

# Virtual Reality Simulation of Ships and Ship-Mounted Cranes

Mohammed F. Daqaq

Thesis submitted to the Faculty of the  
Virginia Polytechnic Institute and State University  
in partial fulfillment of the requirements for the degree of

Masters of Science  
in  
Engineering Mechanics

Ali H. Nayfeh, Chair  
Ronald Kriz  
Scott L. Hendricks  
Ziyad N. Masoud

April 28, 2003  
Blacksburg, Virginia

Keywords: Virtual Reality, CAVE, OpenGL, DIVERSE, Ship, Crane, Motion Base.

Copyright 2003, Mohammed F. Daqaq

# Virtual Reality Simulation of Ships and Ship-Mounted Cranes

Mohammed F. Daqaq

(ABSTRACT)

We present a virtual simulation of ships and ship-mounted cranes. The simulation is carried out in a Cave Automated Virtual Environment (CAVE). This simulation serves as a platform to study the dynamics of ships and ship-mounted cranes under dynamic sea environments and as a training platform for ship-mounted crane operators. A model of the (Auxiliary Crane Ship) T-ACS 4-6 was built, converted into an OpenGL C++ API, and then ported into the CAVE using DiverseGL (DGL). A six-degrees-of-freedom motion base was used to simulate the actual motion of the ship. The equations of motion of the ship are solved using the Large Amplitude Motion Program (LAMP), while the equations of motion of the crane payload are numerically integrated; the interaction between the payload and the ship is taken into consideration. A nonlinear delayed-position feedback-control system is applied to the crane and the resulting simulation is used to compare the controlled and uncontrolled pendulations of the cargo. Our simulator showed a great deal of realism and was used to simulate different ship-motion and cargo transfer scenarios.

This work received support from the Office of Naval Research under Grant No. N00014-99-1-0562.

# Dedication

*To my parents, the greatest parents in the world ...*

*To Palestine ... my homeland ... my past, present, and future ...*

# Acknowledgments

I would like to thank God for blessing me with the family, friends, and guidance that I needed to accomplish this work. Also I am very grateful to my parents and family for their endless love and support. My deep appreciation and sincere gratitude to my advisor Dr. Ali H. Nayfeh, for allowing me to grow as a student, his insightful guidance and distinguished supervision has been invaluable.

I would also like to express my deep appreciation to my favorite teacher and committee member Dr. Scott Hendricks. I would like to thank my other committee members: Dr. Ron Kriz for being always there to push me ahead when I needed him to, his complete trust and endless support are greatly appreciated and Dr. Ziyad Masoud for his remarkable comments, invaluable help, and strong friendship.

Greatest thanks go to my friend Ying Chen for her help throughout the work. Without her, the completion of this project would have been much harder. I would also like to thank Dr. Moumen Idres who collaborated with me in some of the work in this thesis.

Special thanks go to the Nonlinear Dynamics Research Group, in particular, Drs. Eihab Abdel-Rahman and Haider Arafat for their deep discussions and valuable suggestions, and Mohammed Younis for his help, especially in my new life in Blacksburg. Words of thanks go to the Virginia Tech CAVE group specially Dr. Lance Arsenault and Chris Logie for their help. I also would like to thank the Digital Ship Group for their support.

I am especially grateful to my friends, Mohammed Bundukji, Khaled Alhazza, Mohammed Hamlan, Majed Majeed, and Fadi Mantash. In addition to their help, love, and support; they were the greatest friends and the best company.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background and Motivation . . . . .	1
1.1.1	Virtual Reality . . . . .	1
1.1.2	Cave Automated Virtual Environment . . . . .	3
1.1.3	Visual Scene Image-Generation System . . . . .	5
1.2	Literature Review . . . . .	6
1.3	Thesis Objectives . . . . .	8
1.4	Thesis Organization . . . . .	9
<b>2</b>	<b>Theory and Problem Formulation</b>	<b>11</b>
2.1	Mathematical Model . . . . .	11
2.1.1	Ship Equations of Motion . . . . .	12
2.1.2	Cargo Equations of Motion . . . . .	14
2.1.3	Equations Governing $\gamma$ and $\delta$ . . . . .	17
2.2	Computational Technique . . . . .	20
2.2.1	Cargo Pendulation Control . . . . .	21
<b>3</b>	<b>Modeling and Simulation</b>	<b>23</b>
3.1	Building the Model . . . . .	24
3.2	Model Animation . . . . .	27
3.3	Porting the Model to the CAVE . . . . .	28

3.4	Motion Platform . . . . .	32
<b>4</b>	<b>Results of CAVE Simulation</b>	<b>36</b>
4.1	Response of Ships to Dynamic Sea Environment . . . . .	36
4.2	Animation and Visualization of Ship and Crane Systems in a Dynamic Sea Environment . . . . .	37
4.3	Control of Moving Loads Aboard a Ship . . . . .	38
4.4	Crane Operator Training . . . . .	40
<b>5</b>	<b>Conclusion and Future Considerations</b>	<b>44</b>

# List of Figures

1.1	A schematic drawing showing a 3-D object projected inside the CAVE using four projectors. . . . .	4
1.2	The complete components of a CAVE. <sup>2</sup> . . . . .	5
2.1	Typical T-ACS cargo transfer scenario. . . . .	12
2.2	The two coordinate systems used to describe the motion of the ship. . . . .	13
2.3	Position vectors of the cargo being transferred. . . . .	14
2.4	(a) Top view of the boom. The reference axis is rotated by an angle $\beta$ around the positive $z$ -axis. (b) The double-prime coordinate is formed by a rotation of an angle $\gamma$ around the primed $y$ -axis. (c) The triple prime coordinate, which lies on the cable, is formed by a rotation of an angle $\delta$ around the double primed $x$ -axis. . . . .	18
3.1	A block diagram showing the various steps necessary for a complete CAVE simulation . . . . .	23
3.2	LightWave3D ship model. . . . .	24
3.3	Different ship views showing the ship transformed into polygons by PolyTrans. . . . .	25
3.4	Side view of the ship-mounted crane on a desktop screen. . . . .	27
3.5	The DGL coordinate cube. . . . .	30
3.6	A GUI used to switch from one scenario to the other during the CAVE simulation. . . . .	31

3.7	A block diagram showing data flow from the I/O devices to the graphical virtual simulation in the CAVE. . . . .	32
3.8	6-DOF 200E, Model 170E122A motion base. <sup>18</sup> . . . . .	33
3.9	A block diagram showing the data flow from the text files to the motion base.	35
4.1	The T-ACS ship model as part of a simulation to study the response of ships to dynamic sea environments. . . . .	37
4.2	The T-ACS crane-ship as part of a simulation that was carried out in the CAVE at sea state 3. . . . .	38
4.3	The crane operator cabin as part of a crane operator training. The nonlinear-delayed position feedback control is applied to the crane. . . . .	39
4.4	The crane operator cabin as used in crane-operator training at seastate 3. . .	41
4.5	The crane operator cabin used in a crane operator training at seastate 5. . .	42
4.6	The crane operator cabin used in a crane operator training. A heading wave is used to cause a two-to-one ship internal resonance. . . . .	43

# List of Tables

3.1	Maximum capabilities of the motion base. . . . .	34
-----	--	----

# Chapter 1

## Introduction

### 1.1 Background and Motivation

#### 1.1.1 Virtual Reality

Virtual Reality (VR) can be best defined as a graphical interactive computer-based environment that creates a virtual representation of the real world, in which the user can interact and participate. It is used to create a simulated environment when the real world is dangerous to the user, harmful to the environment, consumes time and material, inconvenient, or costly. Also, VR allows engineering researchers to gain a better insight into problems and creates a better understanding to deal with them.

Virtual simulations have been used to create many computer-based environments in military, space training, and medical training. It has also been used in many engineering applications, such as rapid prototyping, flight simulators, molecular geometry, robotics, transportations, and building constructions.

It was in the late 1950's when the idea of the most primitive VR projector, or what is known now as the computer screen, inspired into the mind of Douglas Engelbart, an electrical engineer and a former naval radar technician. At that time, computers were the size of a big room and were only used by those who were familiar with the old esoteric programming

languages. As a result of his experience in radars, Douglas Engelbart knew that signals could be viewed on a screen. Rather than limiting computers to number crunching and calculations, he thought that they could be used as machines to view data and different graphical virtual objects.

In the 1960's when the first computer based on transistors and not on vacuum tubes was built, the idea of connecting a screen to the computer became more realistic. The rapid development in communications and their direct intersection with computer technology and graphical applications resulted in the development of personal computers and computer graphics, and finally in the emergence of the VR technology.

The development of VR in its current shape goes back to the late 1970's and early 1980's with the introduction of the computer-graphics-based Computer Aided Design and Drafting (CADD) systems, which was used to provide a 2-D virtual environment. Current visualization technology benefited from the great contributions of the motion picture industry, the military, and the automotive and aerospace industries. As an example, the fear of nuclear attacks during the cold war with the former Soviet Union prompted the United States to develop a radar system that processes a great amount of data in real time and visualizes them in a form understandable by humans. Another example is military aircraft designers who built VR environments to teach pilots how to fly on the ground, which made training safer and cheaper. Early flight simulators simulated the cockpit with a motion platform that pitched and rolled, however, it lacked virtual feedback. All of these contributions are reflected in the creation of 3-D, 4-D, and the immersive visualizations not only on big workstations but also using small desktop computers.

Brill<sup>1</sup> proposed six types of VR:

1. Immersive first person: the user is immersed into the environment using accouterments, such as head mounted stereoscopic displays, gloves, 3-D sound audio system and body suits.
2. Cab simulator environment: this is used to create an active environment for the user that looks exactly the same as the real environment. It is used to simulate vehicle

training.

3. Through the window: the user sees the 3-D environment through a computer window, and he controls the world using I/O device.
4. Mirror world: people see duplication of themselves, which they can control using any I/O device.
5. Waldo world: the user is connected in real time through a remote controlled mechanical manipulator.
6. CAVE: a 3-D rear projector theater made up of 3 walls and a floor, the pictures are projected in stereo and viewed using stereo glasses.

Any comparison between a 3-D and a 2-D virtual environment shows that 3-D environments create better understanding of physical models. They also provide certain kind of interaction between the user and the scene, which can not be provided by a 2-D simulation. Moreover, the integration of 3-D environments with a motion base enhances the capabilities of the simulation and gives a greater sense of reality, which the user would not be able to feel in a 2-D environment.

For all of the above mentioned advantages, we decided to carry out our simulation in a 3-D CAVE, a brief description of the CAVE is given in the next section.

### **1.1.2 Cave Automated Virtual Environment**

The CAVE is a room size, projection-based VR system that projects the scene onto three walls and a floor. The graphics are rear projected onto the walls and there is an overhead projector pointing into a mirror that reflects the scene onto the floor, Figure 1.1.

The projected scene is animated and controlled using an SGI Onyx Infinite Reality (IR) workstation. A viewer moving inside the CAVE wears a stereo graphics' crystal eyes liquid crystal shutter glass and a 6-DOF tracking device that calculates the new stereoscopic projection each time the user moves inside the CAVE. The stereographic system provides

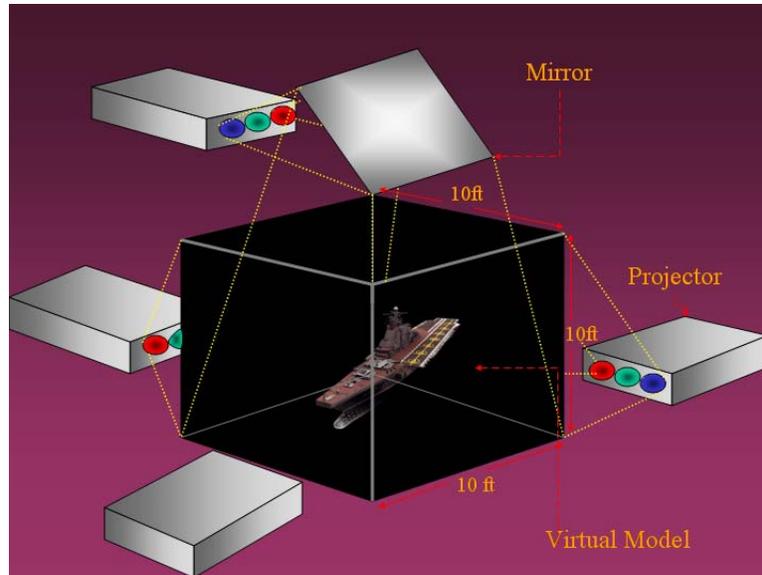


Figure 1.1: A schematic drawing showing a 3-D object projected inside the CAVE using four projectors.

liquid crystal viewing lenses mounted in an eye glass frame, whose polarities are switched on command. These commands provide the system with infrared signals that are synchronized with the rendering update rate of the image generation system. These infrared emitters are mounted at various locations, which can be rearranged if needed to ensure that the receiver on the eyeglass frame receives the command signals regardless of their location and orientation. The user can also use a hand wand to help him interact with the virtual environment, Figure 1.2.

Also the CAVE contains a 3-D sound system that has four components: an audio/serial option, an MIDI interface, speakers, and a synthesizer. The speakers are located on the four corners of the CAVE, and the sound command or “aiff” files are generated internally or transformed to sound by the synthesizer. The system has the capacity for 10 user-defined sounds for each exercise.

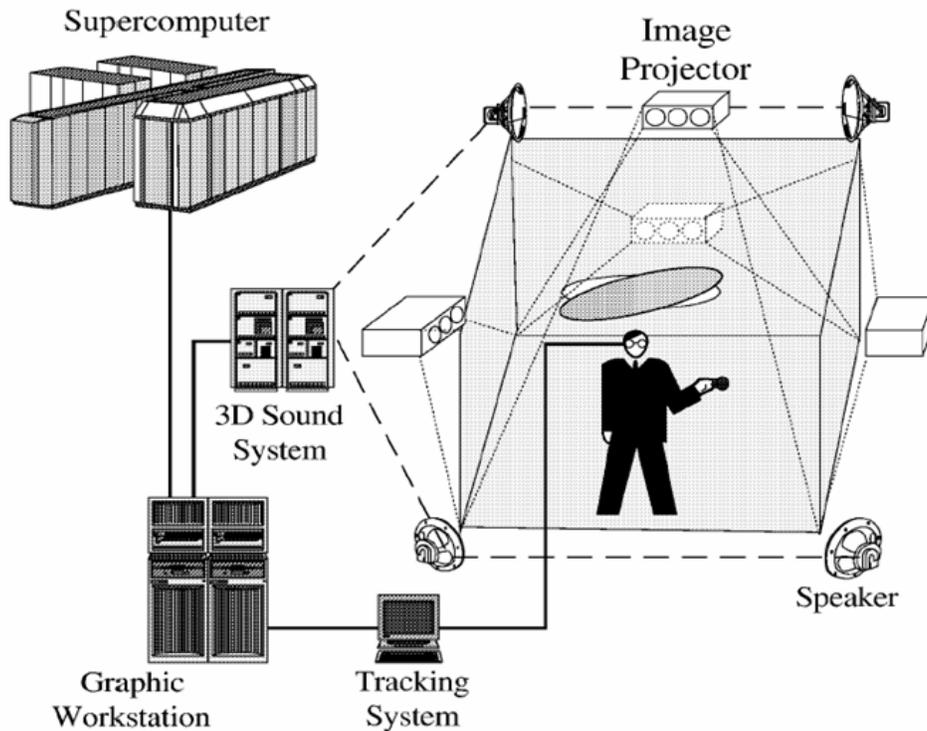


Figure 1.2: The complete components of a CAVE.<sup>2</sup>

### 1.1.3 Visual Scene Image-Generation System

The scene in the CAVE is generated using a three-pipe visual scene image-generation system. It consists of a Silicon Graphics Power Onyx Infinite Reality (IR) computer running the virtual view, software, and an InterSence IS-900 six-degrees-of-freedom motion tracker. In 1997, this system represented state-of-the-art hardware and software technology for real-time visual simulation.

The visual scene image-generation system generates highly realistic scenes of ships, crane systems, and payloads in real time. Its capabilities include a high-scene update rate (smooth motion), a high-scene content, and texture images. The visual scene image-generation system has four output channels, which simultaneously generate mono and stereo images that are displayed by the four CAVE projectors. The system uses a motion tracker to measure the

position and orientation of the head of the system operator to render visual images from his eye point of view and through his line of sight. The IS-900 motion tracker performs position tracking by means of a linear accelerometer combined with an ultrasonic trilateration for drift correction. It performs orientation tracking by means of an inertial sensor unit that measures angular rates. The combined system can compensate for image-generation latencies by predicting in advance motions up to 50 ms. The system has excellent performance characteristics and is immune to either an electromagnetic or an acoustic interference. These features prevent the generation of “slouchy” images, which can be a significant cause of simulator-induced sickness.

The virtual scene is built using OpenGL and Performer software. The visual image resolution is 1280 by 1024 pixels per channel, with a texture memory capability of 272 MB. A scene update rate of approximately 60Hz is maintained with a scene content of at least 10000 visible polygons per channel.

To integrate the hardware and software of the CAVE one needs an Application Programming Interface (API). The API consists of a library of function calls, which the user employs to either provide data to or obtain data from other parts of the simulator. Currently, there are three current APIs used in CAVES: EVL’s CAVE-lib, Iowa State University VR-Juggler, and Virginia Tech’s DIVERSE (Device Independent Virtual Environments- Reconfigurable, Scalable, Extensible).

## 1.2 Literature Review

A tremendous amount of research in the area of VR has been published. Many people studied implementation of VR simulations in industrial applications.<sup>3,4</sup> Others successfully applied training in virtual environments in several areas. In one demonstration, Freund, Rossmann, and Thorsten<sup>5</sup> created a VR environment for the simulation of excavators and construction machines. The simulator worked in real time and presented the interaction between the excavators and the bulk material. They presented their work using a stereoscopic panorama

projection technique with a combination of a Head Mounted Device (HMD) and a head tracking system. Also, many other training environments have been created, such as the virtual robotic manufacturing line for worker training.<sup>6,7</sup> A large scale complex virtual environment was developed by the US Army to provide a training base for ground combat of tanks and mechanized infantry forces.<sup>8</sup> Also, the ground support flight team of the Hubble Telescope received training in a virtual environment.<sup>9</sup>

Wilson, Mourant, Li, and Xu<sup>10</sup> developed a real-time virtual environment for training overhead crane operators using a simple mathematical model consisting of a 2-DOF crane and load model. The resulting two second-order differential equations were solved using a fourth-order Runge-Kutta technique. Their simulator provides a trainee with a 3-D virtual environment of learning and practicing the skills needed for productive crane operation in an actual factory. They used a CAD software package to replicate the factory floor and its surrounding machines and tools, the CAD model was transformed into an API called *Renderwave* and then visualized on a 2-D desktop screen.

Jiing-Yih, Ji-Liang, Jiun-Ren, Ming-Chang, and Chung-Yun<sup>11</sup> developed a virtual simulation system for truck-crane operator training. They used a 3-DOF crane model with a 1-DOF platform to simulate the vibrations of the truck chair due to crane operation. Their model was built by *Pro-Engineer* and then saved as polygonal data to be visualized on a single screen using a single 3-D projector. Although their model was a 3-DOF system and had 1800 polygons which were projected on a single screen, they were still unable to run their simulation in real time. As a result, they restored to another approach in which they used a data base consisting of a series of hook motions under different operating conditions.

Chin-Teng, I-Fang, and Jiann-Yaw<sup>12</sup> created a multipurpose virtual-reality-based motion simulator. They studied the stability of the 6-DOF Stewart Platform used as a flight simulator, and integrated this platform with different VR scenes developed using the *Coryphaeus* VR software.

### 1.3 Thesis Objectives

The objectives of the present work is to create state-of-the-art virtual models in a state-of-the-art ship and crane physics-based test-bed at the CAVE at Virginia Tech (VT). The simulator combines existing VT CAVE capabilities with new hardware and software capabilities required to support the ship and crane simulator and raises the simulation to a new level of reality.

This simulator serves as a platform for testing new technologies in the following areas:

1. Response of ships to a dynamic sea environment.
2. Integrated ship-motion prediction and control.
3. Control of moving loads aboard a ship.
4. Animation and visualization of ship and crane systems in a dynamic sea environment.
5. Ship- and crane-operator training.

To improve the realism of the state-of-the-art simulations, the fixed floor of the CAVE was replaced with a motion base that simulates the dynamics of the ship. When one looks at the sea through the (simulated) bridge windows of a simulator with a fixed floor, the feeling is not the same as when one looks at the sea through the bridge windows of an actual ship.<sup>13</sup> The big difference is that, in the simulator, the floor is not moving. Thus even when the relative motions between the ship and the sea are portrayed correctly, one may experience sea sickness in one situation but not in the other, or one may be able to maneuver the ship well in one situation but not in the other, or one may be able to operate a crane well in seastate three or higher in one situation but not in the other, or one maybe able to smoothly land a helicopter in a high seastate in one situation but not in the other, etc. It has long been recognized that giving the person in the simulator a realistic motion greatly enhances the value of the simulation and, hence, of the training. For this reason, moving platform simulators became an important tool for training aircraft pilots

and heavy-machinery operators. The introduction of the motion base combines analytical tool development with virtual environment experience as a new and innovative method for evaluating model results. As a result one can either virtually move about and inspect the ship or experience its motion in high seastates.

In our simulation, the crane operator functions in a highly realistic virtual environment, complete with high fidelity, 270 degrees scene visualization, ambient sound, base motion, physical control console, and a chair. Having such a simulator makes it more efficient to understand the mathematical physical models as they occur in the simulations. These state-of-the-art simulations will feedback informations about the continuing development of physical models, as well as, feed forward into the design and training processes.

## 1.4 Thesis Organization

The thesis presents a brief description of the procedure used to create a virtual environment for ships and ship-mounted cranes. It provides an insight into the different methods used to program VR scenes and concentrates on VR applications for mathematical and physical models.

In Chapter 2, we provide a description of the T-ACS 4-6 crane-ship that serves as the dynamical model for our simulation. We describe the motion of a ship-mounted crane in a dynamic sea environment. We derive the equations of motion that describe the 6-DOF of the ship and the equations of motion that describe the payload. We take into consideration the interaction between the cargo and the ship. Also we give a short description of the LAMP. This software is used to solve the equations of motion of the ship. Finally, we discuss application of the nonlinear delayed-position feedback control system to the ship-mounted crane and show its effect on suppressing cargo pendulations.

In Chapter 3, we describe the procedure of building a VR model and porting it to the CAVE. We give a short description of OpenGL, and show how to apply dynamics to visual animations. A brief description of DIVERSE an API that serves as a software hardware

integrator at VT CAVE, is provided. This chapter provides a description of the motion platform, explains its modes, degrees of freedom, restrictions, and how to use the DIVERSE Utility Toolkit (DTK) to move it. In addition, we show some Graphical User Interfaces (GUIs) provided by DTK and used in the simulation.

In Chapter 4, we show some figures taken from the CAVE simulation; we discuss them and comment on the visual scenes.

Finally in Chapter 5, we present our conclusion and recommendations for future work.

# Chapter 2

## Theory and Problem Formulation

### 2.1 Mathematical Model

Ship-mounted cranes are used to transfer cargo from large, heavy ships to lighter smaller ones at sea when a port is not available to accommodate the heavy ship. As a result of sea excitations, even in seastates that are relatively mild, large motions of the crane ships can develop, which in return cause large motions of the crane payload, Figure 2.1.

To simulate this problem, Idres, Youssef, Mook, and Nayfeh<sup>14</sup> used an 8-DOF coupled crane-ship dynamic model for the motion of the crane-ship and the payload. This model accounts for the hull motions coupled with nonlinear large payload swings. The ship and crane were treated as one rigid body and the model included arbitrary, bi-angular swings of the suspended load coupled with the 6-DOF of the ship: surge, sway, heave, roll, pitch, and yaw, in addition to the 2-DOF of the cable in-plane and out-of-plane sway angles.

The dynamic model was applied to a TACS 4-6 crane-ship whose basic dimensions are: length between the perpendiculars  $LBP = 177$  m, and beam = 23 m. The crane data are: boom length  $l_B = 36.9$  m, cargo mass  $m_{cargo} = 30$  tons, luff angle, slew angle, base location, and cable length could be changed for different scenarios. In this section we show the derivation of the equations of motion of the ship and the payload.



Figure 2.1: Typical T-ACS cargo transfer scenario.

### 2.1.1 Ship Equations of Motion

As shown in Figure 2.2, two coordinate systems are used to describe the motion of the ship. One of the coordinate systems is fixed to the ground and presents a Newtonian reference frame or an inertial frame of reference. The other system is a body fixed frame, which is fixed to the ship and moves with it. The ship is treated as a rigid body, with fixed moments and products of inertia in the body fixed frame. In Figure 2.2 Point  $A$  is the origin of the body fixed coordinate and is arbitrarily chosen on the mid-plane of the ship near the ship's center of gravity  $G$ .

In vector form, the equations of motion are written as

$$\vec{F} = m_{ship} \vec{a}_G \quad (2.1)$$

$$\vec{M}_A = I_A \vec{\omega} + \vec{r}_G \times m_{ship} \vec{a}_A + \vec{\omega} \times I_A \vec{\omega} \quad (2.2)$$

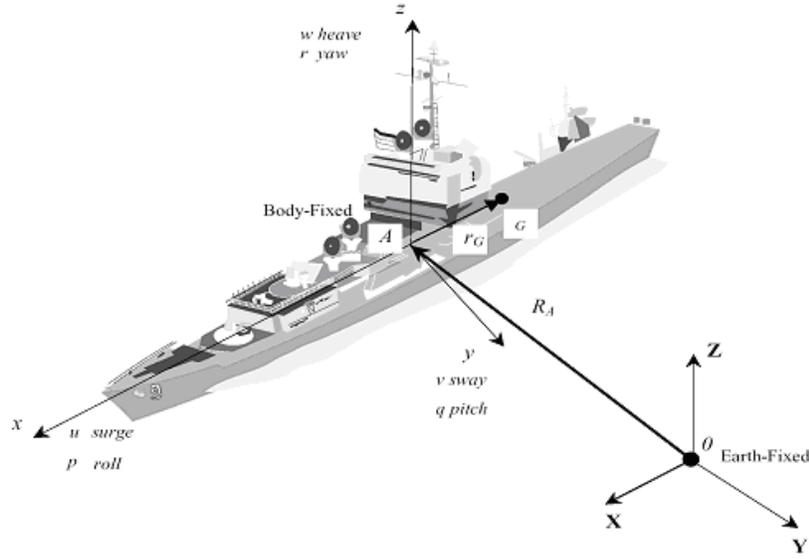


Figure 2.2: The two coordinate systems used to describe the motion of the ship.

where  $\vec{F}$  is the force vector,  $m_{ship}$  is the ship mass,  $\vec{r}_G$  is the position vector of the center of mass,  $\vec{a}_A = \vec{V} + \vec{\omega} \times \vec{V}$  is the linear acceleration of the ship,  $\vec{V} = (u, v, w)$  is the linear velocity of the ship,  $\vec{M}_A$  is the moment vector around point A,  $I_A$  is the moment of inertia matrix around A, and  $\vec{\omega} = (p, q, r)$  is the ship angular velocity vector. Expanding equations (1.1) and (2.1) gives

$$\vec{F}_{hydro} + \vec{F}_{c/s} = m_{ship}[\vec{a}_A + \vec{\omega} \times \vec{r}_G + \vec{\omega} \times (\vec{\omega} \times \vec{r}_G)] \quad (2.3)$$

$$\vec{M}_{hydro} + \vec{M}_{c/s} = I_A \vec{\omega} + \vec{r}_G \times m_{ship} \vec{a}_A + \vec{\omega} \times I_A \vec{\omega} \quad (2.4)$$

where  $\vec{F}_{hydro}$  and  $\vec{M}_{hydro}$  are the hydrodynamic and hydrostatic forces and moments exerted on the ship, respectively, and  $\vec{F}_{c/s}$  and  $\vec{M}_{c/s}$  are the forces and moments exerted on the ship by the payload.

### 2.1.2 Cargo Equations of Motion

In Figure 2.3 a detailed view of the four position vectors used to describe the motion of the cargo are shown. When the ship is at even keel, the  $xy$  –  $plane$  is parallel to the Earth's surface and the  $z$  –  $axis$  is pointing upwards. Point 0 represents the Earth fixed coordinate, whereas Point A is an arbitrarily chosen point on the ship. Point B represents the crane base and Point C represents the projection of Point C onto the  $xy$  –  $plane$ . Points B, C and C form the plane of the crane.

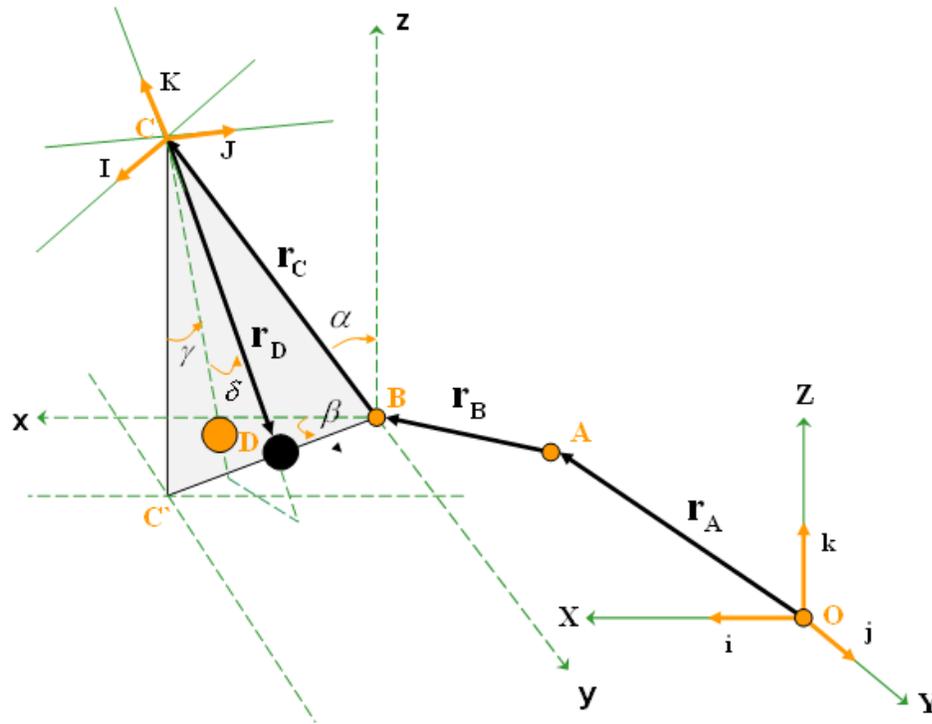


Figure 2.3: Position vectors of the cargo being transferred.

There are two forces exerted on the cargo: the tension force denoted by  $\vec{T}$  and the gravity force. Hence, the equation of motion of the cargo is

$$\vec{T} + m_{cargo} g \vec{K} = m_{cargo} \vec{a}_D \quad (2.5)$$

where  $\vec{K}$  is unit vector parallel to the  $Z$  – axis in the ground fixed reference frame and  $\vec{T}$  is equal to  $\vec{F}_{c/s}$  but in the opposite direction. Therefore,

$$-\vec{F}_{c/s} + m_{cargo} g \vec{K} = m_{cargo} \vec{a}_D \quad (2.6)$$

where

$$\vec{a}_D = \vec{a}_A + \ddot{\vec{r}}_{DA} + \dot{\vec{\omega}} \times \vec{r}_{DA} + 2\vec{\omega} \times \dot{\vec{r}}_{DA} + \vec{\omega} \times (\vec{\omega} \times \vec{r}_{DA}) \quad (2.7)$$

and

$$\vec{r}_{DA} = \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} + l_B \begin{bmatrix} \sin \alpha \cos \beta \\ \sin \alpha \sin \beta \\ \cos \alpha \end{bmatrix} - l_C \begin{bmatrix} \cos \delta \sin \gamma \cos \beta + \sin \delta \sin \beta \\ \cos \delta \sin \gamma \sin \beta - \sin \delta \cos \beta \\ \cos \delta \cos \gamma \end{bmatrix} \quad (2.8)$$

Here,  $\alpha$  is the luff angle,  $\beta$  is the slew angle,  $\gamma$  is the in-plane swing angle,  $\delta$  is the out-of-plane swing angle,  $l_B$  is the boom length, and  $l_C$  is the cable length. From equations (2.6), (2.7), and (2.8) we obtain the following  $[3 \times 1]$  vector:

$$\vec{F}_{c/s} = m_{cargo} \begin{bmatrix} g \sin \theta - \dot{u} - wq + rv - \ddot{x}_{DA} + \dot{r}y_{DA} - \dot{q}z_{DA} - 2q\dot{z}_{DA} \\ + 2r\dot{y}_{DA} + (q^2 + r^2)x_{DA} - pqy_{DA} - prz_{DA} \\ -g \cos \theta \sin \phi - \dot{v} + pw - ru - \ddot{y}_{DA} - \dot{r}x_{DA} + \dot{p}z_{DA} - 2r\dot{x}_{DA} \\ + 2p\dot{z}_{DA} + (p^2 + r^2)y_{DA} - pqx_{DA} - qrz_{DA} \\ -g \cos \theta \cos \phi - \dot{w} - pv + qu - \ddot{z}_{DA} + \dot{q}x_{DA} - \dot{p}z_{DA} - 2p\dot{y}_{DA} \\ + 2p\dot{x}_{DA} + (q^2 + p^2)z_{DA} - prx_{DA} - qry_{DA} \end{bmatrix} \quad (2.9)$$

where  $\theta$  and  $\phi$  are the angles associated with pitch and roll motion of the ship, respectively. The derivatives  $\ddot{x}_{DA}$ ,  $\ddot{y}_{DA}$ ,  $\ddot{z}_{DA}$  are obtained by differentiating equation (2.8) twice, that is,

$$\begin{aligned}
\ddot{x}_{DA} = & \ddot{\alpha} l_B \cos \alpha \cos \beta + \ddot{\beta} [l_C (\cos \delta \sin \gamma \sin \beta - \sin \delta \cos \beta) - l_B \sin \alpha \sin \beta] \\
& - \ddot{\gamma} l_C \cos \delta \cos \gamma \cos \beta + \ddot{\delta} l_C (\sin \delta \sin \gamma \cos \beta - \cos \delta \sin \beta) \\
& - \ddot{l}_C (\cos \delta \sin \gamma \cos \beta + \sin \delta \sin \beta) - \dot{\alpha}^2 l_B \sin \alpha \cos \beta \\
& - \dot{\beta}^2 [l_B \sin \alpha \cos \beta - l_C (\cos \delta \sin \gamma \cos \beta + \sin \delta \sin \beta)] \\
& + \dot{\gamma}^2 l_C \cos \delta \sin \gamma \cos \beta + \dot{\delta}^2 l_C (\cos \delta \sin \gamma \cos \beta + \sin \delta \sin \beta) \\
& + 2 \dot{l}_C [\dot{\beta} (\cos \delta \sin \gamma \sin \beta - \sin \delta \cos \beta) + \dot{\delta} (\sin \delta \sin \gamma \cos \beta - \cos \delta \sin \beta) \\
& - \dot{\gamma} \cos \delta \cos \gamma \cos \beta] + 2 \dot{\delta} \dot{\gamma} l_C \sin \delta \cos \gamma \cos \beta + 2 \dot{\beta} \dot{\gamma} l_C \cos \delta \cos \gamma \sin \beta \\
& - 2 \dot{\alpha} \dot{\beta} l_B \cos \alpha \sin \beta - 2 \dot{\delta} \dot{\beta} l_C (\sin \delta \sin \gamma \sin \beta + \cos \delta \cos \beta) \tag{2.10}
\end{aligned}$$

$$\begin{aligned}
\ddot{y}_{DA} = & \ddot{\alpha} l_B \cos \alpha \sin \beta + \ddot{\beta} [-l_C (\cos \delta \sin \gamma \cos \beta + \sin \delta \sin \beta) + l_B \sin \alpha \sin \beta] \\
& - \ddot{\gamma} l_C \cos \delta \cos \gamma \sin \beta + \ddot{\delta} l_C (\sin \delta \sin \gamma \sin \beta + \cos \delta \cos \beta) \\
& - \ddot{l}_C (\cos \delta \sin \gamma \cos \beta - \sin \delta \cos \beta) - \dot{\alpha}^2 l_B \sin \alpha \sin \beta \\
& - \dot{\beta}^2 [l_B \sin \alpha \cos \beta + l_C (\cos \delta \sin \gamma \sin \beta - \sin \delta \cos \beta)] \\
& + \dot{\gamma}^2 l_C \cos \delta \sin \gamma \sin \beta + \dot{\delta}^2 l_C (\cos \delta \sin \gamma \sin \beta - \sin \delta \cos \beta) \\
& + 2 \dot{l}_C [-\dot{\beta} (\cos \delta \sin \gamma \cos \beta + \sin \delta \sin \beta) + \dot{\delta} (\sin \delta \sin \gamma \sin \beta + \cos \delta \cos \beta) \\
& - \dot{\gamma} \cos \delta \cos \gamma \sin \beta] + 2 \dot{\delta} \dot{\gamma} l_C \sin \delta \cos \gamma \sin \beta - 2 \dot{\beta} \dot{\gamma} l_C \cos \delta \cos \gamma \cos \beta \\
& + 2 \dot{\alpha} \dot{\beta} l_B \cos \alpha \cos \beta + 2 \dot{\delta} \dot{\beta} l_C (\sin \delta \sin \gamma \cos \beta - \cos \delta \sin \beta) \tag{2.11}
\end{aligned}$$

$$\begin{aligned}
\ddot{z}_{DA} = & - \ddot{\alpha} l_B \sin \alpha + \ddot{\delta} l_C \sin \delta \cos \gamma + \ddot{\gamma} l_C \sin \delta \cos \gamma - \ddot{l}_C \cos \delta \cos \gamma \\
& - \dot{\alpha}^2 l_B \cos \alpha + (\dot{\gamma}^2 + \dot{\delta}^2) l_C \cos \delta \cos \gamma + 2 \dot{l}_C [\dot{\delta} \sin \delta \cos \gamma + \dot{\gamma} \cos \delta \sin \gamma] \\
& - 2 \dot{\delta} \dot{\gamma} l_C \sin \delta \sin \gamma \tag{2.12}
\end{aligned}$$

The moment exerted by the cargo about Point  $A$  on the ship is given by:

$$\vec{M}_{c/s} = \vec{r}_{CA} \times \vec{F}_{c/s} \tag{2.13}$$

where

$$\vec{r}_{CA} = \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} + l_B \begin{bmatrix} \sin \alpha \cos \beta \\ \sin \alpha \sin \beta \\ \cos \alpha \end{bmatrix} \quad (2.14)$$

To solve for the motion of the ship, we substitute equations (2.9) and (2.13) into the equations of motion of the ship before integration. However, before the integration could be carried out, the values of  $\delta$  and  $\gamma$  should be found, so we must derive two additional equations for  $\delta$  and  $\gamma$ . In the next section, we show how to derive these equations.

### 2.1.3 Equations Governing $\gamma$ and $\delta$

Because the tension force acts along the cable, the dot product of the tension force with any vector normal to the cable is zero. To find the unit vectors along and perpendicular to the cable, we apply the required rotations to the initial reference coordinate systems as shown in Figure 2.4. Hence, the rotation matrix is

$$\begin{bmatrix} \vec{I} \\ \vec{J} \\ \vec{K} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & \cos \delta & \sin \delta \\ 0 & -\sin \delta & \cos \delta \end{bmatrix} \begin{bmatrix} \cos \gamma & 0 & -\sin \gamma \\ 0 & 1 & 0 \\ \sin \gamma & 0 & \cos \gamma \end{bmatrix} \begin{bmatrix} \cos \beta & \sin \beta & 0 \\ -\sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \vec{i} \\ \vec{j} \\ \vec{k} \end{bmatrix} \quad (2.15)$$

and

$$\vec{T} \cdot \vec{I} = 0 \quad (2.16)$$

$$\vec{T} \cdot \vec{J} = 0 \quad (2.17)$$

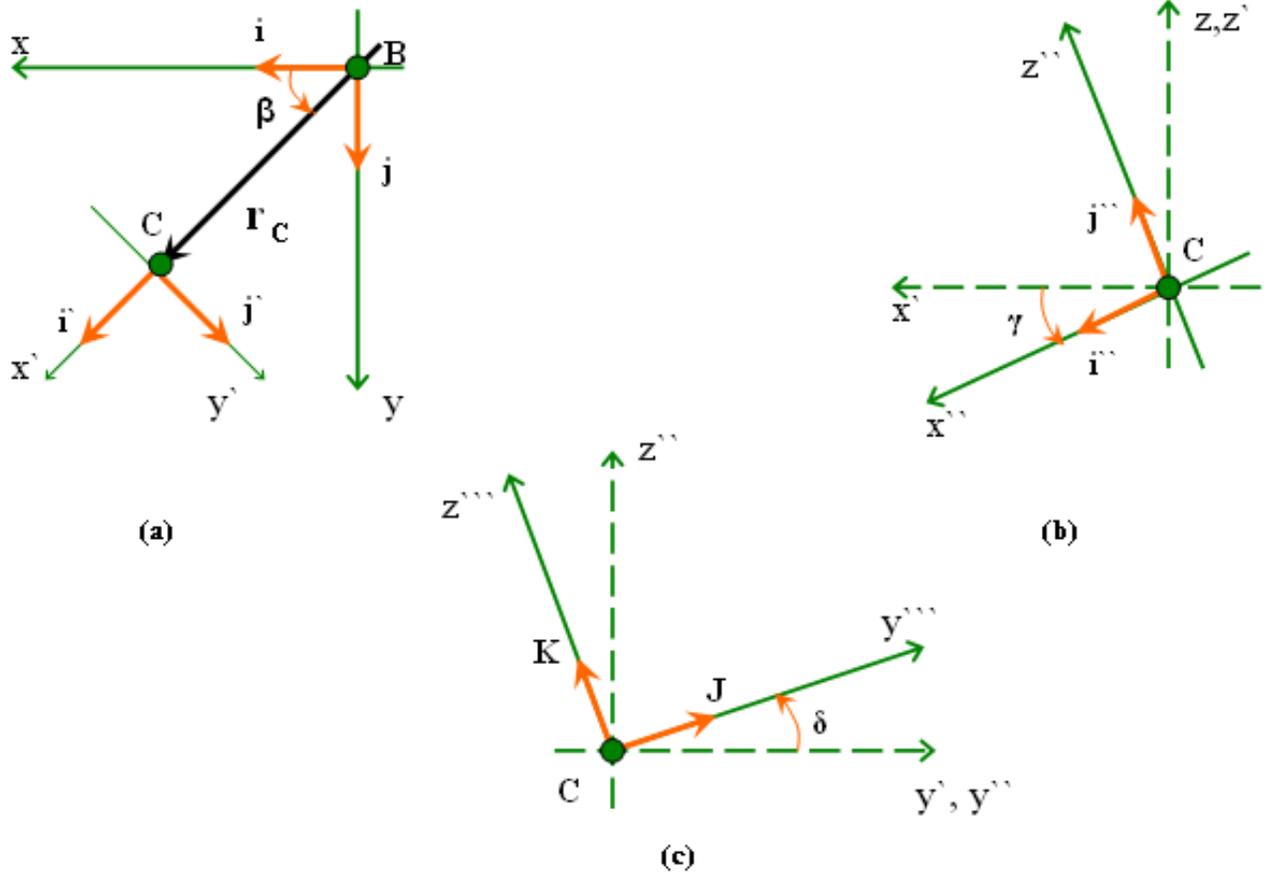


Figure 2.4: (a) Top view of the boom. The reference axis is rotated by an angle  $\beta$  around the positive  $z - axis$ . (b) The double-prime coordinate is formed by a rotation of an angle  $\gamma$  around the primed  $y - axis$ . (c) The triple prime coordinate, which lies on the cable, is formed by a rotation of an angle  $\delta$  around the double primed  $x - axis$ .

which result in the following two equations:

$$-l_C \cos \delta \ddot{\gamma} + G_x \cos \beta \cos \gamma + G_y \sin \beta \cos \gamma - G_z \cos \gamma = 0 \quad (2.18)$$

$$\begin{aligned} -l_C \ddot{\delta} + G_x (\sin \delta \sin \gamma \cos \beta - \cos \delta \sin \beta) + G_y (\sin \delta \sin \gamma \sin \beta + \cos \delta \cos \beta) \\ + G_z \sin \delta \cos \gamma = 0 \end{aligned} \quad (2.19)$$

where

$$G_x = F_x - [g \sin \theta - \dot{u} - wq + rv - \ddot{x}_{DA} + \dot{r}y_{DA} - \dot{q}z_{DA} - 2q\dot{z}_{DA} \\ + 2r\dot{y}_{DA} + (q^2 + r^2)x_{DA} - pqy_{DA} - prz_{DA}] \quad (2.20)$$

$$G_y = F_y - [-g \cos \theta \sin \phi - \dot{v} + pw - ru - \ddot{y}_{DA} - \dot{r}x_{DA} + \dot{p}z_{DA} - 2r\dot{x}_{DA} \\ + 2p\dot{z}_{DA} + (p^2 + r^2)y_{DA} - pqx_{DA} - qrz_{DA}] \quad (2.21)$$

$$G_z = F_z - [-g \cos \theta \cos \phi - \dot{w} - pv + qu - \ddot{z}_{DA} + \dot{q}x_{DA} - \dot{p}z_{DA} - 2p\dot{y}_{DA} \\ + 2p\dot{x}_{DA} + (q^2 + p^2)z_{DA} - prx_{DA} - qry_{DA}] \quad (2.22)$$

and

$$F_x = \ddot{\alpha} l_B \cos \alpha \cos \beta + \ddot{\beta} [l_C (\cos \delta \sin \gamma \sin \beta - \sin \delta \cos \beta) \\ - l_B \sin \alpha \sin \beta] - \ddot{l}_C (\cos \delta \sin \gamma \cos \beta + \sin \delta \sin \beta) \\ - \dot{\alpha}^2 l_B \sin \alpha \cos \beta \\ - \dot{\beta}^2 [l_B \sin \alpha \cos \beta - l_C (\cos \delta \sin \gamma \cos \beta + \sin \delta \sin \beta)] \\ + \dot{\gamma}^2 l_C \cos \delta \sin \gamma \cos \beta + \dot{\delta}^2 l_C (\cos \delta \sin \gamma \cos \beta + \sin \delta \sin \beta) \\ + 2\dot{l}_C [\dot{\beta} (\cos \delta \sin \gamma \sin \beta - \sin \delta \cos \beta) \\ + \dot{\delta} (\sin \delta \sin \gamma \cos \beta - \cos \delta \sin \beta) - \dot{\gamma} \cos \delta \cos \gamma \cos \beta] \\ + 2\dot{\delta} \dot{\gamma} l_C \sin \delta \cos \gamma \cos \beta + 2\dot{\beta} \dot{\gamma} l_C \cos \delta \cos \gamma \sin \beta \\ - 2\dot{\alpha} \dot{\beta} l_B \cos \alpha \sin \beta - 2\dot{\delta} \dot{\beta} l_C (\sin \delta \sin \gamma \sin \beta + \cos \delta \cos \beta) \quad (2.23)$$

$$\begin{aligned}
F_y = & \ddot{\alpha} l_B \cos \alpha \sin \beta + \ddot{\beta} [-l_C (\cos \delta \sin \gamma \cos \beta + \sin \delta \sin \beta) \\
& + l_B \sin \alpha \sin \beta] - \ddot{l}_C (\cos \delta \sin \gamma \cos \beta - \sin \delta \cos \beta) \\
& - \dot{\alpha}^2 l_B \sin \alpha \sin \beta \\
& - \dot{\beta}^2 [l_B \sin \alpha \cos \beta + l_C (\cos \delta \sin \gamma \sin \beta - \sin \delta \cos \beta)] \\
& + \dot{\gamma}^2 l_C \cos \delta \sin \gamma \sin \beta + \dot{\delta}^2 l_C (\cos \delta \sin \gamma \sin \beta - \sin \delta \cos \beta) \\
& + 2 \dot{l}_C [-\dot{\beta} (\cos \delta \sin \gamma \cos \beta + \sin \delta \sin \beta) \\
& + \dot{\delta} (\sin \delta \sin \gamma \sin \beta + \cos \delta \cos \beta) - \dot{\gamma} \cos \delta \cos \gamma \sin \beta] \\
& + 2 \dot{\delta} \dot{\gamma} l_C \sin \delta \cos \gamma \sin \beta - 2 \dot{\beta} \dot{\gamma} l_C \cos \delta \cos \gamma \cos \beta \\
& + 2 \dot{\alpha} \dot{\beta} l_B \cos \alpha \cos \beta + 2 \dot{\delta} \dot{\beta} l_C (\sin \delta \sin \gamma \cos \beta - \cos \delta \sin \beta) \tag{2.24}
\end{aligned}$$

$$\begin{aligned}
F_z = & \ddot{\alpha} l_B \sin \alpha - \ddot{l}_C \cos \delta \cos \gamma - \dot{\alpha}^2 l_B \cos \alpha + (\dot{\gamma}^2 + \dot{\delta}^2) l_C \cos \delta \cos \gamma \\
& + 2 \dot{l}_C [\dot{\delta} \sin \delta \cos \gamma + \dot{\gamma} \cos \delta \sin \gamma] - 2 \dot{\delta} \dot{\gamma} l_C \sin \delta \sin \gamma \tag{2.25}
\end{aligned}$$

## 2.2 Computational Technique

The equations of motion of the ship are solved using LAMP. It is a three-dimensional time-domain solver, which predicts simultaneously and interactively the motion of and the flow around a ship advancing in moderate and severe sea conditions. It provides solutions in the time-domain and is not restricted to periodic motions. It can handle time varying forward speeds, large-amplitude incident waves, and large-amplitude ship motions. LAMP has varying degrees of complexity, including linear, weakly nonlinear, and fully nonlinear options. At the highest level, the free-surface boundary conditions are linearized about the actual incident-wave surface, and the body boundary conditions are satisfied on the instantaneous wetted surface. The wave amplitude may be of the same order as the draft of the ship, but must be an order of magnitude less than the wavelength.

LAMP does not run in real-time in our simulator, instead, the simulator is developed by running LAMP using the SGI Origin 2000 as a preprocessor to determine the ship response to

many environmental conditions. The positions, velocities, accelerations, and sea conditions are stored as data files, which can run in the simulation.

In our simulation, the crane-ship is assumed to be a rigid body floating on the free wavy surface and undergoing arbitrary 6-DOF motion. Surface tension is not taken into consideration and the water depth is assumed infinite. The exact boundary conditions are applied to the submerged portion of the hull surface and the free surface condition is linearized around the incoming wave surface. This approximation can be justified in principle for small wave slopes and slender bodies. The hydrostatic and hydrodynamic forces are calculated, and then the rigid-body equations of motion are integrated using a fourth-order Runge-Kutta method.

To solve the equations of motion of the payload, we integrate the crane-load dynamic equations simultaneously with the ship-dynamic equations. For each time step, equations (2.18) and (2.19) are integrated to obtain new in-plane and out-of-plane angles. Then, the terms  $F_{c/s}$  and  $M_{c/s}$  are calculated using equations (2.9) and (2.13). These terms are then used in the ship equations of motion, equations (2.3) and (2.4).

### 2.2.1 Cargo Pendulation Control

To reduce cargo pendulation, Henry, Masoud, Nayfeh, and Mook<sup>15,16</sup> developed a delayed-position feedback control. This system reduces pendulations of hoisted cargo on ship-mounted cranes. The system is so adaptable that it requires neither major modifications to the current boom structure nor special training for crane operators. The mathematical model includes both geometric and kinetic nonlinearities. The stability of the controlled mathematical model was analyzed and the controller was then applied to a boom crane via standard luff and slew crane actuators.

The delayed-position feedback controller produces damping in the system by forcing the suspension point of the payload to track inertial reference coordinates. A tracking controller is used to ensure proper tracking of these inertial reference coordinates.

Their control concept applies to all types of cranes that use a cable for the purpose of

hoisting and transferring cargo. They applied their control concept to a spherical pendulum model of the payload hoisting cable assembly by actuating the suspension point of the hoisting cable in the  $x$ - and  $y$ -directions. The horizontal motion of the payload relative to the suspension point of the hoisting cable can be measured using several techniques, including accelerometers and inertial encoders that measures angles of the payload hoisting cable.

We used the mathematical model of the delayed position-feedback control of Henry, Masoud, Nayfeh, and Mook by treating the ship and the crane as a rigid body. To simulate the dynamics of the spherical pendulum under most critical conditions, we considered a ship whose pitch frequency is twice its roll frequency and excited it with a regular heading wave whose frequency is equal to the pitch frequency. Then we solved the equations of motion for the cargo using a fourth-order Runge-Kutta subroutine.

# Chapter 3

## Modeling and Simulation

The steps necessary for building a full CAVE simulation are shown in Figure 3.1. Every block in this diagram represents an important step in developing the CAVE simulation. A complete and extensive explanation of each step is given in this chapter.

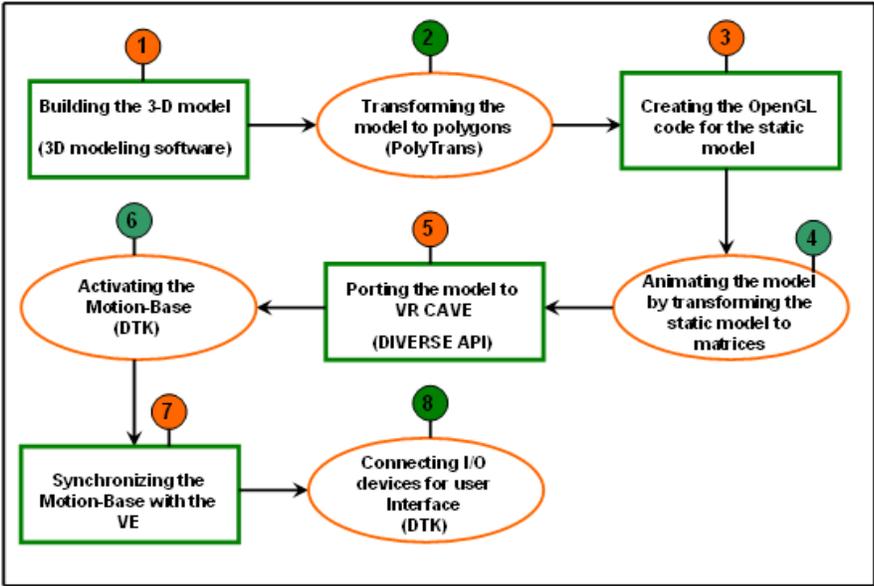


Figure 3.1: A block diagram showing the various steps necessary for a complete CAVE simulation

### 3.1 Building the Model

The first step in any VR simulation is to build the 3-D model. One has to take into consideration the size, complexity, reality, and compatibility of the model that will be projected in the virtual environment, every small deficiency that may not appear on the desktop screen will be large and noticeable in the immersive environment.

The model could be first built using any of the 3-D modeling softwares, such as *3DStudio*, *3DMax*, *LightWave3D*, or any CAD software, in our case *LightWave3D* was used to build the model, Figure 3.2.



Figure 3.2: *LightWave3D* ship model.

Since CAVE APIs use C or C++ as programming languages, the model should be transformed into one of these programming languages. This is achieved in our case using *PolyTrans*. *PolyTrans* (Polygon Transformer) is a software that converts every aspect of a 3-D model into polygons, as shown in Figure 3.3, including all shading parameters, texture mapping coordinates, texture mapping information, and (for selective converters) animation data.

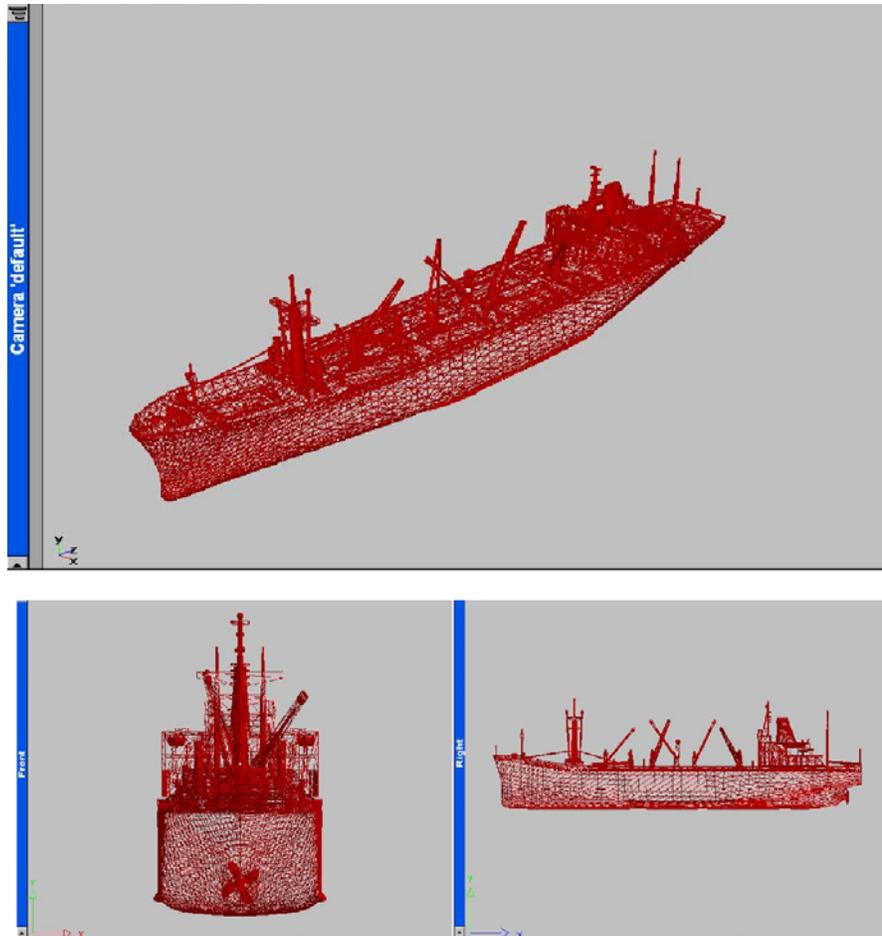


Figure 3.3: Different ship views showing the ship transformed into polygons by PolyTrans.

PolyTrans transforms the model into polygons, which are the basic structure of the Graphical Library of C++ known as *OpenGL*. OpenGL is the C++ graphical library and is one of the most widely supported 2-D and 3-D graphics API. It is well known for its stability, fast rendering, diverse functionality, and most importantly its applicability for any workstation platform.

OpenGL consists of four main libraries:

1. GL: OpenGL main library. It is used to draw lines, vertices, and polygons, to specify colors, lights, and materials, to scale, translate, and rotate polygons, in addition to

many other basic functions.

2. GLU: OpenGL Utility library. This is a set of functions used to create texture mipmaps from a base image, map coordinates between screen and object space, and draw quadric surfaces.
3. GLX: It is used on Unix OpenGL implementations to manage interaction with the X Window System and to encode OpenGL onto the X protocol stream for remote rendering.
4. GLUT: OpenGL Utility Toolkit. This library was developed to enhance the OpenGL capabilities and to get rid of the OpenGL Auxiliary Library (GLAUX), thus making the program faster and more compact. It also works on all workstations.

All the above-mentioned libraries were used in building our model.

The different libraries of OpenGL read the polygons stored in the matrices provided by PolyTrans, combine them to form the desired model shape. The resulting matrices contain the vertices, normals, texture mapping, lighting and many other important data necessary for showing the model in the desired shape. As the model is animated, these matrices are mathematically transformed into new ones by applying rotational and translational transformations.

Figure 3.4 shows a view of the crane ship model on a Linux desktop. The figure shows the texture mapped onto both of the ship and the sea. Since the sea waves are created with variable amplitudes and frequencies, a user of the simulation can specify the amplitude and frequency of the waves that will excite the ship. As a result, one can virtually distinguish different seastates through the virtual scene.

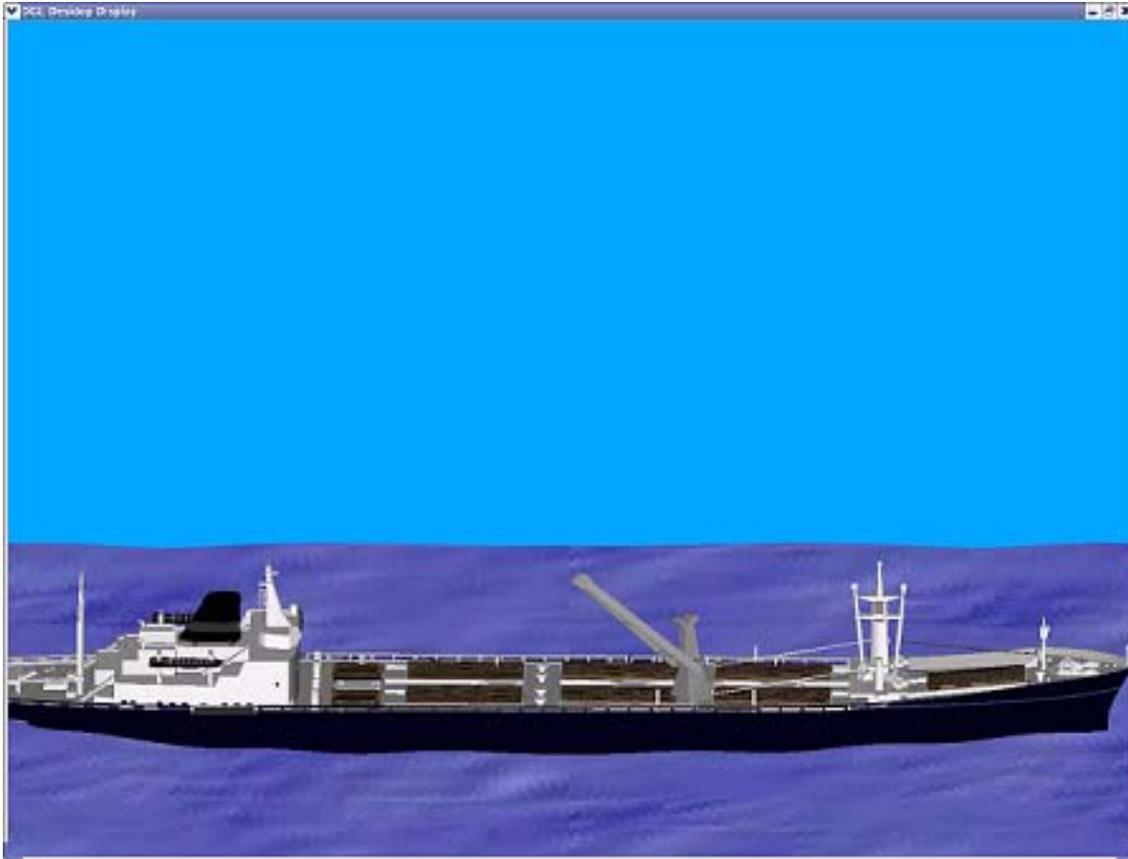


Figure 3.4: Side view of the ship-mounted crane on a desktop screen.

## 3.2 Model Animation

Since the main purpose of our simulation is to study the dynamics of ships and ship-mounted cranes, model animation is the most important part of this effort. To apply the dynamics to our model, we used the GL library, which provides functions for rotations and translations in OpenGL.

To generate the data necessary to animate the ship and cargo for both the uncontrolled and controlled scenarios, we solved the equations of motion using the techniques mentioned in Chapter 2. In order to read the data from the output files, the files must be opened and read line by line in the main body of the C++ code. After each line is read, the OpenGL display

function is called upon to start a new frame, a high frame rate will cause the animation of the model. One of the problems that appear is the application of the transformation matrices to a set of polygons. This transformation causes a coordinate rotation of all the polygons in a set when the rotation is applied to any of its members. In order to solve this problem, we used the Push-Pop Matrix function in the GL library, which serves as a matrix extractor. Any line of code that is written between `glPushMatrix` and the `glPopMatrix` is treated as one entity and any transformation applied inside the Push-Pop matrix will only be applied to that entity. Moreover, the coordinate transformations that are applied to that entity will be reversed back to the initial coordinate system for any other set of polygons.

We treated the ship as a single rigid body and applied the translations and rotations to its body coordinates using the GL functions: *glTranslatef* (Surge, Lateral, Heave), *glRotatef* (1.0, Roll, Pitch, Yaw). We also treated the crane as another matrix entity, which is rotated with the slew angle  $\beta$  and luff angle  $\alpha$ . The crane was also subjected to base excitations caused by the ship's translations and rotations. The cable and cargo were extracted in one matrix to which in-plane and out-of-plane rotations of  $\gamma$  and  $\delta$  were applied, respectively. This creates the whole motion of the ship, crane, cable and cargo according to the data provided by the equations of motion.

The advantage of using OpenGL over any other software is that one can easily change any specifications of the model without going back to the initial model built using the 3-D modeler. For example, in our simulation, the user can specify the position, length, luff and slew angles of the crane at the beginning of the simulation without the need to go back and change these values in the initial model.

### 3.3 Porting the Model to the CAVE

To port our model to the CAVE, we had to thoroughly study and understand the types of APIs used to develop VR models in an immersive stereoscopic environment. The first and most used API is CAVELibs created by the National Center for Supercomputing Applications

(NCSA) at the University of Illinois at Urbana-Champaign. Their CAVE libraries are based on OpenGL. In our opinion, CAVELibs are easy to use, but lack good interface with non-graphical applications, such as the motion platform. On the other hand the VT CAVE research group developed a collection of software packages called DIVERSE.<sup>17</sup> DIVERSE is a collection of software packages that connect end-user programs with any other C++ API to integrate simulations with any virtual environment. Although DIVERSE DGL lacks good documentation, we found it better than CAVELibs because it has a better user interface and it is able to detach graphical packages from non-graphical ones. The DIVERSE team (Kelso, Arsenault, and Kriz) developed the following three packages:

1. **DTK: Diverse ToolKit.** This package performs all non-graphical tasks, networking, and hardware services, such as starting the hand wand and head tracker, creating shared memory, or even networking with the motion platform.
2. **DgiPf: Diverse graphics interface to Performer.** It is built using classes containing methods and data. It works as the OpenGL Performer, can load Dynamically Shared Objects (DSO's), and can run on the CAVE, the head mounted display, or even the desktop or laptop.
3. **DGL: Diverse OpenGL.** It is a new release of DIVERSE, which supports scene rendering using exactly the same OpenGL functions, but it differs from the DgiPf in that it does not have a scenegraph. The scenegraph is a hierarchical structure for building a virtual model, It consists of blocks and nodes and is more organized and much easier for applying different transformations.

To port our model to the CAVE using DIVERSE, we had to integrate DIVERSE's non-graphical package DTK with either the DgiPf or the DGL APIs. Since DGL is more compatible with OpenGL, we decided to use DGL integrated with DTK.

The first step in porting the system is to know the DGL coordinate configuration. The configuration is a normalized right-handed coordinate system whose origin is at the center of the world as shown in Figure 3.5.

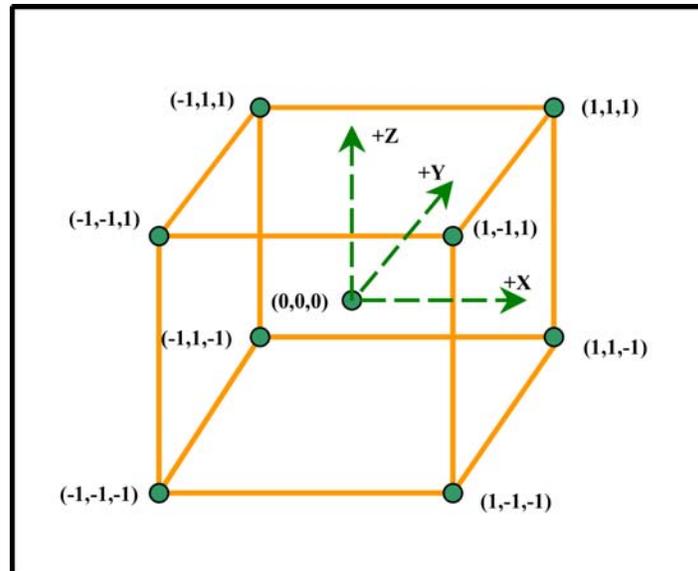


Figure 3.5: The DGL coordinate cube.

To show our model within the CAVE space, we had to study the views, projections, and frustums of the DGL. Converting the OpenGL model to DGL needed small changes in the OpenGL code.

Using the Fast Light ToolKit (FLTK), we created a Graphical User Interface (GUI) to switch from one scenario to another, or from one view to another without a need to stop and start the program. Each time the user decides to try a new scenario, he clicks the buttons that hold the desired scenario and views names as shown in Figure 3.6. In our opinion, this makes it easier for the trainer to compare the results of different scenarios.

To interact with the VR scene, we used DTK. The problem we encountered was that the IRIX machine which runs the CAVE, does not have any USB plugs. This makes it impossible to connect the machine directly to any USB I/O device, such as the joysticks used to rotate the crane or reel the cable. We connected the two joysticks (one for the crane rotation and the other for cable reeling) to another Linux laptop that has USB plugs, then the Linux laptop was connected through the network to the IRIX machine, which runs the CAVE.

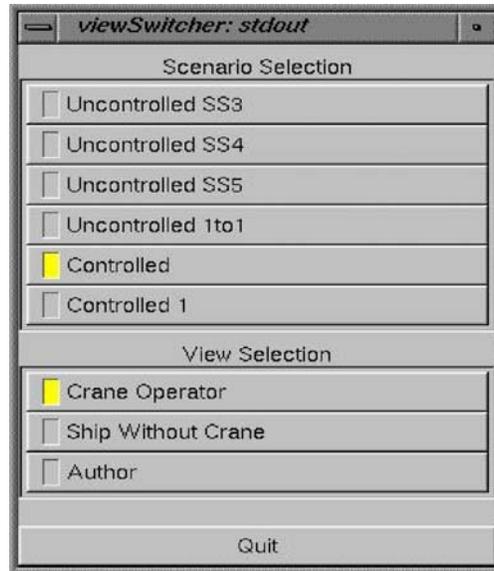


Figure 3.6: A GUI used to switch from one scenario to the other during the CAVE simulation.

The DTK server is run on both machines and the joysticks' data are stored in the shared memory of the Linux laptop then transferred through the network to the IRIX machine and stored in its shared memory. Afterwards, the data is read from the shared memory by the DGL main code and then transformed into translations and rotations in the virtual scene. Figure 3.7 shows the process. It is also worth mentioning that the data format provided by the joysticks through the network does not match the data format on the IRIX machine, to solve this problem we used a DTK function, which sets the data in the right, and readable format for animations.

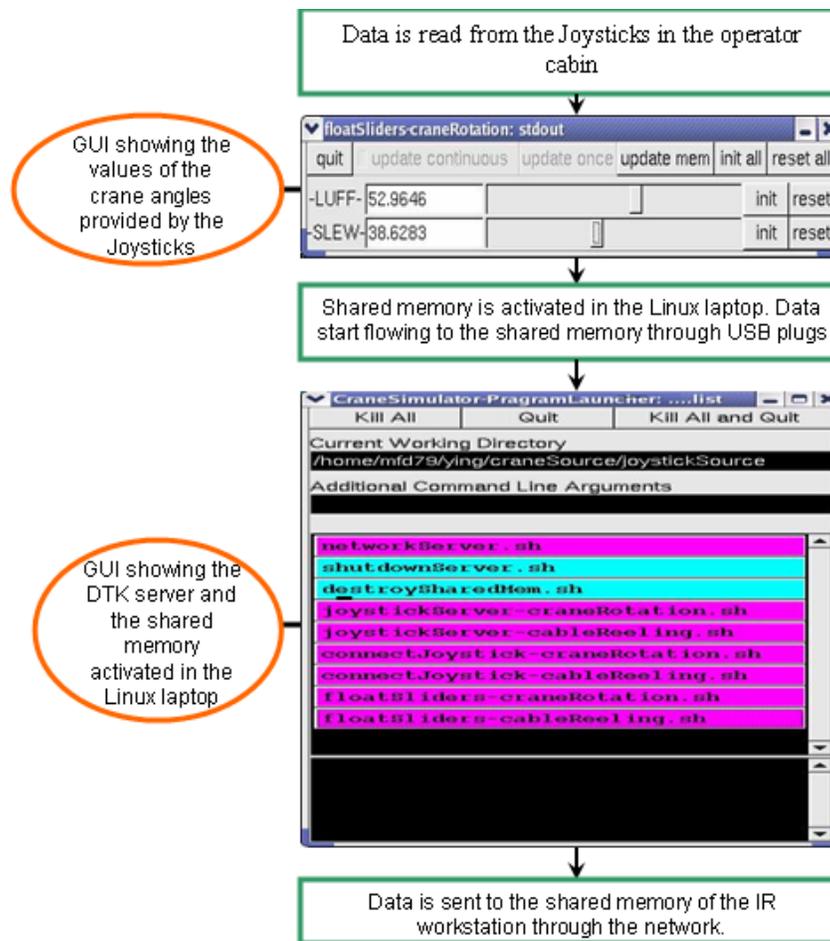


Figure 3.7: A block diagram showing data flow from the I/O devices to the graphical virtual simulation in the CAVE.

### 3.4 Motion Platform

We note that the eyes gather the information needed by the brain to recognize velocity and that the ears gather the information needed to recognize acceleration. If the eyes and the inner ears send mismatched information, then confusion occurs, which might result in an unnatural motion sickness. Some simulations are only capable of providing mismatched information. For example, in some simulation where the floor is fixed, the view from the

bridge correctly portrays the motion of the ship relative to the sea. The image tricks the eye by showing movement but the stationary ear is not tricked in concert with the eye. In this example, the viewer is not accelerating, resulting in an apparent mismatch. This experience for the viewer can be very disturbing as already described. For this reason, simulators that use moving platforms have long been in use for training aircraft pilots among others. To simulate the actual motion of the bridge of the ship and to avoid the confusion that may result from a fixed floor in the simulation, we installed a MOOG 6-DOF 2000E, Model 170E122A motion base in the floor of the CAVE, as the one shown in Figure 3.8.



Figure 3.8: 6-DOF 200E, Model 170E122A motion base.<sup>18</sup>

This 6-DOF motion base provides sufficiently fast response and large accelerations to simulate the significant motions of a ship. The motion base has a built in computer, which is serially connected to the system controller. The serial interface provides commands to all six actuators in two forms, either in the actuator mode in which measurements are made with respect to the initial lengths of the actuators or in terms of its six-degrees-of-freedom (roll, pitch, yaw, surge, lateral, heave). Because our data is already expressed in terms of

(roll, pitch, yaw, surge, lateral, heave), we decided to use the second mode.

The data needs to be scaled to match the maximum values of the 6-DOF of the motion base. The actual motion base capabilities are listed in Table 3.1.

Table 3.1: Maximum capabilities of the motion base.

DOF	MIN	MAX
Roll(Degrees)	-29	+29
Pitch(Degrees)	-33	+33
Yaw(Degrees)	-29	+29
Surge(m)	0.381	-0.381
Lateral(m)	0.381	-0.381
Heave(m)	0.0	0.4752

To activate the motion base, we use DTK, which in this case provides the shared memory in which the data is written, as show in Figure 3.9. As the motion base is activated and connected, its computer starts reading data from the shared memory through the network, and it starts moving the platform according to the data set provided by the scenario text files. The most important thing we had to take into consideration is that the data should be provided at a rate of 60Hz because lower rates induce noticeable vibrations of the base, and higher rates do not allow sufficient time for the motion base to respond.

To synchronize the motion base with the CAVE animation, data are sent to the shared memory of the motion base in the main loop of the DGL code where the display function is called. A timer is initiated to control feeding the data to the shared memory so that every time a new data set is sent to the shared memory, a new frame is activated in the animations, and the motion is synchronized in real time.

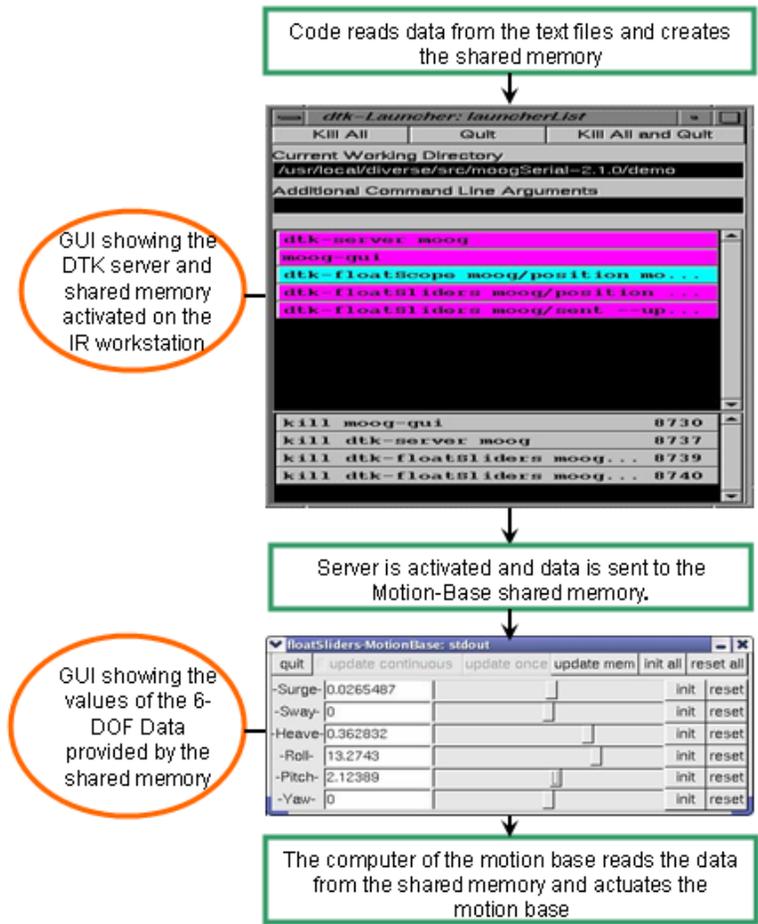


Figure 3.9: A block diagram showing the data flow from the text files to the motion base.

# Chapter 4

## Results of CAVE Simulation

### 4.1 Response of Ships to Dynamic Sea Environment

Most of the virtual simulations that had been done in the past serve one or two purposes at most. Our simulator is a multipurpose one that can be used to serve multiple engineering applications in the area of ships and ship-mounted cranes. As stated before, our simulator can be used to study the response of ships with and without deficiencies to dynamic sea environments. It can also be used to simulate the dynamic response of war ships to vibrations resulting from missile launching or to simulate safe landing of planes on aircraft carriers. All what one has to do is to derive the equations of motion for any of the above scenarios, study the geometry of the ship, obtain the data and feed them into the virtual scene and the motion base. As a result, a full simulation of the desired scenario will be obtained.

Because we constructed the dynamical model for the T-ACS 4-6 ship and solved the equations of motion of this special type of ship, we simulated its response to different seastates. Our simulation showed a great deal of realism and was carried out in real time.

Figure 4.1 shows the T-ACS ship projected in the CAVE 3-D environment, The ship in this scenario is used to simulate the response of ships to certain sea environments.

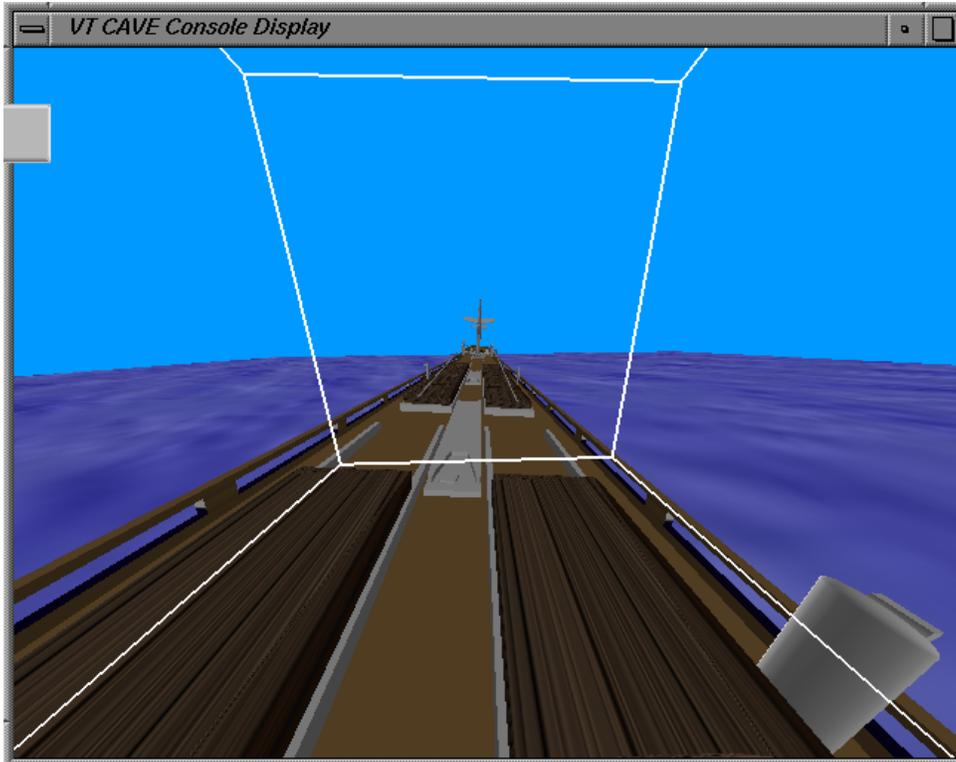


Figure 4.1: The T-ACS ship model as part of a simulation to study the response of ships to dynamic sea environments.

## 4.2 Animation and Visualization of Ship and Crane Systems in a Dynamic Sea Environment

Ship-mounted cranes are another important application for our simulation. A crane is added to the T-ACS ship. This crane is placed at different positions on the ship to simulate different dynamical responses to different sea environments. Moreover, the luff and slew degrees of freedom of the crane can be activated and the cable can be reeled and unreeled. Hence, different simulations for different seastates, crane positions, and crane maneuvers can be carried out.

Figure 4.2 shows the crane positioned at  $(-30.2, 7.6, -3)$  with a boom length of 36.9 m, a

slew angle of 90.0 degrees, and a luff angle of 30 degrees. The cable length is 28.9 m and the simulation is carried out at seastate 3.



Figure 4.2: The T-ACS crane-ship as part of a simulation that was carried out in the CAVE at sea state 3.

### 4.3 Control of Moving Loads Aboard a Ship

Cargo pendulations control is another important feature that can be visualized in our simulator. The simulation demonstrates the great abilities of the nonlinear delayed-position feedback control to suppress cargo pendulations even in artificially created sea environments that may excite the cargo at resonance. Comparison between the controlled and uncontrolled simulations for the same sea conditions showed a great reduction in both of the in-plane and



Figure 4.3: The crane operator cabin as part of a crane operator training. The nonlinear-delayed position feedback control is applied to the crane.

out-of-plane sway angles in the controlled simulations.

In Figure 4.3, the ship pitch frequency is twice its roll frequency and it is excited with an artificial heading wave whose frequency is nearly equal to its pitch frequency. This combination of two-to-one internal resonance and primary response of the ship causes very large excitations of the cargo motion. The controller was able to significantly reduce cargo pendulation.

## 4.4 Crane Operator Training

Our simulator also can serve as a platform to train crane operators, especially at high seastates. This platform will reduce the risks and expenses that are encountered with a real life training situation. The platform can also provide extra scenarios that the crane operator may not face in his actual training period, thereby enhancing his capabilities and providing him with valuable experience.

Besides the real VR scene in which the crane operator sits in and with which he interacts, we used the motion platform to simulate the actual motion of the ship, which is synchronized in real time with the virtual scene of the ship. Hence, the operator does not sense relative motions between the virtual scene of the ship and the moving chair and at the same time the operator can clearly sense his relative motion to the sea. This feeling is very important because sometimes an operator can feel sea sickness in one scenario but not in the other, this may affect his capabilities in maneuvering the crane.

The crane operator sits in a chair moved by the platform and controls the virtual scene using two joysticks: one for actuating the crane luff and slew motors and the other for cable reeling. As the operator moves one of the joysticks in a certain direction, he instantaneously senses the changes in the virtual scene.

We implemented scenarios at seastates 3, 4, and 5, in addition to an artificial one-to-one and two-to-one resonance cases. Some of the results of these scenarios are shown next.

Figure 4.4 shows a crane-operator virtual cabin, it shows the view an operator would see in a real cabin: part of the crane, the cargo, part of the ship, and part of the sea. It also shows part of the operator chair and joysticks. This case was carried out at seastate 3. At this seastate if the wave spectrum does not contain a component near the natural frequency of the payload, the in-plane pendulations will not be large, the out-of-plane pendulations will be very small, and the interaction between them will be small.

Figure 4.5 shows the operator cabin as part of a scenario carried out at seastate 5, it clearly shows that cargo transfer at such a seastate is not possible. The cargo motion is chaotic with very large in-plane and out-of-plane oscillations. The cargo may hit the operator

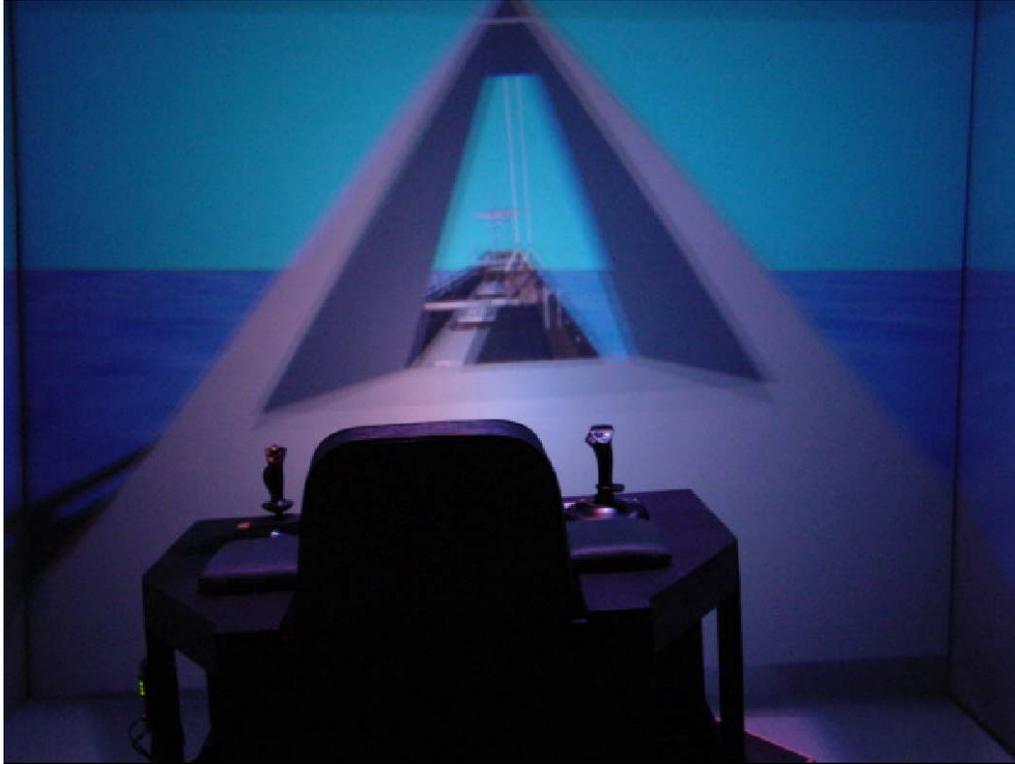


Figure 4.4: The crane operator cabin as used in crane-operator training at seastate 3.

cabin itself at some point. A similar scenario was carried out at seastate 4 and the results also show a large chaotic motion for the cargo.

We also created some artificial resonance conditions, which may cause very critical cargo responses. For example, we considered a ship possessing a two-to-one internal resonance between its pitch and roll modes and excited it with a heading wave whose frequency is near the ship pitch frequency. Figure 4.6 shows the cargo response. The in-plane and out-of-plane oscillations are very large and have approximately the same value.

This VR simulator can also serve as a platform to simulate many other different kinds of scenarios, especially those encountered with interactions between the crane operator and the virtual scene, such as crane maneuvering and cable reeling. Because a real-time solution of the equations of motion for the payload and the ship is not available, we are not able to



Figure 4.5: The crane operator cabin used in a crane operator training at seastate 5.

simulate such cases interactively. We are now in the process of developing a real-time solver to the platform so that we can perform such scenarios in the near future.



Figure 4.6: The crane operator cabin used in a crane operator training. A heading wave is used to cause a two-to-one ship internal resonance.

# Chapter 5

## Conclusion and Future Considerations

Our simulator serves as a multipurpose environment for many engineering applications in the area of ships and ship-mounted cranes. It provides a solid base and a good example of how engineering theories can be transformed into VR applications. This provides a tool to prove and make them interesting as well as give better insight into engineering problems and their applications in real life.

The simulator is used to study the response of different ships to dynamic sea environments. A deep visual understanding of such responses is not possible experimentally because either the expense is too high or the experimental models are scaled to an extent where testing of these real scenarios may not give enough insight into the problem. As a result, and in order to adequately understand such models, a researcher needs to exert a greater effort to be more experienced both theoretically and experimentally.

We also used the simulator as a platform to study the dynamics of cargo ships and the interaction between the cargo and the ship, especially in high seastates. This is very important because the pendulations may increase to an extent where the cargo transfer is very dangerous and sometimes impossible. Our simulator is capable of creating such scenarios so that one can avoid certain cargo transfer types in such seastates.

A comparison between a controlled and uncontrolled cargo-transfer scenario is also possible in our simulator. In fact, we created such a scenario for a nonlinear delayed-position

feedback control system. The simulation showed that a great reduction in the pendulations can be achieved in high seastates.

The simulator has a synchronized motion base to simulate real motions of ships. As a result, it can be used as a platform to train crane operators in VR. This VR training is much better than actual training, which costs much more and does not give enough experience for the trainer who might not encounter different scenarios in his training period, especially those in high seastates.

In order for this simulation to reach its current condition, we ran into many problems, some of which were solvable and some others were not. One of these problems is the lack of documentation. VR technology is still a new technology, and there are not many sources of good documentation on how to port a model into a CAVE. We had and still have some problems in using the new version of the DGL because it is not documented yet. To port our model to the CAVE, we had to study the small number of available examples that run using DGL, thus we recommend using DgiPf or DPf since they are much more mature and better documented as C++ APIs. Another problem is rendering; the UNIX station at the Virginia Tech CAVE does not render more than 30 thousand polygons efficiently, whereas our model contained initially about 200 thousand polygons. As a result, we had to remove many parts of the model to reduce the number of polygons, which affected the reality of our virtual model.

The sea model in our simulation consists of moving waves that have variable frequencies and amplitudes. Unfortunately we had to freeze the motion of these waves and simulate the sea with static waves only. Such motion needs a great amount of calculations for the matrix of the texture-mapped pixels. OpenGL needs to calculate the new position of each polygon for each frame and needs to alter the normals to calculate the new lighting reflections. Such calculations would consume the memory and slow down the simulation.

Because running LAMP interactively with the crane equations requires intense CPU time, we are unable at the present time to generate the data in real time using any existing workstation. As a result, we are now in the process of connecting our simulator to a

CLUSTER of PCs. A CLUSTER is a group of computers that are connected in parallel using an Ethernet or a fast Ethernet network to perform very complicated computational tasks, which enable running LAMP in real time, thus making the simulation interactive. Any changes in the conditions of the simulation will result in changes in the initial conditions of the equations, which will be solved in real time to obtain a new response that will be fed to both of the VR scene and the motion platform.

Because our code is well structured to send and receive data from any I/O device or even a CLUSTER, once the CLUSTER is connected to the CAVE, it will be easy to integrate our code with the resulting computational data. This will enhance the capabilities of our simulator and make it interactive, especially for crane-operator training. Instead of using some play back files of already generated data, we will be able to run the simulator for any scenario interactively.

Finally, we hope that our approach will create a greater interest in VR simulations of engineering applications. We believe that VR will no more be limited to the work of computer scientists. This tool will be of great value to scientists, engineers and educators.

# Bibliography

- [1] Brill, L., “Metaphors for the Traveling Cybernaut,” *Virtual Reality World*, Grindelwald, Vol. 1, No. 1, 1993, pp. q-s.
- [2] Tamura, Y., Kageyama, A., Sato, T., Fujiwara, S., and Nakamura, H., “Virtual Reality System to Visualize and Auralize Numerical Simulation Data,” *Computer Physics Communications*, Vol. 142, 2001, pp. 227-230.
- [3] Cramer, J., Kearney, J., and Papelis, Y., “Driving Simulation: Challenges for VR Technology,” *IEEE Computer Graphics and Applications*, Vol. 16, No. 12, 2000, pp. 1966-1984.
- [4] Anon, “VR in Industrial Training,” *Virtual Reality*, Grindelwald, Vol. 5, No. 4, pp. 31-33.
- [5] Freund, E., Rossman, J., and Thorsten, H., “Virtual Reality Technologies for the Realistic Simulation of Excavators and Construction Machines: From VR-Training Simulators to Telepresence Systems,” in *Proceedings of SPIE - The International Society for Optical Engineering*, 2001, pp. 358-367.
- [6] Adams, N., “A Study of the Effectiveness of Using Virtual Reality to Orient Line Workers in a Manufacturing Environment,” *Master’s Thesis*, Depaul University, Chicago, IL, 1996.
- [7] Anon, “Immersive Virtual Reality Tests Best,” *CyberEdge Journal*, Vol. 4, No. 6, 1994.

- [8] Mastaglio, T., and Callahan, R., "A Large Scale Complex Virtual Environment for Team Training," *Computer*, Vol. 28, No. 7, 1995, pp. 49-56.
- [9] Loftin, R., and Kenney, P., "Training the Hubble Space Telescope Flight Team," *IEEE Computer Graphics and Applications*, Vol. 15, No. 5, 1995, pp. 31-37.
- [10] Wilson, B., Mourant, R., Li, M., and Xu, W., "A Virtual Environment for Training Overhead Crane Operators: Real-Time Implementation," *IIE Transactions*, Vol. 30, 1998, pp. 589-595.
- [11] Jiing-Yih, L., Ji-Liang, D., Jiun-Ren H., Ming-Chang, J., and Chung-Yun G., "Development of a Virtual Simulation System for Crane-Operating Training," in *Proceedings of ASME*, Paper No. 6p 97-AA-45, 1997.
- [12] Chin-Teng, L., I-Fang C., and Jiann-Yaw L., "Multipurpose Virtual-Reality-Based Motion Simulator," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Vol. 5, 2001, pp. 2846-2851.
- [13] Astachova, T. G., "Mathematical Model of the Semicircular Canal of the Vestibular System as an Angular Acceleration Sensor," *Moscow University Mechanics Bulletin (English Translation of Vestnik Moskovskogo Universiteta, Mekhanika)*, Vol. 44, No. 1, 1989, pp. 34-39.
- [14] Idres, M. M., Youssef, K. S., Nayfeh, A. H., and Mook, D. T., "A Nonlinear 8-DOF Coupled Crane-Ship Dynamic Model," in *Proceedings of the 44th AIAA Structural Dynamics Conference*, Paper No. 1855, Norfolk, VA, 2003.
- [15] Henry, R. J., Masoud, Z. N., Nayfeh, A. H., and Mook, D. T., "Cargo Pendulation Reduction on Ship-Mounted Cranes via Boom-Luff and Slew Angles Actuation," *Journal of Vibration and Control*, Vol. 7, 2001, pp. 1253-1264.

- [16] Masoud, Z., “A Control System for the Reduction of Cargo Pendulation of Ship-Mounted Cranes,” PHD Dissertation, Virginia Polytechnic Institute and State University, Blacksburg, VA, 2000.
- [17] Kelso, J., Arsenault, L., Satterfield, S., Ketchan, P., and Kriz, R., “DIVERSE: A Framework for Building Extensible and Reconfigurable Device Independent Virtual Environments,” *Presence*, Vol. 12, No. 1, 2003, pp. 19-36.
- [18] Moog 6-DOF 2000 E, Interface Definition Manual, 1998.

# Vita

Mohammed F. Daqaq was born on May 15, 1979 in Bethlehem, Palestine. In 1996 he moved to Irbid, Jordan, he attended the Jordan University of Science and Technology and graduated with a Bachelor degree in Mechanical Engineering. Mohammed joined Virginia Tech in 2001 to pursue his Master's degree.