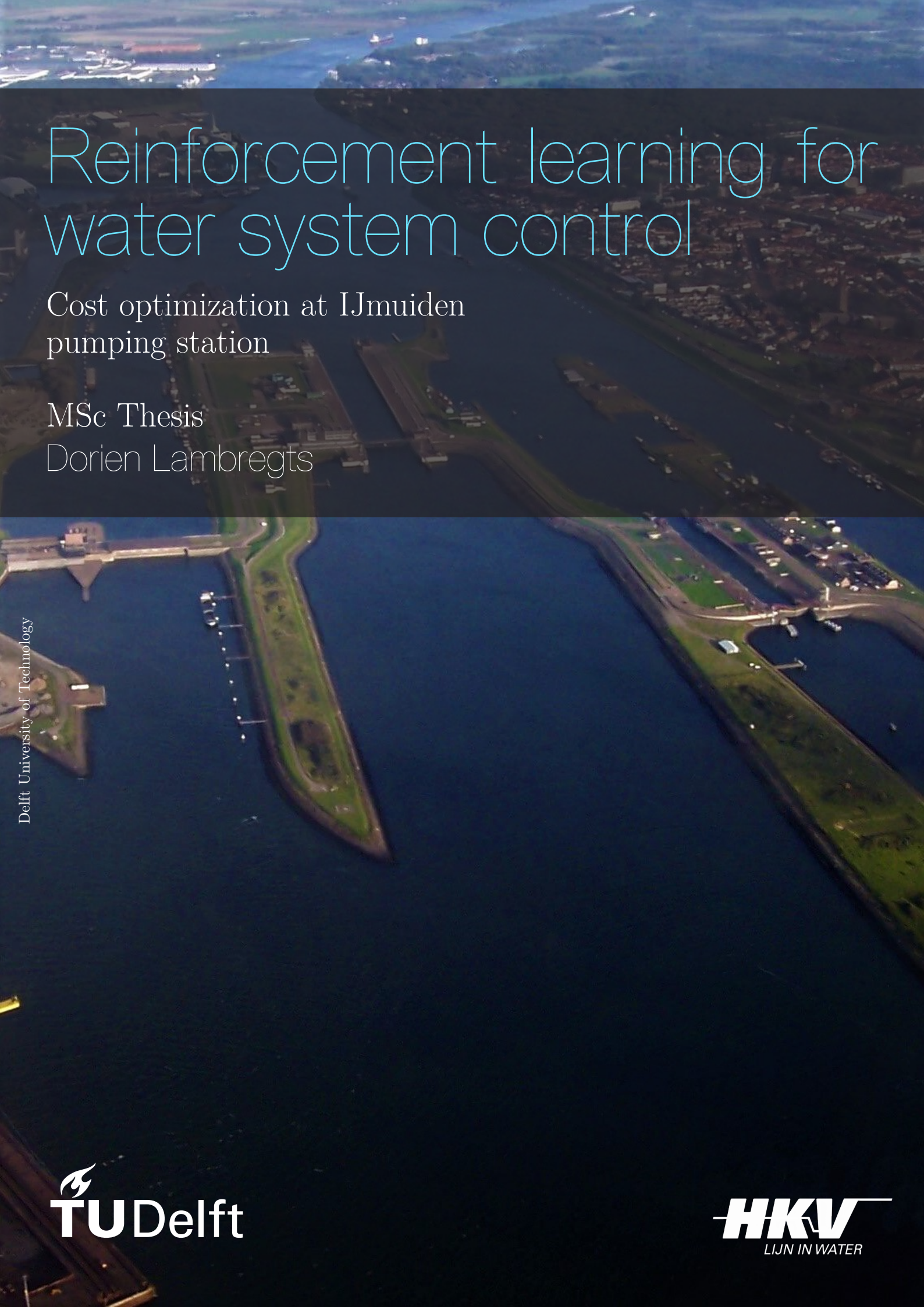


Reinforcement learning for water system control

Cost optimization at IJmuiden pumping station

MSc Thesis

Dorien Lambregts

Delft University of Technology

TU Delft

HKV LIJN IN WATER

Reinforcement learning for water system control

Cost optimization at IJmuiden pumping station

by

Dorien Lambregts

to obtain the degree of Master of Science in Civil Engineering
at the Delft University of Technology,
to be defended publicly on Tuesday, November 15th 2022, at 11:30.

Student number:	4570987	
Project duration:	February 2022 - November 2022	
Thesis committee:	Dr. Juan Pablo Aguilar-López	TU Delft (chair)
	Dr. ir. Edo Abraham	TU Delft
	Prof. dr. ir. Matthijs Kok	HKV Lijn in water, TU Delft
	Ir. Ties van der Heijden	HKV Lijn in water, TU Delft

A digital version of this thesis is available via repository.tudelft.nl

Cover image: An aerial photograph of the Zeesluis IJmuiden and IJmuiden pumping station, viewed from the North Sea towards the Noordzeekanaal. [1]

TU Delft

HKV LIJN IN WATER

Preface

This thesis will conclude my master's degree in Civil Engineering at Delft University of Technology and my time as a student. This research was conducted in cooperation with HKV Lijn in water. I can look back on a project that was challenging, educational, and extremely interesting, and where I was able to combine my passion for computer science with hydraulic engineering.

I would like to thank my daily supervisor, Ties, and Juan for their frequent guidance, encouragement, and enthusiasm throughout the entire project. Being able to reach out easily and discuss any issues I ran into helped me to keep moving forward. I would also like to thank Edo and Matthijs for their support and feedback, that gave a valuable perspective on the project.

I hope you enjoy your reading.

Dorien Lambregts
Delft, November 2022

Abstract

The production and consumption of electricity need to be balanced at all times. Due to the ever-growing shift towards renewable energy generation, this poses an increasingly difficult challenge. Currently, supply is regulated to maintain balance. However, there is potential to improve reliability and save costs by shifting the balancing to the demand side, known as demand response. The flexibility of water systems can play a role in this, thereby benefiting from cheaper price fluctuations and reducing operating costs.

This research investigates the IJmuiden pumping station, which drains water from the Noordzeekanaal-Amsterdam-Rijnkanaal system into The North Sea. The primary focus of the control of this system is ensuring safe water levels as it runs through areas of high economic value. The flexibility of the range of safe water levels allows costs to be minimized by selecting favourable moments to consume electricity. This simultaneously contributes to the stability of the electrical grid. This research explores the potential for a Reinforcement Learning controller for such an optimization problem, as there are some drawbacks to the Model Predictive Control methods that are currently widely used. The research objective is formulated as follows:

To optimize the control of the IJmuiden pumping station using Reinforcement Learning while complying with local water level restrictions and compare it to the state-of-the-art Model Predictive Control methods in terms of constraint violation, energy costs, and computational speed.

The Reinforcement Learning controller will use a deep Q-learning algorithm that chooses the most cost efficient control in IJmuiden while respecting the water level restrictions. To do so, the model makes decisions based on electricity prices and details about the state of the water system for the current time step as well as a forecast of 48 hours ahead. This data is provided as an input to the model.

The inputs of the model consist of historical data, meaning that the associated uncertainties are not included. The water system that the model can interact with is represented by a linear reservoir model. Therefore, the water system is influenced dynamically by the actions taken by the model. The possible actions are determined by the state of the water system.

The trained model was tested on 2 years of unseen data (data that was not used during training). Using the same test data, control plans were generated using Model Predictive Control. The Reinforcement Learning model was very successful in ensuring safe water levels. However, this did result in approximately 50% higher energy costs. The use of the gate was close to optimal but the pumping was not clearly correlated with favourable prices and power consumption. The trained model was robust, with consistently accurate results with regards to respecting the water level constraints.

The most significant difference with the Model Predictive Control was the computation time. The Reinforcement Learning model was able to create a control plan approximately 300 times faster. This opens doors for further development of the model and increased complexity. A more accurate model of the water system can be used to take into account temporal and spatial effects and individually representing the six pumps in IJmuiden.

There are still many steps before such a model can be used for operational control, but the method has potential for such an application. Many aspects of the model can be improved as well as making adjustments to increase the usability for control operators.

Contents

Preface i

Abstract ii

Nomenclature vi

List of Figures vii

List of Tables x

1 Introduction 1
- 1.1 Context . . . 1
- 1.2 Case study . . . 2
- 1.3 Possible control methods . . . 2
  - 1.3.1 Proportional Integral Derivative . . . 3
  - 1.3.2 Model Predictive Control. . . . 3
  - 1.3.3 Machine learning . . . 3
- 1.4 Research objective and questions . . . 4
- 1.5 Thesis outline . . . 5

2 Case Study 6
- 2.1 Water level regime NZK-ARK . . . 8
- 2.2 IJmuiden gate. . . . 8
- 2.3 IJmuiden pumping station . . . 8
- 2.4 Wind set-up. . . . 9
- 2.5 Current control . . . 10

3 Electricity Markets 11
- 3.1 Day-ahead market . . . 11
- 3.2 Intraday market. . . . 12
- 3.3 Suitable markets for water system control. . . . 13

4 Deep Reinforcement Learning 14
- 4.1 Reinforcement learning concepts. . . . 14
  - 4.1.1 Markov Decision Process . . . 15
  - 4.1.2 Reinforcement learning methods. . . . 15
- 4.2 Deep reinforcement learning algorithms . . . 17
  - 4.2.1 Common solution components . . . 17
  - 4.2.2 Popular algorithms . . . 18
  - 4.2.3 Deep Q-network . . . 20
- 4.3 Hyperparameter optimization . . . 23
  - 4.3.1 Bayesian Model-Based Optimization . . . 23

5 Water System Model 25
- 5.1 NZK-ARK system . . . 25
- 5.2 IJmuiden pumping station . . . 26
  - 5.2.1 Surrogate model - pumping station . . . 26
  - 5.2.2 Surrogate model - gate . . . 28
- 5.3 Wind set-up. . . . 29

6 Methodology **30**
6.1 Reinforcement learning method . . . 30
6.1.1 State space . . . 31
6.1.2 Action space . . . 33
6.1.3 Reward structure . . . 34
6.2 Input data . . . 36
6.2.1 Train, validation, test split . . . 36
6.3 Learning procedure . . . 37
6.4 Hyperparameters . . . 38
6.5 Training speed . . . 39

7 Results **40**
7.1 Understanding the control plan . . . 41
7.2 Water level objective . . . 42
7.3 Cost objective . . . 44
7.3.1 Overall model performance . . . 44
7.3.2 Normal conditions . . . 46
7.3.3 High inflowing discharge . . . 49
7.3.4 Operational control . . . 50
7.4 Comparison with Model Predictive Control . . . 51
7.5 Extreme scenarios . . . 54
7.6 Suitable reinforcement learning algorithms . . . 56
7.6.1 High inflow scenario . . . 56
7.7 Alternative reward structure . . . 58

8 Conclusion and Discussion **60**
8.1 Suitable reinforcement learning algorithms . . . 60
8.2 Water level constraints . . . 61
8.3 Cost optimization . . . 61
8.3.1 Water level constraints . . . 61
8.3.2 Cost reduction . . . 62
8.4 Extreme scenarios . . . 62
8.5 Overall reinforcement learning model . . . 63
8.5.1 Operational control . . . 63
8.5.2 Water system model . . . 63
8.5.3 RL training . . . 63
8.5.4 Alternative reward structure . . . 63
8.6 Final conclusion . . . 64

9 Recommendations **65**
9.1 Water system model . . . 65
9.1.1 NZK-ARK . . . 65
9.1.2 Pumping station model . . . 65
9.1.3 Time resolution . . . 65
9.1.4 Operational control . . . 66
9.2 Reinforcement learning method . . . 66
9.2.1 Algorithms . . . 66
9.2.2 State space . . . 67
9.2.3 Training steps . . . 68
9.3 Combination with MPC . . . 68

References **73**

A Power Consumption Optimization **74**
A.1 Optimization model . . . 74
A.2 MIP results . . . 75
A.3 Fitted power consumption . . . 76

B Wind Set-Up **77**

C Simulated Discharges For Training **78**

D Train, Validation, Test Split **80**
D.1 Data clustering . . . 80
D.2 Data split . . . 82

E Test Scenarios **83**
E.1 Discharge scenarios . . . 83
E.2 Sea level scenario . . . 83
E.3 Electricity price scenarios . . . 84

F Hyperparameter optimization **85**

G Snellius Job Script **89**

H Additional Results - Water Level Objective **90**

I Additional Results - Cost Objective **91**
I.1 Largest water level exceedance control plans . . . 91
I.2 MPC control plan - normal conditions . . . 93
I.3 MPC control plan - high inflow conditions . . . 94

J Additional Results - Test Data Set Measurements **95**

K Additional Results - Test Scenarios **96**
K.1 Extreme Q . . . 96
K.2 High Q . . . 98
K.3 Low Q . . . 100
K.4 High sea . . . 102
K.5 High sea high Q . . . 104
K.6 High E . . . 106
K.7 Negative E . . . 108
K.8 Extreme negative E . . . 109

L Additional Results - Suitable RL Algorithms **111**
L.1 Control plans for 2020-02-18 . . . 111

M Additional Results - Alternative Reward **118**
M.1 High water level - 2019-12-17 . . . 118
M.2 High inflow discharge - 2019-03-05 . . . 119
M.3 Extreme Negative E . . . 120

Nomenclature

Abbreviations

A2C	Advantage Actor Critic
A3C	Asynchronous Advantage Actor Critic
AC	Actor Critic
ACER	Actor Critic with Experience Replay
ARK	Amsterdam-Rijnkanaal
BRP	Balance responsible party
DAM	Day-ahead market
DDPG	Deep Deterministic Policy Gradient
DDQN	Double Deep Q-network
DNN	Deep neural network
DP	Dynamic programming
DQN	Deep Q-network
DRL	Deep reinforcement learning
GAE	Generalized Advantage Estimate
IDM	Intraday market
MC	Monte Carlo
MCP	Market clearing price
MDP	Markov Decision Process
MIP	Mixed Integer Programming
MPC	Model Predictive Control
NFQ	Neural Fitted Q Iteration
NN	Neural network
NZK	Noordzeekanaal
PID	Proportional Integral Derivative
PPO	Proximal Policy Optimization
RL	Reinforcement learning
SMBO	Sequential Model-Based Optimization
TD	Temporal-difference
TPE	Tree Parzen Estimator
TRPO	Trust Region Policy Optimization
TSO	Transmission system operator

List of Figures

1.1 The course of the NZK-ARK including important urban areas . . . 2
1.2 Types of machine learning and their applications . . . 4
1.3 Flow chart of the main elements of the proposed methodology . . . 5

2.1 NZK-ARK system including important structures and urban areas . . . 6
2.2 Aerial view of the IJmuiden complex with the important structures . . . 7
2.3 Important water levels in the NZK . . . 8
2.4 An example of a pump that is installed at the pumping station of IJmuiden [24, 25] . . . 9
2.5 A schematization of a discrete MPC scheme [27] . . . 10

3.1 The types of electricity markets with their corresponding time frames . . . 11
3.2 Regular DAM prices in The Netherlands, 01-02 January 2022 [41] . . . 12
3.3 Negative DAM prices in The Netherlands, 15-16 July 2022 [41] . . . 13

4.1 Interaction between agent and environment for a Markov Decision Process . . . 14
4.2 Width and depth of backups for main RL methods . . . 16
4.3 Schematization of a Deep Q-network . . . 21
4.4 DQN loss function with components from the target and prediction network . . . 21
4.5 Grid and random search with nine trials for optimizing a function $f(x, y) = g(x)+h(y) \sim g(x)$ . . . 23
4.6 Tree Parzen Estimator; (a) objective function scores for hyperparameter values; (b) probability distributions for hyperparameters above and below the threshold score . . . 24

5.1 Linear reservoir model of the NZK-ARK . . . 25
5.2 *Q-dH* relationship for pumps showing the feasible workspace . . . 26
5.3 MIP results for the power consumption of the optimal pump configuration . . . 27
5.4 Fitted *P-Q-dH* relationship for the MIP results for the power consumption of the optimal pump configuration . . . 27
5.5 Gate schematization for the calculation of the maximum discharge . . . 28
5.6 *Q-dH* relationship for gate showing the feasible workspace during regular and high water conditions . . . 28
5.7 Basin schematization for wind set-up . . . 29

6.1 RL agent interaction with the environment and reward function . . . 30
6.2 Discharge forecast for the state space (Dataset from [65]) . . . 32
6.3 Location of North Sea level data . . . 32
6.4 Sea level forecast for the state space (Dataset from [65]) . . . 32
6.5 Action discretization for the pumping station and gate . . . 34

7.1 RL control plan for three days starting on 2020-02-25 . . . 41
7.2 RL control plan for two weeks starting on 2019-06-25 with only the water level objective 43
7.3 Hourly water levels from historical measurements, RL model, and MPC for high inflow event on 2020-02-23. Models were run from 2020-02-18 with an initial water level equal to the measurements. . . . 45
7.4 Agent performance improvement during training with the percentage out of target range for the training set which included exploration and validation set . . . 45
7.5 RL control plan for two weeks starting on 2019-06-25 where no pumping is necessary . . . 46
7.6 RL control plan for two weeks starting on 2019-12-17 where the water level was in the upper part of the target range due to higher sea levels . . . 47
7.7 Zoom into two days (2019-12-19 - 2019-12-21) of Figure 7.6 . . . 48

7.8 RL control plan for two weeks starting on 2019-03-05 with high inflowing discharges . . 49
7.9 The resulting water levels for the RL and MPC control plan for test data set, 2019-01-01 - 2021-01-01. The probability density curve is shown in the right hand figures. 52
7.10 Probability density curves for the water levels during the test data set for the RL model, MPC, and historical measurements. The zoom shows the distribution when exceeding the upper target boundary. 52
7.11 The % of time steps in the test data set where a water level occurred between bins of $0.005m$ 53
7.12 Cost of the RL and MPC control plans for all test scenarios with the mean over all scenarios 54
7.13 Percentage out of target water level range with the RL and MPC control plans for all test scenarios with the mean over all scenarios 55
7.14 Comparison of water levels for control plans of multiple RL algorithms for the high water scenario of 2020-02-18 57
7.15 The resulting water levels for the RL with the alternative reward structure for the test data set, 2019-01-01 - 2021-01-01. The probability density curve is shown in the right hand figure. 58
7.16 Figure 7.10 including results from the alternative reward structure showing the probability density curves for the water levels during the test data set. The zoom shows the distribution when exceeding the upper target boundary. 59

A.1 MIP results for the power consumption of the optimal pump configuration 75
A.2 Fitted *P-Q-dH* relationship for the MIP results for the power consumption of the optimal pump configuration 76

C.1 Inflowing discharge distribution 78
C.2 Inflowing discharge change distribution 79
C.3 Simulated inflowing discharges examples 79

D.1 Clustered kernel densities per month wet/dry 80
D.2 Clustered kernel densities per month cheap/expensive 81
D.3 Timelines for wet/dry and cheap/expensive dataset split 81
D.4 Timeline of the train, validation, test split for data per country 82

E.1 Discharge scenarios of 1 week for testing model performance 83
E.2 Sea level scenario of 1 week for testing model performance 84
E.3 Electricity price scenarios of 1 week for testing model performance 84

F.1 Hyperparameters optimization showing the loss for each parameter value and a cross for the best performance found. Trials with a loss above 50 were not included. The loss was set to be the negative reward, which was minimized in the optimization. 86
F.2 Hyperparameters optimization showing the training time for each parameter value and a cross for the best performance found. Trials with a training time above 35 minutes were not included. A linear fit gives an indication of the effect of the parameter value on the training time. 87
F.3 Hyperparameters optimization loss for batch size and update frequency. 88
F.4 Hyperparameters optimization loss for batch size and memory. 88

H.1 The RL control plan for two weeks starting on 2020-03-03 with only the water level objective. 90

I.1 The RL control plan for 14 days starting on 2020-02-18. 91
I.2 The MPC control plan for 14 days starting on 2020-02-18. 92
I.3 The MPC control plan for two days starting on 2019-12-19. 93
I.4 The MPC control plan for two weeks starting on 2019-03-05. 94

J.1 The historical measurements of the water level in the NZK at IJmuiden for the entire test data set. 95

K.1 The RL control plan for the extreme discharge scenario. 96
K.2 The MPC control plan for the extreme discharge scenario. 97
K.3 The RL control plan for the high discharge scenario. 98
K.4 The MPC control plan for the high discharge scenario. 99
K.5 The RL control plan for the low discharge scenario. 100
K.6 The MPC control plan for the low discharge scenario. 101
K.7 The RL control plan for the high sea level scenario. 102
K.8 The MPC control plan for the high sea level scenario. 103
K.9 The RL control plan for the high sea level and high discharge scenario. 104
K.10 The MPC control plan for the high sea level and high discharge scenario. 105
K.11 The RL control plan for the high electricity price scenario. 106
K.12 The MPC control plan for the high electricity price scenario. 107
K.13 The RL control plan for the negative electricity price scenario. 108
K.14 The RL control plan for the extreme negative electricity price scenario. 109
K.15 The MPC control plan for the extreme negative electricity price scenario. 110

L.1 The RL control plan using DQN for 10 days starting on 2020-02-18. 111
L.2 The MPC control plan for 10 days starting on 2020-02-18. 112
L.3 The RL control plan using PPO for 10 days starting on 2020-02-18. 113
L.4 The RL control plan using TRPO for 10 days starting on 2020-02-18. 114
L.5 The RL control plan using Dueling DQN for 10 days starting on 2020-02-18. 115
L.6 The RL control plan using AC for 10 days starting on 2020-02-18. 116
L.7 The RL control plan using A2C for 10 days starting on 2020-02-18. 117

M.1 The RL alternative reward control plan for two weeks starting on 2019-12-17. 118
M.2 The RL alternative reward control plan for two weeks starting on 2019-03-05. 119
M.3 The RL alternative reward control plan for the extreme negative energy prices scenario. 120

List of Tables

2.1 Dimensions of the NZK-ARK [19, 20, 18, 21] 7
2.2 Overview of pumps at IJmuiden pumping station [24] 9
2.3 Pump discharge and power relationships for all six pumps in the IJmuiden pumping station [26] 9

4.1 Overview of Deep reinforcement learning algorithms and their main components [8] . . . 20

5.2 Parameters of the gate for calculation of the maximum discharge [67] 28
5.4 Fetch in the NZK for the important wind directions 29

6.1 Mean and standard deviation used for z-score normalization of input parameters 33
6.5 Agent training steps 38
6.6 Comparison of model coefficients, available observations, and training observations . . . 39

7.1 RL and MPC control plan performance for two weeks starting on 2019-03-05 50
7.2 RL and MPC control plan performance for test data set, 2019-01-01 - 2021-01-01, with an indication of the current control derived from water level measurements. 51
7.3 RL and MPC performance for extreme scenarios of 1 week 54
7.4 Performance of multiple RL algorithms on the test data set using DQN hyperparameters 56
7.5 Performance of multiple RL algorithms for the high water scenario of 2020-02-18 (10 days) 57
7.6 Performance of alternative reward structure 58

A.1 Pump discharge and power relationships for all six pumps in the IJmuiden pumping station [26] 74

F.1 Hyperparameters tuned for the water level objective including the search space, best performing value, and chosen value 85

1 Introduction

1.1. Context

The current global energy system is environmentally unsustainable, which has incentivized the transition towards newer and cleaner energy [2]. Climate change mitigation is the most prominent driver behind this energy transition, however, other major societal benefits are expected as well [3]. Adapting infrastructure to meet the new demands and transitioning to a low-carbon economy poses one of the greatest challenges of our times [4].

Renewable energy sources behave differently to many of the non-renewables with which we are very familiar and to which the energy system has been adapted over many years. Energy generation from renewable sources often exhibits seasonality and unpredictability out of sync with the consumption of energy, thus leading to an imbalanced electrical grid. Electricity generation and consumption need to be balanced at all times. Preventing an imbalance will become an increasingly complex task as the energy system shifts to more sustainably generated energy [5]. Many types of infrastructure can play a role in reducing imbalances as well as reducing emissions and overall energy consumption.

In the past, the electricity network was balanced by adjusting the supply to match demand. This was done by power plants regulating the amount of energy produced. They increase production when sustainable energy sources are unavailable or demand increases. In the future, the portion of renewable sources will only increase, making balancing on the supply side only more complex. There is potential to improve reliability and save costs for electricity systems by shifting the balancing to the demand side [6]. Consumers can use more when supply exceeds demand and vice versa, known as demand response.

The focus of this thesis is the Dutch water management system and the possibilities of system control by taking into account classical constraints, such as water level while minimizing its energy consumption. At the moment, this water management system's primary focus is safety, which it provides by controlling pumps which in turn control the water levels at all scales (e.g. major rivers, groundwater, city canals). In doing so, the system consumes large amounts of energy, approximately 10 million kWh yearly with a cost of 700,000 euros, equivalent to more than 3,000 households [7]. The current strategy is to drain excess rainfall as quickly as possible to the sea. The operators take into account the moments of low head, however, this is still limited. Developments in the field of system control engineering allow for more complex objectives and constraints. This would mean the control system can respect the bounds set by safety regulations, while at the same time optimizing for energy consumption and stability of the electrical grid.

The consumption and production of electricity determine the price, where a higher demand results in increased prices and vice versa. Minimizing energy costs by taking advantage of price fluctuations therefore not only decreases operational costs for a system but also contributes to balancing the electrical grid.

A promising method to deal with these control problems is artificial intelligence [8], which is being applied more and more often in the water sector for forecasting and classification. Reinforcement learning (RL) is a type of learning where a control strategy is developed that learns from its interactions with its environment to better optimize its objectives in the future. A famous application of this is AlphaGo, which was the first computer program able to defeat a Go world champion [9]. Due to the

game's complexity, it is an extremely challenging problem for artificial intelligence. There are also more recent breakthroughs, for example, where RL was used for the control of a nuclear fusion experiment [10]. This highlights the wide range of possible applications and the successes that have been achieved using this method.

1.2. Case study

To explore the potential for a RL controller in a real-world context, a case study was used to conduct this research. The IJmuiden pumping station uses pumps and a gate to drain the water from the Noordzeekanaal (NZK)-Amsterdam-Rijnkanaal (ARK) system into the North Sea. Figure 1.1 shows the course of the water system, with the ARK flowing from Tiel to Amsterdam, after which it flows into the NZK that ends in the North Sea at the IJmuiden pumping station.

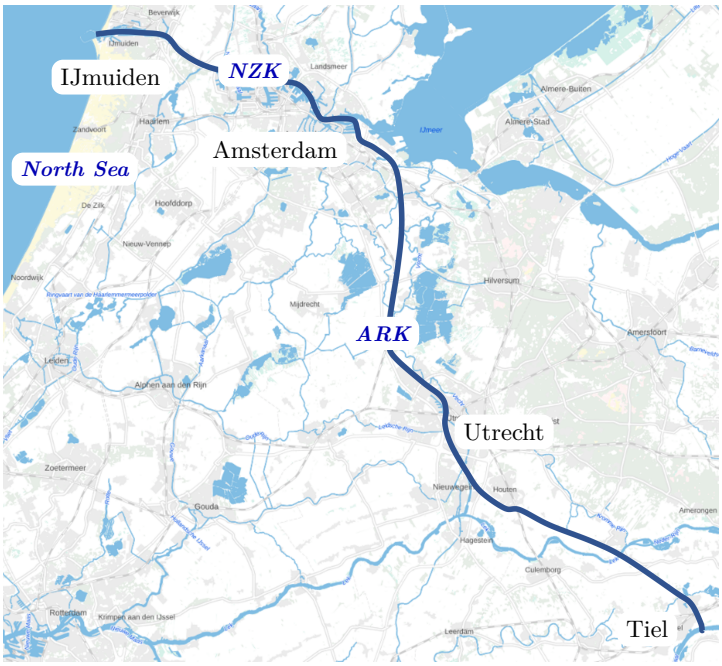


Figure 1.1: The course of the NZK-ARK including important urban areas

Rijkswaterstaat is responsible for managing the NZK-ARK system, the main element of which is the IJmuiden pumping station [11]. The system plays an important role in the freshwater supply and flood safety in the west of The Netherlands [12]. Given that The Netherlands is a low-lying country with a lot of water, and the west is a densely populated area, flood safety is of critical importance as well as creating a robust system that contributes to overall climate resilience.

Maintaining safe water levels is paramount as the system runs through two areas of high economical value; Amsterdam and Utrecht. Important infrastructure such as Schiphol airport, ports, and data centres are located within this region. This highlights an important criterion for a control system of the region, which is to respect the bounds that are set by the safety regulations.

On the other hand, there is a safe range within which the system can be operated, which lends itself to optimization with an additional objective; minimizing costs. Due to the high yearly cost of operation, even a small decrease in energy consumption could yield major economic benefits. In addition, the pump system is relatively flexible and can take advantage of electricity price fluctuations caused by imbalances in the grid. This lowers the operational costs while simultaneously contributing to the stability of the electricity grid.

1.3. Possible control methods

Currently, Model Predictive Control (MPC) is used for the IJmuiden pumping station, which includes optimization for energy costs. However, the electricity is bought on the futures market, which means that bids need to be submitted at least a month ahead of time. The nature of this electricity market

greatly limits the potential for cost optimization. Closer to delivery, the price fluctuations create temporarily lower prices that are more economical that the prices on the futures market.

When exploring the potential for energy cost optimization with electricity markets that allow trading very close to delivery, there are several possible methods. Proportional Integral Derivative (PID) and MPC are mature control systems that are widely used today while Machine Learning has not yet reached that status. However, it is currently a very active field of research because it enables real-time control with complex objective functions.

1.3.1. Proportional Integral Derivative

As mentioned above, PID is a control method that is universally used in applications where optimized automatic control is required. It is a means of controlling process variables, such as temperature, flow, and pressure. The system is forced in the direction of an objective using a control loop mechanism. Proportional tuning corrects for the difference between the target and measurement, known as the error, while integral tuning accounts for past errors to eliminate the residual error. Finally, the derivative tuning has a damping effect to prevent overshooting. [13, 14]

The limitation of PID control is that the feedback system works with constant parameters without knowledge of the dynamics of the system. These controllers have poor performance when the dynamics contain non-linearities and may have a delayed response to large disturbances. For these reasons, a PID controller is not suitable for the complex water system in combination with the energy costs optimization objective. [15]

1.3.2. Model Predictive Control

MPC is a popular method that can optimize the control for a pre-defined cost function while ensuring that the system constraints are satisfied [16]. MPC unlike PID has the predictive ability to anticipate future events. The control is optimized for a finite time horizon. This is a very suitable method for the control in IJmuiden as it ensures that the safety regulations are respected by constraining the system. The drawback of such a method, however, is the computation time and the limits to the complexity of the system.

In order to compute the optimal solution rapidly, the system is often simplified to reduce the complexity of the optimization problem. With these simplifications, the computations can still be time-consuming. On the other hand, as the system dynamics are taken into account, the MPC finds the optimal solution given its inputs.

1.3.3. Machine learning

Machine learning is being increasingly used to solve real-world problems in extremely varying applications. Different methods are suitable for specific types of problems and are then able to analyze data and improve through the use of data and experience. Three main types of machine learning can be identified: supervised learning, unsupervised learning, and RL. These techniques are illustrated in Figure 1.2. In supervised learning, the problem is solved by learning a mapping between inputs and outputs. This is done by training on known input examples with targets, which is suitable for problems such as fraud detection or customer segmentation. In unsupervised learning, the model aims to describe or discover relationships in the data without prior knowledge. Using this method, you are often working on problems such as speech analysis or image recognition. In both cases, there is a data set available and you have an understanding of how to solve the problem.

RL is an increasingly popular technique when dealing with large and complex problem spaces. There is knowledge about what the system should do, but you want to optimize or automate a specific process. In RL an agent (the controller) interacts with an environment and learns to operate through the feedback on its actions.

One of the advantages of RL algorithms is that it is a suitable method for solving systems that include non-linear dynamics that are unknown and/or are affected by large uncertainties [8]. Since the NZK-ARK with the IJmuiden pumping station has these properties, RL is a method with the potential for a controller.

Accurate forecasts far into the future are difficult to make and therefore decisions need to be made based on current expectations. There are many possible actions when controlling the water system resulting in a large problem space. Finally, one of the drawbacks of MPC methods is the computation time, which can be significantly reduced with the use of RL. The trade-off is that a RL model needs to

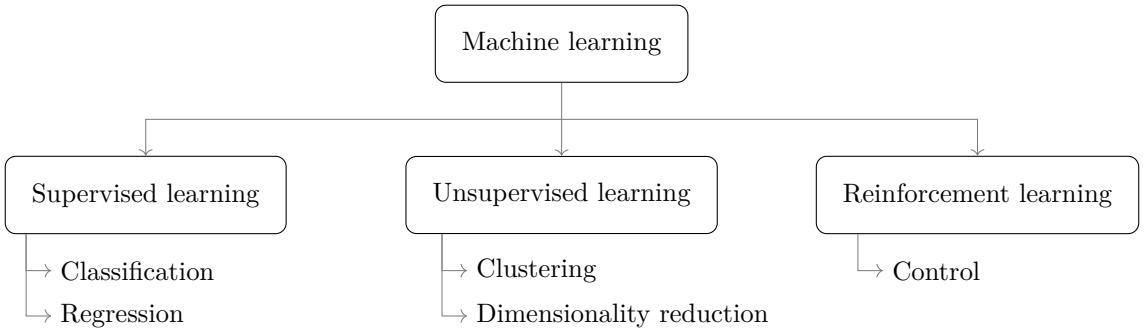


Figure 1.2: Types of machine learning and their applications

be pre-trained, which takes computation time of a similar order to the MPC evaluation time. However, evaluating the control strategy using a pre-trained model can be done in real-time.

1.4. Research objective and questions

In this research, the potential of RL as a water system controller is investigated. This is applied to the NZK-ARK system that ends at the IJmuiden pumping station where the safety regulations should be respected while optimizing for energy consumption. The objective of this research is formulated as follows:

To optimize the control of the IJmuiden pumping station using RL while complying with local water level restrictions and compare it to the state-of-the-art MPC methods in terms of constraint violation, energy costs, and computational speed.

An RL model will be developed, using Tensorforce [17], that receives the state of the water system and forecasts of discharge, sea levels, and energy prices to determine the optimal control plan. The forecasts and energy prices will be based on historic data. The water system itself will need to be modelled to facilitate the dynamic nature of a control system. This will be done with a linear reservoir model that evaluates the influence of specific actions on the state of the system. The RL model will be set up to ensure that water safety is never compromised to save costs.

In order to achieve the research objective, several research questions will be answered. Firstly, it is important to select an appropriate RL algorithm as there are many methods which are suitable for varying types of systems and tackle different issues that can arise when training an RL agent.

Which RL algorithms are suitable for controlling a water system such as the IJmuiden pumping station?

After selecting the algorithm, the model is first developed to deal with the classical constraints for the water level before minimizing the costs. Before the additional complexity of cost optimization, the model should be capable of creating a control plan to ensure the safety of the system.

How can RL deal with the water level restrictions when controlling a water system?

When sufficient performance is achieved, the cost objectives can be included in combination with the requirements for the water level.

How can RL optimize control for objectives regarding energy costs while respecting the water system constraints?

Finally, the RL model needs to be compared to current state-of-the-art MPC controllers to determine how promising the method is for use in water system control. In addition, the two models will be compared for several extreme scenarios. Throughout this research, it is also important to consider what the (dis)advantages are of RL-based control.

How does an RL controller cope with extreme scenarios in the water system compared to MPC?

1.5. Thesis outline

This thesis aims to achieve the research objective, by first introducing the system to be controlled. The IJmuiden pumping station and electricity markets in The Netherlands will be introduced in Chapters 2 and 3 respectively. The background of RL methods follows in Chapter 4. This allows a model of the water system to be made that will be used by the controller, described in Chapter 5. Chapter 6 explains the details of the RL method used for the control optimization, the main elements of which are shown in Figure 1.3. The results of the model are presented in Chapter 7. Finally, the conclusions drawn and recommendations for further research are given in Chapters 8 and 9.

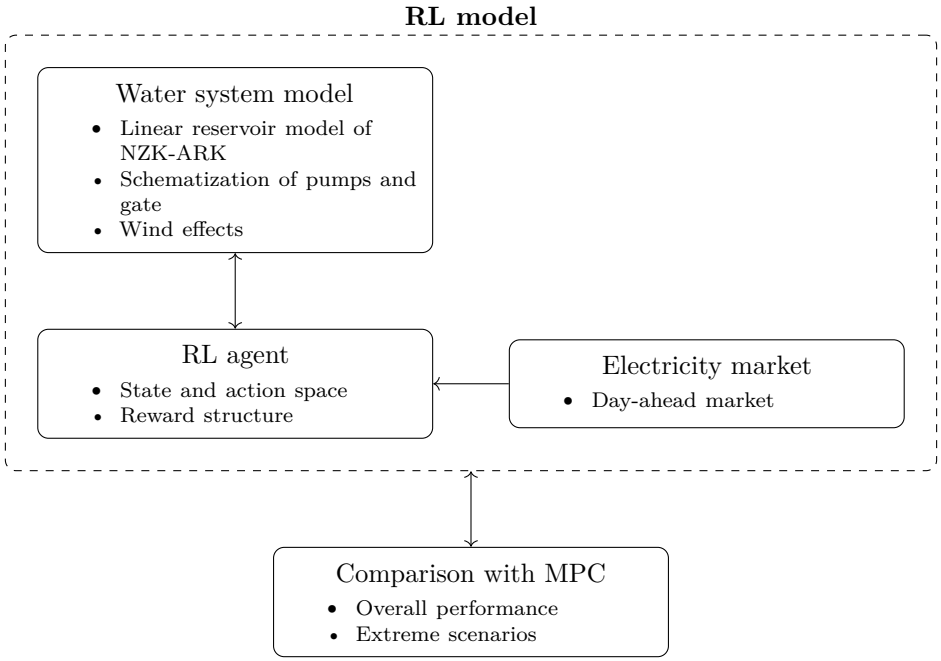


Figure 1.3: Flow chart of the main elements of the proposed methodology

2

Case Study

To conduct this research project, a case study was used to explore the potential for reinforcement learning (RL) in a real-world context. The water system that will be considered is the Noordzeekanaal-Amsterdam-Rijnkanaal and the drainage into the North Sea through the IJmuiden pumping station. The Amsterdam-Rijnkanaal (ARK) runs from Tiel, through Maarssen, to Amsterdam where it flows into the Noordzeekanaal (NZK). The NZK ends in IJmuiden where the water enters the North Sea. An overview of the system can be seen in Figure 2.1.

Figure 2.1: NZK-ARK system including important structures and urban areas

There are several important structures in the NZK-ARK, labelled in Figure 2.1. The ARK starts in Tiel, where the Prins Bernhardsluizen form the connection with the Waal (the main distributary branch of the river Rhine). The ARK intersects the Lek at the Prinses Irenesluizen and later the lekkanaal at the Prinses Beatrixsluizen, 13km downstream. Finally, the ARK flows into the IJ in Amsterdam, where

the Oranjesluizen form the barrier between the IJ and IJmeer (a bordering lake east of Amsterdam). The Oranjesluizen aid in regulating the water level in the NZK as well as reducing salt intrusion into the IJmeer. The flow continues from the IJ into the NZK, finally reaching the North Sea at the Zeesluis IJmuiden and IJmuiden pumping station.

There are two main sources of water for the NZK-ARK system: the ARK and the waterboards. The Lek and Waal provide the majority of the discharge in the ARK. The discharge in the canal is maintained to reduce salt intrusion and provide water for the surrounding area. Secondly, water is discharged into the NZK-ARK by four surrounding waterboards that regulate water levels in the area (Waternet, Rijnland, De Stichtse Rijnlanden, Hollands Noorderkwartier). A large portion of these areas drain into the NZK-ARK system, shown by the green shaded area in Figure 2.1.

The dimensions of the NZK and ARK can be found in Table 2.1. Due to the lack of inundation areas and floodplains (only 1.5% of the total area [18]), the storage area of the canal is considered independent of the water level and therefore has a constant value. The many ports and side channels around Amsterdam also contribute to the storage area. The water is brackish/fresh due to the mixing of salt water from the sea that enters through the locks, and freshwater from the ARK and waterboards. The average yearly discharge in IJmuiden is $3 \times 10^9 m^3/year$. [19]

Table 2.1: Dimensions of the NZK-ARK [19, 20, 18, 21]

Dimension	NZK	ARK
Length	$26km$	$72km$
Width	$270m$	$100 - 120m$
Depth	$15m$ ($11m$ in Amsterdam)	$6m$
Storage area	$36 \times 10^6 m^2$	

Figure 2.2 shows the IJmuiden complex in more detail, which consists of a pumping station, a gate, and several locks to allow shipping through the canal. The discharges through the locks are negligible compared to that of the pumping station. The North Sea is located to the left of the complex and the NZK to the right, which leads to Amsterdam.

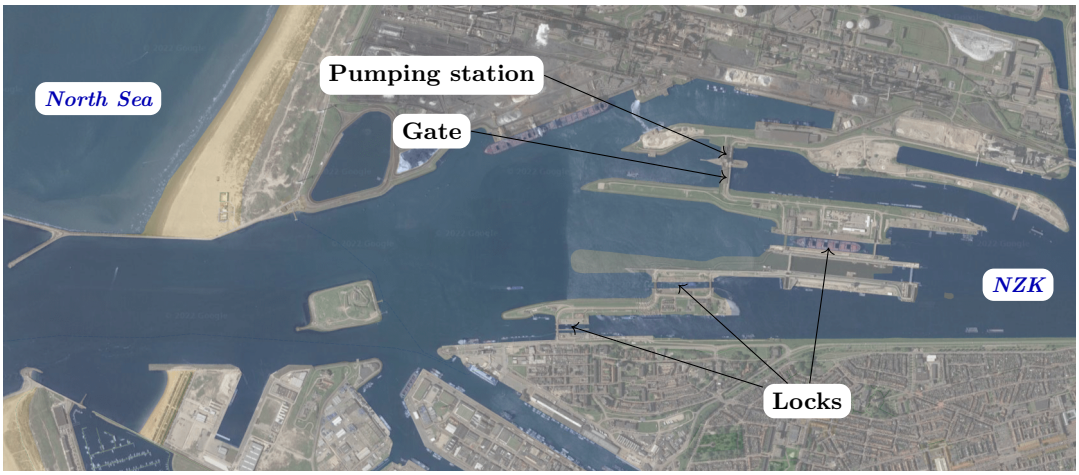


Figure 2.2: Aerial view of the IJmuiden complex with the important structures

2.1. Water level regime NZK-ARK

For safety and shipping, there is a range between which the water level in the NZK should remain. The target water level is $-0.40m{+}NAP$ but the level is allowed to fluctuate between $-0.55m{+}NAP$ and $-0.30m{+}NAP$. When the water level increases above this range, there is a high water situation.

At $-0.20m{+}NAP$ the first problems will arise for shipping as the minimum vertical clearance will not be met. When this level is exceeded, the IJ-front (the connection between the Stadsboezem Amsterdam and the IJ) will be closed off, as the first flooding will occur in Amsterdam at $-0.15m{+}NAP$. The ARK-front (the connection between the Amstellandboezem and the ARK) can also be closed off from the ARK if necessary. The final front that can be closed off is the Amstel-front, which occurs when the water level in Amsterdam rises to $-0.15m{+}NAP$. [22]

A further increase to $0.00m{+}NAP$ will lead to a complete stop of all pumping into the NZK-ARK to minimize further increases in the water level. All these measures help maintain safe water levels, which is paramount as the system runs through two densely-populated areas of high economical value; Amsterdam and Utrecht. A summary of the important water levels in the system can be seen in Figure 2.3. [23]

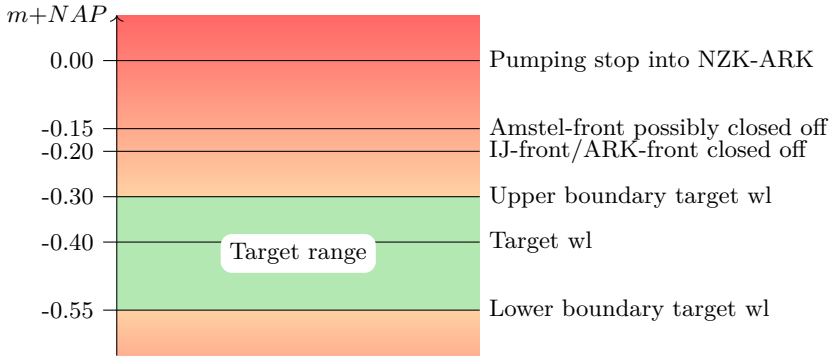


Figure 2.3: Important water levels in the NZK

2.2. IJmuiden gate

To drain water to the North Sea there are two possibilities in IJmuiden: opening the gate or utilizing the pumping station. The gate can be opened at a minimum water level difference of $0.12m$ between the North Sea and the NZK. This higher water level in the NZK is necessary to overcome the pressure difference caused by the lower density of the brackish/fresh water in the canal compared to the saline seawater.

The maximum discharge through the gate is $500m^3/s$ to ensure the stability of the bed around the gate complex. In a high-water situation, this maximum is increased to $700m^3/s$ as flood safety becomes a priority. In addition to these limits of maximum discharge, the discharge is currently kept as evenly distributed as possible. For example, if it is possible to open the gate for 4 hours, it will not be opened for 2 hours with a discharge of $500m^3/s$ but for 4 hours with a discharge of $250m^3/s$. This results in the same total discharge out of the system but with less chance of damage. [11]

2.3. IJmuiden pumping station

When the water level in the North Sea is too high to use the gate, pumping is the only option. Pumping is possible if the sea level is at least as high as the water level in the NZK. As a result, when the water level difference is between 0.12 and $0.0m$, neither pumping nor opening the gate is possible. Generally, around two-thirds of the water can be drained by opening the gate and the remainder needs to be pumped [23]. Currently, the pumping station is controlled using Model Predictive Control (MPC), which is explained in detail in Section 2.5.

There are six pumps installed at the pumping station of IJmuiden with a total capacity of $260m^3/s$. Figure 2.4 shows one of the pumps and Table 2.2 gives an overview of the six pumps with their specifications. All pumps are bulb pumps with an electrical engine.

Figure 2.4: An example of a pump that is installed at the pumping station of IJmuiden [24, 25]

Table 2.2: Overview of pumps at IJmuiden pumping station [24]

Number of pumps	2	2	2	Total
Manufacturer	Stork	Stork	Nijhuis	-
Discharge capacity	$40m^3/s$	40 or $28m^3/s$	$50m^3/s$	$260m^3/s$
Speeds	Fixed speed	Two speed	Variable speed	-
Pump height	$1.2m$	$1.2m$	$1.2m$	-
Max. pump height	$2.35m$	$2.35m$	$2.75m$	-
Power	$1000kW$	$1000kW$	$1540kW$	$7080kW$

The discharge of each pump and its corresponding power depends on the water level difference between the NZK and the North Sea (pump height). These relationships are given in Table 2.3, where Q-dH and P-dH are the discharge - pump height and power - pump height relationships respectively. The relationships depend on the discharge at which the pump is operated. Pumps 5 and 6 are variable speed pumps that have been described using three discharge modes (30, 40, and $50m^3/s$) however all discharges between 0 and $50m^3/s$ can be achieved given a low enough pump height. [26]

Table 2.3: Pump discharge and power relationships for all six pumps in the IJmuiden pumping station [26]

Pumps	Discharge $[m^3/s]$	Q-dH $[m^3/s], [m]$	P-dH $[kW], [m]$
1, 3	40	$Q = -5.4174 \cdot dH + 44.93$	$P = 208.08 \cdot dH + 536.85$
2, 4	40	$Q = -5.4174 \cdot dH + 44.93$	$P = 208.02 \cdot dH + 536.85$
	28	$Q = -6.4977 \cdot dH + 33.149$	$P = 192.36 \cdot dH + 217.26$
5, 6	50	$Q = -1.9822 \cdot dH^2 + 1.9726 \cdot dH + 44.93$	$P = 443.91 \cdot dH + 476.30$
	40	$Q = -1.8544 \cdot dH^2 + 7.7740 \cdot dH + 44.93$	$P = 379.09 \cdot dH + 373.18$
	30	$Q = -7.1021 \cdot dH + 48.164$	$P = 282.97 \cdot dH + 417.32$

There are also time constraints for starting the pumping station to ensure that the electrical grid is not overloaded. Each pump requires 5 minutes to reach the desired discharge, while if all pumps are activated simultaneously, this increases to 25 minutes for full capacity (6 pumps). [26]

2.4. Wind set-up

The wind has a significant effect on the NZK-ARK system. Mainly in the winter months, west winds occur regularly, causing a set-up in the North Sea reducing water level difference with the NZK. This can significantly reduce the possibilities for using the gate, resulting in the necessity to use the pumps. The wind does not only affect the water level in the North Sea but also in the NZK. The last section of the NZK is relatively straight and angled to the North-West, which makes it prone to wind set-up. This further reduces the water level difference. [26]

The set-up occurs as a result of the shear stress created by the wind with the water surface. This is compensated by the gradient of the water level, which means that the water level deviations increase

in magnitude as the length over which the shear stress is exerted (also known as the fetch) increases. The water level deviations are maximum at the ends of the basin, in this case, the straight section of the canal.

2.5. Current control

Currently, the IJmuiden pumping station is controlled with MPC, also known as moving/receding horizon control. It is a popular and successful control method that has been applied in many industries since the 1980s and only gained popularity. MPC computes the optimal control that minimizes a pre-defined cost function while ensuring that the system constraints are satisfied. The future behaviour of the system is determined over a finite time horizon. [16]

Figure 2.5 shows a discrete MPC scheme in more detail where the model has input from the past and then uses this in combination with the model of the system to make predictions over a prediction horizon. The controller requires dynamic models of the system, often in the form of linear empirical models to estimate the effects of its actions. An iterative, finite-horizon optimization is used where a control strategy is computed for a constant period in the future. State trajectories are explored starting in the current state to find a cost-minimizing strategy for the future control period (prediction horizon). After the first step is implemented, the horizon is shifted one timestep forwards and the optimization is repeated starting at the new current state. [28]

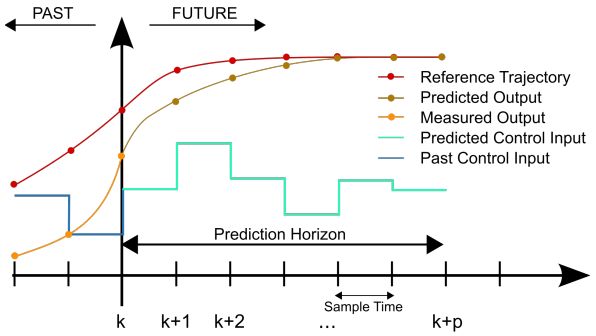


Figure 2.5: A schematization of a discrete MPC scheme [27]

In IJmuiden, an MPC is used that relies on simplified models that describe the effects of pumping and using the gate on the state of the system. Constraints are given to maintain a water level between $-0.30m{+}NAP$ and $-0.55m{+}NAP$. Finally, the optimization is performed to minimize energy use, while energy is bought in advance on the futures market, with a prediction horizon of 24 hours. [29, 26, 30]

In addition to the MPC optimization, the control is set automatically when close to the lower boundary of the target water level range. When the water level is within $2cm$ of the boundary, all outflowing discharge is halted to ensure no further decrease in the water level. Neither pumping nor opening the gate is possible. [7]

3

Electricity Markets

In many situations, as in The Netherlands, it is not yet feasible to store electricity economically and at a significant scale with the current technology. As a result, the consumption and generation of electricity have to be matched perfectly to maintain a safe and stable supply. When electricity storage becomes feasible it will help stabilize the fluctuations in supply and demand, which will only increase with the shift to more renewable energy production. [31, 32]

It is essential to keep supply equal to demand in order to maintain a stable frequency in the power supply. Customers receive an alternating current power, which means it alternates between a negative and positive voltage. The frequency of this oscillation is kept at 50Hz (in Europe, 60Hz in America) at all times. If the electrical frequency deviates slightly from 50Hz, there is a high risk of damage to equipment and infrastructure. For this reason, frequency security is one of the main focuses of transmission system operators (TSOs) [33]. When the demand is higher than the supply, the frequency decreases, and vice versa. The balance is very delicate due to the very slim tolerance of equipment and infrastructure. [34, 35]

The continuous management of supply and demand is performed by the TSO. In The Netherlands this is TenneT. The TSO aims to provide a continuous supply of electricity and facilitate the electricity markets. There are three types of wholesale markets, where electricity is bought and sold before being delivered to the consumer via the grid. Figure 3.1 shows the electricity markets and the relevant time frames. On the forward and futures market, electricity is traded a long time ahead of consumption, between four years and one month ahead. On the day-ahead market (DAM), electricity is bought and sold in hourly blocks for the next day. Finally, the intraday market (IDM) opens after the closing of the DAM, where electricity can be bought and sold up to 5 minutes before delivery. [36]

Figure 3.1: The types of electricity markets with their corresponding time frames

After the IDM closes, the supply and demand are matched in real-time, which is the responsibility of the TSO. In The Netherlands, a balance responsible party (BRP) is responsible for the imbalances that occur in their allocated portfolio. The BRP can also choose to increase their imbalance if that stabilizes the overall system by considering the real-time imbalance prices that are provided by TenneT. [37, 36]

3.1. Day-ahead market

The DAM allows the buying and selling of electricity for the next day, in hourly blocks, with a minimum trading quantity of $0.1MWh$ [38]. The market is run as a daily blind auction that closes at noon (12:00 CET) on the day before delivery. Any bids after this time will not result in a transaction. When the

orders are logged by the market participants, demand and supply curves are established based on the buy- and sell orders for each hour of the following day. The intersection between the two curves is the market clearing price (MCP) and is paid or received by all successful participants of the auction. [39]

These successful participants are all buyers that submitted a higher price and sellers who submitted a lower price than the MCP. The buyers and sellers are not matched individually, but rather there is an overall buy volume equal to the sell volume per hour of the next day. The final MCP is published at 12:55 CET and will remain fixed as the DAM is closed. In 2021, $31,000 GWh$ was traded with an average price of $103 EUR/MWh$ [40]. An example of the DAM prices for The Netherlands can be seen in Figure 3.2.

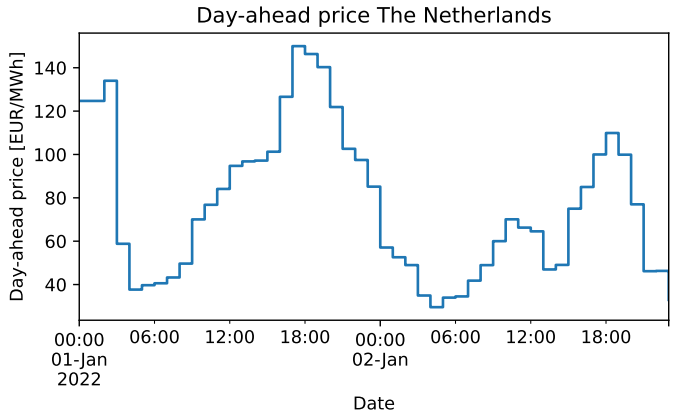


Figure 3.2: Regular DAM prices in The Netherlands, 01-02 January 2022 [41]

The energy bought has to be consumed in the specific hour slot for which it was purchased. In practice, it is hardly ever possible to match the consumption exactly, which creates an imbalance that will be charged in hindsight. This incentivizes the accurate prediction of energy consumption and matching the auction bid as closely as possible.

Due to the high variability in demand across seasons, weeks, and during the day, there are continuous fluctuations in the electricity price. Demand is also influenced by irregular events such as extreme weather or so-called *TV pick-ups* (surges in demand when millions of people watch a television program simultaneously). In the winter months, demand is generally higher than in summer, with an average difference of 36% [42]. This is a result of people spending more time in their homes, an increased demand for hot water, and the use of appliances such as space heaters and electric blankets. During the night, consumption is lower due to the reduced domestic and commercial activity and demand surges in the morning when people start to wake up. These trends and correlations with previous days can be used to estimate prices for the next day when deciding how to bid before the MCP is known.

3.2. Intraday market

After the DAM is closed and the MCP is known, the IDM opens. Trading in quarter-hourly, hourly, or longer interval blocks is possible continuously up to 5 minutes before delivery. The trade is instantly performed as soon as a buy and sell bid are matched. The IDM is important in facilitating the energy transition and optimizing the short-term market. The high variability in renewable energy production means that more flexibility is required to maintain the balance in the power supply. [43]

The flexible nature of the IDM allows BRPs to compensate for imbalances between previously bought energy and their actual consumption.

In 2021, $5,800 GWh$ was traded, which was a significant increase compared to the previous year with $4,300 GWh$ [40], though still at a far smaller scale than the DAM. This highlights the increasing interest in trading in the intraday market, which is also observed in other European countries. This is coupled with the increasing amount of renewable energy production, making the balancing of DAM bids more challenging [44].

3.3. Suitable markets for water system control

Currently, the IJmuiden pumping station is controlled with MPC using energy bought on the futures market, at least a month ahead of consumption. During more extreme weather events, accurate forecasts of the water level in the North Sea can only be made approximately 12 hours beforehand as the windspeeds can be difficult to predict [26]. During calm weather, forecasts can be made accurately longer ahead of time. Forecasts for inflowing discharges from the waterboards and ARK are also far from accurate far ahead into the future. The uncertainties in the forecasts increase when made further into the future. This means that months ahead of time when bids are made on the futures market, the exact water levels cannot be taken into account. On the other hand, with the DAM and IDM, the electricity is bought far closer to the consumption time and forecasts of water level and inflowing discharges into the system are more accurate. These two markets, therefore, show the most potential for cost optimization in water system control, compared to the futures market. Due to the speed of RL models, control plans can be made fast enough to bid on the IDM.

The water system is relatively flexible compared to other infrastructure that may be tied to specific times for consumption or require longer periods of time to change consumption. The pumps require a maximum of 25 minutes to reach full capacity, while the gate can be opened significantly faster. This means that the system is able to respond to changes in the electricity market rapidly and take advantage of lower or even negative prices.

Electricity prices are driven by supply and demand, which means that a lower demand than supply reduces the price to help maintain the balance in the power supply. As a result, negative electricity prices can occur if there is a sudden oversupply, meaning that producers are charged. This stimulates more consumption and reduces production. Producers will weigh the costs of stopping and restarting their plants with the costs of selling at a negative price. As power generation with inflexible renewables increases, negative prices will occur more often in the future. This has already been observed in Germany, where 39 days in 2019 saw negative prices on the DAM [39]. Though the occurrence rate is far lower in The Netherlands, with the increasing percentage of renewable energy production, it is expected to increase in the future. An example of negative electricity prices on the DAM in The Netherlands can be seen in Figure 3.3. With the flexibility of the water system, the negative energy prices can be used to minimize operating costs as well as stabilizing the power supply.

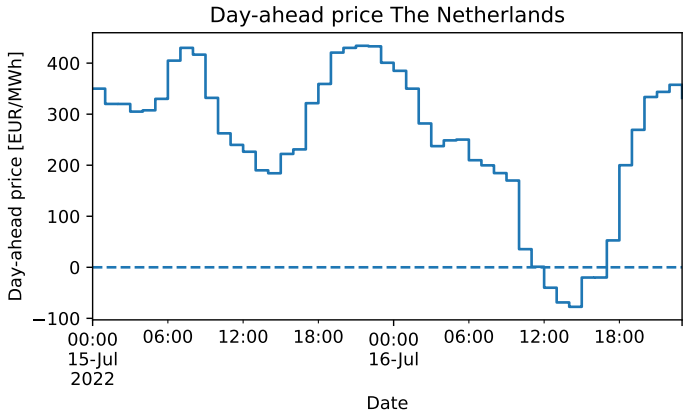


Figure 3.3: Negative DAM prices in The Netherlands, 15-16 July 2022 [41]

By bidding on the DAM as well as the IDM, there is a large potential to reduce pumping costs in IJmuiden. An initial control plan can be made the day before to place a bid on the DAM. At this moment in time forecasts are already reasonably accurate, however, within hours of the actual control, the uncertainties will have decreased further. Reliable forecasts are available at least 2.5 hours ahead of time depending on the weather conditions [26]. Opportunities may also arise in the IDM, which would mean a different control strategy is more economical. If such a market strategy would be used in IJmuiden it may require two control algorithms. The first would create the control plan for bidding on the DAM. The second model would receive the initial control plan as input with the newest forecasts and electricity prices.

4

Deep Reinforcement Learning

In this thesis, we investigate the applicability of RL for the control of the pumps and gates in IJmuiden. The control using RL consists of an agent that can perform actions within an environment. In this case, the environment is a model representation of the IJmuiden water system coupled with the electricity market. The water system changes according to which action is chosen. For example, the agent chooses a specific pump discharge, which combined with the inflowing discharge in the system results in a change in the water level. The state of the water system determines which actions are possible. As described in Section 2.1, the water level difference between the NZK and the North Sea indicates whether pumping or opening the gate is possible.

The agent optimizes an objective function that reflects the desired behaviour. The goal is to minimize electricity costs while ensuring that the water level remains within the target range. When deep reinforcement learning (DRL) methods are used for the optimization, the decisions are made based on the output of neural network (NN). The state of the system is given as input and the output indicates which action should be taken.

This chapter starts by introducing the important concepts in RL, the mathematical background, and the popular methods. Then the different algorithms will be discussed that can be applied to allow the agent to learn. The most suitable and promising algorithms will be presented in detail. Finally, the method for hyperparameter optimization will be introduced, which is an important step in improving the final performance of the model.

4.1. Reinforcement learning concepts

Figure 4.1 shows an illustration of the interaction between the agent and the environment. To make this clearer an explanation is given below of the important terms in RL. [45, 46]

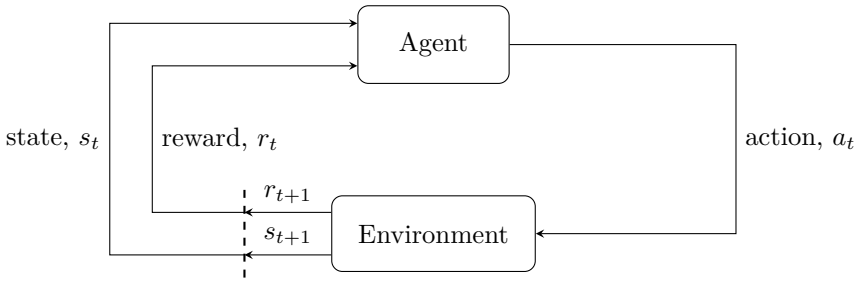


Figure 4.1: Interaction between agent and environment for a Markov Decision Process

Development of an RL algorithm starts by defining the **environment** and the **agent**. The agent is the learner and decision-maker that interacts with the environment. At every step, the agent receives an observation of the environment and a **reward** and performs an action based on these inputs. When following the representation in Figure 4.1 it can be seen that the agent interacts with the environment at time steps, $t = 1, 2, 3, ...$, and at each time step t receives the state of the environment, s_t. Using the

observed state the agent chooses an action from the possible actions in that state, a_t. In the next time step the agent will receive the reward and new state of the environment, r_{t+1} and s_{t+1} respectively.

The reward indicates how good/bad the action was. The agent's goal is to maximize the cumulative reward, also known as the return. This is done by learning a **value function** that gives an estimation of the expected rewards in the future, given the current state. A value function can also be found expressed in terms of a state-action pair. The value function is used to choose the action with the highest expected return. The mapping from the states to actions is known as the **policy**. When the agent learns, the policy is continuously changing, approaching the optimal policy.

4.1.1. Markov Decision Process

In RL problems, the agent makes decisions based on the current state of the environment. When this is repeated it becomes a sequential decision-making problem and the mathematical framework used to formulate the problem this is known as a Markov Decision Process (MDP) [8]. This allows the MDP to take current- as well as future rewards into account and thereby weigh the importance of immediate or delayed rewards. All MDPs need to satisfy the Markov property: given the present state, the future state is independent of past states, as shown in Equation (4.1) [47]. The probability of the next state, s_{t+1}, given that the previous state was s_t is equal to the probability of s_{t+1} given all previous states.

$$P(s_{t+1}|s_t) = P(s_{t+1}|s_1, ..., s_t) \tag{4.1}$$

The (Markov) assumption holds for the dynamics of the IJmuiden pumping station. The model used for the water system is explained in detail in Chapter 5, which shows that all data needed to determine the transition to the new state is included in the previous state. This is often not the case for many tasks performed with RL, which highlights the importance of choosing the appropriate method. Certain methods are more suitable for Markov environments as the Markov property is exploited, whereas other methods perform best in non-Markov environments.

An MDP model contains a set of possible states of the environment, the actions that can be taken in each state, and a real-valued reward function. The agent should make a decision based on long-term rewards rather than purely instantaneous rewards of a single transition towards a new state. The optimal solution can therefore include an action with a lower reward now in order to get a large reward several steps ahead. The return is the long-term sum of rewards with a discount factor, γ, to emphasize short-term rewards and ensure that the return is finite. Equation (4.2) shows the return where $0 \leq \gamma \leq 1$. Discounting future rewards is beneficial as uncertainties in the future may not be accurately represented and therefore short-term rewards may be desirable [47].

$$R = r + \gamma r_{t+2} + \gamma^2 r_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{4.2}$$

The aim is to determine a policy that maps the states to actions while maximizing the return (Equation (4.2)). The value function gives the expected return while following a specific policy. The value function of state s when following policy π is expressed as $v_\pi(s)$, also known as the state-value function for policy π. Equation (4.3) shows the value function where $\mathbb{E}_\pi$ denotes the expected value of a random variable given that policy π is followed.

$$v_\pi(s) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \middle| s \right], \text{ for all } s \in \mathcal{S} \tag{4.3}$$

The policy function can also be defined as $q_\pi(s, a)$, also known as the action-value function for policy π. This is the expected return, starting in state s and taking action a.

$$q_\pi(s, a) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \middle| s, a \right], \text{ for all } s \in \mathcal{S}, a \in \mathcal{A} \tag{4.4}$$

4.1.2. Reinforcement learning methods

The value function is learned using estimates of the return. Two methods that assume a perfect model of the environment in the form of an MDP are dynamic programming (DP) and exhaustive search. DP

methods break the problem down into sub-problems, solve these sub-problems, and then combine the solutions while an exhaustive search goes through all possible paths and possibilities. The assumption of a perfect model and the high computational cost means that the utility of these types of methods is limited in the IJmuiden application. Methods that do not assume full knowledge of the environment are temporal-difference (TD) learning and Monte Carlo (MC). This means that the probabilities of the transitions are not known. No prior knowledge of the environment is needed and learning is done from experience (sample sequences of states, actions, and rewards). An overview of these methods and how they differ is shown in Figure 4.2.

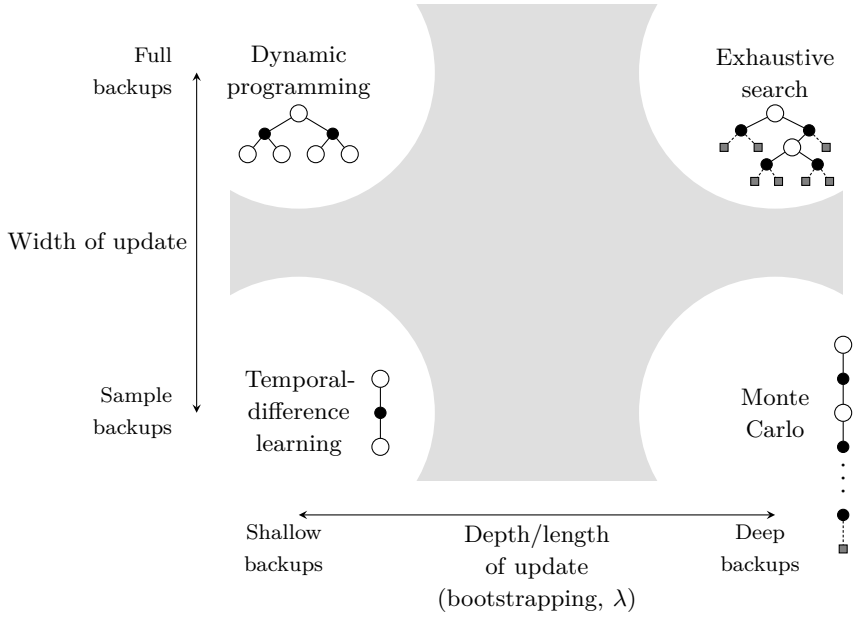


Figure 4.2: Width and depth of backups for main RL methods

MC follows the policy from a given state until the episode terminates to obtain an unbiased sample of the return distribution. The method can have high variance due to the stochasticity of the transitions. In comparison, with TD, the recursive property of value functions is used as there is already an estimate of the value function, as shown in Equation (4.5). This method does not have high variance as the estimate of the value function, $\hat{Q}_\pi(s', \pi(s'))$, provides an expected return distribution from s' as opposed to a sample from the distribution. However, the value function is being learned and is therefore biased.

$$q_\pi(s, a) = r + \gamma \hat{Q}_\pi(s', \pi(s')) \tag{4.5}$$

The value function can be learned for the policy that was used to generate the data (on-policy) or for another policy (off-policy). Off-policy uses samples from an arbitrary policy to reach the optimal policy, which is more efficient in sampling. Learning is generally done with samples obtained with exploratory policies that aim to provide diverse samples to learn from. This ensures that the value function is accurate for all states that can be visited by the optimal policy. The sub-optimal policy may not visit all states that are visited by the optimal state and therefore stochasticity is added to the policy. Secondly, sample diversity is very important when using deep neural networks for the approximation of the value function as large amounts of diverse examples are required for generalization.

Method comparison

The algorithms that are mainly applied are either TD or MC as it is not assumed that there is full knowledge of the environment. In most practical cases this is indeed not known which makes the other methods unsuitable. In the case of IJmuiden, the inputs will contain forecasts which are never exact and these uncertainties mean that the exact dynamics of the system are not known. Below is a list of the key differences between these two learning method types.

- **Backup depth**

  TD can learn before the episode is finished and the final return is known. This also means that online learning and learning from incomplete sequences are possible. On the other hand, MC needs the episode to terminate before learning. TD can be used for MDP that are continuing/cyclic while MC only for those that are episodic/terminating.
- **Variance**

  MC had a high variance and no bias as no estimates of the values are used, only the final return of the episode. TD has low variance with some variance due to the use of value estimates in the updates.
- **Efficiency and convergence**

  TD exploits Markov property where MC does not. This means that TD is usually more efficient in Markov environments and MC in non-Markov environments. MC has good convergence properties, is not sensitive to initial values, and is simple to understand and use. TD is usually more efficient but more sensitive to the initial values.

When considering the IJmuiden pumping station several aspects suggest that TD may be a more suitable technique. Firstly, the control in IJmuiden is continuous and does not have a natural termination as is the case with a game such as PacMan, where the game ends when all the food is eaten or PacMan touches a ghost. The problem formulation means that the water system is a near-Markov environment; the next state is determined by the current state and the chosen action, however, there might be a slight uncertainty due to the use of forecasts.

4.2. Deep reinforcement learning algorithms

As RL problems become more complex it is no longer possible to explicitly store values and actions for all states. This is solved by using a function to approximate and generalize the state space. In addition, for many real-life problems, the true state is unknown, and instead, an observation is used, often with high dimensionality and redundancy. NNs can represent a function given enough parameters to fit. This is a significant advantage as any smooth function can be approximated which allows the network to learn complex non-linear policies/value functions. The large number of parameters does come with a drawback: optimization of these parameters is difficult and overfitting is more likely. The model can learn too well, achieving a high performance on the training data but performing very badly on new, unseen inputs. We require the model to generalize based on the training examples. When a deep neural network (DNN) is used for the approximation, the algorithm is a DRL algorithm. This is often the case for large complex environments such as the IJmuiden pumping station considered in this research.

There are many different DRL algorithms that are suitable for different types of problems. Most aim to find an approximation for the optimal policy/value function (the mapping from states to actions or states to the expected reward).

4.2.1. Common solution components

Several components are often included in DRL algorithms to improve performance or avoid pitfalls [8]. These are also likely to occur in the training of the RL agent for the control in IJmuiden.

- **Delayed targets**

  A strong correlation exists between the network predictions and the bootstrapped value estimates (target values). This occurs because the experience samples that are used for training are gathered by following the current policy. Parameter updates in DNNs have a global effect and therefore poor updates affect predictions in the whole state space. This can cause divergence of the learning process [48], which is solved by calculating target values using a different network (target network). This target network is an older version of the value function and allows the agent to train towards a more stable target.
- **Trust region updates**

  Due to the non-linearity of DNNs, small changes to the parameters can have a significant and unexpected effect on the behaviour of the function. Smaller learning rates reduce this issue however this is often not feasible due to the constraint of computation. There are three possible solutions:

constraining the change to the policy [49], clipping the objective function to only consider small changes to the policy [50], and constraining the parameters to not deviate too far from the running average of previous policies [51].

- **n-step returns**

  A problem arises with bootstrapping methods caused by the bias in updates. These methods generally learn faster, however, there is also a danger that the algorithm fails to learn anything useful. This is less likely when not only one step ahead is considered. As a result, the true reward observed during n time steps and the learned value estimate in time step $n+1$, are used to estimate the return, as shown in Equation (4.6). As with the use of target networks, this technique reduces the correlation between the target value and the value function being learned.

$$q(s_t, a_t) = r_t + \gamma r_{t+1} + ... + \gamma^{n-1} r_{t+n-1} + \gamma^n \hat{Q}(s_{t+n}, a_{t+n}) \tag{4.6}$$

- **Experience replay**

  During the training of DNNs, gradients are calculated which can have a strong temporal correlation with the experiences (sample data). This is avoided by creating a buffer of experiences from which a random batch of samples can be taken. This also improves convergence as even for sudden changes in the current policy, the distribution of training samples only changes slowly as experiences are added to the buffer. In the case of off-policy learning, a large portion of previous experiences can be used for learning, which also improves sample efficiency.

- **Input, activation, and output normalization**

  Non-linear activations used in DNNs can create near-zero gradients when inputs to a layer are not within a sensible range. As a result, the parameters leading up to the activation will not be updated and the network will not learn. There are many normalization techniques such as batch normalization [52], input normalization, layer normalization [53], and weight normalization [54]. On the other hand, it is also important that gradients do not become too large. Due to this DRL algorithms can be sensitive to the scale of the rewards.

When considering the IJmuiden pumping station, several components described above are expected to be beneficial for the performance of the RL agent. The experience samples with be generated by following the current policy resulting in a strong correlation. The agent will choose the outflowing discharge in the system and observe the new state and rewards. These samples are not available before training and will be gathered as the agent is trained to explore the relevant parts of the state- and action space. Secondly, there is a strong temporal correlation between experience samples. This is due to the gradual changes in the state of the water system. When a normal discharge of $50m^3/s$ occurs in the NZK-ARK (with a storage area of $36 \times 10^6 m^2$) there will only be a water level change of $5mm$ after 1 hour. The water levels in the system are highly correlated as well as many other parameters in the state. Therefore, it is expected that algorithms with experience replay will yield good results.

Normalization has shown to be an effective technique to improve DNN performance in many applications, and therefore it is expected to be beneficial for learning speed and final performance in this application as well. If a bootstrapping method is selected, n-step returns are also expected to increase final performance.

4.2.2. Popular algorithms

This section discusses the more popular algorithms that implement varying combinations of the components described above. The algorithms are generally suitable for certain types of RL problems and have different advantages and disadvantages. [8]

- **Neural Fitted Q Iteration (NFQ)**

  This algorithm learns offline using a fixed experience buffer of previously obtained interaction samples. To improve convergence, the target Q-values are calculated for all states at the start of each optimization iteration. Secondly, artificial experience samples are added to the database at goal states, where true Q-values are known.

- **Deep Q-network (DQN)**

  This method builds on the NFQ algorithm but adds new experiences to the buffer (experience replay) as obtaining a fixed set of experiences is often not possible. The constantly changing buffer and learned Q-function means that no accurate targets are available initially. A copy of the Q-function is kept in memory and used as a target network, which is updated to the current parameters at regular intervals (delayed targets). This algorithm has achieved high success in high-dimensional and large-scale problems. It can achieve stable training for MDPs with uncertain environments and can handle a continuous state space. However, the action space is discrete and it is a value-based method, which means that direct optimization of the policy is not possible.
- **Double Deep Q-network (DDQN)**

  The max operator used for the Q-values causes DQN to suffer from a bias due to the overestimation of the returns. The DDQN method uses the same two networks, where the target network is the same as in DQN. The second network is used to determine for which action the target Q-function is evaluated. This small change to the method has been shown to improve the convergence and performance of the algorithm.
- **Deep Deterministic Policy Gradient (DDPG)**

  While DQN is not applicable for environments with continuous action spaces, this closely related actor-critic method is suitable. The critic estimates the value function and the actor updates the policy distribution according to the critic. Both functions are represented by a DNN and a copy is used as a target network for each. The method is off-policy which means that experience replay can be used. The algorithm is most suitable for problems where the domains have stable dynamics.
- **Trust Region Policy Optimization (TRPO)**

  This on-policy method is relatively complicated and has inefficient sampling, however, can reliably improve the policy. It is a policy gradient method that uses a large number of episodes following the current policy to get state-action pairs with MC estimates of the returns.
- **Generalized Advantage Estimate (GAE)**

  The TRPO algorithm can be used in combination with an exponentially-weighted estimator of the advantage function for the value function. The value function is learned from MC estimation including trust region updates. [55]
- **Proximal Policy Optimization (PPO)**

  TRPO prevents the use of certain NN architectures. To solve this and reduce the complexity of the TRPO algorithm, a clipped version of the objective function is used combined with GAE.
- **Asynchronous Advantage Actor Critic (A3C)**

  Rather than collecting environment samples on-policy with single policies consecutively, A3C uses multiple parallel actors with globally shared parameters (weights in the NN for approximation of the value function). Each actor calculates updates for shared parameters which are applied asynchronously. To make sure that different parts of the state and action space are explored, all actors use a different exploration policy. Another variant of this algorithm is Advantage Actor Critic (A2C), which works in the same manner only the updates are not applied asynchronously. [56]
- **Actor Critic with Experience Replay (ACER)**

  On-policy methods (TRPO, PPO, A3C) have a significant drawback in that past experiences following other policies cannot be used. This algorithm combines the good convergence properties of these methods with the higher sample efficiency of off-policy methods. This is done by combining A3C with a trust region update scheme. Actor Critic (AC) is a similar method, the only difference being that experience replay is not included. [51]

A concise overview of the algorithms and their main differences can be found in Table 4.1.

Table 4.1: Overview of Deep reinforcement learning algorithms and their main components [8]

Algorithm	Policy	Return estimation	Update constraints	Data distribution
NFQ	Discrete	1-step Q	Bootstrap with old θ	Off-policy fixed apriori
(D)DQN	Discrete	1-step Q	Bootstrap with old θ	Off-policy experience replay
DDPG	Continuous	1-step Q	Bootstrap with old θ, w	Off-policy experience replay
TRPO	Discrete/continuous	∞-step Q	Policy constraint	On-policy
PPO	Discrete/continuous	n-step advantage [1]	Clipped objective	On-policy
A3C	Discrete/continuous	n-step advantage	-	On-policy
ACER	Discrete/continuous	n-step advantage	Average policy network	On-policy + Off-policy

Several aspects were considered when determining which RL algorithm was most suitable for the IJmuiden pumping station. Firstly, when looking at the action space, both a discrete- and continuous action spaces are possible. Technically, all outflowing discharges within the feasible region are achievable, however, this can easily be discretized without losing control flexibility. As a result, both algorithms with discrete and continuous action spaces are suitable.

When looking at the previous performance of algorithms for other applications, DQN jumps out with state-of-the-art performance on domains with discrete actions in terms of final performance and data efficiency [57]. This is achieved with the addition of extensions to the algorithm such as double Q-learning and n-step returns, which also lead to faster training [46]. These advantages in combination with the relative simplicity of the algorithm suggest that DQN is a suitable RL algorithm for the case study. During model development, tests can be done to determine which extensions are suitable for the specific environment of IJmuiden.

NFQ has a fixed experience buffer which is less suitable for this application. Initially, the model needs to learn to maintain the necessary water levels after which the cost optimization becomes the important objective. This is learned best when the experience buffer can change with the improving performance of the agent. This algorithm does not allow the reward to change during training as the experiences are all sampled before training starts.

DDQN is an extension to DQN that only allows continuous action space that could be advantageous for this case as there is an ordinality in the actions. This ordinality is lost when represented discretely.

TRPO, PPO, A3C, and ACER also allow continuous action spaces. TRPO and A3C have inefficient sampling and TRPO is relatively complicated.

Overall, DQN is a suitable algorithm for the initial implementation of the agent. The algorithm only allows discrete action spaces. After the model has been successfully implemented, other algorithms can be tested to see if performance can be improved further. Using continuous action spaces may yield better results as the ordinality of the actions is preserved.

4.2.3. Deep Q-network

The DQN algorithm was developed in 2015 by DeepMind and had great success in solving a range of Atari games to a sometimes superhuman level [48]. It consists of Q-learning with deep neural networks combined with experience replay. The algorithm only allows discrete action spaces, which means that the possible discharge of the pumps and gate will need to be discretized.

Q-learning is based on the Q-function, $q_\pi(s, a)$, that represents the expected discounted sum of rewards (return) when in state, s, and first taking action, a, after which policy, π, is followed. The Q-function is updated using the Bellman optimality equation iteratively (see Equation (4.7)) which converges to the optimal policy. [58]

$$q_{i+1}(s, a) = \mathbb{E}\Big[r + \gamma \max_{a'} q_i(s', a') \Big] \tag{4.7}$$

The NN is trained using examples of experiences in the environment. These consist of a starting state, chosen action, received reward, and the new state. New experiences are generated by allowing the agent to make decisions following an ϵ-greedy policy. An action is either selected at random, with a probability of ϵ to explore the state space or following the current policy. New experiences in the environment are continuously added to an experience replay from which sampling is randomly performed at regular interaction intervals. The buffer of experiences is constantly changing and therefore target

[1] (GAE)

values cannot be determined a priori. The optimization targets are calculated using the target network with the same architecture as the prediction network. The target network uses frozen parameters ($\boldsymbol{\theta}^-$) that are periodically updated to the values in the prediction network ($\boldsymbol{\theta}$). This leads to more stable training as the target function is fixed for a while. An illustration of the architecture of these networks can be seen in Figure 4.3.

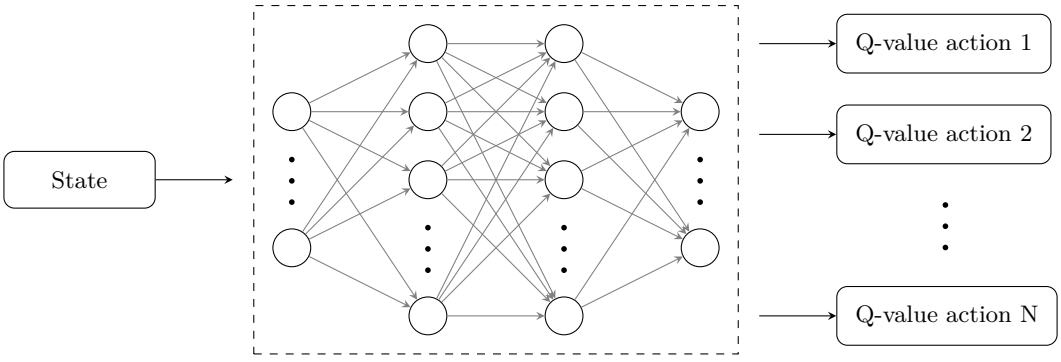


Figure 4.3: Schematization of a Deep Q-network

The use of two networks that are constantly changing highlights the challenge of chasing a non-stationary target. The periodic updating of the target network improves performance as the target network is stable for the interval between each parameter update.

Typically when training a NN we seek to minimize or maximize an objective function. In the process, the parameters of the network are updated according to the gradient of the loss function, which shows the direction that minimizes this loss. Calculating how to update each parameter in the NN relies on the chain rule to propagate the gradient backwards through the layers of the network. The loss function used to update the NN in DQN is shown in Figure 4.4, which consists of two parts from the target and prediction network.

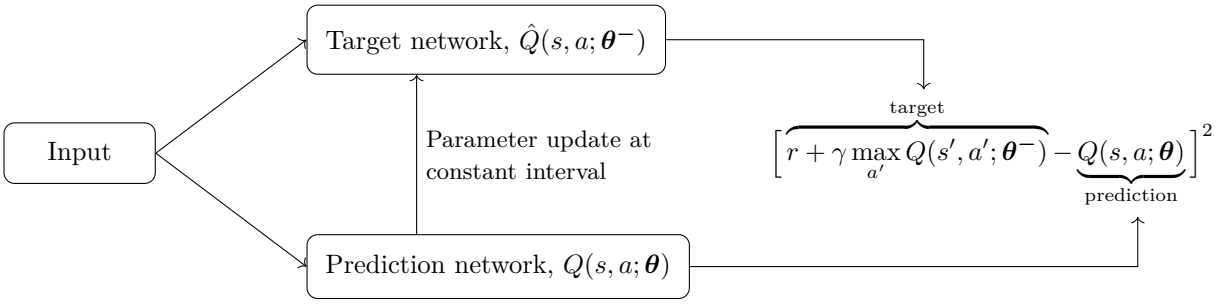


Figure 4.4: DQN loss function with components from the target and prediction network

The learning rate determines how much the NN weights are adjusted with respect to the gradient descent loss. It has to be chosen appropriately as it greatly influences the convergence speed. If the learning rate is chosen too large there is a chance that minima are overshot whereas a small learning rate can take a very long time to converge. The update is performed as shown in Equation (4.8).

$$\underbrace{New\_Q(s,a)}_{\substack{\text{New Q}\\ \text{value}}} = \underbrace{Q(s,a)}_{\substack{\text{Current}\\ \text{Q value}}} + \overbrace{\alpha}^{\substack{\text{Learning}\\ \text{rate}}} [\underbrace{R(s,a)}_{\text{Reward}} + \overbrace{\gamma}^{\substack{\text{Discount}\\ \text{rate}}} \underbrace{\max_{a'} \hat{Q}(s', a'; \boldsymbol{\theta}^-)}_{\substack{\text{Max expected}\\ \text{future reward}}} - Q(s, a; \boldsymbol{\theta})] \tag{4.8}$$

The DQN pseudo code is shown in Algorithm 1, which shows the main steps performed to train the RL agent and how the NN is updated.

Algorithm 1 Deep Q-Network [48, 59, 60]

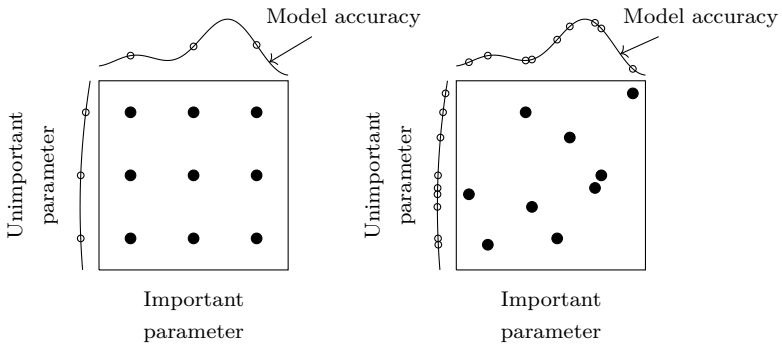
Input: state of the water system
Output: Q action value function (from which we obtain policy and select action)
Initialize replay memory $\mathcal{D}$
Initialize action-value function Q with random weight $\boldsymbol{\theta}$
Initialize target action-value function $\hat{Q}$ with weights $\boldsymbol{\theta}^{-} = \boldsymbol{\theta}$

for each episode = 1 to M **do**
    for each t = 1 to T **do**
        Following ϵ-greedy policy, select $a_t \leftarrow \begin{cases} \text{a random action} & \text{with probability } \epsilon \\ \text{argmax}_a \, Q(s, a; \boldsymbol{\theta}) & \text{otherwise} \end{cases}$
        Execute action a_t and observe reward r_t and new state $s_t + 1$
        Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$
        // experience replay
        Sample random minibatch of transitions (s_j, a_j, r_j, s_{j+1}) from $\mathcal{D}$
        Set $y_j \leftarrow \begin{cases} r_j & \text{if episode terminates at step } j+1 \\ r_j + \gamma \max_{a_{j+1}} \hat{Q}(s_{j+1}, a_{j+1}; \boldsymbol{\theta}^{-}) & \text{otherwise} \end{cases}$
        Perform a gradient descent step on $(y_j - Q(s_j, a_j; \boldsymbol{\theta}))^2$ w.r.t. the network parameter $\boldsymbol{\theta}$
        // periodic update of target network
        Every C steps reset $\hat{Q} = Q$, i.e., set $\boldsymbol{\theta}^{-} = \boldsymbol{\theta}$
    end for
end for

4.3. Hyperparameter optimization

Training times and required memory can be decreased and performance increased by choosing the appropriate hyperparameters. This includes parameters for the neural networks but also those of the chosen RL algorithm, such as the discount rate and exploration. All these parameters are set before training starts.

The most common methods for hyperparameter optimization are manual, random search, grid search, and Bayesian model-based optimization. To illustrate grid and random search, Figure 4.5 shows the search for two parameters, where one parameter is far more important than the other. Both search techniques perform nine trials within the search space. In random search, the space is a bounded domain of parameter values, and points are randomly sampled. Grid search defines the space as a grid of parameters where all positions in the grid are evaluated.



(a) Grid search **(b)** Random search

Figure 4.5: Grid and random search with nine trials for optimizing a function $f(x, y) = g(x) + h(y) \sim g(x)$

Figure 4.5 highlights one of the drawbacks of grid search. The model has to be trained 9 times, while only 3 values are explored for both parameters. As a result, the best model found can still be far from optimal. The search is also an exponential time algorithm as the time taken grows exponentially with the number of parameters to tune. Random search generally performs better as the search space grows and when certain parameters are more important.

However, random search does not take into account any of the previous trials while these do contain information about possible promising values. This forms the basis of Bayesian hyperparameter optimization as this technique creates a probability model of the objective function to select the most promising parameters [61]. These parameters are then evaluated in the actual objective function. In the case of a reinforcement learning problem, the objective function could be the average return for a set of test cases. [62]

4.3.1. Bayesian Model-Based Optimization

The method that will be used for hyperparameter optimization is Sequential Model-Based Optimization (SMBO) with Tree Parzen Estimator (TPE) [62]. Hyperparameter optimization is represented by Equation (4.9), where $f(x)$ is an objective function that gives a score to minimize. The objective function takes the set of hyperparameters as input and returns a single score.

$$x_* = \underset{x \in \mathcal{X}}{\operatorname{argmin}} f(x) \tag{4.9}$$

The challenge is that evaluating the objective function is extremely expensive as the agent needs to be trained and then validated for each set of hyperparameters tested (trial). To reduce the number of trials needed to find a near-optimal set of hyperparameters, a probabilistic model (surrogate model) is used to map the hyperparameters to a probability of a score in the objective function ($p(y|x)$). After each new trial, the new results are incorporated into the surrogate model. This becomes very efficient with a large search space as slightly more time is spent selecting the next hyperparameters to reduce the number of expensive evaluations of the objective function.

The steps described above are the main elements in SMBO. There are multiple methods to construct the surrogate model, such as gaussian processes (where $p(y|x)$ is modelled directly), random forest

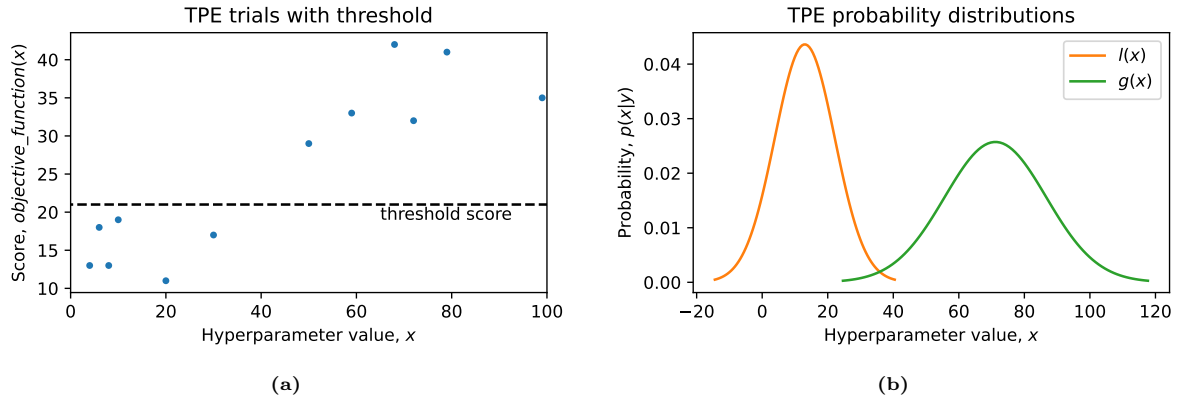
regressions, and TPE (where $p(x|y)$ and $p(x)$ are modelled). [63]

The selection is done by maximizing the expected improvement as shown in Equation (4.10), where y^* is a threshold score of the objective function. This means that the best hyperparameters are chosen under the surrogate model of $p(y|x)$. The TPE only uses performance in previous trials without taking into account the correlation between the hyperparameters.

$$EI_{y^*}(x) = \int_{-\infty}^{\infty} \max(\underbrace{y^*}_{\text{target}} - \overbrace{y}^{\text{loss or score}}, 0)\, p(y|x)\, dy \tag{4.10}$$

In TPE the surrogate model is constructed by using the Bayes rule and does not directly represent $p(y|x)$. The distribution of the probability of the hyperparameters given a score is expressed depending on whether the score is above or below a threshold, as shown in Equation (4.11) and Figure 4.6.

$$p(y|x) = \frac{p(x|y) * p(y)}{p(x)} \quad , \qquad p(x|y) = \begin{cases} l(x) & \text{if } y < y^* \\ g(x) & \text{if } y \geq y^* \end{cases} \tag{4.11}$$



(a) **(b)**

Figure 4.6: Tree Parzen Estimator; (a) objective function scores for hyperparameter values; (b) probability distributions for hyperparameters above and below the threshold score

In TPE samples are drawn from $l(x)$ and evaluated in terms of $l(x)/g(x)$. The samples, therefore, have a higher chance of scoring below the threshold score and a lower chance of resulting in a score above the threshold. The new trial is performed with the set of hyperparameters that performed best and therefore have the highest expected improvement. Subsequently, these parameters are evaluated using the true objective function. The surrogate model is an estimate of the objective function which is repeatedly updated with new trials.

5

Water System Model

This chapter describes how the IJmuiden pumping station and NZK-ARK system have been modelled. This model is used by the RL model to determine the effect of the agent's actions. The model includes the NZK-ARK system, surrogate models for the pumping station and gate as well as the method used to take wind set-up into account.

5.1. NZK-ARK system

A linear reservoir model will be used to represent the NZK-ARK system, as described in [64]. There is a fixed surface area of the system and the net discharge determines the change in water level. The simplification is appropriate as there are no inundation areas or floodplains. A schematization of the linear reservoir can be seen in Figure 5.1, and Equation (5.1) shows how the water level change in the system is calculated.

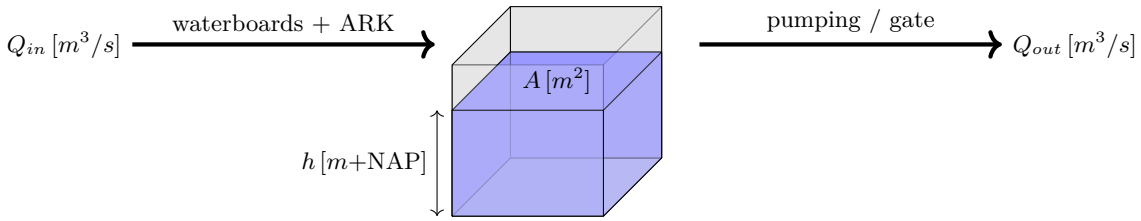


Figure 5.1: Linear reservoir model of the NZK-ARK

$$\Delta h\,[m] = \frac{dt * (Q_{in} - Q_{out})}{A} \tag{5.1}$$

where:

Δh	Water level change in NZK-ARK $[m]$
dt	Time step $[s]$
Q_{in}	Inflowing discharge $[m^3/s]$
Q_{out}	Outflowing discharge $[m^3/s]$
A	Storage area of the NZK-ARK $[m^2]$

The depth of the system is approximately $11m$, with a storage area of $36 \times 10^6 m^2$. The flow velocities in the canal are small, $0.054m/s$ on average during 2021 [65].

The inflowing discharge is found by combining the discharges of the four waterboards and in the ARK at Maarssen. Direct rainfall is neglected since the cumulative rainfall is very small compared to the actual discharge in the system. The outflow is chosen by the RL agent and the possible choices are determined by the pump and gate constraints as well as the water level in the North Sea.

If the sea level is at least $0.12m$ lower than that in the NZK, the model can only choose to use the gate. If the sea level is between 0.12 and $0m$ below the level in the water system, neither pumping nor opening the gate are possible. If the water level at sea is equal to or higher than that in the NZK, pumping is possible.

5.2. IJmuiden pumping station

5.2.1. Surrogate model - pumping station

To simplify the system, the whole pumping station was considered as if it consisted of one pump, as performed in [64]. The single pump was described by a maximum discharge depending on the pump height and a power consumption as a function of the discharge and pump height. To determine the maximum discharge, the *Q-dH* relationships of the individual pumps in Table 2.3 were combined. This was done by taking the case where discharge was maximum for each pump type. The resulting relationship is shown in Equation (5.2).

$$Q_{p,max}\,[m^3/s] = -1.9644 \cdot dH^2 - 17.7244 \cdot dH + 269.58 \tag{5.2}$$

The maximum pump discharge depends only on the pump height. Figure 5.2 shows the feasible discharges for the pumping station.

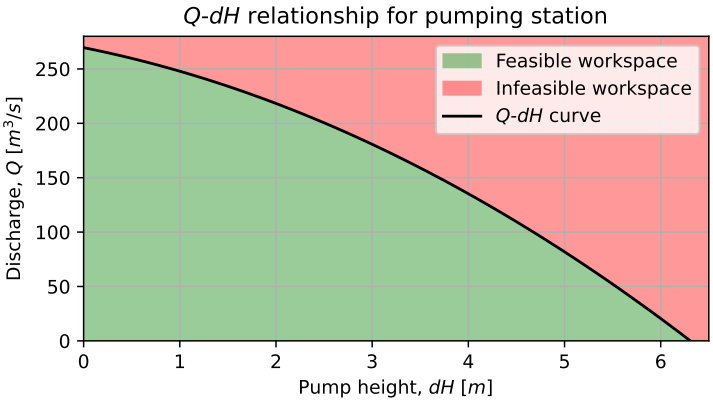


Figure 5.2: *Q-dH* relationship for pumps showing the feasible workspace

The power consumption depends on the pump height and discharge as well as which pumps are used as each pump has a different power curve. The relationship for the simplified single pump was fitted by optimizing the pump configuration for varying combinations of discharges and pump heights to minimize power consumption. The optimization was formulated as a Mixed Integer Programming (MIP) problem, which was subsequently solved using the Gurobi optimizer [66]. MIP problems are commonly described as an objective (minimizing the power consumption) and constraints (six possible pumps to activate). An elaborate explanation of the method can be found in Appendix A.

The resulting power consumption for the optimal pump configurations can be seen in Figure 5.3. This also shows the boundary of the feasible pump discharges as calculated in Equation (5.2). The missing data in the figure was due to some discharge and pump heights not being achievable with combinations of the pump modes. However, all discharges in the feasible region can be achieved in reality with the variable speed pumps. This could not be calculated during the optimization as the variable speed pumps were simplified into three modes (30, 40, and $50m^3/s$) [26]. A function was fitted to the MIP solution to ensure that the power consumption could be calculated throughout the feasible workspace.

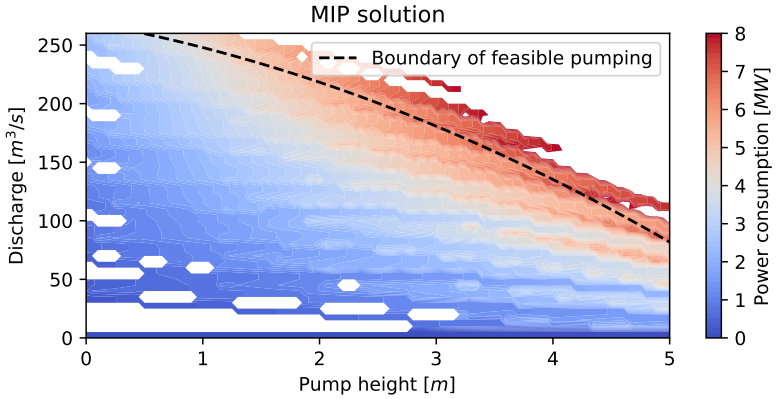


Figure 5.3: MIP results for the power consumption of the optimal pump configuration

A fitted function was found using a least-squares solution to the linear matrix equation as shown in Equation (5.3). The solution can be found in Equation (5.4). This matrix was chosen as it produced an accurate representation of the whole feasible pumping region.

$$\begin{bmatrix} dH & Q & dH^2 & dH \cdot Q & Q^2 \end{bmatrix} \cdot \vec{x} = P \tag{5.3}$$

$$\begin{aligned} P_p\,[kW] &= a \cdot dH + b \cdot Q + c \cdot dH^2 + d \cdot dH \cdot Q + e \cdot Q^2 \\ a &= -2.64e + 02 \qquad d = 8.53e + 00 \\ b &= 8.30e + 00 \qquad e = 2.77e - 03 \\ c &= 1.03e + 02 \end{aligned} \tag{5.4}$$

The relationship between the power consumption and the discharge and pump height can be seen in Figure 5.4.

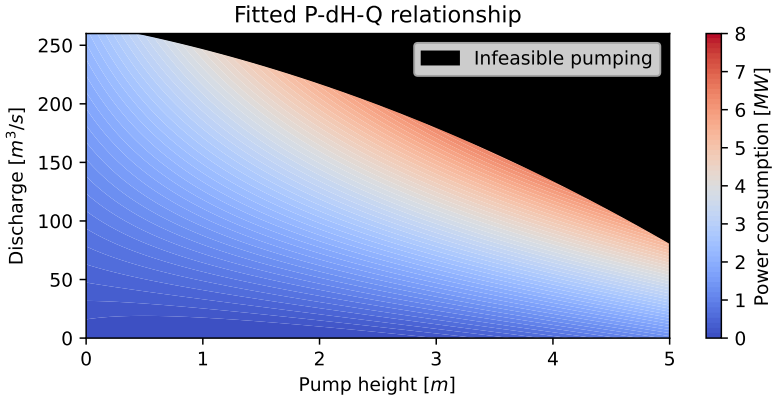


Figure 5.4: Fitted *P-Q-dH* relationship for the MIP results for the power consumption of the optimal pump configuration

This resulted in a simplification of the pumping station where the controller could choose a discharge in the feasible workspace (shown in Figure 5.2), which depended on the pump height. The chosen discharge and the pump height resulted in an optimal power consumption, which can be seen in Figure 5.4.

If all pumps are activated simultaneously, it takes a maximum of 25 minutes for all pumps to reach maximum capacity and a single pump requires 5 minutes [26]. For the purposes of this research, it is assumed that the pumps can instantaneously reach the desired discharge. Time steps of 1 hour are used which means that the start-up time can become a significant portion of the time step. The effect of this assumption will be analysed with the results of the RL model.

5.2.2. Surrogate model - gate

A simplification was made similar to that of the pumping station to simplify the behaviour of the gate and provide possible actions for the controller. The gate consists of seven trumpet-shaped tubes that were simplified into a single gate. The lowest point of the tubes is at $-9.25m + NAP$ and the entirety of the tubes are submerged under water.

The behaviour of the gate was described by Equation (5.5) using the parameters specified in Table 5.2. The schematization is illustrated in Figure 5.5. This shows the relationship between the maximum discharge and the water level difference between the NZK and the North Sea. [67, 68, 69]

Table 5.2: Parameters of the gate for calculation of the maximum discharge [67]

Parameter	Symbol	Value	Unit
Number of tubes	n	7	–
Width of each tube	B	5.9	m
Throat height	h_k	4.8	m
Contraction coefficient	α	1.0	–
Gravitational constant	g	9.81	m/s^2

$$Q_{g,max}\,[m^3/s] = n \cdot \alpha \cdot B \cdot h_k \cdot \sqrt{2 \cdot g \cdot dH} \tag{5.5}$$

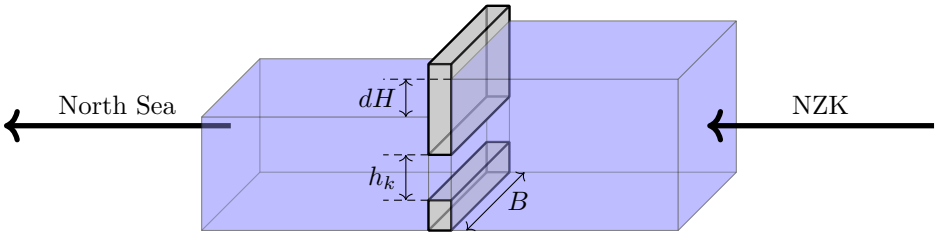


Figure 5.5: Gate schematization for the calculation of the maximum discharge

There are two additional constraints of a maximum discharge of 500 or $700m^3/s$, depending on the water level, and a minimum water level difference of $0.12m$. The maximum discharge is set is in order to limit damage to the bed around the gate. If the water level in the NZK exceeds $-0.3m+NAP$ it is a high water situation and the maximum discharge is higher. The minimum water level difference between the North Sea and the canal reduces salt intrusion near the bottom of the canal.

When the constraints of the gate, as well as those from the water system, were combined, a feasible discharge region was found as shown in Figure 5.6. The two scenarios, with and without high water, are represented by the green region and the combination of the green and orange regions respectively.

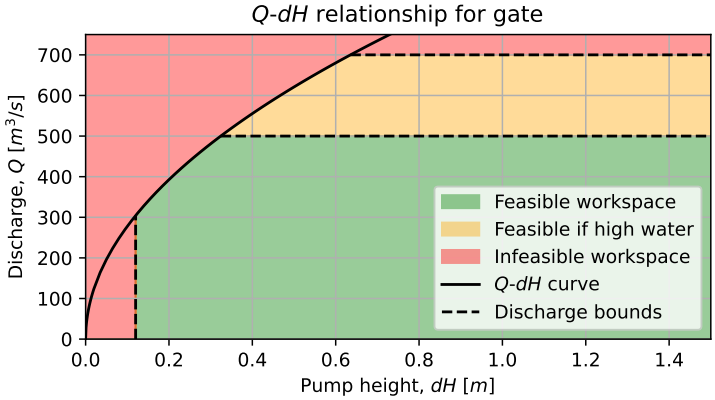


Figure 5.6: *Q-dH* relationship for gate showing the feasible workspace during regular and high water conditions

The power consumption of the gate was not taken into account as the opening and closing consume a negligible amount of power compared to that of the pumps.

5.3. Wind set-up

The effect of wind set-up is shown in Figure 5.7, which illustrates the resulting water levels caused by the shear stress of the wind. The canal has been schematized as a basin. The water level difference is calculated using Equation (5.6), which is explained in more detail in Appendix B.

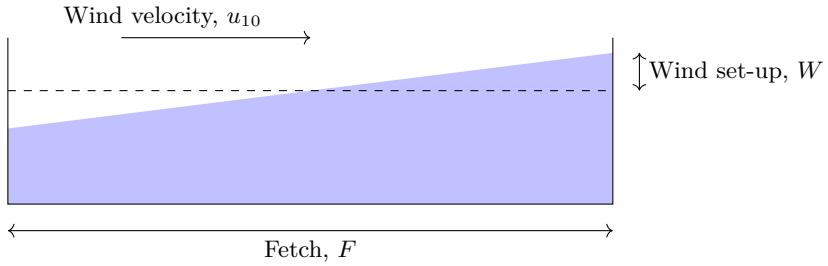


Figure 5.7: Basin schematization for wind set-up

$$W = 0.5 * \kappa * \frac{u_{10}^2}{gd} * F * cos\phi \tag{5.6}$$

where:

W	Wind setup $[m]$
κ	Friction constant $[-]$
u_{10}	Wind velocity at 10m height $[m/s]$
g	Acceleration due to gravity $[m/s^2]$
d	Water depth $[m]$
F	Fetch $[m]$
ϕ	Angle between the land and wind $[rad]$

The friction constant has been fitted and [70] found that a suitable value for the systems in this region was: $\kappa = 3.4 * 10^{-6}$. The wind velocities are taken from measurements in IJmuiden, level with the coastline [71]. The section of the NZK that is exposed to the wind effects has a depth of 15m, which does not fluctuate significantly throughout the year. Finally, the fetch depends on the angle of the wind, and the directions for which the fetch is non-zero can be seen in Table 5.4.

Table 5.4: Fetch in the NZK for the important wind directions

Direction (0° = North, 90° = East, etc.)	Fetch [km]
90°	1.5
100°	4.25
110°	19.25
120°	2.0
270°	-1.5
280°	-4.25
290°	-19.25
300°	-2.0

6

Methodology

The RL model has been set up as shown in Figure 6.1. The agent chooses an action that influences the water system and these changes are determined with the model of IJmuiden as described in Chapter 5. The new state of the water system determines which rewards the agent receives, calculated using a reward function. Finally, the agent receives the new state of the water system, including which actions are currently possible, along with the reward for its action.

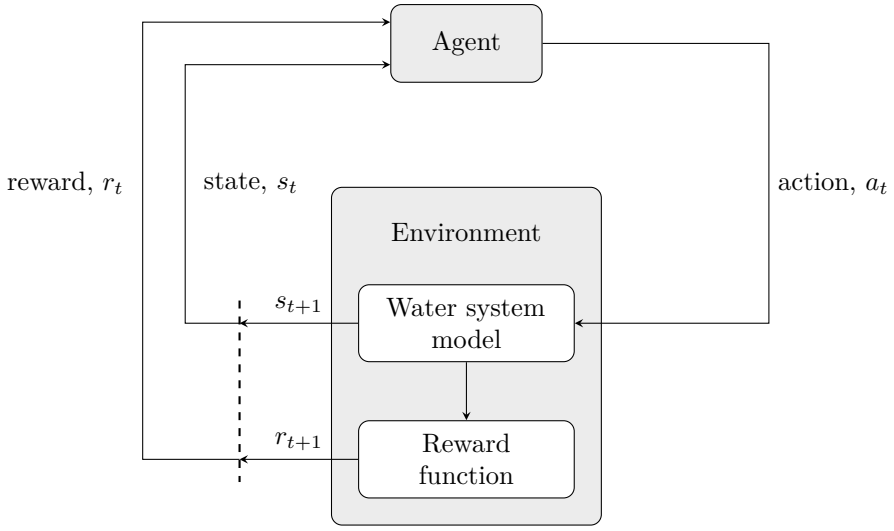


Figure 6.1: RL agent interaction with the environment and reward function

The RL agent selects the action based on the estimate of the q-function, with a time step of 1 hour. The q-function is approximated using a neural network that receives the state as input and returns the expected return for each action. The agent will choose the action with the highest expected return that can be performed in the current state.

6.1. Reinforcement learning method

The TensorFlow library for applied deep reinforcement learning, Tensorforce [17], that was introduced in [72], was used for the implementation of the RL controller. The advantages of the library were that a large number of algorithms were implemented as well as an intuitive setup of the package.

Below is an overview of the main elements that were implemented specifically for the water system in IJmuiden. This includes the features in the state space, the actions and how masking was performed, and the reward structure.

6.1.1. State space

The state space represents the current configuration of the water system and should contain all the necessary information for the agent to make decisions about which action to take. The state contains the following data, which is scaled before being returned to the agent. The state space contains 72 inputs and the size of each is shown in brackets.

- The current water level $[m]$ (1)
- The current wind set-up $[m]$ (1)
- The current inflowing discharge $[m^3/s]$ (1)
- A forecast (48 hours) of the inflowing discharge $[m^3/s]$ (6)
- The current water level in The North Sea $[m]$ (1)
- A forecast (48 hours) of the high and low tides $[m, hrs]$ (14)
- The DAM electricity price for 48 hours ahead $[EUR/MWh]$ (48)

The data for the inflowing discharges and sea levels were made available by [65], the wind set-up data by [71], and the DAM prices by [41].

Water level

The current water level was calculated using the linear reservoir model shown in Equation (5.1). The initial water level in the system was set to the target water level in IJmuiden of $-0.40m+NAP$. During training, the water level was initiated randomly to explore more of the state space including water levels exceeding the target range. This allowed the agent to learn the optimal behaviour when the water level was both above and below the target range.

The wind set-up was added to the water level in the state in order to make sure that this feature was the actual water level. This produced better model performance than leaving the wind set-up completely separate in the state due to the infrequent occurrence of set-ups.

Wind set-up

The conditions needed for large wind set-ups are rare and the influence on the water level is gradual. Between 2014 and 2022 there was a set-up during 2.2% of all days with a maximum water level deviation of $0.04m$ [71]. For these reasons, no forecast was given to the agent, only the current set-up. The set-up was kept as noise on the water level that the model could learn to take into account. The current wind set-up also gave the agent an indication of short-term set-ups due to the high correlation with previous time steps.

Discharge

The inflowing discharge consisted of the discharge in the ARK at Maarssen and the sum of the discharges of the waterboards after that point. The average discharges from each source area: Waternet: $9m^3/s$, Rijnland: $20m^3/s$, De Stichtse Rijnlanden: $7m^3/s$, Hollands Noorderkwartier: $6m^3/s$, Maarssen: $25m^3/s$. [65] The discharges in Maarssen and from Rijnland are the main contributors to the total discharge. The maximum discharges are also far higher in Maarssen and Rijnland.

The discharge information in the state consisted of the current discharge as well as a forecast for the following 48 hours. The forecast in the state did not include all available data as that would result in a large number of input variables, referred to as a high dimensionality. Generally, the modelling task becomes more challenging with more input features, known as the curse of dimensionality [73]. For this reason, average discharges over an 8-hour period were given. An example from 2019 can be seen in Figure 6.2.

By using the 8-hourly instead of hourly inflowing discharge, the feature size was reduced from 48 to 6. During the development of the RL model, different bin sizes and forecast lengths were tested to determine the suitable balance between feature reduction without losing necessary information for choosing actions.

The current model set-up uses the exact data or perfect forecasting as input. Further development of the model should include actual forecasts rather than exact data as there are sometimes significant uncertainties in the forecasts which can influence choices made by the agent. On the other hand, the RL model runs extremely fast (less than a second) which means that the computation can be repeated many times. First estimates of control could be made further ahead of time while the definitive control can be determined just prior to when actions are performed with very accurate forecasts.

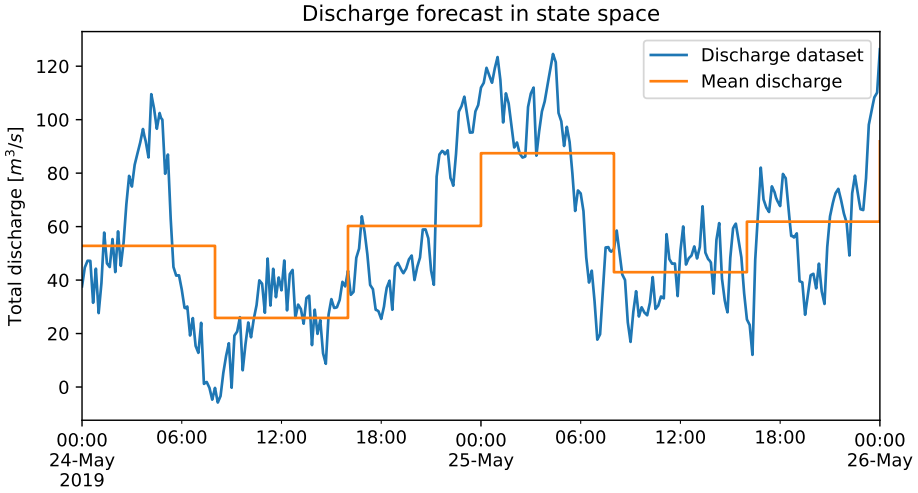


Figure 6.2: Discharge forecast for the state space (Dataset from [65])

Sea level

As with the discharge, the state contained the current sea level as well as a forecast that had been reduced in size. The North Sea level was measured in the IJmuiden Buitenhaven, near the pumping station located in IJmuiden, see Figure 6.3. As the water level was measured directly, the influence of the wind was included.

Figure 6.3: Location of North Sea level data

The sea level information was necessary for the agent to estimate when it was possible to use the gate, which has no costs associated with it. It was therefore important to know when low and high tides will occur as well as the water levels. Not all low tides allow the use of the gate. The agent received the number of time steps/hours to each low/high tide as well as the water level. Figure 6.4 shows the sea level forecast for the same 48 hours as in Figure 6.2. For a forecast length of 48 hours, 7 low/high tides were passed as input to the agent.

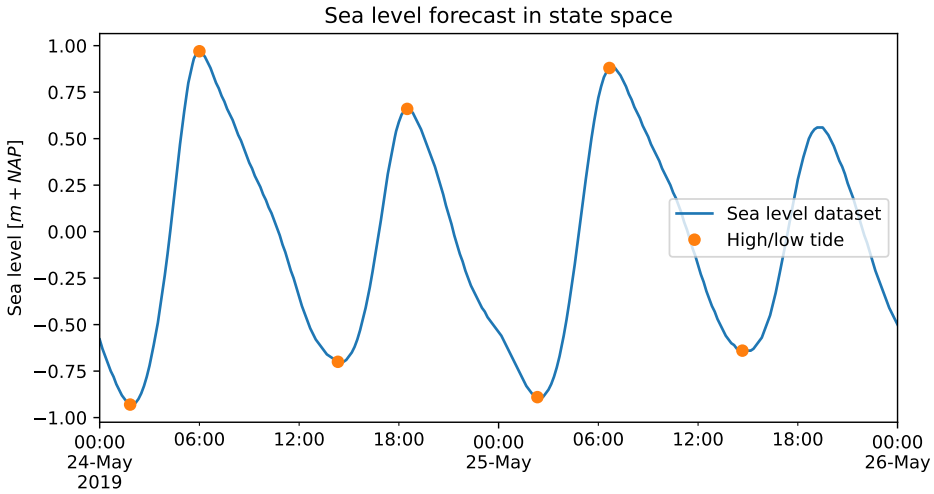


Figure 6.4: Sea level forecast for the state space (Dataset from [65])

During testing, it was found that performance improved with the reduction of the input size to purely the high and low tides instead of hourly sea levels.

Electricity price
The final component of the state space was the DAM price. This would allow the agent to choose the appropriate moment to pump if only using the gate was not sufficient. Due to the large fluctuation in DAM price, hourly prices were given, 48 hours ahead. Examples of 48 hour prices are shown in Figures 3.2 and 3.3.

To increase the size of the dataset for training, and therefore improve the model performance in more scenarios, DAM prices in Belgium and Germany were also used. The amount of renewable energy generation in Germany is increasing and currently, the portion of inflexible power generation is far higher than in The Netherlands. As a result, negative prices occur more frequently. In 2019 negative prices were observed on the DAM on 39 days [39]. Including these two markets allowed the agent to explore more of the state space and improve performance for more varying price scenarios.

Z-score normalization
When training a NN, the parameters/weights are small and updates are made based on the difference between expected and predicted values. If the input variables are not scaled, the learning speed can drastically decrease and training can be unstable. The scales of the data in the state space vary several orders of magnitude from $0.01 - 100$. Scaling the inputs during testing resulted in large performance increases and decreased training times.

The inputs were normalized using a z-score normalization where features are scaled to have a mean of 0 and a standard deviation of 1. This scaling was chosen rather than scaling all inputs to values between 0 and 1. When scaling to a range, outliers can cause a large portion of the data to be scaled to a very small interval.

The values used for the transformation of each type of input parameter can be seen in Table 6.1. These parameters were determined using the training dataset to prevent leakage of information about the test dataset.

Table 6.1: Mean and standard deviation used for z-score normalization of input parameters

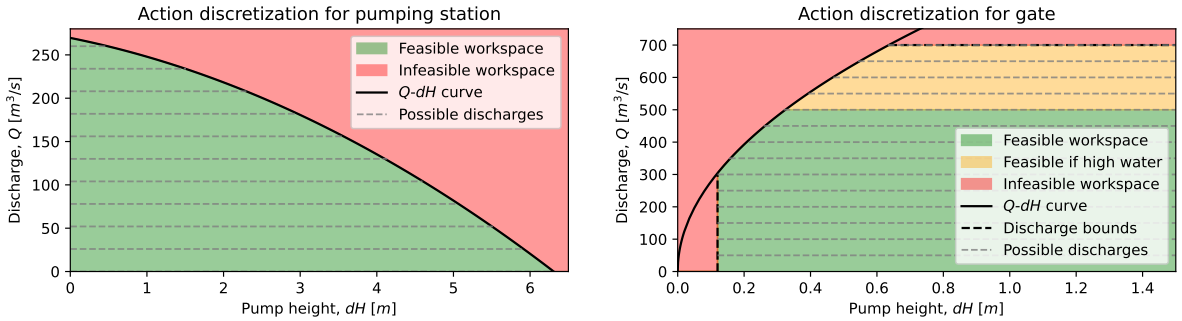
Input parameter	Mean (μ)	Standard deviation (σ)
Water level	$-0.40m$+NAP	$0.50m$+NAP
Wind set-up	$0.00m$	$0.04m$
Inflow discharge	$67.5m^3/s$	$48.4m^3/s$
Sea level (value)	$0.40m$+NAP	$0.56m$+NAP
Sea level (time)	$24.5hrs$	$13.9hrs$
Electricity price	48.7 EUR/MWh	38.47 EUR/MWh

6.1.2. Action space
The agent is able to choose three types of actions: no outflow, pumping, or opening the gate. No outflow is always possible while the possibility for pumping or using the gate depends on the water level in the NZK compared to that in the North Sea. Action masking allows the possible actions to be limited depending on the current state of the environment.

As described in Chapter 5, if the sea level is between $0.12m$ and $0m$ below the level in the water system, neither pumping nor opening the gate is possible. If the water level in the North Sea is at least $0.12m$ below that in the NZK, the gate can be chosen by the agent. The pumping station can be used when the sea level is at least as high as the NZK.

The DQN algorithm is not suitable for continuous action space and therefore the possible outflow discharges are discretized. There are 10 possible pumping discharges ($26m^3/s$ intervals) and the gate can be operated for outflowing discharges with $50m^3/s$ intervals, as shown in Figure 6.5.

It frequently occurs that the pumping station or gate can be used for a limited number of discharges. When this is the case, the relevant actions are masked and the agent is unable to choose these. At every time step, the agent not only receives the new state but also the action mask. This Boolean array contains a True/False for each action to indicate which actions can be selected in the next time step.



(a) Pumping station

(b) Gate

Figure 6.5: Action discretization for the pumping station and gate

Finally, some actions are forced during evaluation. When the model is evaluated no exploration is performed, which means that the agent always chooses the action with the highest expected reward. Current control includes halting all outflowing discharges when a water level of $-0.53m{+}NAP$ is reached to ensure no further decrease [7]. In addition, when the water level reaches $-0.32m{+}NAP$ the maximum discharge action is always performed. This aids the agent to minimize the exceedance of the upper boundary of the target range.

During testing, the agent performed far better when the actions were only forced during evaluation steps and not training. During training, the agent benefited from learning in states close to or outside the target range, without automatically choosing certain actions.

6.1.3. Reward structure

An essential element in the RL method for high performance and fast learning is the reward function. It works as an incentive mechanism and determines the final behaviour of the agent as it is the only indication of how good a chosen action was. The goal is to maximize the final return which can include making sacrifices in the current time step for higher rewards in the future.

After testing the agent for many reward structures with different components and relative scaling, the best performance was found with the following configuration. Before returning the total reward to the agent, the components are scaled.

Water level reward

The water level reward is split into two scenarios. All rewards are negative and can also be regarded as a penalty for the agent.

- Inside target range: When the water level is between $-0.55m{+}NAP$ and $-0.30m{+}NAP$, there is a slight incentive to bring the level closer to $-0.40m{+}NAP$. The reward is calculated as shown in Equation (6.1).

$$R_{inrange} = -(h_{NZK} - h_{target})^2 \tag{6.1}$$

where:

$R_{inrange}$	Reward due to water level when inside target range
h_{NZK}	Water level in the NZK $[m{+}NAP]$
h_{target}	Target water level in the NZK: $-0.40m{+}NAP$

- Outside target range: When outside the target water level range, the reward is the cost of pumping the maximum discharge ($260m^3/s$) with the maximum occurring price in the forecast of 48 hours. The reward is then scaled to increase when the water level is further from the target range boundary. This incentivizes the agent to remain closer to the boundary even when it has been exceeded. As discussed in Chapter 2, the further the water level rises above the target range, the more problems start to arise in the surrounding area.

$$\begin{aligned} R_{above} &= -max\left[E_{max} * P_{max}, 100\right] * (1 + h_{NZK} - h_{max}) && \text{If above } -0.30m{+}NAP \\ R_{below} &= -max\left[E_{max} * P_{max}, 100\right] * (1 + h_{min} - h_{NZK}) && \text{If below } -0.55m{+}NAP \end{aligned} \tag{6.2}$$

where:

R_{above}	Reward due to water level when outside target range
E_{max}	Maximum DAM price in 48 hour forecast $[EUR/MWh]$
P_{max}	Power consumption for pumping $260m^3/s$ $[MWh]$
h_{max}	Maximum water level in the NZK: $-0.30m{+}NAP$
h_{min}	Minimum water level in the NZK: $-0.55m{+}NAP$

The maximum price in the forecast is used instead of that in the current time step to reduce the fluctuations in the reward. As negative prices can occur, a minimum cost of $100EUR$ is set.

Pumping reward
Pumping is penalized with the cost calculated with the current DAM price and chosen discharge. This does mean that the agent can achieve positive rewards when the pump is used during periods when the electricity price is negative.

$$R_{pump} = -E * P(Q_{pump}) \tag{6.3}$$

where:

R_{pump}	Reward due to chosen pump discharge
E	DAM price $[EUR/MWh]$
$P(Q_{pump})$	Power consumption for chosen pump discharge $[MWh]$

Using the gate is not penalized as the power consumption is negligible compared to that of the pumps.

Scaling
To convey the relative importance of the reward components, the rewards are scaled accordingly. The scaling was tested by retraining the model with different scaling factors to find the best performance. As the performance could not be assessed by comparing the reward (as this kept changing with scaling), the costs of pumping and the percentage of time steps that the water levels were outside the target range were compared. Using those assessment parameters, the best performance was achieved with the scaling shown in Equation (6.4). This results in an order of importance, as shown in Equation (6.5).

$$\begin{aligned} R_{inrange} &= 500 * -(h_{NZK} - h_{target})^2 \\ R_{above} &= 10 * -max\left[E_{max} * P_{max}, 100\right] * (1 + h_{NZK} - h_{max}) \\ R_{below} &= 10 * -max\left[E_{max} * P_{max}, 100\right] * (1 + h_{min} - h_{NZK}) \\ R_{pump} &= -E * P(Q_{pump}) \end{aligned} \tag{6.4}$$

$$\text{Rewards in order of importance: } R_{above}/R_{below} \rightarrow R_{pump} \rightarrow R_{inrange} \tag{6.5}$$

With this scaling, the most important objective was maintaining a water level within the target range. Exceeding the target range was penalized with a cost of at least ten times the maximum cost of pumping. The penalty linearly increased as the water level deviated further from the boundary of the target range. This ensured that water safety was not be compromised for energy savings. This factor of 10 for the trade-off was also found to be suitable in the control of seasonal thermal energy storage systems [3].

When the water level was within the target range, the penalty for pumping was the most important. Finally, the reward when not pumping was purely the incentive towards the target water level of $-0.40m{+}NAP$. The factor for the in-range penalty was determined by comparing model performance

to ensure that an efficient pumping strategy resulted in the lowest reward. The in-range reward was only significant when using the gate or not draining any water from the system.

Finally, as with the input parameters of the neural network, the learning is generally more stable and faster if the training target is around 1 in magnitude. Therefore, the total reward is scaled to be the appropriate order of magnitude.

$$R_{total} = 10^{-3} * ([R_{inrange} \text{ or } R_{above} \text{ or } R_{below}] + R_{pump}) \tag{6.6}$$

Discussion
Care needs to be taken when using negative rewards to train an agent. When there are large regions in the state space dominated by negative rewards, the agent may learn to avoid negative rewards by reaching the terminal state as fast as possible. Currently, the episode terminates after a pre-defined number of time steps or with the water level exceeding $-5.0m$ or $5.0m$. These extreme water levels are only reached during the first training runs. At the start of training, the agent makes almost random choices as the estimates of the value function are initially far from the actual values. It was not observed that actions were chosen in order to reach the terminal state as fast as possible.

During testing, other reward structures were considered. Initially, the time step was set to 15 minutes which meant that it was not feasible to change the pump discharge at every time step. It is not possible to reach the desired discharge within that short time frame and constantly changing the discharge of the pumps will damage the machines, reducing their lifetime and increasing maintenance. This was solved by including an additional penalty for choosing a different action than the previous choice. Increasing the time step to 1 hour and removing the action change penalty proved the best solution. This increased the performance of the model with regard to the costs and maintaining a water level inside the target range. This is however something to consider as changing the action hourly may still be more frequent than desired.

When considering maintenance and wear of pumps, it may be beneficial in further development of the model to quantify these effects in terms of costs. This can allow the agent to make a trade-off between the cost of pumping and maintenance costs that may increase with certain use of the pumps.

6.2. Input data

Training/input data for the IJmuiden pumping station was available between 2015 and 2022, which gave the agent a large set of examples from which to train. In addition to the data, discharges were also generated, as described in Appendix C, to allow the agent to encounter more examples of high discharges, which do not occur as often in the data. If the agent doesn't train on enough examples of a certain situation, the model performance will reflect this.

The state space was explored further by initializing the water level outside the target range. This allowed the agent to learn which actions were best in those situations. When evaluating the performance, the water level was initiated at the target level of $-0.40m$+NAP.

To increase the examples of DAM prices, data from Germany and Belgium were also used. This also allowed the model to encounter more examples of negative electricity prices which are currently less likely in The Netherlands. Combining different electricity prices with the same discharge data creates many more samples for training. This can also be achieved by combining discharge data from one year with other inputs in the state from another year.

In addition to a large number of data samples, experiences can vary due to different choices made by the agent. The choices of the agent have a significant influence on the water level and therefore the actions that are possible. As a result, the same input data can produce a very different system state for the agent to deal with.

6.2.1. Train, validation, test split

When training a NN the available data is split to accurately evaluate the performance of the model. The training data was used to train the model and during training, the validation set was used as an unbiased of the model performance. The validation set was therefore never used for the agent to *learn* and only affects the model indirectly by giving insight into the current performance. This allowed changes to be made during training or to determine whether training could be stopped. Finally, after training the model, performance was evaluated with the unseen test data. To ensure that the test set gave an

unbiased reflection of the model performance, parameters in the models, such as in normalization, could only be based on the training set. This prevented leakage of information about the validation and test data.

The split of the data for training, validating, and testing was made based on the wet-/dryness and the electricity cost. The data was split into wet and dry clusters per month as well as cheap and expensive. The data was subsequently split to include an approximately equal distribution of all categories in the training and test set. A portion of the training set was kept aside for validation. The method used to cluster the data and divide the data for testing, validation, and training is elaborated on in Appendix D.

Using the clusters found in the data, the following splits were made for the training, validating, and testing phase. When testing and validating only discharges and water levels from data in IJmuiden were used in combination with DAM prices from The Netherlands. The training was done with discharge data as well as simulated data. Sufficient data for the sea level was available to purely use data from the North Sea outside the port of IJmuiden. Finally, the DAM prices from The Netherlands that were not used for testing or validation were combined with data from Belgium and Germany for training. The country from which the electricity prices were sampled was selected randomly with a probability proportional to the size of the available dataset.

- Testing: 01/01/2019 - 01/01/2021
  - All inputs were measurements from IJmuiden combined with DAM prices from The Netherlands
- Validation: 01/01/2017 - 01/01/2018
  - All inputs were measurements from IJmuiden combined with DAM prices from The Netherlands
- Training set: All remaining data
  - Discharges: 01/06/2015 - 13/09/2021 (excluding dates above) or simulated data
  - Sea level: 31/12/2014 - 28/07/2022 (excluding dates above)
  - Electricity prices (NL): 05/01/2015 - 18/07/2022 (excluding dates above)
  - Electricity prices (BE): 05/01/2015 - 18/07/2022
  - Electricity prices (DE-LU): 01/10/2018 - 18/07/2022

Finally, test scenarios were selected to evaluate the behaviour of the model in specific situations. Scenarios for the discharge, sea level, and electricity prices that required different control behaviour were chosen, as described in detail in Appendix E.

6.3. Learning procedure

The agent learns the behaviour for the region of the state space where experiences are sampled. When the estimate of the value function is not yet accurate, the agent quickly allows the water level to exceed the desired boundaries. If the episode length is too long, a large portion of the experiences are examples of states outside the target water level range. Therefore the agent learns what to do outside the boundaries but is still inadequate for control inside the boundaries. The episode length is gradually increased during training to keep a large enough portion of the experiences within the desired control region.

To optimize control, the most important factor is keeping the water level within the target range. The agent is initially trained with only the water level component of the reward. This means that training is initially focused purely on keeping the water level in range and later the agent is trained further to take pumping costs into account. This speeds up training as learning a single objective is easier for a RL agent. The steps in the training are shown in Table 6.5.

To provide an indication of the current performance, the model is evaluated every 50 training steps. This is done the with validation dataset. Only a limited number of evaluations are performed, 20 episodes, to reduce overall training time. The small evaluation size results in a large uncertainty in the predicted performance and therefore a larger evaluation is performed after every training step, with 100 episodes.

Only after the full training is complete is the final performance of the model calculated using the test dataset, for 1000 episodes.

Table 6.5: Agent training steps

Training step	Number of epochs	Episode length [*days*]	Reward
1	500	2	Only water level objective
2	500	5	Total
3	500	10	Total
4	1000	15	Total
5	1000	15	Total
6	1000	15	Total

6.4. Hyperparameters

The performance of a RL can be drastically improved by optimizing the hyperparameters. Each system is different and there is no best set of hyperparameters that is suitable for all situations. The model was first developed to achieve a sufficient performance, after which a hyperparameter optimization was done, as described in Section 4.3. Appendix F shows the results of the optimization that was performed after the initial model was developed, as well as the final model, including the cost objective. Below is an overview of the optimal parameters found for the final model.

- Batch size: 20
  The number of training samples over which the update to the NN weights was computed.
- Update frequency: 0.15
  The frequency, relative to the batch size, with which the target network was updated to the current parameters of the prediction network.
- Replay memory capacity: 2500
  The size of the buffer of experiences from which a random batch of samples was taken for training.
- Learning rate: 7e-4
  The learning rate of the Adam optimizer, presented in [74], used for the iterative updating of the NN weights. The optimizer computes adaptive learning rates for each network parameter based on the moments of the gradients.
- Horizon: 2
  In n-step DQN: the number of steps ahead for which the discounted-sum reward was used before the target network estimate.
- Discount factor: 0.75
  The discount factor (γ) used in the DQN learning.
- Exploration: 0.3
  The probability of selecting an action at random when generating training samples.
- Policy network configuration:
  - Layers: 4 fully connected
  - Neurons: 16
    The number of units in each layer.
  - Activation function: tanh
  - Weight initialization: Glorot normal
    The method for initializing the weights, suitable for a network with the tanh activation function.
  - Dropout rate: 0.25
    The fraction of units that were dropped from each layer during the training phase. These neurons were not considered during the forward or backwards pass.

It is important to consider the the number of parameters (weights and biases) of the model that need to be trained, compared to the number of available observations of the system. The number of observations needs to be greater to prevent overfitting. The NN architecture contains a weight for each connection between neurons, in addition to a bias for each neuron that is not in the input layer. Table 6.6 shows the number of coefficients to be trained, the data available, and the number of observations used during training. This shows that there is sufficient data used during training to prevent overfitting.

Table 6.6: Comparison of model coefficients, available observations, and training observations

Model coefficients	Available training observations		Observations trained
2,409	Wind set-up and sea level	40,080	1,284,000
	Inflowing discharge	40,080 + generated	
	DAM electricity price	139,368	

6.5. Training speed

The complexity of the system results in the need for a large number of training epochs to reach the desired performance. When using the hardware (CPU/GPU) in a laptop this will take a very long time to complete training. For this reason, the training of the RL agent was done on The Dutch National Supercomputer Snellius. The speed of the available processors greatly reduced the training times, which sped up the model development. In addition, hyperparameter optimization was made possible as it required around 300 models to be trained from scratch, which would not have been possible without this speedup. The agent could be fully trained 3 times as fast on the supercomputer.

A job script was used to queue a training task. An overview of the Slurm settings used can be seen in the job script in Appendix G.

7

Results

This chapter shows the performance of the proposed RL model. Firstly, in Section 7.1 the control plan created by the agent is explained to help understand the setup in later analysis. Section 7.2 shows how the model deals with the classical constraints for the water level. The results, including energy cost objectives, are presented in Section 7.3. The controller was compared to a state-of-the-art MPC controller developed by [64] in Section 7.4 and tested on several extreme scenarios in Section 7.5. Finally, an alternative reward structure was tested and presented in Section 7.7.

After training the agent, it was able to produce a control plan of any length, from hours to months. The final output of the model is an hourly control plan of pump and gate discharges for the entire input length. The current state of the water system, forecasts of the sea level and inflowing discharge, and DAM prices determine the model output.

7.1. Understanding the control plan

The agent makes choices about the outflowing discharge using the pump and gate. This results in a change in the water level in the system, which influences which actions are possible in subsequent time steps. A control plan is visualized in Figure 7.1, showing the important in- and outputs of the model.

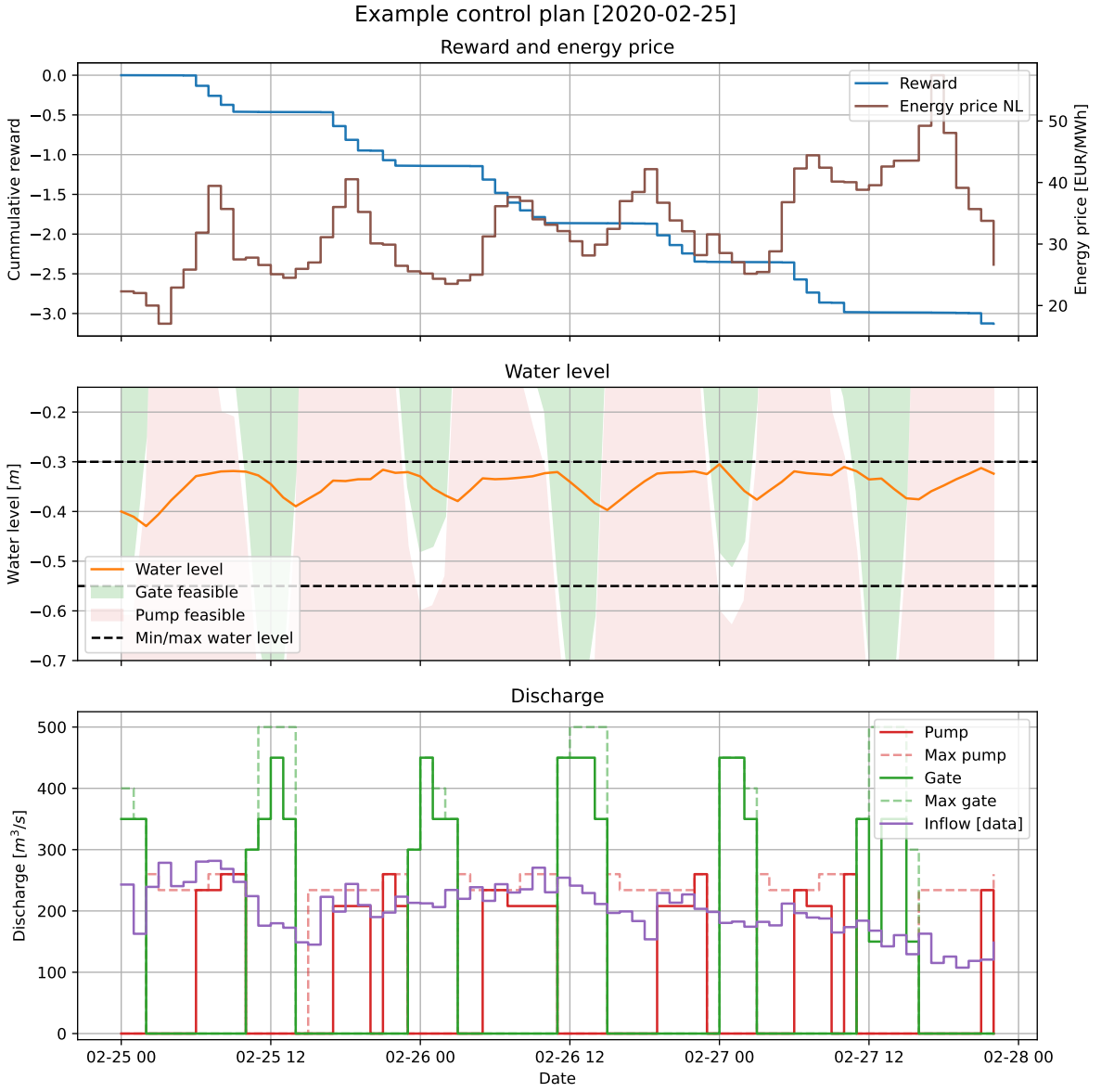


Figure 7.1: RL control plan for three days starting on 2020-02-25

The top figure shows the cumulative reward in blue, which demonstrates the negative rewards associated with the costs of pumping. The magnitude of the reward depends on the DAM price, which is shown in brown, and the power consumption of the pumps. The reward decreases monotonically unless the agent chooses to pump during a time step at which the electricity price is negative. This is the only situation where the agent can receive a positive reward. In the middle figure, the water level in the NZK is plotted in orange, with black dashed lines indicating the boundaries of the target water level range. The red and green shaded areas show when pumping and opening the gate are possible respectively, and no shading marks when neither action is possible. The bottom figure displays the chosen actions with the pump and gate discharges in red and green, respectively, with dashed lines to indicate the maximum possible discharge. Finally, the purple line shows the total inflowing discharge into the system. All figures share the x-axis showing the date of the input data.

7.2. Water level objective

The NZK-ARK system is a watercourse that runs through the major metropolitan areas of Utrecht and Amsterdam. Respecting the safety regulations regarding the permitted water levels is therefore essential. It was necessary to first develop a model that was able to adhere to these constraints before including the cost optimization. Saving costs is only desired when there is no risk of violating water level boundaries.

To run the model with purely the water level objective, the penalty for pumping was set to zero while keeping the rest of the reward structure the same as described in Section 6.1.3. This meant that there was a slight penalty when inside the target range to incentivize the model to maintain a water level close to $-0.40m{+}NAP$. Outside the target range, the penalty was at least an order of magnitude larger to ensure that it was never advantageous to exceed the target range boundaries. The penalty outside the target range was still correlated with the price of pumping the maximum discharge. The dynamics of the water system were identical to those in the full model, including the masking of the pumping and gate actions.

As there was no penalty difference between pumping and opening the gate, this was a simple optimization problem. The agent only needed to take into account the water level and not the different action possibilities. This meant that the agent could be trained quickly, approximately twice as fast as the cost optimization, as fewer epochs were needed. The training was completed in around 45 minutes, rather than 80 minutes. A control plan from the test data set is shown in Figure 7.2.

The top figure shows how the choice of pumping is currently not penalized but that the reward is purely based on the water level in the system. The model has learned that maintaining a water level in the lower region of the target range was safest to prevent exceedance even though this resulted in a slight negative reward due to the deviation from the target water level. This does suggest that the forecast of the inflow is not optimally taken into account. The forecasts show that the inflow remains limited, which means that there is no risk of exceeding the upper boundary.

The agent had a high level of performance with an average of 0.2% time steps out of range for episodes of one month (720 time steps). The worst performance of the entire test set was 2% of the time steps out of range, which occurred during extremely high inflowing discharges. These statistics were determined by training multiple models with different initializations and determining the average of all trained models. The maximum water level reached was $-0.28m{+}NAP$, only $2cm$ above the target range. This high water situation would not result in any measures in the area. At $-0.20m{+}NAP$, the IJ-front would be closed off due to the increased risk of flooding in Amsterdam. Even though no measures would be taken with an exceedance of $2cm$, this will still likely cause alarm and operators will try to restore safe water levels as quickly as possible. The time period that the upper boundary was exceeded was very short, less than 1 hour. The expected length and severity of the water level exceedance are shown in the control plan, which also means that operators can anticipate this event.

During normal conditions, the lower region of the water level range was desirable for the model due to the extremely high consequence of exceeding the upper boundary. The model learned that the slight negative reward due to the deviation from $-0.40m{+}NAP$ was beneficial, as the penalty for exceeding the boundary was relatively a lot larger. During extreme inflowing discharges, there would be a buffer in the system to overcome the highest peaks. It is possible for the inflow to be larger than the maximum outflow discharge, at which point such a buffer would be necessary to ensure safe water levels. However, the necessary information about high inflowing discharges in provided in the forecast, which means that this is not adequately taken into account by the model.

There were also situations where it was not possible to maintain such a low water level due to higher inflowing discharges. The agent was still able to keep the water level below the upper boundary. An example of this can be seen in Figure H.1. This also shows that the agent makes use of the gate, which is not the case in the previous example. Each new instance of the agent resulted in slightly different behaviour after the training was completed. The initialization of the NN weights, as well as the random sampling of experiences, cause the agent to behave slightly differently. The agent that created the control plan in Figure 7.2 trained to generally use the pump unless high inflowing discharges occurred and the water level started to rise. This behaviour was observed in most of the trained agents, likely caused by the larger percentage of time steps where pumping was possible rather than using the gate. In addition, the pumps were able to drain sufficient volumes for a large portion of the inflow situations.

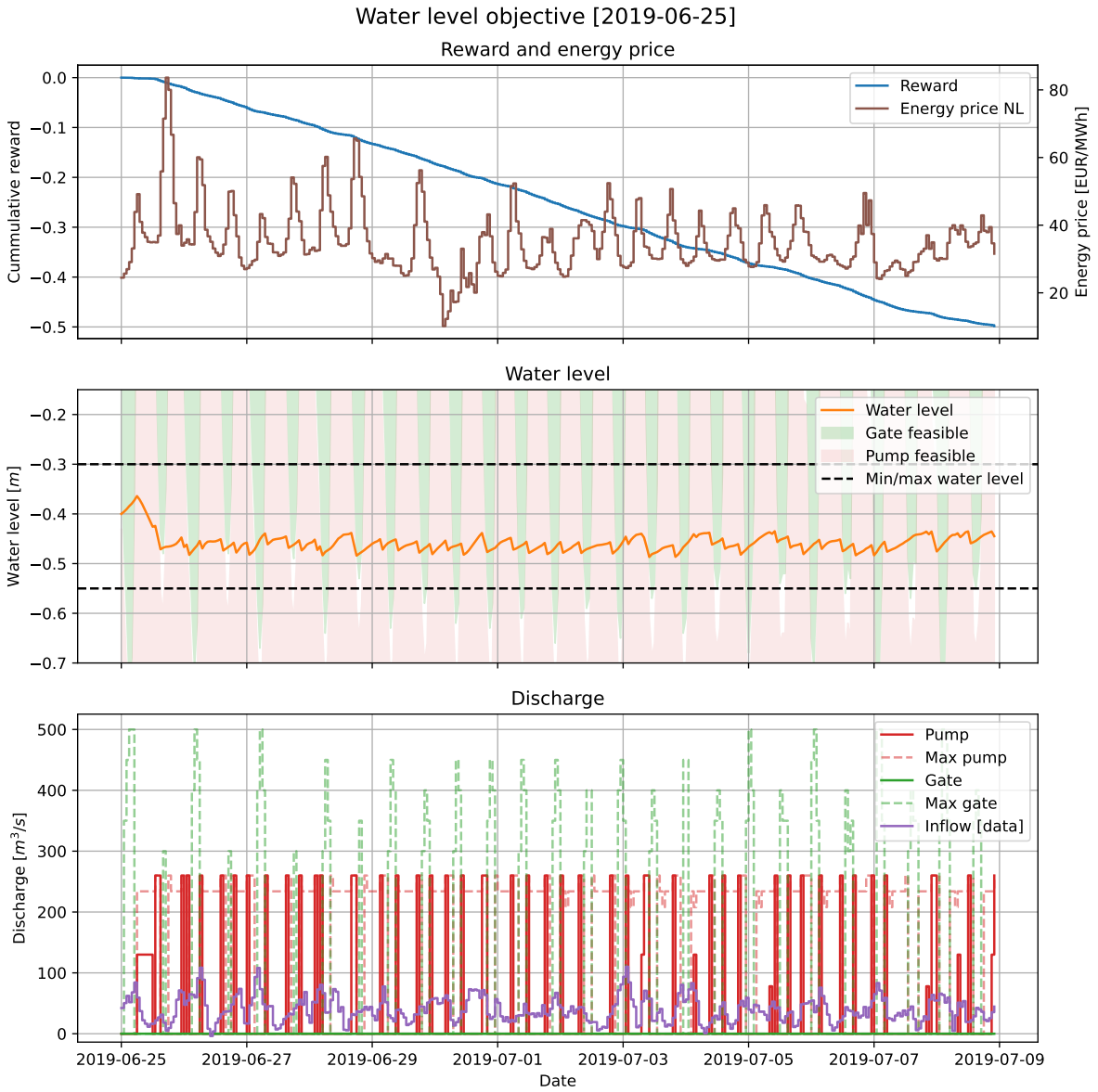


Figure 7.2: RL control plan for two weeks starting on 2019-06-25 with only the water level objective

The development of this model allowed the agent to be tested for a less complex optimization problem to ensure that it was able to learn and reach acceptable performance. This also allowed the water system to be tested to check whether the dynamics of the water system and the reward structure were implemented correctly. When the agent achieved the desired performance, an initial Bayesian hyperparameter optimization was completed to tune the model to the specific problem. The results of the hyperparameter optimization are shown in Appendix F. A significant speedup of a factor 2 and slight performance increase was achieved, which sped up the development of the full model including the cost objective. The agent and environment were very similar for the two models, as the dynamics of the system and in- and outputs of the model were the same. It was only necessary to tune the hyperparameters slightly when the full model was developed.

7.3. Cost objective

After showing the ability of the model to respect the water level restrictions, the cost objective could be included. The high energy consumption of the IJmuiden pumping station cost Rijkswaterstaat 700,000 euros yearly in 2005 [7]. This is expected to have only increased since then due to increased electricity prices. The target water level range gave sufficient flexibility to reduce energy costs within these bounds while ensuring water safety in the surrounding areas. As described in Section 6.1.3, the reward was split into three components to indicate the desired behaviour inside and outside the target water level range and regarding the use of the pumps.

7.3.1. Overall model performance

First, the overall model performance was examined by testing the model on the entire test data set. This was done by running 1000 tests of one month, where the initial water level was set to $-0.40m{+}NAP$. Due to the large number of tests, the months overlapped but the water levels differed between test months for data from the same dates. Only the initial water level was set, which subsequently changed during the month and greatly influenced the action choices. The water level determines which actions are possible as the relative level compared to the sea level is used to calculate the maximum outflowing discharge and whether pumping, opening the gate or neither is possible.

During the 1000 test episodes, the lower boundary of the target range was never exceeded. This constraint was simple to comply with as generally there are only inflowing discharges. It rarely occurs that there is there an outflow (often less than $-20m^3/s$) and the model could easily anticipate this. In addition, the model was set up to prevent any outflowing discharges when a water level of $-0.53m{+}NAP$ was reached to ensure no further decrease in the water level. This was done during evaluation, as in the training phase, no actions were forced automatically.

Over all the test cases of 1 month, the water level exceeded the upper water level boundary an average of 1% of the 720 time steps. In the month where the model's control plan caused a high water situation for the longest time period, the water level was out of range 2% of the time. This occurred during extreme inflowing discharges above $300m^3/s$ on 02-23, 2020.

Largest water level exceedance

During the high inflow on 02-23, 2020, the maximum water level was $-0.27m{+}NAP$, the highest reached during all tests. Measurements in IJmuiden showed that, during the event, the actual water level reached a maximum of $-0.32m{+}NAP$, remaining within the target range [65]. However, 3 days of measurements were missing around this time, so higher levels may have occurred. The results show that the model still produced water levels that were higher for a longer period of time and most likely exceeded the upper boundary by a greater margin. It is possible that during this event further measures were taken by Rijkswaterstaat to prepare for the event that may not have been included in the RL model. The dynamics of the system are also greatly simplified in the model used by the RL agent.

Comparison with the results from the MPC gave more insights as the same dynamics are used for the water system. The water levels produced by the two models, as well as the measurements in IJmuiden, are shown in Figure 7.3. The comparison with the measurements was used to compare general trends in water level changes rather than exact water levels. Full control plans of the RL model and MPC can be found in Appendix I.1.

The RL model and MPC showed similar behaviour with the exception of the day before the peak discharge and several days afterwards. The MPC anticipated the high discharges by lowering the water level during the two days before the peak. This allowed the water level to be kept within the target range initially. This behaviour can also be seen in the measurements. By not lowering the water level, the RL agent was not able to maintain safe water levels. However, after the initial high discharges, the MPC was less successful in continually ensuring the target water levels, compared to the RL agent.

Finally, significant differences can be seen between the measurements and the model results. The models both clearly show when the gate is opened as the water level decreases rapidly while the changes in the measurements are less drastic. There are several differences between reality and the simplified water system dynamics used by the models. The inflowing discharges are considered as a single discharge that enters the system with an immediate change to the water level. The linear reservoir model assumes that the water level changes instantaneously with a net in- or outflow in the system, in addition to the water level being the same throughout the entire system. However, the general behaviour does give an impression of the response of the water system to the chosen actions.

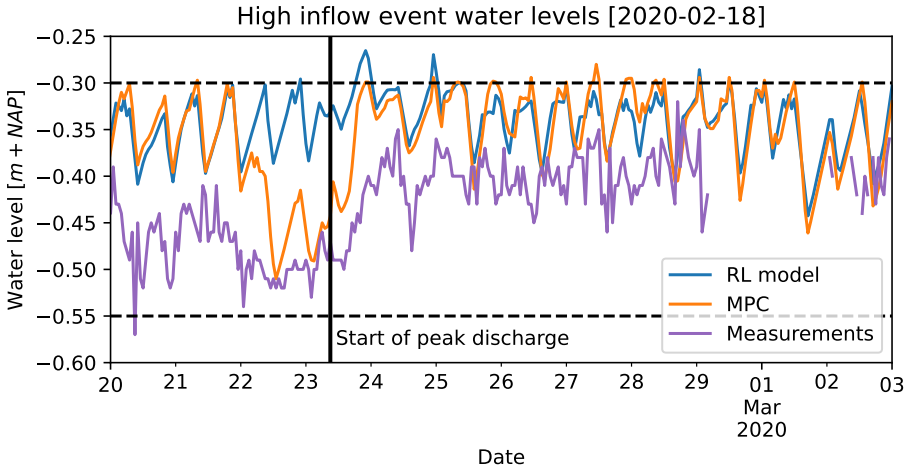


Figure 7.3: Hourly water levels from historical measurements, RL model, and MPC for high inflow event on 2020-02-23. Models were run from 2020-02-18 with an initial water level equal to the measurements.

Training process

The training phase of the agent gives insights into the model performance and whether this can be optimized further. It was difficult to observe the improving performance of the model through the development of the reward. The reward structure was not constant during training and the episode length was gradually increased. Instead, the percentage of time steps that the water level was outside the target range gave a more accurate impression of the performance. It is still important to consider that a longer episode length made the chance of exceeding the target range greater as the target water level was used to initialize the water system.

Figure 7.4 shows how the model learns as more episodes are observed. The different training steps can still be identified where the percentage out of range for the test data significantly increases. After 500 episodes, the cost objective was included and at 1,500 episodes, the episode length was increased to 15 days.

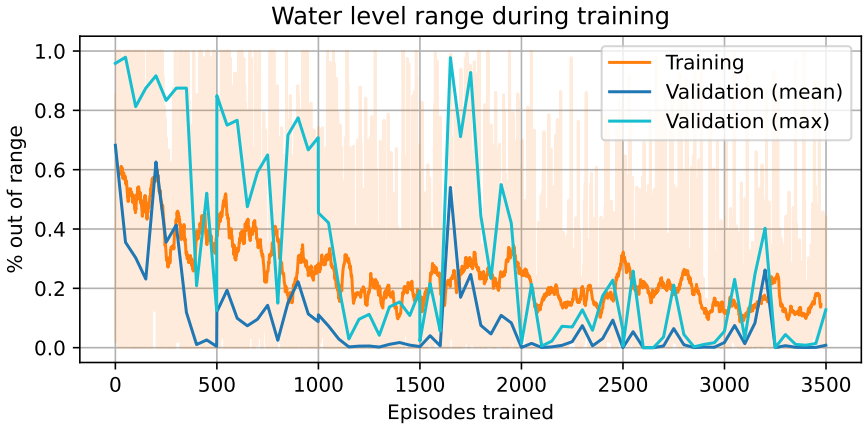


Figure 7.4: Agent performance improvement during training with the percentage out of target range for the training set which included exploration and validation set

The performance during training includes exploration, which means that the agent performs a random action with a probability of 0.3. During validation and testing, the exploration is set to 0.0 to give a more accurate impression of the current performance. Due to the importance of remaining within the water level range, the maximum exceedance was also included to show the worst performance of the model. To assess the robustness of the RL model, the consistency needs to be taken into account.

The improvement of the model is clear in the training and validation results. When training a NN

it is important to ensure that training does not continue too long with the risk of overfitting. This can be identified by a decrease in the validation/test performance while the training performance remains the same or improves. The training steps were found by comparing the final performance of many trained models. Each model turns out slightly different depending on the initialization and the random sampling of experiences. The average performance, therefore, gives an indication of how suitable the training steps are.

Figure 7.4 shows several large fluctuations in the validation performance, in addition to those that can be attributed to the changes in the reward structure or episode length. This suggests that the model is not yet fully optimized. Further improvements and more stable training can likely be achieved with a more suitable NN architecture, reward structure, and/or hyperparameters.

7.3.2. Normal conditions

To examine the model performance in more detail, control plans were made for specific types of situations. During normal conditions in the NZK, it is possible to maintain safe water levels without pumping. Such a control plan is shown in Figure 7.5.

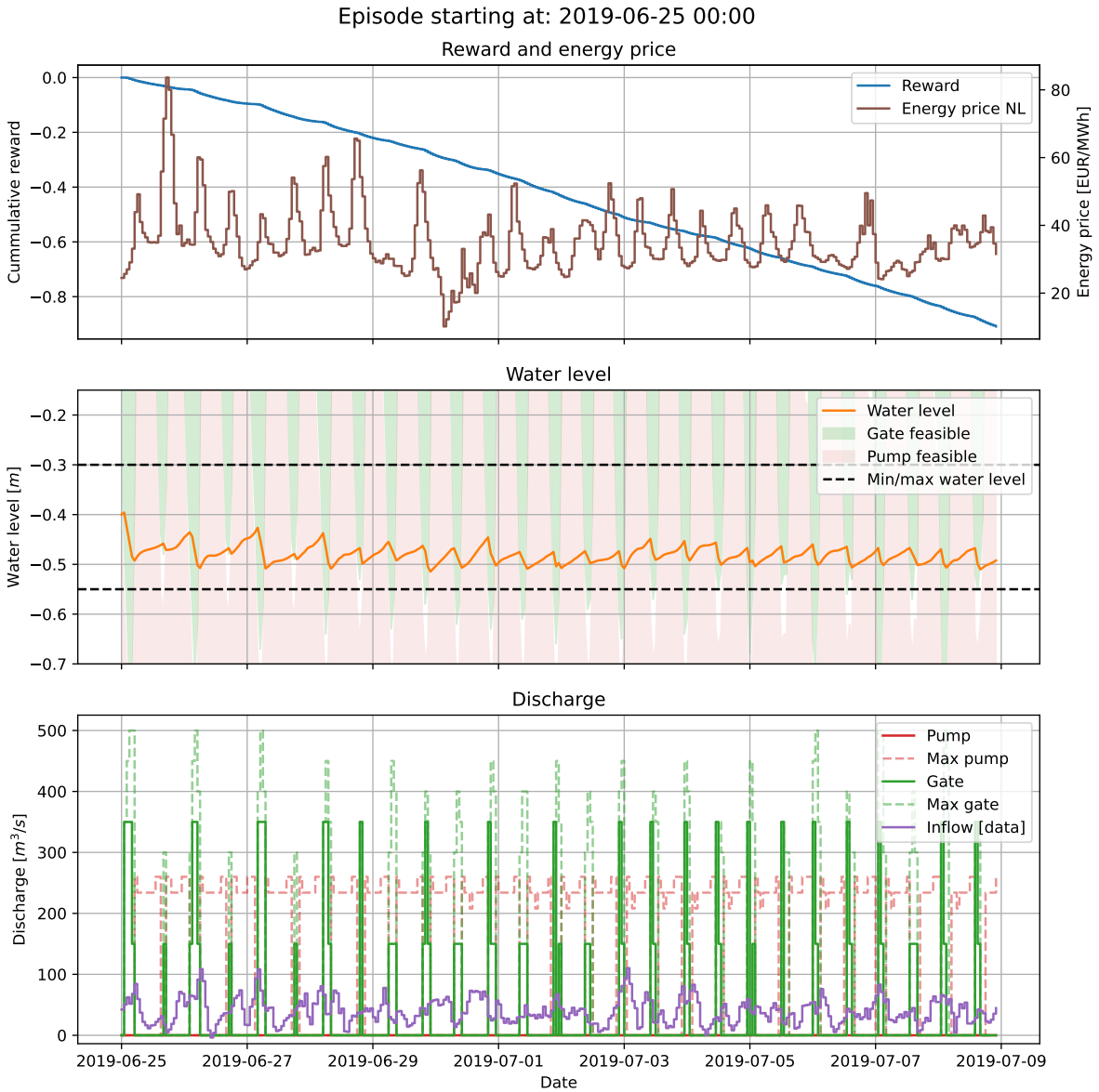


Figure 7.5: RL control plan for two weeks starting on 2019-06-25 where no pumping is necessary

The agent was able to ensure the water level remained within the target range by only utilizing the gate (shown in green in the lower graph of Figure 7.5). The water level tends to be kept near the lower boundary of the target range, as with the model including only the water level objective. The model learned that the risk of exceeding the upper boundary was reduced through this behaviour. Similar behaviour has been found throughout these calm conditions for the test data. These are simple optimization problems for the agent and there are many examples in the training data that allow the performance to be extremely high in such situations. On average, the water level is $-0.45m+NAP$ during these conditions, meaning the water level is kept $5cm$ below the target level on average. However, this is not always the case. The sea level determines whether the gate can be used for lower water levels. There are situations where maintaining a lower water level in the system reduces the possibility of using the gate. This makes it economical to allow the water level to rise closer to the upper boundary. An example of this can be seen in Figure 7.6.

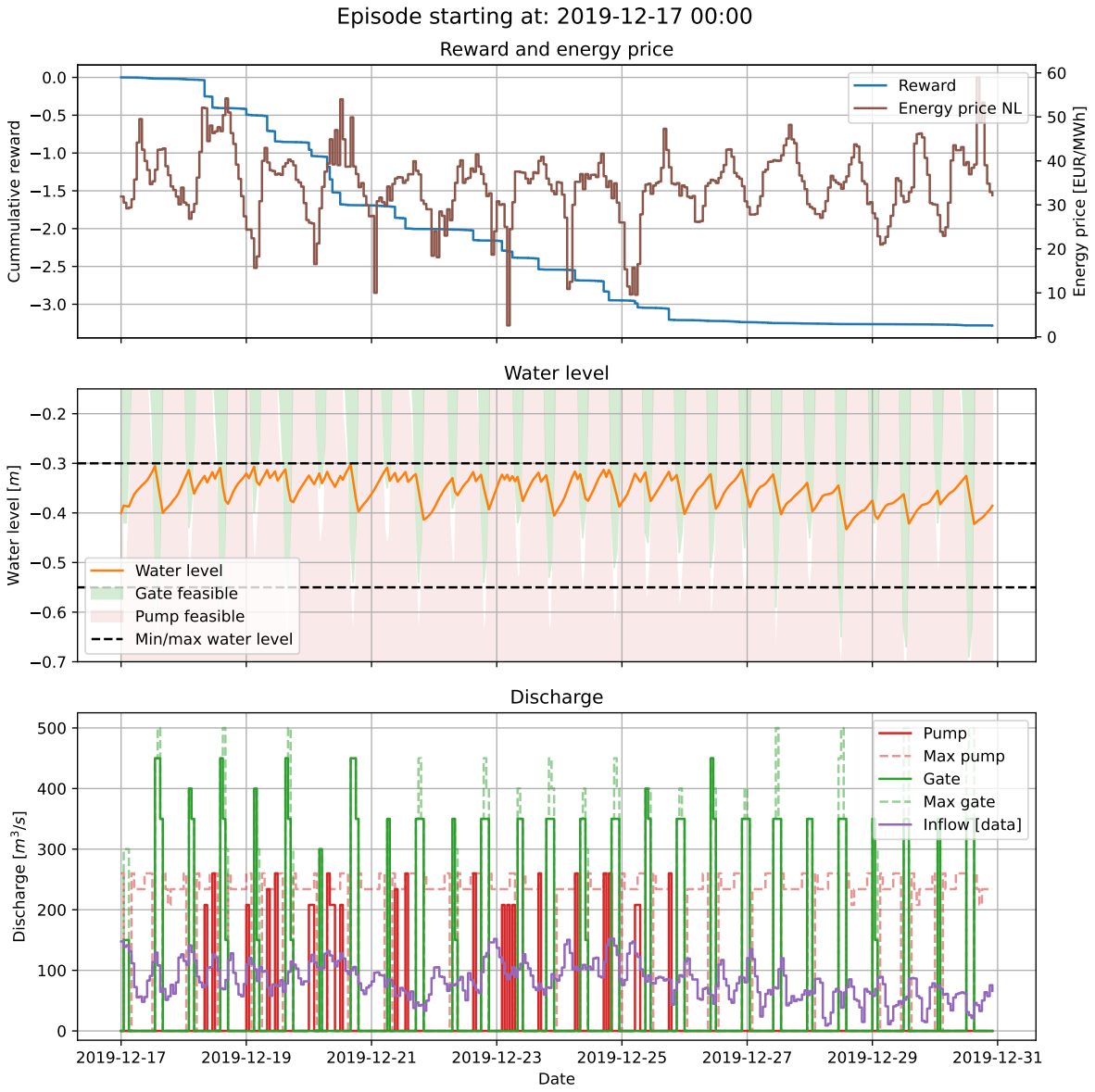


Figure 7.6: RL control plan for two weeks starting on 2019-12-17 where the water level was in the upper part of the target range due to higher sea levels

In the control plan for 2019-12-17, the inflowing discharge is also slightly greater, meaning that pumping is occasionally necessary. Despite this, the RL agent was still able to maintain the target water levels. It is also observed that after 2019-12-26, the sea level and inflowing discharge decrease

which allows the water level in the system to start to decrease. Pumping is also no longer necessary, which can be seen in the reward. If we zoom into the moments when the agent decided to use the pump, there is no clear correlation with the electricity prices. A zoom into the control plan between 12-19 and 12-21 is shown in Figure 7.7.

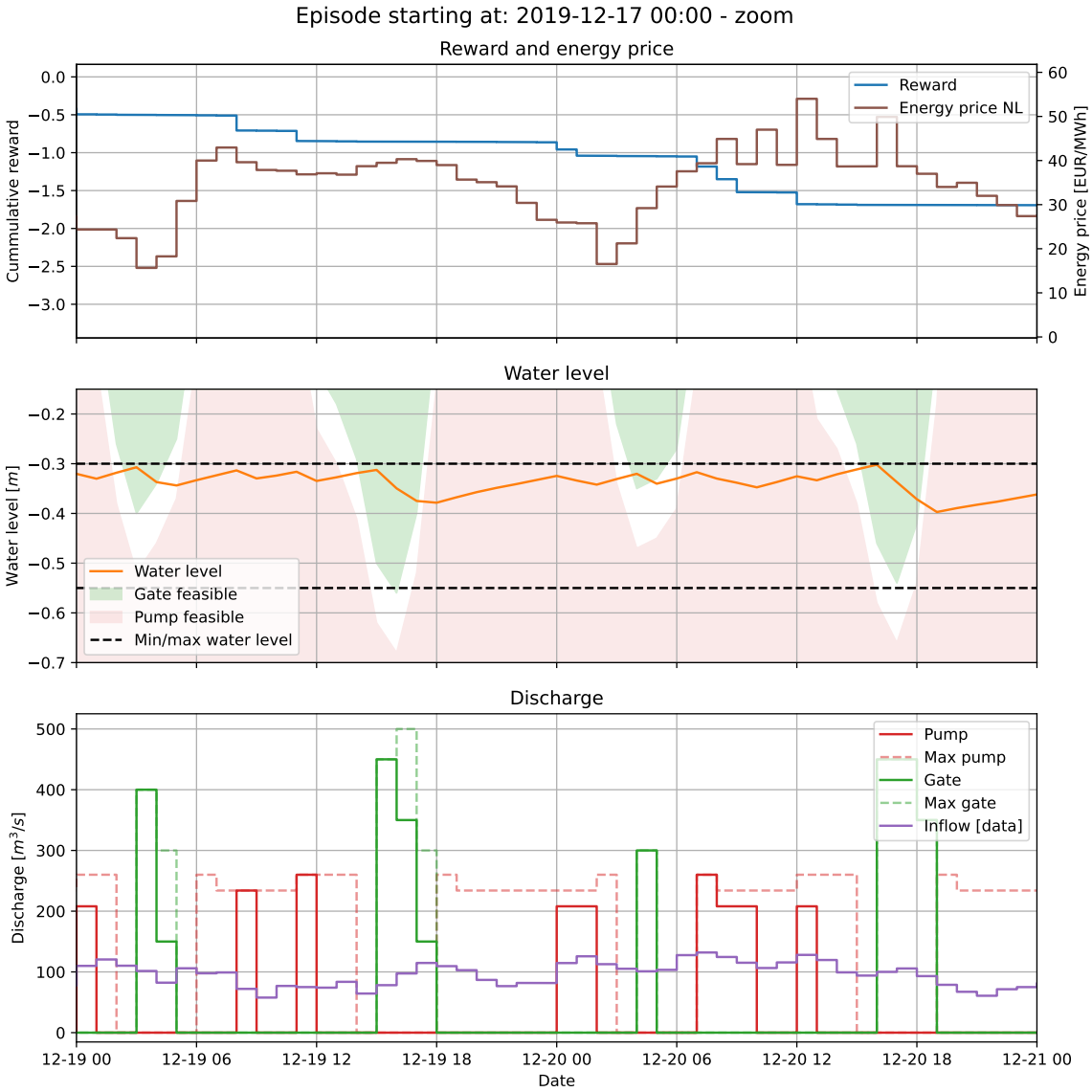


Figure 7.7: Zoom into two days (2019-12-19 - 2019-12-21) of Figure 7.6

The choice of pumping is not clearly linked to the electricity prices provided as input. There are several examples where pumping an hour earlier or later would have been more economical. The reduced costs are due to a combination of lower electricity prices and a smaller pump height, which requires less power to pump the same discharge. Nevertheless, the overall control adheres to the safety regulations for the water level. The pumping volumes are still limited as the majority of the water is drained through the gate.

Figure 7.7 also shows that the full potential of the gates is not always used. On 12-19 at 04:00 and between 16:00 - 18:00, the maximum possible discharge is at least $100 m^3/s$ higher than the action chosen by the agent. However, future possibilities for using the gates may be reduced by selecting the maximum discharge. This behaviour is also observed in the control plan created by the MPC, shown in Figure I.3. The choice to pump at higher costs suggests that further optimization is possible, which is supported by the difference in pumping strategy of the MPC.

The MPC pumps for a lower cost but as a result, the water level is extremely close to the upper

boundary for more time steps than the RL result. There are several time steps where there is a very small exceedance of less than $1cm$ in the control plan of the MPC. The MPC has 525 euros total costs over the two weeks for pumping while the RL costs almost twice as much with 995 euros.

Overall, the RL model performs well in normal situations but is not able to use the full potential of the system to optimize energy costs. The results suggest that this is due to the choice of when to use the pumps as the gates are used similarly to the control plan created by the MPC. It is difficult to determine the energy cost optimization compared to the current control of the system, as the MPC used for comparison is not the model that is currently used for operation. When considering the water levels, the small exceedances of the MPC are still likely to cause alarm. For operation, the MPC may need to be adjusted to maintain slightly lower water levels, resulting in increased costs.

7.3.3. High inflowing discharge

A more difficult control problem occurs when the inflowing discharges are greater and it is not possible to fully rely on the gate to drain all the water out of the system. This also means that there is more opportunity to minimize costs. It is a combination of draining the maximum volume with the gates as well as choosing the economical moments to use the pumps. Higher discharges around $200m^3/s$ occurred on 03-05, 2019, shown in Figure 7.8.

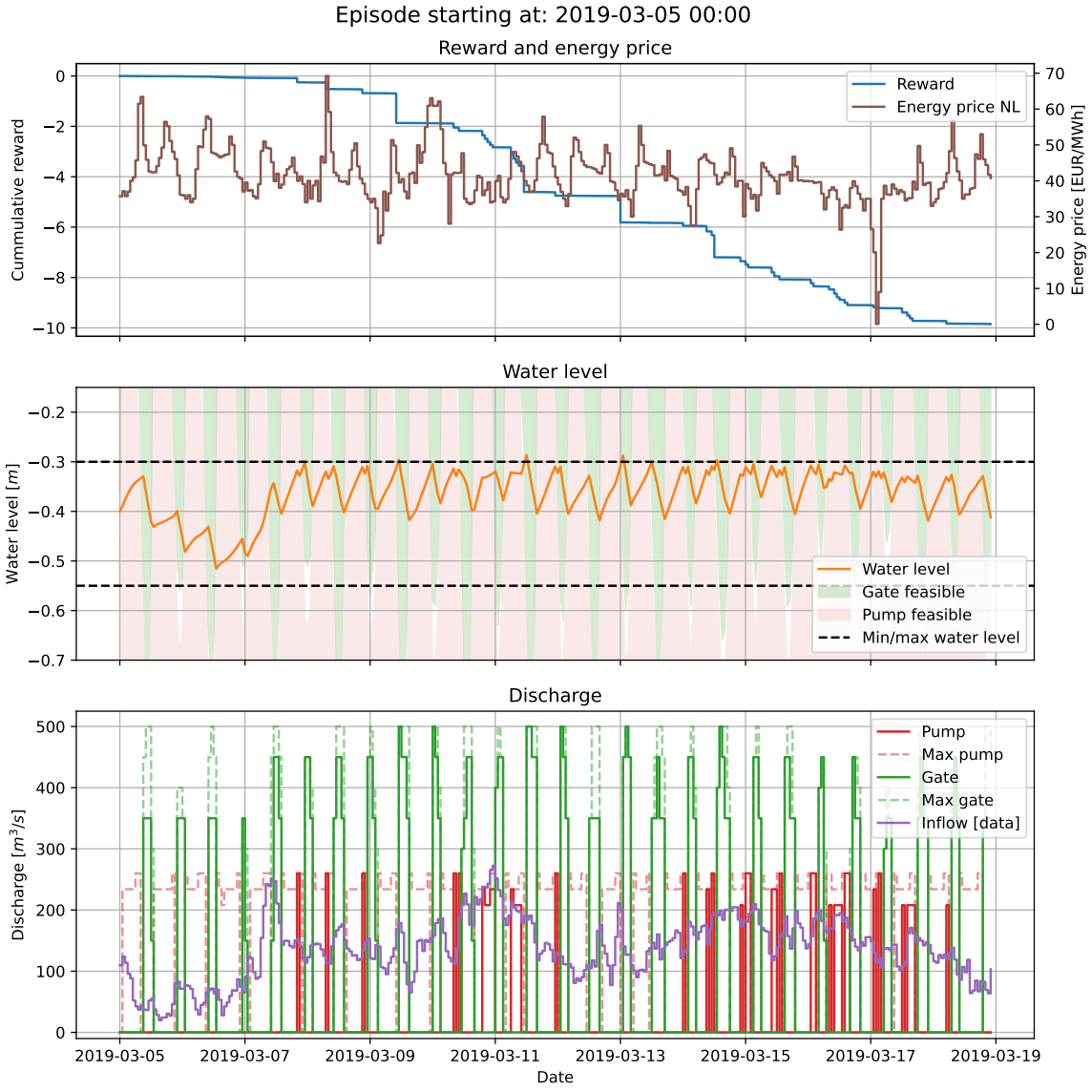


Figure 7.8: RL control plan for two weeks starting on 2019-03-05 with high inflowing discharges

The model is able to drain the majority of the water through the gates, but pumping is also necessary. The upper boundary of the target range is slightly exceeded during two time steps, however, this is by less than $1cm$. The total time that the water level is above the upper boundary is approximately 30 minutes.

Even though the exceedance is minimal, the high consequences of flooding in the area mean this is still undesirable. The advantage of such a control plan is that it can be made in advance and tweaked to reduce the chance of high water. This can be done by using the gate and pump at higher or full capacity in the hours before the expected exceedance. As the water level in the system determines which actions are possible, it is not always beneficial to increase gate/pump discharges. Higher discharges can be drained using the gate and it may be safer to maintain a higher water level that allows more use of the gate.

To assess the performance, a comparison was made with the result of the MPC, shown in Figure I.4. Table 7.1 shows the costs, water level statistics, and computation times for the two models.

Table 7.1: RL and MPC control plan performance for two weeks starting on 2019-03-05

Model	Pumping costs [EUR]	Maximum water level [m+NAP]	Out of range [% time steps]	Computation time [s]
RL	4,400	-0.286	1.2	0.9
MPC	2,700	-0.278	4.5	280.0

As with previous situations, the RL control plan has higher costs than the MPC, though in this case, the difference is smaller. Even so, there are some advantages to the RL result. In addition to the water level exceedance being slightly smaller, the portion of time steps where high water occurs is much smaller. For this water system, the most important criterion for control is ensuring safe water levels. Finally, the largest difference between the two models is in the computation times. The RL agent is able to create the control plan 300 times faster than the MPC.

7.3.4. Operational control

Several assumptions were made in the method and model of the water system. Firstly, no time was taken into account to start up the pumps and reach the desired discharge. It takes a maximum of 25 minutes for all pumps to reach maximum capacity and a single pump requires 5 minutes [26]. The control plans created by the RL agent often switch from not pumping to near maximum discharges, as in Figure 7.8. Multiple pumps would need to be activated to achieve these outflows. In this specific control plan, the pump discharges above $200m^3/s$ often follow a time step where no discharge was pumped. If the start-up time was taken into account, the actual pumped discharge would be lower, changing the resulting water level in the system.

Not including the start-up times for the pumps also means that the costs calculated are not entirely accurate. If the control plans would be adjusted to ensure that the pumps were at the desired discharge when needed, more electricity would be consumed. This electricity would also be consumed during a previous and following time steps, with a different DAM price.

The agent chooses an action that is performed for the following hour. In many cases, the water level changes during this hour, meaning the action may no longer be possible. At 12-19 17:00 in Figure 7.7 an example of this can be seen. The gate is opened for an hour even though this is not possible for around $50min$ of this time step. It is possible to reduce the time step length, however, this would allow the model to turn the pumps on and off at an unrealistic rate. Additional features would need to be included in the model to prevent this behaviour.

7.4. Comparison with Model Predictive Control

In the previous section, the MPC results were used to assess the performance of the RL model for the normal and high inflow situations. To examine the difference between the two models in more detail, both were used to create a control plan for the entire test data set from 2019-01-01 to 2021-01-01. A summary of the results can be seen in Table 7.2. To give an indication of the current control strategy, an estimate was made of the costs if the electricity was bought on the DAM.

The historical measurements were used to determine the change in water level. An estimate of the outflow discharge was made given the water level change using the linear reservoir model. The simplification of the water system combined with the DAM prices rather than those of the futures market mean that this is a very rough estimate. This does, however, give an order of magnitude. Due to the higher prices on the futures market, it is possible that the costs are higher than the calculated estimate.

Table 7.2: RL and MPC control plan performance for test data set, 2019-01-01 - 2021-01-01, with an indication of the current control derived from water level measurements.

Model	Pumping costs [EUR]	Water level range [m+NAP]	Out of range [% time steps]	Computation time
RL	116,000	[-0.535, -0.265]	0.3	37 sec
MPC	74,000	[-0.737, -0.278]	5.5	3.6 hours
Measurements	∼795,000[1]	[-0.6, -0.28]	0.09	-

The same trends can be seen for the two models as those that were identified in previous sections. The RL model creates a plan with higher costs and a lower percentage of time steps outside the target range. When compared to the historical measurements, it can be seen that the water levels exceed the target range far less frequently, but the estimated costs are significantly higher than both models. Taking into account the large uncertainty in the estimated costs, it is still expected that a significant cost reduction can be achieved by both models. To give an impression of the control plan produced by each model, Figure 7.9 shows the resulting water level for both model outcomes, with the probability density curve showing the distribution over the water level range.

Both models behave in a similar manner. General fluctuations in the water levels can clearly be seen at the same moments in time. Between 2019-10 and 2020-01, the water levels are mostly in the upper region of the target range, while in the months before and after, the levels are lower. The higher water levels are due to a combination of higher sea levels and increased inflowing discharges. Both models use the gate similarly, meaning that when the minimum water level for opening the gate is higher, a higher water level is maintained. These fluctuations cannot be seen in the historical measurements of the water level in the NZK in IJmuiden, shown in Figure J.1. This is likely caused by a different control strategy that mainly focuses on flood safety. The flexibility of the system is not used optimally to maximize the volume drained using the gate. There may also be dynamics that are not included in the simplified water model. There may be temporal and spatial effects that are not taken into account.

When comparing the exceedance of the target range, this occurs more often for the MPC control plan. At two moments, 05-2019 and 05-2020, the model does not behave as expected. The water level drops to around $-0.75m$+NAP, which is more than $15cm$ below the target range. This can easily be corrected by implementing the forced actions that are included in the RL model. This does not allow any outflow when the water level is within $2cm$ of the lower boundary. The forced action can clearly be seen in the RL result where the water level always remains at least $1cm$ above the lower water level boundary. If only the upper boundary exceedance is taken into account, the MPC model was out of range 2.1% of the time steps, instead of 5.5%. The exceedance still occurs far more frequently than with the RL model, with 0.3% of the time steps. As described above, due to the high risks, even short periods of high water can cause alarm, making this undesirable behaviour.

[1] Approximate costs if electricity was bought on the DAM based on changes in water level measurements

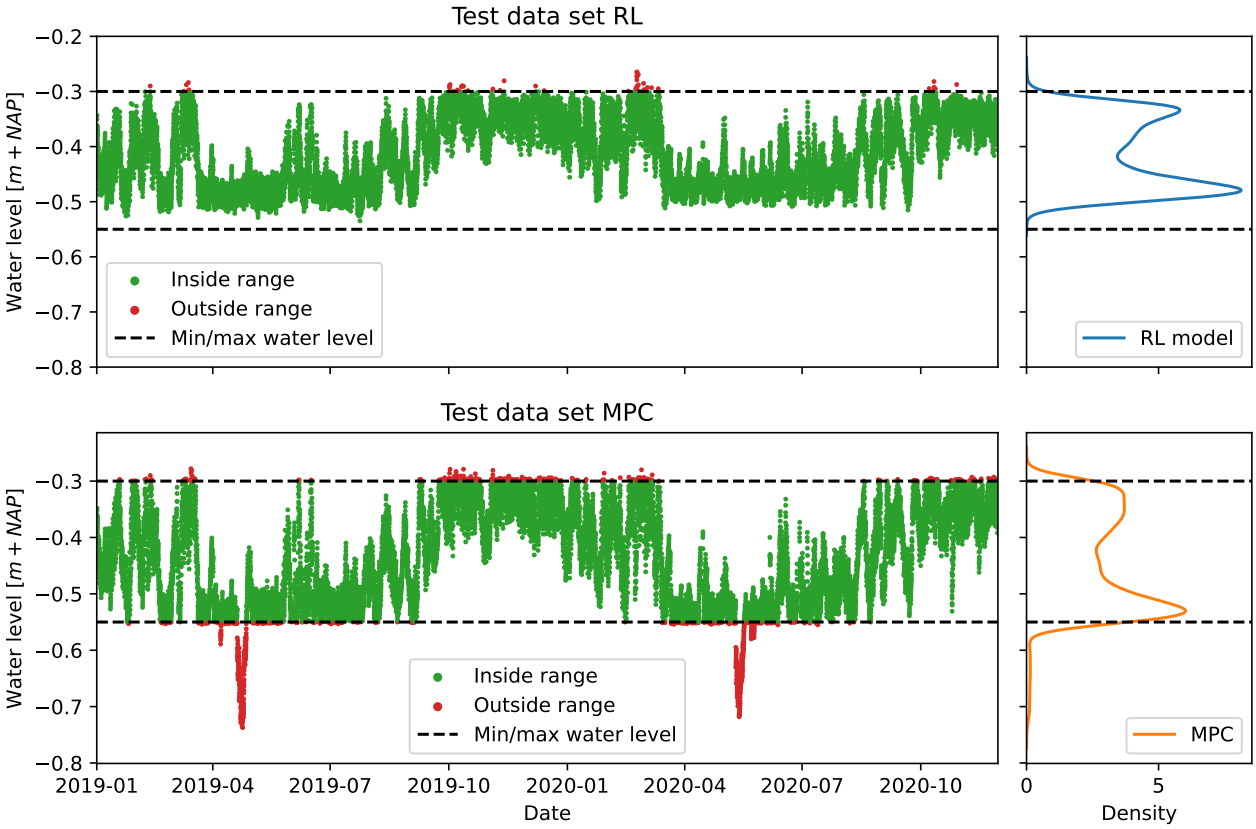


Figure 7.9: The resulting water levels for the RL and MPC control plan for test data set, 2019-01-01 - 2021-01-01. The probability density curve is shown in the right hand figures.

To analyse the exceedance of the upper boundary, Figure 7.10 shows the probability density curves for the two models, as well as the historical measurements. The measurements are rounded to the nearest 0.1m, which results in the fluctuations at regular intervals. A near normal distribution in the centre of the target water level range can be seen for the measurements. The flexibility of the system is not exploited for cost reduction. Both models show a more even distribution over the entire range. The zoom into the upper boundary exceedance reflects the more frequent exceedance of the MPC. Even though the RL model maintains safe water levels for a greater portion of the test period, the maximum water level reached is higher than that of the MPC. This can be seen clearly in Figure 7.11.

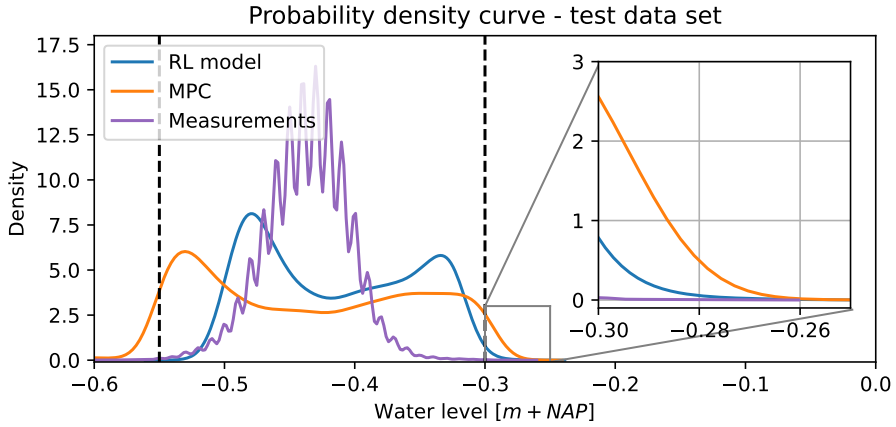


Figure 7.10: Probability density curves for the water levels during the test data set for the RL model, MPC, and historical measurements. The zoom shows the distribution when exceeding the upper target boundary.

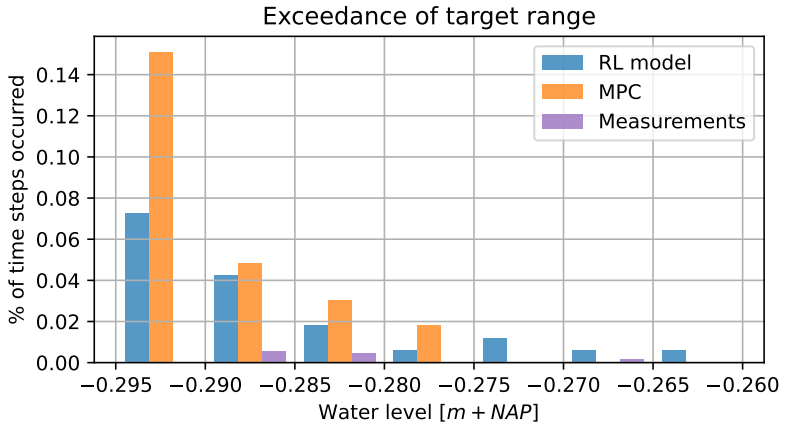


Figure 7.11: The % of time steps in the test data set where a water level occurred between bins of $0.005m$

Figure 7.11 shows how the MPC frequently exceeds the target range by at least $0.5cm$. The RL model performs better in this respect, however, the water levels reached are more extreme, which may be less desirable. When compared to the historical measurements, it can be seen that current control is focused on limiting the exceedance as much as possible. This conservative control is the main reason for the higher estimated costs. When comparing the costs, it is important to consider that the electricity is bought on the futures market. The calculation was performed to analyse the pumping costs if the control strategy was applied in the water system model combined with the DAM.

When the pumping costs are compared for the models, the MPC clearly performs best. The model is able to effectively choose cheap electricity prices for the moments to pump. Both models are extremely similar in the use of the gate. The MPC model does not drain a greater portion of the volume with the gate but selects more economical time steps to use the pumps.

Finally, the most significant difference between the two models is the computation time. While the MPC required 3.6 hours, the RL agent was able to create the control plan in under 1 minute. This allows many scenarios to be run beforehand. The control plans can be made ahead of time as well as right before control when the forecasts are most accurate. Currently, the uncertainties of the forecasts are not taken into account, as historical data is used as input. RL models typically perform well in situations that are affected by large uncertainties [8]. The use of forecasts may, therefore, highlight further differences between these two models. In addition, the fast computation time of the RL model creates opportunities for increased complexity. A more accurate representation of the water system can be included, while remaining significantly faster than the MPC. Even though the speed of the RL model is far greater, the speed of the MPC is still sufficient for the current model setup.

7.5. Extreme scenarios

The general performance was analyzed by creating a control plan for the entire test data set. To investigate the performance in critical situations, several extreme scenarios were selected. The scenarios are described in more detail in Appendix E. An overview of the performance of the RL model and MPC can be seen in Table 7.3. The solver used for the MPC was not able to solve the optimization for the Negative E scenario. All control plans created by both models can be found in Appendix K.

Table 7.3: RL and MPC performance for extreme scenarios of 1 week

Scenario	Model	Pumping costs [EUR]	Maximum water level [m+NAP]	Out of range [% time steps]	Computation time [s]
Extreme Q	RL	6,900	-0.262	5.4	0.6
	MPC	6,200	-0.280	6.5	150.0
High Q	RL	7,900	-0.294	0.6	0.5
	MPC	7,200	-0.290	14.8	150.0
Low Q	RL	0	-0.383	0.0	0.6
	MPC	-120	-0.380	5.3	130.0
High Sea	RL	620	-0.313	0.0	0.5
	MPC	230	-0.297	1.8	150.0
High Sea High Q	RL	8,000	-0.288	1.2	0.5
	MPC	6,000	-0.284	16.6	170.0
High E	RL	0	-0.365	0.0	0.4
	MPC	0	-0.369	0.0	120.0
Negative E	RL	0	-0.400	0.0	0.4
	MPC	-	-	-	-
Extreme Neg E	RL	400	-0.301	0.0	0.6
	MPC	-840	-0.373	1.8	140.0
Average [2]	RL	3,400	-	1.0	0.5
	MPC	2,700	-	6.7	140.0

Before the specific scenarios are analyzed, the general trends are compared to those found for the entire test data set in Section 7.4. Figure 7.12 visualizes the total costs of the control plans of both models for all test scenarios.

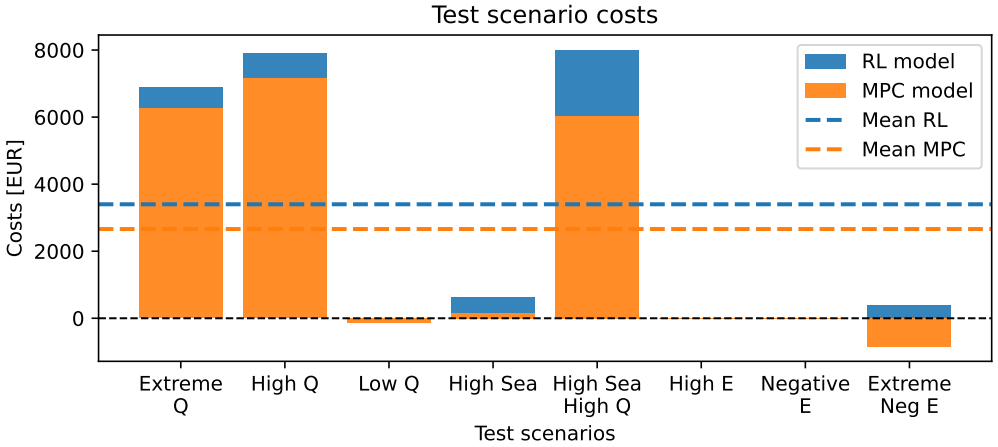


Figure 7.12: Cost of the RL and MPC control plans for all test scenarios with the mean over all scenarios

[2]Excluding results of the negative E scenario

In several scenarios, the costs of the RL model did not differ greatly from those of the MPC. However, the costs were consistently higher throughout all test scenarios except the high energy price scenario. Both models were able to avoid using the pumps altogether, resulting in no costs. The RL was not able to benefit from the negative electricity prices in the low discharge and negative price scenarios. This suggests that the behaviour regarding the choice of when to pump is not yet optimal. In all cases where negative prices occur, the MPC was able to profit from this.

The safety of both models can be estimated using the percentage of time steps that the water level is outside the target range. Figure 7.13 shows these results for all test scenarios.

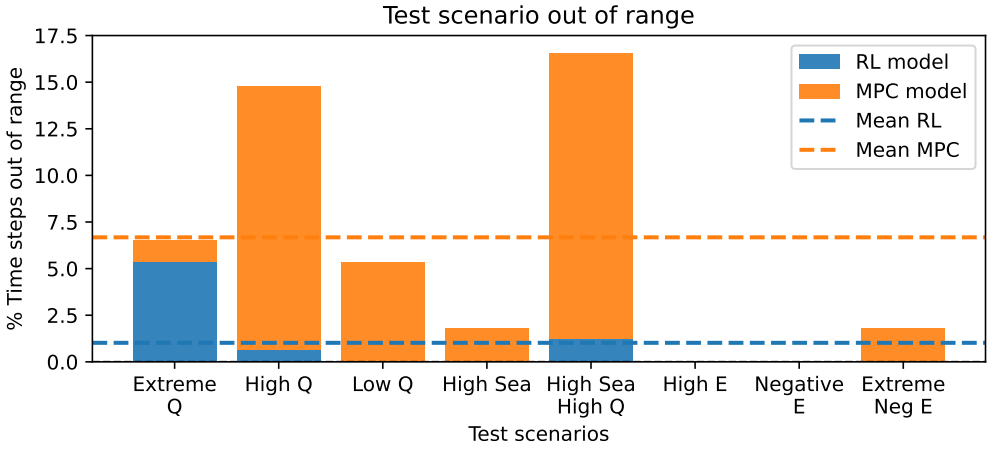


Figure 7.13: Percentage out of target water level range with the RL and MPC control plans for all test scenarios with the mean over all scenarios

The RL agent consistently outperforms the MPC, often not exceeding the target range during the entire scenario. Though the MPC does often allow the water level to rise to high water, this is often less than $1cm$. This does not immediately pose a danger for floods in the area but is still undesirable behaviour.

These two results suggest that the RL model is more conservative in choosing actions, preferring to ensure the necessary water levels rather than using the full flexibility of the system for cost optimization. This is caused by the choice of scaling of the three components of the reward function. This is further enforced by the action masking close to the boundaries of the target range. All outflowing discharge is prevented near the lower boundary. At the upper boundary, the maximum discharge is forced. The model is able to optimize by maximizing the volume drained using the gate, but performs worse selecting the appropriate time step to use the pump.

Ideally, the choice of when to pump should be based on the electricity price as well as when pumping requires the least amount of power. When the water level difference between the NZK and the North Sea is greatest, the power consumption for the same outflowing discharge will be larger. The least power is therefore often consumed closely before or following a low tide. This behaviour can be seen in general for the control plans created by the MPC.

The most significant difference between the two models is in the computation time. The RL agent is approximately 300 times as fast as the MPC. This means that far more scenarios can be run beforehand and there are possibilities for increasing the complexity of the system. A more complex system will most likely require a larger NN, which will increase the computation time. This can increase the applicability and, with further optimization of the RL method, it is expected that the performance can be increased to create more economical control plans.

7.6. Suitable reinforcement learning algorithms

A literature study was used to determine which RL algorithms were suitable for the control of a water system such as the IJmuiden pumping station. DQN was selected for the first implementation of the RL agent. The relative simplicity, the experience replay used for sampling, and the inclusion of delayed targets made the algorithm suitable. The implementation of DQN meant that it would require rewriting a large portion of the code to test algorithms with continuous action spaces. Therefore, all additional algorithms that were tested allowed the use of discrete action spaces.

There were 5 algorithms that could be tested with the same hyperparameters as those used for DQN. This gave a good impression of which algorithms have the potential for further development. The algorithms that could be implemented were; PPO, TRPO, Dueling DQN, AC, and A2C. All algorithms except Dueling DQN, which is an extension to DQN, were introduced in Section 4.2.2.

Dueling DQN uses two estimators, one for the Q-function and the other for the state-dependent advantage function. The advantage is the difference between the Q-value (expected return given the state and action) and V-value (expected return given the state), calculated as follows: $A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s)$. This helps the agent to generalize learning across actions. [75]

Table 7.4 shows the performance of these algorithms for the entire test data set, compared to the final DQN model.

Table 7.4: Performance of multiple RL algorithms on the test data set using DQN hyperparameters

Method	Pumping costs [EUR]	Water level range [m+NAP]	% Out of range[3]	Max % out of range[4]	Computation time [s]	Training time [min]
DQN	116,000	[-0.53, -0.26]	0.3	2.3	37	81
PPO	242,000	[-0.55, -0.22]	0.8	7.6	47	43
TRPO	300,000	[-0.56, -0.22]	0.8	7.4	42	46
Dueling DQN	275,000	[-0.56, -0.27]	0.04	0.7	43	82
AC	378,000	[-0.56, -0.23]	0.4	4.4	43	80
A2C	553,000	[-0.55, -0.24]	0.1	2.4	46	80

There is no significant difference in the computation time of the algorithms. PPO and TRPO clearly require less time for training the same number of epochs. However, the training times are less important for the use of the model in operation. The training is done before the models are used, at which point the evaluation time determines the applicability of the method. All tested RL methods are extremely fast, especially compared to the MPC, which is at least 300 times slower.

The largest differences between the algorithms is the performance regarding costs and water levels. DQN creates the cheapest control plan and only the water levels reached by Dueling DQN remain closer to the target range. As the hyperparameters are tuned to the DQN algorithm, this behaviour was expected. The high performance of the Dueling DQN can be partially due to the high similarity of the algorithm with DQN.

When considering the other algorithms, PPO and TRPO outperform both actor-critic methods in terms of costs. The lower costs have resulted in greater exceedances of the target range. As ensuring the safety of the system is most important, this is not desirable behaviour.

7.6.1. High inflow scenario

To analyze the performance of all algorithms in a complex optimization problem, control plans were made for the high inflow event of 2020-02-18. The behaviour of the algorithms resulted in unique patterns for the water level development. Figure 7.14 shows the water level for a portion of the high water event, where the start of the peak discharge is shown. The performance regarding total costs, water level, and computation times can be seen in Table 7.5. The performance of the DQN algorithm and the MPC are shown as a reference. Appendix L contains the complete control plans for all models.

[3]% of time steps out or range over the 2 year test data set
[4]Maximum % of time steps out of range in 1 month

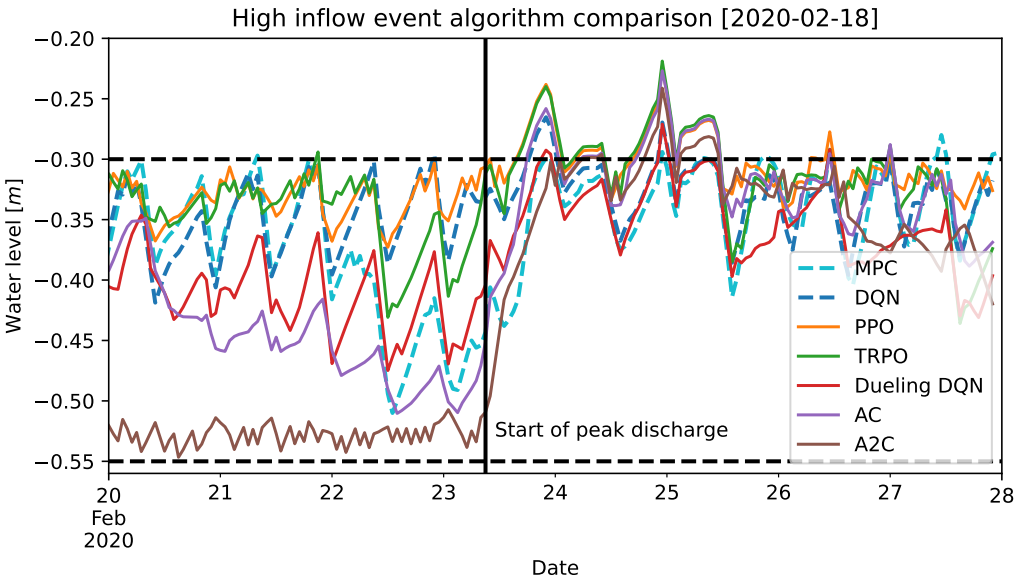


Figure 7.14: Comparison of water levels for control plans of multiple RL algorithms for the high water scenario of 2020-02-18

Table 7.5: Performance of multiple RL algorithms for the high water scenario of 2020-02-18 (10 days)

Method	Pumping costs [EUR]	Water level range [m+NAP]	% Out of range	Computation time [s]
DQN	7,200	[-0.42, -0.27]	4.2	1.8
MPC	6,200	[-0.51, -0.28]	7.1	248.0
PPO	9,200	[-0.40, -0.22]	16.3	1.4
TRPO	8,600	[-0.44, -0.22]	17.2	1.3
Dueling DQN	9,400	[-0.47, -0.27]	2.5	1.6
AC	9,300	[-0.51, -0.23]	13.0	1.3
A2C	14,000	[-0.55, -0.24]	8.4	1.3

There are apparent differences between the models in how the water level changes before the peak discharge. Several algorithms (A2C, AC, and Dueling DQN) anticipate this by (slightly) lowering the water level. A2C maintains a very low water level unless this is not possible due to high inflows, which results in far higher costs than the other methods. The lower boundary of the target range is also often exceeded.

Even with the lowered water level, AC exceeds the upper boundary by a significant amount. The lowered water level only results in higher performance with Dueling DQN, where the percentage out of range is lower than both the DQN and MPC. The overall costs are higher because a smaller volume is drained using the gate.

Finally, PPO and TRPO have very similar performances, most likely due to the similarities in the two algorithms. The costs are similar to AC and Dueling DQN but the water levels rises to 8cm above the upper boundary.

Overall, the Dueling DQN performs extremely well and, with further optimization, is likely a suitable extension to the current implementation of DQN. Due to the use of the DQN hyperparameters, other algorithms may have significantly higher performance if the hyperparameters are tuned appropriately. The anticipation by lowering the water level suggests that an actor-critic method may prove suitable for this application.

7.7. Alternative reward structure

The current reward structure includes a penalty when the water level is within the target range. This can cause suboptimal behaviour, as the full water level flexibility is not exploited to minimize pumping costs. The in range penalty was included to stabilize and speed up the training process. However, several additional tests were done to analyse the performance increase than may be achieved by simplifying the reward to two components. The new RL model that was tested was kept identical to the original model, except for the in range penalty and the training procedure. In order to achieve the necessary performance, the agent was trained for an extra 2000 epochs of 15 days. The in range penalty ensured that there was always an optimal state, even when in range and not pumping. Without it, the model first needed to learn an estimate of future rewards to effectively train within the target range. Initially learning the return estimate meant that the overall training time increased.

Table 7.6 shows the tests that deviated most from the performance of the original model. During normal conditions and the extreme scenarios (except the extreme negative energy prices scenario), the performance was almost identical to that of the original model. The computation times of the two models were very similar, as the NN used to determine the action used the same architecture and hyperparameters. The control plans created by the alternative model for the results in Table 7.6 can be found in Appendix M.

Table 7.6: Performance of alternative reward structure

Scenario	Model	Pumping costs [EUR]	Maximum water level [m+NAP]	Out of range [% time steps]
Entire test data set	$\text{RL}_{original}$	116,000	-0.265	0.3
	$\text{RL}_{alternative}$	102,000	-0.260	0.2
High water level 2019-12-17	$\text{RL}_{original}$	2,400	-0.30	0.0
	$\text{RL}_{alternative}$	1,700	-0.30	0.0
High inflow 2019-03-05	$\text{RL}_{original}$	4,400	-0.29	1.2
	$\text{RL}_{alternative}$	3,800	-0.30	0.0
Extreme Neg E scenario	$\text{RL}_{original}$	400	-0.30	0.0
	$\text{RL}_{alternative}$	90	-0.30	0.0

The source of the improved performance of the alternative reward is highlighted in the resulting water levels for the entire test data set, shown in Figure 7.15. The probability density curves for the water levels for the original RL model, MPC, historical measurements, and alternative reward structure can be found in Figure 7.16.

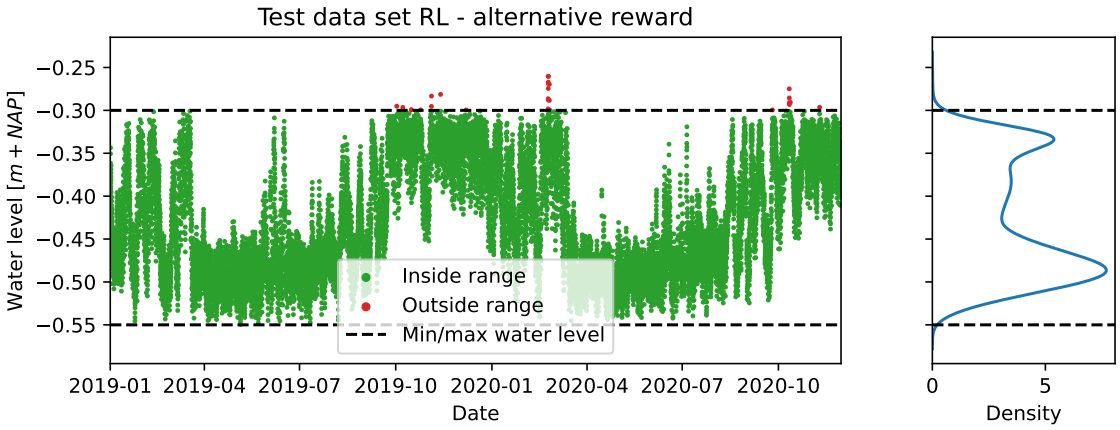


Figure 7.15: The resulting water levels for the RL with the alternative reward structure for the test data set, 2019-01-01 - 2021-01-01. The probability density curve is shown in the right hand figure.

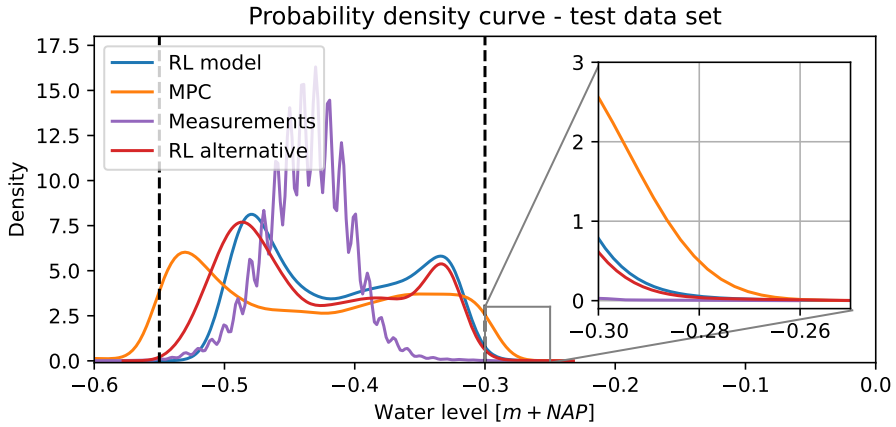


Figure 7.16: Figure 7.10 including results from the alternative reward structure showing the probability density curves for the water levels during the test data set. The zoom shows the distribution when exceeding the upper target boundary.

The alternative reward structure did not restrict the water level when in range. As a result, the model utilized the full flexibility of the system to a greater extent by approaching closer to the lower boundary. It can also be seen that the water level fluctuates more frequently than for the original reward structure. The proximity to the target range boundaries did not result in a higher exceedance rate. The alternative reward performed better in this respect.

The use of the full flexibility allowed the model to create control plans the generally meant lower pumping costs. Slightly anticipating high inflows also had a small contribution to this.

The suboptimal behaviour with respect to the negative electricity prices was still observed with the alternative reward. The model did not pump specifically during negative prices, but mainly based the pumping strategy on the water level in the system. Changes in the training procedure or reward structure are therefore still necessary to solve this. This test does show how changes in the reward structure have a significant influence on the final behaviour of the model. This is the only information the model receives about the desired actions.

8

Conclusion and Discussion

The objective of this thesis was to optimize the control of the IJmuiden pumping station using RL while complying with local water level restrictions and compare it to the state-of-the-art MPC methods in terms of constraint violation, energy costs, and computational speed. This IJmuiden case study allowed the potential for an RL controller to be explored in a real-world context. The NZK-ARK system plays an important role in the flood safety of the major metropolitan areas of Amsterdam and Utrecht, including important infrastructure of high economic value. It is critical that the water level remains within the safe range. Within this range, the flexibility of the system provides opportunities for energy cost optimization.

8.1. Suitable reinforcement learning algorithms

In order to select a suitable RL algorithm for the agent, a literature study was performed. The DQN algorithm was selected due to the relative simplicity, the use of experience replay for sampling, and the delayed targets. In addition, the algorithm can achieve high levels of performance through various extensions in domains with discrete actions [57].

The base implementation of the algorithm was implemented and evaluated. After hyperparameter optimization on the validation set, the model reached a high level of performance on the test set. The water level was nearly always maintained within the target range, performing at least as well as the state-of-the-art MPC. The consistent use of the gate when possible reduced the electricity costs of pumping. However, the pumping strategy was still suboptimal, which resulted in costs at least 50% higher than the MPC. The chosen time step for pumping was based mainly on the water level in the system and the electricity price forecast was not adequately taken into account.

After the base implementation, several extensions were researched and implemented. One extension that was tested was Dueling DQN, which uses two estimators for the Q-function and state-dependent advantage function. Without further optimization, the algorithm outperformed the DQN agent in terms of maintaining safe water levels. This did result in higher costs, but with further tuning of the training procedure, hyperparameters, and reward structure, it is expected that the costs can be decreased further. This can include training more examples with high sea level and low inflowing discharge. This scenario will make pumping necessary without the difficulty of staying within the target water level range. The model can then learn to reduce the penalty for pumping as well as taking advantage of negative electricity prices.

Actor Critic methods showed desirable behaviour by lowering water levels in anticipation of higher inflowing discharges. These methods differed more from the original DQN, which suggests that the hyperparameters may be less suitable and greater improvements can be achieved with tuning. Currently, the Actor Critic control plans have significantly higher costs than the DQN model.

It was not possible to test the performance of methods with continuous action spaces within the scope of this research. Such an algorithm would keep the ordinality in the actions, which may increase model performance. The masking of actions can result in the agent not being able to choose the action with the highest expected return. Currently, the agent is not able to choose the next best action in terms of outflow discharge, but the action with the next highest expected return is selected, which

may not be the same. DDPG, TRPO, PPO, A3C, and ACER are algorithms that all allow the use of continuous action spaces.

Overall, the final performance of the DQN model shows that this algorithm is suitable for this type of system. Further optimization and the addition of extensions to the algorithm have the potential to create a model that can control the system with similar operational costs as the current state-of-the-art MPC.

8.2. Water level constraints

The high economic value of the areas surrounding the water system results in strict water level regulations to ensure the safety around the NZK-ARK. The water level should always be maintained within the target range between $-0.55m{+}NAP$ and $-0.30m{+}NAP$. Before cost reduction was included in the model objective, the agent was first developed to deal with these classical constraints.

As this was a relatively simple optimization problem, the model performed very well. During normal conditions, the water level was never exceeded. Only during extremely high inflowing discharges did the water level rise above the upper boundary. During the most extreme inflow conditions, the water level exceeded the target range during 2% of the time steps over an entire month. The water levels rose to $-0.28m{+}NAP$, $2cm$ above the target range. This is a high water situation, but no measures are taken until a water level of $-0.20m{+}NAP$ is reached. At that point, the first problems will arise for shipping as the minimum vertical clearance will not be met. The IJ-front will be closed off from the NZK-ARK system due to the increased risk of flooding in Amsterdam.

The model did not adequately anticipate for the high inflowing discharge, which meant that choosing the maximum outflow action was not sufficient to maintain the safe water levels. The desired behaviour includes lowering the water level when high inflows are present in the forecasts.

This model allowed the agent to be tested for a simpler optimization problem to ensure that the model was able to learn and that the dynamics of the water system model were correctly implemented. The reward was independent of whether the pump or gate was chosen. This meant that the agent learned to only use the pump, unless a large inflow discharge meant it was not possible to keep the water level within the target range without using the gate.

8.3. Cost optimization

After successfully implementing the agent with only the water level objective, cost optimization was added to the problem. For this water system, the most important criterion for control was respecting the water level restrictions. Within this flexibility of the system, the costs could be minimized.

8.3.1. Water level constraints

Over all the 1000 test cases, the agent was able to ensure that the water level did not drop below the target range during all time steps. A small margin of approximately $1cm$ was maintained from the boundary. This was partly due to all outflowing discharges being prevented when the water level reached $-0.53m{+}NAP$. However, if the water level was slightly above this level, the agent could still choose a large discharge. This could decrease the water level to below the lower boundary. The performance of the agent was very high in this region of the state space. Figure 7.9 shows the water level for the entire test data set, where the margin to the lower boundary can clearly be seen.

A more complex objective was ensuring that the water level did not exceed the upper boundary of the target range. In this respect, the RL agent outperformed the MPC, by allowing high water to occur less frequently over the test period. The largest exceedance reached a water level of $-0.27m{+}NAP$, only $3cm$ above the upper boundary. The total duration of the high water was approximately 4 hours over the episode of 1 month. The first measures are taken in the water system at a water level of $-0.20m{+}NAP$, which was not close during the entire high water event. During this event, the MPC was able to limit the level to $-0.28{+}NAP$, slightly better than the RL. However, the total time that there was high water for the MPC was longer due to multiple small exceedances. This shows how the two models have similar performance in this regard during high inflow events. In terms of total high water time, the RL model performed best. The high water of both models can be seen in Figure 7.3.

The most significant difference in behaviour for this specific event was the choice of actions before the peak discharge. The MPC lowered the water level to create a larger buffer when the peak occurred. This was not observed for the RL model, resulting in a larger water level exceedance. This suggests

that this scenario did not occur frequently enough in the training episodes, as the model did not behave optimally.

The RL agent performed as desired during normal conditions. All water was drained using the gate, which meant that no costs were made while maintaining the necessary water levels during all time steps. For higher inflow, the agent was also able to ensure safe water levels in the system. This did require the use of pumps, as the gate was not sufficient. The behaviour of the agent was not yet optimal as it did not anticipate the peak discharge by preliminarily lowering the water level.

8.3.2. Cost reduction

As mentioned above, during normal conditions in the water system, the RL agent was able to maintain safe water levels without pumping. As the inflow discharge increased, the use of the pumps was required. The agent optimized the use of the gate to reduce costs. This minimized the volume that was drained using the pumps.

When the control plans were examined in more detail, it could be seen that the choice of when to pump was mainly driven by the water level. When the water level approached the upper boundary of the target range, the pumps were used. There was no clear correlation with the electricity prices or power consumption. This was further highlighted by the differences in the control plans created by the MPC. The MPC used the temporarily lower DAM prices to pump as much as possible in combination with moments when the pump height was small, reducing power consumption. Only when it was not possible to delay the pumping to a lower cost time step was the pump used during less than ideal hours.

As a result, the RL model consistently created control plans with higher total costs than those of the MPC. For the entire test data set, the total costs of the RL control plan were approximately 50% higher. Though this difference was significant, the economical use of the gate is expected to reduce overall costs when compared to the current operation. The estimate of current operational costs showed that both models would significantly reduce costs by at least a factor of 6.

Current control includes spreading out the gate discharge over the longest possible period, which reduces the possibility of cost optimization, as this generally lowers the total discharge that can be drained by the gates. As the RL and MPC models do not take this into account, the system is more flexible, and lower costs can be achieved.

Even though the RL model has approximately 50% higher costs, the water level exceeds the upper boundary far less frequently. The upper boundary was exceeded 0.3% of all time steps, while for the MPC this was 2%, as shown in Figure 7.9. As the flood safety is of great importance, high water conditions greatly impact the area. It may therefore be preferable to have higher costs if that reduces the chance of high water occurring. If the MPC were to reduce the time steps where the water level was out of range, this would increase the costs.

The last important difference between the two models is the computation time. The RL control plans can be created more than 300 times faster than those of the MPC. This is a significant advantage as this shows the potential for increasing the complexity of the water system model as well as details regarding the control of the water system. This can include changing the linear reservoir model to using the more accurate one-dimensional Saint-Venant equations. More accurate calculation can be made for the flow and water levels throughout the system. The six pumps have also been simplified into one pump with characteristics that approximate the behaviour of all pumps together. The closer the water system is modelled to reality, the more applicable the model becomes for operational control.

8.4. Extreme scenarios

After examining the performance of the RL model for the test data set, control plans were created for several extreme scenarios and compared to the state-of-the-art MPC model. Similar performance was observed in previous situations. The RL costs were consistently higher than those of the MPC. On average, the costs were 25% higher, which was mainly due to the MPC creating income by pumping during negative electricity prices. The costs of both models were similar for scenarios where pumping was frequently necessary, while larger differences were observed with negative prices or only sporadic pumping. The costs of all scenarios can be seen in Figure 7.12.

The water levels reached were similar for both models, however, the RL maintained safe water levels for a significantly larger portion of the time steps. This included both exceeding the upper and lower boundary of the target range. Several situations occurred where the MPC pumped during negative

prices, thereby lowering the water level below $-0.55m{+}NAP$. The maximum water level was often exceeded by the MPC. The model allowed higher water levels, resulting in exceedance as soon as the inflow was too large. The water level was often less than $1cm$ above the upper boundary.

When compared to the RL model, the water level was normally maintained slightly lower, giving the model a larger buffer during higher inflows. This difference can be seen clearly in the high sea levels and high inflow scenario in Appendix K.5. The MPC has 20% lower costs, however, the maximum water level is exceeded during 16% of the time steps.

As with the test on the full test data set, the computation times of the two models were of different orders of magnitude. The RL model was able to create a control plan in around 0.5 seconds while the MPC required at least 2 minutes.

8.5. Overall reinforcement learning model

After analysing the performance of the RL model on the entire test set, for extreme scenarios, and compared to the state-of-the-art MPC, there were several additional observations.

8.5.1. Operational control

The current setup of the model it is not yet suitable for operational control, regardless of the performance. The start-up time for the pumps to reach the desired discharge is not considered. The RL agent often chooses to switch from not pumping to near-maximum discharges, where multiple pumps would need to be activated. As a result, the outflowing discharge computed will be larger than that in reality. The plans can be made ahead of time, which means that the pumps can be turned on ahead of time. This allows the pumps to be at the desired discharge at the right moment. However, the costs no longer be accurate, as pumping is also performed outside the original time step. In addition, the frequent changing of pump discharge will increase the wear on the pumps, resulting in higher maintenance costs.

Only the water levels at the start of the time step are used to determine which actions are possible. Due to the time step length of one hour, it occasionally occurs that an action is no longer possible during the time step. The model currently does not take this into account. In the case of opening the gate, if the sea level rises rapidly, this action may no longer be possible. One hour is enough for significant changes can occur that make many actions infeasible.

8.5.2. Water system model

Currently, the water system is modelled as a single linear reservoir. This means that the water level throughout the whole water system is the same. The inflowing discharges instantaneously affect the water level in the system, as do the outflowing discharges. The location of the inflowing discharge is also not taken into account. Due to the length of the system and the differences in wind set-up, the water levels in Amsterdam and other critical areas can deviate from that in IJmuiden.

The speed of the current model means that it is feasible to implement a more complex model of the water system that accounts for the spatial and temporal effects. This is a large benefit compared to MPC methods. The computation time of MPC greatly limits the complexity of the system representation. In addition, RL methods are suitable for the approximation of non-linear dynamics.

8.5.3. RL training

During training, there are fluctuations in the validation performance that cannot be attributed to changes in the reward structure or episode length. This suggests that the training is not entirely stable and further optimization can be done. The instabilities were observed for all training instances of the RL agent. The tuning can improve performance in addition to making training more stable. This can include changes to the NN architecture, reward structure, and hyperparameters.

8.5.4. Alternative reward structure

Initial tests of removing the penalty for the water level within the target range showed improvements in the model performance. The model was able to use the full flexibility of the water system, which resulted in a small reduction in costs. The model was not limited by the penalty for deviating from the target water level of $-0.40m{+}NAP$. In addition, the model was able to reduce the time steps where the water level exceeded the target range.

8.6. Final conclusion

Overall, RL methods are capable of controlling a water system, such as the IJmuiden pumping station. The model adheres to the safety regulations regarding the permitted water levels with the exception of small exceedances during extreme inflow events. During these events, the water levels reached are similar to those of the state-of-the-art MPC. However, the total time of exceedance is significantly less. The additional flood safety regarding and a less optimal pumping strategy result in significantly higher costs for the RL.

The testing of 1000 test episodes, as well as multiple extreme scenarios, allowed the consistency of the model to be evaluated. The RL model proved to be robust, with outputs being consistent in complying with the water level constraints. The performance in terms of cost optimization was also consistent throughout the testing.

The biggest advantage of RL over methods such as MPC is the reduction in computation time. Currently, the RL model is able to create a control plan for the pumping station approximately 300 times faster than the MPC. This opens doors for further development of the model and increased complexity. A more accurate representation of the water system can be used. This can include more temporal and spatial effects as well as a more realistic action space. The action space can discretize all six pumps with their individual power and pump height curves. These adjustments will make the model more applicable for the operational control of such a system.

This research has shown that further development of RL methods in this application is still necessary to improve the performance with respect to minimizing costs. This method provides an extreme reduction in computation time, making the control of more complex systems with multiple objectives feasible. These control methods can contribute to the balancing of the electricity network as the portion of renewable energy sources increases while optimizing for the energy consumption of the system.

9 Recommendations

Unlike Proportional Integral Derivative (PID) and MPC, which are mature control systems that are widely used today, RL methods are not yet widespread. RL is a very active field of research and is not currently used for the operational control of a water system, such as the IJmuiden pumping station. This research explored the potential of RL for this application and, naturally, many recommendations can be made. These recommendations will touch upon all facets of the control system. This includes the water system model, the RL method, and finally the potential for a combination with MPC.

9.1. Water system model

9.1.1. NZK-ARK

The assumptions made by modelling the water system as a linear reservoir decrease the accuracy of the calculated changes in the water system. The fast computation speed of the RL model enables the use of a more complex model, such as using the one-dimensional Saint-Venant equations. These are often used to model open channel flow, such as the NZK-ARK system. The equations describe the relationship between water level, discharge, and storage. Currently, all inflowing discharges simultaneously enter the system and have an immediate effect on the water level throughout the entire system. The water level in IJmuiden is also considered equal to that in Amsterdam, which is not always the case due to effects such as wind set-up. Using a more realistic representation of the system will increase the real-world applicability of the RL method. The environment used for its training will more closely resemble the real world.

9.1.2. Pumping station model

Not only the model of the NZK-ARK can be more realistic but also that of the pumping station. The pumps can be represented individually with their individual discharge, pump height, and power relationships. The action space can be discretized to include all pumps instead of a single pumping action. If maintenance costs can be included in the model, the actions can also be selected based on which pump is already operational. The agent can learn that frequently turning a pump on and off increases the wear. In addition, when pumps require maintenance, the control method can still produce a realistic plan by completely masking the inactive pump.

If all states of the system can be expressed in costs, the agent can find the optimal behaviour to reduce total costs. These costs include the specific costs of the use of the pumps and gate as well as the monetary consequence of certain water levels. Currently, the behaviour of the agent is determined by the chosen scaling of the reward function. If the costs of all actions can be established, this will allow a more accurate optimization.

9.1.3. Time resolution

Currently, the actions are chosen hourly. Since data is available every 5 minutes, the state of the water system can be more accurately captured when changes are determined at this frequency. The current set-up of the model uses the average inflowing discharge for the time step to determine the water level changes. However, as the discharge is not constant, the water level fluctuations during the time step are

not taken into account. A linear change in the water level is assumed during the time step. Increasing the time resolution will give a more accurate maximum and minimum water level during specific control episodes. If the inflowing discharge was very high at the start of the time step interval, the maximum water level may have been higher than was calculated with the current method.

Another consequence of the time step length is that the state of the water system can change significantly during a time step. We currently assume that an action can be performed during the entire time step. This is not always the case and therefore the feasibility of the action needs to be considered. To ensure that an action is only performed when the restrictions of the water system allow it, the feasibility of the action needs to be computed more frequently. The model can either be set up to stop all outflowing discharge, continue with the next highest possible action, or be allowed to choose a new action. Setting the outflow to zero until the next time step will probably result in the water level exceeding the target range more frequently. This is the case during high inflows combined with a water level close to the upper boundary. Continuing with the next highest possible action may not be the optimal action for that time step. It is therefore expected that the highest performance can be achieved by allowing the agent to select a new action when the previous action is no longer possible.

9.1.4. Operational control

Before such a model can be used for operational control, the model needs to be tuned to the behaviour of the operators that make the final decision about which control strategy is used. This will ensure that the model is used optimally. Real-world performance may be increased if the upper boundary of the target water level range is set to $-0.32m{+}NAP$ rather than $-0.30m{+}NAP$. Operators are likely to become nervous when the water levels approach $-0.30m{+}NAP$. As a result, they are likely to deviate from the optimal control plan suggested by the model. If this is taken into account by the model, the most optimal plan for those safer constraints can be found which will be followed by the operators of the system.

The current policy for the pumping station includes spreading the gate discharge over a longer period to reduce the maximum discharge. These constraints will reduce the potential of the system to minimize costs by limiting the flexibility of the system. If the maximum discharges are used, this may negatively affect the bed around the pumping station. The impact of this change in control would need to be investigated.

The more frequent changing of the discharge of the pumps is expected to increase maintenance costs. To fully optimize the system, these additional costs need to be quantified. This will allow the agent to make the trade-off between lower electricity costs at the expense of higher maintenance costs in the future.

9.2. Reinforcement learning method

9.2.1. Algorithms

During testing, several other RL algorithms have shown potential for proper control of the IJmuiden water system. In particular, the extension to DQN, Dueling DQN, showed very high performance without any further optimization. As discussed in [57], there are more extensions to the DQN algorithm which can significantly improve performance in the appropriate application. Double Q-learning was discussed previously, which addresses the overestimation bias caused by the maximization in calculating the target return for updating weights [76]. Prioritized Experience Replay will allow us to emphasize important state transitions. This could improve the performance of the model in negative price situations as well as choosing the most cost-efficient time step for pumping. The last popular extension to DQN is Distributional RL where the distribution of the returns rather than the expected returns are learnt [77]. Testing with various combinations of these extensions may yield an algorithm with higher performance than the current DQN, which may even approach the costs reduction achieved with MPC.

There are also other algorithms that showed potential and can be tested further. The algorithms created control plans with less optimal behaviour, however, this is likely due to the use of the DQN hyperparameters, which may not have been suitable. Anticipating high inflows is behaviour that is desired in the controller, which was seen in the Actor Critic methods. These methods can be tested more extensively, with appropriately tuned hyperparameters to give a more appropriate estimate of their performance. The control plans currently have significantly higher costs than the optimized DQN model.

Finally, it is recommended to test the performance of the agent with algorithms that allow a continuous action space. The ordinality in the actions suggests that this might make the optimization problem less complex to learn. The action space can be split into three possibilities, one representing no outflow, the second representing the discharge pumped, and the third the discharge drained using the gate. The value of the NN output is then used to determine the magnitude of the discharge.

9.2.2. State space

Input data

The splits in the data set were made without padding between the training, validation, and test set. There is a temporal relationship between the discharges, sea levels, and electricity prices. By splitting the data on a specific date and time, consecutive hours were used for different training purposes. As a result, there was a small leak of information in the training set about the validation and test set. In future research, an interval of at least 1 month should be kept between the data splits. This will ensure that the validation and test set give an accurate impression of the performance of the model on a new and unseen data set.

Forecasts

Currently, the inputs of the model are historical data rather than forecasts. Due to the computation speed of the RL model, it is expected that accurate forecasts will be available at the moment when the control plan needs to be created [26]. The model currently only needs forecasts 48 hours ahead. Even so, there will be uncertainty in the data, especially during more extreme events. Future models should be further optimized to deal with these uncertainties. There are not only uncertainties associated with forecasts but also those caused by measurement errors.

Predictions of the energy prices will also be used during operational control of the system rather than the definitive DAM prices. The definitive prices are only available after the market has closed and no more electricity can be bought or sold. This means that there is also uncertainty associated with the electricity prices provided as input to the model.

As RL algorithms are suitable for applications affected by large uncertainties [8], it is expected that the model will maintain a high performance when the forecasts are introduced rather than the historical data. If the model is to be used for real-world applications, the use of forecasts is essential as the agent will need to learn to deal with the uncertainties to perform well in operation.

The computation speed of the RL algorithm allows multiple scenarios to be run when the severity of an extreme event is uncertain. If there is a possibility of high inflowing discharges, a control plan can be created for multiple scenarios to determine the system response. This can help to take into account the probability of occurrence of an event.

Electricity markets

As discussed in Chapter 3, the IDM is also suitable for cost optimization in the IJmuiden pumping station. This may also reveal additional benefits to the RL method as speed becomes an increasingly important factor. The bids on the DAM need to be submitted the day before delivery, while on the IDM this can be as short as 5 minutes before. Due to the changing prices and state of the water system, a fast model can benefit effectively from the price fluctuations. The model can be run repeatedly as soon as new price estimates and other important inputs are available. If the IJmuiden pumping station is controlled using electricity bought on both the DAM and IDM, this will likely require two RL models. Each model works with different inputs and with different time steps. The performance and interaction of such models would need to be tested and validated to assess the potential of the combination of these markets.

An adjustment that might simplify the current optimization for the RL agent and increase performance is the use of relative electricity prices. The absolute price is not the most important for the model, but rather the cheapest time steps in the coming forecast period. The agent always chooses the gate if possible and otherwise should select the best price option for pumping. Negative prices could still be represented explicitly, as this is a necessary distinction for cost reduction. The additional benefit of relative pricing is that the agent will not need to be re-trained when the prices change.

9.2.3. Training steps

Due to the speed of the training, more evaluation steps can be performed during training to give a more accurate indication of the performance. When training the agent locally, training times were significant, which slowed the development speed. However, when training on Snellius, this was no longer a limiting factor. Currently, the performance is evaluated by testing 20 episodes every 50 training steps. Increasing the number of episodes tested may help to further optimize the training procedure.

In addition, the training examples can be tuned to increase the model performance for specific scenarios. This can include the pumping strategy and use of the negative electricity prices as well as taking into account the forecast of inflow. To train the pumping strategy, the model can learn from episodes where the sea level is high and the inflowing discharge low. This will force the model to use the pumps, as the use of the gates is not possible. The low inflow will mean that the focus is not on maintaining safe water levels. Similar training scenarios can be created for other regions of the state space where the model performance needs to improve.

9.3. Combination with MPC

The computation time of the MPC optimization can be reduced by initializing the model close to the optimal solution. Using artificial intelligence techniques as a warm-starting procedure for MPC has been demonstrated to reduce computation effort for MPC [78]. As the RL model is not able to decrease the energy costs as effectively as the MPC, it is possible that the computation speed of the MPC can be increased by using the control plan of the RL agent to initialize the model. This may also make it more feasible to use a more complex model for the water system. Operators may also prefer the reliability of a MPC. The control plans of both models are very similar in the use of the gate and the pumping strategies often also overlap.

Trust in a new method such as RL can perhaps be achieved by running the model parallel to a trusted method. This can show the reliability and performance of the method before it is used in operation. The model can also be used for control when there is no risk of flooding. When the water levels rise too close to the upper boundary, control can be switched to the current reliable method.

It is possible that a combination of two models yields the safest and most cost-effective control. An MPC can be used to create the control plan based on the DAM prices, after which an RL model computes changes to the plan with energy bought on the IDM. The speed of the RL makes planning based on the IDM more feasible as bids can be submitted as late as 5 minutes before delivery.

References

[1] Debot at Dutch Wikipedia. *IJmuiden / sluis Noordzeekanaal*. 2005. URL: https://nl.wikipedia.org/wiki/Sluizen_van_IJmuiden.

[2] Benjamin K. Sovacool. "How long will it take? Conceptualizing the temporal dynamics of energy transitions". In: *Energy Research & Social Science* 13 (2016). Energy Transitions in Europe: Emerging Challenges, Innovative Approaches, and Possible Solutions, pp. 202–215. ISSN: 2214-6296. DOI: https://doi.org/10.1016/j.erss.2015.12.020. URL: https://www.sciencedirect.com/science/article/pii/S2214629615300827.

[3] Jesus Lago et al. "Optimal Control Strategies for Seasonal Thermal Energy Storage Systems With Market Interaction". In: *IEEE Transactions on Control Systems Technology* 29.5 (2021), pp. 1891–1906. DOI: 10.1109/TCST.2020.3016077.

[4] Christopher Ernest Clement. "Renewable Energy Transition: Dynamic Systems Analysis, Policy Scenarios, and Trade-offs for the State of Vermont". In: *Graduate College Dissertations and Theses* 601 (2016). URL: https://scholarworks.uvm.edu/graddis/601.

[5] Jesus Lago, Fjo De Ridder, and Bart De Schutter. "Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms". In: *Applied Energy* 221 (2018), pp. 386–405. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2018.02.069. URL: https://www.sciencedirect.com/science/article/pii/S030626191830196X.

[6] Cherrelle Eid et al. "Time-based pricing and electricity demand response: Existing barriers and next steps". In: *Utilities Policy* 40 (2016), pp. 15–25. ISSN: 0957-1787. DOI: https://doi.org/10.1016/j.jup.2016.04.001. URL: https://www.sciencedirect.com/science/article/pii/S0957178716300947.

[7] R. van Weissenbruch. "Gemaal IJmuiden krijgt module voor slim energieverbruik". In: *Land + Water* 4 (2005), pp. 14–15.

[8] Lucian Buşoniu et al. "Reinforcement learning for control: Performance, stability, and deep approximators". In: *Annual Reviews in Control* 46 (2018), pp. 8–28. ISSN: 1367-5788. DOI: https://doi.org/10.1016/j.arcontrol.2018.09.005. URL: https://www.sciencedirect.com/science/article/pii/S1367578818301184.

[9] David Silver Demis Hassabis. *AlphaGo Zero: Starting from scratch*. URL: https://deepmind.com/blog/article/alphago-zero-starting-scratch. 2017.

[10] Jonas Degrave et al. "Magnetic control of tokamak plasmas through deep reinforcement learning". In: *Nature* 602 (2022), pp. 414–419. ISSN: 1476-4687. DOI: https://doi.org/10.1038/s41586-021-04301-9. URL: https://www.nature.com/articles/s41586-021-04301-9.

[11] C.J.M. Vermeulen R.P. Versteeg. *Beslissings Ondersteunend Systeem NZK/ARK: modelvorming*. Tech. rep. HKV Lijn in water, Mar. 2002. URL: https://puc.overheid.nl/rijkswaterstaat/doc/PUC_132487_31/.

[12] Deltaprogramma. *Amsterdam-Rijnkanaal/Noordzeekanaal-gebied*. 2022. URL: https://www.deltaprogramma.nl/gebieden/amsterdam-rijnkanaal-noordzeekanaal.

[13] Elprocus. *What is a PID Controller : Working & Its Applications*. 2022. URL: https://www.elprocus.com/the-working-of-a-pid-controller/.

[14] Dawn Tilbury et al. *Introduction: PID Controller Design*. 2021. URL: https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID.

[15] Su Whan Sung and In-Beum Lee. "Limitations and Countermeasures of PID Controllers". In: *Industrial & Engineering Chemistry Research* 35.8 (1996), pp. 2596–2610. DOI: 10.1021/ie960090+. eprint: https://doi.org/10.1021/ie960090+. URL: https://doi.org/10.1021/ie960090+.

[16] Frank Allgöwer et al. *Model Predictive Control*. 20122. URL: https://www.ist.uni-stuttgart.de/research/group-of-frank-allgoewer/model-predictive-control/.

[17] Alexander Kuhnle, Michael Schaarschmidt, and Kai Fricke. *Tensorforce: a TensorFlow library for applied reinforcement learning*. Web page. 2017. URL: https://github.com/tensorforce/tensorforce.

[18] Ministerie van Verkeer en Waterstaat, Rijkswaterstaat, Directie Noord-Holland. *Noordzeekanaal / Amsterdam-Rijnkanaal: oppervlakte stroomgebied*. Tech. rep. Rijkswaterstaat, Jan. 1987. URL: https://puc.overheid.nl/rijkswaterstaat/doc/PUC_51981_31/.

[19] Rijkswaterstaat. *Noordzeekanaal*. 2022. URL: https://www.rijkswaterstaat.nl/water/vaarwegenoverzicht/noordzeekanaal.

[20] Rijkswaterstaat. *Amsterdam-Rijnkanaal*. 2022. URL: https://www.rijkswaterstaat.nl/water/vaarwegenoverzicht/amsterdam-rijnkanaal.

[21] M. Karelse and J.A.G. Van Gils. *Noordzeekanaal, Amsterdam-Rijnkanaal waterbeweging en zouthuishouding*. Tech. rep. Waterloopkundig laboratorium, Dec. 1991. URL: https://puc.overheid.nl/rijkswaterstaat/doc/PUC_130147_31/.

[22] Nelen & Schuurmans Consultants BV. *Wateropgave Boezemwateren*. Tech. rep. Nelen & Schuurmans Consultants BV, Mar. 2005. URL: https://www.agv.nl/contentassets/af47e62321754920892b2867ccac27c4/wateropgave-boezemwateren-2005.pdf.

[23] Rijkswaterstaat. "Operationeel Watermanagement Amsterdam-Rijnkanaal en Noordzeekanaal". In: Rijkswaterstaat, 2020. URL: https://www.helpdeskwater.nl/publish/pages/188956/rws_wvl_watersystemen-ark-nzk.pdf.

[24] De Nederlandse Gemalen Stichting. *Rijksgemaal IJmuiden*. 2022. URL: https://www.gemalen.nl/gemaal_detail.asp?gem_id=264.

[25] Noordhollands Dagblad. *Hopen dat het niet langdurig gaat plenzen: pomp van gemaal bij IJmuiden is kapot en dat gaat nog even duren*. 2020. URL: https://www.noordhollandsdagblad.nl/cnt/dmf20200717_44597434?utm_source=google&utm_medium=organic.

[26] R. van Weissenbruch. "Onderzoek energieverbruik gemaal IJmuiden". MA thesis. Delft, The Netherlands: TU Delft, 2003.

[27] Martin Behrendt. *Model predictive control*. 2005. URL: https://en.wikipedia.org/wiki/Model_predictive_control.

[28] Michael Nikolaou. "Model predictive controllers: A critical synthesis of theory and industrial needs". In: *Advances in Chemical Engineering* 26 (2001), pp. 131–204.

[29] T.J.T. van der Heijden et al. "Multi-market demand response from pump-controlled open canal systems: an economic MPC approach to pump-scheduling". In: *Journal of Hydroinformatics* 24.4 (2022). URL: http://resolver.tudelft.nl/uuid:22dfa410-bc39-485a-bfa9-b232e4329e9b.

[30] A. Goedbloed. *Kwaliteitsanalyse Beslissings Ondersteunend Systeem Noordzeekanaal/Amsterdam-Rijnkanaal*. Tech. rep. Delft University of Technology, Oct. 2006. URL: https://docplayer.nl/12957106-Kwaliteitsanalyse-beslissings-ondersteunend-systeem-noordzeekanaal-amsterdam-rijnkanaal.html.

[31] European Commission. *Energy storage – the role of electricity*. Tech. rep. European Commission, 2017. URL: https://energy.ec.europa.eu/document/download/f72c1756-20c1-4b26-8e22-bad5206bacf1_en?filename=swd2017_61_document_travail_service_part1_v6.pdf.

[32] European Commission et al. *Study on energy storage : contribution to the security of the electricity supply in Europe*. Publications Office, 2020. DOI: doi/10.2833/077257.

[33] DChao Luo, Jun Yang, and Yuanzhang Sun. "Risk Assessment of Power System Considering Frequency Dynamics and Cascading Process". In: (2018). URL: https://www.mdpi.com/1996-1073/11/2/422/pdf.

[34] Drax. *Why we need the whole country on the same frequency*. 2017. URL: https://www.drax.com/power-generation/need-whole-country-frequency/.

[35] Drax. *The great balancing act: what it takes to keep the power grid stable*. 2018. URL: https://www.drax.com/power-generation/great-balancing-act-takes-keep-power-grid-stable/.

[36] TenneT. *TenneT*. 2022. URL: https://netztransparenz.tennet.eu/.

[37] Florence School of Regulation. *Electricity markets in the EU*. 2020. URL: https://fsr.eui.eu/electricity-markets-in-the-eu/.

[38] Next Kraftwerke. *How does Day-Ahead Trading of Electricity work?* 2022. URL: https://www.next-kraftwerke.be/en/knowledge-hub/day-ahead-trading/.

[39] EPEX SPOT. *Basics of the Power Market*. 2018. URL: https://www.epexspot.com/en/basics powermarket.

[40] EEX Group. *EPEX SPOT Annual Market Review 2021*. Jan. 2022. URL: https://www.eex-group.com/en/newsroom/detail?tx_news_pi1%5C%5Baction%5C%5D=detail&tx_news_pi1%5C%5Bcontroller%5C%5D=News&tx_news_pi1%5C%5Bnews%5C%5D=4177%5C&cHash=97dd7520461e5186532c01766bce4750.

[41] ENTSO-E. *Day-ahead prices*. 2022. URL: https://transparency.entsoe.eu/dashboard/show.

[42] Claire Gavin. *Seasonal variations in electricity demand*. Tech. rep. Electrical Statistics, 2014. URL: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/295225/Seasonal_variations_in_electricity_demand.pdf.

[43] EPEX SPOT. *Trading Products*. 2018. URL: https://www.epexspot.com/en/tradingproducts.

[44] Nord Pool. *Intraday market*. 2021. URL: https://www.nordpoolgroup.com/en/the-power-market/Intraday-market/.

[45] Josh Achiam. *OpenAI Spinning Up*. 2018. URL: https://spinningup.openai.com/en/latest/index.html.

[46] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Second. The MIT Press, 2018. URL: http://incompleteideas.net/book/the-book-2nd.html.

[47] David Silver. *Lectures on Reinforcement Learning*. URL: https://www.davidsilver.uk/teaching/. 2015.

[48] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *Nature* 518 (2015), pp. 529–533. DOI: 10.1038/nature14236. URL: https://doi.org/10.1038/nature14236.

[49] John Schulman et al. *Trust Region Policy Optimization*. 2015. DOI: 10.48550/ARXIV.1502.05477. URL: https://arxiv.org/abs/1502.05477.

[50] John Schulman et al. *Proximal Policy Optimization Algorithms*. 2017. DOI: 10.48550/ARXIV.1707.06347. URL: https://arxiv.org/abs/1707.06347.

[51] Ziyu Wang et al. *Sample Efficient Actor-Critic with Experience Replay*. 2016. eprint: arXiv:1611.01224.

[52] Sergey Ioffe and Christian Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015. DOI: 10.48550/ARXIV.1502.03167. URL: https://arxiv.org/abs/1502.03167.

[53] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. *Layer Normalization*. 2016. DOI: 10.48550/ARXIV.1607.06450. URL: https://arxiv.org/abs/1607.06450.

[54] Tim Salimans and Diederik P. Kingma. *Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks*. 2016. DOI: 10.48550/ARXIV.1602.07868. URL: https://arxiv.org/abs/1602.07868.

[55] John Schulman et al. *High-Dimensional Continuous Control Using Generalized Advantage Estimation*. 2015. eprint: arXiv:1506.02438.

[56] Volodymyr Mnih et al. "Asynchronous Methods for Deep Reinforcement Learning". In: (2016). eprint: arXiv:1602.01783.

[57] Matteo Hessel et al. *Rainbow: Combining Improvements in Deep Reinforcement Learning*. 2017. DOI: 10.48550/ARXIV.1710.02298. URL: https://arxiv.org/abs/1710.02298.

[58] Volodymyr Mnih et al. “Playing Atari with Deep Reinforcement Learning”. In: *CoRR* abs/1312.5602 (2013). arXiv: 1312.5602. URL: http://arxiv.org/abs/1312.5602.

[59] Yuxi Li. *Deep Reinforcement Learning: An Overview*. 2017. DOI: 10.48550/ARXIV.1701.07274. URL: https://arxiv.org/abs/1701.07274.

[60] Hado van Hasselt, Arthur Guez, and David Silver. *Deep Reinforcement Learning with Double Q-learning*. 2015. DOI: 10.48550/ARXIV.1509.06461. URL: https://arxiv.org/abs/1509.06461.

[61] Ian Dewancker, Michael McCourt, and Scott Clark. “Bayesian optimization primer”. In: (2015). URL: https://app.sigopt.com/static/pdf/SigOpt_Bayesian_Optimization_Primer.pdf.

[62] James Bergstra, Daniel Yamins, and David Cox. “Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures”. In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by Sanjoy Dasgupta and David McAllester. Vol. 28. Proceedings of Machine Learning Research 1. Atlanta, Georgia, USA: PMLR, June 2013, pp. 115–123. URL: https://proceedings.mlr.press/v28/bergstra13.html.

[63] James Bergstra et al. “Algorithms for Hyper-Parameter Optimization”. In: *Advances in Neural Information Processing Systems*. Ed. by J. Shawe-Taylor et al. Vol. 24. Curran Associates, Inc., 2011. URL: https://proceedings.neurips.cc/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf.

[64] T. van der Heijden. “Pumping when the wind blows”. MA thesis. Delft, The Netherlands: TU Delft, 2019.

[65] Rijkswaterstaat. *Rijskwaterstaat Waterinfo*. 2022. URL: https://waterinfo.rws.nl/.

[66] LLC Gurobi Optimization. *Gurobi optimizer reference manual*. 2022. URL: http://www.gurobi.com.

[67] Bastiaan Kuijper Chris Geerse. *Probabilistisch model frequentielijnen IJsselmeergebied: Hoofdrapport van model DEZY*. Tech. rep. HKV lijn in water, May 2015. URL: https://aandeslagmetdeomgevingswet.nl/@205902/probabilistisch/.

[68] B. Pengel and H. Geerse. *Overschrijdingskansen van Waterstanden in het Noordzeekanaal en het Amsterdam-Rijnkanaal*. Tech. rep. HKV lijn in water, Nov. 2001. URL: http://resolver.tudelft.nl/uuid:df35bc24-6a8d-47e3-bc52-e628f1c7bf72.

[69] J.A. Zindler et al. *Het Noordzeekanaal in cijfers anno 2004*. Tech. rep. Rijkswaterstaat, 2004. URL: https://puc.overheid.nl/rijkswaterstaat/doc/PUC_163669_31/.

[70] A.G. Maris et al. *Rapport Deltacommissie. Deel 3. Bijdragen 2: Beschouwingen over stormvloeden en getijbeweging*. Tech. rep. Rijkswaterstaat, 1961. URL: http://resolver.tudelft.nl/uuid:046f06e8-5127-4e49-adfd-a496b4fedbb5.

[71] KNMI. *Daggegevens van het weer in Nederland*. 2022. URL: https://www.knmi.nl/nederland-nu/klimatologie/daggegevens.

[72] Michael Schaarschmidt et al. “LIFT: Reinforcement Learning in Computer Systems by Learning From Demonstrations”. In: *CoRR* abs/1808.07903 (2018). arXiv: 1808.07903. URL: http://arxiv.org/abs/1808.07903.

[73] Richard Ernest Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[74] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. DOI: 10.48550/ARXIV.1412.6980. URL: https://arxiv.org/abs/1412.6980.

[75] Ziyu Wang, Nando de Freitas, and Marc Lanctot. “Dueling Network Architectures for Deep Reinforcement Learning”. In: *CoRR* abs/1511.06581 (2015). arXiv: 1511.06581. URL: http://arxiv.org/abs/1511.06581.

[76] Hado Hasselt. “Double Q-learning”. In: *Advances in Neural Information Processing Systems*. Ed. by J. Lafferty et al. Vol. 23. Curran Associates, Inc., 2010. URL: https://proceedings.neurips.cc/paper/2010/file/091d584fced301b442654dd8c23b3fc9-Paper.pdf.

[77] Marc G. Bellemare, Will Dabney, and Rémi Munos. "A Distributional Perspective on Reinforcement Learning". In: *CoRR* abs/1707.06887 (2017). arXiv: 1707.06887. URL: http://arxiv.org/abs/1707.06887.

[78] Martin Klaučo, Martin Kalúz, and Michal Kvasnica. "Machine learning-based warm starting of active set methods in embedded model predictive control". In: *Engineering Applications of Artificial Intelligence* 77 (2019), pp. 1–8. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2018.09.014. URL: https://www.sciencedirect.com/science/article/pii/S0952197618302008.

A

Power Consumption Optimization

To fit a single power curve for the combination of the six pumps at IJmuiden, the Gurobi optimizer was used to find the most efficient combination of pumps for the combinations of discharges, Q, and pump heights, dH. This was achieved by formulating the optimization as a MIP, with an objective and constraints that were solved by the Gurobi optimizer [66].

For each situation, the most energy-efficient combination of pumps was found by using the Q-dH and P-dH curves of all pumps. A summary of these relationships can be seen in Table A.1 with the symbols to distinguish between the pumps and their operating modes that will be used in the optimization. The subscripts show the number of the pump and when multiple modes are possible, the number of the mode.

Table A.1: Pump discharge and power relationships for all six pumps in the IJmuiden pumping station [26]

Pump	Q-dH $[m^3/s],[m]$	P-dH $[kW],[m]$
1, 3	$Q_{1,3} = -5.4174 \cdot dH + 44.93$	$P_{1,3} = 208.08 \cdot dH + 536.85$
2, 4	$Q_{21,41} = -5.4174 \cdot dH + 44.93$	$P_{21,41} = 208.02 \cdot dH + 536.85$
	$Q_{22,42} = -6.4977 \cdot dH + 33.149$	$P_{22,42} = 192.36 \cdot dH + 217.26$
5, 6	$Q_{51,61} = -1.9822 \cdot dH^2 + 1.9726 \cdot dH + 44.93$	$P_{51,61} = 443.91 \cdot dH + 476.30$
	$Q_{52,62} = -1.8544 \cdot dH^2 + 7.7740 \cdot dH + 44.93$	$P_{52,62} = 379.09 \cdot dH + 373.18$
	$Q_{53,63} = -7.1021 \cdot dH + 48.164$	$P_{53,63} = 282.97 \cdot dH + 417.32$

A.1. Optimization model

The model used for the power consumption optimization consisted of several components. Firstly the discharge was determined with the Q-dH curves of all pumps combined with binary variables indicating which pumps were being used. The binary variables, B_x, had a value of 0 or 1 indicating an inactive and active pump respectively.

$$\begin{aligned} Q_{approx}(dH)\,[m^3/s] =& B_1 \cdot Q_1(dH) + B_{21} \cdot Q_{21}(dH) + B_{22} \cdot Q_{22}(dH) + B_3 \cdot Q_3(dH) + \\ & B_{41} \cdot Q_{41}(dH) + B_{42} \cdot Q_{42}(dH) + B_{51} \cdot Q_{51}(dH) + B_{52} \cdot Q_{52}(dH) + \\ & B_{53} \cdot Q_{53}(dH) + B_{61} \cdot Q_{61}(dH) + B_{62} \cdot Q_{62}(dH) + B_{63} \cdot Q_{63}(dH) \end{aligned} \tag{A.1}$$

Constraints were added to ensure that multiple pump modes could not be activated simultaneously. This was done by making sure only one binary variable of each pump could be nonzero. The indexing of the binary variables was done in the same manner as in Table A.1, the first subscript showed the number of the pump and the second the pump mode.

$$
\begin{aligned}
&B_{21} \cdot B_{22} \leq 1e-3 \\
&B_{41} \cdot B_{42} \leq 1e-3
\end{aligned}
\qquad
\begin{aligned}
&B_{51} \cdot B_{52} \leq 1e-3 \\
&B_{51} \cdot B_{53} \leq 1e-3 \\
&B_{52} \cdot B_{53} \leq 1e-3
\end{aligned}
\qquad
\begin{aligned}
&B_{61} \cdot B_{62} \leq 1e-3 \\
&B_{61} \cdot B_{63} \leq 1e-3 \\
&B_{62} \cdot B_{63} \leq 1e-3
\end{aligned}
\tag{A.2}
$$

It was not always possible to achieve the exact desired discharge by combining the six available pump curves. To overcome this, the approximated discharge could have a maximum deviation of $5m^3/s$ from the discharge for which the power was optimized. In reality, all discharges can be achieved within the feasible region (shown in Figure 5.2). However, this could not be done with the optimization as the variable speed pumps (5, 6) were approximated with three distinct discharge modes [26].

$$
(Q_{approx} - Q)^2 \leq 5^2 \tag{A.3}
$$

Finally, the objective of the optimization was to minimize the sum of the power consumption of all the pumps. This total was determined with the *P-dH* curves of all the pumps combined with the binary variables indicating which pumps were activated.

$$
\begin{aligned}
P(dH)\,[kW] = &B_1 \cdot P_1(dH) + B_{21} \cdot P_{21}(dH) + B_{22} \cdot P_{22}(dH) + B_3 \cdot P_3(dH) + \\
&B_{41} \cdot P_{41}(dH) + B_{42} \cdot P_{42}(dH) + B_{51} \cdot P_{51}(dH) + B_{52} \cdot P_{52}(dH) + \\
&B_{53} \cdot P_{53}(dH) + B_{61} \cdot P_{61}(dH) + B_{62} \cdot P_{62}(dH) + B_{63} \cdot P_{63}(dH)
\end{aligned}
\tag{A.4}
$$

A.2. MIP results

To determine an accurate power consumption throughout the whole feasible workspace, the optimization was performed for a range of discharges and pump heights. The discharges considered ranged from $0-260m^3/s$ with a step size of $5m^3/s$. The pump height ranged from $0-5m$ with a step size of $0.05m$. With each discharge and pump height combination, the optimal values for the binary variables were determined, in other words, the optimal pump configuration.

The resulting optimized power consumption can be seen in Figure A.1. The missing data was due to some discharge and pump height combinations not having a possible pump configuration. As explained above, this was due to the approximation of the variable speed pumps. The dotted line shows the boundary of the feasible workspace for pumping.

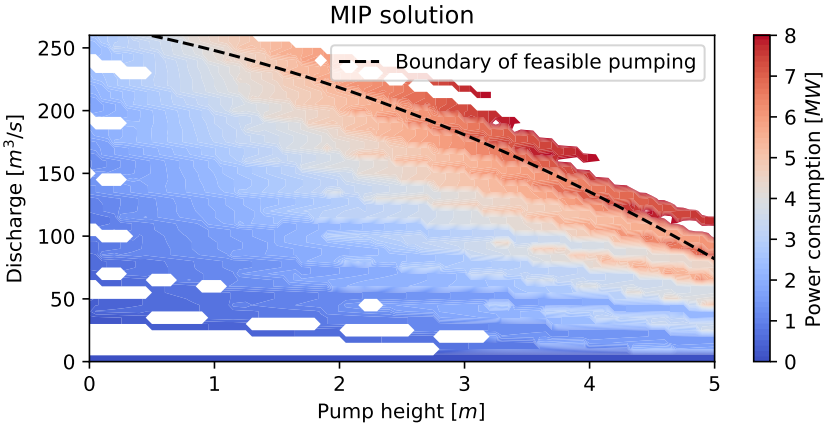


Figure A.1: MIP results for the power consumption of the optimal pump configuration

A.3. Fitted power consumption

In order to apply the derived power relationship, a quadratic curve was fitted using a least-squares optimization, as shown in Equation (A.5). The solution can be found in Equation (A.6). The results outside the feasible pumping region (to the right of the dotted line in Figure A.1) were not included in the least-squares optimization. This ensured the most accurate fit inside the feasible workspace. The power consumption will not be calculated for the infeasible workspace and therefore the power relationship does not need to be accurate for those combinations of discharge and pump height.

$$\begin{bmatrix} dH & Q & dH^2 & dH \cdot Q & Q^2 \end{bmatrix} \cdot \vec{x} = P \tag{A.5}$$

$$\begin{aligned} P_p\,[kW] &= a \cdot dH + b \cdot Q + c \cdot dH^2 + d \cdot dH \cdot Q + e \cdot Q^2 \\ a &= -2.64e + 02 \qquad d = 8.53e + 00 \\ b &= 8.30e + 00 \qquad e = 2.77e - 03 \\ c &= 1.03e + 02 \end{aligned} \tag{A.6}$$

The fitted solution is visualized in Figure A.2. This solution had an average absolute residual error of $0.29MW$, when compared to the MIP solution.

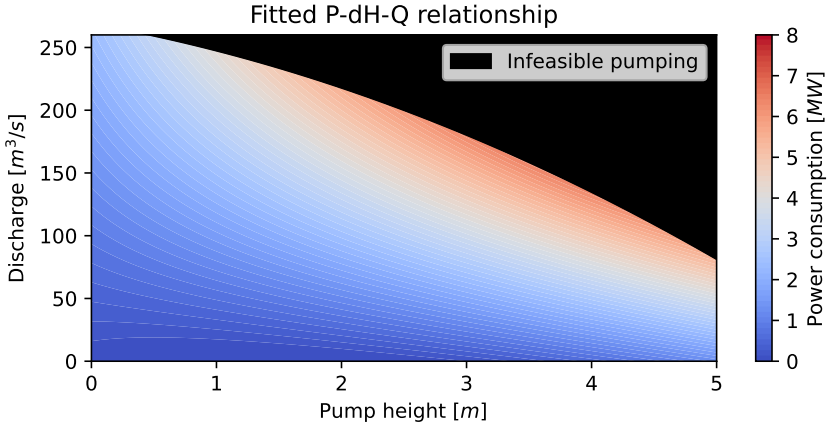


Figure A.2: Fitted *P-Q-dH* relationship for the MIP results for the power consumption of the optimal pump configuration

B

Wind Set-Up

The wind acts as a shear stress on the water surface, which is compensated by the gradient of the water level in closed basins and lakes. The NZK can be treated as such a closed basin. The maximum set-up occurs when the fetch (the length over which the wind shear stress acts) is largest. When the basin is simplified as a rectangular shape, the maximum and minimum wind set-up are equal as the centre of gravity of the water body is halfway the canal.

Equation (B.1) was used for the calculation in combination with the fetch of the canal for all wind directions. The effect of the wind direction, $cos\phi$, was not included in the calculation as the fetch was only non-zero for the relevant directions.

$$W = 0.5 * \kappa * \frac{u_{10}^2}{gd} * F * cos\phi \tag{B.1}$$

where:

W	Wind setup $[m]$
κ	Friction constant $[-]$
u_{10}	Wind velocity at 10m height $[m/s]$
g	Acceleration due to gravity $[m/s^2]$
d	Water depth $[m]$
F	Fetch $[m]$
ϕ	Angle between the land and wind $[rad]$

There are several important assumptions when using Equation (B.1) to calculate the wind set-up. The basin is rectangular with a constant rectangular cross-section and there are no inflowing or outflowing discharges. These assumptions can be used due to the shape of the canal and the relatively small discharges compared to the size of the *basin*. The time required to develop the water level difference is not taken into account.

C

Simulated Discharges For Training

The state space was explored further by training the RL model with historical discharge data as well as simulated discharges. The discharges were simulated based on the distribution of the historical data to produce realistic training examples. An initial discharge was sampled after which random changes in discharge between time steps were chosen.

The initial discharge was randomly selected using one of two distributions. There was a 20% chance of initializing the discharge with a high value. This situation occurs less frequently in the data and therefore was generated more often in the simulated discharges. The discharge was sampled from a uniform distribution between $100m^3/s$ and $250m^3/s$. $250m^3/s$ was chosen as the upper bound as this was close to the maximum possible pump discharge which therefore resulted in a more complex optimization problem that required pumping. This could also create situations where maintaining safe water levels was not possible even if the maximum outflow action was consistently chosen.

For the other 80% of the samples, a more realistic initial discharge was sampled. Figure C.1 shows a kernel density estimate for the training data as well as a fitted Weibull distribution. The Weibull distribution (shape parameter: $\beta = 1.6$, scale parameter: $\eta = 70$) was used to sample initial discharges. Negative discharges were not included as these only rarely occur and in these cases, the discharges are extremely small for short periods of time.

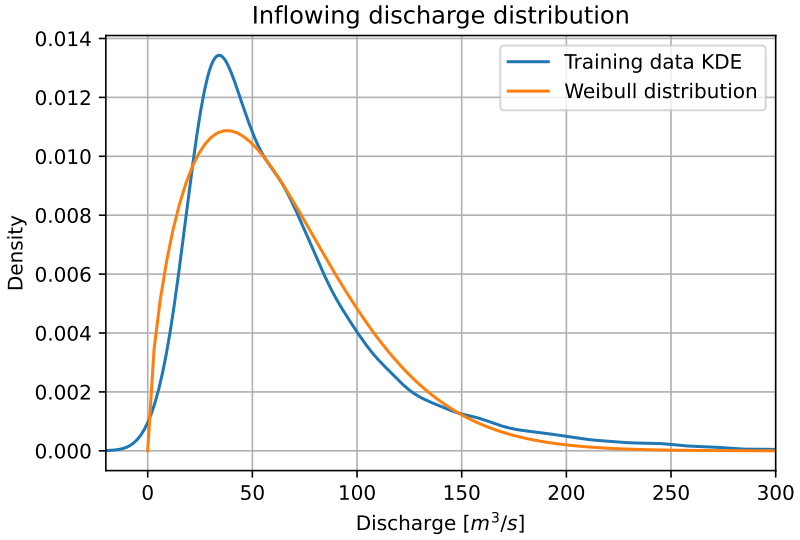


Figure C.1: Inflowing discharge distribution

After sampling the initial discharge, a change in the discharge for every time step was sampled in order to generate a time series for the entire training episode. The change in discharge was sampled from a fitted normal distribution ($\mu = 0m^3/s, \sigma = 13m^3/s$) as shown in Figure C.2. This distribution was used for both high and regular discharge situations.

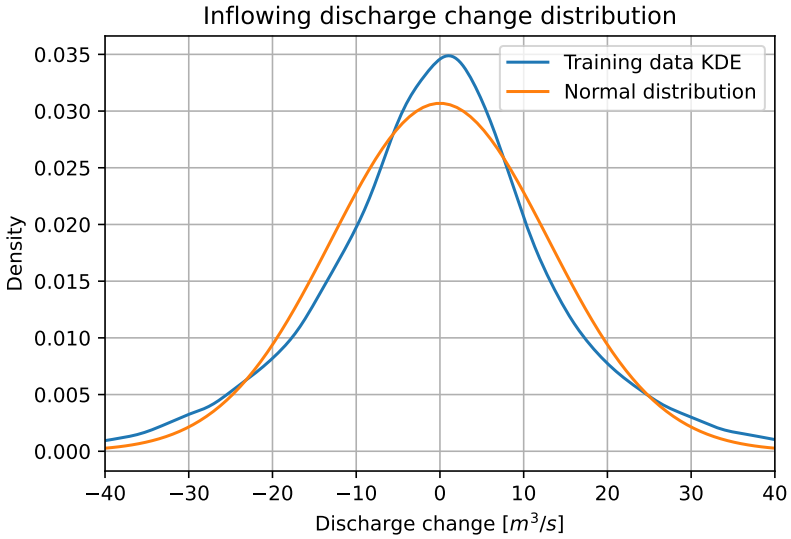


Figure C.2: Inflowing discharge change distribution

All discharges were capped at $-10m^3/s$ and $260m^3/s$. The lower cap allowed negative discharges to occasionally occur. The upper bound was set at $260m^3/s$ as that was the absolute maximum that could be pumped. If there are slight increases in pump height, pumping $260m^3/s$ was no longer possible. As a result, these inflowing discharges would be a challenge to maintain within the target water level range. This allowed the agent to train in extreme cases and improve performance outside of the regular ranges.

Examples of high and regular simulated discharges can be seen in Figure C.3. During training, the time series were generated randomly and therefore varied between retrained models.

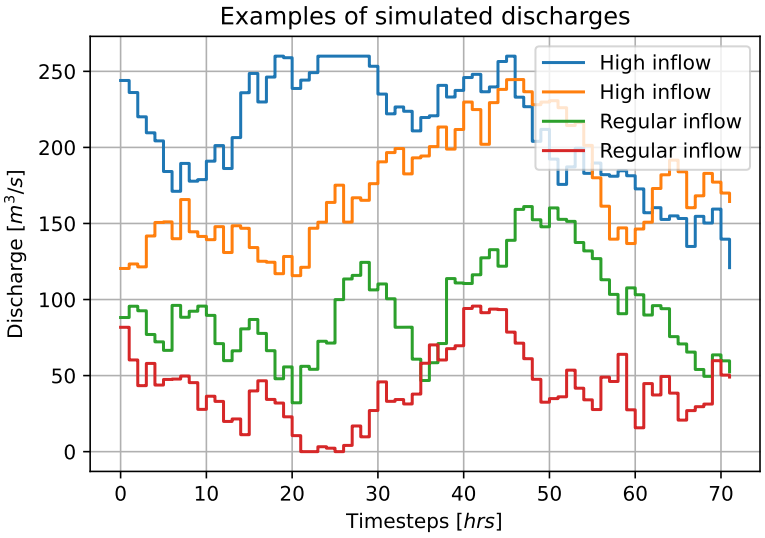


Figure C.3: Simulated inflowing discharges examples

D

Train, Validation, Test Split

D.1. Data clustering

To ensure that the available data was split to incorporate wet and dry data in both the training and test set, the data was clustered by month. The inflowing discharge data was first split by month and a kernel density estimate was computed using a Gaussian kernel with a bandwidth of 15. These kernels were split into groups using the k-means algorithm with two clusters. This separated the samples into groups of equal variance, representing the wet and dry months. The kernel densities and clusters found can be seen in Figure D.1.

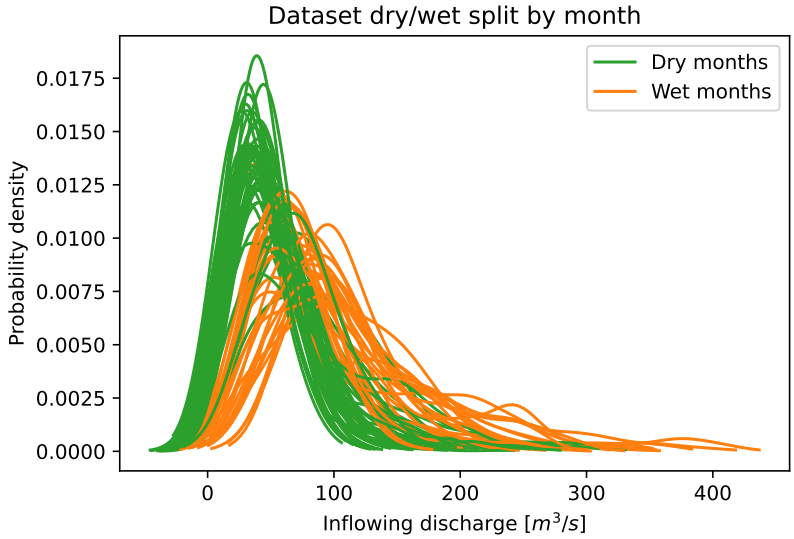


Figure D.1: Clustered kernel densities per month wet/dry

The same clustering was performed for the DAM price, as shown in Figure D.2, to split the data into expensive and cheap months. The size of the two clusters varied significantly with far more months falling into the cheap category.

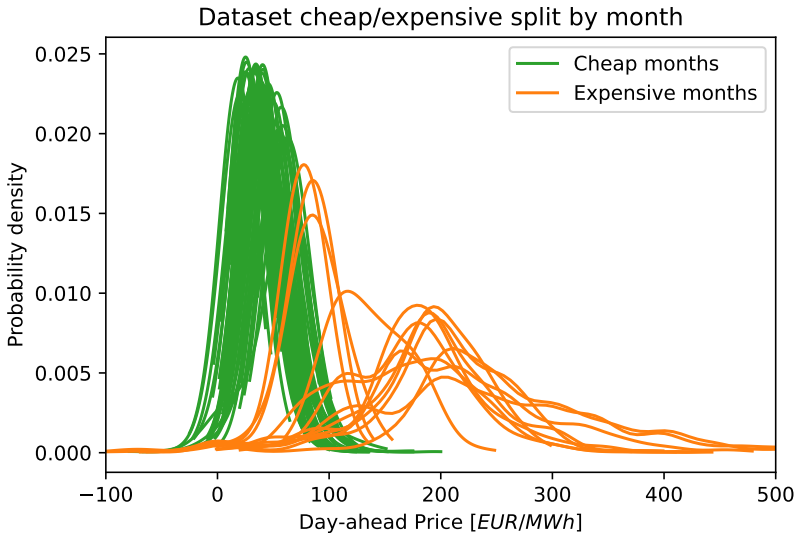
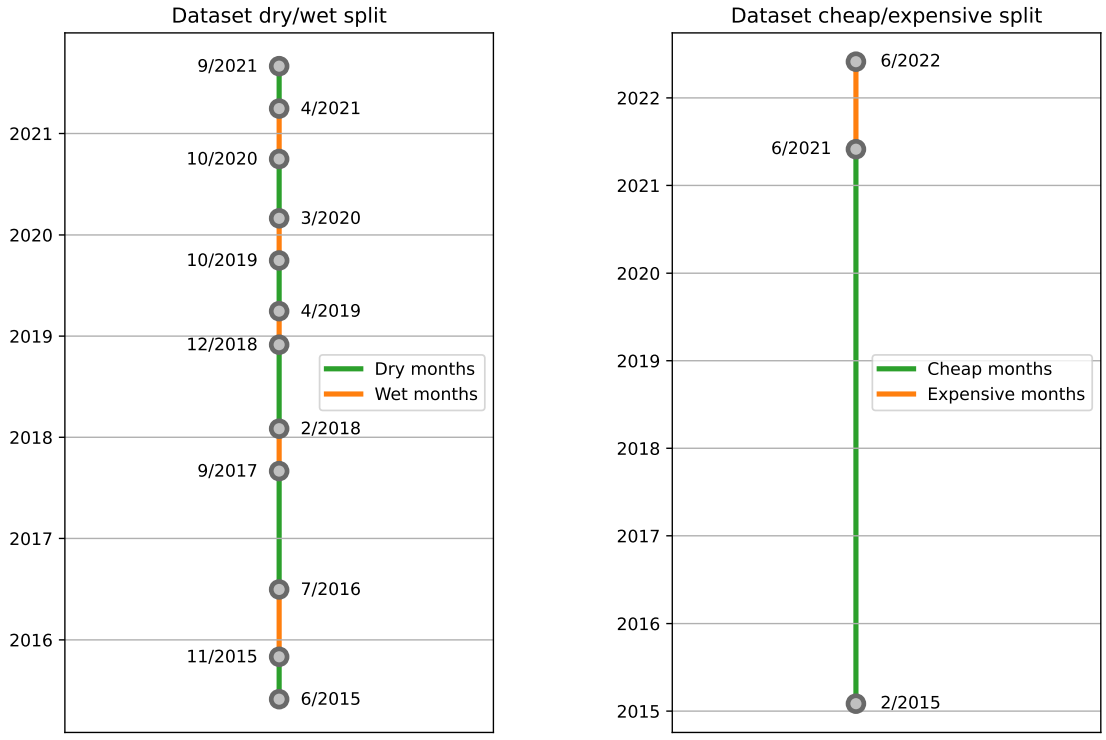


Figure D.2: Clustered kernel densities per month cheap/expensive

To visualize the data splits in time, Figure D.3 shows two timelines for the splits shown in Figures D.1 and D.2. A clear difference is highlighted between the discharge and DAM price trends. The discharges show a seasonal variation where the winter months are generally wetter than the summer months. In comparison, the electricity prices show a drastic increase in the months after June 2021.



(a) Timeline of wet and dry months

(b) Timeline of cheap and expensive months

Figure D.3: Timelines for wet/dry and cheap/expensive dataset split

D.2. Data split

After clustering the data, the available data was separated for the three phases of training the RL agent; training, validation, and testing. All data sets needed to contain enough data for each of the clustered categories. Due to the small size of the expensive DAM prices cluster, these were not included in the validation or testing data sets. To train the agent for these situations, more data would need to be available. The wet and dry clusters could be more easily divided due to the seasonal variations causing nearly all years to contain an adequate amount of wet and dry months.

First, a suitable and relevant section of the data was selected for testing. Two years of data would allow enough examples for the agent performance to be evaluated. Different scenarios could also be run with the same data by initializing the water level with a different value. As shown in Figure D.3, during the summer of 2021 there was a significant increase in electricity prices. Therefore, the years 2019 and 2020 were selected for testing. This ensured all data fell into the cheap cluster and a representative percentage of wet and dry months was included.

Validation was performed with data in 2017. As with testing, different initializations of the water system meant that limited data could still give an accurate estimate of the performance of the model. Both the wet and dry clusters were represented in the data split.

Finally, all the remaining data was used for training. This also included simulated discharges and DAM prices from Belgium (BE) and Germany (DE-LU). This ensured the necessary amount of wet and dry data. A summary of the data used for the three phases can be seen below:

- Testing: 01/01/2019 - 01/01/2021
  - All inputs were measurements from IJmuiden combined with DAM prices from The Netherlands
- Validation: 01/01/2017 - 01/01/2018
  - All inputs were measurements from IJmuiden combined with DAM prices from The Netherlands
- Training set: All remaining data
  - Discharges: 01/06/2015 - 13/09/2021 (excluding dates above) or simulated data
  - Sea level: 31/12/2014 - 28/07/2022 (excluding dates above)
  - Electricity prices (NL): 05/01/2015 - 18/07/2022 (excluding dates above)
  - Electricity prices (BE): 05/01/2015 - 18/07/2022
  - Electricity prices (DE-LU): 01/10/2018 - 18/07/2022

Figure D.4 shows a timeline per country for how the data was split for the three phases. All the discharge and sea level data was used from The Netherlands while the electricity prices were sampled from either The Netherlands, Belgium, or Germany.

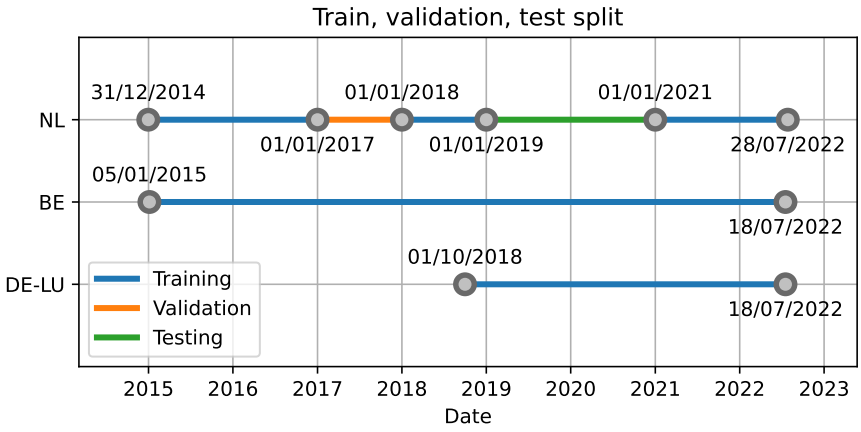


Figure D.4: Timeline of the train, validation, test split for data per country

E

Test Scenarios

To evaluate the performance of the RL model in comparison to the MPC created by [64], test several scenarios were chosen. Each represents a different optimization challenge. The scenarios were selected for specific situations regarding the inflowing discharges, sea levels, and electricity prices.

E.1. Discharge scenarios

Three scenarios were selected; extreme high inflow, high inflow, and low inflow, shown in Figure E.1. This showed how the behaviour of both controllers changes for different situations. The extreme inflow will show how well they are able to cope when it is no longer possible to maintain the water level within the target range. The high inflow will allow the pumping strategy to be compared while the low inflow can be used to determine whether the full potential of the gate is used.

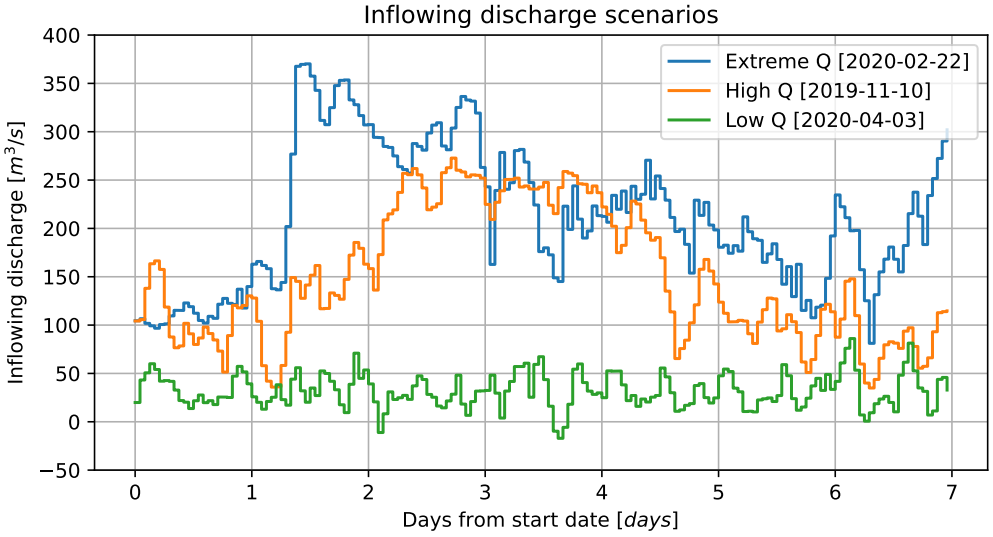


Figure E.1: Discharge scenarios of 1 week for testing model performance

E.2. Sea level scenario

The control can become more challenging if the water level in the North Sea is very high. This reduces the time steps for which opening the gate is possible, necessitating the use of the pumps. This high sea level scenario was run twice, once with the actual inflowing discharges and a second time with increased discharges to force the controllers to use the pumps.

The discharge in the week following 2020-09-10 was very low and therefore the pump was hardly needed even with the high sea levels. To increase the difficulty of the optimization, the inflowing

discharge was increased by $100m^3/s$ for the entire scenario to ensure that the pumps were used.

Figure E.2 shows the sea levels for the test scenario, showing how the time steps where it is possible to open the gate are limited.

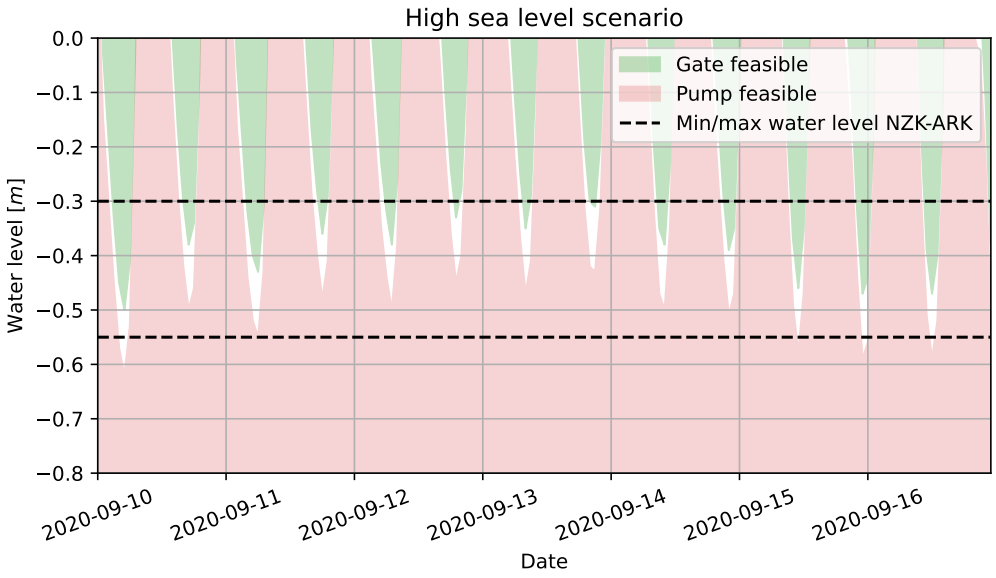


Figure E.2: Sea level scenario of 1 week for testing model performance

E.3. Electricity price scenarios

Several scenarios were chosen for testing how well the models could cope with the electricity price. A relatively high price was chosen to evaluate whether the trade-off between exceeding the water level and paying a high price for electricity is done correctly. Secondly, a week was chosen where the price decreased rapidly for a short period of time. Finally, a data set was generated by transforming the prices of 2020-02-10. This resulted in a week where the price was negative on several occasions.

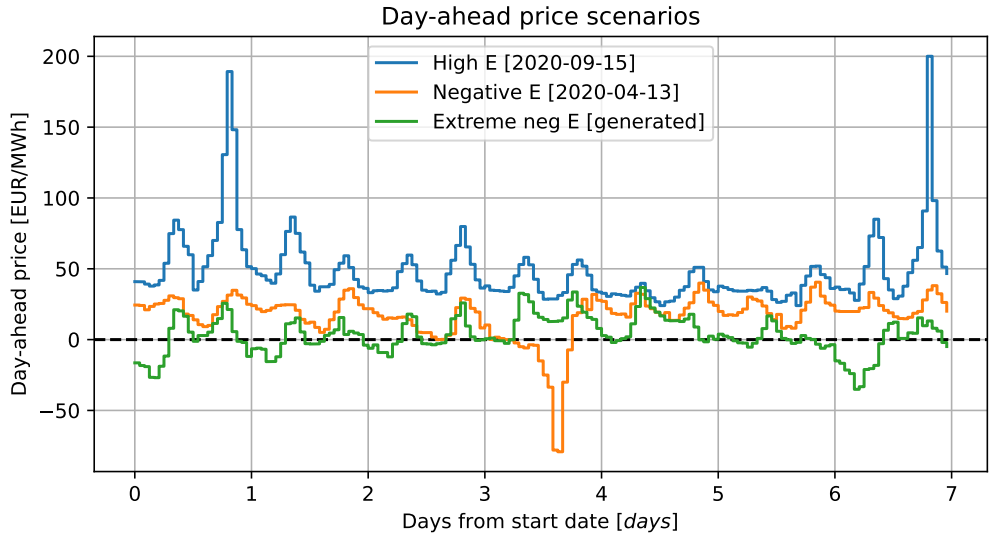


Figure E.3: Electricity price scenarios of 1 week for testing model performance

F

Hyperparameter optimization

After the initial RL model was developed where only the water level objective was included, a first Bayesian hyperparameter optimization was performed. This tuned the model to the specific problem, which resulted in a significant speedup and a slight performance increase. Figures F.1 and F.2 show the results of the hyperparameter optimization and the effect on the training time. The parameters that were tuned are shown in Table F.1 with the search space that was used.

Table F.1: Hyperparameters tuned for the water level objective including the search space, best performing value, and chosen value

Hyperparameter	Search space	Best performance	Chosen value
Batch size	Exp(x), where x has a discrete uniform distribution between [1, 6].	20	20
Update frequency	A uniform distribution between [0, 1].	0.03	0.15
Replay memory	A discrete uniform distribution between [450, 3000] with intervals of 50.	3000	3000
Network depth	Randomly integer between [2, 10].	8	8
Network width	Randomly selected from [16, 32, 64, 128, 256, 512].	16	16
Dropout rate	A uniform distribution between [0, 1].	0.21	0.25
Learning rate	A uniform distribution between [1e-5, 1e-2].	7e-4	7e-4
Horizon	A random integer between [1, 10].	2	2
Discount factor	A uniform distribution between [0, 1].	0.46	0.75

Due to the variability in the final performance of the agent, not only the best performing hyperparameters were considered. The final chosen parameters were based on the 20 best performing agents. The loss, training times, and correlation between the parameters were taken into account. Figures F.3 and F.4 show two examples of the correlation between the batch size and a second hyperparameter. The hyperparameter optimization does not take the correlation into account, and it was therefore considered separately when choosing the final parameters. The update frequency and discount rate were chosen higher than the best performing values due to the other parameter configurations with high performance.

After the implementation of the cost objective in the rewards, a second less extensive hyperparameter optimization was performed. The trials were performed in the same search space as the initial optimization. The similarities with the initial model meant that there were only two parameters that were chosen differently: the replay memory capacity and network depth were set to 2500 and 4, respectively. Fewer hidden layers meant that the training speed increased and that the network was

less complex, which is desirable. Since the agent was able to achieve similar performance, the network depth could be significantly decreased.

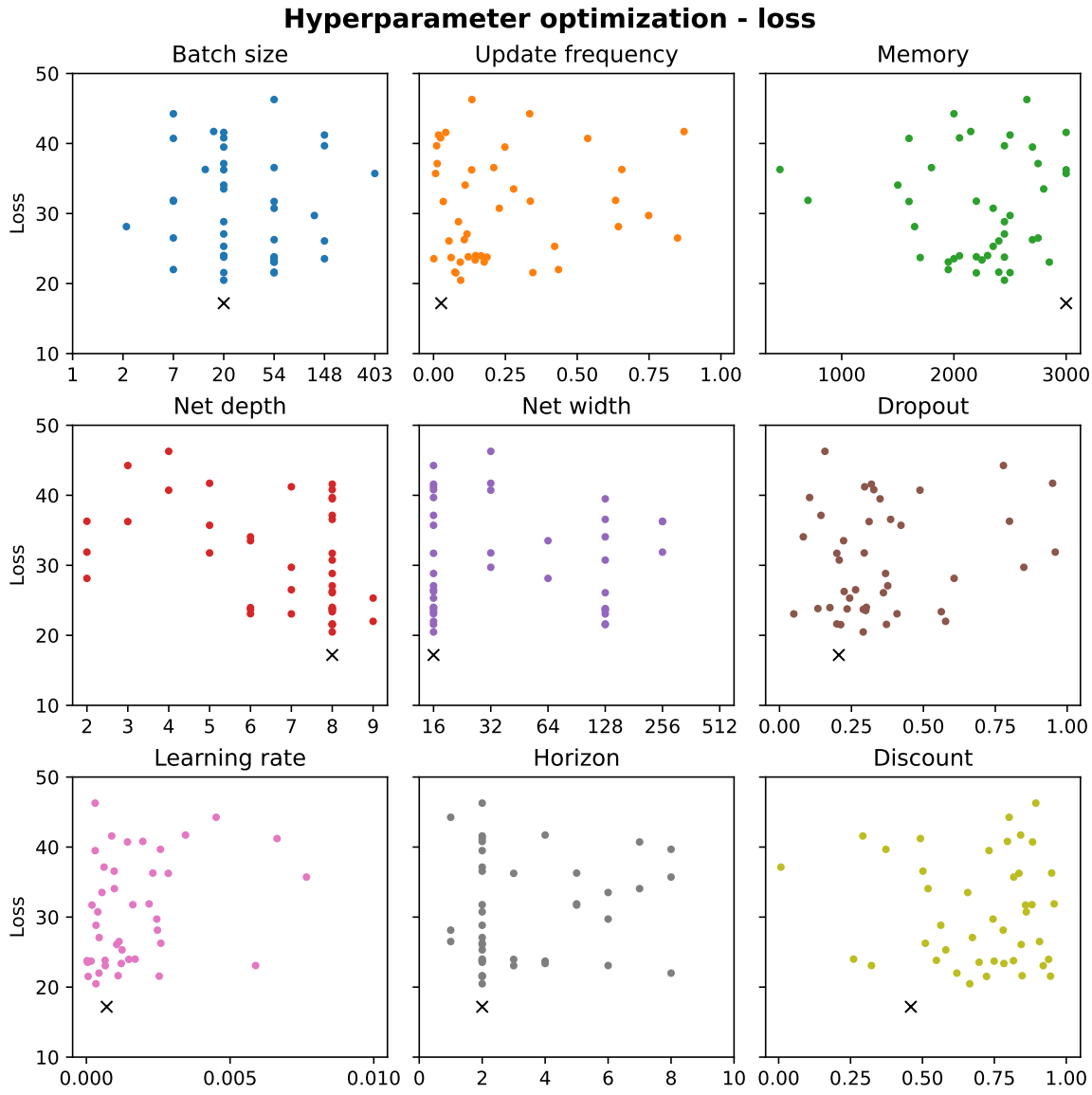


Figure F.1: Hyperparameters optimization showing the loss for each parameter value and a cross for the best performance found. Trials with a loss above 50 were not included. The loss was set to be the negative reward, which was minimized in the optimization.

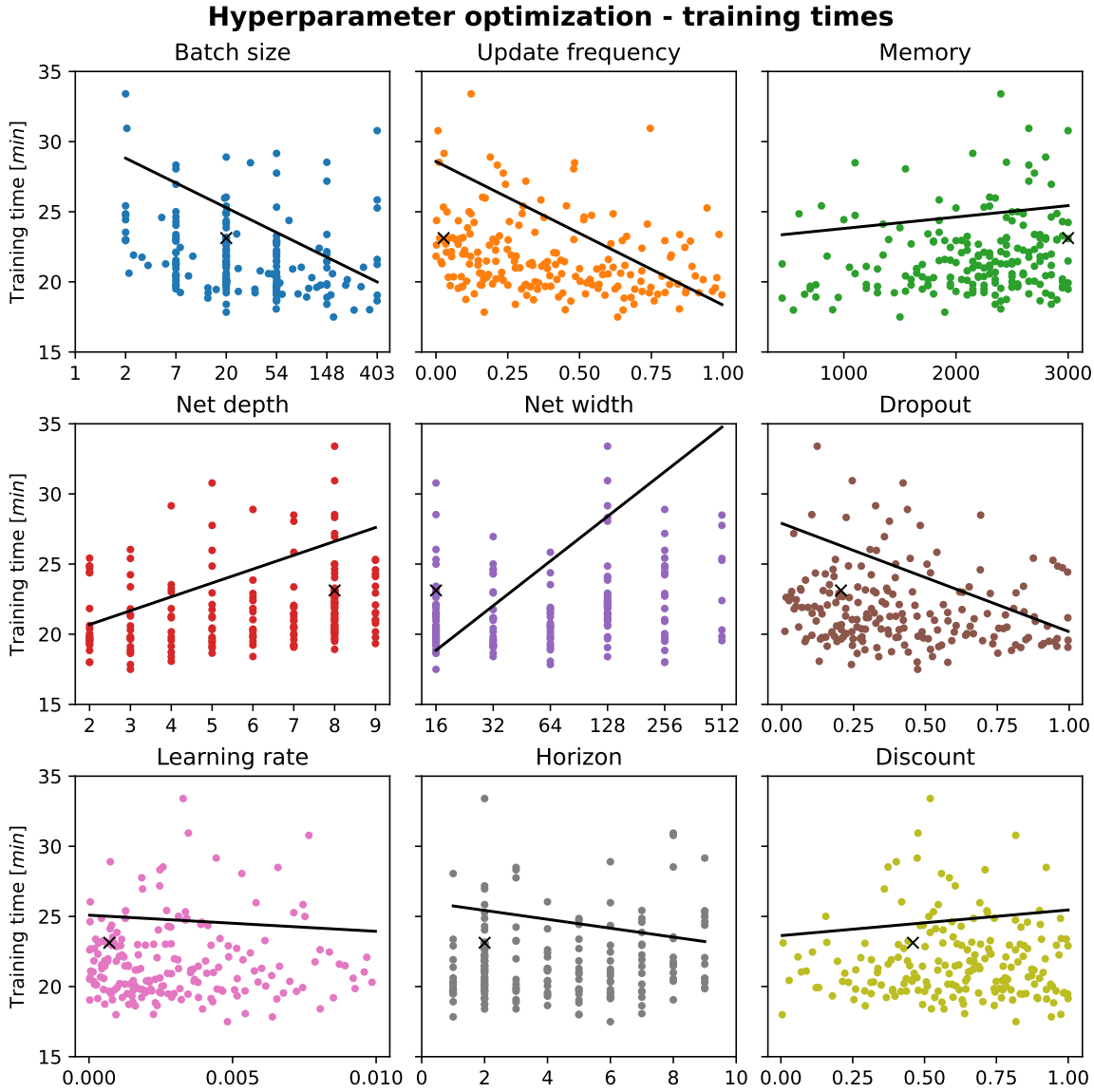


Figure F.2: Hyperparameters optimization showing the training time for each parameter value and a cross for the best performance found. Trials with a training time above 35 minutes were not included. A linear fit gives an indication of the effect of the parameter value on the training time.

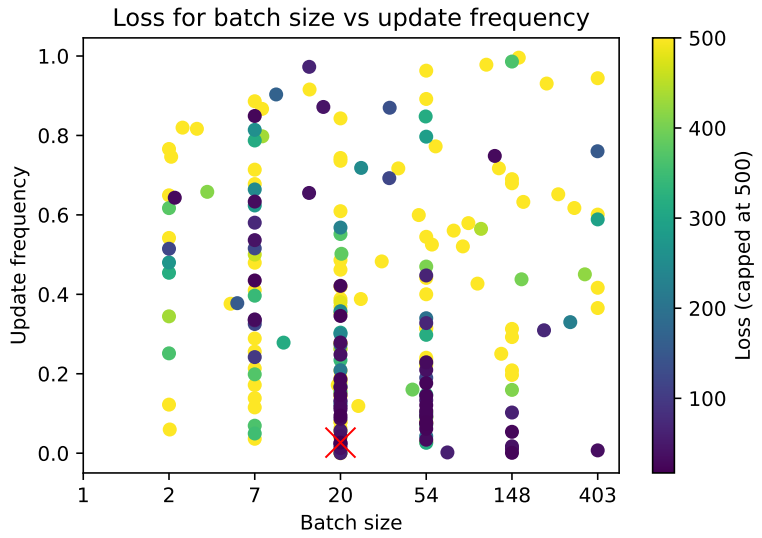


Figure F.3: Hyperparameters optimization loss for batch size and update frequency.

Figure F.4: Hyperparameters optimization loss for batch size and memory.

G

Snellius Job Script

```
# SLURM settings
#SBATCH --job-name=test_job          # Job name
#SBATCH --time=04:00:00              # Time limit [hrs:min:sec]
#SBATCH --nodes=1                    # Number of nodes
#SBATCH --ntasks=1                   # Number of tasks
#SBATCH --cpus-per-task=32           # Number of CPU cores per task
#SBATCH --output=output_%j.log       # Output log
#SBATCH --error=error_%j.log         # Error log

. ~/miniconda3/etc/profile.d/conda.sh
conda activate custom_env            # Virtual env with necessary packages

module load 2021

export OMP_NUM_TREADS=18             # Max threads in parallel region

set -euo pipefail                    # Stop script on error and undefined variables
```

Listing G.1: Snellius job script with Slurm settings

Additional Results - Water Level Objective

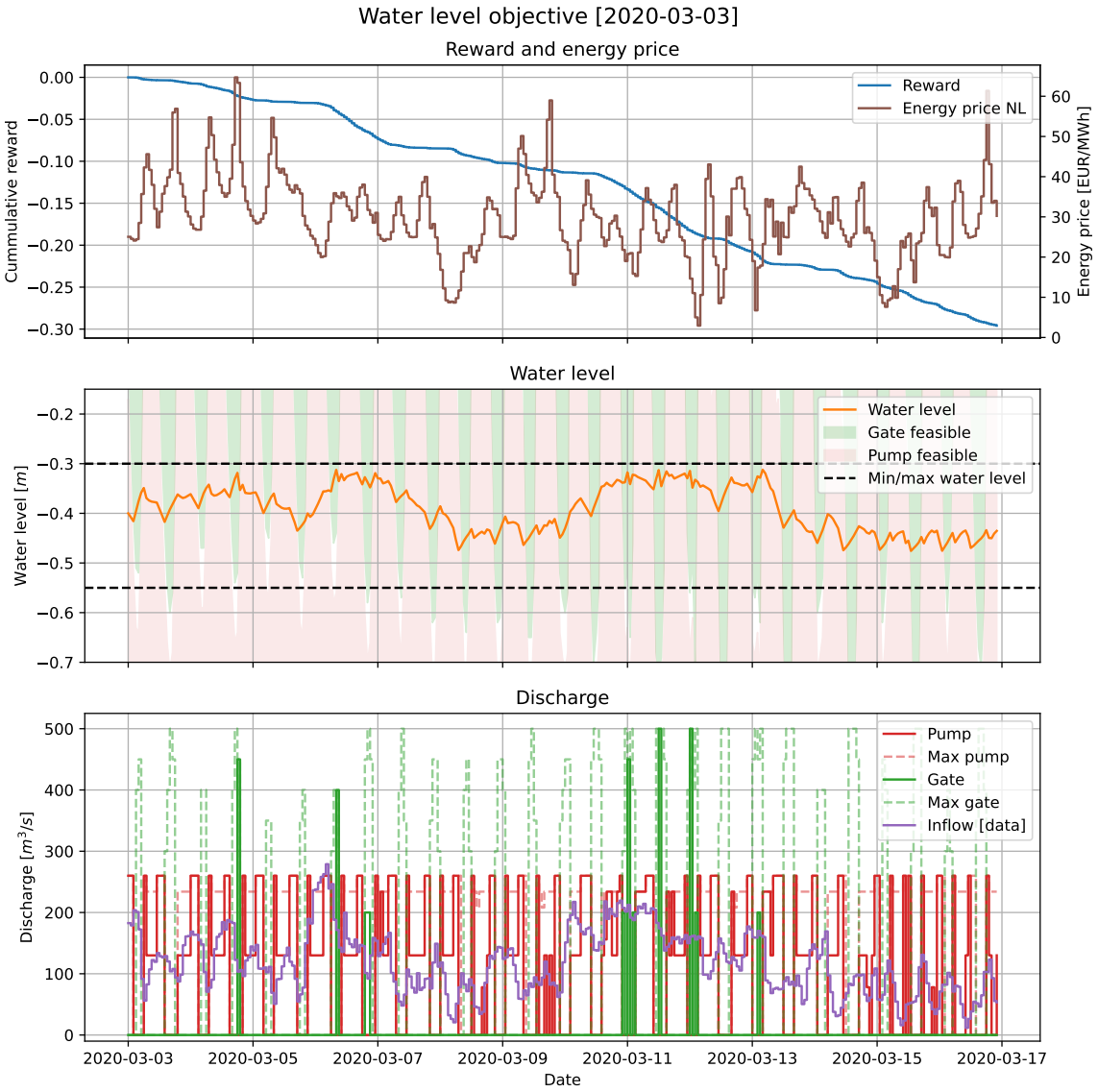


Figure H.1: The RL control plan for two weeks starting on 2020-03-03 with only the water level objective.

Additional Results - Cost Objective

I.1. Largest water level exceedance control plans

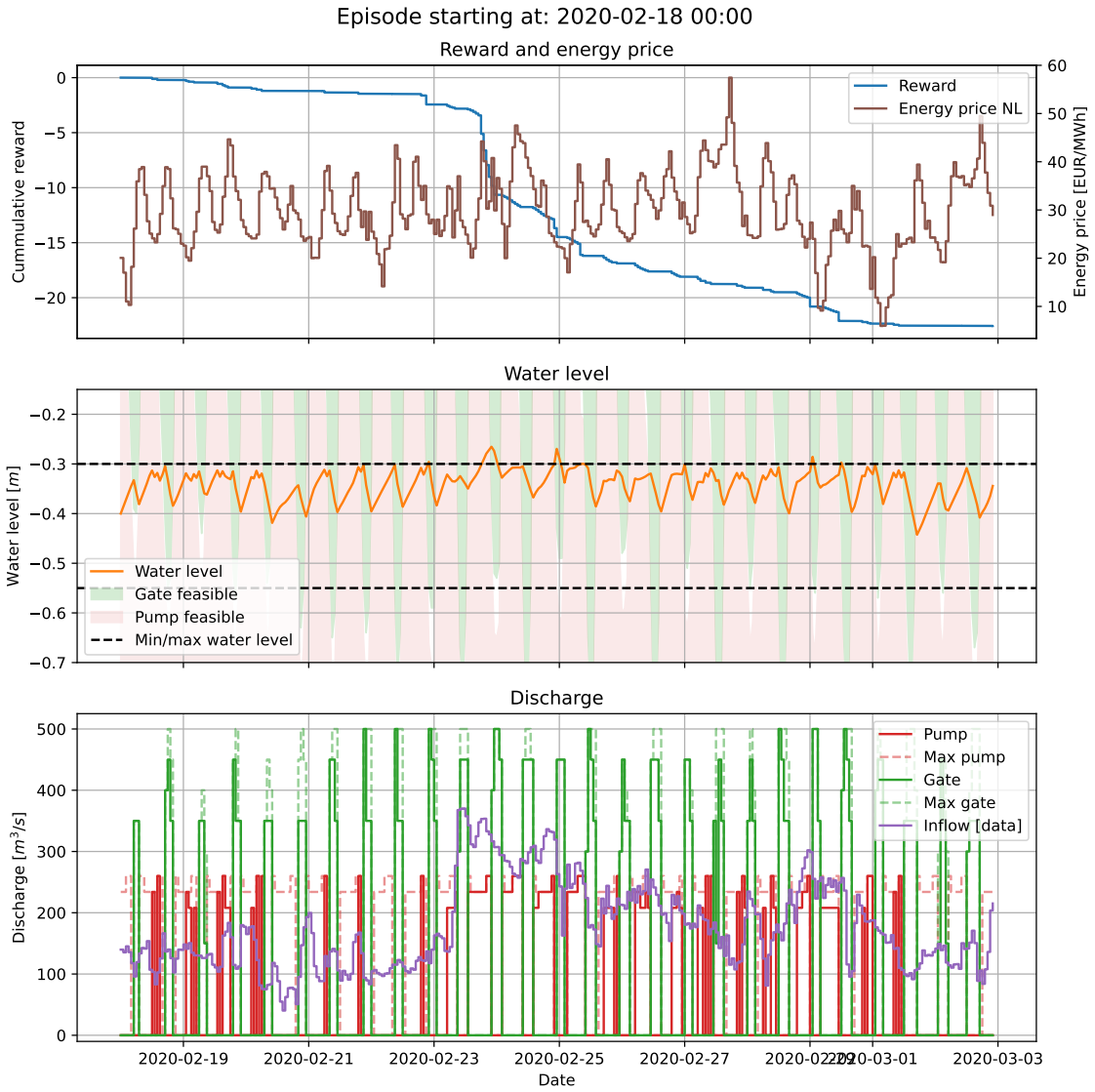


Figure I.1: The RL control plan for 14 days starting on 2020-02-18.

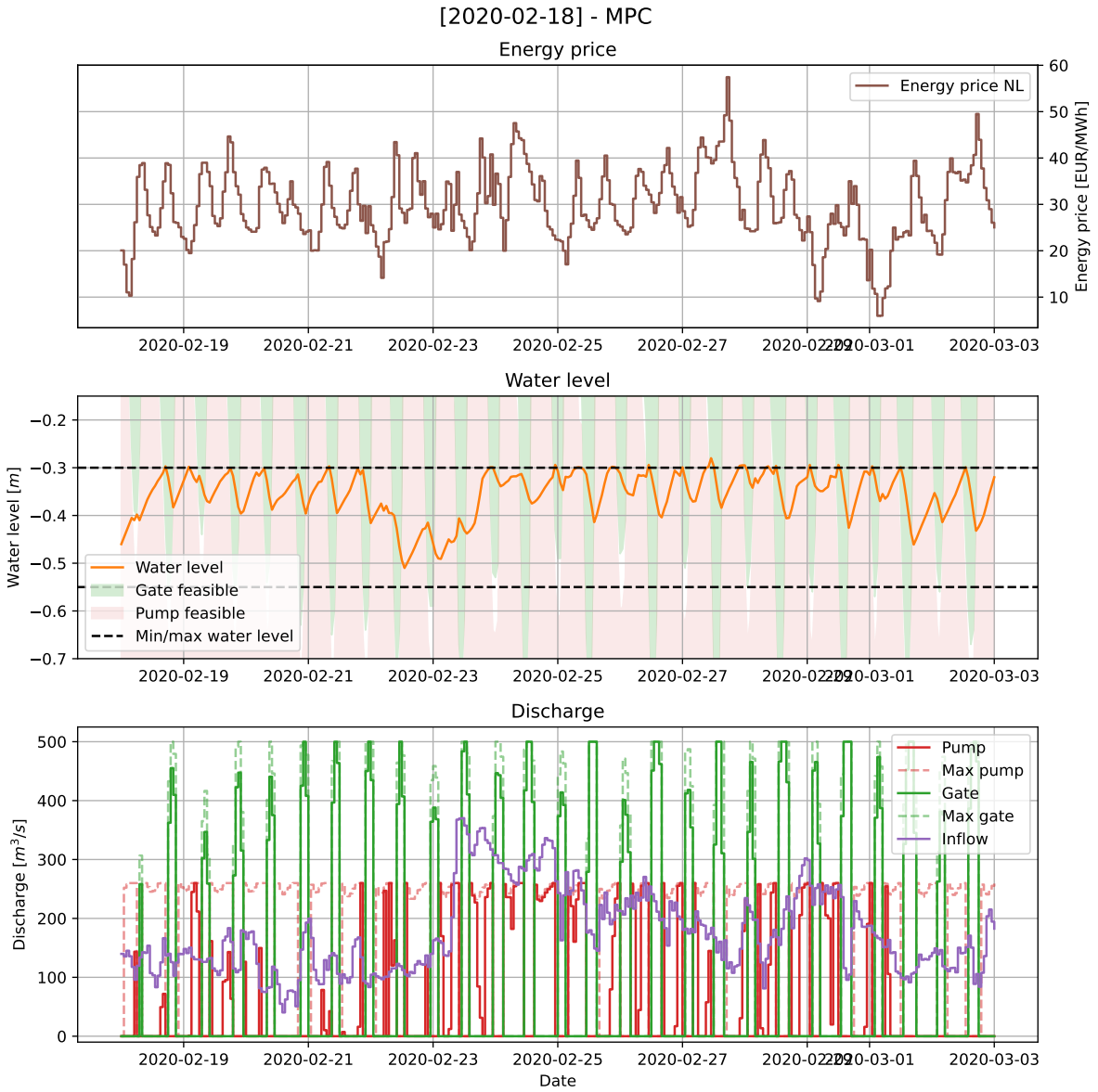


Figure I.2: The MPC control plan for 14 days starting on 2020-02-18.

I.2. MPC control plan - normal conditions

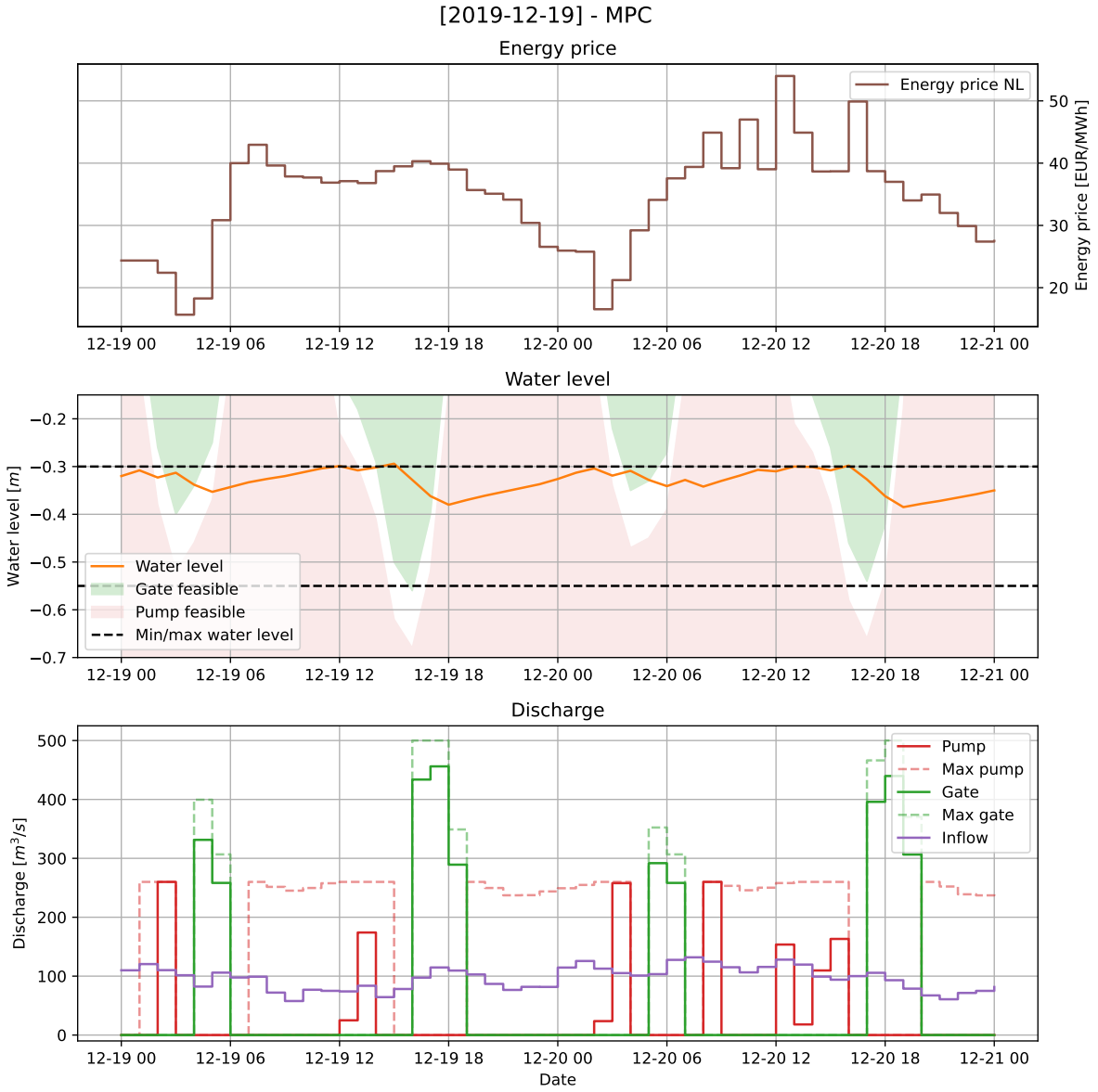


Figure I.3: The MPC control plan for two days starting on 2019-12-19.

I.3. MPC control plan - high inflow conditions

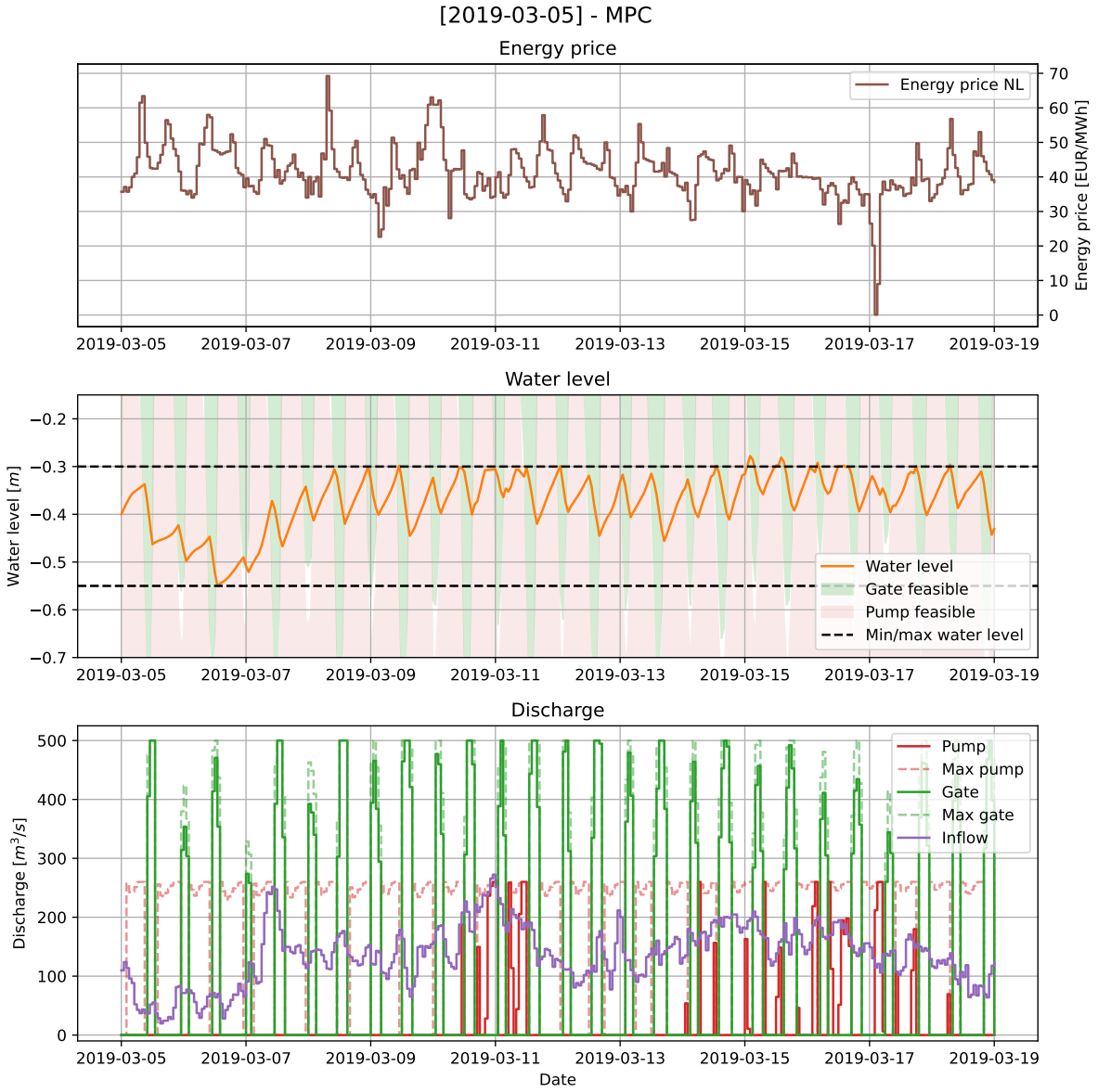


Figure I.4: The MPC control plan for two weeks starting on 2019-03-05.

J

Additional Results - Test Data Set Measurements

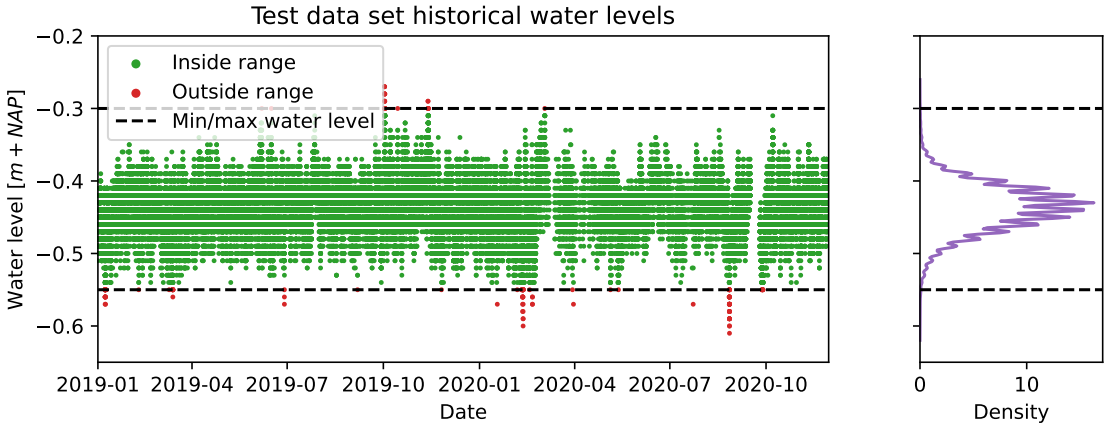


Figure J.1: The historical measurements of the water level in the NZK at IJmuiden for the entire test data set.

K

Additional Results - Test Scenarios

K.1. Extreme Q

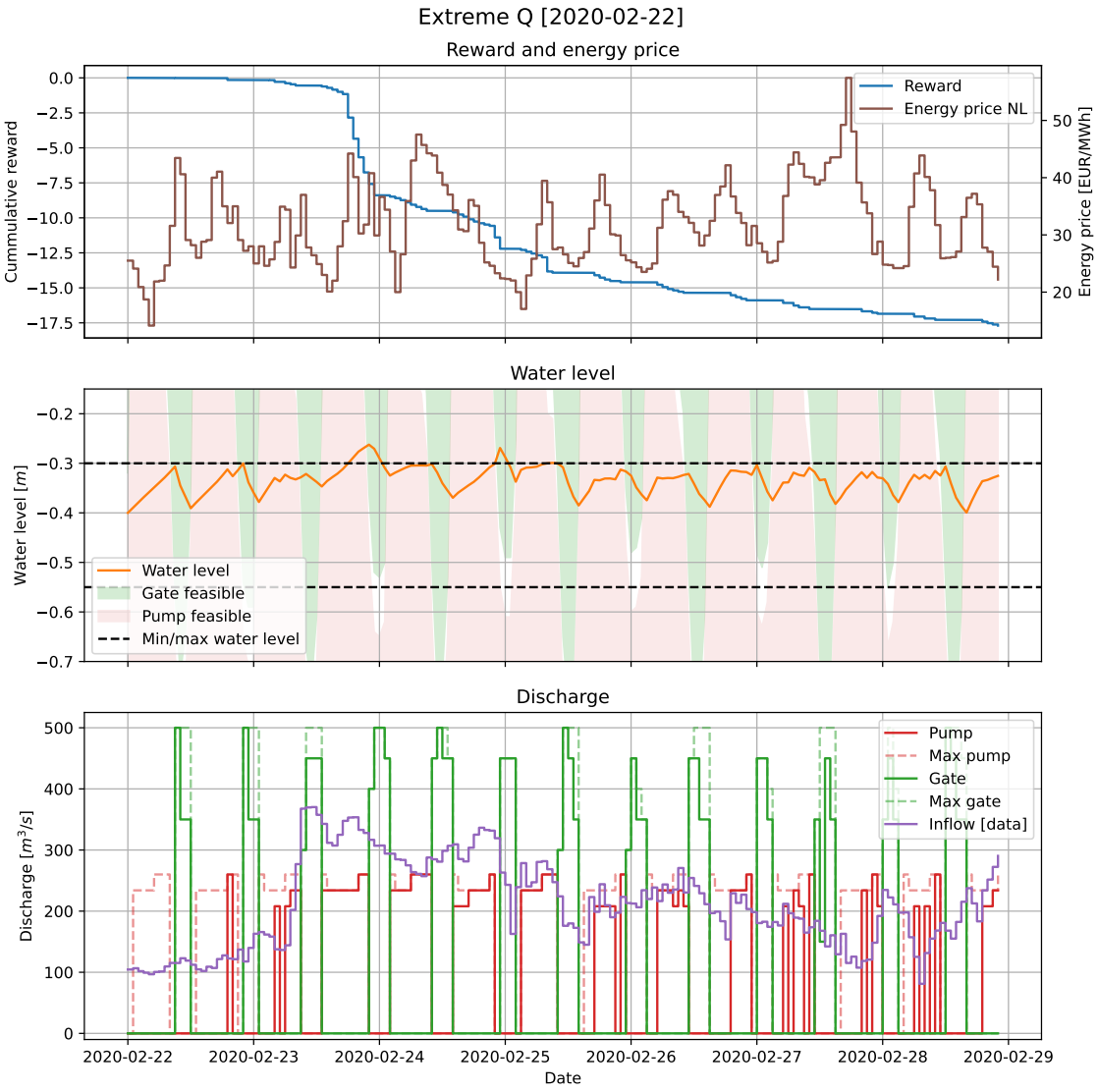


Figure K.1: The RL control plan for the extreme discharge scenario.

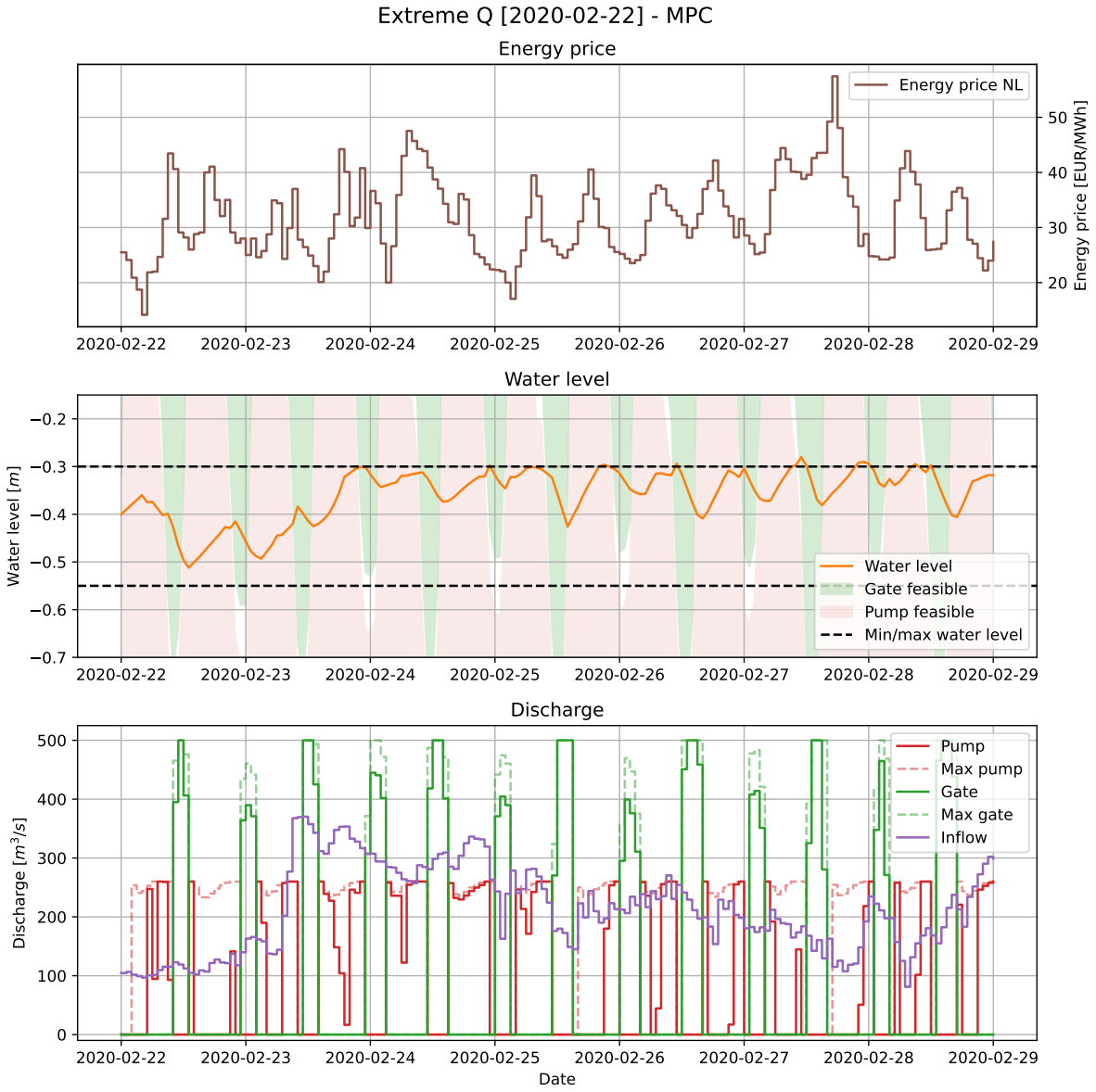


Figure K.2: The MPC control plan for the extreme discharge scenario.

K.2. High Q

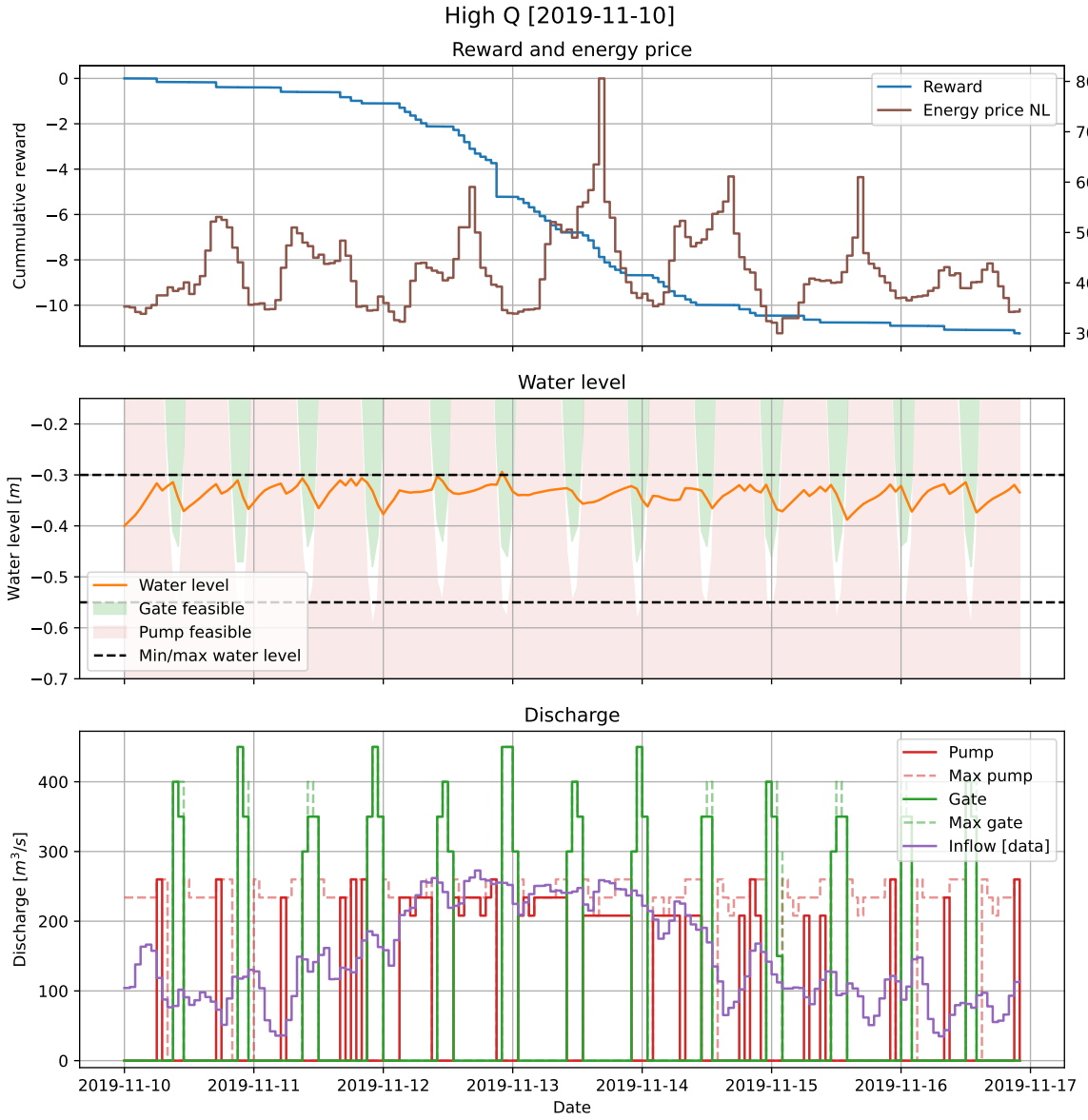


Figure K.3: The RL control plan for the high discharge scenario.

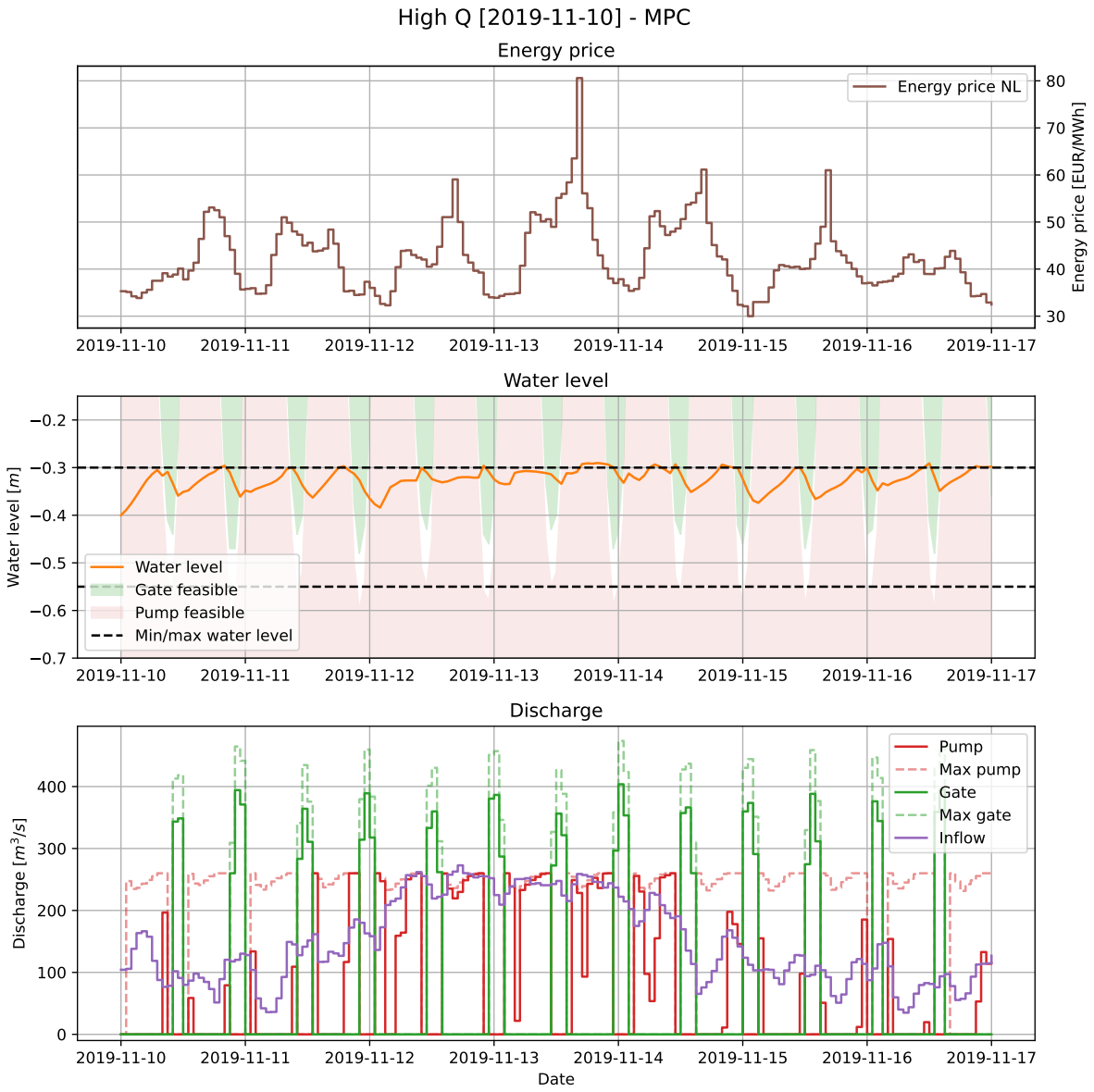


Figure K.4: The MPC control plan for the high discharge scenario.

K.3. Low Q

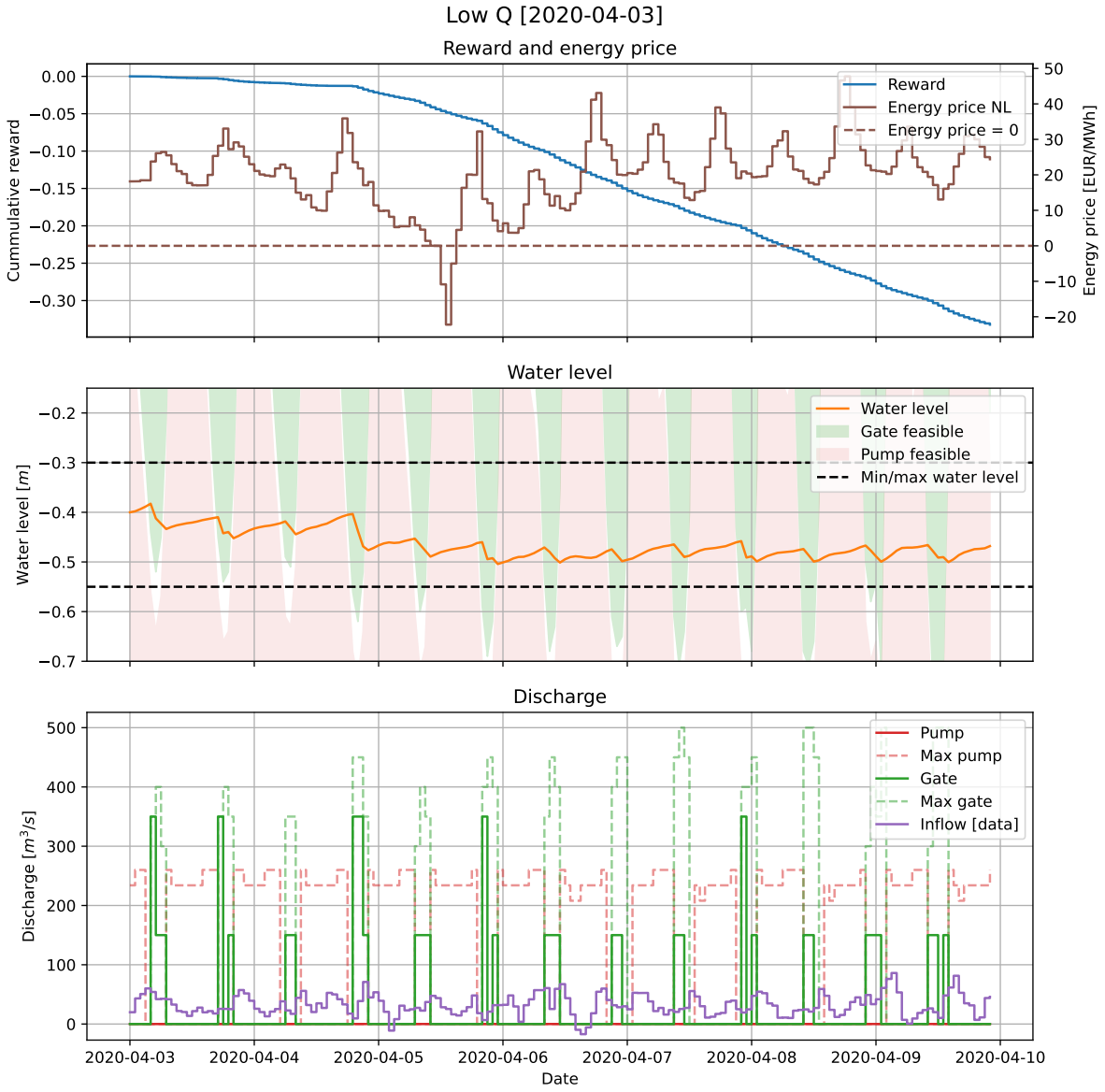


Figure K.5: The RL control plan for the low discharge scenario.

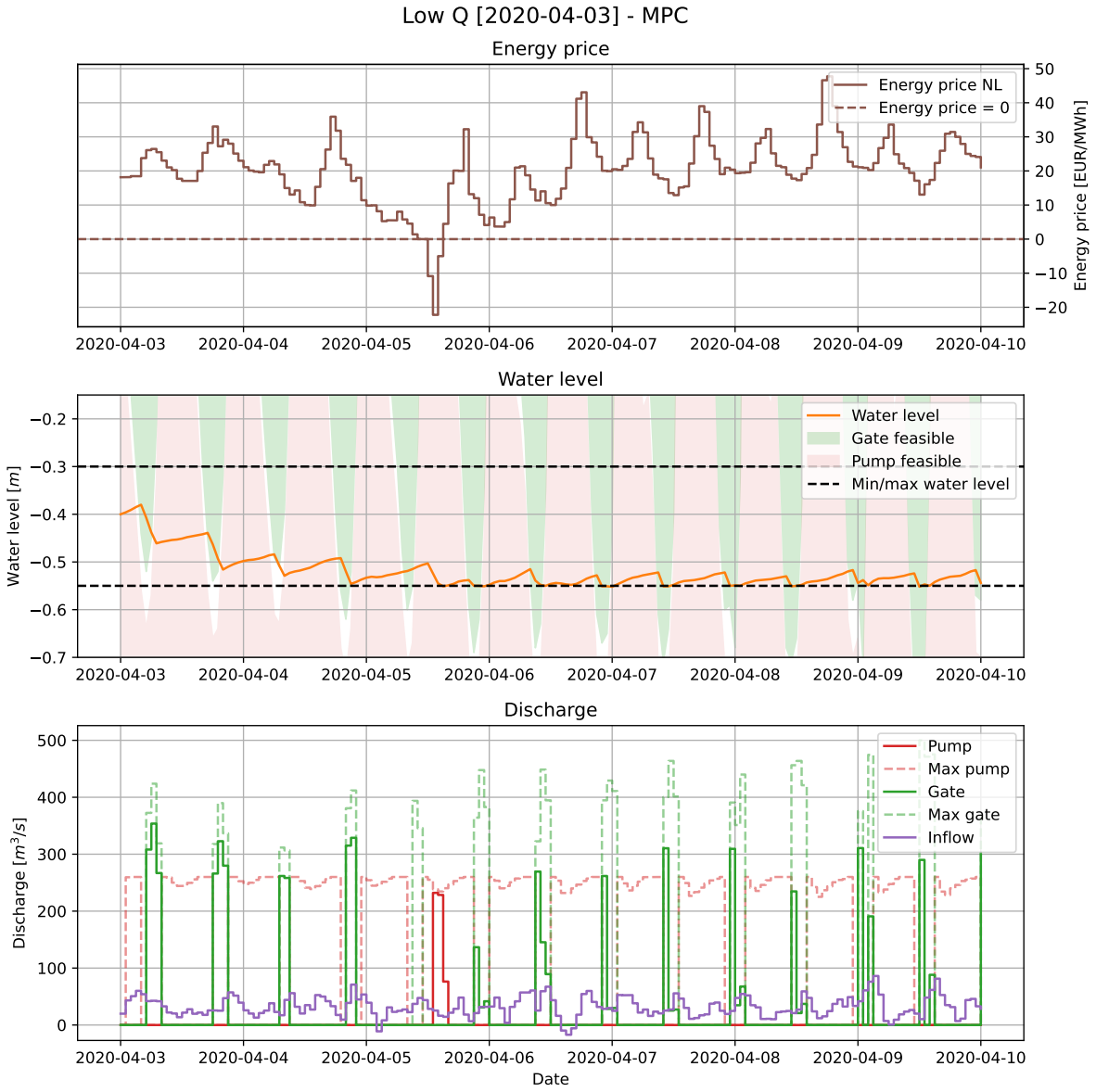


Figure K.6: The MPC control plan for the low discharge scenario.

K.4. High sea

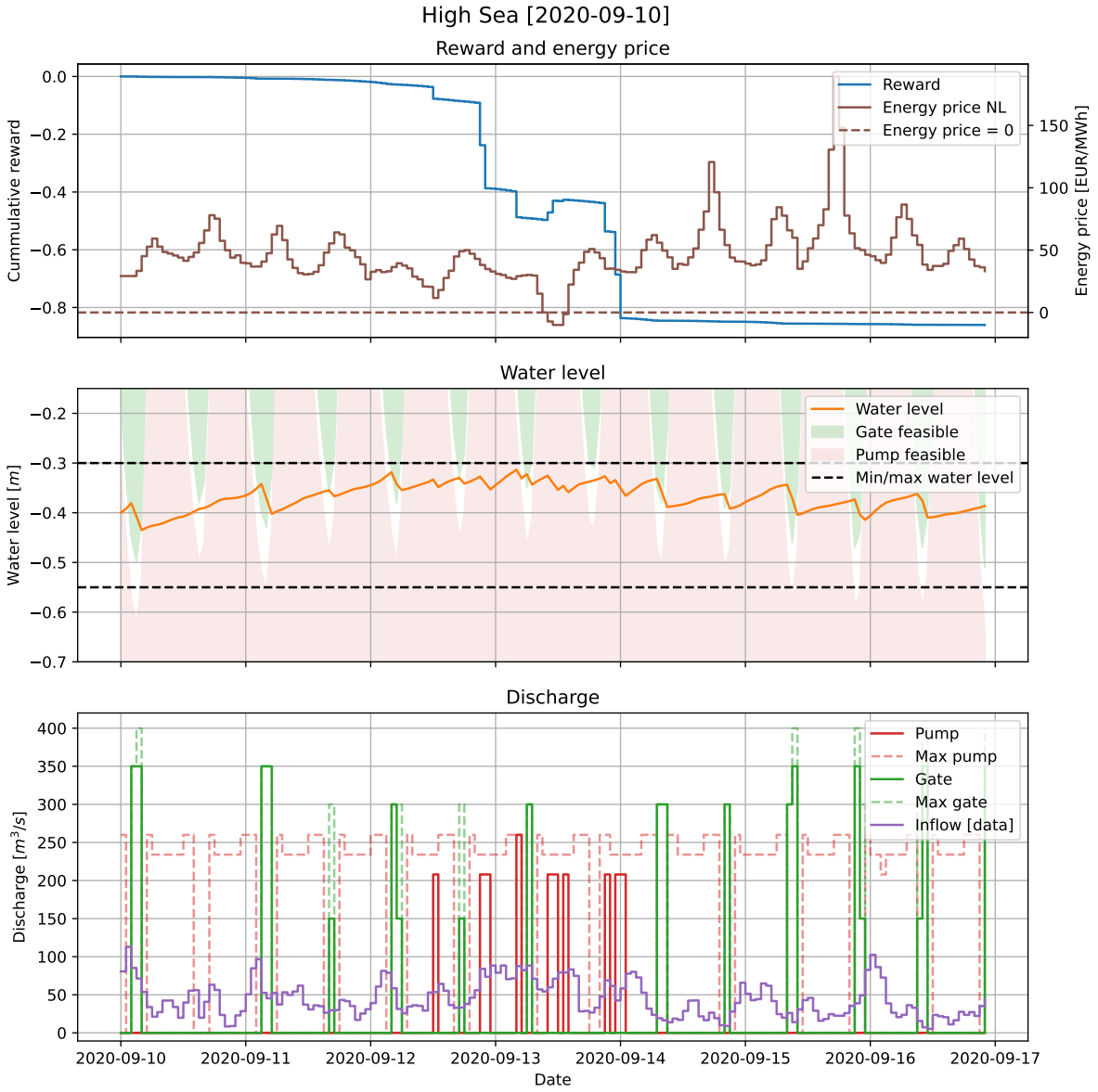


Figure K.7: The RL control plan for the high sea level scenario.

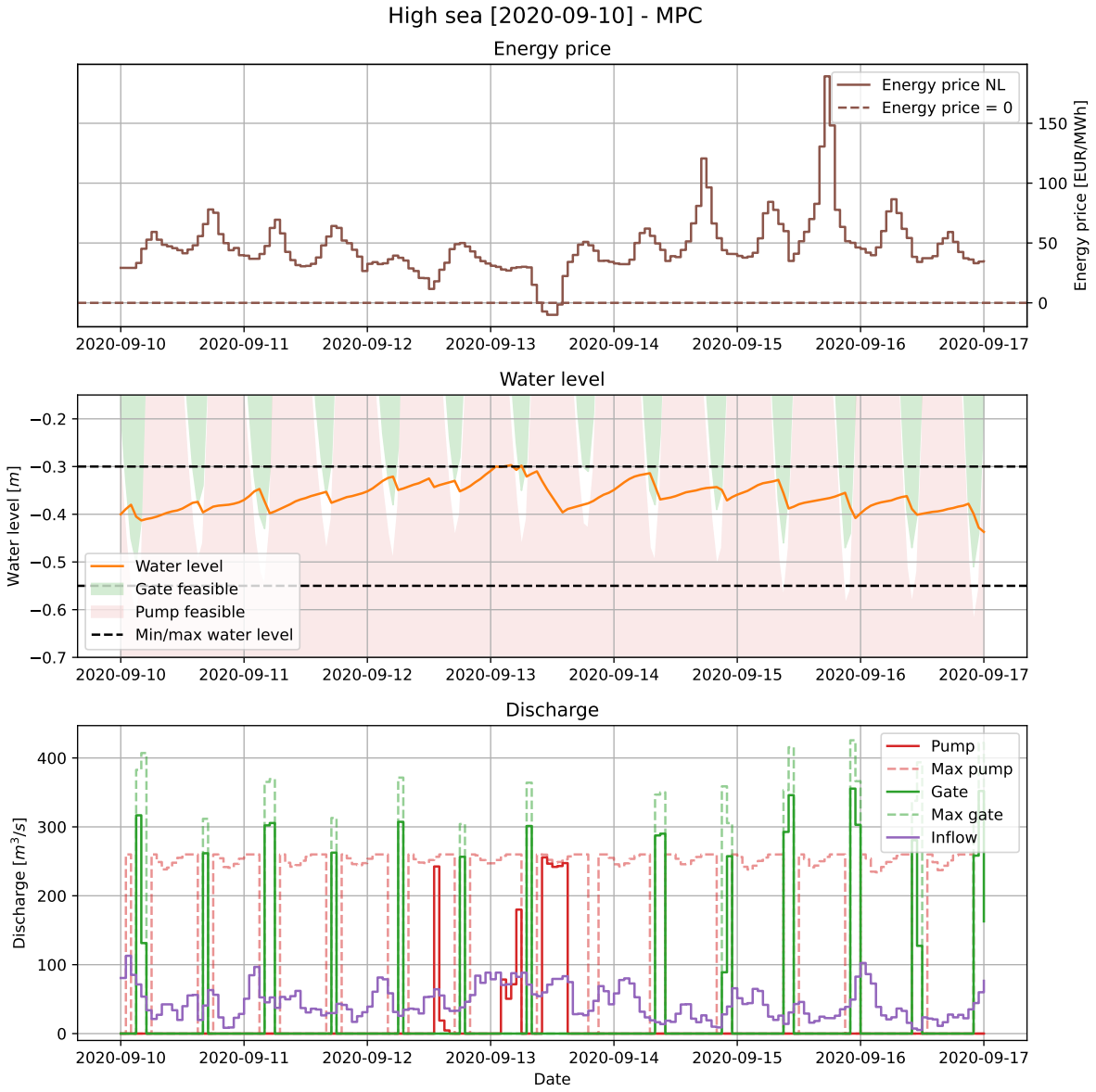


Figure K.8: The MPC control plan for the high sea level scenario.

K.5. High sea high Q

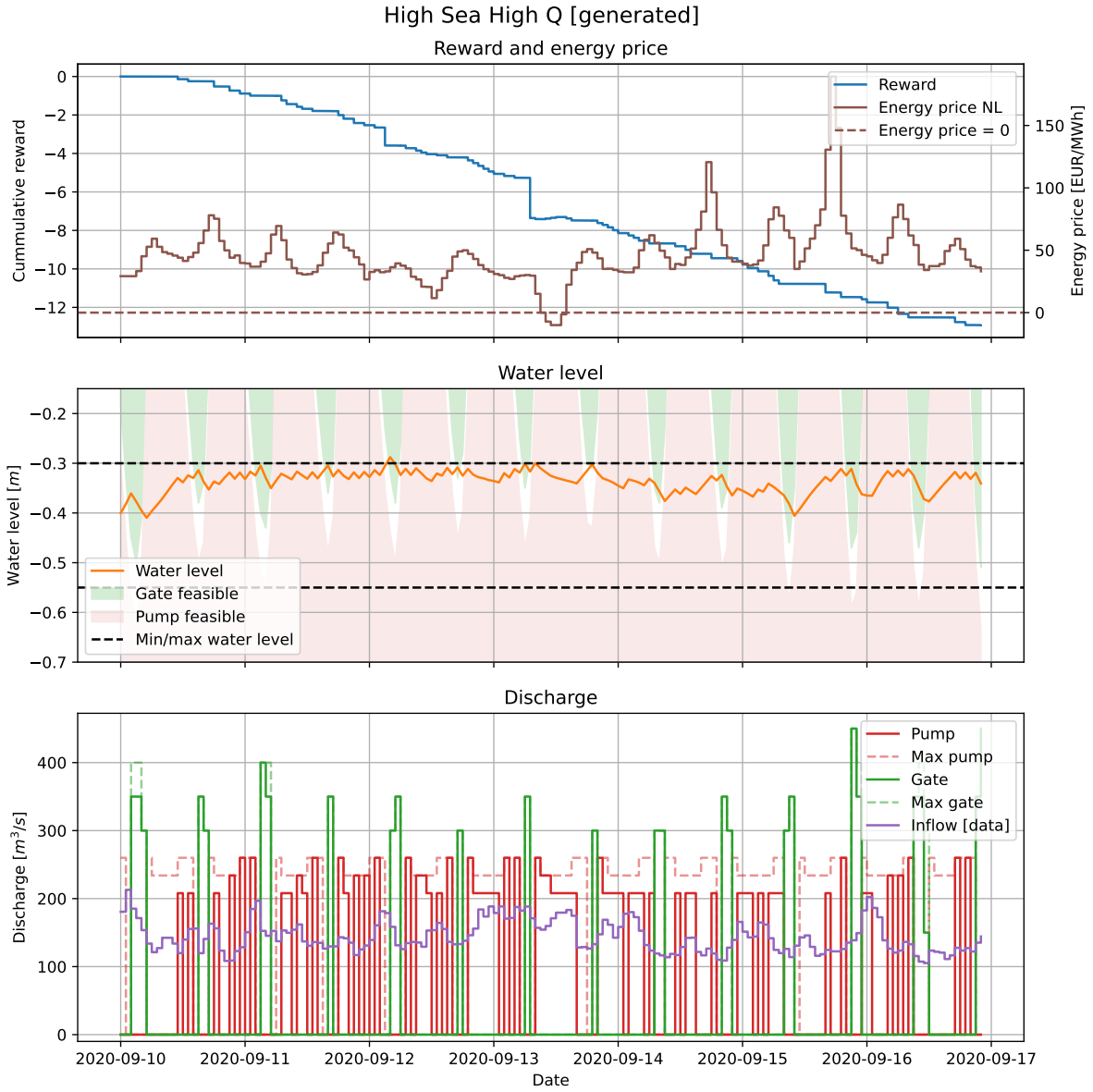


Figure K.9: The RL control plan for the high sea level and high discharge scenario.

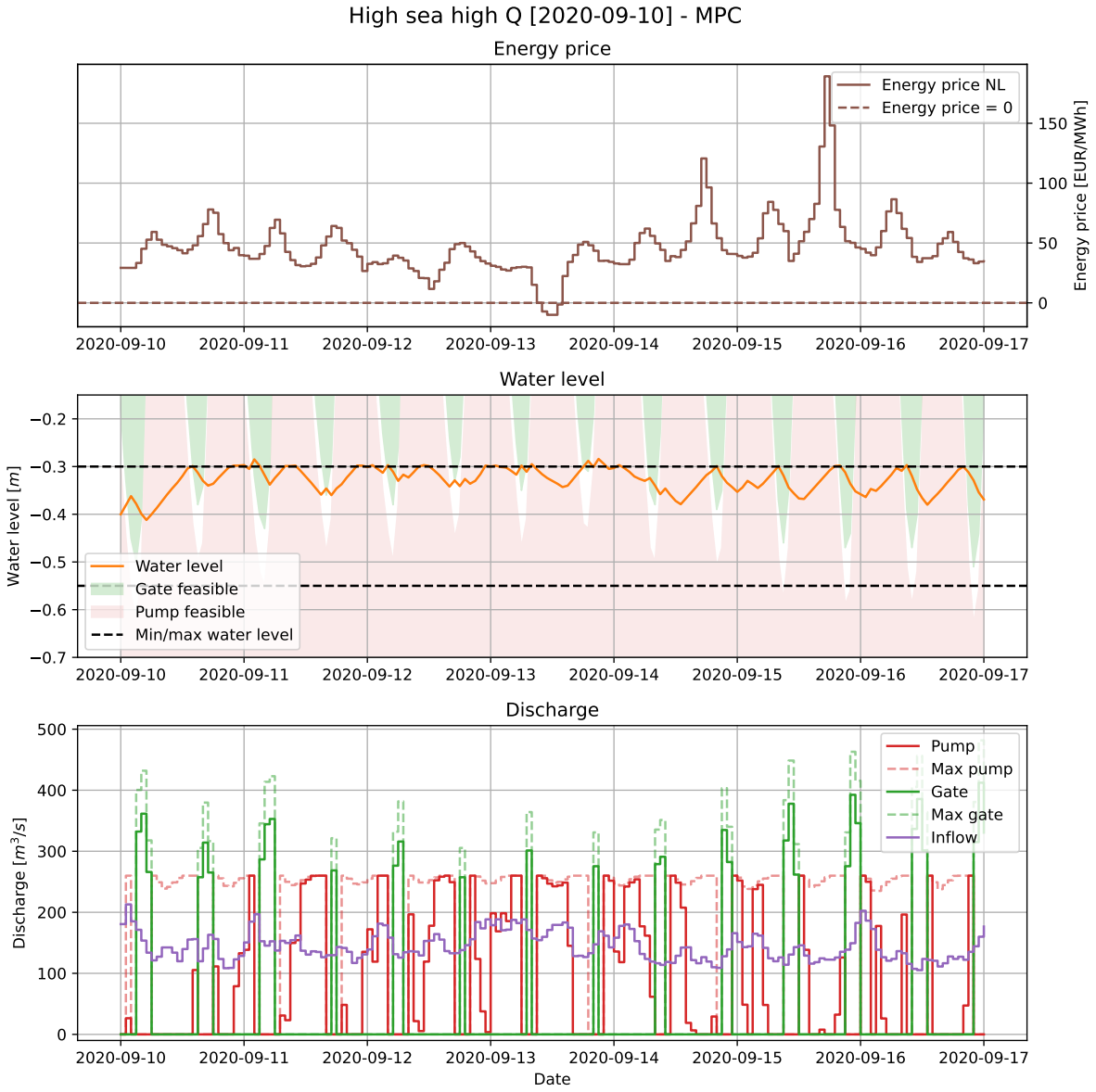


Figure K.10: The MPC control plan for the high sea level and high discharge scenario.

K.6. High E

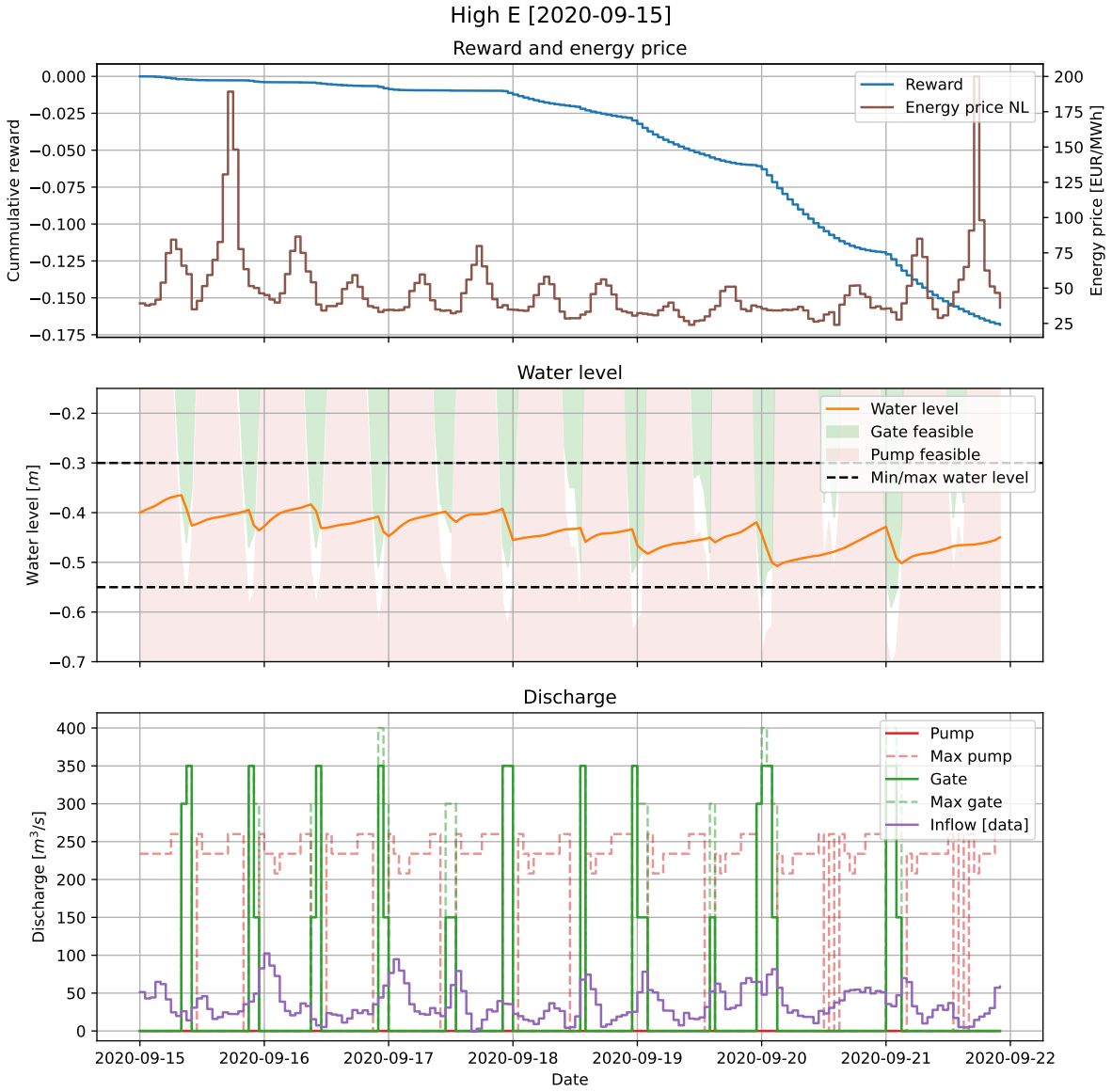


Figure K.11: The RL control plan for the high electricity price scenario.

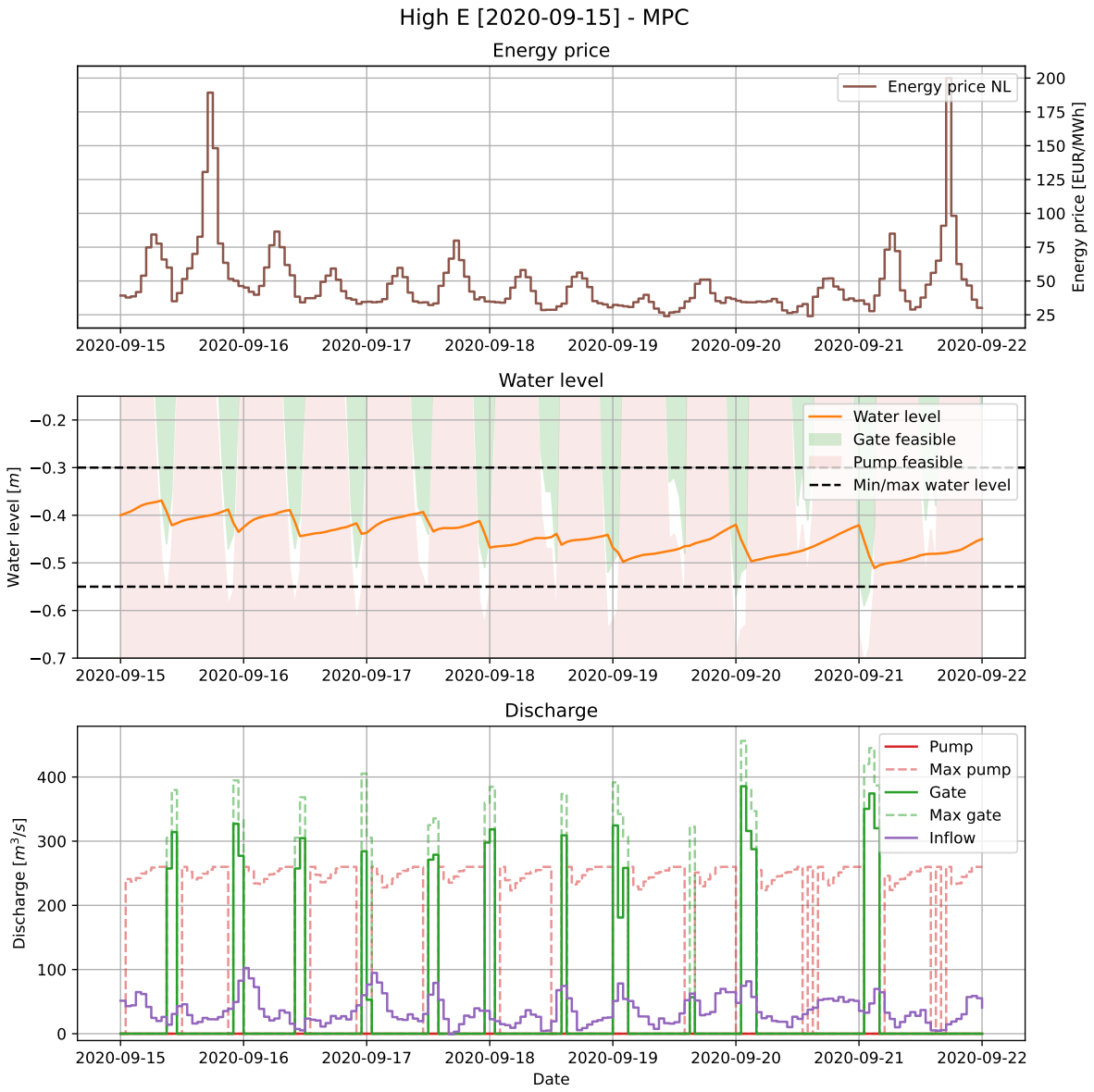


Figure K.12: The MPC control plan for the high electricity price scenario.

K.7. Negative E

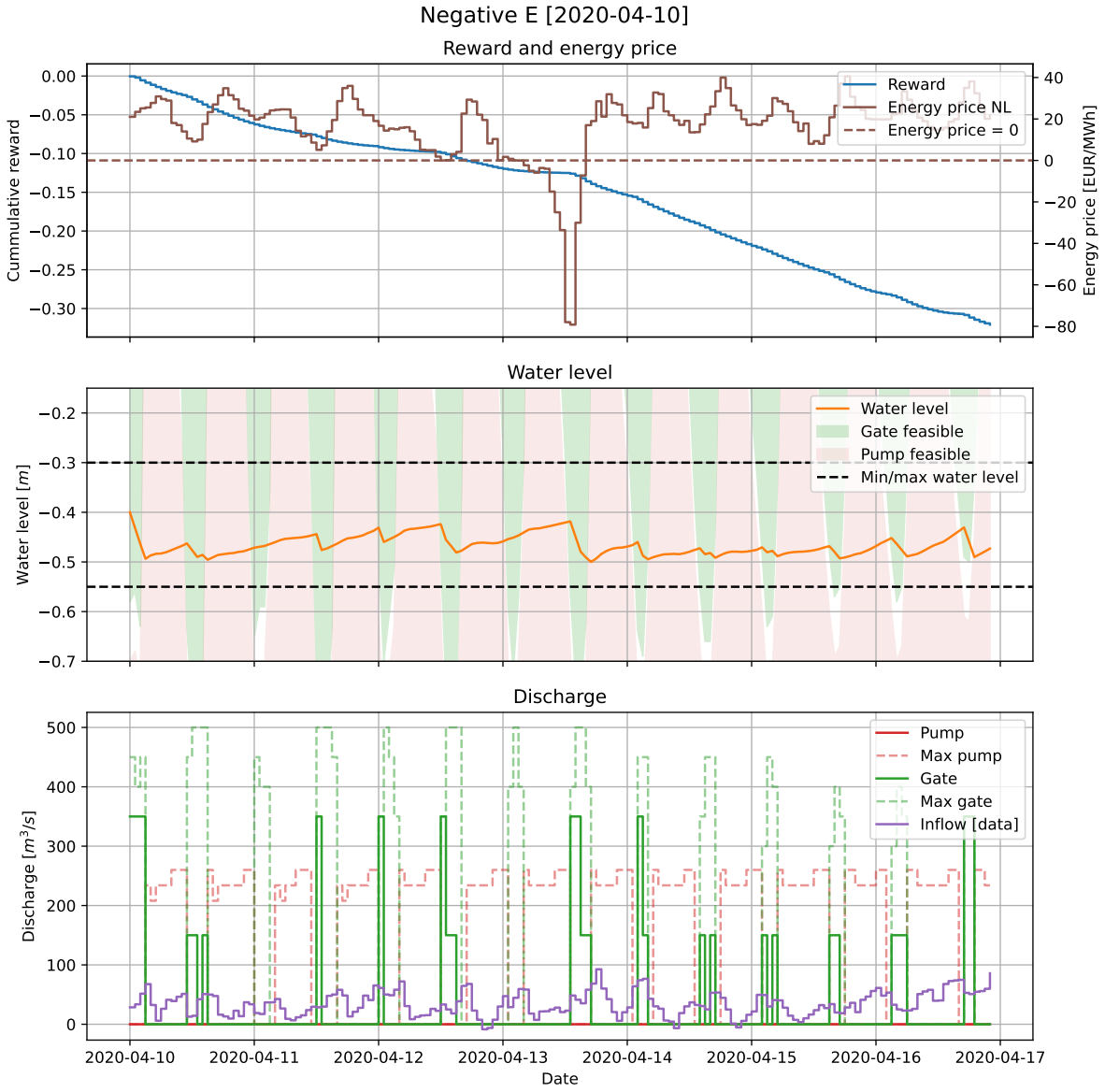


Figure K.13: The RL control plan for the negative electricity price scenario.

K.8. Extreme negative E

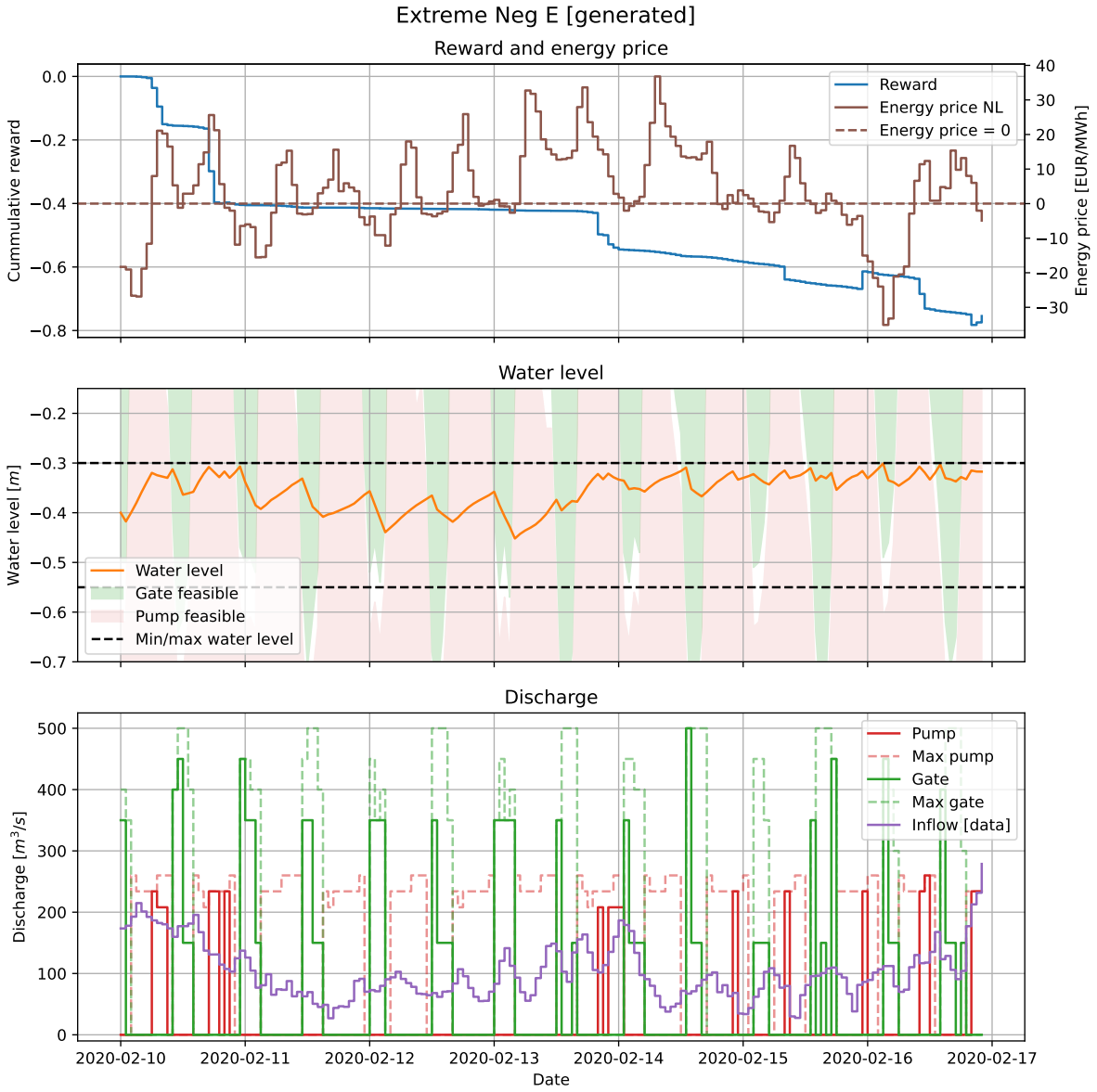


Figure K.14: The RL control plan for the extreme negative electricity price scenario.

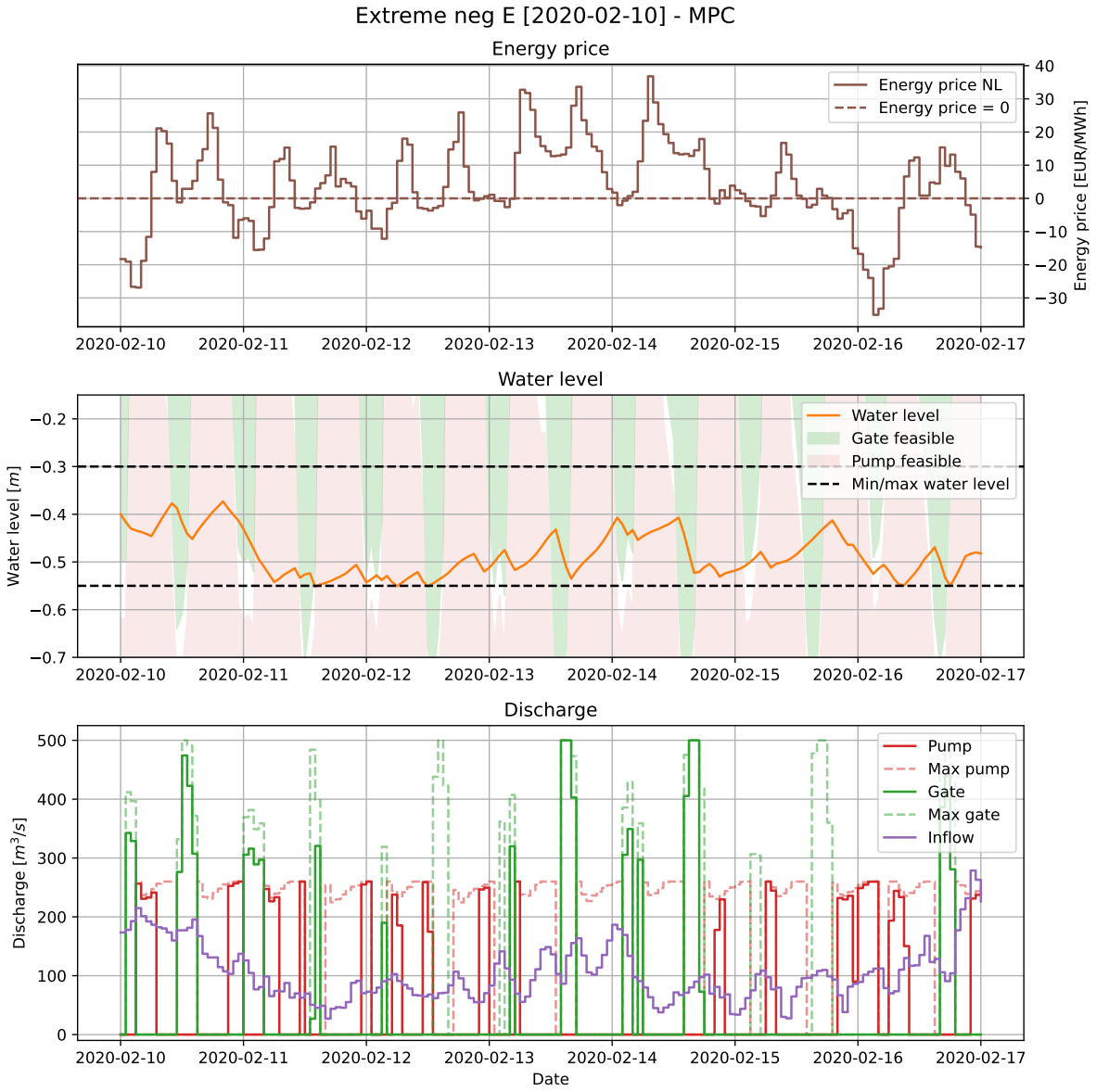


Figure K.15: The MPC control plan for the extreme negative electricity price scenario.

Additional Results - Suitable RL Algorithms

L.1. Control plans for 2020-02-18

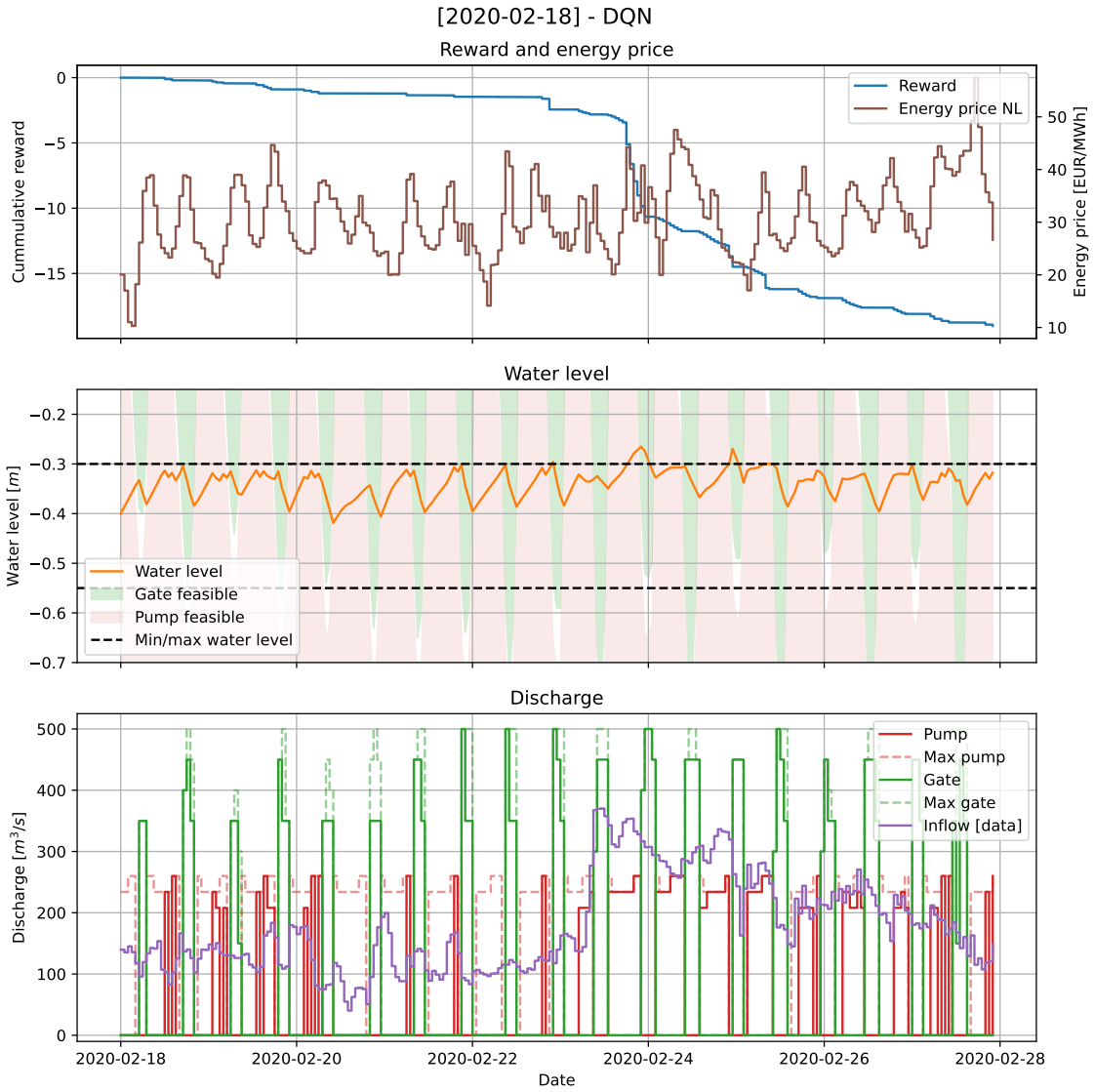


Figure L.1: The RL control plan using DQN for 10 days starting on 2020-02-18.

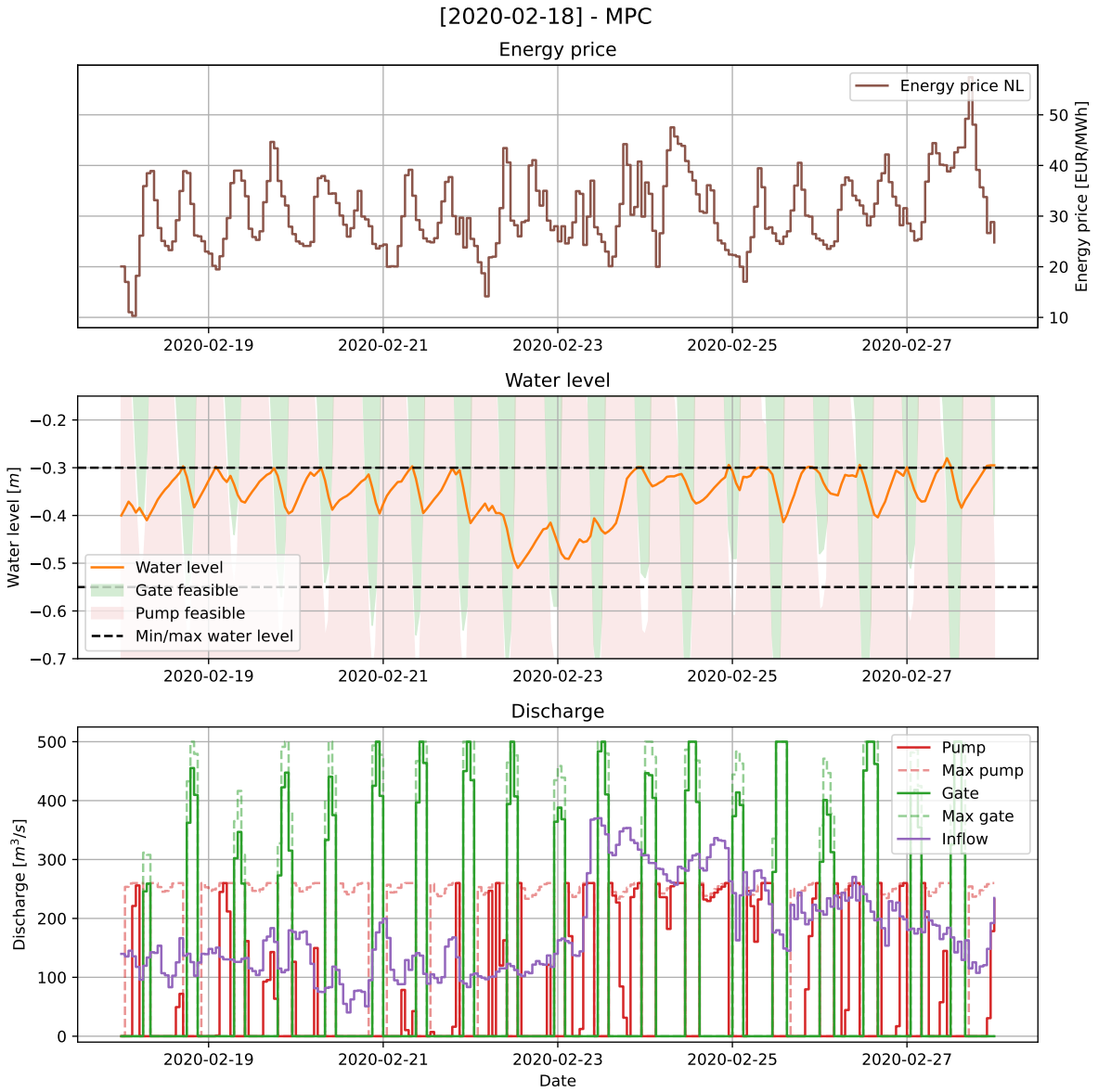


Figure L.2: The MPC control plan for 10 days starting on 2020-02-18.

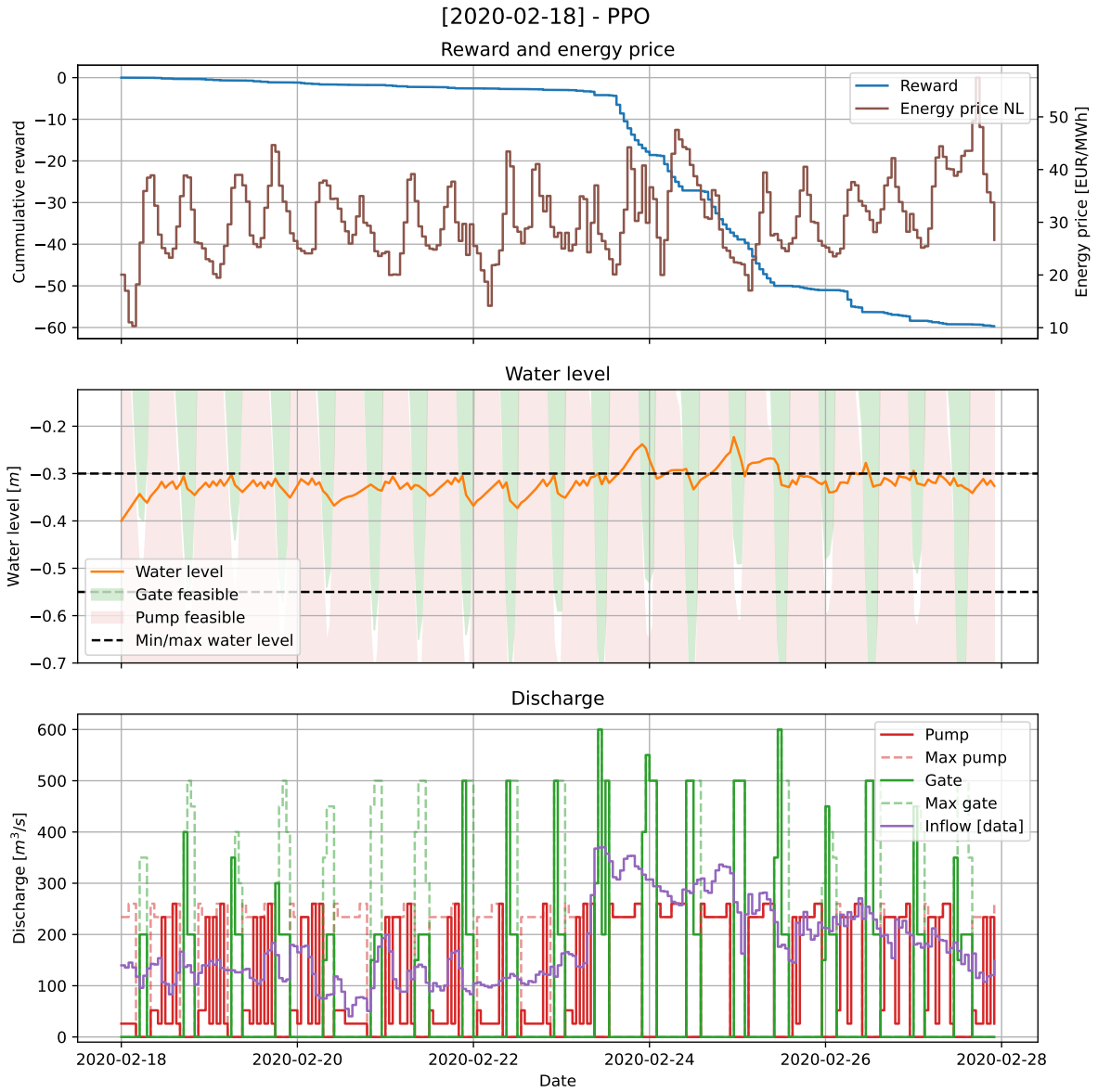


Figure L.3: The RL control plan using PPO for 10 days starting on 2020-02-18.

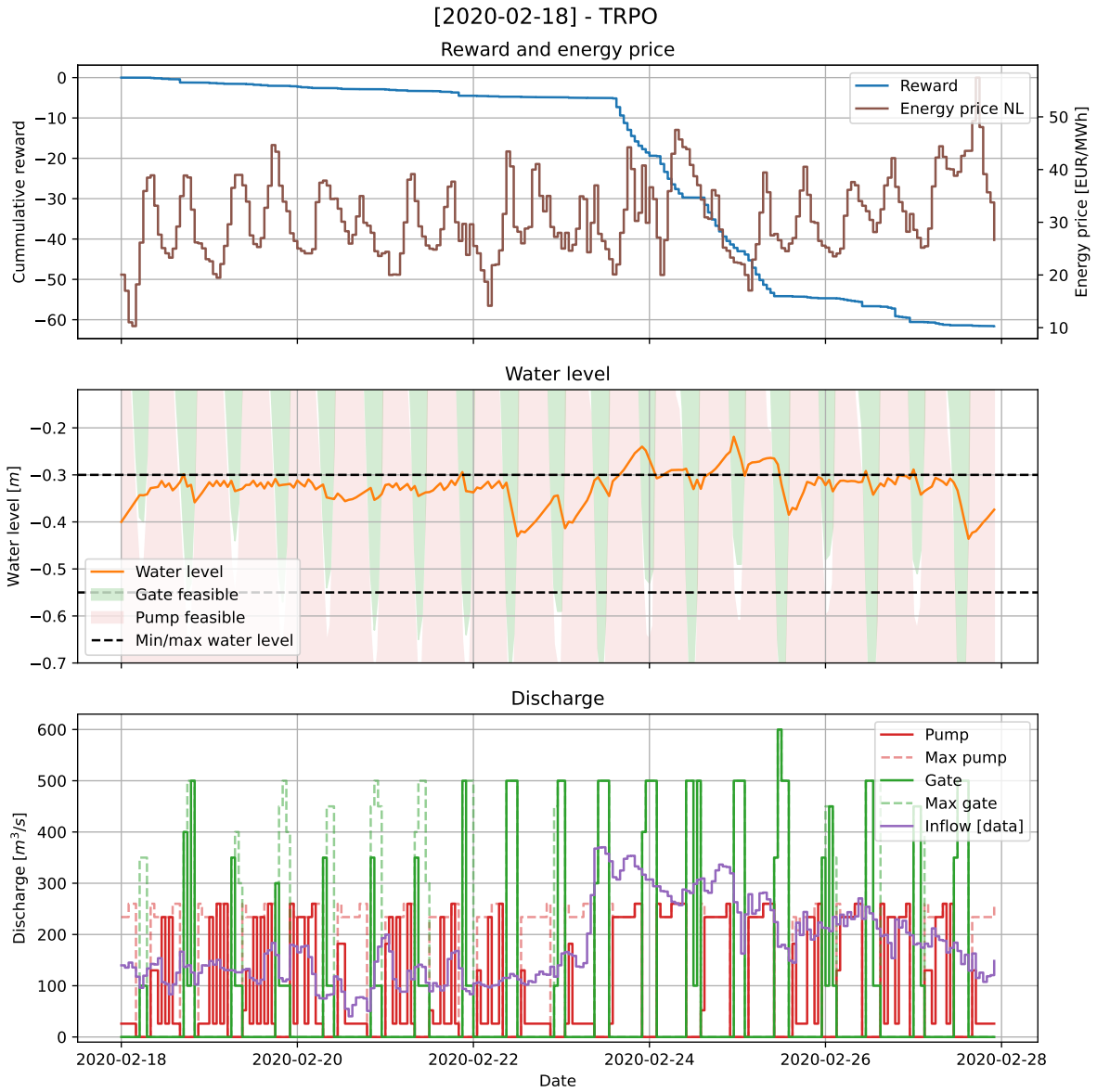


Figure L.4: The RL control plan using TRPO for 10 days starting on 2020-02-18.

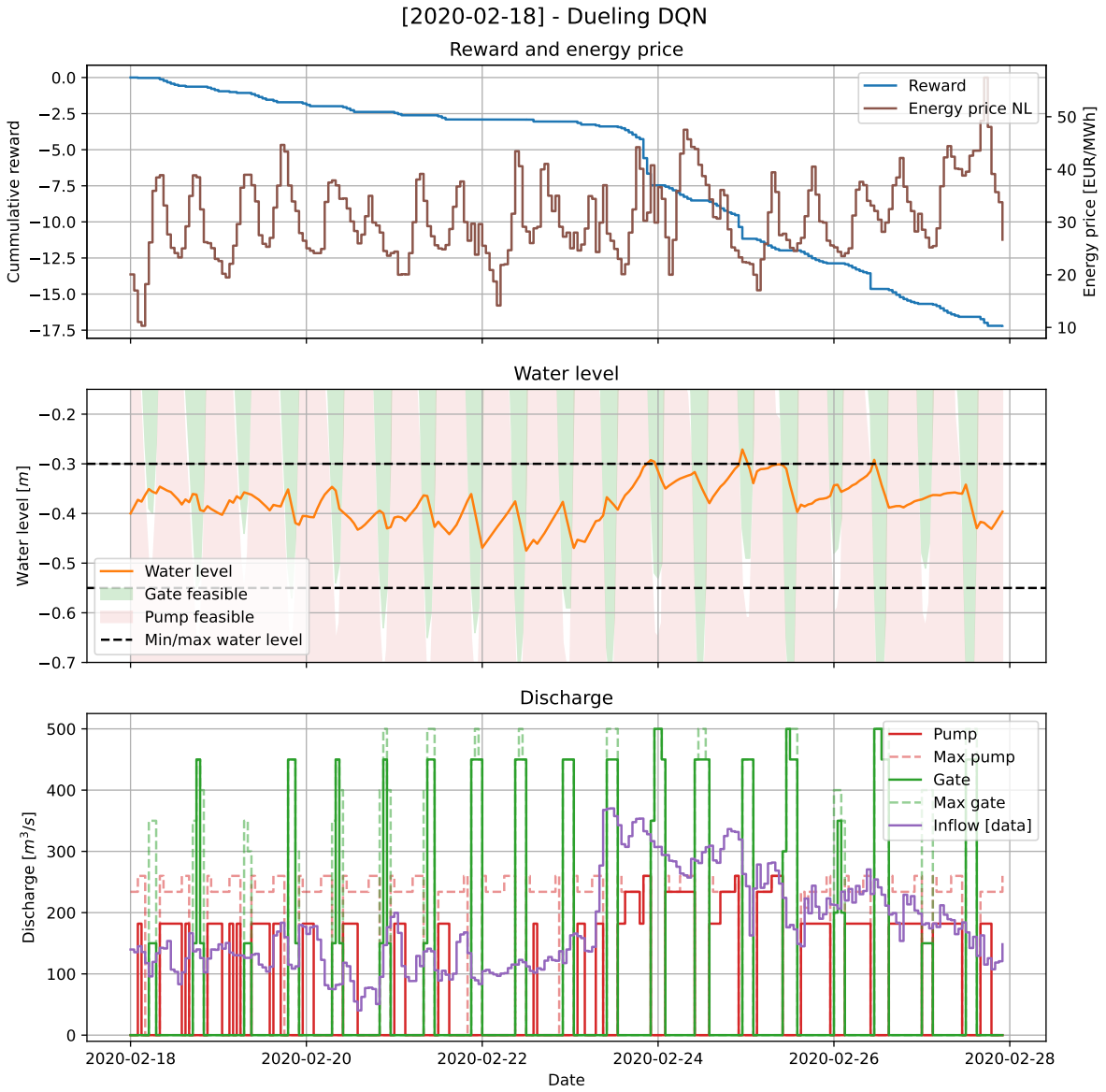


Figure L.5: The RL control plan using Dueling DQN for 10 days starting on 2020-02-18.

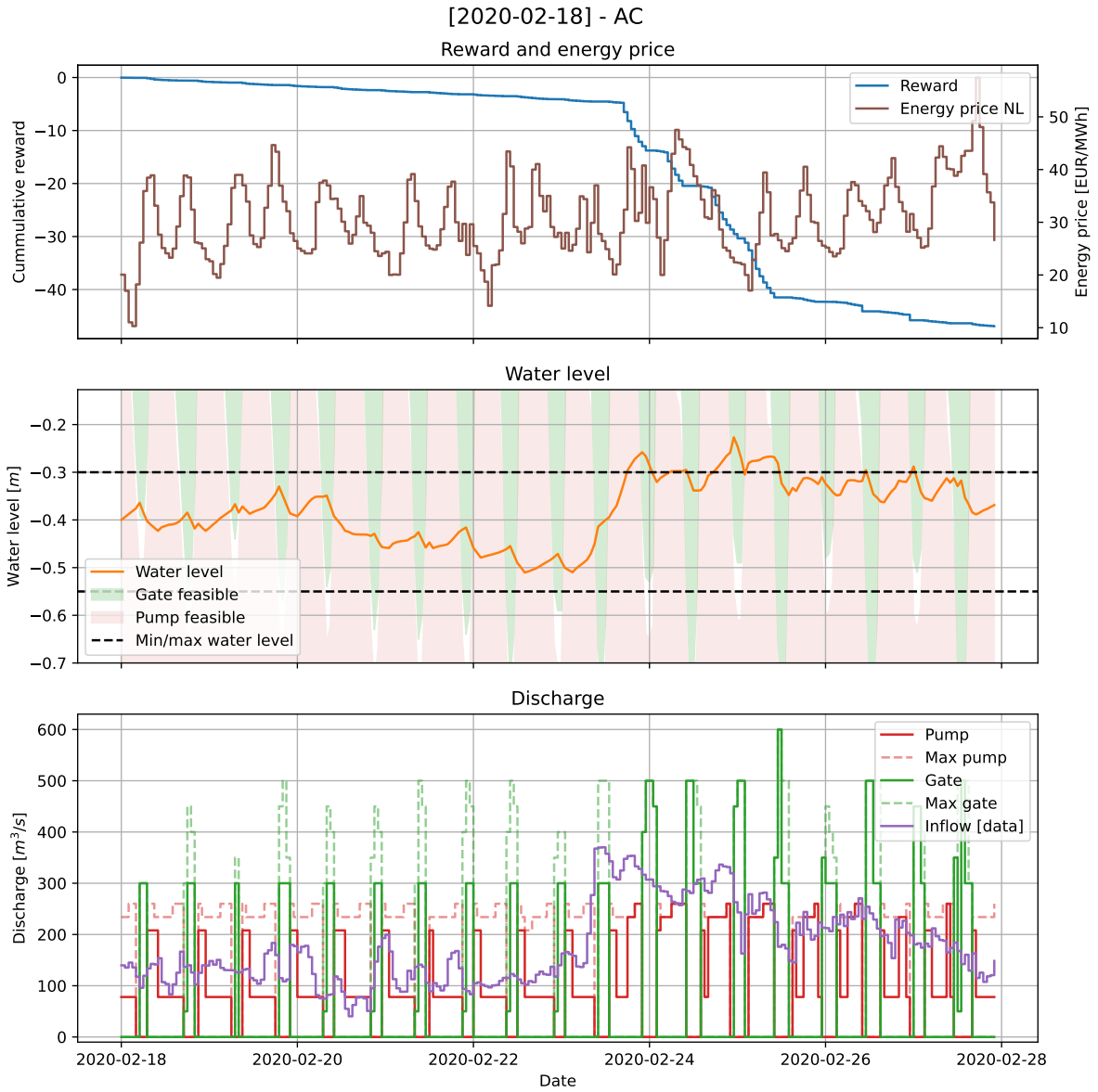


Figure L.6: The RL control plan using AC for 10 days starting on 2020-02-18.

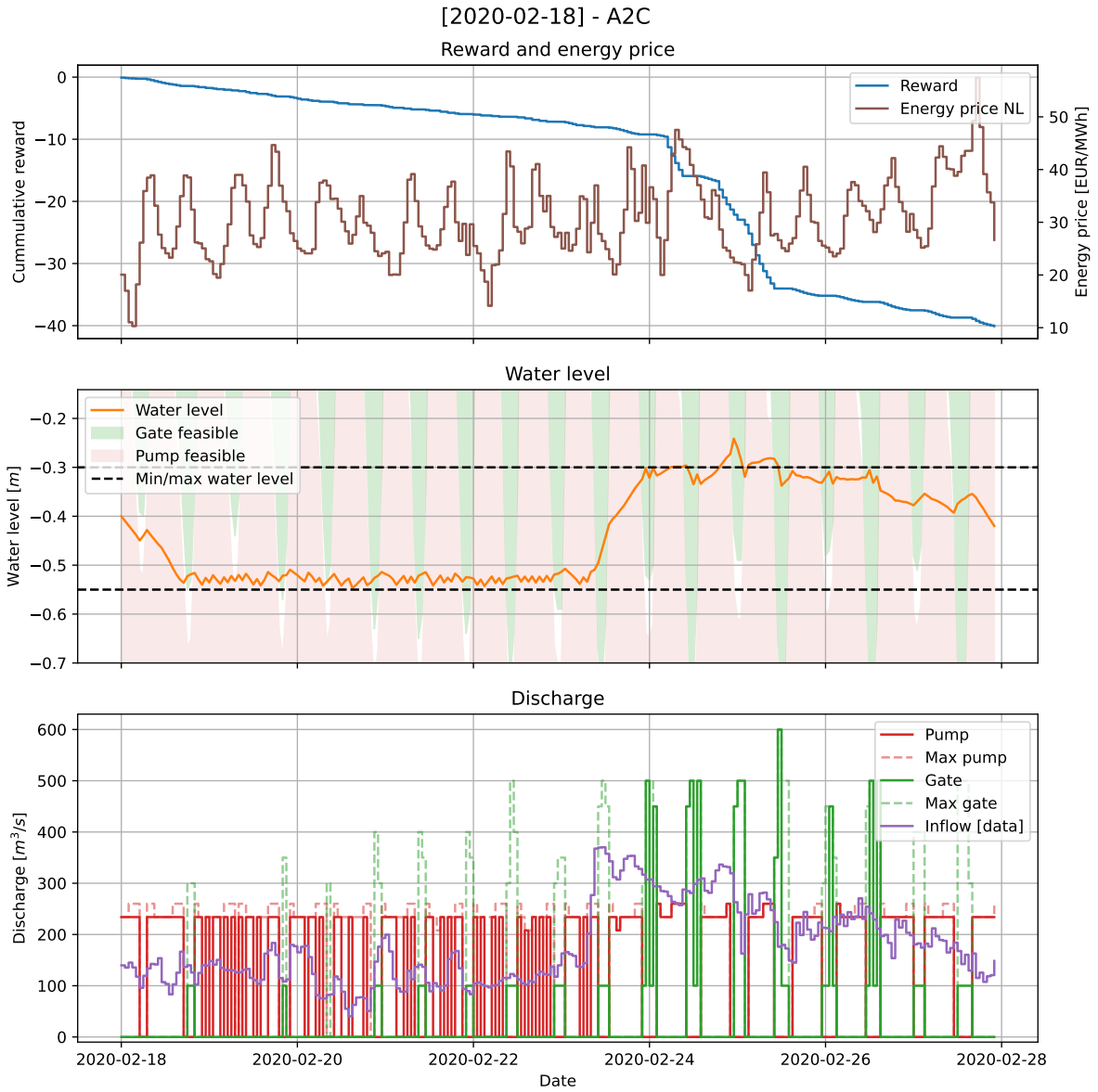


Figure L.7: The RL control plan using A2C for 10 days starting on 2020-02-18.

M

Additional Results - Alternative Reward

M.1. High water level - 2019-12-17

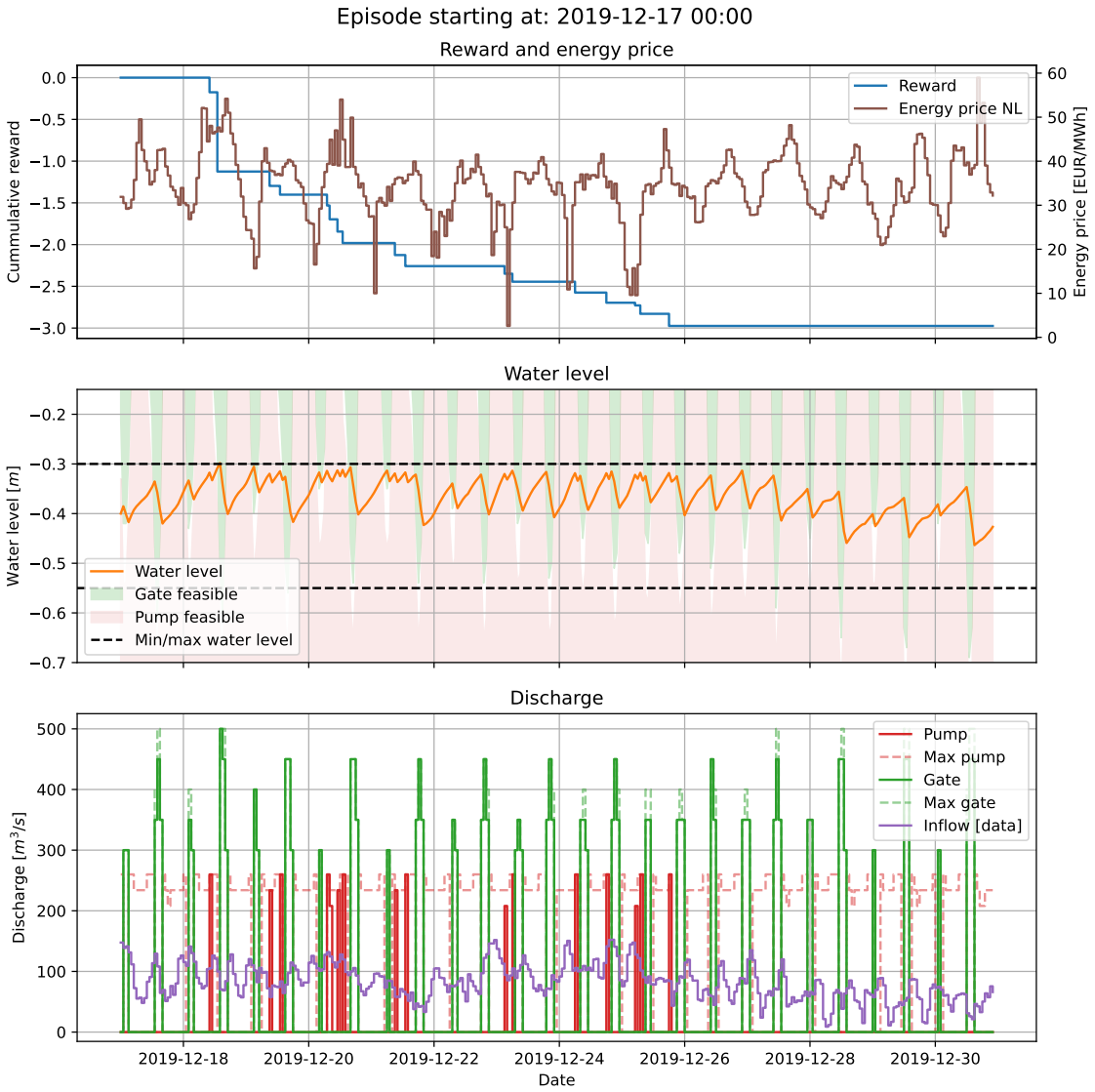


Figure M.1: The RL alternative reward control plan for two weeks starting on 2019-12-17.

M.2. High inflow discharge - 2019-03-05

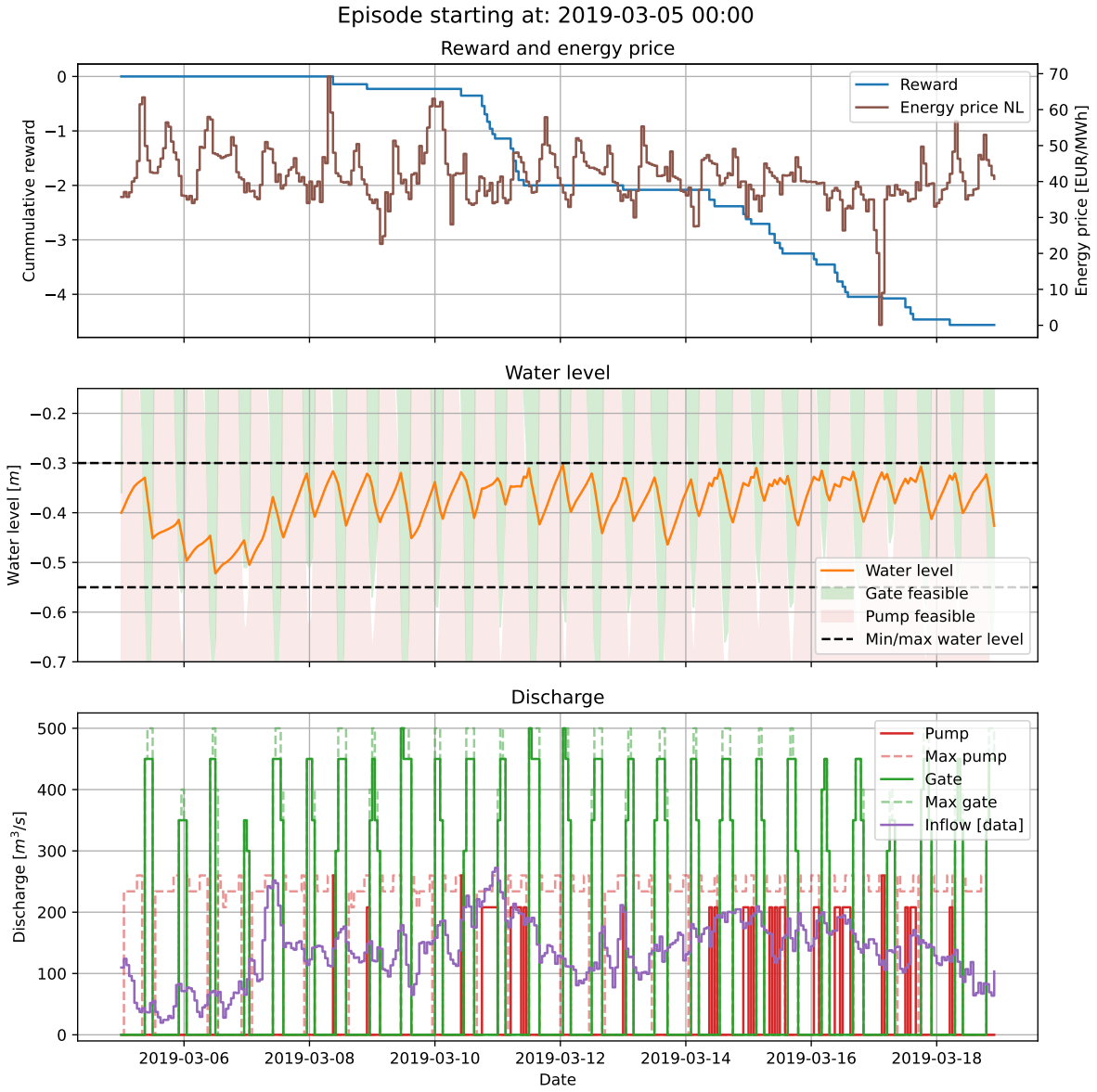


Figure M.2: The RL alternative reward control plan for two weeks starting on 2019-03-05.

M.3. Extreme Negative E

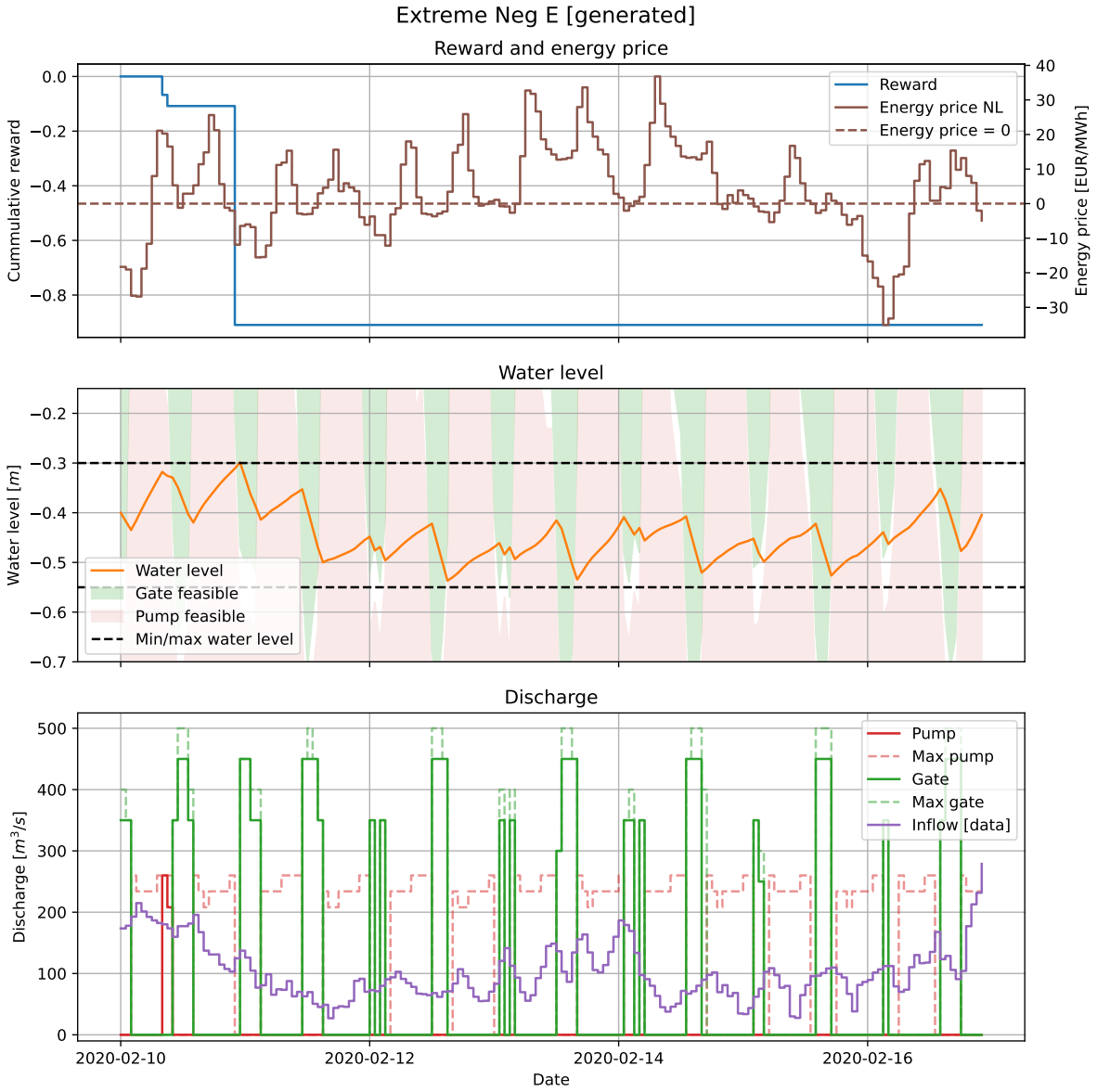


Figure M.3: The RL alternative reward control plan for the extreme negative energy prices scenario.