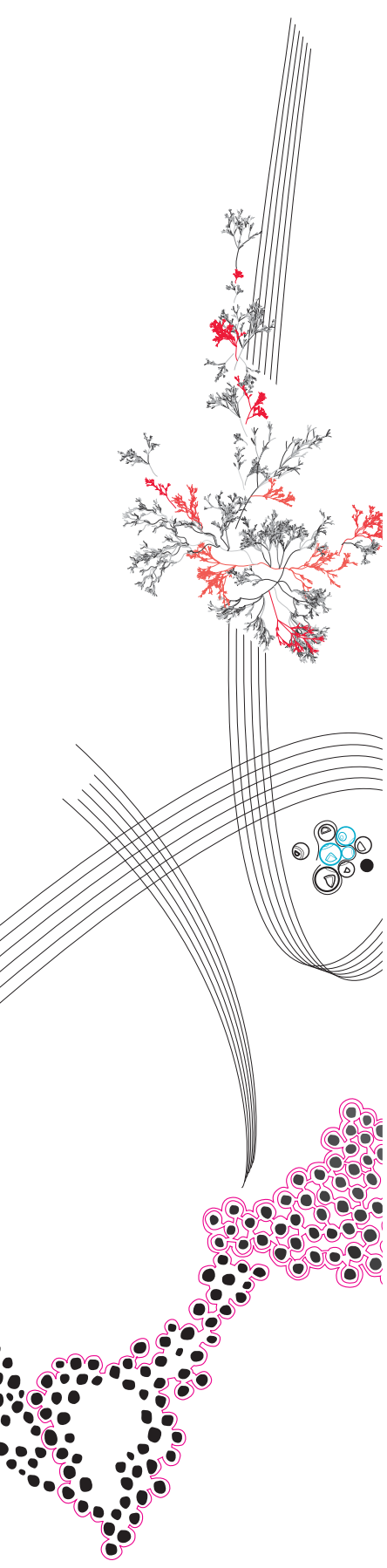


Pipe integrity and pipelay controller optimisation

Master's Thesis

Stan Hop





Pipe integrity and pipelay controller optimisation

Master's Thesis

Stan Hop

Graduation committee:

dr.ir. R.G.K.M. Aarts

dr.ir. J.P. Schilder

dr. J. Dasdemir

Ir. T. Koomen

August, 2024

CONFIDENTIAL

Applied Mechanics and Data Analysis (AMDA)
Mechanics of Solids, Surfaces and Systems (MS3)
Faculty of Engineering Technology (ET)
University of Twente

UNIVERSITY OF TWENTE.

The work presented in this thesis was (partially) performed at, and supported by the University of Twente. Their cooperation is hereby gratefully acknowledged.



The work presented in this thesis was (partially) performed at, and supported by Allseas Engineering B.V. Their cooperation is hereby gratefully acknowledged.



Orcina supported the work presented in this thesis by loaning a license for OrcaFlex. Their cooperation is hereby gratefully acknowledged.

Acknowledgements

I would like to express my sincere gratitude to everyone who supported me during my thesis. I especially would like to express my gratitude to:

Ronald Aarts and Jurnan Schilder for their shared expertise in dynamics and control as well as their friendly guidance during this unique graduation.

Allseas's systems department, particularly Teun Koomen, for his friendly guidance and shared expertise in control and software development.

My peers and dear friends, S. Buse, J.C. den Os and R.A.J. Burger for providing feedback and support during this thesis period.

Even though it seems cliché, I would like to thank my parents for their unconditional and never-ending support.

Thank you all for your contributions and support.

Stan Hop

Hierden, 2024

Abstract

In offshore pipelay, the combination of a real-time model of the vessel and pipe with an optimisation strategy is desired to maximise workability and minimise energy consumption during the laying process. Additionally, this strategy must ensure that the pipe's structural integrity is not exceeded.

From a literature review, it is concluded that an existing corotational FEM approach is the most suited to assess the structural integrity of the pipe and, where possible, give input for optimisation. This formulation has been further expanded, and the contact definition has been enhanced to allow continuous payout and pipe pulls. An adaptive feedforward algorithm has been developed to handle changing circumstances to optimise workability and minimise energy consumption during pipelay.

The model has been statically and dynamically compared with the industry standard OrcaFlex and shows good agreement. The adaptive algorithm concept has been demonstrated, but more validation tests based on past project data are recommended.

Contents

1	Introduction and Objectives	1
2	Background and data analysis on the pipelay process	5
2.1	Project definition and PIT	6
2.2	Real-time process	8
2.3	Tensioner control	10
2.4	Data analysis on pipe pulls	12
2.5	Concluding remarks	14
3	Literature Review	15
3.1	Pipe models	16
3.2	Tensioner models	19
3.3	Vessel models	19
3.4	Control strategies	20
3.5	Concluding remarks	21
4	Corotational FEM	23
4.1	Geometric nonlinear beam element	24
4.2	Vessel with stinger element	29
4.3	Contact modelling	30
4.4	Environmental loads	34
4.5	Solving statics and dynamics	37
5	Control architecture	41
5.1	Control goal	42
5.2	An adaptive control for changing circumstances	44
5.3	Concluding remarks	48
6	Results	49
6.1	Statics	50
6.2	Dynamics	53
6.3	Control	56
6.4	Concluding remarks	60
7	Conclusion and recommendations	61
7.1	Research Objective	62
7.2	Conclusion	63
7.3	Recommendations	64

Chapter 1

Introduction and Objectives

To facilitate the transport of large amounts of oil and gas, pipelines are installed on the seafloor. This process is called offshore pipelay. These, often enormous, pipes are laid from an installation vessel or barge. The two main installation procedures are the industry called S-lay and J-lay methods. The S-lay method, see Figure 1.0.1, is often preferred for its high production rate. This is because the firing line, in which new pipe segments of commonly 24 meters are prepared, preheated, welded, and checked, consists of multiple consecutive stations and, therefore, can be easily operated in S-lay with a horizontal configuration [1].

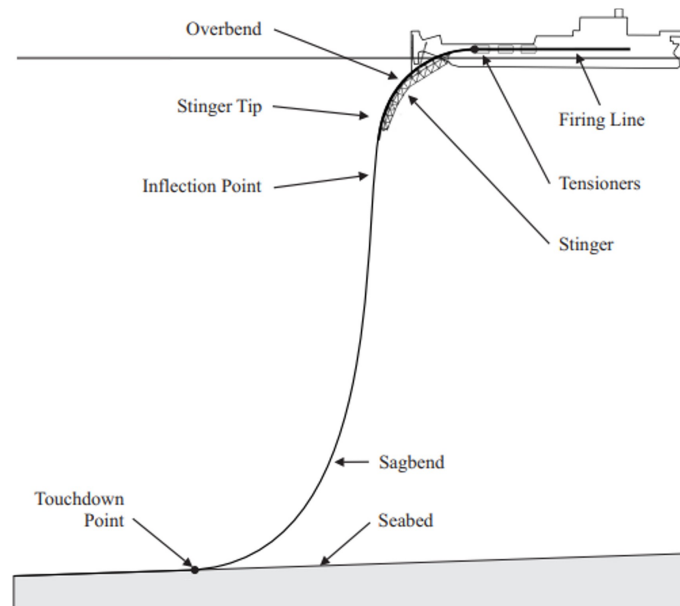


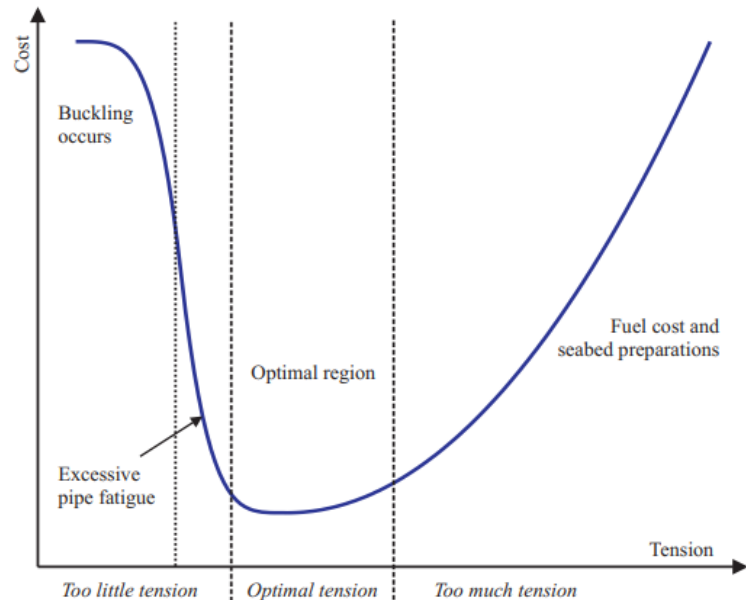
Figure 1.0.1: Schematic S-lay process (modified from [2])

After the firing line, the pipe transfers through the tensioners, see Figure 1.0.2a. These tensioners consist of large tracks between which the pipe is clamped, and axial tension is applied to hold the pipe from the seabed. The tension can not be too high, this requires excessive fuel and can lead to excessive strain. On the contrary, the tension must be high enough to prevent the pipe from buckling and excessive fatigue damage, as shown in Figure 1.0.2b [2]. After the tensioners the pipe rolls over rollerboxes which are mounted on a rigid structure called the stinger. This

ensures a minimum curvature on the pipe to avoid failure. This region is called the overbend. Further down the pipe the sag bend region is located, which is a bending of the pipe the opposite direction before the pipe reaches the seabed. Every time a new segment of pipe is welded to the pipe, the vessel moves 24 meters forward, and the tensioner pays out 24 meters of pipe.



(a) Tensioner [3]



(b) Tension-Cost diagram [2]

Figure 1.0.2: Tensioner (left) and Tension-cost diagram (right)

Multiple types of models are available for the modelling of the pipelay process. From more simple analytical models as the catenary equation [4] to very complex numerical models [5]. The simpler models often do not consider the complex, often non-linear phenomena encountered during pipelay. These phenomena are, among others, plastic material behaviour, contact interaction and large deformations. These phenomena are required to accurately predict the structural load during the pipelay process. Commercial software such as OrcaFlex [6] are available in which advanced models are implemented, but these also have their limitations.

Due to the complexity of the modelling and limited solving speed, most process automation is done by setting control parameters manually and conservatively. This is done mostly based on offline run simulations and the expertise of experienced operators. Examples are tension setpoints in the tensioners and position setpoints in the DP (dynamic positioning) system.

Thanks to computers' recent advancements and the optimization of numerical models, even complex models can now be solved in real-time. This opens up new possibilities for online optimization, allowing for dynamic adjustments based on actual loads rather than conservative offline predictions. This can be used to maximize workability and minimise energy consumption.

This thesis aims to investigate what better models can be used and/or what other, potentially more effective, control strategies can be executed during the pipelay process. New in this work is an enhanced contact formulation based on an existing corotational FEM model to allow for large payouts and pipe pulls. Additionally, a new adaptive feedforward algorithm is introduced to optimize the interaction between vessel movements and pipe payout as a first step to maximizing workability and minimizing energy consumption. Both the validity of the model as well as the working principle of the controller is substantiated by simulation.

Objectives

The goal of this thesis is twofold. The first is to create a model that provides a better understanding or a real-time estimate of the limits of the pipe. This limit is highly dependent on the water depth, pipe geometry, stinger configuration, coating, and material which should all be taken into account. The second goal is to come up with a more efficient control strategy using this model and/or a different interaction between the vessel and tensioner. This should be researched and substantiated with simulations, preferably incorporating past project data. From those main two goals, the following sub-objectives can be set up:

- 1: Identify which models are already present/currently used within Allseas.
- 2: Search for alternative, potentially more detailed, models that can be used.
- 3: Create a structural model of the pipe, tensioner and vessel that includes the effect of water depth, pipe geometry, stinger configuration, coating and material.
- 4: Determine the control strategy/algorithm that obtains the best mix between:
 - Maximum workability
 - Minimal energy consumption
 - While being constrained by the structural integrity of the pipe.
- 5: Validate the correct implementation of the model and its accuracy by comparing it with commercial software. Additionally, validate that the controller performs as intended.

Outline

This thesis is divided into the following chapters to achieve these five sub-objectives. The next Chapter provides a more extensive background on the pipelay process by analysing data from two past projects. Chapter 3 summarises the relevant literature to explore the possibilities of models and control strategies. After this, a corotational FEM approach is worked out in Chapter 4, which is combined with a vessel and tensioner model. Following in Chapter 5 an adaptive feedforward algorithm is described to optimize the pipelay process based on the real-time model. Next, in Chapter 6, static and dynamic simulations are performed to validate the model and to show the controller's performance. After this, in Chapter 7, conclusions are drawn, and recommendations are made on the presented work.

Chapter 2

Background and data analysis on the pipelay process

In this Chapter, the pipelay process is explained in more detail with a focus on the operation procedure within Allseas. Information is mainly gathered from internal documents and examining different Allseas staff. Among others are field Engineers, System Engineers, Pipelay Engineers, and operators. Therefore, no extensive references will be made in this chapter unless they are obtained elsewhere.

The pipelay project is divided into three main phases within Allseas based on the timeframe. Starting with a project definition phase and a self-proclaimed PIT (Pipelay Integrity Tool) phase. The last phase encompasses the real-time process of laying pipe.

In Section [2.1](#), the first two phases will be briefly explained. In Section [2.2](#), the real-time process and the operators involved will be discussed in more detail since this phase is most relevant to this thesis. Especially relevant is the tensioner control, which is elucidated further in Section [2.3](#). After that, Section [2.4](#) is devoted to an analysis of past project data to visualize how actual projects went. In the last Section, conclusions are drawn.

2.1 Project definition and PIT

From the first client contact to the first pipe being laid a lot has to be engineered and prepared. This phase is called the project definition. Additionally before the first meters of pipe can be laid an additional checking phase is relevant in which it will be decided if the pipelay process can continue for the upcoming hours, the PIT phase. Both will be briefly explained.

Project definition

In this phase, the project is set up. From deciding what vessels to use, which type of pipe to lay as well as the complete supply chain of the pipe segments to the vessel during operation is determined. For this thesis, it is relevant that numerous structural simulations are done to ensure a correct installation without failures. These simulations were mostly done using Offpipe [7]. However, Orcaflex has been used more recently.

Additionally, in this phase, the pre-laid survey takes place, in which the route for the pipe and the water depth along this route are determined. Most pipes start at the coast in shallow water and go through deeper water in the middle of the sea. As seen in Figure 2.1.1, at different water depths, different loads are experienced by the pipe. In shallow water, vessel motions result in significant tension fluctuations since the pipe has a more horizontal orientation, and axial stiffness is dominant. In deeper water, the stinger configurations are set to encompass up to 90-degree bends, and due to depths up to two kilometres, the resulting dominant bending stiffness is generally lower. Therefore the **vessel's motion - tension fluctuation coupling** is less or even negligible in deeper water. More about this difference is presented in the following Sections.

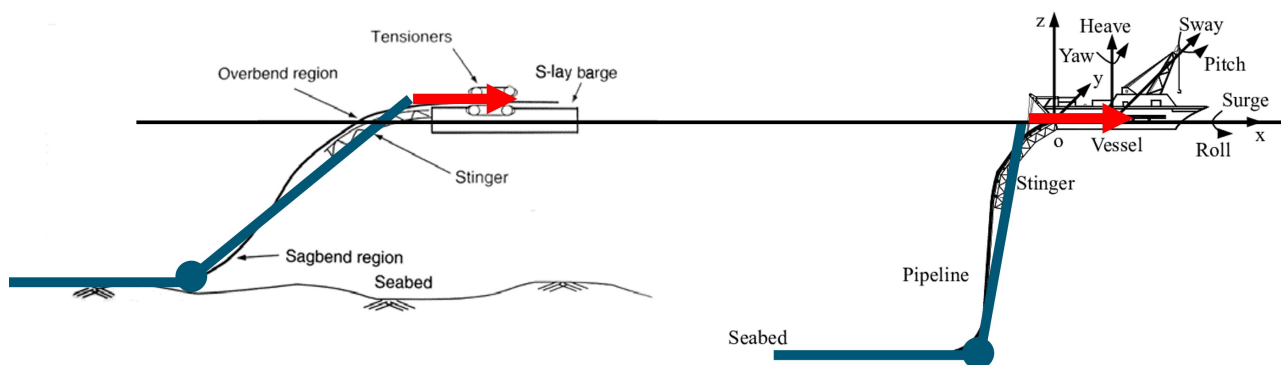


Figure 2.1.1: Sketch of shallow water (left) and deep water (right) pipelay

PIT

PIT is an abbreviation for Pipe Integrity Tool. This tool is developed within Allseas which uses the upcoming weather forecasts from the survey department and runs multiple Orcaflex simulations based on this to decide if the pipelay process can continue in the next few hours.

This tool comes down to transferring the weather forecast to a sea state (wave height, wave period, and wave direction) that is likely to occur. In OrcaFlex waves are computed from this sea state. Based on these waves vessel motions are computed by using the vessel's RAO (response amplitude operator). Since the generated waves are based on a spectrum, the simulation will be run for 3 hours. Using the DNVGL-OS-F101 standard [8], the loading of the pipe will be assessed using unity checks. Table 5-15 in this standard lists several installation limit state criteria. For the S-lay method, it states a unity check for the overbend and sag bend as:

$$UC = \left[\gamma_M * \gamma_{SC, LB} * \frac{|M_{sd}|}{\alpha_c * M_p} + \left(\gamma_M * \gamma_{SC, LB} * \frac{S_{sd}}{\alpha_c S_p} \right)^2 \right]^2 + \left[\gamma_M * \gamma_{SC, LB} * \frac{p_e - p_i}{p_c} \right]^2 \leq 1 \quad (2.1)$$

in which:

- γ_M = Material resistance factor [-]
- $\gamma_{SC, LB}$ = Safety class resistance factor [-]
- α_c = Characteristic flow stress ratio [-]
- M_{sd} = Occurring moment [Nm]
- M_p = Plastic moment limit [Nm]
- S_{sd} = Occurring true axial force [N]
- S_p = Plastic true axial force limit [N]
- p_e = External pressure [Pa]
- p_i = Internal pressure [Pa]
- p_c = Collapse pressure [Pa]

This equation of equivalent loading is based on two failure criteria. The first is excessive strain consisting of the bending moment combined with axial force, and the latter is the contribution due to external overpressure.

Calculating this UC (unity check) at all points in the pipe and taking the maximum in the overbend and sag bend gives a single indication of the load relative to its structural integrity for both pipe regions. With a real-time advanced numerical model in place, these UCs could be calculated live and form a control input.

2.2 Real-time process

The previous section describes the preparation and prediction of the pipelay process. When it is decided that pipelay can be executed close cooperation is required between mostly the operators of dynamic positioning (DP), survey, the tensioner and the firing line. The different departments and their mutual coordination will be briefly discussed based on [Figure 2.2.1](#).

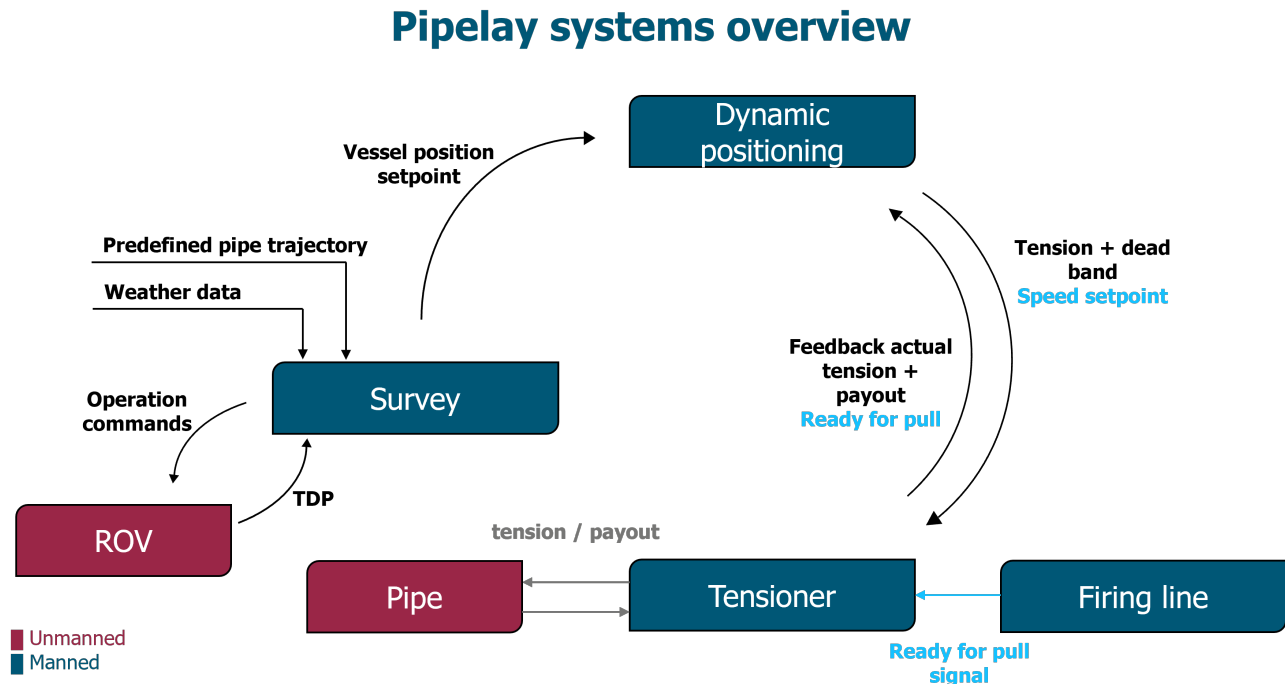


Figure 2.2.1: Overview of the different stations involved in pipe lay

Survey

The process begins at the survey department. Based on the PIT report, the survey operators decide whether the process can be executed. The vessel position setpoint is given to DP based on, among other information sources, the predefined pipe trajectory and the current touch-down point (TDP) of the pipe. The latter is commonly defined as the position of the remotely operated vehicle (ROV) [2]. This ROV is kept close to the pipe to monitor the region where the pipe is first in contact with the seafloor. Therefore, its position can be used as the TDP. Also, information from the in Section 2.1 explained pre-survey is available, such as the water depths along this predefined pipe trajectory, indicated with KP in [Figure 2.2.2](#) as an example. To end up with a pipe sufficiently close to this predefined pipe trajectory, the required vessel trajectory is calculated, as will be described in Section 3.4. From this vessel trajectory the vessel position setpoints are generated.

Dynamic positioning

The DP operator controls the vessel's position and ensures that the vessel follows the forwarded setpoints by survey. The DP system rejects motion due to waves and other environmental disturbances by setting the individual thrusters' direction and speed. Also, the tension in the tensioner is considered since this force also needs to be cancelled, or rather generated by the vessel and is

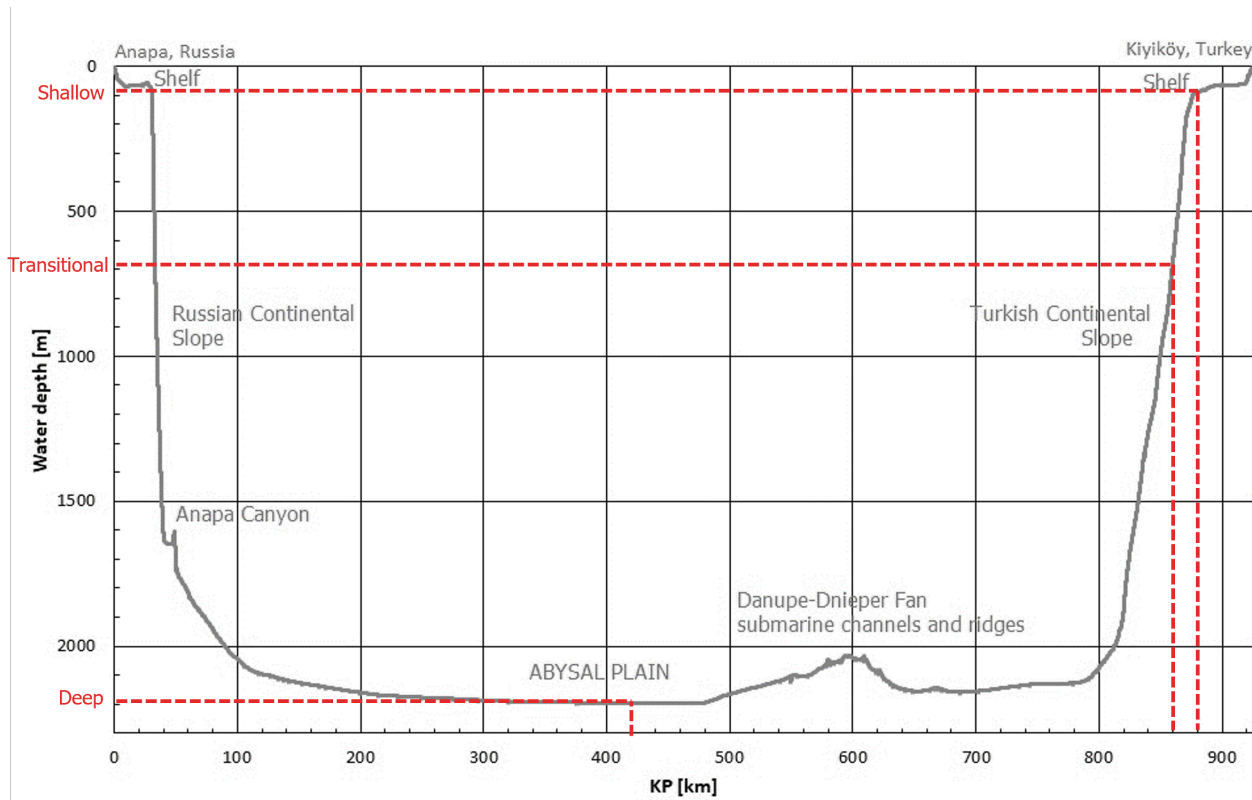


Figure 2.2.2: Waterdepth along the pipe route Turkstream (internal document)

communicated to the tensioner operator. Additionally, when the DP operator receives the "ready for pull" signal, it will move 24 meters in shallow water. When in deeper water, it will move more towards a constant speed, as elucidated further in the next Section. This movement profile is also communicated with the tensioner operator.

Tensioner and firing line

The control strategy is discussed in more detail in the next Section due to its high relevance. The coordination between the tensioner and firing line operators comes down to waiting for the "Ready for pull" signal from the firing line and forwarding this to DP in order to execute a pipe pull. During this pipe pull the tensioner will pay out 24 meters of pipe. The firing line consists of the consecutive stations in which the actual joining of pipes is realised. Depending on the type of pipe and coating, these consist of (multiple) preheating, welding and coating stations, each 24 meters apart and suspended from the ship. This suspension allows small movements in the longitudinal direction of the pipe. When the tension rises or drops too much, the tensioner will pay the pipe in or out to compensate for this. These small movements are also applied to the stations such that there is no relative motion between the welding stations and the pipe. Although these movements disturb the welders, the welding can continue during these small movements. When a station is ready with its task, a light will turn on, visible to the tensioner operator. When the light of all stations is turned on, it indicates the "ready for pull" signal.

2.3 Tensioner control

As explained, the tensioner can pay pipe in or out with respect to the vessel. This can be done in two ways. The first is due to the tension feedback loop. This controller is fed with tension measurements of loadcells, and when the measurements are outside of the deadband (a certain percentage above and below the tension setpoint), the pipe is paid in or out. The second way is using a feedforward signal. Both are explained more extensively in this Section:

Tension feedback control and pipe pull

The tensioner pays pipe in or out with respect to the ship only when the deadband is exceeded. For example, these are minor payouts for large waves or vessel drift to prevent overloading the pipe. Additionally, when the firing line gives the "ready for pull" signal and a new pipe segment is welded onto the pipe, the vessel will perform a pull. This is a major payout of 24 m. This payout can be achieved in **shallow water** by this tension feedback control automatically since the tension strongly depends on the position of the vessel. Therefore, when a pull is made, the vessel accelerates up to a certain velocity at which it remains until it decelerates again, resulting in a 24 m displacement. Since the tension rises quickly above the deadband, the tension controller will pay the pipe out until the 24 m is reached and the tension comes within the deadband again. This behaviour is visible in [Figure 2.3.1](#). In subfigure a and b, the tension payout nicely tracks the vessel's position, and in e, the corresponding speed graph shows a mountain-like profile.

Pipe ahead feedforward control

For **deep water**, the tension fluctuates hardly when the vessel moves 24 m. Therefore, a "pipe ahead" feedforward is built into the tensioner control scheme. This pipe ahead function has a velocity feedforward reference profile with a certain constant acceleration until a constant payout speed is reached. This velocity is decelerated with the same acceleration after a distance such that the same 24 m payout is reached. If the tension reaches outside the deadband, the tension control would overrule this feedforward with more or less payout/pay-in, but in deep water, this is hardly the case. During this procedure, the vessel can move at an almost constant speed and is, therefore, fuel-efficient. This behaviour is also visible in [Figure 2.3.1](#), subfigure d. The deep water speed graph, seen in subfigure e, has a more steep block-like profile. This clearly indicates the use of the pipe ahead feedforward. The steeper increase indicates a higher acceleration, which is possible since the tension will not change that much and thus will not "shock/disturb" the DP system nor exceed the deadband. Therefore, the pipe pull duration is shorter, and the step in the pull is faster compared with shallow water. Also, it is visible that the vessel position can be set to a constant and that the "difference" between the vessel position and tensioner payout does not result in tension outside the deadband since otherwise, the constant tension control would have overruled.

Transitional water depths

As indicated the pipe is often laid from shallow to deepwater in a continuous manner as shown in [Figure 2.2.2](#). Therefore there will come a transitional phase where the "position error - tension increase" coupling becomes less. In this phase, the constant tension control will not yield a clear 24 m pull. But, the steep pipe ahead profile and constant sailing speed will result in deadband crossings in which the tension feedback control will intervene, and therefore, this will also not yield a clear pull. Currently, experienced operators will set a mild version of the pipe ahead in combination with tension feedback control. Also, DP will sail in a milder step, as can be seen in [Figure 2.3.1](#), subfigure c. This behaviour is very dependent on the type of pipe, weather condi-

tions, and the vessel. Therefore, this requires a lot of expertise and experience of past projects by the tensioner operator and is therefore not optimised.

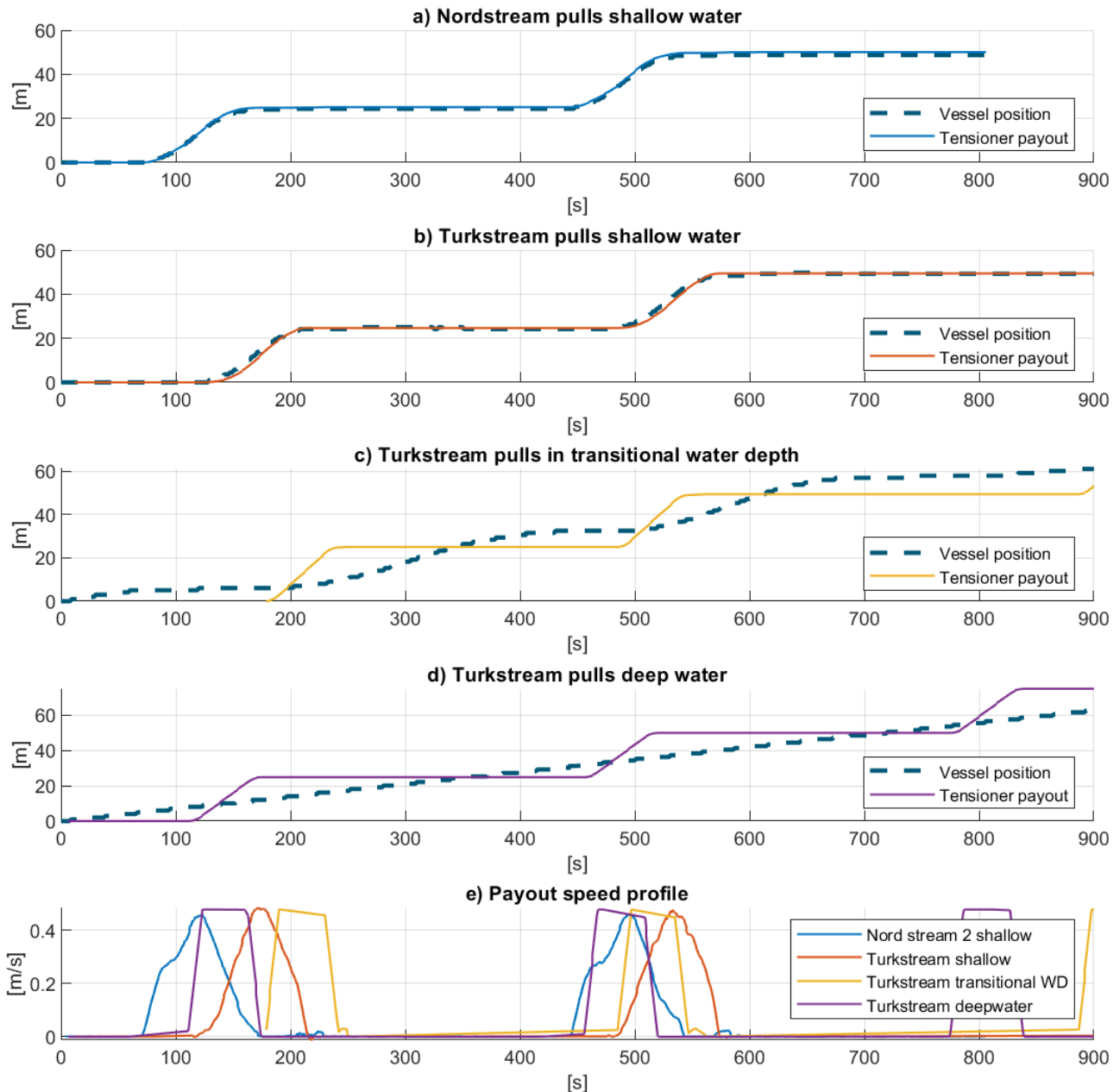


Figure 2.3.1: Different pull profiles

Additional tensioner control features

The above subsections describe how, for different scenarios, a pipe is being laid. Three more features worth mentioning in the control system have additional capabilities. First, a motion compensation feedforward can be enabled. The inertial measurement unit onboard registers all vessel motions. The pitch and heave motions can result in a significant tension change due to bad weather. Hence, there is a feature which translates the pitch and heave motion to a pipe payout to compensate for this motion, which can be added to the tensioner as feedforward. The second and third features are a manual and break operation mode. The break will lock the motors, and thus, the pipe will not be paid out. In manual mode, the payout can be set on the control panel, which the tensioner will execute. This can be used in exceptional circumstances.

2.4 Data analysis on pipe pulls

In the previous Section, the individual pulls are analysed, and their control strategies for different scenarios are explained. With the optimisation goal of this research in mind an analysis will be performed on multiple pulls. For the Nordstream, a shallow water project, as well as Turkstream, a deep water project. a full day of data is considered. For Nordstream, the data is shown in [Figure 2.4.1](#). Subfigure a shows the vessel position and payout resulting in 5.5 kilometers of pipe laid in 24 hours. Each spike in subfigure b indicates a pipe pull, 201 in total. In shallow water, the water depth can be measured, as shown in subfigure c. For deep water, this is not possible and the pre-survey graph is used, as shown in [Figure 2.2.2](#). [Figure 2.4.2](#) has the same graphs but zoomed on the first 1200 seconds.

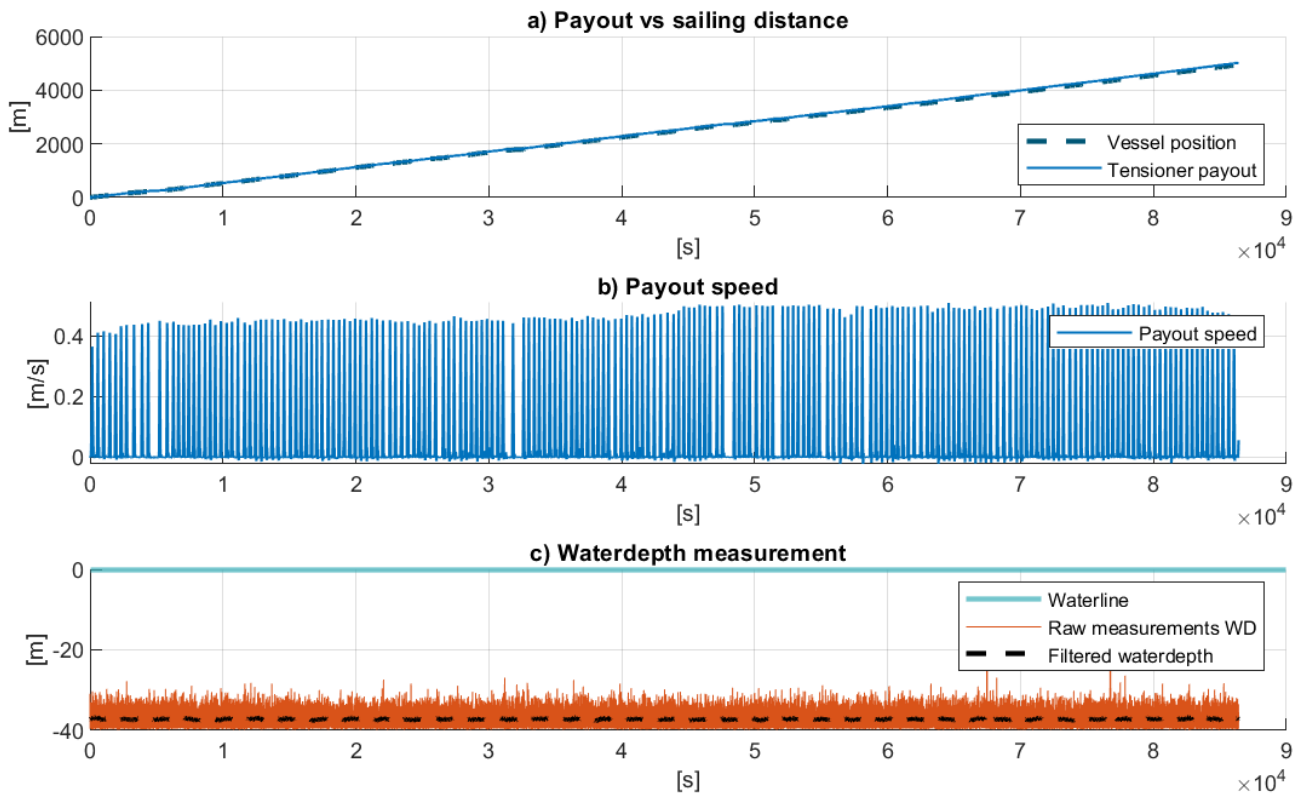


Figure 2.4.1: Nord stream 2 measurements 3th of July

For possible optimisation strategies that affect multiple pulls, it is critical to know how consistent the pulls are. Consistency implies the time it takes the welders to weld a pipe segment and a new pull can be executed. Therefore, the data for the Nordstream shallow water day and the Turkstream deep water day is analysed. The first step is to detect a pull start and pull end. Since there are some discrepancies in the payout speed measurement, as seen in [Figure 2.4.2](#) subfigure b, the signal is filtered, and a threshold is added to detect pull starts and pull ends. This is done for both days, and the time between pull-start and pull-end is called "pull duration", and the time between the pulls in which a new pipe is welded is called "welding duration".

If this data is sorted on duration from lowest to highest subfigure a of [Figure 2.4.3](#) is obtained. When deciding if predictive control optimization would be an option, it is most relevant to look into the deviations between the welding durations. If this is very consistent, the vessel can already use this information. Therefore, visible in subplots b and c of [Figure 2.4.3](#) a 90 percent bounding box is drawn to show in what distribution most pipes are welded.

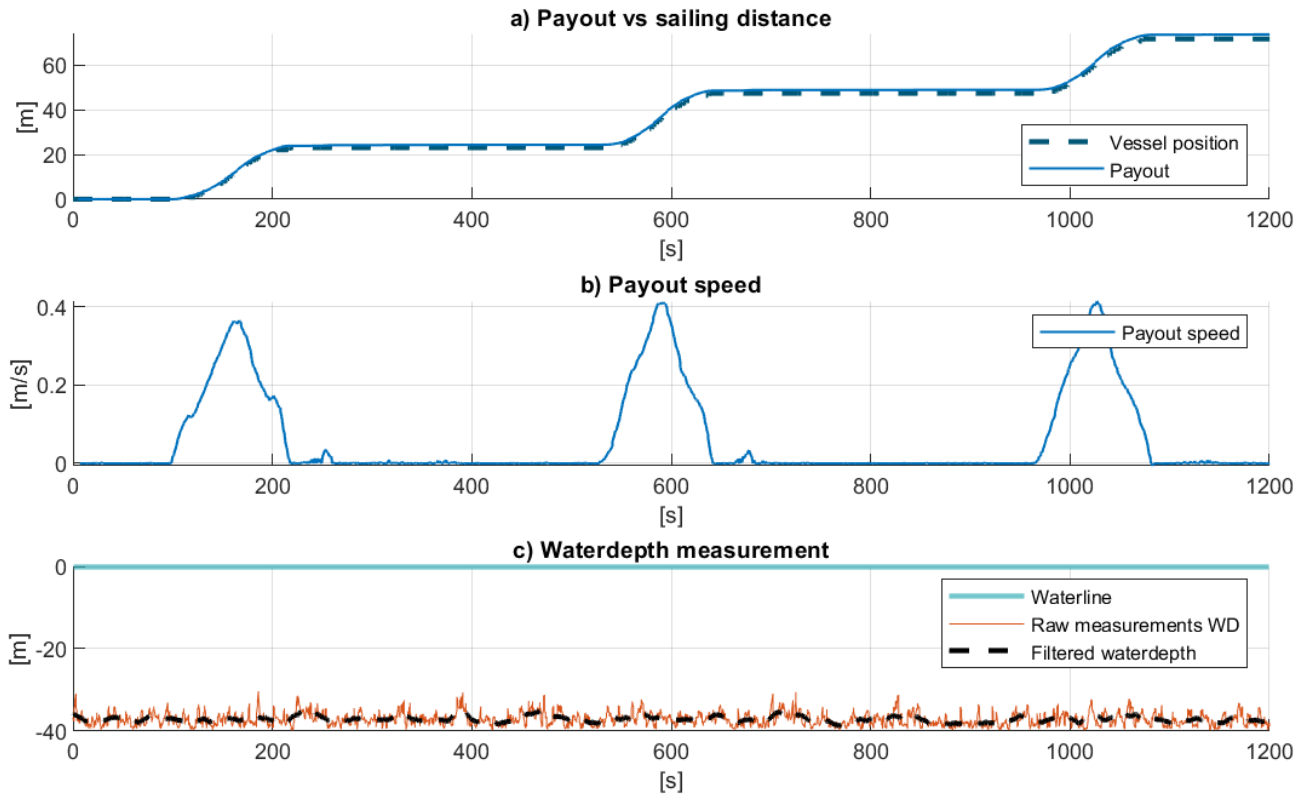


Figure 2.4.2: Nord stream 2 measurements fragment

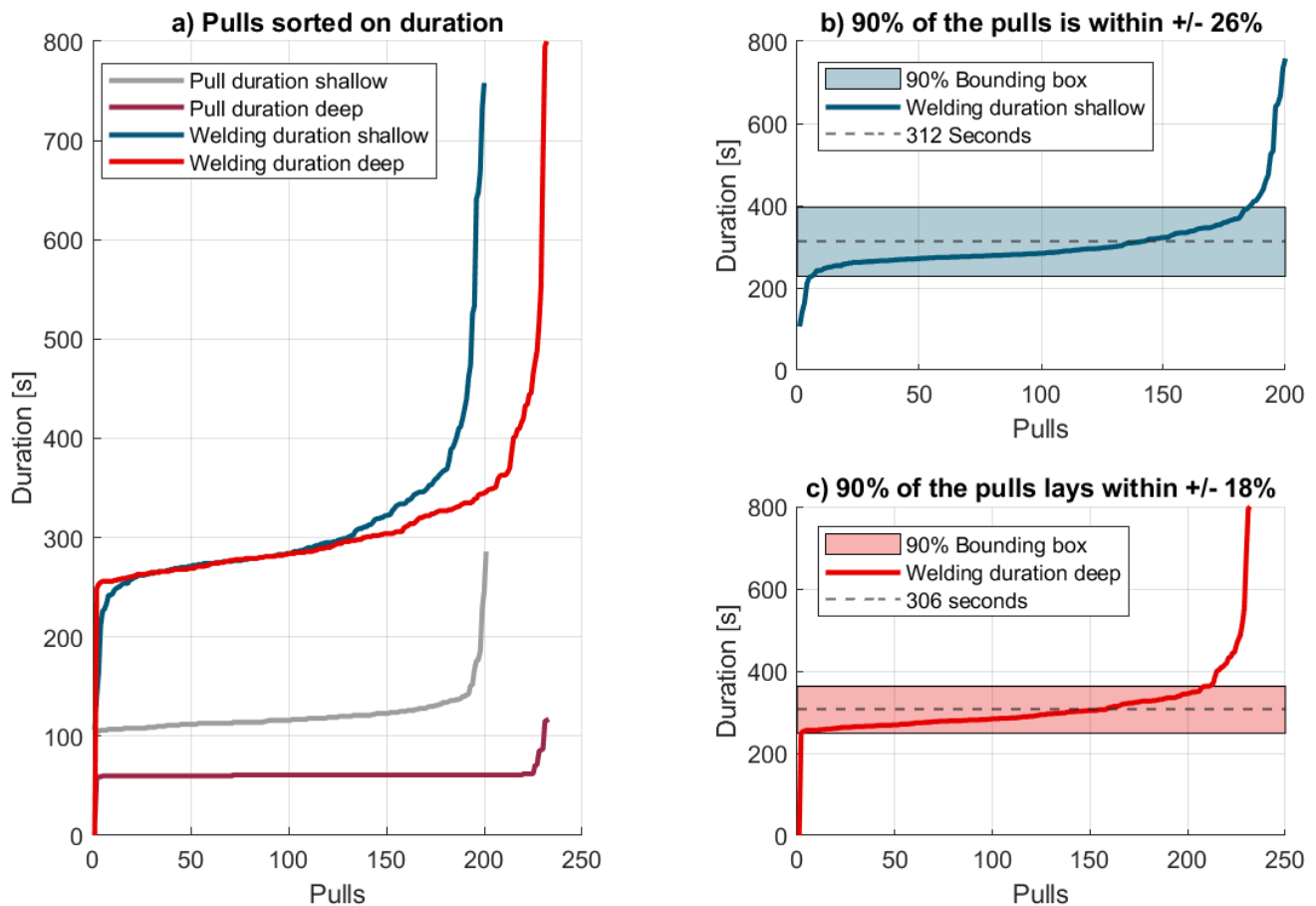


Figure 2.4.3: Duration of pipe pulls

2.5 Concluding remarks

The current models utilize OrcaFlex simulations that are run offline. Live feedback control is then implemented, based on predetermined tension setpoints with a specified deadband. The weather forecast, in combination with the intended deadband, is run through the PIT to assess if the structural integrity is not exceeded based on unity checks from the appropriate DNV-GL standard. Additionally, a feedforward is used to achieve pulls in deeper water, in which settings are set by experienced operators.

A secondary conclusion that can be drawn is that due to the current procedure of using a forecast and simulating **worst-case** scenarios, a very robust operation window is considered and either marked acceptable or not. This way, no optimisation is performed during the operation itself. Additionally, no calculation of the **actual** occurring loads is made.

Regarding optimization possibilities, the data analysis concludes that the welding time is not very consistent. Therefore, an adaptive control strategy is more suitable than a predictive control strategy.

Chapter 3

Literature Review

To search for alternative, potentially more detailed models a literature review is conducted regarding structural modelling of a pipe, modelling of a vessel and tensioner control. For the modelling and control, the following points are important:

- **Geometric non-linearity:** Since the pipe is welded straight in the firing line, bent up to 90 degrees in the overbend and bent back in the sagbend sometimes a few kilometres further, we can conclude that the model needs to allow for rigid body motions and large displacements.
- **Plasticity:** Some projects allow plasticity in the overbend during the laying process so plasticity should be modelled.
- **Structural loading:** Structural integrity of the pipe must be conserved thus the model should model the loading in the whole pipe, or at least in the critical locations accurately.
- **Computational time:** Since the model is the input for a controller, real-time or almost real-time simulation is required.
- **Constraints/contact:** The contact of the pipe with the tensioner, stinger and seabed must be adequately modelled.
- **Pipe pulls:** Since the controller is used not only for keeping position but also for performing pulls the model should allow for this.

3.1 Pipe models

For the modelling of a pipe, multiple models are found in literature. The most relevant will be briefly discussed, and for further details, refer to the cited literature.

Catenary

One of the simplest models is that of a string loaded with its own weight, otherwise called the catenary equation [4]. The curve is described by the horizontal tension H , the submerged weight w_s in cartesian coordinates as:

$$y(x) = \frac{H}{w_s} \left(\cosh\left(\frac{xw_s}{H}\right) - 1 \right) \quad (3.1)$$

This formulation can be extended to the stiffened catenary by adding bending stiffness, but it still does not model any dynamics. Additionally, it completely neglects the overbend, as seen in Figure 3.1.1a. However, it is a lightweight model that could serve an initial guess.

Hyper redundant manipulator

Another approach is a robotic pipe model [4]. In other words, it consists of many rigid links with the mass $\mathbf{M}_l$ lumped in the middle of the links, joined together as a string, see Figure 3.1.1b. In the joints, artificial torsional springs $\bar{K}_\tau$ represent the bending stiffness. It is therefore described by the classic equation of motion in joint space $\vec{q}$ as:

$$\mathbf{M}_l(\vec{q})\ddot{\vec{q}} + \mathbf{C}(\vec{q}, \dot{\vec{q}})\dot{\vec{q}} + \mathbf{E}(\vec{q}, \dot{\vec{q}})\vec{q} + \bar{K}_\tau(\vec{q}) + \vec{g}(\vec{q}) = \vec{\tau} \quad (3.2)$$

where Enviromental loads $\mathbf{E}(\vec{q}, \dot{\vec{q}})$ such as drag and added mass as well as gravity/buoyancy $\vec{g}$ can be added. The loads of the vessel $\vec{\tau}$ only act on the last (few) joints, which makes it a "hyper redundant manipulator". Although more effects are introduced compared with the catenary, it still does not accurately represent the pipe's structural integrity due to the artificial stiffness. Also, including the stinger contacts will be nontrivial due to the inverse kinematics from stinger coordinates to joint space. It is, however, still a very fast model, which is desirable. Similar models in cartesian coordinates instead of joint coordinates are also available [9]. These models are often still 2D approximations, and elements commonly still consist of artificial fit stiffness.

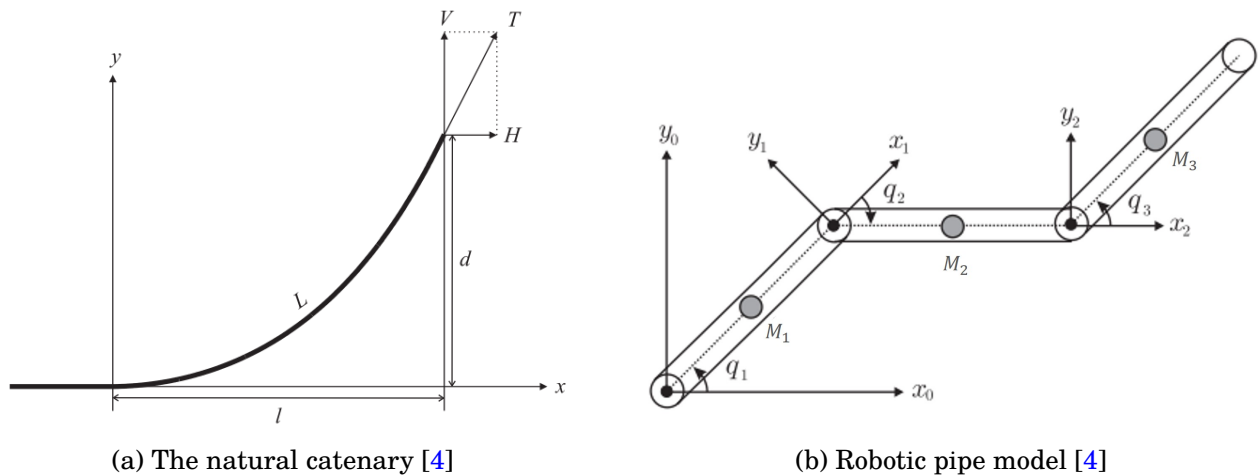


Figure 3.1.1: Schematics of natural catenary (left) and robotic pipe model (right)

OrcaFlex's line model

OrcaFlex has a 3D line model to represent a pipe. This model splits the line into nodes and segments. All inertia-related properties, such as mass, weight, buoyancy, and drag, are lumped in the nodes. Stiffness and damping properties of the pipe are again artificially modelled by springs and dampers which act between different coordinate frames for the nodes n_i and segments $s_{i,i}$ as indicated in Figure 3.1.2. This way of modelling requires five steps in a particular order to accurately calculate the forces and moments in the nodes. This is described in [10] but omitted here. Although artificially modelled, OrcaFlex can include variable data sources, such as a non-linear curvature-bending moment relation. This way, plastic material behaviour can be modelled.

Solving in OrcaFlex requires first solving the statics. From this static equilibrium, the dynamics can be solved. In principle for statics, a Newton method is used, and for dynamics, if implicit is selected, the generalized- α method is used. The procedure is more involved depending on the types of elements included in the model. These elements are vessel, winch, line, and others, but for that is referred to OrcaFlex's documentation [10].

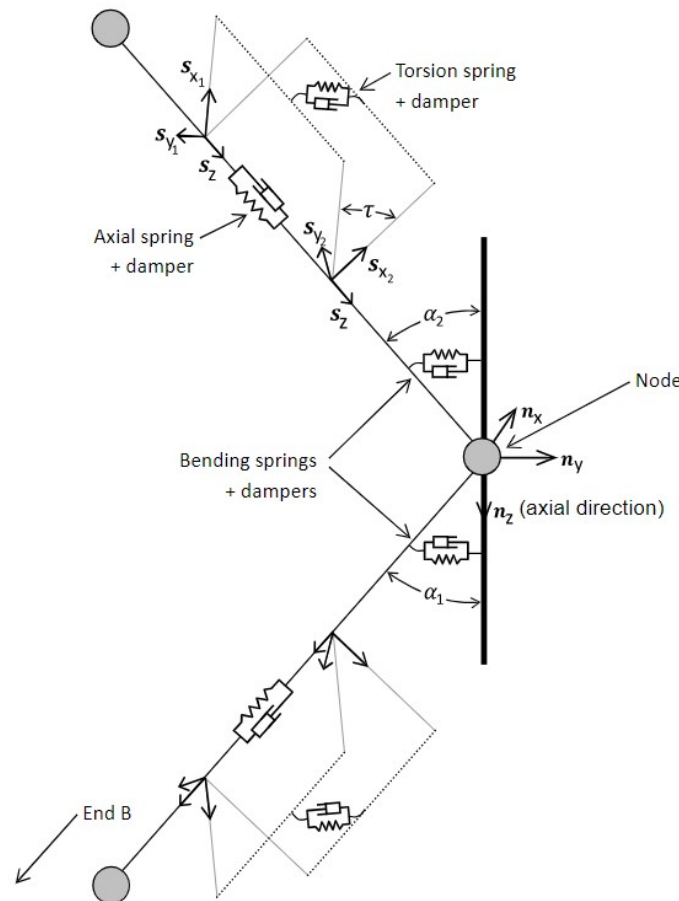


Figure 3.1.2: Schematics of OrcaFlex's detailed line model [10]

Advanced numerical models

More advanced numerical models are also available. The main downside are the high computational cost and difficulty involved in proving stability. For modelling of a flexible pipe the following methods are available in literature: The inertial frame formulation (**IFF**), the corotational frame formulation (**CFF**) and the floating frame formulation (**FFF**), all respectively drawn from left to right in Figure 3.1.3. More methods are available, especially for beam elements, but those are omitted here.

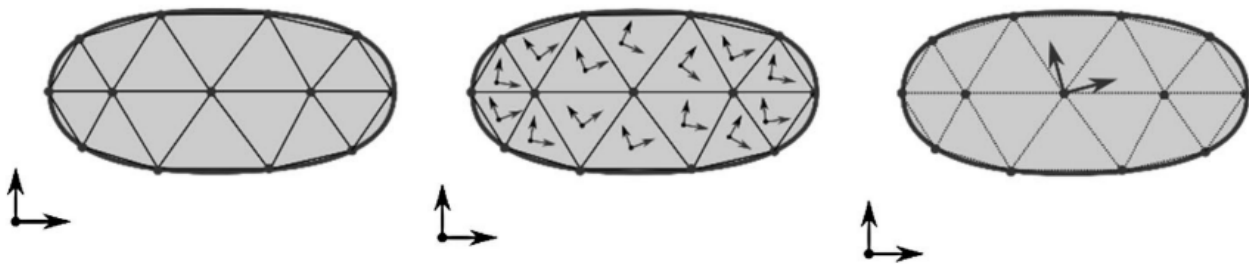


Figure 3.1.3: Schematics of coordinate frames in IFF (left), CFF (middle) and FFF (right) [11]

IFF is based on the nonlinear Green-Lagrange strain definition, and all the nodal coordinates are the DOFs expressed in the inertial frame. Due to this definition, no distinction can be made between rigid rotation and flexible behaviour. Due to the large displacements during pipelay, this is not an appropriate method.

The CFF, compared to IFF, provides all elements with a corotational frame for large rigid body motions. Superimposed on this rigid body motion is a Cauchy strain model. This is a linear elastic strain model that models elastic behaviour. Although the corotational frame is used, the absolute nodal coordinates are part of the DOFs, and thus, constraints can be enforced directly on them. The latter is also the case for IFF.

The FFF, however, can be seen as an extension of rigid multi-body dynamics in which local deformations are described using deformation shapes. The latter allows powerful model order reduction techniques, but the interface points are not part of the DOFs, and thus, Lagrange multipliers are required to solve the system. This leads to extra equations to solve. However, it is demonstrated that these extra equations can be eliminated when using superelements [12] as well as specifically for a chain-like topology such as a pipe [13].

Gullik Anthon Jensen [14] and Frans H. de Vries [5] both wrote their PhD thesis on the topic. Gullik also briefly looked at the control perspective and coupled his PDE model (evaluated with FEM) with a PID controller. De Vries's more recently published thesis worked out the corotational formulation for a geometric non-linear beam element. This way, the structural loading is part of the solution process, the roller-box contact is implemented using the penalty method, and simulations show that real-time solving is possible. Additionally, plasticity was incorporated in this model.

Results, however, still need to be verified with the more recently used Orcaflex [6]. Additionally, his contact procedure works only with pipe nodes located at the roller boxes. During a pipe pull, these nodes move. Therefore, either ALE-elements must be incorporated [15] or another workaround should be found.

3.2 Tensioner models

Some approaches are described in literature to model the tensioner, although these are limited. Two main parts are described, the dynamics of the tensioner itself and the control scheme. Regarding the dynamics, the tensioner pays the pipe in and out by rotating a motor that drives a track in which the pipe is clamped. For more details, refer to [16]. The components are schematically shown in Figure 3.2.1. Here, the tension setpoint P_{set} is compared with the measured tension P in a regulator, which forms reference torque for the motor T_{ref} . Based on this reference torque, an actual motor torque T_M is generated, which is put into a reducer. This results in T_R , and with an effective diameter specified by the track geometry, results in the tension at the pipe again. Due to these components, the tensioner has inertia, resistance and stiffness. These dynamic properties are often neglected or generalized with a stiffness and damping factor at the end of the pipe. Regarding control, multiple schemes are found in literature. Simple PID controllers are used to keep the tension at its setpoint, or experiments are conducted with Fuzzy logic [17]. Also, energy-saving techniques are investigated regarding the hydraulic pressure system [18], but this is all for low level control.

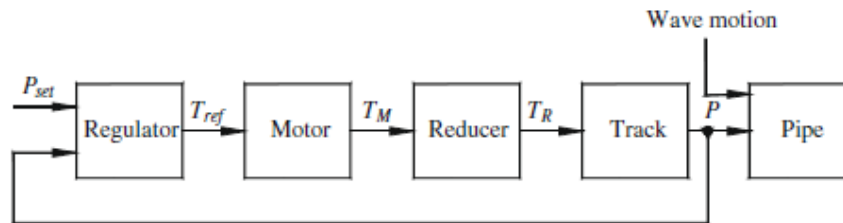


Figure 3.2.1: Block model of the tensioner [19]

For the high-level scope of this thesis, it is more logical to use a more top-level tensioner model in which the control system can set the payout. Keeping a maximum payout speed and maximum tension in mind. In OrcaFFlex a winch element can be used which is a wire between the pipe and a stationary point in the firing line. This winch element is either length-controlled or tension-controlled. In the first mode the length is set during the simulation and the resulting tension is calculated. In tension control, it is the other way around. The required payout is calculated and instantly applied to keep the tension either constant or in the deadband. De Vries describes a payout proportional to the displacement of the stinger tip and implements this as a continuous function, thus neglecting a deadband [5].

3.3 Vessel models

Literature agrees that in the scope of pipelay, the vessel can be modelled as a single rigid body with 6 DOFs in three dimensions [20, 5, 6]. For the motions of the vessel, response amplitude operators are often used. These transfer functions map waves, characterised by significant wave height, wave period and incoming angle to vessel motions. These RAOs are experimentally determined for waves at different angles and periods [20]. These experiments are commonly done with a stinger but not with a pipe attached to the vessel. It can be doubted if this simplification is justified since a very stiff pipe will resist vessel motions. Fortunately, in the scope of this thesis, the vessel motions do not have to be calculated on a stochastic wave spectrum as is the case for the PIT, but on the actual vessel motions measured by the IMU onboard, as is done in the monitoring tool described in [21].

3.4 Control strategies

Literature has many separate control strategies for pipelay. First, the DP system takes care of the vessel's positioning. This system has multiple modes, mainly to stay in position. Second, "The pipelay control problem", as formulated by Jensen, which calculates the vessel setpoints on sea level, to end up with the pipe at the intended trajectory on the seafloor as can be seen in [Figure 3.4.1](#) [2]. The third strategy is tension control in the tensioners. This controller aims to keep the tension constant or at least within a deadband around a setpoint. PID style or fuzzy logic controllers are mainly used. As already explained, the former is used within Allseas, although for deeper water, a feedforward is required, which needs to be set by an experienced operator. Additionally, the effect of a PI controller on the tension has been verified using Orcaflex by Simanjuntak [22], see [Figure 3.4.2](#). This setup is appropriate for simulation since the input for the controller is only the top tension, which onboard can be measured by sensors.

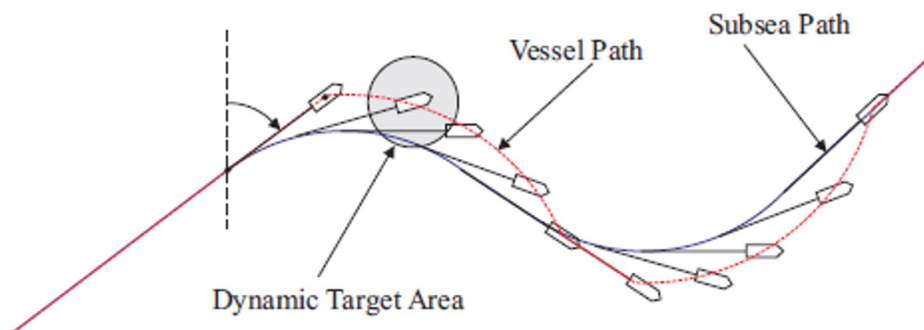


Figure 3.4.1: Vessel trajectory in order to achieve pipe trajectory (modified from [2])

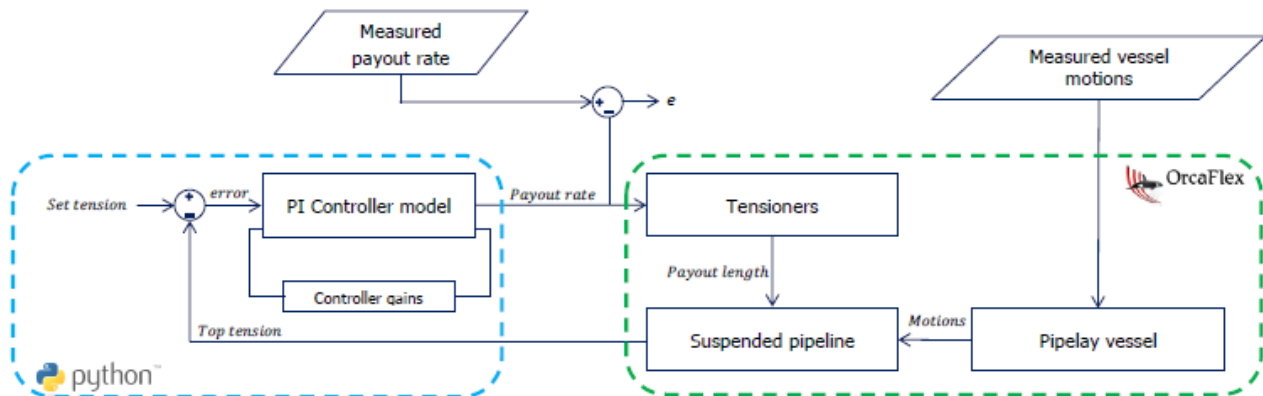


Figure 3.4.2: Python PI controller integration with Orcaflex [22]

All of the above control problems are solved separately, but as stated by Jensen [2], these could, with the current technologies, be automated, integrated and optimized. This is desirable since now the different departments, DP, survey and tensioner require operators with a lot of experience and are dependent on oral communication.

3.5 Concluding remarks

Starting with control. The lower-level control problems as position keeping of the vessel, path planning and tension control are all worked out but not integrated nor optimized and require experienced operators. It would, therefore, be very valuable to generate a real-time estimate of the systems (Vessel, tensioner and pipe) such that a higher-level controller can optimize the interaction of the different systems. A high-level control architecture can be set up to secure the actual structural integrity and, based on the actual state of the systems, maximise workability and minimise energy consumption by adapting based on algorithms, thus requiring less experienced operators.

Keeping the aforementioned conclusion on the control goal and the important points listed in the introduction in mind, the following conclusion can be drawn regarding modelling. Besides multiple simpler options, it can be concluded, given the complexity of the phenomena involved in pipelay, that it is best to continue on the model as presented by de Vries. The developed efficient corotational FEM formulation was the result of a project started by Allseas as well. The developed model runs in real-time and can incorporate geometric non-linearity and plasticity to give an accurate estimation of the structural loading.

Different in the scope of this thesis are the actual laying process and the control part. His contact method needs therefore to be enhanced to allow for pipe pulls or larger payouts in general. Additionally, the tensioner element must be elaborated on.

Chapter 4

Corotational FEM

As is concluded from the literature review, it has been decided to continue on the model set up by de Vries [5]. His geometric non-linear beam element is used and introduced briefly. Subsequently, the model for the vessel and stinger are set up. In this formulation, the payout of the tensioner is included, but how this is calculated will be elaborated on in the next Chapter. Next, the contact modelling is adapted to allow for pipe pulls. A Section is spent on the environmental loads that have been included. After that, a Section is assigned to the assembly of the different models, see [Figure 4.0.1](#), and the solving procedure using the Newton-Ralphson method for statics. Additionally, this is combined with the HHT- α numerical time integration (NTI) scheme to solve the dynamics, both are iterative schemes. Parts that are updated every iteration will be pointed out when introduced but the updating procedure is also indicated in an algorithm at the end of each Section.

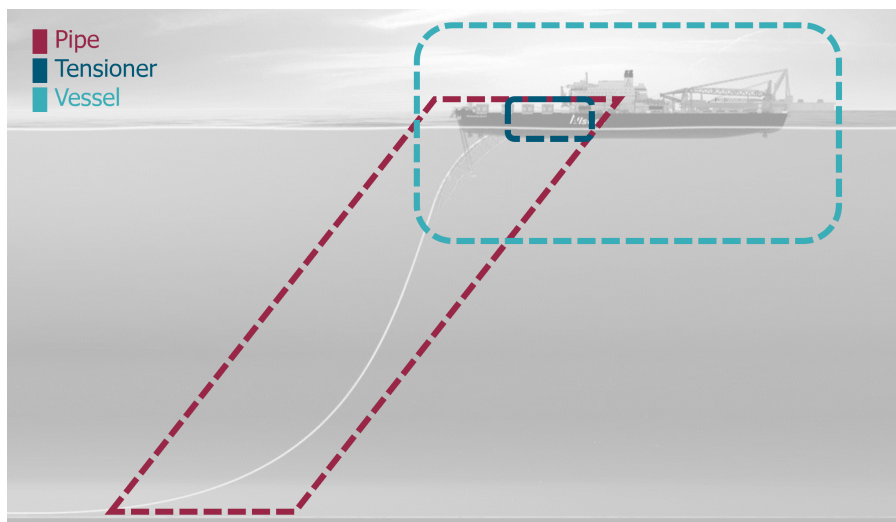


Figure 4.0.1: Different models and their interconnection

4.1 Geometric nonlinear beam element

De Vries took the corotational beam formulation of Crisfield [23] and made it more effective by replacing the rotation matrix such that the moments are now continuous at the nodes. This prevents incorrect results when torsional moments do play a role. This new rotation matrix also results in a simpler geometric stiffness matrix, which is less computationally expensive. The improvements are all documented in his PhD thesis [5] and are repeated without further comparison or extensive derivation.

First, the local and global degrees of freedom (DOFs) and their coherence are introduced. Subsequently, the transformation, rotation, and stiffness matrices are presented. Finally, the elemental update is explained using the introduced concepts.

Local degrees of freedom

A beam element consisting of two nodes commonly has 12 DOFs in total. Three translational and three rotational DOFs per node. In the corotational approach, a local coordinate frame is defined, which rotates with the element. By defining the origin on the first node, the translational DOFs of this node (3) are always zero. When defining the x-axis in this local frame in the direction of the second node the y- and z-displacements of the second node are also zero (2). This results in Crisfield's 7 local DOFs:

$$\vec{p}_l = [\psi_1 \quad \theta_1 \quad \phi_1 \quad u_L \quad \psi_2 \quad \theta_2 \quad \phi_2]^T \quad (4.1)$$

respectively representing the rotations around the x-, y- and z-axis of the node indicated with the subscript and u_L as the local displacement of the second node in the x-direction of local coordinate frame. These 7 local DOFs are calculated from the 12 global DOFs. Before those are introduced, based on initial position $\vec{x}_i$ and displacement $\vec{d}_i$, two auxiliary vectors are defined:

$$\begin{aligned} \vec{x}_{21} &= \vec{x}_2 - \vec{x}_1 \\ \vec{d}_{21} &= \vec{d}_2 - \vec{d}_1 \end{aligned} \quad (4.2)$$

The subscript refers to the corresponding node as can be seen from Figure 4.1.1. From these auxiliary vectors the initial and deformed lengths of the element can be calculated:

$$\begin{aligned} L_0 &= ||\vec{x}_{21}|| \\ L &= ||\vec{x}_{21} + \vec{d}_{21}|| \end{aligned} \quad (4.3)$$

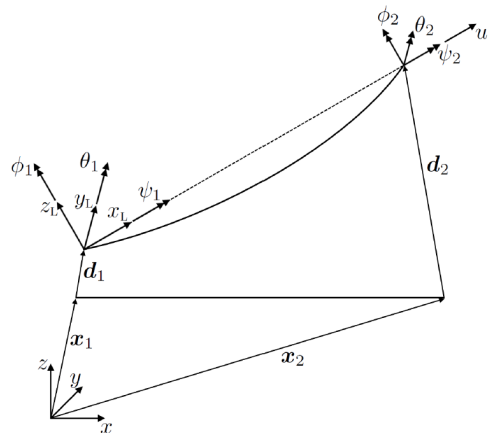


Figure 4.1.1: Local coordinate system with local DOFs [5]

Global degrees of freedom

In order to link the local DOFs to the global DOFs first nodal frames are introduced. The first node gets a frame $\mathbf{T}$ assigned, the second node a frame $\mathbf{U}$, and the element itself gets an element frame $\mathbf{E}$ assigned. The frames are defined by three orthogonal unit vectors:

$$\begin{aligned}\mathbf{T} &= [\vec{t}_1 \quad \vec{t}_2 \quad \vec{t}_3] \\ \mathbf{U} &= [\vec{u}_1 \quad \vec{u}_2 \quad \vec{u}_3] \\ \mathbf{E} &= [\vec{e}_1 \quad \vec{e}_2 \quad \vec{e}_3]\end{aligned}\tag{4.4}$$

The DOFs are commonly, as said, 3 translations and 3 rotations per node, adding up to 12 for an element. To allow for large rotations, this geometric non-linear beam element does not have the 3 angles but the incremental spin vector $\Delta\vec{\alpha}_i$ as rotational degrees of freedom:

$$\vec{p}_g = \begin{bmatrix} \vec{d}_1^T & \Delta\vec{\alpha}_1^T & \vec{d}_2^T & \Delta\vec{\alpha}_2^T \end{bmatrix}^T\tag{4.5}$$

This incremental spin vector represents the axis of rotation of the nodal frames $\mathbf{T}$ and $\mathbf{U}$ and its norm the angle of rotation. The incremental part implies that the nodal triads are updated every iteration n in the solution process with this increment. This is similar for $\mathbf{T}$ and $\mathbf{U}$ as:

$$\mathbf{T}_{n+1} = \Delta\mathbf{T}(\Delta\vec{\alpha}_i)\mathbf{T}_n\tag{4.6}$$

Here $\Delta\mathbf{T}(\Delta\vec{\alpha}_i)$ is the rotation matrix from the incremental spin vector obtained by Rodrigues' formula [5]. After the update of $\mathbf{T}$ and $\mathbf{U}$ the element frame $\mathbf{E}$ is updated by defining $\vec{e}_1$ from node 1 to node 2 as can be seen in Figure 4.1.2 using the two defined auxiliary vectors:

$$\vec{e}_1 = \frac{1}{L}(\vec{x}_{21} + \vec{d}_{21})\tag{4.7}$$

The rest of the element triad is defined to rotate the frame halfway from node 1 to node 2. This is achieved by defining first the rotation from $\mathbf{U}$ to $\mathbf{T}$ as $\mathbf{R}_\gamma$ and subsequently calculating the mean rotation matrix $\bar{\mathbf{R}}$ which involves Spurrier's algorithm as documented in [23] but omitted here. With the orthogonal components of $\bar{\mathbf{R}}$ as $\vec{r}_1, \vec{r}_2$ and $\vec{r}_3$ the remaining elemental frame components are defined as:

$$\begin{aligned}\vec{e}_2 &= \frac{\vec{r}_3 \times \vec{e}_1}{\|\vec{r}_3 \times \vec{e}_1\|} \\ \vec{e}_3 &= \vec{e}_1 \times \vec{e}_2\end{aligned}\tag{4.8}$$

With the nodal frames defined subsequently the 7 local DOFs can be calculated:

$$\begin{aligned}2 \sin(\psi_1) &= -\vec{t}_3^T \vec{e}_2 + \vec{e}_3^T \vec{t}_2 \\ 2 \sin(\theta_1) &= \vec{t}_3^T \vec{e}_1 - \vec{e}_3^T \vec{t}_1 \\ 2 \sin(\phi_1) &= -\vec{t}_2^T \vec{e}_1 + \vec{e}_2^T \vec{t}_1 \\ 2 \sin(\psi_2) &= -\vec{u}_3^T \vec{e}_2 + \vec{e}_3^T \vec{u}_2 \\ 2 \sin(\theta_2) &= \vec{u}_3^T \vec{e}_1 - \vec{e}_3^T \vec{u}_1 \\ 2 \sin(\phi_2) &= -\vec{u}_2^T \vec{e}_1 + \vec{e}_2^T \vec{u}_1 \\ u_l &= L - L_0\end{aligned}\tag{4.9}$$

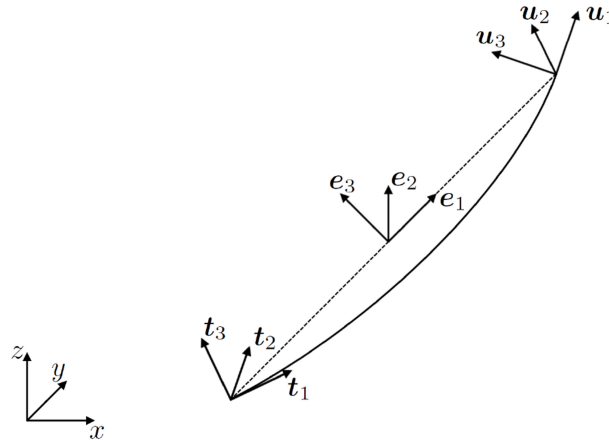


Figure 4.1.2: Element frame and nodal frames [5]

Transformation matrix

To calculate the other way around from global to local coordinates the transformation matrix $\mathbf{B}$ is used. This 7×12 matrix is derived by taking the variation of the local DOFs. This is a long derivation from Chrisfield [23], which is omitted here. The result is built up from the components that have already been introduced in three steps. The first step is given here:

$$\begin{aligned} \bar{\mathbf{e}}_2 &= \bar{\mathbf{r}}_3 \times \bar{\mathbf{e}}_1 \\ \mathbf{A} &= \frac{1}{L} (\mathbf{I} - \bar{\mathbf{e}}_1 \bar{\mathbf{e}}_1^T) \\ \mathbf{Q}_2 &= \frac{\mathbf{I} - \bar{\mathbf{e}}_2 \bar{\mathbf{e}}_2^T}{\|\bar{\mathbf{e}}_2\|} \begin{bmatrix} -\bar{\mathbf{r}}_3 \mathbf{A} & \frac{1}{2} \bar{\mathbf{e}}_1 \bar{\mathbf{r}}_3 & \bar{\mathbf{r}}_3 \mathbf{A} & \frac{1}{2} \bar{\mathbf{e}}_1 \bar{\mathbf{r}}_3 \end{bmatrix} \\ \mathbf{Q}_3 &= [\bar{\mathbf{e}}_2 \mathbf{A} \quad \bar{\mathbf{0}} \quad -\bar{\mathbf{e}}_2 \mathbf{A} \quad \bar{\mathbf{0}}] + \bar{\mathbf{e}}_1 \mathbf{Q}_2 \end{aligned} \quad (4.10)$$

Here $\mathbf{I}$ is a 3×3 identity matrix and $\bar{\mathbf{0}}$ a 1×3 zero vector. The tilde operator denotes the skew-symmetric cross product of the vector. These results are substituted in:

$$\begin{aligned} \bar{b}_1 &= [\bar{\mathbf{0}} \quad \bar{\mathbf{e}}_2^T \bar{\mathbf{t}}_3 - \bar{\mathbf{e}}_3^T \bar{\mathbf{t}}_2 \quad \bar{\mathbf{0}} \quad \bar{\mathbf{0}}] & + \bar{\mathbf{t}}_2^T \mathbf{Q}_3 - \bar{\mathbf{t}}_3^T \mathbf{Q}_2 \\ \bar{b}_2 &= [\bar{\mathbf{t}}_3^T \mathbf{A} \quad \bar{\mathbf{e}}_1^T \bar{\mathbf{t}}_3 - \bar{\mathbf{e}}_3^T \bar{\mathbf{t}}_1 \quad -\bar{\mathbf{t}}_3^T \mathbf{A} \quad \bar{\mathbf{0}}] & + \bar{\mathbf{t}}_1^T \mathbf{Q}_3 \\ \bar{b}_3 &= [\bar{\mathbf{t}}_2^T \mathbf{A} \quad \bar{\mathbf{e}}_1^T \bar{\mathbf{t}}_2 - \bar{\mathbf{e}}_2^T \bar{\mathbf{t}}_1 \quad -\bar{\mathbf{t}}_2^T \mathbf{A} \quad \bar{\mathbf{0}}] & + \bar{\mathbf{t}}_1^T \mathbf{Q}_2 \\ \bar{b}_4 &= [-\bar{\mathbf{e}}_1^T \quad \bar{\mathbf{0}} \quad \bar{\mathbf{e}}_1^T \quad \bar{\mathbf{0}}] & \\ \bar{b}_5 &= [\bar{\mathbf{0}} \quad \bar{\mathbf{0}} \quad \bar{\mathbf{0}} \quad \bar{\mathbf{0}}] & + \bar{\mathbf{u}}_2^T \mathbf{Q}_3 - \bar{\mathbf{u}}_3^T \mathbf{Q}_2 \\ \bar{b}_6 &= [\bar{\mathbf{u}}_3^T \mathbf{A} \quad \bar{\mathbf{0}} \quad -\bar{\mathbf{u}}_3^T \mathbf{A} \quad \bar{\mathbf{0}}] & + \bar{\mathbf{u}}_1^T \mathbf{Q}_3 \\ \bar{b}_7 &= [\bar{\mathbf{u}}_2^T \mathbf{A} \quad \bar{\mathbf{0}} \quad -\bar{\mathbf{u}}_2^T \mathbf{A} \quad \bar{\mathbf{0}}] & + \bar{\mathbf{u}}_1^T \mathbf{Q}_2 \end{aligned} \quad (4.11)$$

These are again substituted in:

$$\mathbf{B} = \begin{bmatrix} \bar{b}_1^T & \frac{-\bar{b}_2^T}{2 \cos \psi_1} & \frac{\bar{b}_3^T}{2 \cos \phi_1} & \bar{b}_4^T & \frac{\bar{b}_5^T}{2 \cos \psi_2} & \frac{-\bar{b}_6^T}{2 \cos \theta_2} & \frac{\bar{b}_7^T}{2 \cos \phi_2} \end{bmatrix}^T \quad (4.12)$$

Rotation- and stiffness matrix

The **transformation** matrix is a non-linear map from global to local coordinates. The map is evaluated for every iteration since it depends on the local DOFs, which converge to the solution more in each iteration. To rotate the local forces back to global forces, the transposed of this map can be used, but this would, in turn, result in a computationally expensive global tangential stiffness matrix. Therefore, the suggestion of de Vries is followed by using a different **rotation** matrix for the force mapping. This **R** consists of the already introduced nodal and elemental triads:

$$\mathbf{R} = \begin{bmatrix} \mathbf{E}^T & & & \\ & \mathbf{T}^T & & \\ & & \mathbf{E}^T & \\ & & & \mathbf{U}^T \end{bmatrix} \quad (4.13)$$

This way the the forces are rotated with the elemental frame and the moments with their nodal triads:

$$\vec{F}_g = \mathbf{R}^T \vec{F}_l \quad (4.14)$$

in which:

$$\begin{aligned} \vec{F}_l &= [N_1 \ F_{y1} \ F_{z1} \ T_1 \ M_{y1} \ M_{z1} \ N_2 \ F_{y2} \ F_{z2} \ T_2 \ M_{y2} \ M_{z2}]^T \\ \vec{F}_g &= [F_{x1}^g \ F_{y1}^g \ F_{z1}^g \ M_{x1}^g \ M_{y1}^g \ M_{z1}^g \ F_{x2}^g \ F_{y2}^g \ F_{z2}^g \ M_{x2}^g \ M_{y2}^g \ M_{z2}^g]^T \end{aligned} \quad (4.15)$$

Since this rotation matrix is 12×12 and there are 7 local DOFs a 12×7 local material stiffness matrix is derived. Without derivation, this is:

$$\mathbf{K}_{mat} = \frac{1}{L_0} \begin{bmatrix} 0 & 0 & 0 & -EA & 0 & 0 & 0 \\ 0 & 0 & 6\frac{EI}{L_0} & 0 & 0 & 0 & 6\frac{EI}{L_0} \\ 0 & -6\frac{EI}{L_0} & 0 & 0 & 0 & -6\frac{EI}{L_0} & 0 \\ GJ & 0 & 0 & 0 & -GJ & 0 & 0 \\ 0 & 4EI & 0 & 0 & 0 & 2EI & 0 \\ 0 & 0 & 4EI & 0 & 0 & 0 & 2EI \\ 0 & 0 & 0 & EA & 0 & 0 & 0 \\ 0 & 0 & -6\frac{EI}{L_0} & 0 & 0 & 0 & -6\frac{EI}{L_0} \\ 0 & 6\frac{EI}{L_0} & 0 & 0 & 0 & 6\frac{EI}{L_0} & 0 \\ -GJ & 0 & 0 & 0 & GJ & 0 & 0 \\ 0 & 2EI & 0 & 0 & 0 & 4EI & 0 \\ 0 & 0 & 2EI & 0 & 0 & 0 & 4EI \end{bmatrix} \quad (4.16)$$

in which E is the elasticity modules, I is the moment of inertia, G is the shear modulus and A is the surface area of the pipe. Since these are all constants within the simulation, they have to be computed only once. If an equal mesh size over the whole pipe is used (and thus L_0 is equal in all elements), it is also equal for all individual elements. For clarity, this is the local material stiffness matrix, which maps the 7 local DOFs to the 12 local element forces. This is different from the tangential and effective stiffness matrix which will be introduced in a later subsection. Combining the above equations leads to the global material stiffness matrix $\mathbf{K}_{mat}^g$:

$$\vec{F}_g = \mathbf{R}^T \mathbf{K}_{mat} \mathbf{B} \vec{p}_g = \mathbf{K}_{mat}^g \vec{p}_g \quad (4.17)$$

This can be interpreted as the mapping from global DOFs to local DOFs, to local element forces, to global forces through the three matrices read from right to left.

Tangential stiffness matrix

The solving procedure will be explained in Section 4.5. The Newton-Raphson method is used. In short, the first-order derivative of the stiffness matrix is calculated with which an update to the solution of the previous iteration is calculated. This first-order derivative is called the tangential stiffness matrix. It is derived by taking the variation of Equation 4.14 towards the global DOFs. Due to the product rule, this results in two parts. The variation of forces leads to the material stiffness matrix as in Equation 4.16 and the variation of the rotation matrix to the geometric stiffness matrix. The latter can be interpreted as the change in the internal force vector due to a change in orientation. An analogy here is a beam that is very stiff axially but less stiff in the bending direction. When the beam's orientation changes, a different stiffness is felt globally. Omitting the derivation leads to the following geometric stiffness matrix:

$$\mathbf{K}_{geo} = \begin{bmatrix} -N_1 \mathbf{A} & \mathbf{0} & N_1 \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_1 & \mathbf{0} & \mathbf{0} \\ -N_2 \mathbf{A} & \mathbf{0} & N_2 \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{G}_2 \end{bmatrix} + \begin{bmatrix} F_{y1} \mathbf{Q}_2 + F_{z1} \mathbf{Q}_3 \\ \mathbf{0}^{3 \times 12} \\ F_{y2} \mathbf{Q}_2 + F_{z2} \mathbf{Q}_3 \\ \mathbf{0}^{3 \times 12} \end{bmatrix} \quad (4.18)$$

With $\mathbf{0}$ being the 3x3 zero matrix and:

$$\begin{aligned} \mathbf{G}_1 &= -T_1 \tilde{\mathbf{t}}_1 - M_{y1} \tilde{\mathbf{t}}_2 - M_{z1} \tilde{\mathbf{t}}_3 \\ \mathbf{G}_2 &= -T_2 \tilde{\mathbf{u}}_1 - M_{y2} \tilde{\mathbf{u}}_2 - M_{z2} \tilde{\mathbf{u}}_3 \end{aligned} \quad (4.19)$$

Together they form the tangential stiffness matrix:

$$\mathbf{K}_{tan} = \mathbf{R}^T \mathbf{K}_{mat} \mathbf{B} + \mathbf{K}_{geo} \quad (4.20)$$

Element updates

Now that the element is introduced the update procedure with which a new iteration is calculated can be summarized in pseudo-code as shown in Algorithm 1. This algorithm will be referred back to later when the complete simulation setup is explained.

Algorithm 1 Element update

- 1: **for** all elements **do**
 - 2: Update nodal translational coordinates $\vec{d}_i$
 - 3: Calculate rotation matrix $\Delta \mathbf{T}(\Delta \alpha)$
 - 4: Update nodal triads $\mathbf{T}_{n+1}$ and $\mathbf{U}_{n+1}$ using Equation 4.6
 - 5: Calculate mean rotation matrix $\bar{\mathbf{R}}$
 - 6: Calculate element frame $\mathbf{E} = [\vec{e}_1 \vec{e}_2 \vec{e}_3]$
 - 7: Calculate the local DOFs using Equation 4.9
 - 8: Calculate local internal forces using Equation 4.16
 - 9: Calculate the transformation matrix using Equation 4.12
 - 10: Calculate the rotation matrix using Equation 4.13
 - 11: Calculate the geometric stiffness matrix using Equation 4.18
 - 12: Calculate the tangential stiffness matrix using Equation 4.20
 - 13: Add contribution of element to global internal force vector using Equation 4.14
 - 14: **end for**
-

4.2 Vessel with stinger element

The vessel and stinger is modelled as a single rigid body due to its high stiffness compared with the pipe. Therefore, this vessel element has 6 DOFs:

$$\vec{p}_v = [x_v \ y_v \ z_v \ \phi_v \ \theta_v \ \psi_v]^T \quad (4.21)$$

The first three are the position of the centre of mass (COM) in the global or world frame. The second three is the attitude, defined as the roll, pitch, and yaw angle. These three angles form with Rodrigues' formula [5], the rotation matrix ${}^v\mathbf{R}_W$ from the world frame Ψ_W to the local frame on the COM of the vessel Ψ_v , see Figure 4.2.1. In this local frame, the roller boxes on the stinger have a stationary location (coordinates $\vec{R}_{i,l}$) and move rigidly with the vessel. Therefore when the 6 DOF of this vessel element are known, either calculated by an RAO such as in OrcaFlex based on a sea-state or measured by GPS/IMU onboard, the coordinates of all roller boxes and tensioners are known in the global frame by:

$$\vec{R}_{i,g} = \begin{bmatrix} x_v \\ y_v \\ z_v \end{bmatrix} + {}^W\mathbf{R}_v \vec{R}_{i,l} \quad (4.22)$$

in which ${}^W\mathbf{R}_v$ is the transposed of ${}^v\mathbf{R}_W$ due to the properties of a rotation matrix. In a similar way the beginning of the pipe $\vec{P}_A$ can be calculated since this is the stationary position of the tensioner within the moving vessel minus the payout λ . This payout is calculated based on the controller defined in the next Chapter. Once this payout is known, the first pipe coordinate in the global frame can be calculated as:

$$\vec{P}_{Ag} = \begin{bmatrix} x_v \\ y_v \\ z_v \end{bmatrix} + {}^W\mathbf{R}_v \begin{bmatrix} x_{T,l} - \lambda \\ y_{T,l} \\ z_{T,l} \end{bmatrix} \quad (4.23)$$

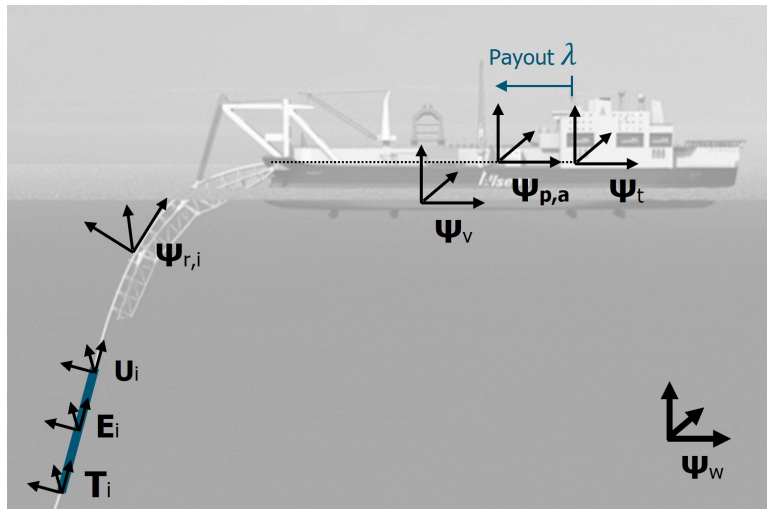


Figure 4.2.1: Different coordinate systems in the vessel with stinger element

Note here that a positive payout is in this case towards the back of the ship, therefore the minus sign since Ψ_v is orientated with the positive x-axis towards the front of the ship and pipe is layed by sailing forward. This is the case for Audacia and Lorelay. The Pioneering spirit, however, lays pipe by sailing backwards and paying pipe out from the front side. The payout only affects the local x-axis since the firing line is horizontal in the x,z-plane.

4.3 Contact modelling

For the modelling of contact, an enhancement has been made to the method of De Vries. His method requires a pipe node at each roller box in the stinger. This way, a pipe pull cannot be simulated since that would imply the movement of nodes over the stinger. Additionally, a uniform mesh is not possible. Therefore, in this section, de Vries's penalty method formulation is adapted to overcome those restrictions. The contact with the ground is modelled identically as de Vries describes it. Therefore, it is omitted here since it is a simplified version of the adapted contact method shown here [5].

Adapted contact modelling Roller-boxes

For all roller boxes, their coordinates $\vec{R}_{i,g}$ in the world frame can be calculated using Equation 4.22, and the coordinates $\vec{d}_i$ of the pipe nodes are also available. To find the nearest element to rollerbox i , first the nearest node j is found as:

$$\operatorname{argmin}_j \left(\|\vec{R}_{i,g} - \vec{d}_j\| \right) \quad (4.24)$$

Subsequently, the two adjacent nodes are compared to find the nearest element. This element is approximated with a straight line between the two nodes as indicated in Figure 4.3.1. A coordinate frame $\mathbf{S}$ is defined at the point of potential contact, defined in the world frame as $\vec{\kappa}$. The first unit vector $\vec{s}_1$ is defined as the vector from the first node of the contact element to the second node, which was already defined as $\vec{e}_1$ in Equation 4.7. $\vec{s}_2$ is defined orthogonal on that by taking the cross product with $\vec{v}_3$, which is the vector in the vessel frame pointing upwards and therefore the third column of Ψ_v . This ensures that $\vec{s}_2$ is always pointing towards the side of the vessel. Finally, $\vec{s}_3$ is defined by orthogonality. This results in the following rotation matrix:

$$\mathbf{S} = [\vec{s}_1 \quad \vec{s}_2 \quad \vec{s}_3] \quad (4.25)$$

with:

$$\begin{aligned} \vec{s}_1 &= \vec{e}_1 \\ \vec{s}_2 &= \vec{s}_1 \times \vec{v}_3 \\ \vec{s}_3 &= \vec{s}_1 \times \vec{s}_2 \end{aligned}$$

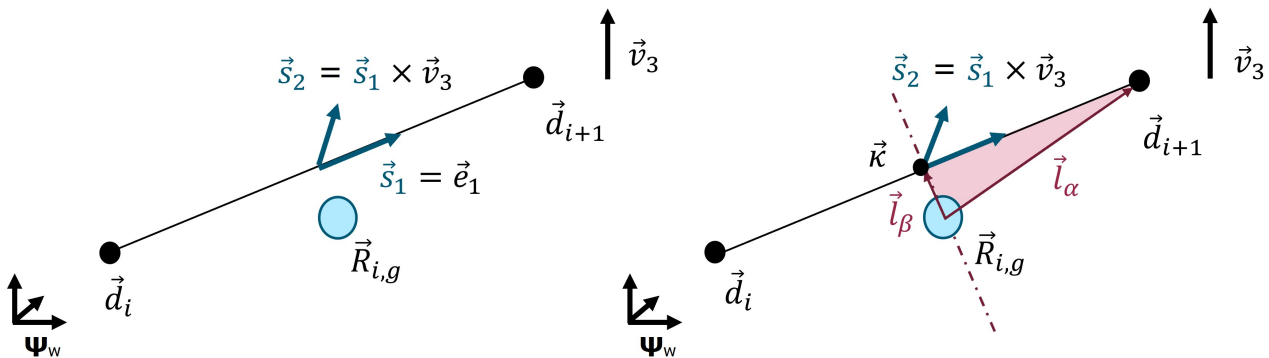


Figure 4.3.1: Schematics for calculation of potential contact point $\vec{\kappa}$

Subsequently, point $\vec{\kappa}$ is found by projecting the vector from the rollerbox to the second node of the contact element onto the pipe's ($\vec{s}_1$) and subtracting the result from this vector such that:

$$\vec{l}_\alpha = \vec{d}_{i+1} - \vec{R}_{i,g} \quad (4.26)$$

$$\vec{l}_\beta = \vec{l}_\alpha - (\vec{l}_\alpha \cdot \vec{s}_1) \vec{s}_1 \quad (4.27)$$

$$\vec{\kappa} = \vec{R}_{i,g} + \vec{l}_\beta \quad (4.28)$$

With point $\vec{\kappa}$ defined contact can be checked by calculating the penetration. For a V-shaped rollerbox the two normal vectors of these rollerboxes are defined in the plane spanned by $\vec{s}_2$ and $\vec{s}_3$ as shown in Figure 4.3.2. The angle of the rollers is fixed and taken here as α but can differ per roller box. This leads to the following definition for the normal vectors:

$$\vec{n}_1 = \vec{s}_2 \sin(\alpha) + \vec{s}_3 \cos(\alpha)$$

$$\vec{n}_2 = -\vec{s}_2 \sin(\alpha) + \vec{s}_3 \cos(\alpha)$$

When the vessel rotates in the yaw direction for example it can occur that the pipe only has contact with one side of the roller box. Therefore both sides are checked separately for penetration. This can be done by projecting the $\vec{l}_\beta$ onto the normal vectors and when its dot product is positive there exists a gap and when negative penetration occurs. The rollers have a non-zero diameter D_r and the pipe itself has a penetration diameter D_p which should be taken into account. The latter is different from the stress diameter in the sense that a coating might increase the diameter for contact but only partially contributes to the strength. This results in the following definition for penetration:

$$\Gamma_1 = -\vec{n}_1 \cdot \vec{l}_\beta + \frac{1}{2}(D_r + D_p)$$

$$\Gamma_2 = -\vec{n}_2 \cdot \vec{l}_\beta + \frac{1}{2}(D_r + D_p)$$

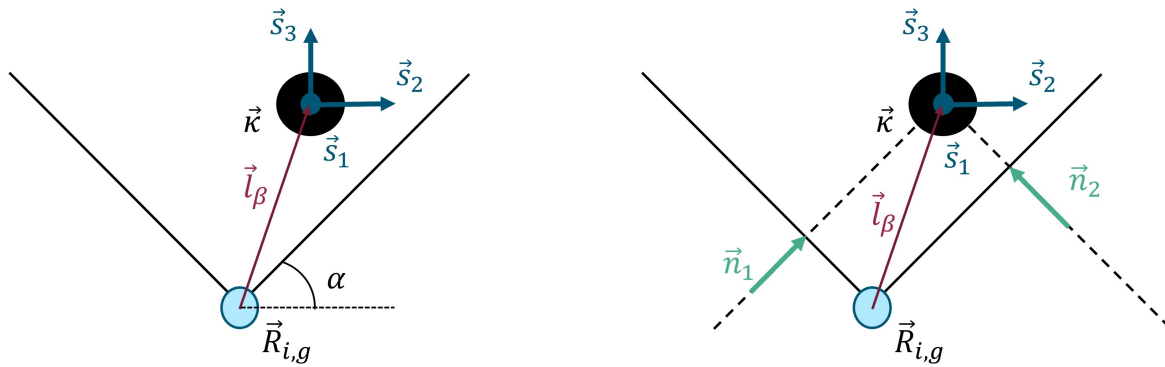


Figure 4.3.2: Schematic for calculation penetration Γ_i

Subsequently the contact force can be calculated for the sides that have a positive penetration separately. Additionally also a friction force could be calculated in the tangential direction but since the rollers can roll and the contact can become wet this is very difficult to model correctly. Additionally a wet rolling surface would not yield much friction force thus this is omitted. The friction force is determined by a penalty factor P times the penetration in the direction of the normal force:

$$\vec{F}_c = P\Gamma_i \vec{n}_i = F_c \vec{n}_i \quad (4.29)$$

The penalty factor should represent the contact stiffness between the pipe and roller. This force, however, should be applied at the attachment point $\vec{\kappa}$. Due to the limitations of FEM, loads can only be applied at the nodes. Therefore, an equivalent load is calculated to add on the first and second node of the element, which should represent the calculated contact force as indicated in Figure 4.3.3. An example of a statically and kinematically equivalent load vector of a distributed load is given in [24]. This approach corresponds with the approach of de Vries as will be elaborated on in section 4.4. For a point load a kinematically equivalent load vector cannot be derived and therefore only a statically equivalent approximation can be made. This is done for a point load in the centre of the beam in [24], which is here extended for an arbitrary location on the beam based on Warren [25]. For simplicity first the equivalent loading is calculated in the plane spanned by $\vec{n}_i$ and $\vec{s}_1$ as shown in Figure 4.3.3. Using this convention the equivalent force and moment in the first node A and second node B becomes:

$$\begin{aligned} F_A &= \frac{F_c}{L^3}(L - l_\gamma)^2(L + 2l_\gamma) \\ M_A &= \frac{-F_c l_\gamma}{L^2}(L - l_\gamma)^2 \\ F_B &= \frac{F_c l_\gamma^2}{L^3}(3L - 2l_\gamma) \\ M_B &= \frac{-F_c l_\gamma^2}{L^2}(L - l_\gamma) \end{aligned}$$

with:

$$l_\gamma = \|\vec{\kappa} - \vec{d}_i\| \quad (4.30)$$

Next, these loads should be transformed into a global frame such that they can be added to the global external load vector. The forces can be projected using the normal vector, The moments act out of the plane spanned by $\vec{n}_i$ and $\vec{s}_1$ thus can be projected on the vector resulting from their cross product as:

$$\vec{F}_{ext} = \begin{bmatrix} F_A \vec{n}_i \\ M_A (\vec{n}_i \times \vec{s}_1) \\ F_B \vec{n}_i \\ M_B (\vec{n}_i \times \vec{s}_1) \end{bmatrix} \quad (4.31)$$

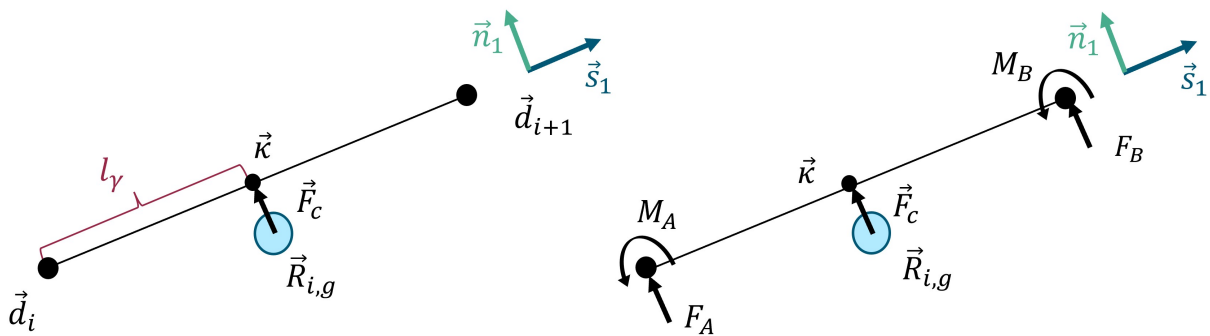


Figure 4.3.3: Equivalent load vector

Contact algorithm

In the previous Subsection, the procedure is explained to calculate the external force vector of a single roller box. This procedure has to be repeated for every roller box and additionally, a contact state procedure is applied for a more numerically stable algorithm and thus faster convergence. For details refer to de Vries. The algorithm is summarized in Algorithm 2.

Algorithm 2 Rollerbox forces algorithm

```

1: for all rolleboxes do
2:   Find 2 nearest nodes (and thus nearest element where contact may occur)
3:   Calculate attachment point  $\vec{\kappa}$  of the element based on a straight line approximation
4:   Calculate penetration  $\Gamma_i$ 
5:   if Penetration > 0 or previouscontact = true then
6:     Calculate contact force
7:     Calculate generalized force vector based on contact force and attachment point.
8:     Add generalized force vector to external force vector.
9:   end if
10:  if Penetration > 0 then
11:    previouscontact = true
12:  else
13:    previouscontact = false
14:  end if
15: end for

```

4.4 Environmental loads

Two types of environmental loads are considered that affect the pipe, hydrostatic and hydrodynamic loads. Since the displacements of the vessel are prescribed with measurements, the underlying loads which result in those movements do not have to be modelled and only loads on the pipe are considered.

Hydrostatic loads

Pipe elements in the ship or on the part of the stinger above sea level are only affected by a gravitational load of q_g per meter pipe. Pipe elements that are fully submerged additionally have a buoyancy force per meter q_b due to Archimedes' law. The net submerged weight w per meter can be calculated for a pipe of a homogeneous material as:

$$\begin{aligned} q_g &= (A_{ext} - A_{in})\rho_p g \\ q_b &= (\rho_w A_{ext} - \rho_{in} A_{in})g \\ w &= q_g - q_b \end{aligned}$$

in which:

- A_{ext} = External area of the pipe [m^2]
- A_{in} = Internal area of the pipe [m^2]
- ρ_p = Density of the pipe [kg^3]
- ρ_w = Density of the seawater [kg^3]
- g = gravitational acceleration [m^2]

One important remark must be made when using this. Applying Archimedes on an element assumes that the pressures acts on all surfaces of the element, including the cross-sectional area. Therefore the axial forces in submerged elements are overestimated, as indicated in [Figure 4.4.1](#). The calculated axial forces is called the effective axial force and the actual occurring normal forces are called the true [\[5\]](#), or wall tension [\[6\]](#). This can be calculated for a linear elastic material model as:

$$N_{true} = N_{eff} - p_{ext}A_{ext} + p_{in}A_{in} \quad (4.32)$$

in which the internal pressure p_{in} equals the air pressure inside the pipe during installation, and the external pressure p_{ext} can be calculated as the hydrostatic pressure:

$$p_{ext} = \rho_w g d_{i,z} \quad (4.33)$$

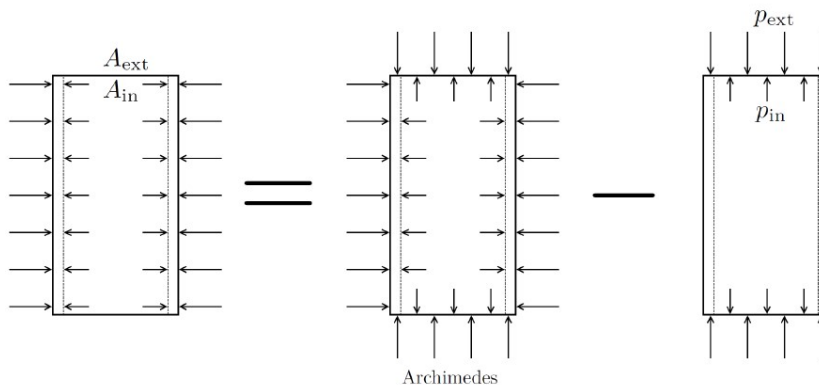


Figure 4.4.1: Submerged weight modelled with Archimedes [\[5\]](#)

For pipes with a relatively light coating, for example a concrete coating compared with a steel pipe the buoyancy is a significant portion of the gravity load. Therefore, considering an element as in-air when the second node is just above the sea level is too big of an approximation with relatively large elements as simulations showed. To model this more accurately a semi-submerged element has a proportional weight q_p as visualized in Figure 4.4.2 and calculated as:

$$q_p = \frac{l_\delta}{l_\delta + l_\epsilon} q_g + \frac{l_\epsilon}{l_\delta + l_\epsilon} w \quad (4.34)$$

in which the partitions l_δ and l_ϵ can be calculated from the z-component of the nodes $\vec{d}_i$.

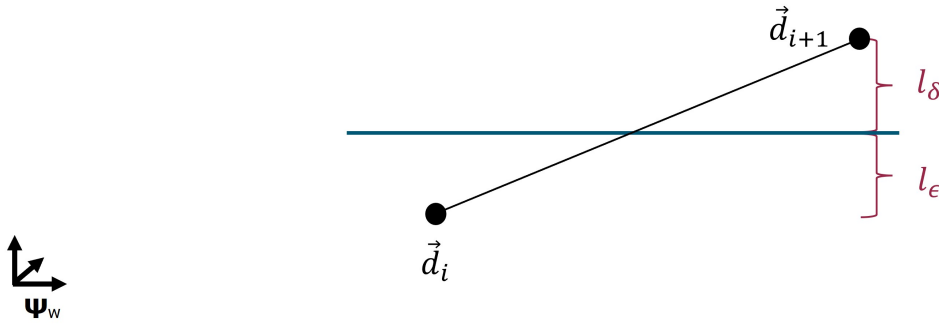


Figure 4.4.2: Schematic of semi-submerged element

Hydrodynamic loads

Literature and Orcaflex agree upon using Morison's equation for the modelling of the hydrodynamic loads on a pipe [26, 20, 6]. This equation consists of three parts: drag, added mass, and the pressure gradient. These parts depend on velocity and acceleration in the normal direction of the pipe. The first two terms are based on the relative motion between the pipe and the water. Assuming still water ($\vec{d}_w = \vec{d}_w = 0$) or known water conditions the relative velocity and acceleration can be calculated as:

$$\begin{aligned} \vec{d}_r &= \vec{d}_w - \vec{d} \\ \vec{\ddot{d}}_r &= \vec{\ddot{d}}_w - \vec{\ddot{d}} \end{aligned}$$

Subtracting the tangential parts results in the normal relative velocity and acceleration:

$$\begin{aligned} \vec{d}_{rn} &= (\mathbf{I} - \vec{e}_1 \vec{e}_1^T) \vec{d}_r \\ \vec{\ddot{d}}_{rn} &= (\mathbf{I} - \vec{e}_1 \vec{e}_1^T) \vec{\ddot{d}}_r \end{aligned}$$

from which the hydrodynamic loads per meter $\vec{q}_h$ pipe can be calculated using:

$$\vec{q}_h = \frac{1}{2} \rho_w C_d D \vec{d}_{rn} \|\vec{d}_{rn}\| + \rho_w C_a \frac{\pi}{4} D^2 \vec{\ddot{d}}_{rn} + \rho_w \frac{\pi}{4} D^2 \vec{\ddot{d}}_{wn} \quad (4.35)$$

in which C_d and C_a are the respective drag and added mass coefficients. During operation ideally an acoustic doppler (wave sensor) should be employed to measure the motions of the sea. If that is not available, a worst-case estimate should be used.

Distributed load

Both hydrostatic as hydrodynamic loads are set up as a distributed load per meter. As explained in the previous section, a distributed load should be added to the external force vector through an equivalent load vector. This load vector for a distributed load is statically and kinematically equivalent for a general distributed load q as:

$$\vec{F}_{ext} = \begin{bmatrix} \frac{1}{2} L_0 \vec{q} \\ \frac{1}{12} (\mathbf{I} - \vec{e}_1 \vec{e}_1^T) \vec{q} L_0^2 \\ \frac{1}{2} L_0 \vec{q} \\ -\frac{1}{12} (\mathbf{I} - \vec{e}_1 \vec{e}_1^T) \vec{q} L_0^2 \end{bmatrix} \quad (4.36)$$

4.5 Solving statics and dynamics

The Newton-Raphson method is used to solve statics. In dynamic simulations, this is combined with the Hilbert-Hughes-Taylor- α NTI scheme. After both methods are introduced, an initialization procedure for statics and the complete solving algorithm for dynamics is described.

Newton-Raphson

For linear static simulations, the internal forces equal the external forces in equilibrium. Such systems can be solved directly by subsequently reducing the system, inverting the stiffness matrix, and back substitution of the found displacements. But in the case of this corotational formulation, the internal forces as well as the external forces are dependent on the displacements in a non-linear manner:

$$\vec{F}^{int}(\Delta\vec{d}) = \vec{F}^{ext}(\Delta\vec{d}) \quad (4.37)$$

Therefore the iterative Newton-Raphson algorithm is used which calculates the residual error:

$$\vec{R}_s = \vec{F}^{int}(\Delta\vec{d}) - \vec{F}^{ext}(\Delta\vec{d}) \quad (4.38)$$

and finds the incremental displacement with which the initial guess must be updated to set this residual error to zero. This update is based on the static effective stiffness matrix:

$$\hat{\mathbf{K}}_s = \mathbf{K}^{int} - \mathbf{K}^{ext} \quad (4.39)$$

In which $\mathbf{K}^{int}$ is the already introduced tangential stiffness matrix. Here also the external loads depend on the displacements and therefore that effect ($\mathbf{K}^{ext}$) must be taken into consideration and the result is therefore called the effective stiffness matrix $\hat{\mathbf{K}}$. For all detailed contributions is referred to [5]. The incremental update is found by evaluating:

$$\delta\vec{d} = -\hat{\mathbf{K}}_s^{-1}\vec{R}_s \quad (4.40)$$

Subsequently, the displacements are updated from which elements, loads and the effective stiffness matrix can be updated. This process is repeated until the relative error norm is below a set threshold:

$$\epsilon = \frac{\|\vec{R}_s\|}{\|\vec{F}^{ext}\|} \quad (4.41)$$

Numerical time integration

A similar approach is used for dynamic simulations. For the numerical time integration, the Hilber-Hughes-Taylor- α method is used. This is a generalization of the common Newmark method [27]. This is an implicit, second-order accurate, and unconditionally stable numerical time integration scheme. It has one controllable parameter, α , which sets the numerical damping and with it the Newmark parameters β and γ :

$$0 \leq \alpha \leq \frac{1}{3} \quad \beta = \frac{(1 + \alpha)^2}{4} \quad \gamma = \frac{1}{2} + \alpha \quad (4.42)$$

$$(4.43)$$

Additionally, we define the following constants:

$$a_0 = \frac{1}{\beta \Delta t^2} \quad a_1 = \frac{\gamma}{\beta \Delta t} \quad a_2 = \frac{1}{\beta \Delta t} \quad (4.44)$$

$$a_3 = \frac{1}{2\beta} - 1 \quad a_4 = \frac{\gamma}{\beta} - 1 \quad a_5 = \frac{\Delta t}{2} \left(\frac{\gamma}{\beta} - 2 \right) \quad (4.45)$$

These constants define the velocity and accelerations of the next time step when the values for the current timestep and the displacement increment of the next timestep, $\Delta \vec{d}$, are known:

$$\vec{d}_{t+\Delta t} = a_0 \Delta \vec{d} - a_2 \vec{d}_t - a_3 \vec{d}_t \quad (4.46)$$

$$\vec{d}_{t+\Delta t} = a_1 \Delta \vec{d} - a_4 \vec{d}_t - a_4 \vec{d}_t \quad (4.47)$$

But the reverse is also true. This way, considering the following equation of motion:

$$\mathbf{M}^{sys} \vec{d}_{t+\Delta t} + \vec{F}_\alpha^{int} = \vec{F}_\alpha^{ext} \quad (4.48)$$

with $\mathbf{M}^{sys}$ the system mass matrix and:

$$\vec{F}_\alpha^{int} = \vec{F}_t^{int} + (1 - \alpha) \Delta \vec{F}_\alpha^{int}$$

$$\vec{F}_\alpha^{ext} = \vec{F}_t^{ext} + (1 - \alpha) \Delta \vec{F}_\alpha^{ext}$$

Equation 4.47 can be substituted in the EOM, which yields only the displacements increment $\Delta \vec{d}$ as unknown variables to solve for. This is again done with the Newton-Raphson incremental solving procedure. Only the inertia contributions to the residual and the effective stiffness matrix have to be taken into account, which yields now without the statics "s" index:

$$\vec{R} = \vec{F}_\alpha^{int} - \vec{F}_\alpha^{ext} + \mathbf{M}^{sys} (a_0 \Delta \vec{d} - a_2 \vec{d} - a_3 \vec{d}) \quad (4.49)$$

$$\hat{\mathbf{K}} = a_0 \mathbf{M}^{sys} + (1 - \alpha) (\mathbf{K}^{int} - \mathbf{K}^{ext}) \quad (4.50)$$

Once the algorithm has converged and the relative error is below the threshold the solved displacement increment can be back-substituted in Equation 4.47 to obtain the velocities and accelerations.

Mass matrix

The lumped mass matrix from de Vries is used here:

$$\mathbf{M}_L = m \begin{bmatrix} \frac{1}{2} & & & & & & & & & & \\ & \frac{1}{2} & & & & & & & & & \\ & & \frac{1}{2} & & & & & & & & \\ & & & 0 & & & & & & & \\ & & & & 0 & & & & & & \\ & & & & & 0 & & & & & \\ & & & & & & \frac{1}{2} & & & & \\ & & & & & & & \frac{1}{2} & & & \\ & & & & & & & & \frac{1}{2} & & \\ & & & & & & & & & 0 & \\ & & & & & & & & & & 0 \\ & & & & & & & & & & & 0 \end{bmatrix} \quad (4.51)$$

Due to the equal diagonal terms, the mass matrix is equal in its local and global frame since a transformation as:

$$\mathbf{M}_L = \mathbf{R}^T \mathbf{M}_L \mathbf{R} \quad (4.52)$$

yields itself.

Simulating statics

To simulate a static simulation, an initialization algorithm is used to guide the algorithm to its equilibrium. It was found that a few incremental loading steps are required to get to the static equilibrium of S-lay. The simulation starts with a straight unstressed pipe, which is fixed at the vessel side. In the first increment, the pipe is loaded with an axial tension. Next, the gravity/buoyancy is added in a few incremental steps. The axial force ensures that the pipe does not completely fall and prohibits divergence. Next, the contact stiffness of the ground and the stinger is increased from 0 to the correct values "catching" the pipe. The last step is to decrease the axial bottom tension till the desired top tension is achieved. See [Figure 4.5.1](#) for the incremental steps. From this point, the fixures switch from the vessel side of the pipe to the seafloor side of the pipe. This allows the vessel to be moved in the dynamic simulation, which can be performed next from this static equilibrium.

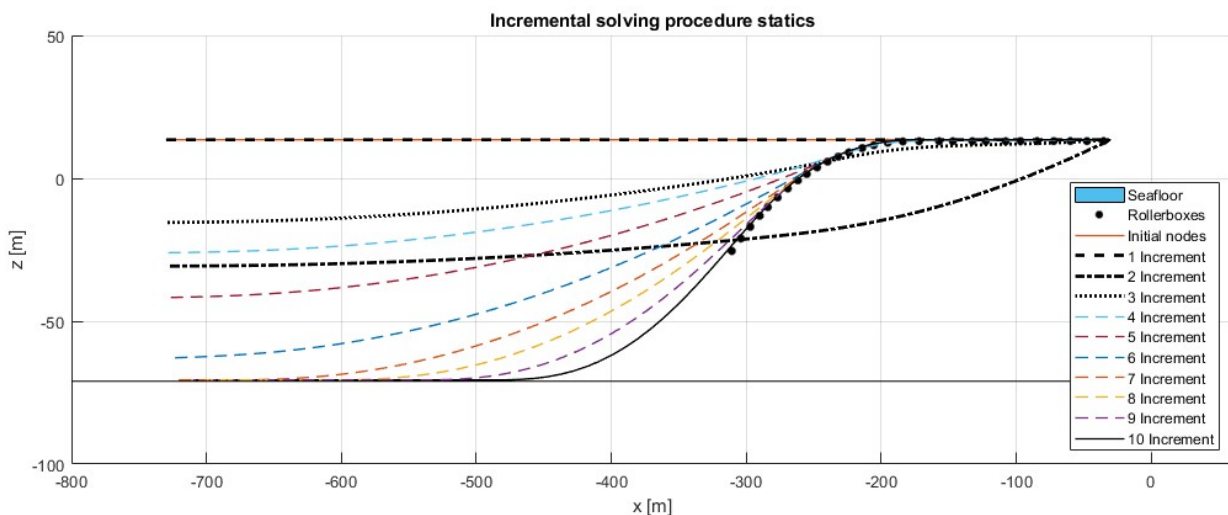


Figure 4.5.1: Incremental solving procedure

Simulating dynamics

Since statics have a simplified procedure of the dynamics simulation algorithm, the algorithm is omitted, and only the dynamic algorithm is included. The simulation procedure is as follows. First, statics are run, and the fixtures are switched such that the pipe lays in equilibrium and the vessel's position can be set every timestep based on measurements. In the Matlab code, a lot of matrices and figures are initialised, but omitting these initialisation steps, the simulation can be summarized in Algorithm 3.

Algorithm 3 Dynamic simulation

```

1: Run Statics
2: Initialise tensioner
3: for time = 1:end do
4:   Set Vessel position from measurements
5:   Calculate tensioner payout using controller introduced in Chapter 5
6:   for all elements do
7:     Calculate hydrostatic loads and add to  $\vec{F}^{ext}$ 
8:   end for
9:   while  $\epsilon > \text{Threshold}$  AND iteration < max iterations do
10:    for all elements do
11:      Assemble  $\mathbf{K}^{sys}$ ,  $\mathbf{K}^{int}$ , and  $\mathbf{M}^{sys}$ .
12:      Add seafloor contact contribution to  $\mathbf{K}^{ext}$ 
13:    end for
14:    for all roller-boxes do
15:      Add roller-box contact contribution to  $\mathbf{K}^{ext}$ 
16:    end for
17:    Calculate  $\hat{\mathbf{K}}^{sys}$ 
18:    Reduce system matrices and force vectors
19:    Calculate Residual
20:    Calculate incremental update
21:    Update elements using Algorithm 1
22:    Add hydrodynamic force to  $\vec{F}^{ext}$ 
23:    Add roller-box contact force to  $\vec{F}^{ext}$  if contact using Algorithm 2
24:    Add seafloor contact force to  $\vec{F}^{ext}$  if contact
25:    Add boundary forces to  $\vec{F}^{ext}$  based on  $\vec{F}^{int}$  at the first and last node
26:    Calculate  $\epsilon$ 
27:    iteration = iteration + 1
28:  end while
29:  update velocities and accelerations
30: end for

```

Chapter 5

Control architecture

As elucidated in Chapter [2](#), currently in shallow water, a tension feedback controller realises a pipe pull due to the vessel moving 24 meters. The pipe ahead functionality feeds forward a 24-meter payout to realise the pipe pulls in deeper water. In transitional waters, a mix is used in which an experienced operator sets the pipe ahead settings. Although small changes/improvements can be made in this Chapter, the focus is on a larger and potentially more impactful improved control strategy. This is done by bringing up an alternative control strategy that can maximize workability and minimise energy usage while constrained by structural integrity. First, the latter control goals are made specific, after which a strategy is described for achieving them.

5.1 Control goal

The control goal is maximizing workability and minimizing energy consumption while being constrained by the structural integrity of the pipe.

Structural integrity

As explained in Section 2.1 the structural integrity of the pipe is defined through the unity checks ζ as defined in DNVGL-ST-F101 for the overbend ζ_O as well as the sag bend ζ_S . Both UCs must stay below 1, this is a hard constraint which should be met at all times. By defining a maximum allowable ζ_{max} lower than one, for example, at 0.9, more safety can be built in.

Workability

Hundreds of people are usually onboard the vessel during offshore pipelay, thus the faster the project is completed, the better. Therefore, the main goal of the controller is to perform pipe pulls as fast as possible when the ready-for-pull signal is given. Within the structural limits, a faster pull is always preferred over a pipe pull that consumes less energy.

Minimizing energy consumption

While the fastest pipe pull is the best pipe pull, within this (time) window, energy consumption must be minimised.

A lot of system-dependent data are required to accurately predict energy consumption as well as potential energy savings. First, all systems involved consume energy, from tensioners to welding equipment in the firing line. As a first assumption, these will be omitted, and only the energy involved in generating thrust by the vessel will be considered since these are assumed to be the most significant.

Additionally, this thrust generation is not trivial. Individual thrusters have different efficiencies at different thrust demands [28]. This implies that a higher thrust for a short duration of time can potentially be more efficient than a lower thrust demand for a longer period of time. A pipelay vessel has multiple thrusters. Therefore, it can be more efficient to use fewer thrusters at higher thrust demands or vice versa depending on the mentioned individual thruster characteristics. Therefore, this thrust allocation is complex and system-dependent. It is important to be aware of this, but it falls outside of the scope of this thesis to include in the control strategy.

Concluding the above, it is assumed that a lower thrust demand, in general, results in lower energy consumption. Therefore, in the rest of this thesis, thrust demands during a pull will be minimised and relating this back to minimising energy consumption will remain a recommendation.

The vessel's thrust demand T_v during a pipe pull, neglecting environmental disturbances, can be related to three phenomena: drag-related thrust T_d , inertia-related thrust T_i , and tension-related thrust T_t as shown in Figure 5.1.1.

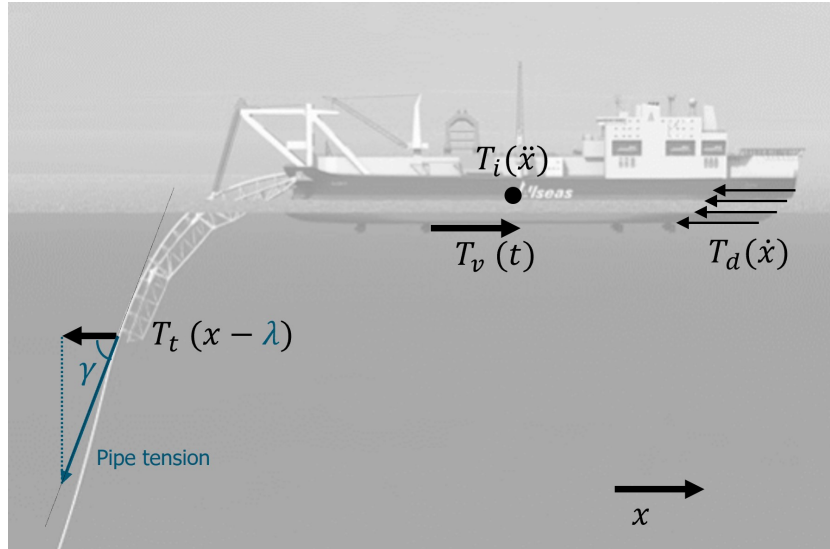


Figure 5.1.1: Thrust balance

All thrust demands will be introduced and as will be shown they depend on the vessel's surge motion x and their time derivatives $\dot{x}$ and $\ddot{x}$. Starting with the drag-related thrust which can be calculated as [28]:

$$T_d(t) = \frac{1}{2} \rho_w C_f S_w \dot{x}^2(t) \quad (5.1)$$

In which:

ρ_w = Water density [kg/m³]

$\dot{x}$ = Vessel speed [m/s]

C_f = Frictional resistance coefficient of the vessel with stinger deployed [-]

S_w = Wetted surface of the vessel [m²]

Since pulls are performed at relatively low speeds, this effect is minor. A greater effect is the inertia-related thrust due to accelerating and decelerating the vessel (and some water around the vessel, the added mass). The required thrust can be calculated as:

$$T_I = m_v \cdot \ddot{x}(t) \quad (5.2)$$

in which m_v is the effective mass of the vessel. This is commonly taken as 110 percent of the vessel's mass to take the added mass into account.

The third and last contribution is due to the horizontal tension in the pipe at the stinger tip. As indicated in Figure 5.1.1 this tension is dependent on the difference between the vessel motions x and pipe payout λ . This is because when the vessel moves forward, the pipe's tension increases and vice versa. The opposite happens when the pipe is paid out, a positive payout will result in a drop of pipe tension and vice versa. Finally, when the vessel moves a certain distance, and the pipe is paid out the same amount of distance, the same equilibrium is found, and the tension will not change. This tension fluctuation becomes less in deeper water but could be estimated with the catenary equation, see Section 3.1.

5.2 An adaptive control for changing circumstances

Considerations

Multiple options can be considered for the control strategy. As stated in the introduction, a different control strategy is used for shallow and deep water. In shallow water, the tension feedback is used, which results in an equal profile of the vessel surge and tensioner payout, and in deep water, a tensioner feedforward is used for the tensioner only. These two strategies can be combined by always using the feedforward for tensioner payout as well as the vessel surge, this way their timing can be controlled. In shallow water, these profiles need to be aligned, but in transitional or deeper water, the vessel's profile can be milder to save energy and ultimately result in a constant vessel speed. This fuses the shallow water and deep water strategies together. The UCs calculated on the previous pull(s) can be used to decide if a faster (improving workability) or more energy-efficient feedforward profile can be used. A similar feedback loop can still be used for the tensioner to compensate for long-term drift or significant sea states. Instead of tension as a control input, the same UCs can be used since these form a better representation of structural integrity than tension only. This way, a payout is desired when $\zeta_O > \zeta_{max}$ and a payin when $\zeta_S > \zeta_{max}$.

Subsequently, multiple options can be considered to determine the feedforward profiles themselves and their timing. As the data analysis concluded, the pipe pulls are not very consistent, and therefore, an adaptive rather than predictive algorithm is most suited.

Feedforward profiles

For the feedforward profiles the same profile as in the currently implemented pipe-ahead function is used. A different type of profile, utilising the drag to slow down the vessel, for example, could be looked into, but this is left as a recommendation since this is not trivial and would require a lot of system knowledge. The feedforward profile is defined with two parameters, a_{max} as a maximum acceleration and v_{max} as a maximum velocity. Combined with the goal of a 24 meter pull, the profile is fully defined. An example is shown in Figure 5.2.1.

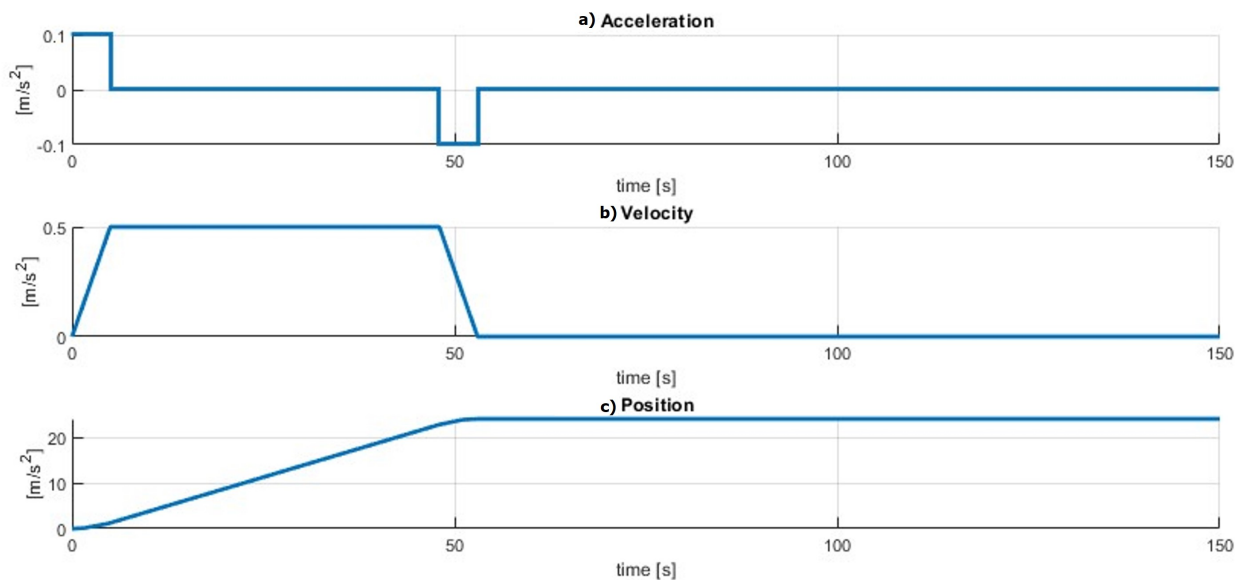


Figure 5.2.1: Feedforward reference profile with $a_{max} = 0.1[m/s^2]$ and $v_{max} = 0.5[m/s]$

Controller operation

The strategy and its "adaptiveness" will be explained based on [Figure 5.2.2](#), starting from shallow water with an undesired, wild sea state to deep water with a mild sea state in 5 steps, corresponding with 5 control modes. In [Figure 5.2.2](#) an equal thin line is drawn in all 5 control modes to visualise the mutual differences. The profiles corresponding to each mode are shown in the bold dashed lines. Here, only 5 control modes are used as an example, but this can be set higher to achieve a more continuous transition between shallow water and deep water.

In shallow water with a wild sea state, a slow pull is performed with, in theory, no difference in between the vessel surge and tensioner payout. If, in practice, this relative position due to good timing and correct feedforward remains small, the UCs will not exceed ζ_{max} , and the feedback control will not intervene. The values for the UCs will be recorded and evaluated after a few pulls. The amount of pull can be set arbitrarily as v_p .

If, after a few pulls, it shows that the UCs are still relatively high, the mode remains the same. If, after a few pulls, it turns out that the UCs are relatively low, for example, due to improving sea state conditions or an increase in water depth, the controller will adapt. In this case from mode 1 to mode 2 by increasing v_{max} and a_{max} for both the vessel and tensioner. This results in a faster pull, thus increasing the workability. Here, mode 2 already indicates the maximum allowable settings, but this can be done in more than one step.

Now that workability is maximised, implying the fastest possible pull, the energy consumption can be minimised. This can be easiest achieved by lowering v_{max} and a_{max} for the vessel only. This requires an earlier start of the vessel surge motion and/or a longer end. Since the data analysis from Chapter 2 shows that it is hard to predict the ready-for-pull signal, only the latter is possible. This results in position difference since the vessel's profile starts to lack behind, as can be seen in mode 3 and higher. This will result in all thrust-demanding terms to decrease (T_d , T_i and T_t). Due to this position difference, a relatively mild sea state and/or deeper water is necessary for the position-tension coupling to weaken. After a few times of adapting this strategy, it will converge to a constant vessel speed in deep water where the energy consumption is minimal. The previous explanation was from shallow to deep water (mode 1 to mode 5), but this works the other way around as well.

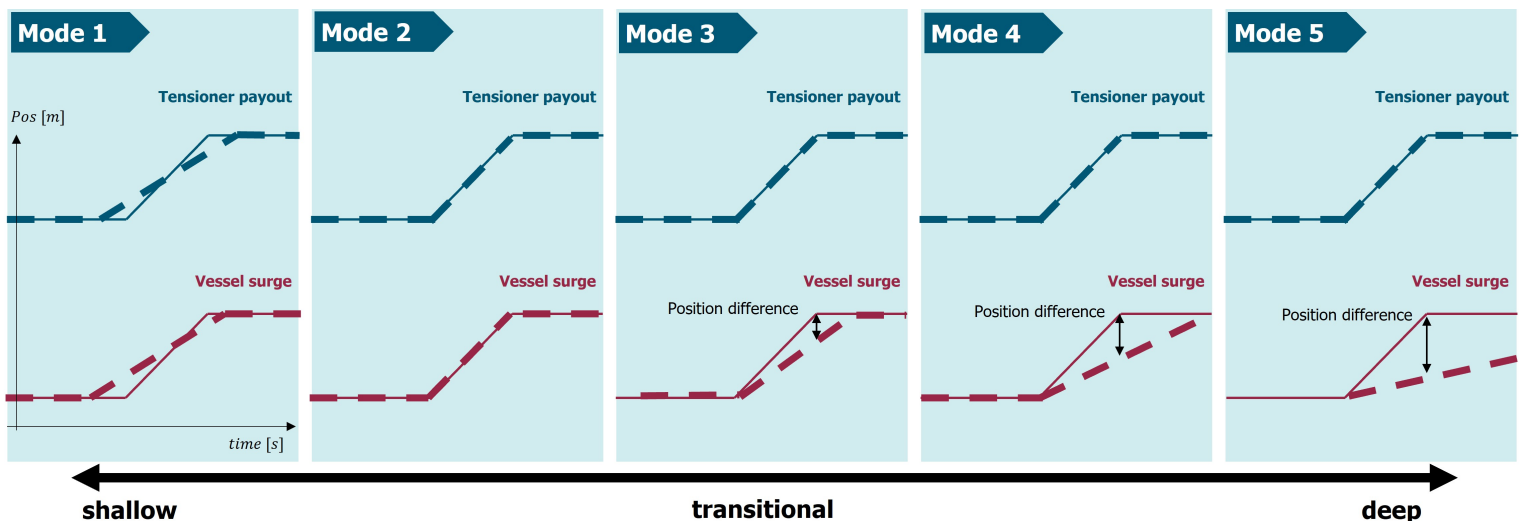


Figure 5.2.2: Adaptive control for changing circumstances

Implementation

The previous Subsection explained how the controller operates in words, but implementation-wise, there are some things that need to be set up. A schematic block diagram of the control architecture is shown in [Figure 5.2.3](#). Here, the two control blocks and the pipe model are shown. The inertial measurement unit (IMU) measures the vessel's motions and is fed into the pipe model. The calculated UCs (ζ_O and ζ_S) are fed back into the feedback controller to act if ζ_{max} is exceeded and the feedforward controller to be recorded.

In the feedforward controller, the ready-for-pull signal is also fed to activate a feedforward profile for the tensioner payout and the surge of the vessel. The surge is directly fed to the DP system. The payout, minus potentially a feedback payout, is fed to the motor drives of the tensioner. An implementation of the control logic of the feedforward controller is given in [Algorithm 4](#) and the feedback controller in [Algorithm 5](#). In this algorithm, C_m indicates the current control mode, and C_{max} is the number of different control modes. Additionally when ζ_{low} is defined under which all UCs must stay in order to increase the mode and a ζ_{up} is defined above which the UCs must come at least ones for the mode to be decreased again. this prevents endless mode switching. For an example is referred to [Figure 6.3.4](#).

Finally, a problem arises after multiple pulls. Since the firing line is included in the model (about 200 meters or 8 pulls), the simulation can be run without problems, but after that, the firing line runs out. To ensure this is not the case, additional elements should be added after each pull. It is convenient to take an element size that is an integer divisor of 24 meters, such as 6 meters. The introduction of these elements and the jump of the endpoint, as shown in [Figure 5.2.4](#), can cause numerical instabilities. Simulations show that when adding these elements pre-stressed with the axial tension of the first element, the next timestep smoothly continues without any problems. Without pre-stressing, there is a large difference in tension between the new and old elements, which will induce axial oscillations in the firing line, which are undesirable.

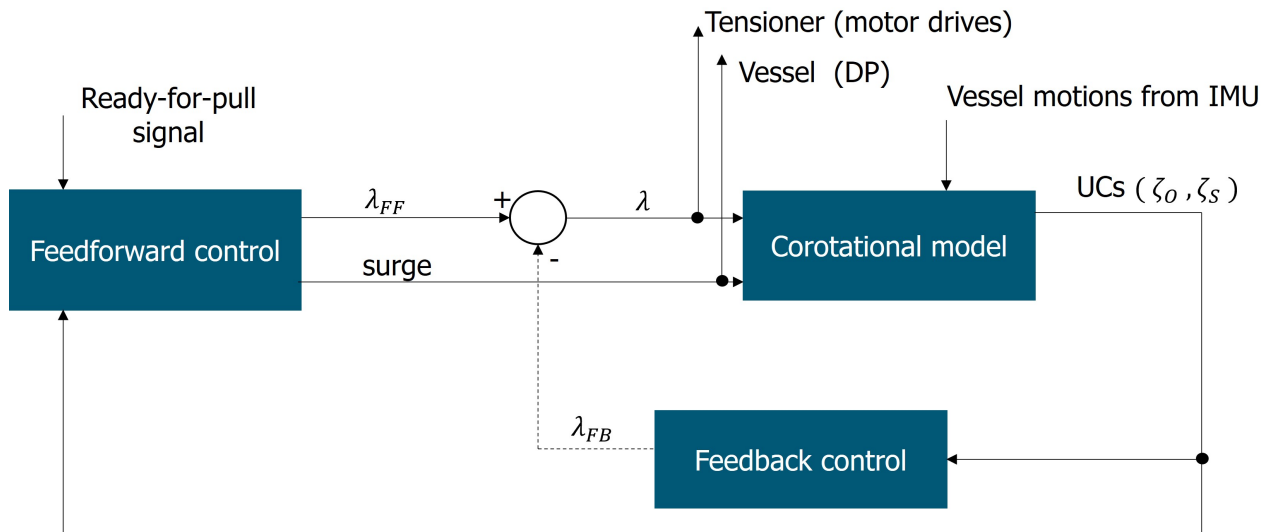


Figure 5.2.3: Schematic block diagram of the control architecture

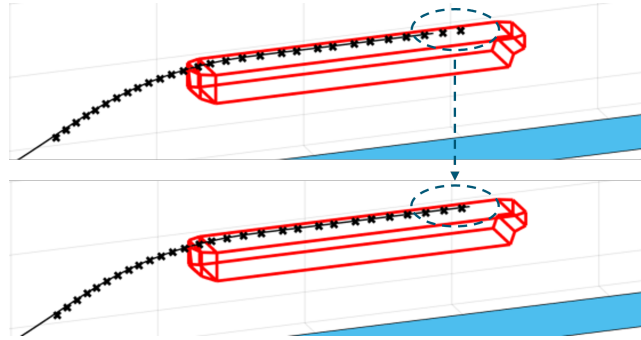


Figure 5.2.4: Adding elements after pull

Algorithm 4 Feedforward control

```

1:  $v_i = 0$ 
2:  $C_m = 1$ 
3: start recording UCs
4: while process is active do
5:   if ready for pull == 1 then
6:     Perform pull with  $C_m$ 
7:      $v_i = v_i + 1$ 
8:     Add elements
9:   end if
10:  if  $v_i = v_p$  then
11:    if  $\zeta_O$  AND  $\zeta_S < \zeta_{low} \forall$  time AND pull profile active <  $C_{max}$  then
12:       $C_m = C_m + 1$ 
13:    else if  $\zeta_O$  OR  $\zeta_S > \zeta_{up}$  Ones AND pull profile active > 1 then
14:       $C_m = C_m - 1$ 
15:    end if
16:     $v_i = 0$ 
17:    reset recording UCs
18:  end if
19: end while

```

Algorithm 5 Feedback control

```

1: while process is active do
2:   if any  $\zeta_O > \zeta_{max}$  OR  $\zeta_S > \zeta_{max}$  then
3:     calculate payout with PD controller
4:   end if

```

5.3 Concluding remarks

Since feedback control will intervene when the UCs come above ζ_{max} , the structural integrity is always ensured. When the UCs remain very small for a few pulls, the controller will adapt in order to maximize workability and subsequently minimise energy consumption. When the UCs become larger, the adaptive controller will scale back. This way, all three control goals are met with a relatively simple controller implementation. Therefore, the proposed algorithm automatically incorporates changing weather conditions as well as changing water depths, thus relying less on experienced operators.

The definition of energy consumption, however, is primitive and, therefore, needs further research, taking system-specific data into account.

Chapter 6

Results

This chapter presents simulation results to validate how the model and proposed controller perform. Three types of simulations are performed on two scenarios. First, the static configuration is computed and verified with OrcaFlex for a typical shallow and deep water scenario. This is done using the old and the enhanced contact formulation to make a comparison between the two. The configuration, tension and curvature in the pipe are discussed.

Second, a full dynamic simulation is performed to compare the model with OrcaFlex without a form of payout control. The absence of payout control makes it a suitable stress test for the pipe model subjected to vessel motions due to waves. Tension, curvature and unity checks are discussed.

In the last simulation, a series of pipe pulls is performed with the controller in place. This simulation forms a proof of concept for the adaptive feedforward algorithm and its implementation. Finally, conclusions are drawn on the results, and comments are made on the trade-offs between accuracy and computational time.

6.1 Statics

As stated in the introduction, a shallow water and deep water scenario will be discussed for statics. Both scenarios are based on past projects. The shallow water scenario is a simulation with Allseas's vessel Audacia, laying a large axially stiff pipe with a water depth of 71 meters. The deep water scenario, in this case, is represented by a less axially stiff pipe laid with the Lorelay at a water depth of 1782 meters. The relevant simulation parameters are summarised in Table 6.1 with in between brackets the OrcaFlex parameters if they differ from the corotational approach. The resulting static configuration after the initialisation algorithm as discussed in section 4.5 is shown for both scenarios in Figure 6.1.1.

Table 6.1: Simulation parameters

Property [Unit]		Shallow water	Deep water
Young's modulus [GPa]	E	207	207
Poisson's ratio [-]	ν	0.3	0.3
Stress Diameter [m]	D_σ	0.762	0.464
Penetration Diameter [m]	D_{pen}	0.853	0.464
Inside diameter [m]	D_{in}	0.725	0.401
Density seawater [kg m^{-3}]	ρ_{sea}	1.025	1.025
Density air [kg m^{-3}]	ρ_{air}	1.25	1.25
Waterdepth [m]	WD	71	1782
Specific weight [kg m^{-1}]		671	300
Buoyant weight [kg m^{-1}]		85	127
Elements [-]		100 (280)	200 (410)
Pipe length [m]		700	3000
Vessel		Audacia	Lorelay

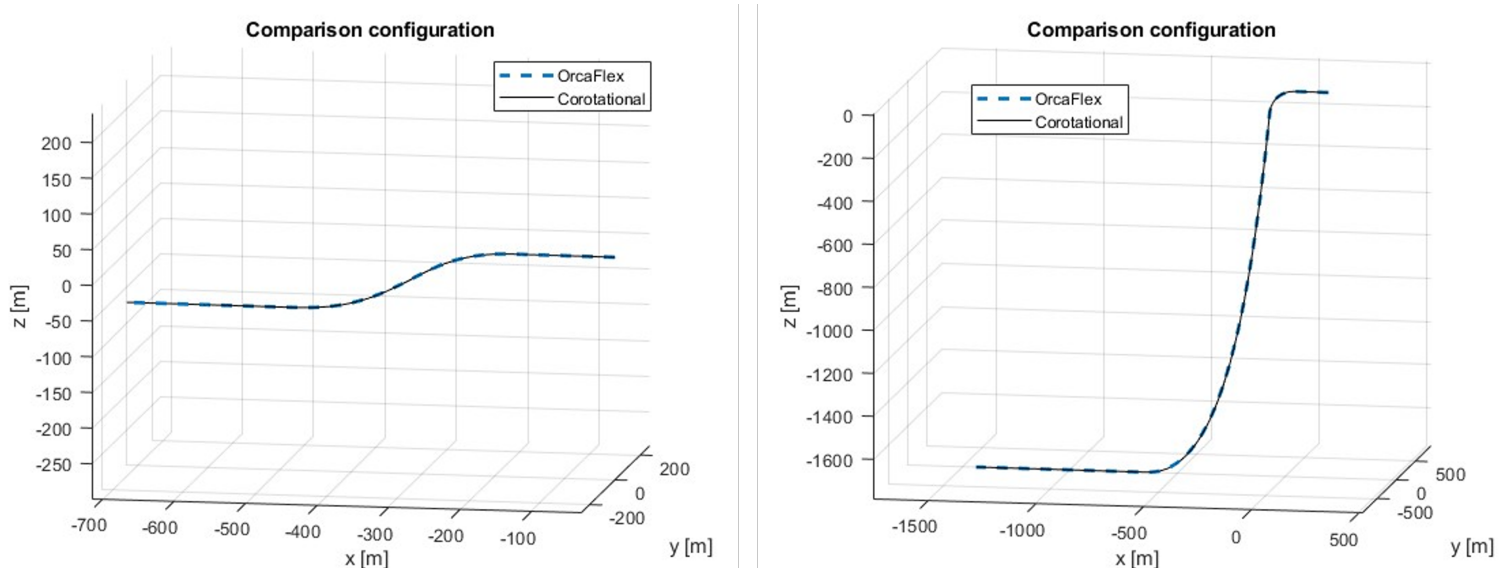


Figure 6.1.1: Configuration plots for shallow (left) and deep (right) water scenario.

As can be concluded from these figures, the configurations look in agreement with each other, but at this scale, it is hard to see differences. Therefore, looking at the tension along the arc length as well as the curvature is a suitable comparison measure. Three sets of plots are shown here, respectively showing shallow water with the old contact formulation, shallow water with the new

contact formulation and deep water with the old contact formulation in Figure 6.1.2, Figure 6.1.3 and Figure 6.1.4.

Looking from 0 onwards, first, the pipe in the firing line is shown with constant tension and small curvature. This curvature represents the sagging of the pipe between rollerboxes. The second phase is the overbend, in which tension decreases and the pipe is bent. Characteristic is here for shallow water, an inflection point after which the sagbend begins immediately, and for the deep water, a large pipe length with no curvature slowly building up to the sagbend. Last there is the pipe length on the seafloor with no curvature. In the corotational formulation, no seafloor stiffness is modelled, and thus, the tension remains constant. OrcaFlex uses a seafloor stiffness model, and therefore, the tension decreases further. Since this pipe length is not relevant to this thesis, this difference has no influence. In section 4.4 the difference between effective and wall tension has been explained. To clearly show the difference the reference configuration is also shown such that it becomes clear that the two tension lines start to separate at arclength that the pipe becomes submerged.

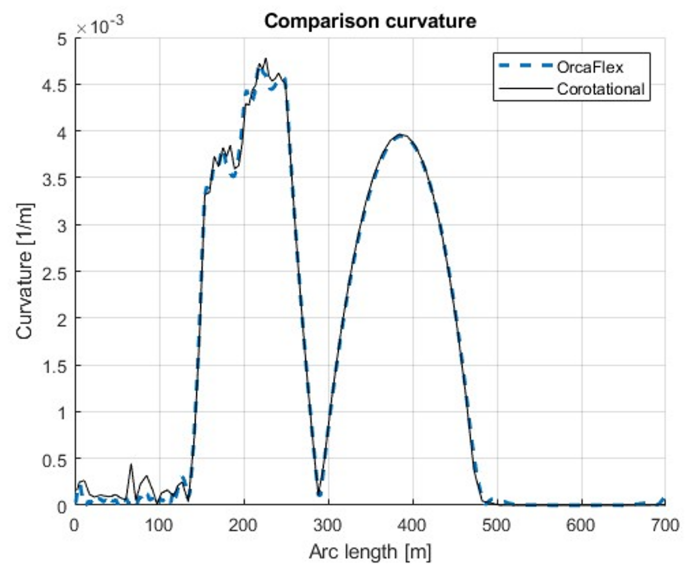
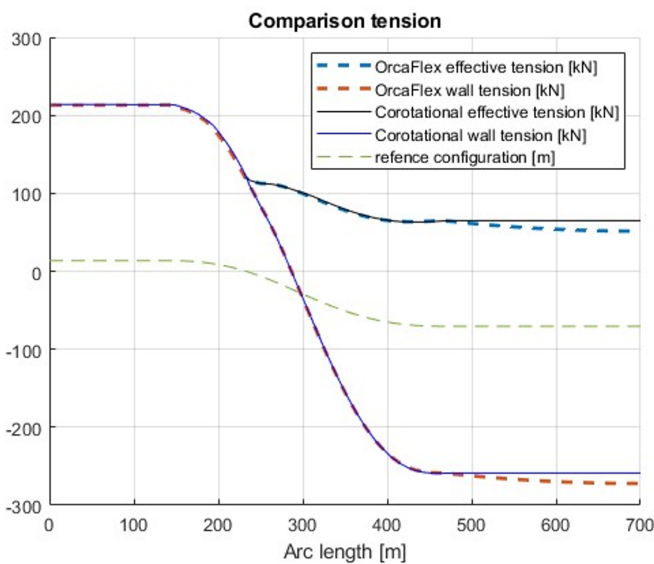


Figure 6.1.2: Tension and curvature plots along arc length for shallow water scenario with the old contact method.

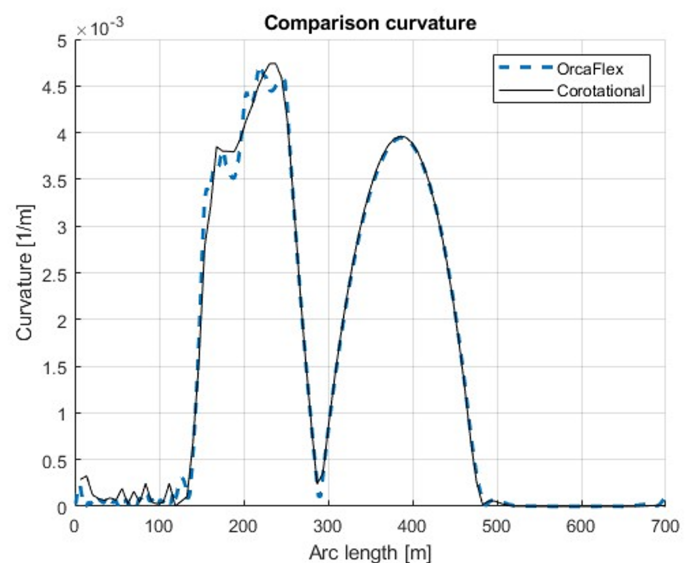
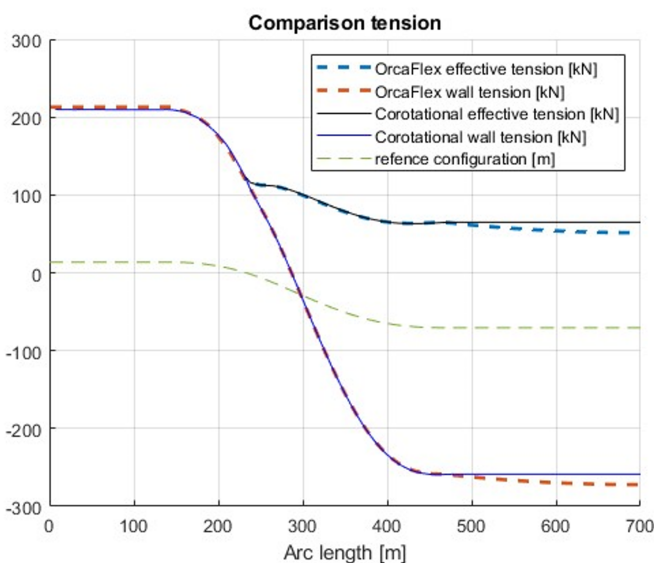


Figure 6.1.3: Tension and curvature plots along arc length for shallow water scenario with the new contact method.

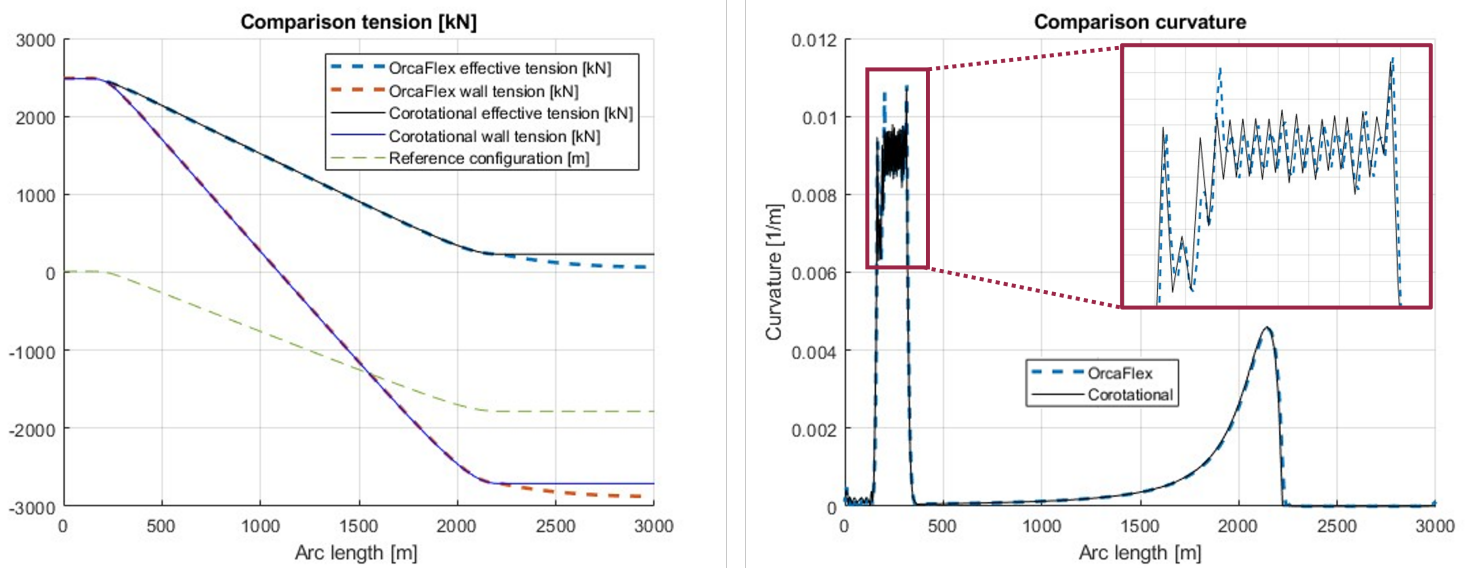


Figure 6.1.4: Tension and curvature plots along arc length for deep water scenario with the old contact method.

First of all we can conclude that all plots of the corotational formulation are in good agreement with OrcaFlex. We can also see a small difference in the overbend with the new contact formulation. Although noticeable, it is still in good agreement and good enough for the unity check calculation on which the controller should act. The last conclusion will be further substantiated in the next section.

6.2 Dynamics

Simulator

During pipelay, it is very insightful to see and visualise the current state of the pipe. With that in mind, the following visuals are created during simulation. These results can be played back for analysis afterwards in the simulator.

First, the vessel and pipe configuration is shown in [Figure 6.2.1](#). Additionally, the vessel's motions are recorded in [Figure 6.2.2](#). Also, see [Figure 6.2.3](#), at every time instant, the curvature and effective tension are plotted against the arc length resulting in a dynamic deviation band. For reference, the static equilibrium is also plotted. With this curvature and effective tension, the nodes corresponding to the overbend and sagbend are located at every time step, and the unity checks are calculated for these nodes. The maximal occurring unity check in the overbend and sagbend for each timestep are compared in [Figure 6.2.4](#).

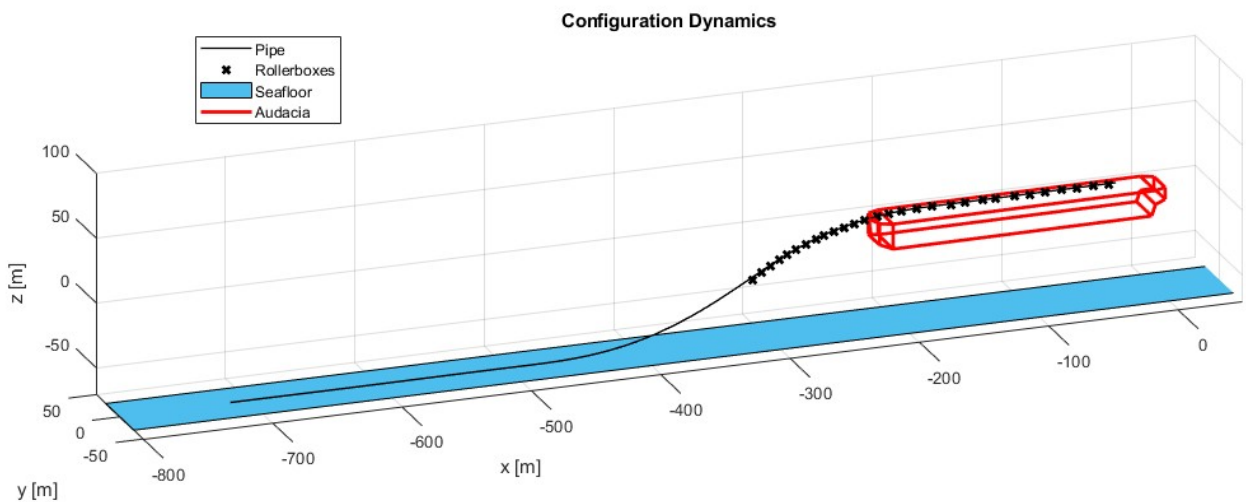


Figure 6.2.1: Configuration dynamic simulation

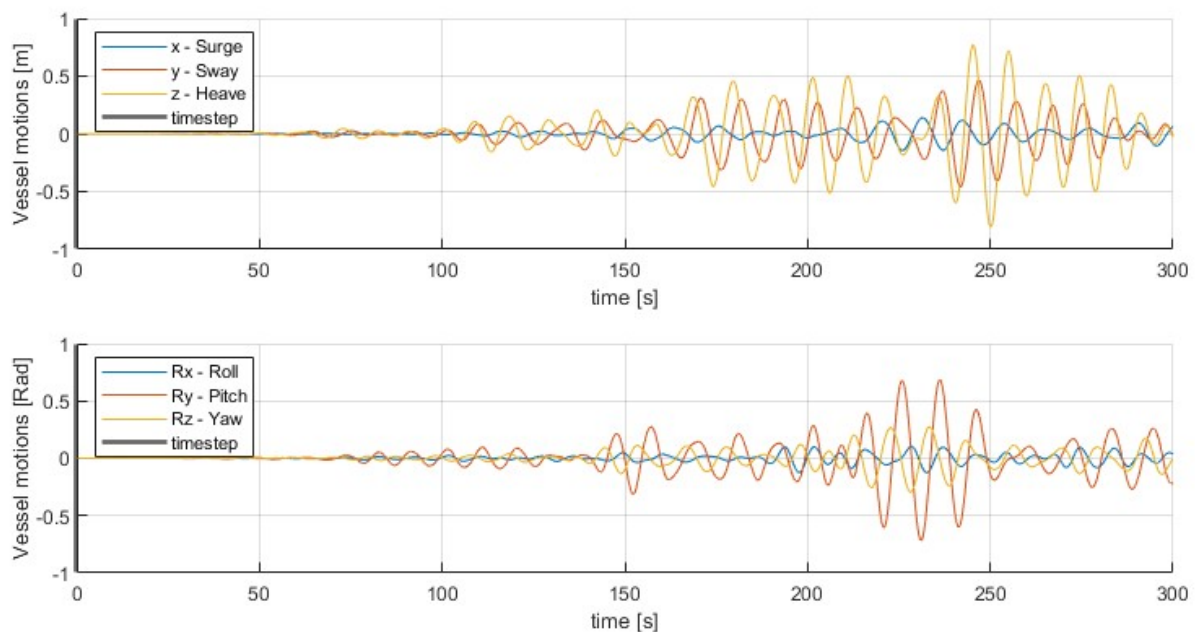


Figure 6.2.2: Vessel motions due to waves

Simulation setup

The shallow water scenario introduced earlier is used here to validate how well the model performs dynamically. In OrcaFlex, an ISSC wave type, in other words a modified Pierson-Moskowitz spectrum, is used with a significant wave height H_s of 1.5 meters, a mean wave period T_z of 7 seconds and a direction of 110 degrees. The sampled waves are multiplied by the Response Amplitude Operator (RAO) of the Audacia to achieve the vessel's motions. These exact motions are extracted from OrcaFlex, using its API to put on the corotational formulation to perform the same simulation. This is logical since, during pipelay, the vessel's motions will be measured and form the same input for the model.

In OrcaFlex, a deadband type of tensioner is included. Since the goal of this simulation is to validate the pipe model's performance, this tensioner's payout is set to 0, so it will not intervene or impact the results.

Results

In the figures mentioned earlier, the static equilibrium is shown at 0 seconds in the configuration plot. A simulation is performed for 300 seconds, showing the vessel's motions. Since tension and curvature fluctuate heavily (especially with no tensioner) a deviation bandwidth is shown in the plot to give a sense of the dynamic behaviour. Due to the different mesh in Orcaflex and the corotational formulation, it is not obvious to calculate the difference in curvature and tension in time, but since the goal of the model is to calculate the unity checks anyway, they form the validation between OrcaFlex and corotational formulation.

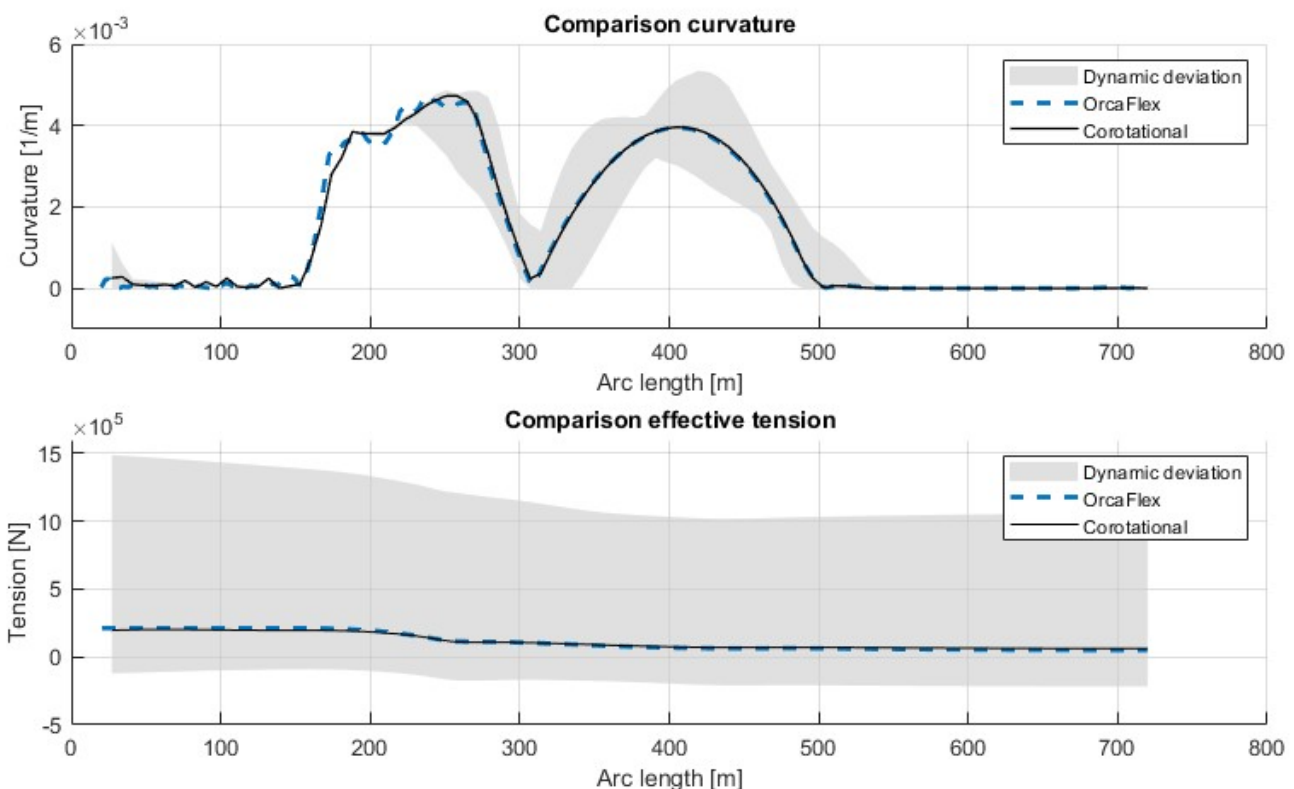


Figure 6.2.3: Dynamic deviations of curvature and tension with respect to static equilibrium

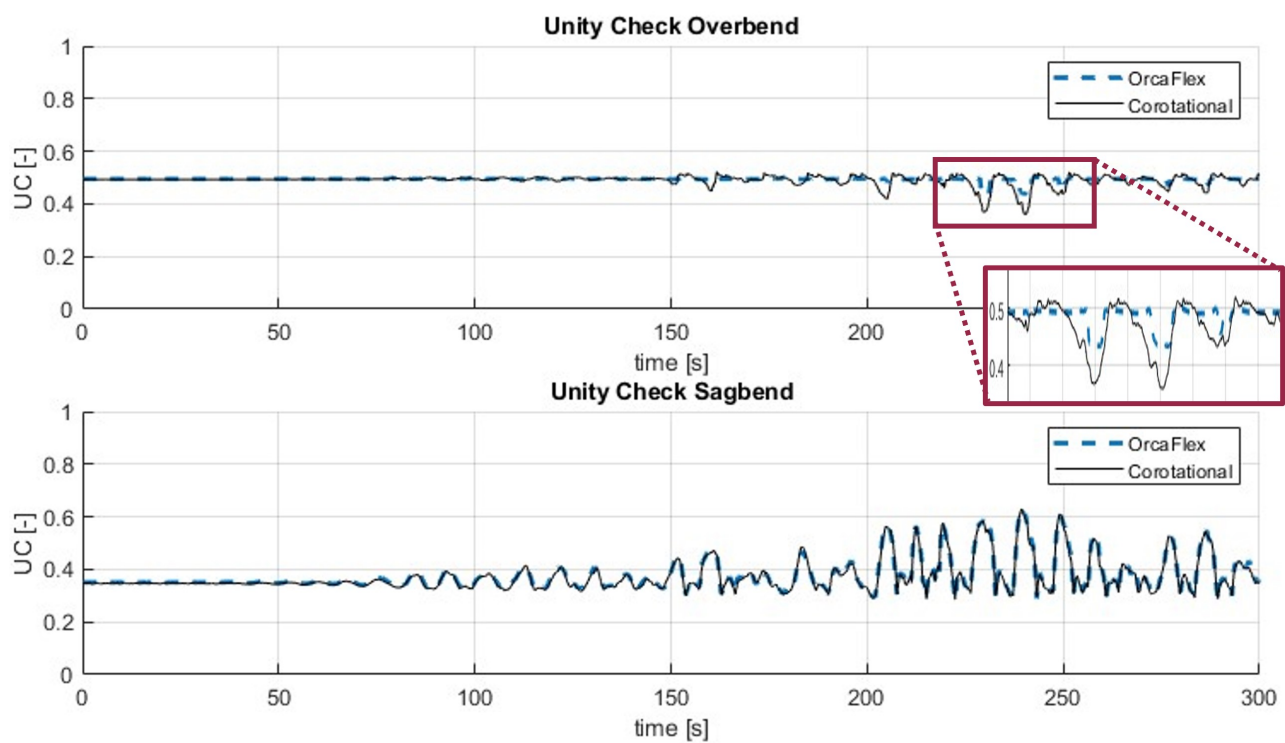


Figure 6.2.4: Resulting unity check in overbend and sagbend

6.3 Control

Simulation setup

In the previous simulations, the modelling was validated using OrcaFlex. The goal of this simulation is to show the implementation of the controller. To clearly show the effect of the controller itself, and because the implementation of this controller in OrcaFlex would be far from trivial, only the corotational approach is shown here without vessel motions due to waves. This is done with the same shallow water set-up, thus the same parameters as shown in Table 6.1. Only one exception is the number of elements is increased to 120 to achieve a element length of 6 meters instead of 7 meters to be a divisor of 24 meters such that after each pull exactly 4 elements are added.

First, the parameters, as explained in Chapter 5, are set. In Chapter 5, the general controller is explained using 5 modes from shallow-transition-deep water. Since this is the simulation for shallow water specificity 7 modes only incorporating shallow-transition are used.

The feedforward profile parameters are shown in Table 6.2. With these 7 profiles, 7 modes are set as shown in Figure 6.3.1. The profiles used in each mode for the vessel surge and the tensioner payout are shown between brackets in the legend. Starting from modes 1 to 3, the workability is optimized by achieving a faster pull while the vessel and tensioner have the same profile. Next, in modes 4 to 7, energy consumption is minimized by lowering the acceleration and maximum speed of the vessel. During these modes, the payout is kept the same, thus prioritizing workability. This results in a position difference between the profiles, as is shown in subfigure h of Figure 6.3.1. This difference, in the beginning, is due to the acceleration difference, and the linear increasing part in the middle is due to the difference in maximum speed. Later, the vessel's profile catches up again, bringing the difference back to 0.

Regarding the controller, ζ_{max} is set at 0.95, ζ_{low} at 0.7, and ζ_{up} at 0.85. Since no wave motions are included, every pull will look the same, so instead of evaluating the UCs every few pulls, this is done every pull. During production, the ready-for-pull signal will come when the welding is done. In this simulation, this signal is given at timestep 1 second, and repeated every 150 seconds for 9 times in total.

Table 6.2: Feedforward profile parameters

Mode	$a_{max} [m/s^2]$	$v_{max} [m/s]$
Profile 1	0.1	0.200
Profile 2	0.1	0.300
Profile 3	0.1	0.500
Profile 4	0.085	0.500
Profile 5	0.07	0.475
Profile 6	0.06	0.470
Profile 7	0.05	0.465

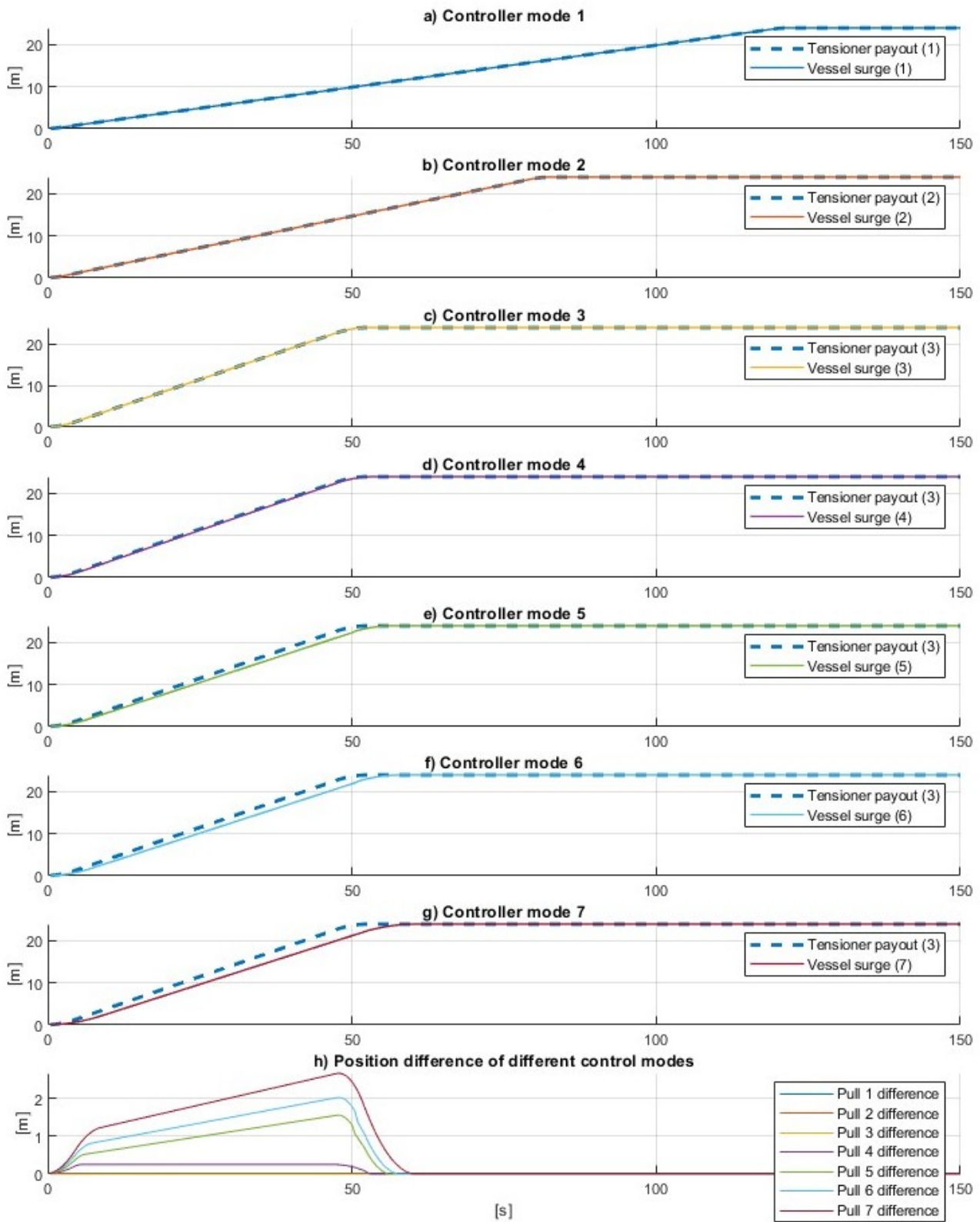


Figure 6.3.1: Different control modes

Results

In Figure 6.3.2, the resulting vessel motions due to the controller are shown. Since there are no waves, only the surge motion is non-zero, in which the 9 pulls are clearly visible. In Figure 6.3.3, the vessel surge is plotted against the tensioner payout in subfigure a. In subfigure b, these pipepulls are normalized. The difference in the first 3 pulls is clearly visible. The latter pulls are not that clear, therefore in subfigure c the position difference is plotted.

These resulting pulls are due to changing feedforward modes. To indicate this, the UCs and selected control modes are plotted in Figure 6.3.4. Starting from modes 1 to 3, the UCs are hardly affected due to the same pull profiles for the vessel and tensioner. From mode 4 to 6 the sagbend's UC starts to increase since the payout is more than the surge resulting in increased compression as well as increase curvature. From mode 6, a part of the pull results in a UC above ζ_{low} but below ζ_{up} , which is the sweet spot and therefore indicated in green. From this mode, no increase or decrease in mode is desired. Therefore, the remaining pulls are in mode 6.

Zooming in on the resulting UCs of Figure 6.3.4 it shows a positive correlation with the position difference of the control modes shown in subfigure h of Figure 6.3.1. This is the "vessel's motion - tension fluctuation coupling" as typical for shallow water as explained in Section 2.1.

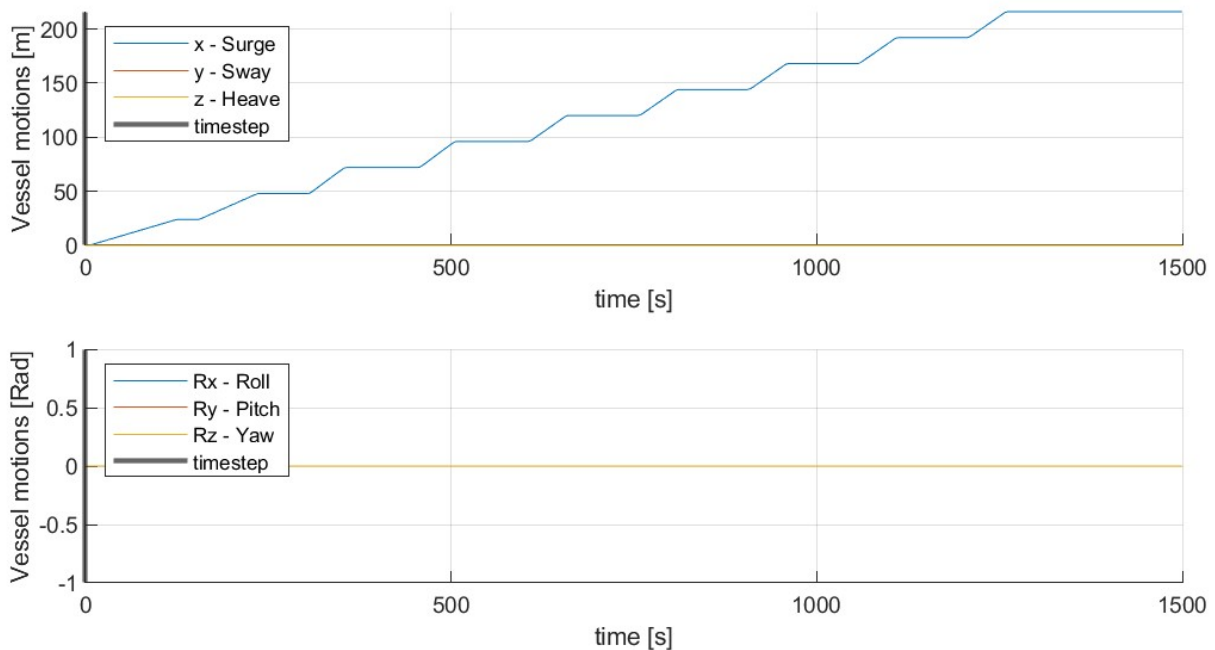


Figure 6.3.2: Vessel motions due to the controller

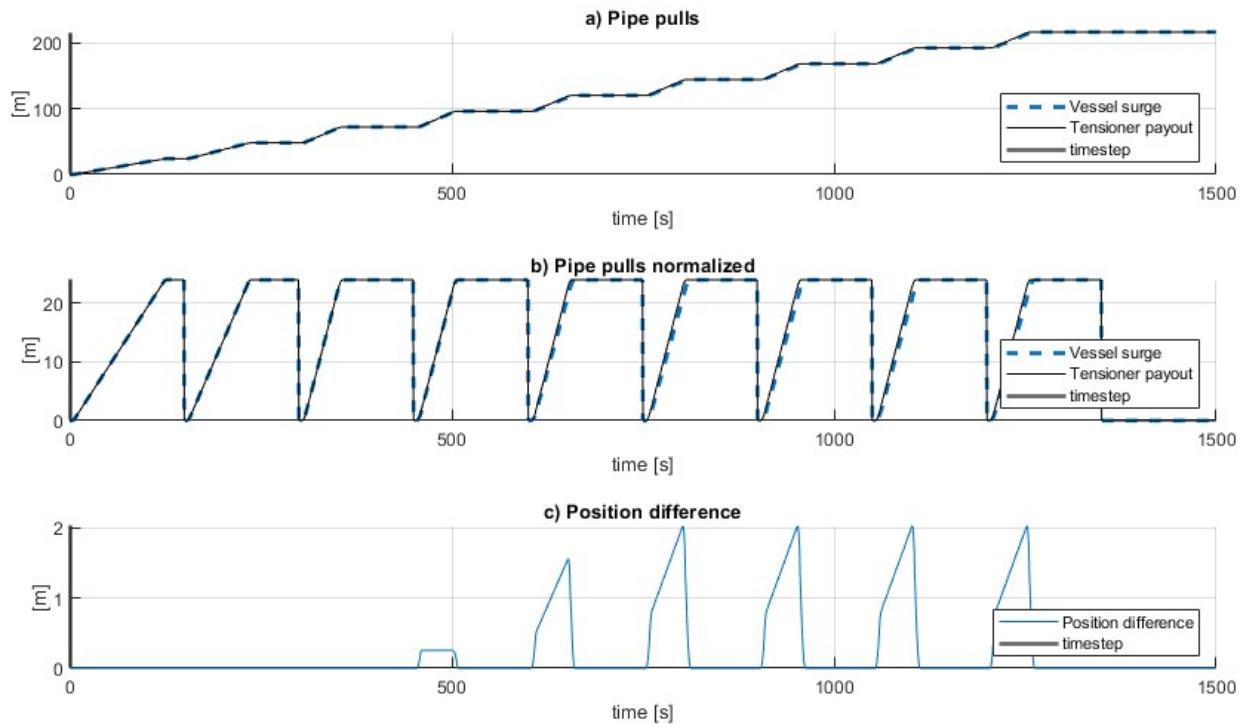


Figure 6.3.3: Pipe pulls due to the controller

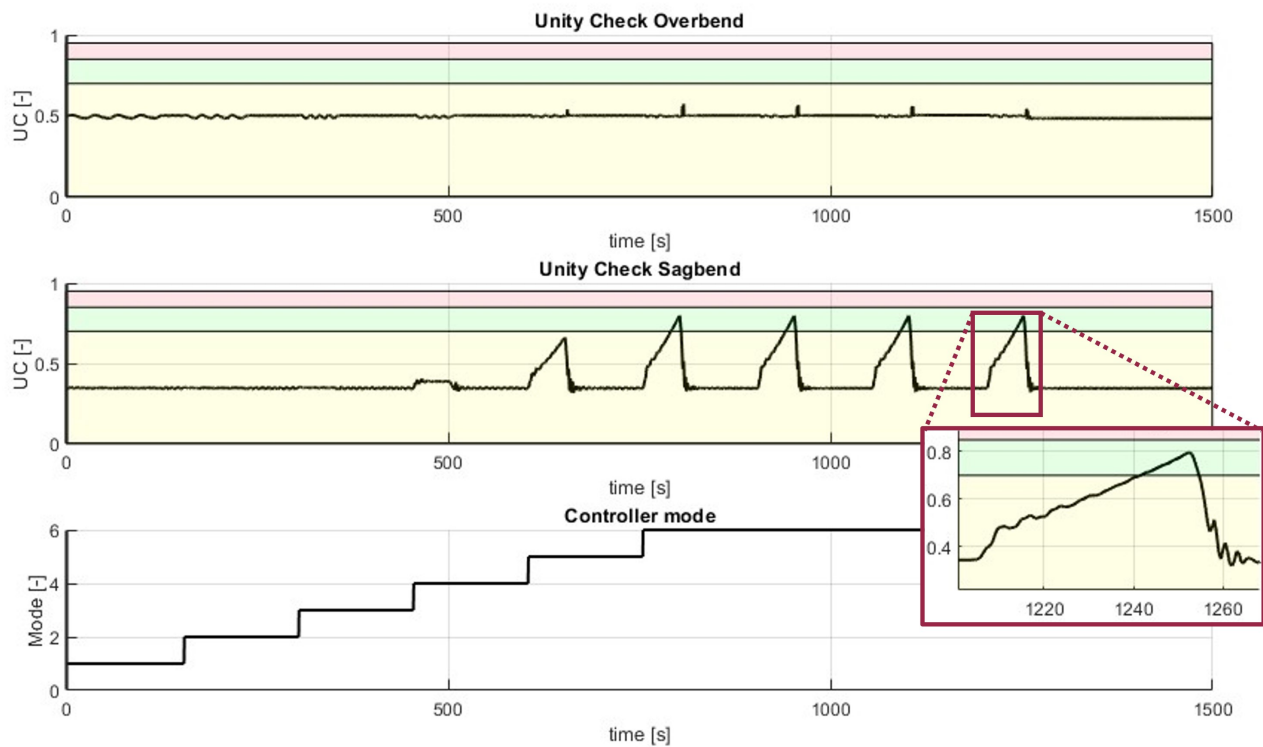


Figure 6.3.4: Resulting unity check in overbend and sagbend

6.4 Concluding remarks

From the static, dynamic and control simulation, some concluding remarks can be made:

Statics

Regarding the static results, we can conclude that the results of the corotational formulation described in this thesis are in agreement with OrcaFlex for the old and new contact formulations. Characteristic plots for curvature and tension are found for shallow and deep water scenarios. Comparing the new and old contact formulations, it can be seen that in the overbend, a slight difference is shown in the curvature peak in the new formulation. In this simulation, the corotational approach is discretized with fewer elements since this reduces the simulation time. However, when the simulation is discretized using more elements, this difference is reduced significantly.

Dynamics

Similarly, the dynamic results of the corotational formulation are also in good agreement with OrcaFlex. Without any form of payout, axial damping or tensioner in place, the axial tension fluctuates a lot, but this is to be expected. The resulting unity checks also show agreement between both models. During the more significant vessel motions, the UC of the overbend shows some deviation. The corotational formulation fluctuates (both positively and negatively) more than the OrcaFlex results. A deeper analysis showed that this is due to a relatively small difference in the overbend curvature peak, as described in the static results.

This simulation is done with the new contact formulation, which is required for payouts during the control simulation. Additionally, the simulation must be run in real-time. In applications where this is not required, such as during the PIT phase, the old contact formulation or a local mesh refinement will increase the accuracy in statics. It is also very likely that this will also yield more agreement in the UCs in dynamic simulations.

Since when pipe pulls are made, the nodes move over and away from the overbend, the old contact formulation cannot be used, and local mesh refinements are not trivial since the discretization needs to be re-meshed during simulation. Therefore, a well-considered trade-off between accuracy and computational time must be considered, depending on the application.

Control

The control simulation demonstrates the working principle for the shallow water case. Therefore, it can be concluded that the controller works and performs well in this ideal simulation environment. This includes the element-adding algorithm, the selection of the correct next mode, and online calculation of UCs of the previous pull.

However, more simulations are required for different scenarios and actual sea states to build trust and validate the controller's robustness before it is ready to be implemented offshore. While doing this, the algorithm can already be used in parallel with the current systems as a visualisation tool and to suggest pipe-ahead profile settings to operators.

Chapter 7

Conclusion and recommendations

This thesis presents an approach for the modelling of the offshore pipelay process to obtain a real-time estimate of the pipe's structural loading. This estimate can give input to a high-level control algorithm that optimises workability and minimises energy consumption within the pipe's structural limits. This thesis also presents a proof of concept for such a high-level control algorithm.

The model and control algorithm explained in this thesis are the result of a set research objective. This objective and the sub-objectives are repeated for convenience. Based on this, conclusions are drawn, and recommendations are made.

7.1 Research Objective

The goal of this thesis is twofold. The first is to create a model that provides a better understanding or a real-time estimate of the limits of the pipe. This limit is highly dependent on the water depth, pipe geometry, stinger configuration, coating, and material which should all be taken into account. The second goal is to come up with a more efficient control strategy using this model and/or a different interaction between the vessel and tensioner. This should be researched and substantiated with simulations, preferably incorporating past project data.

From those main two goals, the following sub-objectives can be set up:

- 1:** Identify which models are already present/currently used within Allseas.
- 2:** Search for alternative, potentially more detailed, models that can be used.
- 3:** Create a structural model of the pipe, tensioner and vessel that includes the effect of water depth, pipe geometry, stinger configuration, coating and material.
- 4:** Determine the control strategy/algorithm that obtains the best mix between:
 - Maximum workability
 - Minimal energy consumption
 - While being constrained by the structural integrity of the pipe.
- 5:** Validate the correct implementation of the model and its accuracy by comparing it with commercial software. Additionally, validate that the controller performs as intended.

7.2 Conclusion

An extensive background and data analysis of past projects has been conducted to conclude on the relevant aspects of the offshore pipelay process. Especially structural loading and control parameters for shallow and deep water pipelay and how these change in the transitioning phase. Currently, Allseas has a worst-case structural loading check every few hours for the structural integrity during pipelay and controller settings are manually tuned by experienced operators. The current process lends itself to the possibility of optimisation and automated tuning based on the actual loading during the process.

To provide an estimate of the actual loading state of the pipe from a literature review, it has been concluded that an existing corotational FEM approach suits the application best. Mainly because of the complex non-linear phenomena that occur during offshore pipelay. In the current contact formulation, the nodes of the beam elements need to align with the location of the rollerboxes to calculate the contact forces in the overbend. During a significant payout, or a pipe pull, the pipe, and thus the nodes will move over the roller boxes, and therefore, this contact definition will not suffice anymore. To overcome this, the contact algorithm has been enhanced to calculate the contact force at an arbitrary location in the element and calculate an equivalent load vector to represent the contact forces at the nodes.

Regarding control, some individual parts are currently already solved and implemented, such as a tension control in the pipe or the vessel setpoint to ensure the pipe will end up at the target zone on the seafloor. However, a high-level control scheme that considers the current structural loading and performs an online optimisation has not yet been implemented. Since the data analysis showed that the timing of the pipe pulls is not very consistent, an adaptive rather than predictive control strategy is more suited. An adaptive feedforward algorithm is proposed that takes the structural loading into account in the form of the DNV-GL unity checks in the overbend and sagbend of the past few pipe pulls and decides if there is room for a more optimal pipe pull profile. The pipe pull profile prescribes the tensioner payout and vessel surge. With deeper water and milder sea states, it is possible to accelerate faster and have a higher constant speed for both the tensioner payout and the vessel surge, thus maximising workability. Continuing the vessel's surge profile can start to lag behind the tensioner payout, thus requiring less acceleration and a lower maximum velocity of the vessel, minimising the energy consumption while the tensioner payout and, thus, workability remain the same. The unity checks may rise as the water becomes more shallow and/or a worse sea state. If this happens, the adaptive algorithm will suggest a more conservative pipe pull profile again.

The model has been validated with the current industry standard OrcaFlex for a shallow water and a deep water scenario. These two scenarios are based on two past projects with different vessels, pipe sizes, coatings, and water depths. Static and fully dynamic validation is conducted, and the corotational approach shows good agreement with the OrcaFlex models. A robust initialisation procedure is set up to go from a straight pipe to the static equilibrium S-shape. Therefore, no initial guess is required. Additionally, a simulation is performed in which the adaptive feedforward algorithm concept is proved. However, more (detailed) simulations should be performed to draw clear conclusions on its effectiveness and the amount of optimisation that is achieved compared with the current process.

7.3 Recommendations

This thesis continued on the works of de Vries [**VriesFransVries**] and contributed mainly to the initialisation algorithm, the tensioner implementation and foremost, the contact algorithm to allow for pipe-pulls. Among others, these were also his recommendations. Yet more recommendations can be made on the current enhancements.

Computational efficiency and modelling

First, it was tested that the implementation of de Vries in C++ yielded real-time performance due to the efficient modelling and consistently derived effective stiffness matrix. The latter ensures that the Newton-Raphson algorithm will converge quadratically. This thesis was implemented in MATLAB for its ease of use and compatibility with the OrcaFlex API, which allows easy comparison and visualisation between results. The disadvantage is that it is slower. Additionally, the new contact formulation's contribution to the effective stiffness matrix is an approximation, thus quadratic convergence was lost. Therefore, it is recommended to implement the model back again in C++ and look into the derivation of a consistent effective stiffness matrix.

Additionally, plasticity and the second-order terms in the elements are here omitted. These can be added back into the model. The latter will allow for larger elements, which yields faster solving again. Presumably, a larger element will make the straight-line assumption in the enhanced contact method no longer valid. Thus, this curvature should be incorporated as well.

It is also noted that a significant part of the computational time goes to evaluating the effective stiffness matrix, which is evaluated in every iteration. However, near convergence no significant rotations occur, but mostly small contact-related adjustments are made, hence a form of the modified Newton-Raphson method might be an interesting improvement to look into.

Control

Regarding the proposed controller, it is recommended that more simulations are performed in more diverse sea states and with different and changing water depths to tune the feedforward profiles better. Also, the thrust requirements are identified as a summation of drag, inertia, horizontal tension, and neglected environmental disturbances. More research should be conducted on the actual systems to make a correct and accurate bridge between thrust requirements and actual energy consumption. One of the main effects is thrust allocation and non-linear thruster efficiency. Since thrusters operate with different efficiencies at different thrust demands and the total thrust can be generated with different numbers of thrusters at individual speeds. This is not an obvious mapping. Lower maximum velocity and lower accelerations will yield a lower thrust and, thus, lower energy consumption. However, the actual minimum energy consumption will be systems and thus vessel-dependent.

Possible applications

Within Allseas, I see three possible applications for the current model. First, for the current PIT, let this algorithm compiled in C++ run in parallel with the OrcaFlex variants so that it can be scaled up with multiple pipe-laying projects and/or multiple scenarios at the same time. Secondly, it can be run as an onboard simulation tool with the proposed controller to improve the pipelay process. Additionally, this yields information on the actual loadings during a pipe project to either document for clients or to use as an information source for further improvements. The third application is to use this model in the HIL setup (Hardware in the loop), on which control software is tested before commissioning on actual vessels to give a more accurate model on which to test.

Bibliography

- [1] Allseas. *Linkedin post Allseas*. 2023. URL: <https://www.linkedin.com/feed/update/urn:li:activity:7104730968480964609/>.
- [2] Gullik A Jensen, Morten Breivik, and Thor I Fossen. “Offshore pipelay operations from a control perspective”. In: OMAE (2009).
- [3] Allseas Engineering. *Allseas activities*. URL: <https://allseas.com/activities/pipelines-and-subsea/pipeline-installation/>.
- [4] Gullik A. Jensen and Thor I. Fossen. “Mathematical models for model-based control in offshore pipelay operations”. In: *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE* 3.Omae (2009), pp. 269–276. doi: [10.1115/OMAEE2009-79372](https://doi.org/10.1115/OMAEE2009-79372).
- [5] Frans H De Vries. “Efficient modelling of offshore pipe-laying”. PhD thesis. University of Twente, 2020. ISBN: 9789036550307.
- [6] Orcina. *OrcaFlex*. Cumbria UK. URL: <https://www.orcina.com/orcaflex/>.
- [7] Offpipe. *Offpipe*. Houston. URL: <https://www.offpipe.com>.
- [8] DNV GL. “DNVGL-ST-F101 Rules for submarine pipeline systems”. In: October (2017).
- [9] Yingfei Zan et al. “Real-time dynamic analysis of J-laying”. In: *Chaos, Solitons and Fractals* 89 (2016), pp. 381–390. ISSN: 09600779. doi: [10.1016/j.chaos.2016.01.029](https://doi.org/10.1016/j.chaos.2016.01.029).
- [10] Orcina. *OrcaFlex documentation*. URL: <https://www.orcina.com/webhelp/OrcaFlex/Default.htm>.
- [11] Marcel Ellenbroek and Jurnan Schilder. “On the use of absolute interface coordinates in the floating frame of reference formulation for flexible multibody dynamics”. In: *Multibody Syst Dyn* (2017). ISSN: 1384-5640. doi: [10.1007/s11044-017-9606-3](https://doi.org/10.1007/s11044-017-9606-3). URL: <http://dx.doi.org/10.1007/s11044-017-9606-3>.
- [12] J. Schilder. “Flexible multibody dynamics, Superelements using absolute interface coordinates in the floating frame formulation”. PhD thesis. 2018. ISBN: 9789036546539.
- [13] J. Schilder, M. Ellenbroek, and A. De Boer. “Recursive thoughts on the simulation of the flexible multibody dynamics of slender offshore structures”. In: *IOP Conference Series: Materials Science and Engineering* 276.1 (2017). ISSN: 1757899X. doi: [10.1088/1757-899X/276/1/012029](https://doi.org/10.1088/1757-899X/276/1/012029).
- [14] Gullik Anthon Jensen. *Offshore Pipelaying Dynamics*. February. 2010. ISBN: 9788247118047.
- [15] Depeng Liu et al. “ALE-ANCF modeling of the lowering process of a J-lay pipeline coupled with dynamic positioning”. In: *Ocean Engineering* 269. January (2023), p. 113552. ISSN: 00298018. doi: [10.1016/j.oceaneng.2022.113552](https://doi.org/10.1016/j.oceaneng.2022.113552). URL: <https://doi.org/10.1016/j.oceaneng.2022.113552>.

- [16] Tarek Elsharhawy. “Advanced Pipe-Lay Tensioner Control System”. PhD thesis. California State Polytechnic University, Pomona, 2017.
- [17] Bao Chang Xu et al. “Research on constant tension control method of tensioner based on fuzzy adaptive algorithm”. In: *Proceedings - 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2012* Fskd (2012), pp. 356–360. doi: [10.1109/FSKD.2012.6233753](https://doi.org/10.1109/FSKD.2012.6233753).
- [18] Junliang Zhang et al. “Energy saving study on the tension system of a tensioner for a pipelaying vessel”. In: *Asia-Pacific Power and Energy Engineering Conference, APPEEC* (2009), pp. 0–2. ISSN: 21574839. doi: [10.1109/APPEEC.2009.4918334](https://doi.org/10.1109/APPEEC.2009.4918334).
- [19] Giuliana Mattiazzo, Stefano Mauro, and Paolo Serena Guinzio. “A tensioner simulator for use in a pipelaying design tool”. In: *Mechatronics* 19.8 (2009), pp. 1280–1285. ISSN: 09574158. doi: [10.1016/j.mechatronics.2009.08.004](https://doi.org/10.1016/j.mechatronics.2009.08.004). URL: <http://dx.doi.org/10.1016/j.mechatronics.2009.08.004>.
- [20] Shunfeng Gong et al. “Numerical modelling on dynamic behaviour of deepwater S-lay pipeline”. In: *Ocean Engineering* 88 (2014), pp. 393–408. ISSN: 00298018. doi: [10.1016/j.oceaneng.2014.07.016](https://doi.org/10.1016/j.oceaneng.2014.07.016).
- [21] Facheng Wang et al. “Development and sea trial of real-time offshore pipeline installation monitoring system”. In: *Ocean Engineering* 146.September (2017), pp. 468–476. ISSN: 00298018. doi: [10.1016/j.oceaneng.2017.09.016](https://doi.org/10.1016/j.oceaneng.2017.09.016). URL: <https://doi.org/10.1016/j.oceaneng.2017.09.016>.
- [22] Teddy Andreas Simanjuntak. “Significance of Active Tension Compensation During S-lay Pipeline Installation”. PhD thesis. TU Delft, 2017.
- [23] M. A. Crisfield. “Nonlinear finite element analysis of solids and structures Volume 2 Advanced Topics”. In: *European Journal of Mechanics - A/Solids* 17.6 (1998), p. 1044. ISSN: 09977538. doi: [10.1016/S0997-7538\(98\)90517-4](https://doi.org/10.1016/S0997-7538(98)90517-4).
- [24] T. Meinders and A.H. van den Boogaard. *An introduction to the finite element method*. 2014, pp. 40–42. doi: [10.7227/IJMEE.33.3.8](https://doi.org/10.7227/IJMEE.33.3.8).
- [25] Warren C Young and Richard G Budynas. *Roark’s Formulas for Stress and Strain*. 7th ed. 2006, p. 318. ISBN: 9780313060304.
- [26] Gullik A. Jensen et al. “A nonlinear PDE formulation for offshore vessel pipeline installation”. In: *Ocean Engineering* 37.4 (2010), pp. 365–377. ISSN: 00298018. doi: [10.1016/j.oceaneng.2009.12.009](https://doi.org/10.1016/j.oceaneng.2009.12.009).
- [27] Michael Plesha Robert Cook David Malkus. *Concepts and applications of finite element analysis*. 1989.
- [28] J.M.J. Journée and W.W. Massie. *Offshore Hydromechanics*. January. 2001. ISBN: 9781457717659. doi: [10.1109/ICSENS.2012.6411159](https://doi.org/10.1109/ICSENS.2012.6411159).