NTNU

Norwegian University of
Science and Technology

Control of an Underwater Robot System Connected to a Ship by a Slender Marine Structure

Wei Li

Master of Science in Engineering Cybernetics
Submission date: May 2008
Supervisor: Tu Duc Nguyen, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem Description

This report addresses the stabilization problem of a marine structure (i.e. cable/riser), connected to a surface vessel at one end and to a thruster unit at the other. The passivity of the system shall be analyzed and a controller shall be designed. A simulation of the system shall be implemented in Matlab.

Assignment given: 20. January 2008
Supervisor: Tu Duc Nguyen, ITK

NTNU

Norwegian University of Science and Technology

Master thesis

Control of an Underwater Robot System Connected to a Ship by a Slender Marine Structure

Wei Li

Supervisor:
Tu Duc Nguyen

Trondheim, May 26, 2008

Faculty of Information Technology, Mathematics and Electrical Engineering
DEPARTMENT OF ENGINEERING CYBERNETICS

NTNU
Norges teknisk-naturvitenskapelige universitet

Fakultet for informasjonsteknologi, matematikk og elektroteknikk
Institutt for teknisk kybernetikk

Diplom oppgave

Kandidatens navn: Wei Li

Fag: Teknisk Kybernetikk

Oppgavens tittel (norsk):

Oppgavens tittel (engelsk): Control of an Underwater system connected to a ship by a Slender Marine Structure

Oppgavens tekst:

1. Develop a mathematical model of the system or study the mathematical model given in ([Schjølberg&Egeland,1996]).
2. Controller Design
3. Implementation (in Matlab)

Oppgaven gitt: 20. jan. 2008

Besvarelsen leveres: 15. jun. 2008

Besvarelsen levert: 27. mai. 2008

Utført ved Institutt for teknisk kybernetikk

Veileder: Tu Duc Nguyen

Trondheim, den 20.01. 2008

Tu Duc Nguyen

Faglærer

Preface

This thesis is submitted in fulfilment of the degree Master of Science at Norwegian University of Science and Technology (NTNU), in the Department of Engineering Cybernetics.

I wish to express special thanks to my supervisor Tu Duc Nguyen for helpful guidance and comments during the prosjekt, I also wish to thank my classmate Lars Gronvold for some suggestions in Matlab programming.

Abstract

This report addresses the stabilization problem of a marine structure (i.e. cable/riser), connected to a surface vessel at one end and to a thruster unit at the other. Here, only motion in the lateral direction has been considered. Stabilization control laws are designed for position and velocity control of the robot system. The passivity of the control system is analyzed, and the closed loop system is shown to be asymptotically stable. Simulation results are presented.

Contents

Chapter 1

Introduction

In this report the system being studied consist of a cable, production riser, conductor or an other similar slender structure connected to a surface vessel at the top end and to a mass module or to a robot system at the bottom end. The robot system may be a manipulator, a tool system equipped with thrusters, a simple thruster system or an ROV. The function of the robot system may be several, one is to perform maintenance and repair on underwater installations. In this case the marine structure is a cable providing power and control signals for the robot system. Another function may be to connect the riser to a well head, the robot system need then only consist of a thruster unit. In floating oil production systems risers are connected to the well heads by controlling the surface vessel. Attaching a thruster unit to the riser bottom end improves performance and time scheduling for this kind of operation. Slender marine structures undergo deformations induced by the motion of the surface vessel, wave and current forces, and these deformations lead to reduced performance of the robot system. This results in the need of a robust and high performance controller for the robot system.

The aim of this report is to describe the dynamics of the total system and to design a controller for position and velocity regulation of the robot system, such that the vibrations and oscillations in the cable/riser unit are attenuated. The control system is designed based on a mathematical model derived for the system. The mathematical model used for the cable/riser system and vessel motion is based on [1]. These equations are combined with the equations of motion of an underwater thruster unit, taking into account the reaction forces between the two systems. The model of the total system is written in a compact form and this facilitates the control system design and passivity analysis. In this report only the motion in the lateral direction has been considered.

This report is outlined as follows: The mathematical model of the system

is derived in Chapter 2. In Chapter 3, passivity analysis and controller design are presented. FEM-modeling is derived in Chapter 4. Chapter 5 and 6 holds the simulation and conclusion.

Chapter 2

Mathematical Model

Figure 2.1 depicts the system to be investigated. It consists of a surface vessel and a cable/riser system connecting an underwater tool system equipped with thrusters to the vessel. In the present approach only the motion in lateral direction (x-direction) has been considered. The mathematical model of the system is adopted from [1] and [7]. It is assumed that the cable/riser is connected to the vessel and to the tool system by means of ball-joints and that this results in small angles of deflection and zero bending moment.

2.1 Wave Loads and Water Current Loads

The cable/riser system is affected by sea waves and water current loads. The lateral wave velocity $\omega(z,t)$ for regular waves is according to [4] defined

$$\omega(z,t) = \omega_n \xi_\alpha e^{-bz} sin(\omega_n t) \tag{2.1}$$

where ξ_α is the wave amplitude, ω_n is the nominal dominating wave frequency, $b = \frac{2\pi}{\lambda}$ and λ is the wave length.

The current velocity U_c is assumed to be independent of time

$$U_c(t,z) = U_c(z) \tag{2.2}$$

Assuming U_c is a linear function. According to fig.(2.2)

$$U_c(0) \in R^+ \quad U_c(0) = U_{c0} > 0 \tag{2.3}$$

$$U_c(z) = -\frac{U_{c0}}{L} z + U_{c0} \qquad z \in [0, L] \tag{2.4}$$

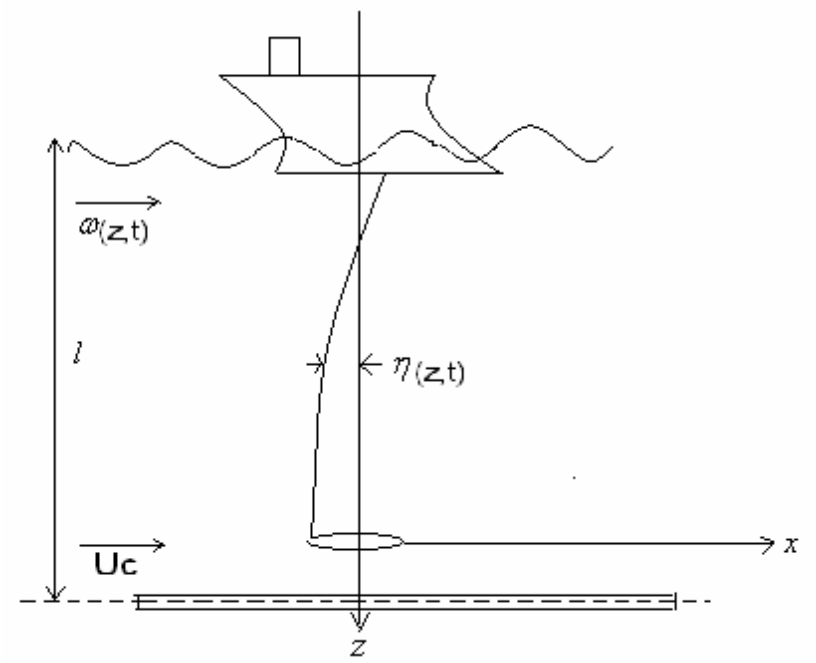

Figure 2.1: A slender marine structure connecting a thruster unit to a ship

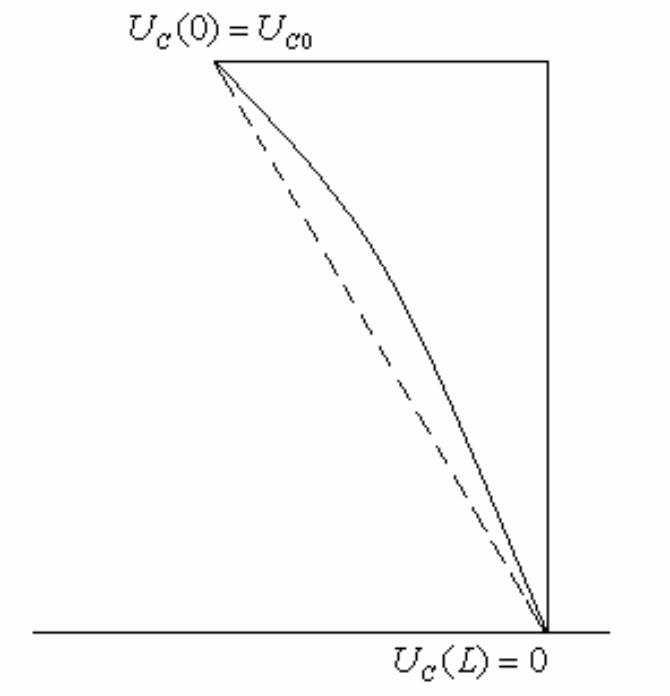

Figure 2.2: Illustration of function $U_c(z)$

2.2 Equations of Motion of the Surface Vessel

The equation of motion of the surface vessel in the lateral direction at depth (z=0) is given by

$$M\ddot{\eta}(0,t)+D_1\dot{\eta}(0,t)+D_2[\dot{\eta}(0,t)-U_c(0)]\,|\dot{\eta}(0,t)-U_c(0)| = \tau_0-f_{cx}(0,t) \quad (2.5)$$

where

$$f_{cx}(0,t) = [(EI\eta_{zz})_z]_{z=0} - [T\eta_z]_{z=0} \quad (2.6)$$

$\eta(0,t)$ is the surface vessel position in x-direction, M is the mass and added mass of the surface vessel in x-direction, D_1 is the linear damping coefficient, D_2 is the drag coefficient, $U_c(0)$ is the current velocity at depth $z=0$, τ_0 is the truster force in the lateral direction and $f_{cx}(0,t)$ is the lateral cable/riser force acting on the surface vessel. The Coriolis and centrifugal forces are small compared to the damping forces and are therefore neglected.

2.3 Equations of Motion of the Tool System

The equation of motion of the tool system in the lateral direction at depth (z=L) is given by

$$m\ddot{\eta}(L,t) + d_1\dot{\eta}(L,t) + d_2[\dot{\eta}(L,t)]\,|\dot{\eta}(L,t)| = \tau_L + f_{cx}(L,t) \quad (2.7)$$

where

$$f_{cx}(L,t) = [(EI\eta_{zz})_z]_{z=L} - [T\eta_z]_{z=L} \quad (2.8)$$

$\eta(L,t)$ is the tool system position in x-direction, m is the mass and added mass of the tool system in x-direction, d_1 is the linear damping coefficient, d_2 is the drag coefficient, τ_L is the truster force in the lateral direction and $f_{cx}(L,t)$ is the lateral cable/riser force acting on the tool system. The Coriolis and centrifugal forces are small compared to the damping forces and are therefore neglected.

2.4 Cable/Riser Dynamics

The horizontal equation of motion of an underwater cable/riser can for small angles of deflection be bescribed by the differential equation [3]

$$\frac{\partial^2}{\partial z^2}(EI(z)\frac{\partial^2\eta(z,t)}{\partial z^2})-\frac{\partial}{\partial z}(T_e(z)\frac{\partial\eta(z,t)}{\partial z})+c\dot{\eta}(z,t)+\rho\ddot{\eta}(z,t) = f_{xs}(z,t) \quad (2.9)$$

where EI is the cable/riser stiffness, T_e is the cable/riser tension, c is the structural damping and ρ is the mass of the cable/riser per unit length. The lateral displacement $\eta(z,t)$ is normal to the cable/riser axis in some direction and varies with the time t and longitudinal displacement z, which is the displacement along the cable/riser axis. It is reasonable to assume that the cable/riser is designed so that the stiffness is constant, $EI(z) = EI$. It is assumed that the cable/riser has a moderate tension and the tension $T_e(z)$ is approximated by the mean value T. This gives a good approximation of the vibration frequencies.

The fluid loading is according to Morison's equation given as

$$f_{xs}(z,t) = c_1\dot{\omega}(z,t) + c_2(\omega(z,t) - \dot{\eta}(z,t) - U_c(z))\,|\omega(z,t) - \dot{\eta}(z,t) - U_c(z)| \tag{2.10}$$

where $c_1 = \frac{\pi}{4}\rho_w C_m D_0^2$, $c_2 = \frac{1}{2}\rho_w C_d D_0$, ρ_w is the mass density of wather, D_0 is diameter of the cable/riser, C_m and C_d are drag coefficients. It is common to assume that C_m and C_d does not vary along the cable/riser. The system is assumed to be designed so that is neutrally buoyant. This results in less energy consumption in the longitudinal direction. Lift forces have been neglected. Moreover it is common to assume that the wave and current velocities are much larger than the cable/riser velocity, this gives

$$f_{xs}(z,t) = c_1\dot{\omega}(z,t) + c_2(\omega(z,t) - U_c(z))\,|\omega(z,t) - U_c(z)| \tag{2.11}$$

2.5 Boundary Conditions

As we mentioned in mathematical modeling, it is assumned that the cable/riser is connected to the vessel and to the tool system by means of ball-joins and that this results in small angles of deflection and zero bending moment. Hence, in addition to (2.5) and (2.7), we have the following static boundary conditions

$$\left[EI\frac{\partial^2\eta}{\partial z^2}\right]_{z=0} = \left[EI\frac{\partial^2\eta}{\partial z^2}\right]_{z=L} = 0 \tag{2.12}$$

Chapter 3

Passivity Analysis and Controller Design

The concept of passivity is very useful in control systems analysis and design. If a system is passive in the sense that it can store and dissipate energy, but it cannot produce energy, then it can be concluded that the total energy of the system will decrease or hold constant, which under certain assumption implies that the system is stable. In this chapter, we will analyse the system and show that the system is passive. Then proper controllers are designed.

3.1 Analysis of Passivity

First of all, let all external forces be zero

$$U_c = 0, \omega(z,t) = 0, f_{cx}(z,t), f_{xs}(z,t) = 0 \tag{3.1}$$

Then, we assume that we have the input signals

$$\tau(t) = [\tau_0(t), \tau_L(t)]^T \tag{3.2}$$

and the measurements

$$y(t) = [\dot{\eta}(0,t), \dot{\eta}(L,t)]^T \tag{3.3}$$

Consider the storage function

$$E_{total} = E_v + E_t + E_c \tag{3.4}$$

where

$$E_v = \frac{1}{2} M \dot{\eta}^2(0,t) \tag{3.5}$$

$$E_t = \frac{1}{2} m \dot{\eta}^2(L,t) \tag{3.6}$$

$$E_c = \frac{1}{2}\int_0^L \rho \dot{\eta}^2(z,t)dz + \frac{1}{2}\int_0^L T\eta_z^2(z,t)dz + \frac{1}{2}\int_0^L EI\eta_{zz}^2(z,t)dz \tag{3.7}$$

Note that E_v, E_t and E_c represent energy functions for the surface vessel, the tool system and the cable/riser.

The time derivative of E_{total} along the solutions of the system (2.5),(2.7),(2.9) and (2.10) is

$$\dot{E}_{total} = \dot{E}_v + \dot{E}_t + \dot{E}_c \tag{3.8}$$

where

$$\begin{aligned} \dot{E}_v &= M\dot{\eta}(0,t)\ddot{\eta}(0,t) \\ &= \{\tau_0 - [(EI\eta_{zz})_z]_{z=0} + [T\eta_z]_{z=0} - D_1\dot{\eta}(0,t) - D_2\dot{\eta}(0,t)\,|\dot{\eta}(0,t)|\} \times \dot{\eta}(0,t) \end{aligned} \tag{3.9}$$

$$\begin{aligned} \dot{E}_t &= m\dot{\eta}(L,t)\ddot{\eta}(L,t) \\ &= \{\tau_L + [(EI\eta_{zz})_z]_{z=L} - [T\eta_z]_{z=L} - d_1\dot{\eta}(L,t) - d_2\dot{\eta}(L,t)\,|\dot{\eta}(L,t)|\} \times \dot{\eta}(L,t) \end{aligned} \tag{3.10}$$

$$\dot{E}_c = \int_0^L \rho\dot{\eta}\ddot{\eta}dz + \int_0^L T\eta_z\dot{\eta}_z dz + \int_0^L EI\eta_{zz}\dot{\eta}_{zz}dz \tag{3.11}$$

Applying the homogeneous equation of (2.9) to the first part in (3.11) gives

$$\int_0^L \rho\dot{\eta}\ddot{\eta}dz = \int_0^L [-EI\frac{\partial^4\eta}{\partial z^4} + T\eta_{zz} - c\dot{\eta}] \times \dot{\eta}dz \tag{3.12}$$

$$\begin{aligned} \int_0^L EI\frac{\partial^4\eta}{\partial z^4}\dot{\eta}dz &= [EI\frac{\partial^3\eta}{\partial z^3}\dot{\eta}]_0^L - \int_0^L EI\frac{\partial^3\eta}{\partial z^3}\dot{\eta}_z dz \\ &= [EI\frac{\partial^3\eta}{\partial z^3}\dot{\eta}]_0^L - [EI\frac{\partial^2\eta}{\partial z^2}\dot{\eta}_z]_0^L + \int_0^L EI\frac{\partial^2\eta}{\partial z^2}\dot{\eta}_{zz}dz \\ &= (EI\eta_{zz})_z\,|_{z=L} \times \dot{\eta}(L,t) - (EI\eta_{zz})_z\,|_{z=0} \times \dot{\eta}(0,t) \\ &\quad - \underbrace{[EI\eta_{zz}\dot{\eta}_z]_0^L}_{=0,due to (2.12)} + \int_0^L EI\frac{\partial^2\eta}{\partial z^2}\dot{\eta}_{zz}dz \end{aligned} \tag{3.13}$$

$$\int_0^L T\eta_{zz}\dot{\eta}dz = [T\eta_z\dot{\eta}]_0^L - \int_0^L T\eta_z\dot{\eta}_z dz \tag{3.14}$$

Insertion of (3.12-3.14) to (3.11) yields

$$\begin{aligned}\dot{E}_c &= -\left[(EI\eta_{zz})_z\right]_{z=L} \times \dot{\eta}(L,t) + \left[(EI\eta_{zz})_z\right]_{z=0} \times \dot{\eta}(0,t) - \int_0^L EI\eta_{zz}\dot{\eta}_{zz}dz \\ &\quad + \left[T\eta_z\right]_{z=L} \times \dot{\eta}(L,t) - \left[T\eta_z\right]_{z=0} \times \dot{\eta}(0,t) - \int_0^L T\eta_z\dot{\eta}_z dz - \int_0^L c\dot{\eta}\dot{\eta}dz \\ &\quad + \int_0^L T\eta_z\dot{\eta}_z dz + \int_0^L EI\eta_{zz}\dot{\eta}_{zz}dz \end{aligned} \tag{3.15}$$

Hence

$$\begin{aligned}\dot{E}_{total} &= \dot{E}_v + \dot{E}_t + \dot{E}_c \\ &= \left[\tau_0 - D_1\dot{\eta}(0,t) - D_2\dot{\eta}(0,t)\left|\dot{\eta}(0,t)\right|\right] \times \dot{\eta}(0,t) \\ &\quad + \left[\tau_L - d_1\dot{\eta}(L,t) - d_2\dot{\eta}(L,t)\left|\dot{\eta}(L,t)\right|\right] \times \dot{\eta}(L,t) - \int_0^L c\dot{\eta}\dot{\eta}dz \\ &\leq \begin{bmatrix} \tau_0 \\ \tau_L \end{bmatrix}^T \begin{bmatrix} \dot{\eta}(0,t) \\ \dot{\eta}(L,t) \end{bmatrix} \end{aligned} \tag{3.16}$$

where (3.9),(3.10) and (3.15) have been applied. This shows that the system is passive with input vector $\tau(t)$ and measurement vector $y(t)$.

3.2 Design of Controllers

The objectives of the controllers are to control the position and velocity of the thruster unit and the surface vessel such that $\{\eta(0,t), \eta(L,t), \dot{\eta}(0,t), \dot{\eta}(L,t)\} \rightarrow \{0,0,0,0\}$ as $t \rightarrow \infty$. Additionally, the designed controllers should also be able to attenuate the vibrations and oscillations in the system due to the sea loads, i.e. $\{|\eta(z,t)|, |\dot{\eta}(z,t)|\} < \infty$, $\forall z \in (0,L)$ and $t \geq 0$.

Due to the passivity analysis above, we propose the DP controllers

$$\tau_0 = -Kd_1 \times \dot{\eta}(0,t) - Kp_1 \times \eta(0,t) \tag{3.17}$$

$$\tau_L = -Kd_2 \times \dot{\eta}(L,t) - Kp_2 \times \eta(L,t) \tag{3.18}$$

for $t \geq 0$, where $Kd_1, Kp_1, Kd_2, Kp_2 > 0$ are controller gains. Since the system is passive with respect to the input vector τt and measurements vector $y(t)$, the stability at the closed loop system (2.5)-(2.9)is guaranteed. (see Theorem 6.1, 6.2 and Lemma 6.8 on page 247-248 in [2]).

We choose another storage function

$$\tilde{E} = E_{total} + \frac{1}{2}Kp_1 \times \eta^2(0,t) + \frac{1}{2}Kp_2 \times \eta^2(L,t) \tag{3.19}$$

where E_{total} is given by (3.4). Taking the time derivative of $\tilde{E}$ along solution trajectories of (2.5), (2.7) and (2.9) gives

$$\begin{aligned}
\dot{\tilde{E}} &= \dot{E}_{total} + Kp_1 \times \eta(0,t)\dot{\eta}(0,t) + Kp_2 \times \eta(L,t)\dot{\eta}(L,t) \\
&= [-Kd_1 \times \dot{\eta}(0,t) - Kp_1 \times \eta(0,t) - D_1\dot{\eta}(0,t) \\
&\quad -D_2\dot{\eta}(0,t)\,|\dot{\eta}(0,t)|] \times \dot{\eta}(0,t) \\
&\quad +[-Kd_2 \times \dot{\eta}(L,t) - Kp_2 \times \eta(L,t) - d_1\dot{\eta}(L,t) \\
&\quad -d_2\dot{\eta}(L,t)\,|\dot{\eta}(L,t)|] \times \dot{\eta}(L,t) \\
&\quad -\int_0^L c\dot{\eta}\dot{\eta}dz + Kp_1 \times \eta(0,t)\dot{\eta}(0,t) + Kp_2 \times \eta(L,t)\dot{\eta}(L,t) \\
&= -Kd_1 \times \dot{\eta}^2(0,t) - D_1\dot{\eta}^2(0,t) - D_2\dot{\eta}^2(0,t)\,|\dot{\eta}(0,t)| \\
&\quad -Kd_2 \times \dot{\eta}^2(L,t) - d_1\dot{\eta}^2(L,t) - d_2\dot{\eta}^2(L,t)\,|\dot{\eta}(L,t)| \\
&\quad -\int_0^L c\dot{\eta}\dot{\eta}dz \\
&\leq 0
\end{aligned} \tag{3.20}$$

The time dirivative of $\tilde{E}$ is negative semidefinite, which implies that $\tilde{E}(t) < \tilde{E}(0)$, for all $t \geq 0$. This shows that all the states, $\eta(0,t)$, $\eta(L,t)$, $\dot{\eta}(0,t)$, $\dot{\eta}(L,t)$ are bounded.

In order to prove convergence of states, we need to establish $\dot{\tilde{E}}(t)$ is uniformly continous. To prove this, it is sufficient to prove that $\ddot{\tilde{E}}(t)$ is bounded $\forall t \geq t_0$.

We take the second derivative of $\tilde{E}$ and get

$$\begin{aligned}
\ddot{\tilde{E}} &= [-2Kd_1 \times \dot{\eta}(0,t)\ddot{\eta}(0,t) - 2D_1\dot{\eta}(0,t)\ddot{\eta}(0,t) \\
&\quad -2D_2\dot{\eta}(0,t)\ddot{\eta}(0,t)\,|\dot{\eta}(0,t)| - D_2\dot{\eta}^2(0,t)\,|\ddot{\eta}(0,t)] \\
&\quad +[-2Kd_2 \times \dot{\eta}(L,t)\ddot{\eta}(L,t) - 2d_1\dot{\eta}(L,t)\ddot{\eta}(L,t) \\
&\quad -2d_2\dot{\eta}(L,t)\ddot{\eta}(L,t)\,|\dot{\eta}(L,t)| - d_2\dot{\eta}^2(L,t)\,|\ddot{\eta}(L,t)|] \\
&\quad +[-2\int_0^L c\dot{\eta}\ddot{\eta}dz]
\end{aligned} \tag{3.21}$$

Equation (2.5) and boundedness of $\eta(0,t)$, $\dot{\eta}(0,t)$ can be utilized to show that $\ddot{\eta}(0,t)$ is bounded $\forall t \in [0,\infty)$.

Similarly, (2.7) and boundedness of $\eta(L,t)$, $\dot{\eta}(L,t)$ can be utilized to show that $\ddot{\eta}(L,t)$ is bounded $\forall t \in [0,\infty)$.

Hence, $\ddot{\tilde{E}}(t)$ is bounded, which proves that $\dot{\tilde{E}}(t)$ is uniformly continous. According to Barbalat's Lemma (page 323 i [2]), $\dot{\tilde{E}}(t) \to 0$ as $t \to \infty$, and then, the states of the system converge to zero as $t \to \infty$.

Now since $\eta(0,t) = 0$, $\forall t \in [0,\infty)$ and all states converge to zero as $t \to \infty$, it is concluded that $\eta(z,t)$ converges to zero as $\forall t \in [0,\infty)$.

Hence, the system is asymptotically stable.

Chapter 4

FEM-Modeling

An alternative technique for analyzing the Euler Bernoulli beam is to use the finite-element method. The finite-element method can be seen as a model formulation based on the Galerkin method [5], where special set of shape functions are used. The characteristic feature of the finite-element method is that the shape functions are locally defined in the sense that they are nonzero only in short intervals of the beam. An alternative way of seeing the finite-element method is that the beam is divided into beam elements. The equations of motion are then derived for the beam element using a cubic shape function, and then the beam model is obtained by connectiong the beam element models using multiport techniques.

4.1 Beam Element

In a finite-element model of an Euler Bernoulli beam the basic building block of the model is a beam element of length h. The element is defined for the interval $0 \leq x \leq h$. At $x = 0$ the shear force is V_1 and the bending moment is M_1, the elastic displacement is ω_1, and the elastic angle is ω_1'. This can be seen as one port with effort V_1 and flow $\dot{\omega}_1$, and one port with effort M_1 and flow $\dot{\omega}_1'$.

At $x = h$ the shear force is V_2, the bending moment is M_2, the elastic deflection is ω_2, and the elastic angle is ω_2'. This is described as a port with effort V_2 and flow $\dot{\omega}_2$, and one port with effort M_2 and flow $\dot{\omega}_2'$.

The usual finite-element model of the Euler Bernoulli beam is based on the displacement formulation where the inputs to the model are the forces and torques, and the outputs are the displacements and the displacement angles.

The displacement in the element is modeled as the cubic expression

$$\omega(x,t) = c_0(t) + c_1(t)x + c_2(t)x^2 + c_3(t)x^3 \tag{4.1}$$

The motivation for using this expression is that in the stationary case the displacement satisfies $\omega'''' = 0$, which has solution (4.1). The generalized coordinates $a_i(t)$ of the beam element are defined as

$$a_1(t) = \omega_1(t) \quad a_2(t) = \omega_1'(t) \tag{4.2}$$
$$a_3(t) = \omega_2(t) \quad a_4(t) = \omega_2'(t) \tag{4.3}$$

Combination of (4.1),(4.2) and (4.3) leads to

$$\omega(z,t) = \sum_{i=1}^{4} \alpha_i(x) a_i(t) \tag{4.4}$$

where the shape functions $\alpha_i(x)$ are given by

$$\alpha_1(x) = 1 - 3(\frac{x}{h})^2 + 2(\frac{x}{h})^3 \tag{4.5}$$

$$\alpha_2(x) = h[(\frac{x}{h}) - 2(\frac{x}{h})^2 + (\frac{x}{h})^3] \tag{4.6}$$

$$\alpha_3(x) = 3(\frac{x}{h})^2 - 2(\frac{x}{h})^3 \tag{4.7}$$

$$\alpha_4(x) = h[-(\frac{x}{h})^2 + (\frac{x}{h})^3] \tag{4.8}$$

These cubic shape functions are called the Hermitian shape functions. Galerkin's method for the beam element leads to

$$\mathbf{M}_e\ddot{\mathbf{a}} + \mathbf{K}_e\mathbf{a} = \mathtt{f} \tag{4.9}$$

where the mass matrix of the element is given by

$$\mathbf{M}_e = \int_0^h \rho\alpha\alpha^T dx = \frac{\rho h}{420}\begin{bmatrix} 156 & 22h & 54 & -13h \\ 22h & 4h^2 & 13h & -3h^2 \\ 54 & 13h & 156 & -22h \\ -13h & -3h^2 & -22h & 4h^2 \end{bmatrix} \tag{4.10}$$

and the stiffness matrix of the element is given by

$$\mathbf{K}_e = \int_0^h EI\alpha''(\alpha'')^T dx = \frac{2EI}{h^3}\begin{bmatrix} 6 & 3h & -6 & 3h \\ 3h & 2h^2 & -3h & h^2 \\ -6 & -3h & 6 & -3h \\ 3h & h^2 & -3h & 2h^2 \end{bmatrix} \tag{4.11}$$

and $\mathtt{f} = (f_1, f_2, f_3, f_4)^T$ where

$$f_i = \int_0^h \alpha_i(x) f(x) \tag{4.12}$$

4.2 Assembling a structure

To establish the model for a beam of length L where $L = Nh$ it is necessary to connect N beam elements. Elements k and $k+1$ can be connected by requiring that the end-point variables satisfy $a_{k,3} = a_{k+1,1}$ and $a_{k,4} = a_{k+1,2}$. Then, there must be forces and torques of constraints to hold the two element together, and the equations of motion for elements k and $k+1$ are given by

$$\mathbf{M}_e \frac{d^2}{dt^2}\begin{pmatrix} a_{k,1} \\ a_{k,2} \\ a_{k,3} \\ a_{k,4} \end{pmatrix} + \mathbf{K}_e \begin{pmatrix} a_{k,1} \\ a_{k,2} \\ a_{k,3} \\ a_{k,4} \end{pmatrix} = \begin{pmatrix} f_{k,1} \\ f_{k,2} \\ f_{k,3} + f_3 \\ f_{k,4} + f_4 \end{pmatrix} \tag{4.13}$$

$$\mathbf{M}_e \frac{d^2}{dt^2}\begin{pmatrix} a_{k+1,1} \\ a_{k+1,2} \\ a_{k+1,3} \\ a_{k+1,4} \end{pmatrix} + \mathbf{K}_e \begin{pmatrix} a_{k+1,1} \\ a_{k+1,2} \\ a_{k+1,3} \\ a_{k+1,4} \end{pmatrix} = \begin{pmatrix} f_{k+1,1} - f_3 \\ f_{k+1,2} - f_4 \\ f_{k+1,3} \\ f_{k+1,4} \end{pmatrix} \tag{4.14}$$

These forces and torques of constraint are eliminated by adding rows 3 and 4 of element k to rows 1 and 2 of element $k+1$. This gives the model

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{u} \tag{4.15}$$

where $\mathtt{q} = (a_{k,1}, a_{k,2}, a_{k+1,1}, a_{k+1,2}, a_{k+1,3}, a_{k+1,4})$. The mass matrix is obtained from

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}_e & \mathbf{0}_{4,2} \\ \mathbf{0}_{2,4} & \mathbf{0}_{2,2} \end{pmatrix} + \begin{pmatrix} \mathbf{0}_{2,2} & \mathbf{0}_{2,4} \\ \mathbf{0}_{4,2} & \mathbf{M}_e \end{pmatrix} \tag{4.16}$$

In the same way the stiffness matrix is obtained from

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}_e & \mathbf{0}_{4,2} \\ \mathbf{0}_{2,4} & \mathbf{0}_{2,2} \end{pmatrix} + \begin{pmatrix} \mathbf{0}_{2,2} & \mathbf{0}_{2,4} \\ \mathbf{0}_{4,2} & \mathbf{K}_e \end{pmatrix} \tag{4.17}$$

Alternatively, the model of the two elements can be written

$$\bar{\mathbf{M}}\frac{d^2}{dt^2}\bar{\mathbf{a}} + \bar{\mathbf{K}}\bar{\mathbf{a}} = \bar{\mathbf{f}} \tag{4.18}$$

$$\bar{\mathbf{a}} = (\mathbf{a}_1, \cdots, \mathbf{a}_p)^T \tag{4.19}$$

$$\bar{\mathbf{M}} = blog \;\; diag(\mathbf{M}_{e1}, \cdots, \mathbf{M}_{ep}), \quad \bar{\mathbf{K}} = blog \;\; diag(\mathbf{K}_{e1}, \cdots, \mathbf{K}_{ep}) \tag{4.20}$$

where the connection of the elements is obtained by requiring

$$\bar{\mathbf{a}} = \mathbf{Cq}, \quad \mathbf{u} = \mathbf{C^T}\bar{\mathbf{f}} \tag{4.21}$$

where

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.22}$$

when N=2. Then the mass matrix and the stiffness matrix are found from

$$\mathbf{M} = \mathbf{C}^T\bar{\mathbf{M}}\mathbf{C}, \quad \mathbf{K} = \mathbf{C}^T\bar{\mathbf{K}}\mathbf{C} \tag{4.23}$$

to be

$$\mathbf{M} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} & 0 & 0 \\ m_{21} & m_{22} & m_{23} & m_{24} & 0 & 0 \\ m_{31} & m_{32} & m_{33}+m_{11} & m_{34}+m_{12} & m_{13} & m_{14} \\ m_{41} & m_{42} & m_{43}+m_{21} & m_{44}+m_{22} & m_{23} & m_{24} \\ 0 & 0 & m_{31} & m_{32} & m_{33} & m_{34} \\ 0 & 0 & m_{41} & m_{42} & m_{43} & m_{44} \end{pmatrix} \tag{4.24}$$

$$\mathbf{K} = \begin{pmatrix} k_{11} & k_{12} & k_{13} & k_{14} & 0 & 0 \\ k_{21} & k_{22} & k_{23} & k_{24} & 0 & 0 \\ k_{31} & k_{32} & k_{33}+k_{11} & k_{34}+k_{12} & k_{13} & k_{14} \\ k_{41} & k_{42} & k_{43}+k_{21} & k_{44}+k_{22} & k_{23} & k_{24} \\ 0 & 0 & k_{31} & k_{32} & k_{33} & k_{34} \\ 0 & 0 & k_{41} & k_{42} & k_{43} & k_{44} \end{pmatrix} \tag{4.25}$$

and the resulting model is

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{u} \tag{4.26}$$

4.3 Finite Element Model and Galerkin's Method

A finite-element model for an Euler Bernoulli beam can alternatively be established by applying Galerkin's method with shape functions $\phi_i(x)$ based on the element shape functions in (4.5-4.8). For the Euler Bernoulli beam, N nodes are defined at $x_1 < x_2 < \cdots < x_N$, and the deflection is described by

$$\omega(x,t) = \sum_{j=1}^{N} [\alpha_{j,1}(x)a_{j,1}(t) + \alpha_{j,2}(x)a_{j,2}(t)] \tag{4.27}$$

which is expressed in the form

$$\omega(x,t) = \sum_{j=1}^{2N} \phi_j(x)q_j(t) \tag{4.28}$$

where the generalized coordinates are $q = (a_{1,1}, a_{1,2}, \cdots, a_{N,1}, a_{N,2})^T$ and the mode shape vector is $\phi = (\phi_{1,1}, \phi_{1,2}, \cdots, \phi_{N,1}, \phi_{N,2})^T$. The shape functions $\alpha_{j,1}(x)$ and $\alpha_{j,2}(x)$ for the Euler Bernoulli beam are selected in agreement with (4.5-4.8) as the Hermitian shape functions

$$\alpha_{i,1}(x) = \begin{cases} 1 - 3\frac{(x-x_i)^2}{\ell_i^2} + 2\frac{(x-x_i)^3}{\ell_i^3} & \text{if } x_i \le x \le x_{i+1} \\ 3\frac{(x-x_{i-1})^2}{\ell_{i-1}^2} - 2\frac{(x-x_{i-1})^3}{\ell_{i-1}^3} & \text{if } x_{i-1} \le x \le x_i \\ 0 & \text{otherwise} \end{cases}$$

$$\alpha_{i,2}(x) = \begin{cases} x - 2\frac{(x-x_i)^2}{\ell_i} + \frac{(x-x_i)^3}{\ell_i^2} & \text{if } x_i \le x \le x_{i+1} \\ -\frac{(x-x_{i-1})^2}{\ell_{i-1}} + \frac{(x-x_{i-1})^3}{\ell_{i-1}^2} & \text{if } x_{i-1} \le x \le x_i \\ 0 & \text{otherwise} \end{cases}$$

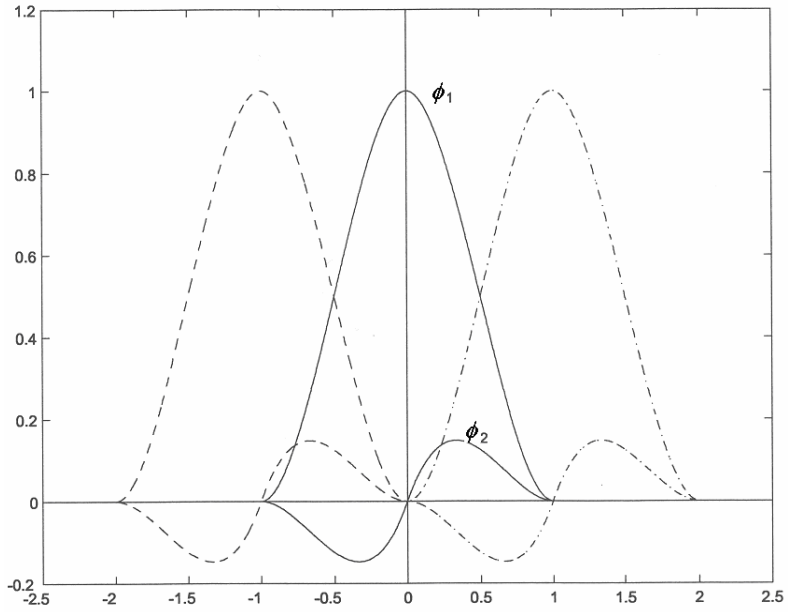

Figure 4.1: Shape function for Euler-Bernoulli beam

These shape functions satisfy

$$\phi_{2k-1} = \alpha_{j,1}(x_k) = \delta_{jk}, \quad \phi'_{2k-1} = \alpha'_{j,1}(x_k) = 0 \tag{4.29}$$

$$\phi_{2k} = \alpha_{j,2}(x_k) = 0, \quad \phi'_{2k} = \alpha'_{j,2}(x_k) = \delta_{jk} \tag{4.30}$$

This gives the following physical interpretation of the generalized coordinates $q_{2k-1} = a_{k,1}(t)$ and $q_{2k} = a_{k,2}(t)$:

$$q_{2k-1} = a_{k,1}(t) = \omega(x_k, t) \tag{4.31}$$

$$q_{2k} = a_{k,2}(t) = \omega'(x_k, t) \tag{4.32}$$

Insertion of (4.28) gives

$$\sum_{i=1}^{2N} [\rho \ddot{q}_j(t)\phi_j(x) + \rho c^2 q_j(t)\phi_j''''(x)] = b(x)u(t) \tag{4.33}$$

In the Galerkin method the equation of motion is premultiplied by $\phi_i(x)$ and intergrated over the interval $x \in [0, \ell]$. This gives the expression

$$\int_0^\ell \phi_i(x) \sum_{i=1}^{2N} [\rho \ddot{q}_j(t)\phi_j(x) + \rho c^2 q_j(t)\phi_j''''(x)]dx = \int_0^\ell \phi_i(x)b(x)u(t)dx \tag{4.34}$$

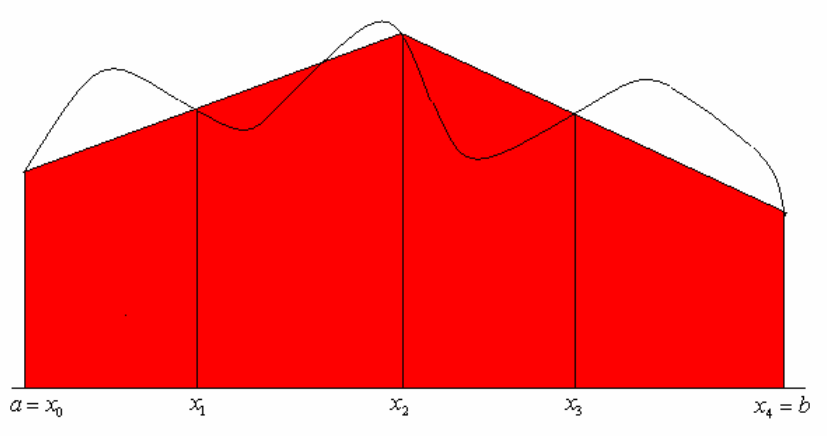

Figure 4.2: Trapezoid Rule

4.4 Trapezoid Rule

The Trapezoid Rule [9] is applied to get the approximation of the fluid loading $f_{xs}(z,t)$.

The Trapezoid Rule is based on an estimation of the area beneath a curve using trapezoids. The estimation of $\int_a^b f(x)dx$ is approached by first dividing the interval $[a,b]$ into subintervals according to the partition $P = \{a = x_0 < x_1 < x_2 < \cdots < x_n = b$. For each such partition of the interval (the partition points x_i need not be uniformly spaced), an estimation of the integral by the trapezoid rule is obtained. We denote it by $T(f;P)$. Fig.(4.2) shows what the trapezoids are.

A typical trapezoid has the subinterval $[x_i, x_{i+1}]$ as its base, and the two vertical sides are $f(x_i)$ and $f(x_{i+1})$.(see fig.(4.3)).

The area is equal to the base times the average height, and we have the Basic Trapezoid Rule for the subinterval $[x_i, x_{i+1}]$

$$\int_{x_i}^{x_{i+1}} f(x)dx \approx A_i = \frac{1}{2}(x_{i+1} - x_i)[f(x_i) + f(x_{i+1})] \tag{4.35}$$

Hence, the total area of all the trapezoids is

$$\int_a^b f(x)dx \approx T(f;P) = \sum_{i=0}^{n-1} A_i = \frac{1}{2}\sum_{i=0}^{n-1}(x_{i+1} - x_i)[f(x_i) + f(x_{i+1})] \tag{4.36}$$

which is called the Composite Trapezoid Rule.

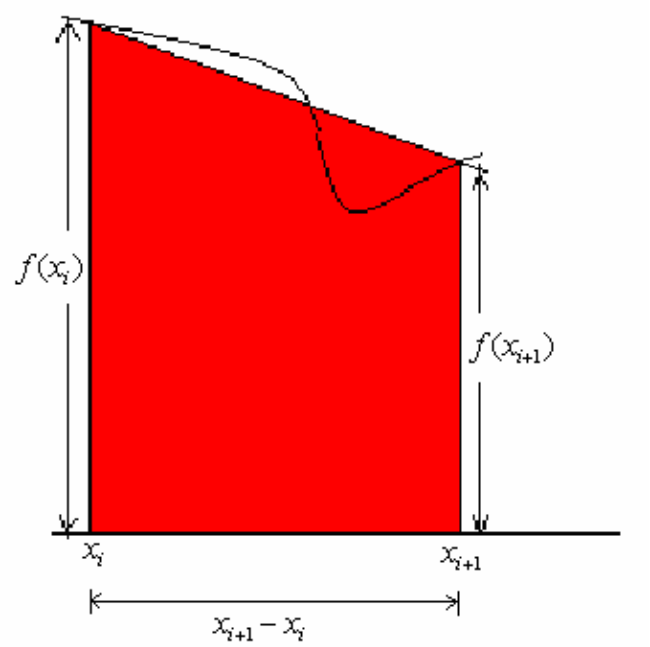

Figure 4.3: Typical trapezoid

The division points x_i are equally spaced, $x_i = a+ih$, where $h = (b-a)/n$ and $0 \leq i \leq n$, so, the formula for $T(f;P)$ can be given in simpler form because $x_{i+1} - x_i = h$. Thus

$$T(f;P) = \frac{h}{2}\sum_{i=0}^{n-1}[f(x_i) + f(x_{i+1})] \tag{4.37}$$

It should be emphasized that, in order to economize the amount of arithmetic, the computationally preferable formula for the composite trapezoid rule is

$$\int_a^b f(x)dx \approx T(f;P) = h\left\{\sum_{i=1}^{n-1} f(x_i) + \frac{1}{2}[f(x_0) + f(x_n)]\right\} \tag{4.38}$$

4.5 FEM-modeling

For Cable/Riser: we use the homogeneous equation of (2.9) first, and then we will use the Trapezoid Rule to get a approximation of fluid loading $f_{xs}(z,t)$.

$$\rho\ddot{\eta}(z,t) + EI\frac{\partial^4\eta(z,t)}{\partial z^4} - T\frac{\partial^2\eta(z,t)}{\partial z^2} + c\dot{\eta}(z,t) = 0 \tag{4.39}$$

In the Galerkin method the equation of motion is premultiplied by test function ϑ and intergrated over the interval $x \in [0, L]$. This gives the expression

$$\int_0^L \vartheta[\rho\ddot{\eta} + EI\frac{\partial^4\eta}{\partial z^4} - T\frac{\partial^2\eta}{\partial z^2} + c\dot{\eta}]dz = 0 \tag{4.40}$$

$$\vartheta \in \{\phi_1, \phi_2, \phi_3, ...\phi_N\} \tag{4.41}$$

$$\eta(z,t) = \phi(z)^T q(t) \tag{4.42}$$

Inserting (4.42) to (4.40) gives

$$\begin{aligned}\int_0^L \rho\vartheta\phi(z)\ddot{q}(t)dz + \underbrace{\int_0^L EI\vartheta\phi''''(z)q(t)dz}_{eq.(4.44)} \\ - \underbrace{\int_0^L T\vartheta\phi''(z)q(t)dz}_{eq.(4.45)} + \int_0^L c\vartheta\phi(z)\dot{q}(t)dz = 0\end{aligned} \tag{4.43}$$

$$\begin{aligned}\int_0^L EI\vartheta\phi''''(z)q(t)dz &= [EI\vartheta\phi'''(z)q(t)]_0^L - \int_0^L EI\vartheta'\phi'''(z)q(t)dz \\ &= [EI\vartheta\phi'''(z)q(t)]_0^L - \underbrace{[EI\vartheta'\phi''(z)q(t)]_0^L}_{=0, due to (2.12)} \\ &\quad + \int_0^L EI\vartheta''\phi''(z)q(t)dz\end{aligned} \tag{4.44}$$

$$\int_0^L T\vartheta\phi''(z)q(t)dz = [T\vartheta\phi'(z)q(t)]_0^L - \int_0^L T\vartheta'\phi'(z)q(t)dz \tag{4.45}$$

Let $\vartheta = \{\phi_1, \phi_2, \phi_3, ...\phi_N\}$. From (4.44) and (4.45), we get

$$\int_0^L EI\underbrace{\begin{bmatrix}\phi_1\\ \phi_2\\ \phi_3\\ ...\\ \phi_N\end{bmatrix}}_{\phi(z)}\phi''''(z)q(t)dz = [EI\underbrace{\begin{bmatrix}\phi_1\\ \phi_2\\ \phi_3\\ ...\\ \phi_N\end{bmatrix}}_{\phi(z)}\phi'''(z)q(t)]_0^L + \int_0^L EI\underbrace{\begin{bmatrix}\phi_1''\\ \phi_2''\\ \phi_3''\\ ...\\ \phi_N''\end{bmatrix}}_{\phi''(z)}\phi''(z)q(t)dz \tag{4.46}$$

$$\int_0^L T \underbrace{\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ ... \\ \phi_N \end{bmatrix}}_{\phi(z)} \phi''(z)q(t)dz = [T \underbrace{\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ ... \\ \phi_N \end{bmatrix}}_{\phi(z)} \phi'(z)q(t)]_0^L - \int_0^L T \underbrace{\begin{bmatrix} \phi_1' \\ \phi_2' \\ \phi_3' \\ ... \\ \phi_N' \end{bmatrix}}_{\phi'(z)} \phi'(z)q(t)dz \tag{4.47}$$

Thus

$$\begin{aligned} &[\int_0^L \rho\phi(z)\phi(z)dz]\ddot{q}(t) + [EI\phi(z)\phi'''(z)q(t)]_0^L - [EI\phi'(z)\phi''(z)q(t)]_0^L \\ &+[\int_0^L EI\phi''(z)\phi''(z)dz]q(t) - [T\phi(z)\phi'(z)q(t)]_0^L + [\int_0^L T\phi'(z)\phi'(z)dz]q(t) \\ &+[\int_0^L c\phi(z)\phi(z)dz]\dot{q}(t) = 0 \end{aligned} \tag{4.48}$$

The equation of motion of the surface vessel is

$$\begin{aligned} &M\ddot{\eta}(0,t) + D_1\dot{\eta}(0,t) + D_2[\dot{\eta}(0,t) - U_c(0)]\,|\dot{\eta}(0,t) - U_c(0)| \\ &= \tau_0 - [(EI\eta_{zz})_z]_{z=0} + [T\eta_z]_{z=0} \end{aligned} \tag{4.49}$$

We deal with $D_2[\dot{\eta}(0,t) - U_c(0)]\,|\dot{\eta}(0,t) - U_c(0)|$ later, because it contains ulinear element.(see calculation of matrix G_3 in Chapter 8.2) The equation of the surface vessel is premultiplied by ϑ, so it gives

$$\begin{aligned} &[\vartheta M\phi(z)\ddot{q}(t)]_{z=0} + [\vartheta D_1\phi(z)\dot{q}(t)]_{z=0} \\ &= [\vartheta\tau_0]_{z=0} - [\vartheta EI\phi'''(z)q(t)]_{z=0} + [\vartheta T\phi'(z)q(t)]_{z=0} \end{aligned} \tag{4.50}$$

when $\vartheta = \{\phi_1, \phi_2, \phi_3, ...\phi_N\}$, and we do as same as eq.(4.46), the expression for the surface vessel is

$$\begin{aligned} &[M\phi(z)\phi(z)\ddot{q}(t)]_{z=0} + [D_1\phi(z)\phi(z)\dot{q}(t)]_{z=0} \\ &= [\phi(z)\tau_0]_{z=0} - [EI\phi(z)\phi'''(z)q(t)]_{z=0} + [T\phi(z)\phi'(z)q(t)]_{z=0} \end{aligned} \tag{4.51}$$

The equation of motion of the tool system is

$$
\begin{aligned}
&m\ddot{\eta}(L,t) + d_1\dot{\eta}(L,t) + d_2[\dot{\eta}(L,t)]\,|\dot{\eta}(L,t)| \\
&= \tau_L + [(EI\eta_{zz})_z]_{z=L} - [T\eta_z]_{z=L}
\end{aligned} \tag{4.52}
$$

The same reason as the surface vessel, we deal with $d_2[\dot{\eta}(L,t)]\,[\dot{\eta}(L,t)]$ later.(see calculation of matrix G_3 in Chapter 8.2) The equation of the tool system is premultiplied by ϑ, so it gives

$$
\begin{aligned}
&[\vartheta m\phi(z)\ddot{q}(t)]_{z=L} + [\vartheta d_1\phi(z)\dot{q}(t)]_{z=L} \\
&= [\vartheta\tau_L]_{z=L} + [\vartheta EI\phi'''(z)q(t)]_{z=L} - [\vartheta T\phi'(z)q(t)]_{z=L}
\end{aligned} \tag{4.53}
$$

The same method as the surface vessel, the expression for the tool system is

$$
\begin{aligned}
&[m\phi(z)\phi(z)\ddot{q}(t)]_{z=L} + [d_1\phi(z)\phi(z)\dot{q}(t)]_{z=L} \\
&= [\phi(z)\tau_L]_{z=L} + [EI\phi(z)\phi'''(z)q(t)]_{z=L} - [T\phi(z)\phi'(z)q(t)]_{z=L}
\end{aligned} \tag{4.54}
$$

(4.48) + (4.51) + (4.54) →:

$$
\begin{aligned}
&\underbrace{\left\{[\int_0^L \rho\phi(z)\phi(z)dz] + [M\phi(z)\phi(z)]_{z=0} + [m\phi(z)\phi(z)]_{z=L}\right\}}_{M-matrix}\ddot{q}(t) \\
&+\underbrace{\left\{[\int_0^L c\phi(z)\phi(z)dz] + [D_1\phi(z)\phi(z)]_{z=0} + [d_1\phi(z)\phi(z)]_{z=L}\right\}}_{D-matrix}\dot{q}(t) \\
&+\underbrace{\left\{\int_0^L EI\phi''(z)\phi''(z)dz + \int_0^L T\phi'(z)\phi'(z)dz\right\}}_{K-matrix}q(t) = [\phi(z)\tau_0]_{z=0} + [\phi(z)\tau_L]_{z=L}
\end{aligned} \tag{4.55}
$$

The right side av the equation above is

$$
\begin{aligned}
&[\phi(z)\tau_0]_{z=0} + [\phi(z)\tau_L]_{z=L} = \\
&\underbrace{\begin{bmatrix} -Kd_1 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & -Kd_2 & 0 \\ 0 & \cdots & \cdots & 0 \end{bmatrix}}_{KD-matrix}\begin{bmatrix} \dot{\eta}(0,t) \\ 0 \\ \vdots \\ \dot{\eta}(L,t) \\ 0 \end{bmatrix} + \underbrace{\begin{bmatrix} -Kp_1 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & -Kp_2 & 0 \\ 0 & \cdots & \cdots & 0 \end{bmatrix}}_{KP-matrix}\begin{bmatrix} \eta(0,t) \\ 0 \\ \vdots \\ \eta(L,t) \\ 0 \end{bmatrix}
\end{aligned} \tag{4.56}
$$

The eq.(4.55) and eq.(4.56) give the model

$$M\ddot{q} + (D - KD)\dot{q} + (K - KP)q = 0 \tag{4.57}$$

or:

$$\ddot{q} = M^{-1}[(-D + KD)\dot{q} + (-K + KP)q] \tag{4.58}$$

Chapter 5

Simulation

5.1 System Data

L	=	600	Length of the cable[m]
ρ	=	1	mass density of cable [kg/m]
EI	=	4.27×10^8	stiffness of the cable[Nm^2]
M	=	9.6×10^7	mass of the rig/master vessel[kg]
m	=	30	mass of the tool system[kg]
D_1	=	0.9×10^6	linear drag coefficient of the surface vessel
D_2	=	1×10^6	quadratic drag coefficient of the surface vessel
d_1	=	100	linear drag coefficient of the tool system
d_2	=	820	quadratic drag coefficient of the tool system
λ	=	20	the wave length[m]
Uc	=	2	the current velocity[m/s]
ω_n	=	10π	the nominal dominationg wave frequency[Hz]
ξ_a	=	0.1	the wave amplitude[m]
b	=	$2\pi/\lambda$	
D_0	=	0.1	the outer diameter of the cable[m]
C_m	=	1	added mass coefficient
C_d	=	1.6	friction coefficient
c_1	=	$\frac{\pi}{4}\rho C_m D_0^2$	hydrodynamic added mass coefficient
c_2	=	$0.5\rho C_d D_0$	hydrodynamic drag coefficient

5.2 Simulation

In fact, we choose the following PID controller in the system

$$\tau_0 = -Kd_1 \times \dot{\eta}(0,t) - Kp_1 \times \eta(0,t) - Ki_1 \times \int_0^L \eta(z,t)dz \qquad (5.1)$$

$$\tau_L = -Kd_2 \times \dot{\eta}(L,t) - Kp_2 \times \eta(L,t) - Ki_2 \times \int_0^L \eta(z,t)dz \qquad (5.2)$$

where $Kd_1, Kp_1, Ki_1, Kd_2, Kp_2, Ki_2 > 0$ are controller gains.

The purpose of the integral is to wield the influence of the sea waves and water current. The stability of the closed loop is not proved her, but it is proved in [8].

To illustrate the theoretical results and the properties of the closed loop system, simulation results are presented. The figures below are divided three cases, each case has one figure in 2 dimension and one or two figures in 3 dimension. In all cases, we use the same controller gains, which are

$$\begin{aligned} Kd_1 &= 5 \times 10^7 \\ Kd_2 &= 20 \\ Kp_1 &= 8 \times 10^7 \\ Kp_2 &= 15 \\ Ki_1 &= 1 \times 10^7 \\ Ki_2 &= 18 \end{aligned}$$

The initial conditions are

Case 1: $\int_0^L \eta(z,0)dz = 0, \eta(z,0) = 0, \dot{\eta}(z,0) = 0.$

Case 2: $\eta(z,0) = sin(2\pi\frac{i}{N})$.

Case 3: Only $\eta(5,0) = -0.5$.

All the figures show that the controllers (5.1) and (5.2) are well designed, which can hold the system stable.

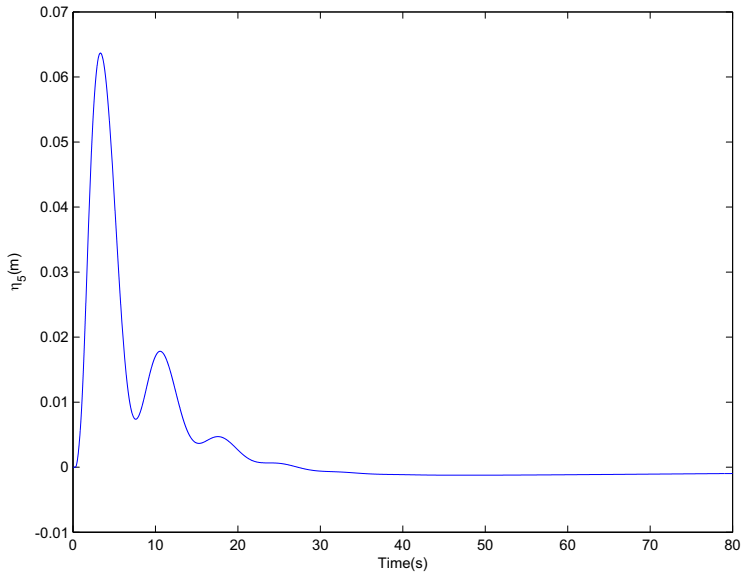

Figure 5.1: Case 1: 2D-plot for node 5

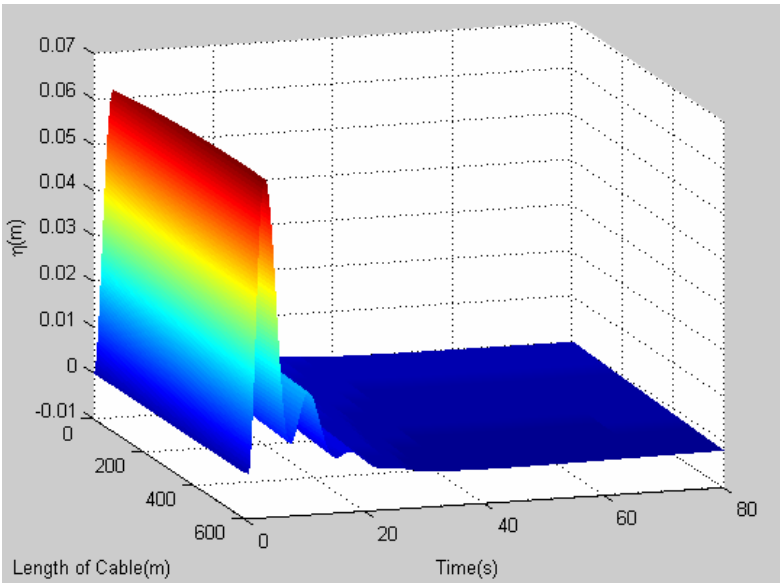

Figure 5.2: Case 1: 3D-plot for the system

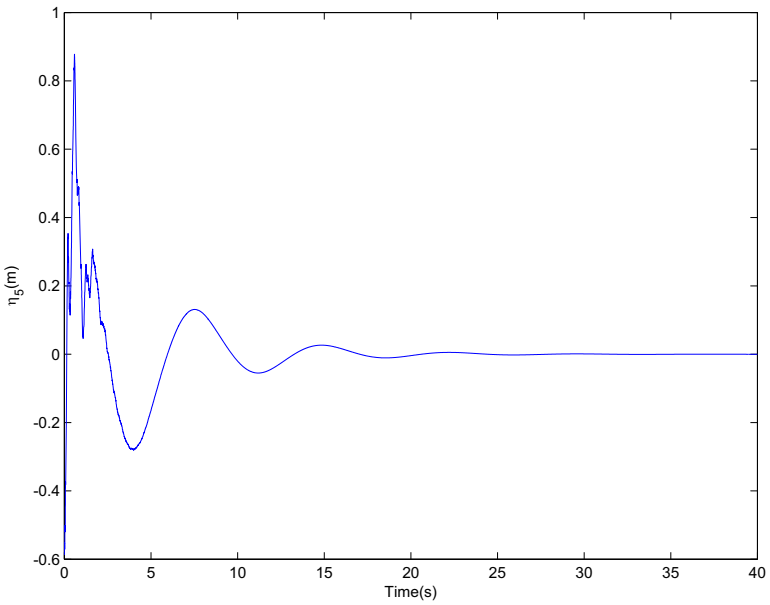

Figure 5.3: Case 2: 2D-plot for node 5

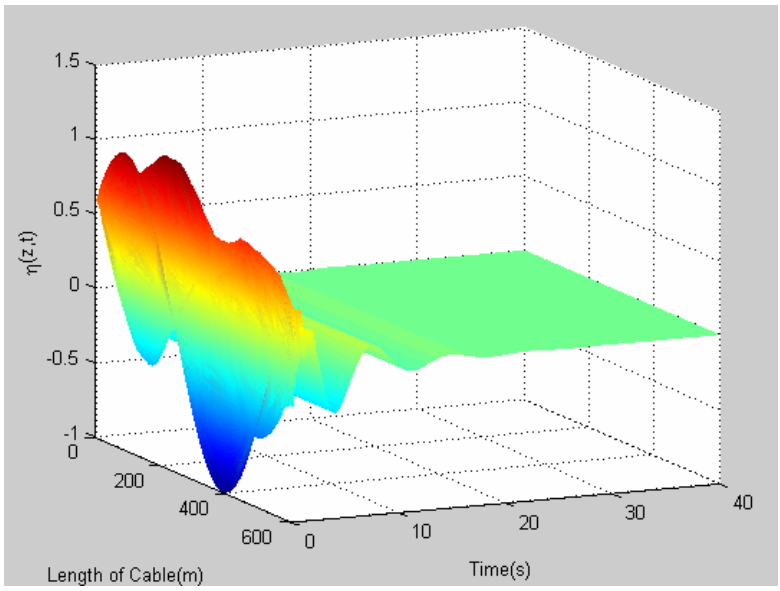

Figure 5.4: Case 2: 3D-plot for the system

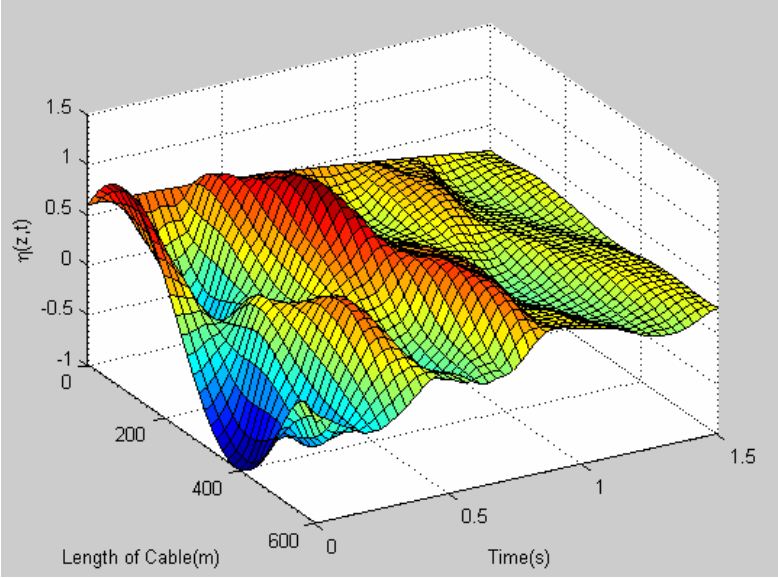

Figure 5.5: Case 2: 3D-plot for the system in the first 1.5s

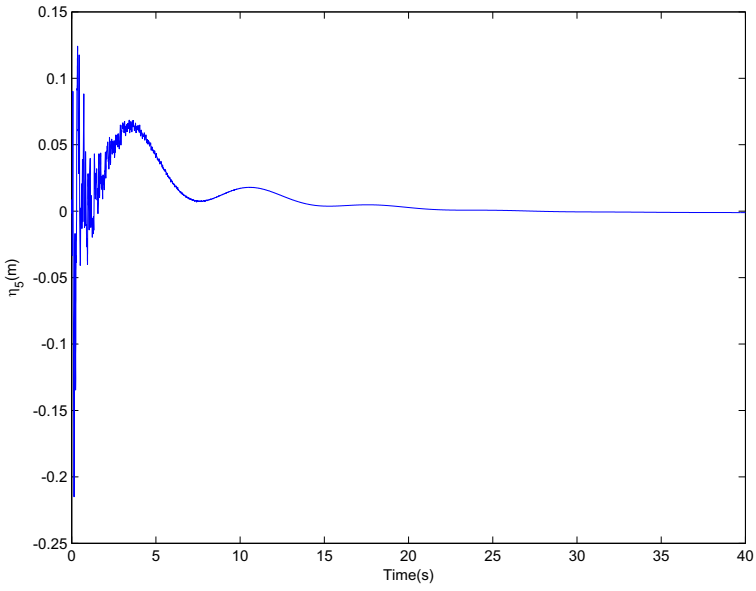

Figure 5.6: Case 3: 2D-plot for node 5

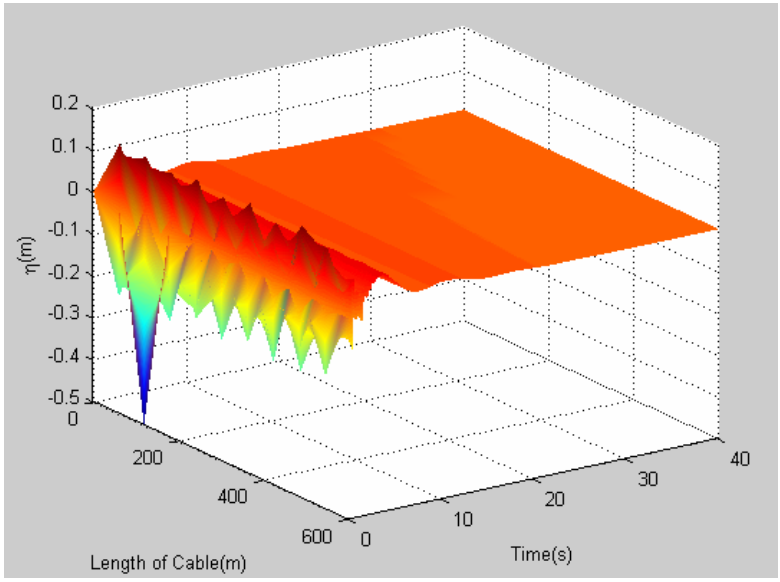

Figure 5.7: Case 3: 3D-plot for the system

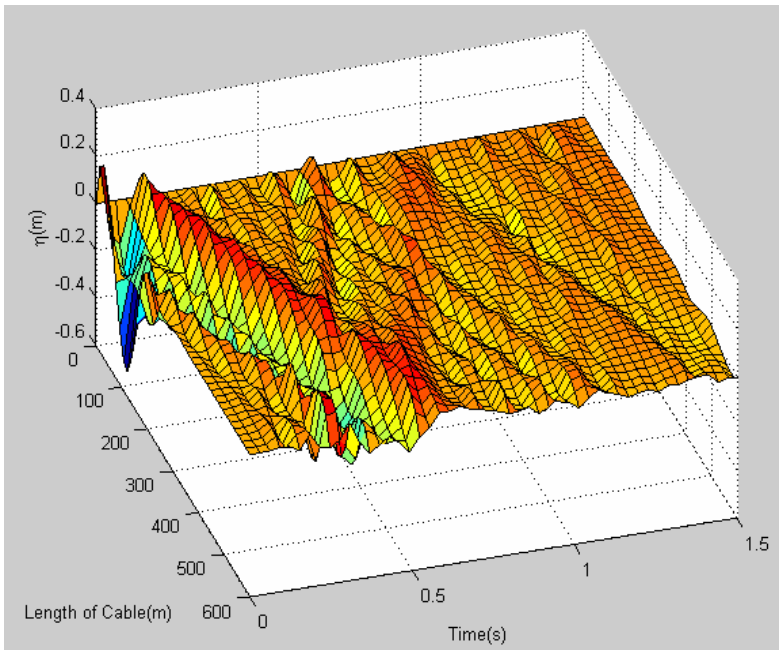

Figure 5.8: Case 3: 3D-plot for the system in the first 1.5s

Chapter 6

Conclusion

This report describes modeling and control of a system consisting of a cable/riser connecting to a surface vessel on the top, and an underwater thruster system at the bottom. A passivity analysis of the system has been presented and it was shown that a classical PID-controller for position control of the thruster system gives good system performance and vibration damping in the cable/riser system. Simulations confirm these results.

Chapter 7

Reference

1. Ingrid Schjølberg and Olav Egeland, *Controll of an Underwater Robot System Connected to a Ship by a Slender Marine Structure*, Department of Eng. Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway, 1996

2. Hassan K.Khalil, *Nonlinear System*, Department of Electrical and Computer Engineering, Michigan State University, Third Edition, 2002

3. S.K Chakrabarti and R.E. Framptom, *Review of Riser Analysis Techniques.* Applied Ocean Research, 4(2): 73-90, 1982

4. O.M. Faltinsen, *Sea Loads on Ships and Offshore Structures.* Cambridge University Press, New York, 1990

5. Olav Egeland and Jan Tommy Gravdahl, *Modeling and Simulation for Automatic Control.* Norwegian University of Science and Technology, 2003

6. Svein Ersdal, *An Experimental Study of Hydrodynamic Forces on Cylinders and Cables in Near Axial Flow.*Norwegian University of Science and Technology, 2004

7. Fossen, T. I., *Marine Control Systems: Guidance, Navigation, and Control of Ships, Rigs and Underwater Vehicles*, Marine Cybernetics, Trondheim, Norway, 2002

8. Tu Duc Nguyen, *Boundary Stabilization of Marine Structure.* Norwegian University of Science and Technology, Trondheim, Norway, 2008

9. Ward Cheney and David Kincaid, *Numerical Mathematics and Computing.* The University of Texas at Austin, Fourth Edition

Chapter 8

Appendix Program Code

Here are program code for simulation. Except the last one is executed by Maple, all the others are executed by Matlab.

1. M-matrise.m; define the system constant, calculate matrix M, K, D, G_1, G_2, G_4, and then simulate the system and get the figures
2. pro.m; calculate f_{xs} and matrix G_3
3. baat-test.m; this is a test program which makes a controller just for the surface vessel
4. baat-system-test.m; a function which is used in baat-test.m
5. system-matrix.mw; program Maple is used to calculate some matrix

8.1 M-matrise.m

```
global M D G1 G2 K N omega_n xi_a b Uc h c1 c2 D2 d2 G4 L
L=600;
N=10;
h=L/N;
rho=1;
EI=4.27*10^8;
C_damping=1;
m_bot=9.6*10^7;
m_ubot=30;

D_0=0.1;
Cm=1;
```

```
Cd=1.6;


c1=pi/4*rho*Cm*D_0^2;
c2=0.5*rho*Cd*D_0;

T=1.11*10^6;
d1=100;
d2=820;
D1=0.9*10^6;
D2=1*10^6;
lamda=20;
Uc=2;

omega_n= 2*pi*5;
xi_a= 0.1;
b=2*pi/lamda;



Me=rho*h/420*[156   22*h    54      -13*h; ...
              22*h   4*h^2   13*h    -3*h^2; ...
              54     13*h    156     -22*h; ...
              -13*h  -3*h^2  -22*h   4*h^2];


Kd=2*EI/h^3*[6     3*h     -6       3*h;...
             3*h   2*h^2   -3*h     h^2;...
             -6    -3*h    6       -3*h;...
             3*h   h^2     -3*h    2*h^2;];

 Kdd=[6/(5*h)     1/10       -6/(5*h)    1/10;...
    1/10         2*h/15     -1/10      -1*h/30;...
    -6/(5*h)     -1/10       6/(5*h)     -1/10;...
    1/10         -1/30*h     -1/10       2/15*h;];



M=zeros((N+1)*2,(N+1)*2);

K1=zeros((N+1)*2,(N+1)*2);
```

```
K2=zeros((N+1)*2,(N+1)*2);


for i=1:N
    a=2*i-1;
    b=2*(i+1);
    M(a:b,a:b)=M(a:b,a:b)+Me;
    K1(a:b,a:b)=K1(a:b,a:b)+Kd;
    K2(a:b,a:b)=K2(a:b,a:b)+Kdd;
end

D=M/rho*C_damping;
D(1,1)=D(1,1)+D1;
D(end-1,end-1)=D(end-1,end-1)+d1;

M(1,1)=M(1,1)+m_bot;
M(end-1,end-1)=M(end-1,end-1)+m_ubot;

K=K1+K2*T;

Kd1=5*10^7;
Kd2=20;
Kp1=8*10^7;
Kp2=15;
Ki1=1*10^7;
Ki2=18;

G1=zeros((N+1)*2,(N+1)*2);
G1(1,1)=-Kd1;
G1(end-1,end-1)=-Kd2;

G2=zeros((N+1)*2,(N+1)*2);
G2(1,1)=-Kp1;
G2(end-1,end-1)=-Kp2;

G4=zeros((N+1)*2,(N+1)*2);
G4(1,1)=-Ki1;
G4(end-1,end-1)=-Ki2;

% test=[  zeros(4,4) eye(4,4);...
```

```
%
%       -M^(-1)*(K-G2) -M^(-1)*(D-G1)];

% eig(test)
first = 1;
%%

q0=zeros((N+1)*2,1);

%q0(5)=-0.5;

% for i=1:N
%     q0(i*2-1)=sin(i/N*2*pi);
% end



[T,Q]=ode45('pro', [0 80], [zeros((N+1)*2,1);q0;zeros((N+1)*2,1)]);


%% continue simulation
if first == 0
    [Tcont,Qcont]=ode45('pro', [0 20]+T(end), Q(end,:));
    Q=[Q;Qcont(2:end,:)];
    T=[T;Tcont(2:end)];
end
first = 0;
%%

z=((1:(N+1))-1)*h;
surf(0:h:L,T(1:end),Q(1:end,end/3+1:2:end*2/3))
% surf(0:h:L,T(1:end),Q(1:end,1:2:end/3))
shading interp

%%

XI = linspace(0,L,N*4); % dybde
% YI = T(1:20:end*0.05);%linspace(0,T(end),100); % tid
YI = linspace(0,1.5,50);
[XI,YI] = meshgrid(XI,YI)
ZI = interp2(0:h:L,T(1:end),Q(1:end,end/3+1:2:end*2/3),XI,YI,'cubic');
```

```
figure
surf(XI,YI,ZI);
shading faceted

hold on
%%

figure
plot(T,Q(:,end/3 + 11),'-b');
```

8.2 pro.m

```
function [y_dot] = pro(T,Y)
global M D G1 G2 K N omega_n xi_a b Uc h c1 c2 D2 d2 G4 L

q_int=Y(1:(N+1)*2);
q = Y((N+1)*2+1:4*(N+1));
q_dot=Y((N+1)*4+1:end);


% size(M)
% size(D)
% size(G1)
% size(q_dot)
%
% size(K)
% size(G2)
% size(q)

    function U=Ucc(z)
        U=Uc-Uc*z/L;
    end

    function w=omega(z,t)
        w=omega_n*xi_a*exp(-b*z)*sin(omega_n*t);
    end
```

```
    function w=omega_dot(z,t)
        w=omega_n^2*xi_a*exp(-b*z)*cos(omega_n*t);
    end


    function f = f_xs(z,t)
        f = c1*omega_dot(z,t)+c2*(omega(z,t)-Ucc(z))*abs(omega(z,t)-Ucc(z));
    end

% F=zeros(size(q));
%
% for i=1:(N+1)
%     z=(i-1)*h;
%     F(2*i-1)=h*f_xs(z,T);
% end
nose = 10;

F=zeros(size(q));

for n=0:N
    sum1=0;
    sum2=0;

    if n>0
        for sei=-(nose-1):-1
            z=h/nose*sei;
            alfa=3*(z+h)^2/h^2-2*(z+h)^3/h^3;
            beta=-(z+h)^2/h+(z+h)^3/h^2;
            f = f_xs(z+n*h,T);
            sum1=alfa*f;
            sum2=beta*f;
        end
    end

    if n<N
        for sei=0:(nose-1)
            z=h/nose*sei;
            alfa=1-3*z^2/h^2+2*z^3/h^3;
            beta=z-2*z^2/h+z^3/h^2;
            f = f_xs(z+n*h,T);
            sum1=alfa*f;
```

```
            sum2=beta*f;
        end
    end

    F(2*n+1)=h/nose*sum1;
    F(2*n+2)=h/nose*sum2;

end


G3=zeros(size(q_dot));

G3(1) = (q_dot(1)-Uc)*abs(q_dot(1)-Uc)*D2;

G3(end-1) = q_dot(end-1)*abs(q_dot(end-1))*d2;



q_dotdot=M^(-1)*((-D+G1)*q_dot+(-K+G2)*q+F-G3+G4*q_int);

[G1(1)*q_dot(1)  G2(1)*q(1) G4(1)*q_int(1) F(1) G3(1) -D(1,:)*q_dot -K(1,:)*q
% % T

y_dot = zeros(3*(N+1)*2,1);

y_dot(1:(N+1)*2)=q;
y_dot((N+1)*2+1:(N+1)*4)=q_dot;
y_dot((N+1)*4+1:end)=q_dotdot;


end
```

8.3 baat-test.m

```
function baat_test()
global m D1 D2 Uc P I D

m=9.6*10^7;
D1=0.9*10^6;
```

```
D2=1*10^6;
Uc=2;

P = -8*10^7;
I = -1*10^7;
D = -5*10^7;

y0=[0;0;0];

[T,y]=ode45('baat_system_test', [0 100], y0);


figure
plot(T,y(:,2),'-or');
```

8.4 baat-system-test.m

```
function [y_dot] = baat_system_test(t,y)

global m D1 D2 Uc P I D

x_int=y(1);
x=y(2);
x_dot=y(3);


paadrag = P*x + I*x_int + D*x_dot;

x_dotdot=m^(-1)*(paadrag-D1*x_dot-D2*(x_dot-Uc)*abs(x_dot-Uc));

y_dot=[x; x_dot; x_dotdot];
```

8.5 system-matrix.mw

```
restart:

>alpha := array(1 .. 4);
>alpha[1] := 1-3*(x/h)^2+2*(x/h)^3;
alpha[2] := h*(x/h-2*(x/h)^2+(x/h)^3);
alpha[3] := 3*(x/h)^2-2*(x/h)^3;
alpha[4] := h*(-(x/h)^2+(x/h)^3);

>‘&alpha;d‘ := map(diff, alpha, x);

>with(LinearAlgebra);
>‘&alpha;&alpha;‘ := Multiply[Z](Transpose(Vector(alpha)), Vector(alpha));

>subs(x = 0, ‘&alpha;&alpha;‘);
>subs(x = h, ‘&alpha;&alpha;‘);

>‘&alpha;dd‘ := map(diff, ‘&alpha;d‘, x);
>map(int, ‘&alpha;&alpha;‘, x = 0 .. h);

>‘&alpha;dd&alpha;dd‘ := Multiply[Z](Transpose(Vector(‘&alpha;dd‘)),
Vector(‘&alpha;dd‘));

>map(int, ‘&alpha;dd&alpha;dd‘, x = 0 .. h);

>‘&alpha;d&alpha;d‘ := Multiply[Z](Transpose(Vector(‘&alpha;d‘)),
Vector(‘&alpha;d‘));

>map(int, ‘&alpha;d&alpha;d‘, x = 0 .. h);
```

List of Figures