Work-ability increase using tugger control

N.Sanders

Master of Science Thesis

TU Delft

Boskalis

Work-ability increase

using tugger control

by

N. Sanders

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday August 29, 2019 at 13:00.

Student number:	4175085	
Project duration:	November 25, 2018 – August 28, 2019	
Thesis committee:	Prof. dr. ir. A.V. Metrikine,	TU Delft, chairman
	Dr. H. Hendrikse,	TU Delft supervisor
	Ir. V. Koster,	Boskalis

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Acknowledgments

This report is written to conclude my Master of Science in Offshore and Dredging Engineering at the Delft University of Technology. I would like to take this opportunity to thank some people who have helped me throughout my Master and during the thesis.

First of all, I would like to thank all members of my graduation committee. I would like to thank Dr. Andrei Metrikine, for his insight, critical questions and feedback. Furthermore, I would like to state my gratitude towards Ir. Hayo Hendriks for his input, motivation and support during both the masters and thesis. A special thanks to Ir.Panagiotis Antonakas and Ir.Vincent Koster for their guidance and helping me finding the right tools and connections within Boskalis.

I would also like to thank Boskalis for the opportunity and interesting thesis subject. The tools and knowledge available helped me tackle the thesis. In addition a great thanks to the energy department of Boskalis. The positive work environment created by all the employees in general was of great help. Additionally, a special thanks to all fellow graduate students from Boskalis, for their support, guidance and a very pleasant time at Boskalis.

Furthermore, many thanks to all my friends from my hometown, bachelor, and master for their amazing contribution to my student life. A special thanks to Franciscus Huinink, and Martin Koole for their support during my whole student career.You both always had the ability to give me energy and motivation, and I am very grateful to count you as my friends.

Last but not least I want to especially thank my parents, Rene Sanders and Carmen Sanders and little brother, Jordy Sanders for their unconditional support. They motivated me to pursue my passion and dreams.

Nick Sanders
Den Haag, August 2019

Abstract

For the offshore market an increase in renewable projects can be observed. The technological innovations are continuously decreasing the costs of renewable projects. Especially offshore wind is getting more cost effective, and is receiving a lot of attention. However, with the growth of the sector, new challenges arise. Water depths are increasing, soil parameters worsening and greater distances from shore need to be overcome. Whereas prices of installation are under pressure. This drives the market towards larger and more innovative installation vessels. A trend can be observed towards monohull craning vessels which combine a large crane with a large storage space on deck. During lifting the monohull vessel experiences large motions of the lifting configuration at even small wave loading. The large motions are mainly the result of resonance within the system, which results in large crane forces. It is these forces which decreases the vessels work-ability. Applying damping to the system is a way to counter the resonance. The tugger winches can be used to apply damping to the lifting configuration.

The aim of the thesis is to investigate the potential damping effect of a tugger damping system for the Bokalift 1 during a jacket lifting operation. With special emphasis on the effect of different control systems of the tugger winches. Two different models are used in the thesis to research the effects of tugger damping on the dynamic behavior of the lifting configuration. Namely, a 2D matlab model, and a more extensive 3D Orcaflex model.

The Orcaflex model is build for researching the effects of tugger damping and different control systems. Comparing the responses of the model to airy waves loading. To have control in both longitudinal and transverse direction boom winches on the crane are used, in combination with deck winches. Which are located on both ends of the deck. The hydro static properties of the model are calculated with the program GHs. The hydro dynamic properties are calculated with the help of AQWA.

The matlab model is used to enable quick research on the effects of tugger damping and different control systems. The model is based on a 5 degree of freedom (DOF) mass-spring-damper-system. The equations of motion (EOM) are derived using the lagrange formalism with help of the program Maplesoft. The model is validated with help of proven software Orcaflex.

Seven different control systems are made and tested in the matlab model. The control systems are examined in two different simulations. Firstly the roll motion is studied when the model is subjected to wave loading. Secondly the damping capabilities for the lifting configuration with initial displacement are tested. The first analysis shows the PID controller has the highest roll motion reduction of the jacket. Linear control system scores higher than quadratic. The results from the second analysis shows that stepwise controller takes the shortest to fully damp the lifting configuration.

Following on the results of the 2D model analysis the 3D model will further inspect the effect of tugger winches for the linear, quadratic, and PID controller. A model analysis shows there are 4 important modes within the wave excitation range. The combination of these modes results in the highest crane forces at a wave period of 5.5s. The linear and quadratic control system are once again compared, only now in the 3d model. The assessment shows the linear model as more efficient in reducing the crane tip forces. Lastly the linear and PID control systems are compared towards each other and a model without tugger damping. Based on the results, the linear control system increases the total forces on the crane. Where as the PID reduces all forces.

It is shown that tugger damping does not necessarily decrease forces acting on the crane tip for head on waves. From the tested control system the PID controller reduces the forces most effectively and efficiently. However the effect of the PID controller depends on the loading wave frequency and for which it is tuned.

Abbreviations

O&G - oil and gas
t&I - transport and installation
RAO - Response amplitude operator
EOM - Equation of motion
DOF - Degree of freedom
ODE - Ordinary differential equations
SBJ - suction bucket jacket
COG - centre of gravity
COB - centre of buoyancy
LC - load case
DP - dynamic positioning
JLT - Jacket lifting tool
UC - Unity check

Contents

List of Figures

List of Tables

1

Introduction

The energy transition brings new difficult challenges with it, which require smart technological solutions. As offshore activity is pulled into deeper waters, heavy lifting from a floating vessel is becoming increasingly important. However, it is found that the vessels created for these installations experience trouble with motions caused by environmental loads during the lift. As these motions limit the work-ability of the ship, it is of great importance to find ways of limiting the dynamics of the lifted object.

1.1. Background

In 2015, 183 nations agreed to limit the global average temperature rise to a maximum of 2 degrees, also known as the Paris agreement. To achieve this, targets have been set for among other things emissions, clean energy production and energy efficiency [2] [1]. The agreement challenges the future of fossil fuels and creates a shift of attention within the energy market.

For the offshore market an increase in renewable projects can be observed. The technological innovations are continuously decreasing the costs of renewable projects. Especially offshore wind is getting more cost effective, and is receiving a lot of attention. However, with the growth of the sector, new challenges arise. Water depths are increasing, soil parameters worsening and greater distances from shore need to be overcome. Meanwhile, prices of installation are under pressure. This drives the market towards larger and more innovative installation vessels. Originally wind turbines would be installed with the use of jack-up vessels. Although these vessels can be very useful, they come with a few downsides. Namely, they require proper soil conditions to support the vessel. Furthermore, the legs have a limited water depth on which they can operate. In addition, the legs use up potential deck storage space. A trend can be observed towards monohull craning vessels which combine a large crane with a large storage space on deck. A few examples are: Aegir from Heerema (see figure 1.1), Oleg Strashnov from Seaway Heavy Lifting (see figure 1.2) and Bokalift 1 from Boskalis (see figure 1.3). The ships are designed to be able to operate on both wind energy projects and oil and gas (O&G) projects. The vessels are able to install while floating by relying on their dynamic positioning (DP) system. However, heavy lifting while floating comes with its disadvantages. With the heavy lifting vessels and corresponding cranes becoming larger, small displacements can result in significant motions of the load. The motions result in forces and will influence the work-ability of the vessel. The work-ability of a vessel is influenced by a limiting factor. This can be the exceedance of a certain deviation, or exceendance of forces in the crane, crane lines or of the lifted object. The costly daily rates of vessel, make work-ability an important feature. To stay competitive these costs need to be reduced. Hence, this is why increasing the work-ability of the installation vessels is desirable.

1.2. Boskalis

Boskalis operates around the world and is a leading player in the field of dredging, offshore energy and maritime services. They provide innovative and competitive all-round solutions for multiple offshore sectors. From the start of Boskalis offshore in 2010, Boskalis has increased its active sectors within the offshore world. Their renewable capabilities have been growing rapidly (see figure 1.4). Investing heavily into innovative and smart ways of improving their solutions for their clients. With eye on the mega-trend towards renewables,

Figure 1.1: The Aegir from Heerema

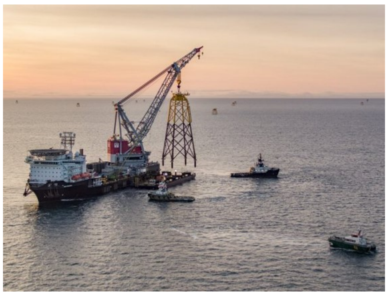


Figure 1.2: The Oleg Strashnov from Seaway heavy lifting

Figure 1.3: Bokalift 1 from Boskalis

Boskalis is looking at ways to further improve their capabilities of transport and installation (T&I). The subject of this report belongs to one of these possible improvements, namely, improving the work-ability of a vessel by tugger control. Boskalis does not yet own lifting vessels which contain tugger damping systems. However, after positive experiences on other vessels it is considered for implemented on their own fleet. Tugger damping has the potential to shorten installation time and consequently costs. This report will help in creating a better understanding on the potential of different control systems, and its possible applications on different lifting loads. Helping Boskalis in finding innovative ways of improving their service and reinforcing its position in the offshore heavy lifting market.

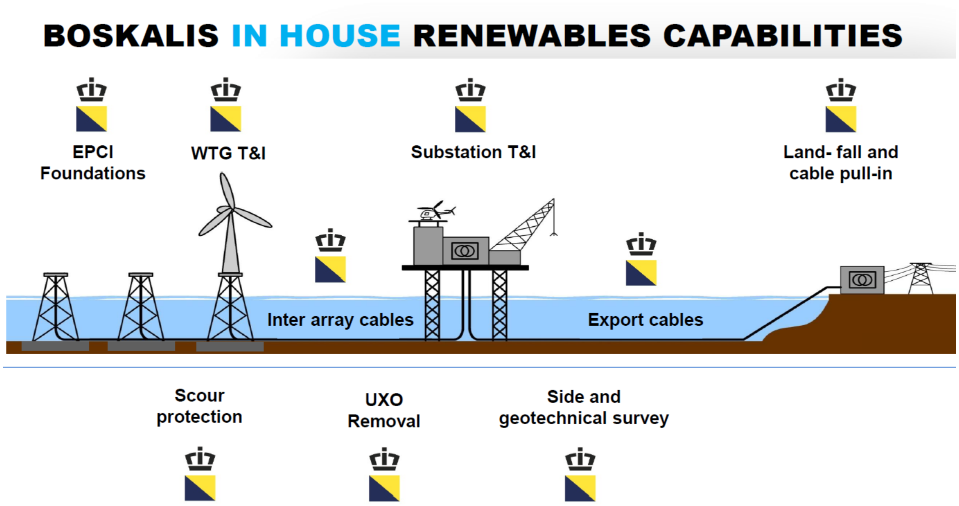


Figure 1.4: In house renewables capabilities of Boskalis

1.3. Tugger system

A Tugger system consists of one or multiple tagline(s) connected to winches. The system is occasionally used to counter-act the swinging motions of the load, or crane block, and to mitigate roll motions of the lifting vessel. The more or less horizontal lines are to be connected between the load and the crane. On the crane end, the line is attached to a winch. The winch applies force on the cable, together with a pretension on the line to avoid slack. However, by adjusting the force output by the winch the tugger system can have different effects on the lifting operation. Traditionally, the winch would know two modes, namely:

- Fixed length
- Constant tension

[11]

For "fixed length" the winch will be put on brake after applying pretension to the line. The line will have a "fixed length". This way the line acts as a horizontal spring with the stiffness equal to that of the line. By adding stiffness, the natural frequency of the pendulum motion is lowered (the pendulum and double pendulum are illustrated in figure 1.5). Often leading to less excitation by the environmental loads. "Fixed length" only works for smaller loads and movements. As with higher loads and the possibility of slack, the lines have high chance of breaking. Breaking lines offshore is a risks that many do not want to take, for this reason "Constant tension" is often preferred above "Constant length".

With "constant tension" the winches try to keep constant tension in the lines. In theory this means a constant force is pulling on the load. Which has little effect on the natural period of the lifting system, and contributes very little to the dynamics and dampening of the system. Despite this, "constant tension" in practice does have a dampening effect on the load. The winches used do not function perfectly and will always have errors. This results in the applied forces fluctuating around the mean. This fluctuation leads to dissipation of energy within the system and thus acts as a damper.

To summarize, tugger damping is being used within the offshore industry. Often to reduce the swinging motion of the suspended load. If used correctly, it is possible to use the velocity driven winches to apply damping during a lift. Though this damping effect depends on a lot of parameters. The control system of the winches, the specifics and capabilities of the used winches, the amount of lines used, location of the attachments, and stiffness of the lines are a few examples. Each parameter influence the system on their own way, and because of this it is hard to state something is the "best way" of applying tugger damping.

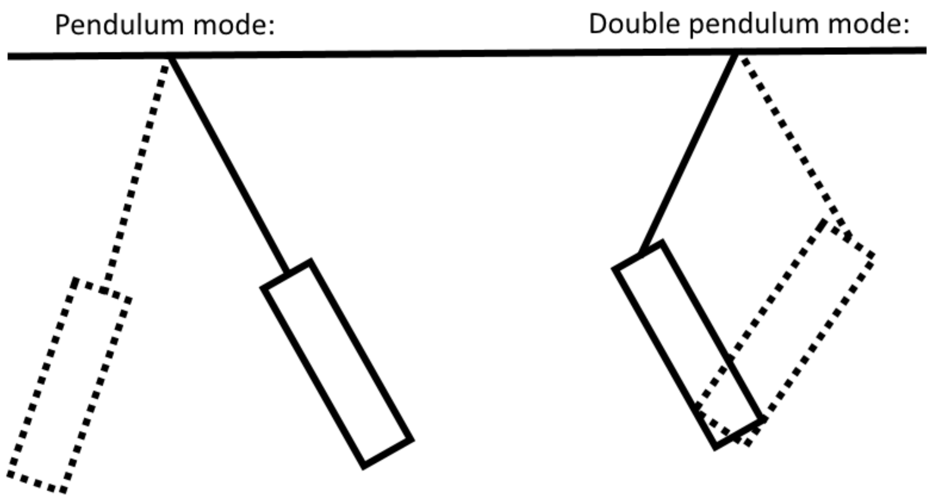


Figure 1.5: The pendulum and pendulum modes illustrated

1.4. Problem description

The changing offshore market drove the design and production of new high end monohull lifting vessels like the Bokalift 1, Aegir and Oleg Strashnov. However, these vessels experience more motions than the traditional large semi-submersible crane vessels. Especially during lifting, large motions could arise. The largest motions are often the result of excitation of certain modes of the vessel and lifting configuration. For the lifting configuration both excitation of the pendulum and double pendulum mode result in resonance. For the vessel excitation of the pitch, heave, and roll mode result in larger crane tip motions. Especially the roll of the vessel has the largest contribution on the crane tip motions, due to its large arm. Resonance creates energy build up, which results in large deflections of the load and vessel, and consequently forces in the crane, load, and vessel. [9]. Consequently, the deflection and forces lower the work-ability of the lifting vessel. Work-ability of the vessels is of significant importance in the demanding offshore sector. Especially for the installation of wind farms, as higher work-ability can give you an advantage towards competitors. A possible solution to minimize the resonance and thus the displacement of the load is by the use of tugger control. If applied correctly the tugger system can be used to damp the system and thus increase its work-ability [11]. However there is still a lot uncertain with respect to tugger damping and how to implement it most efficiently.

1.5. Objective

The objective of the thesis is: To investigate the potential damping effect of a tugger system, and to optimize a tugger damping system for the Bokalift 1 during a jacket lifting operation The optimization meets a few criterion. Namely, only two load tuggers will be added besides the already existing boom tuggers. Additionally, the damping system is optimized for a "worst case scenario" load case. Within the report the main focus will be on the control system of the winches. Finding the effect of different control systems on the loading configuration, and subsequently being able to explain the different behavior depending on the control system. With the aim on increasing work-ability of the vessel by damping. As damping influences the dynamic behaviour of the lifting system, the tugger system will be used to minimize the motions of the dynamic model.

Throughout the report the following sub-questions are covered. These aid in answering the main problem and help in creating a more complete answer.

- What is the effect on motion and dampening of the jacket for different winch control systems?
- What are important properties concerning the control system?
- Does the PID controller damp the lifting configuration more effective than P controlled winches. Is a PID controlled winch more beneficial than P controlled winch?
- How much can a tugger damping system decrease the forces on the crane top

Throughout the report these sub-question are treated.

1.6. Approach

The report will consists of 7 chapters. Starting with chapter 2 in which the lifting configuration will be treated. This includes, the used vessel,the load,the winches, and the used lifting case. Treating their properties, specifications and limitations. Where after the Orcaflex model is introduced which is used for the analysis of tugger control in chapter 4 and chapter 5. It also elaborates required data and parameters and how they are gained. Besides an Orcaflex model, a 2D matlab model is used for analyzing the effects of different control systems. The creation of this model is treated in chapter 3. Firstly, the 2D 5-Degree of freedom [DOF] mass-spring-damper system is treated. Where after the equation of motion are found for the system using the program maple. Followed by a short description of Matlab and how the model is made up and created. The chapter ends with validation of the matlab model, which is done by comparing the results of a model analysis of the matlab model with the results of a Orcaflex model. Chapter 4 starts with elaborating the concept of a control system, and the goals and limitations set for the used control system. Where after the different P-damping function used in the control system are introduced and explained. Next, a PID controller is shortly elaborated including the different terms and their effects. Thereafter the different control systems will be tested using the 2D matlab model. Starting with elaborating on what tests are done, how, and with what limitations. There after the PID is tuned, after which the control systems are tested for their reduction on roll motion of the jacket, and their ability to damp out the lifting system with a initial displacement. The second half of chapter 4 is dedicated on testing P-control systems in orcaflex. Firstly the normative environmental loading case is treated, together with the limiting factors of the lift. Followed by a explanation on the existing winch controller in orcaflex. The different terms and their effect on the system. Three variants of the P-control system are tested. Finally, a optimized P-controller is found. Chapter 5 treats the creation of a PID controller within the Orcaflex model. The effect of the controller is compared to the optimized P-controller from chapter 4 and a model without tugger control. Lastly the report is finalized with a conclusion and recommendations in chapter 6. A schematic overview of the report is shown in figure 1.6.

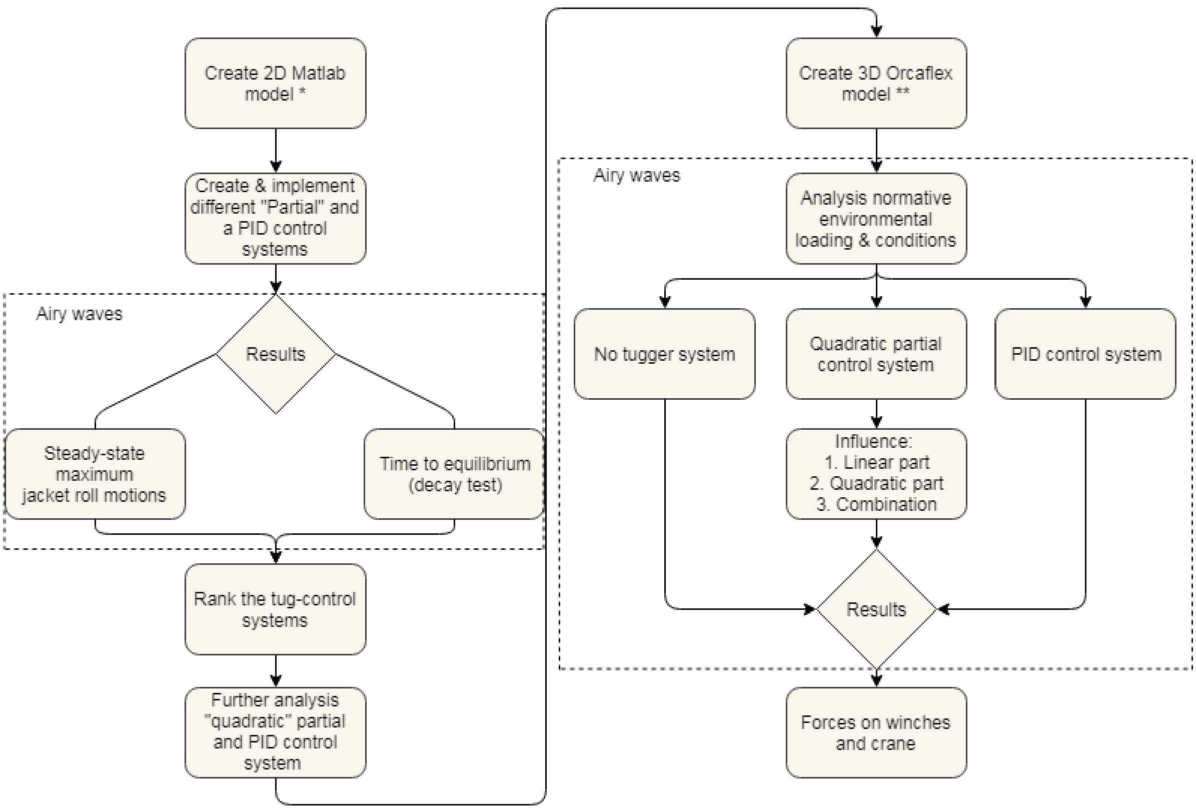


Figure 1.6: * for more information on the creation of the matlab model see figure 3.1 and 3.2
** for more information on the creation of the Orcaflex model see table 2.6

2 Lifting concept

Applying tugger damping on a lifting system should have roughly the same results for different vessel, loads and lifting operations. However, for analysis of the tugger system it is important to make well considered decisions on the right equipment, circumstances and load cases. In section 2.1 the vessel is elaborated further. Section 2.2 treats the different loads, where after the load cases is elaborated in section 2.3. Lastly, the created Orcaflex model is treated in section 2.4.

2.1. Bokalift

In 2016 Boskalis converted the Semi-Submersible Heavy Transport Vessel 'Finesse' into a heavy lifting vessel, named Bokalift 1 (see figure 2.1). Boskalis made the decision to upgrade the 'Finesse', with an eye on the shifting offshore market. The ship is outputted with an big, high-end crane. What distinguishes the vessel is the combination of heavy crane and large available deck space, together with the fact that the vessel used to be a heavy transport vessel, which makes it suitable for transport and installation of heavy components. The crane, build and installed by Huisman, has a lifting capacity of up to 3,000 MT for jackets and monopiles. The maximum crane lifting height for the main block is up to 90m. The converted vessel now features 6,300m^2 deck space with a 15,000 t maximum deck load. During the conversion, the ship was also equipped with DP-2 capabilities and cutting out anchor deployment, improving (de-) installation cycle times. The vessel supports accommodation for 149 people, and features a helicopter deck for offshore transfer. A few important properties are presented in table 2.1 [4].

The crane of the Bokalift 1 is positioned roughly halfway the large deck. The crane can rotate around its slewing platform, and thus has reach over the whole deck. The lifting capacity of 3,000 mT is large, however vessels like the Thialf, Saipem or Sleipnir have a significant larger lifting capacity. These vessels are equipped with tub mounted cranes, which take up a large amount of deck space and have a low operational handling speed. Boskalis chose to equip the bokalift with an offshore mast crane, which is better suited for offshore wind project because of its faster handling and lower footprint on deck. An overview of the crane including the important components is given in figure 2.2. The crane has three different hoists, namely, the whip hoist, auxiliary hoist and the main hoist. The specifics of each hoist is shown in table 2.2 [3]. The crane also features tugger lines. Four load tuggers are present on the slew platform of the crane. The tuggers have a safe working load of 45 mt. Currently the winches can operate in two modes, namely, speed mode and constant

Parameter	**Value**	**Unit**
Length overall	216.7	[m]
Breadth (moulded)	43.0	[m]
Depth (moulded)	13.0	[m]
Operational draft while lifting	8.35	[m]
Deadweight	50,000	[mt]
Deck space [L x B]	165 x 43	[m]

Table 2.1: Particulars Bokalift 1

Figure 2.1: The Bokalift 1, during the east Anglia project

	whip hoist	*auxiliary hoist*	*main hoist*
max. lifting height [m]	-	99	90
max. radius [m]	92	50	28
loading capacity [mt]	200	1200	3000

Table 2.2: Crane specifics

tension mode. Note that although for this report the Bokalift is used for analysis, the tugger system should be applicable for multiple vessel.

2.2. Load

Throughout the report, two different loads are used. Namely, a 4 legged jacket and a suction bucket jacket. The suction bucket jacket is used for the more elaborate Orcaflex model. The reason for lifting a SBJ is due to the large surface area of the buckets. Which result into large environmental loading of the waves and thus will act as a worst case scenario for lifting. The 4 legged jacket is used for the Matlab model and the validation of the 2D matlab model. The reason for not lifting a SBJ is due to the more complicated loading and behavior in the splash zone, making it harder to model without the help of extensive software. The 4 legged jackets is relatively basic and more simple to model. In the report the choice is made to analyse the lift of a jacket instead of a monopile. This choice is made for two reasons. Firstly, tugger damping is better applicable and more useful for installing jackets compared to monopiles. This has to do with the current way of installing monopiles. As monopiles are often installed guided by a gripper. The gripper compensates the monopile in sway and surge direction. Obviously, the gripper helps with taking resonance out of the system, making tugger damping needless. Secondly, jackets are becoming increasingly popular for windfarm projects. This, among other things, is due to increasing water depths for wind-farms. As jackets are more stiff and require less steel for production than monopiles. Making them more cost effective for greater water depths. Over the years the O&G sector has gained expertise on the use of jackets. The rapidly growing wind energy market pushes the Jackets towards different sizes and types. They can be founded traditionally by piles, or with the use of suction buckets.

2.2.1. 4-leg jacket

The 2D model uses a 4-leg jacket as lifted object (see figure 2.4). The dimensions of the jacket are based on the heaviest jacket used in the formosa project. The particulars of the jacket are shown in table 2.3. Note that all the jacket member diameters are considered constant. The 4-leg jacket is the most common jacket used

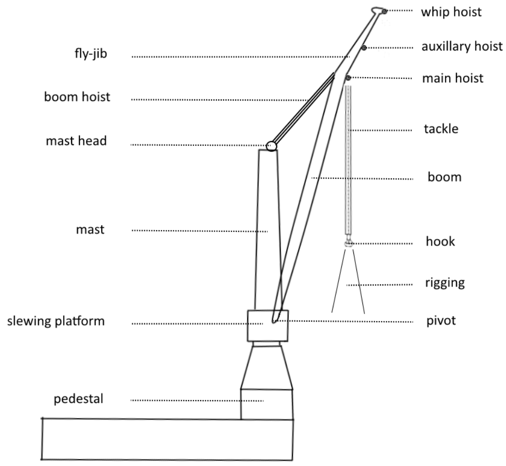


Figure 2.2: Crane overview

for offshore projects.

2.2.2. Suction bucket

Suction bucket jackets (SBJ) uses a relatively new way of installing jackets. They are practical for the same water depths as regular jackets. During installation a SBJ starts with penetrating the soil by own weight. By applying suction on top of the bucket creating an vacuum inside the bucket. The bucket will bury itself till the right depth. No pile driving is required, and it can be installed in place without any extra steps. This greatly reduces the noise produced and brings down installation times. Additionally, the decommissioning of the jackets is relatively easy and fast, as it only requires the reversed process of installation. However, SBJ require the right conditions and especially the right soil specifics, in order to be applicable. The dimensions of the SBJ are based on the largest SBJ used in the Aberdeen offshore wind farm project, see figure 2.6. The dimensions of the bucket itself are shown in table 2.5. These values are used in the Orcaflex calculation done in chapter 5.

2.3. Lifting case

Within a lifting operation, there are multiple load cases to be considered, as they represent important scenarios during the whole lifting operation with each its own challenges. For this report however, only one load case is treated [5].

The load case treated here is used for both the matlab model as well as the Orcaflex model. However note that both models do have different jackets. This results in slightly different arrangements, see table 2.7.

	Matlab model	Orcaflex model	Unit
Crane outreach	31	28	[m]
Height crane tip wrt MWL	142	126	[m]
Main hoist tackle length	52.3	36.4	[m]
Mass of load	1665	1666	[t]
Mass rigging + hoist block	n/a	100	[t]

Table 2.7: General parameters lifting configuration for matlab model & Orcaflex model

The treated load case is one in which the jacket is hanging outwards and inside the splash zone (see figure 2.5). This load case represents the situation in which the jacket is being installed. The crane is positioned most outward to have the furthest reach and thus the largest distance from the ship. Here the jacket is located the furthest away from the ship COG in transverse (y-)direction of the ship. This results in the jacket experiencing great coupling with the roll motion, leading to large movements and loads. This is why often this is the

Item	Values	Unit
Total Jacket Mass	1665	[ton]
Stab pin height (h_3)	8	[m]
Leg Height (h_2)	72	[m]
TP Height (h_1)	8	[m]
Top Leg spacing	15	[m]
Bottom Leg spacing	26	[m]
Main Leg Diameter	1.9	[m]
Brace Leg Diameter	0.9	[m]

Table 2.3: Jacket dimensions

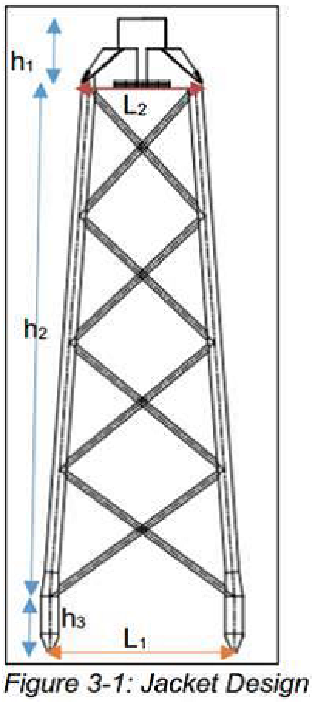


Table 2.4: 4-legged jacket

Item	Values	Unit
Total Jacket Mass	1666	[ton]
Total foundation height	77	[m]
Bucket height	13	[m]
Bucket diameter	10.5	[m]

Table 2.5: SBJ dimensions

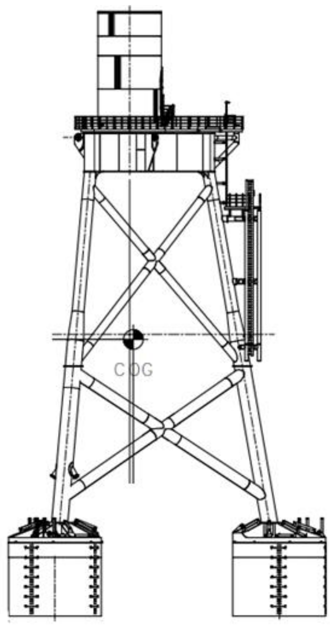


Table 2.6: Suction Bucket Jacket

limiting load case. This is especially the case for suction buckets jackets due to their large exposed surface of the buckets.

The tugger lines have several options for their attachments to both the vessel and jacket. Obviously the location of their attachment has impact on the behavior of the system. In order to have better control over the jacket in both [x] and [y] directions (see figure 2.5) it is desirable to have tugger lines facing both directions. The bokalift 1 already has 4 tugger lines available at the boom. These are good for limiting movement in extension of the crane (y-direction). To be able to counteract movements of the jacket perpendicular to the crane (x-direction) two winches will be placed on the deck. These are placed on both far sides of the bow to maximize their control in x-direction.

The boom tuggers are attached to the gripper of the crane. As the boom tuggers are located relatively high in the crane this also means the lines stay reasonable horizontal and smaller lengths. Applying damping at this location works especially good for damping the double pendulum motion, as this is the location where the largest movements taker place. For damping the single pendulum motion an obvious location would be the lower side of the jacket. The possibility of connect the deck winches to the lower side of the jacket has been analysed, however there can be concluded that connecting to this location had little effect on damping the system. This had to do with the behavior of the SBJ in water. Due to the shape of SBJ and the added mass of the water surrounding the buckets, the system has a low centre of rotation (almost near the water line). Which means that the arm between the force from the tugger line and centre of rotation is relatively small, resulting in lower leverage forces. Also, as the jacket is partially submerged the water itself already provides damping on the lower side of the jacket. Leading to smaller velocities near the bottom. Additionally this means that applying force here will result in smaller responses due to the presence of water. As the water will also damp a portion of the applied forcing. For these reasons, the deck winches are also connected to the gripper of the loading configuration. In addition it has the benefit of not having to (de)install the tugger lines onto the jacket, which requires time.

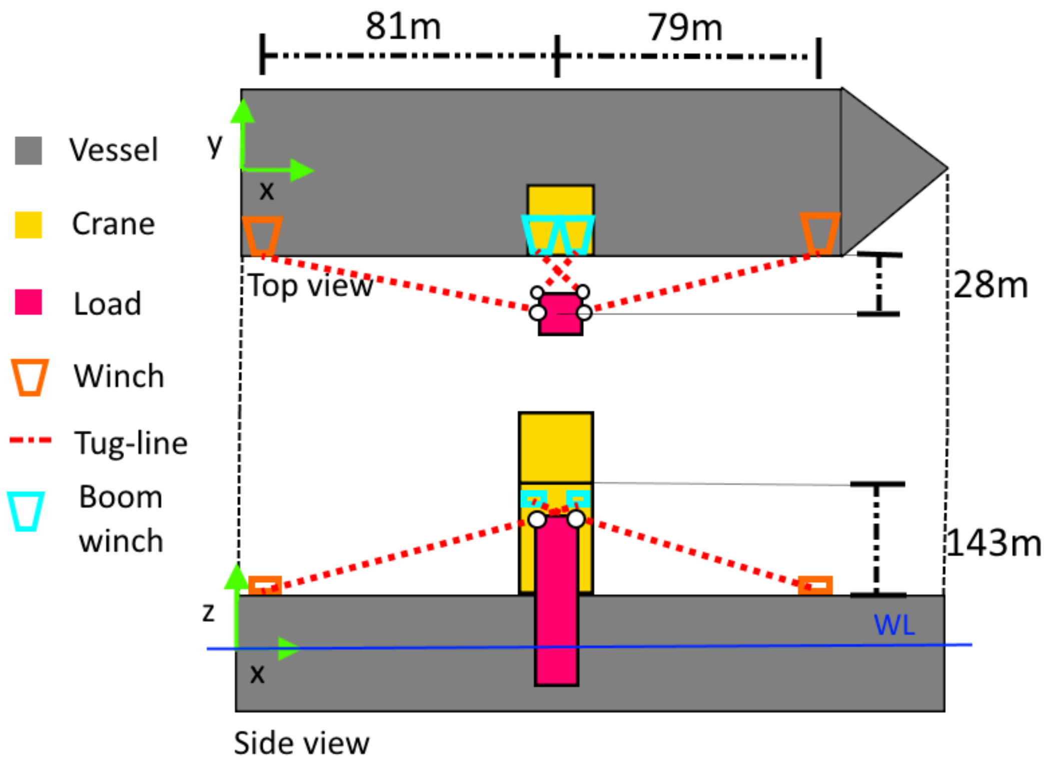


Figure 2.3: Lifting configuration Orcaflex model

2.4. Orcaflex model

Orcaflex is a marine dynamic program used for dynamic analysis of offshore marine systems. Orcaflex enables it user to create different objects, shapes, vessels and components. Both the jacket and the vessel are modelled as a 6D buoy in orcaflex. In order to represent the right project Orcaflex requires input for both 6D buoys. With the input Orcaflex is able to evaluate the responses of the model correctly. Note that for the validation Orcaflex model only a static analysis is required. Hence, the properties needed are:

- Centre of gravity (Vessel & load)
- Centre of buoyancy (Vessel)
- Mass (Vessel & load)
- Moment of inneratia (Vessel & load)
- Hydrostatic stiffness (Vessel)
- Frequency dependent damping (Vessel)
- Response amplitude operators (RAOs)

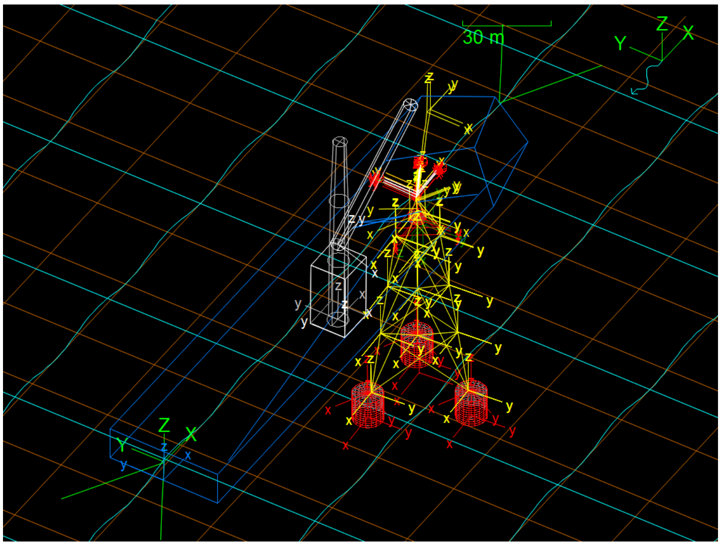


Figure 2.4: Lifting configuration

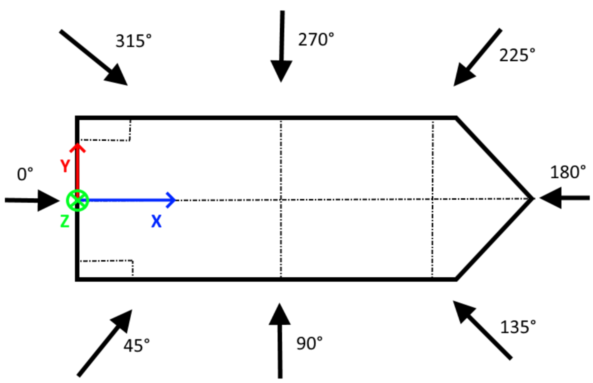


Figure 2.5: Directions environmental loading

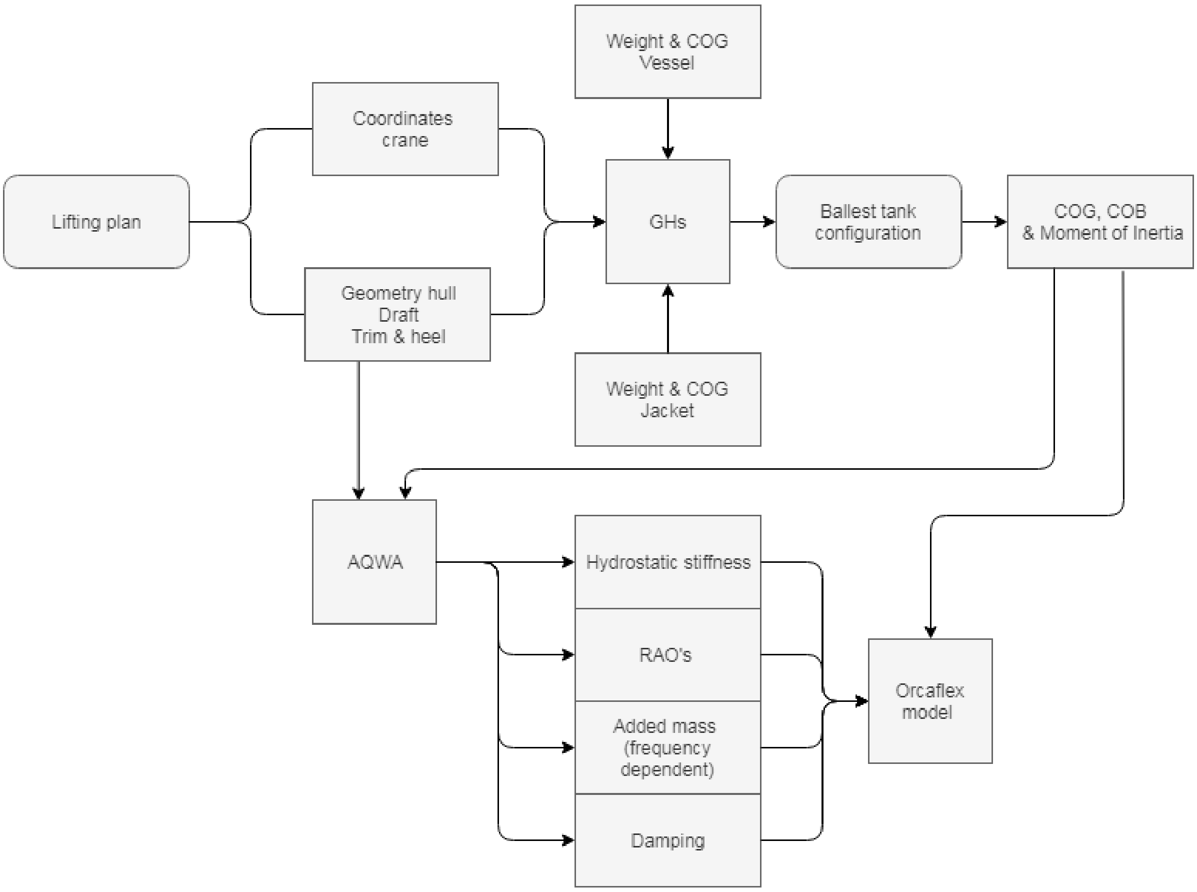


Figure 2.6: Model setup flow diagram

2.4.1. Global Hydro Statics

GHs is a tool used to analyse and calculate hydrostatic proprieties of models of vessels. For this report it is used in order to calculate the configuration of the ballast tanks [5]. So that for a static analysis the model has zero heel, zero trim and the requested draft. In order to calculate this, the program requires the weight and corresponding COG of all parts. For our model these include the vessel and the jacket. Additionally it requires the orientation of the crane. As the COG of the vessel itself depends on it besides being the location where the forces from the jacket engage. Subsequently, the ballast tanks are filled in order to create zero heel,zero trim and the requested trim. By acquiring these values, the new COG of the vessel can be calculated using "steiner's law" (see equation 2.1). Adding up every I_{xx}^{Global} leads to the total moment of inertia for that direction.

$$\begin{aligned} I_{xx}^{Global} &= I_{xx}^{Local} + I_{xx}^{Steiner} \\ I_{xx}^{Steiner} &= m \cdot (\triangle x)^2 \end{aligned} \tag{2.1}$$

2.4.2. Vessel and Response Amplitude Operator

Frequency-dependent Response Amplitude Operators (RAO) are used in the Orcaflex model to calculate the forces or displacement due to wave excitation. Within Orcaflex it is possible to use two kinds of RAOs. Namely: Force RAOs, which per frequency gives the ratio between wave-height and force per DOF. Or Displacement RAOs, which give the ratio between the waveheight and response per DOF. For the used Orcaflex model only the Force RAOs are useful, due to the jacket and vessel coupled being systems. As displacement RAOs could lead to errors especially around resonance frequencies [9].

The force RAOs are calculated with the hydrodynamic diffraction software ANSYS AQWA [5]. The program

	Value	Unit
Weight	55.7E3	[mt]
COG X	109.56	[m]
COG Y	1.405	[m]
COG Z	10.03	[m]
I_{xx}	16.4E6	[mT*m^2]
I_{yy}	169.3E6	[mT*m^2]
I_{zz}	169.6E6	[mT*m^2]
Heave Hydrostatic Stiffness C_{33}	79.30E3	[kN]
Roll Hydrostatic Stiffness C_{44}	5.720E6	[kN*m/rad]
Pitch Hydrostatic Stiffness C_{55}	224.0E6	[kN*m/rad]

Table 2.8: Properties of Orcaflex Bokalift 1 model

is applied for hydrodynamic analysis for all types of offshore and marine structures. The software is based on linear potential theory in frequency domain. According to this theory, "the potential of a floating body is a superposition of the potentials due to the undisturbed incoming wave Φ_w, the potential due to the diffraction of the undisturbed incoming wave on the fixed body Φ_d and the radiation potentials due to the six body motions Φ_j" (see equation 2.2).[7] However, linear potential theory neglects viscous effect. As a consequence it is not possible to calculate the viscous roll damping. This is calculated using the Tanaka methodology. The remaining properties are presented in table 2.2. Both the damping coefficients and added mass are not shown in table 2.2 as they change per loading frequency.

$$\Phi = \sum_{j=1}^{6} \Phi_j + \Phi_w + \Phi_d \tag{2.2}$$

The input for the Orcaflex model is presented in table 2.8

The horizontal mooring cables only exist in the model to constrain the vessel horizontally. In reality the vessel uses the DP system to keep its location. The lines are placed such that they do not contribute to the roll period of the system, see figure 2.7. The lines are mass-less and placed horizontally in order to not affect the system vertically. Additionally their properties are chosen to have the natural periods of the sway, yaw and surge motion to be outside of the wave excitation periods analysed in this report. The properties of the horizontal constrain is shown in table 2.9.

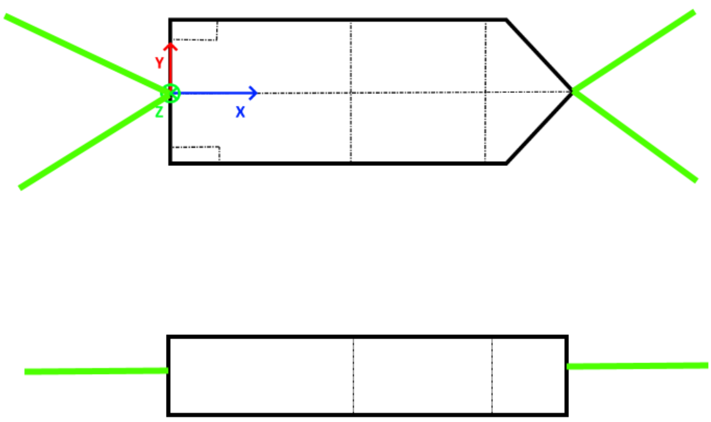


Figure 2.7: Placement mooring lines Orcaflex model

	Value	**Unit**
E	1E4	[kN]
L	39	[m]
K	256.4	[kN/m]
Nat Period Surge	81	[s]
Nat Period Sway	106	[s]
Nat Period Yaw	50.95	[s]

Table 2.9: Properties horizontal mooring lines

	Amount of tuggers	**Max working load**	**Unit**
Boom-tugger	4	45	[mT]
Deck-tugger	2	219.4	[mT]

Table 2.10: Tugger properties

2.4.3. Lifting configuration

The lifting configuration consists of the hook, the jacket, JLT and the winches [5]. Within the model the individual components are modeled as 6D buoys, except for the legs a braces of the jacket. They are modelled as individual lines which all move with respect to the same reference point that represents the behavior of the jacket as a whole. Additionally, each suction bucket is individually modelled as a 6D buoy. For the considered loading case all suction bucket are fully submerged in the water. Due to the shape of the bucket, the added mass is large relative to its own weight. As the suction bucket is considered fully submerged, no air is considered inside the bucket.

The model contains the four boom winches already present on the bokalift. Additionally the two deck winches are added, as mentioned earlier they are positioned on the far sides of the deck on the starboard edge. The maximum work load of 2000 mT is set for both deck winches (an overview of the winch properties is shown in table 2.10).

3

2D lifting model

In order to analyse the effect of different winch control systems a simplified 2D lifting model is created in matlab. By using a simplified model, a fast but accurate analysis can be made on the order of efficiency of different control systems. The models goal is to achieve a better basic understanding of different control systems. Where after system are analysed in a more detailed and realistic 3D system which is created the program orcaflex. This chapter treats the creation (matlab) and validation (orcaflex) of the 2D matlab model. Section 3.1 starts with an overview of the model creation. Where after in section 3.2 the mass-spring-damper-system used in the model is illustrated and elaborated. Section 3.3 treats the calculation of the equation of motions of the system. Which is done with help of the program Maple. The EOM are used as input for the matlab model which is treated in chapter 3.4. With the 2D model finished the model is validated by comparing it with the output of an 3D Orcaflex model in section 3.6.

3.1. model setup

This chapter treats the creation of a 2D model. This section servers as an overview of how this model is build up. As mentioned earlier, the 2D matlab model will serve as a tool for easy analysis on the effect of tugger damping and multiple control strategies. The creation of the model consists of multiple parts. Firstly, in order to represent reality, the right input for the model is generated. This is done with the use of multiple programs (figure 3.1 illustrates the steps required). Additionally, an Orcaflex model is created to validate the gained values, parameters and the created matlab model. The steps required for creation of this model is illustrated in figure 3.2. Firstly, the lifting setup is to be defined. When the vessel and lifting configuration are defined calculations can start. The combined COG of the system is approached with the help of the program GHs. The program requires the lifting setup and outputs its COG coordinates. The values are checked with help of the Orcaflex model. This is done by checking the static response of the ship in Orcaflex when using the gained information. When the model stays perfectly horizontal it means it is the correct value for that lifting configuration. All hydrodynamic input properties come from Aqwa. With all required input obtained, the matlab model is created. For final validation of the created matlab model, a model analysis is done. The outcome of the analysis is compared with the output of the model analysis of the Orcaflex model.

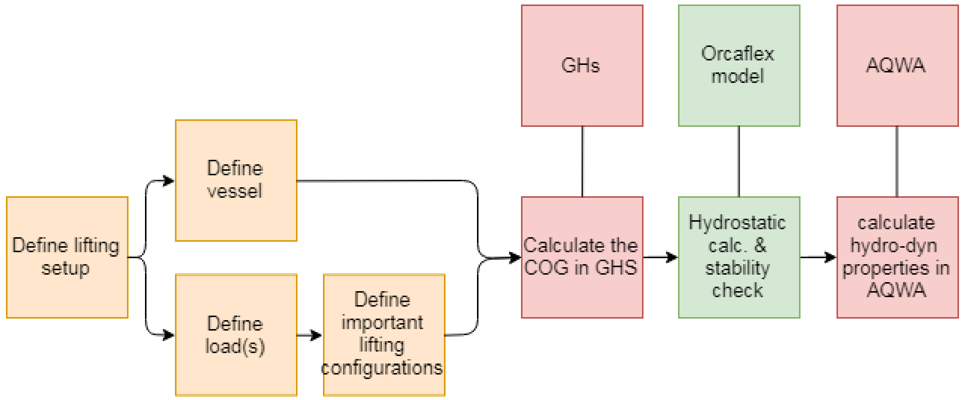


Figure 3.1: Model setup overview 1

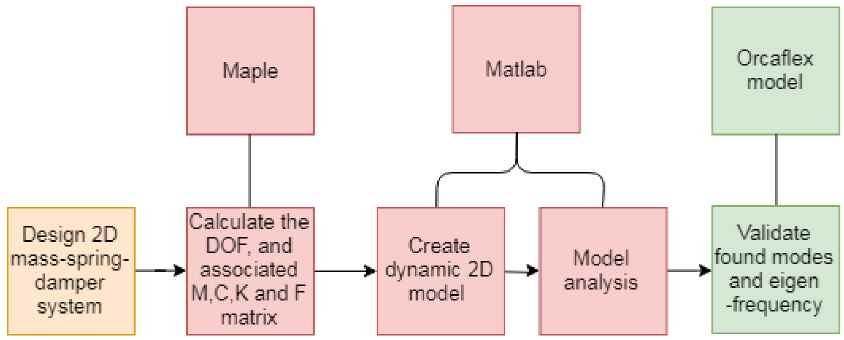


Figure 3.2: Model setup overview 2

3.2. mass-spring-damper-system

To model the behavior of the lifting operation the ship and jacket will be modelled as a mass spring damper system, see figure 3.3. The system is has 5 DOF, namely: θ_k, θ_m, θ_s, x_s and y_s. The mass indicated with an uppercase "M" represents the vessel, the lowercase "m" represents the jacket. The vessel has both horizontal and vertical dampers and springs. As well as a rotational spring and damper. Some dampers and/or springs may seem negligible or inapplicable. However these have been added with the purpose of keeping the system in place and having more adjustable options may the need occur to apply them. The latter one is especially the case with the damper, spring and variable force applied on the load "m". Depending on the behaviour of the winch control system, these can individual or combined be applied on the jacket to represent the tugger damping system. Also, three possible forces apply on the vessel. A horizontal (F_{sh}), vertical (F_{sv}) and rotational force (M_s) work on the vessel. The forces represent the environmental loads on the vessel. The environmental forces on the load are represented by F_m. Gravity is included in the model. Note that horizontal spring $K_s h$ does not exist in reality, but only has a purpose to keep the vessel in place.

Tugger control is modelled as two pair of opposing forces, this is illustrated in figure 3.4. The winch cables are attached to the right upper corner of the vessel and on the hinge between the jacket and tackle. The tension is factorized into a horizontal force T_{tugh} and a vertical force T_{tugv} depending on the angle of the tugger line $theta_{tug}$.

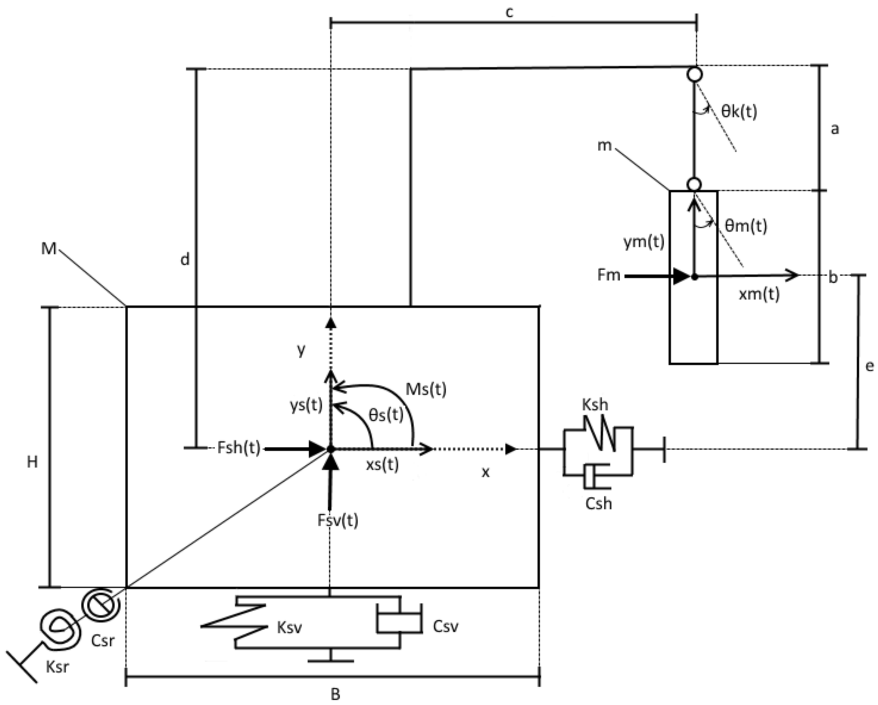


Figure 3.3: Mass-Spring-Damper model

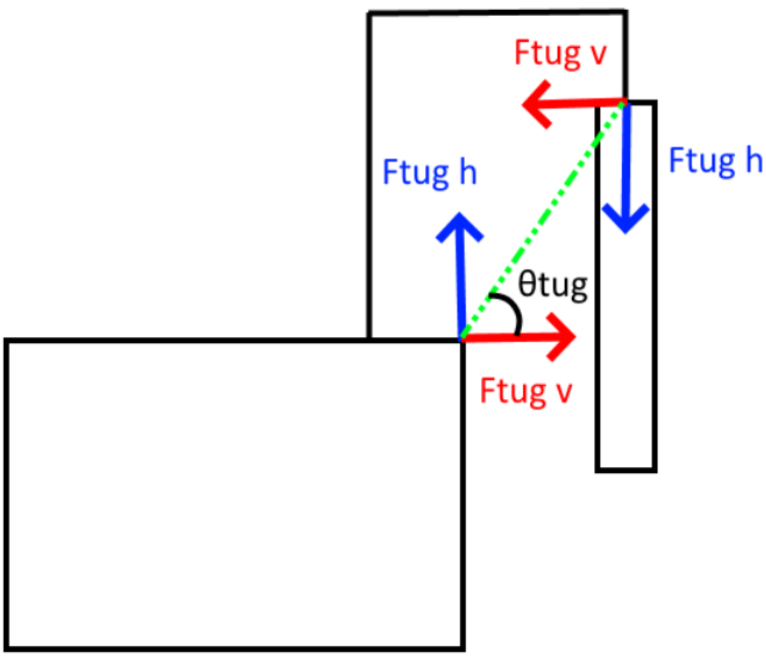


Figure 3.4: Tugger forces

Modelling the lifting system as a 2D Mass-Spring-Damper system requires making assumptions. The first and most obvious assumption is the simplification of the system by assuming it a mass-spring-damper system. Assuming all elements follow linear constitutive laws. With this assumption the buoyancy is modeled as a mechanical spring, the hydrodynamic damping as mechanical dampers and the vessel and jacket as two squares with their COG in the exact centre, with the COB of the system coinciding with the COG of the vessel. Secondly the model is taken as a 2D model where as in reality the lifting configuration is a 3D problem.

The goal of the model is better understanding and to enable faster analysis and comparison of different tugger damping systems. Thus for the analysis of the effect of different winch control systems a 2D model will suffice. This is with the following assumption: The order of damping efficiency of the system will approximately be the same for 2D as for 3D. Thirdly, the tugger damping system is modeled as a variable force. All the environmental forces are modelled as a single point load. Fourthly, the tackle of the crane is modelled as a pendulum (rod with on both ends hinges) and both the crane and tackle are considered to have zero mass and infinite stiffness. Where as the tuggerline has zero mass, infinite stiffness and zero bending stiffness. Lastly the tugger winches have unlimited pulling speed, and thus can always avoid slack of the cables.

3.3. Maple

Maple is an advanced math software often used by mathematicians, engineers and scientist. It has an focus on symbolic computing and it is often used in education. The fact the it can calculate and compute only using symbols is one of the large advances of maple. For this reason it is used in computing the EOM of the system.

3.3.1. Program output

The EOM are derived in maple using the Lagrange formalism. The EOM are in the form shown in figure 3.1. With the m being the mass matrix, μ the added mass, **C** the damping-matrix, **K** the stiffness matrix and **F** the force vector. As input the lagrange requires the potential energy, kinetic energy and work done on the system. With the formula the equation of motion can be derived for all five degrees of freedom. By letting maple isolate all terms that are related to second order time derivatives of each degree of freedom, the Mass-matrix can be derived. The left over matrix term represents a combined matrix consisting of the K-matrix, C-matrix and F-matrix. To derive the individual matrices the same method is used as for the mass-matrix. Obviously all the term related to first order time derivatives belong in the C-matrix. Leaving a combined matrix consisting of both the F and k terms. However in order to rightly divide the F-vector from the K-matrix, terms higher than second order are left out. Where after a linear K-matrix and F-vector are determined.Note that the linear K-matrix and F-vector are only used to compute the eigenfrequency.

An overview of the process is illustrated in figure 3.5

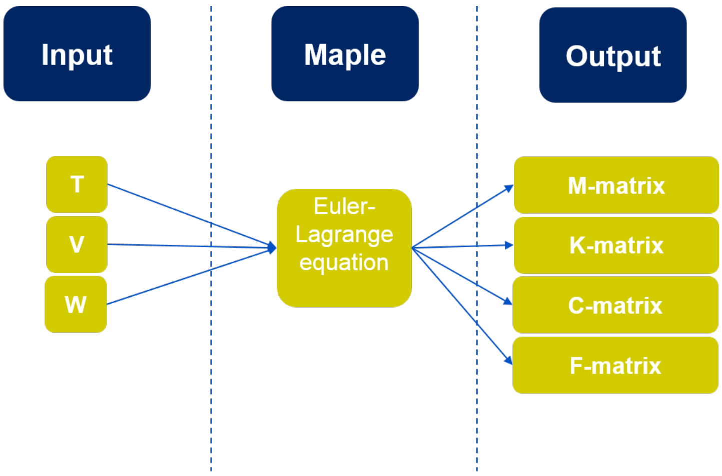


Figure 3.5: Maple overview

Using the lagrange formalism the EOM is derived using the generic Euler-Lagrange, see equation 3.2. As the system contains viscous dampers the lagrange equation contains a D term. Also known as the Rayleigh dissipation function. The L term is the "so-called" lagrangian. It is calculated using the formula shown in 5.1. Where the T and V are the kinetic and potential energy of the system. The W represents the work done by external forces on the system.

$$(m+\mu)\cdot\ddot{x}+c\cdot\dot{x}+k\cdot x=F \tag{3.1}$$

$$\frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{x}}\right)+\frac{\partial D}{\partial \dot{x}}-\frac{\partial L}{\partial x} \tag{3.2}$$

$$L=T-V-W \tag{3.3}$$

For determining the EOM, a small angle approximation is done. This means, trigonometric functions are simplified to second order approximations. As the name suggests, this approximation has a small error for small angles. For large movements this model becomes less accurate. However, this approximation satisfies for the modelling done in this report. As the relative error only exceeds 1 percent at angles larger than 14 degrees for the sinus approximation, and 38 degrees for the cosines approximation. The maple files are shown in appendix A

3.4. 2D Matlab model

Matlab is used for creating the 2D model.

3.4.1. Model output

Matlab is used for simulating the behavior of the 5 DOF system. By calculating the velocity and acceleration numerically for small time steps (T_n) the movement of the system is approached. Firstly, all parameters used are defined. Among which values like model sizes, timesteps, stiffness values, damping values, starting positions and velocities of the jacket loading configuration.

These values are loaded into another script, called the ODE solver, see figure 3.6. The first thing the ODE solver does is calculating the external forcing for the corresponding time step T_n. These are the forces that represent the environmental loading. Where after the M matrix and a combined matrix consisting of the "F,C and K matrix", which are gathered in Maple, are used to calculate the acceleration for time step T_n. The used matrices can be found in appendix D) The velocity and newly calculated acceleration are send back to the main file. Both the velocity and acceleration on timestep T_n together with initial conditions are used to calculate the position and velocity on T_{n+1}. This is done by a function of matlab called ODE45. ODE45 is a single-step solver, and is based on Runge-Kutta formula. As stated earlier this is a numerically method for approaching the actual velocities. Numerical implies there is also an error term. However, for relatively small time steps T_n this error is negligible small. This process is repeated for every time step. The matlab files are shown in appendix B

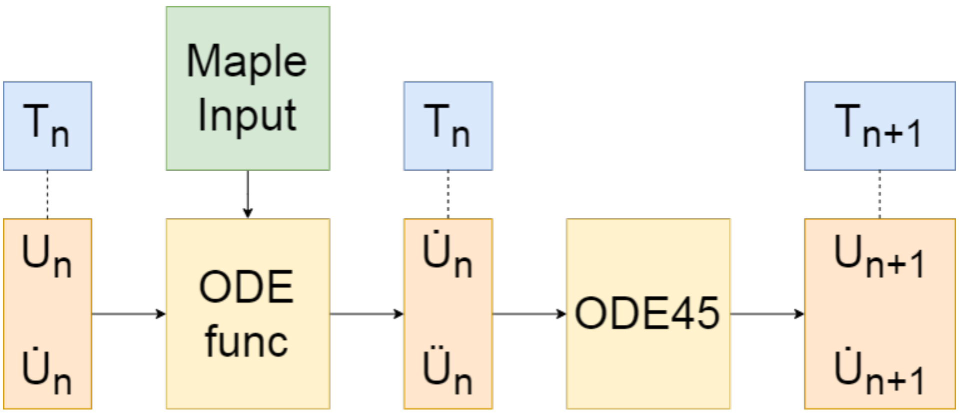


Figure 3.6: Matlab model overview

3.5. Validation Orcaflex model

For the validation of the matlab model, Orcaflex is used. The model treated in section 2.4 will be simplified to better represent the 2d matlab model. After which the eigen frequencies and associated modal shapes are compared to the results of the matlab model. Lastly, to complete the modal analysis, a sensitivity test is done. This creates a better understanding of the sensitivity of each DOF towards the lifting height.

3.5.1. Model Defined

For the validation with the Orcaflex model only a static analysis is required. In order to better represent the matlab model, some simplifications are done compared to the real model. The rigging system has been simplified and downgraded to only one tackle line, as well as leaving out the lifting hook (see figure 3.7). Additionally the mooring system has also been adjusted in order to enable better comparison of input parameters (see figure 3.8).

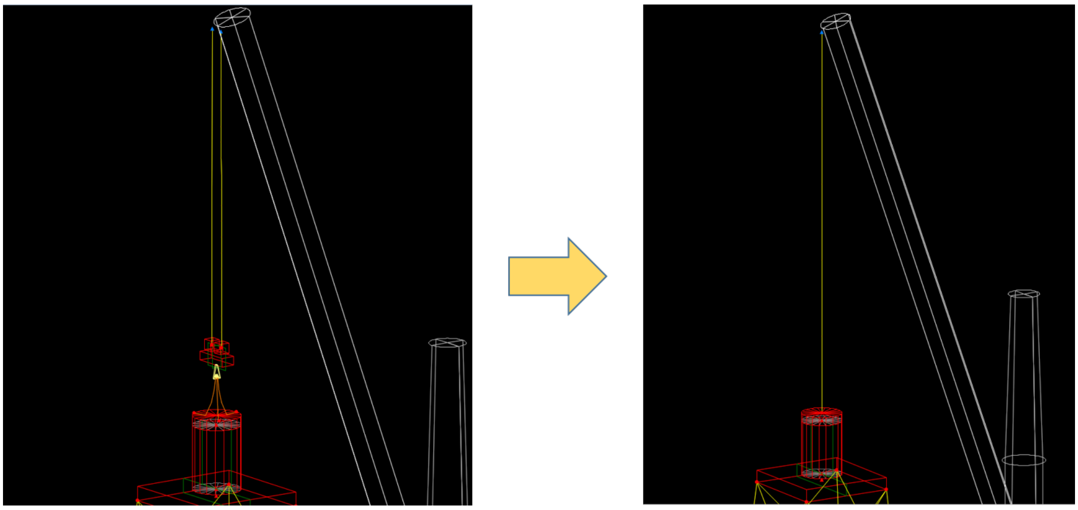


Figure 3.7: Simplification mooring

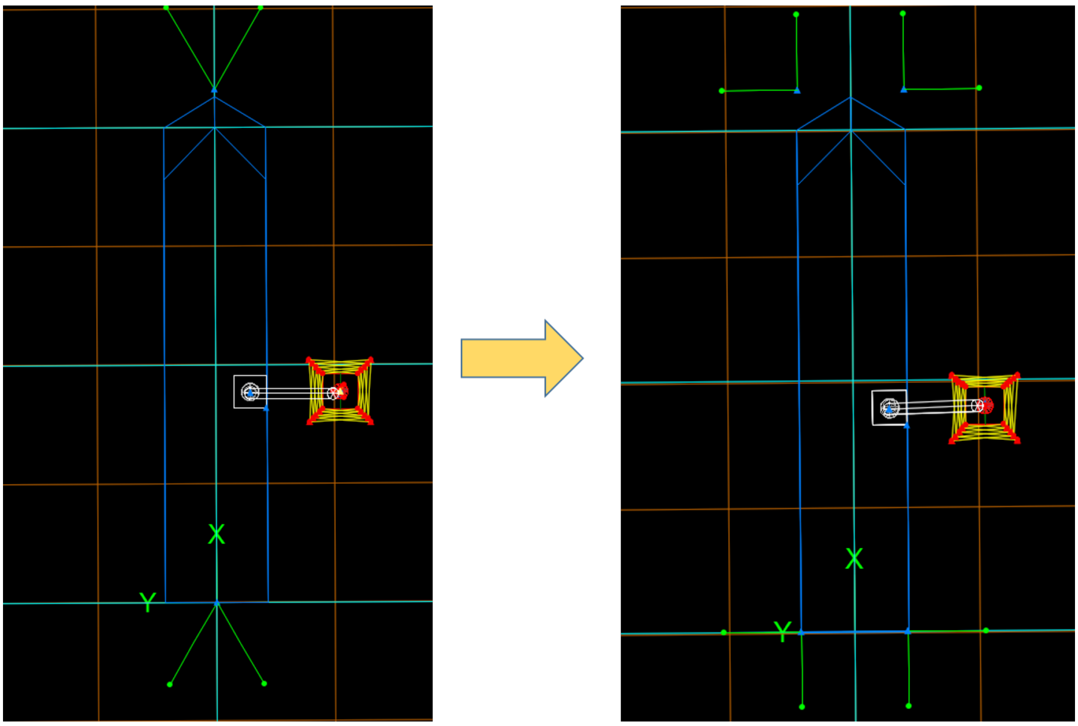


Figure 3.8: Simplification rigging system

Parameter	Value	Unit	Note
g	9.81	m/s^2	Gravitational constant
a	52.3	m	-
b	84.0	m	-
c	52.5	m	-
d	138.3	m	-
e	44.0	m	-
m	1.73e6	kg	Mass of Jacket
M	5.56e7	kg	Mass of Vessel
I_m	1.26e9	kg*m^2	Moment of inertia Jacket
I_M	1.86e10	kg*m^2	Moment of inertia Vessel
nt	1.00e4	-	Amount of steps
ksv	7.93e7	N/m	Vertical stiff. vessel
ksh	133.3e3	N/m	Horizontal stiff. vessel*
ksr	5.65e9	N/m	Rotational stiff. vessel

Table 3.1: Tugger properties *note that the horizontal stiffness of the vessel does not exist in reality

3.6. Validation

For both models a modal analysis is done, and both outcomes are compared against each other. The modal analysis includes the following items:

- Eigen frequency
- Sensitivity test

The input of the matlab model is shown in table 3.1. Additionally parameters are also illustrated in figure 3.3. Note that both the frequency dependent added mass and damping term are not yet included for the validation of the model. The damping term does not influence the natural periods of the model, thus is not yet required. The added mass does influence the natural frequency, however as both models do not include the added mass, it is correct to use for validation. Note that only the frequency dependent added mass is not included. However, the added mass caused by movement of the submerged load is calculated and included by Orcaflex automatically. As both loads are located above the waterline this has no effect on the eigen frequency.

The M-matrix and K-matrix, found in maple, are used in order to calculate the eigen frequencies and modal modes of the system. The eigen frequencies of all DOF of the vessel are shown in table 3.2. Comparing both calculated eigen frequencies shows large difference for θ_k and θ_m. This is mainly caused by the difference in properties of both tackles. As in the matlab model, the tackle is modeled as a pendulum with the rod having a infinite (bending)stiffness. Where as in Orcaflex the tackle is modelled as a line with a designated (bending)stiffness. This causes the difference in eigenfrequency between both models. For the remaining values: θ_s, x_s, and y_s, the values determined with the matlab model lie close to the values determined with the Orcaflex model. However the values still differ from each other slightly. Especially the rotation of the vessel θ_s. This is due to small differences between the models. Firstly, the COG and COB are the same location for the matlab model. As for the Orcaflex model the two have different coordinates. Secondly, the mooring lines in the Orcaflex model also contribute to the rotational stiffness of the model. This is due to the mooring lines not being attached exactly to the centre of rotation of the vessel. As is the case for the matlab model.

	Eigen frequency [Hz]	
DOF	**Matlab**	**Orcaflex**
θ_k (pendulum)	0.0342	0.0318
θ_m (double pendulum)	0.211	0.1711
θ_s (roll vessel)	0.0983	0.0805
x_s (sway vessel)	0.00770	0.00772
y_s (heave vessel)	0.188	0.188

Table 3.2: Eigenfrequency, result Matlab model and Orcaflex model

3.6.1. Sensitivity test

As part of the modal analysis, a sensitivity analysis is done. The analysis looks at the sensitivity of the natural periods for change in tackle length. This is tested by changing the height in steps of 5 m. A schematic overview is shown in figures 3.10. The results are presented in figure 3.9. Note that an eigen-period for DOF x_h does not exist in reality (horizontal stiffness = 0), and its only purpose is to model the DP system and keep the vessel in place. Therefor the value has been left out of the results. Besides showing the sensitivity of the system, the analysis also shows what the eigen-periodes will be troughout the lowering of the jacket for installation.

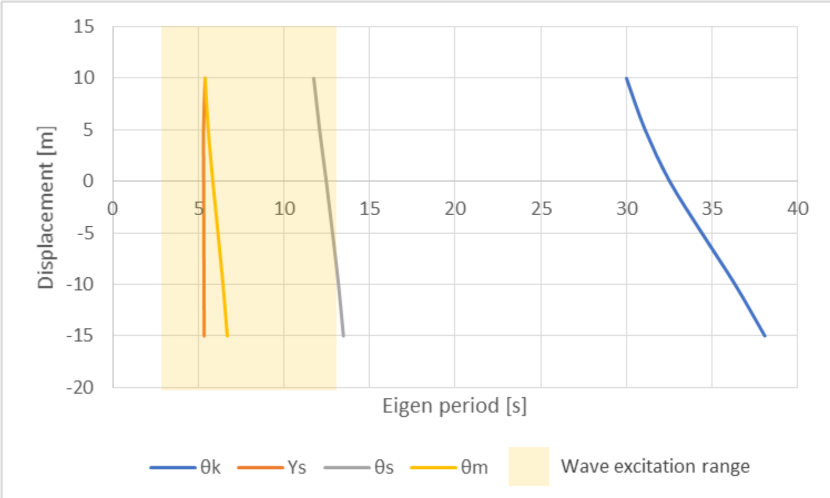


Figure 3.9: Eigen-frequency over change in height

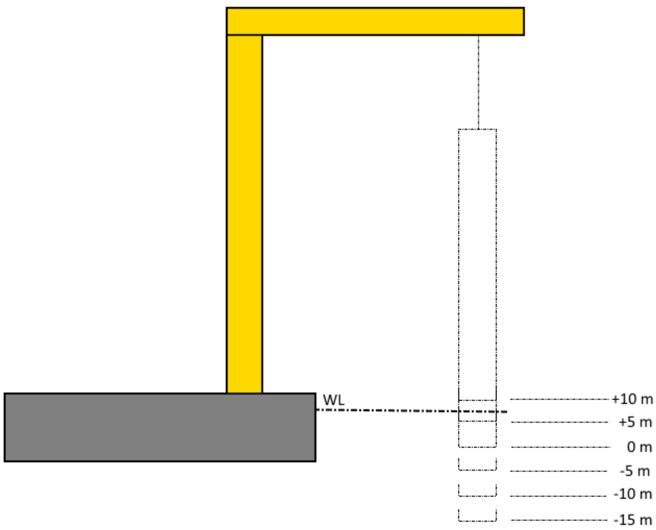


Figure 3.10: Schematic overview sensitivity test

From the sensitivity test three things can be observed (see figure 3.9). Firstly, the graphs shows all the eigen periods to increase with lowering of the jacket. What is as one would expect. As the jacket is lowered the water plane area increases. Causing an increase in added mass, and a slight increase in restoring force for heave motions. The increase in added mass causes the eigen period to rise. Secondly, it shows that the tackle length has the smallest effect on the eigen period for the vertical DOF of the vessel. Reason for this will be the relatively small increase of added mass in heave direction, combined with a slight increase of restoring force. Thirdly, it can be observed that the θ_k experiences the largest change in eigen period due to lowering. This is due to large increase in added mass for roll motion.

4

2D control systems

This chapter treats the testing of the control systems by the 2D model. Chapter 4.1 starts with explaining the basic principle of a control system, where after the damping functions used for the different control systems are introduced in chapter 4.2. Chapter 4.3 treats the last control system, namely the PID controller. Chapter 4.4, start with explaining the tests and model input, after which the PID is tuned. Where after all control systems are tested for their roll motion reduction and damp capabilities. To finalize the chapter is ended with a discussion in section 4.5.

4.1. Control systems

Traditionally, tugger systems works on two different modes, fixed length or constant tension (see section 1.3). To achieve larger damping, the constant tension, can be slightly modified. By varying the tension depending on the pay-out/ pay-in(hauling in) speed of the tugger line. This can be achieved by controlling the tension with the use of reel handling winches . This kind of tension control is illustrated in figure 4.1. The winches itself are often outfitted with sensors to measure the pay-in/pay-out velocity of the winch [13]. As the pay-in/pay-out velocity of the lines are directly related to the jacket motion this if used as input parameter for the tension control. Where the damping function determines the corresponding required wire tension. The requested value is sent to the actuator which in this case is the winch. The winch outputs the requested value as long as it is in the actuator's limits. These limits are:

- No tensions lower than zero
- No tensions larger than the winch capacity

The reason for not allowing tension lower than zero is to avoid slack within the cable. As slack will cause large forces within the cable, leading to failure. Obviously the tension can not exceed the force capacity of the winch. The maximum pay-out/pay-in speed of the winches is evaluated however not set as a hard limit. Additionally other limits such as the maximum increase or decrease rate of the tension are left out of this analysis.

It is important to state that within this report a perfect functioning system is assumed. Meaning, the winch system contains no error. To elaborate, a sensor or actuator always contains an error. No single actuator works perfectly, resulting in a difference between the requested value and outputted value. Additionally, no delays are assumed. The actuator react instantaneously and outputs the requested value directly. Both errors and delays are considered beyond the scope of this report.

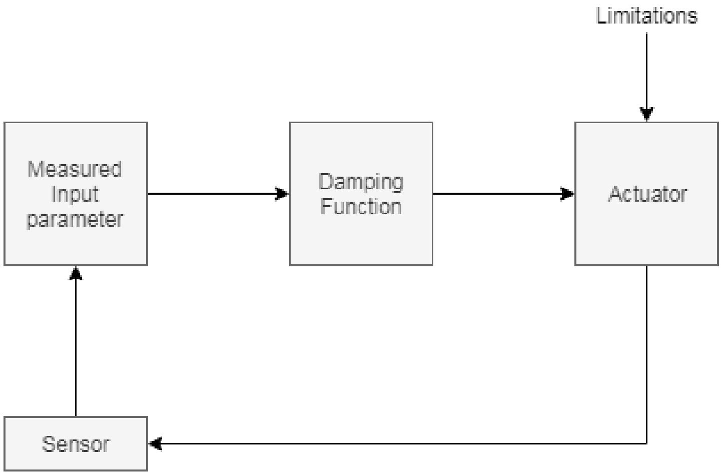


Figure 4.1: Schematic tension control

4.2. Damping functions

Seven different control systems are made using different damping functions. The functions are:

- Step-wise control system
- Linear control system
- Forward shifted linear control system
- Backwards shifted control system
- Quadratic control system
- Quadratic db control system
- PID control system

The PID control system is treated separately in chapter 4.3. Each control system is firstly tested using the 2D Matlab model. Secondly the "quadratic db control system" and "PID control system" are further analysed using the 3D Orcaflex model. All damping functions are shown in figure 4.2

4.2.1. Step-wise control system

The step-wise damping function immediately applies the minimum tension for hauling in and maximum tension for paying out (see figure 4.2a). In theory the winch has a infinite tension increase. In practice a winch will not have these capabilities due to physical limitations, and the curve would contain a slope. The function has no optimization options.

4.2.2. Linear control system

The linear damping function applies a proportional factor on the velocity (see figure 4.2b). The function actually represents what an imperfect stepwise function looks like. The slope can be changed dependening on the desired outcome.

4.2.3. Linear shifted forward control system

The linear forward damping is a variant on the linear damping (see figure 4.2c). In which the winch will start applying tension only at paying out.

4.2.4. Linear shifted backward control system

The linear forward damping is a variant on the linear damping (see figure 4.2d). In which the winch applies maximum tension at zero velocity.

4.2.5. Quadratic control system
The quadratic damping consists of a combination of a linear term and quadratic term (see figure 4.2e). So additional to a term proportional to the velocity of the line, it also contains a term which is proportional the the velocity squared.

4.2.6. Quadratic db control system
The quadratic db damping is a variant on the quadratic damping (see figure 4.2f). The function also adds a deadband. Which is a value the tension skips passing zero velocity. Note that if the deadband is high enough the result will be a stepwise damping function. In practise the deadband is not perfectly vertical due to physical limitations.

4.3. PID-damping controller
The PID-damping function will not apply a damping function as shown in figure 4.2. But will base the required tension on a PID controller. A PID (proportional-integral-derivative) controller is a commonly used control algorithm. The PID and small variations of it are often used in feedback loops. It measures an "output" of a process and controls an "input". A PID controller continuously calculates a error term [6]. This is the difference between the desired set point and the measured value. In case of the tugger system the desired value is zero. The aim of the controller is to make the error approach zero. It tries to accomplish this by treating the error in three different ways simultaneously. A proportional term, a integral term and a derivative term. All three terms contribute to the output of the controller, and it depends on the application of the control system how to weigh each contribution. This is done by adjusting the weight factor (k_p,k_i,k_d) for each term, also known as tuning. [12]

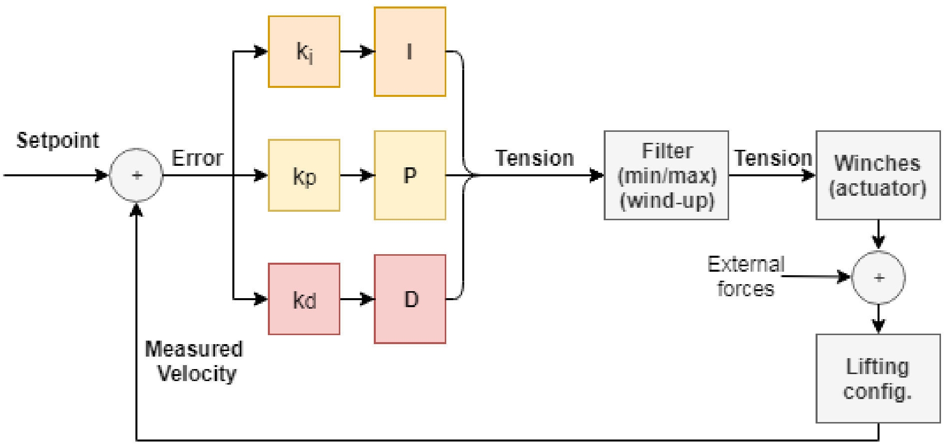


Figure 4.3: Block diagram of PID controller

4.3.1. Proportional term
The proportional term (proportional to the error) represents the present error. Compared to an on-off control system (stepwise function), a proportional control avoids oscillations due to overreacting of the system. As for small error the system reacts proportional. In case of the tugger system the proportional term will react on the winch-line velocity. Just like the systems treated in chapter 4.3. Unfortunately a purely proportional control has the drawback that is mostly results in a static or steady state error. To get rid of the error a integrator path can be added to the controller

4.3.2. Integral term
The integral term (proportional to the integral of the error) represents the past. The integrator term sums up the input signal over time, keeping track of the total over time. With a steady state error, the error is non-zero, so the integral term will increase or decrease, depending on whether the error is positive or negative. Until the steady state error becomes zero, no matter how small the error is. In the case of the tugger system, the integral

(a) Stepwise damping function

(b) Linear damping function

(c) Linear forward daming function

(d) Linear backward damping function

(e) Quadratic damping function

(f) Quadratic db damping function

Figure 4.2: Damping functions of the P-control systems

term is the difference between the current winch line length and the original length. As long as this difference is non-zero the integrator will intervene. In conclusion, a combination of proportional and integral control will get the system to the set-point. However the path to the set-point may not be perfect. As the integrator term will only decline when passing the set-point, creating a overshoot. This is where the derivative term can contribute.

4.3.3. Derivative term
The derivative term (proportional to the derivative of the error) represents the future. The main purpose of the derivative term is to improve stability. It measures the rate of change of the error. In other words, how fast the error is increasing or decreasing. In case of the tugger system this is the acceleration of the winch. By looking at the rate of change of the error it enables the system to prematurely slow down the system, preventing it from overshooting the set-point too much. In other words, if the measured value is coming close to the set-point, the error is decreasing. Which has a negative rate of change, resulting in a negative contribution of the derivative term lowering the total (requested value). In case of the tugger system, the requested tension. It could be said that it dampens the system, which allows for higher proportional term, speeding up the response.

4.3.4. Applying PID control on tugger system
Applying a PID controller on the tugger system will result in a closed loop control system illustrated in figure 4.3. The set-point of the winch velocity is zero. Resulting in the error being the winch line velocity. From the error (P) the corresponding integral (I) and derivate (D) terms are calculated. The controller then calculated the tension using formula 4.1. Where the k0 is the pretension (T_target). This value is sent to the filter. The filter is used to filter out tension requests that do not comply with the set maximum and minimum. For example the winches are not allowed to create compression. Additionally the filter is also used prevent integral windup. Integral windup can occur when the actuator is saturated and the output value takes long for passing the set-point value. What happens is that the integral value becomes such a large value that after the error switching sign and the integral output start to decrease, the requested value is still the saturation limit. Only after the integrator term has decreased enough will the actuator start reacting. There are several ways to deal with integral windup. Here is chosen to switch off the integral term if the winch hits it maximum or minimum. That way the requested value will never exceed a set limit. Once the controller architecture is set, the weight factors need to be tuned (k_p,k_i,k_d). In order to do it correctly, it is important to understand the controlled system and its requirements. Desirable for the tugger system is a controller which is fast, stable and able to deal with changing accelerations of the jacket. As the jackets experiences continuously forcing with changing load and direction. As there exists a model of the system, model-based tuning is used, in which the factors are tuned manually, until it produces the best response.

$$F_{tension} = k_0 + k_p * P + k_i * I + k_d * D \tag{4.1}$$

4.4. 2D model study
The Matlab model is used to test the effect of the different control systems. The input parameters of the model are shown in table 3.1. Additionally, the model includes the added mass of the system (shown in equation 4.2), and the damping terms 4.1. Both are calculated by the program ANSYS Aqwa. Note that in order to keep the model relatively simple the suction bucket jacket is located fully above water line, and experiences no excitation from the waves directly. This analysis is performed using the more extensive software Orcaflex, chapter 2.4. Details on the calculation of the tugger lines length, velocity and acceleration are shown in appendix C. The systems are tested on:

- Maximum Jacket roll motion
- Decay test

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5.98e9 & 2.92e4 & 5.20e5 \\ 0 & 0 & 2.92e4 & 9.22e5 & 3.08e6 \\ 0 & 0 & 5.20e5 & 3.08e6 & 9.34e7 \end{bmatrix}, \begin{bmatrix} \theta_k \\ \theta_m \\ \theta_s \\ x_s \\ y_s \end{bmatrix} \tag{4.2}$$

Parameter	Value	Unit	Note
csv	1.40e4	kg/s	Vertical damp. vessel
csh	1.92	kg/s	Horizontal damp. vessel
csr	541.7	kg/s	Rotational damp. vessel

Table 4.1: Values for damping coefficient, for illustration of these values see 3.3

Comb.	K$_P$	K$_I$	K$_D$	Max. roll	Unit	Motion reduction	Unit
1	1.50E+06	0	0	0.117	[rad]	4	[%]
2	0	1.50E+06	0	0.119	[rad]	3	[%]
3	0	0	1.50E+06	0.230	[rad]	-88	[%]
4	1.50E+06	1.50E+06	1.50E+06	0.113	[rad]	7	[%]
5	1.50E+06	5.00E+05	0	0.107	[rad]	13	[%]
6	1.50E+06	0	5.00E+05	0.122	[rad]	1	[%]
7	5.00E+05	1.50E+06	0	0.117	[rad]	5	[%]
8	0.00E+00	1.50E+06	5.00E+05	0.119	[rad]	3	[%]
9	5.00E+05	0.00E+00	1.50E+06	0.170	[rad]	-39	[%]
10	0.00E+00	5.00E+05	1.50E+06	0.130	[rad]	-6	[%]
11	1.50E+06	5.00E+05	5.00E+05	0.102	[rad]	17	[%]
12	5.00E+05	1.50E+06	5.00E+05	0.116	[rad]	5	[%]
13	5.00E+05	5.00E+05	1.50E+06	0.126	[rad]	-3	[%]
14	1.50E+06	2.00E+05	5.00E+05	0.104	[rad]	15	[%]
15	1.50E+06	5.00E+05	2.00E+05	0.105	[rad]	15	[%]
16	5.00E+05	1.50E+06	2.00E+05	0.116	[rad]	5	[%]
17	2.00E+05	1.50E+06	5.00E+05	0.116	[rad]	5	[%]
18	5.00E+05	2.00E+05	1.50E+06	0.137	[rad]	-12	[%]
19	2.00E+05	5.00E+05	1.50E+06	0.127	[rad]	-3	[%]
20	1.00E+06	1.00E+06	5.00E+05	0.113	[rad]	8	[%]
21	1.00E+06	5.00E+05	1.00E+06	0.094	[rad]	23	[%]
22	5.00E+05	1.00E+06	1.00E+06	0.116	[rad]	5	[%]
23	**1.10E+06**	**5.00E+05**	**8.00E+05**	**0.092**	**[rad]**	**25**	**[%]**
24	5.50E+05	2.50E+05	4.00E+05	0.101	[rad]	17	[%]
25	2.20E+06	1.00E+06	1.60E+06	0.094	[rad]	24	[%]

Table 4.2: PID tuning results. For all cases the T$_{tartget}$ is set at the same value as the linear and quadratic control system

4.4.1. Model loading

The model uses airy waves to load the model. The wave is defined by wave height, period and direction. The used wave height is 1m. The period is chosen close to the pendulum mode of the lifting configuration (T=30.0), to excite resonance. As this is of interest for the analysis. The waves originate from the left side of the model. The corresponding forces are calculated using the Load RAOs.

4.4.2. Winch limits

The minimum tension for the winch is set at 200 kN to take into account the own weight of the lines. As in the model the winch lines are weightless. However, in reality the winch lines will take a catenary shape due to its own weight, creating slack. The maximum value of the winch is set at 15e5 kN.

4.4.3. PID tuning

To enable fair comparison between control systems, the PID requires tuning. Table 4.2 shows 25 different PID combinations. They consist of 6 different cases in which different combinations are tested.

Case 1 [1-4]: Analyze the effect of each term individual
Case 2 [5-10]: Analyze combination of one dominant term, one minor term and one zero term
Case 3 [11-13]: Analyze combination of one dominant term and two minor terms
Case 4 [14-19]: Analyze combination of one dominant term, one minor term, and one small term
Case 5 [20-22]: Analyze combination of two medium terms and one minor
Case 6 [23-25]: Analyze the effect of halving and doubles in the values of the optimized PID combination [23]

The results show the motion reduction compared to a system without tugger damping. A negative reduction indicates a motion increase. Combination 3, 9, 10, 18 and 19 show that a relatively large D term results in a negative effect. This is as expected, as the D term has a inhibitory effect on the output, to reduce overshoot. This means making it too large will eliminate the other terms completely, resulting in a negative effect. Case 5

Constol system	Max. roll motion	Unit	Motion reduction	Unit
Stepwise	0.129	[rad]	-5.3	[%]
Linear	0.116	[rad]	5.5	[%]
Linear forward	0.098	[rad]	20.4	[%]
Linear backward	0.147	[rad]	-19.7	[%]
Quadratic	0.117	[rad]	4.9	[%]
Quadratic db	0.118	[rad]	3.4	[%]
PID	**0.092**	**[rad]**	**24.7**	**[%]**
No tugger	0.123	[rad]	0.0	[%]

Table 4.3: Motion analysis for different control system, wave loading with T=30 sec

shows a combination of a medium P and D, and a minor I works best for the system. From here the optimized combination is found by iterative manually optimize the individual terms of the PID, with steps of 1e5 kN. Till it results in the lowest maximum roll value. Case 6 shows that higher values of the optimized ratio do not necessarily lead to smaller motions. From this it can be concluded that there exists not only a optimum ratio, but also value. The damping function of the PID is shown in figure 4.4.

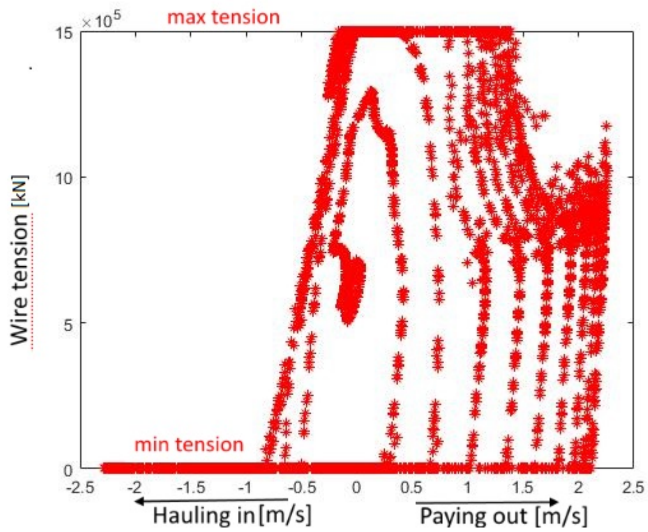


Figure 4.4: PID damping function

4.4.4. Maximum Jacket roll motion

With the model the maximum roll motion of the jacket is calculated of a 500 seconds simulation for all seven control systems. The reason for comparing the maximum roll motion is because the motion is an indication for the forces on the crane tip. Higher roll motions indicate larger forces on the crane. Which is important as it is often the limiting factor for lifting operations. The results are shown in table 4.3. Details on programming of the control system is shown in appendix B.2.

The results indicate that the PID controller is the most effective in reducing the roll motion of the vessel. Which indicate it creates the lowest forces on the crane tip. Both the "Stepwise" and "Linear backward" control systems actually create larger motions instead of reducing it. In theory the stepwise system does not input any energy into the system, and only takes energy out. As the applied force will always have a opposite direction from the velocity, in contrast to the other control systems. However, as the jacket also experiences environmental forces coming from the crane tip, the stepwise can still result in the largest motions, as what can be seen from the results. A deadband causes relatively large forcing at small velocities. Subsequently, when the jacket is moving away from the vessel, near the point at which it switches direction, the deadband control system causes the top of the jacket to slow down faster compared to other control systems. This results in a larger difference in velocity between the top and centre of the jacket causing a rotation of the jacket

	Max. roll motion			Motion reduction		
Constol system	**Original para.**	**Adjustes para.**	**Unit**	**Original para.**	**Adjustes para.**	**Unit**
Stepwise	0,0098	0,0098	[rad]	64,7	64,7	[%]
Linear	0,0113	0,0092	[rad]	59,4	66,9	[%]
Linear forward	0,0123	0,0084	[rad]	55,8	69,8	[%]
Linear backward	0,018	0,0136	[rad]	35,3	51,1	[%]
Quadratic	0,0111	0,0092	[rad]	60,1	66,9	[%]
Quadratic db	0,0094	0,0098	[rad]	66,2	64,7	[%]
PID	0,0138	0,0059	[rad]	50,4	78,8	[%]
No tugger	0,0278	0,0278	[rad]	0,0	0,0	[%]

Table 4.4: Motion analysis for a wave loading with T=5 sec, comparing the parameters used for T=30 seconds against 5 times these parametersO

(roll motion). What also stands out from the control systems which include deadband are the large required velocities when the winches go from paying in to paying out. When the winches are paying in, the winches will pull at 200 kN, until the jacket slowly reaches zero velocity. The first moment the velocity becomes positive the winch immediately will apply full force. Giving the jacket a large acceleration, which can switch the sign of the velocity again, resulting in again an output of 200 kN. This behavior results in large fluctuations in winch force output near the transition from paying in to paying out. Which also result in large velocity peaks of the winch around this area.

The linear backward has the largest pretension of all control systems, namely the set maximum of 15e5 kN. The winch only stops pulling completely when the pay-in speed becomes 1.5 m/s. Which means the winches also put energy in the systems in such an amount that it increases the motion of the jacket.

Comparing "quadratic" and "quadratic db" control systems, shows that the deadband has a negative effect on the motion reduction. The deadband induces the same behavior around zero velocity as the stepwise experiences mentioned earlier. The linear forward control system reduces the motions efficiently. The system has no pretension and does not introduce energy into the system just like the stepwise system. Unlike the stepwise control system, after the velocity changes sign, the tension slowly builds up. Resulting in a smoother movement of the load, and scoring best from all partial damping functions. The PID has the highest motion reduction from all control systems. As the PID controller has a relatively large D term, meaning the acceleration of the jacket has a large effect on the tension. This damps out large fluctuations in the error just enough. In order to test whether the order of motion reduction depends on the loading frequency, the system is also tested with a 5 second period (which does not cause resonance within the system). This is firstly done for the control systems with exactly the same parameters properties as previously optimized for the 30 seconds period wave. The results can be seen in table 4.4 under "original parameters". It firstly shows that, as expected, the roll motions are much smaller. Secondly the stepwise control system together with the quadratic db causes the smallest motions. Both controllers which contain a skip in force when passing zero velocity and scored worst in the previous analysis. This is caused by the smaller motions and velocity due to the wave frequency. The small velocities cause the lines tension for the linear, linear forward, quadratic control and PID systems to not vary a lot anymore, due to the low slopes. If the input terms (linear and quadratic) are multiplied by 5 it shows the same results as for 30 second period wave (see table 4.4 under adjusted parameters). The results show control system which are more tuned, like the PID, to perform better. The results also show that this is no guarantee for other loading frequency. To elaborate, a well tuned control system will performs good for the specific loading frequency for which its tuned. However, it will perform less the further away from the tuned value. Where as a not tuned control system such as the stepwise will perform more consistently.

4.4.5. Decay test

The different control systems dampening effects are tested by a decay test. Here the double pendulum (θ_k and θ_m) is given an initial displacement of 0.085 (5 degrees). Where after is tested how long it takes for the system to be fully damped out. The system is considered fully damped out when the rotation of the jacket does not exceed 0.005 rad anymore. The dampening over time is shown in figure 4.6, and the results are shown in table 4.5. The results show the stepwise and quadratic db dampen the system the fastest. Both control systems have a deadband at zero velocity. From the results can be concluded that a deadband works beneficial for dampening systems which do not experience any additional forcing. However when the system does experience forcing from the outside, the deadband often has a negative effect on the motion dampening. Additional both systems keep a small vibration after being damped out due to to large force fluctuation around zero velocity, reacting to motion it induces itself. The linear backward still induces too much energy into the system that compared to what it takes out, that it scores lowest from all control systems. The linear forward takes too little energy out to dampen the system fast. The PID control system scores best from all system which do not include a deadband. Note that The control systems were not altered or re-tuned for the test. For example, the slope of the linear could be made so high that acts exactly as the stepwise. The PID can even be tuned beyond the result of the stepwise due to its I and D term.

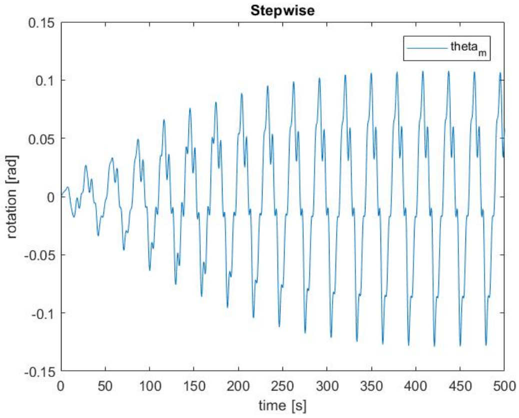
Constol system	**Duration**	**Unit**
Stepwise	93	[s]
Linear	184	[s]
Linear forward	220	[s]
Linear backward	270	[s]
Quadratic	174	[s]
Quadratic db	97	[s]
PID	164	[s]
No tugger	822	[s]

Table 4.5: Decay test results for different control systems

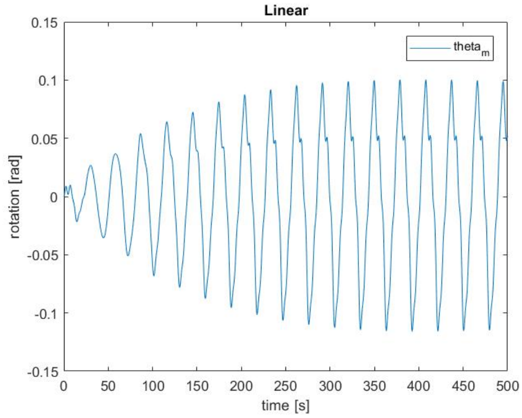
4.5. Discussion

To summarize, the chapter investigates multiple control systems which try to reduce the motions of the systems with use of velocity controlled tugger winches. In total, seven different control systems are tested. In order to have the PID controller most effectively, it requires to be tuned. As a wrongly tuned PID controller will have no contribution or may even increase motions. For the 2D Matlab model a combination of high P and D and a lower I works most beneficial for the motion reduction. It is found that a higher values of the optimized weight factor combination do not necessarily lead to lower motions. There exists an optimum ratio and value of the terms. The different control systems are tested with both a roll motion analysis, and a decay test. Both the "stepwise" and "linear backwards" control system cause larger roll motions compared to a model without a tugger system. The stepwise control system shows that larger forcing does not always lead to higher motion reduction. As when the model also experiences continuous changing environmental loading, the combination of the forces may lead to larger motions. The systems including a large deadband such as the "stepwise" and "quadratic db" or system with high pretension such as the "linear backwards", apply relatively high forcing when the velocity approaches zero when paying out. The jacket still has velocity directed away from the vessel. As the forcing is applied on top of the jacket directed towards the vessel this creates a rotational motion of the jacket. Resulting in a roll motion of the jacket. The other control systems apply the forcing more gradually when the velocity approaches zero, causing a smaller velocity difference between the top and COG of the jacket. Resulting in relatively smaller roll motions.

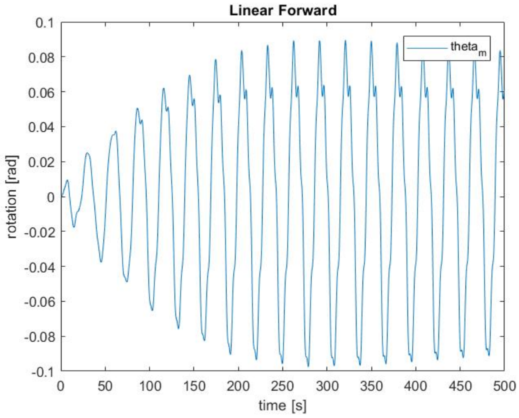
The 2D model shows the "linear" control system performing slightly better compared to the "quadratic" control system. However, both the linear forward and PID score best in roll motion reduction. When correctly tuned the PID control systems has to highest roll motion reduction. Compared to a model without tugger system it reduces the maximum roll motion by almost 25 percent. However, this is not necessarily the case for other wave loading frequencies. As the loading frequency and wave height determines the motion and velocity of the load, the control systems requires to be tuned depending on the properties of the wave. As slower motions often require steeper functions, to damp out more effectively, and vice versa. Furthermore, when looked at system which does not experience any additional forcing, functions containing a deadband damp the motion the most effective from all the dampening function. Note that a deadband does only exist in theory, and can not exist in this form in reality. A deadband works effectively for damping the lifting configuration when a system does not undergo continues loading. The shift in forcing between paying in and paying out cause higher counter forcing while minimizing energy input into the system.



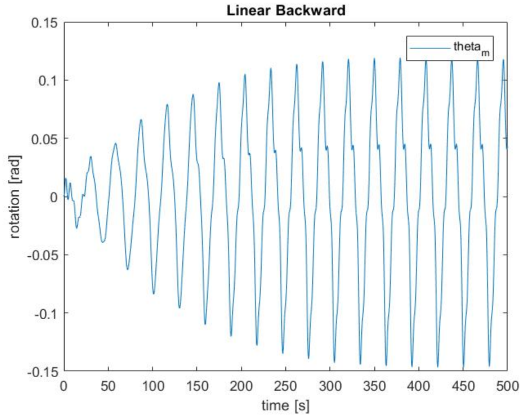
(a) Motion Stepwise control



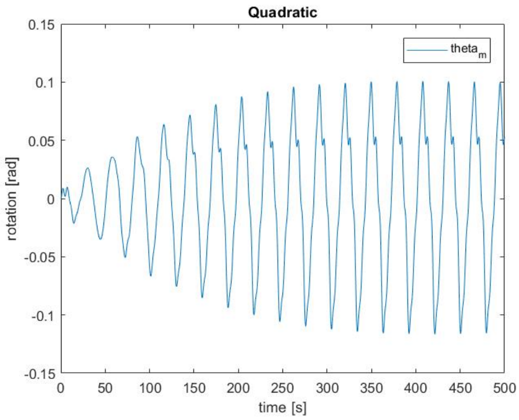
(b) Motion linear control



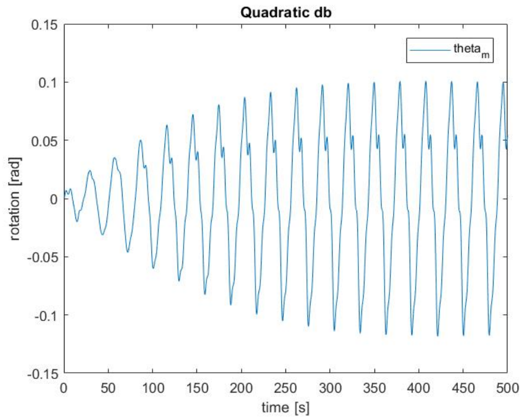
(c) Motion linear forward control



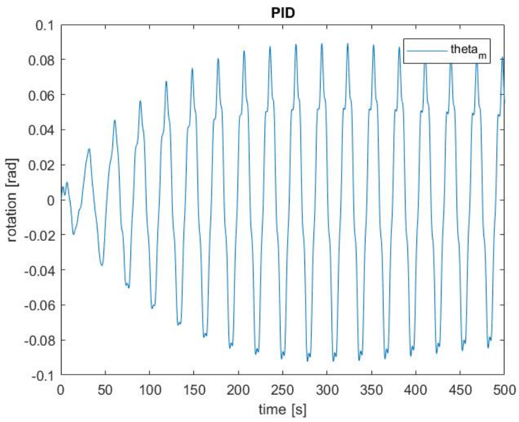
(d) Motion linear backward control



(e) Motion quadratic control

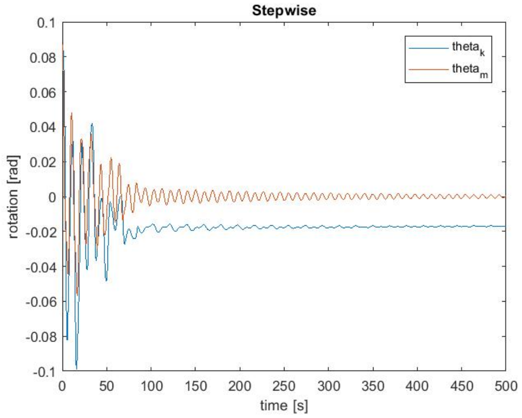


(f) Motion Quadratic db control

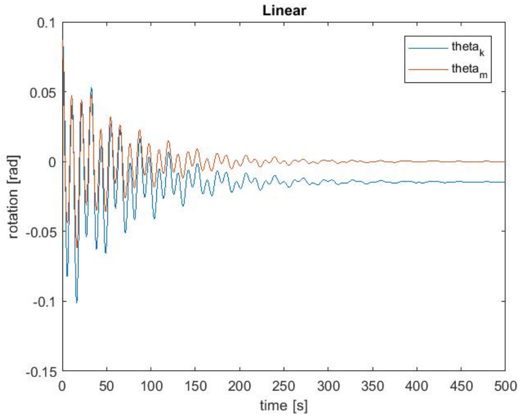


(g) Motion PID control

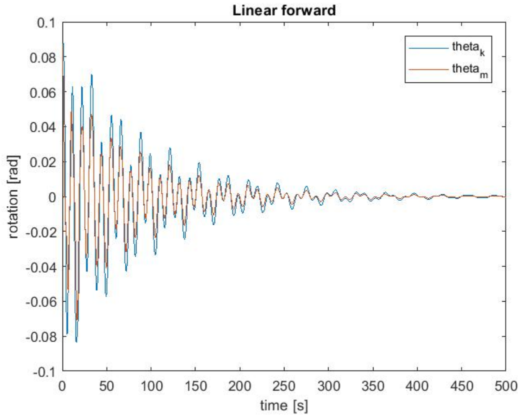
Figure 4.5: Jacket roll motion due to wave loading



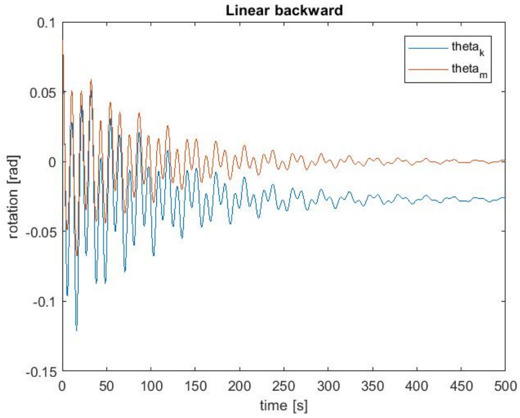
(a) Decay test stepwise control



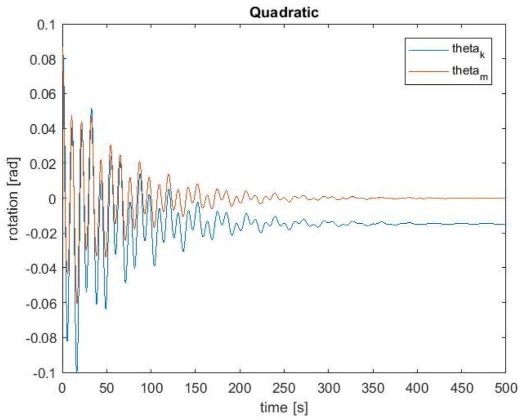
(b) Decay test linear control



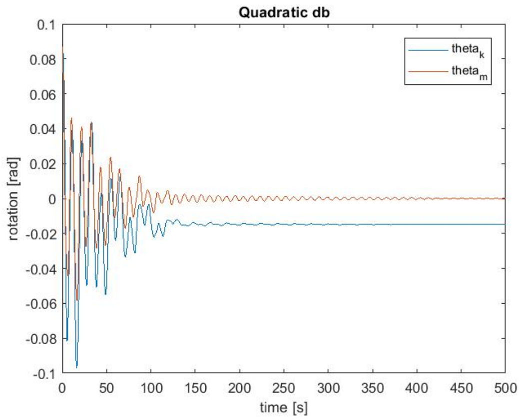
(c) Decay test linear forward control



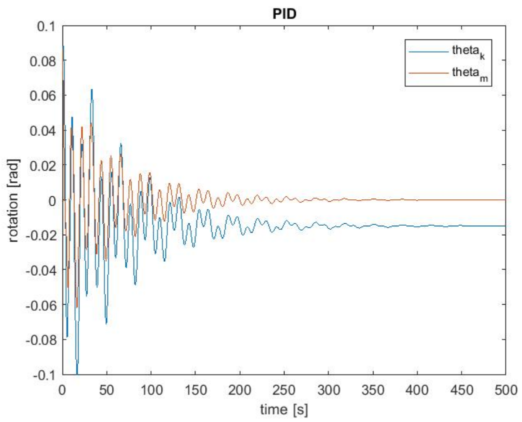
(d) Decay test linear backward control



(e) Decay test quadratic control



(f) Decay test quadratic db control



(g) Decay test PID control

Figure 4.6: Decay test results over time

5
3D control systems

Following on the results of chapter 4. This chapter will research the effect of a linear-control, quadratic-control and PID control systems only now using the 3D matlab model (The model used is described in chapter 2.4.). Additionally, checking if the tugger damping can increase the maximum operational wave height. Instead of comparing the reduction in roll motion and the effect on a decay test, the control systems are tested on their reduction in crane tip forces. For this reason the chapter starts with treating the different failure mechanisms of the crane en winches in section 5.0.1. In order to compare the resulting forces, firstly it must be known how the model is loaded. A modal analysis is done for the system in section 5.0.2 to find which wave periods are of interest. After that the model is tested for different wave periods in order to find the normative wave period. After the loading conditions are known three different p-control systems are created using the pre-defined winches in Orcaflex. The winches are shortly explained where after a linear control system, a quadratic control system (containing only a quadratic term), and a combination control system (containing a linear and quadratic term) are analyzed in section 5.1.1. After which the optimized linear controller is presented in chapter 5.1.2. Next, a PID controller is created for the Orcaflex model in section 5.1.3. In Chapter 5.2 the results for the Linear control system, PID control system are compared with each other and compared to a no tugger model. The chapter is finalized with a discussion in chapter 5.3.

5.0.1. Failure mechanisms

For heavy lift operations there are obviously limits which need to be met. Often the bottle neck on the lifting operations are the forces on the crane. Besides the crane limits, the tugger system itself also has limitations. For the report the following failure mechanisms are accounted for:

Crane failure: (see figure 5.1)

- **Vertical load**: The exceedance of the maximum force the crane can have in z-direction
  Allowable vertical load: 22977 kN
- **Offlead**: The exceedance of the maximum force the crane can have in x-direction
  Allowable offlead load, directed towards the crane: 1748 kN
  Allowable offlead load, directed away from the crane: 798 kN
- **Sidelead**: The exceedance of the maximum force the crane can have in y-direction
  Allowable sidelead load: 1197 kN

Winch failure:

- **Min tension deck/boom winches**: lower values than 0 indicates slack in the tugger lines.
- **Max tension deck/boom winches**: the exceedance of the maximum tension the winches can withhold.
  Boom winch: 441 kN
  Deck winch: 2000 kN

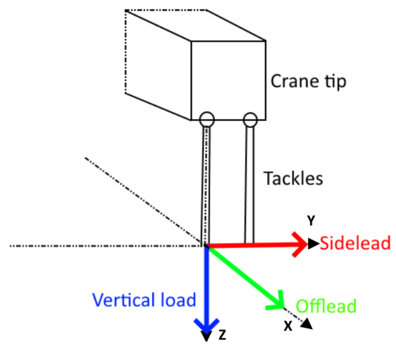


Figure 5.1: Crane tip forces directions

5.0.2. Modal analysis

Mode shapes describe the coupling between all degrees of freedom. Each mode shape is a natural vibration of the system. In order for better understanding of the system and its behavior for certain loading frequencies the mode frequencies and shapes of the model are calculated. The eigenvalues of the system are obtained with the modal analysis of orcaflex. The most important modes are shown below. Modes with a higher period are not included as they are mostly horizontal modes which are the results of the mooring lines, and modes related to the vessel which do not cause large motion of the lifting configuration. Modes with a lower period than 5.3 sec are not of interest as well, as they are the modes of the tackle or so small that the wave height corresponding to that period is too small to have a large impact on the system.

Important modes: [Vessel-Jacket].

[1] 5.3s: Pitch-Pitch [double pendulum] (out of phase)
[2] 5.5s: Heave-Roll [double pendulum]
[3] 5.7s: Roll-Roll [double pendulum] (out of phase)
[4] 6.5s: Pitch-Pitch [double pendulum]
[5] 16.9s: Roll-Heave

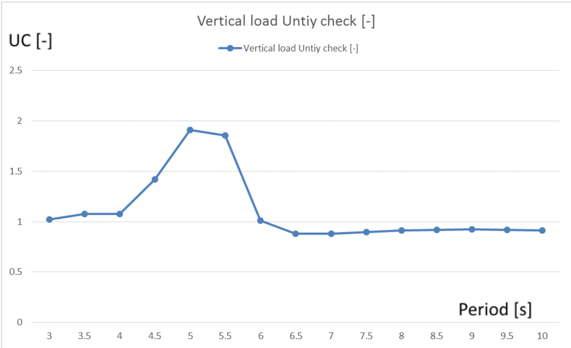
Note that the first term of the name refers to the motion of the vessel and the second to the jacket. Roll of the jacket corresponds to rotation of the object in the same plane as roll of the vessel, the same goes for pitch.

5.0.3. Wave loading

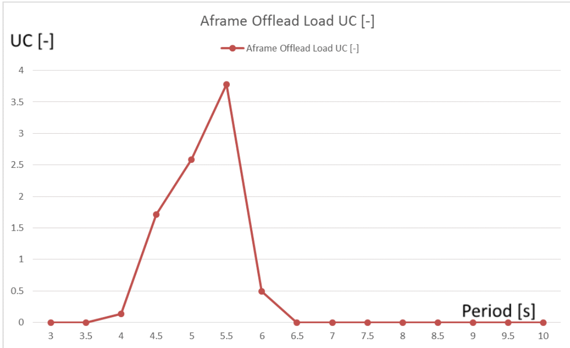
The model will be loaded with linear airy waves. This allows easy comparison of the results using different winch settings. With airy waves it is easier to stimulate a single mode and cause resonance. Because of this it is considered a conservative analysis to load the system with airy waves. The system only considers first order wave loading as second order loads are of important for station keeping, but do not contribute to considered resonance problem.

The Bokalift 1 features a DP-2 system. This enables the Bokalift 1 to keep position with the most favourable heading. For this reason the airy waves will meet the vessel head on (180 degrees). As the vessel has more stability in the longitudinal direction. It also prevents large roll motions of the vessel. Which has large coupling with the lifting configuration.

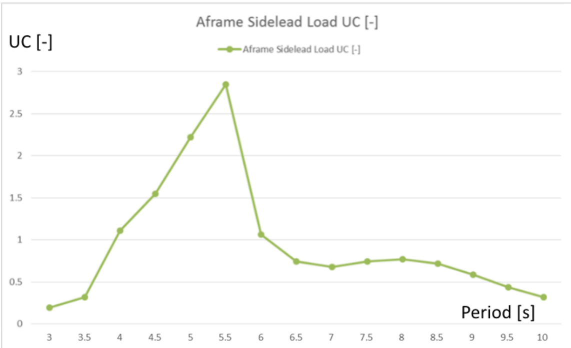
The wave height is taken at 3.3m, as this value normally is higher than the operational limits of the vessel. To research if with a tugger damping system it is possible to operate with this wave height while meeting all limits. Consequently the model requires the wave period which induced the highest forces. To find this wave period, the system is loaded with head on waves with a wave height of 3.3m with wave period varying from 3 to 10 seconds. Both ends fall inside the "wave spectrum" and contain the important modes of the system. Every period between 3 and 10 second is tested with steps of 0.5 second. This creates 15 different load cases. The effect is tested for each failure mechanism, the results are shown in figures 5.2.



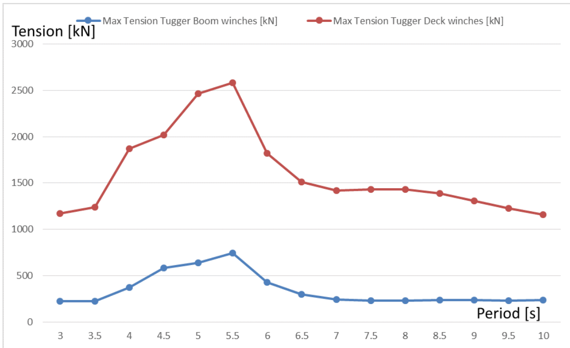
(a) The UC for vertical crane loads for different wave periods



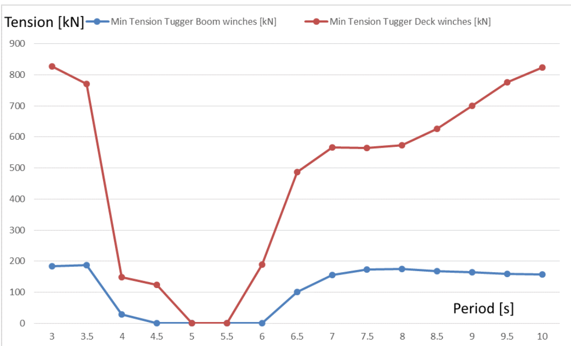
(b) The UC for offlead crane loads for different wave periods



(c) The UC for sidelead crane loads for different wave periods



(d) The maximum winch wire tension for different wave periods



(e) The minimum winch wire tension for different wave periods

Figure 5.2: Result of different wave periods for different failure mechanisms

The forces on the crane show relatively small loading between a period of 3 to 4 seconds. Where after the forces increase and reach its peak around a wave period of 5.5 second. This is close to both the period of mode [1], [2] and [3] and results in circular motions of the jacket as both pitch and roll motions are fueled. After a period of 6.5 second the forces goes down again and reach normal values again. From the graphs can be seen what influence the modes can have on the crane tipe forces. Note that for the offload forces (figure 5.2b), the graph states the forces go down to zero. This is not actual the case, and the force is just directed away from the crane which is a less crucial limit.
For the limit of the winches a same pattern can be observed. The maximum tension build up from 3 to 5.5s periods with its limit at 5.5s. Where after the maximum tension slowly declines again. The minimum tension starts high but rapidly declines to zero and only after period 6 reaches above zero again. From the graphs can be concluded that a 5.5 s period is the worst period for this system, and for reason is used for the analysis on the tugger system.

5.0.4. Current and wind loading

Both current and wind are not taken into account for this report. As the focus of the report lies at the damping the dynamics of the lifting configuration. Current does not influence significantly to these dynamics and thus falls out of scope for this report. The wind is also left out of the simulations, as it will apply aerodynamic damping to the system. Which effects are not of interest, as for the report only the dampening effect of the tugger system is of interest. Wind gusts will not contribute to the resonance, as the loading direction and frequency does not correspond to the wave loading.

5.1. Orcaflex winch systems

Now the the systems failure mechanisms and loading conditions are known the model can be used to analysis on tugger control. The tugger control is modelled using the predefined winches of Orcaflex. The winches have two different classes and types. Further elaboration on this can be found in appendix E.2. For the model a detailed winch which is tension controlled is used. Just like in the matlab model the output of the damping function depends on the pay out/pay in velocity of the winch. The damping function can be altered by adjusting the winch input parameters. This makes up the equations shown equation **??**. The corresponding damping function is illustrated in figure 5.3. This allows to create both linear and quadratic P-control systems. Note that the Orcaflex winch model lacks the option to set a maximum or minimum tension. This results that the winch parameters need to be carefully chosen in order to not exceed the tension limits.

$$f(v) = \begin{cases} f(0) - d + c_i v - d_i v^2 & \text{if } v < 0 \\ f(0) & \text{if } v = 0 \\ f(0) + d + c_i v + d_i v^2 & \text{if } v > 0 \end{cases} \qquad (5.1)$$

where:

v = rate of payout = dl_0/dt

d = winch drive deadband

c_i,c_o = winch drive damping terms for haul in and pay out

d_i,d_o = winch drive drag terms for haul in and pay out

$f(0)$ = t_{target} + s(l_0 - l_{00})

s = winch drive stiffness

l_{00} = original value of l_0 at the start of the simulation (set by static analysis)

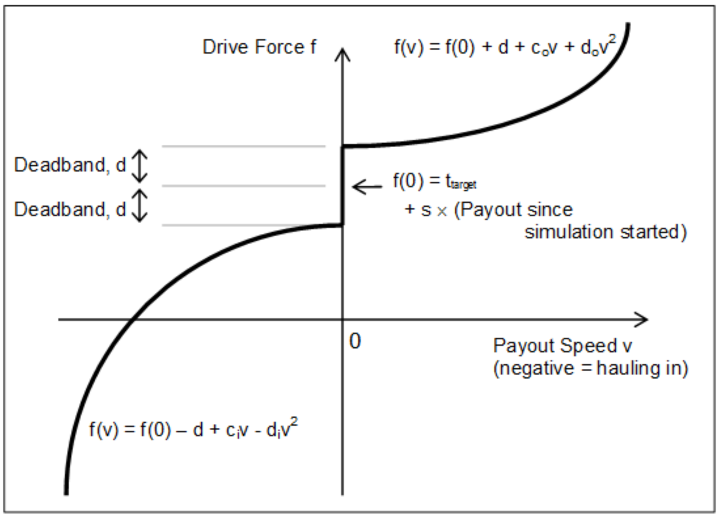


Figure 5.3: Tension-control operation for detailed winches

5.1.1. P-control systems assessment

Three different control systems are tested. One control system uses a combination of linear and quadratic forcing ($c_i v$ $d_i v^2$), making it a quadratic control system. The second case only uses the linear part ($c_i v$), making it a linear control system. The third case uses only quadratic forcing ($d_i v^2$), making it a pure quadratic control system. To fairly compare the three cases, the theoretical maximum and minimum total tension in the winch lines is kept the same. To further elaborate, for the first case, reasonable values are chosen for the $c_i v$ and d_i. The model is tested and the corresponding maximum velocity and minimum velocity and corresponding minimum and maximum tensions are calculated. The second case only contains the $c_i v$ term, however the contribution of the d_i on the max./min. tension is added to the c_i term (while keeping the corresponding velocity the same). For the third case exactly the same thing is done, only now the contribution from the c_i term is added to the d_i term. Leading to the following three damping functions (see figure 5.4). The model is loaded with head on airy waves with a wave height of 3 meter and period of 5.5 seconds. The three cases are tested for three different T_{target} for the deck winches. Namely, T_{target}= 750 kN, T_{target}=1000 kN and T_{target}=1250 kN. While keeping the T_{target} for boom at its maximum of 220 kN. The input parameters and results are shown in table 5.5.

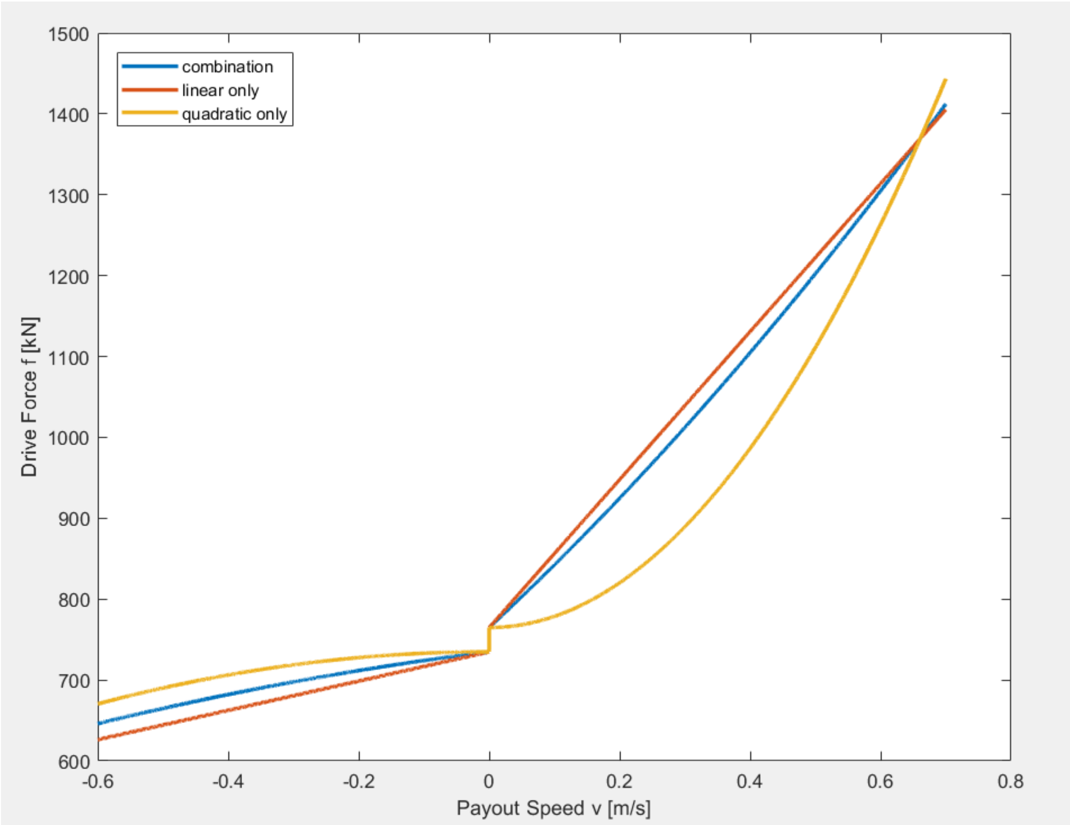


Figure 5.4: The damping functions for the three different cases

The results show that a higher contribution of the linear term leads to lower maximum tension in the winches. This is caused by the linear term already applying higher forcing at lower velocities compared to the quadratic or combination system, as shown in figure 5.4. Consequently, the “quadratic only” system causes the lowest minimum tension due to higher forcing at negative (hauling in) velocities. The maximum vertical forcing on the crane experiences little change for the different control systems but us lowest for the linear system. Whereas the maximum offlead forcing on the crane is the lowest with the quadratic only system. Caused by the lower average forcing of mainly the boom winches in offlead direction. Consequently, the lower control of the jacket causes the sidelead load to be the highest for the quadratic variant and lowest for the linear variant. Overall the results show that a control system with a purely linear dependency on the winch speed scores best and the purely quadratic system scores worst. Logically the combination scores in between both outer cases.

Looking at the data for different "target tensions" (pre-tension) shows roughly the same relative results. The higher target tension of the deck winches relieves the boom winches and result into lower minimum and maximum tensions. Obviously the minimum and maximum tension of the deck winches increases together with the offlead force. The sidelead load decreases slightly due to higher total forcing and better control of the jacket. Note that a higher target tension for the deck winches also results into a higher vertical load on the crane tip. As the deck winches also have a downwards vertical component when pulling on the jacket. This limits the maximum tension a deck winch can apply.

	Deck winch target tension [kN]	Boom winch target tension [kN]	Linear haul in Boom winches [kN]	Linear pay out Boom winches[kN]	Quadratic haul in Boom winches[kN]	Quadratic pay out Boom winches [kN]	Linear haul in Deck winches [kN]	Linear pay out Deck winches[kN]	Quadratic haul in Deck winches[kN]	Quadratic pay out Deck winches [kN]	Min Tension Tugger Boom winches [kN]	Max Tension Tugger Boom winches [kN]	Min Tension Tugger Deck winches [kN]	Max Tension Tugger Deck winches [kN]	Vertical load Untiy check [-]	Aframe Offlead Load UC [-]	Aframe Sidelead Load UC [-]
Combination	750	220	100	750	80	250	300	750	150	250	88	818	260	1585	0.911	0.65	0.84
Linear			181	915	0	0	467	980	0	0	80	788	253	1581	0.909	0.65	0.83
Quadratic			0	0	179	1385	0	0	419	1066	74	902	239	1592	0.914	0.59	0.90
Combination	1000		100	750	80	250	300	750	150	250	86	807	518	1812	0.920	0.71	0.84
Linear			181	915	0	0	521	926	0	0	79	772	474	1776	0.917	0.70	0.82
Quadratic			0	0	179	1385	0	0	353	1318	66	882	546	1878	0.928	0.67	0.89
Combination	1250		100	750	80	250	300	750	150	250	84	796	774	2050	0.932	0.76	0.85
Linear			181	915	0	0	569	866	0	0	79	762	699	1984	0.927	0.75	0.81
Quadratic			0	0	179	1385	0	0	317	1873	58	830	847	2223	0.943	0.73	0.88

Figure 5.5: Results of model tests for the linear, quadratic and combination systems

5.1.2. P-controller optimization

With the newly gained insight, a purely linear damping function is optimized. By varying each parameter individually the best combinations are found. The most effective combination is considered the combination which decreases the crane tip forces the most within the set tension limits for the winches. Note that the winch drive stiffness term is kept zero, as this adds a integral term to the control system, and this section only focuses on p-control systems. The optimized input parameters are shown in table 5.1.

	Boom winches	Deck winches	Unit
T_{target}	220	980	kN
c_i	100	800	kN/m
d	10	10	kN

Table 5.1: Winch input parameters for optimized linear damping function

5.1.3. PID controller Orcaflex model

Winches in Orcaflex also allow for external functions to determine the output tension. A PID controller is designed with the use of the program python. The Python scripts can be found in appendix F. The PID controller works on the same principle as the one describes for the matlab model in section 4.3. Instead of only one winch, the model has six winches. Each winch works independently, meaning each winch will reacts on the velocity of their winch line. When the simulation starts, Orcaflex sends the velocity term of every winch together with initial conditions to the Python script. The Python script calculated the tension as shown in equation 4.1. Where after these values are "filtered" and send back to orcaflex. This process is repeated for every timestep. For the tuning of the PID constants, the five best PID weight factor ratio's are taken from the PID tune results of the 2D model (tabel 4.2). These are combinations:

Controller 1: 21
Controller 2: 14
Controller 3: 15
Controller 4: 11
Controller 5: 23

The ratio's are scaled for both the boom winches and deck winches. Each combination is tested in a 120 seconds simulation with head on airy waves, calculating the crane tip forces. The input weight factors with resulting crane tip forces are shown in table 5.2.

	Boom winch			Deck winch						
Controller	$\mathbf{k}_P$	$\mathbf{k}_I$	$\mathbf{k}_D$	$\mathbf{k}_P$	$\mathbf{k}_I$	$\mathbf{k}_D$	**Vertical load**	**Offlead**	**Sidelead**	**unit**
1	300	150	300	60	30	60	-20420	-1892	-528	kN
2	500	50	150	100	10	30	-20735	-1991	-627	kN
3	500	150	50	100	30	10	-20833	-2036	-714	kN
4	500	167	167	100	33	33	-20455	-1926	-599	kN
5	500	230	360	100	45	72	**-19933**	**-1756**	**-467**	kN

Table 5.2: PID tuning results

The results show controller 5 as the most effective, then controller 1, 4, 2 and lastly controller 3. Comparing this with the results from the 2D PID tuning shows exactly the same results. This confirms the 2D model is a right representation for the 3D model, and can be used to rightly predict the order of effectiveness of different weight factor combinations.

5.2. Airy wave simulations

In this section the influence of the linear control systems from section **??** and PID control system from chapter 5.1.3 are compared to eachother and the situation without tugger damping. The models are loaded with 5.5 second period airy wave in 120 s simulation. First 60 seconds the wave height will slowly build up until at 60 seconds the wave height is at his maximum of 3.30m. Where after the wave will continue to load the system for the remaining 60 seconds. The results from the simulation are shown in table 5.3. Note that a negative or positive value determines the direction of the force.

The results show the maximum vertical force is the highest for the linear control system. It is slightly higher than without tuggers. This could be due to the forcing from the deck winches downwards, however as the PID does not suffer from lower vertical forces it should be a combination between larger motions and the winch forcing. The PID significantly reduces the vertical motions on the crane. The maximum offlead forces for the linear and PID models points toward the vessel. Which is as expected, as the winches are only able to pull towards the ship creating the largest forces on the crane in this direction. The forcing from the winches is also the reason for the maximum offlead forces being the largest for the models with tuggers. The maximum sidelead force for the linear and PID models points opposite to the wave direction. As the waves push the jacket away, the deck winches try to hold it in place, pulling on both the jacket and crane. For the no tugger model the maximum sidelead forces for both the negative or positive direction are very similar. What stand out is the large offlead force compared to the PID model and no tugger model, this is also shown by the large pitch motion of the linear model (figure 5.6 compared to the PID model (figure 5.7) and no tugger model (figure 5.8). What causes this is mode [1], which is a pitch pitch mode out of phase. As the linear is a proportional only control system is does not prematurely lower the tension by a derivative term. This causes a overshoot of the motion. As the deck winches engages the jacket from both sides it results in the winches are actually feeding the mode motion, leading to a larger roll motions and offlead forces. The maximum and minimum tension for the boom winches are comparable for both models, as both models use the maximum of these winches. However the maximum tension for the deck winches differ considerably. This is caused by the derivative term in PID by prematurely lowering the tension leading to lower maximum tensions. Additionally the more effective PID controller causes smaller velocities resulting in even lower winch tensions. The results show the PID control system a superior control system compared to the linear control system. The same conclusion was made with the 2D model. However unlike the 2D model, the no tugger scores worst for vertical and sidelead loads, and although it has a lower unity check (UC) for offlead direction the maximum offlead force is higher.

	Crane tip force			Boom winch tension		Deck winch tension		
Control system:	**Max. Vert.**	**Max. Off.**	**Max. Sid.**	**Max**	**Min**	**Max**	**Min**	**Unit**
Linear	21852	-2365	-1299	410	7	1980	3	kN
PID	19933	-1756	-467	418	28	1471	497	kN
No tuggers	21787	1554	994	-	-	-	-	kN
Reduction PID compared to linear [%]								
PID	9	26	64	-2	-300	26	-16467	%
UC [-]								
Linear	0.96	1.35	1.09	0.93	-	0.99	-	-
PID	0.87	1.00	0.39	0.95	-	0.74	-	-
No tuggers	0.95	1.95	0.83	-	-	-	-	-

Table 5.3: Overview results of Linear and PID control system compared to no tugger model

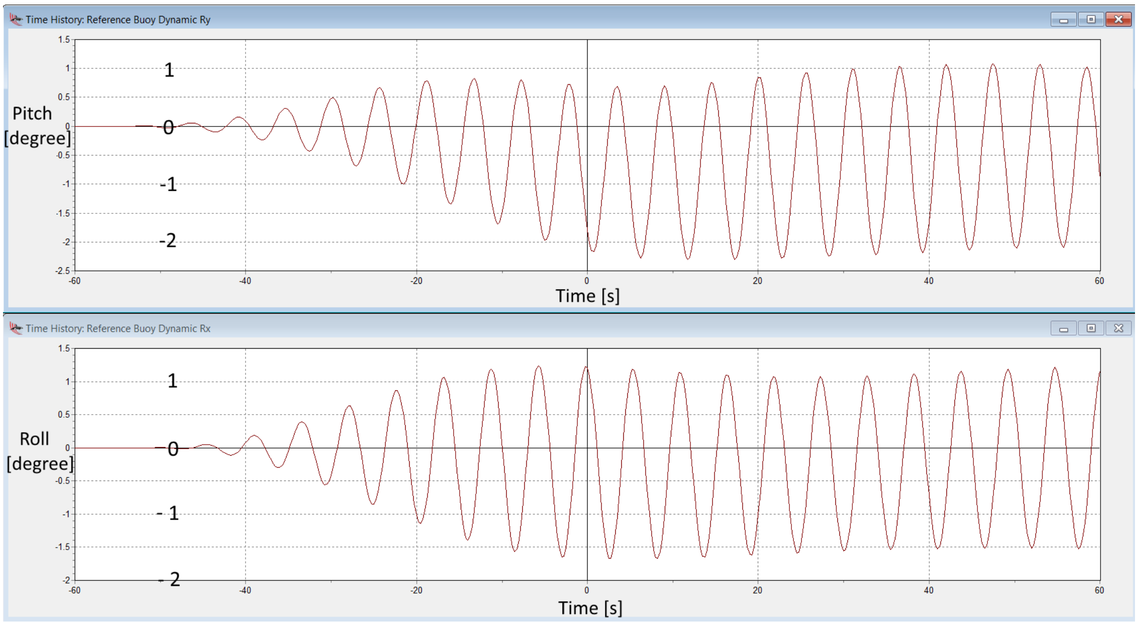


Figure 5.6: Roll and Pitch motion of the jacket for linear control system

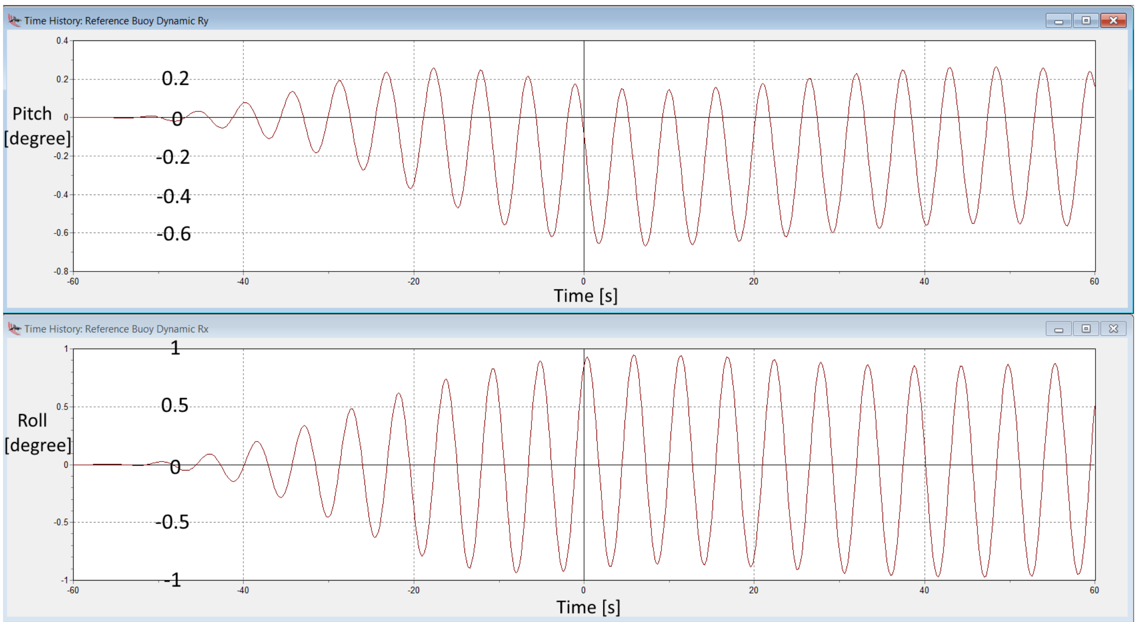


Figure 5.7: Roll and Pitch motion of the jacket for PID control system

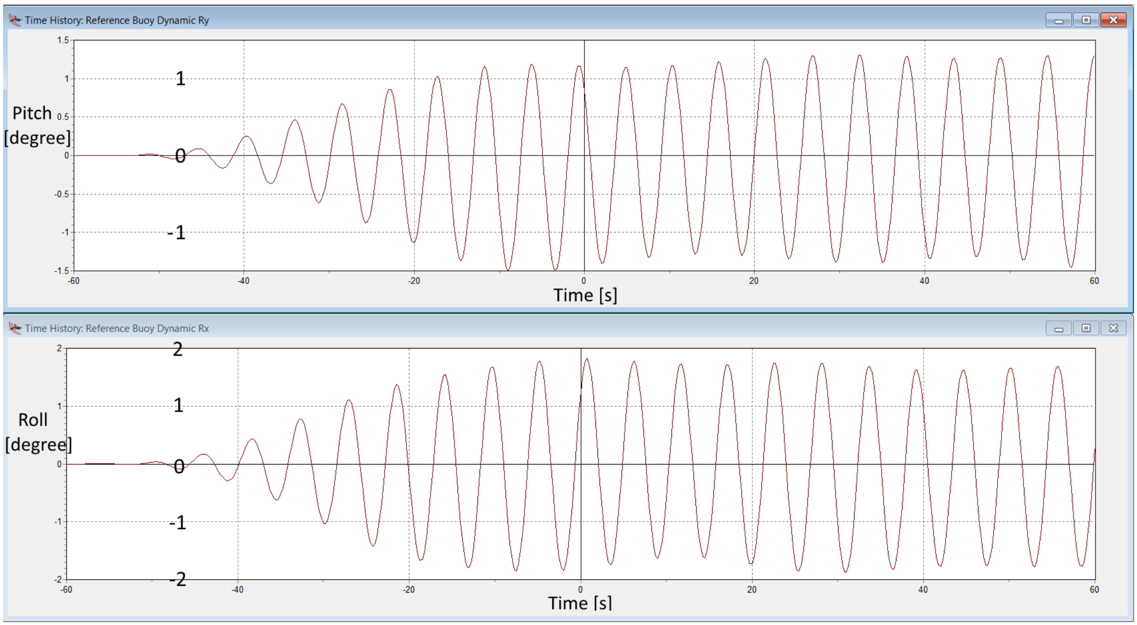


Figure 5.8: Roll and Pitch motion of the no tugger system

5.3. Discussion

Control systems are now analyzed in a 3D model. The systems are tested for their effect on the forces acting on the crane tip with time-domain simulations whil loaded by airy waves. The modal analysis together with wave loading tests showed that for both the crane tip forces and the winch forces the highest forces are induced at wave load periods around 5.5 seconds. Around this frequency a combination of modes are activated, resulting in large motions of the system.

The parameters of the control systems are scaled and tuned for the 3D model. The order of effectiveness of different PID weight factors combinations is the same for both the 2D and 3D model. Showing the relevance of the 2D model in testing the order off effectiveness the damping systems.

Using the predefined Orcaflex winch model the effectiveness of linear control systems is tested against quadratic control system. It shows that linear control system decreases the maximum vertical load and sidelead load on the crane tip most effectively. This is due to the higher forcing the linear control system applies at lower velocities. However, the higher forcing of the linear system creates higher offlead forces. The strength of the crane for offlead forces pointed towards the crane are relatively high. Because of this, the UC for the offlead direction is highly unlikely to be the failing mechanism. As other failing mechanisms lay closer towards their limits. Concluding: the linear control systems is more effective in decreasing the maximum forces acting on the crane tip, the quadratic the least, and the combination scores in between. As a result only a linear control system is optimized for further analysis.

Testing both the linear control system against the system without tugger damping shows that the system actually increases the maximum forces acting on the crane tip. Indicating the linear control system is unstable for this wave excitation, as motions of the lifting configuration combined with the forcing of the tuggers cause larger crane tip forces. The only difference between the models being the direction of the maximum offlead force. Which as mentioned earlier has greater strength in the negative direction (towards the crane). Resulting in only a lower UC for the offlead direction when using a tugger damping system with a linear control system.

Comparing the PID controller against a model without a tugger system shows that the PID control significantly decreases the crane tip forces. It has a positive effect on every set failure mechanisms. Compared to the linear system the winch forces are drastically lower for the PID control, while the force reductions are higher. This allows for smaller deck winches then the set limit of 2000 mt. The model including PID control is the only one which passes all unity checks, and as the model is loaded with airy waves with wave periods which specifically induce the highest forces. It is shown that with the PID controller the maximum operational wave height of 3.3m can be achieved.

6
Conclusion and recommendation

Monohull lifting vessels experience large motions during heavy lifting operations, resulting in low work-ability of the vessels.

During heavy lifting the lifting configuration can experience large motions from environmental loading, resulting in low work-ability of the heavy lifting vessel. The thesis investigated the option of reducing the motion by the means of applying damping to the system by the use of tugger winches. The main focus of the thesis lies at the effectiveness of the control systems which drive the winches. With extra emphasis on the more complex PID controller. The conclusion made for the thesis are treated in section 6.0.1. Recommendations for further research are given in section 6.0.2

6.0.1. Conclusion

To enable quick analysis for the effect of the control systems a 2D mass-spring-damper-system is made. The model is validated with a Orcaflex model. The model is used to show the order of effectiveness of different which control systems. Six different p-control systems and an optimized PID control system are designed. The PID is tuned model based. 25 different combination of PID weight factors are tested on the model which is loaded by waves. For this specific loading case a combination of large proportional and derivative weight factor works best for the model. The steady state error is most effectively reduced with the integral weight factor roughly half the value of the proportional term.

The different control models are firstly tested in two different ways using the 2D model. Firstly the maximum roll motion of the jacket is calculated when the model is experiencing wave loading with a wave frequency close to the pendulum mode of the lifting configuration. Secondly a decay test for the lifting configurations. The results from the first test show the PID as the most effective control system. It is able to reduce the maximum roll motion by almost 25 %. Comparing the results for the linear and quadratic control systems shows that linear control systems perform better. The different linear controllers show that applying zero pretension has a positive effect on the maximum roll motion. Where as high pretensions cause large roll motions due to high forcing at the lower velocities.

Additionally, the model is loaded with a wave frequency of 5 seconds, to test if the order of effectiveness of each control system is the same for a different load frequency. The results showed that this is not the case. The tuning does not apply for all frequencies, and will require different tuning for different wave frequencies. Testing the different control systems on a decay test show that systems containing a deadband scored best on damping the system. Already a small deadband included in the quadratic db control system drastically decreased the decay time. While the huge swing in force when passing zero velocity has a negative effect on the roll motion of the load, it is very effective for damping out an initial displacement. P-control systems can not damp more effective then the stepwise. Where as the PID is able to tune the lifting configuration even faster due to the additional I and D term.

The linear, quadratic and PID control systems are further analyzed in a more detailed Orcaflex model. The model is used to compare the different control systems, and its effects compared to a model without a tugger system. Often the limiting factor for heavy lifting is the the exceedance of the crane tip forces. The modal analysis shows four modes that fall in the wave excitation range that are of importance to the model. The model experiences the highest forces when loaded with 5.5s period waves. As a combination of modes results in large motions. P-control systems assessment showed that a control system only containing a term linear to the velocity leads to lower sidelead forces compared to a control system with only a quadratic term or a control system which contain both. Where as the offlead forces is the highest for the linear and lowest for the quadratic control system. The strength of the crane in sidelead direction is much lower compared to the offlead strength. Hence, it is fair to state that reduction of the sidelead forces is of higher importance. Resulting in the linear control system considered the prefered control system.

Tuning the PID controller for the 3D model showed that the best performing weight factor ratios from the 2D analysis also perform best in a 3D model. The airy wave simulations show that using a linear control system result in higher crane tip forces compared to a no tugger model. It does redirect the maximum offlead forcing towards a more favorable direction. Indicating the linear control system is unstable for this wave excitation. The motions of the lifting configuration combined with the forcing of the tuggers cause larger crane tip forces. The PID reduces the vertical and sidelead forces and also redirects the maximum offlead forcing resulting in a lower UC than the no tugger model. Compared to the linear it is 9% lower forcing in vertical direction, 26% lower in offlead direction, and 64% lower forcing in sidelead direction. Additionally the PID only uses a maximum of 1471 kN pulling force of the 2000 kN deck winches.

In short: Winch tuggers do not neccessarily have a possitive effect on the lifting systen, and can even create larger forces than without. The PID control system dampens the system the effective and the most efficient, compared to P-control systems. However, PID controller only works effective around specific wave loading frequency for which it is tuned. Although PID control system looks like the best choice, it depends on what you are looking for within the lifting configuration. As results vary per control system per loading frequency.

6.0.2. Recommendations

- Model based control designs: Instead of using a ordinary PID controller, use a model based control design. Which have an advantage in controlling more complex systems and have a better global system stability. These control system are more complex to create, however unlike the PID controller are less affected by changes of the model. [13]
- Include actuator and sensor errors: Within the thesis the tugger damping system behaved perfect. It worked without errors or delays. In reality this is obviously not the case. Further research is required on the effects of the delay of the actuator, actuator output errors, sensor errors, or sensor noise. [8]
- Alternative winch drive parameter: The thesis works with winches which are velocity driven. The winches measure the velocity of their own winch lines. It could be beneficial to have winches which receive input from the jacket directly. Such as wi-fi connected accelerators on the jacket.
- Vector control damping: The winches act independently from each other. It is an option to let them communicate and calculate the optimal required total vector for the largest damping effect. Where the control system calculate the most effective forcing combination.
- Different tugger configurations: For the thesis the winch configuration was limited to the 4 already existing boom tuggers and 2 deck tuggers. However further research is required on the optimal amount of tuggers and placement of these tuggers. Or change the angles of approach of the winch lines by the use of certain arms that redirect the lines.
- Extra extensions into PID controller: research the effect of adding in extra terms in the PID controller. Such as a term which prematurely predict the motions of the whole vessel based on the wave climate surrounding the vessel.

A

Maple files

Lagrange of 5 DOF

```
> restart
```

Kinematic Equations

```
> x_m(t) := c + x_s(t) + theta_k(t)*a + 1/2*b*theta_m(t) - 1/2*c*theta_s(t)^2 - theta_s(t)*d:
> y_m(t) := e + y_s(t) + 1/2*a*theta_k(t)^2 + 1/2*b*theta_m(t)^2 + c*theta_s(t) - 1/2*d*theta_s(t)^2:
> x_m_dot := diff(x_m(t),t):
> y_m_dot := diff(y_m(t),t):
> y_s(t) := y_s(t):
> x_s(t) := x_s(t):
> x_tug(t) := c + x_s(t) + theta_k(t)*a - 1/2*c*theta_s(t)^2 - theta_s(t)*d:
> y_tug(t) := (d - a) + y_s(t) + 1/2*a*theta_k(t)^2 + c*theta_s(t) - 1/2*d*theta_s(t)^2:
> x_tug_dot := diff(x_tug(t),t):
> y_tug_dot := diff(y_tug(t),t):
> x_s_tug(t) := B_ship+x_s(t)+ theta_s(t)*H_ship - 1/2*B_ship*theta_s(t)^2:
> y_s_tug(t) := H_ship + y_s(t) + theta_s(t)*B_ship - 1/2*H_ship*theta_s(t)^2:
> x_s_tug_dot := diff(x_s_tug(t),t):
> y_s_tug_dot := diff(y_s_tug(t),t):
> theta_s(t) := theta_s(t):
> theta_s_dot := diff(theta_s(t),t):
> y_s_dot := diff(y_s(t),t):
> x_s_dot := diff(x_s(t),t):
> L_t := sqrt((x_tug(t)-x_s_tug(t))^2+(y_tug(t)-y_s_tug(t))^2):
> V_t := diff(L_t,t):
> CodeGeneration['Matlab']( V_t );
Warning, the function names {diff, theta_k, theta_s} are not recognized in the target language
cg = ((c + theta_k(t) * a - c * theta_s(t) ^ 2 / 0.2e1 - theta_s(t) * d - B_ship - theta_s(t) * H_ship + B_ship * theta_s(t) ^ 2 / 0.2e1) ^ 2 + (d - a + a * theta_k(t) ^ 2 / 0.2e1 + c * theta_s(t) - d * theta_s(t) ^ 2 / 0.2e1 - H_ship - theta_s(t) * B_ship + H_ship * theta_s(t) ^ 2 / 0.2e1) ^ 2) ^ (-0.1e1 / 0.2e1) * (0.2e1 * (c + theta_k(t) * a - c * theta_s(t) ^ 2 / 0.2e1 - theta_s(t) * d - B_ship - theta_s(t) * H_ship + B_ship * theta_s(t) ^ 2 / 0.2e1) * (diff(theta_k(t), t) * a - c * theta_s(t) * diff(theta_s(t), t) - diff(theta_s(t), t) * d - diff(theta_s(t), t) * H_ship + B_ship * theta_s(t) * diff(theta_s(t), t)) + 0.2e1 * (d - a + a * theta_k(t) ^ 2 / 0.2e1 + c * theta_s(t) - d * theta_s(t) ^ 2 / 0.2e1 - H_ship - theta_s(t) * B_ship + H_ship * theta_s(t) ^ 2 / 0.2e1) * (a * theta_k(t) * diff(theta_k(t), t) + c * diff(theta_s(t), t) - d * theta_s(t) * diff(theta_s(t), t) - diff(theta_s(t), t) * B_ship + H_ship * theta_s(t)
```

```
* diff(theta_s(t), t))) / 0.2e1;
> A_t := diff(V_t,t):
> CodeGeneration['Matlab']( A_t );
Warning, the function names {diff, theta_k, theta_s} are not
recognized in the target language
cg5 = -((c + theta_k(t) * a - c * theta_s(t) ^ 2 / 0.2e1 -
theta_s(t) * d - B_ship - theta_s(t) * H_ship + B_ship *
theta_s(t) ^ 2 / 0.2e1) ^ 2 + (d - a + a * theta_k(t) ^ 2 /
0.2e1 + c * theta_s(t) - d * theta_s(t) ^ 2 / 0.2e1 - H_ship
- theta_s(t) * B_ship + H_ship * theta_s(t) ^ 2 / 0.2e1) ^
2) ^ (-0.3e1 / 0.2e1) * (0.2e1 * (c + theta_k(t) * a - c *
theta_s(t) ^ 2 / 0.2e1 - theta_s(t) * d - B_ship - theta_s
(t) * H_ship + B_ship * theta_s(t) ^ 2 / 0.2e1) * (diff
(theta_k(t), t) * a - c * theta_s(t) * diff(theta_s(t), t) -
diff(theta_s(t), t) * d - diff(theta_s(t), t) * H_ship +
B_ship * theta_s(t) * diff(theta_s(t), t)) + 0.2e1 * (d - a
+ a * theta_k(t) ^ 2 / 0.2e1 + c * theta_s(t) - d * theta_s
(t) ^ 2 / 0.2e1 - H_ship - theta_s(t) * B_ship + H_ship *
theta_s(t) ^ 2 / 0.2e1) * (a * theta_k(t) * diff(theta_k(t),
t) + c * diff(theta_s(t), t) - d * theta_s(t) * diff(theta_s
(t), t) - diff(theta_s(t), t) * B_ship + H_ship * theta_s(t)
* diff(theta_s(t), t))) ^ 2 / 0.4e1 + ((c + theta_k(t) * a -
c * theta_s(t) ^ 2 / 0.2e1 - theta_s(t) * d - B_ship -
theta_s(t) * H_ship + B_ship * theta_s(t) ^ 2 / 0.2e1) ^ 2 +
(d - a + a * theta_k(t) ^ 2 / 0.2e1 + c * theta_s(t) - d *
theta_s(t) ^ 2 / 0.2e1 - H_ship - theta_s(t) * B_ship +
H_ship * theta_s(t) ^ 2 / 0.2e1) ^ 2) ^ (-0.1e1 / 0.2e1) *
(0.2e1 * (diff(theta_k(t), t) * a - c * theta_s(t) * diff
(theta_s(t), t) - diff(theta_s(t), t) * d - diff(theta_s(t),
t) * H_ship + B_ship * theta_s(t) * diff(theta_s(t), t)) ^ 2
+ 0.2e1 * (c + theta_k(t) * a - c * theta_s(t) ^ 2 / 0.2e1 -
theta_s(t) * d - B_ship - theta_s(t) * H_ship + B_ship *
theta_s(t) ^ 2 / 0.2e1) * (diff(diff(theta_k(t), t), t) * a
- c * diff(theta_s(t), t) ^ 2 - c * theta_s(t) * diff(diff
(theta_s(t), t), t) - diff(diff(theta_s(t), t), t) * d -
diff(diff(theta_s(t), t), t) * H_ship + B_ship * diff
(theta_s(t), t) ^ 2 + B_ship * theta_s(t) * diff(diff
(theta_s(t), t), t)) + 0.2e1 * (a * theta_k(t) * diff
(theta_k(t), t) + c * diff(theta_s(t), t) - d * theta_s(t) *
diff(theta_s(t), t) - diff(theta_s(t), t) * B_ship + H_ship
* theta_s(t) * diff(theta_s(t), t)) ^ 2 + 0.2e1 * (d - a + a
* theta_k(t) ^ 2 / 0.2e1 + c * theta_s(t) - d * theta_s(t) ^
2 / 0.2e1 - H_ship - theta_s(t) * B_ship + H_ship * theta_s
(t) ^ 2 / 0.2e1) * (a * diff(theta_k(t), t) ^ 2 + a *
theta_k(t) * diff(diff(theta_k(t), t), t) + c * diff(diff
(theta_s(t), t), t) - d * diff(theta_s(t), t) ^ 2 - d *
theta_s(t) * diff(diff(theta_s(t), t), t) - diff(diff
(theta_s(t), t), t) * B_ship + H_ship * diff(theta_s(t), t)
^ 2 + H_ship * theta_s(t) * diff(diff(theta_s(t), t), t))) /
0.2e1;
```

Energy Equations0

Kinetic energy

```
> T := 1/2*m*(x_m_dot^2 + y_m_dot^2) + 1/2*M*(y_s_dot^2 +
  x_s_dot^2) + 1/2*I_m*diff(theta_m(t),t)^2 + 1/2*I_M*diff
```

```
(theta_s(t),t)^2:
> simplify(expand(T)):
```

Potential energy

```
> V := 1/2*k_sv*y_s(t)^2 + 1/2*k_sh*x_s(t)^2 + 1/2*k_sr*
  theta_s(t)^2 + M*g*y_s(t) + m*g*y_m(t):
```

Work by external force

```
> W := frac_x_tug * F_tug(t) * x_s_tug(t) + frac_y_tug *
  F_tug(t) * y_s_tug(t) - frac_x_tug * F_tug(t) * x_tug(t) -
  frac_y_tug * F_tug(t) * y_tug(t) + x_m(t) * F_m(t) + y_s
  (t)*F_sv(t) + x_s(t)*F_sh(t) + theta_s(t)*M_s
```

$$W := frac\_x\_tug\, F\_tug(t) \left(B\_ship + x\_s(t) + theta\_s(t)\, H\_ship - \frac{B\_ship\, theta\_s(t)^2}{2}\right) + frac\_y\_tug\, F\_tug(t) \left(H\_ship + y\_s(t) + theta\_s(t)\, B\_ship - \frac{H\_ship\, theta\_s(t)^2}{2}\right) - frac\_x\_tug\, F\_tug(t) \left(c + x\_s(t) + theta\_k(t)\, a - \frac{c\, theta\_s(t)^2}{2} - theta\_s(t)\, d\right) - frac\_y\_tug\, F\_tug(t) \left(d - a + y\_s(t) + \frac{a\, theta\_k(t)^2}{2} + c\, theta\_s(t) - \frac{d\, theta\_s(t)^2}{2}\right) + \left(c + x\_s(t) + theta\_k(t)\, a + \frac{b\, theta\_m(t)}{2} - \frac{c\, theta\_s(t)^2}{2} - theta\_s(t)\, d\right) F\_m(t) + y\_s(t)\, F\_sv(t) + x\_s(t)\, F\_sh(t) + theta\_s(t)\, M\_s \tag{1.2.3.1}$$

Construct the Lagrangian

```
> L := T - V - W:
```

Damping in system

```
> D_d := 1/2*c_sh*x_s_dot^2 + 1/2*c_sv*y_s_dot^2 + 1/2*c_sr*
  theta_s_dot^2:
```

Obtaining the EOM

```
> with(Physics):
> EOM_theta_k := diff( diff(L, diff(theta_k(t), t)), t) + diff
  (D_d, diff(theta_k(t), t)) - diff(L,theta_k(t)):
> EOM_theta_m := diff( diff(L, diff(theta_m(t), t)), t) + diff
  (D_d, diff(theta_m(t), t)) - diff(L,theta_m(t)):
> EOM_theta_s := diff( diff(L, diff(theta_s(t), t)), t) + diff
  (D_d, diff(theta_s(t), t)) - diff(L,theta_s(t)):
> EOM_x_s := diff( diff(L, diff(x_s(t), t)), t) + diff(D_d,
  diff(x_s(t), t)) - diff(L,x_s(t)):
> EOM_y_s := diff( diff(L, diff(y_s(t), t)), t) + diff(D_d,
```

```
diff(y_s(t), t)) - diff(L,y_s(t)):
```

Obtaining the EOMs for nDOF system

```
> with(LinearAlgebra):
> MTRX := GenerateMatrix([EOM_theta_k, EOM_theta_m,EOM_theta_s,
  EOM_x_s,EOM_y_s], [diff(theta_k(t),t$2), diff(theta_m(t),
  t$2),diff(theta_s(t),t$2),diff(x_s(t),t$2),diff(y_s(t),t$2)])
  :
> M_matrix := MTRX[1]:
> M_matrix := simplify(expand(M_matrix));
```

$$M\_matrix := \left[\left[m\,a^2\left(theta\_k(t)^2+1\right), \frac{m\,b\,a\,(2\,theta\_k(t)\,theta\_m(t)+1)}{2}, a\,m\,((c - theta\_s(t)\,d)\,theta\_k(t) - c\,theta\_s(t) - d), m\,a, m\,theta\_k(t)\,a\right], \left[\frac{m\,b\,a\,(2\,theta\_k(t)\,theta\_m(t)+1)}{2}, \frac{b^2\,m}{4} + theta\_m(t)^2\,b^2\,m + I\_m, -\frac{m\,b\,(2\,theta\_s(t)\,theta\_m(t)\,d + c\,theta\_s(t) - 2\,theta\_m(t)\,c + d)}{2}, \frac{m\,b}{2}, m\,theta\_m(t)\,b\right], \left[a\,m\,((c - theta\_s(t)\,d)\,theta\_k(t) - c\,theta\_s(t) - d), -\frac{m\,b\,(2\,theta\_s(t)\,theta\_m(t)\,d + c\,theta\_s(t) - 2\,theta\_m(t)\,c + d)}{2}, m\left(c^2 + d^2\right)theta\_s(t)^2 + m\left(c^2+d^2\right) + I\_M, -m\,(c\,theta\_s(t) + d), m\,(c - theta\_s(t)\,d)\right], \left[m\,a, \frac{m\,b}{2}, -m\,(c\,theta\_s(t)+d), M+m, 0\right], \left[m\,theta\_k(t)\,a, m\,theta\_m(t)\,b, m\,(c - theta\_s(t)\,d), 0, M+m\right]\right] \quad (1.6.1)$$

This F contains the K, C and F matrix

```
> F_matrix := MTRX[2]:
> F_matrix := simplify(expand(F_matrix))
```

$$F\_matrix := \left[\left[-a\left(-m\,(theta\_k(t)\,d + c)\left(\frac{d}{dt}\,theta\_s(t)\right)^2 + theta\_k(t)\left(\frac{d}{dt}\,theta\_k(t)\right)^2 a\,m + theta\_k(t)\left(\frac{d}{dt}\,theta\_m(t)\right)^2 b\,m + (-frac\_y\_tug\,F\_tug(t) + m\,g)\,theta\_k(t) - frac\_x\_tug\,F\_tug(t) + F\_m(t)\right)\right], \left[-\left(-\left(theta\_m(t)\,d + \frac{c}{2}\right)m\left(\frac{d}{dt}\,theta\_s(t)\right)^2 + \left(\frac{d}{dt}\,theta\_k(t)\right)^2 theta\_m(t)\,a\,m + \left(\frac{d}{dt}\,theta\_m(t)\right)^2 theta\_m(t)\,b\,m + m\,g\,theta\_m(t)\right. \right. \quad (1.6.2)$$

$$+\frac{F\_m(t)}{2}\Big)\,b\Big],$$

$$\Big[-theta\_s(t)\,m\,(c^2+d^2)\left(\frac{d}{dt}\,theta\_s(t)\right)^2-c\_sr\left(\frac{d}{dt}\,theta\_s(t)\right)$$
$$+a\,m\,(theta\_s(t)\,d-c)\left(\frac{d}{dt}\,theta\_k(t)\right)^2+m\,b\,(theta\_s(t)\,d-c)\left(\frac{d}{dt}\right.$$
$$\left.theta\_m(t)\right)^2+((B\_ship\,frac\_x\_tug+H\_ship\,frac\_y\_tug-c\,frac\_x\_tug$$
$$-d\,frac\_y\_tug)\,F\_tug(t)+d\,g\,m+F\_m(t)\,c-k\_sr)\,theta\_s(t)+($$
$$-B\_ship\,frac\_y\_tug-H\_ship\,frac\_x\_tug+c\,frac\_y\_tug-d\,frac\_x\_tug)\,F\_tug(t)$$
$$-c\,g\,m+F\_m(t)\,d-M\_s\Big],$$

$$\left[m\,c\left(\frac{d}{dt}\,theta\_s(t)\right)^2-k\_sh\,x\_s(t)-c\_sh\left(\frac{d}{dt}\,x\_s(t)\right)-F\_sh(t)-F\_m(t)\right],$$

$$\Big[\left(\frac{d}{dt}\,theta\_s(t)\right)^2d\,m-\left(\frac{d}{dt}\,theta\_k(t)\right)^2a\,m-\left(\frac{d}{dt}\,theta\_m(t)\right)^2b\,m-M\,g$$
$$-k\_sv\,y\_s(t)-c\_sv\left(\frac{d}{dt}\,y\_s(t)\right)-m\,g-F\_sv(t)\Big]\Big]$$

```
> MTRX_2 := GenerateMatrix([F_matrix[1], F_matrix[2],F_matrix
  [3],F_matrix[4],F_matrix[5]], [diff(theta_k(t),t$1), diff
  (theta_m(t),t$1),diff(theta_s(t),t$1),diff(x_s(t),t$1),diff
  (y_s(t),t$1)]):
> C_matrix := -1 * MTRX_2[1]:
```

This K contains the K and F matrix

```
> R := MTRX_2[2]:
> R_1 := expand(R[1]):
> R_2 := expand(R[2]):
> R_3 := expand(R[3]):
> R_4 := expand(R[4]):
> R_5 := expand(R[5]):
> f   := proc(X) if degree(X,{diff(theta_k(t), t),diff(theta_m
  (t), t),diff(theta_s(t), t),diff(x_s(t), t),diff(y_s(t), t)})
  >1 then 0 else X; end if; end proc:
> F_1 := map(f,R_1):
> F_2 := map(f,R_2):
> F_3 := map(f,R_3):
> F_4 := map(f,R_4):
> F_5 := map(f,R_5):
> MTRX_3 := GenerateMatrix([F_1, F_2,F_3,F_4,F_5], [theta_k(t),
  theta_m(t),theta_s(t),x_s(t),y_s(t)]):
> K := MTRX_3[1];
```

$$K:=[[-F\_tug(t)\,a\,frac\_y\_tug+a\,g\,m,0,0,0,0],\tag{1.6.3}$$
$$[0,m\,g\,b,0,0,0],$$
$$[0,0,-F\_tug(t)\,B\_ship\,frac\_x\_tug-F\_tug(t)\,H\_ship\,frac\_y\_tug$$
$$+F\_tug(t)\,c\,frac\_x\_tug+F\_tug(t)\,d\,frac\_y\_tug-d\,g\,m-F\_m(t)\,c+k\_sr,0,0],$$
$$[0,0,0,k\_sh,0],$$

$[0, 0, 0, 0, k\_sv]]$

```
> K := simplify(expand(K)):
> F_iso := -1 * MTRX_3[2]:
> F_iso := simplify(expand(F_iso));
```

$$F\_iso := \left[\left[a\left(-frac\_x\_tug\, F\_tug(t) + F\_m(t)\right)\right], \left[\frac{b\, F\_m(t)}{2}\right], \left[\left((d + H\_ship)\, frac\_x\_tug - frac\_y\_tug\,(c - B\_ship)\right) F\_tug(t) + c\, g\, m - F\_m(t)\, d + M\_s\right], \left[F\_sh(t) + F\_m(t)\right], \left[F\_sv(t) + (M + m)\, g\right]\right] \quad (1.6.4)$$

```
> CodeGeneration['Matlab']( M_matrix );
```

Warning, the function names {theta_k, theta_m, theta_s} are not recognized in the target language

```
cg0 = [m * a ^ 2 * (theta_k(t) ^ 2 + 1) (m * b * a * (2 * theta_k(t) * theta_m(t) + 1)) / 0.2e1 a * m * ((c - theta_s(t) * d) * theta_k(t) - c * theta_s(t) - d) m * a m * theta_k(t) * a; (m * b * a * (2 * theta_k(t) * theta_m(t) + 1)) / 0.2e1 (b ^ 2 * m) / 0.4e1 + (theta_m(t) ^ 2 * b ^ 2 * m) + I_m -(m * b * (2 * theta_s(t) * theta_m(t) * d + c * theta_s(t) - 2 * theta_m(t) * c + d)) / 0.2e1 (m * b) / 0.2e1 m * theta_m(t) * b; a * m * ((c - theta_s(t) * d) * theta_k(t) - c * theta_s(t) - d) -(m * b * (2 * theta_s(t) * theta_m(t) * d + c * theta_s(t) - 2 * theta_m(t) * c + d)) / 0.2e1 m * (c ^ 2 + d ^ 2) * theta_s(t) ^ 2 + m * (c ^ 2 + d ^ 2) + I_M -m * (c * theta_s(t) + d) m * (c - theta_s(t) * d); m * a (m * b) / 0.2e1 -m * (c * theta_s(t) + d) M + m 0; m * theta_k(t) * a m * theta_m(t) * b m * (c - theta_s(t) * d) 0 M + m;];
```

```
> CodeGeneration['Matlab']( F_matrix );
```

Warning, the function names {F_m, F_sh, F_sv, F_tug, diff, theta_k, theta_m, theta_s, x_s, y_s} are not recognized in the target language

```
cg1 = [-a * (-m * (theta_k(t) * d + c) * diff(theta_s(t), t) ^ 2 + theta_k(t) * diff(theta_k(t), t) ^ 2 * a * m + theta_k(t) * diff(theta_m(t), t) ^ 2 * b * m + (-frac_y_tug * F_tug(t) + m * g) * theta_k(t) - frac_x_tug * F_tug(t) + F_m(t)) -(-(theta_m(t) * d + c / 0.2e1) * m * diff(theta_s(t), t) ^ 2 + diff(theta_k(t), t) ^ 2 * theta_m(t) * a * m + diff(theta_m(t), t) ^ 2 * theta_m(t) * b * m + m * g * theta_m(t) + F_m(t) / 0.2e1) * b -theta_s(t) * m * (c ^ 2 + d ^ 2) * diff(theta_s(t), t) ^ 2 - c_sr * diff(theta_s(t), t) + a * m * (theta_s(t) * d - c) * diff(theta_k(t), t) ^ 2 + m * b * (theta_s(t) * d - c) * diff(theta_m(t), t) ^ 2 + ((B_ship * frac_x_tug + H_ship * frac_y_tug - c * frac_x_tug - d * frac_y_tug) * F_tug(t) + d * g * m + F_m(t) * c - k_sr) * theta_s(t) + (-B_ship * frac_y_tug - H_ship * frac_x_tug + c * frac_y_tug - d * frac_x_tug) * F_tug(t) - c * g * m + F_m(t) * d - M_s m * c * diff(theta_s(t), t) ^ 2 - k_sh * x_s
```

```
(t) - c_sh * diff(x_s(t), t) - F_sh(t) - F_m(t) diff(theta_s
(t), t) ^ 2 * d * m - diff(theta_k(t), t) ^ 2 * a * m - diff
(theta_m(t), t) ^ 2 * b * m - M * g - k_sv * y_s(t) - c_sv *
diff(y_s(t), t) - m * g - F_sv(t)];
> CodeGeneration['Matlab']( C_matrix );
cg2 = [0 0 0 0 0; 0 0 0 0 0; 0 0 c_sr 0 0; 0 0 0 c_sh 0; 0 0
0 0 c_sv;];
> CodeGeneration['Matlab']( K );
Warning, the function names {F_m, F_tug} are not recognized
in the target language
cg3 = [a * (-frac_y_tug * F_tug(t) + m * g) 0 0 0 0; 0 m * g
* b 0 0 0; 0 0 ((c - B_ship) * frac_x_tug + frac_y_tug * (d
- H_ship)) * F_tug(t) - d * g * m - F_m(t) * c + k_sr 0 0; 0
0 0 k_sh 0; 0 0 0 0 k_sv;];
> CodeGeneration['Matlab']( F_iso );
Warning, the function names {F_m, F_sh, F_sv, F_tug} are not
recognized in the target language
cg4 = [a * (-frac_x_tug * F_tug(t) + F_m(t)) b * F_m(t) /
0.2e1 ((d + H_ship) * frac_x_tug - frac_y_tug * (c - B_ship)
) * F_tug(t) + c * g * m - F_m(t) * d + M_s F_sh(t) + F_m(t)
F_sv(t) + (M + m) * g];
```

B
Matlab files

B.1. Controller main

```
clear all; clc; dbstop if error; close('all');clear global
variable;

%% Input

%Control system
CS= 1; %choose between control system 0,1,2,3.1,3.2,4,5 and
6
%With 0 being no control system, 3.1 shifted forward, 3.2
shifted backwards
%and 6 PID system

%Gen. info (for visualisation on the parameters see the
mass-spring-damper
%model)
g        = 9.81;
a        = 52.34;
b        = 84;
c        = 52.5;
d        = 138.3;
e        = d - a - 1/2*b;
m        = 1.73e6;
M        = 5.56e7;
I_m      = 1.263e9;                    % Cylinder:
I=1/4*m*r^2 = 1/12*m*4*b^4    % Beam: I=1/3*m*4*b^2
I_M      = 1.86e10;                    % Blok:
I=1/12*m*(b^2+l^2)

%size boat
H_ship = 4.5;    % half of the boat height
B_ship = 21;     % half of the boat width


%Duration & Timesteps
nt       = 1e3;                   %Amount of steps
T        = 500;                    %Total duration
N        = 5;                      %degree of freedoms
t        = linspace (0,T,nt);      %All steps


%Wave parameters
H_w        = 1;                                  % (m) Wave
height of airy wave
```

```
Lap_w      = 29 ;                              % (s) Wave
period of airy wave

%Forces
F_ma       = 0;                      %Force load, amplitude
F_mf       = 1;                      %Force load, wave period

F_sva      = 11.0e3;                      %(kN/m) Ver-Force
(Heave) ship, RAO amplitude
F_svf      = 196.7;                      %(deg) Ver-Force
ship, Phase

F_sha      = 15.8e3;                      %(kN/m) Hor-Force
(Sway) ship, RAO amplitude
F_shf      = 21.4;                      %(deg) Hor-Force ship,
Phase

M_sa      = 73.3e3;                      %(kN/m) Moment ship,
RAO amplitude
M_sf      = 187.6;                      %(deg) Moment ship,
Phase

%Only for visualizing the forcing
F_m        = F_ma * H_w * sin((2*pi()/Lap_w)*t + F_mf
*(2*pi()/360))   *1e3;       %Force load
F_sv       = F_sva * H_w * sin((2*pi()/Lap_w)*t + F_svf
*(2*pi()/360)) *1e3;      %Vertical Force ship
F_sh       = F_sha * H_w * sin((2*pi()/Lap_w)*t + F_shf
*(2*pi()/360)) *1e3;      %Horizontal Force ship
M_s        = M_sa * H_w * sin((2*pi()/Lap_w)*t + M_sf
*(2*pi()/360))   *1e3;       %Moment Force ship


%Spring
k_sv         = 7.93e7;                    %vertical spring
ship
k_sh         = 133.3e6;                   %horizontal spring
ship
k_sr         = 5.65e9;                    %rotational spring
ship

%Damping [DOUBLE CHECK CORRENSPONDING PERIOD]
c_tug        = 0;                         %(kg/s)damping
tuggerlines
```

```
c_sv         = 6004400.00;                 %(kg/s)vertical
damping ship
c_sh         = 19268000.00;                %(kg/s)horizontal
damping ship
c_sr         = 735780000.00;               %(kg/s)Rotational
damping ship

%Start values [in RAD]
theta_k      = 0;%0.087;                   %starting pendulum
upper angle
theta_m      = 0;%0.087;                   %starting pendulum
lower angle
theta_s      = 0;                   %starting ship angle this
is not zero to take out the initial movement due to gravity
x_s          = 0;                   %starting ship x-coordinate
y_s          = 0;                   %starting ship y-
coordinate, this is not zero to take out the initial
movement due to gravity
theta_k_dot = 0;                    %starting upper angle speed
theta_m_dot = 0;                    %starting pendulum lower
angle speed
theta_s_dot = 0;                    %starting pendulum lower
angle speed
x_s_dot      = 0;                   %starting ship speed x-
direction
y_s_dot      = 0;                   %starting ship speed y-
direction

%Create q_0
q_0         = zeros(2*(N),1);      %starting conditions
q_0(1,1)    = theta_k;
q_0(2,1)    = theta_m;
q_0(3,1)    = theta_s;
q_0(4,1)    = x_s;
q_0(5,1)    = y_s;
q_0(6,1)    = theta_k_dot;
q_0(7,1)    = theta_m_dot;
q_0(8,1)    = theta_s_dot;
q_0(9,1)    = x_s_dot;
q_0(10,1)   = y_s_dot;

%Global
global theta_k_dotdot
global theta_s_dotdot
```

```
theta_k_dotdot = 0;
theta_s_dotdot = 0;
%% Solve problem


P =
struct('g',g,'a',a,'b',b,'c',c,'d',d,'e',e,'m',m,'M',M,'I_m
',I_m,'I_M',I_M,'N',N,'nt',nt,'T',T,'theta_k',theta_k,'thet
a_m',theta_m,'theta_s',theta_s,'x_s',x_s,'y_s',y_s,'theta_k
_dot',theta_k_dot,'theta_m_dot',theta_m_dot,'theta_s_dot',t
heta_s_dot,'x_s_dot',x_s_dot,'y_s_dot',y_s_dot,'F_ma',F_ma,
'F_mf',F_mf,'F_sva',F_sva,'F_svf',F_svf,'F_sha',F_sha,'F_sh
f',F_shf,'M_sa',M_sa,'M_sf',M_sf','c_tug',c_tug,'c_sh',c_sh
,'c_sv',c_sv,'c_sr',c_sr,'k_sv',k_sv,'k_sh',k_sh,'k_sr',k_s
r,'H_w',H_w,'Lap_w',Lap_w,'H_ship',H_ship,'B_ship','
CS',CS);


[t_vec,DOF] = ode45(@(t,q_n) ODE_5_DOF(q_n,P,t), t, q_0);

%% Eigenvalues
%{
M_matrix = [m .* a .^ 2 .* (theta_k .^ 2 + 1) (m .* b .* a
.* (2 .* theta_k .* theta_m + 1)) ./ 0.2e1 ((c - theta_s .*
d) .* theta_k - c .* theta_s - d) .* m .* a m .* a m .*
theta_k .* a; (m .* b .* a .* (2 .* theta_k .* theta_m +
1)) ./ 0.2e1 (b .^ 2 .* m) ./ 0.4e1 + (theta_m .^ 2 .* b .^
2 .* m) + I_m -(m .* b .* (2 .* theta_s .* theta_m .* d + c
.* theta_s - 2 .* theta_m .* c + d)) ./ 0.2e1 (m .* b) ./
0.2e1 m .* theta_m .* b; ((c - theta_s .* d) .* theta_k - c
.* theta_s - d) .* m .* a -(m .* b .* (2 .* theta_s .*
theta_m .* d + c .* theta_s - 2 .* theta_m .* c + d)) ./
0.2e1 m .* (c .^ 2 + d .^ 2) .* theta_s .^ 2 + m .* (c .^ 2
+ d .^ 2) + I_M -m .* (c .* theta_s + d) m .* (c - theta_s
.* d); m .* a (m .* b) ./ 0.2e1 -m .* (c .* theta_s + d) M
+ m 0; m .* theta_k .* a m .* theta_m .* b m .* (c -
theta_s .* d) 0 M + m;];
%M_added = 0;%[0    0    0    0    0; 0    0    0    0    0; 0
0    2.56E+08    4.51E+05    7.99E+05; 0 0    4.51E+05
1560    2422.5; 0    0    7.99E+05    2422.5   1.02E+05];
%M_matrix = M_sys + M_added;
%
K_matrix = [a*g*m 0 0 0 0; 0 m*g*b 0 0 0; 0 0 (-d*g*m+k_sr)
0 0; 0 0 0 k_sh 0; 0 0 0 0 k_sv];
```

```
M_1 = M_matrix(1,1);
M_2 = M_matrix(2,2);
M_3 = M_matrix(3,3);
M_4 = M_matrix(4,4);
M_5 = M_matrix(5,5);

K_1 = K_matrix(1,1);
K_2 = K_matrix(2,2);
K_3 = K_matrix(3,3);
K_4 = K_matrix(4,4);
K_5 = K_matrix(5,5);

% Get eigen info
[V_1,D_1] = eig(M_1,K_1);
[V_2,D_2] = eig(M_2,K_2);
[V_3,D_3] = eig(M_3,K_3);
[V_4,D_4] = eig(M_4,K_4);
[V_5,D_5] = eig(M_5,K_5);

%{
% Get natural frequencies
nat_freq_1 = 1 / sqrt(D_1) ;
nat_freq_2 = 1 / sqrt(D_2) ;
nat_freq_3 = 1 / sqrt(D_3) ;
nat_freq_4 = 1 / sqrt(D_4) ;
nat_freq_5 = 1 / sqrt(D_5) ;

% Get period
nat_period_1 = 2*pi*sqrt(D_1);
nat_period_2 = 2*pi*sqrt(D_2);
nat_period_3 = 2*pi*sqrt(D_3);
nat_period_4 = 2*pi*sqrt(D_4);
nat_period_5 = 2*pi*sqrt(D_5);
%}

[V,D] = eig(M_matrix,K_matrix);

nat_freq_1 = 1 / sqrt(D(3,3)) ;
nat_freq_2 = 1 / sqrt(D(1,1)) ;
nat_freq_3 = 1 / sqrt(D(4,4)) ;
nat_freq_4 = 1 / sqrt(D(5,5)) ;
nat_freq_5 = 1 / sqrt(D(2,2)) ;
```

```
nat_period_1 = sqrt(D(3,3));
nat_period_2 = sqrt(D(1,1));
nat_period_3 = sqrt(D(4,4));
nat_period_4 = sqrt(D(5,5));
nat_period_5 = sqrt(D(2,2));


V_new(:,1) = V(:,3);
V_new(:,2) = V(:,1);
V_new(:,3) = V(:,4);
V_new(:,4) = V(:,5);
V_new(:,5) = V(:,2);

D;
[D_1 D_2 D_3 D_4 D_5];
%}
%% Orcaflex comparison
%{
orca_nf1 = 1/(nat_period_1*2*pi());
orca_nf2 = 1/(nat_period_2*2*pi());
orca_nf3 = 1/(nat_period_3*2*pi());
orca_nf4 = 1/(nat_period_4*2*pi());
orca_nf5 = 1/(nat_period_5*2*pi());

orca_np1 = nat_period_1*2*pi();
orca_np2 = nat_period_2*2*pi();
orca_np3 = nat_period_3*2*pi();
orca_np4 = nat_period_4*2*pi();
orca_np5 = nat_period_5*2*pi();
%}
%% Plot force
%{
figure()
plot(t_vec,F_m);
hold on
plot(t_vec,F_sh);
hold on
plot(t_vec,F_sv);
hold on
plot(t_vec,M_s);
legend('F_m','F_sh','F_sv','M_s')
%}
%% Plot results
%{
```

```
figure()
title('Vertical velocity is thick black line')
x = linspace(0,T,nt);
%plot(t_vec,DOF(:,1));
%hold on
plot(t_vec,DOF(:,2));
hold on
title('Rotation Jacket')
xlabel('time [s]')
ylabel('rotation [rad]')
legend('theta_m')
%}
%{
plot(t_vec,DOF(:,3));
hold on
plot(t_vec,DOF(:,4));
hold on
plot(t_vec,DOF(:,5));
%}
%{
xlabel('time [s]')
ylabel('displacement [m,rad]')
legend('theta_k', 'theta_m', 'theta_s', 'x_s','y_s')
%}
%{
figure()
plot(t_vec,DOF(:,6));
hold on
plot(t_vec,DOF(:,7));
hold on
plot(t_vec,DOF(:,8));
hold on
plot(t_vec,DOF(:,9));
hold on
plot(t_vec,DOF(:,10));
hold on
xlabel('time [s]')
ylabel('velocity [m/s,rad/s]')
legend('theta_k dot','theta_m dot','theta_s dot','x_s
dot','y_s dot')
%}

%% Animation/Movie
%{
```

```
figure

% Setting
mv_on = 0; % {0,1}  0: show animation,  1: make a movie

% Initialize movie maker
if mv_on == 0

    % Make 4 movies
    writerObj = VideoWriter('test_movie','MPEG-4');
    writerObj.Quality = 100;
    writerObj.FrameRate = 60;

    % Start
    open(writerObj);

end


boat        = [-B_ship,B_ship,B_ship,-B_ship,-B_ship; -
H_ship,-H_ship,H_ship,H_ship,-H_ship]; % [x_locs; y_locs]
crane_v     = [0.3*B_ship,0.3*B_ship;H_ship,d];
crane_h     = [0.3*B_ship,c;d,d];
cable       = [0,0;0,-a];
load        = [-1/30*b,1/30*b,1/30*b,-1/30*b,-1/30*b; -b,-
b,0,0,-b];


% Define 2D rotation matrix
R = @(phi) [cos(phi), -sin(phi); sin(phi), cos(phi)];


% Initialize handles
H_vertices_ship     = line(0,0);
H_cog_ship          = line(0,0,'marker','.');
H_vertices_crane_v  = line(0,0);
H_vertices_crane_h  = line(0,0);
H_vertices_cable    = line(0,0);
H_vertices_load     = line(0,0);


% Set axis
f = 2;
axis([-80 80 -20 160])
axis('manual')
```

```
% Loop over time
ent = 0;
for t_current = t_vec.'
    ent = ent + 1;

    theta_k      = DOF(ent,1);
    theta_m      = DOF(ent,2);
    theta_s      = DOF(ent,3);
    x_s          = DOF(ent,4);
    y_s          = DOF(ent,5);

    x_m(ent,1) = theta_k*a + 0.5*b*theta_m ;%-
0.5*c*theta_s^2 - theta_s*d;

    % Get rotate vectices
    vboat         = R(theta_s)*boat;
    vcrane_v      = R(theta_s)*crane_v;
    vcrane_h      = R(theta_s)*crane_h;
    vcable        = R(theta_k)*cable;
    vload         = R(theta_m)*load;



   % Update location of COG of boat
    H_cog_ship.XData = x_s;
    H_cog_ship.YData = y_s;

    % Update location of vertices of ship
    H_vertices_ship.XData = vboat(1,:) + x_s;
    H_vertices_ship.YData = vboat(2,:) + y_s;


    % Update location of vertices of crane_v
    H_vertices_crane_v.XData = vcrane_v(1,:) + x_s;
    H_vertices_crane_v.YData = vcrane_v(2,:) + y_s;


    % Update location of vertices of crane_h
    H_vertices_crane_h.XData = vcrane_h(1,:) + x_s;
    H_vertices_crane_h.YData = vcrane_h(2,:) + y_s;


    % Update location of vertices of cable
```

```
    H_vertices_cable.XData = vcable(1,:) + c + x_s - c*(1-
cos(theta_s)) - d*sin(theta_s);
    H_vertices_cable.YData = vcable(2,:) + d + y_s +
c*sin(theta_s) - d*(1-cos(theta_s));

    % Update location of vertices of load
    H_vertices_load.XData = vload(1,:) + c + x_s - c*(1-
cos(theta_s)) - d*sin(theta_s) + a*sin(theta_k);
    H_vertices_load.YData = vload(2,:) + d - a + y_s +
c*sin(theta_s) - d*(1-cos(theta_s)) + a*(1-cos(theta_k));



    % Update figure / capture frame
    if mv_on == 1
        writeVideo(writerObj,getframe(gcf));
    else
        drawnow
    end

end
x_m_SS = x_m(700:end,:);
Max_dev_right_SS= max(x_m_SS)
Max_dev_left_SS = min(x_m_SS)
%}

roll_angle = DOF(:,2);
Max_roll_angle_left = min(roll_angle);
Max_roll_angle_left = Max_roll_angle_left *-1;
Max_roll_angle_right= max(roll_angle);
Max_roll_angle =
max(Max_roll_angle_right,Max_roll_angle_left)
```

B.2. Controller PID

```matlab
function q_n_dot = ODE_5_DOF(q_n, P, t)
%% Input

g         =P.g;
a         =P.a;
b         =P.b;
c         =P.c;
d         =P.d;
e         =P.e;
m         =P.m;
M         =P.M;
I_m       =P.I_m;
I_M       =P.I_M;
N         =P.N;
nt        =P.nt;
T         =P.T;
F_ma      =P.F_ma;
F_mf      =P.F_mf;
F_sva     =P.F_sva;
F_svf     =P.F_svf;
F_sha     =P.F_sha;
F_shf     =P.F_shf;
M_sa      =P.M_sa;
M_sf      =P.M_sf;
c_tug     =P.c_tug;
c_sh      =P.c_sh;
c_sv      =P.c_sv;
c_sr      =P.c_sr;
k_sv      =P.k_sv;
k_sr      =P.k_sr;
k_sh      =P.k_sh;
H_w       =P.H_w;
Lap_w     =P.Lap_w;
CS        =P.CS;

H_ship     =P.H_ship;
B_ship     =P.B_ship;


theta_k = q_n(1);
theta_m = q_n(2);
theta_s = q_n(3);
x_s     = q_n(4);
y_s     = q_n(5);
```

```
theta_k_dot = q_n(6);
theta_m_dot = q_n(7);
theta_s_dot = q_n(8);
x_s_dot     = q_n(9);
y_s_dot     = q_n(10);

%% Global

global theta_k_dotdot
global theta_s_dotdot
%% Calculate L_t and V_t

x_rl = c + x_s - theta_s .* d + theta_k .* a -
0.5.*c.*theta_s.^2;                         %x-coordinate for
tugger attachment load
y_rl = (d - a) + y_s + theta_s .* c + 0.5.*a.*theta_k.^2 -
0.5.*d.*theta_s.^2;           %y-coordinate for tugger
attachment load

x_rs = B_ship + x_s - theta_s .* H_ship -
0.5.*B_ship.*theta_s.^2;      %x-coordinate for tugger
attachment ship
y_rs = H_ship + y_s + theta_s .* B_ship -
0.5.*H_ship.*theta_s.^2;      %y-coordinate for tugger
attachment ship

L_tc            = [x_rl - x_rs;y_rl - y_rs];        %Vector
for the length of the tugger line
L_t             = sqrt(L_tc(1).^2 + L_tc(2).^2);    %The
length of the tugger line

V_t = ((c + theta_k .* a - c .* theta_s .^ 2 ./ 0.2e1 -
theta_s .* d - B_ship - theta_s .* H_ship + B_ship .*
theta_s .^ 2 ./ 0.2e1) .^ 2 + (d - a + a .* theta_k .^ 2 ./
0.2e1 + c .* theta_s - d .* theta_s .^ 2 ./ 0.2e1 - H_ship
- theta_s .* B_ship + H_ship .* theta_s .^ 2 ./ 0.2e1) .^
2) .^ (-0.1e1 ./ 0.2e1) .* (0.2e1 .* (c + theta_k .* a - c
.* theta_s .^ 2 ./ 0.2e1 - theta_s .* d - B_ship - theta_s
.* H_ship + B_ship .* theta_s .^ 2 ./ 0.2e1) .*
(theta_k_dot .* a - c .* theta_s .* theta_s_dot -
theta_s_dot .* d - theta_s_dot .* H_ship + B_ship .*
theta_s .* theta_s_dot) + 0.2e1 .* (d - a + a .* theta_k .^
2 ./ 0.2e1 + c .* theta_s - d .* theta_s .^ 2 ./ 0.2e1 -
H_ship - theta_s .* B_ship + H_ship .* theta_s .^ 2 ./
```

```
0.2e1) .* (a .* theta_k .* theta_k_dot + c .* theta_s_dot -
d .* theta_s .* theta_s_dot - theta_s_dot .* B_ship +
H_ship .* theta_s .* theta_s_dot)) ./ 0.2e1;

frac_x_tug = L_tc(1)./L_t;          %constant for the amount
of force in x-direction from tugger line
frac_y_tug = L_tc(2)./L_t;          %constant for the amount
of force in y-direction from tugger line

%% No control system

if CS==0
    F_tug=0;
end

%% Control system #1 (step-wise damper)
if CS==1

F_tug_haul=2e2;  %Value for the tugger tension when the
tugger is hauling in
F_tug_pay =15e5;  %Value for the tugger tension when the
tugger is paying out


if V_t<=0
    F_tug = F_tug_haul;
elseif V_t>0
    F_tug = F_tug_pay;
end
plot(V_t,F_tug,'r*')
hold on
end

%% Control system #2 (linear damper)
if CS==2

F_tug_min = 2e2;                          %minimum value for
the tugger tension
F_tug_max = 15e5;                          %maximum value for
the tugger tension
FOD =10e5;                                 %[N./m] value for
first order term
t_target = (F_tug_min+F_tug_max)./2;     %[N] target tension
```

```
F_tug = t_target + FOD .* V_t;           %Control system
formula

 if F_tug<F_tug_min
     F_tug = F_tug_min;
 elseif F_tug>F_tug_max
     F_tug = F_tug_max;
 end


 plot(V_t,F_tug,'r*')
hold on
title('Damping function')
xlabel('Winch line velocity [m/s]')
ylabel('Line tension [kN]')

end
%% Control system #3 (linear shiften damper forward)
if CS==3.1
F_tug_min = 2e2;                        %minimum value for
the tugger tension
F_tug_max = 15e5;                      %maximum value for
the tugger tension
FOD =10e5;                              %[N./m] value for
first order term
t_target = F_tug_min;                   %[N] target tension

F_tug = t_target + FOD .* V_t;           %Control system
formula

 if F_tug<F_tug_min
     F_tug = F_tug_min;
 elseif F_tug>F_tug_max
     F_tug = F_tug_max;
 end

plot(t,F_tug,'r*')
hold on

end

%% Control system #3 (linear shiften damper forward)
if CS==3.2
```

```
F_tug_min = 2e2;                               %minimum value for
the tugger tension
F_tug_max = 15e5;                             %maximum value for
the tugger tension
FOD =10e5;                                      %[N./m] value for
first order term
t_target = F_tug_max;                          %[N] target tension

F_tug = t_target + FOD .* V_t;                 %Control system
formula

 if F_tug<F_tug_min
     F_tug = F_tug_min;
 elseif F_tug>F_tug_max
     F_tug = F_tug_max;
 end

plot(V_t,F_tug,'r*')
hold on

end
%% Control system #4 (Quadratic damper)
if CS==4
F_tug_min = 2e2;                                   %minimum value
for the tugger tension
F_tug_max = 15e5;                                 %maximum value for
the tugger tension
FOD =10e5;                                          %[N./(m./s)]
value for first order term
SOD =5e5;                                           %[N./(m./s).^2]
value for second order term
t_target =(F_tug_min+F_tug_max)./2;         %[N] target
tension

if V_t<0
    F_tug = t_target + FOD.*V_t - SOD.*V_t.^2;
elseif V_t>=0
    F_tug = t_target + FOD.*V_t + SOD.*V_t.^2;
end

 if F_tug<F_tug_min
     F_tug = F_tug_min;
 elseif F_tug>F_tug_max
     F_tug = F_tug_max;
```

```
 end
plot(V_t,F_tug,'r*')
hold on

end
%% Control system #5 (orcaflex winch control system)
if CS==5
F_tug_min = 2e2;                      %minimum value for
the tugger tension
F_tug_max = 15e5;                     %maximum value for
the tugger tension

t_target=7.5e5;                      %[N] target tension
DB = 2e5;                             %[N] winch drive
deadband
FOD =10e5;                            %[N./(m./s)] value
for first order term
SOD =5e5;                             %[N./(m./s).^2] value
for second order term

if V_t<0
    F_tug=(t_target - DB + FOD.*V_t - SOD.*V_t.^2);
elseif V_t==0
    F_tug=t_target;
elseif V_t>0
    F_tug=(t_target + DB + FOD.*V_t + SOD.*V_t.^2);
end

 if F_tug<F_tug_min
     F_tug = F_tug_min;
 elseif F_tug>F_tug_max
     F_tug = F_tug_max;
 end
plot(V_t,F_tug,'r*')
hold on

end
%% PID controller
if CS==6
V_ref = 0;
error = V_t - V_ref;
t_target = 7.5e5;

Kp =22e5;
```

```
Ki =10e5;
Kd =16e5;

P = error;
I = -L_t + 87.0136;
D =  -(-((c + theta_k .* a - c .* theta_s .^ 2 ./ 0.2e1 -
theta_s .* d - B_ship - theta_s .* H_ship + B_ship .*
theta_s .^ 2 ./ 0.2e1) .^ 2 + (d - a + a .* theta_k .^ 2 ./
0.2e1 + c .* theta_s - d .* theta_s .^ 2 ./ 0.2e1 - H_ship
- theta_s .* B_ship + H_ship .* theta_s .^ 2 ./ 0.2e1) .^
2) .^ (-0.3e1 ./ 0.2e1) .* (0.2e1 .* (c + theta_k .* a - c
.* theta_s .^ 2 ./ 0.2e1 - theta_s .* d - B_ship - theta_s
.* H_ship + B_ship .* theta_s .^ 2 ./ 0.2e1) .*
(theta_k_dot .* a - c .* theta_s .* theta_s_dot -
theta_s_dot .* d - theta_s_dot .* H_ship + B_ship .*
theta_s .* theta_s_dot) + 0.2e1 .* (d - a + a .* theta_k .^
2 ./ 0.2e1 + c .* theta_s - d .* theta_s .^ 2 ./ 0.2e1 -
H_ship - theta_s .* B_ship + H_ship .* theta_s .^ 2 ./
0.2e1) .* (a .* theta_k .* theta_k_dot + c .* theta_s_dot -
d .* theta_s .* theta_s_dot - theta_s_dot .* B_ship +
H_ship .* theta_s .* theta_s_dot)) .^ 2 ./ 0.4e1 + ((c +
theta_k .* a - c .* theta_s .^ 2 ./ 0.2e1 - theta_s .* d -
B_ship - theta_s .* H_ship + B_ship .* theta_s .^ 2 ./
0.2e1) .^ 2 + (d - a + a .* theta_k .^ 2 ./ 0.2e1 + c .*
theta_s - d .* theta_s .^ 2 ./ 0.2e1 - H_ship - theta_s .*
B_ship + H_ship .* theta_s .^ 2 ./ 0.2e1) .^ 2) .^ (-0.1e1
./ 0.2e1) .* (0.2e1 .* (theta_k_dot .* a - c .* theta_s .*
theta_s_dot - theta_s_dot .* d - theta_s_dot .* H_ship +
B_ship .* theta_s .* theta_s_dot) .^ 2 + 0.2e1 .* (c +
theta_k .* a - c .* theta_s .^ 2 ./ 0.2e1 - theta_s .* d -
B_ship - theta_s .* H_ship + B_ship .* theta_s .^ 2 ./
0.2e1) .* (theta_k_dotdot .* a - c .* theta_s_dot .^ 2 - c
.* theta_s .* theta_s_dotdot - theta_s_dotdot .* d -
theta_s_dotdot .* H_ship + B_ship .* theta_s_dot .^ 2 +
B_ship .* theta_s .* theta_s_dotdot) + 0.2e1 .* (a .*
theta_k .* theta_k_dot + c .* theta_s_dot - d .* theta_s .*
theta_s_dot - theta_s_dot .* B_ship + H_ship .* theta_s .*
theta_s_dot) .^ 2 + 0.2e1 .* (d - a + a .* theta_k .^ 2 ./
0.2e1 + c .* theta_s - d .* theta_s .^ 2 ./ 0.2e1 - H_ship
- theta_s .* B_ship + H_ship .* theta_s .^ 2 ./ 0.2e1) .*
(a .* theta_k_dot .^ 2 + a .* theta_k .* theta_k_dotdot + c
.* theta_s_dotdot - d .* theta_s_dot .^ 2 - d .* theta_s .*
theta_s_dotdot - theta_s_dotdot .* B_ship + H_ship .*
```

```
theta_s_dot .^ 2 + H_ship .* theta_s .* theta_s_dotdot)) ./
0.2e1);


F_tug = t_target + Kp*P + Ki*I + Kd*D;


F_tug_min = 2e2;                        %minimum value for
the tugger tension
F_tug_max = 15e5;                       %maximum value for
the tugger tension

if F_tug<F_tug_min
     F_tug = F_tug_min;
 elseif F_tug>F_tug_max
     F_tug = F_tug_max;
end
%{
plot(V_t,F_tug,'r*')
hold on


plot(t,F_tug,'r*')
hold on


plot(V_t,F_tug,'r*')
hold on

F_tug_c=F_tug/10e5;
plot(t,P,'r*')
hold on
plot(t,I,'b*')
plot(t,D,'g*')
plot(t,F_tug_c,'y*')
%}
end
%% Forces

F_m        = F_ma .* H_w .* sin((2.*pi()./Lap_w).*t + F_mf
.*(2.*pi()./360))    .*1e3;                          %Force load
F_sv       = - ((m+M).*g) + F_sva .* H_w .*
sin((2.*pi()./Lap_w).*t + F_svf .*(2.*pi()./360)) .*1e3;
%Vertical Force ship
```

```
F_sh       = F_sha .* H_w .* sin((2.*pi()./Lap_w).*t + F_shf
.*(2.*pi()./360)) .*1e3;                       %Horizontal
Force ship
M_s        = -(m.*c.*g) + M_sa .* H_w .*
sin((2.*pi()./Lap_w).*t + M_sf .*(2.*pi()./360))   .*1e3;
%Moment Force ship

%% M and F
%For import from Maple replace the following terms in the
right order with the use of ctrl+f:
% 1: Replace * by .* (check q_n_dot, as here will spawn an
extra "." as well)
% 2: Replace ^ by .^
% 3: Replace / by ./
% 4: Replace diff(theta_k(t), t) by theta_k_dot
% 5: Replace diff(theta_m(t), t) by theta_m_dot
% 6: Replace diff(theta_s(t), t) by theta_s_dot
% 7: Replace diff(x_s(t), t) by x_s_dot
% 8: Replace diff(y_s(t), t) by y_s_dot
% 9: Replace theta_k(t) by theta_k
% 10: Replace theta_m(t) by theta_m
% 11: Replace theta_s(t) by theta_s
% 12: Replace x_s(t) by x_s
% 13: Replace y_s(t) by y_s
% 14: Replace F_m(t) by F_m
% 15: Replace F_sh(t) by F_sh
% 16: Replace F_sv(t) by F_sv
% 17: Replace M_s(t) by M_s
% 18: Replace diff(theta_k_dot, t) by theta_k_dotdot
% 19: Replace diff(theta_s_dot, t) by theta_s_dotdot

%% Input
M_sys = [m .* a .^ 2 .* (theta_k .^ 2 + 1) (m .* b .* a .*
(2 .* theta_k .* theta_m + 1)) ./ 0.2e1 a .* m .* ((c -
theta_s .* d) .* theta_k - c .* theta_s - d) m .* a m .*
theta_k .* a; (m .* b .* a .* (2 .* theta_k .* theta_m +
1)) ./ 0.2e1 (m .* b .^ 2) ./ 0.4e1 + (m .* theta_m .^ 2 .*
b .^ 2) + I_m -(m .* b .* (2 .* theta_s .* theta_m .* d + c
.* theta_s - 2 .* theta_m .* c + d)) ./ 0.2e1 (m .* b) ./
0.2e1 m .* theta_m .* b; a .* m .* ((c - theta_s .* d) .*
theta_k - c .* theta_s - d) -(m .* b .* (2 .* theta_s .*
theta_m .* d + c .* theta_s - 2 .* theta_m .* c + d)) ./
0.2e1 m .* (c .^ 2 + d .^ 2) .* theta_s .^ 2 + m .* (c .^ 2
+ d .^ 2) + I_M -m .* (c .* theta_s + d) m .* (c - theta_s
```

```
.* d); m .* a (m .* b) ./ 0.2e1 -m .* (c .* theta_s + d) M
+ m 0; m .* theta_k .* a m .* theta_m .* b m .* (c -
theta_s .* d) 0 M + m;];

F_matrix = [-a .* (-m .* (theta_k .* d + c) .* theta_s_dot
.^ 2 + theta_k .* theta_k_dot .^ 2 .* a .* m + theta_k .*
theta_m_dot .^ 2 .* b .* m + (-frac_y_tug .* -F_tug + m .*
g) .* theta_k - -F_tug .* frac_x_tug + F_m) -b .* (-
(theta_m .* d + c ./ 0.2e1) .* m .* theta_s_dot .^ 2 +
theta_m .* theta_k_dot .^ 2 .* a .* m + theta_m .*
theta_m_dot .^ 2 .* b .* m + m .* g .* theta_m + F_m ./
0.2e1) -theta_s .* m .* (c .^ 2 + d .^ 2) .* theta_s_dot .^
2 - c_sr .* theta_s_dot + a .* m .* (theta_s .* d - c) .*
theta_k_dot .^ 2 + m .* b .* (theta_s .* d - c) .*
theta_m_dot .^ 2 + ((B_ship .* frac_x_tug + H_ship .*
frac_y_tug - c .* frac_x_tug - d .* frac_y_tug) .* -F_tug +
d .* g .* m + F_m .* c - k_sr) .* theta_s + (-B_ship .*
frac_y_tug - H_ship .* frac_x_tug + c .* frac_y_tug - d .*
frac_x_tug) .* -F_tug - c .* g .* m + F_m .* d - M_s m .* c
.* theta_s_dot .^ 2 - c_sh .* x_s_dot - k_sh .* x_s - F_sh
- F_m theta_s_dot .^ 2 .* d .* m - theta_k_dot .^ 2 .* a .*
m - theta_m_dot .^ 2 .* b .* m - M .* g - k_sv .* y_s -
c_sv .* y_s_dot - m .* g - F_sv];

F_matrix = transpose(F_matrix);

%% Added mass
% Search for the right added mass in Orcaflex model take
the following steps:
% 1. Choose the right period in orcaflex
% 2. Fill in the matrix with the right corresponding DOF
% [x;x;Roll.../Pitch;Sway.../Surge;Heave]
% 3. For this file the vector of the matrix looks like:
% [0;0;Pitch;Surge;Heave]
% 4. Fill in the vector found for specific period from
orcaflex into the
% belonging excel input file
% 5. Fill in the outcome into the M_added term

M_added = 1000.*[0   0    0    0    0; 0     0    0    0    0; 0
0   5.98e06 2.92e01 5.20e02; 0   0   2.92e01 922.53  3080.9;
0   0   5.20e02 3080.9  9.34e04];

M_matrix= M_sys + M_added;
```

```
%% Calculate q_n_dot
q_n_dot = zeros(N.*2,1);

q_n_dot(1:N)=q_n(N+1:end);

q_n_dot(N+1:end)=M_matrix\F_matrix;

%% Update

theta_k_dotdot = q_n_dot(6);
theta_s_dotdot = q_n_dot(8);

end
```

C

Matlab model control systems

Show the calculation of tugger line length, velocity and acceleration from maple

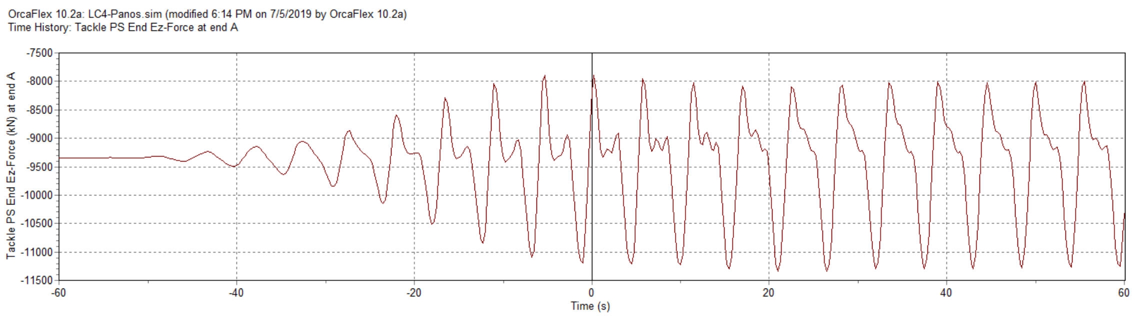


Figure C.1: Vertical crane tip force over time for linear control system

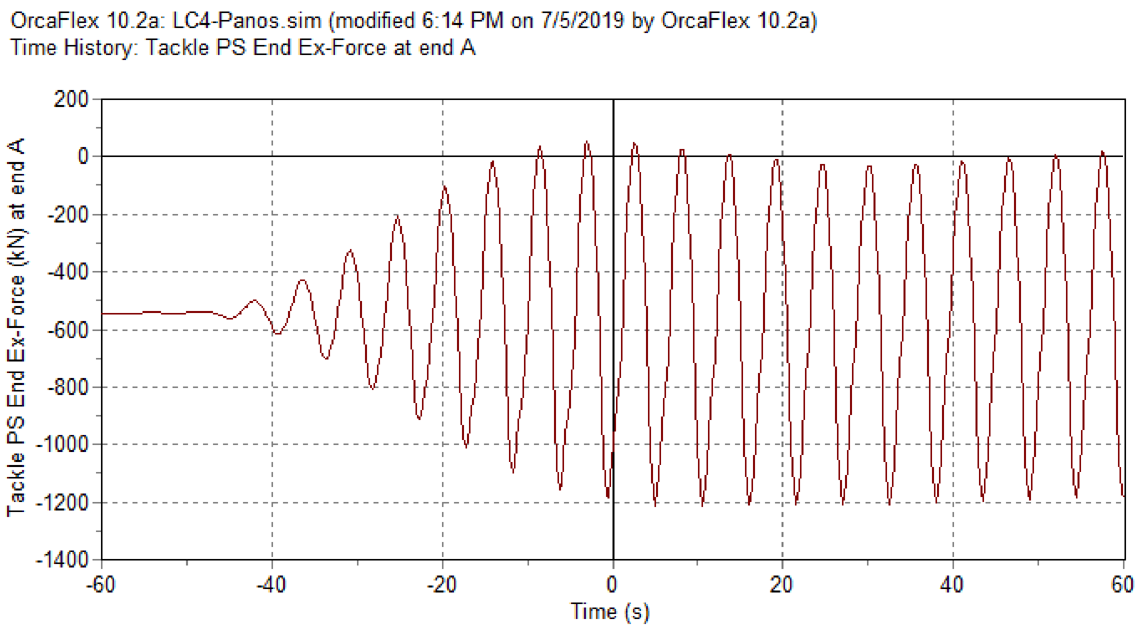


Figure C.2: Offlead crane tip force over time for linear control system

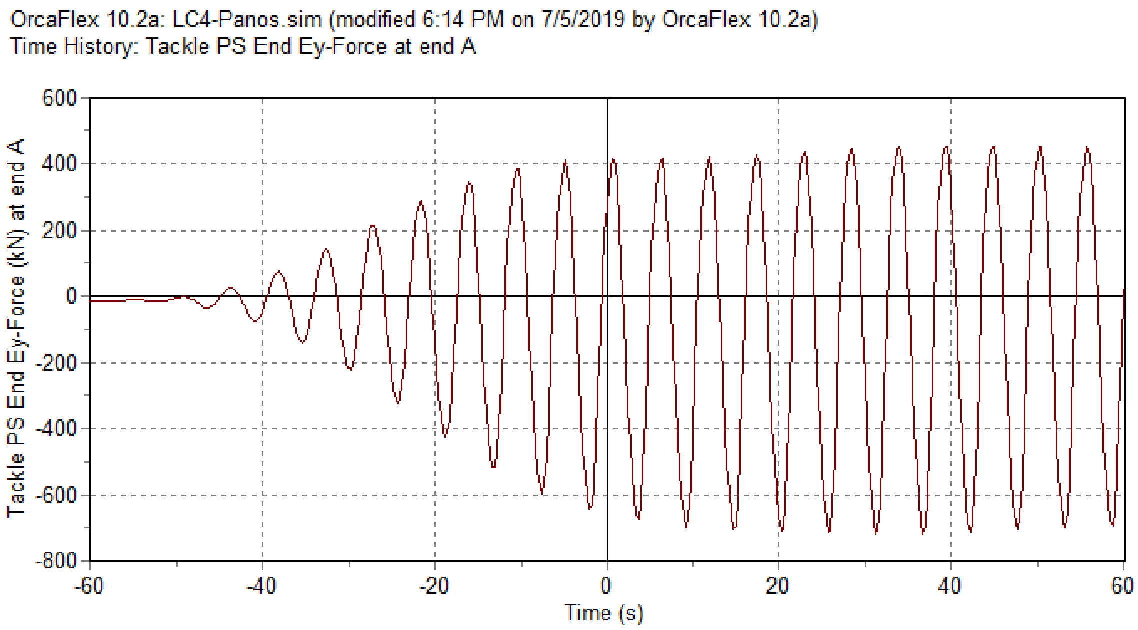


Figure C.3: Sidelead crane tip force over time for linear control system

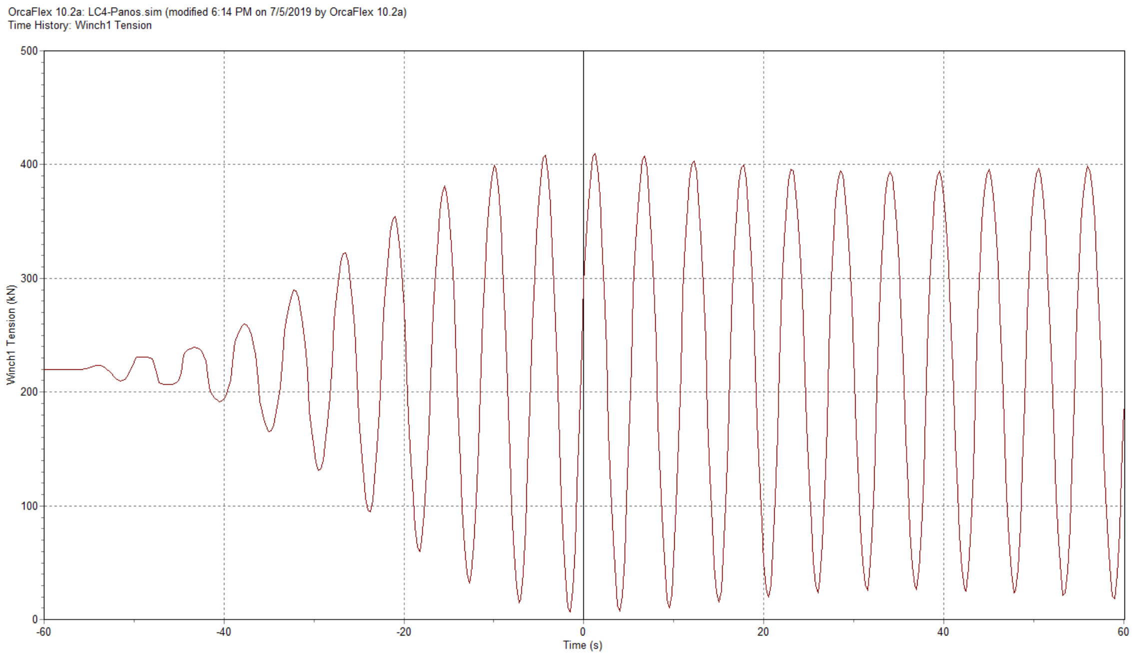


Figure C.4: Boom winches tension over time for linear control system

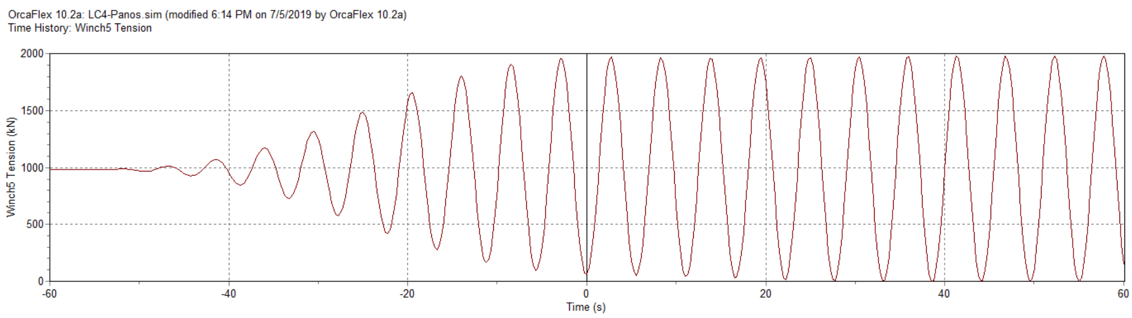


Figure C.5: Deck winches tension over time for linear control system

D

Matlab matrices

$$\mathit{M\_matrix} := \begin{bmatrix} m\,a^2\left(\mathit{theta\_k}(t)^2+1\right) & \frac{m\,b\,a\,(2\,\mathit{theta\_k}(t)\,\mathit{theta\_m}(t)+1)}{2} & a\,m\,((c-\mathit{theta\_s}(t)\,d)\,\mathit{theta\_k}(t)-c\,\mathit{theta\_s}(t)-d) & m\,a & m\,\mathit{theta\_k}(t)\,a \\ \frac{m\,b\,a\,(2\,\mathit{theta\_k}(t)\,\mathit{theta\_m}(t)+1)}{2} & \frac{m\,b^2}{4}+m\,\mathit{theta\_m}(t)^2\,b^2+\mathit{I\_m} & -\frac{m\,b\,(2\,\mathit{theta\_s}(t)\,\mathit{theta\_m}(t)\,d+c\,\mathit{theta\_s}(t)-2\,\mathit{theta\_m}(t)\,c+d)}{2} & \frac{m\,b}{2} & m\,\mathit{theta\_m}(t)\,b \\ a\,m\,((c-\mathit{theta\_s}(t)\,d)\,\mathit{theta\_k}(t)-c\,\mathit{theta\_s}(t)-d) & -\frac{m\,b\,(2\,\mathit{theta\_s}(t)\,\mathit{theta\_m}(t)\,d+c\,\mathit{theta\_s}(t)-2\,\mathit{theta\_m}(t)\,c+d)}{2} & m\left(c^2+d^2\right)\mathit{theta\_s}(t)^2+m\left(c^2+d^2\right)+\mathit{I\_M} & -m\,(c\,\mathit{theta\_s}(t)+d) & m\,(c-\mathit{theta\_s}(t)\,d) \\ m\,a & \frac{m\,b}{2} & -m\,(c\,\mathit{theta\_s}(t)+d) & m+M & 0 \\ m\,\mathit{theta\_k}(t)\,a & m\,\mathit{theta\_m}(t)\,b & m\,(c-\mathit{theta\_s}(t)\,d) & 0 & m+M \end{bmatrix}$$

Figure D.1: The M matrix

$$\mathit{C\_matrix} := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathit{c\_sr} & 0 & 0 \\ 0 & 0 & 0 & \mathit{c\_sh} & 0 \\ 0 & 0 & 0 & 0 & \mathit{c\_sv} \end{bmatrix}$$

Figure D.2: The C matrix

$$K := \begin{bmatrix} -\mathit{F\_tug}(t)\,a\,\mathit{frac\_y\_tug}+a\,g\,m & 0 & 0 & 0 & 0 \\ 0 & m\,g\,b & 0 & 0 & 0 \\ 0 & 0 & -\mathit{F\_tug}(t)\,\mathit{B\_ship}\,\mathit{frac\_x\_tug}-\mathit{F\_tug}(t)\,\mathit{H\_ship}\,\mathit{frac\_y\_tug}+\mathit{F\_tug}(t)\,c\,\mathit{frac\_x\_tug}+\mathit{F\_tug}(t)\,d\,\mathit{frac\_y\_tug}-d\,g\,m-\mathit{F\_m}(t)\,c+\mathit{k\_sr} & 0 & 0 \\ 0 & 0 & 0 & \mathit{k\_sh} & 0 \\ 0 & 0 & 0 & 0 & \mathit{k\_sv} \end{bmatrix}$$

Figure D.3: The K matrix

$$F\_iso := \left[\begin{array}{c} a\left(-frac\_x\_tug\,F\_tug(t) + F\_m(t)\right) \\ \dfrac{b\,F\_m(t)}{2} \\ \left(\left(d + H\_ship\right) frac\_x\_tug - frac\_y\_tug\left(c - B\_ship\right)\right) F\_tug(t) + c\,g\,m - F\_m(t)\,d + M\_s \\ F\_sh(t) + F\_m(t) \\ F\_sv(t) + (m + M)\,g \end{array}\right]$$

Figure D.4: The F vector

E

Further elaboration on Orcaflex

E.1. Orcaflex

It has non-linear time domain finite element capabilities, and makes used of comprehensive 3D graphics to aid understanding. This includes animated replays plus full graphical and numerical presentation. Additionally, the program can do modal analysis and can calculate RAOs for any result variable. Orcaflex is often used for applications in Defence, Oceanography and renewable energy. [10]

E.2. Predefined winches of Orcaflex

This appendix treats further explonations of the predefined winches present in Orcaflex. The winch object in orcaflex provides the option to connect at least two coordinates by a mass-less winch wire. Each connection can be fixed, anchored or connected to other objects. The program contains two types of winches. Namely, simple winches and detailed winches. The simple winches behave "perfect". In the sense that it achieves its requirements immediately and perfectly. This option comes in handy if very little is known on the winch parameters which is tried to be simulated. The detailed winches require input to be able to operate.

The winches are able to operate on different control modes, which can be divided into two classes. The winch wires can be length controlled or tension controlled. As mentioned earlier with length control (Fixed length) the winch behaves as if the winch has its brakes on after applying pretension. With tension control the system tries to achieve a defined tension in the winch wire. For "simple" winches this means the wire always applies its target force instantaneously. In essence the line acts as a constant tension system. With constant tension control (CT) no real damping takes place. It tries to to achieve the desired tension by applying force on one side of the winch wire. By trying to oppose the force applies on the other end, adding up to the desired tension value.
Detailed winches work with a damping function as described in section 4.2.

F

Python PID script

PID controller Python

```
import sys

import json

import math

import OrcFxAPI

import random

import numpy


PIDe_ID = 0

PIDdedt_ID = 1

PIDiedt_ID = 2

time = 0



class PIDstate(object):


    def __init__(self):

        self.valid = False

        self.time = -OrcFxAPI.OrcinaInfinity()

        self.signal = 0.0

        self.iedt = 0.0

        self.dedt = 0.0




    def getStateAttributes(self):

        return {'valid': self.valid,

                'time': self.time,
```

```
            'signal': self.signal,
            'iedt': self.iedt,
            'dedt': self.dedt}

    def setStateAttributes(self, attributes):
        self.valid = attributes['valid']
        self.time = attributes['time']
        self.signal = attributes['signal']
        self.iedt = attributes['iedt']
        self.dedt = attributes['dedt']

class PID_left(object):

    def Initialise(self, info):
        self.periodNow = OrcFxAPI.Period(OrcFxAPI.pnInstantaneousValue)
        self.ObjectExtra = OrcFxAPI.ObjectExtra()
        self.ObjectExtra.RigidBodyPos = (0.0, 0.0, 0.0)
        params = info.ObjectParameters

        def GetParameter(paramName, default=None):
            if paramName in params:
                param = params[paramName]
                if isinstance(default, float):
                    param = float(param)
                elif isinstance(default, int):
                    param = int(param)
            elif not default is None:
                param = default
            else:
```

```
            raise Exception('Parameter %s is required but is not included in the object parameters.' %
paramName)
          return param

        ##Determine Object and controlled variables
        self.ControlledObject = info.Model[GetParameter('ControlledObject')]

        #self.ControlledVariable = GetParameter('ControlledVariable')
        self.ControlledVariable = 'Velocity'


        ##Get PID parameters
        self.TargetValue = GetParameter('TargetValue', 0.0)
        self.k0 = GetParameter('k0', 0.0) # constant part
        self.kP = GetParameter('kP', 0.0) # scaling constant for the proportional part
        self.kI = GetParameter('kI', 0.0) # scaling constant for the integral part
        self.kD = GetParameter('kD', 0.0) # scaling constant for the differential part
        self.ControlStartTime = GetParameter('ControlStartTime', -OrcFxAPI.OrcinaInfinity())
        self.MinValue = GetParameter('MinValue', -OrcFxAPI.OrcinaInfinity())
        self.MaxValue = GetParameter('MaxValue', OrcFxAPI.OrcinaInfinity())
        self.DelayActuator = GetParameter('DelayActuator', 0.0)
        self.MaxErrorActuator = GetParameter('MaxErrorActuator', 0.0)
        self.MaxSpeed = GetParameter('MaxSpeed', 0.0)
        self.MaxTensionIncrease = GetParameter('MaxTensionIncrease', 0.0)


        self.prev = PIDstate()
        self.now = PIDstate()
        if info.StateData:
```

```
            state = json.loads(info.StateData)
            self.now.setStateAttributes(state['now'])
            self.prev.setStateAttributes(state['prev'])
        else:
            self.prev.iedt = GetParameter('Initial e/D', 0.0)
            self.now.dedt = GetParameter('Initial De', 0.0)

        #print 'Initialised OK.'

    def RegisterResults(self, info):
        info.ExternalResults = [
            {'ID': PIDe_ID, 'Name': 'e', 'Units': ''},
            {'ID': PIDdedt_ID, 'Name': 'dedt', 'Units': ''},
            {'ID': PIDiedt_ID, 'Name': 'iedt', 'Units': ''},
        ]

    def Calculate(self, info):



        if info.SimulationTime < self.ControlStartTime:
            return

        if info.NewTimeStep and self.now.valid:
            self.prev.time = self.now.time
            self.prev.signal = self.now.signal
            self.prev.iedt = self.now.iedt
            self.prev.dedt = self.now.dedt
```

```
    self.prev.valid = True

self.now.time = info.SimulationTime
self.now.signal = self.ControlledObject.TimeHistory(
  self.ControlledVariable,
  self.periodNow, # set up in Initialise() method to give the value 'now'
  self.ObjectExtra # set up in Initialise() method
)[0] # TimeHistory returns an array, which in this case contains just 1 item, the value now
self.now.iedt = self.prev.iedt

self.now.valid = True

e = self.now.signal - self.TargetValue
if self.prev.valid:
    prev_e = self.prev.signal - self.TargetValue
    dt = self.now.time - self.prev.time;
    self.now.dedt = (prev_e-e)/dt;
    self.now.iedt += dt*(e+prev_e)/2.0;

info.Value = self.kP * e + self.kI * self.now.iedt + self.kD * self.now.dedt + self.k0
if info.Value<=self.MinValue:
    info.Value=self.MinValue
    self.now.iedt=self.prev.iedt #integrator anti wind up
if info.Value>=self.MaxValue:
    info.Value=self.MaxValue
    self.now.iedt=self.prev.iedt #integrator anti wind up

#actuator error
self.error =  random.randrange(-self.MaxErrorActuator, self.MaxErrorActuator, 1)
```

```
    info.Value = info.Value + self.error

def StoreState(self, info):
    state = {'now': self.now.getStateAttributes(), 'prev': self.prev.getStateAttributes()}
    info.StateData = json.dumps(state)

def LogResult(self, info):
    info.LogData = json.dumps([self.now.signal - self.TargetValue, self.now.dedt,self.now.iedt])



def DeriveResult(self, info):
    values = json.loads(info.LogData)
    if info.ResultID==PIDe_ID:
        info.Value = values[0]
    elif info.ResultID==PIDdedt_ID:
        info.Value = values[1]
    elif info.ResultID==PIDiedt_ID:
        info.Value = values[2]
```

G

Airy wave loading from other directions

For the report it is assumed the vessel is able to orient the vessel to face the most disadvantageous waves head on when lifting. For both the linear and PID controller is checked what the results on the crane tip forces are when for certain reasons the vessel cannot face the waves head on. The model has both roll and pitch modes around the 5.5 seconds period (see section 5.0.3). Both controllers are tested with waves coming from 150, and 210 degrees direction. The results from the simulations are shown in table G.1. When loaded with head on waves, mostly the pitch modes are stimulated, whereas with waves which origin from 150 and 210 degrees both the roll and pitch are stimulated. The results show that the head on waves create larger forces in Offlead and Sidelead direction. The vertical load is the lowest for the head on waves. The unity checks show the offlead force as the bottleneck for the lifting operation (see figure G.2). It can be concluded that head on waves are conservative check for this wave period.

	PID			Linear		
Direction	**180°**	**150°**	**210°**	**180°**	**150°**	**210°**
Offlead [kN]	-1756	-1444	-1355	-2365	-1854	-1558
Sidelead [kN]	-468	-444	-436	-1299	-1273	-1259
Vertical [kN]	-19934	-21043	-21221	-21852	-22848	-22060

Figure G.1: Maximum forces on crane tip for different wave directions

	PID			Linear		
Direction	**180°**	**150°**	**210°**	**180°**	**150°**	**210°**
Offlead [kN]	1.00	0.83	0.77	1.35	1.06	0.89
Sidelead [kN]	0.39	0.37	0.36	1.09	1.06	1.05
Vertical [kN]	0.87	0.92	0.92	0.95	0.99	0.96

Figure G.2: UC of crane tip for different wave directions

Bibliography

[1] 2030 climate & energy framework | Climate Action. URL https://ec.europa.eu/clima/policies/strategies/2030_en.

[2] Clean Energy Can Meet 90% of Paris Energy-Related Goals | UNFCCC. URL https://unfccc.int/news/clean-energy-can-meet-90-of-paris-energy-related-goals.

[3] EQUIPMENT SHEET BOKALIFT 1 3,000 TONS DP-2 CRANE VESSEL CONSTRUCTION / CLASSIFICATION. Technical report. URL www.boskalis.com/offshore.

[4] Heavy lift vessels | Boskalis. URL https://boskalis.com/about-us/fleet-and-equipment/offshore-vessels/heavy-lift-vessels.html.

[5] SEAGREEN OWF PROJECT Suction Bucket Jacket Installation Motion Analysis Boskalis Offshore Wind Seagreen OWF Project Suction Bucket Jacket Installation Motion Analysis Revision Section Description of Hold Boskalis Offshore Wind Seagreen OWF Project Suction Bucket Jacket Installation Motion Analysis. Technical report, 2019.

[6] F. Greg Shinskey. PID control. In *Measurement, Instrumentation, and Sensors Handbook: Spatial, Mechanical, Thermal, and Radiation Measurement, Second Edition*, pages 91–1. CRC Press, 1 2017. ISBN 9781439848890. doi: 10.1201/b15474.

[7] J M J Journée and W W Massie. OFFSHORE HYDROMECHANICS Second Edition (2008) Based on Original Lecture Notes (2000) by. Technical report.

[8] Ljubinko Kevac, Mirjana Filipovic, and Aleksandar Rakic. Dynamics of the process of the rope winding (unwinding) on the winch. *Applied Mathematical Modelling*, 48:821–843, 8 2017. ISSN 0307904X. doi: 10.1016/j.apm.2017.02.023.

[9] T Lengkeek. Floating installation of offshore wind turbines in a single lift. Technical report, 2018. URL http://repository.tudelft.nl/.

[10] Orcina Ltd. OrcaFlex Manual. Technical report. URL www.orcina.com.

[11] Geert Meskers. A DAMPING TUGGER SYSTEM FOR OFFSHORE HEAVY LIFTS. Technical report, 2012. URL http://www.asme.org/about-asme/terms-of-use.

[12] peterdevos. OE5663 Dictaat 2008 v2_1. Technical report, 2008.

[13] Stian Skjong and Eilif Pedersen. Model-based control designs for offshore hydraulic winch systems. *Ocean Engineering*, 121:224–238, 7 2016. ISSN 00298018. doi: 10.1016/j.oceaneng.2016.05.036.