Lappeenranta-Lahti University Of Technology LUT
School of Energy Systems
Electrical Engineering

Krister Gräsbeck

SIMULATION OF FIELD DEVICES IN TESTING OF A MARINE CONTROL SYSTEM

Examiners: Prof. Olli Pyrhönen
Dr. Tuomo Lindh

ABSTRACT

Lappeenranta-Lahti University of Technology LUT
School of Energy Systems
Electrical Engineering

Krister Gräsbeck

SIMULATION OF FIELD DEVICES IN TESTING OF A MARINE CONTROL SYSTEM

Master's Thesis
2021
61 pages, 23 figures.

Examiners: Prof. Olli Pyrhönen
Dr. Tuomo Lindh

Keywords: PLC, testing, simulation, marine control system

Testing the programmable logic controller (PLC) software part of a control system before the system is commissioned and all the field devices are available and functional is challenging. In this thesis a simulation tool is developed to simulate the devices that are part of a Danfoss Editron marine system. The types of simulated devices include power electronic converters, electric motors, generators, and other devices managed by the control system. The simulator is a Python program which reads the outputs of a Beckhoff PLC, generates feedback signals and writes them to the inputs of the PLC. By simulating the field devices, the PLC software can be tested in an office environment before the system commissioning on the marine vessel.

Two different vessels are simulated to verify the simulations and to determine the feasibility of using the simulation tool. Two errors were found in the control logic of the first vessel which had already been commissioned previously. The second vessel was simulated to test the logic and HMI panel functionality before the commissioning. Consequently, several errors were found and fixed. The extended testing made possible by the simulation tool was found to improve software quality, save time spent during the commissioning phase, and to decrease the costs of commissioning the vessel.

TIIVISTELMÄ

Lappeenrannan-Lahden teknillinen yliopisto LUT
School of Energy Systems
Sähkötekniikka

Krister Gräsbeck

KENTTÄLAITTEIDEN SIMULOINTI LAIVAJÄRJESTELMÄN OHJAUKSEN TESTAUKSESSA

Diplomityö
2021
61 sivua, 23 kuvaa.

Tarkastajat: Prof. Olli Pyrhönen
TkT Tuomo Lindh

Hakusanat: PLC, testaus, simulointi, ohjausjärjestelmä

Ohjausjärjestelmän osana olevan ohjelmoitavan logiikan eli PLC:n ohjelmiston testaus on haastavaa ennen kuin järjestelmää ollaan ottamassa käyttöön ja järjestelmän osana olevat toimilaitteet ja anturit ovat toiminnassa. Tässä diplomityössä kehitetään työkalu Danfoss Editronin laivajärjestelmän laitteiden simulointiin. Järjestelmän simuloituja laitteita ovat mm. taajuusmuuttajat, sähkömoottorit ja generaattorit. Simulaattori on Python-kielinen ohjelma, joka lukee Beckhoff PLC:n lähdöt, luo palautesignaalit ja kirjoittaa ne PLC:n tuloihin. Simuloimalla kenttälaitteet voidaan logiikan testausta suorittaa toimistoympäristössä ennen aluksen käyttöönottoa.

Simuloinnin toimintaa ja hyödyllisyyttä arvioidaan simuloimalla kaksi eri laivajärjestelmää. Ensimmäisessä tapauksessa, jo toimitetusta projektista, löytyi simuloinnin avulla kaksi ohjelmistovirhettä. Toisessa tapauksessa simuloinnin avulla testattiin logiikan sekä ohjausnäyttöpaneelin toimintaa ennen aluksen käyttöönoton aloitusta. Useita virheitä saatiin paikallistettua ja korjattua. Simuloinnin mahdollistama laajennettu testaus todettiin parantavan ohjausohjelmiston laatua, lyhentävän käyttöönottovaiheessa kuluvaa aikaa sekä pienentävän käyttöönoton kustannuksia.

PREFACE

I would like to thank the examiners of this thesis, Professor Olli Pyrhönen and Dr. Tuomo Lindh, for the feedback and guidance during the making of this thesis. To my supervisor at Danfoss Editron, Dr. Risto Tiainen, thank you for the initial hiring to spend the summer at the company almost two years ago and the subsequent opportunities to work with interesting marine projects which inspired this thesis. I appreciate the support and guidance received in making of this thesis. Also, thank you to all my co-workers at Editron for the great atmosphere and working environment.

Thank you to my friends Matti, Sakke, Ossi, J-P, and all the other fine folks I have gotten to know during the studies at LUT for the memories and time spent together.

Lastly, thank you to my family for the unconditional love and support, and of course to my dog Börje who is a very good boy and an immense source of joy.

Lappeenranta, March 19, 2021.

Krister Gräsbeck

CONTENTS

LIST OF ABBREVIATIONS & SYMBOLS

Abbreviations

AC	Alternating current
ADS	Automation Device Specification
AoA	Angle-of-attack
BLDC	Brushless DC
BMS	Battery management system
CPP	Controllable pitch propeller
DC	Direct current
ECS	Editron Control System
EMF	Electromotive force
FAT	Factory acceptance testing
FB	IEC61131-3 function block
FBD	Function block diagram
FOC	Field-oriented control
FPP	Fixed pitch propeller
FUN	IEC61131-3 function
HAT	Harbor acceptance testing
HIL	Hardware-in-the-loop
HMI	Human-machine interface
HW IO	Hardwired input/output
IAS	Integrated Automation System
IGBT	Insulated gate bipolar transistor
IL	Instruction list
LD	Ladder diagram
LNG	Liquefied natural gas
OPC	Open Platform Communications
PLC	Programmable logic controller
PMSM	Permanent-magnet synchronous machine

POU	Program Organization Unit
PRG	IEC61131-3 program
PTI	Power take-in
PTO	Power take-off
PWM	Pulse-width modulation
RTOS	Real-time operating system
SAT	Sea acceptance testing
SFC	Sequential function chart
SPWM	Sinusoidal pulse-width modulation
ST	Structured text
SVPWM	Space vector pulse-width modulation
VC	Virtual commissioning

Symbols

$\boldsymbol{T}_{\mathrm{C}}$	Clarke tranformation matrix
$\boldsymbol{T}_{\mathrm{P}}$	Park tranformation matrix
$\boldsymbol{T}_{\mathrm{dq0}}$	dq0 tranformation matrix
δ_{s}	Load angle
$\dot{m}$	Mass flow rate
η	Efficiency
ω	Angular velocity
Φ	Magnetic flux
ψ_{s}	Stator flux
ψ_{PM}	Permanent magnet flux
ρ	Density
τ	Time constant
A	Area
a	Axial inflow factor
C	Capacitance
E	Energy
e	Electromotive force
F	Force

f_{s}	Swithing frequency
i_{d}	Current, d-axis
i_{q}	Current, q-axis
i_{s}	Stator current
L	Lift force
L_{d}	Synchronous inductance, d-axis
L_{q}	Synchronous inductance, q-axis
L_{md}	Magnetizing inductance, d-axis
L_{mq}	Magnetizing inductance, q-axis
$L_{\mathrm{s}\sigma}$	Leakage inductance
m	Modulation index
N	Number of turns
n	Rotational speed
P	Power
p	Number of pole pairs
p	Pressure
s	Laplace variable
T	Torque inducing force
T	Torque
T_{e}	Electromagnetic torque
T_{s}	Sample, switching period
U	Voltage
u_{s}	Stator voltage
U_{DC}	DC-link voltage
U_{LL}	Line-to-line voltage
V	Volume
v	Velocity
W	Work
w	Fluid velocity

1 INTRODUCTION

Programmable logic controllers (PLC) are widely used to control processes and machines found in industrial environments such as factories [1]. Figure 1.1 shows a block diagram which depicts the relationships between a controller and a plant or process which is controlled. The plant has inputs and outputs. Inputs, with the means of actuators such as valves, motors, and relays, affect the state of the process. Outputs are process variables which are detected using sensors. Programmed in the controller is a control algorithm that takes the process outputs as inputs and generates outputs to drive the process as defined in the control algorithm. Above the process level is a supervisory level. On the level can be an automatic system which manages several controlled plants or a human operator, who uses a human-machine interface (HMI) to view the status of the process and change its operational parameters, for example. The HMI is often a touchscreen with a graphical user-interface.

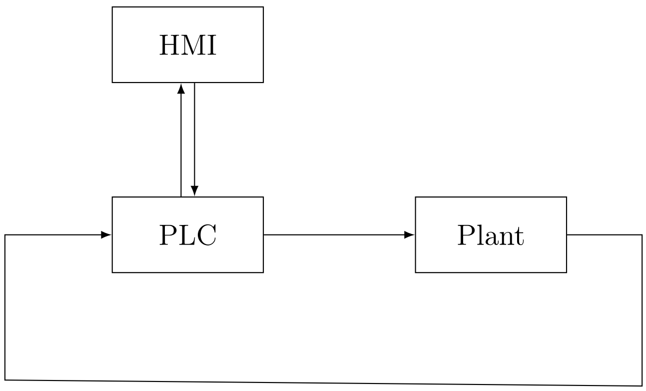


Figure 1.1: A control system which consists of a plant which is controlled by a PLC. HMI is used to view the status of the system and to provide controls for an operator.

When a new plant is built, the operation logic is defined and implemented in the programmable controller. Testing of the developed control software is needed to find errors. Straight-forward method of testing is to wait until installation work of the equipment is complete and to try whether the plant operates as intended. This poses the risk of erroneous logic operating the system in an unsafe manner. Fixing the logic takes time and can lead to delays and increased costs during the commissioning process. Performing testing before the commissioning has the problem of not having access to the actuators, process, and sensors. When the controller receives no feedback on its actions, the logic may go to a fault state, for example, instead of stepping through a sequence.

1.1 Background and motivation

Danfoss Editron provides electric and hybrid powertrain solutions for off-highway, on-highway, and marine applications. The Editron system for marine vessels, based on electric machines and power electronic converters, is a power system solution for series and parallel hybrid as well as full electric vessels [2]. The power system handles generation of electric power which is consumed in propulsion and other electric loads. The selling points of the system are compact design and good energy efficiency. Control software running on PLCs manages the individual field devices as a system.

Between vessel projects custom control software must be developed because each project has unique properties. The software development is done in the office based on design documents. Commissioning work takes place at the shipyard and can start when installation work of the controller cabinet, the controlled devices, and the cabling has been completed. When the inputs and outputs of the control system have been checked to work correctly, the system functionality is tested. Because of the dependence on hardware, the commissioning of the control software is in the final stages of the overall commissioning of the vessel. Working out problems at this stage can lead to delays in the commissioning schedule. For the company the commissioning work involves costs from travel and long working days. Delays affect negatively customer relations. Being able to test the functionality of the control system in the development phase has the potential of shortening the time required for commissioning and reducing the related costs.

1.2 Objectives and delimitations

The objective of this thesis is to develop a piece of software that enables efficient testing of the control software in an office environment by simulating the field devices. The simulated devices must respond to the PLC outputs and generate inputs that resemble the operation of the real system. The use of simulation should also be cost and time effective. The end goals of testing with the help of simulation are:

- Shortening of time required for control software commissioning.
- Reduction of costs of control software development.
- Improvement of control software quality.

Testing with simulated field devices is not to replace any testing that is done with the real system during commissioning. The level of detail captured by the simulation is limited to what is needed to test the control software. The power system is thus not simulated to the extent that it could be used to e.g. determine fuel consumption, analyze losses, or be used in system design.

1.3 Structure of the thesis

In the next chapter the theory of power systems on-board vessels is introduced. The third chapter deals specifically with the Editron marine system principles and components. The fourth chapter involves PLCs and their programming. The requirements for the simulations are formulated and options for implementing the requirements are touched. In the fifth chapter details on the developed simulation tool are presented. The sixth chapter shows how two vessel projects were used to first verify the simulations and then to test the second project before commissioning. Finally, the seventh chapter concludes this thesis.

2 SHIPBOARD POWER SYSTEMS

Marine vessels have been used by humans for thousands of years to transport people and goods across bodies of water. Vessels need a source of power to move forwards in the water. Human-powered boats date back to prehistoric times and the first sailing vessels to some 3000 years before common era. The advancements in building of sailing ships and navigation tools enabled exploration of the world and colonization which led to the Age of Sail starting from the 17th century when international trading began to rely on shipping, and naval warfare developed as leading powers built fleets of heavily armed Men-of-War. The Industrial Revolution and the development of the steam engine mechanized ship propulsion in the first half of the 1800s. Steel replaced wood as the ship building material. The beginning of the 20th century saw the introduction of steam turbines and coal was replaced by oil as fuel. The marine diesel engine was also created, and it is nowadays the most common engine type in merchant vessels [3].

In addition to propulsion, vessels need power to run service loads such as lighting, air conditioning, heating, galley appliances, water pumps, cargo handling and other electric equipment. The term hotel load is used for electric loads that are mainly needed to support the people on-board. Propulsion is the largest consumer and the other loads vary with vessel type. Passenger-carrying vessels, especially cruise ships have large hotel loads due to potentially thousands of people on-board. The propulsion power consumed by a car ferry with passenger amenities comparable to a cruise ship operating in the Baltic Sea equates to about half of the total energy consumption. The other half is shared by heating and electric loads [4]. Tankers need electric power for their powerful cargo handling pumps. Bow and stern transverse thrusters that vessels use for maneuvering are most commonly electric.

In this chapter general theory of marine propulsion is introduced. The interaction between propellers and water is described using the change in the momentum of water. Additionally, the forces affecting a propeller are described using single blade elements. Although detailed hydrodynamics of vessels are not considered in the simulations, the power used by propulsion is to be simulated. Also in this chapter the marine diesel engine and and three-phase electric machines are briefly introduced and how they are used in a vessel.

2.1 Marine propulsion

At first steam powered vessels used rotating paddle wheels installed at the stern or on the sides to achieve propulsion. The paddle wheel was eventually mostly replaced by the screw propeller. A famous early screw propeller design was based on the Archimedes' screw; a simple machine used to pump water, and the first steamship with the design was consequently named Archimedes. Following Archimedes, screw propulsion gained popularity and with the development of the theory of operation, the screw propeller developed into what are found in almost every vessel today; a rotating hub with a number of helical blades attached to it. [5]

Momentum theory describes how a propeller produces thrust. In the theory the propeller is replaced by an infinitesimally thin actuator disc that causes an abrupt change in pressure. The pressure difference causes the fluid to accelerate which appears as thrust. Flowing fluid forms a vector field where at every point the fluid has a velocity i.e. speed and direction. Lines drawn in the field that are always tangential to the velocity will form streamlines which never cross each other and whose density is proportional to the magnitude of the velocity. A closed curve of streamlines forms a streamtube [6]. Figure 2.1a shows a streamtube where there is an actuator disc in the middle at position B and 2.1b shows the flow velocity and pressure along the tube. The disc is moving to the left with the vessel it propels at some speed v_A. Attaching the frame of reference to the disc the fluid on the left at position A is flowing towards the disc at equal speed. Position A is far enough out that the disc does not affect the static pressure p_0. Closing in on the disc the flow speeds up and consequently the pressure starts decreasing. As the speed increases the tube diameter becomes smaller because conservation of mass requires that the mass flow rate

$$\dot{m} = \rho A_\mathrm{A} v_\mathrm{A} = \rho A_\mathrm{B} v_\mathrm{B} = \rho A_\mathrm{C} v_\mathrm{C} \,, \tag{2.1}$$

where ρ is the density of the fluid and A are areas, through the tube is equal at every position along the tube. At position B flow speed is v_B and there is a pressure discontinuity where the pressure increases. Moving away from B to position C the flow speed continues increasing and pressure is dropping. At position C the speed has reached v_C and the distance to the disc is large enough that the pressure is p_0.

The propeller works to change the kinetic energy of the fluid with power

$$P = \frac{\dot{m}}{2}\left(v_\mathrm{C}^2 - v_\mathrm{A}^2\right) \tag{2.2}$$

and produces thrust

$$F_\mathrm{T} = \dot{m}\left(v_\mathrm{C} - v_\mathrm{A}\right) \tag{2.3}$$

that is the rate at which the fluid momentum is changing. Combining equations (2.2) and (2.3) and writing power using v_B yields

$$P = \frac{1}{2} F_\mathrm{T}\left(v_\mathrm{C} + v_\mathrm{A}\right) = F_\mathrm{T} v_\mathrm{B} \,, \tag{2.4}$$

From which it can be seen that

$$v_\mathrm{B} = \frac{1}{2}\left(v_\mathrm{C} + v_\mathrm{A}\right) \tag{2.5}$$

i.e. v_B is the mean of v_A and v_C which means half of the acceleration happens before the propeller and the rest after the propeller. Some of the power delivered to the propeller is lost in the fluid as its kinetic energy increases. Ideal efficiency is the ratio of work done per unit time on propelling the vessel forwards $F_\mathrm{T} v_\mathrm{A}$ to total power delivered to the propeller $F_\mathrm{T} v_\mathrm{B}$, when other energy losses such as rotation are neglected. Writing v_B and v_C using v_A as

$$v_\mathrm{B} = v_\mathrm{A} + a v_\mathrm{A} \tag{2.6}$$

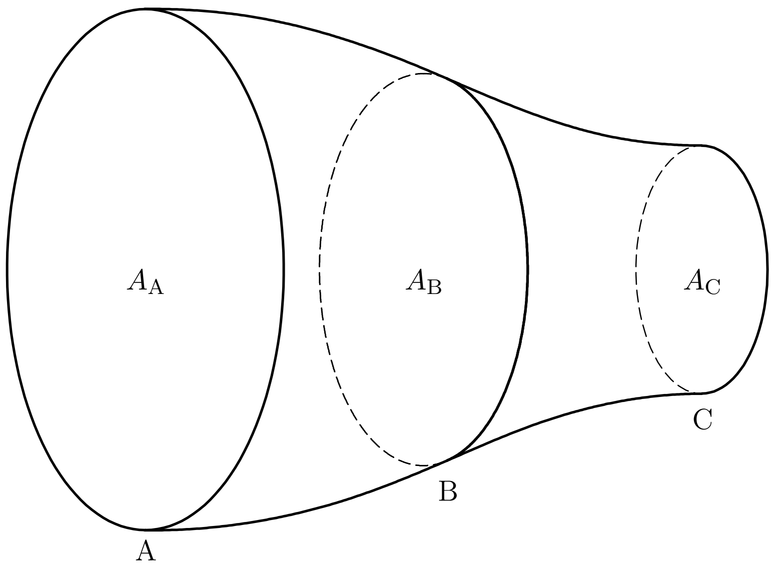
$$v_\mathrm{C} = v_\mathrm{A} + 2 a v_\mathrm{A} \,, \tag{2.7}$$

where a is the axial inflow factor the ideal efficiency of the propeller is

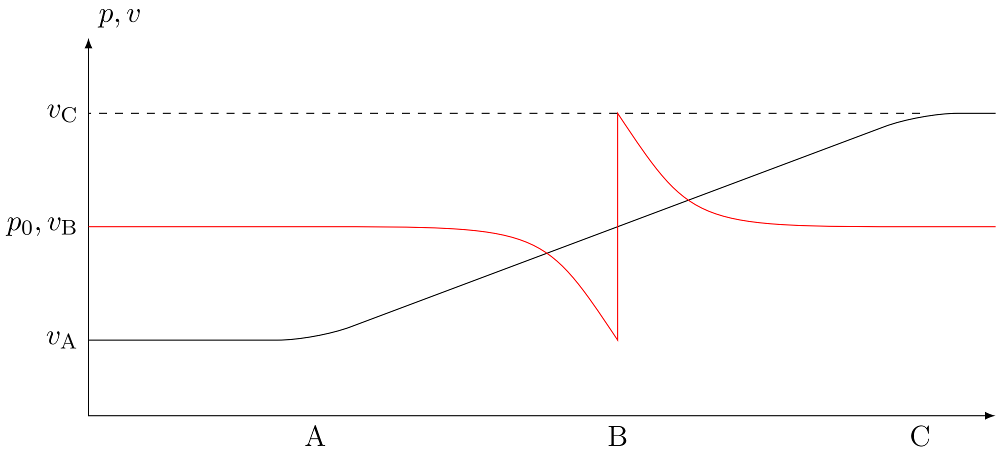
$$\eta_\mathrm{i} = \frac{F_\mathrm{T} v_\mathrm{A}}{F_\mathrm{T} v_\mathrm{B}} = \frac{1}{1+a} \,. \tag{2.8}$$

It follows that the efficiency improves as the change in fluid speed decreases. It is therefore more efficient for a vessel to have a large diameter propeller rotating at low speed versus a small diameter propeller and high speed. Using the equations it can also be seen that propulsion power is proportional to vessel speed raised to the third power. [5, 7]

The momentum theory does not consider the geometry of the propeller. Blade element theory looks at differential elements of the propeller blades and the forces subject to them. A blade element is an airfoil which will produce a force when moving in a fluid. Figure 2.2



(a) Streamtube with an actuator disc at position B causing acceleration of flow velocity. The tube diameter decreases as the flow accelerates to satisfy conservation of mass.



(b) Pressure (red) and velocity in the tube. The actuator disc causes an increase in velocity. With increasing velocity, the pressure drops from the static value. At the location of the disc there is a pressure discontinuity after which the pressure decreases to the static value.

Figure 2.1: Streamtube and a graph of the pressure and velocity along the tube.

shows an example of a blade element, the velocity of the fluid relative to the element, and the resulting forces. The fluid velocity w is the sum of the advancement velocity v_A and the tangential velocity v_r caused by rotation of the propeller. The angle between airfoil chord line and fluid velocity is the angle-of-attack (AoA) which affects the resulting lift force L which contributes to thrust force F_T and torque inducing force T. Increasing AoA increases the lift force and thus thrust until the airfoil stalls. The pitch of the blade element increases towards the center to compensate for the decreasing tangential velocity with AoA, so that the thrust produced by an element stays approximately constant along the blade.

The thrust produced by a propeller can be varied by adjusting its rotational speed or the pitch angle of the blades. Propellers installed in vessels are either fixed-pitch (FPP) or controllable-pitch (CPP). The use of a CPP enables precise control of thrust and reversing when the propeller is directly driven by a machine that offers little speed variation. The propeller can also be optimized for different speeds and sailing vessels benefit from

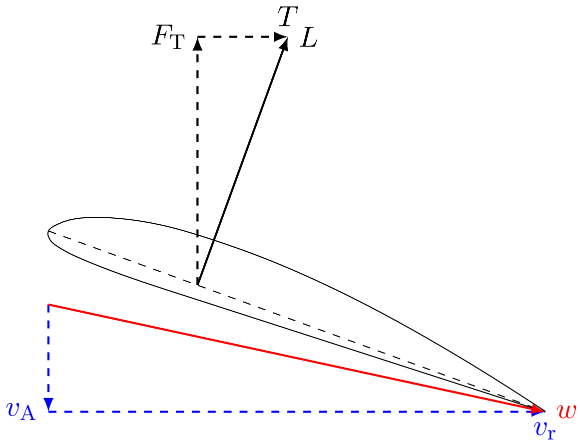


Figure 2.2: A propeller blade element. The fluid velocity relative to the blade w is the sum of the advancement speed through the fluid v_{A} and rotation v_{r}. The blade element produces lift force L which causes thrust and torque forces F_{T}, T.

feathering of the propeller when sails are used. The disadvantages of CPPs are higher cost and maintenance requirements arising from the added mechanical complexity.

2.2 Machinery

Machines on-board vessels perform energy conversions to provide propulsion and electric power. Prime movers convert energy of a primary source, usually fuel oil, to mechanical energy. Electric machines convert between mechanical and electrical energies, the process being bidirectional i.e. the machine can function as a motor or generator. Propulsion can either be run by a prime mover or by an electric machine.

First prime movers were reciprocating steam engines. A reciprocating steam engine uses the expansion of steam to drive a piston in a cylinder. The linear motion of the piston is converted to rotating motion of the crankshaft. Coal was first used as fuel to produce steam in boilers, being later replaced by fuel oil which has roughly double the energy density and as a liquid is easier to handle. To improve efficiency the triple-expansion engine where the steam was expanded in three stages was common. After expansion the steam is condensed back to water and returned to the boilers. Reciprocating steam engines were replaced by steam turbines and internal combustion engines. Steam turbines have superior efficiency and smaller size compared to a reciprocating steam engine of similar power. Steam turbines are nowadays used in nuclear powered vessels and liquefied natural gas (LNG) carriers. LNG boils off in transportation and carriers burn the boil-off gas in boilers.

The internal combustion engine and more specifically the diesel engine quickly gained popularity as a prime mover after its first adaptation in vessels. The efficiency of the diesel engine has made it the most common prime mover, available in powers ranging from tens of kilowatts to tens of megawatts. Dual-fuel diesel engines that can use both liquid and gaseous fuel are replacing steam turbines in new LNG carriers [3]. Seeking further improvements in efficiency and fuel cost savings, and tightening emissions regulations, electrification of marine traffic has gained interest. With the automotive industry moving towards battery-powered full-electric vehicles, batteries have started to find their way to new types of hybrid and full-electric vessels.

2.2.1 Diesel engine

The diesel engine is a reciprocating internal combustion engine that ignites fuel by first compressing air in the combustion chamber after which fuel is introduced into the chamber. The high temperature due to compression ignites the fuel. This is in contrast to a gasoline engine which ignites the air-fuel mixture with an electric spark.

The theoretical thermodynamic cycle describing a modern diesel engine is the dual cycle where heat is added in two parts; constant pressure (isobaric) and constant volume (isochoric). Figure 2.3 shows the pV-diagram of the cycle. Starting from point 1 where the piston is at its down position, the piston moves up and performs adiabatic compression of the air inside the cylinder. In adiabatic compression no transfer of heat occurs. At point 2 fuel is added which starts the addition of heat as combustion. Between points 2 and 3 pressure quickly increases while the piston is at its top position which constitutes the isochoric part. Between points 3 and 4 the combustion is still in progress while the piston has just started to move down resulting in isobaric addition of heat. From point 4 onward adiabatic expansion occurs while the piston moves down until point 5, after which hot exhaust gas is vented and heat is removed from the system in an isochoric process, completing the thermal cycle. The piston does work during its power stroke i.e. between points 3 and 5 in the diagram. During the compression, however, work is done on the system to compress the air. The energy for compression comes from the rotational energy of the crankshaft. Large inertia of the flywheel attached to the crankshaft keeps the rotational speed change during compression small. The useful work done by the engine during one cycle is given by the integral.

$$W = \oint p \, \mathrm{d}V \, , \tag{2.9}$$

i.e. the area enclosed in the diagram. Depending on the engine the cycle can be completed in two or four piston strokes. [8]

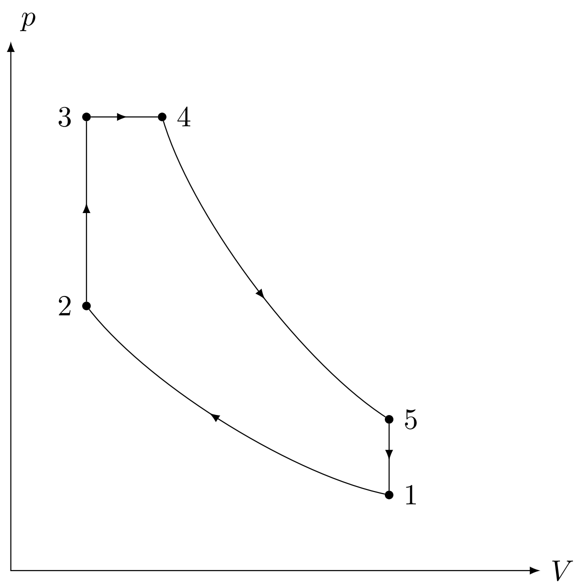


Figure 2.3: Pressure-volume-diagram of an ideal thermodynamic cycle describing the operation of a diesel engine.

Marine diesel engines are categorized by their speed to low-, medium-, and high-speed. The largest and most powerful engines are two-stroke low-speed engines that run below 400 rpm, usually at about 100 rpm which is suitable for driving a propeller directly. The engines work with cheap, lower quality fuel oils and have lower maintenance needs. Higher speed engines, with medium-speed range extending to 1200 rpm are smaller and lighter four-stroke engines. They are used to run electric generators and turn propellers through a gearbox. [9, 10]

2.2.2 Electric machine and three-phase systems

Electric machines are used on-board to generate electricity for service and hotel loads. Some vessels also use electric motors to run the propulsion. The basis for electrical generation and distribution on-board vessels is the same as on shore; electromagnetic induction and three-phase systems.

Generally, with the exception of solar power, electricity is generated by rotating electric machines. The rotating part of such machine is the rotor while other parts belong to the stator. To generate electricity the rotor is magnetized and rotated by a mechanical power source. The magnetic field of the rotor causes a magnetic flux that goes through loops of conductors, the windings, in the stator. Due to rotor rotation the magnetic flux through a winding changes sinusoidally with the frequency being proportional to the rotational speed. A changing magnetic flux induces an electromotive force (EMF) e in the winding according to

$$e = -N \frac{\mathrm{d}\Phi}{\mathrm{d}t} \,. \tag{2.10}$$

where N is the number of turns in the winding and Φ is magnetic flux. When an electric load is placed between the winding terminals the electromotive force will cause a current to flow. According to Lenz's law the magnetic field caused by the induced current opposes the field that induced the current. The mechanical effect is that the rotor experiences torque that opposes its rotation. As a result, power is converted from mechanical to electrical. By equipping the stator with three windings that are separated spatially by 120 degrees, the induced voltages are also separated by a 120-degree phase shift. The most notable advantages of having three phases compared to a single-phase generator is the ability to transfer triple the amount of power with one additional conductor and the power being constant instead of sinusoidal [11].

By connecting the stator windings of the machine to a three-phase power source a rotating magnetic field is created which is the basis for AC motors, which can either be synchronous or asynchronous. Synchronous machines have a rotor that is magnetized electrically by field windings or with permanent magnets. The rotor's rotation is synchronous due to the magnetic fields of the rotor and stator interacting. Asynchronous machines have rotors with conducting bars that are short-circuited at the ends. The rotating field cuts through these bars and as a consequence current is induced. The current carrying bars will experience Lorentz force due to them being in a magnetic field. The force imposes a torque on the rotor which causes rotation. The speed of an asynchronous motor is slightly less than the synchronous speed because at synchronous speed no currents are induced in the rotor bars and thus no torque is produced.

2.3 Propulsion system configurations

Propulsion systems are classified by how the power to the propulsion is transmitted from the prime mover, mechanically or electrically. The propulsion system also affects the production of electrical power for service loads.

2.3.1 Mechanical

The conventional way to turn the propeller is to connect it mechanically to a prime mover through shafts and gearing depending on the engine speed range. For large container ships and tankers, a single large low-speed diesel direct-driving a propeller is the preferred propulsion system. Low-speed engines having the best thermal efficiency [12] and the absence of losses due to gearing combined with the load profile; most time is spent crossing oceans with the engine loaded optimally, makes the mechanical propulsion the best system economically for these kinds of vessels. Medium-speed diesels require gearing to reduce the shaft speed to a speed suitable for the propeller. Gearing makes it also possible to drive a propeller using two or more engines. Electricity for service loads is generated using separate auxiliary generator sets.

2.3.2 Electric

In a diesel-electric propulsion system electric motors run the propulsion while diesel gensets produce electricity for them. Diesel-electric propulsion can use dedicated diesel gensets to produce electricity for propulsion and separate auxiliary gensets for service loads, or in the case of integrated electric propulsion there is a common electrical bus for both propulsion and service [13]. Electric propulsion systems have been used for decades; the first Finnish diesel-electric icebreaker Sisu was built in 1939 and since then all Finnish icebreakers have used electric propulsion. DC-motors were first used due to their speed controllability, the development of power electronics and variable speed AC-drives enabled the use of AC-machines which require less maintenance.

Electric propulsion systems bring many advantages. The placement of generators is flexible; instead of rigid shafts the power is transmitted using cables that can be run with ease from different positions. This allows, for example, the placement of generators higher up on a ship, where the engine air supply and exhaust air ducts can be shorter and less space-consuming. Other equipment can be placed in positions normally taken by shafts, further reducing the volume needed for machinery. The saved space can be used for revenue-generating spaces such as cargo or passenger space. Electric motors generate full torque from zero to nominal speed, their speed is controlled and can be varied quickly which increases maneuverability. The need for maintenance is reduced as there is no gearbox needed and there are no long mechanical shaft lines whose alignment need to be maintained. Despite the added losses due to energy conversion from mechanical to electrical and back to mechanical, electric propulsion brings fuel savings when operating at reduced speeds where the propulsion power demand is lower. Having several generators provide propulsion power, some can be turned off when only a part of total available propulsion power is in use. The remaining generators will run close to the loading point where their fuel consumption is optimal. [3, 13]

The benefits of electric propulsion have been found in the cruise ship industry; the famous ocean liner *Queen Elizabeth II* was converted from mechanical steam turbine propulsion system to diesel-electric. The large hotel loads in a cruise ship and ABB introducing the electric *Azipod* podded propulsion system have made electric propulsion dominant in cruise ships. Podded propulsion system has the electric motor and propeller placed in a pod under the hull. The pod is an azimuth thruster meaning it can rotate 360 degrees and thus increases maneuverability and eliminates the need for a rudder, especially beneficial for cruise ships as they make many port visits on a cruise. Other types of azimuth thrusters are the Z-drive and L-drive where the motor is located inside the hull and power is transmitted mechanically to the thruster. Azimuth thrusters also improve efficiency as it places the propeller in cleaner flow. Often the propeller is pointed towards the bow to pull the vessel forwards. Other vessels benefiting from electric propulsion and azimuth thrusters include ferries, icebreakers, offshore supply vessels, research vessels and cable layers. Some vessels have a dynamic positioning system where thrusters are computer controlled to keep the vessel's position and heading relative to the seabed or another vessel, a task benefiting from the fast dynamics of electric motors. [13]

The traditional vessel with electric propulsion has synchronous generators making the on-board 50/60 Hz grid at voltages such as 660 V, 6600 V or 11 kV, depending on the power plant size. Transformers are used to lower the voltage to levels suitable for service loads. Motor drives alter the frequency and voltage to be able to drive propulsion motors at variable speeds. Depending on the drive type the frequency conversion can happen directly to a lower frequency or via an intermediate conversion to DC voltage or current.

A newer, and the one used in the Editron Marine System, electric system architecture on-board a vessel is DC based. The AC power generated by gensets is rectified to DC. Converters are used to convert the DC back to AC to supply the service loads and propulsion motors. The benefits of a DC based system are the ability to run generators at variable speeds which improves fuel efficiency, size and weight savings, simpler generator parallelization, generators having a power factor of one, and easier integration of energy storage systems such as batteries [14].

In recent years battery power has started to find its way into new types of hybrid and even full-electric vessels. Battery-hybrid technology offers abilities such as emissions-free operation in harbors, engine load optimizations, power redundancy without having extra engines running, and engine load peak shaving [15]. Battery powered full-electric vessels have been shown to be viable in short range ferry traffic. One such vessel is the e-ferry *Ellen*. Her gross tonnage is 996 and she has a battery capacity of 4.3 MWh. The ferry operates a round trip of 22 NM in Denmark with automated charging connection at one end of the route. The charger can deliver a maximum charging power of 4 MW. The electrification is estimated to save annually 2520 t of CO_2 emissions and the higher initial investments are estimated to be paid off after 5 to 8 years of operation [16].

3 SYSTEM ARCHITECTURE

The EDITRON marine system is a complete solution for a vessel's electrical power system. It includes generators, propulsion motors, AC grid for hotel and service loads, energy storage systems, and shore connection. The system is suitable for vessels with propulsion shaft power up to 1.5 MW and highly customizable. In addition to the diesel-electric serial hybrid and full-electric systems, parallel hybrid is also an option where an electric machine is placed on the same mechanical shaft as the prime mover and propeller.

The basis of the marine system are permanent magnet electric machines and power electronic converters. The converters transfer power bidirectionally between different AC and DC voltage levels. All the converters connect to a DC-link. Figure 3.1 shows a simplified single-line diagram of an example system. The horizontal line is the DC-link to which a number of subsystems connect as branches. In a single-line diagram, electrical connections are represented as a single line regardless of the number of conductors used. At the top there are four generating sets consisting of a prime mover, generally a medium speed diesel engine, an electric machine and a converter, which converts the three-phase AC voltage of the generator to a DC voltage. The converters work to keep the DC voltage stable by controlling the torque of the machine. When the load increases the voltage will start to drop to which the converter responds by increasing torque and thus the power that is drawn from the electric machine and prime mover. Because of the conversion to DC, the electric machine speed and thus the frequency of the generated AC does not have to stay constant. The prime mover speed can thus be varied in such a way that the speed minimizes the fuel consumption for a given load level.

The battery connects to the DC link via a DC/DC converter, which regulates the power flow to and from the battery. Shore connection is often used when the vessel is docked in the harbor for charging of batteries and powering the hotel grid without having to run generators. At the bottom of the diagram are two propulsion motors driving propellers. The motor converters control the speed of the motors according to a speed reference originating from throttle levers installed at the wheelhouse. The hotel grid converter maintains the vessel's hotel AC grid.

Omitted from the diagram are fuses, isolators, bus ties, transformers, and filters. Fuses protect against overcurrent situations and isolators are used to galvanically isolate a branch if, for example, maintenance work has to be performed and the rest of the system is to be operational. Propulsion motors are to be also isolatable because they will function as generators when they are not used, due to propeller windmilling when the vessel is moving. Often the DC-link is divided into two parts with a bus tie connecting the parts, enabling separating the system into two individual pieces. Transformers are used between the converter and hotel or shore grid and filters smoothen the converter AC voltage output which is made of pulses.

An integral part of the marine system is the Editron Control System (ECS). ECS is implemented in a PLC and communicates with and controls the different field devices. When a vessel has several gensets and propulsion motors, the system is divided into two parts and equipped with two ECS to provide redundancy. In case of a ECS failure the other side can continue operation. The electric machines, power electronic converters, and ECS are discussed in more detail in the following sections.

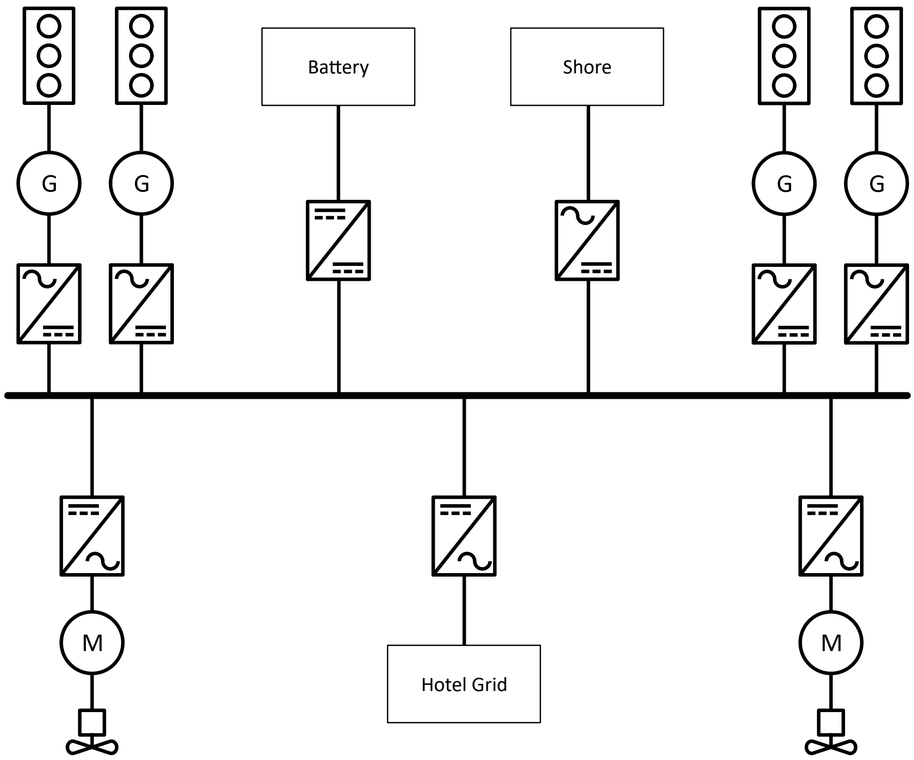


Figure 3.1: Simplified single-line diagram of an example EDITRON marine system configuration.

3.1 Permanent magnet electric machines

The electric machines in the marine system most often are Editron permanent magnet synchronous machines (PMSM), and more specifically ones that use synchronous reluctance assisted permanent magnet technology. Synchronous machines are AC machines in which the rotor rotation is synchronized to the rotating magnetic field inside the stator. The rotating magnetic field is produced by the stator windings when a three-phase alternating current is flowing in them. The stator three-phase system is balanced i.e. the impedance of the windings are approximately equal and the three phase currents add up to zero as there is no neutral conductor. The rotor has one or more magnetic pole pairs which get locked with the rotating magnetic field resulting in synchronism. The rotor magnetization can be achieved by using field windings in the rotor, or by permanent magnets in PMSMs. The use of permanent magnets improves efficiency as there are no field windings in the rotor subject to Joule losses, however, permanent magnets are conductive and eddy currents are induced in them resulting in some losses. The lack of field windings also makes the construction simpler and more compact as slip rings and brushes, or a more complex brushless design, to carry the magnetization current into the rotor are not needed.

3.1.1 PMSM structure and operation

Fig. 3.2 shows a cross section of a simple two pole PMSM consisting of a stator and its windings and a rotor centered in the stator. The stator windings are coiled around the stator teeth such that one phase current magnetizes two opposing teeth. The three phases, colored green, yellow and purple form three magnetic axes a, b and c, respectively. When a current flows in the phase conductors a magnetic field is created that is aligned with its magnetic axis. A magnetomotive force pushes magnetic flux around the stator into the opposing tooth, across an air gap into the rotor, and finally across the second air gap to the stator, completing a magnetic circuit. The magnetic flux will take the path of least magnetic resistance called reluctance. The three dashed-line vectors on the magnetic axes represent the magnetic fields induced by the stator currents at one time instant, the circular conductors around the stator teeth are marked with the direction of current at that time instant, dot is outwards from the page and cross is inwards. The blue vector is the sum of the individual magnetic field vectors. As time passes the component vectors will oscillate sinusoidally, whereas the sum vector has a constant magnitude of 3/2 times the maximum amplitude of the component vectors and rotates at the electrical frequency creating a rotating magnetic field. The rotor has a pair of permanent magnets embedded in its surface. The magnets are colored red and blue corresponding to the north and south magnetic poles on the rotor surface. [17]

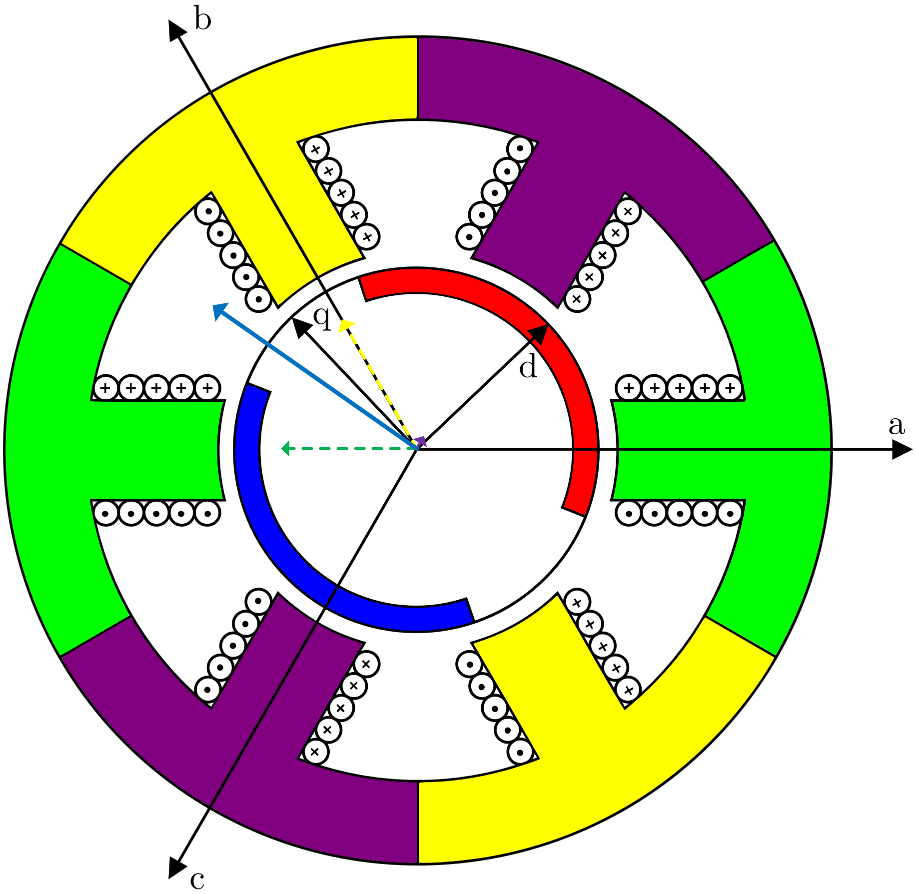


Figure 3.2: Cross section schematic of a two pole PMSM. Two frames of reference, the abc-frame fixed to the stator and the dq-frame fixed to the rotor are shown. The vectors represent magnetic fields induced by the phase currents and their sum.

The scalar electrical quantities voltage, current and magnetic flux in a three-phase system can be represented by space vectors and vector operations can be applied to them. The space vectors are formed by adding three vectors formed by the phases together just as the magnetic field vector is formed in Fig. 3.2. The resulting vector is scaled by 2/3 so its magnitude matches the peak value of the phase quantity [18]. Attached to the rotor is a reference frame represented by the direct (d) and quadrature (q) axes. The d-axis is aligned with the magnetic field that the rotor produces, and the q-axis is perpendicular to that. In the stator frame of reference, the space vectors rotate at the electrical frequency, however, in the rotating rotor frame of reference the vectors appear stationary. In addition, the stator inductances which vary by rotor angle, transform into constant d and q synchronous inductances $L_\mathrm{d}, L_\mathrm{q}$ which are the sum of the magnetizing inductance $L_\mathrm{md}, L_\mathrm{mq}$ and leakage inductance $L_{\mathrm{s}\sigma}$. Inductance is the ratio of magnetic flux to electric current. Space vector theory was developed to better understand the behavior of AC machines in transients. Space vectors and coordinate transformations simplify the machine representation greatly and is the basis for vector control.

For a rotating electrical machine to do work it needs to produce torque. Torque is proportional to the cross product of flux and current vectors, given by

$$\boldsymbol{T} = \frac{3}{2} p \boldsymbol{\psi}_\mathrm{s} \times \boldsymbol{i}_\mathrm{s}\,, \tag{3.1}$$

where $\boldsymbol{T}$, $\boldsymbol{\psi}_\mathrm{s}$, $\boldsymbol{i}_\mathrm{s}$ are torque, stator flux and current vectors, respectively. The pole pair number of the machine is p. Opposite magnetic poles of the rotating field and rotor attract and impose torque on the rotor when the poles are not aligned. The cross product implies that torque is produced by the components of the flux and current vectors that are perpendicular to each other. In addition to attractive force between opposite poles, torque can also be present due to reluctance. A ferromagnetic object placed in an external magnetic field will tend to orient in such a way that the reluctance in the magnetic circuit is minimized. A rotor that is subject to reluctance torque is a salient pole rotor. Saliency is a measure of the difference between the d- and q-axis inductances. As permanents magnets have a permeability similar to that of air, the rotor in Fig. 3.2 has a higher reluctance and thus a lower inductance in the d-axis. If the magnets were mounted on the surface the inductances would be approximately equal and thus the rotor would be a non-salient type and no reluctance torque would be produced. Saliency is determined by magnet placement and rotor design. [19]

Practical high-performance machines have several pole pairs meaning it takes p electrical cycles to complete one mechanical revolution in a machine with p pole pairs. The number of pole pairs is determined by the windings. The winding pattern is repeated until the desired number of pole pairs are present. The rotor is made to have an equal number of pole pairs. The stator depicted in Fig. 3.2 has concentrated windings. Such windings are commonly found in brushless DC (BLDC) machines. The principle of operation is the same in PMSM and BLDC. Concentrated windings produce flux that varies in a trapezoidal manner in the air gap resulting also in a trapezoidal back emf. Consequently, the torque is not smooth but cogged. In PMSM the windings are distributed between several stator slots with the number of conductors in the slot varying sinusoidally. This results in sinusoidal flux, back emf and torque. The stator can also be wound with two or more galvanically isolated three-phase systems which are driven by individual inverters to achieve higher power than what one inverter is rated for. [20]

3.1.2 Control of the machine

As the speed of an AC machine depends on the electrical frequency the speed of the machine is controlled by controlling the frequency of the electrical power. The simplest way to control the machine is using scalar control also known as V/f control. In scalar control the voltage to frequency ratio is kept constant. Synchronous machines run at the set frequency, but asynchronous machines have slip meaning there will be a steady state speed error if not compensated for with e.g. a feedback controller. Scalar control works well in applications where the dynamic behavior of the machine in transients is of no big concern. For precise control the torque produced by the machine is the significant quantity. Speed is proportional to the integral of torque and thus speed control is best done by controlling torque.

DC machines have been used in applications where precise control of torque is needed. In a DC machine flux and torque can be controlled individually by controlling field and armature winding currents. Field-oriented control (FOC) also known as vector control was developed on basis of the space vector theory to achieve control performance comparable to DC machines for AC machines. DC machines are inferior when it comes to maintenance; brushes wear out and need to be replaced regularly. In FOC the three phase currents are transformed to two orthogonal currents; the flux producing d-axis current i_d and the torque producing q-axis current i_q, which are controlled individually [18].

The steady state operation of a machine can be represented by a space vector diagram, drawn in Fig. 3.3 for an arbitrary PMSM. The diagram shows the stator voltage, current and flux space vectors and their components. The stator flux linkage ψ_s comprises of permanent magnet flux ψ_PM, and the flux $L_\mathrm{d} i_\mathrm{d}$, $L_\mathrm{q} i_\mathrm{q}$ produced by the stator currents. The angle δ_s between ψ_s and d-axis is known as load angle. The angle φ between voltage u_s and current determines the power factor. Voltage integrates to flux and thus the angle between voltage and stator flux, when stator resistance is neglected, is 90 degrees. Evaluating

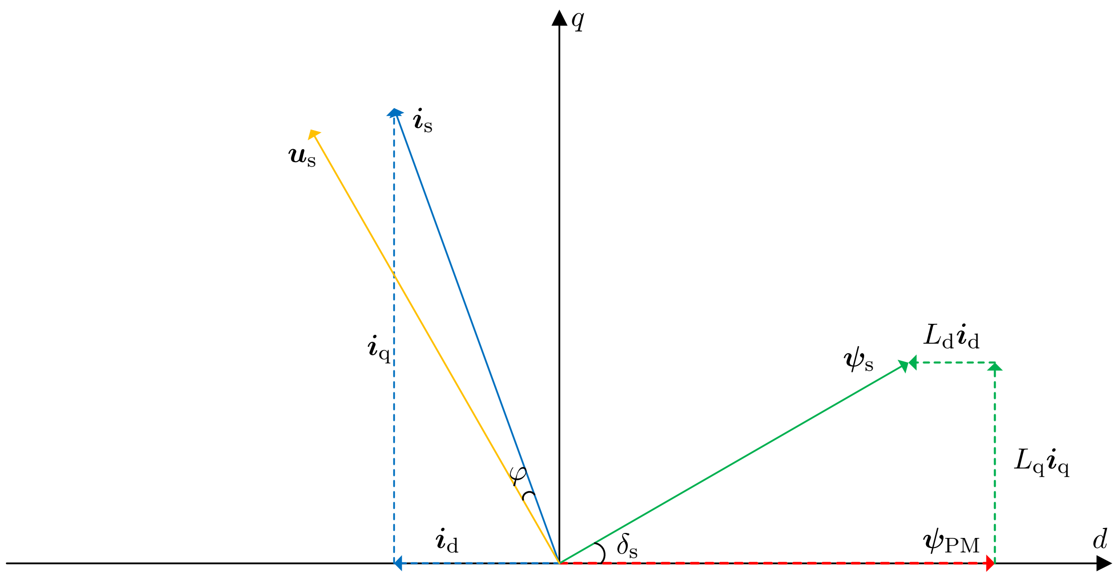


Figure 3.3: Space vector diagram of a PMSM.

equation (3.1) with the vectors in the space vector diagram reveals the expression

$$T_\mathrm{e} = \frac{3}{2} p \left[\psi_\mathrm{PM} i_\mathrm{q} - (L_\mathrm{q} - L_\mathrm{d}) i_\mathrm{d} i_\mathrm{q} \right] \tag{3.2}$$

for electromagnetic torque T_e of a PMSM in steady state. The first term shows that the torque is proportional to q-axis current and permanent magnet flux ψ_PM. The second term is the reluctance torque and it is proportional to the difference between inductances. Efficient control of torque requires that the ratio of torque to current is maximized. For a non-salient pole machine the torque equation (3.2) shows that i_d should be kept at zero. For a salient pole PMSM with $L_\mathrm{q} > L_\mathrm{d}$ maximum torque per amp is reached with a negative i_d meaning the machine should be driven in slight field weakening.

To be able to control the current in the dq-frame, transformations between the stator abc- and the rotor dq-frame are needed. The three phase currents are first presented using two orthogonal currents. This transformation is known as alpha-beta or Clarke transformation. The $\alpha\beta$ frame is fixed to the stator and the α-axis is aligned with the a-axis, β-axis is perpendicular to that. The Clarke transformation is given by

$$\begin{bmatrix} i_\alpha \\ i_\beta \\ i_0 \end{bmatrix} = \boldsymbol{T}_\mathrm{C} \begin{bmatrix} i_\mathrm{a} \\ i_\mathrm{b} \\ i_\mathrm{c} \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} i_\mathrm{a} \\ i_\mathrm{b} \\ i_\mathrm{c} \end{bmatrix}, \tag{3.3}$$

where i_α, i_β, i_0 are the current components in the $\alpha\beta$ frame, $\boldsymbol{T}_C$ is the Clarke transformation matrix, and $i_\mathrm{a}, i_\mathrm{b}, i_\mathrm{c}$ are the phase currents. The zero-sequence component i_0 is zero in a balanced three phase system. Changing the frame of reference from the $\alpha\beta$ frame to the rotating dq frame is achieved by rotating the vector clockwise by the rotor angle θ i.e. the angle between α- and d-axis. The rotation, known as dq0 transformation is given by

$$\begin{bmatrix} i_\mathrm{d} \\ i_\mathrm{q} \\ i_0 \end{bmatrix} = \boldsymbol{T}_\mathrm{dq0} \begin{bmatrix} i_\alpha \\ i_\beta \\ i_0 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \\ i_0 \end{bmatrix}, \tag{3.4}$$

where $\boldsymbol{T}_\mathrm{dq0}$ is the dq0 transformation matrix. Combining the Clarke and dq0 transformation matrices results in transformation from the abc frame to dq frame. This transformation is the Park transformation and is given by

$$\begin{bmatrix} i_\mathrm{d} \\ i_\mathrm{q} \\ i_0 \end{bmatrix} = \boldsymbol{T}_\mathrm{P} \begin{bmatrix} i_\mathrm{a} \\ i_\mathrm{b} \\ i_\mathrm{c} \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos\theta & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ -\sin\theta & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} i_\mathrm{a} \\ i_\mathrm{b} \\ i_\mathrm{c} \end{bmatrix}, \tag{3.5}$$

where $\boldsymbol{T}_\mathrm{P} = \boldsymbol{T}_\mathrm{dq0}\boldsymbol{T}_\mathrm{C}$ is the Park transformation matrix [21]. Figure 3.4 shows the waveforms of a three phase currents in the different reference frames. The transformations make the controlling of current an easy task; instead of having to track a sinusoidal reference, the control is done in the dq frame where the references are DC values. Consequently, a simple proportional-integral (PI) controller can be used.

Figure 3.5 shows the process block diagram of field oriented control of a PMSM. Some feedback of the state of machine is needed such as the phase currents and the rotor position θ and speed ω. It suffices to only measure two of the three phase currents, the third one is calculated by summing the first two. Rotor angle and speed are measured by an encoder or a resolver. They can also be estimated using a model of the machine in which case no rotation sensor is needed. When speed is controlled the actual speed is compared to a reference speed and an error signal is generated. The error signal goes to a PI controller which outputs a current reference signal $i_\mathrm{q,ref}$ which is proportional to the speed error and its integral, meaning that when there is a speed error more torque producing current is requested. The integrating part makes sure that the reference is reached, and torque is produced when there is no speed error. The current reference is compared with its measured value and the current controller then calculates the voltage that is to be applied to the stator windings. The flux producing d-axis current is controlled with its own control loop. An appropriate reference is calculated to achieve maximum torque per current or to apply field weakening. Space vector pulse width modulation (SVPWM) and a transistor bridge is used to apply the correct voltages to the motor windings.

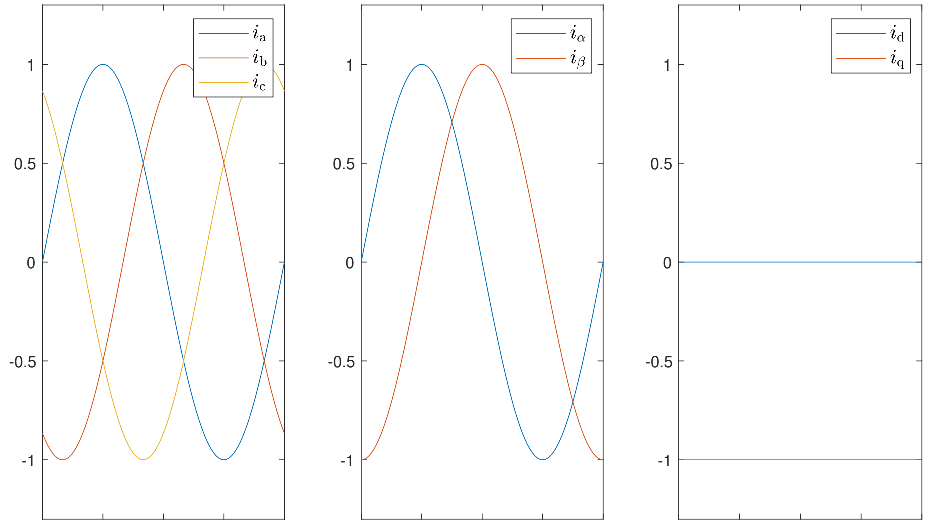


Figure 3.4: Transformation of three phase currents (left) into two orthogonal currents (middle), and into two DC currents (right).

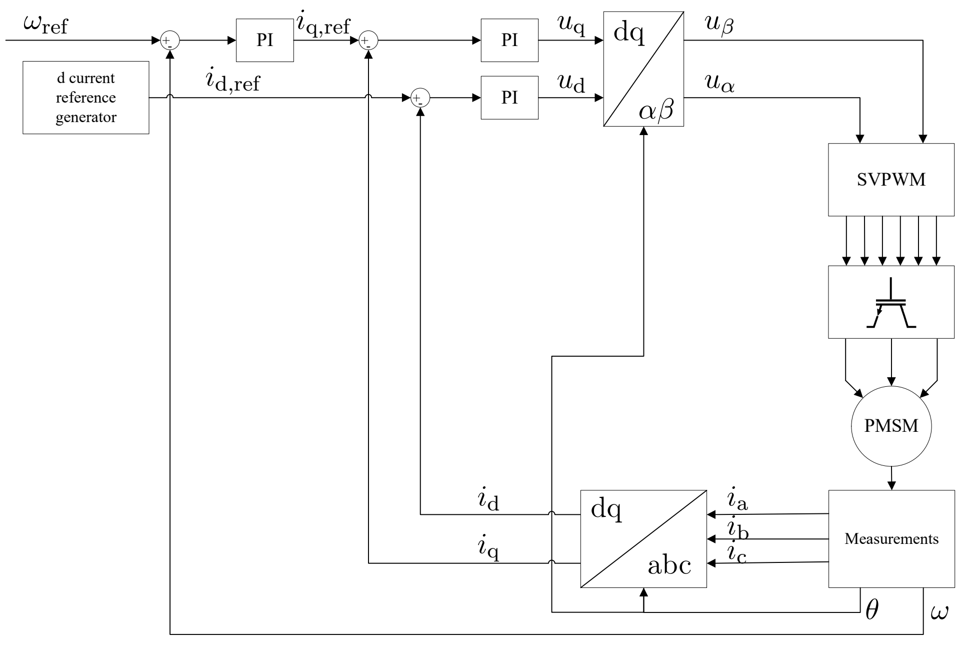


Figure 3.5: Diagram of the speed control of a PMSM. The process consists of PI controllers, coordinate transformations and space vector pulse width modulation.

3.2 Power electronic converters

Power electronic converter refers to a device which uses power electronics to convert electric power between frequency and voltage levels i.e. AC/DC, AC/AC and DC/DC. In addition to the general term converter, the devices are also called rectifiers when the power flows primarily from AC to DC, and inverters when power flows from DC to AC. The converter that is used in EDITRON systems is the Danfoss EC-C1200, pictured in Fig. 3.6. The converter features a rugged and compact design, can handle a power of up to 300 kW, and is liquid cooled [22]. The converter is software configured to perform in different roles. Same hardware is used to control motors and generators, provide power to the hotel grid, connect to shore grid, and convert between DC voltages to charge and discharge batteries or supercapacitors.

Figure 3.6: Danfoss EC-C1200 electric converter.

The topology of a three-phase inverter circuit is shown in Fig. 3.7. The DC-side of the circuit has a large capacitor and forms a DC-link with voltage U_{DC}. Next, there are six transistors Q1-Q6 acting as switches, and three phase leads: a,b,c. The capacitor stores energy in its electric field and keeps the DC voltage stable; it works as a constant voltage source and hence this kind of an inverter is a voltage source inverter. It is possible also to construct current source inverters where the DC-side has an inductor working as a current source. The transistors are insulated-gate bipolar transistors (IGBT). IGBT combines the high current capability of bipolar transistors and the simple driving with voltage of field-effect transistors. The IGBTs contain body diodes that provide a reverse current path and protect the transistors from inductive voltage spikes by clamping the voltage across the transistor to the diode forward-voltage. The IGBTs are turned on and off by gate drivers according to control signals generated by the converter processing system.

The transistor bridge in Fig. 3.7 has 6^2 combinations of switching states. To prevent short-circuiting the DC link, each pair Q1-Q2, Q3-Q4, Q5-Q6 are operated complementary, meaning when one transistor of the pair is conducting the second one is not, reducing the possible switching combinations to eight. A transistor pair thus works as a switch that connects the phase to either the positive or negative DC rail, or if referenced to the center of the DC voltage the phase is connected to $\pm U_{DC}/2$. A sinusoidal phase voltage

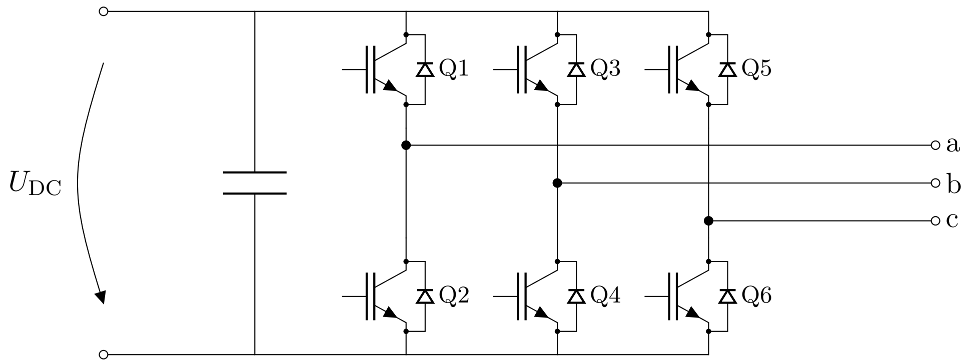


Figure 3.7: Circuit representing the topology of a three phase voltage source inverter. The circuit consists of six transistors working as switches and a DC link capacitor.

is achieved by modulating the DC voltage using pulse-width modulation (PWM) with a sinusoidally changing duty ratio. In sinusoidal PWM (SPWM) the sinusoidal reference voltage is compared with a triangular carrier signal and positive voltage is applied when the reference signal is greater than the carrier. The amplitude of the output is varied by varying the ratio between reference and carrier signal amplitudes which is quantified by the modulation index m. This is done individually for each phase using the same carrier signal. The frequency of the carrier determines the switching frequency, e.g. the EC-C1200 uses a switching frequency of $f_{\mathrm{s}} = 8\,\mathrm{kHz}$. Figure 3.8 shows an example of sinusoidal pulse width modulation, the voltage is made of pulses but the current is sinusoidal due to inductive load. In sinusoidal PWM the fundamental peak voltage of the phase leads is $U_{\mathrm{DC}}/2$ and thus the RMS line-to-line voltage in sinusoidal PWM for $m \leq 1$ is

$$U_{\mathrm{LL,spwm}} = m\frac{\sqrt{3}}{2\sqrt{2}}U_{\mathrm{DC}} \approx 0.612 \cdot mU_{\mathrm{DC}}\,. \tag{3.6}$$

Overmodulation occurs when $m > 1$ i.e. the reference signal has a greater amplitude than the carrier signal. Overmodulating increases the output voltage with the cost of harmonic distortion. Increasing the overmodulation eventually results in a square wave output. As the voltage waveform is made of pulses, a modulated sinusoidal voltage contains harmonics of the switching frequency in addition to the fundamental frequency. Electric machines tolerate such a voltage as the inductive windings filter the current to be sinusoidal. When the inverter is used to create a micro grid i.e. the vessel hotel grid, or to connect to the shore electrical grid, filters are installed to suppress the harmonic content.

Widely used method of modulation to create a three-phase voltage is space vector pulse width modulation (SVPWM). Instead of modulating each phase individually, as in SPWM, SVPWM controls the transistor bridge as one unit to synthesize voltage vectors. The switching combinations result in eight realizable voltage vectors, six active and two zero vectors. Figure 3.9 shows the available voltage vectors; $\boldsymbol{u}_1$ to $\boldsymbol{u}_6$ are active vectors, $\boldsymbol{u}_0$ and $\boldsymbol{u}_7$ are zero vectors. The three digits mark the state of the transistors Q1, Q3, and Q5, respectively, where one is conducting. For example $\boldsymbol{u}_2$ is synthesized by turning Q1, Q3 on and Q5 off (Q2, Q4 off and Q6 on) which results in phase voltages of $\left\{\frac{1}{3}, \frac{1}{3}, -\frac{2}{3}\right\} U_{\mathrm{DC}}$ for phases a, b and c, respectively, when the phases are connected to a balanced load. The resulting voltage vector is thus

$$\boldsymbol{u}_2 = \frac{2}{3}U_{\mathrm{DC}}\left(\frac{1}{3}\boldsymbol{a} + \frac{1}{3}\boldsymbol{b} - \frac{-2}{3}\boldsymbol{c}\right) = -\frac{2}{3}U_{\mathrm{DC}}\boldsymbol{c}\,, \tag{3.7}$$

where $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ are unit vectors pointing in the positive direction of the magnetic axes a, b, c.

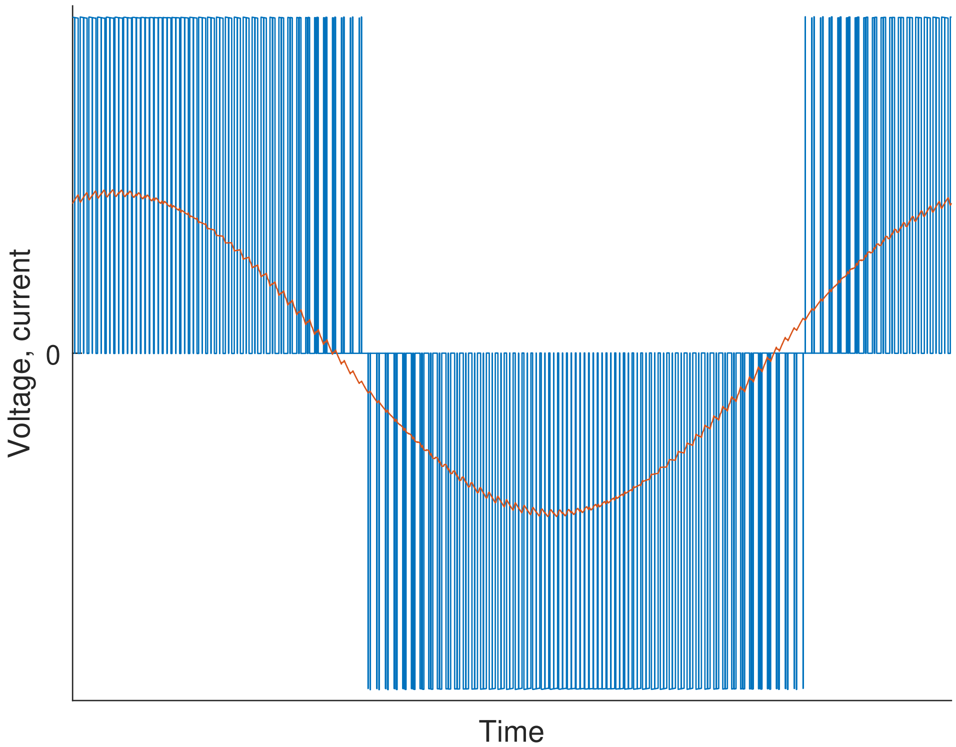


Figure 3.8: Waveforms of a pulse width modulated voltage (blue) and the resulting current (orange) of an inductive load. The voltage is made of small pulses, but the current is sinusoidal due to inductance limiting how fast the current can change.

The length of the active vectors is then

$$|\boldsymbol{u}_{1\ldots6}| = \frac{2}{3} U_{\mathrm{DC}} \,. \tag{3.8}$$

Other voltage vectors are formed by pulse width modulating two adjacent active vectors and the zero vectors. The angle of the resulting voltage vector is determined by the on time of one active vector relative to the other. Zero vectors are used to reduce the amplitude [23]. During the switching period $T_{\mathrm{s}} = 1/f_{\mathrm{s}}$ the vectors are applied in an optimal order to minimize switching losses and harmonics; no two switches are switched simultaneously. The switching period is divided into two equal parts where during the first part the order of vectors is: first zero vector, first active vector, second active vector, second zero vector. During the second part the order is reversed creating a symmetric pattern. In the figure a reference vector $\boldsymbol{u}_{\mathrm{ref}}$ is marked which is synthesized by the sequence $\boldsymbol{u}_0, \boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_7, \boldsymbol{u}_7, \boldsymbol{u}_2, \boldsymbol{u}_1, \boldsymbol{u}_0$. Marked with dashed blue lines are the adjacent vector components. The time relative to the switching period an active vector is applied, is the ratio of the length of the component vector to the active vector. The remaining time is allocated to zero vectors.

To produce a constant amplitude sinusoidal voltage, the length of the rotating reference vector should also be constant which sets a limit on the length as shown by the circle fit inside the hexagon in the figure. The maximum voltage vector amplitude attainable without overmodulation is found to be

$$|\boldsymbol{u}_{\mathrm{ref,max}}| = \cos 30^{\circ} \, |\boldsymbol{u}_{1\ldots6}| = \frac{\sqrt{3}}{3} U_{\mathrm{DC}} \tag{3.9}$$

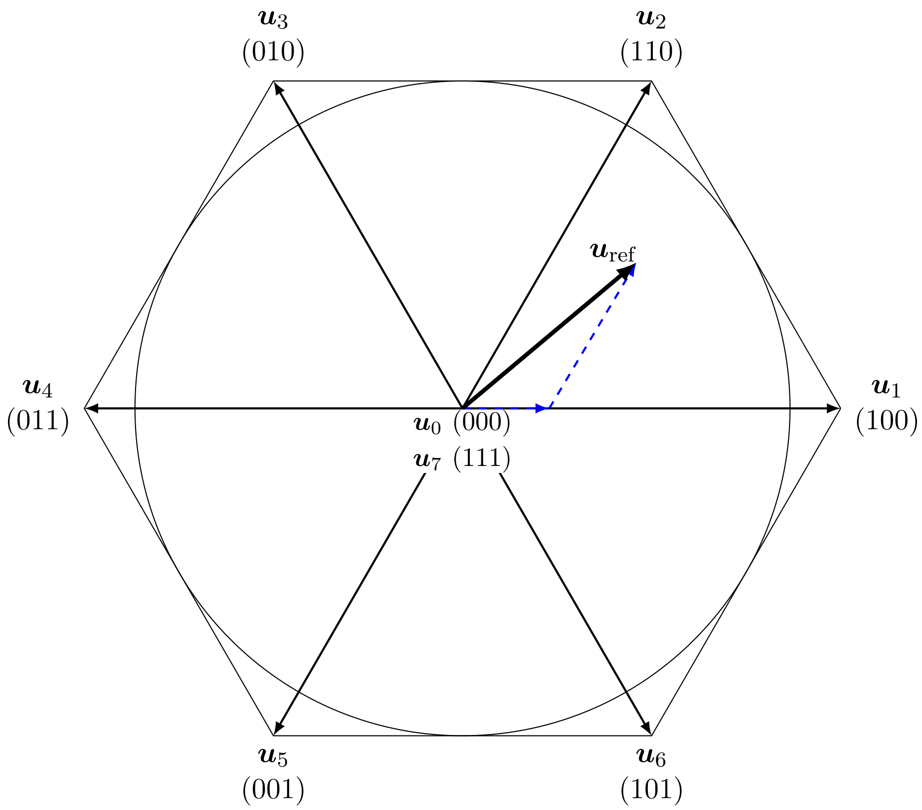


Figure 3.9: Voltage vectors used in SVPWM. A reference vector is constructed using adjacent vectors and the zero vectors.

and the maximum line-to-line voltage is then

$$U_{\text{LL,svpwm}} = |\boldsymbol{u}_{\text{ref,max}}| \frac{\sqrt{3}}{\sqrt{2}} = \frac{U_{\text{DC}}}{\sqrt{2}} \approx 0.707 \cdot U_{\text{DC}} \,. \tag{3.10}$$

SVPWM thus increases the utilization of the available DC voltage by approximately 15 % compared to SPWM.

As the circuit in Fig. 3.7 allows power flow to be bidirectional the same inverter can be also used as a rectifier i.e. AC power is converted to DC. Rectifying occurs when an electric machine controlled by the converter is working as a generator i.e. the signs of its torque and speed are opposite, and when the converter is used to draw power from an existing AC network. When the converter is configured to interface with an AC grid and deliver power to the DC link the converter is called an active front-end (AFE) or line converter. The use of an AFE as opposed to a passive diode bridge brings advantages such as better power quality due to active control, and the ability to regenerate power back to the grid.

In addition to motor control, microgrid, and AFE operation, when accompanied with an external inductance unit the converter can also function as a DC/DC converter to connect energy storage such as a battery or a supercapacitor to the DC link. When the energy storage is charged i.e. current is flowing from the higher voltage DC-link to the lower voltage energy storage the converter works as a buck converter. To discharge, the converter works as a boost converter which allows the current to flow from the energy storage to the DC-link.

3.3 Editron Control System

To operate all the devices as a system a control system is needed. The Editron Control System performs tasks such as starting, stopping and speed control of engines, commanding of converters, system monitoring, and interfacing with the Integrated Automation System (IAS) or Human-Machine Interface (HMI). To interface with the field the ECS uses hardwired inputs and outputs (HW IO) and fieldbuses. HW IO are either analog or digital and every signal uses at least one dedicated conductor. Examples of common digital inputs are fuse statuses, isolator statuses, contactor statuses, engine running status, physical buttons. Digital outputs are used to e.g. command engine start and stop. Analog inputs are either current or voltage signals that are used to e.g. indicate throttle lever position and temperatures. Analog outputs are useful for setting engine speed reference and driving physical gauges. Fieldbuses used by the ECS are CANopen and Modbus TCP. Fieldbuses enable devices to communicate in real-time by using a shared medium, such as a twisted-pair cable, which drastically reduces the required cabling. CANopen is used to command the converters and read their status info, whereas Modbus is used to receive commands from IAS or HMI and to send system information such as alarms to them.

The ECS is implemented on a *Beckhoff CX9020* PLC which is shown in Fig. 3.10. The HW IO connect to removable cards installed on the right side of the device. A card that is used to measure temperatures using resistance temperature detectors can be seen installed along with a terminator card. On the left there is a 9-pin connector for the CANopen bus and two ethernet ports that are used for Modbus and for connecting to the development computer.

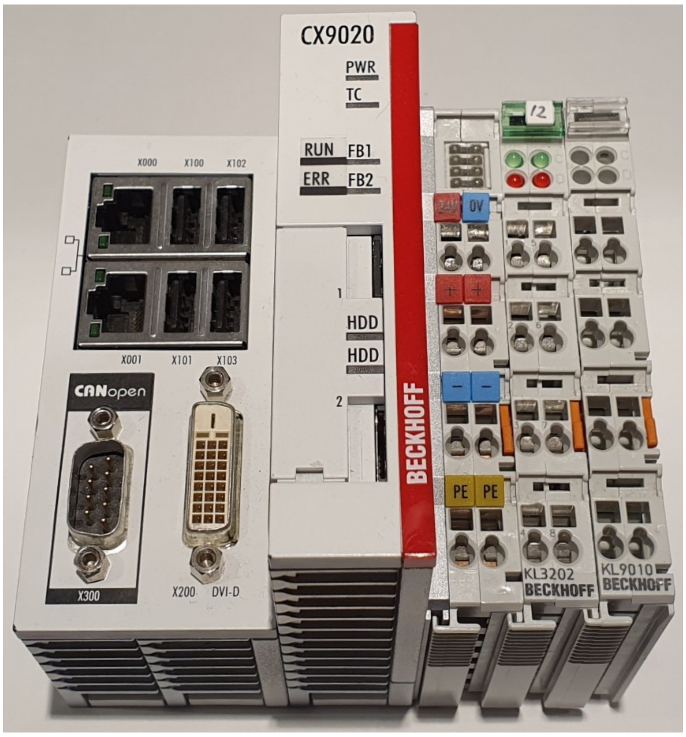


Figure 3.10: Beckhoff CX9020 PLC.

3.3.1 ECS software structure

At the heart of the ECS is the software running in the PLC. The software is modular and configurable so that modules can be reused between different vessels. Due to the nature of every vessel project being unique, new and modified software is needed between vessels, which brings the need for easy testing at the office. Fig. 3.11 shows an example block diagram of the software architecture when the hardware layout is according to the single-line diagram in Fig. 3.1 and the system is divided to two PLCs. The software is divided into managers and hardware modules. The managers interface with other managers and hardware modules, which interface with the devices in the field. At the top is the HMI interface, which is used by a HMI or IAS to give commands and access information about the system. The PLCs communicate with each other as the system should in normal conditions act as one, thus one PLC would act as a master and relay commands to the second PLC acting as a slave.

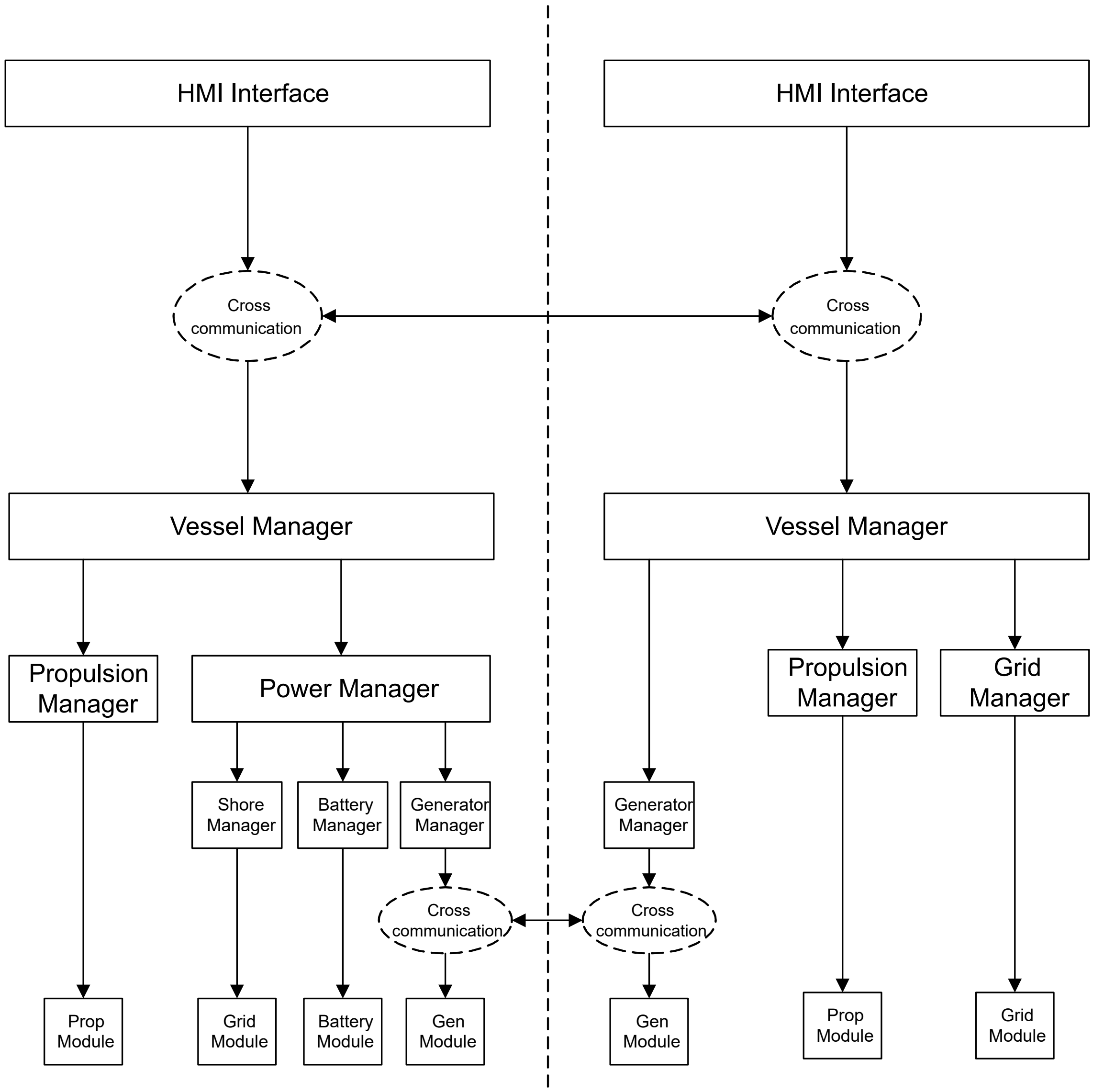


Figure 3.11: The ECS software structure is modular. Modules are re-usable but due to differences between vessels the manager modules are more or less customized to cater the vessel needs.

Highest layer in the system control is the vessel manager. The duties of this manager are to execute the commands given by the operator, such as starting the vessel from a cold and dark state where no systems controlled by ECS are active. The state where the DC-link is energized by a producer and hotel grid is up is referred to as *vessel on*. When the vessel is on the propulsion system may be started. Typical modes present in a battery-hybrid vessel include diesel-electric, hybrid, full-electric, and shore. Upon turning the vessel on the power manager is enabled, which then enables other producer controlling managers as required by the mode.

The propulsion manager and module control the propulsion system. The manager interprets the throttle signal and commands the inverter to start running the motor at requested speed when the throttle lever is moved. The propulsion hardware module is responsible for setting the command word which is sent to the inverter appropriately and reading the statuses off the inverter. The module interface includes relevant data from the inverter such as speed, power, torque, temperatures, and faults, which are passed to the HMI interface.

The shore manager is responsible for controlling the AFE that provides power to the DC-link from the shore AC grid along with other tasks such as controlling the pre-magnetization of the shore transformer and the closing of contactors. The grid module can be either in shore or hotel grid mode and can be changed on the fly. In some instances the same converter is used for both purposes, in this example there are dedicated shore and hotel inverters, with the shore converter situated on one side and the hotel inverter on the other side.

Battery manager and module control the DC/DC converter and the third-party battery management system (BMS). When the battery is enabled the BMS is commanded to connect the batteries. If the DC-link voltage is low compared to the battery voltage, the connection is first made through a current limiting pre-charge resistor. Once the batteries are connected the converter is started and the batteries will start to charge or discharge depending on the availability of other producers and their voltage reference. When in full-electric mode, the batteries are the only producers and will discharge until the state of charge becomes too low at which point the power manager brings the generators up to avoid black-out. In diesel-electric mode the voltage reference is set such that batteries are charged from the gensets, whereas in hybrid mode the batteries assist the gensets by responding fast to load changes while the generators can ramp their power up slowly, thus reducing stress and emissions.

Control of generators is done by the generator manager and module. The manager handles that enough generators are running. When there are multiple gensets the number of running gensets is adjusted to match the total load automatically. When the load is above some threshold for a certain amount of time a new genset is brought up. Conversely, when the load is below some value a generator is automatically shut down. This way the engines are loaded near optimally at all times. One side controls all the generators by communicating with the other side PLC, if the cross communication fails, both sides revert to controlling gensets on its side. The hardware module controls the generator inverter and commands the prime mover to start and stop and adjust its speed reference. The speed reference follows a load curve that optimizes the specific fuel consumption. Starting a genset involves first starting the prime mover and waiting for it to stabilize which is usually indicated by a ready-to-load signal by the prime mover. When the prime mover is running the inverter is started in voltage control and its torque limit is smoothly increased. In a DC-based system generators can be brought fast up as there is no synchronization with already running generators as in AC systems. The running generators need to share the load and it is achieved by drooping the voltage reference which means the voltage reference of an individual generator is decreased as its load increases.

4 PROGRAMMABLE LOGIC CONTROLLERS, PROGRAMMING AND TESTING

Programmable logic controllers are widely used in automation applications, an example being controlling an industrial process. When a new automation system is commissioned its functionality is tested. Software testing concepts are also applicable to testing of PLC programs. In the case of ECS, the system functionality is demonstrated to the customer and a classification society in three stages.

4.1 PLC functionality

Before microprocessor-based PLCs were available, automation logic was implemented using relays and other electromechanical components [24]. As more complex logic was needed the systems grew in size. Together with the increasing number of possible points of failure and trouble of altering the logic due to the hard wiring a better solution for automation control was developed, known as a programmable logic controller. PLCs are digital computers designed for reliability and operation in harsh environments. Emphasis was put on making learning the new controllers easy for automation engineers and extensive program monitoring and manipulation features.

A picture of a modern PLC was shown in previous chapter in Fig. 3.10. The functionality of a PLC can be attributed to its processor, memory, and physical inputs and outputs. Inputs are used to read the status of buttons, switches, sensors, and other devices. Outputs interact with other field devices by e.g. operating actuators. Memory stores the state of variables and the program that the processor executes. A PLC cycle consists of reading the inputs, executing the program and writing the outputs. The cycle is executed periodically, for example every 10 ms. Executing the cycle will take some time depending on how heavy the program is after which the rest of the time is spent idling or performing "housekeeping" tasks. The cycle should be executed very precisely at the specified interval for the PLC to operate deterministically. The random variation in the interval between cycles is jitter and for a PLC is typically in the order of microseconds. As such PLCs are real-time systems.

To achieve this real-time operation the PLC contains a real-time operating system (RTOS). A RTOS must have minimal interrupt latency and must complete tasks within a deadline. A hard real-time system can always meet the timing requirements [25]. To execute the PLC cycle the operating system interrupts the currently running task and changes to executing the PLC task. Preemptive scheduling is used to interrupt a running task, saving its context and changing to a higher priority task that needs to be executed at that time, after which the interrupted task will continue from where it was interrupted. PLC manufacturers use different operating systems such as *RTLinux* and *VxWorks*.

Beckhoff PLCs differ from others in the market in that the PLCs are implemented in software on Windows PCs. Real-time capability is provided with a custom kernel that runs alongside the Windows kernel. One benefit of software PLC is that the program can be run in real-time on the development PC i.e. normal office computer, permitting fast prototyping as there is no need set up a hardware PLC and its power supply. In addition to the software, Beckhoff manufactures embedded and industrial PCs, such as the CX9020, that are optimized for real-time applications and provide the interfaces for communicating with field devices. Such a PC is in this thesis referred to as a PLC. A software PLC running on the development PC is referred to as a local PLC.

4.2 PLC programming

Several languages are available for programming PLCs. The languages are standardized under the third part of the IEC 61131 standard, which means the same languages are used to program PLCs from different manufacturers. The development environments, however, do differ between brands.

A PLC program consists of one or more program organization units (POU). A POU can be either a program (PRG), function (FUN) or a function block (FB). The main program of a modular PLC project is a PRG which calls other POUs. The main program is called by the cyclic PLC task. Difference between FUN and FB is that a function's outputs are always the same when called with same inputs, when the function does not use global variables. The function's local variables are always declared when the function is called and thus their values do not retain between calls. A function block on the other hand must be instantiated before calling and its local variables are retained, i.e. it has its own memory. In addition to POUs, there are global variable lists (GVL) where global variables are declared. Variables that are linked to the physical IO are usually made global. User specified data types such as structures and enumerations can also be created, and they are known as data unit types (DUT).

4.2.1 IEC 61131-3 languages

The standard describes five languages that can be used when creating PLC projects. Within a project the language can be changed between POUs. Ladder diagram (LD), function block diagram (FBD) and sequential function chart (SFC) are graphical whereas structured text (ST) and instruction list (IL) are textual. Usage of the languages differs regionally. Generalizing, LD is more widely used in America whereas Europeans prefer ST.

Ladder diagram

A ladder diagram program looks very much like a schematic for a relay logic circuit and was designed for easy adaptation by e.g. technicians who don't have programming experience. The two building blocks unique to LD are contacts and coils. Contacts work as switches that provide a path for "electricity" to energize a coil. A simple program that could be used to start and stop a machine is shown in Fig. 4.1. In the declaration part there are three boolean variables; start, stop, and running. Suppose start and stop are linked to physical digital inputs such as buttons and running is a digital output which determines if the machine is running. The start/stop logic is implemented using three contacts and one coil. When the start button is pressed the start contact is closed, the stop contact is normally closed so the coil will be connected to "power" and the output would turn on. Simultaneously as the coil is energized the running contact will provide a parallel path so the machine will stay on after the start button is no longer pressed. Pressing the stop button cuts power to the coil and the machine is stopped. LD is especially useful for handling boolean logic, more complex tasks such as feedback controllers can be implemented in function blocks and called from the LD program.

Structured text

Structured text is a textual high-level language that is in syntax similar to Pascal. A ST program is a series of statements, program flow is altered using if-else and case statements and iteration is achieved using for and while loops. The example program shown in the LD section could be realized in a single ST statement:

```
PROGRAM PRG_LD
VAR
    bStart   : BOOL ;
    bStop    : BOOL ;
    bRunning : BOOL ;
END_VAR

```

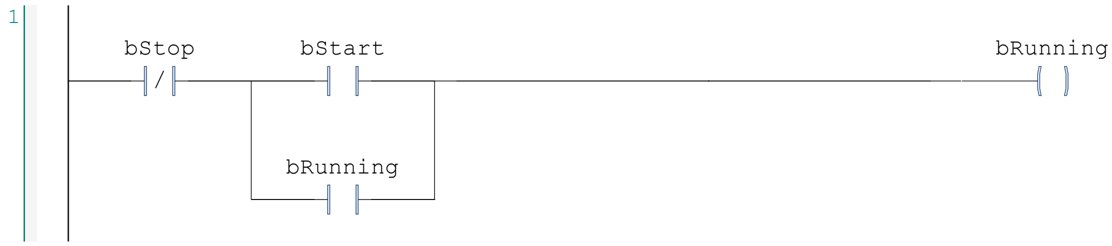


Figure 4.1: A PLC program programmed in LD that is used to start and stop a machine.

```
bRunning := (bRunning OR bStart) AND NOT bStop;
```

Alternatively, if statement could be used:

```
IF bStart THEN
    bRunning := TRUE;
END_IF
IF bStop THEN
    bRunning := FALSE;
END_IF
```

ST is suitable for all sorts of PLC programming tasks, especially so for complex programs with mathematical calculations and arrays.

Function block diagram

A function block diagram consists of a number of function blocks with connections between their inputs and outputs. Programming in FBD resembles drawing a block diagram of the system. A FBD program implementing the machine start/stop logic is shown in Fig. 4.2.

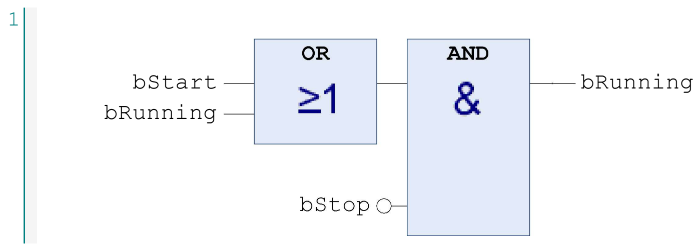


Figure 4.2: A PLC program programmed in FBD that is used to start and stop a machine.

Sequential function chart

A sequential function chart resembles a flowchart. It has a number of steps which contain actions. Between steps there are transitions. A SFC program is shown in Fig. 4.3. There are two steps; stopped and running. The steps contain actions that can be programmed with the other languages, in this case the steps contain ST statements that set the running variable either true or false. When the transition condition evaluates to true the program transitions to the following step. SFC is especially useful in making state machines.

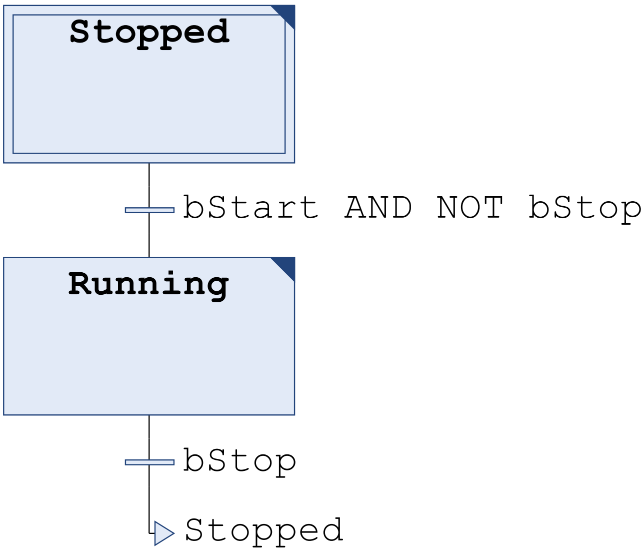


Figure 4.3: A PLC program programmed in SFC that is used to start and stop a machine.

Instruction list

Instruction list is a low-level textual language that resembles assembly language. The start/stop logic is implemented in IL as:

```
OR bStart
OR bRunning
ANDN bStop
ST bRunning
```

IL has been deprecated in the third edition of IEC 61131-3.

4.3 PLC software testing

As with any piece of software, PLC programs need to be tested to find errors. As the PLC interacts in real-time with the field devices, proper testing prior to commissioning, when the devices are not yet available, poses a challenge. If errors can be found and fixed before commissioning the time required for commissioning decreases along with costs. PLC development environments have good debugging tools where the variable values can be seen and altered. The use of breakpoints allows stepping the program forward one statement at a time.

The software development and testing process can be divided to several phases where for each development phase there is a corresponding testing phase. System design determines what roles software should have in the system. A set of requirements for the software are formulated after which the required modules are designed and how they interface with each other. When the individual modules and their functionality is defined code is produced. Unit testing focuses on the modules and tries to show that they don't fulfill their requirements. After testing the modules, integration testing focuses on the architecture of the software and how the modules are combined. Once the modules are integrated, system testing takes place where the fulfillment of requirements established earlier are tested. After the system level requirements are met, acceptance testing determines if the system fulfills its requirements.

Different testing methods are used at different stages. White box testing is done by individuals who know the inner workings of the software. Tests are designed such that e.g. all the branches inside the software are executed at some point and is usually employed in unit testing. Black box testing, on the other hand, is done without knowledge of how the software works to try to fulfill its requirements and is suitable for system and acceptance testing. Integration testing can be done by integrating all the modules at once or incrementally adding modules one-by-one. [26, 27]

Testing of an industrial control system that controls, for example, a manufacturing process is traditionally first properly tested during commissioning when all the controlled actuators and sensors are installed. A significant time of the commissioning is taken by fixing errors in the control system logic and design. To make the actual commissioning time shorter, virtual commissioning (VC) methods have been developed [28, 29]. Virtual commissioning involves replacing real field devices with simulation models. The simulation reads the outputs of the PLC as inputs, executes the models and generates outputs that are read as inputs by the PLC. One study showed that employing VC increased software quality in terms of fulfilled requirements by 100 % and decreased commissioning time by 75 % and total time-to-market by 15 % [30]. The total time savings depend on the time required to build the required simulation models with adequate trueness to real life. Small and simple PLC programs can be tested by manually writing the inputs and looking at the outputs. However, creating a model of a complex system such as one involving kinematics takes a lot of effort and time. Other benefits of VC are the ability to test fault situations safely without needing actual faulting field devices.

Virtual commissioning can be performed by utilizing hardware-in-the-loop (HIL) simulation, where the control software runs on real control hardware that is connected to a real-time simulation running on separate hardware such as another PLC [31]. In HIL, from the point of view of the tested software, the simulated system is undisguisable from the real system. A HIL simulator uses the same fieldbuses and IO as the actual system or emulates its behavior to take into account e.g. delays.

Another approach to simulation is to use a program running on a general-purpose computer. To connect the PLC to the simulation, OPC (Open Platform Communications) can be used. OPC is a series of standards for communicating with industrial controllers. A computer application that implements the standards can communicate with PLCs of different manufacturers. When running the simulation in a non-real-time environment the simulation is not deterministic and can lead to unexpected behavior due to time jitter and slow sampling [32].

4.4 Testing of Editron Control System using simulation

When developing the control software for a vessel project, testing without simulation is troublesome. Without real or simulated devices the different modules do not receive the feedback required and e.g. state machine state transitions fail to timeouts. Different projects use a varying number of the same hardware modules which have been formerly tested and their functionality proven in unit tests. The testing to be done during development then focuses on the manager modules and their integration into a complete system. To properly test the managers the hardware modules which they control should be functional which can be achieved by simulation. System and acceptance testing can also be practiced in advance before doing them properly with the actual system setup.

The control system is formally tested in three stages of acceptance testing; factory, harbor, and sea acceptance testing (FAT, HAT, SAT). Even though the term acceptance testing is used, the testing done can be considered to be also integration and system testing. When a vessel is built it is assigned a class by a classification society which certifies that it is built according to a set of rules and standards maintained by the societies. Classification is often also a requirement for insuring the vessel. The acceptance testing programs are to be accepted by the classifier. FAT is done for the system components such as electric machines and the ECS at the manufacturing premises.

The FAT for ECS typically includes the PLC cabinets which contain the PLC, network switch, relays, and terminal blocks to which the HW IO signal wires coming from outside the cabinet are connected to. An HMI panel may also be included in the FAT. As there are no converters, engines etc. present the tests are limited to e.g. indications of digital inputs and temperatures in the HMI and the generation of alarms. The redundancy of network connection in the event of a single point failure as required by class [33] is also demonstrated. The FAT can be considered to be more of integration than system testing as the system is still very much incomplete at that stage. The use of simulation of field devices has the potential of expanding the coverage of FAT to include system functionality such as starting the vessel up from dead-ship state, changing of operating modes, and power plant control.

HAT is done during commissioning at the shipyard. The testing is done with the vessel moored and the purpose is to demonstrate the functionality of the whole system and that the vessel is ready for sea trials. During the HAT phase all the parts of the system are present and thus system testing can be done as far as practical. SAT is the final testing phase and is done at sea. Full power of the vessel it at disposal and last system level tests can be performed. When the SAT is successfully completed, and all possible remarks are addressed the system commissioning is finished.

4.4.1 Simulation requirements

To develop a simulation solution for testing, a set of requirements and the level of simulation detail needed needs to be identified. The purpose of simulation is to test the PLC software of the ECS. Testing is made possible by reacting to the program outputs and generating inputs so that the program can enter different states. As an example, the program commands a motor inverter to start running the motor. The program expects feedback that the inverter is running or else it will time out and issue a start timeout fault. As it is the inverter and not the PLC which performs the control of the motor, no electromagnetic model of the motor is needed. Another example is some bi-stable device such as a contactor which can be either open or closed and it is actuated by a PLC output. The simulator's response to the open command would then be to indicate that the contactor is open. Discrete event simulation can thus be used to model the field devices. Discrete event models have a number of states which change in response to events. Between the events no changes occur in the model.

In addition to the discrete event simulation, a model of the power dynamics is needed to test e.g. generator control algorithms. When the simulated propulsion is running it should consume power that the generators produce. Other quantities related to power, speed and torque should also be represented so that the HMI panel, if present, can also be validated to show the information correctly. To model the system dynamics a set of differential equations that describe the system are formulated. As the goal of simulation is not to e.g. tune propeller acceleration such that the diesel engine power does not rise too rapidly, the level of simulation detail can be kept reasonably low.

The primary purpose of the simulation is to make effective testing during the development process possible. The physical environment where the testing is done is the office or home office and as such setting up the simulation environment should be straight forward and not require special hardware, a PC and a PLC should be everything that is needed. The PLC IO should be mimicked virtually and thus no IO cards or wiring is needed to run the simulation. The PLC project that is to be tested should require minimal configuration changes to adapt to simulation with no changes to the actual code.

One great advantage of simulation is the ability to simulate fault conditions which would be troublesome to generate with the real devices. The simulation tool should be interactive where the user can issue commands to trigger fault situations such as converter fault and too high temperatures. The throttle lever input signal should also be manipulatable from a user interface.

A list of requirements general requirements for the ECS simulation tool along with requirements for individual simulated devices is given here.

General requirements

- No hardware in addition to a development PC and a PLC is required.
- No modification of source code needed.
- User interface for interacting with the running simulation.
- Ability to simulate a system that uses two PLCs.

DC based power system

- Simulates DC link voltage as a function of total power.
- Simulates two DC links that are connected with a bus tie.

Prime mover

- Responds to start and stop digital output signals by giving a ready to load input signal.
- Responds to speed reference analog output signal with actual speed analog input signal.

Power electronic converters

- Simulates general command interface.
  - Responds to enable command with ready to run status.
  - Responds to run command with running status when ready to run.
  - Responds to reset faults command by clearing fault status and fault word.
  - Gives fault status, clears running and ready to run status when fault command is given from user interface, ability to set fault word value.
- Simulates converter working as a propulsion motor inverter.
  - Simulates motor speed, torque, and power as a function of speed reference.
- Simulates converter working as a generator inverter.
  - Simulates torque and power as a function of DC voltage reference.

    - Simulates load sharing between several generators.

- Simulates converter working either as a hotel grid inverter or shore inverter.
- Simulates converter working as a DC/DC converter.

Other HW IO devices

- Simulates a controlled bi-stable device.
    - Responds to set and reset commands with a corresponding status.
- Simulates an uncontrolled bi-stable device.
    - Toggles its state and gives status indication when command is given from user interface.
- Simulates a throttle lever.
    - Throttle movable with user command, throttle position given as an analog input signal.
    - Gives lever at zero indication as a digital input.
- Simulates temperature measurements.
    - Measurement changes with user command.
    - Simulates loop fault such as broken wire.

4.4.2 Simulation tool options

To build the simulation solution and implement the functionality fulfilling the requirements a number of methods was looked into and one chosen. The most straight-forward option would be to implement the simulation on the same PLC that the system to be tested runs on. The simulation could be implemented as a program that is called last from the main program. By disabling the physical IO, the simulation program could read and write to the variables that are located in the input and output memory addresses. The drawback of this kind of implementation is that the simulation may affect the program which is tested by consuming real-time resources. It was decided that the simulation is implemented on a separate piece of hardware.

The simulation could be implemented on another PLC. The benefits of using a PLC is that it is a real-time system. Simulating dynamical systems involves sampling the inputs at a known interval and doing numerical calculations and updating the outputs before the end of the sampling period. To connect the simulation PLC to the PLC that runs the software to be tested, OPC could be used. Because the ECS does not use OPC it would require the extra step of setting up the OPC server and making the relevant variables available to the server, which would require changes in the source code. Because the simulation requirements can be fulfilled with discrete event simulation and with simple dynamic models that don't have to be very accurate, a choice was made to use a PC program for the simulation. Using a PC simplifies the hardware requirements as no extra PLC is needed and testing can be done on a single PC by running the PLC project on a local PLC.

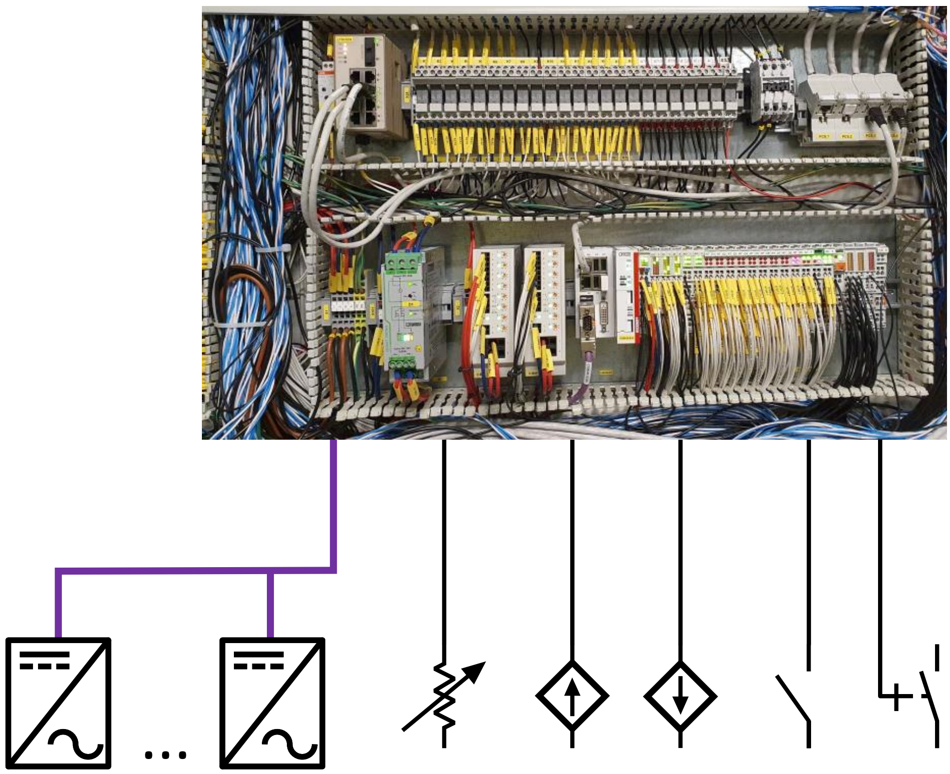
Before selecting how the simulation should be implemented, the method of communication between the PLC and the simulation should be decided. Because of the cumbersomeness of using OPC and the usage of Beckhoff PLCs, ADS (Automation Device Specification) proved to be effective. ADS specifies the communication between Beckhoff software modules.

Libraries are provided by Beckhoff for creating custom applications that can use ADS for communication. For implementing the simulation, instead of using commercial software such as WinMOD or Simulink as a platform, it was chosen to make a custom program using the ADS libraries. Using ADS has the major advantage of being native to the Beckhoff PLC environment; no additional software is required to be installed to the PLC and also no changes are required in the PLC program to be able to access its variables.

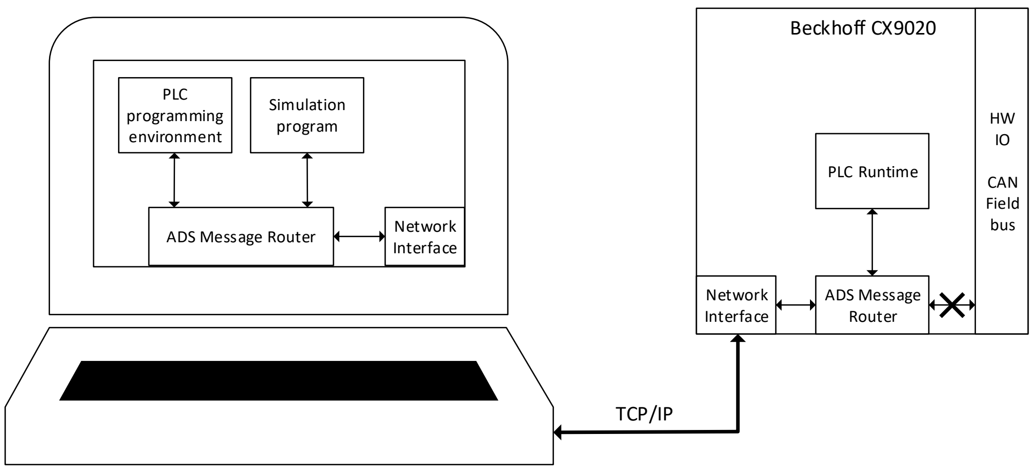
With the availability of an ADS library wrapper *pyads* [34], Python was chosen as the implementation language for the simulation. Python is an interpreted language that has a clear syntax and is easy to pick up by a programmer. Due to interpretation there is no compilation between writing and running code, making it fast to make modifications to the program. The extensive availability of libraries makes Python suitable for a large range of tasks. Alternatively, the simulation tool could be programmed using .NET framework or C++ language. Python was chosen mainly because of previous experience; no time was consumed to learn new programming environments and workflows. In addition, pyads documentation contains clear examples which made it fast to get started.

Figure 4.4 illustrates the setup when real field devices are used and when they are replaced with simulation models. Fig. 4.4a shows the insides of a cabinet with a CX9020 PLC installed in a vessel and what kind of wiring is required. Multiple converters are connected to the PLC via the CANopen fieldbus. Discrete wires connect resistance temperature detectors, analog inputs, analog outputs, digital inputs, and digital outputs which are illustrated with the variable resistor, controlled current sources, switch, and controlled switch symbols.

Figure 4.4b shows the simulation setup. A general office computer has the Beckhoff software installed which contains the development environment and a ADS message router. The ADS communication is integral to the Beckhoff system and it is used to log in to the PLC and download the program. With the help of the ADS libraries and pyads, the simulation Python program can also use ADS to communicate with the PLC. An ethernet cable connects the devices through network interfaces. The PLC has the ADS message router, too, along with a PLC runtime which executes the PLC program in real-time. As there are no field devices connected to the PLC, the HW IO and CAN fieldbus is disabled as to not interfere with the inputs coming from the simulation. The PLC runtime resides also in the development computer which makes it possible to run a testing setup with one PLC on the development computer.



(a) PLC and peripherals installed in a control cabinet. Wiring connects the different field devices installed in the vessel to the PLC.



(b) Testing setup realizable in an office environment. The development laptop is running a simulation program which interfaces with the PLC over ADS connection. The PLC has its IO and fieldbus devices disabled.

Figure 4.4: Real setup installed in a cabinet and a testing setup where field devices have been replaced with simulations.

5 DEVELOPMENT OF THE SIMULATION TOOL

After selecting the programming language, the simulation tool development started with laying out the general architecture on which the simulations of the individual devices are built. The architecture contains program structures for communicating with the PLC, calculating dynamic models at set step intervals, a configurable main program where the simulation is configured for a particular vessel project, and a text-based user interface.

5.1 General architecture

The simulation program is divided into two parts; a library of devices and a main program where the devices to be simulated are configured to match the system that is to be simulated and where the user interface is executed. Object-oriented programming is used where each simulated device type is a class. Each class has properties and methods that define the functionality. In the configuration part, instances of the classes are created according to what devices are part of the system.

5.1.1 Device library

To implement the simulation models in the device library, program structures for communicating with the PLC, executing simulation code at a specified interval, and executing user commands are needed.

Communication

To communicate with the PLC, the device library contains a class for a PLC variable which the simulator reads or writes. The class has properties for the name, which is a string that identifies the variable in the PLC and for value, size, and data type. To read variables off the PLC, ADS notifications are used. The class has a method for subscribing to receive notifications, meaning every time the variable changes in the PLC a notification is sent to the simulation program and a function given as input to the method is called. This has the advantage that the simulator does not have to constantly read the variables from the PLC, reducing the overhead when the variable is changing infrequently. Execution of the callback function happens automatically in the background. For writing a PLC variable the class contains a write method.

Below is a sample code that demonstrates the communication with the PLC using pyads and the created class for a PLC variable. First an instance of *pyads.Connection* is made using the PLC address and port number and the connection is opened. The PLC has two variables declared in its Main PRG; integer *nNum* and boolean *bFlag*. Instances of the class *PlcVariable* are created for each variable. When *bFlag* changes state, an action, in this case storing its value, needs to be done. For the action a function is declared, and it is passed to the subscribe method. Finally, the value 5 is written to *nNum*.

```
plc = pyads.Connection('5.48.1.251.1.1', 851)
plc.open()
plc_int = PlcVariable(plc, "MAIN.nNum", "INT")
plc_bool = PlcVariable(plc, "MAIN.bFlag", "BOOL")
def read_plc_bool(handle, name, timestamp, value):
    plc_bool.value = value
```

```
    return read_plc_bool
plc_bool.subscribe(read_plc_bool)
plc_int.write_to_plc(5)
```

Cyclic code execution

Cyclic code execution is needed to e.g. numerically integrate or to write simulated values to the PLC at some interval. Each simulated device whose model includes dynamics should execute its calculation at some frequency and update the results to the PLC. Python threading module is used to run several loops concurrently. When using Python threads, however, only one processor core is used which executes one thread at a time but switches rapidly between the different threads. Time module is used to sleep the thread until next execution.

Below is a sample code that demonstrates the execution of simulation code 10 times in a second in a separate thread. A function *simulate_ dynamics* is defined where there is an infinite loop. The simulation is executed in *do_ simulation()* after which the sleep time is calculated. If the sleep time is negative the code execution has taken too long to complete at the specified interval, in which case the sleep time is set to zero. Additionally, a warning could be issued. In the main program a new thread is created which executes the function. By setting the daemon property the thread execution ends when the main thread ends.

```
Ts = 0.1
def simulate_dynamics():
    next_call = time.time()
    while True:
        do_simulation()
        next_call = next_call + Ts
        sleep_time = next_call - time.time()
        if sleep_time < 0:
            sleep_time = 0
        time.sleep(sleep_time)

t = threading.Thread(target=simulate_dynamics)
t.daemon = True
t.start()
```

User commands

For processing user commands, each device class has a dictionary with the commands and arguments. To execute a command a method is called with the command name and the parameter.

5.1.2 Main program

In the main program first the PLC connections are made and opened using *pyads.Connection* instances. The simulated device instances are then created, devices with user commands are placed in a dictionary. Example is shown below, where the *Devices* dictionary contains an instance of *MotorInverter* with the key “inverter”. The simulated inverter class has inputs for the command and status word which are declared in the PLC global variable list. When a real inverter is used the command word is sent to the inverter from the PLC via the CAN bus and the status word is read back, but when the inverter is simulated the simulator looks for changes in the command word and creates the status word and sends it to the PLC.

```
plc = pyads.Connection('5.78.103.24.1.1', 851)
plc.open()
Devices = {"inverter" : MotorInverter(plc,"GVL.nCmdWord","GVL.nStatusWord")}
user_interface(Devices)
```

After the devices have been declared the user interface program is called. The user interface runs the loop of asking the user for a device name with Python's *input()* and checks from the dictionary if such a device has been declared. If so, the list of available commands for the device is shown. When the command and its argument is entered, the command gets executed and the interface returns to asking for device name. Because all the simulation tasks run in the background in separate threads the user interface stays responsive.

5.2 Device simulations

The device library contains classes for the different simulated device types. When an instance of the class is created the simulation for that device starts executing.

5.2.1 DC link

The DC link class models the power connections between the power electronic converters. Power flows into or out from the DC link towards the converters or to another DC link via the bus tie. The DC link voltage changes according to the net power which increases or decreases the voltage. The voltage measurement is given to the PLC via an analog input.

The DC link is modeled as a capacitor that stores energy in an electric field. The amount of energy is proportional to the capacitance and the voltage squared according to equation

$$E = \frac{1}{2}CU^2\,, \tag{5.1}$$

where E is energy, C is capacitance, and U is voltage. The energy is found by integrating the net power over time.

The class has inputs for a list of converters that are connected to the DC link, the name of the voltage measurement variable in the PLC, various parameters such as the capacitance, and a second DC link instance. Two methods, one for calculating the voltage and the second for updating the PLC voltage measurement are defined. The methods are executed cyclically as described in section 5.1.1.

The voltage calculation starts by summing up the powers of the converters and the bus tie power. If a second DC link instance is given as input, the bus tie power is modeled to be proportional to the square of the voltage difference between the links, making power flow into the link with smaller voltage. Once the net power is calculated it is then integrated into energy. Backward Euler method is used for integration. The energy at current time step is found by adding current value of power multiplied by the time step to the previous value of energy, as given by equation

$$E_k = E_{k-1} + T_\mathrm{s}P_k\,, \tag{5.2}$$

where k denotes the time step index and T_s is the sample time. The voltage is found from the energy according to equation (5.1).

5.2.2 Motor inverter

The motor inverter handles simulating the propulsion and generator inverter command interface. The detailed working of the inverter such as FOC and SVPWM described in sections 3.1.2 and 3.2 are not considered because those are controlled by the converter firmware which is not related to the PLC software.

The class implements reading the command word, control mode, and voltage and speed references from the PLC by using the ADS notifications. Statuses written to the PLC include status and fault words, speed, torque, power, DC voltage, winding and internal temperatures. The command word contains bits for enable, run, and fault reset commands, whereas the status word has ready to run, running, and fault statuses. When the command word changes the status word is modified according to the requirements in section 4.4.1 and written to the PLC. The simulation of speed, torque, and power is implemented in a separate electric machine class. The DC voltage is received from the simulated DC link.

The simulated motor inverter has user commands for setting the fault status along with the fault word and changing the inverter temperature measurements. Simulating faults is useful for testing the system behavior when a fault occurs and by changing the temperature values it can be verified that the HMI displays them correctly.

5.2.3 Electric machine

The electric machine class implements the simulation of an electric machine operating either as a generator or a motor. Because the electric machine can have multiple stator windings a list of motor inverter instances is given to the electric machine instance as an input. The control mode command of the inverter decides whether motoring or generating is simulated. For generation an instance of a prime mover is used to couple the two. A bistable device can be used to simulate the operation of a clutch between the prime mover and electric machine.

Speed control

When the connected inverter is in speed control mode, the simulation loop consists of first checking if the inverter is running and the speed reference. Zero speed reference is used if the inverter is not running. The closed-loop dynamics of a real motor is approximated by following the speed reference according to first order dynamics. No electromagnetic model is implemented where the voltage applied to the windings by the converter causes current to flow and magnetic flux to form which leads to torque production. Instead, the torque follows the speed according to a pre-determined curve. From the speed and torque, power is obtained. The torque and power is then updated to the inverters by equal division. This way the simulation mimics the operation of a real propulsion motor without the model being overly complex relative to the simulation requirements.

The speed is modeled with a first order time-continuous transfer function of the form

$$\frac{n(s)}{n_{\mathrm{ref}}(s)} = \frac{1}{\tau s + 1}, \tag{5.3}$$

where n, n_{ref} are the speed and speed reference of the motor, τ is the time constant, and s is the complex Laplace variable. To compute the response the transfer function is discretized using zero-order hold to obtain the difference equation describing the system. Zero-order hold keeps the input constant between samples. The discrete transfer function is

$$\frac{n(z)}{n_{\mathrm{ref}}(z)} = \frac{1 - \exp\left(-\frac{T_{\mathrm{s}}}{\tau}\right)}{z - \exp\left(-\frac{T_{\mathrm{s}}}{\tau}\right)}, \tag{5.4}$$

where z is the z-domain variable comparable to the Laplace variable s [35]. By writing $\alpha = 1 - \exp(-T_\mathrm{s}/\tau)$ the computationally convenient difference equation for speed

$$n_k = n_{k-1} + \alpha(n_{\mathrm{ref,k-1}} - n_{k-1}) \tag{5.5}$$

is obtained.

Generally, the load torque on a propeller is proportional to rotational speed squared. When the current speed is calculated the torque is determined from the relation. Nominal speed and torque are given as inputs and the torque is nominal when speed is nominal. The power is the product of torque and angular velocity.

Voltage control

When in voltage control the inverter speed controller is replaced by a voltage controller. The controller applies torque to regulate the voltage in the DC link. Torque whose sign is opposite to the rotational speed causes mechanical power to be converted to electrical power which increases the voltage. Voltage control is used to control genset power. Several parallel gensets need to share their load which is achieved by drooping the used voltage reference as the load increases. The produced torque is limited such that the machine never works as a motor to turn the prime mover.

When the inverter is in voltage control mode the simulated electric machine adopts the speed of the prime mover that is connected to the machine. Torque is applied in the opposite direction of rotation causing the the machine to function as a generator. To distribute the load of the simulated generators evenly simple droop control is used in determining the magnitude of torque. If the voltage is a certain amount below the reference, then nominal torque is applied. The torque is reduced to zero by the time the voltage is above the reference by the droop amount. For example, if the droop is 4 % and voltage reference is 750 V then nominal torque is applied when the voltage is $0.96 \cdot 750\,\mathrm{V} = 720\,\mathrm{V}$ and zero torque when voltage is $1.04 \cdot 750\,\mathrm{V} = 780\,\mathrm{V}$. Between the points the torque varies linearly.

When the DC link is not yet energized and the generator is started but the inverter is not yet running, the emf of the generator charges the DC link through the IGBT body diodes. To model the behavior the torque is set to be proportional to the voltage difference between the DC link and the generator.

5.2.4 Prime mover

The prime mover class models the operation of a prime mover such as a diesel engine which is controlled by start, stop, and speed reference signals. Torque production through burning fuel and completing a thermodynamic cycle is not modeled. Instead, when started, the speed follows the speed reference with first order dynamics as described in section 5.2.3 regardless of the simulated load on the prime mover. In a real system putting load on an engine has an effect on the speed and the engine can be stalled or overstressed if too high a load is applied too fast. As the goal is not to simulate the system with such precision that parameters such as torque ramps can be tuned, a simple speed reference following model is sufficient. After issuing the start command the PLC waits for the ready to load feedback before starting the generator inverter. The prime mover is modeled to be ready for load when its speed is at or above the idle speed.

For testing of fault handling the class has user commands fault and stall. When the fault command is given the engine stops responding to start and stop commands, when stalled the engine is shut down.

5.2.5 Hotel and shore inverter

The hotel and shore inverter class implements simulation for an inverter working as either hotel microgrid or shore AFE. The class has the same features as the motor inverter class but with the addition of power simulations for the hotel and shore modes.

Hotel grid simulation

When the inverter is operating as a hotel grid inverter the power flows from the DC link to the hotel AC grid and the inverter follows the reference AC voltage and frequency. As it is the load that determines the power, a user command is provided for setting the apparent power. The simulation responds to the voltage and frequency references by setting the corresponding statuses to the reference value. Power factor can be set by a user command for HMI verification.

Shore simulation

When the inverter works as an AFE to connect to the shore, power flows from the grid to the DC link. The power follows the DC voltage droop curve similar to the generator simulation but as rotation is not involved the power is set directly.

5.2.6 DC/DC converter

The simulated DC/DC converter works very much like the simulated shore AFE. The difference is that the power is bidirectional meaning the battery is either charging or discharging. The droop curve controls the DC voltage by taking power from the DC link when the voltage is above the reference and vice versa.

5.2.7 Throttle lever

The throttle lever class enables setting the speed references of the simulated propulsion motors. By changing the simulated lever position the simulated power load can be easily varied. The position is changed with an user command by entering the position as a percentage where 100 % is full ahead and −100 % is full astern. The lever's range can also be configured to between zero and full ahead, in case reversing is not done by changing the direction of rotation. The given position is then scaled to the 16-bit integer range and written to the PLC analog input. In some cases, the lever also gives a separate zero position signal which needs to be active before e.g. motor can be started. The signal is active when the lever is at zero within a tolerance.

5.2.8 Temperature measurement

The temperature measurement class simulates the measurement of e.g. filter and transformer temperatures using a resistance temperature detector. The measured temperature is changed by a user command to enable the testing of alarm generation and tripping of equipment due to too high temperatures. The real temperature measurement card features monitoring the signal validity and can detect a broken or short-circuited wire. The response to faulty measurement can be tested by inputting the PLC variable that contains the error information to the simulated device and setting the error state with a command.

5.2.9 Bistables

Three variants of bistable devices are simulated with the bistables classes. One is controlled by two signals, the second with a single signal, and the third is uncontrolled. The state of the bistables is given to the PLC with digital inputs and can represent information such as the open/closed status of a contactor, whether a fuse is blown or if a clutch is engaged.

The bistable device with two control inputs changes state to true or false when the respective command has a rising edge. The state is communicated back to the PLC. User commands can be used to change the state manually. The variant with one control input simply follows the input. The third variant has no control inputs and its state only changes with user command.

6 TESTING WITH THE SIMULATION TOOL

The developed simulation tool was used with two ECS projects. The first one, a previously commissioned double-ended ferry project was used to mainly test the simulations. Because it was known that the PLC program was functioning correctly, established by the completed FAT, HAT, and SAT, it could be used two find errors in the simulation program and verification that the simulation reflects the system with adequate accuracy. The second project, a vessel with a parallel-hybrid water jet propulsion, was simulated to test the PLC logic and the included HMI panel before commissioning to establish the feasibility of using simulation in testing of the ECS.

6.1 Double-ended ferry

The vessel completed in the first project is a small double-ended road extension ferry. The ferry is powered by four generating sets. Propulsion is powered by two electric L-drive azimuth thrusters where each motor has two sets of stator windings and is run by two inverters. Hotel grid is powered from the DC link by a hotel inverter and it's duplicated for redundancy. The system is divided into two symmetric parts; aft and fore.

External systems communicating via Modbus with the ECS are the HMI and the third-party azimuthing propulsion system. The azimuth system gives signals for enabling the electric motor and its speed reference. Modbus devices are not included in the simulation and as such these signals are set manually in the PLC programming environment with the online write functionality. The HMI panel was included in the ECS scope of delivery and its software can be run on a normal PC, making it possible to command and view the state of the simulated vessel the same way as on the real vessel.

The most complicated part of the system in regard to the ECS is the control of generators. The four gensets are started and stopped automatically based on the load. When the load is above a threshold for a certain amount of time, the next genset is started. Similarly, when the load is below a threshold for long enough, a genset is shut down. The user can change the thresholds and delays from the HMI. Additionally, the minimum number of running gensets and priorities can be adjusted. The priorities define the order at which gensets are started. The feature can be used to reduce engine running hours by shutting down one or more engines for the time it takes to unload and load the cars.

6.1.1 Simulation configuration

To configure the simulation of the vessel the required device instances are created as described in section 5.1.2. Some manual work is needed to go through the PLC process image and copying the variable names which the simulation uses for communicating with the PLC. The devices dictionary consists of 50 uncontrolled bistables, six temperature measurements, four prime movers, eight motor inverters, two hotel inverters, and two throttle levers. Two DC links and six electric machines are needed to simulate the power production and consumption.

The bistables represent digital inputs such as fuse and isolator statuses, 24 V power and ethernet redundancy statuses, and other signals that are active in normal situation. At simulation start the signals, which are initialized as false in the PLC program, are written true to establish the normal system state. The temperatures of the propulsion motor bearings and hotel grid filter are measured by the PLC and are included in the simulated devices to be able to test their indication on the HMI and the reaction to too high temperatures. The throttle lever input analog inputs are used as a backup in case the Modbus communication between the thruster system and ECS fails.

6.1.2 Testing procedures

A series of tests was done with the simulated system. The tests are chosen such that the features of the simulation tool are tried out and verified to reflect the real system with sufficient depth. The starting and stopping of simulated equipment is checked by turning the vessel on and off. The simulated power generation and consumption is tested by running the propulsion system. Interactivity with the simulation is used to generate fault situations. To start testing the PLC programs are downloaded to the PLCs and the HMI is started. The simulation program is started at which point the fuse and isolator indicators are shown closed on the HMI and the system is ready for start-up.

Vessel start-up

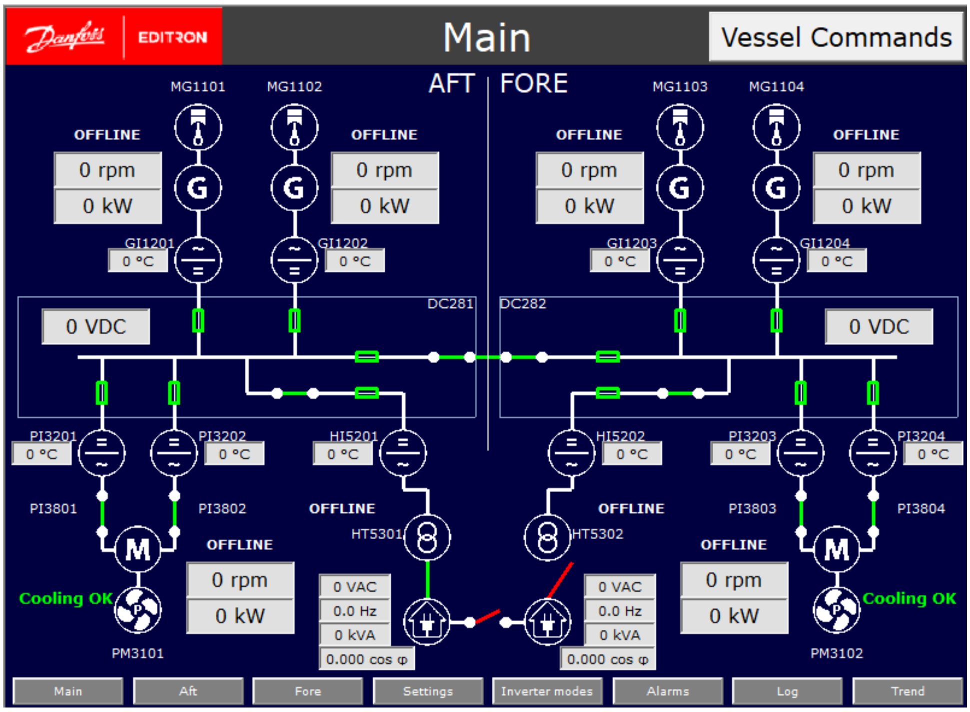
To start the vessel a button on the HMI is pressed. The ECS will then command a genset to start. When the engine reports it is ready to load, the generator inverter is started in voltage control mode and the DC link voltage rises to the reference. The two hotel inverters are also started. When the vessel has turned on, the propulsion is started when the start command and a speed reference is received via Modbus. Running the propulsion consumes power to which the generator inverter responds by increasing torque. The power system will settle in equilibrium where consumed power equals produced power.

Figure 6.1 shows the HMI screen before start-up and after the vessel has been started and the aft side propulsion is running. Fore side genset generates the power required by the hotel inverters and the propulsion. Power is transmitted from the fore side DC link to aft side via the bus tie. A slight oscillation in the DC link voltage and genset power was noted, caused by the Euler integration of the power when determining the current voltage. The oscillation can be mitigated by increasing the virtual capacitance to make the dynamics slower or by decreasing the time step. A good combination of $T_s = 0.01\,\mathrm{s}$ and $C = 2\,\mathrm{F}$ was found which provided stable simulation. The capacitance is far greater than in a real DC link but poses no problem as the fast dynamics of the real system are not considered.

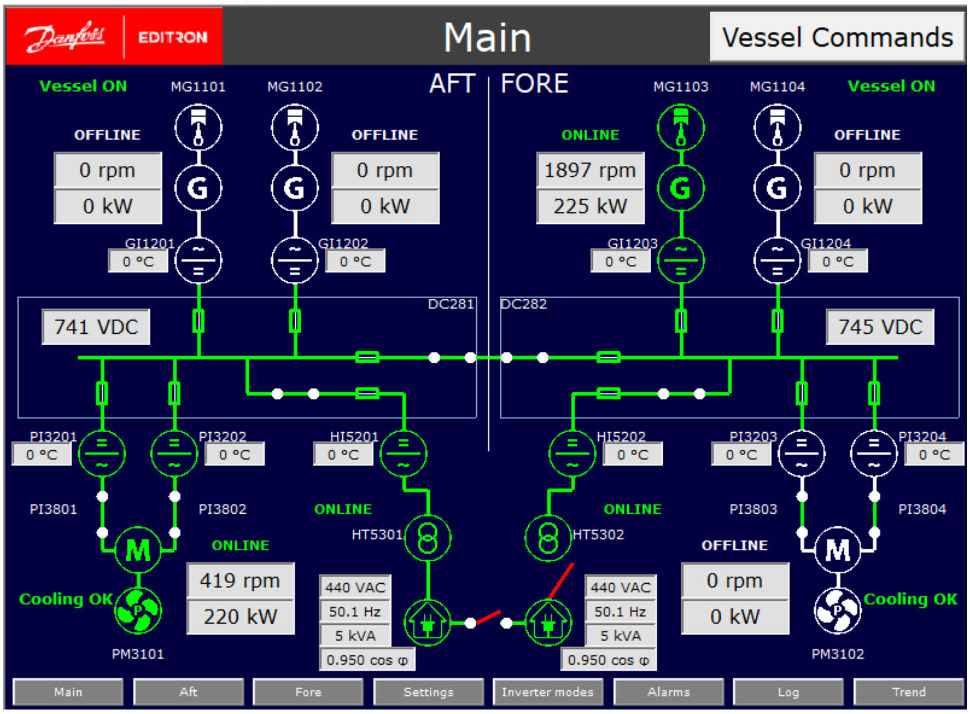
Genset parameters and load dependency

To further test the simulation of power dynamics it was tried whether the automatic starting and stopping of generators based on the load worked. Propulsion power was increased gradually until the next generator was started. Fig. 6.2 shows the HMI screen when all four gensets have been started in response to the high propulsion power. The power is shared equally among the gensets as it should. As the propulsion power was decreased the extra gensets were shut down as specified by the changeable load threshold and delay time. The simulated power system mimics the real system in such detail that the generator control logic can be tested.

Other features that can be tested using the simulation are the changing of generator priorities, minimum number of gensets, and the stability of the load dependency. When the priorities are changed the new genset should first start before the genset that it replaces is shut down. The load dependency should be stable meaning if the start and stop load thresholds are such that the stopping of a genset increases the load above the start threshold, the automatic shutdown should not occur to avoid oscillation where a genset is started and stopped periodically. The features can be tested successfully using the simulation. Peculiar



(a) Before start the DC link is not energized.



(b) One genset has been started and it produces power to keep the DC link energized. Aft side propulsion is running, and the hotel grids have been started.

Figure 6.1: HMI screens of the ferry before and after the simulated vessel has been started.

behavior, however, was noticed when the priorities or the minimum number was changed when the vessel is in off state; the new values are not updated until start is commanded, causing a genset that should not have started to start and immediately be replaced by the correct genset. The behavior was not noticed during commissioning or was deemed not to be an issue.

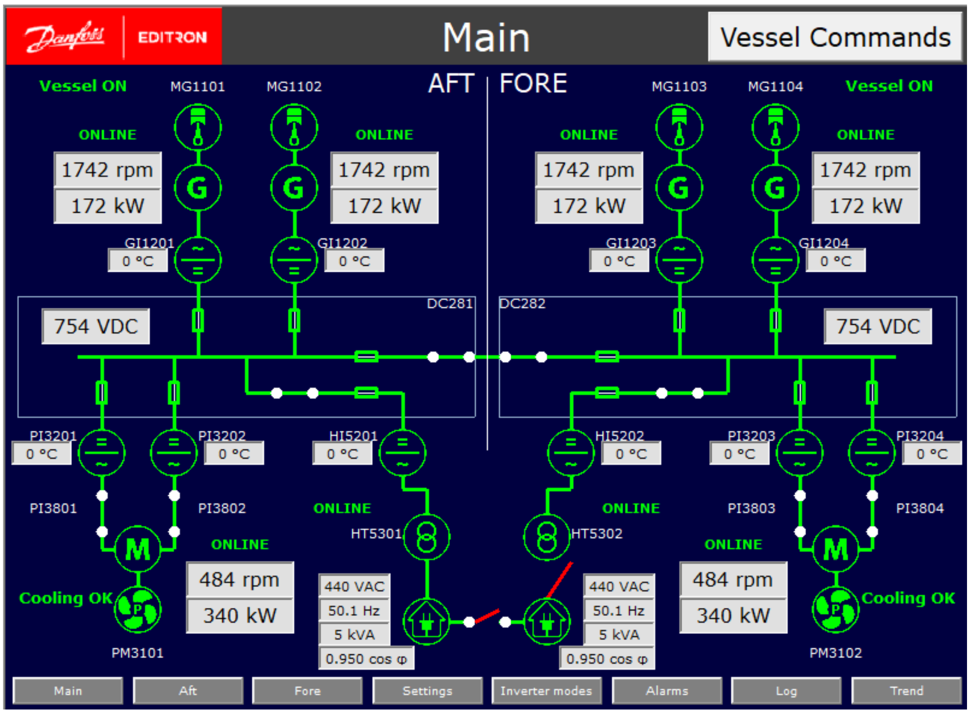


Figure 6.2: HMI screen of the ferry when the propulsion power has been increased and the PLC logic has automatically started the gensets. Simulated power is divided equally between the gensets.

Generator inverter trip

The simulation of inverter faults was tested by triggering the trip of a running generator inverter using the user command from the simulation tool. When the inverter tripped the PLC logic correctly started the next genset and shut down the faulted genset. When the reset fault command was given from the HMI the inverter fault status cleared but the genset did not start as it should. Upon inspecting the PLC code, it was noticed that when the inverter trips the generator will get flagged as tripped and will be disabled. When the reset command is given the generator is marked available and re-enabled, but because the time it takes for the reset command to reach the simulator, to be processed, and the status to be communicated back to the PLC is longer than the PLC cycle time, the generator will trip again immediately. The delay is present also in the real system because of the time delay of the CAN bus communication. The result is that the generator is disabled with no indication to the operator. A second reset command is needed after the inverter fault has cleared to return the genset to operation. To fix the problem the PLC logic would have to look at the inverter fault status and re-enable only when the fault has actually been cleared.

Engine stall and start fault

The response to the engine stalling or not starting can be easily tested with the user commands. When the engine stalled i.e. suddenly stopped or failed to start an alarm was correctly given and the next genset started. When the fault was reset the faulted engine was restarted.

Propulsion motor bearing overtemperature

The ability to simulate temperature measurements was tested by changing a propulsion motor bearing temperature. When the temperature reached the warning level an alarm was raised. Further increasing the temperature triggered a fault and the propulsion motor was stopped. No measurement validity monitoring feature is implemented in the vessel, thus when the breaking of the temperature sensor wire was simulated, overtemperature fault occurred because the measurement goes to the maximum value when there is too high a resistance measured.

6.1.3 Results from testing

With the use of the simulation the vessel state transitioned successfully from off to on. Without simulation the transition would have failed because of no feedback from the genset engine and inverter. Writing the required feedback signals manually would be laborious and prone to errors. The simulation of the propulsion drive works correctly as it is able to run and follows the speed reference and consumes power according to the set propeller curve. The simulated genset starts up and provides the power for the consumers. The simulation of the DC link performs as it should, the starting of the genset energizes it and the starting of propulsion and hotel inverters causes load which the genset matches resulting in a stable voltage.

The accuracy of the simulated power system is sufficient to test automated load dependent genset starting/stopping logic. An undesirable behavior was uncovered where if the minimum number of running gensets and genset priorities are changed when the vessel is in off state, the old values are used at start-up resulting in unnecessary starting and stopping of a genset.

Generating of fault situations at command is useful for testing response to faults and recovering from them. An error was found where resetting a tripped genset required two fault reset commands where after the first reset no indication of fault was given, but internally the fault state was still active. The simulation of temperature measurements is effective in testing the generation of alarms and automatic shut-down of equipment in case of overtemperature.

Overall, the simulation provides means to test the ECS functionality with a minimal setup of PLCs and a PC. Errors were found that had not been uncovered during the commissioning and by the acceptance tests, highlighting clearly the benefits of using simulation in testing of PLC software. In addition to improved software quality before delivery, the ability to reproduce faulty behavior, fix the cause, and verify that the fix is working using simulation has the potential of shortening warranty claim resolution times.

6.2 Parallel-hybrid vessel

The second project is a high-speed crew transfer vessel with water jet propulsion. Two diesel engines and jets are installed. Part of the shafts are electric machines that function either as generators in power take-off (PTO) mode or motors in power take-in (PTI) mode. In PTI mode the diesel engine is disconnected from the shaft with the use of a clutch and power is taken from a battery to run the water jet. An electric fan which creates an air cushion which reduces the vessel's draught and thus the drag force is also part of the system. Simulation was used to test the logic of changing between the operating modes, enabling the fan, the functionality of the HMI panel from which the system is controlled, and to perform other tests part of HAT program, before the commissioning took place.

6.2.1 Simulation configuration

To configure the simulation 26 uncontrolled bistables, two controlled bistables, six temperature measurements, two levers, three motor inverters, two prime movers, two hotel inverters, one DC/DC converter, three electric machines, and two DC links are needed. The uncontrolled bistables represent isolator and fuse statuses and are used to test that their state is indicated on the HMI. The clutches between the electric machines and the engines are simulated by the controlled bistables. The system is asymmetrical in the sense that port side has the first hybrid drive, battery, and the lift fan, whereas starboard side contains the second hybrid drive and an inverter that operates either as a hotel inverter or shore AFE. The operation of the engine is different in that the PLC does not control it. The engine speed follows the throttle lever signals and is started and stopped manually. A modified version of the prime mover class was created where user commands are used for starting and speed reference is adopted from a lever.

6.2.2 Testing procedures

The main focus in testing is the propulsion manager which handles mode changes between PTO, PTI, and off. The manager module was created from the ground up as a custom solution for the vessel. Other functionalities of interest are the starting and driving of the lift fan according to Modbus commands, the start-up of batteries and hotel drives, and other tests in the HAT program such as HMI indications and temperature measurements.

Hybrid system

The starboard and port side hybrid systems are commanded individually to either off, PTO or PTI from the HMI panel. Starting of the hybrid system in PTO mode involves engaging the clutch when the engine is idle and starting the inverter in voltage control mode. Torque limit is ramped up gradually to avoid a sudden load to the engine. In PTO mode the electric machine is a shaft generator that extracts power from the engine shaft to power the hotel grid and charge batteries. Generation power limit follows the speed with a configurable curve. To change from PTO to PTI mode the battery has to be online or the other side in PTO mode. The generation torque limit is ramped down, and the clutch is disengaged when the engine is at idle speed. When the clutch is disengaged the inverter is switched to speed control mode. The throttle lever position controls the speed reference. Going back to PTO mode involves waiting for the engine to start and run at idle to engage the clutch and change the inverter to voltage control. Timeouts prevent the transitions from getting stuck if the engine is not started or the clutch does not respond to its command.

Changing between the modes was tested by running the simulation and pressing the control buttons in the HMI display. Mode change timeouts were also verified to function. Figure 6.3 shows the state of the simulated system as indicated by the HMI when all the subsystems, except the battery, have been turned on. Figure 6.4 shows detailed views of the hybrid systems. To test the response to a low battery state-of-charge a separate BMS simulator was used which writes battery related info to the PLC via Modbus.

Using the simulations, programming errors were noticed in the PLC logic. When the propulsion manager had been driven to fault state by tripping the inverter, the reset of the fault situation did not work. A misplaced statement marking the end of a case-structure caused some statements to not execute. Also, a portion of code that should have been executed only once when entering the fault state was run continuously. The result of the programming errors was that the state machine got stuck. Another issue found was that

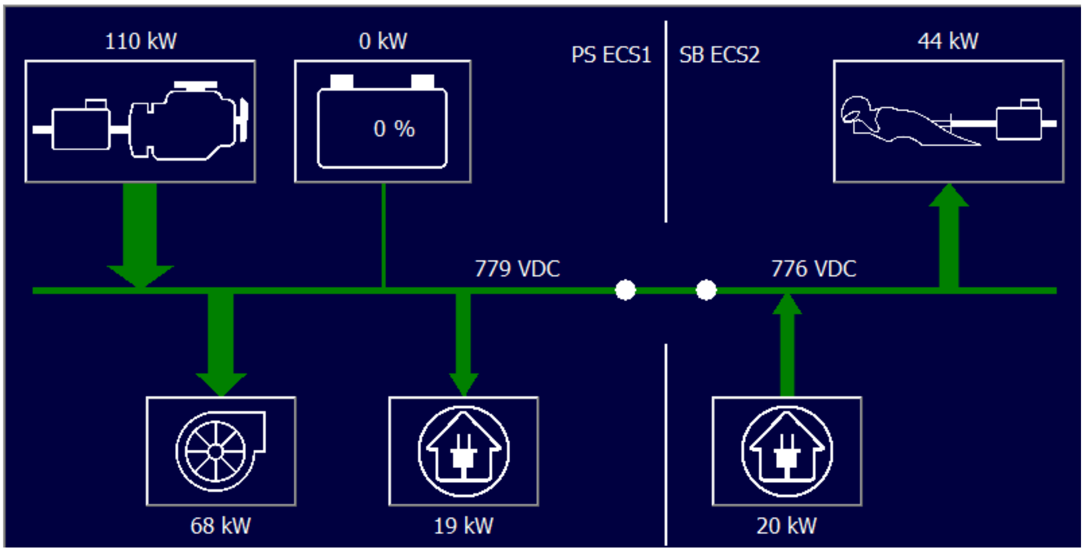


Figure 6.3: Overview of the simulated power system. Port side hybrid drive is in PTO mode whereas starboard side is in PTI mode. Lift fan is running, and the port side hotel inverter powers the hotel grid. Starboard side hotel inverter is operating as an AFE and is powering the DC link from shore power.

the mode change from PTI to PTO did not time out if the engine was not started, caused by a missing negation of a boolean variable.

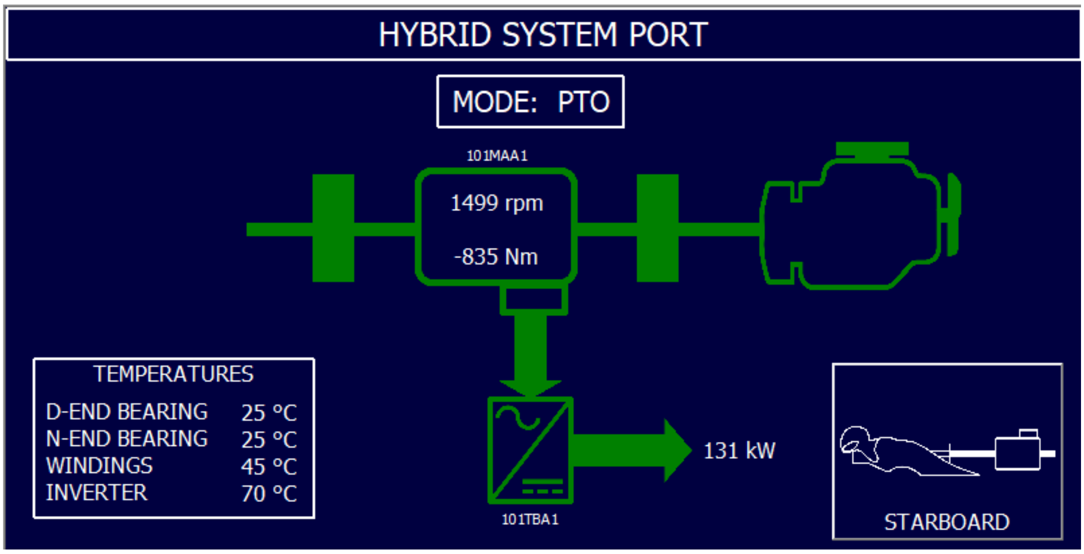
HMI

Instead of writing every signal that the HMI displays manually to test it, the simulation makes the testing of the HMI more straightforward. When the system responds to the commands given from the HMI, the panels indications can be tested at the same time the PLC logic is tested. In Fig. 6.3 the power flow to and from the subsystems is shown with arrows. The arrow indicates the direction of power and the width of the arrow is relative to the magnitude of power. By changing the simulated system operating point by e.g. increasing throttle and changing operating modes the HMI mimics can quickly be verified. The more detailed views of the hybrid systems in Fig. 6.4 also have many dynamic elements that need to be tested. When the engine is running, its icon is green, which can be tested by using the user command to start the engine. By changing modes between PTO and PTI the indication for the clutch status, machine speed, power, and torque are verified. The temperature measurements can be changed in the simulation with user commands.

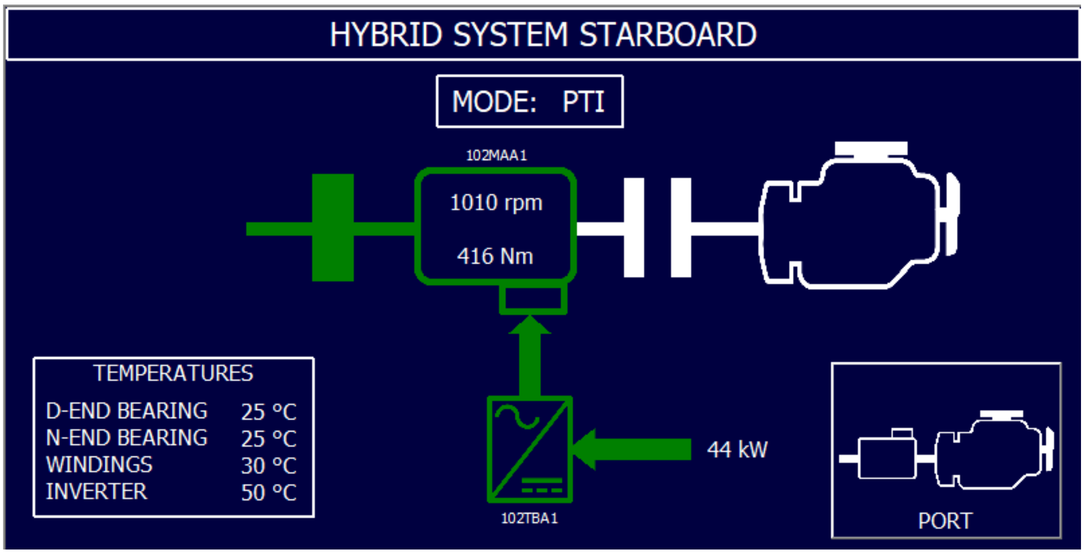
During testing, several errors were noticed in the HMI panel. As the linking of variables to the graphical elements is done by hand, the process is prone to human error. Errors such as reading the wrong bit of a word, or mixing up port and starboard side variables caused errors that were quickly noticed and fixed when the HMI panel, thanks to simulating the system, could be used just as with the real system.

6.2.3 Results from testing

The testing of the propulsion manager successfully revealed errors that were fixed before commissioning. Simulation allowed the easy reproduction of the erroneous behavior, helping narrow the location of the error in the PLC code. As a consequence, no time was spent debugging the propulsion manager under an already tight schedule during the commissioning. Errors in the HMI panel indications were also discovered, caused by errors such as mixing



(a) Hybrid drive in PTO mode. Power is taken from the propulsion shaft using the electric machine. Direction of torque is opposite to speed meaning the machine is working as a generator.



(b) Hybrid drive in PTI mode. Engine is shut down and the clutch is disengaged. The electric machine powers the propulsion shaft.

Figure 6.4: HMI pages showing the status of the port and starboard side hybrid systems. By simulating the system, the different operating modes and changing between them can be tested before commissioning.

up starboard and port side system variables. Problems in the PLC code discovered during commissioning were related to working with a third-party battery management system which had not been used before. Other changes done to PLC logic were related to system details which were not known before commissioning.

Overall, the use of the simulation sped up the software commissioning process; an estimated one day of working with the PLC code during commissioning was saved by testing beforehand. Although no numbers are presented for the effect on commissioning costs, the moving of the time required to find and fix the errors from busy commissioning days involving overtime to less busy days in the development phase that are paid with salary has a positive impact on costs. The testing is also time-effective; configuring the simulation for a new vessel is fast because the hardware used between vessels is similar and the configuration can mostly be copied over from a previous one.

6.3 Future work

A possible improvement to the simulation tool is the reading and writing of PLC data using structures. The configuration of the simulation would be quicker and simpler; e.g. the configuration of inverter simulation would require only the name of the structure containing the inputs and outputs on the PLC, instead of having to enter each variable separately.

An useful additional feature would be the ability to automate testing. A series of actions could be scripted to easily repeat tests after changes have been done to the software. Instead of having to enter user commands to the simulation, the script would perform them automatically. For example, the simulated throttle could automatically slowly ramp up and down to verify that generators start and stop automatically.

Extending beyond the scope of this thesis, modeling and simulation could prove beneficial to the development of better control software and new features. Detailed models capturing the realistic behavior of the power system could be used to e.g. optimize how batteries are used alongside engines. A different approach than what is used in this thesis should then be employed. Instead of a real-time control system and a simulation models executed in real time, the control software would also be modeled and the models simulated in non-real-time to make it possible to use more detailed models and computationally heavier numerical solvers.

7 CONCLUSION

In this thesis a tool for simulating field devices of a marine power system was developed. The tool is a Python program running on a general-purpose computer that reads the outputs of a PLC, executes simulations, and writes the PLC inputs so that the PLC receives feedback on its actions. As a result, the PLC software can be tested before the commissioning phase, when the field devices, such as diesel engines and power electronic converters, are not available.

The power systems of marine vessels were introduced and the components and subsystems specific to the Editron system were presented. The structure of the ECS software was also shown. The functionality and programming of PLCs were discussed. Options on how to implement the simulation requirements were given. The implementation of the requirements and the contents of the simulation program were described.

Two real-life vessel projects were used to obtain the results. First, the functionality of the simulation tool was verified by simulating a vessel which was already commissioned previously. The control software could be tested successfully using the simulations. In addition, errors in the control software that had not been discovered during the commissioning and acceptance testing were found, which demonstrated that quality of the software can be increased by testing with the simulation tool.

In the second project the simulation tool was used as intended; to test before the commissioning phase. The testing focused in the parallel-hybrid propulsion system and the HMI. Errors were found and fixed which lead to time savings during the commissioning. The time savings also had an impact in the related costs. Because of employing the simulation tool in only one project during the making of this thesis, the collected experiences are still limited. Future projects will tell more of the benefits.

Concluding, the requirements presented for the simulation tool were fulfilled with the selected approach of using Python programming language and Beckhoff ADS for communication between the simulation and PLC. Based on the collected results, the goals presented in the introduction chapter were completed.

REFERENCES

[1] B. R. Mehta and Y. J. Reddy. *Industrial Process Automation Systems: Design and Implementation*. Elsevier, 2015. ISBN: 978-0-12-800939-0.

[2] *EDITRON MARINE*. URL: `https://www.danfoss.com/en/about-danfoss/our-businesses/power-solutions/danfoss-editron/editron-marine/` (visited on 01/2021).

[3] D. Woodyard. *Pounder's Marine Diesel Engines and Gas Turbines*. 9th ed. Butterworth-Heinemann, 2009. ISBN: 978-0-7506-8984-7.

[4] F. Baldi et al. "Energy and exergy analysis of a cruise ship". In: *Energies* 11.10 (2018), p. 2508.

[5] J. Carlton. *Marine Propellers and Propulsion*. 3rd ed. Butterworth-Heinemann, 2012. ISBN: 978-0-08-097123-0.

[6] G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, 2000. ISBN: 9780521663960.

[7] J. P. Ghose and R. P. Gokarn. *Basic Ship Propulsion*. KW Publishers Pvt Ltd, 2004. ISBN: 81-7764-606-0.

[8] K. Mollenhauer and H. Tschöke. *Handbook of Diesel Engines*. Springer-Verlag Berlin Heidelberg, 2010. ISBN: 978-3-540-89082-9.

[9] A. Molland, S. Turnock, and D. Hudson. *Ship Resistance and Propulsion*. Cambridge University Press, 2017. ISBN: 9781107142060.

[10] J. Babicz. *Wärtsilä Encyclopedia of Ship Technology*. Wärtsilä Corporation, 2015. ISBN: 978-952-93-5535-8.

[11] J. Fleckenstein. *Three-phase Electrical Power*. CRC Press, 2016. ISBN: 9781351821346.

[12] S. C. Misra. *Design Principles of Ships and Marine Structures*. CRC Press, 2016. ISBN: 978-1-4822-5447-1.

[13] M. R. Patel. *Shipboard Propulsion, Power Electronics, and Ocean Energy*. Taylor & Francis, 2012. ISBN: 978-1-4398-8851-3.

[14] R. Prenc, A. Cuculić, and I. Baumgartner. "Advantages of using a DC power system on board ship". In: *Journal of Maritime & Transportation Science* 52 (Dec. 2016), pp. 83–97. DOI: `10.18048/2016.52.05`.

[15] O. Alnes, S. Eriksen, and B. Vartdal. "Battery-Powered Ships: A Class Society Perspective". In: *IEEE Electrification Magazine* 5.3 (2017), pp. 10–21.

[16] A. Kortsari et al. *Evaluation report of the E-Ferry*. Tech. rep. E-ferry project, 2020.

[17] S. Chapman. *Electric Machinery Fundamentals*. McGraw-Hill, 2005. ISBN: 0-07-246523-9.

[18] J. Pyrhönen, V. Hrabovcová, and S. R. Semken. *Electrical Machine Drives Control: An Introduction*. John Wiley & Sons Ltd, 2016. DOI: `10.1002/9781119260479`.

[19] J. Pyrhönen, T. Jokinen, and V. Hrabovcová. *Design of Rotating Electrical Machines*. John Wiley & Sons Ltd, 2013. DOI: `10.1002/9781118701591`.

[20] T. J. E. Miller. *Brushless Permanent-Magnet and Reluctance Motor Drives*. Oxford University Press, USA, 1989. ISBN: 0-19-859369-4.

[21] C. J. O'Rourke et al. "A Geometric Interpretation of Reference Frames and Transformations: dq0, Clarke, and Park". In: *IEEE Transactions on Energy Conversion* 34.4 (2019), pp. 2070–2083.

[22] *Liquid cooled heavy duty converter*. EC-C1200-450. Danfoss. Apr. 2020.

[23] H. W. van der Broeck, H.-C. Skudelny, and G. V. Stanke. "Analysis and realization of a pulsewidth modulator based on voltage space vectors". In: *IEEE Transactions on Industry Applications* 24.1 (1988), pp. 142–150.

[24] R. A. C. Terry Borden. *Technician's Guide to Programmable Controllers*. 6th. Cengage Learning, 2012. ISBN: 9781111544096.

[25] K. C. Wang. *Embedded and Real-Time Operating Systems*. 1st. Springer Publishing Company, Incorporated, 2017. ISBN: 3319515160.

[26] R. S. Pressman. *Software Engineering - A Practitioner's Approach*. 5th ed. McGraw-Hill, 2001. ISBN: 9780073655789.

[27] G. J. Myers and C. Sandler. *The Art of Software Testing*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2004. ISBN: 0471469122.

[28] C. G. Lee and S. C. Park. "Survey on the virtual commissioning of manufacturing systems". In: *Journal of Computational Design and Engineering* 1.3 (2014), pp. 213–222.

[29] P. Hoffmann et al. "Virtual commissioning of manufacturing systems a review and new approaches for simplification." In: *ECMS*. 2010, pp. 175–181.

[30] G. Reinhart and G. Wünsch. "Economic application of virtual commissioning to mechatronic production systems". In: *Production Engineering* 1 (Jan. 2007), pp. 371–379. DOI: 10.1007/s11740-007-0066-0.

[31] R. Bettendorf. "Winder software testing with real-time dynamic Simulation". In: *IEEE Transactions on Industrial Electronics* 52.2 (2005), pp. 489–498. DOI: 10.1109/TIE.2005.844228.

[32] H. Carlsson et al. "Methods for Reliable Simulation-Based PLC Code Verification". In: *IEEE Transactions on Industrial Informatics* 8.2 (2012), pp. 267–278. DOI: 10.1109/TII.2011.2182653.

[33] *Rules and Regulations for the Classification of Ships*. Lloyd's Register, 2020.

[34] S. Lehmann. *pyads*. URL: https://github.com/stlehmann/pyads (visited on 12/2020).

[35] G. Franklin, J. Powell, and M. Workman. *Digital Control of Dynamic Systems*. Addison-Wesley, 1998. ISBN: 9780201820546.