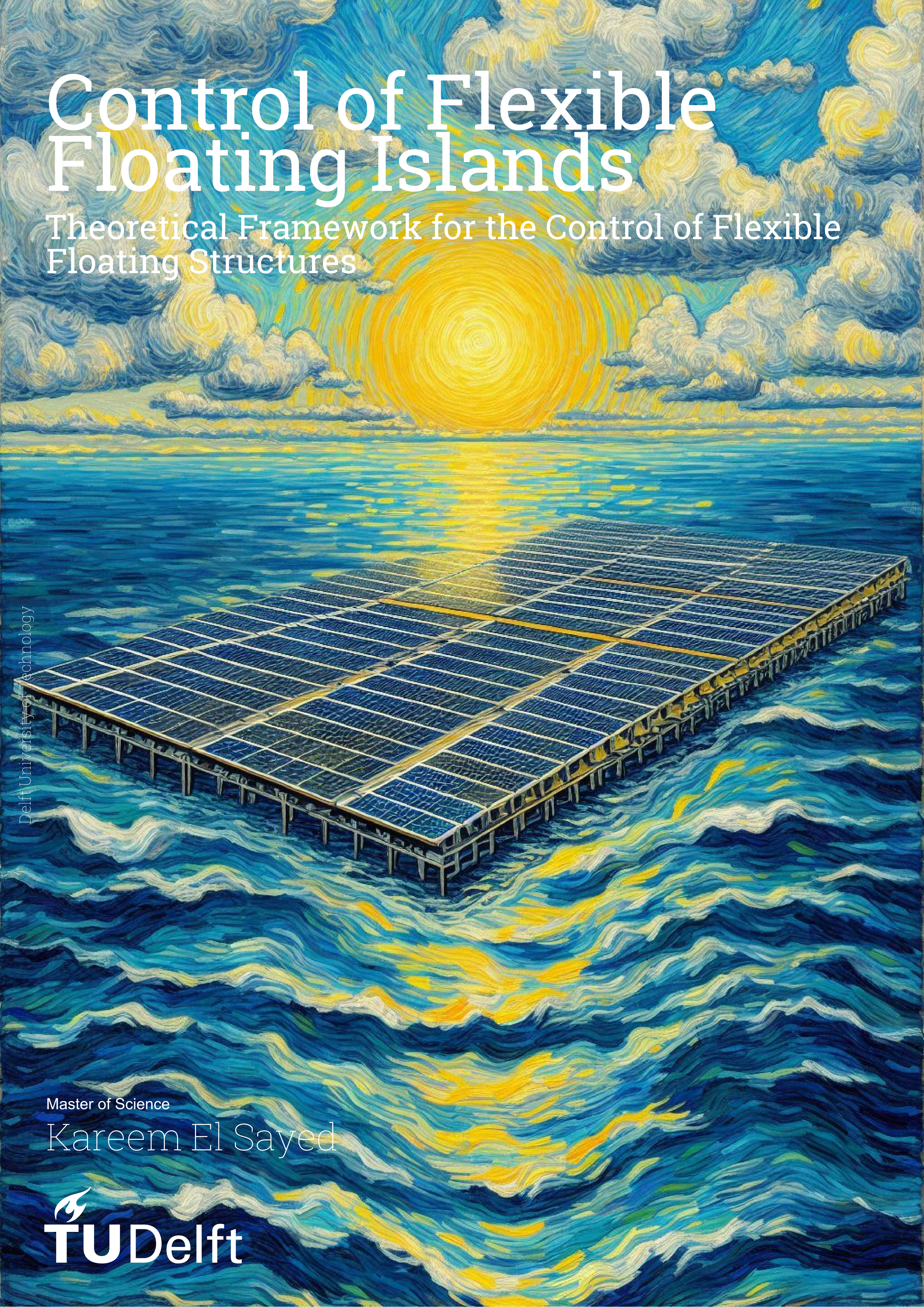
Control of Flexible Floating Islands

Theoretical Framework for the Control of Flexible Floating Structures



Delft University of Technology

Master of Science

Kareem El Sayed

TU Delft

Control of Flexible Floating Islands

Theoretical Framework for the Control of Flexible Floating Structures

by

Kareem El Sayed

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Tuesday November 12, 2024 at 02:00 PM.

Student number:	5110777	
Project duration:	February 12, 2024 – November 12, 2024	
Thesis committee:	Dr. O. Colomès,	TU Delft, supervisor & examiner
	Dr. S. Agarwal,	TU Delft, supervisor
	Dr. E. Lourens,	TU Delft, examiner

Preface

This work is the culmination of over a year of ideas on implementing control in offshore structures. What began as a challenging suggestion from my supervisor, Oriol, during a lecture ultimately led me to take a control course and successfully apply these concepts. I am proud to present this work, where I met the challenge within a year starting with zero knowledge.

I would like to extend my gratitude to Oriol and Shagun, with whom I worked closely throughout this project. During my research internship at the start of my second year, they were incredibly supportive and accessible, enabling me to extend my research into a publishable paper alongside this thesis. Though it was a time-intensive challenge to manage two researches, their assistance and encouragement kept me motivated. I am also grateful to Eliz-Mari for her valuable feedback and her time as a member of my thesis committee. I would also like to thank my friends who were working on their theses at the same time. The feedback and support we shared were invaluable throughout this journey.

Finally, I would like to thank my family for their support and care.

Kareem El Sayed
Delft, November 2024

Summary

In this study, we establish a foundational framework for controlling flexible floating structures by integrating state-of-the-art Model Predictive Control (MPC) with a high-fidelity Finite Element model representing these structures. Our specific objective is to enhance the stability of floating structures in offshore engineering applications. We commence with a comprehensive literature review to identify potential challenges, which informs the development of a well-structured research plan that addresses these challenges comprehensively.

From the insights gained during the literature review, we formulate a theoretical framework that serves as the basis for our methodology. We subsequently examine two distinct control scenarios—regular and irregular wave conditions—serving as a Proof of Concept, and we discuss the significant observations derived from these experiments. In conclusion, we summarize our findings and provide recommendations for future research directions.

We utilize a two-way monolithic Finite Element formulation of the flexible floating structure as a representation of the real-world system we aim to control. For our control strategy, we implement Model Predictive Control, which facilitates the incorporation of advanced functionalities, including the application of constraints. To enhance computational efficiency, we employ a Reduced Order Model (ROM) through Dynamic Mode Decomposition with Control (DMDc), trained using open-loop data derived from the Finite Element model. Additionally, we implement a Kalman filter to reconstruct the system's state from sparse and noisy measurements obtained from the floating structure.

We design two Reduced Order Models specifically for controlling the floating structure under the aforementioned conditions. Prior to executing the control strategies in these distinct scenarios, we conduct an in-depth investigation of DMDc, exploring its relationship with Koopman theory. We generate open-loop data free from pollution using the Finite Element model, which is subsequently utilized to derive the Reduced Order Model. A convergence study is performed by analyzing the eigenvalues and amplitudes of the DMDc, following methodologies established in prior research. We validate the DMDc models against validation datasets, allowing us to select the model exhibiting the smallest validation error. Furthermore, we ensure that the training datasets inherently encompass the wet modes by employing Proper Orthogonal Decomposition.

Our initial control scenario involves managing the floating structure under regular wave conditions. This case is pivotal for acquiring fundamental insights into the control mechanisms. We observe that the frequency of the control input aligns precisely with the excitation frequency. Subsequently, we extend our study to encompass the control of the floating structure in irregular wave conditions, characterized by a sea state defined by the JONSWAP spectrum. Consistent with our findings from the regular wave scenario, we discover that the control mechanism exploits the natural frequencies of the floating structure, which closely correspond to the energy-dense region of the sea state's spectrum. This is achieved by amplifying the reflected wave, thereby counteracting the incoming wave and reducing energy input into the system.

In conclusion, we advocate for further research to bolster the proposed method by examining various structural properties and wave environments, thereby providing robust evidence to validate the approach presented in this study. Our findings indicate that the integration of advanced control strategies, such as MPC and DMDc, holds significant promise for optimizing the performance and stability of flexible floating structures in dynamic offshore conditions.

Nomenclature

Abbreviations

Abbreviation	Definition
MPC	Model Predictive Control
ROM	Reduced Order Model
FEM	Finite Element Method
WEC	Wave Energy Converter
POD	Proper Orthogonal Decomposition
DMD	Dynamic Mode Decomposition
DMDc	Dynamic Mode Decomposition with Control
RNN	Recurrent Neural Network
CFD-FEA	Coupled model with Computational Fluid Dynamics and Finite Element Analysis
FSI	Fluid-Structure Interaction
DoF(s)	Degree(s) of Freedom
SINDy	Sparse Identification of Nonlinear Dynamics
MLP	Multilayer Perceptron
LMPC	Linear Model Predictive Control
NMPC	Non-linear Model Predictive Control
QP	Quadratic Programming problem
SVD	Singular Value Decomposition

Symbols

Symbol	Explanation
$\boldsymbol{\Theta}$	Total DoFs of the two-way monolithic FEM.
$\boldsymbol{\eta}$	DoFs of the structure deformation of the monolithic FEM model.
$\boldsymbol{\phi}$	DoFs of the linear velocity potential of the monolithic FEM model.
$\boldsymbol{\kappa}$	DoFs of the free surface elevation of the monolithic FEM model.
X	Snapshot matrix containing the measured η from the FEM model along the timespan of the simulation.
X'	Shifted snapshot matrix by one time-instant.
Υ	Snapshot matrix containing the applied open-loop control signal on the FEM model.
$\mathbf{x}$	Full order representation of the structure deformation, i.e. the considered true state of the system $\mathbf{x} = \boldsymbol{\eta}$.

Symbol	Explanation
$\hat{\mathbf{x}}$	Full order representation of the structure deformation within the controller, i.e. the estimated state of the system.
$\mathbf{u}$	Control-input.
A	System matrix of the DMDc model.
B	System matrix of the DMDc model capturing the influence of the control-input on the state of the system.
$\tilde{\mathbf{x}}$	Reduced order representation of the structure deformation $\boldsymbol{\eta}$ in the DMDc model, i.e. the estimated reduced order state of the system within the controller.
$\tilde{A}$	System matrix of the Reduced Order DMDc model.
$\tilde{B}$	System matrix of the Reduced Order DMDc model capturing the influence of the control-input on the reduced order state of the system.
$\mathbf{y}$	Noisy sparse measurements from the system we opt to control.
$\hat{\mathbf{y}}$	Reconstructed sparse measurements estimated by the controller.
$\mathbf{v}$	Process white-noise found in the system we opt to control.
$\mathbf{w}$	Measurement white-noise found in the system we opt to control.
D	Selection matrix such that $\hat{\mathbf{y}} = D\hat{\mathbf{x}}$
$\hat{\mathbf{p}}$	The full-order signal that is intended to follow a specified reference trajectory.
$\tilde{\mathbf{p}}$	The reduced-order signal that is intended to follow a specified reference trajectory.
$\mathbf{r}$	The full-order representation of the reference trajectory.
$\tilde{\mathbf{r}}$	The reduced-order representation of the reference trajectory.
C	Selection matrix such that $\hat{\mathbf{p}} = C\hat{\mathbf{x}}$.
G	Augmented system matrices such that $G \triangleq [A\ B]$.
Ω	Found in two contexts. **Context 1**: Ω represents the $2d$ fluid-domain of the FEM model. **Context 2**: Ω is the augmented matrix containing the data of the state measurements and the induced control inputs such that $\Omega = [X\ \Upsilon]^\top$.
$\hat{U}$	State-transformation matrix to go from full-order space to reduced order-space, such that $\mathbf{x} \approx \hat{U}\tilde{\mathbf{x}}$.
$\tilde{r}$	Truncation level of the SVD for the augmented input data Ω.
r	Truncation level of the SVD for the output snapshot matrix X'.

Symbol	Explanation
Λ	Eigenvalues of the system matrix A.
Φ	Dynamic modes of the DMD(c).
$\mathbf{b}$	Amplitudes of the dynamic modes of the DMD(c).
N_s	Number of snapshots considered in the DMDc training data.
Q	Process noise covariance matrix.
R	Measurement noise covariance matrix.
K_k	Adaptive Kalman filter gain.
P	Error covariance matrix of the difference between the true state and estimated state of the system.
F	Controller gain.
N_c	The control horizon in the MPC is used to forecast the system's behavior over a specified future period.
$\mathcal{Q}$	Weight to the emphasize the tracking performance of the controller, i.e. prioritizing alignment of the signal $\hat{\mathbf{p}}$ with the reference trajectory $\mathbf{r}$.
$\mathcal{R}_1$	Weight to fine-tune the smoothness of the control signal by penalizing the changes in the control signal.
$\mathcal{R}_2$	Weight to penalize the magnitude of applied the control input.
$\mathcal{M}$	Forecasting matrix used to project the current estimated reduced-order state, $\tilde{\mathbf{x}}_k$, into the predefined control horizon.
$\mathcal{C}$	Forecasting matrix used to project the control-input, $\mathbf{u}_k$, into the predefined control horizon.
J	Cost/Objective function of the MPC.

Contents

1
Introduction

This chapter introduces the thesis by discussing the context of the research. There we will provide general information needed before preceding. We then present the background information regarding the study itself followed by the objectives we opt to fulfill. At the end of the introduction we present the structure of the thesis.

1.1. Research context

In the world of offshore engineering there is a high urgency for constructing floating structures. Floating structures are, as the name says. structures that exist fully on the surface of the water. Often times these structures are moored to the sea-ground to ensure there steady place. As there is less space for nature and humans on land this can provide numerous opportunities in the nearby future. Moreover, floating offshore structures enable us to deploy renewable sources with huge potential. This can assist in the transition from fossil energy to green energy.

An example is the deployment of so-called floating photovoltaic structures, or simply said, floating solar panels. Multiple feasibility studies have been carried out and are still being carried out [14, 15] which emphasize the big interest in such structures. A new innovative type of offshore structure is currently being studied by multiple research institutes and start-ups and that is the so-called Wave Energy Converter (WEC). Wave Energy Converters are usually floating structures that are able to harvest the wave energy from offshore waters from the mechanical excitation induced in the structure. These WECs come in different shapes and size but often they are designed as point absorbers. Point absorbers are often designed as floating buoys with a generator, often fixed to the sea bottom, that capture the oscillating wave energy using a mooring connection between the buoy and the generator. To be able to capture a large span of wave energy one would have to deploy an array of multiple point absorbers [20]. These type of WECs are considered relatively easy to model and have been tested extensively. A visual representation of the various forms of floating structures can be seen in Fig. 1.1.

A more convenient way to capture wave energy is by deploying so-called floating visco-elastic membranes that are made of dielectric or piezoelectric material, enabling to transform mechanical energy into electrical energy [5]. This type of WEC has been modelled using the Finite Element Method [3] and shows big potential when optimizing the membrane's properties to maximize its energy output [12].

However, this in contrast to point-absorbers, has not been tested in a large scale environment. A second challenge that arises in these kind of structures is the two-way interaction between structure and fluid [32], often difficult to model using analytical models. A third challenge which may occur in practice is the fact that the mooring of these floating structures can be subject to high thrust forces induced by the incoming waves. However, we can say to a certain extent that floating structures have big potential in various forms if the challenges can be dealt with.

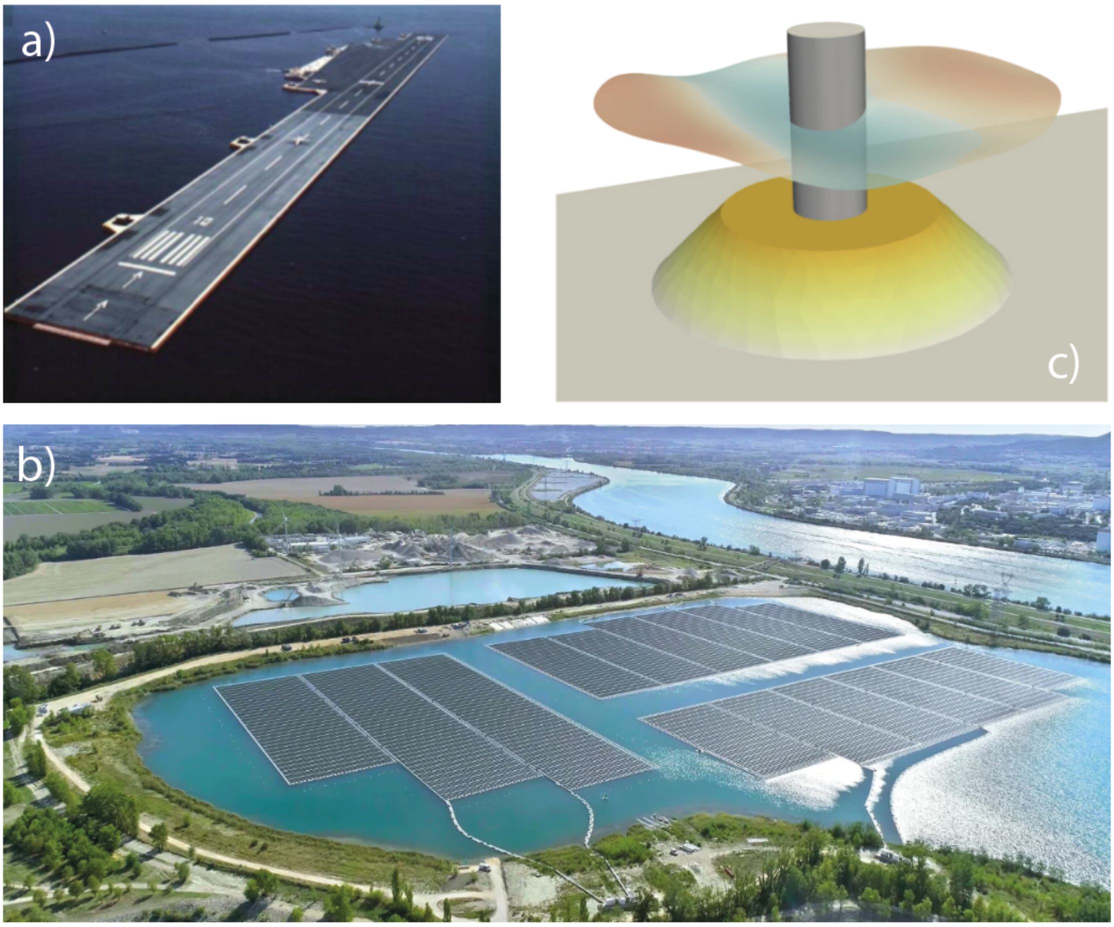


Figure 1.1: A depiction of the various forms of floating structures. All the floating structures in this figure have approximately the same order of magnitude regarding the length but vary strongly in the order of magnitude regarding the height. This shows to a great extent the broad range of flexibility found in floating structures. a) A floating landing strip (height $\sim 10^0$ m) [38] ; b) Floating Photovoltaic Structures (height $\sim 10^{-1}$ m) [44] ; c) Floating viscoelastic membrane as breakwater (height $\sim 10^{-2}$ m) [3]

1.2. Research background

Civil engineers often tend to design structures that are passively controlled. Examples are floating wind turbines that are moored to the seabed. Passive control means that we do not apply any kind of (active) input to the structure, instead we design the structure such that, if it is safely designed, always go back to a stable position. The problem in this is approach is that offshore floating structures tend to deviate relatively more with respect to this stable position in comparison with onshore structures. In this study we opt to design an active control strategy to deal with the numerous challenges described earlier. Active control is the technique where we use measurements from the environment and structure such that we can then find a control input to manipulate the behaviour of the structure to our liking. Active control is widely used in all-day dynamical systems, from the optimization of financial portfolio's to autonomous driving vehicles. The concept boils down to the following: perform an online optimization of an objective function to find the current control input subject to constraints.

Multiple studies have been done studying the active control of floating structures using different types of control input. In the general context of floating beam structures we find literature investigating boundary control, that is the control input acts on boundaries of the floating structure, to minimize the vertical vibration of the floating structure [49]. In a different study, not related to floating structures, we find modal-based control of multibody very flexible structures [27]. In this study they investigated the control of flexible beam like structures that are connected with hinges. It is shown that flexible structures containing hinges have highly nonlinear dynamics. A closely related study to a control strategy in the context of floating structures has shown great potential by implementing stiffness-adjustable hinges in the floating structure [47].

However, in these studies we do not find the controllers to be implemented in a multi-physical context, that is taking the two-way interaction between the fluid and structure into account. It is common practice to use classical control theory as long as that is possible. However, to be able to use classical control theory a deep understanding of the model is required. Classical control theory often requires state-space models that are formulated in continuous time-domain. A commonly used control strategy known for its robustness is the so-called Model Predictive Control (MPC). Model Predictive Control is considered to be state-of-the-art technique that allow us to design a controller for multi physical problems. We will thereby implement the MPC in combination with a Finite Element Model such that we can present a reliable control strategy for a coupled problem taking the fluid-structure interaction into account. Different studies have proven the effectiveness of control in combination with Finite Element Models [11, 7]. This raises multiple challenges including computational demands that may not be feasible for online control.

In the world of control we often make use of so-called Reduced Order Models, most commonly denoted as ROM. ROMs are accurate representations of the full-order model using as few degrees of freedom by reducing the full-order dimensionality, hence its name. A wide range of techniques, spanning from the mathematically well-understood Proper Orthogonality Decomposition (POD) [50] to data-driven methods like employing Machine Learning through a reduced-order Recurrent Neural Network-Model Predictive Control (RNN-MPC) [45], can achieve this objective.

1.3. Research objectives

Floating structures encapsulate a wide range, each with their own specific characteristics. However, in this study we opt to use a general formulation applicable for the described structures earlier. We will consider the floating structures as floating beams. In the context of the wide range of functionalities accompanied with floating structures we find multiple objectives that can be reached with control. However, in this study we limit the scope by testing our control framework for the following objective:

1. *To enhance serviceability conditions and minimize the wear of structural elements, the dynamic response of the floating structure should be reduced.*

By defining new objective functions and introducing new kind of control inputs we can consider different cases. Hence, we define our primary aim as follows

Primary Aim

The aim of this study is to develop a framework and evaluate a control strategy for flexible floating structures by integrating Model-Predictive Control (MPC) with a Finite Element Model (FEM), with the goal of optimizing the performance and stability of flexible floating structures in offshore engineering applications.

To be able to achieve this aim we first have to find answers to the following research questions

1. *Exploring Existing Studies and Methodologies*
   (a) What are the existing methodologies employed in controlling flexible floating structures, especially regarding the excitation of the structure with the free surface elevation?
   (b) What insights can be derived from previous studies utilizing MPC and discrete time-integration scheme in the context of large-scale structures and dynamical systems?
   (c) How does the use of MPC in controlling boundaries or stiffness-adjustable hinges impact the performance and stability of flexible floating structures?
2. *Practical Implementation and Feasibility:*
   (a) What are the practical implications and challenges of implementing MPC within the context of controlling flexible floating structures?
   (b) How feasible is the integration of MPC within a general framework for flexible floating structures, considering the complexity of different structural characteristics?
3. *Optimization and Control Objectives for Floating Structures:*
   (a) How can MPC be adapted/implemented to achieve different control objectives, such as reducing the dynamic responses?
   (b) How can Reduced Order Models (ROMs) be implemented to enhance the efficiency and effectiveness of online control systems in dynamic environments?

With this approach we would like to fill the research gap regarding the fact that no literature has been found that implemented MPC to control flexible floating structures in a multi-physical context. The multi-physical formulation is able to mimic the reality more accurately since we consider more aspects of the entire model. In case of successful implementation of MPC to the multi-physical formulated problem we hope to show the big potential of controlling flexible floating structures.

1.4. Structure of the thesis

This thesis is structured as follows: in chapter 2 we present a comprehensive review of the considered theory found in existing literature on Model Predictive Controlled (flexible) floating structures and the integration of MPC with Finite Element Modelling. Moreover, we explore literature about the use of reduced order modelling in the case of online optimization. We will then precede to chapter 3 where we discuss the relevant factors influencing the research including a detailed explanation of the independent, dependent and confounding variables. We will then combine all the required components into one framework depicting the inter-dependencies between the separate components. There we assess each component and discuss the limitations of the components and the effects of the found inter-dependencies. In chapter 4 we describe the used (mathematical) framework to describe the implementation of the online MPC. In 5 we present the obtained results and findings and we discuss these results and findings. We finally wrap up the thesis in chapter 6 where we conclude the research and give insights and recommendations for future work related to this study.

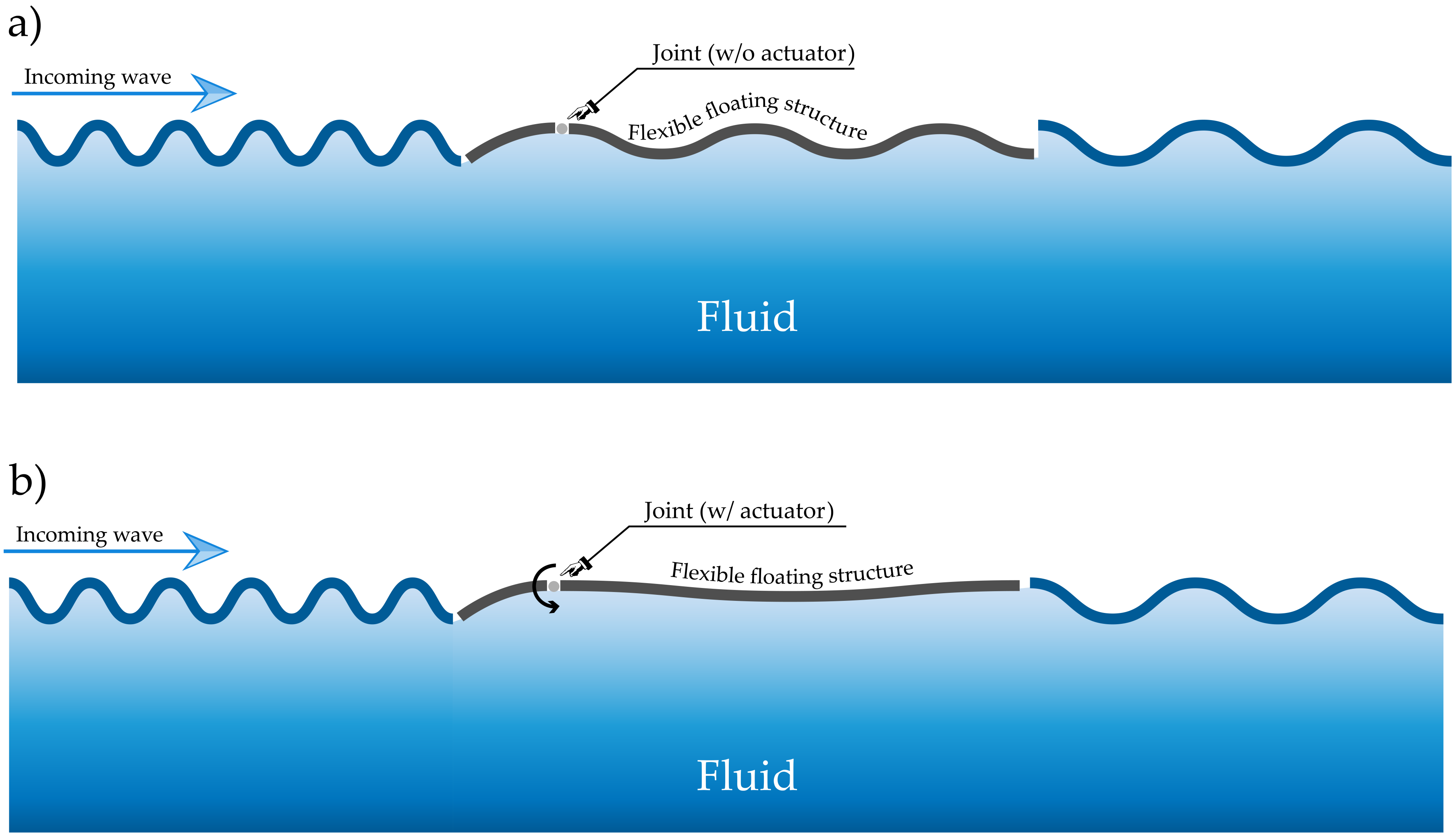


Figure 1.2: Comparison of a flexible floating structure with and without actuation at the joint. a) The floating structure without an actuator at the joint shows passive response to the incoming wave. b) The floating structure with an actuator at the joint demonstrates active control, allowing for adjustment in response to the incoming wave. The actuator enables more precise control of the structure's motion, potentially improving energy absorption or reducing unwanted motion in wave conditions.

2 Literature Review

In this chapter we present a brief overview of the theory found in the literature regarding the essential topics that are part of this study. In the first section 2.1 we discuss the variety of floating structure models found in literature. We then continue to section 2.2 to discuss the control strategies of floating structures and how these are effectively deployed to achieve the control objectives. In section 2.3 we discuss the opportunities and challenges when combining a control strategy within a Finite Element Model. Subsequently, we explore the opportunities of reduced order modeling in section 2.4 to achieve feasible control strategies. In section 2.5 we introduce the control theory we opt to implement to reach maximum performance. Finally, we discuss optimization techniques in section 2.6 to determine which optimization algorithms are the most suitable for this study.

2.1. Modeling of (flexible) floating structures

Before diving into the control of (flexible) floating structures we need to understand how these structures are modelled in the literature. Floating structures are a hot topic for years and therefore a broad range of studies have been carried out exploring the modeling techniques of these relatively complex structures. In the literature we find three main modeling methods, each with their advantages and disadvantages.

2.1.1. CFD-FEA method

In the broad context of floating structures we should consider floating structures that have complicated components. An example are the so-called Large multi-body-floating offshore structure [31] usually found in the design of offshore ports [31]. Although the structure may seem to be very large and rigid, they are mainly modelled as extremely flat and flexible due to their extreme lengths and relatively small heights. As can be seen in Fig. 2.1 these multi-body floating structures consist of multiple substructures (partially) submerged under the waterline separated by equally distanced spacing. Because of the relatively small size of the floating substructures compared to the entire structure we find strongly nonlinear phenomena [31].

A possible approach to model these type of floating structures is by coupling a Computational Fluid Dynamics model with a Finite-Element Model [31]. This proposed method uses commercial software to model and calculate fluid quantities, such as pressure and shear forces in the fluid domain, using the finite volume method. Subsequently, these quantities are applied to the floating structure in a finite-element model using again a different commercial software. These quantities will therefore influence the velocities and acceleration of the nodes in the structural Finite-Element Model. Having obtained the deformations of the structure this will then be returned to the CFD software to update the pressure field and the velocity field used in the next time-step. Similarly, the FEM now has the velocities and accelerations of the nodes that are

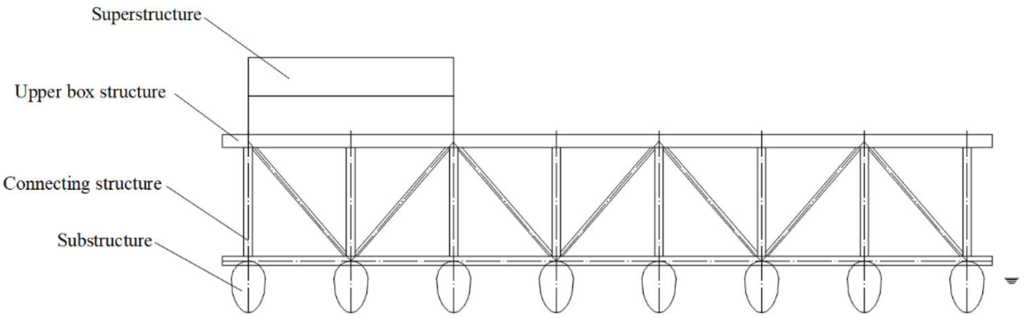


Figure 2.1: A possible cross-section of a multi-body floating structures. [31]

used to compute the next time-step dynamics. A clarifying scheme can be found in Fig. 2.2. According to the conducted study that proposed this method, they were able to fully consider the non-linear two-way interaction between the elastic structure and fluid. They were even able to capture phenomena such as wave overtopping [31]. Although this method is able to capture complex nonlinear behaviours, one should realize that in the context of control we usually want to use a model that is as simple as possible, yet complex enough to be used in practice. One could therefore consider this implementation of this method as a realistic ”plant” that is controlled by the use of a simpler model. In this study we will mainly focus on obtaining the simpler model we can use to control floating structures.

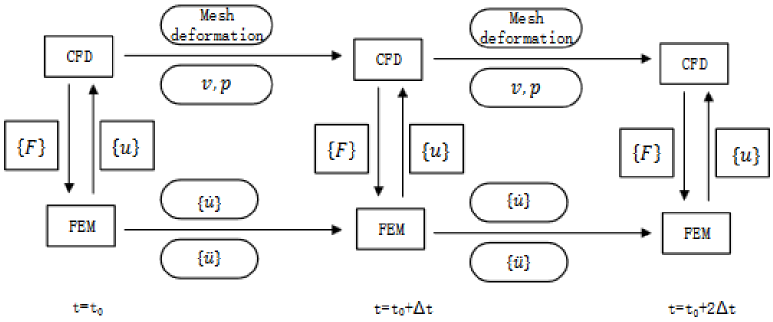


Figure 2.2: Proposed CFD-FEA framework to model complex nonlinear behaviour of VLFS. [31]

2.1.2. Two-way decoupled FSI analysis

A different approach for modeling floating structures is by decoupling the fluid-structure interaction problem into two problems: a hydrodynamic problem used for the modeling of the fluid and a mechanical problem used for the modeling of the structure [9]. Similarly to subsection 2.1.1 we consider the small ratio of height to horizontal dimensions, forcing us to model the floating structures as (large) flexible structures taking the hydro-elastic behaviour into account. However, contrary to subsection 2.1.1 this method uses the linear hydro-elastic theory of floating structures, introducing the following assumptions [9]:

Assumption 1 The flow within the fluid-domain Ω is considered to be incompressible, inviscid and irrotational. This corresponds to the Linear Potential Flow Theory.

Assumption 2 The floating structure is considered to be homogeneous and shows linear behaviour.

Assumption 3 The transverse displacement of the structure and steepness of the incident wave are considered small, and there is no significant surge displacement.

In the proposed method they used the modal expansion method to decouple the fluid-structure

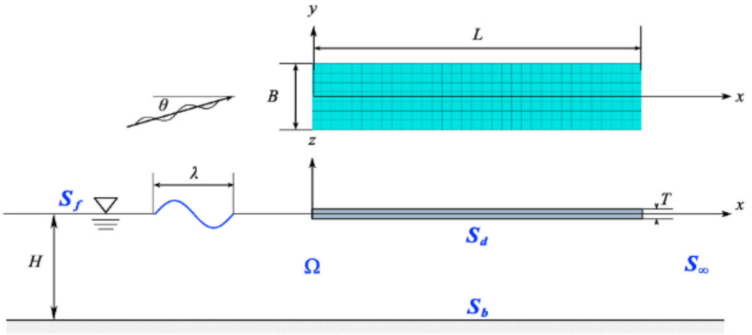


Figure 2.3: The problem setting: bottom figure is the side view showing the fluid domain with an incident wave heading to a floating beam; the upper figure shows an incident wave with defined by an attack angle heading to a rectangle plate floating on the fluid domain. [9]

interaction. However, the structural analysis of the floating structure has been done using the **dry-modal functions** of the structure. It has been shown that this can lead to incorrect modeling of the fluid-structure interaction [3], since the added mass can highly influence the natural frequencies of the flexible floating structures. Hence, this method offers simplicity but neglects the essential added-mass contribution to the fluid-structure interaction, which may be crucial in controlling floating structure. Furthermore, the authors propose a new wave-generation model, which they call *NewWave model* [9]. This model is used to generate focused wave groups such that realistic extreme sea states can be simulated.

2.1.3. Two-way monolithic FSI analysis

The following approach may be considered as the compromising method between the two previously mentioned methods. In this method the two-way interaction between the floating structure and fluid is modelled using a monolithic formulation [33]. This monolithic formulation enables us to model the considered problem as a mixed-dimensional problem and to solve this mixed-dimensional problem without the use of an iterative algorithm or using a partitioned scheme, as has been done in the methods described in 2.1.1 & 2.1.2. This approach therefore inherently takes the wet-modal response of the floating structure into account, which is a major advantage compared to the method described in section 2.1.2. Similarly, this method uses the same assumptions made in section 2.1.2 which makes it therefore unable to capture complex nonlinear wave behaviour. However, in the context of control this method is relatively simple but yet able to capture the wet-modal behaviour of the structure. Furthermore, because of the monolithic formulation it is computationally convenient to use in combination with optimization techniques. The used problem setting can be found in Fig. 2.4.

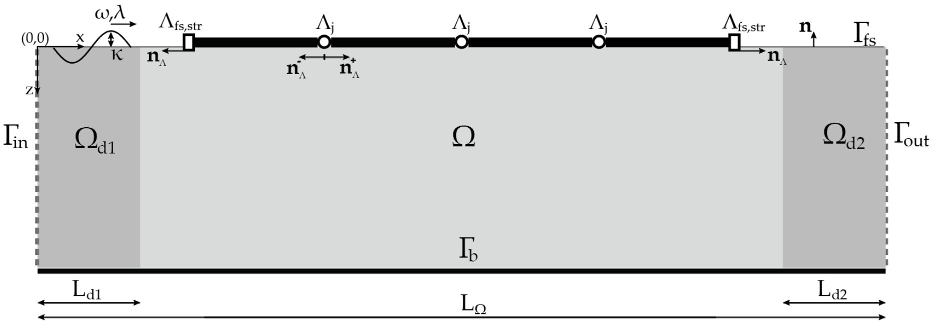


Figure 2.4: The proposed problem setting depicting a floating structure with a rotational spring floating on the surface of the fluid domain Ω with an incident wave causing the excitation of the floating structure due to the boundary and interface conditions of the monolithic mixed-dimensional formulation of the problem. [33]

2.2. Control strategies

Floating structures are complex structures that deserve special treatment when studying a control strategy. First, we investigate the literature found which implements MPC for the control of dynamical systems that have a large number of degrees of freedom. This is necessary to obtain valuable insights such that we can anticipate on possible challenges. We will then mainly look into two things: possible control inputs, and their performances in the literature, and measurement strategies used in the control scheme.

2.2.1. Control of high-dimensional dynamical systems

Different studies have implemented control for high-dimensional dynamical systems, demonstrating the possibility to successfully control such systems.

A similar study to ours, although a different system, is the control of soft-robots. The modeling of soft-robots often require the use of Finite Element Models because of the high non-linear behaviour and complex geometries encountered in such systems. Therefore, one will deal with the Degrees of Freedom of the FEM as the states of the system. In a previous study they managed to control soft-robots, described by 9768 DoFs as its states, by implementing a Reduced Order Model Predictive Controller [41].

Since in this study we will be considering the control of flexible floating structures, we find a good amount of inspiration from such studies. Therefore, later on in the literature study, we will be presenting possible techniques to find such Reduced Order Models of high-dimensionsal systems.

2.2.2. Control inputs

In the literature we find three interesting types of control inputs: boundary control using actuators, the use of stiffness-adjustable springs and the control of the mooring stiffness.

Boundary control

A possible and effective way of controlling floating structures is by actively controlling its boundaries. This can be done by the use of (pneumatic) actuators. It has been shown that the active control of the boundaries of a floating structure, those are the upstream and downstream ends of the structure, resulted in significant reduction of the hydro-elastic response [49]. This enhanced the serviceability of the structure. However, this active control technique was mainly effective at suppressing the floating structure's motion at the ends and severe vibrations could still be observed in the middle of the structure. This could however be solved with the use of passive control. Energy dissipation device where installed between the ends of the floating structure to absorb the energy from the incident waves [49].

Stiffness-adjustable springs

Floating structures are often designed as multiple modules, each can be modelled as a beam structure, connected by hinges. As depicted in Fig. 2.4 these hinges can be modelled as rotational springs with linear rotational stiffnesses. The idea is to control the structure by imposing control forces delivered by an actuator that is hydraulically driven [47]. This is proven to be effective and has the benefit of having small output requirements compared to thrust generation forces as control input [47].

Mooring control

An approach that has not been studied closely as the previous two control-input strategies is the control of the mooring line. The control mooring lines in the context of floating structures can be challenging due to the fact that mooring lines in reality exhibit highly non-linear behaviour. Only when the mooring is deployed as a taut cable we can often observe linear behaviour which can then be modelled as a spring with translational stiffness. We find inspiration for this type of control in the control of mesh reflector antennas [48]. Mesh reflector antennas are membrane like structures often installed in satellites. The surface of these membranes have to be controlled to exhibit desirable electromagnetic behaviour. In these cases we find that the membrane can be controlled by active cables consisting of piezoelectric actuators. These type of cable can therefore in a sense be compared to taut mooring that can be specifically deployed in cases of very thin floating structures, such as visco-elastic membranes [3]. Unfortunately, no literature has been found implementing this type of control in the context of flexible floating structures. It is therefore difficult to draw conclusions on the performance of this type of control.

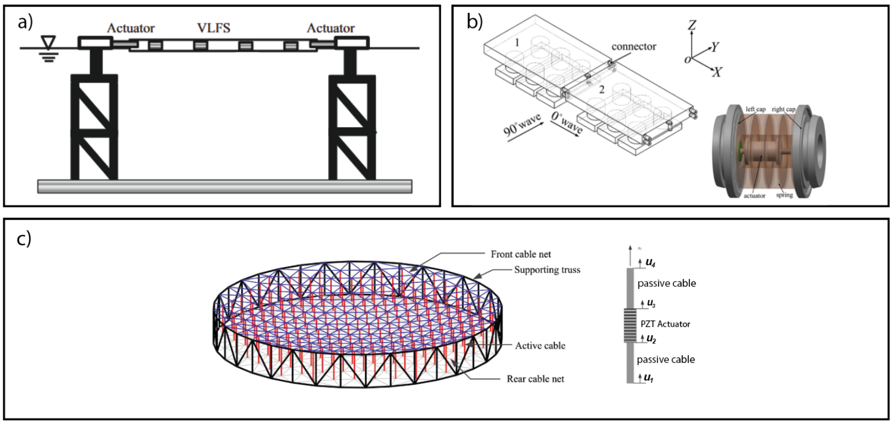


Figure 2.5: An illustration of possible control-inputs. a) Boundary control using (penumatic) actuator at the boundaries of the VLFS. [49] b) Stiffness-adjustable actuators connecting separate modules of the VLFS. [47] c) Inspiration for the control of linear mooring systems. [48]

2.2.3. Measurement strategies

An important aspect in control are measurements we conduct to gather information from the surrounding environment. The driving force of floating structures is primarily from the incident waves. In the context of control of WECs we find that the upstream measurements are often limited in size by deploying a small number of probes in the spatial domain [40]. However, in this study we opt to control the floating structure using a small number of probes only located on the floating structure. We hypothesize that to control the structure, especially if the structure has large inertia, it is sufficient to know its deformation. This hypothesis will be justified later in the report.

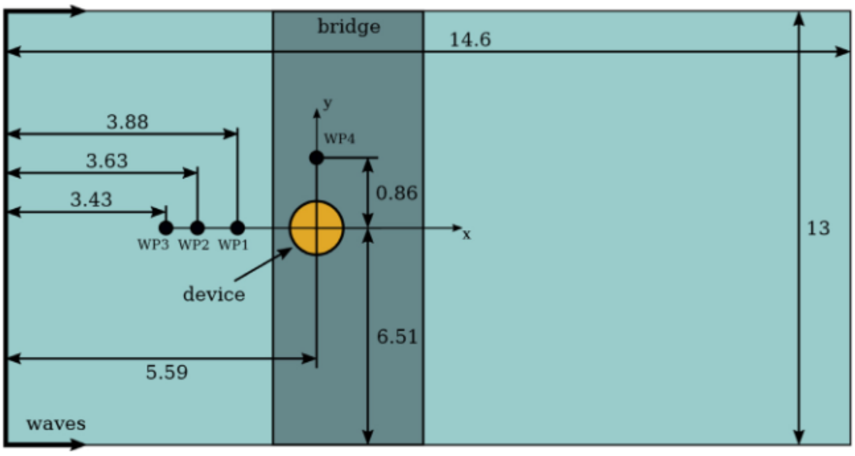


Figure 2.6: Top-view of the problem setup used in a competition to control a point-WEC. The yellow colored circle represents the point-WEC where a total of four wave-probes, denoted as WP in the figure, are used to gather upstream and even parallel wave measurements for the control strategy. [40]

2.3. Optimal control with Finite Element Models

To efficiently and accurately control floating structures we need a good model taking the fluid-structure interaction into account. These models are often difficult to formulate using analytical solutions or even semi-analytical solutions, let alone control using these type of formulations. A more convenient method in this case is to implement control in combination with a Finite Element Model system. In this section we explore the opportunities and difficulties arising in the implementation of this approach.

2.3.1. Time-stepping in Finite Element Method

In order to understand the main method let us define a general mathematical framework for the problem we are dealing with. As described above we will be dealing with a multi-physical problem where we are concerned with the deformation of the structure, the linear velocity potential and the free surface elevation, denoted as $\eta(\boldsymbol{x},t)$, $\phi(\boldsymbol{x},t)$ and $\kappa(\boldsymbol{x},t)$ respectively. Let us now combine all these quantities of interest into one vector $\boldsymbol{\theta}$.

$$\boldsymbol{\theta} = \begin{bmatrix} \eta(\boldsymbol{x},t) \\ \phi(\boldsymbol{x},t) \\ \kappa(\boldsymbol{x},t) \end{bmatrix} \tag{2.1}$$

In control problems we are mostly dealing with the behaviour of the system in the time-domain. In the continuous case we can write a generalized PDE that when solved describes the coupled behaviour of our quantities of interest.

$$f(\ddot{\boldsymbol{\theta}}, \dot{\boldsymbol{\theta}}, \boldsymbol{\theta}) = f^{ext} \tag{2.2}$$

In such a case, we are dealing with a system of coupled non-linear PDEs having a source term on the right hand-side. To be able to solve this PDE we first will need to solve this PDE in space using the Finite Element Method where we approximate our quantities of interest in Eq.2.1 using the Finite Element functions such that $\eta(\boldsymbol{x},t) \approx \eta_h(\boldsymbol{\eta}(t), u(\boldsymbol{x}))$, $\kappa(\boldsymbol{x},t) \approx \kappa_h(\boldsymbol{\kappa}(t), v(\boldsymbol{x}))$ and $\phi(\boldsymbol{x},t) \approx \phi_h(\boldsymbol{\phi}(t), w(\boldsymbol{x}))$. In simple terms, we approximate our unknown quantities in Eq.2.1 using a discretized mesh of the domain where we solve a system of equations obtained from the governing equations such that we find the degrees of freedom associated to each discretized cell. Now using the basis functions $[w, v, u]$ we can interpolate between each cell to form the approximated fields of our unknowns. However, note that in Finite Elements we are mainly solving for the degrees of freedom since the basis functions are predefined. For full derivation see **??**. To solve the problem in time we can then "march" our quantities of interest through the discretized time-domain using the initial conditions. Now let us consider the following vector that describes the discretized quantities of interest in space such that

$$\boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{\phi} \\ \boldsymbol{\kappa} \end{bmatrix} \tag{2.3}$$

That is we have a vector containing the stacked DoFs of interest. Using the governing equation we construct a second-order dynamic equation such that

$$M\ddot{\boldsymbol{\Theta}}_{k+1} + C\dot{\boldsymbol{\Theta}}_{k+1} + K\boldsymbol{\Theta}_{k+1} = \mathbf{f}^{ext}_{k+1} \tag{2.4}$$

Such systems can be solved in time using a time-integration scheme such as the Newmark-β scheme [33, 7]. We can then rewrite the system of Eq. 2.4 in a first-order state equation such that we can impose a control-input $\boldsymbol{u}_k$ on the system.

$$\boldsymbol{\Theta}_{k+1} = f(\boldsymbol{\Theta}_k, \boldsymbol{\Theta}_{k-1}, \dot{\boldsymbol{\Theta}}_{k-1}, \ddot{\boldsymbol{\Theta}}_{k-1}, \mathbf{u}_k) \tag{2.5}$$

In this notation we have our subsequent solution $\boldsymbol{\Theta}_{k+1}$ that can be computed using the current solution $\boldsymbol{\Theta}_k$ and the previous solution $\boldsymbol{\Theta}_{k-1}$, accompanied with its time derivatives up to the second order, and the control input $\mathbf{u}_k$. Now to compute this control input we will be making use of a Model Predictive Controller since the coupling between the quantities of interest can be difficult to comprehend. We find that a Model Predictive Controller is robust and often effective in such cases. However, we first need to find a simpler representation of the FE model which we can use in our MPC.

2.4. Reduced Order Modeling (ROM)

Model Predictive Control, or control in general, prefers low-dimensional systems for feasibility and computational-efficiency reasons. Therefore, using a full Finite Element Model in the controller can be very disadvantageous for these obvious reasons. Implementing Eq. 2.5 directly as our model in the controller will reduce the computational speed significantly, especially if the used Finite Element Model has a high resolution for convergence reasons. Instead, we opt for a so-called Reduced Order Model, commonly known as ROM. A Reduced Order Model (ROM) captures the most dominant behaviour of a dynamical system using less degrees of freedom found in the state-description. In this chapter we discuss which techniques are found in the literature to obtain a ROM from high-dimensional systems.

2.4.1. RNN-MPC

Artifical Intelligence is currently being used in various cases. Hence, we find multiple studies where Neural Networks in the form of auto-encoders are used to map high-dimensional problems onto a latent-space resulting in a ROM. In the study done by [45] they developed an auto-encoder mapping the states of a non-linear problem onto a latent-space. This is using data obtained from an open-loop simulation where a control-action was induced on the system. By doing this one can obtain data to feed to an auto-encoder to learn this simpler latent-space describing the highly non-linear process. Additionally, they used a so-called Recurrent Neural Network (RNN) to capture the temporal dynamics of the system.

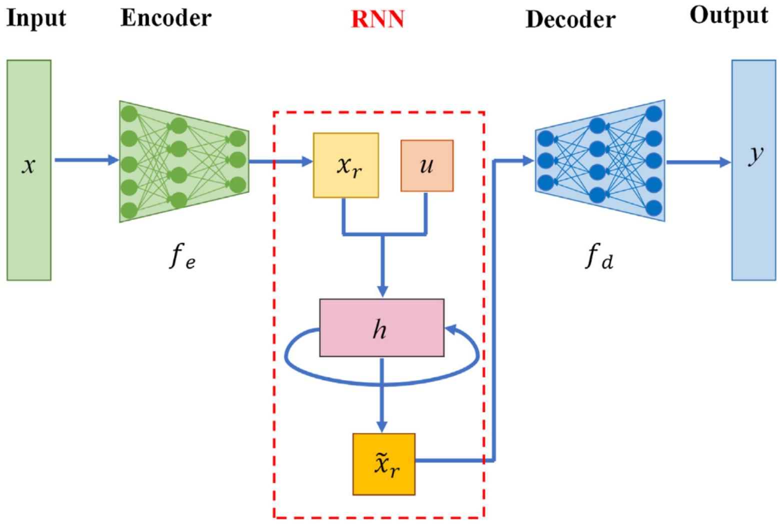


Figure 2.7: An auto-encoder Multi-Layer Perceptron (MLP) coupled to a RNN to learn the dynamics of high-dimensional system. [45]

In order to make use of the RNN-MPC we need to have data of our states and the corresponding control inputs. The ROM in Fig. 2.7 is built such that we feed the MLP the full-state to encode it using a non-linear encoder function such that

$$x_r = f_e(x) \tag{2.6}$$

where x_r is the latent-space of the state. In the RNN we add the corresponding control input such that RNN learns that the current state and its corresponding open-loop control input result in the latent future states $\tilde{x}_r$. Having trained the RNN to successfully predict the future states in time we can then decode this latent future state to the full-dimensional future state y using a non-linear decoder such that

$$y = f_d(x_r) \tag{2.7}$$

Neural Networks are known for their robustness and can capture highly non-linear behaviour by only using data. This is because Neural Networks in theory can map any input to any output by training any function, this is in accordance with the *universal approximation theorem*. However, this results in the main disadvantages of Neural Networks. The complexity of Neural Networks often lead to black-box models. Neural Networks are dependent on various hyper-parameters, such as the activation functions and the depth of the network. Each different set of hyper-parameters will result in a different model. This makes it often difficult to find any physical meaning in the ROM.

2.4.2. SINDy auto-encoder with control

A similar to RNN-MPC but more interpretable approach would be the implementation of a SINDy auto-encoder with control. SINDy stands for *Sparse Identification of Nonlinear Dynamics*. This method, as it name says, opts to discover the governing equations of the data using the fewest terms as possible [22]. The SINDy auto-encoder uses a MLP to obtain a latent-space z as can be seen in Fig.2.8. We can then make a so-called library of terms that we construct using knowledge about the considered problem. By constructing a loss function that takes the reconstruction error of the auto-encoder and the SINDy loss we can find interpretable governing equations depicting the relation between the dynamical system and the control input. These governing equations can be considered as the ROM and could be directly implemented in the MPC.

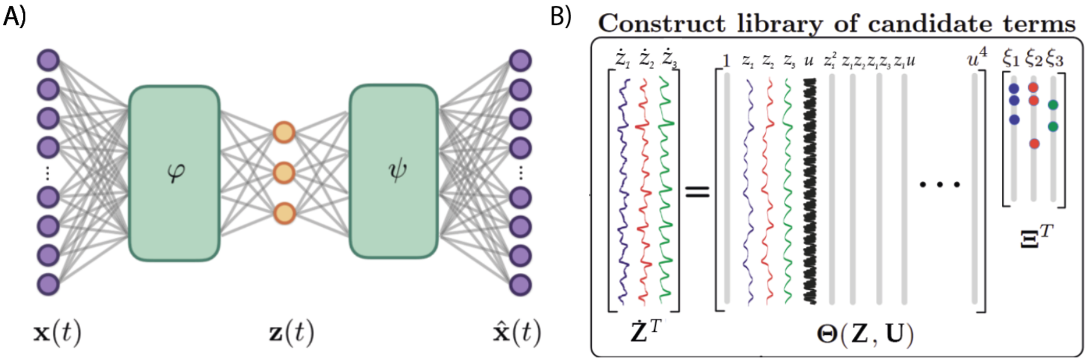

A)
φ
ψ
$\mathbf{x}(t)$
$\mathbf{z}(t)$
$\hat{\mathbf{x}}(t)$
B)
Construct library of candidate terms
$\dot{z}_1$ $\dot{z}_2$ $\dot{z}_3$
1 z_1 z_2 z_3 u z_1^2 z_1z_2 z_1z_3 z_1u
u^4
ξ_1 ξ_2 ξ_3
=
$\dot{\mathbf{Z}}^T$
$\mathbf{\Theta}(\mathbf{Z},\mathbf{U})$
$\mathbf{\Xi}^T$

Figure 2.8: A) The auto-encoder mapping the high-dimensional input x to a latent-space z. B) A SINDy model where the matrix Ξ^T is learned such that the fewest terms in the matrix $\Theta(\boldsymbol{Z}, \boldsymbol{U})$, where $\boldsymbol{U}$ is the control input from the data, can construct a ROM in the form of Ordinary Differential Equations. [22, 21]

SINDy is a strong method that is proven to be able to model highly non-linear dynamics. However, SINDy is highly influenced by the library containing the candidate terms and thus depends on the knowledge of the system. Moreover, the implementation of SINDy can be error prone because it requires a tedious workflow.

2.4.3. DMD with control (DMDc)

In both previously mentioned methods the use of Neural Networks was inevitable. This reduces the ability to work with a ROM that can be expressed as in Eq.2.9 and Eq.2.10. Because we are dealing with a problem containing fluid-structure interaction it is less convenient to consider the dry-modes of the floating structure. In multiple structural analysis cases where FEM is been used one would often compute the modes using the Modal Expansion Method. This would then allow us to formulate a ROM using the orthogonality principle of the dry-modes. However, because we are opting for a formulation using the wet-modes, since they capture the behaviour involving the fluid-structure interaction due to the added mass and added stiffness, this will not

be possible. This is due to the fact that wet-modes are not orthogonal to each other, which is a prerequisite in the Modal Expansion Method. However, it is still possible to obtain a ROM by the use of the so-called Dynamic Mode Decomposition (DMD) method. This is a data-driven method relying on snapshots obtained from, in this case, numerical simulations [18]. This method allow us to obtain a ROM expressed in the dynamic modes instead of the high-dimensional full model as described in Eq.2.5. Contrary to the Modal Expansion Method we now obtain dynamic modes that may not be orthogonal to each other. Additionally, the DMD returns temporal information depciting the growth and decay rate of the modes in the system.
Since we are dealing with control we have to consider the closely related method which is Dynamic Mode Decomposition with control [23]. By defining a control input we can obtain data from the full FE-model. We then split this data into three parts: the "current" state $\mathbf{x_k}$, the applied control input $\mathbf{u_k}$ and the "future" state $\mathbf{x_{k+1}}$. Mind that the "future" state is simply the "current" state shifted by one time instant. As can be seen in Fig. 2.9 these measurements are then assembled into three matrices:X, X' and Υ.

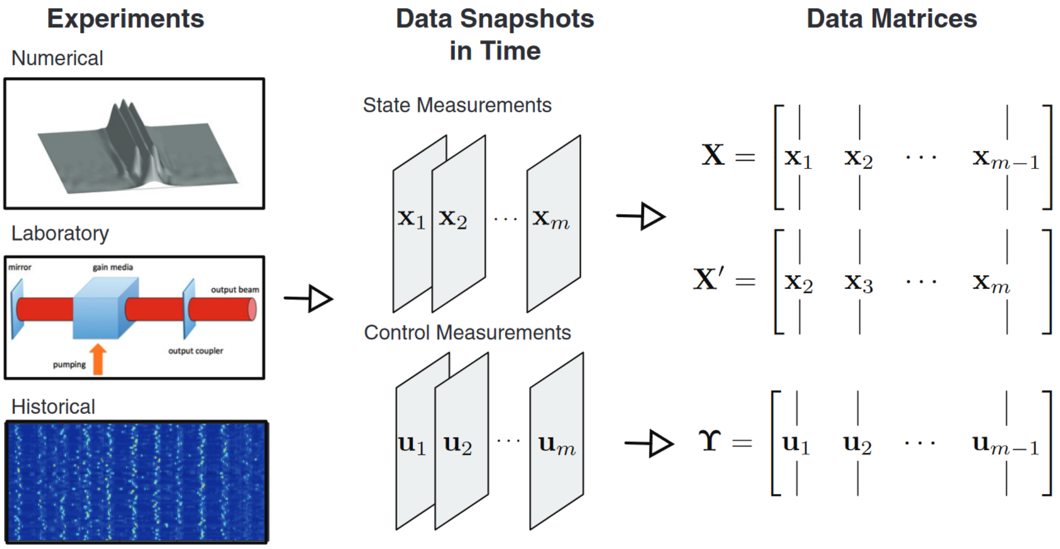

Experiments
Numerical
Laboratory
mirror
gain media
output beam
pumping
output coupler
Historical
Data Snapshots in Time
State Measurements
$\mathbf{x}_1$ $\mathbf{x}_2$ ... $\mathbf{x}_m$
Control Measurements
$\mathbf{u}_1$ $\mathbf{u}_2$ ... $\mathbf{u}_m$
Data Matrices
$\mathbf{X} = [\mathbf{x}_1 \; \mathbf{x}_2 \; \cdots \; \mathbf{x}_{m-1}]$
$\mathbf{X}' = [\mathbf{x}_2 \; \mathbf{x}_3 \; \cdots \; \mathbf{x}_m]$
$\Upsilon = [\mathbf{u}_1 \; \mathbf{u}_2 \; \cdots \; \mathbf{u}_{m-1}]$

Figure 2.9: The principle of DMDc is applicable for different experiments. In this study the data will be obtained from a numerical model, namely the FE-model. [18]

There are two main uses of DMDc. The first use is *system identification*. This is primarily used when the underlying model is highly non-linear or the underlying model is completely unknown. Using DMDc we are able to find a state-space model of the form found in eq. 2.9. The second use, which is more interesting for this study, is the construction of a ROM. The underlying model in this study is known through our FE-model and the governing equations. However, we would like to find a ROM expressed in its dynamic modes such that we have the following ROM

$$\tilde{\mathbf{x}}_{k+1} \approx \tilde{A}\tilde{\mathbf{x}}_k + \tilde{B}\mathbf{u}_k \tag{2.8}$$

where the state $\tilde{\mathbf{x}}$ is the reduced-order state. A similar study has applied this method to a 3D FE-model controlling the temperature of a 3-dimensional domain using thermal actuators [10]. A full FE-model of the heat equation is used to produce open-loop simulations. In this study they succeeded to obtain a ROM for the controller that was described using 28 states instead of the original 98997 states of the FE-model.

2.5. Model Predictive Control

In Model Predictive Control (MPC) the control strategy involves calculating the current input to the system by finding a optimum of a user-defined objective function at every sampling moment. This process begins with the current system state as the starting point. By optimizing a defined performance index, an optimal sequence of control actions is determined. From this sequence, the first control action is selected and implemented on the system [7].

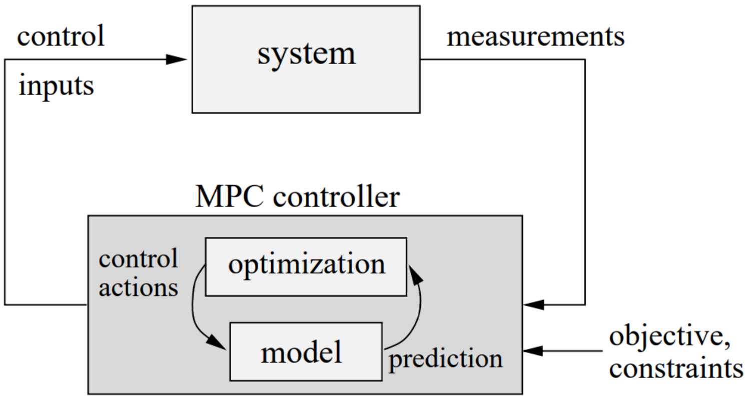


Figure 2.10: A general visualization depicting the concept of a Model Predictive Controller. [26]

As described before we will be making use of a simpler representation of the system in our MPC controller. In this study, we use *system identification*, which is a widely used technique in control to capture the relation between the states of the system we want to control and the control input imposed on the system. This can be done by experiments of the system or using data generated from detailed simulations, in our case from the FE model. Specifically, we collect time-series data of the system's states and the corresponding open-loop control inputs. As depicted in Fig. 2.10 we have the actual true system we want to control and the MPC controller which contains a model of the true system. It is important to understand that the actual *exact* dynamics of the true system are unknown and that the model in the MPC controller is an approximation of reality which we find using the identified reduced order system from the data.

2.5.1. True system

The true system in this study may be confusing and therefore we need to distinguish two type of ”true systems”.

There is the actual true system, which in practice would be a real-life floating structure. For obvious reasons we replace this real-life floating structure with a Finite Element model which takes as much of the actual dynamics into account. In other words, the Finite Element model acts as a very high-fidelity model of a real-life flexible floating structure. For this study we consider the Finite Element model flawless, taking most of the complex dynamics of a real-life floating structure into account.

Then there is our (data-driven) model which should describe the dynamics of this true system, read Finite Element model, as much as possible. Moreover, since we are only interested in the control of the floating structure we limit our state-space description by the consideration of only the structure deformation. Hence, the state we consider from the true system, which is the FE model as described before, can be assembled as $\mathbf{x} = \boldsymbol{\eta}$. We assume that the dynamics of the floating structure behave in the following canoncial form

$$\mathbf{x}_{k+1} \approx A\mathbf{x}_k + B\mathbf{u}_k + \mathbf{v}_k \tag{2.9}$$

$$\mathbf{y}_k = D\mathbf{x}_k + \mathbf{w}_k \tag{2.10}$$

where $\mathbf{x}(k) \in \mathbb{R}^n$, $\mathbf{u}(k) \in \mathbb{R}^q$ and $\mathbf{y}(k) \in \mathbb{R}^p$ represent the state, control action and the measured outputs respectively [16]. The state-space matrices A and B will be obtained by the use of *system identification* as mentioned before. Since it is practically impossible to know the exact system's behaviour, we add the process noise $\mathbf{v}_k \in \mathbb{R}^n$ to Eq. 2.10. We assume this process noise to be white noise with a zero mean and a covariance matrix $\mathbb{Q}^{n\times n}$ [6]. The matrix D in Eq.

2.10 resembles a selection matrix such that our measurements are a subset of the entire state of the system. Moreover, for completeness, we assume that our measurements obtained from the probes may be noisy. Therefore, we define the so-called measurement noise as $\mathbf{w}_k \in \mathbb{R}^p$, which again is assumed to be white noise with zero mean and a covariance matrix $\mathrm{R}^{p\times p}$ [6]. In control we often deal with limited measurements which are a function of the true state of the system. It is important to realize that the Eqs. 2.9 and 2.10 are the assumed dynamics of the system as depicted in Fig.2.10. Let us therefore derive the model in the MPC controller which will be used for the computation for the control input that will act on the true system.

2.5.2. Observation (MPC) model

As mentioned before we do not have knowledge about the true state of the system we opt to control. In other words, we have only access to sparse measurements obtained from the true system from which we need to compute a control input, as depicted in Fig. 2.10. Hence, we base our control input on an *estimated state* which is the state of the system according to the controller.

In the field of control this is often called the *observation model*. We define this estimated state by $\hat{\mathbf{x}}$. Moreover, having this estimated state we can then compute the estimated measurements denoted by $\hat{\mathbf{y}}$ using the relation in Eq. 2.12. Let us now define the used model in the MPC controller:

$$\hat{\mathbf{x}}_{k+1} \approx A\hat{\mathbf{x}}_k + B\mathbf{u}_k + K_k(\mathbf{y}_k - \hat{\mathbf{y}}_k) \tag{2.11}$$

$$\hat{\mathbf{y}}_k = D\hat{\mathbf{x}}_k \tag{2.12}$$

$$\hat{\mathbf{p}}_k = C\hat{\mathbf{x}}_k \tag{2.13}$$

Note that we added a so-called observer gain K_k which serves as an adaptive matrix that tries to minimize the difference between the actual measurement $\mathbf{y}_k$ and the estimated measurements $\hat{\mathbf{y}}_k$. The idea behind this is that by minimizing the error between the actual measurement and the estimated measurements there should be convergence of the estimated state $\hat{\mathrm{x}}$ to the actual state $\mathbf{x}$. The derivation of the observer gain is dependent on the covariance matrices Q and R. Moreover, we expand our system with the control variable $\hat{\mathbf{p}}_k$, which is the signal we want to follow a reference trajectory $\mathbf{r}$. This will be derived later in the thesis.

2.6. Optimization techniques

One of the key-elements in Model Predictive Control is the optimization technique to find the optimum control that maximizes the objective function. The type of optimization is essential in determining the speed of the controller. However, the optimization procedure heavily relies on the type of the model. There are two type of MPC methods that should be considered: linear and non-linear MPC. Each type requires different optimization techniques.

2.6.1. Linear Model Predictive Control (LMPC)

We speak of a system that can be solved using a Linear Model Predictive Controller when we have a linear model of the system without any inequality constraints and has a quadratic performance index of the following form [6]

$$J(u,k) = \sum_{j=0}^{N-1} \hat{z}^T(k+j|k)\Gamma(j)\hat{z}(k+j|k) \tag{2.14}$$

where the vector $\hat{z}(k+j|k)$ denotes the prediction of $z(k+j)$, which is a signal we want to minimize, at time-instant k. The matrix Γ is a selection matrix that distinguishes the predicted tracking error and predicted control effort terms found in standard performance indices, such as the *Generalized Performance Index (GPC)* and *Linear Quadratic Performance Index (LQPC)* [6]. In unconstrained or equality-constrained cases we can often find a closed-form solution without iteratively conducting an optimization problem. This is favourable due to the fact that we can instantaneously return a control input by finding a control law [6]. In such cases, we speak of a LMPC. However, this is in most cases not possible since complex systems are almost always constrained by inequality conditions, such as physical ones. Moreover, this approach is limited to quadratic performance indices limiting us to these type of objective functions.

2.6.2. Non-linear Model Predictive Control (NMPC)

Non-linear Model Predictive Control is more common in practice but requires more sophisticated optimization algorithms that are non-linear, i.e. we are not able to formulate a closed-form control law that return the optimum control input without iterating. Non-linear Model Predictive Controllers are required when one of the following is not satisfied [6]:

1. The optimization problem is convex, hence guarantying a unique solution.
2. In case of constraints, they should be solely equality constraints.

The biggest disadvantage that arises for these kind of optimization problems is the time-consuming procedure while solving. This can be a major problem, especially if the considered system requires high-frequency control. Since the optimization has to be carried out on-line this can be a condition that is hard to satisfy. Let us consider two non-linear cases: *inequality constrained quadratic programming (QP) problems* and *general inequality constrained problems*.

Inequality Constrained QP Problem

The *quadratic programming problem* can be defined as follows [6]

$$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta}^T H \boldsymbol{\theta} + \boldsymbol{\theta}^T f + c \tag{2.15}$$
$$s.t. \quad A\boldsymbol{\theta} \leq \boldsymbol{b}$$

which can be solved by the use of convex optimization algorithms such as [6]

1. *The modified simplex method*
2. *The interior point method*

General Inequality Constrained Problems

A *general inequality constrained problem* is where the objective function is any user-defined function not necessarily of the quadratic form as described in Eq. 2.15. Hence, this general problem can be defined as follows

$$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \tag{2.16}$$
$$s.t. \quad A\boldsymbol{\theta} \leq \boldsymbol{b}$$

which can be solved by the use of so-called *non-linear programming (NLP)* algorithms. For *NLP* we find two types that are dominantly distinguished between in literature. The first type are the so-called *gradient-based* optimization algorithms, where the gradient of the objective function with respect to the control input should be analytically or numerically defined. In case the gradient is numerically defined these algorithms tend to be relatively slow. The second type of algorithms are the so-called *(meta)-heuristic* optimization algorithms, also known as *gradient-free* optimization algorithms. These algorithms as the name indicates do not require gradient information unlike the *gradient-based* algorithms.

One of the main draw backs of *NLP* algorithms are their iterative nature which makes on-line controlling a challenge. However, because in MPC we predict future outcomes in each optimization step we can make use of a smart trick to reduce the computational time significantly. By setting a time-constraint to our optimization algorithm we initially will not find our optimum control-input in the first few steps of control. However, we can still implement the *best* control input so far and use this control input as initialization for the next optimization. This is what is called *warm-start* in the literature [17]. This trick is built around the assumption that the objective function at $k+1$ is in most cases very similar to the objective function at time-instant k. This will eventually require more time for the system to converge to the reference-trajectory but enables on-line optimization at relatively high speeds.

Key takeaway

To ensure a successful study we considered and investigated the following five key topics from the literature:

1. **Modeling of flexible floating structures**:
   - **CFD-FEA**: Combines CFD and FEA to capture detailed nonlinear fluid-structure interactions. Computationally demanding, making it suitable for high-fidelity ”plant” models.
   - **Two-way decoupled FSI**: Simplifies modeling by decoupling fluid and structural interactions, using linear hydro-elastic theory. Efficient but lacks added-mass effects.
   - **Two-way monolithic FSI**: Balances complexity and efficiency, capturing wet-mode dynamics without complex nonlinear wave behavior.
2. **Control strategies for high-dimensional systems**:
   - Inspiration from high-dimensional systems like soft robots, where ROMs manage large DoFs.
   - Key control inputs:
     - **Boundary control**: Actuators reduce edge vibrations, with passive damping needed for central stability.
     - **Stiffness-adjustable springs**: Controls structure with low power by adjusting hinge stiffness.
     - **Mooring control**: Underexplored due to nonlinearity in real mooring behavior, with limited data on performance.
3. **Reduced Order Modeling (ROM) techniques**:
   - **RNN-MPC**: Uses neural networks to reduce model complexity but can lack interpretability.
   - **SINDy auto-encoder**: Identifies sparse governing equations for interpretable ROMs.
   - **DMDc**: Extracts dynamic modes for efficient reduced-order modeling of fluid-structure interactions.
4. **Model Predictive Control (MPC)**:
   - MPC optimizes control inputs based on current state predictions, with a ROM to approximate real system dynamics.
   - System identification approximates complex system behavior from simulation or measurement data.
5. **Optimization techniques in MPC**:
   - **Linear MPC (LMPC)**: Solves efficiently for linear, unconstrained systems using closed-form solutions.
   - **Non-linear MPC (NMPC)**: Handles complex constraints via iterative methods, with warm-start initialization for faster computation.

3 Theoretical Framework

In this chapter, we present the theoretical framework for the control system designed to manage flexible floating structures. First, we introduce the control system framework, highlighting its main components and their interactions. We will briefly explain the role of each component and how they contribute to the overall system. Following this, we will discuss each component in more detail, addressing the assumptions, limitations, and dependencies on other elements within the framework. It is important to note that this chapter is written as generally as possible to provide a flexible control framework applicable to various objectives and types of floating structures. In the methodology chapter we will present a possible implementation for a considered case.

3.1. Control system framework

The control framework is depicted in Fig. 3.1 where we have two main components: the MPC and the true system which we want to control. The MPC is our workhorse where most of the control takes place. The true system is, in general, any given dynamical system we can control or stabilize to our desire by the use of our MPC. Let us recall the concept of MPC: MPC operates by continuously **predicting** the future behaviour of the true system within a predefined **horizon** and adjusting the control inputs based on these predictions. The primary objective is to ensure that the system follows a desired reference trajectory while satisfying system constraints and mitigating the effects of disturbances. Below we discuss these two main components along with the feedback mechanism that allows communication between these two components.

3.1.1. True system

The true system in this framework represents the flexible floating structure modeled using the Finite Element model. This high-fidelity model simulates the dynamic behaviour of the structure in response to the control signal $\mathbf{u}_k$ generated by the optimization block. One of the main challenges we encounter is the external forcing due to the incoming wave found in offshore conditions. This forcing is much more significant and cannot be considered as an external disturbance, i.e. process noise, since it determines to a great extent the dynamics of the system. In the context of floating structure we can deal with this challenge using the diffraction theory, which will be discussed later on in the report.

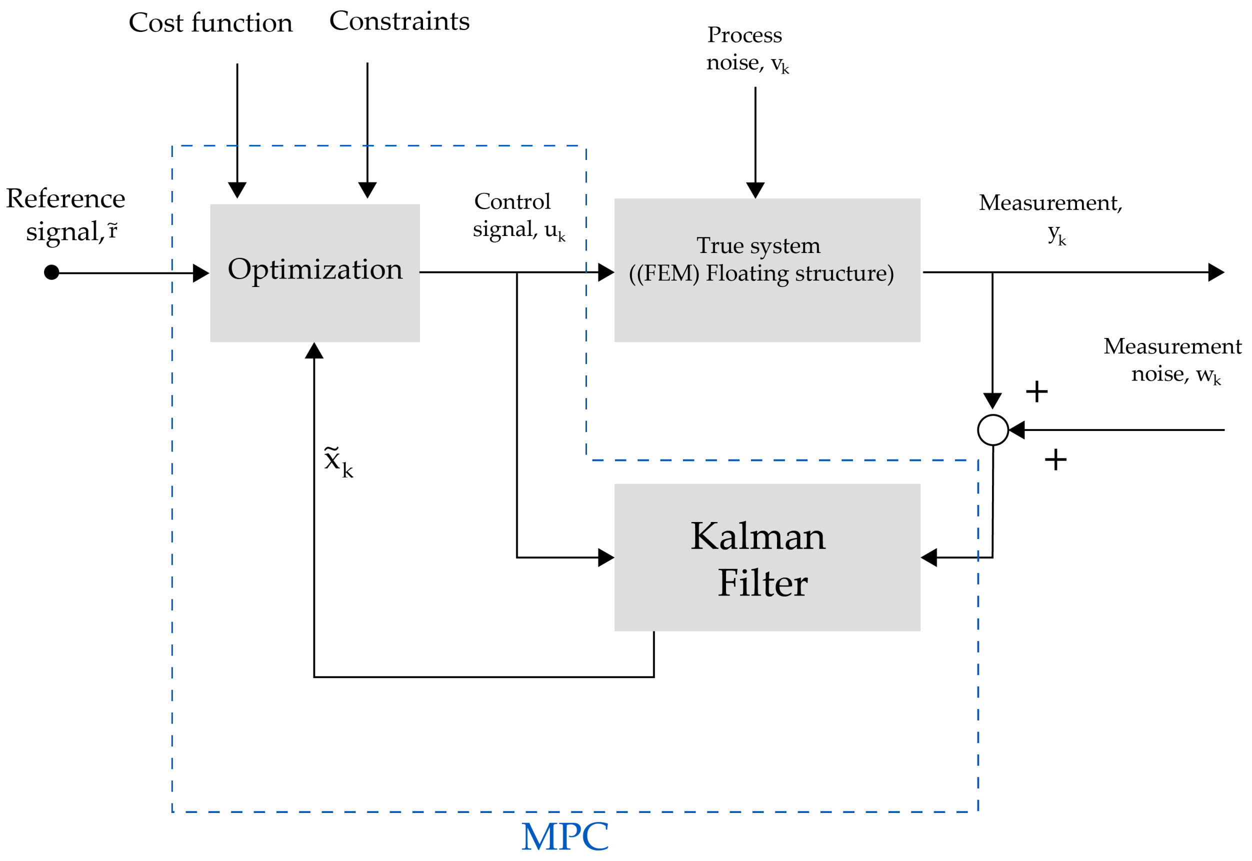


Figure 3.1: General depiction of the MPC in combination with the true system. [43]

3.1.2. MPC framework for floating structures

The MPC can be broken down into two components: the so-called Kalman filter and the optimization block.

Kalman filter

The Kalman filter is the first component in the MPC that interacts with the true system while controlling.

For control we need to know the current state of the true system to find a control input. One can imagine that at the moment we start to control we first obtain measurements from the true system. In practice we take sparse measurements using sensors from the true system. As mentioned before, these measurements are polluted with measurement noise $\mathbf{w}_k$.

Along with the measurements, we can also run a parallel simulation of the true system to estimate the current state. We can do this using the previous state and previous control input such that

$$\hat{\mathbf{x}}_{k|k-1} \approx A\hat{\mathbf{x}}_{k-1|k-1} + B\mathbf{u}_{k-1}. \tag{3.1}$$

We call this estimation of the state the *prior state estimate*. However, even by doing this we will not be able to know the actual true state of the system because we do not know the amount of process noise $\mathbf{v}_k$.

That's when the Kalman filter comes into play by combining the sparse measurements obtained directly from the true system with the prior estimate of the state.

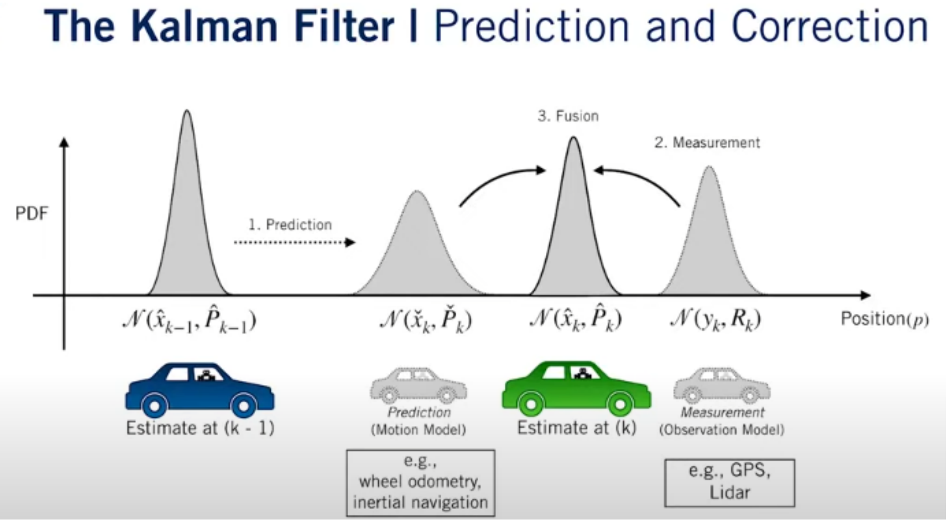


Figure 3.2: The Kalman Filter is a method used to improve estimates of a system's state (like the position of a car) by combining predictions and measurements. First, it predicts the current state based on the previous estimate and information from the car's movement (using a model). Then, it corrects this prediction by using new measurements from sensors (like GPS or Lidar). Since both the prediction and the measurements have some uncertainty or ”noise,” the Kalman Filter combines them to get a better, more accurate estimate of the car's current position. [1]

Because of the process and measurement noise defined in the description of our model we find ourself working with stochastic variables and hence Probability Density Functions (PDFs). In Fig. 3.2 we see how the Kalman filter predicts the current state using the previous state and then corrects this prior estimate with the measurements from the true system to obtain a *posterior state estimate* that is even more accurate.

This filtering process minimizes the effect of measurement errors and uncertainties, providing more accurate information about the system's current state. This posterior state estimate is then used by the optimization block to find the control action applied to the true system. Note that in Fig. 3.1 we feed $\tilde{\mathrm{x}}$ to the optimization block, which is the posterior state estimate but in reduced space for computational efficiency reasons. This will be clearly explained later on in the report.

Optimization block

The optimization block represent the core of the control system. At each time step, it computes the optimal control input, $\mathbf{u}_k$, by solving an optimization problem that balances the system's performance with its physical constraints. The control input is determined based on a reference trajectory $\mathbf{r}$, which defines the desired outcome (e.g., the optimal movement or position of the floating structure). The optimization also takes into account a predefined cost function that measures the performance of the system, ensuring that the control actions lead to the best possible outcome while respecting the system's physical limitations.

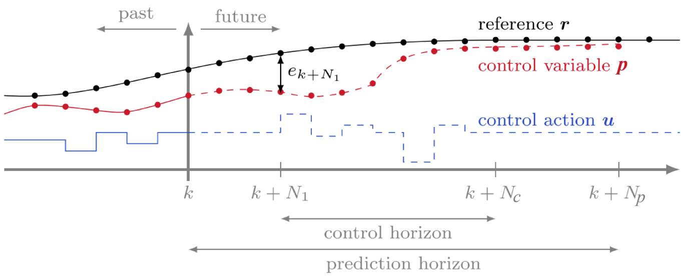


Figure 3.3: This figure illustrates the concept of Model Predictive Control (MPC), a control strategy that involves predicting the future behavior of a system and making decisions at the present moment to influence future outcomes. It shows how, at the current time step k, MPC uses a model to predict future outputs (control variable p) over a prediction horizon, and determines the optimal control actions (control action u) to apply over a control horizon, such that the control variable p follows a desired path (reference r). [29]

3.1.3. Feedback loop

The MPC framework operates as a continuous feedback loop. The control signal $\mathbf{u}_k$ generated by the optimization block is applied to the true system, which responds by producing a measurable output $\mathbf{y}_k$. Due to environmental factors and noise, this measurement is often noisy. The Kalman filter processes this measurement, filtering out noise to estimate the system's true state. This estimated reduced order state $\tilde{\mathbf{x}}_k$ is then fed back into the optimization block, where the next control signal is calculated. This loop enables the system to dynamically adjust its actions in real-time, responding to both the desired trajectory and the changing environment.

Key takeaway

We presented a general depiction of the considered framework such that:

1. **Control framework**:
   - The framework consists of an MPC and a high-fidelity model (true system), with MPC predicting system behavior and adjusting control inputs over a set horizon.
2. **True system**:
   - A Finite Element model of a flexible floating structure, with external wave forces and Fluid-Structure Interaction.
3. **MPC components**:
   - **Kalman filter**: Refines noisy measurements and prior estimates for accurate state feedback.
   - **Optimization block**: Computes control inputs to follow a reference trajectory while respecting constraints.
4. **Feedback loop**:
   - Real-time adjustments via continuous feedback, enabling dynamic response to changing conditions and goals.

3.2. Problem setting

In this section we lay the foundation for the problem setting. The problem setting describes the considered physics in detail to describe a flexible floating structure in offshore conditions. From the derived PDE-problem including the proper boundary conditions we will formulate a Finite Element Model in order to solve this problem using a numerical approach, which will be presented in detail later on in the methodology chapter. Important to note that this Finite Element Model will act as the ”plant”, i.e. the real-world equivalent of the system, we opt to control.

In this section we will zoom in on the **true system block** depicted in Fig. 3.4. In order to reach a control framework that is as general as possible we decide to only be concerned with the structure deformation as our observable state, that is we take $\mathbf{x} = \boldsymbol{\eta}$. Hence, we assume we do not have any **direct knowledge** about the (incident) excitation wave. This approach can be justified from the diffraction theory, where the total wave field around the structure can be decomposed into an incident, and transmitted, wave and an reflected wave.

Because of the complex dynamics which lead to high-fidelity of the considered model in comparison to real-world data, there is no analytical solution which returns the deformation of the floating structure in a two-way interaction with the surrounding fluid. Usually, when there is an analytical solution to the considered PDE-description of the model, one is able to directly implement the analytical solution in the MPC for optimization and forecasting purposes to find a control signal. However, the implementation of a numerical model requires expensive computations and therefore lead to infeasible control frequency, which in practice should be as high as possible to keep up with the dynamics of the system.

In order to control the floating structure we need to find a simpler representation of the Finite Element Model which we can directly implement in the MPC to use for optimization and forecasting computations. As depicted in Fig. 3.4 we assume our true system to behave according to a linear system where the state of the system at time-instant $k+1$, $\mathbf{x}_{k+1}$, is linearly dependent on the state of the system at k, $\mathbf{x}_k$, and the acted control input at time-instant k, $\mathbf{u}_k$. We will use the DMDc to find the system matrices A and B which explain the dynamics of the system in full-space. However, DMDc additionally is able to return a ROM by finding $\tilde{A}$ and $\tilde{B}$. Because of its data-driven nature, we will discuss the convergence of these system matrices and investigate their generalizing ability for different sea states. Moreover, we take the presence of process noise $\mathbf{v}_k$, which is a zero-mean Gaussian noise. Finally, we measure the response of the system by taking sparse measurements along the span of the floating structure, denoted as $\mathbf{y}_k$, which is fed to the MPC.

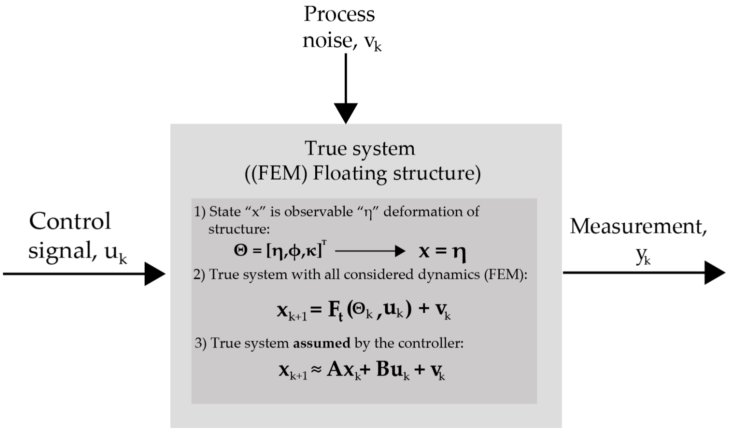

Process
noise, v_k
True system
((FEM) Floating structure)
1) State “x” is observable “η” deformation of structure:
$\Theta = [\eta,\phi,\kappa]^T \longrightarrow x = \eta$
2) True system with all considered dynamics (FEM):
$x_{k+1} = F_t(\Theta_k, u_k) + v_k$
3) True system **assumed** by the controller:
$x_{k+1} \approx Ax_k + Bu_k + v_k$
Control
signal, u_k
Measurement,
y_k

Figure 3.4: Zoomed-in block of the true system depicted in Fig. 3.1. The true system is represented by the high-fidelity FEM polluted with process noise to mimic a realistic scenario. However, the true system is assumed to behave according to a data-driven simple, linear model capturing only the deformation of the structure. The difference between the FEM and the data-driven model is a *model-mismatch*.

3.2.1. Physics background

In this study we use the same setting of a thin floating structure with internal joints as proposed by [33]. Here we take the external moments on the joints as our control variables, where we denote the joints by Λ_j. We model this problem by considering a fluid domain Ω which has the following four boundaries: the inlet boundary Γ_{in}, the free-surface of the fluid Γ_{fs}, the outlet boundary Γ_{out} and the bottom surface Γ_b. On the free surface of the fluid domain we find the floating structure which has two upper and lower ends denoted by $\Gamma_{fs,str}$. Since we are considering the monolithic formulation of a floating structures proposed by [33] we will be dealing with *linear potential flow theory* and *linear beam theory*. Hence the following assumptions should be made:

Assumption 1 The flow within the fluid-domain Ω is considered to be incompressible, inviscid and irrotational. This corresponds to the Linear Potential Flow Theory.

Assumption 2 The VLFS is considered to be homogeneous and shows linear behaviour.

Assumption 3 The transverse displacement of the structure and steepness of the incident waves are considered small (Airy wave theory), and there is no air gap between structure and fluid.

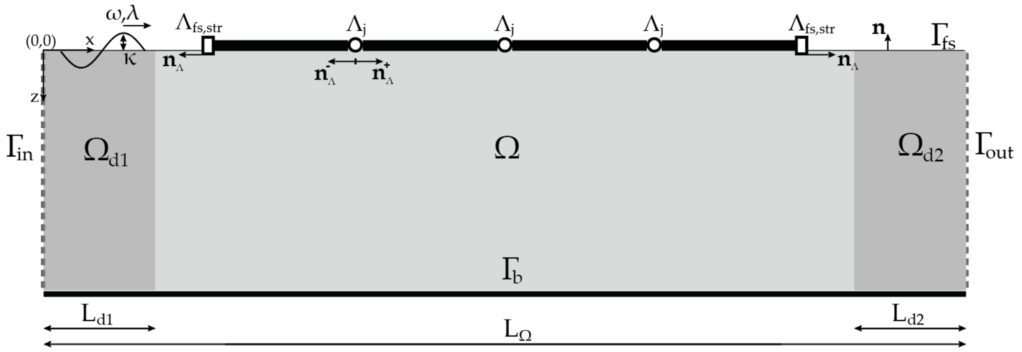


Figure 3.5: The problem setup with stiffness-adjustable joints connecting the modules of the floating structure. [33]

As we can see from figure Fig. 3.5 there are two main components we have to deal with because of the fluid-structure interaction. The first component is the fluid behaviour. The second component is the structure behaviour. By finding the corresponding boundary and interface conditions we can consider the FS-interaction. Let us therefore divide the problem into two sections: *Linear potential flow theory* and *Euler-Bernoulli beam theory*.

3.2.2. Linear potential flow theory

The first assumption we made states that the fluid is incompressible, inviscid and irrotational. Let us define the fluid velocity as $\boldsymbol{u}$, which is a vector containing the velocity components in the space of the fluid domain, that is $\Omega \to \mathbb{R}^d$, where $d = 2$ in this study. Because of incompressibility we know that

$$\nabla \cdot \boldsymbol{u} = 0 \quad \text{in} \quad \Omega \tag{3.2}$$

Now let us define a scalar potential field, say $\phi : \Omega \to \mathbb{R}$. One can prove that if we take the following definition

$$\boldsymbol{u} = \nabla\phi \quad \text{in} \quad \Omega \tag{3.3}$$

this will satisfy the assumption that the fluid flow is inviscid and irrotational. Hence, we can now obtain our governing equation that describes the linear potential flow in Ω. Let us combine Eq. 3.2 and Eq. 3.3 such that

$$\Delta\phi = 0 \quad \text{in} \quad \Omega \tag{3.4}$$

Given this governing equation we need appropriate boundary conditions on the domain boundaries. We define the vector $\mathbf{n}$ as an outward pointing vector normal to the boundaries of the domain Ω, as depicted in Fig. 3.5. Furthermore, let us distinguish two quantities on the top boundary: the surface elevation, κ, and the structure deformation η. For convenience, we use the notations $(\cdot)_t$ and $(\cdot)_{tt}$ denoting the first and second time derivatives respectively. We can now present the following kinematic boundary conditions

$$\mathbf{n} \cdot \nabla \phi = 0 \quad \text{on} \quad \Gamma_b \tag{3.5}$$

$$\mathbf{n} \cdot \nabla \phi = u_{in} \quad \text{on} \quad \Gamma_{in} \tag{3.6}$$

$$\mathbf{n} \cdot \nabla \phi = u_{out} \quad \text{on} \quad \Gamma_{out} \tag{3.7}$$

$$\mathbf{n} \cdot \nabla \phi = \kappa_t \quad \text{on} \quad \Gamma_{fs} \tag{3.8}$$

$$\mathbf{n} \cdot \nabla \phi = \eta_t \quad \text{on} \quad \Gamma_{str} \tag{3.9}$$

Eq. 3.5 simply arises due to the modeling choice we make where we say that the flow cannot penetrate the boundary. Since we want to model a floating structure in offshore conditions it is not convenient, or even possible for computational reasons, to take the domain as large as the offshore environment. Hence, we use a domain limited in size where we impose a prescribed inlet and outlet normal velocities, mimicking the offshore environment as can be read from Eq.3.6 and Eq.3.7. Eqs. 3.8 and 3.9 arise due to our assumption that the steepness of the incident waves are considered small on the free surface and on the boundary between the fluid and structure. Therefore, the velocity of the fluid normal to the free surface can be well approximated by the vertical velocity of the free surface elevation, κ_t.

Lastly, let us consider the dynamic boundary condition where we describe the pressure at the free surface. We are allowed to state that the pressure at the free surface is equal to the atmospheric pressure. Hence, using the linearized Bernoulli equation

$$p \simeq -\rho_w \phi_t - \rho_w g \kappa \quad \text{on} \quad \Gamma_{fs} \tag{3.10}$$

we obtain the following dynamic boundary condition where ρ_w and g are the fluid density and gravitational acceleration respectively.

$$\rho_w \phi_t + \rho_w g \kappa = 0 \quad \text{on} \quad \Gamma_{fs} \tag{3.11}$$

3.2.3. Euler-Bernoulli beam theory

In this study we will use the Euler-Bernoulli beam theory to model our floating structure. We model this Euler-Bernoulli beam as a 1-dimensional Euler-Bernoulli beam floating on the free surface of the 2-dimensional fluid domain, as can be seen in Fig.3.5. Under assumption 3, we are allowed to consider the free surface elevation and the structure deformation as a single quantity whose behavior changes depending on the region, whether that is Γ_{fs} or Γ_{str}. This has similarly been done in some of the formulations found in [33]. However, as described before, we consider the structure deformation η as a separate quantity from the free surface elevation κ. Now let us recall the well-known Euler-Bernoulli beam equation.

$$\rho_b h_b \eta_{tt} + D \Delta^2 \eta = p \quad \text{on} \quad \Gamma_{str} \tag{3.12}$$

Eq. 3.12 relates the structure deformation η to the external pressure p applied on the beam. Furthermore, we have that ρ_b represents the beam's density, h_b represents the structure height and D is the-so called structural rigidity given as $D \overset{\text{def}}{=} Eh_b^3/12$ [33]. The assumption that there is no air gap between the free surface and the floating structure means that the surface elevation κ under the structure has to be equal to the structure deformation η. Hence, from Eq. 3.10, the pressure between the floating structure and the free surface can be written as follows:

$$p \mid_{\Gamma_{str}} = -\rho_w \phi_t - \rho_w g \eta \tag{3.13}$$

Let us now equate Eq. 3.12 with Eq. 3.13 to obtain the following coupling boundary condition between the fluid and structure

$$\frac{\rho_b h_b}{\rho_w} \eta_{tt} + \frac{D}{\rho_w} \Delta^2 \eta + \phi_t + g\eta = 0 \quad \text{on} \quad \Gamma_{str} \tag{3.14}$$

Lastly, we have to define the boundary conditions acting on the ends of the structure and the interface conditions that arise due to the internal joints. We consider the floating structure to have free ends. Hence, the moment, M, and shear force, V, acting on the ends are zero.

$$M \overset{\text{def}}{=} D\Delta\eta = 0 \qquad \text{on} \quad \Lambda_{fs,str} \tag{3.15}$$

$$V \overset{\text{def}}{=} \nabla(D\Delta\eta)\cdot\mathbf{n}_\Lambda = 0 \qquad \text{on} \quad \Lambda_{fs,str} \tag{3.16}$$

We consider three interface conditions for each internal joint. However, since these internal joints will be connecting separate modules, discontinuities in the rotation between the modules will arise. Therefore, we introduce two important operators to make sure that the interface conditions stay continuous between the modules. The jump operator, $\|\cdot\|$, takes the difference between two discontinuous quantities at a given interface. The mean operator, $\langle\cdot\rangle$, take the mean between two discontinuous quantities at a given interface [33].

The first interface condition is by simply relating the relative rotation between the modules of the Euler-Bernoulli beam to the mean moment at that joint. This moment arises due to the linear rotational spring that has a spring constant $k_{\varphi,j}$, where j denotes the individual springs. Additionally, we introduce our control input acting on the joint, which is an external moment. The second and third interface condition state that since there are no external shear forces or moments acting on the joints we must have continuous shear forces and moments at these joints.

$$\langle D\Delta\eta\rangle = -k_{\varphi,j}\|\nabla\eta\cdot\mathbf{n}_\Lambda\| + M_{\text{ext}} \qquad \text{on} \quad \Lambda_j \tag{3.17}$$

$$\|D\Delta\eta\mathbf{n}_\Lambda\| = 0 \qquad \text{on} \quad \Lambda_j \tag{3.18}$$

$$\|\nabla(D\Delta\eta)\cdot\mathbf{n}_\Lambda\| = 0 \qquad \text{on} \quad \Lambda_j \tag{3.19}$$

3.2.4. Damping zone

A modeling problem that arises using the current formulation is the accumulation of energy close to the inlet and outlet boundaries. In offshore conditions reflected and transmitted waves radiate away from the structure until their energy gets dissipated. Hence, we need to mimic this dissipation of energy by introducing so-called damping zones as can be seen in Fig. 3.5. Because we are considering the Airy wave theory we are allowed to decompose the velocity potential ϕ and the free surface elevation κ as follows

$$\phi = \phi_I + \phi_R + \phi_T \tag{3.20}$$

$$\kappa = \kappa_I + \kappa_R + \kappa_T \tag{3.21}$$

where the I stand for *incident*, R stands for *reflected* and T stands for *transmitted*. The velocity potential and the free surface elevation as the entities of these three parts together is an unknown we need to solve for. That is because the reflected quantities are dependent on the interaction of the fluid with the structure. However, the incident velocity potential ϕ_I and the incident free surface elevation κ_I is something we do know since we induce these quantities on the system to mimic incoming waves. From linear wave theory we can solve Eq. 3.4 assuming no structure. We then find the solution $\phi = \phi_I$ and $\kappa = \kappa_I$ as [33]

$$\phi_I((x,z),t) = \frac{\kappa_0\omega}{k_\lambda}\frac{\cosh(k_\lambda z)}{\sinh(k_\lambda H)}\sin(k_\lambda x - \omega t) \tag{3.22}$$

$$\kappa_I(x,t) = \kappa_0\cos(k_\lambda x - \omega t) \tag{3.23}$$

where κ_0 is the induced free surface elevation at Γ_{in}, ω is the wave frequency, κ_λ is the wave number and H is the depth of the fluid domain Ω. Let us know recall Eq. 3.6 and 3.7 where we define our inlet and outlet velocities using Eq. 3.22

$$\mathbf{n}\cdot\nabla\phi = -\kappa_0\omega\frac{\cosh(k_\lambda z)}{\sinh(k_\lambda H)}\cos(k_\lambda x - \omega t) \qquad \text{on} \quad \Gamma_{in} \tag{3.24}$$

$$\mathbf{n}\cdot\nabla\phi = 0 \qquad \text{on} \quad \Gamma_{out} \tag{3.25}$$

Hence, we have an prescribed inlet velocity at Γ_{in} and zero outlet velocity on Γ_{out}. As mentioned before, we are still dealing with the energy accumulation due to the *reflected* and *transmitted* waves near the inlet and outlet boundaries. Hence, we want to damp the following quantities near Γ_{in} and Γ_{out}

$$\phi_R + \phi_T = \phi - \phi_I \tag{3.26}$$

$$\kappa_R + \kappa_T = \kappa - \kappa_I \tag{3.27}$$

Let us now adjust Eq. 3.8 and 3.11 where we implement the damping terms as proposed by [30] and validated by [33]

$$\mathbf{n} \cdot \nabla\phi = \kappa_t + \mu_2(\kappa - \kappa_I) \quad \text{on} \quad \Gamma_{fs} \tag{3.28}$$

$$\phi_t + g\kappa + \mu_1(\mathbf{n} \cdot \nabla(\phi - \phi_I)) = 0 \quad \text{on} \quad \Gamma_{fs} \tag{3.29}$$

with

$$\mu_1(x) = \begin{cases} \mu_0 \left[1 - \sin\left(\frac{\pi x}{2L_{d1}}\right)\right] & \text{if } x < L_{d1}, \\ \mu_0 \left[1 - \cos\left(\frac{\pi(x-x_d)}{2L_{d2}}\right)\right] & \text{if } x > L_\Omega - L_{d2}, \\ 0 & \text{otherwise}, \end{cases}$$

$$\mu_2(x) = k_\lambda \mu_1(x).$$

Key takeaway

We explained the considered model and introduced briefly our strategy to find a Reduced Order Model which we implement in the controller. We discussed the following four main points:

1. **Problem setting overview**:
   - We model a flexible floating structure in offshore conditions, focusing on structure deformation as the observable state.
   - The derived Finite Element Model (FEM) serves as the high-fidelity ”plant” for control, representing real-world dynamics.
2. **Simplified control model**:
   - Due to the FEM’s complexity, a simpler model is needed for the controller. Using DMDc, we approximate the system’s dynamics for efficient use in MPC.
   - The DMDc method will provide system matrices for a (reduced-order) linear model. These system matrices will be examined using model convergence.
3. **Physics background**:
   - The model follows assumptions based on Linear Potential Flow Theory and Linear Beam Theory, ensuring computational feasibility.
   - External moments at joints serve as control inputs, allowing for flexibility in managing structure dynamics.
4. **Fluid-Structure Interaction (FSI) components**:
   - **Fluid dynamics**: Assumed incompressible, inviscid, and irrotational fluid, governed by linear potential flow theory.
   - **Structure dynamics**: Modeled as an Euler-Bernoulli beam interacting with fluid forces and joint moments.
   - Boundary and interface conditions ensure realistic FSI, with damping zones near boundaries to manage energy buildup.

3.3. DMD with control (DMDc)

In this section we explain one of the most important tools to reach our research objective: the Dynamic Mode Decomposition with control. As explained before we have to find a simple linear representation of the FE model by using FE model data to extract the important dynamics. This will be done using DMDc. This powerful method requires extensive investigation to understand its strengths and weaknesses. This will aid us in finding the system matrices A and B such that we in the first place obtain good control performances and in the second place it should enable us to control the floating structure in various sea states.

Furthermore, we will explain the Koopman theory which is the backbone of the DMD(c). There we show how the DMDc and the found system matrices are directly linked to the linear Koopman operator. This will emphasize the power of DMDc and its ability to capture complex nonlinear dynamics through data.

We then precede to discussing the sensitivity of the DMDc to the data and how we can assess the convergence of the obtained system matrices that describe our simplified model.

Finally, we introduce the implementation of DMDc for two cases: regular waves and sea states. Each case requires a different approach. Most importantly, we want to obtain a DMDc model that is as general as possible for all sea states.

3.3.1. Mathematical background of DMDc

As mentioned before we only consider the structure deformation in our state-space formulation. We assume that state-space can be described by the canonical discrete linear dynamical system [23].

$$\mathbf{x}_{k+1} \approx A\mathbf{x}_k + B\mathbf{u}_k \tag{3.30}$$

where we take $\mathbf{x} = \boldsymbol{\eta}$, such that $\mathbf{x}_k \in \mathbb{R}^n$, $\mathbf{u}_k \in \mathbb{R}^q$, $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times q}$. $n \gg 1$ is the total number of Degrees of Freedom associated to the FE function η_h and q is the number of control inputs. Important to realize is that we haven't reached any reduced order model yet by doing this assumption. Similarly to DMD without control we opt to find best-fit approximations for A and B from open-loop simulations using a wide range of control inputs to capture the behaviour more accurate. We can then store the state snapshot measurements in a matrix X and the corresponding control inputs in a matrix Υ as can be seen from Fig. 2.9. We can therefore expand Eq. 3.30 as

$$X' \approx AX + B\Upsilon \tag{3.31}$$

where X' is the one time-step shifted equivalent of X. Let us rewrite Eq. 3.31 into a more compact form

$$X' \approx [A\, B]\begin{bmatrix} X \\ \Upsilon \end{bmatrix} = G\Omega \tag{3.32}$$

where $G \triangleq [A\, B] \in \mathbb{R}^{n\times(n+q)}$ is the augmented matrix containing the operators we opt to find and Ω is the augmented matrix containing the data of the state measurements and the induced control inputs [23]. Our goal is to find the best-fit solution of the operator G by minimizing the so-called Frobenius norm [23]

$$\min_G \|X' - G\Omega\|_F \tag{3.33}$$

This can be done by conducting a Singular Value Decomposition, SVD, on the augmented data matrix such that $\Omega = \tilde{U}\tilde{\Sigma}\tilde{V}^*$. Let us rewrite Eq. 3.32 into

$$G \approx X' \underbrace{\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}^*}_{=\Omega^\dagger} \tag{3.34}$$

where we can split the matrix $\tilde{U}$ consisting of the left singular vectors into two separate matrices $\tilde{U}_1^*$ and $\tilde{U}_2^*$ such that [23]

$$[A,\, B] \approx \left[\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}_1^*,\, \tilde{V}\tilde{\Sigma}^{-1}\tilde{U}_2^*\right] \tag{3.35}$$

where $\tilde{U}_1 \in \mathbb{R}^{n\times\tilde{r}}$ and $\tilde{U}_2 \in \mathbb{R}^{q\times\tilde{r}}$. The truncation order of the SVD is denoted as $\tilde{r}$. In other words, we only consider $\tilde{r}$ singular vectors that capture the largest variance found in our input data Ω, given a certain threshold [23]. Note that we still have not reached a reduced order model. To do so, we perform a second SVD on X' such that $X^{'} = \hat{U}\hat{\Sigma}\hat{V}^*$ where again we only consider r singular vectors capturing the largest variance in our output data, given a certain threshold. We find that $\hat{U} \in \mathbb{R}^{n\times r}$, $\hat{\Sigma} \in \mathbb{R}^{r\times r}$ and $\hat{V}^* \in \mathbb{R}^{r\times m-1}$, where m is the total number of snapshots [23]. Finally, using the state transformation $\mathbf{x} \approx \hat{U}\tilde{\mathbf{x}}$ we find the reduced-order approximations of A and B as

$$\tilde{A} = \hat{U}^* X' \tilde{V} \tilde{\Sigma}^{-1} \tilde{U}_1^* \hat{U} \tag{3.36}$$

$$\tilde{B} = \hat{U}^* X' \tilde{V} \tilde{\Sigma}^{-1} \tilde{U}_2^* \tag{3.37}$$

We can now obtain the ROM having $\tilde{A} \in \mathbb{R}^{r\times r}$ and $\tilde{B} \in \mathbb{R}^{r\times q}$ where $r \leq \tilde{r} \ll n$

$$\tilde{\mathbf{x}}_{k+1} \approx \tilde{A}\tilde{\mathbf{x}}_k + \tilde{B}\mathbf{u}_k. \tag{3.38}$$

where now we consider the reduced states which are represented by the dynamic modes of $\tilde{A}$, instead of the high-dimensional DoFs of the floating structure.

The dynamic modes of A can be determined by solving the eigenvalue problem for the matrix $\tilde{A}$, where the decomposition is expressed as:

$$\tilde{A}W = W\Lambda \tag{3.39}$$

where the matrix Λ contains the eigenvalues of the system matrix A along its diagonal and the matrix W contain the dynamic modes of the operator $\tilde{A}$. Note that the dynamic modes of $\tilde{A}$ are the modes of the reduced system which we will use for the actual control. However, often for visualization reasons we want to visualize these reduced modes in the full-order space.

We can compute the matrix Φ which contains the full-space representation of the dynamic modes along its columns by

$$\Phi = X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}_1^*\hat{U}\tilde{W}. \tag{3.40}$$

Finally, if we want to project any original state $\mathbf{x}$ of the floating structure on the dynamic modes of the operator A, we can compute the corresponding amplitudes of each dynamic mode represented in that state by

$$\mathbf{b} = \Phi^{\dagger}\mathbf{x} \tag{3.41}$$

where the entries of the vector $\mathbf{b}$ contain the amplitudes of the dynamic modes.

Key takeaway

- The DMDc model estimates A and B matrices, which govern the dynamics of structure deformation in response to control inputs.
- Singular Value Decomposition (SVD) is applied to observed data, optimizing the augmented matrix $G = [A\ B]$, capturing key dynamics.
- The state transformation matrix $\hat{U}$ reduces the high-dimensional state $\mathbf{x}$ to a manageable reduced-order state $\tilde{\mathbf{x}}$. Note that all quantities related to the controller which are denoted with a tilde ($\tilde{\cdot}$) are in the reduced-order space.
- The ROM uses reduced matrices $\tilde{A}$ and $\tilde{B}$, describing system behavior in lower dimensions while retaining essential dynamics for efficient control.
- r and $\tilde{r}$ are truncation levels for capturing the largest variances; r is used for the system outputs in X', and $\tilde{r}$ for the augmented input data in Ω.

3.3.2. Koopman operator theory and DMDc

After having presented the DMDc, it is essential to introduce the **Koopman operator theory**, which serves as the mathematical foundation for DMD(c). Understanding the concepts behind Koopman theory will help us to appreciate the strengths and limitations of DMDc. Will explain the Koopman theory for the full order formulation of the system, hence not the reduced order. Nevertheless, the main concept will become clear down the line. The theory below can be found in [35] which gives a complete description.

The Koopman theory is highly effective for complex nonlinear systems because it transforms the nonlinear dynamics into a linear framework by focusing on measurements denoted by $g(\mathbf{x}, \mathbf{u})$ of the system rather than the state itself. Instead of working directly with the nonlinear system, Koopman theory considers how these measurements evolve linearly over time under the action of a linear operator, the Koopman linear operator to be exact. This linear evolution, despite the system's nonlinear internal dynamics, allows us to use linear techniques for prediction and control, making it much more practical to handle complex, real-world systems.

This concept is depicted in Fig. 3.6 where we consider our system as an unknown complex system which evolves in a nonlinear fashion. We only assume that the state of the system is dependent on a previously applied control input and the previous state. We then search for a linear Koopman operator such that if we measure some quantity which is a function of the state and the control input, we can find the subsequent measurement of that quantity by simply applying this linear Koopman operator on the previous measurements.

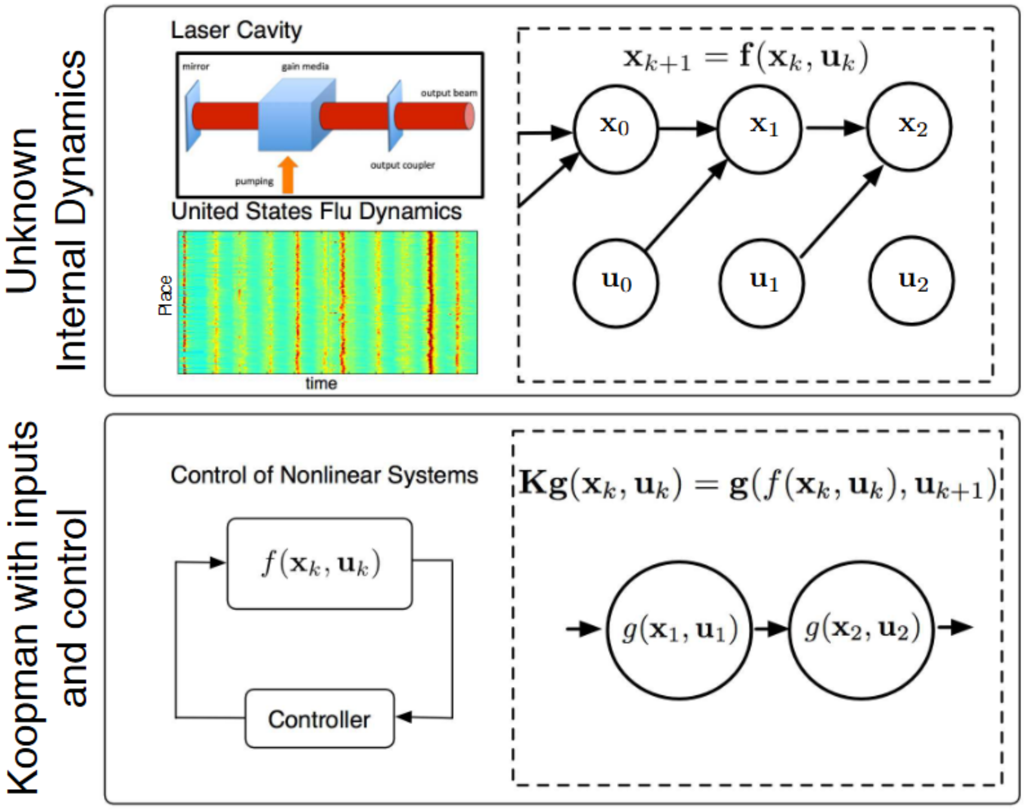

Unknown Internal Dynamics
Laser Cavity
mirror
gain media
output beam
output coupler
pumping
United States Flu Dynamics
Place
time
$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$
$\mathbf{x}_0$
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{u}_0$
$\mathbf{u}_1$
$\mathbf{u}_2$
Koopman with inputs and control
Control of Nonlinear Systems
$f(\mathbf{x}_k, \mathbf{u}_k)$
Controller
$\mathbf{K}\mathbf{g}(\mathbf{x}_k, \mathbf{u}_k) = \mathbf{g}(f(\mathbf{x}_k, \mathbf{u}_k), \mathbf{u}_{k+1})$
$g(\mathbf{x}_1, \mathbf{u}_1)$
$g(\mathbf{x}_2, \mathbf{u}_2)$

Figure 3.6: The figure illustrates the motivation behind developing Koopman operator theory with inputs and control. The first row depicts a complex system evolving over time with unknown underlying dynamics. The second row highlights the primary objective of Koopman operator theory: to identify an operator capable of advancing a set of measurements forward in time for prediction and control, even in the absence of an explicit model of the system. [35]

Hence, the **Koopman operator theory** provides a powerful framework for analyzing nonlinear dynamical systems by ”lifting” the system into an infinite-dimensional space of measurement functions g, where the dynamics become linear. Consider a dynamical system of the form:

$$\mathbf{x}_{k+1} = \mathbf{F}_t(\mathbf{x}_k, \mathbf{u}_k), \tag{3.42}$$

where $\mathbf{x}_k \in \mathbb{R}^n$ is the state of the system at time instant k, $\mathbf{u}_k \in \mathbb{R}^q$ is the control input at time instant k and $\mathbf{F}_t$ is a (potentially nonlinear) function that governs the system's evolution. In our case $\mathbf{F}_t$ is the representation of the complex Finite Element Model of the flexible floating structure. Instead of using a scalar-valued measurement function g, we will consider a vector-valued measurement function $\mathbf{g}$ for our purpose. This will become evident later on in the explanation. The Koopman operator $\mathcal{K}$ acts on each component of the vector-valued measurement function $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}^q \rightarrow \mathbb{C}^m$ as follows:

$$\mathcal{K}\mathbf{g}(\mathbf{x}_k, \mathbf{u}_k) = \mathbf{g}(\mathbf{F}_t(\mathbf{x}_k, \mathbf{u}_k), \mathbf{u}_{k+1}) = \begin{bmatrix} g_1(\mathbf{F}_t((\mathbf{x}_k, \mathbf{u}_k), \mathbf{u}_{k+1})) \\ g_2(\mathbf{F}_t((\mathbf{x}_k, \mathbf{u}_k), \mathbf{u}_{k+1})) \\ \vdots \\ g_m(\mathbf{F}_t((\mathbf{x}_k, \mathbf{u}_k), \mathbf{u}_{k+1})) \end{bmatrix} = \mathbf{g}(\mathbf{x}_{k+1}, \mathbf{u}_{k+1}). \tag{3.43}$$

This linear operator governs the evolution of the measurement $\mathbf{g}$ over time, even though the original system dynamics are nonlinear.

DMDc can be viewed as a practical, data-driven approximation of the Koopman operator. Given a set of snapshot pairs of the state $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{N_s-1}]$, $X' = [\mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_{N_s}]$, and snapshots of an open-loop control signal $\Upsilon = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{N_s-1}]$, DMDc aims to find a best-fit linear operator of A and B that approximates the dynamics:

$$X' \approx AX + B\Upsilon. \tag{3.44}$$

The Koopman operator theory applies to all measurement functions. Let us consider the vector-valued measurement function $\mathbf{g}$ is equal to the state which is an allowed operation since the Koopman theory allows for any measurement function $\mathbf{g}$

$$\mathbf{g}(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} g_1(\mathbf{x}_k, \mathbf{u}_k) \\ g_2(\mathbf{x}_k, \mathbf{u}_k) \\ \vdots \\ g_m(\mathbf{x}_k, \mathbf{u}_k) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix}. \tag{3.45}$$

Applying the Koopman operator to $\mathbf{g}(\mathbf{x}_k, \mathbf{u}_k)$ yields:

$$\mathcal{K}\mathbf{g}(\mathbf{x}_k, \mathbf{u}_k) = \mathcal{K}\begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{u}_{k+1} \end{bmatrix}. \tag{3.46}$$

If we look closely to the linear Koopman operator in Eq. 3.46 one may notice how we can subdivide the Koopman operator into four blocks such that

$$\begin{bmatrix} \mathcal{K}_{11} & \mathcal{K}_{12} \\ \mathcal{K}_{21} & \mathcal{K}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{u}_{k+1} \end{bmatrix}. \tag{3.47}$$

which shows that the Koopman operator can model any nonlinear dynamical system as a infinitely large linear operator acting on the states and control input. Because we are not interested in the control input $\mathbf{u}_{k+1}$ we take $\mathcal{K}_{21} = \mathbf{0}$ and $\mathcal{K}_{22} = \mathbf{0}$ such that we obtain

$$\underbrace{\begin{bmatrix} \mathcal{K}_{11} & \mathcal{K}_{12} \end{bmatrix}}_{\overline{\mathcal{K}}} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} = \mathbf{x}_{k+1}. \tag{3.48}$$

Since the Koopman operator is infinite-dimensional, practical implementations rely on finite-dimensional approximations, such as DMDc. Note the high similarity between Eq. 3.48 and Eq. 3.32. The matrices A and B which we compute from the DMDc approach, approximates the Koopman operators $\mathcal{K}_{11}$ and $\mathcal{K}_{12}$ respectively, providing a finite-dimensional representation.

We now consider the linear Koopman operator $\overline{\mathcal{K}}$ where its eigenvalue problem reads as

$$\overline{\mathcal{K}} b_i = \lambda_i b_i, \tag{3.49}$$

where $b_i(\mathbf{x}, \mathbf{u})$ are the Koopman eigenfunctions. These eigenfunctions form a coordinate system that map the state and control input as a sum of Koopman modes ϕ_i. The system's state can be expanded as:

$$\mathbf{g}(\mathbf{x}) = \sum_{i=1}^{\infty} b_i(\mathbf{x}_k, \mathbf{u}_k)\phi_i. \tag{3.50}$$

Finally, the dynamics evolve linearly via the Koopman operator:

$$\mathbf{x}_{k+1} = \sum_{i=1}^{\infty} \lambda_i b_i(\mathbf{x}_k, \mathbf{u}_k)\phi_i, \tag{3.51}$$

where λ_i are the Koopman eigenvalues describing the evolution of the Koopman modes. In practice, DMDc approximates this infinite sum with a finite number of modes r.

$$\mathbf{x}_{k+1} \approx \sum_{i=1}^{r} \lambda_i b_i(\mathbf{x}_k)\phi_i. \tag{3.52}$$

Note how we can express the evolution of our dynamical system **without control** using a finite linear superposition and the computed quantities in Eqs. 3.39, 3.40 and 3.41. Hence, we map the considered current state $\mathbf{x}_k$ on the dynamic modes Φ, using the corresponding amplitudes computed in $\mathbf{b}$ and the corresponding time dynamics represented by the eigenvalues of Λ.

Very important to note that in Eq. 3.52 we only look at the evolution of our state as a function of the current state $\mathbf{x}_k$ and hence without a control input at time-instant k. That is because the considered modes are the dynamic modes of the matrix A solely and not the dynamic modes of the concatenated matrix $[A\ B]$, which is the case in the Koopman theory as depicted in Eq. 3.48.

Key takeaway

- The Koopman operator theory allow us to transform nonlinear dynamics into a linear framework by operating on measurement functions $\mathbf{g}(\mathbf{x}, \mathbf{u})$ rather than states, enabling linear control of complex systems.
- DMDc provides a practical, finite-dimensional approximation of the Koopman operator, using matrices A and B to capture system dynamics and control inputs.
- System evolution is expressed as a linear superposition of Koopman eigenfunctions and eigenvalues, allowing simplified dynamic predictions without direct control inputs.
- DMDc enables effective control of nonlinear systems by representing them through Koopman eigenvalues (λ) and modes (ϕ) with a finite set of modes.

3.3.3. Sensitivity and convergence study

From the Koopman theory we understand the power of the DMDc due to its ability to potentially capture complex nonlinear dynamics. However, it is limited due to the finite-size of the system matrices compared to the infinitely large linear Koopman operator.

From the Koopman theory we learn that the obtained system matrices A and B have a direct relationship with the Koopman linear operator. However, the matrices A and B are highly dependent on the quality of the data: that is how discretized is the data in space and how many snapshots do we need such that the most important dynamics are captured and which specific underlying dynamics should be present in the data such that we can use the obtained system matrices for generalization purposes. The Koopman linear operator is infinitely large, whereas the system matrices A and B are finitely large and hence the finite information they contain is depend on the data.

In this study, we will discuss the convergence of the system matrices as a function of N_s, which is the number of snapshots in the data. From Eq. 3.32 we can see that the obtained system matrices from the DMDc is not unique, since each different dataset Ω will return a different augmented matrix $G \triangleq [A\, B]$. Hence, the system matrices A and B are not unique.

However, it is still possible to obtain the actual underlying dynamical modes from the data by considering the eigenvalue problem of the reduced order system matrix $\tilde{A}$, expressed in Eq. 3.36, as done in [23]. It is shown that the DMDc is able to reconstruct the underlying dynamic modes, even when the data is polluted with an (open-loop) control signal. Assuming that there is no introduction of energy into our system, except through the control signal which we do not consider for the convergence analysis since we only consider $\tilde{A}$, we know that the relevant modes should stay the same [46]. We study the convergence of the DMDc modes by studying the convergence of the corresponding eigenvalues and amplitudes.

We will also elaborate how we obtained the system matrices from one specific dataset to control floating structures in various sea states, which we discuss in 3.3.4. The choice of the simulated data will be elaborated and the underlying dynamics we inherently hope to include in the data will be searched by using the theoretical wet-modes.

Key takeaway

- DMDc's system matrices A and B approximate the infinite Koopman operator, so capturing essential dynamics depends on data quality.
- High spatial and temporal resolution and sufficient snapshots (N_s) are crucial for ensuring accurate convergence of dynamic modes.
- We analyze eigenvalues and amplitudes of $\tilde{A}$ as N_s increases, aiming to determine the minimum samples needed to capture all relevant dynamics.

3.3.4. DMDc control performance: matching vs. mismatched source functions

One of the challenges of DMDc is its inability to generalize outside its training data if the source function in the PDE describing the system changes. If we consider the dynamics of the system to act according to Eq. 3.42 we know in our case that the function $F_t(\cdot)$ is dependent on the incoming excitation wave. However, using diffraction theory for offshore structures we may be able to control floating structures in various sea states. The reasoning behind this can be found below.

System matrices for regular wave conditions

In the case of regular wave conditions, the source term driving the system remains constant. The source function used for the actual control of the Finite Element Model is identical to the source function used during the data collection phase. As a result, the system matrices A and B

closely approximate the behavior of the Finite Element model that represents the original system dynamics.

Key takeaway

- The system matrices are obtained by collecting time-series data of the FE-model of the floating structure in regular wave condition.
- The DMDc method is then applied to this data, decomposing the dynamic system into modes and estimating the matrices $\tilde{A}$ and $\tilde{B}$, which represent the system's dynamics and control input influence, respectively in the reduced order space.
- We use the obtained system matrices to control a floating structure in the exact same conditions as for the conditions used to obtain the data for the DMDc.

Since the source term remains unchanged, the system approximation retains high accuracy in predicting the dynamics of the floating structure, and the DMDc system matrices are effective at representing the system's behavior under these conditions.

System matrices for sea state conditions

In the case of sea state conditions, the source term is no longer consistent, and the wave characteristics vary over time. This introduces a mismatch between the source function used for collecting data and the source function encountered during control. As a result, the DMDc system matrices A and B face challenges in generalizing to these new conditions.

Despite this mismatch, short-term forecasting using the DMDc model remains accurate, particularly when the control horizon is short. This is because the hydrodynamic forcing on the structure is primarily influenced by the structural properties of the floating object, rather than the specific source term over longer time scales.

Key takeaway

- The diffraction theory suggests that the interaction between the structure and the incoming wave primarily determines the forcing on the structure. As a result, the short-term response can be predicted accurately by DMDc even under mismatched source conditions.
- When the forecast horizon is kept short, the DMDc model continues to track the system's behavior with reasonable accuracy, because the long-term effects of the varying source function only manifest over extended time periods.

Thus, while the system matrices are less accurate at representing the long-term dynamics under varying source conditions, their short-term performance remains reliable, making DMDc a feasible tool for control applications in sea state conditions when the forecast horizon is not too long. It is therefore essential to determine the length of the control horizon by finding this short-term time-frame where the hydrodynamic force is dominating the dynamics of the system rather than the incoming excitation wave which is the mismatched source term.

3.3.5. DMD(c) against POD

DMDc and Proper Orthogonal Decomposition (POD) are two popular data-driven techniques for analyzing system dynamics. While both methods decompose the system into modes, they have key differences that make DMDc more suitable for controlled dynamic systems.

POD is designed to extract the most energetic modes of a system and rank it accordingly from highest energy content to lowest energy content [46], but it does not capture the temporal dynamics of the system. This makes POD useful for dimensionality reduction but less applicable to control applications where time evolution is critical. In contrast, DMDc captures the system's dynamic modes and accounts for control inputs through the system matrices A and B. Moreover, because the DMDc eigenvalues are complex-valued, we find that the temporal dynamics contain information about the oscillation rate and the growth/decay rate.

In floating structures, the wet modes—modes influenced by fluid-structure interactions—are essential for capturing the system's hydrodynamic behavior. In that case, POD can still play a valuable role in ensuring data quality. By applying POD to the dataset, we can verify that the most energetic modes of the floating structure, which should be the wet-modes, are present. This helps us ensure that the DMDc model has access to the relevant dynamics, allowing it to generalize to different sea states.

By combining POD's ability to assess data quality with DMDc's strength in dynamic modeling, we can build robust models that generalize well to varying sea conditions, improving the accuracy of short-term forecasting and control performance.

Key takeaway

- POD captures dominant energy modes but lacks temporal dynamics, limiting its use in control. DMDc, with system matrices A and B, captures both dynamic modes and control inputs, making it better suited for controlled systems.
- In floating structures, wet modes are essential for hydrodynamic behavior. POD helps verify these high-energy modes, ensuring relevant dynamics for DMDc.
- Using POD for data quality with DMDc for dynamic modeling enhances robustness, enabling better forecasting and control under varying sea conditions.

3.4. Design of MPC

In this section we derive the control law and the adaptive observer gain in case we are dealing with an unconstrained LMPC and we briefly introduce a possible implementation of a constrained NMPC. First, we present the MPC in a scheme. From there, we discuss the necessary analysis that should be performed to ensure the ability of control and stability of the MPC to prevent abnormal behaviour due to abnormal control signals.

3.4.1. MPC block

The first component which acts first in the MPC during an operation at a time instant k, is the *Kalman filter*. The Kalman filter is, as explained before, the component that reconstructs the entire (reduced order) state from sparse measurements. The Kalman filter is highly dependent on both process noise and measurement noise, which are modeled as uncertainties in the system. These uncertainties are captured in the process noise covariance matrix Q and the measurement noise covariance matrix R. The Kalman filter updates its estimates based on the noisy measurements and the system dynamics, balancing trust between the model's prediction and the observed measurements. By adjusting Q and R, the filter can be tuned to be more sensitive to either the measurements or the model's predictions, thus handling noise and uncertainties effectively. A more intuitive way to look at the Kalman filter is by considering it as a parallel simulation of the true system which keeps a register of the previous state to reconstruct an prediction of the current state and then refines this prediction with the current measurements to obtain a better estimate of the state of the true system.

This reconstructed (reduced order) state is then passed down to the second component, which is the *optimizer*. The optimizer minimizes an objective function, which is designed to forecast the system's behavior N_c steps ahead based on the current state and the system matrices. The objective function typically combines terms for minimizing control effort and ensuring the system tracks a desired reference trajectory, often expressed as a quadratic cost function involving the state and control variables. This quadratic cost function can be tuned using weights such that we enforce a hard type of control or a more smooth type of control, depending on the stability of the optimizer and the user's preference. Using the system's dynamics and the predicted future states, the optimizer computes an optimized control action sequence. However, only the first control input from this optimized forecast is applied to the true system at each time step. After applying this control input, the system takes new measurements, and the cycle repeats, starting with an updated state estimate from the Kalman filter and proceeding to a new optimization. In the case of an unconstrained NMPC, we use an iterative optimization method which requires a recursive workflow by feeding the control action sequence back to the MPC model until an optimized control action sequence is found.

As one may notice, the MPC contains a number of inter-dependencies which leads to the following points we should take into account when designing the MPC:

MPC as a general block

1. **Observability and stabilizability conditions**: The observability and stabilizability conditions are conditions that ensure the ability to uniquely reconstruct the system's state from the set of sparse measurements and the ability to stabilize the floating structure respectively. By ensuring these two conditions we have certainty that the MPC will be able to successfully reach its objective or at least reach its objective to obtain satisfactory results.

Kalman filter

1. **Influence of the covariance matrices of the noise on the Kalman filter**: one way to investigate this is by examining the poles of the (steady) Kalman filter gain as a function of different Q and R.
2. **Stability of the Kalman filter:** do the picked covariance matrices lead to a stable Kalman filter and hence a stable MPC?

Optimizer

1. **Design of the objective function**: how is the objective function designed and how does each part of this quadratic cost function contribute to the objective function? How are the weights chosen to fine-tune the optimizer?
2. **The length of the control horizon** N_c: the length of the control horizon N_c highly influences the stability of the MPC. Because of the large model mismatch that will be present when controlling the flexible floating structure in sea state, we need to find the N_c where the system matrices A and B mostly describe the region where the hydrodynamic forcing is the most dominant. A big N_c may lead to big model mismatch because of the mismatch between the incoming wave used in the training data and the actual control case with a random sea state.

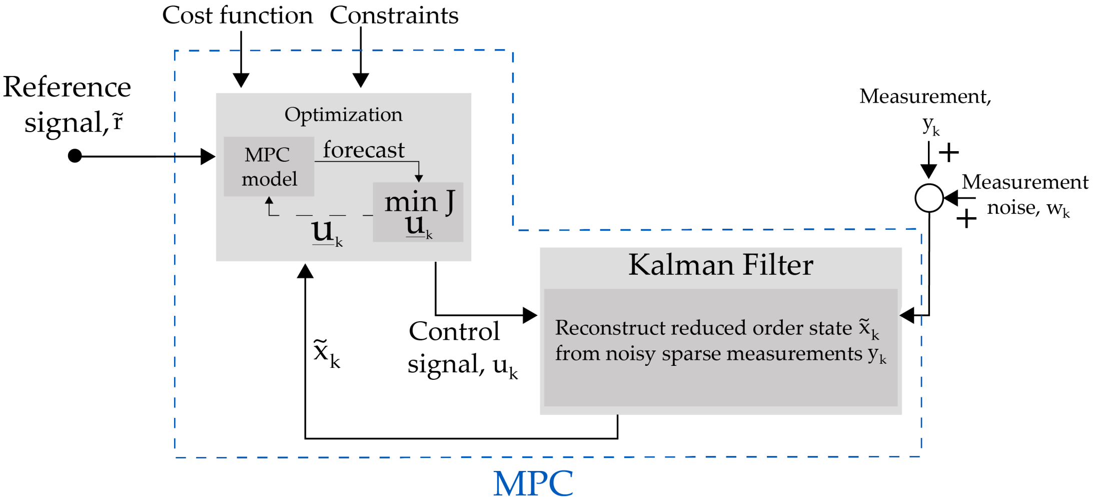


Figure 3.7: The block diagram illustrates the MPC framework. The Kalman filter reconstructs the reduced-order state $\tilde{x}_k$ from noisy, sparse measurements y_k. This estimated state is passed to the optimizer, which minimizes an objective function to compute the optimal control sequence $\underline{\mathbf{u}}_k$ over a prediction horizon N_c. Only the first control input u_k is applied to the true system, depicted in Fig.3.4, and the process repeats with updated measurements. The optimizer considers both the cost function and constraints during each cycle to ensure stability and desired system performance.

3.4.2. Unconstrained Linear MPC

In this subsection we will derive the control law for an unconstrained LMPC. As described in 2.6 this is applicable when dealing with Linear Model Predictive Control (LMPC). In this study we will prescribe the LMPC in the reduced space to achieve high performances for on-line control.

Let us consider the Finite Element model to be approximated by a full order representation as described in Eq. 3.30. We assume that at time instant k we can measure a part of the full state of the system such that $\mathbf{y}_k = D\mathbf{x}_k + \mathbf{w}_k$. In our case, this means we only have sparse measurements of the deformation of the floating structure and these measurements may contain the noise $\mathbf{w}_k$ due to imperfect sensors, which is common in real-life practical applications.

In the context of control we expand this by adding a signal $\hat{\mathbf{p}}_k \in (\mathbb{R}^w \subset \mathbb{R}^n)$, that is $\hat{\mathbf{p}}$ is a subset of the full order estimated state variable $\hat{\mathbf{x}}$, which we opt to control. An example would be the control of the some part of the floating structure, in that case the signal $\hat{\mathbf{p}}$ describes the deformation of that specific part we want to control or stabilize, such as the rear part located after a joint.

We can now write our state-space system **according** to the controller as

$$\hat{\mathbf{x}}_{k+1} = A\hat{\mathbf{x}}_k + B\mathbf{u}_k + K_k(\mathbf{y}_k - \hat{\mathbf{y}}_k) \tag{3.53}$$
$$\hat{\mathbf{y}}_k = D\hat{\mathbf{x}}_k \tag{3.54}$$
$$\hat{\mathbf{p}}_k = C\hat{\mathbf{x}}_k. \tag{3.55}$$

It is very important to realize that the state-space system described above is the description of the system in the MPC. We denote this fact using the $\hat{}$ as superscript. This means that within the MPC we keep a register of the **estimated** state of the system since the control does not have the ability to know the **true** state of the system. The Kalman filter corrects this error by minimizing the difference between the true measurements $\mathbf{y}_k$ and the reconstructed "measurements" $\hat{\mathbf{y}}_k$. The idea is that if the actual measurements lie close to the reconstructed "measurements" then the state of the system according to the controller will lie close to the actual state of the true system. A stable and correct design of the Kalman filter is therefore very important to ensure that the optimized control signal suits the true state of the system and affects it accordingly.

However, one challenge we still deal with is the high-dimensionality of the state due to the Degrees of Freedom that describe the deformation of the floating structure. Therefore, we perform a state-transformation to map our problem onto a reduced-space which allow on-line optimization for high-frequency control without loss of performance.

Let us now transform our state-space representation to the estimated reduced space using our transformation matrix $\hat{U}$ such that we can rewrite 3.53, 3.54 and 3.55 into

$$\tilde{\mathbf{x}}_{k+1} = \tilde{A}\tilde{\mathbf{x}}_k + \tilde{B}\mathbf{u}_k + \tilde{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k) \tag{3.56}$$

$$\hat{\mathbf{y}}_k = D\hat{U}\tilde{\mathbf{x}}_k = \tilde{D}\tilde{\mathbf{x}}_k \tag{3.57}$$

$$\tilde{\mathbf{p}}_k = \tilde{C}\tilde{\mathbf{x}}_k \tag{3.58}$$

Hence, we now denote our **reduced and estimated** signals described in Eqs.3.53, 3.54 and 3.55 using the $\tilde{}$ as superscript.

Moreover, the matrices $\tilde{A}$ and $\tilde{B}$ are found by the use of the DMDc from the data obtained from the FEM. As for the matrix $\tilde{C}$ we first need to determine what subset in the state vector $\mathbf{x}$ we want to control. By first constructing the selection matrix C we can then find our matrix $\tilde{C}$ from deriving that

$$\tilde{\mathbf{x}} = \hat{U}^{-1}\hat{\mathbf{x}} \tag{3.59}$$

$$\tilde{\mathbf{p}} = \hat{U}_{\text{sub}}^{-1}\hat{\mathbf{p}} \tag{3.60}$$

$$\tilde{\mathbf{p}} = \hat{U}_{\text{sub}}^{-1}C\hat{\mathbf{x}} = \underbrace{\hat{U}_{\text{sub}}^{-1}C\hat{U}}_{\tilde{C}}\,\tilde{\mathbf{x}} \tag{3.61}$$

where we can interpret the matrix $\hat{U}_{\text{sub}}^{-1}$ as the transformation matrix to obtain the reduced order signal we want to control, since the signal $\hat{\mathbf{p}}$ is a set of the state variables. This matrix can be computed using the DMDc by simply considering only the set of state variables we want to control when feeding the data to the DMDc.

Finally, it is noteworthy to emphasize that the Eqs.3.56 and 3.57 are primarily important for the design of the *Kalman filter*, whereas the signal we opt to control, $\tilde{\mathbf{p}}_k$, is primarily important for the design of the *optimizer*.

3.4.3. Hautus Lemma's

Before designing the actual control mechanism we need to ensure that the system is observable and stabilizable. These are inherent properties of the dynamic system which will prove their importance as discussed below. Observability and stabilizability are both dependent on the used system matrices of the model found in the MPC. Therefore, after obtaining the system matrices from the DMDc we always check for these conditions first.

Observability

A system is said to be observable if given a finite number of N measurements $(\mathbf{y}_0, ..., \mathbf{y}_{N-1})$ we can uniquely distinguish the initial state $\mathbf{x}_0$ [36]. If this condition is satisfied we are able to accurately reconstruct the system's state from the sparse measurements given that the designed observer is stable. To check if the system is observable we make use of the following Lemma **Lemma 1** (*Hautus Lemma for observability*). A system is said to be observable if and only if

$$\text{rank}\begin{bmatrix}\lambda I - \tilde{A} \\ \tilde{D}\end{bmatrix} = n \quad \text{for all } \lambda \in \text{eig}(A)$$

Stabilizability

Stabilizability is the condition that, if met, allows us to drive the state of a dynamical system within a finite amount of time to *approach* any stable state. It is important to understand that we do not check for controllability, which is a stronger condition than stabilizability. That is because we are dealing with incoming waves which are practically impossible to dampen out completely. Controllability means that we can drive the state of the dynamical system exactly to the reference trajectory. Therefore, in stabilizability, we only require the state of the system to *approach* any stable state. It is true that if a system is controllable, then it is also stabilizable, but the reverse is not necessarily true. This means that if some modes are not controllable, they can at least be stabilized to a steady-state [36].

Lemma 2 (*Hautus lemma for stabilizability*). A system is said to be stabilizable if and only if

$$\text{rank}\begin{bmatrix}\lambda I - \tilde{A} & \tilde{B}\end{bmatrix} = n \quad \text{for all } |\lambda| \geq 1 \text{ where } \lambda \in \text{eig}(A)$$

Hence, we check that all the uncontrollable modes, where the corresponding absolute value of the eigenvalue is greater than 1, can be stabilized by the introduction of the control matrix B.

Observer gain - the Kalman filter approach

There are various methods to design the observer gain for a system. One common method is to derive a Luenberger observer, which utilizes a constant observer gain determined by solving the discrete Riccati equation, given the covariance matrices Q and R, as discussed in similar works like [10]. The primary advantage of this approach is its simplicity: the constant gain only needs to be computed once by solving the discrete Riccati equation.

However, because in reality our system may have uncertainties arising from model mismatches between the actual system and the assumed model, as mentioned in [42], we prefer an adaptive approach using the Kalman filter. The Kalman filter's strength lies in its adaptive nature, which allows it to continuously update and refine the state estimates in response to new measurements. This adaptive feature is crucial in dealing with unknown model mismatches that we assume are present in our study. The adaptive part is introduced by the use of the so-called error covariance matrix

$$P_k = \mathbb{E}\left[\left((\hat{U}^{-1}\mathbf{x}_k) - \tilde{\mathbf{x}}_k\right)\left((\hat{U}^{-1}\mathbf{x}_k) - \tilde{\mathbf{x}}_k\right)^T\right] \tag{3.62}$$

where we treat the difference between the reduced true state of the system, denoted as $\hat{U}^{-1}\mathbf{x}_k$ and the estimated reduced state of the system, denoted as $\tilde{\mathbf{x}}_k$, as a stochastic variable. This is due to the stochastic nature of the measurements $\mathbf{y}$ from which we reconstruct the estimated state of the system, this will be elaborated more down below.

Therefore, the goal of the Kalman filter is to minimize the variance of this stochastic variable, which is as mentioned earlier the difference between the true state of the system and the estimated state. Eventually when we minimize this variance, we are allowed to assume that the estimated state of the system should be very close to the actual true state of the system.

This covariance matrix comes in two forms, the prior covariance matrix $P_{k|k-1}$ and the posterior covariance matrix $P_{k|k}$. The prior covariance matrix $P_{k|k-1}$ denotes the variance between the actual reduced state $\hat{U}^{-1}\mathbf{x}_k$ and prior estimated reduced state $\tilde{\mathbf{x}}_{k|k-1}$, whereas the posterior covariance matrix $P_{k|k}$ denotes the variance between the actual reduced state $\hat{U}^{-1}\mathbf{x}_k$ and the posterior estimated reduced state $\tilde{\mathbf{x}}_{k|k}$. We find that in the Kalman approach the prior covariance matrix is used to obtain the Kalman gain before the measurement is observed and gets updated to the posterior covariance matrix after the posterior state estimate is computed [24].

The Kalman filter works by generating a posterior prediction of the state $\tilde{\mathbf{x}}_{k|k}$ by correcting a prior estimate $\tilde{\mathbf{x}}_{k|k-1}$ with the current measurement $\mathbf{y}_k$ [42]. The process can be summarized in the following steps:

1. Initialize $k = 0$ and start with an initial guess of the system's state $\tilde{\mathbf{x}}_{0|0}$, the initial control input $\mathbf{u}_{0|0}$, and the initial error covariance matrix $P_{0|0}$.
2. Predict the system's state using

$$\tilde{\mathbf{x}}_{k+1|k} = \tilde{A}\tilde{\mathbf{x}}_{k|k} + \tilde{B}\mathbf{u}_{k|k}$$

where $\tilde{\mathbf{x}}_{k+1|k}$ is the current prior estimate of the subsequent state.

3. Compute the predicted error covariance matrix

$$P_{k+1|k} = \tilde{A}P_{k|k}\tilde{A}^T + Q$$

4. Calculate the Kalman gain

$$\tilde{K}_{k+1} = P_{k+1|k}\tilde{D}^T(\tilde{D}P_{k+1|k}\tilde{D}^T + R)^{-1}$$

5. Wait for the true system to response and update the time-instant such that $k + 1 \to k$. Hence, all the "subsequent" quantities are now the current quantities.
6. Obtain the system response and measure $\mathbf{y}_k$.
7. Update the prior current state estimate to the posterior current state estimate by incorporating the measurement

$$\tilde{\mathbf{x}}_{k|k} = \tilde{\mathbf{x}}_{k|k-1} + \tilde{K}_k(\mathbf{y}_k - \tilde{D}\hat{U}\tilde{\mathbf{x}}_{k|k-1})$$

8. Update the error covariance matrix to obtain the posterior covariance matrix

$$P_{k|k} = (I - \tilde{K}_k\tilde{D}\hat{U})P_{k|k-1}$$

9. Feed $\tilde{\mathbf{x}}_{k|k}$ to the optimizer to determine the optimal control input $\mathbf{u}_{k|k}$.
10. Repeat from step 2.

Optimizer

Before diving into the objective function let us recall that *in MPC we solve an optimization problem at each time-instant k* by forecasting the evolution of the system dynamics N_c time-instant ahead and compute an **optimized control signal sequence $\underline{\mathbf{u}}_k$** at time-instant k where eventually we only implement the very first element $\mathbf{u}_{k|k}$ as control input on the true system.

The objective function we will derive is a quadratic cost function. In this quadratic cost function we want to ensure that we include the following:

- the forecasted reduced order signal $\tilde{\mathbf{p}}$ should follow a reduced order desired reference trajectory $\tilde{\mathbf{r}}$,
- the user should be able to fine-tune the smoothness of the control signal as it advances in time by penalizing the forecasted control increments $\Delta\mathbf{u}$
- and lastly the user should be able to penalize the magnitude of the control signal that will act on the true system by penalizing the very first element $\mathbf{u}_{k|k}$of the optimized, forecasted control signal.

$$J(\underline{\tilde{\mathbf{z}}}_k, \underline{\mathbf{u}}_k) = \sum_{i=1}^{N_c}\left(\tilde{\mathbf{z}}_{k+i|k}^\top \mathcal{Q}\tilde{\mathbf{z}}_{k+i|k}\right) + \sum_{i=1}^{N_c-1}\left(\Delta\mathbf{u}_{k+i|k}^\top \mathcal{R}_1\Delta\mathbf{u}_{k+i|k}\right) + \mathbf{u}_{k|k}^\top\mathcal{R}_2\mathbf{u}_{k|k} \qquad (3.63)$$

In the literature we often find this type of objective function, formulated in Eq. 3.63, which is known as the Generalized Predictive Control (GPC) performance index [28, 6]. We penalize each term by using $\mathcal{Q}$, $\mathcal{R}_1$ and $\mathcal{R}_2$ as penalty weights respectively. A concise way to write this objective function is as follows:

$$J(\underline{\tilde{\mathbf{z}}}_k, \underline{\mathbf{u}}_k) = \sum_{i=1}^{N_c} \left(\|\tilde{\mathbf{z}}_{k+i|k}\|^2_{\mathcal{Q}}\right) + \sum_{i=1}^{N_c-1} \left(\|\Delta \mathbf{u}_{k+i|k}\|^2_{\mathcal{R}_1}\right) + \|\mathbf{u}_{k|k}\|^2_{\mathcal{R}_2}, \tag{3.64}$$

where $\tilde{\mathbf{z}} = \tilde{\mathbf{p}} - \tilde{\mathbf{r}}$, and $\tilde{\mathbf{r}}$ is the reference we want our signal to follow, such that it goes

$$\tilde{\mathbf{z}}_{k+1|k} = \tilde{C}\tilde{A}\tilde{\mathbf{x}}_{k|k} + \tilde{C}\tilde{B}\mathbf{u}_{k|k} - \tilde{\mathbf{r}}_{k+1|k} \tag{3.65}$$

$$\tilde{\mathbf{z}}_{k+2|k} = \tilde{C}\tilde{A}^2\tilde{\mathbf{x}}_{k|k} + \tilde{C}\tilde{A}\tilde{B}\mathbf{u}_{k|k} + \tilde{C}\tilde{B}\mathbf{u}_{k+1|k} - \tilde{\mathbf{r}}_{k+2|k} \tag{3.66}$$

$$\vdots$$

$$\tilde{\mathbf{z}}_{k+N_c|k} = \tilde{C}\tilde{A}^{N_c}\tilde{\mathbf{x}}_{k|k} + \tilde{C}\tilde{A}^{N_c}\tilde{B}\mathbf{u}_{k|k} + ... + \tilde{C}\tilde{B}\mathbf{u}_{k+N_c-1|k} - \tilde{\mathbf{r}}_{k+N_c|k} \tag{3.67}$$

which we can rewrite in a more compact form such that

$$\underline{\tilde{\mathbf{z}}}_k = \mathcal{M}\tilde{\mathbf{x}}_k + \mathcal{C}\underline{\mathbf{u}}_k - \underline{\tilde{\mathbf{r}}}_k \tag{3.68}$$

$$\underline{\mathbf{u}}_k = \begin{bmatrix} \mathbf{u}_{k|k} \\ \vdots \\ \mathbf{u}_{k+N_c-1|k} \end{bmatrix}, \quad \underline{\tilde{\mathbf{r}}}_k = \begin{bmatrix} \tilde{\mathbf{r}}_{k+1|k} \\ \vdots \\ \tilde{\mathbf{r}}_{k+N_c|k} \end{bmatrix} \tag{3.69}$$

$$\mathcal{M} = \begin{bmatrix} \tilde{C}\tilde{A} \\ \tilde{C}\tilde{A}^2 \\ \vdots \\ \tilde{C}\tilde{A}^{N_c} \end{bmatrix}, \quad \mathcal{C} = \begin{bmatrix} \tilde{C}\tilde{B} & 0 & \cdots & 0 \\ \tilde{C}\tilde{A}\tilde{B} & \tilde{C}\tilde{B} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{C}\tilde{A}^{N_c}\tilde{B} & \tilde{C}\tilde{A}^{N_c-1}\tilde{B} & \cdots & \tilde{C}\tilde{B} \end{bmatrix} \tag{3.70}$$

which allow us to write our objective function in matrix-form notation

$$J = \underline{\tilde{\mathbf{z}}}_k^\top \mathcal{Q}\underline{\tilde{\mathbf{z}}}_k + \underline{\mathbf{u}}_k^\top \mathcal{R}_1\underline{\mathbf{u}}_k + \underline{\mathbf{u}}_k^\top \mathcal{R}_2\underline{\mathbf{u}}_k. \tag{3.71}$$

Eventually, having defined the objective function, our goal is to find the **optimized control signal sequence $\underline{\mathbf{u}}_k^\star$**:

$$\underline{\mathbf{u}}_k^\star = \arg\min_{\underline{\mathbf{u}}_k} \left(J = \underline{\tilde{\mathbf{z}}}_k^\top \mathcal{Q}\underline{\tilde{\mathbf{z}}}_k + \underline{\mathbf{u}}_k^\top \mathcal{R}_1\underline{\mathbf{u}}_k + \underline{\mathbf{u}}_k^\top \mathcal{R}_2\underline{\mathbf{u}}_k\right). \tag{3.72}$$

In the case our quadratic cost function is unconstrained, which is the case we consider, we are able to find a closed-form solution.

For the unconstrained case, we want to find the closed form solution by setting

$$\frac{\partial J}{\partial \underline{\mathbf{u}}_k} = 0. \tag{3.73}$$

The weight matrices $\mathcal{Q}$, $\mathcal{R}_1$, and $\mathcal{R}_2$ in the objective function enable fine-tuning of the controller by adjusting the impact of different terms in the cost function:

- **Tracking Performance**: $\mathcal{Q}$ controls the weight on $\tilde{\mathbf{z}}_{k+i|k}$, which measures the deviation from the reference trajectory $\tilde{\mathbf{r}}$. Increasing $\mathcal{Q}$ emphasizes accurate tracking, prioritizing alignment with the reference path over control effort.
- **Smoothness of Control**: $\mathcal{R}_1$ penalizes changes in the control signal $\Delta\mathbf{u}$, influencing the smoothness of control actions over time. A higher $\mathcal{R}_1$ discourages abrupt control changes, resulting in smoother adjustments.
- **Control Effort**: $\mathcal{R}_2$ penalizes the magnitude of the applied control input $\mathbf{u}_{k|k}$, balancing aggressive control actions with energy efficiency. Larger values of $\mathcal{R}_2$ help reduce excessive control input, which is useful in systems where energy efficiency or actuator limitations are important.

By adjusting these weights, the controller can be tailored for various objectives, achieving a balance between tracking precision, control smoothness, and energy efficiency according to specific requirements.

Solving Eq. 3.73 leads to the optimized control signal sequence over the control horizon

$$\underline{\mathbf{u}}_k^\star = \mathcal{H}^{-1}\mathcal{S}^\top \underline{\tilde{\mathbf{r}}} - \mathcal{H}^{-1}\mathcal{F}\tilde{\mathbf{x}}_k \tag{3.74}$$

Definitions

$$\mathcal{G} = \mathcal{M}^\top \mathcal{Q} M \tag{3.75}$$

$$\mathcal{F} = \mathcal{C}^\top \mathcal{Q}\mathcal{M} \tag{3.76}$$

$$\mathcal{W} = \mathcal{Q}\mathcal{M} \tag{3.77}$$

$$\mathcal{H} = \mathcal{C}^\top \mathcal{Q}\mathcal{C} + \mathcal{R}_1 + \mathcal{R}_2 \tag{3.78}$$

$$\mathcal{S} = \mathcal{Q}\mathcal{C} \tag{3.79}$$

For the full derivation of this result we refer to App. B.

Note how this optimized control signal sequence over the control horizon is dependent on the forecasting matrices $\mathcal{M}$ and $\mathcal{C}$ and therefore makes the control signal sequence dependent on the length of the control horizon. Moreover, note how this optimized control signal sequence is dependent on the weight matrices $\mathcal{Q}$, $\mathcal{R}_1$ and $\mathcal{R}_2$, which directly means that the choice of these weights will affect the stability of the controller. This will be discussed later on in this chapter.

3.4.4. Constrained MPC

In practice we often deal with control actuators which are limited to a certain extent. For instance, in this study we control the floating structure by applying an external moment acting on the joint coupling two sub-components of the floating structure. This actuator can in general apply an external moment up to a certain upper and lower bound, where the upper bound represents the maximum absolute moment possible in the positive direction and the lower bound represents the maximum absolute possible moment possible in the negative direction.

For objective functions where we have to deal with these kind of constraints we are not able to find a closed-form solution as described in the previous section. 3rd party packages like *JuMP* [25] are used for objective functions expressed in any form similar to Eq.2.16. In this study we consider an objective function of the same form as expressed in Eq.3.71 expanded by constraints putting upper and lower bounds for the control input. Hence, we have

$$\min_{\underline{\mathbf{u}}} J \quad \text{s.t.} \quad \underline{\mathbf{u}}_{\min} \leq \underline{\mathbf{u}} \leq \underline{\mathbf{u}}_{\max} \tag{3.80}$$

Note that in this thesis, we will not implement the constrained MPC. However, it is important to note that adding the constrained MPC, as described above, requires only a few lines of code and is relatively straightforward. One of the big advantages of MPC is that we control our floating structure by solving an optimization problem and hence allowing for constraints, this is often not possible in classic control, such as a LQR controllers or PID controllers. This framework allows MPC to handle complex systems, such as flexible floating structures, with high adaptability and efficiency, providing robust control and stability in dynamic environments.

3.4.5. Closed-loop stability

The introduction of the Kalman filter in combination with the optimization algorithm is what combined gives us the MPC. In the world of control we often need to prove stability of the controller. There are two types of stability we need to ensure: stability of the observer gain and the stability of the controller gain.

The stability we consider is only dependent on the true state of the actual system we control and the estimated state of the system according to our controller. From Eq. 3.74 we find that the control input can be found by *considering only the first control input along the control horizon* from the optimization such that if we consider a zero reference-trajectory we get

$$\mathbf{u}_{k|k} = -F\tilde{\mathbf{x}}_k \tag{3.81}$$

where the *controller gain* is defined as

$$F = \begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix} \mathcal{H}^{-1}\mathcal{F}\tilde{\mathbf{x}}_k \tag{3.82}$$

By augmenting the Eqs. 2.9, 2.10, 2.11 and 2.12 by the use of substitution and neglecting the process and measurement noise we find our closed-loop system in the reduced order space to be defined as

$$\begin{bmatrix} \tilde{\mathbf{x}}^{\text{true}}_{k+1} \\ \tilde{\mathbf{x}}_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} \tilde{A} & -\tilde{B}F \\ \tilde{K}_k\tilde{D} & \tilde{A} - \tilde{B}F - \tilde{K}_k\tilde{D} \end{bmatrix}}_{A_{cl}} \begin{bmatrix} \tilde{\mathbf{x}}^{\text{true}}_{k} \\ \tilde{\mathbf{x}}_{k} \end{bmatrix} \tag{3.83}$$

For stability, we want to ensure that the true reduced state of the system $\tilde{\mathbf{x}}^{\text{true}}_k$ is bounded as we progress through time. We also want to make sure that the error of the estimated reduced state and true reduced state stays bounded. That is $\tilde{\mathbf{e}}_k = \tilde{\mathbf{x}}_k - \tilde{\mathbf{x}}^{\text{true}}_k$. Let us introduce these two new states by the use of the transformation matrix [6]

$$\begin{bmatrix} \tilde{\mathbf{x}}^{\text{true}}_{k} \\ \tilde{\mathbf{x}}_k - \tilde{\mathbf{x}}^{\text{true}}_{k} \end{bmatrix} = \begin{bmatrix} I & 0 \\ -I & I \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}^{\text{true}}_{k} \\ \tilde{\mathbf{x}}_{k} \end{bmatrix}, \quad T^{-1} = \begin{bmatrix} I & 0 \\ I & I \end{bmatrix} \tag{3.84}$$

Our new system matrix $\tilde{A}_{cl}$ is now

$$\tilde{A}_{cl} = TA_{cl}T^{-1} \tag{3.85}$$

such that

$$\begin{bmatrix} \tilde{\mathbf{x}}^{\text{true}}_{k+1} \\ \tilde{\mathbf{e}}_{k+1} \end{bmatrix} = \begin{bmatrix} \tilde{A} - \tilde{B}F & -\tilde{B}F \\ 0 & \tilde{A} - \tilde{K}_k\tilde{D} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}^{\text{true}}_{k} \\ \tilde{\mathbf{e}}_{k} \end{bmatrix} \tag{3.86}$$

where the closed-loop stability is guaranteed when the eigenvalues of the matrices $\tilde{A} - \tilde{B}F$ and $\tilde{A} - \tilde{K}_k\tilde{D}$ lie within the unit circle. The Kalman filter gain $\tilde{K}_k$ is an adaptive observer gain. In case these eigenvalues would lie outside the unit circle, we would observe an unbounded grow of the prescribed quantities.

Hence, to ensure the observer gain stability we make sure that the poles of the matrix $\tilde{A} - \tilde{K}_{ss}\tilde{D}$ lie within the unit circle where the steady-state Kalman filter gain is denoted by $\tilde{K}_{ss}$. The steady-state Kalman filter gain is reached when the error covariance matrix reaches its steady-state such that

$$P_{k|k} = P_{k|k-1} = P_{ss} \tag{3.87}$$

where it has been shown that the steady-state error covariance matrix P_{ss} satisfies the steady-state discrete Riccati equation [4]

$$P_{ss} = Q + \tilde{A}\left(P_{ss}^{-1} + \tilde{D}^T R^{-1}\tilde{D}\right)^{-1} + \tilde{A}^T \tag{3.88}$$

where by solving the discrete Riccati equation in eq.3.88 we can subsequently find $\tilde{K}_{ss}$

$$\tilde{K}_{ss} = P_{ss}\tilde{D}^T\left(\tilde{D}P_{ss}\tilde{D}^T + R\right)^{-1} \tag{3.89}$$

It is important to realize that the weight matrices, i.e. $\mathcal{Q}$, $\mathcal{R}_1$ and $\mathcal{R}_2$, directly influence the stability of the controller. Since the controller gain in Eq. 3.82 is directly related to $\mathcal{H}$ in Eq. 3.78. Moreover, the stability of the state estimator, i.e. the Kalman filter, is directly influenced by the chosen covariance matrices of the process and measurement noise as can be read from Eq. 3.89. How we choose the weight matrices and covariance matrices will be explained later on in the report when discussing the actual results of the control.

Model mismatch

Since in this study we opt for a simple, linear representation of the true system using data-driven DMDc, a challenge arises due to model mismatch. Model mismatch occurs when the actual dynamics of the true system do not match the dynamics of the data-driven model used in the MPC. It is important to note that model mismatch is distinct from process noise; model mismatch can often be more critical and misleading when assessing the stability of the controller. We define model mismatch as the difference between the data-driven model and the true system, as expressed by the following equation:

$$\text{model mismatch} = |\underbrace{A\mathbf{x}_k + B\mathbf{u}_k}_{\text{DMDc}} - \underbrace{\mathbf{F}_t(\mathbf{\Theta}_k, \mathbf{u}_k)}_{\text{FEM}}|. \tag{3.90}$$

where we recall that $\mathbf{x} = \boldsymbol{\eta}$ and $\mathbf{\Theta} = [\boldsymbol{\eta}, \boldsymbol{\phi}, \boldsymbol{\kappa}]$. Due to this model mismatch, the closed-loop stability of the MPC can only serve as an indicator of the overall stability of the controller and should not be taken as a definitive measure.

Key takeaway

- Model Predictive Control (MPC) uses a prediction horizon, N_c, to forecast system behavior and optimize control inputs based on state estimates. At each time step, only the first control input from the optimized sequence is applied to the system. This rolling approach enables adaptive control by recalculating future actions at each time instant based on the updated state.
- The Kalman filter reconstructs the full state of the system from sparse measurements, handling measurement noise and process noise through covariance matrices Q and R. These matrices determine the filter's sensitivity, adjusting the balance between trusting model predictions and noisy measurements. Thus, the filter corrects estimated states by refining them with real-time data, providing an accurate input for the control optimization.
- The optimizer minimizes a quadratic objective function, which includes terms for tracking the desired reference trajectory, smoothing control actions, and regulating control input magnitudes. Weight matrices $\mathcal{Q}$, $\mathcal{R}_1$, and $\mathcal{R}_2$ enable fine-tuning:
  - $\mathcal{Q}$: Enhances tracking performance by penalizing deviation from the reference trajectory.
  - $\mathcal{R}_1$: Controls the smoothness of control inputs, discouraging abrupt changes.
  - $\mathcal{R}_2$: Limits the magnitude of the applied control action, balancing aggressive corrections with energy efficiency.
- For MPC to function effectively, it must satisfy observability and stabilizability. These ensure the system's state can be reconstructed accurately and that the MPC can stabilize the floating structure against external disturbances. Stability is achieved by verifying the eigenvalues of the closed-loop system matrices, which need to fall within the unit circle.
- To handle high-dimensional states efficiently, the model reduces them to core dynamic modes via a transformation matrix, preserving key dynamics in a lower-dimensional space. This approach enables fast computation, critical for real-time applications, without significant loss of performance.

- The adaptive observer gain in the Kalman filter allows the system to adjust to uncertainties due to noise. The stability of this filter is monitored through the steady-state gain, ensuring that the closed-loop control remains stable across varying conditions.

4 Methodology

4.1. DMDc

In this section, we will analyze the behavior of our floating structure in two distinct wave environment. The first scenario involves a single regular incoming wave acting on the structure. We generate various datasets using different open-loop control signals to construct our ROM using the DMDc as previously described. The ROM obtained is tailored to control a FE plant operating in the same regular wave environment. We will also investigate the control of a floating structure in sea state conditions. This will be done by creating a new DMDc model using an irregular wave profile. This will be justified later in the report.

In both cases we first represent the conditions used to obtain the training data. Along with that we explain the choice for the design of the open-loop control signal to capture the distinct effect of the control input on the states of the system. We will then discuss the convergence of the DMDc reduced model as a function of the length of the data. Let us recall the convergence study that we shall carry out:

- given one training data along with its control signal, how do the eigenvalues and amplitudes of the dynamic modes converge as a function of the length of the data?

Lastly, we compute an error metric that determines which obtained reduced model is able to generalize the best by validating it against the other data.

4.1.1. Regular wave profile

We start by simulating data from the Finite Element Model using four different control signals that do not have any physical meaning. The goal is to find the relation between the control input and the states of the system by using this simulated data. The used parameters for the data can be found in table 4.1. In Figure 4.1, we present the four control signals under consideration. Each control signal incorporates the transient aspect of the FEM model by starting with a zero value to avoid any instabilities during the transient phase. The initiation of the actual control action occurs in the steady-state portion of the FEM, under the assumption that control operations will be executed in stationary conditions in real scenarios. Below we discuss the distinct features of the open-loop control signals and the motivation behind the different patterns.

Control signal: 1 & 2

Since the plant we opt to control in regular wave conditions is excited by a single incoming wave, it is justifiable to expect that the applied control signal during the actual control to be sinuoisdal. Because of this we designed the first two control signals to be sinusoidals. These first two control signals are similar, differing only in their amplitudes. Each of these signals includes

Parameter	Symbol	Value	Unit
Total length of the structure	L_{str}	12.5	m
Parameter relative locations of the joints	β_j	0.2	$-$
mass structure per water density	$\frac{\rho_b h_b}{\rho_w}$	8.36	m^3
Rigidity of the structure modules	(EI_1, EI_2)	$(47100, 47100)$	$N\,m$
Water depth	H	1.1	m
Gravitational constant	g	9.81	$m\,s^{-2}$
Water density	ρ_w	1025	$kg\,m^{-3}$
Joint stiffness per unit length	k_φ	$0.0 \cdot \frac{EI}{L_{str}}$	$kg\,N\,m^{-3}$
Incoming wave length	λ_{wave}	$0.249 \cdot L_{str}$	m
Wave amplitude	κ_0	0.1	m
Wave frequency	ω	4.40	$rad\,s^{-1}$

Table 4.1: Parameters used to obtain the data for the regular wave case.

four oscillatory components that correspond to the excitation frequency ω: the first component oscillates at $0.25\,\omega$, the second at $0.50\,\omega$, the third at $0.75\,\omega$, and the fourth at $1.0\,\omega$.

Control signal: 3 & 4

The third and fourth control signals are designed to capture a diverse array of control actions, hence the incorporation of oscillatory segments interspersed with step controls. In the third control signal we again start with a zero value before entering the steady-state portion of the FEM. As soon as we enter this steady-state portion we apply an open-loop control signal which is sinusoidal and decays in time. The frequency of this first decaying sinusoidal signal is $0.25\,\omega$. After some time we start applying different linear functions following each other. We then again apply a decaying sinousidal signal with $0.50\,\omega$ as frequency. Finally, we apply again different linear functions.

As for the fourth control signal we consider three main regions. Again, we start with the zero value signal and then we apply a sinusoidal signal with $0.08\,\omega$ as frequency. Note that we also applied an upper bound for this sinusoidal signal, such that it gets flattened as soon it reaches this upper bound. After a while we apply an increasing step signal that varies over time and finally we apply again a sinusoidal signal with $0.03\,\omega$ as frequency where we have a lower bound that flattens the signal when it tries to dip below this lower bound.

The third and fourth control signals are designed to capture a diverse range of control actions, incorporating oscillatory segments alongside step controls. In the third control signal, we begin with a zero value before transitioning to the steady-state phase of the FEM. Once we enter this steady-state phase, we apply an open-loop control signal that is sinusoidal and decays over time. The frequency of this first decaying sinusoidal signal is $0.25\,\omega$. After a period of time, we introduce a series of linear functions, one after the other. Following this, another decaying sinusoidal signal is applied, this time with a frequency of 0.50ω. Finally, we conclude with another set of linear functions.

In the fourth control signal, we divide the input into three main regions. As with the third signal, we start with a zero value and then apply a sinusoidal signal with a frequency of 0.80ω. An upper bound is applied to this sinusoidal signal, flattening it as soon as it reaches the threshold. After some time, we apply an increasing step signal that evolves over time, and lastly, we introduce a sinusoidal signal with a frequency of $0.03\,\omega$. For this final sinusoidal signal, we impose a lower bound, which flattens the signal when it attempts to drop below this limit.

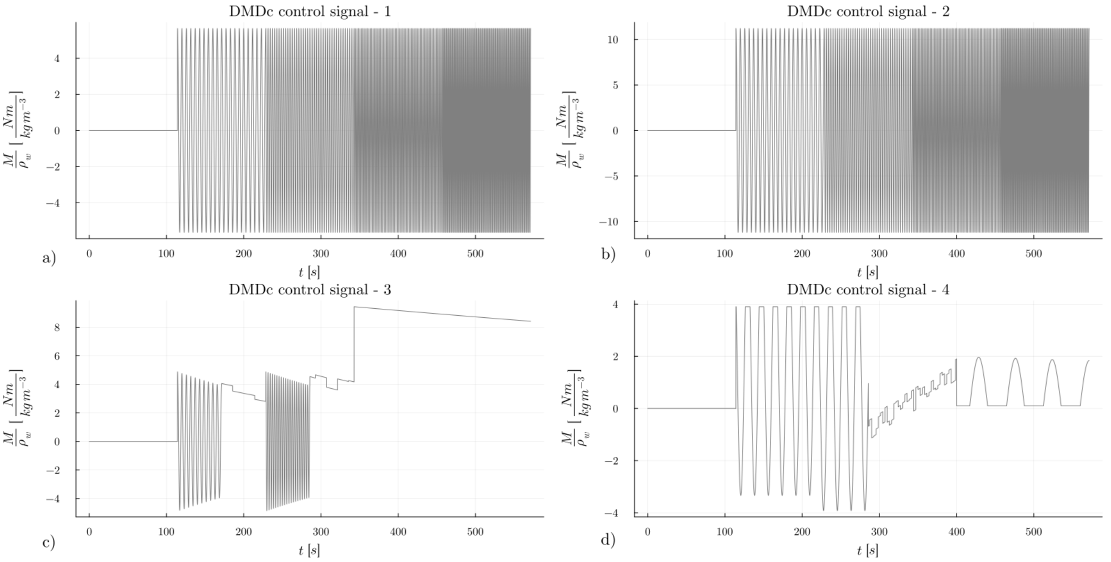


Figure 4.1: The four considered control signals are an applied external moment at the joint of the floating structure.

Justification of open-loop control signals

From the description of the open-loop control signals, it is evident that we incorporated as many patterns as possible. The signal used to obtain our DMDc model does not need to be specifically tailored to the actual control setting. Even if we know nothing about the model, designing a rich open-loop control signal with different patterns should be sufficient to distinguish the A matrix, which represents the system dynamics, and the B matrix, which reflects the effect of the control input on the system. According to [23], a white-noise signal could even be applied to capture the B matrix. Nevertheless, we choose for the above described control signals since they contain sinusoidal patterns with different frequencies, linear piece-wise functions, step function and cut-off signals using lower-bounds and upper-bounds, which may represent the effect of constraints.

Obtaining the DMDc model

Let us again recall the theory discussed in section 3.3 where we discussed how using a dataset, in this case we obtain the dataset from the FEM, we can find a data-driven representation of the dynamics. The evolution of the dynamics in full-order space is then expressed as

$$\mathbf{x}_{k+1} \approx A\mathbf{x}_k + B\mathbf{u}_k \tag{4.1}$$

where the matrices A and B approximate the dynamics of the FEM in full-space. However, for control we need a reduced representation where we utilize the DMDc dynamic modes found in the data such that the evolution of the dynamics in reduced order space can be expressed as

$$\tilde{\mathbf{x}}_{k+1} \approx \tilde{A}\tilde{\mathbf{x}}_k + \tilde{B}\mathbf{u}_k. \tag{4.2}$$

where the reduced order state is now represented by the dynamic modes of obtained from the DMDc. By doing a convergence study we are able to determine how many r dynamic modes we should consider in the reduced model to capture the total dynamics sufficiently. This convergence study will be explained in detail later on in this chapter.

Eventually, we will validate each obtained DMDc model against the other datasets. This approach will tell us which DMDc model for a specific dataset along with its open-loop control signal is able to generalize the best. This will also prove the strength of the DMDc to generalize for different patterns of control signals, even if the training dataset didn't capture these patterns.

Convergence of the DMDc model

For this study we take r to be equal to $\tilde{r}$, that is we we take the same number of components that explain the largest variance in the input data as the number of components that explain the variance in the output data. Using control signal 1 we perform the DMDc as explained in 3.3 to compute the matrix $\tilde{\Sigma}$ which contains the singular values along its diagonal. Each entry in this matrix resembles the portion variance the corresponding DMDc mode explains from the total considered DMDc components r, as can be seen from Fig.4.2. If we consider the first 100 DMDc modes of the data we find that the first 40 modes capture more than 99.9% of the variance of the data. Therefore, we consider only the first 40 modes for our ROM.

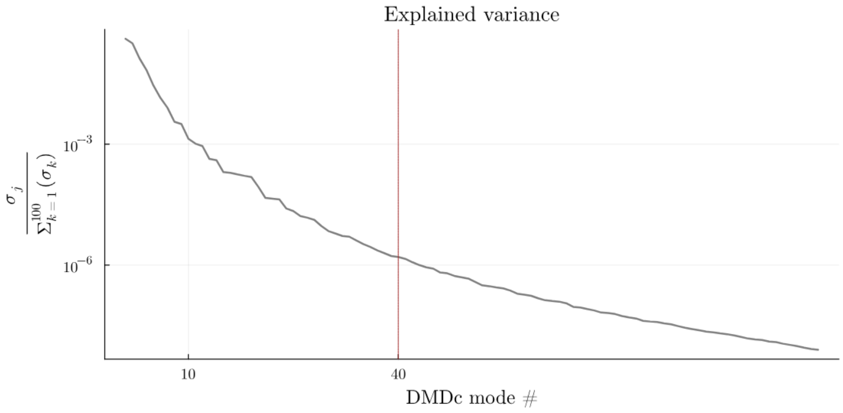


Figure 4.2: The explained variance is a metric that explains the captured variance in the data for the considered dynamical modes of the DMDc.

The used training data has a sampling length N_s which is equal to 12500. To ensure that the ob-

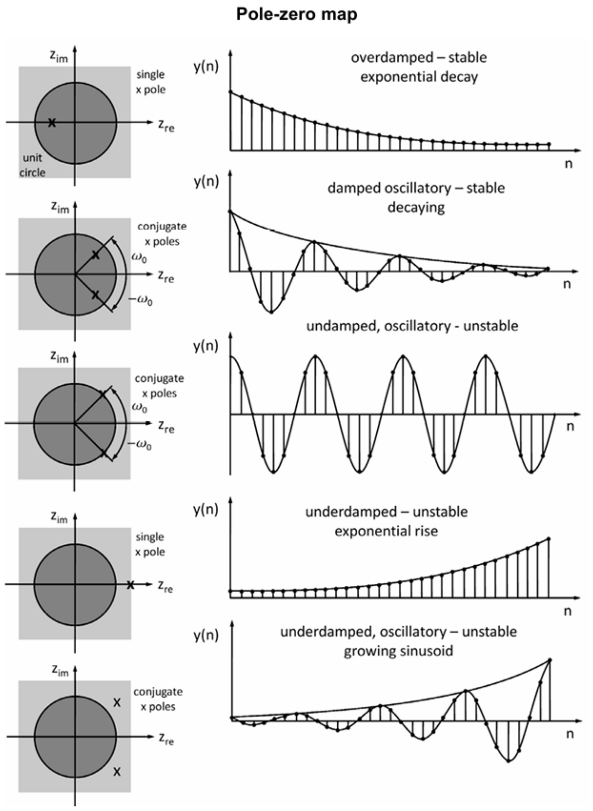


Figure 4.3: The figure illustrates the pole-zero map and its corresponding time-domain responses for different types of system dynamics. The left column shows pole locations, i.e. eigenvalues, in the complex plane, while the right column shows the corresponding output response for each case. [37]

tained DMDc model is converged using this length we compute the eigenvalues and amplitudes of the DMDc modes. Important to realize that the eigenvalues of the DMDc modes carry information about the oscillatory behaviour of its corresponding DMDc mode. Since the eigenvalues, often called *poles*, are complex, it is important to study their placement in the z-plane. Therefore, let us first study the z-plane and how the placement of the poles represent the oscillatory behaviour.

From Fig.4.4 we see five main locations of the poles on the z-plane and how each location represents a different oscillatory signal. Poles that are outside the unit circle are considered unstable and depicts unstable dynamics. In our case, we may assume that the dominant dynamics which are present in the steady-state are (highly) un-damped. The placement of the poles corresponding to such dynamics is near the boundary of the unit-circle as depicted in the third signal in Fig. 4.4.

Therefore, let us zoom in on the poles of the four considered training data near this ”un-damped” region of the z-plane. For each dataset we consider different sampling length where we simply expand the length of the data we feed to the DMDc. Finally, we consider the length we actually use for the obtained DMDc model, which is 12500. From there we compare how the poles converge as the data increases in size as similarly done by [46]. From Fig. 4.4 we can clearly see how increasing the size of the dataset eventually leads to a converged model in all the considered datasets. Along with the convergence of the poles we also need to make sure that the corresponding amplitudes converge. For this we plot the relative amplitude, as depicted in Fig. 4.5. From Fig. 4.5 we can indeed confirm that the used sampling length of 12500 returns a converged model since the corresponding amplitudes of the poles also converge.

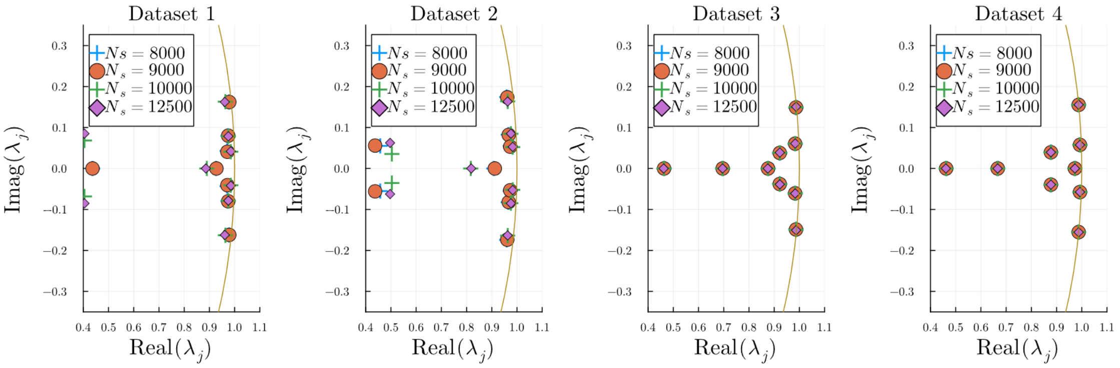


Figure 4.4: Convergence of 15 DMDc poles for different datasets as the number of samples (N_s) increases. For each dataset, four different sample sizes are considered: 8000, 9000, 10000, and 12500. As the sample size increases, the poles converge, particularly for the largest sample size of $N_s = 12500$. This demonstrates the impact of increasing data length on the stability and accuracy of the DMDc model across all datasets.

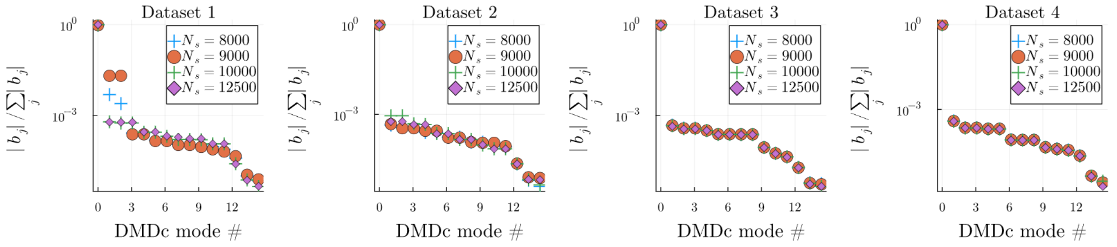


Figure 4.5: Convergence of 15 DMDc mode amplitudes for different datasets as the number of samples (N_s) increases. The relative amplitude $|b_j / \sum b_j|$ for each mode is plotted for four different sample sizes: 8000, 9000, 10000, and 12500. From the figure, it is evident that increasing the sample size leads to convergence of the mode amplitudes, particularly for $N_s = 12500$, which indicates a stable and converged model across all datasets.

Validation of the DMDc model

We can validate the DMDc models of each dataset by cross-validating the obtained DMDc models against the other datasets. Note that the description below of this procedure is something that is largely already done, it simply describes in detail how we obtain the snapshot data en eventually validate the DMDc models against each other.

We validate our DMDc by applying a predefined training open-loop control signal, Υ_{TRN}, to our FEM. This yields the system's response in a matrix, $X_{\text{FEM,TRN}}$, whose columns correspond to spatial degrees of freedom and whose rows represent discrete moments in time. Utilizing this data, we perform DMDc to derive the matrices A and B as presented earlier in Eq.A.8. It is important to note that these matrices are not analogous to those of the ROM; rather, they approximate the full-order dynamics of the FEM as specified earlier in Eq. A.1 and repeated again in Eq. 4.1. In a similar manner, we introduce the predefined validation control signal Υ_{VAL} to the FEM to capture the system's response in $X_{\text{FEM,VAL}}$, paralleling the procedure used for the training phase. We then use this validation dataset to create approximated validation data by implementing Eq.A.1, starting with the first column of $X_{\text{FEM,VAL}}$ as the initial state, together with the validation control signal. This process results in the approximated matrix X_{DMDc}. To quantify the accuracy, we calculate the element-wise absolute differences between the actual and approximated matrices, resulting in matrix E, which contains the absolute errors for each degree of freedom over time. This concept is depicted in Fig.4.6.

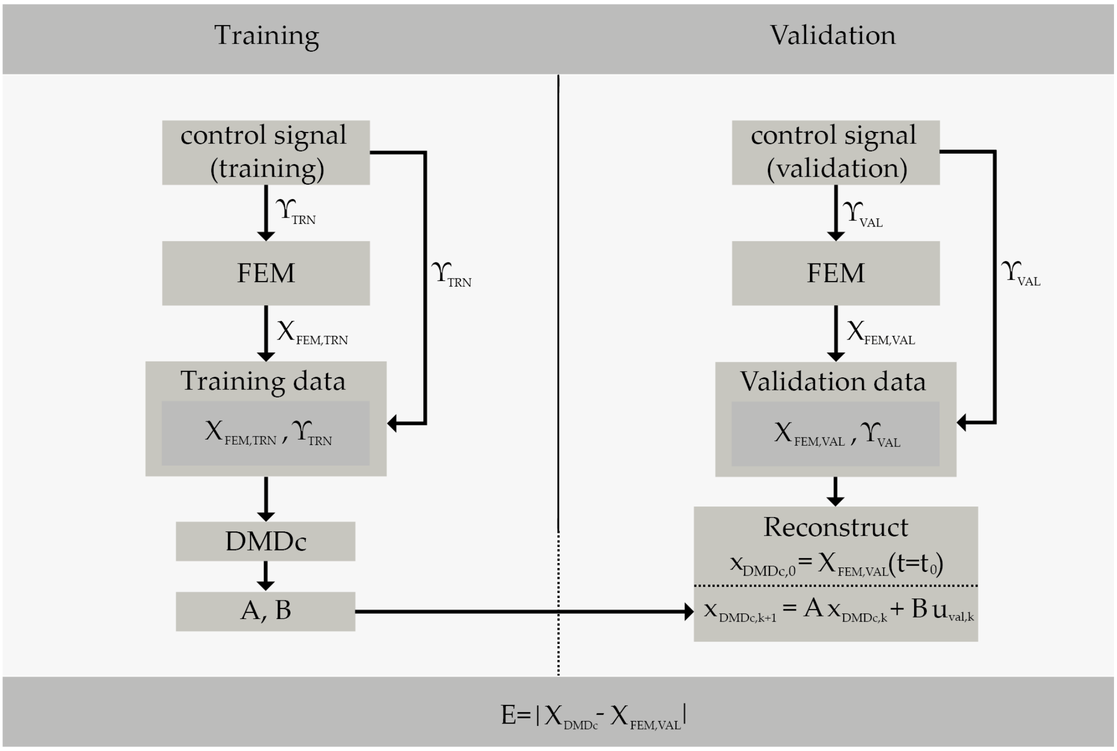

Training
Validation
control signal (training)
Υ_{TRN}
FEM
Υ_{TRN}
$X_{\text{FEM,TRN}}$
Training data
$X_{\text{FEM,TRN}}$, Υ_{TRN}
DMDc
A, B
control signal (validation)
Υ_{VAL}
FEM
Υ_{VAL}
$X_{\text{FEM,VAL}}$
Validation data
$X_{\text{FEM,VAL}}$, Υ_{VAL}
Reconstruct
$x_{\text{DMDc},0} = X_{\text{FEM,VAL}}(t=t_0)$
$x_{\text{DMDc},k+1} = A\,x_{\text{DMDc},k} + B\,u_{\text{val},k}$
$E = |X_{\text{DMDc}} - X_{\text{FEM,VAL}}|$

Figure 4.6: Flowchart of the training and validation process. Training involves generating training data with FEM and deriving system matrices using DMDc. Validation uses these matrices to reconstruct state data, comparing it to actual validation data to compute error $E = |X_{\text{DMDc}} - X_{\text{FEM,VAL}}|$.

To ensure that the selected dataset yields the most robust DMDc model across various types of control signals, we systematically validate each trained model against the other considered datasets. Ultimately, we select the optimal DMDc model based on the error metric *EM*, as outlined in Eq.4.3. This error metric involves initially calculating the mean error across the spatial domain, where n denotes the number of spatial degrees of freedom. Subsequently, we compute the mean error over time, with m representing the total number of discretized time instants in the dataset.

$$\text{EM} = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{n}\sum_{j=1}^{n}E_{ij}\right) \tag{4.3}$$

From Table.4.2 we find that the DMDc model resulted from control signal 2 is the most robust when validated against all the other data.

		Validation				
		1	2	3	4	Σ(EM)
Training	1	0.0164	0.0241	0.0206	0.0166	0.0893
	2	0.0154	0.0171	0.0214	0.0166	0.0705
	3	0.1710	0.3346	0.0194	0.0173	0.5423
	4	0.4647	0.9178	0.0250	0.0184	1.426

Table 4.2: Error metric for the considered datasets.

Key takeaway

- We obtained a ROM data-driven model for the control of the floating structure in regular wave conditions.
- The actual control of the true system will happen in the same conditions as the training data is obtained from.
- We investigated the convergence of the used training datasets by studying the poles and amplitudes of the DMDc dynamic modes.
- Finally, we cross-validated the datasets against each other to get the best data-driven ROM for the actual control.

4.1.2. Irregular wave profile

In this section, we present how we developed a data-driven DMDc model to control a floating structure under irregular wave conditions. Since DMDc assumes that control is applied under the same conditions as those used to generate the training data—meaning the source function remains unchanged—we must address this limitation with a novel approach.

First, we explain how, by using a specific type of simulation for the training datasets, we can capture the dominant dynamics of the floating structure. This enables us to apply control across various sea states. This approach is primarily guided by diffraction theory, which we will detail in this section. Additionally, we will study the wet modes of the floating structure, which will become crucial, as will be evident through diffraction theory.

As with the case under regular wave conditions, we will perform a convergence study to ensure that the datasets used provide converged DMDc models. Furthermore, we will verify that the training data inherently captures the wet modes. By applying Proper Orthogonal Decomposition (POD), as demonstrated in previous studies, we will show that these wet modes are indeed present in our data.

Lastly, we will validate the training data using the process outlined in Fig. 4.6. Additionally, we will investigate the control horizon length to ensure that within this horizon, the data-driven model is primarily influenced by hydrodynamic forces rather than by incoming waves.

Limitations of DMDc: mismatched source function

It is important to mention the limitations of DMD(c). DMD, and hence DMDc, lacks the ability to generalize outside the training dataset. As can be seen in Fig.4.7, the data is generated from, assumingly, an unknown potentially nonlinear function. In our case we know that this function is directly dependent on the incoming wave that excites the structure. As mentioned before we will be controlling the floating structure by only considering the dynamics of the floating structure in the DMDc, that is $\mathbf{x} = \boldsymbol{\eta}$. We will justify this approach using the diffraction theory.

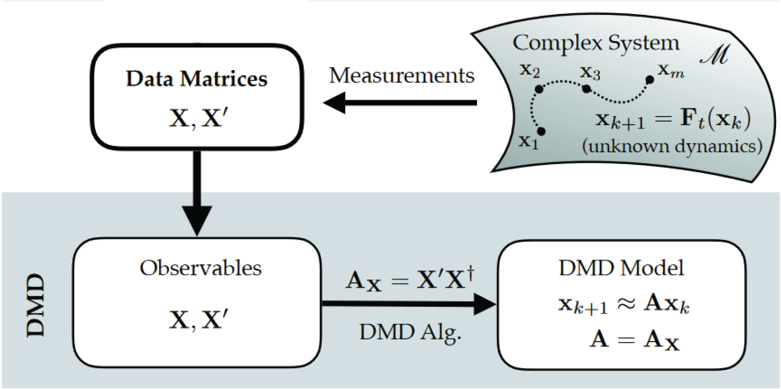

Data Matrices
$\mathbf{X}, \mathbf{X}'$
Measurements
Complex System $\mathscr{M}$
$\mathbf{x}_2$
$\mathbf{x}_3$
$\mathbf{x}_m$
$\mathbf{x}_1$
$\mathbf{x}_{k+1} = \mathbf{F}_t(\mathbf{x}_k)$
(unknown dynamics)
DMD
Observables
$\mathbf{X}, \mathbf{X}'$
$\mathbf{A}_\mathbf{X} = \mathbf{X}'\mathbf{X}^\dagger$
DMD Alg.
DMD Model
$\mathbf{x}_{k+1} \approx \mathbf{A}\mathbf{x}_k$
$\mathbf{A} = \mathbf{A}_\mathbf{X}$

Figure 4.7: The concept of DMD where it is shown how the DMD is an equation-free method. Hence, the DMD inherently does not know the physics behind the data, rather it captures the dynamics solely from the given data [23].

Velocity potential and the wet-modes

In the considered problem we solve for the velocity potential ϕ which, according to the linear potential flow theory, can be decomposed in an incident, transmitted and reflected components denoted by ϕ_I, ϕ_T and ϕ_R respectively. Let us investigate how these separate components influence the response of the system and their relation to their wet-modes.

The incident velocity potential, ϕ_I, is directly correlated to the incoming wave that excites the structure. Hence, this component can not be influenced by the structure and is independent of the FSI. From Eq. 4.5 we see that this indeed is the case since the wet-modes of the structure are independent of the excitation term. The transmitted velocity potential, ϕ_T, can be seen as the component that passes by the floating structure if one would consider the structure to be fixed and, somehow, not to interact with the fluid. Both the incident and transmitted velocity potential are due to environmental and excitation loads [8].
The reflected velocity potential ϕ_R is the component that arises due to the interaction of the structure with the fluid, but also due to the added mass and added damping. Here we can see how the coupling between the structure and fluid arises: the incoming wave not only applies a load on the structure but gives also rise to deformation of the structure, which consecutively leads to reflected waves.

This phenomena is depicted in Fig. 4.8. Hence, as stated earlier, the goal of the control is not to tame the incoming wave that excites the structure, but to control the dynamics of the floating structure. Assuming that the incoming wave that excites the structure is an unknown and difficult to predict, we should then mainly focus on the dynamics and forces that arise due to the reflected wave. From the wet-mode analysis we know that our wet-modes capture the added mass and added damping, which have large influence on the reflected velocity potential. It is therefore more likely that a floating structure with large dimensions and high inertia can be controlled more easily than floating structure with relatively small inertia, since the dynamics of small inertia structures will be mainly dominated by incoming waves.

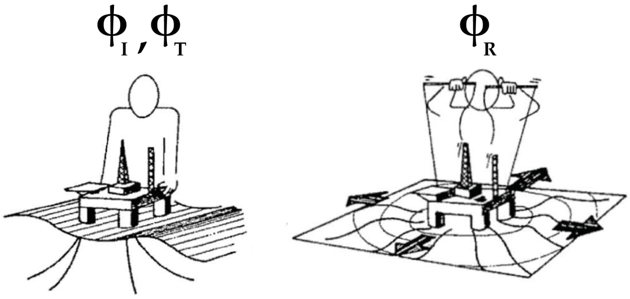


Figure 4.8: Illustration of velocity potential components for a floating structure: ϕ_I represents the incident wave potential, ϕ_T, denotes the transmitted wave potential, and ϕ_R indicates the reflected wave potential around the structure [2].

Since our focus is on floating structures in offshore environments, we will primarily have to deal with various sea states. Our main objective is to control floating structures affected by waves with irregular patterns. Therefore, as outlined in section 4.1, we need to generate data that generalizes effectively to ensure that the ROM can be applied to different sea states. For this purpose we will try to incorporate the wet modes of the floating structure in our training dataset. A flat load spectrum is employed to excite the floating structure. We justify this approach which has been used in an earlier study where a flat load spectrum has been applied on a ship-like structure to obtain the wet-modes using POD [39].

Despite the fact that the actual wet modes are non-orthogonal due to their interdependencies caused by FSI, we can still identify these non-orthogonal modes using POD, which returns orthogonal modes. This may sound contradictory, but since POD is data-driven, it tries to find the modes that capture the most energy in the system. The implemented POD provides a functional basis that captures more energy than any other orthogonal basis. In this way, even though POD returns orthonormal modes, these modes are the most effective in approximating the energy-dominant wet modes present in the system [39].

By ensuring that the wet modes are inherently present in the training data of the DMDc, we hypothesize that the DMDc will perform well, especially when the length of the control horizon is chosen such that the predictions in the MPC are mostly dominated by the hydrodynamic forces. This concept may seem confusing, so let's clarify some key aspects of MPC.

MPC uses the current state of the floating structure, $\mathbf{x}_k$, to predict the dynamics over the control horizon N_c. Based on this prediction, it calculates a control input, $\mathbf{u}_k$, to influence the system at the next state, $\mathbf{x}_{k+1}$, in a way that meets the specified objectives. It is crucial that the model used by MPC for prediction closely matches the actual dynamics of the system.

However, since the sea state is variable, the DMDc model may result in significant mismatches due to changes in the source function. MPC addresses this challenge by selecting a control horizon where the system dynamics are primarily dominated by hydrodynamic forces, which are closely related to the wet modes of the structure. If the control horizon is too long, incoming waves will begin to influence the dynamics of the floating structure, leading to incorrect control inputs and poor performance.

By setting an appropriate control horizon, MPC can compute a control input that accurately reflects the actual system dynamics. The system then reacts, and the subsequent state is measured, allowing us to ”correct” the model mismatch in each cycle by using real-time data from the true system.

The Fig. 4.9 shows the dynamics of a floating structure under variable sea state conditions. The upper row represents the *MPC model forecast*, while the lower row shows the *true system evolution*. The control horizon, N_c, is defined such that the predictions made by the MPC (upper row) are dominated by hydrodynamic forces, which are closely related to the wet modes of the structure. The green areas in the forecast represent regions where the dynamics are influenced

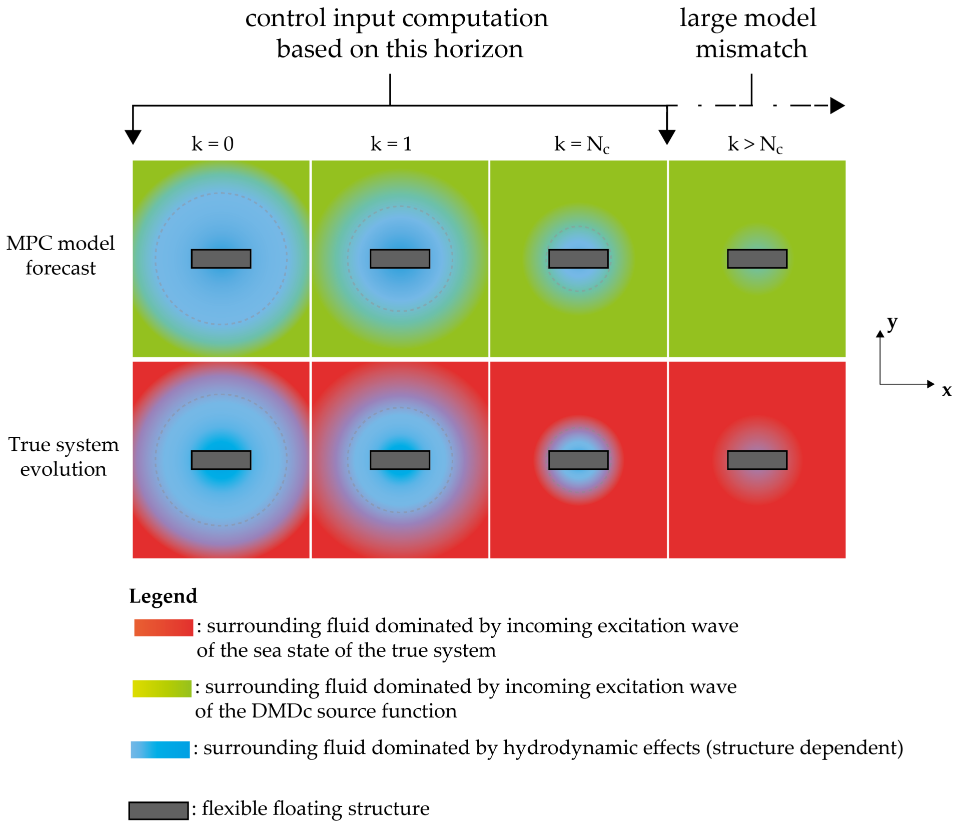

control input computation
based on this horizon
large model
mismatch
k = 0
k = 1
k = N_c
k > N_c
MPC model
forecast
True system
evolution
y
x
Legend
: surrounding fluid dominated by incoming excitation wave of the sea state of the true system
: surrounding fluid dominated by incoming excitation wave of the DMDc source function
: surrounding fluid dominated by hydrodynamic effects (structure dependent)
: flexible floating structure

Figure 4.9: Comparison between MPC model forecast and true system evolution for different time steps within and beyond the control horizon.

by the DMDc source function, and the blue areas show where hydrodynamic forces dominate. The true system, however, is influenced by the actual sea state (indicated in red) beyond $k = N_c$. When the control horizon is too long (for $k > N_c$), the incoming waves begin to penetrate the dynamics of the floating structure, leading to a significant mismatch between the predicted and true system states. This mismatch is visually represented by the increasing presence of red regions in the true system row beyond the control horizon. As MPC operates, it corrects the model mismatch by incorporating real-time measurements of the system's state, though this is not explicitly shown in the figure.

The imposed excitation frequencies are uniformly distributed over a range from 0.5 rad/s to 5.0 rad/s, with each frequency step incremented by 0.01 rad/s. We assign the same amplitude to all the frequencies and assign an uniformly distributed random phase shift to each frequency. Below you can find the used setup for obtaining the training data.

Wet-modes

We need to find the actual wet-modes of the floating structure such that eventually we can compare these theoretical wet-modes with the POD modes of the training data, to ensure the validity of the used approach with the flat load spectrum.

We consider the weak form as defined in Eq.A.30 in the frequency-domain by making use of the separation of variables where we assume that our unknown quantities to solve for have the form $\phi(x,z,t) = \overline{\phi}(x,z)e^{-i\omega t}$, $\eta(x,t) = \overline{\eta}(x)e^{-i\omega t}$ and $\kappa(x,t) = \overline{\kappa}(x)e^{-i\omega t}$, where the space-dependent quantities are complex-valued [12]. Furthermore, we consider the steady-state, that is $t \to \infty$, and substitute the assumed quantities in similar fashion as the weak form defined in

Parameter	Symbol	Value	Unit
Total length of the structure	L_{str}	12.5	m
Parameter relative locations of the joints	β_j	0.2	$-$
mass structure per water density	$\frac{\rho_b h_b}{\rho_w}$	8.36	m^3
Rigidity of the structure modules	(EI_1, EI_2)	$(47100, 47100)$	$N\,m$
Water depth	H	10.0	m
Gravitational constant	g	9.81	$m\,s^{-2}$
Water density	ρ_w	1025	$kg\,m^{-3}$
Joint stiffness per unit length	k_φ	$0.0 \cdot \frac{EI}{L_{str}}$	$kg\,N\,m^{-3}$
Wave amplitude (one for all frequencies)	κ	0.001	m
Sea state	$-$	Flat spectrum	$-$

Table 4.3: Parameters used to obtain the data for the irregular wave case.

Eq.A.30, to obtain the following matrix-vector we need to solve for

$$\begin{bmatrix} -\omega^2\mathbf{M}_{11} + \mathbf{K}_{11}^d + \mathbf{K}_{11}^w & -i\omega\mathbf{C}_{12} & 0 \\ i\omega\mathbf{C}_{21} & \mathbf{K}_{22} & \mathbf{C}_{23} \\ 0 & \mathbf{C}_{32} & \mathbf{K}_{33} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_P \\ \boldsymbol{\phi}_P \\ \boldsymbol{\kappa}_P \end{bmatrix} = \begin{bmatrix} \mathbf{F}_\eta \\ \mathbf{F}_\phi \\ \mathbf{F}_\kappa \end{bmatrix} \tag{4.4}$$

$$\begin{aligned}
\mathbf{M}_{11} &= \left(\frac{\rho_b h_b}{\rho_w}\eta, v\right)_{\Gamma_{str}} \\
\mathbf{K}_{11}^d &= \left(\frac{D}{\rho_w}\Delta\eta, \Delta u\right)_{\Gamma_{str}} - \left(\left\langle \frac{D}{\rho_w}\Delta\eta \right\rangle, \|\nabla u \cdot \mathbf{n}_\Lambda\|\right)_{\Lambda_B} - \left(\left\langle \frac{D}{\rho_w}\Delta u \right\rangle, \|\nabla\eta \cdot \mathbf{n}_\Lambda\|\right)_{\Lambda_B} \\
&\quad + \left(\frac{D\gamma}{h}\|\nabla\eta \cdot \mathbf{n}_\Lambda\|, \|\nabla u \cdot \mathbf{n}_\Lambda\|\right)_{\Lambda_B} + \left(\frac{k_{\varphi,j}}{\rho_w}\|\nabla\eta \cdot \mathbf{n}_\Lambda\|, |\nabla u \cdot \mathbf{n}_\Lambda\|\right)_{\Lambda_j} \\
\mathbf{K}_{11}^w &= (g\eta, u)_{\Gamma_{str}} \\
\mathbf{K}_{22} &= (\nabla\phi, \nabla w)_\Omega - (ik\phi, w)_{\Gamma_{in},\Gamma_{out}} \\
\mathbf{K}_{33} &= (g\kappa, v)_{\Gamma_{fs}} + (g\kappa, v)_{\Gamma_{str}} \\
\mathbf{C}_{12} &= (\phi, u)_{\Gamma_{str}} \\
\mathbf{C}_{21} &= (\eta, w)_{\Gamma_{str}} \\
\mathbf{C}_{23} &= (i\kappa\omega, w)_{\Gamma_{fs}} + (i\kappa\omega, w)_{\Gamma_{d1},\Gamma_{d2}} \\
\mathbf{C}_{32} &= (i\phi\omega, v)_{\Gamma_{fs}} - (i\phi\omega, v)_{\Gamma_{d1},\Gamma_{d2}}
\end{aligned}$$

where $\boldsymbol{\eta}_P$, $\boldsymbol{\phi}_P$ and $\boldsymbol{\kappa}_P$ are the steady-state degrees of freedom. Furthermore, we know that the wet-modes are intrinsic properties of the system and hence independent of the excitation term. Therefore, we assign a zero right-hand side to perform the wet mode analysis. Note that we added radiation boundary conditions along Γ_{in} and Γ_{out} in the $\mathbf{K}_{22}$ term to damp the reflected and transmitted waves, as similarly was done in the transient formulation of the problem. By the use of elimination of variables we can express the system expressed in Eq.4.4 in terms of the structure deformation $\boldsymbol{\eta}_P$

$$(-\omega^2(\mathbf{M}_{11} + \mathbf{A}) + \mathbf{K}_{11}^d + \mathbf{K}_{11}^w)\boldsymbol{\eta}_P = 0 \tag{4.5}$$

$$\text{where} \quad \mathbf{A} = \mathbf{C}_{12}\mathbf{M}_\phi^{-1}\mathbf{C}_{21}, \quad \mathbf{M}_\phi = \mathbf{K}_{22} - \mathbf{C}_{23}\mathbf{K}_{33}^{-1}\mathbf{C}_{32}$$

which leads to a nonlinear eigenvalue problem because of the frequency dependency found in the added mass $\mathbf{A}$. Note that the fluid also introduces the added stiffness $\mathbf{K}_{11}^{w}$ to the system. Finding the actual wet-modes and the modal shapes of the structure along with their wet natural frequencies require a recursive workflow as demonstrated in [3].

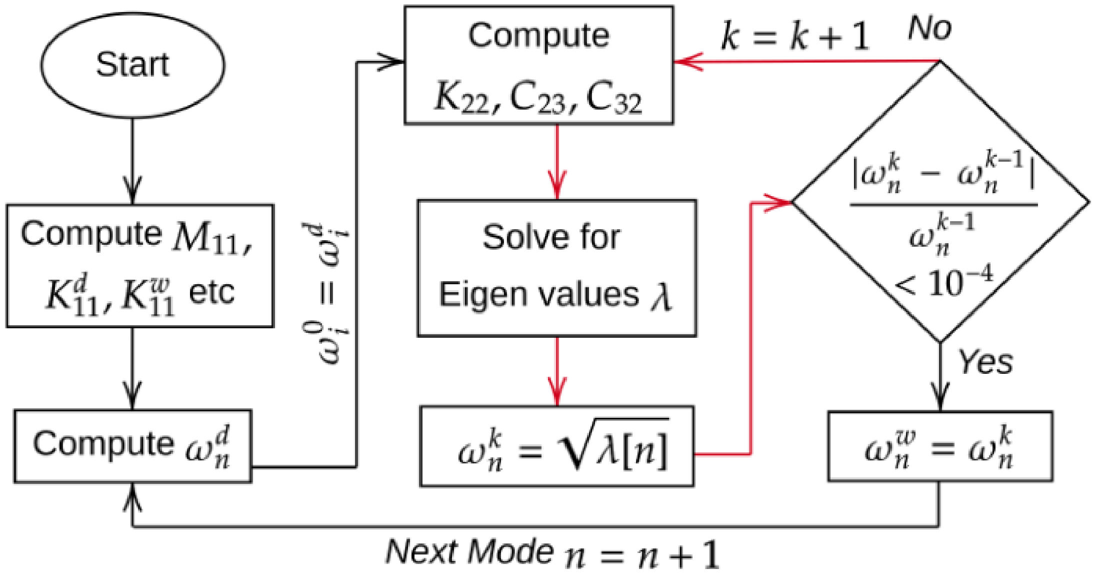

Start
Compute M_{11}, K_{11}^{d}, K_{11}^{w} etc
Compute ω_n^d
$\omega_i^0 = \omega_i^d$
Compute K_{22}, C_{23}, C_{32}
Solve for Eigen values λ
$\omega_n^k = \sqrt{\lambda[n]}$
$\frac{|\omega_n^k - \omega_n^{k-1}|}{\omega_n^{k-1}} < 10^{-4}$
$k = k + 1$
No
Yes
$\omega_n^w = \omega_n^k$
Next Mode $n = n + 1$

Figure 4.10: Recursive algorithm to find the wet-modal shapes and corresponding wet natural frequencies of the floating structure as demonstrated in [3].

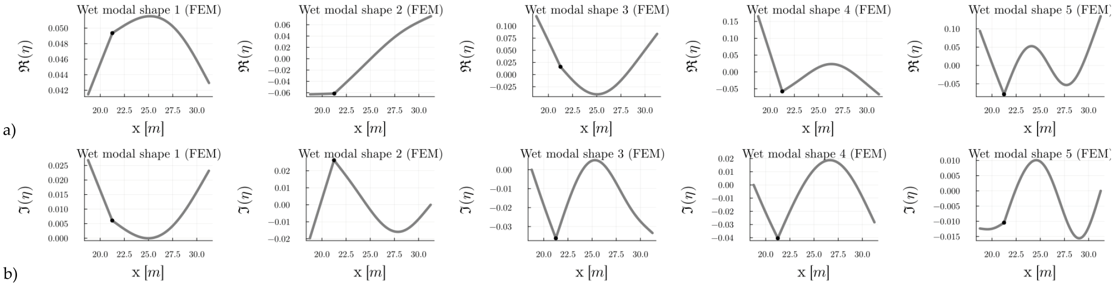


Figure 4.11: The wet-modes of the floating structure with the first four wet natural frequencies $\{\omega_1^{\mathrm{W}}, \omega_2^{\mathrm{W}}, \omega_3^{\mathrm{W}}, \omega_4^{\mathrm{W}}, \omega_5^{\mathrm{W}}\} \approx \{1.11, 1.95, 2.67, 3.29, 5.24\}$ rad/s.

In Fig.4.11 we see the wet-modal shapes accompanied with their wet natural frequencies of the floating structure, given the setup described in Table 4.3.

The next step is to perform the POD on an *uncontrolled* dataset to see if we can identify the theoretical modes of the floating structure in the data. Note that we conduct the POD without any influence of control on the structure, such that we obtain the POD modes of the structure solely due to the interaction with the fluid.

POD modes

A different technique often used to capture a data-driven underlying model of dynamical systems is Proper Orthogonal Decomposition (POD). POD is based on finding a orthonormal coordinate system that explains the largest spatial variance in the data. This coordinate system we call Proper Orthogonal Coordinates (POC) where the dimensions are components of this coordinate system are the so-called POD-modes Ψ_{POD}. Along with these POD-modes we obtain the temporal dynamics such that for a considered $i-$ th POD-mode $\boldsymbol{\Psi}_i$ we have the temporal coefficient $\mathbf{a}_i = \{a_i(t_1), ..., a_i(t_{N_s})\}$. In other words, for each POD-mode we have a signal that describes how strongly this considered mode is present along the considered timespan from t_1 up to t_{N_s}. Finally, this allow us to express the dynamics found in the data using the POD-modes and the corresponding temporal coefficient such that [39]

$$X(t) = \sum_{i=1}^{\infty} a_i(t)\ \mathbf{\Psi}_i. \tag{4.6}$$

To perform the POD, the uncontrolled data matrix X, representing the system's snapshots over time, is first used to compute the covariance matrix $C = \frac{1}{N_s - 1} X^\top X$, where N_s is the number of time samples. Eigenvalue decomposition of C yields the eigenvalues and eigenvectors, from which the POD modes are reconstructed as $\Psi_{\text{POD}} = XV$, where V are the eigenvectors [39].

The temporal coefficients are obtained by projecting the data onto the *sorted* POD modes, that is we sort the POD modes from biggest to smallest eigenvalue, such that the temporal coefficients are stored in the $A_{\text{POD}} = \Psi_{\text{POD}}^\top X$. This matrix A_{POD} consists of rows which are of the length N_s, each row is a signal which describes the evolution of the corresponding mode throughout data. The corresponding dominant frequencies of the POD modes are computed by performing a Fast Fourier Transform on each mode's temporal coefficient and identifying the frequency with the maximum power in the power spectrum, excluding the zero-frequency component. The computed frequencies are then converted to angular frequencies $\omega_{\text{POD}} = 2\pi f_{\text{POD}}$, where f_{POD} is the frequency in Hz [39].

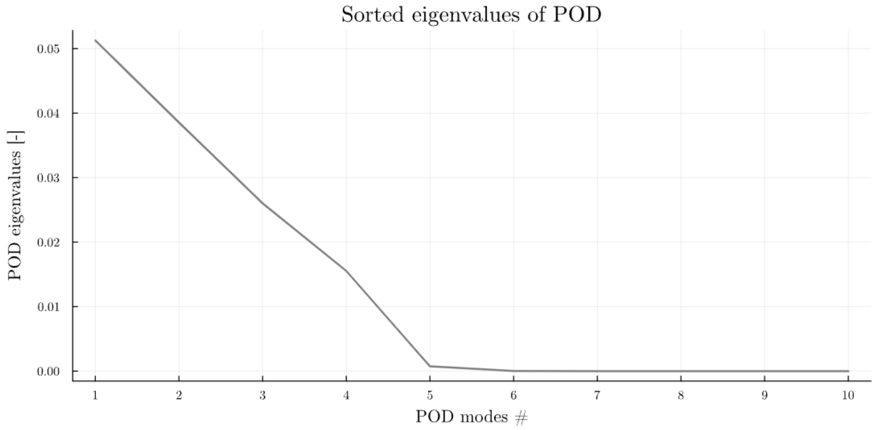


Figure 4.12: This figure displays the sorted eigenvalues obtained from the POD analysis. The eigenvalues represent the energy contribution of each POD mode, with higher eigenvalues indicating modes that capture more of the system's total variance. The rapid decay of the eigenvalues suggests that the dominant dynamics of the system are well-represented by the first few POD modes, while subsequent modes contribute minimally to the overall energy.

From Fig. 4.12 we decide to focus on the five most dominant POD-modes. These are the POD-modes which we are going to compare with the theoretical wet-modes to ensure our approach with the flat load spectrum.

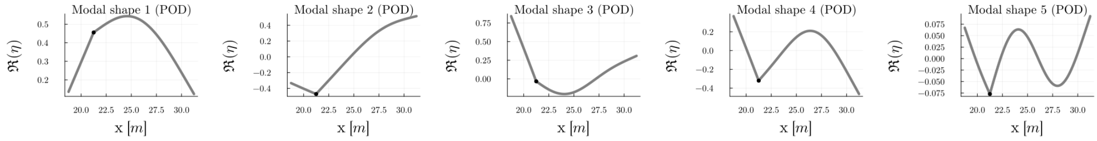


Figure 4.13: The POD-modes of the floating structure with their corresponding five frequencies $\{\omega_1^{\text{POD}}, \omega_2^{\text{POD}}, \omega_3^{\text{POD}}, \omega_4^{\text{POD}}, \omega_5^{\text{POD}}\} \approx \{1.04, 2.00, 2.42, 3.04, 4.76\}$ rad/s

To evaluate the similarity between the POD-modes and the actual wet-modes of the system, we use the Modal Assurance Criterion (MAC). The MAC is a statistical measure commonly used to compare theoretical modes Ψ_A, such as the wet-modes of the floating structure, with experimentally derived modes Ψ_X, like the POD modes of the structure [34].

$$\text{MAC}(r,q) = \frac{\left|\left\{\Psi_A^T\right\}_r \left\{\Psi_X^*\right\}_q\right|^2}{\left(\left\{\Psi_A^T\right\}_r \left\{\Psi_A^*\right\}_r\right)\left(\left\{\Psi_X^T\right\}_q \left\{\Psi_X^*\right\}_q\right)}. \tag{4.7}$$

In Eq. 4.7, we use notation that allows us to consider the MAC index between the r-th theoretical mode and the q-th experimental mode. The MAC index lies between 0 and 1, where values near zero indicate low resemblence between the modes and values near 1 indicate high resemblence between the modes. Moreover, it should be clearly noted that the MAC index only takes the shape of the modes into account and hence one should additionally compare the frequencies of the modes.

		$\Psi_{X,q}$				
		1	2	3	4	5
$\Psi_{A,r}$	1	0.90	-	-	-	-
	2	-	0.96	-	-	-
	3	-	-	0.81	-	-
	4	-	-	-	0.65	-
	1	-	-	-	-	0.86

Table 4.4: MAC index between POD modes and wet-modes of the floating structure.

The excitation of the floating structure with a flat load spectrum indeed seems to activate the wet-modes in the data, which can be validated by the MAC-indices of the most dominant 5 POD-modes in table 4.4. The first two POD-modes correspond the most with the theoretical wet-modes in shape and in frequency. This indicate to a great extent that our data inherently contains the dynamics induced by the hydrodynamic force, which is as mentioned earlier essential to control the floating structure in various sea states.

After having validated the flat-load spectrum using POD, we can now continue to generating training data using the flat-load spectrum approach along with open-loop control signals.

Similar to the regular case we need to construct open-loop control signals to obtain a DMDc model. Let us therefore present the used open-loop control signals.

Control signal: 1

Since we are now dealing with a floating structure in irregular wave conditions it is more likely that the control signal during the actual control will be correlated with the natural frequencies of the floating structure. Therefore, similar to the regular wave condition case, we construct the first control signals using 5 regions with an oscillation. These 5 oscillating regions correspond to the natural frequency ω_n^w in ascending order from the start of the open-loop control signal towards its end.

Control signal: 2

Finally, we construct the second control signal using the same reasoning as for the regular wave condition. In the third control signal we again start with a zero value before entering the steady-state portion of the FEM. As soon as we enter this steady-state portion we apply an open-loop control signal which is sinusoidal and decays in time. The frequency of this first decaying sinusoidal signal is 0.125 rad/s. After some time we start applying different linear functions following each other. We then again apply a decaying sinousidal signal with 0.25 rad/s as frequency. Finally, we apply again different linear functions. Note that for this third control signal we use relatively small frequencies to ensure generalization for calm sea states, which most likely require a control signal with a low-frequency.

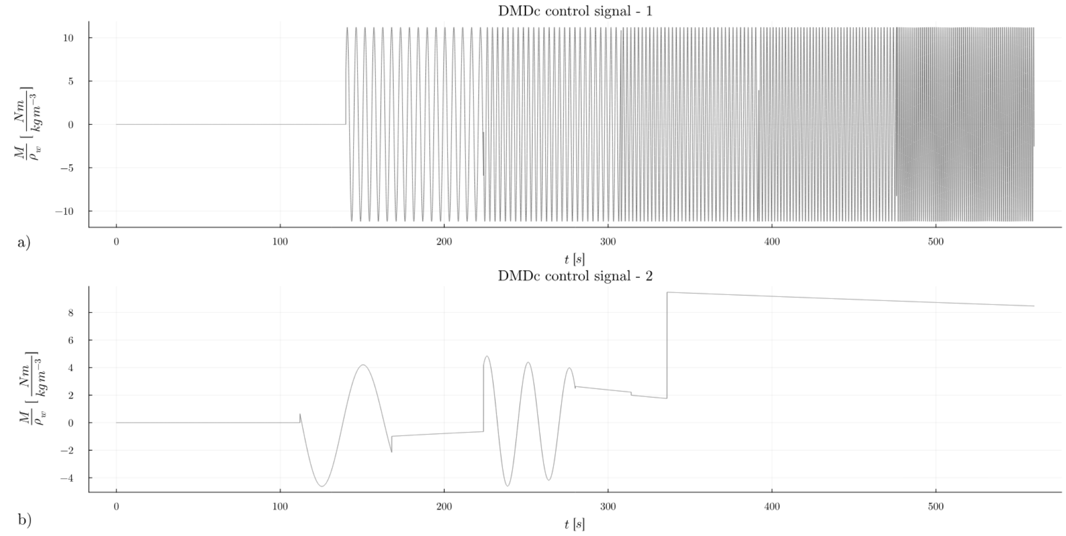


Figure 4.14: The two open-loop control signals used along with a flat-load spectrum to obtain training data for the DMDc. a) An open-loop control signal consists out of 5 regions oscillating at the first 5 natural frequencies of the wet-modes in ascending order. b) An open-loop control signal with random patterns such as low-frequency decaying oscillations and step functions.

Convergence of the DMDc model

Similar to the regular wave case, we first need to determine the number of components, read the DMDc modes, required to adequately capture the dynamics present in the training data using the ROM obtained from DMDc. Again, we take r to be equal to $\tilde{r}$. For this case, we choose to consider 20 DMDc modes, as Fig. 4.15 shows that these 20 modes capture most of the variance in the data. It's important to note that the resulting ROM will eventually be applied to a different sea state, so overfitting would not improve the results.

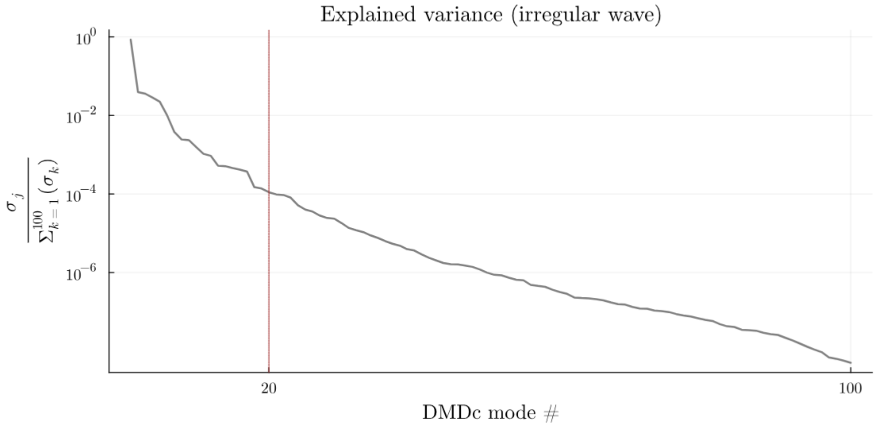


Figure 4.15: This plot shows the cumulative explained variance of the DMDc modes as a function of the mode number for the irregular wave dataset. The red vertical line at mode 20 marks the cutoff, beyond which additional modes contribute minimally to the explained variance.

Having found that 20 DMDc modes explain most of the variance found in the data, we now continue to check if the DMDc model converged given the number of snapshots used in the data. Similarly to the regular wave case we check for the convergence of the eigenvalues and the amplitudes of the eigenvalues. For the convergence of the eigenvalues we mainly investigate the eigenvalues near the boundary of the unit circle, as depicted in Fig. 4.16, since these are the eigenvalues that have little damping and will most likely dominate the undamped dynamics of the system. Finally, we check for the convergence of the first 15 DMDc modes with the largest amplitudes.

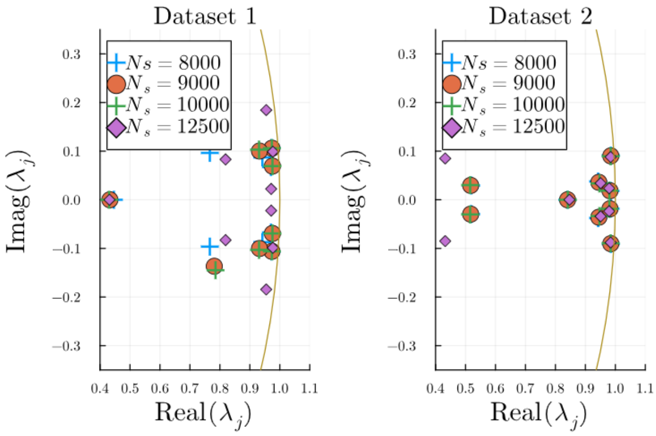


Figure 4.16: Distribution of DMDc eigenvalues obtained from the flat-load spectrum training datasets. DMDc eigenvalues in the complex plane for datasets 1 and 2, with sample sizes N_s ranging from 8000 to 12500.

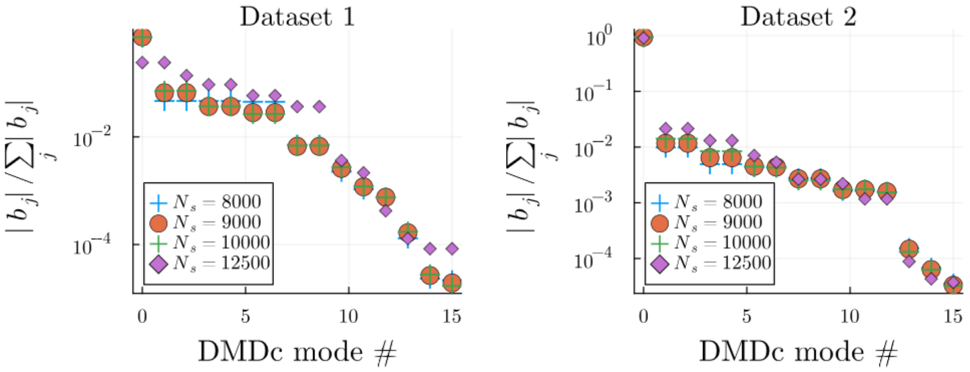


Figure 4.17: This figure shows the normalized amplitudes $|b_j|/\sum_j |b_j|$ for each DMDc mode in two datasets with varying sample sizes N_s (8000, 9000, 10000, and 12500). The similarity in mode amplitudes across different sample sizes suggests consistency in the dominant modes captured in each dataset.

It is evident from Figs. 4.16 and 4.17 that the first dataset doesn't seem to have converged. This already indicates that the used open-loop control signal didn't capture all of the dynamics in the dataset. However, if we look at the second dataset we see a much better convergence of the eigenvalues and amplitudes of the DMDc modes. In that case this should indicate that the used open-loop control signal did manage to capture most of the relevant dynamics of the true system and the relation between the control signal and the states.

Now to ensure that the second dataset indeed manages to return a DMDc model that captures the dynamics of the flat-load excited structure, we will have to validate the returned DMDc model using datasets obtained in a sea state condition. This is necessary to ensure that our DMDc model in various sea states will manage to return satisfactory results.

Validation of the DMDc model

This case requires a different approach to validate the DMDc model since it should be capable of generalizing for various sea states. Therefore, we consider the validation as follows: validate the converged DMDc model against a complete different dataset with a different source function. Hence, our validation dataset is obtained from a FE simulation where the floating structure is excited using a JONSWAP spectrum sea state.

The described validation will be carried out in the following steps:

1. Given different ”initial” states from the validation dataset we forecast the prediction using the DMDc model. These ”initial” states represent the current state $\mathbf{x}_k$ which is obtained from the true system in the actual control.

2. We compute the error between the forecast of the DMDc model and the actual validation dataset.
3. We determine the length of N_c which returns satisfactory forecasting results such that the model-plant mismatch in the actual MPC is kept minimum.

Hence, we ask ourselves the question: *given the current state $\mathbf{x}_k$ of the true system, how long should N_c be such that the hydrodynamic forcing is still present?* This will make sure that the obtained DMDc model, in theory, can be used for the control of flexible floating structures in various sea states.

We consider four validation datasets obtained from a FE simulation in different sea states. Along with each dataset we obviously have a control signal which is a part of the validation. We evaluate the performance of our DMDc model at three different time-instants in the validation datasets. We then reconstruct the sea state dataset along a horizon of 50 steps long and compute the following relative error

$$\text{Relative Error} = \frac{|X_{\text{sea}} - X_{\text{DMDc}}|}{\|X_{\text{sea}}\|_{\infty}} \tag{4.8}$$

where X_{sea} represents the validation datasets and X_{DMDc} represent the constructed dataset using the DMDc model. This relative error is a matrix containing the relative error in space and time. Using a simple heatmap we can investigate how long N_c should be such that the dynamics of the DMDc model, obtained from the flat-load excited structure, doesn't deviate too much from a sea state excited floating structure. Again, this assumption is based on the diffraction theory where we try to find N_c where the hydrodynamic forces are the most dominant.

The used validation control-signals can be found in Fig.4.18.

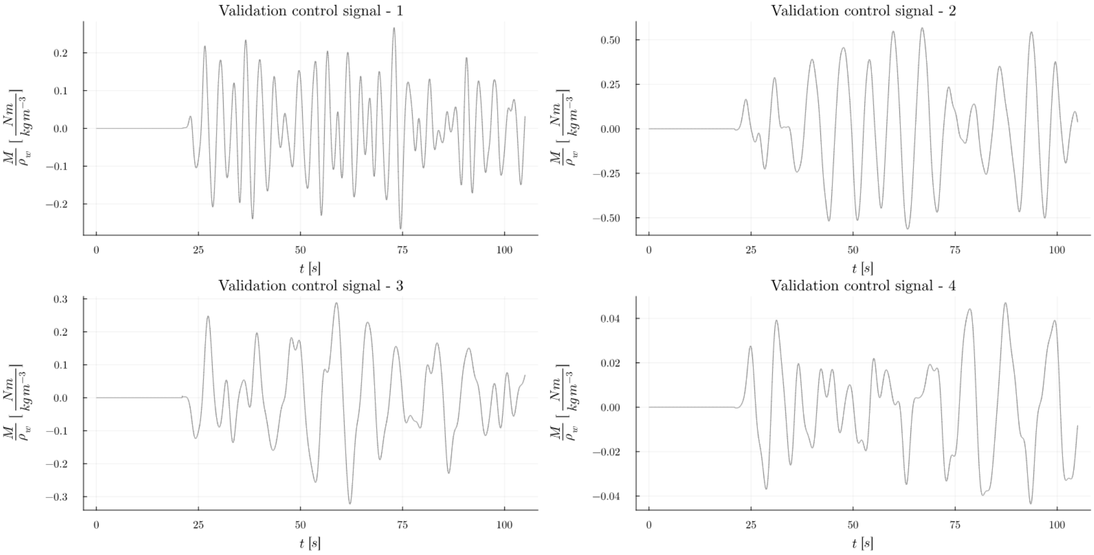


Figure 4.18: This figure shows the validation control signals for four datasets with different peak wave periods: $T_p = 4.0$ s, $T_p = 8.0$ s, $T_p = 10.0$ s, and $T_p = 12.0$ s for plots 1 to 4, respectively.

Note how in Fig. 4.19 we indeed observe that the error for all validation datasets is nearly zero close to the "initial" state we consider at those time-instants. Especially up to 5 steps ahead there is nearly any error. This corresponds to the theory shown in Fig. 4.8. Because N_c is a bit too short for control, we decide to have a control horizon $N_c = 10$, which should still give satisfactory results according to Fig. 4.19.

Note how after 10 steps of prediction the relative error increases significantly and propagates along the length of the floating structure. This clearly shows that a long control horizon will lead to bad forecasting in the MPC, leading to incorrect control.

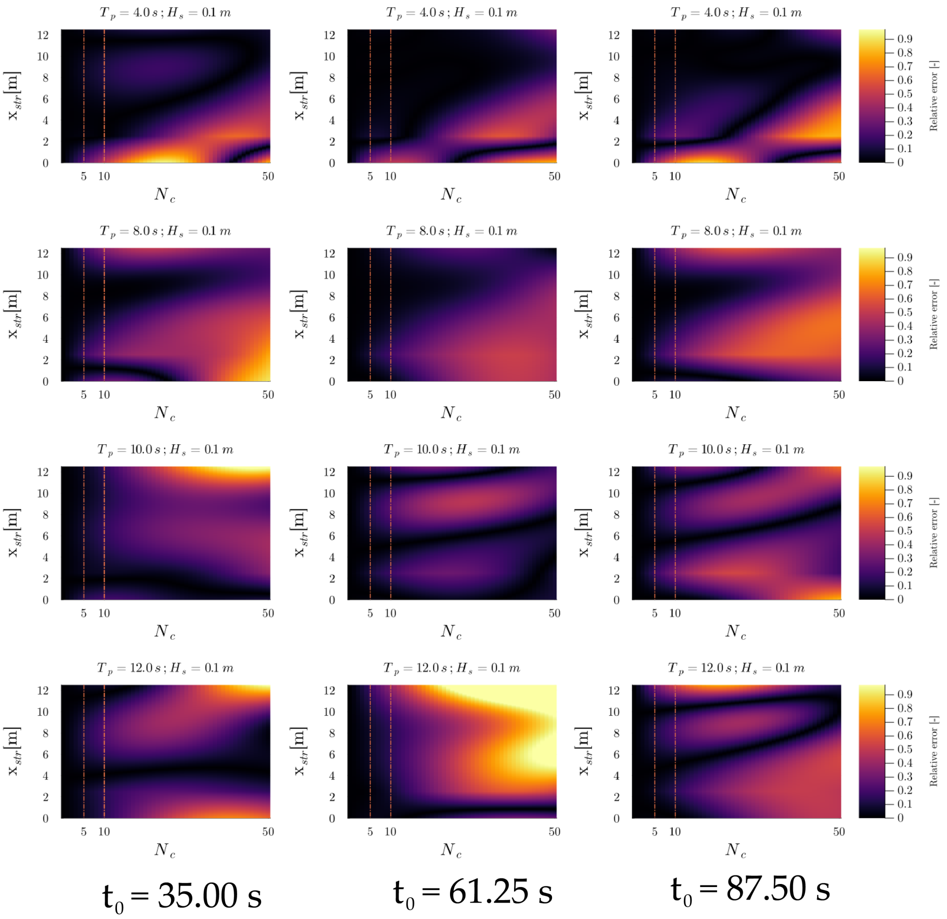


Figure 4.19: Relative error heatmaps between DMDc model predictions and FE simulation datasets with peak periods $T_P = \{4.0, 8.0, 10.0, 12.0\}$ s and wave height $H_s = 0.1$ m. Columns correspond to time instants $t_0 = 35.00$ s, 61.25 s, 87.50 s, while the x-axis shows horizon length N_c over 50 steps. These heatmaps indicate the optimal N_c for maintaining hydrodynamic force dominance and minimizing model-plant mismatch across sea states.

Key takeaway

- We built a DMDc model for controlling a floating structure in irregular wave conditions. This involved designing simulations that capture the structure's dominant dynamics.
- Through diffraction theory and Proper Orthogonal Decomposition (POD), we confirmed the presence of wet-modes in our data, ensuring the training dataset captures key dynamics.
- We verified model convergence by examining eigenvalues and mode amplitudes, selecting 20 DMDc modes as sufficient to represent the system's variance.
- Validated the model against different sea states to find an optimal control horizon, N_c, ensuring predictions stay dominated by hydrodynamic forces, improving generalization across sea states.

4.2. Cost function

In this study we will present the case where we opt to control the rear part of the floating structure as illustrated in Fig. 4.20. By control we mean reducing the motion of the rear part as much as possible, i.e. stabilizing the rear part of the structure. We opt to show that by using the front part as a flapping foil we can reflect some of the energy due to the incoming wave and stabilize the rear part. Let us first define the reference trajectory in the full order space at one time instant k as follows

$$\mathbf{r}_{\text{rear}} = \mathbf{0}_{1\times n_{\text{rear}}} \tag{4.9}$$

Hence, our reference trajectory in the full space is a vector filled with zeros that has the length n_{rear} which is the number of degrees of freedom associated to the rear part of the structure. Since the control will be happening in the reduced space we need to express the reference trajectory in Eq. 4.9 in the reduced space as well. Let us recall Eq. 3.60 where we showed how for any subset of the full state we can use the transformation matrix $\hat{U}_{\text{sub}}$ to obtain the reduced order substate. Hence, for the rear part we can compute the transformation matrix of this subset, which we call $\hat{U}_{\text{rear}}$. From here it follows that the reduced order signal we want to control, which is the rear part of the floating structure, is formulated as

$$\tilde{\mathbf{p}}_{\text{rear}} = \hat{U}_{\text{rear}}^{-1}\mathbf{p}_{\text{rear}} \tag{4.10}$$

Finally, we can compute the signal trajectory along our control horizon by using our prediction matrix $\mathcal{M}$ as derived previously

$$\underline{\tilde{\mathbf{p}}}_{\text{rear}} = \mathcal{M}\tilde{\mathbf{p}}_{\text{rear}} \tag{4.11}$$

Moreover, it should again be clearly stated that we are able to reconstruct the full state measurements by using the Kalman filter. Hence, in order to compute the actual reference trajectory of our plant in the reduced space from the full state measurements we follow Eq.3.61 such that

$$\tilde{\mathbf{p}}_{\text{rear}} = \underbrace{\hat{U}_{\text{rear}}^{-1}C\hat{U}}_{\tilde{C}}\,\tilde{\mathbf{x}} \tag{4.12}$$

Having derived these quantities for the rear part we can follow what is derived in 3.4.

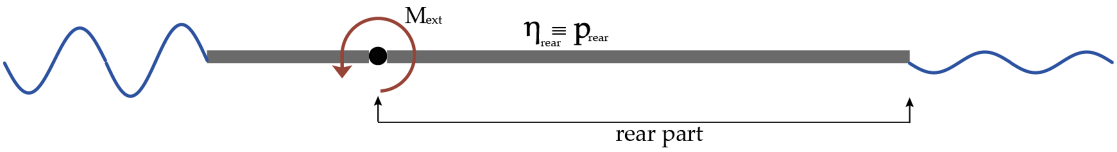


Figure 4.20: The considered signal we opt to control is associated to the degrees of freedom of the approximated FE solution η_h such that $\boldsymbol{\eta}_{rear} \in \boldsymbol{\eta}$. We find $\boldsymbol{\eta}_{rear}$ by the use of a selection matrix C operating on the full-state $\mathbf{x}$ of the plant such that $\mathbf{p}_{rear} = C\mathbf{x}$.

4.3. Control algorithm

In Algorithm 1 we find the pseudocode for the implementation of the MPC to control the FEM plant. We implement the ROM in the MPC to obtain our control signal. The control signal is found by solving an unconstrained or constrained optimization problem.

Algorithm 1 MPC for controlling FEM plant using the ROM

```
procedure MPC-FEM($U, V, \Delta t, \gamma_t, \beta_t$)
    Define weak formulations: $m(t,\cdot), c(t,\cdot), a(t,\cdot), b(t,u,\cdot)$
    Setup Gridap Newmark solver: ode_solver = Gridap.Newmark(LUSolver(), $\Delta t$, $\gamma_t$, $\beta_t$)
    Initialize:
        $k = 1$
        Choose $u_{k-1} = 0$
        op_init = Gridap.TransientAffineFEOperator($m$, $c$, $a$, $b(u_{k-1})$, $U$, $V$)
        $x_{k-1}, v_{k-1}, a_{k-1}$ at $t_{k-1} = 0$
        $x_k \leftarrow$ Gridap.solve(ode_solver, op_init, $(x_{k-1}, v_{k-1}, a_{k-1}), t_{k-1}, \Delta t$ )
        $t_k \leftarrow t_{k-1} + \Delta t$
        Store $x_k, x_{k-1}, v_{k-1}, a_{k-1}$
    repeat
    Compute current state's velocity and acceleration:
        $v_k = \frac{\gamma_t}{\beta_t \Delta t}(x_k - x_{k-1}) + \left(1 - \frac{\gamma_t}{\beta_t}\right) v_{k-1} + \Delta t \left(1 - \frac{\gamma_t}{2\beta_t}\right) a_{k-1}$
        $a_k = \frac{1}{\beta_t \Delta t^2}(x_k - x_{k-1}) - \frac{1}{\beta_t \Delta t} v_{k-1} - \frac{1-2\beta_t}{2\beta_t} a_{k-1}$
    Optimize to find control input $u_k$:
        $\hat{x}_k \leftarrow$ KalmanFilter($y_k$)
        Transform full-order estimated state $\hat{x}_k$ to reduced order estimated state $\tilde{x}_k$
        Find $\underline{\mathbf{u}}$
        $u_k = \underline{\mathbf{u}}(k)$
    Apply control input $u_k$ to plant (FEM) to obtain new state $x_{k+1}$
        op_n = Gridap.TransientAffineFEOperator($m$, $c$, $a$, $b(u_k)$, $U$, $V$)
        $x_{k+1} \leftarrow$ Gridap.solve(ode_solver, op_k, $(x_k, v_k, a_k), t_k, \Delta t$)
    Update state and prediction horizon:
        Store $x_{k+1}, x_k, v_k, a_k$
        $t_k \leftarrow t_k + \Delta t$
        $k \leftarrow k + 1$
    until end condition is met
    return final state and control trajectory
end procedure
```

The algorithm begins by defining the weak formulations for the problem, including the mass matrix m, damping matrix c, stiffness matrix a, and the source vector b which is a function of the control input u. Next, the Gridap Newmark solver is set up with the appropriate parameters. Initialization follows, where the initial control input u_{k-1} is set to zero, and the initial transient affine finite element operator is created by passing the governing matrices and the trial and test FE spaces U and V respectively. The state variables x_{k-1}, v_{k-1}, and a_{k-1} are initialized at time $t_{k-1} = 0$. The initial state x_k is then computed, and the time step is updated.

The iterative process begins with the computation of the current state's velocity and acceleration using the Newmark integration formulas. The control input u_k is found by transforming the full-order estimated state $\hat{x}_k$ to a reduced order state $\tilde{x}_k$, finding the optimal control signal along the control hoirzon $\underline{\hat{\mathbf{u}}}$, and setting u_k to the corresponding value. This control input is applied to the FEM plant to obtain the new true state x_{k+1}. The transient affine finite element operator is updated with the new control input u_k, and the new state x_{k+1} is computed using the Gridap solver. The state variables and the current time are updated, and the time step is incremented. This process repeats until the end condition is met, at which point the final state and control trajectory are returned.

5 Results & Discussion

In this section, we present the results of stabilizing the rear part of a floating structure, as outlined in section 4.2. The results serve as a Proof of Concept for the proposed control framework of flexible floating structures. We consider the control of the floating structure in regular wave condition and in sea state. We first will present the controller's stability and explain the tuning parameters that are used for the design of the controller. We will then precede by presenting the effectiveness of the Kalman filter. Lastly, we present the actual controlled case and show how the Finite Element model was stabilized successfully by presenting a FE defined cost function.

5.1. Control strategy

For the considered actual control we do not perform any sensitivity analysis regarding the MPC parameters since this requires extensive simulations for different MPC parameters to assess the performance of the FEM model in combination with the control. Instead, we use the stability analysis as explained in section 3.4.5, which serves as an indicator of the overall stability and not a definitive measure due to the possible model mismatch between the DMDc model and the actual plant, i.e. FEM model. Nevertheless, the MPC parameters can still be chosen such that our control strategy is satisfied. We define this control strategy as follows

- *We prioritize the control performance by not limiting the used actuator for the flapping motion as long it stays within the boundary of stability, i.e. we want a maximum control signal and the rate of change in the control signal is not constrained.*
    1. Maximize the entries of $\mathcal{Q}$ as long as it respects the stability.
    2. Minimize the entries of $\mathcal{R}_1$ as long as it respects the stability.
    3. Minimize the entries of $\mathcal{R}_2$ as long as it respects the stability.
- *We mimic a real scenario by polluting our measurements from the FEM model such that we prove that our Kalman filter is robust.*

5.2. Regular wave

In this section we consider the control of the flexible floating structure in regular wave conditions where we use the exact same setup used to obtain the DMDc model for this considered case. We first explain the MPC parameters used. We then explain the ramping function which is an intermediate function to smoothly introduce the control input to the system. We then present the performance of the Kalman filter and compare the estimated states with the true states. Finally, we present the results of the control on the actual true system.

5.2.1. MPC parameters

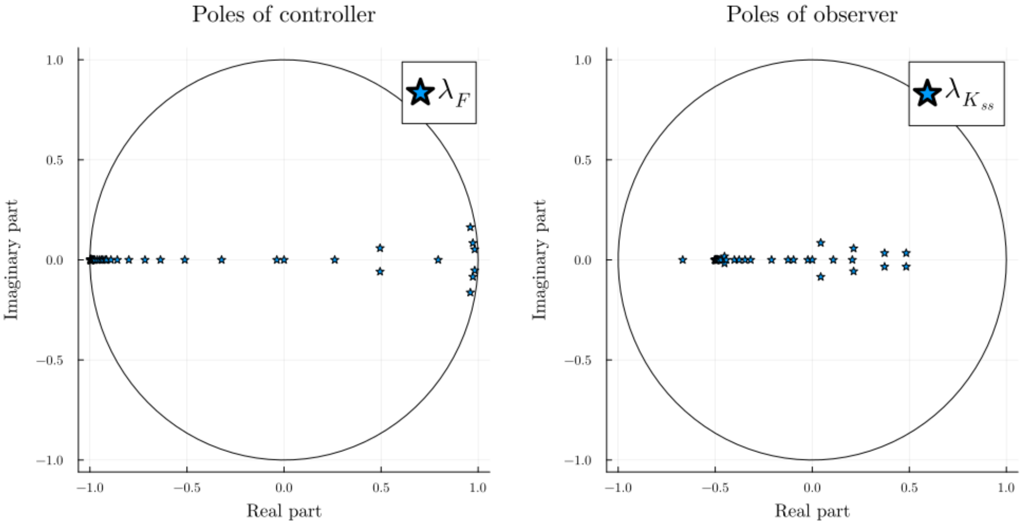


Figure 5.1: Distribution of the poles for the controller and observer within the unit circle, illustrating the stability of the system under the implemented MPC strategy.

We control the true system by exciting our floating platform using the FEM model where we use the exact same parameters as expressed in Table 2.9. Since the used reduced model in the MPC arises from a similar FEM we expect that the model mismatch between the true system and the MPC model to be small.

Moreover, we take the process noise to have a covariance matrix $Q = 100\,I$ and the measurement noise to have a covariance matrix $R = 10\,I$. The choice for these covariance values are not investigated but are chosen such that the observer is stable. We assume that there are no physical constraints on the actuator that applies the external moment as our control input. Hence, for the controller gain we have chosen for an aggressive kind of control. As advised by [6], ζ should be taken as 0 if that does not lead to stability problems. Therefore, we assign $\zeta = 0$. That means that we do not penalize the increments between the control input along the horizon. However, we do penalize the first control input, which will be the actual control input we apply to the system. Therefore, we have taken $\zeta_0 = 10$. Finally, we have chosen $\tilde{\mathcal{Q}} = 50$ to emphasize the tracking accuracy, that is we want our reference to be reached quite fast.

Moreover, we have taken the control horizon and prediction horizon to have the same length of 10 steps. As mentioned before, the system cannot be fully controlled since we are dealing with some uncontrollable modes. Therefore, we know that if we would take the control horizon to be smaller than the prediction horizon, the system would eventually return to the uncontrolled state. Furthermore, the length of 10 steps has shown to effectively control the system and satisfies the stability of the controller. As explained before we are able to assess the stability of our controller by checking the poles as can be seen in Fig. 5.1.

5.2.2. Ramping function

To prevent shocks in the FE model, which is unpleasant since it can lead to instabilities, we introduce a ramping function. This ramping function will make sure that the applied control input is gradually introduced to the system in the initial phase of the control. This ramping function used to saturate the control input in the finite element model (FEM) is defined as follows

$$r(i) = \frac{\tanh\left(4 \cdot \left(\frac{i}{n_{\text{steps}}-1} - 0.5\right)\right) + 1}{2}, \quad \text{for } i = 0, 1, 2, \ldots, n_{\text{steps}} - 1 \tag{5.1}$$

where n_{steps} is the number of steps in the control sequence, and $r(i)$ calculates the value of the ramp function at step i to smoothly transition the control efforts and avoid shocks. The ramping

Table 5.1: Summary of Control Parameters for the MPC Controller

Parameter	**Value**
Process noise covariance matrix, Q	$100I$
Measurement noise covariance matrix, R	$10I$
Physical constraints on actuator	None
Penalization on control increments, ζ (related to $\mathcal{R}_1$)	0
Penalization on initial control input, ζ_0 (related to $\mathcal{R}_2$)	10
Tracking accuracy weight, $\tilde{\mathcal{Q}}$ (related to $\mathcal{Q}$)	50
Control horizon length	10 steps
Prediction horizon length	10 steps
Stability assessment	Fig. 5.1

function is visualised in Fig.5.2.

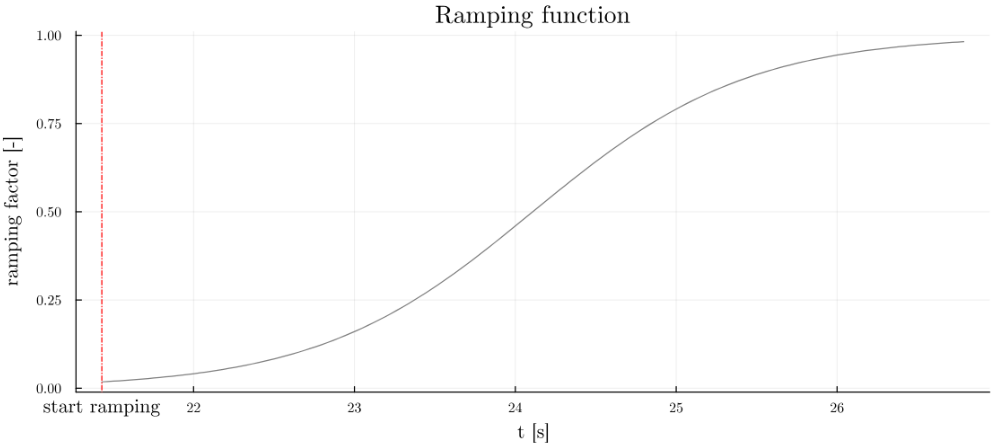
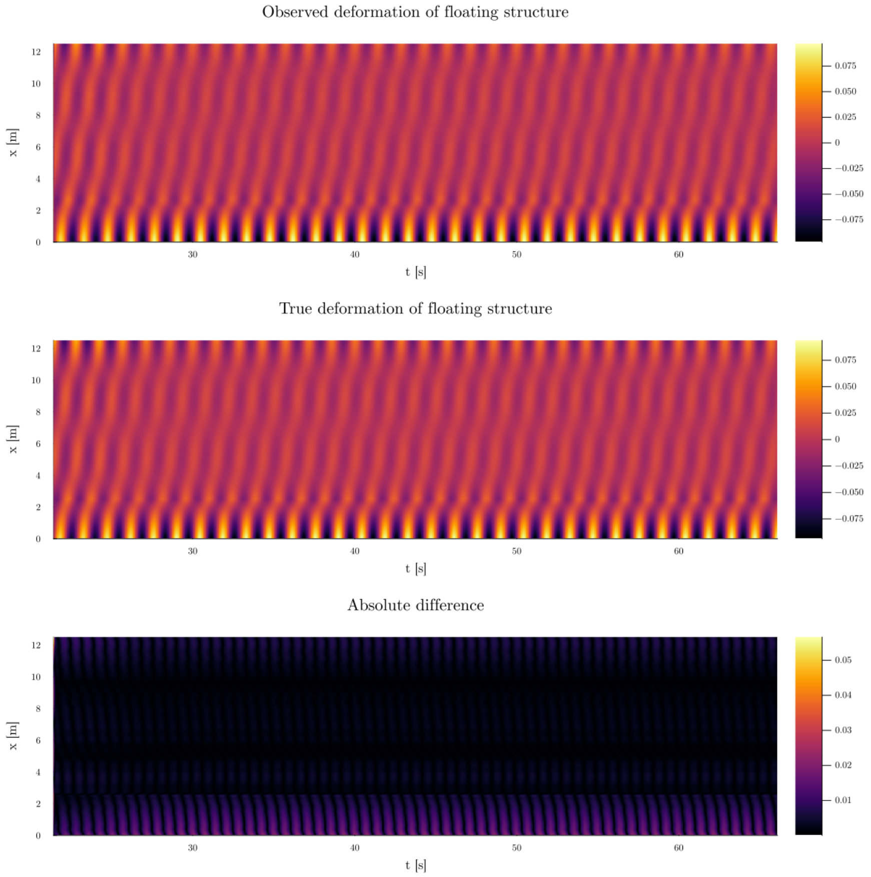


Figure 5.2: Visualization of the ramping function used to smoothly transition control inputs and prevent shocks in the finite element model (FEM), ensuring system stability over the initial phase of control application.

5.2.3. Performance of the Kalman filter

We can assess the performance of the used Kalman filter by the use of a heat map. It is important that our Kalman filter reconstructs the full states of the system from the relatively sparse measurements. We have chosen for 50 probes that are placed evenly along the length of the floating structure. Note that the full state of the system has a length of 1000. From Fig. 5.3 we can see how the reconstructed full state resembles quite accurately the actual full state of the floating structure. The Kalman filter performs well, especially in the region between the joint and downstream end of the structure. The error found in the upstream front part of the structure could be due to the discontinuity introduced by the joint and the relatively small length of the front part, which makes it difficult for the Kalman filter to fit a continuous approximation of the deformation profile to the true discontinuous profile. Nevertheless, the Kalman filter performs well in general.


Observed deformation of floating structure

True deformation of floating structure

Absolute difference

Figure 5.3: Comparison of observed and true deformations of the floating structure using a Kalman filter. The top plot shows the deformation estimated by the Kalman filter, the middle plot shows the true deformation, and the bottom plot presents the absolute difference, highlighting the accuracy of the filter.

5.2.4. FE cost function

To assess the effectiveness of the control framework, we analyse a Finite Element defined cost function. Since we are able to compute the actual deformation of the true system we opt to control from the FEM simulation, it is more convenient to analyse this deformation field of the actual plant. The FE cost function is defined as

$$J_{\mathsf{FE}} = \int \eta_h^2 d\Gamma_{str,2} \tag{5.2}$$

that is we integrate the squared deformation field of the rear part of the floating structure at each time-step of the simulation. Fig.5.4 depicts the evaluation of the FE cost function. In the initial stage of the simulation we find our system to be in the transient part. In this part we excite the structure gradually by introducing the incoming sinusoidal wave as defined in Eq.3.23. Subsequently, our floating structure reaches the stationary phase where the FE cost function follows a sinusoidal-like pattern. We then start the control as can be seen in Fig.5.4 where we ramp up the control gradually using the ramping function. The FE cost function shows this gradual decrease in magnitude due to this ramping. Eventually, we reach the full control potential of our MPC. We can see a quite significant decrease in magnitude of the FE cost function in the uncontrolled and controlled part of the simulation. Furthermore, we see that in our controlled part, there is still some oscillation observed, proving our assumption that our system is not fully controllable but stabilizable because of the energy continuously introduced in the system.

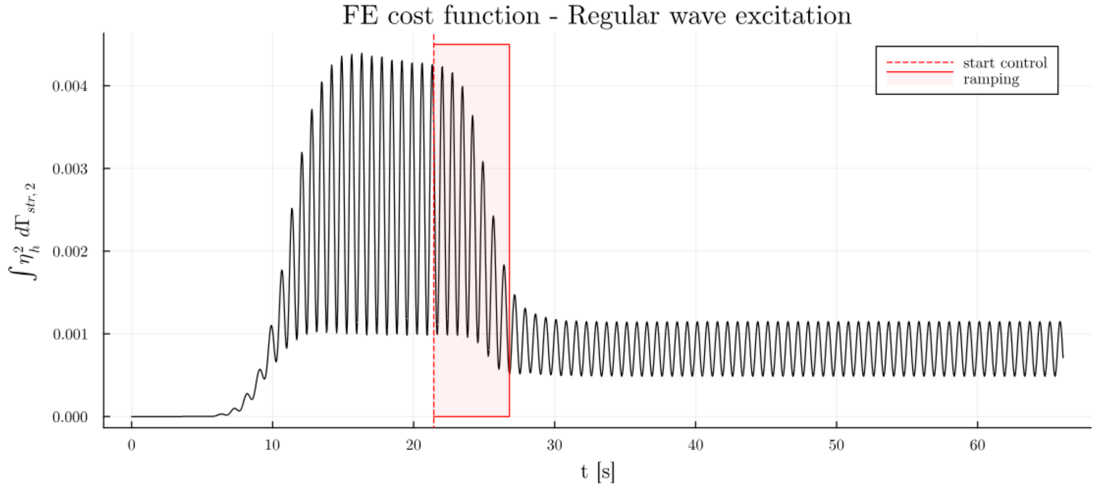


Figure 5.4: Evaluation of the FE cost function under regular wave excitation. The plot illustrates the transient phase, where the structure is excited by an incoming sinusoidal wave, followed by the control phase where the ramping function is applied. The decrease in the FE cost function magnitude during the controlled part of the simulation demonstrates the effectiveness of the control framework, despite the presence of some residual oscillations, indicating that the system is stabilizable even though it is not fully controllable.

As for the control input, which can be seen in Fig. 5.5, we first do not impose any control on the floating structure as explained before. Similar to the cost function, we observe the ramping part when we start the control. We find that for an incoming wave with an amplitude $\kappa_0 = 0.10\, m$ the maximum absolute external moment of the control input signal is $M_{\text{ext}} \approx 320\, Nm$. Eventually, we reach a stationary control input showing a sinusoidal pattern.

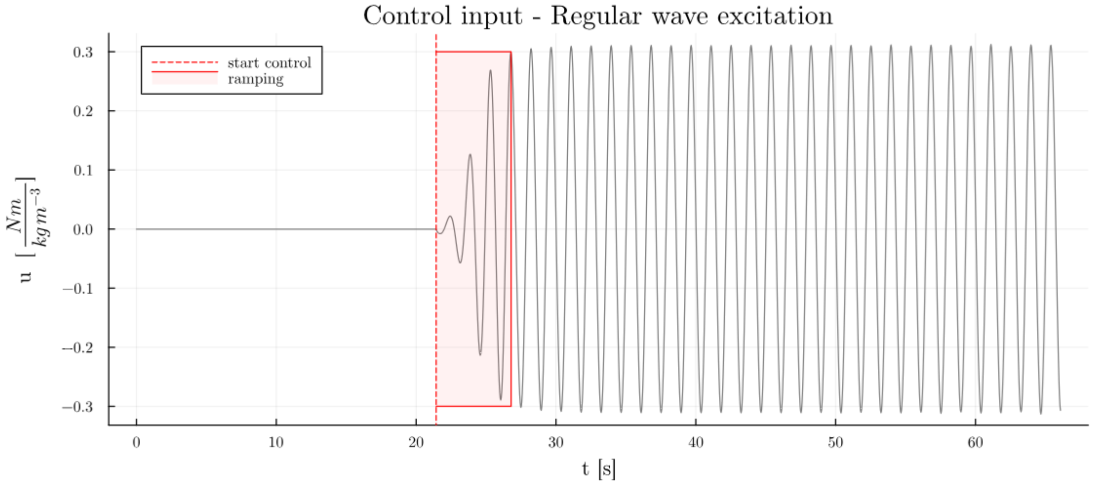


Figure 5.5: Control input under regular wave excitation. The plot shows the control input applied to the system, with the initiation of control marked by the red dashed line and the ramping period highlighted. The control input stabilizes after the ramping phase, demonstrating the effectiveness of the applied control strategy in managing the external wave excitation.

By looking at Fig.5.5 we assume that there is a correlation between the frequency found in the control input signal and the frequency of the incoming wave. We can confirm this by carrying out a Fourier transformation on the stationary part of the control input signal. We indeed find that the frequency of the incoming wave is directly correlated to the most dominant frequency found in the stationary control input signal. From the assumptions of the model we know that there is no air gap between the structure and fluid. Hence, to stabilize the rear part of the structure, it is evident that we should reduce the transmitted wave such that the fluid passing under the rear part is minimal and subsequently the deformation of the rear part. Since the used wave theory assumes conservation of mass we find that the reflected wave will increase in amplitude when controlling. In other words, the control in essence will try to generate an outgoing wave from the structure emigrating in the negative direction from the structure towards the inlet boundary such that the energy in the transmitted wave gets minimized. These statements can be confirmed from Fig.5.7. This result corresponds to the discussed diffraction theory in 4.1.2: the controller does not opt to tame the incoming wave, rather the controller emits the energy from the wet-modes by generating a reflected wave.

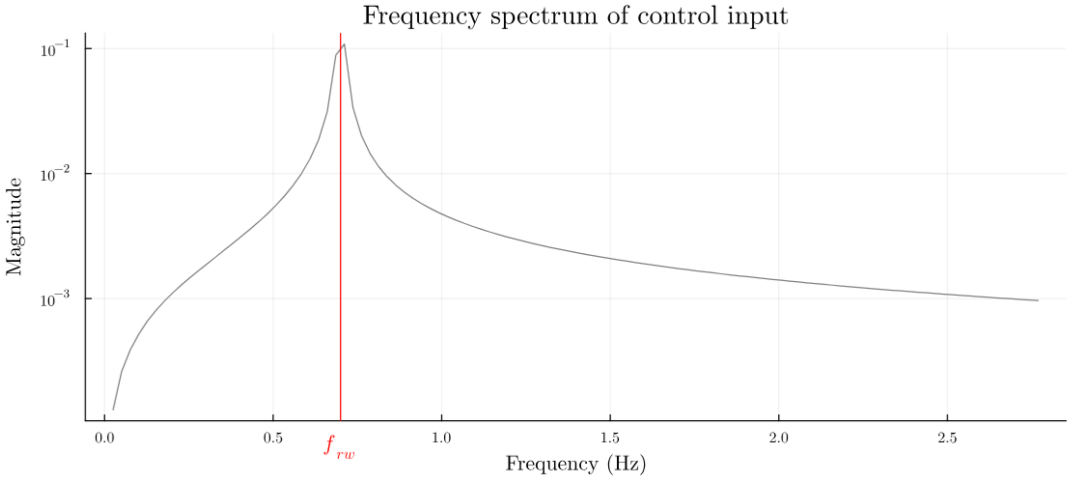


Figure 5.6: Frequency spectrum of the control input. The plot shows the magnitude of the control input across different frequencies, with a significant peak at the excitation frequency f_{rw} , indicating the control system's response to the primary excitation frequency.

From dynamics, we know that if a structure is subjected to an external force that is sinusoidal and consists of a single frequency, it primarily excites the mode of the structure whose natural frequency is close to the excitation frequency. This is evident in Figure C.5, where the time dynamics of the most dominant mode show oscillations at a frequency that matches the excitation frequency exactly. Naturally, the controller aims to dampen this mode using a control input at the same frequency. This finding helps us better understand the relationship between the control input and the wet-modes of the structure.

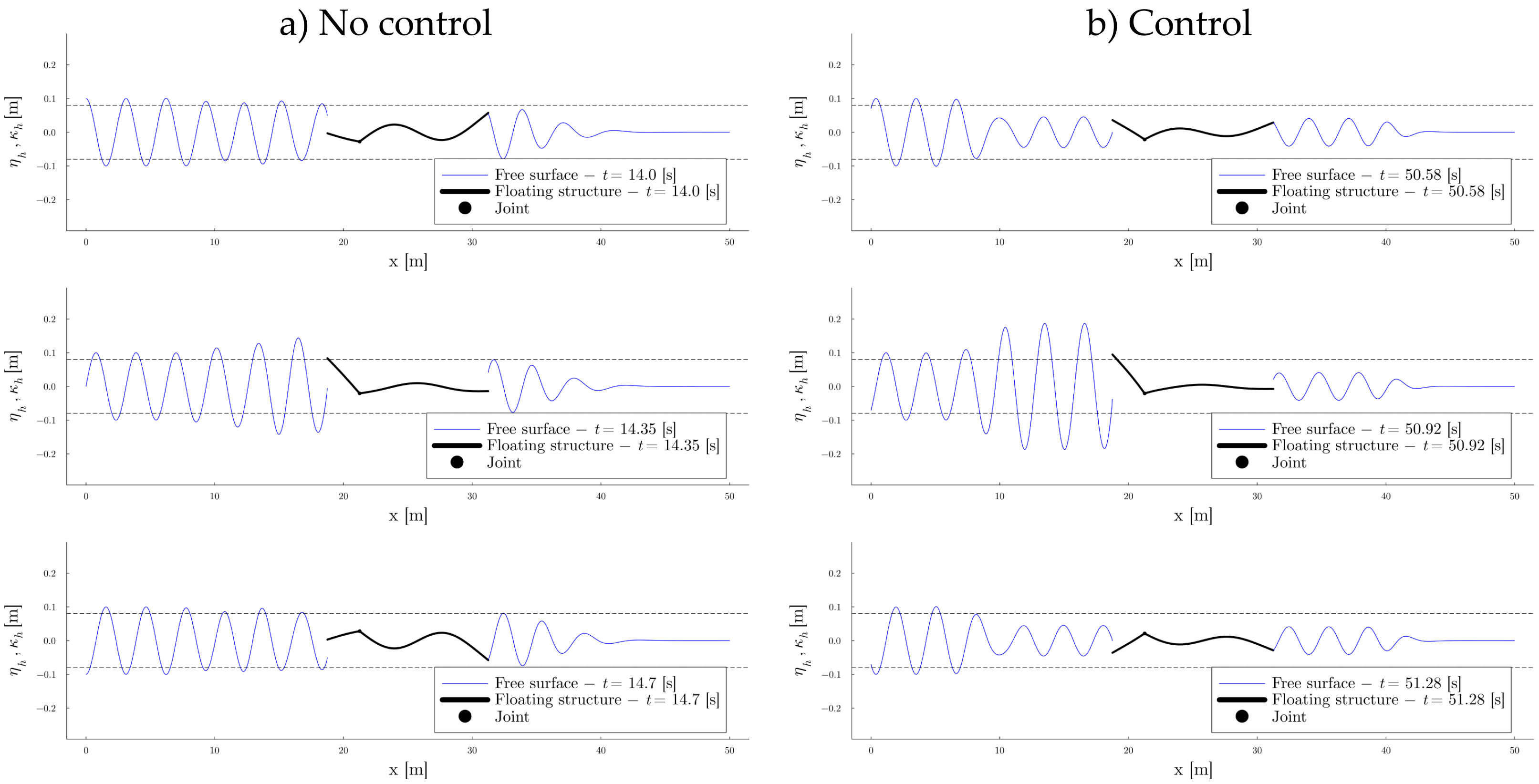


Figure 5.7: Comparison of the structure's deformation without control (a) and with control (b). In the controlled scenario, the control input is tuned to minimize the transmitted wave under the structure, resulting in a decrease in rear deformation by generating a reflected wave that propagates in the opposite direction. The plots demonstrate the effectiveness of the control strategy, where the reduction in transmitted wave energy correlates with the stabilization of the rear part of the floating structure, as evidenced by the diminished deformation in the controlled case.

5.3. Sea state

In this section, we focus on controlling a flexible floating structure under sea state conditions. Unlike the scenario where we dealt with regular wave conditions, this setup differs from the one used to derive the DMDc model implemented in the MPC controller. The key difference is that instead of using a flat-load spectrum to excite the structure, we use an actual sea state, represented by the JONSWAP spectrum. Additionally, we opted for relatively small dimensions for the structure to demonstrate that, even in a worst-case scenario—where the structure has low inertia—we can still achieve effective control. The results are presented in a manner similar to the previously discussed case.

5.3.1. MPC parameters

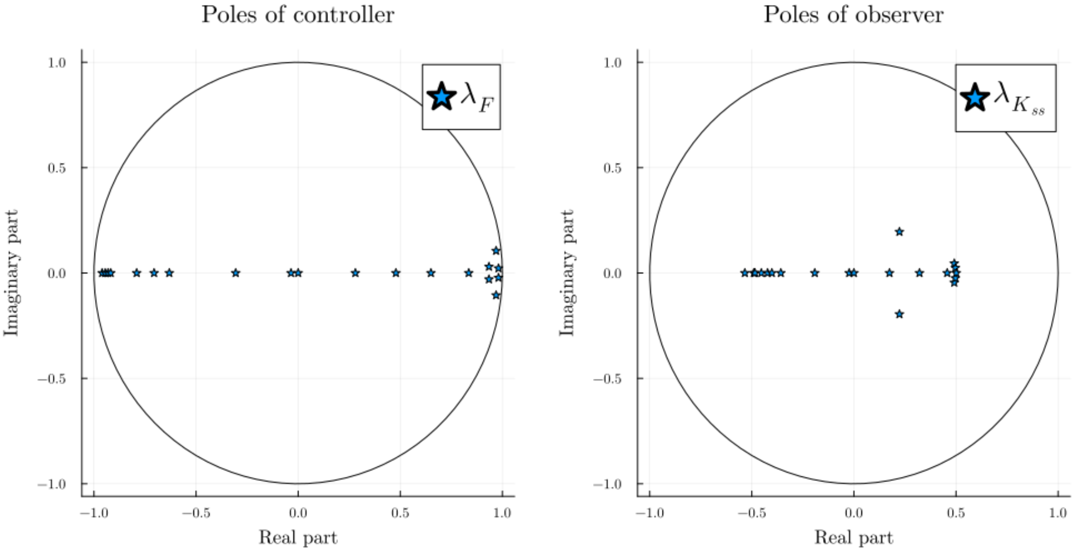


Figure 5.8: Distribution of the poles for the controller and observer with the unit circle (flat load obtained DMDc model).

Finally, we reach the case where we opt to control the floating structure in a sea state. For this case we implement the DMDc model obtained from the flat load spectrum in our MPC controller. We choose a relatively high peak period to be equal to 2.5 seconds and we choose a significant wave height to be equal to 0.1 meters. These are not entirely random chosen parameters, rather we choose the sea state to contain high frequency waves since our structure have natural frequencies which are relatively high because of the small dimensions our structure. Moreover, we choose a significant wave height of 0.1 meters again because of the relatively small dimensions of the floating structure. It is obvious that if the sea state has a high significant wave height it would be more difficult for a relatively small structure to be controlled. In conclusion, we choose an appropriate sea state for this considered floating structure to demonstrate the ability of control. Moreover, we assess the stability of the used model as can be seen in Fig.5.8. Similar to the case where we controlled the floating structure in a regular wave environment, we use the same ramping function for the same reasons.

Table 5.2: Summary of (Control) Parameters for the MPC Controller (Sea-state)

Parameter	**Value**
Process noise covariance matrix, Q	$100I$
Measurement noise covariance matrix, R	$10I$
Physical constraints on actuator	None
Penalization on control increments, ζ	0
Penalization on initial control input, ζ_0	$\sqrt{10}$
Tracking accuracy weight, $\tilde{\mathcal{Q}}$	200
Control horizon length	10 steps
Prediction horizon length	10 steps
Stability assessment	Fig.5.8
Peak period, T_p	2.5 s
Significant wave height, H_s	0.1 m

5.3.2. Performance of the Kalman filter

For this considered case we use the exact same Kalman filter as in the case of the regular wave. We again assess the error between the observed deformation and the true deformation using the heat map found in Fig.5.9. From the heat map we see that the Kalman filter manages to observe quite accurately the true deformation of the structure from the sparse measurements.

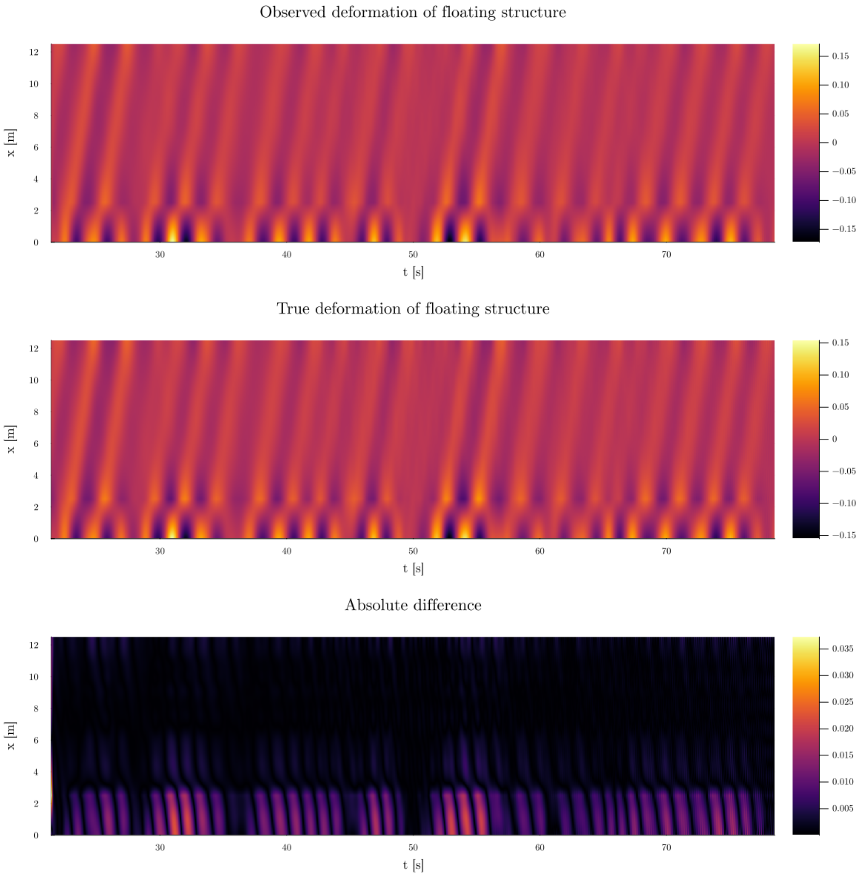


Figure 5.9: Comparison of observed and true deformations of the floating structure using a Kalman filter in a sea state environment. The top plot shows the deformation estimated by the Kalman filter, the middle plot shows the true deformation, and the bottom plot presents the absolute difference, highlighting the accuracy of the filter.

5.3.3. FE cost function

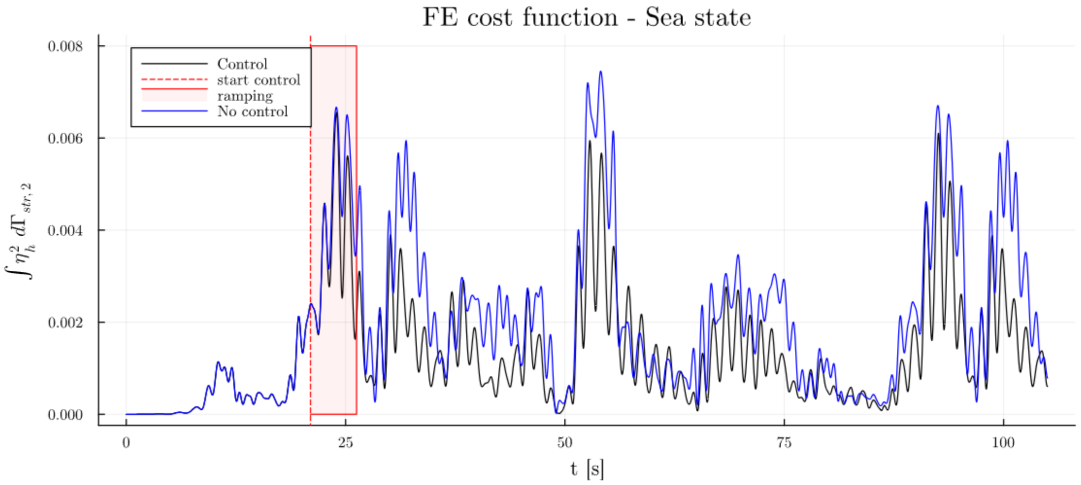


Figure 5.10: Comparison of the FE-defined objective functions of the rear part of the structure between the controlled and uncontrolled cases. This figure illustrates the differences in deformation when control is applied to manage wave interaction.

We use the same Finite Element defined cost function as described in Eq.5.2 to assess the effectiveness of the control framework in a sea state environment. However, in this case we plot the controlled FE defined cost function against the uncontrolled FE defined cost function. In the uncontrolled case we use the exact same setup, except that we do not control the floating. Hence, we can closely examine the effect of control on the cost function in comparison with the uncontrolled case.

In Fig.5.10, both cases initially start without control, which is why the two cost functions overlap during this period. Once the control is activated, we can clearly see that the controlled FE cost function reaches lower values compared to the uncontrolled FE cost function. However, there are instances where the control does not yield satisfactory results. This is because controlling the rear part of the structure requires the front part to generate a flapping motion to reflect the incoming waves. Since the rear part is connected to the front part, the initial section of the rear part is inevitably influenced by the movement of the front part.

To prove that this is indeed the case, let us present two new figures. Let us recall the cost function defined in 4.2. In the first figure, we plot the summed deformation of the rear part of the controlled and uncontrolled case against each other. In the second figure we plot the same but now we shift the beginning of the rear part further away from the joint. Hence, for the first figure we showcase the following quantity

$$J_{\text{rear},1} = \sum_{i=1}^{n_{\text{rear}}} p_{\text{rear,i}}^2 \tag{5.3}$$

and for the second figure we showcase the following quantity

$$J_{\text{rear},2} = \sum_{i=200}^{n_{\text{rear}}} p_{\text{rear,i}}^2. \tag{5.4}$$

Note that the rear part of the floating structure has a total of 800 DoFs. Thus we shift our analysis by neglecting the first 200 DoFs since these are highly influenced by the flapping motion of the front part.

Note how Fig.5.11 has the exact same shape as the FE cost function, except that the values are different because in the FE defined cost function we integrate the deformation whereas in Fig.5.11 we sum the deformations of the DoFs.

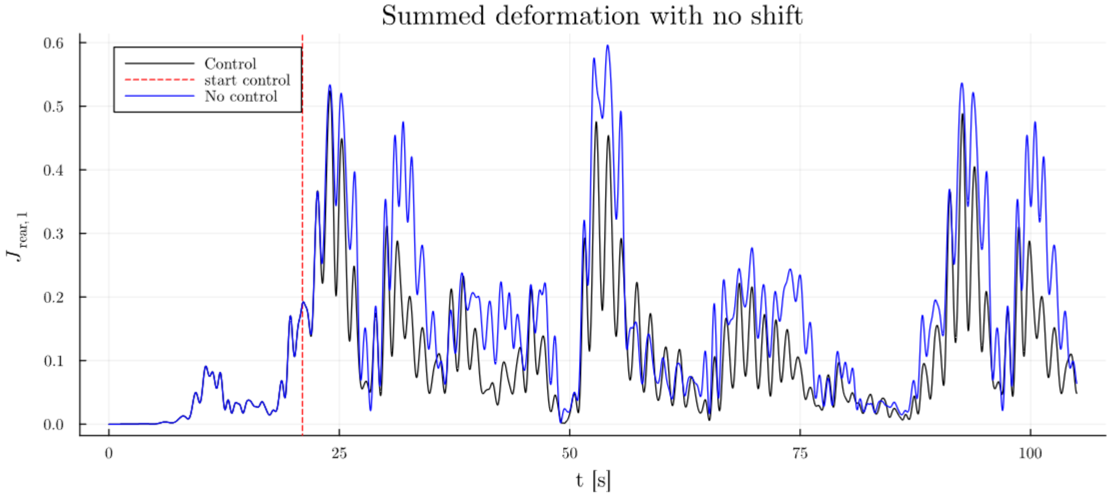


Figure 5.11: Comparison of the summed deformation of the rear part of the structure between the controlled and uncontrolled cases. This figure illustrates the differences in deformation when control is applied to manage wave interaction.

The observation we stated earlier, that is the rear part is greatly influenced by the motion of the front part, can be confirmed from the Fig.5.11 and Fig.5.12. From Fig.5.12 we can clearly see how the control successfully reduced the motion of the rear part against the uncontrolled case. We can quantify this reduction by studying the ratio of the objective functions between the controlled and uncontrolled scenario. This ratio is depicted in Fig. 5.11 where we can clearly see significant reduction in the objective function. Hence, this shows how important the dimensions of the structure are to control the structure. For next cases we should take the length of the rear part into account. However, this requires an entire new DMDc model and is therefore out of the scope of this study.

Finally, we perform a frequency spectrum analysis on the control input. To carry out this analysis, we must compare the natural frequencies of the floating structure and the JONSWAP spectrum with the frequency spectrum of the control input.

In Fig. 5.13, it is evident that our control input is primarily dominated by lower frequencies. Additionally, we observe that the JONSWAP spectrum for the specified sea state aligns with the control spectrum. This aligns with our findings from the previous scenario, where we analyzed a regular wave condition, and found that the control spectrum peaked at the excitation wave's frequency. This is expected, as the modes of the floating structure with the highest energy are those with greater inertia, resulting in lower natural frequencies. Consequently, the control strategy focuses on these low-frequency modes.

Concerning the structure's natural frequencies, we see that the second, third and fourth natural frequencies also exhibit peaks with the control spectrum. Given that these natural frequencies hold the most energy and their associated modes are excited since their natural frequencies fall within the energy-dense part of the sea state's spectrum, we observe that controlling these modes result in effective stabilization of the structure.

In summary, the control mechanism works by stabilizing the modes whose natural frequencies align with the energy-dense frequencies of the sea state. If the natural frequencies of the floating structure's most dominant modes do not fall within this energy-rich portion of the sea state spectrum, the controller cannot effectively leverage these modes to resist the incoming wave.

Observe in Fig.5.14 that the reflected wave has a larger envelope in the controlled case compared to the uncontrolled case. As anticipated, and in accordance with diffraction theory, the

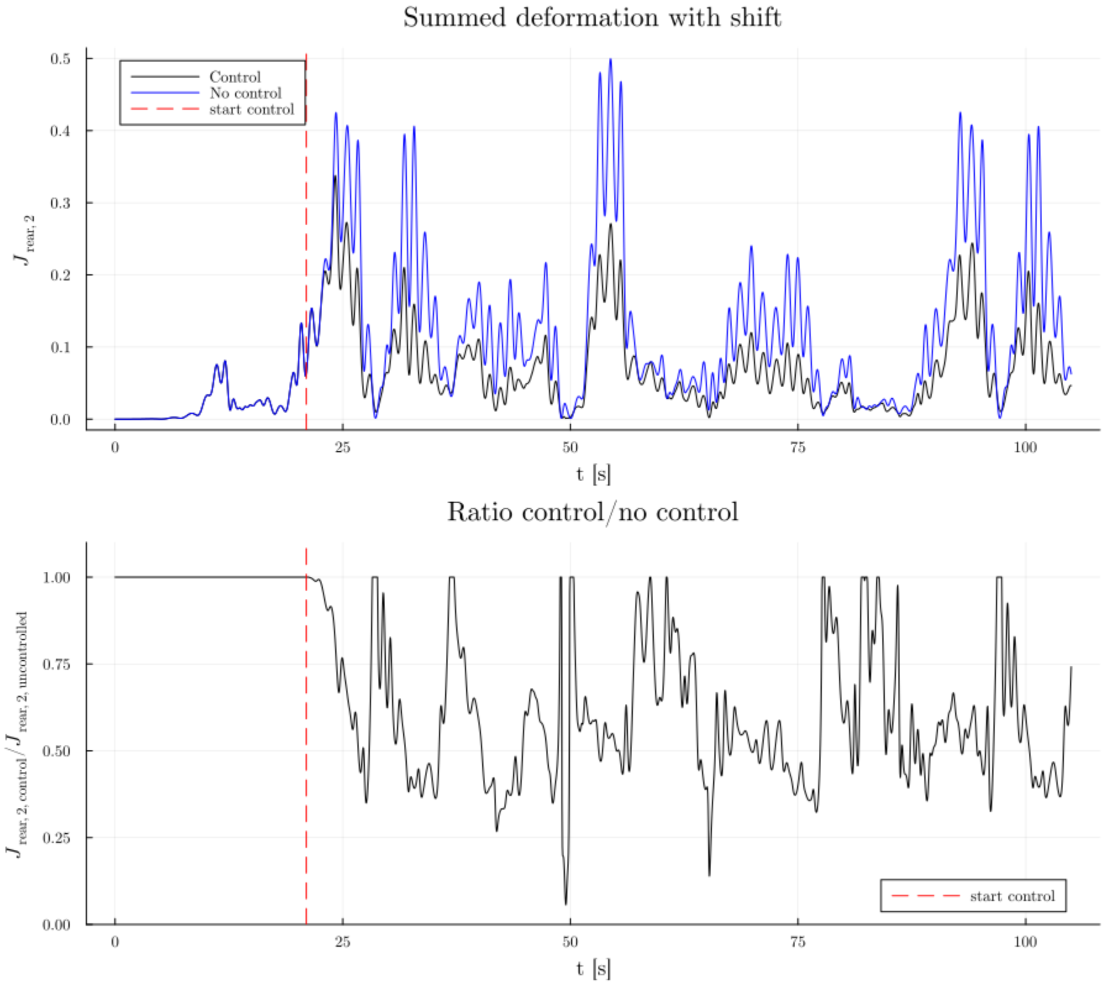


Figure 5.12: Comparison of the summed deformation of the rear part of the structure, with the beginning of the rear part shifted further away from the joint. This figure demonstrates how adjusting the position of the rear part affects the deformation in both controlled and uncontrolled scenarios. The ratio of the objective functions in the controlled and uncontrolled scenario shows how the objective function is reduced at certain points during control.

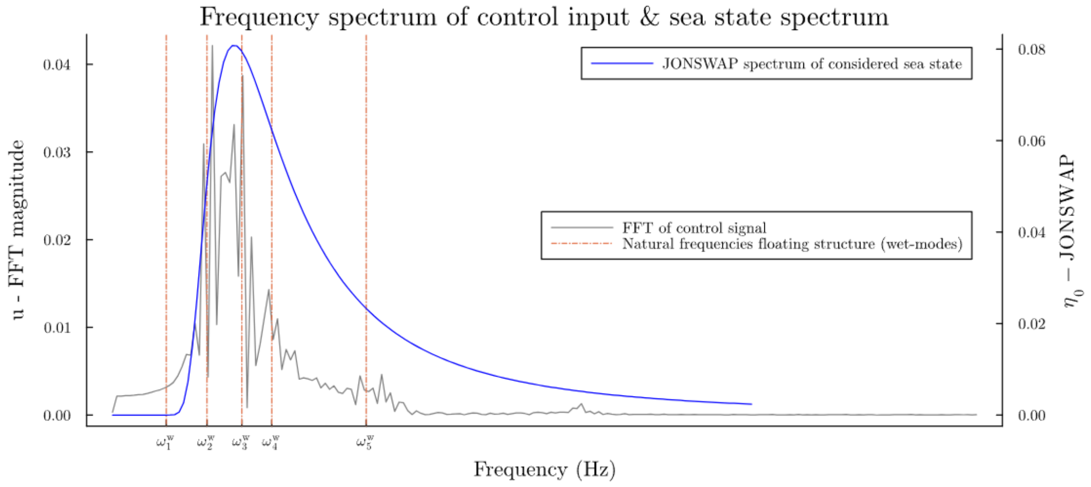


Figure 5.13: Frequency spectrum of the control input under sea state conditions, illustrating the distribution of control efforts across different frequencies. Peaks in the spectrum indicate the dominant frequencies at which the control system is actively managing the structure's response to sea state disturbances.

transmitted wave exhibits a smaller envelope in the controlled case than in the uncontrolled case. In the discussed results from the regular wave condition we observed the same phenomenon, which seems to play an important role in the control of floating structures.

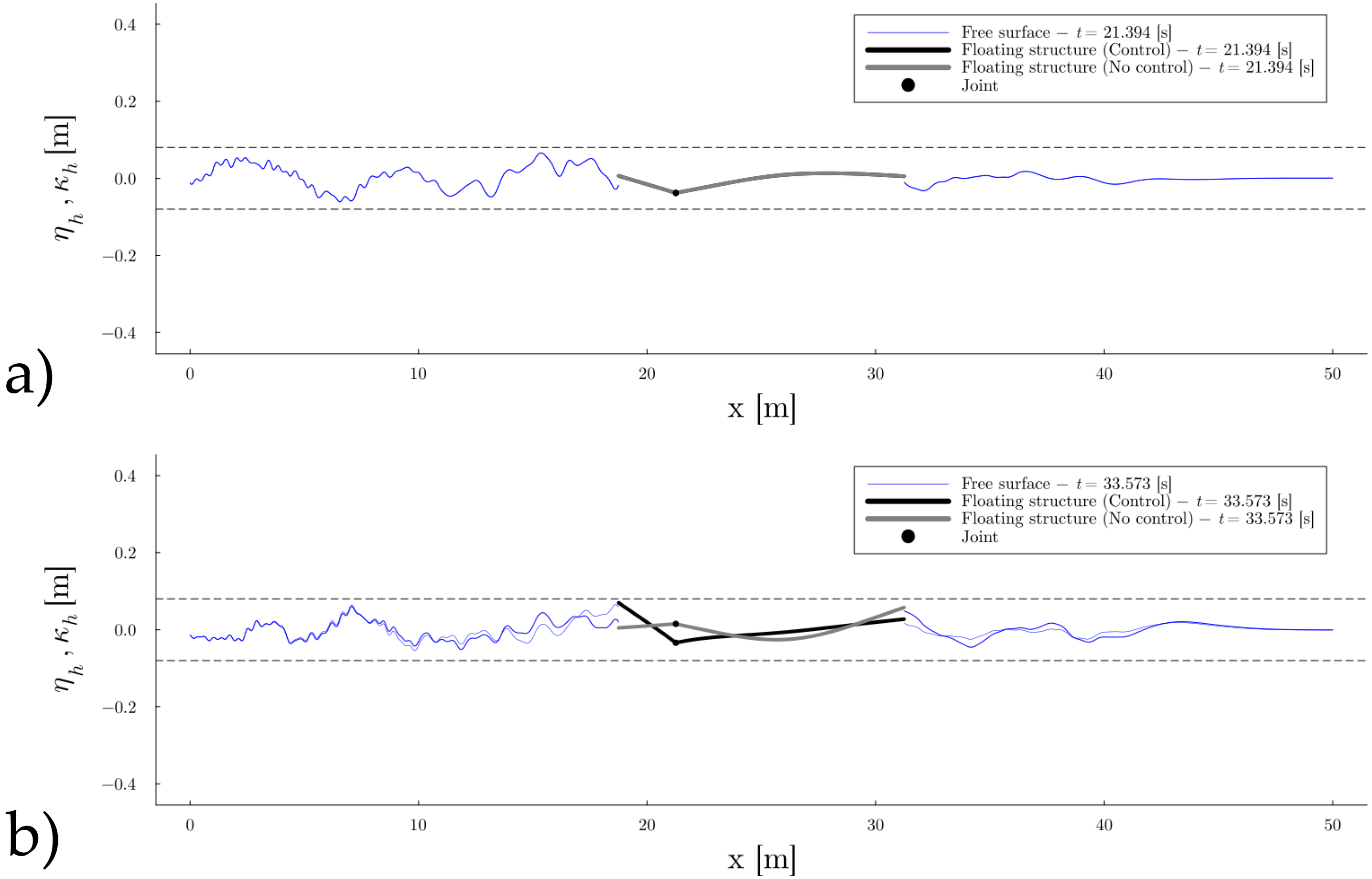


Figure 5.14: Comparison of the structure's deformation without control (a) and with control (b). Note how in the controlled scenario the reflected wave is bigger than in the uncontrolled case. Similarly, the transmitted wave in the controlled case is smaller than in the uncontrolled case.

5.4. Interpretation of the results

In both considered cases for control we observed that the controller in essence tries to generate a reflected wave by making a flapping move using the front part of the floating structure. In order to interpret the motivation behind this strategy we should investigate the forces acting on the floating structure which the controller takes into account. Hence, as discussed before, we assume that the controller has no knowledge of the exciting forces due to the incoming wave.

5.4.1. Hydromechanical loads

For simplicity, we consider our floating structure as a mass-spring system. Moreover, since the system under consideration is linear, we know that our structure is influenced by two types of loads, which we can superimpose [19].

1. **Incoming wave-induced forces** are the forces acting on the restrained structure due to the incoming offshore waves.
2. **Hydromechanical forces** are the forces produced by the oscillations of the floating body in still water conditions.

Let us recall the undamped mass-spring system, which we used to find the wet-modes of the structure:

$$(-\omega^2(\mathbf{M}_{11} + \mathbf{A}) + \mathbf{K}_{11}^d + \mathbf{K}_{11}^w)\boldsymbol{\eta}_P = 0 \tag{5.5}$$

We know from the wet-mode analysis that the wet-modes are independent of any external forces. Therefore, in Eq.5.5 we do not consider any external forces such as the incoming wave-induced forces. This equation is, in essence, Newton's second law in the frequency domain. It is important to note that our system does not contain any damping; that is, energy in the structure does not dissipate when the structure is excited. The hydromechanical forces can be found by moving the spring-term in Eq.5.5 to the right-hand side, which gives us:

$$\omega^2(\mathbf{M}_{11} + \mathbf{A})\boldsymbol{\eta}_P = (\mathbf{K}_{11}^d + \mathbf{K}_{11}^w)\boldsymbol{\eta}_P \tag{5.6}$$

Here, the added mass $\mathbf{A}$ depends on ω. Assuming we have carried out the wet-mode analysis and found the natural frequencies of the wet-modes, we can establish the relationship between the mass of the structure and the hydromechanical force. Note that a structure with large inertia—meaning that the matrices $\mathbf{M}_{11}$ and $\mathbf{A}$ have significant contributions to the system—leads to a large hydromechanical force. The water particles near the structure are thus heavily influenced by the inertia of the structure, leading to a standing wave, which is the reflected component of the total velocity potential [19]. As a result, we observe a reflected wave generated by the structure. One way to interpret this is by considering the floating structure as a separate wave generator, emitting waves away from the structure.

In the case of no control, these reflected waves due to the hydromechanical force do not contribute to any dissipation of energy. In the current formulation of Eq.5.5, we consider no (added) damping. However, we know that if damping is present, the floating structure would generate waves that propagate away from the structure [19]. Note that in these reflected waves, energy is dissipated from the structure. If we recognize that our controller aims to stabilize the rear part of the floating structure, this essentially involves dissipating energy from this rear part. Therefore, our controller functions as a "damper". The consequence of is therefore the generation of these reflected waves that dissipate energy from the rear part.

It is essential to note that this does not imply that the floating structure can be controlled equally well across all types of sea states, as the wet modes are not dependent on the excitation force. Instead, due to Fluid-Structure Interaction, the energy-dense frequencies of the sea state spectrum will primarily excite those modes of the floating structure whose natural frequencies are close to these energy-dense frequencies. Therefore, the controller will be primarily effective when the dominant natural frequencies of the floating structure are aligned with the energy-dense frequencies of the sea state. This will allow the controller to leverage the reflected wave as a resisting force to withstand the incoming wave. If the low-energy modes are excited due to a sea state which its energy-dense frequencies do not align with the most dominant natural frequencies of the structure, the controller will most likely still try to leverage these low-energy modes but not as effective as for the dominant modes.

This is achieved by amplifying the reflected wave due to the spring-term through the flapping movement of the front part. Because of the conservation of mass, this directly results in a smaller transmitted wave under the structure and, consequently, a lower input value of energy introduced into the structure. Moreover, to amplify this reflected wave, the controller should apply the control input with a frequency that lies close or is exactly the same as the natural frequency related to that reflected wave. Therefore, in Fig.5.6 we observed how the control input has the exact same frequency as the frequency of the incoming wave. Since the structure is excited by one single frequency, we know that the most dominant mode acting in that case has a natural frequency which lies close to the excitation frequency.

Note that this analysis is not limited to flexible floating structures. This analysis is in general applicable for any offshore structure. Note that structures with larger inertia will lead to a larger hydromechanic force which in turn will allow the controller to amplify this hydromechanic force even more to counteract the incoming wave-induced force. This means that even the low-energy modes could be leveraged to resist the incoming wave. However, this will require also larger machinery to be able to cope with the dimensions of the system.

An important aspect that should not be left out consideration is the correct implementation of the wet-modes in the model of the MPC. In this study we showed how POD can be used to incorporate the wet-modes of the structure into the DMDc model by comparing the similarities between the POD modes and the actual wet-modes. However, this approach relies heavily on data and may not be scientifically accurate. Nevertheless, if the POD modes and the wet-modes show relatively large similarities, the suggested approach seems to return valuable results and insights.

Key takeaway

- The controller amplifies the structure by generating reflected waves through a flapping motion, redirecting energy away from the structure without relying on wave force data.
- Control performance is highest when the structure's natural frequencies align with the energy-dense frequencies of the sea state, enhancing energy dissipation.
- Hydromechanical forces from the structure's inertia generate reflected waves, dissipating energy when matched with the controller's input frequency. This amplifies the reflected wave and hence dissipating more energy away from the structure.

6 Conclusion & Recommendations

6.1. Conclusion

In this study, we aimed to develop a framework and evaluate a control strategy for flexible floating structures by integrating Model Predictive Control with a Finite Element Model. Our goal was to optimize the performance and stability of flexible floating offshore structures for offshore engineering applications. To achieve this, we began with a literature review to explore existing studies and methodologies. The review revealed a notable gap: no existing studies had implemented a control strategy for flexible floating structures with fully coupled fluid-structure interaction, leaving the fundamental FSI component unaddressed.

A primary challenge in this context is the high dimensionality involved in modeling flexible floating structures. Analytically, the deformation field theoretically includes an infinite number of degrees of freedom that we aim to control. Numerically, this challenge can be approached by discretization, resulting in a high but finite number of degrees of freedom. Furthermore, control of these systems often relies on sparse measurements of the structure's deformation, necessitating a state observer capable of reconstructing the structure's state from limited, noisy data.

After a thorough investigation, we decided to implement Model Predictive Control, a state-of-the-art control strategy, in combination with a Reduced Order Model to mitigate the risks of infeasible control due to hardware limitations. These limitations make it challenging to directly implement a high-fidelity model within the MPC framework. MPC was selected because it enables the incorporation of arbitrary objective functions alongside constraints, which are commonly required in real-world applications.

Among ROM techniques, Dynamic Mode Decomposition with Control emerged as the most interpretable and data-driven method for generating a ROM suitable for MPC. Unlike neural networks, DMDc produces a simple linear system that includes dynamic modes derived from the data, allowing direct integration within conventional MPC theory.

We began by designing a general framework, integrating MPC with the Finite Element model of the floating structure, which serves as the plant to be controlled. This framework comprises two main components: the MPC and the true system. The true system is represented by the FE model of the floating structure, with added noise to simulate realistic conditions. Sparse, noisy measurements from the true system are then provided to a state observer within the MPC. The purpose of this state observer is to reconstruct the true state of the system as accurately as possible by estimating it based on limited measurements.

For the state observer, we used an adaptive Kalman filter, chosen for its robustness and adaptability through the incorporation of prior knowledge. Finally, the estimated state of the system is fed into an optimization block, which solves an optimization problem to generate a control input that aligns with the control objectives. The results demonstrated the effectiveness of this framework in controlling flexible floating structures, even under challenging sea state conditions.

We based the formulation of the floating structure on the approach presented in [33] and extended it by incorporating an actuator within the joint to apply an external moment. The FE model of the floating structure served two primary purposes: first, to generate open-loop controlled data used to develop the ROM and second, to provide a high-fidelity representation of an actual floating structure for control purposes.

For the first purpose, we found that the choice of the open-loop control signal significantly impacts the quality of the ROM. Therefore, it is crucial to design an open-loop control signal that captures the most essential dynamics of the system. For optimal performance, we recommend the following iterative approach: begin with an open-loop control signal informed by theoretical knowledge and intuition, then apply it in an actual case. If the control results are initially unsatisfactory but show partial success at certain points in the simulation, use this data to refine the ROM. This iterative process can continue until the model produces satisfactory results. Additionally, the quality of the ROM as a function of data length can be assessed through a convergence study.

In this study, we designed various open-loop control signals. However, due to time constraints, we were unable to refine the ROM using the iterative approach described earlier. Ideally, the open-loop control signal should be as versatile as possible to capture a broad range of dynamics. Furthermore, the quality of the ROM could be further evaluated by validating it against additional data. For effective ROM design to control the floating structure in sea states, we found that including wet-modes in the data is crucial. According to diffraction theory, incoming waves excite these wet-modes, which have natural frequencies that align with the energy-dense frequencies typical of sea states. We determined that exciting the floating structure using a flat-load spectrum is the most effective approach for capturing these wet-modes in the ROM through DMDc. The presence of wet modes in the data was confirmed using Proper Orthogonal Decomposition (POD), as outlined in [39].

The results demonstrate the effectiveness of the proposed control framework in stabilizing flexible floating structures under both regular wave and sea state conditions. By implementing an MPC strategy that prioritizes control stability and robustness, the framework successfully minimizes structural deformation, as evidenced by the reduced FE cost function values in controlled cases. The Kalman filter also performed well, accurately reconstructing the system's full state from sparse measurements, confirming its robustness in noisy environments.

The controller's approach to stabilization relies on generating reflected waves through a flapping motion, redirecting energy away from the structure without relying on precise wave force data. This approach is particularly effective when the natural frequencies of the structure's modes align with the energy-dense frequencies of the sea state, allowing for enhanced energy dissipation. The frequency spectrum analysis further supports this, showing that the control input's frequency aligns with dominant excitation frequencies, enabling targeted damping of the most active structural modes.

In sea state conditions, the framework continues to perform effectively, even for a structure with low inertia, proving its versatility and adaptability to different wave environments. The analysis highlights the critical role of hydromechanical forces generated by the structure's inertia, which the control strategy amplifies to enhance energy dissipation. Additionally, the integration of wet modes into the DMDc model via POD has shown promise, though further validation of this approach would strengthen its applicability.

In summary, this study provides a proof of concept for a versatile control strategy applicable to flexible floating structures, with potential for broader application to various offshore structures. Future work could further optimize the approach by testing it with larger structures and varied sea states, potentially expanding the method's effectiveness in harsh offshore conditions.

6.2. Recommendations

Nevertheless, there is plenty of room for improvement of this method. As described in 4.1.2 we expect that larger offshore structures will be easier to control in harsh offshore conditions. The theory discussed in 4.1.2 showcases how inertia of the floating structure can be of great influence on the control strategy, especially when using only the structure deformation as the observed model. Hence, in the context of this study, one should do a parameter analysis to prove the relation between the dimensions of the flexible floating and the control strategy. Moreover, it is essential to test for multiple cases with different sea states to prove the effectiveness of the proposed control strategy in different scenarios.

Furthermore, we motivate other researchers in the field of offshore structure to find simpler representation of the Finite Element Model that are physically more sound than the DMDc. The DMDc has proved itself in this study to be a reliable method to control offshore floating structures, but due to its high dependency on the quality of data it is difficult to prove that the method is sound regarding the physical interpretation of the DMDc modes. We therefore suggest a combination of Neural Networks and DMDc to fine-tune the system matrices obtained by the DMDc. This will make it possible to generalize the obtained DMDc model for different conditions. Moreover, in this study we used only one control input to control the structure. However, the proposed method allows Multiple Inputs, Multiple Outputs (MIMO), which could be implemented for more advanced cases, such as the control using multiple mooring lines to reach more than one objective.

Finally, we should underscore the importance of the floating structure's properties, such as its flexibility, length, and mass. The results indicate that the structure's natural frequencies play a critical role in effective control. Deploying an offshore flexible floating structure in a sea state that does not align with the structure's dominant natural frequencies can make control less effective. Based on the findings, it is recommended to test control strategies on larger inertia structures, as their increased hydromechanical forces, resulting from added mass, can enhance controllability. This, as we concluded earlier, makes it possible to leverage the reflected wave as a resisting wave to the incoming wave. In comparison to low-inertia structures, these larger systems are likely to respond more effectively to control inputs, even at low-energy modes. Testing with higher inertia structures could provide valuable insights into optimizing control performance under varied sea states and may further validate the effectiveness of the proposed control framework for large-scale offshore applications.

In conclusion, although the proposed method shows promising results, there is still room to proof the effectiveness of the control strategy using hard evidence and more simulations using different dimensions of the structure and environmental conditions.

References

[1] June 2021. url: https://www.youtube.com/watch?v=LioOvUZ1MiM.

[2] Dec. 2022. url: https://sea-man.org/mathematical-model.html.

[3] Shagun Agarwal, Oriol Colomés, and Andrei V. Metrikine. “Dynamic analysis of viscoelastic floating membranes using monolithic Finite Element method”. In: *Journal of Fluids and Structures* 129 (2024), p. 104167. issn: 0889-9746. doi: https://doi.org/10.1016/j.jfluidstructs.2024.104167. url: https://www.sciencedirect.com/science/article/pii/S0889974624001026.

[4] Nicholas Assimakis and Maria Adam. “Iterative and Algebraic Algorithms for the Computation of the Steady State Kalman Filter Gain”. In: *International Scholarly Research Notices* 2014.1 (2014), p. 417623. doi: https://doi.org/10.1155/2014/417623. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2014/417623. url: https://onlinelibrary.wiley.com/doi/abs/10.1155/2014/417623.

[5] Blake C. Boren. “Distributed Embedded Energy Converters for Ocean Wave Energy Harvesting: Enabling a Domain of Transformative Technologies”. In: (2021).

[6] Ton J.J. van den Boom. *Model Predictive Control*.

[7] Yuzhen Chen et al. “A Novel Fast Model Predictive Control for Large-Scale Structures”. In: *Journal of Vibration and Control* 23 (Oct. 2015). doi: 10.1177/1077546315610033. url: https://www.researchgate.net/publication/282750967_A_Novel_Fast_Model_Predictive_Control_for_Large-Scale_Structures.

[8] Oriol Colomes, Shagun Agarwal, and Frank Lange. *Hydrodynamic loads week 3 - Diffraction theory*. Aug. 2000.

[9] Dengshuo Chen et al. “A coupled frequency and time domain approach for hydroelastic analysis of very large floating structures under focused wave groups”. In: *Ocean Engineering* 255 (July 2022), pp. 111393–111393. doi: 10.1016/j.oceaneng.2022.111393.

[10] Dirk Wolfram and Meurer Thomas. “DMD-Based Model Predictive Control for a Coupled PDE-ODE System”. In: *IFAC-PapersOnLine* (). url: https://www.sciencedirect.com/science/article/pii/S2405896323021985.

[11] Rickey Dubay et al. “Finite element based model predictive control for active vibration suppression of a one-link flexible manipulator”. In: *ISA Transactions* 53.5 (2014), pp. 1609–1619. issn: 0019-0578. doi: https://doi.org/10.1016/j.isatra.2014.05.023. url: https://www.sciencedirect.com/science/article/pii/S0019057814001128.

[12] Kareem El Sayed, Shagun Agarwal, and Oriol Colomès. “Structural optimization of viscoelastic floating membranes through adjoint-based PDE-constrained optimization”. In: ().

[13] G. Engel et al. “Continuous/discontinuous finite element approximations of fourth-order elliptic problems in structural and continuum mechanics with applications to thin beams and plates, and strain gradient elasticity”. en. In: *Computer Methods in Applied Mechanics and Engineering* 191.34 (July 2002), pp. 3669–3750. issn: 00457825. doi: 10.1016/S0045-7825(02)00286-4. url: https://linkinghub.elsevier.com/retrieve/pii/S0045782502002864 (visited on 03/13/2024).

[14] Hanaa M. Farghally and Emad A. Sweelem. “Feasibility Study and Design of a Stand-alone Floating Photovoltaic Structure for Toshka Lake”. In: *Engineering World* 5 (Nov. 2023), pp. 151–162. doi: 10.37394/232025.2023.5.17.

[15] Jacopo Agnelli et al. “Feasibility study of floating photovoltaic panels in the Adriatic Sea”. In: *2023 IEEE International Conference on Environment and Electrical Engineering and 2023 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)* (June 2023). doi: `10.1109/eeeic/icpseurope57605.2023.10194722`.

[16] Johan Löfberg and Johan Löfberg. “Linear Model Predictive Control Stability and Robustness”. In: (Jan. 2001).

[17] Jon S. Dæhlen and Jon Dæhlen. “Nonlinear Model Predictive Control using Derivative-Free Optimization”. In: (Jan. 2013).

[18] Joshua L. Proctor et al. “Dynamic Mode Decomposition with Control”. In: *Siam Journal on Applied Dynamical Systems* 15.1 (Jan. 2016), pp. 142–161. doi: `10.1137/15m1013857`.

[19] J.M.J. Journée et al. *Offshore Hydromechanics*. TU Delft, 2000. url: `https://books.google.nl/books?id=eN66jwEACAAJ`.

[20] Junzheng Wang and Xi Wang. “Ant Colony Optimization based Array Design of Wave Energy Converters”. In: *2023 5th Asia Energy and Electrical Engineering Symposium (AEEES)* (Mar. 2023). doi: `10.1109/aeees56888.2023.10114112`.

[21] Eurika Kaiser et al. “Sparse identification of nonlinear dynamics for model predictive control in the low-data limit”. In: *Proceedings of The Royal Society A: Mathematical, Physical and Engineering Sciences* (Nov. 2017). doi: `10.1098/rspa.2018.0335`.

[22] Kathleen Champion et al. “Data-driven discovery of coordinates and governing equations”. In: *Proceedings of the National Academy of Sciences of the United States of America* 116.45 (Nov. 2019), pp. 22445–22451. doi: `10.1073/pnas.1906995116`.

[23] J. Nathan Kutz et al. *Dynamic Mode Decomposition*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2016. doi: `10.1137/1.9781611974508`. eprint: `https://epubs.siam.org/doi/pdf/10.1137/1.9781611974508`. url: `https://epubs.siam.org/doi/abs/10.1137/1.9781611974508`.

[24] Tony Lacey. “Tutorial: the Kalman Filter 11.1 Introduction 11.2 Mean Squared Error”. In: 1998. url: `https://api.semanticscholar.org/CorpusID:48574040`.

[25] Miles Lubin et al. “JuMP 1.0: Recent improvements to a modeling language for mathematical optimization”. In: *Mathematical Programming Computation* 15 (2023), pp. 581–589. doi: `10.1007/s12532-023-00239-3`.

[26] M. Arnold et al. “Multi-area predictive control for combined electricity and natural gas systems”. In: *European Control Conference* (Aug. 2009), pp. 1408–1413. doi: `10.23919/ecc.2009.7074603`.

[27] Marc Artola et al. “Modal-based Model Predictive Control of Multibody Very Flexible Structures”. In: *IFAC-PapersOnLine* 53.2 (Jan. 2020), pp. 7472–7478. doi: `10.1016/j.ifacol.2020.12.1302`.

[28] *Mark Cannon*. July 2024. url: `https://markcannon.github.io/assets/downloads/teaching/C21_Model_Predictive_Control/C21_MPC_Lecture_Notes.pdf`.

[29] Max Schwenzer et al. “Review on model predictive control: an engineering perspective”. In: *The International Journal of Advanced Manufacturing Technology* 117 (Aug. 2021), pp. 1327–1349. doi: `10.1007/s00170-021-07682-3`.

[30] Min Woo Kim et al. “Numerical analysis of various artificial damping schemes in a three-dimensional numerical wave tank”. In: *Ocean Engineering* 75 (Jan. 2014), pp. 165–173. doi: `10.1016/j.oceaneng.2013.10.012`.

[31] Ning Gu et al. “A CFD-FEA Method for Hydroelastic Analysis of Floating Structures”. In: *Journal of Marine Science and Engineering* 11.4 (Mar. 2023), pp. 737–737. doi: `10.3390/jmse11040737`.

[32] O. Anwar Bég et al. “Two-way fluid-structure interaction hydroelastic simulation of vibrating membranes with applications in marine engineering using ANSYS fluent FSI”. In: (Apr. 2020).

[33] Oriol Colomés et al. "A monolithic finite element formulation for the hydroelastic analysis of very large floating structures". In: *International Journal for Numerical Methods in Engineering* (Oct. 2022). doi: 10.1002/nme.7140.

[34] Miroslav Pastor, Michal Binda, and Tomáš Harčarik. "Modal Assurance Criterion". In: *Procedia Engineering* 48 (2012). Modelling of Mechanical and Mechatronics Systems, pp. 543–548. issn: 1877-7058. doi: https://doi.org/10.1016/j.proeng.2012.09.551. url: https://www.sciencedirect.com/science/article/pii/S1877705812046140.

[35] Joshua L. Proctor, Steven L. Brunton, and J. Nathan Kutz. "Generalizing Koopman Theory to Allow for Inputs and Control". In: *SIAM Journal on Applied Dynamical Systems* 17.1 (2018), pp. 909–930. doi: 10.1137/16M1062296. eprint: https://doi.org/10.1137/16M1062296. url: https://doi.org/10.1137/16M1062296.

[36] J.B. Rawlings, D.Q. Mayne, and M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2017. isbn: 9780975937730. url: https://books.google.nl/books?id=MrJctAEACAAJ.

[37] Larry Reed. "Quantum Wave Mechanics Ch 30 Laplacian". In: Oct. 2022, pp. 309–314. isbn: ISBN 978-1-63492-964-6.

[38] *Research*. Feb. 2024. url: http://www.naoe.eng.osaka-u.ac.jp/naoe/naoe4/research/?lang=en.

[39] Riccardo Mariani et al. "Analysis of the global bending modes of a floating structure using the proper orthogonal decomposition". In: *Journal of Fluids and Structures* 28 (Jan. 2012). MAG ID: 2048350275 S2ID: ca1d268e1fdeb32838bd214cc5bbc68cb803f168, pp. 115–134. doi: 10.1016/j.jfluidstructs.2011.11.009.

[40] John V. Ringwood et al. "The wave energy converter control competition (WECCCOMP): Wave energy control algorithms compared in both simulation and tank testing". In: *Applied Ocean Research* 138 (2023), p. 103653. issn: 0141-1187. doi: https://doi.org/10.1016/j.apor.2023.103653. url: https://www.sciencedirect.com/science/article/pii/S0141118723001943.

[41] Sander Tonkens et al. "Soft Robot Optimal Control Via Reduced Order Finite Element Models". In: *arXiv: Robotics* (2020). ARXIV_ID: 2011.02092 MAG ID: 3096150585 S2ID: c56865b9f00b56aced30cf06942dfbad9546265f. doi: 10.1109/icra48506.2021.9560999 .

[42] Sadaf Shariati. "Linear model predictive control based on state space models combined with multi-rate state estimation using moving horizon estimation". PhD thesis. Sept. 2011.

[43] S. Sudhahar, C. Ganesh Babu, and D. Sharmila. "Kalman Filter-Based MPC Control Design and Performance Assessment of MIMO System". In: *Data Engineering and Communication Technology*. Ed. by K. Ashoka Reddy et al. Singapore: Springer Singapore, 2021, pp. 305–314. isbn: 978-981-16-0081-4.

[44] *Superlative floating solar farms*. June 2021. url: https://www.en-former.com/en/superlative-floating-solar-farms/.

[45] Tianyi Zhao et al. "Machine Learning-Based Reduced-Order Modeling and Predictive Control of Nonlinear Processes". In: *Chemical engineering research & design* (Feb. 2022). doi: 10.1016/j.cherd.2022.02.005.

[46] Zhao Wu et al. "Proper orthogonal decomposition and dynamic mode decomposition of jet in channel crossflow". In: *Nuclear Engineering and Design* 344 (2019), pp. 54–68. issn: 0029-5493. doi: https://doi.org/10.1016/j.nucengdes.2019.01.015. url: https://www.sciencedirect.com/science/article/pii/S0029549318311026.

[47] Shuyan Xia et al. "Vibration Control of Multi-Modular VLFS in Random Sea Based on Stiffness-Adjustable Connectors". In: *Applied Sciences* 12.3 (Jan. 2022), pp. 1117–1117. doi: 10.3390/app12031117.

[48] Guangbin Xun et al. “Active Shape Adjustment of Large Cable-Mesh Reflectors Using Novel Fast Model Predictive Control”. In: *Journal of Aerospace Engineering* 31.4 (2018). _eprint: https://ascelibrary.org/doi/pdf/10.1061/%28ASCE%29AS.1943-5525.0000858, p. 04018038. doi: 10.1061/(ASCE)AS.1943-5525.0000858. url: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29AS.1943-5525.0000858.

[49] Jia Sheng Yang and Rui Ping Gao. “Active Control of a Very Large Floating Beam Structure”. In: *Journal of Vibration and Acoustics* 138.2 (Jan. 2016), p. 021010. issn: 1048-9002. doi: 10.1115/1.4032039. url: https://doi.org/10.1115/1.4032039.

[50] Yong-Ho Kim and Jan Heiland. “Convolutional Autoencoders, Clustering and POD for Low-dimensional Parametrization of Navier-Stokes Equations”. In: *arXiv.org* (Feb. 2023). doi: 10.48550/arxiv.2302.01278.

A

Finite Element Model - weak form

In this section we will solve the governing equations found in chapter 3 by the use of a Finite Element Model. This model will be used to generate data for the DMDc and will be used as the plant that we opt to control.

A.1. Weak form

First, we need to derive the so-called *weak formulation* of the considered problem in order to be able to solve the problem. Let us define a function space $\mathcal{V}$ such that $\mathcal{V} \overset{\text{def}}{=} L^2(0, T; H^2(\Omega))$. In other words, we have a set of functions, from a Sobolev space H^2,living in our fluid domain Ω that vary in time from 0 to T and are square-integrable up to their second-order derivative [33]. Furthermore, we define the trace spaces of $\mathcal{V}$, which are Sobolev spaces associated with the boundaries of the domain Ω. These trace spaces consist of the restrictions of functions from the Sobolev space to the boundary, representing the collections of function values and, where appropriate, the values of certain derivatives that 'live' on the boundary Γ. Let us therefore define the trace spaces $\mathcal{V}_{\Gamma_{\text{fs}}}$ and $\mathcal{V}_{\Gamma_{\text{str}}}$ [33]. Having defined these spaces we want to find our unknown quantities ϕ, κ and η such that they are part of these function spaces. Hence, we need to solve the following problem by finding $[\phi, \kappa, \eta] \in \mathcal{V} \times \mathcal{V}_{\Gamma_{\text{fs}}} \times \mathcal{V}_{\Gamma_{\text{str}}}$.

$$B([\phi, \kappa, \eta], [w, v, u]) = L([w, v, u]) \quad \forall [w, v, u] \in \mathcal{V} \times \mathcal{V}_{\Gamma_{\text{fs}}} \times \mathcal{V}_{\Gamma_{\text{str}}} \tag{A.1}$$

which boils down to a bi-linear term B on the left-hand side and the linear term L on the right-hand side. For simplicity, let us consider it as a system with the quantities we want to solve encapsulated in the bi-linear term B where the linear term L represents the source. By multiplying Eq. 3.4 with the so-called test function w and integrating it over the full domain we find the weak form

$$\int \nabla^2 \phi \, w \, d\Omega = 0 \tag{A.2}$$

Now let us integrate Eq.3.4 by parts and use the *product rule of differentation* to obtain the following

$$\int \nabla^2 \phi \, w \, d\Omega = \int_\Omega \nabla \phi \, \nabla w \, d\Omega - \underbrace{\int_\Gamma (\mathbf{n} \cdot \nabla \phi) \, w \, d\Gamma}_{\text{boundary term}} \tag{A.3}$$

Note how integration by parts naturally resulted in a boundary term that looks similar to the boundary conditions defined in Eq.3.5 up to Eq.3.9. Let us now use the following notation such that the upcoming derivation stays clear [33]

$$(u,v)_\Omega \overset{\text{def}}{=} \int_\Omega u(\mathbf{x})\,v(\mathbf{x})\,d\Omega \tag{A.4}$$

$$(u,v)_\Gamma \overset{\text{def}}{=} \int_\Gamma u(\mathbf{x})\,v(\mathbf{x})\,d\Gamma \tag{A.5}$$

Hence, it is more convenient to split this boundary term in Eq.A.3 into smaller chunks such that we can write the weak form as

$$(\nabla\phi,\nabla w)_\Omega - \underbrace{((\mathbf{n}\cdot\phi),w)_{\Gamma_{\text{b}}} - ((\mathbf{n}\cdot\phi),w)_{\Gamma_{\text{out}}} - ((\mathbf{n}\cdot\phi),w)_{\Gamma_{\text{fs}}} - ((\mathbf{n}\cdot\phi),w)_{\Gamma_{\text{str}}}}_{\text{boundary terms}} = \underbrace{((\mathbf{n}\cdot\phi),w)_{\Gamma_{\text{in}}}}_{\text{source term}} \tag{A.6}$$

Note that we moved the inlet boundary condition to the right-hand side as a source term. Having now decomposed the weak form we can consider each boundary separate to derive its part of the weak form.

A.2. Free surface - Γ_{fs}

The free surface can be decomposed into three parts: the first and second damping zones Γ_{d1} and Γ_{d2} near the inlet and outlet boundary and an actual free surface part where there is no damping $\Gamma_{fs'}$, such that

$$\Gamma_{fs} = \Gamma_{d1} \cup \Gamma_{fs'} \cup \Gamma_{d2} \tag{A.7}$$

where

$$((\mathbf{n}\cdot\phi),w)_{\Gamma_{\text{fs}}} = ((n\cdot\phi),w)_{\Gamma_{\text{d1}}} + ((\mathbf{n}\cdot\phi),w)_{\Gamma_{\text{fs'}}} + ((\mathbf{n}\cdot\phi),w)_{\Gamma_{\text{d2}}} \tag{A.8}$$

A.2.1. Free surface (no damping)

Let us recall the kinematic boundary condition from Eq. 3.28 and the dynamic boundary from Eq. 3.29 condition where we find for $\Gamma_{fs'}$

$$\text{Kinematic B.C.}: \quad \mathbf{n}\cdot\nabla\phi = \kappa_t \quad \text{on} \quad \Gamma_{fs'} \tag{A.9}$$

$$\text{Dynamic B.C.}: \quad \phi_t + g\kappa = 0 \quad \text{on} \quad \Gamma_{fs'} \tag{A.10}$$

Now we can compute the weak form as previously done by multiplying the kinematic boundary condition with the test function w and integrate over $\Gamma_{fs'}$. For stability reasons as described by [33] we have to multiply the dynamic boundary condition by a modified test function $\alpha_f w + v$, integrate over $\Gamma_{fs'}$ and finally weighting it by β. Since Eq. A.10 is equal to zero we can subtract it from the left-hand side of Eq. A.9.

$$((\mathbf{n}\cdot\phi),w)_{\Gamma_{\text{fs'}}} = (\kappa_t,w)_{\Gamma_{\text{fs'}}} - \beta(\phi_t + g\kappa, \alpha_f w + v)_{\Gamma_{\text{fs'}}} \tag{A.11}$$

where according to [33]

$$\alpha_f = \frac{-i\omega}{\beta g}(1-\beta) \quad ; \quad \beta = 0.5$$

A.2.2. Damping zone 1

Again using the kinematic boundary condition from Eq.3.28 and the dynamic boundary condition from Eq. 3.29 we find for Γ_{d1}

$$\text{Kinematic B.C.}: \quad \mathbf{n}\cdot\nabla\phi = \kappa_t + \mu_{2,in}(\kappa - \kappa_I) \quad \text{on} \quad \Gamma_{d1} \tag{A.12}$$

$$\text{Dynamic B.C.}: \quad \phi_t + g\kappa + \mu_{1,in}(\mathbf{n}\cdot\nabla(\phi - \phi_I)) = 0 \quad \text{on} \quad \Gamma_{d1} \tag{A.13}$$

where we take

$$\mu_{1,in} = \mu_1(x) \quad \text{if} \quad x < L_{d1}$$
$$\mu_{2,in} = k_\lambda \mu_1(x)$$

Following the same procedure as for the un-damped free surface where we multiply the kinematic boundary condition by w and the dynamic boundary condition by the modified test function we obtain

$$\begin{aligned}(\mathbf{n}\cdot\nabla\phi, w)_{\Gamma_{\text{d1}}} &= (\kappa_t, w)_{\Gamma_{\text{d1}}} + (\mu_{2,in}\kappa, w)_{\Gamma_{\text{d1}}} \\ &\quad - \beta(\phi_t + g\kappa, \alpha_f w + v)_{\Gamma_{\text{d1}}} \\ &\quad - (\mu_{1,in}\mathbf{n}\cdot\nabla\phi, \alpha_f w + v)_{\Gamma_{\text{d1}}} \\ &\quad + \underbrace{(\mu_{1,in}\mathbf{n}\cdot\phi_I, \alpha_f w + v)_{\Gamma_{\text{d1}}} - (\mu_{2,in}\kappa_I, w)_{\Gamma_{\text{d1}}}}_{\text{source terms}} = 0 \end{aligned} \tag{A.14}$$

A.2.3. Damping zone 2

The second damping zone is very similar to the first damping zone except that the incident terms found in Eq.3.26 and Eq. 3.27 are now zero. Hence for Γ_{d2} we find

$$\text{Kinematic B.C. :}\quad \mathbf{n}\cdot\nabla\phi = \kappa_t + \mu_{2,out}\kappa \qquad \text{on}\quad \Gamma_{d2} \tag{A.15}$$

$$\text{Dynamic B.C. :}\quad \phi_t + g\kappa + \mu_{1,out}\mathbf{n}\cdot\nabla\phi = 0 \qquad \text{on}\quad \Gamma_{d2} \tag{A.16}$$

where we take

$$\begin{aligned}\mu_{1,out} &= \mu_1(x) \quad \text{if} \quad x > L_\Omega - L_{d2} \\ \mu_{2,out} &= k_\lambda\mu_1(x)\end{aligned}$$

Self-evidently, we find

$$\begin{aligned}(\mathbf{n}\cdot\nabla\phi, w)_{\Gamma_{\text{d2}}} &= (\kappa_t, w)_{\Gamma_{\text{d2}}} + (\mu_{2,out}\kappa, w)_{\Gamma_{\text{d2}}} \\ &\quad - \beta(\phi_t + g\kappa, \alpha_f w + v)_{\Gamma_{\text{d2}}} \\ &\quad - (\mu_{1,out}\mathbf{n}\cdot\nabla\phi, \alpha_f w + v)_{\Gamma_{\text{d2}}} = 0\end{aligned} \tag{A.17}$$

A.3. Structure - Γ_{str}

Before deriving the weak form of the governing equations found on the structural domain we need to introduce a set of interior points Λ_b, as can be seen in Fig. A.1. These interior points can be considered as interface boundaries we place in between the discretized elements of the structure where we can apply jump conditions that will arise while deriving the weak form of the Euler-Bernoulli equation [13]. However, since we apply rotational joints in between the elements we have to take these into account by defining them as the interior points Λ_j. Hence, we can treat these joints similarly to the interior points Λ_b, except that for the joints we simply apply the interface conditions defined in Eqs. 3.17, 3.18 and 3.19. Let us recall the governing Eq. 3.14.

$$\frac{\rho_b h_b}{\rho_w}\eta_{tt} + \frac{D}{\rho_w}\Delta^2\eta + \phi_t + g\eta = 0 \quad \text{on} \quad \Gamma_{str} \tag{A.18}$$

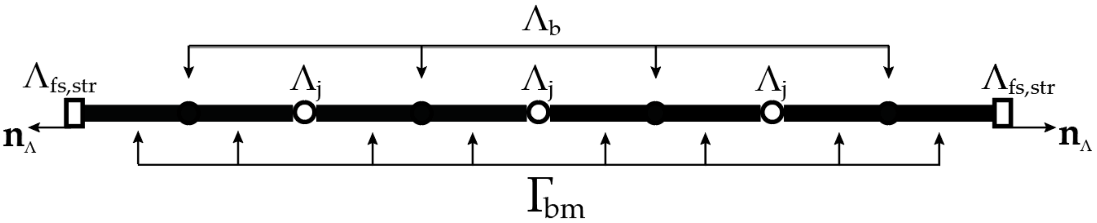


Figure A.1: Schematic representation of a discretized structural domain showing the placement of interior points Λ_b as interface boundaries between elements Γ_{bm}. Additionally, rotational joints denoted by Λ_j are introduced to account for specific interface conditions according to Eqs. 3.17, 3.18, and 3.19. At the ends of the structure we find the exterior boundaries $\Lambda_{fs,str}$.

where we have a fourth-order derivative of the structure deformation η with respect to space. In this study we derive the weak form of eq. A.18 using the proposed method by [13] and that is applied by [33]. This method proposes to solve fourth-order elliptic problems using Lagrangian polynomials. Since Lagrangian polynomials are not C^0 continuous, meaning (higher) derivatives of the polynomial are not continuous across the elements, we will have to penalize or eliminate jumping terms arising in the weak form by applying some jump conditions at the interior points Λ_b. Let us therefore first derive the weak form of the fourth-order derivative and later on we can consider the other parts in Eq. A.18.

Hence, we multiply this fourth-order derivative with our test function u, integrate over each element Γ_{bm} and sum all the local contributions as follows

$$\sum_{\Gamma_{bm}} \left[\int_{\Gamma_{bm}} \Delta \left(\frac{D}{\rho_w} \Delta \eta \right) u \, d\Gamma_{bm} \right] \tag{A.19}$$

Now integrating once by parts we obtain

$$\sum_{\Gamma_{bm} \in \Gamma_{str}} \left[-\int_{\Gamma_{bm}} \nabla \left(\frac{D}{\rho_w} \Delta \eta \right) \nabla u \, d\Gamma_{bm} + \sum_{E \in \partial \Gamma_{bm}} \int_{E} \nabla \left(\frac{D}{\rho_w} \Delta \eta \right) \cdot \mathbf{n}_\Lambda u \, dE \right] \tag{A.20}$$

where E denotes the boundaries on the ends of the discretized beam elements. For 1D elements these can be the exterior boundaries $\Lambda_{fs,str}$, the interior points Λ_b or the joints Λ_j. For 1D elements we find that the integral of the form $\int_E f(x) \, dE$ is simply the function $f(x)$ evaluated at x_b minus the the evaluation at x_a. Now because in Eq. A.20 we are summing over all boundary ends of the discretized elements Γ_{bm} we find that for two neighboring elements K and $K+1$ a similar term like the following will arise

$$\text{jump}(f(x_b)) = \|f(x_b)\| = f_{K+1}(x_b) - f_K(x_b) \tag{A.21}$$

Hence, between elements we find these discontinuous quantities which has to be dealt with. Let us use the property $\|fg\| = \|f\|\langle g \rangle + \langle f \rangle \|g\|$ such that we can rewrite Eq. A.20 into

$$\begin{aligned} &\sum_{\Gamma_{bm} \in \Gamma_{str}} \left[-\int_{\Gamma_{bm}} \nabla \left(\frac{D}{\rho_w} \Delta \eta \right) \nabla u \, d\Gamma_{bm} \right] + \\ &\sum_{E \in \partial \Gamma_{bm}} \left[\int_{E} \left\| \nabla \left(\frac{D}{\rho_w} \Delta \eta \right) \cdot \mathbf{n}_\Lambda \right\| \langle u \rangle + \left\langle \nabla \left(\frac{D}{\rho_w} \Delta \eta \right) \cdot \mathbf{n}_\Lambda \right\rangle \|u\| \, dE \right] \end{aligned} \tag{A.22}$$

Contrary to Eq.A.20 we now sum once for each joint, interior point or exterior point since we have applied the jump operator $\|\cdot\|$ that takes the jumps between the elements into account. For both the rotational joints Λ_j as the interior points Λ_b we have that the shear force across the elements should be continuous since there are no point loads applied at these points. Therefore, the jump in shear force, $\|\nabla (D\Delta\eta) \cdot \mathbf{n}_\Lambda\|$ should be zero. Moreover, because we are working with Lagrangian polynomials that are C^0 continuous we find that the jump in our test function u, $\|u\|$, should also be zero. As for the exterior boundaries, where it goes that the moment and shear force are zero, we find that these terms will also naturally drop. Hence, eliminating these terms from Eq. A.22 we obtain

$$\sum_{\Gamma_{bm} \in \Gamma_{str}} \left[-\int_{\Gamma_{bm}} \nabla \left(\frac{D}{\rho_w} \Delta \eta \right) \nabla u \, d\Gamma_{bm} \right] \tag{A.23}$$

which we can again integrate by parts to obtain

$$\begin{aligned}
&\sum_{\Gamma_{bm}\in\Gamma_{str}}\left[\int_{\Gamma_{bm}}\left(\frac{D}{\rho_w}\Delta\eta\right)\Delta u\, d\Gamma_{bm}\right]+\\
&\sum_{\Lambda_b\in\partial\Gamma_{bm}}\left[\int_{\Lambda_b}-\left\langle\frac{D}{\rho_w}\Delta\eta\right\rangle\|\nabla u\cdot\mathbf{n}_\Lambda\|\, d\Lambda_b\right]+\\
&\sum_{\Lambda_j\in\partial\Gamma_{bm}}\left[\int_{\Lambda_j}-\underbrace{\left\langle\frac{D}{\rho_w}\Delta\eta\right\rangle}_{=\frac{-k_{\varphi,j}}{\rho_w}\|\nabla\eta\cdot\mathbf{n}_\Lambda\|+\frac{M_{\text{ext}}}{\rho_w}}\|\nabla u\cdot\mathbf{n}_\Lambda\|\, d\Lambda_j\right]
\end{aligned}\tag{A.24}$$

where again in this case the exterior boundary $\Lambda_{fs,str}$ term is eliminated naturally because of the zero moment and zero shear conditions we imposed. Note how we can now substitute the interface conditions of the rotational joints from Eq. 3.17 into Eq. A.24. As one can see in Eq. A.24 we have derivatives of the shape function arising which we want to enforce continuity on. We do this in a weak sense by adding penalty terms such that we penalize the jump of derivatives across the elements, $\|\nabla\eta\cdot\mathbf{n}_\Lambda\|$. Let us express the union of all interior points Λ_b as Λ_B. Now combining all the integrals and adding the penalizing terms we obtain the following

$$\begin{aligned}
&\int_\Omega\frac{D}{\rho_w}\Delta\eta\Delta u\, d\Gamma_{str}\\
&+\int_{\Lambda_B}\left[-\left\langle\frac{D}{\rho_w}\Delta\eta\right\rangle\|\nabla u\cdot\mathbf{n}_\Lambda\|-\left\langle\frac{D}{\rho_w}\Delta u\right\rangle\|\nabla\eta\cdot\mathbf{n}_\Lambda\|+\frac{D\gamma}{h}\|\nabla\eta\cdot\mathbf{n}_\Lambda\|\,\|\nabla u\cdot\mathbf{n}_\Lambda\|\right]d\Lambda_B\\
&+\sum_{\Lambda_j\in\partial\Gamma_{bm}}\left[\int_{\Lambda_j}\frac{k_{\varphi,j}}{\rho_w}\|\nabla\eta\cdot\mathbf{n}_\Lambda\|\,\|\nabla u\cdot\mathbf{n}_\Lambda\|-\frac{M_{\text{ext}}}{\rho_w}\|\nabla u\cdot\mathbf{n}_\Lambda\|\, d\Lambda_j\right]
\end{aligned}\tag{A.25}$$

where similar to [33] we take $\gamma = 1$. Note how the control input can be moved to the source term since it is a term independent of the current deformation of the structure.

As mentioned before we can now assemble the entire weak form of Eq. A.18 by multiplying each term with its corresponding test function and integrating over the corresponding domain. Since the weak form of Eq. A.18 equals zero we can subtract it from the weak form of Eq.3.9 as we have similarly been doing for the previous cases. Hence, we obtain

$$\begin{aligned}
(\mathbf{n}\cdot\nabla\phi, w)_{\Gamma_{\text{str}}} &= (\eta_t, w)_{\Gamma_{\text{str}}}-\int_\Omega\frac{D}{\rho_w}\Delta\eta\Delta u\, d\Gamma_{str}\\
&-\int_{\Lambda_B}\left[-\left\langle\frac{D}{\rho_w}\Delta\eta\right\rangle\|\nabla u\cdot\mathbf{n}_\Lambda\|-\left\langle\frac{D}{\rho_w}\Delta u\right\rangle\|\nabla\eta\cdot\mathbf{n}_\Lambda\|+\frac{D\gamma}{h}\|\nabla\eta\cdot\mathbf{n}_\Lambda\|\,\|\nabla u\cdot\mathbf{n}_\Lambda\|\right]d\Lambda_B\\
&-\sum_{\Lambda_j\in\partial\Gamma_{bm}}\left[\int_{\Lambda_j}\frac{k_{\varphi,j}}{\rho_w}\|\nabla\eta\cdot\mathbf{n}_\Lambda\|\,\|\nabla u\cdot\mathbf{n}_\Lambda\|-\frac{M_{\text{ext}}}{\rho_w}\|\nabla u\cdot\mathbf{n}_\Lambda\|\, d\Lambda_j\right]\\
&+\left(\frac{\rho_b h_b}{\rho_w}\eta_{tt}, u\right)_{\Gamma_{\text{str}}}+((\phi_t+g\eta), u)_{\Gamma_{\text{str}}}
\end{aligned}\tag{A.26}$$

A.4. Outlet, inlet and bottom - Γ_{out}, Γ_{in}, Γ_b

For the outlet boundary condition we recall Eq.3.25 where for the weak form we find

$$(\mathbf{n}\cdot\nabla\phi, w)_{\Gamma_{\text{out}}} = 0 \tag{A.27}$$

For the inlet boundary condition, which is considered as a source term, we recall Eq. 3.24 where for the weak form we find

$$(\mathbf{n}\cdot\nabla\phi, w)_{\Gamma_{\text{in}}} = (u_{in}, w)_{\Gamma_{\text{in}}} \tag{A.28}$$

For the bottom boundary condition we recall Eq. 3.5 where for the weak form we find

$$(\mathbf{n}\cdot\nabla\phi, w)_{\Gamma_{\mathsf{b}}} = 0 \tag{A.29}$$

A.5. Bi-linear and linear term

We can now substitute all the terms in Eq. A.6 that we have derived to find the bi-linear term and the linear term. Since the bi-linear term is a lengthy expression we will express the complete weak form as follows

$$B([\phi,\kappa,\eta],[w,v,u]) = \underbrace{(u_{in}, w)_{\Gamma_{\mathsf{in}}} + (\mu_{1,in}\mathbf{n}\cdot\phi_I, \alpha_f w + v)_{\Gamma_{\mathsf{d1}}} - (\mu_{2,in}\kappa_I, w)_{\Gamma_{\mathsf{d1}}} + \left(\frac{M_{\mathsf{ext}}}{\rho_w}, \|\nabla u\cdot\mathbf{n}_\Lambda\|\right)_{\Lambda_j}}_{L([w,v,u])} \tag{A.30}$$

A.6. Spatial discretization

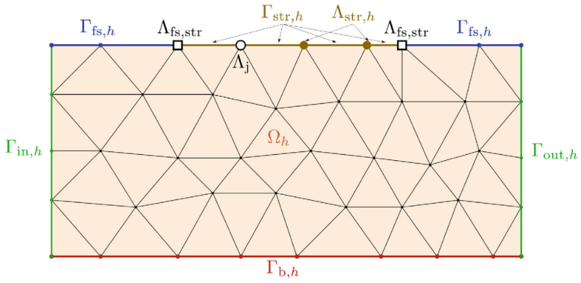


Figure A.2: The scheme depicting the discretized entities of the considered problem [33].

Given the fluid domain Ω which we discretize in triangles such that we obtain the triangulation of the domain denoted as Ω_h, as can be seen in Fig.A.2. The triangulation Ω_h is done such that the shapes and angles of our predefined boundaries found in the problem setup are preserved. We denote our discretized boundaries as a set of facets that lie on the boundary of Ω as $\Gamma_{\mathsf{in},h}$, $\Gamma_{\mathsf{b},h}$, $\Gamma_{\mathsf{out},h}$, $\Gamma_{\mathsf{fs},h}$ and $\Gamma_{\mathsf{str},h}$. Additionally, we will be dealing with $\Lambda_{\mathsf{str},h}$ which are the edges of the facets of the floating structure that are not the joints [33]. Let us define the finite-dimensional finite-element spaces for the velocity potential, free surface elevation and the structure deformation, $\mathcal{V}_h, \mathcal{V}_{\Gamma_{\mathsf{fs},h}}, \mathcal{V}_{\Gamma_{\mathsf{str},h}}$, respectively, as [33, 3]

$$\mathcal{V}_h = \left\{w_h \in C^0(\Omega) : w_h|_K \in \mathcal{P}_r(K), \forall K \in \Omega_h\right\} \tag{A.31}$$

$$\mathcal{V}_{\Gamma_{\mathsf{fs},h}} = \{w_h|_E : w_h \in \mathcal{V}_h, \forall E \in \Gamma_{\mathsf{fs},h}\} \tag{A.32}$$

$$\mathcal{V}_{\Gamma_{\mathsf{str},h}} = \{w_h|_E : w_h \in \mathcal{V}_h, \forall E \in \Gamma_{\mathsf{str},h}\} \tag{A.33}$$

where $\mathcal{P}_r(K)$ denotes the space of Lagrange polynomials of degree r, where $r = 4$ in this study, in a discretized element K and where E are the facets of the domain Ω. Finally, we can formulate our discretized problem as follows: *Find* $[\phi_h, \kappa_h, \eta_h] \in \mathcal{V}_h \times \mathcal{V}_{\Gamma_{\mathsf{fs},h}} \times \mathcal{V}_{\Gamma_{\mathsf{str},h}}$ *such that*

$$B([\phi_h,\kappa_h,\eta_h],[w_h,v_h,u_h]) = L([w_h,v_h,u_h]) \quad \forall [w_h,v_h,u_h] \in \mathcal{V}_h \times \mathcal{V}_{\Gamma_{\mathsf{fs},h}} \times \mathcal{V}_{\Gamma_{\mathsf{str},h}} \tag{A.34}$$

with the bilinear function B and linear function L derived and defined in section A.1.

A.7. Time discretization

For steady-state analysis it is often sufficient to study the system in frequency-domain. However, in the context of control steady-state analysis is often not applicable. It is therefore essential to analyse the system in the time domain because of the transient behaviour that will arise during the control. Let us recall Eq. 2.4 where as mentioned before these type of systems can be solved by the Newmark-β scheme. As similarly done by [33] we take $\gamma_t = 0.5$ and $\beta_t = 0.25$ since it is proven that these parameters results in an unconditionally stable scheme. Using the notation of Eq. 2.3 it is found that [33]

$$\mathbf{\Theta}^{k+1} = \mathbf{\Theta}^{k} + \Delta t \mathbf{\Theta}_t^{k} + \Delta t^2 \left[\left(\frac{1}{2} - \beta_t \right) \mathbf{\Theta}_{tt}^{k} + \beta_t \mathbf{\Theta}_{tt}^{k+1} \right] \tag{A.35}$$

$$\mathbf{\Theta}_t^{k+1} = \delta_t (\mathbf{\Theta}^{k+1} - \mathbf{\Theta}^{k}) + \frac{1-\gamma_t}{\beta_t} \mathbf{\Theta}_t^{k} + \Delta t \frac{1-\gamma_t}{2\beta_t} \mathbf{\Theta}_{tt}^{k} \tag{A.36}$$

$$\mathbf{\Theta}_{tt}^{k+1} = \delta_{tt} (\mathbf{\Theta}^{k+1} - \mathbf{\Theta}^{k}) + \frac{1-\gamma_t}{\beta_t \Delta t} \mathbf{\Theta}_t^{k} + \Delta t \frac{1-2\beta_t}{2\beta_t} \mathbf{\Theta}_{tt}^{k} \tag{A.37}$$

where we take $\delta_t = \frac{\gamma_t}{\beta_t \Delta t}$ and $\delta_{tt} = \frac{1}{\beta_t \Delta t^2}$. Furthermore, the notations $\{\cdot\}_t$ and $\{\cdot\}_{tt}$ denote the first and second order time-derivatives respectively. Substituting the right-hand side of Eq. A.37 into Eq. A.35 allow us to compute $\mathbf{\Theta}_{k+1}$ such that Eq. A.36 and Eq. A.37 can be computed explicitly.

B

LMPC - Full derivation of control law

1. Derivation of $\underline{\tilde{\mathbf{z}}}_k^\top \mathcal{Q} \underline{\tilde{\mathbf{z}}}_k$

Let us substitute Eq.3.68 for $\underline{\tilde{\mathbf{z}}_k}$ such that

$$\begin{aligned}(\underline{\tilde{\mathbf{p}}}_k - \underline{\tilde{\mathbf{r}}}_k)^\top \mathcal{Q}(\underline{\tilde{\mathbf{p}}}_k - \underline{\tilde{\mathbf{r}}}_k) &= (\mathcal{M}\tilde{\mathbf{x}}_k + \mathcal{C}\underline{\mathbf{u}}_k - \underline{\tilde{\mathbf{r}}}_k)^\top \mathcal{Q}(\mathcal{M}\tilde{\mathbf{x}}_k + \mathcal{C}\underline{\mathbf{u}}_k - \underline{\tilde{\mathbf{r}}}_k) \\ &= \tilde{\mathbf{x}}_k^\top \mathcal{M}^\top \mathcal{Q}\mathcal{M}\tilde{\mathbf{x}}_k + \tilde{\mathbf{x}}_k^\top \mathcal{M}^\top \mathcal{Q}\mathcal{C}\underline{\mathbf{u}}_k - \tilde{\mathbf{x}}_k^\top \mathcal{M}^\top \mathcal{Q}\underline{\tilde{\mathbf{r}}}_k \\ &+ \underline{\mathbf{u}}_k^\top \mathcal{C}^\top \mathcal{Q}\mathcal{M}\tilde{\mathbf{x}}_k + \underline{\mathbf{u}}_k^\top \mathcal{C}^\top \mathcal{Q}\mathcal{C}\underline{\mathbf{u}}_k - \underline{\mathbf{u}}_k^\top \mathcal{C}^\top \mathcal{Q}\underline{\tilde{\mathbf{r}}}_k - \underline{\tilde{\mathbf{r}}}_k^\top \mathcal{Q}\mathcal{M}\tilde{\mathbf{x}}_k - \underline{\tilde{\mathbf{r}}}_k^\top \mathcal{Q}\mathcal{C}\underline{\mathbf{u}}_k + \underline{\tilde{\mathbf{r}}}_k^\top \mathcal{Q}\underline{\tilde{\mathbf{r}}}_k\end{aligned} \tag{B.1}$$

where $\mathcal{Q}$ is a weight matrix which we can use to emphasize the tracking accuracy allowing us to tune our control sensitivity

$$\mathcal{Q} = \begin{bmatrix} \bar{\mathcal{Q}} & 0 & \cdots & 0 \\ 0 & \bar{\mathcal{Q}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \bar{\mathcal{Q}} \end{bmatrix}. \tag{B.2}$$

For simplicity we consider $\mathcal{Q} = \bar{\mathcal{Q}}\, I$. However, note that in general we are allowed to penalize each term in the forecast separately, allowing for finer control. For instance, one may want to have the signal $\tilde{\mathbf{p}}$ follow the desired trajectory $\tilde{\mathbf{r}}$ as soon as possible in the control horizon. This can be done by picking higher penalty terms found in the upper-left corner of this weight matrix $\mathcal{Q}$.

2. Derivation of $\underline{\mathbf{u}}_k^\top \mathcal{R}_1 \underline{\mathbf{u}}_k$

Now the second part found in the objective function Eq.3.71 is the part where we penalize the increments between the control inputs along the control horizon such that we can fine-tune the smoothness of the control signal, as discussed before. We can expand this term such that we obtain

$$\underline{\mathbf{u}}_k^\top \mathcal{R}_1 \underline{\mathbf{u}}_k = \zeta^2 \left((\mathbf{u}_{k+1|k} - \mathbf{u}_{k|k})^2 + (\mathbf{u}_{k+2|k} - \mathbf{u}_{k+1|k})^2 + \cdots + (\mathbf{u}_{k+N_c-1|k} - \mathbf{u}_{k+N_c-2|k})^2 \right) \tag{B.3}$$

where one can see the summation of the different increments with a penalty term ζ^2. In order to obtain this expanded form we need to design the $\mathcal{R}_1$ matrix such that it computes the control signal increments by acting on the vector $\underline{\mathbf{u}}_k$, which is expressed in Eq.3.69. Moreover, we include the penalty term ζ^2 which penalizes these increments to our liking. This can be done by constructing the $\mathcal{R}_1$ matrix such that:

$$\mathcal{R}_1 = \begin{bmatrix} -I & I & 0 & \cdots & 0 \\ 0 & -I & I & \cdots & 0 \\ 0 & 0 & -I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & -I & I \end{bmatrix}^\top \zeta^2 \begin{bmatrix} -I & I & 0 & \cdots & 0 \\ 0 & -I & I & \cdots & 0 \\ 0 & 0 & -I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & -I & I \end{bmatrix}_{N_c-1\times N_c} \tag{B.4}$$

3. Derivation of $\underline{\mathbf{u}}_k^\top \mathcal{R}_2 \underline{\mathbf{u}}_k$

Finally, we reach the derivation of the third term found in Eq.3.71. Note that this third term only acts on the first entry of the control signal over the horizon $\underline{\mathbf{u}}_k$.

$$\underline{\mathbf{u}}_k^\top \mathcal{R}_2 \underline{\mathbf{u}}_k = \zeta_0^2 \mathbf{u}_{k|k}^2 \tag{B.5}$$

Hence, the matrix $\mathcal{R}_2$ should only act on this very first entry. Furthermore, we include the penalty term ζ_0^2 which allow us to penalize the magnitude of this very first entry of $\underline{\mathbf{u}}_k$. Let us again recall that this first entry will be the actual control input we apply on the true system. Therefore, this entry is considered separately to allow for even more fine-tuning. This can be done by constructing the $\mathcal{R}_2$ matrix such that

$$\mathcal{R}_2 = \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}^\top \zeta_0^2 \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{N_c\times N_c} \tag{B.6}$$

4. Derivation control law

Let us now gather all the terms derived above and remove the terms that cancel each other out such that we reach the following form of J

$$J = \tilde{\mathbf{x}}_k^\top \mathcal{G} \tilde{\mathbf{x}}_k + 2\tilde{\mathbf{x}}_k \mathcal{F}^\top \underline{\mathbf{u}}_k - 2\tilde{\mathbf{x}}_k^\top \mathcal{W}^\top \underline{\tilde{\mathbf{r}}} + \underline{\mathbf{u}}_k^\top \mathcal{H} \underline{\mathbf{u}}_k - 2\underline{\mathbf{u}}_k^T \mathcal{S}^\top \underline{\tilde{\mathbf{r}}} + \underline{\tilde{\mathbf{r}}}^\top \mathcal{Q} \underline{\tilde{\mathbf{r}}} \tag{B.7}$$

and take the derivative of each component separately

$$\frac{\partial(\tilde{\mathbf{x}}_k^\top \mathcal{G} \tilde{\mathbf{x}}_k)}{\partial \underline{\mathbf{u}}} = \mathbf{0} \tag{B.8}$$

$$\frac{\partial(2\tilde{\mathbf{x}}_k \mathcal{F}^\top \underline{\mathbf{u}}_k)}{\partial \underline{\mathbf{u}}} = 2\mathcal{F}\tilde{\mathbf{x}}_k \tag{B.9}$$

$$\frac{\partial(-2\tilde{\mathbf{x}}_k^\top \mathcal{W}^\top \underline{\tilde{\mathbf{r}}})}{\partial \underline{\mathbf{u}}} = \mathbf{0} \tag{B.10}$$

$$\frac{\partial(\underline{\mathbf{u}}_k^\top \mathcal{H} \underline{\mathbf{u}}_k)}{\partial \underline{\mathbf{u}}} = 2\mathcal{H}\underline{\mathbf{u}}_k \tag{B.11}$$

$$\frac{\partial(-2\underline{\mathbf{u}}_k^\top \mathcal{S}^\top \underline{\tilde{\mathbf{r}}})}{\partial \underline{\mathbf{u}}} = -2\mathcal{S}^\top \underline{\tilde{\mathbf{r}}} \tag{B.12}$$

$$\frac{\partial(\underline{\tilde{\mathbf{r}}}^\top \mathcal{Q} \underline{\tilde{\mathbf{r}}})}{\partial \underline{\mathbf{u}}} = \mathbf{0} \tag{B.13}$$

Gathering all the derivatives:

$$2\mathcal{F}\tilde{\mathbf{x}}_k + 2\mathcal{H}\underline{\mathbf{u}}_k - 2\mathcal{S}^\top \underline{\tilde{\mathbf{r}}} = 0 \tag{B.14}$$

C

DMD modes

The DMD analysis, assuming the dynamics without influence of control, returns a diagonal matrix Λ where the entries are the corresponding eigenvalues of each DMD mode, denoted as λ_k [23]. The future dynamics of any considered system $\mathbf{x}$ can be projected in the future given an initial condition using the following convention

$$\mathbf{x}(t) \approx \sum_{k=1}^{r} \boldsymbol{\phi}_k \, e^{\omega_k t} \, b_k \tag{C.1}$$

where $\boldsymbol{\phi}_k$ represents the DMD modes, b_k is the initial amplitude of each mode and ω_k is the continuous complex frequency, with the real part a and imaginary part b, found using [23]

$$\omega_k = \frac{\ln \lambda_k}{\Delta t}. \tag{C.2}$$

Using Euler's theorem we find the time-dynamics to be a complex number such that

$$\omega_k = e^{at}(\cos bt + i \, \sin bt). \tag{C.3}$$

Hence, to study the time-dynamics of the DMD modes we distinguish two important parts

$$\text{Damping:} \quad \text{real}(\omega_k) = a \tag{C.4}$$

$$\text{Frequency:} \quad |\text{imag}(\omega_k)| = b. \tag{C.5}$$

Below you can find the time-dynamics of the DMD modes we opt to control for the considered case with a regular wave.

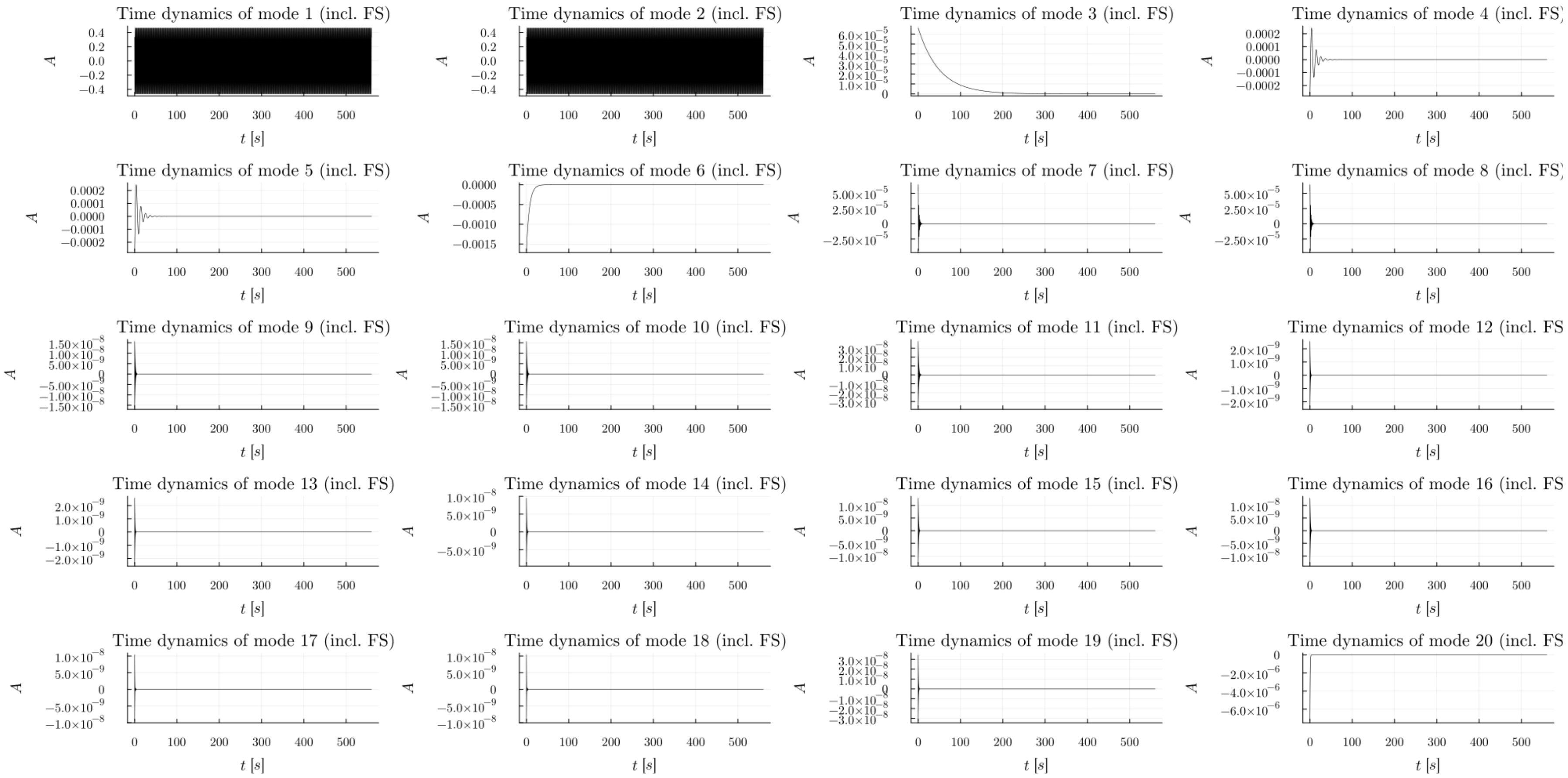


Figure C.1: Time dynamics for the first 20 DMD modes (regular wave condition).

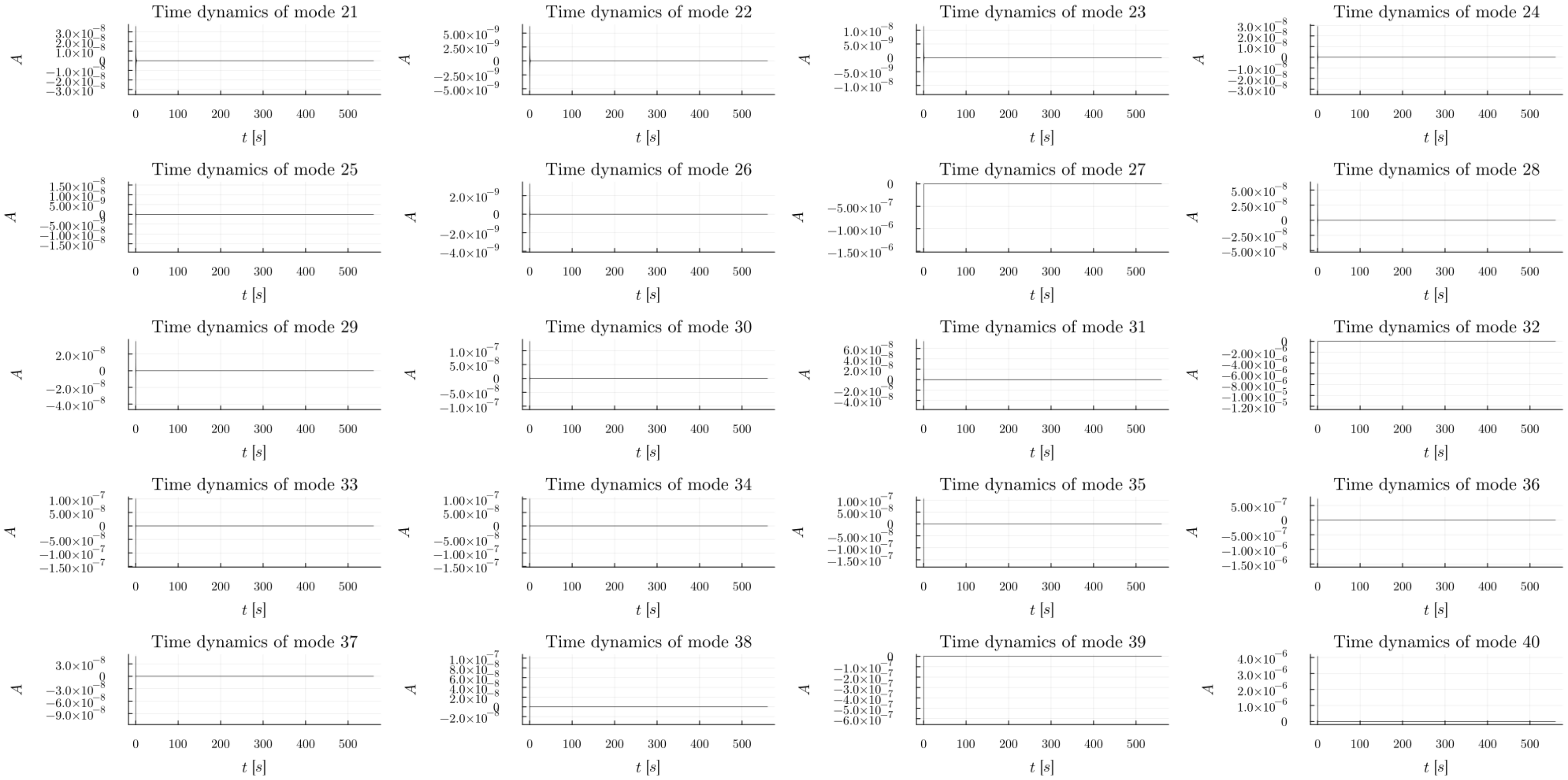


Figure C.2: Time dynamics for the last 20 DMD modes (regular wave condition).

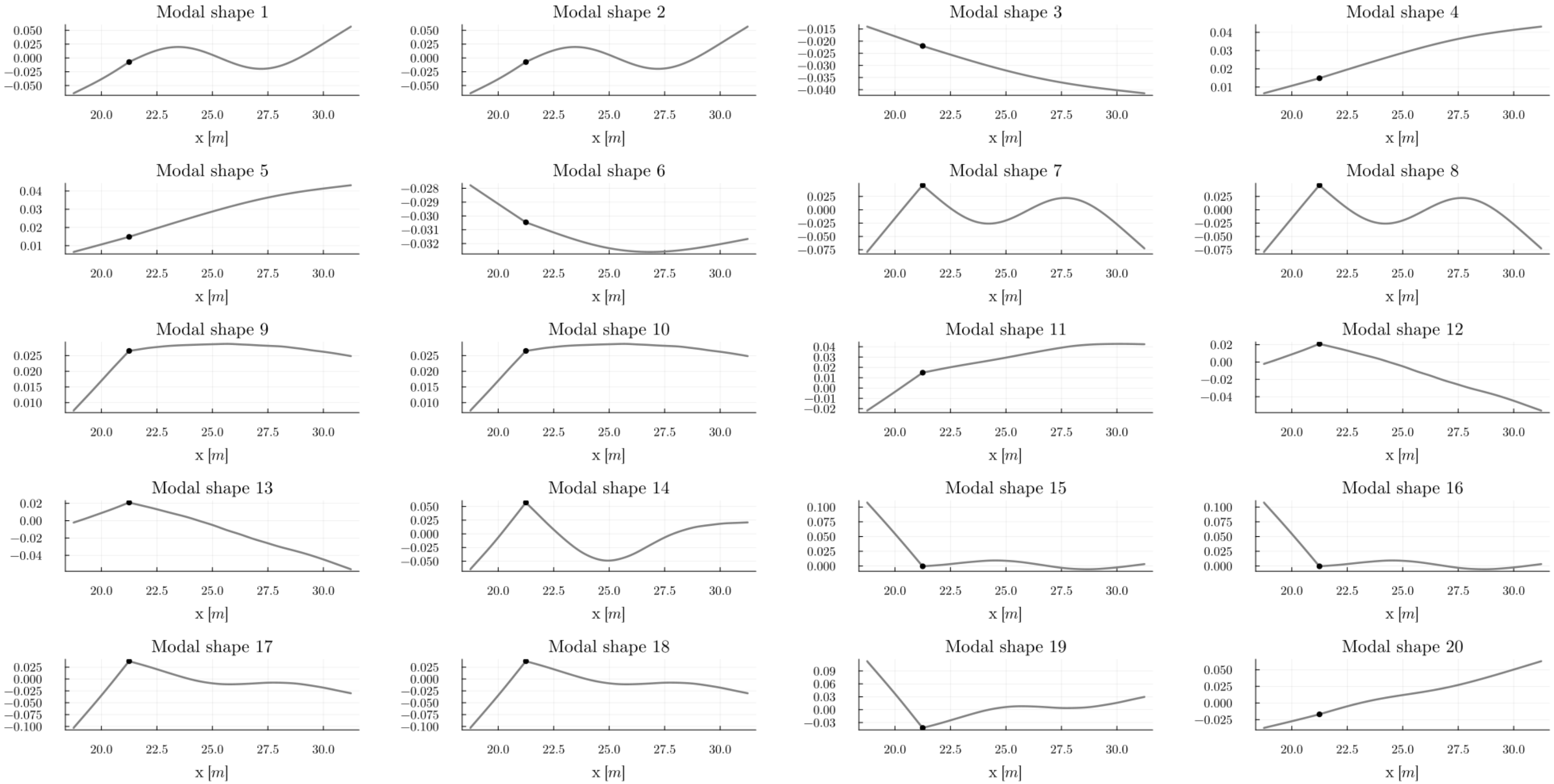


Figure C.3: The first 20 DMD modal-shapes (regular wave condition).

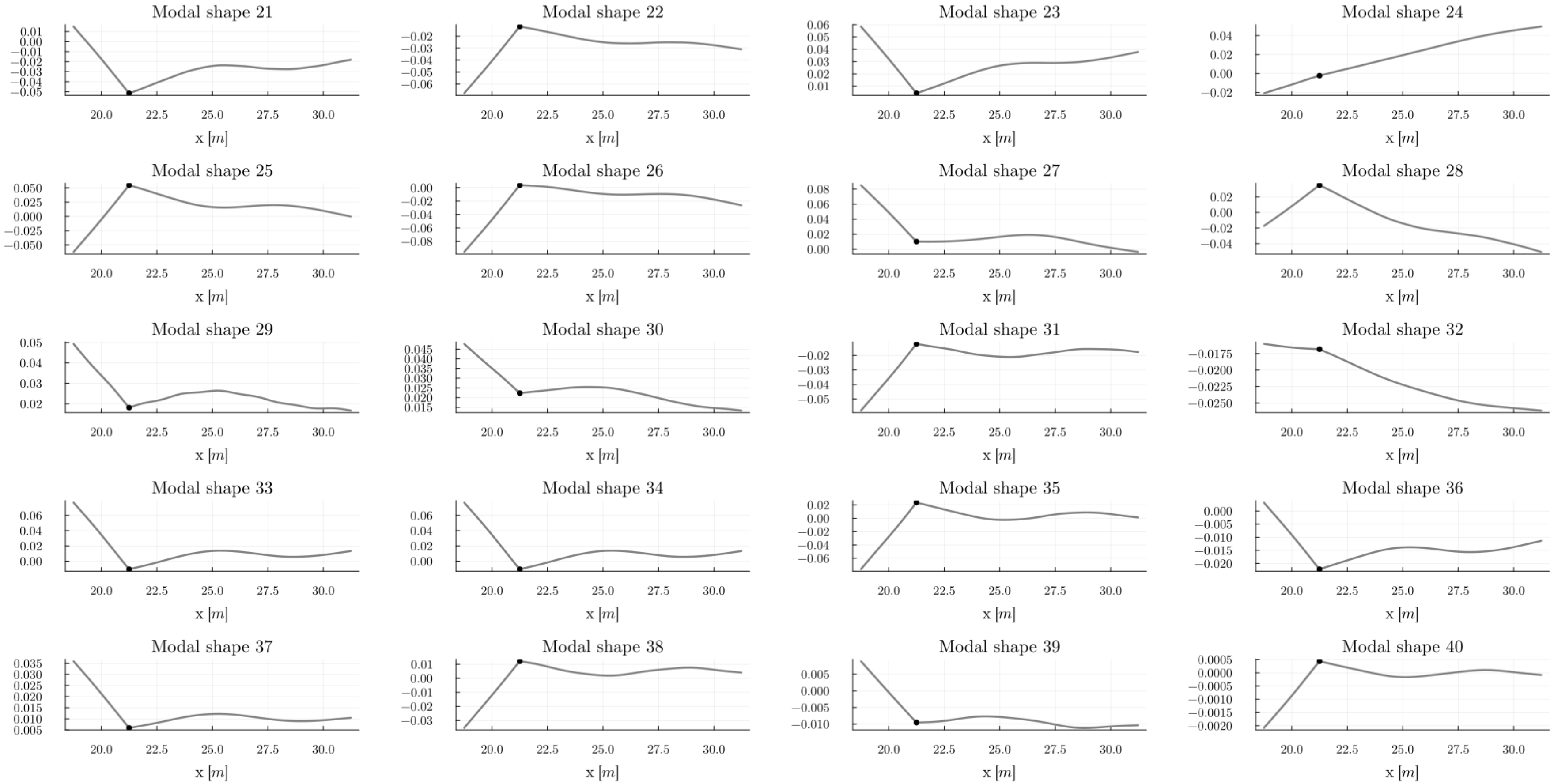


Figure C.4: The last 20 DMD modal-shapes (regular wave condition).

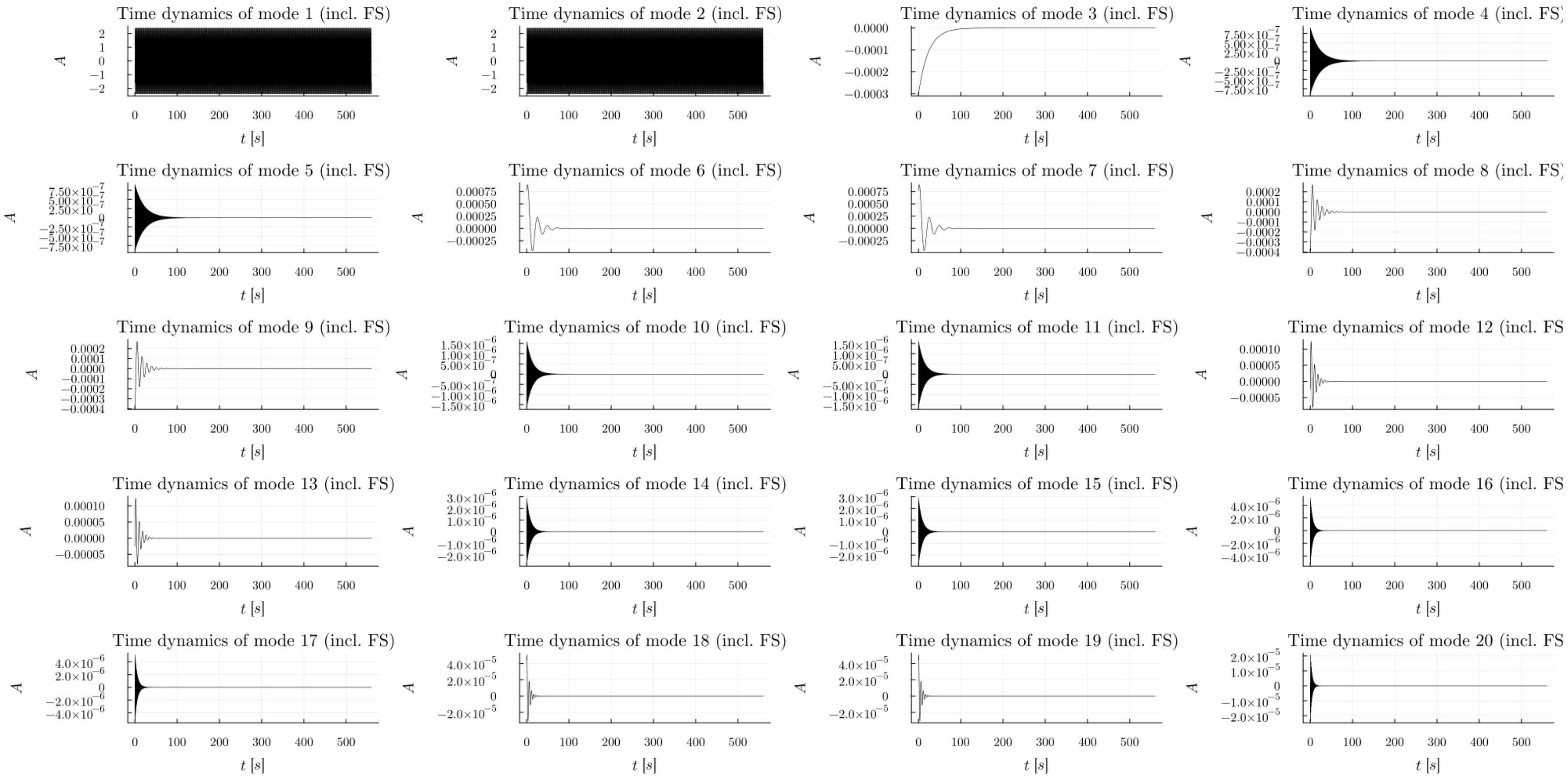


Figure C.5: Time dynamics for the first 20 DMD modes where the free-surface elevation is included in the DoFs (regular wave condition).

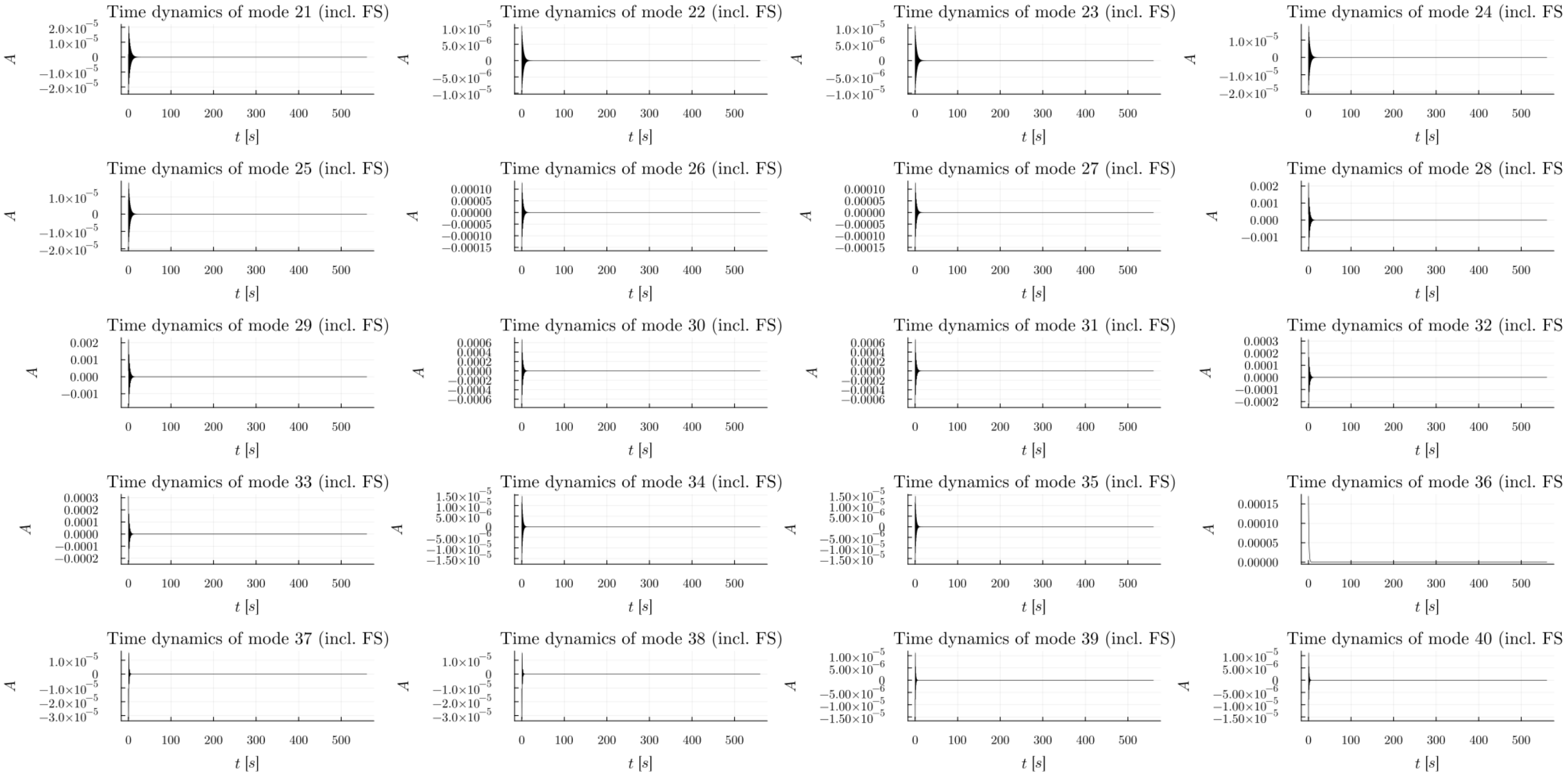


Figure C.6: Time dynamics for the last 20 DMD modes where the free-surface elevation is included in the DoFs (regular wave condition).

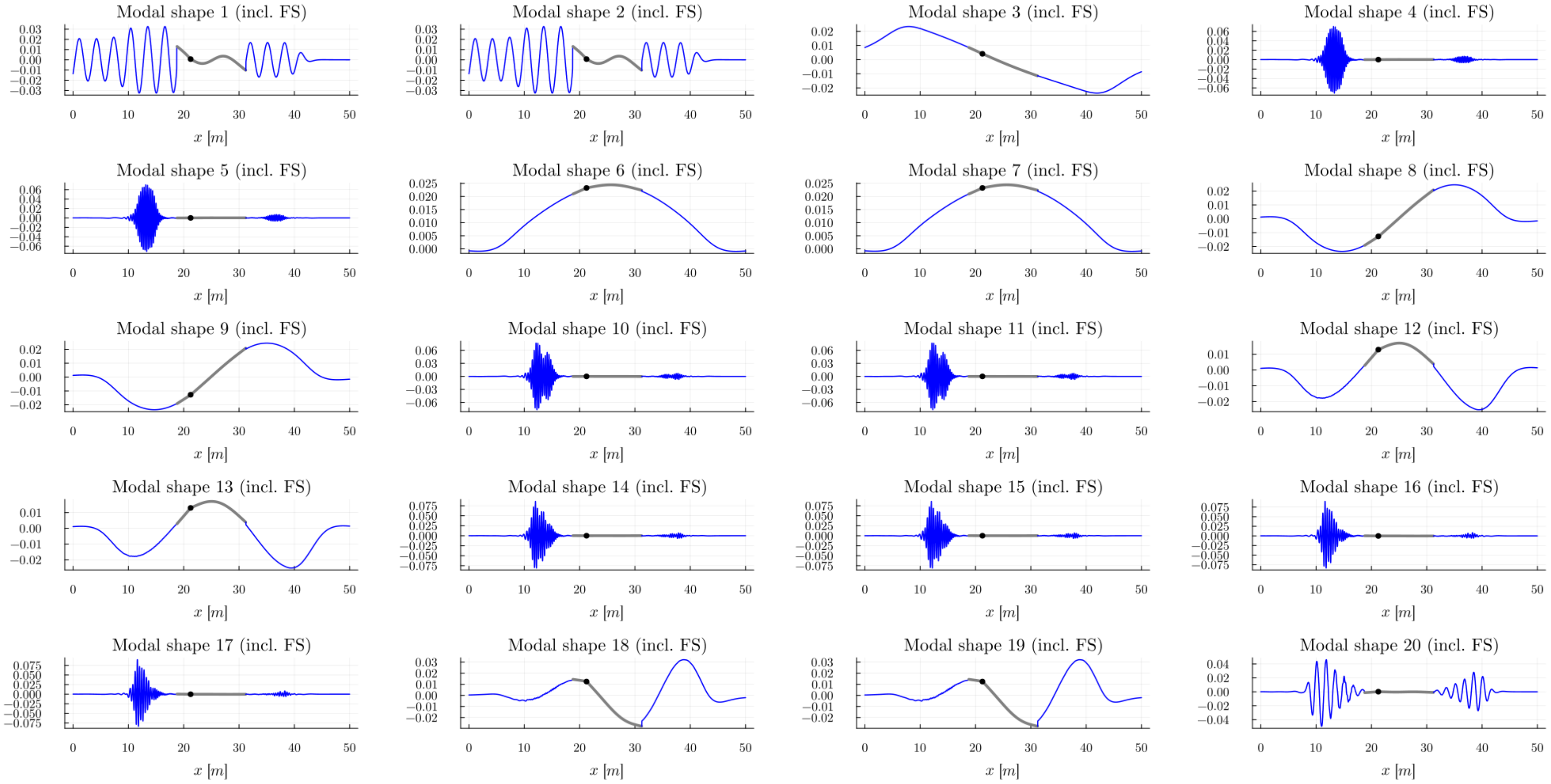


Figure C.7: The first 20 DMD modal-shapes where the free-surface elevation is included in the DoFs (regular wave condition).

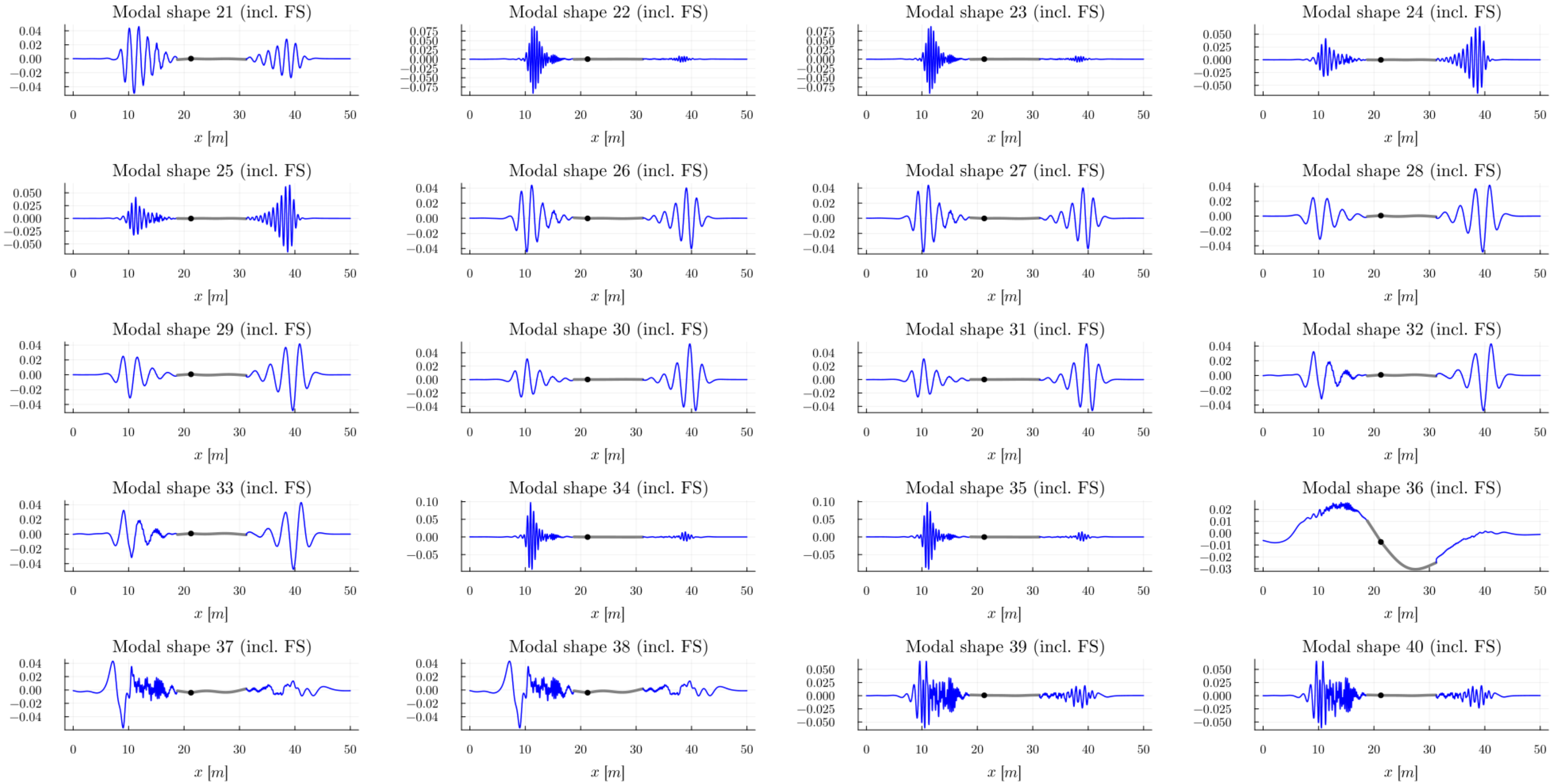


Figure C.8: The last 20 DMD modal-shapes where the free-surface elevation is included in the DoFs (regular wave condition).