less as possible.* This means only implementing what is necessary to account for every normal and abnormal situation but don't include parts which are not necessary. On the one hand, this will make the model less complex, and on the other hand, it saves resources for the execution of the simulation. But in truth, this question needs to be answered by continuous testing. Continuously testing, repetition and most of all, experience from the Engineer. As there is no previous knowledge about the system and, in fact, no further information at all, this thesis is looking at the first cycle of this repetition, a basic model. The equations for this model will be introduced in Section 3.5.

The nominal parameter like *nominal torque*, *nominal current*, *nominal speed* and *nominal power* describe the motor under nominal conditions. These parameters can be used to implement safety functions for the motor but are not necessarily needed for the simulation. Therefore these four parameters are neglected for the simulation.

The same reasoning goes for the parameters *holding torque* and *holding current*, which all describe the standstill (holding) condition of the motor. Again, these are interesting for safety functions for the motor but are not needed for the simulation of the motor. Therefore they can also be neglected. The parameters *winding losses* and *total losses* are both of interest for the total power situation. But as the model is not covering this part, the parameters can also be neglected.

Compared to the other Parameters, the parameters *torque constant*, *back EMF constant*, *idle speed*, *inductance per phase*, *DC bus voltage* and *electrical time constant* describe the motor behaviour better. But even from these, the only parameter which would be useful is the *inductance per phase* parameter. But this parameter will be split into its inductance values regarding the d-axis (L_d) and q-axis (L_q). The values for these parameters are also known.

The only parameters which are interesting for modelling the motor are the *motor constant*, *resistance per phase* and *number of pole pairs.*

Table 3.2.: Parameter

i_d		d axis current
i_q		q axis current
I_{dRef}	0 $[A]$	Reference current
U_{DC}	100 $[V]$	DC bus voltage
U_d		d axis voltage
U_q		q axis voltage
L_d	1.31e-3 $[H]$	d axis inductance
L_q	1.77e-3 $[H]$	q axis inductance
ω_{el}		Electrical speed
ω_M		Mechanical speed
k_m	2.759 $[Nm/\sqrt{W}]$	Motor constant
r	0.112 $[\Omega]$	Resistance in phase
PM		Permanent magnet
T_M		Mechanical torque generated by the motor
P	30	Number of pols
J	0.06 $[kgm^2]$	Inertia of the motor
b		Coefficient of viscous friction
T_l		torque backlash from the valve actuation system

The table 3.2 summarises all the parameters which are relevant for the modelling of the digital twin. Parameters without a value will be discussed further down the chapter. They will either be calculated as part of the model or on the fly. Parameters, which are listed in this table with a value but are not listed in the preliminary table are I_{dRef} and P. P, which is the number of pols follows the equation $P = 2 * n$, with n being the number of pole pairs. The last parameter is I_{dRef}, the reference current in d-axis. As this thesis does not contain any field weakening control this parameter is always zero.

3.5. Motor equations

This section discusses the motor equations, split up into the electrical and mechanical dependencies and the interfaces it faces. The equations introduced in this section and its subsections were derived from [13].

3.5.1. Electrical dependencies

This section contains the equations for the electrical dependencies inside the PMSM. This includes the calculation of i_d, the current in the d-axis, which is responsible for lowering the magnetic field of the permanent magnet, and i_q, the current in the q-axis, which

is responsible for generating torque. The parameters U_d and U_q, which are part of the equations and are also mentioned in the parameter table, are generated through the use of a direct-quadrature-zero transformation on the supply voltage of the PMSM.

$$\frac{di_d}{dt} = \frac{U_d}{L_d} + i_q * \frac{L_q}{L_d} * \omega_{el} - i_d * \frac{R_s}{L_d} \tag{3.18}$$

$$\frac{di_q}{dt} = \frac{U_q}{L_q} - i_d * \frac{L_d}{L_q} * \omega_{el} - i_q * \frac{R_s}{L_q} - \omega_{el} * \frac{PM}{L_q} \tag{3.19}$$

3.5.2. Mechanical dependencies

This section introduces the equations needed for the mechanical dependencies inside the PMSM. This includes the calculation of the generated torque and rotational speed of the motor.

First of all: PM. The value of the permanent magnet. With k as the motor constant and P as the poles of the motor, the equation needs the number of pole pairs, therefore, it is $\frac{P}{2}$.

$$PM = \frac{2 * k}{3 * \sqrt{2} * \frac{P}{2}} \tag{3.20}$$

The torque is generated with the following equations

$$T_M = \frac{3}{2} * \frac{P}{2} * (PM * i_q + (i_q * i_d) * (L_d - L_q)) \tag{3.21}$$

followed by the equation of the rotational speed.

$$\frac{d\omega_{mech}}{dt} = \frac{1}{J} * (T_M - T_l - \omega_{mech} * b) \tag{3.22}$$

The relation between ω_{mech}, which is generated by the motor as its rotational speed and ω_{el}, which is used in the electrical equations, is displayed in the following equation.

$$\omega_{el} = \omega_{mech} * \frac{P}{2} \tag{3.23}$$

3.5.3. Interfacing

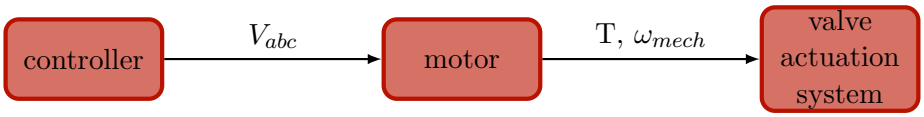


Figure 3.3.: Interfaces

The motor has an interface for each of its inputs and outputs. On its inputs, it faces a three-phase power supply (V_{abc}) created by the controller. On its outputs, the motor generates torque (T) and rotational speed (ω_{mech}), which both are related to each other. These are the most basic in- and outputs of the motor. Regarding the implementation in Matlab in chapter 4 additional inputs and outputs will be introduced.

3.6. Space vector pulse width modulation

As already mentioned, the motor faces a three-phase power supply, created by the controller, at his input. This three-phase power supply has a changing frequency depending on how fast the motor should rotate. To generate this changing frequency it is necessary to create a PWM signal. In this particular case, it is a space vector pulse width modulation (SVPWM).

The SVPWM is used to create the six signals needed for a three-legged inverter, which is shown as an example in figure 3.4.

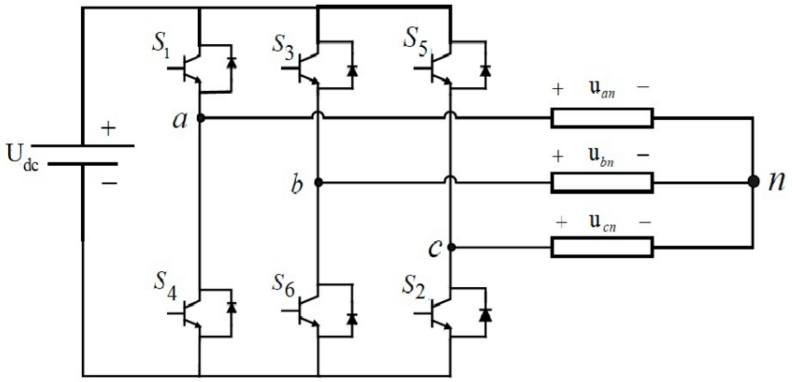


Figure 3.4.: 3 phase inverter circuit[19]

Each switch has its own signal but uses a binary logic where only one of the two switches of one leg can be active simultaneously. This means if the signal is '1', the upper switch is experiencing the '1', while the lower switch is experiencing the negated signal, which means '0'. The same applies when the signal is '0'. The upper switch is experiencing the '0', while the lower one is experiencing the negated signal, which means '1'. This system prevents the inverter from experiencing a short circuit through one of its legs.

This logic allows for eight possible constellations, which are displayed in table 3.3. The idea behind SVPWM is to use these constellations in a space vector representation to

generate the PWM signal. The eight constellations result in eight vectors, from which six can be considered active vectors (V_1 to V_6) and the last two can be considered zero vectors (V_0 and V_7). The active vectors are used to compute the PMSM, and the zero vectors put the motor in an idle state.

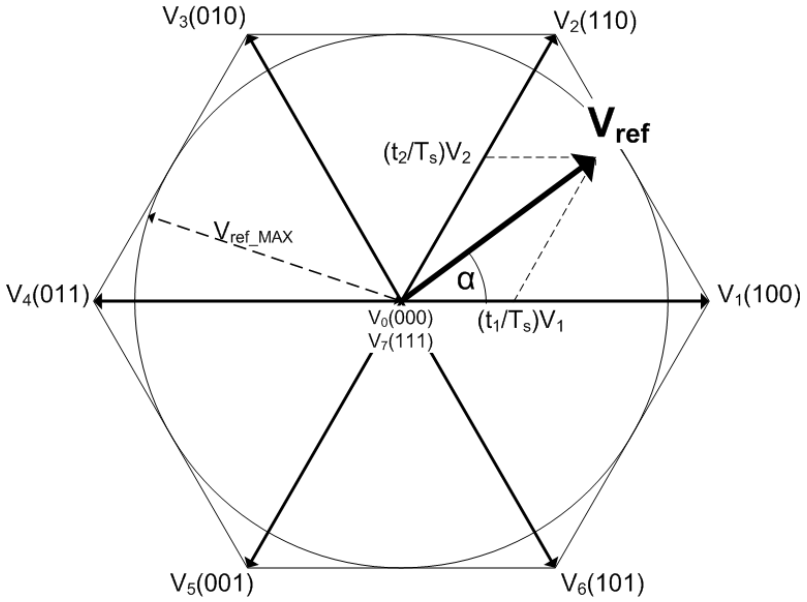


Figure 3.5.: Space vector modulation[8]

The displayed vector V_{ref} in figure 3.5 is manipulated in a space vector representation, where its length is a representation of the voltage output. Depending on where it is in the space vector room, it needs different active vectors, displayed in table 3.3.

Table 3.3.: Table for switching vectors[17]

Vector	V_a	V_b	V_c
V_0	0	0	0
V_1	1	0	0
V_2	1	1	0
V_3	0	1	0
V_4	0	1	1
V_5	0	0	1
V_6	1	0	1
V_7	1	1	1

In the displayed situation (figure 3.5), the space vector algorithm is switching between the active vectors V_1 and V_2 and the zero vectors V_0 and V_7. The time spent in each vector

depends on the needed frequency (f_s). Which, in this case, follows:

$$T_{PWM} = \frac{1}{f_s} \tag{3.24}$$

$$V_{ref} = V_1 T_1 + V_2 T_2 + V_0 T_0 \tag{3.25}$$

and

$$T_{PWM} = T_0 + T_1 + T_2 \tag{3.26}$$

This leads to the switching pattern displayed in table 3.4 with sector one being between V_1 and V_2 and continue counting in a counterclockwise direction.

Table 3.4.: Switching pattern[12]

Sector	V_a	V_b	V_c
1	$T_1 + T_2 + 0.5 * T_0$	$T_2 + 0.5 * T_0$	$0.5 * T_0$
2	$T_1 + 0.5 * T_0$	$T_1 + T_2 + 0.5 * T_0$	$0.5 * T_0$
3	$0.5 * T_0$	$T_1 + T_2 + 0.5 * T_0$	$T_2 + 0.5 * T_0$
4	$0.5 * T_0$	$T_1 + 0.5 * T_0$	$T_1 + T_2 + 0.5 * T_0$
5	$T_2 + 0.5 * T_0$	$0.5 * T_0$	$T_1 + T_2 + 0.5 * T_0$
6	$T_1 + T_2 + 0.5 * T_0$	$0.5 * T_0$	$T_1 + 0.5 * T_0$

Regarding the three-legged inverter, it was assumed that the switching frequency of the switches would slow down the simulation and, in general, lead to issues with the simulation. Additionally, the safety demonstration discussed in this thesis shall verify the logic controlling the safety valve and will therefore not directly interface with the switching circuits. Therefore, there is no need to implement the creation of the switching pattern, and the signal can thus be directly implemented with a Matlab block.

3.7. Operation and control

To test the system with failure modes, it first needs to be described how the system is working in its role as the actuator, which means under normal working conditions. Afterwards, the control functions implemented into the digital twin will be defined, and lastly, the failure modes implemented into the system will be described.

3.7.1. Role as an actuator

In its role as an actuator, the motor is supplied by a three-phase SVPWM. This three-phase power supply is then transformed with the help of a DQ0 -transformation into V_d

and V_q. These two voltage parameters are then, in turn, converted into i_d and i_q, with the help of the equations introduced in Section 3.5.1. Lastly, under the equations introduced in Section 3.5.2 and under consideration of the last input, T_l turned into torque and rotational speed, which are both the outputs of the motor.

3.7.2. System control functions

The following table 3.5 displays the control functions which can be considered for the system. As the only information regarding the motor was its parameters and nothing else, it had to be assumed that the motor does not have any control or protective functions. Therefore typical protective functions of a motor are listed along. The control functions were derived from the functions a typical motor should be able to execute and extended by the functions the controller is adding to the system. Furthermore, functions which are likely already installed when a new motor is bought were also considered in this table.

Table 3.5.: Control functions

Function	Notes	N / P / D	Implemented
Motor start	to start the motor	N	Yes
Motor stop	to stop the motor	N	Yes
Generating torque	to generate torque	N	Yes
Generating rotation	to generate rotations	N	Yes
Position control	to control the position	N	Yes
Rotational speed control	to control the rotational speed	N	Yes
i_d control	to control the current in d-axis	N	Yes
i_q control	to control the current in q-axis	N	Yes
Over speed	to prevent over speed	P	No
Over / under voltage	to prevent over or under voltage	P	No
Temperature	to prevent overheating	P	No
Vibration	to recognise vibrations	D	No
Overload	to prevent and recognise overload	P / D	No
Short-circuit	to prevent short-circuit	P	No
Circuit breaker	to protect the motor	P	No

To explain the table 3.5, the first column states the function's name. The second column describes in a brief statement what the main objective of the function is, followed by the third column, which states if the function is a normal (N), a protective (P) or a diagnostic (D) function. The last column displays whether or not the function is implemented into the digital twin as part of the thesis.

Therefore, the table states that only the normal functions needed for the motor operation

are integrated into the digital twin. No protective or diagnostic functions are implemented because it is unknown which exact motor model is used in this thesis. As assuming the existence of a safety function which is, in the end, not implemented in the motor is a safety risk in itself, the decision was made to assume that no safety, protective or diagnostic function is implemented in the motor or the control system. Therefore, the failure modes later discussed in this thesis will show which safety, protective or diagnostic functions are necessary, which can be neglected and which were forgotten in the table.

3.8. System structure analysis and FMECA

To identify failure modes and, therefore, the abnormal operation conditions, the first thing to consider is a system structure analysis and, based on the system structure analysis, a failure modes, effects, and criticality analysis (FMECA). A system structure analysis (displayed in figure 3.6) divides the system into multiple manageable units, which normally are its functional elements. This allows to look at each functional element separately and helps to figure out which part of the particular element can lead to which kind of failure. A FMECA, on the other hand, is a technique used to display '[a]ll potential failure modes of the various parts of a system', '[t]he effects these failures may have on the system' and '[h]ow to avoid the failures, and/or mitigate the effects of the failures on the system' (all [11]). The FMECA is displayed in figure A.1 in the appendix A.2 on page 76.

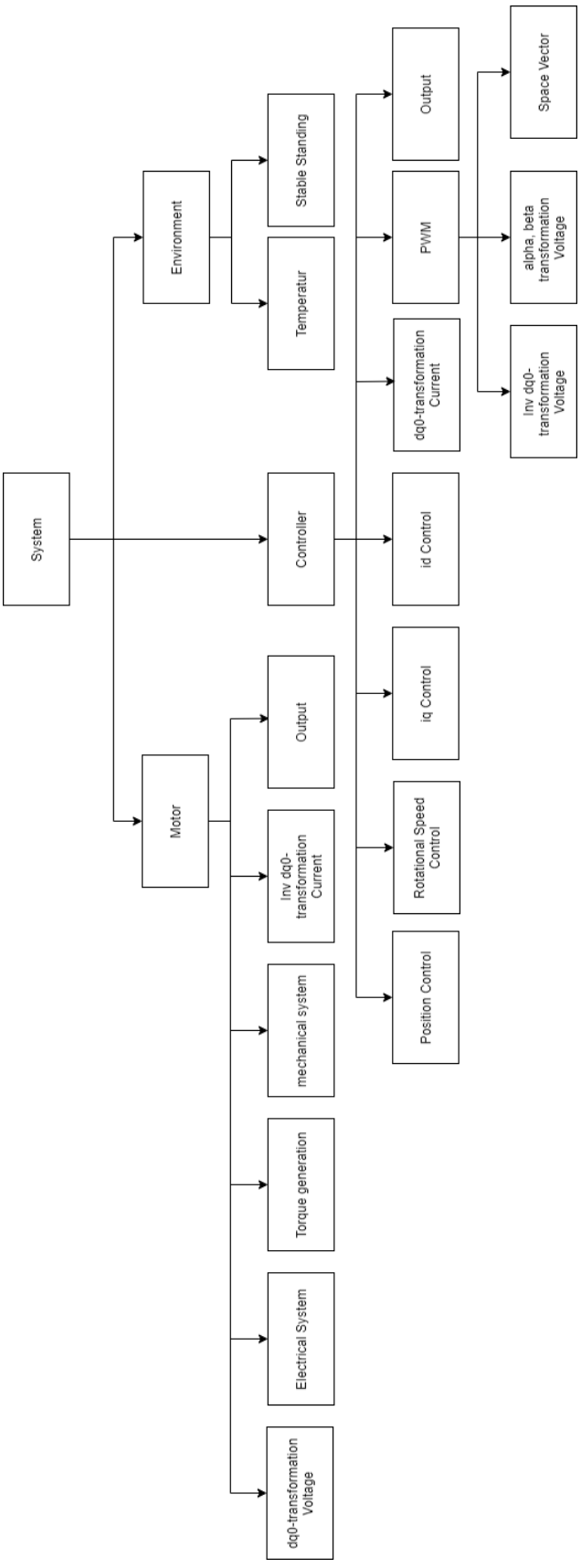


Figure 3.6.: System structure analysis

The FMECA is created based on the system structure analysis. Besides the failures displayed in the FMECA, many more failures can be considered, but as mentioned in delimitation, the thesis is only looking at failures which have their source in degradation over time. Additionally, as most failures in the FMECA increase the friction in the system, only one failure mode per behaviour is chosen to be implemented. This reduces the number of tests and prevents testing the same behaviour multiple times.

3.8.1. Time frames

Another way to further lower the amount of test cycles is to split the whole movement of the valve into three different sections. These sections, in the following called time frames, allow lowering the amount of test cycles due to the fact those specific failure modes are only implemented in the time frame in which they can be recognized or have an impact on the system.

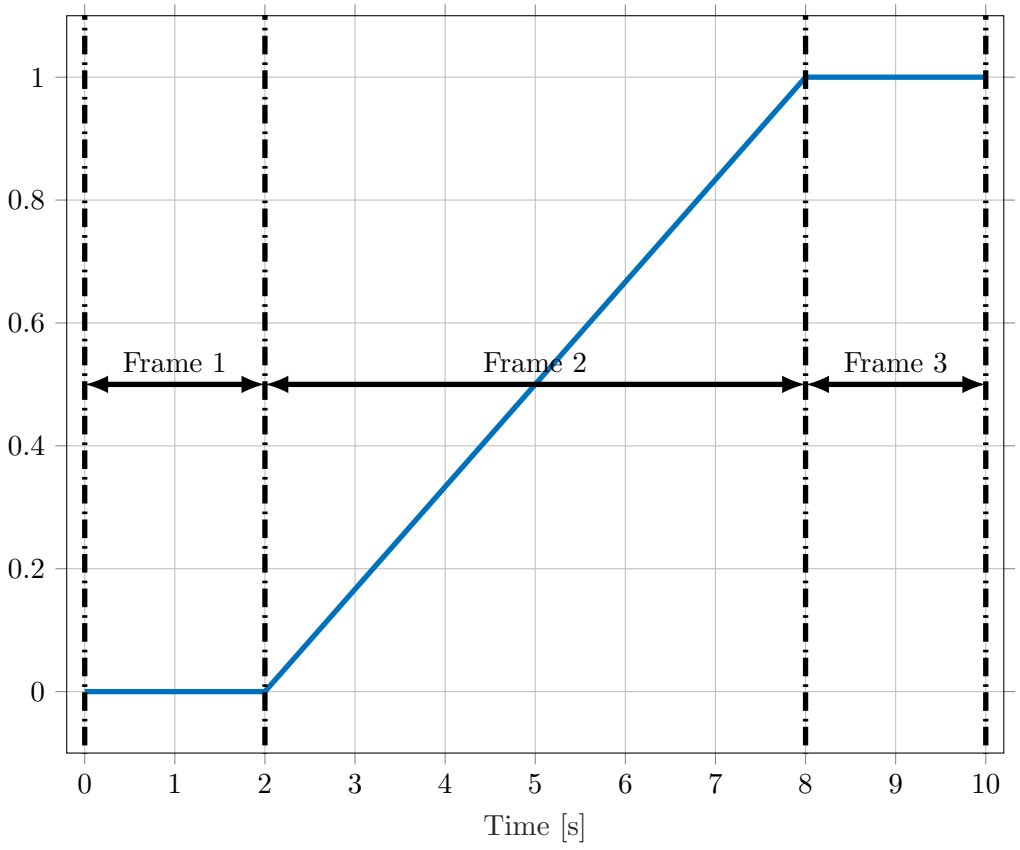


Figure 3.7.: Time frames

To simplify the explanation, figure 3.7 shows the three time frames. The scaling of the x-axis, y-axis and the gradient is chosen randomly.

The first time frame includes the time from starting the simulation till the actual moving of the valve. This time frame intends to show what will happen if the failure condition already exists by the time the valve starts to close.

The second time frame starts when the first time frame stops, at the moment when the actual movement of the motor and, therefore, the valve begins. The time frame ends when the valve reaches, under normal conditions, the end position and is, therefore, fully closed. This time frame intends to show what will happen if the failure condition occurs while the motor is turning and trying to close the valve. There is no information on how many turns the motor needs to make until the valve is at its final position, so the assumption of 30 rotations is made.

The third and last time frame starts when the valve is fully closed and the motor is no longer operating. It intends to show failure conditions that impact the system after the valve is fully closed.

3.9. Failure modes

After stating the term failure mode now multiple times throughout this thesis, this section will introduce what the term means in the context of this thesis, how they were derived, which ones will be implemented and why they will be implemented.

A failure generally is a situation where things don't go according to plan. In this case, it means that the system is not operating as it should. A failure mode, in this case, means that a specific situation or failure occurred which doesn't allow the system to operate normally anymore. This means the system is working in an abnormal condition and is, therefore, unable to function fully. As something like this should not happen, failure modes were developed that cover these failures, so if they occur, there are regulations in place that can mitigate the failure and allow the system to stay fully functional. Or if mitigation is not possible, to bring the system into a safe state and thus prevent it from damaging itself and its environment.

The failure modes in this thesis were derived from the FMECA introduced in Section 3.8. They mostly describe situations where the system is getting older and is no longer able to work under normal conditions. This is also the reason why failure modes are implemented in this thesis. The digital twin this thesis is producing should be able to simulate how the system behaves under abnormal working conditions. If these simulations show that the system's behaviour endangers the system itself, human lives or environmental structures it should be able to help to develop strategies to prevent this abnormal operation condition

from occurring or, if that is not possible, to reduce the risk to an acceptable level. As the system is developed as an exact copy of the actual device in mind, these tests allow to look into the model and see where things went wrong. This allows the adaptation of the system through the repetition of tests and reviews until the system can exactly copy the actual device or until the behaviour is close enough to be sufficient.

Most of the following failure modes are listed in the FMECA with are severity ranking of ten or almost ten. This means that if these failure modes would happen to the real system, the system would be unable to operate. Therefore the behaviour of these failure modes is necessary to know.

- wrong sensor signal - resolver
- missing phase in the motor power supply
- phase unbalance in the motor power supply
- higher resistance torque due to wear down of stem or bearing
- no resistance torque - broken Stem
- overspeed

For easier handling of the failure modes, some of them will be named differently throughout this thesis. So will the *wrong sensor signal - resolver* be renamed into sensor drift, *higher resistance torque due to wear down of stem and bearing* will be shortened to wear down, and *no resistance torque - broken stem* will also be shortened to broken stem.

Table 3.6.: Failure mode matrix

Failure mode	Time frame 1	Time frame 2	Time frame 3
Sensor drift	No	**Yes**	No
Missing phase	**Yes**	**Yes**	No
Phase unbalance	No	**Yes**	No
Wear down	**Yes**	No	No
Broken stem	**Yes**	**Yes**	No
Overspeed	No	**Yes**	No

The table 3.6 states which failure mode has an impact in which time frame or which failure mode should be run as a test in this specific time frame.

The first failure mode which will be discussed is **sensor drift**. Sensor drift means that the rotational signal of the motor is not transmitted correctly. This would mean that the motor controller assumes it is getting the rotational movement signal of the motor, but

in truth, it is getting a deviated signal. As this is the only information the controller is getting regarding the position, it needs to assume all the time that this signal is valid. This situation can become tricky if implemented in the three time frames. In the first and third frames, the controller would assume that no rotation is happening due to being idle. It can be assumed that the resolver used for measuring the rotation cannot provide a signal if there is no movement from the motor. Therefore there is no need to test this failure mode in these two time frames. In the second time frame, however, as there is movement from the motor, the resolver may be transmitting a faulty rotation signal. To generate this case, the feedback loop of the resolver will be modified.

The second failure mode is the **missing phase**. Under normal circumstances, this will be a situation when one or more of the three phases from the controller, which supply the motor with its supply voltage, gets loose, breaks or short circuit. This leads to the situation that the motor is no longer provided with a three-phase power supply. In the first frame, this should not be necessary to test, as there is no movement in this time frame, but as the motor is a PMSM, a synchronous machine should be unable to start with only two phases without being supported by a starter. It might be interesting to test this start condition by simulating this behaviour. When the motor loses one phase while the system is already running, the system should be able to continue running. But after losing two phases, the motor should be unable to continue running. Therefore it is worthwhile to test this failure mode in the second time frame. But in the third frame, as there is no movement, testing for a missing phase is unnecessary. Therefore this test can be neglected here.

The third failure mode is a **phase unbalance** in the three-phase supply voltage. This means that the line-to-line voltage between the three phases differentiates by more than 1%. This situation can seriously damage electronic equipment and motors as unbalanced voltage leads to unbalanced currents in the motor winding, which leads to rising winding temperature and, therefore, reduced life expectancy. As there is no neutral wire, the balance will need to be kept, which means that the imbalance occurs in at least two of the three phases. As there will be no movement and, therefore, no voltage in the first or third time frame, the test will only be executed in the second time frame.

The fourth failure mode is **wear down**. This is a naturally occurring phenomenon. When equipment gets old, it becomes worn out, increasing the system's friction. As this is not a behaviour which can happen from one second to the next but only occurs over time, it will not be tested in the second time frame. Additionally, there is no need to test this failure mode in the third time frame, as there is no movement. But as it is a behaviour which builds up over time, it is worth testing when it is already implemented at the beginning of the movement, which means in the first time frame.

The fifth failure mode is **broken stem**. As mentioned in the fourth failure mode, when equipment gets old, it wears down, and with time it even breaks. As this situation can happen from one second to the next, even more so when there is a change of load, it will be considered worth being tested in the second time frame. Additionally, it is worth testing it, assuming that the stem broke before the movement is initiated (first time frame). But in the last (third) time frame, there is no need to test for it, as there is no movement.

The sixth failure mode is **overspeed**. Overspeed is a situation where the motor turns faster than it is technically designed for. This can lead to the destruction of the motor. This failure mode will look into what will happen if the closing signal is given in a time frame that is too small to close the valve fully and stay under the technical dependencies. Under this condition, it is only worthwhile to test this failure mode in the second time frame.

Additionally, to the failure modes derived from the FMECA, one other specific failure mode will be implemented. It is not part of the FMECA, but it is a situation worthwhile to be tested. This is a scenario when the movement, in the beginning, experiences higher friction than later on in the movement. This behaviour is comparable to the stick-slip effect.

3.10. Safety demonstration

After introducing the motor and its components, along with the failure modes implemented, this section will introduce why safety demonstration is necessary as part of this thesis.

The all-electric safety valve approach is still in its theoretical steps. To be clear, it is a novel technology in this area of use. Therefore it needs to prove that the technology is as safe as the one implemented right now or even safer. A safety demonstration must be conducted to demonstrate the technology's safety. But without years of testing, there is the need for a different approach to prove the technology's safety. This is where a digital twin comes into play. The digital twin can demonstrate the behaviour of the technology on a simulated basis without having an actual device or prototype. Simulating without a real device or prototype significantly lowers the cost and time needed to provide the necessary information. Additionally, these digital twins can be used to provide evidence that the system they represent provides operational safety under any circumstances.

Any device or vehicle on the open market needs to follow specific regulations. When it comes to the oil and gas industry, these regulations are even more important, as leaks

and failures can endanger not only the personnel working on the oil platforms but also marine life forms and environmental structures. To prevent these endangerments, the oil and gas industry has multiple regulations for risk assessment and treatment. The regulations include *ISO 31000*, *NORSOK Z-013*, *ISO 17776*, *NORSOK S-001*, *IEC 61508*, *IEC 61511* and *NOG 070* [7]. The traditional way to prove that the system follows all of these regulations was to test it in a real-world environment. With this, it was possible to determine the entire system's performance under real-world conditions. However, not every test scenario can efficiently be executed without endangering the whole system under test, the environment or even the humans involved in the testing. Therefore real-world testing needs to follow specific safety criteria too. Testing under enclosed conditions can reduce the risk for human and environmental structures, but at the same time, they reduce the number of possible tests overall. So, in either case, if there is a need to perform exhaustive testing, it may result in high costs and require a significant amount of time and infrastructure. These are points where it becomes interesting to test with digital twins. With digital twins, there is no need for multiple prototypes anymore. No personal or environmental structure is at risk when something goes wrong. And any test scenario you can simulate can also be tested.

3.10.1. Test strategy

This thesis introduced a motor simulation under the motor's known differential equations and mathematical behaviour to prove the possibility of using a digital twin for a safety demonstration. Furthermore, a simulated representation of a control system was also introduced, as the motor cannot perform without a controller controlling it. As described in the previous section, this digital twin will be operated under normal operating conditions and the influence of multiple failure modes. The information provided by these test executions allows for further improvements of the digital twin and the tests themselves. Also, it allows to specify if there are issues with the new technology.

To make these improvements happen, the test executions will be part of a test strategy. To develop the test strategy, the following questions need to be discussed:

- what are the objectives of the tests
- which information is needed
- which coverage is needed
- how should the tests be executed
- how will the results of the tests be used

- are there any risks related to the execution, and how will they be mitigated

The **objectives** of these tests, as already introduced at the beginning of this subsection, are to prove that the digital twin can fully simulate the real-life motor and its behaviour under normal and abnormal conditions.

The motor model and the controller are both parts of the **information** needed to execute the tests. Additional information is the motor parameter, the controller parameter, the closing signal, the failure modes and the hardware used for measuring signals.

The motor and controller models will be fully introduced and implemented in chapter 4. As no information was available on which specific motor was used and how this motor was controlled, assumptions had to be made. One assumption was that the controller uses a cascade control system, maintaining the position, rotational speed, and electrical currents under a gain margin of around 20 dB and a phase margin of around 60 degrees. These parameters can be considered less aggressive, but it was also assumed that closing the valve generally has a higher priority than closing the valve fast. Therefore the controllers were implemented under less aggressive behaviour to not additionally damage the system.

Even though it was unknown which motor was used, its parameters were known. But for any electrical device, its parameters are always listed under the most optimal conditions. Additionally, deviation from these parameters is very common due to manufacturing tolerances. To **cover** for these deviations, the test strategy should account for test executions in which the parameters used to have a deviation of up to five to ten per cent of the given parameters. To not heavily increase the load on this thesis, the test strategy in this thesis is considering the parameters to be exact.

As the motor and, therefore, the model is part of a more extensive system, the motor should officially close a safety valve. But modelling the more comprehensive valve actuation system and the valve, in general, is not part of the scope of this thesis. To still generate valid data, an assumption for the closing signal was made, which is 30 rotations. The electrical current and rotational speed measurement units were also neglected due to unknown internal parameters. The only information known is that a resolver measures the rotational speed, but other characteristics were unknown.

As one of the topics for this thesis contains the implementation of failure modes, these failure modes must also be covered in the test strategy. Therefore, each failure mode individually and the general information about the **test execution** is covered in the following Section 5.1.

After each test execution, the results will be saved in a data inspector file with a time stamp

and which failure mode was tested. These results will then be introduced in chapter 6 and discussed in chapter 7. Depending on the results, changes will either be suggested for the test case or for the model, which will be further suggested in chapter 9 for further projects based on this topic.

As every test execution is executed in a virtual environment and as a simulation, there is no need to look into risks related to the execution in itself. The only relevant risk is the risk the system will introduce into the real world. But for using the digital twin the way it is intended, further projects will need to account for execution risks.

4. Modelling

This chapter provides information about the modelling part of the thesis. How the motor and the control system representation is modelled in Matlab Simulink. Furthermore, it displays how the failure modes are implemented.

4.1. Implementation in Simulink

This section is going to describe how the transformation equations introduced in Section 3.2 and Section 3.3 on the pages 15 and 17 will be implemented in Matlab Simulink.

4.1.1. Alpha-beta equations

The alpha-beta transformation, introduced in section 3.3 on page 17 transforms the three-phase reference frame into a two-phase reference frame, where the α-axis is perpendicular to the a-axis and the β-axis being the imaginary axis. After introducing the formulas (3.12) and (3.13), the following figure 4.1 displayed the implementation in Matlab Simulink.

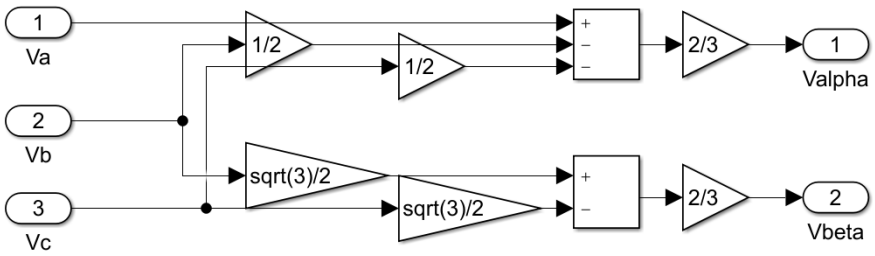


Figure 4.1.: Implementation of alpha-beta transformation

4.1.2. Direct-quadrature-zero equations

The direct-quadrature-zero transformation, introduced in section 3.2 on page 15 transforms the three-phase coordinate system into a stationary two-phase coordinate system,

in which the signals are no longer sinusoidal but stationary. The implementation for the formulas (3.3) and (3.4) and the formulas (3.7), (3.8) and (3.9) for the inverse transformation are displayed in the following two figures 4.2 and 4.3.

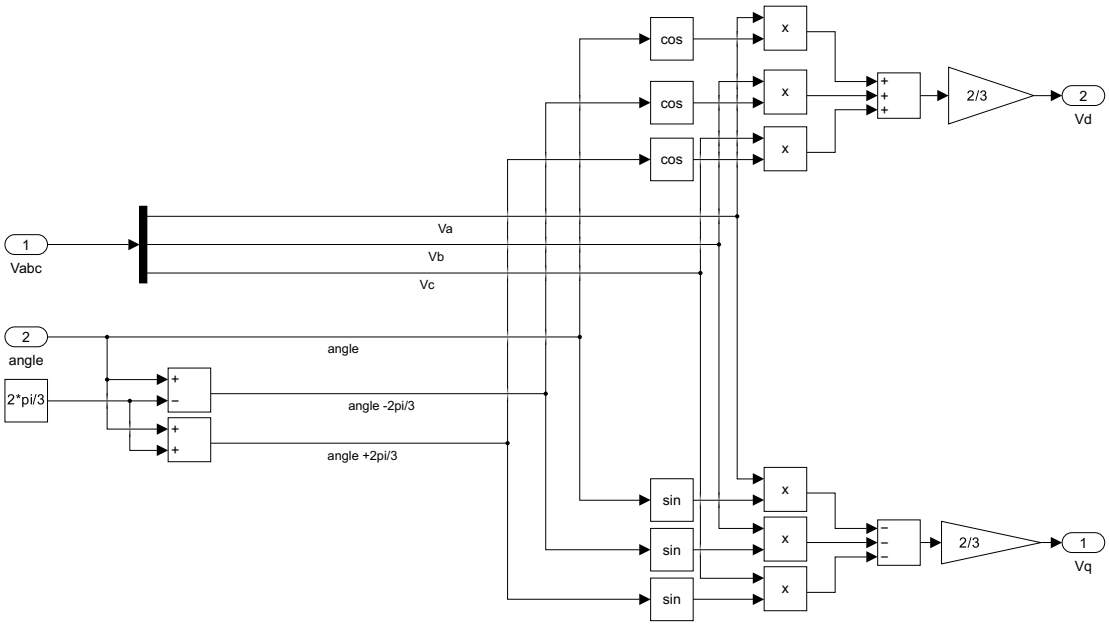


Figure 4.2.: Implementation of direct-quadrature-zero transformation

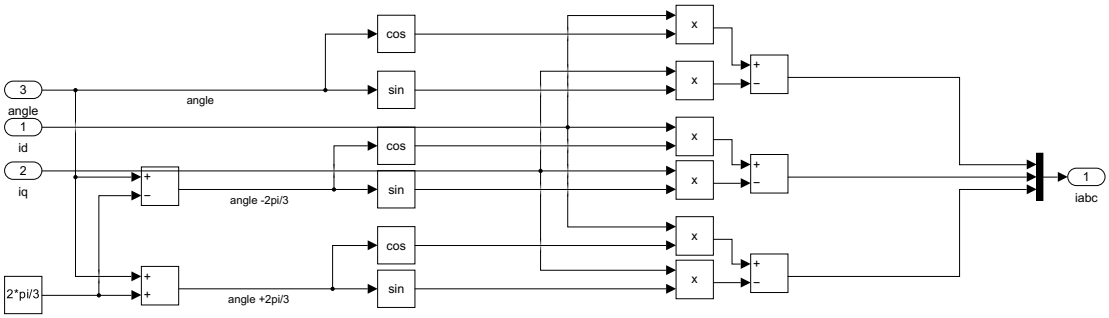


Figure 4.3.: Implementation of inverse direct-quadrature-zero transformation

4.2. Motor equations

This section will describe the implementation of the electrical and mechanical dependencies of the PMSM. Introduced in Section 3.5.1 and Section 3.5.2, the following three figures 4.4, 4.5 and 4.6 display the implementation of the electrical and mechanical dependencies

inside the model.

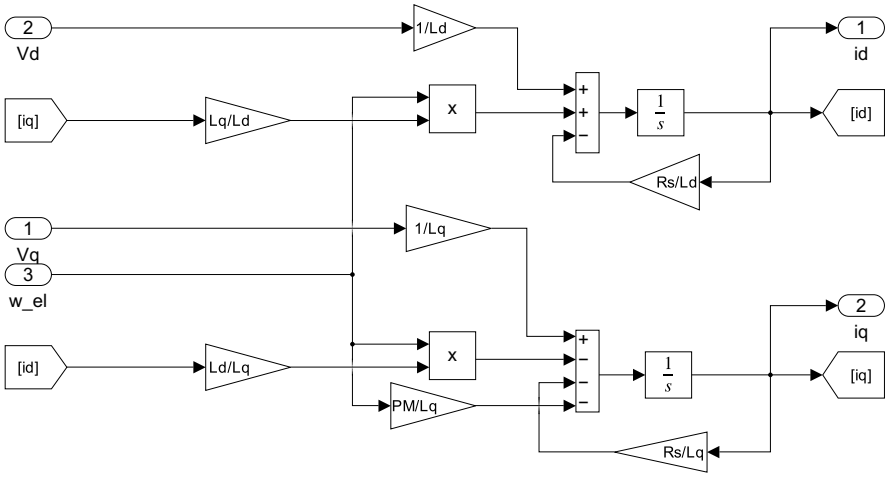


Figure 4.4.: Implementation of the electrical dependencies

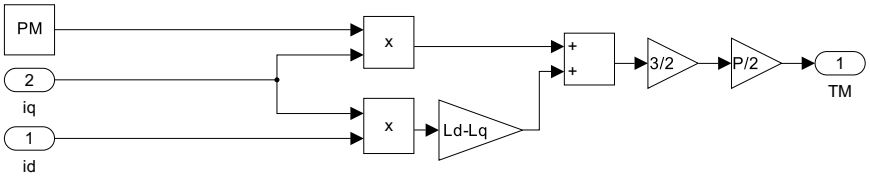


Figure 4.5.: Implementation of the mechanical dependencies

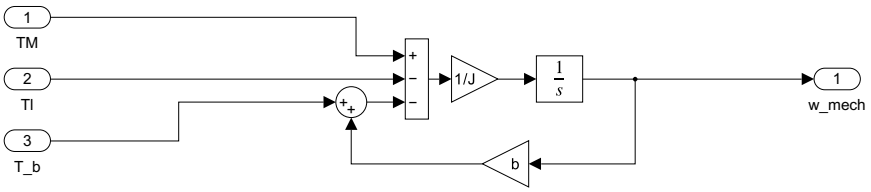


Figure 4.6.: Implementation of rotational speed generation

4.3. System architecture

After the previous sections in this chapter introduced the components' implementation, this section will introduce how they are connected. This starts with figure 4.7, which

displays the architecture of the motor. Fed with the three-phase power supply provided

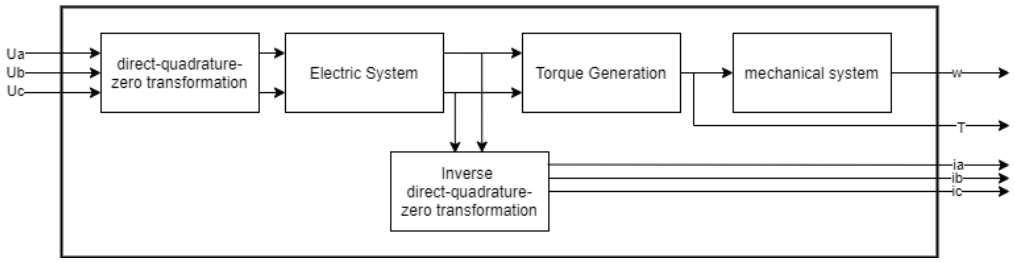


Figure 4.7.: System architecture - motor

by the controller, it generates the outputs rotational speed (ω_{mech}, marked in the figure with w), the motor current (i_a, i_b and i_c) and the torque (T). The implementation in Matlab is displayed in figure 4.8. Differently than displayed in the architecture, there are two more inputs in the Matlab implementation. One is the resistance torque, fed back from the valve actuation system the motor is trying to turn. The other one is part of a failure mode. This will be further discussed later in this chapter. Additionally, the input voltage, as well as the output current, are combined into one signal in the implementation and not split up into three as displayed in the architecture.

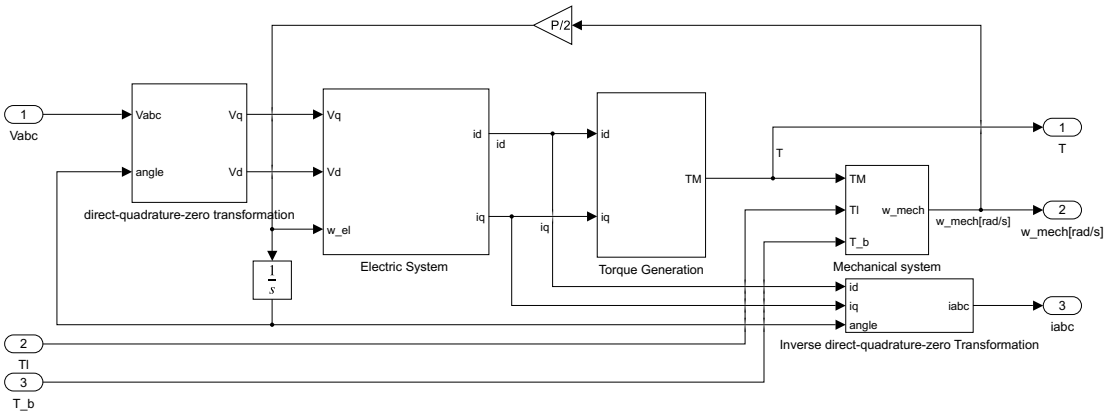


Figure 4.8.: Matlab implementation - motor

The controller's architecture, which controls the motor, is displayed in figure 4.9.

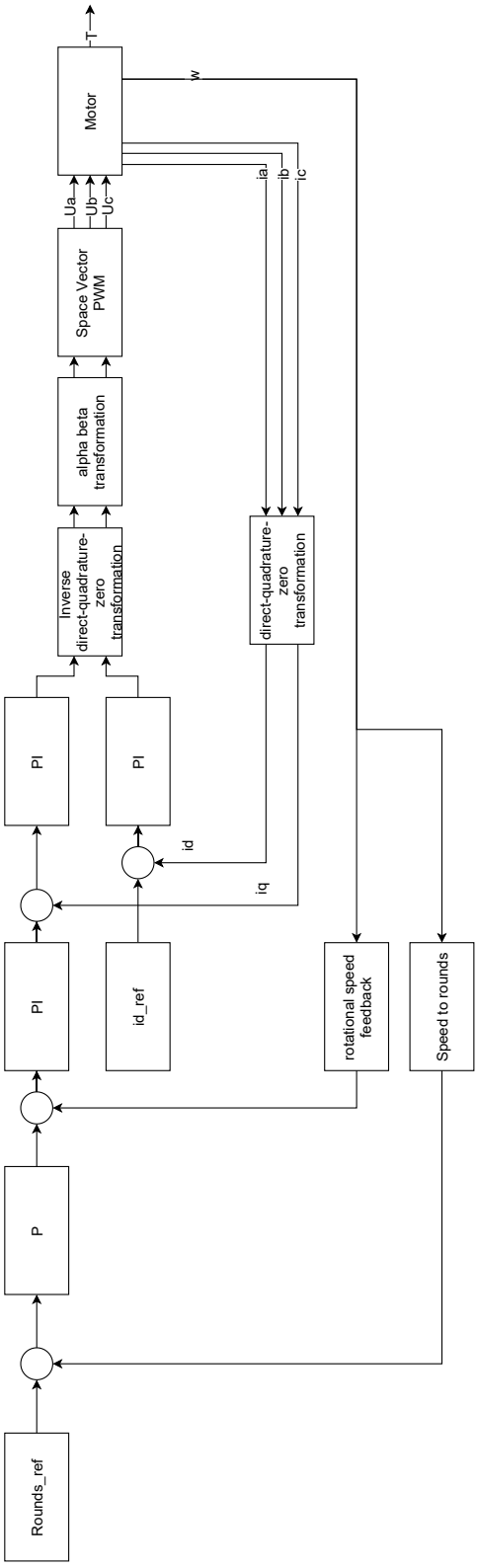


Figure 4.9.: System architecture - controller

The increase from zero to the necessary 30 rotations is implemented as a linear movement with a stable gradient to prevent the motor and the controller from overworking. It might take more time to close the valve, but it prevents the motor and controller from damaging themselves. The slower closing can lead to increased risk for the environment, but an even more significant risk would be that the motor and the system behind it can be permanently damaged if the valve closes too fast. The following figure 4.10 displays the implementation of the controller in Matlab.

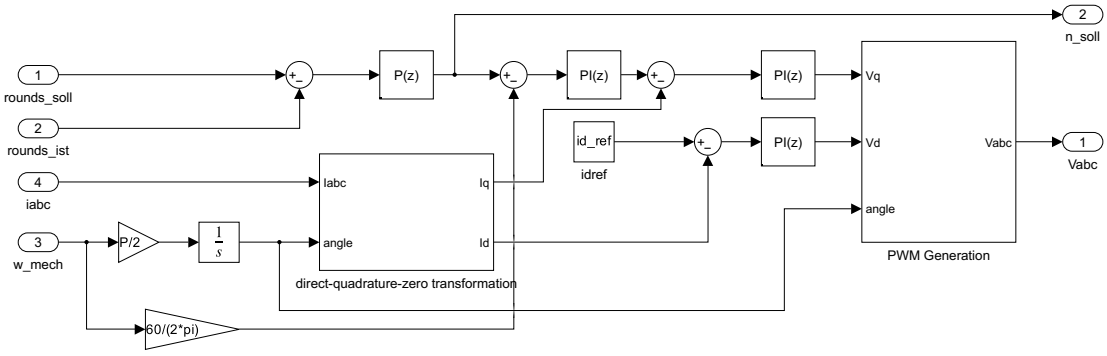


Figure 4.10.: Matlab implementation - controller

To make it easier for further projects based on this thesis, the controller and the motor were both separated into extra files and then included in the failure mode file with a model reference block. This allows exchanging the controller and the motor on a flying basis. This also means that the necessary information needed for the controller to operate is fed into it via input points. This includes the amount of rotations it should have done and is doing, the motor current and the rotational speed, both of which are needed for the controllers. Additionally, it outputs the three-phase power supply for the motor and, for diagnostic reasons, the setpoint position of the rotational speed too.

4.4. Failure mode implementation

After the previous section introduced the implementation of the motor and the controller in Matlab, this section will introduce how the failure modes were implemented in Matlab and will give some basic information about them. Further information about how the failure mode will be triggered and how it works will be discussed in chapter 5.

The implementation of the failure mode **sensor drift**, as displayed in figure 4.11, is accomplished by adding and subtracting a constant and a changing value on top of the regular rotation signal. This will lead to a deviation of up to half a rotation. The signal

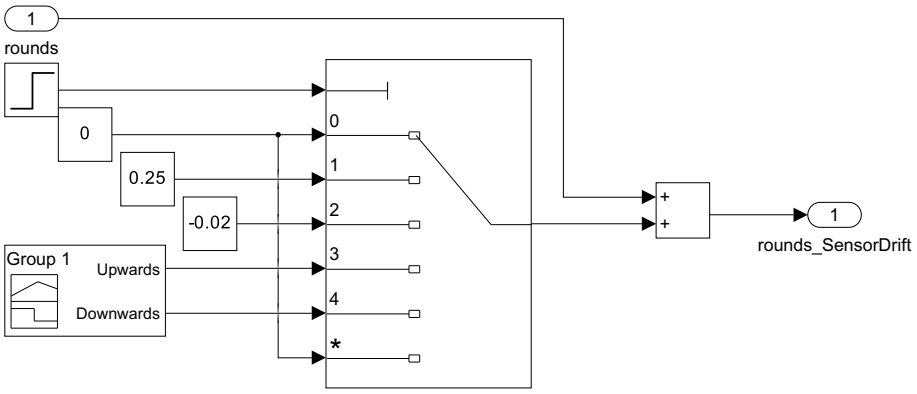


Figure 4.11.: Implementation failure mode sensor drift

rounds is directly taken after the resolver and describes the amount of cycles the motor is executing. The signal *rounds_SensorDrift* is fed to the controller as the actual rotational position.

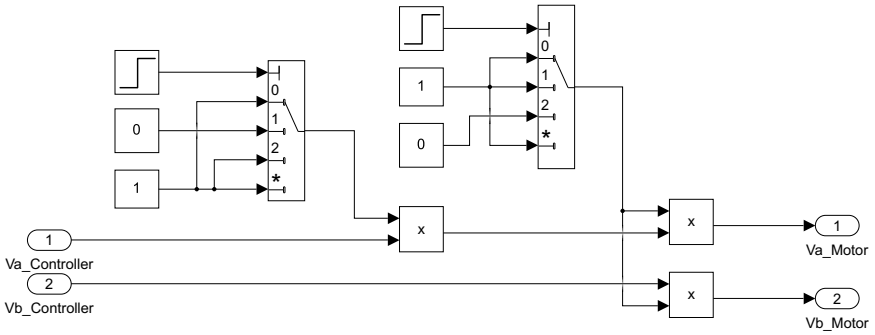


Figure 4.12.: Implementation failure mode missing phase

Failure Mode **missing phase**, with its implementation displayed in figure 4.12, is implemented in the connection between the controller and motor. The idea behind this implementation is that multiplication with zero is always zero. Therefore it is an easy way to simulate a missing or broken phase.

Failure Mode **unbalance**, with its implementation displayed in figure 4.13, is also implemented in the connection between the controller and the motor, as this is the easiest way to add an offset to the signals going into the motor.

The implementation for failure mode **wear down** is displayed in figure 4.16 of the failure mode increased friction as both failure modes can be included in the same area. The

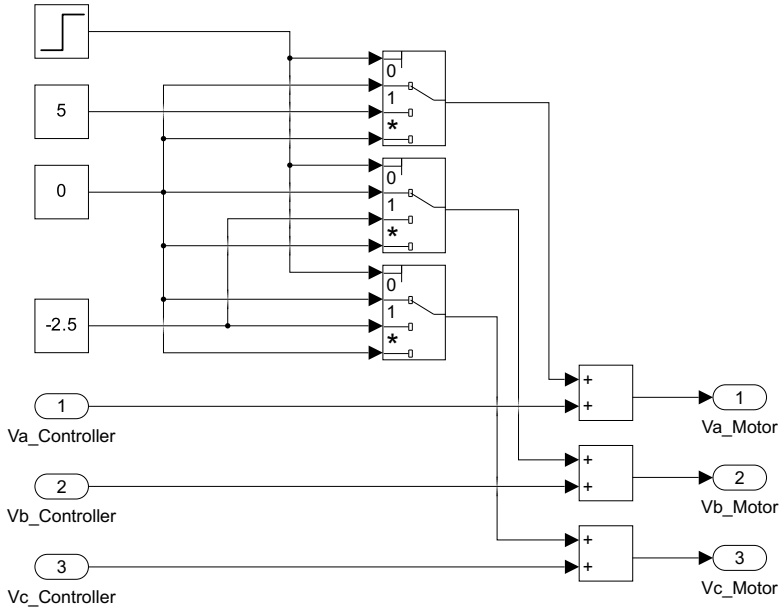


Figure 4.13.: Implementation failure mode unbalance

failure mode increases the overall friction in the system and can therefore be added as an additional resistance torque. The only difference between the two failure modes is the way it is getting triggered.

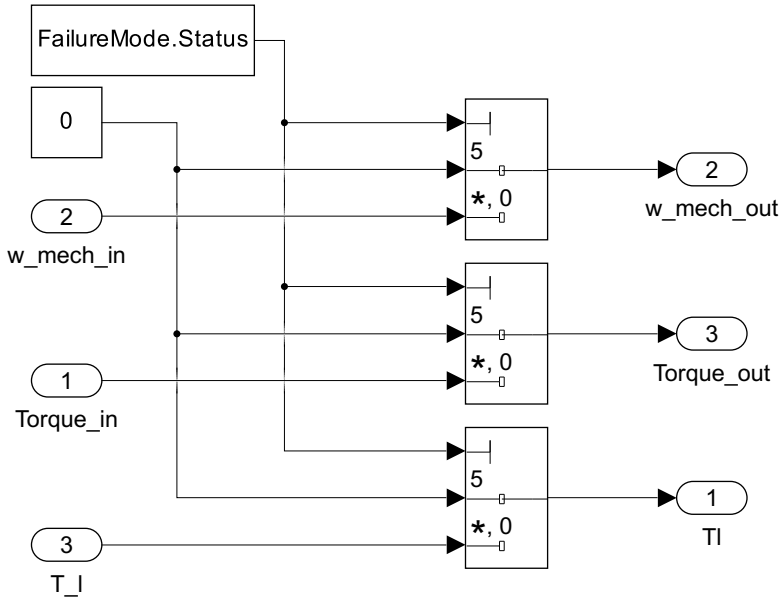


Figure 4.14.: Implementation failure mode broken stem

The failure mode **broken stem**, with its implementation displayed in figure 4.14, is implemented similarly to the missing phase failure mode. The failure mode leads to the situation that the valve actuation system after the motor will not receive any torque or rotational speed and, therefore, cannot feed back the resistance torque it generates, leading to all three signals being zero.

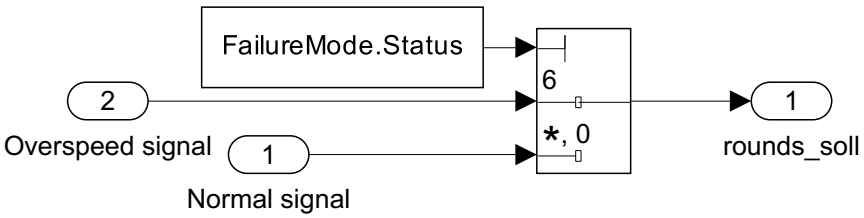


Figure 4.15.: Implementation failure mode overspeed

The failure mode **overspeed**, displayed in figure 4.15, changes the used setpoint position signal, which reaches the necessary rotations in a shorter time, which in turn leads to an increased rotational speed.

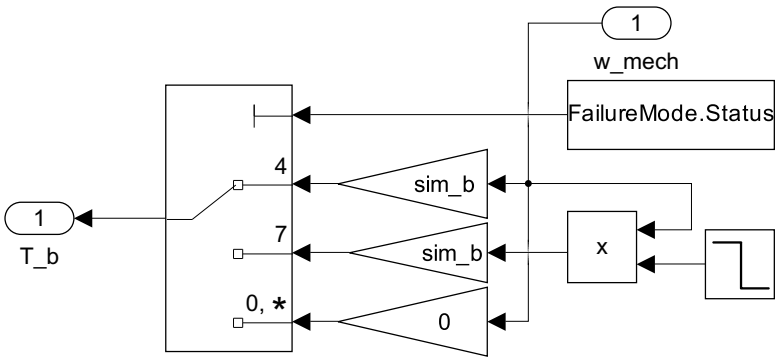


Figure 4.16.: Implementation failure mode increased friction

Failure Mode **increased friction**, displayed in figure 4.16, will be fed into the motor's mechanical system as additional friction besides the regular friction. It uses the same approach as the failure mode wear down, the only difference being that the failure mode increased friction resets the friction to zero after three seconds, while the failure mode wear down does not. That is why the step function is only connected to one of the multiport inputs and not to both.

5. Implementation of safety demonstration

This chapter provides information about how the model is tested under requirements following a safety demonstration. It explains how the digital twin and the control system representation are operated to generate the results.

Figure 5.1 displays the whole system used for the safety demonstration. The motor is implemented in the model reference block labeled *DigitalTwin_motorModell*. The controller representation used to control the motor is implemented in the model reference block *DigitalTwin_ControllerModell*. The other subsystems displayed in the figure are part of the failure modes or implemented for diagnostic reasons. The use of reference blocks makes it easier to exchange both models for more detailed or different models in future projects.

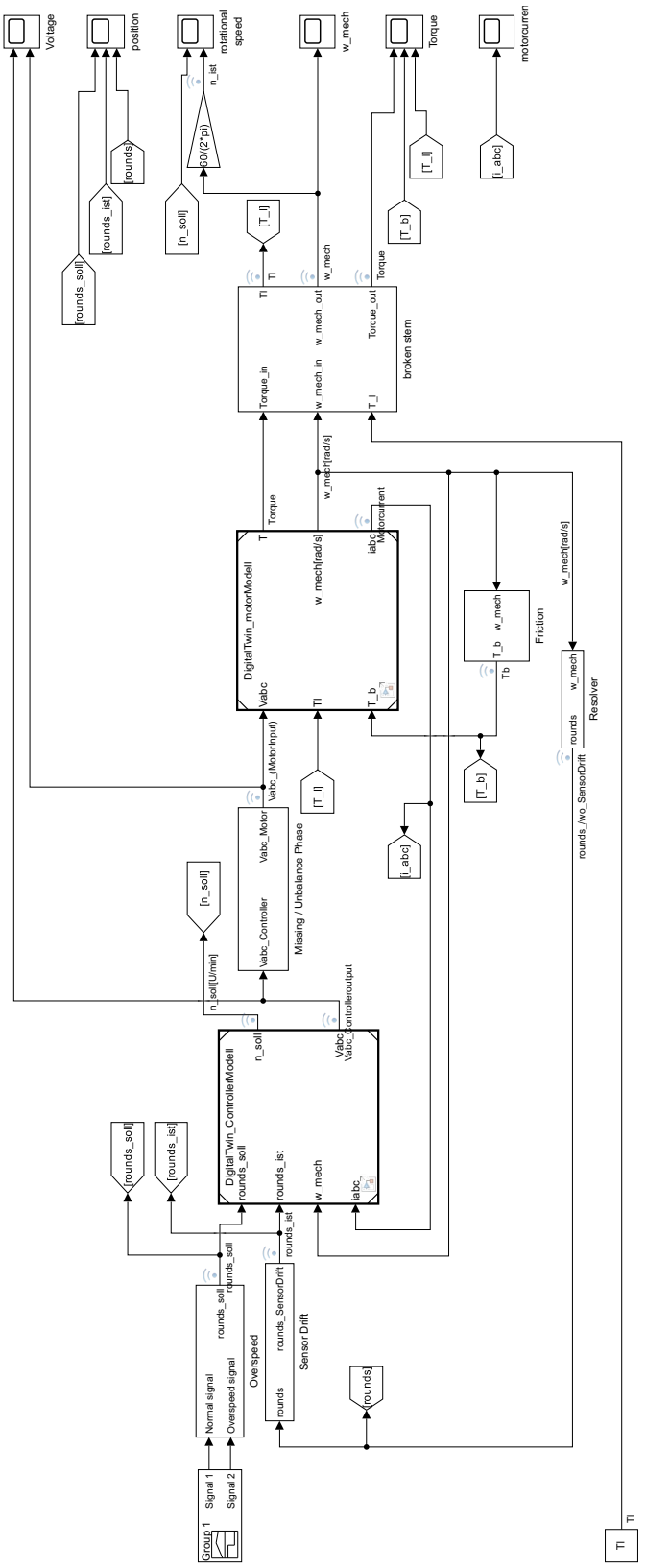


Figure 5.1.: Model

5.1. Test execution and failure modes

With the introduction of the test strategy in Section 3.10.1 on page 33, this section continues on this topic and explains how the tests are going to be executed

To trigger the failure modes for the digital twin, a Matlab m-script (Appendix A.2) is used. The script can adjust which failure mode should be simulated by adjusting the variable *FailureMode.Status* inside of it. Depending on the variable's value, going from zero to seven, a different failure mode will be simulated. Depending on the failure mode, this will lead to one and up to four simulations of the respective failure mode. Each time with different values for the failure.

The Value **zero** for the variable will lead to a regular run of the model without any failure modes.

The Value **one** for the variable will lead to the failure mode **sensor drift**. As figure 4.11 on page 42 displays, the failure mode contains four different values and, therefore, four different executions. Each execution has its own value. The step function, which is connected to the multiport switch, will change from zero to the necessary value depending on which execution is simulated right now. This step happens four seconds after the simulation starts and, therefore, in the second time frame. The first execution will add an offset of 0.25 rotations to the signal, which will be fed back as the actual position signal. The second execution is lowering the rotation count by 0.02. In the third execution, the offset increases over time, starting at 0.0 and increases till the end of the simulation to 0.5. The fourth execution lowers the offset with a falling signal, starting at 0.0 and lowering to -0.1 rotations. The two stable values are implemented to simulate an offset to the actual position. The two increasing and decreasing signals are implemented to simulate the behaviour if the resolver slowly starts to drift away from the actual value over time.

The third value **two** will start the simulation for the failure mode **missing phase**. This failure mode contains two simulations. The first simulation should display the behaviour of the motor when it should start turning but is only supplied with two phases from a three-phase power supply. The second simulation should simulate what happens when two power supply phases get cut, lost or broken in the middle of rotating. For this, the simulation simulates a cut of the connection by multiplying the value zero with the signal, as everything multiplied with zero is zero. The figure 4.12 on page 42 displays the architecture behind this failure mode. For the first simulation, the displayed step function on the left steps up after one second (first time frame). Therefore, it sets the signal to zero before the controller generates any signal. For the second simulation, the step function to the right is stepping up after four seconds (second time frame), setting the signal to zero

while the motor is rotating.

The fourth value **three** will start the failure mode **unbalanced phases**, which leads to the situation, that the altitudes between all three phases differ more then they should. Figure 4.13 on page 43 displays the architecture behind this failure mode. 0.05 seconds after the motor is supplied with its three-phase power supply, the step function triggers the multiport switches. This leads to an increase in altitude in phase V_a of about 5V and a decrease in the other two phases of about 2.5V each. A difference of 10V is already enough to simulate the wanted behaviour.

The fifth value **four** will start the failure mode **wear down**. This failure mode simulates the behaviour when the system ages and the bearings and the stem are no longer fully functional. The typical behaviour of this is increased friction in the system. To simulate this, an additional resistance torque in the form of friction is added to the system.

The sixth value **five** will start the failure mode **broken stem**. As displayed in figure 4.14 on page 43, as long as the value for *FailureMode.Status* is not equal to five, the respective signals are fed through the displayed multiport switches. But when the failure mode is executed, a signal with the value zero is transmitted through the multiport switch instead of the respective signal. In this failure mode, a broken stem refers to the connection between the motor and the valve actuation system after the motor. If this stem is broken, no torque or rotational speed is transmitted to the valve actuation system. Additionally, no resistance torque from the valve actuation system is fed back to the motor.

The seventh value **six** will start the failure mode **overspeed**. Overspeed, as displayed and described in figure 4.15 on page 44, is triggered when *FailureMode.Status* is equal to six. Then the multiport switch feeds a different setpoint signal to the controller. This setpoint signal has a steeper incline than the regular signal and reaches the end position faster, leading to more rotational speed.

And the eight value **seven** will start the failure mode **increased friction**. When the variable *FailureMode.Status* equals seven, triggering the failure mode for increased friction. This failure mode uses a different friction coefficient at the beginning of the movement. After five seconds of simulation, which means three seconds after the motor started moving, the step function decreases to zero and changes the friction coefficient to zero. Which in turn changes the friction in the system.

After each execution, the simulation recording will be saved to the hard drive for further comparison. This comparison is made to the simulation results from the execution under normal working conditions. Compared to this, the system's behaviour will be analysed, and recommendations for changes will be introduced.

6. Results

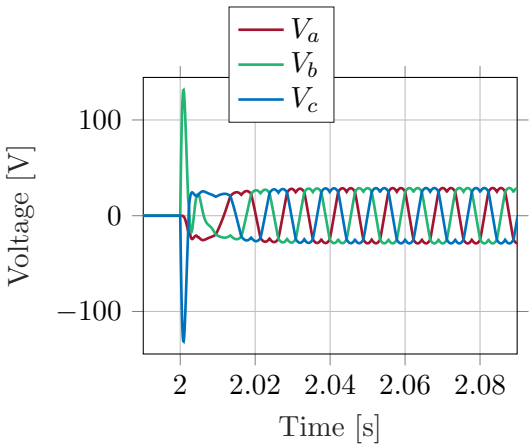
This chapter provides the results of the test plan execution. It displays the results of each failure mode and mentions the first impression about the results.

After executing the test plan, each failure mode was saved separately in a *.mldatx* file, generated by the Simulink program *Simulation Data Inspector*. These files were used to compare the results with an execution under normal conditions.

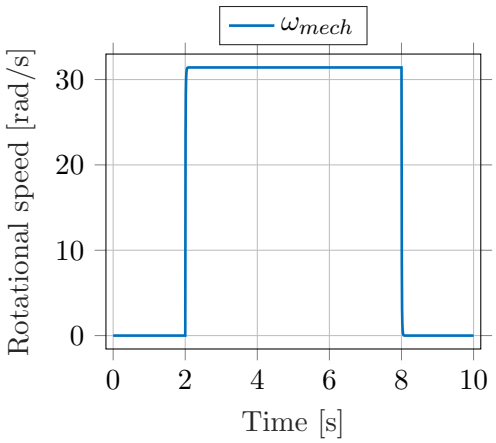
6.1. Results under normal working conditions

Figure 6.1 displays the behaviour of the system under normal working conditions. The sub figures display the three-phase power supply the motor is supported with, the rotational speed in $[rad/s]$ the motor is generating, the torque the motor is generating, the current the motor is generating, the comparison between the rotational speed in $[rpm]$ the motor should execute and is actually executing and lastly the comparison between the rounds the motor is taking and should take. The controller which in the end controls the motor is provided with the rounds signal. From this signal generates the controller the necessary voltage to turn the motor.

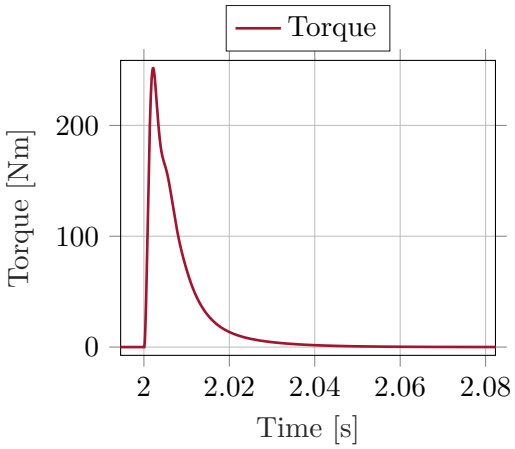
With an increase of five rotations per second (30 rotations in six seconds) or 300 rotations per minute, the controller is generating the necessary voltage for the motor. To overcome the change from zero rotations per second to five rotations per second the controller is generating a current of almost 100A and therefore a voltage spike of over 100V and lowers the necessary voltage and current afterwards to the amount needed to keep the motor moving. Based on this spike the simulated motor is generating a torque of over 200Nm. With the voltage and current falling directly afterwards again, the torque is also falling to a lower amount. This behaviour happens as there are no protective functions implemented in the digital twin. Without any protective functions this behaviour can seriously damage controller and motor respectively.



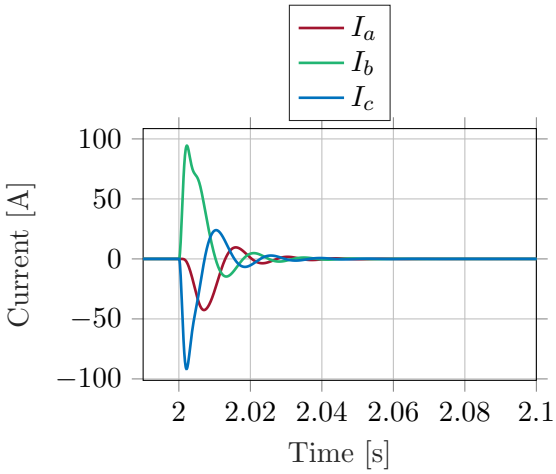
(a) Voltage



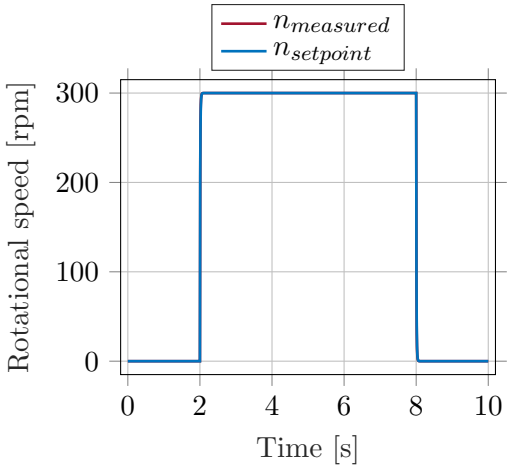
(b) ω_{mech}



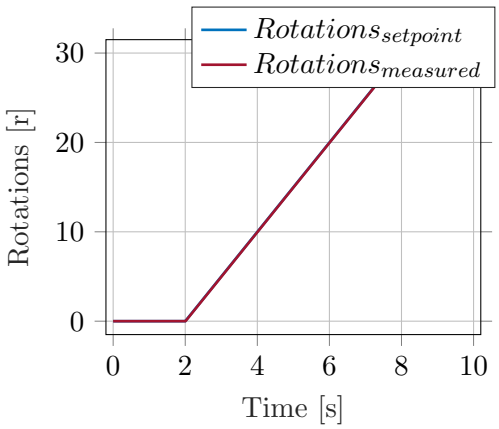
(c) Torque



(d) Current



(e) Rotational speed



(f) Rotations

Figure 6.1.: Run under normal condition

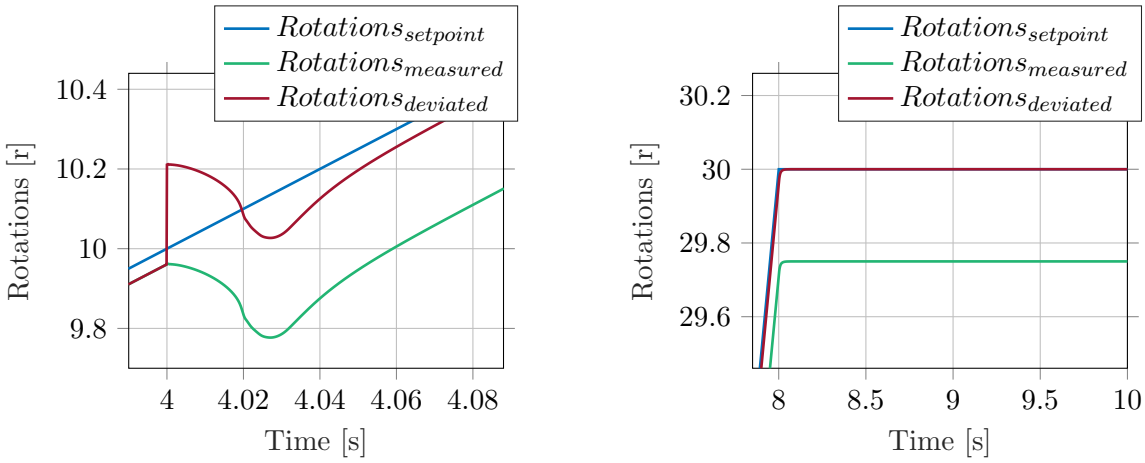
Visible in the fact that the motor is not generating any current in its stable conditions, it is evident that the motor is running in idle state. Therefore, when further work, based on this thesis, gets more knowledge about the behaviour of the valve actuation system, each one of this tests will need to be executed again to get more clearer results.

6.2. Results of failure modes

After stating the results of execution under normal operation conditions, this section introduces the results of each failure mode.

6.2.1. Sensor drift

The following four figures 6.2, 6.3, 6.4 and 6.5 display the relevant results from the failure mode sensor drift. The displayed three signals contain the setpoint position ($Rotations_{setpoint}$), the deviated position ($Rotations_{deviated}$) and the measured position ($Rotations_{measured}$) that the motor has. Due to the deviation in each case of the failure mode, the controller only faces the difference between the setpoint signal and the deviated signal, it assumes the deviated signal is the measured position signal.



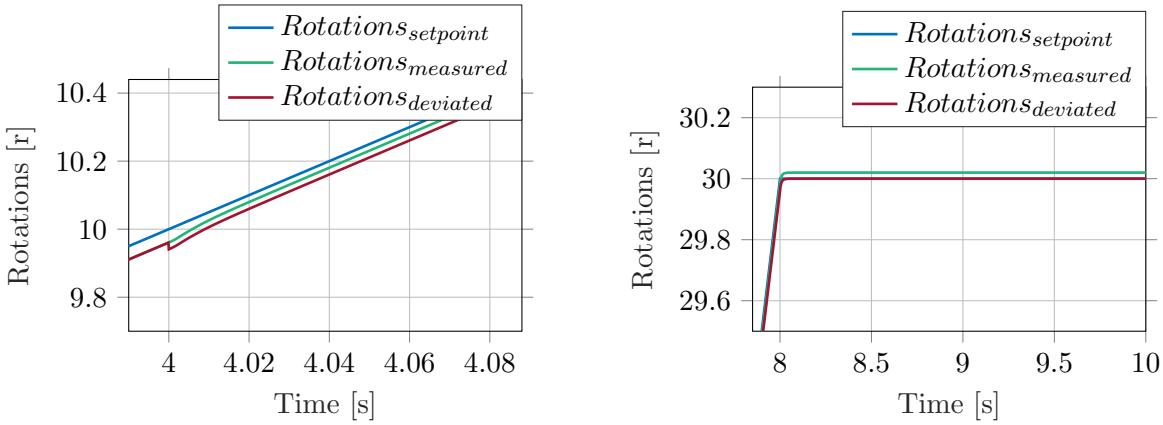
(a) Activation

(b) Resulting deviation

Figure 6.2.: Results sensor drift first case

The first case (figure 6.2) is a positive offset of around 0.25 rotations. When the failure mode is triggered, the feedback signal, $Rotations_{measured}$, is increased by 0.25. But it is directly corrected by the controller. As the controller only experiences the deviated signal, the actual position is also corrected. This, in turn, results in a deviation after reaching

the 30th rotation. Therefore the controller assumes the motor turned its 30 rotations, but it only rotated for 29.75 rotations.

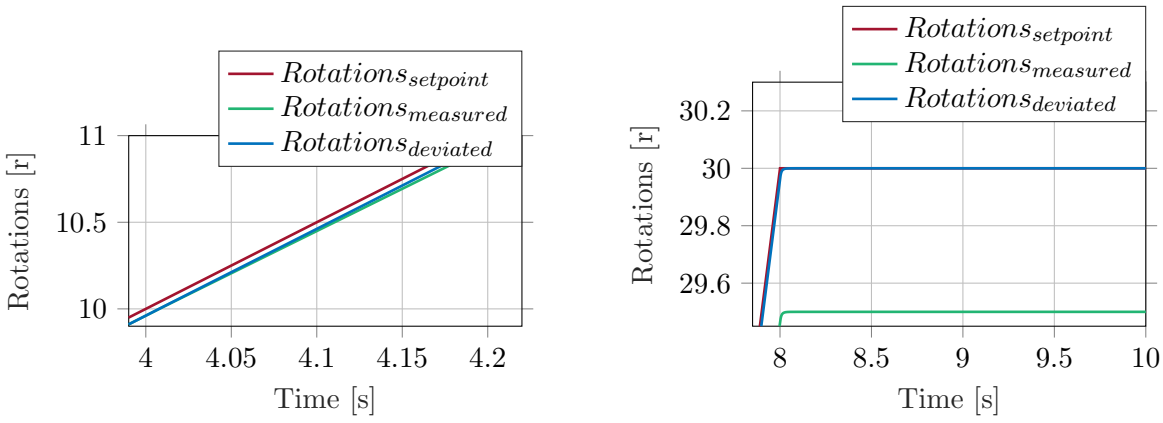


(a) Activation

(b) Resulting deviation

Figure 6.3.: Results sensor drift second case

The same goes for the second case (figure 6.3). This deviation is a negative deviation of 0.02 rotations. A negative deviation leads to a situation where the motor turns for more rotations than it should. The deviation in itself is minimal, but the fact that the deviation exists is already an issue for which this failure mode is implemented. Due to this failure mode, the motor did reach a position of 30.02 rotations at the end of the simulation.



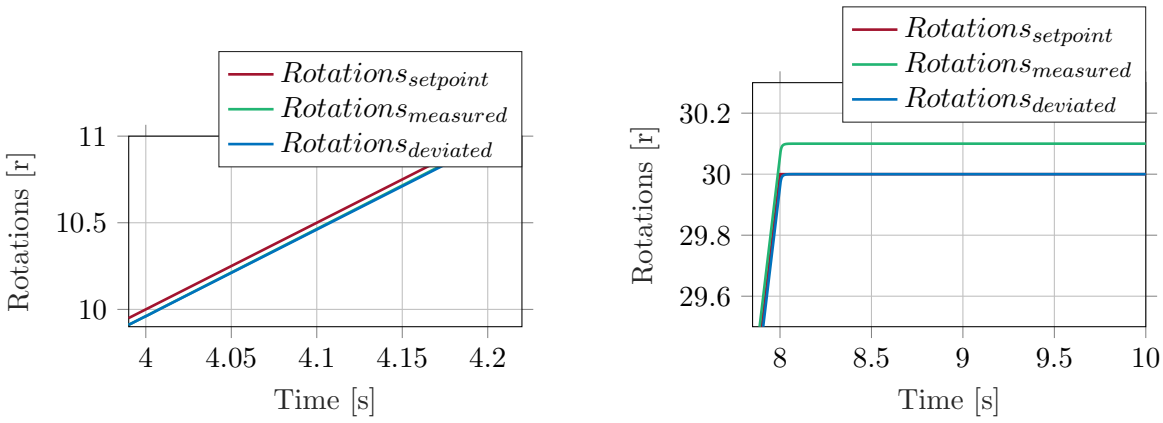
(a) Activation

(b) Resulting deviation

Figure 6.4.: Results sensor drift third case

The intention of the third case (figure 6.4) is to display the system's behaviour when the deviation increases over time. To simulate this behaviour, the offset slowly rises from the four seconds mark till the eight seconds mark before staying there with a deviation of 0.5

rotations, which means that the motor only turned for 29.5 rotations.

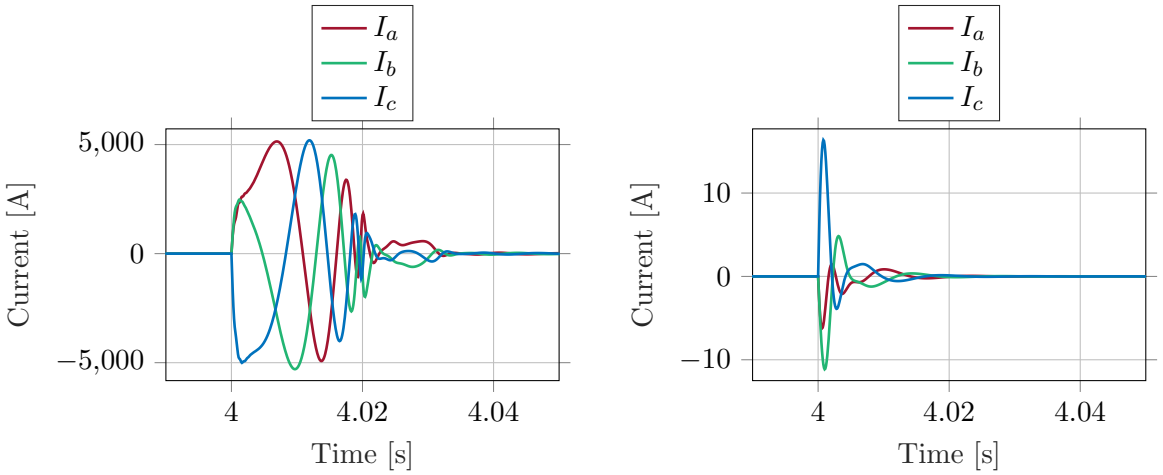


(a) Activation

(b) Resulting deviation

Figure 6.5.: Results sensor drift fourth case

The fourth case (figure 6.5) displays the last case in this failure mode. Here the deviation was implemented with a slowly falling deviation starting from the fourth second and ending at the eighth second. This leads to a deviation of around 0.1 rotations. Due to the small deviation, the deviation in itself is barely visible in the beginning of the movement.



(a) Current first case

(b) Current fourth case

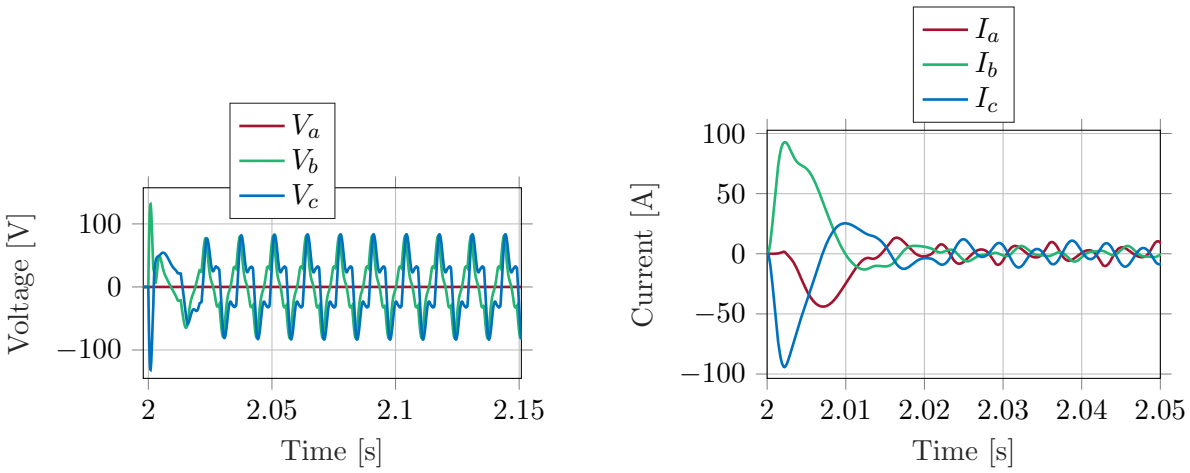
Figure 6.6.: Current

Just by looking at the last four cases, the failure mode sensor drift might be a good way on how to implement a deviation of the sensor signal. But figure 6.6 displays the system's current behaviour. The left figure shows the first case, and the right figure the fourth case. When the deviation of 0.25 rotations is implemented, the controller tries to negate the

deviation, generating a current of over 5,000A, which could seriously damage the whole system. And even the implementation of a slowly increasing deviation leads already to a current of 10 to 15A. Which is still below the nominal current, but it is not good for the motor.

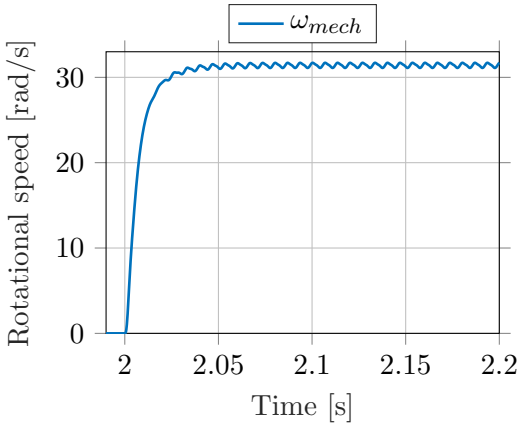
6.2.2. Missing phase

The first missing phase case is where the motor starts turning but only two out of the three-phase supply voltage are able to supply voltage to the motor. This situation is displayed in figure 6.7a. It is visible that one phase stays zero while the other two try to generate the necessary voltage, but the controller is unable to generate a stable voltage. This unstable voltage leads to the rippling behaviour of the rotational speed and torque, as displayed in the figures 6.7c and 6.7d. Additionally, as displayed in figure 6.7b, the current of the motor is also quite unsymmetrical, which burdens the motor further. Therefore, even though the motor is able to operate, it will end up being destroyed prematurely.

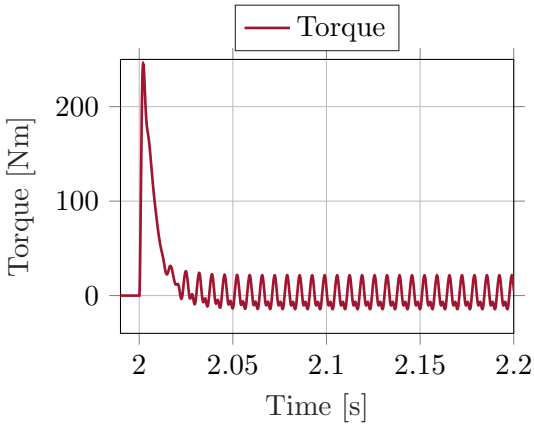


(a) Voltage

(b) Current



(c) ω_{mech}



(d) Torque

Figure 6.7.: Results missing phase first case

The second case from the failure mode missing phase is when the motor is already turning, but two out of the three phases supplying it with voltage are lost. The system cannot comprehend this situation and generates a fast-changing and huge voltage in the last phase. Which ends the simulation prematurely. With the simulation ending prematurely it is safe to assume that the digital twin is unable to continue operate.

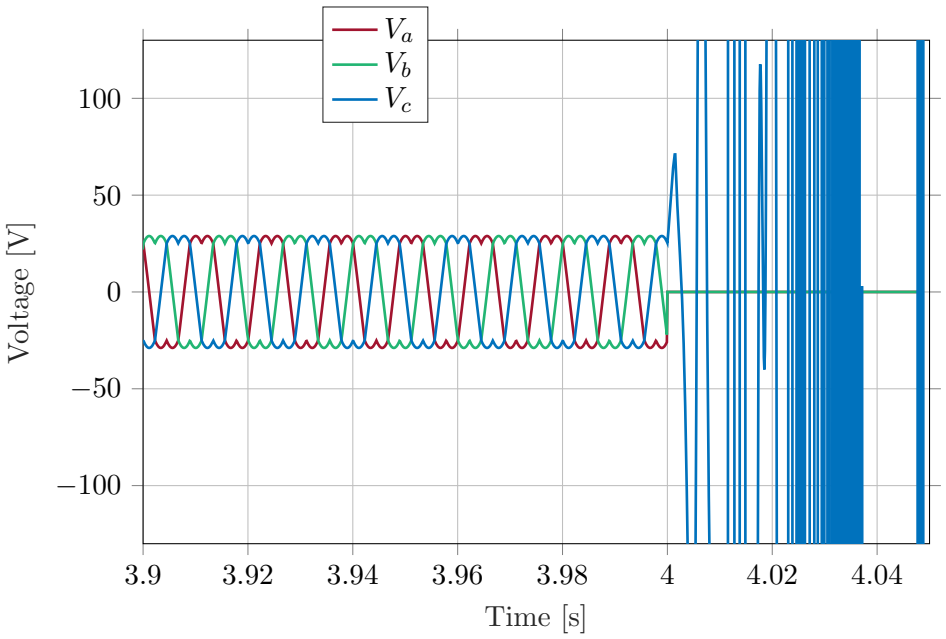
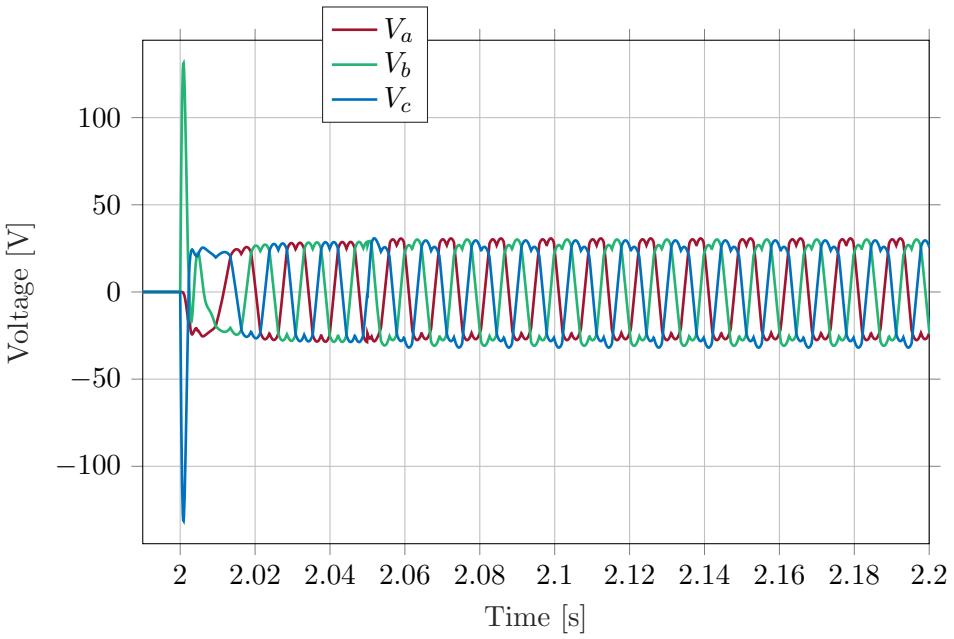


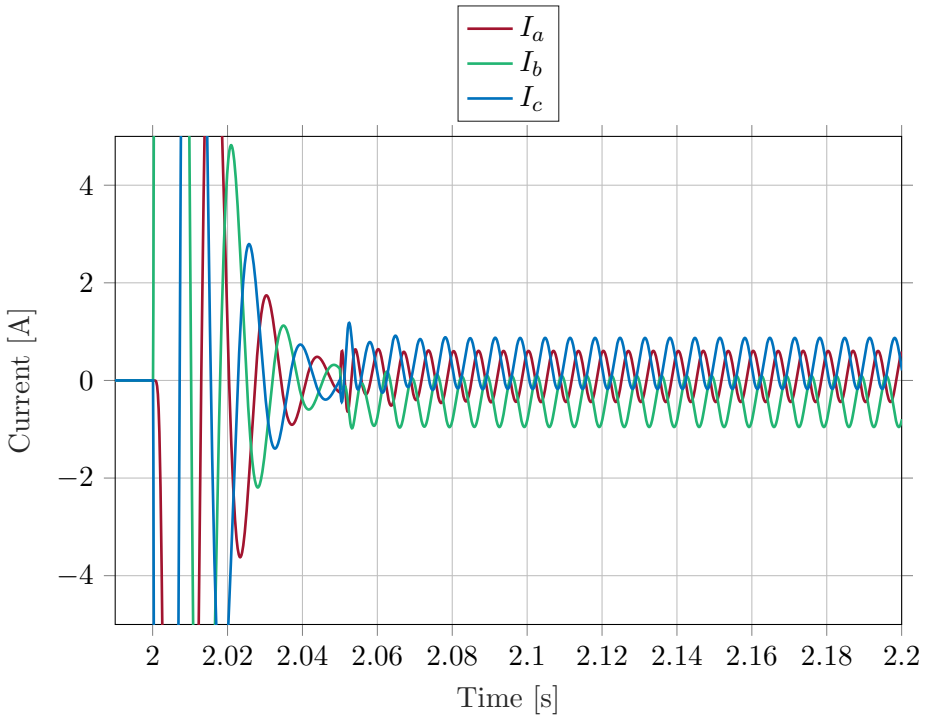
Figure 6.8.: Voltage, results missing phase second case

6.2.3. Phase unbalance

The failure mode phase unbalance results, displayed in figure 6.9, displays the voltage and current behaviour. Comparing to figure 6.1d on page 50 after reaching the stable condition the motor current should be basically zero. But, due to the system's phase unbalance, the system is experiencing a current of around 1A. This doesn't sound like much, but due to this unbalanced current, as already stated when the failure modes were discussed the first time (chapter 3, on page 14), the temperature inside the motor is rising, and this leads to a shorter lifetime and prematurely system failure.



(a) Voltage

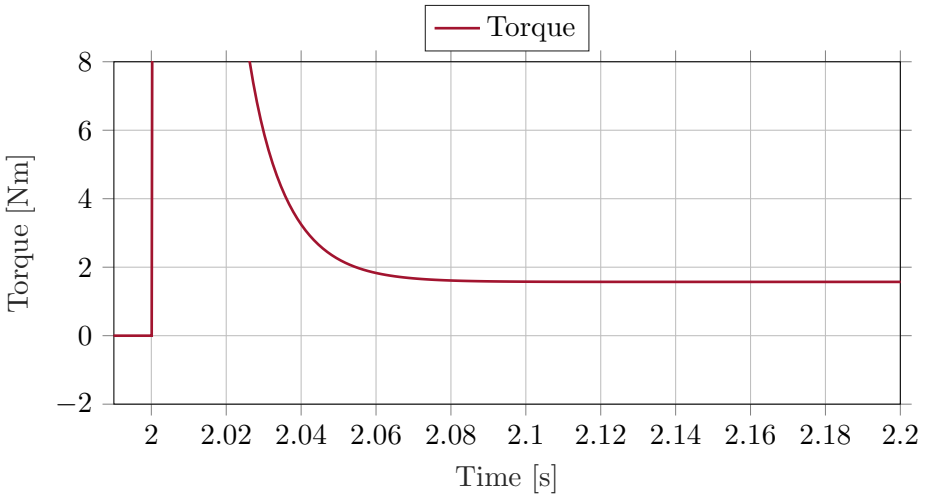


(b) Current

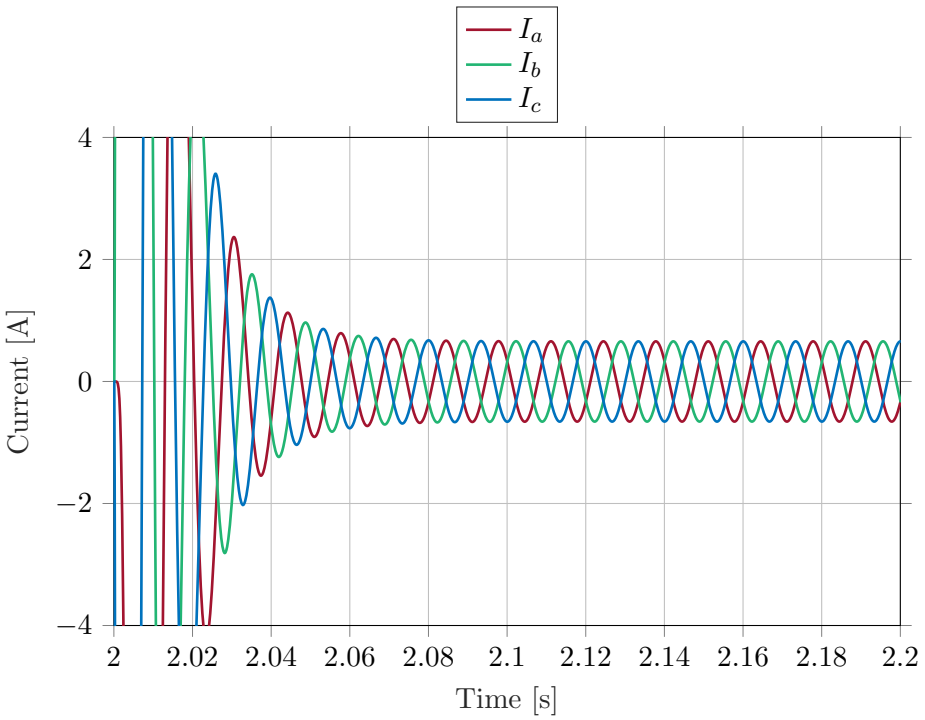
Figure 6.9.: Results phase unbalance

6.2.4. Wear down

The failure mode wear down results show an increased torque overall. Compared to figure 6.1c, the most visible is it after the motor reaches his stable conditions. Where under normal conditions the torque is zero, in the failure mode wear down the torque is almost 2Nm. This leads to an increased current in the motor of around 0.5A (see figure 6.10b). Differently compared to the phase unbalance failure mode, as this current is not unbalanced but in fact balanced, it is not as much of a burden to the motor. In fact, this is the normal behaviour of the motor when it is not operating in idle mode.



(a) Torque

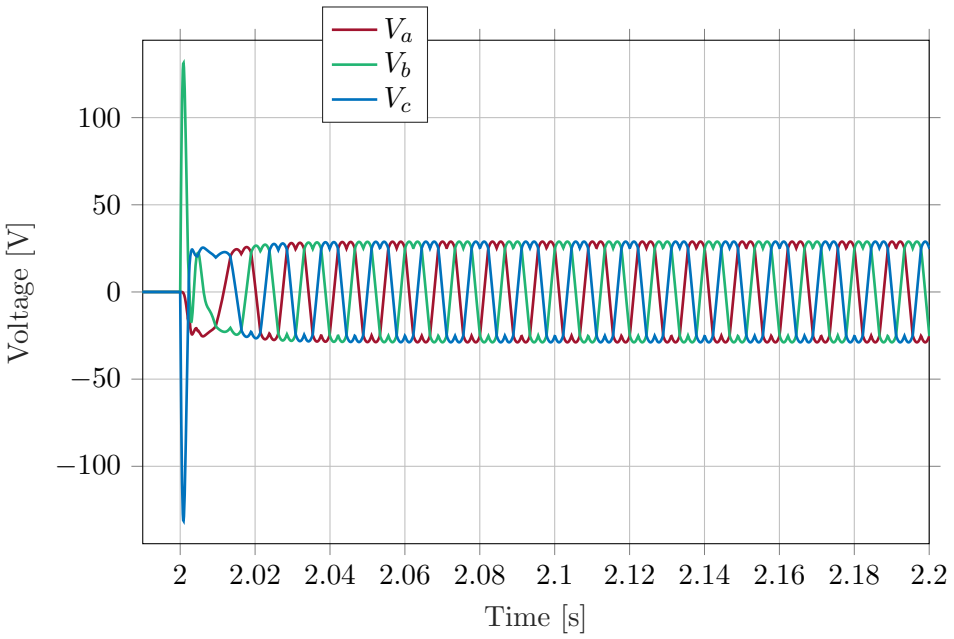


(b) Current

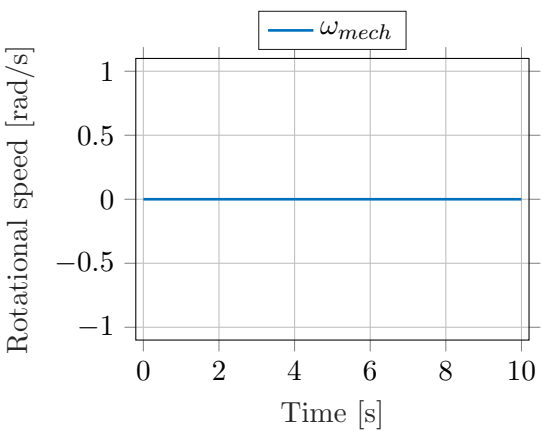
Figure 6.10.: Results wear down

6.2.5. Broken stem

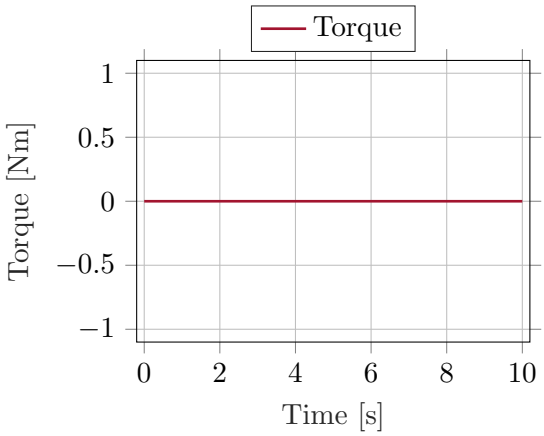
The results of the failure mode broken stem show that even though the motor is supplied with a three-phase power supply, the rotational speed and the torque stay zero. For reference this is intended, as the failure mode should display the behaviour of the system when the connection between motor and the valve actuation system is broken.



(a) Voltage



(b) w_{mech}

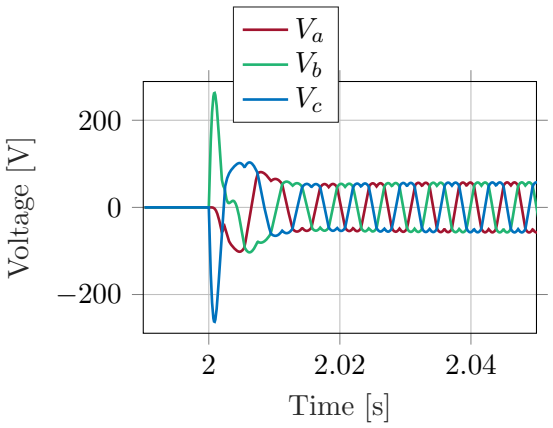


(c) Torque

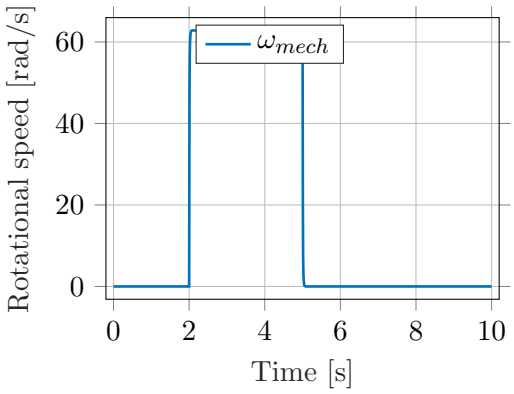
Figure 6.11.: Results broken stem

6.2.6. Overspeed

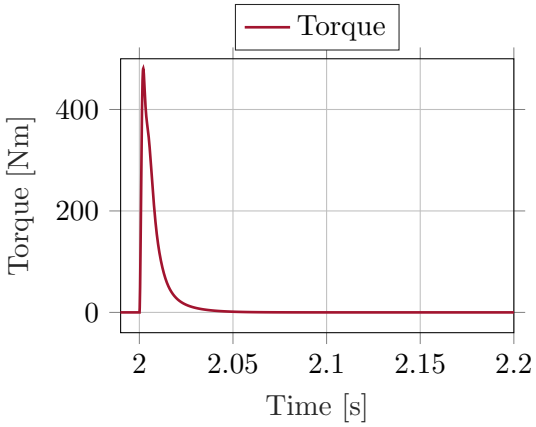
Compared to figure 6.1, which displays the execution of the model under normal working conditions, the following figure 6.12 shows the closing conditions on the system in half the time. The failure mode Overspeed is specified to simulate the closing conditions in half the time from the assumed normal working conditions. This change leads to a situation where almost all values in the system double.



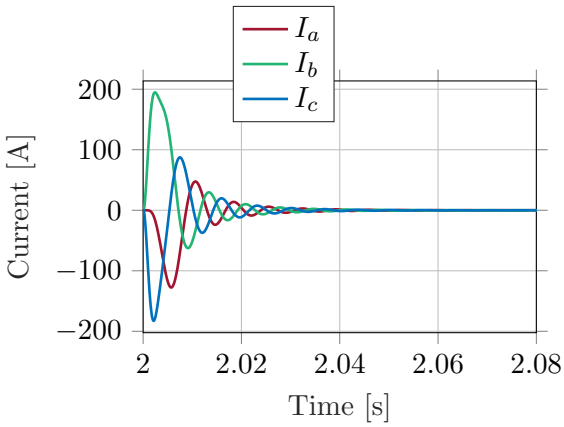
(a) Voltage



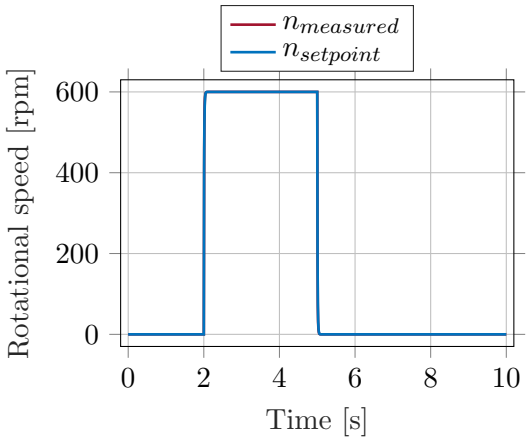
(b) ω_{mech}



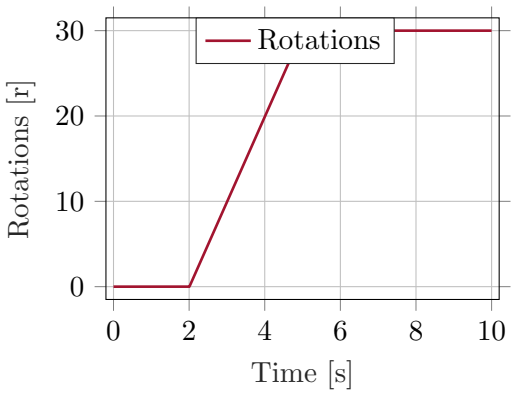
(c) Torque



(d) Current



(e) Rotational speed

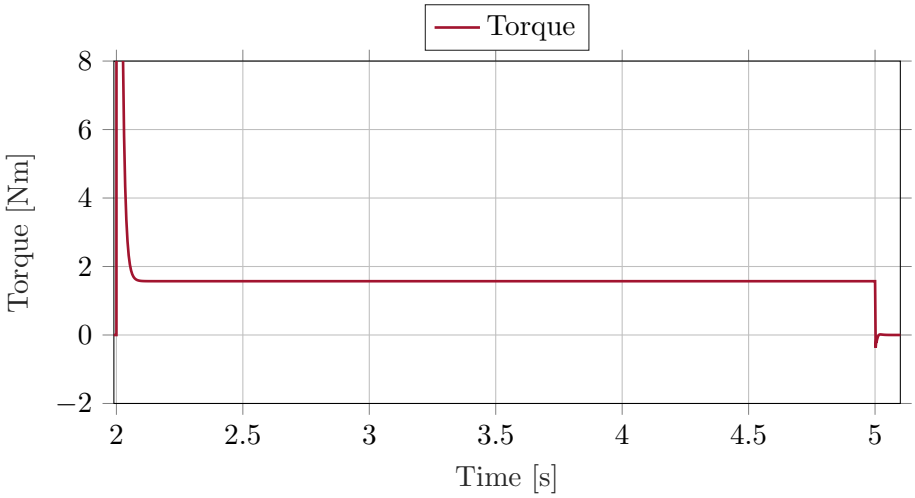


(f) Rotations

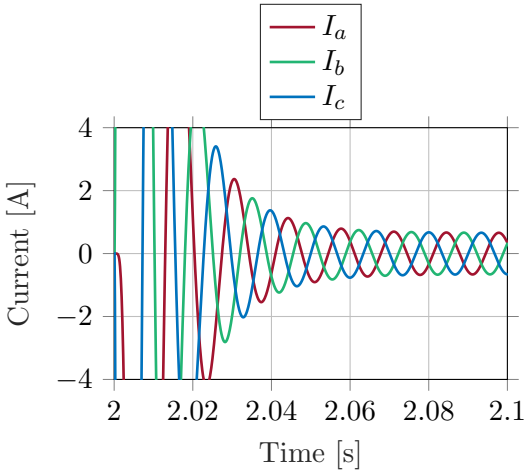
Figure 6.12.: Results overspeed

6.2.7. Increased friction

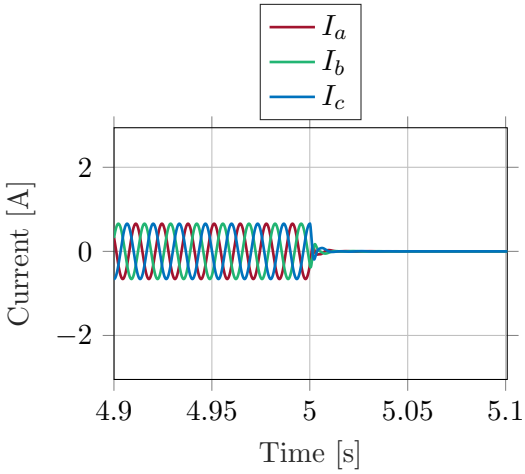
The last failure mode which was looked at was increased friction. The results displayed in figure 6.13 show the increased torque needed to compensate for the friction in the system. At the same time to generate the torque, the motor generates a current. With the disappearance of the friction, the motor does not need to generate the increased torque anymore, which in turn lets the current drop to zero, as it is also no longer needed.



(a) Torque



(b) Current



(c) Current around second five

Figure 6.13.: Results increased friction

7. Discussion

This chapter discusses the results introduced in the previous chapter and further discusses how reliable these results are. Furthermore, it continues the discussion from the failure modes to the objective statements and the contribution to the project.

7.1. Results of the test plan

Overall, the test plan confirmed the system's expected behaviour made under this thesis's assumptions.

The biggest concern about the failure modes is the spike in voltage and current when the motor starts working. It has nothing to do with any failure mode, as it is also displayed in the results from the test under normal conditions. These spikes above 100V and a little below 100A can seriously damage the motor. Even though the voltage is still around the nominal voltage, the current is five times higher than the nominal current. To prevent the motor from getting damaged due to these spikes, it is recommended to implement a protective function that measures the voltage the motor receives and the current it generates based on this voltage. This protective function should also limit the voltage the controller can provide to the motor. The changes based on implementing this protective function would also lower the torque the motor generates to move the valve actuation system. Even though the protective function would allow the motor to survive the start, it must also be mentioned that the motor is starting and moving slower with this protective function. This would also lead to a change of the setpoint signal of the controller to prevent the protective function from constantly being in use. Without a change in the setpoint signal, the controller tries to change the motor above its working conditions and therefore, the protective function needs to mitigate that. Therefore, it is recommended to change the setpoint signal to prevent that from happening and lessen the burden on the protective function.

The results of implementing the failure mode **sensor drift** do not look as anticipated. The idea was to simulate a deviation of the position signal, which worked as expected.

But the backlash of the motor current needed to mitigate the change was not anticipated. The deviation was too wide for the controller, and without any protective functions, any real motor would have been destroyed. This situation shows a perfect example of using a digital twin in a safety demonstration. The implementation of the protective function, discussed on the previous page, would lead to a slower deviation adjustment. Considering the consequences of motor damage, a slower adjustment is the lesser evil.

Another way to mitigate the problem would be to change the controller to be even less aggressive. But this would need further adjustments and redefinitions depending on when the closing movement becomes too slow for closing in time.

The failure mode **missing phase** mainly showed the expected behaviour. The motor cannot work when two of its three power supply phases are lost. The fact that the motor starts turning with only two phases was not anticipated because, usually, under these conditions, the motor would need an additional external starter. The normal situation would be that if the motor is already running, two phases are enough to let the motor continue to work, but without a starter, the motor should be unable to start. Losing two of the three power supply phases mid-turn will lead to a nonfunctional situation, as anticipated and displayed in the results.

The failure mode **phase unbalance** displays the anticipated failure conditions. Even though the failure mode influences the system, the motor can still function. Because long-term exposure to this failure mode can seriously damage the motor, it would be suggested to fix the issue as fast as possible when it gets recognised. It is not safety critical but a hazard that can destroy the motor over time.

The failure modes **wear down** and **increased friction** also work as anticipated. The only effects of these failure modes are the increased torque needed to be generated by the motor, which leads to a higher motor current. This higher torque compensated for the friction torque created by the friction. The increased motor current is not safety critical, but for recognising these failure modes, it might be helpful to watch the torque and the current of the motor, which are its outputs.

The last failure mode **overspeed** shows mostly the expected behaviour. Due to the situation that the closing signal would close the valve in half the usual time, all significant parameters doubled, such as voltage, current and rotational speed. This would lead to severe damage to the motor. It can be mitigated by the already mentioned protective functions and watching the inputs and outputs.

7.2. Objective statements

At the beginning of the thesis, the questions *what specific requirements are placed on the digital twin to be useful for safety demonstration?*, *how to decide on what needs to be modelled?* and *what are the constraints of using such a model for safety demonstration?* were asked.

Requirements placed on the digital twin include that the digital twin can demonstrate the behaviour of the real motor sufficiently enough for the intended purpose. The requirements in this thesis contained the electrical and mechanical dependencies of the motor and its interactions with the motor control unit. These requirements include the absolute basic model of the motor. Based on this model, it is possible to create test scenarios and discuss the result of these test scenarios. With these results and the discussion as a basis, it is possible to discuss further what the model and the motor control unit are missing to be able to be used for a safety demonstration. This work routine is based on the phrase *as much as necessary, but as less as possible*. Therefore, the model only contains information about what is required and not everything that is possible. Further requirements include how the digital twin is used in the safety demonstration. Is it used in a real-time demonstration, like overlooking a system and reacting to specific circumstances? Or is it used to predict situations in the future based on the information at hand?

One constraint about using a digital twin for safety demonstration is that the model will only cover what the developer models it to cover. Suppose the developing process didn't include one specific failure, and therefore, this failure wasn't included in the model. In that case, the model cannot simulate this failure and, thus, cannot cover it in the safety demonstration. The same goes for unexpected real-life scenarios. If the specific scenario is not covered in the model, the digital twin might be unable to compensate for or even recognise it. This scenario continues in that everything needs to be explained in a way that can be modelled. It cannot be simulated if it can't be described in mathematical relations. But that doesn't mean that a test with an actual prototype would not be able to perform it.

Regarding the reliability of the test plan results, under the assumptions made in this thesis, the results are reliable. But for the results to be used in the discussion for safety demonstration, the model and the failure modes need to be further adapted. Additionally, this thesis covers nowhere near enough information to reason the implementation of PMSM in the technology. Therefore, it can be said that the results are reliable, but only in the context of the thesis.

7.3. Protective functions

Based on the discussion about the result of the test plan in the previous chapter, it is recommended to implement a control function which overlooks the inputs and outputs of the motor. From the protective functions discussed in table 3.5 in an earlier chapter of this thesis, the protective functions *over/under voltage* and *circuit breaker* should at least be implemented. These two protective functions cover the voltage input of the motor and prevent that voltage is neither too high nor too low for operation. These protective functions could prevent the voltage spike at the beginning of the movement. But before the circuit breaker protective function shuts down the connection to the motor, it is recommended to change the setpoint signal to prevent the spike from even happening.

The digital twin does not consider any other protective functions listed in that table. Failure modes covering failures which would make the implementation of these protective functions necessary were not part of the scope of this thesis, which does not mean that further work on this thesis can neglect these failures.

7.4. Use in overall research project

Based on the previous discussion, a few things need to be reconsidered before the digital twin created in this thesis can be used in the overall research project.

The first thing is the protective function. Under the assumed closing condition of the whole system, the motor would not be able to function correctly. The current spike of 100 amperes would damage, if not destroy, the motor. Therefore, implementing the protective functions discussed in the previous section is recommended.

The second thing would be a control function which measures the motor's inputs and outputs and can recognise the pattern which hints at the failure modes discussed in this thesis. Mostly, the current of the motor gives the most evident hint about which failure it could be. This implementation is not necessary, but it might be helpful in the long run to recognise failures more quickly and easily.

Besides these two missing parts, the thesis model is fully prepared to be used in the research. It might need some adjustments in the Matlab file regarding the inputs and outputs of the failure mode file, but nothing further needs to be adjusted.

8. Conclusion

Designing and creating a digital twin for safety demonstration requires a proper understanding of the fundamentals of the digital twin itself and the influential behaviour the surrounding has on it. This thesis demonstrates the implementation of the motor of a subsea safety valve which is implemented under an all-electric approach as a digital twin. The motor is controlled through a cascade controller and generates an output which is used to operate the valve actuation system to close the valve. Neither the valve nor the valve actuation system is part of this thesis. Furthermore, the digital twin is able to simulate failures to test the control system and display how the motor and the control system behave under these failures.

The digital twin was created using the fundamental information which was accessible to develop the model. The failure modes which were implemented into the thesis were based on a FMECA. The thesis then further discusses the failure modes' impact and suggests ways to mitigate them. In the end, this thesis displays the first step of creating a digital twin for the safety demonstration of the motor of a subsea all-electrical safety valve.

The results of this thesis are important for an ongoing PhD research project which is part of the SUBPRO centre. Additionally, it displays the basic work on which further projects can continue working on creating a digital twin for a safety demonstration.

9. Recommendations for further work

Multiple points can be further improved based on the results shown in this thesis.

The first thing which should be done is the implementation of the discussed protective functions. This includes the implementation of a control function which controls the input and output of the motor, a limiter which limits the output of the controller, and therefore the input of the motor to its nominal parameters or at least to values which will not destroy the motor and lastly the change of the setpoint signal of the position controller and the control behaviour of all the controllers themselves to a less aggressive behaviour.

The failure modes discussed in this thesis must also be further adjusted. In the failure mode sensor drift, it is recommended to change the deviation itself, either by lowering the deviation or trying different approaches for the failure mode. In the failure mode missing phase, it is recommended to look further into the start with only two phases and with and without an external starter. Furthermore, it is recommended to design a control function which recognises the situation when the connection of one, two or all three phases is lost. It should either be able to mitigate the situation or stop the system's operation before it can seriously damage itself. For the failure mode phase unbalance, the recommendation is to create a protective function which is at least able to recognise the failure and depending on the situation, might even be able to mitigate the unbalance until a maintenance crew can fix the problem.

Besides necessary adjustments in the failure modes, further recommendations can be made to the thesis. To keep the workload of this thesis in an acceptable range, the coverage of the parameters was lowered. The motor parameters in this thesis were assumed to be perfectly exact, but due to manufacturing tolerances, it is recommended to look again at the parameter with a deviation of up to ten per cent. This will allow to account for any influence which originates from a deviation. The same applies to the control system used in this thesis. It was assumed that the control system was a cascade field oriented control. But it is also possible to use different controller approaches, like a field weakening control. Therefore it is also recommended to look into using different control systems.

Bibliography

[1] Ruth Bolton et al. 'Customer experience challenges: bringing together digital, physical and social realms'. In: *Journal of Service Management* 29 (Oct. 2018). DOI: 10.1108/JOSM-04-2018-0113.

[2] 'digital twin - Explore - Google Trends'. In: (2022). URL: https://trends.google.com/trends/explore?date=today%5C%205-y&q=digital%5C%20twin.

[3] DNV. *Safety 4.0 DNV*. [Online; accessed 15-December-2022]. DNV. 2022. URL: https://www.dnv.com/research/energy/safety-40.html.

[4] Michael Grieves. 'Digital Twin: Manufacturing Excellence through Virtual Factory Replication'. In: (Mar. 2015).

[5] Michael Grieves. *Virtually Perfect: Driving Innovative and Lean Products through Product Lifecycle Management*. Nov. 2011. ISBN: 0982138008.

[6] Michael Grieves and John Vickers. 'Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems'. In: *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*. Ed. by Franz-Josef Kahlen, Shannon Flumerfelt and Anabela Alves. Cham: Springer International Publishing, 2017, pp. 85–113. DOI: 10.1007/978-3-319-38756-7_4. URL: https://doi.org/10.1007/978-3-319-38756-7_4.

[7] Meine van der Meulen et al. *Demonstrating safety of software-dependent systems*. DNV AS, 2022.

[8] Mikey likes mountains. *Space vector modulation - space vector modulation — Wikipedia, The Free Encyclopedia*. https://commons.wikimedia.org/w/index.php?curid=5882951. [Online; accessed 22-November-2022]. 2009.

[9] *Permanent Magnet synchronous Motor Control*. English. freescale. 2012. 3 pp.

[10] Mark Purdy et al. 'How Digital Twins Are Reinventing Innovation'. In: (Jan. 2020). URL: https://sloanreview.mit.edu/article/how-digital-twins-are-reinventing-innovation/.

[11] Marvin Rausand and Mary Ann Lundteigen. *Chapter3 FMECA*. English. Version 0.1. [Online; accessed 16-December-2022]. 2004. URL: https://www.ntnu.edu/ross/books/sis/slides.

[12] JACOB RÖING and CARL JENSEN. 'Modelling and design of PMSM position drivesusing model following control'. [Online; accessed 23-November-2022]. MA thesis. 2021. URL: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-299844.

[13] Embedded Society. *Modeling and Simulation of a Permanent Magnet Synchronous Motor "PMSM"*. [Online; accessed 13-December-2022]. Youtube. 2021. URL: https://www.youtube.com/watch?v=cCibYNBowl8.

[14] *Subsea Trees — Aker Solutions*. [Online; accessed November 17, 2022]. 2022. URL: https://www.akersolutions.com/globalassets/images_1920x1080/aasta_hansteen_installation_1920x1080.jpg.

[15] TheMathworks. *Perform transformation from three-phase (abc) signal to $\alpha\beta 0$ stationary reference frame or the inverse - Simulink - MathWorks Nordic*. https://se.mathworks.com/help/sps/powersys/ref/abctoalphabetazeroalphabetazerotoabc.html. [Online; accessed 12-December-2022]. 2013.

[16] TheMathworks. *Perform transformation from three-phase (abc) signal to dq0 rotating reference frame or the inverse - Simulink - MathWorks Nordic*. https://se.mathworks.com/help/sps/powersys/ref/abctodq0dq0toabc.html. [Online; accessed 12-December-2022]. 2013.

[17] Wikipedia. *Space vector modulation — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Space_vector_modulation. [Online; accessed 22-November-2022]. 2022.

[18] WizataSA. 'Difference Between Digital Twin, Digital Model, and Digital Shadow'. In: (Feb. 2022). URL: https://www.wizata.com/knowledge-base/difference-between-digital-twin-digital-model-and-digital-shadow.

[19] Abdelhalim Zekry and Ahmed Abdalrahman. 'Digital Control Techniques for Grid-Connected Inverters'. PhD thesis. Jan. 2013.

A. Appendix

A.1. Matlab Code

Listing A.1: Parameter script

```
clc;
clear;
%% Parameter PMSM
Ld = 1.31e-3;                   %[H]          % Indunctance in d-axis
Lq = 1.77e-3;                   %[H]          % Indunctance in q-axis
id_ref = 0;
k = 2.759;                      %Nm/sqrt(W)]% motor constant
P = 30;                                       % Number of pols /
                                              % number of Polpairs * 2
PM = 2*k /(3*sqrt(2)*P/2);                    % Permanentmagnet
Rs = 0.112;                     %[Ohm]        % Resistance in Phase
J=0.06;                         %[kgm^2]      % Inertia Motor
Tl = 0;                         %Load Torque
b=0;                            %Damping Constant
sim_b = 5e-2;

%% Initiate Variable with 0 for testing
FailureMode.Status=0;
FailureMode.SensorDrift = 0;
FailureMode.MissingPhase = 0;
FailureMode.Unbalanced = 0;
```

Listing A.2: Execution Matlab script

```
clc;
clear;
%% Parameter PMSM
Parameter;
```

```
%% Define Failure Mode:
FailureMode.Status=0;
%NormalRun=0;
%SensorDrift=1;
%MissingPhase=2;
%PhaseUnbalance=3;
%WearDown=4;
%BrokenStem=5;
%Overspeed=6;
%IncreasedFriction=7;

%% Failure Modes:
% Load model
Model = 'PMSM_Controller_Motor_Failuremodes_per_rounds';
load_system(Model)

switch FailureMode.Status
    case 0
        % NormalRun
        disp('Normal Run')
        % Modify run naming rule
        Simulink.sdi.setRunNamingRule('Normal Condition Run <
            run_index>')
        % Simulate system
        sim(Model)
    case 1
        % Run with SensorDrift
        disp('Failure Mode Sensor Drift')
        % Modify run naming rule
        Simulink.sdi.setRunNamingRule('Failure Mode Sensor Drift
            Run <run_index>')
        %Increased
        FailureMode.SensorDrift = 1;
        % Simulate system
        sim(Model)
        %Decreased
        FailureMode.SensorDrift = 2;
        % Simulate system
```

```
41          sim(Model)
42          % steadily increasing
43          FailureMode.SensorDrift = 3;
44          % Simulate system
45          sim(Model)
46          %steadily decreasing
47          FailureMode.SensorDrift = 4;
48          % Simulate system
49          sim(Model)
50      case 2
51          % Run with MissingPhase
52          disp('Failure Mode missing phase')
53          % Modify run naming rule
54          Simulink.sdi.setRunNamingRule('Failure Mode missing
                phase Run <run_index>')
55          % First Time Frame
56          FailureMode.MissingPhase = 1;
57          % Simulate system
58          sim(Model)
59          % Second Time Frame
60          FailureMode.MissingPhase = 2;
61          try
62              % Simulate system
63              sim(Model)
64          catch
65              sprintf('error occured')
66          end
67      case 3
68          % Run with Unbalanced Phases
69          disp('Failure Mode unbalanced phases')
70          % Modify run naming rule
71          Simulink.sdi.setRunNamingRule('Failure Mode unbalanced
                phases Run <run_index>')
72          FailureMode.Unbalanced = 1;
73          % Simulate system
74          sim(Model)
75
76      case 4
```

```
77          % Run with WearDown
78          disp('Failure Mode Wear Down')
79          % Modify run naming rule
80          Simulink.sdi.setRunNamingRule('Failure Mode Wear Down
                Run <run_index>')
81          b = sim_b;
82          % Simulate system
83          sim(Model)
84      case 5
85          % Run with Broken Stem
86          disp('Failure Mode Broken Stem')
87          % Modify run naming rule
88          Simulink.sdi.setRunNamingRule('Failure Mode broken stem
                Run <run_index>')
89
90          % Simulate system
91          sim(Model)
92      case 6
93          % Run with Overspeed
94          disp('Failure Mode Overspeed')
95          % Modify run naming rule
96          Simulink.sdi.setRunNamingRule('Failure Mode Overspeed
                Run <run_index>')
97
98          % Simulate system
99          sim(Model)
100     case 7
101         % Run with increased friction in movement
102         % StickSlip
103         % Run with increased friction in the beginning, friction
                is lowered
104         % after some movement
105         disp('Increased Friction')
106         % Modify run naming rule
107         Simulink.sdi.setRunNamingRule('Increased Friction Run <
                run_index>')
108         % Simulate system
109         sim(Model)
```

```
110
111     otherwise
112         % NormalRun
113         disp('Normal Run')
114         % Modify run naming rule
115         Simulink.sdi.setRunNamingRule('Normal Condition Run <
                run_index>')
116 end
117 % Modify run naming rule
118 Simulink.sdi.setRunNamingRule('Failure Mode 1 Run <run_index>')
119 %% Last Row
120 % Save to File and clear the Data Inspector
121 % Save with Timestamp and which failure mode got executed
122 string = sprintf('FailureModeRuns/Failuremode_%.0f_%s.mldatx',
        FailureMode.Status, datestr(now));
123 % Necessary Formatting due to the fact that Simulink.sdi.save
        cannot handle
124 % spaces or : in its filename
125 string = replace(string,' ','_');
126 string = replace(string,':','_');
127
128 Simulink.sdi.save(string)
129 Simulink.sdi.clear
130 Simulink.sdi.setSubPlotLayout(1,1)
131 % print a confirmation message that the run is done and the file
        is saved
132 sprintf('Run saved')
```

A.2. FMECA

Description of Unit			Description of failure		
Ref no	Function	Operational mode	Failure Mode	Failure cause or mechanism	Detection of failure
Motor	Import 3 phase	running	Phase Unbalance	Differnet resistance in wire	Control of inputs of motor
				Issue with battery management system	BMS
			unstable power supply	Battery management system	BMS
			short circuit	loose wire	Control of inputs of motor
			Phase missing	broken wire	Control of inputs of motor
	generate torque	running	no resistance	broken stem	Control of inputs and outputs of motor
			increased resistance	foreign particle in well	Control of inputs and outputs of motor
				corroded rotor bar	Control of inputs and outputs of motor
				dynamic eccentricity	Control of inputs and outputs of motor
				stator eccentricity	Control of inputs and outputs of motor
			Partial discharge	loose wire	Control of inputs and outputs of motor
			Leakage current / Earth failure	loose wire	Control of inputs and outputs of motor
			overvoltage	Battery management system	Control of inputs and outputs of controller
	generate rotational speed	running	higher resistance torque	increased friction in the system afterwards	Control of inputs and outputs of motor
			no resistance	broken stem	Control of inputs and outputs of motor
			overspeed	turning to fast	Control of inputs and outputs of motor
Controller	Control position	running	wrong sensor signal	resolver not working properly	Control of inputs and outputs of controller
			no feedback signal	broken wire	Control of inputs and outputs of controller
	Control rotational speed	running	wrong sensor signal	resolver not working properly	Control of inputs and outputs of controller
			no feedback signal	broken wire	Control of inputs and outputs of controller
	control iq	running	wrong sensor signal	measurement unit not working properly	Control of inputs and outputs of controller
			no feedback signal	broken wire	Control of inputs and outputs of controller
	control id	running	wrong sensor signal	measurement unit not working properly	Control of inputs and outputs of controller
			no feedback signal	broken wire	Control of inputs and outputs of controller
	generate 3 phase	running	Phase missing	broken wire	Control of inputs and outputs of controller
			gate-drive circuit failure	PWM faulty	Control of PWM generation
Environment	Temperature	idle, running	operation above thermal limits	increased friction till non functioning	Measurement of temperature
	Stable standing	idle, running	Vibration	loose screw	Vibration measurement

Figure A.1.: FMECA Part 1

Effect of failure		Severity ranking	Risk reducing measures	Comments
on the subsystem	on the system function			
damaging of hardware	can lead to system failure	7	Maintenance	
damaging of hardware	can lead to system failure	7	Reset BMS	
damaging of hardware	can lead to system failure	8	Reset BMS	
-	system failure	10	Maintenance	
-	system failure	10	Maintenance	
-	system failure	10	Maintenance	
-	increased friction, system needs to generate more torque	5	install filter in well	
damaging of hardware	increased friction, system needs to generate more torque	5	Maintenance	
damaging of hardware	increased friction, system needs to generate more torque	5	Maintenance	
damaging of hardware	increased friction, system needs to generate more torque	5	Maintenance	
-	can lead to system failure	8	Maintenance	
-	can lead to system failure	8	Maintenance	
damaging of hardware	can lead to system failure	8	Reset BMS	
-	increased friction, system needs to generate more torque	5	Maintenance	
-	system failure	10	Maintenance	
damaging of hardware	can lead to system failure	7	control unit which limits supply	
damaging of hardware	can lead to system failure	8	Maintenance	
-	system failure	10	Maintenance	
damaging of hardware	can lead to system failure	8	Maintenance	
-	system failure	10	Maintenance	
damaging of hardware	can lead to system failure	8	Maintenance	
-	system failure	10	Maintenance	
damaging of hardware	can lead to system failure	8	Maintenance	
-	system failure	10	Maintenance	
-	system failure	10	Maintenance	
damaging of hardware	can lead to system failure	8	control unit for PWM	
damaging of hardware	can lead to system failure	6	Temperatur control and Maintenance	
damaging of hardware	depending on severity, can lead to system failure	4	Maintenance	

Figure A.2.: FMECA Part 2