Development of Methods for Modular Power Plant Controls

DIPLOMARBEIT

Ausgeführt zum Zwecke der Erlangung des akademischen Grades eines
Diplom-Ingenieurs (Dipl.-Ing.)

unter der Leitung von

Univ.-Prof. Dr.sc.techn. Georg Schitter
Dipl.-Ing. Martin Melik Merkumians

eingereicht an der

Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik
Institut für Automatisierungs- und Regelungstechnik

von

Tobias Schwabel BSc.
Matrikelnummer: 11719753

Wien, im Oktober 2024

Technische Universität Wien
Karlsplatz 13, 1040 Wien, Österreich

Eigenständigkeitserklärung

Hiermit erkläre ich, dass die vorliegende Arbeit gemäß dem Code of Conduct, insbesondere ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel, angefertigt wurde. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In noch im Ausland in gleicher oder in ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

4.11.2024
Datum

[redacted]
Tobias Schwabel, BSc.

Acknowledgement

First, I would like to thank Prof. Dr. Georg Schitter, Dipl.Ing. Dr. Walter Fraißler and Dipl.Ing. Christoph Kukovic for making it possible for me to write such an exciting thesis and for always being available to answer relevant questions.

I would especially like to thank my supervisor Martin Melik Merkumians for his constant and patient support and for providing me with valuable input from the design to the final draft.

I would also like to thank the entire HIS of VERBUND AG for providing me with a very enjoyable working atmosphere during the preparation of my thesis.

Abstract

Power plants are manufactured almost exclusively as prototypes due to highly individual site requirements. This requires intensive planning not only in construction but also in control system design. This work offers a modular design for hydropower plant control systems, providing a scalable and flexible method through the use of pre-fabricated control modules to enable the quick and secure realization of projects. To this end, a methodology is presented that leads from the breakdown of power plant specific control elements into their functionality via a hierarchical structure and a random search cluster algorithm to a final control structure. For this purpose, the behavior of the power plant is divided into three different operating flows, which can be integrated separately into a machine module. The result of the designed method is a structure with module proposals which provides a separation between the flow specification to machines and weirs. These can be duplicated and adapted as required with the coordination of a state module, provided that specified interfaces are observed. The cluster algorithm leads to a reduction in the number of modules from 46 to 17. This makes it possible to design power plants more efficiently for rivers with different water volumes and to integrate machine sets individually into the control structure of other power plant types such as pumped storage power plants.

Kurzfasssung

Kraftwerke werden aufgrund sehr individueller Standortanforderungen fast ausschließlich als Prototypen gefertigt. Das bedeutet nicht nur intensive Planung im Bau, sondern auch in der Steuerungsauslegung. Diese Arbeit bietet als Lösung eine modulare Auslegung von Wasserkraftwerkssteuerungen und stellt durch den Einsatz vorgefertigter Steuermodule eine skalierbare und flexible Methode dar, um eine schnelle und sichere Realisierung von Projekten zu ermöglichen. Dazu wird eine Methodik vorgestellt, die von der Aufschlüsselung kraftwerksspezifischer Steuerelemente in deren Funktionalität über eine hierarchische Struktur und einen Random-Search-Clusteralgorithmus zu einer abschließenden Steuerstruktur führt. Dafür wird das Verhalten des Kraftwerks in drei verschiedene Betriebsflüsse unterteilt, die sich getrennt in ein Maschinemodul einbinden lassen. Das Ergebnis der konzipierten Methode ist eine Struktur mit Modulvorschlägen, die eine Separierung der Durchflussvorgabe zu Maschinen und Wehren vorsieht, die mit der Koordinierung eines Zustandsmoduls beliebig dupliziert und adaptiert werden können, unter der Voraussetzung, dass vorgegebene Schnittstellen eingehalten werden. Der Clusteralgorithmus führt zu einer Reduzierung der Modulanzahl von 46 zu 17 Modulen. Damit ist es möglich, Kraftwerke effizienter auf Flüsse mit unterschiedlichen Wassermengen auszulegen und Maschinensätze individuell in die Steuerstruktur anderer Kraftwerkstypen wie Pumpspeicherkraftwerken einzubinden.

Acronyms

CIM Computational Independent Model. 18

COMP components. 22, 23

DPM Design Property Matrix. 20

DSM Design Structure Matrix. 3, 4, 14, 23–27, 38, 40–42, 66

FB Function Block. 13

FC Function. 13

FEA Modular Functional Units. 22

HMI Human Machine Interface. 12, 23, 48, 49

HQ100 hundred-year flood. 47

MDA Model Driven Architecture. 14, 17–19, 27

MFD Modular Function Deployment. 14, 19, 24–27, 38

MIM Module Indication Matrix. 20

MP Modular Plant. 22

MTP Modular Type Package. 14, 23

PEA Modular Process Units. 22, 23

PIM Platform Independent Model. 18

PLC Programmable Logic Controller. 12, 13, 21, 47–50

PLCs Programmable Logic Controllers. 12, 13, 37, 47

POL Process Orchestration Layer. 23

POU Program Organization Unit. 13

PRG Program. 13

PSM Platform Specific Model. 18

QFD Quality Function Deployment matrix. 20

RTUs Remote Terminal Units. 12

SFC Static Frequency Converter. 13, 14, 34, 35, 42

UML Unified Modeling Language. 14–16, 18, 27–30, 32, 48, 66

List of Figures

Contents

CHAPTER 1

Introduction

With a history of over 3,500 years, the use of hydropower is one of the oldest forms of energy production. Until the end of the 19th century, hydropower was used almost exclusively to operate mills and pumping wheels to irrigate fields.

Following the invention of the electrodynamic generator by Werner von Siemens, the construction of hydroelectric power plants to generate electricity began around the turn of the century. To date, hydroelectric power plants are one of the most important electricity generation methods, accounting for 16% [1] of the world's electricity, which is becoming even more important due to the dangers of climate change.

Today, hydroelectric power plants are built and used in various designs. They are implemented for both direct energy production, in the form of dammed up water in rivers in front of run-of-river power plants, and for energy storage, in the form of storage or pumped storage power plants.

1.1 Mission Statement

The development of hydroelectric power plants is a very expensive business. Construction costs alone make small power plants quickly unprofitable, as construction and planning costs can only be amortized over many years. Excavations must be performed for water storage and passage, concreting work must be carried out for the construction of dams and machine houses, and diversions of the waterways must be planned [2]. All of these efforts represent a certain investment risk, which makes it even more important that the finished power plant runs as quickly and error-free as possible. Hydroelectric power plants are currently manufactured almost exclusively as prototypes, from design to control, as each power plant requires different designs due to local conditions. Since power plants are not mass produced, new innovations and strategies with more efficient tools are constantly developed in the planning phase of newer power plants to ensure the maximum achievable performance of the site. This greatly complicates the reuse of components and associated controls.

With a service life of more than 50 years, longevity is the most important asset of these facilities. In order to be able to carry out future modernization work on existing power plants quickly and safely, a control system is required that is flexible, well structured, and easy to read. The current design of individual controls for each power plant carries the risk of errors that can cause damage, as well as the need for extensive test phases that cause project delays. One way to offer planning security and reduce investment risk is the design of power plants using a modular approach.

Modularity describes the division of entire systems into smaller sub-components or modules that can be reassembled in a suitable form and function. Components are defined by their function and are separated from the outside via suitable interfaces. This enables reusability to ensure the exchange and variation of new components in order to achieve the desired system behavior. This motivates the first research question.

Research Question 1

Is it possible to develop a modular control approach that can also be used to map the operating cases of other hydropower plants?

Exchanging or expanding modules offer the advantage that individual system requirements can be incorporated into existing systems, and the size, and thus scalability, in production and design can be achieved without changing the entire system. During the design of the modules, modularity offers the advantage of splitting know-how into different system parts, and thus a clearer separation of necessary human resources. Modular systems are also characterized by the fact that they have a reduced time to market, as existing modules have already been tested, leading additionally to increased safety. Another important aspect for this work is that modular systems enable easier reverse engineering. Systems do not have to be strategically redesigned, components can be classified based on their function and put together to form an entire system. Since power plants can have different designs with the same functions but different sizes and layouts, modularity and functional separation can help design modules in variable sizes and integrate them into the control system using suitable structures and interfaces. The second research question can therefore be derived for hydropower plant control systems.

Research Question 2

Which logic modules can be reused in a scalable manner?

The challenge of creating modular systems is to find a structure with systematic dependencies that allows system adaptations without affecting basic functions. This normally leads to less system optimization and loss of uniqueness, at the cost of comprehensibility and ease of adaptation. The ongoing need for coordination and development of different modules, leads to ongoing interdisciplinary exchange to maintain system boundaries [3]. In addition, a balance must be found in the design between project-independent and project-related engineering activities with regard to module variability [4] in order to ensure both reusability and applicability.

1.2 Outline

This work will evaluate the feasibility of modularity, and find a methodology for how control structures for hydroelectric power plants can be constructed in a modular way. Modularity can be used to respond to the individual needs of power plant locations without redesigning the entire power plant structure. Chapter 2 explains the state of the art of how power plants are constructed, and what current modularization methods are. Furthermore, an approach is given in Chapter 3 from breaking down the operating cases of run-of-river hydroelectric power plants to clustering modules using Design Structure Matrix (DSM). Finally, in Chapter 4 the control structure is tested on a specially designed prototype in order to test a modular flow control, vary functions, and analyse behaviour of exchanging modules. After analyzing the structure for run-of-river power plants, a theoretical approach is developed in Chapter 5 as to what a control structure for storage and pumped storage power plants looks like and is then compared with modules of the run-of-river power plant.

CHAPTER 2

State of Art

In order to find a scalable modular structure for hydroelectric power plants, it is first necessary to break down the different types of hydroelectric power plants and how they differ. Section 2.1 deals with differentiating between different types of construction, breaking down the components of a power plant and describing the basics of the control system in power plants. After describing the hydroelectric power plants, strategies are described for how modular systems can be designed. Finally, a clustering method using DSM is described to group large modular structures into simpler, more manageable blocks.

2.1 Water Power Plants

Water power plants are classified according to technical (river engineering and structural engineering), topographical, energy and water management aspects, the useful net head, the mode of operation, the installed capacity, and the size of the body of water [5]. In order to differentiate between the different designs for control design, this work distinguishes between the technical and structural aspects. These are divided into run-of-river power plants, storage power plants with natural inflow, pumped storage power plants, use of ocean energy, depression power plants, glacier power plants and hydropower plants with underground storage systems. In this work, particular attention is paid to the forms of run-of-river power plants and their modular functional reusability will be tested on pumped storage power plants [5].

River power plant:

These constructions are installed on rivers to convert the usable water head and thus the potential energy of the water into electricity. They are characterized by their low head and a constant water flow. These power plants are designed in such a way that during regular operation they precisely coordinate the outflow from the machines and weir systems into the downstream water with the inflow to the upstream water.

There is no build-up behaviour during regular operation. Energy can therefore only be generated depending on the current inflow. River power plants are divided into two subcategories, run-of-river plants and diversion power plants.

Run-of-river power plant:

Run-of-river power plants have the property of having weir systems and machinery in the same course of the river. The incoming water is drained away in a predetermined ratio via the installed systems. If there is a flood that exceeds the maximum flow through the machinery, adjacent weir systems drain the difference in water so that there is no increase in the head water. Run-of-river power plants differ in their design, for how the weir systems and machines are arranged. The five basic designs are shown in Figure 2.1 and are now briefly explained.

Block construction (a):

This is the most commonly used design. Here, all of the power plant's machines and generators are combined in a machine house being connected on one side of the river bank. In an extension of the machine house, the weirs are built till the opposite bank of the river.

Twin construction (b):

In contrast to block construction, twin construction methods are built on rivers with a large developed flow and a small head in order not to create a disproportion of the bay width to the original river width due to large power house lengths on one bank. In this design, the powerhouse is therefore divided and designed symmetrically on both banks of the river.

Pillar power plants (c):

These structures consist of an alternating design between weir fields and turbine pillars. This is a newer design that results in a very even flow across the system components and less wear and tear. There is no need to build a connected power house.

Overflowable run-of-river power plants (d):

In the overflowable power plant, in contrast to the side-by-side design of the power plant and weir system, the functionalities such as accommodating the machine sets, storage and flood relief are compactly integrated into one structure. This combined component extends over the entire width of the river and has a weir attachment with individual flaps at its crown for keeping the water dammed. The machine sets are located at the bottom within the developer and are centrally designed across the entire width of the river.

Bay Power Plant (e):

In this design, the machine house is located in an artificially created bay on the edge of the river. This design is used when the flow cross section of the water must not be narrowed in order to continue to drain hundred-year floods. This variant is an adapted form of block construction.

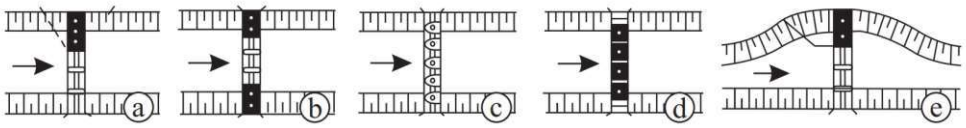

Figure 2.1: Components Arrangement of Run-of-River Power Plants [5].

Diversion power plant:

Here the machine house and weir systems are installed on separate river paths, in contrast to run-of-river systems. A distinction is made here between canal, loop and ribbon power plants. These power plants are used at low heads and offer the advantage that they can be built in a dry assembly. If a flood occurs here, the inflow must be diverted through the separate river paths into the flow path of the weir systems at an early stage so that there is no excessive level and flooding in the flow path of the mechanical systems.

Pumped storage power plants:

These plants, unlike river power plants, are not power plants that constantly try to convert the kinetic energy of rivers, but rather store energy in potential form in reservoirs and make it available as needed. Due to the large fluctuations in daily electricity consumption, there is a large difference between the base load and peak load of the entire network. Due to these fluctuations, controllable power plants are required that provide both peak coverage and fine control. Pumped storage power plants offer cost-effective operation to use electricity during low-load times to fill the storage basins and to supply electricity during peak-load times. This form of energy storage is becoming particularly important as the expansion of renewable energies such as wind turbines and solar systems makes production more dependent on the weather and the power supply is exposed to greater fluctuations. Phases of bad weather can then be bridged using these storage systems.

2.1.1 Construction and Components of Water Power Plants

An important part of this work is classifying the main components of a power plant in order to be able to build a holistic control structure later. Power plants are critical infrastructure facilities, which is why there is very little documentation in the literature on the control structures of the control technology and switched signals.

Components of pumped storage power plant

A rough overview of the structure of the control technology in automation units is offered in [6], which are made up of functional areas, which in turn are divided into functional groups. Each automation unit controls and monitors the functional area assigned to it and is connected to the process LAN. Automation units provide data nodes and controls that record and transmit signals from the functional areas and forward them to the central control room.

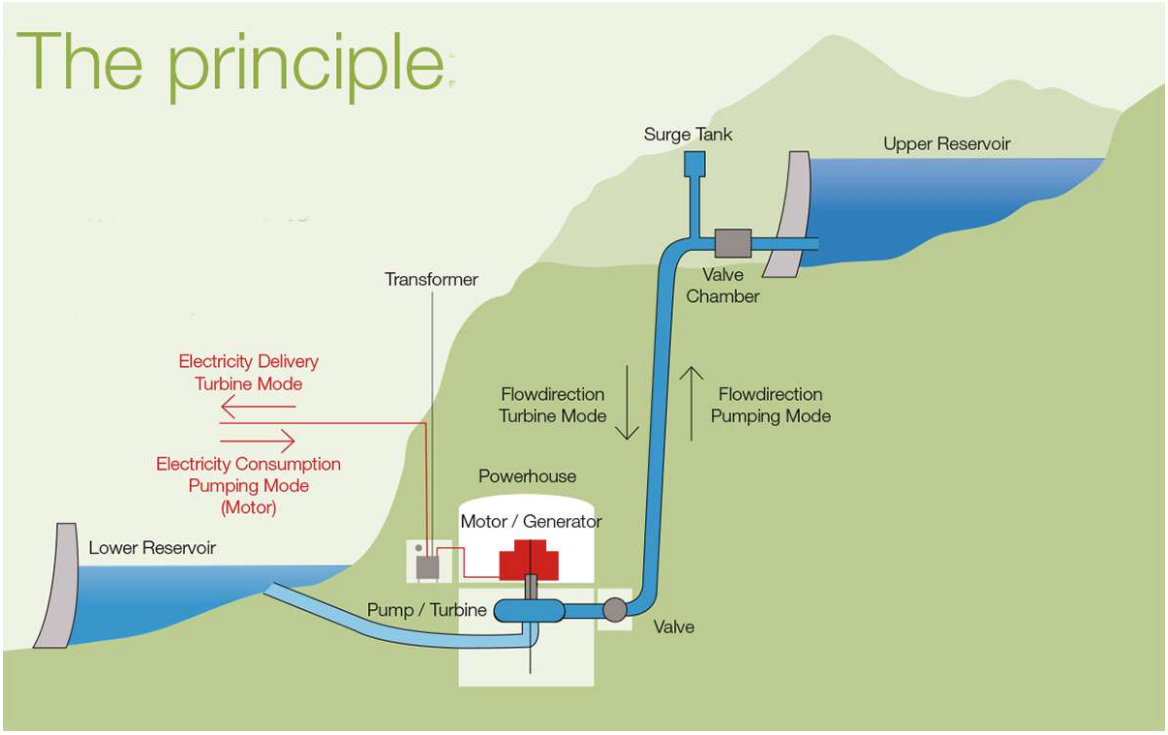

Figure 2.2: Pumped Storage Power Plant Working Principle [7].

The functional areas differ depending on the type of power plant. The functional areas of the Reißegg II pumped storage power plant in Carinthia/Austria are: switchgear and related systems in the transformer cavern area, switchgear and related systems in the machine cavern area, static frequency converter, hydraulic protection, machine, ball valve, intake manifold valve, cooling water system and drainage system with external facilities.

Looking closer at the functional area of the machine, it is visible that it contains all the equipment for operation, measurement and message recording and monitoring for the respective machine set. The functional groups of the functional area of the machine are block transformer, generator output, motor-generator, pump turbine, turbine controller, vibration monitoring, synchronization device, electrical block protection, excitation device, machine auxiliaries motor generator/pump turbine, ball valve, intake manifold gate, static frequency converter, and protective devices (e.g. shut-off valves and inspection flaps).

Functional groups can have their own independent controls, this is especially the case for critical units such as ball valve, intake manifold gate and turbine controllers, but it is not explained in more detail what the controls switch are and whether there is a more precise stored structure here. Looking at the functional groups more closely, individual components of a pumped storage power plant can already be classified. Figure 2.2 shows a schematic visualization of the arrangement of the basic components of such a facility and illustrates the basic working principle.

In order to create the basis for the control structure for later, the components of such a system will now be briefly explained in their function:

Storage: The basic building block of a pumped storage power plant are the two water reservoirs between which water can be pumped or turbined to absorb or output power. If there is a change in the grid frequency as described by [8], a power requirement can be determined from the turbine´s moment of inertia and the speed. This is achieved by circulating the water volumes of the lakes.

Water conductor system: In pumped storage power plants and storage power plants, the water supply lines are used like pipes to supply and drain water to the machines. The valve chamber, the ball valve, and the surge chamber are valves to limit or shut off the water volumes. The valve chamber is a measuring and control device to shut off the water flow. The ball valve is similar in its function, except that it also provides an emergency closure device that closes automatically and is placed in front of the machines. The surge chamber is an artificially exposed opening to compensate for fluctuations in the pressure tunnel and enable a more even pressure distribution.

Block transformer: The transformer is used to transform the energy generated by the machine to a 380kV voltage level for discharge into the grid. In this work, transformers are to be implemented as pure tap changers that create a balance between the supply and supplied voltage.

Ternary machine set: Pumped storage power plants basically have two different types of turbine design. The designs that were more common in the past are ternary machine sets shown in Figure 2.3, where the pump and turbine are separate but on one shaft with the generator. This design has the advantage that it requires less starting power, as the hydraulic short circuit function can be used here. Here, water is both turbined and pumped, so that the non-controllable pump does not have to be connected to the grid at full power, but part of the energy can come from the head water route [6]. The turbine design here is a Pelton turbine due to the large head. This design is divided into controllable units: a needle as a flow throttle and a jet deflector as an emergency discharge. The other design is pump turbines with built-in Francis turbines. This type of turbine has the advantage of being more compact to install, as only one component needs to be installed for both pumping and turbine operation. To change from pump to turbine operation, the machine only needs to be reversed and the direction of rotation changed. The disadvantage is that the pump turbine's switching times are slower, which is essential, especially in times of greater fluctuations in producers and consumers.

Generator: The generator in power plants is almost exclusively installed as a synchronous generator. Synchronous generators have the advantage that once they are synchronized with the grid, they run at a synchronous speed, just depending on the number of pole pairs of the generator and the grid frequency. This means that they do not have to be additionally speed-controlled to deliver the required grid frequency. The generator can be controlled via the torque supplied as well as the applied excitation current.

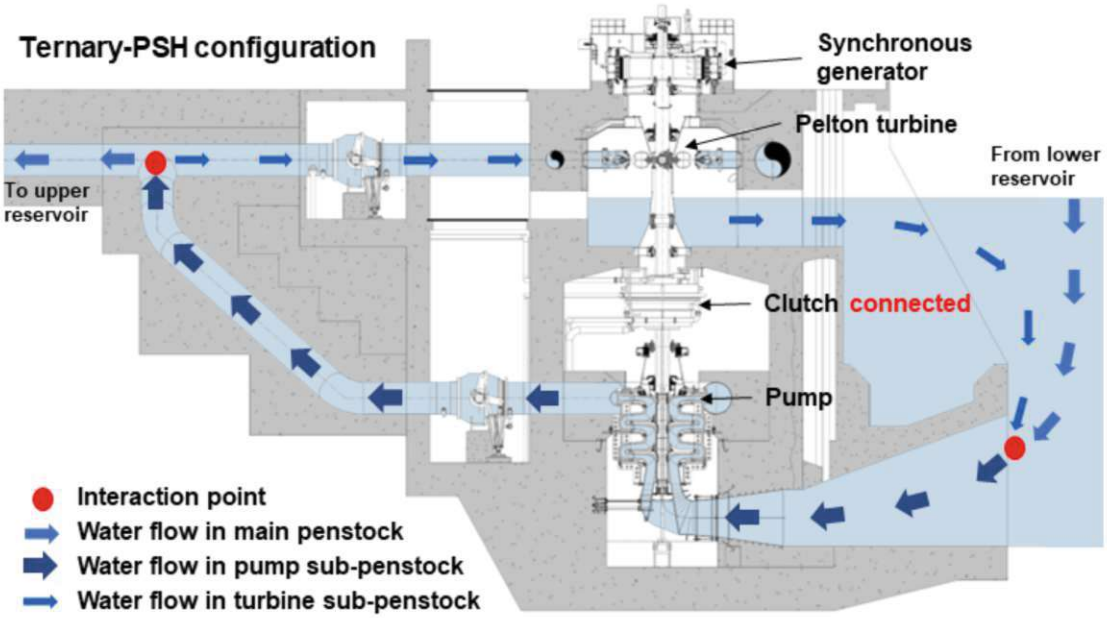

Figure 2.3: Ternary Machine Set, separated pump and turbine [9].

Synchronization setup: It is important to note that synchronization to the grid is not achieved simply by switching the circuit breaker, but rather the generator must meet the four synchronization conditions. These consist of the same phase position, the same phase sequence, the same voltage, and the same frequency. If one of these conditions is not met, power surges and damage to the system may occur.

Static frequency converter: Since pumped storage power plants are used to compensate for fluctuations in the grid, it is essential that the turbine sets can be controlled. As already mentioned, ternary machine sets can be controlled in turbine mode by throttling the water volume and in pump mode by hydraulic short circuit. In pump turbines, however, the operating point of the pump must be set via a variable speed of the otherwise non-controllable pump; this can be done via an intermediate static frequency converter. In turbine mode, this also created the option of varying the speed for more efficient operating points and higher working ranges. Due to the relationship between the static speed and run-of-river power plants, the ternary machine set is considered for the implementation of the modular control.

Turbine controller: The turbine controller is often listed as a separate component and takes over the entire control to guide the turbine to the desired operating point. It takes care of calculating the flow rates of the machine sets and keeps the speed of the turbine constant in island operation, on site operation and idle operation. It also establishes the connection to the power plant control system. Since the exact design of this component is also part of the critical infrastructure, documentation on this is very sparse. The function of the turbine controller is enabled in this work via staggered modules in the project planning of the modular structure.

Blow-out facility: An essential controllable component in pumped storage power plants is a blow-out device for starting the pump. Since the pump has to set a column of water that is sometimes hundreds of meters long in motion and the forces exerted under load would be too great, the water is blown out of the pump to allow it to start freely and is only switched off again when the pump has reached its nominal speed.

Excitation device: In order for a magnetic field to be generated in the generators stator to induce voltage, direct current must flow through the rotor. The excitation system provides the required voltage and regulates it. A distinction is made here between external excitation, which corresponds to a supply via the grid, and self-excitation, which corresponds to a self-connected excitation machine. With external excitation this is done via slip rings and brushes with an intermediate rectifier or with larger machines via an excitation transformer. The level of the current depends on the load and voltage. With self-excitation, the excitation current itself is generated via a so-called GSM auxiliary excitation machine, which splits up again, especially in older generators, into an auxiliary and a main excitation machine, with the auxiliary excitation machine supplying the current for the main excitation machine.

Protective equipment & auxiliary equipment: The protective devices include all components that monitor system limits and trigger emergency shutdowns. These include vibration monitoring, electrical block protection, shut-off valves, inspection flaps and intake manifold protection. The auxiliary systems include hydraulic systems and cooling systems that provide essential services for the operation of the systems but are not controlled by a central control system in normal cases.

Components of run-of-river power plants

Like the pumped storage power plant, the components of the run-of-river power plant and their function are shown in Figure 2.4. The design is similar here, but it differs in the low head and the difference in the continuous water inflow. This results in other designs that can be divided into the following components:

Kaplan turbine: In contrast to Pelton turbines, which are used for large head heights, Kaplan turbines are mostly used in river power plants. The difference to Pelton turbines is that the water inflow is not throttled by a nozzle needle but by a guide vane mechanism. This is designed as a circular segment in front of the turbine blades, which uses hydraulic flaps to proportionally control the speed and direction of the inflow to the turbine inside. The advantage of the Kaplan turbine is that the turbine blades can be adjusted depending on the inflow quantity and speed. These are regulated so that they run twist-free and the water can flow away unhindered.

Bypass & Emergency release: Similar to the Pelton turbine's jet deflector, a bypass can be installed in the Kaplan turbine to divert the water column that would hit the machine, unloaded and disconnected from the grid, in the event of an emergency shutdown. The bypass is installed as a pure flap opening. In the event of high water levels or other reasons for additional discharge of the head water, an emergency discharge

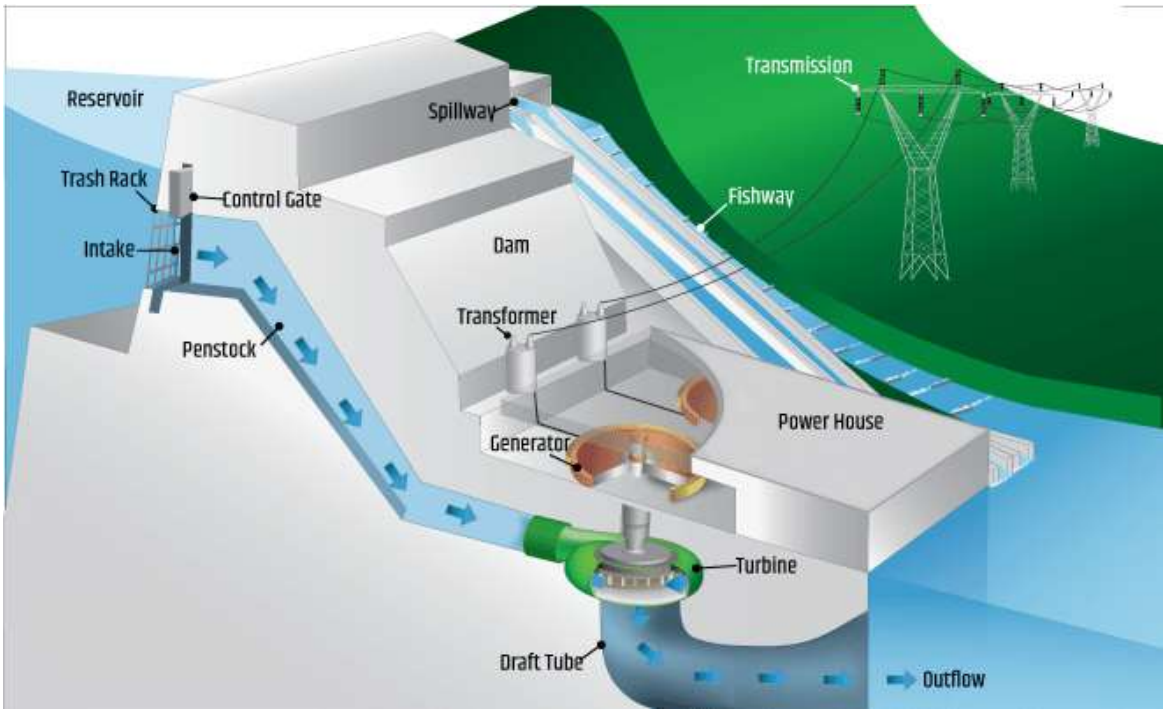

Figure 2.4: Run-of-River Working Principle [10].

is installed next to the machine house. This is constructed similarly to a weir field and can be controlled by tilting a weir wall.

Synchronous machine: As already explained for the pumped storage power plant, synchronous machines are also used here because they achieve particularly high levels of efficiency and have a constant speed regardless of the load.

Weir: In normal operation, water is released purely through the turbines. Water only flows over the weirs when the plant is shut down, is undergoing maintenance, or during flood operations, when the flow through the machine has already been exhausted and the excess water must also be drained away. The weir can be installed in various designs, but the common feature is that all of them open segments that release a desired water opening area and allow water to flow away. Power plants normally have at least 2 weirs to prevent flooding even if one weir fails.

Intake manifold valve: The intake manifold valve is a closing device similar to the slide chamber in a pumped storage power plant. It serves as an additional safety device when the machine is out of operation to prevent flow as a second level.

Static frequency converter: This component does not always have to be installed, but can be helpful when machines are to be started or synchronized. If no static frequency converter is installed for starting the machines, the speed of the turbine in idle or isolated operation can be controlled purely via the water supply. The speed is then reduced by closing the control device.

Hydraulic protection: The hydraulic protection connects all automation units of the hydraulic protection in the locations. This automation unit records and processes the process signals relating to the protection in the respective external locations and transmits them to the relevant functional areas via the respective process LAN [6]. The mechanical protection monitors all relevant process signals such as pressure, flow, level, etc. in the transformer and turbine areas to ensure that they are functioning properly within the set parameters. If the set limit values are exceeded or not met, a warning is issued or the machine is shut down in accordance with the threshold deviation and in accordance with the implemented trigger matrix [11].

Human Machine Interface: The visualization system is designed as a client/server configuration and is separate from the process automation system. In addition to the operating elements required for manual operation, the individual images also contain the display of all system measurements, electrical and/or mechanical parameters required for operation [11]. Remote status specifications can then be set via the control room.

2.1.2 Control Systems of Hydroelectric Power Plants

Hydroelectric power plants are highly automated, self-regulating, self-sufficient systems that are operated by electronic control devices, and enable operation via external control rooms. The tasks of the process and control technology are designed to control, regulate and monitor the system parts and to ensure the recording of the data necessary for operation such as water levels, flows, forecasts, weir positions and machine data [5]. In the higher-level control, the measurement data are combined, prepared, and processed the impeller - stator wheel connection, vibration diagnosis, or operational management. This topology of the control technology is usually designed in a redundant ring structure [11]. This ensures that in the event of failures, other system areas do not become inaccessible, but that all remaining sections can continue its function. The control devices within the system areas are Programmable Logic Controllers (PLCs) or Remote Terminal Units (RTUs) running close to the system, which provide the direct interface between software and hardware by transferring the signals received from other PLCs, sensors and Human Machine Interface (HMI) according to specified logic to outputs in the form of analog signals or PWMs.

More precisely, these PLCs work in such a way that their functional behaviour can be cyclically divided into 4 steps. At the beginning of a cycle, the Programmable Logic Controller (PLC) reads the input data from the connected interfaces such as sensors and other PLCs and saves them in a process image. With this stored data, the logic implemented on the PLCs is now processed and writes the resulting data as outputs to the connected devices such as actuators and other PLCs which influence the remaining process steps. In the final step of the cycle, the remaining time of the implemented cycle time is waited out, to ensure synchronous operation with the connected units. [12]

What has to be taken into account here is that if an error occurs immediately after the first cycle step, reading in the data, it will require two full cycle times to be able to react to it. First the current cycle step is processed, then the error that occurred is read in and at the end of processing the system response is issued. In addition, when implementing the logic, care must be taken to ensure that no loops waiting for events

are implemented within these systems, as this would result in the current cycle never being completed. Data from the next cycle and thus the event would never be read in. The system would be in an endless loop.

The logic within the PLCs is implemented using the programming languages of the standard IEC 61131-3. This consists of text-based programming languages and graphical programming languages. It also defines the three Program Organization Units (POUs): Programs (PRGs), Function Blocks (FBs) and Functions (FCs). These POUs help to structure PLC code and make it reusable. Some form of this software architecture is important to ensure the quality of the software. PRGs and FBs essentially differ from FCs as they have internal memory and FBs can be instantiated. FBs are blocks that can be made up of various logic gates such as AND/OR/XOR etc., but also more complex controllers.

However, since IEC 61131 was not designed to be used flexibly in systems and to ensure scalability and reconfigurability, the IEC 61499 has been developed [13]. The motivation of IEC 61499 was to take the step from decentralized control technology to distributed control technology. With decentralized control technology, the control software is processed locally in the PLCs in switch cabinets, whereas distributed control technology follows a modular philosophy in which automation components are completely relocated to the field and function independently and in a network with each other. This modular approach now offers an open architecture, with which components from different manufacturers can be integrated and the corresponding control code can be created in an adaptable manner.

2.1.3 Operation of Water Power Plants

For the implementation of a modular control structure in this work, it is important to understand the behavior of a power plant in normal operation and what requirements are placed on the control system. The behaviour consists of a sophisticated interaction of the components listed in Section 2.1.1. These interactions can be divided into different operating cases. A distinction between the following operating cases from [6] for the control of a machine set in the pumped storage power plant based on Reißegg II can be made: starting up in the turbines or pumping operation, turbine operation, pumping operation, full load shedding with de-excitation, phase shifter operation, and braking. The full load shedding operation case is triggered when the power grid is overloaded and the circuit breaker has to be opened briefly. The sudden removal of the load can lead to an increase in the speed of the turbine, in which case additional protective measures must be taken. The phase shifter operation is a form of operation for reactive power compensation in the network. All valves are closed and all the water is drained from the machine set. The machine and pump now run idle and draw active power from the network to supply the excitation current. The reactive power fed in can be regulated and fed in by setting the excitation current.

Furthermore, in selected systems there is the operation case of switching operation between turbine to pump operation and pump to turbine operation as well as back-to-back operation, in which if the Static Frequency Converter (SFC) fails, a second machine set in turbine operation is used to start up the pump until it is synchronized with the grid. In addition, for automatic start-up and shutdown of the machines, a

distinction can be made between the operating goals of turbine idling and pump standby operation. In contrast to the operation cases, these operating goals represent states that can be started up in a stationary manner. Turbine idling is a state in which the machine is at nominal speed and nominal voltage, but is not yet connected to the grid. For the operating goal of pump standby operation, the machine is started up via the SFC and brought to nominal speed, but is not yet synchronized with the grid, similar to the turbine idling. [11]

2.2 Modularity

Referencing on the benefits mentioned in Chapter 1, a modular approach for a control design is considered to be able to design power plants more quickly and safely. Modularity is defined as "decomposition of a product into building blocks modules with specific interfaces, driven by company-specific reasons [14]". A distinction is made between physical and functional modularity. With physical modularity, modules are physically separated from each other and form a building block of a system. Functional modularity refers to the logical division of a system into its functional tasks or processes. These units are implemented by software in the programming languages of IEC 61131, IEC 61499 or similar and are separated from each other via defined interfaces.

Modularity is already used in different areas of application to address modern problems. Within the construction industry, physical modularity is used to outsource the production of buildings or system components and to carry out the final installation at the site [15]. By outsourcing production, resources do not have to be relocated to remote areas where, on the one hand, it can be difficult to find the appropriately trained staff and, on the other hand, the technologies to be able to produce cost effectively. These pre-constructed units can be considered as miniature projects which need to be designed and built independently [16]. In addition, the outsourced modular production of plant components offers protection from harsh weather conditions and saves accommodation for the labor force [17].

The challenge for the production industry is that customer requests are becoming more and more individualized and production systems must be designed with flexibility. By utilizing a modular design for process steps across various parts of the system and arranging them flexibly through designed interfaces, producers can expand their product variety without needing to establish a new production line. [18].

Now there are a few different methods for designing software systems modularly. The methods differ in fundamental aspects such as the functional representation of complex dependencies and signal exchanges of systemic functionalities using visual modeling languages (e.g., Unified Modeling Language (UML)), the extension of modules based on requirements to existing systems (e.g., Model Driven Architecture (MDA)), qualitative approaches to fulfill system requirements (e.g., Modular Function Deployment (MFD)), the identification of modules for the modularization of existing non-modular systems, the clustering of modules (e.g., Design Structure Matrix (DSM)) and interface standards (e.g., Modular Type Package (MTP)). These methods will be addressed in this section.

2.2.1 Unified Modeling Language

Since power plants are very wide-ranging systems, a method is needed that model systems very simply and understandably in the early design. Unified Modeling Language (UML) is a visual modeling language standard by Object Management Group (OMG) to represent the behavior and design of such complex software systems. UML is not a programming language, but diagrams can be read in via software and used to generate code for control systems automatically [19]. It is used to model complex mechatronic systems in an object-oriented way to make it possible to consider components of mechatronic systems that belong to the control software domain as well as to any physical domain [20]. Through its standardized notation with classes, objects, activities, states, and more, it offers well-suited and interpretable modelling of software architectures and relationships. UML has many types of diagrams, however the basic distinction is made between behavior diagram and structural diagram.

Structural diagrams provide information about how a system is build up and therefore about the static structure of components. Here, building blocks, and their relationships and inheritance are presented without revealing any information about a chronological sequence. Examples of using structural diagrams include software where classes, such as products and users, are defined, and linked to objects of those classes with differing properties. These users can then be subject to relationships, but there is never a dynamic connection between the individual components, only rigid relationships or connections.

Behavior diagrams are more suitable for depicting the functionality of modular software systems. In behavior diagrams, dynamic aspects are visualized and documented during their runtime in order to clarify internal dependencies. Depending on the diagram selected, different aspects can be highlighted. A fundamental distinction is made in UML2 between Activity, Interaction, State Machine, and Use Case diagrams.

Activity diagrams are ideal for mapping workflows using step by step solutions. For this purpose, a notation with a limited number of components is used to depict a structure that is easy to understand. This can be implemented from [21]:

- round-edged rectangles to represent actions
- diamonds to represent a decision node
- bars to represent the combination or separation of flow between steps
- black circle to represent the starting point of the process flow
- outlined black circle to represent the endpoint of the process flow
- send/receive symbols to send trigger signals between modules

Figure 2.5 shows the notation of the basic symbols. Starting from the black circle, the aim is to follow a chronological progression up to the outlined black circle. If there are too many loops within the flow, or if there are splits within the loops that are connected to other processes, this can lead to unreadable interconnections or unclear state changes. Therefore, when designing a loop, a split needs to be merged again before the loop is repeated. Alternatively, state machine diagrams can be helpful. A

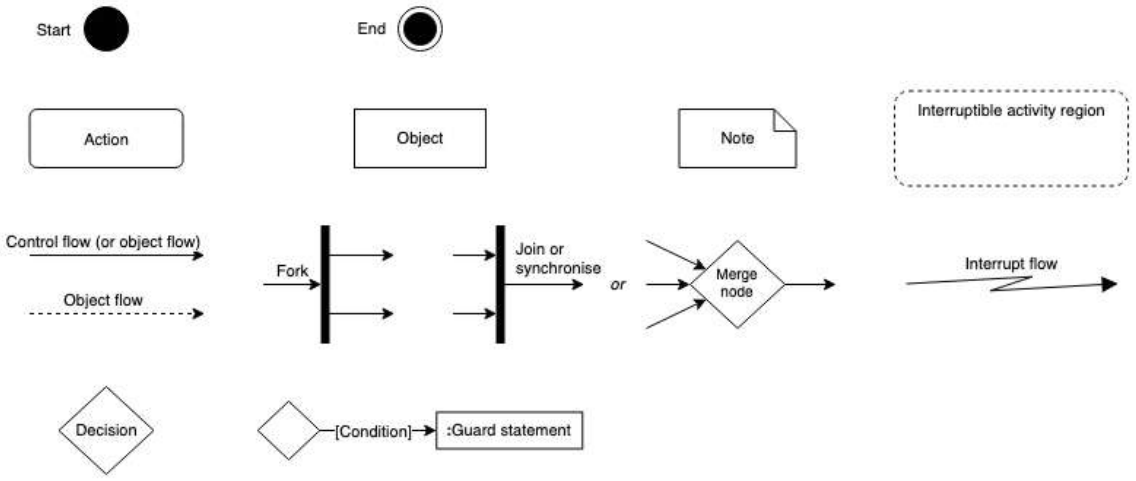

Figure 2.5: UML Notation for Activity Diagrams [22].

state machine diagram can provide information about the current operating state of a process and how the process behaves when state changes occur. Using the example of a hydroelectric power plant, individual components can be depicted and their behavior illustrated when active. Examples of this are valves or switches that follow a certain routine after receiving a signal or systems that initialize or calibrate based on time stamps. This diagram is therefore not suitable for setting up a power plant in which ongoing operations must be mapped and signals are not only transmitted and processed in binary form. However, it can provide information about the standalone components and their behavior.

The advantage of UML is that it creates a comprehensible abstraction that makes processes understandable, in which teams of experts from different disciplines can discuss a topic and introduce ideas and adaptations into implementations without requiring precise knowledge of process communication types or the detailed implementation of the states. In addition, systems running in different languages can be connected, regardless of whether the subsystems are coupled analogue or digitally. An UML is therefore ideal for providing a methodology for the entire power plant.

Another way to map process flows would be the SysML activity diagram. The SysML activity diagram is an extension of the UML activity diagram with a focus not on software development, but on system technology and system architecture. A requirement and parametric diagram are also attached here. This corresponds to a rigid structure, which can be very helpful when designing new systems because of its very hardware-related perspective. However, it is not necessary to modularize existing systems first, since the dimensioning of the connections has already been parameterized, and now only the modular control structure is desirable. This is relevant for the scalability of new systems, as it can be recorded here which system limits the current modules have and which modules need to be replaced. Hardware modules must have the same operating ranges to avoid overloads, the interface must therefore transmit the system limits to the control modules. Then the control structure can react automatically when the modules are changed.

Since UML is primarily characterized by its ability to create an overview of complex

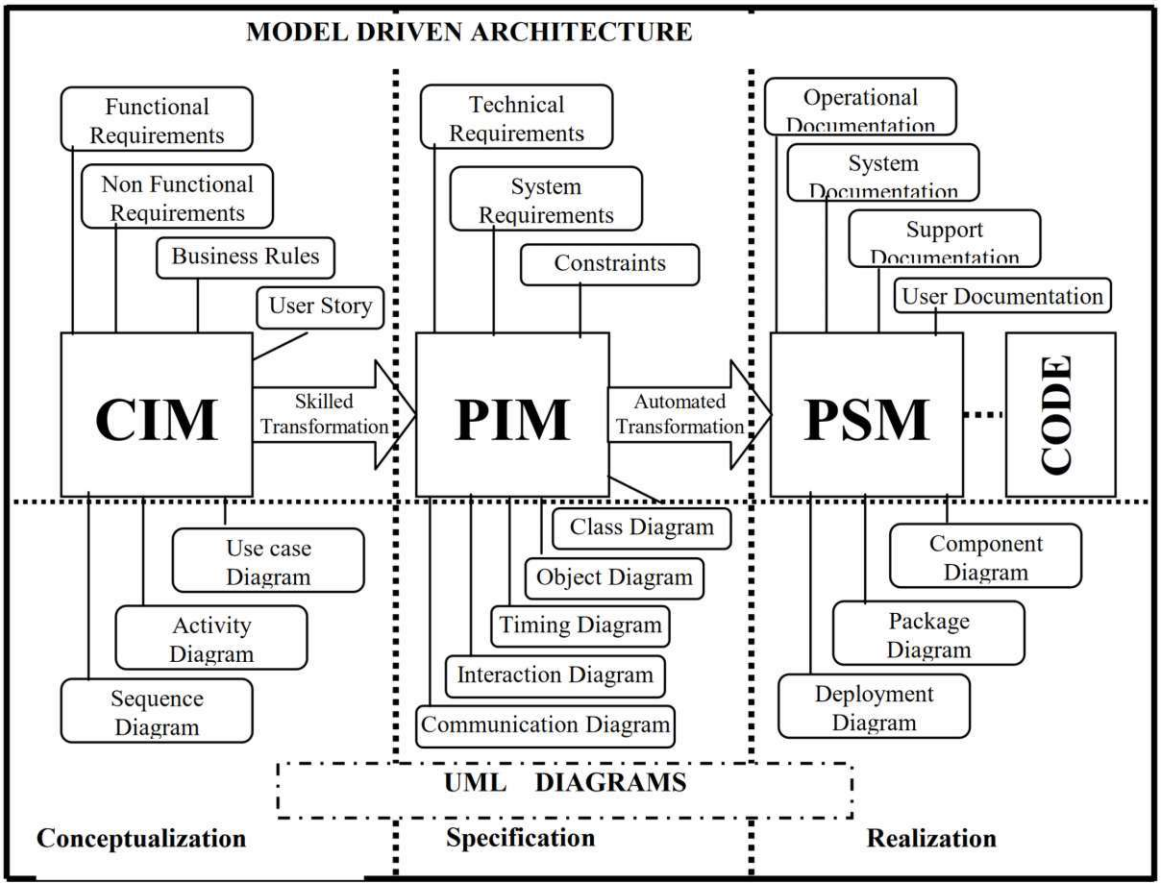

Figure 2.6: Model Driven Architecture Working Principle - illustration of the different model stages with associated model information and transformation up to the code [25].

systems in early design, it can be used to design the basic functional structure of power plants. This work does not focus on the methods used to automatically generate code, as generation [19] and automatic testing [23] have already been adequately covered in the literature.

2.2.2 Model Driven Architecture

Another method of mapping and expanding modular control systems is the so-called Model Driven Architecture (MDA). This is a procedure for building or expanding systems step by step. Here, a basic framework or software domain is assumed, and new desired functionalities and requirements are defined. The existing system is now tested and validated at various modeling levels for the new system requirements. The aim of MDA is to automatically create software code through generative programming and thus increase the efficiency of software programming, but to create better reusability, interoperability and portability through architectural separations [24]. Thus, MDA is intended to help minimize the programming effort for code generation and is a software development strategy that focuses on models and transformations rather than code generation [25].

To do this, the MDA uses 3 levels of abstraction, the Computational Independent Model (CIM), the Platform Independent Model (PIM), and the Platform Specific Model (PSM). The challenge here is to enable the transition from the created models to the more specific models and then generate the automatic code from them.

Figure 2.6 illustrates the flow of the five steps of the process. These consist of creating the CIM, resulting in separate functional solutions for the PIM, creating the combined PIM, transforming the PIM into different PSM levels, and combining the resulting PSMs. As can be seen, the CIM now serves to collect and model all requirements, including the business rules for the company system. Here, no technical information about a system is included yet. It does not provide any information about how a process is managed, only which output is to be expected, and which system inputs need to be dealt with, thus representing a system analysis.

The CIM provides the basis for the PIM, which is already formulated in a more technical way, which is broken down in the form of a UML activity diagrams [26] on how system-independent processes are connected and on how conclusions can be drawn from an input to an output. Furthermore, it lists all the links necessary and records constraints that arise due to logical interconnections and, therefore, is already referred to as the design step.

Another challenge is to convert the PIM into the PSM. Most research approach this transformation because of the similarities between these two levels [27, 28]. However, attention must be paid to ensure that no undesirable behavior occurs from the CIM to the PIM. This is one of the most critical steps, as special attention and testing must be paid on how to not lose the requirements, while creating a technical process [29].

In the last step, the PSM maps the implementation. The PSM is now purely about applying the existing PIM to a system and making it possible to generate code based on the existing facility and finally generating software. Various PSMs are required in software development, such as relational PSM or Web PSMs, to map the logical connections in an applicable way and make them combinable. Conversely, this also means that there can be different PSMs for the same requirements, depending on which system the PIM is to be applied to.

The advantage of MDA is that by separating the system functionality into specification and implementation, technology plan forms can be viewed separately from each other. A function can be designed directly based on the business logic without having to go directly into the hardware. There are function-oriented and component-based approaches [30], but to simplify transformations from the CIM to the PIM, responsibilities are considered as connectors between features and components, which makes it difficult to map interdependent system dynamics. This offers a very targeted process for implementing new technologies or expanding existing systems. However, it does not represent a method for mapping existing systems or defining them in a modular manner. The aim is to build a system based on desired behavior. Therefore backstepping is not possible with this methodology, as the functionality and behavior is only determined by the components, their interconnection, and interaction. It is advisable to exchange modular units and explore new modular blocks, but not to form modular blocks out of existing systems.

The focus of this work is to map existing power plant structures and, on this basis, derive a modular behavior structure, similar to the PIM. Requirements cannot be

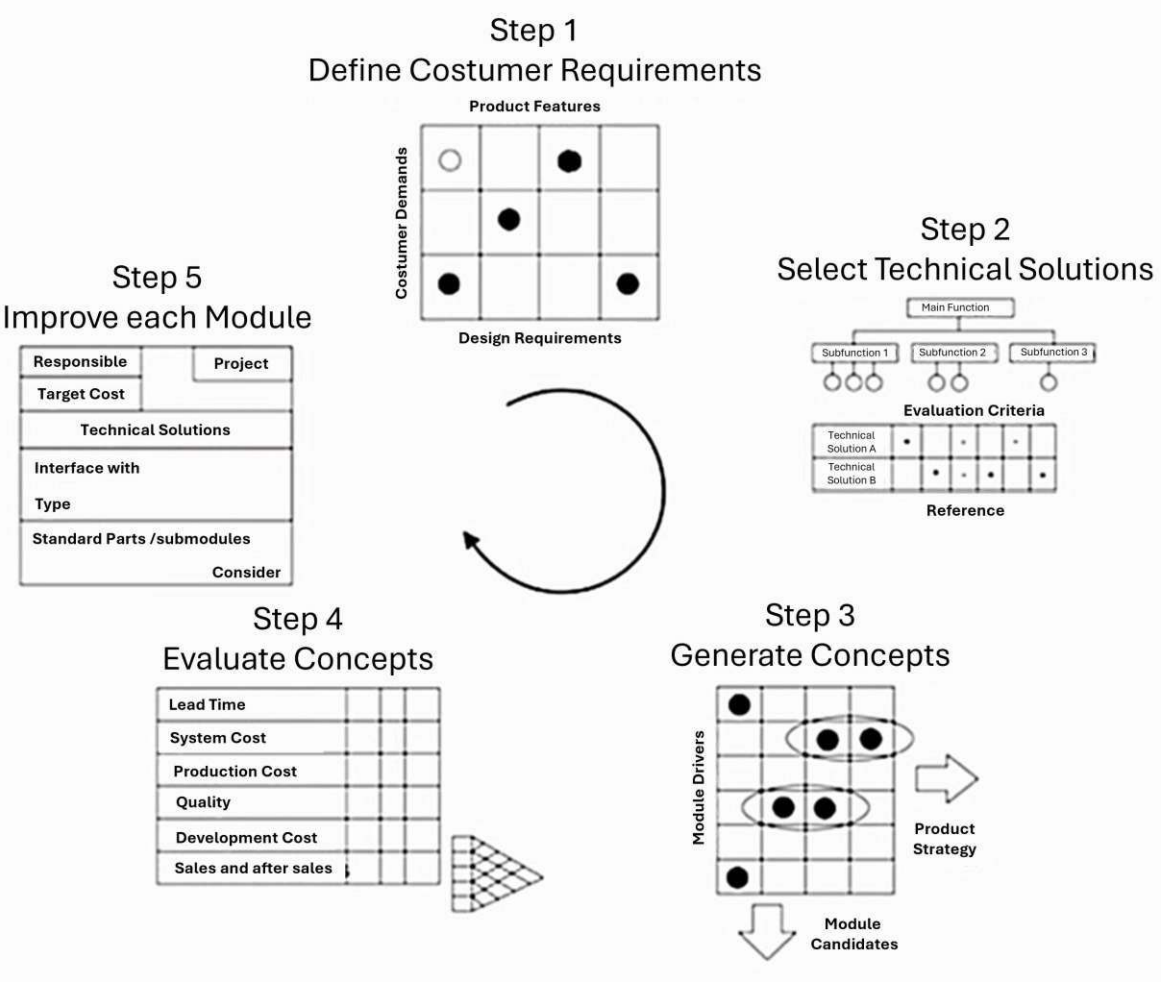

Figure 2.7: Modular Function Deployment 5 Step Working Principle, describing how to improve modules based on customer requirements [33], modified for readability.

assumed here, as these were already defined when the power plants were implemented. Once the structure has been established, new requirements can only be incorporated into the existing system using MDA. In addition, the transformations are a very complex process that requires special tools that go beyond the scope of this work [31].

2.2.3 Modular Function Deployment

Modular Function Deployment (MFD) [32] offers a cyclical approach to modularize systems and continually adapt them. This methodology offers a very qualitative approach, consisting of a five-step process with the target of incorporating strategic properties into the system design via modular drivers. The process is more management oriented than engineering-oriented, as it does not take system boundaries into account but can be combined with other processes in system design.

The five-step process flow shown in Figure 2.7 can be explained as follows [14]: In Step 1, customer requirements are defined and linked to product properties. For this

purpose, a Quality Function Deployment matrix (QFD) based on quality management is used. Product requirements are incorporated into the modular system design right from the first step.

In the next step, various technical solutions that enable the realization of customer demands [33] are developed from Step 1. A Design Property Matrix (DPM) is used to document which product properties influence which technical solutions. This connection has both a qualitative, as well as a quantitative aspect. A strong link means that a feature of the product causes a particularly large amount of variance in the linked technical solution. The DPM can be used to identify which technical solutions vary due to the same or similar product properties.

Until this step, there is no emphasis on modularity as this development process can be applied to any system. This happens in the next step by defining modules by creating a Module Indication Matrix (MIM), which evaluates the technical solutions found with the modular drivers by distributing scores depending on how strongly the solution favors the individual modular drivers. Modules can now be separated for reasons of technological evolution, generalization, or aftermarket activities. The optimal number n of modules is estimated as $n = \sqrt{N}$, where N represents the total number of components [34]. The scores for the individual drivers are then added together and the highest-ranked ones are considered module candidates. Components with the same modular drivers are good candidates for grouping into common modules. Modules with weaker scores and similar module drivers can be analyzed for combinations. Then the modular architecture is explored on the basis of the interfaces to provide information about the module arrangement. By sensibly dividing the modules into their modular drivers, it is then possible to evaluated the modular drivers, influence them by replacing modules, and create an understanding of what needs to be adapted when the system changes over time.

In the final step, the modules are optimized and specified in order to meet all requirements for implementation. To do this, software and hardware interactions must be understood and adapted to standardized implementation procedures. Here again, it is important to take the development needs into account so that modules can be quickly linked to new additional modules.

Each module therefore has 3 clearly assignable elements: a strategy, a function, and an interface. The methodology offers a clear assignment to classify modules from system goals and assess the influence of different solutions. However, this methodology is incomplete [35]. The methodology is very subjective, lacking objectivity and prescriptive rules to decide how to integrate components. It only describes which modules would be practical to combine, but does not provide any basis for whether modules can be combined at all and what other challenges and dependencies arise for the architecture as soon as modules are combined. In addition, the established matrices are used to map connections between functionalities and modular drivers, but the method only depicts 2-dimensional connections and does not provide any qualitative information about whether modular drivers also influence each other within functions. Weighting the relationships is very subjective, as there are no specifications as to how the weighting distribution can be extrapolated across different requirements in a standardized manner.

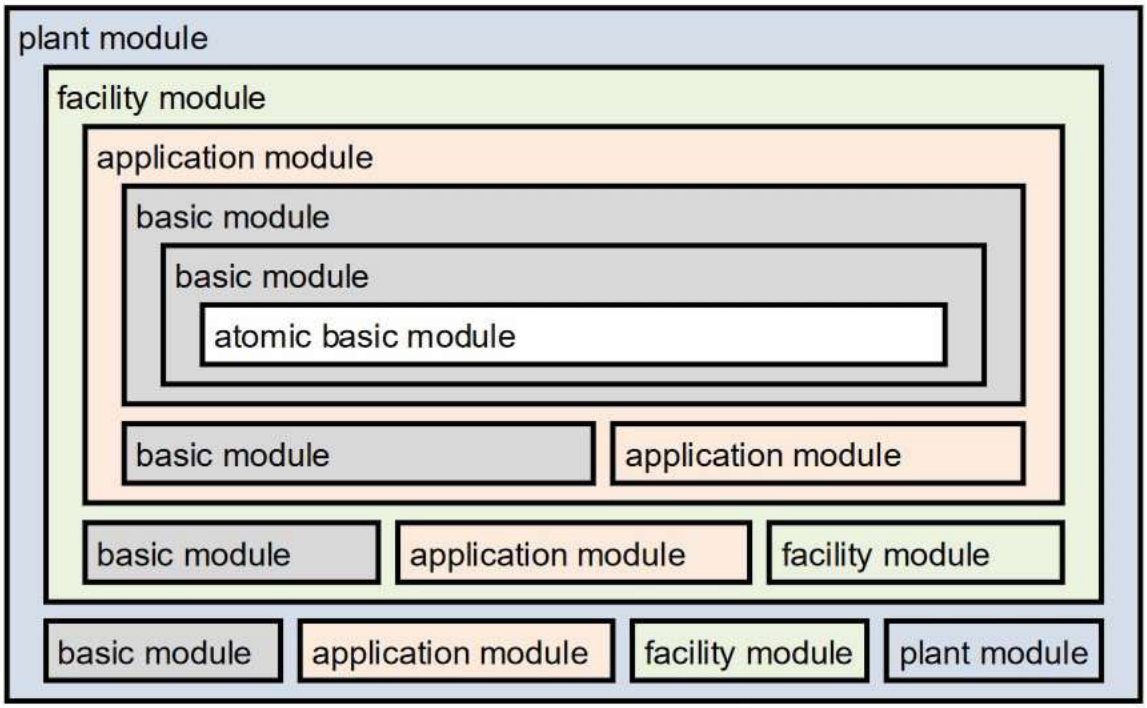

Figure 2.8: Architectural Layers of Automated Production Systems [12].

2.2.4 Architectural Layers of PLC Controlled Systems

Three methods are presented above on how systems can be constructed and designed based on objectives. Though, the question still remains on how existing systems that are not modular can be used and redesigned into a modular form. From a control engineering perspective, hydroelectric power plants are highly nonlinear and large-scale multiple input, multiple output (MIMO) systems for which no approach can be found in the literature that embeds the control in simpler, independent modular structures. In systems, such as cyclical production systems or process engineering plants, in the pharmaceutical, chemical, biotechnological, and petrochemical industries, the literature on plant components of production lines offers a lot of information to find separate structures from each other. It should now be considered whether the modularization of these structures can also be applied to hydroelectric power plants.

To do this, the hierarchical structure of these nonmodular systems must be analyzed first. To this end, [36] identified three levels of granularity in automated production systems. These result in basic modules, application modules, and facility modules. An analysis of seven German companies and their software architecture in the machine and plant manufacturing industry for PLC code, shows that the three levels can be further extended via a plant module, an atomic basic module, and flexible interconnection of the modules [37].

Figure 2.8 visualizes these modular abstraction levels and their possible level of detail. It shows that plants are normally designed top down until an application model with subordinate basic or additional atomic basic modules is created. Described in a top-down approach, the plant module describes an entire plant with all its components

such as storage, production units, or reactors. These modules represent facility modules, which, when interconnected, represent the entire process flow of a system. All facilities that support these facility modules are assigned to the underlying level of the application modules. Their function is, for example, to supply or remove material or to initiate conversion processes via mixing devices, energy conversions, or can take on control tasks. The last modular level of [36], basic modules, describes all the facilities that are controlled to trigger the applications, such as motors, valves, and sensors in measuring devices. The last level, which is added by Figure 2.8, the atomic basic modules, now provides a precise description of these basic modules by clearly breaking down the internal functional principle and interfaces into indivisible modules.

Looking at the modular analysis, it becomes clear that top-down, every higher-level module can contain a large number of lower-level modules, regardless of its level. Only atomic basic modules can be contained purely in basic modules, as these cannot be separated from parts of the system. The higher the level, the less flexible the modules are when it comes to reuse, as there are increasingly strong application dependencies. Basic modules represent controls with a high degree of reuse, such as motor controls or sensors and actuators. These modules occur in large numbers in systems, which means that they must also be structured accordingly.

The advantage of this analysis is that it results in a transparent arrangement of systems in which the functionality can be clearly divided into modules. It offers a structured approach to how existing systems are constructed and, therefore, provides a starting point for the gradual decomposition into its functionalities and interfaces to hardware modules in order to enable ever greater levels of detail using reverse engineering.

One difficulty for a control system is that each process step in the system analysis according to this structure is linked to its own modules, which results in a large number of sub-modules. The large number of sub-modules makes relationships difficult to understand before basic modules are created. The basic modules then have to be combined into application modules. It can be a complicated task if several process steps access the same basic modules and duplication of modules may not be allowed for reasons of economical process management.

A standardized method for building modular system structures is provided by the guidelines "VDI 2776 Process engineering plants - Modular plants - Fundamentals and planning modular plants" [38] and "VDI/VDE/NAMUR 2658 Automation engineering of modular plants in the process industry" [39]. This sets out how the subdivision of the units in process engineering plants must be structured in order to create a modular structure with independent units including subunits.

In VDI 2776, the plant structure is hierarchically assembled into a four-layer model of components, which is structured similarly to the one of [37]. From a process engineering perspective, the physical form of the units is divided into Modular Plant (MP), Modular Process Units (PEA), Modular Functional Units (FEA), and components (COMP) [40]. In VDI 2776, the level of the MP corresponds exactly to the previously broken down plant module, the PEAs to the facility modules, which can function autonomously as they can independently carry out all the necessary functions and process steps. FEAs correspond to the modules of the application layer and the COMP to those of the basic modules, which cannot be further separated from an automation engineering perspective.

Although the analysis of the system architecture of [37] only represents a confirmation of the guideline at first glance, the units of VDI 2776, offer a clearer hierarchical assignment in their subdivision. Here, communication from a hierarchically higher level can only ever take place to the level below it. There can be no communication of information between layers as shown in Figure 2.8. Previously, it was also possible to place a basic module at the facility module level and separate it from its area of responsibility. In VDI 2776, this module or COMP must now be assigned to an area of responsibility and thus functions before it can be included in the overall modular system. The way these modules communicate with each other is laid out in VDI/VDE/NAMUR 2658.

A distinction is made between three module variants [41]:

Autonomous modules: These are modules that are closed units and communicate autonomously with the outside world and react independently to input and output flows. They have their own decoupled automation units that can only be addressed centrally. They react to entry events and always carry out the same process steps according to their implemented logic.

Integrable modules: Integratable modules are not only integrated into a system in terms of energy or material, but also in terms of automation via a standardized interface in the Process Orchestration Layer (POL) explained later. Like autonomous modules, these modules have their own automation units, but can be addressed via the POL interface, and the process flow can be adapted individually. This enables automatic integration into a control system, for which the standard MTP can be used.

Modular modules: These are modules that can be divided into modules again. This division ensures the variability of being able to change functionalities individually and in isolation. The overall process remains the same, but interchangeable modules mean that the systems can be adapted or maintained with little engineering effort without having to reassess the overall process.

The VDI/VDE/NAMUR 2658 describes the engineering of automation technology as a guideline and explains the definition of interfaces and functions in order to make modules in the POL configurable and linkable. The POL represents the level in which the PEAs can be configured before commissioning is dynamically combined, as well as systems such as the HMI or other services. The difference from conventional programming of the control system is that the configuration is not part of the programming but can be individually adapted on a regular basis. The PEAs are read into the POL via MTP, which creates a data model of the PEAs to allow access to the functions contained therein.

2.2.5 Design Structure Matrix

As already mentioned, a large number of modules can arise during the analysis and implementation of large systems. A large number of modules also means a large number of connections and interfaces. A strategy is required early in the design phase to minimize this level of functional and technical complexity. An effective method for this is the use of a Design Structure Matrix (DSM) for clustering [42]. The DSM allows

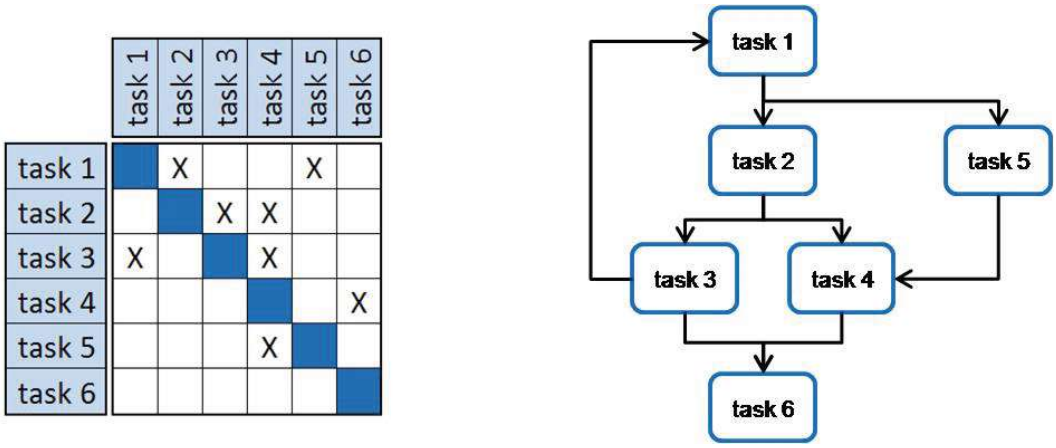

Figure 2.9: Design Structure Matrix with Corresponding Process Flow, interfaces from the process flow are registered in the cells of the matrix [44].

visualization, modeling, and analysis of the networking of elements in architectures and processes. This variable application form is used in this work to simplify control processes in a modular form for hydroelectric power plants.

A DSM is created in square form. All tasks or elements of a system that are carried out one after the other for a process are plotted in the correct order on the axes. The relationships between the tasks are then entered into the cells of the matrix. The diagonal remains free; the elements above the diagonal, are dependencies in the direction of the process flow (feed-forward), and the elements below the diagonal are feed-back dependencies. Mirrored versions of DSMs can also be found in the literature [43], which is why it is important to know not only the DSM but also the process. Figure 2.9 represents the structure of a binary DSM in the context of a process flow. This results in a from-to dependency in the cells.

The DSM's objective is to minimize the number of feedbacks and simplify the process through restructuring and grouping [43]. Neighboring cell entries close to the diagonal indicate elements that can be grouped well. A better and reproducible way to find groupable elements however is to weight the relationships. By weighting the cell entries, algorithms are designed that differentiate between different strategic and technical viewpoints much more precisely for clustering. In addition, clustering no longer depends on the arrangement of the elements on the axis. In large systems with several parallel flows, this is not possible anyway. Sequencing and restructuring the axes is then no longer necessary for further steps.

An example of a complete design draft up to the cluster algorithm is given in [35]. Here, the previously explained MFD method is used to create a weighted DSM for the biofuel production process. In this weighted form, which this time does not decide between the information direction of the elements, both strategic and technical influences can be formulated in separate DSMs. By omitting the process direction, pure upper triangular matrices are the result. Strategic assessments of whether elements are beneficial to be clustered later or always separated can be achieved using manually set high weightings. A combined DSM can be created by prioritizing strategic or technical goals. This is used to feed an algorithm that iteratively clusters elements and solves an

optimization task using a cost function to determine the best combination of elements.

The algorithm works as follows [35, 45]:

1. Each element is placed in its own cluster. This results in n clusters with N elements. In addition, a check counter $q1 = 0$ and a stability counter $q2 = 0$ are initialized.

2. The coordination cost C is calculated by adding the cost C_I for candidates that are inside a cluster and the cost C_E for candidates outside a cluster.

3. The check counter $q1$ is raised by 1. A random element is selected and bid values B_n are calculated for each previously determined cluster set.

4. A random number between 0 and 1 is generated. If the random number is larger than $1/pN$ with p being a selectable value (typically 2) and the highest or second highest bid candidate still being below the max cluster size, the algorithm proceeds at step 5. If both clusters are full, $q2$ will be raised by 1 and the algorithm jumps back to step 3.

5. If the random element from step 3 is already a candidate in the highest or accordingly second highest bid cluster then $q2$ is raised by 1 and the algorithm jumps back to step 3 to find a new random element. If not, the coordination cost C_temp is calculated temporarily for when the candidate becomes a member of the cluster.

6. If C_temp is better than the current coordination cost C, $C = C\_temp$ and the clustersets are updated, and if the element was a sole member of a cluster, the total number of clusters n is reduced by 1. $q2$ is reset to 0. If C_temp is worse than C, $q2$ is raised by 1 and the clusters stay unchanged.

7. If $q1$ is smaller than a set counter value for testing new candidate elements repeat at step 3, if not and $q2$ is smaller than a counter value for repeating the algorithm without change set $q1 = 0$ and repeat at step 3. If both are higher, the algorithm stops.

Since this is a random search algorithm, different results can arise when it is run multiple times, depending on the system. Therefore, only the top 1% of total cost solutions should be examined by the designer [35]. The results are influenced by the parameter settings. However, no general approach on how to set those parameters is formulated. In step 4 the random number acts as a simulated annealing method avoiding getting stuck in a local optimum, however [46] states that the simulated annealing can produce a worse final solution, which is why the currently best solution found must be saved.

The advantage of the DSM is that it is very engineering-oriented and offers directly practical solutions for clustering modules.

The distinctive aspect of the development described in [35] is its integration of two methodologies, MFD and DSM, to examine clusters within a modular system. In this work, it is crucial to explore not only the application to system architectures but also the process structures of power plant controls. Here, it is not necessary to set up an

entire DSM using MFD via strategic weightings, but it is explored whether a DSM can be found that can be simplified with this algorithm structure.

It is worth mentioning that the literature includes methods for agglomerative hierarchical clustering of modules and arranging them into a dendrogram structure [47]. However, these mainly deal with the clustering of rigid structures such as static constructions or architectures and evaluate optimal clustering based on similarity of the input data. They are less specialized in solving classical optimization problems, as they do not pursue optimization goals, but are rather explorative methods that group data. In rigid structures, a detailed analysis is often not required, as the focus here is on assignments, and data such as distances or similarity of labels provide sufficient data for an assignment pattern.

For the grouping of control processes, on the other hand, extensive analyses of control modules have to be created, the dependencies of these modules and their arrangement broken down for each requirement pattern of the entire system and a comprehensive analysis of the reactions provided and evaluated. This amount of data can then provide sufficient information for an allocation pattern of clusters. However, from this point of view, it is too costly to provide such a detailed breakdown that is aimed purely at function and practical reusability. In addition, extending a system requires a completely new evaluation, which is not in line with a simple reusability.

CHAPTER 3

Designing a Modular Control System for River Power Plants

After a breakdown of various methodologies for the modular construction of systems, a methodology can be derived that can be used for the modular design of controls for hydroelectric power plants. In this chapter, this approach will be examined and implemented in more detail for a river power plant. Since this is about the construction of a structure that is not well covered in the literature, the procedure must be easy to understand and roughly formulated in order to enable future work to be continued easily. It is structured in a way that the structure of a basic run-of-river power plant is broken down first according to its layout. A very clear and easy-to-understand method for this is a combination of the structural layers of [12], which can be filled in, according to the information from [6, 11]. The advantage is that the functional areas allow for an immediate assignment and a coherent picture of the procedures emerges. This method is able to reverse engineer, which means that unlike MDA and MFD, requirements do not need to be defined and evaluated in advance. With these methods, systems have to be designed from scratch, which involves a considerable amount of work and the potential for errors. After the components are broken down, a fully automated modular control system must transfer the power plant to the desired states using defined functions. MDA uses different model levels to include new functions step by step in an existing system, while MFD uses a multilevel evaluation scheme to determine which functions in which form and modification meet the requirements of an existing system best. For the basic construction of a new structure, however, the question is not one of inclusion, but of merging functions. The operating cases of a power plant specify control functions that can be clearly presented in modules in UML diagrams. What stands out is that similar structures can be integrated into the activity diagram using decision nodes and operational cases can be treated as states. The three operating modes turbine operation, automatic startup, and automatic stop are intended to serve as states for a functioning power plant in this work. After the structural interconnection of the operational flows, a large number of modules are created that can be clustered using a DSM, similar to Section 2.2.5. Finally, the clustered result is implemented in the form of a flow control on a model and is examined for its behavior and the reusability of module variations as

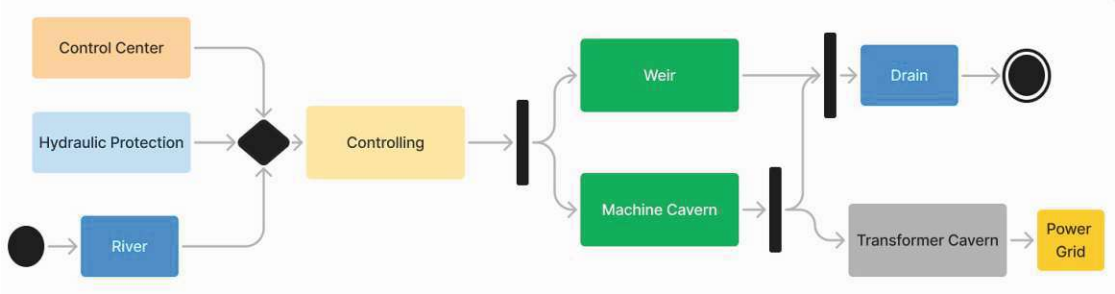

Figure 3.1: Plant Module UML, physical connection of different power plant areas.

well as its scalability.

3.1 Plant Module

As already described in Section 2.2.4, the plant module offers the highest and therefore coarsest level of abstraction of a system. In the case of a hydroelectric power plant, where the control structure is to be set up, this consists of the functional areas described in Section 2.1.1 [6]. Although the plant module basically only represents a breakdown of the components, when setting up a modular control structure it is helpful to map a process flow here right away in order to directly map a functional order for later steps with smaller modules. This also breaks down the operating principle of a plant more and more precisely and offers understanding to staff who do not have to understand the plant in detail. For a run-of-river power plant, the functional areas are the machine cavern, transformer cavern, the weir area, the hydraulic protection, the control center and an outsourced control unit. The area covered by the protective equipment is correspondingly included in the hydraulic protection area and assigned to the individual caverns. Here, all the units that can be represented in the module are declared as objects because they are technical facilities that physically exist. The *Controlling* module can cause misunderstanding, but here an outsourced control unit is meant as a physical component on which the implementation of the control is implemented decentrally. These units combined in an operational flow representation using UML, the basic functionality can be abstracted and results in Figure 3.1.

The function of the hydroelectric power plant is shown here starting from the black circle to end at the outlined circle. The *Controlling* unit receives a selection from the measured inflow and water level of the river, the protective devices and manual inputs from the *Control Center*. The *Controlling* now provides flow specifications to the *Machine Cavern* and the *Weir* systems, based on measurement curves, empirical values, and implemented automatics. The *Weir* and the *Machine Cavern* adjust their internal equipment according to the specifications received, which results in the drain from upstream water to downstream water. The energy is transmitted from the *Machine Caverns* to the *Transformer Cavern*, which feeds the electricity into the grid at the correct voltage level.

It is important to understand how a river power plant works in principle. These power plants try to generate as much electricity as possible during normal operation.

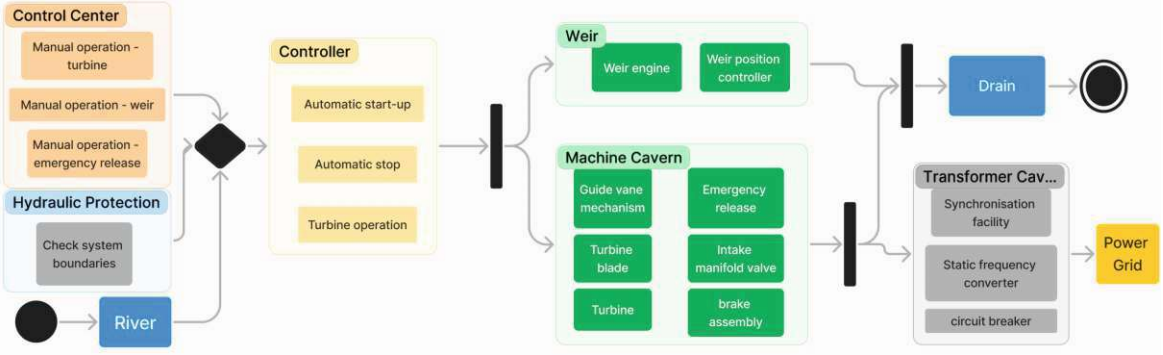

Figure 3.2: Facility Module UML, separation of the areas into its components.

The available water inflow is distributed across all running machine sets until their nominal level is reached and before any excess water has to be drained off via weirs. Another special case is the shutdown of the machines during flooding, when the level between the downstream and upstream water is too low to even start the machines.

3.2 Facility Module

The next level represents the facility module. In the same UML structure as before, this module provides an overview of all functional groups located in the functional areas mentioned above. As shown in Figure 3.2, these results in all sub-facilities, which can also have their own control to handle their supply facilities.

This subdivision into more detailed components not only provides an allocation of physical components to the functional areas but also provides an overview of the location of the basic functions. The *Control Center* module can be subdivided more precisely into several manual operating modes. From here, individual turbines, weirs, or emergency devices such as drains can be addressed directly. In its function as a pure monitoring device, the *Hydraulic Protection* provides the interface to measurement data and, like the *Control Center*, can influence the subsequent operational flow. The *Controller* has the special feature that it sits between the *Control Center* and the units to be controlled. It wants to bring the power plant to a target state fully automatically depending on the requirements of the received signals. The detailed implementation for adjustment of the modules for the target-actual comparison represents an auxiliary operation and can be carried out in subordinate modules. Like already mentioned before, this work tried to create a modular logic for the three process flows Automatic start-up, Automatic stop, and Turbine operation, located in the facility module of the *Controller*.

3.3 Application Layer - Operation Cases

At the level of the application module, all supply modules that the facility equipment needs to be able to perform its functions would now be broken down to its elements. Since publicly accessible resources in the literature no longer provide solid data and

the precise breakdown of all auxiliary equipment for power plants would not add value to the pure methodology of modular control design, an application module is established that breaks down control modules in the same UML structure, based on the operating cases explained in Section 2.1.3. Meaning the application modules shall be depicted as functional control modules interacting with the facilities, broken down in the facility module, performing the behaviour of the operating cases. The advantage of this application module interpretation from a software design perspective is that interfaces can be drawn in directly and the planning for implementation can begin. Modules are examined for their effectiveness, signals are analyzed for their accuracy, and it is clear whether each module has all the interfaces it needs. This not only enables visualization of resource allocation and processing as before, but also a control flow of how units react to deviating signals. It is important to find a control structure that is independent of the scalability and size in its form. Scalability is achieved purely by duplicating modules and equipment, and modules arranged above them must be able to react independently to subsequent changes.

Three operating cases are selected, on the basis of which a functional run-of-river power plant can be created in a simplified form. This results in turbine operation, automatic start-up, including synchronization, and automatic stop.

3.3.1 Turbine Operation

Turbine operation is defined as when a machine is synchronized to the grid and automatically regulated to maximum power. To understand turbine operation, it is necessary first to explain the function of what controls a power plant when a turbine set is already in operation and how it can be ensured that all parameters remain in balance. The basic element of a river power plant is the flow control. A continuous flow of water comes from the river, which is converted into electrical energy as efficiently as possible. The power plant is designed so that each machine set can handle a maximum power requirement and the machine set is multiplied across the width of the river so that the average flow, measured over the year, can be processed purely by the machines. The rest of the river width is designed with weirs, which are activated only in the event of a flood or in other cases in which unused water needs to be discharged. How machines automatically move to the optimal operating point and weirs assume their desired position so that just as much water flows into the downstream water as flows into the upstream water is, among other things, carried out by the turbine controller and is worked out in Figure 3.3. Control modules address facilities in a coordinated manner in the sense of the turbine controller, and its function is explained in the following paragraphs.

Looking at the upper part, this involves evaluating the inflow and calculating the desired outflow. Inflow cannot be measured directly and therefore must be calculated from the water level at measuring points upstream using a *Track Water Level Curve* module. Deviations from the current water level are passed on, in meters, to the next module *Calculate Inflow*. This module converts a change in water level `d_m` into a change of inflow `Q_meas` measured in cubic meters per second. Location-dependent measurement curves and diagrams are used for this purpose, which vary depending on the river and its water level. The *Calculate Desired Drain* module is used to calculate

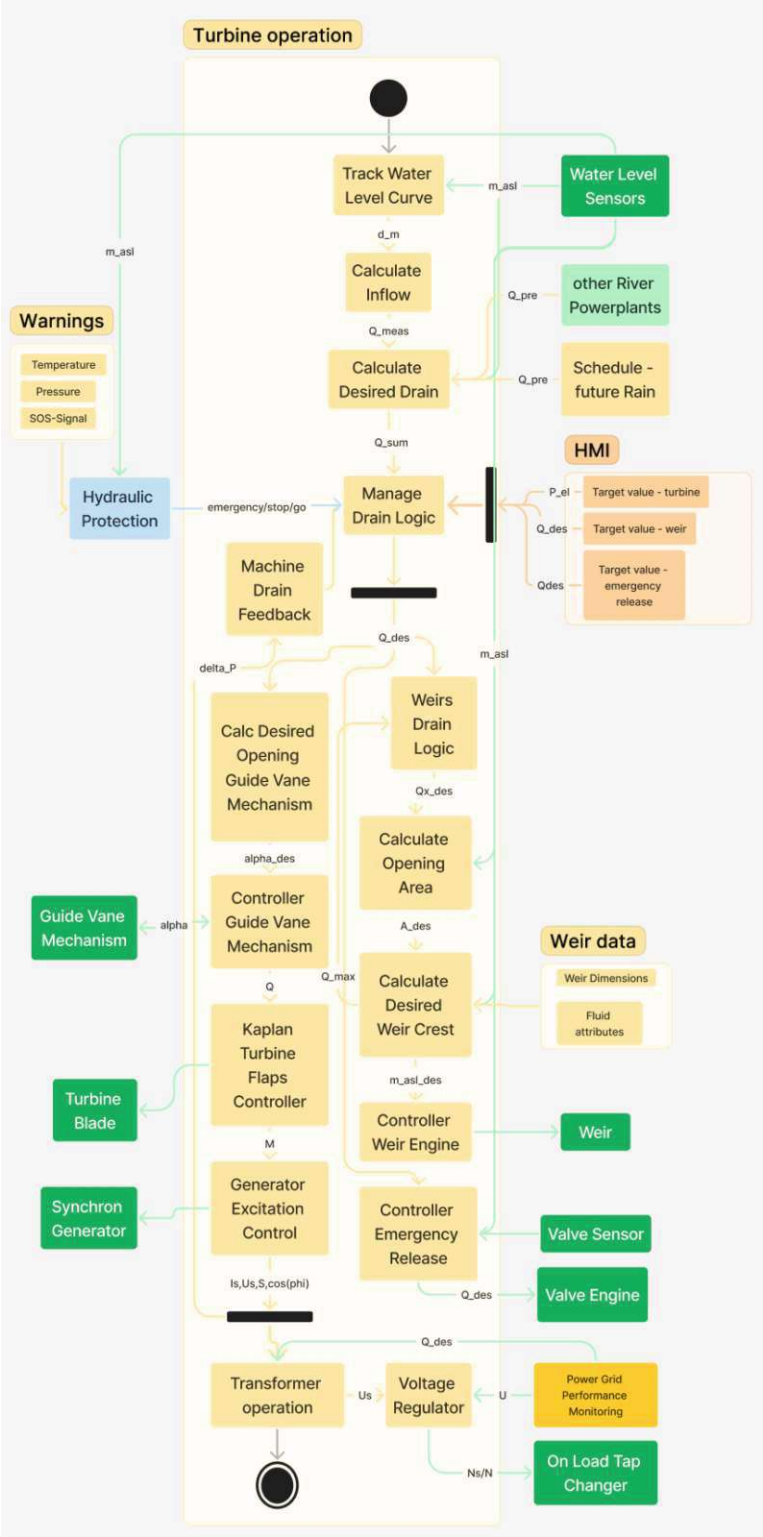

Figure 3.3: Turbine Operation, process flow for the functional procedure for energy generation.

how much water should be drained from the power plant to maintain a constant water level upstream. This module combines the water level measurements explained above, the measurement data of the river course or road maps of higher-level power plants, and the measurement data from weather forecasts to create a `Q_sum` value. Road maps of higher-level power plants mean that due to strategic or economic reasons these power plants could outlet more or less water than normal, so other plants need to react coordinated.

A key module that is interpreted in the plant module and facility module UML as the decision node is the *Manage Drain Logic* module. This module not only selects how much water shall be passed through machines or outlets but also has an implemented criticality ranking system, which input values are to be approached preferentially. Its inputs consist of the calculated inflow mentioned before to keep the current water level, control room target specifications defined in the *Human Machine Interface (HMI)* or emergency signals from protection facilities. As output the module delivers a desired discharge value `Q_des` to one machine, one weir, and one emergency release in the unscaled case. For control inaccuracies, the machine returns its drain, based on its power output to the drain logic. Since all machines try to work as efficiently as possible with their provided water input and the direct convertibility between flow and power, all further control features can be outsourced to later modules. The *Manage Drain Logic* contains information on the maximum throughputs of its connected machine sets to know how much water shall be drained unused. This information is not necessary for the weir facilities since a subsequently *Weir Drain Logic* module is intended. This module receives the maximum drain `Q_max` each weir can handle as input values.

As already mentioned, weirs are facilities that open almost exclusively in flood or machine maintenance scenarios, which means that they are rarely used. In order to keep all weirs in regular operation and to minimize long downtimes for weir maintenance reasons, a separate logic is required that enables alternative controls without a major programming effort. Depending on the implementation, the *Manage Drain Logic* module thus supplies alternating target flows `Qx_des` to all of its weir systems, shown here with just one weir. The weir itself is given a target flow value and, in combination with the water level, can calculate how large the opening area must be to discharge the corresponding `Qx_des`. The value of the opening area is then passed on to the *Calculate Weir Crest* module, which contains the exact dimensions and specifications of the weir in question. This module decides whether it is a weir with an underflow or an overflow and to which position the weir crest must be moved. The water level is relevant as input as, in the case of an overflow the "dead area" of the upper edge must first be covered up to the water level, before water is effectively discharged. Once the position of the weir crest has been determined, the target value in `meters_above_sea_level` is transferred to the motor control module and the position of the motor is controlled from this. It is not necessary to feed the position back to the *Manage Drain Logic* here, since in the event of insufficient discharge, the headwater level automatically rises and this is then interpreted as additional inflow, which is automatically reinterpreted as a larger opening of the weirs. The same applies to a weir that is opened too much and the water level falls. These inaccuracies can be ignored due to rigidity and many other uncertainties such as sediment shifts in a power plant. The *Controller Emergency Release* can be seen as mentioned in Section 2.1.1 as a beam deflector or some sort

of external weir that has either the same structure as the weir, just explained, with different dimensions. In the application module in Figure 3.3 it is depicted as a simple valve control for simplification.

For the machine control branch the most essential part is, that almost all generators in a river power plant are built as synchronous generator. Therefore, the rotation frequency of the directly coupled Kaplan turbine is constant to the net frequency of 50 Hz in relation to the number of pole pairs. Like the speed, the voltage is also fixed in value. The main parameters to manipulate in net-synchronized mode are the momentum on the turbine and therefore on the shaft of the generator and the exciting current. The turbines momentum is changed over the *Guide Vane Mechanism*, that opens water flow and directs it to the blades of the turbines. The idea behind the positioning of the turbine blades lies behind the twist-free condition. Meaning the blades are set in a position where water can pass through without being stopped. The position controller therefore is fed with the inflow `Q` and provides the resulting momentum to the *Generator Excitation Control*. When the machine set is not connected to the grid the excitation manipulates the voltage of the generator, however when it is connected to the grid the voltage is fixed and the excitation controls the reactive power behavior of the machine and therefore the operating point via $\cos(\phi)$. The last part of the turbine operation sets the control of the transformer during runtime. With a *Power Grid Performance Monitoring* the *Voltage Regulator* can set the transformation ratio for the *On-load tap changer*. This results in a constant and load-independent power supply.

3.3.2 Automatic Startup

The second operation mode represents the Automatic startup mode. In this mode a machine set not synchronized to the grid needs to start its operation until it is synchronizable. Further, all conditions for a safe connection need to be checked before the circuit breaker is closed. Since this operation mode must be compatible with the turbine operation the structure of the whole operation is kept the same, just the machine part is changed for its new requirements.

For a synchronous machine set to be synchronizable the four synchronization conditions:

1. same frequency,
2. same voltage,
3. same phase sequence, and
4. same phase position

need to be fulfilled. Fist, the machine is brought to speed. Therefore, the turbine is flooded with water. The closing valve that completely shuts off the water flow is the intake manifold valve. It must be opened before any other facilities. Next it needs to be determined where the power supply for operating the facilities comes from. Basically there are 3 different options, a foreign network, the local power grid controlled over a static frequency converter or a self supply, normally ensured over a local redundant

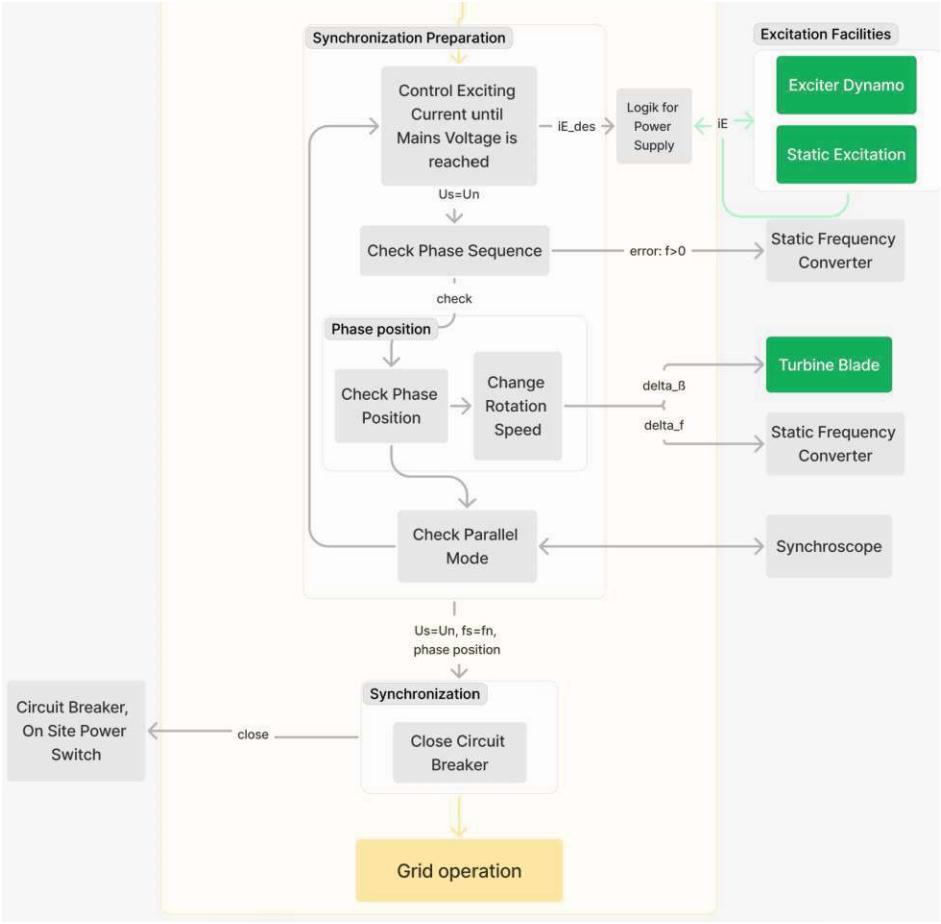

Figure 3.4: Machine Synchronisation, necessary synchronization steps for a synchronous generator.

smaller excitation dynamo. The starting process of a machine now depends on the option of the power supply and its usable facilities.

When controlled over a SFC, the excitation frequency and therefore the machine's rotation speed can be controlled directly. The turbine blades need to be brought into a default position, the SFC sets the target frequency depending on the pole pairs, and the guide vane mechanism controls the momentum over the water throughput to the turbine. During a start without direct control of the frequency `f`, the rotation speed must be controlled over the water throughput. Therefore, a sophisticated control between guide vane mechanism for water amount and turbine blades for water deceleration keeps the machine at the required speed. Finally, the nominal rotation speed is reached and, therefore, the first synchronization condition is fulfilled.

Figure 3.4 illustrates the final stage of the generator synchronization. The second condition to approach for same voltage of grid and generator is controlled by increasing the exciting current. Depending on the current, the stator voltage increases in a non-synchronized machine until the grids voltage is reached. The switching control of the transformer devices to supply the required current is visualized here in a simplified manner via the *Logic for Power Supply* module. After the stator voltage `Us` meats the necessary grids voltage `Un`, the phases need to be checked for their correct sequence.

This point is only relevant when starting up the first time, as this would be the case of incorrect wiring, but it takes no extra effort to carry out this check at each synchronization. Finally, when checking the phase position, it is necessary that the phase positions of the generator and the grid overlap as much as possible. When the circuit breaker is closed, the generator phases will align abruptly with the grids phase positions. The farther away the phase positions are from each other, the greater the jump will be and the machine will feel a mechanical jerk that can lead to damage. Therefore, the circuit breaker must close at the right time. If the turbine and the grid run exactly at the same speed, the phases will stay in the same offset. To minimize the offset, the rotation speed of the turbine needs to be changed slightly so that the phases slowly align each other. That is reached either with changing the turbine blade positions and slowing down the water at same throughput or changing `f` directly over the SFC. In the moment they are aligned, the *Synchroscope* briefly checks over voltage, phase position, and frequency and if they are in range, directly gives the approval to close the *Circuit breaker*. Additionally an *on site power switch* is closed, which ensures the power plant's self-sufficiency. The machine is now in grid operation and can be controlled as in Section 3.3.1.

3.3.3 Automatic Stop

The last operation mode necessary for a simple functional water power plant control is to bring a working machine to a standstill. Since this is a rigid system, every plant needs to have two different options, an emergency stop and a normal stop. An emergency stop means that there has been a failure with the machine or related, and the machine needs to be taken off grid mode immediately. The water running through the turbine cannot be stopped immediately due to its inertia. If the machine is taken off the grid, the load is taken off as well, and an uncontrolled machine would accelerate since water is still pushing on the blades. Therefore, there needs to be another control path to switching off a turbine normally where water can be reduced continuously. Figure 3.5 shows the two different versions of the emergency and normal stop.

As mentioned, during a normal stop there is enough time to reduce the water flow and to shut the valves one after the other. To reduce water flow, the guide vane mechanism, the water dosing device, is closed continuously. When the water has come to a standstill or there is no longer a danger that the turbine accelerates over its dimensioning, the automatic switch can be opened. Illustrated is the slowest step-by-step process, in which the machine waits until the guide vane mechanism is completely closed and the machine is in idle operation. Theoretically, the system is stable in this state, the machine continues to rotate at grid frequency because it is still connected to the grid. However, it is recommended to pass this stage as fast as possible, since the turbine is not, like mentioned later in Section 5.2 in air and wastes energy. A more efficient but more complex way is to check the system boundaries parallel with the *Controller Guide Vane* and open the *on site power switch* and the *circuit breaker* when safe. The *Automatic Switch* module is a control module for the *on site power switch*, which is a circuit, connected to the facilities for the self-sufficiency of the plant while the machines are running.

The *Automatic Switch* always needs to be opened before the *circuit breaker*, that is

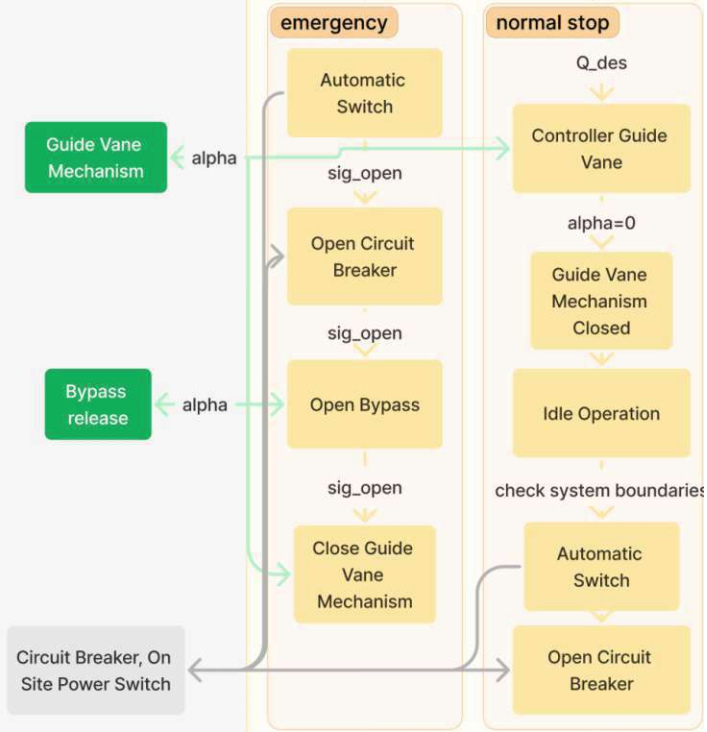

Figure 3.5: UML Stop Paths, a separation for normal stop behaviour and emergency stop is needed.

why in the emergency stop, the first control parts are the *Automatic Switch* and then *Open Circuit Breaker*. Then the *Bypass release* is opened to let the water column flow past the turbine. This is especially the case for diversion power plants or power plants with longer tube systems in front of the turbine, such as storage and pumped storage power plants. Afterwards or in parallel, the *Guide Vane Mechanism* is closed.

To this control point, independently of the stop mode, the machine is no longer synchronized to the grid and the water input is closed. However, the turbine could still be rotating and the generator is still excited. In order to reduce the magnetic field controlled in the generator, the excitation is reduced with a specific control module. If the machine has not come to a standstill to this point, a break assembly reduces the rotation frequency with a brake control module. When the turbine is stopped and the generator is demagnetized, the machine is brought into a default state with a specified flap position and a closed *Intake manifold valve* as a safety lock.

As in the other operating cases, the flow of water to the machine is controlled via the *Manage Drain Logic* module and monitored via the *Machine Drain Feedback*. Therefore, it makes sense to control the remaining water drainage devices in parallel as like before.

3.3.4 Combination of Operating Cases

As all the desired operating cases of an entire power plant have been set up for different river inflow rates in a compatible form, they can be combined with little effort. Every case starts with an evaluation of the inflow. The control of the weir systems is carried out

continuously via varying outlet specifications of the Drain Logic, so it is independent of the states of the machines and retains its structure in the combined form of the operating cases. The machine control, on the other hand, must be combined via a separate *State Module*, as each machine must be set to an operation case in a coordinated manner. The authorization of which control path is enabled is therefore the responsibility of this module. The decision on which case the assigned machine set should take results from input signals from the *Manage Drain Logic* and hydraulic protection devices. If the inputs are outside of a suitable operating range, a shutdown is initiated. If a valid input is present, the machine can be automatically started.

It is already noticeable that many modules access the same hardware, and the overview of the entire system is already hard to understand due to the numerous interfaces between the modules. However, the best modular architecture is one, in which each element of a product can only be assigned to one module or, in this case, hardware should be controlled by one module and there are therefore as few external interfaces as possible. Therefore, a clustering similar to the approach of [35] explained in Section 2.2.5 is beneficial.

3.4 Clustering

The broken-down structure in Section 3.3 is now the simplest possible representation of a functioning power plant, which only takes into account the operation, the turning on and off of the machine systems, and the auxiliary equipment to discharge the excess inflowing water. However, even in this basic water power plant module, there are 46 control modules of varying size and complexity that have to be implemented via PLCs. The advantage of modular structures is that they can be easily designed in a reusable form. If modules are strongly linked to each other, either because they occur exclusively in bundled form or simply because they must always be executed directly with each other from a critical safety point of view, clustering is advantageous. Another reason why clustering is needed is that the design of similar systems may result in a large number of modules in the future due to module universality. Control modules need to be universally designed to minimize the reusable programming effort of similar systems and to be as usable as possible. As already mentioned, this universality leads to a loss of optimization because it can happen that a functionality has to be broken down into smaller sub-modules. These subdivisions lead to more code and, therefore, more modules that need to be bundled together retroactively. The advantage of larger clusters is that knowledge of the details of the processes is not required. Similarly to the abstraction levels, it requires merging, which increases the understanding and the usability for the applicability.

An important prerequisite is to break down the modules as small as possible but as large as necessary before clustering. This means that modules that cannot be separated under any circumstances must be clustered in advance or defined as an existing module. During clustering, modules are bundled together on the basis of interfaces. This bundling takes place according to prioritization. If not taken into account, the order of control, which can be one of the most important aspects in a modular structure, can be lost. Therefore, a decision is needed beforehand whether a cluster algorithm should be

selected that focuses purely on interfaces or also on other aspects.

A solution that can be applied to the structure in a modified form is that of [35]. Here, a DSM is weighted and generated step by step by combining strategic and technical considerations using MFD. The resulting DSM is then clustered using a random search algorithm and various solutions are analyzed. Since no analysis is required to optimize power plants but existing power plants must be represented, strategic considerations for the methodology of the control structure created here, are not necessary. The resulting DSM can be generated directly from the combined control structure. With this, the verifiability of the methodology for valid results can be checked for hydropower plants, based on the state of the art.

3.4.1 Design Structure Matrix

The consideration of how a DSM is built from the combined structure must be viewed from different perspectives. There are modules that are designed in such a way that they must be adapted with every system modification or at least checked after final implementation. These are central modules that normally have several interfaces such as *Mange Drain Logic* or *Operation State*. Since these modules are modified regularly, they should not be too nested into submodules and therefore have a low weighting for clustering. A distinction also needs to be made as to how mission-critical signals between modules are. In the structure set up in Section 3.3, there is an exchange between measured values and status messages. A loss of data transmission for status messages normally leads to the process flow being delayed by one cycle, but a loss or incorrect transmission of measured values can lead to unwanted system behavior or damage. Another important part of the DSM is the arrangement of the modules. The arrangement of matrix entries as diagonally as possible ensures a good understanding of the represented process flow. In principle, there are no clear specifications as to how the modules must be arranged within the DSM. In systems with many parallel signal flows, it is also difficult to assign which module should be arranged first. There are always entries off the diagonal that provide the connection between the parallel signals.

The DSM allows to create and record two different relationships between modules. Entries above the diagonal and entries below the diagonal. The distinction can be used to differentiate inputs and outputs between modules, but they can also be used to make adjustments to prevent the explained central modules from clustering. Table 3.1 shows the DSM created for the flow diagram of the hydropower plant.

As can be seen from the DSM, there are modules that occur more frequently, as this is also desired in modular systems. The reusability results in consistent programming and separate clustering is possible. The 46 modules are plotted on the axes, the diagonal elements are not filled, as these would symbolize interfaces of modules with themselves. As a higher weighting of the algorithm described in Section 3.4.2 results in a higher cluster formation, measured values are weighted 2, which are weighted higher, than status messages weighted 1. Interfaces that belong to the same operating flow and therefore have a strong link are weighted 5. These modules are executed in series in the DSM. In the entries below the diagonal, entries of the modules *Manage Drain Logic* and *Manage Weir Logic* are weighted with -99 to prevent possible clustering. These two modules are core modules for the system behavior and therefore always need to be

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
track water level curve	x	2																																												
calc inflow		x	2																																										
calc desired drain			x	2																																								
manage Drain Logic			-99	x	2				2	1	2				2			2														2												
weir drain logic				-99	x	2																																						
calc opening area					-99	x	2																																						
calc desired weir crest							x	2																																				
ctrl weir engine								x																																				
ctrl emergency release				-99					x																																			
operation state				-99						x	1					1		1					1				1																		1
calc desired opening guide vane				-99							x	2																																
ctrl guide vane												x	1																															
kaplan turbine flaps controller													x	1																													
generator excitation control														x	2	2																				2								
machine Drain Feedback				-99											x																												
transformer operation																x	2																											
voltage regulator																	x																											
ctrl guide vane				-99	-99													x	5																								
guide vane closed																			x	5																							
idle operation																				x	5																						
automatic switch																					x	5																					
open circuit breaker																						x																					
automatic switch																							x	5																			
open circuit breaker																								x	5																		
open bypass																									x	5																	
close guide vane																										x																	
ctrl intake manifold valve																											x	1															
logic for facility power supply																												x	1	1	1												
ctrl guide vane																													x	5				2									
ctrl turbine flaps																														x				2									
position turbine flaps																															x	5											
sfc controller																																x	5										
open guide vane mechanism				-99																													x	2									
nominal rotation speed reached																																		x	1								
parallel open circuit operation																																			x							1		
Logic for power supply																																				x	2					2		
ctrl for generator excitation																																					x	1						
reduce rotor frequency																																						x	1					
close kaplan turbine flaps																																							x	1				
close intake manifold valve																																								x				
ctrl exciting current to grid																																									x	5			1	
check phase sequence																																										x	5			
check phase position																																											x	5	5	
change rotation speed																																											x		
check parallel mode																																												x	1	
close circuit breaker																																													x

Table 3.1: Design Structure Matrix, negative entries stand for modules that should not be clustered, higher for closely related system dynamics.

considered separately.

The weighting is intentionally carried out with as few individual values as possible, as this is tantamount to manual clustering. With regard to large systems, however, structural weighting is preferable, as only selected modules can be adapted above a certain level.

3.4.2 Algorithm

The DSM now provides a rough overview of how the dependencies of all modules relate to each other. In order to be able to determine whether a clustering is valid, a strategy for evaluating good mergers and a strategy for which modules are clustered together first are required. One possibility is the method [35] explained in Section 2.2.5, which uses coordination costs to assess a good clustering. Clustering is performed using a random search algorithm as the optimization problem is dynamic and nontrivial. The design of the DSM results in many local maxima in a strongly nonlinear search space. Since the search space also changes with each clustering, due to the reduction of one element, a robust method that works independently of the arrangement of modules in the DSM is needed. Random search algorithms work by iteratively considering a random candidate within a search space as a new position from a starting position up to a termination criterion. If the new point has a better cost, the new point is used as the better option.

The same now leads to an applicable procedure for the DSM in Table 3.1. The aim is to maximize a cost function. The cost function Φ results from the sum of all the weights w_i within the DSM that are located between the modules, representing the interfaces of a cluster C_k

$$\max \Phi = \sum_{i \in C_k} w_i \ . \tag{3.1}$$

The basic structure is shown in the form of a pseudo-code in Listing 3.1. At the start of the algorithm, the stop counter is set to a high positive value, and the weights of all elements without an entry are set to a low negative value. This ensures that clustering of modules that have no interfaces with each other is prevented, as this leads to a reduction in the cost value. Each module is then packed into its own cluster. The algorithm therefore starts with 46 clusters.

A loop is executed until the stop counter reaches an arbitrary, pre-set value. First, a random module is selected. The module is checked whether it has already been clustered; if not, all interfaces from the selected module to all clusters are evaluated for their costs according to the breakdown in Table 3.1 and a list with their costs is generated. If the module is already clustered, it is removed from the cluster, and the DSM is recalculated with the module removed. A list with all cost values of the module for all remaining clusters is also created. In addition, here the stop-value is increased by 1, since a removal of a cluster indicates that the algorithm is running towards an optimal solution. As there is now a list with all weightings, the best cluster candidate is considered to be the one with the highest weighting. For testing purposes, the cost function is calculated before and after clustering with the best element. For this, a template of the current clusters runs in parallel, and the cost function is always

recalculated from the entire DSM. If the cost function has increased or remained the same, a better clustering is concluded, and the module is added to the cluster. However, if the costs have fallen, this means that the best module was better off in its former cluster and shall not be connected to the supposedly best cluster now. The stop counter is then increased by one, since there has been an indication that the algorithm runs towards a maximum. If the stop counter stays below the predefined value, the process is repeated and a new random module is selected.

```
1  import random
2
3  weight = -0.001
4  stopcounter = 1000
5
6  i=0
7  dsm = read_DSM_from_file()
8  cluster = set_clusters_from_DSM(dsm)
9  clustered_dsm = dsm
10
11 while i < stopcounter:
12      rand_module = random.choice(dsm.index)
13
14      if check_if_module_alone(cluster, rand_module):
15          sum_inter_list = get_sum_inter_list(clustered_dsm,
               ↪ rand_module, weight)
16      else:
17          cluster = extract_module_from_cluster(cluster,
               ↪ rand_module)
18          clustered_dsm = update_dsm(dsm, cluster, weight)
19          sum_inter_list = get_sum_inter_list(clustered_dsm,
               ↪ rand_module, weight)
20          i += 1
21
22      cost = calc_clustercost(dsm, cluster)
23      best_cluster = get_best_pair(sum_inter_list)
24      cluster_try = add_module_to_cluster(best_cluster,
           ↪ rand_element)
25      cost_try = calc_clustercost(dsm, cluster_try)
26
27      if cost_try > cost:
28          cluster = cluster_try
29          clustered_dsm = update_dsm(dsm, cluster, weight)
30      else:
31          i += 1
```

Listing 3.1: Cluster Algorithm.

Concluding, the algorithm broken down here takes the form of an agglomerative random search clustering with a cost function based on weighted interfaces. It must

be kept in mind that a random search algorithm can always lead to different results and run into local minima or maxima. A good evaluation structure of the DSM leads to fewer local minima, as the order in which modules belong together can be clearly assigned.

3.4.3 Results and Setting Parameters

Now there are two setting parameters that lead to different results. These are the stop counter, how long the clustering algorithm shall run, and the weighting of unset elements. If $stopcounter = 1000$ and $weight = -0.01$ are selected, this leads to good results and to a merging of the modules from 46 to 17 modules. Of these 17 modules, 3 are standalone as desired in the manual weighting below the diagonal of the DSM. These 3 modules are *Manage Drain Logic*, *Weir Drain Logic* and *Controller Emergency Release*. There is no clustering with *Controller Emergency Release* as this module has its only interface with the stand-alone module *Manage Drain Logic*.

When repeated, the result hardly varies despite random search, there is always a cluster that can be assigned to the modules of the water balance calculation at the beginning of the control structure, there is always a cluster with the three modules of the weir control, a cluster of the three startup modules using SFC, as well as a larger cluster with the modules of the self-sufficient startup modules including the two upstream modules of the automatic startup. The other operating behaviors are still split into two or have separate clusters for the following modules.

The reason for this behavior can be easily identified by changing the setting parameters. With a low stop counter, even smaller clusters are formed. At $stopcounter = 10$ the number of clusters is already reduced by a third compared to the start DSM. Until the first unsuccessful attempts of clustering, which increase the stop counter, mainly pairs that share interfaces are formed, there are only occasional clusters with 3 or more modules. No clear recurring structure is yet recognizable, and an assignment of the operating flows is not yet foreseeable. Up to $stopcounter = 100$, only 19 clusters are formed. Here, the modules can already be assigned, but parts of an operating flow are still to be clustered, which the algorithm joins together when the stop counter is increased even more. However, clusters are already formed that consist not only of pairs, but also of more modules. Stop counter values above 1000 only lead to more computational effort, but do not lead to any useful improvements as the algorithm runs into an optimum where unclustering leads to reclustering with the same modules.

Changing the weighting of elements of the DSM that do not represent an interface leads to different results depending on whether they are selected as positive or negative. No weighting means that clustering is always preferred to non-clustering. This means that it is always advantageous for the algorithm to perform a clustering, unless a module is removed from an existing cluster and the benefit of the clustering does not exceed that of the new clustering. This leads to fewer clusters in total, but their reliability is no longer always ensured, as it depends on the random selection of the modules that have already been clustered first. This leads to a overlap of operating cases. A positive weight makes no sense at all, as the longer the algorithm runs, the larger the clusters become until all modules are grouped into a huge cluster. The larger the cluster, the more elements without an interface of the DSM outweigh elements with an interface.

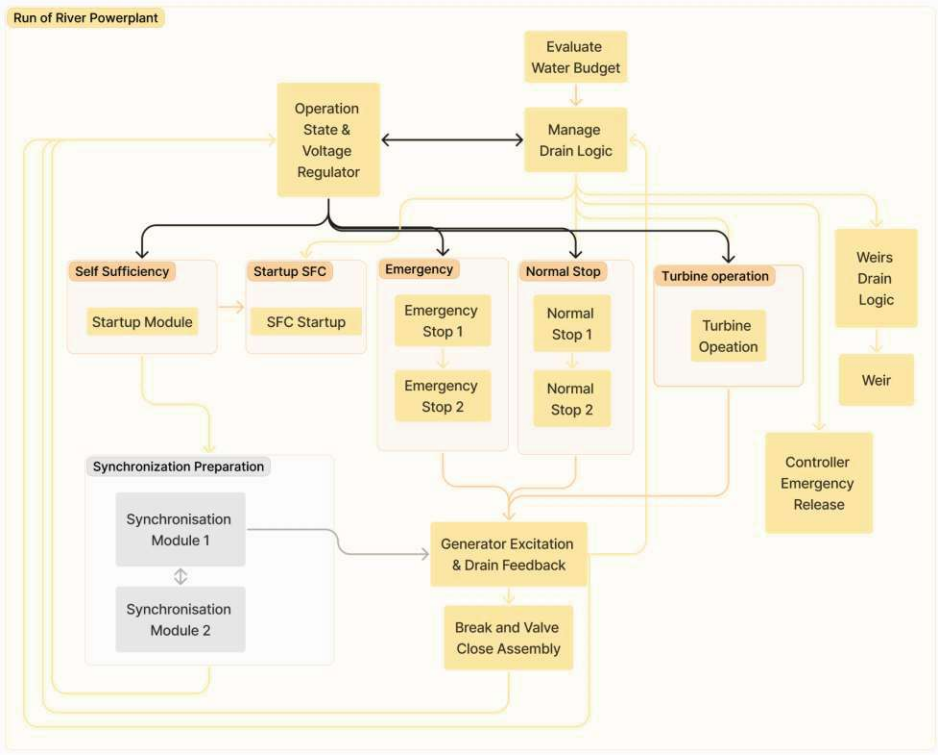

Figure 3.6: Clusters after Algorithm.

However, a negative weighting ensures that clusters do not become too large and that only those sharing interfaces are clustered. Reducing the weighting to -0.1 already leads to smaller cluster sizes of a maximum of 4 modules, and therefore pairs that shall be merged are not merged. From a value of $weight = -0.01$ there is no further improvement in the results.

Therefore, the best setting for this scenario is achieved with the lowest possible negative value of $weight = -0.01$ and a $stopcounter = 1000$. The resulting modules results in the form visible in Figure 3.6.

It can be seen that the structure of the operating flows is maintained. However, a few corrections are still required to achieve optimum clustering for the scenario. Modules that clearly belong to the same operation flow without further external interfaces are not all clustered. The cluster of *Operation State & Voltage Regulator* shall be ideally separated, as the module of *Voltage Regulator* is only used in turbine operation and has nothing to do with the other operating states. This separation also frees *Operation State* to a standalone and easily expandable form, as this module simplifies a key role for the scalability of other operating modes. Furthermore, the cluster of *Generator Excitation and Drain Feedback* offers a solution for efficient excitation control. However, there is an overlap of different operating states where excitation is only reduced in the stop routine and turbine operation, where excitation plays a major role in controlling the reactive power requirement. This results in a further interface needed for this cluster, which gives the signal from the *Operation State* as to whether the machine is in a stop routine or in operation and whether an excitation feed is required due to the torque on the turbine.

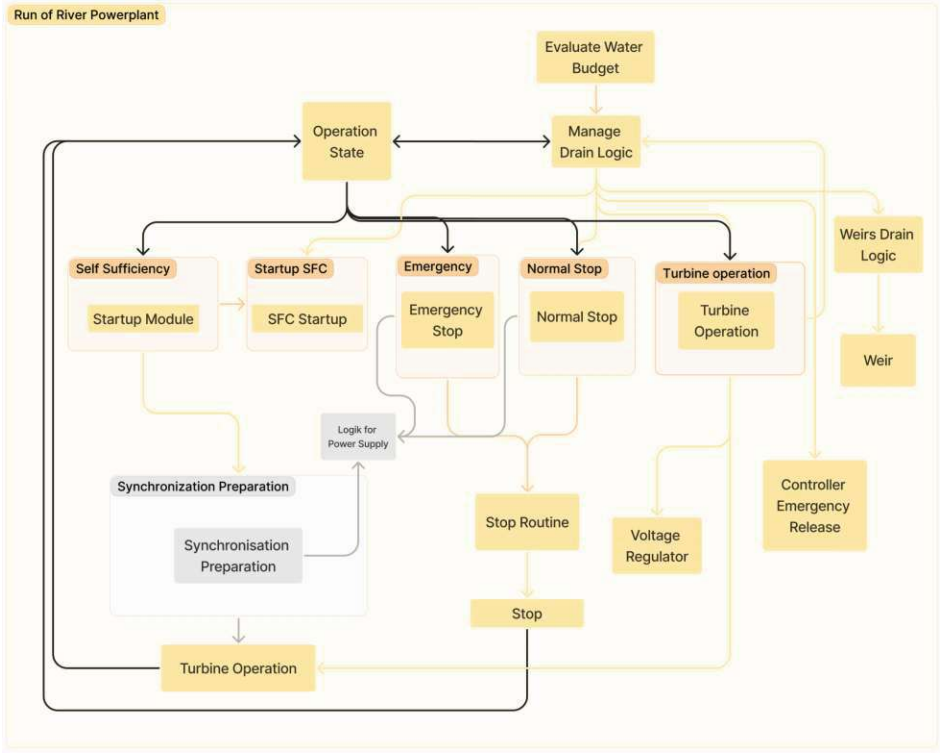

Figure 3.7: Clusters with Manual Correction.

After these corrections, a structure is created that can be more clearly outlined in its function and, therefore, offers a better overview. In addition, there is the great advantage that a comprehensive cluster of the entire machine with Kaplan Turbine can now be formed with a multi-layered subdivision.

Finally, it can be seen that around modules that were manually selected for non-clustering no overlap clustering across the board happens. So they can be set up as targeted safety barriers to prevent the formation of too large clusters in desired regions. The structure offers reasonably robust clustering, and is also carried out without a feedback loop of the synchronization for failed attempts and leads to the same result using the setting parameter $stopcounter = 1000$ increments and $weight = -0.01$. The result shows that it is a good choice to set up the desired system behavior as unconnected as possible in the operating flow structure, since it is clear that most errors occur in modules with many overlapping interfaces. The aim is therefore to organize the modular structure into separate operations and to reuse modules that have interfaces with other processes running in parallel as much as possible, thereby scaling the entire structure.

3.5 Scalability of River Power Plants

An essential aspect is now to adapt the resulting shape to as many run-of-river Power Plants as possible. As described in the beginning of Section 2.1, river power plants differ in their design and dimensions. The design has already been taken into account by the choice of a modular structure. If it is a different type of plant, the new module

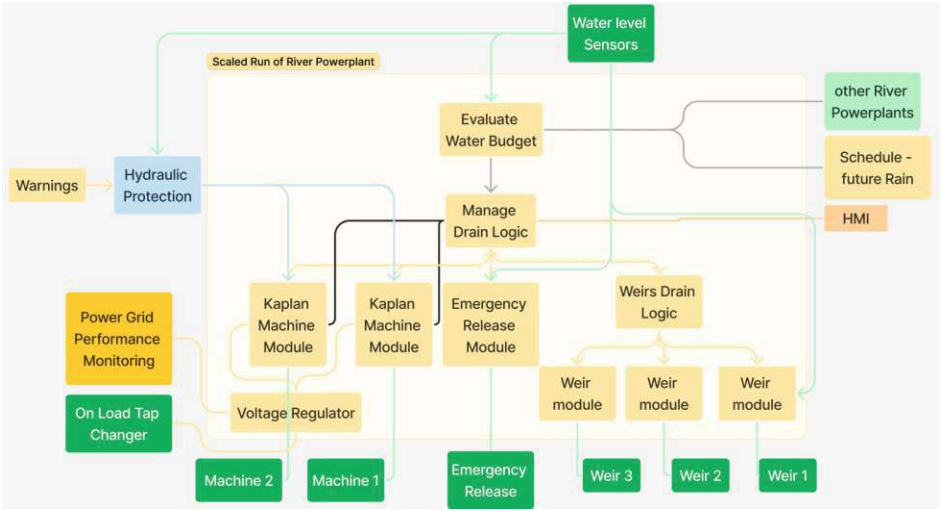

Figure 3.8: Scaled River Power Plant Structure.

only requires an interface for the target flow rate and the water level. The module can now be dimensioned in such a way that the modules are duplicated. The structure has already been designed in such a way that excess water that cannot be converted into energy due to flooding or too much inflow is released to the *Weirs Drain Logic* module. This module first decides how much discharge and in what volume individual weir modules are used. For the entire structure above the *Weir Drain Logic*, it therefore makes no difference what the design of the weirs is and how many are included on the river. To scale the output of a power plant, either larger machines or more machines are required. Since machines ideally operate at nominal load, but the inflow varies over time, a duplication of the machine sets is normally preferred in construction. To ensure that such scaling can be applied to the structure, each machine has 2 interfaces to the *Manage Drain Logic*. One is the state in which the machine is located and the other is the desired flow data. The module must be adapted so that it knows the number and sizes of the machines. Figure 3.8 shows what a scaled version with three weir modules and two machine sets looks like.

CHAPTER 4

Testing of Run-of-River Power Plant

A practical implementation in a simulation environment is the most meaningful means of testing the model set up with the methodology used. As mentioned at the beginning, the control of a power plant is divided into three control structures: power, speed, and flow control. It is considered too complex to implement all three control structures. Since run-of-river power plants are characterized mostly by the fact that they must always operate with an incoming flow of water and cannot be switched on and completely shut down as required like other types of power plant, optimal testing of the modular control system can be achieved through flow control.

4.1 Practical Implementation of the Flow Control

For this purpose, a structure consisting of a single machine, two drum weirs, and an emergency outlet, also in the form of a drum weir, is realized based on real river power plants. This takes into account all flow facilities except for the fish ladder and, in the case of large rivers, a ship lock. The built test environment, called ICS Firing Range, can be seen in Figure 4.1.

In order to test this control structure, a separate structure consisting of simulation and control is set up. The simulation of the technical equipment and the water balance of the river is decoupled from the test equipment of the control system purely via sensors and actuators. The simulation transfers to the control system only the status values for the inflow, which is transferred directly here in simplified form, and therefore not by calculating the change in the level of the underwater level and the upstream level. In conclusion, the simulation functions in such a way that it simulates an inflow curve of the river. In reality, the inflow of a river behaves in such a way that flood events can occur sporadically within a year. This happens more frequently in the spring and autumn months. The inflow increases slowly until it reaches its maximum inflow in a few days due to persistent rain. Since only the behavior of the weirs, guide gates, and emergency outlets and the indirectly coupled headwater levels are to be simulated

Figure 4.1: ICS Firing Range - Water Power Plant Model.

here, it is sufficient to set the inflow using a sine function with an amplitude between a hundred-year flood hundred-year flood (HQ100) and the inflow of a dry phase.

The control system is divided into three linked PLCs in order to be able to simulate the proximity of the weir control system and machine control system. As is usual in VERBUND power plants, the flow logic is implemented in such a way that an outsourced flow value calculation takes place. A central control-PLC determines the flow logic and transmits the target flow values to the PLC for the weirs and the PLC for the machine. Comparing this philosophy with the theoretical design, the control-PLC is a calculation unit for the modules *Track Water Level Curve*, *Calculate Inflow*, *Calculate Desired Drain* and *Manage Drain Logic*. This also raises the question of the PLC assignment of the module *Weirs Drain Logic*. In order to realize completely outsourced logic to the control-PLC, this block is implemented in this PLC. However, the main difference from outsourcing the weir logic to the weir PLC is that the communication between the devices is a single interface, the total target flow, and not any number of weirs with their subdivided target flow. The decision as to which weir discharges how much water is made by the control system, taking into account what a minimum workload of the second weir means for the current inflow. The logic module only knows the maximum discharge capabilities of a weir. Therefore, it does not require information on the current water level. The second weir is only active when the inflow is greater than the first weir can discharge at maximum level. If the inflow is slightly lower, the water level rises until the maximum level is reached and equilibrium is reached. This behavior is desirable because a high water level equals a good water budget. For an even workload of the weirs, the *Weir Drain Logic* module simply has to divide the target discharge Q into fractions and transfer them to the weirs. Each weir calculates its target position independently. To avoid further increasing the number of PLCs installed

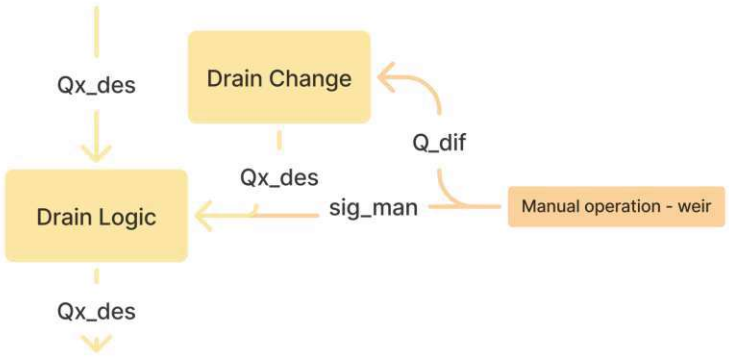

Figure 4.2: Added Modules for Manual Operation.

for economic reasons, the *Controller Emergency Release* module is also outsourced to the PLC control system. Normally, as implemented in most power plants, a separate PLC would have to be installed for the emergency release to remain controllable in the event of a weir-PLC failure. As mentioned above, the PLC for the weir control therefore receives the set flow rate for the weirs as an interface to the control system. In addition, as can be seen from the UML, the weir controller requires the current headwater level to be able to discharge the desired amount of water.

In order to simulate manual operation on site, the flow rate specification was also adapted so that the setpoint specification of the flow rate can come not only from the control system, but also directly from the weir control system via the HMI. Therefore, the existing UML was extended by a HMI, which enables an additional `Q_des` specification. A module is inserted between Weir's drain logic and the Calculate Opening Area, which distinguishes which `Qx_des` shall be used for future calculations. The decision to switch to values specified by the HMI is controlled by a button and thus a digital signal from the HMI.

Calculating the desired weir opening was realized, as theoretically designed, via a step-by-step calculation from the target opening area to the crest of the weir to be controlled. The flow rate `Q[m³/s]` results from the opening area of the weir `A[m²]` relative to the upstream water level, the water pressure and a discharge constant μ which describes the flow characteristics of the outlet. The correlation is shown in Equation (4.1)

$$Q = A * \sqrt{2 * h * g} * \mu \ . \tag{4.1}$$

This means in contrast that it is possible to deduce an opening to be approached as a result of a flow rate resulting in Equation (4.2)

$$A = \frac{Q}{\sqrt{2 * h * g} * \mu} \ . \tag{4.2}$$

Implemented in IEC 61131-3 Function Block Diagram the module *Calculate Opening Area* results in Figure 4.3.

The model works with pure overflow systems, which means that the position of the weir crest is strongly dependent on the water level. The opening area results from the

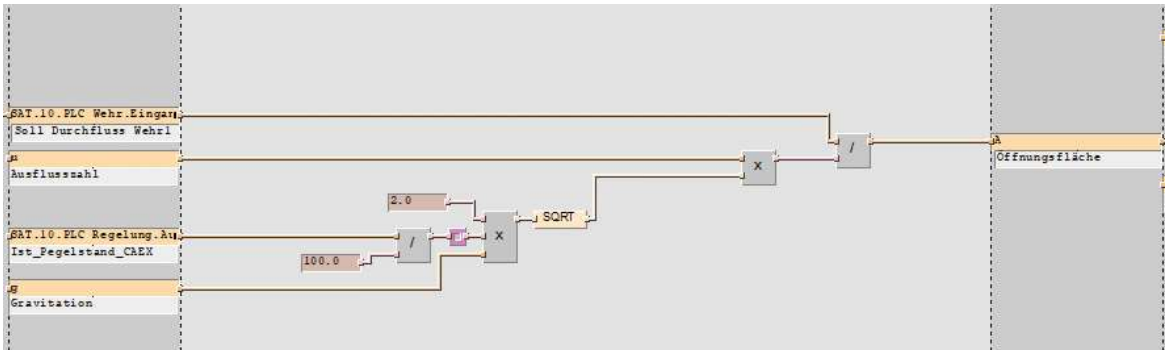

Figure 4.3: Modul Calculate Opening Area.

width of the weir `b[m]` and the height as the difference between the headwater level `h[m]` and weir crest. The length of the weir corresponds to `l[m]`. The opening area therefore results with a percentage angle α 0-100% to Equation (4.3)

$$A = b * (\frac{h}{100} - cos(\frac{\alpha * \pi}{2 * 100}) * l) \ . \tag{4.3}$$

Converted according to the required angle α, this module results in Equation (4.4)

$$\alpha = \frac{2 * 100 * arccos(\frac{\frac{h}{100} - \frac{A}{b}}{l})}{\pi} \ . \tag{4.4}$$

The implementation of the *Calculate Desired Weir Crest* module based on the calculated opening area is shown in Figure 4.4.

To understand, the proportion of the required opening area is calculated up to the central node in the image, in front of the ACOS gate. In the upper half, the weir‘s closing area is calculated for the HMI. Therefore, the result only needs to be multiplied by 100 and subtracted from 100. The covered area can then be easily displayed on the HMI. In the lower half, the calculation of the motor’s actuation percentage takes place. Noting that a conversion of the arc dimension is required. For example, an opening area of 50% corresponds to a proportion of roughly 67% to be transferred. This can be derived from a right-angled triangle in which the adjacent side corresponds to 0.5 times the weir length. The motor’s actuation is scaled from 0-100 with 0 being fully closed and 100 being fully open. The MUX gate is required because it filters out values that are too large or too small and assigns them maximum values to prevent overdriving.

Based on the same considerations with inverted formulas, the current flow rates for the display on the HMI can now be derived from the weir engine‘s position sensor data.

For the machine as a pure flow control system, only the guide vane mechanism had to be controlled with regard to its opening component. As in the theoretical version of the modular structure, the control system transfers to the PLC of the machine the desired water flow `Q_des` that is to be discharged. The controller knows what the maximum possible flow rate through the machine is and transfers this accordingly as the maximum value. The modules *Calc Desired Opening Guide Vane Mechanism* and *Controller Guide Vane Mechanism* are therefore implemented in the machine’s controller. For demonstration purposes, the opening is calculated here in a simplified

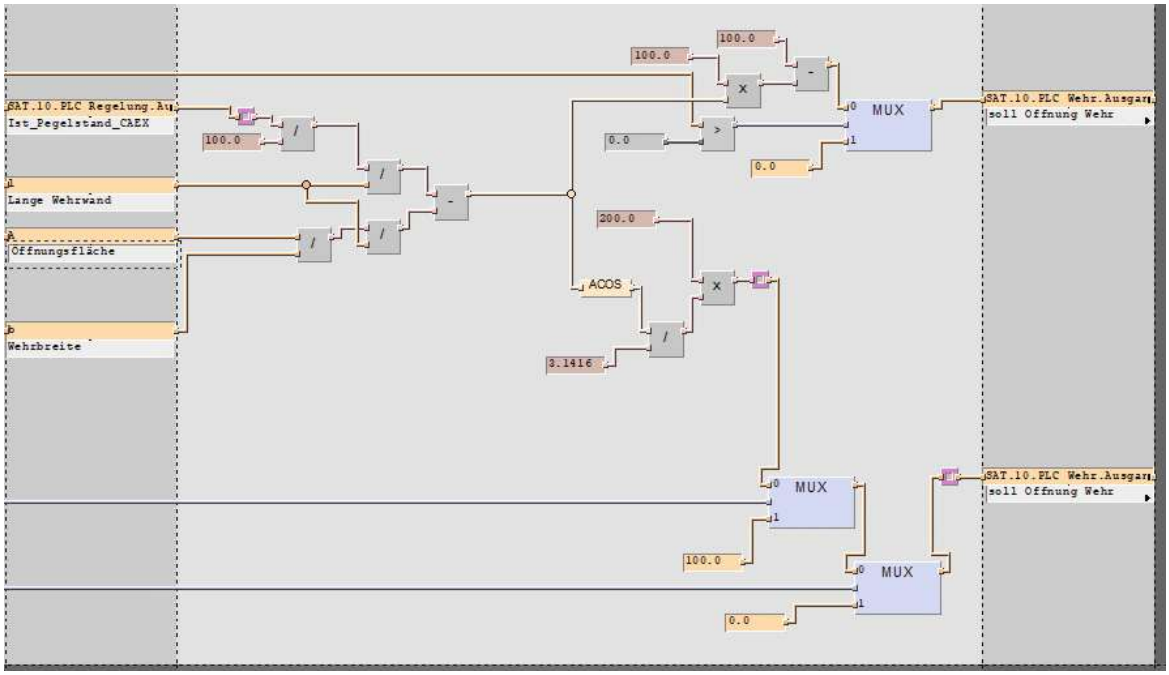

Figure 4.4: Module Calculate Desired Weir Crest.

manner using a proportional opening of the flaps as the ratio of the target flow rate to the maximum flow rate. The implementation therefore does not require the connection to the water level sensor in order to take the water pressure into account.

4.2 Simulation

To test the practical implementation, it is necessary, as already mentioned, to have a simulation environment separate from the control system. For this purpose, an external calculation unit was implemented which calculates the water level based on simulated inflow data, outflow data, weir positions, and turbine openings. For the simulated river power plant, it requires the assumption of a storage basin and a maximum water level at which the water bursts its banks. A function of the basin profile indicates the water level as a function of the fill volume. Tailwater can be realized using a simple discharge function. A constant discharge from the tailwater was assumed. The water level data are then transmitted to the control PLC, similarly to the inflow function. Figure 4.5 now shows the behavior of the system on oscillating inflow values. In green, an inflow function was selected that corresponds to a maximum of HQ100, at which a weir is fully utilized, and a minimum of full utilization of the machine, but no utilization of the weirs. Gray shows the position of the weir motor and light blue the corresponding level-dependent outflow over a weir.

It can be seen that as the inflow increases, the control of the opening angle also increases, which corresponds to a higher outflow. This results in a controlled water level that oscillates in red around a value of approximately 480 cm. The curves clearly show the low sensor resolution, which results in the step pattern. It can be observed that when the weir is fully open and the water level drops during this time, there is a

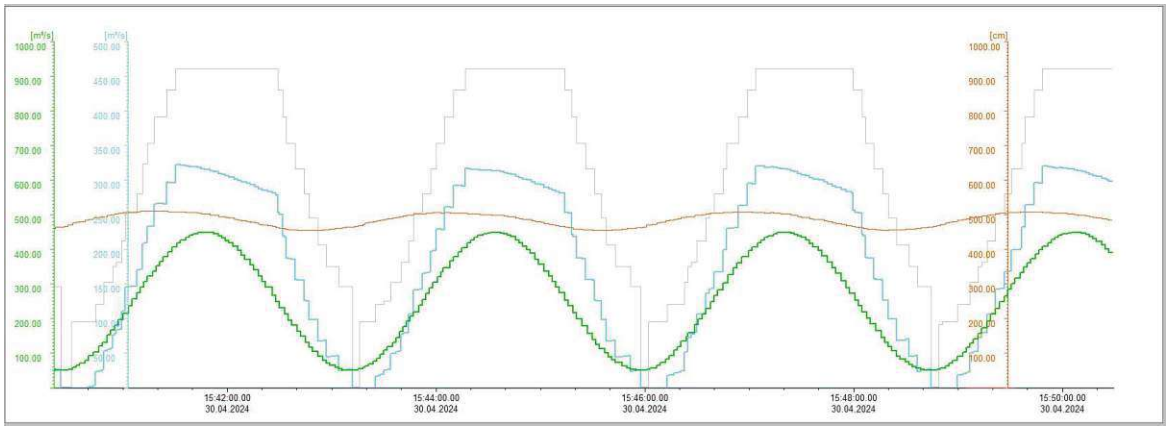

Figure 4.5: Power Plant Reaction to Oscillating Inflow - measured.

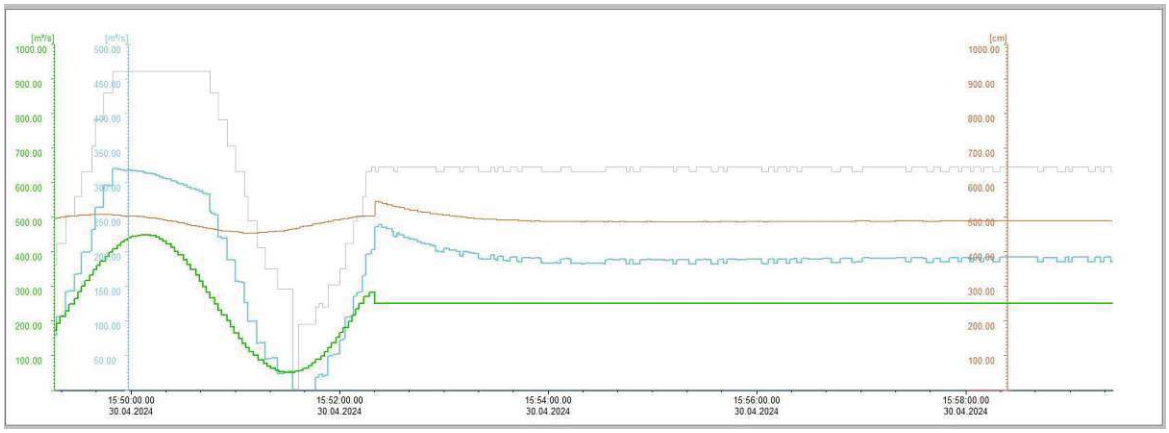

Figure 4.6: Power Plant Reaction to Different Inflow - measured.

drop in discharge over the weirs, as the opening area of the water decreases. It can also be observed that the water level fluctuates slightly. The difference between the highest and the lowest water level is approximately around 50 cm resulting in a maximum level of 510 cm and a minimum of 460 cm . It must be taken into account, that since it takes far too long to simulate a temporally real inflow increase and the benefit of verifiability can also be represented by a fast function, this is a time-lapse recording of a complete flood cycle within 2min 50sec. The weirs are accelerated for the desired compensation. The complete closure of the sluggish weirs would normally take up to ten minutes after full opening.

An additional behavior of the controller can be seen in Figure 4.6. It can be seen that the control also exhibits the expected behavior when the sinusoidal inflow function is switched to a constant inflow by approaching a constant weir position with a constant outflow. The sensor noise can also be clearly seen here, and thus the noise of the discharge calculation at the current position. This noise does not noticeably affect the holding of the water level, so it is kept constant at 590 cm.

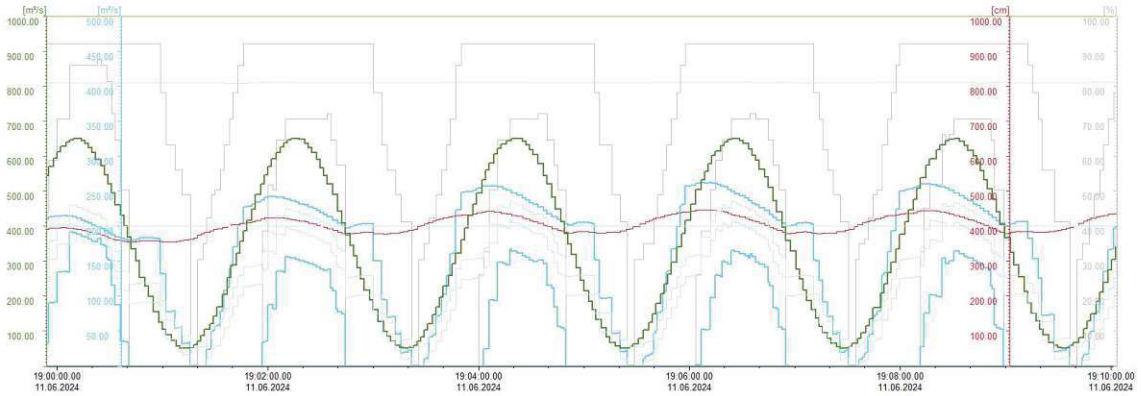

Figure 4.7: Reaction of two Weirs to Oscillating Inflow - measured.

4.2.1 Behavior on Scalability and Exchangeability

There is also the question of scalability and exchangeability. In order to test the usefulness of a modular structure, these two aspects must be feasible. Since flow control is implemented in the realized model, the most obvious thing to test is scalability to higher flows. The easiest way to scale power plants with higher flow rates is to duplicate weir systems. For this purpose, a second weir was added to the structure and the inflow rate was increased via the maximum flow rate of a weir. Figure 4.7 shows the behavior of the new system.

A significantly stronger increase in the inflow can be seen in green. However, the water level in red is still almost constant. The weir logic was set so that the second weir only becomes active as soon as the first is fully utilized. Following the rise of the first gray line for the first weir, it can be seen that the second weir responds as soon as the first is fully open. The inflow is not large enough to fully open the second weir, which means that it discharges too little water, marked in blue. The delay of the weirs to the inflow is again clearly vis8ible via the time shift, which again causes the fluctuation in the water level. However, the functionality is confirmed for larger inflows by quadruplication.

Scalability can be mapped via the duplication of modules and can also be used to determine whether modules can be quickly modified or included with a different design. The system should work just as well and adapt its behavior when modules of a different size are integrated, as long as they have the same interfaces. The higher-level module of the weir logic receives the maximum flow rate as an interface, allocates the target discharges, and the lower-level weir module reacts automatically to the specification. The result when the first weir module is executed with a reduced flow rate can be shown in Figure 4.8.

This shows that the first weir has a steeper opening curve due to its lower maximum discharge volume. Now, not only does the second weir have to react earlier, as the first weir is utilized earlier, but it also opens at a higher rate. Both weirs are now operating at full capacity for the same inflow as before. After the first weir is fully open, the second weir begins its opening process.

Another possibility of changing system behavior through interchangeability is to

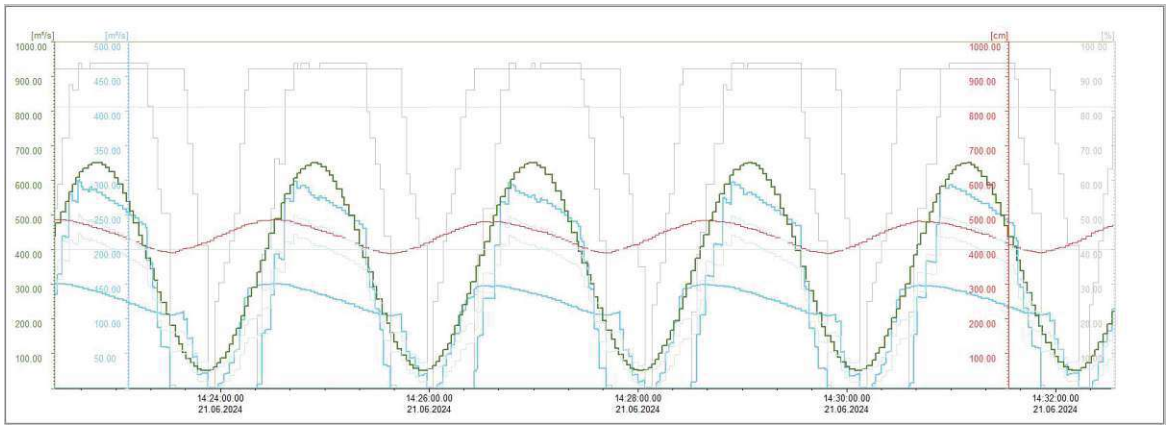

Figure 4.8: Scalability with Different Weir Size - measured.

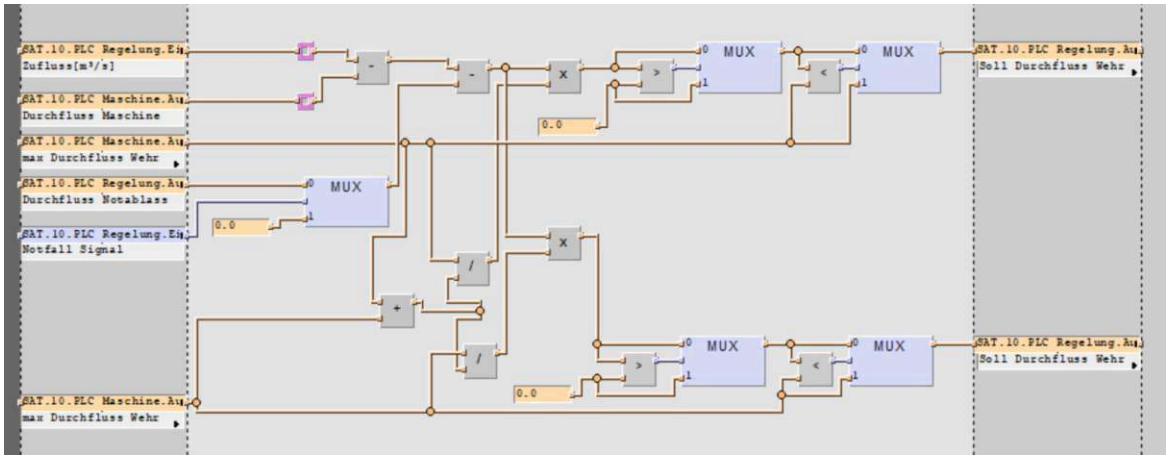

Figure 4.9: Adaptability with Different Weir Control Logic - code.

replace the *Weirs Drain Logic Module* with another processing logic. The response behavior of the weirs shall not run one after the other, but synchronously, for example, in order to achieve a more even use and thus wear of the modules and to prevent too long downtimes of the systems. As in Figure 4.9, this is done by calculating equal shares of the target flows of the weirs. The module receives as input the maximum flow rates of all weirs and the target flow rate to be discharged. In order to achieve uniform utilization, it is not sufficient to divide by the number of weirs, as the previously adapted weir sizes is then not taken into account. Therefore, a calculation form according to Equation (4.5) is required, which is implemented in Figure 4.9 in IEC 61131

$$Q_{des1} = \frac{Q_{max1}}{Q_{max1} + Q_{max2} + ...} * Q_{dessum} \ . \tag{4.5}$$

Now the difference is not taken from the second module as before, but the proportion divided by all weirs so that the degree of opening is the same everywhere. The weirs now run in parallel with the same result as before, only by changing a single module.

This shows that with the same inflow function, neither of the two weirs ever has to open fully, as both weirs together are dimensioned to be able to discharge the set inflow

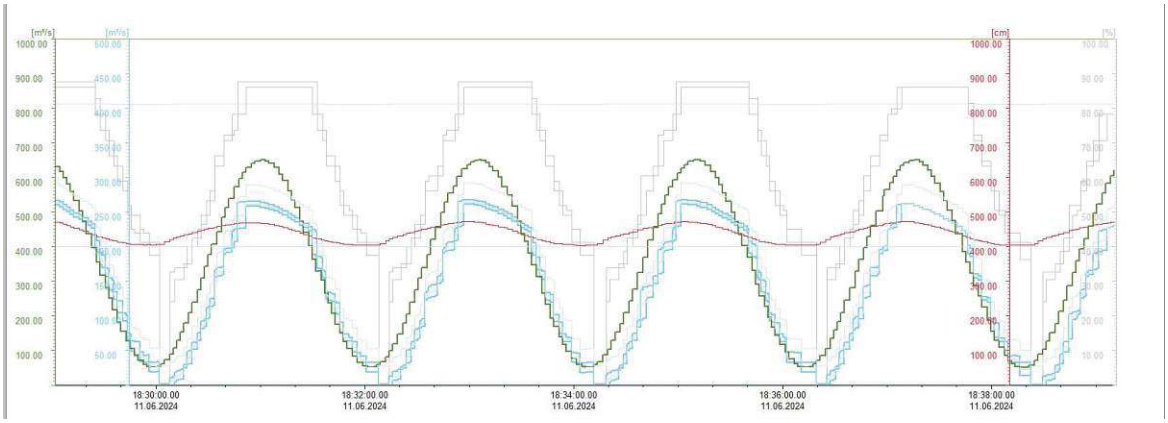

Figure 4.10: Adaptability with Different Weir Control Logic - measured.

with a certain buffer. Both weirs run exactly in parallel, except for a slight time delay caused by the activation. The water level no longer oscillates as much as before with smaller weirs. The reaction of the model is always a regulating behavior of the inflow. A more detailed interpretation of the results is given in Chapter 6 and takes a closer look at the plots and how other system behaviors can integrated.

CHAPTER 5

Designing a Modular Control System for Various Power Plant Types

Finally, the question arises as to whether the findings of the modular control system for run-of-river power plants described in the previous chapters can also be adapted to other types of power plants. Of particular importance here is which modules can be reused and which modules have to be redesigned for other types of power plants.

In Section 2.1 it has already been noted that pumped storage power plants are needed to balance the daily fluctuations in the electricity grid on the consumer and generator side. Unlike run-of-river power plants, these power plants do not have to expect a permanent inflow, but can also be shut down. Inflows into the reservoir normally do not require continuously acting controls, as they do not cause any excessively rapid level fluctuations in relation to the reservoir. In contrast to run-of-river power plants, pumped storage power plants are not base load power plants but peak load power plants. They do not run continuously at full load but can change their output very quickly to compensate for fluctuations caused by consumption in the grid. These power plants also differ in their components, broken down in Section 2.1.1, due to their different applications and therefore different designs. These components must be analyzed for their functionality and a modular structure of the new power plants must be developed. Since the implementation of a storage power plant or a pumped storage power plant requires a new model to verify the system behavior, but this is too expensive to implement, these two types of power plants will only be discussed theoretically in the following sections.

5.1 Storage Power Plant

Since pumped storage power plants not only feed power into the grid, but can also be used to compensate for and store energy, they also have functionally different operating flows than run-of-river power plants. However, one type of power plant that can provide an insight into the relationship between these plants is pure storage power plants. They

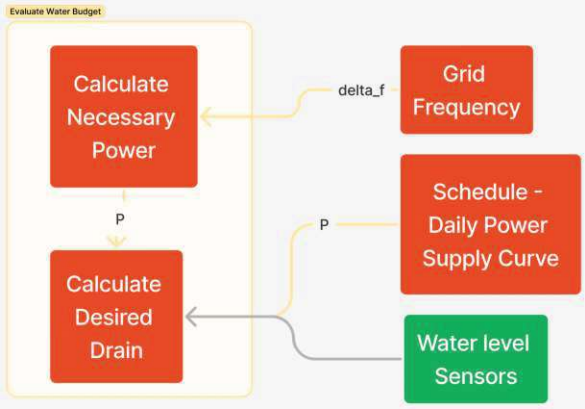

Figure 5.1: Water Budget Module.

are similar in function to run-of-river power plants as the machine sets as well have control valves that regulate the water inflow and must be synchronized for operation according to the same principle. The difference to run-of-river power plants and the similarity to pumped storage power plants, however, is that they do not draw their operating requirements from a continuous inflow but from the grid. The consideration of a storage power plant therefore offers a logical transition to be able to draw a conclusion about pumped storage power plants later on.

If the process flow in the storage power plant is divided up as before, two clear sections emerge. The first is the evaluation of the water budget and the second is how the available water can be optimally converted into energy. As mentioned, the first section is fundamentally different from the river power plant. Since the aim is to cover the network demand and not maintain a level of water as in the power plant, the *Evaluate Water Budget* module needs to be completely restructured. This can now take the form as in Figure 5.1.

Here, deviations in the grid frequency are used to infer a power requirement. The grid frequency is an indicator of whether more energy is required in the grid. If the grid frequency falls, this means that more energy needs to be fed in. In addition, the evaluation of whether the power plant shall start the machines can also be evaluated from the daily power supply curve. From known characteristics and schedules of the machines, a power requirement can then be deduced to a flow rate through the machine sets. In simple form, the thumb formula Equation (5.1) can be used as a good approximation, whereby the power is made up of efficiency, density, gravity, head and flow rate

$$P[\mathrm{W}] = \eta[\%] \cdot \rho[\mathrm{kg/m^3}] \cdot g[\mathrm{m/s^2}] \cdot h[\mathrm{m}] \cdot Q[\mathrm{m^3}/s] \ . \tag{5.1}$$

The interface to the *Manage Drain Logic* module, comparable to the flow power plant, thus consists only of the converted flow Q.

If the components are now considered, it can be seen that they have the same basic functionalities as run-of-river power plants. In the run-of-river power plant there is the intake manifold valve as the final closing body, in the storage power plant this corresponds to a ball valve which is part of the water conductor system. The biggest

difference, however, is the turbine design. As run-of-river power plants are mainly used for low heads, Kaplan turbines are used here. Storage power plants are only profitable at higher outputs and therefore also at higher heads of up to 2000 meters, where Pelton turbines are predominantly used. However, a comparison of the controllable elements also shows that functional correlations can always be drawn.

The functionality is similar to the design with guide vane turbine blades of the Kaplan turbine, except that the water is fed to the bucket blades via a nozzle in Pelton turbines. A nozzle needle is located inside the nozzle, which can be used to throttle the inflow very precisely by means of an adjusting device until it is completely closed. As the adjusting device can only be operated slowly and undesirable pressure shocks can occur if the flow changes too quickly, a beam deflector is installed downstream of the nozzle. This steel deflector is similar to the emergency outlet of the run-of-river power station and guides the water escaping from the nozzle past the bucket vanes in the event of a load shedding. In addition, this device can also be used to shift the current load point, similar to the way in which a reversal of the blade position of the Kaplan turbine reduces the output through turbulence. For the modular structure, the only relevant question here is whether a functional control module can be formed that does not require any additional interfaces and can achieve a comparable system manipulation to the former module by controlling its assigned system units.

It can therefore be seen that the only modules that are different in terms of their function within the structure are those that are used to evaluate water budget requirements triggered by power fluctuations. However, these can be integrated independently of the rest of the structure. Figure 5.2 shows the clustered overview, which represents the basic modules in a similar way to each of Figure 3.8.

The parts that need a complete redesign are shown in dark red, while the Pelton machine module is shown in light red. Most of its modules can be reused, but the control modules of the component interfaces need to be updated. The *Weir module* is also relevant in the storage power plant, as a controlled emergency outlet must also be available for overfilling the reservoir, which can take the form of weirs in a run-of-river power plant. Other options are valves in the dam wall, but if weirs are used, the control module can be controlled directly via a target discharge specification.

Storage power plants have a much greater head than run-of-river power plants, which is why there are significantly more sensors for safety checks of the hydraulic protection, but it shall be noted that the basic structure can be designed modularly, similar to that of a run-of-river power plant. It is possible that modules in the storage power plant get more detailed than in the run-of-river power plant. This is the case with the ball valve, for example. Immense forces are exerted on this control element, as it can also be used as an emergency closing element to close off the pressure tunnel of the hundreds of meters long water column. When the valve opens, there are separate pressure equalization valves to ensure a smooth opening. It is therefore also possible to replace a module that operates a hydraulic system in a power plant with another that maps an entire system logic elsewhere, as long as it provides the same outputs to the downstream units for the overall process flow with the same inputs. This is the case in the theoretical approach of the storage power plant. Any control modules that are directly responsible for the actuation characteristics of the differently designed components must therefore be redesigned to deliver the desired output with the same

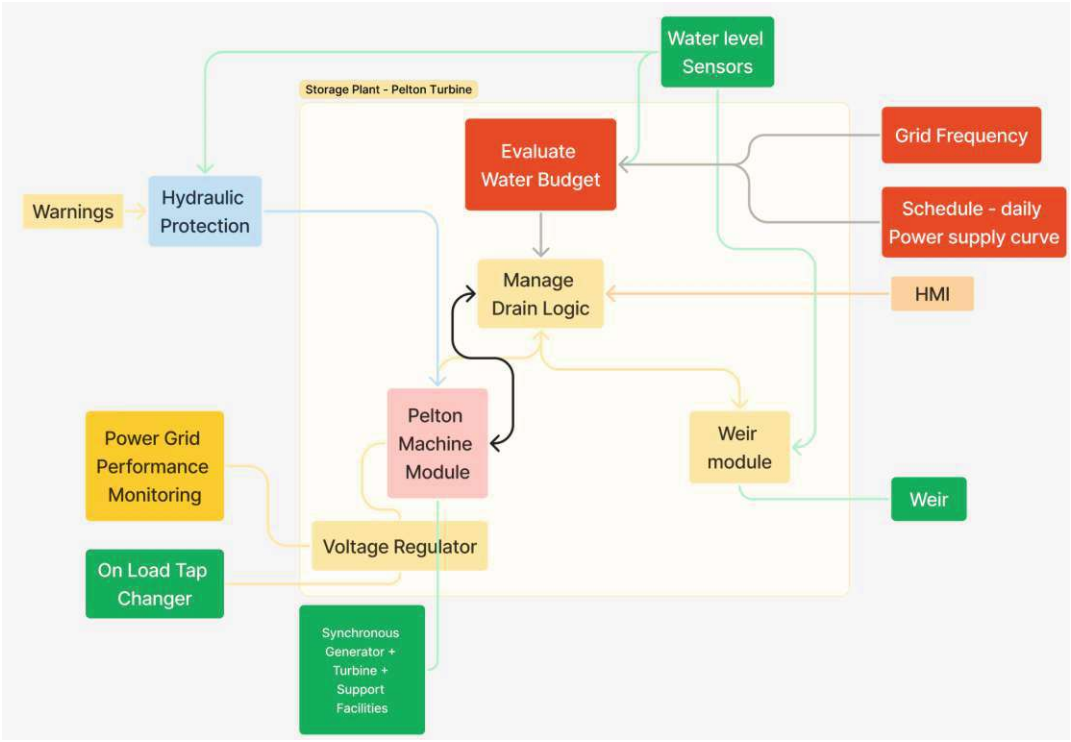

Figure 5.2: Storage Power Plant Structure.

setpoint specification. Control modules without significant dynamics, such as switches or valves, can be reused without having to change the control structure as a whole.

5.2 Pump Storage Power Plant

Since it is analyzed theoretically that the structure of storage power plants can be applied to other hydropower plants in a modified form, there is a comprehensible transition to pumped storage power plants. Pumped storage power plants differ fundamentally from storage power plants in their pumping function. They can not only balance power requirements but also store energy. This means that there is also a strong structural connection here. The most common types are pump-turbine designs, in which the direction of rotation of the machine sets is reversed for pumping operation, or ternary machine sets, in which there is a separate pump and turbine design, visible in Figure 2.3.

As modular structures are the easiest to expand, the ternary design is the best option when considering a pumped storage power plant. It is only necessary to consider how the additional operating flow of the pump operation can be integrated into the control structure. The operating flow of the turbine operation can then be maintained in all its versions due to the separation of the *Operation State* module. The significant difference between the ternary design and the pump turbine design for this consideration is the more complex controllability of the pump turbine speed in pumping mode. Using a synchronous generator, a pump will always rotate at the same speed and thus have a constant output. Either a different machine design or a controllable frequency converter therefore has to be interposed. The situation is different with ternary machine sets.

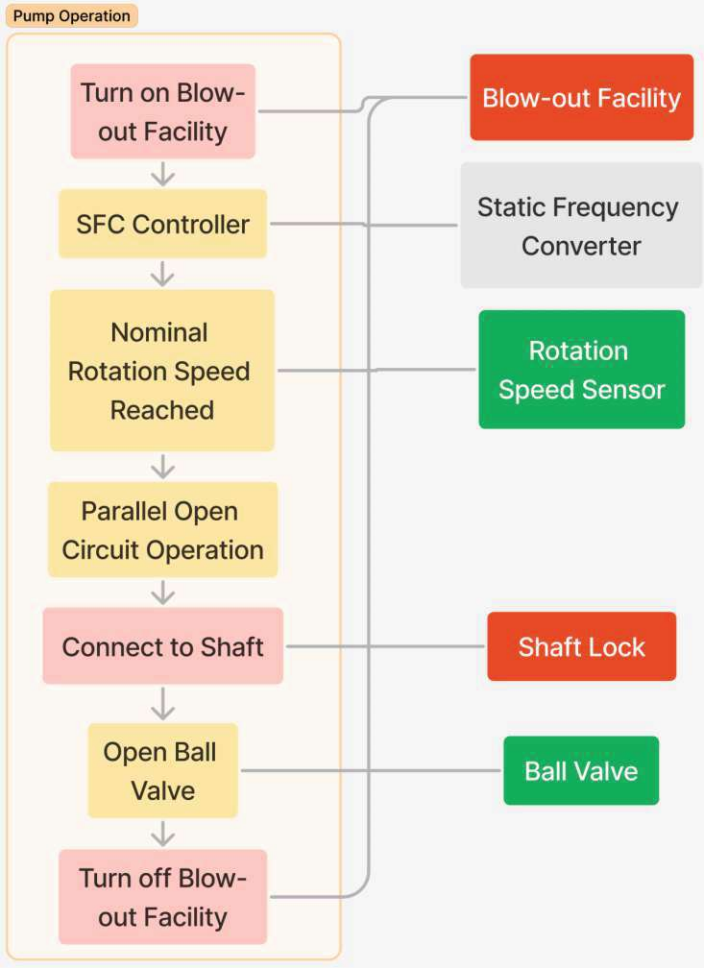

Figure 5.3: Pump Operation.

Although the pump is not controllable as well and therefore always pumps with the same power, the separate design means that the so-called hydraulic short circuit can be used. If less power is to be drawn from the grid than the pump delivers, the generator is operated in parallel in turbine mode and compensates for the power difference. As in Figure 2.3, water is therefore circulated in a short circuit.

The new operating flow *Pump Operation*, visible in Figure 5.3, is required for the pump to be put into operation.

As in the storage power plant, the new devices *Blow-out-facility* and *Shaft Lock* are shown in red. To put the pump into operation, it must first be brought up to speed. This time, this is not possible via a controlled flow through the paddles, as the pump has to be started up continuously via a static frequency converter. To do this, the pump must be dried with a fan, as the starting load would be too high. Once the pump is at synchronous speed, it is connected to the generator via the shaft lock, and the ball valve is opened. Water can now be pumped uphill.

This operating flow must then be included in the *State operation* module, and for the hydraulic short circuit it must be possible to run *Pump operation* and *Turbine operation* in parallel. The *Manage Drain Logic* module can then also process negative

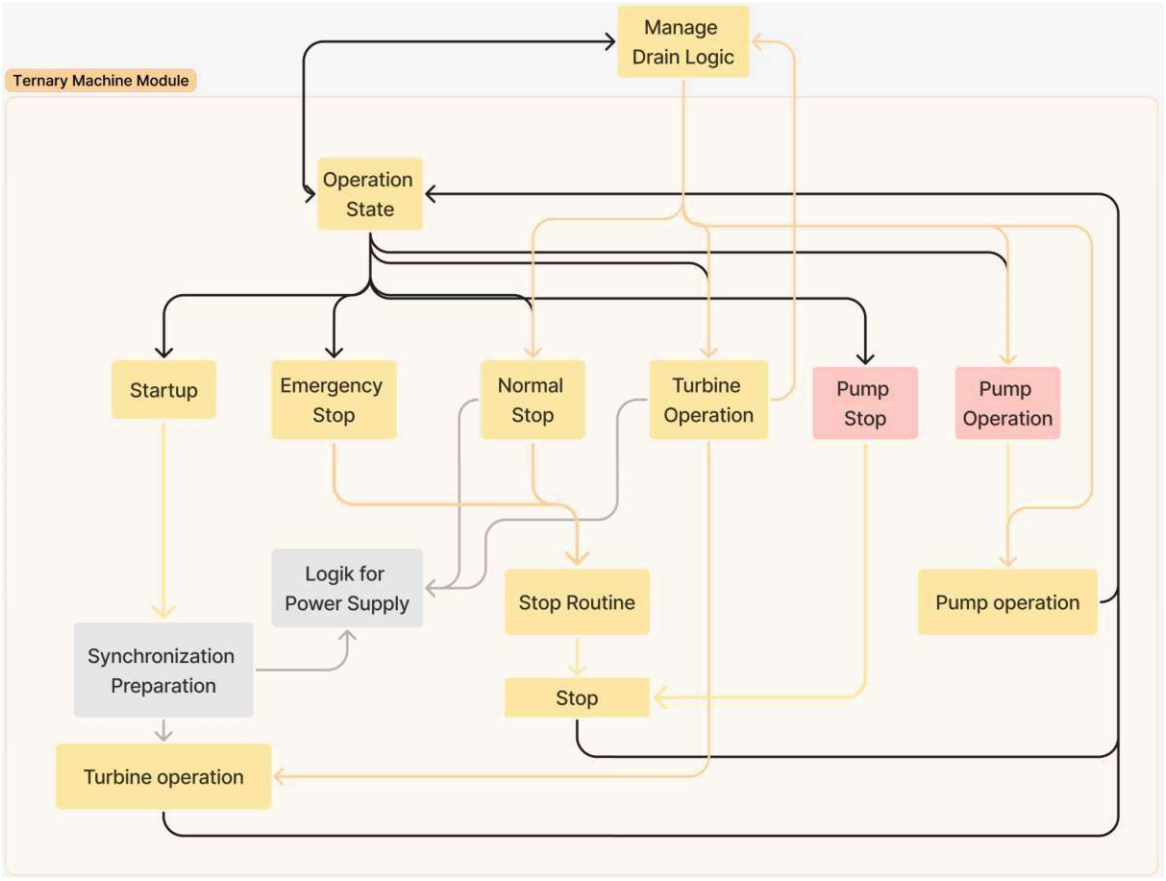

Figure 5.4: Ternary Machine Module.

flows and distribute them to the states. The Water Budget module can retain its form in the same way as in the storage power plant and thus be reused the same, it must only be designed to infer a negative flow value from an increasing grid frequency.

This results in an extended ternary machine module of the previous turbine module with a pump function from Figure 5.4. Here, the direct connection from the *Manage Drain Logic* to the interfaces of the first modules for flow specification, as well as to the *operation state* module cause state changes. There is also an empty connection without an intermediate module between the *Pump* operation state and the *Manage Drain Logic* module, which represents the non-controllable pump operation. The *Pump Stop* module does not require an interface to the *Manage Drain Logic*, as this is only used to decouple the shaft and close the ball valve.

The clustered version for the pumped storage power plant is then the same as the one of the storage power plant from Figure 5.2, only that the *Pelton Machine* module must be exchanged with the *Ternery Machine* module.

CHAPTER 6

Discussion and Conclusion

This section, first discusses the two research questions regarding the transferability and reusability of different power plant control modules. Next, the key factors of the structural approach are explained to ensure the applicability of the control structure. Then, a contextual analysis of the clustering is carried out, to explain why the algorithm does not always clusters modules, belonging to the same process flow. Next, artifacts in the constructed physical model are observed to better assign the behaviour of the simulation. These elements highlight the potential of the modular system and areas for further refinement and calibration. Finally, the conclusions of the work is presented, and the focus of future research is highlighted.

6.1 Research Questions

Research Question 1

Is it possible to develop a modular control approach that can also be used to map the operating cases of other hydropower plants?

Chapter 5 demonstrates that the modular control approach designed for run-of-river power plants can be applied to storage as well as pumped storage power plants. This confirms research question 1, which asks whether a modular control approach can be found that also models the operational cases of other hydroelectric power plants. Due to the modular functional separation of components, it is possible to realize necessary status changes of components, with different dynamics, without altering the control structure as a whole, by only replacing individual control modules. These modules can be integrated on the basis of the similarity of their functionalities to represent other types of water-driven power plants. By creating a flow-driven structure and converting the desired system behavior into a water demand, which serves for grid balancing in storage power plants and flow regulation in run-of-river power plants, it is possible to create power plants of different designs using the same modular structure.

The operating cases of a run-of-river power plant can thus be partially transferred to other types of power plants. New operating cases can be incrementally added to the existing structure through a separate state control module, as demonstrated by the pump operation of a ternary unit.

Research Question 2

Which logic modules can be reused in a scalable manner?

Since functions are clearly assignable to facilities and separable from each other, the clustering in Section 3.4.3 results in two different types of basic modules. These represent, on one hand, the modules of the weirs, and on the other hand, the modules associated with the machines. Replication of these modules enables the scalability of power plants. This allows research question 2, which logic modules can be reused in a scalable manner, to be answered through the analysis of the use of these two basic modules.

In this modular approach, the weirs operate independently, reacting only to outflow targets and water level data. The control modules of the weirs can be reused if they are designed to first calculate the necessary opening area based on an outflow target, and only in the final process step incorporate plant-specific properties such as size specifications for the control of actuators.

Machine-specific modules can be reused if they control actuators in components with identical design and dynamics. By functionally assigning the modules and precisely defining the interfaces, modifications are required solely in their control behavior to accommodate machines of the same design but different sizes. For the reuse of modules in machines with different designs, such as the transition from Kaplan turbines to Pelton ones, it is necessary to replace all modules that handle turbine-specific functions. Due to the possibility of functional separation of hardware, these modules can be deployed at the same control points, thus avoiding any control structural redesign efforts. For the control design of a ternary turbine set, the modules of a Pelton turbine can be reused and only need to be expanded with the pump operation modules through the adaptation of a state-coordinating module. Modules responsible for maintaining process coordination through binary signal transmission to valves or switches can be reused directly.

6.2 Key Factors of the Structural Approach

Although the literature lacks approaches for modular control design in hydropower plants to enable cost-effective and efficient implementation, this work establishes the foundation for modular structures. Consequently, in the state of the art, there is no overview available of the fundamental structure needed to develop modules and determine which controllable units achieve the desired status changes for optimal power plant behavior. Based on the specific operational flows of the power plant, a control structure can be developed that is built and expanded step by step. This allows for the classification of the associated control modules. The implemented control modules are connected directly to power plant facilities. Therefore, it must be ensured that only

one control module controls a resource at any given time and that each module has all the data it needs to continue the process flow in real-time. The monitoring center in the created structure is the *Operation State* module, which always opens only one process flow, or in the ternary case also allows several selected ones per machine. This prevents several modules from providing contradictory specifications. To extend the system, a new operating flow must be added and enabled from the *Operation State*. If a process is to be executed in parallel, simultaneous access to the same resource must be prevented either by adapting the *Operation State* module or by redesigning process flows. For the applicability of this approach, it is therefore essential to check interfaces and coordinate parallelism.

6.3 Contextual Analysis of Clustering

To facilitate the reuse of created modules, it is advantageous to cluster modules of the same operational flow.

The aim is to show that it is possible to automatically combine modules with strong ties via the evaluation of a cost function in order to maintain manageability in large systems. To represent the core factors for functional relations in modular functional structures, interfaces serve as evaluation factors. However, not all modules belonging to the same process flow are fully merged, as it is not always advantageous for the iterative cost function to merge 2 clusters, each with 2 modules into a cluster of 4. In the clustering process, a module is removed from its cluster of 2 to perform an assessment of the best interfaces. Only if the other cluster of 2 turns out to be the better candidate, the previously removed module is added to the new cluster, resulting in a cluster of 3 and a cluster of 1. If the remaining module is then randomly selected in a single run, the current size 3 cluster is the best option, and the desired cluster of 4 emerges. The algorithm still tends to re-cluster initially clustered modules due to its iterative random search approach. In order to force alternatives more frequently, various tricks can be used to extend the algorithm. One possibility is to check whether the module has already been reclustered with its best candidate before and instead connects to the second best. However, since this does not necessarily mean that the algorithm then runs to a better local optimum, a buffer can be used to run different solutions in parallel. Another option would be an evaluation function that favors larger clusters. However, this may not be desirable in all cases, as larger clusters automatically attract more dependencies and the administrative effort increases. Adaptations are then associated with significantly greater effort.

Ultimately, the algorithm uses a random search method, which is why the results are not always reproducible and the results are highly application dependent. The best way to mitigate these problems is to calculate a large number of optimization results and compare them with the use cases, as done in [35]. The aim in this work is to minimize interfaces and cluster modules that are as function-oriented as possible. Since the modular structure has to be checked before implementation anyway, the clustering is only a suggestion to minimize design effort and accelerate design processes according to modular construction methods. Another approach is to carry out the clustering of the components in a way that is already recognizable via the functional areas. This

however is more difficult to assign for larger systems. A higher degree of automation also leads to stronger linking of functional areas. The more interfaces are overlapping, the more areas merge and thus the affiliation of the meaningful control modules is harder to extinguish.

6.4 Signal Artifacts in ICS Firing Range

The final implementation in Section 4.2 demonstrates that the modular system exhibits the desired flow behavior with modular properties such as interchangeability and adaptability for a run-of-river power plant. However, the measurements reveal signal artifacts that require further investigation. Figure 4.5 shows that the devices of the power plant model have the expected reactions. The aim is to maintain the water level by controlling the flow, however it can be observed that the water level fluctuates slightly. This results on the one hand from the high frequency of the change in inflow and on the other hand from the slow response of the weirs, but above all from the low resolution of the actuators. Due to the time lag of the weirs and the sensor resolution, where the control point is never exactly on the transmitted values, the fluctuations occur. A new position is only approached from a threshold value. If the inflow increases, too little water is discharged as a result of the lagging, and the water level rises minimally. If the inflow falls, too much is discharged, which again balances out the curve. This behavior is always balanced out by the continuous inflow function. If a function with different rise and fall behavior were to be incorporated, better hardware is needed in the model, or it has to be switched to water level control. It is possible to implement water level control by replacing the desired drain module. However, this is not the norm in power plants, but is used sporadically.

The fact that the control also works correctly with alternating inflow characteristics can be best seen in Figure 4.6. The water level remains constant as desired with a constant inflow. However, the noise of the motor is clearly visible, along with the calculated outflow over the weir. At the beginning of the inflow function change from alternating to constant, it can be observed that the water level drops and runs to a constant value. The jump in the water level is due to the simulation being reset in the model when changing the inflow function. It is observable that the water level is suddenly set to a starting value. The degree of opening of the weirs is constant, as the inflow function is also constant. The water level and thus also the directly dependent discharge function however, falls slowly, as the low actuator resolution does not allow more precise positioning until the water level is in equilibrium with the current discharge value and maintains the water level without further changes.

Figure 4.8 shows that modules can not only be duplicated but also function in other implementations with the same interfaces. This is particularly important to ensure the scalability of different rivers. However, the reason why multiple modules are used in all power plants is for reasons of redundancy. Power plants must be fail-safe, which is why it is common practice to design at least two versions of equipment to protect against failures. If the interfaces are clearly defined and function independently via simple interfaces as shown, they can simply be inserted in different sizes. With regard to the interchangeability of modules, it can be noted that this has already been taken into

account in the design of the entire structure, right down to the clustering. The upper part up to the *Manage Drain Logic* provides the specification for flow control. If the system is switched to level control, e.g. for threshold operation, where the maximum water level is approached, this section can be replaced or extended, as this part supplies the desired discharge to maintain the water level or correspondingly to assume a desired water level. A certain discharge must always be present due to the dry running of the river or for shipping, but this can be implemented individually in the modules.

Taking a closer look at the weirs' opening behavior, a jump as soon as the weirs open is observable. This can be seen particularly well in Figure 4.10. This is due to the fact that the design automatically adjusts the opening behavior for different water levels. The weir must first open a certain amount until the weir crest passes the upper water edge and effectively more water is discharged with increasing opening. However, as this behavior is already implemented in the weir modules, it does not have to be taken into account in other modules. Only the interface definitions must be considered.

This shows that river power plants can be fundamentally structured in a modular way, which also meets the modular requirements of scalability and interchangeability. A challenge of modular systems is the high maintenance effort required to incorporate new functionalities as well as lower system optimization. It is always necessary to check correlating dependencies when adjusting existing modules so that no undesired side effects occur. Therefore, even before implementation, consideration has to be given to which modules require more regular processing so that they remain more accessible in the clustering to larger modules so that no dependencies are formed that regularly increase the maintenance effort. This is best illustrated by the branching of the control structures.

6.5 Conclusion and Outlook

Overall, this work presents a methodology for designing hydroelectric power plants in a modular and expandable way. The approach involves breaking down the facilities into functional, separable modules, which can be automatically clustered using a specialized algorithm to ensure module reusability. As a result, a control structure emerges that facilitates the integration of additional power plant-specific behaviors by incorporating them into operational flows. A coordinating state module is used to ensure the current operating mode without conflicting signals to the same facilities. This flexible structure is not only adaptable to various types of power plants, but also enables the reuse of overarching modules. The structure shows the desired flow control behavior within a purpose-built simulation environment and exhibits modular properties through the exchange of implemented modules. This marks the first introduction of a modular control approach for hydroelectric power plants, laying the foundation for more advanced structures.

The aim of future work is to analyze the transformer cavern in more detail, integrate protective components for cooling or overvoltage control, and to analyze whether new interfaces to the machine cavern arise. This would expand the system in its depth, as many control structures are naturally implemented to voltage regulation in power plants, yet only the components of the actuators for water balance and power regulation were

taken into account here. It is recommended to investigated whether the new modules can be integrated directly into the turbine operation or whether a separate parallel operating flow must then be added. In the future, the main focus should therefore be on expanding functions and continuously refining the depth of the model. In doing so, the clusters should always be reevaluated if interfaces arise that lead out of more precise subdivisions of the modules. When integrating new operating flows, it must be checked whether the state module still makes sense as an authorization unit or whether different strategies like token handling make more sense.

A useful way to extend the methodology would be to automatically convert the UML structure directly into a DSM. This would minimize errors and effort. There are repeating modules in the DSM, which is also desired for the creation of clustered modules. It is not essential in which order the modules are embedded. An ordered structure along the diagonal is just for understanding, since the algorithm does not distinguish at which position modules are located. This means that no evaluation of the layout of the DSM axis is necessary and automatic transfer can be performed without reference to the application. Furthermore, when adapting modules, it must always be considered that all duplicated modules are kept up-to-date. Therefore, it would make sense to build up a library of modules and use pure templates as clusters to only address modules in an orderly manner. A change in one module will cause changes in all modules.

The transfer and reuse of modules from run-of-river power plants to other hydropower plants is only described theoretically, but cannot be verified in a measurable way, which means that a test environment needs to be created here to check if and how interfaces change. Another important point is to check the structure for pump turbines. Today, pump turbines are installed in newer pumped storage power plants due to their ability to permit smaller cavern sizes. As already explained, pump turbines must be speed controlled, which requires a precise analysis of modular applicability. It is of interest however to see whether modules can be reused then as well.

An extension of the modular pumped storage power plant structure to phase-shift operation for reactive power compensation would provide many insights into applicability of further operation modes. In run-of-river power plants, reactive power compensation was not a relevant issue, as it is common practice to operate at a constant phase angle. However, in pumped storage power plants, this can be dynamic in order to cover the reactive power requirements of the grid. It is of interest to see whether the control of the machine then changes in its flow behavior.

Lastly, as a next step for the ICS Firing Range, a higher resolution of the actuators is to be implemented so that more control points of the weirs can be approached in order to be able to make a better assessment of different control methods.

Bibliography

[1] C. Schusser, "Modell zur ermittlung des wasserwertes von speicher- und pumpspeicheranlagen," Ph.D. dissertation, Technischen Universität Graz, Graz, Feb. 2013.

[2] "Are modular systems the future for hydropower?" National Hydropower Association. (Jul. 26, 2021), [Online]. Available: https://www.hydro.org/powerhouse/article/are-modular-systems-the-future-for-hydropower/ (visited on 07/03/2024).

[3] S. Feldmann, C. Legat, and B. Vogel-Heuser, "An analysis of challenges and state of the art for modular engineering in the machine and plant manufacturing domain," *IFAC-PapersOnLine*, vol. 48, no. 10, pp. 87–92, 2015.

[4] C. R. Maga and N. Jazdi, "Interdisciplinary modularization in product line engineering: A case study," in *Proceedings of 2012 IEEE International Conference on Automation, Quality and Testing, Robotics*, Cluj-Napoca, Romania: IEEE, May 2012, pp. 179–184.

[5] J. Giesecke and S. Heimerl, *Wasserkraftanlagen: Planung, Bau und Betrieb*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.

[6] K. Gemeinde, *Das Kraftwerk im Berg: Die Baugeschichte des Pumpspeicherkraftwerks Limberg II*, 1st ed. St. Pölten: Residenz, Oct. 15, 2011, 288 pp.

[7] ANDRITZ AG. "Pumped storage by ANDRITZ." (), [Online]. Available: https://www.andritz.com/products-en/group/products/pumped-storage (visited on 07/09/2024).

[8] M. d. Martin, "Definition, Spezifikation und Analyse stressrelevanter Testfälle für Turbinenregler in Wasserkraftwerken," Accepted: 2020-06-29T12:54:30Z Journal Abbreviation: Definition, specification and analysis of stress test cases for power plant governor controllers, Thesis, TU Wien, Wien, 2019.

[9] Z. Dong, "Dynamic model development and system study of ternary pumped storage hydropower," Ph.D. dissertation, Auburn University, Auburn, Alabama, Apr. 18, 2019.

[10] U.S. Department of Energy. "Types of hydropower plants." (), [Online]. Available: https://www.energy.gov/eere/water/types-hydropower-plants (visited on 08/19/2024).

[11] *Die Alpenbatterie: Ökostrom aus dem Berg*. Wien: Kremayr & Scheriau GmbH & Co. KG, Jul. 10, 2016, 240 pp.

[12] B. Vogel-Heuser, J. Fischer, S. Feldmann, S. Ulewicz, and S. Rösch, "Modularity and architecture of PLC-based software for automated production systems: An analysis in industrial companies," *Journal of Systems and Software*, vol. 131, pp. 35–62, Sep. 1, 2017.

[13] J. Christensen, T. Strasser, A. Valentini, V. Vyatkin, and A. Zoitl, "The IEC 61499 function block standard: Overview of the second edition," presented at the ISA Automation Week 2012, Orlando, Florida, USA, Sep. 24, 2012.

[14] G. Erixon, "Modular function deployment: A method for product modularization," Ph.D. dissertation, KTH, Dept. of Manufacturing systems, Stockholm, 1998.

[15] H. Taghaddos, U. Hermann, S. AbouRizk, and Y. Mohamed, "Simulation-based multiagent approach for scheduling modular construction," *Journal of Computing in Civil Engineering*, vol. 28, no. 2, pp. 263–274, Mar. 2014.

[16] O. Bedair, "Rational design of pipe racks used for oil sands and petrochemical facilities," *Practice Periodical on Structural Design and Construction*, vol. 20, p. 04 014 029, May 2015.

[17] E. Ikpe, J. Kumar, and G. Jergeas, "Analysis of modularization compared to total project cost in alberta oil and gas projects," *Global Advanced Research Journal of Management and Business Studies*, pp. 116–120, Mar. 2015.

[18] M. Blackenfelt, "Managing complexity by product modularisation," Publisher: Maskinkonstruktion, Ph.D. dissertation, Royal Institute of Technology, Stockholm, Sweden, 2001, 100 pp.

[19] B. Vogel-Heuser, D. Witsch, and U. Katzke, "Automatic code generation from a UML model to IEC 61131-3 and system configuration tools," in *2005 International Conference on Control and Automation*, ISSN: 1948-3457, vol. 2, Jun. 2005, 1034–1039 Vol. 2.

[20] C. Secchi, M. Bonfe, and C. Fantuzzi, "On the use of UML for modeling mechatronic systems," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 1, pp. 105–113, Jan. 2007, Conference Name: IEEE Transactions on Automation Science and Engineering.

[21] M. Dumas and A. H. M. ter Hofstede, "UML activity diagrams as a workflow specification language," in *UML 2001 — The Unified Modeling Language. Modeling Languages, Concepts, and Tools*, M. Gogolla and C. Kobryn, Eds., Berlin, Heidelberg: Springer, 2001, pp. 76–90.

[22] draw.io. "Draw UML activity diagrams." (Nov. 23, 2022), [Online]. Available: https://www.drawio.com/blog/uml-activity-diagrams (visited on 08/19/2024).

[23] B. Kormann, D. Tikhonov, and B. Vogel-Heuser, "Automated PLC software testing using adapted UML sequence diagrams," *IFAC Proceedings Volumes*, 14th IFAC Symposium on Information Control Problems in Manufacturing, vol. 45, no. 6, pp. 1615–1621, May 23, 2012.

[24] C. Glenn, R. Kalawsky, and G. Watson, "Model driven architecture for autonomous systems - modelling representations," presented at the Proceedings of the 5th SEAS DTC Technical Conference, Edinburgh, Jul. 2010.

[25] Y. Singh and M. Sood, "The impact of the computational independent model for enterprise information system development," *International Journal of Computer Applications*, vol. 11, no. 8, Dec. 10, 2010.

[26] S. Kherraf, É. Lefebvre, and W. Suryn, "Transformation from CIM to PIM using patterns and archetypes," in *19th Australian Conference on Software Engineering (aswec 2008)*, ISSN: 2377-5408, Mar. 2008, pp. 338–346.

[27] M. Melouk, Y. Rhazali, and H. Youssef, "An approach for transforming CIM to PIM up to PSM in MDA," *Procedia Computer Science*, vol. 170, pp. 869–874, 2020.

[28] Y. Liu and Y. Ma, "An approach for MDA model transformation based on JEE platform," in *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, ISSN: 2161-9654, Oct. 2008, pp. 1–4.

[29] Y. Rhazali, Y. Hadi, and A. Mouloudi, "A methodology of model transformation in MDA: From CIM to PIM," *International Review on Computers and Software (IRECOS)*, vol. 10, no. 12, p. 1186, Dec. 31, 2015.

[30] W. Zhang, H. Mei, H. Zhao, and J. Yang, *Transformation from CIM to PIM: A Feature-Oriented Component-Based Approach*. Oct. 2, 2005, 248 pp., Pages: 263.

[31] S. Chong *et al.*, "Model driven system engineering for vehicle system utilizing model driven architecture approach and hardware-in-the-loop simulation," in *2011 IEEE International Conference on Mechatronics and Automation*, ISSN: 2152-744X, Aug. 2011, pp. 1451–1456.

[32] G. Erixon, A. Von Yxkull, and A. Arnström, "Modularity – the basis for product and factory reengineering," *CIRP Annals*, vol. 45, no. 1, pp. 1–6, 1996.

[33] M. Sonego, M. Echeveste, F. S. Fogliatto, L. M. Tonetto, and C. S. t. Caten, "MODULAR FUNCTION DEPLOYMENT ADAPTED," in *DS 84: Proceedings of the DESIGN 2016 14th International Design Conference*, Dubrovnik - Croatia, 2016, pp. 1407–1416.

[34] A. Ericsson and G. Erixon, *Controlling Design Variants: Modular Product Platforms*. Society of Manufacturing Engineers, 1999, 162 pp.

[35] F. Rossi, S. Arfelli, S. J. Hu, T. A. M. Tolio, and T. Freiheit, "A systematic methodology for the modularization of manufacturing systems during early design," *Flexible Services and Manufacturing Journal*, vol. 31, no. 4, pp. 945–988, Dec. 2019.

[36] U. Katzke, B. Vogel-Heuser, and K. Fischer, "Analysis and state of the art of modules in industrial automation," *Automatisierungstechnische Praxis (atp)*, vol. 46, no. 1, p. 23, 2004.

[37] B. Vogel-Heuser, J. Fischer, S. Rosch, S. Feldmann, and S. Ulewicz, "Challenges for maintenance of PLC-software and its related hardware for automated production systems: Selected industrial case studies," in *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, Bremen, Germany: IEEE, Sep. 2015, pp. 362–371.

[38] VEREIN DEUTSCHER INGENIEURE, *Verfahrenstechnische anlagen - modulare anlagen - grundlagen und planung modularer anlagen*, version Blatt 1, Düsseldorf, Jun. 2019.

[39] VEREIN DEUTSCHER INGENIEURE, *Automatisierungstechnisches engineering modularer anlagen in der prozessindustrie - allgemeines konzept und schnittstellen*, Düsseldorf, Oct. 2019.

[40] L. Bittorf, J. Oeing, T. Kock, R. Garreis, and N. Kockmann, "Design of module type package services for modular downstream units and process analytic technology," *Chemical Engineering & Technology*, vol. 46, no. 7, pp. 1502–1510, 2023, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/ceat.202200390.

[41] NAMUR, *Anforderungen an die automatisierungstechnik durch die modularisierung verfahrenstechnischer anlagen*, version NE 148, Leverkusen, Oct. 2013.

[42] D. Williamsson and U. Sellgren, "An approach to integrated modularization," *Procedia CIRP*, vol. 50, pp. 613–617, 2016.

[43] A. Mollajan and S. H. Iranmanesh, "Modularisation of system architecture to improve system recoverability: A unique application of design structure matrix," *Journal of Engineering Design*, vol. 32, no. 12, pp. 703–750, Dec. 2, 2021.

[44] Kreimeyer. "Basic design structure matrix (DSM) as example for process structure with corresponding graph of process." (Feb. 1, 2009), [Online]. Available: https://commons.wikimedia.org/wiki/File:DSM_Tutorial_BasicDSM.jpg (visited on 08/19/2024).

[45] C. I. Gutierrez Fernandez, "Integration analysis of product architecture to support effective team co-location," Thesis, Massachusetts Institute of Technology, Massachusetts, Jun. 1998.

[46] R. E. Thebeau, "Knowledge management of system interfaces and interactions from product development processes," Master's thesis, Massachusetts Institute of Technology, Feb. 2001.

[47] Z.-k. Li, S. Wang, and W.-w. Yin, "Determining optimal granularity level of modular product with hierarchical clustering and modularity assessment," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 41, no. 8, p. 342, Jul. 30, 2019.