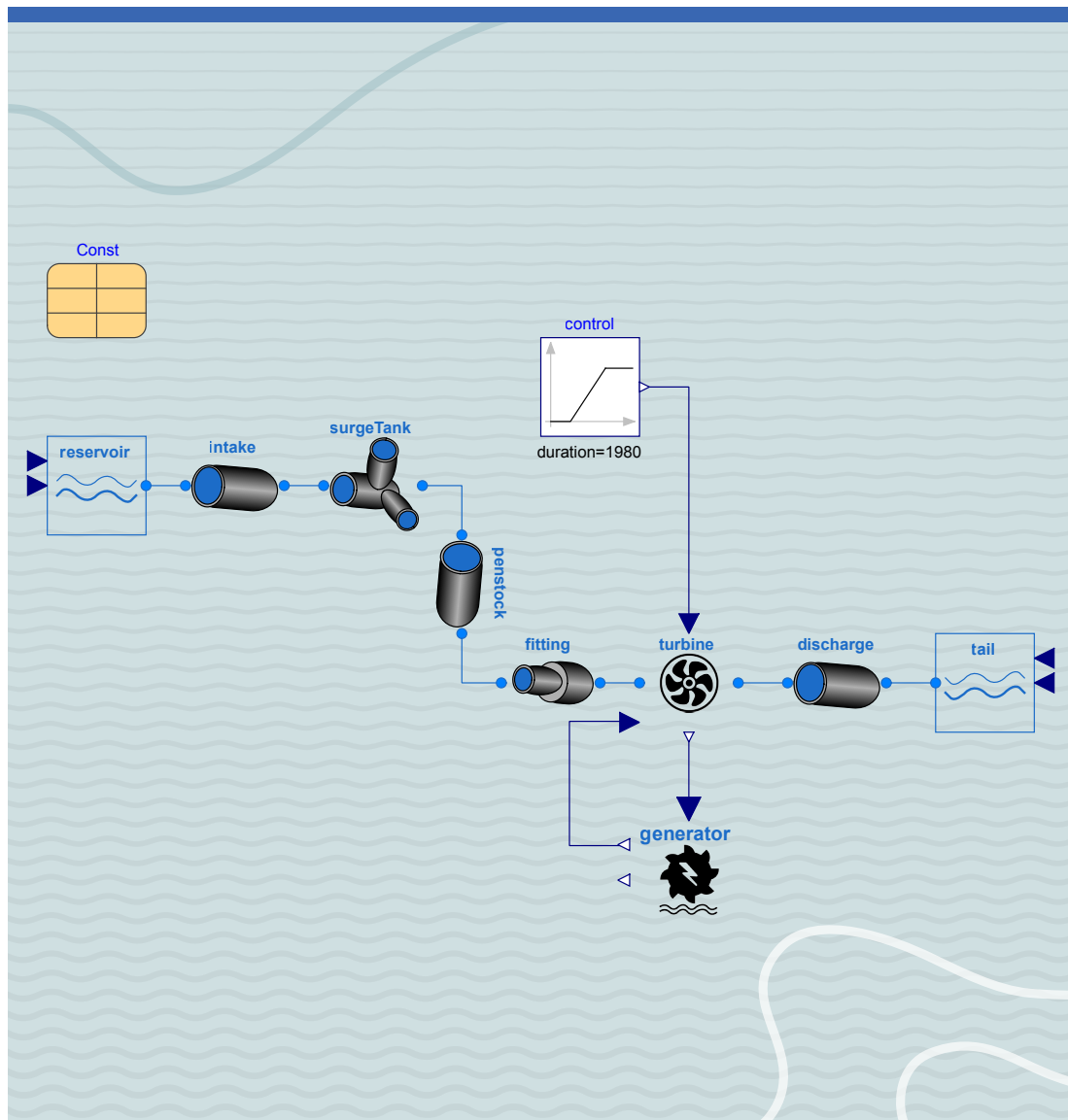


Liubomyr Vytvytskyi

Dynamics and model analysis of hydropower systems





Liubomyr Vytvytskyi

**Dynamics and model analysis
of hydropower systems**

A PhD dissertation in
Process, Energy and Automation Engineering

© 2019 Liubomyr Vytvytskyi

Faculty of Technology, Natural Sciences and Maritime Studies
University of South-Eastern Norway
Porsgrunn, 2019

Doctoral dissertations at the University of South-Eastern Norway no. 37

ISSN: 2535-5244 (print)

ISSN: 2535-5252 (online)

ISBN: 978-82-7206-533-0 (print)

ISBN: 978-82-7206-534-7 (online)



This publication is, except otherwise stated, licenced under Creative Commons. You may copy and redistribute the material in any medium or format. You must give appropriate credit provide a link to the license, and indicate if changes were made.

<http://creativecommons.org/licenses/by-nc-sa/4.0/deed.en>

Print: University of South-Eastern Norway

Preface

This thesis is submitted to the University of South-Eastern Norway (USN) as partial fulfillment of requirements for the degree of Philosophiae Doctor (PhD) in the Process, Energy, Automation Engineering program. The work has been funded by the Norwegian Ministry of Education and Research from August 2016 to August 2019. The PhD project takes place within two research groups in USN: Telemark Modelling and Control Center (TMCC) and Hydropower, Transmission and Distribution, which is a part of a cooperation with the Norwegian Research Centre for Hydropower Technology (HydroCen).

This research project has been under the supervision of Professor Bernt Lie and with co-supervision by Associate Professor Roshan Sharma. Most of the work has been carried out at USN, with some assistance from Skagerak Kraft AS in practical matters.

The thesis is article based, i.e., a collection of scientific papers, and contains two main parts. An introductory section with a literature review and overall discussion in relation to this PhD project are provided in the first part. The second part consists of seven scientific papers that lay the foundation for this thesis.

The work is mainly related to the development of a hydropower Modelica library that includes various models of the hydropower system. Furthermore, a few different analysis tools are implemented and tested. I hope that the developed hydropower library and its analysis tools will be used for solving hydropower problems both in academia and in industry in the future.

Porsgrunn, 8th August 2019

Liubomyr Vytvytskyi

Acknowledgements

First of all, I would like to express my most sincere gratitude and thanks to my main supervisor, Bernt Lie, for his guidance and support during these years. I would also like to thank my co-supervisor, Roshan Sharma, for his constructive advice and discussions. Thanks to both of my supervisors, I managed to complete this PhD project successfully and on time.

Special thanks to Dietmar Winkler (Assistant Professor at USN) for his practical advice and assistance in the hydropower library development and for sharing knowledge on hydropower in general. I am grateful to Gunne John Hegglid (Adjunct Professor at USN and CTO at Skagerak Energi) for sharing his expert knowledge on processes in hydropower systems. Thanks to Dr Thomas Øyvang for sharing his knowledge on generator modelling.

I would like to express my appreciation to Dr Beathe Furenes and Ingunn Granstrøm at Skagerak Kraft for their help. Beathe assisted me with the hydrology model and provided all data needed for the runoff estimation of the Sundsbarm hydropower plant. Ingunn contributed to the testing and validation of the Francis turbine model and its design algorithm. She provided the turbine design data found using *Alab*. My gratitude also goes to Erik Jacques Wiborg at Statkraft Energi, who provided the experimental and other related data from the Trollheim power plant for the case study.

Special thanks to my friend and fellow PhD student Ole Magnus Brastein for hours of discussions, various ideas, and motivation in my work. My gratitude also goes to faculty members and fellow PhD students for their help in my work and in everyday life.

Finally, I am grateful to my parents, Volodymyr and Liudmyla, my brother, Ostap, and my sweetheart, Nataliia, for their invaluable support and love.

Abstract

Globalization and growth in energy consumption in combination with climate challenges, make the use of sustainable energy production a necessity. Important sustainable energy sources (solar, wind) are intermittent in character, and hydropower from high head water reservoirs is a possible energy source for balancing out the variation in intermittent sources.

The combination of several energy types as well as counteracting the intermittent energy production make energy management and control much more challenging; optimal energy production will require advanced control based on detailed system models. Another side of globalization is the possibility for everyone to contribute to solutions. This requires low-cost software tools. The combination of open-source software tools with the publication of models and methods provides transparency in research, and increases the possibility of developing optimal and reliable models and methods.

In this thesis, a key contribution is the development of a library of mechanistic models for the waterway of hydropower production, using the modelling language Modelica with open-source tool OpenModelica. Another key contribution is the development of analysis methods using open-source languages Python and/or Julia.

Mathematical models for various units of a hydropower system are presented and assembled in a hydropower Modelica library — *OpenHPL*. Detailed studies for modelling different hydropower units are demonstrated in this work, i.e., the study of different complexities of waterway models, and the study of a mechanistic Francis turbine model and its design algorithm. The library can be used in an open-source software OpenModelica, which makes it freely and widely available. In addition, the library also supports modelling and simulation in the commercial environment Dymola.

The *OpenHPL* makes it possible to model different hydropower systems and connect them with models from other libraries, e.g., with models of the power system or other power generating sources. This synergy of our library with other Modelica libraries makes it possible to model the hydropower system starting from precipitations/reservoir to final consumer of electricity. The application and validation of the developed library are presented for case study measurements from the real Trollheim hydropower plant.

Various model analysis tools are implemented and tested for the *OpenHPL* hydropower models. These analysis tools are encoded in open-source scripting languages Python

and Julia. The practical use of existing APIs for running and controlling simulations of OpenModelica models via Python or Julia is presented.

One of the studied analysis tools is an automatic linearization that is provided by the APIs, and is tested for the hydropower models of different complexity. Opportunities to use the linearized hydropower models for control analysis and synthesis are demonstrated.

Another studied analysis tool is a state estimation that provides the possibility to estimate quantities in the hydropower system that are of interest and cannot be directly measured. The use of stochastic and deterministic approaches for the state estimation are demonstrated. Two types of nonlinear Kalman filters, UKF and EnKF, are studied and tested for the hydropower models for estimating the pressure and flow rate in various positions. Alternatively, the reduced order nonlinear observer is developed for the hydropower system.

The last studied analysis tool is related to a structure analysis of the linearized hydropower models. This structure analysis is implemented based on directed graphs and provides structural observability/controllability, which in turn gives a necessary requirement for actual observability/controllability. It is also shown how the developed structure analysis of the linearized hydropower models can be used for analysis related to state estimation and control: observability is a requirement for state estimators to work properly, controllability is required for control design, and relative degree is important in the design of nonlinear feedback controllers.

Contents

Preface	i
Acknowledgements	iii
Abstract	v
Contents	viii
I Overview	1
1 Introduction	5
1.1 Background	5
1.2 State of the art	6
1.3 Objectives and scope	7
1.4 Contributions	8
1.5 List of publications	9
1.6 Outline	10
2 Problem overview	13
2.1 System description	13
2.2 Literature review	14
2.2.1 Hydropower modelling	14
2.2.2 Model analysis tools	19
3 Methods	23
3.1 Modelling in Modelica	23
3.2 Solving PDEs	24
3.3 Turbine Euler equation and design algorithm	25
3.4 State estimation techniques	25
3.5 Structural observability and controllability	26
4 Summary of the papers	27
4.1 Paper A - Comparison of penstock models	27
4.2 Paper B - Francis turbine model	28

Contents

4.3	Paper C - Automatic model linearization	28
4.4	Paper D - Combining Measurements with Models	29
4.5	Paper E - Nonlinear observer for hydropower system	30
4.6	Paper F - Structural analysis	31
4.7	Paper G - Case study	32
5	Unpublished results - Hydrology model	33
5.1	Hydrology model	33
5.1.1	Snow routine	33
5.1.2	Soil moisture routine	35
5.1.3	Runoff routine	36
5.2	Results and discussion	37
6	Conclusions and future work	41
6.1	Hydropower library	41
6.2	Model analysis tools	42
6.3	Future work	43
	Bibliography	45
II	Scientific papers	53
A	Comparison of penstock models	55
B	Francis turbine model	67
C	Automatic model linearization	75
D	Combining Measurements with Models	83
E	Nonlinear observer	107
F	Structural analysis	119
G	Case study	131

Part I

Overview

Nomenclature

The nomenclature and abbreviations used in the thesis are listed below:

Symbol	Explanation
ANN	Artificial Neural Network
API	Application Programming Interface
AS	Norwegian, Aksje Selskap (company with limited (Ltd) responsibility)
CFD	Computational Fluid Dynamics
DAE	Differential Algebraic Equation
EKF	Extended Kalman filter
EnKF	Ensemble Kalman filter
EPFL	French, École polytechnique fédérale de Lausanne
FME	Norwegian, Forskningscentre for Miljøvennlig Energi
HBV	Swedish, Hydrologiska Byråns Vattenbalansavdelning
HPP	Hydro Power Plant
HydroCen	Norwegian Research Centre for Hydropower Technology
KF	Kalman Filter
KP	Kurganov-Petrova
MOC	Method of Characteristic
MPC	Model predictive control
ODE	Ordinary Differential Equation
OMJulia	Julia API for OpenModelica
OMPpython	Python API for OpenModelica
OpenHPL	Open Hydro Power Library
OpenIPSL	Open-Instance Power System Library
PDE	Partial Differential Equation
PDEModelica	Modelica language extension for 1-dimensional PDEs
PID	Proportional–Integral–Derivative
UKF	Unscented Kalman filter
WHAMO	Water Hammer and Mass Oscillation

1 Introduction

1.1 Background

A transition towards more renewable energy sources is currently happening in Europe and all over the world, with increasing use of flexible hydropower plants to compensate for the highly varying production from intermittent energy sources such as wind and solar irradiation. Flexible hydropower plants involve pumping water up into reservoirs during periods of surplus intermittent power, and high head storage is the most efficient way to store such surplus power.

Around 96% of generated electricity (138 GWh) in Norway is produced by hydropower systems, which have a total capacity in excess of 30 GW; over 1500 hydropower plants are in operation.¹ This makes Norway the 7th largest hydropower producer in the world.² The high head hydropower plants generate more than 75% of all electricity produced at hydropower plants in Norway.

In addition to hydropower plants, power production from wind power plants in Norway is increasing every year (13.4% increase from 2014 to 2015³) due to this technology becoming cheaper and more mature. With a long coastline, wind power has a huge potential for producing power in Norway. On the other hand, this renewable energy source is intermittent and creates considerable disturbances in the power grid. From this perspective, hydropower can be used to compensate for disturbances from wind power.

To optimize the combination of intermittent power and stored power, the possibilities for modelling and simulating the hydropower system play an important role in making an efficient analysis tool for testing a designed controller for stability and performance in different operating regimes. That is why the main part of this work is related to the research of hydropower modelling and simulation with the emphasis on hydropower library development using an open-source object-oriented, equation-based programming language — Modelica.⁴ In addition, the study and development of various analysis tools for hydropower models have been of interest.

¹<https://goo.gl/ftPPSk>

²<https://snl.no/vannkraft>

³<https://www.ssb.no/en/energi-og-industri/statistikker/elektrisitet/aar>

⁴<https://www.modelica.org/>

1.2 State of the art

Modelling of the hydropower system has been an important research topic in countries where hydropower generation plays a significant role, e.g., Norway, Sweden, China, Brazil, Switzerland, etc. Much work has been done on detailed CFD modelling of different units of the hydropower system to study what affects the hydropower plant operation, i.e., turbine efficiency, the water hammer effect, cavitation problems, etc. On the other hand, use of simplified hydropower models is considered for studies related to electricity production from the hydropower system, e.g., studies of power systems. Similar simplified hydropower models might be used for studies of hydropower control systems, however, the possibility of improvements with more details for these models are advantage.

A few tools that can be used for hydropower modelling already exist. Examples of existing software solutions include the WHAMO programme developed by the US Army Corps of Engineers, the LVTrans software which is used in NTNU Trondheim, TOPSYS tool used in Wuhan and Uppsala Universities, SIMSEN software developed by research group in the EPFL⁵, and the Hydro Power Library (Modelica library) developed by Modelon.⁶ However, some of these mentioned tools are based on analogue models (models represented by an electric circuit), others are not freely available due to commercial aspects, etc. A detailed description of each of these software tools is provided in Section 2.2.

As well as the waterway model of the hydropower systems, a hydrological model for forecasting a runoff to water reservoirs has a significant role in the electricity production. Various hydrological models exist in literature, e.g., the widely used HBV⁷ model. However, simulation of hydropower systems that consist of both hydrology and waterway parts is a challenging task due to a significant difference in timescales for the hydrology and waterway models.

Much work has also been done on modelling the electrical part of the hydropower system, e.g., models for the generator and other elements of the electrical grid. Moreover, a huge variety of tools/libraries for power system modelling and simulation exist and it is of interest to use those tools instead of developing new models of electrical elements. However, using such power system modelling libraries in combination with hydropower waterway models can be challenging, because they are not designed to interact.

A variety of tools for analysis of dynamic systems for control studies have been presented in the literature. Examples of these tools are automatic linearization, decomposition/de-coupling of the system, state and parameter estimation, structural analysis, etc.

Open-source software is becoming more and more popular. This is because free software can be developed in accordance with purely technical requirements and enables faster

⁵<https://lmh.epfl.ch/>

⁶<https://www.modelon.com>

⁷Hydrologiska Byråns Vattenbalansavdelning

innovation and more flexible technology. Not least due to the fact that open-source software is easy for everyone to use. The increasing focus on cost, especially in academia, also makes open-source software more important. On the other hand, open-source software has also some downsides, such as a less sophisticated user interface, it is slow to introduce new capabilities and has no debugging requirements, etc.

1.3 Objectives and scope

As described above, an increase of renewable energy sources with highly varying production leads to a greater need for flexible hydropower plants to ‘balance’ the intermittent production from renewable sources.. Hence, this study aims at developing a set of model units relevant for hydropower systems, as well as studying possible tools for analysis of these models for control purposes. These models and tools, in turn, can be used to improve hydropower control systems by making them more adaptable and enabling them to frequently changing power generation.

The hydropower unit models will be encoded in a Modelica library: Modelica is a multi-domain and component-oriented modelling language that is suitable for complex system modelling. This hydropower library should be able to interact with other libraries ‘downstream’ from the generator, e.g., including transmission and consumption. In order to develop the library, OpenModelica⁸ is used as an open-source Modelica-based modelling and simulation environment.

A commercial Hydro Power Library from Modelon exists, but the goal to develop a new Modelica library is to make it open-source and sufficiently rich to enable realistic studies in academia. In addition, it is of interest to include in the library some advanced models of various hydropower units (e.g., various pipe fitting, mechanistic turbine models, etc.), a method of discretization of PDEs for more complex models of conduits with elastic walls and compressible water (e.g., the Kurganov-Petrova scheme) and also models for other types of hydropower plants (e.g., open channel model for run-of-river hydropower plants). In order to enable the study of the entire chain of the waterway, it is also of interest to develop a hydrological model as a tool to describe the transport from precipitation in the catchment of a reservoir, to the reservoir.

Developing various methods for efficient analysis of the models such as decomposition into different timescales, decoupling into subsystems, and similar is also of aim of this study. The developed methods should be implemented as tools. A possible way to do this, is to develop the tools in Python⁹ or Julia¹⁰ (both are powerful open-source programming languages), which can be interfaced to OpenModelica via an API. Examples of

⁸<https://openmodelica.org/>

⁹<https://www.python.org>

¹⁰<https://julialang.org>

1 Introduction

tools for analysis should include automatic linearization, interface to Python/Julia control libraries, graph-based large-scale analysis tools, as well as tools for analysing decomposition/decoupling of the system, control structures, etc.

The developed models/libraries, methods and tools should be tested in at least one case study, preferably several case studies. The possibility of doing such a case study with Skagerak Energi AS has been considered, but other possibilities such as the FME HydroCen are also of interest.

1.4 Contributions

The contributions of this thesis are divided into three parts:

- A hydropower library: *OpenHPL* has been developed which provides the capability for the modelling of hydropower systems of different complexity. The library development includes the following units:
 1. Various waterway units are modelled based on mass and momentum balances, i.e., reservoirs, conduits, surge tank, fittings. A modern method for solving more detailed models (PDEs) is implemented in the library, and enables the modelling of waterway with elastic walls and compressible water as well as open channel.
 2. A hydrology model has been implemented and makes it possible to simulate the water inflow to reservoirs.
 3. Mechanistic models as well as simple look-up table turbine models are implemented for the Francis and Pelton turbine types. The Francis turbine model also includes a turbine design algorithm that gives all needed parameters for the model, based on the turbine's nominal operating values.
 4. The capability for multiphysics connections and work with other libraries is ensured, e.g., connecting with the Open-Instance Power System Library *OpenIPSL* makes it possible to model the electrical part for the hydropower system.
- Model analysis tools are tested for the hydropower library both in Python and Julia using existing OpenModelica APIs for these programming languages. These tools/APIs include:
 1. Automatic linearization (available in APIs) that has been used and tested to provide linear approximation of hydropower models with the standard time invariant matrices A , B , C and D . This linearized hydropower model has been used in control studies.

2. State estimation with nonlinear Kalman filters for the developed hydropower models. These estimation techniques can be used for state feedback controller design or to provide more information about the hydropower plant than is available solely from measurements, only. As an alternative, a nonlinear reduced order observer has been developed and tested for the hydropower model.
 3. Structural analysis based on graph theory. This tool is used to clarify the hydropower model structure and information about model controllability/observability, especially useful in cases with complex/large scale models.
- The application of *OpenHPL* to a real hydropower system is performed. The developed library is validated and tested by simulating a case study model of the Trollheim hydropower plant and by fitting the model to real experimental data. The *OpenHPL* has also been validated against a commercial Hydro Power Library from Modelon for the case of modelling the Sundsbarm hydropower plant.

1.5 List of publications

The research has resulted in papers for peer-reviewed journals and conferences:

- Paper A.** Vytvytskyi, L., & Lie, B. (2017). Comparison of elastic vs. inelastic penstock model using OpenModelica. *Proceedings of the 58th Conference on Simulation and Modelling (SIMS58)*, 138, 20–28. Linköping University Electronic Press. Doi:10.3384/ecp1713820.
- Paper B.** Vytvytskyi, L., & Lie, B. (2018). Mechanistic model for Francis turbines in OpenModelica. *IFAC-PapersOnLine*, 51(2), 103–108. Doi:10.1016/j.ifacol.2018.03.018.
- Paper C.** Vytvytskyi, L., & Lie, B. (2018). Linearization for Analysis of a Hydropower Model using Python API for OpenModelica. *Proceedings of the 59th Conference on Simulation and Modelling (SIMS59)*, 153, 216–221. Linköping University Electronic Press. Doi:10.3384/ecp18153216.
- Paper D.** Vytvytskyi, L., & Lie, B. (2019). Combining Measurements with Models for Superior Information in Hydropower Plants. *Flow Measurement and Instrumentation*. Doi:10.1016/j.flowmeasinst.2019.101582.
- Paper E.** Vytvytskyi, L., Sharma, R., & Lie, B. (2019). Nonlinear observer for hydro-power system. Accepted in *Modeling, Identification and Control*.
- Paper F.** Vytvytskyi, L., & Lie, B. (2019). Structural analysis in Julia for dynamic systems in OpenModelica. Accepted for *the 60th Conference on Simulation and Modelling (SIMS60)*.

1 Introduction

Paper G. Vytvytskyi, L., & Lie, B. (2019). OpenHPL for Modelling the Trollheim Hydropower Plant. *Energies*, 12(12), 2303. Doi:10.3390/en12122303.

Other contributions during the PhD study are as follows:

- Splavska, V., Vytvytskyi, L., & Lie, B. (2017). Comparison of Mechanistic and Table Look-up Turbine Models. *Proceedings of the 58th Conference on Simulation and Modelling (SIMS58)*, 138, 368–373. Linköping University Electronic Press. Doi:10.3384/ecp17138368.
- Vytvytskyi, L. & Lie, B. (2018). Developing a Hydropower Modelica Library HydroPowerUSN. *The 10th Annual OpenModelica Workshop, 5 February 2018. Linköping University, Sweden*. Presentation.
- Vytvytskyi, L. (2019). Innovation and digitalization in hydropower production. *The 10th annual Subsea Valley (SSV) Conference, “energy:connected’19”, 9–11 April 2019. Telenor Arena, Fornebu, Norway*. Presentation.

1.6 Outline

The structure of this thesis consists of two main parts. In the first part, the necessary background to this research is presented. The second part provides all the scientific papers selected for this thesis. The structure of the first part is as follows:

Chapter 2 gives a problem overview of this study. First, a detailed description of the structure of a hydropower system is demonstrated to give the reader appropriate knowledge about such systems. Then, an extensive literature review is presented to indicate the current state of the art for the topics of this thesis.

Chapter 3 provides information about the methods that have been used for this research. First, common rules for modelling and library development in Modelica are discussed. Then other methods used in the thesis, such as a scheme for discretizing PDEs, equations for Francis turbine modelling, state estimation techniques, etc., are briefly described. The description of each of the methods used cites the relevant publication in which the methods have been discussed in details.

Chapter 4 describes the contribution of each scientific paper selected for this thesis. Short summaries of each selected paper are given in this chapter.

Chapter 5 provides the work on hydrology modelling that has been done during the PhD study and has not been published. First, a description of a hydrology model is given. Then, simulation results of this hydrology model and its comparison to the real data are presented.

Chapter 6 summarizes the thesis with the conclusions of the studies performed. Finally, proposals for future work are described.

2 Problem overview

2.1 System description

A hydropower station converts the kinetic and/or potential energy of water into electricity. In mountainous countries (Norway, Nepal, New Zealand, Switzerland, etc.), this water is typically collected and stored in reservoirs, with tunnels leading the relatively small flow of water down a considerable height difference to the aggregated turbine and generator. In countries dominated by large rivers (Ukraine, China, Canada, Sweden, Brazil, Paraguay, Russia, Mozambique, etc.), dams are built across the river to provide a similar reservoir, but here the height difference is much smaller, while the water flow rate is much higher. The study of high head systems is the main focus.

A typical structure for a high head hydropower plant is depicted in Figure 2.1. Here, water is transported from a reservoir, where it is stored, through a tunnel known as an intake race/conduit/headrace closer to the powerhouse, where the turbine and generator are installed. The intake race can be a considerable length, normally with a small slope. After the intake race, the water flows down a steep slope with a relatively small flow rate to the turbine through a pipe known as the penstock or pressure shaft. There is a large pressure change in this pipe due to the big height difference. Moreover, the water hammer (hydraulic shock) phenomenon can occur in the penstock due to the sudden reduction of water flow in the system. The water hammer is simply a pressure wave propagation/pulsation in the pipe due to a momentum change. Hence, a surge tank may be installed at the point where the penstock is connected to the intake race, in order to reduce the water hammer effect (pressure pulses) and keep the water mass oscillation within limits. This surge tank can be constructed as a pipe and is then denoted a surge shaft, but can also be variations such as an open or closed volume. It is also relatively common to use a surge tank downstream from the turbine in order to decouple the mass of the tail race from the rest of the mass.

Two types of turbine are common for high head hydropower systems. When the height difference is large (300–4000 m), a Pelton turbine is used with a relatively small flow rate ($< 30 \text{ m}^3/\text{s}$). Alternatively, with smaller height differences (40–600 m) and larger flow rates ($< 1000 \text{ m}^3/\text{s}$), Francis turbines are preferred. The turbine is rotating a shaft which also holds the rotor of the generator producing electricity to the power grid. Normally, a synchronous generator is used.

2 Problem overview

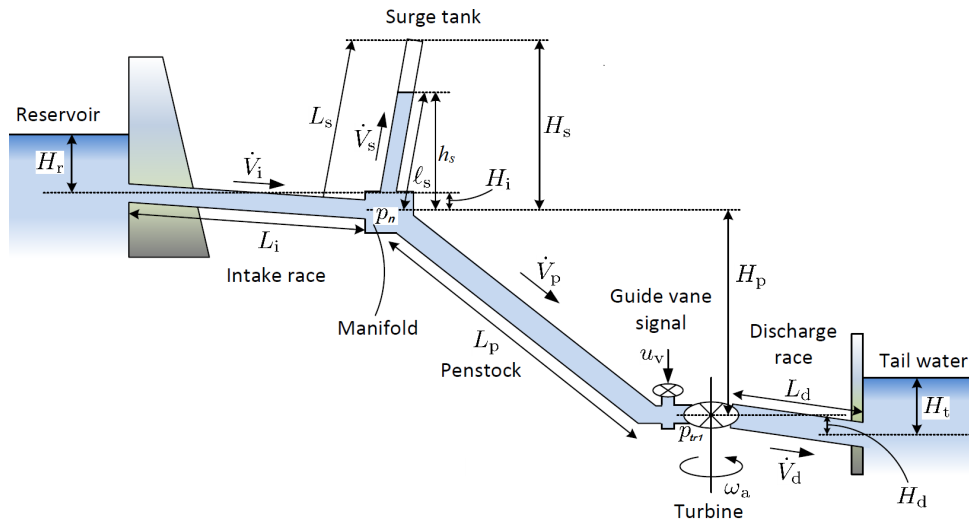


Figure 2.1: Structure of the hydropower system [1].

After the turbine, the water can flow directly to a downstream lake or river, often denoted as tail water, or the water can flow through a discharge/tail race if the powerhouse is situated some distance from the tail water. Normally, a Francis turbine is installed below the level of the tail water in order to increase the pressure at the outlet of the turbine to reduce the possibility of cavitation.

2.2 Literature review

This section provides an extensive literature review of the subject of this thesis. The section consists of two main parts, where the current state of the art for modelling of the hydropower system as well as available hydropower libraries and tools for modelling and simulation are described first. Next, an overview of the model analysis techniques and tools is presented.

2.2.1 Hydropower modelling

Hydropower system modelling can mainly be divided into the following components: hydrology or runoff modelling, waterway modelling, turbine modelling, and finally modelling of the electrical part including generator, grid, governor, etc.

Precipitation

Precipitation takes a key place in the hydrological cycle and in a water balance equation. Precipitation and air temperature information for a catchment area are the main inputs to most hydrology models [2]. That is why it is important to have good meteorological predictions for precipitation and air temperature in order to provide an accurate forecast of runoff to reservoir. Moreover, correct measurements of these precipitations and air temperatures are also crucial for calibrations and improvements in hydrology models. It is possible to purchase ensembles of meteorological predictions as well as meteorological archive data from the Norwegian Meteorological Institute¹ or the Norwegian Water Resources and Energy Directorate.²

Hydrology

A hydrological model for forecasting a runoff to reservoir in a catchment area plays a significant role in the electricity production from hydropower stations. Common examples of the hydrological model are different versions of the HBV model, as proposed in [3]–[5]. Some of these runoff predictions based on the mass balance and physical laws have been presented in [6], [7]. The HBV model is a widely used model in parts of the world with precipitation in the form of snow, and is used for hydrological forecasting and runoff simulation, for example as a tool when designing dams for the hydropower industry [8]. However, a large number of other hydrology models are also available and provide their own unique characteristics and respective applications. A brief discussion of some of them is given in [9].

Reservoir

Normally, dynamics of water storage reservoirs are considerably slow, with long time-frames for water level varying. Consequently, a reservoir can simply be assumed to be an open water tank with inlet and outlet water flows, and described by mass and momentum balances for water level; or even simpler as a tank with a constant water level. In order to get more detailed reservoir dynamics, a reservoir might be represented as an open channel and described as a Saint Venant model that consists of PDEs, [10], [11]. Possible discretization methods for simulating this open channel model are described in the next subsection.

¹<https://www.met.no>

²<https://www.nve.no>

Waterway

In general, waterway dynamics of high head hydropower systems can be represented by the model of the water flow in the pipe. Hence, the mass and momentum balances are used for the modelling of the waterway units [12]–[14]. Depending on the desired level of accuracy, models of the waterway unit can be described by ordinary differential equations (ODEs) in cases with incompressible water and inelastic walls. Partial differential equations (PDEs) appear for the more realistic cases with elastic walls and compressible water [15], e.g., for units with large pressure variations.

A dynamic model for the more detailed description of the waterway dynamics has been developed and studied in a previous work [14], where the Staggered grid scheme [16], [17] was used for discretization of a part of the model with PDEs. Alternatively, PDEs can be solved by the Method of Characteristics (MOC) [18], [19], by the orthogonal collocation method [20]–[22], or by using other Finite Volume methods [23]–[25]. One scheme for discretization is the Kurganov-Petrova (KP) scheme, described in [10], [26]. Comparison of the KP scheme with the Staggered grid scheme for an open channel model using MATLAB was done in [11] and showed superior performance for the KP scheme.

In [27], a Modelica extension for modelling with partial differential equations, PDEModelica, was proposed. An updated version of PDEModelica with implementation for OpenModelica was recently presented in [28]; however this extension is not fully completed.

The waterway models presented as a transfer function using the Laplace transformation have also been presented in literature [13], [29], [30]. These models can therefore be implemented in MATLAB/Simulink^{3,4} for simulation. Moreover, the description of how the waterway models can be represented as analogue models is given in [31], [32].

The surge tank is a principal part of the hydropower waterway that reduces the water hammer pressure and keeps the water mass oscillation within limits. Similarly to other waterway units, the surge tank can be modeled using the mass and momentum balances [12]–[14]. However, a closed surge tank might be considered in some cases and then its model becomes more complicated and includes hydraulics and heat transfer calculations [33].

In the case of run-of-river hydropower plants, the waterway is simply an open channel and can be modelled by the Saint-Venant equations. The use of this open channel model has been presented in previous studies [11], [34].

³<https://www.mathworks.com/products/matlab.html>

⁴<https://www.mathworks.com/products/simulink.html>

Turbine

The turbine is a vital part of the waterway, and this is where the potential-kinetic energy of the water is converted to power that drives the rotation of the aggregate of the turbine and the generator. Previous works on the development of a mechanistic model of a Francis turbine based on the Euler turbine equation have been discussed in [1], [35]–[37]. In order to use a mechanistic model, the turbine geometry parameters are needed. These geometry parameters can be found using a turbine design algorithm. A hydraulic design of the runner and guide vanes of existing Francis turbines have been discussed in [38] and [39]. Both of these studies are mainly based on research work by Brekke and publications based on his experiences with developing and designing hydraulic turbines [40].

Alternatively, the Francis turbine can simply be modelled based on a valve characteristic and the energy balance, as shown in [41]. Moreover, this valve-based model might also be used for modelling other turbine types, i.e., Pelton or Kaplan turbines. However, this simplified model is not able to simulate very large transients of the turbine and has low accuracy for results of the turbine rotational speed.

Similarly to the Francis case, the Pelton turbine can also be modelled in detail based on the Euler turbine equation that has been discussed in [1], [35], [42]. This mechanistic model also requires the Pelton turbine geometry that can be found using the design algorithm developed in [42], [43].

Empirical turbine models also exist and require fitting to experimental data. These empirical models typically consist of look-up tables that show how turbine efficiency varies with flow rate, control input, etc.

Electrical part

A synchronous generator is a key component of the hydropower plant from the electrical point of view. Modelling and analysis of this generator type have been presented in various works [12], [44], [45]. The generator and other components of the electrical part, e.g., governor, electrical grid, loads, etc., have also been previously modelled and studied for the hydropower system in [13], [14], [46], [47].

Timescales

If a model is built to include units for precipitation, hydrological transport to reservoir, and waterways, the system exhibits multiple timescales, and this must be addressed in order to provide efficient solutions depending on the model usage. Some of these techniques, such as multirate methods and other approaches have been described for large

2 Problem overview

power systems [48]. Otherwise, the standard timescale decomposition can be performed [49].

Hydropower libraries

Various examples of software tools for hydropower modelling exists, and a basic overview of some of them has been given in [30]. WHAMO (Water Hammer and Mass Oscillation) is a modelling and simulation tool that provides dynamic simulation of fluid distribution systems and hydropower plants, developed by U.S. Army Corps of Engineers, and presented in [50]. WHAMO uses an implicit finite difference method for simulating differential equations, and PDEs in particular. The turbines in this application are based on turbine look-up tables, which requires fitting a model to experimental data or use of static characteristic curves for specific turbines.

The LVTrans⁵ tool is based on LabView⁶ for transient simulations of hydropower plants, which was developed by Bjørnar Svingen at Rainpower⁷, and is used at NTNU [51]. LVTrans uses a MOC solver for simulation of partial differential equations. Turbine units in LVTrans are described by physical models that only need the turbine nominal operating values and do not need look-up tables for simulation.

Another tool, TOPSYS from Wuhan and Uppsala Universities, is based on Visual C++ and provides models for various components of HPPs [46], [52]. This software allows simulation of elastic conduits with compressible water described by PDEs. Turbine models in TOPSYS are based on look-up tables.

SIMSEN⁸ is a simulation tool for the analysis of electrical power networks, adjustable speed drives and hydraulic systems, developed and used at EPFL [32]. This software uses analogue models (RLC components) for modelling hydraulic elements. Turbine units in SIMSEN are also modelled based on look-up tables.

The commercial Modelica Hydro Power Library⁹ from Modelon¹⁰ provides a framework for modelling and simulating hydropower plant operations in a commercial Modelica-based modelling and simulation environment — Dymola¹¹ [53], [54]. This Modelica library uses staggered grid for discretization and simulation of PDEs. Description of the turbines is based on look-up tables for this Hydro Power Library.

⁵<http://svingentech.no/index.html>

⁶<http://www.ni.com/en-no/shop/labview/labview-details.html>

⁷<https://www.rainpower.eu>

⁸<https://simsen.epfl.ch>

⁹<https://www.modelon.com/library/hydro-power-library>

¹⁰<https://www.modelon.com>

¹¹<https://www.3ds.com/products-services/catia/products/dymola>

Another program exists for the hydraulic design of various hydropower plant units — *Alab*.¹² However, *Alab* is essentially a *design* tool. It may be used to find turbine look-up tables.

Among these modelling and simulation tools for hydropower systems, only LVTrans is currently open-source software and freely available. However LVTrans is based on and requires the commercial tool LabView, and is thus not truly openly accessible.

Besides the hydropower libraries, a huge variety of software tools for power system modelling and simulation exists, e.g., commercial tools DiGSILENT PowerFactory¹³, PSS@E¹⁴, EMTP-RV¹⁵, etc. Various Modelica power system libraries are also available and an overview of some of them is given in [55]. Among these power system libraries, the Open-Instance Power System Library (OpenIPSL¹⁶) is of particular interest as it is an open-source library of power system component models written in the Modelica language that can be used for power system dynamic analysis [56].

2.2.2 Model analysis tools

The presence of intermittent energy sources such as wind and solar-based energy in the power grid dramatically increases the disturbance in energy systems. Increased disturbance necessitates the use of more advanced control algorithms in order to achieve the required performances in, for example, grid frequency and grid voltage. More advanced control algorithms for large parts of the grid will require a structural analysis of the system, system decomposition, etc. Therefore, a literature review on these analysis techniques and tools is presented here.

The use of open-source programming languages, i.e., Python or Julia, is considered in this study for developing various analysis tools. The free software languages enable faster innovation and more flexible technology [57].

A Python API for OpenModelica, OMPython¹⁷, already exists and provides possibilities for controlling simulations of OpenModelica models via Python [58]. Python in turn, gives much broader possibilities for plotting, analysis and optimization compared to what is possible in OpenModelica (e.g., using Python packages *matplotlib*, *numpy*, *scipy*, etc.).

The Julia API of OpenModelica, OMJulia, is similar to OMPython. The OMJulia package¹⁸ has the capability to run simulations in Julia [59]. Julia in turn gives rich possibilities

¹²<http://www.alab.no/Alab-Hydropower-Software>

¹³<https://www.digsilent.de/en/powerfactory.html>

¹⁴<https://goo.gl/15KJg6>

¹⁵<http://www.emtp-software.com/page/overview>

¹⁶<https://github.com/OpenIPSL/OpenIPSL>

¹⁷<https://www.openmodelica.org/doc/OpenModelicaUsersGuide/latest/ompython.html>

¹⁸<https://www.openmodelica.org/doc/OpenModelicaUsersGuide/latest/omjulia.html>

2 Problem overview

for plotting, analysis and optimization (e.g., using Julia packages *Plots.jl*, *LightGraph.jl*, *JuMP.jl*, etc.). In addition, models of various dynamic systems can be directly modelled in Julia using the *DifferentialEquations.jl*¹⁹ [60], [61] and the *ControlSystems.jl*²⁰ packages.

Automatic linearization

Even though almost all processes in the real world are nonlinear in nature, nonlinear theory for analysing nonlinear models is far less developed than the theory for linear models. Therefore, model linearization is important in order to have efficient analysis tools for nonlinear models, such as hydropower models.

Automatic linearization of OpenModelica models is available in the OMPython and OMJulia [58]. Basic mathematics and control theory needed to model, analyse and design feedback systems have been presented in [62]–[65]. Based on these methods, the linearized hydropower model can be further tested and analyzed for control purposes using a Python²¹ package — *python-control* (The Python Control Systems Library).²² *python-control* is a Python module, where basic operations for analysis and design of feedback control systems are implemented.

Similarly to the OMPython, OpenModelica with OMJulia can be used for the model linearization [59]. Alternatively, a dynamic system is modelled directly in Julia with the *DifferentialEquations.jl* package [60], [61], and can then be linearized using the *ForwardDiff.jl* package²³ [66]. In Julia, the *ControlSystems.jl* package can further be used for testing and analysis of the linearized model for control purposes.

State and parameter estimation

Full state information on the model of a dynamic system is needed to design an advanced feedback controller such as model predictive control (MPC) or state feedback control. In practice, states are not available, but must be inferred by combining measurements with a dynamic model. Such state estimation has a value in itself, i.e., without feedback control, in that state estimation can be considered a soft sensor for estimating quantities that can not be measured directly.

Modern state estimation theory appeared in the middle of the 20th century, and since then the data assimilation idea has spread to almost all areas of engineering and science.

¹⁹<https://github.com/JuliaDiffEq/DifferentialEquations.jl>

²⁰<http://juliacontrol.github.io/ControlSystems.jl/latest>

²¹<https://www.python.org>

²²<https://sourceforge.net/projects/python-control>

²³<https://github.com/JuliaDiff/ForwardDiff.jl>

A basic presentation of a variety of state estimation techniques (standard KF, EKF, UKF, etc.) for a general system is provided in [67]. The hydropower plant model can be highly nonlinear, and application of the widely used EKF can lead to difficulties in Jacobian computation. Moreover, the EKF relies on a linear approximation to propagate the mean and covariance of the states, and this can cause unreliable estimates. At the same time, the UKF and EnKF are much simpler to use as no Jacobian needs to be computed: the linearization problem is avoided by using the unscented transformation and Monte Carlo simulations of the nonlinear model, respectively [68], [69]. These two methods allow the approximation of the variation of the mean and covariance of random variables that propagate through the nonlinear model. An implementation of the UKF to estimate states of a hydropower plant that balances the uncertainty in pressure measurements, has been demonstrated in [70]. The use of the EnKF to predict runoff or groundwater flow for hydrology models that are slightly related to hydropower systems is given in [71] and [72].

State estimation is traditionally used with mechanistic models. However, state estimation can also be used with data-driven models, e.g., artificial neural networks (ANN). See for example [73], [74]. An application of ANN for estimation of hydropower plant water inflow is provided in for example [75], [76]. However, data-driven models provide a physical interpretation only for the inputs and outputs of the system, while for mechanistic models, internal states and auxiliary variables also have a physical interpretation. This means that with a mechanistic model, it is possible to find internal quantities in the system with some certainty. In data-driven models, internal quantities have no physical interpretation unless they are calibrated during an extended and complex experimental phase.

As an alternative to the KF based on a stochastic approach, an observer based on a deterministic approach can be used for the state estimation, and was first proposed by [77] for linear systems. Extension of the linear observer design for nonlinear systems has been presented in [78]. Although a huge variety of techniques for designing nonlinear observer exist [79]–[82], the design of nonlinear observers is still a challenging task.

Structural analysis

Model-based analysis of state observability and controllability is important for control design, and it is of interest to consider tools for aiding such analysis. Classically, observability and controllability properties might be checked using the well-known tests based on rank conditions [67], [83]. However, numerical problems can arise for the rank computations in complex, large-scale systems. Structural observability and controllability based on the system structure can be used in such cases due to the simplicity of these methods. In addition, the relative degree of the system indicates how directly control inputs affect outputs, and can also be expressed based on the system structure. Assuming linear mod-

2 Problem overview

els, analysis tools based on graph theory can be implemented in Julia, for example, using the *LightGraphs.jl* package.²⁴

Basic graph theory for different engineering applications is provided in [84]. Structural modelling and analysis of complex systems are described by [83], [85]–[88]. Based on this theory, large-scale systems can be further tested and analysed for control and parameter estimation purposes. In [89], such a state estimation and optimal control study for an industrial copper electrowinning process using Python and JModelica²⁵ is discussed.

²⁴<https://github.com/JuliaGraphs>

²⁵<https://jmodelica.org/>

3 Methods

3.1 Modelling in Modelica

Using open-source Modelica software facilitates the mathematical modelling of the dynamics of components from various domains. Moreover, such components from, for example, hydraulic, mechanical, electrical, control and other domains, can be connected together to create a multiphysical formulation that can be easily solved using a Modelica tool. Developed unit models can then be combined into subsystems, systems, or even architectures. The key to Modelica’s technical success is its support for a wide range of modelling formalisms that allow the description of both continuous and discrete behaviour framed in the context of hybrid differential-algebraic equations [90]. All this makes Modelica an object-oriented modelling language, with support for creating and reusing library units.

Using the Modelica language, a library for modelling the hydraulic part of the hydropower system — *OpenHPL* was developed. This hydropower library consists of drag and drop elements that describe different units of the hydropower system such as reservoir, conduits, surge tank and turbine. These elements can be connected and structured in a flow sheet in the same way as the hydropower plant, see the hydropower structure in Fig. 2.1 and the model flow sheet in Fig. 3.1. In this model flow sheet, all drag and drop elements can be specified with appropriate geometry of the hydropower plant by double-clicking on the elements. Connectors that join each element contain information about the pressure in the connector and mass flow rate that flows through the connector — similar to the connection in an electrical circuit with voltage and current, or similar to the idea of potential and flow in Bond Graph models.

A more detailed overview of the mathematical models and methods used in this library for the waterway components is given in selected Paper A, Part II, as well as in [91] and the library user guide [92]. Paper G, Part II in turn provides an overview of the whole library and shows the application of *OpenHPL* to a real hydropower system and validation of the library with real experimental data.

3 Methods

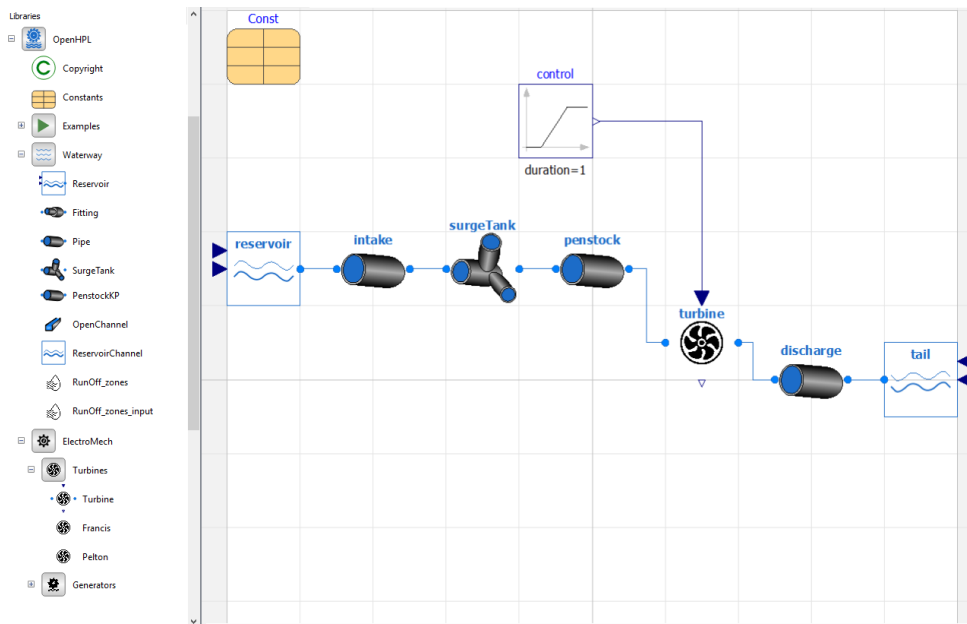


Figure 3.1: Flow sheet of the model of the hydropower system in OpenModelica. Modelled using *OpenHPL*.

3.2 Solving PDEs

In this work, some unit models are described by PDEs, a result of inclusion of additional details to these models such as water compressibility and pipe shell elasticity. Modelica in turn provides possibilities to solve ODEs and DAEs in time, however PDEs are not supported. Thus, this thesis includes studies for discretizing PDEs into DAEs, in other words: discretizing PDEs into semi-discrete models which are ODEs or DAEs in time. From a dynamical systems viewpoint, a PDE has an infinite-dimensional state space, which is approximated by the finite-dimensional state space of a high-dimensional ODE/DAE [93]. There are many discretization techniques for PDEs, but from previous work, [11], it was observed that the semi-discrete KP scheme shows good results for hyperbolic PDEs such as those involved in water flow. The KP scheme is a modern, high-quality and well-balanced scheme [10], and is chosen as a good candidate for PDEs discretization in Modelica.

A detailed description of the KP scheme together with some examples of use is given in [10]. An overview of the KP scheme implementation in Modelica and a comparison with simpler models are given in Paper A, Part II, as well as in the library user guide [92].

3.3 Turbine Euler equation and design algorithm

Instead of using an empirical turbine model that requires extensive experiments on real turbines, a simple mechanistic Francis turbine model based on the Euler turbine equation is considered in this thesis. Detailed descriptions of this Euler turbine equation and its use for the Francis turbine modelling are given in [1], [35].

The mechanistic turbine model still contains a number of design parameters which must be found. However, the design parameters have a physical interpretation, and algorithmic design rules exist for choosing these parameters based on information such as nominal water head and nominal flow rate for the hydropower plant. Such a design algorithm is provided in detail in [40].

A detailed overview of the mechanistic Francis turbine model implementation in Modelica with the Francis turbine design algorithm is presented in Paper B and Paper G of Part II.

3.4 State estimation techniques

Popular state estimation methods include the classic Kalman Filter (KF) with a wide range of extensions that apply depending on model type, computational effort, etc. Due to nonlinearities in the hydropower model, nonlinear types of KF such as UKF and EnKF are considered in this thesis. The classical nonlinear estimator, the EKF, is also widely used. However, the EKF assumes the existence of the model state Jacobian, and has relatively poor accuracy due to the linear approximation used in the Kalman gain computation. On the other hand, the UKF and EnKF take more advantage of the nonlinear model in the Kalman gain computation, which theoretically leads to better performances. Detailed descriptions of these nonlinear KF techniques and their application are provided in [67]–[69].

The KF is based on a stochastic system description. Alternatively, an observer based on a deterministic approach is also used for state estimation/observation in this thesis. This alternative is of particular interest for nonlinear systems such as the hydropower one, because the standard linear KF only provides an optimum solution for linear systems, [94]. In addition, a nonlinear observer can lead to significant reduction of computational effort compared to a KF. Such observers can quickly find an estimate of the states and provide these as an input to fast nonlinear control algorithms. On the other hand, the proof of convergence is non-trivial for nonlinear observers. A description and the uses of these nonlinear observers are given in [78]–[81].

A detailed overview of nonlinear state estimation techniques (UKF, EnKF, and reduced nonlinear observer) for the hydropower models and their implementation in Python using

the Python API OMPython for OpenModelica is discussed in Paper D and Paper E of Part II.

3.5 Structural observability and controllability

In control theory for dynamic systems, information about observability and controllability of states play a key role in evaluating the possibility to observe states from outputs, and use inputs to move states to a desired position, respectively. Automatic determination of observability and controllability is possible, in particular for linear models where observability and controllability gramians are typically considered. In the case of large scale systems, such as complex models of regional energy systems, standard analysis becomes challenging. For large-scale systems, structural analysis based on directed graphs is an interesting alternative: structural observability (or: controllability) is a necessary requirement for actual observability (or: controllability). Directed graphs can be set up directly for linear models, but can also be extracted from nonlinear models. More details of the structural modelling and analysis of complex systems are described in [83], [85], [89].

An overview of structural analysis for hydropower models and its implementation in Julia is provided in Paper F, Part II. Here, use of the *LightGraphs.jl* package for the work with graphs and the OMJulia package for OpenModelica model integration is also described.

4 Summary of the papers

The investigations of this thesis are described in detail in the following seven papers. Summaries of each of these papers are provided in this chapter. The complete papers are provided in Part II.

4.1 Paper A - Comparison of penstock models

In this paper, the comparison between different penstock models for the modelling and simulation of a high head hydropower system is explored. The Kurganov-Petrova second order central upwind scheme is implemented to discretize the more complicated penstock model with compressible water and elastic walls that is described by PDEs. The models for simulating the hydropower system with different scenarios have been developed and simulated in OpenModelica using *OpenHPL*.¹ This in-house hydropower library has been validated against an already existing commercial Hydro Power Library.

Based on three simulation scenarios, it is shown that the system with an inelastic penstock model exhibits somewhat simpler dynamics. However, it has the advantage that the simulation time is only one-third of the same system with an elastic penstock model (discretized by the KP scheme). It is also observed that the wave propagation, which is taken into account in the elastic penstock model, affects the system. Despite all these differences, the inelastic penstock model can perhaps be used for control design problems due to its simplicity, since simulation is less time-consuming and accuracy is reasonably good. Clearly, a model-based controller based on a simple model should be tested on a more rigorous model, and, if necessary, re-tuned in order to ensure good performance. For a more detailed and accurate design, the elastic penstock model could be used in the controller, due to better representation of the system dynamics resulting from the inclusion of the speed of the wave propagation in the model. Even better dynamic representation of the system can be reached by using the elastic model for the conduit. Then the delay caused by the speed of the wave propagation leads to more dramatic changes in the system dynamics.

Our limited experience in this study indicates that the simpler model appears to cover the worst case due to higher amplitude of the oscillation just after a disturbance, even

¹The library name has changed over the years and can differ in this and other papers.

4 Summary of the papers

though it does not show proper dynamics. The systems with elastic and inelastic models do not behave in the same way for the outlet turbine pressure, and the amplitude of the first oscillation after disturbance can differ. This can be very important for a cavitation study.

4.2 Paper B - Francis turbine model

The Francis turbine model based on the Euler equation given in [1] is presented in this paper. In order to simulate this mechanistic model, a set of turbine geometry data is needed to specify the model parameters. To this end, a turbine design algorithm that can calculate the geometry base on the nominal turbine parameters such as net head, flow rate and power production is also proposed.

The turbine design algorithm shows reasonable results for the turbine geometry parameters in comparison to the results from the *Alab* software. There is a small difference in the inlet runner width and blade angle, which is caused by a different philosophy of blade shape — the runner cross-section area increases from inlet to outlet by 10% for the *Alab* software, while it is the opposite for the design algorithm. However, these contrasts in the designed geometry are actually minor and the design algorithm can be used for defining all the geometry parameters for the mechanistic Francis turbine model. Exceptions apply to the friction term coefficients, which should be further studied in order to develop an algorithm for their calculation with respect to the turbine nominal operational values.

The presented Francis turbine model shows good results that fit the *Alab* model quite well, especially for the turbine shaft power. There is some deviation in the turbine efficiency curves for both case studies, which may be caused by differences in the design or efficiency and friction terms calculation. The *Alab* tool does not include the whole hydropower system dynamics for the calculation, while the mechanistic turbine model uses the system dynamics (it uses the real/dynamic net head of the system instead of the nominal static head). For a better comparison, the turbine efficiency from the mechanistic model has been calculated in the same way as in *Alab*. However, the simulations have been carried out with the whole hydropower plants (dynamic system), which has probably led to the deviation in the turbine efficiency curves. Nevertheless, this Francis turbine model can be further used for different studies of the high head hydropower system.

4.3 Paper C - Automatic model linearization

The possibility of automatic linearization of OpenModelica models through Python using the Python API is explored in this paper.

4.4 Paper D - Combining Measurements with Models

Two cases with different model complexity for the hydropower system are linearized in order to show the linearization capability of the Python API. Despite the model complexity, the linearization algorithm finds the state space matrices A , B , C , D .

After linearization, linear theory could be further used for model analysis and synthesis. Examples of analysis are presented by creating a Bode plot and designing a simple PI controller, using the already existing packages in Python (*numpy* and *python-control*). The Bode diagram has been plotted for two cases of the hydropower system in order to show the frequency response of the models. The possibility of PI controller design is then shown for the two cases. The designed PI controller is also tested for the original (nonlinear) models in OpenModelica.

Besides the presented examples of linear analysis, many more other possibilities for analysis and synthesis of the linearized model could be used, such as sensitivity or stability analyses, etc.

4.4 Paper D - Combining Measurements with Models

It is shown in this paper that state estimation based on the assimilation of a mechanistic model and measured data could be used to improve the information for hydropower plants. Nonlinear UKF and EnKF estimators are developed in Python via the OMPython for *OpenHPL*-based models in OpenModelica. The estimation results from both the UKF and EnKF are presented. It is clearly observed that the UKF and EnKF estimates show good results for the penstock flow rate. Specifically, the results are promising for the cases where the flow rate is not measured. The estimates from the UKF and EnKF for the surge tank flow rate and water height deviate more from the model simulations than the estimates for the penstock flow rate. However, the results for these other states are still useful.

In general, both the UKF and the EnKF appear to give good results when the estimators use correct information about covariance matrices — whether there is random disturbance or not. In real life, the ‘correct’ information about covariance matrices is not known.

When the estimator assumes a larger disturbance covariance than the real system has, the estimates become noisy. This is to be expected: assuming too large disturbance covariance is akin to assuming too low measurement noise covariance, which will lead to too much trust in the measurements, hence a too large Kalman gain, and ‘bleeding’ of measurement noise into the estimates. It is interesting to observe that the UKF appears to give poorer estimates than the EnKF for this case. A possible explanation for this is that since the EnKF uses more particles than the number of sigma points in the UKF, the EnKF tends to smooth out this noise in a superior way to that of the UKF.

4 Summary of the papers

Both the UKF and the EnKF give good results with a single pressure measurement (inlet turbine pressure or manifold node pressure); the UKF works fine with either pressure measurement, while the EnKF works best with the manifold node pressure measurement.

A case with two measurements, the turbine flow rate and the inlet turbine pressure, has also been considered. Combining two measurements leads to good performance for both estimators, and is an improvement on the single measurement. However, two measurements at different locations may be more informative, and should also be considered for a future study.

In almost all cases, the state covariance (uncertainty) converges faster for the UKF than for the EnKF.

Regarding the choice of estimator algorithm, it is hard to make a final decision. The UKF has an advantage for low order models with regard to computational speed, but both the UKF and the EnKF are straightforward to parallelize, and with modern multicore/multi-threading processors this advantage is perhaps not vital. The EnKF appears to handle incorrect assumptions about covariances better; this is also an important feature.

4.5 Paper E - Nonlinear observer for hydropower system

The design and proof of convergence for the reduced order nonlinear observer for the simplified hydropower system is presented in this paper. This observer is designed to estimate the surge tank flow rate using the measurements from two other states: the penstock flow rate and the surge tank water height. The observer was implemented in OpenModelica and added to our in-house hydropower library (*OpenHPL*). A number of simulations were then run in Python using the Python API for OpenModelica in order to study the designed observer.

The dynamic simulations were performed to show the observer behaviour under the presence of input disturbance. Moreover, the influence of the measurement noise on the observer was also presented for dynamic simulations. The simulation demonstrated that the performance of the reduced order nonlinear observer mostly depends on one observer gain. Furthermore, it has been shown that the observer with only the penstock flow rate measurement shows promising results that are not affected by the measurement noise.

Nevertheless, the model showed that in the case with measurements from the more detailed/realistic hydropower model, the observer with only one measurement gives a poor performance. Adding another measurement for the observer leads to better estimates (lower offset). This improvement shows that both measurements are important for the observer's performance, and that a trade-off is required between speed of convergence and ability to reduce noise (noise amplification).

This paper also shows that a reduced order nonlinear observer can be designed and used for state estimation in a hydropower system. The design of the observer is based on a simplified model with certain assumptions. This lowers the complexity in the design of the observer. When the observer is tested against the simplified model, the estimates are proper without any offsets. However, the observer is also tested against data from a more detailed/complex hydropower model where simplifying assumptions are not considered. Under such conditions, the observer has to be re-tuned. With a properly calibrated/tuned reduced order observer, the observer can still provide satisfactory estimates of the unmeasured state.

4.6 Paper F - Structural analysis

This paper explores the possibilities of using graph theory methods for structural analysis of dynamic systems. Although the chosen examples hardly qualify as complex/large-scale, graph methods scale up well to very large systems such as regional/national power grids. The presented methods are implemented in Julia using the *LightGraphs.jl* and *GraphPlot.jl* packages. Using the *OpenHPL* hydropower library in OpenModelica and OMJulia for OpenModelica, the structural analysis methods are tested on hydropower models of different complexity.

The results of testing the developed structural analysis functions look reasonable and can be further used for analysis related to state estimation and control: observability is a requirement for state estimators to work properly, controllability is required for control design, and relative degree is important in the design of (nonlinear) feedback controllers. One experience with the developed tools is that sometimes it can be hard to make a good visualization of the graph structure of a complex (large-scale) system: it can be hard to see the whole picture of the system structure (small subsystems are not easily seen) using circular layout for the graph plotting. However, the user can do some testing of different layout types for the graph plotting to find the most appropriate one. Moreover, the graph can be stored in a large image with a higher resolution in order to see the system structure more clearly. In addition, developers of *LightGraphs.jl* and *GraphPlot.jl* packages are planning to improve the plotting possibilities of graphs in future, for example by improving the display of self loop edges, etc.

In summary, this paper has explored some possibilities with structural analysis. Further work should be put into streamlining the functions into a package, with better use of Julia coding conventions, integration with other modelling tools, integration with control packages, etc.

4.7 Paper G - Case study

In this paper, the use of the Modelica hydropower library, *OpenHPL*, for hydropower system modelling is presented. The hydropower library provides hydropower system unit models of different complexity that can be used to create a model of the hydropower system with different accuracy, depending on the user's needs. Models of various complexity are presented in this study for the case of the Trollheim hydropower plant. The experimental data from the hydropower plant is also presented in order to validate the developed hydropower models.

The experimental data is used first for library element tuning, whereby the friction loss coefficients of the Francis turbine model (the *Francis* element) are tuned. In addition, the Francis turbine design algorithm presented in paper B is improved with analytical expressions for the friction loss coefficients as functions of the nominal turbine head.

Another parameter such as roughness height of pipe units of the modelled hydropower system is also tuned for each unit. The simulation showed that this roughness parameter affects the transient dynamics amplitude and damping of the inlet turbine pressure. Moreover, the draft tube unit presented by the simple *Pipe* element is also included in the hydropower model. This draft tube element affects the outlet turbine pressure dynamics and is also tuned. These tuned parameters are then used for the hydropower models of different complexity.

Finally, the hydropower model results are compared with the measured data. Despite the model complexity, all the hydropower models show good fitting with the experimental results. Some small deviations exist between the models and measured results, but these are believed to be relative insignificant.

5 Unpublished results - Hydrology model

The widely used HBV model for runoff/inflow forecasting is studied here [6], [7]. This hydrology model has been implemented in OpenModelica and added to our in-house hydropower library — *OpenHPL*. The developed hydrology model is also calibrated and validated with real data from a catchment area of the the Sundsbarm hydropower plant. The calibration and validation were carried out in Python using the Python API for running the developed hydrology model in OpenModelica.

First, a brief description of the HBV hydrology model is presented here. Then, the simulation results of the developed model are presented and compared to the real data.

5.1 Hydrology model

Similarly to many other hydrological models, the HBV model is based on the land phase of the hydrological (water) cycle, see Figure 5.1. The figure shows that the HBV model consists of four main water storage components connected in a cascade form. Using a variety of weather information, such as air temperature, precipitation and potential evapotranspiration, the dynamics and balances of the water in the presented water storages are calculated. Hence, the runoff/inflow from some defined catchment areas can be found [4], [7].

The model is developed for each water storage component to define dynamics and balances of the water. In addition, the catchment area is divided into elevation zones (usually not more than ten) where each zone has the same area. The air temperature and the precipitation are provided for each elevation zone. Hence, all calculations within each water storage component are performed for each elevation zone.

5.1.1 Snow routine

In the snow routine segment, the snow storage as well as snow melt are computed. This computation is performed for each elevation zone [4]. Using the mass balance [7], the change in the dry snow storage volume $V_{s,d}$, is found as follows:

$$\frac{dV_{s,d}}{dt} = \dot{V}_{p,s} - \dot{V}_{d2w} \quad (5.1)$$

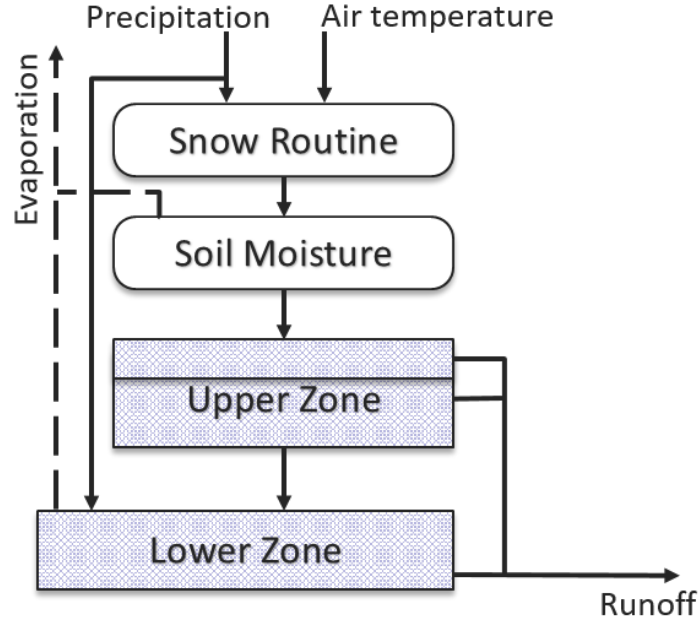


Figure 5.1: Structure of the HBV model.

Here, the flow of the precipitation in the form of snow is denoted as $\dot{V}_{p,s}$. This precipitation in the form of snow is defined from the input precipitation flow, \dot{V}_p , based on the information about the air temperature, T , a threshold temperature for snow melts, T_T , and for the area that is not covered by lakes (the fractional area covered by lakes, a_L , is used):

$$\dot{V}_{p,s} = \begin{cases} \dot{V}_p K_{CR} K_{CS} (1 - a_L), & \text{if } T \leq T_T \\ 0, & \text{if } T > T_T \end{cases} \quad (5.2)$$

Precipitation correction coefficients K_{CR} and K_{CS} are also used here, for the rainfall and snowfall precipitations, respectively. Then, the flow of the precipitation in the form of rain is defined as follows:

$$\dot{V}_{p,r} = \begin{cases} \dot{V}_p K_{CR} (1 - a_L), & \text{if } T > T_T \\ 0, & \text{if } T \leq T_T \end{cases} \quad (5.3)$$

The flow of the melting snow (melting of snow from dry form to water form), \dot{V}_{d2w} , can be found using the following expression based on the degree-day factor K_{dd} and the area of the elevation zone A_e :

$$\dot{V}_{d2w} = \begin{cases} A_e K_{dd} (T - T_T) (1 - a_L), & \text{if } T > T_T \text{ and } V_{s,d} > 0 \\ 0, & \text{otherwise} \end{cases} \quad (5.4)$$

Finally, the flow out of the snow routine to the next soil moisture segment, \dot{V}_{s2s} , is found as a sum of flows of precipitation in the form of rain, and the melted snow:

$$\dot{V}_{s2s} = \dot{V}_{p,r} + \dot{V}_{d2w} \quad (5.5)$$

It should be noted that a simplification related to the threshold temperature, T_T , is assumed here. This threshold temperature describes both the snow melt and rainfall to snowfall transition temperatures in the presented model. In reality, this threshold temperature might differ for each of these processes. In addition, a storage of snow in water form is not considered here, mostly due to the simplification with the threshold temperature.

5.1.2 Soil moisture routine

In the soil moisture segment, the water storage in the ground (soil) is found together with actual evapotranspiration from the snow-free areas [4]. The net runoff to the next segment (upper zone) is also defined here. Using the mass balance [7], the volume of the soil moisture storage, $V_{s,m}$, is found as follows:

$$\frac{dV_{s,m}}{dt} = \dot{V}_{s2s} - \dot{V}_{s2u} - \alpha_e \dot{V}_{s,e} \quad (5.6)$$

Here, \dot{V}_{s2u} is the net runoff to the next segment (the upper zone). $\dot{V}_{s,e}$ is actual evapotranspiration from the soil, that is taken into account only for the snow-free areas (zones). To define these snow-free zones, coefficient α_e is used and equals one for snow-free areas and zero for covered-by-snow areas. The actual evapotranspiration can be found from the potential evapotranspiration, \dot{V}_e , volume of the soil moisture storage, $V_{s,m}$, the area of the elevation zone A_e , and the field capacity — threshold soil (ground) moisture storage, g_T :

$$\dot{V}_{s,e} = \begin{cases} \frac{V_{s,m}}{A_e g_T} \dot{V}_e, & \text{if } V_{s,m} < A_e g_T \\ \dot{V}_e, & \text{if } V_{s,m} \geq A_e g_T \end{cases} \quad (5.7)$$

The potential evapotranspiration, \dot{V}_e , is defined as the input to the hydrology model, similarly to the air temperature and precipitations.

The output of the soil moisture segment — the net runoff to the next segment, \dot{V}_{s2u} , can be found based on the field capacity, g_T , as follows:

$$\dot{V}_{s2u} = \begin{cases} \left(\frac{V_{s,m}}{A_e g_T} \right)^\beta \dot{V}_{s2s}, & \text{if } 0 \leq V_{s,m} < A_e g_T \\ \dot{V}_{s2s}, & \text{if } V_{s,m} \geq A_e g_T \end{cases} \quad (5.8)$$

Here, β is an empirical parameter for specifying the relationship between the flow out of the snow routine, the soil moisture storage, and the net runoff from the soil moisture. Typically, $\beta \in [2, 3]$, which leads to a nonlinearity in Eq. 5.8.

5.1.3 Runoff routine

The upper and lower zones from Figure 5.1 are combined into one segment — the runoff routine. In this segment, the runoff from the catchment area is found based on the outflow from the soil moisture. The effects of the precipitation to and evapotranspiration from lakes in the catchment area are also taken into account here [4].

The upper zone characterizes components with quick runoff. The following mass balance is used for the upper zone description:

$$\frac{dV_{u,w}}{dt} = \dot{V}_{s2u} - \dot{V}_{u2l} - \dot{V}_{u2s} - \dot{V}_{u2q} \quad (5.9)$$

Here, $V_{u,w}$ is the water volume in the upper zone that depends on the saturation threshold, s_T , which defines the surface (fast) runoff, \dot{V}_{u2s} , and the fast runoff, \dot{V}_{u2q} . \dot{V}_{u2b} is the runoff to the lower zone and is defined by the percolation capacity, K_{PC} , for the area that is not covered by lakes:

$$\dot{V}_{u2l} = A_e(1 - a_L)K_{PC} \quad (5.10)$$

The surface runoff, \dot{V}_{u2s} , can be found using the the saturation threshold, s_T , and the water volume in the upper zone, $V_{u,w}$:

$$\dot{V}_{u2s} = \begin{cases} a_1(V_{u,w} - A_e s_T), & \text{if } V_{u,w} > A_e s_T \\ 0, & \text{if } V_{u,w} \leq A_e s_T \end{cases} \quad (5.11)$$

Here, a_1 is a parameter that represents the recession constant for the surface runoff. A similar recession constant, a_2 , is used for the fast runoff, \dot{V}_{u2q} , calculations:

$$\dot{V}_{u2q} = a_2 \min(V_{u,w}, A_e s_T) \quad (5.12)$$

The lower zone characterizes the lake and the ground water storages and defines the base runoff from the catchment area. The following mass balance equation is used for the lower zone description:

$$\frac{dV_{l,w}}{dt} = \dot{V}_{u2l} + a_L \dot{V}_p - \dot{V}_{l2b} - a_L \dot{V}_e \quad (5.13)$$

The water volume in the lower zone is denoted as $V_{l,w}$. As mentioned previously, \dot{V}_p and \dot{V}_e are the precipitation and the potential evapotranspiration flows, respectively. a_L is the fractional area covered by lakes. \dot{V}_{l2b} is the base runoff from the lower zone that can be found as follows:

$$\dot{V}_{l2b} = a_3 V_{l,w} \quad (5.14)$$

Here, a_3 is the recession constant similar to a_1 and a_2 .

The total runoff from the catchment, \dot{V}_{tot} , is a sum of the base, quick, surface runoffs for each elevation zones, and is defined as follows:

$$\dot{V}_{tot} = \sum_{i=1}^n (\dot{V}_{l2b,i} + \dot{V}_{u2s,i} + \dot{V}_{u2q,i}) \quad (5.15)$$

Here, the base $\dot{V}_{2b,i}$, quick $\dot{V}_{u2q,i}$, and surface $\dot{V}_{u2s,i}$ runoffs are first summed up for each of the n elevation zones and then these sums of the base, quick and surface runoffs are added together.

5.2 Results and discussion

The presented hydrology model is encoded and then simulated in OpenModelica. The input data for the hydrology model is provided for the catchment of the Sundsbarm hydropower plant for a 12-year period. The provided data includes everyday measurements for the air temperature, precipitation and potential evapotranspiration for each of the elevation zones. The actual runoff from the catchment is also provided. A description of the Sundsbarm catchment area is also available with the values for areas of each elevation zone and area of lakes. This catchment area description is given in Table 5.1.

Table 5.1: Catchment area for the Sundsbarm hydropower plant.

Height zone	From altitude, m.a.s.l.	To altitude, m.a.s.l.	Area, m ²	Lake area, m ²
1	603.0	702.0	41.3	15.43
2	702.0	743.0	41.3	3.97
3	743.0	793.0	41.3	1.79
4	793.0	826.0	41.3	0.81
5	826.0	868.0	41.3	1.27
6	868.0	917.0	41.3	1.44
7	917.0	967.0	41.3	1.03
8	967.0	1017.0	41.3	2.32
9	1017.0	1099.0	41.3	1.31
10	1099.0	1430.0	41.3	0.57
Total	603.0	1430.0	413.0	29.94

All model parameters are listed in Table 5.2, where a short description, value ranges and the tuned values for each parameter are provided. The tuned parameter values are found using the least squared error method for fitting the model results (runoff) with the real measured data.

Simulations of the hydrology model are run over a 12-year period, from 1 January 2005 to 31 December 2017, with a sampling time of one day. All volumes of water storage are initialized to zero except initial values of the volume of dry snow that are set to a small amount of snow for all elevation zones due to the simulations starting in the winter season. As mentioned above, the least squared error method is performed in order to calibrate the model parameters by fitting the model runoff with the measured data. The hydrology model simulation results for the runoff from the Sundsbarm catchment area are

Table 5.2: Parameters for the hydrology model.

Name	Description	Value range	Tuned value	Unit
T_T	Threshold temperature	[-1.0, 2.0]	-0.398	$^{\circ}\text{C}$
K_{dd}	Degree-day factor	[3.0, 6.0]	3.14	$\frac{\text{mm}}{^{\circ}\text{C}\cdot\text{Day}}$
K_{CR}	Precipitation correction for rainfall	[1.05, 1.2]	1.05	—
K_{CS}	Precipitation correction for snowfall	[1.15, 1.5]	1.2	—
g_T	Field capacity in soil moisture routine	[75, 300]	100	mm
β	Parameter in soil moisture routine	[1.0, 4.0]	2.0	—
s_T	Threshold level for quick runoff in upper zone	[10, 40]	40	mm
a_1	Recession constant for the surface runoff in upper zone	[0.05, 0.15]	0.122	$\frac{1}{\text{Day}}$
a_2	Recession constant for the quick runoff in upper zone	[0.1, 0.5]	0.42	$\frac{1}{\text{Day}}$
K_{PC}	Percolation from upper zone to lower zone	[0.5, 1.0]	0.6	$\frac{\text{mm}}{\text{Day}}$
a_3	Recession constant for the base runoff in lower zone	[0.0005, 0.002]	0.0016	$\frac{1}{\text{Day}}$

shown in Figure 5.2, where measurements of the runoff are also presented for comparison. The presented simulation results are taken from simulations of the developed hydrology model where the previously tuned parameters values are used (values from Table 5.2).

Figure 5.2 demonstrates that the hydrology model gives reasonable results with good fitting of the real measurements, with the R^2 error function being equal to 0.817. The developed hydropower model is included in *OpenHPL*. The library ensures connection of the hydrology model with the reservoir of the hydropower system and gives the opportunity to model and simulate either a separate hydrology model or the whole hydropower system starting from precipitations/catchment area up to electricity generation.

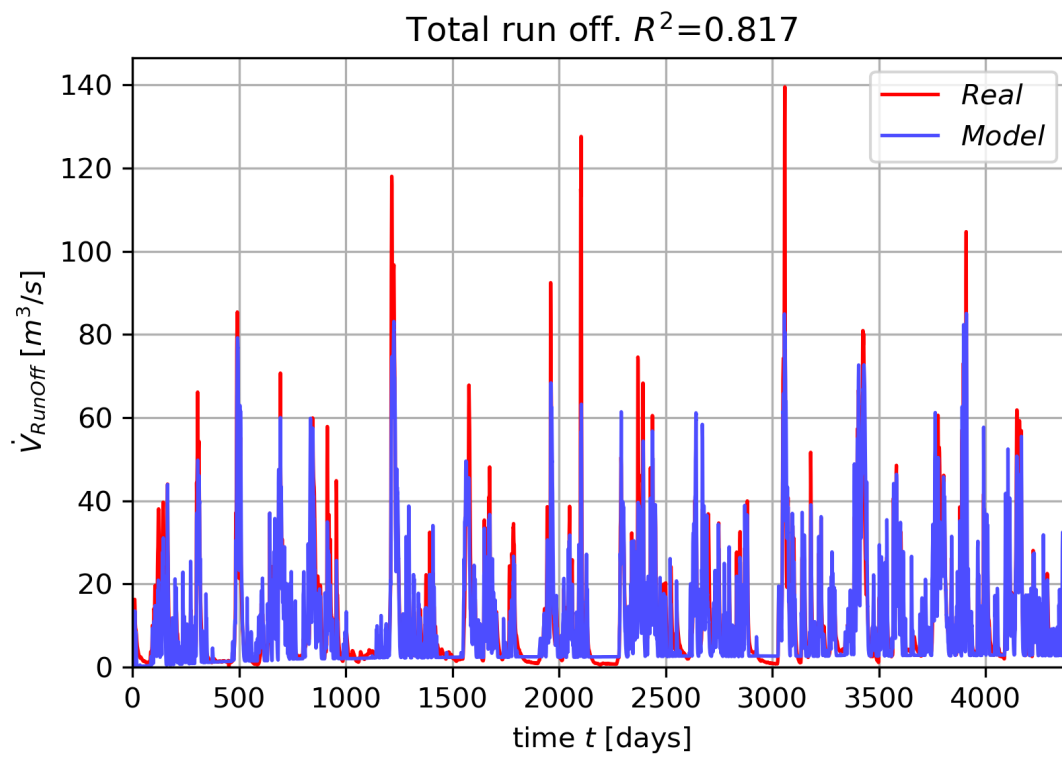


Figure 5.2: Simulation results of the hydrology model for the runoff.

6 Conclusions and future work

In this thesis, research has been conducted on the modelling and simulation of hydropower systems. In addition, the developed hydropower models have been analysed for control purposes. This chapter summarizes the contributions with respect to two main topics: hydropower modelling and model analysis. Recommendations for future work are also provided.

6.1 Hydropower library

The hydropower library *OpenHPL* has been developed in this work and consists of mathematical models of various hydropower system units:

- Models for the waterway units of the hydropower system have been developed and tested. The hyperbolic PDE solver based on the KP scheme is implemented in the library. *OpenHPL* provides possibilities for modelling various conduits with different complexity (including water compressibility, wall elasticity), e.g., the intake race, penstock and discharge race. Specific models for the surge tank and the reservoir have been developed for *OpenHPL*. Based on real experimental data, validation and application of all these waterway models have been carried out. In addition, a model for open channel flow is also included in the library, and enables the modelling of run-of-river hydropower systems, as well as reservoirs with more details.
- Mechanistic models for Francis and Pelton turbines based on the Euler equation are included in the library. A simplified turbine model based on a look-up table for the efficiency is also given and can be used for both mentioned turbine types. A Francis turbine design algorithm has been developed and tested. This design algorithm provides all needed parameters for the mechanistic Francis turbine model based on the nominal operating parameters. Validation and application of the Francis turbine model have been performed using real measurements as well as data from *Alab*.
- Connection to power system Modelica libraries is ensured, and gives a wide range of possibilities for modelling the hydro-mechanical part of the hydropower system together with the electrical part. A good option among various power system libraries is *OpenIPSL*, which contains many different models of synchronous generators,

6 Conclusions and future work

governors and other power grid components. In addition, simple generator and governor models have been encoded in *OpenHPL*.

- The HBV hydrology model for runoff estimation has also been developed in the library. Connection between this hydrology model and the reservoir is ensured. Calibration of the developed HBV model has been performed using real data from the Sundsbarm hydropower plant catchment. However, the tuned parameters are not general and must be re-calibrated for other catchments.

All these developed models in *OpenHPL* provide the capability to model the hydropower system starting from precipitations/reservoir to the final consumer of electricity. The library is an open-source product that can be run in both free OpenModelica and commercial Dymola Modelica based modelling and simulation environments, as well as other Modelica tools.

6.2 Model analysis tools

Various model analysis tools have been developed and tested in Python and Julia for the hydropower models in OpenModelica. The Python and Julia APIs for executing and controlling simulations in OpenModelica have been used. The following methods have been tested:

- Automatic linearization from the APIs has been tested for hydropower models of various complexity. The simulation showed that the linearization algorithm provides accurate linear approximation and returns correct linear time invariant matrices A , B , C and D . As an example, classical linear control and synthesis (Bode plot and PID controller design) has been applied and demonstrated for the linearized hydropower systems.
- Various nonlinear state estimation techniques have been studied and tested for the hydropower models. The implementation and use of nonlinear estimators such as Ensemble and Unscented Kalman filters have been demonstrated for estimating the states of the hydropower system based on water flow and/or pressure measurements. It has been shown that the combination of a mathematical model and available measurements provide good results and can be used for estimating states that are not directly measured in the hydropower system.
- Also, a deterministic reduced order nonlinear observer has been studied. The design and performance of this observer for estimating one of the states using the measurements from two other states have been shown. The results of the developed observer look promising, and the observed states can be used as an input to fast, nonlinear control algorithms.

- Structural analysis tools based on graph theory have been explored and developed for the hydropower models. It has been shown how these analysis tools can test structural observability and controllability as well as a relative degree of the system, based on the directed graph of the system structure. The results of testing the developed structural analysis functions look reasonable and can be used for analysis related to state estimation and control: observability is a requirement for state estimators to work properly, controllability is required for control design, and relative degree is important in the design of linear and nonlinear feedback controllers.

6.3 Future work

Based on the experience from the presented research work, a few topics are proposed for future studies:

- For cavitation studies, a description of two-phase liquid/vapour water can be included either for all the waterway models or only locally for low pressure units (turbine outlet/draft tube).
- Work on development of a Pelton turbine design algorithm to define all needed geometry parameters for the Pelton mechanistic model is a possibility for research. Moreover, testing and validation of this Pelton turbine mechanistic model with experimental data are of interest.
- Study of the reversible turbine-pump is also of interest. Such turbines can be used in pump power systems, i.e., pumping water up into reservoirs during periods of surplus intermittent energy.
- The developed library can also be extended with a mechanistic Kaplan turbine model, and experiments of Kaplan turbine/run-of-river system might be carried out.
- *OpenHPL* can also be extended with models for closed surge tanks. In addition, the effect of the position of the surge tank in the hydropower system might be of interest to study.
- Work on modelling the waterway of the hydropower system can be extended by taking into account the water temperature variation in the system.
- Study of the hydrology model can be extended to include the use of the Kalman filters, e.g., EnKF, in order to estimate the HBV model parameters.
- Work on developing model analysis tools for the hydropower models with studies on sensitivity or stability analysis as well as on design/implementation of advanced control, e.g., model predictive control.

Bibliography

- [1] B. Lie, ‘Lecture notes in course FM1015 Modelling of Dynamic Systems’, University of South-Eastern Norway, Porsgrunn, Norway, Tech. Rep., 2018.
- [2] C.-y. Xu, ‘Hydrologic Models’, Uppsala University, Tech. Rep., 2002.
- [3] S. Bergström and A. Forsman, ‘Development of a conceptual deterministic rainfall-runoff model’, *Hydrology Research*, vol. 4, no. 3, pp. 147–170, 1973. DOI: 10.2166/nh.1973.0012.
- [4] S. Bergström, *The HBV Model: Its Structure and Applications*, ser. SMHI Reports Hydrology. SMHI, 1992. [Online]. Available: <https://books.google.no/books?id=u7F7mwEACAAJ>.
- [5] G. Lindström, B. Johansson, M. Persson, M. Gardelin and S. Bergström, ‘Development and test of the distributed HBV-96 hydrological model’, *Journal of Hydrology*, vol. 201, no. 1-4, pp. 272–288, 1997. DOI: 10.1016/S0022-1694(97)00041-3.
- [6] M. Fjeld and S. Aam, ‘An Implementation of Estimation Techniques to a Hydrological Model for Prediction of Runoff to a Hydroelectric Power-Station’, *Modeling, Identification and Control: A Norwegian Research Bulletin*, vol. 2, no. 1, pp. 37–56, 1981. DOI: 10.4173/mic.1981.1.3.
- [7] S. Shafiee, ‘Automatic updating of hydrological models for runoff/inflow forecasting to hydropower system’, Master’s thesis, Telemark University College, Porsgrunn, Norway, 2013.
- [8] K. Berg and K. Nyström, ‘Hydrological modeling in Modelica’, in *Proceedings of the 4th International Modelica Conference*, 2005, pp. 149–154.
- [9] G. K. Devia, B. Ganasri and G. Dwarakish, ‘A Review on Hydrological Models’, *Aquatic Procedia*, vol. 4, pp. 1001–1007, 2015. DOI: 10.1016/j.aqpro.2015.02.126.
- [10] R. Sharma, ‘Second order scheme for open channel flow’, Porsgrunn: Telemark University College, Tech. Rep., 2015. [Online]. Available: <http://hdl.handle.net/11250/2438453>.
- [11] L. Vytvytskyi, R. Sharma and B. Lie, ‘Model based control for run-of-river system. Part 1: Model implementation and tuning’, *Modeling, Identification and Control: A Norwegian Research Bulletin*, vol. 36, no. 4, pp. 237–249, 2015. DOI: 10.4173/mic.2015.4.4.

Bibliography

- [12] G. Munoz-Hernandez, S. Mansoor and D. Jones, *Modelling and Controlling Hydropower Plants*, ser. Advances in Industrial Control. Springer London, 2013.
- [13] H. Fang, L. Chen, N. Dlakavu and Z. Shen, ‘Basic Modeling and Simulation Tool for Analysis of Hydraulic Transients in Hydroelectric Power Plants’, *IEEE Transactions on Energy Conversion*, vol. 23, no. 3, pp. 834–841, 2008. DOI: 10.1109/TEC.2008.921560.
- [14] B. R. Sharefi, ‘Modeling for Control of Hydropower Systems’, Master’s thesis, Telemark University College, Porsgrunn, Norway, 2011.
- [15] N. Kishor, R. Saini and S. Singh, ‘A review on hydropower plant models and control’, *Renewable and Sustainable Energy Reviews*, vol. 11, no. 5, pp. 776–796, 2007.
- [16] G. S. Stelling and S. P. A. Duinmeijer, ‘A staggered conservative scheme for every Froude number in rapidly varied shallow water flows’, *International Journal for Numerical Methods in Fluids*, vol. 43, no. 12, pp. 1329–1354, 2003. DOI: 10.1002/flid.537.
- [17] B. van’t Hof and A. E. Veldman, ‘Mass, momentum and energy conserving (MaMEC) discretizations on general grids for the compressible Euler and shallow water equations’, *Journal of Computational Physics*, vol. 231, no. 14, pp. 4723–4744, 2012. DOI: 10.1016/j.jcp.2012.03.005.
- [18] C. C. Warnick, *Hydropower engineering*. Englewood Cliffs, NJ : Prentice-Hall, 1984.
- [19] E. B. Wylie and V. L. Streeter, *Fluid Transients in Systems*. Englewood Cliffs, NJ : Prentice Hall, 1993.
- [20] E. Isaacson, *Analysis of Numerical Methods*. Dover Publications Inc., 7th Jun. 1994, 576 pp., ISBN: 0486680290. [Online]. Available: https://www.ebook.de/de/product/3303159/eugene_isaacson_analysis_of_numerical_methods.html.
- [21] A. Quarteroni and A. Valli., *Numerical Approximation of Partial Differential Equations*, 1st edition. Berlin: Springer, 2008, ISBN: 9783540852674.
- [22] A. Jinasena, G.-O. Kaasa and R. Sharma, ‘Use of Orthogonal Collocation Method for a Dynamic Model of the Flow in a Prismatic Open Channel: For Estimation Purposes’, in *Proceedings of the 58th Conference on Simulation and Modelling (SIMS 58)*, Linköping University Electronic Press, 2017, pp. 90–96. DOI: 10.3384/ecp1713890.
- [23] A. O. Johansen and E. B. J. N. Sørensen, ‘Implementation and test of a fifth order central WENO scheme for solving hyperbolic balance laws’, *Applied Thermal Engineering*, vol. 1–25, no. 1, 2011.
- [24] A. O. Johansen and B. Elmegaard, ‘A homogeneous two-phase flow model of an evaporator with internally rifled tubes’, *ScienceDirect*, vol. 1–21, 2012.
- [25] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002, 580 pp., ISBN: 0521009243.

- [26] A. Kurganov and G. Petrova, ‘A Second-Order Well-Balanced Positivity Preserving Central-Upwind Scheme for the Saint-Venant System’, *Communications in Mathematical Sciences*, vol. 5, no. 1, pp. 133–160, 2007. DOI: 10.4310/cms.2007.v5.n1.a6.
- [27] L. Saldamli, ‘PDEModelica – A High-Level Language for Modeling with Partial Differential Equations’, PhD thesis, Linköping University, PELAB - Programming Environment Laboratory, 2006, p. 163, ISBN: 91-85523-84-4. [Online]. Available: <http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-5732>.
- [28] J. Silar, F. Jezek and J. Kofránek, ‘PDEModelica and Breathing in an Avalanche’, in *Proceedings of the 12th International Modelica Conference, Prague, Czech Republic, May 15-17, 2017*, Linköping University Electronic Press, 2017. DOI: 10.3384/ecp17132367.
- [29] O. Quiroga, J. Riera and C. Batlle, ‘Identification of partially known models of the Susqueda hydroelectric power plant’, *Latin American applied research*, vol. 33, pp. 387–392, 2003.
- [30] P. Margonis, ‘Modeling and Optimization of a Hydroelectric Power Plant for a National Grid Power System Supply. Case Study: Stratos Hydroelectric Dam’, PhD thesis, Technical University of Crete, 2017.
- [31] O. Souza, N. Barbieri and A. Santos, ‘Study of hydraulic transients in hydropower plants through simulation of nonlinear model of penstock and hydraulic turbine model’, *IEEE Transactions on Power Systems*, vol. 14, no. 4, pp. 1269–1272, 1999. DOI: 10.1109/59.801883.
- [32] C. Nicolet, ‘Hydroacoustic modelling and numerical simulation of unsteady operation of hydroelectric systems’, PhD thesis, ISE, Lausanne, 2007. DOI: 10.5075/epfl-thesis-3751.
- [33] K. V. Vereide, ‘Hydraulics and Thermodynamics of Closed Surge Tanks for Hydropower Plants’, PhD thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2016.
- [34] L. Vytvytskyi, R. Sharma, I. Granström and B. Lie, ‘Modeling for Control of Run-of-River Power Plant Grnvollfoss’, in *Proceedings of the 56th Conference on Simulation and Modelling (SIMS 56)*, Linköping University Electronic Press, 2015. DOI: 10.3384/ecp15119237.
- [35] S. L. Dixon and C. Hall, *Fluid Mechanics and Thermodynamics of Turbomachinery*, Seventh. Butterworth-Heinemann, 2013.
- [36] T. K. Nielsen, ‘Transient characteristics of high-head Francis turbines’, PhD thesis, NTNU, 1990.
- [37] D. R. Giosio, A. D. Henderson, J. M. Walker and P. A. Brandner, ‘Physics-based hydraulic turbine model for system dynamic studies’, *IEEE Transactions on Power Systems*, vol. 32, no. 2, pp. 1161–1168, 2017. DOI: 10.1109/TPWRS.2016.2574330.

Bibliography

- [38] P. J. Gogstad, ‘Hydraulic design of Francis turbine exposed to sediment erosion’, Master’s thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2012.
- [39] M. Eltvik, ‘Sediment erosion in Francis turbines’, PhD thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2013.
- [40] H. Brekke, *Hydraulic Turbines. Design, Erection and Operation*. Trondheim, Norway: Norwegian University of Science and Technology, 2001.
- [41] E. D. Jaeger, N. Janssens, B. Malfliet and F. V. D. Meulebroeke, ‘Hydro turbine model for system dynamic studies’, *IEEE Transactions on Power Systems*, vol. 9, no. 4, pp. 1709–1715, 1994. DOI: 10.1109/59.331421.
- [42] A. Perrig, ‘Hydrodynamics of the free surface flow in Pelton turbine buckets’, PhD thesis, Swiss Federal Institute of Technology in Lausanne, 2007. DOI: 10.5075/epfl-thesis-3715.
- [43] H. 1. Brekke, *A general study on the design of vertical Pelton turbines*. [S.l.]: H. Brekke, 1984, 15 bl. ill.
- [44] P. Kundur, *Power System Stability and Control*. McGraw-Hill Education - Europe, 1994, ISBN: 007035958X.
- [45] T. Øyvang, ‘Enhanced power capability of generator units for increased operational security’, PhD thesis, University of South-Eastern Norway, 2018.
- [46] W. Yang, P. Norrlund, J. Bladh, J. Yang and U. Lundin, ‘Hydraulic damping mechanism of low frequency oscillations in power systems: Quantitative analysis using a nonlinear model of hydropower plants’, *Applied Energy*, vol. 212, pp. 1138–1152, 2018. DOI: <https://doi.org/10.1016/j.apenergy.2018.01.002>.
- [47] R. A. Naghizadeh, S. Jazebi and B. Vahidi, ‘Modeling Hydro Power Plants and Tuning Hydro Governors as an Educational Guideline’, in *Proceedings of International Review on Modelling and Simulations*, vol. 5, Praise Worthy Prize, 2012.
- [48] V. Savcenko and B. Haut, ‘Construction and analysis of an automatic multirate time domain simulation method for large power systems’, *Electric Power Systems Research*, vol. 107, pp. 28–35, 2014. DOI: 10.1016/j.epsr.2013.09.008.
- [49] P. W. Sauer, D. J. LaGesse, S. Ahmed-Zaid and M. A. Pai, ‘Reduced Order Modeling of Interconnected Multimachine Power Systems Using Time-Scale Decomposition’, *IEEE Transactions on Power Systems*, vol. 2, no. 2, pp. 310–319, 1987. DOI: 10.1109/tpwrs.1987.4335124.
- [50] R. Fitzgerald and V. L. Van Blaricum, ‘Water hammer and mass oscillation (WHAMO) 3.0 user’s manual’, US Army Corps of Engineers, Construction Engineering Research Laboratories, Engineer Research and Development Center (U.S.), Tech. Rep., 1998. [Online]. Available: <http://hdl.handle.net/11681/13746>.

- [51] S. U. Sandvåg, ‘Surge Tank Atlas for Hydropower Plants’, Master’s thesis, Norwegian University of Science, Technology, Department of Hydraulic and Environmental Engineering, 2016.
- [52] W. Yang, ‘Hydropower plants and power systems : Dynamic processes and control for stable and efficient operation’, PhD thesis, Uppsala University, Electricity, 2017, p. 140, ISBN: 978-91-554-9871-9.
- [53] D. Winkler, H. M. Thoresen, I. Andreassen, M. A. S. Perera and B. R. Sharefi, ‘Modelling and Optimisation of Deviation in Hydro Power Production’, in *Proceedings of the 8th International Modelica Conference*, vol. 1, Linköping Electronic Conference Proceedings, 2011.
- [54] K. Tuszynski, J. Tuszynski and K. Slättop, ‘HydroPlant – a Modelica Library for Dynamic Simulation of Hydro Power Plants’, in *Proceedings of the 5th International Modelica Conference*, vol. 1, 2006.
- [55] D. Winkler, ‘Electrical Power System Modelling in Modelica - Comparing Open-source Library Options’, in *Proceedings of the 58th Conference on Simulation and Modelling (SIMS 58)*, Linköping University Electronic Press, 2017. DOI: 10.3384/ecp17138263.
- [56] M. Baudette, M. Castro, T. Rabuzin, J. Lavenius, T. Bogodorova and L. Vanfretti, ‘OpenIPSL: Open-Instance Power System Library — Update 1.5 to “iTesla Power Systems Library (iPSL): A Modelica library for phasor time-domain simulations”’, *SoftwareX*, vol. 7, pp. 34–36, 2018. DOI: 10.1016/j.softx.2018.01.002.
- [57] C. Payne, ‘On the security of open source software’, *Information Systems Journal*, vol. 12, no. 1, pp. 61–78, 2002. DOI: 10.1046/j.1365-2575.2002.00118.x.
- [58] B. Lie, S. Bajracharya, A. Mengist, L. Buffoni, A. Kumar, M. Sjölund, A. Asghar, A. Pop and P. Fritzson, ‘API for Accessing OpenModelica Models From Python’, in *Proceedings of the EuroSim 2016, Oulu, Finland*, 2016.
- [59] B. Lie, A. Palanisamy, A. Mengist, L. Buffoni, M. Sjölund, A. Asghar, A. Pop and P. Fritzson, ‘OMJulia: An OpenModelica API for Julia-Modelica Interaction’, in *Proceedings of the 13th International Modelica Conference, Regensburg, Germany, March 4–6, 2019*, Linköping University Electronic Press, 2019. DOI: 10.3384/ecp19157699.
- [60] C. Rackauckas and Q. Nie, ‘DifferentialEquations.jl – A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia’, *Journal of Open Research Software*, vol. 5, no. 1, 2017. DOI: 10.5334/jors.151.
- [61] Q. Nie and C. Rackauckas, ‘Adaptive methods for stochastic differential equations via natural embeddings and rejection sampling with memory’, *Discrete and Continuous Dynamical Systems - Series B*, vol. 22, no. 7, pp. 2731–2761, 2017. DOI: 10.3934/dcdsb.2017133.

Bibliography

- [62] K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2010.
- [63] A. N. M. Panos J. Antsaklis, *Linear Systems*. Springer Basel AG, 2005, ISBN: 0817644342.
- [64] R. F. Stengel, *Optimal Control and Estimation*. Dover Publications Inc., 2003, ISBN: 0486682005.
- [65] J.-J. Slotine, *Applied Nonlinear Control*. Pearson Education, 1990, ISBN: 0130408905.
- [66] J. Revels, M. Lubin and T. Papamarkou, ‘Forward-Mode Automatic Differentiation in Julia’, *arXiv:1607.07892 [cs.MS]*, 2016. [Online]. Available: <https://arxiv.org/abs/1607.07892>.
- [67] D. Simon, *Optimal State Estimation: Kalman, H_∞ , and Nonlinear Approaches*. John Wiley & Sons, 2006.
- [68] G. Evensen, ‘The Ensemble Kalman Filter for Combined State and Parameter Estimation’, *IEEE Control Systems*, vol. 29, no. 3, pp. 83–104, 2009. DOI: 10.1109/MCS.2009.932223.
- [69] S. Julier and J. Uhlmann, ‘A new extension of the Kalman filter to nonlinear systems’, in *Proc. of AeroSense: The 11th Int. Symp. on Aerospace/Defense Sensing, Simulations and Controls*, vol. 3068, 1997, pp. 3068–3068. DOI: 10.1117/12.280797.
- [70] W. Zhou and B. Glemmestad, ‘Implementation of Unscented Kalman filter for non-linear state estimation in hydropower plant’, *IEEE International Conference on Power System Technology (POWERCON)*, pp. 1–5, 2012. DOI: 10.1109/PowerCon.2012.6401460.
- [71] L. Zou, C. Zhan, J. Xia, T. Wang and C. Gippel, ‘Implementation of evapotranspiration data assimilation with catchment scale distributed hydrological model via an ensemble Kalman Filter’, *Journal of Hydrology*, vol. 546, pp. 685–702, 2017. DOI: 10.1016/j.jhydrol.2017.04.036.
- [72] L. Shi, L. Zeng, D. Zhang and J. Yang, ‘Multiscale-finite-element-based ensemble Kalman filter for large-scale groundwater flow’, *Journal of Hydrology*, vol. 468-469, pp. 22–34, 2012. DOI: 10.1016/j.jhydrol.2012.08.003.
- [73] K. Murphy, *Machine Learning: A Probabilistic Perspective*, ser. Adaptive computation and machine learning. MIT Press, 2012, ISBN: 9780262018029.
- [74] J. A. Farrell and M. M. Polycarpou, *Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches (Adaptive and Learning Systems for Signal Processing, Communications and Control Series)*. New York, NY, USA: Wiley-Interscience, 2006, ISBN: 0471727881.

- [75] T. Stokelj and R. Golob, ‘Application of neural networks for hydro power plant water inflow forecasting’, in *Proceedings of the 5th Seminar on Neural Network Applications in Electrical Engineering. NEUREL 2000 (IEEE Cat. No.00EX287)*, 2000, pp. 189–193. DOI: 10.1109/NEUREL.2000.902410.
- [76] R. Sacchi, M. C. Ozturk, J. C. Principe, A. A. F. M. Carneiro and I. N. da Silva, ‘Water Inflow Forecasting using the Echo State Network: A Brazilian Case Study’, in *Proceedings of the International Joint Conference on Neural Networks*, 2007, pp. 2403–2408. DOI: 10.1109/IJCNN.2007.4371334.
- [77] D. G. Luenberger, ‘Observing the State of a Linear System’, *IEEE Transactions on Military Electronics*, vol. 8, pp. 74–80, 1964. DOI: 10.1109/TME.1964.4323124.
- [78] A. J. Krener and A. Isidori, ‘Linearization by output injection and nonlinear observers’, *Systems & Control Letters*, vol. 3, no. 1, pp. 47–52, 1983. DOI: 10.1016/0167-6911(83)90037-3.
- [79] C. Kravaris, V. Sotiropoulos, C. Georgiou, N. Kazantzis, M. Xiao and A. J. Krener, ‘Nonlinear observer design for state and disturbance estimation’, *Systems & Control Letters*, vol. 56, no. 11, pp. 730–735, 2007. DOI: 10.1016/j.sysconle.2007.05.001.
- [80] N. El-Farra, P. Mhaskar and P. D. Christofides, ‘Output feedback control of switched nonlinear systems using multiple Lyapunov functions’, *Proceedings of the 2005, American Control Conference, 2005.*, vol. 6, pp. 3792–3799, 2005.
- [81] V. Andrieu and L. Praly, ‘On the Existence of a Kazantzis–Kravaris/Luenberger Observer’, *SIAM J. Control and Optimization*, vol. 45, no. 2, pp. 432–456, 2006. DOI: 10.1137/040617066.
- [82] M. C. Smith, ‘The general problem of the stability of motion: Translated and Edited by A. T. Fuller. Taylor and Francis, 1992’, *Automatica*, vol. 31, pp. 353–354, 1995. DOI: 10.1016/0005-1098(95)90022-5.
- [83] D. D. Šiljak, *Decentralized Control of Complex Systems*, ser. Dover Books on Electrical Engineering Series. Dover Publications, 2011, ISBN: 9780486486147.
- [84] N. Deo, *Graph Theory with Applications to Engineering and Computer Science*. Dover Publications, 2017, ISBN: 9780486820811.
- [85] D. D. Šiljak, *Large-Scale Dynamic Systems: Stability and Structure*, ser. Dover Civil and Mechanical Engineering Series. Dover Publications, 2007, ISBN: 9780486462851.
- [86] J. Lunze, *Feedback control of large scale systems*, ser. Prentice-Hall international series in systems and control engineering. Prentice-Hall, 1992, ISBN: 9780133183535.
- [87] S. Boyd and L. Vandenberghe, *Introduction to Applied Linear Algebra*. Cambridge University Pr., 2018, 500 pp., ISBN: 1316518965.
- [88] J. Maciejowski, *Multivariable Feedback Design*, ser. Electronic systems engineering series. Addison-Wesley, 1989, ISBN: 9780201182439.

Bibliography

- [89] M. A. S. Perera, ‘State Estimation and Optimal Control of an Industrial Copper Electrowinning’, PhD thesis, University College of Southeast Norway, Faculty of Technology, Porsgrunn., 2016.
- [90] M. Tiller, *Modelica by Example*, Free online book., 2016. [Online]. Available: <http://book.xogeny.com>.
- [91] V. Splavska, L. Vytvytskyi and B. Lie, ‘Hydropower Systems: Comparison of Mechanistic and Table Look-up Turbine Models’, in *Proceedings of the 58th Conference on Simulation and Modelling*, Linköping University Electronic Press, 2017, pp. 368–373. DOI: 10.3384/ecp17138368.
- [92] L. Vytvytskyi, ‘User’s guide for the Open Hydropower library (OpenHPL)’, University of South-Eastern Norway, Tech. Rep., Aug. 2019. [Online]. Available: <http://hdl.handle.net/11250/2607645>.
- [93] T. Insperger and G. Stépán, ‘Semi-discretization method for delayed systems’, *International Journal for Numerical Methods in Engineering*, vol. 55, no. 5, pp. 503–518, 2002. DOI: 10.1002/nme.505.
- [94] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1979.

Part II
Scientific papers

Paper A

Comparison of elastic vs. inelastic penstock model using OpenModelica

Liubomyr Vytvytskyi, & Bernt Lie

Published in *Proceedings of the 58th Conference on Simulation and Modelling (SIMS58)*, 138, 20–28. Linköping University Electronic Press, 2017.

Comparison of elastic vs. inelastic penstock model using OpenModelica

Vytvytskyi Liubomyr Lie Bernt

Department of Electrical engineering, Information Technology and Cybernetics, University College of Southeast Norway, Porsgrunn, Norway, {Liubomyr.Vytvytskyi, Bernt.Lie}@usn.no

Abstract

The possibility for modelling and simulating hydropower systems as accurately as possible take an important role in order to develop a control structure and to make efficient analysis tools for testing a designed controller for stability and performance in different operating regimes. Both the simulation time for such models as well as the accuracy are important.

A high head hydropower system is considered for this study. The pipe with the main part of the height drop is known as pressure shaft or penstock, and it can be modeled with two levels of accuracy which have been compared in this studying. A simple model with one nonlinear ODE considers inelastic walls of the penstock and incompressible water. A more realistic model for large pressure variations assumes a penstock with elastic walls and compressible water column in the penstock. This more detailed model of a penstock is described with two nonlinear PDEs which have been solved using the Kurganov-Petrova scheme.

Comparing results from these two models it can be concluded that the simple ODE model shows by and large the same results as the PDE model with just slightly smoothed dynamics. Obviously, the simulation time for the inelastic penstock model is considerably smaller. Both models show reasonable results and can be further used for control synthesis and analysis. In cases where the time consumption is most important, the simple ODE model for the penstock is preferred. On the other hand, for more accurate studies the elastic/compressible model for the penstock or even for other waterway units, such as conduit, is more useful.

The modeling part for both cases was done in OpenModelica using our own hydropower library, where all models for different units of the hydropower system have been developed and collected.

Keywords: high head hydropower, penstock/pressure shaft, Kurganov-Petrova scheme, OpenModelica

1 Introduction

1.1 Background

A transition towards more renewable energy sources is currently happening in Europe and all over the world, with increasing use of flexible hydropower plants to compensate for the highly changing production from intermittent energy sources such as wind and solar irradiation. Flexible hydropower plants involves pumping water up into reservoirs during surplus of intermittent power, and high head storage is the most efficient way to store such surplus power.

Around 96% of generated electricity (138 GWh) in Norway is produced by hydropower systems, which have a total capacity over 30 GW; over 1500 hydropower plants are operated¹. With these values, Norway occupies the 7th place among the hydropower producers in the world². Among the power plants, the high head hydropower plants generate more than 75% of all electricity produced from hydropower in Norway³.

In addition to hydropower plants, the number of wind power plants in Norway increase from year to year (13.4% increase in 2015 with respect to 2014⁴) due to this technology becoming cheaper and more mature. With long coast line, wind power has a huge potential for producing power in Norway⁵. On the other hand, this renewable energy source is intermittent and create considerable disturbances in the power grid. From this perspective, hydropower can be used to compensate for disturbances from the wind power.

To optimize the combination of intermittent power and stored power, the possibilities for modelling and simulating the hydropower system as accurately as possible take an important role in order to make an efficient analysis tool for testing a designed controller for stability and performance in different operating regimes (Sharefi, 2011).

¹ <http://lvk.no/LVK/Fagomrader/Vannkraftproduksjon/Nokkeltall---Oversikt-over-konsesjonssystemet-for>

² <https://snl.no/vannkraft>

³ https://no.wikipedia.org/wiki/Liste_over_vannkraftverk_i_Norge

⁴ <http://www.ssb.no/en/energi-og-industri/statistikker/elektrisitet/aar>

⁵ <http://www.vindportalen.no/Vindportalen/Vindkraft/Vindkraft-i-Norge>

1.2 Previous work

High head plants typically collect and store water in reservoirs in the mountains, with tunnels leading the relatively small flow of water down a considerable height difference to the aggregated turbine and generator. A dynamic model for such a hydropower system has been developed and studied in a previous work (Sharefi, 2011), where the Staggered grid scheme was used for discretization of a part of the model with partial differential equations (PDEs). Another scheme for discretization is the Kurganov-Petrova (KP) scheme, described in (Sharma, 2015; Kurganov & Petrova, 2007). The comparison of this KP scheme with Staggered grid for an open channel model using Matlab was done in (Vytvytskyi, et al., 2015). In (Saldamli, 2006) a Modelica extension for modeling with partial differential equations, PDEModelica, was proposed. An updated version of PDEModelica with implementation for OpenModelica was recently presented in (Silar, et al., 2017); this extension is not fully completed, though. In addition, a commercial Modelica library for hydropower system exists (Hydro Power Library from Modelon AB⁶), and has been used for modeling a high head hydropower plant in (Winkler, et al., 2011).

1.3 Overview of paper

In this work, a simple model of a hydropower system described with ordinary differential equations (ODEs) is compared with a more realistic model described with PDEs in order to find what accuracy level is needed in control synthesis. The current work uses the semi-discrete KP scheme implemented in Modelica for

solving the hyperbolic PDEs, which is a novelty compared to the commercial Hydro Power Library.

The paper is structured as follows: Section 2 gives a system description of the high head hydropower system. Section 3 consists of formulation and discretization of the model. Then simulation with validation and comparison of the different model complexities are described in Section 4. Finally, discussion and conclusions are given in Section 5.

2 System description

A typical structure for the high head hydropower plant is depicted in Figure 1⁷. Here, the water is transported from a reservoir, where it is stored, through a tunnel known as intake race / conduit / headrace closer to the powerhouse where the turbine and generator is installed. The conduit can have considerable length, normally with a small slope.

After the conduit, the water flows down with a steep slope and a relatively small flow rate to the turbine through a pipe known as the penstock or pressure shaft. There is large pressure change in this pipe due to the big height difference. At the point where the penstock is connected to the intake race, a surge tank may be installed to reduce the water hammer pressure variation and keep the mass oscillation within limits. This surge tank can be constructed as a pipe and is then denoted a surge shaft, but can also be variations such as an open or closed reservoir.

After the turbine, the water can flow directly to a downstream lake or river, often denoted tail water, or the

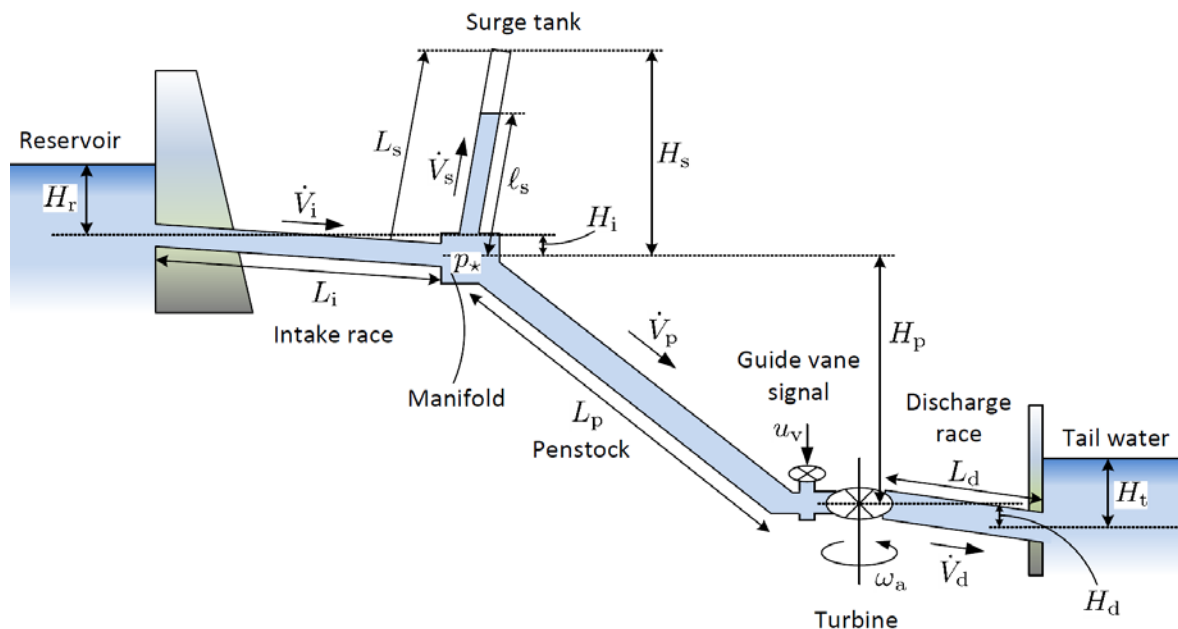


Figure 1. Structure of the high head hydropower plant.

⁶ <http://www.modelon.com/products/modelica-libraries/hydro-power-library>

⁷ Agu, C., E., Vytvytskyi, L., Lie, B. (2016) Project, FM1015 Modelling of Dynamic Systems. University College of Southeast Norway, Porsgrunn, Norway.

water can flow through a discharge race if the powerhouse is situated some distance from the tail water. Normally the turbine is installed below the level of the tail water in order to increase the pressure at the outlet of the turbine to reduce the possibility of cavitation.

Two types of turbines are common for high head hydropower systems. When the height difference is large (300 – 4000 m), a Pelton turbine is used with relatively small flow rate (<30 m³/s). Alternatively, with smaller height differences and large flow rates, Francis turbines are preferred. In our simulations, a Francis turbine is used, and the control of the flow rate through the turbine is done by manipulating the guide vane.

The turbine is rotating an axis which also holds the rotor of the generator producing electricity to the power grid. Normally, a synchronous generator is used.

2.1 Geometry

For simulations in this paper, the data from the Sundsbarm hydropower plant in Telemark, Norway is used with data provided in (Winkler, et al., 2011), see Table 1 and Table 2.

Table 1. The waterway geometry of Sundsbarm hydropower plant.

Waterway unit	Height difference, m	Length, m	Diameter, m
Reservoir	48	–	–
Conduit	23	6600	5.8
Penstock	428.5	600	3
Surge tank	120	140	3.4
Discharge race	0.5	600	5.8
Tail water	5	–	–

Table 2. The turbine geometry of Sundsbarm hydropower plant.

Turbine type	Nominal head, m	Nominal flow rate, m ³ /s	Nominal power, MW
Francis	460	24.3	104.4

3 Modeling and discretization

3.1 Model presentation

Models for all of the waterway units can be described with mass and momentum balances. For the mechanical part (turbine with aggregate), a simplified energy balance is used assuming that the turbine behaves as a simple valve. All of these models for different units of

⁸ Modelica® is a non-proprietary, object-oriented, equation based language to conveniently model complex physical systems. <https://www.modelica.org>

⁹ OpenModelica is an open-source Modelica-based modeling and simulation environment intended for

the hydropower system have been developed and collected for our own hydropower Modelica⁸ library that can be used in either OpenModelica⁹ or Dymola¹⁰.

In this paper, a more detailed description is presented for the models that will be compared, namely a simplified pipe (tunnel) model that can be used for the penstock and a more realistic model with compressible water and elastic walls of the penstock.

3.1.1 Waterway pipe (tunnel)

In some of the waterway units such as conduit, there are only small pressure variations due to the small slope angle (height difference between inlet and outlet of the component). That is why the model for these units can be simplified by considering incompressibility of the water and inelasticity of the walls. A sketch of the pipe with all needed terms for modeling is shown in Figure 2.

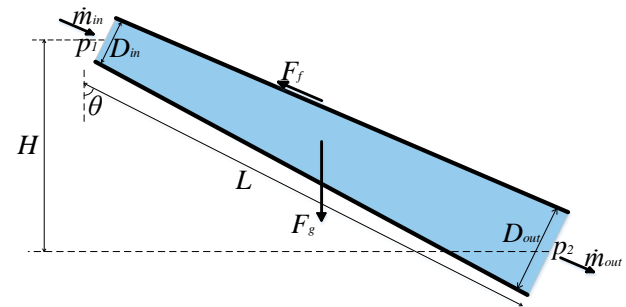


Figure 2. Model for flow through a pipe.

In the case of incompressible water, mass in the filled pipe is constant, and:

$$\frac{dm}{dt} = \dot{m}_{in} - \dot{m}_{out} = 0 \quad (1)$$

Here, the mass of the water in the pipe is $m = \rho V = \rho L \bar{A}$, where ρ is the water density, V – the volume of the water in the pipe, L – the length of the pipe and \bar{A} – the averaged cross section area of the pipe that defined from averaged pipe diameter \bar{D} . The inlet and outlet mass flow rates are equal with $\dot{m}_{in} = \rho \dot{V}_{in}$ and $\dot{m}_{out} = \rho \dot{V}_{out}$ respectively, where $\dot{V}_{in} = \dot{V}_{out}$ – the inlet and outlet volumetric flow rates in the pipe.

The momentum balance for this simplified model can be expressed as:

$$\frac{dM}{dt} = \dot{M}_{in} - \dot{M}_{out} + F_p + F_g + F_f \quad (2)$$

Here, the momentum of the water in the pipe is $M = mv$, where v is the average water velocity and can be define as $v = \dot{V} / \bar{A}$. The inlet and outlet momentum flow

industrial and academic usage.

<https://www.openmodelica.org>

¹⁰ Dymola is a commercial complete tool for modeling and simulating integrated and complex systems; is based on the Modelica open standard language.

<http://www.modelon.com/products/dymola>

rates are $\dot{M}_{in} = \dot{m}_{in}v_{in}$ and $\dot{M}_{out} = \dot{m}_{out}v_{out}$ respectively, where $v_{in} = \dot{V}_{in}/A_{in}$ and $v_{out} = \dot{V}_{out}/A_{out}$ are the velocities in the inlet and outlet of the pipe respectively and are equal in a case with constant diameter of the pipe ($A_{in} = A_{out}$). F_p – the pressure force, due to the difference between the inlet and outlet pressures p_1 and p_2 and can be calculated as follows: $F_p = A_{in}p_1 - A_{out}p_2$. There is also gravity force that is defined as $F_g = mg \cos \theta$, where g – the gravitational acceleration and θ – angle of the pipe slope that can be defined from ratio of height difference and length of the pipe. The last term in the momentum balance is friction force, which can be calculated as $F_f = -\frac{1}{8}L f_D \pi \rho \bar{D} |v|$ using Darcy friction factor f_D .

3.1.2 Penstock with elastic walls and compressible water

Unlike the conduit, the penstock has considerable pressure variation due to a considerable height drop. Thus, to make model for the penstock more realistic, compressible water and elastic walls of the penstock should be taken into account. To express the compressibility/elasticity, some compressibility coefficients which show the relationship between pressure, water density and pipe inner radius, are used.

According to (Sharefi, 2011), isothermal compressibility β_T is defined as follows:

$$\beta_T = \frac{1}{\rho} \frac{d\rho}{dp} \quad (3)$$

Here ρ and p denote density and pressure, respectively. Assuming that the isothermal compressibility is independent of the pressure, this equation can be rewritten in a way that is convenient to calculate fluid density at different pressures:

$$\rho = \rho^{atm} e^{\beta_T(p - p^{atm})} \quad (4)$$

Here p^{atm} is the atmospheric pressure and ρ^{atm} is the water density at atmospheric pressure. The relation between density and pressure from this equation can be seen in Figure 3.

Figure 3 shows fairly linear dependency of the density with respect to the pressure in the range that is normal in hydropower plants. That is why the previous equation (4) can be simplified as follows:

$$\rho \approx \rho^{atm} (1 + \beta_T(p - p^{atm})) \quad (5)$$

In the same way, the relation between pressure and pipe cross section area can be defined using equivalent compressibility coefficient β^{eq} due to the pipe shell elasticity (Sharefi, 2011); after simplification the relation looks as follows:

$$A \approx A^{atm} (1 + \beta^{eq}(p - p^{atm})) \quad (6)$$

Here A^{atm} is the pipe cross section area at atmospheric pressure.

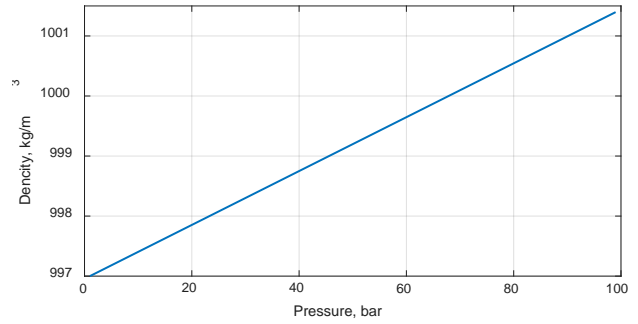


Figure 3. Variation of density with respect to pressure.

In reference to (Sharefi, 2011), it is also possible to define a linear relationship for the product of density and cross sectional area that change with pressure.

$$A \cdot \rho = A^{atm} \rho^{atm} (1 + \beta^{tot}(p - p^{atm})) \quad (7)$$

Here β^{tot} is the total compressibility due to water compressibility and pipe shell elasticity ($\beta^{tot} = \beta^{eq} + \beta_T$), and is related to the speed of sound in water inside the pipe.

Hence, using the previous equations for the relationship between density of the water, cross sectional area of the pipe, and pressure in the pipe, ODEs (1) and (2) for mass and momentum balances can be further developed into the PDEs:

$$\begin{aligned} A^{atm} \rho^{atm} \beta^{tot} \frac{\partial p}{\partial t} &= -\frac{\partial \dot{m}}{\partial x} \\ \frac{\partial \dot{m}}{\partial t} &= -\frac{\partial}{\partial x} (\dot{m}v + Ap) + \rho Ag \cos \theta \\ &\quad - \frac{1}{8} f_D \pi \rho D v |v| \end{aligned} \quad (8)$$

3.2 Model discretization

There are many discretization techniques for PDEs, but from previous work (Vytvytskyi, et al., 2015) it was observed that the well-balanced second order Kurganov-Petrova (KP) scheme shows reasonably good results for hyperbolic PDEs. The KP scheme is therefore also chosen for discretization of the model for the elastic penstock with compressible water. The detailed development of the KP scheme is shown in (Kurganov & Petrova, 2007) with some run-of-river case studies in (Sharma, 2015; Vytvytskyi, et al., 2015; Dissanayake, et al., 2016; Dissanayake, et al., 2017).

Firstly, PDEs (8) for the elastic penstock model should be presented in vector form as a standard formulation for KP scheme (Sharma, 2015):

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} = S \quad (9)$$

with: $U = [p, \dot{m}]^T$ – vector of conserved variables,
 $F = \left[\frac{\dot{m}}{A^{atm} \rho^{atm} \beta^{tot}}, \dot{m}v + Ap \right]^T$ – vector of fluxes,

$S = \left[0, \rho Ag \cos \theta - \frac{1}{8} f_D \pi \rho D v |v| \right]^T$ – source terms vector.

The result of discretizing the elastic penstock model using the KP scheme is the semi-discrete (time dependent ODEs) central-upwind scheme and can be written in the following from:

$$\frac{d}{dt} \bar{U}_i(t) = -\frac{H_{i+\frac{1}{2}}(t) - H_{i-\frac{1}{2}}(t)}{\Delta x} + \bar{S}_i(t) \quad (10)$$

Here, \bar{U}_i is the cell center averaged values while $H_{i\pm\frac{1}{2}}$ – the central upwind numerical fluxes at the cell interfaces, are defined as:

$$H_{i+\frac{1}{2}}(t) = \frac{a_{i+\frac{1}{2}}^+ F(U_{i+\frac{1}{2}}^-) - a_{i+\frac{1}{2}}^- F(U_{i+\frac{1}{2}}^+)}{a_{i+\frac{1}{2}}^+ - a_{i+\frac{1}{2}}^-} + \frac{a_{i+\frac{1}{2}}^+ a_{i+\frac{1}{2}}^-}{a_{i+\frac{1}{2}}^+ - a_{i+\frac{1}{2}}^-} [U_{i+\frac{1}{2}}^+ - U_{i+\frac{1}{2}}^-] \quad (11)$$

$$H_{i-\frac{1}{2}}(t) = \frac{a_{i-\frac{1}{2}}^+ F(U_{i-\frac{1}{2}}^-) - a_{i-\frac{1}{2}}^- F(U_{i-\frac{1}{2}}^+)}{a_{i-\frac{1}{2}}^+ - a_{i-\frac{1}{2}}^-} + \frac{a_{i-\frac{1}{2}}^+ a_{i-\frac{1}{2}}^-}{a_{i-\frac{1}{2}}^+ - a_{i-\frac{1}{2}}^-} [U_{i-\frac{1}{2}}^+ - U_{i-\frac{1}{2}}^-]$$

Here $a_{i\pm\frac{1}{2}}^\pm$ are the one-side local speeds of propagation, and can be defined as the smallest and the largest eigenvalues of the Jacobian $\frac{\partial F}{\partial U}$ of the system. These eigenvalues become:

$$\lambda_{1,2} = \frac{v \pm \sqrt{v^2 + \frac{4A}{A^{\text{atm}} \rho^{\text{atm}} \beta^{\text{tot}}}}}{2} \quad (12)$$

From these eigenvalues, it can be deduced that the speed of sound is given as $c = \sqrt{\frac{A}{A^{\text{atm}} \rho^{\text{atm}} \beta^{\text{tot}}}}$, thus confirming that the total compressibility factor β^{tot} is related to the speed of sound.

3.3 Operational data and parameters

The complete set of models for different units of the hydropower system now consists of a number of ODEs that can be simulated in OpenModelica or Dymola with the *der* operator for d/dt using the *dasdls* solver algorithm with simulation interval time equal to 0.4 s and tolerance 1e-4. All operational parameters for simulation are given in Table 3.

4 Simulation

Firstly, basic simulations of the system with various penstock models (inelastic and elastic penstock) are done to validate the models from our own library with a commercial one. Then, a more detailed comparison of models with different penstock complexities is given for various simulation scenarios.

Table 3. Parameters for simulating the high head hydropower system.

Variable	Value	Unit	Comments
g	9.81	m/s ²	Acceleration due to gravity
ρ (ρ^{atm})	997	kg/m ³	Density of water
μ	0.89e-3	Pa·s	Dynamic viscosity of water
ϵ	1.5e-5	m	Pipe roughness height
β_T	4.5e-10	Pa ⁻¹	Water compressibility
β^{tot}	1.003e-9	Pa ⁻¹	Total compressibility
p^{atm}	1.013e5	Pa	Atmospheric pressure
Δx	60	m	Cell length for penstock discretization
C_V	3.7	m ³ /s	Turbine valve capacity
η_h	0.9	–	Turbine hydraulic efficiency
η_e	0.99	–	Electricity generator efficiency
J_a	2e5	kg·m ²	Moment of inertia of the aggregate
$k_{f,b}$	1e3	$\frac{\text{W}\cdot\text{s}^2}{\text{rad}^2}$	Friction factor in the aggregate bearing

4.1 Validation

It is of interest to validate the developed models with the commercial Hydro Power Library in order to check that our own library shows reasonable results; some basic simulation is done for this validation. The scenario for this simulation is a simple turbine guide vane (valve) opening after time 30 s from 10% over a period of 30 s to 100%. The block diagrams for the hydropower systems using the Hydro Power Library and our own developed models are shown in Figure 4.

In case of both libraries, the turbine is presented as a simple valve. That is why the turbine valve capacity together with the pipe roughness height should be set to similar or equivalent values. After some tuning of these variables for our in-house models, the results of the simulation fits the results from the Modelon Hydro Power Library reasonably well and are shown in Figure 5: the pressure drop variations through the turbine are presented for different cases.

From Figure 5, it is seen that the system with simple, inelastic penstock shows smother dynamics after the beginning of disturbance (at time after 30 sec.), while some small oscillation take part in the results for the in-house elastic penstock model and using the Modelon Hydro Power Library. It should also be noted that the simulation time for the system with inelastic penstock is around one third of the two other models.

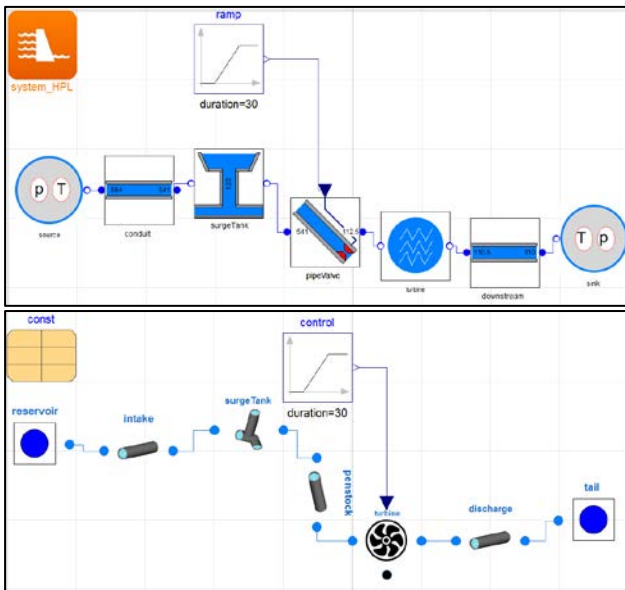


Figure 4. Block diagrams for hydropower system using Hydro Power Library (upper) and in-house library (lower).

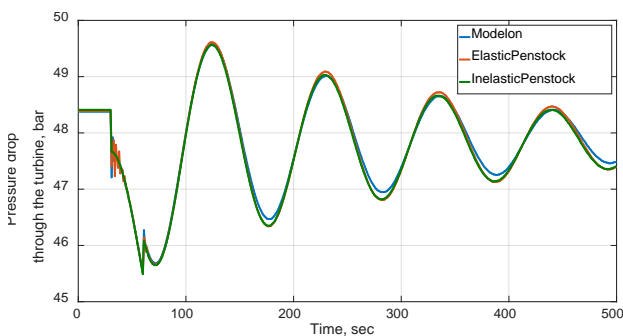


Figure 5. Validation of simulation results from own developed models (elastic and inelastic penstock) with Hydro Power Library.

4.2 Comparison

4.2.1 First simulation scenario

With the rudimentary validation in the previous section, a more detailed comparison of models with different penstock complexities can be studied. Firstly, the simulation scenario with sudden and small closing of the turbine guide vane (valve) is considered. Here, the systems are being simulated for 2000 s and the disturbance is applied at 600 s, when the valve is closed by 5% within 1 s. The results of simulation of the inelastic and the compressible/elastic models are presented in Figure 6, where plots are zoomed and shows just the time interval around disturbance (550 – 750 sec.)¹¹. In this figure the comparison of the pressure drop through the turbine (upper plot) and volumetric flow rate on the turbine (lower plot) are given for both

¹¹ The disturbance is not applied earlier in order to reach the steady state for the systems.

cases: systems with elastic and inelastic penstock models.

We observe smoother dynamics after disturbance for the inelastic penstock for both pressure drop and volumetric flow rate in Figure 6, which is similar to the validation case. From a power production perspective, this difference looks insignificant and may be neglected for control purposes if we take into account that the simulation time for a system with elastic penstock model is 3 times longer. On the other hand, it may be extremely important to observe these pressure oscillations to avoid cavitation and wear and tear of the turbine. That is why the outlet turbine pressure is also presented (see Figure 7) in order to compare models with elastic and inelastic penstock from the perspective of the cavitation problem.

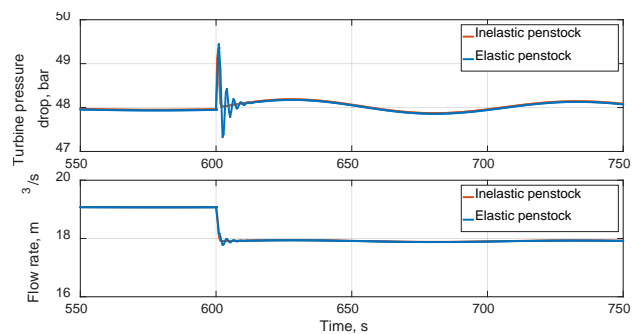


Figure 6. Comparison of simulation results for the systems with elastic vs. inelastic penstock models.

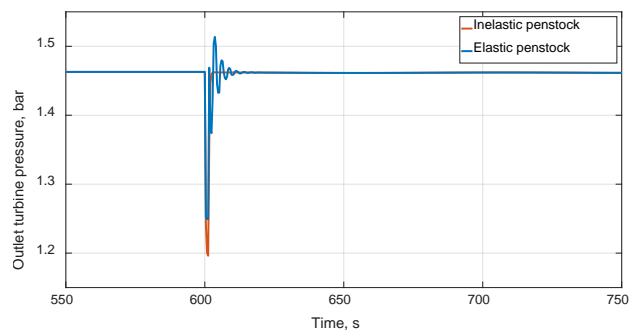


Figure 7. Comparison of the outlet turbine pressure for the systems with elastic vs. inelastic penstock models.

Figure 7 shows the same smoother dynamics with the simple inelastic penstock model, but the amplitude of the first oscillation is slightly higher for the model with elastic penstock.

4.2.2 Second simulation scenario

It is of interest to check another simulation scenario to see the effect of the penstock model complexity: a system without surge tank, but with the same disturbance as in the previous case (5% turbine valve closing at time 600 s). The results of simulating this scenario is shown in Figure 8, where only the

comparison of the pressure drop through the turbine is presented; the volumetric flow rate through the turbine shows similar results as in the previous simulation.

From Figure 8, it is seen that amplitudes of the first oscillation are different: in the system with inelastic penstock model the pressure drop rise higher after the disturbance than in the system with elastic penstock model. This difference can be related to the speed of increasing the pressure drop after disturbance: the pressure drop rises faster in the case with inelastic penstock model.

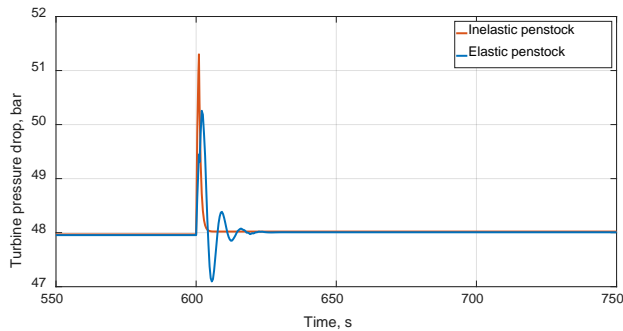


Figure 8. Comparison of the simulation results for the systems without surge tank.

It is also of interest, for these two penstock models, to see the difference at the outlet pressure from the turbine, which takes an important role in cavitation studies: due to the difference between the pressure drops through the turbine, here a point is to check from which side, inlet/outlet, of the turbine this difference was caused. Thus, this outlet turbine pressure is compared in Figure 9, where the simulation scenario is still the same (system without surge tank and 5% closing of the guide vane at time 600 s). Here, it is observed that the dynamics slightly differ between the system with elastic and inelastic penstock models, but the amplitude of the oscillation just after the disturbance is similar. This means that the inlet pressure to the turbine causes the difference in the turbine pressure drop.

The wave propagation is taken into account in the system with elastic penstock model and that is why models take some longer time for the turbine pressure drop to reach the maximum amplitude of the first oscillation after the disturbance, while in the system with inelastic penstock model all variables change simultaneously through the whole system. This can be observed in Figure 10, where the volumetric flow rate through the conduit is compared for cases with elastic vs. inelastic penstock models. Here, the flow rate through the conduit for the system with inelastic penstock starts to decrease at the time (600 s) when the disturbance occurs, while around half a second is needed for the wave in the system with elastic penstock model to reach the conduit.

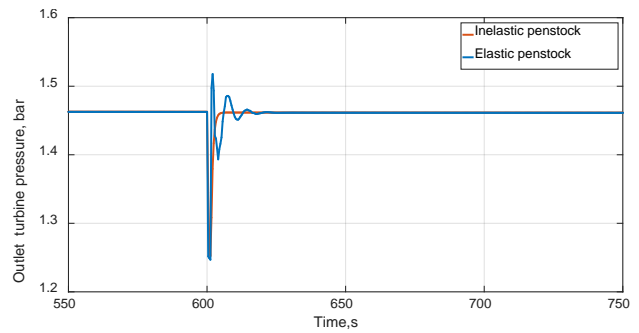


Figure 9. Comparison of the outlet turbine pressure for the systems without surge tank.

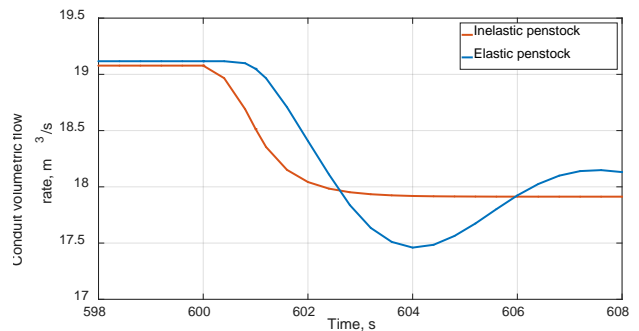


Figure 10. Comparison of the wave propagation in the systems with elastic vs. inelastic penstock models.

4.2.3 Third simulation scenario

In the last scenario, the effect of the wave propagation in the longest part of the hydropower system is studied. For this case, the elastic penstock model with compressible water is used for the conduit and the penstock units; the disturbance is the same as it was in the previous two scenarios. The simulations are done for both systems with (see Figure 11) and without (see Figure 12) surge tank and the turbine pressure drop is compared with the results for the previous two scenarios. It should be noted that the elastic model for the conduit is discretize with the same step length as for the penstock (60 m) and it leads to dramatic increasing of the simulation time, around 20-30 times.

From Figure 11, it is seen that the results for all three cases look almost the same. The system with the elastic conduit shows slightly more oscillatory results, but in general, there are no difference in overall dynamics. On the other hand, the dynamics differ substantially for the system without surge tank; the case with elastic conduit give a rather different behavior, which can be seen in Figure 12. Here, the oscillations after the disturbance need more time to reach a new steady state (compared to two other cases) and the amplitude of the first oscillation for the elastic conduit case is lower than for systems with elastic or inelastic penstock models.

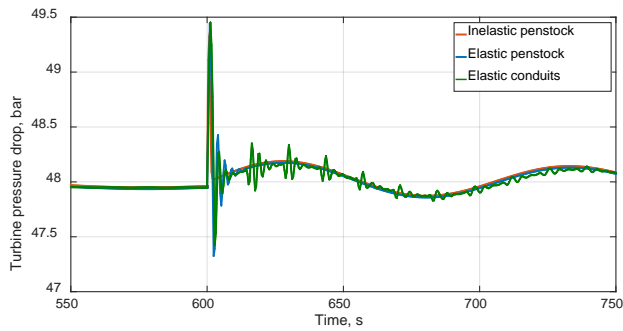


Figure 11. Comparison of the elastic conduit for the system with surge tank.

The longer oscillation time for the system with elastic penstock and conduit models is caused by the water wave moving back and forth (up and down) through the whole system. This is actually one of the reasons of using the surge tank: to keep mass oscillations within the limits: the oscillations in the system with surge tanks are much smoother and their amplitude variation is smaller.

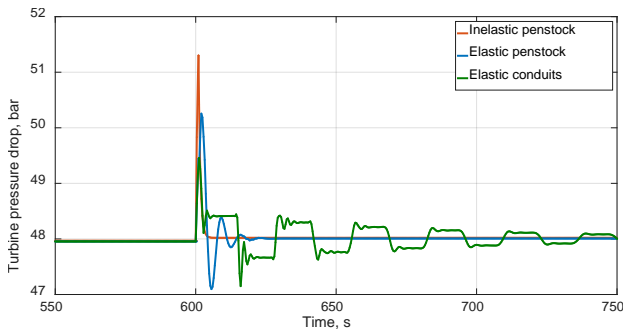


Figure 12. Comparison of the elastic conduit for the system without surge tank.

It is also of interest to compare the behavior of the outlet turbine pressure for the elastic model of the conduit. Thus, the comparisons have been done for both systems with and without the surge tank and results are shown in Figure 13 and Figure 14. Both figures show that the model with both the elastic conduit and penstock behaves more oscillatory, and at the same time, reaches the maximum amplitude values similar to the model with just the elastic penstock.

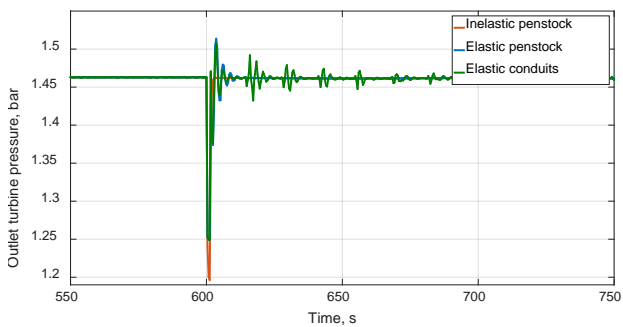


Figure 13. Comparison of the outlet turbine pressure for the systems with surge tank.

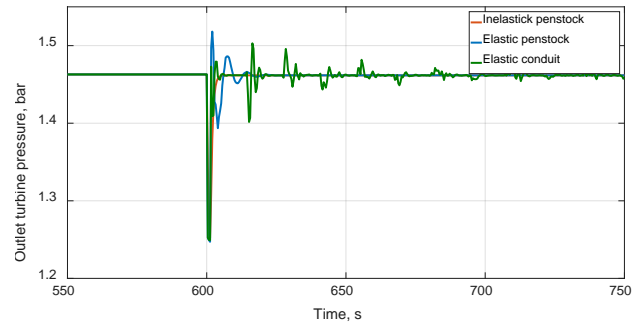


Figure 14. Comparison of the outlet turbine pressure for the systems without surge tank.

5 Discussion and conclusion

In this paper, the comparison of different penstock models for modeling and simulation of a high head hydropower system has been explored. The Kurganov-Petrova second order central upwind scheme has been implemented to discretize the more complicated penstock model with compressible water and elastic walls that is described by PDEs. This scheme has been implemented in Modelica and collected into an in-house library together with all other models for the hydropower elements. Then the models for simulating the hydropower system with different scenarios have been developed and simulated in OpenModelica. This in-house hydropower library has been validated with an already existing commercial Hydro Power Library.

Based on three simulation scenarios, it can be concluded that the system with inelastic penstock model exhibits a somewhat simpler dynamics, but on the other hand it has an advantage in the simulation time which is only one third of the same system but just with elastic penstock model (discretized by KP scheme). It is also observed that the wave propagation, which is taken into account in the elastic penstock model, affects the system, in particular the one without a surge tank. This effect leads to longer oscillations but also a smaller amplitude just after the disturbance (system without the surge tank). Despite all these differences, the inelastic penstock model can perhaps be used for control design problems due to the simplicity of this model, being less time consuming for simulation and reasonably good accuracy. Clearly, a model based controller based on a simple model should be tested on a more rigorous model, and, if necessary, re-tuned in order to ensure good performance. For a more detailed and accurate design, the elastic penstock model could be used in the controller, due to better representation of the system dynamics, which is caused by including the speed of the wave propagation to the model. Even better dynamic representation of the system can be reached by using the elastic model for the conduit. Then the delay caused by the speed of the wave propagation leads to more dramatic changes in the system dynamics (well observed in Figure 12).

Finally, our limited experience indicates that the simpler model appears to cover the worst case due to higher amplitude of the oscillation just after disturbance, even though it does not show proper dynamics. The systems with elastic and inelastic models do not behave in the same way for the outlet turbine pressure, the amplitude of the first oscillation after disturbance can differ, and this can be extremely important for a cavitation study.

References

- Dissanayake, S., Sharma, R. & Lie, B., 2016. *Semi Discrete Scheme for the Solution of Flow in River Tinnelva.* Oulu, Finland, EUROSIM 2016. IEEE, pp. 134-139.
- Dissanayake, S., Sharma, R. & Lie, B., 2017. Third Order Reconstruction of the KP Scheme for Model of River Tinnelva.. *Modeling, Identification and Control*, 38(1), p. 33–50.
- Kurganov, A. & Petrova, G., 2007. A second order well-balanced positivity preserving central-upwind scheme for the saint-venant system. *Communications in Mathematical Science*, Volume 5(1), p. 133–160..
- Saldamli, L., 2006. *PDEModelica A High-Level Language for Modeling with Partial Differential Equations*, Linköping, Sweden: Department of Computer and Information Science Linköping University.
- Sharefi, B. R., 2011. *Modeling for Control of Hydropower Systems*, Porsgrunn, Norway.: Master's Thesis. Telemark University College .
- Sharma, R., 2015. *Second order scheme for open channel flow.*, Porsgrunn, Norway: Technical report, Telemark Open Research Archive (TEORA), Telemark University College.
- Silar, J., Jezek, F. & Kofranek, J., 2017. *PDEModelica and Breathing in an Avalanche.* Prague, Czech Republic, Proceedings of the 12th International Modelica Conference.
- Vytvytskyi, L., Sharma, R. & Lie, B., 2015. Model based control for run-of-river system. Part 1: Model implementation and tuning. *Modeling, Identification and Control*, 36(4), p. 237–249.
- Winkler, D. et al., 2011. *Modelling and Optimisation of Deviation in Hydro Power Production.* Dresden, Germany, Modelica Conference.

Paper B

Mechanistic model for Francis turbines in OpenModelica

Liubomyr Vytvytskyi, & Bernt Lie

Published in *IFAC-PapersOnLine*, 51(2), 103–108, 2018. Part of special issue: 9th Vienna International Conference on Mathematical Modelling.

Mechanistic model for Francis turbines in OpenModelica

Liubomyr Vytvytskyi, Bernt Lie

University College of Southeast Norway, Porsgrunn, Norway (e-mail: {Liubomyr.Vytvytskyi, Bernt.Lie}@usn.no).

Abstract: Often, simulation tools use an empirical turbine model, typically based on look-up tables for turbine efficiency. Such look-up tables can be based on dimensionless hill charts which again require experimental data, or can be directly fitted to experimental data. Thus, empirical models require extensive experiments on real turbines, and it is not straightforward to adapt such models to other systems.

Instead of using an empirical model, it is of interest to use a simple mechanistic model based on the Euler equations for the Francis turbine. Such a simple mechanistic model still holds a number of design parameters which must be found. However, the design parameters have a physical interpretation, and algorithmic design rules exist for choosing these parameters based on information such as nominal water head and nominal flow rate for the hydropower plant. In consequence, the use of such a mechanistic model enables simulation of hypothetical systems with reasonable accuracy, without having to wait until the system is built.

This paper describes as the mechanistic model and a design algorithm for the Francis turbine. The results of the design algorithm and model dynamics (shaft power and hydraulic efficiency) are also compared to designs and turbine behaviors found from the commercial turbine design software *Alab*. These comparisons show good fitting (using a least squares error method) that leads to a good confidence in both the design algorithm, and the mechanistic model. The dynamics comparison is done by simulating two different turbine descriptions using OpenModelica and our in-house hydropower library.

© 2018, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: High head hydropower, Francis turbine, mechanistic model, design algorithm, OpenModelica.

1. INTRODUCTION

1.1 Background

The possibility for modelling and simulating hydropower systems as accurately as possible take an important role in order to develop a control structure and to make efficient analysis tools for testing a designed controller for stability and performance in different operating regimes. The focus is aimed at a high head hydropower system in this case, due to its importance in Norway.

1.2 Previous work

The turbine is a vital part of the waterway, and this is where the potential-kinetic energy of the water is converted to power that drives the rotation of the aggregate of the turbine and the generator. The generator then produces electricity, which is delivered to the electric grid. This paper extends previous work by presenting a simple mechanistic model of a Francis turbine, Dixon and Hall (2013), Lie (2017), Giosio et al. (2017), Splavska et al. (2017) together with an algorithm for selecting model parameters.

Gogstad (2012) and Eltvik (2013) have discussed a hydraulic design of the runner and guide vanes of existing Francis turbines. The current study is mostly based on

research works of Brekke, and publications based on his experience on developing and designing hydraulic turbines, Brekke (2001).

A commercial Modelica library for hydropower system exists (*Hydro Power Library* from Modelon AB¹), where an empirical turbine model based on look-up tables for turbine efficiency, is used. Some work on modeling a waterway for the high head hydropower system using OpenModelica has already been presented, Vytvytskyi and Lie (2017).

1.3 Overview of paper

A presentation of the mechanistic Francis turbine model is given together with a design algorithm for the Francis turbine that enables simulation with limited information about the turbine geometry as for existing and for hypothetical hydropower systems. In addition, the mechanistic turbine model is implemented in an in-house Modelica library; this is an alternative approach compared to the commercial *Hydro Power Library*, where an empirical turbine model based on look-up tables is used.

The paper is structured as follows: Section 2 gives system description of the typical Francis turbine, with case studies. Section 3 includes formulation of the mechanistic

¹ <https://goo.gl/5bLdWC>

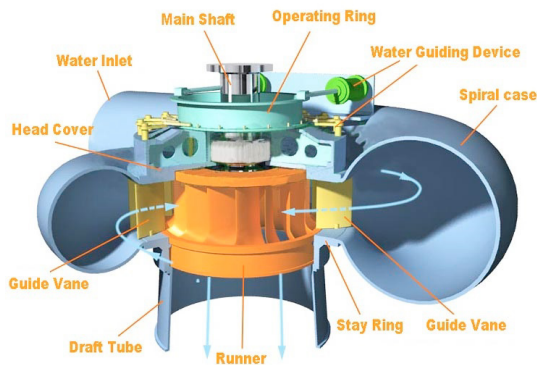


Fig. 1. Structure of the Francis turbine².

turbine model. The presentation of the turbine design algorithm together with turbine design results from the commercial software *Alab* are presented in Section 4. Then simulation with validation and tuning of the developed model is described in Section 5. Finally, discussion and conclusions are given in Section 6.

2. SYSTEM DESCRIPTION

2.1 Overview

A typical structure for the high head hydropower plant has been described in Vytvytskyi and Lie (2017), where the emphasis is on the waterway. A typical Francis turbine for the high head hydropower plants is studied in the current paper. This turbine is used for a wide range of height differences (40 – 600 m) and flow rates (10 – 700 m³/s). A common presentation of the Francis turbine is as shown in Fig. 1².

From Fig. 1 it is observed that water first flows into the spiral case in order to make the flow rate approximately the same for each opening pair of guide vanes. Then the water flows through the controlled guide vanes – the guide vanes steer the water at a certain angle towards the runner blades. In the runner, the water proceeds through the blades which leads to turbine rotation and leads to a pressure drop through the runner. Then the water flows out through the draft tube. The turbine is rotating around an axis connected to a shaft which also holds the rotor of the generator producing electricity to the power grid.

2.2 Geometry

In this paper, two cases are used for the mechanistic turbine model study. Sundsbarm hydropower plant in Telemark, Norway is one of the case studies, where geometry data have been published earlier, Vytvytskyi and Lie (2017), and is presented in tables 1 and 2. Another hydropower system with available data is artificial, Valåmo (2016), and the geometry data are in tables 3 and 4.

Vytvytskyi and Lie (2017) give all needed general operational parameters for simulations.

3. MODELING AND PARAMETRIZATION

The waterway models have been assembled in an in-house Modelica library and used previously for the study of

² <https://goo.gl/VjUnrH>

Table 1. The waterway geometry. Sundsbarm plant, Vytvytskyi and Lie (2017)

Waterway unit	Height difference, [m]	Length, [m]	Diameter, [m]
Reservoir	48	–	–
Intake race	23	6600	5.8
Penstock	428.5	600	3
Surge tank	120	140	3.4
Discharge	0.5	600	5.8
Tail water	5	–	–

Table 2. The turbine nominal operational values. Sundsbarm plant, Vytvytskyi and Lie (2017)

Turbine type	Nominal head, [m]	Nominal flow rate, [m ³ /s]	Nominal power, [MW]
Francis	460	24.3	104.4

Table 3. The waterway geometry. Artificial plant, Valåmo (2016)

Waterway unit	Height difference, [m]	Length, [m]	Diameter, [m]
Reservoir	48	–	–
Intake race	15.5	3000	5
Penstock	133	300	4
Penstock	89	200	3
Surge tank	120	140	3.4
Discharge	0.5	600	5.8
Tail water	5	–	–

Table 4. The turbine nominal operational values. Artificial plant, Valåmo (2016)

Turbine type	Nominal head, [m]	Nominal flow rate, [m ³ /s]	Nominal power, [MW]
Francis	270	20.76	52.85

penstock model complexity, Vytvytskyi and Lie (2017). This paper emphasizes the formulation of the Francis turbine model and a design algorithm for this turbine.

3.1 Model presentation

Lie (2017) gives a mechanistic model for the Francis turbine based on the Euler turbine equations. The key quantities of the model are shown in Fig. 2, and the shaft power \dot{W}_s produced in the Francis turbine is defined as follows:

$$\dot{W}_s = \dot{m}\omega \left(R_1 \frac{\dot{V}}{A_1} \cot \alpha_1 - R_2 \left(\omega R_2 + \frac{\dot{V}}{A_2} \cot \beta_2 \right) \right). \quad (1)$$

Here, \dot{m} and \dot{V} are the mass and volumetric flow rate through the turbine, respectively, ω is the angular velocity of the runner. R_1 and R_2 are the inlet and outlet radiuses of the runner, respectively. A_1 and A_2 are the inlet and outlet cross sectional areas, respectively, and can be defined using runner dimensions: R_1 , R_2 , and w_1 which is the inlet width/height of the runner/blades. α_1 is the inlet guide vane angle that is given by a control signal. β_2 is the outlet blade angle.

The total work rate \dot{W}_t removed through the turbine is:

$$\dot{W}_t = \dot{W}_s + \dot{W}_{ft} + \Delta p_v \dot{V}. \quad (2)$$

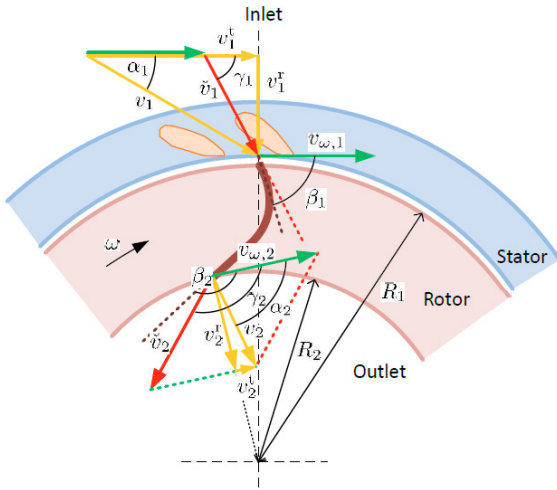


Fig. 2. Key quantities in the Francis turbine model, with blade angles β_1 and β_2 . The water effluent comes out from the paper plane, Lie (2017).

Here, Δp_v is the pressure loss across the guide vane due to friction, and is often neglected. The total work rate might be also formulated based on Bernoulli's law: $\dot{W}_t = \Delta p_{tr} \dot{V} + \frac{1}{2} \dot{m} \dot{V}^2 (\frac{1}{A_0^2} - \frac{1}{A_2^2})$, from where the total pressure loss across the turbine Δp_{tr} can be defined; A_0 is the inlet cross section area to the spiral case. \dot{W}_{ft} – the friction term that represents various friction losses within the turbine is calculated as follows:

$$\dot{W}_{ft} = k_{ft,1} \dot{V} (\cot \gamma_1 - \cot \beta_1)^2 + k_{ft,2} \dot{V} \cot^2 \alpha_2 + k_{ft,3} \dot{V}^2. \quad (3)$$

Here, $k_{ft,1}$, $k_{ft,2}$ and $k_{ft,3}$ are friction coefficients that represent shock, whirl, and pipe friction losses, respectively. These coefficients are tuning parameters for the mechanistic Francis turbine model. β_1 is the inlet blade angle which in the nominal operation condition should be equal to the angle of the relative velocity γ_1 to achieve an influent no-shock condition (the angle of the relative velocity is define from: $\cot \gamma_1 = \cot \alpha_1 - \frac{\omega R_1}{V} A_1$). To satisfy the no-whirl effluent condition, angle α_2 should be equal 0. This angle is defined as $\cot \alpha_2 = \cot \beta_2 + \frac{\omega R_2}{V/A_2}$.

The efficiency of the turbine can be defined as follows:

$$\eta = \frac{\dot{W}_s}{\dot{W}_t} \quad (4)$$

4. TURBINE DESIGN

4.1 Turbine design algorithm

Geometry parameters for the Francis turbine must be found in order to use the mechanistic turbine model as presented above. These parameters, such as blade angles or runner dimensions can be found from design data. Usually, for real (in use) turbines these data are unavailable due to trade confidentiality. Thus it is of interest to develop a design algorithm that can be used to define all the geometry parameters. The structure of this algorithm is shown in Fig. 3, where the input and output values for the design algorithm are presented.

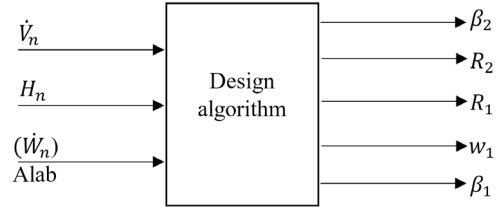


Fig. 3. Block diagram that describes the turbine design algorithm (inputs and outputs).

Brekke (2001) presents a design algorithm for Francis turbine. As input data for the calculation, nominal net head H_n and volumetric flow rate \dot{V}_n are used. A possible turbine design algorithm is as follows, ref. Fig. 2:

- (1) Choose the outlet blade angle β_2 and reference velocity $v_{\omega,2}$. These values are usually in the interval, Gogstad (2012):

$$158^\circ \leq \beta_2 \leq 165^\circ \\ 35\text{m/s} \leq v_{\omega,2} \leq 42\text{m/s} \quad (5)$$

Here, the outlet angle and reference velocity take higher values for higher heads. Brekke (2001) suggests that these values may be chosen as $\beta_2 = 162.5^\circ$ and $v_{\omega,2} = 41\text{m/s}$.

- (2) Define the outlet runner cross section area A_2 (radius R_2) and adjust it together with reference velocity $v_{\omega,2}$ to the normal synchronous rotational speed. First, the meridional velocity is defined as:

$$v_2^r = -\frac{v_{\omega,2}}{\cot \beta_2}, \quad (6)$$

then outlet radius can be define from outlet cross sectional area ($A_2 = \pi R_2^2$):

$$v_2^r = \frac{\dot{V}}{A_2} \Rightarrow R_2 = \sqrt{\frac{\dot{V}}{\pi v_2^r}} \quad (7)$$

Then the turbine rotational speed n [RPM] can be calculated from the angular velocity ($\omega = \frac{\pi n}{30}$):

$$v_{\omega,2} = \omega R_2 \Rightarrow n = \frac{30 v_{\omega,2}}{\pi R_2} \quad (8)$$

After this the turbine speed should be reduced to the nearest synchronous speed (depends on number of pole pairs p in the generator: $n = \frac{60f}{p}$, where frequency f is constant 50 Hz) and then the outlet radius with the reference velocity should be recalculated in reverse order, using (8), (7) and (6).

Normally the information about the turbine rotational speed is available, so the outlet runner radius and the reference velocity can be found directly from (6), (7) and (8).

- (3) Choosing the inlet runner dimension, inlet cross section area A_1 (radius R_1 and width w_1). The inlet radius can be defined from the reference velocity $v_{\omega,1}$ as follows:

$$R_1 = \frac{v_{\omega,1}}{\omega} = \frac{30 v_{\omega,1}}{\pi n} \quad (9)$$

Here, the reference velocity can be chosen from the range for reduced value $\bar{v}_{\omega,1} \in [0.7, 0.75]$, Eltvik (2013), which is dimensionless and expressed as:

$$\bar{v}_{\omega,1} = \frac{v_{\omega,1}}{\sqrt{2gH}} \quad (10)$$

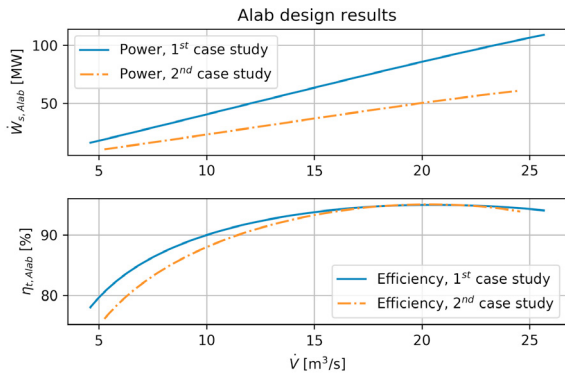


Fig. 4. The turbine efficiency and power curves in respect to the turbine volumetric flow (*Alab* design results).

From Brekke (2001) it is common to use $\bar{v}_{\omega,1} = 0.725$. Regularly, an acceleration of the flow through the runner is desirable, in order to avoid back flow in the runner. That is why the outlet meridional velocity can be chosen approximately ten percent higher than inlet.

$$v_2^r = 1.1v_1^r \quad (11)$$

Then the inlet runner width w_1 can be calculated from the inlet cross sectional area ($A_1 = 2\pi R_1 w_1$):

$$v_1^r = \frac{\dot{V}}{A_1} \Rightarrow w_1 = \frac{\dot{V}}{2\pi R_1 v_1^r} \quad (12)$$

Here, it should be noted that the blade thickness could be included for improving calculation of the inlet cross section area, e.g., 10% of the perimeter, Brekke (2001).

- (4) The inlet blade angle β_1 can be found as follows:

$$\tan(180^\circ - \beta_1) = \frac{v_1^r}{v_{\omega,1} - v_1^t} \quad (13)$$

Here, v_1^t is the tangential velocity and can be defined from dimensionless value $\bar{v}_1^t = 0.48/\bar{v}_{\omega,1}$, using (10) to convert from dimensionless value, Brekke (2001).

4.2 Alab design

In order to validate the Francis turbine model and the turbine design algorithm, commercial software *Alab*³ is used. *Alab* is a tool for hydraulic design of power plants with all possible units, including a turbine design module for various turbine types.

Using the values of the turbine nominal operational regime for the case studies (as mentioned in Fig. 3), the Francis turbines have been designed in *Alab*, where all possible geometry information are provided together with data in a table form for the turbine flow rate, efficiency, and power. These data can be used to compute the turbine efficiency and power curves with respect to the turbine volumetric flow as shown in Fig. 4. It should be noted that the turbine efficiency from *Alab* is not dynamic, but is found by simply dividing the shaft power by the nominal total work: $\eta_t = \dot{W}_s/\dot{W}_t^n$, where $\dot{W}_t^n = \rho g H_n \dot{V}$ and H_n is the nominal static net head of the system.

³ <https://goo.gl/mYDHFx>

4.3 Turbine parameters (design results)

Following the turbine design algorithm, the geometry parameters of the Francis turbine have been calculated for the two case studies. The results are shown in tables 5 and 6, where the parameters from *Alab* are presented as well.

Table 5. Turbine geometry parameters for the Sundsbarm hydropower plant

Design	β_2 , [°]	R_2 , [m]	R_1 , [m]	w_1 , [m]	β_1 , [°]
Algorithm	162.5	0.777	1.32	0.25	117.15
<i>Alab</i>	162.4	0.775	1.32	0.2	109.23

Table 6. Turbine geometry parameters for the artificial hydropower plant

Design	β_2 , [°]	R_2 , [m]	R_1 , [m]	w_1 , [m]	β_1 , [°]
Algorithm	162.5	0.737	1.01	0.3	111
<i>Alab</i>	163.2	0.773	1.01	0.26	109.98

The calculated parameters for both case studies look similar when comparing the turbine design algorithm and the results from the *Alab* software. Some differences are observed for the inlet runner width w_1 and blade β_1 ; this may be caused by a different philosophy for blade shape mentioned by Brekke (2001), where the outlet meridional velocity is chosen 10% higher than the inlet, which is an opposite to the strategy in the *Alab* software calculation⁴; *Alab* suggests the outlet meridional velocity is 10% lower.

5. SIMULATION

For two case studies, the whole hydropower plants are simulated with geometries that are given in tables 1 and 3 in order to get dynamic results of the Francis turbine behavior. All simulations are done in OpenModelica using a Python API, Lie et al. (2016), where the wider range of possibilities for plotting, analysis and optimization exist (using Python packages *numpy*, *matplotlib*, and *scipy*).

It is of interest to tune the Francis turbine model for both case studies in order to get turbine efficiency and shaft power curves comparable to *Alab* results. The simulations of the Francis turbine model are done both for the turbine geometries given from *Alab* and the design algorithm. However, the model fitting is done just for the mechanistic turbine model with the *Alab* designed geometry parameters.

First, the mechanistic turbine model for both cases have been tuned manually (changing the friction term coefficients). Then, in order to improve the manually tuned parameters, a further tuning is performed using a least squares error (LSE) method, which is simply an optimization problem for minimizing the sum of squared errors between the model results and reference values. In this case, the turbine shaft power is compared to the mechanistic model and *Alab* results. Optimal values for the friction term coefficients can be found with this method using the manually fitted results as initial values. The least squares error method is easily implemented using function *leastsq* from the Python package *scipy.optimize*.

⁴ <https://goo.gl/UJikNj>

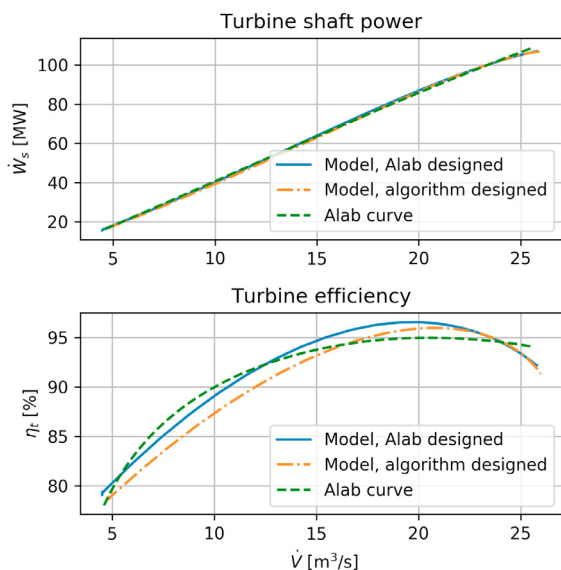


Fig. 5. The turbine shaft power (upper) and efficiency (lower) curves from the mechanistic model tuned by LSE method and designed in *Alab* (Sundsbarm case: $k_{ft,1} = 7.1 \cdot 10^5$, $k_{ft,2} = 1$, $k_{ft,3} = 1.57 \cdot 10^4$).

Sundsbarm case: The set of simulations have been done in order to get the turbine shaft power and efficiency curves (changing the turbine guide vane opening in order to vary the turbine volumetric flow rate). Then these curves have been compared with relevant ones from *Alab* (the turbine efficiency has been calculated in the same way as it is done in *Alab*).

After this, the mechanistic model for the Sundsbarm study with the turbine geometry data from the *Alab* software has been fitted with the least squares method. After the LSE tuning of the mechanistic model, the objective function (sum of squares error) has been reduced by 20% (from 64.73 to 51.90). The results of the tuned Francis turbine model are shown in Fig. 5 (upper plot) for the turbine shaft power, where the tuned values of the friction term coefficients are given in the figure caption. The results for the turbine efficiency are also presented in Fig. 5 (lower plot).

The friction term variations with turbine flow rate are shown in Fig. 6 to better understand their effect on the turbine efficiency and shaft power. This figure shows how the terms affect the friction loss in the turbine depending on the volumetric flow rate.

Artificial plant case: The same manual tuning of the mechanistic Francis turbine model has been done for the second case study. After that the least squares fitting of the mechanistic Francis turbine model with turbine geometry data from the *Alab* software has been performed. Here, the sum of squares error has been reduced by 52% (from 45.34 to 21.65). The results of the turbine shaft power and efficiency from the LSE tuned mechanistic model are presented in Fig. 7 upper and lower plots, respectively. The tuned values of the friction term coefficients for the second case study are also given in the figure caption.

Here, as with the Sundsbarm study, the curves of the turbine shaft power from the mechanistic model fit well

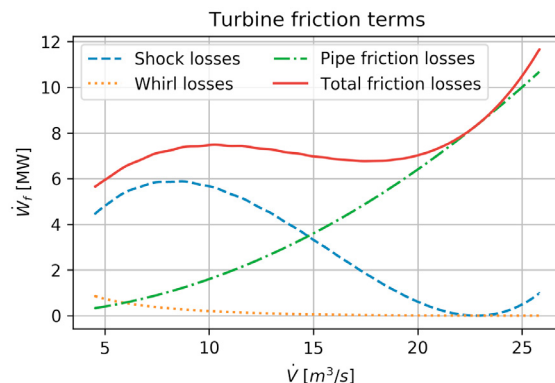


Fig. 6. Simulation results for the turbine friction terms variations in respect to the turbine flow rate (Francis turbine model with *Alab* designed geometry parameters).

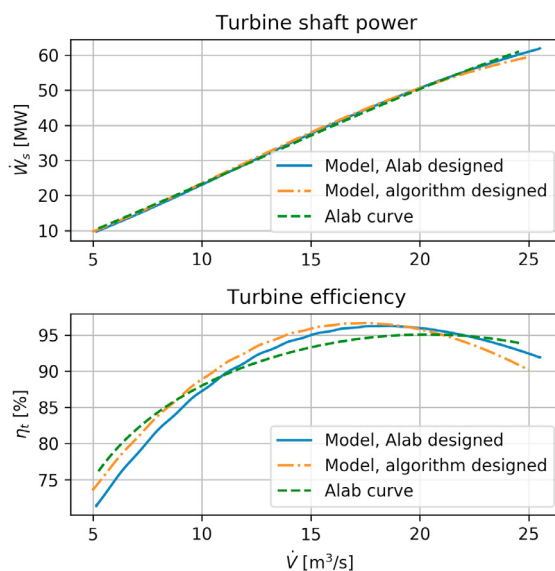


Fig. 7. The turbine shaft power from the mechanistic model tuned by LSE method and designed in *Alab* (Valamo's case: $k_{ft,1} = 1.3 \cdot 10^5$, $k_{ft,2} = 0$, $k_{ft,3} = 4.4 \cdot 10^3$).

the curve designed in *Alab* (see upper plots Fig. 5 and 7). At the same time, the turbine efficiency curves show a bit bigger deviation from the *Alab* results (see lower plots Fig. 5 and 7).

For both case studies, the variation between the efficiency curves from the mechanistic Francis turbine model and *Alab* software may be caused by differences in the design or efficiency calculation and terms that affect it (friction terms). That is why the turbine shaft power looks better than the turbine efficiency when comparing the fitting of the model.

6. DISCUSSION AND CONCLUSIONS

To this end, the Francis turbine model based on Euler equations given in Lie (2017) has been presented. In order to simulate this mechanistic model, a set of turbine geometry data for parametrization is needed. To handle this, the turbine design algorithm that can calculate the

geometry base on the nominal turbine parameters such as net head, flow rate and power production has been also proposed.

The turbine design algorithm shows reasonable results of the turbine geometry parameters in comparison to the results from the *Alab* software. There is a small difference in the inlet runner width and blade, which is caused by a different philosophy of blade shape – the runner cross section area increases from inlet to outlet by 10% for the *Alab* software, while it is the opposite for the design algorithm. However, these contrasts in the designed geometry are actually minor and the design algorithm can be used for defining all the geometry parameters for the mechanistic Francis turbine model. Exceptions are for the friction term coefficients that should be further studied in order to develop an algorithm for their calculation with respect to the turbine nominal operational values.

The presented Francis turbine model shows good results that fit the *Alab* model quite well, especially for the turbine shaft power. There is some deviation in the turbine efficiency curves for both case studies, which may be caused by differences in the design or efficiency and friction terms calculation. The *Alab* tool does not include the whole hydropower system dynamics for the calculation, while the mechanistic turbine model uses the system dynamics (use the real/dynamic net head of the system instead of the nominal static head). As mentioned, for better comparison, the turbine efficiency from the mechanistic model has been calculated in the same way as it is done in *Alab*. However, the simulations have been done with the whole hydropower plants (dynamic systems) that probably has lead to the deviation in the turbine efficiency curves. Nevertheless, this Francis turbine model can be further used for different studies of the high head hydropower system.

ACKNOWLEDGEMENTS

Ingunn Granstrøm from Skagerak Kraft AS, Norway, provided the designed in *Alab* turbine data for both case studies. This is gratefully acknowledged.

REFERENCES

- Brekke, H. (2001). *Hydraulic Turbines. Design, Erection and Operation*. Norwegian University of Science and Technology, Trondheim, Norway.
- Dixon, S.L. and Hall, C. (2013). *Fluid Mechanics and Thermodynamics of Turbomachinery*. Butterworth-Heinemann, seventh edition.
- Eltvik, M. (2013). *Sediment erosion in Francis turbines*. phdthesis, Norwegian University of Science and Technology, Trondheim, Norway.
- Giosio, D.R., Henderson, A.D., Walker, J.M., and Brandner, P.A. (2017). Physics-based hydraulic turbine model for system dynamic studies. *IEEE Transactions on Power Systems*, 32(2), 1161 – 1168. doi:10.1109/TPWRS.2016.2574330.
- Gogstad, P.J. (2012). *Hydraulic design of Francis turbine exposed to sediment erosion*. mathesis, Norwegian University of Science and Technology, Trondheim, Norway.
- Lie, B. (2017). Lecture notes of course Modelling of Dynamic Systems FM1015. techreport, University College of Southeast Norway, Porsgrunn, Norway.
- Lie, B., Bajracharya, S., Mengist, A., Buffoni, L., Kumar, A., Sjölund, M., Asghar, A., Pop, A., and Fritzon, P. (2016). API for Accessing OpenModelica Models From Python. Proceedings of EuroSim 2016, Oulu, Finland.
- Splavska, V., Vytvytskyi, L., and Lie, B. (2017). Hydropower Systems: Comparison of Mechanistic and Table Look-up Turbine Models. Proceedings of 58th SIMS Conference, Reykjavik, Iceland.
- Valåmo, S.B. (2016). *Transient Modelling of Hydropower Plants*. mathesis, Norwegian University of Science and Technology, Trondheim, Norway.
- Vytvytskyi, L. and Lie, B. (2017). Comparison of elastic vs. inelastic penstock model using OpenModelica. Proceedings of 58th SIMS Conference, Reykjavik, Iceland.

Paper C

Linearization for Analysis of a Hydropower Model using Python API for OpenModelica

Liubomyr Vytvytskyi, & Bernt Lie

Published in *Proceedings of the 59th Conference on Simulation and Modelling (SIMS59)*, 153, 216–221. Linköping University Electronic Press, 2018.

Linearization for Analysis of a Hydropower Model using Python API for OpenModelica

Liubomyr Vytvytskyi Bernt Lie

Department of Electrical engineering, Information Technology and Cybernetics, University of South-Eastern Norway, Porsgrunn, Norway, {Liubomyr.Vytvytskyi,Bernt.Lie}@usn.no

Abstract

Even though almost all processes in the real world are described by nonlinear models, nonlinear theory for analysis of these models is far less developed than the theory for linear models. Therefore model linearization is important in order to make efficient analysis tools for these models.

This paper describes the possibility of automatic linearization in Python for a hydropower system modeled in OpenModelica using our in-house hydropower Modelica library *OpenHPL*. Linearization is made using a Python API. Simple uses of the linearized model for analysis and synthesis are indicated.

Keywords: linearization, hydropower, Python API, OpenModelica

1 Introduction

1.1 Background

A transition towards more renewable energy sources is currently taking place in Europe and all over the world. This situation leads to increase in the use of flexible hydropower plants to compensate the highly changing production from intermittent energy sources such as wind and solar irradiation. A high head hydropower system is considered for this study, because it takes up the main part of all hydropower plants in Norway.

The possibility for modelling and simulating this hydropower system takes an important role in order to make efficient analysis tools for testing a designed controller for stability and performance in different operating regimes. One of such analysis tools can provide automatic linearization; an approximate linear model allows for the use of linear theory for analysis and synthesis which is much better developed than nonlinear theory.

1.2 Previous Work

Basic mathematics and control theory needed to model, analyze, and design feedback systems are provided in (As-tröm and Murray, 2010). Based on these methods, the linearized hydropower model can be further tested and analyzed for control purposes using a Python¹ package — *python-control* (The Python Control Systems Library)².

¹<https://www.python.org>

²<https://goo.gl/MtbYtf>

python-control is a Python module, where basic operations for analysis and design of feedback control systems are implemented.

A Python API³ for OpenModelica⁴ already exists that provides possibilities for controlling simulations of the OpenModelica models via Python (Lie et al., 2016). Python in turn gives much wider possibilities for plotting, analysis, and optimization (e.g., using Python packages *matplotlib*, *numpy*, *scipy*, etc.).

Some work on modeling a waterway for the high head hydropower system together with a generator, the Francis turbine, and a governor, has already been done using OpenModelica (Vytvytskyi and Lie, 2017, 2018). Unit models have been assembled in our in-house Modelica⁵ library *OpenHPL*⁶.

1.3 Overview of Paper

In this paper, the main contribution is investigation of how modern computer tools can make the workflow of analysis and design, including linearization and linear control analysis/design. This is the first paper that demonstrates how linearization can be done using the Python API for “non-academic” hydropower models of different complexity. Model implementation is done in OpenModelica using the *OpenHPL* library.

The paper is structured as follows: Section 2 gives a system description of a high head hydropower system. Section 3 gives an overview of the modeling tools and a presentation of the hydropower model. Then automatic linearization and a simple PI controller design are described in Sections 4 and 5. Finally, discussion and conclusions are given in Section 6.

2 System Description

High head plants typically collect and store water in reservoirs in mountains, with tunnels leading the relatively small flow of water down a considerable height difference to the aggregated turbine and generator. The electricity, produced by the generator, is then transferred through power lines to consumers. A typical structure for the high

³<https://goo.gl/Qyjqq2>

⁴<https://openmodelica.org>

⁵<https://www.modelica.org>

⁶Open Hydro Power Library is developed by the first author within his PhD study.

head hydropower plant is depicted in Figure 1 (Vytvytskyi and Lie, 2017).

For simulations in this paper, the data from the Sundsbarm hydropower plant in Telemark, Norway is used with data provided in (Winkler et al., 2011), see Table 1 and 2.

Table 1. The waterway geometry of Sundsbarm hydropower plant.

Waterway unit	Height difference, m	Length, m	Diameter, m
Reservoir	48	—	—
Conduit	23	6600	5.8
Penstock	428.5	600	3
Surge tank	120	140	3.4
Discharge race	0.5	600	5.8
Tail water	5	—	—

Table 2. The turbine geometry of Sundsbarm hydropower plant.

Turbine type	Nominal head, m	Nominal flow rate, m ³ /s	Nominal power, MW
Francis	460	24.3	104.4

3 Modeling

3.1 Modeling Tools

All modeling is done in OpenModelica, which is an open-source Modelica-based modeling and simulation environment intended for industrial and academic usage⁷.

For modeling the hydropower system, library *OpenHPL* is used. This is an in-house hydropower library, where different parts of the waterway components, such as reservoir, conduit, surge tank and turbine, have been assembled. In this library, different waterway components of the hydropower system are described by both mass and momentum balance, and could include compressible/incompressible water or elastic/inelastic pipe walls. A better overview of the mathematical models and methods used in this library is given in (Vytvytskyi and Lie, 2017; Splavska et al., 2017).

In addition, our hydropower library can also be connected with other open source Modelica libraries such as *OpenIPSL*⁸ (Open-Instance Power System Library), where a much wider variety of power system components are presented. Together, the *OpenHPL* and *OpenIPSL* libraries give a possibility to develop a model for the whole hydropower system that starts from the water in the reservoir and ends with the different electrical loads. Linearization also works for more complex/detailed models than

⁷Some tutorials exist for Modelica — <http://book.xogeny.com>, and OpenModelica — <https://goo.gl/76274H>

⁸<http://openipsl.readthedocs.io/en/latest>

used here (e.g., a model for the whole hydropower system), but space limitations restrict our presentation to simpler cases.

3.2 Model Presentation

In this study, two cases of complexity for this system are considered:

1. Simplified system with incompressible water and inelastic pipe.
2. More complex system that includes water compressibility and pipe shell elasticity in the penstock.

Both these cases are straightforward to implement in OpenModelica using the *OpenHPL* library. A block diagram that is relevant for both cases of the hydropower system is presented in Figure 2. For simplicity, the water levels in reservoir and tail water are considered to be constant.

In both cases, the model has one input — turbine gate opening — u_{tr} , and one output — turbine volumetric flow rate — \dot{V}_{tr} .

4 Linearization

4.1 Overview

The Python API (Lie et al., 2016) provides a linearization function that allows approximation of nonlinear DAE models in OpenModelica to linear state space models in Python.

First, the Modelica model is instantiated in Python using the *OMPpython* package and the following command:

```
hps_s=ModelicaSystem("OpenHPL.mo", "
    OpenHPL.Tests.HPLiniazation", "
    Modelica") // for simpler model
hps_kp=ModelicaSystem("OpenHPL.mo", "
    OpenHPL.Tests.HPLiniazationKP", "
    Modelica") // for complex model
```

After this, the input signal and simulation options are set in Python for the simulation. Before linearization, the model parameters are set to steady state values. Automatic linearization is done from Python, where the matrices for the general state-space representation of a linear system are given using the following command:

```
As,Bs,Cs,Ds = hps_s.linearize() // for
    simpler model
Akp,Bkp,Ckp,Dkp = hps_kp.linearize() // for
    complex model
```

4.2 Simple Model

First, the simple hydropower model is linearized. Through linearization, the state-space matrices A , B , C and D are generated:

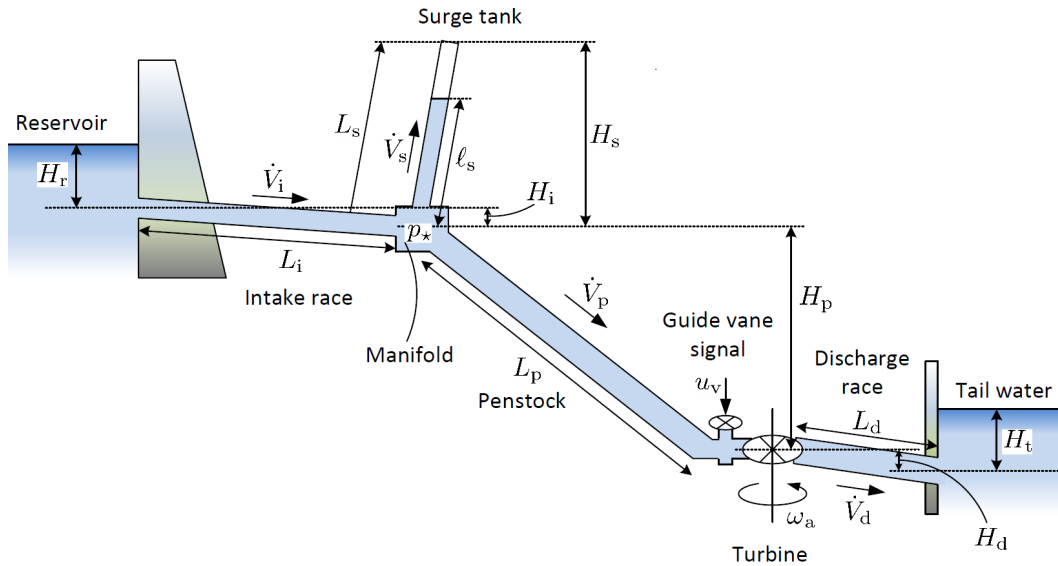


Figure 1. Structure of the high head hydropower plant.

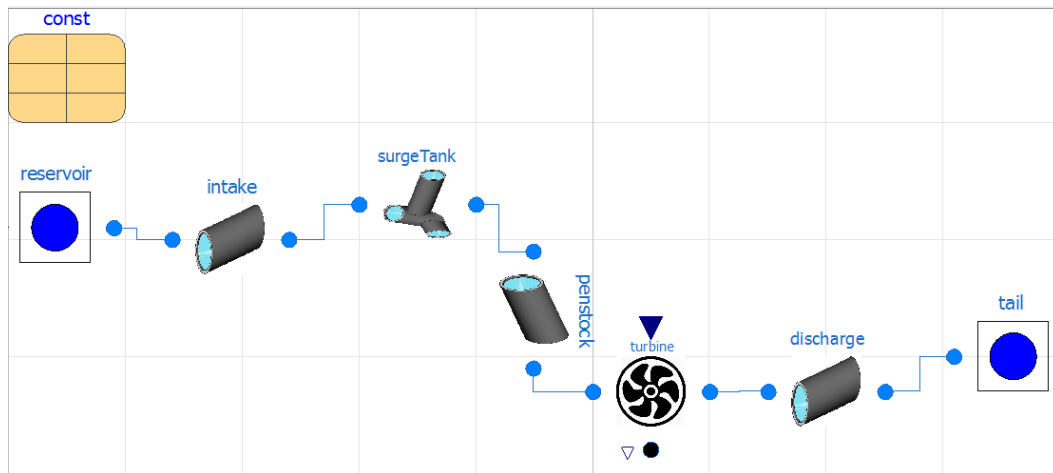


Figure 2. Model of the hydropower system.

$$A = \begin{bmatrix} -4.4 & 3.4 \cdot 10^{-4} & 7.7 \cdot 10^{-6} \\ 4.2 & -4.0 \cdot 10^{-3} & -1.1 \cdot 10^{-5} \\ 0.0 & 997 & 0.0 \end{bmatrix} \quad (1)$$

$$B = \begin{bmatrix} 110.45 \\ -106.62 \\ 0.0 \end{bmatrix} \quad (2)$$

$$C = [1.0 \ 0.0 \ 0.0] \quad (3)$$

$$D = [0.0] \quad (4)$$

Information about the state, input and output variables and their order for the linearized simple model can be checked in Python using the following commands:

```
hps_s.getLinearInputs() // for inputs
hps_s.getLinearOutputs() // for outputs
hps_s.getLinearStates() // for states
```

As mentioned above, this model has one input $u = u_{tr}$ and one output $y = \dot{V}_{tr}$. The linearization algorithm gives

a state vector with 3 elements: $x = [\dot{V}_p, \dot{V}_s, m_s]^T$. Here, \dot{V}_p and \dot{V}_s are the volumetric flow rates in the penstock and surge tank respectively, m_s is the water mass in the surge tank. The linearization algorithm has actually considered two more states (the water masses in the reservoir and tail water). However, due to assumption of the constant water level in those compartments, their rows in the A matrix are zero vectors and can be neglected.

As seen, for this simpler case the linear model is of low order. It is also known that the system is asymptotically stable if all eigenvalues of the A matrix have negative real parts. Using the following command from the *numpy* package, we find the eigenvalues:

```
linalg.eig(As)
```

The eigenvalues of A matrix are as follows:

$$\text{eig}(A) = \begin{bmatrix} -4.367 \\ -0.003 + 0.06j \\ -0.003 - 0.06j \end{bmatrix} \quad (5)$$

4.3 Complex Model

Next, the more complex hydropower model has been linearized in the same way as was presented for the simpler case. The state-space matrices are presented below in simplified form due to their shape:

$$A = \begin{bmatrix} 4.17 & \dots & 0.0 \\ \vdots & \ddots & \vdots \\ 0.0 & \dots & 0.0 \end{bmatrix} \in \mathbb{R}^{22 \times 22} \quad (6)$$

$$B = \begin{bmatrix} 0.0 \\ \vdots \\ 0.0 \end{bmatrix} \in \mathbb{R}^{22 \times 1} \quad (7)$$

$$C = [0.0 \quad \dots \quad 0.0] \in \mathbb{R}^{1 \times 22} \quad (8)$$

$$D = [0.036] \quad (9)$$

The inputs, outputs and states for the linearized complex model are also provided using the following commands:

```
hps_kp.getLinearInputs() // for inputs
hps_kp.getLinearOutputs() // for outputs
hps_kp.getLinearStates() // for states
```

The input and output are the same as for the simpler case. However in this case, the model consists of 22 states that make it more space demanding, $x = [\dot{m}_{p,i}, p_{p,i}, \dot{V}_s, m_s]^T$. Here, two states are also relevant for the surge tank: the volumetric flow rate, \dot{V}_s , and the water mass, m_s . On the other hand, the penstock now is described by 20 equations — 10 for the mass flow rate — $\dot{m}_{p,i}$ and 10 for the pressure — $p_{p,i}$ (here, i is a cell number in range from 1 to n , where n is a number of discretization points of the penstock). This is due to using the Finite Volume method for the discretization of the more complex model with compressible water and elastic pipe walls (the penstock is divided in ten cells here).

In the same way as it was done for the previous simpler case, the eigenvalue analysis of A matrix could be performed. We found that this more complex system is also asymptotically stable.

4.4 Bode Plot Comparison

After the hydropower model has been linearized and the (A, B, C, D) matrices for the general state-space representation are defined for the two cases, some further analysis for the linearized system might be done. For control purposes, the frequency response of a system (Bode plot) can be interesting.

To plot this frequency response, the following commands from the *python-control* package in Python can be used:

```
sys = ss(A, B, C, D)
mag, phase, omega = bode_plot(sys, dB=True)
```

As an alternative, the transfer function $H(s)$ of the system could be found from:

$$H(s) = C(sI - A)^{-1}B + D \quad (10)$$

Here, s is the Laplace operator and for the frequency response, define $s = j\omega$, where ω is frequency in radians. After this, the Bode plot for the linearized hydropower system can be plotted. The Bode diagram for the two cases of the linearized hydropower model are shown in Figure 3.

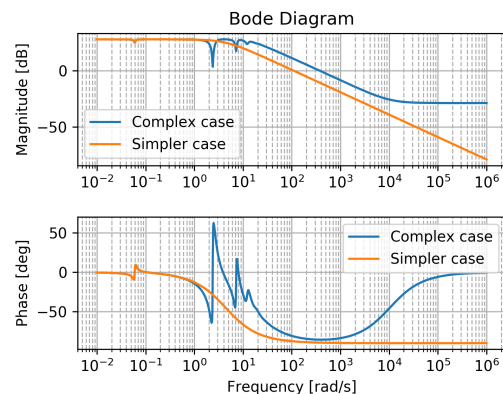


Figure 3. Comparison of Bode plot for two cases of the hydropower system.

5 Design of PI Controller

Using the *python-control* package in Python, a simple PI controller for the linearized hydropower models could be designed and tuned.

First, the step response of the control signal for the linearized hydropower model is found for the two cases, using the following command:

```
sys_s = ss(As, Bs, Cs, Ds) // simple system
youts, Ts = step(sys_s)
sys_kp = ss(Akp, Bkp, Ckp, Dkp) // complex
system
youtkp, Tkp = step(sys_kp)
```

The results of the control signal step response for both the simple and the complex linearized models are shown in Figure 4.

After this, a PI controller $C_r(s) = \frac{K_p s + K_i}{s}$ is tuned.

Then, the controller transfer function is defined in Python using the control package and connected to the hydropower system via feedback using the following commands:

```
CrPI = tf([[Kp, Ki]], [[1., 0.]])
Trs = feedback(sys_s*CrPI, 1)
youts, Ts = step(Trs)
Trkp = feedback(sys_kp*CrPI, 1)
youtkp, Tkp = step(Trkp)
```

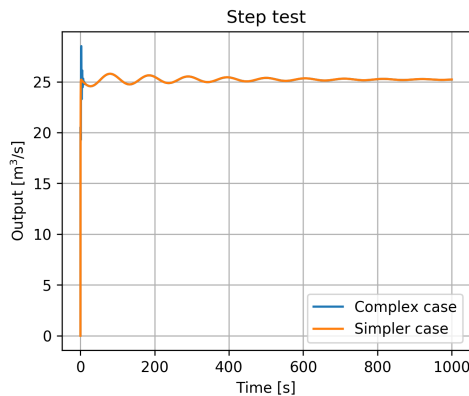


Figure 4. Comparison of the step response for the simple and complex linearized models.

The results of the step response for the reference value for the PI controller that control the hydropower system for the two cases are shown in Figure 5 and Figure 6. For the two cases, the step response is done for two sets of controller parameters.

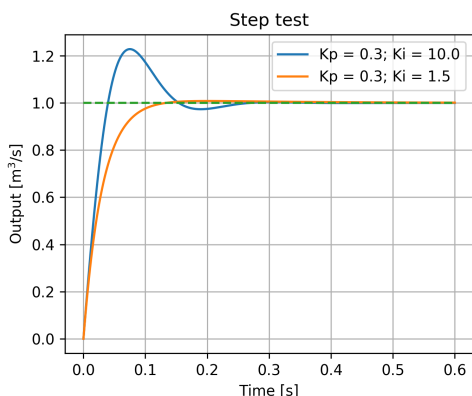


Figure 5. Step response for the simpler model with PI controller.

Finally, the designed and tuned PI control could be checked on the original (nonlinear) hydropower model in OpenModelica. The results of the step test for the output and input are shown in Figure 7 — for the simpler case and Figure 8 — for the more complex case.

6 Discussion and Conclusions

The possibility of automatic linearization of OpenModelica models through Python using the Python API has been presented in this paper.

Two cases with different model complexity for the hydropower system have been linearized in order to show the linearization capability of the Python API. Despite the model complexity, the linearization algorithm finds the state space matrices A , B , C , D .

After linearization, linear theory could be further used for the model analysis and synthesis. Examples of analysis has been presented by creating a Bode plot and designing

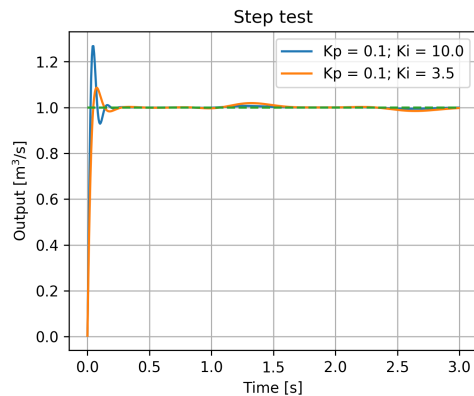


Figure 6. Step response for the complex model with PI controller.

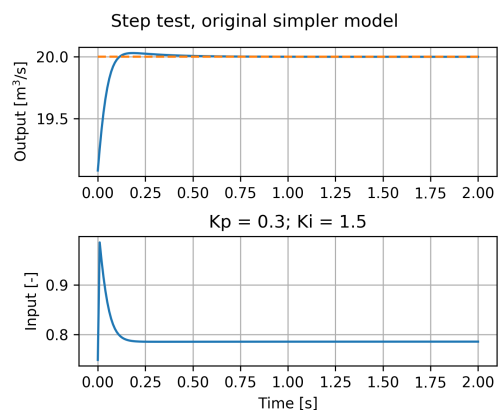


Figure 7. Step test for the nonlinear hydropower model, simpler case.

a simple PI controller, using the already exist packages in Python (*numpy* or *python-control*). The Bode diagram has been plotted for two cases of the hydropower system in order to show the frequency response of the models. Then the possibility of PI controller design has been shown for the two cases. The designed PI controller has also been tested for the original (nonlinear) models in OpenModelica.

Besides the presented examples of linear analysis, many more other possibilities for analysis and synthesis of the linearized model might be used, such as sensitivity or stability analyses, etc.

References

- Karl J. Aström and Richard M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2010.
- Bernt Lie, Sudeep Bajracharya, Alachew Mengist, Lena Bufoni, Arun Kumar, Martin Sjölund, Adeel Asghar, Adrian Pop, and Peter Fritzson. API for Accessing OpenModelica Models From Python. *Proceedings of EuroSim 2016, Oulu, Finland*, 2016.
- Valentyna Splavska, Liubomyr Vytvytskyi, and Bernt Lie. Hy-

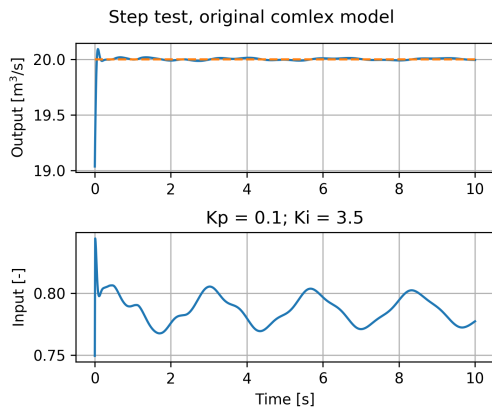


Figure 8. Step test for the nonlinear hydropower model, complex case.

dropower Systems: Comparison of Mechanistic and Table Look-up Turbine Models. *Proceedings of 58th SIMS Conference, Reykjavik, Iceland, 2017.*

Liubomyr Vytvytskyi and Bernt Lie. Comparison of elastic vs. inelastic penstock model using OpenModelica. *Proceedings of 58th SIMS Conference, Reykjavik, Iceland, 2017.*

Liubomyr Vytvytskyi and Bernt Lie. Mechanistic model for Francis turbines in OpenModelica. *Proceedings of Mathmod conference, Wien, Austria, 2018.*

Dietmar Winkler, Hege M. Thoresen, Ingvar Andreassen, Magamage A. S. Perera, and Behzad R. Sharefi. Modelling and Optimisation of Deviation in Hydro Power Production. In *Modelica Conference, 2011.*

Paper D

Combining Measurements with Models for Superior Information in Hydropower Plants

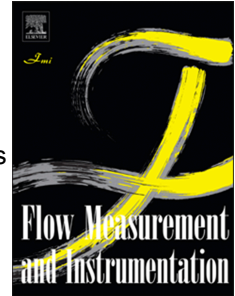
Liubomyr Vytvytskyi, & Bernt Lie

Articles in Press in *Flow Measurement and Instrumentation*, 2019.

Accepted Manuscript

Combining Measurements with Models for Superior Information in Hydropower Plants

Liubomyr Vytvytskyi, Bernt Lie



PII: S0955-5986(18)30234-6

DOI: <https://doi.org/10.1016/j.flowmeasinst.2019.101582>

Article Number: 101582

Reference: JFMI 101582

To appear in: *Flow Measurement and Instrumentation*

Received Date: 28 June 2018

Revised Date: 19 November 2018

Accepted Date: 4 March 2019

Please cite this article as: L. Vytvytskyi, B. Lie, Combining Measurements with Models for Superior Information in Hydropower Plants, *Flow Measurement and Instrumentation*, <https://doi.org/10.1016/j.flowmeasinst.2019.101582>.

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Combining Measurements with Models for Superior Information in Hydropower Plants

Liubomyr Vytvytskyi, Bernt Lie

University of South-Eastern Norway, Porsgrunn, Norway (e-mail: {Liubomyr.Vytvytskyi, Bernt.Lie}@usn.no).

Abstract

Water flow and pressure measurements play essential roles in the operation of hydropower plants. For all methods of measuring flow and pressure, there is a level of uncertainty with regards to sensor noise and sensor failure. In addition, measurements in key locations are hard to obtain. A combination of measurements with a mathematical model of a hydropower plant can be used to improve information about and operation of the hydropower system.

This paper describes the possibility of using nonlinear estimators such as Ensemble or Unscented Kalman filters in order to estimate the states of the hydropower system based on water flow and/or pressure measurements. The implementation of the estimators is done in Python using a Python API for operating OpenModelica simulations, where the hydropower system is modeled using an in-house hydropower Modelica library — *OpenHPL*.

Keywords: Nonlinear state estimation, modelling, hydropower, OpenModelica.

1. Introduction

1.1. Background

A transition towards more renewable energy sources is currently taking place in Europe and elsewhere. This situation leads to an increase in the use of flexible hydropower plants to compensate for the profoundly changing production from intermittent energy sources such as wind and solar irradiation. For this reason, the maintenance and optimal management of existing hydropower plants has become a crucial task.

Optimal operation strongly depends on accurate knowledge and monitoring of the ongoing processes via measuring critical quantities of a hydropower system. However, in many cases, some of these quantities of interest cannot be directly measured. Therefore, it is of interest to consider a combination of available measurements with a mathematical model to estimate the needed quantities and improve the quality of information in hydropower plants.

Popular state estimation methods include the Kalman filter (KF) with a wide range of extensions that apply depending on model type, computational effort, etc. Due to nonlinearities in the hydropower model, nonlinear types of KF such as Unscented (UKF) and Ensemble (EnKF) Kalman filters are considered in this study. The classical nonlinear estimator, the Extended Kalman fil-

ter (EKF), is widely used, too. However, the EKF assumes the existence of the model state Jacobian, and has relatively poor accuracy due to the linear approximation used in the Kalman gain computation. On the other hand, the UKF and EnKF take more advantage of the nonlinear model in the Kalman gain computation, and theoretically leads to better performances.

1.2. Previous work

Modern state estimation theory appeared in the middle of the 20th century, and since then the data assimilation idea has spread to almost all areas of engineering and science. The basic presentation of a variety of state estimation techniques (standard KF, EKF, UKF, etc.) for a general system is provided in Simon (2006); Julier and Uhlmann (1997). The EnKF is a technique for state and parameter estimation, Evensen (2009). An implementation of the UKF to estimate states of a hydropower plant that balances the uncertainty in pressure measurements, has been demonstrated in Zhou and Glemmestad (2012). The use of the EnKF to predict runoff or groundwater flow for hydrology models that are slightly related to hydropower systems is given in Zou et al. (2017) and Shi et al. (2012).

State estimation is traditionally used with mechanistic models. However, state estimation can also be used with data driven models, e.g., artificial neural networks

(ANN). See, e.g., Murphy (2012); Farrell and Polycarpou (2006). An application of ANN for estimation of hydropower plant water inflow is provided in, e.g., Stokelj and Golob (2000); Sacchi et al. (2007). However, data driven models provide a physical interpretation only for the inputs and outputs of the system, while for mechanistic models, internal states and auxiliary variables also have a physical interpretation. This means that with a mechanistic model, it is possible to find internal quantities in the system with some certainty. In data driven models, internal quantities have no physical interpretation unless they are calibrated during an extended and complex experimental phase.

Vytvytskyi and Lie (2017, 2018) discuss work on modeling a waterway for high head hydropower systems together with a generator, the Francis turbine, and a governor using OpenModelica¹. Unit models have been assembled in our in-house Modelica² library *OpenHPL*.

A Python API³ for OpenModelica already exists which provides possibilities for controlling simulations of the OpenModelica models via Python⁴, Lie et al. (2016). Python in turn, gives much broader possibilities for plotting, analysis, and optimization compared to what is possible in OpenModelica.

1.3. Overview of paper

The main contribution of this paper is an investigation of the possibility of state estimation for a hydropower system using nonlinear Kalman filters. Model implementation is done in OpenModelica using the *OpenHPL* library. The estimators are implemented in Python and use the Python API for OpenModelica.

The paper is structured as follows: Section 2 gives a description of the hydropower system. Details of the Ensemble and Unscented Kalman filters are presented in Section 3. Section 4 gives an overview of the *OpenHPL* library and a presentation of the hydropower model. The results of combining the measurements and models for the hydropower system are described in Sections 5. Finally, discussion and conclusions are given in Section 6.

2. System description

A high head hydropower system is considered for this study, due to its significance in Norway. High head

hydropower systems are also more useful for compensating intermittent power than run-of-river systems are, due to their larger buffer capacity from their reservoir.

2.1. System geometry

High head plants typically collect and store water in reservoirs in mountains, with tunnels leading the relatively small flow of water down a considerable height difference to the aggregated turbine and generator. The electricity produced by the generator is then transferred through power lines to consumers. A typical structure for a high head hydropower plant is depicted in Fig. 1, Vytvytskyi and Lie (2017).

For simulations in this paper, data from the Sundsbarm hydropower plant in Telemark, Norway is used with data taken from Vytvytskyi and Lie (2017), see Tables 1 and 2.

Table 1: The waterway geometry of the Sundsbarm hydropower plant.

Waterway unit	Height difference, [m]	Length, [m]	Diameter, [m]
Reservoir	48	—	—
Intake race	23	6600	5.8
Penstock	428.5	600	3
Surge tank	120	140	3.4
Discharge	0.5	600	5.8
Tail water	5	—	—

Table 2: The turbine geometry of the Sundsbarm hydropower plant.

Turbine type	Nominal head, [m]	Nominal flow rate, [m ³ /s]	Nominal power, [MW]
Francis	460	24.3	104.4

2.2. Typical measurements

Typically, measurements from the electric part of hydropower plants are readily available in control systems. Also, water levels (main reservoir and tail water, surge tank) are often available.

In this study, a hydropower model of the waterway is considered, assuming constant water level in reservoirs and with additional simplifications introduced in a later section. Water flow and pressure measurements in the waterway units are of interest. **In real hydropower plants**, only a few of these quantities are measured, e.g., pressure measurements before or after the turbine, water flow rate before the turbine, etc. The turbine flow rate

¹<https://openmodelica.org>

²<https://www.modelica.org>

³<https://goo.gl/Qyjq2>

⁴<https://www.python.org>

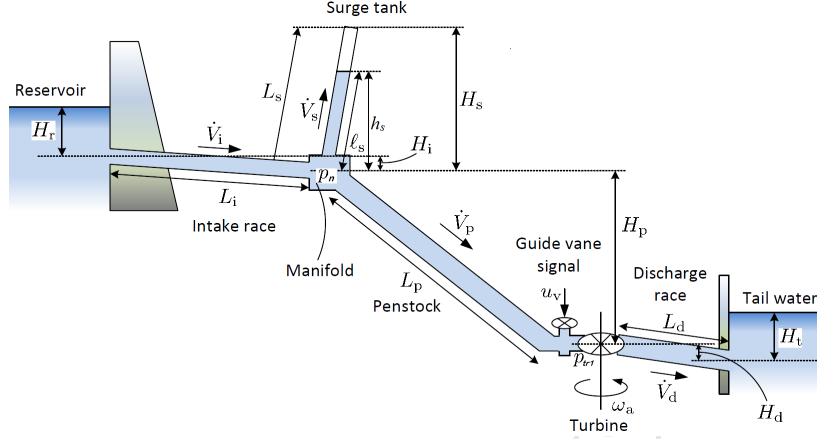


Figure 1: Overview of the structure of a high head hydropower plant.

can sometimes also be approximated from the turbine characteristics.

Typically, the accuracy of the pressure measurements in hydropower plants is in the order of 1%. The flow rate measurement errors can be less precise and are in the range of 1 – 10%, depending on how the measurements are done. For simplicity, an accuracy of 1% is assumed for both pressure and flow rate measurements in this work.

We consider three cases of measurements: the turbine flow rate only is measured for one case, the inlet turbine pressure or the manifold node pressure only is available for the second case, and for the third case, both the flow rate and the inlet turbine pressure are measured. **It should be noted that for the last case with a combination of two measurements, the “less expensive” solution of two measurements at the same position is chosen, — less expensive in the sense that these will be cheaper to install and maintain.**

3. State estimation methods

3.1. Overview

As mentioned above, the hydropower plant model can be highly nonlinear, and use of the widely applied EKF can lead to difficulties in Jacobian computation. Moreover, the EKF rely on a linear approximation to propagate the mean and covariance of the states and this can cause unreliable estimates, Simon (2006). At the same time, the UKF and EnKF are much simpler to use as no Jacobian needs to be computed. Also, they avoid the

linearization problem due to use of the unscented transformation and Monte Carlo simulations of the nonlinear model, respectively. These two methods allow approximating the variation of the mean and covariance of random variables that propagate through the nonlinear model.

In Modelica, models are described as differential and algebraic equations (DAEs), with differential and algebraic variables. OpenModelica by default transforms the DAEs into state space form with auxiliary variables: states are typically a subset of the differential variables, while the auxiliary variables are the remaining variables.

Below, a more detailed description of both the UKF and EnKF are given for a nonlinear dynamic system of the form:

$$\begin{aligned} x_k &= f(x_{k-1}, u_{k-1}, w_{k-1}) \\ y_k &= g(x_k, u_k, v_k) \end{aligned} \quad (1)$$

where, $w_k \sim \mathcal{N}(\bar{w}_k, W_k)$, $v_k \sim \mathcal{N}(\bar{v}_k, V_k)$, $x_1 \sim \mathcal{N}(\bar{x}_1, X_1)$.

Here, $x \in \mathbb{R}^{n_x}$ is the state vector with initial state x_1 which is normally distributed. u is a deterministic input signal vector, and $y \in \mathbb{R}^{n_y}$ is a measurement vector. w and v are vectors of random disturbance and measurement noise, respectively. The disturbance and noise are also assumed normally distributed.

In addition, the notations $\hat{x}_{k|k-1}$ and $\hat{x}_{k|k}$ are introduced. Here, $\hat{x}_{k|k-1}$ is the best possible estimate of x_k when information (i.e., sensor signal values) up to and including time index $k-1$ is used, i.e., $\hat{x}_{k|k-1}$ uses measurements $y_1, \dots, y_{k-2}, y_{k-1}$; $\hat{x}_{k|k}$ is denoted the a pri-

ori estimate. Likewise, $\hat{x}_{k|k}$ is the best possible estimate of x_k when information up to and including time index k is used, i.e., $\hat{x}_{k|k}$ uses the measurements y_1, \dots, y_{k-1}, y_k ; $\hat{x}_{k|k}$ is denoted the *a posteriori* estimate. It follows that estimate $\hat{x}_{k|k}$ should be better/more accurate than estimate $\hat{x}_{k|k-1}$.

The covariance estimate of the states is indicated with symbol X_k , thus $X_{k|k-1}$ is the covariance of $\hat{x}_{k|k-1}$, while $X_{k|k}$ is the covariance of $\hat{x}_{k|k}$. It follows that $X_{k|k-1} \geq X_{k|k}$.

3.2. Ensemble Kalman filter

The Ensemble Kalman filter is based on the use of Monte Carlo simulation of the nonlinear system in order to calculate the cross and innovation covariances. Contrary to the EKF, the EnKF does not need Jacobians in the computation.

First, the initial state is assumed to be normally distributed. Hence, *realizations/particles* of the state are randomly generated in the form of $x_{1|1}^i \sim \mathcal{N}(\bar{x}_1, X_1)$, $i \in \{1, \dots, n_p\}$ where n_p is the number of these realizations/particles. Next, n_p realizations/particles of the state propagates, $x_{k|k-1}^i = f(x_{k-1|k-1}^i, u_{k-1}, w_{k-1}^i)$, and measurements $y_{k|k-1}^i = g(x_{k-1|k-1}^i, u_{k-1}, v_{k-1}^i)$ are similarly computed. Disturbance w_k^i and measurement noise v_k^i are also randomly generated.

The complete EnKF algorithm is given by the following steps:

1. EnKF initialization, $k = 1$:

- draw random initial particle values, $i \in \{1, \dots, n_p\}$:

$$x_{1|1}^i \sim \mathcal{N}(\bar{x}_1, X_1) \quad (2)$$

- *a posteriori* state estimate:

$$\hat{x}_{1|1} = \frac{1}{n_p} \sum_{i=1}^{n_p} x_{1|1}^i \quad (3)$$

- *a posteriori* covariance estimate:

$$X_{1|1} = \frac{1}{n_p-1} \sum_{i=1}^{n_p} (x_{1|1}^i - \hat{x}_{1|1}) \times (x_{1|1}^i - \hat{x}_{1|1})^T \quad (4)$$

2. Propagation step, $k = 2, 3, \dots$:

- draw random disturbance, $i \in \{1, \dots, n_p\}$:

$$w_k^i \sim \mathcal{N}(\bar{w}_k, W_k) \quad (5)$$

- *a priori* state estimate, $i \in \{1, \dots, n_p\}$:

$$x_{k|k-1}^i = f(x_{k-1|k-1}^i, u_{k-1}, w_{k-1}^i) \quad (6)$$

$$\hat{x}_{k|k-1} = \frac{1}{n_p} \sum_{i=1}^{n_p} x_{k|k-1}^i$$

- *a priori* covariance estimate:

$$X_{k|k-1} = \frac{1}{n_p-1} \sum_{i=1}^{n_p} (x_{k|k-1}^i - \hat{x}_{k|k-1}) \times (x_{k|k-1}^i - \hat{x}_{k|k-1})^T \quad (7)$$

3. Information update, $k = 2, 3, \dots$:

- draw measurement noise, $i \in \{1, \dots, n_p\}$:

$$v_k^i \in \mathcal{N}(\bar{v}_k, V_k) \quad (8)$$

- measurement and innovation, $i \in \{1, \dots, n_p\}$:

$$y_{k|k-1}^i = g(x_{k|k-1}^i, u_k, v_k^i) \quad (9)$$

$$\hat{y}_{k|k-1} = \frac{1}{n_p} \sum_{i=1}^{n_p} y_{k|k-1}^i$$

- cross and innovative covariances:

$$Z_{k|k-1} = \frac{1}{n_p-1} \sum_{i=1}^{n_p} (x_{k|k-1}^i - \hat{x}_{k|k-1}) \times (y_{k|k-1}^i - \hat{y}_{k|k-1})^T \quad (10)$$

$$\mathcal{E}_{k|k-1} = \frac{1}{n_p-1} \sum_{i=1}^{n_p} (y_{k|k-1}^i - \hat{y}_{k|k-1}) \times (y_{k|k-1}^i - \hat{y}_{k|k-1})^T \quad (11)$$

- Kalman gain:

$$K_k = Z_{k|k-1} \mathcal{E}_{k|k-1}^{-1} \quad (12)$$

- *a posteriori* state estimate, $i \in \{1, \dots, n_p\}$:

$$x_{k|k}^i = x_{k|k-1}^i + K_k (y_k - y_{k|k-1}^i) \quad (13)$$

$$\hat{x}_{k|k} = \frac{1}{n_p} \sum_{i=1}^{n_p} x_{k|k}^i$$

- *a posteriori* covariance estimate:

$$X_{k|k} = \frac{1}{n_p-1} \sum_{i=1}^{n_p} (x_{k|k}^i - \hat{x}_{k|k}) \times (x_{k|k}^i - \hat{x}_{k|k})^T \quad (14)$$

Based on the similarity to bootstrap statistics, Efron (1982), the number of *realizations/particles* should be equal to, or greater than 50 in order to provide reasonably accurate estimates of mean and covariance. Obviously, it is required that $n_p \geq \min(n_y, n_x)$ to avoid rank loss in the innovation covariance matrix $\mathcal{E}_{k|k-1}$ and the state covariance matrices $X_{k|k-1}$ and $X_{k|k}$.

3.3. Unscented Kalman filter

The main idea of the UKF is closely related to the EnKF in the point that a set of *realizations/particles* are transformed through the nonlinear system, then their results are collected to estimate state mean and covariance. However, in the EnKF the set of n_p *realizations/particles* is randomly generated, whereas in the UKF these points are created based on certain deterministic rules and denoted sigma points, Simon (2006).

The sigma points for state vector $x \in \mathbb{R}^{n_x}$ could be defined based on the mean \bar{x} and covariance X of the states by the following algorithm:

$$x^i = \bar{x} + \tilde{x}^i, \quad i \in \{1, \dots, 2n_x\} \quad (15)$$

where

$$\begin{aligned}\tilde{x}^i &= (\sqrt{n_x X})_i^T, \quad i \in \{1, \dots, n_x\} \\ \tilde{x}^{n_x+i} &= -(\sqrt{n_x X})_i^T, \quad i \in \{1, \dots, n_x\}\end{aligned}\quad (16)$$

where $\sqrt{n_x X}$ is the Cholesky root⁵ of $n_x X$, i.e., $(\sqrt{n_x X})^T \times \sqrt{n_x X} = n_x X$, while $(\sqrt{n_x X})_i$ is the i th row of $\sqrt{n_x X}$.

In addition to the unscented transformation, other possible transformations exist, e.g., the simplex transformation, the spherical transformation, and others, Simon (2006); Julier and Uhlmann (2004). These additional transformations might be applied when computational savings are of interest, or if there are more statistics of the noise, etc.

Similarly to the EnKF, the UKF algorithm can be described as follows:

1. The UKF initialization, $k = 1$:

- *a posteriori* state estimate:

$$\hat{x}_{1|1} = \bar{x}_1 \quad (17)$$

- *a posteriori* covariance estimate:

$$X_{1|1} = X_1 \quad (18)$$

2. Propagation step, $k = 2, 3, \dots$:

- find sigma points:

$$\begin{aligned}x_{k-1|k-1}^i &= \hat{x}_{k-1|k-1} + \tilde{x}^i, \\ i &\in \{1, \dots, 2n_x\}\end{aligned}\quad (19)$$

where

$$\begin{aligned}\tilde{x}^i &= (\sqrt{n_x X_{k-1|k-1}})_i^T, \\ i &\in \{1, \dots, n_x\} \\ \tilde{x}^{n_x+i} &= -(\sqrt{n_x X_{k-1|k-1}})_i^T, \\ i &\in \{1, \dots, n_x\}\end{aligned}\quad (20)$$

- *a priori* state estimate, $i \in \{1, \dots, 2n_x\}$:

$$\begin{aligned}x_{k|k-1}^i &= f(x_{k-1|k-1}^i, u_{k-1}) \\ \hat{x}_{k|k-1} &= \frac{1}{2n_x} \sum_{i=1}^{2n_x} x_{k|k-1}^i\end{aligned}\quad (21)$$

- *a priori* covariance estimate:

$$\begin{aligned}X_{k|k-1} &= \frac{1}{2n_x} \sum_{i=1}^{2n_x} (x_{k|k-1}^i - \hat{x}_{k|k-1}) \\ &\quad \times (x_{k|k-1}^i - \hat{x}_{k|k-1})^T + W_k\end{aligned}\quad (22)$$

3. Information update, $k = 2, 3, \dots$:

- measurement and innovation, $i \in \{1, \dots, 2n_x\}$:

$$\begin{aligned}y_{k|k-1}^i &= g(x_{k|k-1}^i, u_k) \\ \hat{y}_{k|k-1} &= \frac{1}{2n_x} \sum_{i=1}^{2n_x} y_{k|k-1}^i\end{aligned}\quad (23)$$

- cross and innovative covariances:

$$\begin{aligned}Z_{k|k-1} &= \frac{1}{2n_x} \sum_{i=1}^{2n_x} (x_{k|k-1}^i - \hat{x}_{k|k-1}) \\ &\quad \times (y_{k|k-1}^i - \hat{y}_{k|k-1})^T\end{aligned}\quad (24)$$

$$\begin{aligned}\mathcal{E}_{k|k-1} &= \frac{1}{2n_x} \sum_{i=1}^{2n_x} (y_{k|k-1}^i - \hat{y}_{k|k-1}) \\ &\quad \times (y_{k|k-1}^i - \hat{y}_{k|k-1})^T + V_k\end{aligned}\quad (25)$$

- Kalman gain:

$$K_k = Z_{k|k-1} \mathcal{E}_{k|k-1}^{-1} \quad (26)$$

- *a posteriori* state estimate:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (y_k - \hat{y}_{k|k-1}) \quad (27)$$

- *a posteriori* covariance estimate:

$$X_{k|k} = X_{k|k-1} - K_k \mathcal{E}_{k|k-1} K_k^T \quad (28)$$

It is seen that the number of sigma points equals twice the number of states. This means that for systems with few states, the UKF needs lower computation effort than the EnKF does. It is clearly assumed that $n_y \leq 2n_x$ to avoid rank loss in $\mathcal{E}_{k|k-1}$.

It should be noted that in the measurement update, Eq. 23, the previously defined sigma points $x_{k|k-1}^i$ from Eq.(21) have been used for similarity with the EnKF algorithm and to save computational effort. However, the unscented transformation, based on the defined *a priori* state, $\hat{x}_{k|k-1}$, and covariance, $X_{k|k-1}$, estimate, can be used to create new sigma points for the measurement update. The use of these new sigma points can increase the performance of the estimator for nonlinear measurements, Simon (2006).

Both the EnKF and UKF algorithms have been implemented in Python for further use with the model of the hydropower system.

4. Modelling

As mentioned above, a dynamic model of the waterway of a hydropower plant is used in this study. All modelling is done in OpenModelica, which is an open source Modelica-based modelling and simulation environment intended for industrial and academic usage⁶.

4.1. Hydropower library

For modelling the hydropower system, library *OpenHPL* is used. This is an in-house hydropower library, where different parts of the waterway components such as reservoir, conduit, surge tank, and turbine have

⁵Using Python package *numpy*, the Cholesky decomposition can be found by the *numpy.linalg.cholesky()* function.

⁶Some tutorials exist for Modelica — <http://book.xogeny.com>, and OpenModelica — <https://goo.gl/76274H>

been assembled. In this library, different waterway components of the hydropower system are described by both mass and momentum balances, and could include compressible/incompressible water or elastic/inelastic pipe walls. A more detailed overview of the mathematical models and methods used in this library is given in Vytvytskyi and Lie (2017); Splavska et al. (2017).

4.2. Model presentation

In this study, a simplified system with incompressible water and inelastic pipe is considered. A block diagram that is relevant for this hydropower model is presented in Fig. 2. Here, the block diagram consists of drag and drop elements from *OpenHPL* that are structured in the same way as in the hydropower plant, compare with Fig. 1. These elements are also specified with appropriate geometry from Tables 1 and 2. Connectors that join each element hold information about the pressure in the connector and mass flow rate that flows through the connector — similar to the connection in the electrical circuit with voltage and current.

For simplicity, the water levels in the reservoir and tail water are assumed to be constant. Regarding the model states, they consist of volumetric flow rates in the penstock, \dot{V}_p , and surge tank, \dot{V}_s , and the water height in the surge tank, h_s . The model has one input — turbine gate opening u_{tr} , and a few variables might be considered as measured outputs — turbine volumetric flow rate, \dot{V}_{tr} (equals the flow rate in the penstock \dot{V}_p), pressure before the turbine (end of the penstock), p_{tr1} , and pressure in the manifold node (beginning of the penstock), p_n .

This hydropower model can be further used for assimilation with measured data. Using the Python API for OpenModelica, state estimation of this hydropower model is carried out in Python.

5. Results

5.1. Simulation overview

A number of cases for the measured output signal of the model are considered. In two cases, the output is described by only one measured variable. For a third case, two measured outputs are used in order to see if the combination of several measurement can improve the performance of the estimators. The following three cases are considered:

1. Measuring flow — using turbine volumetric flow rate, \dot{V}_{tr} , as the measured output, which means that one of the states is measured (turbine volumetric

flow rate is the same as the flow rate in the penstock).

2. Measuring pressure — using either pressure before the turbine, p_{tr1} , or pressure in the manifold node, p_n .
3. Measuring both flow and pressure — using turbine volumetric flow rate, \dot{V}_{tr} , together with the inlet turbine pressure, p_{tr1} .

Measurement data from a real hydropower plant has not been available. Instead, synthetic/artificial measurements from hydropower model simulations are used. Hence, for all cases below, the hydropower model is first simulated separately to get appropriate synthetic measurements. Next, measurement noise, v_k , is added to the uncorrupted simulated measurement signal and then these synthetic measurements are used for the estimator simulations. Here, the measurement noise covariance is assumed from the accuracy of the measurements and depends on the nominal value of the measured variable. The mean of the measurement noise v_k is the same for all the cases, $\bar{v}_k = 0$, but the noise covariances V_k are different for each case and are as follows:

- Turbine volumetric flow rate, \dot{V}_{tr} , measurement — with nominal value approximately $20 \text{ m}^3/\text{s}$ and accuracy 1%; the measurement noise covariance is $V_k = 0.004$, and $v_k \sim \mathcal{N}(0, 0.004)$.
- Inlet turbine pressure, p_{tr1} , measurement — nominal value is approximately 50 bar and the accuracy is 1%; the measurement noise covariance is set to $V_k \approx 0.03$, and $v_k \sim \mathcal{N}(0, 0.03)$.
- Manifold node pressure, p_n , measurement — nominal value is ca. 8 bar and the accuracy is 1%; the measurement noise covariance is set to $V_k \approx 0.0007$, and $v_k \sim \mathcal{N}(0, 0.0007)$.

Additive random disturbance, w_k , is used in selected simulations to create the synthetic measurements. For simplicity, this random disturbance is given zero mean $\bar{w}_k = 0$ and covariance $W_k = 0.0001$ for all states, i.e. $w_k \sim \mathcal{N}(0, 0.0001)$ — when used. When not used, $w_k \equiv 0$.

Assumed covariances are important design choices in the *estimator*. For optimal estimates, the covariances in the estimator should be equal to the covariances in the real, but unknown, random disturbances and noises. Here, synthetic measurements are used, and for that (unrealistic) case, the true covariances to use in the estimator are known. To test the robustness of the estimator, *disturbance* covariance in the estimator is set to either

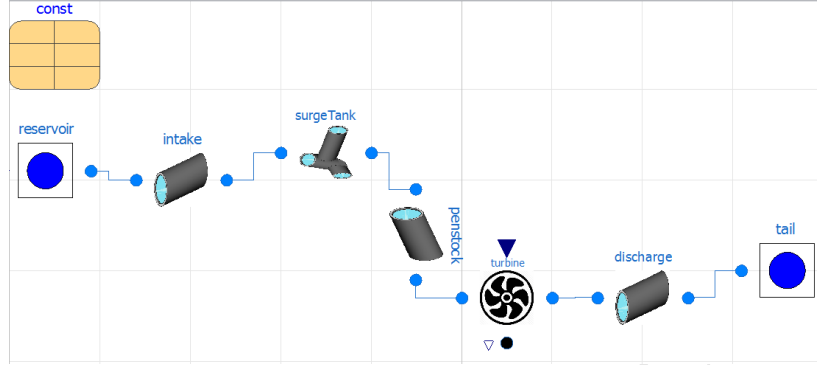


Figure 2: Block diagram of the model of the hydropower system in OpenModelica. Modeled using *OpenHPL*.

zero or $W_k = 10^{-4}$ — thus, the disturbance covariance used in the estimator may deviate from that used for drawing random disturbances for the synthetic measurements. Random measurement noise is always present, and perfect assumption of measurement covariance is used.

For all cases, the simulations start from steady state and last for 60 s with a sample time of 0.5 s. **A disturbance occurs at time 20 s with a closing of the turbine valve: in real systems, the turbine valve is ramped up and down to avoid excessive water hammer effects; in this study, we have used a more informative step change. To limit the damaging effect of water hammer, we have used a 3% step change.** The initial state covariance is the same for all states, $X_1 = 0.01$. Results of state estimation for all cases are shown below, and Table 3 in Section 6 gives a description of all of the simulations with a summary of the estimation results.

The number of *realizations/particles* for the EnKF is set to $n_p = 50$ for all EnKF simulations in this study. The number of UKF sigma points is known from the number of the model states and equals 6. The difference between the number of sigma points for the UKF and the number of *realizations/particles* for the EnKF leads to approximately ten times faster simulation time for the UKF.

5.2. Flow measurement

First, a comparison of estimates from the two KF algorithms (UKF and EnKF) and the model simulations is done for the hydropower system with correct knowledge of zero random disturbance, see Fig. 3.

Note that the information about the measurement noise is used in the estimators, for more details see Table 3. The results from the auxiliary variables (pressures

p_{tr1} — before the turbine and p_n — in the manifold) are also shown in the figure for clarity. These auxiliary variables reflect the states because they are functions of the states. Furthermore, uncertainty ranges for the state estimates for both the UKF and EnKF are demonstrated by the filled area between two boundaries with the same but transparent color as for the related state estimates. These boundaries are calculated as two standard deviations of the state estimates, where the standard deviation is found as the square root of the diagonal element (variance) of the covariance matrix $X_{k|k}$.

Figure 3 shows that the UKF provides better estimates than the EnKF for two states (surge tank flow rate \dot{V}_s , and water height h_s) that are not measured. It is also seen that the uncertainty range for the state estimates converges faster for the UKF than the EnKF, which might be a reason of the UKF's better performance.

To ensure that both the UKF and EnKF provide proper steady state results, a steady state simulation (with constant input signal u_{tr}) for an extended period of 800 s is performed for this case only, with the flow measurement, and without random disturbance, see Fig. 4. Here, the initial value for one of the estimator states deviates slightly from steady state in order to show that estimators results converge to the correct steady state values. From Fig. 4 it is seen that the EnKF needs a shorter time to converge to the steady state than the UKF.

In order to show the impact of the random disturbance on the UKF and EnKF, the estimators have been run with the same synthetic measurements, but with incorrectly assumed disturbance covariance $W_k \geq 0$ in the estimator, see Table 3. The comparison of the estimators with and without disturbance covariance is shown

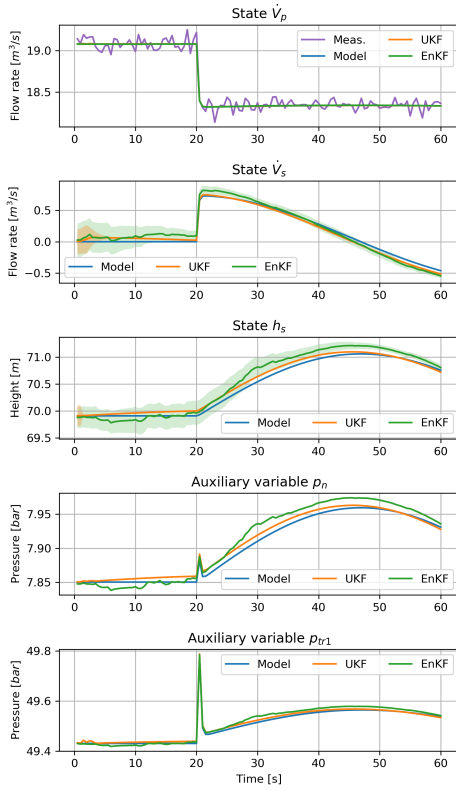


Figure 3: State and auxiliary variable comparisons for the UKF and EnKF without random disturbance in the synthetic data, and with correct assumption of disturbance covariance in the estimators. Turbine flow rate is measured.

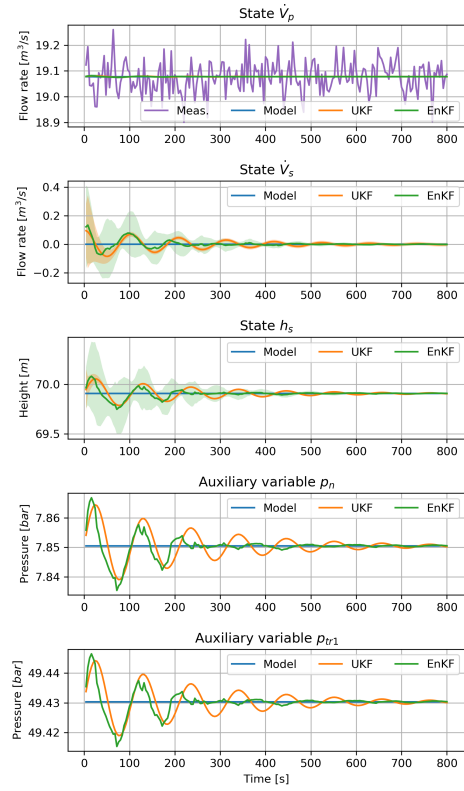


Figure 4: Steady state simulation. State and auxiliary variable comparisons for the UKF and EnKF without random disturbance in the synthetic data, and with correct assumption of disturbance covariance in the estimators. Turbine flow rate is measured.

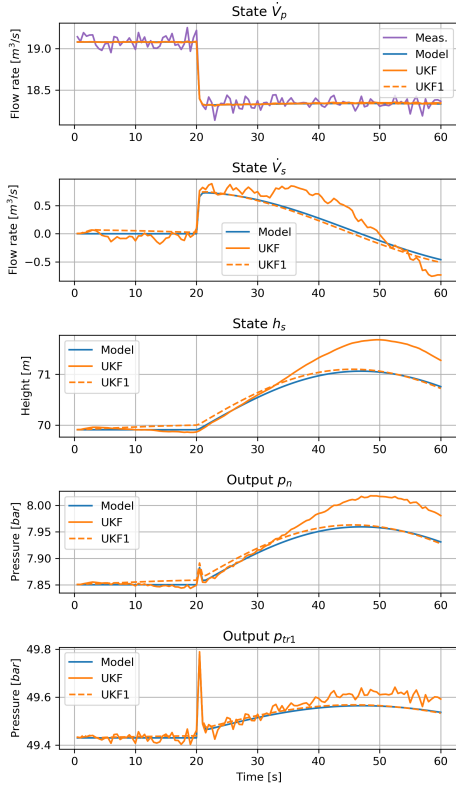


Figure 5: State and auxiliary variable comparisons for the UKF with too large disturbance covariance and UKF1 with correct disturbance covariance. Turbine flow rate is measured.

in Fig. 5 for the UKF and in Fig. 6 for the EnKF. In both figures, the estimators with disturbance covariance $W_k > 0$ are marked as “UKF” or “EnKF” and labels “UKF1” or “EnKF1” are used for the estimators with disturbance covariance $W_k \equiv 0$.

Both Fig. 5 and Fig. 6 show that assuming too large disturbance covariance in the UKF and EnKF makes the estimates more noisy; essentially, assuming larger disturbance covariance is equivalent to assuming lower measurement covariance, thus trusting the measurement too much, with more measurement noise “bleeding” through to the estimates. An interesting side result is that the estimation results of the UKF assuming incorrect disturbance covariance are much poorer than for the UKF with correct disturbance covariance.

Next, estimates of the UKF and EnKF are compared

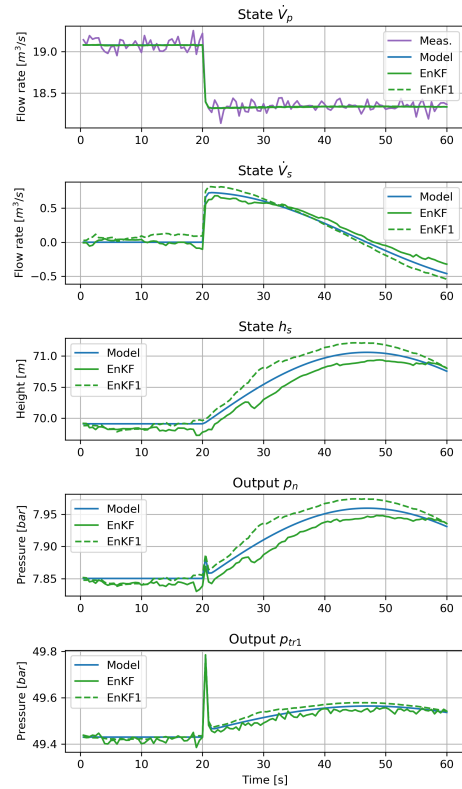


Figure 6: State and auxiliary variable comparisons for the EnKF with too large disturbance covariance and EnKF1 with correct disturbance covariance. Turbine flow rate is measured

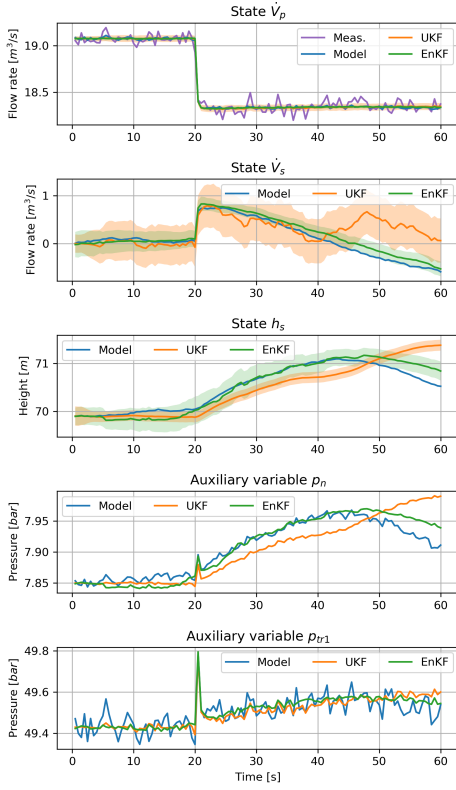


Figure 7: State and auxiliary variable comparisons for the UKF and EnKF with measurement noise and random disturbance in the synthetic measurements, and correct assumption of disturbance covariance in the estimators. Turbine flow rate is measured.

together with the model simulation for the hydropower system when the synthetic measurements include random disturbance w_k and correct disturbance covariance W_k is used in the estimators. For more details see Table 3. The results of this comparison are shown in Fig. 7. It is seen from the figure that the state estimates from the EnKF are close to the model results, while the UKF estimates have more deviation in comparison to the model results.

5.3. Pressure measurement

In addition to the use of the flow rate measurement for the estimators, it is of interest to see how the UKF and EnKF behave when a pressure measurement is used. Both the inlet turbine p_{tr1} and the manifold node p_n

pressures are of interest as measurements. The same set of dynamic simulations that have been done for the flow rate measurement case are performed for the case with pressure measurements. The use of the measured inlet turbine pressure for state estimation is first presented, and then the same state estimates are found based on the manifold node pressure measurement.

5.3.1. Measuring the inlet turbine pressure

Initially, the comparison of estimates from the UKF and the EnKF are done for the hydropower system without random disturbance in the synthetic measurement and with correct assumption of disturbance covariance in the estimators, see Fig. 8. As when measuring the flow rate, information about the measurement noise is also added to the estimators, see Table 3.

Figure 8 shows that the UKF provides a bit better estimates with lower variation with the model simulations than the EnKF for the states \dot{V}_s and h_s . The estimates of the penstock flow rate look promising for both estimators.

Similarly, as for the case with measuring the flow rate, it is of interest to see the impact of incorrectly assumed disturbance covariance on the UKF and EnKF; the estimators have been run with the same pressure measurement as in the previous simulation, but with too large assumed disturbance covariance, see Table 3. The comparison of the estimators with too large assumed disturbance covariance (UKF/EnKF) and with correct disturbance covariance (UKF1/EnKF1) are shown in Fig. 9 for the UKF and Fig. 10 for the EnKF.

From Figs. 9 and 10, it is seen that too large assumed disturbance covariance causes additional fluctuations in the estimates for both the UKF and the EnKF. However, this too large disturbance covariance assumption has less detrimental effect on the EnKF than on the UKF.

Next, the estimates with the UKF and EnKF with correctly assumed measurement noise and disturbance covariance are compared with the synthetic model simulation for the hydropower system. The results of this comparison are shown in Fig. 11. This figure shows that the EnKF estimates deviate less from the model simulation than the UKF estimates. The estimates of the penstock flow rate is good for both estimators.

5.3.2. Measuring the manifold pressure

Next, the use of the manifold node pressure as the measurement is studied as an alternative to measuring the inlet turbine pressure. The comparison of estimates from two KF algorithms (UKF and EnKF) is done first for the hydropower system without random disturbance

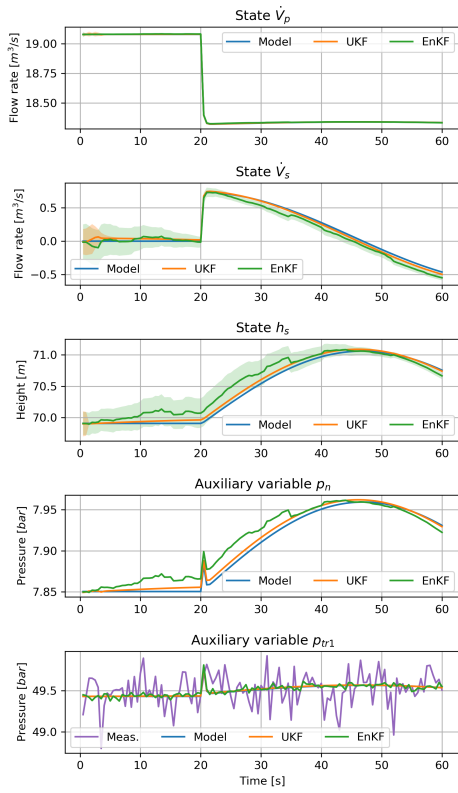


Figure 8: State and auxiliary variable comparisons for the UKF and EnKF with measurement noise and without random disturbance + correctly assumed disturbance covariance in estimators. Inlet turbine pressure is measured.

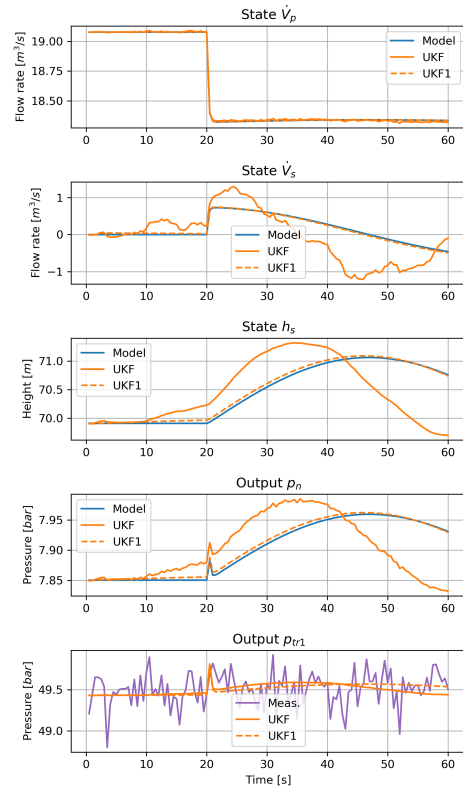


Figure 9: State and auxiliary variable comparisons for the UKF with too large disturbance covariance and UKF1 with correct disturbance covariance. Inlet turbine pressure is measured.

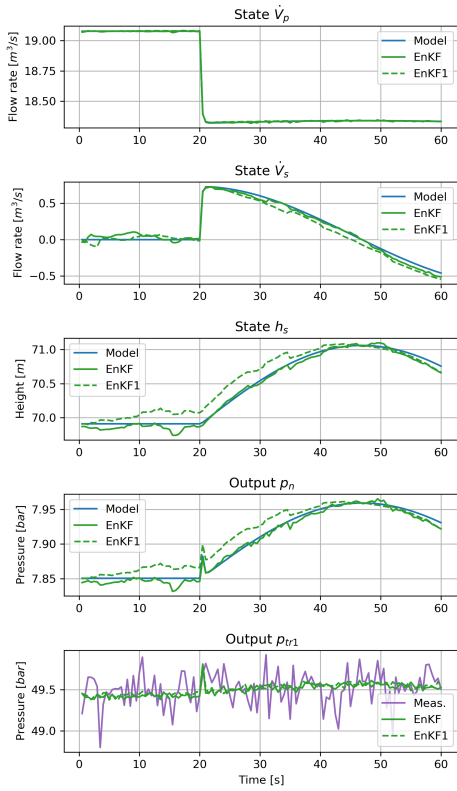


Figure 10: State and auxiliary variable comparisons for the EnKF with too large disturbance covariance and EnKF1 with correct disturbance covariance. Inlet turbine pressure is measured.

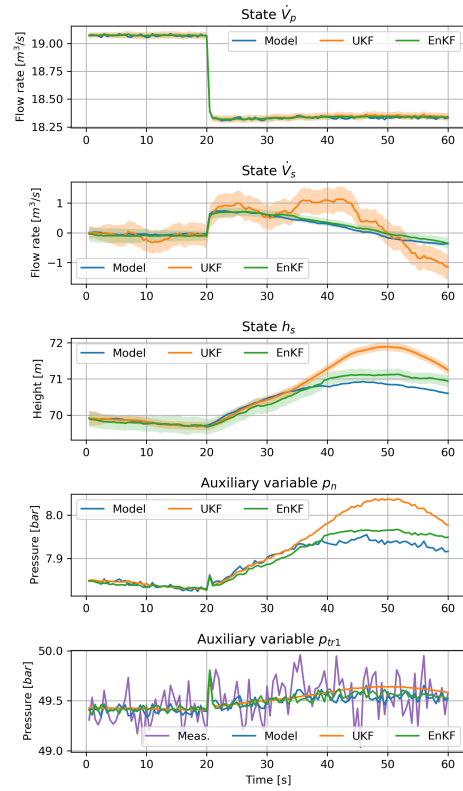


Figure 11: State and auxiliary variable comparisons for the UKF and EnKF with correctly assumed covariances. Inlet turbine pressure is measured.

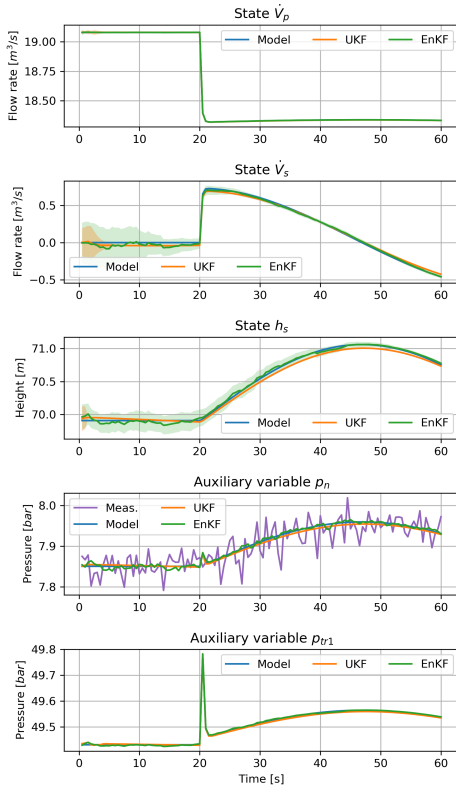


Figure 12: State and auxiliary variable comparisons for the UKF and EnKF without random disturbance in the synthetic data, and with correct assumption of disturbance covariance in the estimators. Manifold pressure is measured.

and with correct assumption of the disturbance covariance. The results are shown in Fig. 12; for more detailed simulation description see Table 3.

The UKF and EnKF estimates show low deviation from the model simulations. Some variation between the EnKF and UKF results is observed for the water height estimates of the surge tank, where the EnKF is closer to the model simulations. In the next comparisons, a too large covariance is assumed in the estimators, see Fig. 13 and Fig. 14. Similarly to the previous simulations, assuming too large disturbance covariance in the KFs leads to more deviation from the synthetic model simulation and more fluctuating estimates for both UKF and EnKF. In addition, the EnKF shows better results for the state estimates of the hydropower

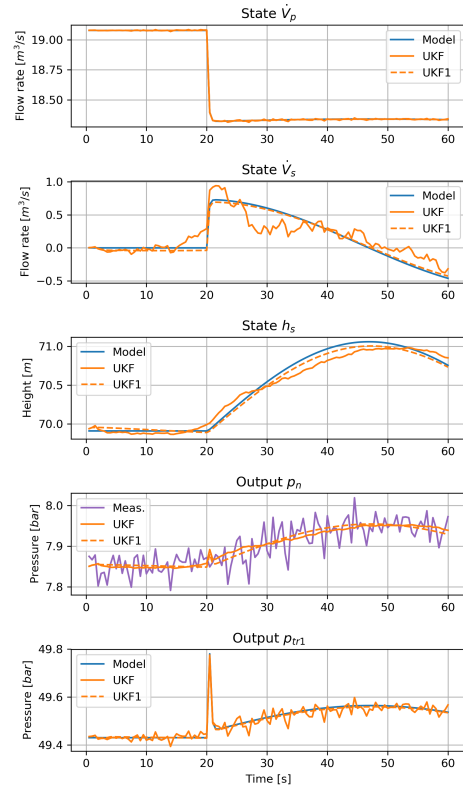


Figure 13: State and auxiliary variable comparisons for the UKF with too large disturbance covariance and UKF1 with correct disturbance covariance. Manifold pressure is measured.

system with random disturbance, see Fig. 15. This is similar to the case with measuring the inlet turbine pressure.

5.4. Measuring both outputs

In addition to the presented cases with only one measured output, it is of interest to see if the use of two measurements can improve the performance of the estimators. Thus, both the turbine flow rate and the inlet turbine pressure are considered as measurements. The same set of dynamic simulations that have been done for previous cases are also performed for this case. A more detailed description of the simulations is found in Table 3. The results of this case with two measured outputs, hydropower model without random disturbance as

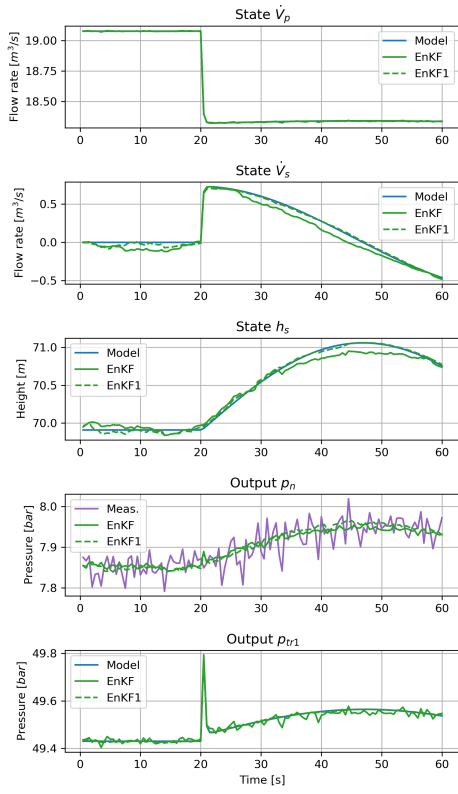


Figure 14: State and auxiliary variable comparisons for the EnKF with too large disturbance covariance and EnKF1 with correct disturbance covariance. Manifold pressure is measured.

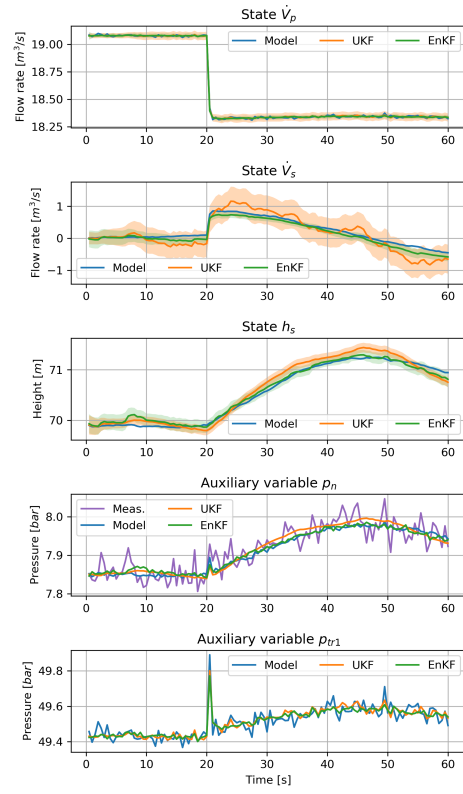


Figure 15: State and auxiliary variable comparisons for the UKF and EnKF with measurement noise and random disturbance (both estimator and model). Manifold pressure is measured.

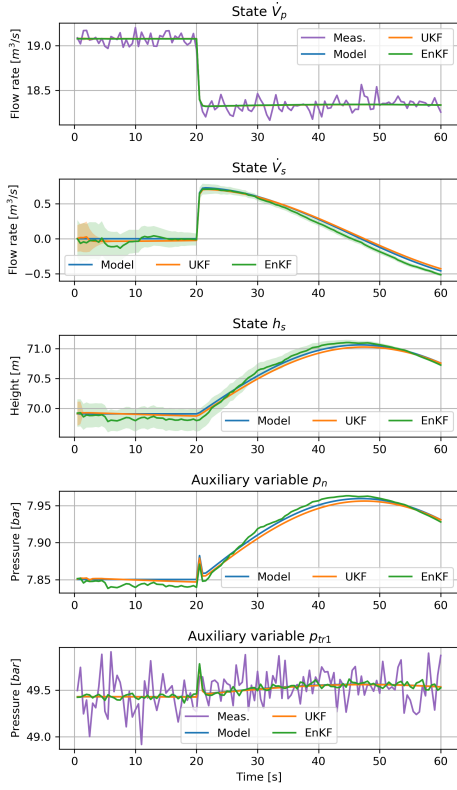


Figure 16: State and auxiliary variable comparisons for the UKF and EnKF with measurement noise, without random disturbance, and with correctly assumed disturbance covariance in the estimators. Both pressure and flow rate are measured.

well as correctly assumed covariance matrices in the estimators are given in Fig. 16.

It is seen from the figure that both the UKF and EnKF estimators provide good estimates and improve their results in comparison to the cases with only one measurement.

Then, in the same way as has been done in the previous cases, the impact of the assumed disturbance covariance on the UKF and EnKF is considered. The estimators have been run with the same two measurements, and with too large assumed disturbance covariance, see Table 3. The comparison of the estimators with too large assumed covariance (UKF, EnKF) and with correctly assumed disturbance (UKF1, EnKF1), respectively, are shown in Fig. 17 for the UKF and Fig. 18 for the EnKF.

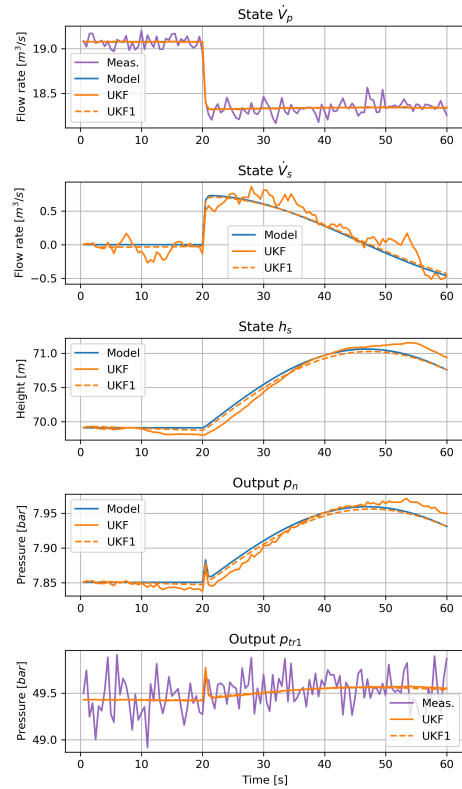


Figure 17: State and auxiliary variable comparisons for the UKF with too large disturbance covariance and UKF1 with correct disturbance covariance. Both pressure and flow rate are measured.

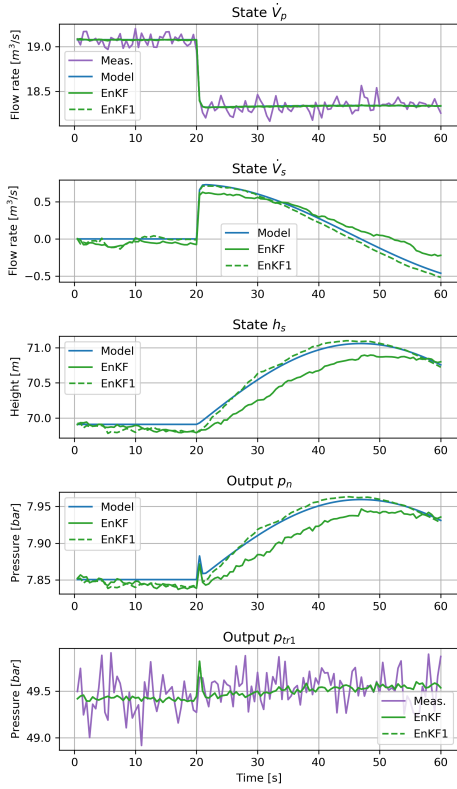


Figure 18: State and auxiliary variable comparisons for the EnKF with too large disturbance covariance and EnKF1 with correct disturbance covariance. Both pressure and flow rate are measured.

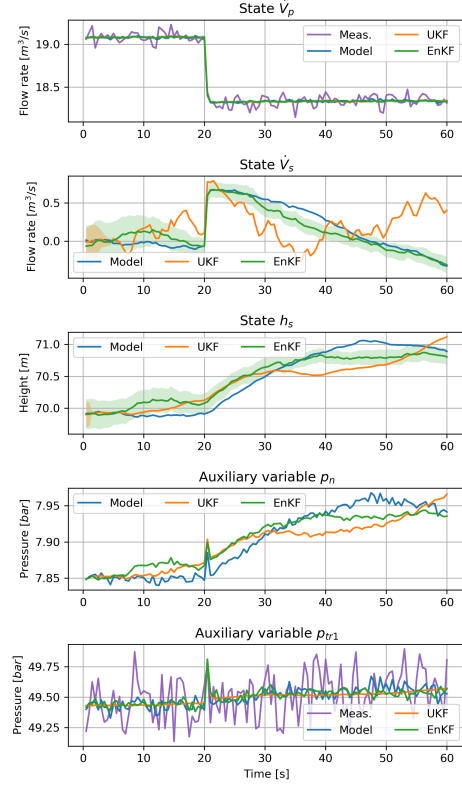


Figure 19: State and auxiliary variable comparisons for the UKF and EnKF with measurement noise and random disturbance, and with correctly assumed disturbance covariance in the estimators. Both pressure and flow rate are measured.

Both figures show that the estimators reduce their performance when assuming too large disturbance covariance: the state estimates become less accurate and noisier.

Finally, the estimates of the UKF and EnKF with correct assumptions of covariance matrices, and synthetic measurements/variables with random disturbance are compared. The results of this comparison are shown in Fig. 19. It is seen from this figure that the EnKF estimates deviate less from the model simulation than the UKF does.

5.5. Measurements from a more detailed model

It is of interest to see how the estimators behave when the synthetic measurements are based on a more de-

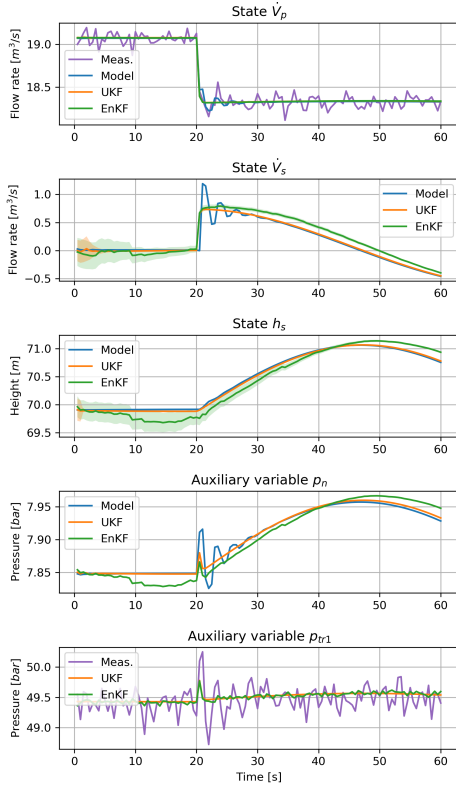


Figure 20: State and auxiliary variable comparisons for the UKF and EnKF with measurement noise and without random disturbance in the detailed simulations/synthetic measurements. The estimators use correct information about covariance matrices. Both pressure and flow rate are measured.

tailed/realistic model while the simple model is used for the estimators. The detailed model used for the synthetic measurements is similar to the model presented above, but includes water compressibility and pipe elasticity in the penstock, see Vytvytskyi and Lie (2017) for more information about this model.

Similarly to the final case above, both the turbine flow rate and the inlet turbine pressure are considered as measurements. No random disturbance is used in detailed model simulation, and random measurement noise is added to create the synthetic measurements. The estimators use correct assumptions about the covariance matrices. The results of state estimation for this case are given in Fig. 20.

It is seen from the figure that the UKF provides a bit better estimates than the EnKF for two states. However, the estimators' results do not vary a lot from the case where the measurements have been provided from the simple hydropower model.

6. Discussion and conclusions

Based on the simulations, the observations of the EnKF and the UKF estimation results are structured in a table for better comparison and analysis, see Table 3.

Regarding the presented estimation results from both the UKF and EnKF, it is clearly observed that the UKF and EnKF estimates show good results for the penstock flow rate. Specifically, the results are promising for the cases where the flow rate is not measured. The estimates from the UKF and EnKF for two other states (the surge tank flow rate and water height) deviate more from the model simulations than the estimates for the penstock flow rate; the results for these other states are still useful, though.

In general, both the UKF and the EnKF appear to give good results when the estimators use correct information about covariance matrices — whether there is random disturbance or not. In real life, the “correct” information about covariance matrices is not known.

When the estimator assumes a larger disturbance covariance than the real system has, the estimates become noisy. This is to be expected: assuming too large disturbance covariance is akin to assuming too low measurement noise covariance, which will lead to too much trust in the measurements, hence a too large Kalman gain, and “bleeding” of measurement noise into the estimates. It is interesting to observe that the UKF appears to give poorer estimates than the EnKF for this case. A possible explanation for this is that since the EnKF uses more particles than the number of sigma points in the UKF, the EnKF tends to smooth out this noise in a superior way as to what the UKF does.

Both the UKF and the EnKF give good results with a single pressure measurement (inlet turbine pressure or manifold node pressure); the UKF works fine with either pressure measurement, while the EnKF works best with the manifold node pressure measurement.

A case with two measurements, the turbine flow rate and the inlet turbine pressure, has also been considered. Combining two measurements leads to good performance for both estimators, and an improvement over using a single measurement. **However, two measurements at different locations may be more informative, and should be also considered for a future study.**

Table 3: The EnKF and UKF estimation results overview.

#	Available measurements from model	Fig.	Reality		Estimator		UKF results	EnKF result
			Random disturbance	Measurement noise	Random disturbance	Measurement noise		
1	Turbine flow rate	3	—	$V_k = 0.004$	—	$V_k = 0.004$	UKF a bit better for two states (\hat{V}_s, h_s). \hat{V}_p good for both.	
2	Turbine flow rate. Steady state	4	—	$V_k = 0.004$	—	$V_k = 0.004$	UKF and EnKF converge to the correct steady state value. EnKF converges faster.	
3	Turbine flow rate	5,6	—	$V_k = 0.004$	$W_k = 10^{-4}$	$V_k = 0.004$	Become much poorer than #1. Plus more fluctuations.	Become a bit poorer than #1. Plus more fluctuations.
4	Turbine flow rate	7	$W_k = 10^{-4}$	$V_k = 0.004$	$W_k = 10^{-4}$	$V_k = 0.004$	EnKF better than UKF for two states (\hat{V}_s, h_s). \hat{V}_p good for both.	
5	Inlet turbine pressure	8	—	$V_k = 0.03$	—	$V_k = 0.03$	UKF better for h_s and almost the same for \hat{V}_s . \hat{V}_p good for both.	
6	Inlet turbine pressure	9,10	—	$V_k = 0.03$	$W_k = 10^{-4}$	$V_k = 0.03$	Become much poorer than #4. Plus more fluctuations.	Become a bit better than #4. But more fluctuations.
7	Inlet turbine pressure	11	$W_k = 10^{-4}$	$V_k = 0.03$	$W_k = 10^{-4}$	$V_k = 0.03$	EnKF better than UKF for two states (\hat{V}_s, h_s). \hat{V}_p good for both.	
8	Manifold pressure	12	—	$V_k = 0.0007$	—	$V_k = 0.0007$	EnKF a bit better for h_s and the same for \hat{V}_s . \hat{V}_p good for both.	
9	Manifold pressure	13,14	—	$V_k = 0.0007$	$W_k = 10^{-4}$	$V_k = 0.0007$	Become poorer than #7. Plus more fluctuations.	Become poorer than #7. Plus more fluctuations.
10	Manifold pressure	15	$W_k = 10^{-4}$	$V_k = 0.0007$	$W_k = 10^{-4}$	$V_k = 0.0007$	EnKF better than UKF for two states (\hat{V}_s, h_s). \hat{V}_p good for both.	
11	Turbine flow rate and inlet pressure	16	—	$V_k = 0.004$ $V_k = 0.03$	—	$V_k = 0.004$ $V_k = 0.03$	UKF tiny better for two states (\hat{V}_s, h_s). \hat{V}_p good for both	
12	Turbine flow rate and inlet pressure	17,18	—	$V_k = 0.004$ $V_k = 0.03$	$W_k = 10^{-4}$	$V_k = 0.004$ $V_k = 0.03$	Become poorer than #11. Plus more fluctuations	Become poorer than #11. Plus more fluctuations.
13	Turbine flow rate and inlet pressure	19	$W_k = 10^{-4}$	$V_k = 0.004$ $V_k = 0.03$	$W_k = 10^{-4}$	$V_k = 0.004$ $V_k = 0.03$	EnKF better but still not really close to model for two states (\hat{V}_s, h_s). \hat{V}_p good for both.	
14	Turbine flow rate and inlet pressure. More detailed model	20	—	$V_k = 0.004$ $V_k = 0.03$	—	$V_k = 0.004$ $V_k = 0.03$	EnKF and UKF show good results. Estimates do not vary a lot from #11	

In almost all cases, the state covariance (uncertainty) converges faster for the UKF than the EnKF.

Regarding the choice of estimator algorithm, it is hard to make a final decision. The UKF has an advantage wrt. computational speed, but both the UKF and the EnKF are straightforward to parallelize, and with modern multicore/multi threading processors, this advantage is perhaps not vital. The EnKF appears to handle incorrect assumption about covariances better; this is also an important feature.

To summarize, it has been shown in this paper that state estimation based on the assimilation of a mechanistic model and measured data, might be used to improve the information for hydropower plants. Moreover, the hydropower mathematical model can be modeled simply by dragging, dropping, and connecting appropriate unit elements of the hydropower system using a visual modeling tool, e.g., our in-house hydropower Modelica library — *OpenHPL* in OpenModelica. For an OpenHPL based model, the model can be operated on in Python, e.g., doing state estimation via the Python API for OpenModelica.

References

- Efron, B., 1982. The Jackknife, the Bootstrap, and Other Resampling Plans. SIAM.
- Evensen, G., 2009. The Ensemble Kalman Filter for Combined State and Parameter Estimation. *IEEE Control Systems* 29, 83 – 104. doi:10.1109/MCS.2009.932223.
- Farrell, J.A., Polycarpou, M.M., 2006. Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches (Adaptive and Learning Systems for Signal Processing, Communications and Control Series). Wiley-Interscience, New York, NY, USA.
- Julier, S., Uhlmann, J., 1997. A new extension of the kalman filter to nonlinear systems, in: Proc. of AeroSense: The 11th Int. Symp. on Aerospace/Defense Sensing, Simulations and Controls, pp. 3068 – 3068. doi:10.1117/12.280797.
- Julier, S.J., Uhlmann, J.K., 2004. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE* 92, 401–422. doi:10.1109/JPROC.2003.823141.
- Lie, B., Bajracharya, S., Mengist, A., Buffoni, L., Kumar, A., Sjölund, M., Asghar, A., Pop, A., Fritzon, P., 2016. API for Accessing OpenModelica Models From Python, in: Proceedings of the EuroSim 2016, Oulu, Finland.
- Murphy, K., 2012. Machine Learning: A Probabilistic Perspective. Adaptive computation and machine learning. MIT Press.
- Sacchi, R., Ozturk, M.C., Principe, J.C., Carneiro, A.A.F.M., da Silva, I.N., 2007. Water Inflow Forecasting using the Echo State Network: a Brazilian Case Study, in: Proceedings of the International Joint Conference on Neural Networks, pp. 2403–2408. doi:10.1109/IJCNN.2007.4371334.
- Shi, L., Zeng, L., Zhang, D., Yang, J., 2012. Multiscale-finite-element-based ensemble Kalman filter for large-scale groundwater flow. *Journal of Hydrology* 468–469, 22 – 34. doi:10.1016/j.jhydro.2012.08.003.
- Simon, D., 2006. Optimal State Estimation: Kalman, Hoo, and Nonlinear Approaches. John Wiley & Sons.
- Splavska, V., Vytvytskyi, L., Lie, B., 2017. Hydropower Systems: Comparison of Mechanistic and Table Look-up Turbine Models, in: Proceedings of the 58th Conference on Simulation and Modelling (SIMS 58) Reykjavik, Iceland, September 25th–27th, 2017, Linköping University Electronic Press, Linköpings Universitet. pp. 368–373. doi:10.3384/ecp17138368.
- Stokelj, T., Golob, R., 2000. Application of neural networks for hydro power plant water inflow forecasting, in: Proceedings of the 5th Seminar on Neural Network Applications in Electrical Engineering. NEUREL 2000 (IEEE Cat. No.00EX287), pp. 189–193. doi:10.1109/NEUREL.2000.902410.
- Vytvytskyi, L., Lie, B., 2017. Comparison of elastic vs. inelastic penstock model using OpenModelica, in: Proceedings of the 58th Conference on Simulation and Modelling (SIMS 58) Reykjavik, Iceland, September 25th–27th, 2017, Linköping University Electronic Press, Linköpings Universitet. pp. 20–28. doi:10.3384/ecp1713820.
- Vytvytskyi, L., Lie, B., 2018. Mechanistic model for Francis turbines in OpenModelica. *IFAC-PapersOnLine* 51, 103 – 108. doi:10.1016/j.ifacol.2018.03.018.
- Zhou, W., Glemmestad, B., 2012. Implementation of unscented Kalman filter for nonlinear state estimation in hydropower plant. *IEEE International Conference on Power System Technology (POWERCON)*, 1 – 5doi:10.1109/PowerCon.2012.6401460.
- Zou, L., Zhan, C., Xia, J., Wang, T., Gippel, C., 2017. Implementation of evapotranspiration data assimilation with catchment scale distributed hydrological model via an ensemble Kalman Filter. *Journal of Hydrology* 546, 685 – 702. doi:10.1016/j.jhydro.2017.04.036.

- Improved information about and operation of the hydropower system by measurements assimilation with a mathematical model of a hydropower plant.
- Investigation of the possibility of state estimation for a hydropower system using Unscented and Ensemble Kalman filters.
- Implementation of the state estimators (Unscented and Ensemble Kalman filters) in Python using a hydropower model in OpenModelica via Python API.

ACCEPTED MANUSCRIPT

Paper E

Nonlinear observer for hydropower system

Liubomyr Vytvytskyi, Roshan Sharma, & Bernt Lie

Accepted in *Modeling, Identification and Control*.



Nonlinear observer for hydropower system

Liubomyr Vytvytskyi Roshan Sharma Bernt Lie

Department of Electrical engineering, Information Technology and Cybernetics, University of South-Eastern Norway, Porsgrunn, Norway. E-mail: {Liubomyr.Vytvytskyi,Roshan.Sharma,Bernt.Lie}@usn.no

Abstract

Estimation of unmeasured states plays an essential role in the design of control systems as well as for monitoring of hydropower plants. The standard Kalman filter gives the optimum state estimates for linear systems. However, this optimality is not relevant for nonlinear models and a choice between stochastic and deterministic approaches is not so obvious in this case. Thus the application of a nonlinear observer in a hydropower system is of interest here as an alternative to the widely used extended Kalman filter.

This paper provides a study and design of a reduced order nonlinear observer to estimate the states of a hydropower system. Implementation of the nonlinear observer is done in OpenModelica and added to our in-house hydropower Modelica library — *OpenHPL*, where different models for hydropower systems are assembled. Simulations and analysis of the designed observer are done in Python using a Python API for operating OpenModelica simulations.

Keywords: State estimation, reduced order nonlinear observer, hydropower model

1 Introduction

1.1 Background

A transition towards more renewable energy sources is currently happening in Europe and all over the world. This situation leads to increased use of flexible hydropower plants to compensate for the highly changing production from intermittent energy sources such as wind and solar irradiation. For this reason, development of advanced control structures or optimization of existing controllers for hydropower plants become a key task.

Full state information of the model of a hydropower system is needed to design an advanced controller such as model predictive control (MPC) or other model based controllers. However, some of the states can not be directly measured. Instead, a combination of a mathematical model and available measurements can be used to estimate the unmeasured states in hydropower plants.

Popular state estimation methods include the Kalman filter (KF) with a wide range of extensions that

apply depending on model structure, computational effort, etc. The Kalman filter is based on a stochastic approach. Alternatively, an observer based on a deterministic approach can be used for state estimation. This alternative is of particular interest for nonlinear systems such as the hydropower one, because the standard Kalman filter provides an optimum solution only for linear system, [Anderson and Moore \(1979\)](#). In addition, a nonlinear observer can lead to significant reduction of computational effort compared to a Kalman filter. Such observers can quickly find an estimate of the states and provide these as an input to fast nonlinear control algorithms. On the other hand, the proof of convergence is non-trivial for nonlinear observers.

1.2 Previous Work

A basic introduction to a variety of state estimation techniques based on the stochastic approach for general linear and nonlinear systems is provided in [Simon \(2006\)](#). As an alternative, an observer based on a deterministic approach can be used for the state estimation

and was first proposed by Luenberger (1964) for linear systems. Extension of the linear observer design for nonlinear systems has been presented in Krener and Isidori (1983). Although a huge variety of techniques for designing nonlinear observer exist, Kravaris et al. (2007); El-Farra et al. (2005); Andrieu and Praly (2006), the design of nonlinear observers is still a challenging task.

Some work on modeling the waterway for a high head hydropower system together with a generator, the Francis turbine, and a governor, using OpenModelica¹, is given in Vytvytskyi and Lie (2017, 2018b). Unit models have been assembled in our in-house Modelica² library *OpenHPL*.

A Python API³ for OpenModelica already exists and provides the possibilities for performing simulations of OpenModelica models via Python⁴, Lie et al. (2016). Python in turn gives much wider possibilities for plotting, analysis, and optimization than what is possible in OpenModelica Vytvytskyi and Lie (2018a).

1.3 Overview of Paper

In this paper, the main contribution is the development and simulation study of a reduced order nonlinear observer for state estimation in a hydropower system. Implementation of the observer is done in OpenModelica using *OpenHPL*. Simulation and analysis of the designed observer is done in Python using the Python API for OpenModelica.

The paper is structured as follows: Section 2 gives a system description of a high head hydropower plant. Section 3 gives an overview of the hydropower model. Design and proof of convergence for the nonlinear reduced order observer for the hydropower system is provided in Section 4. Simulation results from a number of case studies are presented in Sections 5 and 6. Finally, discussion and conclusions are given in Section 7.

2 System description

High head plants typically collect and store water in reservoirs in mountains, with tunnels leading the relatively small flow of water down a considerable height difference to the aggregated turbine and generator. The electricity produced by the generator is then transferred through power lines to consumers. A typical structure for a high head hydropower plant is depicted in Fig. 1, Vytvytskyi and Lie (2017).

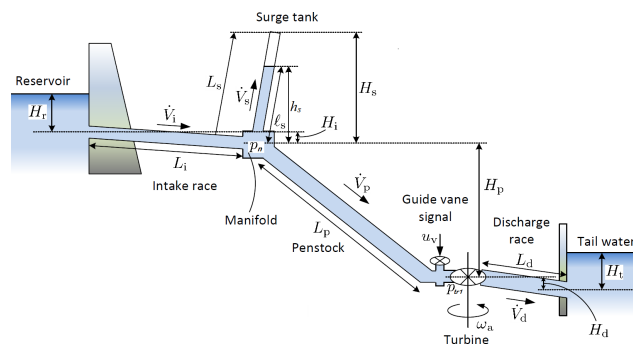


Figure 1: Overview of the structure of the high head hydropower plant.

For simulations in this paper, data from the Sundsbarm hydropower plant in Telemark, Norway is used with data provided in Vytvytskyi and Lie (2017).

3 Model

3.1 Model overview

For modeling the hydropower system, Modelica library OpenHPL is used. This is an in-house hydropower library, where different parts of the waterway components such as reservoir, conduit, surge tank, and turbine, have been assembled. In this library, different waterway components of the hydropower system are described by both mass and momentum balance, and include compressible/incompressible water or elastic/inelastic pipe walls. An overview of the mathematical models and methods used in this library is given in Vytvytskyi and Lie (2017); Splavska et al. (2017).

In this study, a simple hydropower model for the waterway is considered with the following assumptions for simplification:

- constant water level in reservoirs,
- power generation simplifications (constant speed of turbine),
- incompressible water and inelastic pipe.

All these simplifications lead to a simple hydropower model that consists of only three states Vytvytskyi and Lie (2018a).

3.2 Model presentation

In Modelica, models are described as differential algebraic equations (DAEs), with differential and algebraic variables. OpenModelica by default transforms the DAEs into state space form with auxiliary variables: states are typically a subset of the differential

¹<https://openmodelica.org>

²<https://www.modelica.org>

³<https://goo.gl/Qyjqq2>

⁴<https://www.python.org>

variables, while the auxiliary variables are the remaining variables. For the design of nonlinear observers, the model of the hydropower plant is represented by ordinary differential equations (ODE). The states of the model are given in vector x as follows:

$$x = \begin{bmatrix} \dot{V}_p \\ \dot{V}_s \\ h_s \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (1)$$

Here, \dot{V}_p is the volumetric flow rate through the penstock (equal to the flow rate in the discharge), \dot{V}_s is the volumetric flow rate in the surge tank, and h_s is the level in the surge tank, see Fig. 1. The sum of the flow rates in the penstock and the surge tank defines the flow rate in the intake: $\dot{V}_i = x_1 + x_2$. The states are described by differential equations as follows:

$$\frac{dx_1}{dt} = \frac{Z_p}{\rho} (p_n - p_{tr1}) + A_p g \frac{H_p}{L_p} - \frac{1}{8} \pi f_D B_p x_1 |x_1| \quad (2)$$

$$\frac{dx_2}{dt} = \frac{Z_s}{\rho} (p_n - p^{\text{atm}}) - A_s g \frac{H_s}{L_s} - \frac{1}{8} \pi f_D B_s x_2 |x_2| \quad (3)$$

$$\frac{dx_3}{dt} = \frac{x_2 H_s}{A_s L_s} \quad (4)$$

Here, p_n is the manifold node pressure, p_{tr1} is the turbine inlet pressure, p^{atm} is the atmospheric pressure. A_p and A_s are the cross section areas of the penstock and surge tank, respectively. H_p , H_s , and L_p , L_s are the height differences and the lengths of the penstock and surge tank, respectively. g is the gravitational acceleration and ρ is the water density. f_D is the Darcy friction factor and is assumed to be constant for simplicity. B_p , B_s , Z_p are geometrical parameters of the penstock and surge tank, and are presented in Eq. 5 and 6.

$$Z_s = \frac{A_s H_s}{x_3 L_s}, \quad Z_i = \frac{A_i}{L_i}, \quad Z_p = \frac{A_p}{L_p}, \quad Z_d = \frac{A_d}{L_d} \quad (5)$$

$$B_s = \frac{D_s}{A_s^2}, \quad B_i = \frac{D_i}{A_i^2}, \quad B_p = \frac{D_p}{A_p^2}, \quad B_d = \frac{D_d}{A_d^2} \quad (6)$$

Here, Z_p , Z_s , Z_d , Z_i are ratios of the cross section area to the length of the penstock, surge tank, discharge, and intake, respectively. B_p , B_s , B_d , B_i are

also ratios of the diameter to the squared cross section area of the penstock, surge tank, discharge, and intake, respectively.

The manifold and turbine inlet pressures are defined as follows:

$$p_{tr1} = K_z \left[Z_p p_n + \rho g \left(A_p \frac{H_p}{L_p} - A_d \frac{H_d}{L_d} \right) + Z_d \left(\frac{x_1^2 p^{\text{atm}}}{C_v^2 u_v^2} + p_t \right) + \frac{1}{8} \pi \rho x_1 |x_1| (f_{D,d} B_d - f_{D,p} B_p) \right] \quad (7)$$

$$p_n = K_p \left[Z_i p_r + Z_s p^{\text{atm}} + K_{pdz} \left(\frac{x_1^2 p^{\text{atm}}}{C_v^2 u_v^2} + p_t \right) + \rho g \left(A_p \frac{H_p}{L_p} (K_{pd} - 1) - K_{pd} A_d \frac{H_d}{L_d} + A_s \frac{H_s}{L_s} + A_i \frac{H_i}{L_i} \right) + \frac{1}{8} \pi \rho (f_{D,s} B_s x_2 |x_2| - f_{D,i} B_i (x_2 + x_1) |x_2 + x_1| + (f_{D,d} B_d K_{pd} + f_{D,p} B_p (1 - K_{pd})) x_1 |x_1|) \right] \quad (8)$$

Here, C_v is the turbine valve capacity and u_v is the turbine valve signal. p_r and p_t are the inlet penstock and the outlet discharge pressures, respectively. These pressures are defined from the atmospheric pressure and depths of the reservoir — H_r , and tail water — H_t as follows:

$$\begin{aligned} p_r &= p^{\text{atm}} + \rho g H_r \\ p_t &= p^{\text{atm}} + \rho g H_t \end{aligned} \quad (9)$$

Coefficients K_z , K_{pd} , K_{pdz} , and K_p are defined from the geometric ratios Z as follows:

$$\begin{aligned} K_z &= \frac{1}{Z_d + Z_p}, & K_{pdz} &= Z_p Z_d K_z \\ K_{pd} &= Z_p K_z, & K_p &= \frac{1}{Z_i + Z_s + K_{pdz}} \end{aligned} \quad (10)$$

This simple hydropower model is next used to design the reduced order nonlinear observer.

4 Observer design

Consider a general nonlinear system with state x , control signal u , output y and nonlinear functions $f(\cdot)$ and $g(\cdot)$ for the state and output, respectively:

$$\begin{aligned} \frac{dx}{dt} &= f(x, u) \\ y &= g(x, u) \end{aligned} \quad (11)$$

A standard presentation of any full order observer is given by Eq. 12,

$$\begin{aligned} \frac{d\hat{x}}{dt} &= f(\hat{x}, u) + L(y - \hat{y}) \\ \hat{y} &= g(\hat{x}, u) \end{aligned} \quad (12)$$

Here, \hat{x} and \hat{y} are the estimates of state x and output y , respectively. L is the observer gain and could be a nonlinear function, $L = L(x, u)$. The observation error dynamic, \tilde{x} , is described as:

$$\frac{d\tilde{x}}{dt} = \frac{dx}{dt} - \frac{d\hat{x}}{dt} = f(x, u) - f(\hat{x}, u) - L(y - \hat{y}) \quad (13)$$

The observer gain should be chosen or designed such that $\frac{d\tilde{x}}{dt}$ is an asymptotically stable system. This design can be done using Lyapunov based nonlinear system stability analysis, [Smith \(1995\)](#), and is, in general, a challenging task.

In real hydropower plants, the states x_1 and x_3 are usually available as measurements. Thus, in this study these two states are assumed to be known. The unknown state is x_2 . A reduced order observer for estimating the unmeasured state x_2 is designed for the hydropower system. The following variable transformation is chosen for observing x_2 :

$$\zeta_2 = x_2 + L_1 x_1 + L_3 x_3 \quad (14)$$

Here, L_1 and L_3 are the observer gains, also known as injection gains. Injection from both measured states x_1 and x_3 are used to design the nonlinear observer. However, the discussion on the estimation of x_2 by injecting only one of the measured states is provided in Section 5. From Eq. 14, the dynamics for ζ_2 is,

$$\frac{d\zeta_2}{dt} = \frac{dx_2}{dt} + L_1 \frac{dx_1}{dt} + L_3 \frac{dx_3}{dt} \quad (15)$$

The dynamics of the measured states x_1 and x_3 from Eq. 2 and 4 are inserted into Eq. 15.

$$\begin{aligned} \frac{d\zeta_2}{dt} &= \frac{Z_s}{\rho} (p_n - p^{\text{atm}}) - A_s g \frac{H_s}{L_s} \\ &\quad - \frac{1}{8} \pi f_{D,s} B_s x_2 |x_2| + L_3 \left(\frac{x_2 H_s}{A_s L_s} \right) \\ &\quad + L_1 \left(\frac{Z_p}{\rho} (p_n - p_{\text{tr1}}) \right. \\ &\quad \left. + A_p g \frac{H_p}{L_p} - \frac{1}{8} \pi f_{D,p} B_p x_1 |x_1| \right) \end{aligned} \quad (16)$$

Now, the observer for x_2 is,

$$\begin{aligned} \frac{d\hat{\zeta}_2}{dt} &= \frac{Z_s}{\rho} (p_n - p^{\text{atm}}) - A_s g \frac{H_s}{L_s} \\ &\quad - \frac{1}{8} \pi f_{D,s} B_s \hat{x}_2 |\hat{x}_2| + L_3 \left(\frac{\hat{x}_2 H_s}{A_s L_s} \right) \\ &\quad + L_1 \left(\frac{Z_p}{\rho} (p_n - p_{\text{tr1}}) \right. \\ &\quad \left. + A_p g \frac{H_p}{L_p} - \frac{1}{8} \pi f_{D,p} B_p x_1 |x_1| \right) \end{aligned} \quad (17)$$

The estimation error dynamic is defined as:

$$\frac{d\tilde{\zeta}_2}{dt} \equiv \frac{d\zeta_2}{dt} - \frac{d\hat{\zeta}_2}{dt} \quad (18)$$

After inserting $\frac{d\zeta_2}{dt}$ and $\frac{d\hat{\zeta}_2}{dt}$ from Eq. 16 and 17 into Eq. 18, the estimation error dynamic is,

$$\begin{aligned} \frac{d\tilde{\zeta}_2}{dt} &= \frac{1}{8} \pi f_{D,s} B_s [(Z_s + L_1 K_{p\text{dz}}) K_p - 1] \\ &\quad \times (x_2 |x_2| - \hat{x}_2 |\hat{x}_2|) + \frac{L_3 H_s}{A_s L_s} \tilde{\zeta}_2 \\ &\quad - \frac{1}{8} \pi (Z_s + L_1 K_{p\text{dz}}) K_p f_{D,i} B_i \\ &\quad \times ((x_2 + x_1) |x_2 + x_1| - (\hat{x}_2 + x_1) |\hat{x}_2 + x_1|) \end{aligned} \quad (19)$$

The estimation error dynamic should be analyzed for stability so that its state converges to zero in a finite amount of time. For this reason, Lyapunov stability analysis is used and a candidate Lyapunov function V is considered as follows:

$$V = \frac{1}{2} \tilde{\zeta}_2^2 \quad (20)$$

For the Lyapunov analysis, it must be shown that a derivative of the Lyapunov function is less than or equal to zero: $\frac{dV}{dt} \leq 0$. So, the derivative of this Lyapunov function is,

$$\frac{dV}{dt} = \tilde{\zeta}_2 \frac{d\tilde{\zeta}_2}{dt} \quad (21)$$

Inserting the observation error dynamic from Eq. 19, the derivative of the Lyapunov function $\frac{dV}{dt}$ becomes as follows:

$$\begin{aligned} \frac{dV}{dt} = & \frac{1}{8}\pi f_{D,s} B_s [(Z_s + L_1 K_{pdz}) K_p - 1] \\ & \times (x_2 |x_2| - \hat{x}_2 |\hat{x}_2|) \tilde{\zeta}_2 + \frac{L_3 H_s}{A_s L_s} \tilde{\zeta}_2^2 \\ & - \frac{1}{8}\pi (Z_s + L_1 K_{pdz}) K_p f_{D,i} B_i \\ & \times ((x_2 + x_1) |x_2 + x_1| - (\hat{x}_2 + x_1) |\hat{x}_2 + x_1|) \tilde{\zeta}_2 \quad (22) \end{aligned}$$

Equation 22 has three terms on its right hand side. Each of these terms will be analyzed separately and it will be shown that each term fulfills the conditions for Lyapunov stability. In the process of analyzing Eq. 22, the conditions for observer gains L_1 and L_3 will be calculated. From the first term on the right hand side of Eq. 22, let us suppose,

$$f_2 = (x_2 |x_2| - \hat{x}_2 |\hat{x}_2|) (x_2 - \hat{x}_2) \quad (23)$$

Then, the following conditions exist:

$$\begin{cases} \text{if } x_2 - \hat{x}_2 = \tilde{\zeta}_2 = 0, & \text{then } f_2 = 0 \\ \text{if } x_2 - \hat{x}_2 = \tilde{\zeta}_2 \neq 0, & \text{then } f_2 > 0 \end{cases} \quad (24)$$

Hence, $f_2 \geq 0$, i.e., f_2 is positive semidefinite. This implies that,

$$\frac{1}{8}\pi f_{D,s} B_s [(Z_s + L_1 K_{pdz}) K_p - 1] \leq 0 \quad (25)$$

From Equation 25, a condition for the observer gain L_1 can be defined as follows:

$$L_1 \geq \frac{Z_s}{K_{pdz}} - \frac{1}{K_{pdz} K_p} \quad (26)$$

Next, the second term on the right hand side of Eq. 22 can be analyzed. Here, it is obvious that $\tilde{\zeta}_2^2 \geq 0$. For this second term to be negative semi definite, the condition given by Eq. 27 should be fulfilled. Thus, the condition for the observer gain L_3 is,

$$\frac{L_3 H_s}{A_s L_s} \leq 0 \rightarrow L_3 \leq 0 \quad (27)$$

Finally, from the third term on the right hand side of Eq. 22, let us suppose,

$$\begin{aligned} f_{1,2} = & ((x_2 + x_1) |x_2 + x_1| \\ & - (\hat{x}_2 + x_1) |\hat{x}_2 + x_1|) \times (x_2 - \hat{x}_2) \quad (28) \end{aligned}$$

Knowing that $x_1 \geq 0$ (flow rate in the penstock is positive), it follows that,

$$\begin{cases} \text{if } x_2 - \hat{x}_2 = \tilde{\zeta}_2 = 0, & \text{then } f_{1,2} = 0 \\ \text{if } x_2 - \hat{x}_2 = \tilde{\zeta}_2 \neq 0, & \text{then } f_{1,2} > 0 \end{cases} \quad (29)$$

Hence, $f_{1,2} \geq 0$, i.e., $f_{1,2}$ is positive semidefinite. This implies that,

$$\frac{1}{8}\pi (Z_s + L_1 K_{pdz}) K_p f_{D,i} B_i \geq 0 \quad (30)$$

From here, another condition for the observer gain L_1 is found,

$$L_1 \leq \frac{Z_s}{K_{pdz}} \quad (31)$$

Thus from Eq. 26, 27 and 31, the required condition for the stable dynamics of the estimation error dynamic $\frac{d\tilde{\zeta}_2}{dt}$ is:

$$\frac{Z_s}{K_{pdz}} - \frac{1}{K_{pdz} K_p} \leq L_1 \leq \frac{Z_s}{K_{pdz}} \quad (32)$$

Hence, the designed reduced order nonlinear observer for estimating the volumetric flow rate in the surge tank (x_2) is written as,

$$\hat{x}_2 = \hat{\zeta}_2 - L_1 x_1 - L_3 x_3 \quad (33)$$

Here, $\hat{\zeta}_2$ is the estimate of the transformed coordinate for x_2 and its dynamics is defined using Eq. 17. L_1 and L_3 are the observer gains that are defined using conditions in Eq. 32. A summary of the developed reduced order nonlinear observer for the hydropower system is given in Table 1.

Thus, the observer convergence is proved here, and this in turn proves observability of the state x_2 . Then, the observer simulations are illustrated in the next section.

5 Results and discussions

5.1 Simulation setup

The reduced order nonlinear observer is implemented in OpenModelica, which is an open source Modelica based modeling and simulation tool designed for industrial and academic usage. Using the Python API for OpenModelica, simulations of the observer are carried out in Python.

Three cases are presented in this section for the dynamic simulations of the developed reduced order nonlinear observer:

Table 1: Summary of the reduced order nonlinear observer for the hydropower system.

Plant	$\begin{aligned} \frac{dx_1}{dt} &= \frac{Z_p}{\rho}(p_n - p_{tr1}) + A_p g \frac{H_p}{L_p} - \\ &\frac{1}{8}\pi f_D B_p x_1 x_1 \\ \frac{dx_2}{dt} &= \frac{Z_s}{\rho}(p_n - p^{atm}) - A_s g \frac{H_s}{L_s} - \\ &\frac{1}{8}\pi f_D B_s x_2 x_2 \\ \frac{dx_3}{dt} &= \frac{x_2 H_s}{A_s L_s} \end{aligned}$
Observer	$\begin{aligned} \hat{x}_2 &= \hat{\zeta}_2 - L_1 x_1 - L_3 x_3 \\ \frac{d\hat{\zeta}_2}{dt} &= \frac{Z_s}{\rho}(p_n - p^{atm}) - A_s g \frac{H_s}{L_s} - \\ &\frac{1}{8}\pi f_{D,s} B_s \hat{x}_2 \hat{x}_2 + L_3 \left(\frac{\hat{x}_2 H_s}{A_s L_s} \right) + \\ &L_1 \left(\frac{Z_p}{\rho}(p_n - p_{tr1}) + A_p g \frac{H_p}{L_p} - \right. \\ &\left. \frac{1}{8}\pi f_{D,p} B_p x_1 x_1 \right) \end{aligned}$
Design variables	$\begin{aligned} \frac{Z_s}{K_{pdz}} - \frac{1}{K_{pdz} K_p} &\leq L_1 \leq \frac{Z_s}{K_{pdz}} \\ L_3 &\leq 0 \end{aligned}$

- Only the penstock volumetric flow rate $\dot{V}_p \approx 19.07$ (x_1) is injected, i.e., $L_3 = 0$.
- Only the surge tank water height h_s (x_3) is injected, i.e., $L_1 = 0$.
- Both the penstock volumetric flow rate and the surge tank water height (x_1 and x_3) are injected.

Measurement data from a real hydropower plant are not available, instead outputs from hydropower model simulations are used. Two hydropower models are used to represent reality: (a) a detailed model, and (b) the simplified model as described in Section 3. Hence, for all cases below, the hydropower models are first simulated separately to get the appropriate synthetic measurements.

A set of dynamic simulations for the reduced order nonlinear observer is performed, where the simulations start from the steady state ($\dot{V}_p \approx 19.07$ m³/s and $h_s \approx 69.91$ m) and last for 60 s with a sample time of 0.5 s. A disturbance K_p occurs at time 10 s with a rapid closing of the turbine valve by 3%. First, an appropriate synthetic measurement data for the penstock flow rate and for the surge tank water height are created from the simulation of the simplified hydropower model. These measurements are shown in Fig. 2.

Moreover, the effect of measurement noise on the estimated state is also of interest. White measurement

noise, v_k , is added to the measurement signals and used for the observer simulations. For both measurements, the mean of the measurement noise v_k is zero, i.e., $\bar{v}_k = 0$, and the noise covariance V_k are the same for both measurements for simplicity and equals $V_k = 0.0001$, i.e., $v_k \sim \mathcal{N}(0, 0001)$. The noisy measurements are also shown in Fig. 2.

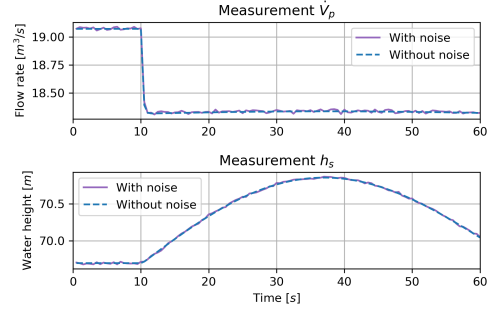
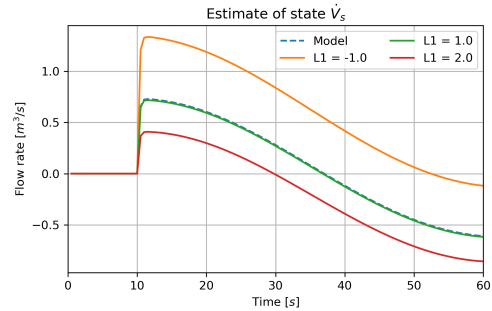


Figure 2: Measurements from the simplified hydropower model with and without noise for the dynamic simulations.

5.2 Using one measurement

First, the studies of the observer behavior that uses only one measurement (one state is injected) are performed, i.e., one of the observers' gains is set to zero. The results of the dynamic simulations without any measurement noises for these cases are shown in Fig. 3 for the observer that use only the penstock flow rate measurement ($L_3 = 0$) and in Fig. 5 for the observer that use only the surge tank water height measurement ($L_1 = 0$). In both figures, three values for the nonzero observer gain are used to study the observer behavior.


 Figure 3: Observer behavior when only measurement x_1 is injected. Gain L_1 is varied to study its effect on convergence.

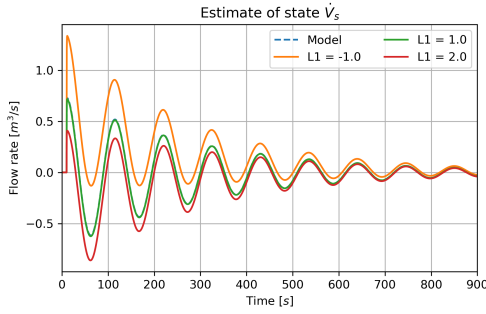


Figure 4: Observer behavior for longer simulation time when only measurement x_1 is injected.

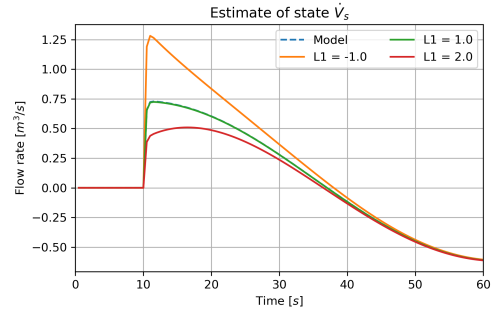


Figure 6: Observer behavior when both measurement x_1 and x_3 are injected. Gain L_1 is varied to study its effect on convergence.

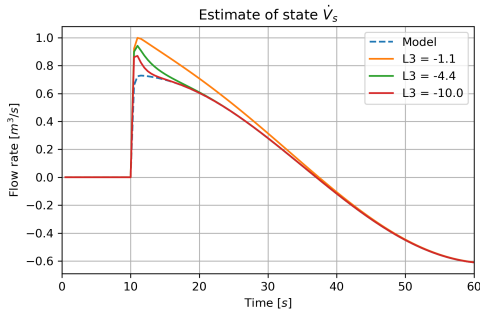


Figure 5: Observer behavior when only measurement x_3 is injected. Gain L_3 is varied to study its effect on convergence.

From Fig. 3, it is seen that the estimates from the observer that uses only the penstock flow rate measurement as injection converges properly for the gain value of $L_1 = 1.0$. With the values of observer gain as $L_1 = -1.0$ and $L_1 = 2.0$, significant overshooting and undershooting are seen during the transient period, respectively. With these values for the gain L_1 , the observer requires a relatively longer time for convergence, see Fig. 4 with extended simulation time of 900 s. Thus a proper choice of observer gain L_1 seems to be necessary for obtaining faster convergence.

Figure 5 shows that when only the surge tank water height measurement is used as injection, the observer shows some overshooting in the estimates when the disturbance occurs, and then converges to the correct value. Here, a lower value of the observer gain L_3 leads to faster convergence.

From both these cases, it can be concluded that it is possible to estimate the unmeasured state x_2 by using only one of the available measurements. However, it is also of interest to check the observer’s performance when both measurements are used together.

5.3 Using both measurements

The next set of simulations show the behavior of the observer that uses both the penstock flow rate and the surge tank water height as available measurements. First, dynamic simulations for the reduced order nonlinear observer without any measurement noises are presented in Fig. 6. Here, the observer gain L_1 is varied and another gain is set to a fixed value of $L_3 = -1.0$.

As compared to Fig. 3 (where only one measurement was injected), Fig. 6 shows that by using both the available measurements, the performance of the observer is significantly improved. With the same choices of gain L_1 (for both cases), it can be seen that the observer converges much faster when both measurements are used in the observer design. It is also noticed but not shown in Fig. 6 that with lower values for the observer gain L_3 , the estimates converges relatively faster to their correct values.

However, the observer with the fixed gain $L_1 = 1.0$ and varying gain L_3 produces very similar results for different choices of gain L_3 . This is shown in Fig. 7, where measurements without noises are used for dynamic simulation of the reduced order nonlinear observer. Here, different values are used for the observer gain L_3 and the other gain is set to a fixed value of $L_1 = 1.0$.

As compared to Fig. 5 (where only one measurement was used), Fig. 7 shows that the performance of the observer is vastly improved when both measurements are used. In this case, the overshoots are reduced significantly. It can be concluded that when both measurements are used in the observer design, and with proper choices of gains L_1 and L_3 , the performance of the observer can be vastly improved. However, it is also possible to estimate the unmeasured state by using only one measurement. With this, the proof of convergence of the observer design perhaps will also be simplified.

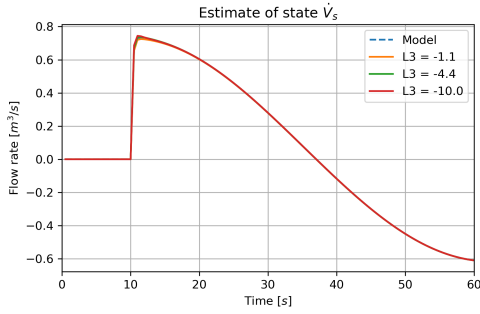


Figure 7: Observer behavior when both measurement x_1 and x_3 are injected. Gain L_3 is varied to study its effect on convergence.

In addition, in some cases it might be considered to use a simple observer for estimating x_2 (the surge tank flow rate) that is based only on Eq. 4 and use changes in x_3 (the surge tank water height) as measurements (through model inversion). This simple observer will be more sensitive to noise, but otherwise it would have many advantages, such as simplicity (no tuning parameters) and ease of implementation. However, this simple observer will work fine as long as the model is considered to be perfect. In reality, model parameters might not be exactly known and/or there might be other unknown input disturbances acting on the system. Under such conditions, open-loop observers may not function well and it might be necessary to introduce feedback by injecting the measurements. Furthermore in Fig. 7, it has been shown that by using measurement x_1 in addition to x_3 as the measurement, the estimation can be significantly improved.

5.4 Measurement noise influence

The performance of the observer is affected by the measurement noise and the choice of the observer gains. To illustrate this, the observer gain L_3 is varied while the gain L_1 is kept constant. The influence of the measurement noise on the estimate is shown in Fig. 8.

As the gain L_3 is lowered, the observer converges faster, however, this also makes the estimates more noisy. This justifies the fact that under the presence of measurement noises, the speed of convergence and the occurrence of noisy estimates should be balanced properly by proper choice of observer gains. To increase the performance of the observer with noisy measurements, the measurement data should probably be pre-processed and filtered before feeding it to the observer in practise.

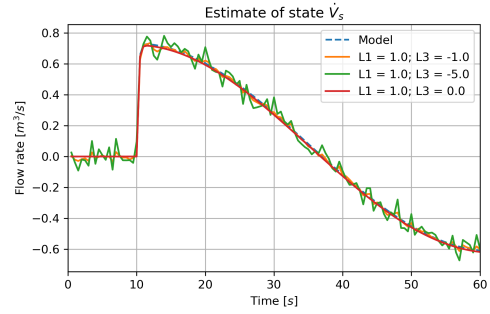


Figure 8: Effect of measurement noise on the state estimates with varying gain.

6 Testing observer with measurement data from a detailed model

It is of interest to see how the reduced order nonlinear observer behave when the synthetic measurements are based on a more detailed/realistic model while the simple model is used for the design of the observer. The detailed model used for the synthetic measurements is similar to the model presented above, but includes water compressibility and pipe elasticity in the penstock, see [Vytvytskyi and Lie \(2017\)](#) for more information about this model. Moreover, the Darcy friction factor varies with the flow rate for all pipe units in this detailed model.

First, appropriate synthetic measurements for the penstock flow rate and for the surge tank water height are created from the simulation of the more detailed hydropower model. These measurements are shown in Fig. 9 for the cases without and with measurement noise. Similarly to the previous case, the same level of measurement noise v_k is used here.

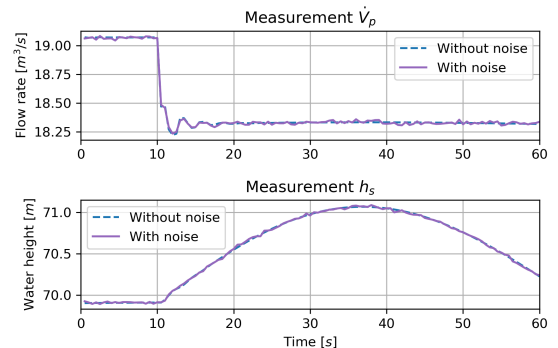


Figure 9: Measurements from the more detailed hydropower model.

The dynamic simulations for the reduced order nonlinear observer that use these measurements from the detailed model without considering measurement noise are shown in Fig. 10. Here, the results of the observer with re-tuned gains L_1 and L_3 that provides relatively good estimation is shown in the figure.

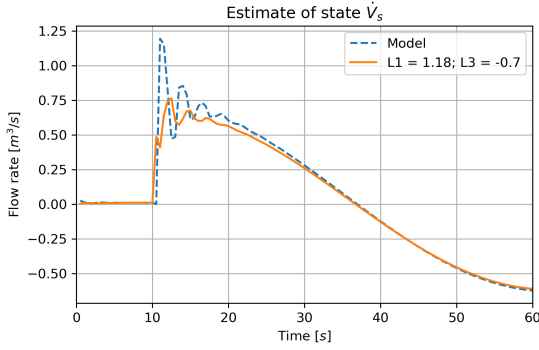


Figure 10: Observer behavior when both measurement x_1 and x_3 are injected. Measurements without noise from the more detailed hydropower model.

Figure 10 shows that fine tuning of the L_1 and L_3 observer gains reduce the offset between the estimates from the observer and the results from the detailed hydropower model.

Simulations with noisy measurements from the detailed hydropower model are performed to see the influence of the noise on the estimation. The results from the observer simulation are shown in Fig. 11, where it can be seen that the measurement noise affects the observer results, and the estimates from the observer become noisier. This behavior is expected and is similar to the results from the previous case study (Fig. 8). It is also clearly visible that observer produces relatively good estimates even under the presence of measurement noises from the detailed hydropower model.

It is worth mentioning that the observer gains had to be re-tuned when it is applied on a detail model. In both cases (with/without considering measurement noises), it can be seen that the observer converge to the true state values. However, the performance of the observer is relatively poor compared to the case where perfect model (no model-observer mismatch) is considered. Under the presence of measurement noises, the estimates also become slightly noisy as shown in Fig 11.

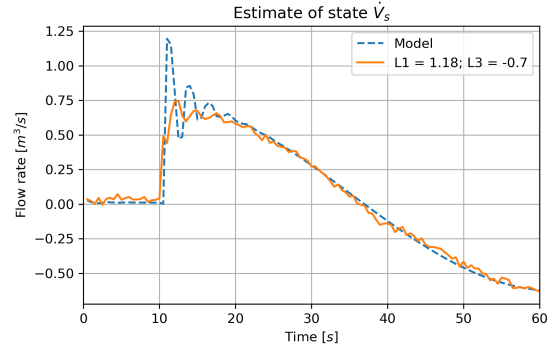


Figure 11: Observer behavior when both measurement x_1 and x_3 are injected. Measurements with noise from the more detailed hydropower model.

7 Conclusions

The design and proof of convergence for the reduced order nonlinear observer for the simplified hydropower system has been presented in this paper. This observer is designed to estimate the surge tank flow rate using the measurements from two other states (the penstock flow rate and the surge tank water height). The observer has been implemented in OpenModelica and added to our in-house hydropower library (*OpenHPL*). Then a number of simulations have been run in Python using the Python API for OpenModelica in order to study the designed observer.

The dynamic simulations have been performed to show the observer behavior under the presence of input disturbance. These simulations have been carried out for three cases, where the measured states have been injected separately first, and then both measured states have been used. Moreover, the influence of the measurement noise on the observer has also been presented for the dynamic simulations. The simulations showed that the performance of the reduced order nonlinear observer mostly depends on the observer gain L_1 for overshooting/undershooting of the estimates, and the gain L_3 for the convergence speed.

It has also been checked, but not shown in the paper, how the observer behaves if the designed conditions for the observer gains are broken, e.g., $L_3 > 0$ or $L_1 \leq \frac{Z_s}{K_{pdz}} - \frac{1}{K_{pdz}K_p}$ and $L_1 \geq \frac{Z_s}{K_{pdz}}$. In case with $L_3 > 0$, the system becomes unstable and the observer fails. For another case, with L_1 , the simulation runs without failures. However the results become poorer.

To summarize, it has been shown that a reduced order nonlinear observer can be designed and used for the state estimation in the hydropower system. The design of the observer is based on a simplified model with cer-

tain assumptions. This lowers the complexity in the design of the observer. When the observer is tested against the simplified model, the estimates are proper without any offsets. However, the observer is also tested against the data from a more detailed/complex hydropower model where simplifying assumptions are not considered. Under such condition, the observer has to be re-tuned. With a properly calibrated/tuned reduced order observer, the observer can still provide satisfactory estimates of the unmeasured state.

References

- Anderson, B. D. O. and Moore, J. B. *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, NJ, 1979.
- Andrieu, V. and Praly, L. On the Existence of a Kazantzis–Kravaris/Luenberger Observer. *SIAM J. Control and Optimization*, 2006. 45(2):432–456. doi:[10.1137/040617066](https://doi.org/10.1137/040617066).
- El-Farra, N., Mhaskar, P., and Christofides, P. D. Output feedback control of switched nonlinear systems using multiple Lyapunov functions. *Proceedings of the 2005, American Control Conference, 2005.*, 2005. 6:3792–3799.
- Kravaris, C., Sotiropoulos, V., Georgiou, C., Kazantzis, N., Xiao, M., and Krener, A. J. Nonlinear observer design for state and disturbance estimation. *Systems & Control Letters*, 2007. 56(11):730 – 735. doi:[10.1016/j.sysconle.2007.05.001](https://doi.org/10.1016/j.sysconle.2007.05.001).
- Krener, A. J. and Isidori, A. Linearization by output injection and nonlinear observers. *Systems & Control Letters*, 1983. 3(1):47 – 52. doi:[10.1016/0167-6911\(83\)90037-3](https://doi.org/10.1016/0167-6911(83)90037-3).
- Lie, B., Bajracharya, S., Mengist, A., Buffoni, L., Kumar, A., Sjölund, M., Asghar, A., Pop, A., and Fritzson, P. API for Accessing OpenModelica Models From Python. In *Proceedings of the EuroSim 2016, Oulu, Finland*. 2016.
- Luenberger, D. G. Observing the State of a Linear System. *IEEE Transactions on Military Electronics*, 1964. 8:74–80. doi:[10.1109/TME.1964.4323124](https://doi.org/10.1109/TME.1964.4323124).
- Simon, D. *Optimal State Estimation: Kalman, H_∞ , and Nonlinear Approaches*. John Wiley & Sons, 2006.
- Smith, M. C. The general problem of the stability of motion: Translated and Edited by A. T. Fuller. Taylor and Francis, 1992. *Automatica*, 1995. 31:353–354. doi:[10.1016/0005-1098\(95\)90022-5](https://doi.org/10.1016/0005-1098(95)90022-5).
- Splavska, V., Vytvytskyi, L., and Lie, B. Hydropower Systems: Comparison of Mechanistic and Table Look-up Turbine Models. In *Proceedings of the 58th Conference on Simulation and Modelling (SIMS 58) Reykjavik, Iceland, September 25th–27th, 2017*, 138. Linköping University Electronic Press, Linköpings Universitet, pages 368–373, 2017. doi:[10.3384/ecp17138368](https://doi.org/10.3384/ecp17138368).
- Vytvytskyi, L. and Lie, B. Comparison of elastic vs. inelastic penstock model using OpenModelica. In *Proceedings of the 58th Conference on Simulation and Modelling (SIMS 58) Reykjavik, Iceland, September 25th–27th, 2017*, 138. Linköping University Electronic Press, Linköpings Universitet, pages 20–28, 2017. doi:[10.3384/ecp1713820](https://doi.org/10.3384/ecp1713820).
- Vytvytskyi, L. and Lie, B. Linearization for Analysis of a Hydropower Model using Python API for OpenModelica. In *Proceedings of the 59th Conference on Simulation and Modelling (SIMS 59) Oslo, Norway, September 26th–28th, 2018*, 153. Linköping University Electronic Press, Linköpings Universitet, pages 216–221, 2018a. doi:[10.3384/ecp18153216](https://doi.org/10.3384/ecp18153216).
- Vytvytskyi, L. and Lie, B. Mechanistic model for Francis turbines in OpenModelica. *IFAC-PapersOnLine*, 2018b. 51(2):103 – 108. doi:[10.1016/j.ifacol.2018.03.018](https://doi.org/10.1016/j.ifacol.2018.03.018).

Paper F

Structural analysis in Julia for dynamic systems in OpenModelica

Liubomyr Vytvytskyi, & Bernt Lie

Accepted for the 60th Conference on Simulation and Modelling (SIMS60).

Structural analysis in Julia for dynamic systems in OpenModelica

Liubomyr Vytvytskyi Bernt Lie

Department of Electrical engineering, Information Technology and Cybernetics, University of South-Eastern Norway, Porsgrunn, Norway, {Liubomyr.Vytvytskyi,Bernt.Lie}@usn.no

Abstract

In control theory for dynamic systems, information about observability and controllability of states plays a key role to evaluate the possibility to observe states from outputs, and use inputs to move states to a desired position, respectively. Automatic determination of observability and controllability is possible, in particular for linear models where typically observability and controllability grammians are considered. In the case of large scale systems, e.g., complex models of regional energy systems, standard analysis becomes challenging. For large scale systems, structural analysis based on directed graphs is an interesting alternative: structural observability (or: controllability) is a necessary requirement for actual observability (or: controllability). Directed graphs can be set up directly for linear models, but can also be extracted from nonlinear models.

Modelica is a suitable language for describing large scale models, but does not support graph algorithms. One possibility is to integrate the Modelica model into a language supporting graph algorithms, e.g., Julia: this integration can be done using package OMJulia which works with the free tool OpenModelica. OMJulia does not give direct access to the nonlinear model in Modelica, but a linear model approximation can be extracted and used for setting up the system graph. In this study, an experimental implementation of automated structural analysis is done in Julia using the *LightGraphs.jl* package. As an example, this structural analysis is tested on hydropower models of different complexity that are modelled in OpenModelica using our in-house hydropower Modelica library — *OpenHPL*, where different models for hydropower systems are assembled.

Keywords: observability, controllability, structural analysis, graph theory

1 Introduction

1.1 Background

Modelling and simulation of dynamic systems (e.g., a hydropower system in this paper) plays an important role as efficient analysis tools for control analysis and design. As an example, tools for designing a new or testing an existing controller for stability and performance in different operating regimes might be of interest.

Model based analysis of state observability and control-

lability is important for control design, and it is of interest to consider tools for aiding such analysis. Classically, observability and controllability properties might be checked using the well known tests based on rank conditions, (Simon, 2006; Šiljak, 2011). However, numerical problems can arise for the rank computations in complex, large scale systems. Still, structural observability and controllability based on the system structure can be used in such cases due to the simplicity of these methods. In addition, a relative degree of the system indicates how directly control inputs affect outputs, and can also be defined based on the system structure. Assuming linear models, analysis tools based on graph theory can be implemented in Julia¹, e.g., using the *LightGraphs.jl* package.

Models of various dynamic systems might be directly modeled in Julia using the *DifferentialEquations.jl*² (Rackauckas and Nie, 2017) and *ControlSystems.jl*³ packages. However, an object-oriented language such as Modelica⁴ has richer support for describing complex, large scale systems with inputs and outputs. One such Modelica based tool is OpenModelica⁵ which offers an open-source modeling and simulation environment. OpenModelica also comes with the OMJulia.jl package which offers integration of Modelica models in Julia.

1.2 Previous Work

Basic graph theory for different engineering applications is provided in (Deo, 2017). Structural modeling and analysis of complex systems are described by (Šiljak, 2007, 2011; Lunze, 1992; Boyd and Vandenberghe, 2018). Based on this graph theory, large scale systems can be further tested and analyzed for control and parameter estimation purposes; see, e.g., (Perera, 2016) who used structural analysis of Modelica models in JModelica⁶ and Python⁷ to analyze an industrial copper electrowinning process.

The OMJulia package⁸ (Julia API) for OpenModelica provides possibilities to run simulations and carry out linearization of OpenModelica. Julia in turn gives rich possibilities for plotting, analysis, and optimization (e.g., using

¹<https://julialang.org/>

²<https://goo.gl/5wxZfR>

³<https://goo.gl/d2xyf2>

⁴<https://www.modelica.org>

⁵<https://openmodelica.org>

⁶<https://jmodelica.org/>

⁷<https://www.python.org>

⁸<https://goo.gl/WpAMds>

Julia packages *Plots.jl*, *LightGraph.jl*, *JuMP.jl*, etc.).

Some work on modeling a waterway for high head hydropower system together with a generator, a Francis turbine, and a governor, has already been carried out using OpenModelica (Vytvytskyi and Lie, 2017, 2018a). Unit models have been assembled in our in-house Modelica library *OpenHPL*⁹. Similarly to Julia, a Python API¹⁰ for OpenModelica already exists and a use of this API for the *OpenHPL* is presented in (Vytvytskyi and Lie, 2018b).

1.3 Overview of Paper

In this paper, the main contribution is the prototyping and testing of automated structural analysis for dynamic systems in Julia using directed graphs from the *LightGraphs.jl* package.

The paper is structured as follows: Section 2 gives an overview of the graph and structural analysis theory. The Julia implementation of these analysis methods is discussed in Section 3. Applying the structural analysis methods on hydropower models is presented in Section 4. Finally, discussion and conclusions are given in Section 5.

2 Structural analysis

Structural analysis of models is the evaluation of model behavior base on a model structure. In this study, the model structure is represented by graphs. That is why, the graph theory is described first.

2.1 Graph theory

A graph G connects nodes (vertices, points) $N = \{n_1, n_2, \dots, n_N\}$ via edges (lines) $E = \{e_1, e_2, \dots, e_E\}$. Here, we will consider a *directed graph* (digraph); a digraph may be defined by a relation R consisting of a set of *ordered pairs* (n_i, n_j) with unidirectional information flow between these nodes.

As examples, Fig. 1 shows the undirected graph G_1 (left) and the directed graph G_2 (right). Observe that each pair corresponds to an edge; G_1 has 5 edges because its relation R_1 holds 5 (unordered) pairs, while G_2 has 6 edges because R_2 holds 6 (ordered) pairs.

Instead of describing the graph via a relation, it can be described via either an *adjacency matrix* A or an *incidence matrix* I . The incidence matrix description I with $\dim I = n_E \times n_N$ relates edges and nodes. The incidence matrix is not suitable for describing self edges, and is not discussed further here.

The adjacency matrix relates unidirectional flow between two nodes, and is defined by $A_{i,j} = 1$ for $(i, j) \in R$, or $A_{i,j} = 0$ for $(i, j) \notin R$. The adjacency matrix is square, $\dim A = n_N \times n_N$, with nodes represented by both rows and columns. Adjacency matrix A_2 for G_2 in Fig. 1 is

$$A_2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \text{ Observe that nonzero diagonal elements}$$

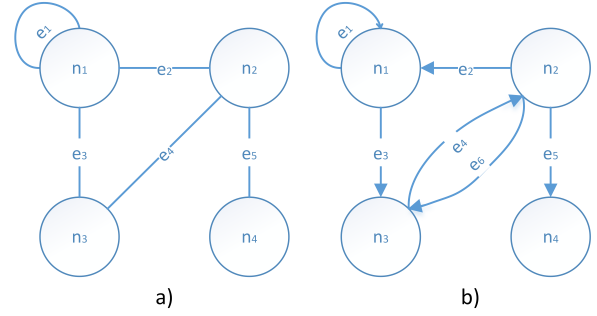


Figure 1. Examples of (a) undirected graph, and (b) directed graph.

in the adjacency matrix implies self edges, i.e., edges that emanates from the node and returns to the node.

An important concept in graph theory is the length ℓ between two nodes: the length is the number of nodes traversed to go from node n_i to node n_j . In G_2 of Fig. 1, the length in going from n_1 to n_1 is $\ell = 1$ because of the self edge. The length in going from n_1 to n_3 is $\ell = 1$ because there is an edge from n_1 to n_3 . The length in going from n_1 to n_2 is $\ell = 2$: it is necessary to first go from n_1 to n_3 , and then from n_3 to n_2 .

The same principal can be used to represent a model structure of linear models with inputs and outputs, (Šiljak, 2011). These models can be represented by Eq. 1:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (1)$$

Here, $x \in \mathbb{R}^{n_x}$ is the state, $u \in \mathbb{R}^{n_u}$ is the input/control signal, and $y \in \mathbb{R}^{n_y}$ is the output. $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, $C \in \mathbb{R}^{n_y \times n_x}$ and $D \in \mathbb{R}^{n_y \times n_u}$ are constant matrices and consist of elements $a_{i,j}$, $b_{i,j}$, $c_{i,j}$ and $d_{i,j}$, respectively.

In order to represent a structure of this system, the interconnection square matrix M should be created, (Šiljak, 2007). This interconnection matrix M combines information from all the constant matrices, A , B , C , D , and represents the relationships between states, inputs and outputs. The matrix M is found as follows:

$$M = \begin{bmatrix} A & B & 0 \\ 0 & 0 & 0 \\ C & D & 0 \end{bmatrix} \quad (2)$$

In M , the second block row is zero because u_k is an input and not a response variable, while the third block column is zero because y_k is a response variable and not an input.

2.2 Structural controllability

In control theory, the mathematical duals observability and controllability are important properties of control systems. Using controllability, it is possible to evaluate the capability of the external input capability to influence the internal state. Observability, on the other side, gives an understanding of the possibility of a system state to be inferred from an external output.

⁹Open Hydro Power Library is developed by the first author within his PhD study.

¹⁰<https://goo.gl/Qyjq2>

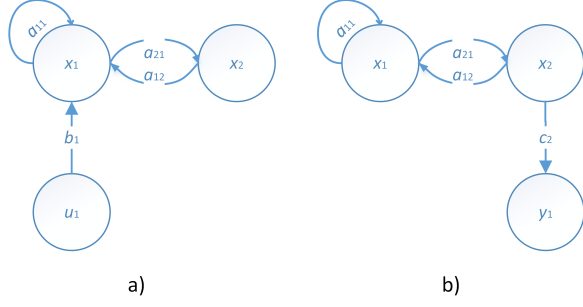


Figure 2. Examples of (a) structurally controllable, and (b) structurally observable systems.

As mentioned above, numerical problems can arise in the classical methods for observability and controllability computations in complex, large scale systems. As an alternative, structural observability and controllability are considered in this study due to the simplicity of these methods. In addition, structural observability and controllability provide necessary conditions for observability and controllability. This means that if the complex system is not structurally observable or not controllable, then it is not observable or not controllable. On the other hand, the system may be structurally observable/controllable, while in reality the system is not observable/controllable, (Šiljak, 2011).

Consider a linear system with an external input, see Eq. 1, e.g., $A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ and $B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$. First, the system structure based on the digraph G should be created with the interconnection matrix, $M = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, see Fig. 2 (a). Then, structural controllability of the system can be demonstrated if there is a directed path in the digraph G from (at least one) input-node to every single state-node. As seen in Fig. 2 (a), the system is structurally controllable, because there are paths from the input-node u_1 to the state-node x_1 with the edge b_1 and to the state x_2 with the edges b_1 and $a_{2,1}$.

2.3 Structural observability

Similarly to structural controllability, structural observability requires that there is a directed path from every single state-node to (at least one) output-node in the digraph G . Let us suppose a linear system with an external output, see Eq. 1, e.g., $A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ and $C = \begin{bmatrix} 0 & 1 \end{bmatrix}$. The interconnection matrix is then as follows, $M = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$. The system structure based on the digraph G is created using the matrix M and is shown in Fig. 2 (b). Here, the structural observability of the system is proven due to state-nodes x_1 and x_2 with directed paths $(a_{2,1}, c_2)$ and (c_2) to the output node y_1 , respectively.

2.4 Relative degree of system

Another property that can be found from structural analysis is the relative degree of the system, which shows how the input affects the system output. More precisely, the

relative degree r represents the number of differentiations of the output y needed for the input u to appear, (Slotine and Li, 1991). From the digraph structure of the system, this relative degree can be found as the smallest number of state-nodes through which a directed path from an input-node to output-node goes.

Combining the two previous examples with one input and one output, e.g., $A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$, $B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $C = \begin{bmatrix} 0 & 1 \end{bmatrix}$, it can be shown that the relative degree of the system equals two, see Fig. 2 (a) and (b) together. This is because the path from the input-node u_1 to the output-node y_1 is $(b_1, a_{2,1}, c_2)$ and this path goes through two state-nodes x_1 and x_2 . This statement can be also shown by the condition for relative degree r from (Slotine and Li, 1991):

$$r = \min_{\rho} \left\{ L_g L_f^{\rho-1} h(x) \neq 0 \right\} \quad (3)$$

Here, we consider the system $\dot{x} = f(x) + g(x)u$ and $y = h(x)$. Symbols L_g and L_f are the Lie derivatives of $h(x)$ along $g(x)$ and $f(x)$, respectively, i.e., $L_g h(x) = \frac{\partial h(x)}{\partial x} g(x)$ and $L_f h(x) = \frac{\partial h(x)}{\partial x} f(x)$. Hence, in our example it is proven by Eq. 4 that the relative degree r equals two:

$$\begin{aligned} \text{for } \rho = 1: & \quad L_g h(x) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 0 \\ \text{for } \rho = 2: & \quad L_g L_f h(x) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 1 \end{aligned} \quad (4)$$

In the case of multiple inputs or outputs, a set of relative degrees appears for each output-node. In such cases, in addition to this set of relative degrees, a total relative degree of the system is defined. The total relative degree is nothing but sum of the set of relative degrees, (Slotine and Li, 1991).

3 Julia implementation

For the prototype tools in this paper, a linear model in state space form as in Eq. 1 is assumed. Such a representation might be found in two ways. In one case, a dynamic system is modeled directly in Julia with *DifferentialEquations.jl* package, (Rackauckas and Nie, 2017), and then can be linearized using the *ForwardDiff.jl* package¹¹, (Revels et al., 2016). Alternatively, the model can be represented in Modelica, and OpenModelica with OMJulia can be used for the model linearization. The Julia API of OpenModelica with OMJulia is similar to the Python API of OpenModelica with OMPython which has been discussed in previous work, (Vytyvtskyi and Lie, 2018b).

In order to work with graphs in Julia the *LightGraphs.jl* package¹² and *GraphPlot.jl* package¹³ can be used for graph creation and plotting, respectively. In addition to these packages, other Julia packages for this study are also required, i.e., *Plots.jl*¹⁴ and *DataFrames.jl*¹⁵ packages.

¹¹<https://goo.gl/en5JMu>

¹²<https://goo.gl/tveMx1>

¹³<https://goo.gl/ifVwlp>

¹⁴<https://github.com/JuliaPlots/Plots.jl>

¹⁵<https://github.com/JuliaData/DataFrames.jl>

It should be noted that examples with results of using all functions presented in this section, are given in Section 4.

3.1 Graphical structure of system

The *LightGraphs.jl* package¹⁶ can be used to create a digraph of the linear system structure in Julia. Using interconnection matrix M (described in eq. 2) as an input to the *DiGraph()* command, the digraph G can be created and then plotted with the *gplot()* command from the *Graph-Plot.jl* package¹⁷. An example, Julia code for creating and plotting the digraph of a simple three by three interconnection matrix M previously presented for the controllability example (see Fig. 2 (a)) looks as follows:

```
M = [ 1.0 1.0 1.0;
      1.0 0.0 0.0;
      0.0 0.0 0.0] // intercon. matrix
G = DiGraph(M) // create the digraph
gplot(G, layout=spring_layout,
      NODESIZE = 0.1,
      nodefillc=colorant"turquoise",
      NODELABELSIZE = 5,
      nodelabel=["x1", "x2", "u1"],
      nodelabelc = colorant"black",
      EDGELINEWIDTH=0.5,
      edgestrokec=colorant"grey",
      arrowlengthfrac=0.08,
      arrowangleoffset = pi/10) // plotting
```

Here, plotting of the graph with *gplot()* command has various options:

- various possibilities for graph layout (random, circular, spring, shell, stressmajorize, and spectral layouts);
- setting size and color for nodes (*NODESIZE*, *nodefillc*), nodes' labels (*NODELABELSIZE*, *nodelabelc*), or edges (*EDGELINEWIDTH*, *edgestrokec*);
- specifications of nodes' labels names (*nodelabel*);
- setting edges' arrows shape (*arrowlengthfrac*, *arrowangleoffset*).

It should be noted that all color settings in the *gplot()* command might be specified with a vector of colors, one for each nodes/edges, similarly to the presented name vector for the nodes' labels. All the discussed options can be specified by the user according to their choice.

The results of running the presented code is shown in Fig. 3, where the simple digraph of three nodes is presented. Hence, using the presented commands for graphs creation and plotting, our own functions for system structure construction can be developed. The first function is named *obtain_graph_structure()* and provides digraph G together with interconnection matrix M and a data table, where the nodes' labels are structured with respect

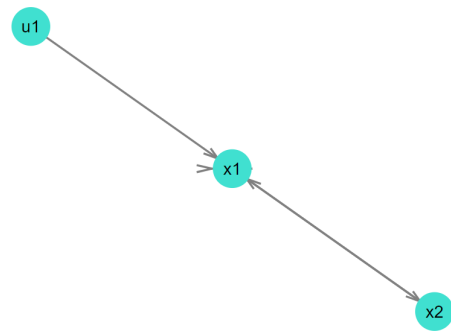


Figure 3. Digraph from the simple example of Julia code for the system in Fig. 2 (a).

to state/input/output names. This function also gives specific color arrays for nodes, nodes labels and edges. All nodes are colored according to their type, i.e., individual colors for states (turquoise), inputs (light blue), and outputs (light green). Edges in turn are colored based on their connection, i.e., individual color for self loops (red), state interactions (grey), connections from input (blue), and to output (brown). The inputs to this function are the constant system matrices A , B , C , and D . Three string arrays with the variables' names of state, input and output are inputs as well.

An example of the *obtain_graph_structure()* function calling is provided below. Here, commands for displaying of the graph G and the data table df are also used. As it is seen, the *gplot()* command for the graph plotting is specified with different options found with the *obtain_graph_structure()* for the names of nodes and color vectors for nodes, labels and edges.

```
G,M,df,Node_c,Edg_c,Nodelable_c,Nodelble =
  obtain_graph_structure(A,B,C,D,
    StateName,InputName,OutputName);
println(df) // display the data table
gplot(G, layout=circular_layout,
      NODESIZE = 0.05,
      NODELABELSIZE = 5,
      nodefillc=Node_c,
      EDGELINEWIDTH=0.3,
      edgestrokec=Edg_c,
      nodelabel=Nodelble,
      nodelabelc = Nodelable_c,
      arrowlengthfrac=0.08,
      arrowangleoffset = pi/10)
```

In cases when the user does not want to display (print/plot) the results, another function *system_structure()* can be used. The function gives the possibility to show the structure of the system directly after execution. The use of this function is provided below:

```
system_structure(A,B,C,D,
  StateName,InputName,OutputName)
```

¹⁶<https://goo.gl/tveMx1>

¹⁷<https://goo.gl/ifVwlp>

3.2 Structural observability and controllability

As presented above, structural digraph paths should be checked in order to show the structural observability or controllability of the system. To be structurally observable, there should be a directed path from every state-node to at least one output-node. Similarly, there should be a directed path from at least one input-node to every state-node in order to be structurally controllable.

Functions for checking structural observability and controllability of the system have been developed: `check_sys_observ()` and `check_sys_control()`, respectively. The calling of these functions are the same as for the functions presented in the previous subsection, and an example is given below:

```
check_sys_observ(A,B,C,D,  
                StateName, InputName, OutputName)
```

```
check_sys_control(A,B,C,D,  
                 StateName, InputName, OutputName)
```

Both functions operate in similar way, and in the case that all states of the system are structurally observable/controllable, they return a message with the following text: “All states are structurally observable/controllable”. Otherwise, these functions provide information of which states are structurally unobservable/uncontrollable. In both cases, the functions also display the digraph with a structure of the system. In addition, in the case with some unobservable/uncontrollable states, some transparency colors are used to display these state-nodes and the edges connected to these nodes.

In some cases, it might be of interest to specifically check some of the system states for observability or controllability. Because of this, another two functions that check structural observability/controllability of specified states are developed. The use of these functions are similar to the previous two functions, but here the user should also specify the state that will be checked. The state number from the node’s label (`state_num`) is used for this specification. An example looks as follows:

```
check_state_observ(A,B,C,D, StateName,  
                  InputName, OutputName, state_num)  
check_state_control(A,B,C,D, StateName,  
                   InputName, OutputName, state_num)
```

Both functions return a message that shows if the specified state is structurally observable/controllable or not. They also display the digraph with a structure of the system where the specified state-node and a path (edges and nodes) which shows its structural observability/controllability are highlighted. Colors of all other nodes and edges are somewhat transparent.

3.3 Relative degree of system

In order to determine the relative degree of the system presented by digraph G , a smallest number of state-nodes should be found through which a directed path from

an input-node to output-node goes. For this task, the `sys_relative_degree()` function is developed. This function defines the relative degree of the system and then returns a message that shows the value of the defined relative degree. For a system with multiple inputs and outputs, information about the total relative degree is provided together with a set of the relative degrees of all outputs. Moreover, a digraph is displayed with the structure of the system. In this digraph, colors of all nodes and edges are a bit muted, except for the path/paths (edges and nodes) that is/are the basis for the relative degree.

The use of the function for checking the relative degree looks as follows:

```
sys_relative_degree(A,B,C,D,  
                  StateName, InputName, OutputName)
```

4 Results

The various hydropower models that are implemented in OpenModelica using our in-house hydropower library, *OpenHPL*, are used here for testing of the developed functions for system structural analysis. Description and information about these hydropower models already have been presented previously, (Vytvytskyi and Lie, 2017, 2018b). For use with the structural analysis code, these models are first linearized in Julia using package OMJulia for OpenModelica. The constant A , B , C , and D matrices for the linearized hydropower state space models together with ordered lists (vectors) of state, input and output names are then used for structural analysis.

4.1 Simple waterway model

First, a simple model of the hydropower system with basic models for the waterway (incompressible water and inelastic pipes, (Vytvytskyi and Lie, 2017)) is used. This model consists of 5 states and has one input and one output. The `system_structure()` function provides the model structure, see Fig. 4. Here, the states ($x_1 - x_5$) are colored turquoise and consist of the volumetric flow rates in the penstock and surge tank, and the water masses in the surge tank, reservoir, and tail water. The input (u_1) is the control signal for the turbine and is colored light blue. The output (y_1) is colored light green and represents the flow rate in the turbine which is the same as the penstock flow rate in this model. Figure 4 shows the digraph with the model structure using the circular layout for the graph plotting. This can be changed to another style in options to the `gplot()` command.

Next, the hydropower model can be checked for structural observability and controllability using `check_sys_observ()` and `check_sys_control()` commands. The results for these studies are shown in Fig. 5 for observability and in Fig. 6 for controllability. It is seen from Fig. 5 that the system is structurally observable because all system states transmit information through digraphs to the output. In the same way, there are two uncontrollable states which make system structurally

```
7x4 DataFrames.DataFrame
```

Row	NodeLabel	State	Input	Output
1	x1	penstock.V_dot		
2	x2	reservoir.m		
3	x3	surgeTank.V_dot		
4	x4	surgeTank.m		
5	x5	tail.m		
6	u1		u	
7	y1			dotV

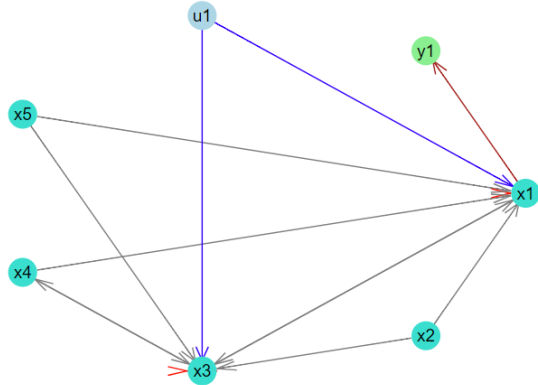


Figure 4. The digraph with the simple model structure determined by the `system_structure()` function.

uncontrollable, see Fig. 6. This uncontrollability for the water masses in the reservoir and tail water is caused by model simplification: these masses are kept constant in the model, (Vytvytskyi and Lie, 2018b); the uncontrollability is thus fictitious in this case.

The relative degree of this simple hydropower model can be found using the developed `sys_relative_degree()` function. The result of running this command for the simple model is shown in Fig. 7. Here, it is seen that the relative degree, r , equals one, which means that the control signal directly affects a state that influences the output.

4.2 Detailed waterway model

A more detailed model of the hydropower system is used next. This model is similar to the previous simple model, but here the penstock unit is described by a more detailed pipe model instead of the basic pipe model (here, compressible water and elastic pipes are considered in the penstock, (Vytvytskyi and Lie, 2017)). This model consists of 24 states and also has one input and one output. The result of the `system_structure()` function provides the model structure, see Fig. 8. Here, the states ($x_1 - x_{24}$) consist of the pressures ($U[1, \dots, 10]$) and mass flow rates ($U[11, \dots, 20]$) in the penstock segments, volumetric flow rate in the surge tank, reservoir, and tail water. The input (u_1) is the control signal for the turbine and the output (y_1) is the flow rate through the turbine. The state, input, and output nodes are colored in the same way as previously. It is seen from Fig. 8 that for more complex systems (more nodes), it becomes harder to observe visually how the nodes are connected. One way to study the system structure is to decompose the system in smaller subsystems. This can easily be

All states are structurally observable

```
7x4 DataFrames.DataFrame
```

Row	NodeLabel	State	Input	Output
1	x1	penstock.V_dot		
2	x2	reservoir.m		
3	x3	surgeTank.V_dot		
4	x4	surgeTank.m		
5	x5	tail.m		
6	u1		u	
7	y1			dotV

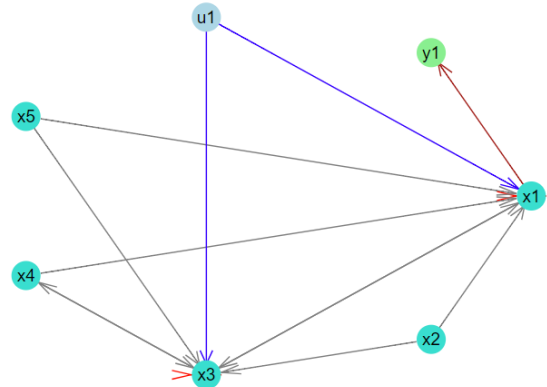


Figure 5. The results of checking the structural observability for simple model by the `check_sys_observ()` function.

The uncontrollable states are: reservoir.m, tail.m

```
7x4 DataFrames.DataFrame
```

Row	NodeLabel	State	Input	Output
1	x1	penstock.V_dot		
2	x2	reservoir.m		
3	x3	surgeTank.V_dot		
4	x4	surgeTank.m		
5	x5	tail.m		
6	u1		u	
7	y1			dotV

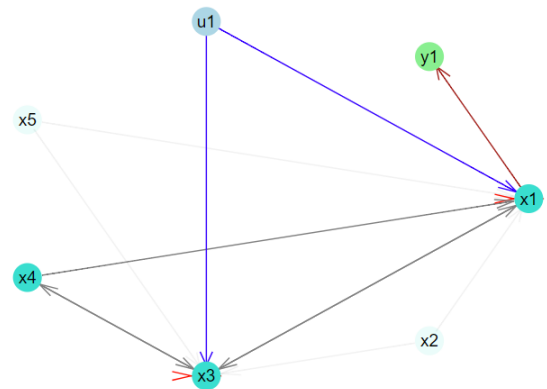


Figure 6. The results of checking the structural controllability for simple model by the `check_sys_control()` function.

Relative degree of the system is: $r = 1$

Row	NodeLabel	State	Input	Output
1	x1	penstock.V_dot		
2	x2	reservoir.m		
3	x3	surgeTank.V_dot		
4	x4	surgeTank.m		
5	x5	tail.m		
6	u1		u	
7	y1			dotV

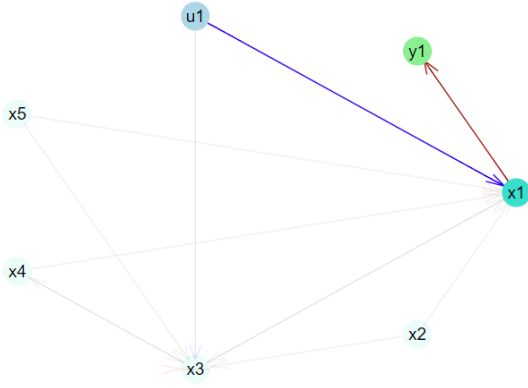


Figure 7. The results of checking the relative degree for simple model by the `sys_relative_degree()` function.

Row	NodeLabel	State	Input	Output
1	x1	penstock.U[1]		
2	x2	penstock.U[2]		
3	x3	penstock.U[3]		
4	x4	penstock.U[4]		
5	x5	penstock.U[5]		
6	x6	penstock.U[6]		
7	x7	penstock.U[7]		
8	x8	penstock.U[8]		
...				
18	x18	penstock.U[18]		
19	x19	penstock.U[19]		
20	x20	penstock.U[20]		
21	x21	reservoir.m		
22	x22	surgeTank.V_dot		
23	x23	surgeTank.m		
24	x24	tail.m		
25	u1		u	
26	y1			dotV

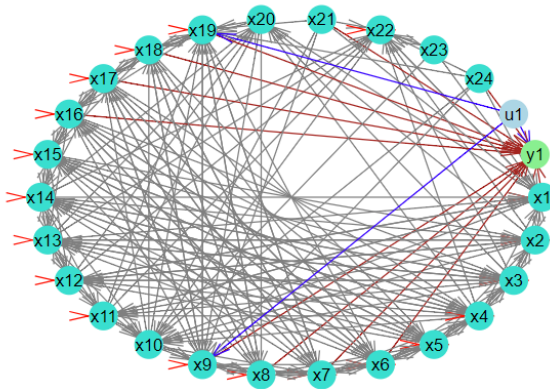


Figure 8. The digraph with the detailed model structure determined by the `system_structure()` function.

done by picking up the appropriate rows and columns in the A , B , and C matrices with respect to interested states.

On the other hand, the structural analysis for observability, controllability, and relative degree of the system may be still performed for the complete model. This can be done by running the same functions for the detailed model: `check_sys_observ()` — for observability, `check_sys_control()` — for controllability, and `sys_relative_degree()` — for relative degree. The results of these functions are not presented here to save space. However, the resulting information is as follows:

- “All states are structurally observable”.
- “The uncontrollable states are: reservoir.m, tail.m”, similarly to the case with simple waterway model.
- “Relative degree of the system is: $r = 0$ ”, the input signal directly affects the output, i.e., the constant matrix D is not zero.

4.3 Simple waterway model with generator

Here, the simple model presented above is studied with a model of a synchronous generator that is connected to the grid. The models of this electrical part (generator, grid, etc.) are taken from the *OpenIPSL*¹⁸ library, and is used in OpenModelica. OpenIPSL is the Open-Instance Power System Library, where a wide variety of power system components are available. The model of the simple hydropower waterway and generator consists of 7 states and has one input and 3 outputs. Here, the states ($x_1 - x_7$) consist of the generator shift angle and angular velocity, the volumetric flow rates in the penstock and surge tank, and the water masses in the surge tank, reservoir and tail water. The input (u_1) is the control signal for the turbine and the outputs ($y_1 - y_3$) are the generator power production and angular velocity, and flow rate through the turbine. The state, input and output nodes are colored in the same way as previously.

The result of the `system_structure()` function provides the model structure, Fig. 9. This structure is presented with the digraph of another layout type (spring layout), in order to demonstrate another structural view. The structural analysis for observability, controllability and relative degree of the system is also performed for this model case. This can be done by running the same functions as for the detailed model: `check_sys_observ()` — for observability, `check_sys_control()` — for controllability, and `sys_relative_degree()` — for relative degree. The results of executing these functions are not shown here, but the results are summarized as follows:

- “All states are structurally observable”.
- “The uncontrollable states are: reservoir.m, tail.m”, similarly to the two previous cases.

¹⁸<https://openips1.readthedocs.io/en/latest/>

Row	NodeLabel	State	Input	Output
1	x1	order2_1.delta		
2	x2	order2_1.w		
3	x3	penstock.V_dot		
4	x4	reservoir.m		
5	x5	surgeTank.V_dot		
6	x6	surgeTank.m		
7	x7	tail.m		
8	u1		u	
9	y1			P
10	y2			dotV
11	y3			w

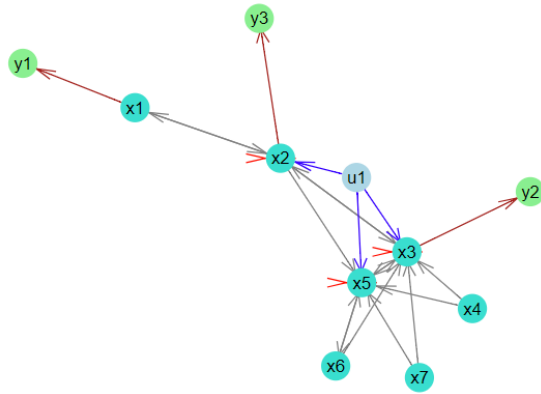


Figure 9. The digraph of the model structure determined by the `system_structure()` function. The model of the simple hydropower waterway and generator is used.

- “The system have relative degree (2, 1, 1). Total relative degree of the system is: $r = 4$ ”: here is shown first the relative degrees for each output and then the total relative degree of the system.

5 Discussion and Conclusions

This paper has explored the possibilities of using graph theory methods for structural analysis of dynamic system. Although the chosen examples hardly qualify as complex/large scale, graph methods scale well to huge systems. The presented methods have been implemented in Julia using the `LightGraphs.jl` and `GraphPlot.jl` packages. Using the `OpenHPL` hydropower library in `OpenModelica` and `OMJulia` for `OpenModelica`, the structural analysis methods have been tested on hydropower models of different complexity.

The results of testing the developed structural analysis functions look reasonable and can be further used for analysis related to state estimation and control: observability is a requirement for state estimators to work properly, controllability is required for control design, and relative degree is important in the design of nonlinear feedback controllers. One experience with the developed tools is that sometimes it can be hard to make a good visualization of the graph structure of complex (large scale) system. It can be hard to see the whole picture of the system structure (small subsystems are not easily seen) using the circular layout for the graph plotting. However, the user can do some testing of different layout types for the graph plotting to find the most appropriate one. Moreover, the graph

can be stored in a picture with higher resolution and bigger size that can help to see the system structure in a better way. In addition, developers of `LightGraphs.jl` and `GraphPlot.jl` packages are planning to improve the plotting possibilities of graphs in future, e.g., to improve the display self loop edges, etc.

In summary, this paper has explored some possibilities with structural analysis. Further work should be put into streamlining the functions into a package, with better use of Julia coding conventions, integration with other modeling tools, integration with control packages, etc.

References

- S. Boyd and L. Vandenberghe. *Introduction to Applied Linear Algebra*. Cambridge University Pr., 2018. ISBN 1316518965.
- N. Deo. *Graph Theory with Applications to Engineering and Computer Science*. Dover Publications, 2017. ISBN 9780486820811.
- J. Lunze. *Feedback control of large scale systems*. Prentice-Hall international series in systems and control engineering. Prentice-Hall, 1992. ISBN 9780133183535.
- Magamage Anushka Sampath Perera. *State estimation and optimal control of an industrial copper electrowinning*. PhD thesis, University College of Southeast Norway, Faculty of Technology, Kongsberg, Norway, 2016.
- C. Rackauckas and Q. Nie. `Differentialequations.jl` – a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, 5(1), 2017. doi:10.5334/jors.151.
- J. Revels, M. Lubin, and T. Papamarkou. Forward-mode automatic differentiation in julia. *arXiv:1607.07892 [cs.MS]*, 2016. URL <https://arxiv.org/abs/1607.07892>.
- Dan Simon. *Optimal State Estimation: Kalman, H ∞ , and Nonlinear Approaches*. John Wiley & Sons, 2006.
- Jean-Jacques E Slotine and W. Li. *Applied nonlinear control*. Prentice-Hall International Editions. Prentice-Hall, Englewood Cliffs, N.J, 1991.
- D. D. Šiljak. *Large-Scale Dynamic Systems: Stability and Structure*. Dover Civil and Mechanical Engineering Series. Dover Publications, 2007. ISBN 9780486462851.
- D. D. Šiljak. *Decentralized Control of Complex Systems*. Dover Books on Electrical Engineering Series. Dover Publications, 2011. ISBN 9780486486147.
- L. Vytvytskyi and B. Lie. Comparison of elastic vs. inelastic penstock model using `OpenModelica`. In *Proceedings of the 58th Conference on Simulation and Modelling (SIMS 58) Reykjavik, Iceland, September 25th–27th, 2017*, number 138, pages 20–28. Linköping University Electronic Press, Linköpings Universitet, 2017. doi:10.3384/ecp1713820.
- L. Vytvytskyi and B. Lie. Mechanistic model for Francis turbines in `OpenModelica`. *IFAC-PapersOnLine*, 51(2):103–108, 2018a. doi:10.1016/j.ifacol.2018.03.018.

- L. Vytvytskyi and B. Lie. Linearization for Analysis of a Hydropower Model using Python API for OpenModelica. In *Proceedings of the 59th Conference on Simulation and Modelling (SIMS 59) Oslo, Norway, September 26th–28th, 2018*, 2018b.

Paper G



OpenHPL for Modelling the Trollheim Hydropower Plant

Liubomyr Vytvytskyi, & Bernt Lie

Published in *Energies*, 12(12), 2303, 2019.

Article

OpenHPL for Modelling the Trollheim Hydropower Plant

Liubomyr Vytvytskyi  and Bernt Lie * 

Department of Electrical engineering, Information Technology and Cybernetics, University of Southeastern Norway, 3901 Porsgrunn, Norway; Liubomyr.Vytvytskyi@usn.no

* Correspondence: Bernt.Lie@usn.no

Received: 23 April 2019; Accepted: 12 June 2019; Published: 17 June 2019

Abstract: An increase of tightly integrated, renewable energy sources with highly varying production leads to a greater need for flexible hydropower plants to “balance” the intermittent production from these sources. From a systems perspective, the intermittency aspect constitutes a disturbance, while the tight integration leads to a multivariable systems, both of which causes increased challenges for good control. This instigates the need for a model based analysis and synthesis tool which can describe dynamic phenomena, supports multiphysics problems, and allows for immediate use in advanced control analysis and design. Previously, an open source Hydro Power Library (*OpenHPL*) have been developed within the multiphysics Modelica eco-system which satisfies the above requirements: *OpenHPL* contains a set of model units relevant for hydropower systems, as well as examples of tools for analysis of these models for control purposes. Thus far, *OpenHPL* has been validated by comparison against other simulation tools. However, a real test of *OpenHPL* is to use it to describe a real hydropower plant using minimal plant information, to tune model parameters against experimental data, and to validate the model. As a case study, experimental data from the Trollheim hydropower plant in Norway have been made available, and used to test *OpenHPL*. By flow sheeting a hydropower model in OpenModelica using *OpenHPL*, the model can be simulated from within a script language (Python via the OMPython API, Julia via the OMJulia API) and further analyzed using analysis tools in the script language. Based on default model parameter suggestions from *OpenHPL*, least squares model fitting was carried out in Python, and the model was validated with good model fit. Important parts of the modelling phase were the development of a new friction model for the Francis turbine, and iterations on the description of the turbine outlet geometry. The results are complemented with a discussion of possible uses of the model.

Keywords: modelling; hydropower; modelica library

1. Introduction

1.1. Background

A transition towards more renewable, and tightly integrated energy sources is currently happening in Europe and all over the world. This situation leads to increase in the use of flexible hydropower plants to compensate the highly changing production from intermittent energy sources such as wind and solar irradiation. Intermittency implies disturbance from a control systems perspective, and tight integration implies multivariate behavior. This instigates the need for new, modern multiphysics simulation tools with support for hydropower systems, which at the same time can be tightly integrated with tools for advanced control analysis and synthesis. A modern trend is towards open source tools which can be freely used in education, and which allows for user participation in the further development. High head hydropower plants are particularly important for balancing intermittent energy sources, and also play an important role in Norway.

A general research question is then: how should an open source simulation tool for hydropower systems be designed? A partial answer lies in the previous development of a hydropower library *OpenHPL* within the multiphysics modelling language Modelica, and the development of analysis tools in, e.g., Python or Julia, which can interact with the Modelica code via APIs OMPython and OMJulia in the free tool OpenModelica. Because Modelica is a multiphysics language, a waterway model based on *OpenHPL* can be directly connected to a model of the electric grid (based on electric libraries), and to any other system modelled in Modelica—as long as the interfaces are compatible.

In this paper, we consider the more specific research question: how suitable is *OpenHPL* for describing a real hydropower plant, and how can model parameters be tuned to fit experimental data? To answer this question, experimental data from the Trollheim hydropower plant in Norway have been made available.

1.2. Previous Work

In general, waterway dynamics of high head hydropower systems can be represented by a model of water flow in pipes. Hence, mass and momentum balances are normally used for modelling of the waterway units. Depending on the desired level of accuracy, models of the waterway unit can be described by ordinary differential equations (ODEs) in cases with incompressible water and inelastic walls. Partial differential equations (PDEs) appear for the more realistic cases with elastic walls and compressible water; various principles of solving the resulting hyperbolic PDEs are used. Model of turbine units can be based on using the Euler turbine equation in the case of a mechanistic model, or based on look-up tables (static characteristic curves for specific turbines)—for an empirical model. Some work on models for these units of the hydropower systems has already been presented in the literature [1–4]. Applications of hydropower models have also been published in [5–7]. Various examples of software tools for hydropower modelling also exist, and a basic overview of some of them has been given in [8].

One of these hydropower modelling tools is LVTrans [9] which is based on LabView [10] for transient simulations of hydropower plants [11]. LVTrans uses a MOC (Method of Characteristics) solver for simulation of PDEs. Turbine units in LVTrans are described by physical models that only need the turbine nominal operating values and do not need look-up tables for simulation.

Another tool, TOPSYS, a combined effort of Wuhan University and Uppsala University, is based on Visual C++ and provides models for various components of hydropower plants [6,12]. This software allows for simulation of elastic conduits with compressible water described by PDEs. Turbine models in TOPSYS are based on look-up tables, which requires fitting a model to experimental data or use of static characteristic curves for specific turbines.

SIMSEN [13] is a simulation tool for the analysis of electrical power networks, adjustable speed drives and hydraulic systems, developed and used at École Polytechnique Fédérale de Lausanne [14]. This software uses analogue models (RLC components) for modelling hydraulic elements. Turbine units in SIMSEN are also modelled based on look-up tables.

The commercial Modelica Hydro Power Library [15] from Modelon [16] provides a framework for modelling and simulating hydropower plant operations in a commercial Modelica-based modelling and simulation environment—Dymola [17–19]. This Modelica library uses staggered grid for discretization and simulation of PDEs. Description of the turbines is based on look-up tables for this Hydro Power Library.

Among these modelling and simulation tools for hydropower systems, only LVTrans is currently open-source software and freely available. However, LVTrans is based on and requires the commercial tool LabView, and is thus not truly openly accessible. In addition, LVTrans is used only for transient simulations of hydropower plants and is not the multiphysics tool. The Modelica Hydro Power Library from Modelon as well as SIMSEN support multiphysics systems.

Our work on modelling the waterway and the Francis turbine for high head hydropower systems using the multiphysics tool OpenModelica [20], has been reported in [21–23]. Unit models have been

developed for our in-house open-source Modelica [24] library—*OpenHPL* (Open Hydro Power Library) is developed by the first author within his PhD study).

A Python API named OMPython [25] for OpenModelica already exists which provides possibilities for controlling simulations of the OpenModelica models via Python [26]. Python, in turn, gives much broader possibilities for plotting, analysis, and optimization compared to what is possible in OpenModelica (e.g., using Python packages *matplotlib*, *numpy*, *scipy*, etc.) [27,28].

1.3. Overview of Paper

In this paper, the main contribution is testing the suitability of *OpenHPL* by building a model of a real hydropower system (Trollheim hydropower plant), fitting model parameters based on experimental data from the plant, and validating the model. Particular emphasis is put into describing the model geometry of the turbine outlet. Moreover, the turbine design algorithm developed in [23] is improved in this study with an analytical expression for the turbine loss coefficients. Both of these contributions try to answer the particular research question: “how suitable is *OpenHPL* for describing a real hydropower plant with minimal prior information, and how can model parameters be tuned to fit experimental data?”.

The paper is structured as follows: Section 2 provides a system description of the Trollheim hydropower plant and describes the experimental conditions. Section 3 gives an overview of hydropower modelling with the developed *OpenHPL* library and presents model cases of the Trollheim hydropower plant. Simulation results of the hydropower models that are tuned and fitted to real measured data are presented in Section 4. Finally, the discussion is given in Section 5 and the final section provides the conclusions.

2. System Description

2.1. Geometry Data

High head hydropower plants typically collect and store water in reservoirs in mountains, with tunnels leading the relatively small flow of water down a considerable height difference to the aggregated turbine and generator. The electricity produced by the generator is then transferred through power lines to consumers. More detailed description of the hydropower system has been presented in our previous work [21,22]. A typical structure of a high head hydropower plant is depicted in Figure 1 [21].

In this paper, the Trollheim hydropower plant [29] operated by Statkraft is used as a case study. The geometry data of this hydropower plant are provided in Tables 1 and 2. It should be noted that the structure of the Trollheim plant is similar to that presented in Figure 1, but all of the conduits (penstock, intake and discharge races) have more complex structures. These units consist of series of connected conduits with different geometry (length, slope, etc.), e.g., the intake race is divided into three parts: intake race #1, #2 and #3.

From Table 1, it should be clarified that the height difference for the reservoir corresponds to the water level in this water storage. In addition, a negative height difference represents a rising slope of the conduit, e.g., in the intake race #2 and discharge race #2: this negative height difference means that the inlet to the conduit has lower altitude in respect to sea level than the outlet.

It should also be noted that not all of the waterway units (e.g., intake or discharge races) are circular conduits. However, Table 1 presents diameter values for each waterway unit, which means that the circular conduit is used in our library for all unit models and the given diameter provides the same cross sectional area as the conduits have in reality.

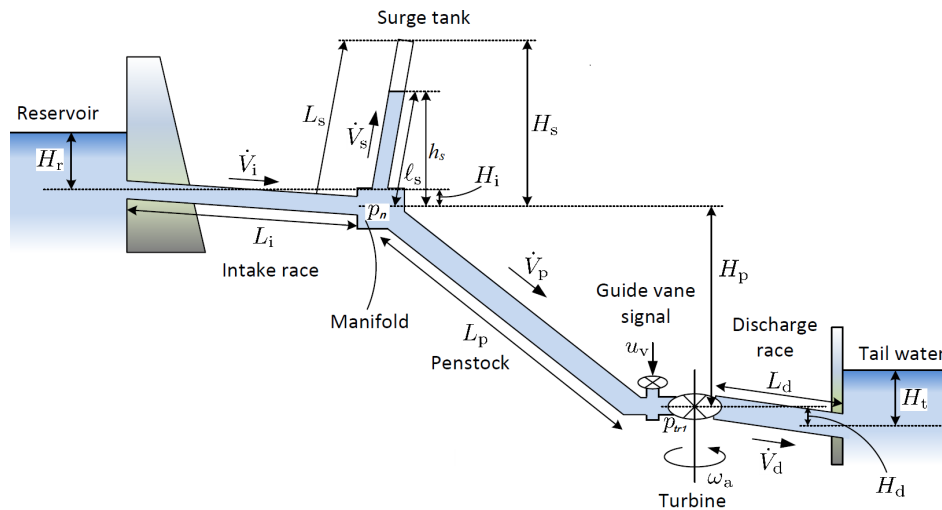


Figure 1. Overview of the structure of the high head hydropower plant.

Table 1. The waterway geometry; Trollheim plant.

Waterway Unit	Height Difference, (m)	Length, (m)	Diameter, (m)
Reservoir	46.5	—	—
Intake race #1	9	81.5	6.3
Intake race #2	−2	395	6.3
Intake race #3	9	4020	6.3
Penstock #1	233	363	4.7
Penstock #2	105	145	3.3
Surge tank	75.5	87	3.4
Discharge race #1	3.5	601	6.3
Discharge race #2	−8.6	21	6.3

Table 2. The turbine nominal operational values; Trollheim plant.

Turbine Type	Nominal Head (m)	Nominal Flow Rate (m ³ /s)	Nominal Power (MW)	Nominal Rotational Speed (RPM)
Francis	371	37	130	37

2.2. Experimental Data

The experiment on the Trollheim hydropower plant has been done using the HydroCord monitoring system, developed by [30]. The experimental measurements are taken within a one-hour window when the hydropower plant was run from zero to full load and after some time back from full to zero load. The experimental data consist of measurements with a sampling time of one second. All measured variables are listed in Table 3, and their values are available in Github [31]. Among these measured quantities are the variables that provide input information for our models. Other quantities belong to the models' outputs and can then be used for model fitting and validation. It should be mentioned that not all variables listed as "model input" in Table 3 are proper inputs/causes for the system. The proper input is the turbine control signal, while turbine rotational speed and tail water level are responses of this proper input. However, we have not included models of the rotational speed or tail water level, so, from a model fitting point of view, we need to use "model input" to simulate the system.

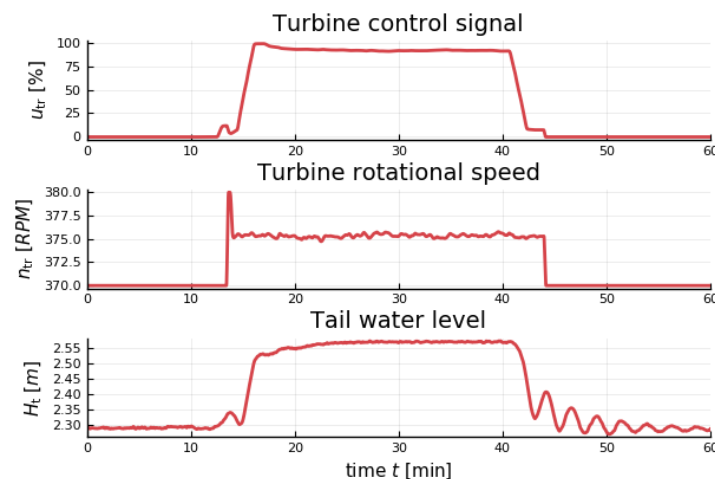
Table 3. Measured quantities.

Quantity Name	Symbol	Information
Turbine guide vane control signal	u_v	Model input
Turbine rotational speed	n_{tr}	
Water level in the tail water	H_t	
Turbine inlet pressure	p_{tr1}	Model output
Turbine outlet pressure	p_{tr2}	
Turbine volumetric flow rate	\dot{V}_{tr}	
Generated power	\dot{W}_{tr}	

The variation in the water level in the reservoir is also provided, but these changes are so small that we neglect them here and assume the reservoir water level to be constant.

For better understanding of the experiment, the experimental measurement of the signals that belong to model inputs are shown in Figure 2. The output measurements are presented in Section 4, where comparisons of these measurements and model outputs are shown. It is seen from Figure 2 that, after approximately 12 min of fully closed turbine guide vanes, the control signal increases to 100% in approximately 6 min. Then, the control signal slightly decreases to a nominal value of approximately 93% and the hydropower plant runs with the nominal load for 25 min. After approximately 40 min, the control signal closes the turbine guide vane openings over 3 min, and the remaining time these guide vanes stay closed.

Figure 2 shows that, even with closed guide vanes (no water flow through the turbine), the rotational speed is still observed and is close to nominal value. This is because the generator is working as an electrical consumer in this regime and rotate the shaft with the turbine runner. This is done in order to make generator synchronization easier when the power plant is turned on full load. The generator starts to produce electricity only when the nominal rotational speed is reached.

**Figure 2.** Measurements of the experimental conditions.

3. Hydropower Modelling

In this section, a description of how to model the hydropower system using the *OpenHPL* is given as well as a presentation of various cases of the Tollheim hydropower plant model for the further comparison with the measurements. As mentioned above, all modelling is done in OpenModelica, which is an open source Modelica-based modelling and simulation environment intended for industrial and academic usage. Modelica in turn is a multi-domain and component-oriented modelling language that is suitable for complex system modelling (some tutorials exist for Modelica [32], and OpenModelica [33]).

3.1. Hydropower Library Overview

For modelling of the hydropower system, library *OpenHPL* is used. This is an in-house hydropower library consisting of drag and drop elements that describe different units of the hydropower system such as reservoir, conduits, surge tank, and turbine. These elements can be structured in a flow sheet in the same way as the hydropower plant—see the hydropower structure in Figure 1 and the flow sheet in Figure 3. Then, in this flow sheet, all drag and drop elements can be specified with appropriate geometry of the hydropower plant by double-clicking on the element. Connectors that join each element hold information about the pressure in the connector and mass flow rate that flows through the connector—similar to the connection in an electrical circuit with voltage and current.

In comparison to other hydropower modelling tools presented in Section 1.2, *OpenHPL* has some advantages, perhaps the foremost being that the library is openly accessible with multiphysics support, and support for analysis. Another *OpenHPL* benefit is related to the included mechanistic Francis turbine model which makes it possible to simulate systems without detailed turbine characteristics; turbine efficiency charts are rarely openly available.

In the sequel, a more detailed description is given of *OpenHPL* elements related to the waterway and turbine units of the hydropower system.

3.1.1. Waterway Units

In *OpenHPL*, different waterway components of the hydropower system, e.g., reservoir, conduits, surge tank, are described by both mass and momentum balances, and could include compressible/incompressible water or elastic/inelastic pipe walls. A more detailed overview of the mathematical models and methods used in this library for the waterway components is given in [21,22]. The *Pipe* element in the library (see the left panel in Figure 3) represents a simple model of a circular conduit with incompressible water and inelastic pipe. This element might be used to model waterway elements, e.g., the intake race, penstock, and discharge race. In some cases, water compressibility and pipe elasticity should be taken into account for a more detailed model of waterway units exposed to large pressure differences, e.g, the penstock. For these cases, the *PenstockKP* element is available in our library and represents a detailed model of the circular conduit with either compressible water and inelastic pipe or compressible water and elastic pipe. A well-balanced Kurganova–Petrova, Finite Volume scheme is used to solve the partial differential equations [21]. For this method, the number of discretization segments should be set in the *PenstockKP* element. All other parameters for this element are similar to the *Pipe* parameters and geometry listed in Table 1.

There are also available *Reservoir* and *SurgeTank* elements that represent simple models for the reservoir and surge tank, respectively. More details of these models are given in [22].

3.1.2. Turbine Unit

The turbine unit can be expressed either with a simple turbine model based on a look-up table (turbine efficiency vs. guide vane opening) or with a detailed mechanistic Francis turbine model. The simple turbine model is described by Equation (1), where the mechanical turbine shaft power \dot{W}_{tr} is defined as:

$$\dot{W}_{tr} = \eta_h \Delta p_{tr} \dot{V}_{tr}. \quad (1)$$

Here, η_h gives the turbine hydraulic efficiency that is found from a standard turbine look-up table, and depends on the turbine control signal, u_v . Δp_{tr} is the pressure drop through the turbine, defined as the difference between inlet and outlet turbine pressures: $\Delta p_{tr} = p_{tr1} - p_{tr2}$. The relationship between the turbine volumetric flow rate \dot{V}_{tr} and the pressure drop Δp_{tr} is described through a simple valve-like expression as follows:

$$\dot{V}_{tr} = C_v u_v \sqrt{\frac{\Delta p_{tr}}{\rho}}. \quad (2)$$

Here, C_v in Equation (2) is the guide vane “valve capacity” that can be tuned by using the nominal turbine net head (nominal pressure drop) and the nominal turbine flow rate listed in Table 2. p^a is the atmospheric pressure.

Based on Equations (1) and (2), the simple turbine model is implemented in *OpenHPL* as the *Turbine* element. Our library also includes a mechanistic Francis turbine model that has been presented and described in detail in [23]. In [23], the mechanistic Francis turbine model is provided together with a design algorithm for all the needed turbine geometry parameters that are used in this model. Input data for the design algorithm are the turbine nominal operational values that can be found in Table 2 of this study. Both the Francis turbine model and the turbine design algorithm are realized in the *Francis* turbine element in our library. However, the friction coefficients that represent shock, $k_{ft,1}$, whirl, $k_{ft,2}$, and pipe friction losses, $k_{ft,3}$, should also be defined for the Francis turbine model [23]. In previous work, these coefficients were tuned using the least squares method to fit the turbine characteristic (power production vs. turbine flow rate) from the model with the same characteristic from the real turbine. The same principle is used and presented in Section 4.

3.2. Model Presentation

Using *OpenHPL*, models of different complexity for the Trollheim hydropower plant can be modelled in OpenModelica. An example of such a model for the Trollheim plant is shown in the flow sheet in Figure 3. Here, all components are set with geometry data from Tables 1 and 2. The input signals of the guide vane opening and the water level in the tail water are used in this model.

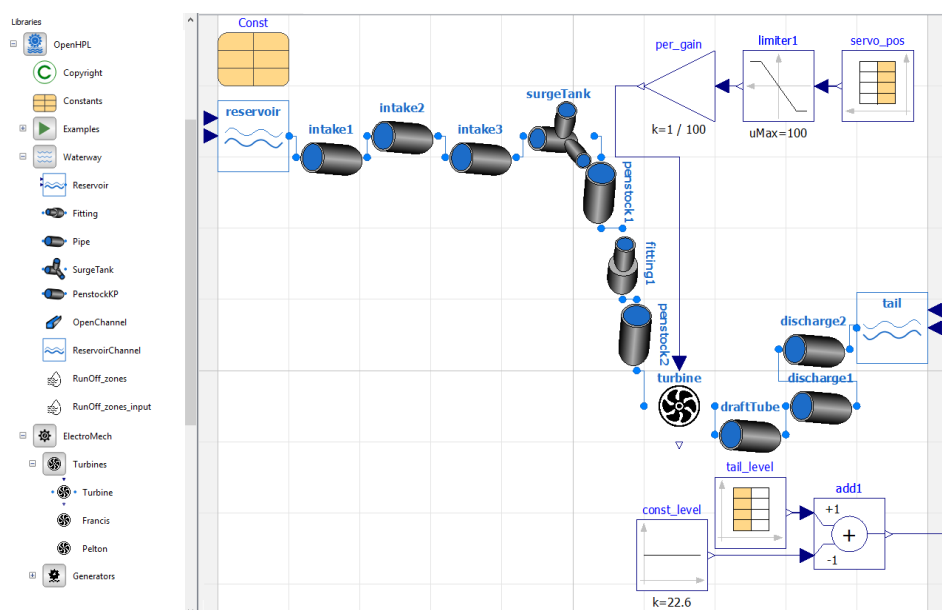


Figure 3. Flow sheet of the model of the hydropower system in OpenModelica. Modelled using *OpenHPL*.

Four models will be considered in the sequel; the first of these is based on the flow sheet in Figure 3, while the three others represent minor modifications of this flow sheet:

1. The model presented in Figure 3 is implemented using the simple *Pipe* with incompressible water and inelastic pipe walls for all waterway units. Moreover, a simple turbine model based on the look-up table for turbine efficiency is used here.
2. Hydropower model with detailed turbine—using the more detailed Francis turbine model (the *Francis* element in the library) instead of the simple turbine model based on the look-up table (the *Turbine* element in the library).

3. Hydropower model with detailed penstock—using the more detailed penstock model with compressible water and elastic walls (the *PenstockKP* element in the library) instead of the simple conduit model (the *Pipe* element in the library).
4. Hydropower model with both detailed turbine and penstock (both previous changes together).

Information about the turbine rotational speed is not needed for the simple turbine model (#1, #3 with the *Turbine* element), hence the measured turbine rotation is not used as the input in the hydropower model presented in Figure 3. However, the measured turbine rotation is used in the cases with the more detailed Francis turbine model that needs turbine rotational speed information.

It should also be noted that, currently, our hydropower library does not include a proper model for a draft tube unit [34]. However, Figure 3 shows that the flow sheet of the hydropower model includes a draft tube unit described by the simple *Pipe* element, similarly to modelling of the intake race, discharge race, etc. This *Pipe* element is used here as a tuning factor and its geometry and shape does not represent the real draft tube. More information about tuning of this draft tube element is provided in Section 4.

4. Model Fitting and Simulation

4.1. Simulation Overview

All simulations that are presented in Section 4 are done in OpenModelica using Python API OMPython to control the simulations, and handling the results in Python. For plotting in Python, the *matplotlib* package is used. In order to define model tuning parameters that provide the best model fit with the measured data, optimization package *scipy* is used. This package provides various algorithms that can be used for curve fitting with the least squares error method. More detailed information about the Python API and its use has been presented in [27,28].

The simulations are performed in the same way that the experiment has been done, using the same control input and sampling time. Here, a number of simulations for different models of the hydropower system are presented. First, tuning of the Francis turbine model parameters is shown. After that, the previously discussed four cases of the hydropower model are simulated and compared with the measured data.

4.2. Friction Coefficients for Francis Turbine Model

In order to find the turbine loss coefficients: $k_{ft,1}$, $k_{ft,2}$, and $k_{ft,3}$ that are used in the mechanistic Francis turbine model, the case of the hydropower model with the detailed mechanistic *Francis* turbine element is simulated first. The simulations for this model are performed over the time horizon of one hour (the same as in the experiment), but the results within a time interval of approximately 14–44 min is used for comparison with the experimental data. This is due to the need to compare turbine characteristics (the generated power vs. the turbine flow rate) from the model and the experimental data. This characteristic in turn is relevant when the generated power is greater than zero—the turbine runner rotates with the nominal rotational speed (time interval 14–44 min in Figure 2).

4.2.1. Tuning Loss Coefficients

First, the turbine loss coefficients $k_{ft,1}$, $k_{ft,2}$ and $k_{ft,3}$ have been manually tuned in order to reach an initial fitting of the model simulation results with the measured data from the experiment. In a previous study [23], it was found that typically, $k_{ft,1} \sim 10^5$, $k_{ft,2} = 0$, and $k_{ft,3} \sim 10^3$. By some manual tuning, here, we found the initial values $k_{ft,1} = 3 \cdot 10^5$, $k_{ft,2} = 0$ and $k_{ft,3} = 8 \cdot 10^3$.

Next, we fine tune the loss coefficients using a least squares method in SciPy [35], by using the above initial values as a starting point. From least squares tuning, we find the following loss coefficients: $k_{ft,1} = 3.095 \cdot 10^5$, $k_{ft,2} = 0$ and $k_{ft,3} = 8.33 \cdot 10^3$. Comparison of the power production vs. turbine flow rate characteristics found from the model simulation with least squares loss coefficients and the experimental data are shown in Figure 4.

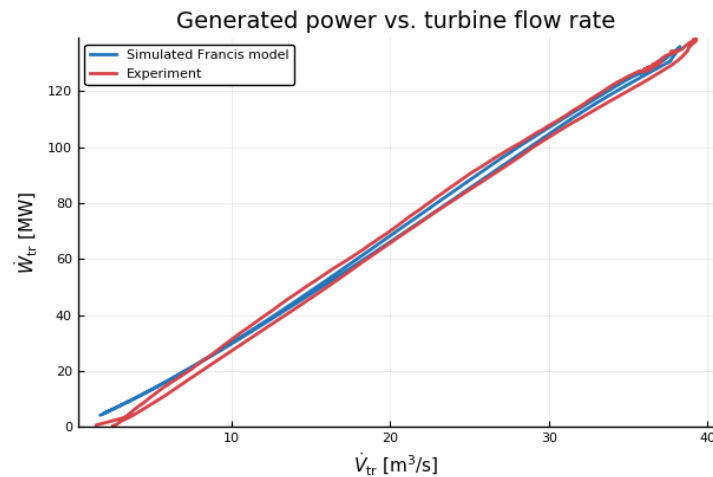


Figure 4. Comparison of the model simulation and experimental power vs. turbine flow rate characteristics.

It is seen from Figure 4 that the characteristic from the Francis turbine model (blue curve) fits reasonably well the experimental results on the real turbine. There is some deviation between the curves for low flow values close to zero. However, this deviation is not crucial and is caused by the difference between the measured generated power and the turbine shaft power taken from our model simulation. In addition, this deviation confirms the fact that the turbine model does not work well with the low load regimes (low turbine flow rate).

4.2.2. Analytic Expression for Loss Coefficients

Using the values found from the least squares error method, the turbine design algorithm might be improved with an analytic expression for finding the loss coefficient. This expression can be found using information from two other cases that have been previously described in [23]. Comparing the values for the loss coefficients and the turbine nominal values for all three cases, a relation between the turbine nominal head and two of the coefficients, $k_{ft,1}$ and $k_{ft,3}$, is observed. Loss coefficient, $k_{ft,2}$, seems to be negligible and is zero in all three cases. The relation between the turbine nominal head and natural logarithm of the loss coefficients, $k_{ft,1}$ and $k_{ft,3}$, is shown in Figure 5. Here, the three diamond markers in the data represent the value of the loss coefficients for each case of the hydropower system. The figure includes analytic expressions that characterize reasonably well the relations between the turbine nominal head and the loss coefficients. By studying results from model fitting, it has been observed that the logarithm of the turbine loss coefficients appear to be linearly related to the turbine nominal head, Figure 5. We thus propose the following expressions for the turbine loss coefficients:

$$k_{ft,1} = 11.6 \cdot 10^3 e^{8.9 \cdot 10^{-3} H_n}, \quad (3)$$

$$k_{ft,3} = 720 e^{6.7 \cdot 10^{-3} H_n}. \quad (4)$$

The exponential expressions for the $k_{ft,1}$ and $k_{ft,3}$ turbine loss coefficients represent an important improvement of the turbine design algorithm developed in [23]. With Equations (3) and (4), all the parameters needed for the mechanistic Francis turbine model can be found based on the turbine nominal operation values only.

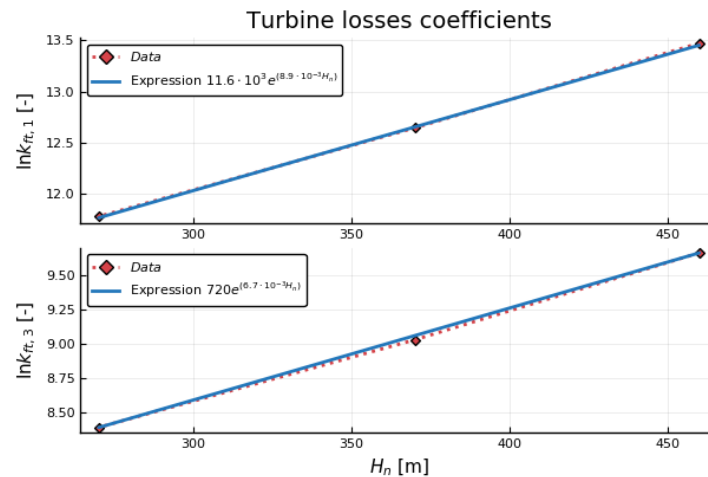


Figure 5. Linear relationship between natural logarithm of the turbine loss coefficients, $k_{ft,1}$ and $k_{ft,3}$ and the nominal head.

4.3. Simple Hydropower Model

The model for the complete hydropower system is simulated next in order to see how well the model results fit real experimental data. First, the model that has been shown in Figure 3 is considered with simple turbine and penstock (incompressible water and inelastic walls). The simulations are performed for one hour with the same scenario and sampling time as during the experiment.

A comparison of the simulation results and the experimental data are shown in Figure 6, where four plots for the turbine flow rate, the turbine inlet and outlet pressures, and the generated power are presented. It should be noted that the model has only been set with the geometry parameters of the Trollheim hydropower plant (Tables 1 and 2) without any additional tuning. The draft tube is excluded from this simulation to emphasize the importance of including it. It is seen from the figure that the model results fit the experimental data reasonable well, especially for the turbine flow rate and the generated power. The model results for the transient dynamics of the inlet turbine pressure slightly deviate from the measured results (model results with higher oscillation amplitude). For the outlet turbine pressure, the model results show significant deviation in comparison to the experimental measurements. The accuracy of the outlet turbine pressure results is quite important, especially for cavitation studies.

4.3.1. Pipe Roughness: Initial Manual Tuning

In order to improve the model results, the presented hydropower model can be further tuned. An important parameter for controlling friction and damping of oscillations is the pipe surface roughness height, ε . Thus far, an identical value ($\varepsilon = 0.05$ mm) has been used for all pipes. Because the inner pipe surfaces range from that of a rough, drilled tunnel in the intake race, to lined penstocks, this is clearly unrealistic. The ranges of the absolute roughness values for common construction materials are known and could be found from various absolute roughness tables [36].

As mentioned in Section 2, in reality, some of the waterway units have non-circular shape, but in the presented hydropower model all these units are considered as circular conduits. This simplification can also affect the Darcy friction factor because the ratio between roughness height and hydraulic diameter of the conduit, ε/D , is used to define the friction factor. For circular conduits, the hydraulic diameter equals simply the diameter of the conduit, however for conduits with other shapes, this hydraulic diameter equals conduit cross sectional area multiplied by four and divided by perimeter of the conduit. Hence, by tuning the roughness parameter, the conduit shape simplification is taken into account.

In an initial phase, we manually adjust the roughness heights—the results will later be used as initial guesses in a least squares fitting phase. It is assumed that the intake race roughness is the

highest in the waterway, i.e., as a long, rocky tunnel. In the same way, the penstock might have the lowest value for the roughness height, i.e., like a smooth steel pipe. Roughness values for the surge tank and the discharge race might lie somewhere between, i.e., as a smooth concrete pipe. As expected, the manual tuning phase confirms that changes in the roughness heights affect the amplitude of the transient oscillations. A comparison of the inlet turbine pressure from the experimental data, and two models with (i) common roughness height ε , and (ii) individual roughness heights ε are shown in Figure 7. For the model with individual roughness values, the roughness height for each waterway unit is set as follows: $\varepsilon = 0.5$ mm for the intake race #3; $\varepsilon = 0.1$ mm for the surge tank and intake races #1 and #2; $\varepsilon = 0.05$ mm for discharge races #1 and #2; and finally penstock #1 and #2 have $\varepsilon = 0.0005$ mm.

Figure 7 shows that the model with individually fitted roughness values for each waterway unit provides better results than the model with fixed (common) roughness for all units. The improvements are visible in the amplitude of the transient oscillation and the results of the model with different roughness deviate less from the experimental data. These improvements are also confirmed by the root mean squared error (RMSE) between the model and experiment inlet pressures. The RMSE is reduced from 0.142 bar for the model with common roughness to 0.104 bar for the model with individual roughness.

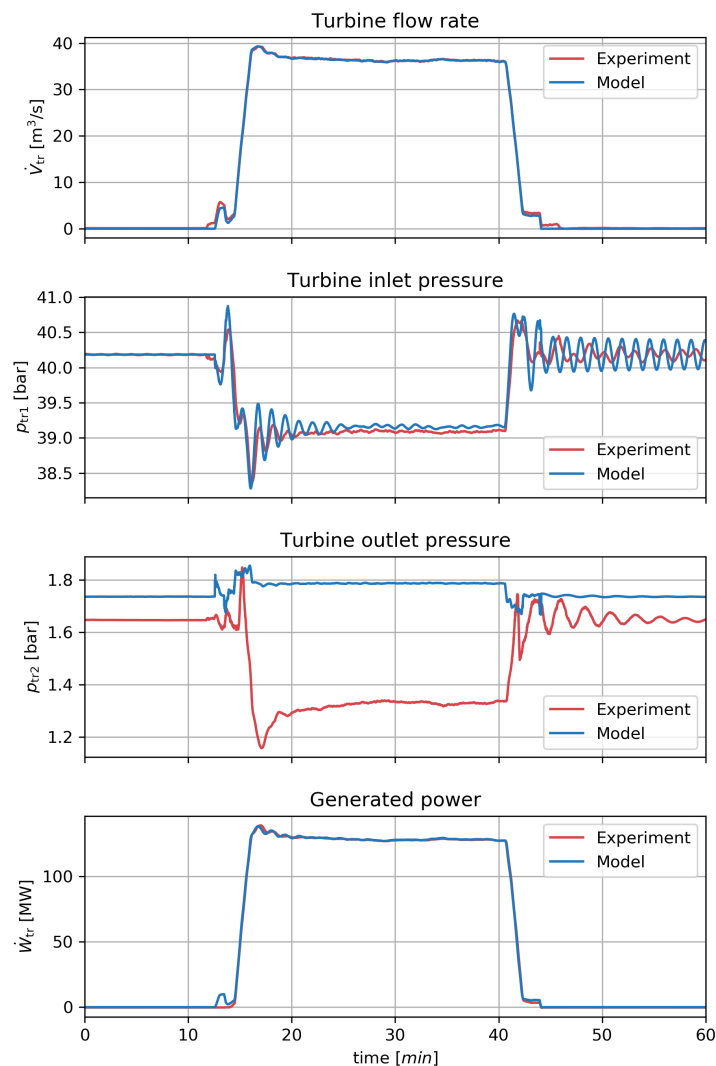


Figure 6. Comparison of the simple hydropower model simulation results and the experimental data.

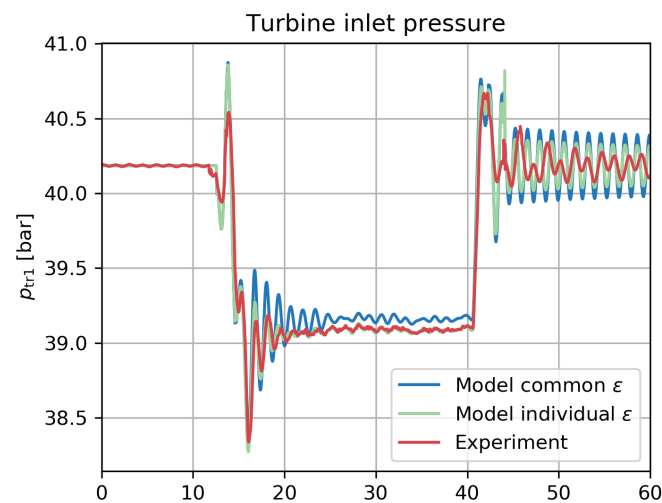


Figure 7. Comparison of the turbine inlet pressure from the simple model simulations with individual ε settings and experimental measurements.

4.3.2. Draft Tube Geometry: Initial Manual Tuning

As mentioned above, the draft tube is also used in this hydropower model. However, the geometry of the draft tube situated in the Trollheim hydropower plant is not known in detail, and the simple *Pipe* element is used in the model. With this pipe element, added geometric parameters are available as tuning parameters for the model. However, the geometry and shape of this *Pipe* element might not represent the real draft tube.

In the previous simulation of the hydropower model, Figure 6, the draft tube element was not been included in the model simulation. With the *Pipe* element of *OpenHPL*, it is possible to specify the *Pipe* element as an expanding circular vertical pipe. Thus, the length, the inlet diameter, the outlet diameter, and the roughness height of this pipe can be tuned. After some manual tuning of the geometry of the draft tube model, it has been observed that this unit significantly affects the turbine outlet pressure dynamic. The simulation results of the hydropower model with and without this draft tube are shown in Figure 8, where the comparison of the turbine outlet pressure from the measurements and the model simulations is provided.

From manual tuning, the draft tube pipe is designed with an expanding diameter from 2.2 m to 3.04 m, 12 m long, and with 0.001 mm roughness height. This pipe geometry deviates from what is expected in a real draft tube: especially the real draft tube shape [34] is much more complicated than the simple vertical expanded pipe used here. Figure 8 shows that the outlet pressure from the model with the simplified draft tube element fits the experimental data much better than the results from the model where the draft tube is not considered. This better fitting is also confirmed by the RMSE of the outlet pressure that has reduced from 0.315 bar for the model without draft tube to 0.038 bar for the model with draft tube.

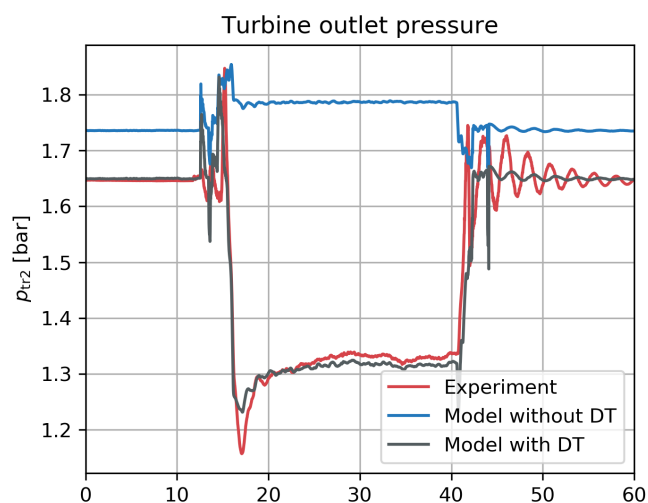


Figure 8. Comparison of the turbine outlet pressure from the simple model simulations with different draft tube (DT) settings and experimental measurements.

4.3.3. Pipe Roughness and Draft Tube Geometry: Least Squares Tuning

Starting with the initial parameter guesses from Sections 4.3.1 and 4.3.2, the least squares method is used to fine tune the model parameters. The results of the simple hydropower model with the parameters tuned by the least squares method are shown in Figure 9, where the four plots: the turbine flow rate, the turbine inlet and outlet pressures, and the generated power are presented. The tuning parameters found with the least squares error method are given in Tables 4 and 5 for the roughness heights and the draft tube geometry, respectively.

It is seen from Figure 9 that the tuned model provides good results that fit the measured data reasonably well. Some deviation in pressure oscillations is still visible, but the results look promising and are greatly improved in comparison to the results from the original model (Figure 6). This improvement is also confirmed by the root mean squared error between the model and experiment curves. The RMSE has been reduced for the turbine inlet pressure from 0.142 to 0.101 bar, and for the turbine outlet pressure from 0.315 to 0.04 bar. For the the turbine flow rate and the generated power, the RMSE has not been changed significantly.

Table 4. The waterway parameters tuned with the least squares error method.

Waterway Unit	Roughness Height, ϵ , (mm)
Intake race #1	0.01
Intake race #2	0.014
Intake race #3	0.59
Penstock #1	0.16
Penstock #2	$5 \cdot 10^{-5}$
Surge tank	$5 \cdot 10^{-5}$
Disch. race #1	0.027
Disch. race #2	0.01

Table 5. The draft tube pipe parameters tuned with the least squares error method.

Waterway Unit	Roughness Height, ϵ , (mm)	Expand Diameter, (m)	Length, (m)
Draft tube	$8.4 \cdot 10^{-4}$	2.18 → 3.01	12

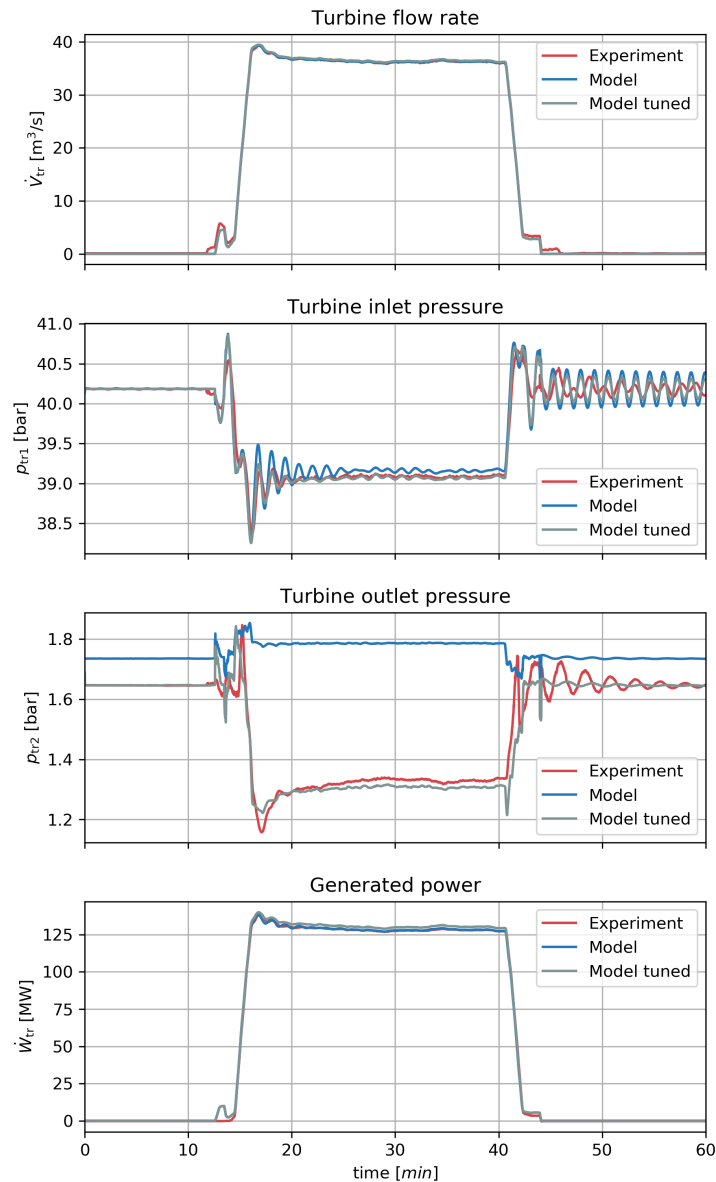


Figure 9. Comparison of the results from the tuned simple hydropower model (with tuned draft tube and roughness) and the experimental data.

4.4. Detailed Hydropower Models

Thus, far, the model tuning has been based on extending the simple model (model #1 in Section 3.2) with a draft tube. It is of interest to compare the experimental measurements with the simulation results from the other hydropower models (models #2–#4 in Section 3.2), with friction coefficient expressions from Section 4.2.2 and tuned parameters from Sections 4.3.3, where appropriate. For the cases involving a detailed penstock model, each of the *PenstockKP* elements (for two penstock units) is set with five discretization segments. The simulations are performed for the same operational conditions and the same scenario as during the experiment.

Comparison of the simulation results from these more detailed model cases with the measured data is shown in Figure 10. Similarly to the previous simple model simulations, Figure 10 provides four plots: the turbine flow rate, the turbine inlet and outlet pressures, and the generated power.

Figure 10 shows that the results from the more detailed hydropower models look similar and fit the experimental data relatively well. Some extra oscillations for the outlet pressure exist for the cases with the mechanistic Francis turbine model. These oscillations are caused by the turbine

model uncertainties when the zero load regime is reached. Moreover, the presented results from the more detailed hydropower models do not show a significant difference from the results with the simple model case (Figure 9). Because of this, we do not further tune the parameters in the more detailed models.

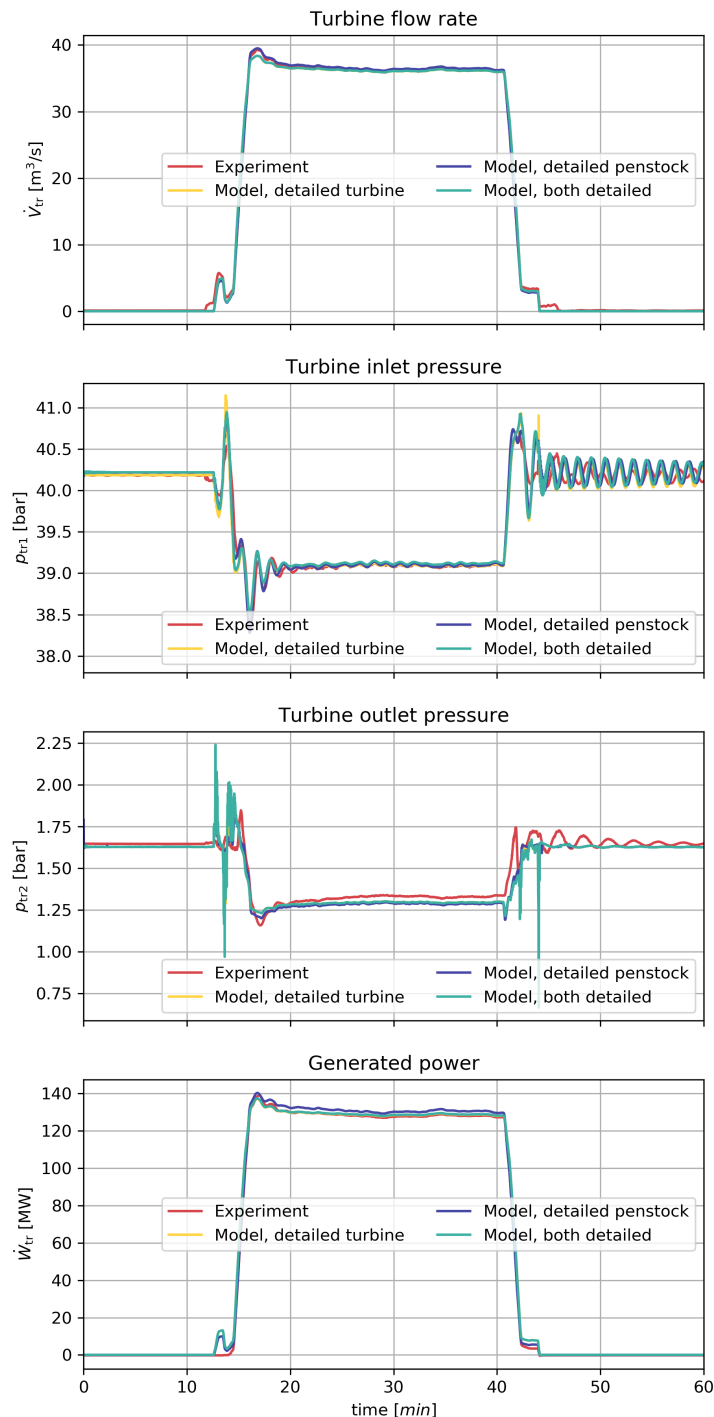


Figure 10. Comparison of the results from the detailed hydropower models and the experimental data.

5. Discussion

In this paper, the use of the open-source Modelica library *OpenHPL* for hydropower system modelling has been presented. *OpenHPL* provides models at different levels of complexity of hydropower system units which can be used to create a model of a hydropower system, depending on

the user's needs. Models of various complexity levels have been presented in this study for the case of the Trollheim hydropower plant. The experimental data from the hydropower plant have been presented and discussed.

The experimental data were first used for library element tuning, in tuning the friction loss coefficients of the Francis turbine model (the *Francis* element). Based on the resulting values of friction loss coefficients in this paper and in previous work, it was observed that there appears to be simple relationships between friction coefficients and the turbine nominal head. Expressions relating the nominal head and the friction coefficients have been proposed, which allows for an improvement of the Francis turbine design algorithm presented in [23]. With the friction coefficient expressions in Section 4.2, the turbine model can be completely specified by knowledge of nominal turbine head and nominal flow rate through the turbine. Further work to validate these expressions should be carried out.

Next, a simple model (table look-up turbine model, incompressible water, inelastic pipes) was extended with a simplified draft tube for the turbine, and it was demonstrated that, by tuning the friction roughness heights of the pipes and the geometry of the draft tube, good model fit to experimental data was achieved. Model fit was achieved by first manually tuning model parameters in order to gain an understanding of the importance of parameters, and next by using the manually fitted parameters as initial guesses in multiparametric least squares fitting. Tuning of the roughness height (and thereby the friction energy loss) affects the transient amplitude/oscillations while the geometry of the draft tube has strong implications for the turbine outlet pressure. Correct description of the turbine outlet pressure is essential for avoiding cavitation.

Finally, more detailed hydropower models have been equipped with the model parameters from the simple model and simulation results have been compared with the experimental data. The results indicate that these more complex models also exhibit good model fitting. Some minor deviation between the models and measured results exists, but this deviation is believed to be relatively insignificant. The fact that the model parameters from the simple model generalizes well to the more complex models indicates that the model variables/parameters can be given a physical interpretation.

Overall, the developed models using *OpenHPL* look promising and might find the following uses for hydropower systems:

- Modelling and simulation [21],
- State estimation/smart sensor that enables estimation of unmeasured states/parameters based on combining models and available measurements [37],
- Linearization for control design [28],
- Advanced control design and testing, e.g., model predictive control (MPC),
- Integration of waterway model with electric grid model by combination with power system libraries, e.g., Open-Instance Power System Library (*OpenIPSL* [38]),
- Large scale control analysis, for example, based on a graph theory.

6. Conclusions

At the outset, some research questions were posed. Based on the results in this paper, we have confirmed that, by proper tuning, *OpenHPL* is suitable for describing real hydropower plants with little prior plant knowledge. Except for the turbine outlet pressure, the original model gave realistic predictions of the hydropower plant performance. This implies that future work should be put into developing improved models for the draft tube, with good default geometry parameters based on nominal turbine height and flow rate. The developed expressions for Francis turbine friction coefficients should be tested further. In addition, *OpenHPL* should be extended with models for other turbine types.

Because *OpenHPL* is based on the multiphysics language Modelica, available libraries for modelling the electric grid make it possible to model and study regional hydropower systems, but also

to integrate the study with other, renewable energy sources when Modelica libraries for these other energy sources become available.

Integration of simulation tools such as OpenModelica with scripting languages such as Python and Julia allows for model based analysis and synthesis in a simple and straightforward way. Furthermore, by basing the tool chain on open source tools, this makes it simple to use these tools in education, and allows for contributions and extensions from the community of users.

Author Contributions: Both authors have significant contribution in the paper and more detailed individual contributions are specified here: conceptualization, L.V. and B.L.; methodology, L.V.; software, L.V.; validation, L.V.; formal analysis, L.V. and B.L.; investigation, L.V. and B.L.; writing—original draft preparation, L.V.; writing—review and editing, B.L.; visualization, L.V.; supervision, B.L.

Funding: This research received no external funding.

Acknowledgments: Erik Jacques Wiborg from Statkraft Energi AS, Oslo, Norway, provided the experimental and other related data from the Trollheim power plant for the case study. Bjarne Børresen from Multiconsult AS, Oslo, Norway helped put us in touch with Erik Wiborg. This is gratefully acknowledged.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Munoz-Hernandez, G.; Mansoor, S.; Jones, D. *Modelling and Controlling Hydropower Plants*; Advances in Industrial Control; Springer: London, UK, 2013.
- Nielsen, T.K. Transient Characteristics of High-Head Francis Turbines. Ph.D. Thesis, Norwegian University of Science and Technology, Trondheim, Norway, 1990.
- Yang, W.; Yang, J.; Guo, W.; Zeng, W.; Wang, C.; Saarinen, L.; Norrlund, P. A Mathematical Model and Its Application for Hydro Power Units under Different Operating Conditions. *Energies* **2015**, *8*, 10260–10275. [[CrossRef](#)]
- Fang, H.; Chen, L.; Dlakavu, N.; Shen, Z. Basic Modeling and Simulation Tool for Analysis of Hydraulic Transients in Hydroelectric Power Plants. *IEEE Trans. Energy Convers.* **2008**, *23*, 834–841. [[CrossRef](#)]
- Guo, W.; Yang, J.; Yang, W.; Chen, J.; Teng, Y. Regulation quality for frequency response of turbine regulating system of isolated hydroelectric power plant with surge tank. *Int. J. Electr. Power Energy Syst.* **2015**, *73*, 528–538. [[CrossRef](#)]
- Yang, W.; Norrlund, P.; Bladh, J.; Yang, J.; Lundin, U. Hydraulic damping mechanism of low frequency oscillations in power systems: Quantitative analysis using a nonlinear model of hydropower plants. *Appl. Energy* **2018**, *212*, 1138–1152. [[CrossRef](#)]
- Quiroga, O.; Riera, J.; Batlle, C. Identification of partially known models of the Susqueda hydroelectric power plant. *Lat. Am. Appl. Res.* **2003**, *33*, 387–392.
- Margonis, P. Modeling and Optimization of a Hydroelectric Power Plant for a National Grid Power System Supply. Case Study: Stratos Hydroelectric Dam. Ph.D. Thesis, Technical University of Crete, Chania, Greece, 2017.
- LVTrans. Open Source Transient Simulations Software for Hydraulic Piping Systems. Available online: <http://svingentech.no/index.html> (accessed on 14 June 2019).
- National Instruments. LabVIEW 2019. Available online: <http://www.ni.com/en-no/shop/labview/labview-details.html> (accessed on 14 June 2019).
- Sandvåg, S.U. Surge Tank Atlas for Hydropower Plants. Master's thesis, Department of Hydraulic and Environmental Engineering, Norwegian University of Science and Technology, Trondheim, Norway, 2016.
- Yang, W. Hydropower Plants and Power Systems: Dynamic Processes and Control for Stable and Efficient Operation. Ph.D. Thesis, Electricity, Uppsala University, Uppsala, Sweden, 2017.
- École Polytechnique Fédérale de Lausanne (EPFL). Simulation Software for Power Networks SIMSEN. Available online: <https://simсен.epfl.ch> (accessed on 14 June 2019).
- Nicolet, C. Hydroacoustic Modelling and Numerical Simulation of Unsteady Operation of Hydroelectric Systems. Ph.D. Thesis, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2007.
- Modelon. Hydro Power Library. Available online: <https://www.modelon.com/library/hydro-power-library> (accessed on 14 June 2019).

16. Modelon. System Simulation without Boundaries. Available online: <https://www.modelon.com> (accessed on 14 June 2019).
17. Dassault Systems. DYMOLA Systems Engineering. Available online: <https://www.3ds.com/products-services/catia/products/dymola> (accessed on 14 June 2019).
18. Winkler, D.; Thoresen, H.M.; Andreassen, I.; Perera, M.A.S.; Sharefi, B.R. Modelling and Optimisation of Deviation in Hydro Power Production. In Proceedings of the 8th International Modelica Conference, Linköping Electronic Conference, Dresden, Germany, 20–22 March 2011; Volume 1.
19. Tuszynski, K.; Tuszyński, J.; Slättorp, K. HydroPlant—a Modelica Library for Dynamic Simulation of Hydro Power Plants. In Proceedings of the 5th International Modelica Conference, Vienna, Austria, 4–5 September 2006; Volume 1.
20. OpenModelica. Available online: <https://openmodelica.org> (accessed on 14 June 2019).
21. Vytvytskyi, L.; Lie, B. Comparison of elastic vs. inelastic penstock model using OpenModelica. In Proceedings of the 58th Conference on Simulation and Modelling, Reykjavik, Iceland, 25–27 September 2017; Volume 138, pp. 20–28. [CrossRef]
22. Splavska, V.; Vytvytskyi, L.; Lie, B. Hydropower Systems: Comparison of Mechanistic and Table Look-up Turbine Models. In Proceedings of the 58th Conference on Simulation and Modelling, Reykjavik, Iceland, 25–27 September 2017; Volume 138, pp. 368–373. [CrossRef]
23. Vytvytskyi, L.; Lie, B. Mechanistic model for Francis turbines in OpenModelica. *IFAC-PapersOnLine* **2018**, *51*, 103–108. [CrossRef]
24. Modelica. The Modelica Association. Available online: <https://www.modelica.org> (accessed on 14 June 2019).
25. OpenModelica. OpenModelica Python Interface and PySimulator. Available online: <https://www.openmodelica.org/doc/OpenModelicaUsersGuide/latest/ompython.html> (accessed on 14 June 2019).
26. Python. Available online: <https://www.python.org> (accessed on 14 June 2019).
27. Lie, B.; Bajracharya, S.; Mengist, A.; Buffoni, L.; Kumar, A.; Sjölund, M.; Asghar, A.; Pop, A.; Fritzsøn, P. API for Accessing OpenModelica Models From Python. In Proceedings of the EuroSim 2016, Oulu, Finland, 12–16 September 2016.
28. Vytvytskyi, L.; Lie, B. Linearization for Analysis of a Hydropower Model using Python API for OpenModelica. In Proceedings of the 59th Conference on Simulation and Modelling, Oslo, Norway, 26–27 September 2018; Volume 153, pp. 216–221. [CrossRef]
29. Statkraft. Trollheim Hydropower Plant. Available online: <https://www.statkraft.com/energy-sources/Power-plants/Norway/Trollheim> (accessed on 14 June 2019).
30. Wiborg, E.J. Continuous Efficiency Measurements on Hydro Power Plants. Ph.D. Thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2016.
31. GitHub. Trollheim Experiment. Available online: <https://github.com/liubomyrv/TrollheimExperiment.git> (accessed on 14 June 2019).
32. Tiller, M.M. Modelica by Example. Available online: <http://book.xogeny.com> (accessed on 14 June 2019).
33. OpenModelica. User Documentation. Available online: <https://openmodelica.org/useresresources/userdocumentation> (accessed on 14 June 2019).
34. The National Programme on Technology Enhanced Learning (NPTEL). Francis Turbine. Available online: https://nptel.ac.in/courses/Webcourse-contents/IIT-KANPUR/machine/chapter_7/7_7.html (accessed on 14 June 2019).
35. SciPy. Numpy and Scipy Documentation. Available online: <https://docs.scipy.org/doc> (accessed on 14 June 2019).
36. EnggCyclopedia. Absolute Pipe Roughness. Available online: <https://www.enggcyclopedia.com/2011/09/absolute-roughness> (accessed on 14 June 2019).

37. Vytvytskyi, L.; Lie, B. Combining Measurements with Models for Superior Information in Hydropower Plants. *Flow Meas. Instrum.* **2019**, accepted.
38. GitHub. OpenIPSL: Open-Instance Power System Library. Available online: <https://github.com/OpenIPSL/OpenIPSL> (accessed on 14 June 2019).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Doctoral dissertation no. 37

2019

Dynamics and model analysis of hydropower systems

Dissertation for the degree of Ph.D

Liubomyr Vytvytskyi

ISBN: 978-82-7206-533-0 (print)

ISBN: 978-82-7206-534-7 (online)

usn.no

