Modelica driven development of the thermal management control system for a zero emission yacht

A. van Dam[1] B. van Groos[1] C. von Ohlen[2] F. Theel[2] J. Brunnemann[2] J. Eiden[2]

[1]Foundation ZERO, Amsterdam, Netherlands, {alje.vandam, boudewijn.vangroos}@foundationzero.org
[2]XRG Simulation GmbH, Hamburg, Germany {vonohlen, theel, brunnemann, eiden}@xrg-simulation.de

Abstract

This paper describes the use of a MODELICA system model to support the development of a heat management control system for a fossil-free sailing yacht. Due to tight project timelines, the control system was developed and tested virtually, avoiding delays associated with waiting for the physical system to become available. The system model covers key functions such as heat recovery and heat dumping, enabling automated testing of various operational scenarios. This approach not only accelerates development but also reveals early insights into interactions between the control logic and system dynamics. The model is designed to be seamlessly replaced by the real system once it is built. Future comparisons between simulated and real-world performance will guide refinements to improve model accuracy and support model-based tuning of the control system.

Keywords: HVAC, control design, FMU, thermal management, zero emission yacht, ship

1 Introduction

In the year 2022 about 2% of the worldwide greenhouse gas was emitted by the shipping industry (International Energy Agency 2023). In absolute values this means an emission of 858 billion tons of greenhouse gas (Clarke et al. 2023). A transition to cleaner technologies is necessary to reduce those values. FOUNDATION ZERO is a relatively new organization which aims to share technical innovations to promote sustainable technologies in various sectors. It is active in the maritime industry, having previously provided its PowerHub to provide the sail team BCN in the 2024 America's Cup with fossil free energy (Foundation Zero 2024b). The source code of which has since been open sourced (Foundation Zero 2025b). One of the bigger items being covered by FOUNDATION ZERO is PROJECT ZERO, which is attacking the problem of shipping emissions with traditional propulsion with sails and the comfort of today. It is developing a 69-meter sailing yacht that aims to be the first of its kind to operate entirely free of fossil fuels. To achieve this, the project relies on the development of advanced technologies that balance energy generation, storage, and demand on board. The project also aims to share these innovations on an open-source basis, with the goal of accelerating the maritime industry's energy transition (Leslie-Miller and Someren 2022). Key systems enabling the yacht's fossil-free operation include a propulsion system that is capable of hydrogeneration by reversing the propellers, generating up to 750 kW of power that can be stored in a 5 MWh battery bank. Top-of-the-line insulation (Foundation Zero 2024c) and a heating, ventilation and air conditioning system (HVAC) that includes adiabatic cooling and natural ventilation through the mast (Foundation Zero 2025a) reduce the energy needed to control the climate on board, which typically accounts for up to 50 % of a yacht's energy use. Additional energy savings are achieved through the "Thermal Harvesting and Recovery System" (THRS), which is the focus of this paper. A simplified diagram is shown in Figure 1.

The THRS reduces electrical energy consumption by leveraging thermal energy (heat) alongside electrical energy. It harvests heat using custom-developed PVT panels capable of generating both electrical (PV) and thermal (T) energy, optimizing the limited surface area available on board (Foundation Zero 2024a). Additionally, it recovers waste heat from the propulsion system and electrical components, providing an extra source of thermal energy while reducing cooling demand, as this heat would otherwise accumulate inside the yacht. Thermal energy is stored using a phase change material (PCM) that acts as a thermal battery, providing heat to the domestic hot water supply and an adsorption chiller. The adsorption chiller supplies cool water to wall and ceiling panels throughout the yacht, reducing the load on the electrically driven HVAC system.

Recovering heat that would otherwise be discarded via cooling water and reusing it in place of electrical energy, for example for domestic hot water use, can offer significant energy savings. However, it also introduces the challenge of efficiently managing thermal harvesting, storage, and consumption while maintaining safe and stable operation of critical subsystems. This requires an intelligent control system with the ability to respond other systems on board, integrating inputs such as trip itineraries, weather forecasts, and consumption profiles,

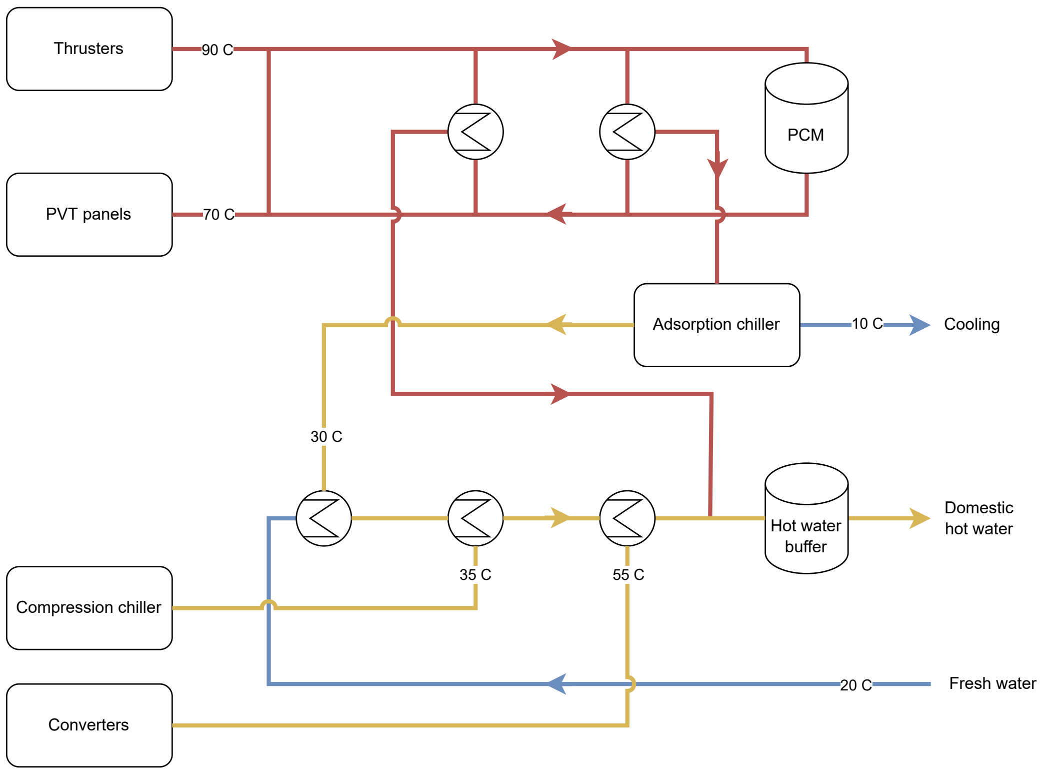

Figure 1. Simplified diagram of the thermal harvesting and recovery system

while also providing feedback to crew—for example, recommending optimal times for energy-intensive activities like laundry.

To manage the complexity of the system, it is divided into functional submodules, each with its own control logic to deliver basic functionality. The THRS system submodules are the high temperature circuit, the hot water circulation circuit, the buffer tank boost circuit, the buffer tank fill circuit, the chiller circuit, the booster circuit and two low-medium temperature circuits. These submodules operate under distinct control modes and setpoints, determining, for example, whether heat from the propulsion system should be harvested or rejected via seawater cooling—and at what temperatures. The high-over control system can then coordinate the modes and setpoints across submodules based on current or forecasted onboard conditions. The control modes of the submodules each have a flow configuration. That is achieved through valve and pump settings and includes basic control loops to maintain temperature and flow targets.

A more practical challenge is the tight timeline of the construction and commissioning of the yacht. In addition to these reasons, there are issues of availability and high costs, meaning that Hardware-in-the-Loop (HiL) or even proper testing of the controller in the real system are out of the question. This has motivated the creation of a dynamic model of the physical system, which facilitates co-development of the control logic alongside the physical system, effective troubleshooting before the system is installed, and isolation of individual subsystems for testing. MODELICA is used for the Software-in-the-Loop (SiL) approach to model the plant model. The controller is implemented in Python, as the goal is to use the implemented Python code in the real system as well. In addition, the entire project focuses on open access, so Python is preferred over conventional tools, which are mostly commercial. Further details of how the implementation of the control logic in Python is pointed out in Section 3. Once validated, the control logic can seamlessly transition from simulation model to real-world operation by replacing the dynamic model with the yacht's actual input/output interfaces during commissioning (see Figure 2). When replacing the plant model with the real system, latency effects are expected to occur, but these can be neglected due to the high capacities and inertia of the THRS. This paper demonstrates this approach for a submodules component of the THRS that recovers heat from the yacht's propulsion system.

The paper is structured as follows: In Section 2 a functional description of the thermal harvesting and recovery system is given. Section 3 describes how MODELICA is integrated in the project architecture and explains the interface between the developed MODELICA model and the rest of the tool chain. To demonstrate the viability of this approach, the focus lies on a single component of a submodule that recovers heat from the thrusters. A detailed description of the thrusters component and its modelling in MODELICA is given in Section 4. Section 5 presents simulation results of the MODELICA model coupled with the developed Python control. Finally, a summary and outlook are given in Section 6.

2 Description of the Thermal Harvesting and Recovery System

The THRS (see Figure 1) comprises components that produce, consume, or store thermal energy. When sailing—either in propulsion or hydrogeneration mode—the thrusters generate heat and require cooling, providing a potential heat source of up to 13.3 kW at 90 °C that can be recovered by the system. PVT panels contribute additional thermal input that is also available when at anchor, typically harvesting at around 70 °C during peak sunlight hours. Other onboard sources include the waste heat from the compression chiller and electrical converters, which offer a more constant source of energy, but at lower temperatures.

Thermal energy is consumed by domestic hot water demands at 48 °C (e.g. showers, laundry, and dishwashing) and by an adsorption chiller, which requires hot water between 50 °C and 95 °C to produce chilled water for onboard cooling. To balance fluctuating availability and demand throughout the day, heat can be stored in both a phase change material (PCM) storage unit and a hot water buffer tank. To manage thermal energy efficiently across different temperature levels, the system is divided into a high-temperature and a medium-temperature circuit. This split enables harvesting energy at both high and low temperatures - for example allowing use of the waste heat from the adsorption chiller to preheat water entering the medium-temperature loop. The high-temperature circuit primarily supplies the adsorption chiller and can supplement the medium-temperature circuit as needed. Figure 1 presents a simplified schematic of the system. For clarity, elements such as valves, pumps, and seawater coolers are omitted. In practice, each critical component includes a seawater-based backup cooler to ensure safe operation when thermal energy cannot be stored or utilized.

As an illustrative example, the thrusters component is highlighted, which is responsible for cooling the thrusters during operation and, when possible, feeding the recovered heat into the rest of the system. This component is described in detail in Section 4, which also presents its implementation in MODELICA.

3 Using MODELICA to design a control system

The main objective of the project underlying this paper is to develop a dynamic model of the THRS that can be used as a development environment for the control logic. A scheme of the according software architecture is shown in Figure 2: The box at the top shows the Python control module. The Module is implemented with pytransitions which is a library for finite state machines (Alexander Neumann 2025). It takes as input sensor values and computes the desired control outputs based on the control logic as output. Sensor values include temperature and flow sensor readings, but also states of other systems on board, such as the propulsion system. On the receiving end of the control, shown at the bottom box, one finds either the simulation model or the ship's real alarming, monitoring and control system (AMCS). The AMCS provides an interface for critical systems on board, such as power management and propulsion control. It handles most of the ship's sensors data and control signals and is connected to the physical components of the THRS through a programmable logic controller (PLC). The interface between the control logic and the simulation or the AMCS is realized with the message queuing telemetry transport (MQTT) protocol (Banks et al. 2019).

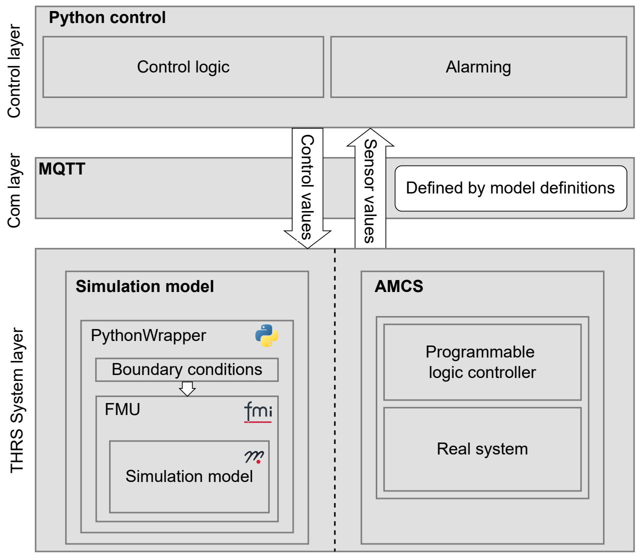

Figure 2. Architecture - Dividing into control layer, communication layer and THRS system layer. Simulation model and AMCS are interchangeable.

In this setup, the THRS IO and control are completely separated, and simulation and AMCS are interchangeable. This means that the control logic can be developed and tested with the simulation model, and then seamlessly transitioned to the AMCS without changes to the control code. Other than changing the MQTT broker to which the control code connects there is no meaningful difference between simulation and the actual system. Therefor potential difficulties which might ordinarily occur when going from model to compiled code don't apply. One can expect latency effects between the computer running the control code and the AMCS. It is assumed that those effects are small compared to thermal capacities and inertia of the THRS.

3.1 Setup of the THRS system model

The THRS components are modeled with the MODELICA library HVAC (XRG Simulation GmbH 2025) in combination with the MODELICA STANDARD LIBRARY (Modelica Association 2023). The HVAC library is designed for fast simulations of thermo-hydraulic and air distribution systems. It provides models for cooling towers, heaters, turbo machines, heat exchangers, heat pumps, chillers, humidifiers, water separators, storages, sensors, photovoltaic panels, thermal collectors, fittings, pipes, valves, sinks, sources and other HVAC relevant components (see example icons in Figure 3). To model the THRS a selection of the above-mentioned models are used. In addition to that, a model for PCM based on Brunnemann (2020) is integrated to the system. A detailed description of modeling the entire THRS would be beyond the scope.

The model of the thruster component (see Figure 4) is embedded into an interface model in MODELICA. Real inputs, representing control values or boundary conditions, and outputs, representing sensor values, are defined in a way to match the real system. To prevent sudden changes or unexpected jumps of the input signals, a first order block is placed behind the inputs to smooth them before they are passed on to the thrusters component. To use the MODELICA model as the replacement of the

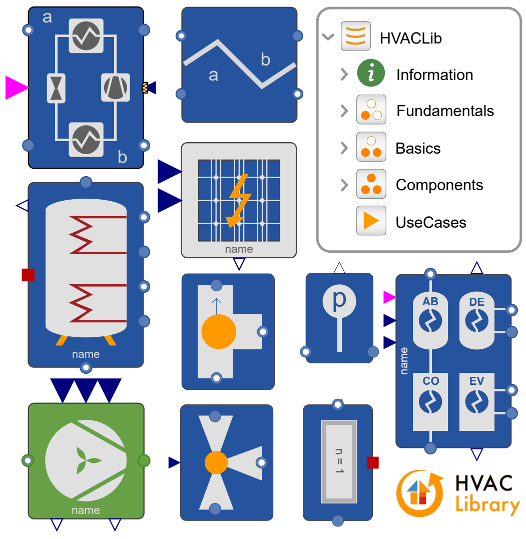

Figure 3. HVAC LIBRARY with example models for heat pump, buffer tank, fan, heat exchanger, photovoltaic panel, fitting, sensors and adsorption chiller

AMCS, it is compiled into a FUNCTIONAL MOCK-UP UNIT (FMU). A Python wrapper enables the FMU to communicate via MQTT with the control module (see Subsection 3.2). The development and compiling of the model is done in DYMOLA 2025X. Co-Simulation with the solver Cvode is selected. The FMI version chosen is 2.0. To enable recompilation of the FMU on another operating system, the source code export is included. It is recommended to deactivate the tools flag "evaluate". Another way to enable the model to communicate with the control module via MQTT is to use the MODELICA DEVICE DRIVERS LIBRARY (Thiele et al. 2017). Since there are more options to integrate a FMU into any workflow the compilation in a FMU is the preferred way. This opens greater flexibility in the event of changes to the process.

3.2 Python integration

The FMU is executed using FMPy (Dassault Systèmes 2025), enabling compilation and execution of the FMU in a Python environment. From here, a Python wrapper is constructed that provides the MQTT interface to the FMU. The wrapper is responsible for reading the control values from the MQTT broker and writing them to the FMU inputs. It also reads the sensor values from the FMU outputs and writes them to the MQTT broker. The data schema for all these values is defined using the Pydantic Python package (Colvin et al. 2025). Pydantic is a data validation and serialization library for Python, which allows for the definition of data models with type annotations. This enables automatic validation of the data being passed to and from the FMU and AMCS.

One of the key aspects of the setup is the ability to exchange the real system with the simulation model. This means that the control code is fully unaware of that difference. Specifically, the interface of the simulation (the sensor values it sends, and the control values it receives) should be identical to those of the real-world system (i.e. the AMCS). This is done as follows:

- Model definitions: Pydantic models are defined that describing sensor values and control values. The models follow the structure of the MQTT messages as defined by the AMCS.

- FMU mapping: The Pydantic models are translated to the FMU's input and output variables. This is done by maintaining a strict naming convention for the FMU's variables.

- FMU boundary condition input: The system model requires some additional inputs that are not part of the control values (e.g. seawater temperature), which are injected in the simulation as boundary conditions. These are also defined by Pydantic models as either constants or time series.

- Unsimulated sensor values: any sensor values that are not simulated in the FMU are added to the simulation output (and must also be defined as simulation inputs). An example is the signals from the ship's propulsion control system indicating which thrusters are active.

With this setup there is a complete stand-in for the real-world system, abstracted behind a Python class which receives control values and output sensor values. This class can read and write MQTT messages and translate them to FMU inputs and outputs. The control loop reads the sensor values, enters those into the control logic and feeds the control values back into the simulation.

4 Thrusters Component in Modelica

This section describes in more detail the simulation model of the thrusters component within MODELICA (see Figure 4). Its main subcomponents can be identified as a circulating pump (1), the two thrusters, a forward thruster (4) and an aft thruster (8) and a seawater cooler (14). It also includes several control valves, fittings and different types of physical sensors ((3), (6), (7), (10), (16), (18), (19), (27), (28)). The medium, a 20 % glycol-water mixture, is assumed with constant fluid properties.

The module has three main operating modes, an idle mode, the heat recovery mode and the direct cooling mode, which are described in Section 5.

4.1 Circulation Pump

The flow in the thrusters component is provided by the circulation pump (1) (highlighted in light purple in Figure 4). For reasons of redundancy there are two parallel pumps which are represented by one circulation pump. The two identical pumps do not operate at the same time, so it is sufficient to use only one pump model with a built-in pump changeover sequence. The pump model is parametrised with one characteristic pump curve for a specific speed. The pump gets a relative speed as control parameter and operating points deviating from nominal values are approximated from characteristic curve with affinity laws.

4.2 Forward and Aft Thruster

As mentioned before, the thrusters of the yacht are generating heat during operation and must be cooled. Therefor the thruster model ((4), (8)) is simplified to transfer a certain heat load to the coolant and to have a linear pressure loss model, which is defined by nominal parameters. The heat load is given through an input and can be varied during the simulation. Forward and aft thruster are shown with a red frame in Figure 4.

4.3 Valves and fittings

There are three types of valves used in the thruster module, the switching valves ((11), (20)), the mixing valves ((13), (21), (22)) and the flow control valves ((5), (9)). The switching and mixing valves are three-port valves and are based on the same model. The flow control valves are two-port valves. Each valve is parametrised with a KV nominal value, provided by their data sheets. The opening for the two-port valves and the flow direction for the three-port valves are set as relative ratios and can be varied during the simulation. There is a leakage of 0.01 % to make the overall model more robust against numerical issues. The opening is implemented with a time delay behaviour.

The fittings can be divided into splits ((2), (25)) and joins ((12), (15), (17), (23), (24), (26) (29)). The joins are designed with a control volume-based approach and flows are ideal mixed. Splits can be used either with predefined split ratio or with a computed ratio provided by outer components which are generating mass flows.

4.4 Seawater cooler

The seawater cooler (14), highlighted in green in Figure 4, is implemented to guarantee the cooling of the two thrusters. The heat transfer is modelled as an eps-NTU approach (Hesselgreaves 2001). Nominal data from datasheets parameterize the model. A seawater medium with constant data at 30 °C is created for seawater side of the heat exchanger. A simple quadratic nominal pressure loss is implemented to model the relation between flow and pressure. The volume flow and temperature of the seawater are variables and can be set as inputs during the simulation.

5 Application Scenario

5.1 Thrusters control

Based on the sensor values coming from the simulation model, the control module determines control values to reach the desired setpoints. The control logic for the thrusters module covers several key functions:

- **Flow control:** The total flow is calculated as the sum of the setpoints for each individual thruster and is regulated by a PID controller that adjusts the pump speed accordingly. The flow is balanced between the thrusters using the flow control valves ((5), (9)). Each valve is controlled by a PID controller, with control values offset to ensure that one valve always remains fully open to avoid unnecessary counterpressure. The system also adapts to thruster activity—closing the flow control valve and reducing pump speed when only one of the two thrusters is active.

- **Mode switching:** As mentioned before, there are three operational modes : idle (no flow), heat recovery, and direct cooling. The control must support all modes. Each mode has distinct flow and temperature setpoints. Mode transitions are triggered by sys-

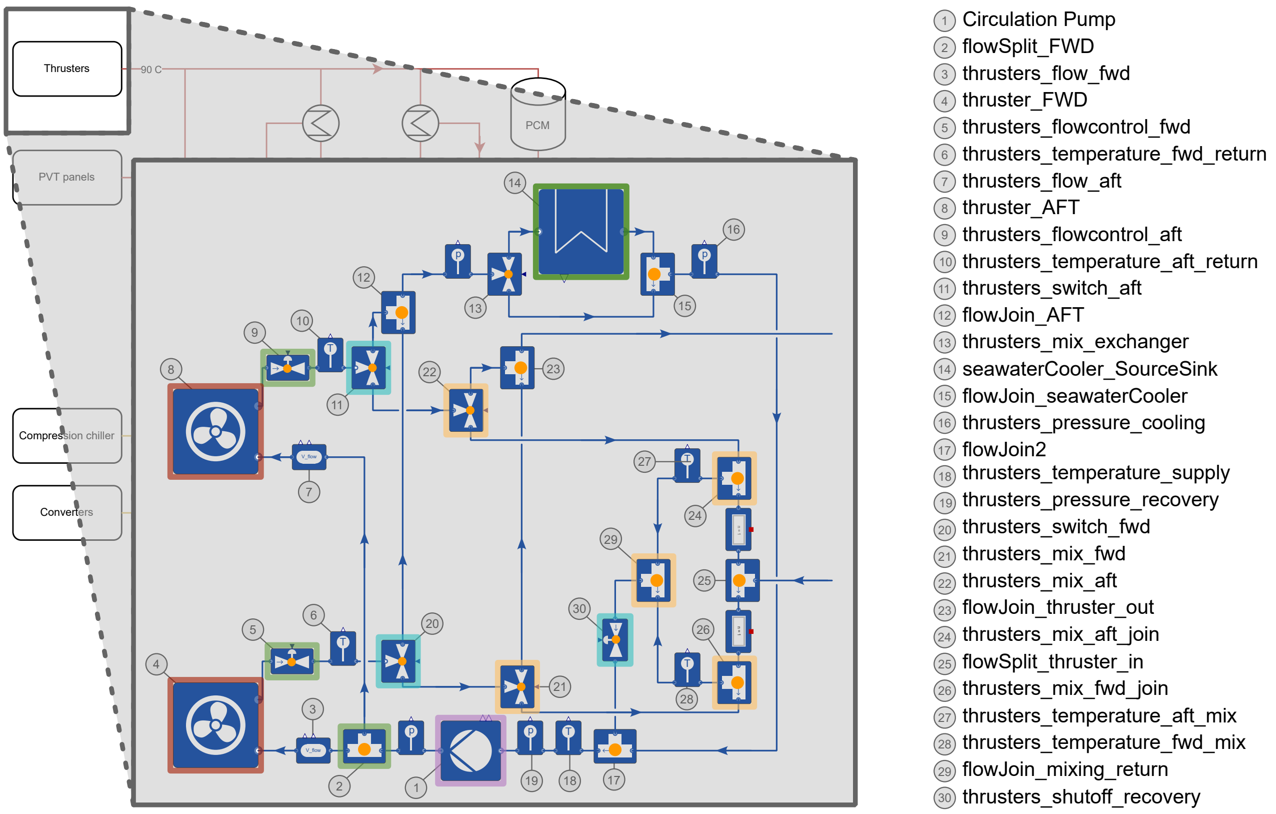
Thrusters
90 C
PCM
PVT panels
Compression chiller
Converters
1 Circulation Pump
2 flowSplit_FWD
3 thrusters_flow_fwd
4 thruster_FWD
5 thrusters_flowcontrol_fwd
6 thrusters_temperature_fwd_return
7 thrusters_flow_aft
8 thruster_AFT
9 thrusters_flowcontrol_aft
10 thrusters_temperature_aft_return
11 thrusters_switch_aft
12 flowJoin_AFT
13 thrusters_mix_exchanger
14 seawaterCooler_SourceSink
15 flowJoin_seawaterCooler
16 thrusters_pressure_cooling
17 flowJoin2
18 thrusters_temperature_supply
19 thrusters_pressure_recovery
20 thrusters_switch_fwd
21 thrusters_mix_fwd
22 thrusters_mix_aft
23 flowJoin_thruster_out
24 thrusters_mix_aft_join
25 flowSplit_thruster_in
26 thrusters_mix_fwd_join
27 thrusters_temperature_aft_mix
28 thrusters_temperature_fwd_mix
29 flowJoin_mixing_return
30 thrusters_shutoff_recovery

Figure 4. MODELICA model of the thrusters component with legend on the right side.

tem conditions — for example, engaging a thruster switches from idle to recovery, while an overheating alarm may switch from recovery to cooling. Overheating can occur when there is no thermal demand in the rest of the system, causing the inlet temperature to rise too high for safe thruster cooling. In such cases, excess heat must be dumped via the seawater cooler (14).

- **Cooling temperature control:** In direct cooling mode (see Figure 5a), the primary goal is to ensure safe thruster operation by rejecting heat directly, without recovering it for onboard use. It is triggered when the supply temperature of the thrusters passes a set threshold, to prevent overheating. The flow path in this mode is marked in orange. From the circulation pump (1) the flow starts its way through the forward (4) and aft thruster (8). The shutoff recovery control valve (30) is closed and the three-way valves after the forward thruster (22) and the aft thruster (11) are controlled to direct the flow to the flow join (12) where both streams are mixed. After passing the flow join the coolant reaches the three-way valve (13) in front of the seawater cooler (14). This valve is implemented to control the ratio between the flow through the seawater cooler (14) and its bypass flow. This is done by a PID controller to reach the desired setpoint temperature of return flow. From the seawater cooler (14) the coolant flows via a flow join (15) back to the circulation pump (1).

- **Recovery temperature control:** The heat recovery mode (see Figure 5b) is engaged when there is a demand for thermal energy (or available storage) in the rest of the system. The flow path in this mode is shown in green. From circulation pump to the three-port valves ((11), (20)) the flow takes the same pass as in direct cooling mode. Instead of the coolant flowing in the direction of the flow join (12) upstream of the seawater cooler, it heads into the two mixing three-port valves ((22), (21)) which are highlighted with a yellow frame. One part of the medium leaves the thrusters component via the flow join (23) into the high temperature circuit of the yacht. The other part of the medium is directed to two mixing flow joins ((24), (26)). Both are connected via the flow split (25) with supply flow from high temperature circuit into the thrusters component. After mixing both flows with the supply flow the water-glycol mixture is headed back to the circulation pump with the shutoff recovery control valve (30) which is opened in this mode. Regarding the control, the temperature of the return flow is controlled by a PID controller that adjusts the three-port mixing valves ((21), (22)) regulating the amount of coolant that recirculates past the thrusters to reach the desired return

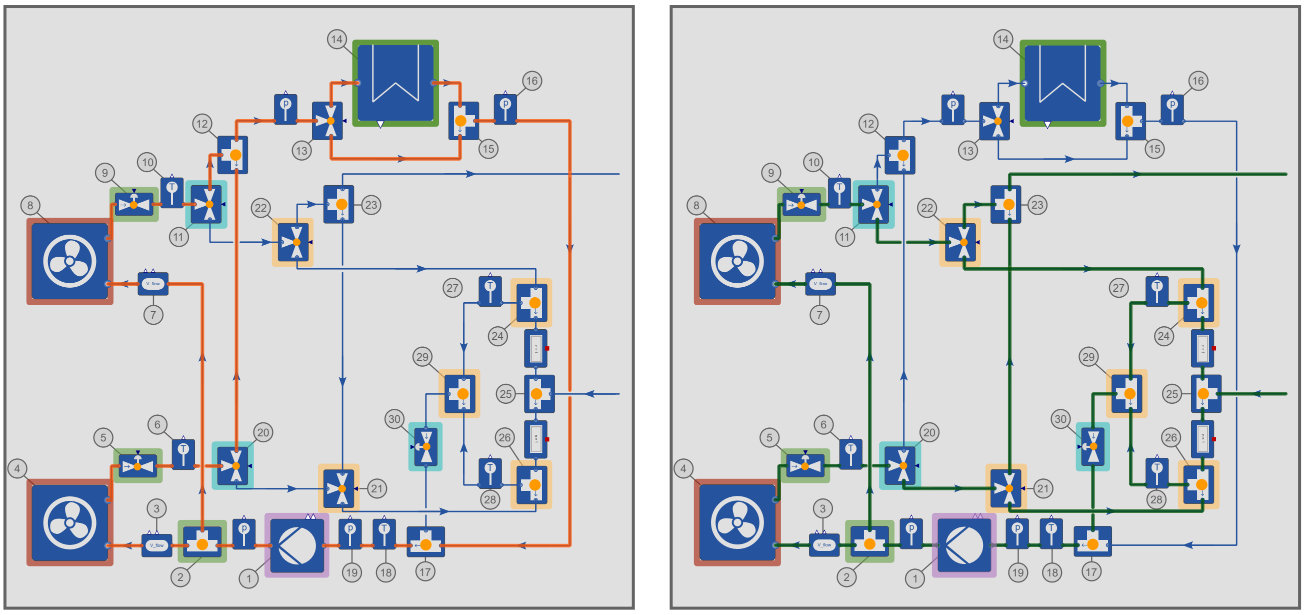
(a) Direct Cooling Model **(b)** Heat Recovery Model

Figure 5. Operating modes of the thrusters component

Cooling mix setpoint	40	°C
Recovery thruster flow	10	l/min
Cooling thruster flow	22	l/min
Maximum temperature	70	°C
Recovery temperature setpoint	60	°C

Table 1. Control setpoints

Aft thruster heat	4.3	kW
Fwd thruster heat	9	kW
Seawater supply temperature	32	°C
Seawater supply flow	20	l/min
Module supply temperature	50	°C

Table 2. Simulation parameters

temperature. This way, a minimum return temperature of the module is guaranteed (flow which is leaving the join (23)).

The control logic is demonstrated through a simulated operation of the thrusters module governed by a Python-based controller. Figures 6 and 7 present consecutive segments of a two-hour simulation run. Figure 6 illustrates the first hour, during which the system transitions from idle into heat recovery mode with both thrusters active. The simulation parameters and control setpoints are provided in Tables 1 and 2, respectively.

Each figure includes a top plot showing control signals sent by the module, consisting of valve setpoints and the pump duty point, which regulates flow depending on the current control mode (cooling or recovery). The bottom three plots display system variables—primarily sensor readings—except for the module supply temperature, which is predetermined as a simulation parameter.

While the top plots of both figures appear similar, they represent different data: the uppermost shows valve setpoints (desired positions) given by the control, whereas the second plot reflects actual valve positions that are read as sensor values. These diverge because physical valve actuation takes time (approximately 1 degree per second), resulting in about 90 seconds for valve to open or close.

5.2 Heat recovery

In the first five minutes of the simulation, the thrusters are off, the system is in idle mode, and there is not flow. The seawater return temperature starts at the default 30 °C but stabilizes to 32 °C, as specified in the parameters.

After five minutes, the thrusters are activated and trigger the control to go into recovery mode. The pump starts running and the flow control valves of each thruster are opened. The aft and forward thruster return temperatures rise due to flow past their respective heat sources (9 kW and 4.3 kW). Once the aft return temperature reaches the 60 °C recovery setpoint, the *aft mix* valve opens, allowing heated return flow to exit the module (*module return*) and supply the THRS. Simultaneously, the aft inflow begins drawing from the *module supply*, lowering the *mix aft* temperature and ultimately returning it to the supply temperature of 50 °C as the mix valves fully opens.

The forward thruster, generating roughly half the heat of the aft thruster, warms more slowly. When its return temperature eventually reaches the 60 °C setpoint, the *fwd mix* valve begins opening. Due to insufficient heating power, the PID controller causes the valve position to oscillate before settling near 0.25. This behavior

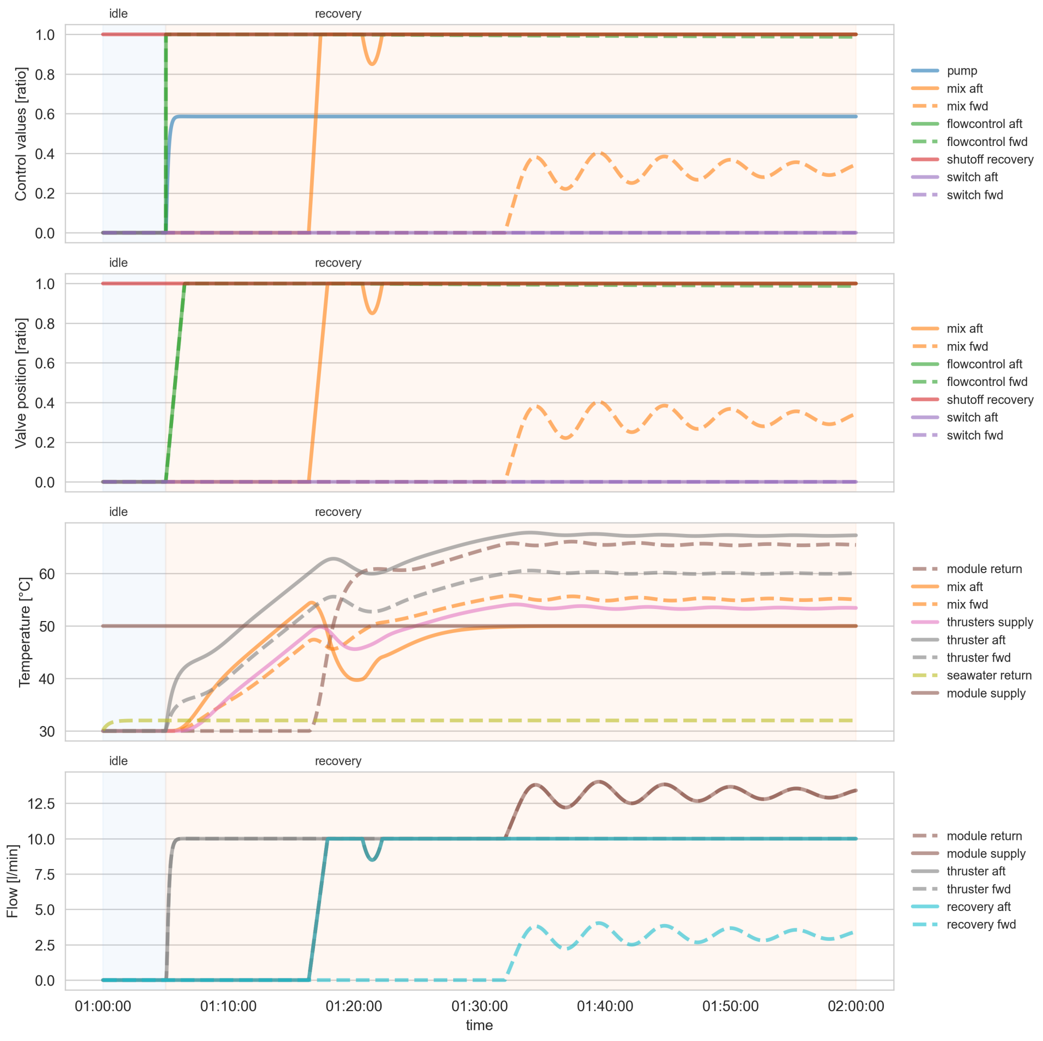

Figure 6. First hour of simulation in recovery mode, showing initialization and mixing control of the aft thruster.

indicates that some recirculation is required to maintain the desired temperature, and results in the *thrusters supply* temperature stabilizing slightly above the *module supply* temperature of 50 °C.

5.3 Mode Switching

Figure 7 depicts the second half of the simulation. At 02:00, the forward thruster is deactivated. In response, the controller shuts off flow to the forward thruster by closing *flow control aft*, and adjusts the *pump* duty to maintain the aft flow setpoint.

At 02:30, the *module supply* temperature is raised from 50 °C to 80 °C. This leads to the *thrusters supply* temperature to pass the 70 °C limit, triggering the cooling mode. This switches the valves *switch aft*, *switch fwd* and *shutoff recovery*. Note the difference between the control signal and the actual movement of the valves. In cooling mode, the *seawater return* temperature increases as heat is expelled through the heat exchanger, while the *thruster aft* temperature decreases as the control will maintain the cooling setpoint.

6 Summary and Outlook

This paper introduces a workflow to use MODELICA in the control system design and development process of a zero emission yacht's thermal system. It is shown that a MODELICA model of the thermal system can be compiled into a FMU, which is then embedded to a python wrapper to enable communication via MQTT to the control layer.

This enables the design of a control system even though the real system has not yet been finalized. It is also possible to assess scenarios that are quite difficult to evaluate in the real world. When the project has a tight schedule, the time taken to implement the control parameters of the real system can be reduced. An assessment of the time and cost savings by using this method may also be of interest.

For this project, the creation of the remaining submodules and their assembly into the whole system is still pending. Once the yacht has been constructed, it will be possible to evaluate the quality of the parameters

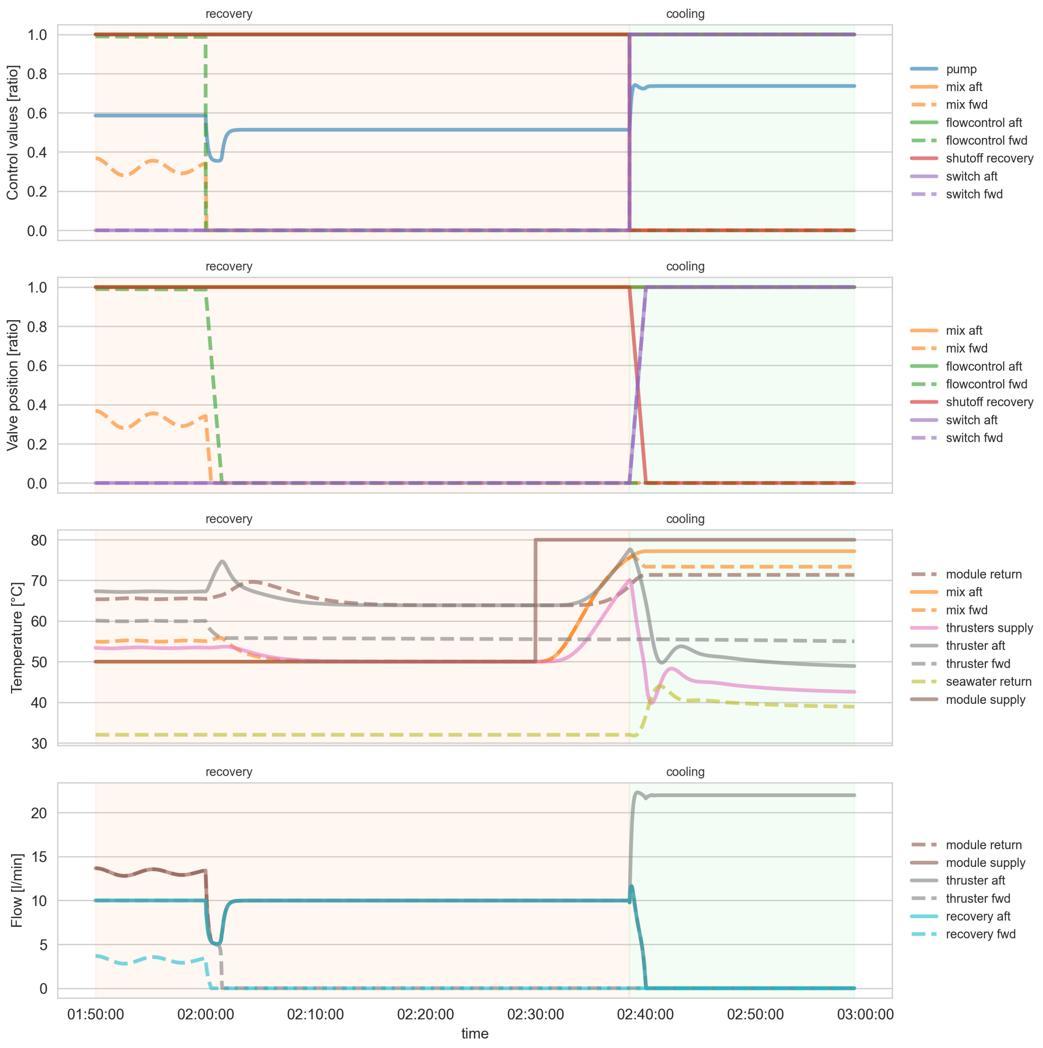

Figure 7. Continuation of the simulation shown in Figure 6, including shutdown of the forward thruster at 02:00 and supply temperature increase at 02:30.

determined by the approach this paper has shown.

Acknowledgements

The authors acknowledge FOUNDATION ZERO for the possibility of sharing the results in this paper.

References

Alexander Neumann (2025). *pytransitions Version Release 0.9.3*. URL: https://github.com/pytransitions/transitions.

Banks, Andrew et al. (2019-03). *MQTT Version 5.0*. Tech. rep. OASIS. URL: https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html.

Brunnemann, Johannes (2020). *NAKULEK - Naturumlaufkühlung für Leistungselektronik : Schlussbericht XRG Simulation GmbH : Laufzeit: 01.07.2016-30.06.2020*. Hamburg. DOI: 10.2314/KXP:1755577478. URL: https://www.tib.eu/de/suchen/id/TIBKAT%3A1755577478.

Clarke, Daniel et al. (2023). "CO2 emissions from global shipping: A new experimental database". In: *OECD Statistics Working Papers* 2023/04. DOI: 10.1787/bc2f7599-en.

Colvin, Samuel et al. (2025-04). *Pydantic*. Version v2.11.4. URL: https://github.com/pydantic/pydantic.

Dassault Systèmes (2025). *FMPy Version 0.3.23*. URL: https://github.com/CATIA-Systems/FMPy.

Foundation Zero (2024a). *An exploration into making PVT panels suitable for use at sea*. URL: https://www.foundationzero.org/insights/pvt-panels.

Foundation Zero (2024b). *The Foundation^0 Power Hub: A sustainable energy source for the future*. URL: https://www.foundationzero.org/insights/power-hub.

Foundation Zero (2024c). *Uncovering the results of lightweight insulation*. URL: https://www.foundationzero.org/insights/insulation.

Foundation Zero (2025a). *How mast ventilation can reduce energy expenditure in the maritime industry*. URL: https://www.foundationzero.org/insights/mast-ventilation.

Foundation Zero (2025b). *Power Hub*. URL: https://github.com/foundation-zero/power-hub.

Hesselgreaves, John E. (2001). *Compact Heat Exchangers*. Oxford: Pergamon. ISBN: 978-0-08-042839-0. DOI: 10.1016/B978-0-08-042839-0.X5000-9. URL: https://www.

sciencedirect . com / book / 9780080428390 / compact - heat - exchangers.

International Energy Agency (2023). “Tracking Clean Energy Progress 2023”. In: URL: https : / / www . iea . org / reports / tracking-clean-energy-progress-2023.

Leslie-Miller, Mark and Bob van Someren (2022-11). “Hydro generation, solar, electric and heat generation”. In: 27th International HISWA symposium. URL: https : / / www . foundationzero.org/insights/the-boat.

Modelica Association (2023-03). *Modelica – A Unified Object-Oriented Language for Systems Modeling. Language Specification Version 3.6*. Tech. rep. Linköping: Modelica Association. URL: https://specification.modelica.org/maint/3.6/MLS.html.

Thiele, Bernhard et al. (2017-05). “Towards a Standard-Conform, Platform-Generic and Feature-Rich Modelica Device Drivers Library”. In: *Proceedings of the 12th International Modelica Conference*. Ed. by Jiří Kofránek and Francesco Casella. Prague, Czech Republic, pp. 713–723. DOI: 10.3384/ecp17132713.

XRG Simulation GmbH (2025-03). *HVAC Library v.3.4.0*. Tech. rep. Hamburg: XRG Simulation GmbH. URL: https://xrg-simulation.de/seiten/hvac-library.