



NTNU – Trondheim
Norwegian University of
Science and Technology

Dynamic Modelling, Simulation and Visualization of Marine Crane Operations on DP Vessels

Wenzhe Zhang

Marine Technology

Submission date: June 2015

Supervisor: Eilif Pedersen, IMT

Norwegian University of Science and Technology
Department of Marine Technology



NTNU – Trondheim
Norwegian University of
Science and Technology

Dynamic Modelling, Simulation and Visualization of Marine Crane Operations on DP Vessels

Wenzhe Zhang

Submission date: June 2015

Supervisor: Eilif Pedersen, Associate Professor, IMT

Co-Supervisor: Stian Skjong and Børge Rokseth, PhD Candidate, IMT
Håvard Holm, Associate Professor, IMT

Norwegian University of Science and Technology
Department of Marine Technology

**MASTER THESIS IN MARINE ENGINEERING
SPRING 2015
FOR
STUD.TECH.WENZHE ZHANG**

**DYNAMIC MODELLING, SIMULATION AND VISUALIZATION OF
MARINE CRANE OPERATIONS ON DP VESSELS**

Work Description:

Dynamic modelling of systems such as marine surface vessels, underwater vehicles, robotic manipulators and more, have become increasingly important in the last decades due to, among other things, the increased use of model based motion control. These systems can however be complicated to model, especially when several systems are interconnected, such a vessel performing a heavy lifting crane operation. Furthermore, the dynamics of these systems are typically subject to various external loads such as waves, wind and actuator systems. The Offshore Ship Operation Model (OSOM) is a bond graph based model platform developed for such systems which allows for both effective production of models, and flexibility in interfacing other systems and external loads. The platform is implemented in the 20-sim software and visualized in GLview. Currently, the OSOM platform consists of a DP vessel model with relevant sub-systems, and three degrees of freedom crane.

The scope of work for this thesis is to firstly, develop a 3D visualization for the OSOM platform, and secondly to equip the crane model with a hydraulic system. In addition, simulation studies should be performed on the model, and presented as plots and animations.

Scope of Work

1. Review literature on computer visualization techniques.
2. Review literature on crane hydraulic systems.
3. Review literature and study the OSOM platform.
4. Implement the hydraulic system for the marine crane in the OSOM that model with:
 - (a) Hydraulic power system.
 - (b) Actuator system.
 - (c) A simple position control system, including reference model.

- (d) Study and discuss the friction model for the crane joints.
 - (e) Discuss the interface between crane hydraulic system and OSOM platform.
5. Design an interface between 20-sim and some third party 3D-visualization software such as PARAView, GLview, or Visual Python, in order to enhance 3D-animations of OSOM.
 6. Explore the possibility of running 3D-animations on third party software on-line during simulations.
 7. Document each step in the process.

The report shall be written in English and edited as a research report including literature survey, description of mathematical models, description of control algorithms, simulations results, discussion and conclusion including a proposal for further work. Source code developed shall be provided on a CD with code listing enclosed in appendix.

It is supposed that Department of Marine Technology, NTNU, can use the results freely in its research work by referring to the students work.

The master thesis should be submitted in three copies within 10.06.2015.

Supervisor: Eilif Pedersen, Associate Professor, IMT
Co-Supervisor: Stian Skjong and Børge Rokseth, PhD Candidate, IMT
Håvard Holm, Associate Professor, IMT

Preface

This master thesis is submitted to the Norwegian University of Science and Technology (NTNU) in the request of the specialization in Marine Engineering (Marine Machinery), Department of Marine Technology. This master thesis mainly aims to study the methods of modelling, simulation, control, analysis and visualization of marine crane system on Dynamic Positioning(DP) vessel in marine operations.

This master thesis has been conducted at the Department of Marine Technology, under the supervision of Associate Professor Eilif Pedersen. First and foremost I offer my sincerest gratitude to him, who has given me a lot of supports, suggestions and freedom when writing the master thesis. I also would like to extend my thanks to my co-supervisors: PhD candidate Børge Rokseth and Stian Skjong, who has given me a lot of guidances on basic theories and software implementation, as well as interesting discussions about different topics. I really appreciate their help and encouragement. In addition, Associate Professor Håvard Holm gave me many hints on 3D visualization of OSOM platform, on both theory aspect and technical aspect. Again, it is my great pleasure to acknowledge the supports from all of my supervisors.

Finally, I would like to thank the Department of Marine Technology for giving the freedom to select and develop topics of my interest in this master thesis, and the necessary resources that provided by NTNU.

Wenzhe Zhang
Trondheim, Norway.
Monday 8th June, 2015

Abstract

The research presented in this thesis is based on the idea that to find a better method of modelling, simulation and controlling the dynamic mechanical systems by using Lagrange method and bond graph approach. Based on the Offshore Ship Operation Model (OSOM), different relevant sub-systems of marine crane is studied and developed in 20-sim. The 3D visualization of marine crane operations on DP vessels in OSOM will be developed in GLview.

In the filed of marine operations, especially for the subsea technologies for Oil and Gas industry which is highly rely on the Platform Supply Vessel (PSV) or Offshore Construction Vessel (OSV). These vessels are equipped with marine crane system or winch system that can provide heavy lifting operation for the subsea constructions. The OSOM platform emphasizes on the interconnected behavior of the vessel and the equipment on board, as well as the environmental loads, such as wave load, current load and wind load during the operations. This is because the equipment on board and the DP vessel will have significant impact on the dynamic behavior between each other. This thesis mainly focus on marine crane and its sub-systems on the DP vessel, and set a framework for modelling, simulation and analysis of the dynamics of rigid body systems through Lagrange method and bond-graph approach. Based on this framework, many sub-systems such as actuators, hydrodynamics and controllers can be interfaced. In addition, in order to build an efficient and handy marine operation simulator, the visualization of the model is very important. This thesis provides the method to achieve the requirements by using Dynamic Link Library to connect simulator 20-sim and visualization tool GLview. The .DLL file is written by C code and print the visualization code to .VTF file for 3D animation in GLview.

The OSOM platform is mainly developed by Rokseth and Skjong (2015) which in order to develop a high fidelity simulator of an offshore construction vessel with heavy duty crane. This thesis is based on the OSOM and the specialization project of the author himself. The OSOM is still under development and several further works is provided in the last chapter.

Key Words : Crane-Vessel system, bond graph modelling, 3D visualization, crane hydraulic system, position control, system dynamics, dynamic linked library.

Contents

List of Figures	ix
List of Tables	xiii
1 Introduction	1
1.1 Background and Motivation	1
1.2 Problems and Challenges	3
1.3 Related Work	4
1.4 Structure of the Master Thesis	5
2 Background and Basic Theories	7
2.1 Basic Theories about Kinematics	7
2.1.1 Position and Orientation of Rigid Bodies System	7
2.1.2 Rotation Matrix	8
2.1.3 Direct Kinematics	10
2.1.4 Inverse Kinematics	10
2.2 Lagrangian Mechanics	11
2.2.1 Kinetic Energy	12
2.2.2 Potential Energy	13
2.2.3 Coordinates	13
2.2.4 Generalized Forces	15
2.2.5 Equations of Motion	16
2.3 System of Linked Rigid Bodies	19
2.3.1 Position and orientation	20
2.3.2 Geometric Jacobian Matrix	21
2.3.3 Lagrange Hamiltonian Equation	23
2.4 Bond Graph Theory	25

2.4.1	Basic Knowledge of Bond-graph	25
2.4.2	Multi-port Fields	30
3	Basic Models Studies	35
3.1	Offshore Crane Operation	35
3.2	OSOM Platform	36
3.2.1	Marine Vessel Dynamics	38
3.2.2	Combine the Vessel Dynamics and Marine Crane Dynamics .	43
3.2.3	Bond Graph Implementation of OSOM Platform	46
3.2.4	Sub-models Connect with OSOM Platform	47
4	3D Visualization of OSOM Platform	53
4.1	Advantages and Applications of Visualization	53
4.2	Softwares	54
4.3	Challenges in 3D Visualization	56
4.4	Communication Between 20-sim and GLview	57
4.5	Programming and Modelling Steps in 20-Sim	60
4.6	Programming and Modelling Steps in C Programming	62
4.6.1	Wave Theories	63
4.6.2	Programming	65
4.7	Programming and Modelling Steps in GLview Inova	73
4.7.1	Preparing the ship 3D model	73
4.7.2	Build the Crane 3D Model	75
4.7.3	VTF Code Structure	78
4.7.4	Comparison of Two Methods of Implementation in GLview .	83
4.8	Visualization Results and Studies	84
4.8.1	Case One	85
4.8.2	Case Two	91
4.8.3	Real Time Visualization	96
4.8.4	Further Works for Visualization of OSOM	98
5	Crane Hydraulic Sub-systems in OSOM	101
5.1	Modelling the Crane Hydraulic System	101
5.1.1	Hydraulic Pump	104
5.1.2	System Accumulator	105
5.1.3	Check Valve	106
5.1.4	Relief Valve	106
5.1.5	Compensate Valve	108
5.1.6	4/3 Directional Control Valve	110
5.1.7	Hydraulic Cylinder and Motor	111
5.1.8	Pipe	113
5.1.9	Position Control System	115

5.1.10	Friction Model	116
5.1.11	Geometry Relations between Crane Arm and Hydraulic Actu- ators	117
5.1.12	Crane Working Space	119
5.2	Completed Bond-Graph Model	120
5.2.1	Interface with OSOM	120
5.2.2	Co-Simulation	122
6	Hydraulic System Case Studies	125
6.1	Hydraulic Pump	125
6.2	Relief Valve Studies	126
6.3	Speed Limitation and Controller Gain Test	128
6.4	Variable Bulk Modulus Studies	134
6.5	Payload Studies	139
6.6	Crane Operation Simulation	144
6.6.1	System Sensitivity Study	149
7	Conclusion and Further Work	151
7.1	Conclusion	151
7.2	Further Work	152
	Bibliography	153
	Appendices	
A	Acronyms	155
B	C-code of Visualization DLL File	157
C	20-Sim Code for Wave Block in OSOM	163
D	VTF File Structure of Example 4.2	167
E	Parameters of Hydraulic System	173

List of Figures

1.1	OCV Normand Vision with crane and winch (Source: Ocean Installer).	2
2.1	Position and orientation of a rigid body	8
2.2	System of example 2.2	17
2.3	Open chain rigid bodies	20
2.4	Power bond and its direction	26
2.5	Elements in different system (Source: Pedersen and Engja (2010))	28
2.6	Element in bond-graph (Source: Pedersen and Engja (2010))	31
2.7	N-port C-field	32
2.8	N-port IC-field	33
3.1	Marine vessel dynamic simulation results	42
3.2	Vessel crane system (Source:Pål et al. (2012))	43
3.3	Basic model IC field implementation	46
3.4	Overview of OSOM in 20-sim	48
3.5	Thrusters allocation	49
3.6	Joint displacements	50
3.7	Torques on each joint	51
4.1	Initial 3D animation of OSOM in 20-sim	55
4.2	Communication between 20-Sim and GLview	59
4.3	Window of parameter setting	62
4.4	JOWSWAP spectrum under Hs=6m Tp=8s	64
4.5	Wave carpet elements	68
4.6	Wave elevation generated from C code in GLview	70
4.7	PSV modle in Delft Ship	74
4.8	Compare the STL and CAD format	74

4.9	Improt model in GLview	75
4.10	The finished crane 3D model	76
4.11	The components of crane 3D model	77
4.12	The completed DP vessel with marine crane	77
4.13	Simple example in GLview	84
4.14	3D visualization from y direction	85
4.15	3D visualization from z direction	86
4.16	3D visualization front view	86
4.17	3D visualization with wave vectors	86
4.18	3D visualization from y direction	87
4.19	3D visualization from z direction	87
4.20	3D visualization front view	88
4.21	3D visualization with wave vectors	88
4.22	Position of COG in case 1.1	89
4.23	Position of COG in case 1.2	89
4.24	Roll, pitch, yaw angles in case 1.1	90
4.25	Roll, pitch, yaw angles in case 1.2	90
4.26	Crane dynamics in case 1.1	91
4.27	Crane dynamics in case 1.2	91
4.28	3D visualization in GLview	92
4.29	3D visualization in GLview	92
4.30	3D visualization in GLview	93
4.31	3D visualization in GLview	93
4.32	Position of COG in case 2.1	94
4.33	Position of COG in case 2.2	94
4.34	Roll, pitch, yaw angles in case 2.1	95
4.35	Roll, pitch, yaw angles in case 2.2	95
4.36	Crane dynamics in case 2.1	96
4.37	Crane dynamics in case 2.2	96
4.38	FPS setting in GLview	97
4.39	Short-crest wave model in GLview	98
5.1	Marine crane components and structure	102
5.2	System layout of hydraulic cylinder	103
5.3	System layout of crane hydraulic system	103
5.4	HPU in bond graph	104
5.5	Bladder-type accumulator	105
5.6	Check valve(20-sim lib, Kleijn and Groothuis (2014))	106
5.7	Relief valve layout	107
5.8	Bond graph model of relief valve	107
5.9	Compensator layout	109
5.10	Bond graph model of compensator	109

5.11	Symbol and bond graph model of 4/3 directional valve	110
5.12	Symbol and structure of hydraulic cylinder	111
5.13	Bond graph model of hydraulic cylinder	112
5.14	Bond graph model and symbol of hydraulic motor	113
5.15	Bond graph model of pipe	114
5.16	Compare of transfer function	115
5.17	Position control system design	116
5.18	Geometry relations of crane	118
5.19	Working space of crane	119
5.20	Completed bond graph model of marine crane hydraulic system	121
5.21	Actuators connect with IC field	122
5.22	Structure of FMI	123
6.1	Variable flow pump	126
6.2	Simulation results of case 6.2.1	127
6.3	Simulation results of case 6.2.2	128
6.4	Piston velocity and position with speed limitation 1 m/s	130
6.5	Piston velocity and position with speed limitation 0.5 m/s	131
6.6	Simulation results with gain 0.001	132
6.7	Simulation results with gain 0.005	132
6.8	Simulation results with gain 0.01	133
6.9	Simulation results with gain 0.1	133
6.10	Relation between bulk modulus and pressure	135
6.11	Hydraulic cylinder system with constant bulk modulus	136
6.12	Hydraulic cylinder system with variable bulk modulus	137
6.13	Swing motor system with constant bulk modulus	138
6.14	Swing motor system with variable bulk modulus	138
6.15	System dynamics with 10 metric tons load	140
6.16	System dynamics with 20 metric tons load	140
6.17	System dynamics with 30 metric tons load	141
6.18	System dynamics with 35 metric tons load	141
6.19	35 metric tons load after increasing the accumulator volume	142
6.20	Extreme condition with 170 metric tons load	143
6.21	Crane operation step 1	144
6.22	Crane operation step 2	145
6.23	Crane operation step 3	145
6.24	Crane operation step 4	146
6.25	Crane operation step 5	146
6.26	Dynamics of crane joint 3 (Crane jib)	147
6.27	Dynamics of crane joint 2 (Crane boom)	148
6.28	Dynamics of joint 1 (Crane base)	148
6.29	Pressure of accumulators	149

List of Tables

3.1	Parameters in OSOM simulation example	42
4.1	Top-level identifiers in GLview	78
4.2	Parameters and settings of OSOM visualization case 1.1	85
4.3	Parameters and settings of OSOM visualization case 1.2	87
4.4	Parameters and settings of OSOM visualization case 2.1	92
4.5	Parameters and settings of OSOM visualization case 2.2	93
6.1	Parameters and results of relief valve studies	128
6.2	Parameters of single cylinder actuator	129
6.3	Parameters of hydraulic swing motor system	139
6.4	Key parameters in simulation of crane operation	147
6.5	Parameters sensitivity study	150

Chapter 1

Introduction

This chapter gives the general introduction and description about this master thesis, pointing out the backgrounds and motivations, as well as problems, challenges and related work. The structure of this master thesis is presented in the end of this chapter for readers to catch the outline easily.

1.1 Background and Motivation

The field of marine operations has evolved rapidly over the past thirty years and become a mature technology with several widespread applications. Especially in offshore Oil and Gas industry, the drilling technology and subsea operations had a quick development. Advanced modelling, simulation, control and visualization methods based on advanced computers, sensors, actuators and materials make it possible to do more accurate and reliable marine operations. It is used in many applications such as lifting, subsea installation and offshore platform construction by heavy duty marine crane in figure 1.1. This master thesis mainly focus on modelling, simulation and control of marine crane sub-systems based on OSOM. In addition, the simulation results that developed in 20-sim(bond-graph simulation software) based on OSOM platform is visualized in 3D visualization software GLview. The method used for modelling the linked rigid bodies system such as crane mounted on a Dynamic Positioning(DP) vessel is Lagrange method with quasi-coordinate based on bond graph approach. The method for 3D visualization is implemented by using C programming to generate Dynamic Link Library file, which connects the 20-sim and GLview, and finally prints the code of .VTF file for the animation in



Figure 1.1: OCV Normand Vision with crane and winch (Source: Ocean Installer).

GLview. By doing this, we can further develop some marine operation simulators for the personnel training and operation planning, which can increase the reliability and reduce the overall cost during different operations under different environments.

Bond-graph approach is a flexible and readable modelling method. This method has systematic rules and principles for the different dynamic systems to follow, such as for mechanical systems, hydraulic systems, electrical systems, etc. The property that can fit different dynamic systems make it much easier to connect the marine crane system to its sub-systems and other sub-models, for instance, electrical power system and the motion of DP vessel. By properly development and setting the internal and external interfaces can these goals be achieved.

The marine manipulators are usually classified as electric powered and hydraulic powered. For the electric powered manipulators, which are used on small size Remote Operated Vehicle (ROVs) that of observation class and inspection class. For the hydraulic powered manipulators or crane, which is usually equipped on work class ROVs, seabed-working vehicles or marine surface vehicles like Platform Supply Vessel (PSVs) or Offshore Barges that need much more lifting or turning ability. In this master thesis the heavy duty crane on marine surface vessel is discussed. During the marine operations, especially lifting operation in high seas, the interconnected dynamics of marine vehicles and sub-systems with marine cranes are important. The interactions of each systems will influence the behaviors of vessel and cranes much, hence they play very important roles for the safe and efficient operation as well.

A complex simulation model usually contains thousands of parameters and results. How to manage and show the structure of these simulation results and data is very important, i.e. turn the research achievements to engineering applications. 3D

visualization can use its rich graphic design to unfold the simulation results in a much more understandable way and hence provides a easy and efficient path to communication between users and softwares.

Here are three main motivations that encourage the author to write this master thesis. The first one is that the marine vessel-crane systems need to be treated in details. There are a lot of sub-systems and sub-models that worth researching and developing in order to create a better model for further use, such as crane hydraulic system. The second one is that the proper modelling and simulation of marine crane system on DP vessel plays an important role in all kinds of marine operations. It will be of great benefits if we find more efficient and accurate method to build the model as well as simulate and control the model. This can help us have a better understanding about the details of the dynamics in the marine crane operation, which can finally simplify the modelling and simulation steps, also the deign of the control laws. The third one is that to make the simulation results more accessible. More specifically, to visualize the dynamics of DP vessel and the equipment mounted on it. This is a very important step for industry application, such as the simulator for marine operation task verifying, planning and personnel training. A good simulator enable reduce the cost and increase the reliability during the process of marine operations.

1.2 Problems and Challenges

Modelling, simulation and visualization for complex systems faces many challenges. Since the limitations of available data for certain types of equipment, inaccurate estimates, assumptions and the interaction between different sub-systems or sub-models. For 3D visualization, the challenge is how to communication in different coding format between simulation software 20-sim and visualization software GLview. Besides that, controlling the time steps and making a real time 3D visualization simulation also need some efforts to meet the requirements.

The marine crane arm built for high seas environment is far more challenging than what can be found on a factory floor and it must be given the same characteristics in the mobile base in terms of design, control, and intelligence. Marine crane for example need to be built with the same specifications and requirements as marine vessels, which can be a PSV or OCV here.

Marine vessels are normally programmed to suppress environmental effects and disturbances. Also the crane arms need to be able to adapt to the environment. In the same way that crane arms need to possess the characteristics normally associated with mobile based(DP vessel) and the marine vessel need to take on the characteristics of factory floor manipulators. The requirements for a marine vessel that is to be used

in manipulation tasks are thus far higher in terms of accuracy, repeatability, and speed.

Increased performance in terms of accuracy, speed, and repeatability requires good mathematical models of the vessel-crane system. Accurate mathematical models are important both to obtain good simulation tools and for accurate control. It is a great challenge to find a mathematical framework that is both robust, intuitive and computationally efficient. It is also the reason why several sub-systems of marine crane system are discussed here, as they influence a lot on the quality of modelling and the simulation results.

Mathematical framework in this master thesis combines the Lagrangian mechanics and bond-graph approach to modelling and simulate the dynamics of marine crane system that interact with the marine vessels. The equations of dynamics will be implemented in the IC-field in bond-graph approach. It contains the basic elements of dynamic systems, such as inertia forces, Coriolis forces and centrifugal forces. The readability may decrease due to the unique and comprehensive structure of IC-field. However, IC-field is designed for vector values inputs, which can simplify the structure of bond graph, avoid differential causality, algebraic loop and increase the calculation efficiency. The next step is to connect the sub-system or other relative elements to the framework of the main model. For instance, the electrical power system, hydraulic actuators, motions of vessel and environmental loads.

1.3 Related Work

Modelling and simulation of the dynamics of marine crane systems mainly based on Lagrangian mechanics and bond-graph approach. The idea is to implement the equations of motion that developed from Lagrange-Hamilton equation. The main references about Lagrangian mechanics used here are Pål et al. (2012) and Ginsberg (1998). These references provide the basic principles and fundamental theories about rigid bodies motions and dynamics, also different kinematics in difference reference frames such as Cartesian-coordinates and Quasi-coordinates.

References about bond-graph used here are Pedersen and Engja (2010) and Wolfgang (2009). These references give the framework of bond-graph concept and modelling methods. Modelling method for rigid multi-body dynamics, multi-port field (IC-field) and interdisciplinary systems are provided. Some other papers also related to this area will be introduced later. The master thesis Sunde (2014) is the reference that helps me on some common topics in 3D visualization.

The reference Sciavicco and Siciliano (2000) and Kelly et al. (2005) present the basic theories in manipulator systems. The Sciavicco and Siciliano (2000) mainly

introduces the kinematics of manipulator system. The Kelly et al. (2005) focus on different control method for position and motion control for the manipulator system. The master thesis Rokseth (2014) helps me familiarize the basic modelling method and the method to implement the mathematical model into software program to run the simulation. The master theses Rokseth (2014) and Skjong (2014) help a lot with the similar topics that I have developed for the dynamics of crane space motion and marine crane sub-systems here.

The reference about the 3D visualization used here is Kernighan and Ritchie (1988) for C programming, which indicates to generate Dynamic Link Library file that connect the simulation software 20-sim and 3D visualization software GLview. For the coding structure in GLview, the reference Ceetron (2006) that from Ceetron GLview AS is used.

The reference Rokseth and Skjong (2015) concludes and discusses the OSOM platform in details, as well as the implementation in bond graph method.

Other references related to different topics in this master thesis will be introduced in the corresponding chapters. The details can be found in the Bibliography.

1.4 Structure of the Master Thesis

The structure of this master master thesis is introduced here. This will help readers to catch the outline of this master thesis.

Chapter 2 presents the background about Lagrangian mechanics and linked rigid bodies system modelling method. The bond-graph method will be introduced, especially the IC-field and interface for different sub systems in bond-graph. It provides a efficient way to implement the Lagrangian-Hamilton equations of motion for DP vessel and other sub systems such as hydraulic actuators, wire model and friction model of marine crane system etc.

Chapter 3 explains the basic theories about the dynamics of OSOM platform and the approaches that are used here for modelling and simulation. Offshore crane operation will be briefly introduced. Corresponding examples of each topics will be provided in the related chapter later.

Chapter 4 is mainly for the 3D visualization part of this master thesis. There will be some introductions about the advantages and applications of 3D visualization. More discussions will be provided for software implementation, programming and how to build dynamic communication between 20-sim and GLview. Finally, visualization

coding structure and modelling steps will be documented, several cases will be shown which can help the readers to have a better understanding.

Chapter 5 explains the sub-systems of marine crane in OSOM, which mainly focus on developing the crane hydraulic power system, crane hydraulic actuator system, position control system and discussion about crane joint friction model. The assumptions and simplifications are made in order to increase the efficiency of simulation by catching the main characteristics of the crane hydraulic system. The properties and modelling method of each components in the system will be introduced, and several simulation results will be discussed.

Chapter 6 mainly discusses the properties of system and sensitivity analysis of the system due to different variables in hydraulic system. Several case studies will be provided, followed by discussions.

Chapter 7 is the conclusion part that summarizes this master thesis. Several further work have been proposed for people who want to continue or enhance these models.

Appendix A Acronyms

Appendix B C-code of Visualization DLL File

Appendix C 20-Sim Code for Wave Block in OSOM

Appendix D VTF File Structure of Example 4.2

Appendix E Parameters of Marine Crane Hydraulic System

Chapter 2

Background and Basic Theories

In this chapter, the fundamental knowledge and basic theories will be presented for the further topics developed in this master thesis. Basic theories for this master thesis such as Lagrange method used in spatial motion of linked rigid bodies and bond-graph implementation approach is introduced.

2.1 Basic Theories about Kinematics

This section presents the basic knowledge of kinematics in marine crane system. The marine crane or linked rigid bodies are connected by means of revolute or prismatic joints. In this master thesis only the lower pair joint is considered, which means each joint only have one degree of freedom(DOF). The chain connected with mobile base (such as a DP vessel) in one direction and the end effector on another direction.

2.1.1 Position and Orientation of Rigid Bodies System

The rigid bodies are described in space by their positions and orientations with respect to the reference frame. As shown in figure 2.1, the rigid bodies should be indicated in the base frame by unit vectors on the frame axes. These vectors are indicated with respect to the reference frame.

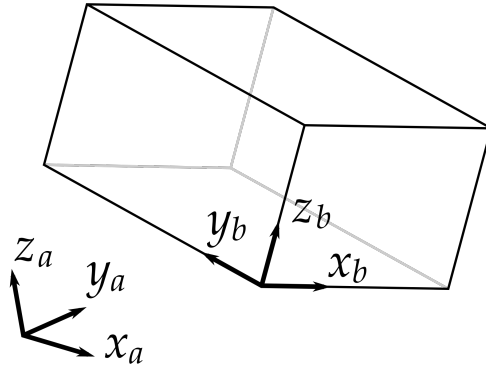


Figure 2.1: Position and orientation of a rigid body

$$\mathbf{x}' = x'_x x + x'_y y + x'_z z \quad (2.1)$$

$$\mathbf{y}' = y'_x x + y'_y y + y'_z z \quad (2.2)$$

$$\mathbf{z}' = z'_x x + z'_y y + z'_z z \quad (2.3)$$

2.1.2 Rotation Matrix

The three unit vectors describing the body orientation with respect to the reference frame can be combined in the 3×3 matrix which is the rotation matrix \mathbf{R} .

$$\begin{aligned} \mathbf{R} = \begin{bmatrix} \mathbf{x}' & \mathbf{y}' & \mathbf{z}' \end{bmatrix} &= \begin{bmatrix} x'_x & y'_x & z'_x \\ x'_y & y'_y & z'_y \\ x'_z & y'_z & z'_z \end{bmatrix} \\ &= \begin{bmatrix} x'^T x & y'^T x & z'^T x \\ x'^T y & y'^T y & z'^T y \\ x'^T z & y'^T z & z'^T z \end{bmatrix} \end{aligned} \quad (2.4)$$

The matrix \mathbf{R} is orthogonal against the unit vectors in the base frame. Therefore, \mathbf{R} has the following properties:

$$\mathbf{R}^T \mathbf{R} = \mathbf{I} \quad (2.5)$$

\mathbf{I} is the 3×3 identity matrix. Then we can develop our useful result:

$$\mathbf{R}^T = \mathbf{R}^{-1} \quad (2.6)$$

Therefore, the transpose of the rotation matrix is same as its inverse which is discussed in details in Ginsberg (1998).

New frame can be described by elementary rotations of the reference frame about one of the axes. The positive rotation is defined as the counter-clockwise of the relative axes, which fulfill the right hand rule.

Considering that when reference frame $O - xyz$ is rotated with an angle α about axis z , the unit vector of the new frame can be described in terms of their components with respect to the reference frame:

$$\mathbf{x}' = \begin{bmatrix} \cos(\alpha) \\ \sin(\alpha) \\ 0 \end{bmatrix} \quad (2.7)$$

$$\mathbf{y}' = \begin{bmatrix} -\sin(\alpha) \\ \cos(\alpha) \\ 0 \end{bmatrix} \quad (2.8)$$

$$\mathbf{z}' = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (2.9)$$

Hence the rotation matrix of frame $O - x'y'z'$ with respect to frame $O - xyz$ is:

$$\mathbf{R}_z(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

Based on the same principle, the rotations by an angle β about axis y and an angle γ about axis x are shown in the following matrices:

$$\mathbf{R}_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad (2.11)$$

$$\mathbf{R}_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{bmatrix} \quad (2.12)$$

These matrices are very useful when describe the systems of rigid bodies in spatial motion. Assuming there is a vector s in the space, it can has two different matrices in two different coordinates. Therefore, the vector \mathbf{s} and \mathbf{s}' have the relation:

$$\mathbf{s} = \mathbf{R}\mathbf{s}' \quad (2.13)$$

$$\mathbf{s}' = \mathbf{R}^T \mathbf{s} \quad (2.14)$$

The \mathbf{R} here is the product:

$$\mathbf{R} = \mathbf{R}_z(\alpha)\mathbf{R}_y(\beta)\mathbf{R}_x(\gamma) \quad (2.15)$$

For any set of two reference frames applying a sequence of three, or fewer rotations that will align one with the other, and thus relate any two reference frames using the three principal rotation matrices described above. Two reference frames will move relatively to each other. Any vector defined in the directions of one of the reference frames can be indicated in terms of others. This is very helpful when solving the kinematics problems in a system of rigid bodies in spatial motion.

2.1.3 Direct Kinematics

Direct kinematics can calculate the positions and orientations of the end effector as a function of joint variables. The joints considered in this master thesis are mainly low pair joints, which means that only one DOF each joint.

The open-chain manipulator or crane has $n + 1$ linked bodies or arms that connected by n joints. Each joint provides the mechanical structure with a single DOF. It is necessary to define a coordinate frame corresponding to each link. Hence describing the position and orientation of frame n with respect to frame O which has been described in the Ginsberg (1998), where the $\mathbf{T}_n^0(\mathbf{q})$ means the overall transfer function from frame n to frame O , and $\mathbf{A}_n^{n-1}(\mathbf{q}_n)$ represent the transfer function from the previous body to the next body.

$$\mathbf{T}_n^0(\mathbf{q}) = \mathbf{A}_1^0(q_1)\mathbf{A}_2^1(q_2)\dots\mathbf{A}_n^{n-1}(q_n) \quad (2.16)$$

To calculate the direct kinematics equations for open chain manipulator or crane, a systematic, general method is to be used to define the relative positions and orientations of two consecutive links. More specifically, that is how to calculate the rotation matrix between the two frame that attached on two linked bodies correspondingly.

2.1.4 Inverse Kinematics

For the forward or direct manipulator modelling and control, it is always necessary to know the joint variables and then calculate the position and orientation of the end effector. However, in many industry applications, joint variables are the function of the position and orientation of the end effector. More specifically, the variables in each joint need to be decided by the motion of end effector. This is defined as the inverse kinematics problem.

\mathbf{T} is the overall matrix that perform the transform function. This can be described in the general matrix form as:

$$\mathbf{T}_0^n = A_1(q_1)A_2(q_2)\dots A_n(q_n) = \mathbf{H} \quad (2.17)$$

Here the matrix \mathbf{H} is the homogeneous transformation, which shows the desire position and orientation of the end effector. It is much easier to just set the desire position and orientation of end effector. However, this non-linear equation will result in unknown variables. Moreover, due to the complexity of these equations which makes them hard to solve. In addition, there may exist several solutions to the motion of each body of the manipulator system. These solutions are only feasible based on mathematical calculations. For the engineering applications, several physical limitations need to be taken into consideration. These limitations are limited by the working space on ship, the physical properties of actuator that drives the manipulator or for the safe operation consideration. Therefore, the calculation results must be checked before applying into engineering applications. When several solutions are proved feasible, they are not unique. Thus choosing an optimal solution is very important and necessary.

2.2 Lagrangian Mechanics

In this section, Lagrangian equations of motion for a system of rigid bodies in spatial motion are presented. Lagrangian mechanics is a re-formulation of classical mechanics using the principle of stationary action which is presented in Goldstein (2001). These are derived from expressions for the kinetic and potential energy of the system. The potential energy can be expressed in terms of a set of generalized coordinates, \mathbf{q} , while the kinetic energy can be expressed in terms of the generalized coordinates, and their time derivatives, $\dot{\mathbf{q}}$. There are two ways to express the Lagrangian mechanics, the first one is the standard Lagrangian format. The second one is the quasi-coordinate format, which will be used for develop the equation of motion for marine vessels in 6-DOF. The quasi-coordinate is shown in the OSOM platform for modelling and simulate the dynamics of DP vessel.

For the rigid body dynamics, there are a lot of approaches that can derive the dynamics of mechanical systems. The author need to choose and consider the system properties that are of most interest to the problems met. For instance, in the mathematical models for simulation and controlling, the forces and state variables that will affect the motion of system will be extracted. Constrain forces and internal forces are very important when it comes to the mechanical design and analysis. However, for simulation and control engineer usually eliminate these forces in order to keep the equation simple.

For mechanical systems based on the modelling, simulation and controlling consideration, the interests are put on the effect of actuator and external forces affect the system. Actually, the equations presented by Lagrange are just a coordinate invariant formulation of Newton's law $F = ma$ (Ginsberg (1998)). The dynamic equations can be developed by the following steps:

1. Define a set of generalized coordinates \mathbf{q} and corresponding generalized velocities $\dot{\mathbf{q}}$.
2. Find the expressions for the kinetic energy and potential energy.
3. Write out the Lagrange equations.

The equation of general Lagrangian mechanics can be described same as the description in Ginsberg (1998):

$$\frac{d}{dt} \left(\frac{\partial \mathbf{T}}{\partial \dot{\mathbf{q}}} \right) - \left(\frac{\partial \mathbf{T}}{\partial \mathbf{q}} - \frac{\partial \mathbf{V}}{\partial \mathbf{q}} \right) = \boldsymbol{\tau} \quad (2.18)$$

2.2.1 Kinetic Energy

The first and most important thing that need to be derived is the rigid body kinetic energy. The kinetic energy T is given by the masses and the velocities of bodies in the system. The kinetic energy T is indicated as:

$$T(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}} \quad (2.19)$$

Assumed that the coordinate frame of the rigid body set the origin to the center of gravity of the rigid body while axes are aligned with the principle directions of the inertia. Hence, the constant inertia matrix (\mathbf{I}_b) can be written as a diagonal matrix:

$$\mathbf{M} = \mathbf{I}_b = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_x & 0 & 0 \\ 0 & 0 & 0 & 0 & I_y & 0 \\ 0 & 0 & 0 & 0 & 0 & I_z \end{bmatrix} \quad (2.20)$$

For the kinetic energy, which is a function of the linear velocity in the center of gravity plus the angular velocity of the bodies. Hence, we can develop this equation into more systematic format:

$$T_i(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} (\mathbf{v}_{\mathbf{c}g\mathbf{i}})^T m_i \mathbf{v}_{\mathbf{c}g\mathbf{i}} + \frac{1}{2} \mathbf{w}_i^T I_{b_i} \mathbf{w}_i \quad (2.21)$$

$\mathbf{v}_{\mathbf{c}g\mathbf{i}}$ is the linear velocity of the body, $\mathbf{w}_{\mathbf{i}}$ is the angular velocity of the body, and $\mathbf{I}_{\mathbf{b}\mathbf{i}}$ is the inertia of the body. These give a 3×3 matrix that different from the I_b which is a 6×6 matrix defined earlier for both mass and inertial. The subscript i represents the index of the body. The total kinetic energy is indicated:

$$T_i(\mathbf{q}, \dot{\mathbf{q}}) = \sum_{i=1}^m T_i(\mathbf{q}) \quad (2.22)$$

2.2.2 Potential Energy

Kinetic energy is quite universal that can be described in the same way for different rigid bodies. However, potential energy is not the same case as the kinetic energy, since it depends on the properties of the systems. For instance, for a car, the potential energy can be given by the gravitational forces and spring forces. For the marine vessel, it can also from the buoyancy force. Before calculating the potential energy, we need to define the position of the center of gravity of body i with respect to the original inertial reference frame a $\mathbf{r}_{\mathbf{c}g}^{\mathbf{i}}$. The mass of the rigid body is $\mathbf{m}_{\mathbf{i}}$, and the acceleration of gravity is g . The potential energy can be described as:

$$V_i(\mathbf{q}) = -m_i \mathbf{g}^T \mathbf{r}_{\mathbf{c}g}^{\mathbf{i}}(\mathbf{q}) + f_{ki}(\mathbf{q}) \quad (2.23)$$

In this equation (Rokseth and Pedersen (2014)), the position of the center of gravity of body i is relative to the origin of an inertial reference frame as $\mathbf{r}_{\mathbf{c}g}^{\mathbf{i}}$. Mass of bodies as m_i and acceleration of gravity g . The second term on the right hand side of the equation represent other contributions to the potential energy, such as the energy stored in springs.

The total potential energy can be described as:

$$V(\mathbf{q}) = \sum_{i=1}^m V_i(\mathbf{q}) \quad (2.24)$$

2.2.3 Coordinates

Reference frame gives us a standard to describe and monitor spatial motion of linked rigid bodies system. First and foremost is to find the motion a coordinate system. A coordinate system is determined by two choices:

1. Choose a point O , as the origin, which is fixed to the reference frame.
2. A set of three linearly independent orientations that is also fixed to the frame.

However, if engineers know some informations about the rigid bodies system, they can make great improvements to the final expression and simplify it. For instance,

if one point on the rigid body does not move when the body is moving, then set this point as origin point can be a smart choice. In the marine crane system, or linked rigid bodies systems, the origin point on the base which rotates around a fixed axis is usually a good choice, because all the points on this axis will not move. In addition, if the origin can be chosen on the center of gravity of one rigid body as well, then the expression for the inertia matrix can be greatly simplified. There are many ways to describe the coordinates here, one of them is the generalized coordinates, another approach is the quasi-coordinate. Actually the quasi-coordinate are linear combination of the generalized coordinate rates, not an alternative for generalized coordinates as the quasi-coordinates are functions of the generalized coordinates. The generalized coordinate will be introduced first.

Generalized Coordinates

Generalized coordinates are a set of linearly independent unique coordinates that define the position and orientation of a system. In order to describe the system in a unique way, system at least need the number of generalized coordinates equal to the number of degree of freedom. In some cases, the number of generalized coordinates is greater than the number of degree of freedoms n_{dof} . Then constrain equations are needed to restrict the motion of system. The number of constrain equations can be defined as (Ginsberg (1998)), where n_{ce} is the number of constraint equations.

$$n_{ce} = n_{gc} - n_{dof} \quad (2.25)$$

A set of generalized coordinates are not a unique set, meaning that for a particular system, there are generally many methods to define the set of generalized coordinates. It is advisable to consider the choice of generalized coordinates well. Simple case of a particle moving in a plane. We can set the Cartesian coordinates in several ways. For instance, $\mathbf{q} = [x, y]^T$ represents the position of a particle that defined by the x and y axis, $\mathbf{q} = [r, \theta]^T$ defines the position of particle by the displacement from original point to the particle and the angle between displacement vector and x axis or $\mathbf{q} = [x, \theta]^T$ that defines the particle position by the distance in x axis and the angle between x axis and the displacement vector from original point. It depends on what variables you have chosen.

Quasi-Coordinates

For systems with non-Euclidean configuration spaces, it is often easier to set the dynamic equations in terms of velocity variables that cannot be written as the time derivative of the position variables. For instance, when it comes to the angular motion, the angular velocity is not the time derivative of angular changes. In this case, it is useful to define a set of quasi-coordinates based on the generalized coordinates in the form as Meirovitch (2003):

$$\mathbf{w} = \alpha^T \dot{\mathbf{q}} \quad (2.26)$$

Here the \mathbf{w} is the quasi-coordinates, the time integral of \mathbf{w} does not have any physical meaning. α^T is the linear transformation matrix and $\dot{\mathbf{q}}$ is the generalized coordinates rates. After the transformation, the quasi-coordinates can describe the angular velocity of the rigid body. For the manipulator system, we will not use the quasi-coordinates, however, when we develop the motion of vessel, this method should be introduced.

Here we can see an example of the quasi-coordinates. Since part of the kinetic energy is contributed by the angular velocities. So the kinetic energy then can be expressed in terms of generalized coordinates combine with quasi-coordinates. From the equation (2.26), we can get:

$$\dot{\mathbf{q}} = \beta \mathbf{w} \quad (2.27)$$

Here the β defined as $(\alpha^T)^{-1}$. So the kinetic energy can be described as:

$$T(\mathbf{q}, \dot{\mathbf{q}}) = T(\mathbf{q}, \beta \mathbf{w}) = \bar{T}(\mathbf{q}, \mathbf{w}) \quad (2.28)$$

However, the factors determine the potential energy such as displacement of body along the gravity direction is not the derivative of generalized coordinates, so there is no need to introduce quasi-coordinates in this situation.

2.2.4 Generalized Forces

Energy usually will not be generated internally for the mechanical system (except such as an engine if you look at it as an integral system) we consider here. Therefore, the input forces and torques are very important and can affect the dynamics of the marine crane-vessel system much. In order to calculate these non-conservative forces, the generalized forces are discussed here. The non-conservative forces \mathbf{F}_j act on a rigid body and have a virtual displacement δr_j . Then the work done by these forces can be described as (Rokseth (2014)):

$$\delta \mathbf{W}_j = \mathbf{F}_j^T \delta r_j \quad (2.29)$$

The virtual displacement of the j -th non-conservative force can be expressed in the formulation with respect to the generalized coordinates as:

$$\delta \mathbf{W}_j = \sum_{i=1}^n \frac{\partial r_j}{\partial \mathbf{q}_i} \delta \mathbf{q}_i \quad (2.30)$$

Then we can insert the generalized force τ_j in the generalized coordinates:

$$\delta \mathbf{W}_j = \sum_{i=1}^n \tau_j^i \delta \mathbf{q}_i \quad (2.31)$$

By combining these expressions, we can get the i -th generalized force due to the non-conservative force \mathbf{F}_j :

$$\tau_{ij} = \mathbf{F}_j^T \frac{\partial r}{\partial \mathbf{q}_i} \quad (2.32)$$

Then we can put all the τ_{ij} together into a matrix and place it in the right hand side of the equation of motion.

2.2.5 Equations of Motion

Until now, everything needed for generate the Lagrangian equations of motion are prepared. In the rigid bodies system that describes in n generalized coordinates, we can get the final equation of motion:

$$\frac{d}{dt} \left(\frac{\partial \mathbf{T}}{\partial \dot{\mathbf{q}}} \right) - \left(\frac{\partial \mathbf{T}}{\partial \mathbf{q}} - \frac{\partial \mathbf{V}}{\partial \mathbf{q}} \right) = \boldsymbol{\tau} \quad (2.33)$$

We can put all the kinetic energy vectors and potential energy vectors of all the bodies into one equation. This makes it easier to implement and organize.

Now, the dynamic equations can be put in matrix form. The matrix form of the dynamic equations is quite convenient for system dynamics analysis and stability analysis, also easier for model based control design. In the cases that this thesis mainly focus on, the most important matrices for the manipulator system and vessel positioning systems are inertia matrices \mathbf{M} and the Coriolis and centrifugal matrix \mathbf{C} . The total mass and the moments of the body will contribute and form the inertia matrix. This matrix provides a bridge between the forces acting on the system and the resulting acceleration, similar with the Newton's second law.

However, for the rigid bodies system, there are other effects affecting the system. For example, the centrifugal and Coriolis forces that included in the Coriolis matrix. If \mathbf{F} defined as the external forces, then the dynamic equation can be indicated as:

$$\mathbf{M}(\mathbf{x})\ddot{\mathbf{x}} + \mathbf{C}(\mathbf{x}, \dot{\mathbf{x}})\dot{\mathbf{x}} = \mathbf{F} \quad (2.34)$$

Based on this expression, more properties of the system can add in this equation. For instance, we can add damping forces $\mathbf{D}(\dot{\mathbf{x}})\dot{\mathbf{x}}$, which is very important for the marine vessels either PSVs or underwater robotics. Potential forces such as gravitational forces $G(x)$ can also be included:

$$\mathbf{M}(\mathbf{x})\ddot{\mathbf{x}} + \mathbf{C}(\mathbf{x}, \dot{\mathbf{x}})\dot{\mathbf{x}} + \mathbf{D}(\dot{\mathbf{x}})\dot{\mathbf{x}} + G(x) = \mathbf{F} \quad (2.35)$$

From here we can see that the potential forces can be included in the terms of Lagrangian equations of motion, and can also be in the state variables.

Similarly, the equation of motion based on Lagrangian mechanics can also be set in the matrix form:

$$\begin{bmatrix} mI & 0 \\ 0 & Ib \end{bmatrix} \begin{bmatrix} \dot{v}_{0/b}^b \\ \dot{w}_{0/b}^b \end{bmatrix} - \begin{bmatrix} -m\hat{w}_{0/b}^b & 0 \\ 0 & \hat{I}_b \hat{w}_{0/b}^b \end{bmatrix} \begin{bmatrix} v_{0/b}^b \\ w_{0/b}^b \end{bmatrix} = \begin{bmatrix} \tau_v \\ \tau_w \end{bmatrix} \quad (2.36)$$

Example 2.1

Now, we will go through a simple example based on Lagrangian mechanics.

The planar mechanical system is shown in figure 2.2 that has a body 1 of mass m and a pendulum 2 with the mass M concentrated at B. The length of AB is L and the stiffness of spring R is k . The spring will constrained in horizontally motion. Generalized coordinate of the system are used. The generalized coordinates

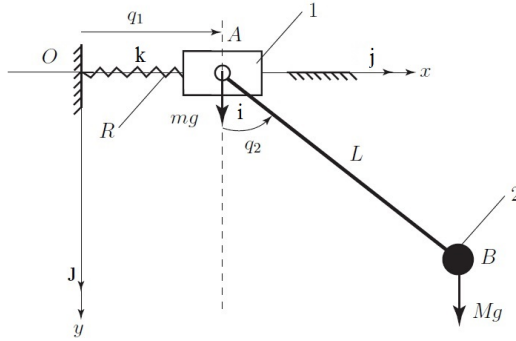


Figure 2.2: System of example 2.2

are associated with the position of the system. The generalized coordinate 1 $q_1(t)$ contributes the linear displacement of the slider. The generalized coordinate 2 $q_2(t)$ is the angle between the AB and the vertical direction.

Kinematics

A Cartesian reference frame $O - xyz$ and the vectors $[i, j, k]$ is chosen. The position vector of mass 1 is described as:

$$r_1 = r_A = q_1(t)i \quad (2.37)$$

The position vector of mass 2 is:

$$r_2 = r_B = [q_1(t) + L\sin q_2(t)]i + L\cos q_2(t)j \quad (2.38)$$

The velocity of the slider 1 is:

$$V_A = \frac{\partial r_A}{\partial t} = \dot{r}_A = \dot{q}_1 i \quad (2.39)$$

And the velocity of the B is:

$$V_B = \frac{\partial r_B}{\partial t} = \dot{r}_B = (\dot{q}_1 + L\dot{q}_2 \cos q_2) i - L\dot{q}_2 \sin q_2 j \quad (2.40)$$

kinetic Energy

The kinetic energy of the slider 1 is

$$T_1 = \frac{1}{2} m V_A V_A = \frac{1}{2} m \dot{q}_1^2 \quad (2.41)$$

And the kinetic energy of the mass 2 is:

$$T_2 = \frac{1}{2} M V_B V_B = \frac{1}{2} M (\dot{q}_1^2 + 2L\dot{q}_1\dot{q}_2 + L^2\dot{q}_2^2) \quad (2.42)$$

Hence the total kinetic energy is:

$$T = T_1 + T_2 \quad (2.43)$$

Generalized Forces

The force that acts on mass 1 at A are spring force and gravity force:

$$F_A = -kq_1 i + mgj \quad (2.44)$$

The gravity force acts on mass 2 at B is:

$$F_B = Mgj \quad (2.45)$$

There are two generalized forces. Generalized forces due to q_1 is:

$$Q_1 = F_A^T \frac{\partial r_A}{\partial q_1} + F_B^T \frac{\partial r_B}{\partial q_1} = -kq_1 \quad (2.46)$$

Generalized forces due to q_2 is:

$$Q_2 = F_A^T \frac{\partial r_A}{\partial q_2} + F_B^T \frac{\partial r_B}{\partial q_2} = -MgL \sin q_2 \quad (2.47)$$

Lagrange's Equation

The two Lagrangian equations are:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_1} \right) - \left(\frac{\partial T}{\partial q_1} \right) = Q_1 \quad (2.48)$$

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_2} \right) - \left(\frac{\partial T}{\partial q_2} \right) = Q_2 \quad (2.49)$$

Based on equations above, we can develop the expression for the equation 2.48 as:

$$\frac{\partial T}{\partial \dot{q}_1} = (m + M)\dot{q}_1 + LM\dot{q}_2 \cos q_2 \quad (2.50)$$

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_1} \right) = (m + M)\ddot{q}_1 + LM\ddot{q}_2 - LM\dot{q}_2^2 \sin q_2 \quad (2.51)$$

$$\frac{\partial T}{\partial q_1} = 0 \quad (2.52)$$

For q_2 we have:

$$\frac{\partial T}{\partial \dot{q}_2} = LM(\dot{q}_1 \cos q_2 + L\dot{q}_2) \quad (2.53)$$

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_2} \right) = LM(\ddot{q}_1 \cos q_2 - \dot{q}_1 \dot{q}_2 \sin q_2 + L\ddot{q}_2) \quad (2.54)$$

$$\frac{\partial T}{\partial q_2} = -LM\dot{q}_1 \dot{q}_2 \sin q_2 \quad (2.55)$$

Hence, the equation of motion can be described as:

$$(m + M)\ddot{q}_1 + LM\ddot{q}_2 \cos q_2 - LM\dot{q}_2^2 \sin q_2 + kq_1 = 0 \quad (2.56)$$

$$LM\ddot{q}_1 \cos(q_2) + ML^2\ddot{q}_2 + MgL \sin q_2 = 0 \quad (2.57)$$

We can put the equations in the matrix form:

$$\begin{bmatrix} (m + M)\ddot{q}_1 + LM\ddot{q}_2 \cos q_2 - LM\dot{q}_2^2 \sin q_2 + kq_1 \\ LM\ddot{q}_1 \cos(q_2) + ML^2\ddot{q}_2 + MgL \sin q_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (2.58)$$

After that, we can implement the equations to some software solver to simulate the system.

2.3 System of Linked Rigid Bodies

This section will use some basic concepts and knowledge that stated from the previous sections. Different from Denavit-Hartenberg approach, here the physical

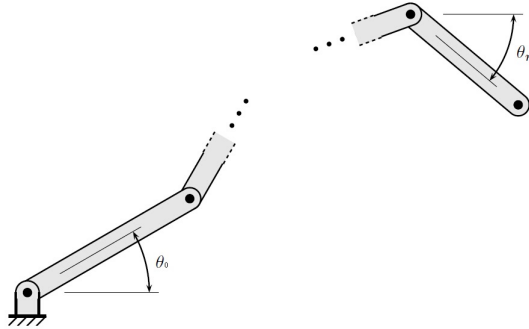


Figure 2.3: Open chain rigid bodies

properties will also be discussed. This includes various geometric relations in algorithm. In order to find the potential energy and kinetic energy needed in calculation of the Lagrangian mechanics, relevant coordinates and velocities should be discussed. First, in order to find the coordinates, the center of gravity of each bodies should be defined relate to certain reference frame. It is also necessary to find the point of each joint relate to the same reference frame that the center of gravity used. Based on the coordinates of the joints we can calculate the corresponding coordinates at the center of gravity of the bodies. Hence we can find the kinetic energy based on the linear velocities of the joints and angular velocities of them.

As shown in the figure 2.3, the center of gravity of each bodies (i -th) can be set, but the coordinate is defined in each joint of bodies. So if we want to know the position of center of gravity of body i , one more step that transfers the motion and position along the body need to be done. In the figure 2.3, the inertia reference frame denoted 0 is on the left (usually is the base, can be stable), a series of reference frames i are defined in each joint that attached to body i . On the right hand, the end effector denote as body n which is the body that we are most interested in the real applications such as marine crane.

2.3.1 Position and orientation

The coordinate $r_{i+1/i}^i$ indicates that the origin point of reference $i + 1$ relative to reference i is expressed in reference frame i . From the previous part we know that in order to find the position of center of gravity of body i , one more step need to be done. For one of the bodies locally, making body i as an example, the center of gravity of body i relative to the origin of local reference frame i and also describe in the reference frame i can be expressed as $r_{cg/i}^i$. The unit vector that along the body i (displacement) defined as e_i . By applying rotation matrices, the vectors describe

i -th body can be expressed in $i - 1$ -th reference frame. For instance, the vector of i -th body, r_i , which is defined and described in the i -th reference frame. Then, the reference frame i can be transferred to align with reference frame i with the rotation matrix $\mathbf{R}_i^{i-1}(\mathbf{q}_i)$, which is explained in the previous section.

$$r^{i-1} = \mathbf{R}_i^{i-1}(\mathbf{q}_i)r^i \quad (2.59)$$

We can set the rotation matrices from i -th body to the base, which is the 0-th body. Hence we can have the following equation:

$$r^0 = \mathbf{R}_1^0(\mathbf{q}_1)r^1 = \mathbf{R}_1^0(\mathbf{q}_1)\mathbf{R}_2^1(\mathbf{q}_2)\dots\mathbf{R}_{i-1}^{i-2}(\mathbf{q}_{i-1})\mathbf{R}_i^{i-1}(\mathbf{q}_i)r^i \quad (2.60)$$

Then,

$$r^0 = \mathbf{R}_i^0(\mathbf{q})r^i \quad (2.61)$$

We can also conclude that:

$$\mathbf{R}_i^0(\mathbf{q}) = \mathbf{R}_1^0(\mathbf{q}_1)\mathbf{R}_2^1(\mathbf{q}_2)\dots\mathbf{R}_{i-1}^{i-2}(\mathbf{q}_{i-1})\mathbf{R}_i^{i-1}(\mathbf{q}_i) \quad (2.62)$$

Based on what has been developed so far, the position of center of gravity of body i can also be described in the base reference frame relate to the origin of the base reference frame, which is the 0-th reference frame mentioned above.

$$r_{cgi/0}^0 = r_{1/0}^0 + r_{2/1}^0 + \dots + r_{i/i-1}^0 + r_{cgi/i}^0 \quad (2.63)$$

Here we can find that if one joint is a revolute joint, then the vector $r_{i/i-1}^{i-1}$ is constant. If one joint is prismatic, the following equation can be set (e_i is the vector along the body i):

$$r_{i/i-1}^{i-1} = r_{io/i-1} \quad (2.64)$$

The subscript $io/i - 1$ means that the origin of reference frame i relative to the origin of reference frame $i - 1$.

2.3.2 Geometric Jacobian Matrix

The relation between different joint positions has already been discussed. Based on that, we can connect the linear velocities and angular velocities with the derivation of the joint positions. The matrix that can complete this transfer is called geometric Jacobian matrix. Besides that, analytical Jacobian matrix is helpful when the information about end effector is very limited.

In order to find the linear velocity at the center of gravity and the angular velocity of corresponding bodies, we need to find the details of some geometric Jacobian matrix \mathbf{J} . For a point p at random linked rigid body that have velocity v_p , we can set the equation as (Kelly et al. (2005)):

$$v_p = \mathbf{J}(q)\dot{\mathbf{q}} \quad (2.65)$$

Then, we can find the linear velocity at the center of gravity of i -th body. This velocity will be the consequence and contribution of the previous body, here we set it as k -th body. In the coordinate k , hence the expression can be:

$$v_{cgi/0}^{\dot{q}_i} = \mathbf{J}_{q_k}^{v_cgi} \dot{\mathbf{q}}_k \quad (2.66)$$

Here q_k means that the where the velocity contribution from, and v_cgi shows the objective that receive the velocity contribution from q_i .

Generally, we can write the geometric Jacobian matrix in the differential kinematic equation as following:

$$\dot{\mathbf{X}} = \begin{bmatrix} v_n^0 \\ w_n^0 \end{bmatrix} = \begin{bmatrix} \mathbf{J}_v \\ \mathbf{J}_w \end{bmatrix} \dot{\mathbf{q}} = \mathbf{J}_n^0(\mathbf{q}) \dot{\mathbf{q}} \quad (2.67)$$

The rigid body dynamics in spatial space includes linear velocity and angular velocity. Linear velocity is along three axes and angular velocity is about three axes. Hence, the Jacobian matrix is a $6 \times n$ matrix, here n represent the number of links.

First, we can find a random joint i . If joint i rotate about the $i - 1$ -th axis, then the angular velocity of i -th body can be expressed in the $i - 1$ frame as following:

$$\mathbf{w}_i^{i-1} = \dot{\mathbf{q}}_i e_i \quad (2.68)$$

Base on this, the angular velocity of the end effector \mathbf{w}_n^0 can now be described as:

$$\mathbf{w}_n^0 = \sum_{i=1}^n s_i \dot{\mathbf{q}}_i e_k^i \quad (2.69)$$

The s_i here is a yes or no value that judge if the $i - th$ joint is prismatic or revolute. The s_i equal to 0 if the joint is prismatic and equal to 1 if the joint is revolute. Notice that the coordinate vector k should be expressed in the base frame 0. Thus get the Jacobian matrix for angular velocity as:

$$\mathbf{J}_w = [s_1 e_1^0, s_2 \mathbf{R}_1^0 e_2^0, \dots, s_n \mathbf{R}_{n-1}^0 e_n^0] \quad (2.70)$$

Now the angular velocity can be described as:

$$\mathbf{w}_{i/0}^0 = [\mathbf{J}_{q_1}^{w_i}, \mathbf{J}_{q_2}^{w_i}, \dots, \mathbf{J}_{q_i}^{w_i}, \mathbf{0}_{3 \times (n-i)}] \dot{\mathbf{q}} = \mathbf{J}_i^w(q) \dot{\mathbf{q}} \quad (2.71)$$

The other part of the Jacobian matrix is related to the linear velocity, similar to the angular velocity that developed above, we have:

$$\dot{\mathbf{q}}_{i/0}^0 = \sum_{i=1}^n \frac{\partial r_{i/0}^0}{\partial \mathbf{q}_i} \quad (2.72)$$

For each joint, they have a corresponding linear velocity Jacobian matrix, the expression is:

$$\mathbf{J}_i = \begin{bmatrix} J_v^i \\ J_w^i \end{bmatrix} \quad (2.73)$$

$$= \begin{cases} r_{cgi/i}^0 \times e_i^0 & \text{if } i \text{ is revolute} \\ e_i^0 & \text{if } i \text{ is prismatic} \end{cases}$$

Hence the sum of linear velocity can be described similar way as the angular velocity shown above:

$$v_{cgi/0}^0 = [\mathbf{J}_{\mathbf{q}_1}^{\mathbf{v}_{cgi}} \mathbf{q}_1, \mathbf{J}_{\mathbf{q}_2}^{\mathbf{v}_{cgi}} \mathbf{q}_2 \dots \mathbf{J}_{\mathbf{q}_i}^{\mathbf{v}_{cgi}} \mathbf{q}_i, \mathbf{0}_{3 \times (n-i)}] \dot{\mathbf{q}} = \mathbf{J}_{cgi}^v(q) \dot{\mathbf{q}} \quad (2.74)$$

To a conclusion, now both the linear velocity of body i and angular velocity of body i has developed, and can be put into matrix form:

$$v_i = \begin{bmatrix} v_{cgi/0}^0 \\ w_{i/0}^0 \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{cgi}^v(q) \\ \mathbf{J}_i^w(q) \end{bmatrix} \dot{\mathbf{q}} = \mathbf{J}_i(q) \dot{\mathbf{q}} \quad (2.75)$$

2.3.3 Lagrange Hamiltonian Equation

In this section, the Lagrangian equations of motion will be described in the state space form. The reason for introducing Lagrange Hamiltonian equation here is that this method has the generalized coordinates with the displacement q and the corresponding momentum p as the system state. It exactly fits the implementation format of the bond-graph approach. In the right hand side of the state space equation, there are forces and torques that act on the system as inputs.

Based on what we have done in the previous sections, the kinetic energy of the i -th body can be described as:

$$T_i(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} m_i \mathbf{v}_{cgi}^T \mathbf{v}_{cgi} + \frac{1}{2} \mathbf{w}_i^T I_{bi} \mathbf{w}_i \quad (2.76)$$

In addition, based on the geometric Jacobian matrix for linear velocity and angular velocity developed above, for i -th body:

$$\mathbf{v}_{cgi/0}^0 = \mathbf{J}_{cgi}^v(q) \dot{\mathbf{q}} \quad (2.77)$$

$$\mathbf{w}_{i/0}^0 = \mathbf{J}_i^w(q) \dot{\mathbf{q}} \quad (2.78)$$

In the state space form:

$$\mathbf{v}_i = \begin{bmatrix} v_{cgi/0}^0 \\ w_{cgi/0}^0 \end{bmatrix} = \mathbf{J}_i(q) \dot{\mathbf{q}} \quad (2.79)$$

Then explaining the details of mass and inertia of the body i in matrix form, here denoting M_i as the mass and inertia matrix:

$$\mathbf{M}_i = \begin{bmatrix} \mathbf{m}_i \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{bi} \end{bmatrix} \quad (2.80)$$

Insert this expression into kinetic energy equation:

$$\mathbf{T}_i(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \begin{bmatrix} \mathbf{m}_i \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{bi} \end{bmatrix} \mathbf{v}_i^2 \quad (2.81)$$

In this equation, the $\mathbf{I}_{3 \times 3}$ is the identity matrix. I_{bi} is the inertia of body i , inserting the v_i by Jacobian matrix form, the following expression can be found:

$$T(q, \dot{q}) = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{J}_i^T(q) \mathbf{M}_i \mathbf{J}_i(q) \dot{\mathbf{q}} = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{B}_i(q) \dot{\mathbf{q}} \quad (2.82)$$

Here we denote $\mathbf{B}_i(q) = \mathbf{J}_i^T(q) \mathbf{M}_i \mathbf{J}_i(q)$ as the mass and inertia matrix of body i , hence the kinetic energy of the linked rigid bodies can be described as:

$$T(q, \dot{q}) = \frac{1}{2} \dot{\mathbf{q}}^T \left(\sum_{i=1}^n \mathbf{B}_i(q) \right) \dot{\mathbf{q}} = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{B}(q) \dot{\mathbf{q}} \quad (2.83)$$

The Lagrangian equation is:

$$\frac{d}{dt} \left(\frac{\partial \mathbf{T}}{\partial \dot{\mathbf{q}}} \right) - \left(\frac{\partial \mathbf{T}}{\partial \mathbf{q}} - \frac{\partial \mathbf{V}}{\partial \mathbf{q}} \right) = \boldsymbol{\tau} \quad (2.84)$$

For each element in the equation above, the expressions can be described as follows:

$$\frac{d}{dt} \left(\frac{\partial \mathbf{T}}{\partial \dot{\mathbf{q}}} \right) = \frac{d}{dt} (\mathbf{B}(q) \dot{\mathbf{q}}) \quad (2.85)$$

For the second item in the Lagrangian equation:

$$\frac{\partial \mathbf{T}}{\partial \mathbf{q}} = \frac{1}{2} \dot{\mathbf{q}}^T \frac{\partial \mathbf{B}(q)}{\partial \mathbf{q}} \dot{\mathbf{q}} \quad (2.86)$$

The $g(q)$ is the gravity forces in generalized coordinate and $k(q)$ is the spring force in generalized coordinate. For the third item in the Lagrangian equation:

$$\frac{\partial \mathbf{V}}{\partial \mathbf{q}} = \mathbf{g}(q) + \mathbf{k}(q) \quad (2.87)$$

This part is the restoring forces, this including gravity force and spring force in the mechanical system. The partial differential of \mathbf{V} , which is the potential energy of the system, will result in restoring forces, such as gravity force and spring force in the mechanical system.

Combining all these above, we can get the Lagrangian equation of motion as follow:

$$\frac{d}{dt}(\mathbf{B}(q))\dot{\mathbf{q}} - \left(\frac{1}{2}\dot{\mathbf{q}}^T \frac{\partial \mathbf{B}}{\partial \mathbf{q}} \dot{\mathbf{q}} - g(q) - k(q)\right) = \tau \quad (2.88)$$

In this equation, τ is the external force inputs, $\mathbf{B}(\mathbf{q})\dot{\mathbf{q}}$ is the momentum of the system, named \mathbf{p} . So we can develop the relation ship between $\dot{\mathbf{q}}$ and \mathbf{p} as:

$$\mathbf{p} = \mathbf{B}(q)\dot{\mathbf{q}} \quad (2.89)$$

So for $\dot{\mathbf{q}}$:

$$\dot{\mathbf{q}} = \mathbf{B}^{-1}(q)\mathbf{p} \quad (2.90)$$

Hence equation (2.96) become:

$$\dot{\mathbf{p}} = \frac{1}{2}\dot{\mathbf{q}}^T \frac{\partial \mathbf{B}}{\partial \mathbf{q}} \dot{\mathbf{q}} - \mathbf{g}(q) - \mathbf{k}(q) + \tau = f_p(p, q) + \tau \quad (2.91)$$

The mass and inertia matrix is symmetric, as well as the inverse mass and inertia matrix, hence we can get the result: $(\mathbf{B}^{-1})^T = \mathbf{B}^T$.

Then the state space equation that can be easily implemented into the IC-field in bond-graph:

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{B}^{-1}(q)\mathbf{p} \\ \dot{\mathbf{p}} &= f_p(p, q) + \tau \end{aligned} \quad (2.92)$$

All these matrices, vectors and forces are in the generalized coordinates.

2.4 Bond Graph Theory

This section mainly introduces the basic elements and methodologies of bond-graph modelling approach in physical dynamic systems. Here should be noticed that bond-graph is a topological representation of the conceptual structure and should not be confused with spatial structure. Bond-graph fits the spectrum of graphical model which suited for the engineering systems with effects from multiple energy domain.

2.4.1 Basic Knowledge of Bond-graph

In this section, many of the statements are based on Wolfgang (2009), Pedersen and Engja (2010) and Rokseth (2014).

The interface between a bond and a multi-port node, is represented by effort $\mathbf{e}(t)$ and flow $\mathbf{f}(t)$, which are dynamically conjugated. Their product is called power $\mathbf{P}(t)$. These variables are called power conjugated, the bond is called a power bond, and

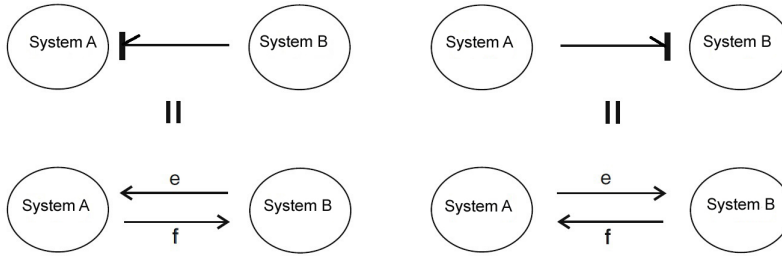


Figure 2.4: Power bond and its direction

the port to which it is connected is a power port. Bond graphs that only contain power bonds and signals are sometimes called true bond graphs as opposed to the larger class of pseudo-bond graphs in which the efforts and flows are considered to be only dynamically conjugated. The relation of power transmission based on effort and flow can be expressed as:

$$\mathbf{P}(t) = \mathbf{e}(t)\mathbf{f}(t) \quad (2.93)$$

A signal to a node is called a signal port. There are two types of signal ports: inputs and outputs. The full arrow of the signal is attached to an input and its open end to an output. The multi-port relates the efforts and flows at its ports fundamentally in such a way that each power port has a dependent and an independent (effort or flow) variable. These relations are called constitutive law and may also depend on the input signals of the multi-port. Output signals may depend on any of the independent variables of a multi-port. The Figure 2.4 shows the relations between power bonds and ports. This shows that bond-graph method also a energy or power based modelling method for dynamic systems. In the engineering dynamic systems, there are many elements or blocks that represent different physical means. For instance, elements for storing energy, which can be the spring in mechanical system and capacitor in electrical system; elements for dissipating energy, which can be damping in mechanical system and resistor in electrical system; elements for transforming energy, which can be a motor; or elements that can supply energy, which can be a engine in mechanical system and a battery in electrical system. As mentioned above, each elements have power ports, so the power in the system can be transmitted among these elements, ports and power bonds. In the figure 2.4, the direction of power bond can be found. In the left system, the positive power is defined same as the direction of arrow, which is from System *B* to System *A*.

Based on the two important variables in bond-graph modelling and integrate them when time varies, we can get the corresponding two new variables in bond-graph

modelling. The momentum $\mathbf{p}(t)$ and the displacement $\mathbf{q}(t)$, or more specifically, the generalized momentum and generalized displacement since this law can be used in the system from other disciplines as well. The expression is:

$$\begin{aligned}\mathbf{p}(t) &= \int_{t=0}^t dt + p_0 \\ \mathbf{q}(t) &= \int_{t=0}^t dt + q_0\end{aligned}\tag{2.94}$$

Of course, the energy can also be expressed here as the time integral of power that explained above:

$$E(t) = \int_0^t \mathbf{P}(t)dt = \int_0^t \mathbf{e}(t)\mathbf{f}(t)dt\tag{2.95}$$

From previous parts, the following states can be set:

$$\begin{aligned}\mathbf{e}dt &= d\mathbf{p} \\ \mathbf{f}dt &= d\mathbf{q}\end{aligned}\tag{2.96}$$

The flow can be the function of generalized momentum and the effort can be the function of generalized displacement, Hence we can get:

$$E(p) = \int_0^p \mathbf{f}(p)dp\tag{2.97}$$

And:

$$E(q) = \int_0^q \mathbf{e}(q)dq\tag{2.98}$$

Here the $E(p)$ represents kinetic energy in the system and $E(q)$ represents the potential energy of the system.

p_0 and q_0 are the initial values here. Therefore, until now, there are four variables in total that we need in order to model the system in bond-graph approach. These are effort \mathbf{e} , flow \mathbf{f} , generalized momentum \mathbf{p} and generalized displacement \mathbf{q} . For different systems and applications, following conclusion can be listed. The details of the elements in bond-graph modelling can be found in at Pedersen and Engja (2010). The key point in building a bond-graph is to identify the major components or subsystems, and bonds are introduced to connect power interaction between different elements, system and environment or among systems. This is based on describing the energetic properties of a system. Here we can go through some common used

Energy domain	Effort e	Flow f	Generalized momentum p	Generalized displacement q
Translational mechanics	Force F [N]	Velocity v [m/s]	Momentum p [Ns]	Displacement x [m]
Rotational mechanics	Angular moment M [Nm]	Angular velocity ω [rad/s]	Angular momentum p_ω [Nms]	Angle θ [rad]
Electro-magnetic domain	Voltage u [V]	Current i [A]	Linkage flux λ [Vs]	Charge q [As]
	Magneto motive force V [A]	Magnetic flux rate $\dot{\phi}$ [Wb/s]	–	Magnetic flux ϕ [Wb]
Hydraulic domain	Total pressure p [N/m ²]	Volume flow Q [m ³ /s]	Pressure momentum p_p [N/m ² s]	Volume V_C [m ³]
Thermo-dynamic	Temperature T [K]	Entropy flow \dot{S} [J/K/s]	–	Entropy S [J/K]
Chemical domain	Chemical potential μ [J/mole]	Molar flow \dot{N} [mole/s]	–	Molar mass N [mole]

Figure 2.5: Elements in different system (Source: Pedersen and Engja (2010))

and important elements.

First, there are four 1-port element, i.e. sources, capacitors, inertias and resisters. The sources means the element that can supply energy to the system, the power or energy reservoir is come from external of the system. There are two types of sources, which is the source of efforts, denote as \mathbf{S}_e and the source of flow, denote as \mathbf{S}_f . The capacitor is defined as store energy. For instance, in the marine crane modelling case, for the linked rigid bodies system, the energy can store as potential energy. The element is denote as \mathbf{C} and have the corresponding constitutive law $\mathbf{q} = \mathbf{C}\mathbf{e}$. The inertia element is another energy storage element, denote as \mathbf{I} , which represents the relationship between momentum \mathbf{p} and flow \mathbf{f} . The last one is the resistor element, it has the relationship between effort \mathbf{e} and flow \mathbf{f} . This is the only element that can set as power dissipation and energy consumption. The resistor element is denoted as \mathbf{R} and has the constitutive law $\mathbf{e} = \mathbf{R}\mathbf{f}$.

There are two ports elements as well. First is the transformer elements, defined as \mathbf{TF} . The function of transformer element is transfer efforts or flows from one port to another port based on some factors. It has the following relation:

$$\begin{aligned} \mathbf{e}_1 &= m\mathbf{e}_2 \\ m\mathbf{f}_1 &= \mathbf{f}_2 \end{aligned} \quad (2.99)$$

Another two ports element is called the gyrator, this element connects the effort with flow. This element denotes as \mathbf{GY} , has the constitutive law as:

$$\begin{aligned} \mathbf{e}_1 &= r\mathbf{f}_2 \\ r\mathbf{f}_1 &= \mathbf{e}_2 \end{aligned} \quad (2.100)$$

In the end, there are two junction elements which help build the frame of the bond-graph of a system. All these junction elements are considered ideal, which means neither store or dissipate energy. Hence, at certain time, the power into the junction must be equal to the power out of the junction.

First one is the 0-junction that defined as the common effort junction (to the bonds that connected with 0-junction). The constitutive law can be described as:

$$\begin{aligned} \mathbf{e}_1 &= \mathbf{e}_2 = \mathbf{e}_3 \\ \mathbf{f}_1 + \mathbf{f}_2 + \mathbf{f}_3 &= 0 \end{aligned} \quad (2.101)$$

The 0-junction represent the Kirchoff's current law in the real applications.

The other junction element is the 1-junction element, compare to the 0-junction,

1-junction has the constitutive law:

$$\begin{aligned} \mathbf{f}_1 &= \mathbf{f}_2 = \mathbf{f}_3 \\ \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3 &= 0 \end{aligned} \quad (2.102)$$

Now, all the basic elements are introduced and we can conclude them in a table as well.

2.4.2 Multi-port Fields

In the section above we defined and discussed the 1-port elements, 2-port elements and the junction elements. Modelling the marine crane system, which mainly the dynamics of rigid bodies systems in spatial motion. How to express the physical laws and energy relations in the vector field is very important. In addition, this can help simplify the bond-graph, making it more compact in both calculation and software implementation. However, some vector elements will make the bond-graph of the system not good for visualization. Some relations and interactions that can be easily found in graphical symbols will be lost here, but the fields are still more useful in modelling complex multi-port systems than the scalar elements, especially for the Lagrangian equation implement.

There are many multi-port elements that can deal with field vectors, such as I-field, C-field and multi-port transformer etc. Here we mainly discuss about IC-field which will be used a lot in this thesis due to its closed link with Hamiltonian-Lagrangian equation. In addition, IC-field can partly avoid large amount of algebraic loops and differential causality in complex system.

Before we discuss IC-field, we need to study the I-field and C-field. A C-field explain the relationship between efforts and displacements related to n-port, which means has n power bonds connect to this C-field. The constitutive law of C-field is:

$$\begin{aligned} \mathbf{e} &= \phi_C^{-1}(q) \\ \mathbf{q} &= \phi_C(e) \end{aligned} \quad (2.103)$$

In this case, the $\mathbf{e}=[e_1, e_2, \dots, e_{n-1}, e_n]^T$, $\mathbf{q}=[q_1, q_2, \dots, q_{n-1}, q_n]^T$. In the C-field, it can have both integral and differential causality. The constitutive law can be described as:

$$\mathbf{e}_i = \phi_{C_i}^{-1}(q_1, q_2, \dots, q_{n-1}, q_n), \quad i = 1, 2, \dots, n \quad (\text{integral causality}) \quad (2.104)$$

$$\mathbf{q}_i = \phi_{C_i}(e_1, e_2, \dots, e_{n-1}, e_n), \quad i = 1, 2, \dots, n \quad (\text{differential causality}) \quad (2.105)$$

Bond graph element	Constitutive relations
$S_e \xrightarrow{e} \searrow$	$e = e(t)$
$S_f \xrightarrow{f} \searrow$	$f = f(t)$
$\xrightarrow{e} \searrow C$ f	General : $q = \Phi_C(e)$ Linear : $q = Ce$
$\xrightarrow{e} \searrow I$ f	General : $p = \Phi_I(f)$ Linear : $p = If$
$\xrightarrow{e} \searrow R$ f	General : $e = \Phi_R(f)$ Linear : $e = Rf$
$\xrightarrow{e_1} \searrow \xrightarrow{e_2} \searrow$ f_1 TF f_2	$e_1 = me_2$ $mf_1 = f_2$
$\xrightarrow{e_1} \searrow \xrightarrow{e_2} \searrow$ f_1 GY f_2	$e_1 = f_2r$ $f_1r = e_2$
$\xrightarrow{e_1} \searrow \xrightarrow{e_2} \searrow$ f_1 0 f_2 $e_3 \uparrow f_3$	$e_1 = e_2 = e_3$ $f_1 - f_2 - f_3 = 0$
$\xrightarrow{e_1} \searrow \xrightarrow{e_2} \searrow$ f_1 I f_2 $e_3 \uparrow f_3$	$f_1 = f_2 = f_3$ $e_1 - e_2 - e_3 = 0$

Figure 2.6: Element in bond-graph (Source: Pedersen and Engja (2010))

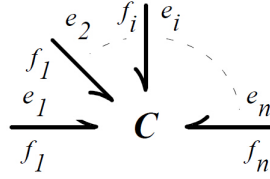


Figure 2.7: N-port C-field

The I-field is quite similar to the C-field here, we will not extend the discussion but only give out the corresponding expressions here. The I-field described the relation between flow and momentum in the n-port system. Similarly, the constitutive law is given as:

$$\begin{aligned} \mathbf{f} &= \phi_I^{-1}(p) \\ \mathbf{p} &= \phi_I(f) \end{aligned} \quad (2.106)$$

Where the $f = [f_1, f_2, \dots, f_{n-1}, f_n]$, $p = [p_1, p_2, \dots, p_{n-1}, p_n]$. Same with the C-field, the causality can be both integral and differential. The following expression will show the details of it:

$$\mathbf{f}_i = \phi_{I_i}^{-1}(p_1, p_2, \dots, p_{n-1}, p_n) \quad i = 1, 2, \dots, n \quad (\text{integral causality}) \quad (2.107)$$

$$\mathbf{p}_i = \phi_{I_i}(f_1, f_2, \dots, f_{n-1}, f_n) \quad i = 1, 2, \dots, n \quad (\text{differential causality}) \quad (2.108)$$

Then we can discuss the application of IC-field in the Hamiltonian-Lagrangian equations and mechanics. If we express the relations between flow and effort, the description can be:

$$\begin{aligned} \mathbf{e} &= \mathbf{f}(\mathbf{p}, \mathbf{q}) \\ \mathbf{f} &= \mathbf{g}(\mathbf{p}, \mathbf{q}) \end{aligned} \quad (2.109)$$

In many cases it is very convenient to combine the I-field and C-field, which is IC-field. Recall the Hamiltonian-Lagrangian equations in the previous section. The expression for kinetic energy $T(q, \dot{q})$ and potential energy $V(q)$ are the function of N-th displacement in n -th generalized coordinates. They can be expressed as $\mathbf{q} = [q_1, q_2, \dots, q_N]^T$, and corresponding generalized momentum is $\mathbf{p} = [p_1, p_2, \dots, p_N]^T$. The the expression is (German and Delarmelina (2007)):

$$\mathbf{p}_i = \frac{\partial \mathbf{T}}{\partial \dot{\mathbf{q}}_i} \quad \text{where } i = 1, 2, \dots, N - 1, N \quad (2.110)$$

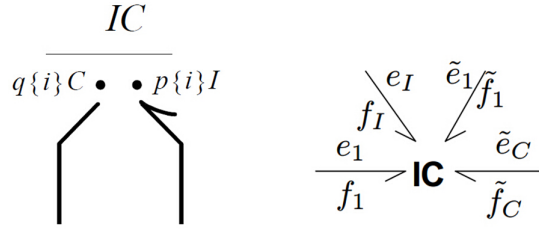


Figure 2.8: N-port IC-field

If we insert this into the Lagrangian equation of motion, we can get:

$$\dot{\mathbf{p}} = \mathbf{e}'(\mathbf{q}, \dot{\mathbf{q}}, t) + E \quad (2.111)$$

Where the $\mathbf{e}' = (\mathbf{q}, \dot{\mathbf{q}}, t)$ is equal to $\frac{\partial \mathbf{T}}{\partial \dot{\mathbf{q}}_i} - \frac{\partial \mathbf{V}}{\partial \mathbf{q}_i}$. Based on the expression of generalized momentum \mathbf{p}_i , we can find the expression for \mathbf{q} , the generalized displacement in the dynamic system.

$$\dot{\mathbf{q}} = \mathbf{M}^{-1}(q, t)(p - a) \quad (2.112)$$

Where the a is caused by the time varied velocity sources. The two equations can be implemented into a bond-graph model by IC-field.

Chapter 3

Basic Models Studies

In this chapter, the general purposes and methods of developing the Offshore Ship Operation Model(OSOM) platform will be explained. Based on the theories that introduced in the previous section in this chapter, this section will help the readers to catch the picture of the OSOM platform, which is very important to this master thesis. In addition, the basic knowledge about the offshore crane operation will be introduced, which is a very important background to both this master thesis and the OSOM platform.

3.1 Offshore Crane Operation

The marine operation plays a very important role in offshore construction, subsea installation and many kinds of maintenance work related to ocean engineering activities.

There are many kinds of marine operations nowadays, which can be general classified as two categories. The first one is topside marine operations, which including load transfer operations, lifting, towing, offshore installation and transit of mobile offshore unit. The second one is subsea marine operations, which including lifting, installation, IMOR(Inspection, maintenance, operation and repair of subsea equipment), well intervention and pipe laying. In this master thesis, the model that related to lifting operation by marine crane on DP vessel is the main topic.

The marine operations is facing a lot of challenges today. For instance, operation in deep waters and in arctic areas have more strict requirements to cost, operation

efficiency and safety. Based on these, understanding the details of system behaviour in challenging environments such as waves, winds, currents and extreme temperature are very important.

In both offshore installation and subsea installation by lifting, the object is lifted off, upended and set by crane that mounted on the OSV or PSV. In subsea lifting particularly, there are ROV launching and retrieval, pipelines installation and risers installation operation. The lifting phases are:

1. On board lift-off
2. Lifting objects through splash zone
3. Lowering objects in deep water
4. On bottom lading/installation

In these process, the vessel motions, lift dynamics and hydrodynamic loads, crane heave compensation systems and capabilities are very important topics that need to be taken into consideration.

In lifting operation, heavy lifting and light lifting have some differences on the effect of interacting between crane and vessel. For heavy lifting operation which involves a heavy lifting crane vessel or large barge, the dynamic and hydrodynamic interaction between the object and the crane vessel need to be taken into consideration. For light lifting, the vessel only slightly affected by the lifted object. The heave compensation is often used in light lifting operation, such as Active Heave Compensation Systems(AHC) and Passive Heave Compensation Systems(PHC). In this master thesis, the general dynamics of marine crane will be introduced. Then, the author focus more on the sub-systems such as hydraulic systems of marine crane which proves to be very important in modelling and simulation the crane-vessel system.

3.2 OSOM Platform

The purpose of the Offshore Ship Operation Model platform is to build a well developed and enhanced mathematical model of crane-vessel model by using the bond graph approach. This model platform is original developed by Rokseth and Skjong (2015). My scope of work in this part is to enhance the OSOM platform, such as 3D visualized the simulation results, modelling and simulate the sub-systems of marine crane system mounted on the DP vessel. Here the sub-systems is mainly

about hydraulic system of marine crane. The OSOM platform contains the dynamic properties of marine vessel and the equipment that mount on the vessel. In addition, the environment loads such as winds, currents, waves and working loads such as lifting objects in marine operations can be interfaced and interconnected with the dynamics of vessel-crane model. Developing the OSOM platform can contribute a lot for the engineering application. For instance, planning and testing the marine operation procedures and choosing weather windows. Besides that, this platform can help to develop a enhanced model based controller. By enhancing the 3D visualization, one can find a method to communicate between simulation software and visualization software, hence a offshore crane operation simulator can be designed. The simulator can reduce the cost of personnel training and increase the reliability of operation that under unknown environments and weather conditions.

The OSOM platform is developed by using bond graph approach. The bond graph modelling method is more visible for readers and much easier than the conventional mathematical modelling method to connect the different sub-systems and sub-models. In this process, the users or can control the complexity and detailed level of the basic model that enhanced by sub-systems or sub-models that users added in order to fulfill different requirements. In the OSOM platform, these sub-systems or sub-models including hydrodynamic damping, environment loads such as waves loads and currents loads. In addition, the actuator systems like DP vessel thrusters and hydraulic motors for the marine crane are included.

As described in the previous section in this chapter, the dynamic equations for the crane and the DP vessel will be developed by the Lagrangian method with quasi-coordinates. The details of system dynamics development and implementation can be found in the master thesis Rokseth (2014) and paper Rokseth and Skjong (2015). Please notice that only the dynamics that related to the kinetic energy will be developed by Lagrangian method. The reason is that by using Lagrangian method, the dynamics of the system can be easily implemented into bond graph approach. However, the dynamics related to the potential energy can be implement into bond graph directly. These dynamics equations that developed from Lagrangian equations will be put into the IC element in 20-sim software.

The model is a DP vessel with heavy duty crane that performs subsea lifting and installation operations. The crane has three degree of freedom and lower pair joint, which means only has one degree of freedom per joint. In the OSOM platform, the basic crane-vessel model has been extended. The sub-systems or sub-models are interfaced to the basic dynamic model of crane-vessel. When modelling and simulating the sub-systems, the OSOM platform can demonstrate how the sub-systems and sub-models can be implemented by bond graph approach and connect efficiently

with basic model. On the other hand, how the sub-systems and sub-models can effect on the dynamics of basic model is demonstrated. These sub-systems and sub-models including the vessel restoring force and gravity force, crane gravity force, wave excitation forces, added mass, hydrodynamic damping, thruster system and DP control system, actuators for the crane and crane motion control system, and finally, the hydraulic winch and wire model that developed in the master thesis Skjong (2014).

3.2.1 Marine Vessel Dynamics

In the previous section in this chapter, the basic theories about kinematics, Lagrangian mechanics and system of linked rigid bodies have been discussed. In this subsection, we will look through the dynamics of marine vessel, where a state space model will be explained. Then, the feasible method to connect the dynamic of marine vessel and dynamics of marine crane will be introduced in the next subsection.

Recall the main advantages of using Lagrangian method is this method is easier to develop the state space equation and also easier to implement into the bond graph than the traditional state space model. This method is not only used for the linked rigid bodies system but also the vessel dynamics.

The position of DP vessel can be expressed by $\mathbf{r}_{b/0}^0$, here 0 represent the inertial reference frame, and b represent the vessel body frame. The vector express the position from the original of vessel body fixed frame to the original of inertial reference frame that expressed in term of inertial reference frame. The orientation of the vessel is expressed in the Euler angles $\Theta = [\phi, \theta, \psi]^T$. The ϕ is the angle that vessel rotates about x-axis. Similarly, the θ and ψ is the angle that the vessel rotates about y-axis and z-axis respectively. As explained in the previous section about the coordinate, after the rotation about the Euler angle, the vessel body fixed coordinate will have the same orientation of the inertial reference frame. Hence, the angular velocity of the DP vessel can be expressed in body fixed frame as

$$\boldsymbol{\omega}_{b/0}^b = \mathbf{i}_b \dot{\phi} + \mathbf{j}'_b \dot{\theta} + \mathbf{k}''_b \dot{\psi} = \mathbf{T}_{\Theta}^{-1}(\Theta) \dot{\Theta} \quad (3.1)$$

In this equation, the \mathbf{i}_b is the unit normal vector along the x-axis, \mathbf{j}'_b is the unit normal vector along y-axis after the rotation of ϕ , same as the \mathbf{k}'' , is the unit normal axis along the z-axis. With the transformation defined and explained in the previous section in this chapter about the position, orientation of rigid bodies system and rotation matrix, we can get the transformation matrix that can transfer the vessel body fixed reference frame to the inertial reference frame

$$\mathbf{R}_b^0 = \mathbf{R}_z(\psi) \mathbf{R}_y(\theta) \mathbf{R}_x(\phi) \quad (3.2)$$

The \mathbf{R}_z , $\mathbf{R}_y(\theta)$ and $\mathbf{R}_x(\phi)$ are given in the section 2.1, which will not be explained in details here. With the properties of

$$(\mathbf{R}_b^0)^{-1} = (\mathbf{R}_0^b)^T = \mathbf{R}_0^b \quad (3.3)$$

Similar with the linked rigid bodies developed before, the kinetic energy of the vessel is

$$T = \frac{1}{2}((\mathbf{v}_{cg/0}^b)^T \mathbf{M} \mathbf{v}_{cg/0} + (\mathbf{w}_{b/0}^b)^T \mathbf{I}_g \boldsymbol{\theta}_{b/0}^0) \quad (3.4)$$

The $\mathbf{M} = m\mathbf{I}_{3 \times 3}$ is the mass of the vessel times the corresponding identity matrix. \mathbf{I}_g is the DP vessel inertial. The \mathbf{M} matrix and \mathbf{I}_g matrix times the quadratic of linear velocity of central of gravity of vessel $\mathbf{v}_{cg/0}$ and angular velocity $\boldsymbol{\theta}_{b/0}^0$ respectively.

In Lagrangian method, the kinetic energy should be expressed in generalized coordinate and the time rate. The generalized coordinates can uniquely define the position and orientation of the DP vessel, in the OSOM which is chosen as

$$\mathbf{q} = [(\mathbf{r}_{b/0}^0)^T, \boldsymbol{\Theta}^T]^T \quad (3.5)$$

With the choice here, the linear velocity now can expressed as

$$\begin{aligned} \mathbf{v}_{cg/0}^b &= \mathbf{v}_{b/0}^b + \boldsymbol{\theta}_{b/0}^b \times \mathbf{r}_{cg/b}^b \\ &= \mathbf{R}_0^b(\boldsymbol{\Theta})\dot{\mathbf{r}}_{b/0}^0 + \mathbf{T}_{\boldsymbol{\Theta}}^{-1}(\boldsymbol{\Theta})\dot{\boldsymbol{\Theta}} \times \mathbf{r}_{cg/b}^b \end{aligned} \quad (3.6)$$

Here the vector $\mathbf{r}_{cg/b}^b$ represents the the vector from the original of the vessel body fixed reference frame to the central of mass of the vessel. Now we can find that the kinetic energy of the vessel can be expressed as the function of generalized coordinates and the derivative of generalized coordinates. In order to avoid the heavy calculation and difficult in implementation in bond graph. So here we want to replace the $\dot{\mathbf{q}}$ by using the quasi coordinate. The quasi coordinate has been briefly explained in the section 2.2. From the equation here we can define a set of generalized speeds as the time derivatives of the generalized coordinates, which can also be defined as the linear combinations of the time derivatives of the generalized coordinates. In quasi coordinate, we only need the time derivative of the quantities that have actual physical mean. However, the integral of quasi velocity has no physical interpretation. Therefore, there is no actual position that linked with the quasi coordinate. As explained before, the equation of motion here will only contain the time derivative of the corresponding coordinate.

In the vessel crane system, the quasi coordinate is used here to describe the velocity state of the system. For instance, for the rigid body motion in three dimensional Euclidean space. A transformation of a rigid body in this space is non-Euclidean because we need to include rotational motion in three degree of freedom.(Pål et al. (2012)) For more details about quasi coordinate can be found in Ginsberg (1998) and

Pål et al. (2012) etc.

Based on the upon discussion, we can do some modification to the original velocity expression that depend on $\dot{\mathbf{q}}$, set it into quasi coordinates as

$$\boldsymbol{\omega} = \begin{bmatrix} v_{b/0}^b \\ \boldsymbol{\omega}_{b/0}^b \end{bmatrix} = \begin{bmatrix} \mathbf{R}_0^b & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{T}_{\Theta}^{-1} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{r}}_{b/0}^0 \\ \dot{\Theta} \end{bmatrix} = \boldsymbol{\alpha}^T \dot{\mathbf{q}} \quad (3.7)$$

From this equation, we can find that the equation of motion is depend on the linear velocity and angular velocity in the body fixed frame instead of depended on the linear velocity in the inertial reference frame and the rates of Euler angle. By inverting this equation

$$\dot{\mathbf{q}} = \boldsymbol{\beta} \boldsymbol{\omega} \quad (3.8)$$

We can get

$$\boldsymbol{\beta} = (\boldsymbol{\alpha}^T)^{-1} = \begin{bmatrix} \mathbf{R}_b^0 & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{T}_{\Theta}^{-1} \end{bmatrix} \quad (3.9)$$

Then we can change the expression $T(\mathbf{q}, \dot{\mathbf{q}})$ to $T(\mathbf{q}, \boldsymbol{\beta} \boldsymbol{\omega})$, which equals to $\bar{T}(\mathbf{q}, \boldsymbol{\omega})$. In this expression, the linear velocity of the DP vessel in central of mass now can be expressed in the quasi coordinate, hence the equation

$$\begin{aligned} \mathbf{v}_{cg/0}^b &= \mathbf{v}_{b/0}^b + \mathbf{w}_{b/0}^b \times \mathbf{r}_{cg/b}^b \\ &= \mathbf{R}_0^b(\Theta) \dot{\mathbf{r}}_{b/0}^0 + \mathbf{T}_{\Theta}^{-1}(\Theta) \dot{\Theta} \times \mathbf{r}_{cg/b}^b \end{aligned} \quad (3.10)$$

Now can be expressed as

$$\mathbf{v}_{cg/0}^b = [\mathbf{I}_{3 \times 3}, \mathbf{i}_b \times \mathbf{r}_{cg/0}^b, \mathbf{j}_b \times \mathbf{r}_{cg/0}^b, \mathbf{k}_b \times \mathbf{r}_{cg/0}^b] \boldsymbol{\omega} = \mathbf{J}_b^{\omega} \boldsymbol{\omega} \quad (3.11)$$

The \mathbf{J}_b^v is the geometric Jacobian matrix related to the linear velocity of DP vessel in the center of mass. Hence the general velocity vector now is $\mathbf{v}_b = [(\mathbf{v}_{cg}^b)^T, (\boldsymbol{\omega}_{cg/0}^b)^T]^T$. It is quite clear to see that the linear velocity in the center of mass of the vessel and the angular velocity of in body fixed frame. Using the geometric Jacobian matrix, we can simplify the expression as

$$\mathbf{v}_b = \begin{bmatrix} \mathbf{J}_b^v \\ \mathbf{J}_b^{\omega} \end{bmatrix} \boldsymbol{\omega} = \mathbf{J}_b \boldsymbol{\omega} \quad (3.12)$$

Collecting all these equations, the kinetic energy expression in the quasi coordinates can be expressed as

$$\begin{aligned} \bar{T}_b(\mathbf{q}, \boldsymbol{\omega}) &= \frac{1}{2} \boldsymbol{\omega}^T \mathbf{J}_b^T \begin{bmatrix} \mathbf{M} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_g \end{bmatrix} \boldsymbol{\omega} \mathbf{J}_b \\ &= \frac{1}{2} \boldsymbol{\omega}^T \mathbf{B}_b \boldsymbol{\omega} \end{aligned} \quad (3.13)$$

The \mathbf{B}_b is the vessel mass inertial matrix.

Now, we can look into the equation of motion for marine vessel. As explained before, the equation of motion that expressed by the Lagrangian method that in term of the generalized coordinates and the rates of generalized coordinates is

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial T}{\partial \mathbf{q}} = \boldsymbol{\tau} \quad (3.14)$$

Here the $\boldsymbol{\tau}$ is the generalized force vector. The potential energy will not be discussed here because the potential energy related equations will not be developed in quasi coordinate. Now the kinetic energy is implemented in the equation of motion in terms of quasi coordinate. The equation of motion now is

$$\frac{d}{dt} \left(\frac{\partial \bar{T}}{\partial \boldsymbol{\omega}} \right) + \beta^T \gamma \frac{\partial \bar{T}}{\partial \boldsymbol{\omega}} - \beta^T \frac{\partial \bar{T}}{\partial \mathbf{q}} = \beta^T \boldsymbol{\tau} \quad (3.15)$$

Then, with the similar steps to develop the Lagrangian equation in the linked rigid bodies system and the implementation to the IC field in bond graph, we can set the equation of motion in quasi coordinate as

$$\dot{\mathbf{p}} = \frac{d}{dt} \left(\frac{\partial \bar{T}}{\partial \boldsymbol{\omega}} \right) \quad (3.16)$$

Here \mathbf{p} is the moment of the quasi coordinate, which is expressed in vessel body fixed reference frame. From the previous equation, we can find

$$\frac{\partial \bar{T}}{\partial \boldsymbol{\omega}} = \mathbf{B}_b \boldsymbol{\omega} \quad (3.17)$$

And inserting the $\mathbf{p} = \frac{\partial \bar{T}}{\partial \boldsymbol{\omega}}$, the quasi coordinate then can be expressed as

$$\boldsymbol{\omega} = \mathbf{B}^{-1} \mathbf{p} \quad (3.18)$$

Then we recall the equation of motion that developed in terms of quasi coordinate. By inserting the equation above, we can get the expression about the time derivative of momentum of the quasi coordinate

$$\begin{aligned} \dot{\mathbf{p}} &= -\beta^T \gamma \frac{\partial \bar{T}}{\partial \boldsymbol{\omega}} + \beta^T \frac{\partial \bar{T}}{\partial \mathbf{q}} + \beta^T \boldsymbol{\tau} \\ &= \mathbf{f}_p(\mathbf{q}, \boldsymbol{\omega}) + \beta^T \boldsymbol{\tau} \end{aligned} \quad (3.19)$$

Combine and conclude the state space equations that developed so far for the marine vessel dynamics

$$\boldsymbol{\omega} = \mathbf{B}_b^{-1} \mathbf{p} \quad (3.20)$$

$$\dot{\mathbf{q}} = \beta \boldsymbol{\omega} \quad (3.21)$$

Table 3.1: Parameters in OSOM simulation example

Setting Parameters		Environment Parameters	
Name	Value	Name	Value
Ship initial heading	0.0 rad	Spectrum peak period(T)	8 s
Winch load	50 t	Significant wave height(H)	6 m
Winch wire length	400 m	Spectrum shape parameter(γ)	3 s
Simulation time	180 s	Lowest wave period component	3 s
Length of ship	107 m	Largest wave period component	21 s
Width of ship	22 m	Current speed	0.2 m/s
Mass of ship	2350 t		
Power saturation thrusters	3.5 MW		

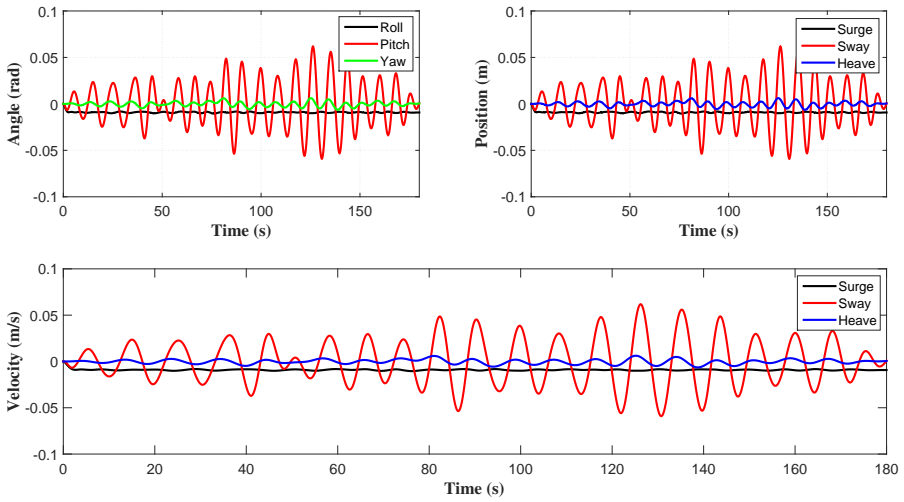


Figure 3.1: Marine vessel dynamic simulation results

$$\dot{\mathbf{p}} = \mathbf{f}_p(\mathbf{q}, \mathbf{w}) + \boldsymbol{\beta}^T \boldsymbol{\tau} \quad (3.22)$$

In table 3.1 the initial states and parameters setting of the model are provided, which includes system parameters and environment parameters. In figure 3.1, the angles of roll, pitch, yaw, positions of x , y , z and velocities of heave roll and pitch of the DP

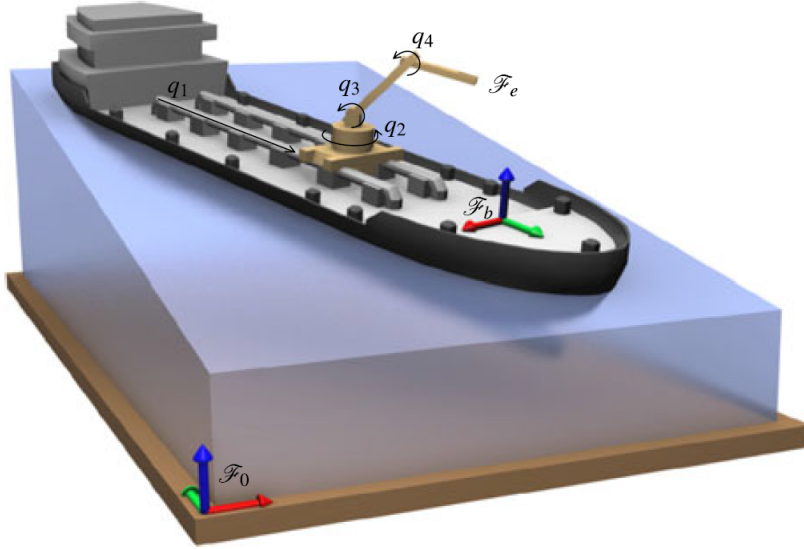


Figure 3.2: Vessel crane system (Source:Pål et al. (2012))

vessel are shown. Please notice that the heading, or yaw angle is set to 0.0 as initial value. The forces that act on the vessel also shown. Here we simplified the problem, focus on the forces that influence the heave, roll and yaw.

3.2.2 Combine the Vessel Dynamics and Marine Crane Dynamics

Now we can combine the two systems namely marine crane dynamics and marine vessel dynamics together to form the basic platform of OSOM, which other sub-systems and sub-models can be interfaced with. For the basic dynamic of marine crane that already explained in the chapter 2.3, which mainly explains the system of linked rigid bodies. Therefore, the basic theories and details will not be repeated here.

The similar methods and approaches are used to build the mathematic model of marine crane and marine vessel, which both are the motion of rigid bodies in space. We can easily and conveniently combine the dynamic of marine vessel and dynamic of marine crane in the IC field in bond graph approach since the format of the state space equations are the same. The very general and typical marine vessel crane system is shown in the figure. When including crane dynamics, with the rotation matrix, we can express all local coordinates in the center of mass and the location of the joints based on the DP vessel body fixed reference frame. Therefore, the dynamic

of marine crane depend much on the dynamic of marine vessel, which explained in the previous section.

By connecting the marine crane on marine vessel, we can find the position of center of mass for all the linked rigid bodies related to the joints or coordinates we want. Since the marine crane is mounted on the marine vessel now, we need to find the linear velocity of center of mass of linked rigid body i that comes from the linear velocity of marine vessel. Which can be expressed as

$$\mathbf{v}_{cgi/0}^{vb/0} = \mathbf{v}_{b/0}^b = \mathbf{I}_{3 \times 3} \mathbf{v}_{b/0}^b = \mathbf{J}_{vb}^{vcgi} \mathbf{v}_{b/0}^b \quad (3.23)$$

This equation explains where the velocity component comes from. Similar with the linear velocity, the component that comes from the angular velocity is

$$\begin{aligned} \mathbf{v}_{cgi/0}^{\omega b/0} &= \boldsymbol{\omega}_{b/0}^b \times \mathbf{r}_{cgi/b}^b \\ &= [\mathbf{i}_b \times \mathbf{r}_{cgi/b}^b, \mathbf{j}_b \times \mathbf{c}g\mathbf{i}/\mathbf{b}^b, \mathbf{k}_b \times \mathbf{r}_{cgi}^b] \mathbf{r}_{cgi/b}^b \\ &= \mathbf{J}_{wb}^{vcgi} \boldsymbol{\omega}_{b/0}^b \end{aligned} \quad (3.24)$$

Please notice that in the equation, the components of the velocity that come from the previous linked rigid bodies relative to the link i , i.e. the components of linear velocity to link i come from the rate of joints j , here $j \leq i$. There are differences between revolute joint and prismatic joint as well. For the revolute joint we have $(\mathbf{e}_j^b \times \mathbf{r}_{cgi/j}^b) \dot{q}_{ej}$ and the prismatic joint can be described as $\mathbf{e}_j^b \dot{q}_{jp}$. The j here is the joint j that can contribute to the velocity of joint i . Hence the linear velocity of the center of mass of body i can be expressed as following equations in the body fixed reference frame

$$\mathbf{v}_{cgi/0}^b = \mathbf{J}_i^v(\mathbf{q}) \boldsymbol{\omega} \quad (3.25)$$

Similarly, the angular velocity components that come from different parts will be discussed.

The details of how to develop the dynamics of marine vessel crane system can be found on Pål et al. (2012) and the paper Rokseth and Skjong (2015). Where they both discuss the steps and methods that used here to include the dynamic properties of both marine vessel and marine crane. The linear velocity of marine vessel will not contribute any to the angular velocity of marine crane. The component from the angular velocity can be expressed as $\boldsymbol{\omega}_i^{b/0} = \mathbf{J}_{\omega}^{\omega i} \boldsymbol{\omega}_{b/0}^b$. Similarly to the linear velocity of the vessel that will not contribute to the angular velocity of crane. The joints that are prismatic of linked rigid bodies or more specific, the marine crane, will also not contribute to the angular velocity of joint i in the crane when $i < j$.

The overall angular velocity of joint i can be described similarly with the overall linear velocity of joint i as

$$\boldsymbol{\omega}_{i/0}^b = \mathbf{J}_i^w(\mathbf{q}) \boldsymbol{\omega} \quad (3.26)$$

Now we can collect the linear velocity and angular velocity together as $\mathbf{v}_i = [(\mathbf{v}_{cg_i/0}^b)^T, (\boldsymbol{\omega}_{i/0}^b)^T]^T$. The corresponding Jacobian matrix is

$$\mathbf{J}_i(\mathbf{q}) = \begin{bmatrix} \mathbf{J}_i^v(\mathbf{q}) \\ \mathbf{J}_i^\omega(\mathbf{q}) \end{bmatrix} \quad (3.27)$$

By doing this, we can connect the general velocity with the quasi coordinates as

$$\mathbf{v}_i = \mathbf{J}_i(\mathbf{q})\boldsymbol{\omega} \quad (3.28)$$

The the kinetic energy can be calculated the kinetic energy of the marine crane. Same as the marine vessel, we leave the potential energy for now. The kinetic energy for the linked rigid body i can describe as

$$\bar{T}(\mathbf{q}, \boldsymbol{\omega}) = \frac{1}{2}\boldsymbol{\omega}^T \mathbf{J}_i^T(\mathbf{q}) \begin{bmatrix} \mathbf{M}_i & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{i}_b \end{bmatrix} \mathbf{J}_i(\mathbf{q})\boldsymbol{\omega} \quad (3.29)$$

Please notice that the generalized coordinates and quasi-coordinates are include crane joint now. Here the the mass matrix \mathbf{M}_i is for calculate the kinetic energy that comes from the linear velocity and the inertial matrix is for calculating the kinetic energy of linked rigid body i that comes from angular velocity. Here we can combine the mass-inertial matrix with the Jacobian matrix together as $\mathbf{B}_i(\mathbf{q})$.

$$\mathbf{B}_i(\mathbf{q}) = \mathbf{J}_i^T(\mathbf{q}) \begin{bmatrix} \mathbf{M}_i & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{i}_b \end{bmatrix} \mathbf{J}_i(\mathbf{q}) \quad (3.30)$$

To conclude, now we can combine the kinetic energy that from the marine vessel and the marine crane

$$\begin{aligned} \bar{T}(\mathbf{q}, \boldsymbol{\omega}) &= \mathbf{T}_b(\mathbf{q}, \boldsymbol{\omega}) + \sum_{i=1}^k (\bar{T}_i(\mathbf{q}, \boldsymbol{\omega})) \\ &= \frac{1}{2}\boldsymbol{\omega} \mathbf{B}(\mathbf{q})\boldsymbol{\omega} \end{aligned} \quad (3.31)$$

Now we can find the overall state space equations for the whole system, which is exactly the same format for the marine vessel and marine crane that developed separately before. These state space equations are also explained in the bond graph implementation, which more specifically, to implement these dynamic state equations related with mass-inertial matrix in the IC field. The final space state equations for the vessel-crane system can be expressed as

$$\boldsymbol{\omega} = \mathbf{B}^{-1}\mathbf{p} \quad (3.32)$$

$$\dot{\mathbf{q}} = \boldsymbol{\beta}\boldsymbol{\omega} \quad (3.33)$$

$$\dot{\mathbf{p}} = \mathbf{f}_p(\mathbf{q}, \boldsymbol{\omega}) + \boldsymbol{\beta}^T \boldsymbol{\tau} \quad (3.34)$$

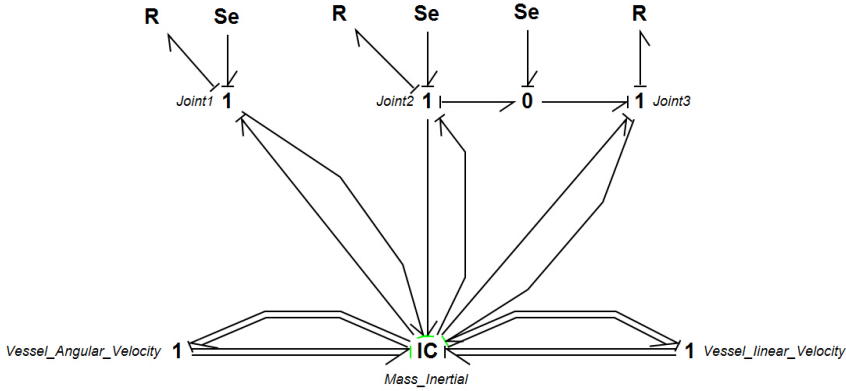


Figure 3.3: Basic model IC field implementation

3.2.3 Bond Graph Implementation of OSOM Platform

The dynamic equations we create so far are suitable for implement in bond graph method, especially the IC field in bond graph as shown in figure 3.3. Even the visual connection between different components in IC field is not obvious as normal bond graph that connects the I element and C element. This approach is still selected compare to other methods because it simplifies the structure and implementation work a lot. After building the basic model, several sub-systems and the interface between sub-systems and basic model will be discussed.

The state space equations we developed so far contain \mathbf{q} , the generalized coordinates, \mathbf{p} the momentum of the system and $\boldsymbol{\omega}$ the quasi coordinates. From the IC field that discussed in chapter 2 and the figure 3.3 shows the basic structure and concept. We can set the effort $\mathbf{e}_1 = \dot{\mathbf{p}}_1$ and the flow $\mathbf{f}_2 = \boldsymbol{\omega}_2$ for joint 1, which are the input value for the IC field. Same values assigned to the joint 2 and joint 3 in this figure. Noticing that the Se elements here indicates the torque that acts on each crane joint in order to control the motion of the crane, which is based on some references and control signals. The R elements here represents the friction model in each joint, which may depend on the load and angular velocity of each joint of the marine crane. Recall the properties of the 1 junction, which has equal flow in each bond that connected with and the sum of effort outputs should be equal to the sum of effort inputs. The constitutive relations can be described as

$$\boldsymbol{\omega}_i = \mathbf{B}^{-1}(\mathbf{q})\mathbf{p}_i \quad (3.35)$$

$$\dot{\mathbf{p}}_i = \mathbf{f}_p(\mathbf{q}, \boldsymbol{\omega}_i) \quad (3.36)$$

Now we can find the quasi coordinates that developed in this model are expressed in the two 1 junctions, which represent the linear velocity of the marine vessel and the angular velocity of marine vessel. The dynamics in the IC field, which are the state equations mentioned above.

3.2.4 Sub-models Connect with OSOM Platform

The figure 3.4 shows the overview of the OSOM platform. From this figure we can see the IC field that explained in the previous section as well as other sub-models. Apart from the 1 junctions which represent the three joints of marine crane, linear velocity and angular velocity of marine vessel respectively. There are many other sub-models or sub-systems included here.

Environmental Loads

This part has been shown in the figure 3.4. The box named "waves" and "current" is the sub-models that calculating the environment loads act on the DP vessel, and the marine crane as well. In addition, there are other elements that not include the environmental loads but also influence the system dynamics. For instance restoring forces and friction forces that act on the DP vessel. These loads will influence the dynamics of the marine vessel significantly.

The restoring forces or torques are generated by the weight of marine vessel and the buoyancy force act on the vessel body. The restoring forces will also be determined by the linear motion of the vessel, hence it only need to be connected with the 1 junction that represents the linear velocity of vessel in the bond graph. Since the restoring forces are in terms of inertial reference frame, we need to transfer it from the body fixed coordinates. The *MTF* elements are used here, the transformer module is $\mathbf{r}_{b/0}^0$. By doing this, we can interface the restoring forces with the basic model in OSOM.

Now we can look at the restoring torque, which should be connected to the 1 junction that represents the vessel angular velocity and in terms of body fixed reference frame. Other forces such as gravity forces of the linked crane bodies and the vessel, buoyancy force etc. are all shown in the figure 3.4. The details of them can be found at the paper Rokseth and Skjong (2015). The theories that used here to implement the effect of waves and current can be found at Faltinsen (1990) and Fossen (2011). The detail of waves excitation force, current forces, added mass and hydrodynamic damping that influence the dynamic of vessel will be discussed in details in the chapter 4 since they are very important in modelling the wave carpet and vessel motion.

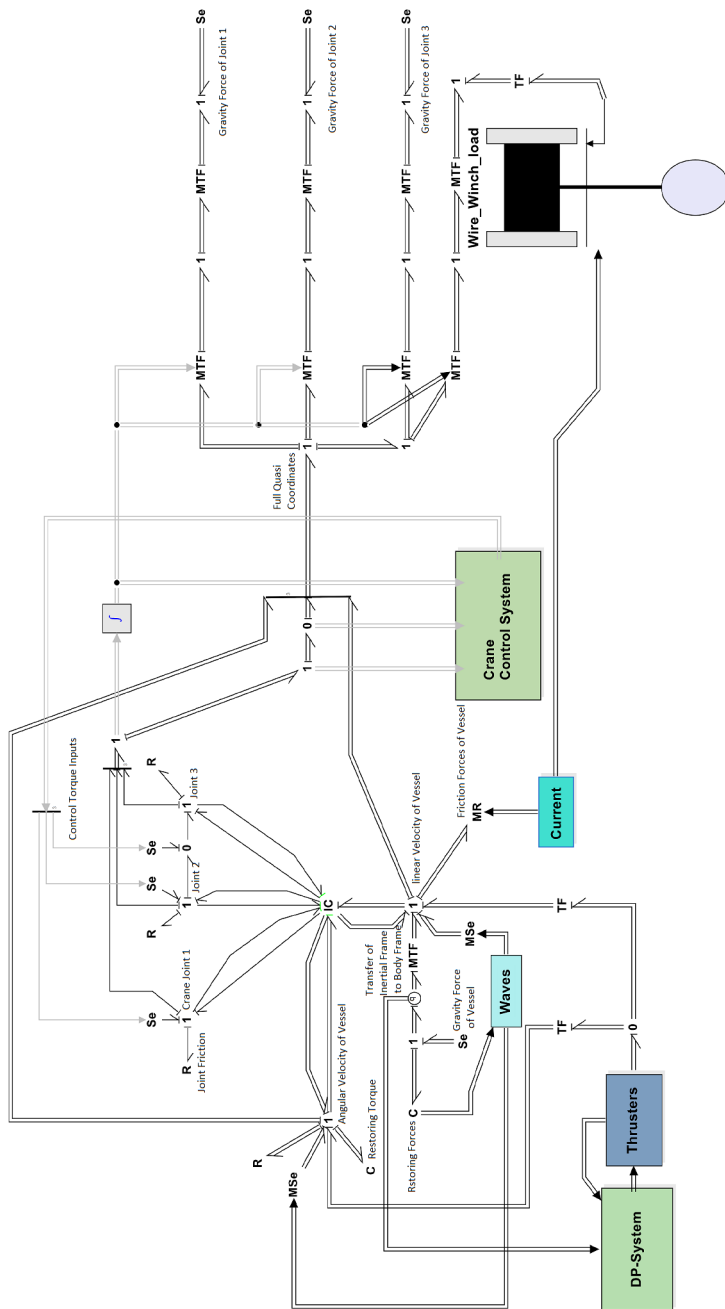


Figure 3.4: Overview of OSOM in 20-sim

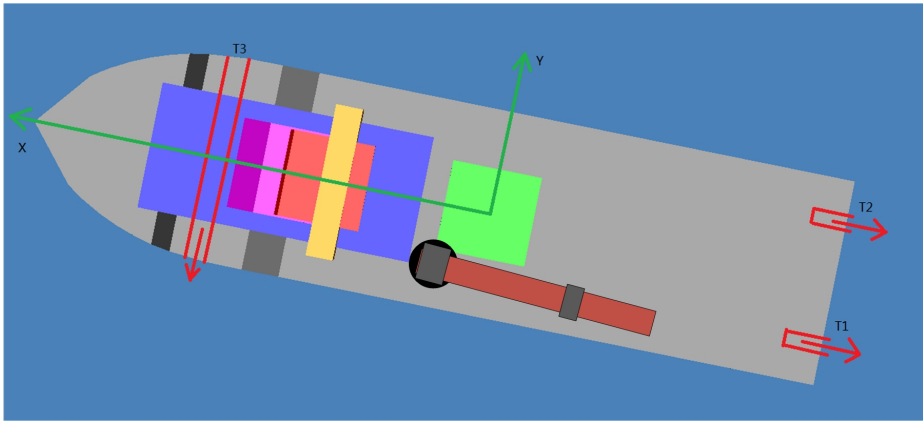


Figure 3.5: Thrusters allocation

Thruster System and Dynamic Positioning System

The DP vessel has two main thrusters T1 and T2, which is mainly used in high speed, that is, during the long distance transit operation. It also can work together with the tunnel thruster T3 to obtain the required thrust outputs in order to fulfill the requirement of dynamic positioning when performing some installation operation in the high seas.

The two sub-models are shown in the figure 3.4 as the box "thrusters" and "DP-system". Same with the restoring forces, the thrust will act on the linear velocity and angular velocity respectively. After the TF elements that provide the reference frame transformation, they connect with the two 1 junctions that represent the linear velocity and angular velocity respectively.

Winch and Wire Models

This model is based on the master thesis Skjong (2014) and the papers in the appendix. The basic idea is divided the wire into several smaller components and perform like a spring-mass-damper system. Dynamics of wire is modeled in parts and connected by power bond. The payload in the end of the wire is modeled as a Se element, which add the force on negative z axis. The sub-model connects with the basic model in OSOM via the crane body 3, then transfers the dynamics to the DP vessel. The connection is shown in the figure 3.4 on right part which represents the gravity forces of crane linked bodies and the payload.

Crane Actuator System and Control System

The crane on the deck is the main equipment that used on almost all kinds of marine operations. Since this is a heavy duty marine crane, the hydraulic system usually chosen in the actuator system. This is because that hydraulic system can handle much higher payload with fast response and high accuracy, more importantly, they will not lose the working pressure or working position when suddenly lose the power on board compare to the electrical powered actuator system.

In the OSOM platform, the hydraulic system is not included for the marine crane yet. The control torque is added on each joints of the crane directly from the marine control system.

There are some reference models in the crane control system that generate the reference position from the inputs from joystick. There are also measured values from the system. By comparing the difference from reference value and measured value we get the error dynamics. Then the error will be inputted to the control system that calculate the required torques act on each joints in order to follow the position signals generated from the joystick.

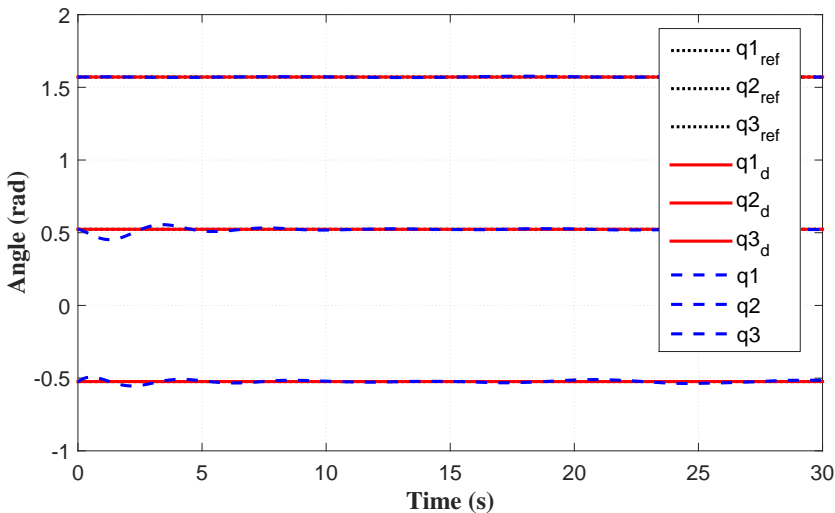


Figure 3.6: Joint displacements

This part will be discussed in details in this master thesis, the complete hydraulic system will be developed and modeled to replace the direct torque outputs. This will include the geometry of marine crane, hydraulic pump, safety valve, 4-way control

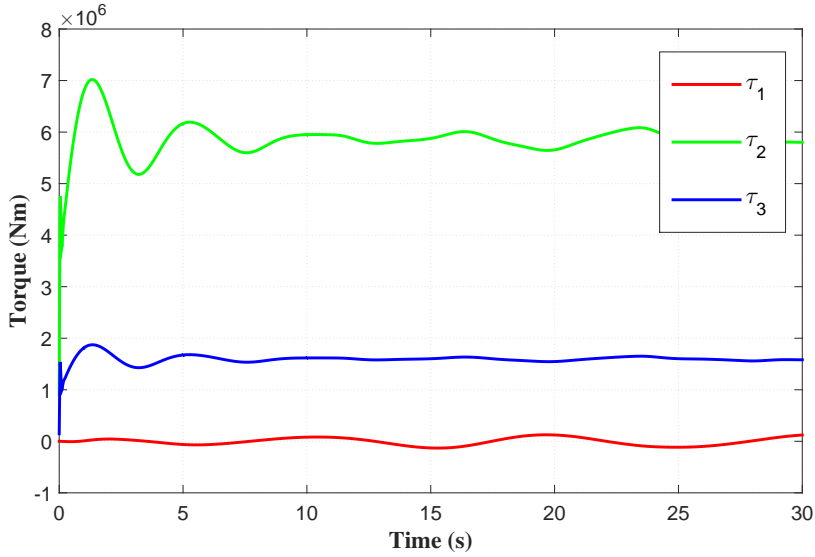


Figure 3.7: Torques on each joint

valve, piston actuator, motor actuator etc. The simulation results from OSOM are shown figure 3.6 and figure 3.7. Figure 3.6 shows the reference angular position which generate from guidance system, and actual joint angular displacements for three joints of the marine crane. Figure 3.7 shows the control torques that act on the three joints of the marine crane. Please notice that in the OSOM now, the control torques are directly generated by the reference torques and measured torques through a hybrid control system, which can be seen as the output from the fictitious actuator systems. The hydraulic actuator system will be built and discussed later.

Chapter 4

3D Visualization of OSOM Platform

Nowadays the visualization has expanding applications in many areas such as science and engineering. The visualization usually are graphics or animations on computer. In this chapter the author aimed to develop the 3D visualization of the Offshore Construction Ship with dynamic positioning ability on an irregular wave sea surface by GLview, the data is generated from the simulator in 20-sim and connect with GLview by the Dynamic Link Libraries (.DLL) file which generated by C programming.

4.1 Advantages and Applications of Visualization

From the general definition, the visualization is any technique for creating images, diagrams or animations to communicate the message. This message is usually not easy to express in its original form, the form of the message can be pure data or other elements in engineering. Visualization can make the huge and complex models easier to be understood. This is a good interface between human beings and the huge amount of data or models. For us, a great visualization can give us less limitation, more inspiration, more joy while improving our focus and accuracy.

The visualization in OSOM platform is very important. The OSOM platform aims to build the offshore operation simulator, the visualization can give the operator more awareness and direct feedback from the environment. By doing this, we can

save the money spending on personnel training with the same training standard on the real equipment. In the simulator, we can even train the operator with several situation that the real equipment cannot perform, for instance, the failures and accidents that possible happened during the operations. In addition, the simulator can be used to test the operation procedures in different situation, such as different payload the crane need to handle, different environment, etc. A well developed visualization can make all these even more effective and reliable.

There are two things that are very important to the visualization, one is close to the reality, another one is data the model support the visualization. The offshore construction ship has DP ability. The DP system that designed based on the marine vessel dynamics and environment loads have been introduced and discussed in Chapter 3. Here we mainly focus on the application of these systems and theories, as well as the communications between different softwares instead of explaining the basic theories that related with these topics.

4.2 Softwares

As mentioned in the section 4.1, the visualization cannot be developed without any simulation data. The simulation method used here is the bond graph method that explained in both chapter 2 and chapter 3. The software to achieve the bond graph modelling method is 20-sim.

This software is a modelling and simulation software, with this software you can model and simulate the dynamic systems, and analyze the behavior of different systems under different situations. The system here can be the mechanical system, electrical system or hydraulic system etc. And 20-sim fully supports the graphical modelling, this gives the developers a more intuitive and user-friendly way.

20-sim gives the framework for the bond graph modelling. We can set the dynamic equations, which mostly the constitutive law in each element in bond graph. Then linked them by power bond with effort and flow that communicate and connect the blocks into a whole bond graph. Besides the basic function of modelling under the bond graph framework, 20-sim also provides many toolbox, such as control toolbox, frequency domain toolbox, mechanization toolbox, 3D mechanics toolbox and animation toolbox.

In the initial stage of building the OSOM platform, the 3D animation toolbox is used in order to give more direct information about the simulation results and to judge the results in a more straightforward and effective way. This is a real time animation shown in figure 4.1. And the video of it will be included in the CD-ROM

of this master thesis. There are two camera view in this animation. The top one

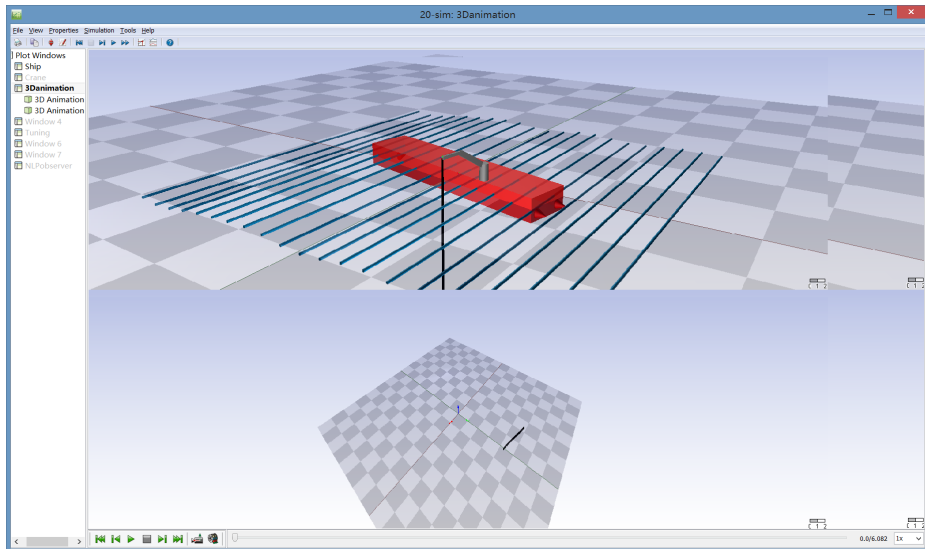


Figure 4.1: Initial 3D animation of OSOM in 20-sim

is the the DP vessel mounted with crane that do the subsea installation operation in the irregular wave surface. The box or barge represent the DP vessel. The crane which performs the operation is mounted on the vessel that connect with a wire which load the equipment to be installed. The lied cylinders represent the sea surface. The bottom figure shows the wire displacement relative to the subsea location that the installation operation desired.

This 3D animation shows the basic properties of the OSOM platform and can give us certain amount of information. However, the interface of 3D animation toolbox in 20-sim has its limitation, for instance, quite hard to draw an advanced model and not powerful enough to interface with other software or device directly. Hence, to develop an advanced and easy accessible 3D animation or visualization for OSOM is very important. The GLview is used in this master thesis.

The GLview Inova is a visualization software that developed by Ceetron in Trondheim. This software is a fully featured post-processor and 3D visualization tool with direct file interface with all major FEA and CFD tools on the market Ceetron (2014). These features make GLview Inova a great general purpose processor for numerical simulations. For instance, ABAQUS, ANASYS, FEMAP etc. GLview can provide extreme graphical performance for visualization. This software also integrates with strong presentation, import and export ability. The import ability can offer a high

level data interface to most CAE software in a uniform way, the export can also provide the data by VTFx format.

Now, both the numerical simulation software and visualization processor software are prepared. The next step is how to connect them in order to get real time visualization or even on line visualization. This will be explained in details in section 4.4. In order to achieve this, some other software or program are needed. Notepad++, which is used to write the codes when programming; the C code compiler, MinGw, which is a open source program that can generate Dynamic Link Libraries (.DLL) files and execute file.

Please notice that here are many other choices of software for the 3D visualization, such as Paraview or Visual Python. The steps of the application in this master thesis is mainly for the GLview, however, the ideas in the process of 3D visualization of simulation results are quite similar that can be extended to other application as well.

4.3 Challenges in 3D Visualization

Due to the complexity of the OSOM platform, it takes times to manage different challenges from different aspects, such as data processing, software implement and proper programming etc. In order to perform the visualization in a way that not only intuitive but also reliable and accuracy. Sensors and actuators play very important role in data transfer in both numerical software and visualization software. This is the key point when we get the information from outside or more specifically, the environment. Then we can control the system dynamics. Since the visualization will show this process to the operator, how to get data, process data and turn them into act is very important.

Although the sensors are all consider ideally and most of the values in system dynamic can be detected in the simulation, we still need to pay attention to it in the reality because sensors themselves can be the source of error. There are two categories of sensors, one type of them is used to detected the internal values, such as crane joint position, crane joint angular or linear velocity and torque that act on each joint. These values determine if the system will perform what it should be in the reality as in the 3D visualization. Another type of sensor is designed for measure the environment value, providing the valuable information about the surroundings. They can be a force sensor, wind velocity sensor, GPS, range sensor or gyroscope. All the information and data comes from the external sensor will be the influence on the system dynamics, they determine the reaction of the system. In a conclusion, sensors are the eyes for the system and actuators are the hands for the system. The

visualization should know what we can see and what we should do. Hence how to choose and arrange suitable sensor is the first challenge that we meet.

The second challenge is the real time or even on-line visualization of simulation results. If the purpose of the visualization is only for recording and presenting the dynamics of the systems, this point is not that important. However, if the purpose of this 3D visualization is to develop the offshore operation simulator, then how to run the system in real time or on-line is very important.

As mentioned before, in order to build a more advanced and complex visualization that can show more useful details, we need more powerful graphical software, here is the GLview. However, when we increase the complexity and add more details to the whole system, the simulation can become very slow even cannot perform the real time visualization. This can be due to the calculation ability of the computer or the methods of algorithm. This is about how to design and programming the visualization in a more effective and neater way. Several methods that used in this master thesis will introduce and discuss later.

The last challenge here is about the interface between different softwares. All these softwares are designed and used in different engineering areas, hence how to connect them to achieve the common goal is worthy sometime to think about. For instance, the import of CAE models from other softwares to the GLview. Such as from Delftship, we need to transfer the CAE file into .stl format file under the ASCII code, and make some modifications to the code in order to fulfill the requirement of the .VTF format file which can be read by the GLview Inova. Besides the models, the one thing that is even more important is how to transfer the data from simulation software 20-sim to visualization software GLview. This cannot be done by only using these two software, other program have to be used in order to transfer and reallocate the results, codes, format and structure. This will be explained in the next section.

4.4 Communication Between 20-sim and GLview

Actually it is hard to separate these softwares to describe the visualization of OSOM platform. Therefore, before discussing the programming and modelling steps in each software, it is better to explain the communication between 20-Sim and GLview first.

The communication between 20-Sim and GLview here is done by the Dynamic Link Library file, which we use .DLL file for short. It is a type of file that contains instructions that other programs can call upon to do certain things. Hence, multiple programs can share the abilities programmed into a single file. The programming of .DLL file is done by the C programming here, and compiling in MinGW under

Windows OS environment (The DLL file can also be programmed from C++ and Python). Compare to the .m file of Matlab, which can be opened from Matlab that makes it more accessible, the .DLL file are compiled out of C code and hence make them inaccessible, but much faster.

In 20-Sim, there are many toolboxes and functions that defined can use the external .DLL file to run the simulation, and in the same time, exchange the data between 20-Sim and .DLL file. The syntax of call the external .DLL file in 20-Sim is: $Y = dll(filename, functionname, X)$; That is given a function ($functionname$) of a DLL ($filename$), the DLL function returns the output values (Y) for a given input (X). We can compile the source code by C or C++ etc, to create the user defined function with input and output relations.

Example 4.1

```

1 //Define parameters and variables
2 parameters
3 filename = 'external.dll';
4 function = 'externalfunction';
5 variables
6 real x[3],y[3];
7 //x is the input for the dll function
8 equations
9 x = [sin(a);cos(b);1];
10 y = dll(filename , function ,x);

```

Please notice that in this example, the size of X and Y, no matter it is scalar or matrix, must be correspond with the size that is expected in the DLL function. In addition, in the DLL call, 20-Sim cannot generate whole code because it only contains the name of DLL file, function name in the DLL file and the arguments. Therefore, when programming the code of DLL file in C for example, the following structure should be used:

```

1 %dll_name%_%function_name%(double *inarr , int inputs , double *outarr ,
   int outputs , int major)

```

This should be done in the C programming together with the details of each functions inside C code. Before defining the DLL file, we usually initialized DLL function first. The frame work in the C code is as following

```

1 int dll_functionname(double *inarr , int inputs , double *outarr , int
   outputs , int major){
2
3 C_functionname(inarr [ ] . . . ) ;
4
5 return 0; //mean that this function is terminated successfully

```

6 }
}

Now, the dynamic link between 20-Sim and C can be built. The code for the VTF file in GLview is generated by the C programming. C code can use print function and read function with open and close a written file.

```

1 //headings
2 #include<stdio.h>
3 #include<math.h>
4 #include<time.h>
5
6 //both *fp and *fr is indicator here, defined as global value
7 static FILE *fp;
8 static FILE *fr;
9
10 //read the model data from outside
11 fr = fopen("filename_in.vtf", "r");
12 ...
13 fclose(fr);
14
15 //wirrtten visualization data to a vtf file (GLview)
16 fp = fopen("filename_out.vtf", "w");
17 ...
18 fprintf(fp, "...");
19 ...
20 fclose(fp);

```

By doing this, the data that calculated and rearranged in C can be linked with the GLview by generating .VTF file. The file is coded in ASCII format and contain all the geometry and dynamic properties of the model.

In conclusion, the communication structure between 20-Sim and GLview can be shown in the figure 4.2.

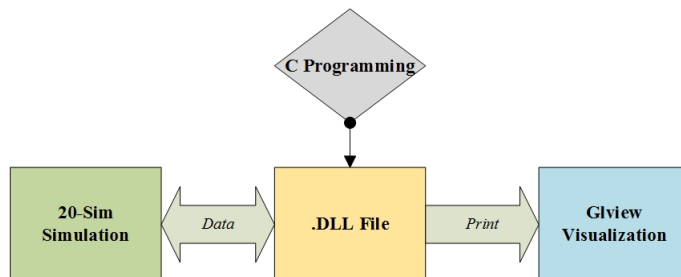


Figure 4.2: Communication between 20-Sim and GLview

4.5 Programming and Modelling Steps in 20-Sim

The steps for section 4.5, 4.6 and 4.7 will be explained in each software, together with the method that discussed in section 4.4. A complete result and code will be given in the Appendix B.

The main blocks that will be used in OSOM platform here is the "wave" and "current" (can be seen in figure 3.4), which decides the wave carpet and calculate the forces and torques that acted on the DP vessel. Inside of the wave block, there are DLL functions can call the external DLL file and set parameters that needed to do the calculation and to run the simulation.

In the wave block, the irregular waves and wave force calculation is based on the JONSWAP spectrum. The surface elevation and wave forces on ship calculated in the DLL file. Forces are approximated by numerical integration of the dynamic pressure.

Calculated forces and torques will interface with the linear velocity and angular velocity of the DP vessel that represented by 1 junctions. These two 1 junctions will connect with the *IC* field that contains the Lagrangian equations of motion, etc, both for the marine vessel and marine crane. The current in the current block is modified and estimated as a constant force acting on the DP vessel. In addition, the current will also give the wire and load that need to be installed on the seabed a constant force after passing through the splashing zone. The original code which modified to fit the DLL file inside the wave block will be provided in the Appendix C. The simple pseudo-code will be shown here in order to help readers to get the main structure of coding.

```

1 //Inside the wave box of OSOM in 20-sim
2 parameters*
3 define the ship dimension , environment parameters , animation settings ,
   etc .
4
5 variables*
6 define the variables , the inarr matrix and outarr matrix that
   communication with DLL file . The elements in the matrix including
   all the dynamics that need to be calculated and visualized .
7
8 initialequations*
9 here define the variabes that need to be calculated before simulation ,
   this section will only be executed once before the time steps . (
   Exchange and link with initial function in DLL file )
10
11 equations*
12 do the wave related calculation tasks , exchange data with DLL file . (
   Exchange and link with main function in DLL file )

```

```

13 |
14 | finalequations*
15 | this section will only be executed once, usually generated ending
    | information for the whole simulation. (Exchange and link with final
    | function in DLL file)

```

Inside this block, operator can set the parameters such as the dimension of DP vessel (L_{ship} , B_{ship} , H_{ship}), lowest/highest wave period component (T_{down} , T_{up}), number of wave components (nw), peak period of spectrum (T), significant wave height (H), the factor of wave sharpness ($gamma$). All these parameters that mentioned above are related to the wave itself and wave elevation calculation. You can also set the parameters for the 3D visualization in GLview as well.

Here the parameters related with visualization in GLview including wave carpet length/width (L_{hav} , B_{hav}), number of wave elevation elements (nel), how many steps or frames will be skipped ($skip$) in the final animation in GLview. The $tstp$ is the time step in VTF file, more specifically, it is a counter in order to give a index for each animation frame step.

The wave excitation force will strongly influence the dynamic of DP vessel, by introducing $r_{0/b}^0$ and $R_{0/b}$, which are linear displacement of x, y, z direction and rotation matrix of the vessel correspondingly. We can describe the motion of the DP vessel, as well as the crane base that mounted on the deck. For simplification, here we only consider the force or torque that influence heave, yaw and pitch. They are described as *tan lin* and *tau ang*.

For the general structure of the code, there are initial equations, equations and final equations. The initial equations and final equations will be only executed once at the beginning of the simulation and at the end of the simulation correspondingly. The equation will be executed through all the time steps of the simulation.

In each of the equations groups, they will call the same external DLL file but for different DLL function inside. For instance, the initial equations will respond for setting up the dimension of wave carpet, reading the model data from one VTF file and writing processed visualization data to VTF file in its unique format under ASCII. The equation will calculate the wave elevation for each step, give linear and rotation movements to the DP vessel based on the forces and torques here and print the main body of VTF-ASCII code to the VTF file that being opened in the initial equations. The final equations will decide how many steps to skip in the animation, write the last part of VTF-ASCII code and close the file. This is the working procedure and function in the OSOM platform wave block. The window for managing all the parameters and variable is shown in figure 4.3.

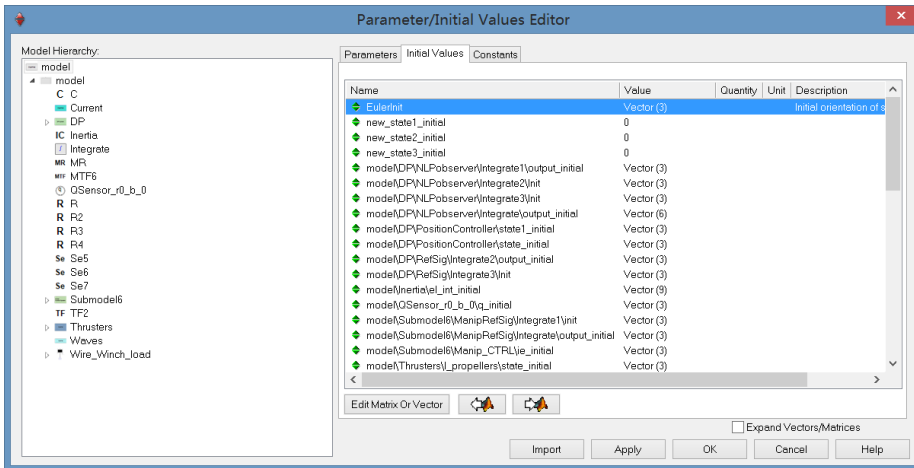


Figure 4.3: Window of parameter setting

In figure 4.3, the marked one which called "EulerInit" is the initial orientation of the DP vessel. During the simulation, we can change this value and see how it influences the dynamic of DP vessel from visualization. So, 20-Sim acts like a commander, giving out the necessary settings and data to the visualization that can be shown through the GLview. You can change wave data to generate different wave carpet under different weather conditions, and you can also change the position and orientation of the DP vessel to visualize the different working conditions. The message here will be transferred to the VTF file in GLview by the external DLL file as explained in the section 4.4.

4.6 Programming and Modelling Steps in C Programming

The purpose of C programming is to generate the DLL file that do the calculation task for 20-Sim to run the simulation and to print the VTF ASCII code to a VTF file that can be read from GLview.

Before explaining the programming in this section, we need to go through the basic hydrodynamic theories in order to build the wave carpet. For the dynamics of the DP vessel, it is already explained before, so here we will not repeat that.

There are some assumptions for the sea environment. The sea water is assumed incompressible and inviscid, while motion of fluid is non-rotational. The kinetic boundary conditions and dynamic of free surface conditions will be considered when

developing the statistical description of the waves.

4.6.1 Wave Theories

The wave is long-crested irregular sea propagating along a certain direction. Linear theory is used to simulate the sea state. There are many kinds of environment loads can excite the ship. For instance, the waves can generate dynamic pressure on the ship hull, then transfer as forces act on the ship hull; the sum of radiation and diffraction forces, which contributes to the added mass forces and damping forces. The sea propagating in positive x direction can be written as the sum of a large number of wave components as describe in Faltinsen (1990).

$$\zeta = \sum_{j=1}^N A_j \sin(\omega_j t - k_j x + \epsilon_j) \quad (4.1)$$

In the equation, the A_j is the wave amplitude, ω_j is the circular frequency, k_j is the wave number and ϵ_j is the random phase angles for wave component j . The phase angles are distributed between $0 - 2\pi$ and keep constant with time. The wave number k is given as

$$k = \frac{2\pi}{\lambda} \quad (4.2)$$

Here the λ is the wave length, in deep water, or can be estimated as infinite water depth. The wave length can be described as

$$\lambda = \frac{gT^2}{2\pi} \quad (4.3)$$

The wave period T can be described by

$$T = \frac{2\pi}{\omega} \quad (4.4)$$

By using the linear wave theory, the wave propagating with positive x direction is

$$\Phi = \frac{gA_j}{\omega_j} e^{kz} \cos(\omega_j t - k_j x + \epsilon_j) \quad (4.5)$$

Here g is the gravity acceleration and z is the vertical distance below the average wave surface. Please notice that only the x-direction is included. Now the wave potential is calculated, from it we can get the wave dynamic pressure.

$$pD = \rho \frac{\partial \Phi}{\partial t} = \sum_{j=1}^N \rho g A_j e^{kz} \sin(\omega t - kx + \epsilon_i) \quad (4.6)$$

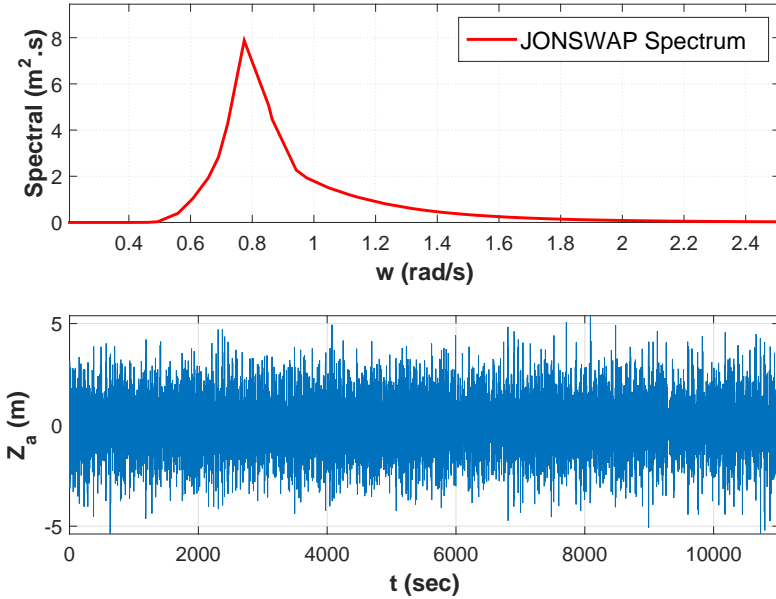


Figure 4.4: JOWSWAP spectrum under $H_s=6m$ $T_p=8s$

Wave parameters that used here is determined by the JONSAWP(Joint North Sea Wave Project) spectrum. The distribution of the JONSWAP spectrum under $H_s = 6m$ and $T_p = 8s$ is shown in figure 4.4. The spectrum function that related with frequency ω can be described as

$$S_n(\omega) = \frac{5}{32\pi} H_s^2 T_p \left(\frac{\omega_p}{\omega}\right) e^{-\frac{5}{4}\left(\frac{\omega_p}{\omega}\right)^4} (1 - 0.287 \ln \gamma) \gamma^\epsilon e^{-\frac{(\frac{\omega}{\omega_p} - 1)^2}{2\sigma^2}} \quad (4.7)$$

Here the H_s is the significant wave height, which is $\frac{1}{3}$ of the highest wave. T_p is the peak period of the wave spectrum, ω_p is the peak frequency, γ is a parameter of the spectrum. σ equals to 0.07 for $\omega < \omega_p$ and equals to 0.09 for $\omega > \omega_p$.

The excitation forces are from integrate the dynamic wave pressure of the wet surface of the ship. In the OSOM platform, simplification are made. Only the dynamic pressure that on the bottom of the ship is taken into consideration. This reduces the number of forces to only three, i.e, forces in surge, sway and heave

$$F_i = - \int_s p D n_i ds \quad (4.8)$$

Here the S is the wet surface area of the ship. The excitation forces obtained from numeric integration of the pressure over the hull bottom are

$$\begin{aligned}
 F_{heave/b} &= \sum_{i=1}^N \sum_{j=1}^M pD(x_i)\Delta_i\Delta_j \\
 M_{roll/b} &= \sum_{i=1}^N \sum_{j=1}^M -y_{pi}pD(x_i)\Delta_i\Delta_j \\
 F_{pitch/b} &= \sum_{i=1}^N \sum_{j=1}^M x_{pi}pD(x_i)\Delta_i\Delta_j
 \end{aligned} \tag{4.9}$$

Here we can use rotation matrix \mathbf{R}_b^0 to transfer the forces or torque from body fixed reference frame to inertial reference frame. After transferring into the inertial reference frame, we can see that the wave-ship interaction already included under the body fixed reference frame.

There are added mass and hydrodynamic damping existing for the vessel as well. In the OSOM platform, only nonlinear vortex shedding forces and linear hull friction forces and torques are implemented. The vortex shedding forces are described as

$$\mathbf{D}_{NL}(\mathbf{v}_{b/c}^b) = \frac{1}{2}C_D\rho(\mathbf{v}_{b/c}^b)^T \mathbf{A}_p|\mathbf{v}_{b/c}^b| \tag{4.10}$$

The C_D is the drag force coefficient and $\mathbf{A}_p = \text{diag}(A_u, A_v, A_w)$. The A_u, A_v, A_w are the projection of underwater area in surge, sway and heave.

The linear hull friction forces and torques are

$$\begin{aligned}
 \boldsymbol{\tau}_{fv}^b &= \mathbf{D}_{NLv}(\mathbf{v}_{b/c}^b) + \mathbf{D}_L v \mathbf{v}_{b/c}^b \\
 \boldsymbol{\tau}_{f\omega}^b &= \mathbf{D}_{NL\omega}(\boldsymbol{\omega}_{b/0}^b) + \mathbf{D}_L \omega \boldsymbol{\omega}_{b/0}^b
 \end{aligned} \tag{4.11}$$

The \mathbf{D}_{NLv} and $\mathbf{D}_{NL\omega}$ are the nonlinear damping forces, $\mathbf{D}_L v$ and $\mathbf{D}_L \omega$ are the diagonal matrices of linear friction forces.

4.6.2 Programming

The function of this C code is to generate the external DLL file that do the calculation for the "wave" block in OSOM in 20-sim, create and print the VTF ASCII code to a VTF file for visualization. The whole codes are attached in the Appendix D.

First defining the general and global values, including constant values and pointers in order to write syntax to the same VTF file in different inner functions. Before calculating the wave elevation, wave forces act on ship and wave torques act on ship, the spectrum value for a given frequency component should be calculated here as well based on the function 4.7.

```

1 #define g 9.81
2 #define pi 3.1415
3 #define rho_w 1025
4 //-----global pointers for print/read file-----
5 static FILE *fp;
6 static FILE *fr;
7
8 //---Calculate the spectral value for a given frequency component---
9
10 double Spectrum(double w, double T, double H, double gamma){
11
12 double t1, t2, sigma, w_p, t3, t4, t5, t6;
13 float S_value;
14 if (w == 0) S_value = 0;
15 else{
16 t1 = (5 / (32 * pi)) * H * H * T;
17 w_p = (2 * pi) / T;
18 t2 = w_p / w;
19 if (w <= t2)
20 sigma = 0.07;
21 else
22 sigma = 0.09;
23 t3 = exp((-5 *pow (t2, 4)) / 4);
24 t4 = 1 - 0.287 * log(gamma);
25 t5 = (1 / t2) - 1;
26 t6 = exp(- (t5 * t5) / (2 * sigma * sigma));
27
28 S_value = t1 * pow(t2, 5) * t3 * t4 * pow(gamma, t6);
29 }
30
31 return S_value;
32 }

```

Please notice that $t1...t6$ is the temporary value in the calculation. The JONSWAP spectrum is implemented here. The S_{value} will return in the end of the function and will store for later use.

Then we come to the initial function, which is called by the wave block in OSOM in 20-Sim in the initial equation there. Here the first task is to load the ship model in a VTF file, the command used here is

```

1 //start reading the PSV.VTF file as input and print to .VTF for GLview
   3D visualization
2 char buf[1024];
3
4 fr = fopen("PSV.vtf", "r");
5 while(fgets(buf, sizeof(buf), fr) != NULL)
6 {
7 fputs(buf, fp);

```

```

8   printf("%s", buf);
9   }
10  fclose(fr);
11  //finish reading the PSV.vtf data file from outside

```

After reading the external PSV model file, another VTF file for the visualization is created and open for writing. The function is under the instruction of pointers f_r and f_p here. The syntax of open and write a new file can be

```

1  fp = fopen("visualization.vtf", "w");
2  fprintf(fp, "content1\n");
3  fprintf(fp, "content2\n");
4  ...

```

After reading the ship model file and opening a new VTF file, now the elements of wave carpet should be included. The wave carpet contains many meshes. We need to decide the area of the carpet and how many elements it has along the wave moving direction, which is chosen as x direction in the OSOM. The code is

```

1  //generate wave elements in coordinates and print them to .VTF file
2  for (i = 0; i <= (n_el);)
3  {
4  xw = x_max - i*dx;
5  fprintf(fp, "%f %f %f\n", xw, y_max, zo);
6  fprintf(fp, "%f %f %f\n", xw, y_min, zo);
7  ++i;
8  }
9  fprintf(fp, " \n");
10 i = 0;
11
12 fprintf(fp, "!Wave elements\n");
13 fprintf(fp, "*ELEMENTS 11\n");
14 fprintf(fp, "%NAME \"Wave Carpet\"\n");
15 fprintf(fp, "%DESCRIPTION \"wave carpet elements\"\n");
16 fprintf(fp, "%NODES #11\n");
17 fprintf(fp, "%QUADS\n");
18
19 for (i = 1; i <= n_el;)
20 {
21 fprintf(fp, "%d %d %d %d\n", ((i * 2) - 1), ((i * 2) + 1), ((i * 2) + 2)
    , (2 * i));
22 ++i;
23 }
24 fprintf(fp, " \n");

```

The z_0 is set to zero here since in the first step only build the wave carpet in absolutely calm situation. Later, the heave motion of waves will be added to in the z direction. The content in the *fprintf* function will be explained in the next

section. Here only the coding method and coding structure are our interesting. $((i * 2) - 1), ((i * 2) + 1), ((i * 2) + 2), (2 * i)$ is the code to generate the coordinate of wave carpet. These four number indicate the coordinate of the four corner points in one wave element. The definitions of the geometry are shown in figure 4.5. The length of dx is the length of each element in the x direction.

Now the wave carpet is developed, the next step is to calculate the wave surface elevation and add the motion to the wave carpet just developed. Hence the second DLL function is set, which named `int Waves()`.

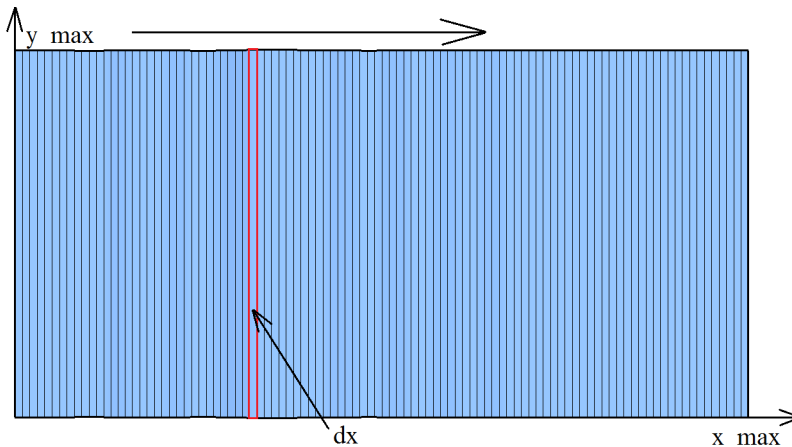


Figure 4.5: Wave carpet elements

The function is

```

1 int Waves(double ni, double nj, double L_ship, double B_ship, double
   H_ship, double r0i, double r0j, double r0k, double phi, double
   theta, double psi, double w_dwn, double w_up, double nw, double T,
   double H, double gamma, double t, int n_el, float L_hav, float
   B_hav, double tstp, double R0_b11, double R0_b12, double R0_b13,
   double R0_b21, double R0_b22, double R0_b23, double R0_b31, double
   R0_b32, double R0_b33, int skip, double *outarr)
2 {
3 }

```

All the variables in the brackets are the input values to this function. They are the output from wave block in OSOM in 20-Sim that mentioned earlier. Except for the last one called `*outarr` is the output of DLL function that will send the calculated data back to 20-Sim for simulation.

Inside this function, the calculation will be done for the animation that build inside of 20-Sim as well. After defining the variables and parameters that needed here, the main body of this DLL function will be implemented.

```

1 //random phase for wave components
2 delta [0] = a;
3 delta [1] = b;
4 delta [2] = c;
5 More random phase...
6 ...
7 ...

```

This is the random phase of wave components, which is a very important parameters in order to build the irregular wave surface. The parameters and rotation matrices for the ship motion is also defined here as

```

1 //define parameters and variables for the ship motion
2 di = L_ship / ni;
3 dj = B_ship / nj;
4 dA = di * dj;
5
6 F3 = 0;
7 T1 = 0;
8 T2 = 0;
9
10 cpsi = cos(psi);
11 ctheta = cos(theta);
12 spsi = sin(psi);
13 cphi = cos(phi);
14 stheta = sin(theta);
15 sph = sin(phi);
16
17 //rotation matrix (R0_b)
18 R11 = cpsi * ctheta;
19 R12 = -spsi * cphi + cpsi * stheta * sph;
20 R13 = spsi * sph + cpsi * cphi * stheta;
21 R21 = spsi * ctheta;
22 R22 = cpsi * cphi + sph * stheta * spsi;
23 R23 = -cpsi * sph + stheta * spsi * cphi;
24 R31 = -stheta;
25 R32 = ctheta * sph;
26 R33 = ctheta * cphi;

```

The F_3, T_1, T_2 are the forces in heave direction, torques in yaw and roll direction respectively. d_i, d_j, d_A is used when calculating the wave excitation forces by dividing the waterline area into small elements that possible to get the forces as the sum of contributions from every small elements. The rotation matrices will transfer the ship from body fixed reference frame into inertial reference frame. Now, we will print the calculated wave elevation to the VTF file.

```

1  int inttstp = 0;
2  inttstp = tstp;
3
4  //print the results of wave carpet after skip steps
5  if (inttstp%skip == 0)
6  {
7  fprintf(fp, "*RESULTS ");
8  fprintf(fp, "%d\n", inttstp / skip);
9
10 fprintf(fp, "%DIMENSION 3\n");
11 fprintf(fp, "%PER_NODE #11\n");
12 }
13 dx = (x_max - x_min) / n_el;
14
15 for (i = 0; i<= n_el; i++)
16 {
17 xw = x_max - i * dx;
18 surf_el = 0;
19
20 for (j = 0; j < nw; j++)
21 {
22 surf_el = surf_el + zetaA[j] * sin(w[j] * t - k[j] * xw + delta[j]);
23 }
24
25 if (inttstp%skip == 0)
26 {
27 fprintf(fp, "%f %f %f \n", xo, yo, surf_el);
28 fprintf(fp, "%f %f %f \n", xo, yo, surf_el);
29 }
30 }

```

These codes here will print the *surf_el*, which is the wave surface elevation results to the VTF file. The printed file that shown in GLview is shown in figure 4.6.

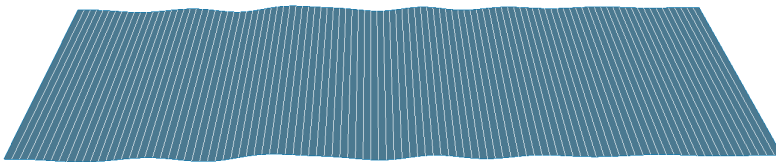


Figure 4.6: Wave elevation generated from C code in GLview

```

1  if (inttstp%skip == 0){
2  }

```

There is one trick in generating the 3D visualization, which will use the above codes. The reason for the code is the step time in 20-Sim for simulation is very small, for example it is 0.0004s in the OSOM platform. It generates all the position and motion information for each time step, hence the visualization file will become extra huge. And there is no need for us to do that because the resolution is far higher than the ability of human can tell apart. For instance, usually we set the default number as 30FPS. Reducing the unnecessary data will not influence the quality of the final visualization.

By using this method, we can scale down the size of visualization data. This will reduce the overall calculating task in GLview. Then the forces and torques that act on the vessel should be calculated, the data will send back to 20-Sim as well, and give feedback to DLL file as the rotation matrices and displacement of the vessel. This will be shown on the 3D animation directly. The code is

```

1
2 //calcualte the force and torque of ship
3 for ( i = 1; i < ni+1; i++)
4 {
5 //local x-pos
6 x = -L_ship / 2 * (1 + 2 * (1 - i) / ni) + di / 2;
7 for ( j = 1; j < nj+1; j++)
8 {
9
10 //local y-pos
11 y = -B_ship / 2 * (1 + 2 * (1 - j) / nj) + dj / 2;
12
13 //r0 (positions of element i,j in NED)
14 r0x = R11 * x + R12 * y + R13 * H_ship / 2 + r0i;
15 r0y = R21 * x + R22 * y + R23 * H_ship / 2 + r0j;
16
17 //calculate forces and torques act on ship
18 for(h=0; h < nw; h++){
19 p = rho_w * g * zetaA[h] * exp(-k[h] * H_ship / 2) * sin(w[h] * t + k[h]
    ] * r0x + delta[h]);
20
21 F3 = F3 + dA * p;
22 T1 = T1 + y * p * dA;
23 T2 = T2 - x * p * dA;
24 }
25 }
26
27 }
28 outarr[21] = F3;
29 outarr[22] = T1;
30 outarr[23] = T2;
31 return 0;
32
33 }

```

After this, the end function that related with the final equation in 20-Sim is implemented. This end DLL function will write the ending of the visualization VTF file. Then close the VTF file to complete the task. The code will not be shown here, but some of them will be discussed in the next section about the ASCII VTF format file generation.

Finally, the 20-Sim interface in the C code will be implemented. It is the key to connect the external DLL file that based on C code to 20-Sim. The format and structure of the interface had been already described in the section 4.4 about the communication between GLview and 20-Sim, therefore the code will be shown here. There are three DLL functions inside the C code, so three interfaces will be designed as following

```

1  //-----20-sim interface-----
2
3
4  int UpdateWaves(double *inarr , int inputs , double *outarr , int outputs ,
5                 int major){
6
7  //This function is called from 20-sim once each time "tstp"
8  Waves(inarr [0] , inarr [1] , inarr [2] , inarr [3] , inarr [4] , inarr [5] , inarr [6] ,
9         inarr [7] , inarr [8] , inarr [9] , inarr [10] , inarr [11] , inarr [12] , inarr [13] ,
10        inarr [14] , inarr [15] , inarr [16] , inarr [17] , inarr [18] , inarr [19] ,
11        inarr [20] , inarr [21] , inarr [22] , inarr [23] , inarr [24] , inarr [25] ,
12        inarr [26] , inarr [27] , inarr [28] , inarr [29] , inarr [30] , inarr [31] ,
13        outarr );
14
15  return 0;
16
17 }
18
19 int InitiateWaveSimulation(double *inarr , int inputs , double *outarr ,
20                           int outputs , int major){
21
22 //This function is called from 20-sim once before the simulation starts
23 //This function do not give any value back to the 20-sim, hence there
24 //are no outarr
25 printHeader(inarr [0] , inarr [1] , inarr [2] , inarr [3] , inarr [4] , inarr [5])
26 ;
27
28 return 0;
29
30 }
31
32 int TerminateWaveSimulation(double *inarr , int inputs , double *outarr ,
33                             int outputs , int major){
34
35 //This function is called from 20-sim once after the simulation starts
36 //This function do not give any value back to the 20-sim, hence there
37 //are no outarr

```



```

27 printEnding(inarr[0], inarr[1]);
28
29 return 0;
30 }
31
32 //-----E N D-----

```

The variables that send from 20-Sim are collected as *inarr*[] and the calculated data that send back to 20-Sim are represented by *outarr*[]. The number of *inarr*[] should be same in the 20-Sim interface, in DLL function and in the function in 20-Sim. The order of the variables should also be taken into consideration.

4.7 Programming and Modelling Steps in GLview Inova

The communication between GLview and 20-Sim has been already developed now. The VTF-ASCII format code and structure will be described now, including preparing the 3D model. Simple example will be provided in order to help the readers to catch the main idea how to build the VTF file in GLview, since the visualization file of OSOM platform is huge and complex and thus hard to follow.

4.7.1 Preparing the ship 3D model

GLview can read almost all the common format CAE/CAD files from external softwares. The simulation or visualization results from GLview will be written in the compact and flexible VTF or VTFx file format. The VTF file can be in both ASCII or Binary format and ASCII format is chosen in this master thesis. For instance, the ship model can be exported from software Delft Ship. The PSV model in Delft Ship is in .FBM format. The model is shown in figure 4.7. This model can be exported to .STL file in ASCII format. The STL file is a file format native to the stereo-lithography CAD software created by 3D Systems, defined from Inc. (1989). STL which means Standard Tessellation Language, the example of STL and CAD format can be shown in figure 4.8. We can use the Notepad++ to open the STL file and see the structure and format as following code shows, the model is made up of many defined triangle elements.

```

1 solid
2 facet normal 0.0 1.0 0.0
3 outer loop
4 vertex 7.2121 11.125 9.1389
5 vertex 7.2106 11.125 10.2395
6 vertex 8.4301 11.125 10.2403
7 endloop

```

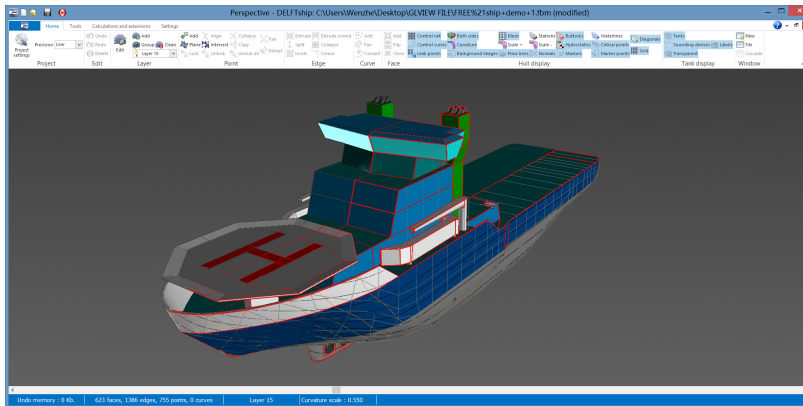


Figure 4.7: PSV model in Delft Ship

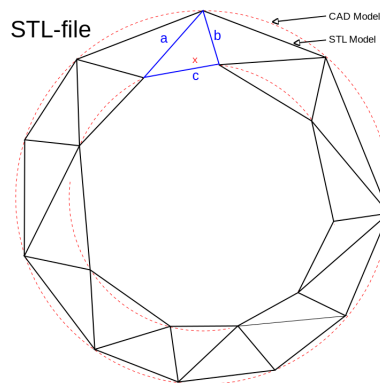


Figure 4.8: Compare the STL and CAD format

```

8 | endfacet
9 | facet normal 0.0 -1.0 0.0
10 | outer loop
11 | vertex 7.2121 -11.125 9.1389
12 | vertex 8.4301 -11.125 10.2403
13 | vertex 7.2106 -11.125 10.2395
14 | endloop
15 | endfacet
16 |
17 | ...
18 |
19 | endsolid

```

The GLview can read the STL format, however, the STL file only contains the surface geometry of 3D object. The model after importing in GLview is shown in figure 4.9. The 3D model import is evaluated here, the purpose of this section is to demonstrate

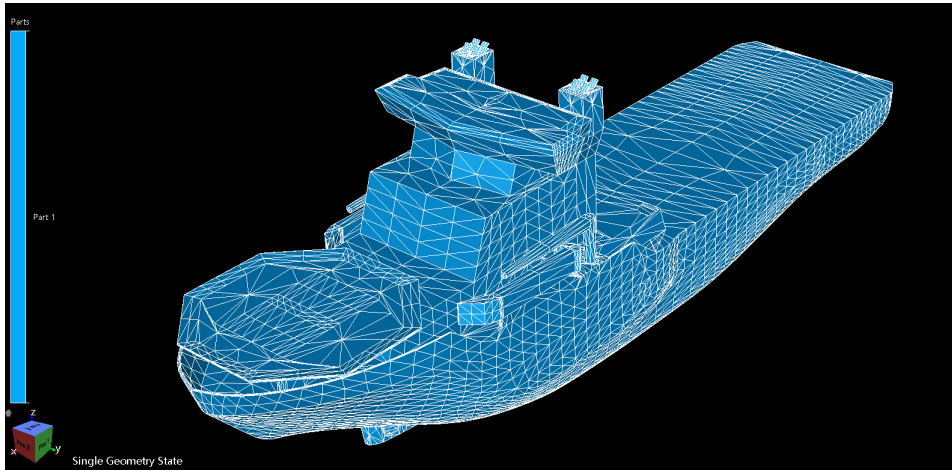


Figure 4.9: Improt model in GLview

the methods of import 3D model from other softwares. There are many other formats that the GLview can read, therefore there are many other possibilities to import the 3D model.

Please notice that the PSV model used in the visualization of OSOM is different from the model from Delft Ship, which is much simpler and with 10 components in different colors. The PSV model finally chosen is built in VTF-ASCII directly with less unnecessary details of the geometry. This can help increase the overall efficiency of visualization because it avoid the unnecessary elements implemented in the heavy calculation.

4.7.2 Build the Crane 3D Model

In this section, method of build and implement the marine crane will be introduced. Brief introduction of crane components and dynamics will be provided here as well.

Build the Model in SoildWorks

This crane model is built in SoildWorks. The crane have the similar dimension with general marine crane, which can be adjusted in different design and simulation. Rendering such as change colors can be applied in SoildWorks. However, the STL file which explained before, will only contain the geometry information of the crane

based on triangle elements. This means that all the rendering features will be lost during the model format translate process, and the color in different parts will be defined again in GLview. The original SLDPRT file and corresponding transferred STL file will be provided in the CD-ROM. The finished marine crane model is shown in figure 4.10.

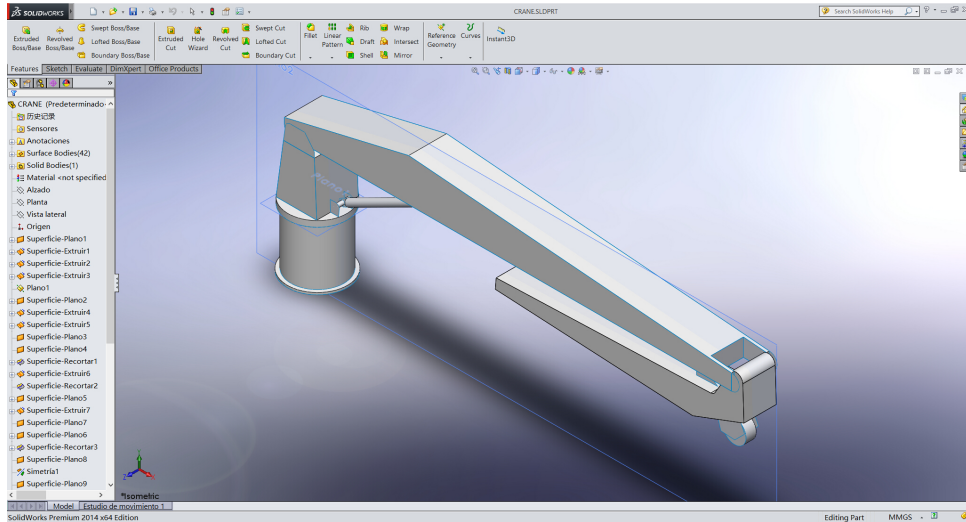


Figure 4.10: The finished crane 3D model

Implement the Model to GLview under OSOM

The method of implementing the 3D model of marine crane is similar with the ship, which is to transfer the STL format file to the ASC-II format VTF file and merge with the ship model that developed before. All these models will be read by the DLL file to generate the 3D animation for GLview.

Special attention should be pay on the bodies and joints of marine crane. Three linked rigid bodies and three low pair joints should have clear coordinates. In the simulation of OSOM, rotation matrices for each joints is sent to DLL file and written in the VTF file for visualization. The joint angle or joint displacement is controlled by the torque generated from hydraulic system. However, in the OSOM platform, the hydraulic system has not been implemented yet. The control torque is given from the control system directly. In the OSOM, there is a reference system representing the signal from joystick. Sensors will send the measured joint displacement to the control system and compare with the reference joint displacement to get the error dynamics, which is the base for control system operating. Since the dynamic of marine crane is still under developing, it has not been included in the animation

for now. However the 3D model of marine crane is not difficult to implement for the later use. The components of marine crane are shown in figure 4.11. The crane is modeled with three linked rigid bodies, one hydraulic motor and two hydraulic piston. The working space for body 2 or crane boom on x-y plane is shown in figure 4.11 as well. The rotation base that actuated by the hydraulic motor will rotate about the y-axis. The crane boom and crane jib will rotate about the z-axis that actuated by the hydraulic piston 1 and hydraulic piston 2 respectively.

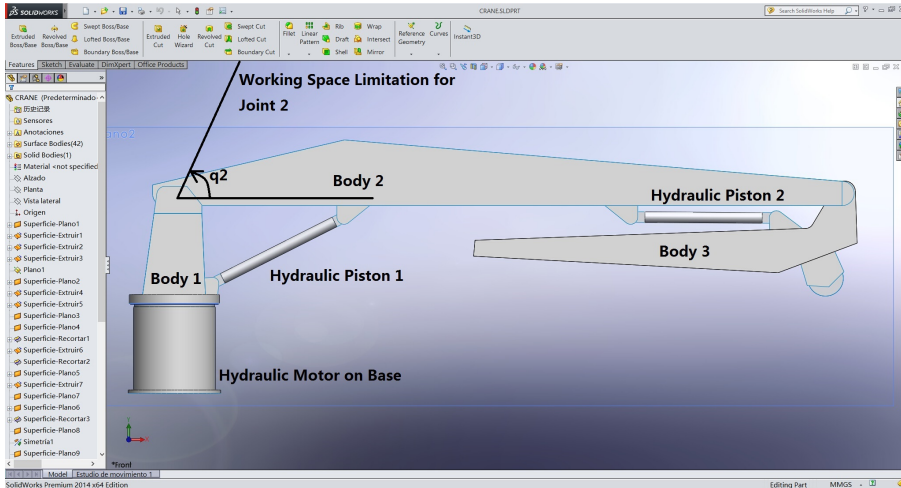


Figure 4.11: The components of crane 3D model

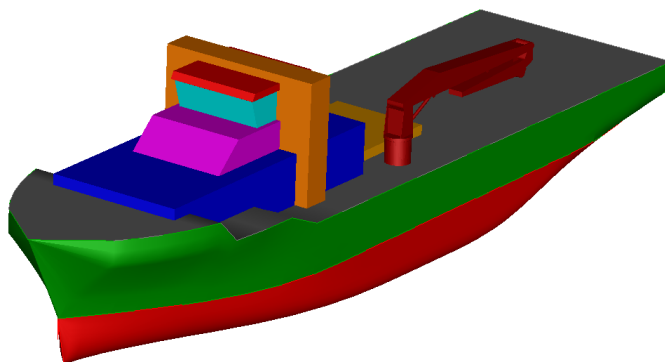


Figure 4.12: The completed DP vessel with marine crane

4.7.3 VTF Code Structure

Table 4.1: Top-level identifiers in GLview

Identifier	Description
*VTF-1.00	ASCII file header
*NODES [<ID>]	block of nodes
*ELEMENTS [<ID>]	block of elements
*GLVIEWGEOMETRY [<ID>]	GLview geometry
*RESULTS [<ID>]	block of results
*GLVIEWSCALAR [<ID>]	GLview scalar result.
*GLVIEWVECTOR [<ID>]	GLview vector result.
*GLVIEWPOSITIONSCALAR [<ID>]	sequence of scalars with pos.
*GLVIEWPOSITIONVECTOR [<ID>]	GLview separate vector.
*TRANSFORMATIONRESULT [<ID>]	One trans formation matrix
*GLVIEWTRANSFORMATION [<ID>]	GLview transformation result.
*GLVIEWDISPLACEMENT [<ID>]	GLview displacement result.

The file begins with a header, which is called "*VTF-1.00". Followed by this head, there are many other blocks that defined by the top-level identifiers. All of them will begin with the symbol "*" and perhaps followed by symbol "%". Each blocks will also have an ID which can be as a reference when processing the data. The table shows the main top-level identifiers in GLview in table 4.1, which is provided by Ceetron (2006). A simple example will be explained to help readers catch the main idea of VTF code structure.

Example 4.2

```

1 |=====
2 |!start programming and define header
3 |=====
4 |
5 |*VTF-1.00
6 |
7 |=====
8 |!define nodes and elements
9 |=====
10 |
11 |*NODES 1
12 |...(Nodes that build this element)
13 |*ELEMENTS 1
14 |...(Define how these nodes form an element)
15 |
16 |=====

```

```

17 *NODES 10
18 ...(Nodes that build this element)
19 *ELEMENTS 10 #Until here, all the nodes are for PSV
20 ...(Define how these nodes form a element)
21
22 *NODES 11
23 ...
24
25 *ELEMENTS 11 #This is for the nodes of wave carpet
26 ...
27
28 !=====
29 !define geometry, can only have one in each vtf file
30 !=====
31
32 *GLVIEWGEOMETRY 1
33 1, 2, 3 ... 11 #This will include all the elements
34
35 !=====
36 !results for the wave carpet, in vector;
37 !=====
38
39 *RESULTS 1
40 %DIMENSION 3
41 %PER_NODE #11 #The refer to nodes group 11 (NODES 11)
42 ...(The result of wave elevation)
43
44
45 *RESULTS N
46 %DIMENSION 3
47 %PER_NODE #11
48 ...(The result of wave elevation)
49
50 !=====
51 !results for the ship rotation.
52 !=====
53
54 *TRANSFORMATIONRESULT 1
55 %ELEMENT_BLOCK_ID #1
56 ...(4*3 rotaion matrix with linear movement)
57
58 *TRANSFORMATIONRESULT 2
59 %ELEMENT_BLOCK_ID #2
60 ...(4*3 rotaion matrix with linear movement)
61 ...(Rotation results that applied to all the elements of ship)
62
63 *TRANSFORMATIONRMESULT 10
64 %ELEMENT_BLOCK_ID #10 #The ship has 10 elements
65 ...(4*3 rotaion matrix with linear movement)
66
67 *RESULTS M
68 %DIMENSION 1

```

```

69 %PER_ELEMENTS #1
70 ... (4*3 rotation matrix with linear movement)
71 ... (Rotation results that applied to all the elements of ship)
72
73 *RESULTS M+10
74 %DIMENSION 1
75 %PER_ELEMENTS #10
76 ... (4*3 rotation matrix with linear movement)
77
78 =====
79 !steps based on results , define the animation
80 =====
81
82 *GLVIEWTRANSFORMATION 1 #for ship motion
83 %STEP 1
84 ...
85 ...
86 ...
87 %STEP N
88
89 *GLVIEWVECTOR 1 #for wave carpet
90 %STEP 1
91 ...
92 ...
93 ...
94 %STEP N
95
96 =====
97 !end programming
98 =====

```

All the data in results block is calculated from 20-Sim or C program, and then print to the VTF file by DLL file that developed before. Now the meaning of each command will be explained.

The **VTF - 1.00* defines the VTF-ASCII header, which tells the GLview what kind of code the input data is. This block only contains this message. *NODES* describes the nodes of each element in 3D coordinate. This command is followed by the [*<ID>*], which is a reference to the GLview when processing other command that related to this code in the VTF file. The sub-command *NO_ID* and *WITH_ID* can also be included. The function of them is to decide if the a group of nodes (usually 3 values represent x, y, z coordinate) will be given a ID for reference. For instance

```

1 !NODES1:
2 *NODES 1
3 %WITH_ID
4 10 10.0 4.0 -2.0
5 20 -10.0 4.0 -2.0

```



```

6 | 30 -10.0 -4.0 -2.0
7 | 40 10.0 -4.0 -2.0
8 | 50 10.0 4.0 3.0
9 | 60 -10.0 4.0 3.0
10 | 70 -10.0 -4.0 3.0
11 | 80 10.0 -4.0 3.0

```

The NODES ID is 1 here. This code will define a group of nodes that form different elements. Then the elements should be defined based on the nodes. There are many sub-commands in the **ELEMENTS* block, here we only use the basic one, that is *%NODES #* and the shape of the geometry. The shape can be *%QUADS*, *%BEAMS*, *%TRIANGLES* and *%HEXAHEDRONS* etc. For generating the wave carpet, we use *%QUADS* sub-command to produce rectangle. The example of elements can be described as

```

1 | !BOXHULL:
2 | *ELEMENTS 1
3 | %NAME "box1"
4 | %DESCRIPTION "box hull"
5 | %NODES #1
6 | %WITH_ID
7 | %HEXAHEDRONS
8 | 1 10 20 30 40 50 60 70 80

```

After defined the nodes and elements in GLview, they need to be grouped into a complete geometry. The elements block ID will be used here.

```

1 | *GLVIEWGEOMETRY 1
2 | %ELEMENTS
3 | 1 2 3

```

The example of **GLVIEWGEOMETRY* is listed above, this means combine element 1, 2, 3 together. The 3D model now has been set up in the VTF file in ASCII format, now the dynamics must be included in order to obtain the motions of each elements on each step. The results actually have two parts, one defined the data of the result, another one define what time to show the result, that is, the frame or step of the animation. The example is provided

```

1 | *RESULTS 3
2 | %DIMENSION 3
3 | %PER_NODE #3
4 | 0. 0. 0.
5 | 0. 0. 1.
6 | 0. 0. 0.
7 | 0. 0. 0.
8 | 0. 0. 2.
9 | 0. 0. 0.

```

```

10 0. 0. 0.
11 0. 0. 3.
12 0. 0. 0.
13 0. 0. -1.
14 0. 0. -3.
15 0. 0. 2.
16 0. 0. 1.
17 0. 0. -2.
18 0. 0. 1.
19 0. 0. 2.
20 0. 0. 1.
21 0. 0. 3.
22 0. 0. 0.
23 0. 0. 0.
24 0. 0. 0.
25 0. 0. 0.
26 0. 0. 0.
27 0. 0. 0.
28 0. 0. 0.

```

This is a result data of 3 dimensional for all the nodes in node group 3. This value can be scalar or can be a vector. This can be done by using command `*GLVIEWSCALAR[ID]` and `*GLVIWVECTOR[ID]`. The command selected for visualized the OSOM platform is vector value. This value, which more specifically, the wave elevation will add on the z direction of the wave carpet coordinate. A simple example will show how this works

```

1 *GLVIEWGEOMETRY 1
2 %ELEMENTS
3 1 2 3

```

The example of `*GLVIEWGEOMETRY` is listed above, This means combine element 1, 2, 3 together. The 3D model now has been set up in the VTF file in ASCII format. The dynamics must be included in order to obtain the motions of each elements on each step. The results actually have two parts, one defines the data of the result, another one defines what time to show the result, that is, the frame or step of the animation. The example is provided

```

1 *GLVIEWVECTOR 1
2 %NAME "Displacement" %1, 2, 3 in line 4, 6, 8 refer to the result IDs
3 %STEP 1
4 1
5 %STEP 2
6 2
7 %STEP 3
8 3

```

The 3 steps here mean that the animation will have 3 frames and for each step, one result that mention earlier will be executed. It can be several results that executed in the same steps, but we need to make sure that the type of results must be the same. As we know, the ship not only has linear motion, but also rotation motion. Hence we need some other command to programming the dynamics of the DP vessel. The command chosen here is called **TRANSFORMATIONRESULT #*, and each of this result block contains a 4×3 matrix. The above 3×3 area is the ship rotation matrix which generated from 20-Sim, and describes the angular displacement of the ship. The lower 1×3 matrix describe the linear displacement of the center of mass of the DP vessel in x, y and z direction. The matrix is printed by the DLL file, the details of the programming can be found in the Appendix D. The simple example will be provided below

```

1 *TRANSFORMATIONRESULT 8
2 %ELEMENT_BLOCK_ID #1
3 1 0 0
4 0 1 0
5 0 0 1
6 0 0 0
7
8 *GLVIEWTRANSFORMATION 1
9 %STEP 1
10 4, 5
11 %STEP 2
12 6, 7
13 %STEP 3
14 8, 9

```

For each animation step, there will be 2 results executed. This 2 results can be from the same element or different elements respectively. One thing need to pay attention to is no matter the sub-command *%STEP#* represents what kind of result, as long as they have the same step number, they will be executed at the same frame in animation. By using this characteristic, we can show the motion of waves and ship at the same time in different method(for wave carpet is vector results and for ship is rotation results in GLview). The overall code will be provided in Appendix D and the animation steps will shown in figure 4.13. The basic steps to build the VTF file is same for the visualization of OSOM, due to the extra huge size of the data in that VTF file. Here we use a simple and clear example to demonstrate the process. The source code of OSOM platform VTF file will be collected in the CD-ROM that delivered to the Department of Marine Technology.

4.7.4 Comparison of Two Methods of Implementation in GLview

Actually there are two methods to implement the ship and crane dynamics in the VTF file. One method is by applying the **TRANSFORMATIONRESULT* and

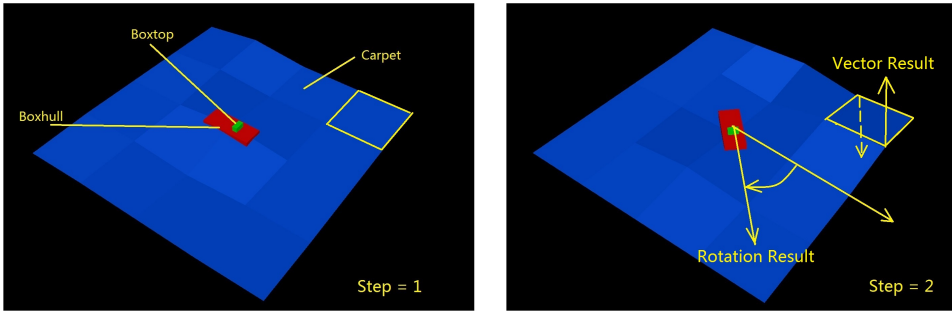


Figure 4.13: Simple example in GLview

*GLVIEWTRANSFORMATION command as mentioned in the previous chapter. Another method is by using the *GLVIEWDISPLACEMENT command. In the second method, the dynamic results of ship and crane will be converted to the position displacement for each time step, that is, the coordinates of all the positions in the model will be updated by each time step. These two implement methods will have the same visual result, however, the visual file size of the second method will much bigger than the first method. For instance, the size of the 3D visualization file in a 30 seconds simulation is up to 2 GB instead of 20 MB by applying the first method. The reason is there are lots of nodes defined in the complex ship and crane 3D model. The number can be up to one million. If updating all of them for each time step, the file size will be expanded dramatically. In the method that chosen for in this thesis, for each time step in the simulation, only one rotation matrix for the ship and three rotation matrices for the crane are needed. These rotation matrices will be applied to all the nodes in the controlling block, thus avoid repeating generating all the nodes that form the 3D model. This will increase the efficiency of simulation and visualization dramatically.

Another advantage of using the first method is that the result matrices that needed in generating the VTF file can be imported through the DLL file directly from the simulation in 20-sim. This is done by adding the matrix results variables in the `inarr[]` function and `outarr[]` function. If applying the second method here, the coordinate of nodes in 3D model should be calculated externally and print to the VTF file through DLL file or other approaches.

4.8 Visualization Results and Studies

Here is the finished 3D visualization of OSOM platform, which is shown in figure 4.14. As mentioned before, the vessel here is used the compact and simpler one than the model export from Delft Ship. The vessel model is in figure 4.12. Please

notice that, the ship model file (VTF) should be located with the same folder of the generated visualization file, otherwise you cannot actually read the model file and this will cause some error messages.

4.8.1 Case One

The case one will show and compare the 3D visualization results together with the information of vessel position in 20-Sim under different vessel heading. The headings are set to 0.2 rad and 0.0 rad respectively, and the environment condition is the same. The parameters and setting values are collected in the table 4.2. The JONSAWP spectrum is used here since most of the PSVs or CSVs will operate at Norwegian Continental Shelf(NCS) in North Sea.

Table 4.2: Parameters and settings of OSOM visualization case 1.1

Setting Parameters		Environment Parameters	
Name	Value	Name	Value
Ship heading	0.2 rad	Spectrum peak period(T)	8 s
Length of wave carpet(L_{hav})	400 m	Significant wave height(H)	6 m
Width of wave carpet(B_{hav})	200 m	Spectrum shape parameter(γ)	3 s
Wave element number	100 m	Lowest wave period component	3 s
Animation frame skip number	50	Largest wave period component	21 s

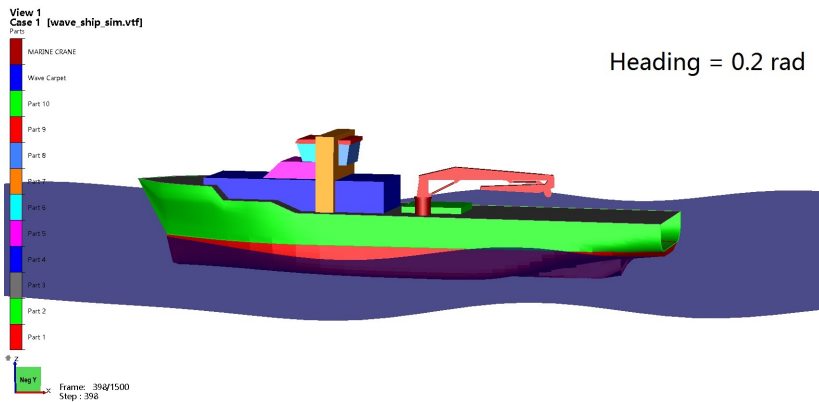


Figure 4.14: 3D visualization from y direction

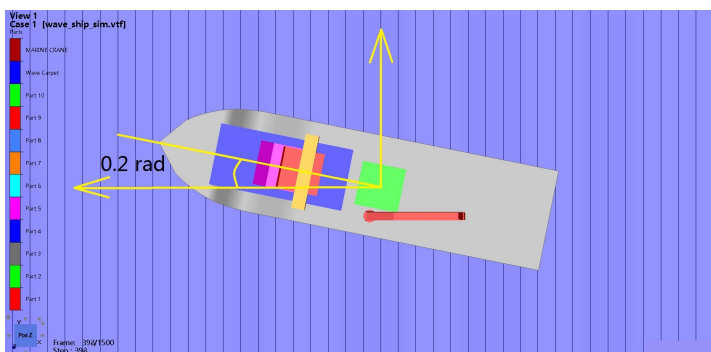


Figure 4.15: 3D visualization from z direction

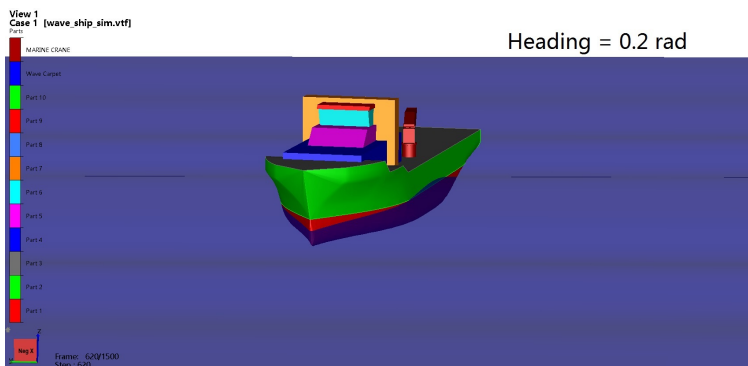


Figure 4.16: 3D visualization front view

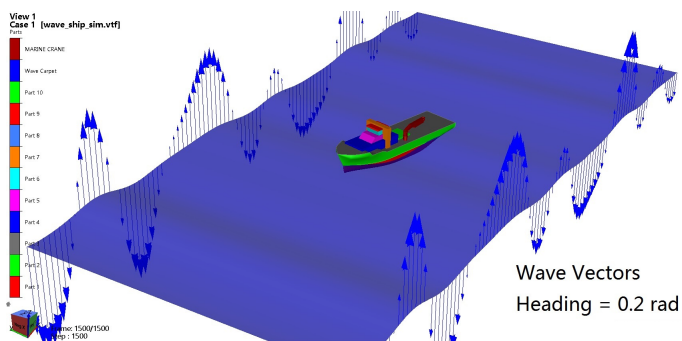


Figure 4.17: 3D visualization with wave vectors

Now, the vessel with heading 0 will be shown here as well. The parameters and settings are shown in table 4.3.

Table 4.3: Parameters and settings of OSOM visualization case 1.2

Setting Parameters		Environment Parameters	
Name	Value	Name	Value
Ship heading	0.0 rad	Spectrum peak period(T)	8 s
Length of wave carpet(L_{hav})	400 m	Significant wave height(H)	6 m
Width of wave carpet(B_{hav})	200 m	Spectrum shape parameter(γ)	3 s
Wave element number	100 m	Lowest wave period component	3 s
Animation frame skip number	50	Largest wave period component	21 s

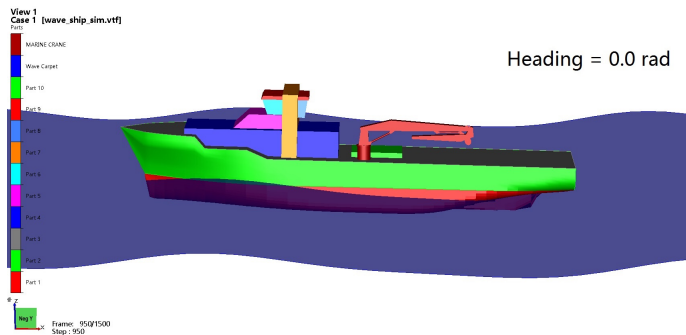


Figure 4.18: 3D visualization from y direction

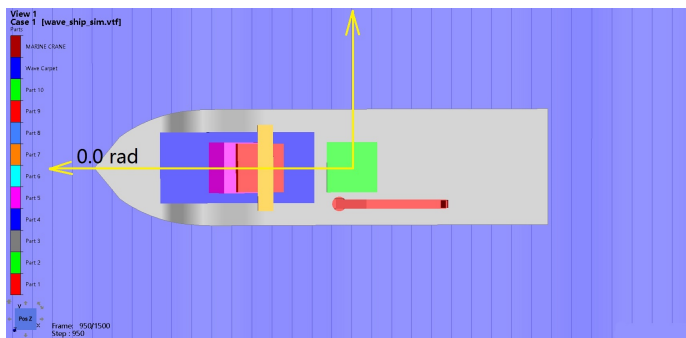


Figure 4.19: 3D visualization from z direction

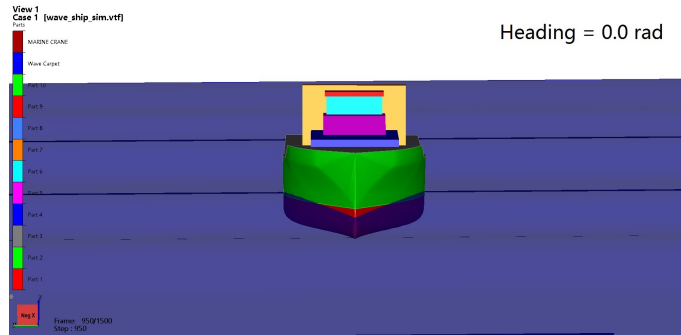


Figure 4.20: 3D visualization front view

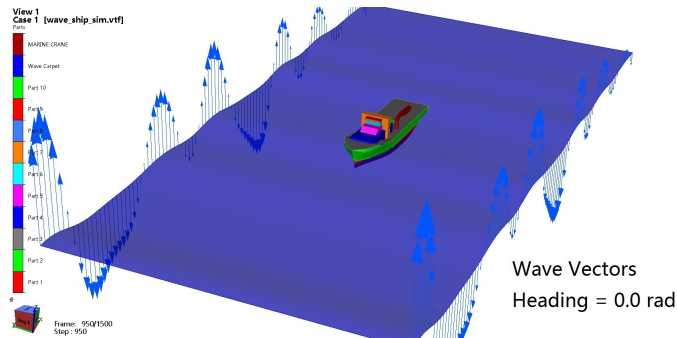


Figure 4.21: 3D visualization with wave vectors

The visualization is shown in pictures here. The animation file in MPG format and in VTFx format (For GLview Express only) will be included in the CD-ROM that delivered together with this master thesis. From the figures we can easily see the heading change in different visual angles. Here the position of the central of gravity of the DP vessel will be provided in figure 4.22 and 4.23 under different heading angle. These figures show that on both headings the DP system works well, on x, y and z direction the displacement range is around $0.5m$. For 0.2 rad heading, the surge motion is a bit stronger than the 0.0 rad heading. This is due to the ship has more project area towards the wave prorating direction. And this result can be seen through the 3D visualization as well. The roll, pitch and yaw angle of heading 0.0 rad and heading 0.2 rad is shown in figure 4.24 and figure 4.25 respectively. The dynamics of crane under different environments or initial state are provided in figure 4.26 for case 1.2 and in figure 4.27 for case 2.2.

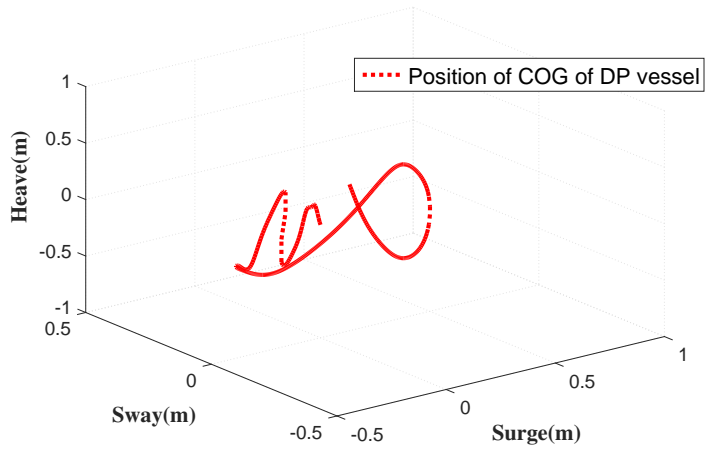


Figure 4.22: Position of COG in case 1.1

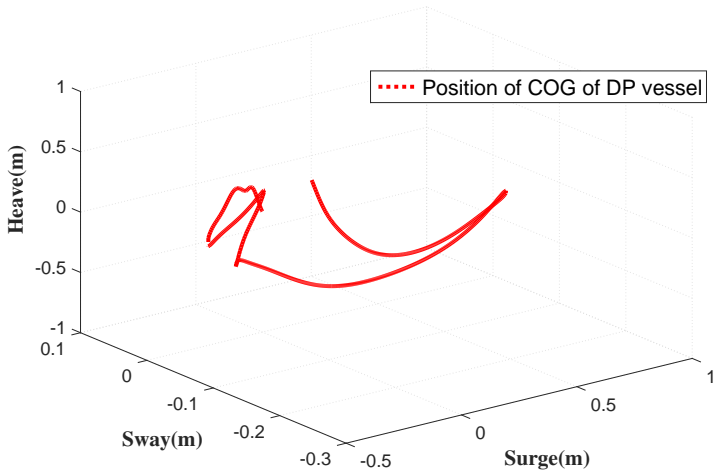


Figure 4.23: Position of COG in case 1.2

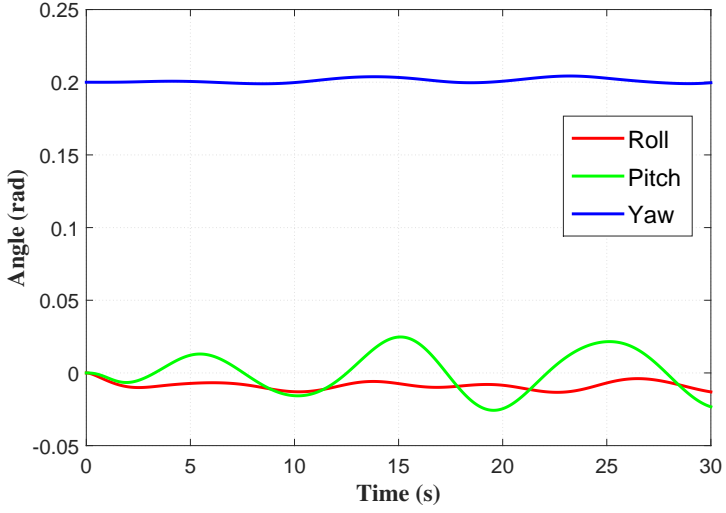


Figure 4.24: Roll, pitch, yaw angles in case 1.1

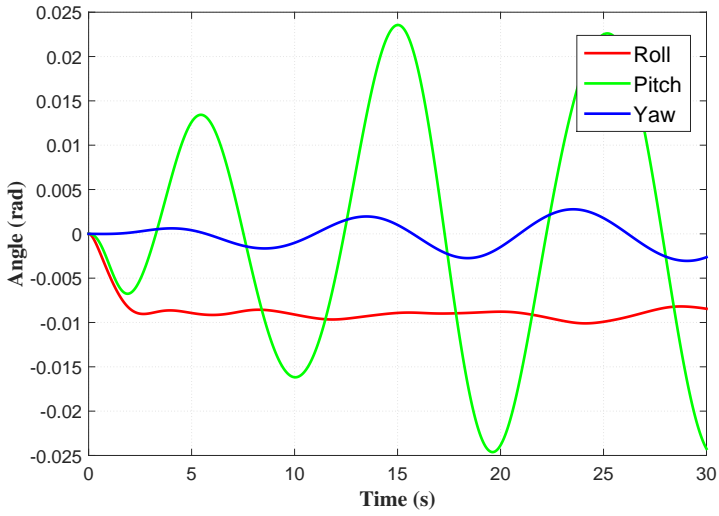


Figure 4.25: Roll, pitch, yaw angles in case 1.2

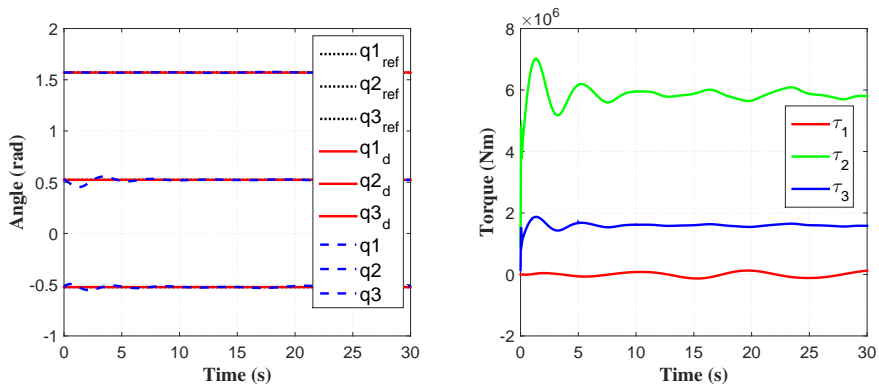


Figure 4.26: Crane dynamics in case 1.1

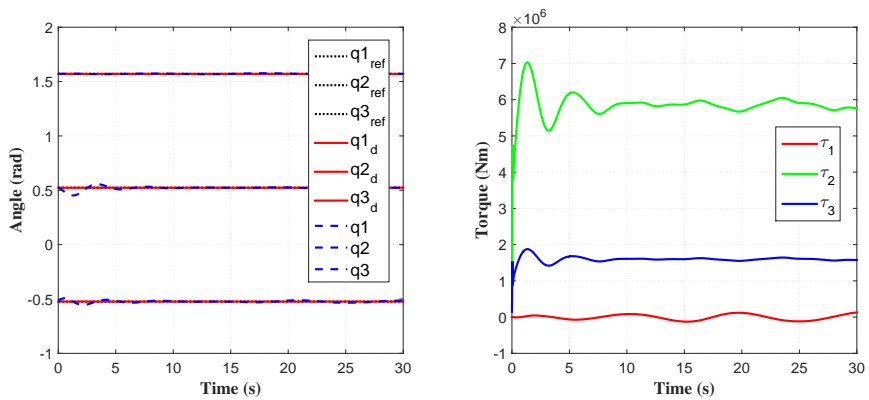


Figure 4.27: Crane dynamics in case 1.2

4.8.2 Case Two

The case two will evaluate the situation that under different wave conditions. Only main visual view figures will be provided here since most of the structure is same with case one. The parameters and settings is provided in the table 4.4 and table 4.5 for different wave conditions respectively.

Table 4.4: Parameters and settings of OSOM visualization case 2.1

Setting Parameters		Environment Parameters	
Name	Value	Name	Value
Ship heading	0.0 rad	Spectrum peak period(T)	10 s
Length of wave carpet(L_{hav})	400 m	Significant wave height(H)	10 m
Width of wave carpet(B_{hav})	200 m	Spectrum shape parameter(γ)	3 s
Wave element number	100 m	Lowest wave period component	3 s
Animation frame skip number	50	Largest wave period component	21 s

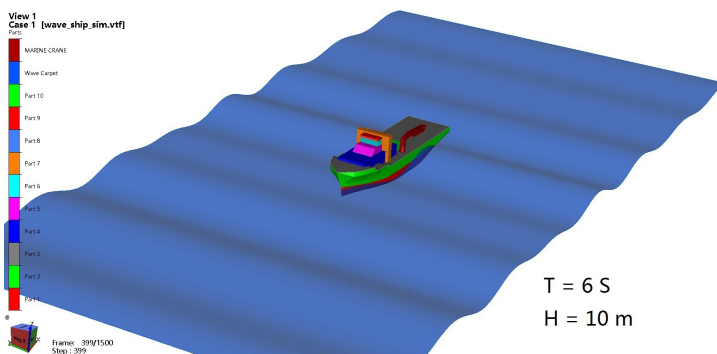


Figure 4.28: 3D visualization in GLview

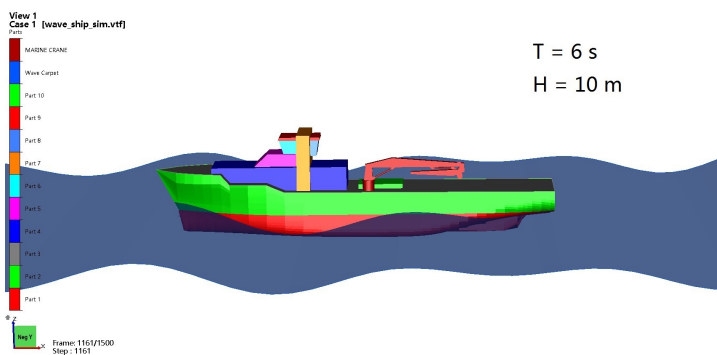


Figure 4.29: 3D visualization in GLview

Table 4.5: Parameters and settings of OSOM visualization case 2.2

Setting Parameters		Environment Parameters	
Name	Value	Name	Value
Ship heading	0.0 rad	Spectrum peak period(T)	10 s
Length of wave carpet(L_{hav})	400 m	Significant wave height(H)	4 m
Width of wave carpet(B_{hav})	200 m	Spectrum shape parameter(γ)	3 s
Wave element number	100 m	Lowest wave period component	3 s
Animation frame skip number	50	Largest wave period component	21 s

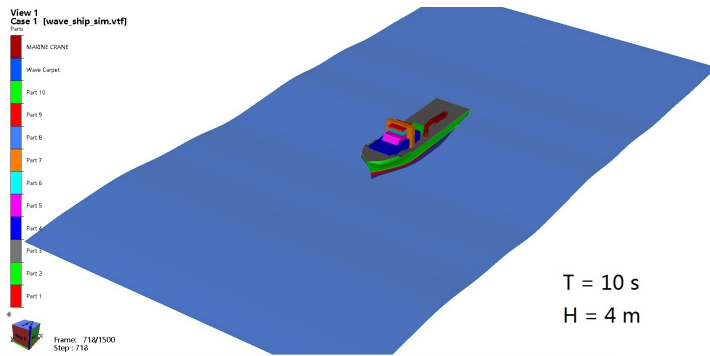


Figure 4.30: 3D visualization in GLview

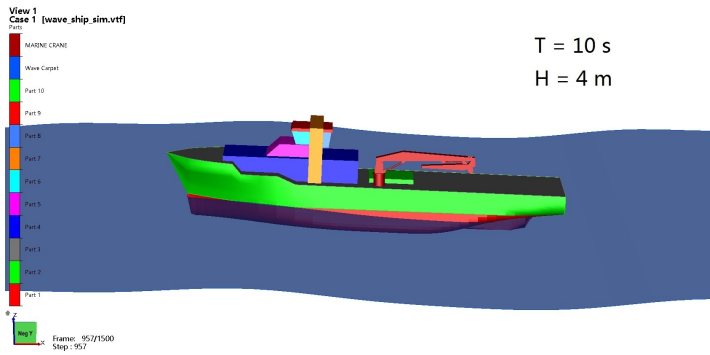


Figure 4.31: 3D visualization in GLview

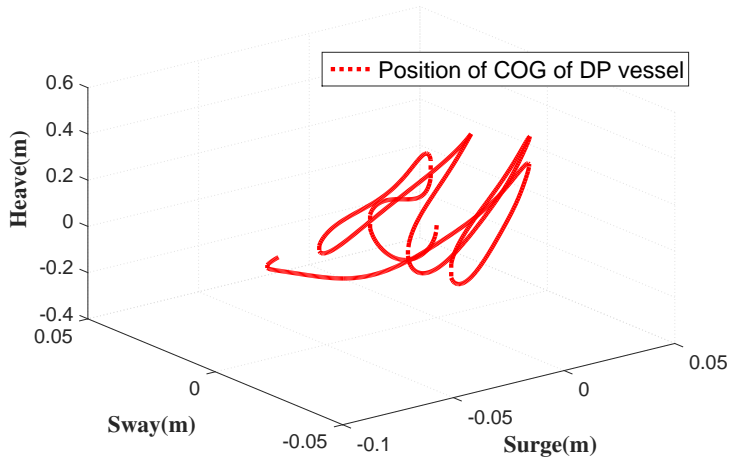


Figure 4.32: Position of COG in case 2.1

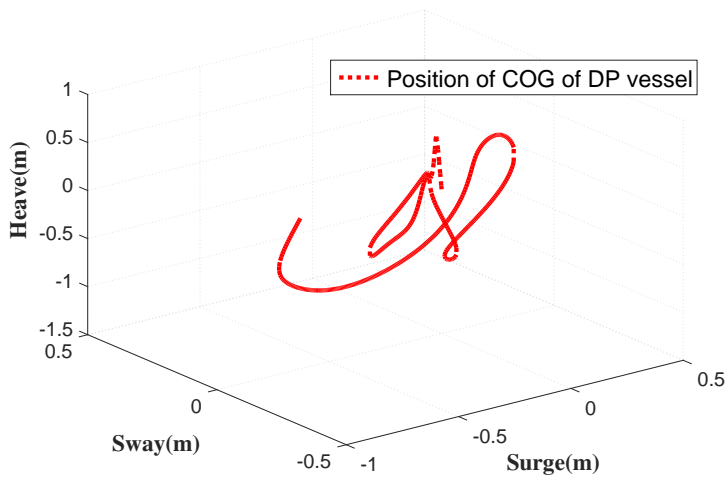


Figure 4.33: Position of COG in case 2.2

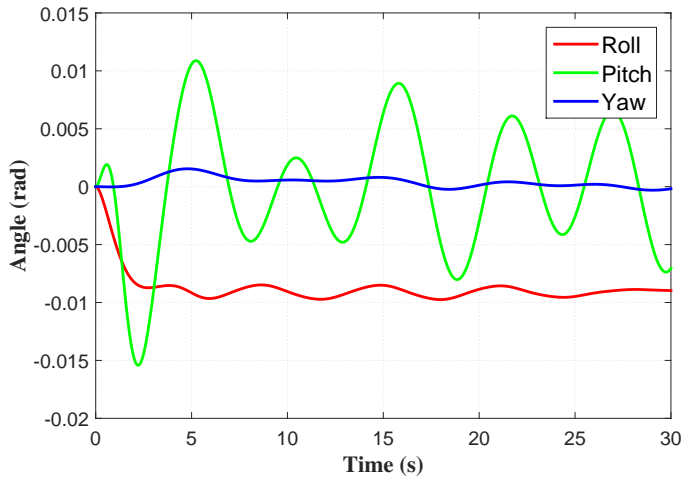


Figure 4.34: Roll, pitch, yaw angles in case 2.1

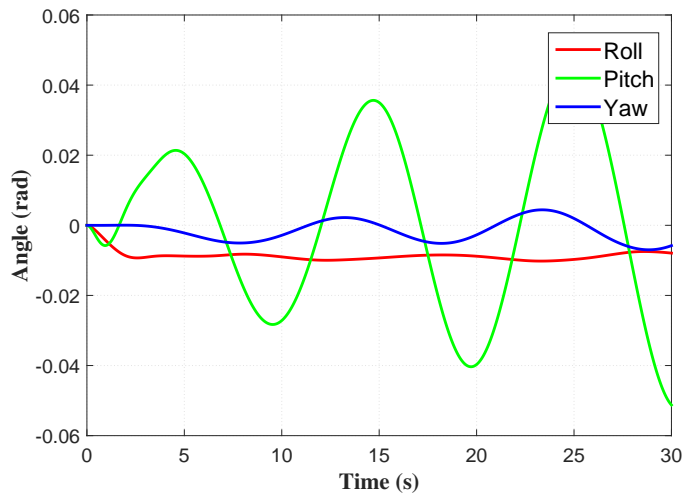


Figure 4.35: Roll, pitch, yaw angles in case 2.2

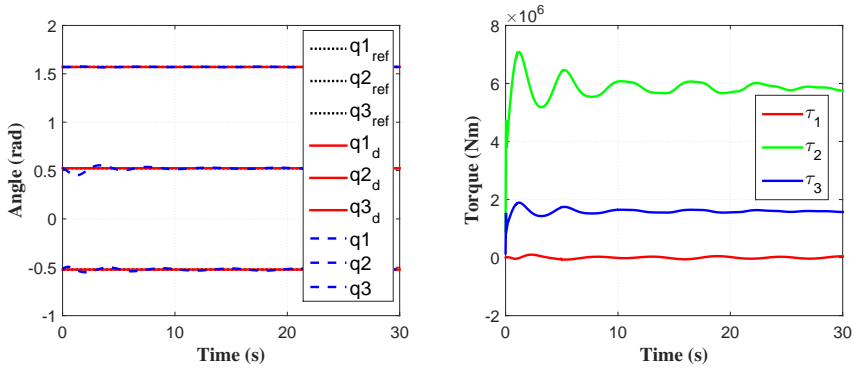


Figure 4.36: Crane dynamics in case 2.1

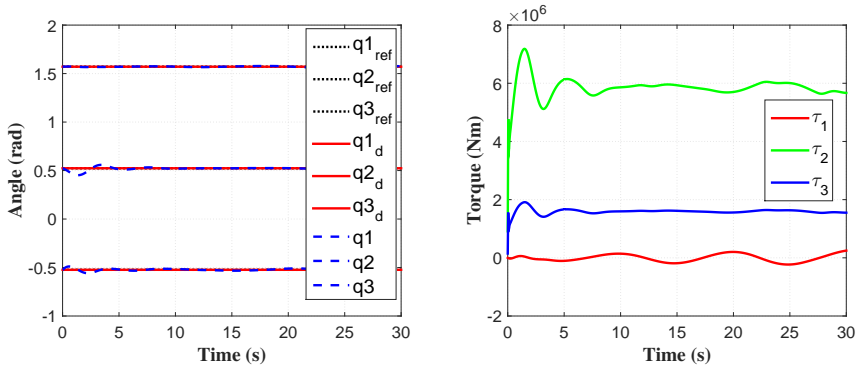


Figure 4.37: Crane dynamics in case 2.2

Combining the results from both 3D visualization and data plot we can find the surge motion, sway motion, roll angle and yaw angle are quite close, but the heave motion and pitch angle has relative big difference. In the case $T = 10$ (s) and $H = 4$ (m), both heave motion and pitch angle is bigger than the case with $T = 10$ (s) and $H = 10$ (m).

4.8.3 Real Time Visualization

The main part of 3D visualization of OSOM has been done. Some settings and further works that can be done to increase the quality of 3D visualization of OSOM will be raised in the next sections in this chapter.

The first setting which is a key point in develop a simulator is how to make the visualization real-time. In order to achieve this, the time step in 20-Sim, the number of skipped frame, and the FPS in GLview should be taken into consideration.

For instance the time step size in 20-Sim simulation is 0.0004 (s). If the simulation runs 1 second, then there will be 2500 results or steps. Another parameter "skip" in 20-Sim which represent that C code will write 1 step per skip number of steps. For example, here the value of skip is set to 50, this means only 1 step or frame will be shown in visualization per 50 steps which generated from 20-Sim. This setting can help reducing the unnecessary steps, also size of visualization file. After scaling down the the steps from 20-Sim, the new steps per second is $2500/50 = 50$ now.

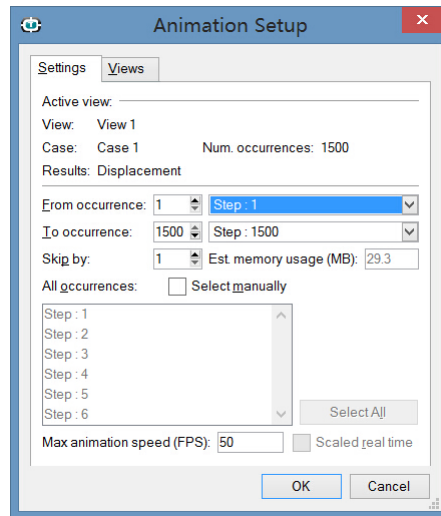


Figure 4.38: FPS setting in GLview

The data in the VTF file for GLview will have the same steps after scaling down, which is 50 steps (will use 1 second to visualize it, same with the simulation time) per second. Now, the concept of FPS should be explained. FPS, which means Frame Per Second, is the frame rate or frame frequency. The human eye and its brain interface, the human visual system, can process 10 to 12 separate images per second, perceiving them individually, this information is from Read and Meyer (2000). Hence, in order to keep the visual system of human being believe the film or animation is continuous, the FPS should be higher than at least 12. Here we have 50 steps per second after scaling down by DLL file. In order to make it a real-time visualization, the FPS need to set to 50 in the GLview. The command window is shown in figure 4.38.

4.8.4 Further Works for Visualization of OSOM

The methods and main structure has been already discussed and explained in this chapter, however, there are still many aspects can be added and improved. Several close related topics will be proposed here.

On-line Visualization

The on-line visualization that can transfer the data from joystick or simulator to visualization directly, which may exist a very small time delay. Or more specifically, how to write and utilize the VTF file at the same time. This is different from post process visualization and is the key point to build the on line simulator. Several possible approaches are here, for instance, if the GLview can use the VTF file even when DLL file is still writing on it. It will be of great advantage if the on-line visualization can be done through the inter-software simulation.

Short-Crest Waves

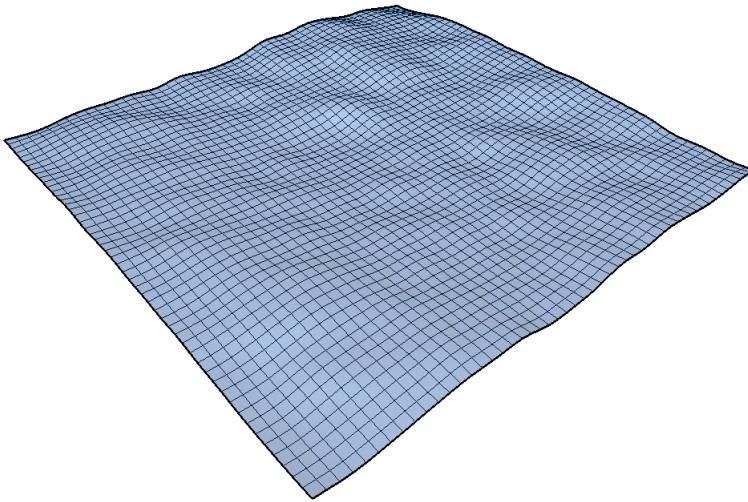


Figure 4.39: Short-crest wave model in GLview

The wave implemented here is the long-crest irregular waves, which is the same wave that choose for the simulation in OSOM. The wave is good enough to test the dynamics of marine vessel and marine crane. However, it is still little different from the reality, which most of the sea state is called "short-crest" waves. The different between the long-crest wave and short-crest wave is the short-crest wave has more than one wave sources, or in more demotic words, it is a superposition of many

different waves from different direction. The wave implemented here in GLview as is shown in figure 4.39.

Chapter 5

Crane Hydraulic Sub-systems in OSOM

This chapter will present the marine crane hydraulic system modelling based on bond graph method. The marine crane hydraulic system mainly contains hydraulic power system (high pressure system), hydraulic actuator system and control system. The models are developed in 20-sim. Modelling approaches, components characteristics, system dynamics and simulation results will be discussed in details.

5.1 Modelling the Crane Hydraulic System

The crane used in the industry are usually electrically or hydraulic driven. The electric powered crane will mount the electric motor on each joint and actuate the crane arms. This design has many applications on small scale crane or manipulator that lifting light load. The crane that powered by hydraulic system will be larger because the system includes a tank, a pump, valves, pipelines and hydraulic cylinder or motor. Nowadays the marine crane are mostly powered by hydraulic system, which is more stable, safer and with high lifting capability. However, the hydraulic system has complex dynamics with non-linear properties based on the fluid dynamics. A good model is, can catch the main properties that important to the operations and studies, not the model that including all the details even not necessary. Thus, some assumptions and simplifications are made. These simplifications are still be able to describe the physical models and efficiently enough to predict the system behaviors

accurately. In addition, the level of model complexity can be easily increased by need in bond graph method. Most of the elements in the hydraulic system developed here are modeled by using basic bond graph elements instead of using hydraulic library in 20-sim.

The components and structure of marine crane are shown in figure 5.1. This marine crane has three degree of freedom, which come from three low pair joints. They are the crane rotation base, crane boom joint and crane jib joint respectively. There is a winch system mounted on the crane for lowering and hoisting the load during marine operations. The rotation crane base is powered by a swing hydraulic motor. The crane boom and crane jib are actuated by hydraulic cylinder, which will control the motion of crane. The details of crane geometry relationship and working space limitation will be discussed in the later section. The system layout of the hydraulic

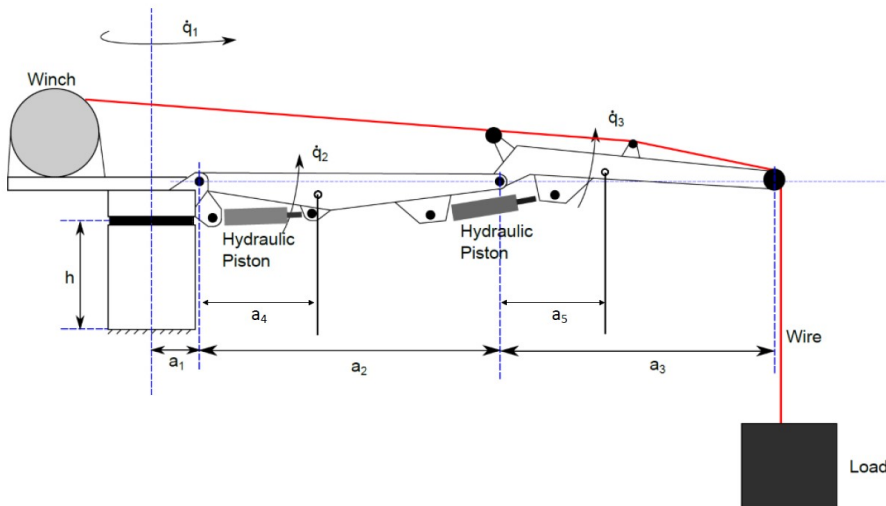


Figure 5.1: Marine crane components and structure

system of one piston is shown in figure 5.2. The layout almost same for both pistons in crane hydraulic system. This layout is also similar to the hydraulic swing motor, the differences are the piston now is replaced by a hydraulic motor and the pressure in the swing motor system will be lower. This is because for crane rotation, the inertial forces and friction forces will have much more influence on the crane base than the loads.

In the hydraulic system shown in figure 5.2, the main components including hydraulic pump, system accumulator, a check valve, a safety valve (pressure relief valve),

pressure compensating valve, 4/3 directional control valve, hydraulic cylinder, load, bumper, logic box and control system. The overall layout of hydraulic system for the whole crane is provided as schematic drawing in figure 5.3. The control loop, actuators and crane are linked as a complete system. The details of modelling approaches and properties of each component will be discussed next.

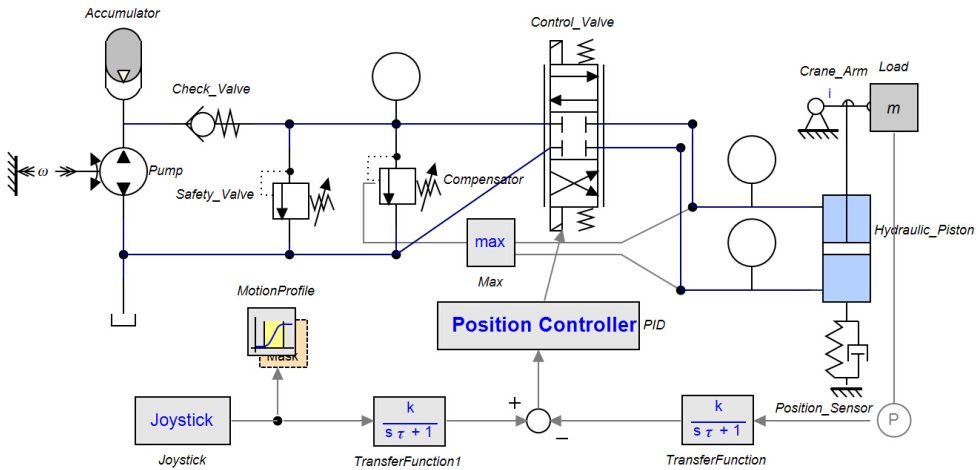


Figure 5.2: System layout of hydraulic cylinder

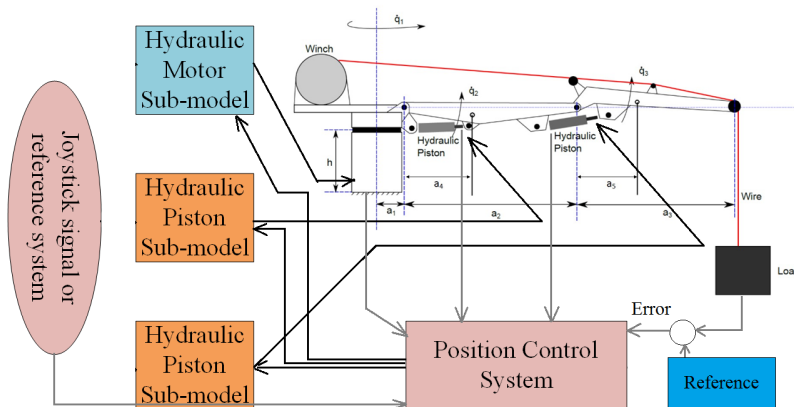


Figure 5.3: System layout of crane hydraulic system

5.1.1 Hydraulic Pump

The pump is the fluid source, the flow will be generated here. There are two types of pump, one is the displacement pump and another one is the rotor dynamic pump. There are two types in the displacement pump as well, which are reciprocating pump and rotary pump. The rotary pump will generate more stable flow than the reciprocating pump, which is used here. The flow rate of the pump, or Hydraulic Power Unit(HPU) can be calculated as

$$\dot{V}_p = \frac{V_R}{2\pi} \omega_P \tag{5.1}$$

Where the \dot{V}_p is the flow rate generated from HPU, V_R is the volume displacement per radian. One of the important properties of hydraulic pump is the leakage through it, especially in high pressure system. The leakage can be modeled as a *MR* element, it needs the input signal of the pressure generated in HPU to calculate the flow loss. In bond graph, the pump can be modeled as following. The *Sf* element here

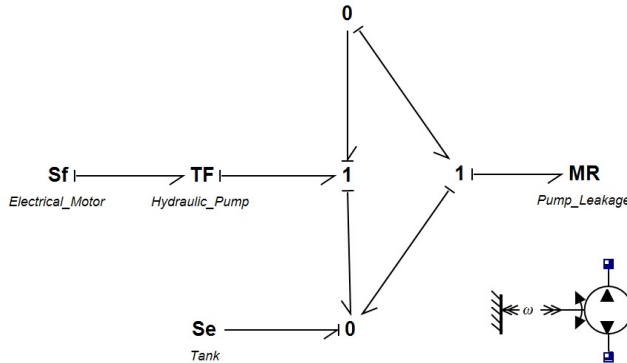


Figure 5.4: HPU in bond graph

represents the prime driver, which can be a electrical motor. The effort in bond graph, which is torque here will be transfer by *TF* element to pressure. The flow in bond graph, which is angular velocity here will be transfer to flow rate. There is a *Se* element connected with 0 junction that represent the hydraulic tank, which be given a constant pressure at 1 bar. In the hydraulic model that developed in this thesis for OSOM, the HPU is modeled as a constant flow pump here. The reason for doing this is to simplify the structure of the model since electrical power transfer is not the topic that highlight here.

5.1.2 System Accumulator

The bladder-type accumulator is used here for the whole system. Another type of fluid capacitor is used for hydraulic system is based on the volume of compressible fluid, as the fluids is not rigid. The bulk modulus will be implemented here, bulk modulus β represents the compressibility of the fluid. First the bladder-type accumulator will be demonstrated. The bladder-type accumulator has a chamber with measurable

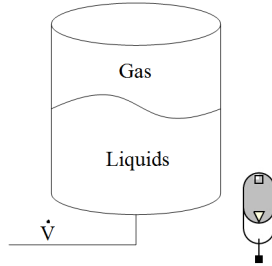


Figure 5.5: Bladder-type accumulator

volume oil under pressure with gas charged. In this type of accumulator, the chamber is separated by a bladder into oil chamber and gas chamber. Since the gas above will be compressed by the liquid when charging, several state conditions should be taken into consideration. The compression or expansion can take place under isothermal, isentropic or polytropic. The selection of them will depend on the compression or expansion speed. In rapid process, the isentropic condition will be used and if the process is slow, the isothermal condition will be used. The isothermal process has the constitutive law as

$$P = \frac{P_0 V_0}{V_0 - \int \dot{V} dt} \quad (5.2)$$

If the condition is polytropic, the constitutive law here should be

$$P = \left[\frac{P_0 V_0}{V_0 - \int \dot{V} dt} \right]^k P_0 \quad (5.3)$$

The system accumulator is modeled as a C element in the bond graph with the corresponding constitutive law implement inside. The condition in this thesis is assumed polytropic with the specific heat ratio $k = 1.4$. The accumulator will decide a lot on the stiffness of the whole system. If designed with smaller volume, the time step for simulation will be smaller, thus more time consuming on simulation. A complex hydraulic system can result in an extremely slow simulation, which is not recommended in building a real time simulator.

5.1.3 Check Valve

Check valve, or one way valve is the valve that control the flow only go through one direction. In the hydraulic system here, this valve will protect the hydraulic pump from inverse flow, which may cause from the change of directional valve when hoisting and lowering loads. The inverse flow may damage the pump as the pump is still working in the positive direction. The check valve implementation in 20-sim is shown in figure 5.6. P_a is the upstream pressure and P_b is the downstream pressure.

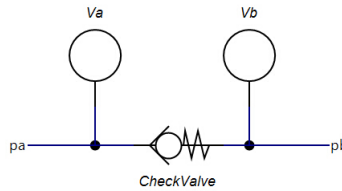


Figure 5.6: Check valve(20-sim lib, Kleijn and Groothuis (2014))

The difference of them is called back pressure, which is modeled as dp inside of this element. The setting pressure that makes the valve open is called cracking pressure, which is modeled as P_{close} inside this element. This means there is no flow pass under this pressure. The pressure that enables the valve fully open is called P_{open} in this element. The leakage when the valve is closed is also considered here.

5.1.4 Relief Valve

The relief valve, or safety valve, is the valve can limit the system pressure that build up in the upstream. The valve will open under the designed pressure in order to protect the pipeline and instrument from being damaged by the high pressure in the hydraulic system, especially in the high pressure system that developed in this thesis. When the pressure is higher than the per-designed value, the value will open and divert to the fluid flow to the reservoir or tank. When the relief valve is opening, the flow path through the relief valve will be the "least resistance oath" in the system. In practical, once the relief valve is opened, it will remain opening to ensure the safety of the hydraulic system until operator to reset or repair the valve. The structure of the pressure relief valve is shown in figure 5.7. The flow that with inlet pressure will add a force on the valve piston. When the pressure is big enough, the force act on the piston will open the seat disc on the nozzle, thus the flow can return to the tank. The opening pressure will depend on the stiffness of spring and the valve piston area. This can easily be calculated as

$$P_{in}A_{pist} = X_0k \quad (5.4)$$

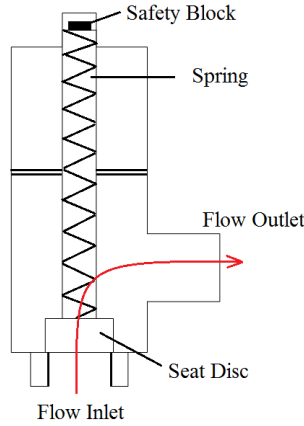


Figure 5.7: Relief valve layout

Where the X_0 is the pre-compressed displacement of the spring. This will determine the force act on the valve piston. K is the stiffness of the spring. P_{in} is the controlled pressure in the upstream, and A_{pist} is the area of valve piston. The way of modelling

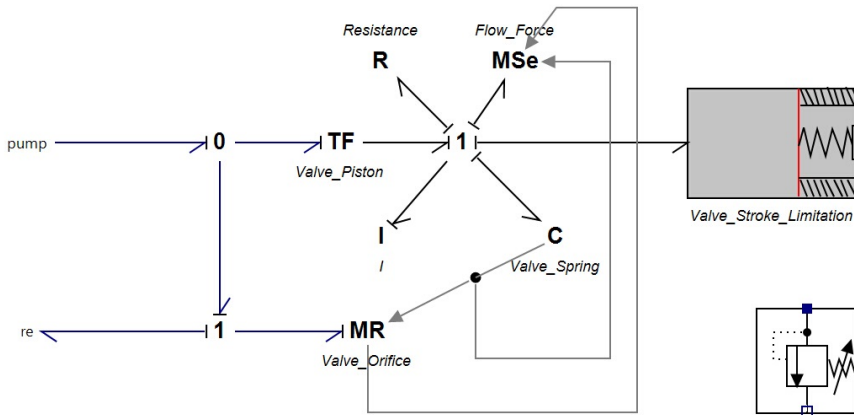


Figure 5.8: Bond graph model of relief valve

relief valve in bond graph is shown in figure 5.8. The TF element represents the piston area that transfer pressure to force. R element represents the fluid resistance in the valve. C element is the spring, the stiffness can be set inside this element. I is the inertial of the valve piston since it has mass. The MR element represents the valve nozzle, which decides the flow rate through the valve. The constitutive law

here is

$$Q = \alpha(x_{pist})A_{op}sign(\Delta p)\sqrt{\frac{2}{\rho}|\Delta p|} \quad (5.5)$$

Where the $\alpha(x_{pist})$ is the flow coefficient. A_{op} is the opening area, it depends on the seat disc area and the disc opening displacement. Δp is the pressure difference between inlet port and outlet port. The *MSe* element represents the flow force act on the valve piston. The flow force is 0 when the valve is not opening. When the valve open, the force can be calculated based on the following equation

$$F = sign(Q)\frac{k(x_{pist})\rho Q^2}{A_{op}(x_{pist})} \quad (5.6)$$

This constitutive law is implemented in the *MSe* element, the flow rate Q and the displacement of valve piston x_{pist} is collected as input signals from *C* element and *MR* element. The last element in the relief valve model is the bumper, or the restriction of the valve piston or seat disc displacement. The bumper can limit the minimum and maximum displacement of the seat disc, which limits the opening area as well. The bumper can be regarded as a combination of dumping and spring system, where the damping ratio and spring stiffness is very high. When the piston hit the bumper, it will stop within a negligible distance. In the relief valve model, the bumper limits the piston stroke within $q_{min} \leq x_{psit} \leq q_{max}$.

5.1.5 Compensate Valve

For the structure, the pressure relief valve and compensate valve or compensator is similar. The difference is that the compensator also has pressure force act on the same side with the spring force. This pressure is called pilot pressure. The pilot pressure is the max pressure of the pressure difference before and after hydraulic actuator in the system. Thus a logic box decided the maximum pressure or which pressure should the pilot pressure be. The compensator has higher flow through ability than the relief valve. It is self-controlled for open and close the seat disc without resetting it every time as pressure relief valve. The compensator will ensure the pressure in the main system adjusting itself only few bars higher than the pressure that needed to lifting the load. The spring stiffness decides how many bars pressure higher than the needed system pressure here. The main closing force acts on the valve piston is the pilot pressure. Thus, the system pressure now is controlled as $P_{sys} = P_{pilot} + k_c x_0$, the $k_c x_0$ can be 3 bars here. The basic constitutive law are same as in the relief valve. The bond graph model is shown in figure 5.10. The pilot pressure is the pressure that selected by the logic box based on the pressure difference between each side of the hydraulic actuator. The *R* element on the pilot path is the nozzle effect, the flow passes through it can be described as

$$Q = C_d A_{op} sign(\Delta p)\sqrt{\frac{2}{\rho}|\Delta p|} \quad (5.7)$$

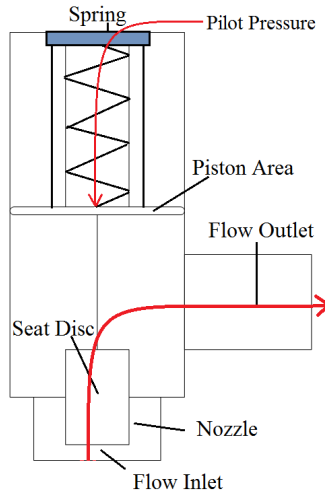


Figure 5.9: Compensator layout

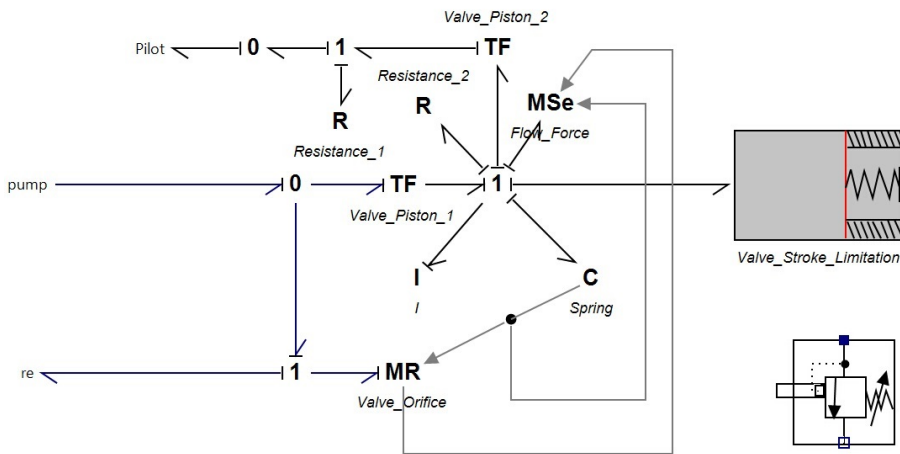


Figure 5.10: Bond graph model of compensator

Where C_d is the flow coefficient. A C element can be add on the 0 junction on the pilot path represent the volume in the nozzle, which will be filled with pilot fluid. This C is neglected here as the valve is connected with the accumulator, while the volume here can be consider as part of the volume of accumulator.

5.1.6 4/3 Directional Control Valve

The 4/3 directional valve or pilot valve is driven by the electromagnet. The response and execute time can be assume very fast compare to the rest of the system, thus we can simplify the dynamic of valve as instantaneous. The error brought to the model by simplification is small. The bond graph model and symbol of 4/3 directional

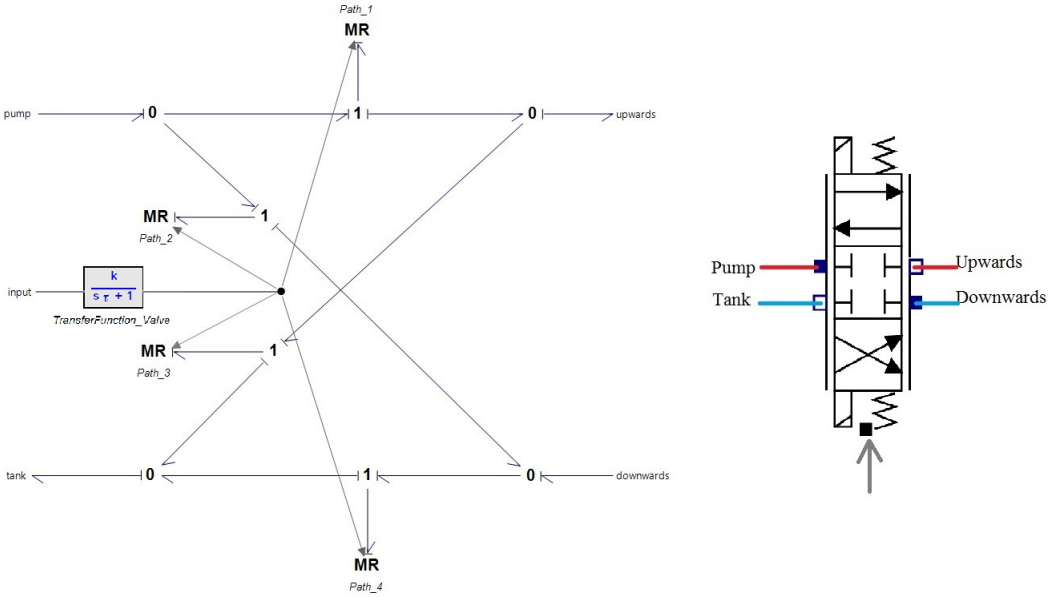


Figure 5.11: Symbol and bond graph model of 4/3 directional valve

valve is shown in figure 5.11. 4/3 means 4 ports and 3 ways. There are four ports connected to this valve. They are port "Pump", which connect to the pump, or HPU. A tank port, which connect to tank under atmospheric pressure. "Upwards" port connects to the piston chamber A that can lifting the load with hydraulic oil filled in or the positive direction of a hydraulic motor. The last port "Downwards" connects to the chamber B which will lower the load with fluid in. There are three combinations of path through the valve, which can lifting, lowering, and stop motion respectively.

The valve path is modeled as *MR* element that receive the signal from position controller. The *MR* element can decide if the path is open or close, if open, the opening area as well. The constitutive law inside the *MR* element is

$$Q = C_d(r)A_v(r)sign(\Delta p)\sqrt{\frac{2}{\rho}|\Delta p|} \tag{5.8}$$

This equation is the same as the nozzle function in compensator that explained before. The differences are opening area A_v and the flow coefficient C_d are determined by the signal r from the controller. The A_v is limited by the minimum and maximum stroke of valve seat disc. The path 1 and path 4 will only work when the control signal is bigger than zero, otherwise they will keep closing. Correspondingly, the path 2 and path 3 will only work when the signal is smaller than zero. However, the variable opening area will not affect by the sign as the absolute value is used when describe the opening area A_v . For more accurate modelling method, the CFD analysis results or data sheets from the manufacturer are needed.

5.1.7 Hydraulic Cylinder and Motor

Hydraulic Cylinder

The hydraulic cylinder with piston inside is the actuator in the system that transfer the hydraulic energy to mechanical energy. When lifting the load, the high pressure fluid will flow into the chamber A, and fluid will discharge from chamber B to the tank. The structure and symbol of hydraulic cylinder is shown in figure 5.12. In the

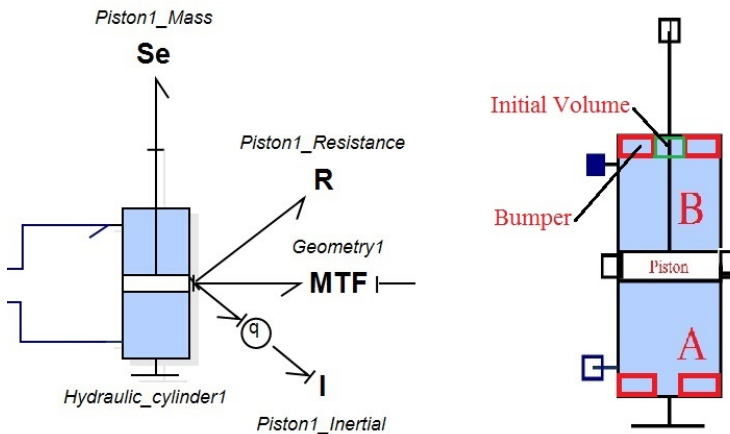


Figure 5.12: Symbol and structure of hydraulic cylinder

figure, the MTF represents the geometry transformation relations between hydraulic cylinder and crane arm. The details of geometry relations will be discussed in section 5.1.11. Please notice that in practical, each of the chamber in cylinder will not become dry even in the maximum or minimum stroke. Hence the initial volume should be considered here. The initial volume will also reduce the pressure oscillating in cylinder in the beginning stage. The piston cannot exceed the limitation of stroke. Therefore, the bumpers are used here on each end of the hydraulic cylinder. The

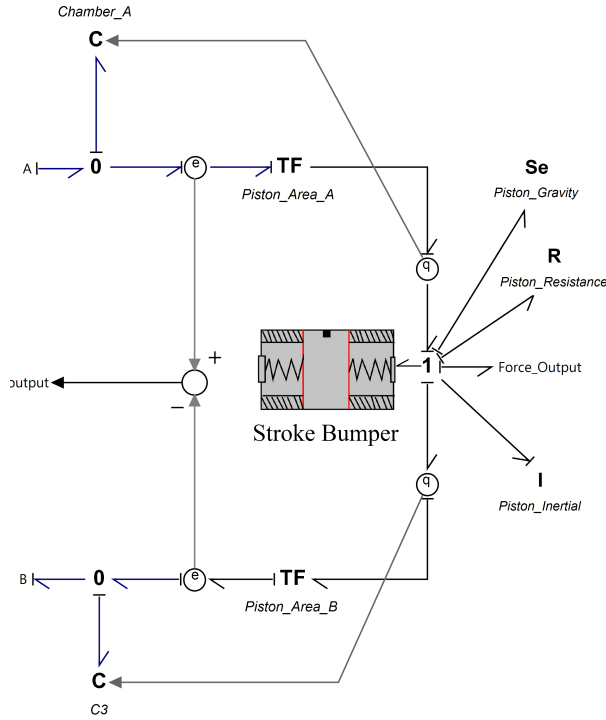


Figure 5.13: Bond graph model of hydraulic cylinder

chamber is modeled as C element, with constitutive law in chamber A and B as

$$P_a = P_0 + \left(\frac{\beta}{(V_0) + A_{pist}x_p} \right) V_{ca} \tag{5.9}$$

$$P_b = P_0 + \left(\frac{\beta}{(V_0) + A_{pist}(S - x_p)} \right) V_{cb} \tag{5.10}$$

Where P_a and P_b is the pressure in chamber A and chamber B respectively. V_0 is the initial volume that mentioned earlier. A_{pist} is the piston area. x_p is the load position, or piston position related to the original point. S is the maximum length of piston stroke and V_{ca} , V_{cb} are the volume in chamber A and chamber B in different time. The piston position is modeled as output signal from the q sensor. The differential pressure between chamber A and chamber B is calculated and sent to compensator to control the pressure only few bars higher than needed within the design range. On the 1 junction that represents the piston, the mass of piston, inertial of piston, force output to the crane arm, and piston friction force are connected. This can be seen from the figure 5.12.

Hydraulic Motor

The basic principle of hydraulic motor is similar with hydraulic cylinder. The differences are:

- The *TF* element or transformer in hydraulic motor is not the area but the volume displacement per radian.
- Less leakage than the hydraulic cylinder since the pressure in swing motor is much lower.
- The hydraulic motor is running with high rotation speed, in order to rotate the crane, a gearbox is needed.
- No bumper installed since the rotation base can rotate 360° . The bond graph model and symbol of hydraulic motor is provided in figure 5.14. The *MR* element represent the inner leakage fo the hydraulic motor. The gearbox, which modeled as *MTF* will placed after the volume displacement transfer modulus and before connect to the crane rotation base.

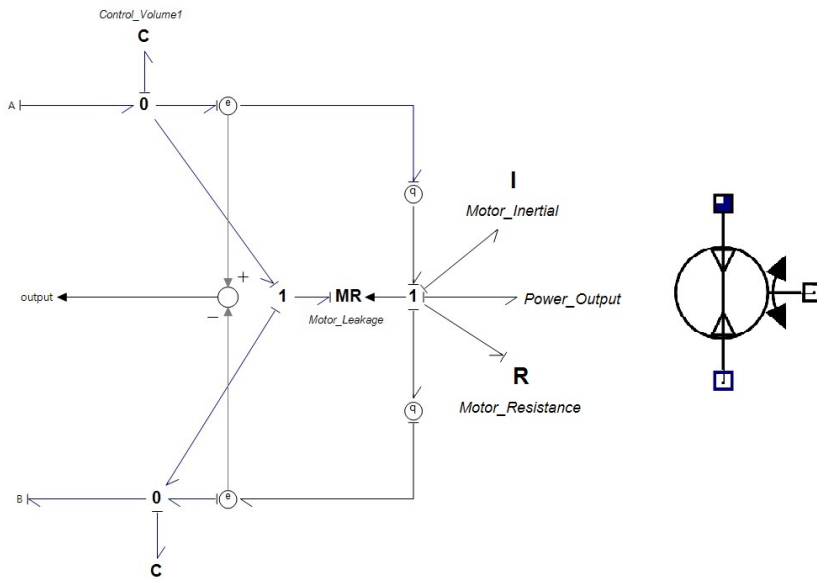


Figure 5.14: Bond graph model and symbol of hydraulic motor

5.1.8 Pipe

The pipeline model describes the hydraulic properties for each section. Here the physical model mainly includes pipe friction loss, fluid inertial and fluid compressibility. The bond graph model is shown in figure 5.15. The control volume here represents the compressibility of fluid, which modeled by *C* element. The friction between

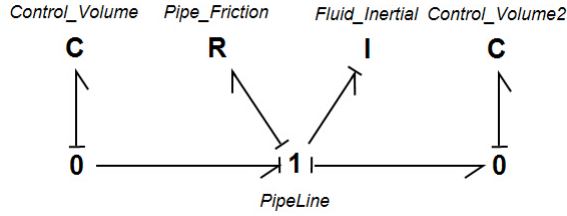


Figure 5.15: Bond graph model of pipe

pipeline and fluid is modeled by R element, the inertial of fluid in this section of pipeline is modeled by I element here. Please notice that the control volume, or C element here are mandatory since it can avoid the causality conflicts in the system. The constitutive law for the R element is

$$F_f = \frac{\rho l \nu^2 k}{2d} \quad (5.11)$$

Here the ρ is the density of fluid, l is the pipeline segment length, d is the diameter of pipe cross section. The k is the friction factor, which is decided by the type of flow through Darcy-Weisbach equation. The k can be calculated as

$$\begin{aligned} k &= \frac{64}{Re} && \text{For laminar flow} \\ k &= \frac{0.316}{Re^{0.25}} && \text{For turbulence flow} \end{aligned} \quad (5.12)$$

For the inertial element, the constitutive law can be concluded as Chu and Øyvind (2013).

$$\int P dt = \frac{\rho l \nu}{A} \quad (5.13)$$

Where the P is the pressure in the pipeline, ν is the velocity of fluid flow. The constitutive law in the control volume element can be concluded as

$$P = \frac{\beta \int \nu dt}{V} \quad (5.14)$$

Here the β means bulk modulus and V means the volume of fluid in the pipeline. In this thesis, the system is hydraulic power system, not the transfer system, thus the length of pipeline is not long, the mass of fluid inside the pipeline is smaller as well. Therefore, the friction loss and inertial of fluid only have minor effect on the dynamic of the system here. This conclusion had been proved in master thesis Skjong (2014). In the purpose of simplifying the system to achieve a fast and efficient simulation, the pipe model is not included in the hydraulic system developed in this thesis.

5.1.9 Position Control System

The position control here is based on PID control. The signal about the piston position or angular displacement will be measured by sensors. The reference signal can be generated from joystick, or from a signal source generator in this model. These two signals will be modified by the first order transfer function, which can smooth the signal without sudden change that may cause system oscillating or unstable. More importantly, it helps to protect the actuators like valves. The transfer function is

$$y = \frac{k}{s\tau + 1} \quad (5.15)$$

The k will decide the gain of the signal and the τ is the time constant. The difference is compared in figure 5.16 by choosing $k = 1$ and τ equal to 1 on the left and 0.1 on the right respectively. It can be easily found that with smaller time constant, the signal or system will have quicker response.

The error between measured position signal after transfer function and the reference

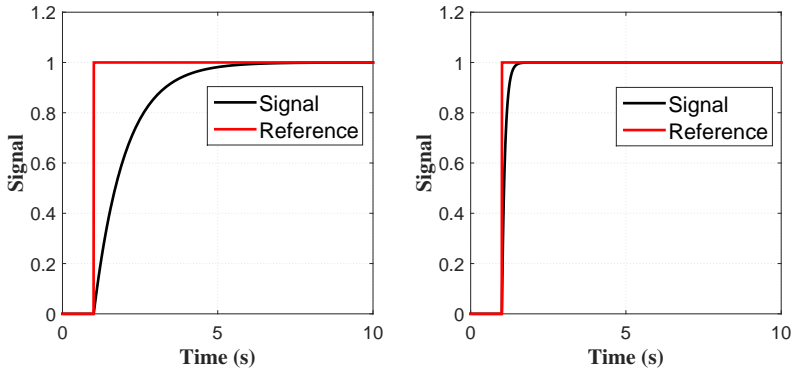


Figure 5.16: Compare of transfer function

signal after transfer function is calculated as shown in figure 5.17. The error dynamic will be send to the PID or P controller here. The output control signal generated from the controller will be sent to the 4/3 directional valve, which controls the motion of valve opening area. The simulation results will be shown in the later section. The controller developed here is proved to be able to control the motion of marine crane in this thesis. The control laws for the standard PID controller and P controller are

$$y(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \quad PID \quad (5.16)$$

$$y(t) = K_p e(t) \quad P$$

The transfer function linked with reference signal can also limit the maximum speed of the piston. In equation 5.15, when the gain $k = 1$, the function can be modified as what is shown in Skiong and Pedersen (2014).

$$y = \frac{V_{max}}{s + V_{max}} \quad (5.17)$$

The maximum speed is then decided by the time constant τ with $\frac{1}{\tau}$. This will be simulated and proved in later section. However, when the limitation of piston speed is low, the dynamic response or time delay of the system will be large, which may cause problem in practice.

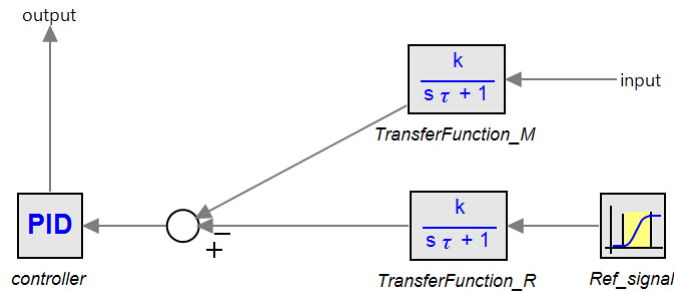


Figure 5.17: Position control system design

5.1.10 Friction Model

Most of the friction model is simplified to linear or constant in this thesis as they will not influence the dynamic of the system much. The details of friction model is still summarized here.

The friction is non-linear and almost can be found everywhere in the mechanical or hydraulic system. For instance, the friction force on the motor bearing, in the valve, between the inner wall of pipe and fluid etc. The friction sometimes is difficult to be calculated by mathematical analysis result, as it relies more on the experimental test and statistics analysis. The friction model mainly includes static friction and dynamic friction.

Static Friction Model

Coulomb Friction The coulomb friction model is the ideal model. The friction force is only determined by the force that act on the friction surface F_N , the friction coefficient μ and the sign of velocity. The relation described in Armstrong-Hélouvry

(1991) is

$$F_c = \mu F_N \text{sign}(\nu) \quad (5.18)$$

Viscous Friction When the piston or other object changing their speed fast, the lubricant or fluid between the friction surface will also generate friction force. The shear force will increase when the relative speed between two objects is increasing. This will result the friction as from Armstrong-Hélouvry (1991)

$$F_\nu = k_\nu |\nu|^{\delta_\nu} \text{sign}(\nu) \quad (5.19)$$

Where k_ν is the viscous friction coefficient and δ_ν is dependent on the non-linear relations.

There are some other types of hybrid static friction model such as Stribeck friction model, which will be decided by if it is sticking or slipping condition.

Dynamic Friction Model

The well developed static friction model is good enough to describe most of the system in most applications. However, more advanced friction is developed in order to reach the high requirement about the precious of the model. In the LuGre model, the friction force is not only determined by the relative velocity between the contact surface, but also the rate of force change. The details of this model can be found in master thesis Ekevid (2007).

5.1.11 Geometry Relations between Crane Arm and Hydraulic Actuators

When the hydraulic actuators model is ready, the motion of them will be transfer to the motion of crane since dynamics of crane will be studied to perform different type of operations. The motion of swing motor that drive the crane base is easily transfer to the rotation of crane base with only adding the gearbox between them, thus the details will not be discussed here. In this section, the relationship between the hydraulic pistons rod motion transfer to crane boom and crane jib motion will be described.

In the crane boom, the three nodes that form the triangle shown in figure 5.18. One node is the crane joint that connect the crane boom and crane base, and the remain two nodes are the connect point between hydraulic cylinder and crane arm on each side of the piston. The length of the triangle are L_{B1} , L_{B2} and L_{BP} respectively. Angle B and angle ϕ_B are shown in the figure as well. The TF element represent the transformer modulus here, which will transfer the force that piston act on the crane arm to the torque about the joint of crane arm. The TF element will also

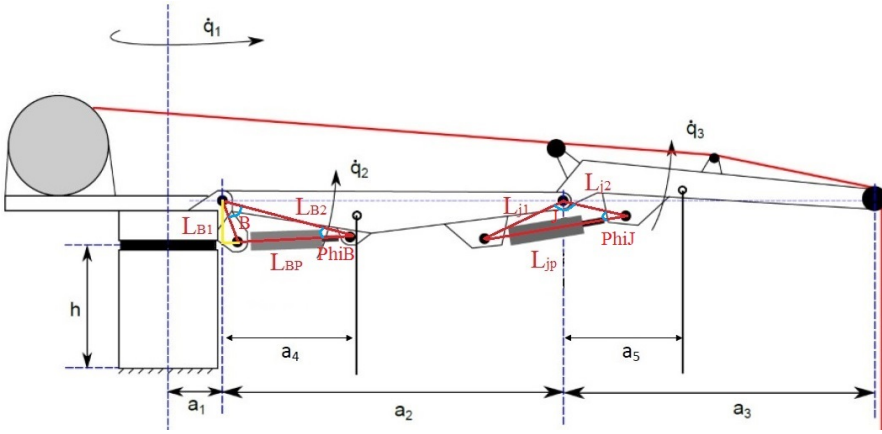


Figure 5.18: Geometry relations of crane

transfer the linear velocity of the piston to the angular velocity about the crane arm joint. The constitutive law can be concluded as

$$\begin{aligned}\tau_b &= R_2^b(q_2)F_p2 \\ \dot{q}_2 &= \frac{v_p2}{R_2^b(q_2)}\end{aligned}\quad (5.20)$$

The R_2^b , which is defined inside of the TF element has the relationship as

$$\begin{aligned}F_p &= \mathbf{r}\tau_p \\ \omega_p &= \mathbf{r}v_p\end{aligned}\quad (5.21)$$

Where F_p is the piston force, τ_p is the crane joint torque, v_p is the piston linear speed, ω_p is the angular velocity of crane joint and \mathbf{r} can be calculated as

$$\mathbf{r} = L\sin(\phi_b)\quad (5.22)$$

The L_{BP} represent the length of actuator rod can be calculated based on the cosine rule

$$L_{BP}(q_2) = \sqrt{L_{B1}^2 + L_{B2}^2 - 2L_{B1}L_{B2}\cos(B)}\quad (5.23)$$

The angle ϕ_b can be found as

$$\phi_b = \cos^{-1}\left(\frac{L_{B2}^2 + L_{BP}^2 - L_{B1}^2}{2L_{B2}L_{BP}(q_2)}\right)\quad (5.24)$$

Please notice that the angle is $B = B_0 + q_2$ here. B_0 is the initial angle here. The second linear hydraulic actuator (cylinder) has the same relationship here.

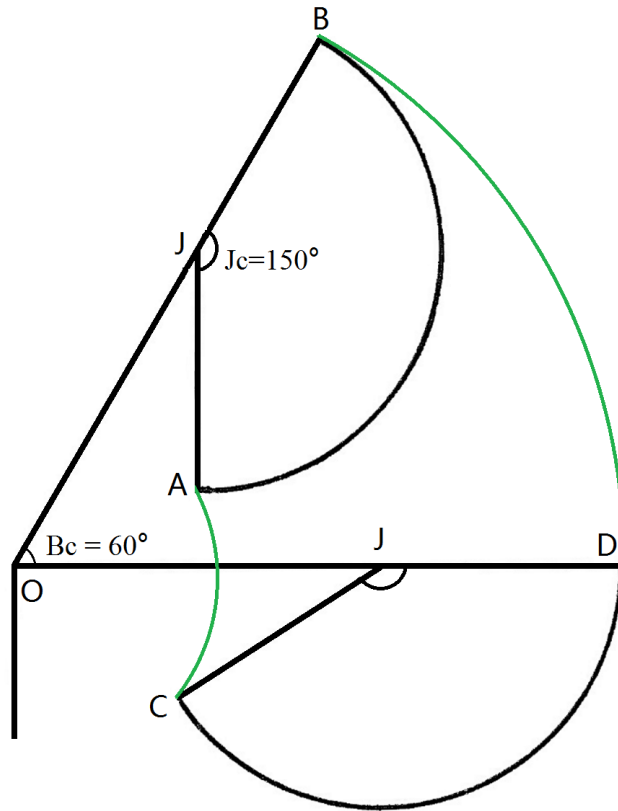


Figure 5.19: Working space of crane

5.1.12 Crane Working Space

After discussing the geometry relations between the actuators and crane bodies, the crane working space will be determined in this subsection. The working space is designed here. The maximum crane boom angle relative to the horizontal is set to 60° due to fact that if the angle Bc is close to 90° , therefore the force that required on the hydraulic actuator will increase dramatically with the constant load. This is because the actuator will almost becomes parallel to the crane boom that result in a very small moment arm which requires more force to balance the torque from the load. In addition, the system will become oscillating and unstable due to the lifting capability limitation. This will damage the system and cause serious safety problem when the crane is lifting heavy load. The angle Jc is designed with maximum 150° , this is mainly restricted by the second linear actuator (cylinder). When the crane jib is fully folded, there need space to place the actuator system between crane boom

and crane jib. When the crane jib is fully stretched, the angle Jc will also restrict by the reachable area of piston rod.

The working area then is represent by area $A B C D$ (crane tip position) marked with green curves. This working area actually is defined as the reachable area of crane tip. These limitations will be implement into the bumper elements that mentioned in the previous section.

5.2 Completed Bond-Graph Model

The finished bond graph model is shown in figure 5.20. The overall model contains two linear actuator hydraulic system that connecting together and one rotation actuator hydraulic system independently. This design is due to the fact that the load, or required pressure is much higher than the pressure in the swing motor hydraulic system. If connecting the swing motor hydraulic system with linear actuator (cylinder) hydraulic system, the pressure might be too high for the swing motor because the compensator decide the working pressure on the maximum pressure difference through the actuators. Hence, making two hydraulic system, one for high pressure and one for relative lower pressure will solve this problem. Meanwhile, by doing this, energy can be saved as the pump work can be smaller in the swing motor hydraulic system.

There are two other methods to solve this problem, one is adding an additional compensator on the low pressure branch, that is, the swing motor hydraulic system. Another method is to make a better controller that have logic section that can adjust the pressure based on both hydraulic cylinder pressure and swing motor pressure.

The load on the right hand side is modeled based on real operation, the MSe element represent the overall torque act on the crane joint includes the torque from load and crane self-weight. The swing motor has no specific load, but rotation inertial about the vertical axis and the bearing friction during the rotation.

5.2.1 Interface with OSOM

The complete hydraulic system can be merged with the OSOM platform by connecting the output of actuator force or torque to the 1 junction that connected with the IC field. Each 1 junction represents a joint of the marine crane. The IC field has already explained in chapter 2. In the IC field, the mass, inertial and dynamic properties of crane are included.

Please notice that the force or torque (after transfer) act on the joint 3, or the crane jib is not adding on the 1 junction for represent joint 3 directly, but together

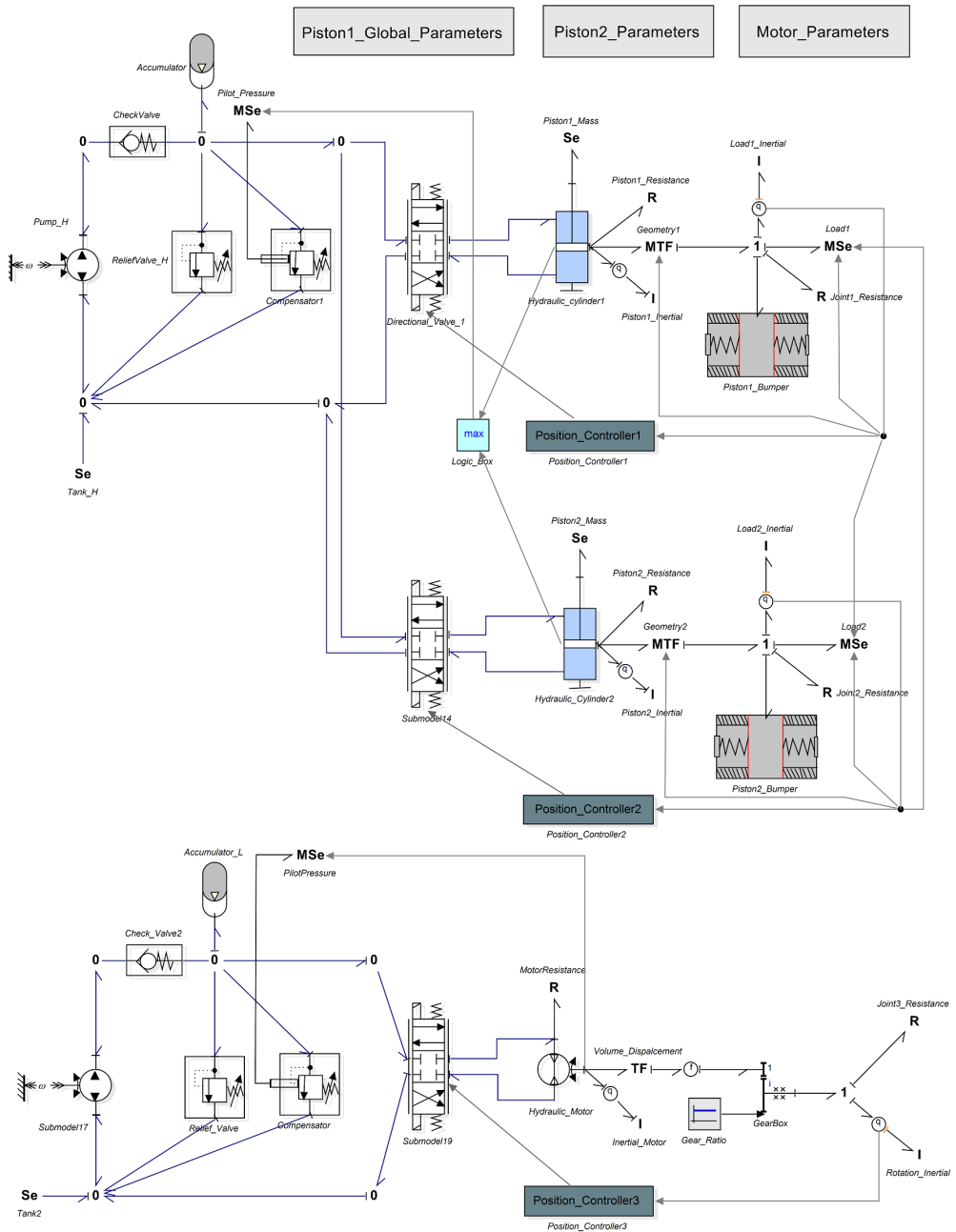


Figure 5.20: Completed bond graph model of marine crane hydraulic system

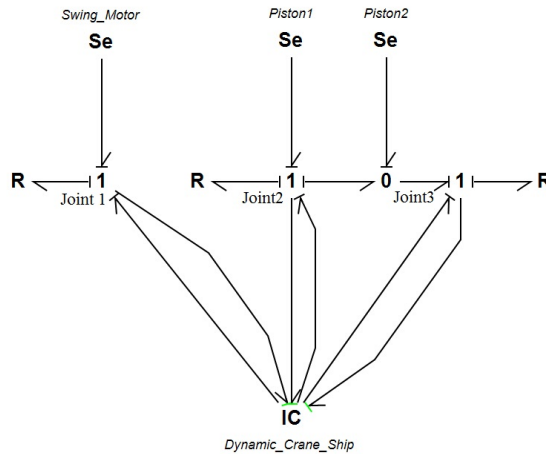


Figure 5.21: Actuators connect with IC field

with the torque generated from piston 1 that act on the crane boom. This is because joint 2 and joint 3 of crane revolve about the parallel axis, the force or torque that act on the crane boom will also influence on the crane jib.

The complete hydraulic system model has not connected to the OSOM at this moment. This is because the numerical solver used for them are different, which the hydraulic system is more sensitive that cannot apply the Euler method which proved to be the fastest solver in OSOM. The solver used for the crane hydraulic system is Vode Adams method. This problem can be solved by several methods. One of them is called co-simulation, which will give a brief introduction in the next section. However, the method that connect the crane hydraulic system to the OSOM is exactly the same as connected to the *IC* field, with no need for more modifications.

5.2.2 Co-Simulation

The reason for difficulty to change solver is because in the hydraulic system, the time step is variable, but in the OSOM, the time step is fixed. Due to the compressibility of fluid that varies with pressure, the *C* element in the bond graph model of hydraulic system will decide the time step most of the time. If we force the hydraulic system to fix step, for lager time step, it may cause numerical errors with some values cannot converge. However, if very small time step is chosen, the simulation will run extreme slowly that is far away from real-time simulation.

Co-simulation is a method that can run different simulation for different sub-system

under one master level. The master will coordinat different simulation that might developed in different tools with different methods and transfer the data among them. The master also response for make sure the simulation time is same in all the sub-simulations. The advantages of the co-simulation include:

- In hybrid system, there are many differential, algebraic, time continuous or discrete problems. Complex system cannot run simulation in one solver. Thus co-simulation can apply different tools for different sub-systems.
- Based on the properties of sub-system, the co-simulation give them space to choose the best simulation tool or solver that both increase the quality and speed of simulation.
- Can communicate data among different simulators during the simulation. For the co-simulation usually has three phases, the initialization phase, simulation phase and closing phase. However co-simulation will also lead to stability and accuracy problems, which is mainly caused by the stability of the sub-model that under the same master.

The Functional Mock-up Interface (FMI) for co-simulation is an interface standard. The FMI will provide solution of time dependent coupled system. The FMI has separation of: (1) description of interface data(XML file). (2) functionally (C-code or binary). According to the description on DaimlerAG (2015), a FMU is the file that contains the XML description file and the source implementation. The concept structure is shown in figure 5.22.

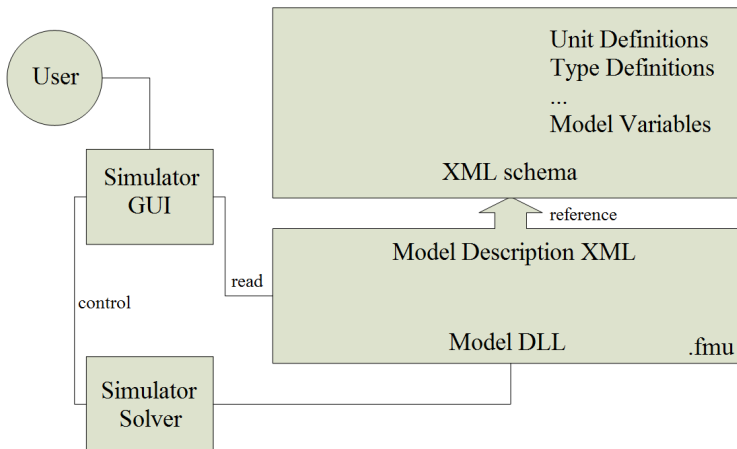


Figure 5.22: Structure of FMI

Please notice that the DLL file is only for the windows OS, not for Mac OS or Unix OS.

Chapter 6

Hydraulic System Case Studies

The crane hydraulic model is already built up by bond graph method. The modelling approaches and the properties of each components have been demonstrated in the chapter 5. Now, the simulation results of the crane hydraulic system will be studied, as well as the performance of the system. For designing and testing the hydraulic system, the parameters and fluid properties coefficients must be defined. The key parameters and coefficients are list in the appendix E. Part of the important parameters and coefficients that are studied in different simulation will be provided in this chapter as well.

6.1 Hydraulic Pump

There are two methods to control the flow and pressure that build in the hydraulic power system, one is control the flow rate from the pump and another is relying on the control valve (bleed the flow to tank if pressure reach the design level). Comparing to the valve control, the pump control will save more energy due to the power only developed when the actuators need.

However, the performance of servo valve control will be better than the pump control. This is because only the valves have frequency response, which gives the system higher performance with tight close loop control. In pump control, the pump displacement is controlled in order to control the motion of actuators. In conclusion, the servo valve control has better performance and more advantages in critical control design, and the pump control has higher overall efficiency if the system does not

need advanced control system.

The load sensing and pressure compensate pump are designed based on this. The pump used in this thesis is the constant flow rate pump with inner leakage, but the performance of the pressure compensate pump in pressure build up process is still shown in figure 6.1. The constitutive law in the Sf element which represent the

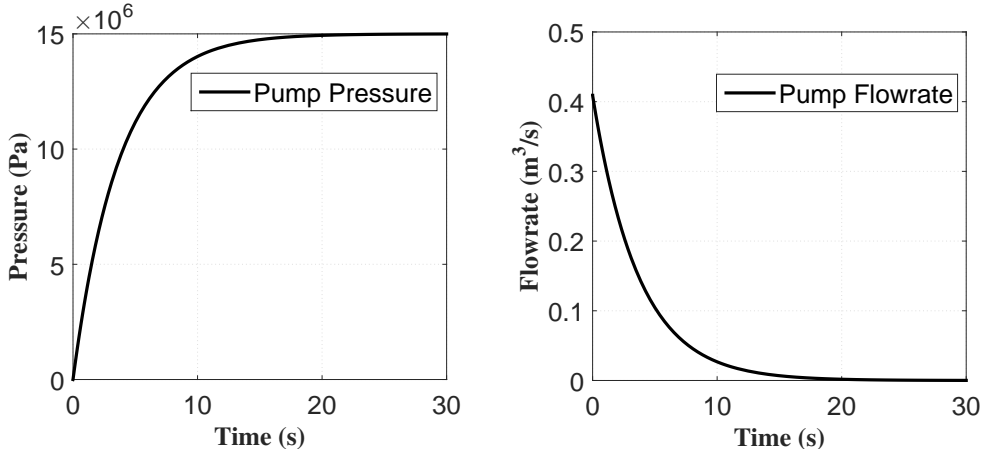


Figure 6.1: Variable flow pump

pump is

$$Q_m = \frac{V_m \omega}{2\pi} \quad (6.1)$$

$$Q = \frac{P_{set} - P}{\Delta P} Q_m$$

Where the V_m is the maximum pump volume displacement, P_{set} is the controlled working pressure, ω is the pump rotation speed, ΔP is the pressure deviation from controlled pressure and Q is the actual flow rate output from pump.

6.2 Relief Valve Studies

The relief valve will set the maximum pressure that possible in the hydraulic system, and protect the system from overpressure. The design of the relief valve is important. The relief valve working pressure should be exactly at the set point. The valve spring stiffness k and pre-compressed displacement x_{c0} will decide the closing force that act on the valve disc. The valve disc area or radius of disc r_{rel} and the set maximum system pressure will decide the opening force act on the valve disc. The maximum

pressure set here is 150 bar. Thus, the equation should be

$$P_{sys}\pi r_{rel}^2 = kx_{c0} \quad (6.2)$$

The x_{c0} here is set to a constant equal to 0.1 m, so only the stiffness of spring and the radius of relief valve disc need to be adjusted to fulfill the requirement. However, this equation will only calculate the opening pressure at the set point. The maintained system pressure can be higher than the set maximum pressure as the nozzle effect of the valve. More specifically, the flow rate that need to discharge through the valve is higher than the maximum design flow rate of the valve. Therefore, extra pressure is built up in the system even the relief valve is fully open. Since the nozzle effect in the valve opening area is highly non-linear, several cases will be studied here in order to demonstrate the theories and design the parameters.

In the case 6.2.1, the valve disc radius is chosen as 0.02 m, design opening pressure is 150 bar, the pre-compressed displacement of valve spring is 0.1 m. Based on the equation 6.2, the required spring stiffness is calculated as 188490 N/m. Then the opening pressure is exactly 150 bar. The simulation result for pressure building up process is shown in figure 6.2. The case 6.2.1 shows that even the opening pressure is

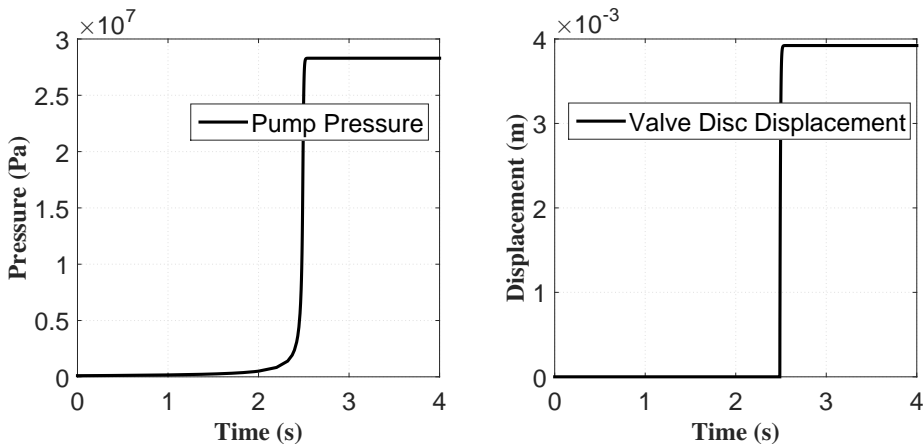


Figure 6.2: Simulation results of case 6.2.1

150 bar, the maintained pressure build up in the system can be very high due to the small size of valve disc, or nozzle. In order to solve this problem, the radius of valve disc now change to 0.08 m, the stiffness of the valve spring adjust to 3015000 N/m in order to open at 150 bar. The simulation results of case 6.2.2 is provided in figure 6.3. From the results of case 6.2.2 we can see that the system maintained pressure is only few bars higher than the set point, which is 150 bar. This proves that the nozzle effect in this design is in the reasonable range. Small modification is applied here, by

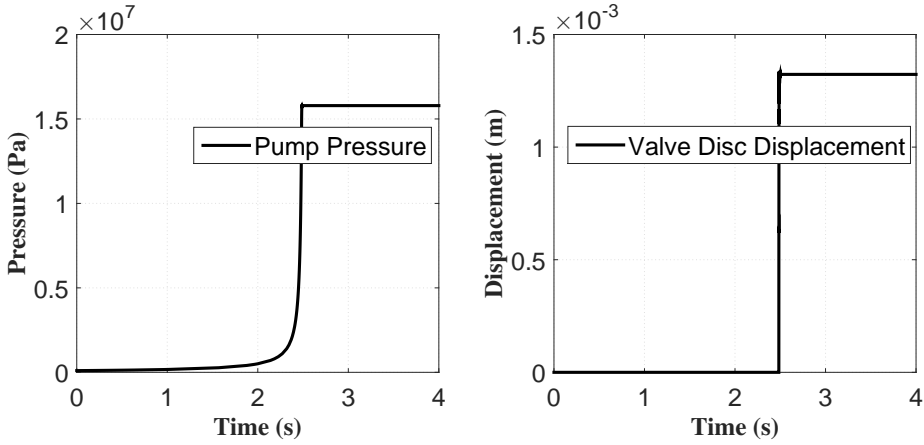


Figure 6.3: Simulation results of case 6.2.2

increasing the radius of valve disc (or nozzle) to 0.082 m , the opening pressure will be around 6 bar lower than 150 bar . However, this design can keep the system pressure under 150 bar even add the pressure build up by nozzle effect, which is 149.7 bar . The case studies results are collected in the table 6.1. Please notice that for the

Table 6.1: Parameters and results of relief valve studies

	Case 6.2.1	Case 6.2.2	Final Design
Pump flow (m^3/s)	0.1	0.1	0.1
Spring k (N/m)	188490	301500	301500
Valve disc radius (m)	0.02	0.08	0.082
Opening pressure (bar)	150.0	150.0	144.2
Maintained pressure (bar)	281.6	152.7	149.7

servo valves such as 4/3 directional valves and check valves, the nozzle should be well designed as the small nozzle can cause much pressure drop across the valve. This has negative influence on the overall efficiency and loading capacity of the hydraulic actuators.

6.3 Speed Limitation and Controller Gain Test

In order to focus on the specific part of the system, the bond graph model for signal cylinder (linear actuator) is separated and modified for some case studies instead of the whole crane hydraulic system developed in chapter 5. The only differences

here is the actuator set to lift the load in vertical direction without any geometry relations with the crane arm. Thus, the load hold capacity will increase as there is no long moment arm cause high joint torque but only linear force act directly in vertical direction. The key parameters of the isolated and modified single hydraulic cylinder is provided in table 6.2. In this single piston model, the linear actuator, or hydraulic cylinder will lift the load by piston rod to its maximum stroke and then lower the load to its original point.

Table 6.2: Parameters of single cylinder actuator

Parameters		Parameters	
Name	Value	Name	Value
Pump flow rate	0.03 m^3/s	Pressure limit	150 bar
Piston diameter	0.4 m	Accumulator volume	0.2 m^3
Maximum piston stroke	2.5 m	Load	30000 kg
Piston mass	50 kg	Constant bulk modulus	1.6 GPa

Velocity Limitation

As mentioned in chapter 5, the transfer function can also limit the maximum piston speed when operating. Recall the expression

$$y = \frac{k}{s\tau + 1} \quad (6.3)$$

$$y = \frac{V_{max}}{s + V_{max}}$$

The maximum velocity of the piston has to be constrained for the safety consideration, or under different operation. This is because for some heavy lifting operations, the speed should under certain value for safety, or for some precise installation operations, the low speed of actuator will help fine adjustment of the location of load.

First, the piston that without any velocity limitation is shown in figure 6.4. The maximum speed during the lifting phase is 0.585 m/s and the maximum speed in the lowering is 0.633 m/s . The curve of the piston velocity is smooth, thus no speed oscillating and speed suddenly change here that may damage the system, or bring unstable states to the control system. From the figure 6.4, the reference piston position and measured (actual) piston position are shown. The maximum delay between them is happened in the lifting phase, which around 0.5 s . The control gain can make the delay smaller, this will be explained in the next sub-section.

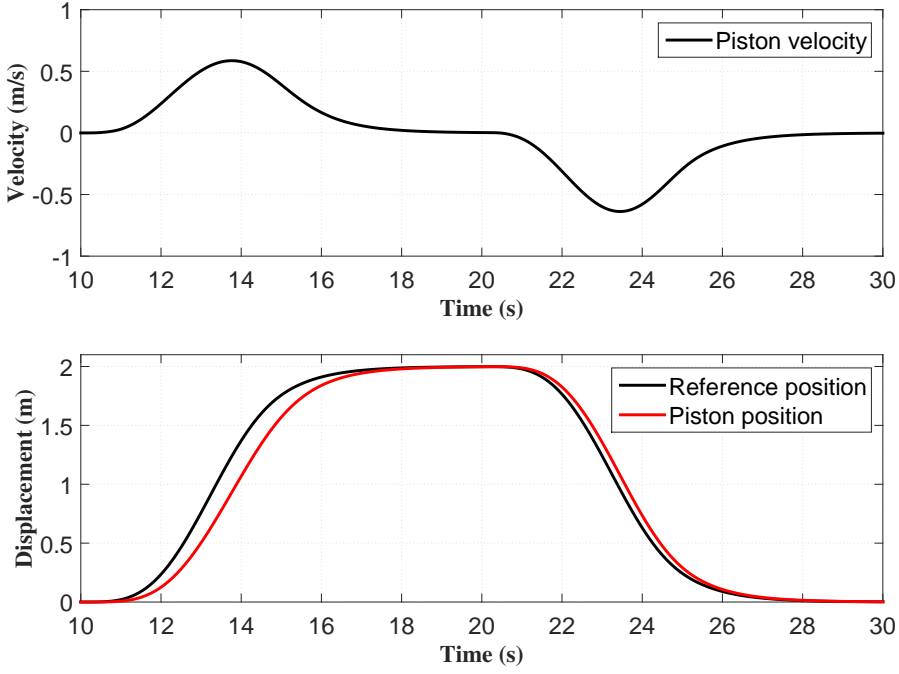


Figure 6.4: Piston velocity and position with speed limitation 1 m/s

Now, implementing the speed limitation for 0.5 m/s by changing the $\tau = 2$. The simulation results are provided in figure 6.5. The speed of the piston in both lifting and lowering the load now is under 0.5 m/s . With the maximum lifting speed 0.451 m/s and maximum lowering speed 0.476 m/s . The velocity curve is still smooth, but it takes more time to back to zero as the stroke is constant. The delay of the piston position in this system is almost the same as in the case with 1 m/s speed limitation. From this simulation result, the result of even more lower speed limitation can be predicted. The piston cannot reach its maximum stroke under slow speed as the lifting time is constant here. However, combining both of the simulation results, the speed limit function proved working well and will not influence the dynamics of the system much except the total piston displacement in a given time.

Controller Gain

The control gain in P controller can decide the piston position delay and other dynamics of the system such as pressure oscillating in the system. The error dynamic is decided by the reference position signal and the measured piston position. The

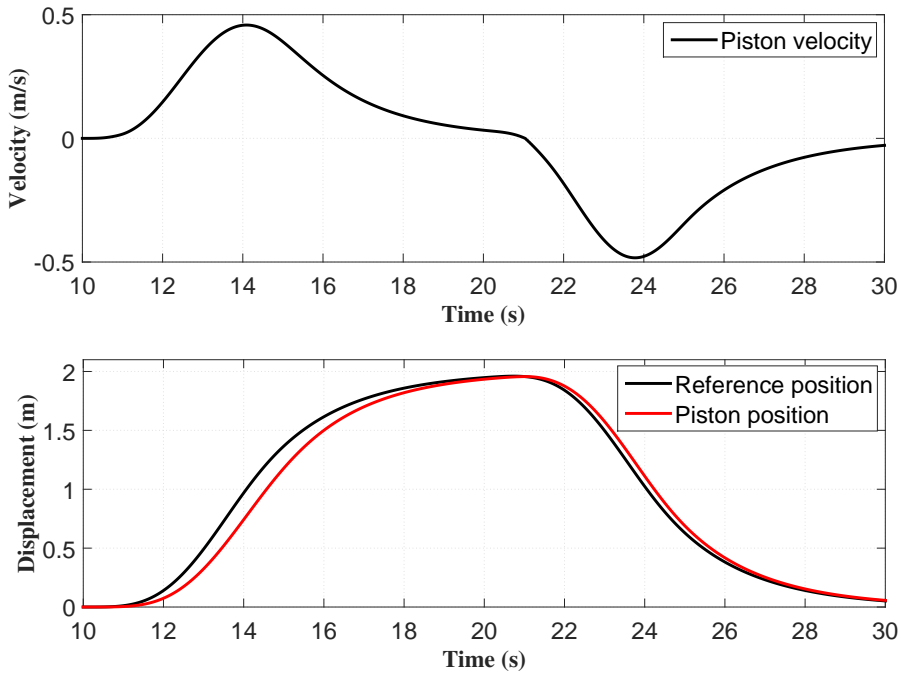


Figure 6.5: Piston velocity and position with speed limitation 0.5 m/s

control signal after the controller will be sent to the 4/3 directional valve to control the motion of valve opening area by controlling the solenoid in valve. The control gain with 0.001, 0.005, 0.01 and 0.1 will be studied and compared here.

The simulation results are shown in figure 6.6, figure 6.7, figure 6.8 and figure 6.9. In the case with control gain equal to 0.001, piston position cannot follow the reference position well, and the maximum delay in this case is approximately 11 seconds. The piston is even not able to reach the maximum stroke before it lowering down to the original point. In case with control gain 0.005, the piston position is able to follow the reference signal, however, the delay is not optimal during the lifting phase. In case with control gain equal to 0.01, the situation is quite satisfied, and the maximum delay is about 0.3 second. If continuing increasing the gain factor, the system delay can be neglected, but this causes another problem which is the pressure oscillating in the cylinder chamber. The oscillating of pressure influence the piston velocity directly. The velocity also has high frequency oscillating (from 13.2 s to 14.3 s in figure 6.9) and may cause safety problems and damage the system components. The pressure oscillating from 10 s to 10.4 s in figure 6.6 is caused by

insufficient pressure when system trying to lift the load.

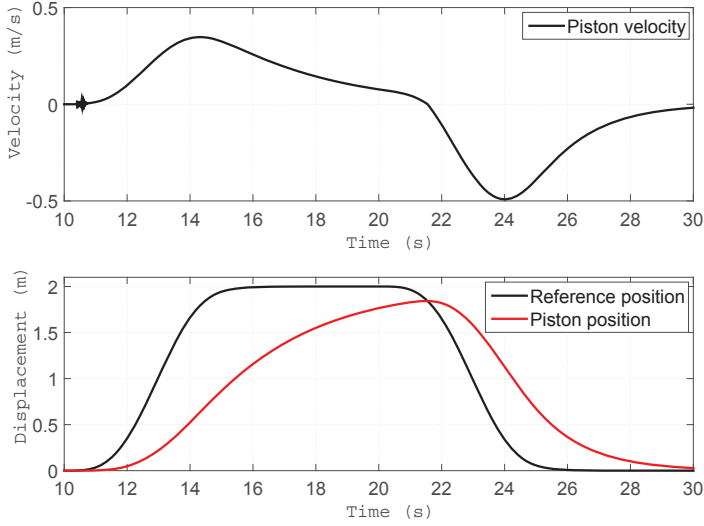


Figure 6.6: Simulation results with gain 0.001

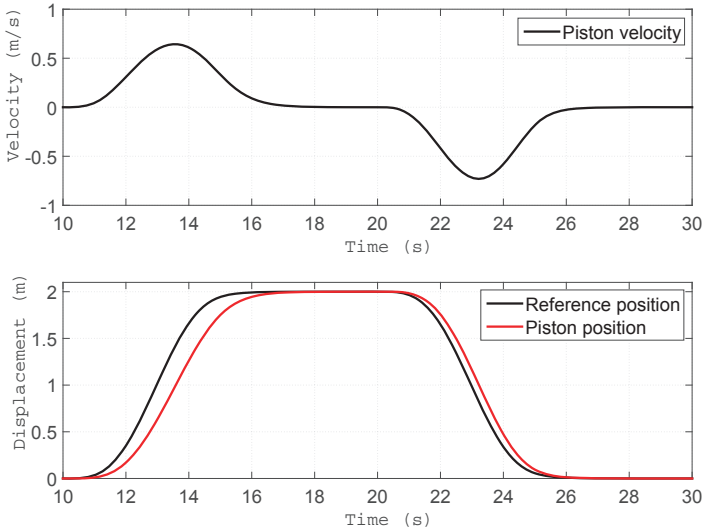


Figure 6.7: Simulation results with gain 0.005

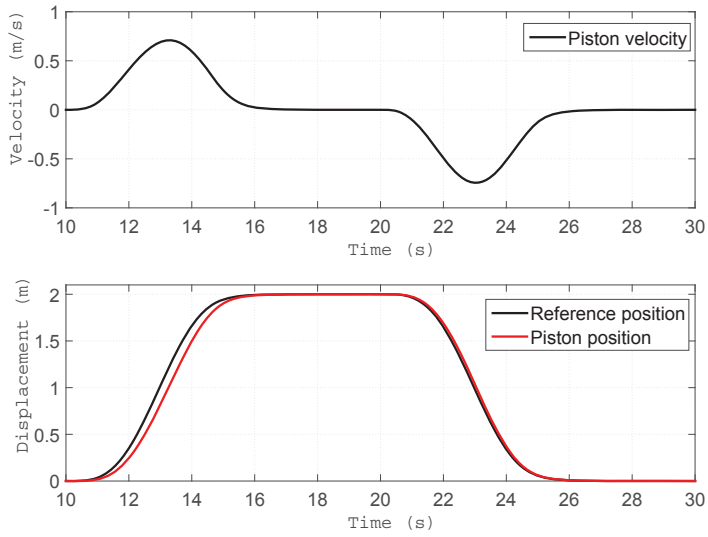


Figure 6.8: Simulation results with gain 0.01

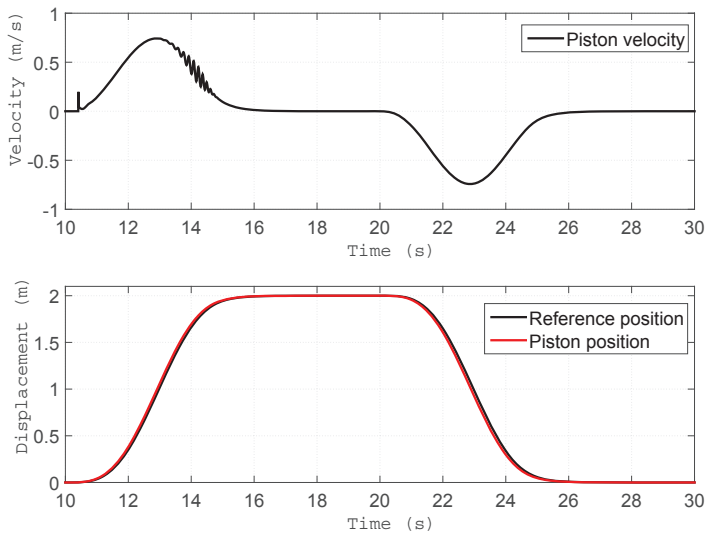


Figure 6.9: Simulation results with gain 0.1

6.4 Variable Bulk Modulus Studies

Different from the constant bulk modulus that set in the previous case, the variable bulk modulus here is pressure dependent. All fluids have some degree of compressibility. In some cases such as the system not need the tight control of response or fluids transfer system under low pressure. However, in the high pressure hydraulic power system with high flow rate of fluid, the energy can be stored in the compressed fluid.

The compressibility of fluid under high pressure can delay the system response. The hydraulic actuators will only start moving after the fluid in the system power line is fully compressed. And on the opposite way, the actuator may continue working even the servo valves already closed as the energy stored in the compressed fluid may expand the volume of the fluid. This is the reason for design less volume in the system between the servo valves and the actuator. In order to describe this properties of fluid, the bulk modulus is introduced.

Actually, the bulk modulus is a measure of the resistance to compressibility of fluids. The bulk modulus is affected by the fluid temperature and the air trapped in fluid. For the temperature, increasing it will make the fluid expand to larger volume that may create additional pressure in the system. Apart from the environment temperature change, compress of working fluid may also change the fluid temperature. Considering the tightly control and fast response of the system is needed here, the thermal process is considered as rapidly action. Thus the isentropic process is considered here. For the air trapped in the fluid, it has a significant effect on the bulk modulus as the air has much higher compressibility than the fluids. The air in the fluid is even more important in the high pressure hydraulic system as the solubility of air in the fluid will increase with the pressure increase. When pressure decreases, the bubbles may occur in the hydraulic fluid.

From the McCloy and Martin (1980), the variable bulk modulus is

$$\frac{1}{\beta_s} = \frac{1}{\beta_p} + \frac{1}{\beta_f} + \frac{1}{1.4p} \quad (6.4)$$

Where the β_p is the pipeline bulk modulus, β_f is the constant fluid bulk modulus, and the $n = \frac{V_g}{V_f}$ here represents the volume ratio of air and fluid. The pipeline bulk modulus is set to 200 *GPa*, the constant fluid bulk modulus is chosen as 1.6 *GPa*. Based on this equation, the relation between bulk modulus and pressure from 1 *bar* to 170 *bar* can be found in figure 6.10 under different volume ration of air and fluid. From the figures we can find that with higher volume ratio of air and fluid, the bulk modulus (*y*-axis) will be lower, especially in the low system pressure area (*x*-axis). The $n = 0.0003$ is chosen in the hydraulic system developed in this master thesis.

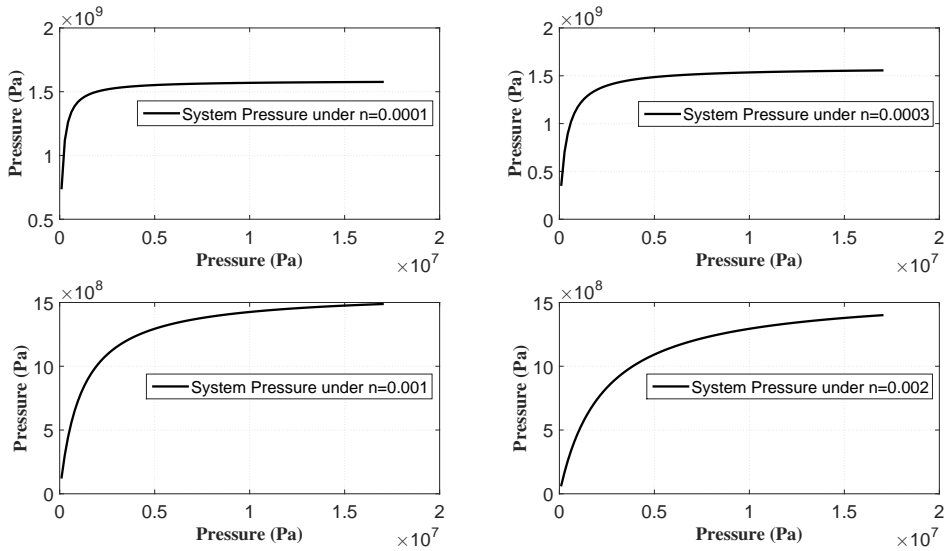


Figure 6.10: Relation between bulk modulus and pressure

The test and study of variable bulk modulus will be based on the single cylinder actuator system (high pressure) from section 6.3, except method to implement the bulk modulus. In order to compare how bulk modulus influence the system in both high pressure and low pressure, the single hydraulic swing motor model will be studied here as well.

Hydraulic Cylinder (Higher Pressure)

Variable bulk modulus and constant bulk modulus will be compared in the simulation. The variable bulk modulus is added on most of the C elements in the bond graph model. The volume in the compensator, and cylinder volume on each side of the piston have been modified to apply the variable bulk modulus in the single cylinder actuator hydraulic system.

Please notice that the system accumulator, or bladder type accumulator has not applied the variable bulk modulus as the constitutive law implement here is independent from the bulk modulus. In addition, when implementing the variable bulk modulus in the C elements in the bond graph model, there will be algebraic loop inside, which may cause numerical errors in some solvers. The algebraic loop will result in increasing simulation time as well. In order to handle this problem, a *while* logic loop is used in the C element to avoid the algebraic loop. The main idea of this

while logic loop is to calculate the system pressure by using constant bulk modulus first, then use this pressure to calculate the pressure with variable bulk modulus. When getting the results, comparing the different between these two pressure. If the error is smaller than the setting number, then use the pressure here. If not, iterates again until it fulfills the requirement.

First, the single hydraulic cylinder with and without variable bulk modulus will be demonstrated. The piston will lift the load from 10 s to 15 s to its maximum stroke 2 m high, then hold the load for 5 s, finally lowering the load to its original position from 20 s to 25 s. The key parameter is shown in table 6.2. The constant bulk modulus here is 1.6GPa. From the results in figure 6.11 and figure 6.12 we can see

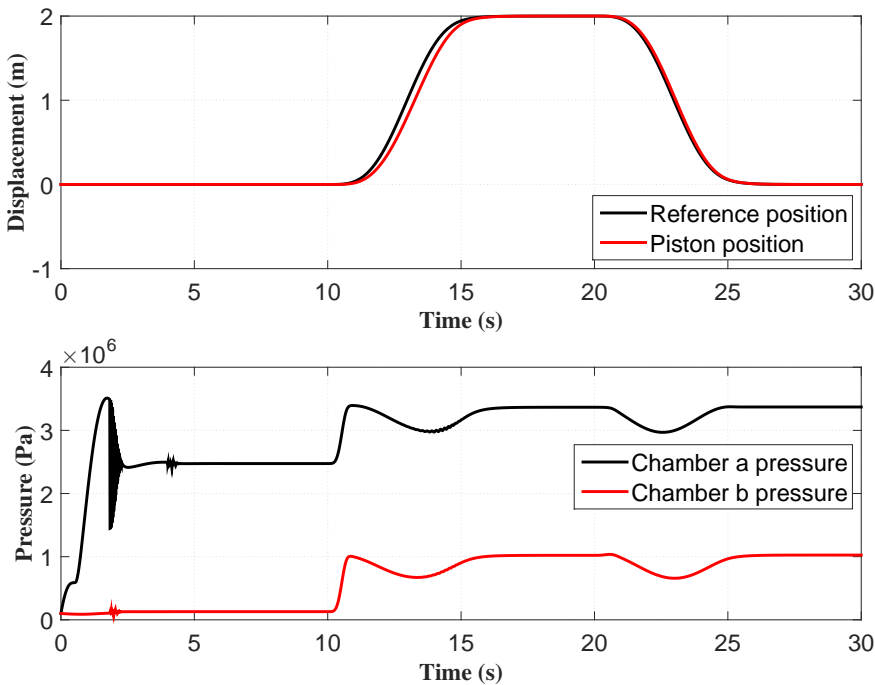


Figure 6.11: Hydraulic cylinder system with constant bulk modulus

the piston position followed the reference signal well for both of them. However, the mean and highest pressure in variable bulk modulus model is higher than the constant bulk modulus model. The general difference here is 3 bar. The varies pressure difference is around 8.21% based on chamber A pressure in constant bulk modulus model.

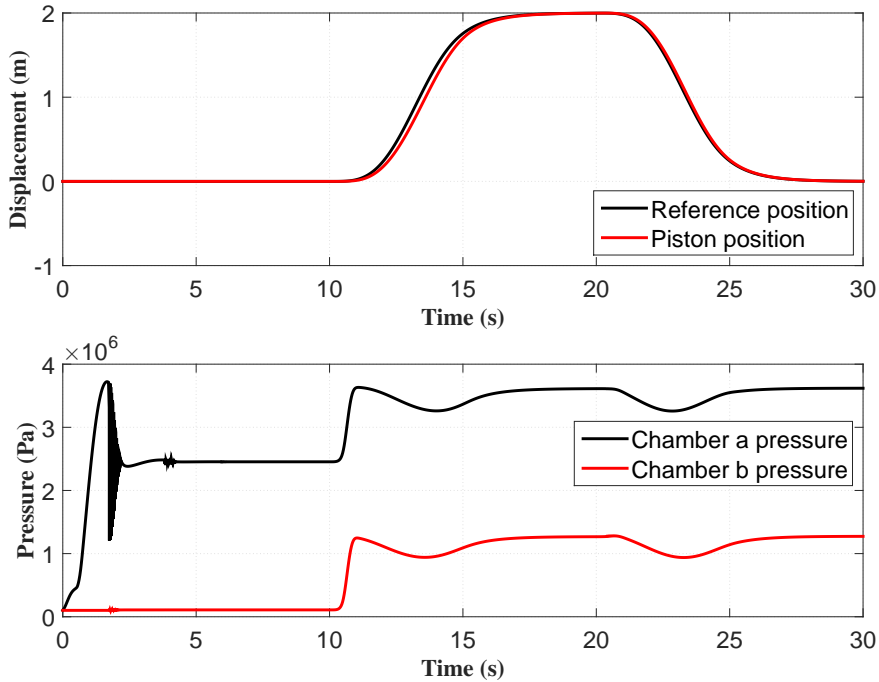


Figure 6.12: Hydraulic cylinder system with variable bulk modulus

This is because higher pressure should be applied on the variable bulk modulus case in order to get the same volume displacement as the constant bulk modulus case to drive the actuator.

Hydraulic Swing Motor (Lower Pressure)

The system under lower pressure such as the hydraulic swing motor system for the crane base will be studied next. The pressure in the control volume in each side of the hydraulic motor will be plot in figure 6.13 and figure 6.14. The motor will rotate in positive direction from 10 s to 15 s to 1 rad, then stop for 5 s, and finally back to original position from 20 s to 25 s. From the simulation results, the conclusion is same for the hydraulic cylinder case study. The system pressure is higher in the variable bulk modulus model. The difference of mean pressure between the constant bulk modulus and variable bulk modulus in the hydraulic swing motor system (lower pressure) is 2.4 bar. The varying pressure difference is around 17.91% based on chamber A pressure in constant bulk modulus model.

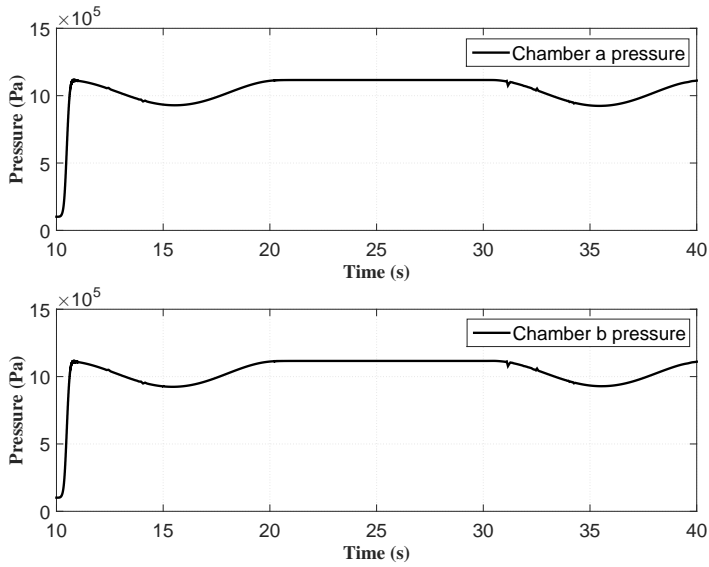


Figure 6.13: Swing motor system with constant bulk modulus

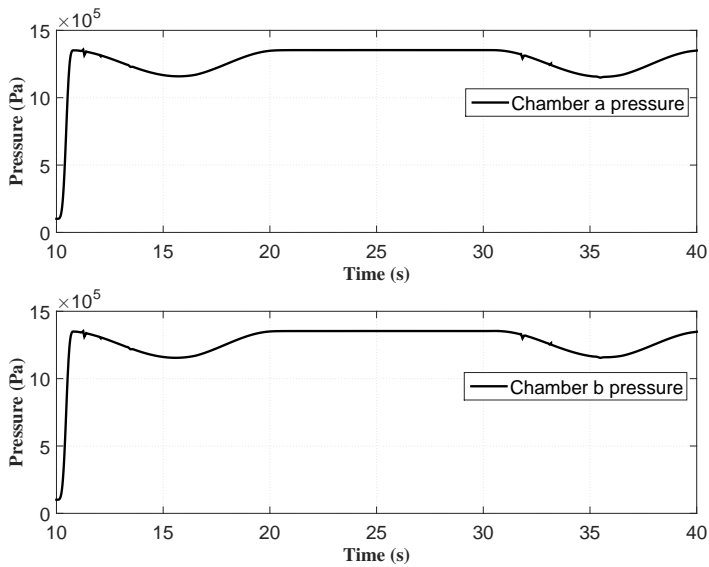


Figure 6.14: Swing motor system with variable bulk modulus

From the varying pressure difference in higher pressure hydraulic system and lower pressure hydraulic system, we can see that the bulk modulus has much more significant influence on the low pressure system. Actually, this conclusion has been proved in the figure 6.10. The bulk modulus is very low in the low pressure area and increase dramatically when pressure increasing. Finally, the bulk modulus will converge to a certain value which around 1.6 *GPa* for the fluid here. The 1.6 *GPa* is also the setting value of constant bulk modulus in this case.

Based on these simulation results and conclusions so far, we can also conclude that the constant bulk modulus, which set close to the stable bulk modulus after pressure increasing, will have much less influence on the high pressure system. This is because the difference between real bulk modulus and constant modulus (set to 1.6 GPa) is very small in the high pressure area.

This characteristic can be used to simplify the simulation, as the calculation of variable bulk modulus may bring algebraic loop inside the bond graph model and slow down the simulation speed. If the system is working under high pressure condition, the constant bulk modulus set around the high working pressure can be used, which will not bring much errors into the system. The decision of which bulk modulus is to be used based on the requirements of simulation speed and accuracy.

Table 6.3: Parameters of hydraulic swing motor system

Parameters		Parameters	
Name	Value	Name	Value
Pump flow rate	0.3 m^3/s	Pressure limit	150 <i>bar</i>
Displacement Volume	0.005 m^3	Accumulator volume	0.15 m^3
Constant bulk modulus	1.6 <i>GPa</i>	Motor leakage factor	5 10^{-9}

6.5 Payload Studies

The payload study will be based on the single hydraulic cylinder system described in the table 6.2. The variable bulk modulus is applied in this case study. Different payload will be tested and the simulation results for the system dynamic behavior will be plot in this section. The payload added on the system will be 10 metric tons, 20 metric tons, 30 metric tons and 35 metric tons respectively. The load position, load velocity, system accumulator pressure and two cylinder chambers pressure will be studied.

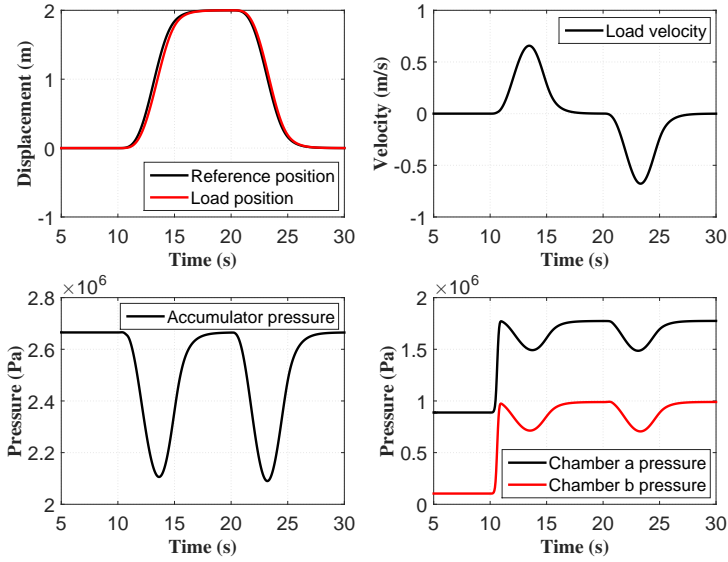


Figure 6.15: System dynamics with 10 metric tons load

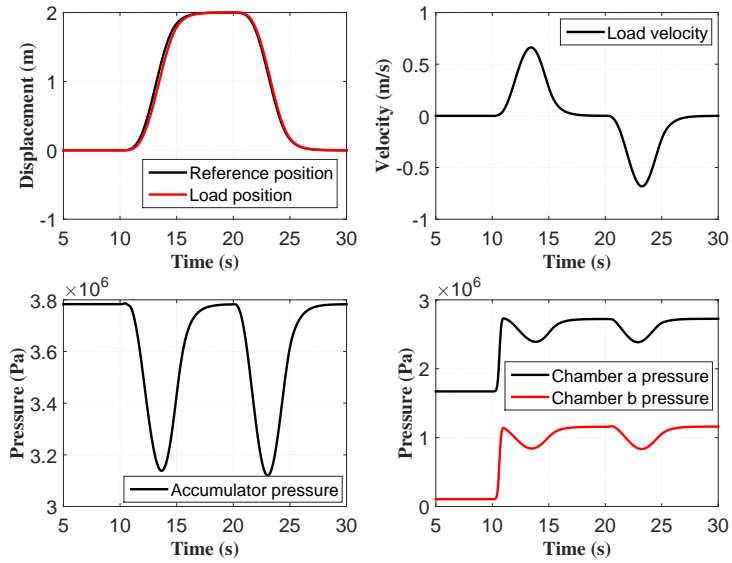


Figure 6.16: System dynamics with 20 metric tons load

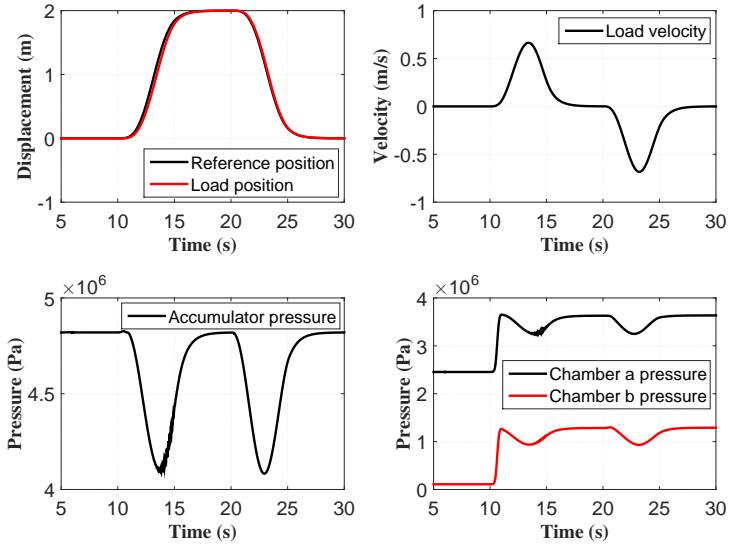


Figure 6.17: System dynamics with 30 metric tons load

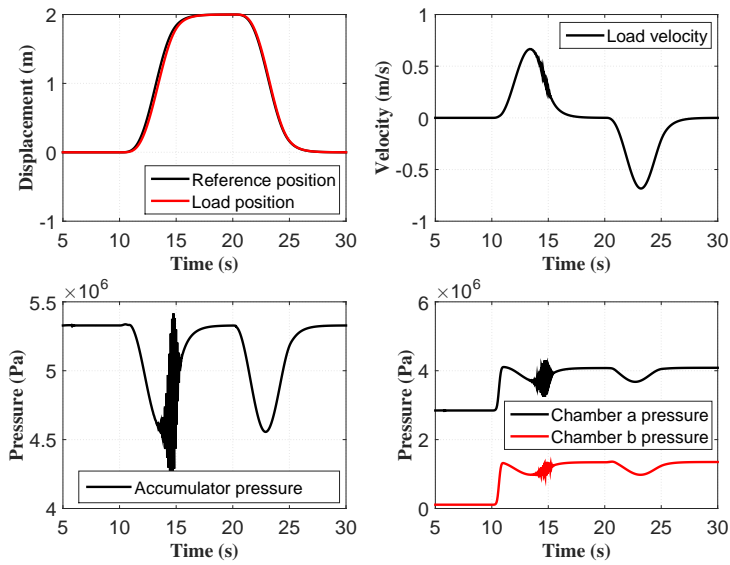


Figure 6.18: System dynamics with 35 metric tons load

In all of the cases here shown in figure 6.15, figure 6.16, figure 6.17 and figure 6.18. The load position followed the reference signal well with very small time delay which is mainly caused by the measurement delay and fluid compressibility. For the load velocities here, all of them have smooth velocity curve under certain speed limitation except the case with 35 *mt* payload that has small speed oscillating during the lifting phase. All of these cases can lift the load to its desired position within the given time under certain speed limitation.

It can be seen from the results that both the system accumulator pressure and hydraulic cylinder pressure increase with the load increasing. This is reasonable as more pressure should be act on the piston in order to get higher lifting force. However, when the load increased to 35 *mt*, the pressure in both system accumulator and hydraulic cylinder will oscillate during the lifting phase. This may be caused by many reasons, such as the natural frequency of the system, or affected by the stability of solver that applied here.

In order to adjust the overall stiffness of the system, changing the capacity of the system accumulator is a efficient and direct method. By double the volume of the system accumulator, which is changing the natural frequency of the system, the oscillating no longer exists under the 35 *mt* or even heavier load. The new simulation result of 35 *mt* case after double the size of system accumulator is shown in figure 6.19.

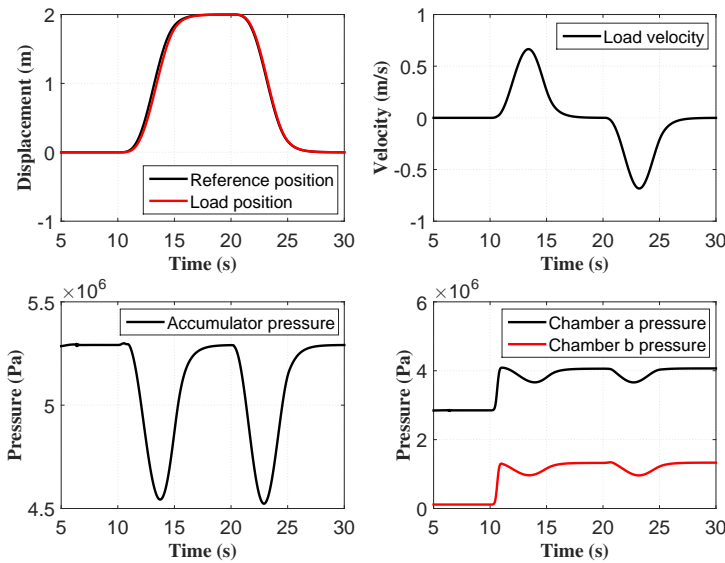


Figure 6.19: 35 metric tons load after increasing the accumulator volume

From the figure we can see that the oscillating disappeared and the system works perfectly. After find the solutions for the pressure oscillating problem, the system load capacity limitation can be tested. After testing and simulation, the maximum lifting load of this hydraulic cylinder system is around 170 *mt*. The theoretical load capacity should be

$$\begin{aligned}
 A_{pist}P_{max} &= m_{max}g \\
 m_{max} &= \frac{A_{pist}P_{max}}{g} \\
 &= \frac{0.2^2\pi 1.5 \times 10^7}{9.8} \\
 &= 192.3 \text{ metric tons}
 \end{aligned}
 \tag{6.5}$$

Where A_{pist} is the piston cross section area, P_{max} is the maximum pressure can archive in the system, which decided by the relief valve (safety valve). The maximum pressure is set to 150 *bar* in this master thesis. m_{max} is the maximum mass of load and g is the local gravity acceleration. The simulation results of maximum load case will be provided in figure 6.20. The pressure both in the system accumulator and

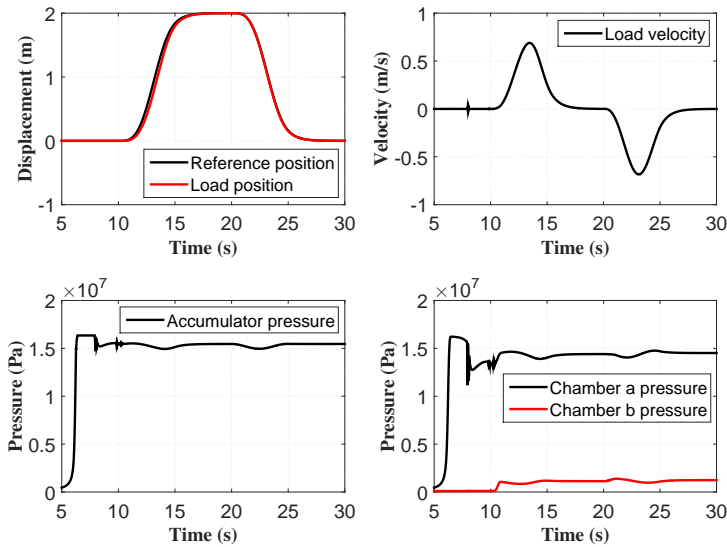


Figure 6.20: Extreme condition with 170 metric tons load

hydraulic cylinder can be found in the figure. It shows that the system pressure is almost reach 150 *bar* when stable, and a little bit exceed 150 *bar* when build up the

pressure in the system. There are many reasons for this maximum lifting load is smaller than the theoretical value calculated in equation 6.5. For instance, the piston actuator weight, pump inner leakage, actuator inner leakage, valve pressure drop, fluid resistance in the system and friction in the mechanical part such as friction between piston and cylinder. Thus, during design and operation, all these factors should be taken into consideration.

6.6 Crane Operation Simulation

In this section, the complete crane operation case will be demonstrated based on the complete crane hydraulic system model. The operation will be introduced and the dynamics of the hydraulic system will be shown by plots.

The operation is to lifting a 20 *mt* subsea structure from the starboard of the DP vessel and then place it in the deck area in stern of the vessel. The crane will start from the initial state in figure 6.21, the load is in the starboard of the vessel. Then the crane boom rotates to 45° about the horizontal line and lift the subsea structure out of sea water, which is shown in figure 6.22. After lifting the load, the whole crane will rotate 90° about the crane base to the deck area of the vessel in figure 6.23. Finally the crane jib will rotate 45° to lowering the load, this can be seen in figure 6.24 and figure 6.25. The key parameters of the operation is concluded in table 6.4.

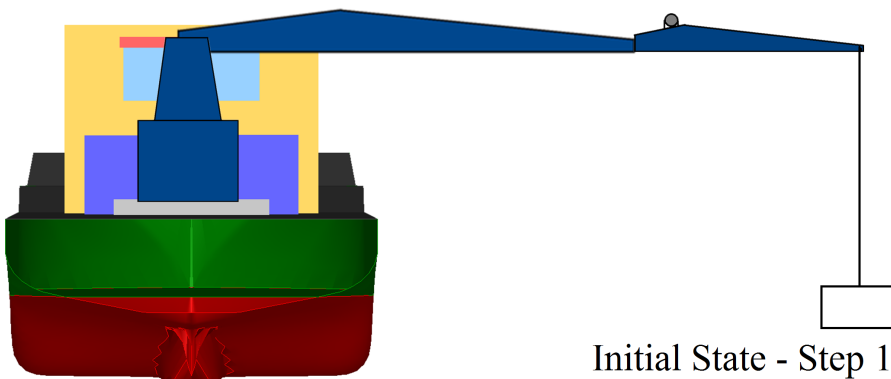


Figure 6.21: Crane operation step 1

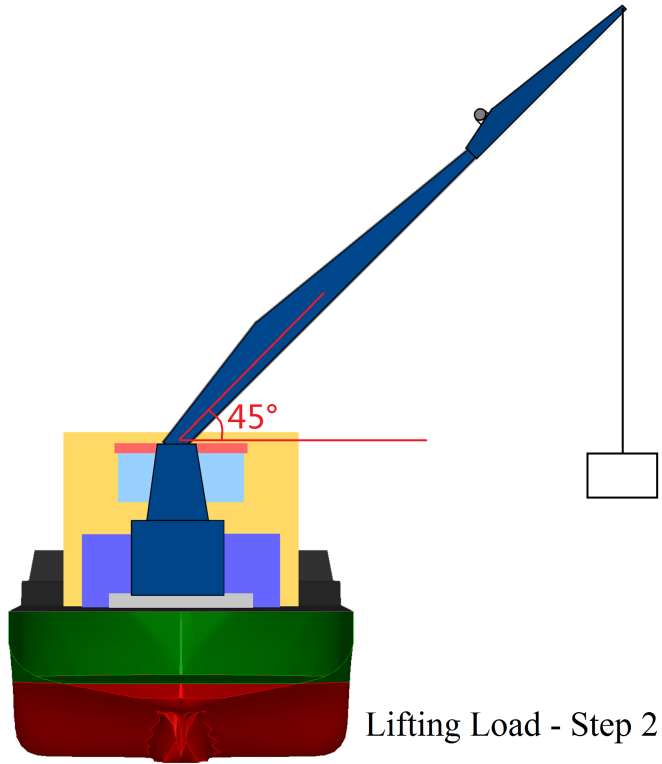


Figure 6.22: Crane operation step 2

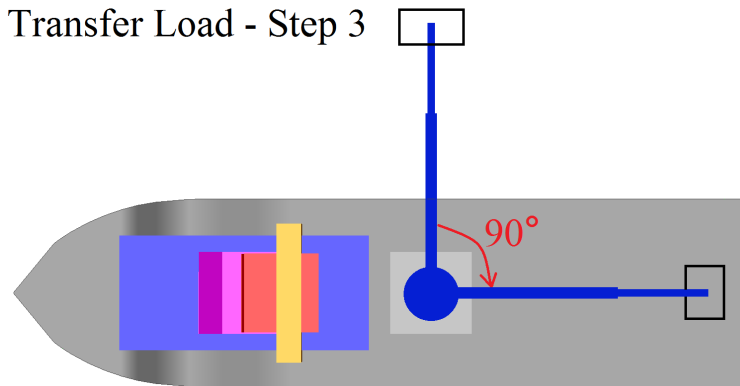


Figure 6.23: Crane operation step 3

Preparing Lowering Load - Step 4

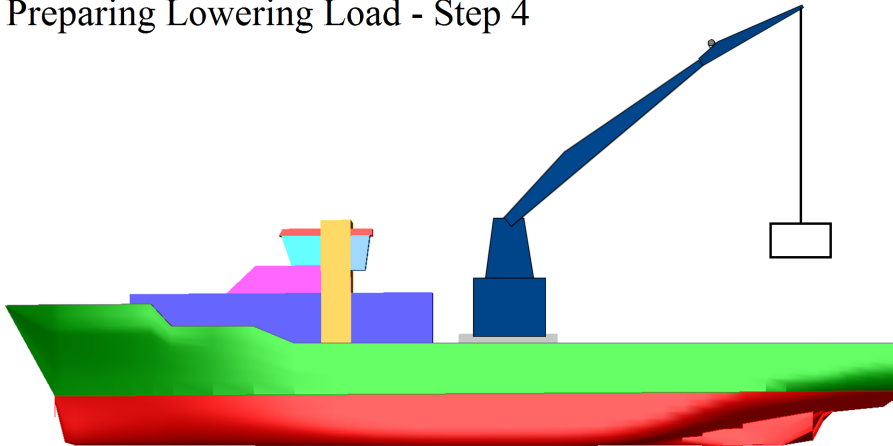


Figure 6.24: Crane operation step 4

Lowering Load - Step 5

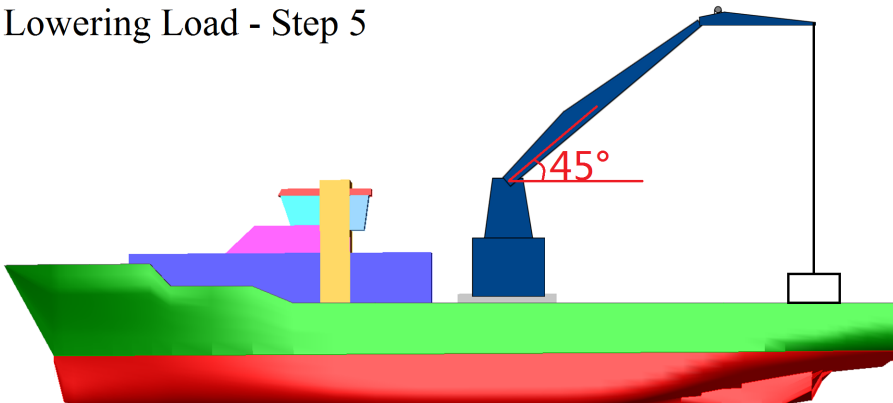


Figure 6.25: Crane operation step 5

Table 6.4: Key parameters in simulation of crane operation

Key Parameters	Value (Unit)
Pump flow rate for cylinders	0.04 (m^3/s)
Pump flow rate for motor	0.04 (m^3/s)
Cylinder piston diameter	0.4 (m)
System accumulator total volume	0.4 (m^3)
Load	20 (<i>metric tons</i>)
Mass of crane boom	2.5 (<i>metric tons</i>)
Mass of crane jib	1 (<i>metric tons</i>)
Mass of crane base	5.5 (<i>metric tons</i>)
Motor volume displacement	0.005 (m^3/rad)
Gear box ratio	40:1

The simulation results of this operation are shown in figure 6.27, figure 6.26, figure 6.28 and figure 6.29.

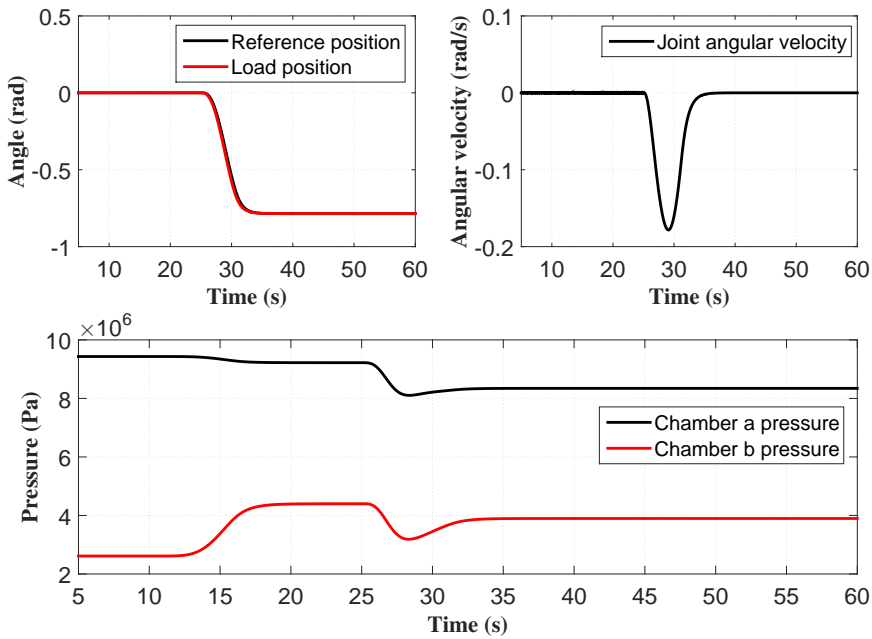


Figure 6.26: Dynamics of crane joint 3 (Crane jib)

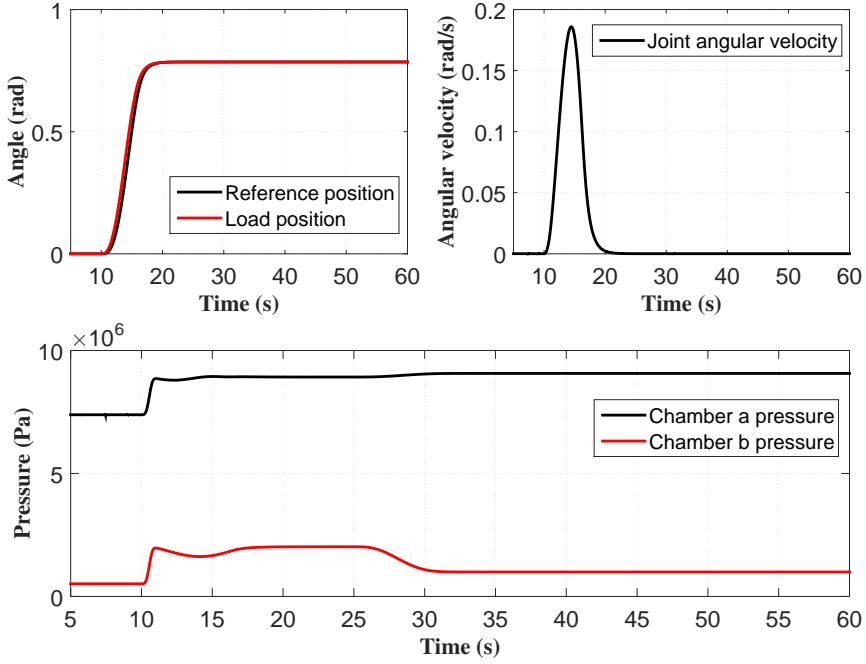


Figure 6.27: Dynamics of crane joint 2 (Crane boom)

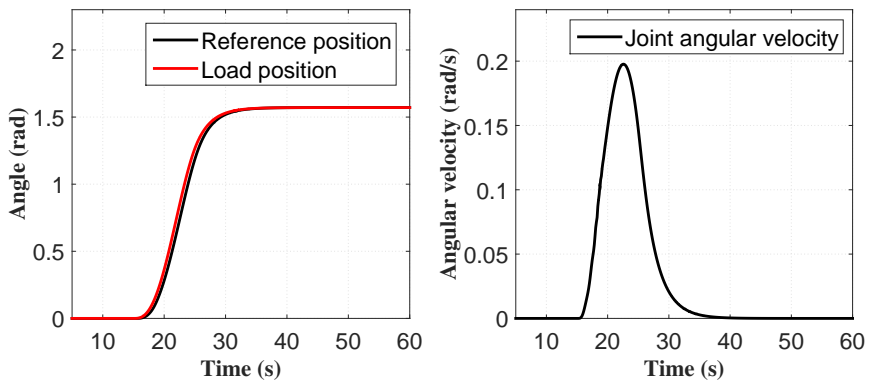


Figure 6.28: Dynamics of joint 1 (Crane base)

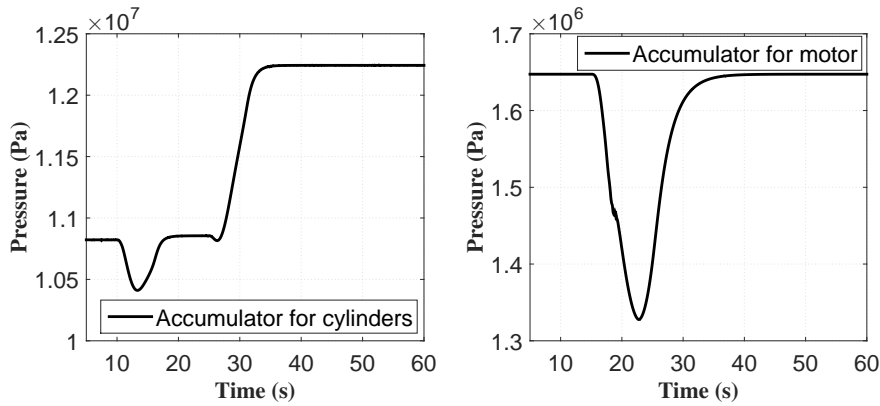


Figure 6.29: Pressure of accumulators

From the simulation results, we can find that the motion of all the crane bodies followed the reference signal well, which with very small and acceptable time delay. Both the system accumulator pressure and actuator pressure are under the safety value. The velocity of all the joints are within design limit under the speed limitation function in the controller, and the velocity curve is smooth as well.

These simulation results indicate that the crane hydraulic system developed here is able to handle the crane operation. The system is stable and has quick response to the reference signal with a certain amount of load capacity. Please notice that the ship is operated in the calm sea water, and the dynamics of the DP vessel here is assumed stable with minor motion that can be neglect.

The result studies here also prove that the bond graph modelling method is efficient and good enough to model such hydraulic system. In addition, the simplifications and assumptions here only bring very small errors into the system. It is reasonable to consider the errors as negligible in order to improve the speed of simulation, and relieve the modelling work loads.

6.6.1 System Sensitivity Study

These sensitivity analysis results are concluded during the model test process. Most of them are qualitative conclusions from the quantitative facts when tuning the system parameters. They can be useful when designing the crane hydraulic system and studying the properties of the system, in order to build a more efficient and robust system.

Table 6.5: Parameters sensitivity study

Key Parameters	Increase	Decrease
Pump flow rate	faster system response	slower system response
Bulk modulus	lower pressure required	higher pressure required
Control gain factor P	faster system response	slower system response
Frictions	slower system response less oscillating in system	faster system response more oscillating in system
Inner leakage	slower system response	faster system response
Spring stiffness in valves	minor effect	minor effect
Accumulator volume	less pressure oscillating	stronger pressure oscillating
Valve nozzle area	faster system response	slower system response
Flow coefficient	faster system response	slower system response
Fluid inertial in pipe	minor effect	minor effect
Pipeline frictions	minor effect	minor effect

Chapter 7

Conclusion and Further Work

7.1 Conclusion

This master thesis is based on the OSOM platform that developed by Rokseth and Skjong (2015), and focus on the efficient method for 3D visualization between simulator 20-sim and visualization tool GLview (Ceetron (2014)) about the Offshore Ship Operation Model (OSOM). The work also includes developing the model of crane hydraulic sub-system in OSOM by using bond graph theory (Pedersen and Engja (2010)). In addition, the dynamics of DP vessel and crane are studied based on the project thesis of the author and the OSOM. Background and basic theories are provided before the OSOM model studies.

For the 3D visualization of OSOM, two main topics are studied and developed. The first topic is data communication between the simulator and visualization tool. The DLL file is used in this thesis in order to achieve the goal. This DLL file is compiled from C programming, it has many functions. The first function is to do the heavy calculations for generating wave carpet, wave dynamics and vessel dynamics, because the simulator 20-sim (Kleijn and Groothuis (2014)) only defines and controls several key parameters of the wave. The wave theories are mainly from Faltinsen (1990) while the JONSWAP wave spectrum is used as the study background is in NCS. The second function is to build the communication between 20-sim and GLview. The DLL file receives data from 20-sim, executes calculations, sends back the results to simulator for simulation and prints the VTF file for 3D visualization. The second

topic in visualization is to find efficient method when building the VFT file. After studying the structure of VTF file, the rotation results method is used instead of using total displacement method, which scales down the size of the file much and saves the visual simulation time. The 3D visualization method developed for the OSOM in this thesis is proved to be efficient and good enough to demonstrate the dynamics of OSOM from visual aspect.

In the crane hydraulic sub-system, the model is built based on bond graph theory and implemented in 20-sim. The hydraulic system contains a pump system, a system accumulator, a tank, a check valve, safety valves, compensators, 4/3 directional valve hydraulic cylinders, hydraulic swing motor and position controllers. The hydraulic system is studied both on component and system level with several simplifications and assumptions are made in order to reduce the modelling work load and simulation time. These simplifications and assumptions are proved only has minor effect on the final results. The load capacity of single hydraulic cylinder developed here is around 175 *mt*. In the complete model with crane bodies, a 20 *mt* subsea structure is lifted from sea surface to the deck area of the vessel. It performs perfectly, which means it has stable system dynamics and fast system response.

7.2 Further Work

This section provides recommendations for further works for the people that interested in continuing improving and enhancing this model and corresponding studies.

- On-line 3D visualization of simulator 20-sim.
- Connect crane hydraulic system to OSOM by co-simulation methods from DaimlerAG (2015).
- Introduce joysticks to control the motion of DP vessel and crane mounted on it, together with on-line visualization.

Bibliography

- Armstrong-Hélouvry, B. (1991). *Control of Machines with Friction*. 1 edition.
- Ceetron (2006). *GLview Reference Guide*. 7.1 edition.
- Ceetron (2014). Glview inova. <https://ceetron.com/products/end-user-tools/glview-inova>.
- Chu, Yingguang, Vilmar, s. Z. H. and Øyvind, B. (2013). Modelling and simulation of an offshore hydraulic crane. 28th European Conference on Modelling and Simulation.
- DaimlerAG (2015). Functional mock-up interface. www.functional-mockup-interface.org.
- Ekevid, T. (2007). Modeling, simulation and control of a hydraulic crane. Master's thesis, Vaxjo University.
- Faltinsen, O. (1990). *Sea Loads on Ship and Offshore Structure*. 1 edition.
- Fossen, T. I. (2011). *Handbook of Marine Craft Hydrodynamics and Motion Control*. 1 edition.
- German, F. and Delarmelina, D. (2007). Dynamic of multibody system with bond graphs. *Mecanica Computacional*, XXXVI(2943-2958).
- Ginsberg, J. (1998). *Advanced Engineering Dynamics*. 2 edition.
- Goldstein, H. (2001). *Classical Mechanics*. 3 edition.
- Inc., D. S. (1989). *StereoLithography Interface Specification*.

- Kelly, R., Santibanez, V., and Loria, A. (2005). *Control of Robot Manipulators in Joint Space*. 1 edition.
- Kernighan, B. W. and Ritchie, D. M. (1988). *The C Programming Language*. 2 edition.
- Kleijn, I. C. and Groothuis, I. M. A. (2014). *20-sim 4.5 Reference Manual*. 2 edition.
- McCloy, D. and Martin, H. R. (1980). *Control of Fluid Power, Analysis and Design*. 2 edition.
- Meirovitch, L. (2003). *Methods of Analytical Dynamics*. 1 edition.
- Pedersen, E. and Engja, H. (2010). *Mathematical Modelling and Simulation of Physical Systems*. 3 edition.
- Pål, J. F., Jan, T. G., and Pettersen, K. Y. (2012). *Vehicle-Manipulator Systems*. 1 edition.
- Read, P. and Meyer, M. (2000). *Restoration of Motion Picture Film*. 1 edition.
- Rokseth, B. (2014). A bond graph approach for modelling systems of rigid bodies in spatial motion. Master's thesis, Norwegian University of Science and Technology.
- Rokseth, B. and Pedersen, E. (2014). Bond graph modelling of marine vehicle with manipulator equipment. *Mecanica Computacional*.
- Rokseth, B. and Skjong, S. (2015). Modelling of generic offshore ship operation simulator with focus on strong rigid body connections and energy conservation.
- Sciavicco, L. and Siciliano, B. (2000). *Modelling and Control of Robot Manipulators*. 2 edition.
- Skiong, S. and Pedersen, E. (2014). Control of hydraulic winch system (1/2).
- Skjong, S. (2014). Modeling, simulation and control of hydraulic winch system. Master's thesis, Norwegian University of Science and Technology.
- Sunde, H. C. (2014). 3d visualization of autonomous underwater robots. Master's thesis, Norwegian University of Science and Technology.
- Wolfgang, B. (2009). *Bond-Graph Methodology*. 1 edition.

Appendix A

Acronyms

3D	Three Dimensional
AHC	Active Heave Compensation
CAE	Computer Aided Engineering
CFD	Computational Fluid Dynamics
D-H	Denavit-Hartenberg
DLL	Dynamic Linked Library
DOF	Degree of Freedom
DP	Dynamic Positioning
EOM	Equations of Motion
FEA	Finite Element Analysis
FMI	Functional Mock-up Interface
GPS	Global Positioning System
HPU	Hydraulic Power Unit
JONSWAP	Joint North Sea Wave Project
NCS	Norwegian Continental Shelf

OSOM Offshore Ship Operation Model
OSV Offshore Construction Vessel
PMS Power Management System
PSV Platform Supply Vessel
PID Proportion Integration Differentiation
PHC Passive Heave Compensation
ROV Remotely Operated Vehicles
STL Standard Tessellation Language
VM System Vehicle-Manipulator System

Appendix B

C-code of Visualization DLL File

Please notice that the full original code is provided in the CD-ROM, here some pseudo-code are used in order to short the list and make a clear outline of programming. The code in gray represent the pseudo-code.

```
1
2 /*****
3
4 =====
5 C code for generate .dll file that communication between 20-sim and .
6   vtf file for GLview
7 (This C code is based on OSOM)
8 =====
9 *
10 * PURPOSE: To calculate the wave spectrum of irregular waves, receive
11 * the data from 20-sim and send the results back to 20-sim for
12 * simulations. In this C code, the code for .vtf file in GLview is
13 * printed, hence the simulation results can be visualized by 3D
14 * animation.
15 *
16 * METHOD: Use the theory explained in book 'Sealloads on ship and
17 * offshore structures' (O.M.Faltinsen, 1990) for calculation data of
18 * wave as well as the forces and torques that act on the vessel. Use
19 * DLL file to manage the calculation and communicate between 20-sim
20 * and GLview and generate the 3D animation file.
21 *
22 * VARIABLES
23 * name          I/O          format          description
24 * =====
```

16	* w	I	double	Frequency
17	* H	I	float	Significant wave height
18	* T	I	float	Top period
19	* gamma	I	float	Parameter of Jonswap/gamma
20	* t1	-	double	Temporary value
21	* t2	-	double	Temporary value
22	* t3	-	double	Temporary value
23	* t4	-	double	Temporary value
24	* t5	-	double	Temporary value
25	* t6	-	double	Temporary value
26	* sigma	-	double	Parameter of Jonswap
27	* w_p	-	double	Parameter of Jonswap
28	* S_value	O	float	The spectral value
29	* nw	I	int	No. wave components
30	* w[]	I	double	Frequencies
31	* zetaA []	I	double	Amplitude of all wave components
32	* n_el	I	int	No. of sea surface elements
33	* L_hav	I	float	Extent of sea surface in x- direction
34	* B_hav	I	float	Extent of sea surface in y- direction
35	* L_ship	I	double	The length of ship
36	* B_ship	I	double	The width of ship
37	* H_ship	I	double	The height of ship
38	* e []	I	float	Random phase for wave component
39	* x_max	-	float	Largest value for x
40	* x_min	-	float	Smallest value for x
41	* y_max	-	float	Largest value for y
42	* y_min	-	float	Smallest value for y
43	* dx	-	float	Length of an element in x- direction
44	* xw	-	float	Coordinate
45	* x	-	float	Coordinate
46	* i	-	int	Counter
47	* j	-	int	Counter
48	* h	-	int	Counter
49	* n_nodes	-	int	No. nodes
50	* surf_el	-	float	Surface elevation
51	* k	-	float	Wave number
52	* t	-	float	Time keeper
53	* R0_bij	I	double	Rotation matrices of the ship
54	* ni	I	double	Component in x direction for 20-sim
55	* nj	I	double	Component in y direction for 20-sim
56	* skip	I	int	number of skiped steps of animation
57	* p	-	double	Temporary value

```

58 * zeta []           -           double      Output values to 20-sim
    animation
59 * delta_w         -           double      Delta value of wave
    components
60 * w_up            I           double      Upper frequency of wave
    component
61 * w_dwn           I           double      Lower frequency of wave
    component
62 * tstp            I           double      counter for time steps in
    20-sim
63 * T1              -           double      Torque of ship
64 * T2              -           double      Torque of ship
65 * F3              -           double      Force of ship
66 * r0_             I           double      Position of element i, j, k
    in NED
67 * phi             I           double      Yaw angle of ship
68 * theta           I           double      Pitch angle of ship
69 * psi             I           double      Roll angle of ship
70 *
71 * Programmer: Wenzhe Zhang
72 * Date/Version: 03.04.2015 / 1.2
73 *****/
74
75 #include<stdio.h>
76 #include<math.h>
77 #include<time.h>
78
79 #define g 9.81
80 #define pi 3.1415
81 #define rho_w 1025
82
83 //-----global pointers for print/read file-----
84
85 static FILE *fp;
86 static FILE *fr;
87
88
89 //---Calculate the spectral value for a given frequency component---
90
91 double Spectrum(double w, double T, double H, double gamma){
92
93     S_value = (5 / (32 * pi)) * H * H * T * pow(w_p / w, 5) * exp((-5 * pow
        (t2, 4) / 4) * (1 - 0.287 * log(gamma)) * pow(gamma, exp(-(t5 * t5)
        / (2 * sigma * sigma))));
94
95     return S_value; }
96
97
98 //-----print initiate function-----
99
100 Define the parameters and variables here.
101

```

```

102 //start reading the PSV.vtf file as input and print to .vtf for GLview
      3D visualization
103 fr = fopen("PSV.vtf", "r");
104 fp = fopen("wave_ship_sim.vtf", "w");
105
106 //finish reading the PSV.vtf file
107 fclose(fr);
108 fprintf(fp, "\n")
109
110 //print nodes for the wave carpet and the first 10 elements are read
      from PSV.vtf file
111 fprintf(fp, "!Wave nodes\n");
112 fprintf(fp, "*NODES number\n");
113
114 //generate wave elements in coordinates and print them to .vtf file
115
116 VTF file descriptions here.
117 for (i = 1; i <= n_el;)
118 {
119 fprintf(fp, "%d %d %d %d\n", ((i*2)-1), ((i*2)+1), ((i*2)+2), (2 * i));
120
121 ++i; //generating the nodes for wave carpet
122 }
123 fprintf(fp, "\n");
124
125 Form the nodes to rectangle and form wave carpet finally.
126
127 //generate "GLVIEWGEOMETRY" command to form all the elements as a
      single file for GLview
128
129 VTF code for combine the elements here.
130
131 //-----calculate the wave surface elevation and force, torque of ship
      -----
132 {
133 Defination of parametes and variables.
134
135 //random phase for wave components
136 delta[0] = 0;
137 delta[1] = pi/2;
138 delta[2] = 1;
139 ...
140
141 //define parameters and variables for the ship motion
142
143 Forces and torque that act on the vessel, Define angles for rotation
      matrices.
144
145 //rotation matrix (R0_b)
146

```



```

147 | Rotation matrices of the DP vessel calculated from 20-sim, these
    | rotation matrices have the dynamics of linear motion as well.
148 |
149 | //-----wave calculate and print to .vtf file for GLview-----
150 |
151 | //print the results of wave carpet after skip steps
152 | surf_el = surf_el + zetaA[j] * sin(w[j] * t - k[j] * xw + delta[j]);
153 |
154 | fprintf(fp, "%f %f %f \n", xo, yo, surf_el);
155 | fprintf(fp, "%f %f %f \n", xo, yo, surf_el);
156 |
157 | //print the results of ship motion after skip steps
158 | fprintf(fp, "%f %f %f\n", R0_b11, R0_b12, R0_b13);
159 | fprintf(fp, "%f %f %f\n", R0_b21, R0_b22, R0_b23);
160 | fprintf(fp, "%f %f %f\n", R0_b31, R0_b32, R0_b33);
161 | fprintf(fp, "%f %f %f\n", r0i, r0j, r0k);
162 |
163 | //calcualte the forces and torques of ship
164 | p = rho_w * g * zetaA[h] * exp(-k[h] * H_ship / 2) * sin(w[h] * t + k[h]
    | * r0x + delta[h]);
165 |
166 | //localx-pos
167 | x = -L_ship / 2 * (1 + 2 * (1 - i) / ni) + di / 2;
168 |
169 | //localy-pos
170 | y = -B_ship / 2 * (1 + 2 * (1 - j) / nj) + dj / 2;
171 |
172 | //calculate forces and torques act on ship
173 | F3 = F3 + dA * p; //heave
174 | T1 = T1 + y * p * dA; //roll
175 | T2 = T2 - x * p * dA; //pitch
176 |
177 | //-----print function end-----
178 |
179 | //end print to the .vtf file
180 |
181 | fclose(fp);
182 | }
183 |
184 |
185 | //-----20-sim interface-----
186 |
187 | int UpdateWaves(double *inarr, int inputs, double *outarr, int outputs,
    | int major){
188 |
189 | //This function is called from 20-sim once each time "tstp"
190 | Waves(inarr[0], inarr[1], inarr[2], inarr[3], inarr[4], inarr[5], inarr[6],
    | inarr[7], inarr[8], inarr[9], inarr[10], inarr[11], inarr[12], inarr[13],
    | inarr[14], inarr[15], inarr[16], inarr[17], inarr[18], inarr[19],
    | inarr[20], inarr[21], inarr[22], inarr[23], inarr[24], inarr[25],
    | inarr[26], inarr[27], inarr[28], inarr[29], inarr[30], inarr[31],
    | outarr);

```

```
191 return 0;
192
193 }
194
195 int InitiateWaveSimulation(double *inarr, int inputs, double *outarr,
196     int outputs, int major){
197     //This function is called from 20-sim once before the simulation starts
198     //This function do not give any value back to the 20-sim, hence there
199     //are no outarr
200     printHeader(inarr[0], inarr[1], inarr[2], inarr[3], inarr[4], inarr[5])
201     ;
202     return 0;
203 }
204
205 int TerminateWaveSimulation(double *inarr, int inputs, double *outarr,
206     int outputs, int major){
207     //This function is called from 20-sim once after the simulation starts
208     //This function do not give any value back to the 20-sim, hence there
209     //are no outarr
210     printEnding(inarr[0], inarr[1]);
211     return 0;
212     //-----E N D-----
```

Appendix C

20-Sim Code for Wave Block in OSOM

```
1  /*
2  Original code in wave block in 20-Sim
3
4  Irregular waves and wave forces based on the JONSWAP spectrum.
5  Surface elevation and wave forces on ship calculated in the waves.dll.
6  Forces approximated by numerical integration of the dynamic pressure
7
8  */
9
10 //=====
11 parameters
12
13 // Ship parameters
14 real ni = 10; //# partitions of hull in x-direction
15 real nj = 5; //# partitions of hull in y-direction
16 real global L_ship, B_ship, H_ship;
17
18
19 // Parameters for the JONSWAP spectrum
20 real T_dwn = 3; // Lowest wave period component considered
21 real T_up = 21; // Largest wave period component considered
22 real nw = 50; // Number of wave components
23 real T = 8; // Spectrum peak period
24 real H = 6; // Significant wave height
25 real gamma = 3; // Spectrum shape parameter (sharpness)
26
```

```

27 //Inputs for the wave dll file
28 real L_hav = 400;
29 real B_hav = 200;
30 real n_el = 100;
31 real global skip = 50; //parameters for decided how many steps should
    skip in the final animation
32
33
34 //=====
35 variables
36 real w_dwn,w_up;
37 real global phi, theta, psi;
38 real x[32];
39 real out[24];
40 real tstp; // time step, for the counter in VTF file
41
42
43 real zeta1, zeta2, zeta3, zeta4, zeta5, zeta6, zeta7, zeta8, zeta9, zeta10,
    zeta11, zeta12, zeta13, zeta14, zeta15, zeta16, zeta17, zeta18, zeta19,
    zeta20, zeta21;
44 real x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12, x13, x14, x15, x16, x17, x18, x19
    , x20, x21;
45 real y1, y2;
46
47
48 real testforce_lin [3];
49 real global R0_b[3,3];
50
51 //Inputs for the 20-sim interface function
52 real in [6];
53 real in2 [2];
54
55
56 //=====
57 initialequations
58 w_dwn = 2 * pi / T_up;
59 w_up = 2 * pi / T_dwn;
60 tstp = 0;
61
62
63 x1 = -100;
64 x2 = -90;
65 x3 = -80;
66 x4 = -70;
67 x5 = -60;
68 x6 = -50;
69 x7 = -40;
70 x8 = -30;
71 x9 = -20;
72 x10 = -10;
73 x11 = 0;
74 x12 = 10;

```

```

75 x13 = 20;
76 x14 = 30;
77 x15 = 40;
78 x16 = 50;
79 x17 = 60;
80 x18 = 70;
81 x19 = 80;
82 x20 = 90;
83 x21 = 100;
84
85 y1 = -100;
86 y2 = 100;
87
88 //dll(call InitiateWaveSimulation), this should be run before "equation
    " and run only one time.
89
90 in =[w_dwn; w_up; nw; n_el; L_hav; B_hav];
91 out = dll('wave_ship_sim.dll', 'InitiateWaveSimulation', in);
92
93 //=====
94 equations
95 tstp = tstp + 1;
96
97 x = [ni;nj;L_ship;B_ship;H_ship;r0_b_0[1];r0_b_0[2];r0_b_0[3];phi;theta
    ;psi;w_dwn;w_up;nw;T;H;gamma;time; n_el; L_hav;B_hav; tstp;R0_b
    [1,1];R0_b[1,2];R0_b[1,3];R0_b[2,1];R0_b[2,2];R0_b[2,3];R0_b[3,1];
    R0_b[3,2];R0_b[3,3];skip];
98 out = dll('wave_ship_sim.dll', 'UpdateWaves', x);
99
100 zeta1 = out[1];
101 zeta2 = out[2];
102 zeta3 = out[3];
103 zeta4 = out[4];
104 zeta5 = out[5];
105 zeta6 = out[6];
106 zeta7 = out[7];
107 zeta8 = out[8];
108 zeta9 = out[9];
109 zeta10 = out[10];
110 zeta11 = out[11];
111 zeta12 = out[12];
112 zeta13 = out[13];
113 zeta14 = out[14];
114 zeta15 = out[15];
115 zeta16 = out[16];
116 zeta17 = out[17];
117 zeta18 = out[18];
118 zeta19 = out[19];
119 zeta20 = out[20];
120 zeta21 = out[21];
121
122 tau_lin = [0;0;out[22]];

```

```
123 tau_ang = [out[23];out[24];0];
124
125 testforce_lin = R0_b*tau_lin;
126
127 //=====
128 finalequations
129
130 //dll(TerminateWaveSimulation) This should be run after "equation" and
    run only one time.
131
132 in2 = [tstp;skip];
133 out = dll('wave_ship_sim.dll', 'TerminateWaveSimulation', in2);
```

Appendix D

VTF File Structure of Example 4.2

```
1 *VTF-1.00
2
3 !NODES1:
4 *NODES 1
5 %WITH_ID
6 10 10.0 4.0 -2.0
7 20 -10.0 4.0 -2.0
8 30 -10.0 -4.0 -2.0
9 40 10.0 -4.0 -2.0
10 50 10.0 4.0 3.0
11 60 -10.0 4.0 3.0
12 70 -10.0 -4.0 3.0
13 80 10.0 -4.0 3.0
14
15 !NODES2:
16 *NODES 2
17 %WITH_ID
18 90 4.0 2.0 3.0
19 100 2.0 2.0 3.0
20 110 2.0 -2.0 3.0
21 120 4.0 -2.0 3.0
22 130 4.0 2.0 5.0
23 140 2.0 2.0 5.0
24 150 2.0 -2.0 5.0
25 160 4.0 -2.0 5.0
26
27 !CARPET
28 *NODES 3
29 %WITH_ID
```

```
30 | 1001 -50 50 0
31 | 1002 -25 50 0
32 | 1003 0 50 0
33 | 1004 25 50 0
34 | 1005 50 50 0
35 | 1006 -50 25 0
36 | 1007 -25 25 0
37 | 1008 0 25 0
38 | 1009 25 25 0
39 | 1010 50 25 0
40 | 1011 -50 0 0
41 | 1012 -25 0 0
42 | 1013 0 0 0
43 | 1014 25 0 0
44 | 1015 50 0 0
45 | 1016 -50 -25 0
46 | 1017 -25 -25 0
47 | 1018 0 -25 0
48 | 1019 25 -25 0
49 | 1020 50 -25 0
50 | 1021 -50 -50 0
51 | 1022 -25 -50 0
52 | 1023 0 -50 0
53 | 1024 25 -50 0
54 | 1025 50 -50 0
55 |
56 | !BOXHULL:
57 | *ELEMENTS 1
58 | %NAME "BOX1"
59 | %DESCRIPTION "box hull"
60 | %NODES #1
61 | %WITH_ID
62 | %HEXAHERONS
63 | 1 10 20 30 40 50 60 70 80
64 |
65 | !BOXTOP:
66 | *ELEMENTS 2
67 | %NAME "BOX2"
68 | %DESCRIPTION "box top"
69 | %NODES #2
70 | %WITH_ID
71 | %HEXAHERONS
72 | 2 90 100 110 120 130 140 150 160
73 |
74 | !CARPET
75 | *ELEMENTS 3
76 | %NAME "CARPET"
77 | %DESCRIPTION "CARPET"
78 | %NODES #3
79 | %WITH_ID
80 | %QUADS
81 | 1 1001 1002 1007 1006
```



```
82 | 2 1002 1003 1008 1007
83 | 3 1003 1004 1009 1008
84 | 4 1004 1005 1010 1009
85 | 5 1006 1007 1012 1011
86 | 6 1007 1008 1013 1012
87 | 7 1008 1009 1014 1013
88 | 8 1009 1010 1015 1014
89 | 9 1011 1012 1017 1016
90 | 10 1012 1013 1018 1017
91 | 11 1013 1014 1019 1018
92 | 12 1014 1015 1020 1019
93 | 13 1016 1017 1022 1021
94 | 14 1017 1018 1023 1022
95 | 15 1018 1019 1024 1023
96 | 16 1019 1020 1025 1024
97 |
98 | *GLVIEWGEOMETRY 1
99 | %ELEMENTS
100 | 1 2 3
101 |
102 | *TRANSFORMATIONRESULT 4
103 | %ELEMENT_BLOCK_ID #1
104 | 1 0 0
105 | 0 1 0
106 | 0 0 1
107 | 0 0 0
108 |
109 | *TRANSFORMATIONRESULT 5
110 | %ELEMENT_BLOCK_ID #2
111 | 1 0 0
112 | 0 1 0
113 | 0 0 1
114 | 0 0 0
115 |
116 | *TRANSFORMATIONRESULT 6
117 | %ELEMENT_BLOCK_ID #1
118 | 0.7071 -0.7071 0
119 | 0.7071 0.7071 0
120 | 0 0 1
121 | 0 0 0
122 |
123 | *TRANSFORMATIONRESULT 7
124 | %ELEMENT_BLOCK_ID #2
125 | 0.7071 -0.7071 0
126 | 0.7071 0.7071 0
127 | 0 0 1
128 | 0 0 0
129 |
130 |
131 | *TRANSFORMATIONRESULT 8
132 | %ELEMENT_BLOCK_ID #1
133 | 1 0 0
```

```
134 | 0 1 0
135 | 0 0 1
136 | 0 0 0
137 |
138 | *TRANSFORMATIONRESULT 9
139 | %ELEMENT_BLOCK_ID #2
140 | 1 0 0
141 | 0 1 0
142 | 0 0 1
143 | 0 0 0
144 |
145 | *RESULTS 1
146 | %DIMENSION 3
147 | %PER_NODE #3
148 | 0. 0. 0.
149 | 0. 0. 3.
150 | 0. 0. 0.
151 | 0. 0. 0.
152 | 0. 0. 0.
153 | 0. 0. 0.
154 | 0. 0. 2.
155 | 0. 0. 0.
156 | 0. 0. 0.
157 | 0. 0. 0.
158 | 0. 0. 0.
159 | 0. 0. 0.
160 | 0. 0. 3.
161 | 0. 0. 0.
162 | 0. 0. 0.
163 | 0. 0. 0.
164 | 0. 0. 0.
165 | 0. 0. -2.
166 | 0. 0. 0.
167 | 0. 0. 0.
168 | 0. 0. -1.
169 | 0. 0. 0.
170 | 0. 0. 0.
171 | 0. 0. 0.
172 | 0. 0. 1.
173 |
174 | *RESULTS 2
175 | %DIMENSION 3
176 | %PER_NODE #3
177 | 0. 0. 1.
178 | 0. 0. 2.
179 | 0. 0. 3.
180 | 0. 0. -2.
181 | 0. 0. -1.
182 | 0. 0. 0.
183 | 0. 0. 1.
184 | 0. 0. -2.
185 | 0. 0. 3.
```

```
186 0. 0. 2.
187 0. 0. 0.
188 0. 0. 0.
189 0. 0. 0.
190 0. 0. 1.
191 0. 0. 0.
192 0. 0. 0.
193 0. 0. 0.
194 0. 0. 3.
195 0. 0. 0.
196 0. 0. 0.
197 0. 0. 0.
198 0. 0. 0.
199 0. 0. 0.
200 0. 0. 0.
201 0. 0. 0.
202
203 *RESULTS 3
204 %DIMENSION 3
205 %PER_NODE #3
206 0. 0. 0.
207 0. 0. 1.
208 0. 0. 0.
209 0. 0. 0.
210 0. 0. 2.
211 0. 0. 0.
212 0. 0. 0.
213 0. 0. 3.
214 0. 0. 0.
215 0. 0. -1.
216 0. 0. -3.
217 0. 0. 2.
218 0. 0. 1.
219 0. 0. -2.
220 0. 0. 1.
221 0. 0. 2.
222 0. 0. 1.
223 0. 0. 3.
224 0. 0. 0.
225 0. 0. 0.
226 0. 0. 0.
227 0. 0. 0.
228 0. 0. 0.
229 0. 0. 0.
230 0. 0. 0.
231
232 *GLVIEWVECTOR 1
233 %NAME "Displacement"
234 %STEP 1
235 1
236 %STEP 2
237 2
```

```
238 %STEP 3
239 3
240
241 *GLVIEWTRANSFORMATION 1
242 %STEP 1
243 4, 5
244 %STEP 2
245 6, 7
246 %STEP 3
247 8, 9
```

Appendix E

Parameters of Hydraulic System

Parameters of Components	Value (Unit)
Pump flow rate for cylinders	0.04 (m^3/s)
Pump flow rate for motor	0.04 (m^3/s)
Pump leakage factor	5×10^{-9}
Bulk modulus	1.6×10^9 or variable
Polytropic index	1.4
Check valve p_{closed}	1.0×10^5 (Pa)
Check valve p_{open}	1.5×10^5 (Pa)
Check valve discharge coefficient	0.8
Check valve flow area	2.0×10^{-4} (m^2)
System accumulator H volume	0.4 (m^3)
System accumulator L volume	0.35 (m^3)
Safety valve maximum pressure	150 (bar)
Safety valve discharge coefficient	0.9
Safety valve bumper limitation	0 to 0.1 (m)
Compensator differential pressure	3 (bar)
Compensator discharge coefficient	0.9
Compensator bumper limitation	0 to 0.1 (m)

Parameters of Components	Value (Unit)
4/3 directional valve discharge coefficient	0.9
Cylinder 1 piston diameter	0.4 (m)
Cylinder 1 initial volume	0.005 (m ³)
Cylinder 1 piston mass	50 (kg)
Cylinder 2 piston diameter	0.4 (m)
Cylinder 2 initial volume	0.005 (m ³)
Cylinder 2 piston mass	40 (kg)
Load	20 (metric tons)
Length of crane boom	8 (m)
Position of boom COG from joint 2	3 (m)
Mass of crane boom	2.5 (metric tons)
Joint 2 bumper limitation about horizontal	0° to 60°
Length of crane boom	6 (m)
Position of jib COG from joint 3	2 (m)
Mass of crane jib	1 (metric tons)
Joint 2 bumper limitation about boom	-150° to 0°
Height of crane base	6 (m)
Mass of crane base	5.5 (metric tons)
Motor volume displacement	0.005 (m ³ /rad)
Motor leakage factor	5 × 10 ⁻⁹
Position controller 1 P gain	1.3 × 10 ⁻²
Position controller 2 P gain	6 × 10 ⁻³
Position controller 3 P gain	9 × 10 ⁻³
Position controller speed limitation	Variable
Gear box ratio	40:1