NTNU – Trondheim
Norwegian University of
Science and Technology

Fault Tolerant Control of Thruster-Assisted Position Mooring System

Zhengru Ren

Marine Technology
Submission date: June 2015
Supervisor: Roger Skjetne, IMT
Co-supervisor: Hans-Martin Heyn, IMT

Norwegian University of Science and Technology
Department of Marine Technology

NTNU Trondheim
Norwegian University of Science and Technology
Department of Marine Technology

Fault-Tolerant Control of Thruster-Assisted Position Mooring System

Zhengru Ren

June 2015

MASTER THESIS

Department of Marine Technology

Norwegian University of Science and Technology

Supervisor 1: Professor Roger Skjetne

Supervisor 2: PhD Candidate Hans-Martin Heyn

Preface

This report is the master thesis for graduation at Norwegian University of Science and Technology (NTNU), supervised by Professor Roger Skjetne. This report concludes the works done at the department of Marine Technology at NTNU in Trondheim.

The main research object is the fault-tolerant control of the thruster-assisted position mooring (TAPM) systems. Researches on the TAPM are so insufficient that it leaves me a great extend of freedom and potential to try various unconventional or crazy ideas. During the thesis, I felt the wholehearted happiness and passionateness. Positive feedbacks from the quickly grown fruit enhance my academic interest exponentially.

I had a few more ideas, which are potential to become journal papers and can be theoretically proved in Lyapunov direct method. However, I had to give up them after tangling the programs for a few of weeks. This is largely due to the superficial understanding toward the algorithm that I gained in the short period. I will try to find the flaws if I am lucky enough to continue this topic in my PhD stage.

During the thesis, I gained a better understanding toward what I have learnt, both the cybernetics and mechanics. Additionally, I learnt the most important academic writing rules from my supervisor's patient corrections. Frankly speaking, I used to deem that I was not bad at writing. But now I have seen my shortages clearly, and I will try to fix it as soon as possible.

To sum up, the thesis is enjoyable-I sharp my mind and feel extremely happy.

Trondheim, 2015-6-10

(Your signature)

Zhengru Ren

Summary

This thesis focuses on the fault-tolerant control algorithm of the thruster-assisted position mooring system (TAPM). TAPM is an energy-efficient and reliable stationkeeping method for deep water structures. System failures significantly endanger the marine control systems, and ultimately reduce the reliability and safety during operation and production.

First of all, mooring line breakage is difficult to detect without failure alarms. The statistic data shows that 50% of the floating production, storage and offloading (FPSO) in the North Sea cannot monitor line tension in real time, and 78% of them do not have line failure alarms in 2005. Therefore, line break detection is a crucial issue for the TAPM systems, especially those built decades ago.

Additionally, the global positioning system (GPS) signals may experience drifts during some extreme conditions, such as solar events and sudden ionospheric disturbances (SID). A typical position reference (posref) drift failure mode in a dynamic positioning (DP) system is that all GPS measurements starts to drift due to SID.

Furthermore, the tensioned-based localization accuracy is not sufficient when the anchors are unknown. Additionally, class society news shows that one anchor lost per 100 ships a year. To remove breaking anchor and chain is an expensive task.

Finally, the position of the upper moored vessel not only influences the mooring force arrangement, but also the riser angles at the top and bottom ends. Previous works applies finite element method (FEM) model and winch control to determine the vessel's equilibrium point. However, FEM model does not have analytical solution, and winch control is not applied during practical operation. Therefore, new setpoint chasing algorithm of TAPM is required to ascertain the safety of the operation and production.

Main contributions:

First of all, a TAPM toolbox was developed in Simulink. MSS TAPM is a expansion pack of MSS GNC and MSS Hydro toolbox. MSS TAPM is a Simulink lib for modelling TAPM system, especially the turret-based mooring system. In this lib, there are a group of different modules and examples, including two mooring line modules, a turret module, a TAPM module, an animation generation function, etc. Additionally, two papers were refined based on the works from the specialization project which are submitted to and accepted by the 10th IFAC

Conference on Manoeuvring and Control of Marine Craft (MCMC2015) after refinement. Another two papers will be submitted after further refinements and more strict verification.

Paper I presents a fault-tolerant control scheme based on an estimator-based supervisory control methodology to detect the line failure with only position measurements. After detecting a line break, a supervisor switches automatically a new controller into the feedback loop to keep the vessel within the safety region. Numerical simulations are conducted to verify the performance of the proposed technique, for a turret-based mooring system. This paper has been accepted by the MCMC2015.

Paper II presents a novel idea on a tension-based localization approach as a redundancy measure to handle the situation when the posref signals are not available or significant global navigation satellite system (GNSS) drifts occur, such as sudden ionospheric disturbances, for TAPM. The only information needed is the tension measurements from tension cells. This method can improve the redundancy and safety of offshore operation, by detecting and verifying posref failure modes. It can even take over the posref function if one no longer trust the main posref measurements. Based on a residual signal, a fault detection and estimation approach is introduced and verified through simulations. This paper has been accepted by the MCMC2015.

Paper III introduces an extended Kalman filter (EKF)-simultaneous localization and mapping (SLAM) algorithm to TAPM, which can locate the uncertain anchors, as well the moored vessel when GPS signal is not available. A sensor network scheme and a state-space model are built. EKF is applied to handle the uncertain anchor positions. Fairleads are considered to provide a more realistic and robust solution. The line-of-sight (LOS) range mapping from tension measurements is discussed. An additional application of this technique is to find the broken anchors with stored data offline for any vessel equipped with tension cells. This paper is an evolvement from Paper II, with more robust and applicable solution. It will be submitted after further refinement.

Paper IV proposes a new setpoint chasing algorithm for the TAPM based on deflection equation and convex optimization. The analytical solution of the riser end angles are deduced from classic structural mechanics. Comparing with the previous works, this method control the top tension and the desire position simultaneously to reduce the overall risk and cost during operation and production. The theory section of this paper, as well as the model verification in commercial FEM software, has been finished.

Acknowledgment

First of all, I would like to express my greatest and sincerest appreciation to my supervisor, Professor Roger Skjetne. Thank you for the encouragements; thanks you for the guidance at the key points; thank you for the replies from late nights; thank you for the correction again and again at non-office hours; thanks for giving me this productive topic; thank you for everything.

I am grateful to Dr. Vahid Hassani and Dr. Øivind Kåre Kjerstad for their helpful advices and feedbacks during preparing my conference papers. Also, they helped me aware my weaknesses in writing. I hope to say thank you to Professor Martin Enqvist from Linkøping University. During the interview talk with him, I knew a new realm, sensor network, which further promoted the idea of one paper.

I would like to express my sincere thank to Dr. Bo Zhao. He patiently helped me choose the topic when I was confused. I would also thank to Dr. Shaoji Fang for the explanation of his previous works. I would like to thank Dr. Xingwei Zhen, who solved my large amount of puzzles in mooring systems. I would like to thank to Professor Asgeir Johan Sørensen for his advices to my first conference paper. I would like to acknowledge Professor Ole Morten Aamo. Without his programs, I would suffer more resistance during the program. I would like to thanks to my co-supervisor, Hans-Martin Heyn.

I also want to thank my group project partners in the last year, Chunsheng Wang and Wenzhe Zhang. Without their helplessnesses in the projects, I can never be pushed to gain such a comprehensive understanding toward marine stationkeeping systems in the limited time.

Last but not least, I would like to thanks to my parents, Xudong Ren and Lihong Wang, who fund me and back up my academic dream all the time.

Zhengru Ren.

PROJECT DESCRIPTION SHEET

Name of the candidate:	Zhengru Ren
Field of study:	Marine control engineering
Thesis title (Norwegian):	Feiltolerant styring av thruster-assistert forankringssystem for offshore fartøy.
Thesis title (English):	Fault-Tolerant Control of Thruster-Assisted Position Mooring System.

Background

Stationkeeping operations for offshore vessels (drillrigs, drillships, construction and intervention vessels, PSVs, etc.) are essential for offshore field development and oil and gas production. There has been much attention in the research community on stationkeeping operations, especially by DP. In this project the focus is TAPM of turret-anchored offshore vessels. Stationkeeping operations by position mooring are challenging in varying environmental conditions, due to the nonlinear variations of mooring stiffness and drag on mooring lines due to oscillations of the surface vessel in waves. The challenges for the control system is therefore to determine how to position the vessel and damp vessel motions as much as possible with minimum use of the thrusters, how to minimize the risk of line break in any mooring line, how to online detect and handle the event of a line break failure, and how to redundantly estimate the positions of the vessel and/or anchors based on tension measurements. This includes use of all available, possibly redundant, measurements in the system (GNSS, IMUs, load cells, etc.), and possibly more advanced models of the mooring system to estimate the state of the system and predict risk of failures. The project aims to investigate several methods for estimation of important system parameters, fault-diagnosis, etc.

Work description

1. Perform a literature review to provide background and relevant references on:
   - Use of advanced offshore stationkeeping operations at different geographic loca-

tions worldwide.

- State-of-the-art industrial mooring solutions and TAPM control systems, including available sensors and typical limitations in state-of-the-art systems (e.g. systems with and without tension sensors).
- Moored stationkeeping operations in extreme environments (deep-water moorings, Arctic stationkeeping, extreme wave loads, etc.)
- Relevant research articles on TAPM control systems for offshore operations, and dynamic modeling of relevant mooring systems.

Write a list with abbreviations and definitions of terms and concepts, explaining relevant concepts related to position mooring control systems.

2. Develop and implement a TAPM vessel simulation model in Matlab/Simulink. This should include environmental loads from, wind, waves, and current, and a nonlinear (high-fidelity) mooring line model. The model should also include functionality for testing relevant failure modes in the TAPM control system (e.g. linebreak failures). Discuss the level of fidelity of the simulation model with respect to the intended use of verifying algorithms for fault-diagnosis, parameter estimation, and control laws.

3. Model and implement at least 2-3 GPS-based position reference systems (posref) for the vessel and a tension sensor for each mooring line in the TAPM simulation model. All these sensors should have a realistic noise characteristics added to the measurement.

4. Present a method for sensor fusion in the control system, to combine several posref measurements into a common measurement used by the control system for thruster-assist. Propose a nominal control law, including a setpoint chasing strategy, which ensures stationkeeping by the TAPM system. Perform simulations on the numerical simulation model for a set of relevant environmental conditions and discuss the responses.

5. For model-based control and estimation designs, derive and present relevant low-fidelity control design models used for your development of various algorithms. Simulate this model in parallel with the high-fidelity model in a few relevant test scenarios to show the conformity and differences between the models.

6. For a TAPM system without tension measurements, develop an algorithm for detecting linebreak failures based on available position/motion measurements. Simulate the system, verify the ability of your algorithm to detect linebreak failures, and discuss the performance of the system under realistic environmental loads and sensor noise.

7. For a TAPM system with tension measurements, develop algorithm(s) for estimating the positions of the vessel and/or the anchors and/or the anchor line touchdown points relative to the field zero point based only on the tension measurements. Show how the algorithm can be used as a redundant posref measurement to detect and handle GPS-based posref failure modes. Simulate the system, verify the ability of your algorithms to estimate the respective positions, and discuss the resulting performance under realistic environmental loads and sensor noise.

   Tentatively:

8. Consider the TAPM control problem as a generic maneuvering problem, where the geometric task is to control the position of the vessel to stay within a safe operating circle, and the dynamic task is to satisfy a setpoint chasing strategy within the circle. Propose a parametrization of the geometric manifold corresponding to the geometric task and an assignment law corresponding to the dynamic task. Use then the thrust vector as control input to derive a maneuvering control law that solves the geometric and dynamic tasks.

Guidelines The scope of work may prove to be larger than initially anticipated. By the approval from the supervisor, described topics may be deleted or reduced in extent without consequences with regard to grading.

The candidate shall present his personal contribution to the resolution of problems within the scope of work. Theories and conclusions should be based on mathematical derivations and logic reasoning identifying the various steps in the deduction.

The report shall be organized in a rational manner to give a clear exposition of results, assessments, and conclusions. The text should be brief and to the point, with a clear language. The report shall be written in English (preferably US) and contain the following elements: Abstract, acknowledgements, table of contents, main body, conclusions with recommendations for further work, list of symbols and acronyms, references, and optional appendices. All figures, tables, and equations shall be numerated. The original contribution of the candidate and material taken from other sources shall be clearly identified. Work from other sources shall be properly acknowledged using quotations and a Harvard citation style (e.g. natbib Latex package). The work is expected to be conducted in an honest and ethical manner, without any sort of plagiarism and misconduct. Such practice is taken very seriously by the university and will have consequences. NTNU can use the results freely in research and teaching by proper referencing, unless otherwise agreed upon.

The thesis shall be submitted with a printed and electronic copy to 1) the main supervisor and 2) the external examiner, each copy signed by the candidate. The final revised version of this thesis description must be included. The report must appear in a bound volume or a binder according to the NTNU standard template. Computer code, pictures, videos, data series, and a PDF version of the report shall be included electronically.

Start date: 15 January, 2015 **Start date:** As specified by the administration

Supervisor: Professor Roger Skjetne

Co-advisor(s): PhD candidate Hans-Martin Heyn

List of Theorems

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and motivation

With the depletion of onshore oil and gas resources and energy companies have brought about increasing interests in the exploration and exploitation of offshore and deeper resources. DP systems are used in offshore drilling applications to ensure stationkeeping of the drilling vessel. A TAPM system is another solution in which the mooring system decreases the level of thrust needed. This allows the thrusters to improve the positionkeeping performance.

The high initial outlay makes DP become a less cost-effective stationkeeping method within 1500 depth water. For a passive mooring system, the capital expenses (CapEx) increases dramatically with the increase of the water depth (see Figure 1.1). While a TAPM system, as a combination of them, only requires the vessel to be kept in an accepted region by the mooring lines in normal seas. The key difference between DP and TAPM is that the full DP vessel/unit will lose its position when the environmental loads exceed the extreme capacity of the thruster system. For a DP system, both the position and heading should be kept at the desired values by the thrusters only. In a normal sea, a TAPM system will keep the vessel in a reasonable region by the mooring lines, while the thrusters support the system in keeping the optimal heading and additional surge-sway damping. Thruster-assisted system only controls heading and provides additional damping in calm seas and normal seas. Additive position control becomes necessary only in high seas or extreme conditions. While a full DP unit needs to have continuous control in both surge and heading Aalbers et al. (1995). Such primary difference leaves an attractive characteristic of TAPM for researchers, that is TAPM

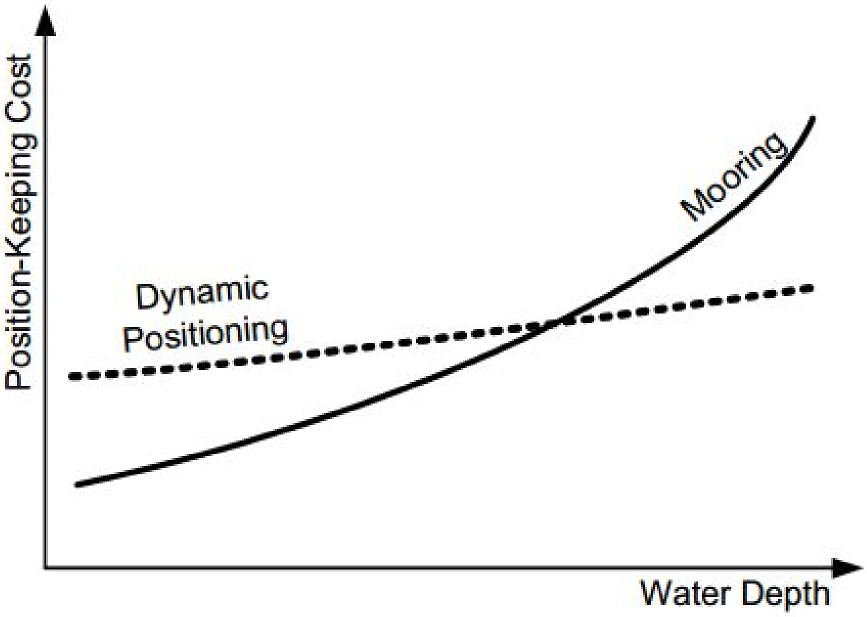


Figure 1.1: Cost curves of position-keeping

largely reduces the operational expenses (OpEx) in a long-term perspective. However, TAP-M systems, meanwhile have their shortages. The most prominent one is the mooring lines significantly increase the construction, maintenance and operation costs.

The first widely acknowledged DP vessel was Eureka built in 1961. The amount of DP vessels grew from approximately 65 to 150 in the first half of the 1980s. A rapid growth happened in the following 20 years due to the increasing concern on environmental sustainability and safety production. With the largest share in the global DP market, Kongsberg almost " delivered one DP system every day in 2013", and the total amount DP installations from Kongsberg has passed 3000 (Kongsberg, 2013). The data also shows the tremendous potential of the application of the TAPM systems in the marine market.

1.2 Research problems

1.2.1 Mooring line failure

Mooring line failures can lead to loss of position-keeping capability of the floating structure. Hence, it can endanger human lives, equipments and the environment. Mooring line failures can happen at either the upper or the bottom end, and they may not be found within several months. A series of guidance documents and standards about position mooring system were published from the main associations and class societies, such as API (2005), ABS (2014), and DNV (2010). All these guidelines and standards require a redundancy to line break during the mooring design stage. Many industrial products have been invented to provide real-time monitoring of the mooring and riser systems. Pulse Structural Monitoring developed the world's first low cost motion logger, the INTEGRIpod™, in 1998 (Gauthier et al., 2014). With integrated data loggers for measuring the movement of subsea structures, the system will alert when the tension in a mooring line exceeds the preset threshold. Similarly, Inter-M Pulse™ is suitable for moored FPSO and mobile offshore drilling units (MODU) to provide full history data with acoustic signals (Elman et al., 2013). However, according to the statistical data from Brown et al. (2005), 50% of the FPSO in the North Sea cannot monitor line tension in real time, and 78% do not have line failure alarms.

1.2.2 GNSS failure

The GNSS provides the earth-fixed position information to surface-based motion systems. GPS, being the most widely used, has been a necessary part to navigation systems of modern marine systems in recent decades. However, the GPS signals may experience drifts during some extreme conditions, such as solar events and SID. SID is a phenomenon with sudden increase of electron density in the ionosphere caused by solar flares, earthquake, storm, or tsunami. It results in a sudden decrease of the upper medium frequency and lower high frequency components in radio-waves (Afraimovich et al., 2000). Normally, SID happens simultaneously with ionospheric storms. These phenomena can last for 1-3 days, even 10 days, significantly degrading the reliability of marine control systems (Tsugawa et al., 2011).

A typical posref drift failure mode in a DP system is that all GPS measurements start to drift due to SID. The Hydroacoustic position reference (HPR) system does not drift, but due to the superior signal quality of the GPS signals over the HPR signals, the DP control system chooses to believe in the GPS signals and automatically disables the HPR measurement, thus making the situation worse with a resulting DP system drive-off.

Considering the long-term duration of TAPM stationkeeping operations, the probability of experiencing such drift events is high. Doherty et al. (2004) reported that three very large sunspot clusters happened in October-November 2003 which caused strong magnetic storms. A Large amount of satellites failed jeopardize the safety of the operations.

1.2.3 Lost anchor and chain

News has reported the main class societies experience one anchor lost per 100 ships a year (Nord, 2011). The risk of losing anchor and chain is tremendous when considering the service life in more than two decades. Increasing number of port authorities require removing the lost anchors and chains from the sea bed. Therefore, techniques which can quickly locate and remove the lost anchors are considerable.

1.2.4 Setpoint chasing algorithm

In classic structural mechanics, the deflection equation of a beam is uniquely specified based on the beam geometrical and material feathers, external loads, and the axial loads. The deflection equation theory has been verified in more than two centuries' engineering applications. Besides, deflection superposition principle is a useful simplified method to calculate the deflection with complex loads.

Especially in deep water, frequently happened vortex-induced vibration (VIV) results in structural-damaging fatigue to risers. VIV is the motion caused by the pass fluid. When the VIV frequency is close to the riser's natural frequency, lock-in phenomenon happens, that is the motion induced on body interacting with an external fluid flow (Faltinsen, 1993).

Previous setpoint algorithms always focus on a specific point of the whole system. For example, Nguyen et al. (2011) applies a passive controller together with the reliability index to optimize the setpoint chasing scheme. However, the analytic solution is a result based on quadratic objective function. For more complex objective function, it will be difficult to obtain a solution by hand. Additionally, winches control are always expensive or not attainable in practical operation.

1.3 Structure of the thesis

The thesis is divide into two main parts. The main body reviews the basic definition, state-of-art technical advancements, mathematical model, and theoretic background of the TAPM. It is organized as follows

Chapter 1: A brief introduction to the most state-of-art offshore stationkeeping methods is summarized, as well as a comparison among them.

Chapter 2: A brief literature review reviews of prevalent researches on TAPM.

Chapter 3: The kinematics and kinetics models are introduced. The mathematical description of TAPM models are covered, including both the process plant model and the control plant model. Different mooring models, include the linear model, catenary model and FEM are introduced and compared. A group of various controllers, observers, thrust allocation, setpoint chasing algorithm and reference system are summarized.

Chapter 4: Schemes to merge the signal from multiple sensors are summarized. Basic sensor fusion theory are presented.

Chapter 5: Fault-tolerant control theory is briefly concluded. Basic concepts, categories, control structure, and control schemes are covered.

Chapter 6: Two mooring line breakage detection schemes are presented. The first scheme depends on supervisory control theory which can detect the line failure the residual errors. The second solution is based on the tension-based localization algorithm proposed in Chapter 7.

Chapter 7: A complete model is built to locate the moored vessel of only tension measurements, based on sensor network and SLAM. Range measurement based on tension-range mapping is discussed. The fairleads are considered. Another two simplified models are illustrated.

Chapter 8: A setpoint chasing algorithm is proposed. The optimal setpoint is obtained through mulch-objective function optimization.

Chapter 6: This chapter concludes the thesis and propose recommendations for future works.

The appendix includes the user manual of the MSS TAPM, the MATLAB codes and, the Simulink models which are used for simulations in the papers.

Chapter 2

Literature Review

This chapter reviewed the most state-of-art commercial software, industrial solutions, and scientific researches on the TAPM systems.

2.1 Industrial Solutions

2.1.1 Design guidance and standards from class societies

A series of guidance documents and standards about mooring system from the main associations and class societies, such as, API (2005), ABS (2014) and DNV (2010). These documents provide detailed requirements to mooring systems or TAPM systems.

2.1.2 Software for mooring system design

There are some existed mooring design and analysis software. For example, Mooring Design and Dynamics (MDD) is a MATLAB package developed by University of Victoria to assist the design and evaluation the loads of single point mooring (Dewey, 1999). The newest version is Version 2.2. MIMOSA (Lie et al., 2002) and further MOOROPT-2 (Fylling, 2005) from SINTEF, GMOOR32 from Global Maritime (Morandi and Shi, 2011), and MOSES from Ultramarine (2013) are commercial software which can be applied on mooring design and analysis. Though some of the softwares are applicable in time domain analysis, They are difficult to utilize with a combination of control theories.

2.2 State-of-art stationkeeping solutions

Mooring systems safeguard marine structures, such as riser systems and drilling units, through providing excursion limits. There are seven commonly applied types of the mooring system in offshore industry, e.g. catenary, taut leg, semi-taut, spread, single point and dynamic positioning. This section will review several mooring solutions.

2.2.1 Catenary mooring system

The catenary mooring system is the most commonly applied mooring solution, especially in shallow water. With a part lying on the seabed, the horizontal resorting forces come from a component of the weight of the mooring lines in the water. Therefore, the construction cost and vertical loads upsurges in according to the water depth in an economical perspective. A mooring system consists chains, wire ropes, synthetic rope, connecting hardware, clump weights, buoys, winches, fairleads and anchors (API, 2005).

2.2.2 Taut leg system

The taut leg system is characterized as pre-tensioned mooring lines. The ropes, typically made of polyester ropes, normally have an angle with the seafloor between 30 and 45 degrees. The main advantages are (1) lower material cost, (2) easier to control and smaller tension due to more linear stiffness, (3) better load sharing between adjacent mooring lines. However, designers have to ensure that the mooring lines must have sufficiently large elasticity in case of overloading. To sum up, taut leg mooring system, comparing with the catenary mooring system, is a better solution for deep water. Figure 2.1 shows the difference between catenary mooring system and taut leg system.

2.2.3 Semi-taut system

The semi-taut system, as shown in Figure 2.2, is a kind of hybrid system combining the catenary and taut mooring systems for deepwater application. It supplies a material-saving solution with shorter mooring lines and less seafloor space.

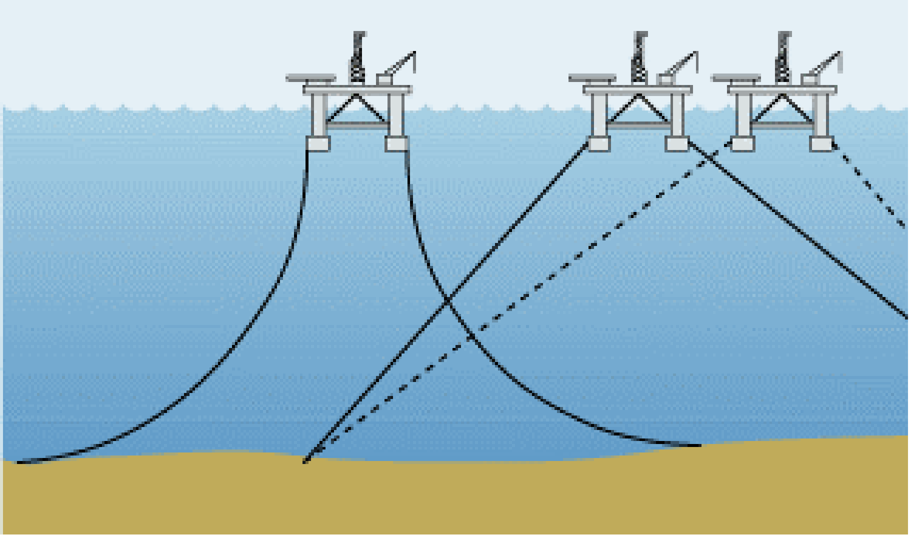


Figure 2.1: Catenary mooring (left) and Taut mooring (right) (Hycalin.nl, nd)

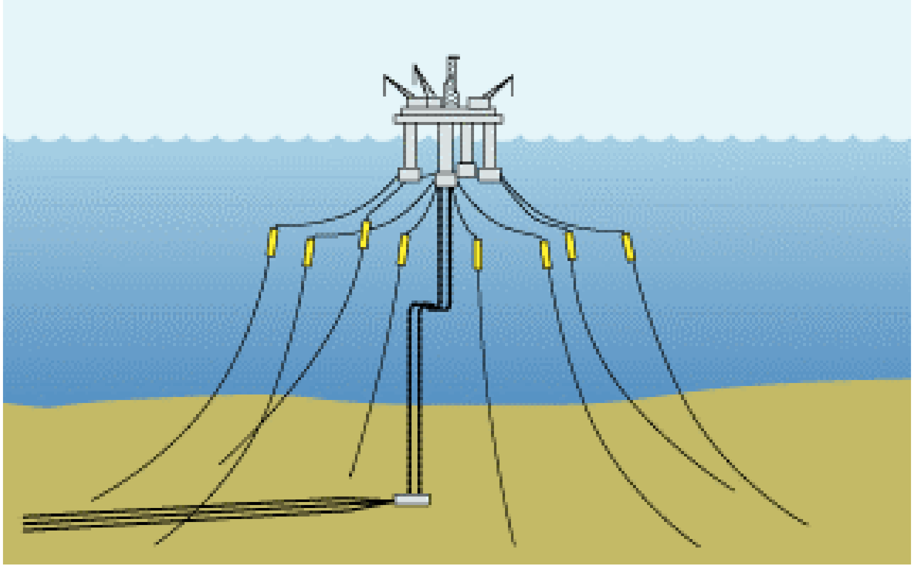


Figure 2.2: Semi-taut mooring system (Hycalin.nl, nd)

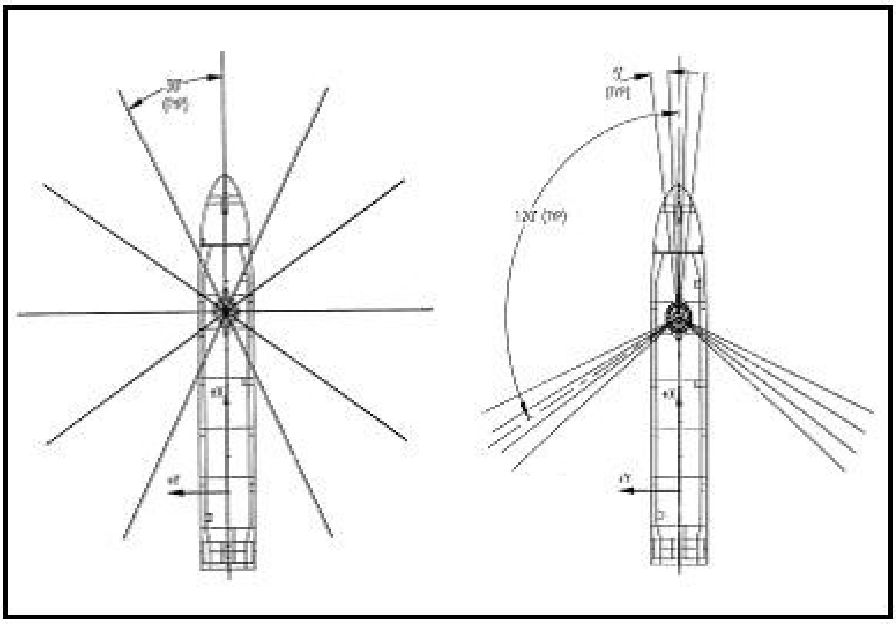


Figure 2.3: Equally spread mooring (left) and grouped spread mooring (right) (dredgingengineering.com, nd)

2.2.4 Spread mooring system

A spread mooring system is a series of mooring lines attached to the bow and stern of the vessel at a fixed heading. Normally, the arrangement is symmetric. Equally spread mooring pattern and grouped spread mooring are the two main arrangement methods. See Figure 2.3. The grouped spread mooring provides high redundancy. This kind of solution is exactly suitable for the vessels which are insensitive to environmental distribution and loads. The bow and the stern mooring groups primarily provide restoring forces and system stiffness, respectively.

2.2.5 Single point mooring system

For a single point mooring system, all the mooring lines are connected to one internal or external supported turret column. In addition, the system consists of buoys, mooring lines, anchor elements, a product transfer system and other components. This solution is normally applied on ships. It allows the ship rotating 360 degrees based on the external environment. The turret arrangement is mainly influenced by the bow sea condition, which minimizes the resistance or environmental disturbances. Figure 2.4 and Figure 2.5 are the internal and external turret mooring system, respectively.

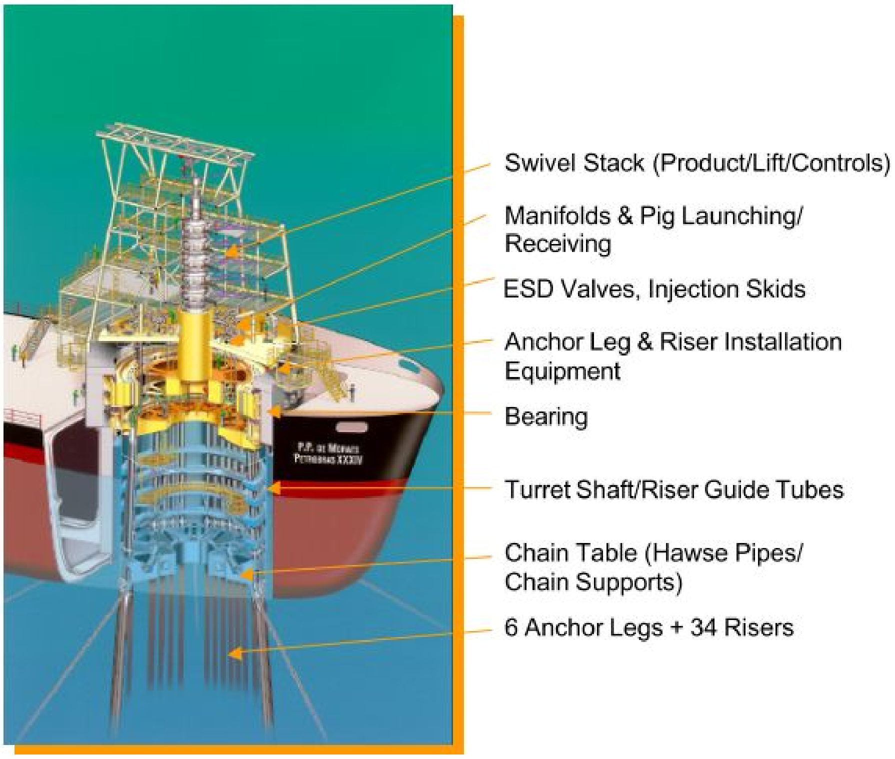

Swivel Stack (Product/Lift/Controls)
Manifolds & Pig Launching/ Receiving
ESD Valves, Injection Skids
Anchor Leg & Riser Installation Equipment
Bearing
Turret Shaft/Riser Guide Tubes
Chain Table (Hawse Pipes/ Chain Supports)
6 Anchor Legs + 34 Risers
P.P. de Moraes
Petrobras XXXIV

Figure 2.4: The P-34 Internal Turret Mooring System

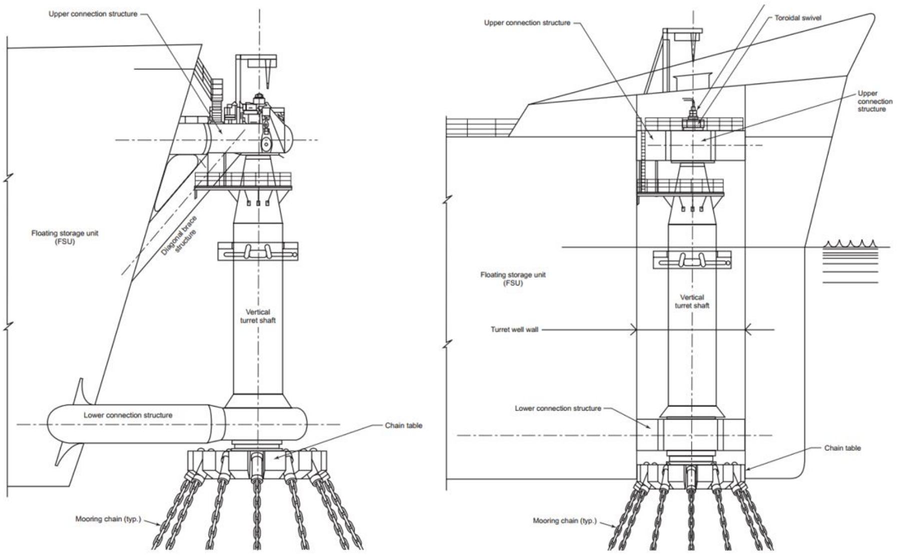

Upper connection structure
Floating storage unit (FSU)
Diagonal brace structure
Vertical turret shaft
Lower connection structure
Chain table
Mooring chain (typ.)
Upper connection structure
Toroidal swivel
Upper connection structure
Floating storage unit (FSU)
Vertical turret shaft
Turret well wall
Lower connection structure
Chain table
Mooring chain (typ.)

Figure 2.5: Turret external (left) and internal (right) turret mooring arrangement

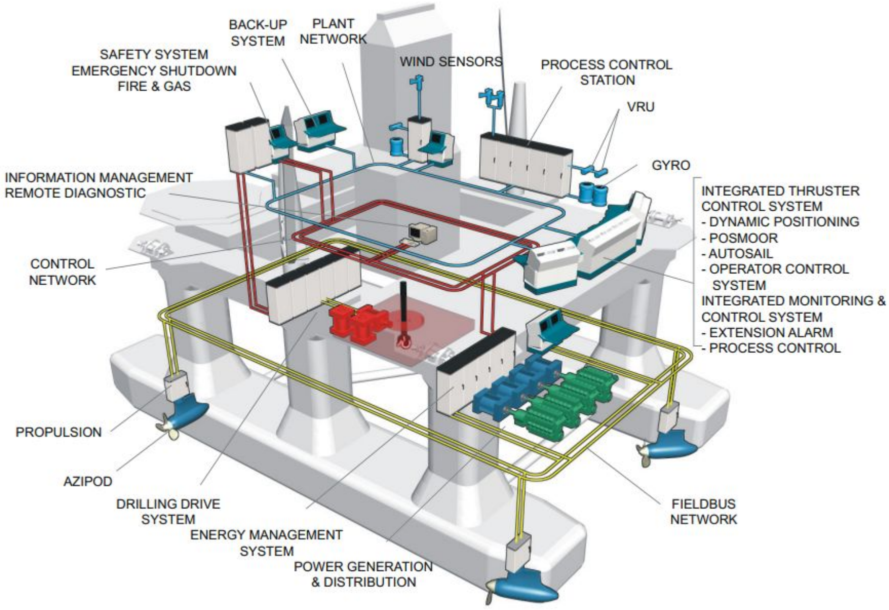


Figure 2.6: Dynamic positioning system (Sørensen, 2012)

2.2.6 Dynamic positioning system

The dynamic positioning system (see Figure 2.6) maintains the position and heading with the assistance of thrusters and propellers. Unlike other passive methods, it need no mooring lines, but more fuel. Therefore, this is a more flexible solution for short-term production.

2.2.7 Thruster-assisted position mooring system

Thruster-assisted position mooring system, or called as position mooring, is a combination of mooring and thrusters. Thrusters are used to control the heading in normal conditions and reduce mooring loads in extreme environment. The thruster assistance has a different role to that of DP system. For a TAPM system, thruster assistance mainly provides damping in horizontal motions, including the surge, sway and yaw motion. Additionally, thruster assistance is used to avoid line tension exceed thresholds in high sea states.

2.2.8 Application of different stationkeeping solutions

The choice of the stationkeeping solution is a trade-off between oil companies and the designer, which primarily determined by funding and safety requirements. Oil companies would likely to apply the cheapest solution which can ensure safety standard at the same time. For the designer, the primary design considerations for mooring system are design loads, design life, operation and maintenance, as well as riser and subsea equipment. Mooring systems are categorized as permanents and mobile mooring systems. Permanent mooring is commonly used for marine structures with relative longer design lives of decades, for example, the floating production system (FPS). While mobile mooring systems are designed for shorter periods of working task, such as MODU and service vessels. DP is less cost-effective for a long-term operation, and mooring system shows low economical effectiveness for units with high mobility.

Additionally, each stationkeeping method is suitable for some specific marine units and working conditions. Single point mooring systems are usually used by ship-shaped vessels, such as FPSO. The spreading mooring system is mostly adeptly by semi-submersibles and spars. Dynamic position has a wider scope of application. Tension leg is suitable for tension leg platform (TLP). DP can be applied as the sole source of station-keeping or the assistance of catenary mooring system in a range of marine structures, including both ship shaped vessels, drillships and semi-submersibles.

Furthermore, the waterdepth is a key factor to decide stationkeeping method. Normally, mooring lines, and anchors and connectors are key components to a mooring system. Being the main part, the material of the mooring lines generally have three options - synthetic fiber rope, wire and chain, or, a combination of them can be applied. The selection is mainly based on the waterdepth. For example, chains are applied in shallow water, 0-100 meter waterdepth. Since the steel wire ropes are lighter and higher elasticity than chains, they are used in deep water, say 300-2000 meters. For the ultra-deep water, which means deeper than 2000 meters, either chain-synthetic fiber rope, or chain-wire rope-synthetic fiber rope combination is a choice.

2.3 Stationkeeping solutions in extreme environments

2.3.1 Deepwater mooring

In deep water, anchor mooring is still a useful stationkeeping method(Ehlers et al., 2004; Colliat et al., 2002). Light-weighted materials becomes more popular due to the heavy chains. Polyester Rope is another useful solution with lighter weight, superb fatigure performance, smaller footprint, shorter lines, and littler vessel offset (Price et al., 2003; Davies et al., 2002).

However, polyester moorings have following shortages. Characterization of polyester rope stiffness is difficult to design. Design guidance, for example, DNV-OS-E303 and DNV-RP-E305, can be applied to handle this problem (DNV, 2010).

2.3.2 High seas

Many studies on high seas are accompanied to satisfied the operational needs in the extreme environments. Nguyen and Sørensen (2009) designs a hybrid controller to control the DP in different sea states. An improved hybrid controller is shown in Brodtkorb et al. (2014). N-guyen and Sorensen (2009) presents optimal desired setpoints to follow. An setpoint can be generated offline and ensure that the tension below a safety value in risky conditions. Different controllers are compared in Hassani et al. (2012a). Model uncertainties and superposition are considered in Lin et al. (2013). The most important failure modes for TAPM are a loss of a sub-sea mooring line buoyancy element and line breakage (Fang and Blanke, 2011). Additional failure modes include sensor faults, such as GPS drift (Ren et al., 2015b), and power system faults, for instance, fuel system failures, mechanical failures, and control system failures (May, 2003; Radan, 2008). Structural reliability and fault-tolerant control are applied in a proposed position mooring system (Leira et al., 2004; Berntsen et al., 2008; Wang et al., 2014; Fang et al., 2015; Ren et al., 2015a). Application of fault monitoring and fault recovery control techniques to position-moored vessels is reported in (Fang and Blanke, 2011).

2.3.3 Arctic environment

Arctic exploration has been aroused increasing concerns in recent years, especially the gas and oil resources (Fissel et al., 2008), and new ship routes (Reeves et al., 2014). The attraction of the Arctic area is growing due to technological advancement and global warming (Assessment, 2004). The special climatic and geographic conditions in the Arctic area demand a higher level risk management, more intelligent automation, and stricter system reliability (Bonnemaire et al., 2007). Ice conditions can be categorized as first-year ice, multi-year ice, and icebergs (Hamilton et al., 2011). Ice loads are hard to estimate due to the complex ice form and ice interaction, and it normally received through Froude scaling and empirical correction depended on model-based experiment results (Comfort et al., 1999; Palmer and Dempsey, 2009). Due to the icy climate, an Arctic stationkeeping system has a variety of d-ifferences, such as sub-surface ice transport, disconnection and reconnection (Gudmestad et al., 2009; Van Der Nat et al., 2012), icebreaker assistance (Hamilton et al., 2011; Gudmestad et al., 2009), ice drift (Uttal et al., 2002; Jorgensen and Skjetne, 2012), and vertical ice load accumulation (Bonnemaire et al., 2007). Being a crucial issue of Stationkeeping operation in the Arctic area, ice management is an efficient strategy for load reduction in ice and will be most likely to be a part of any operation philosophy of an Arctic floating vessel in severe conditions (Haugen et al., 2011).

2.4 System modelling

2.4.1 DP models

The basic mathematical models can be found in Fossen (2011) and Sørensen (2012). The models can be categorized into process plant model (PPM) and control plant model (CPM). PPM is a comprehensive model of the actual physical process. CPM is a simplified model for the control system. It is used to design the controller and observer (Sørensen, 2012).

Researches on the TAPM system focus on the turret-based mooring system, due to its simpler dynamic characteristics. Six degrees of freedom (DOF) model is used as PPM, CPM can be simplified to 3DOF due to DP's or TAPM's slow velocity.

2.4.2 Mooring line models

To simulate the mooring system, normally there are three models-the linear model, the catenary equation and the FEM model. Linear model are used in Sorensen et al. (1999). Catenary equation is used to mathematically describe the U-like curves of the ideal hanging chains or cables under its own weight Faltinsen (1993). It is a static analysis method to design mooring systems. The form of a catenary curve is only determined by the forces and the position of the supported end, as well as the weight and length of the chain. For a mooring lines, the seafloor part has a constant value of z. Given the length of a suspended part, the end points could be found. The vertical and horizontal forces of the end point are influenced by the length of the suspended part. Divide the length of the mooring line into a series of segments, from the anchor to the turret. (Aamo and Fossen, 2000) employs the finite element method (FEM) to simulate cables suspended in water. This model is derived from the partial differential equations (PDE) of cable dynamics. The existence and uniqueness are also proved.

Since it is no longer a quasi-steady approach, FEM model is valuable to conduct full dynamic analysis of the turret-based mooring oil production ships.

The dynamics of the mooring system is strongly governed by the design parameters, for example, materials, pretension, length of mooring lines, positions of the anchors and the fairleads (Garza-Rios and Bernitsas, 2001). Studies show that mooring lines tend to lose its stability with the increasing of waterdepth (Garza-Rios and Bernitsas, 1999).

2.4.3 Riser models

Risers are categorized as rigid risers and flexible risers. A rigid top tensioned riser (TTR) is modelled as a simple supported Euler–Bernoulli beam (How et al., 2009). While flexible ones can be also modelled as catenary equation (Niedzwecki and Liagre, 2003). Other riser models include the 2D FEM model (Rustad et al., 2008) and the distributed-parameter model (Niedzwecki and Liagre, 2003). The availability of the FEM model is discussed and verified in Rustad (2007). It is used to control riser angles, payout, and vibration Leira et al. (2004); Nguyen et al. (2010). Previous researches assume the risers do not provide restoring forces to the upper vessel.

2.5 Observers

Observers are applied to filter the measurement noise, eliminate the wave-induced wear and tear effects, and estimate the unmeasured states. Filters are characterized into Bayesian filters and non-Bayesian filters (Gustafsson, 2010). Bayesian filters, for example, are Kalman filter (KF) (Kalman, 1960), EKF (Balchen et al., 1980), unscented Kalman filter (UKF) (Wan and Van Der Merwe, 2000), particular filter (Zhao et al., 2012), and decentralized Kalman filter (Brown et al., 1997).

The nonlinear passive observer (NPO) is the most widely applied non-Bayesian filter, which is proved in direct Lyapunov method (Fossen and Strand, 1999; Strand and Fossen, 1999). NPO is implemented and verified to be an observer which work loads on tuning is much smaller than EKF (Fossen and Strand, 1999; Strand and Fossen, 1999). Hassani et al. (2012b,a) compares the performance of NPO and the KF in different sea states. The results show that KF is more helpful to estimate the LF motion. While the combination of proportional-integral-derivative (PID) controllers and passive observer is the simplest choice.

2.6 Controllers

2.6.1 Linear controllers

Early DP systems are implemented using PID controllers with notch filters in cascade with lowpass filters (Fossen, 2011). A variety of improved PID controllers in different vessel operation condition (VOC) (Nguyen and Sørensen, 2009; Nguyen et al., 2010). Traditional PID controllers are simpler and easier to use. As PID controller is a kind of fuzzy controllers, it is well suitable for low-cost implementations, which means less accuracy sensors and slow onboard controller.

A series of model-based controllers are designed. linear-quadratic-Gaussian (LQG) controller is simple to implement and has no requirement of environmental data, with a faster computational speed (Di Masi et al., 1986). But it is time-consuming to tune when choosing reasonable weight matrix.

2.6.2 Nonlinear controllers

As the development of sensor technology and stricter control requirements, more advanced nonlinear controllers are needed to develop, for example, backstepping technology (Fossen and Grovlen, 1998), slide mode controller, $H\infty$ controller (Tannuri et al., 2001). Backstepping is a useful nonlinear robust control design method. Nevertheless, the robust control typically introduces large oscillation when the system uncertainty is in a large range. Adaptive control can estimate the uncertain term in real time and in case the controller ”use excessive actions to regulate the process” (Lavretsky and Wise, 2012). Sliding mode control maintains the stability with reasonable modelling error, as well the varying of environmental disturbances. Modern computer advancement has solved the key drawback-large computational capacity required. Though nonlinear controllers have many advantages, their performances are largely dependent on the accuracy of system identity and modelling. Additionally, they are sensitive to noises and environmental disturbances which may finally cause wear and tear effects to the thrusters (Skjetne et al., 2005).

Normally the mooring forces depend on the position of the surface units. Aamo and Fossen (1999) demonstrates the potential for reducing the fuel consumption by controlling the line tension with a passivity observer and the FEM model. However, the fast disturbance will cause wear and tear to the thrusters. Chen (2013) applies neural network approximation and backstepping to control the mooring system. Structural reliability criterion is a combination of control theory and structure is conducted (Fang et al., 2013). Nonlinear passive weather optimal position control (WOPC) is presented and empirically verified in Fossen and Strand (2001).

2.6.3 Fault-Tolerant Control

(Fang and Blanke, 2011; Fang et al., 2011, 2013, 2015) conduct a series of researches on structural reliability criterion and fault-tolerant control. A combination of control theory and structure are conducted firstly for position mooring system (Fang et al., 2011).

Computational simulations verify the performance of the fault-tolerance control for both line breakage and the loss of mooring line buoyancy. The system will move to another equivalent point based on a new fault-accommodating position algorithm after one mooring line breaks. Comparing with Barth Berntsen et al. (2008), the new the optimal position algorithm can be applied to more than one mooring line.

Fang et al. (2013) produces an optimal setpoint chasing algorithm for a TAPM system based on a structural reliability criterion. This new algorithm aims to prevent mooring line failures due to fatigue based on the measured mooring tension data directly. From computational simulation results, the position mooring reliability (PMR) controller has better performance than conventional PID controller in a changing current disturbance. This algorithm could also implement in industry through both automatic detection and operator-assisted decision.

2.6.4 Riser control

Vibration control has been the widely studied (Ge et al., 2010; Dareing and Huang, 1979; Rho et al., 2007; How et al., 2009; Trim et al., 2005). Another research issue is to control the payouts among a group of risers through controlling the top tension at the end of the riser for a TLP Rustad et al. (2008). It can reduce the risk of collision among the risers. Finally, the control of risers' end angles is essential to safe operation. Nguyen et al. (2011) applies the FEM model to get the relation between the riser angle and the vessel position for a drillship. Aamo and Fossen (1999) investigates the potential a control scheme by controlling the tension.

2.7 Reference system

2.7.1 Setpoint chasing algorithm

One important control objective of TAPM is the heading control. Nguyen et al. (2007) gives an optimal desired setpoint to follow. This algorithm is based on an assumption that the tension of the mooring line has linear relations with the vessel's position. The slope is the stiffness of the most loaded mooring lines. An setpoint then can be generated offline to the reference system by ensure the tension of the most loaded mooring line is smaller than a safety factor times the critical tension value.

Based on the setpoint algorithm, Nguyen and Sorensen (2009) introduces a switch control which can detect the sea states from the WF motion estimated by the observer. A bank of controllers with different PID parameters is applied for various VOC. The supervisory can automatically choose a reasonable controller for different VOCs. In different sea states, there exists a bank of specific controllers and setpoint algorithms. Stability analysis and an experiment show its industrial advantages in cost saving and safety maintenance.

2.7.2 Structural reliability criterion

In practice, position moored MODU operates with risers or other drill units. It is of great important to control the risers in a combination with the TAPM controlling. The research tendency about TAPM is toward a combination of cybernetics and structure. Leira et al. (2004) applies reliability methods to control riser angles and a DP system by employing a reliability-index. The simplified reliability-index capture both static and dynamic response components. Further researches are conducted on this topic (Barth Berntsen et al., 2008; Berntsen et al., 2008; Nguyen et al., 2010; Fang et al., 2011, 2015).

2.8 Research methods

2.8.1 Numerical simulation

Numerical simulation is a cheaper and much faster research method at the beginning stage. Investigations on simulations are reported in Barth Berntsen et al. (2008), Fang and Blanke (2011), and Chen (2013). MATLAB$^{®}$ and Simulink$^{®}$ are the main simulation environments.

Most studies are conducted based on the Marine System Simulator (MSS) toolbox, which is developed by NTNU (Perez et al., 2006). The mooring model can be the linear model, the catenary equations or the FEM model (Mavrakos et al., 1996; Dewey, 1999; Aamo and Fossen, 2000; Jeon et al., 2013). FEM model is proved to be useful during simulation (Nguyen and Sorensen, 2009). Floating sea ice environments can be simulated in simulators, such as Metrikin (2014).

2.8.2 Model experimental

Model experiments can provide more reliable results and help find the flaws in the simulations before practical applications. Stationkeeping experiments can be done in the Marine Cybernetics Laboratory (MCLab). The experimental set-up are shown in Berntsen et al. (2009) and Nguyen et al. (2010). A pulley system is used to simulate the mean wind and current loads. Experiments with buoys connected to the mooring lines can be found in Blanke et al. (2012). Experiences in ice environment can be conducted at Hamburgische Schiffbau-Versuchsanstalt (HSVA) (Kjerstad and Skjetne, 2014).

Chapter 3

Mathematical Modeling

This chapter addresses the main process plant model and control plant model for the TAPM system. Additionally, the observer, controller and reference system are introduced.

3.1 Kinematics

3.1.1 The Earth-fixed reference frame

The Earth-fixed reference frame is equal to the North-East-Down (NED) frame. This reference frame is tangential to the Earth's surface with the z-axis pointing downward normal to the Earth's surface, the x-axis pointing towards the true North and the y-axis pointing toward the true East.

The NED-frame is usually expressed in terms of $\boldsymbol{n} = (x_n, y_n, z_n)$with origin O_n, but it can also be expressed in terms of latitude and longitude. The position vector $\boldsymbol{\eta}$ is defined in NED-coordinates.

3.1.2 The body-fixed reference frame

The body-fixed reference frame is a coordinate system moving with the body. The reference system is defined as $\boldsymbol{b} = (x_b, y_b, z_b)$ with origin O_b, which is usually located in the horizontal midpoint in the waterline. The velocity vector $\boldsymbol{\nu}$ is defined in body coordinates. The motions of the vessel in the body reference frame are defined in Figure 3.1.

The Earth-fixed position $\boldsymbol{\eta} = [\boldsymbol{\eta_1}, \boldsymbol{\eta_1}]^\top$, $\boldsymbol{\eta_1} = [x,y,z]^\top$ and the orientation $\boldsymbol{\eta_2} = [\phi,\theta,\psi]^\top$ have a following relation with the body-fixed translation $\boldsymbol{\nu_1} = [u,v,w\ ]^\top$ and rotation velocity $\boldsymbol{\nu_2} =$

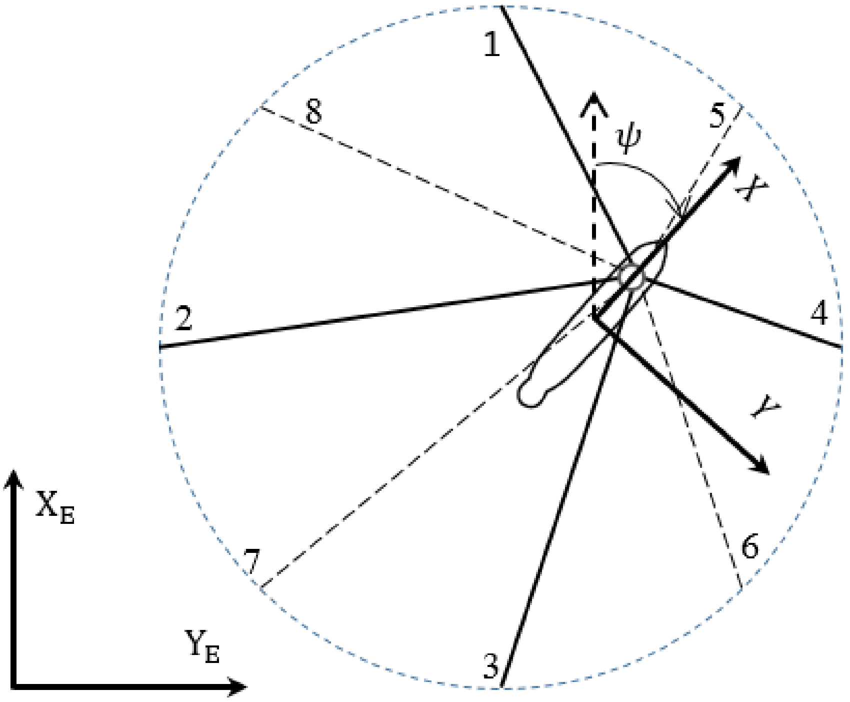


Figure 3.1: Catenary equation

$[p,q,r]^\top$, such that

$$\begin{bmatrix} \dot{\boldsymbol{\eta}}_1 \\ \dot{\boldsymbol{\eta}}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_1(\boldsymbol{\eta}) & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \boldsymbol{J}_2(\boldsymbol{\eta}) \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{v}}_1 \\ \dot{\boldsymbol{v}}_2 \end{bmatrix} = \boldsymbol{J}(\boldsymbol{\eta}_2)\boldsymbol{v}. \tag{3.1}$$

3.1.3 Current reference frame

The current coordinate $\boldsymbol{c} = (x_c, y_c, z_c)$ is defined that the x direction is same as the current direction, and z axis points downward with an origin overlapped at $[0,0,0]^\top$.

3.2 Low-frequency and wave-frequency motion

In calm seas and normal seas, wave-frequency (WF) motion introduces extra control inputs to the controller, and then the thrusters suffers wear and tear effects. Wear and tear effects impact the DP's performance, lessen the system reliability, and finally gain the maintain costs. However, WF is a considered as a part of the control system in extreme seas where the amplitude and the period of WF motion are remarkably large. The definition of the sea states is shown in Table 3.1. This classification is based on the Joint North Sea wave project (JONSWAP) (Hasselmann et al., 1980).

3.3 Kinetics

There are two kinds of models, the PPM and the CPM. The PPM should be as in detail as possible to include all physical factors. PPM is used to simulate the real process in the control system. CPM is a simplified model which is used to design the controller and observer (Sørensen, 2012).

3.3.1 Process plant model

The PPM of a TAPM system is given by

$$\begin{aligned} \boldsymbol{M}(\omega)\dot{\boldsymbol{\nu}} + \boldsymbol{C_{RB}}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{C_A}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{D}(\boldsymbol{\nu}_r) + \boldsymbol{G}(\boldsymbol{\eta}) \\ = \boldsymbol{\tau_{wind}} + \boldsymbol{\tau_{wave2}} + \boldsymbol{\tau_m} + \boldsymbol{\tau_{thr}}, \end{aligned} \tag{3.2}$$

where $\boldsymbol{M}\epsilon\mathbb{R}^{6\times6}$ represents system inertia matrix, including the added mass, $\boldsymbol{C}_{RB}\epsilon\mathbb{R}^{6\times6}$ and $\boldsymbol{C}_A\epsilon\mathbb{R}^{6\times6}$ are skew symmetric Coriolis and centripetal matrices of the rigid body and added mass, $\boldsymbol{D}$ denotes damping vector including linear and nonlinear, $\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{\nu}_c$ is the relative velocity vector, and the current velocity $\boldsymbol{\nu}_c$ is expressed in body frame. $\boldsymbol{G}(\boldsymbol{\eta})\epsilon\mathbb{R}^6$ is the generalized restoring force caused by buoyancy and gravity, $\boldsymbol{\tau}_{wind}$, $\boldsymbol{\tau}_{wave1}$, $\boldsymbol{\tau}_m$, and $\boldsymbol{\tau}_{thr}\epsilon\mathbb{R}^6$ are the wind load vector, the second-order wave drift load vector, the mooring load vector,and thruster-induced forces vector, respectively.

3.3.2 Control plant model

The CPM is given by

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu}, \tag{3.3}$$

$$\dot{\boldsymbol{b}} = -\boldsymbol{T_b}\boldsymbol{b} + \omega_b, \tag{3.4}$$

Table 3.1: Definition of Sea States from (Price and Bishop, 1974)

Sea Status	Dominated Wave Frequency $\omega_0(rad/s)$	Significant Wave Height $H_s(m)$
Calm Seas	> 1.11	< 0.1
Moderate Seas	$[0.74, 1.11]$	$[0.1, 1.69]$
High Seas	$[0.53, 0.74]$	$[1.69, 6.0]$
Extreme Seas	< 0.53	> 6.0

$$\dot{\boldsymbol{v}} = -\boldsymbol{M}^{-1}D_L\boldsymbol{v} + \boldsymbol{M}^{-1}\boldsymbol{R}^{\top}(\psi) + \boldsymbol{M}^{-1}(\boldsymbol{\tau}_c - \boldsymbol{G}), \tag{3.5}$$

where, $\boldsymbol{M}\epsilon\mathbb{R}^{3\times3}$ is the mass matrix, $\boldsymbol{D}_L\epsilon\mathbb{R}^{3\times3}$ is the linear damping term, $\boldsymbol{b}\epsilon\mathbb{R}^3$ is the slow varying disturbances, $\boldsymbol{\tau}_c\epsilon\mathbb{R}^3$ is the thruster input.

3.4 Force allocation

3.4.1 Mooring force allocation

For the LF motion model, a horizontal-plane spread mooring model is formulated as

$$\boldsymbol{\tau_m} = -\boldsymbol{R}^{\top}(\boldsymbol{\eta}_2)\boldsymbol{g}_{mo}(\boldsymbol{\eta}) - \boldsymbol{d}_{mo}(\boldsymbol{v}), \tag{3.6}$$

where the mooring system is assumed to be symmetrically arranged. Assuming fixed anchor line length, damping effects of mooring line can be approximated by a linearized mooring damping matrix $\boldsymbol{d}_{mo}(\boldsymbol{v})\epsilon\mathbb{R}^3$. It is a common practice to estimate the linear damping of the mooring line by about $10-20\%$ of critical damping of the entire system. We have augmented the estimated linear damping of the mooring system into the damping term $\boldsymbol{Dv}$ in the left hand side of Eq. 3.2. $\boldsymbol{g}_{mo}(\boldsymbol{\eta})\epsilon\mathbb{R}^3$ is the Earth-fixed restoring force component, given by

$$\boldsymbol{g}_{mo} = \boldsymbol{T}(\boldsymbol{\beta}, \bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})\boldsymbol{h}_m, \tag{3.7}$$

where $\boldsymbol{\beta}\epsilon\mathbb{R}^N$ is the mooring line orientation vector consisting of the angles between the mooring lines and the x-axis, for $i = 1\cdots N$, and the moment arm vector are $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{y}}$. The mooring line configuration matrix $\boldsymbol{T}(\boldsymbol{\beta})$ is given by

$$\boldsymbol{T}(\boldsymbol{\beta}, \bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) = \begin{bmatrix} cos\beta_1 & \cdots & cos\beta_N \\ sin\beta_1 & \cdots & sin\beta_N \\ \bar{x}_1 sin\beta_1 - \bar{y}_1 cos\beta_1 & \cdots & \bar{x}_N sin\beta_N - \bar{y}_N cos\beta_N \end{bmatrix}, \tag{3.8}$$

where $\bar{x}_i\epsilon\bar{\boldsymbol{x}}$ and $\bar{y}_i\epsilon\bar{\boldsymbol{y}}$ are the horizontal displacements of the i^{th} mooring line between turret terminal point (TP) and Anchor i cable. The horizontal mooring force vector is denoted by $\boldsymbol{h}_m$= $[H_1, H_2, \cdots, H_N]^{\top}$, where H_i represents the horizontal force component at TP_i. Fig. 3.2 shows the configuration of a single mooring line.

3.4.2 Thrust allocation

Thrust configuration matrix

For a marine craft equipped with r thrusters in n DOFs,

$$\tau_c = \boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{K}\boldsymbol{u}, \tag{3.9}$$

where $\tau_c = [\tau_{surge}, \tau_{sway}, \tau_{yaw}]^\top \epsilon \mathbb{R}^3$ is the vector of the actuator forces and moments, $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_p]^\top \epsilon \mathbb{R}^p$ is a vector of azimuth angles, $\boldsymbol{T}(\boldsymbol{\alpha}) \epsilon \mathbb{R}^{n \times r}$ is the thrust configuration matrix which describes the geometry or location of the thrusters, $\boldsymbol{K}$= $diag(K_1, K_2, \cdots, K_r) \epsilon \mathbb{R}^{r \times r}$ is a diagonal force coefficient matrix, $\boldsymbol{u}$= $[u_1, u_2, \cdots, u_r]^\top$ is a vector of control inputs.
For an azimuth thruster, the i^{th} column of the 3DOF thruster configuration matrix is given by

$$\boldsymbol{t}_i = \begin{bmatrix} cos(\alpha_i) \\ sin(\alpha_i) \\ l_{xi} sin(\alpha_i) - l_{yi} cos(\alpha_i) \end{bmatrix}. \tag{3.10}$$

For a main propeller, the i^{th} column of the 3DOF thruster configuration matrix is given by

$$\boldsymbol{t}_i = \begin{bmatrix} 1 \\ 0 \\ -l_{yi} \end{bmatrix}. \tag{3.11}$$

For a tunnel thruster and aft rudder, the i^{th} column of the 3DOF thruster configuration matrix is given by

$$\boldsymbol{t}_i = \begin{bmatrix} 0 \\ 1 \\ l_{xi} \end{bmatrix}. \tag{3.12}$$

Moore-Penrose Pseudo Inverse

When the number of thrusters is bigger than the amount of the DOFs, it is unavailable to find the inverse matrix with basic linear transformation. Moore-Penrose pseudo inverse is used to here to allocate the thruster inputs. The representation is given by

$$\boldsymbol{T}_w^\dagger(\boldsymbol{\alpha}) = \boldsymbol{T}^\top(\boldsymbol{\alpha})[\boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{T}^\top(\boldsymbol{\alpha})]^{-1}. \tag{3.13}$$

However, the control cost for each thruster differs. A more advanced method to solve this problem is to employ a weight matrix $\boldsymbol{W}$,

$$\boldsymbol{T}_w^{\dagger}(\boldsymbol{\alpha}) = \boldsymbol{W}^{-1}\boldsymbol{T}^{\top}(\boldsymbol{\alpha})[\boldsymbol{T}(\alpha)[W^{-1}\boldsymbol{T}]^{\top}(\boldsymbol{\alpha})]^{-1}. \tag{3.14}$$

Furthermore, large amounts of researches have been conducted on the control allocation problem. For more details, please refer to Johansen and Fossen (2013). The control input of each thruster is given by

$$\boldsymbol{\tau}_{thr} = K^{-1}\boldsymbol{T}_w^{\dagger}(\boldsymbol{\alpha})\boldsymbol{\tau}_c. \tag{3.15}$$

3.5 Mooring models

3.5.1 Linear model

The linear model is the simplest model. The mooring force is assumed to have a linear relation with the horizontal displacement toward the equilibrium point. The mooring force is given by

$$\boldsymbol{\tau}_{moor} = -\boldsymbol{R}^{\top}(\psi)\boldsymbol{G}_{mo}(\boldsymbol{\eta}-\boldsymbol{\eta}_0) - \boldsymbol{D}_{mo}\boldsymbol{\nu}, \tag{3.16}$$

where $\boldsymbol{\eta}\epsilon\mathbb{R}^6$, $\boldsymbol{\eta}_0\epsilon\mathbb{R}^6$, $\boldsymbol{\nu}\epsilon\mathbb{R}^6$ are the real time position, the equilibrium position and the velocities, respectively, in the Earth-fixed frame and body-fixed frame, and $\boldsymbol{G}_{mo}$ and $\boldsymbol{D}_{mo}$ are linearized mooring damping and the stiffness matrices assumed to only contribute to the horizontal-plane, such that

$$\boldsymbol{G}_{mo} = \left.\frac{\partial \boldsymbol{g}_{mo}}{\partial \boldsymbol{\eta}}\right|_{\boldsymbol{\eta}=\boldsymbol{\eta}_0}, \tag{3.17}$$

$$\boldsymbol{D}_{mo} = \left.\frac{\partial \boldsymbol{d}_{mo}}{\partial \boldsymbol{\nu}}\right|_{\boldsymbol{\nu}=\boldsymbol{\nu}_0}. \tag{3.18}$$

For simplicity, the Earth-fixed frame is often placed in the natural equilibrium point of the mooring system, i.e. $\boldsymbol{g}_{mo} = \boldsymbol{0}$. They can, for symmetrical mooring patterns about the $xz-$ and $yz-$ planes, be formulated as

$$\boldsymbol{G}_{mo} = diag(g_{m11}, g_{m22}, 0, 0, 0, g_{m66}), \tag{3.19}$$

$$\boldsymbol{D}_{mo} = diag(d_{m11}, d_{m22}, 0, 0, 0, d_{m66}). \tag{3.20}$$

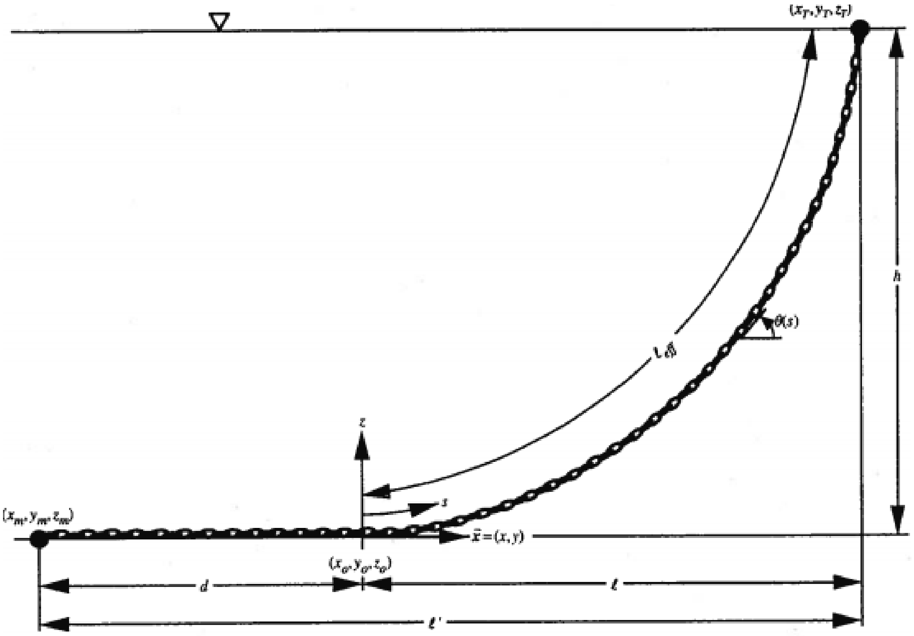


Figure 3.2: Catenary equation

3.5.2 Catenary equation model

Fig. 3.2 shows the configuration of a single mooring line. The 2D catenary equations (3.21) and (3.22) are used in this case to calculate H_i and the vertical mooring force component V_i. The catenary equations are given by

$$x_i(s) = \frac{H_i}{E_m A_m} s + \frac{H_i}{\omega_m}\left\{ sinh^{-1}\left[\frac{V_i - \omega_m (L_m - s)}{H_i}\right] - sinh^{-1}\left[\frac{V_i - \omega_m L_m}{H_i}\right]\right\}, \tag{3.21}$$

$$z_i(s) = \frac{1}{E_m A_m}\left[V_i s + \frac{\omega_m}{2}\left((L_m - s)^2 - L_m^2\right)\right] + \frac{H_i}{\omega_m}\left[\sqrt{1+\left(\frac{V_i - \omega_m (L_m - s)}{H_i}\right)^2} + \sqrt{1+\left(\frac{V_i - \omega_m L_m}{H_i}\right)^2}\right], \tag{3.22}$$

where s is the path parameter along the cable, $x_i(s)$ and $z_i(s)$ are the positions of each segment centered at length s along the i^{th} cable, L_m is the unstretched line length of the mooring lines, ω_m is the weight in water per unit length, E_m is the Young's modulus of elasticity, A_m stands for the cross-section area of the line, $T_i = \sqrt{V_i^2 + H_i^2}$ is tension at the end of the i^{th} mooring line, and $\phi_i = arctan(V_i/H_i)$ is the angle between the line tension and its vertical component.

The core algorithm can be summarized as Algorithm 1 to Algorithm 3, and flow diagram is shown in Figure 3.3.

Algorithm 1:

1. Given V_m and φ at the top point, calculate H_m, set $i = ns$, where ns is the number of segments.

2. Based on V_m and H_m, calculate the horizontal and vertical displacement of segment i using the catenary equations given in (2.44) and (2.45)

3. Update V_m by subtracting the weight of segment i. If $i > 1$ go to step 2.

4. Add up the displacements for all segments.

Algorithm 2:Depending on the result of Algorithm 1, V_m and φ are adjusted one at a time. For given V_m, a binary search is carried out in term of φ in order to hit the correct vertical displacement, which is the water depth D_w, as follows.

1. Maximum angle φ_{max} is taken to be slightly less than 90^o, and minimum angle is taken to be slightly more than 0^o.

2. Let $d = 0.5(\varphi_{max} - \varphi_{min})$. If d is less than an error tolerance, then stop. Otherwise set $\varphi = \varphi_{min} + d$, and perform Algorithm 1.

3. If the vertical displacement is less than D_w, set $\varphi_{min} = \varphi$, otherwise set $\varphi_{max} = \varphi$. Go to step 2.

Algorithm 3: It remains to find V_m, such that a binary search is again performed on V_m until the horizontal displacement meets the prescribed value, which is the horizontal displacement X, as follows.

1. Take the total mass of the system as the initial guess for V_m.

2. Do Algorithm 2. If the resulting horizontal displacement is larger than X, a touch down point is needed other than the anchor point, and the guess is upper bound. In this case, set $V_{max} = V_m$ and V_{min} = the weight of a piece of the upper end cable. Otherwise, the anchor point will have vertical load, and the guess is lower bound, that is $V_{min} = V_m$. In this case, increase V_m by some percentage and do Algorithm 2 until the horizontal displacement is larger than X. Set V_{max} to this value V_m.

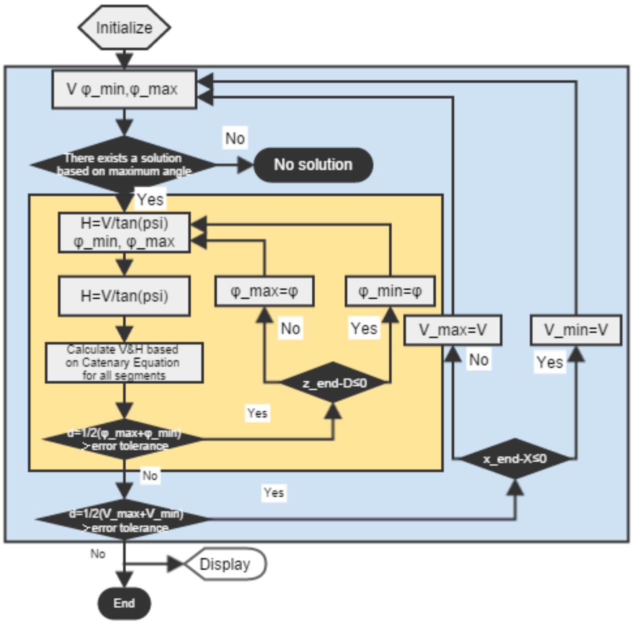

Initialize
V φ_min,φ_max
There exists a solution based on maximum angle
No
No solution
Yes
H=V/tan(psi) φ_min, φ_max
H=V/tan(psi)
φ_max=φ
φ_min=φ
No
Yes
V_max=V
V_min=V
No
Yes
Calculate V&H based on Catenary Equation for all segments
z_end-D≤0
Yes
d=1/2(φ_max+φ_min) >error tolerance
No
x_end-X≤0
Yes
d=1/2(V_max+V_min) >error tolerance
No
Display
End

Figure 3.3: Flow diagram of the catenary equation model

3. Let $d = 0.5(V_{max} - V_{min})$. If d is less than an error tolerance, then stop. Otherwise, set $V_m = V_{min} + d$.

4. Calculate the length of the suspended cable according to V_m, and do Algorithm 2. Using the resulting horizontal tension at the bottom, calculate the length of the remaining part along the seafloor. Add the result to the horizontal displacement resulting from Algorithm 2.

5. If the horizontal is less than X, set $Vmin = V_m$. Otherwise, set $Vmax = V_m$. Go to step 3.

3.5.3 Finite element method model

The state equation could be expressed into two parts. The first part is a more general representation for all nodes except the top end which connects to the turret. These nodes are influenced by their neighboring nodes. The equation is given by

$$\boldsymbol{M}_k^j \dot{\boldsymbol{v}}_k^j + \boldsymbol{D}_k^j \boldsymbol{v}_k^j + \boldsymbol{k}_k^j + \boldsymbol{g}_k^j = 0, \quad k = 1, 2, \cdots, n-1, \quad j = 1, 2, \cdots, m \tag{3.23}$$

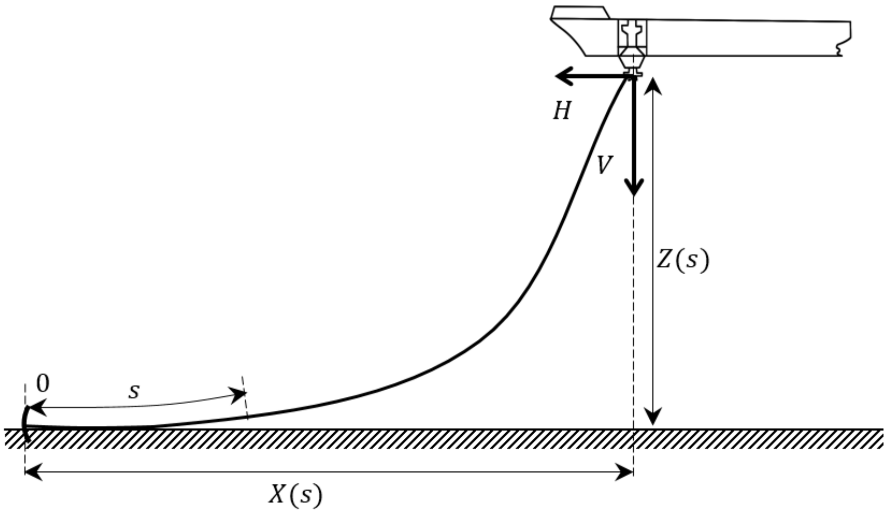


Figure 3.4: FEM model

$$\boldsymbol{M}_k^j = [rho_0 l_j + C_1^j(\epsilon_k^j + \epsilon_{k+1}^j)]\boldsymbol{I}_{3\times 3} - \frac{C_1^j}{2}(\epsilon_k^j \boldsymbol{P}_k^j + \epsilon_{k+1}^j \boldsymbol{P}_{k+1}^j) \tag{3.24}$$

$$\begin{aligned}\boldsymbol{D}_k^j =& \frac{C_2}{2}[|\dot{\boldsymbol{r}}_k \boldsymbol{l}_k|\boldsymbol{P}_k^j] + \frac{C_3}{2}[\epsilon_k|(\boldsymbol{I}_{3\times 3} - \boldsymbol{P}_k^j)\dot{\boldsymbol{r}}_k|\boldsymbol{I}_{3\times 3} - \boldsymbol{P}_k^j) + \\ &\epsilon_{k+1}|(\boldsymbol{I}_{3\times 3} - \boldsymbol{P}_{k+1}^j)\dot{\boldsymbol{r}}_k||\boldsymbol{I}_{3\times 3} - \boldsymbol{P}_{k+1}^j)]\end{aligned} \tag{3.25}$$

$$\boldsymbol{k}_k^j = \frac{E_j A_{0_j}}{l_j}[\frac{\epsilon_k^j - l_j}{\epsilon_k}\boldsymbol{l}_k - \frac{\epsilon_k^j - l_j}{\epsilon_{k+1}}\boldsymbol{l}_{k+1}] \tag{3.26}$$

$$\boldsymbol{g}_k^j = l_j \rho_0 \frac{\rho_c - \rho_w}{\rho_c}\begin{bmatrix}0 & 0 & g\end{bmatrix}^\top \tag{3.27}$$

The second part is the top end note which is connected to the turret. This node is only influenced by the former node. After simplifying the first part representation, the second part representation is given by

$$\boldsymbol{M}_n^j \dot{\boldsymbol{v}}_n^j + \boldsymbol{D}_n^j \boldsymbol{v}_n^j + \boldsymbol{k}_n^j + \boldsymbol{g}_n^j = 0, \quad j = 1, 2, \cdots, m \tag{3.28}$$

$$\boldsymbol{M}_n^j = [\rho_0 l_j + C_1^j(\epsilon_n^j + \epsilon_{k+1}^j)]\boldsymbol{I}_{3\times 3} - C_1^j(\epsilon_n^j \boldsymbol{P}_n^j \tag{3.29}$$

$$\boldsymbol{D}_n^j = C_2[|\dot{\boldsymbol{r}}_n \boldsymbol{l}_n|\boldsymbol{P}_n^j] + C_3\epsilon_k^j|(\boldsymbol{I}_{3\times 3} - \boldsymbol{P}_n^j)\dot{\boldsymbol{r}}_n|(\boldsymbol{I}_{3\times 3} - \boldsymbol{P}_k^j) \tag{3.30}$$

$$\boldsymbol{k}_n^j = \frac{E_j A_{0_j}}{l_j}[\frac{\epsilon_n^j - l_j}{\epsilon_n^j}\boldsymbol{l}_n^j] \tag{3.31}$$

$$\boldsymbol{g}_n^j = l_j \rho_0^j \frac{\rho_c - \rho_w}{\rho_c}\begin{bmatrix}0 & 0 & g\end{bmatrix}^\top \tag{3.32}$$

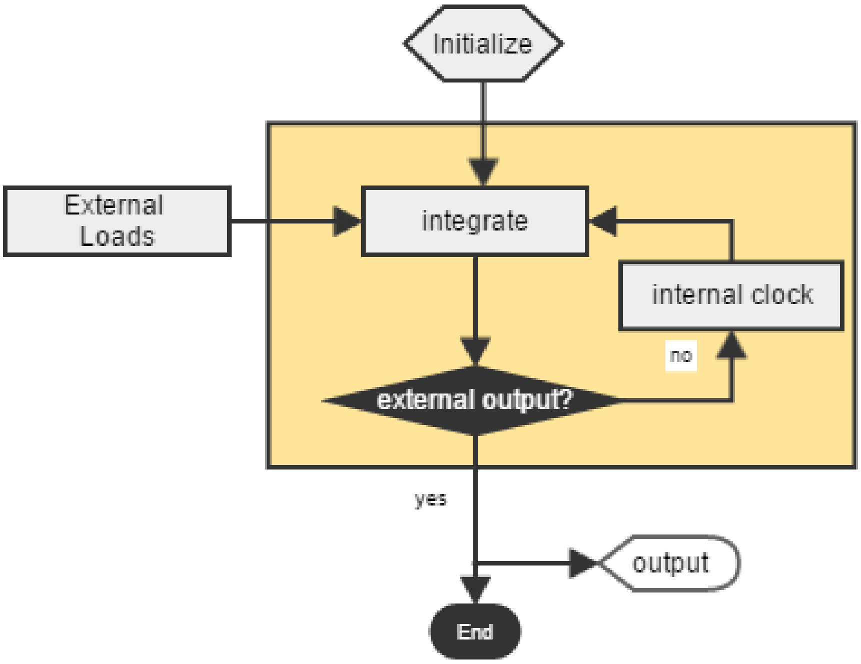


Figure 3.5: Flow diagram of the FEM model

For a marine control system, the FEM model can be applied to model the dynamics of flexible structures, such as mooring lines' and risers' dynamics. Unlike ordinary differential e-quations (ODE), PDE has more than one variable. PDE should be solved with boundary conditions (Sørensen, 2012).

3.5.4 Turret dynamics

For a low-frequency (LF) model, a horizontal-plane spread mooring model is formulated as

$$\boldsymbol{g}_{mo} = \begin{bmatrix} \boldsymbol{g}^{t}_{mo,1:2} \\ D^{t}_{z}\dot{\tilde{\psi}}_{t} \end{bmatrix} \tag{3.33}$$

where ψ_t is the angle of the turret compared with the reference, $\tilde{\psi}_t = \psi_t - \psi$ is the relative angle between the turret and the heading of the moored vessel, I^t_z is the mass inertia of moment of the turret, and D^t_z is the damping between the vessel and the turret. The dynamic model of $\tilde{\psi}_t$ is given by

$$I^{t}_{z}\ddot{\tilde{\psi}}_{t} = -\boldsymbol{g}_{mo,3} - D^{t}_{z}\dot{\tilde{\psi}}_{t}. \tag{3.34}$$

The restoring forces and moment $\boldsymbol{g}^t_{mo}(\boldsymbol{\eta}) \epsilon \mathbb{R}^3$, which the mooring lines exert on the turret, is given by

$$\boldsymbol{g}^t_{mo}(\boldsymbol{\eta}) = \boldsymbol{T}(\boldsymbol{\beta}, \bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})\boldsymbol{h}_m \tag{3.35}$$

where $\boldsymbol{\beta} \epsilon \mathbb{R}^M$ is the mooring line orientation vector consisting of the angles between the mooring lines and the x-axis, for $i = 1 \cdots M$, and the moment arm vectors are $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{y}}$.

3.6 Nonlinear passive observer

The passive observer is introduced in the late 1990s by Strand and Fossen (1999). The main motivation is to avoid tuning numerous parameters in designing a Kalman filter, as well as reducing the number of code lines. Nonlinear passive observer has significant advantages during the tuning period. Based on the following assumptions, the Coriolis terms, $\boldsymbol{C_{RB}}(\boldsymbol{\nu})\boldsymbol{\nu}$ and $\boldsymbol{C_A}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r$, and the nonlinear damping term, $\boldsymbol{D}_{NL}|\boldsymbol{\nu}|\boldsymbol{\nu}$, are small due to the small velocity assumption for DP and TAPM vessels.

Assumption 1. The amplitude of the wave-induced yaw motion is small, that is, less than $2-3$ (deg) during normal operation of the vessel and less than 5 (deg) in extreme weather conditions.

Assumption 2. The heading rate $\dot{\psi}$ is small and close to zero.

Assumption 3. Position and heading sensor noises are neglected.

The NPO admits the realization

$$\dot{\hat{\boldsymbol{\eta}}}_w = \boldsymbol{A_{pw}}(\omega_0)\hat{\boldsymbol{\eta}}_w + \boldsymbol{K_1}\tilde{\boldsymbol{y}}, \tag{3.36}$$

$$\dot{\hat{\boldsymbol{\eta}}} = \boldsymbol{R}(\psi)\hat{\boldsymbol{\nu}} + \boldsymbol{K_2}\tilde{\boldsymbol{y}}, \tag{3.37}$$

$$\dot{\hat{\boldsymbol{b}}} = -\boldsymbol{T_b}^{-1}\hat{\boldsymbol{b}} + \boldsymbol{K_3}\tilde{\boldsymbol{y}}, \tag{3.38}$$

$$\boldsymbol{M}\dot{\hat{\boldsymbol{\nu}}} = -\boldsymbol{D}\hat{\boldsymbol{\nu}} + \boldsymbol{R}(\psi)^\top\hat{\boldsymbol{b}} - \boldsymbol{R}(\psi)^\top\boldsymbol{T}(\hat{\boldsymbol{\beta}})\boldsymbol{\tau}_{H,m} + \boldsymbol{\tau}_c + \boldsymbol{K_4}\boldsymbol{R}(\psi)^\top\tilde{\boldsymbol{y}}, \tag{3.39}$$

$$\hat{\boldsymbol{y}} = \hat{\boldsymbol{\eta}} + \boldsymbol{C}_{pw}\hat{\boldsymbol{\eta}}_w, \tag{3.40}$$

where $\hat{\boldsymbol{\eta}}_w \epsilon \mathbb{R}^6$ is the state of the WF model, $\boldsymbol{A_{pw}} \epsilon \mathbb{R}^{6\times6}$ is assumed Hurwitz, and describes the first-order WF-induced motion as a mass-damper-spring system, $\boldsymbol{b} \epsilon \mathbb{R}^3$ is the bias vector accounting for both slowly varying disturbances and unmodeled dynamics, $\boldsymbol{\tau}_c \epsilon \mathbb{R}^3$ is the control input vector, $\boldsymbol{y}$=$\boldsymbol{y}-\hat{\boldsymbol{y}}$ is the output estimation error, $\boldsymbol{K_1} \epsilon \mathbb{R}^{6\times3}$, $\boldsymbol{K_2} \epsilon \mathbb{R}^{3\times3}$, $\boldsymbol{K_3} \epsilon \mathbb{R}^{3\times3}$, and

$\boldsymbol{K_4} \epsilon \mathbb{R}^{3\times3}$ are the observer gain matrices.

$$\boldsymbol{A_{pw}} = \begin{bmatrix} \mathbf{0}_{3\times3} & \boldsymbol{I}_{3\times3} \\ \boldsymbol{\Omega}^2 & -2\boldsymbol{\Gamma\Omega}, \end{bmatrix} \tag{3.41}$$

where $\boldsymbol{\Omega}$= $diag(\omega_1,\omega_2,\omega_3)$ is a diagonal matrix containing the dominating wave response frequencies, and $\boldsymbol{\Gamma\Gamma} = diag(\lambda_1,\lambda_2,\lambda_3)$ is a diagonal matrix of damping ratios. $\lambda_i$is often set between 0.05 and 0.2.

Following formulas are the filter gains in $K_1(w_0)$ and K_2,

$$K_{1i}(w_{oi}) = -2(\xi_{ni} - \lambda_i)\frac{w_{ci}}{w_{oi}}, \tag{3.42}$$

$$K_{1(i+3)}(w_{oi}) = 2w_{oi}(\xi_{ni} - \lambda_i), \tag{3.43}$$

$$K_{2i} = w_{ci}. \tag{3.44}$$

The parameters are chosen as typical values $\xi_{ni} = 1.0$, $\lambda = 0.1$ (Fossen, 2011). As discussed in Section 2, this part could be omitted.

3.7 PID controller

Based on the estimation from the observer, output feedback PID controller is applied to keep the LF heading and position in reasonable range. Nguyen et al. (2007) designs a hybrid control system which could set different PID parameters for divergent sea states, such as calm sea, normal sea, and extreme sea.

$$\dot{\hat{\boldsymbol{\xi}}} = \hat{\boldsymbol{\eta}}_e, \tag{3.45}$$

$$\hat{\boldsymbol{\tau}}_c = -\boldsymbol{H}_i\boldsymbol{K}_i\boldsymbol{R}^\top(\psi_y)\hat{\boldsymbol{\xi}} - \boldsymbol{H}_p\boldsymbol{K}_i\boldsymbol{R}^\top(\psi_y)\hat{\boldsymbol{\eta}}_e - \boldsymbol{H}_d\boldsymbol{K}_d\boldsymbol{R}^\top(\psi_y)\hat{\boldsymbol{\nu}}_e, \tag{3.46}$$

where, $\hat{\boldsymbol{\eta}}_e = \hat{\boldsymbol{\eta}} - \boldsymbol{\eta}_d$, $\hat{\boldsymbol{\nu}}_e = \hat{\boldsymbol{\nu}} - \boldsymbol{\nu}_d$. $\boldsymbol{\eta}_d$ and $\boldsymbol{\nu}_d$ are the desired position and velocity generated by reference system. $\boldsymbol{K}_p$, $\boldsymbol{K}_i$, and $\boldsymbol{K}_d \epsilon \mathbb{R}^{3\times3}$ are non-negative P, I, D controller gain matrices. ψ_y is the heading angle. $\boldsymbol{H}_p$, $\boldsymbol{H}_i$, and $\boldsymbol{H}_d \epsilon \mathbb{R}^{3\times3}$ are the projection matrices with only 0 or 1 at the diagonal entries, which are used to enable or disable the control forces in different DOFs.

In all sea states, heading control is necessary. Pure heading control is expressed as

$$\hat{\boldsymbol{\tau}}^{\psi}_{PID} = -\boldsymbol{H}^{\psi}_i\boldsymbol{K}_i\boldsymbol{R}^\top(\psi_y)\hat{\boldsymbol{\xi}} - \boldsymbol{H}^{\psi}_p\boldsymbol{K}_i\boldsymbol{R}^\top(\psi_y)\hat{\boldsymbol{\eta}}_e - \boldsymbol{H}^{\psi}_d\boldsymbol{K}_d\boldsymbol{R}^\top(\psi_y)\hat{\nu}_e, \tag{3.47}$$

where, $\boldsymbol{H}_p^{\psi} = \boldsymbol{H}_i^{\psi} = \boldsymbol{H}_d^{\psi} = diag[0,0,1]$.

Additionally, damping control (3.48) and restoring control (3.49 and 3.50) are needed in normal sea states and extreme sea states.

$$\hat{\boldsymbol{\tau}}_D^{xy} = -\boldsymbol{H}_d^{xy}\boldsymbol{K}_d\boldsymbol{R}^{\top}(\psi_y)\hat{v}_e, \tag{3.48}$$

$$\hat{\boldsymbol{\tau}}_P^{xy} = -\boldsymbol{H}_p^{xy}\boldsymbol{K}_i\boldsymbol{R}^{\top}(\psi_y)\hat{\boldsymbol{\eta}}_e, \tag{3.49}$$

$$\hat{\boldsymbol{\tau}}_I^{xy} = -\boldsymbol{H}_i^{xy}\boldsymbol{K}_i\boldsymbol{R}^{\top}(\psi_y)\hat{\xi}, \tag{3.50}$$

where, $\boldsymbol{H}_p^{xy} = \boldsymbol{H}_i^{xy} = \boldsymbol{H}_d^{\psi} = diag[1,1,0]$.

Table 3.2: Plant Controllers in different vessel of condition

Heading	Damping	Restoring	Mean force	Plant Controller	VOC
✓				Eq.(3.47)	Calm sea
✓	✓			Eq.(3.47)+(3.48)	Normal sea
✓	✓	✓		Eq.(3.47)+(3.48)+(3.49)	Normal sea
✓	✓		✓	Eq.(3.47)+(3.48)+(3.50)	Extreme sea
✓	✓	✓	✓	Eq.(3.47)+(3.48)+(3.49)+(3.50)	Extreme sea

3.8 Reference system

As introduced in(Nguyen et al., 2007a), the reference model provides setpoint signal to the controller. It is a low pass signal of the LF position vector $\boldsymbol{\eta}$, and the setpoint is estimated by

$$\dot{\boldsymbol{\eta}}_r = -\Lambda\boldsymbol{\eta}_r + \Lambda\boldsymbol{\eta}, \tag{3.51}$$

where, $\boldsymbol{\eta}_r$ is the reference signal output from the reference system, Γ is the first order diagonal and non-negative filter gain matrix with the cut-off frequencies $\frac{1}{T_{s_i}}$(Hz), $\Lambda = diag(\frac{1}{T_{s_1}}, \frac{1}{T_{s_2}}, \frac{1}{T_{s_3}})$.

Additionally, Fossen (2011) illustrates more advanced reference systems. One is the velocity reference (see Figure 20),

$$\ddot{\boldsymbol{v}}_d + 2\Delta\Omega\dot{\boldsymbol{v}}_d + \Gamma^2\boldsymbol{v}_d = \Gamma^2\boldsymbol{r}_b \tag{3.52}$$

.

The other one is the third order position reference with an expression as

$$\boldsymbol{\eta}_d^{(3)} + (2\Delta + I)\Gamma\ddot{\boldsymbol{\eta}}_d + +(2\Delta + I)\Gamma\dot{\boldsymbol{\eta}}_d + \Gamma^3\dot{\boldsymbol{\eta}}_d = \Gamma^3 r_n, \tag{3.53}$$

where $\boldsymbol{\eta}_d \epsilon \mathbb{R}^3$ is the desired position, $\Delta \epsilon \mathbb{R}^{3\times3}$ is the relative damping ratios, and $\Delta = diag(\zeta_1, \zeta_2, \zeta_3)$ to maintain critical damping, $\Gamma \epsilon \mathbb{R}^{3\times3}$ is the natural frequencies matrix, and it is selected as $\Gamma = diag(\omega_{n1}, \omega_{n2}, \omega_{n3})$.

Chapter 4

Sensor fusion technique

In this chapter, basic sensor fusion questions and approaches are reviewed. Sensor fusion is the combining of sensory data or data derived from sensory data from disparate sources such that the resulting information has less uncertainty than would be possible when these sources were used individually.

4.1 Least square approach

The vector form of a linear sensor model is

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x}^{o} + \boldsymbol{e}, \qquad cov(e) = \boldsymbol{R}. \tag{4.1}$$

where $\boldsymbol{R} = diag(R_1, R_2, \cdots, R_N)$ and $R_k = cov(e_k)$.

The state $\boldsymbol{x}$ has a n_s dimension, and the measurement state $\boldsymbol{y}$ has N sensors.

Table 4.1: Least square and weighted least square sensor fusion

	Least square (LS)	Weight least square (WLS)
Optimization problem	$\hat{x}^{LS} = \arg\min_{x} V^{LS}(x)$	$\hat{x}^{WLS} = \arg\min_{x} V^{WLS}(x)$
Objective function	$V^{LS}(x) = (\boldsymbol{y}\boldsymbol{H}\boldsymbol{x})^{\top}(\boldsymbol{y}\boldsymbol{H}\boldsymbol{x})$	$V^{WLS}(x) = (\boldsymbol{y}\boldsymbol{H}\boldsymbol{x})^{\top}\boldsymbol{R}^{-1}(\boldsymbol{y}\boldsymbol{H}\boldsymbol{x})$
Solution	$\hat{x}^{LS} = (\boldsymbol{H}^{\top}\boldsymbol{H})^{-1}\boldsymbol{H}^{\top}\boldsymbol{y}$	$\hat{x}^{WLS} = (\boldsymbol{H}^{\top}\boldsymbol{R}^{-1}\boldsymbol{H})^{-1}\boldsymbol{H}^{\top}\boldsymbol{R}^{-1}\boldsymbol{y}$
Regarding true value x^{o}	$\hat{x}^{LS} = x^{o} + (\boldsymbol{H}^{\top}\boldsymbol{H})^{-1}\boldsymbol{H}^{\top}\boldsymbol{e}$	$\hat{x}^{WLS} = x^{o} + (\boldsymbol{H}^{\top}\boldsymbol{R}^{-1}\boldsymbol{H})^{-1}\boldsymbol{H}^{\top}\boldsymbol{R}^{-1}\boldsymbol{e}$

4.2 Maximum likelihood approach

The maximum likelihood estimation $\hat{x}^{ML}$ is

$$\hat{x}^{ML} = \arg\min_{x} p(\boldsymbol{y}|x), \tag{4.2}$$

corrector-predictor for ship navigation using tow measurement rates are shown in Algorithm 1 and Algorithm 2.

4.3 GPS

The sensor fusion algorithms of GPS and IMU are shown in Algorithm 1 and Algorithm algorithm:GPS2 (Fossen, 2011).

```
Data: Sampling time $h$, counter for GPS measurements, initial state vector $\bar{\boldsymbol{x}} = \boldsymbol{x}_0$
while estimating do
    $\boldsymbol{y}_{IMU} = measurement$;
    $\boldsymbol{y}_{GPS} = measurement$;
    $\boldsymbol{K}_d = [hk_{IMU}, 0]^\top$;
    if GPS=10 then
        $\boldsymbol{K}_d = [hk_{IMU}, 10hk_{GPS}]^\top$;
        GPS=0;
    end
    if dead-reckoning (no updates) then
        $\boldsymbol{K}_d = [0, 0]^\top$ ;
    end
    $\boldsymbol{y} = [\boldsymbol{y}_{IMU}^\top, \boldsymbol{y}_{GPS}^\top]^\top$;
    $\hat{\boldsymbol{x}} = \bar{\boldsymbol{x}} + \boldsymbol{K}_d[\boldsymbol{y} - \boldsymbol{H}\bar{\boldsymbol{x}}]$ ;
    $\boldsymbol{u} = controlsystem(optionally)$ ;
    $\bar{\boldsymbol{x}} = \bar{\boldsymbol{x}} + h\boldsymbol{f}(\hat{\boldsymbol{x}}, \boldsymbol{u})$;
    $GPS = GPS + 1$ ;
end
```

Algorithm 1: Method A

4.4 Collaborative position localization

A crucial application of sensor network is to locate the nodes in the sensor network. All the sensors and the unknown positions are named as *nodes*, where the nodes with known positions are called *anchors* or the *reference nodes*. While the other unknown nodes are called as *unlocalized nodes*.

```
algorithm:GPS2 Data: Sampling time $h$, counter for GPS measurements, initial state
        vector $\bar{\boldsymbol{x}} = \boldsymbol{x}_0$
while estimating do
    $\boldsymbol{y}_{IMU} = measurement$;
    $\boldsymbol{K}_d = [hk_{IMU}, 10hk_{IMU}]^\top$;
    if GPS=10 then
        $\boldsymbol{y}_{GPS} = measurement$;
        GPS=0;
    end
    if dead-reckoning (no updates) then
        $\boldsymbol{K}_d = [0,0]^\top$ ;
    end
    $\boldsymbol{y} = [\boldsymbol{y}_{IMU}^\top, \boldsymbol{y}_{GPS}^\top]^\top$;
    $\hat{\boldsymbol{x}} = \bar{\boldsymbol{x}} + \boldsymbol{K}_d[\boldsymbol{y} - \boldsymbol{H}\bar{\boldsymbol{x}}]$ ;
    $\boldsymbol{u} = controlsystem(optionally)$ ;
    $\bar{\boldsymbol{x}} = \bar{\boldsymbol{x}} + h\boldsymbol{f}(\hat{\boldsymbol{x}}, \boldsymbol{u})$;
    $GPS = GPS + 1$ ;
end
```

Algorithm 2: Method B

Algorithms can be categorised as *range-based* or *range-free* localization based on of the range measurement is needed. Additionally, the technique can be sorted as *anchor-based* or *anchor-free*, relying on whether known anchor positions are need. Thirdly, the localization problem can be divided into *probabilistic* or *deterministic* localization problem. Deterministic method does not depend on additional information about the quality or reliability of the solution. The probabilistic localization approach solve the problem based on probabilistic inference.

second-order cone programming (SOCP) and semi-definite programming (SDP) are two most widely applied range-based optimization-based deterministic localization approaches in sensor network. SOCP and SDP do not need the initial estimations. They are derived from the maximum likelihood and the convex optimization.

The SOCP objective function and covariant functions are given by

$$\begin{aligned} v_{scop} = & \min_{\{\boldsymbol{x}_i\},\boldsymbol{u},\{q_{ij}\},v} \quad v \\ subject\,to\, & |\boldsymbol{u}| \le v \\ & g_{ij}|q_i j - d_i j| \le t_{ij} \\ & |\boldsymbol{\Psi}_i^{-0.5}(\boldsymbol{a}_i - \boldsymbol{x}_i)| \le s_i \\ & |\boldsymbol{x}_i - \boldsymbol{x}_j| \le q_{ij} \end{aligned} \tag{4.3}$$

The SDP objective function and covariant functions are given by

$$\begin{aligned}
v_{sdp} = & \min_{\boldsymbol{X},\boldsymbol{Y},\{\Xi_i\},\{\gamma_{ij}\},\{r_{ij}\}} \sum_{i=k+1} \sum_{j=1} g_{ij}^2 (\gamma_{ij} - 2d_{ij} r_{ij}) + \sum_{i=1} (\boldsymbol{tr(\psi_i^{-1} \Xi_i)} - 2\boldsymbol{a}_i^\top \Psi_i^{-1} \boldsymbol{x}_i) \\
subject\,to \quad & \gamma_{ij} = y_{ii} + y_{jj} - y_{ij} - y_{ji}, \quad i = k+1, k+2, \cdots, m, \quad j = 1, 2, \cdots, m \\
& r_{ij}^2 \leq \gamma_{ij}, \quad i = k+1, k+2, \cdots, m, \quad j = 1, 2, \cdots, m \\
& \boldsymbol{tr}(\Xi_i) = y_{ii} \\
& \boldsymbol{x}_i = [y_{im+1} y_{im+2}]^\top \\
& \begin{bmatrix} y_{m+1m+1} & y_{m+1m+2} \\ y_{m+2m+1} & y_{m+2m+2} \end{bmatrix} = \boldsymbol{I}_2 \\
& \begin{bmatrix} \Xi_i & \boldsymbol{x}_i \\ \boldsymbol{x}_i^\top & 1 \end{bmatrix} \succeq \boldsymbol{0}_3 \\
& \boldsymbol{Y} \succeq \boldsymbol{0}_{m+2}
\end{aligned} \tag{4.4}$$

where there are k known sensors in m sensors in the sensor network, $m-k$ unknown sensors, $\boldsymbol{X} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_m}]$, $y_{ij} = \boldsymbol{x_i}^\top \boldsymbol{x_j}$, $\Xi_i = \boldsymbol{x_i}^\top \boldsymbol{x_i}$, $\gamma_{ij} = r_{ij}^2 = \|\boldsymbol{x_i} - \boldsymbol{x_j}\|$, and define

$$g_{ij} = \begin{cases} \delta_{ij}/2\sigma_{ij}^2, & i > k \quad and \quad j > k \\ \delta_{ij}/\sigma_{ij}^2, & otherwise. \end{cases} \quad . \tag{4.5}$$

The SOCP has a faster but less accurate performance than the SDP. After implement the SOCP algorithm (see Section 11.1.4).

Chapter 5

TAPM Localization with Only Tension Measurements

This section proposes a new tension-based posref. The basic assumptions and models are discussed and deduced. It can be applied to detect the GPS failure, which is shown in Section 6.5.

5.1 Problem statement

Regardless of the environmental loads and thruster force, the moored vessel becomes stable at the equilibrium point with only restoring forces from mooring lines. When each mooring line is equipped with a tension cell, the tension is measured. Through this, we can estimate the relative position to the equilibrium point based on the tension measurements. Section 5.3 gives an easy deterministic localization algorithm based on the maximum likelihood optimization.

However, the algorithm is not robust enough. Sometimes, we cannot ensure precise anchor localization. Without accurate information of the anchors, we cannot have satisfied estimation. Finally, the fairleads are not considered during the calculation. In this 5.4 and 5.3, another scheme based on the sensor network and SLAM is introduced. The most significant benefit is that EKF can estimate the position with uncertain anchors, Additionally, fairleads are considered to give a more realistic result. Finally, a sensor network is built, and a complete EKF model and two simplified forms are concluded.

5.1.1 Term definitions

In this chapter, *anchor* and *anchor node* are two different terms with unlike meanings. To avoid confusion, we firstly define them as follows.

Definition 5.1 (Anchor)**.** *An anchor is a heavy device attached to a cable or chain and used to prevent the craft from drifting due to environmental loads (Oxford Advanced Learner's Dictionary, nd).*

Definition 5.2 (Anchor node)**.** *An anchor node is a node in a sensor network whose position is expected to have been known (Zekavat and Buehrer, 2011).*

5.2 Range measurement

5.2.1 Mapping from tension to distance

All the algorithms are based on the assumption 5.1.

Proposition 5.1 (Tension-range mapping assumption)**.** *Regarding a specific catenary mooring line, the tension aroused from gravity performs as a restoring force to the moored vessel. The tension force at its upper end exists a mapping relation to the horizontal projected distance between the anchor and the upper end, such that*

$$T_i^0 = f_i(X_i) + \Delta T_{i,c} + \Delta T_{i,v}, \tag{5.1}$$

$$T_i = T_i^0 + v_i, \tag{5.2}$$

where T_i^0 and T_i are the noiseless and noisy axial tension at the top end of the i^{th} mooring line, f_i is the mapping function, X_i is the distance between the i^{th} anchor and the moored vessel, $\Delta T_{i,c}$ and $\Delta T_{i,v}$ are the tension variance due to current and heave motion.

Proof. See Section 9.1.1. ■

When a mooring line is subjected to current loads, we can find the top tension is influenced.

Proposition 5.2 (Tension variance due to current)**.** *For a specific mooring line in an open environment which subjected to current loads, the variances in top tension, deformation, and* *the top end angle due to the current is quadratically proportional to the current speed. Additionally, they are affected by the current direction, which is concluded as follows. We employ a mooring line-fixed coordinate to simplify the description. The direction along the cable from the bottom to the top end in the horizontal plane is 0, and the reserve direction is* π.

- *The horizontal force component at the top end reach its maximum in the* $\frac{\pi}{2}$*-direction current; the vertical force component is minimized in the* $\frac{\pi}{2}$*-direction current and maximized in the* 0*-direction or* π*-direction current.*

- *Being the resultant force, the top tension reaches its maximum when the current direction is approximately* $\frac{\pi}{4}$ *or* $\frac{3\pi}{4}$.

- *The X/Y deformation along the cable is quadratically proportional to the current velocity component in X/Y direction.*

- *The top angle is reduced by a* 0*-direction current, while it increases with a* π*-direction current. Current with a direction smaller than* $\frac{\pi}{2}$ *has approximately equal, but slightly higher, influence than the current in a direction greater than* $\frac{\pi}{2}$.

Proof. Since the drag force in the Morison equation is quadratically proportional to the current speed, the results due to the current is expected. The proof is based on the simulation results of the FEM model, see Section 9.1.2. ■

Remark: The influence due to the current is limited when comparing with the top tension, Therefore, we have an LOS assumption.

5.2.2 Noise existed in the range measurement

If we assume the amplitude of the wave-induced heave motion is 4 meters, the result is shown in Figure 5.1. Assume the real position is B. When the vessel heaves at $z = 4m$, the inverse mapping gives a distance as the point A; the distance appears to stay at the point C when $z = -4m$. Based on the mapping relation at $z = 0$, the amplitude of the biases are 3m and 1.5m in the left segment and the right segment respectively. The wave-induced noise increases with the horizontal distance. Wave-induced motion introduce a zero-mean noise to the tension measurement, and the average heave motion is zero. Hence, we assume the dynamic tension from wave-induced heave is another Gaussian white noise process.

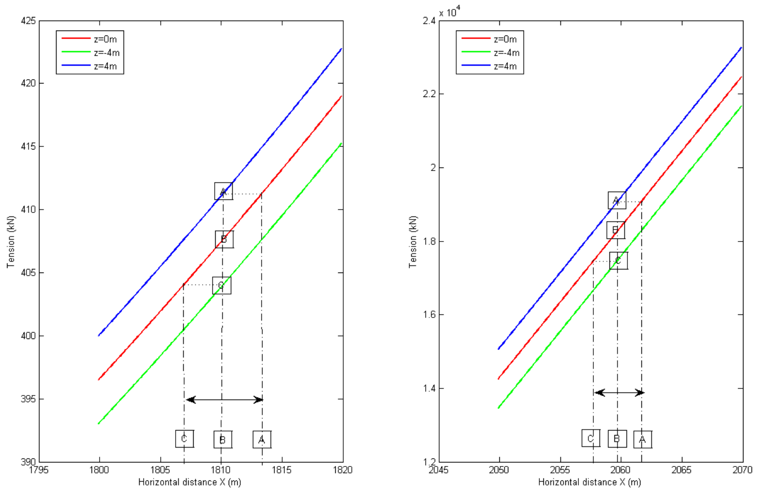


Figure 5.1: The influence of the wave-induced heave motion at different part.

Proposition 5.3 (Tension variance due to heave motion)**.** *For a catenary mooring line, the tension variance due to heave motion is given by*

$$\Delta T_i = -K_i z. \tag{5.3}$$

From Garza Rios et al. (1997), a simplified tension is given by

$$T_i(s = L_m) = H_i - K_i z. \tag{5.4}$$

As we assume a slow-speed stationkeeping model without roll and pitch motion, the heave motion at the fairlead is same as the vessel. Furthermore, the amplitude of the heave motion is much smaller than the cable length; hence, we assume the heave motion does not influence the projected suspended length. Therefore, the partial derivative of the tension with respective to heave motion is almost linear.

In Figure 5.2, the blue line is the curve about axial tension and the projected suspended length at the seafloor. The key parameters are tabulated in Table 8.2. The black lines are the tension with constant heaves. The interval between the lines is 0.4 m. From the figure 5.3, we notice the tension variance due to heave motion is almost a constant.

Gobat and Grosenbaugh (2001) presents an empirical model for dynamic tension due to heave-induced vertical motion at the end of the catenary mooring line. This model is applicable to the moored vessel with neglected horizontal effects and not-too-large elastic com-

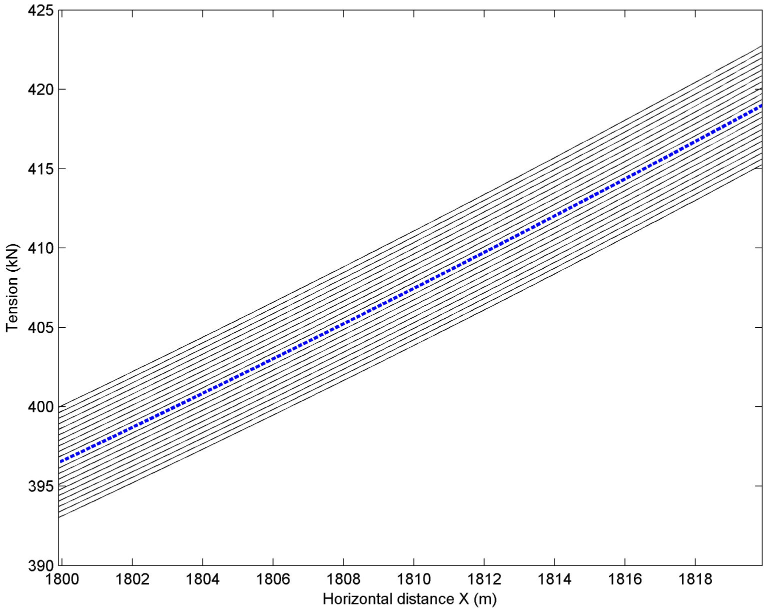


Figure 5.2: The influence of the wave-induced heave motion to the tension measurement.

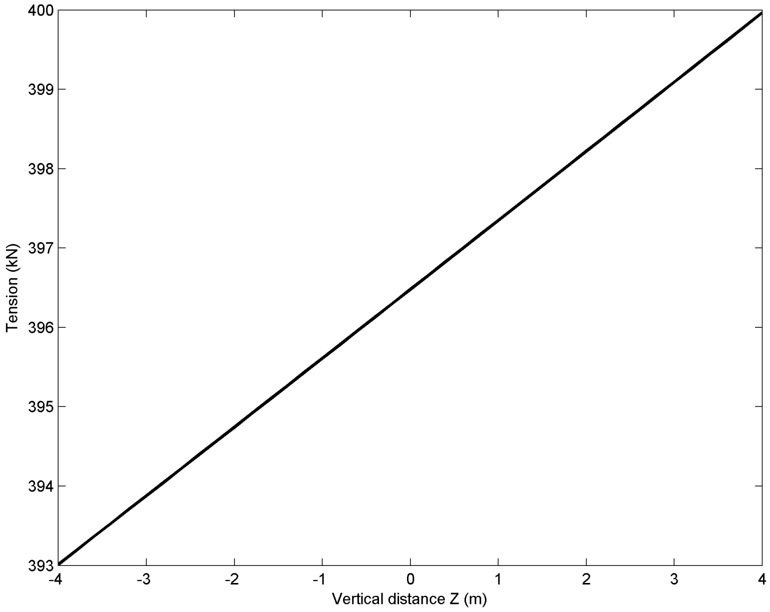


Figure 5.3: The influence of the wave-induced heave motion to the tension measurement when X=1800 m.

pliance. In another words, the slow-varying drift loads do not cause horizontal motion to the moored vessel, and the cables are not made of fibers. This just meets our requirements. The standard deviation of the dynamic tension due to narrow-banded wave frequency vertical motions is given by

$$\sigma_{Td} = M(1+\Delta\tau)\sigma_a + \frac{1}{2}\rho C_d \Delta\tau d H \sigma_{v|v|}, \tag{5.5}$$

where the integrate mass M and C_d are two model coefficients which can be determined from a linear least squares fit with experiments, ρ is the fluid density, d is the effective diameter of the mooring line, H is the water depth, σ_{Td}, σ_a, and $\sigma_{v|v|}$ are the standard deviation of tension, heave acceleration, and heave quadratic velocity. A non-dimensional mean tension $\Delta\tau$ is defined as

$$\Delta\tau = \frac{T_{mean} - T_0}{T_0}, \tag{5.6}$$

where $T_0 = \omega_m H$ is the suspended weight of the mooring at slack current, and T_{mean} is the mean tension.

Simplify the dynamic tension variance σ^2_{Td} with linear approximation, e.g. $\sigma_{v|v|} = 3\sigma^4_v$. The simplified model is given by

$$\sigma^2_{Td} = [M(1+\Delta\tau)]^2\sigma^2_a + \left[3\left(\frac{1}{2}\rho C_d \Delta\tau d H\right)^2 \sigma^2_v + \sqrt{3}\rho M(1+\Delta\tau) C_d \Delta\tau d H \sigma_a\right]^2 \sigma^2_v. \tag{5.7}$$

5.3 Simplified deterministic position reference system

We regard the moored vessel and all the anchors in the same horizontal plane. Assume all anchors are fixed anchor points and the vessel is a mobile unit in a sensor network. Then, we can estimate the position of the moored vessel through range-based localization techniques.

5.3.1 Mooring tension approximation

The system has N mooring lines, where each line is equipped with a tension measurement unit. The basic assumption for the method is that the tension has a quadric relation with horizontal distances, that is,

$$T_i = p_{1i}X^2_i + p_{2i}X_i + p_{3i} \quad i = 1, \cdots, N, \tag{5.8}$$

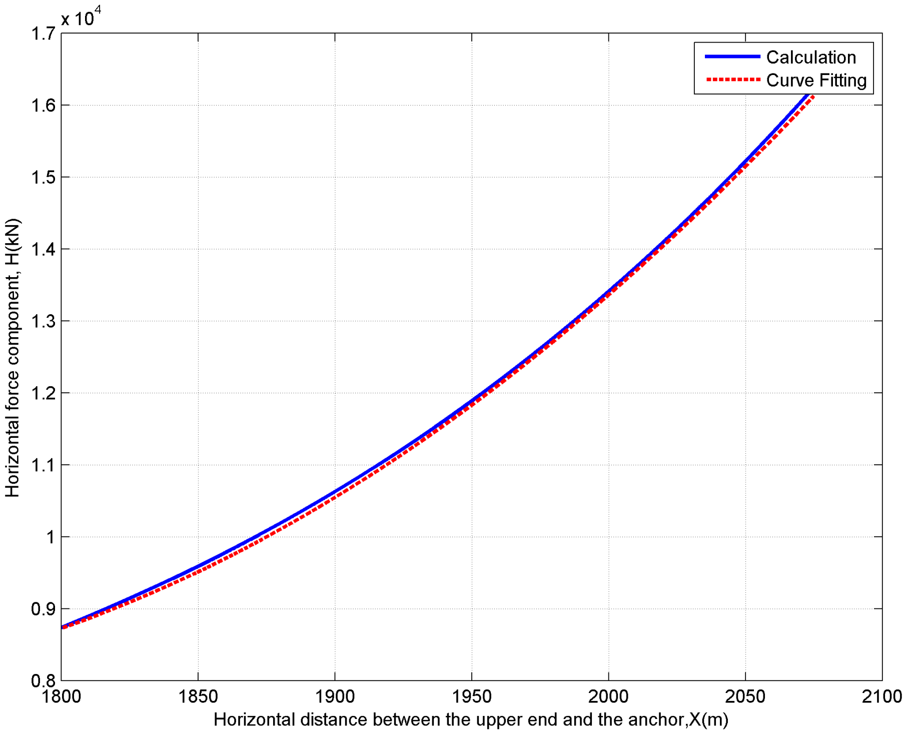


Figure 5.4: The relationship between horizontal distance X_i and tension force T_i.

where $X_i = \sqrt{\bar{x}_i^2 + \bar{y}_i^2}$, and p_1, p_2, and p_3 are coefficients. The coefficients are found from a weighted least square curve fit. The result is shown in Fig. 5.4. We get

$$X_i = \sqrt{\frac{T_i}{p_{1i}} - \frac{p_{3i}}{p_{1i}} + \frac{p_{2i}^2}{4p_{1i}^2}} - \frac{p_{2i}}{2p_{1i}}. \tag{5.9}$$

5.3.2 Position estimation

After knowing the distances between the node and the anchor points, lateration can be utilized to locate the vessel. The horizontal distance between TP estimated position $\hat{\boldsymbol{p}} = [\hat{x}, \hat{y}]^\top$ and the i^{th} anchor is $\hat{X}_i$. The triangular relation is given by

$$\hat{X}_i^2 = (\hat{x} - x_{a_i})^2 + (\hat{y} - y_{a_i})^2, \quad i = 1, \cdots, N. \tag{5.10}$$

There are N equations. Now we subtract the N^{th} equation from the first to $(N-1)^{th}$ equations and write the result in a vector form,

$$\boldsymbol{A}\hat{\boldsymbol{p}} = \boldsymbol{b}, \tag{5.11}$$

where

$$\boldsymbol{A} = \begin{bmatrix} 2(x_{a_1} - x_{a_N}) & 2(y_{a_1} - y_{a_N}) \\ 2(x_{a_2} - x_{a_N}) & 2(y_{a_2} - y_{a_N}) \\ \cdots & \cdots \\ 2(x_{a_{N-1}} - x_{a_N}) & 2(y_{a_{N-1}} - y_{a_N}) \end{bmatrix}, \tag{5.12}$$

and

$$\boldsymbol{b} = \begin{bmatrix} x_{a_1}^2 + y_{a_1}^2 - x_{a_N}^2 - y_{a_N}^2 - X_1^2 + X_N^2 \\ x_{a_1}^2 + y_{a_1}^2 - x_{a_N}^2 - y_{a_N}^2 - X_2^2 + X_N^2 \\ \cdots \\ x_{a_{N-1}}^2 + y_{a_{N-1}}^2 - x_{a_N}^2 - y_{a_N}^2 - X_{N-1}^2 + X_N^2 \end{bmatrix}. \tag{5.13}$$

Using the Moore-Penrose pseudo-inverse, we get the least-squares solution, that is

$$\hat{\boldsymbol{p}} = \boldsymbol{A}^\top (\boldsymbol{A}\boldsymbol{A}^\top)^{-1} \boldsymbol{b}. \tag{5.14}$$

This is a special case of the weighted pseudo-inverse,

$$\hat{\boldsymbol{p}} = \boldsymbol{W}^{-1} \boldsymbol{A}^\top (\boldsymbol{A}\boldsymbol{W}^{-1}\boldsymbol{A}^\top)^{-1} \boldsymbol{b}, \tag{5.15}$$

where the positive definite matrix $\boldsymbol{W} = \boldsymbol{I}$ (Fossen, 2011).

5.4 Sensor network construction

When the anchors are not precisely located, Algorithm in Section 5.3 is not robust enough. Based on the knowledge from collaborative localization position, three (or four) is the minimum number of the anchor nodes to locate another unknown nodes in 2D (or 3D) coordinate (Shang et al., 2004). We can receive the positions of the center of turret (COT) generated from reference frame transformation of the GPS signals, but the anchors are less possible to obtain the GPS signal due to the attenuation in deep water. Therefore, the anchors, in this case, are the uncertain nodes, while the turret is the anchor nodes.

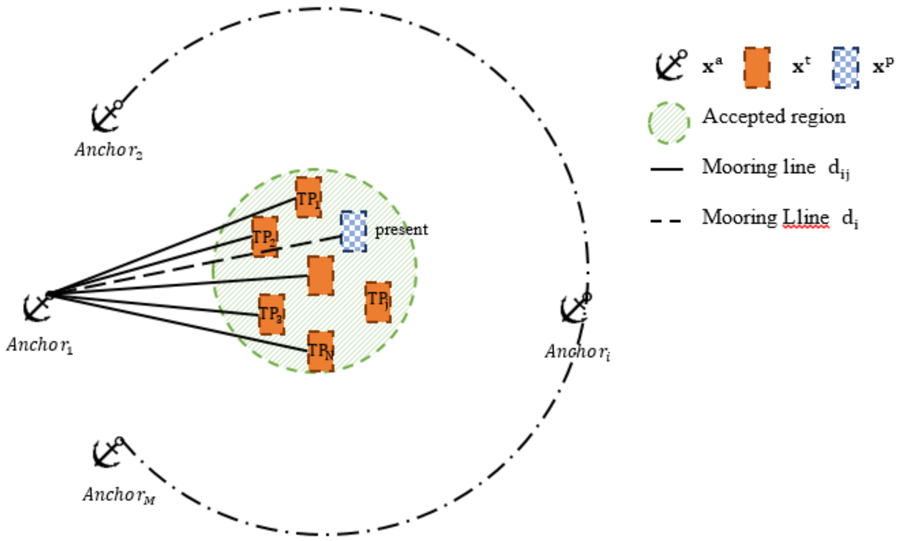


Figure 5.5: Sensor network construction.

The TAPM has one anchor node-the COT, and M unknown nodes, at a given time instant. Consequently, there is no solution in this condition. But we can construct a sensor network with N virtual vessels, which were collected and stored before. The arrangement is shown in Figure 5.5.

In normal sea states, a vessel can stably stay around an equilibrium point where the main environmental loads are balanced by the mooring forces. In these conditions, the thruster-assisted system is only responsible to control the heading and damping. Normally, the equilibrium point in a sea state is uniformly determined by the parameters such as the magnitudes and directions of second-order waves, current, and wind loads. The positions from the observers and the tension measurements are collected and stored when the vessel positioning stably at different equilibrium points separately. When recalling all the previous data, we have a group of N vessels, which are labelled as TP_j in Figure 5.5. The model is shown in Section 5.3. The application of this model is to locate the unknown anchors.

Now we additionally assume the present posref of a TAPM is not available due to failures. The goal is to locate the vessel simultaneously. Then the new sensor network has N anchor nodes and $M+1$ uncertain nodes. The problem is solvable in the sense of sensor network. This model is derived in Section 5.5.

5.4.1 Problem statement

Each mooring line is connected to the turret at a specific fairlead. The distance between the fairlead and the COT is r. The turret can rotate about a vertical axis at the COT.

For the sake of simplification, we disregard the water depth and hereafter mention 2D sensor network. The recent position of the turret is $\boldsymbol{x}^p = [x^p, y^p]^\top$ and $\boldsymbol{\eta} = [\boldsymbol{x}^{p\top}, 0]^\top$. The position of the i^{th} anchor is defined as

$$\boldsymbol{x}_i^a = [x_i^a, y_i^a]^\top, \quad i = 1, \cdots, M. \tag{5.16}$$

The position of the j^{th} turret is defined as

$$\boldsymbol{x}_j^t = [x_j^t, y_j^t]^\top, \quad j = 1, \cdots, N. \tag{5.17}$$

Define two vectors contain all the anchor position $\boldsymbol{x}^a$ and turret position $\boldsymbol{x}^t$ as,

$$\boldsymbol{x}^a = \left[\boldsymbol{x}_1^{a\top} \cdots \boldsymbol{x}_m^{a\top}\right]^\top \quad and \quad \boldsymbol{x}^t = \left[\boldsymbol{x}_1^{t\top} \cdots \boldsymbol{x}_n^{t\top}\right]^\top. \tag{5.18}$$

The noisy tension measurement of the present TAPM is given by

$$T_i = T_i^0 + v_p, \quad i = 1, 2, \cdots, M, \tag{5.19}$$

where T_i and T_i^0 are the noisy and noiseless tension measurement of the i^{th} anchor from the present vessel, and v_p is the Gaussian white noise in present model. Notice we introduce two noise variances. This is because the variance of the two noises are different. We will introduce this in Section 5.5.

We assume LOS estimate model (Skjetne et al., 2011), then the LOS question is defined as

$$T_{ij} = T_{ij}^0 + v_t, \quad i = 1, 2, \cdots, M, \quad j = 1, 2, \cdots, N, \tag{5.20}$$

where T_{ij} and T_{ij}^0 are the noisy and noiseless tension measurement of the i^{th} anchor from the j^{th} vessel, and v_t is the Gaussian white noise in the tension measurements.

Assumption 5.4 (LOS assumption)**.** *For the sake of simplicity, the current-induced tension variance is disregarded. Then the tension-range mapping is simplified as*

$$T_{ij}^0 = f_{ij}(X_{ij}) \quad and \quad T_i^0 = f_i(X_i), \tag{5.21}$$

where X_{ij} is the distance between the i^{th} anchor and its fairlead of the j^{th} turret, and X_i is the distance between the i^{th} anchor and its fairlead of the present vessel.

The localization algorithm is discussed and verified in Ren et al. (2015b). However, the application of the algorithm is quite limited, since the algorithm heavily replies on the precise position of the anchors, and the performance is not satisfied when the anchor positions have full extents of uncertainty.

The problem is to locate the uncertain anchors $\boldsymbol{x}^a$ with the noisy GPS signals $\boldsymbol{x}^t$ at the COT, as well the tension measurements. We assume the anchors have fixed but uncertain positions during the localization. Additionally, there is no information exchange among the anchors. The only information is the best estimated anchor positions during installation.

Theorem 5.5 (Variance of range measurement for the sensor network)**.** *We assume the GPS noise, the tension cell noise σ_{Tn}, and dynamic tensionσ_{Td} are independent, the sum of the variance of the tension measurement noise σ_T is given by*

$$\sigma_T^2 = \sigma_{Tn}^2 + \sigma_{Td}^2. \tag{5.22}$$

Therefore, the sum variance of the noise in distance mapping is

$$\sigma_d^2(v_t) = \sigma_{X_{ij}}^2 = \frac{\sigma_T^2}{k_{ij}^2}, \tag{5.23}$$

$$\sigma_d^2(v_p) = \sigma_{X_i}^2 = \frac{\sigma_T^2}{k_i^2} + \sigma_{GPS}^2, \tag{5.24}$$

where σ_{GPS} is the variance of the GPS noise, k_i=k_{ij} is the stiffness, which is given by

$$k_{ij} = \frac{\partial T_{ij}}{\partial X_{ij}} \simeq constant. \tag{5.25}$$

The stiffness k_{ij} is divided into two constants, as shown in the Figure 9.1. From Figure 9.1, we can find mooring line performs as a two-segment linear spring. If the horizontal distance is smaller than 1970 meters, the stiffness is a constant. If the horizontal distance is larger than 2030 meters, the stiffness is another constant. Since the TAPM is always running around the equilibrium point, which is determined by the environmental loads, the stiffness will not change for a specific virtual vessel. In this case, k is in the level of 10^5, the influence from tension noise is quite small.

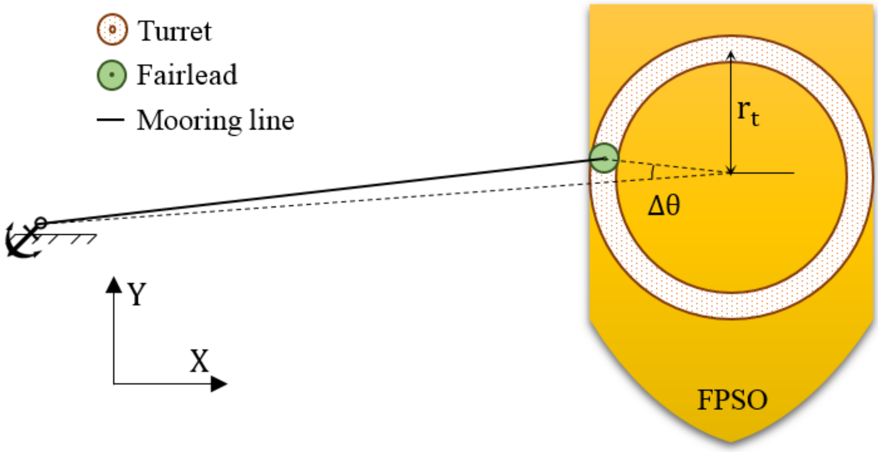


Figure 5.6: The fairlead arrangement.

5.4.2 Influence of the fairleads

Previous works always assume all the mooring line TP are connect at the COT. However, the fact is that they are connected to the fairleads of the turret, and normally the influence from fairleads cannot be disregarded. The arrangement is illustrated in Figure 5.6.

We first assume the friction between the turret and the vessel is not remarkable. The turret can rotate with the torque aroused from horizontal components of the tension forces. Additionally, the magnitude of the turret radius is much smaller than the lengths of the mooring lines. The arrangement is represented in Figure 5.6. Since the angle $\Delta\theta$ is a small angle, we assume the COT is located in the extension line to the fairlead and the corresponding anchor with a certain distance r_t to the fairlead.

Theorem 5.6 (Influence of range measurement due to the turret)**.** *The mathematical expression is given by*

$$d_{ij} = f_i^{-1}(T_i) + r_t + v_t \quad and \quad d_i = f_i^{-1}(T_i) + r_t + v_p, \tag{5.26}$$

where d_{ij} is the distance between the i^{th} anchor and the j^{th} COT. d_i is the distance between the i^{th} anchor and the present COT

Here we define two vectors which contains all d_{ij} and d_i respectively. They are

$$\boldsymbol{d_{IJ}} = [d_{11}, d_{12}, \cdots, d_{1N}, d_{21}, d_{22}, \cdots, d_{2N}, \cdots, d_{M1}, d_{M2}, \cdots, d_{MN}]^{\top}, \tag{5.27}$$

$$\boldsymbol{d_I} = [d_1, d_2, \cdots, d_M]^{\top}. \tag{5.28}$$

The position of the fairleads are given by

$$\boldsymbol{x}_{f,i} = \boldsymbol{x}_{COT} + \begin{bmatrix} r_t cos(\gamma_i) \\ r_t sin(\gamma_i) \end{bmatrix}, \quad i = 1, \cdots, m \tag{5.29}$$

where $\boldsymbol{x}_{COT}$ is the position of the COT, $\boldsymbol{x}_{f,i}$ is the position of the i^{th} fairlead, r_t is the radius of the circles where the fairleads locate, and γ_i is the angle of the i^{th} fairlead compare to the reference angle.

5.5 Simultaneous localization and mapping

The only information we have is the heading and estimated anchor positions during installation. EKF and GPS signals are used to find the best estimation in this paper. Particle filter and other FastSLAM approaches are left for future researches.

5.5.1 Observation model

Assumption 5.7 (Constant heading variance)**.** *The heading is almost a constant value, that is* $\dot{\psi} = 0$.

Regarding the sensor network built in Section 5.4, the stored vessel position data, the stored tension data, real-time heading, and the online tension measurements are available. To simplify the representation, we employ the range measurements directly after the tension mapping. The CPM is given by

$$\dot{\boldsymbol{\eta}}_w = \boldsymbol{A}_w \boldsymbol{\eta}_w, \tag{5.30}$$

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu}, \tag{5.31}$$

$$\dot{\boldsymbol{b}} = -\boldsymbol{T}_b^{-1}\boldsymbol{b} + \boldsymbol{E}_b \boldsymbol{w}_b, \tag{5.32}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{D}\boldsymbol{\nu} = \boldsymbol{R}(\psi)^\top \boldsymbol{b} - \boldsymbol{R}^\top(\boldsymbol{\psi})\boldsymbol{G}\boldsymbol{\eta} + \boldsymbol{\tau_c}. \tag{5.33}$$

The anchors have fixed positions, but a noise is need to balance the distance between the estimated and the real position.This is similar to Eq. 5.32 which is applied to simulate the slow-varing loads. The discrete form is given by

$$\boldsymbol{x}_i^a(k+1) = \boldsymbol{x}_i^a(k) + \boldsymbol{w}_i, \tag{5.34}$$

where $\boldsymbol{x}^p$ is the first two elements in $\boldsymbol{\eta}+\boldsymbol{C}_w\boldsymbol{\eta}_w$. The continuous form is given by

$$\dot{\boldsymbol{x}}^a = \boldsymbol{E}_a\boldsymbol{\omega}_a. \tag{5.35}$$

Theorem 5.8 (State space equation for a TAPM with only tension measurements.). *The EKF is based on the nonlinear model*

$$\begin{aligned}\dot{\boldsymbol{x}} =& \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{E}\boldsymbol{w},\\ \boldsymbol{y} =& \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{v},\end{aligned} \tag{5.36}$$

where

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{A}_w\boldsymbol{\eta}_w \\ \boldsymbol{R}(\psi)\boldsymbol{v} \\ -\boldsymbol{T}_b^{-1}\boldsymbol{b} \\ -\boldsymbol{M}^{-1}\boldsymbol{D}\boldsymbol{v} + \boldsymbol{M}^{-1}\boldsymbol{R}(\psi)^\top\boldsymbol{b} - \boldsymbol{M}^{-1}\boldsymbol{R}^\top(\boldsymbol{\psi})\boldsymbol{G}\boldsymbol{\eta} + \boldsymbol{M}^{-1}\boldsymbol{\tau}_{\boldsymbol{c}} \\ \boldsymbol{zero}_{3\times(15+2M+2N)} \\ \boldsymbol{zero}_{3\times(15+2M+2N)} \end{bmatrix}, \tag{5.37}$$

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{0}_{6\times3} \\ \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} \\ \boldsymbol{M}^{-1} \\ \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} \end{bmatrix}, \quad \boldsymbol{E} = \begin{bmatrix} \boldsymbol{E}_w & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{E}_b & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{E}_a \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \end{bmatrix}, \quad \boldsymbol{h}(\boldsymbol{x}) = \begin{bmatrix} diag\{0,0,1\}(\boldsymbol{\eta}+\boldsymbol{C}_w\boldsymbol{\eta}) \\ \boldsymbol{x}^t \\ \boldsymbol{G}_I(\boldsymbol{x}^a,\boldsymbol{\eta}) \\ \boldsymbol{G}_{IJ}(\boldsymbol{x}^a,\boldsymbol{x}^t) \end{bmatrix}, \tag{5.38}$$

where the state vector is $\boldsymbol{x}= [\boldsymbol{\eta}_w^\top, \boldsymbol{\eta}^\top, \boldsymbol{b}^\top, \boldsymbol{v}^\top, \boldsymbol{x^a}^\top, \boldsymbol{x^t}^\top]^\top$, $\boldsymbol{\eta} = [\boldsymbol{x}^{p\top}, 0]^\top$, *the measurement vector is* $\boldsymbol{y} = [\psi, \boldsymbol{x}^{t\top}, \boldsymbol{d}_I{}^\top, \boldsymbol{d}_{IJ}{}^\top]^\top$, *and* $\boldsymbol{w} = [w_w^\top, w_b^\top, w_a^\top]^\top$. *The distance mapping functions are given by*

$$\boldsymbol{d}_I = \boldsymbol{G}_I(\boldsymbol{x}^a, \boldsymbol{x}^p), \tag{5.39}$$

$$\boldsymbol{d}_{IJ} = \boldsymbol{G}_{IJ}(\boldsymbol{x}^a, \boldsymbol{x}^t), \tag{5.40}$$

where the estimated distance is given by

$$\hat{d}_i = g_i(\hat{\boldsymbol{x}}_i^a, \hat{\boldsymbol{x}}^p) = \sqrt{(\hat{x}_i^a - \hat{x}^p)^2 + (\hat{x}_i^a - \hat{x}^p)}, \tag{5.41}$$

$$\hat{d}_{ij} = g_{ij}(\hat{\boldsymbol{x}}_i^a, \hat{\boldsymbol{x}}_j^t) = \sqrt{(\hat{x}_i^a - \hat{x}_j^t)^2 + (\hat{x}_i^a - \hat{x}_j^t)}. \tag{5.42}$$

The Jacobian matrix is given by

$$\boldsymbol{H} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial \psi}{\partial \boldsymbol{\xi}} & \frac{\partial \psi}{\partial \boldsymbol{\eta}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \\ \frac{\partial \boldsymbol{d_I}}{\partial \boldsymbol{\xi}} & \frac{\partial \boldsymbol{d_I}}{\partial \boldsymbol{\eta}} & \boldsymbol{0} & \boldsymbol{0} & \frac{\partial \boldsymbol{d_I}}{\partial \boldsymbol{x}^a} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \frac{\partial \boldsymbol{d_{IJ}}}{\partial \boldsymbol{x}^a} & \frac{\partial \boldsymbol{d_{IJ}}}{\partial \boldsymbol{x}^t} \end{bmatrix}. \tag{5.43}$$

The elements of the Jacobian matrix are as follows.

$$\frac{\partial \psi}{\partial \boldsymbol{\xi}} = \frac{\partial \psi}{\partial \boldsymbol{\eta}} = \boldsymbol{0} \tag{5.44}$$

$$\frac{\partial d_i}{\partial \xi_k} = \frac{[(\boldsymbol{C}_w\boldsymbol{\xi})_i - x_1^a]\boldsymbol{C}_w(1,k) + (\boldsymbol{C}_w\boldsymbol{\xi})_2 - y_i^a]\boldsymbol{C}_w(2,k)}{d_i}, \qquad \frac{\partial \boldsymbol{d_I}}{\partial \boldsymbol{\xi}} = \begin{bmatrix} \frac{(\bar{\boldsymbol{C}}_w\boldsymbol{\xi} + \bar{\boldsymbol{\eta}} - \boldsymbol{x^a})^\top}{d_1} \\ \vdots \\ \frac{(\bar{\boldsymbol{C}}_w\boldsymbol{\xi} + \bar{\boldsymbol{\eta}} - \boldsymbol{x^a})^\top}{d_M} \end{bmatrix} \bar{\boldsymbol{C}}_w, \tag{5.45}$$

where $\bar{\boldsymbol{C}}_w$ is the first two rows of $\boldsymbol{C}_w$, and $\bar{\boldsymbol{\eta}} = [x, y]^\top$.

$$\frac{\partial d_i}{\partial \boldsymbol{x}_i^a} = -\frac{1}{d_i}\begin{bmatrix} (\bar{\boldsymbol{\eta}} + \bar{\boldsymbol{C}}_w\boldsymbol{\xi}) - \boldsymbol{x}^a \\ 0 \end{bmatrix}, \qquad \frac{\partial d_I}{\partial \boldsymbol{x}^a} = \begin{bmatrix} \frac{\partial d_1}{\partial \boldsymbol{x}_1^a} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \frac{\partial d_2}{\partial \boldsymbol{x}_2^a} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \frac{\partial d_M}{\partial \boldsymbol{x}_M^a} \end{bmatrix}. \tag{5.46}$$

$$\frac{\partial d_i}{\partial \boldsymbol{\eta}} = \begin{bmatrix} \frac{1}{d_1} & \frac{1}{d_2} & \cdots & \frac{1}{d_M} \end{bmatrix}^\top (\bar{\boldsymbol{C}}_w\boldsymbol{\xi})^\top \bar{\boldsymbol{C}}_w. \tag{5.47}$$

$$\frac{\partial \boldsymbol{d_{IJ}}}{\partial \boldsymbol{x^a}} = \begin{bmatrix}
\frac{x_1^a - x_1^t}{d_{11}} & \frac{y_1^a - y_1^t}{d_{11}} & & & & & \\
& & \frac{x_2^a - x_1^t}{d_{21}} & \frac{y_2^a - y_1^t}{d_{21}} & & & \\
& & & & \ddots & & \\
& & & & & \frac{x_M^a - x_1^t}{d_{M1}} & \frac{y_M^a - y_1^t}{d_{M1}} \\
\frac{x_1^a - x_2^t}{d_{12}} & \frac{y_1^a - y_2^t}{d_{12}} & & & & & \\
& & \frac{x_2^a - x_2^t}{d_{22}} & \frac{y_2^a - y_2^t}{d_{22}} & & & \\
& & & & \ddots & & \\
& & & & & \frac{x_M^a - x_2^t}{d_{M2}} & \frac{y_M^a - y_2^t}{d_{M2}} \\
& & & \vdots & & & \\
\frac{x_1^a - x_N^t}{d_{1N}} & \frac{y_1^a - y_N^t}{d_{1N}} & & & & & \\
& & \frac{x_2^a - x_N^t}{d_{2N}} & \frac{y_N^a - y_2^t}{d_{2N}} & & & \\
& & & & \ddots & & \\
& & & & & \frac{x_M^a - x_N^t}{d_{MN}} & \frac{y_M^a - y_N^t}{d_{MN}}
\end{bmatrix}. \tag{5.48}$$

$$\frac{\partial \boldsymbol{d_{IJ}}}{\partial \boldsymbol{x^t}} = -\frac{\partial \boldsymbol{d_{IJ}}}{\partial \boldsymbol{x^a}}. \tag{5.49}$$

Remark: We notice that the matrix $\frac{\partial \boldsymbol{d_{IJ}}}{\partial \boldsymbol{x^a}}$ is made up of N almost diagonal formed matrices. To write it in this form has an advantage that every block is related to one anchor. In another words, we can choose the interested anchor as we want.

Remark: Based on the slow motion assumption, the DP vessel does not have horizontal displacement. Therefore, assume $\frac{\partial \boldsymbol{d}_i}{\partial \boldsymbol{x}_j^t} = 0$ can enhance the computation speed.
There are two additional simplified forms for the observation model.

Lemma 5.9 (Simplified 1. Known anchor positions and GPS failure)**.** *When the anchor positions are known, the goal is to estimate the upper vessel $\boldsymbol{x}^p$ with tension measurements. The state vector hence is $\boldsymbol{x} = [\boldsymbol{\xi}^\top, \boldsymbol{\eta}^\top, \boldsymbol{b}^\top, \boldsymbol{v}^\top]^\top$, the observation vector is $\boldsymbol{y}$= $[\psi, \boldsymbol{d_I}^\top]^\top$, and the estimated distance is given by*

$$\hat{d}_i = g_{ij}(\boldsymbol{x}_i^a, \hat{\boldsymbol{x}}^p). \tag{5.50}$$

Lemma 5.10 (Simplified 2.Unknown anchors)**.** *Another simplified form is to only estimated the anchors. The state vector is* $\boldsymbol{x} = [\boldsymbol{\xi}^\top, \boldsymbol{\eta}^\top, \boldsymbol{b}^\top, \boldsymbol{v}^\top, \boldsymbol{x}^{\boldsymbol{a}\top}, \boldsymbol{x}^{t\top}]^\top$*, the observation vector is* $\boldsymbol{y} = [\boldsymbol{x}^{t\top}, \boldsymbol{d}_{IJ}{}^\top]^\top$*, and the estimated distance is given by*

$$\hat{d}_{ij} = g_{ij}(\hat{\boldsymbol{x}}_i^a, \boldsymbol{x}_j^t). \tag{5.51}$$

5.5.2 Extended Kalman filter

The EKF is tabulated in Table 5.1. In the table, $\boldsymbol{K}(k)$ is the Kalman gain matrix, $\hat{\boldsymbol{P}}(k)$ represents the error covariance matrix, $\boldsymbol{Q} = \boldsymbol{Q}^\top$ and $\boldsymbol{R} = \boldsymbol{R}^\top$ are covariance matrices,

Table 5.1: Discrete-time extended Kalman filter.

Design matrics	$\boldsymbol{Q} = \boldsymbol{Q}^\top > 0$, $\boldsymbol{R} = \boldsymbol{R}^\top$
Initial conditions	$\bar{\boldsymbol{x}}(0) = \boldsymbol{x}_0$
Kalman gain matrix	$\boldsymbol{K}(k) = \bar{\boldsymbol{P}}\boldsymbol{H}^\top(k)[\boldsymbol{H}(k)\bar{\boldsymbol{P}}(k)\boldsymbol{H}^\top(k) + \boldsymbol{R}(k)]^{-1}$
State estimate update	$\hat{\boldsymbol{x}}(k) = \bar{\boldsymbol{x}}(k) + \boldsymbol{K}(k)[\boldsymbol{y}(k) - \boldsymbol{g}(\bar{\boldsymbol{x}}(k))]$
Error covariance update	$\hat{\boldsymbol{P}}(k) = [\boldsymbol{I} - \boldsymbol{K}(k)\boldsymbol{H}(k)]\bar{\boldsymbol{P}}(k)[\boldsymbol{I} - \boldsymbol{K}(k)\boldsymbol{H}(k)]^\top + \boldsymbol{K}(k)\boldsymbol{R}(k)\boldsymbol{K}^\top(k)$
State estimate propagation	$\bar{\boldsymbol{x}}(k+1) = \boldsymbol{f}_k(\hat{\boldsymbol{x}}(k), \boldsymbol{u}_k)$
Error estimate propagation	$\bar{\boldsymbol{P}}(k+1) = \boldsymbol{\Phi}(k)\hat{\boldsymbol{P}}(k)\boldsymbol{\Phi}^\top(k) + \boldsymbol{\Gamma}(k)\boldsymbol{Q}(k)\boldsymbol{\Gamma}^\top(k)$

$$\boldsymbol{f}_k(\hat{\boldsymbol{x}}(k), \boldsymbol{u}_k) = \hat{\boldsymbol{x}}(k) + T[\boldsymbol{f}\hat{\boldsymbol{x}}(k) + \boldsymbol{B}\boldsymbol{u}(k)], \tag{5.52}$$

$$\bar{\boldsymbol{\Phi}}(k) = \boldsymbol{I} + T\frac{\partial \boldsymbol{f}_k(\boldsymbol{x}(k), \boldsymbol{u}(k))}{\partial \boldsymbol{x}(k)}\Big|_{\boldsymbol{x}(k)=\hat{\boldsymbol{x}}(k)}, \tag{5.53}$$

$$\boldsymbol{\Gamma} = T\boldsymbol{E}. \tag{5.54}$$

Remark: To enhance the computation speed, the vector $\boldsymbol{x}^t(k+1)$, in state estimation propagation step, can be written as

$$\bar{\boldsymbol{x}}^t(k+1) = \hat{\boldsymbol{x}}^t(k+1). \tag{5.55}$$

This is because the following position information has been collected and stored after filters. There is no need to filter it twice. Therefore, the element $\boldsymbol{x}^t$ in the state vector $\boldsymbol{x}$ can be removed. The state estimation propagation process then becomes open-looped.

It is easy to find the state estimation propagation is different from normal EKF. It is an open-loop process which updates with the stored data. Instead of $\boldsymbol{\eta}$, we applied $\boldsymbol{\eta} + \boldsymbol{\eta}_w$ in the propagation. This is because the wave-induced motion $\boldsymbol{\eta}_w$ really happens and really influences the tension measurements.

Chapter 6

Fault Tolerant Control and Line Breakage Detection

This chapter reviews the basic definitions of the fault-tolerant control theory. Additionally, two mooring line breakage detection schemes are proposed.

6.1 Fault-tolerant control

fault-tolerant control (FTC) allows maintaining the current performance close to desirable one and preserves stability conditions in the presence of component and/or instrument faults (Blanke et al., 2006). At first, two terms, fault and failure, are defined.

Definition 6.1 (Fault)**.** *A fault means there exists a perturbation or variance in the control structure or the control parameter. It can be coped with the FTC.*

Definition 6.2 (Failure)**.** *A failure presents a system or one of its component lost its capacities. At this moment, the system or the component needs to shut off.*

Remark: Comparing with failure, the notion of a fault only means that the system performs in a undesirable way but it can still be operational through fault-tolerant control. However, failure addressed irrecoverable inability of a system or its components to accomplish it functions (Blanke et al., 2006).

FTC aims to prevent from developing a fault into failures which may arouse safety hazards, see Figure 6.1. There are three categories of faults, which are plant faults, sensor faults and

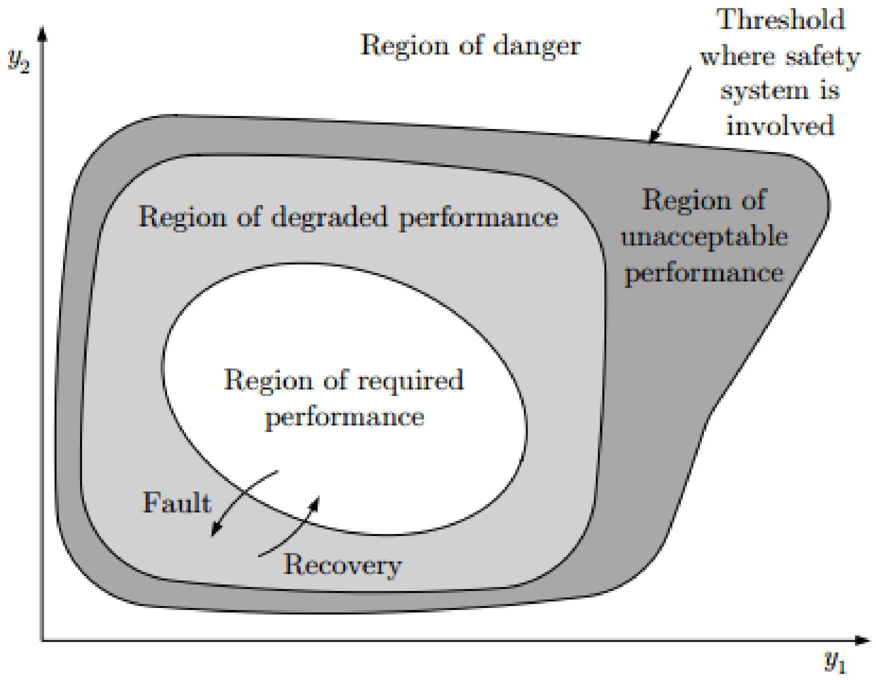

y_2
Region of danger
Threshold where safety system is involved
Region of degraded performance
Region of unacceptable performance
Region of required performance
Fault
Recovery
y_1

Figure 6.1: Region of required and degraded performance parameter space (Blanke et al., 2006).

actuator faults respectively. As shown in Fig. 6.2, different faults happen in distinct section of the system.

Definition 6.3 (Plant fault)**.** *Such faults change the dybanucak I/O properties of the system.*

Definition 6.4 (Sensor fault)**.** *The plant properties are not addicted , but the sensor reading have substantial errors.*

Definition 6.5 (Actuator fault)**.** *The plant properties are not affected, but the influence of the controller on the plant is interrupted or modified.*

The set of faults donates $\mathscr{F}$, where fault $f \epsilon \mathscr{F}$. f_0 represents the faultless case.
The behavior $\mathscr{B}$ is defined as a set of all the possible I/O pairs which may occur for a specific plant. Mathematically, the behavior is given by

$$\mathscr{B} = (u, y) : y = k_s u \tag{6.1}$$

where k_s is the static gain, $\mathscr{B} \subset \mathscr{U} \times \mathscr{Y}$.

Definition 6.6 (Failure mode)**.** *Failure mode means the particular way in which a failure can occur.*

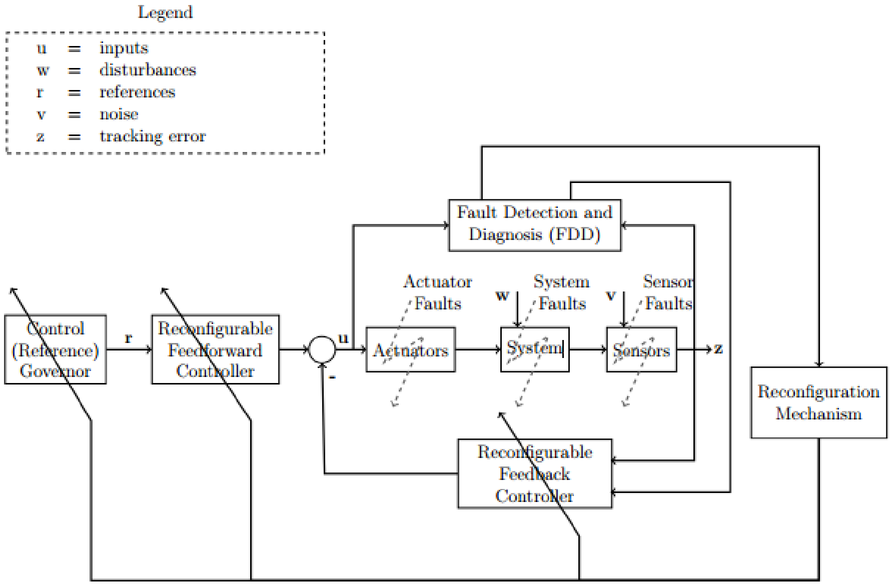

Legend
u = inputs
w = disturbances
r = references
v = noise
z = tracking error
Fault Detection and Diagnosis (FDD)
Control (Reference) Governor
r
Reconfigurable Feedforward Controller
u
-
Actuator Faults
System Faults
Sensor Faults
w
v
Actuators
System
Sensors
z
Reconfiguration Mechanism
Reconfigurable Feedback Controller

Figure 6.2: The Structure of the fault tolerant control scheme (Stoican and Olaru, 2013).

Definition 6.7 (Failure effect)**.** *Failure effect is the consequence of a failure mode on the operation function, or status of an item.*

6.2 The Structure of Fault-Tolerant Control System

For a FTC system, the two main parts are the fault diagnosis and the control re-design.

Definition 6.8 (Fault diagnosis)**.** *The existence of fault has to be detected and the faults have to be identified. In short, For a given I/O pair (U,Y), find the fault f.*

Three subprocesses of fault diagnosis are fault detection, fault isolation, and fault estimation. Fault detection can detect if the fault happens. Fault isolation locates where is the fault. Fault estimation evaluates the magnitude of the fault.
fault detection and isolation (FDI) techniques can be categories as the model-based FDI and the data-based FDI. The model-based FDI detects the occurrence of a fault based on some models of the system. The most recently applied methods are

- State estimation: observer and Kalman filter,
- Parameter estimation,
- Simultaneous state/parameter estimation, and

- Parity space:input-output method and state-space-based method.

In this thesis, the FDI mostly based on residual signal, which is one of the easiest methods to apply.

Definition 6.9 (Residual)**.** *Fault information carrying signals, based on deviation between measurement and model based computation, that is*

$$r(t) = y(t) - \hat{y}(t). \tag{6.2}$$

There are three important residual construction methods, they are, measurement equation-based residual, observer-based residual, and receding observation window-based residual.

Definition 6.10 (Control re-design)**.** *The controller has to be adapted to the faulty situations so that the overall system continuous to satisfy its goal.*

The two subprocesses are the fault accommodation and the control reconfiguration.

6.3 Problem statement

6.3.1 Passive Fault Tolerant Control

Passive fault tolerant control consists of the design of a control that will be robust against a set of predefined faults.

6.3.2 Active fault tolerant control

Active fault tolerant control reacts to a detected fault and reconfigures the control actions so that the stability and the performances can be verified.

6.4 Mooring line failure based on supervisory control

In supervisory control, the controllers are pre-designed, and the fault diagnosis is conducted through online monitoring. Quantitative model-based methods can be divided into three main categories: state estimation, parameter estimation, and parity space method (Stoican and Olaru, 2013). In this paper, a state estimation method is employed.

$$
\begin{array}{rl}
a_1: & T_1 = g_t(u_1, u_2, \cdots, u_k) \\
a_2: & T_2 = g_t(u_1, u_2, \cdots, u_k) \\
a_3: & T_3 = g_t(u_1, u_2, \cdots, u_k) \\
c_1: & \begin{aligned} \boldsymbol{M}\dot{\boldsymbol{\nu}} =& \boldsymbol{H}_x y \boldsymbol{T}[T_1, T_2, T_3]^T + [g_w^x(\nu_w) g_w^y(\nu_w)]^T + \sum_{j=1}^{n} \boldsymbol{A}_{mo}^{xy}(\boldsymbol{p}, \psi) \boldsymbol{A}_{mo}^{xy}(\boldsymbol{T}_{moi}) \\ & - \boldsymbol{D}[\boldsymbol{\nu}, \dot{\psi}]^T + \boldsymbol{H}_{xy} \boldsymbol{T}_{wave} \end{aligned} \\
c_2: & \boldsymbol{I}_{zz}\ddot{\psi} = \boldsymbol{H}_\psi \boldsymbol{T}[T_1, T_2, T_3]^T g_w^\psi(\nu_w) + \sum_{j=1}^{n} \boldsymbol{A}_{mo}^\psi(\boldsymbol{p}, \psi) \boldsymbol{H}_{mo}^\psi(\boldsymbol{T}_{moi}) + \boldsymbol{H}_\psi \boldsymbol{T}_{wave} \\
c_3: & \dot{\boldsymbol{p}} = \boldsymbol{A}_{ve}(\psi)\boldsymbol{\nu} + \boldsymbol{v}_c \\
c_4: & \boldsymbol{p}_{G1} = \boldsymbol{p} + \boldsymbol{R}(\phi, \theta, \psi)\boldsymbol{l}_{G1} \\
c_5: & \boldsymbol{p}_{G2} = \boldsymbol{p} + \boldsymbol{R}(\phi, \theta, \psi)\boldsymbol{l}_{G2} \\
c_6: & \boldsymbol{p}_{H1} = \boldsymbol{p} + \boldsymbol{R}(\phi, \theta, \psi)\boldsymbol{l}_{H1} \\
c_{2i+5}: & \boldsymbol{T}_{moi} = g_{mo}(\boldsymbol{p}, \psi, \boldsymbol{T}_{mbi}) \\
c_{2i+6}: & \boldsymbol{T}_{moi} = g_{mo}(\boldsymbol{p}, \psi) \\
d_1: & \dot{\boldsymbol{\nu}} = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\nu} \\
d_2: & \dot{\boldsymbol{p}} = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{p} \\
d_3: & \dot{\psi} = \frac{\mathrm{d}}{\mathrm{d}t}\psi \\
d_4: & \ddot{\psi} = \frac{\mathrm{d}}{\mathrm{d}t}\dot{\psi} \\
m_1 \ldots m_3: & h_{1\cdots 3} = \psi \\
m_4: & \boldsymbol{p}_{G1}^m = \boldsymbol{p}_{G1} \\
m_5: & \boldsymbol{p}_{G2}^m = \boldsymbol{p}_{G2} \\
m_6: & \boldsymbol{p}_{H1}^m = \boldsymbol{p}_{H1} \\
m_7 \ldots m_9: & q_{1\cdots 3} = [z\phi\theta] \\
m_{10}: & \boldsymbol{v}_m = \boldsymbol{v} \\
m_{11,12}: & \boldsymbol{w}_{m1,m2} = \boldsymbol{v}_w \\
m_{13}: & \boldsymbol{c}_m = \boldsymbol{v}_c \\
m_{14}: & \boldsymbol{T}_{momi} = \boldsymbol{T}_{moi}
\end{array}
$$

The supervisory controller consists of a bank of candidate controllers designed such that each of them can control the system during a specific mooring line failure. A supervisor uses real-time input and output data, and prior information about the system, to generate a switching signal that determine which controller to use at what time. In this methodology, a separate controller that provides satisfactory performance for all the possible mooring line failures, should be designed and included in the bank of controllers.

For each failure mode in the mooring line system, an individual observer is also designed. The resulting set of observers forms a bank which runs in parallel. At each sampling instant a nonlinear function of the measurement residuals are used to compute a performance signal for each observer. The rationale is that the most accurate estimator will have the best performance signal. In each sample time, the performance signals are assessed to decide

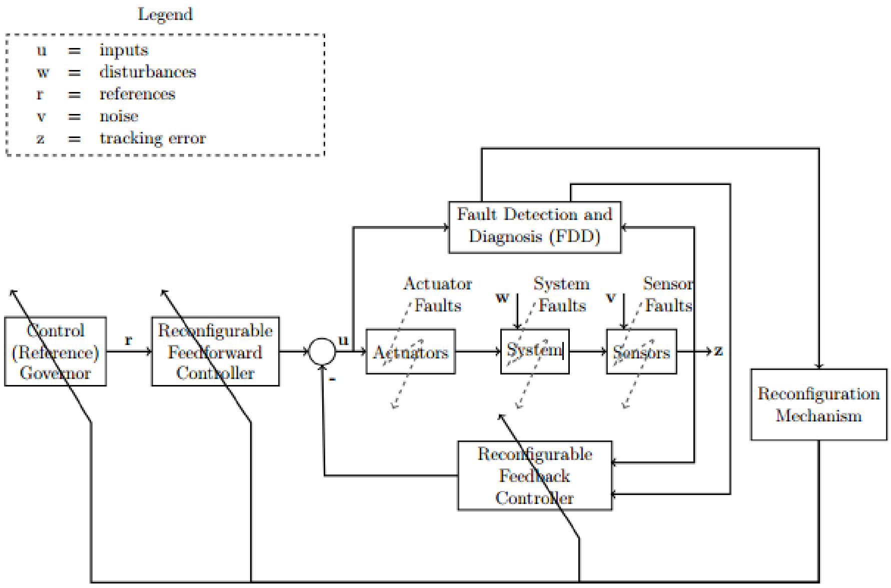

Legend
u = inputs
w = disturbances
r = references
v = noise
z = tracking error
Fault Detection and Diagnosis (FDD)
Actuator Faults
System Faults
Sensor Faults
w
v
Control (Reference) Governor
r
Reconfigurable Feedforward Controller
u
-
Actuators
System
Sensors
z
Reconfiguration Mechanism
Reconfigurable Feedback Controller

Figure 6.3: The Structure of the fault tolerant control scheme.

which controller to select. Fig. 6.4 shows the architecture of the multi-estimator based supervisory control. This is built upon a well-known heuristic idea of certainty equivalence, where the principle of the switching logic is to find the observer with closest output to the process output (Hespanha, 2001). When a line break occurs, the corresponding observer in the bank will perform better in estimating the states of the system; hence, the supervisor will detect the line breakage, by comparing the performance signal of each observer. Then the corresponding controller is activated in the feedback loop.

6.4.1 Mooring forces approximation

The mooring force in the observers are approximated by a second order polynomial to get an approximate relationship with the horizontal distance between the anchor and the turret TPs, that is

$$\tau_{H,m_i} \approx P_1 X_i^2 + P_2 X_i + P_3, \quad i = 1 \cdots N \tag{6.3}$$

where X_i is the horizontal distance between the i^{th} anchor and TP, $X_i = \sqrt{(x_E - x_{a_i})^2 + (y_E - y_{a_i})^2}$. The factors in the polynomial, $[x_E, y_E]^\top$ is the position of TPs in the earth-fixed reference frame, and $[x_{a_i}, y_{a_i}]^\top$ is the position of the i^{th} mooring anchor in the earth-fixed reference frame. P_1, P_2, and P_3 are generated from curve fitting, as presented in Fig. 6.5.

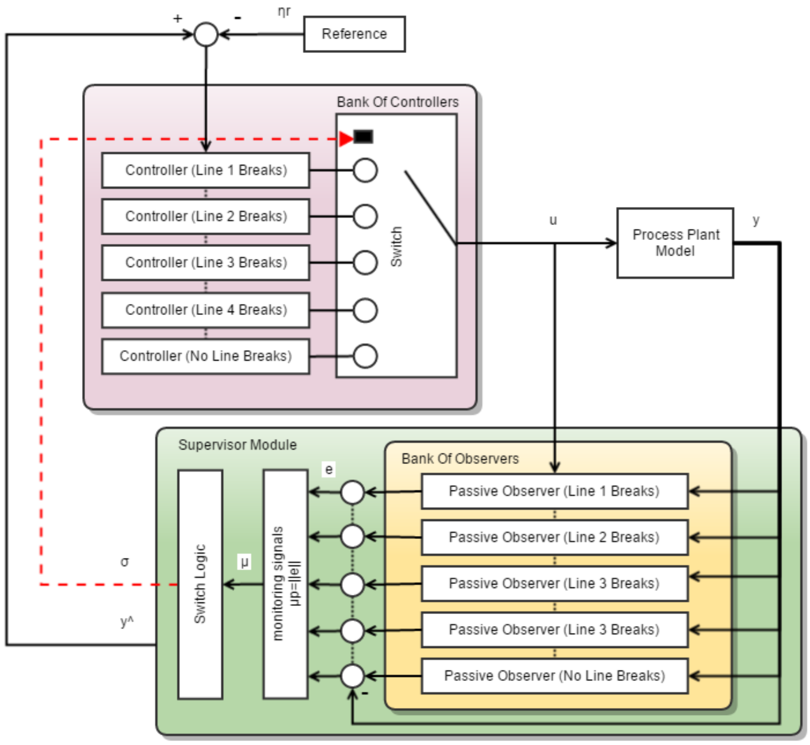

+
-
ηr
Reference
Bank Of Controllers
Controller (Line 1 Breaks)
Controller (Line 2 Breaks)
Controller (Line 3 Breaks)
Controller (Line 4 Breaks)
Controller (No Line Breaks)
Switch
u
Process Plant Model
y
Supervisor Module
Bank Of Observers
e
Passive Observer (Line 1 Breaks)
Passive Observer (Line 2 Breaks)
Passive Observer (Line 3 Breaks)
Passive Observer (Line 3 Breaks)
Passive Observer (No Line Breaks)
σ
μ
y^
Switch Logic
monitoring signals
μp=||ep||
-

Figure 6.4: Supervisory control for a TAPM system, adjusted from Nguyen et al. (2007).

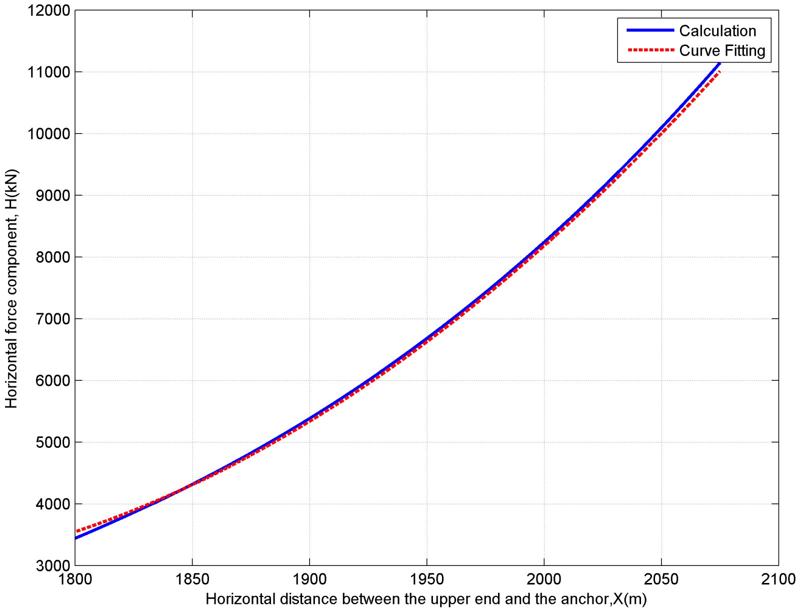


Figure 6.5: Result of curve fitting.

6.4.2 Hysteresis switching logic

The switching algorithm is borrowed from Hespanha (2001). The switching logic depends on the estimation errors, defined as

$$\boldsymbol{e}_j = \hat{\boldsymbol{y}}_j - \boldsymbol{y}, \tag{6.4}$$

where $\hat{\boldsymbol{y}}_j$ is the estimated output of the j^{th} observer. To avoid a chattering problem, hysteresis-based switching logics is used. The monitoring signals μ_j is then realized by the lowpass filter

$$\dot{\mu}_j = -\lambda \mu_j + |\boldsymbol{e}_j|_2^2, \tag{6.5}$$

where $\lambda \epsilon \mathbb{R}$ is a non-negative constant forgetting factor and $| \cdot |_2$ denotes the L_2 norm. The switching signal σ in the feedback loop is determined by the switching logic

$$(1+h)\mu_\sigma \leq \min_j \mu_j, \tag{6.6}$$

where $\sigma = \arg\min\limits_j \mu_j$, and $h \epsilon \mathbb{R}$ is a positive constant hysteresis factor. See Hespanha (2001) for details on Scale-Independent Hysteresis Switching. All possible configurations of the mooring system, including normal and failure modes, are listed in Table 6.1.

Table 6.1: Events for the supervisory index.

Modes σ	Description	Observer	Controller
p_0(normal)	No line breaks	Observer 0	Controller 0
p_1, p_2, p_3, p_4 p_5, p_6, p_7, p_8 (faulty)	Line 1/2/3/4 5/6/7/8 breaks	Observer 1/2/3/4 5/6/7/8	Controller 1/2/3/4 5/6/7/8

6.4.3 Comprehensive description

Regarding the NPO from Eq. 3.36 to 3.40, if one mooring line breaks, the respective element in the mooring restoring force vector $\boldsymbol{\tau}_{moor}$ will significantly decrease, or simply reduce to zero. Therefore, the right part of Eq. 3.39 cannot reach equilibrium with the same $\tilde{y}$. The equilibrium point is based on a new $\tilde{y}$. As $\tilde{\boldsymbol{y}} = \boldsymbol{y} - \hat{\boldsymbol{y}}$, and Eq. 3.40, a constant bias between the estimation and actual position appears. For different line failure, the residual errors are also distinguishing. Therefore, after measuring the residual signal error, we can find where is the line failure.

6.4.4 Controller design

The main emphasis of this paper is on multi-estimator-based identification of mooring line breakage. However, for the sake of completeness, a bank of multi-variable PID controllers, each designed for specific line breakage, is used to control the position of the moored vessel. The corresponding PID controller is designed to keep the LF heading and position at desire values given by an outer loop guidance system. There are several concept for this, where the various methods of "setpoint chasing" are promising (Nguyen and Sorensen, 2009). Each PID controller admits the realization

$$\dot{\xi} = \tilde{\boldsymbol{\eta}}, \tag{6.7}$$

$$\boldsymbol{\tau}_c = -\boldsymbol{K}_i \boldsymbol{R}(\psi)^\top \boldsymbol{\xi} - \boldsymbol{K}_p \boldsymbol{R}(\psi)^\top \tilde{\boldsymbol{\eta}} - \boldsymbol{K}_d \tilde{\boldsymbol{v}}, \tag{6.8}$$

where $\tilde{\boldsymbol{\eta}} = \hat{\boldsymbol{\eta}} - \boldsymbol{\eta}_d$, $\tilde{\boldsymbol{v}} = \hat{\boldsymbol{v}} - \boldsymbol{v}_d$, $\boldsymbol{K}_p$, $\boldsymbol{K}_i$, and $\boldsymbol{K}_d \epsilon \mathbb{R}^{3\times 3}$ are diagonal non-negative PID controller gain matrices.

6.5 GPS failure based on tension cells

6.5.1 Fault-tolerant control scheme

This algorithm is based on the tension positioning algorithm in Section 5.3. The structure of the FTC scheme is shown in Fig. 6.6. In the normal mode, the system is in closed-loop with the GPS signals and an open-loop tension-based localization algorithm. The estimated positions from the GPS and the tension-based posref is sent as input to a fault diagnosis module, which will detect the fault if the mean square error (MSE) overpasses a specific accommodation upper threshold T_u. The controller can reconfigure to the faulty mode based on the σ signal which is generated from the fault diagnosis module. It thereafter recovers based on the estimated states from the tension localization observer module. A lowpass filter is included in the fault diagnosis scheme to offset the rapid switch-on-and-off, which may be caused by measurement noise or instant faults, in case of wear and tear effects.

When the MSE between the GPS measurement and the tensions-based estimation reduces to another specific lower threshold value T_l, the controller recovers to the GPS signal. Normally, $T_l \leq T_u$ to avoid unnecessary switches. In this simulation, we set $T_u = 8(\mathrm{m})$ and $T_l = 4(\mathrm{m})$.

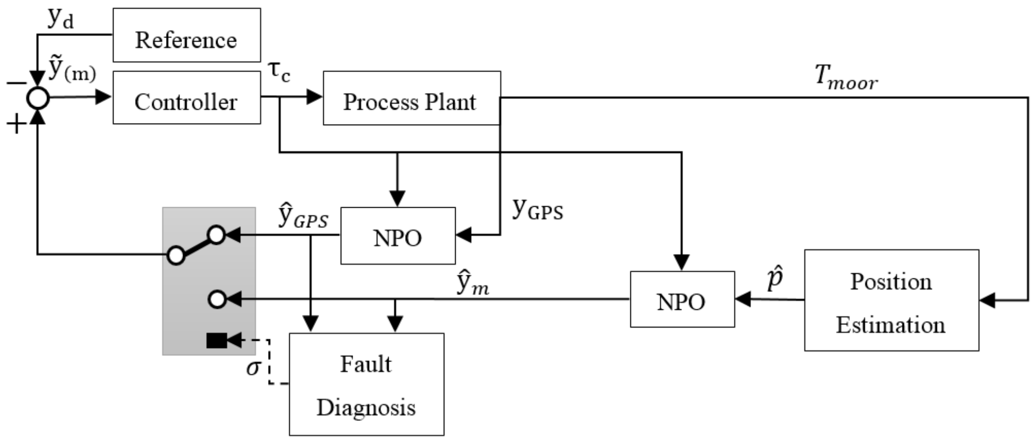


Figure 6.6: The structure of the fault-tolerant controller.

6.6 Sensor fault error prevention

To over come the tension sensor faults, a group of N different tension based observers is built. The tension-based localization is based on the signal of $N-1$ tension cells. We label the k^{th} is the estimation of all the mooring tensions but the k^{th} one.

When a fault happens, a supervisor needs to detect if the fault is a GPS failure or a tension sensor failure. If this is a GPS failure, then all the estimators alert. While the k^{th} estimator does not alert if this is the sensor fault of the k^{th} tension cell.

Chapter 7

Setpoint Chasing Algorithm Based on Deflection Equation

This chapter proposes a few theorems and objective functions to improve the setpoint chasing algorithm. A new model of the TTR end angles is deduced from classic mechanics.

7.1 Problem Statement

An offshore vessel is moored by M catenary mooring lines connected at the turret through fairleads, and L thrusters, with a group of N risers. The system arrangement is shown in Figure 7.1. A riser tensioners is equipped at the top end of every riser to provide relatively stable upward top tension. In this system, the actuators are the thrusters and the riser tensioners. The vessel is heading waves. A riser is modelled as a simple supported beam. The current velocity profile is $v_c(z)$ and the current direction is β_c in NED coordinate. The control inputs are the distance between the bottom and the top end of the riser X and the top tension of the riser T_{top}. At an equilibrium point, the mooring force is balanced by the environmental loads.

7.2 Reference frame

Three reference frames are applied in this paper:

The *Earth-fixed* coordinate $\{E\} = (x_E, y_E, z_E)$ has its origin at the center of the Earth, as well as fixed directions.

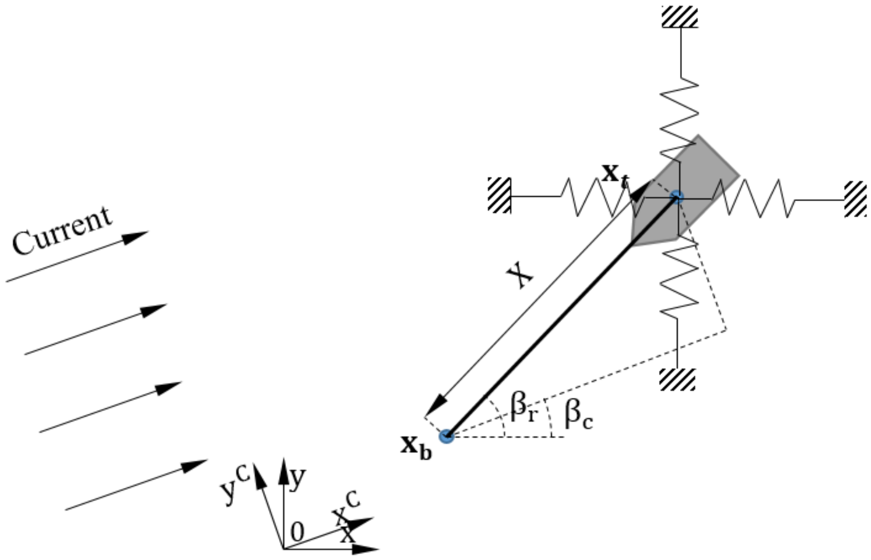


Figure 7.1: Reference frames.

The *north-east-down (NED)* coordinate $\{n\} = (x, y, z)$ with an origin at the equilibrium point.

The *current-fixed* coordinate $\{c\} = (x^C, y^C, z^C)$ is defined as the x direction is same as the current direction, and z direction point downward with a origin overlapped at $[00]^\top$.

The bottom and top ends of a riser $\boldsymbol{x}_b$ in NED are

$$\boldsymbol{x}_b = \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} \quad and \quad \boldsymbol{x}_t = \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix}. \tag{7.1}$$

The bottom and top end of a riser $\boldsymbol{x}_b$ in current frame are

$$\boldsymbol{x}_b^C = \begin{bmatrix} x_b^C \\ y_b^C \\ z_b^C \end{bmatrix} \quad and \quad \boldsymbol{x}_t^C = \begin{bmatrix} x_t^C \\ y_t^C \\ z_t^C \end{bmatrix}. \tag{7.2}$$

The relation in the NED coordinate and the current coordinate is given by

$$\boldsymbol{x}^C = \boldsymbol{R}_z(\beta_c)\boldsymbol{x}, \tag{7.3}$$

where the transfer matrix is given by

$$\boldsymbol{R}_z(\beta_c) = \begin{bmatrix} cos\beta_c & -sin\beta_c & 0 \\ sin\beta_c & cos\beta_c & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{7.4}$$

7.3 Thruster input

Assumption 7.1 (Slow-varying environmental loads assumption)**.** *The environmental loads, including the second-order wave loads, the wind loads and the current loads, are slow-varying; therefore, quasi-static analyses are applied to calculate the deflection of the riser.*

Assumption 7.2 (Linear restoring force assumption)**.** *Around the equilibrium point, the earth-fixed restoring force from the mooring lines are almost linear. when all the mooring lines are arranged symmetrically, the restoring force is given by*

$$\boldsymbol{\tau}^E_{moor} = \boldsymbol{G}_{mo}(\boldsymbol{\eta}) = \boldsymbol{g}_{mo}\boldsymbol{\eta}, \tag{7.5}$$

where $\boldsymbol{g}_{mo} = [g_{mo_{11}}, g_{mo_{22}}, g_{mo_{66}}]$.

Lemma 7.3 (Best TAPM motion direction.)**.** *In calm water, a turret-based TAPM stays stably at the equilibrium point* $A(\boldsymbol{x}_1)$. *In sea states, it is balanced at another equilibrium point within an accepted radius a,* $B(\boldsymbol{x}_2)$, *where the total force of the mooring force and the environmental loads is zero. Obviously* $\overline{AB} \leq a$. *Due to the change of work condition, there is a need to move the vessel to a new equilibrium point, such that the accepted radius is b, and* $b \leq a$. *Then the new running point* $C(\boldsymbol{x}_3)$ *should have an intersection angle* ψ_2 *to minimize the energy consuming in stationkeeping,* $\overline{AC} = b$. *The objective function is*

$$subject\ to \qquad \min_{\psi_2} = \boldsymbol{P}_t \boldsymbol{K}_t^{-1} \boldsymbol{T}_w^{\dagger}(\boldsymbol{\alpha}) \boldsymbol{R}(\psi_1) \begin{bmatrix} g_{mo_{11}}(a - b\cos\psi_2) \\ g_{mo_{22}}(-b\sin\psi_2) \end{bmatrix}, \tag{7.6}$$

where $\boldsymbol{P}_t = [P_{t1}, P_{t2}, \cdots, P_{tl}] \epsilon \mathbb{R}^l$ *is a control price coefficient vector to all thrusters, the control input vector for the thrusters* $\boldsymbol{\tau}_{thr}$, $\boldsymbol{K}_t$ *is a diagonal force coefficient matrix, and* $\boldsymbol{T}_w^{\dagger}(\boldsymbol{\alpha})$ *is the least square optimized control allocation matrix.*

Specifically, when $g_{mo_{11}} = g_{mo_{22}}$, *the vessel should stay in the line between the equilibrium in* *clam sea and the equilibrium point in the calculated sea state, to minimize the energy consumed for stationkeeping,*

Proof. At first, we assume the environmental loads is τ_{env}, τ_{env} is assumed to be a constant vector in a short term, due to the slow-varying environmental loads assumption. Hence, , that is

$$\tau_m + \tau_{env} = \boldsymbol{g}_{mo}(\boldsymbol{x}_2 - \boldsymbol{x}_1) + \tau_{env} = \mathbf{0}. \tag{7.7}$$

Based on Assumption 7.2, the force balance equation at the new equilibrium point $C(\boldsymbol{x}_3)$ is given by

$$\boldsymbol{g}_{mo}(\boldsymbol{x}_3 - \boldsymbol{x}_1) + \tau_{env} + \boldsymbol{\tau}_c = \mathbf{0}, \tag{7.8}$$

where $\tau_c = [\tau_{surge}, \tau_{sway}, \tau_{yaw}]^\top \epsilon \mathbb{R}^3$ is the vector of the actuator forces and moments. Keep the heading, such that $\boldsymbol{\eta}_C = \boldsymbol{\eta}_B + [\Delta x, \Delta y, 0]^\top$. Substitute Eq. 7.7 from Eq.7.8, we have

$$\boldsymbol{\tau}_c = -g_{mo}(\boldsymbol{x}_2 - \boldsymbol{x}_3) = \begin{bmatrix} g_{mo_{11}}(a\cos\phi_1 - b\cos(\phi_1 + \phi_2)) \\ g_{mo_{22}}(a\sin\phi_1 - b\sin(\phi_1 + \phi_2)) \\ g_{mo_{66}} \cdot 0 \end{bmatrix} = -\boldsymbol{R}(\psi_1) \begin{bmatrix} g_{mo_{11}}(a - b\cos\psi_2) \\ g_{mo_{22}}(-b\sin\psi_2) \end{bmatrix}. \tag{7.9}$$

When $g_{mo_{11}} = g_{mo_{22}}$,

$$\begin{aligned} |\boldsymbol{\tau}_c| &= |g_{mo}(\boldsymbol{x}_2 - \boldsymbol{x}_3)| \\ &= g_{mo}\sqrt{(a\cos(\phi_1) - b\cos(\phi_1 + \phi_2))^2 + (a\sin(\phi_1) - b\sin(\phi_1 + \phi_2))^2} \\ &= g_{mo}\sqrt{2 - 2ab\cos(\psi_2)}. \end{aligned} \tag{7.10}$$

The thruster input reaches the minimal when $\psi_2 = 0$. The control input vector for the thrusters $\boldsymbol{\tau}_{thr} = [u_1, u_2, \cdots, u_L]^\top$ is

$$\boldsymbol{\tau}_{thr} = \boldsymbol{K}_t^{-1}\boldsymbol{T}_w^\dagger(\boldsymbol{\alpha})\boldsymbol{\tau}_c, \tag{7.11}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_p]^\top \epsilon \mathbb{R}^p$ is a vector of azimuth angles, $\boldsymbol{K}_t = diag(K_{t1}, K_{t2}, \cdots, K_{tl}) \epsilon \mathbb{R}^{L\times L}$ is a diagonal force coefficient matrix, and $\boldsymbol{T}_w^\dagger(\boldsymbol{\alpha})$ is the least square optimized control allocation matrix, and it is given by

$$\boldsymbol{T}_w^\dagger(\boldsymbol{\alpha}) = \boldsymbol{W}^{-1}\boldsymbol{T}^\top(\boldsymbol{\alpha})[\boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{W}^{-1}\boldsymbol{T}^\top(\boldsymbol{\alpha})]^{-1}, \tag{7.12}$$

where $\boldsymbol{W}$ is a diagonal matrix weighting the control forces and $\boldsymbol{T}(\boldsymbol{\alpha}) \epsilon \mathbb{R}^{3\times L}$ is the thrust con-

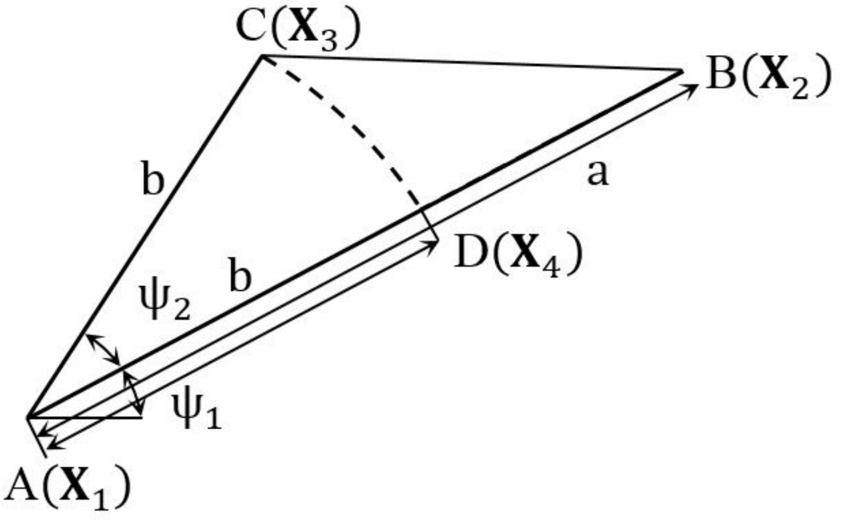


Figure 7.2: Diagram for proof of the best direction.

figuration matrix which describes the geometry or location of the thrusters. ■

7.4 Riser dead angle

Assumption 7.4 (Light-weight uniform riser assumption)**.** *The mass of a riser is disregarded, as well as the deformation caused by the gravity. The geometric properties and physical properties are constant along the length the riser, including the diameter, density, material, mass, stiffness, etc. Additionally, the riser does not provide restoring force to the moored vessel.*

The light-weight assumption is to continue the calculation based on beam theory. Since the riser mass influence the axial tension, we cannot disregarded it in the beam theory. Additionally, the gravity has a different affect to dissimilar nodes along the riser. Then the top tension is given by

$$T_t op = T_{tensioner} + \frac{w_r l}{2}, \tag{7.13}$$

where $w_r = (\rho_s A - \rho_w \frac{\pi D_o^2}{4})$, ρ_s is the density of the riser material, ρ_w is the density of water, $A = \frac{\pi}{4}(D_o^2 - D_i^2)$ is the cross section area, D_o is the outer diameter of the riser, D_i is the inner diameter of the riser, and l is the length of the riser.

Superposition principle is applied here to calculate the dead angles. We assume the displacement of the whole riser is a superposition of two parts: (I) a beam with constant load and axial tension, (II) a straight line connect the well on the seafloor and the surface vessel. This is shown in Figure 7.4.

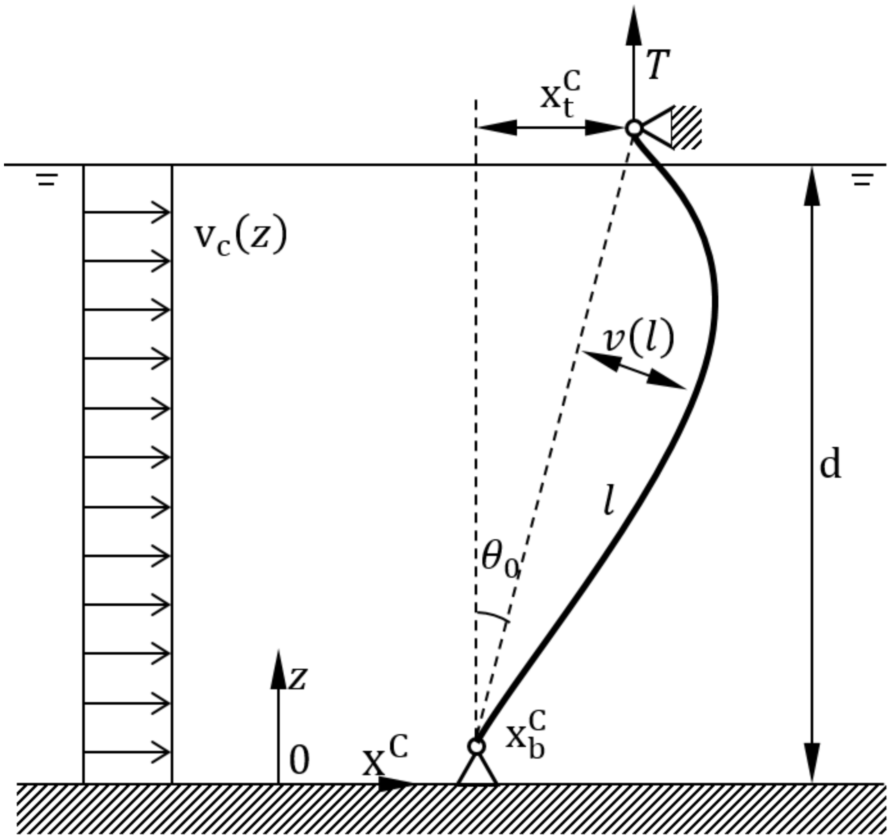


Figure 7.3: Reference frames.

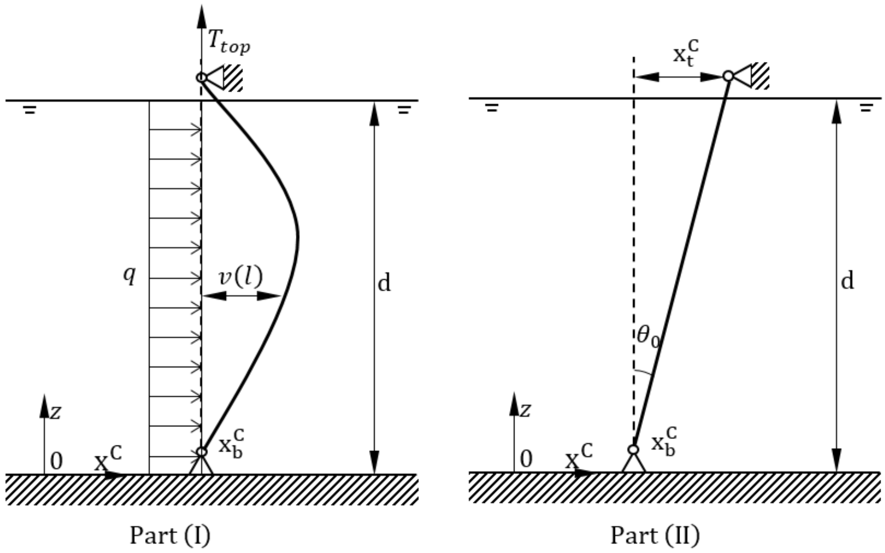


Figure 7.4: Reference frames.

However, we cannot ensure the current has a same direction with the vessel displacement. Hence, 2D is not enough to solve this problem. Additionally, FEM model stands out for the simulator, while it is not easy to control or obtain analytical solutions.

Lemma 7.5 (Changed top and bottom end angles of a riser due to current and top tension)**.** *Regarding a riser subjected in a constant current, the 2D changed angle at the top end* $z = 0$ *and the bottom end* $z = -l$ *are*

$$\Delta\theta_t^{2D} = \frac{dv(z)}{dz}|_{z=0} = \frac{ql^3}{EI}\psi_1(u) \quad and \quad \Delta\theta_b^{2D} = \frac{dv(z)}{dz}|_{z=-l} = -\frac{ql^3}{EI}\psi_1(u), \tag{7.14}$$

where $\Delta\theta_t$ *and* $\Delta\theta_b$ *are the changed top end and bottom end angles, the main force per unit length due to the current* $q(z) = \frac{1}{2}\rho_w C_D D_o v_c^2(z)$, ρ_w *is water density,* C_D *denotes the empirical drag coefficient,* $v_c(z)$ *is the current velocity along the vertical direction,* E *is modulus of elasticity,* $I = \frac{\pi}{2}\left[(\frac{D_o}{2})^4 - (\frac{D_i}{2})^4\right]$ *is the moment of inertia of the riser,* $\psi_1(u) = \frac{1}{8u^3}(u - thu)$ *is a monotonic decreasing function,* $u = \frac{kl}{2}$, $k = \sqrt{\frac{T_{top}}{EI}}$, *and* T_{top} *is the top tension,* $sh\cdot = \frac{e^x - e^{-x}}{2}$, $ch\cdot = \frac{e^x + e^{-x}}{2}$, *and* $th\cdot = \frac{sh\cdot}{ch\cdot}$.

Proof. Based on Morison equation, the force acting on unit length riser is given by

$$F(z) = \rho_w C_M \frac{\pi D^2}{4}\dot{v}_c(z) + \frac{1}{2}C_D\rho_w D v_c(z)|v_c(z)|, \tag{7.15}$$

where the formal component is called Froude–Kriloff force, which is introduced by unsteady pressure field. Therefore, it is reasonable to assume that the added mass coefficient C_M does not influence the performance of the riser with constant current speed.

The governing equation for a beam with axial tension is given by.

$$EI\frac{\partial^4 v(z,t)}{\partial z^4} - T_{top}\frac{\partial^2 v(z,t)}{\partial z^2} = q(z) \tag{7.16}$$

A general solution to this differential equation is

$$v(x) = v_0 + \frac{\theta_0}{k}shkz + \frac{M_0}{EIk^2}(chkx - 1) + \frac{N_0}{EIk^3}(shkx - kx) + \int_c^z \frac{q(\xi)\,d\xi}{EIk^3}[shk(z-\xi) - k(z-\xi)]. \tag{7.17}$$

The boundary conditions are that the displacements and the moment at both the bottom and the top end is zero. They are

$$\partial v(z=0)=0 \quad and \quad \frac{\partial^2 v(z=0)}{\partial x^2}=0, \tag{7.18}$$

$$\partial v(z=-l)=0 \quad and \quad \frac{\partial^2 v(z=-l)}{\partial x^2}=0. \tag{7.19}$$

After substituting the boundary conditions, Eq. 7.17 becomes

$$v(z)=\frac{ql^4}{EI(2u)^4}\left[\frac{chk(z-l/2)}{chu}-1+\frac{u^2-k^2(z-2/l)^2}{2}\right]. \tag{7.20}$$

The position in current frame is

$$\boldsymbol{x}_I^C(z)=\begin{bmatrix} v(z) \\ 0 \end{bmatrix}, \tag{7.21}$$

Through differential Eq. 7.21, the changed angle is

$$\Delta\theta^{2D}(z)=\frac{dv(z)}{dz}=\frac{ql^4}{EI(2u)^4}\left[\frac{shk(z-\frac{l}{2})\cdot k}{chu}-k^2(z-\frac{l}{2})\right]. \tag{7.22}$$

■

Remark: when T_{top} is constant, u has a constant value. $\Delta\theta_1$ and $\Delta\theta_2$ are in proportion the current load q.

Theorem 7.6 (Changed top and bottom end angles of a riser due to current, top tension, and vessel displacement)**.** *In a current with constant current speed and direction $v_c(z)$, a top tensioned riser connects with the top tensioner at the vessel. The 3D changed top end angle α_t and the bottom end angle α_b are given by*

$$\alpha_b \simeq tan(\alpha_b)=\sqrt{(aX+b\psi_1(u))^2+cX^2}, \tag{7.23}$$

$$\alpha_t \simeq tan(\alpha_t)=\sqrt{(aX-b\psi_1(u))^2+cX^2}, \tag{7.24}$$

where X is the displacement of the upper vessel, $a=\frac{sin(\beta_r-\beta_c)}{l}$, $b=\frac{ql^3}{8EI}$, $c=(\frac{cos(\beta_r-\beta_c)}{l})^2$, β_r is the riser angle in the horizontal plane with the x axis, β_r is the current angle.

Proof. The total position of the riser is a superposition of the position in Part I and Part II, that is

$$\boldsymbol{x}^C(z) = \boldsymbol{x}_I^C(z) + \boldsymbol{x}_{II}^C(z). \tag{7.25}$$

Next, we need to add the displacement caused by the vessel motion to the deform caused by the current loads as calculated before. The position in current frame is

$$\boldsymbol{x}_{II}^C(z) = \boldsymbol{x}_b^C + \begin{bmatrix} sin(\beta_r - \beta_c) \\ cos(\beta_r - \beta_c) \end{bmatrix} \frac{X}{l} z. \tag{7.26}$$

The 3D tangent vector at the bottom, that is $z = 0$, is

$$\begin{bmatrix} \frac{\partial \boldsymbol{x}^C}{\partial z} \\ \frac{\partial z}{\partial z} \end{bmatrix} = \begin{bmatrix} \frac{sin(\beta_r - \beta_c)}{l} X + \theta_b \\ \frac{cos(\beta_r - \beta_c)}{l} X \\ 1 \end{bmatrix}. \tag{7.27}$$

Therefore the bottom riser angle is

$$\begin{aligned} tan(\alpha_b) &= \sqrt{\left[\frac{sin(\beta_r - \beta_c)}{l} X + \frac{ql^3}{8EI}\psi_1(u)\right]^2 + \left[\frac{cos(\beta_r - \beta_c)}{l} X\right]^2} \\ &= \sqrt{(\frac{X}{l})^2 + (\frac{ql^3}{8EI})^2 \psi_1^2(u) + \frac{ql^2 sin(\beta_r - \beta_c)}{4EI}\psi_1(u)} \end{aligned}. \tag{7.28}$$

To calculate the top end angle is a same process. ■

7.5 Vortex-induced vibration

Another important issue is to control the vortex-induced vibration.

Three non-dimensional parameter are Reynold number, Keulegan-Carpenten number, and Strouhal number.

When the current passing a cylinder, Reynold number is defined as

$$Re = \frac{UD_o}{v}. \tag{7.29}$$

where U is the velocity, v is the kinematic viscosity.

Strouhal number is given by

$$St = \frac{f_{st} D}{U}. \tag{7.30}$$

where f_{st} is the vortex sheding process; U is the free stream velocity. For the fluid with $200 < Re < 2 \times 10^5$, St number can be treated as a constant, 0.2.

Keulegan-Carpenten number is given by

$$KC = \frac{U_m}{fD}. \tag{7.31}$$

where U_m is amplitude of flow velocity, f is the oscillation frequency.

The eigenmodes are assumed in a form

$$\phi(x) = \phi_{0n} sin(\frac{n\pi}{l}x)\ n = 1,2,3,\cdots, \tag{7.32}$$

where l is the riser length and ϕ_{0n} is the amplitude of the n^{th} mode. Therefore, a simplified vortex-induced vibration frequency is given by

$$f_v = \frac{0.2U}{D_o}. \tag{7.33}$$

For a tensioned riser, it can be simplified as a combination of a tensioned string with moment-free end support at both ends and a bending dominated beam. The n^{th} mode eignefrequency of a tensioned string is given by

$$\omega_{s,n} = \frac{n\pi}{l}\sqrt{\frac{T}{m_r}}\ n = 1,2,3,\cdots, \tag{7.34}$$

where T is the tension interacting on the sting, $m_r = \frac{M_r}{l}$ is the cylinder mass per unit length. The n^{th} mode eigenfrequency of the beam is given by

$$\omega_{b,n} = \frac{n^2\pi^2}{l^2}\sqrt{\frac{EI}{m_r}}\ n = 1,2,3,\cdots, \tag{7.35}$$

where EI is the beam stiffness. The overall eigenfrequency for a riser is given by

$$\omega_{b,n} = \frac{n\pi}{l}\sqrt{\frac{T}{m_r} + \frac{n^2\pi^2}{l^2} \cdot \frac{EI}{m_r}}\ n = 1,2,3,\cdots. \tag{7.36}$$

7.6 multiobjective optimization

For a specific current profile, the shedding frequency is assumed to be a constant in a short term. Therefore, the top tension should be kept as a value that lock-in cannot happen. As lower orders of natural frequency have more remarkable influences to the whole system, we only consider the first k^{th} order deformations. Therefore, the allowed top tension is divided into several subsets. We can find that the top tension set is divided by the values which may cause resonance, hence the tension set is no longer convex. However, we can construct a few of convex optimal questions. After getting the optima in all subsets, the best scheme can be solved.

Therefore, the allowance of the top tension between the

$$SF_l T_{top,k+1} \leq T_{top} \leq SF_u T_{top,k}, \quad k = 1, 2, \cdots, k, \tag{7.37}$$

where $SF_l > 1$ and $SF_u < 1$ is a pair of lower and the upper safety factors to ensure the riser's natural frequency at the specific top tension stays away from the vortex-induced vibration frequency.

Multiobjective optimization can be applied here to give the best setpoint position and the mean pre-tension of the riser tensioner. The object function is the energy used the during operation, as well as the risk of mooring line and riser breakage. The current profile is uniform distributed.

The control price objective function is given by

$$f_c = \boldsymbol{P}_t \boldsymbol{K}_t^{-1} \boldsymbol{T}_w^{\dagger}(\boldsymbol{\alpha}) \boldsymbol{g}_{mo} \Delta\boldsymbol{\eta} + \boldsymbol{P}_r \boldsymbol{K}_r \boldsymbol{W}_r^{\top}, \tag{7.38}$$

where $\Delta\boldsymbol{\eta}$ is the changed position from the equilibrium point to the changed potion.

The riser risk objective function (Sørensen, 2012) is given by

$$f_r = w_b \alpha_b^2 + w_t \alpha_t^2, \tag{7.39}$$

where w_b and w_t are thew weighted factor for the bottom end angle and the top end angle.

The fatigue of the mooring lines objective function (Fang et al., 2013) is given by

$$f_m = \sum_{i=1}^{M} \alpha_i (\theta_{ci} - \theta_i)^2, \tag{7.40}$$

where θ_{ci} is the critical structural reliability index which considers the extreme value of the tension in all sea states. The structural reliability index is given by $\theta_i = \frac{T_{ci}-T_{exi}}{\sigma_{ci}}$, where T_{ci} the mean critical strength of the ith mooring line according to the manufacturer's specification, T_{exi} is the extreme value of the mooring-line tension, and σ_{ci} is the standard deviation of the critical strength, normally, $\sigma_{ci} 0.035 T_{ci}$.

$$\begin{aligned}
minimize \quad & \min \quad f_c, \\
& \min \quad f_r \\
& \min \quad f_m \\
subjectto \quad & T_i - SF_u T_{top,k} \leq 0, \\
& SF_l T_{top,k+1} - T_{top} \leq 0, \\
& \|\boldsymbol{\eta}_r\|_2 \leq r
\end{aligned} \tag{7.41}$$

where $\boldsymbol{P}_t = [P_{t1}, P_{t2}, \cdots, P_{tl}] \epsilon \mathbb{R}^l$ is a control price coefficient vector to all thrusters, $\boldsymbol{P}_r = [P_{r1}, P_{r2}, \cdots, P_{rn}]^\top \epsilon \mathbb{R}^n$ is a control price coefficient vector to all riser tensioners, and r is the radius of the accepted motion region.

Remark: For a drillship without mooring lines, the objective function can be further simplified with disregarding the f_m.

7.7 Reference system

A reference model provide setpoint signal to the controller. It is a low pass signal of the LF position vector $\boldsymbol{\eta}$, and the setpoint is estimated by

$$\dot{\boldsymbol{\eta}}_r = -\Lambda \boldsymbol{\eta}_r + \Lambda \boldsymbol{\eta}, \tag{7.42}$$

where $\boldsymbol{\eta}_r$ is the reference signal output from the reference system, Γ is the first order diagonal and non-negative filter gain matrix with the cut-off frequencies $\frac{1}{T_{s_i}}$(Hz), $\Lambda = diag(\frac{1}{T_{s_1}}, \frac{1}{T_{s_2}}, \frac{1}{T_{s_3}})$.

Chapter 8

Simulation Overview

8.1 Simulation environment and model parameters

The simulations are conducted in the MATLAB® and Simulink® environment. The MSS toolbox from NTNU is a Matlab/Simulink library and simulator for marine systems (MSS, 2010). A Simulink lib which contains the most recently applied module for the TAPM is built. For more detail, see Appendix A. The key parameters are tabulated in Table 8.1, 8.2, and 8.3.

Table 8.1: Vessel main particulars.

Principle Dimension	Values
Vessel Type	FPSO
Length between perp. L_{pp}	200
Breadth B(m)	44
Draught T(m)	12
Mass M(kg)	1.004e+08
Center of gravity C_G	[0,0,11]
Trans. metacentric height $\overline{GM_T}(m)$	5500
Long. metacentric height $\overline{GM_L}(m)$	7.95
Density of cable $\rho_k(kg/m^3)$	251.4
Thruster arrangement	
$[x_{thr_1}, y_{thr_1}](m)$	$[75, 0]$
$[x_{thr_2}, y_{thr_2}](m)$	$[-100, 5]$
$[x_{thr_3}, y_{thr_3}](m)$	$[-100, -5]$

Table 8.2: Mooring line dimensions.

Principle Dimension	Values
Dens. of ambient water $\rho_w (kg/m^3)$	1025
Length of the cable $L_m(m)$	2250
Elastic modulus $E_m Pa(N/m^2)$	4.5757×10^{10}
Cable cross section area $A_m(m^2)$	0.005
Cable diameter $d_m(m)$	0.08
Max strain ε	0.005
Position of the anchors	
$[x_{a_1}, y_{a_1}, z_{a_1}](m)$	$[1950, 0, -1000]$
$[x_{a_2}, y_{a_2}, z_{a_2}](m)$	$[0, 1950, -1000]$
$[x_{a_3}, y_{a_3}, z_{a_3}](m)$	$[-1950, 0, -1000]$
$[x_{a_4}, y_{a_4}, z_{a_4}](m)$	$[0, -1950, -1000]$
$[x_{a_5}, y_{a_5}, z_{a_5}](m)$	$[1378.9, 1378.9, -1000]$
$[x_{a_6}, y_{a_6}, z_{a_6}](m)$	$[-1378.9, 1378.9, -1000]$
$[x_{a_7}, y_{a_7}, z_{a_7}](m)$	$[-1378.9, -1378.9, -1000]$
$[x_{a_8}, y_{a_8}, z_{a_8}](m)$	$[1378.9, -1378.9, -1000]$

8.2 Simulation I. Mooring line model comparison

Since the catenary equation initialize the FEM model, the mooring force without current is the same. In real sea states, drags due to current can effect the tension force and the top end angle. Simulation I is conducted to compare the catenary equations and high-fidelity FEM model. The key parameters of the mooring lines are tabulated in Table 8.2.

8.3 Simulation II. Riser model verification

As the fact that the setpoint chasing algorithm is impossible to verify based on merely numerical simulations. Only the riser model in Theorem 7.5 and 7.6 is verified here. The verification is conducted in SIMA®. SIMA is a simulation and analysis tool for marine operation and floating system (Afriana, 2011). The solver is RIFLEX®, a commercial FEM program for static and dynamic analysis of risers and mooring lines (Fylling et al., 1995).

A uniform distributed riser is built, see Figure 8.1. The only environmental load is the uniform current. The vessel is not coupled with the riser. Static simulations are conducted. Top tension, or so-called initial stress in the software, is added by initially elongate the riser. The boundary conditions are all fixed in 6 DOFs. The detail parameter of the riser and environments is tabulated in Table 8.3. Results are collected after 500 steps, with volume force, current force, initial stress force and tension force. The mesh density is 0.5m to enhance the accuracy of the simulation. As out model is quite simple, high mesh density will not influence the computational speed. The end angle is calculate as

$$\alpha_b = atan(\frac{w(z=-195m) - w(z=-200m)}{5}), \tag{8.1}$$

$$\alpha_t = atan(\frac{w(z=-5m) - w(z=0m)}{5}), \tag{8.2}$$

where $w(z) = \sqrt{x^2(z) + y^2(z)}$.

Table 8.3: Principle dimensions of mooring mines.

Parameter	Description	Values
C_D	Drag force coefficient	1
C_M	Added mass coefficient	2
D_1	Riser inner diameter	0.15 m
D_2	Riser outer diameter	0.30 m
E	Modulus of elasticity	206 Gpa
h	Riser length	200 m
L	Length of the riser	200 m
ρ_s	Steel density	7850 kg/m^3
ρ_w	Water density	1025 kg/m^3
ρ_o	Oil density	800 kg/m^3
T_{min}	Upper tension limit	2700 kN
T_{max}	Lower tension limit	350 kN

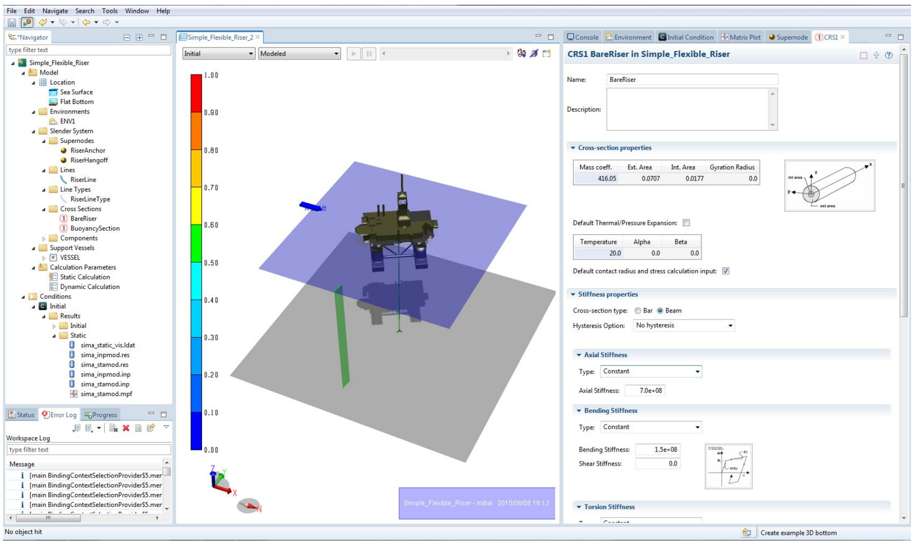
Simulation III.1: The vessel is fixed at (0,0,0), and the current direction is 0 deg. The current speed is 0.3,0.6,0.9,1.2,1.5 m/s.

Simulation III.2:The vessel is fixed at (0,0,0), the current speed is 1 m/s, and the current direction is 0 deg. The top tension is changed due to the initial stress.

Simulation III.3: The vessel is fixed at (0,0,3), and the current speed is 1 m/s. The current direction is 0,30,60,90 deg.

8.4 Simulation III. Mooring line breakage based on supervisory control

A line break occurs when the tension overpasses its breaking strength, for duration Δt. At this moment, the tension of the broken line becomes zero. Table 8.1 presents the main pa-



[main BindingContextSelectionProvider$5.mer
[main BindingContextSelectionProvider$5.mer
[main BindingContextSelectionProvider$5.mer
[main BindingContextSelectionProvider$5.mer

CRS1 BareRiser in Simple_Flexible_Riser

Name: BareRiser

Description:

Cross-section properties

Mass coeff.	Ext. Area	Int. Area	Gyration Radius
416.05	0.0707	0.0177	0.0

Default Thermal/Pressure Expansion: □

Temperature	Alpha	Beta
20.0	0.0	0.0

Default contact radius and stress calculation input: ☑

Stiffness properties

Cross-section type: ○ Bar ● Beam

Hysteresis Option: No hysteresis

Axial Stiffness

Type: Constant

Axial Stiffness: 7.0e+08

Bending Stiffness

Type: Constant

Bending Stiffness: 1.5e+08

Shear Stiffness: 0.0

Torsion Stiffness

Figure 8.1: SIMA model.

rameters of the simulation model. The International Towing Tank Conference (ITTC) spectrum (Fossen, 2011) was used to simulate the irregular waves with significant wave height $H_s = 5.5\,m$ and mean wave direction $\beta = 45\,deg$. The current is assumed to have a constant speed in earth frame $v_c = 0.1\,m/s$ and direction $\beta_c = 0\,deg$.

The mooring force inputs to the observers are estimated using a second-order polynomial with curve fitting while keeping the heave equal to zero. With the collection of the positions and tension data from quasi-static analysis of the mooring lines, we can generate a quadric relation with the *cftool* toolbox in MATLAB®. To ensure computational efficiency and save simulation time, a 2D lookup table was used to calculate the mooring forces from the position of each mooring line terminal point. This 2D lookup table was generated offline where the inputs are the horizontal distances and the heave motion.

A line break occurs when the tension overpasses its breaking strength. for duration Δt. At this moment, the tension of the broken line becomes zero. Table 8.1 presents the main parameters of the simulation model. The ITTC spectrum (Fossen, 2011) was used to simulate the irregular waves with significant wave height $H_s = 5.5\,m$ and mean wave direction $\beta = 45\,deg$. The current is assumed to have a constant speed in earth frame $v_c = 0.1\,m/s$ and direction $\beta_c = 0\,deg$.

In this simulation study we have considered two different mooring configurations.

Simulation III.1: The FPSO is moored by four mooring lines, labeled 1-4 in Fig.3.1. In this simulation, we assume line break happens at $t_f = 250$ after the FPSO reach its equilibrium point.

Simulation III.2: The FPSO is moored by all eight mooring lines in Fig. 3.1. In this simulation, line breakage happens at $t_f = 250$ seconds after the FPSO reach its equilibrium point.

In this simulation study we have considered two different mooring configurations.

8.5 Simulation IV. GPS failure detection based on tension measurement

The faulty measurement can be expressed as

$$\boldsymbol{y} = \boldsymbol{\eta} + \boldsymbol{C_f} \boldsymbol{y}_f(\tau), \tag{8.3}$$

where $\boldsymbol{C} = \begin{bmatrix} \boldsymbol{I}_{2\times2} & \boldsymbol{0}_{2\times1} \end{bmatrix}$, and $\boldsymbol{y}_f(\tau) \epsilon \mathbb{R}^2$ represents the measurement faults in x and y direction, respectively. The fault is a bias that occur in the period from t_{fs} to t_{ft},

$$\boldsymbol{y}_f(\tau) = \begin{cases} \boldsymbol{f} & \tau \epsilon [t_{fs}, t_{ft}] \\ \boldsymbol{0} & else. \end{cases} \tag{8.4}$$

In this simulation, all the GPS signals are influenced due to the radio-wave absorption in upper medium frequency and lower high frequency ranges. Assume there are no additional faults occurring at the same time with the sensor fault. The design requirement is that the TAPM system can reconfigure the control inputs after the fault happens and recover after the fault disappears.

The TAPM system is exposed to ITTC spectrum waves and constant-velocity currents. The detailed environmental parameters are tabulated in Table 8.4. The GPS fault $\boldsymbol{f} = [10, 5, 0]^\top$ (m) happens at $t_f = 2000$ seconds and ends at $t_f = 4000$ seconds.

Table 8.4: Environments distributions.

Environments Parameters	Symbol and Unit	Values
Current velocity	v_c(m)	0.1
Current direction	β_c(deg)	0
Significant wave height	H_s(m)	5.5
Mean wave direction	β(deg)	45

8.6 Simulation V. Simplified tension-based localization

8.6.1 Simulation V.1 Availability of tension-based localization

The detailed parameters are shown in Table 8.1, as well as the thruster arrangement. MSE is employed as the criteria here to evaluate the performance of the position performance, given by

$$MSE = |\boldsymbol{y} - \hat{\boldsymbol{y}}|_2^2. \tag{8.5}$$

The smaller the criteria is, the better performance the estimator will have.

Simulation V.1a: The first simulation aims to test if the tension-based localization method is applicable. Assume that there are no noise in the tension measurements. Then, the NPO is not necessary in simulation I.1. For the sake of simplicity, it is assumed that the FPSO can move in the horizontal plane without any control limits. An additional assumption is that the turret does not have heave motion.

Simulation V.1b: There exists noise in the tension measurements during practical operations. Therefore, noisy signal is a crucial issue to be considered. Independent Gaussian white noise is added to the measurements. The moored FPSO moves in 6DOF, which means the heave motion influences the tensions. Hence, the NPO is used in this simulation.

8.6.2 Simulation V.2 Influence analysis of the tension measurement noise

Simulation V.1c: The MSE method is used. In this simulation, the tension noise varies in a large range. The aim is to test the influence of the tension measurement noise to the accuracy of the localization,

8.6.3 Simulation V.3 Second order cone programming localization performance

In this simulation, the SOCP algorithm is implemented and tested. *CVX* is applied to solve this problem (Grant and Boyd, 2008, 2014). It is a MATLAB toolbox for specifying and solving convex programs. This simulation is conducted in an ideal condition. Specifically, there is no wave-induced heave motion and GPS noise. The unknown nodes are the anchors, and the positions of the vessels are received from GPS. This is a scaling simulation which will make the figure clearer. All eight anchors are assumed to locate around eight points which are distributed evenly in a circle with 50 meters radius.

Simulation V.3a: The FPSO is allowed to run in a large circle which is much larger than practical situation. This simulation aims to verify the codes during the simulation.

Simulation V.3b: The FPSO is only allowed to move in a relatively small region around the equilibrium point. This is the practical situation. It is conducted to test whether SOCP exactly suitable for a TAPM.

8.7 Simulation VI. Simultaneous localization performance with unknown anchors

All the environmental parameters are generated randomly, including the actual anchor positions, the significant wave height, the wave direction, and the current speed. The anchors are placed randomly in circles of radius 150 meters centered at the best initial estimated positions respectively, see Table 8.5. Turret positions are determined by different environmental parameters. The current direction is randomly in a $\pm 30deg$ region from the wave direction. There is no control inputs in the surge and sway. The controller only controls heading. Then we have a group of virtual vessels running in different environments at various equilibrium points.

Data are only collected after the vessels have stable performance at the new equilibrium points. Though the vessels can never become motionless due the time-varying loads, it is not a problem if the vessels have small motions. This is because the $\bar{\boldsymbol{x}}^t$ update process is provided by the NPO.

In this simulation, we assume there exist 10 virtual vessels. The arrangement is shown in Figure 8.2. The initial estimations are distributed on a circle with a 1950 radius.

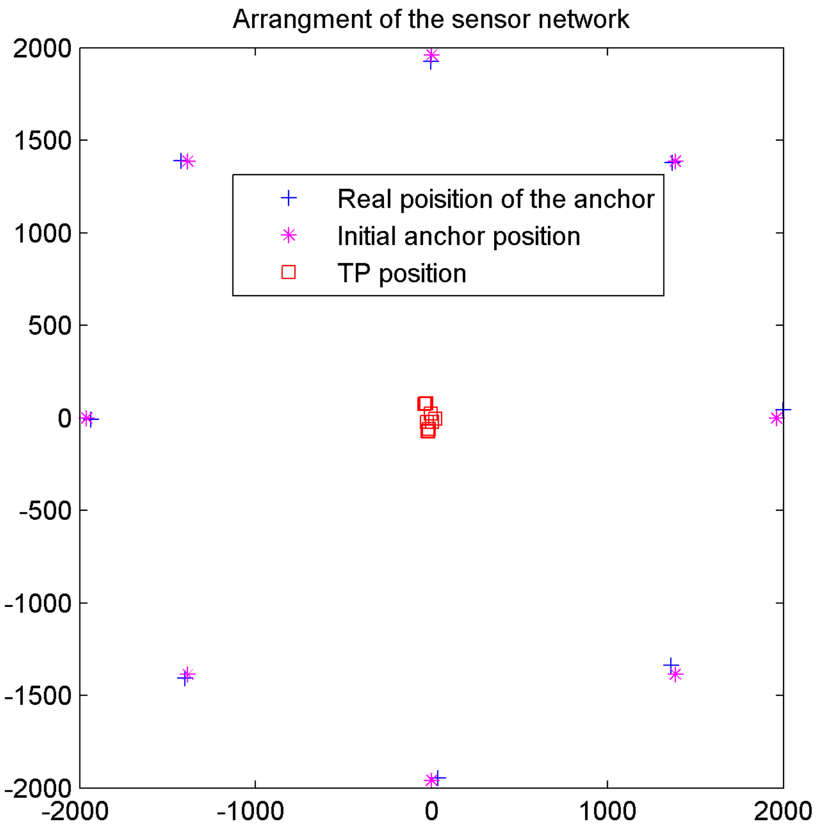


Figure 8.2: Anchor position estimation with LF and WF

Table 8.5: Anchor positions and initial estimations.

Index	Position of the anchors	Initial estimation
$[i]$	$[x_{a_i}, y_{a_i}, z_{a_i}]^\top (m)$	$[\hat{x}_{a_i}(0), \hat{y}_{a_i}(0), \hat{z}_{a_i}(0)]^\top (m)$
1	$[0, 1960, -1000]^\top$	$[1999.5, 44.2, -1000]^\top$
2	$[-1960, 0, -1000]^\top$	$[1369.4, 1379.7, -1000]^\top$
3	$[-1960, 0, -1000]^\top$	$[-2.9, 1924.9, -1000]^\top$
4	$[0, -1960, -1000]^\top$	$[-1422.3, 1389.2, -1000]^\top$
5	$[1385.9, 1385.9, -1000]^\top$	$[-1937.4, -10.1, -1000]^\top$
6	$[-1385.9, 1385.9, -1000]^\top$	$[-1400.1, -1407.4, -1000]^\top$
7	$[-1385.9, -1385.9, -1000]^\top$	$[36.9, -1947.4, -1000]^\top$
8	$[1385.9, -1385.9, -1000]^\top$	$[1360.1, -1338.1, -1000]^\top$

Table 8.6: Vessel equilibrium points.

Index	Position of equilibrium points
$[j]$	$[x_{a_j}, y_{a_j}]^\top (m)$
1	$[-28.1, 76.5]^\top$
2	$[5.1, -22.7]^\top$
3	$[-23.6, -70.5]^\top$
4	$[-25.9, -23.6]^\top$
5	$[-34.9, 77.3]^\top$
6	$[-39.6, 74.7]^\top$
7	$[22.6, -6.4]^\top$
8	$[-19.3, -74.9]^\top$
9	$[-13.9, -61.4]]^\top$
10	$[-2.2, 22.1]^\top$

Chapter 9

Results: Model Comparison

9.1 Results of Simulation I. Mooring line model comparison

9.1.1 Mooring force

The mooring forces from a quasi-static analysis is shown in Figure 9.1. The results are calculated based on the catenary equations. Its core algorithm is the binary search (Aamo, 1999). The three curves in Figure 9.1 means the mapping between the horizontal distance and the mooring force of the vessel moving on water surface without heave variance. The curves are almost parallel. We notice that a specific heave only causes a small bias to the $z = 0m$ curve. Therefore, heave motion has little influence to the range estimation.

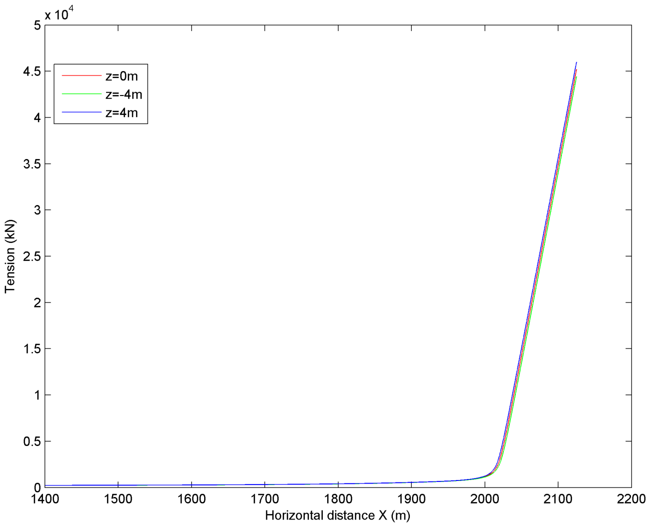


Figure 9.1: The relation between horizontal distance and tension from quasi-static analysis

Therefore, we can get a range measurement through a tension measurement.

9.1.2 Comparison between the catenary equations and the FEM model

A group of simulations is conducted to test the influence of the current to a mooring line. The current speed varies from $0\,m/s$ to $1\,m/s$ with a $0.1\,m/s$ interval. There is no vertical current in this set of simulations, since the vertical current is quite slow in practice. The current direction will also influence the results. The values collected at 2000 seconds after the starting of all simulations. It aims to collect the stable value after the deformation happening and stay at the new equilibrium points. All the data are conducted to one specific mooring line. The top end and the bottom end are fixed for all the simulations. The result from FEM model is show in the following figures. Since a pair of current direction, β_c and $2\pi - \beta_c$, are expected to have symmetric effects to a mooring line. The figures only contain the results of current direction in $[0, \pi]$.

Notice that the current can influence the tension, horizontal and vertical force components, mooring line angle and deformation. From Figure 9.2, a larger current speed has more remarkable influence to tension measurement. The influence from the current direction is also considerable see Figure 9.3. When the current goes vertically with the mooring line horizontal displacement, the current almost do not influence the tension. However, the angle between the current and the mooring line displacement is $\frac{\pi}{4}$ or $\frac{3\pi}{4}$, the current has the largest influence to the tension.

The current also influence the end angle sinusoidally. When the current goes in the same direction from the bottom to the top end, the angle variance reaches its negative maximum. The variance will reach its maximum when the current goes in the opposite direction. With the increasing magnitude of the current speed, the angular variance enhances. But, the current direction have little influence to this angle when the current goes vertically to the mooring line's horizontal displacement. The $\frac{\pi}{2}$-direction current has larger influence than the inline current, since the effective length for the 0-direction current is the water depth d, while it is the cable length s for the $\frac{\pi}{2}$-direction current. Obviously, $s > d$.

This is the same as expected. The current deforms the mooring line and finally influences the top end angle. Additionally, larger current speed results in greater 0-direction force, which heightens the deformation, and thereafter influences the top end angle. Since the deformation of the current is subtle to the original mooring line, the augment in the vertical direction hardly effects the top end angle after a 3D angular transformation.

The mooring line has the largest deformation at the middle part of the mooring line. This is because both the top end and the bottom end are fixed. The influence increase quadratically with the augment of the current speed. The deformations in $v_c = 0.5m/s$ are almost 25 times of those in $v_c = 0.1m/s$. The same quadratically happens to the other plots, which are not included. The reason is that the 0-direction current load is quadratically proportionally to the current speed due to the Morison's equation.

All the simulation results verify Proposition 5.2.

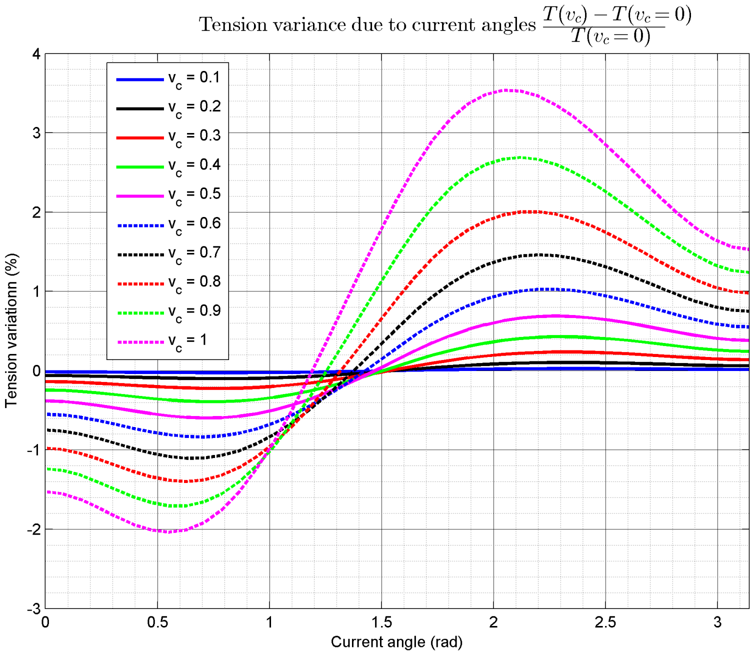


Figure 9.2: Tension variance due to the current direction.

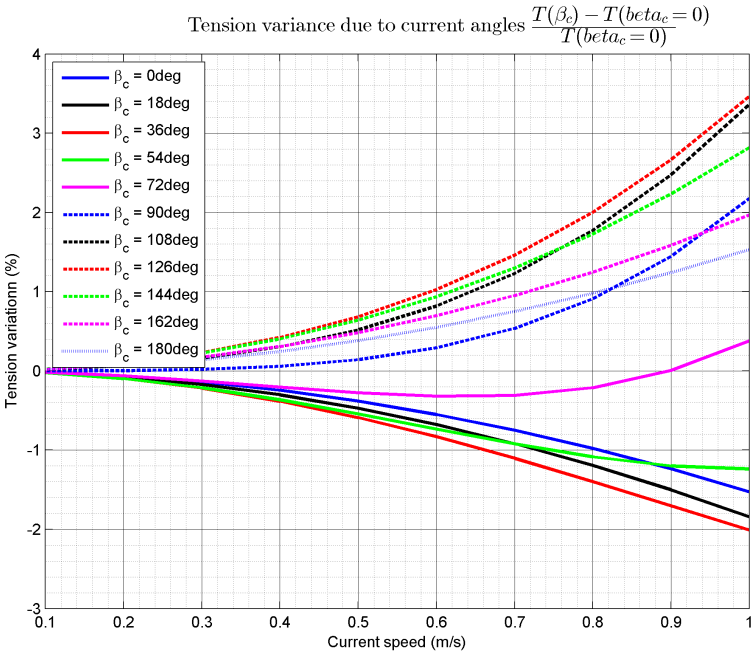


Figure 9.3: Tension variance due to the current speed.

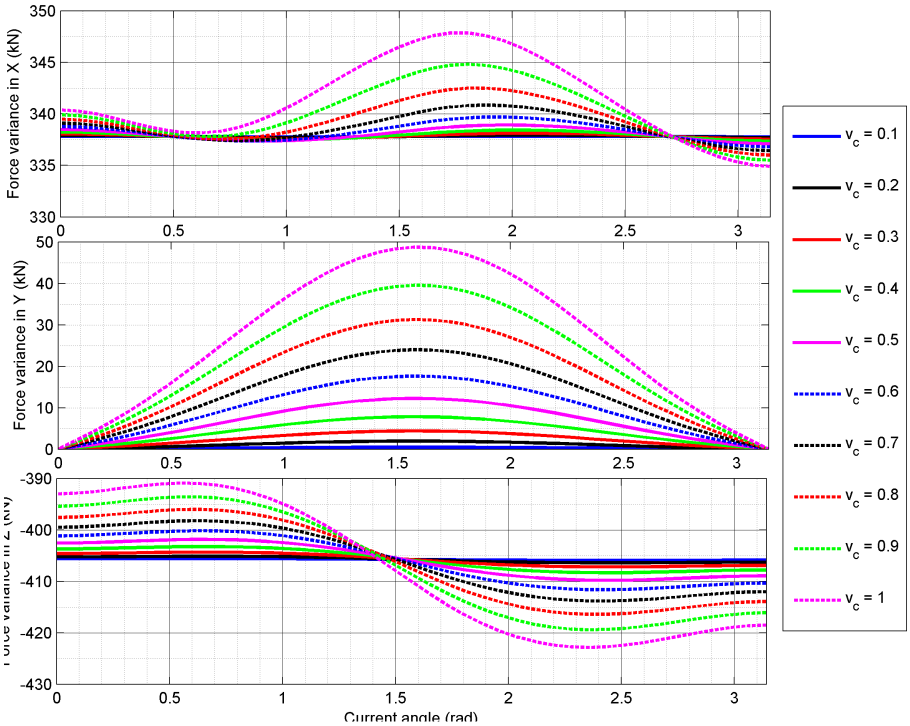


Figure 9.4: Force component variances at the top end due to the current direction.

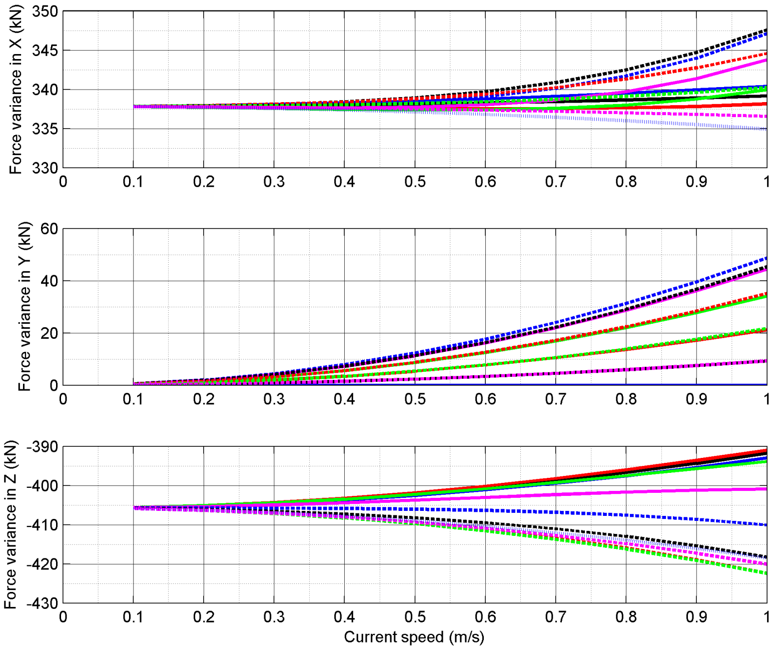


Figure 9.5: Force component variances at the top end due to the current speed.

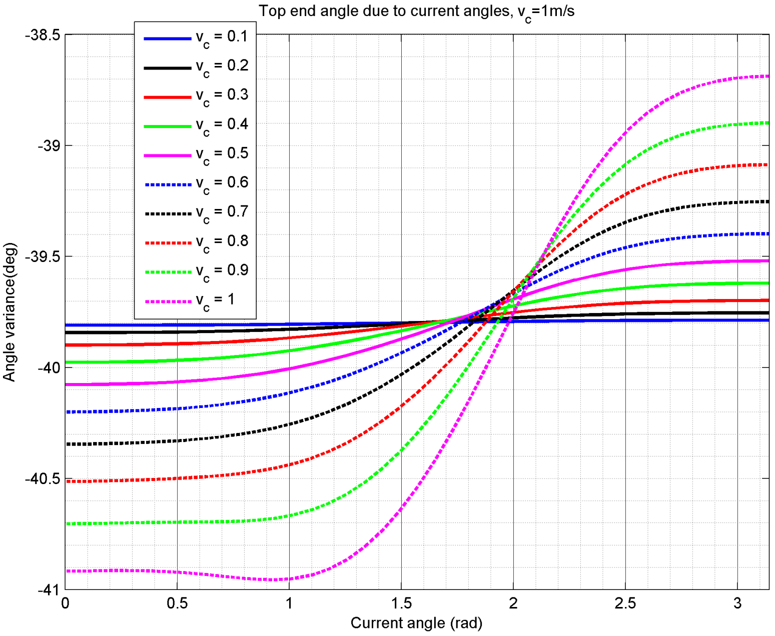


Figure 9.6: The angle variance between the mooring line and the z axis at the top end variance due to the current direction.

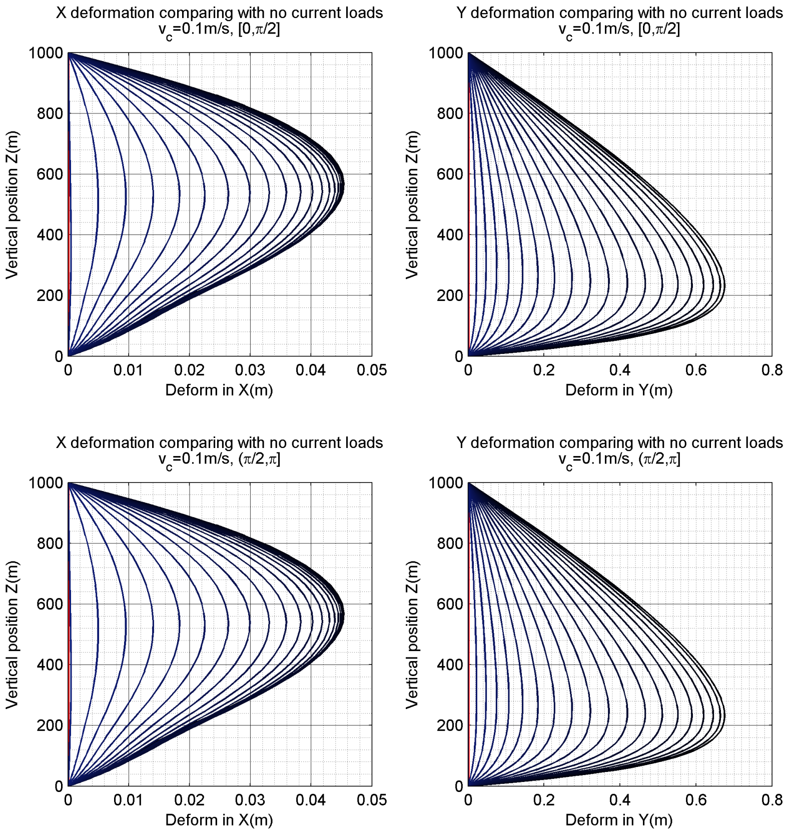


Figure 9.7: The deformation due to the current direction, $v_c = 0.1\, m/s$.

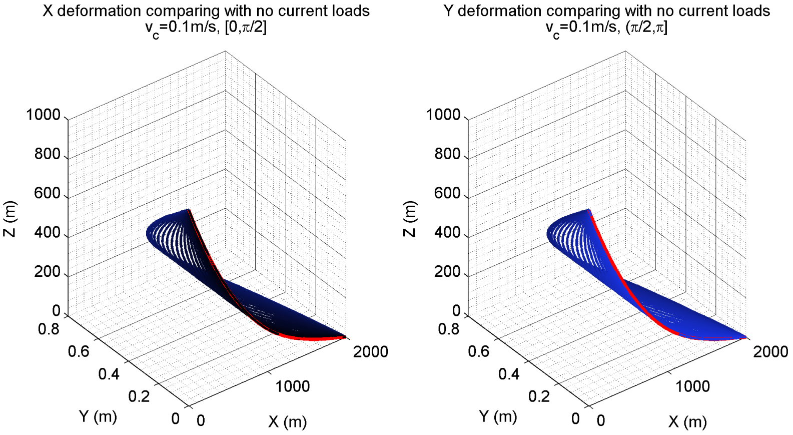


Figure 9.8: The deformation in 3D due to the current direction, $v_c = 0.1\, m/s$.

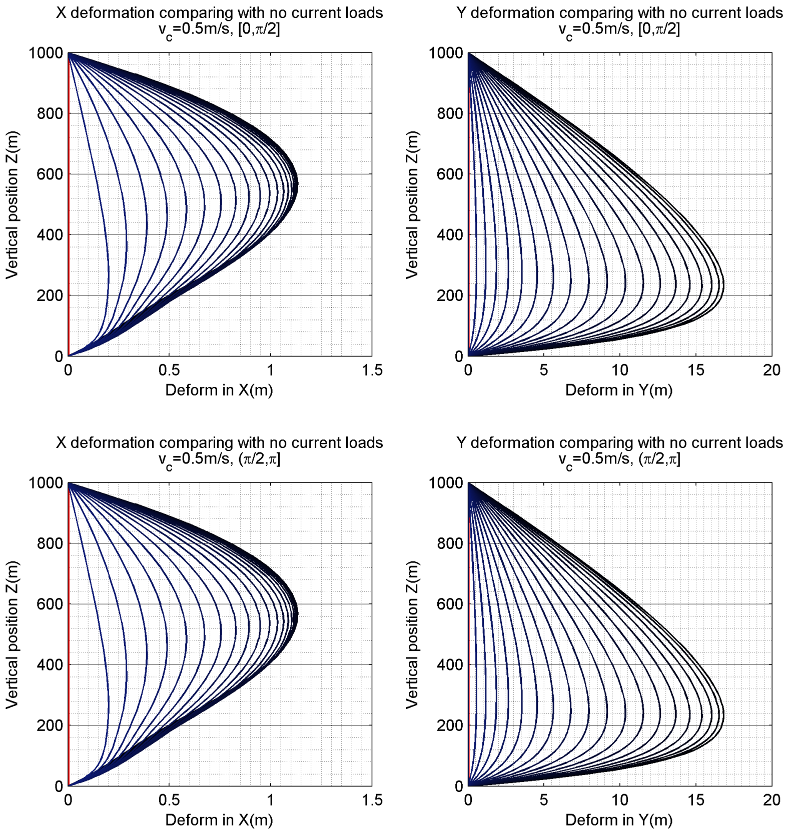


Figure 9.9: The deformation due to the current direction, $v_c = 0.5\,m/s$.

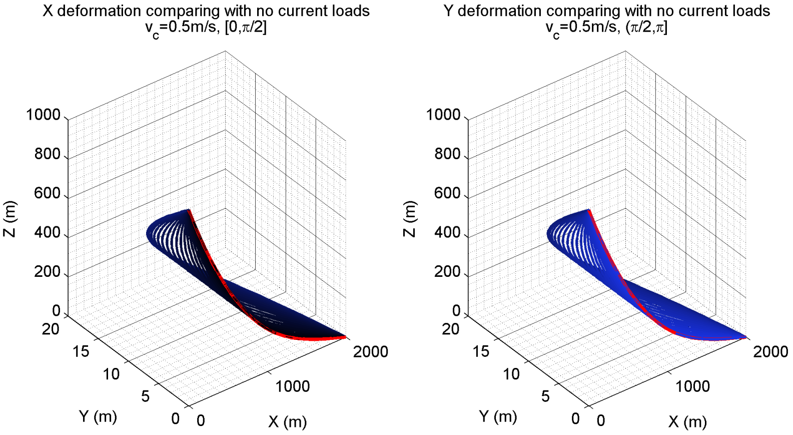


Figure 9.10: The deformation in 3D due to the current direction, $v_c = 0.5\,m/s$.

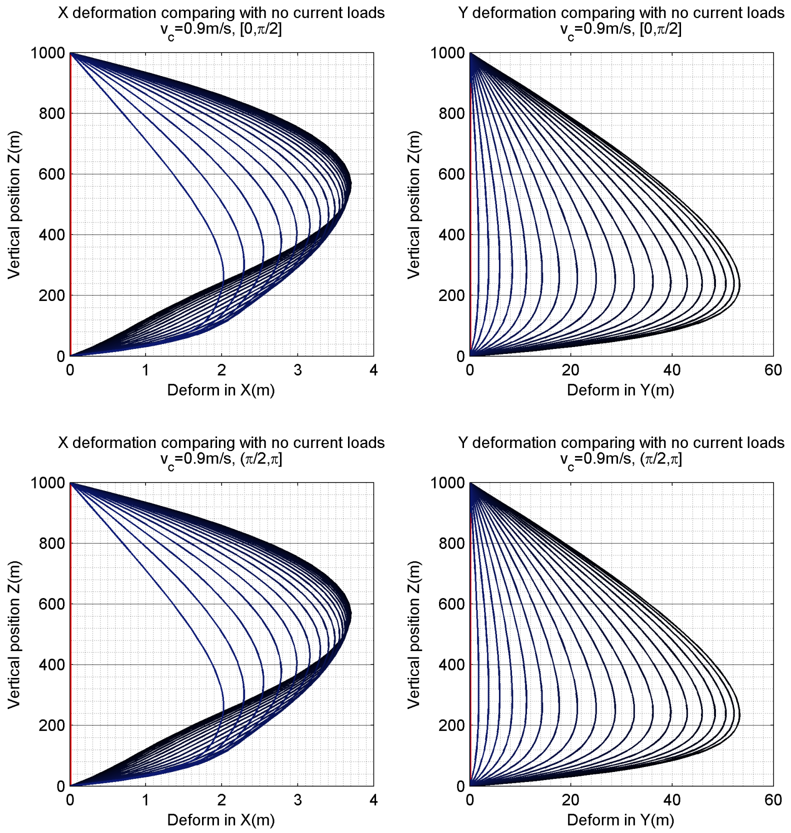


Figure 9.11: The deformation due to the current direction, $v_c = 0.9\,m/s$.

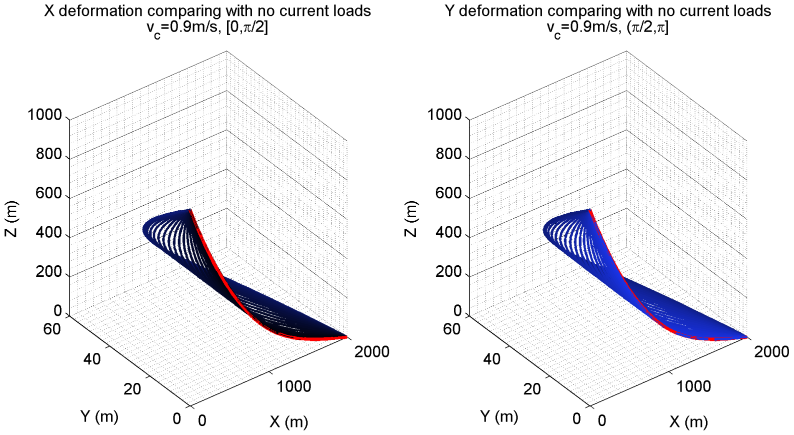


Figure 9.12: The deformation in 3D due to the current direction, $v_c = 0.9\,m/s$.

9.2 Results of Simulation II. Riser model verification

Simulation results show that the added mass coefficient C_M does not influence the static analysis.

9.2.1 Simulation II.1 TTR model with changed current velocity

Figure 9.13 and 9.14 show the influence of the riser deformation caused by a group of current speed with an interval of 0.3 m/s. We notice that

- the deformation and end angle influenced by the current speed,
- the curve analytical solution has a good fitting performance to the curves from RIFLEX. The error of the end angles enhance with the magnitude of the current speed,
- end angles are quadratically proportional to the current velocity, which is demonstrated in Lemma 7.5.

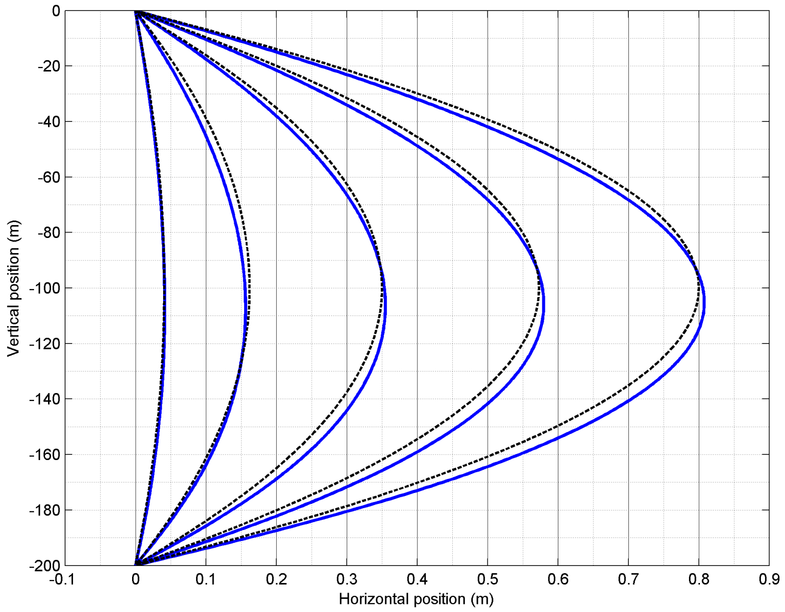
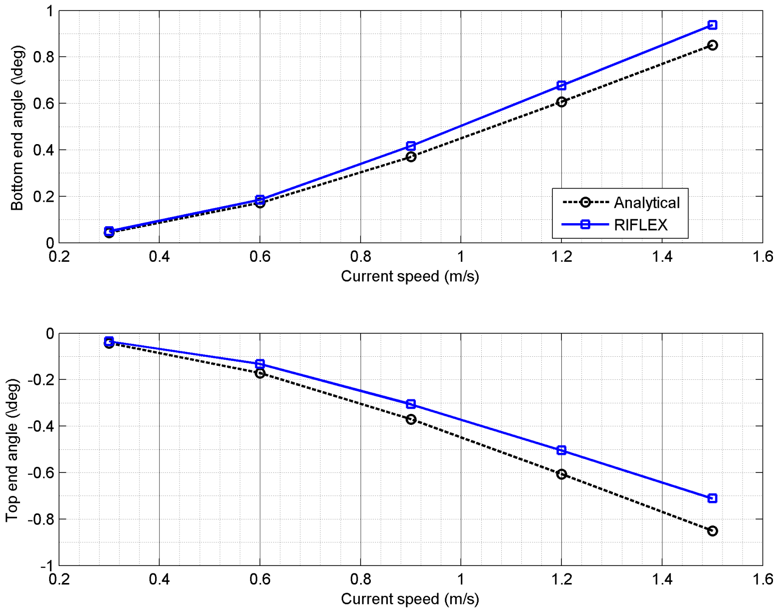


Figure 9.13: Static configuration and enveloping curves in RIFLEX (-) and analytical (- -) for uniform current. Current speed: 0.3, 0.6, 0.9, 1.2, 1.5 m/s.



[H]

Figure 9.14: End angles in RIFLEX (-) and analytical (- -) for uniform current. Current speed: 0.3, 0.6, 0.9, 1.2, 1.5 m/s.

9.2.2 Simulation II.2 TTR model with changed top tension

The influence of the top tension is represented in Figure 9.15 and 9.16. The error between the analytical solution and the curves from RIFLEX decreases with the increasing of the top tension.

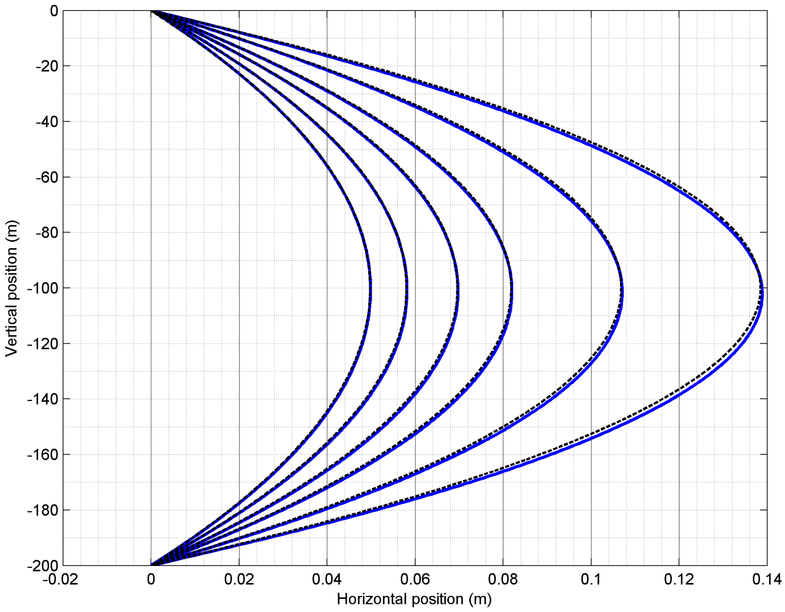


Figure 9.15: Static configuration and enveloping curves in RIFLEX (-) and analytical (- -) for uniform current with different top tension.

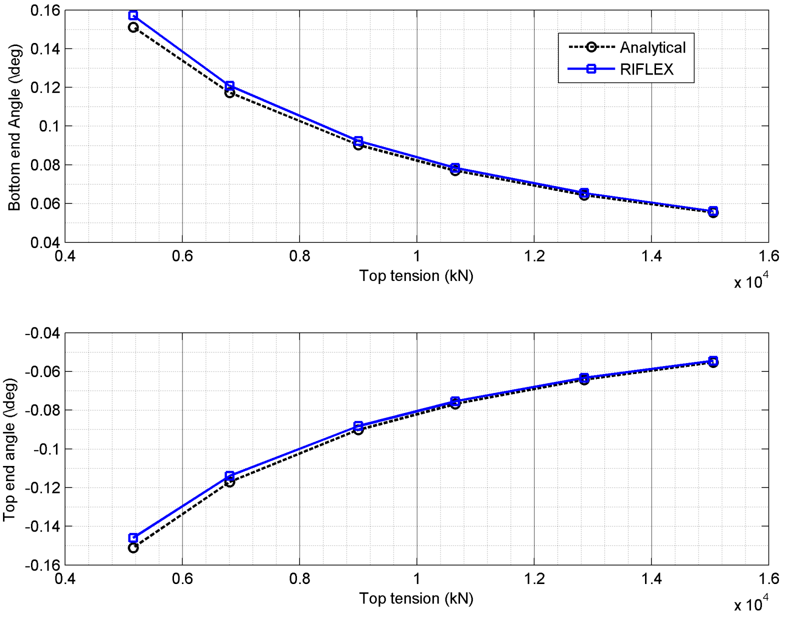


Figure 9.16: End angles in RIFLEX (-) and analytical (- -) for uniform current with different top tension.

9.2.3 Simulation II.3 TTR model with changing current direction

When the vessel has a $3m$ ($3/200 = 1.5\%$) motion, the curve fitting becomes less accurate. The trend of the two sets of results are the same. However, there exists a relatively large bias, $0.4deg$, for the bottom end angle.

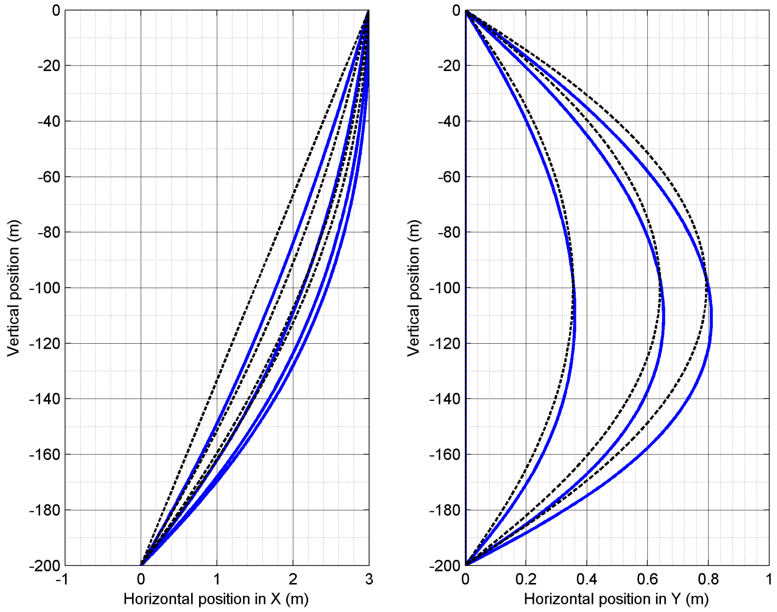


Figure 9.17: Static configuration and enveloping curves in RIFLEX (-) and analytical (- -) for uniform current. Current speed: 1m/s, current direction: 0, 30, 60, 90 deg.

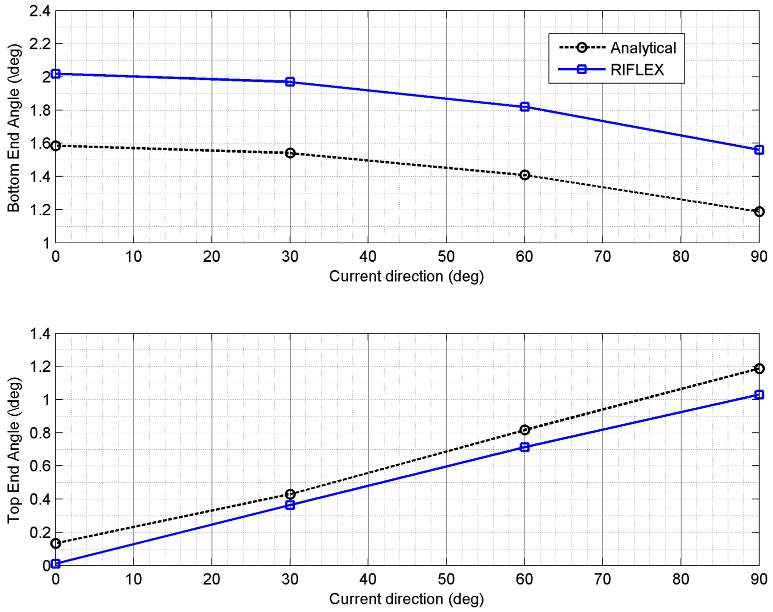


Figure 9.18: End angles in RIFLEX (-) and analytical (- -) for uniform current. Current speed: 1m/s, current direction: 0, 30, 60, 90 deg.

The bias can be caused by various factors. First of all, the riser has weight in water, while we assume the riser weight can be negated. Since a riser only provides limited restoring force to the vessel, which is regardless during design process. Whereas, the FEM commercial software definitely considers the influence, for example, the self weight, which can cause axial deformation and other influence. Even we add an additional tension to the top tension, the coefficient is just approximate. Secondly, there are some unknown parameters, which can also effects the final results.

From all the three simulations above, a primary conclusion is that the analytical solution based on beam theory shows good similarity of the riser in the static analysis. Theorem 7.5 and Theorem 7.6 are verified.

Chapter 10

Results: Fault Detection

10.1 Results of Simulation III. Mooring line breakage based on supervisory control

10.1.1 Simulation III.1 Supervisory control with 4 mooring lines

The results from simulation of Case I are shown from Figure 10.1 to Figure 10.4. Before a TAPM system is functional, the state estimates and integral action should converge to steady state performance. This initial phase may take tens of minutes. The presented simulations start after this initial transient phase is completed.

From Figure 10.1 and 10.4, we observe that the system detects the line break after 20 seconds. At this moment, the supervisor activates the corresponding PID controller, and the vessel drifts only about 10 meters before the new controller effectively brings the position back within the safe region, see Figure 10.2.

Therefore, the supervisor control perfectly works for the TAPM with four mooring lines. This is because the residual signal is distinguishable enough when any of the mooring line breaks.

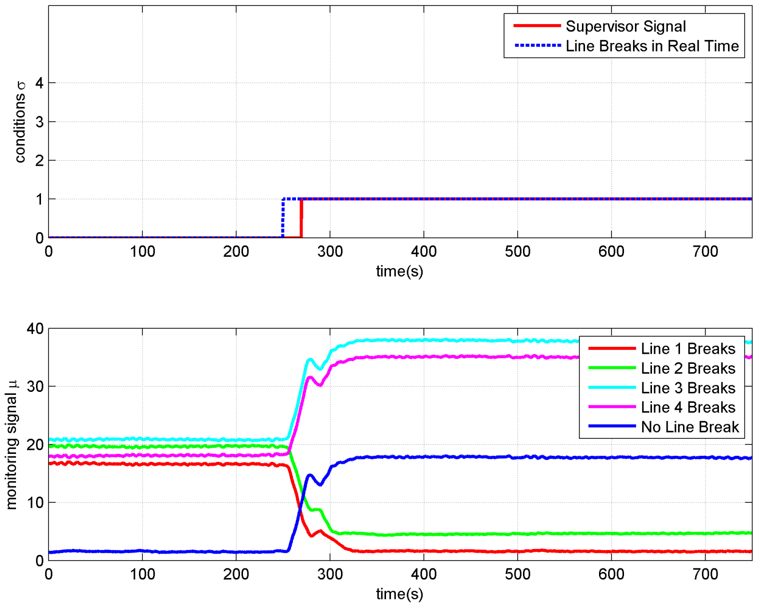


Figure 10.1: Switching logic outputs. Line 1 breaks in Simulation III.1.

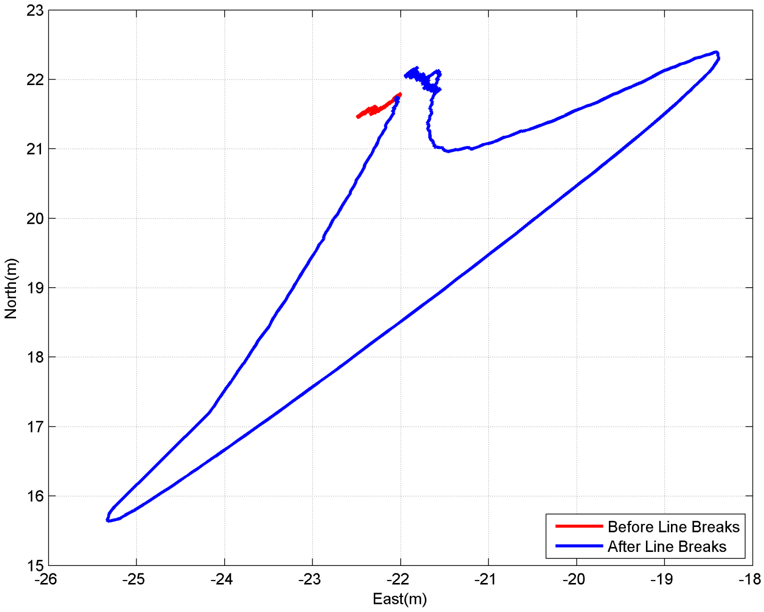


Figure 10.2: Horizontal position of the FPSO. Line 1 breaks in Simulation III.1.

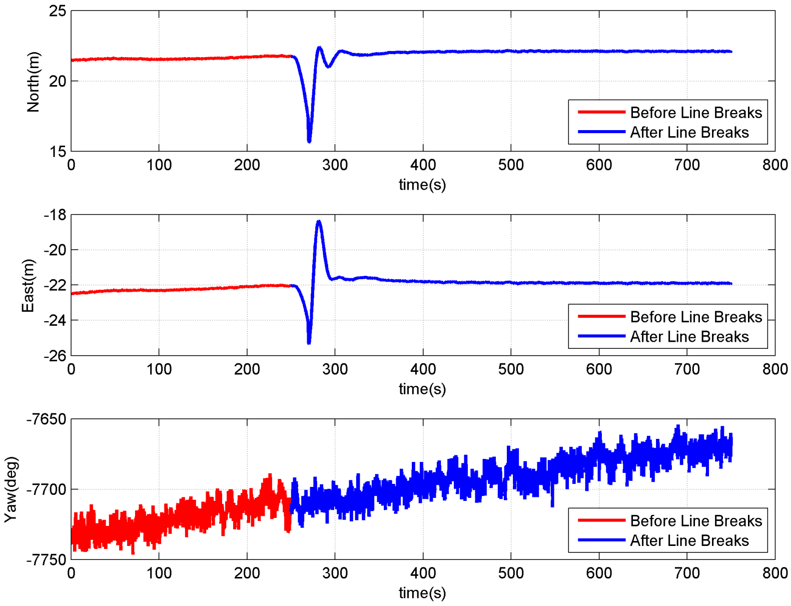


Figure 10.3: Position and rotation in time domain. Line 1 breaks in Simulation III.1.

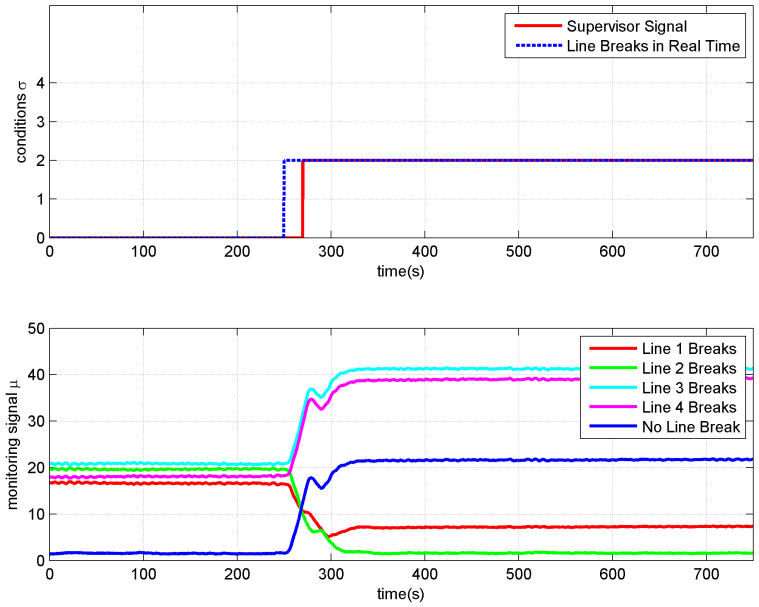


Figure 10.4: Switching logic outputs. Line 2 breaks in Simulation III.1.

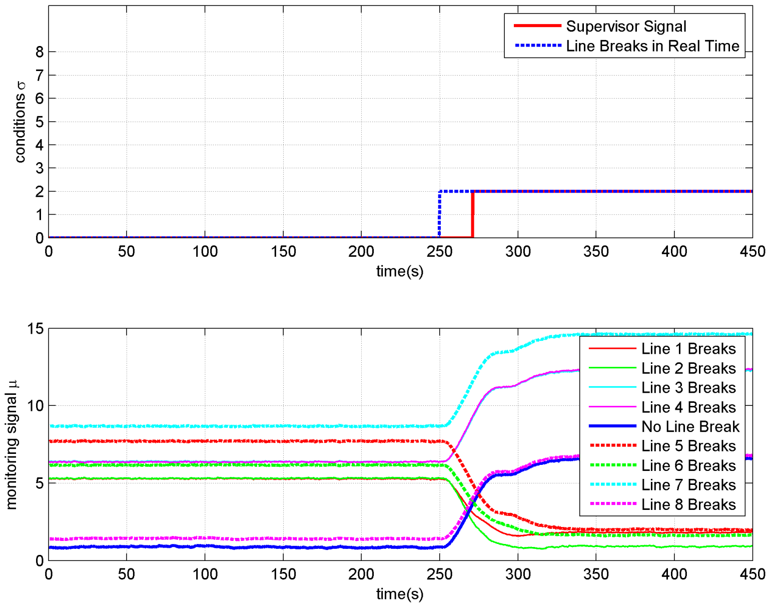


Figure 10.5: Switching logic outputs. Line 2 breaks in Simulation III.2.

10.1.2 Simulation III.2 Supervisory control with 8 mooring lines

Figure 10.5 to Figure B.32 present the results for Simulation III.2. Additional results are shown in Appendix B.3.

- Figure 10.5 is the simulation result when Line 2 breaks. The correct controller is activated about 20 seconds after line break. The decrease rate of the monitoring signal from failure line is faster than any other signals.
- Figure 10.6 and 10.7 show that the vessel drift stays in an accepted region.
- Figure 10.8 shows the responses when Line 7 breaks. The fault diagnosis is slower with a faulty isolated mode at the first 70 seconds after the break. Monitoring signal from Line 4 decreases faster at the beginning 40 seconds. But the drift is still not remarkable, see Figure 10.9.

It has been noticed that increasing current velocity affects the performance of the proposed fault detection methodology. The reason for this is that the current velocity, v_c is assumed to be negligible in the observer. Mitigation of this issue is left for further work.

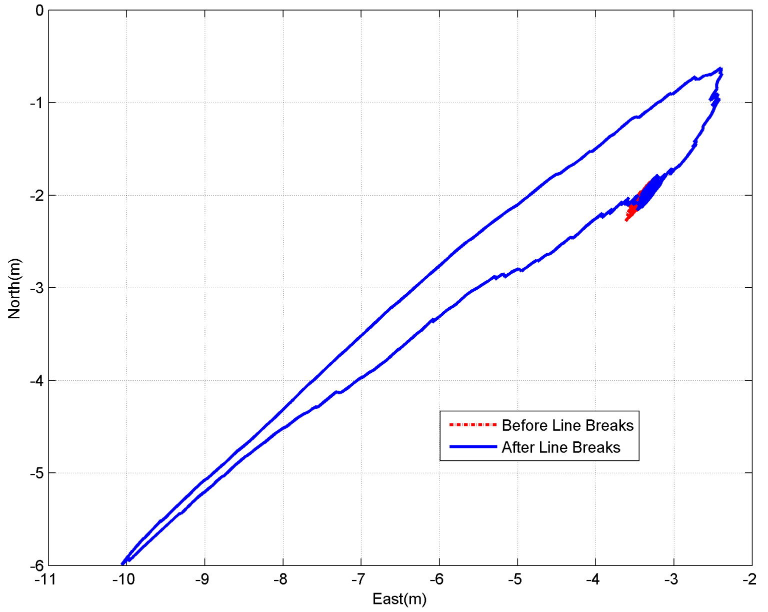


Figure 10.6: Horizontal position of the FPSO. Line 2 breaks in Simulation III.2.

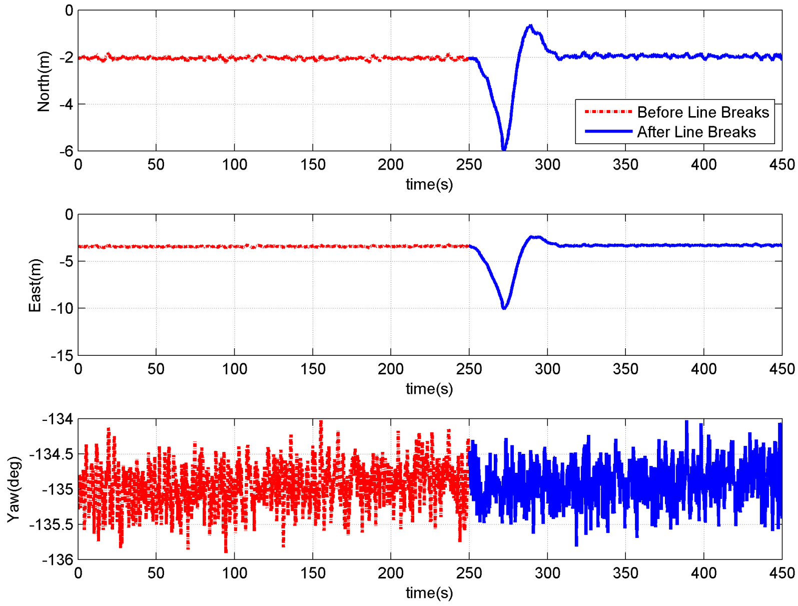


Figure 10.7: Position and rotation in time domain. Line 2 breaks in Simulation III.2.

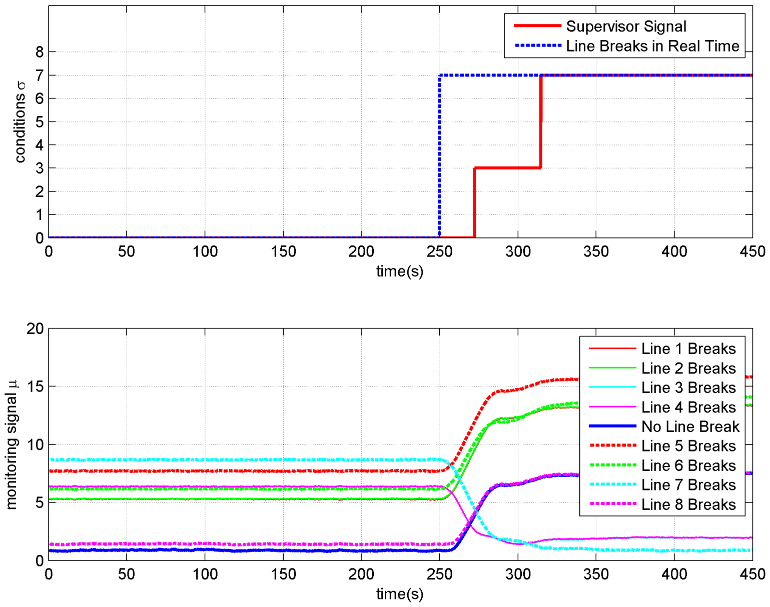


Figure 10.8: Switching logic outputs. Line 7 breaks in Simulation III.2.

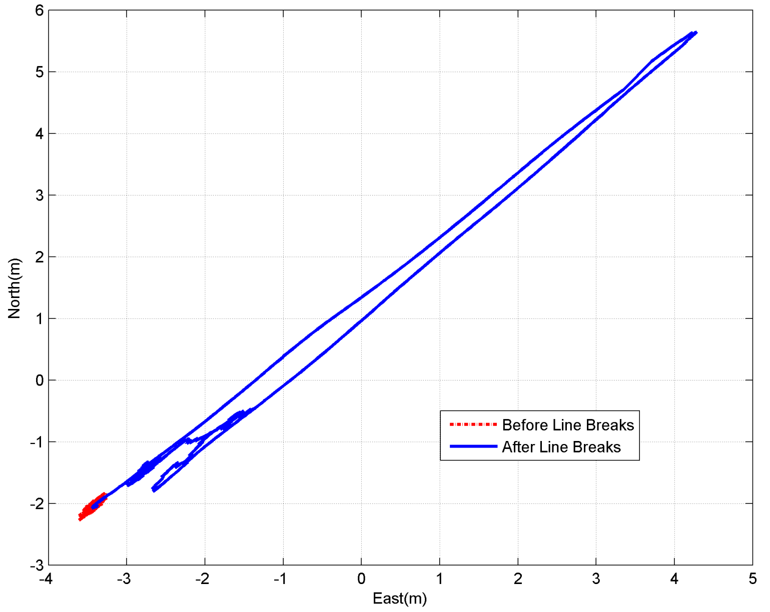


Figure 10.9: Horizontal position of the FPSO. Line 7 breaks in Simulation III.2.

10.2 Results of Simulation IV. GPS failure detection based on tension measurement

The simulation results are shown in Figure 10.10. After the measurement fault on the GPS position happens, the controller quickly reconfigures the position information to tension-based localization estimates. The system is still stable. When the fault disappears at 4000 seconds, the controller recovers to the normal condition with the GPS measurements. The tension-based localization algorithm then has a more rapid oscillation than the GPS-based positions. However, the tension-based localization method provides reliable position signals after the fault happens, and the moored FPSO avoids a drive-off.

Remark:[Remark 1:] After the GPS fault happens, it takes four seconds to detect the faults. This delay is due to the integrator process in the observer, as well as the defensive logic in the diagnosis process. The reaction speed depends on the preset threshold values. Such delay also happens to the recovery process.

Remark:[Remark 2:] The simulation results prove the effectiveness of the proposed fault diagnosis strategy. The moored vessel remains in stable performance with only tension-based reference position.

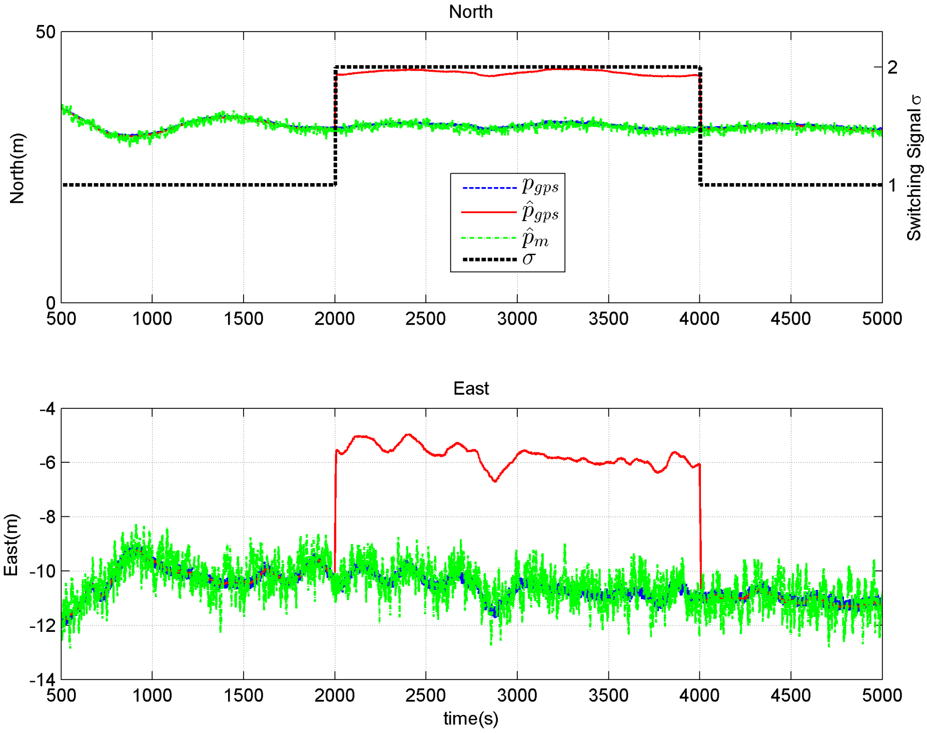


Figure 10.10: NED position of the TAPM system with GPS fault happening in [2000,4000] seconds.

Chapter 11

Results: Tension-based Localization

11.1 Results of Simulation V Simplified tension-based localization

11.1.1 Simulation V.1a Availability of tension-based localization

From Fig. 11.1, we observe the value of the estimations from the mooring tension is quite close to the real GPS signals. From this simulation, the tension-based localization algorithm is applicable to provide position reference information to a moored vessel without noise. Additionally, we notice that the estimation error is smaller near the equilibrium point, and increases when the vessel is further away from the equilibrium point.

11.1.2 Simulation V.1b Tension-based localization performance with nonlinear passive observer

From Fig. 11.2 and Fig. 11.3, we notice that the position estimation error is bounded within a small range. The estimator with the NPO gives better estimation with smaller residuals. The errors between the tension-based position signals and the GPS signals are limited within 1.5 meter. The delay is not considerable. Therefore, the NPO gives good estimates of the vessel's position based merely on tension measurements.

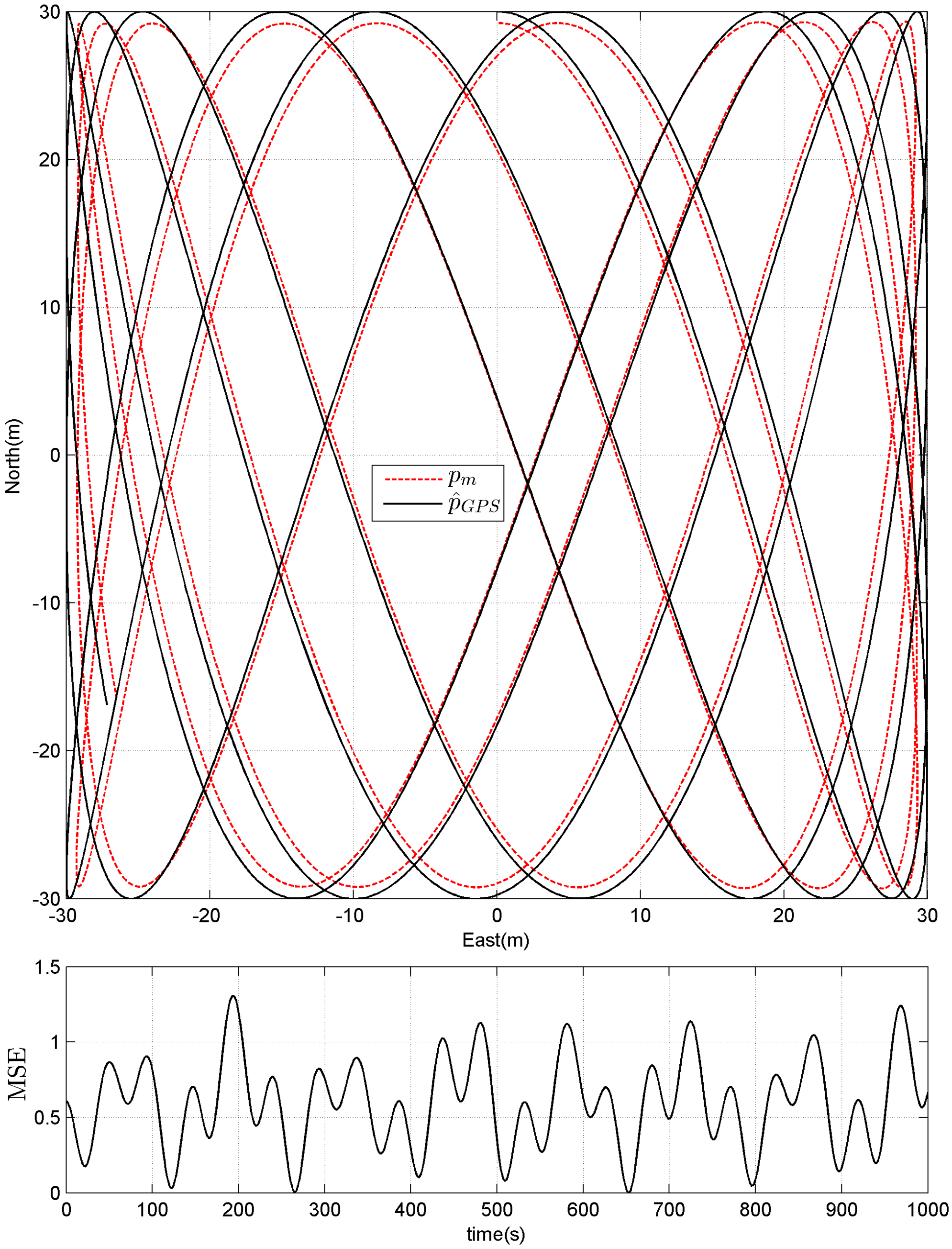


Figure 11.1: Position estimate with no tension measurement noise.

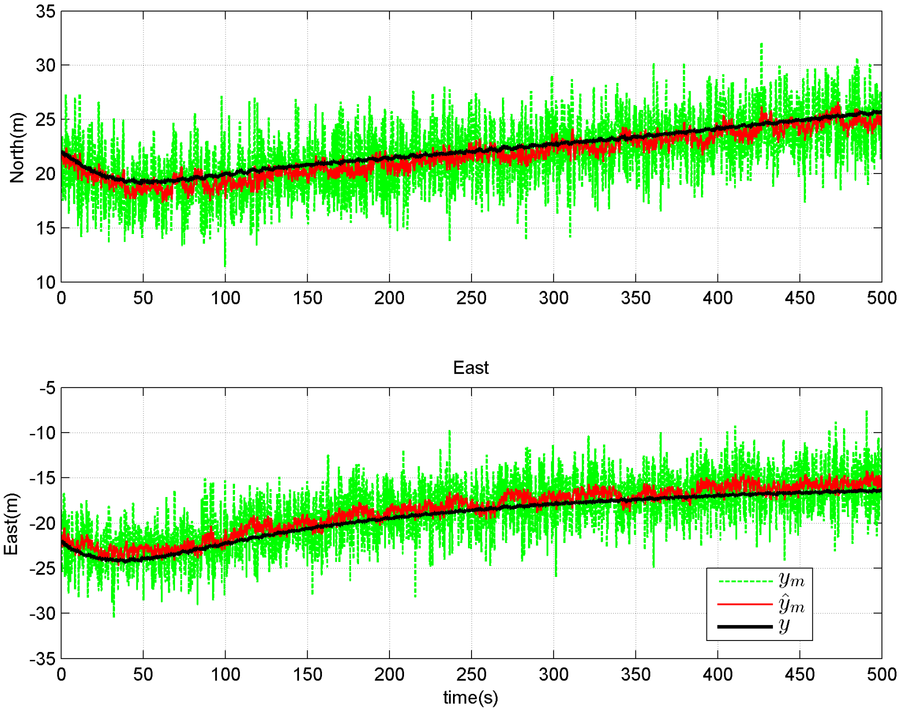


Figure 11.2: Position estimate test with tension measurement noise.

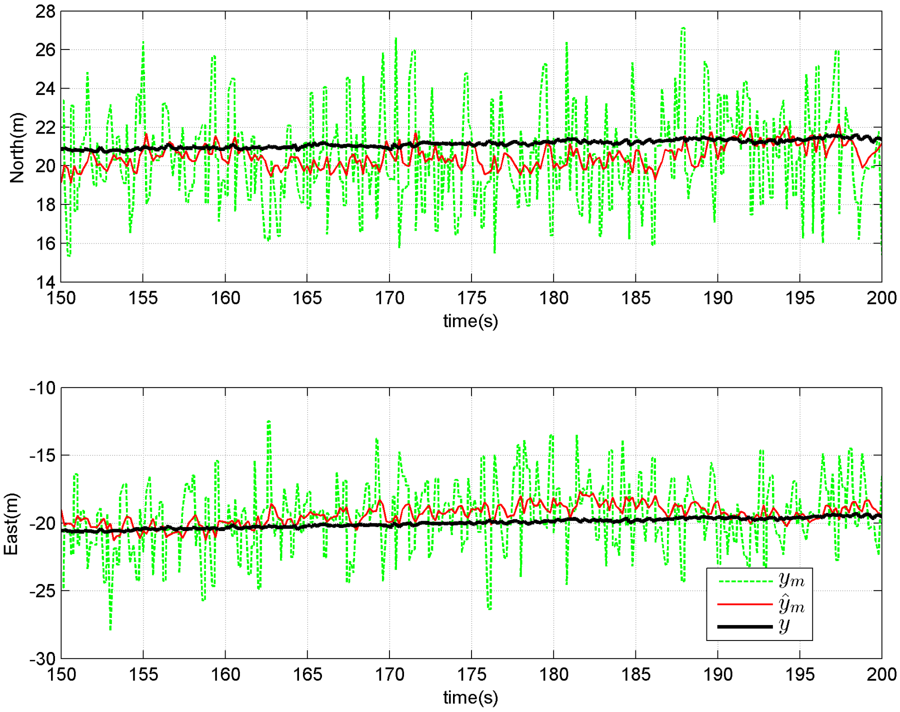


Figure 11.3: Position estimate test with tension measurement noise (zoomed in).

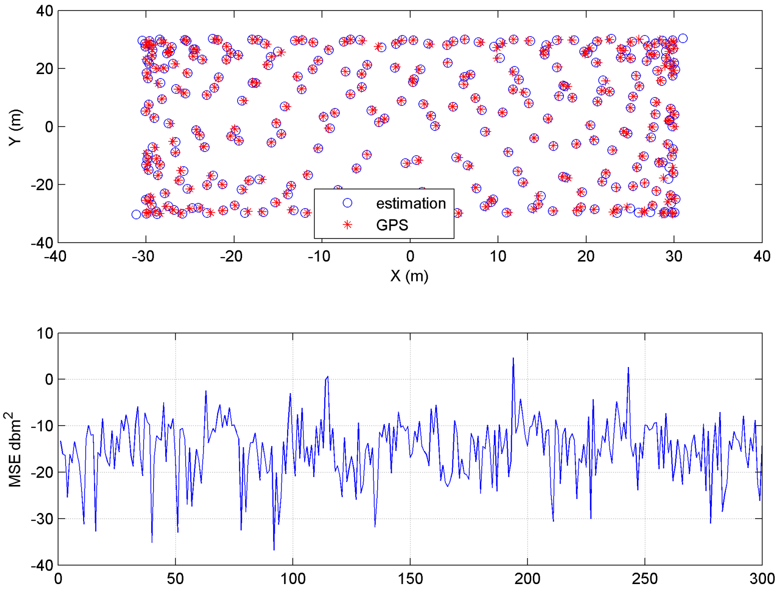


Figure 11.4: The localization performance without wave-induced heave.

11.1.3 Simulation V.2 Influence analysis of the tension measurement noise

A parametric analysis is conducted to test the influence of the tension measurement noise to the localization performance. Since to locate the unknown vessel with the known anchors is an equivalent question, we analyse the tension cell noise based on the performance of localization of the upper vessel. Maximum likelihood criterion is applied here, which is given by

$$\min_{\hat{\boldsymbol{x}}_j^t} \quad \Sigma(|\boldsymbol{x}_i^a - \hat{\boldsymbol{x}}_j^t| - d_{ij})^2. \tag{11.1}$$

For a specific mooring tension variance, the simulation result is shown in Figure 11.4. The overall simulation result is shown in Figure 11.5. In the figure, every point is the mean value of a group of 300 independent simulations. The unit, dbm^2, comes from $10log_{10}(\cdot)$. We notice the variance of localization is proportional to the variance of the tension measurement.

11.1.4 Simulation V.3 Second order cone programming localization performance

Figure 11.6 and 11.7 show the simulation result of the SOCP approach. It is obviously that the estimation performance is quite good in Figure 11.6, when the TP moves randomly within a

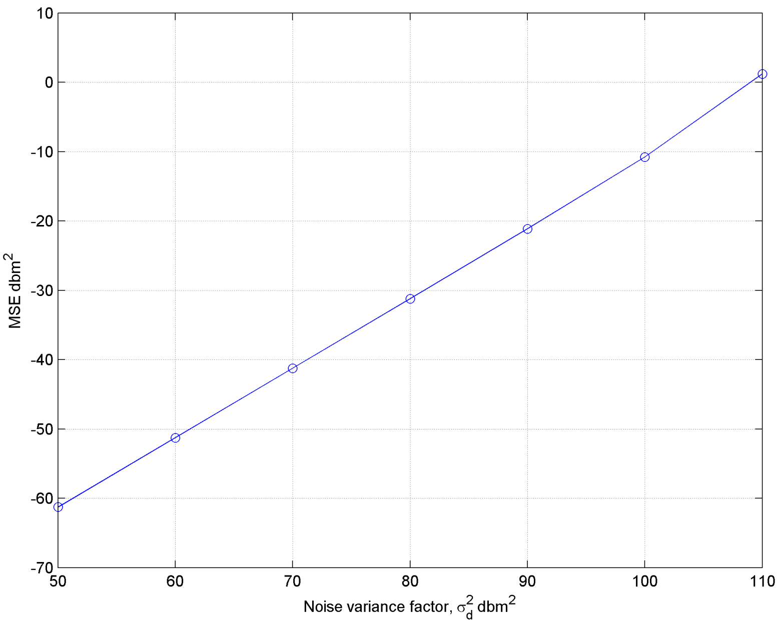


Figure 11.5: A parameter analysis between the variance of the tension noise and the localization performance.

region of 100 meters. Notice that this is a scaling simulation. This simulation is just to show this method is applicable, and the program works properly. The best estimation only means the position of the anchors when we installed them. It does not input to the algorithm. However, this is never practical for a TAPM.

In a practical situation, the result is represented in Figure 11.7. SOCP does not work well as expected. When the vessel keeps running in a small accepted region of 10 meters, the convex optimization only give out a local optimum, and the results are insufficient to provide accurate localization information. The main reason is the relatively small accepted range of the upper moored vessel.

11.2 Simulation VI. Simultaneous localization performance with unknown anchors

The simulation results is shown in Fig 11.8 and Fig 11.11. From the curves, we notice that the position estimation coverages to the real values within 200 steps.

In the sensor network, the anchors are independent. In another words, we can detect whichever anchor we want. Therefore, it is applicable to all vessels equipped with tension cells.

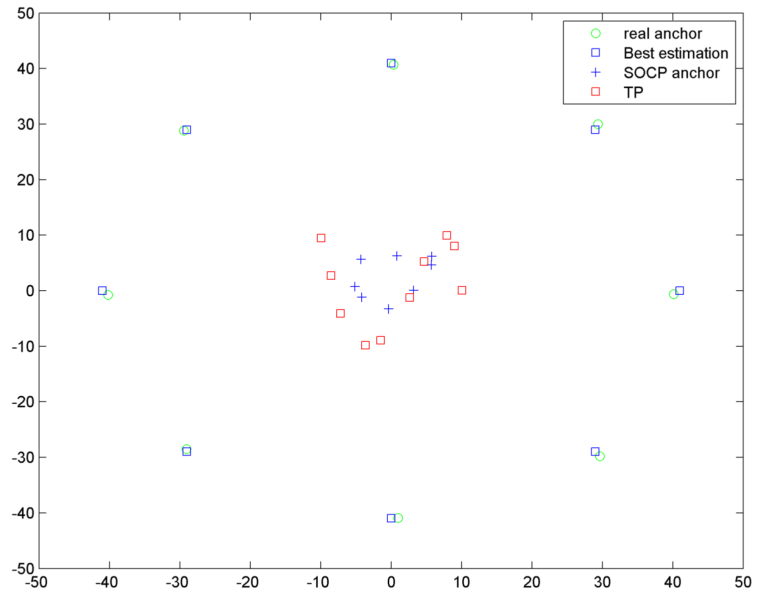


Figure 11.6: Localization performance of SOCP when the vessel run in a region of 100 meters.

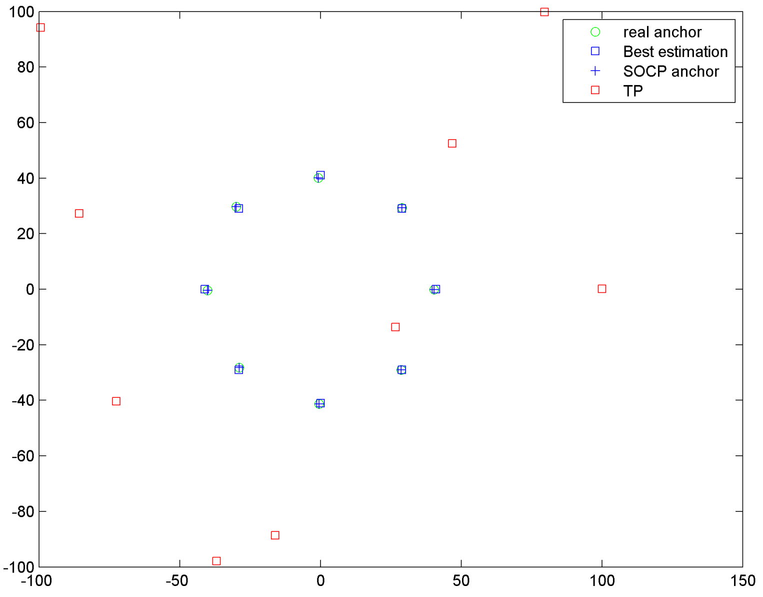


Figure 11.7: Localization performanceof SOCP when the vessel run in a region of 10 meters.

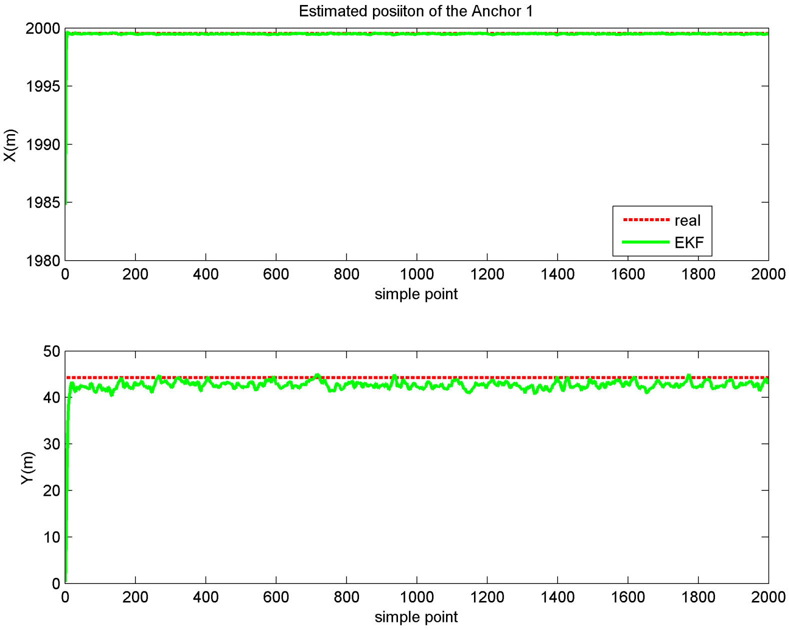


Figure 11.8: Anchor 1 position estimation with LF and WF.

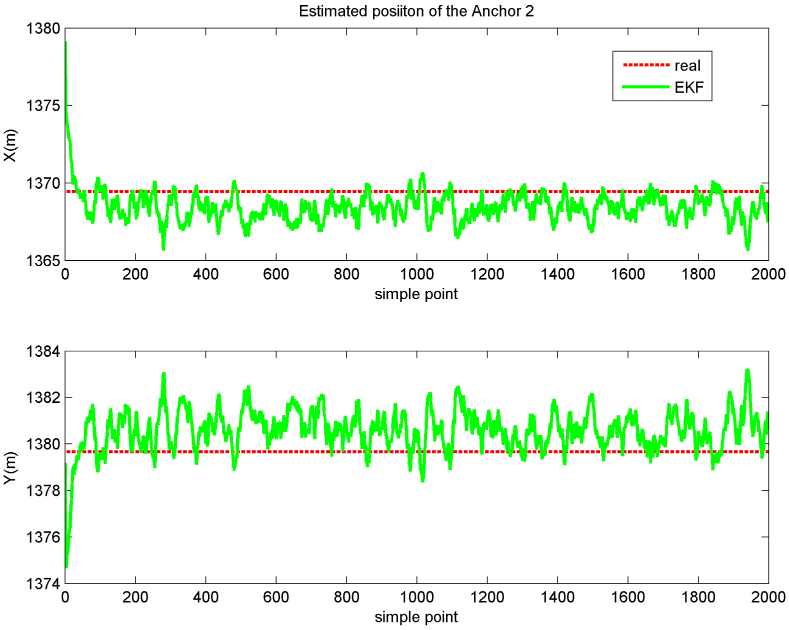


Figure 11.9: Anchor 2 position estimation with LF and WF.

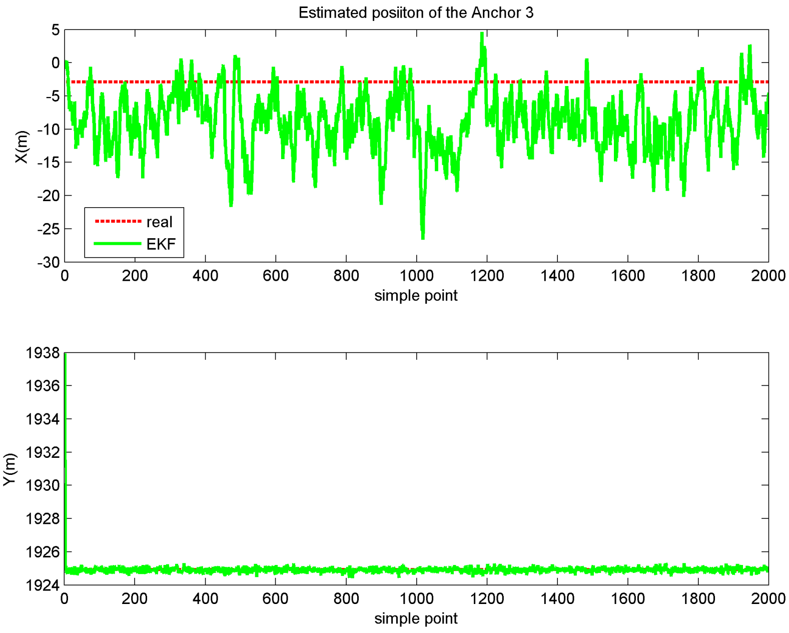


Figure 11.10: Anchor 3 position estimation with LF and WF.

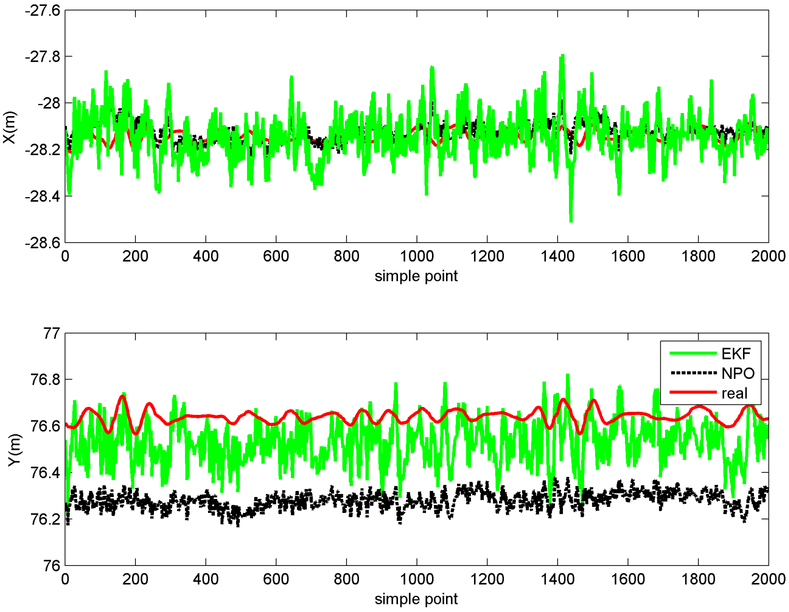


Figure 11.11: TP position estimation with LF and WF.

We notice that there exists a trivial bias between the estimated and real position. This is because the sea state parameter estimation are not totally accurate. Additionally, the current speed influences the passive filter. Finally, the stiffness is not the same for all mooring lines. The mooring line in the direction where the vessel's new equivalent point goes can have a much smaller stiffness. It can influence the covariance of the estimated range measurement noise, and then influences the EKF.
Therefore, the EKF-SLAM approach is applicable to a TAPM.

Chapter 12

Conclusion and Future Works

12.1 Conclusion

The main achievements are listed as follows.

- The development of a Simulink lib for TAPM.

- A comparison between different mooring models is conducted.

- A fault-tolerant control scheme based on an estimator-based supervisory control methodology is presented. An online fault diagnosis algorithm is designed based on residual signals from a bank of pre-designed observers.

- A simplified position algorithm for TAPM systems depended on tension measurements.

- A sensor network scheme is built for a TAPM.

- A complete model is deduced which introduced EKF-SLAM algorithm to TAPM system to locate uncertain anchors.

- A set point algorithm for TAPM after switching work condition is deduced.

- An analytical scheme is built to optimize the control of the end angle of a riser in constant-speed current profile.

- A new setpoint chasing algorithm is proposed based on multi-objective optimization, which minimize the consumed energy, reduce the risk of structural damage.

12.2 Suggestion for future works

Future works are shown as follows.

- The application of the methodology developed to a higher fidelity model of moored vessels. The effect of environmental loads on the detection performance will be further studied, as well as a quantification of false alarms and unsuccessful detection,

- Due to the influence of the current, a non-line-of-sight localization algorithm is needed to develop, which can enhance the value of this technique in practical applications. The development of tension-based posref with non-line-of-sight tension measurement which includes the current influence, mooring line dynamics, sensor bias, etc.

- Test of the setpoint chasing algorithm with experiments and practical data, such as data mining.

Bibliography

Aalbers, A., Janse, S., De Boom, W., et al. (1995). Dp assisted and passive mooring for fpso's. In *ANNUAL OFFSHORE TECHNOLOGY CONFERENCE*, volume 2, pages 281–281. Offshore Technology Conference.

Aamo, O. M. (1999). *Adding mooring systems to the ABB Integrated Vessel Simulator – Implementation*. Department of Engineering Cybernetics, NTNU, Trondheim, Norway.

Aamo, O. M. and Fossen, T. I. (1999). Controlling line tension in thruster assisted mooring systems. In *Control Applications, 1999. Proceedings of the 1999 IEEE International Conference on*, volume 2, pages 1104–1109. IEEE.

Aamo, O. M. and Fossen, T. I. (2000). Finite element modelling of mooring lines. *Mathematics and computers in simulation*, 53(4):415–422.

ABS (2014). Guide for building and classing mobile offshore units.

Afraimovich, E. L., Kosogorov, E. A., and Leonovich, L. A. (2000). The use of the international gps network as the global detector (globdet) simultaneously observing sudden ionospheric disturbances. *Earth, planets and space*, 52(11):1077–1082.

Afriana, R. (2011). Coupled dynamic analysis of cylindrical fpso, moorings and riser based on numerical simulation.

API, R. (2005). 2sk (2005) design and analysis of stationkeeping systems for floating structures. *American Petroleum Institute*.

Balchen, J. G., Jenssen, N. A., Mathisen, E., and Sælid, S. (1980). A dynamic positioning system based on kalman filtering and optimal control.

Barth Berntsen, P., Aamo, O. M., and Leira, B. J. (2008). Thruster assisted position mooring based on structural reliability. *International Journal of Control*, 81(9):1408–1416.

Berntsen, P. I. B., Aamo, O. M., and Leira, B. J. (2009). Ensuring mooring line integrity by dynamic positioning: Controller design and experimental tests. *Automatica*, 45(5):1285–1290.

Berntsen, P. I. B., Aamo, O. M., Leira, B. J., and Sørensen, A. J. (2008). Structural reliability-based control of moored interconnected structures. *Control Engineering Practice*, 16(4):495–504.

Blanke, M., Fang, S., Galeazzi, R., and Leira, B. J. (2012). Statistical change detection for diagnosis of buoyancy element defects on moored floating vessels. In *Proc. IFAC SAFE-PROCESS*, pages 462–467.

Blanke, M., Kinnaert, M., Lunze, J., and Staroswiecki, M. (2006). *Diagnosis and Fault-Tolerant Control, 2nd Edition*. Springer, Berlin, Heidelberg.

Bonnemaire, B., Jensen, A., Gudmestad, O. T., Lundamo, T., and Løset, S. (2007). Challenges related to station-keeping in ice. In *INTSOK Conference, Houston*.

Brodtkorb, A. H., Sørensen, A. J., and Teel, A. R. (2014). Increasing the operation window for dynamic positioned vessels using the concept of hybrid control. In *ASME 2014 33rd International Conference on Ocean, Offshore and Arctic Engineering*, pages V01AT01A046–V01AT01A046. American Society of Mechanical Engineers.

Brown, L. D., Hwang, J. G., and Munk, A. (1997). An unbiased test for the bioequivalence problem. *The annals of Statistics*, pages 2345–2367.

Brown, M. G., Hall, T., Marr, D., English, M., Snell, R., et al. (2005). Floating production mooring integrity jip-key findings. In *Offshore Technology Conference*. Offshore Technology Conference.

Chen, F. (2013). Dynamic design of the relay platform and anchor system. In *The Kuroshio Power Plant*, pages 87–120. Springer.

Colliat, J. et al. (2002). Anchors for deepwater to ultradeepwater moorings. In *Proceedings of the 34th Annual Offshore Technology Conference (OTC'02)*, pages 2695–2703.

Comfort, G., Singh, S., and Spencer, D. (1999). Evaluation of ice model test data for moored structures.

Dareing, D. and Huang, T. (1979). Marine riser vibration response determined by modal analysis. *Journal of Energy Resources Technology*, 101(3):159–166.

Davies, P., François, M., Grosjean, F., Baron, P., Salomon, K., Trassoudaine, D., et al. (2002). Synthetic mooring lines for depths to 3000 meters. *OTC14246*.

Dewey, R. K. (1999). Mooring design & dynamics—a matlab® package for designing and analyzing oceanographic moorings. *Marine Models*, 1(1):103–157.

Di Masi, G., Finesso, L., and Picci, G. (1986). Design of an lqg controller for single point moored large tankers. *Automatica*, 22(2):155–169.

DNV (2010). Dnv-os-e301–position mooring.

Doherty, P., Coster, A., and Murtagh, W. (2004). Space weather effects of october–november 2003. *GPS Solutions*, 8(4):267–271.

dredgingengineering.com (n.d.). Catenary mooring system.

Ehlers, C. J., Young, A. G., Chen, J.-h., et al. (2004). Technology assessment of deepwater anchors. In *Offshore technology conference*. Offshore Technology Conference.

Elman, P., Bramande, J., Elletson, E., Pinheiro, K., et al. (2013). Reducing uncertainty through the use of mooring line monitoring. In *OTC Brasil*. Offshore Technology Conference.

Faltinsen, O. (1993). *Sea loads on ships and offshore structures*, volume 1. Cambridge university press.

Fang, S. and Blanke, M. (2011). Fault monitoring and fault recovery control for position-moored vessels. *International Journal of Applied Mathematics and Computer Science*, 21(3):467–478.

Fang, S., Blanke, M., and Leira, B. J. (2015). Mooring system diagnosis and structural reliability control for position moored vessels. *Control engineering Practice*, 36:12–26.

Fang, S., Leira, B. J., and Blanke, M. (2011). Reliability-based dynamic positioning of floating vessels with riser and mooring system. In *4th International Conference on Computational Methods in Marine Engineering*.

Fang, S., Leira, B. J., and Blanke, M. (2013). Position mooring control based on a structural reliability criterion. *Structural Safety*, 41:97–106.

Fissel, D., Marko, J., and Melling, H. (2008). Advances in marine ice profiling for oil and gas applications. In *Proceedings of the Icetech 2008 Conference*.

Fossen, T. I. (2011). *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons.

Fossen, T. I. and Grovlen, A. (1998). Nonlinear output feedback control of dynamically positioned ships using vectorial observer backstepping. *Control Systems Technology, IEEE Transactions on*, 6(1):121–128.

Fossen, T. I. and Strand, J. P. (1999). Passive nonlinear observer design for ships using lyapunov methods: full-scale experiments with a supply vessel. *Automatica*, 35(1):3–16.

Fossen, T. I. and Strand, J. P. (2001). Nonlinear passive weather optimal positioning control (wopc) system for ships and rigs: experimental results. *Automatica*, 37(5):701–715.

Fylling, I. (2005). Mooropt-trunc-a program for truncation of mooring lines. *MARINTEK Report*, (570002.06):01.

Fylling, I., Larsen, C., Sødahl, N., Ormberg, H., Engseth, A., Passano, E., and Holthe, K. (1995). Riflex–theory manual. *SINTEF report no. STF70 F*, 95219:53.

Garza-Rios, L. and Bernitsas, M. (1999). Slow motion dynamics of turret mooring and its approximation as single point mooring. *Applied ocean research*, 21(1):27–39.

Garza-Rios, L. O. and Bernitsas, M. M. (2001). Effect of size and position of supporting buoys on the dynamics of spread mooring systems. *Journal of Offshore Mechanics and Arctic Engineering*, 123(2):49–56.

Garza Rios, L. O., Bernitsas, M. M., and Nishimoto, K. (1997). Catenary mooring lines with nonlinear drag and touchdown. Technical report, University of Michigan.

Gauthier, S., Elletson, E., et al. (2014). Mooring line monitoring to reduce risk of line failure. In *The Twenty-fourth International Ocean and Polar Engineering Conference*. International Society of Offshore and Polar Engineers.

Ge, S. S., He, W., Voon Ee How, B., and Choo, Y. S. (2010). Boundary control of a coupled nonlinear flexible marine riser. *Control Systems Technology, IEEE Transactions on*, 18(5):1080–1091.

Gobat, J. I. and Grosenbaugh, M. A. (2001). A simple model for heave-induced dynamic tension in catenary moorings. *Applied ocean research*, 23(3):159–174.

Grant, M. and Boyd, S. (2008). Graph implementations for nonsmooth convex programs. In Blondel, V., Boyd, S., and Kimura, H., editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited. `http://stanford.edu/~boyd/graph_dcp.html`.

Grant, M. and Boyd, S. (2014). CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`.

Gudmestad, O. T., Dalane, O., and Aksnes, V. (2009). On the disconnection of a moored floater in hard ice conditions. In *Proceedings of the International Conference on Port and Ocean Engineering Under Arctic Conditions*, number POAC09-137.

Gustafsson, F. (2010). Statistical sensor fusion.

Hamilton, J., Holub, C., Blunt, J., Mitchell, D., Kokkinis, T., et al. (2011). Ice management for support of arctic floating operations. *OTC-22105*.

Hassani, V., Sørensen, A. J., and Pascoal, A. M. (2012a). Evaluation of three dynamic ship positioning controllers: from calm to extreme conditions. In *Navigation, Guidance and Control of Underwater Vehicles*, volume 3, pages 158–163.

Hassani, V., Sørensen, A. J., and Pascoal, A. M. (2012b). Robust dynamic positioning of offshore vessels using mixed-μ synthesis, part ii: Simulation and experimental results,". In *Proc. ACOOG 2012-IFAC Workshop on Automatic Control in Offshore Oil and Gas Production*.

Hasselmann, D., Dunckel, M., and Ewing, J. (1980). Directional wave spectra observed during jonswap 1973. *Journal of physical oceanography*, 10(8):1264–1280.

Haugen, J., Imsland, L., Løset, S., Skjetne, R., et al. (2011). Ice observer system for ice management operations. In *Proc. 21st Int. Offshore (Ocean) and Polar Eng. Conf., Maui, Hawaii, USA*.

Hespanha, J. P. (2001). Tutorial on supervisory control. In *Lecture Notes for the workshop Control using Logic and Switching for the 40th Conf. on Decision and Contr., Orlando, Florida.*

How, B., Ge, S., and Choo, Y. (2009). Active control of flexible marine risers. *Journal of Sound and Vibration*, 320(4):758–776.

Hycalin.nl (n.d.). Mooring systems.

Jeon, S., Cho, Y., Seo, M., Cho, J., and Jeong, W. (2013). Dynamic response of floating substructure of spar-type offshore wind turbine with catenary mooring cables. *Ocean Engineering*, 72:356–364.

Johansen, T. A. and Fossen, T. I. (2013). Control allocation—a survey. *Automatica*, 49(5):1087–1103.

Jorgensen, U. and Skjetne, R. (2012). Dynamic estimation of drifting ice topography using underwater mobile measurements. In *American Control Conference (ACC), 2012*, pages 301–306. IEEE.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45.

Kjerstad, Ø. K. and Skjetne, R. (2014). Modeling and control for dynamic positioned marine vessels in drifting managed sea ice.

Kongsberg (2013). Annual report 2013.

Lavretsky, E. and Wise, K. (2012). *Robust and Adaptive Control: With Aerospace Applications.* Springer Science & Business Media.

Leira, B. J., Sørensen, A. J., and Larsen, C. M. (2004). A reliability-based control algorithm for dynamic positioning of floating vessels. *Structural safety*, 26(1):1–28.

Lie, H., Mo, K., and Kaasen, K. E. (2002). Mimosa user's documentation version 5.6. *Norway, MARINTEK, Software Report*, (516413-00441322).

Lin, X., Xie, Y., Bian, X., and Zhao, D. (2013). Dynamic positioning controller based on unified model in extreme seas. *Journal of Computational Information Systems*, 9(20):8089–8097.

Mavrakos, S., Papazoglou, V., Triantafyllou, M., and Hatjigeorgiou, J. (1996). Deep water mooring dynamics. *Marine structures*, 9(2):181–209.

May, J. (2003). Improving engine utilization on dp drilling vessels. In *Dynamic Positioning Conference*.

Metrikin, I. (2014). A software framework for simulating stationkeeping of a vessel in discontinuous ice.

Morandi, A. and Shi, X. (2011). Practical methods for assessing the reliability of mooring systems. In *ASME 2011 30th International Conference on Ocean, Offshore and Arctic Engineering*, pages 893–902. American Society of Mechanical Engineers.

Nguyen, D. H., Nguyen, D. T., Quek, S. T., and Sørensen, A. J. (2010). Control of marine riser end angles by position mooring. *Control Engineering Practice*, 18(9):1013–1021.

Nguyen, D. H., Nguyen, D. T., Quek, S. T., and Sørensen, A. J. (2011). Position-moored drilling vessel in level ice by control of riser end angles. *Cold Regions Science and Technology*, 66(2):65–74.

Nguyen, D. T. and Sorensen, A. J. (2009). Setpoint chasing for thruster-assisted position mooring. *Oceanic Engineering, IEEE Journal of*, 34(4):548–558.

Nguyen, D. T. and Sørensen, A. J. (2009). Switching control for thruster-assisted position mooring. *Control Engineering Practice*, 17(9):985–994.

Nguyen, T. D., Sørensen, A. J., and Quek, S. T. (2007). Design of hybrid controller for dynamic positioning from calm to extreme sea conditions. *Automatica*, 43(5):768–785.

Niedzwecki, J. and Liagre, P.-Y. (2003). System identification of distributed-parameter marine riser models. *Ocean Engineering*, 30(11):1387–1415.

Nord (2011). Loss of anchors and chain.

Oxford Advanced Learner's Dictionary (n.d.). Viewed 28.04.2015.

Palmer, A. and Dempsey, J. (2009). Model tests in ice. In *Proceedings of the International Conference on Port and Ocean Engineering Under Arctic Conditions*, number POAC09-40.

Perez, T., Smogeli, Ø. N., Fossen, T. I., and Sørensen, A. J. (2006). An overview of the marine systems simulator (mss): A simulink® toolbox for marine control systems. *Modeling, identification and Control*, 27(4):259–275.

Price, J. C. et al. (2003). New technical issues in materials engineering and fabrication for deepwater hydrocarbon development. In *The Thirteenth International Offshore and Polar Engineering Conference*. International Society of Offshore and Polar Engineers.

Price, W. G. and Bishop, R. E. D. (1974). *Probabilistic theory of ship dynamics*. Halsted Press.

Radan, D. (2008). Integrated control of marine electrical power systems.

Reeves, R. R., Ewins, P. J., Agbayani, S., Heide-Jørgensen, M. P., Kovacs, K. M., Lydersen, C., Suydam, R., Elliott, W., Polet, G., van Dijk, Y., et al. (2014). Distribution of endemic c-etaceans in relation to hydrocarbon development and commercial shipping in a warming arctic. *Marine Policy*, 44:375–389.

Ren, Z., Skjetne, R., and Hassani, V. (2015a). Supervisory control of line breakage for thruster-assisted position mooring system. In *10th IFAC Conference on Manoeuvring and Control of Marine Craft (MCMC2015)*. Copenhagen Denmark.

Ren, Z., Skjetne, R., and Kåre, K. Ø. (2015b). A tension-based position estimation approach for moored marine vessels. In *10th IFAC Conference on Manoeuvring and Control of Marine Craft (MCMC2015)*. Copenhagen Denmark.

Rho, J.-B., Korobkin, A. A., Jung, J.-J., Shin, H.-S., and Lee, W.-S. (2007). Coupled analysis of deepwater floating system including viv in time domain. In *ASME 2007 26th International Conference on Offshore Mechanics and Arctic Engineering*, pages 639–649. American Society of Mechanical Engineers.

Rustad, A. M. (2007). *Modeling and control of top tensioned risers*. Fakultet for ingeniørvitenskap og teknologi.

Rustad, A. M., Larsen, C. M., and Sørensen, A. J. (2008). Fem modelling and automatic control for collision prevention of top tensioned risers. *Marine Structures*, 21(1):80–112.

Shang, Y., Rumi, W., Zhang, Y., and Fromherz, M. (2004). Localization from connectivity in sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 15(11):961–974.

Simulator, M. M. S. (2010). Viewed 30.10.2014.

Skjetne, R., Fossen, T. I., and Kokotović, P. V. (2005). Adaptive maneuvering, with experiments, for a model ship in a marine control laboratory. *Automatica*, 41(2):289–298.

Skjetne, R., Jorgensen, U., and Teel, A. R. (2011). Line-of-sight path-following along regularly parametrized curves solved as a generic maneuvering problem. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 2467–2474. IEEE.

Sorensen, A., Strand, J. P., and Fossen, T. I. (1999). Thruster assisted position mooring system for turret-anchored fpsos. In *Control Applications, 1999. Proceedings of the 1999 IEEE International Conference on*, volume 2, pages 1110–1117. IEEE.

Sørensen, A. J. (2012). *Marine Control Systems Propulsion and Motion Control of Ships and Ocean Structures Lecture Notes*. Citeseer.

Stoican, F. and Olaru, S. (2013). *Set-theoretic Fault-tolerant Control in Multisensor Systems*. John Wiley & Sons.

Strand, J. P. and Fossen, T. I. (1999). Nonlinear passive observer design for ships with adaptive wave filtering. In *New Directions in nonlinear observer design*, pages 113–134. Springer.

Tannuri, E. A., Donha, D., and Pesce, C. (2001). Dynamic positioning of a turret moored fpso using sliding mode control. *International Journal of Robust and Nonlinear Control*, 11(13):1239–1256.

Trim, A., Braaten, H., Lie, H., and Tognarelli, M. (2005). Experimental investigation of vortex-induced vibration of long marine risers. *Journal of fluids and structures*, 21(3):335–361.

Tsugawa, T., Saito, A., Otsuka, Y., Nishioka, M., Maruyama, T., Kato, H., Nagatsuma, T., and Murata, K. (2011). Ionospheric disturbances detected by gps total electron content observation after the 2011 off the pacific coast of tohoku earthquake. *Earth, planets and space*, 63(7):875–879.

Uttal, T., Curry, J. A., Mcphee, M. G., Perovich, D. K., Moritz, R. E., Maslanik, J. A., Guest, P. S., Stern, H. L., Moore, J. A., Turenne, R., et al. (2002). Surface heat budget of the arctic ocean. *Bulletin of the American Meteorological Society*, 83(2):255–275.

Van Der Nat, C., Heideman, R., et al. (2012). Developing a turret mooring system for arctic fpso units. In *OTC Arctic Technology Conference*. Offshore Technology Conference.

Wan, E. A. and Van Der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. IEEE.

Wang, Y., Zou, C., Ding, F., Dou, X., Ma, Y., and Liu, Y. (2014). Structural reliability based dynamic positioning of turret-moored fpsos in extreme seas. *Mathematical Problems in Engineering*, 2014.

Zekavat, R. and Buehrer, R. M. (2011). Collaborative position location. In Buehrer, R. M. and Jia, T., editors, *Handbook of position location: Theory, practice and advances*, chapter 12, pages 755–810. John Wiley & Sons.

Zhao, B., Blanke, M., and Skjetne, R. (2012). Particle filter based fault-tolerant rov navigation using hydro-acoustic position and doppler velocity measurements. In *9th IFAC Conference on Manoeuvring and Control of Marine Craft*, pages 280–286.

Acronyms

CapEx Capital expenses.

COT Center of turret.

CPM Control plant model.

DGPS Differential global positioning system.

DOF Degrees of freedom.

DP Dynamic positioning.

KF Kalman filter.

FEM Finite element method.

FPS Floating production system.

FPSO Floating production, storage and offloading.

GNSS Global navigation satellite system.

GPS Global positioning system.

HPR Hydroacoustic position reference.

HSVA Hamburgische Schiffbau-Versuchsanstalt.

JONSWAP Joint North Sea wave project.

LOS Line-of-sight.

LQG Linear-quadratic-Gaussian.

MCLab Marine Cybernetics Laboratory.

MDD Mooring Design and Dynamics.

MODU Mobile offshore drilling units.

MSS Marine System Simulator.

NED North-East-Down.

NPO Nonlinear passive observer.

OpEx Operational expenses.

PDE Partial differential equations.

PID Proportional-integral-derivative.

PMR Position mooring reliability.

posref Position reference.

PPM Process plant model.

SDP Semi-definite programming.

SOCP Second-order cone programming.

SID Sudden ionospheric disturbances.

TAPM Thruster-assisted position mooring.

TTR Top tensioned riser.

VIV Vortex-induced vibration.

VOC Vessel operation condition.

WF Wave-frequency.

WOPC Weather optimal position control.

Symbols

A_m Cross-section area of the line.

D_z^t Damping of the turret.

E_m Weight in water per unit length.

H_i Horizontal force component.

I_z^t Mass inertia of moment of the turret.

L_m Unstretched line length of the mooring lines.

T_i Tension at the end of the i^{th} mooring line.

$\bar{x}_i$ Horizontal displacements in X direction of the i^{th} mooring line between TP and Anchor i.

$\bar{y}_i$ Horizontal displacements in Y direction of the i^{th} mooring line between TP and Anchor i.

$\boldsymbol{A_{pw}}$ WF-induced motion as a mass-damper-spring system.

$\boldsymbol{C}_A$ Coriolis and centripetal matrices of the added mass.

$\boldsymbol{C}_{RB}$ Coriolis and centripetal matrices of the rigid body.

$\boldsymbol{D}$ Hydrodynamic damping matrix of the vessel.

$\boldsymbol{D}_L$ Linearized hydrodynamic damping matrix of the vessel.

$\boldsymbol{D}_{mo}$ Linearized damping term of mooring system.

$\boldsymbol{G}$ Hydrostatic restoring matrix.

$\boldsymbol{G}_{mo}$ Linearized stiffness of mooring system.

$\boldsymbol{K}$ Diagonal force coefficient matrix.

$\boldsymbol{K}_d$ Diagonal Non-negative D controller gain matrices.

$\boldsymbol{K}_i$ Diagonal Non-negative I controller gain matrices.

$\boldsymbol{K}_p$ Diagonal Non-negative P controller gain matrices.

$\boldsymbol{M}$ Mass matrix.

$\boldsymbol{T}(\boldsymbol{\alpha})$ Thrust configuration matrix.

$\boldsymbol{T}(\boldsymbol{\beta})$ Mooring line configuration matrix.

$\boldsymbol{\Gamma}$ Diagonal matrix of damping ratios.

$\boldsymbol{\Omega}$ Diagonal matrix containing the dominating wave response frequencies.

$\boldsymbol{\eta}$ Position vector of vessel.

$\boldsymbol{\eta}_d$ Desired position.

$\boldsymbol{\eta}_r$ Reference signal output from the reference system.

$\boldsymbol{\nu}$ Velocity vector of vessel.

$\boldsymbol{\nu}_c$ Current velocity vector.

$\boldsymbol{\nu}_r$ Relative velocity vector.

$\boldsymbol{\tau}_c$ Thruster input.

$\boldsymbol{\tau}_m$ Mooring system forces and moment vector.

$\boldsymbol{\tau}_{thr}$ Thruster-induced force vector.

$\boldsymbol{\tau}_{wave1}$ First-order wave force vector.

$\boldsymbol{\tau}_{wind}$ Wind forces vector.

$\boldsymbol{b}$ Slow varying disturbances.

$\boldsymbol{d}_{mo}$ Damping term of mooring system.

$\boldsymbol{g}_{mo}$ Stiffness of mooring system.

$\boldsymbol{h}_m$ Horizontal mooring force vector.

$\boldsymbol{u}$ Vector of control inputs.

$\boldsymbol{x}_{COT}$ Position of the COT.

$\boldsymbol{x}_{f,i}$ Position of the i^{th} fairleads.

γ_i Angle of the i^{th} fairlead compare to the reference angle.

$\hat{\boldsymbol{\eta}}_w$ State of the WF model.

ω_m Weight in water per unit length.

ϕ_i Angle between the line tension and its vertical component.

ψ_t Angle of the turret comparing with the reference.

$\tilde{\boldsymbol{y}}$ Output estimation error.

$\tilde{\psi}_t$ Relative angle of the turret.

r_t Radius of the circles where the fairleads locate.

s Path parameter along the cable.

$x_i(s)$ Horizontal positions of each segment centered at length s along the i^{th} cable.

$z_i(s)$ Vertical positions of each segment centered at length s along the i^{th} cable.

Appendix A

MSS TAPM User's Guide

A.1 Introduction

MSS TAPM is a expansion pack of MSS GNC and MSS Hydro toolbox. MSS TAPM is a Simulink lib for modelling thruster-assisted position mooring (TAPM or POSMOOR) system, especially the turret-based mooring system. It is built by Zhengru Ren during his master thesis. In this lib, there are a group of different modules and examples. Two mooring models are included in this lib, that is, the catenary equations and the FEM model.

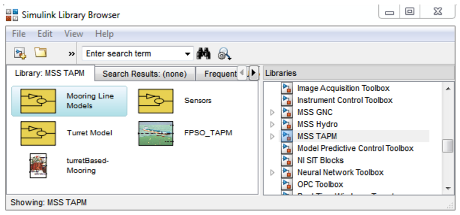
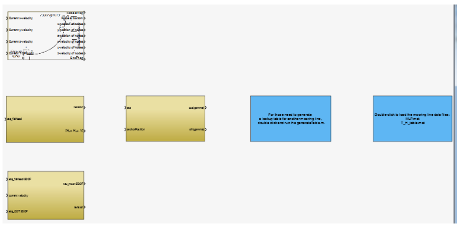


Figure A.1: The MSS TAPM lib.



For those need to generate a lookup table for another mooring line, double click and run the generateTable.m.

Double-click to load the mooring line data files: MLP.mat T_H_table.mat

Figure A.2: Mooring line sublib.

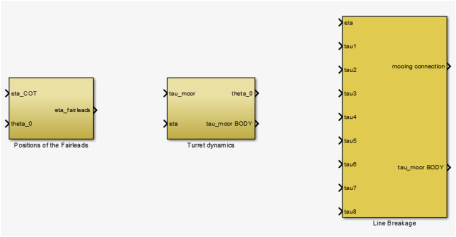


Figure A.3: Turret sublib.

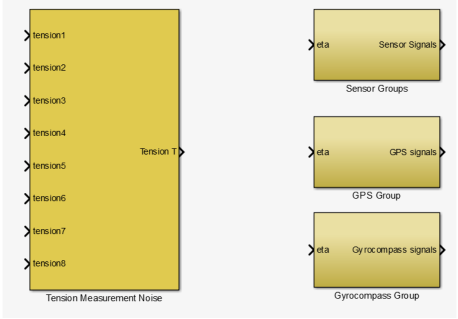


Figure A.4: Sensor sublib.

A.2 Licensing

MSS TAPM is free for use in academic in NTNU. There is no warranty for this free toolbox.

A.3 Installation

MSS TAPM is supported on 32-bit and 64-bit versions of Linux, Mac OSX, and Windows. For 32-bit platforms, MATLAB version 7.5 (R2007b) or later is required; for 64-bit platforms, MATLAB version 7.8 (R2009a) or later is required. MSS GNC and MSS Hydro are needed to be installed to provide some fundamental function to support MSS TAPM.

To install it:

Step 1. Down MSS TAPM toolbox and unzip it to any path.

Step 2. Click the "Set Path" in Matlab.

Step 3. Chick "Add with subfolders" and include the root folder.

Step 4. Restart MATLAB and you can find an "MSS TAPM" Lib in the Simulink library; see Figure A.1.

A.4 Development guidance

All the blocks are developed according to the MC-SIM Simulink guidance. First of all, different colors are applied to sort all blocks into three main categories, that is,

- Green: Sources (inputs ports, constants, etc.),
- Red: Sinks (output ports, terminators, to workspace blocks, etc.),
- Yellow: All remaining blocks.

Furthermore, describing names are given to all the subsystems, all the input/output ports, and all modules. Finally, user friendly interface (mask) are built with simplified description. *Enable* and *Visible* properties are called back when checking the check boxes. This improvement can help the users reduce the cost of learning, and it can make the interface more clear and less confusing.

A.5 A quick start

A.5.1 Mooring line model

Figure A.5.1 shows the *mooring line model* which implements a mooring line. There are two models, the catenary equations and the FEM model, are available to provide the mooring line forces in three DOFs. In this case we assume the turret can rotate around z axix with a center at the center of the turret (COT). The mooring lines are connected to the fairleads which rotates with the turret.

The model is also applicable to tune to the TAPM without a turret or fairleads. To reduce it to a condition that all the mooring lines are connected at the COT, please replace the *moment generate* module with the *COT moment generate* module.

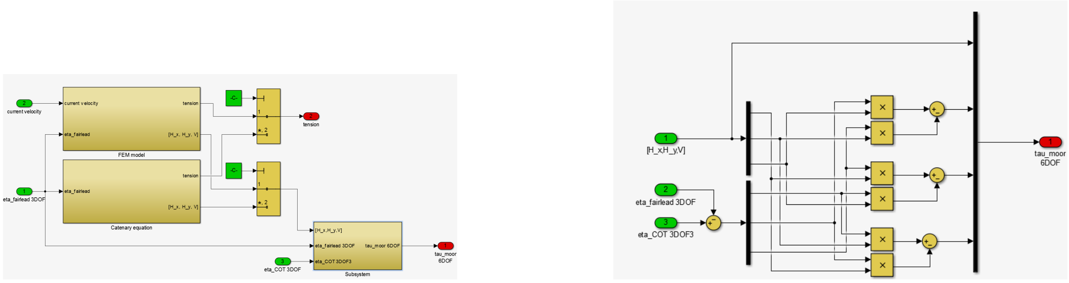


Figure A.5: The structure of the mooring line model block.

A.5.2 Catenary equations

Catenary equations and the corresponding algorithms are introduced in Section 3.5.2. MATLAB codes in Listing A.1 are applied to generate the 2D lookup tables which contain the mooring tensions, the horizontal and vertical components of the mooring tensions.

Listing A.1: The MATLAB code for generating a 2D lookup table.

```
%% Lookup Table GenerationDiscription: This matlab script is used to ...
    generate a 2D lookup table for a single mooring line for based on ...
    the catenary equations, by calling the funcitons written by Prof. ...
    Ole Morten Aamo. The lookup table can be used to provide quasi ...
    static modelling in Simulink.

```

```
% Additionally, this script also generates a structure, MLP (abbr. of ...
    mooring line properties), which contains the necessary ...
    information for the Simulink turrentBasedMooring model and ...
    TAPM_FPSO model.

% Author:    Zhengru Ren
% Date:      07/11/2014

% References:
%   M.S. Triantafyllou: Cable Mechanics with Marine Applications.
%        Department of Ocean Engineering,
%        Massachusetts Institute of Technology, 1990.
%
%   O.M. Aamo: Adding mooring systems to the ABB IVS.
%        ABB 1999.
clear
clc

disp('WARNING! The lookuptable is so big that it will cost the ...
    computer several hours to calculate.')
disp('If needed, press "Ctrl+C" to terminate it.');

MLP.rok     =   5500;       % cable density (kg/m^3)
MLP.rov     =   1025;       % water density (kg/m^3)
MLP.L       =   2250;       % Length of the mooring line.
MLP.E       =   2.06e11;    % Elastic modulus.
MLP.A0      =   0.005;      % Cross sectional area of the cable.

MLP.emfact  =   1;          % Factor for stiffness calculation. ...
    1=normal, 2=chain.
MLP.my      =   1;          % Friction coefficient for bottom ...
    interaction.
MLP.m       =   0;          % Weight in water for lump masses.
MLP.D       =   1000;       % Water depth.


MLP.ne      =   20;          % Vector with one value for each segment.
MLP.me      =   0.05;        % Max strain.
MLP.s       =   [0;[MLP.L/MLP.ne:MLP.L/MLP.ne:MLP.L]'];         % ...
```

```
   Vector of s-values for which to return x and z values.
MLP.errtol  =   1.0E-6;     % Error tolerance.

g           =   9.81;
MLP.omega   =   g*(MLP.rok-MLP.rov).*MLP.A0;     % Weight per unit ...
   length in water.

save('MLP','MLP');          % Save MLP.

%%
scope   =   80;
step    =   0.1;

j=1                                 % Initialize j.
for Z = -4:0.2:4                    % Vertical motion range of the ...
   top point (TP) in the lookup table.
    i=1;                            % Inilialize i.
    for Xi=(-scope-570):step:(scope)    % Horizontal motion range of ...
       TP in the lookup table.
        X=2050+Xi;
        [V,H,TD,V0,H0,x,z,et]=elastic1(MLP.omega, MLP.L, MLP.E, ...
           MLP.A0, MLP.emfact, MLP.my, MLP.m, MLP.D+Z, X, MLP.me, ...
           MLP.s, MLP.ne+1, MLP.errtol);
        T_H_table.V(i,j)=V;         % Vertical force at TP.
        T_H_table.H(i,j)=H;         % Horizontal force at TP.
        T_H_table.TD(i,j)=TD;       % Distance from anchor point to ...
           touchdown point along seafloor.
        T_H_table.V0(i,j)=V0;       % Vertical force at anchor.
        T_H_table.H0(i,j)=H0;       % Horizontal force at anchor.
        T_H_table.X(i)=X;           % Horizontal distance between the ...
           anchor and TP.
%         T_H_table.x{i,j}=x;
%         T_H_table.z{i,j}=z;
        i=i+1;
    end
    T_H_table.Z(j)=Z;
    j=j+1
end

```

```
for j=1:length(T_H_table.Z)
    for i=1:length(T_H_table.X)
        T_H_table.T(i,j)=sqrt(T_H_table.V(i,j)^2+T_H_table.H(i,j)^2); ...
             %  Tension force at TP.
    end
end

save('T_H_table.mat','T_H_table');  % Save T_H_table.
```

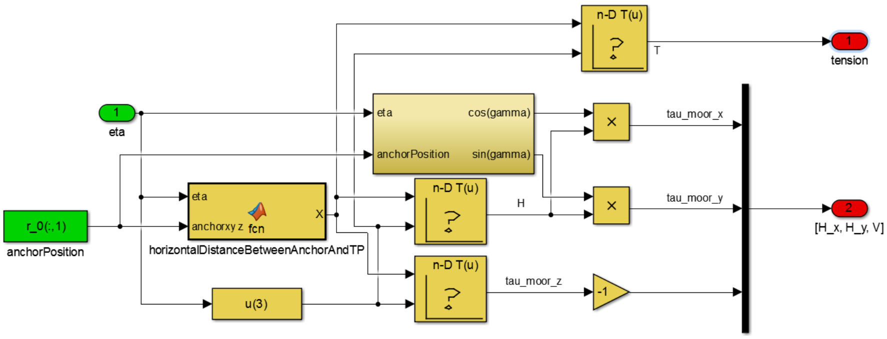


Figure A.6: The Simulink structure of the catenary model.

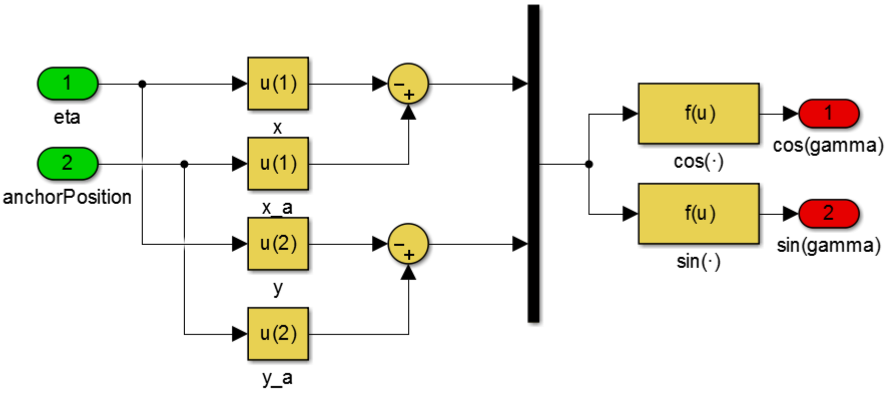


Figure A.7: The Simulink structure of transformer in the catenary model.

A.5.3 FEM model

The FEM model is implemented in a S-function. The S-function is programmed by Professor Ole Morten Aamo.

- *Initial top point position*: is applied to determine the initial position of the top end. This top end can be a fairlead, or any other points.

- *Bottom point position*: is applied to set the bottom position of the mooring line. For the *Initial top point position* and *Bottom point position*, the positive z-axis points downwards.

- *Cable diameter*: is the diameter of the mooring line, unit [m].

- *Young's modulus*: is determined by the material of the chain, unit [Pa].

- *Factor of stiffness*

- *unstreched length of cable*: is the length of the chain without stress, unit [m].

- *Normal/Tangential drag coefficient*: is the inline/crossline damping force coefficients for the mooring line. These values are abstained empirically.

- *Added mass coefficient*: is the added mass coefficients for the mooring line. This value is abstained empirically.

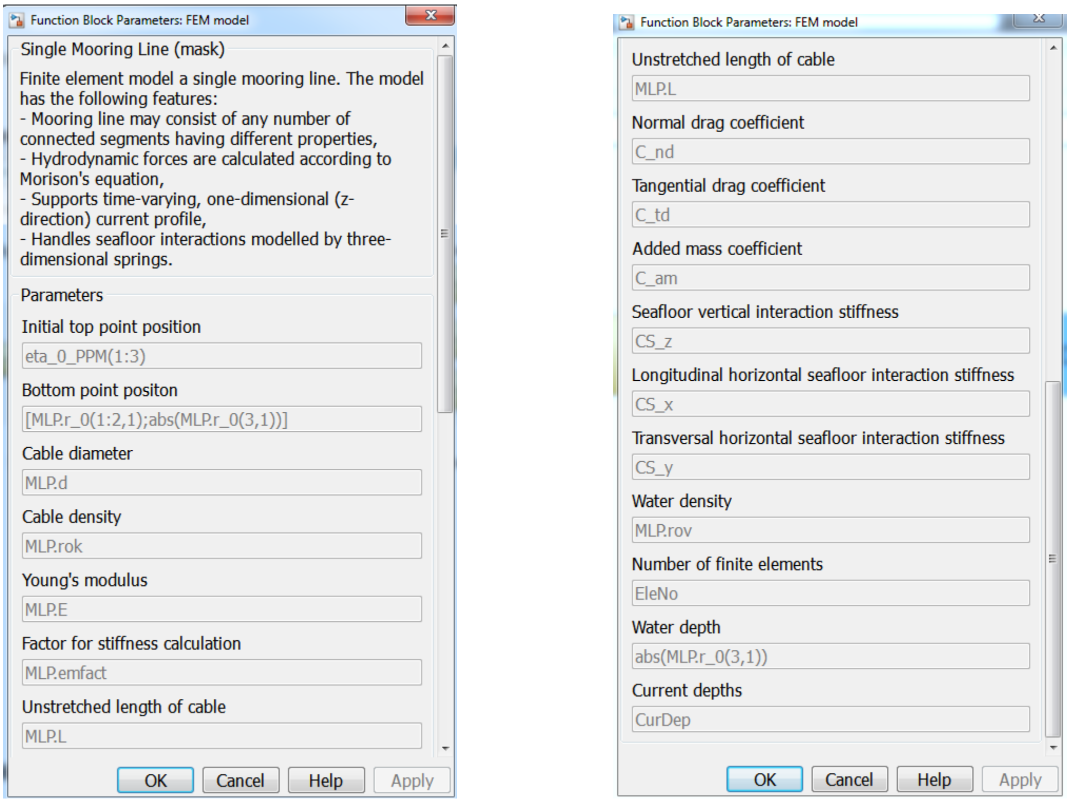
- *Seafloor vertical/longitudinal/horizontal interaction stiffness*:

- *water density*: is the density of the fluid around the mooring line, unit $[kg/m^3]$. Normally sea water, 1025 kg/m^3

- *Number of finite element*: is the number of the finite elements to be simulate in the FEM model.

- *Water depth*: is depth where the mooring line interacts with the fluids. This is a positive value.

- *Current depth*: determines the current profile.



Function Block Parameters: FEM model

Single Mooring Line (mask)

Finite element model a single mooring line. The model has the following features:
- Mooring line may consist of any number of connected segments having different properties,
- Hydrodynamic forces are calculated according to Morison's equation,
- Supports time-varying, one-dimensional (z-direction) current profile,
- Handles seafloor interactions modelled by three-dimensional springs.

Parameters

Initial top point position: eta_0_PPM(1:3)

Bottom point positon: [MLP.r_0(1:2,1);abs(MLP.r_0(3,1))]

Cable diameter: MLP.d

Cable density: MLP.rok

Young's modulus: MLP.E

Factor for stiffness calculation: MLP.emfact

Unstretched length of cable: MLP.L

Function Block Parameters: FEM model

Unstretched length of cable: MLP.L

Normal drag coefficient: C_nd

Tangential drag coefficient: C_td

Added mass coefficient: C_am

Seafloor vertical interaction stiffness: CS_z

Longitudinal horizontal seafloor interaction stiffness: CS_x

Transversal horizontal seafloor interaction stiffness: CS_y

Water density: MLP.rov

Number of finite elements: EleNo

Water depth: abs(MLP.r_0(3,1))

Current depths: CurDep

Figure A.8: The interface of the FEM model block.

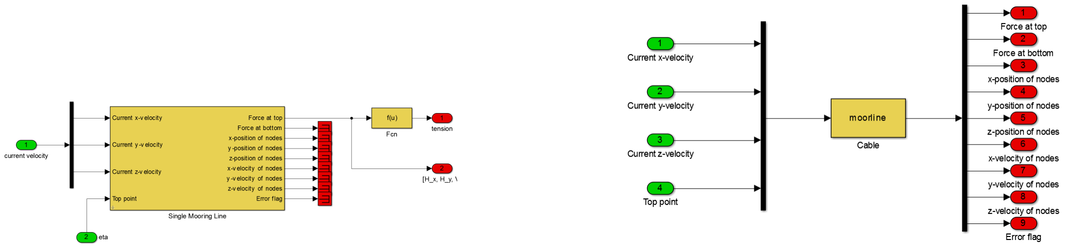


Figure A.9: The structure of the FEM model block.

A.6 Model of the turret

This block implements the mooring forces with 8 mooring lines whose upper terminal points are evenly distributed in a circle. in this mooring system. This is a simplified model. All the mooring lines are assumed to connect with the turret at the center of the turret (COT). The COT locates at the center of the ship. Therefore, there are no moments from mooring lines.

- *Load mooring line parameter*: will load the structure in the root folder, which can assign values to the variables and initialize the simulation.

- *Mooring model selection*: popup can help the user select from the catenate equation and the FEM model.

- *Distance betwen the turret and the fairlead*: influence the distribution of the fairleads.

- *Inverse moment of the interial of the turret*: is a parameter influencing the turret rotation, I_z^{t-1}. Directly input the inverse value to enhance the computational speed.

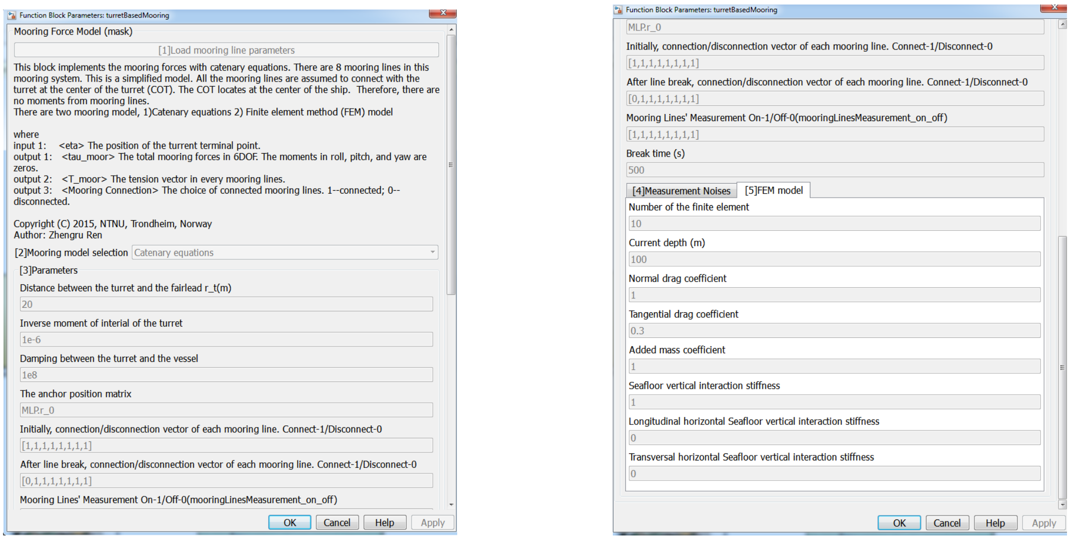
- *Damping between the turret and the vessel* contributes a restoring yaw moment to the vessel. The damping is mainly due to friction. Here, we only consider line con-friction for the sake of simplification.

- *The anchor position matrix*: is a matrix contains all the 3D positions of the anchors. The matrix is $3 \times N$, where each column is the vector of an anchor.

- *Initial connection vector*: is a vector where each element represents the output percentage of the tension to the vessel. The number is same as the column index in *Initial/After line break, connection vector*: contains the information of the connection of all mooring lines. Zero in the i^{th} element means the i^{th} mooring force do not input to the vessel, that is, mooring line disconnection, or breakage. One represent the mooring force interacting on the vessel. One also can give a value between 0 and 1 to simulate the breakage happens at the the middle or near the bottom end. However, since most line breakage happens near the top end, 0 is an suggested value.

- *Mooring lines measurement*: contains the information of the measurement of all mooring line. When the tension cells fell, zero is applied here.

- *Break time*: is the time instant when the mooring line breaks, unit [s]. If there is NO mooring line failure in the simulation, 999999 can be used here.



Mooring Force Model (mask)

[1]Load mooring line parameters

This block implements the mooring forces with catenary equations. There are 8 mooring lines in this mooring system. This is a simplified model. All the mooring lines are assumed to connect with the turret at the center of the turret (COT). The COT locates at the center of the ship. Therefore, there are no moments from mooring lines.
There are two mooring model, 1)Catenary equations 2) Finite element method (FEM) model

where
input 1: <eta> The position of the turrent terminal point.
output 1: <tau_moor> The total mooring forces in 6DOF. The moments in roll, pitch, and yaw are zeros.
output 2: <T_moor> The tension vector in every mooring lines.
output 3: <Mooring Connection> The choice of connected mooring lines. 1--connected; 0--disconnected.

Copyright (C) 2015, NTNU, Trondheim, Norway
Author: Zhengru Ren

[2]Mooring model selection: Catenary equations

[3]Parameters

Distance between the turret and the fairlead r_t(m): 20

Inverse moment of interial of the turret: 1e-6

Damping between the turret and the vessel: 1e8

The anchor position matrix: MLP.r_0

Initially, connection/disconnection vector of each mooring line. Connect-1/Disconnect-0: [1,1,1,1,1,1,1,1]

After line break, connection/disconnection vector of each mooring line. Connect-1/Disconnect-0: [0,1,1,1,1,1,1,1]

Mooring Lines' Measurement On-1/Off-0(mooringLinesMeasurement_on_off)

MLP.r_0

Initially, connection/disconnection vector of each mooring line. Connect-1/Disconnect-0: [1,1,1,1,1,1,1,1]

After line break, connection/disconnection vector of each mooring line. Connect-1/Disconnect-0: [0,1,1,1,1,1,1,1]

Mooring Lines' Measurement On-1/Off-0(mooringLinesMeasurement_on_off): [1,1,1,1,1,1,1,1]

Break time (s): 500

[4]Measurement Noises | [5]FEM model

Number of the finite element: 10

Current depth (m): 100

Normal drag coefficient: 1

Tangential drag coefficient: 0.3

Added mass coefficient: 1

Seafloor vertical interaction stiffness: 1

Longitudinal horizontal Seafloor vertical interaction stiffness: 0

Transversal horizontal Seafloor vertical interaction stiffness: 0

Figure A.10: The interface of the turret model block.

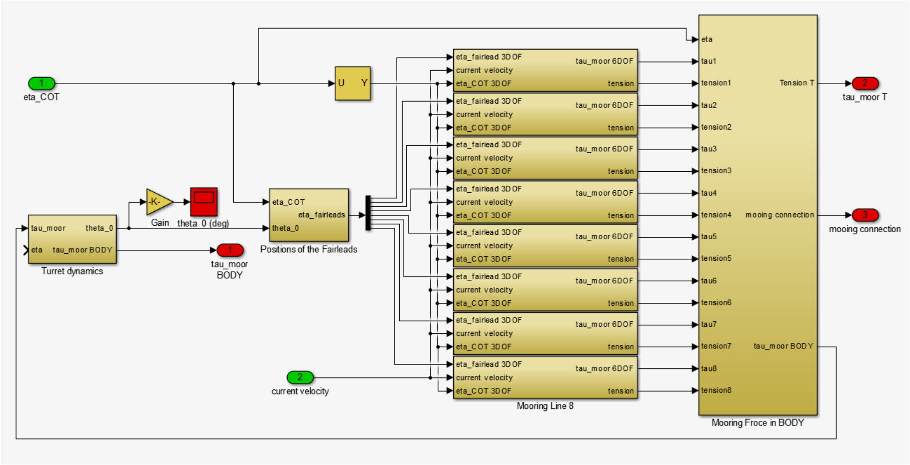


Figure A.11: The Simulink structure of a turret.

A.6.1 Influence of the fairleads

The code to control the position of the fairleads is shown in Listing A.2.

Listing A.2: The MATLAB code for turret dynamics.

```
function eta_fairlead = fairleadPos(eta_COT,theta_fiarlead,r_fairlead)
% This code is used to generate the vector which contains the ...
    positions of
```

```
% the fairleads. All the fairleads are distributed evenly in a circle ...
    which
% the center is the COT and the radius is r. The vertical motions are
% neglected.

% eta_COT: is the position of the center of curret (COT).
% theta_fairlead: is a angle we choose as a standard angle which
%                    represents the rotation of the turret.
% r_fairlead: the radius of the circle where the fairleads locate.

% Author: Ren, Zhengru
% Date: 10/05/2015

n_fairlead=8; % total number of the fairleads.

eta_fairlead=zeros(3*n_fairlead,1);

for i=1:n_fairlead
    eta_fairlead(3*i-2:3*i,1) = ...
        [eta_COT(1)+r_fairlead*cos(theta_fiarlead + ...
        (i-1)*2*pi/n_fairlead); eta_COT(2)+ r_fairlead* ...
        sin(theta_fiarlead + (i-1)*2*pi/n_fairlead); 0];
end
```

A.6.2 Turret dynamics

The dynamics of the turret rotation is shown in Figure A.12.

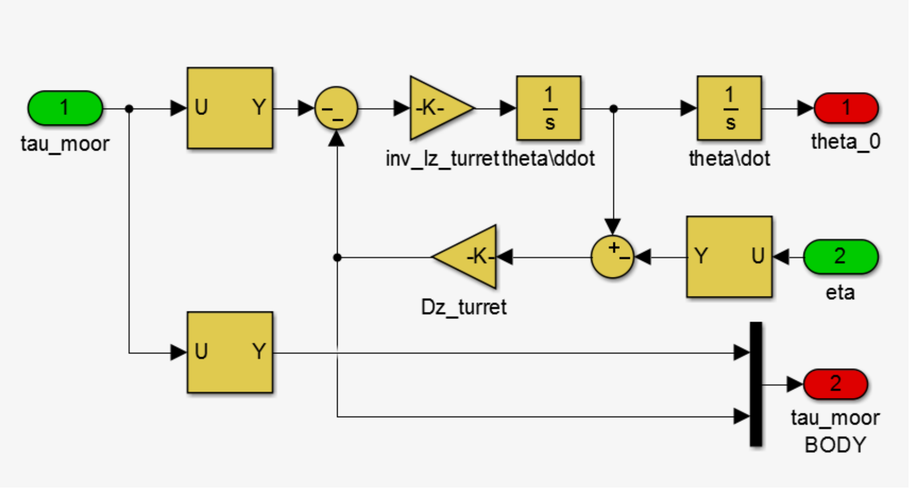
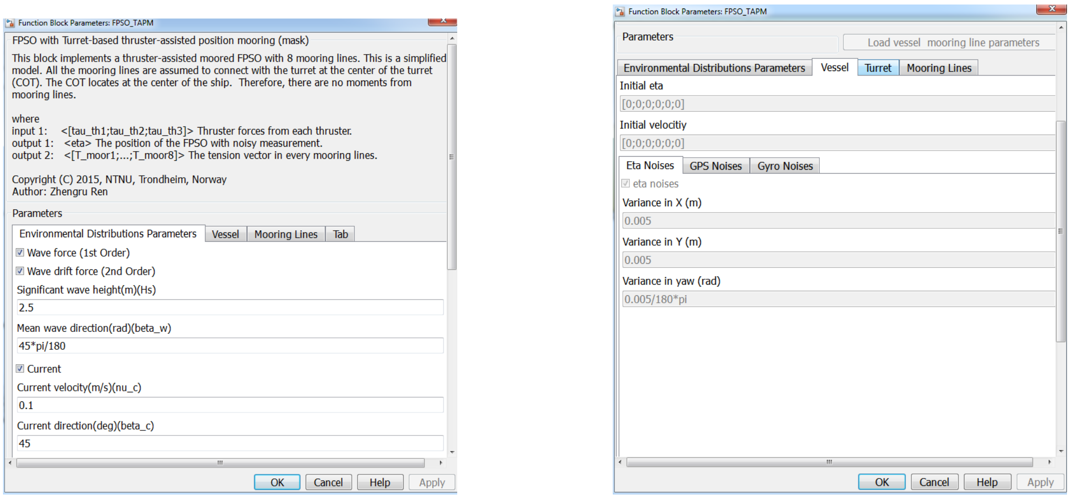


Figure A.12: The Simulink structure of a turret.

A.7 Model of the TAPM

This block implements a thruster-assisted moored FPSO with 8 mooring lines. The property interface is shown in Figure A.13. The process plant model is developed in 6 DOFs with motion RAOs. The environmental loads are wind, current, and waves. There are three GPSs, three Gyrocompasses, and three IMUs. All the sensor signals are independent. The controllable parameters includes the biases and noise variances of all the sensors. All the sensor signals are labelled and output in a bus. The Simulink structure of the GPS signal, GPS group, gyrocompass, and gyrocompass group are represented in Figure A.15, respectively.


Function Block Parameters: FPSO_TAPM
Parameters
Environmental Distributions Parameters
Vessel
Mooring Lines
Tab
OK
Cancel
Help
Apply

FPSO with Turret-based thruster-assisted position mooring (mask)

This block implements a thruster-assisted moored FPSO with 8 mooring lines. This is a simplified model. All the mooring lines are assumed to connect with the turret at the center of the turret (COT). The COT locates at the center of the ship. Therefore, there are no moments from mooring lines.

where
input 1: <[tau_th1;tau_th2;tau_th3]> Thruster forces from each thruster.
output 1: <eta> The position of the FPSO with noisy measurement.
output 2: <[T_moor1;...;T_moor8]> The tension vector in every mooring lines.

Copyright (C) 2015, NTNU, Trondheim, Norway
Author: Zhengru Ren

☑ Wave force (1st Order)
☑ Wave drift force (2nd Order)
Significant wave height(m)(Hs): 2.5
Mean wave direction(rad)(beta_w): 45*pi/180
☑ Current
Current velocity(m/s)(nu_c): 0.1
Current direction(deg)(beta_c): 45

Function Block Parameters: FPSO_TAPM
Parameters
Load vessel mooring line parameters
Environmental Distributions Parameters
Vessel
Turret
Mooring Lines
Eta Noises
GPS Noises
Gyro Noises
OK
Cancel
Help
Apply

Initial eta: [0;0;0;0;0;0]
Initial velocitiy: [0;0;0;0;0;0]
☑ eta noises
Variance in X (m): 0.005
Variance in Y (m): 0.005
Variance in yaw (rad): 0.005/180*pi

Figure A.13: The Interface of a TAPM.

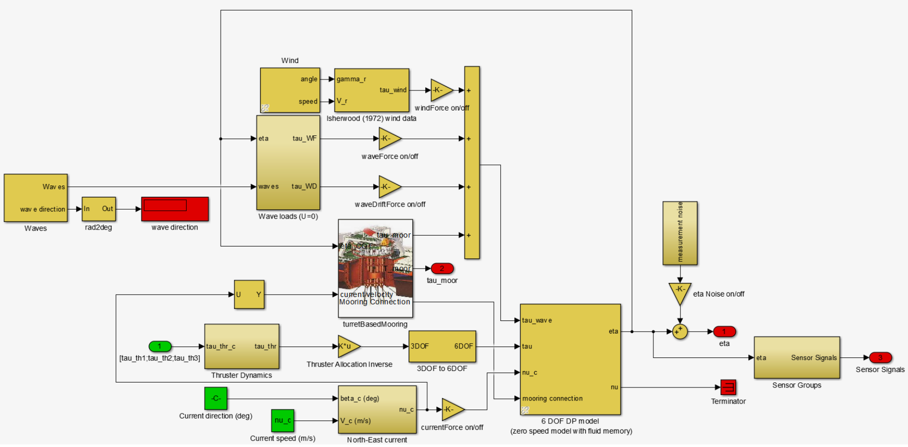


Figure A.14: The Simulink structure of a turret.

Environmental Distribution Parameter tab includes the set of all the environmental loads, including the first-order wave load, second-order wave load, current load, and wind load.

- *Wave force (1st/2nd Order)*: enable the first/second order wave loads. Either of them is checked, the *Significant wave height* and the *Mean wave direction* edits are enabled.
- *Significant wave height*: set the significant wave height, unit [m].
- *Mean wave direction*: set the mean wave direction, unit [rad].
- *Current*: enable the current loads. When it is checked, the *Current velocity* and the *Current direction* edits are enabled.
- *Current velocity*: set the significant wave height, unit [m].
- *Current direction*: set the mean wave direction, unit [rad].

Vessel tab includes the initial position and velocity for the vessel in 6 DOFs. A group of three GPSes and a group of Gyrocompasses can be set respectively. The setting opinions include the variances of noise, biases, and the simple times.

Turret tab is same as Section A.6.

Mooring Line tab is same as Section A.5.3.

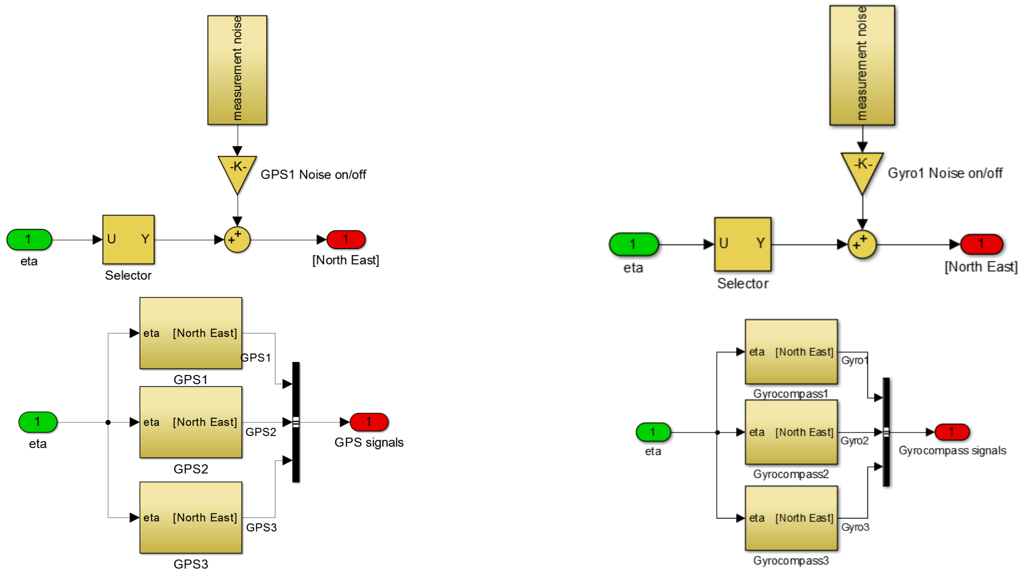


Figure A.15: The Simulink structure of (a) a GPS, (b) a Gyrocompass, (c) the GPS group, and (d) the Gyrocompass group.

A.8 Animation function

An additional function was developed to generate a real-time animation which contain all the nodes of the mooring lines. The function can generate a gif animation with the specific sample frequency and animation speed. The codes are represented in Listing A.3. After running the model in Simulink, the *moorAnimation* function . There are a group of parameters to control the animation.

Listing A.3: The MATLAB code for compiling the FEM mooring models.

```matlab
%% Mooring Line Animation
function MoorAnimation(FigureNo, LineNumber, MoorAni_x, MoorAni_y, ...
    MoorAni_z, FairleadAni_xyz, Tstart, Tend, Fq_GIF, Speed_GIF, ...
    NodeStep, scopeAniX, scopeAniY, scopeAniZ, filename)
% This is a function which is applied to gerenarate an animation of the
% simulated TAPM model. Since the catenary equations model has specific
% forms, we draw the mooring line with 20 nodes.
% FigureNo is the number of the plot.
% LineNumber is the number of the mooring line which will show in the ...
```

```
    animation.
% MoorAni_x is a vector of cells contain all the mooring line ...
    positions in x direction.
% MoorAni_yis a vector of cells contain all the mooring line ...
    positions in y direction.
% MoorAni_z is a vector of cells contain all the mooring line ...
    positions in z direction.
% FairleadAni_xyz is a vector of cells contain all fairlead position ...
    in x, y and z direction.
% Tstart (s) is the time start drawing.
% Tend (s) is a the time end drawing.
% Fq_GIF (Hz) is the frequency of the animation.
% Speed_GIF is the speed of the animation.
% NodeStep is the node interval ploting. When the number of elements ...
    is too
% large, it will be disordered to draw all the nodes.
% scopeAniX is the scope of the x-axis.
% scopeAniY is the scope of the x-axis.
% scopeAniZ is the scope of the x-axis.
% filename is the filename of the save GIF file.

%   E.g.
%       FigureNo=1;
%       LineNumber=1;
%       MoorAni_x={moorLineAni_x1, moorLineAni_x2, moorLineAni_x3, ...
    moorLineAni_x4, moorLineAni_x5, moorLineAni_x6, moorLineAni_x7, ...
    moorLineAni_x8};
%       MoorAni_y={moorLineAni_y1, moorLineAni_y2, moorLineAni_y3, ...
    moorLineAni_y4, moorLineAni_y5, moorLineAni_y6, moorLineAni_y7, ...
    moorLineAni_y8};
%       MoorAni_z={moorLineAni_z1, moorLineAni_z2, moorLineAni_z3, ...
    moorLineAni_z4, moorLineAni_z5, moorLineAni_z6, moorLineAni_z7, ...
    moorLineAni_z8};
%       FairleadAni_xyz={moorLineAni_fairlead1, ...
    moorLineAni_fairlead2, moorLineAni_fairlead3, ...
    moorLineAni_fairlead4, moorLineAni_fairlead5, ...
    moorLineAni_fairlead6, moorLineAni_fairlead7, moorLineAni_fairlead8};
%       Tstart=50;
%       Tend=80;
%       Fq_GIF=10;
%       Speed_GIF=5;
%       scopeAniX=[-2000 2000];
%       scopeAniY=[-2000 2000];
%       scopeAniZ=[0 1000];
%       filename='./gif/test.gif';

% Author: Ren, Zhengru
% Date: 10/05/2015

clear x y z A map;
clf(figure(FigureNo))
figure(FigureNo)

k=1;

for timeNow = Tstart:Speed_GIF/Fq_GIF:Tend
    for i=LineNumber
        j=find(MoorAni_x{i}.time≥timeNow,1,'first');
        x(i,:) = [MoorAni_x{i}.signals.values(j-1,:), ...
            FairleadAni_xyz{i}.signals.values(j-1,1)] + ...
            ([MoorAni_x{i}.signals.values(j,:), FairleadAni_xyz{i}. ...
            signals.values(j,1)] - ...
            [MoorAni_x{i}.signals.values(j-1,:), ...
            FairleadAni_xyz{i}.signals.values(j-1,1)] ) .* ...
            (timeNow-MoorAni_x{i}.time(j-1)) / (MoorAni_x{i}.time(j) ...
            - MoorAni_x{i}.time(j-1));
        y(i,:) = [MoorAni_y{i}.signals.values(j-1,:), ...
            FairleadAni_xyz{i}. signals.values(j-1,2)] + ...
            ([MoorAni_y{i}.signals.values(j,:), FairleadAni_xyz{i}. ...
            signals.values(j,2)] - ...
            [MoorAni_y{i}.signals.values(j-1,:), FairleadAni_xyz{i}. ...
            signals.values(j-1,2)] ) .* ...
            (timeNow-MoorAni_x{i}.time(j-1)) / (MoorAni_x{i}.time(j) ...
            - MoorAni_x{i}.time(j-1));
        z(i,:) = ...
            1000-([MoorAni_z{i}.signals.values(j-1,:),FairleadAni_xyz{i}. ...
            signals.values(j-1,3)] + ...
            ([MoorAni_z{i}.signals.values(j,:),FairleadAni_xyz{i}. ...
            signals.values(j,3)] - ...
            [MoorAni_z{i}.signals.values(j-1,:),FairleadAni_xyz{i}. ...
            signals.values(j-1,3)]) .* ...
            (timeNow-MoorAni_x{i}.time(j-1)) / (MoorAni_x{i}.time(j) ...
            - MoorAni_x{i}.time(j-1)));
     end

     if length(LineNumber)≥2
         for i=LineNumber(1:end-1)
             plot3(x(i,:),y(i,:),z(i,:),'-*');
             hold on
         end
         i=LineNumber(end);
         plot3(x(i,:),y(i,:),z(i,:),'-*');
     elseif length(LineNumber)==1
         plot3(x(1,:),y(1,:),z(1,:),'-*');
     end
     xlim(scopeAniX); ylim(scopeAniY); zlim(scopeAniZ);
     title(['Time:',num2str(timeNow),'seconds']);
     drawnow

     frame=getframe(FigureNo);
     im=frame2im(frame);
     [A,map]=rgb2ind(im,256);
     if k == 1;
         imwrite(A,map,filename, 'gif', 'LoopCount', Inf, 'DelayTime', ...
            1/Fq_GIF);
     else
         imwrite(A,map,filename, 'gif', 'WriteMode', 'append', ...
            'DelayTime', 1/Fq_GIF);
     end
     k=k+1;
 end
```

A.9 Run the model in another PC

After moving this model to another computer, the *moorline.c* S-function should be compiled again. The MATLAB codes are shown in Listing A.4.

Listing A.4: The MATLAB code for compiling the FEM mooring models.

```
mex moorline1.c;
mex moorline2.c;
mex moorline3.c;
mex moorline4.c;
mex moorline5.c;
mex moorline6.c;
mex moorline7.c;
mex moorline8.c;
```

Please run *mex -setup* and ensure your computer has installed the Microsoft Visual C++ environment for C language compilation.

Figure A.16 is a flow chart showing the mechanism of a S-function.

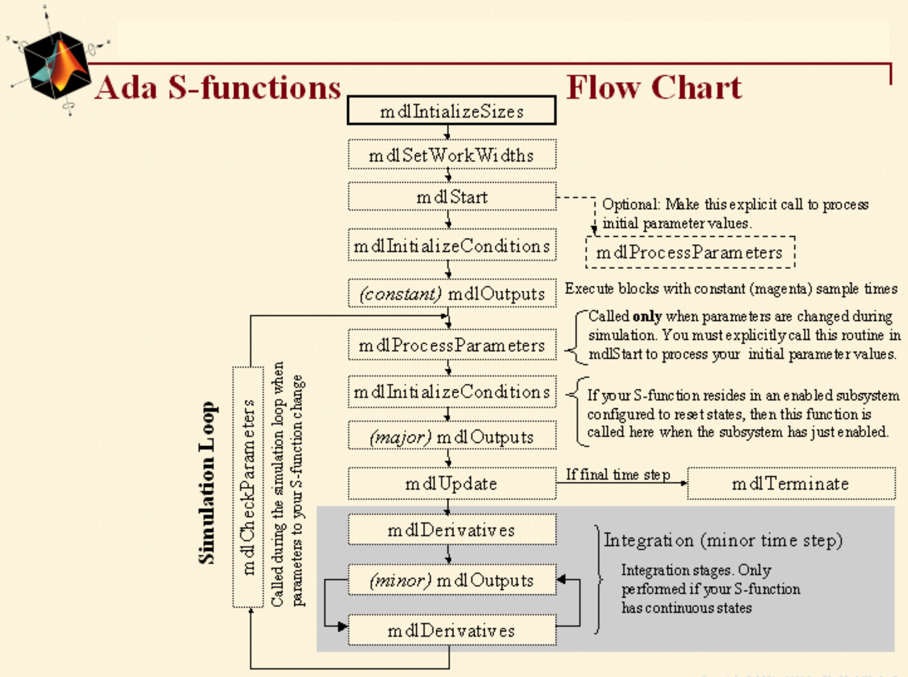

Ada S-functions
Flow Chart
mdlIntializeSizes
mdlSetWorkWidths
mdlStart
Optional: Make this explicit call to process initial parameter values.
mdlProcessParameters
mdlInitializeConditions
(constant) mdlOutputs
Execute blocks with constant (magenta) sample times
mdlProcessParameters
Called only when parameters are changed during simulation. You must explicitly call this routine in mdlStart to process your initial parameter values.
mdlInitializeConditions
If your S-function resides in an enabled subsystem configured to reset states, then this function is called here when the subsystem has just enabled.
(major) mdlOutputs
mdlUpdate
If final time step
mdlTerminate
mdlDerivatives
Integration (minor time step)
Integration stages. Only performed if your S-function has continuous states
(minor) mdlOutputs
mdlDerivatives
Simulation Loop
mdlCheckParameters
Called during the simulation loop when parameters to your S-function change

Figure A.16: The flow chart for a S-function in Simulink.

Appendix B

Simulation Results, Simulink Model, and MATLAB Codes of Supervisor Control

B.1 Simulink model

The Simulink model is shown in Figure B.1-B.7. The key codes are shown in Listing B.1-B.3.

B.2 MATLAB codes

Listing B.1: The MATLAB code for starting the Simulation.

```
clear all
% clf
clc

load('vessel.mat');           % vessel/hydrodynamic data structure
load('vesselABC.mat');        % fluid memory structure
load('MLP.mat');              % mooring line parameters
load('T_H_table.mat');        % mooring line force lookup table
load('p_m.mat');              % mooring line force polynomial (from ...
    curve fitting)

MLP.r_0(:,5)=[1950/sqrt(2);1950/sqrt(2);-1000];
MLP.r_0(:,6)=[-1950/sqrt(2);1950/sqrt(2);-1000];
MLP.r_0(:,7)=[-1950/sqrt(2);-1950/sqrt(2);-1000];
```

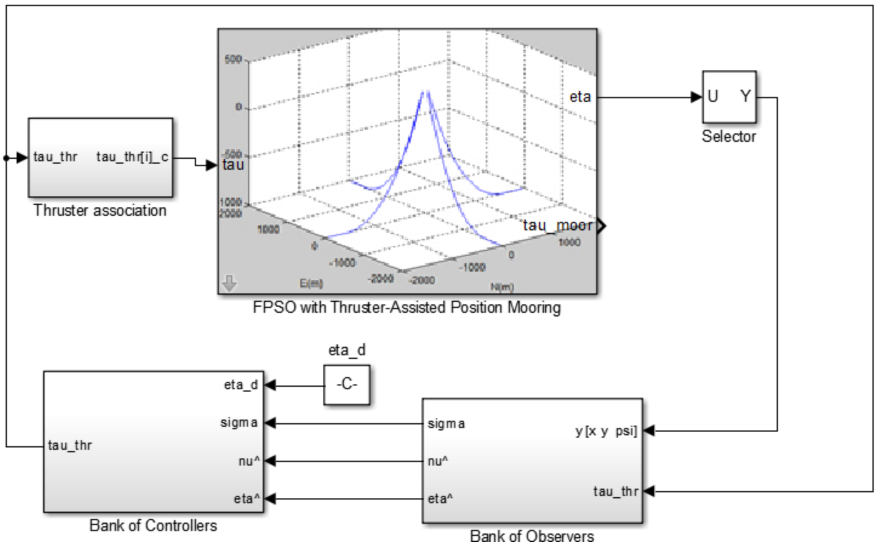


Figure B.1: The Simulink structure of the model.

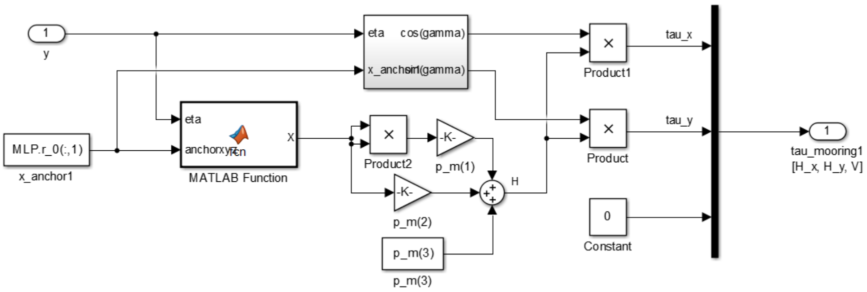


Figure B.2: The Simulink structure of the estimated mooring forces.

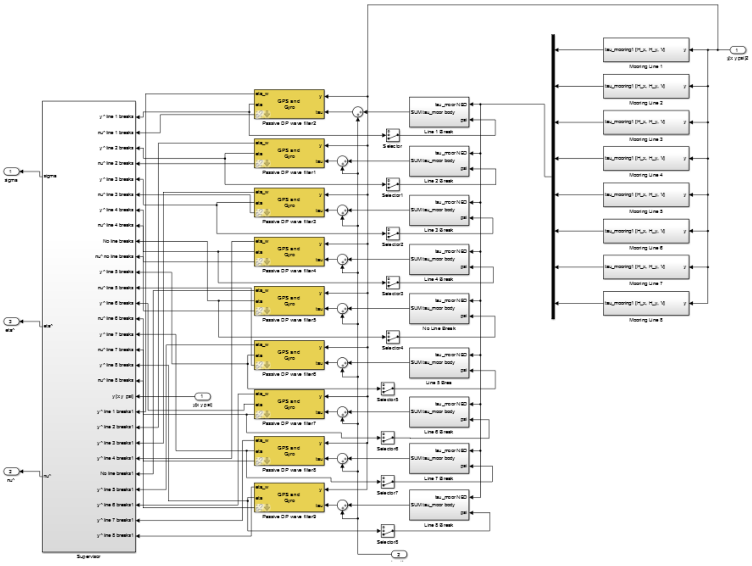


Figure B.3: The Simulink structure of the bank of observers.

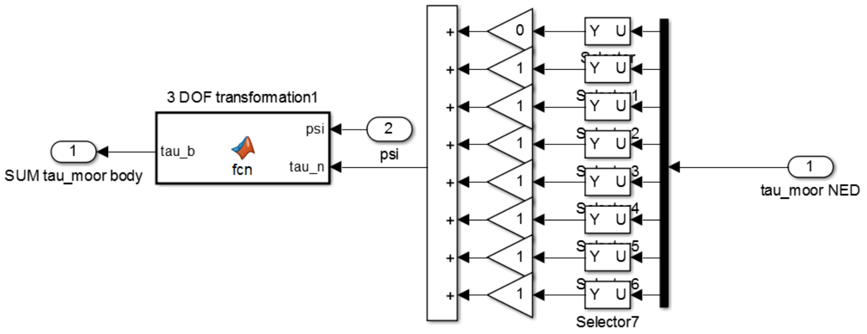


Figure B.4: The Simulink structure of mooring force switch.

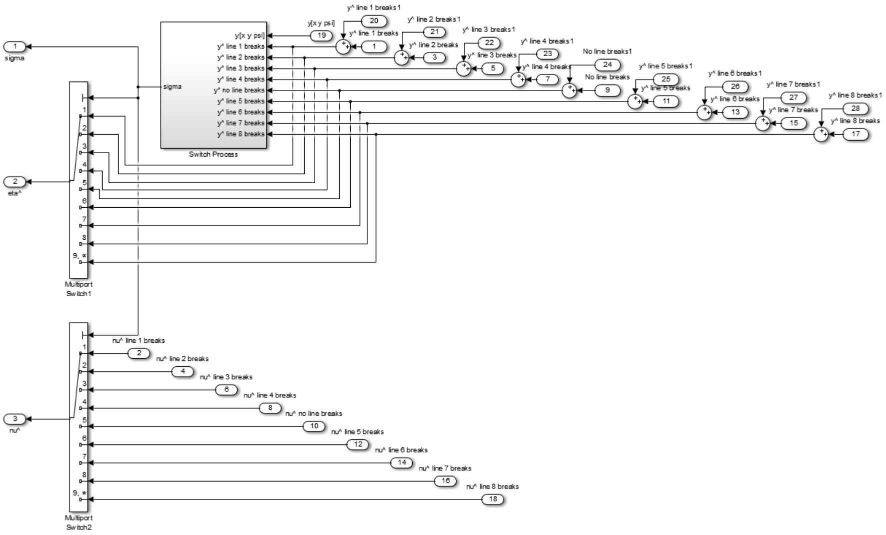


Figure B.5: The Simulink structure of the supervisor block.

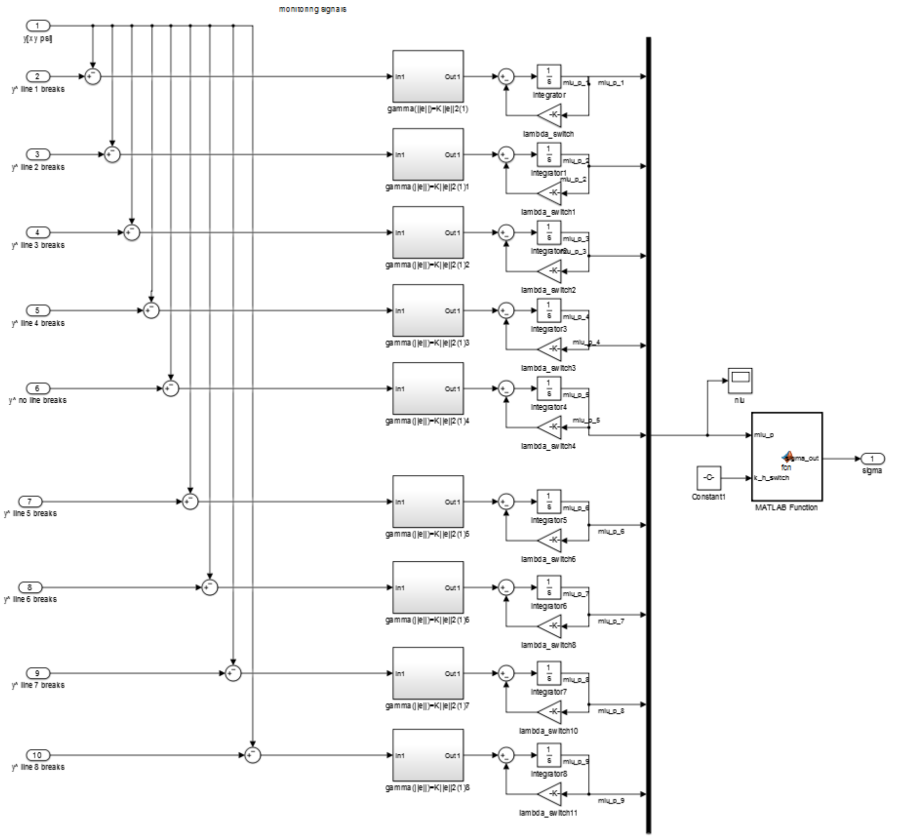


Figure B.6: The Simulink structure of the supervisor logic.

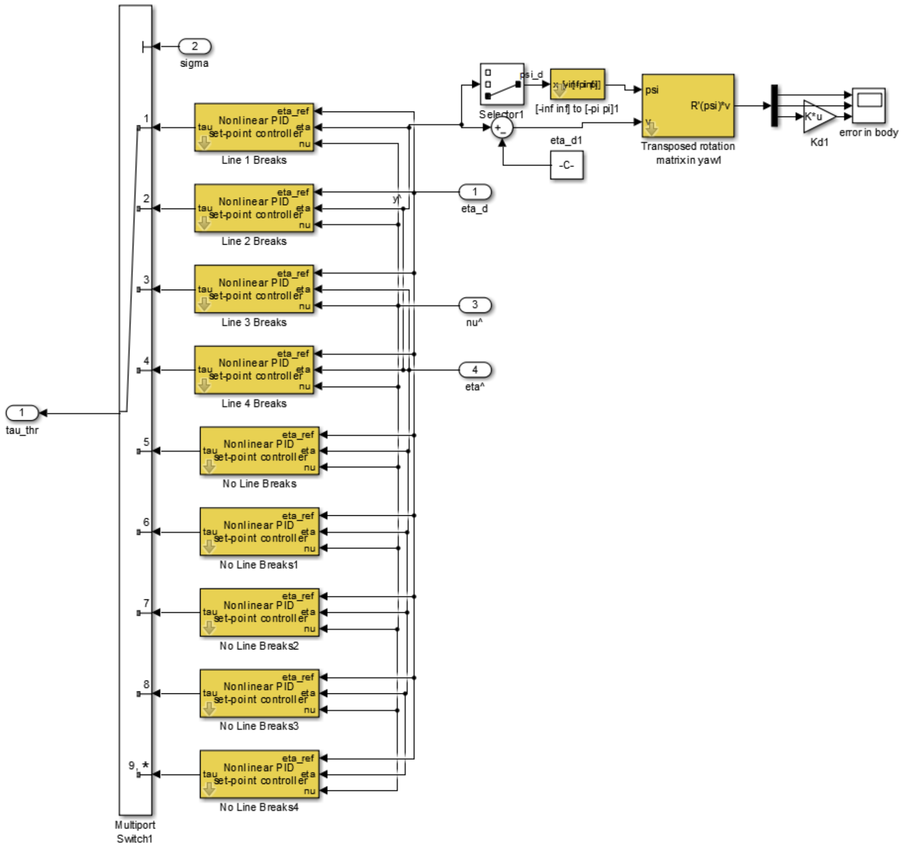


Figure B.7: The Simulink structure of the bank of controllers.

```
MLP.r_0(:,8)=[1950/sqrt(2);-1950/sqrt(2);-1000];
%% Model Parameter
waveForce       =   1;      % wave motion On--1; Off--0
waveDriftForce  =   1;      % wave drift force On--1; Off--0
currentForce    =   1;      % c urrent On--1; Off--0

mooringLines                =   [1,1,1,1,1,1,1,1];      % Mooring ...
   Lines On--1; Off--0
mooringLines1                =  [0,1,1,1,1,1,1,1];      % Mooring ...
   Lines after breaking On--1; Off--0

mooringLinesMeasurement     =   [1,1,1,1,1,1,1,1];      % Mooring ...
   Lines Measurement On--1; Off--0

measurementNoises_on_off  =   1;    % measurement noise On--1; Off--0
noise_x     =   0.005;    % variance of measurement noise in x, DGPS
noise_y     =   0.005;    % variance of measurement noise in y, DGPS
noise_psi   =   0.005/180*pi;   % variance of measurement noise in ...
   yaw, Compass, ...
   http://www.shipmotion.se/files/SMC%20IMU%20User%20Guide%20v22.pdf
MeasurementNoiseSampleTime  =   0.2;    %   measurement noises sample ...
   time
% eta_0   =   [-7,-10,0,0,0,-135/180*pi];
eta_0   =   [-2,-3.3,0,0,0,-135/180*pi];
nu_0    =   [0.2,0.05,0.05,0,0,0];
%% Environmental Parameters
nu_wind     =   8;          % wind velocity
alpha_wind  =   pi/4;       % wind direction

Hs_wave     =   5.5;          % significant wave height
beta_wave   =   45*pi/180;  % mean wave direction

nu_current   =   0.5;           % current velocity(m/s)
beta_current =   0;           % current direction(deg)


% for i=1:length(T_H_table.X)
% T_H_table.X(i)=1950+T_H_table.X(i);
% end
```

```

%%
M_RB    =   vessel.MRB([1,2,6],[1,2,6]);
M_A     =   vesselABC.MA([1,2,6],[1,2,6]);
M       =   M_RB+M_A;
invM    =   inv(M);

D_L     =   vessel.B([1,2,6],[1,2,6],34);
D_mo    =   0.1*vessel.B(:,:,34);




% for j=1:m_moor
%     for k=1:n_moor
%         r{j,k}=r_0((j-1)*3*(n_moor-1)+ 3*(k-1)+ 1:(j-1) ...
   *3*(n_moor-1)+ 3*(k-1)+3);
%     end
% end
%%
%% Thruster Allocation
y_1     =   7.5;
y_2     =   -7.5;
x_3     =   -75;

T_thr   =   [1      1       0;
             0      0       1;
             y_1    y_2     x_3];
T_w_thr =   inv(T_thr);
% T_w =   T'*inv(T*T'); %(Fossen 12.252)


%% %%%%%%%%%%%%    Part 1  Observer    %%%%%%%%%%%%%%
% -------------    Passive Observer    --------------
b_0_PassObs         =   [0;0;0];
nu_0_PassObs        =   [0;0;0];
eta_0_PassObs       =   eta_0([1,2,6]);
epsilon_0_PassObs   =   [0;0;0;0;0;0];
eta_w_0_PassObs     =   epsilon_0_PassObs(1:3);
```

```
xi_w_0_PassObs     =    epsilon_0_PassObs(4:6);
D_L_PassObs=D_L+length(find(mooringLines==1))*D_mo([1,2,6],[1,2,6]);

invTb_PassObs       =   inv(diag([1000,1000,1000]));

zeta_ni     =   1;      % Fossen pp.317
lamda_i     =   0.1;    % Fossen pp.317 Relative damping ratio of ...
   wave spectrum

lambda      =   diag([lamda_i,lamda_i,lamda_i]);


omega_oi    =   [2*pi/7,2*pi/7,2*pi/7];
omega_ci    =   [1.1,1.1,1.1];

K_11        =   -2*(zeta_ni-lamda_i)*omega_ci(1)./omega_oi(1);
K_12        =   -2*(zeta_ni-lamda_i)*omega_ci(2)./omega_oi(2);
K_16        =   -2*(zeta_ni-lamda_i)*omega_ci(3)./omega_oi(3);

K_17        =   2*omega_oi(1)*(zeta_ni-lamda_i);
K_18        =   2*omega_oi(2)*(zeta_ni-lamda_i);
K_112       =   2*omega_oi(3)*(zeta_ni-lamda_i);

K_21        =   omega_ci(1);
K_22        =   omega_ci(2);
K_26        =   omega_ci(3);

OMEGA       =   diag(omega_oi);
DELTA       =   diag([1,1,1]);

Aw          =   [zeros(3)       eye(3);
                -OMEGA*OMEGA    -2*DELTA*OMEGA];
Cw          =   [diag([0,0,0]),diag([0.05,0.05,0.01])];
T_b_PassObs     =   diag([1000,1000,1000]);

K1_PassObs      =   [diag([K_11,K_12,K_16]);diag([K_17,K_18,K_112])];
K2_PassObs      =   diag([K_21,K_22,K_26]);

K4_PassObs      =   diag([0.3,0.3,0.01]);
```

```
K3_PassObs      =   0.1*K4_PassObs;

%% Supervisor Switch
k_e_switch          =   [1,1];          % gain of class K function
lambda_switch       =   0.1;        % constant non-negative ...
   forgetting factor
eta_0_SwitchSup     =   35;         % initial value of the switch ...
   function 15
k_h_switch          =   0.5;        % positive hysteresis constant


%% %%%%%%%%%%%   Part 2  Controller  %%%%%%%%%%%%%%%%
% ------------   PID Controller    ------------------


% 4 Mooring Lines
% mooringLines    =   [1,1,1,1,1,1,1,1];
% Kp_PIDCtr_5     =   diag([1e3 1e3 1e6]);
% Kd_PIDCtr_5     =   diag([0 0 0]);
% Ki_PIDCtr_5     =   diag([2e2 1e3 1e6]);
Kp_PIDCtr_5     =   diag([0 0 3e6]);
Kd_PIDCtr_5     =   diag([0 0 0]);
Ki_PIDCtr_5     =   diag([0 0 1e6]);

% 3 Mooring Lines
% mooringLines    =   [0,1,1,1,1,1,1,1];
Kp_PIDCtr_1     =   diag([10e6 1e6 3e6]);
Kd_PIDCtr_1     =   diag([20e6 0 0]);
Ki_PIDCtr_1     =   diag([2e4 0 1e6]);


% mooringLines    =   [1,0,1,1,1,1,1,1];
Kp_PIDCtr_2     =   diag([5e6 50e5 3e6]);
Kd_PIDCtr_2     =   diag([20e6 0 0]);
Ki_PIDCtr_2     =   diag([2e4 5e4 1e6]);

% mooringLines    =   [1,1,0,1,1,1,1,1];
Kp_PIDCtr_3     =   diag([5e6 10e6 3e6]);
Kd_PIDCtr_3     =   diag([20e6 0 0]);
```

```
Ki_PIDCtr_3     =   diag([2e4 5e4 1e6]);

% mooringLines    =   [1,1,1,0,1,1,1,1];
Kp_PIDCtr_4     =   diag([5e6 10e6 3e6]);
Kd_PIDCtr_4     =   diag([20e6 0 0]);
Ki_PIDCtr_4     =   diag([2e4 5e4 1e6]);


% mooringLines    =   [1,1,1,1,0,1,1,1];
Kp_PIDCtr_6     =   diag([5e6 10e6 3e6]);
Kd_PIDCtr_6     =   diag([20e6 0 0]);
Ki_PIDCtr_6     =   diag([2e4 5e4 1e6]);


% mooringLines    =   [1,1,1,1,1,0,1,1];
Kp_PIDCtr_7     =   diag([5e6 10e6 3e6]);
Kd_PIDCtr_7     =   diag([20e6 0 0]);
Ki_PIDCtr_7     =   diag([2e4 5e4 1e6]);

% mooringLines    =   [1,1,1,1,1,1,0,1];
Kp_PIDCtr_8     =   diag([5e6 10e6 3e6]);
Kd_PIDCtr_8     =   diag([20e6 0 0]);
Ki_PIDCtr_8     =   diag([2e4 0 1e6]);

% mooringLines    =   [1,1,1,1,1,1,1,0];
Kp_PIDCtr_9     =   diag([5e6 10e6 3e6]);
Kd_PIDCtr_9     =   diag([20e6 0 0]);
Ki_PIDCtr_9     =   diag([2e4 5e4 1e6]);

%% Simulation
startDraw=250;
breakTime=400;
tend=600;

% startDraw=0;
% breakTime=250;
% tend=200;

sim Supervisor
```

```
plotSupervisor
```

Listing B.2: The MATLAB code for plotting the figures.

```

clf('reset');

% time=ObserverCompare.time(end*0.95:end);
if mooringLines1==[1,1,1,1,1,1,1,1]
    condition='No_Line';
elseif mooringLines1==[0,1,1,1,1,1,1,1]
    condition='Line_1_break';
elseif mooringLines1==[1,0,1,1,1,1,1,1]
    condition='Line_2_break';
elseif mooringLines1==[1,1,0,1,1,1,1,1]
    condition='Line_3_break';
elseif mooringLines1==[1,1,1,0,1,1,1,1]
    condition='Line_4_break';
elseif mooringLines1==[1,1,1,1,0,1,1,1]
    condition='Line_5_break';
elseif mooringLines1==[1,1,1,1,1,0,1,1]
    condition='Line_6_break';
elseif mooringLines1==[1,1,1,1,1,1,0,1]
    condition='Line_7_break';
elseif mooringLines1==[1,1,1,1,1,1,1,0]
    condition='Line_8_break';
else
    condition='else'
end
%% sigma v.s. mu

timeStart=find(positionNED.time==startDraw);
time=sigma.time(timeStart:end)-startDraw;

for i=1:length(sigma.signals.values)
    if sigma.signals.values(i,1)==5
        sigma.signals.values(i,1)=0;
    end
    if sigma.signals.values(i,1)>5
        sigma.signals.values(i,1)=sigma.signals.values(i,1)-1;
    end
    if sigma.signals.values(i,2)==5
        sigma.signals.values(i,2)=0;
    end
    if sigma.signals.values(i,2)>5
        sigma.signals.values(i,2)=sigma.signals.values(i,2)-1;
    end
end


F1=figure(1);
subplot(2,1,1)
plot(time,sigma.signals.values(end-length(time)+1:end,1),'r-', ...
    'LineWidth',2)
hold on
plot(time,sigma.signals.values(end-length(time)+1:end,2),'b--', ...
    'LineWidth',2)
% title('\sigma');
legend('Supervisor Signal','Line Breaks in Real Time');
xlabel('time(s)')
ylabel('conditions \sigma')
xlim([time(1),time(end)]);
ylim([0,10])
set(gca,'YTick',0:1:8);
grid on;

subplot(2,1,2)
plot(time,miu.signals.values(end-length(time)+1:end,1),'r-', ...
    'LineWidth',1)
hold on
plot(time,miu.signals.values(end-length(time)+1:end,2),'g-', ...
    'LineWidth',1)
hold on
plot(time,miu.signals.values(end-length(time)+1:end,3),'c-', ...
    'LineWidth',1)
hold on
```

```
plot(time,miu.signals.values(end-length(time)+1:end,4),'m-', ...
    'LineWidth',1)
hold on
plot(time,miu.signals.values(end-length(time)+1:end,5),'b-', ...
    'LineWidth',2)
hold on
plot(time,miu.signals.values(end-length(time)+1:end,6),'r--', ...
    'LineWidth',2)
hold on
plot(time,miu.signals.values(end-length(time)+1:end,7),'g--', ...
    'LineWidth',2)
hold on
plot(time,miu.signals.values(end-length(time)+1:end,8),'c--', ...
    'LineWidth',2)
hold on
plot(time,miu.signals.values(end-length(time)+1:end,9),'m--', ...
    'LineWidth',2)
% title('\mu_{\sigma}');
legend('Line 1 Breaks','Line 2 Breaks','Line 3 Breaks','Line 4 ...
    Breaks','No Line Break',...
     'Line 5 Breaks','Line 6 Breaks','Line 7 Breaks','Line 8 Breaks');
xlabel('time(s)')
ylabel('monitoring signal \mu')
xlim([time(1),time(end)]);
grid on;

print(F1,'-dpng',['.\image\switch',condition,'_8.png'],'-r300')
% print(F1,'-dpng',['.\image\switch',condition,'_start.png'],'-r300')

%% Positions in [X Y Eta]
timeBreak=find(positionNED.time==breakTime);
time1=positionNED.time(timeStart:timeBreak)-startDraw;
time2=positionNED.time(timeBreak:end)-startDraw;

F2=figure(2);
subplot(3,1,1)
plot(time1,positionNED.signals(1).values(timeStart:timeBreak,1), ...
    'r-', 'LineWidth',2)
hold on
```

```
plot(time2,positionNED.signals(1).values(timeBreak:end,1), 'b-', ...
    'LineWidth', 2)
% title('Position in NED');
legend('Before Line Breaks','After Line Breaks','Location','SouthEast');
xlabel('time(s)')
ylabel('North(m)')
grid on;

subplot(3,1,2)
plot(time1,positionNED.signals(2).values(timeStart:timeBreak,1), ...
    'r-','LineWidth',2)
hold on
plot(time2,positionNED.signals(2).values(timeBreak:end,1),'b-', ...
    'LineWidth',2)
legend('Before Line Breaks','After Line Breaks','Location', 'SouthEast');
xlabel('time(s)')
ylabel('East(m)')
grid on;

subplot(3,1,3)
plot(time1,positionNED.signals(3).values (timeStart:timeBreak,1)* ...
    180/pi, 'r-', 'LineWidth',2)
hold on
plot(time2,positionNED.signals(3).values(timeBreak:end,1) *180/pi, ...
    'b-', 'LineWidth',2)
legend('Before Line Breaks','After Line Breaks','Location','SouthEast');
xlabel('time(s)')
ylabel('Yaw(deg)')
grid on;

print(F2,'-dpng',['.\image\position',condition,'_8.png'], '-r300')
%% Horizontal Displacement
F3=figure(3);
plot(positionNED.signals(2).values(timeStart:timeBreak,1), ...
    positionNED.signals(1).values (timeStart:timeBreak,1), ...
    'r-','LineWidth', 2)
hold on
plot(positionNED.signals(2).values(timeBreak:end,1), positionNED. ...
    signals(1).values(timeBreak:end,1),'b-','LineWidth',2)
```

```
% title('Position in NED');
legend('Before Line Breaks','After Line Breaks','Location','SouthEast');
xlabel('East(m)')
ylabel('North(m)')
grid on;

print(F3,'-dpng',['.\image\NED',condition,'_8.png'],'-r300')
```

c

Listing B.3: The MATLAB code of the switch logic.

```
function sigma_out  = fcn(miu_p,k_h_switch)
persistent sigma_m
if isempty(sigma_m)
    sigma_m=5;
end

sigma_n1=find(miu_p==min(miu_p));
sigma_n=sigma_n1(1);
Kniu_s=(1+k_h_switch)*miu_p(sigma_n);
niu_sm=miu_p(sigma_m);

if ((1+k_h_switch)*miu_p(sigma_n)<miu_p(sigma_m))
    sigma_out=sigma_n;
else
    sigma_out=sigma_m;
end
sigma_m=sigma_out;
```

B.3 Simulation results of Simulation III.1 and III.2

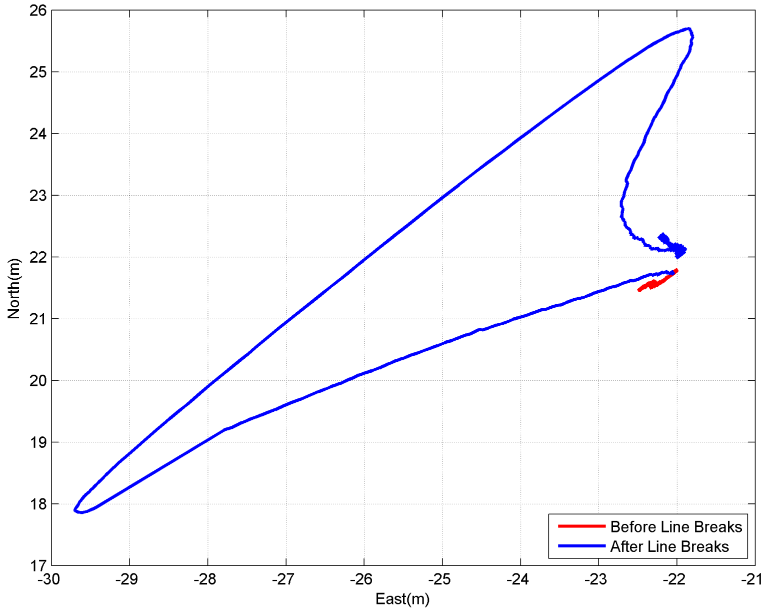


Figure B.8: Horizontal position of the FPSO. Line 2 breaks in Simulation III.1.

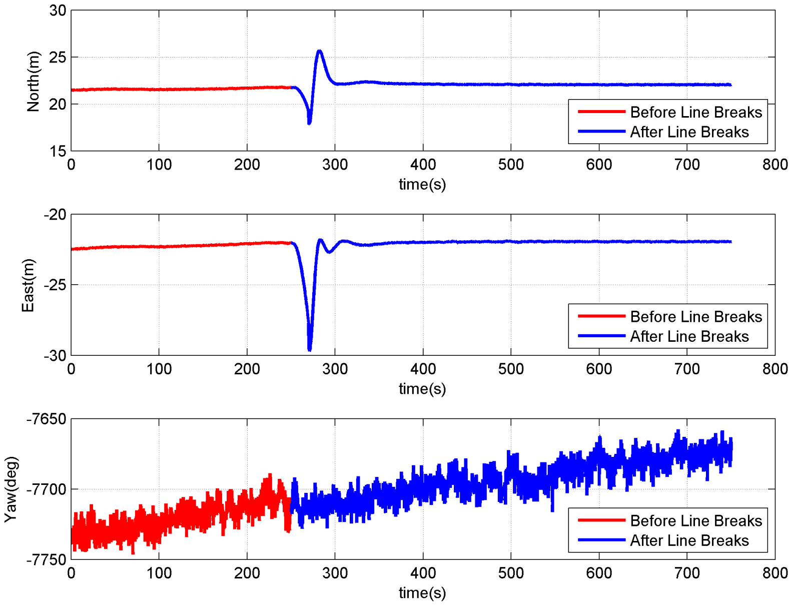


Figure B.9: Position and rotation in time domain. Line 2 breaks in Simulation III.1.

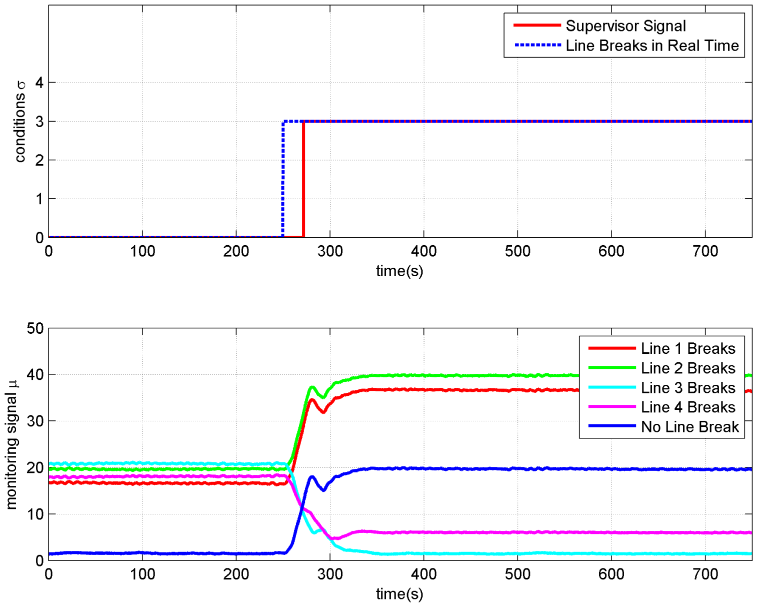


Figure B.10: Switching logic outputs. Line 3 breaks in Simulation III.1.

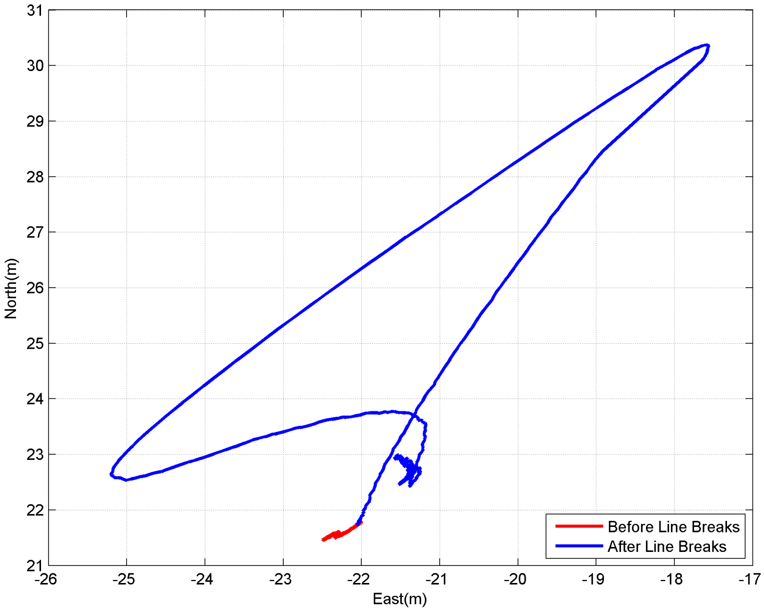


Figure B.11: Horizontal position of the FPSO. Line 3 breaks in Simulation III.1.

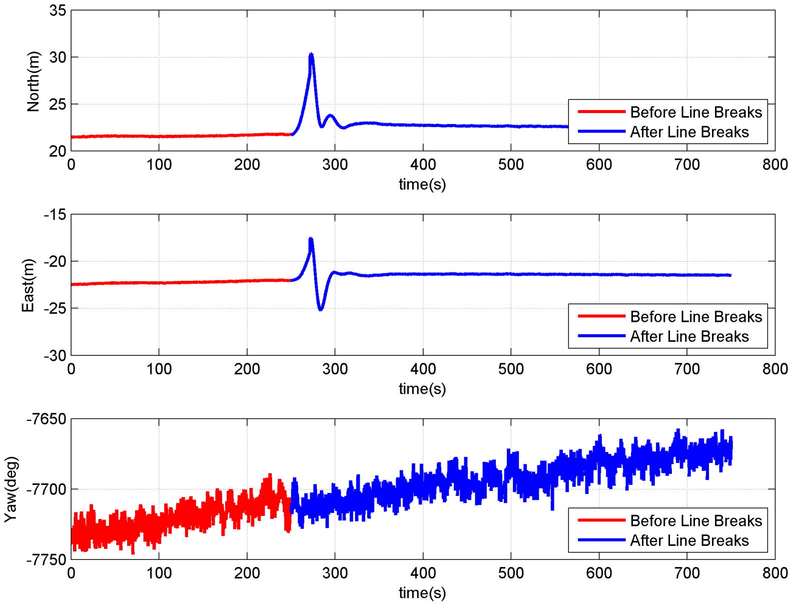


Figure B.12: Position and rotation in time domain. Line 3 breaks in Simulation III.1.

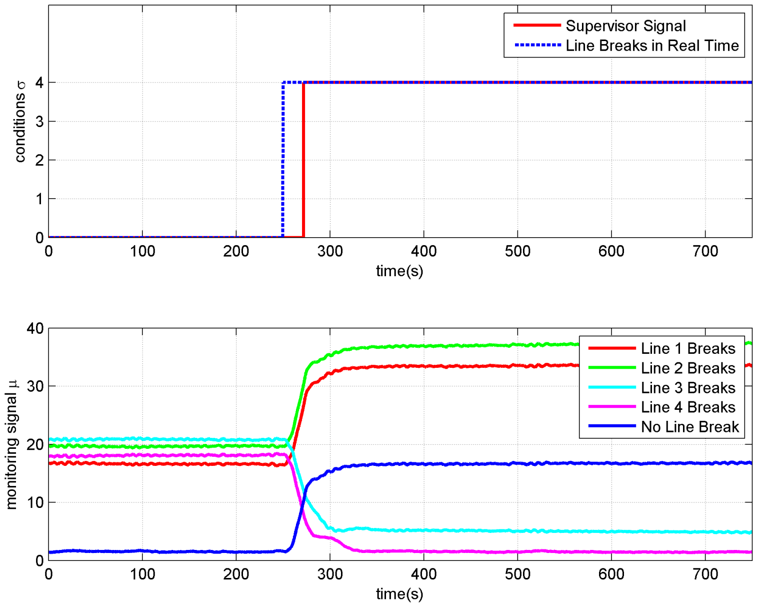


Figure B.13: Switching logic outputs. Line 4 breaks in Simulation III.1.

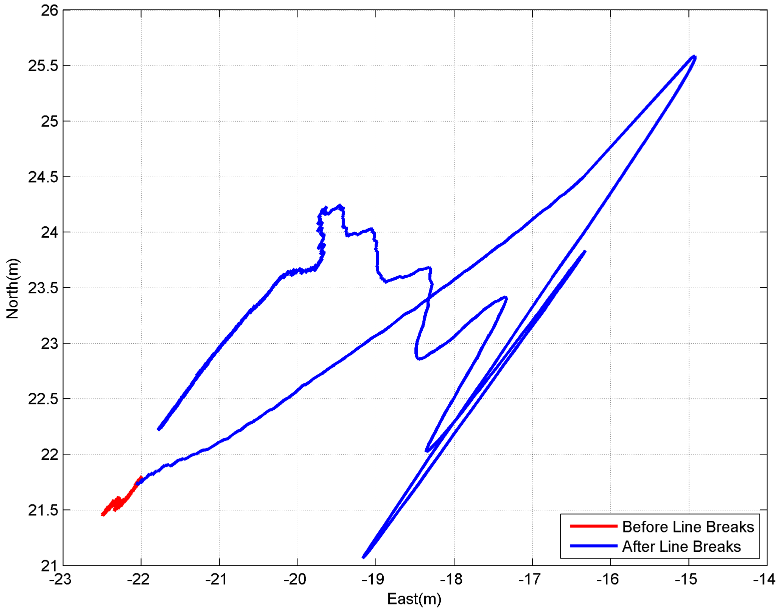


Figure B.14: Horizontal position of the FPSO. Line 4 breaks in Simulation III.1.

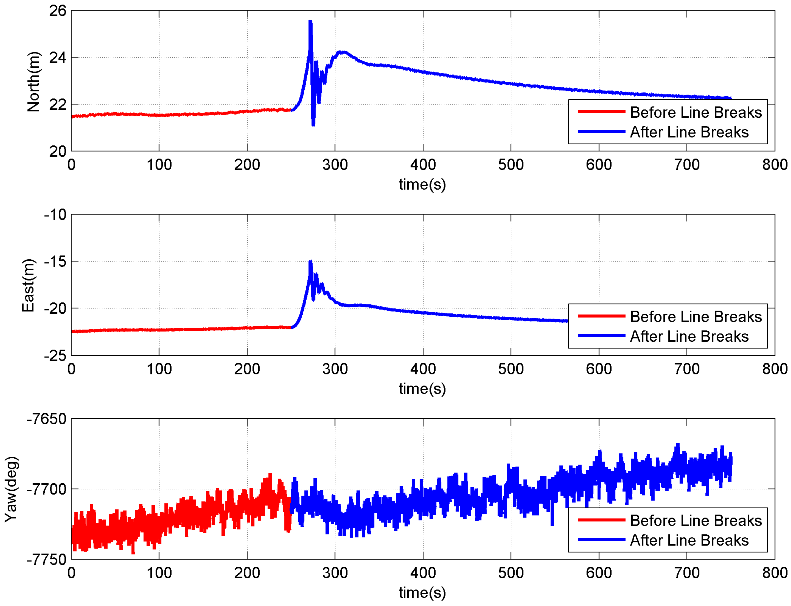


Figure B.15: Position and rotation in time domain. Line 4 breaks in Simulation III.1.

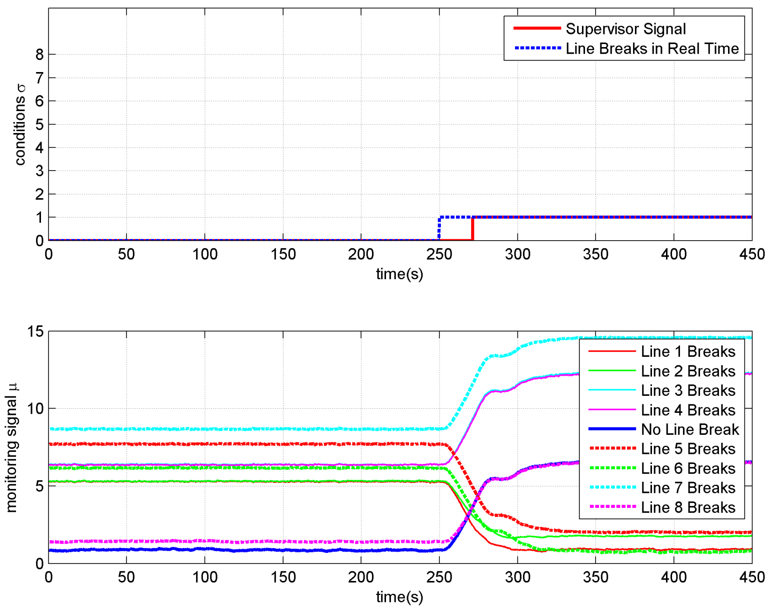


Figure B.16: Switching logic outputs. Line 1 breaks in Simulation III.2.

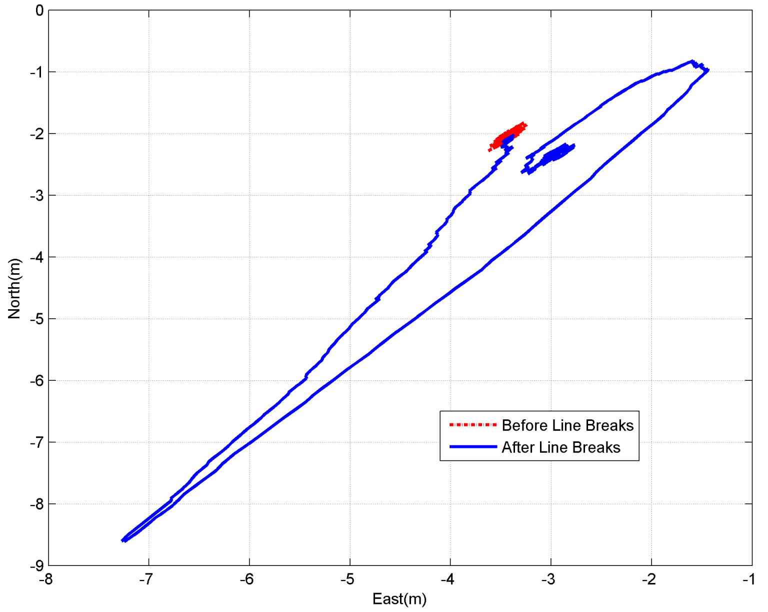


Figure B.17: Horizontal position of the FPSO. Line 1 breaks in Simulation III.2.

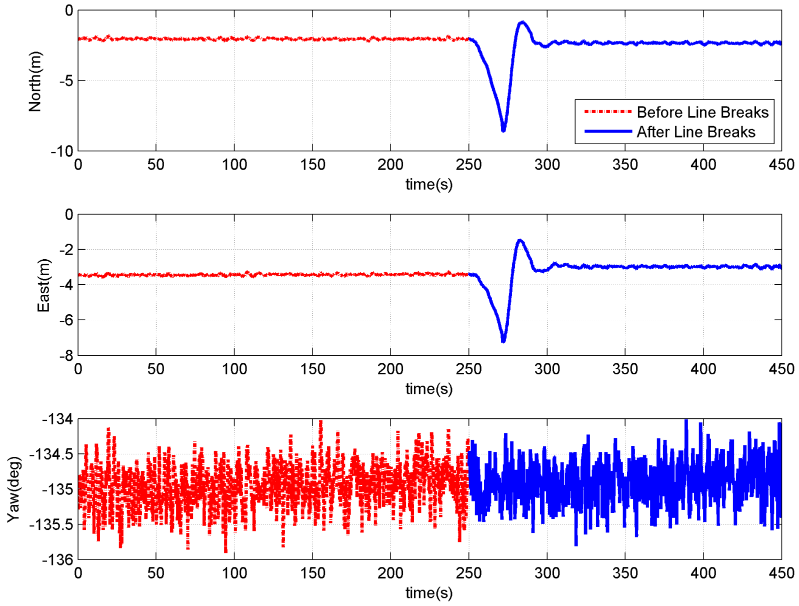


Figure B.18: Position and rotation in time domain. Line 1 breaks in Simulation III.2.

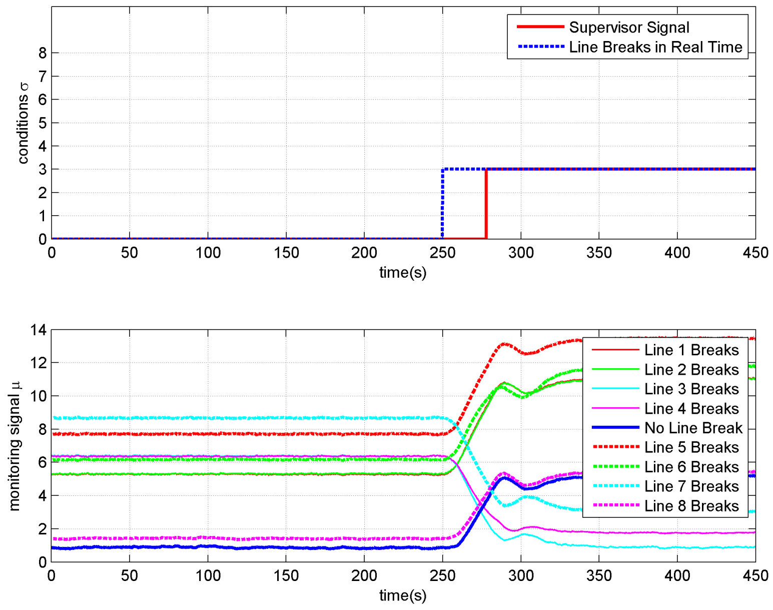


Figure B.19: Switching logic outputs. Line 3 breaks in Simulation III.2.

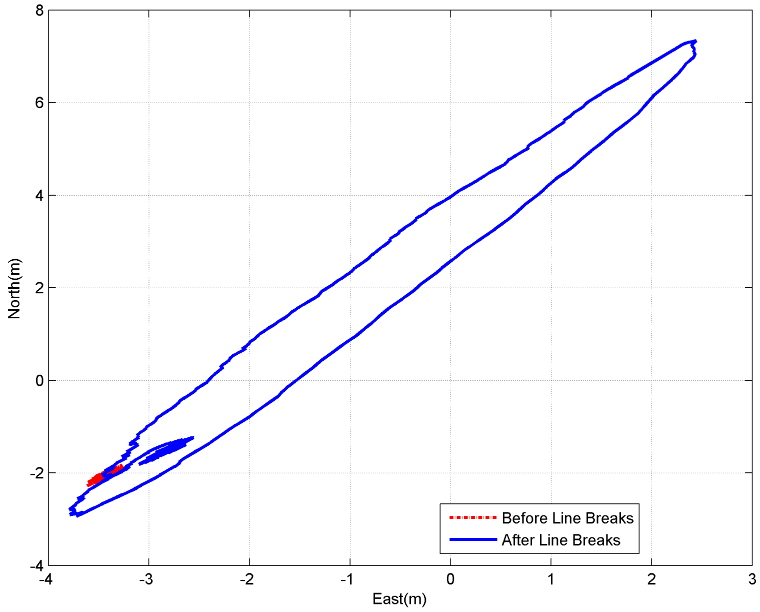


Figure B.20: Horizontal position of the FPSO. Line 3 breaks in Simulation III.2.

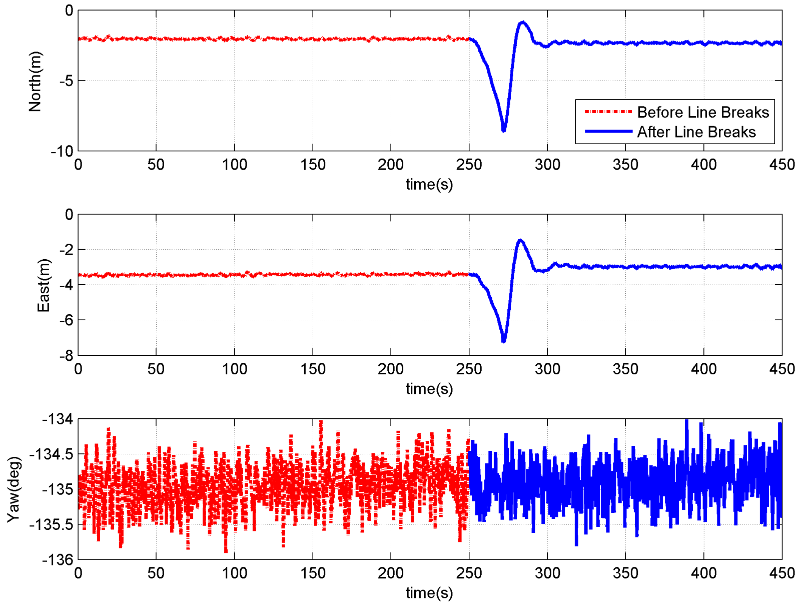


Figure B.21: Position and rotation in time domain. Line 1 breaks in Simulation III.2.

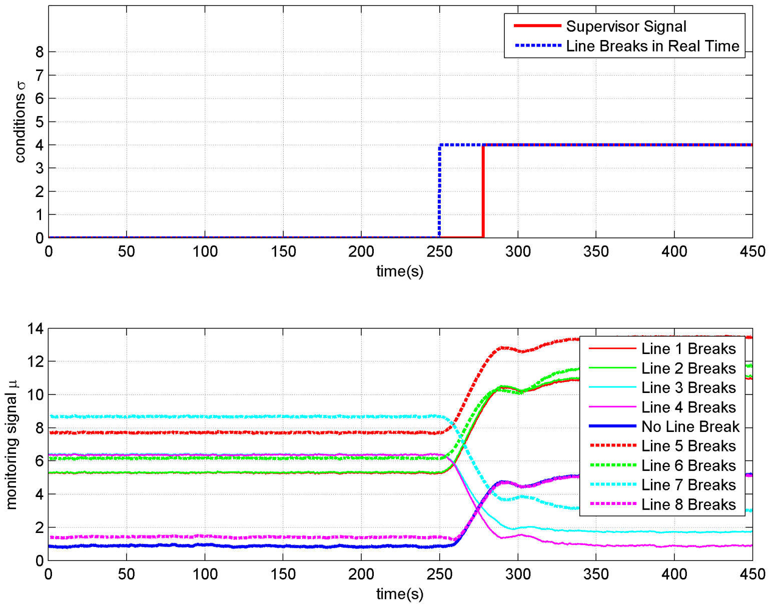


Figure B.22: Switching logic outputs. Line 4 breaks in Simulation III.2.

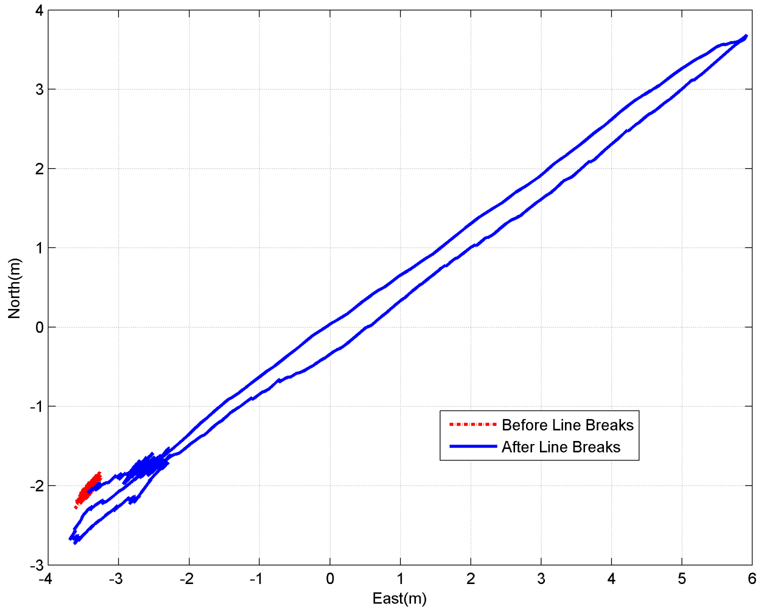


Figure B.23: Horizontal position of the FPSO. Line 4 breaks in Simulation III.2.

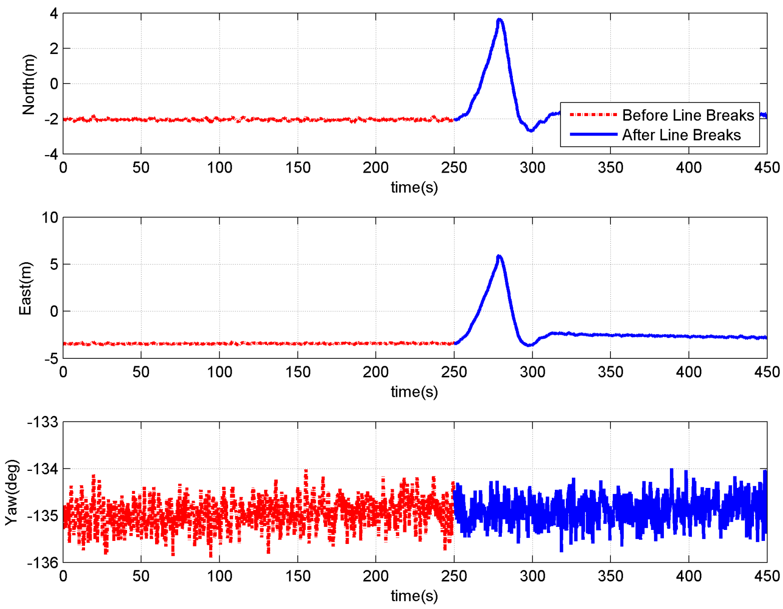


Figure B.24: Position and rotation in time domain. Line 4 breaks in Simulation III.2.

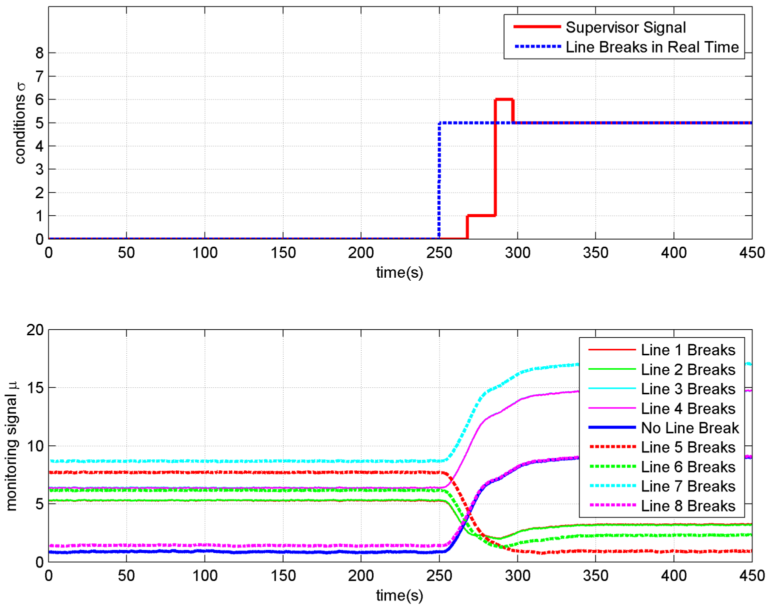


Figure B.25: Switching logic outputs. Line 5 breaks in Simulation III.2.

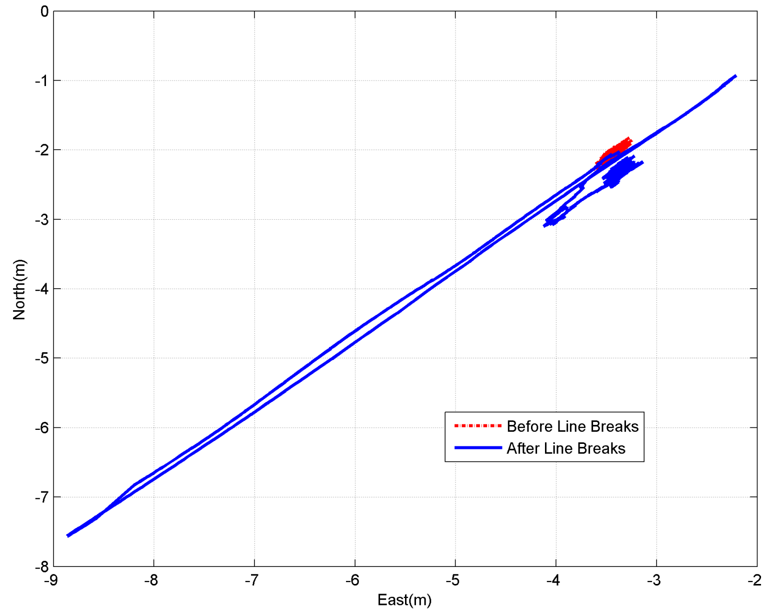


Figure B.26: Horizontal position of the FPSO. Line 5 breaks in Simulation III.2.

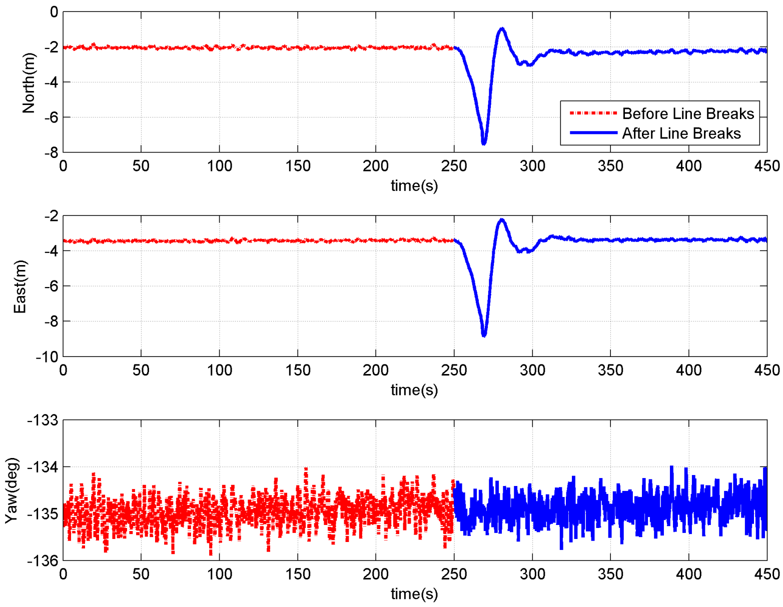


Figure B.27: Position and rotation in time domain. Line 5 breaks in Simulation III.2.

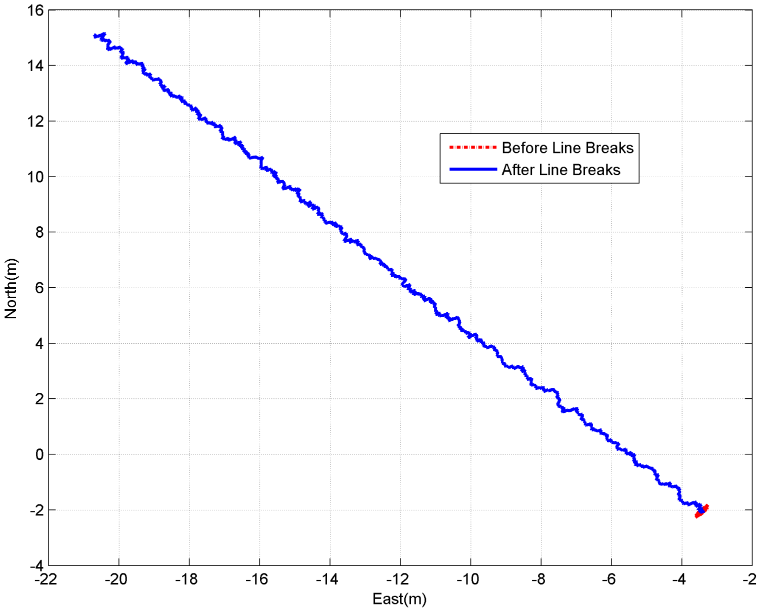


Figure B.28: Horizontal position of the FPSO. Line 6 breaks in Simulation III.2.

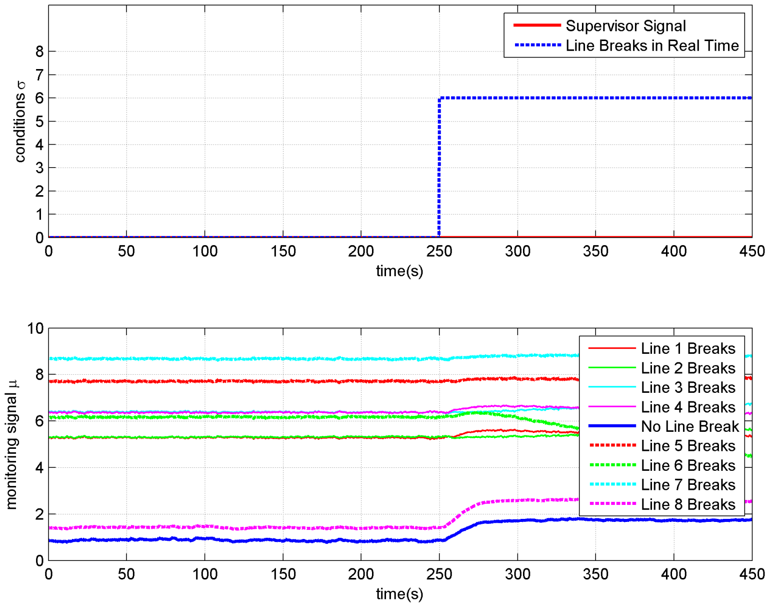


Figure B.29: Switching logic outputs. Line 6 breaks in Simulation III.2.

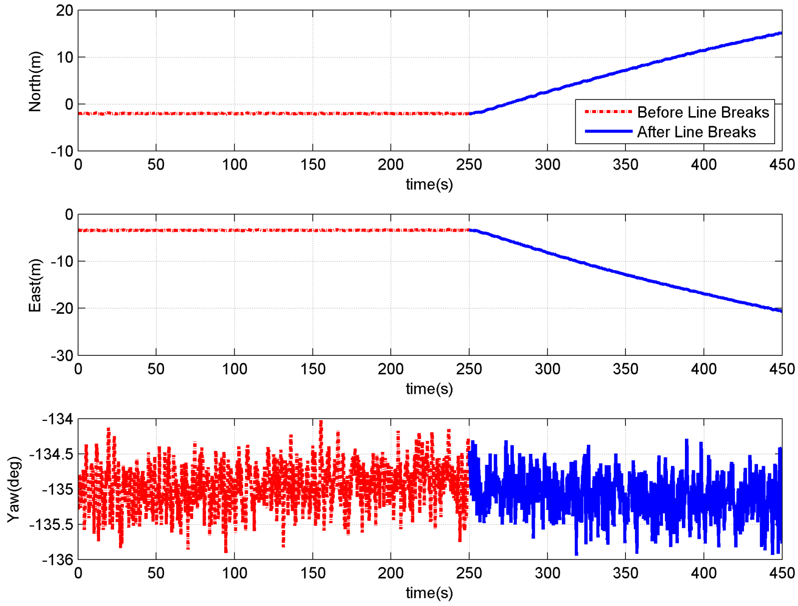


Figure B.30: Position and rotation in time domain. Line 6 breaks in Simulation III.2.

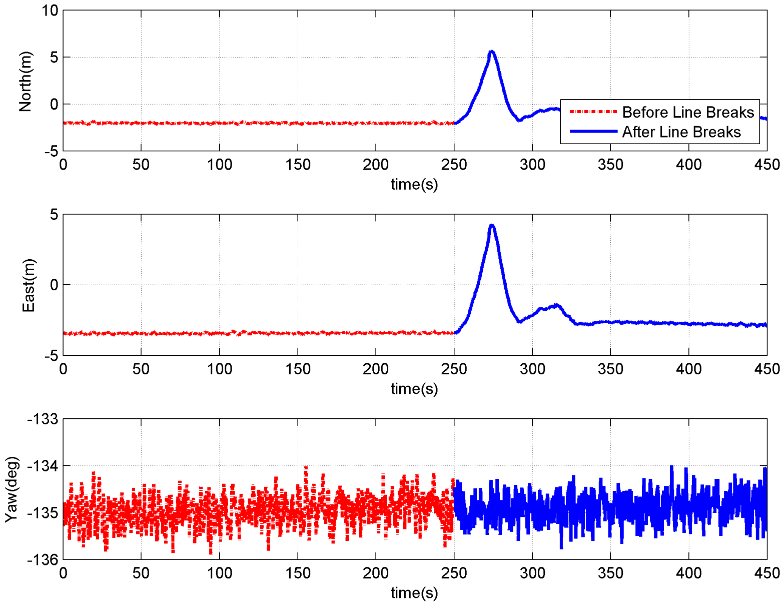


Figure B.31: Position and rotation in time domain. Line 7 breaks in Simulation III.2.

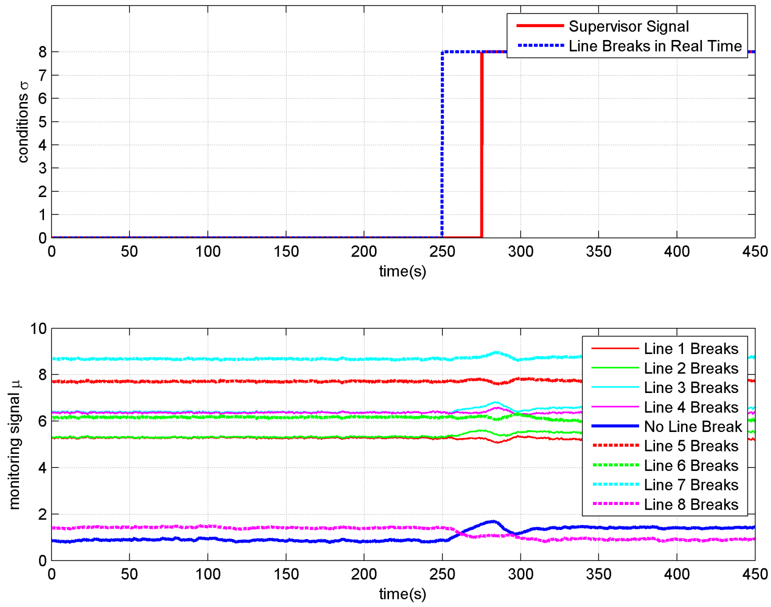


Figure B.32: Switching logic outputs. Line 8 breaks in Simulation III.2.

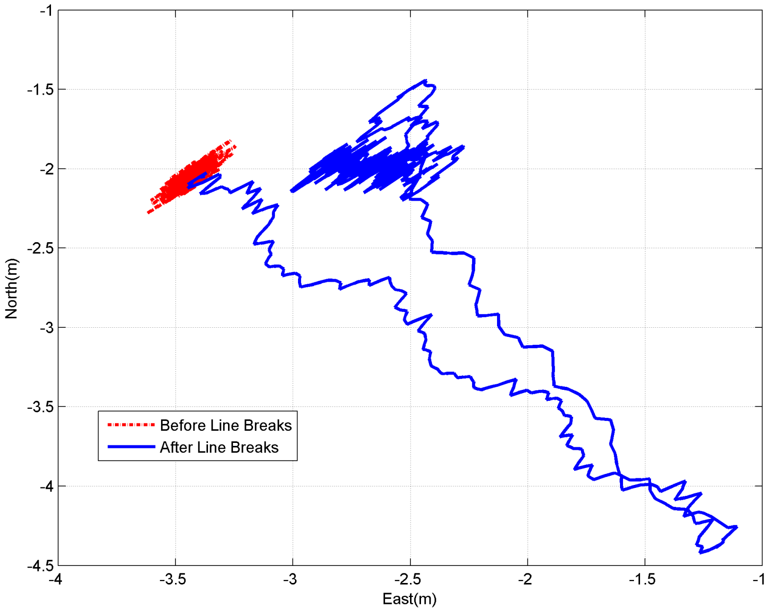


Figure B.33: Horizontal position of the FPSO. Line 8 breaks in Simulation III.2.

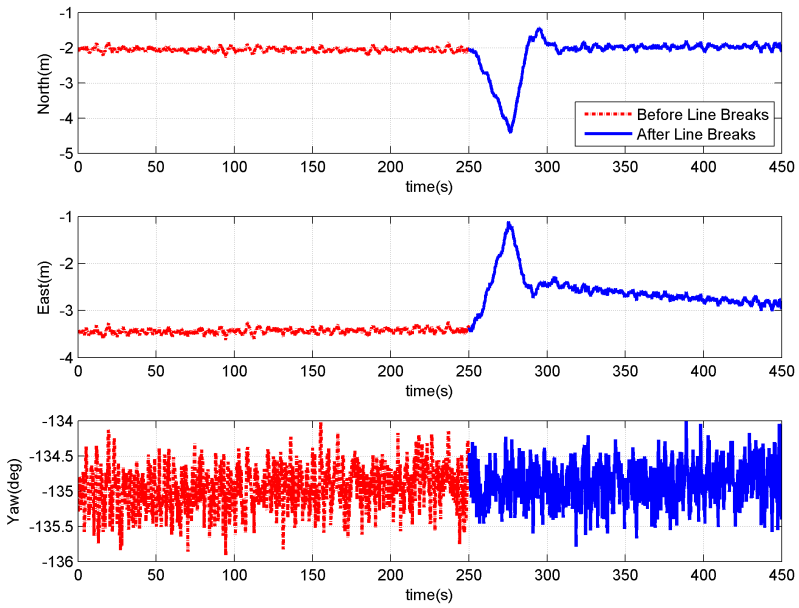


Figure B.34: Position and rotation in time domain. Line 8 breaks in Simulation III.2.

Appendix C

Simulation Results, Simulink Model, and MATLAB Codes of Tension-based Position Reference

Listing C.1: The MATLAB code for starting the Simulation.

```
clear all
% clf
clc

load('vessel.mat');         % vessel/hydrodynamic data structure
load('vesselABC.mat');      % fluid memory structure
load('MLP.mat');            % mooring line parameters
load('T_H_table.mat');      % mooring line force lookup table
load('p_m_TP2.mat');            % mooring line force polynomial (from ...
   curve fitting)
load('p_m.mat');            % mooring line force polynomial (from ...
   curve fitting)
%% Model Parameter
waveForce       =   1;      % wave motion On--1; Off--0
waveDriftForce  =   1;      % wave drift force On--1; Off--0
currentForce    =   1;      % c urrent On--1; Off--0

mooringLines                =   [1,1,1,1,1,1,1,1];      % Mooring ...
   Lines On--1; Off--0
```

```
mooringLines1                 =  [1,1,1,1,1,1,1,1];      % Mooring ...
    Lines after breaking On--1; Off--0

mooringLinesMeasurement     =   [1,1,1,1];      % Mooring Lines ...
    Measurement On--1; Off--0

measurementNoises_on_off  =   1;                  % measurement noise ...
    On--1; Off--0
TD_noises_on_off          =   1;                  % Tension Measurement ...
    Noise
noise_x      =   0.005;     % variance of measurement noise in x, DGPS
noise_y      =   0.005;     % variance of measurement noise in y, DGPS
noise_psi    =   0.005/180*pi;   % variance of measurement noise in ...
    yaw, Compass, ...
    http://www.shipmotion.se/files/SMC%20IMU%20User%20Guide%20v22.pdf
noise_TD     =   0.0001;

MeasurementNoiseSampleTime  =   0.2;     %   measurement noises sample ...
    time
eta_0    =   [22,-22,0,0,0,-135/180*pi];
nu_0     =   [0.2,0.05,0.05,0,0,0];

MLP.r_0(:,5)=[1950/sqrt(2);1950/sqrt(2);-1000];
MLP.r_0(:,6)=[-1950/sqrt(2);1950/sqrt(2);-1000];
MLP.r_0(:,7)=[-1950/sqrt(2);-1950/sqrt(2);-1000];
MLP.r_0(:,8)=[1950/sqrt(2);-1950/sqrt(2);-1000];

%% Environmental Parameters
nu_wind      =   8;           % wind velocity
alpha_wind   =   pi/4;        % wind direction

Hs_wave      =   5.5;           % significant wave height
beta_wave    =   45*pi/180;   % mean wave direction

nu_current    =   0.1;            % current velocity(m/s)
beta_current =   45;             % current direction(deg)


% for i=1:length(T_H_table.X)
% T_H_table.X(i)=1950+T_H_table.X(i);
% end

%%
M_RB    =   vessel.MRB([1,2,6],[1,2,6]);
M_A     =   vesselABC.MA([1,2,6],[1,2,6]);
M       =   M_RB+M_A;
invM    =   inv(M);

D_L     =   vessel.B([1,2,6],[1,2,6],34);
D_mo    =   0.1*vessel.B(:,:,34);


% for j=1:m_moor
%     for k=1:n_moor
%         r{j,k}=r_0((j-1)* 3* (n_moor-1)+ 3*(k-1)+ 1:(j-1) ...
   *3*(n_moor-1)+ 3*(k-1)+3);
%     end
% end
%%
%% Thruster Allocation
y_1     =   7.5;
y_2     =   -7.5;
x_3     =   -75;

vessel.T_thr   =   [1       1       0;
             0      0       1;
             y_1    y_2     x_3];
vessel.T_w_thr  =   inv(T_thr);
% T_w =   T'*inv(T*T'); %(Fossen 12.252)


%% %%%%%%%%%%%%%%%%%    Part 1  Observer     %%%%%%%%%%%%%%%%%%%
% ----------------    Passive Observer    ----------------------
b_0_PassObs         =   [0;0;0];
nu_0_PassObs        =   [0;0;0];
eta_0_PassObs       =   eta_0([1,2,6]);
epsilon_0_PassObs   =   [0;0;0;0;0;0];
eta_w_0_PassObs     =   epsilon_0_PassObs(1:3);
```

```
xi_w_0_PassObs     =    epsilon_0_PassObs(4:6);
D_L_PassObs=D_L+length(find(mooringLines==1))*D_mo([1,2,6],[1,2,6]);

invTb_PassObs       =   inv(diag([1000,1000,1000]));

zeta_ni      =   1;      % Fossen pp.317
lamda_i      =   0.1;    % Fossen pp.317 Relative damping ratio of ...
    wave spectrum

lambda       =   diag([lamda_i,lamda_i,lamda_i]);


omega_oi     =   [2*pi/7,2*pi/7,2*pi/7];
omega_ci     =   [1.1,1.1,1.1];

K_11         =   -2*(zeta_ni-lamda_i)*omega_ci(1)./omega_oi(1);
K_12         =   -2*(zeta_ni-lamda_i)*omega_ci(2)./omega_oi(2);
K_16         =   -2*(zeta_ni-lamda_i)*omega_ci(3)./omega_oi(3);

K_17         =   2*omega_oi(1)*(zeta_ni-lamda_i);
K_18         =   2*omega_oi(2)*(zeta_ni-lamda_i);
K_112        =   2*omega_oi(3)*(zeta_ni-lamda_i);

K_21         =   omega_ci(1);
K_22         =   omega_ci(2);
K_26         =   omega_ci(3);

OMEGA        =   diag(omega_oi);
DELTA        =   diag([1,1,1]);

Aw           =   [zeros(3)       eye(3);
                 -OMEGA*OMEGA    -2*DELTA*OMEGA];
Cw           =   [diag([0,0,0]),diag([0.05,0.05,0.01])];
T_b_PassObs      =   diag([1000,1000,1000]);

K1_PassObs       =   [diag([K_11,K_12,K_16]);diag([K_17,K_18,K_112])];
K2_PassObs       =   diag([K_21,K_22,K_26]);

K4_PassObs       =   diag([0.3,0.3,0.01]);
```

```
K3_PassObs      =   0.1*K4_PassObs;

%% Supervisor Switch
k_e_switch          =   1;          % gain of class K function
lambda_switch       =   0.1;        % constant non-negative ...
    forgetting factor
eta_0_SwitchSup     =   15;         % initial value of the switch ...
    function
k_h_switch          =   0.3;        % positive hysteresis constant


%% %%%%%%%%%%    Part 2  Controller   %%%%%%%%%%%%%%%
% -----------    PID Controller    ------------------
% 4 Mooring Lines
% mooringLines    =   [1,1,1,1];
Kp_PIDCtr_5     =   diag([1e3 1e3 0]);
Kd_PIDCtr_5     =   diag([0 0 0]);
Ki_PIDCtr_5     =   diag([2e2 1e3 0]);

%% Low-Pass Filter
K1_LF=1;
T_LF=0.7;

%% Simulation
breakTime=250;
tend=500;

% sim testSupervisor
sim tensionPositioning

% plottensionPositioning_SimulationI

plottensionPositioning

startDraw=150;
EndDraw=200;
plottensionPositioning_part
```

Listing C.2: The MATLAB code for plotting the figures.

```
clf('reset');

condition='No_Line';

%% sigma v.s. miu
time=TP_NED.time;

F2=figure(1);
subplot(2,1,1)
plot(time,TP_NED.signals(1).values(:,1),'g--','LineWidth',1)
hold on
plot(time,TP_NED.signals(1).values(:,3),'r-','LineWidth',1)
hold on
plot(time,TP_NED.signals(1).values(:,2),'k-','LineWidth',2)
% hold on
% plot(time,TP_NED_LP.signals(1).values(:,1),'c--','LineWidth',2)
% hold on
% plot(time,TP_NED_LP.signals(1).values(:,3),'r-','LineWidth',2)
ylabel('North(m)')
grid on;

subplot(2,1,2)
plot(time,TP_NED.signals(2).values(:,1),'g--','LineWidth',1)
hold on
plot(time,TP_NED.signals(2).values(:,3),'r-','LineWidth',1)
hold on
plot(time,TP_NED.signals(2).values(:,2),'k-','LineWidth',2)
% hold on
% plot(time,TP_NED_LP.signals(2).values(:,1),'c--','LineWidth',2)
% hold on
% plot(time,TP_NED_LP.signals(2).values(:,3),'r-','LineWidth',2)
title('East');
% h=legend('$y_{m}$','$\hat{y}_{m}$', 'y', '$\hat{y}_{moor,LF}$', ...
    '$\hat{y}_{moor,LF and NPO}$');
h=legend('$y_{m}$','$\hat{y}_{m}$','$y$')
set(h,'Interpreter','latex','location','best')
xlabel('time(s)')
ylabel('East(m)')
```

```
grid on;

print(F2,'-dpng',['.\image\position',condition,'.png'],'-r300')
```

Appendix D

Simulation Results and MATLAB Codes of Simultaneous Localization

Listing D.1: The MATLAB code for starting the Simulation.

```
clear all
clc

load('vessel.mat');         % vessel/hydrodynamic data structure
load('vesselABC.mat');      % fluid memory structure
load('T_H_table.mat');
load('curve_x.mat');            % Intepolate X
load('curve_T.mat');            % Intepolate T

% curve_x=T_H_table.X';
% curve_T=T_H_table.T(:,21);
% curve_H=T_H_table.H(:,21);

waveForce       =   1;      % wave motion On--1; Off--0
waveDriftForce  =   1;      % wave drift force On--1; Off--0
currentForce    =   1;      % c urrent On--1; Off--0
nu_wind     =   8;          % wind velocity
alpha_wind  =   pi/4;       % wind direction


```

```
mooringLines                =   ones(8,1);      % Mooring Lines ...
    On--1; Off--0
mooringLines1               =   ones(8,1);      % Mooring Lines after ...
    breaking On--1; Off--0
breakTime=1e10;                                 % Do not break.
mooringLinesMeasurement     =   ones(8,1);     % Mooring Lines ...
    Measurement On--1; Off--0
nu_0    =   [0,0,0,0,0,0]';

measurementNoises_on_off  =   1;                % measurement noise ...
    On--1; Off--0
TD_noises_on_off          =   1;                % Tension Measurement ...
    Noise
noise_x     =   0.005;    % variance of measurement noise in x, DGPS
noise_y     =   0.005;    % variance of measurement noise in y, DGPS
noise_psi   =   0.005/180*pi;   % variance of measurement noise in ...
    yaw, Compass, ...
    http://www.shipmotion.se/files/SMC%20IMU%20User%20Guide%20v22.pdf
noise_TD    =   0.0001;

M_RB    =   vessel.MRB([1,2,6],[1,2,6]);
M_A     =   vesselABC.MA([1,2,6],[1,2,6]);
M       =   M_RB+M_A;
invM    =   inv(M);

D_L     =   vessel.B([1,2,6],[1,2,6],34);
D_mo    =   0.1*vessel.B(:,:,34);

% Thruster Allocation
y_1     =   7.5;
y_2     =   -7.5;
x_3     =   -75;

T_thr   =   [1      1       0;
             0      0       1;
             y_1    y_2     x_3];
T_w_thr =   inv(T_thr);

b_0_PassObs         =   [0;0;0];
```

```
nu_0_PassObs        =   [0;0;0];
epsilon_0_PassObs   =   [0;0;0;0;0;0];
eta_w_0_PassObs     =   epsilon_0_PassObs(1:3);
xi_w_0_PassObs     =    epsilon_0_PassObs(4:6);
D_L_PassObs=D_L+length(find(mooringLines==1))*D_mo([1,2,6],[1,2,6]);

invTb_PassObs       =   inv(diag([1000,1000,1000]));

zeta_ni     =   1;      % Fossen pp.317
lamda_i     =   0.1;    % Fossen pp.317 Relative damping ratio of ...
   wave spectrum

lambda      =   diag([lamda_i,lamda_i,lamda_i]);

omega_oi    =   [2*pi/7,2*pi/7,2*pi/7];
omega_ci    =   [1.1,1.1,1.1];

K_11        =   -2*(zeta_ni-lamda_i)*omega_ci(1)./omega_oi(1);
K_12        =   -2*(zeta_ni-lamda_i)*omega_ci(2)./omega_oi(2);
K_16        =   -2*(zeta_ni-lamda_i)*omega_ci(3)./omega_oi(3);

K_17        =   2*omega_oi(1)*(zeta_ni-lamda_i);
K_18        =   2*omega_oi(2)*(zeta_ni-lamda_i);
K_112       =   2*omega_oi(3)*(zeta_ni-lamda_i);

K_21        =   omega_ci(1);
K_22        =   omega_ci(2);
K_26        =   omega_ci(3);

OMEGA       =   diag(omega_oi);
DELTA       =   diag([1,1,1]);

Aw          =   [zeros(3)       eye(3);
                -OMEGA*OMEGA    -2*DELTA*OMEGA];
Cw          =   [diag([0,0,0]),diag([0.05,0.05,0.01])];
T_b_PassObs     =   diag([1000,1000,1000]);

K1_PassObs      =   [diag([K_11,K_12,K_16]);diag([K_17,K_18,K_112])];
K2_PassObs      =   diag([K_21,K_22,K_26]);
```

```
K4_PassObs      =   diag([0.3,0.3,0.01]);
K3_PassObs      =   0.1*K4_PassObs;


Kp_PIDCtr     =   diag([0 0 8e6]);
Kd_PIDCtr     =   diag([0 0 1e5]);
Ki_PIDCtr     =   diag([0 0 3e6]);

%%
m   =   8;       % the number of the mooring line
n   =   10;      % the number of the TP in different time

%% uncertain anchor positions generate
% load('MLP.mat'); % best estimation
% MLP.r_0=MLP.r_0;

moorRadius=1960;
for i=1:m
    MLP.r_0(1:3,i)=[moorRadius*cos(2*pi/m*(i-1)); ...
       moorRadius*sin(2*pi/m*(i-1)); -1000];
end
save('.\data\MLP.mat','MLP');
moorUncertainRadius=100;  % moorUncertainRadius/2
for i=1:m
    MLPUncertain.r_0(1:2,i)= MLP.r_0(1:2,i)+ moorUncertainRadius* ...
       [rand-0.5;rand-0.5]; % the random position of the anchor is ...
       in a 1.5m circle of the initial position
end
save('.\data\MLPUncertain.mat','MLPUncertain');

%% Sensor network data generation process
tend=5000; MeasurementNoiseSampleTime=0.1; tstable=4500;
acceptedRegion  =   60;     % the diameter of the accepted region of ...
   the TP

    Hs_wave1    =   6*rand(n,1);       % significant wave height
    beta_wave1   =   pi*rand(n,1);  % mean wave direction
    nu_current1  =   0.3*rand(n,1);           % current velocity(m/s)
    beta_current1 =   beta_wave1/(2*pi)*360+90*(rand(n,1)-0.5); ...
                 % current direction(deg)

for j=1:n

    j

    Hs_wave     =   Hs_wave1(j);        % significant wave height
    beta_wave   =   beta_wave1(j);  % mean wave direction
    nu_current   =  nu_current1(j);           % current velocity(m/s)
    beta_current =  beta_current1(j);           % current direction(deg)


    eta_0           =   ...
       [Hs_wave*8*sin(beta_wave);Hs_wave*3*cos(beta_wave); 0; 0; 0; ...
       beta_wave];
    eta_0_PassObs   =   eta_0([1,2,6]);
    mean_GPS        =   zeros(3,1);
    noise_GPS       =   [0.05;0.05;0.00005];
    seed_GPS        =   40*rand(2,1)+1;
    mean_dist       =   zeros(m,1);
    noise_dist      =   1e6*ones(m,1);
    seed_dist       =   100*rand(m,1)+1;

    sim tensionPositioning

    for colll= 1: (length(GPS_real.signals.values)-tstable/ ...
       MeasurementNoiseSampleTime)
        real_position{j}(:, colll)   =  ...
           GPS_real.signals.values(colll+ tstable / ...
           MeasurementNoiseSampleTime, :)';
        noisy_position{j}(:, colll)  =   ...
           Obs.signals.values(colll+tstable/ ...
           MeasurementNoiseSampleTime,:)';
        real_tau_thr{j}(:, colll)    =   ...
           tau_thr.signals.values(colll+ ...
           tstable/MeasurementNoiseSampleTime, :)';
        real_eta_w{j}(:, colll)      =   ...
           eta_w.signals.values(colll+tstable/ ...
        MeasurementNoiseSampleTime,:)';
      noisy_position_includeWF{j}(:,colll) = ...
        noisy_position{j}(:,colll)+ real_eta_w{j}(:, colll);
      noisy_Tension{j}(:, colll)   =   ...
        tau_moor.signals.values(:,:,colll+tstable/ ...
        MeasurementNoiseSampleTime);
      for i=1:m
         noisy_dist{j}(i,colll)= interp1(curve_T, curve_x, ...
           noisy_Tension{j}(i,colll));
      end
      EKF_time(colll,1)=Obs.time(colll);
    end

    TP_position(1,j)=mean(noisy_position_includeWF{j} (1,:));
    TP_position(2,j)=mean(noisy_position_includeWF{j} (2,:));
    TP_position(3,j)=mean(noisy_position_includeWF{j} (3,:));

    Environment_para{j}.Hs_wave     =   Hs_wave;
    Environment_para{j}.beta_wave   =   beta_wave;
    Environment_para{j}.nu_current  =   nu_current;
    Environment_para{j}.beta_current=   beta_current;

    clear GPS_real Obs tau_thr tau_moor

    figure,
    subplot(3,1,1)
    plot(noisy_position_includeWF{j}(1,:),'g'); hold on; ...
      plot(real_position{j}(1,:),'r'); hold on; ...
      plot(TP_position(1,j)* ones (length(noisy_position{j} (1,:)), ...
      1), 'b');
    title('X'); ylabel('m'); legend('LF+WF','real','mean');
    subplot(3,1,2)
    plot(noisy_position_includeWF{j}(2,:),'g'); hold on; ...
      plot(real_position{j}(2,:),'r'); hold on; ...
      plot(TP_position(2,j)* ones(length (noisy_position{j} ...
      (2,:)),1),'b');
    title('Y'); ylabel('m'); legend('real','NPO','mean');
    subplot(3,1,3)
    plot(noisy_position_includeWF{j}(3,:)*360/(2*pi),'g'); hold on; ...
        plot(real_position{j}(3,:)* 360/(2*pi),'r'); hold on; ...
        plot(TP_position(3,j)* 360/(2*pi)* ...
        ones(length(noisy_position{j} (3,:)),1), 'b');
    title('phi'); ylabel('m'); legend('real', 'NPO', 'mean');
    end

save('.\data\real_position.mat','real_position');
save('.\data\noisy_position.mat','noisy_position');
save('.\data\real_tau_thr.mat','real_tau_thr');
save('.\data\noisy_Tension.mat','noisy_Tension');
save('.\data\noisy_dist.mat','noisy_dist');
save('.\data\TP_position.mat','TP_position');
save('.\data\Environment_para.mat','Environment_para');
save('.\data\real_eta_w.mat','real_eta_w');
save('.\data\noisy_position_includeWF.mat','noisy_position_includeWF');
save('.\data\EKF_time.mat','EKF_time');


F1=figure(1)
plot(MLPUncertain.r_0(1,:),MLPUncertain.r_0(2,:),'b*'); hold on
plot(TP_position(1,:),TP_position(2,:),'rs');
legend('the real poisition of the anchor','TP ...
    position','Location','best')
title('arrangment of the sensor network')
print(F1,'-dpng','.\image\arrangement.png','-r300')
```

Listing D.2: The MATLAB code of the EKF.

```
%%
m   =   8;      % the number of the mooring line
n   =   10;     % the number of the TP in different time

acceptedRegion  =   10;     % the diameter of the accepted region of ...
    the TP

clf(figure(2)); clf(figure(3));

```

```
h   =   MeasurementNoiseSampleTime;      % time step
i   =   1;

Phi         =   eye(2*(1+n));
Gamma       =   zeros(2*(1+n),2); Gamma(1,1)=1; Gamma(2,2)=1;


% initialize the EKF
Q       =   diag(10*[0.5;0.5]);
R       =   diag(1*ones(n,1));
x_bar   =   zeros(2*n+2,1);
x_bar(1:2,1)   =   MLP.r_0(1:2,i);

for j=1:n
    x_bar(2*j+1:2*j+2)   =   TP_position(1:2,j);
end
P_bar   =   diag([[1;1];ones(2*n,1)]);

for step=1:2000 %length(noisy_position{1})
    for j=1:n
        y(j,step)       =   noisy_dist{j}(i,step);
        y_hat(j,step)   =   sqrt((x_bar(1,step)- x_bar(2*j+1,step))^2+ ...
            (x_bar(2,step)- x_bar(2*j+2,step))^2);
    end
    y_tilde(:,step)=y(:,step)-y_hat(:,step);

    H   =   zeros(n,2*n+2);
    for j=1:n
        H(j,1:2)=[(x_bar(1,step)-x_bar(2*j+1,step))/ ...
            sqrt((x_bar(1,step)- x_bar(2*j+1,step))^2+ ...
            (x_bar(2,step)-x_bar(2*j+2,step))^2),(x_bar(2,step)- ...
            x_bar(2*j+2,step)) ...
            /sqrt((x_bar(1,step)-x_bar(2*j+1,step))^2+ ...
            (x_bar(2,step)-x_bar(2*j+2,step))^2)];
    end

    K   =   P_bar*H'*inv(H*P_bar*H'+R);
    x_hat(:,step) = x_bar(:,step)+K*y_tilde(:,step);
    P_hat = (eye(2*n+2)-K*H)*P_bar*(eye(2*n+2)-K*H)'+K*R*K';

    x_bar(1:2,step+1) =x_hat(1:2,step);
    for j=1:n
    x_bar(2*j+1:2*j+2,step+1) = ...
        noisy_position_includeWF{j}(1:2,step+1);% real_position
    end

    P_bar = Phi*P_hat*Phi'+Gamma*Q*Gamma';
end

clf('reset')
F2=figure(2)
subplot(2,1,1)
plot(x_hat(1,:),'g'); hold on;
plot(ones(1,length(x_hat(1,:)))*MLPUncertain.r_0(1,i),'r-'); hold on;
plot(ones(1,length(x_hat(1,:)))*MLP.r_0(1,i),'b--'); hold on;
subplot(2,1,2)
plot(x_hat(2,:),'g'); hold on
plot(ones(1,length(x_hat(2,:)))*MLPUncertain.r_0(2,i),'r-'); hold on;
plot(ones(1,length(x_hat(2,:)))*MLP.r_0(2,i),'b--'); hold on;
legend('EKF','real','initial estimation','Localcation','best')
print(F2,'-dpng','.\image\EKF_anchor.png','-r300')

F3=figure(3)
title('TP1')
subplot(2,1,1)
plot(real_position{1}(1,1:length(x_hat)),'r-'); hold on;
plot(noisy_position{1}(1,1:length(x_hat)),'b--'); hold on;
plot(x_hat(3,:),'g','LineWidth',2); hold on;


subplot(2,1,2)
plot(x_hat(4,:),'g-','LineWidth',2); hold on
plot(noisy_position{1}(2,1:length(x_hat)),'b--'); hold on;
plot(real_position{1}(2,1:length(x_hat)),'r-'); hold on;
legend('EKF','NPO','real','Localcation','best')
print(F3,'-dpng','.\image\EKF_GPS.png','-r300')

clear x_hat x_bar Q R P_bar
```

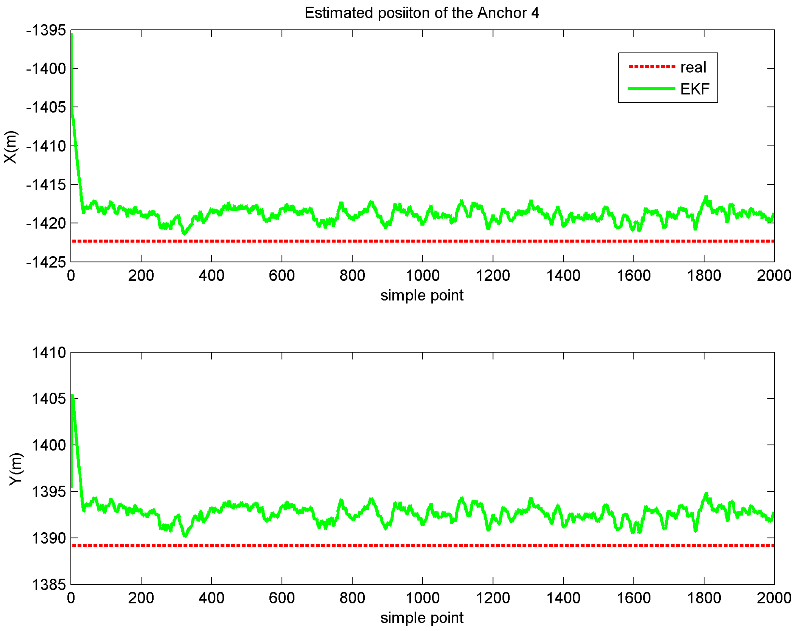
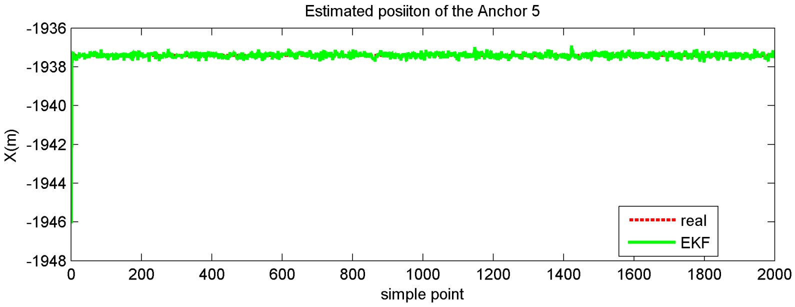


Figure D.1: Anchor 4 position estimation with LF and WF.

D.1 Results of Simulation VI



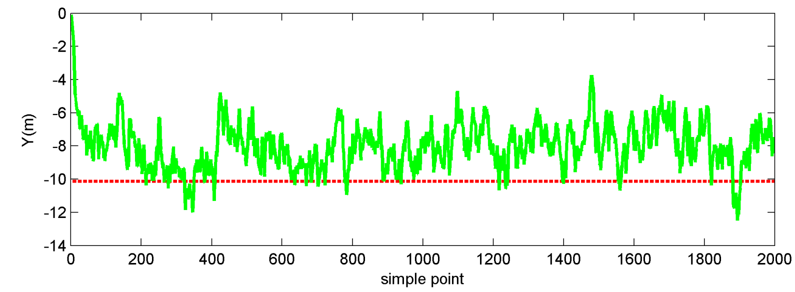


Figure D.2: Anchor 5 position estimation with LF and WF.

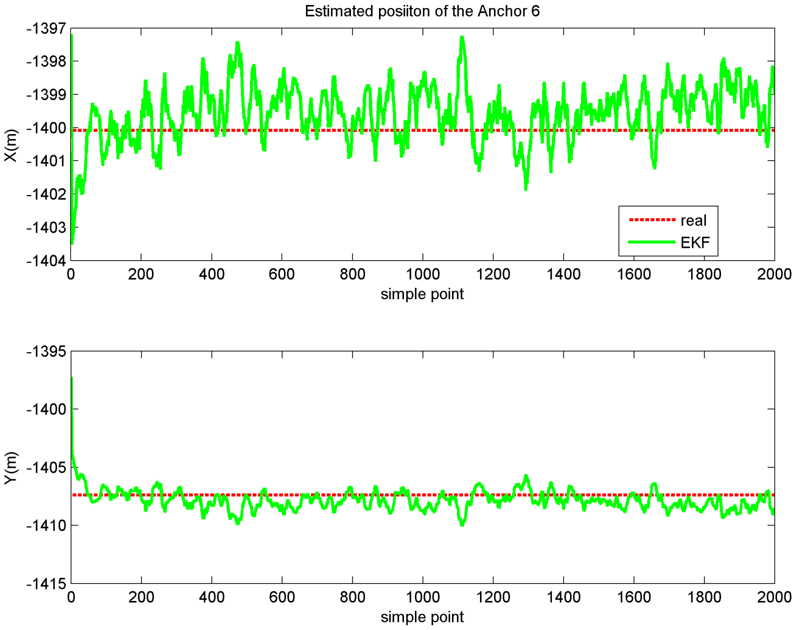


Figure D.3: Anchor 6 position estimation with LF and WF.

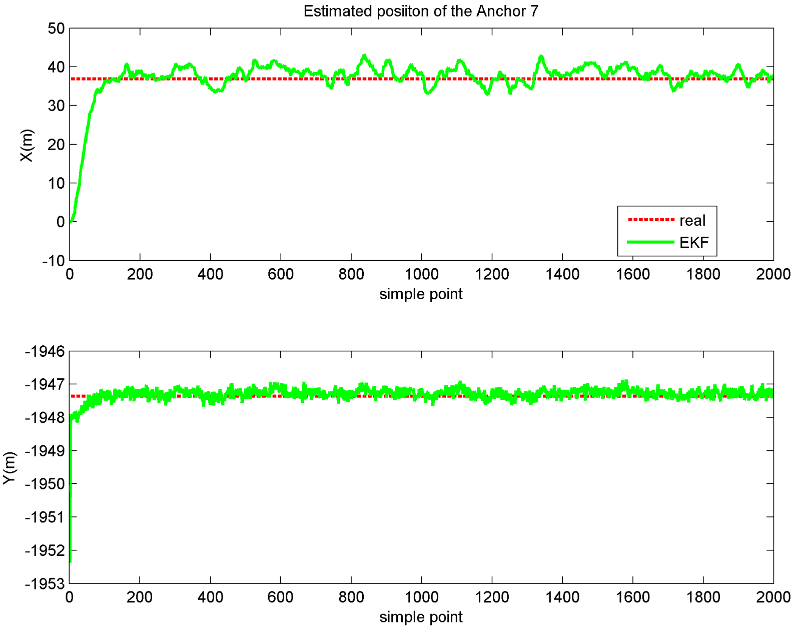


Figure D.4: Anchor 7 position estimation with LF and WF.

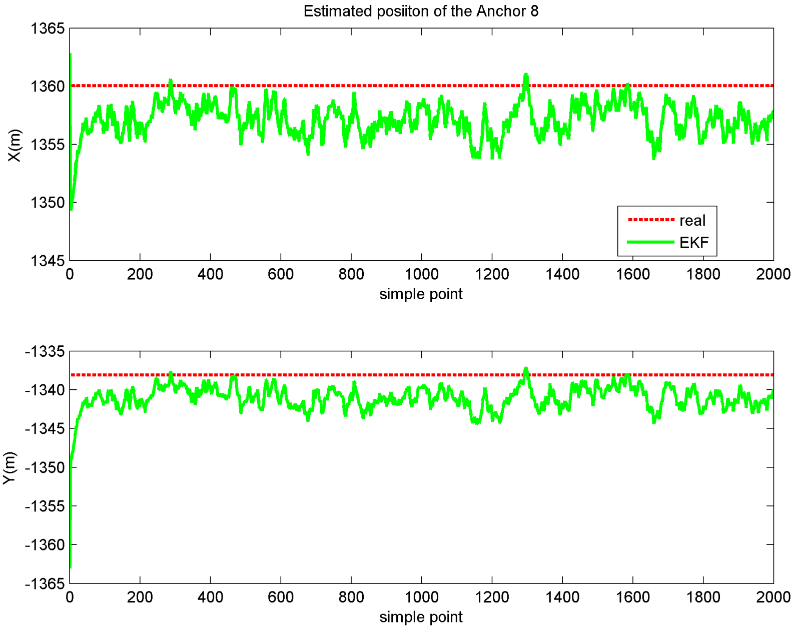


Figure D.5: Anchor 8 position estimation with LF and WF.

Appendix E

Matlab Codes of Second Order Cone Programming

Listing E.1: The MATLAB code for SCOP.

```matlab
clear all
clc
clf('reset')
%%
load('curve_T.mat');
load('curve_x.mat');
load('T_H_table.mat');
load('MLP.mat'); % best estimation
% load('MLPUncertain.mat'); % actual value

MLP.r_0=MLP.r_0*0.02;

%%
tend=1100; TD_noises_on_off=1; noise_mean=zeros(8,1); ...
    noise_TD=0.005*ones(8,1); MeasurementNoiseSampleTime=0.1;
noise_seed=[31;76;396;93;19;74;76;93];
noise_variance=1e6*[1;1.2;0.95;1.2;1.23;1.02;0.92;0.97];

% uncertain anchor positions
for anchor=1:length(MLP.r_0(1,:))
```

```
MLPUncertain.r_0(1:2,anchor)=MLP.r_0(1:2,anchor)+5*[rand-0.5;rand-0.5]; ...
    %%%%%%%%%%%
end
save('.\uncertainAnchorWithNoise\MLPUncertain.mat','MLPUncertain');


noise_X_test=[-10,10,0,0,0,0];

%%
% sim uncertainAnchorWithNoise
tend=100; TD_noises_on_off=1; noise_mean=zeros(8,1); ...
    noise_TD=0.01*ones(8,1); MeasurementNoiseSampleTime=0.1; ...
    noise_seed=[31;76;396;93;19;74;76;93];noise_variance=ones(1,8); ...
    %*[1;1.2;0.95;1.2;1.23;1.02;0.92;0.97];
Amp=100;
sim testtesttest

%%
pointStep=10;
N_line=8;  % i
numberTP_once=30; % j
numberTP=1;

PHI=1e6*ones(N_line,1);
PHI_2=PHI.^(-1/2);

% figure(2)
for k=1:numberTP
for j=1:numberTP_once
    for i=1:N_line
        d_ij(i,j)=d.signals(1).values(i,1,(j-1)*pointStep+1);
        g_ij(i,j)=1/1;
        a(:,i)=GPS.signals.values(:,:,(i-1)*pointStep+1)';
    end
end

cvx_begin
variable x_i(2,N_line)
variable x_j(2,numberTP_once)
```

```

u=[];
variable s_i(1,N_line)
variable t_ij(N_line,numberTP_once)
for i=1:N_line
   u =[u,t_ij(i,:)];
end
u=[u,s_i];

variable q_ij(N_line,numberTP_once)

variable v(1,1)

minimize(v)     % object function

subject to
    norm(u)≤v;                                                         % 1)
    for j=1:numberTP_once
        for i=1:N_line
            g_ij(i,j)*abs(q_ij(i,j)-d_ij(i,j))≤t_ij(i,j);          % 2)
            norm(x_i(1:2,i)-x_j(1:2,j))≤q_ij(i,j);               % 4)
        end
    end
       for i=1:N_line
            norm(PHI_2(i)*(a(1:2,i)-x_i(1:2,i)))≤s_i(i);     % 3)
       end
cvx_end
    Anchor{k}=x_i;
    Vessel{k}=x_j;
%      for i=1:N_line
%      plot(x_i(1,:),x_i(2,:),'g.'); hold on
%      end
end

% Mean
for k=1:numberTP
    for i=1:N_line
    AnchorTot{i}(:,k)=Anchor{k}(:,i);
    end
```

```
end
for i=1:N_line
    x_i_mean(1,i)=mean(AnchorTot{i}(1,:));
    x_i_mean(2,i)=mean(AnchorTot{i}(2,:))
end

F1=figure(1)
plot(MLP.r_0(1,:),MLP.r_0(2,:),'r*'); hold on
plot(MLPUncertain.r_0(1,:),MLPUncertain.r_0(2,:),'go'); hold on
plot(x_i_mean(1,:),x_i_mean(2,:),'b+');
plot(a(1,:),a(2,:),'rs')
legend('best estimation','real anchor','SOCP anchor','TP', ...
    'Location', 'best')
print(F1,'-dpng','.\SOCP_large\performance.png','-r300')
```

Appendix F

MATLAB Codes of Comparison of Analytical Solution for TTR

F.1 Simulation II.1

Listing F.1: The MATLAB code for calculate the deflection.

```
function    [x,z,AngB,AngT]   =   ...
    beamDeflection(uc,T_top,meshNo,l,Cd,rhow,rhos,Do,Di,E)
% This program is applied to calculate the deflection of a top ...
    tensioned riser
% based on the mechanical structure. The loads is constant current ...
    speed. The
% result is in 2D frame. Please transform it into 3D corordinate ...
    after the
% calculation. At the bottom end,z=0. The position direction is upward.

% Author: Zhengru Ren
% Date: 15/06/2015, NTNU, Trondheim
% Inputs:
% uc:   current velocity [m/s]
% T_top:top tension [N]
% meshNo:number of point in the curve
% l:    length of the riser [m]
% Cd:   drag non-dimesional coefficient
% rhow: water density [kg/s^3]
% rhos: water density [kg/s^3]
% Do:   riser outer diagmeter [m]
% Di:   riser outer diagmeter [m]
% E:    Young's moduls[Pa]

% Outputs:
% x:    Displacement perpendicular to the riser [m]
% z:    Displacement along the riser [m]
% AngB: Bottom end angle [deg]
% AngT: Top end angle [deg]

A   =   pi/4*(Do^2-Di^2);
I   =   pi/2*((Do/2)^4-(Di/2)^4);
g   =   9.81;       % gravity acceleration [m/s^2]
W   =   (rhos-rhow)*A*g;
T   =   T_top+W*l/2;
k   =   sqrt(T/(E*I));
u   =   k*l/2;
q   =   0.5*rhow*Cd*Do*1*uc*uc;    %   current load [N]

z   =   0:l/meshNo:l;

for ii=1:length(z)
    x(ii)   =   q*l^4/(E*I*(2*u)^4)*(cosh(k*(z(ii)-l/2)) /cosh(u)-1 + ...
        (u^2-k^2*(z(ii)-l/2)^2)/2);
end
a=2;
AngB=atand((x(a)-x(1))/(z(a)-z(1)));
AngT=atand((x(end)-x(end-a))/(z(end)-z(end-a)));
```

Listing F.2: The MATLAB code for test different current velocity.

```
% This program is to compare the FIFLEX calculation and the thretical
% deflection function. The vessel is fixed at (0,0,0), and the current
% direction is 0 deg. The current speed is 0.3,0.6,0.9,1.2,1.5 m/s.

% Author: Zhengru Ren
% Date: 08.06.2015
```

```

clear

load('D:\GraduateThesisNTNU\reflixData\X0ChangeCurrent\X0V3D.mat')
load('D:\GraduateThesisNTNU\reflixData\X0ChangeCurrent\X0V3T.mat')
load('D:\GraduateThesisNTNU\reflixData\X0ChangeCurrent\X0V6D.mat')
load('D:\GraduateThesisNTNU\reflixData\X0ChangeCurrent\X0V6T.mat')
load('D:\GraduateThesisNTNU\reflixData\X0ChangeCurrent\X0V9D.mat')
load('D:\GraduateThesisNTNU\reflixData\X0ChangeCurrent\X0V9T.mat')
load('D:\GraduateThesisNTNU\reflixData\X0ChangeCurrent\X0V12D.mat')
load('D:\GraduateThesisNTNU\reflixData\X0ChangeCurrent\X0V12T.mat')
load('D:\GraduateThesisNTNU\reflixData\X0ChangeCurrent\X0V15D.mat')
load('D:\GraduateThesisNTNU\reflixData\X0ChangeCurrent\X0V15T.mat')

%%
clf('reset')
F1=figure(1)
plot(X0V3D(:,2),X0V3D(:,4),'b','LineWidth',2); hold on
plot(X0V6D(:,2),X0V6D(:,4),'b','LineWidth',2); hold on
plot(X0V9D(:,2),X0V9D(:,4),'b','LineWidth',2); hold on
plot(X0V12D(:,2),X0V12D(:,4),'b','LineWidth',2); hold on
plot(X0V15D(:,2),X0V15D(:,4),'b','LineWidth',2); hold on

rhow=   1025;       %   water density [kg/s^3]
rhos=   7850;       %   water density [kg/s^3]
l   =   200;        %   length of the riser [m]
Cd  =   1;          %   drag non-dimesional coefficient
Do  =   0.3;        %   riser outer diagmeter [m]
Di  =   0.15;       %   riser outer diagmeter [m]
E   =   206e9;
g   =   9.81;
T_top   =   2045e3;
meshNo  =   200;

a   =   5;

T_top=X0V3T(1,2)*1e3;
uc(1) = 0.3;
```

```
[x,z,AngB(1),AngT(1)]  =  ...
    beamDeflection(uc(1),T_top,meshNo,l,Cd,rhow,rhos,Do,Di,E);
plot(x,z-l,'k--','LineWidth',1.5); hold on
AngB_riflex(1)=atand((X0V3D(a,2)-X0V3D(1,2))/(X0V3D(a,4)-X0V3D(1,4)));
AngT_riflex(1)=atand((X0V3D(end,2)- X0V3D(end-a,2))/ (X0V3D(end,4)- ...
    X0V3D(end-a,4)));


T_top=X0V6T(1,2)*1e3;
uc(2) = 0.6;
[x,z,AngB(2),AngT(2)]  =  ...
    beamDeflection(uc(2),T_top,meshNo,l,Cd,rhow,rhos,Do,Di,E);
plot(x,z-l,'k--','LineWidth',1.5); hold on
AngB_riflex(2)=atand((X0V6D(a,2)-X0V6D(1,2))/(X0V6D(a,4)-X0V6D(1,4)));
AngT_riflex(2)=atand((X0V6D(end,2)- X0V6D(end-a,2))/ (X0V6D(end,4)- ...
    X0V6D(end-a,4)));

T_top=X0V9T(1,2)*1e3;
uc(3) = 0.9;
[x,z,AngB(3),AngT(3)]  =  ...
    beamDeflection(uc(3),T_top,meshNo,l,Cd,rhow,rhos,Do,Di,E);
plot(x,z-l,'k--','LineWidth',1.5); hold on
AngB_riflex(3)=atand((X0V9D(a,2)-X0V9D(1,2))/(X0V9D(a,4)-X0V9D(1,4)));
AngT_riflex(3)=atand((X0V9D(end,2)-X0V9D(end-a,2)) /(X0V9D(end,4)- ...
    X0V9D(end-a,4)));

T_top=X0V12T(1,2)*1e3;
uc(4) = 1.2;
[x,z,AngB(4),AngT(4)]  =  ...
    beamDeflection(uc(4),T_top,meshNo,l,Cd,rhow,rhos,Do,Di,E);
plot(x,z-l,'k--','LineWidth',1.5); hold on
AngB_riflex(4)=atand((X0V12D(a,2)-X0V12D(1,2)) /(X0V12D(a,4)- ...
    X0V12D(1,4)));
AngT_riflex(4)=atand((X0V12D(end,2)-X0V12D(end-a,2)) /(X0V12D(end,4)- ...
    X0V12D(end-a,4)));

T_top=X0V15T(1,2)*1e3;
uc(5) = 1.5;
```

```
[x,z,AngB(5),AngT(5)]   =   ...
    beamDeflection(uc(5),T_top,meshNo,l,Cd,rhow,rhos,Do,Di,E);
plot(x,z-l,'k--','LineWidth',1.5); hold on
AngB_riflex(5)=atand((X0V15D(a,2)-X0V15D(1,2))/(X0V15D(a,4)-X0V15D(1,4)));
AngT_riflex(5)=atand((X0V15D(end,2)-X0V15D(end-a,2))/ (X0V15D(end,4)- ...
    X0V15D(end-a,4)));

ylim([-1,0])
xlabel('Horizontal position (m)');
ylabel('Vertical position (m)');
grid on;
grid minor;
print(F1,'-dpng','.\X0ChangeCurrent\currentVelDisp.png','-r300')

F2=figure(2)
subplot(2,1,1)
plot(uc,AngB,'k--o','LineWidth',1.5); hold on;
plot(uc,AngB_riflex,'b-s','LineWidth',1.5); hold on;
legend('Analytical','RIFLEX','Location','Best')
xlabel('Current speed (m/s)');
ylabel('Bottom end angle (\deg)');
grid on;
grid minor;
subplot(2,1,2)
plot(uc,AngT,'k--o','LineWidth',1.5); hold on;
plot(uc,AngT_riflex,'b-s','LineWidth',1.5); hold on;
% legend('Analytical','RIFLEX','Location','Best')
xlabel('Current speed (m/s)');
ylabel('Top end angle (\deg)');
grid on;
grid minor;
print(F2,'-dpng','.\X0ChangeCurrent\CurrentVelAng.png','-r300')
```

F.2 Simulation II.2

Listing F.3: The MATLAB code for test different top tension.

```
% This program is to compare the FIFLEX calculation and the thretical
% deflection function. The vessel is fixed at (0,0,0), the current
% speed is 1 m/s, and the current direction is 0 deg. The top  ...
    tension is
% changed due to the initial stress.

% Author: Zhengru Ren
% Date: 08.06.2015

clear

load('D:\GraduateThesisNTNU\reflixData\X0ChangeT\X0V1T1D.mat')
load('D:\GraduateThesisNTNU\reflixData\X0ChangeT\X0V1T1T.mat')
load('D:\GraduateThesisNTNU\reflixData\X0ChangeT\X0V1T2D.mat')
load('D:\GraduateThesisNTNU\reflixData\X0ChangeT\X0V1T2T.mat')
load('D:\GraduateThesisNTNU\reflixData\X0ChangeT\X0V1T3D.mat')
load('D:\GraduateThesisNTNU\reflixData\X0ChangeT\X0V1T3T.mat')
load('D:\GraduateThesisNTNU\reflixData\X0ChangeT\X0V1T4D.mat')
load('D:\GraduateThesisNTNU\reflixData\X0ChangeT\X0V1T4T.mat')
load('D:\GraduateThesisNTNU\reflixData\X0ChangeT\X0V1T5D.mat')
load('D:\GraduateThesisNTNU\reflixData\X0ChangeT\X0V1T5T.mat')
load('D:\GraduateThesisNTNU\reflixData\X0ChangeT\X0V1T6D.mat')
load('D:\GraduateThesisNTNU\reflixData\X0ChangeT\X0V1T6T.mat')
%%
clf('reset')
F1=figure(1)
plot(X0V1T1D(:,2),X0V1T1D(:,4),'b','LineWidth',2); hold on
plot(X0V1T2D(:,2),X0V1T2D(:,4),'b','LineWidth',2); hold on
plot(X0V1T3D(:,2),X0V1T3D(:,4),'b','LineWidth',2); hold on
plot(X0V1T4D(:,2),X0V1T4D(:,4),'b','LineWidth',2); hold on
plot(X0V1T5D(:,2),X0V1T5D(:,4),'b','LineWidth',2); hold on
plot(X0V1T6D(:,2),X0V1T6D(:,4),'b','LineWidth',2); hold on

uc   =   1;          %   current velocity [m/s]
rhow=   1025;        %   water density [kg/s^3]
rhos=   7850;        %   water density [kg/s^3]
l    =   200;        %   length of the riser [m]
Cd   =   1;          %   drag non-dimesional coefficient
```

```
Do   =   0.3;        %   riser outer diagmeter [m]
Di   =   0.15;       %   riser outer diagmeter [m]
E    =   206e9;
g    =   9.81;
T_top    =   2045e3;
meshNo   =   200;

a    =   5;

T_top(1)=X0V1T1T(1,2)*1e3;
[x,z,AngB(1),AngT(1)]    =    ...
    beamDeflection(uc,T_top(1),meshNo,l,Cd,rhow,rhos,Do,Di,E);
plot(x,z-l,'k--','LineWidth',1.5); hold on
AngB_riflex(1)=atand((X0V1T1D(a,2)- X0V1T1D(1,2))/(X0V1T1D(a,4)- ...
    X0V1T1D(1,4)));
AngT_riflex(1)=atand((X0V1T1D(end,2)- ...
    X0V1T1D(end-a,2))/(X0V1T1D(end,4)- X0V1T1D(end-a,4)));


T_top(2)=X0V1T2T(1,2)*1e3;
[x,z,AngB(2),AngT(2)]    =    ...
    beamDeflection(uc,T_top(2),meshNo,l,Cd,rhow,rhos,Do,Di,E);
plot(x,z-l,'k--','LineWidth',1.5); hold on
AngB_riflex(2)=atand((X0V1T2D(a,2)- ...
    X0V1T2D(1,2))/(X0V1T2D(a,4)-X0V1T2D(1,4)));
AngT_riflex(2)=atand((X0V1T2D(end,2)- X0V1T2D(end-a,2))/ ...
    (X0V1T2D(end,4)-X0V1T2D(end-a,4)));

T_top(3)=X0V1T3T(1,2)*1e3;
[x,z,AngB(3),AngT(3)]    =    ...
    beamDeflection(uc,T_top(3),meshNo,l,Cd,rhow,rhos,Do,Di,E);
plot(x,z-l,'k--','LineWidth',1.5); hold on
AngB_riflex(3)=atand((X0V1T3D(a,2)- X0V1T3D(1,2))/ ...
    (X0V1T3D(a,4)-X0V1T3D(1,4)));
AngT_riflex(3)=atand((X0V1T3D(end,2)- ...
    X0V1T3D(end-a,2))/(X0V1T3D(end,4) -X0V1T3D(end-a,4)));

T_top(4)=X0V1T4T(1,2)*1e3;
```

```
[x,z,AngB(4),AngT(4)]   =   ...
    beamDeflection(uc,T_top(4),meshNo,l,Cd,rhow,rhos,Do,Di,E);
plot(x,z-l,'k--','LineWidth',1.5); hold on
AngB_riflex(4)=atand((X0V1T4D(a,2)-X0V1T4D(1,2)) /(X0V1T4D(a,4) ...
    -X0V1T4D(1,4)));
AngT_riflex(4)=atand((X0V1T4D(end,2)-X0V1T4D(end-a,2)) ...
    /(X0V1T4D(end,4) -X0V1T4D(end-a,4)));

T_top(5)=X0V1T5T(1,2)*1e3;
[x,z,AngB(5),AngT(5)]   =   ...
    beamDeflection(uc,T_top(5),meshNo,l,Cd,rhow,rhos,Do,Di,E);
plot(x,z-l,'k--','LineWidth',1.5); hold on
AngB_riflex(5)=atand((X0V1T5D(a,2)-X0V1T5D(1,2))/ (X0V1T5D(a,4)- ...
    X0V1T5D(1,4)));
AngT_riflex(5)=atand((X0V1T5D(end,2)-X0V1T5D(end-a,2))/ ...
    (X0V1T5D(end,4)- X0V1T5D(end-a,4)));

T_top(6)=X0V1T6T(1,2)*1e3;
[x,z,AngB(6),AngT(6)]   =   ...
    beamDeflection(uc,T_top(6),meshNo,l,Cd,rhow,rhos,Do,Di,E);
plot(x,z-l,'k--','LineWidth',1.5); hold on
AngB_riflex(6)=atand((X0V1T6D(a,2)-X0V1T6D(1,2)) ...
    /(X0V1T6D(a,4)-X0V1T6D(1,4)));
AngT_riflex(6)=atand((X0V1T6D(end,2)- X0V1T6D(end-a,2))/ ...
    (X0V1T6D(end,4)-X0V1T6D(end-a,4)));

ylim([-l,0])
xlabel('Horizontal position (m)');
ylabel('Vertical position (m)');
grid on;
grid minor;
print(F1,'-dpng','.\X0ChangeT\TDisp.png','-r300')

F2=figure(2)
subplot(2,1,1)
plot(T_top/1000,AngB,'k--o','LineWidth',1.5); hold on;
plot(T_top/1000,AngB_riflex,'b-s','LineWidth',1.5); hold on;
legend('Analytical','RIFLEX','Location','Best')
xlabel('Top tension (kN)');
```

```matlab
ylabel('Bottom end Angle (\deg)');
grid on;
grid minor;
subplot(2,1,2)
plot(T_top/1000,AngT,'k--o','LineWidth',1.5); hold on;
plot(T_top/1000,AngT_riflex,'b-s','LineWidth',1.5); hold on;
% legend('Analytical','RIFLEX','Location','Best')
xlabel('Top tension (kN)');
ylabel('Top end angle (\deg)');
grid on;
grid minor;
print(F2,'-dpng','.\X0ChangeT\TAng.png','-r300')
```

F.3 Simulation II.3

Listing F.4: The MATLAB code for test different current direction.

```matlab
% This program is to compare the FIFLEX calculation and the thretical
% deflection function. The vessel is fixed at (0,0,3), and the current
% speed is 1 m/s. The current direction is 0,30,60,90 deg.

% Author: Zhengru Ren
% Date: 08.06.2015

clear

load('D:\GraduateThesisNTNU\reflixData\X3ChangeDir\X3V10D.mat')
load('D:\GraduateThesisNTNU\reflixData\X3ChangeDir\X3V10T.mat')
load('D:\GraduateThesisNTNU\reflixData\X3ChangeDir\X3V130D.mat')
load('D:\GraduateThesisNTNU\reflixData\X3ChangeDir\X3V130T.mat')
load('D:\GraduateThesisNTNU\reflixData\X3ChangeDir\X3V160D.mat')
load('D:\GraduateThesisNTNU\reflixData\X3ChangeDir\X3V160T.mat')
load('D:\GraduateThesisNTNU\reflixData\X3ChangeDir\X3V190D.mat')
load('D:\GraduateThesisNTNU\reflixData\X3ChangeDir\X3V190T.mat')
%%
clf('reset')
F1=figure(1)
```

```
subplot(1,2,1)
plot(X3V10D(:,2),X3V10D(:,4),'b','LineWidth',2); hold on
plot(X3V130D(:,2),X3V130D(:,4),'b','LineWidth',2); hold on
plot(X3V160D(:,2),X3V160D(:,4),'b','LineWidth',2); hold on
plot(X3V190D(:,2),X3V190D(:,4),'b','LineWidth',2); hold on
% plot(X0V15D(:,2),X0V15D(:,4),'b','LineWidth',2); hold on

subplot(1,2,2)
plot(X3V10D(:,3),X3V10D(:,4),'b','LineWidth',2); hold on
plot(X3V130D(:,3),X3V130D(:,4),'b','LineWidth',2); hold on
plot(X3V160D(:,3),X3V160D(:,4),'b','LineWidth',2); hold on
plot(X3V190D(:,3),X3V190D(:,4),'b','LineWidth',2); hold on

uc  =   1;          %   current velocity [m/s]
rhow=   1025;       %   water density [kg/s^3]
rhos=   7850;       %   water density [kg/s^3]
l   =   200;        %   length of the riser [m]
Cd  =   1;          %   drag non-dimesional coefficient
Do  =   0.3;        %   riser outer diagmeter [m]
Di  =   0.15;       %   riser outer diagmeter [m]
E   =   206e9;
g   =   9.81;
T_top   =   2045e3;
meshNo  =   200;
X_top   =   3;

a=3;

x1=0:X_top/meshNo:X_top;

CDir(1)=0;
T_top=X3V10T(1,2)*1e3;
[x,z]   =   beamDeflection1(uc,T_top,meshNo,l,Cd,rhow,rhos,Do,Di,E);
AngB(1)=atand (sqrt((x(a)*cosd(CDir(1))+x1(a) - ...
    x(1)*cosd(CDir(1))-x1(1))^2 + (x(a)*sind(CDir(1)) - ...
    x(1)*sind(CDir(1)))^2) / (z(a)-z(1)));
AngT(1)=atand (sqrt((x(end)*cosd(CDir(1))+x1(end) - ...
    x(end-a)*cosd(CDir(1))-x1(end-a))^2 + (x(end)*sind(CDir(1)) - ...
    x(end-a)*sind(CDir(1)))^2) / (z(end)-z(end-a)));
```

```
subplot(1,2,1)
plot(x*cosd(CDir(1))+x1,z-l,'k--','LineWidth',1.5); hold on;
subplot(1,2,2)
plot(x*sind(CDir(1)),z-l,'k--','LineWidth',1.5); hold on;
AngB_riflex(1)=atand (sqrt((X3V10D(a,2)-X3V10D(1,2))^2+ (X3V10D(a,3)- ...
    X3V10D(1,3))^2)/ (X3V10D(a,4)-X3V10D(1,4)));
AngT_riflex(1)=atand (sqrt((X3V10D(end,2)-X3V10D(end-a,2))^2+ ...
    (X3V10D(end,3)-X3V10D(end-a,3))^2)/ (X3V10D(end,4)-X3V10D(end-a,4)));

CDir(2)=30;
T_top=X3V130T(1,2)*1e3;
[x,z]  =  beamDeflection1(uc,T_top,meshNo,l,Cd,rhow,rhos,Do,Di,E);
AngB(2)=atand (sqrt((x(a)*cosd(CDir(2))+x1(a) - ...
    x(1)*cosd(CDir(2))-x1(1))^2 + (x(a)*sind(CDir(2)) - ...
    x(1)*sind(CDir(2)))^2) / (z(a)-z(1)));
AngT(2)=atand (sqrt((x(end)*cosd(CDir(2))+x1(end) - ...
    x(end-a)*cosd(CDir(2))-x1(end-a))^2 + (x(end)*sind(CDir(2)) - ...
    x(end-a)*sind(CDir(2)))^2) / (z(end)-z(end-a)));
subplot(1,2,1)
plot(x*cosd(CDir(2))+x1,z-l,'k--','LineWidth',1.5); hold on
subplot(1,2,2)
plot(x*sind(CDir(2)),z-l,'k--','LineWidth',1.5); hold on
AngB_riflex(2)=atand (sqrt((X3V130D(a,2)-X3V130D(1,2))^2+ ...
    (X3V130D(a,3)-X3V130D(1,3))^2)/(X3V130D(a,4)-X3V130D(1,4)));
AngT_riflex(2)=atand (sqrt((X3V130D(end,2)-X3V130D(end-a,2))^2+ ...
    (X3V130D(end,3)-X3V130D(end-a,3))^2)/ ...
    (X3V130D(end,4)-X3V130D(end-a,4)));

CDir(3)=60;
T_top=X3V160T(1,2)*1e3;
[x,z]  =  beamDeflection1(uc,T_top,meshNo,l,Cd,rhow,rhos,Do,Di,E);
AngB(3)=atand (sqrt((x(a)*cosd(CDir(3))+x1(a) - ...
    x(1)*cosd(CDir(3))-x1(1))^2 + (x(a)*sind(CDir(3)) - ...
    x(1)*sind(CDir(3)))^2) / (z(a)-z(1)));
AngT(3)=atand (sqrt((x(end)*cosd(CDir(3))+x1(end) - ...
    x(end-a)*cosd(CDir(3))-x1(end-a))^2 + (x(end)*sind(CDir(3)) - ...
    x(end-a)*sind(CDir(3)))^2) / (z(end)-z(end-a)));
subplot(1,2,1)
plot(x*cosd(CDir(3))+x1,z-l,'k--','LineWidth',1.5); hold on
```

```
subplot(1,2,2)
plot(x*sind(CDir(3)),z-l,'k--','LineWidth',1.5); hold on
AngB_riflex(3)=atand (sqrt((X3V160D(a,2)-X3V160D(1,2))^2+ ...
    (X3V160D(a,3)- X3V160D(1,3))^2)/(X3V160D(a,4)-X3V160D(1,4)));
AngT_riflex(3)=atand (sqrt((X3V160D(end,2)-X3V160D(end-a,2))^2+ ...
    (X3V160D(end,3)-X3V160D(end-a,3))^2)/ ...
    (X3V160D(end,4)-X3V160D(end-a,4)));

CDir(4)=90;
T_top=X3V190T(1,2)*1e3;
[x,z]   =   beamDeflection1(uc,T_top,meshNo,l,Cd,rhow,rhos,Do,Di,E);
AngB(4)=atand (sqrt((x(a)*cosd(CDir(4))+x1(a) - ...
    x(1)*cosd(CDir(4))-x1(1))^2 + (x(a)*sind(CDir(4)) - ...
    x(1)*sind(CDir(4)))^2) / (z(a)-z(1)));
AngT(4)=atand (sqrt((x(end)*cosd(CDir(4))+x1(end) - ...
    x(end-a)*cosd(CDir(4))-x1(end-a))^2 + (x(end)*sind(CDir(4)) - ...
    x(end-a)*sind(CDir(4)))^2) / (z(end)-z(end-a)));
subplot(1,2,1)
plot(x*cosd(CDir(4))+x1,z-l,'k--','LineWidth',1.5); hold on
subplot(1,2,2)
plot(x*sind(CDir(4)),z-l,'k--','LineWidth',1.5); hold on
AngB_riflex(4)=atand (sqrt((X3V190D(a,2)-X3V190D(1,2))^2+ ...
    (X3V190D(a,3)- X3V190D(1,3))^2)/(X3V190D(a,4)-X3V190D(1,4)));
AngT_riflex(4)=atand (sqrt((X3V190D(end,2)-X3V190D(end-a,2))^2+ ...
    (X3V190D(end,3)-X3V190D(end-a,3))^2)/ ...
    (X3V190D(end,4)-X3V190D(end-a,4)));

subplot(1,2,1)
ylim([-1,0])
xlabel('Horizontal position in X (m)');
ylabel('Vertical position (m)');
grid on;
grid minor;
subplot(1,2,2)
ylim([-1,0])
xlabel('Horizontal position in Y (m)');
ylabel('Vertical position (m)');
grid on;
grid minor;
```

```
print(F1,'-dpng','.\X3ChangeDir\currentDirDisp.png','-r300')

F2=figure(2)
subplot(2,1,1)
plot(CDir,AngB,'k--o','LineWidth',1.5); hold on;
plot(CDir,AngB_riflex,'b-s','LineWidth',1.5); hold on;
legend('Analytical','RIFLEX','Location','Best')
xlabel('Current direction (deg)');
ylabel('Bottom End Angle (\deg)');
grid on;
grid minor;
subplot(2,1,2)
plot(CDir,AngT,'k--o','LineWidth',1.5); hold on;
plot(CDir,AngT_riflex,'b-s','LineWidth',1.5); hold on;
% legend('Analytical','RIFLEX','Location','Best')
xlabel('Current direction (deg)');
ylabel('Top End Angle (\deg)');
grid on;
grid minor;
print(F2,'-dpng','.\X3ChangeDir\currentDirAng.png','-r300')
```