ISSN 0280-5316
ISRN LUTFD2/TFRT--5712--SE

Vision-Based Control of the Overhead Crane

Roger Nilsson
Mikael Nordgren

Department of Automatic Control
Lund Institute of Technology
September 2003

Department of Automatic Control
Lund Institute of Technology
Box 118
SE-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
September 2003

Document Number
ISRN LUTFD2/TFRT--5712--SE

Author(s)
Roger Nilsson and Mikael Nordgren

Supervisor
Anders Robertsson and Tomas Olsson at the department Automatic Control, LTH.

Sponsoring organization

Title and subtitle
Vision-Based Control of the Overhead Crane. (Datorseendebaserad reglering av travers med hängande last).

Abstract
Through the years, cranes have been used extensively for many different tasks. For example construction work, shipbuilding and cargo transportation. One of the tedious tasks that cranes perform is loading and unloading containers. This is a process which can be optimized with the aid of control theory.

While moving cargo, the crane operator must make sure that crane motion does not cause the cargo to accelerate too quickly, as the cargo may then start to oscillate. Such motion may disrupt unsecured cargo and put unnecessary stress on secured cargo. Today the crane movement is directly controlled by the crane operator; this however severely limits the crane speed due to human inability to control the crane optimally. This thesis presents and investigates a method of controlling the crane movement. The method utilizes computer vision to determine cargo position which is used to control crane movement optimally.

Keywords
Pendulum; Harris Corner Detection; Vision; FFT; Linear Quadratic Control

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
37

Security classification

Recipient's notes

The report may be ordered from the Department of Automatic Control or borrowed through:
University Library, Box 3, SE-221 00 Lund, Sweden Fax +46 46 222 44 22

Lund Institute of Technology
Lund University

Master of Science Thesis

Vision-Based Control of the Overhead Crane

Roger Nilsson
Mikael Nordgren

September 2003

Master of Science in Computer Science Engineering

Supervisor:	Anders Robertsson, Lund Institute of Technology
Co-Supervisor:	Tomas Olsson, Lund Institute of Technology
Examiner:	Anders Robertsson, Lund Institute of Technology
Opponents:	Pontus Nordfeldt, Lund Institute of Technology
	Henrik Segerbäck, Lund Institute of Technology

Acknowledgments

This work has been accomplished at the Department of Automatic Control at Lund Institute of Technology. Initially, we would like to express our gratitude to our supervisor Anders Robertsson for support, encouragement and information enabling us to carry out this Master of Science Thesis. We would also like to thank Tomas Olsson for his assistance in this work. For proof reading and making sure the language in this thesis is adequate we thank Johannes Admasie and Joakim Roubert. For everlasting support Mikael would also like to thank Sharon Tsai.

Contents

Chapter 1

Introduction

Cranes are used in many different environments for varying tasks, thus there are many interesting problems to be handled. This thesis however concentrates on the controlled movement of a cargo container. As in most tasks there is economic advantage to be gained by enhancing performance, in this specific task the main performance issue is movement speed. In order for the crane to move at optimum speed and still maintain safety both for cargo and surroundings a crane operator needs advanced technology to help with the task.

When moving cargo the operator must make sure not to accelerate too much as this will cause the cargo to sway in a pendulum like motion. Such motion may cause serious damage to cargo and surroundings. In order to accelerate in a reasonable manner a controller is needed. This controller should help the operator by limiting crane acceleration, while still leaving the operator in control. It is possible to construct such a controller by calculating the highest permissable acceleration from previous control signals. However, that is a solution of limited usefulness since the system needs to exactly know the cargo weight. Besides, it is well known that feedback is one of the corner stones of automation. In order to determine cargo position some form of sensor is needed. Thoughts were given to use accelerometers but were discarded because of the difficulty in eliminating drift. The next alternative is to use digital cameras and computer vision to determine cargo location.

The idea would be to use a single digital camera to capture images and have a computer analyze them, in real time, to locate the cargo. The cargo coordinates would then be passed on to the controller which would limit acceleration. The task is complex since size differs as well as background. Computer vision is still evolving and this thesis tries both to simulate real world conditions and to adapt to constraints of computer vision limitations.

The goal of this Master of Science Thesis is to investigate the possibility to use computer vision to automate the process of moving cargo. The controller should be able to move cargo from one coordinate to another as fast as possible and with minimal amount of sway.

In the following chapters the implementation of this program is presented, and the outline of the thesis is as follows:

Chapter 2 describes the robot environment and gives background information about the problem.

Chapter 3 gives an overview of the work that has been done in this thesis and presents the manner in which it has been carried out.

Chapter 4 presents the implementation of the vision program and the accompanying control program.

Chapter 5 contains the results and limitations of the work that has been done.

Chapter 6 presents the summarized conclusions of this thesis and ideas for further work on this subject.

Chapter 2

Robot

In this chapter the robot used is described to give a more complete background to the work accounted for in this thesis.

2.1 Scaling the Problem

To ease development and minimize cost the problem of controlling huge cranes moving large containers, is simplified to controlling an industrial robot moving a small box.

Figure 2.1: The robot used–IRB2000.

2.2 ABB Industrial Robot

In the robotics lab, provided by the Department of Automatic Control, an ABB industrial robot is placed. The robot, an IRB2000, has 6 joints and is highly maneuverable. It is constrained in a secure environment[1] to protect operators from injury.

[1] A cage with electronic doors that shutdown the robot if opened.

Control of the robot is performed by three different computers; master, slave and supervisor. The master gives the actual signals to the robot, the slave listens for trajectories[2] or other control information and tells the master how it should move and the supervisor makes sure that robot acceleration and spatial coordinates are within safety limits. To further increase safety there is a collision protection system installed between the gripper of the robot and the closest joint. It is a pneumatic system that withstands a limited amount of force before it engages and shuts down the robot.

[2]What a trajectory is and how it is generated is described in Appendix A.

Chapter 3

Method

This chapter presents an overview of the work done in this thesis. The work has been divided into various steps, and different methods and algorithms have been tested for each part.

3.1 The Problem

The main goals of this thesis are to stabilize cargo moved with an overhead crane and enable it to follow trajectories using computer vision.

As previously discussed it is a complex problem demanding both control theory and computer vision theory.

3.2 Phases

To handle the difficulties of this problem, the following phases of the work were chosen:

3.2.1 Locating the Cargo

The basis of solving the problem is locating the cargo. A computer vision system that can detect and track the cargo in real time must be developed.

While performing the necessary image analysis the system should filter the coordinates to optimize tracking and tolerate certain erroneous images.

Here a firewire[1] camera mounted on the robot hand will take pictures of the cargo to be analyzed by the vision software.

[1] An external bus, to communicate with peripherals, also known as IEEE 1394.

3.2.2 Controller

When the coordinates of the cargo are determined, the controller can utilize them to control the robot. A controller that can handle the dynamics of robot and cargo as well as move the cargo optimally should be developed.

In order to create such a controller many system parameters need to be determined through practical experiments.

3.2.3 Moving the robot

In order to actually move the robot the controller and vision system need to be tied together. A communication channel to the actual robot must be established and control signals converted into trajectories[2] that the robot can understand.

3.3 Flow Chart

The firewire camera grabs an image which is passed on to the vision program. The coordinates of the cargo are determined by the vision program which hands them over to the controller. After the controller has calculated the control signals they are used for generating trajectories to move the robot. The schedule then repeats.

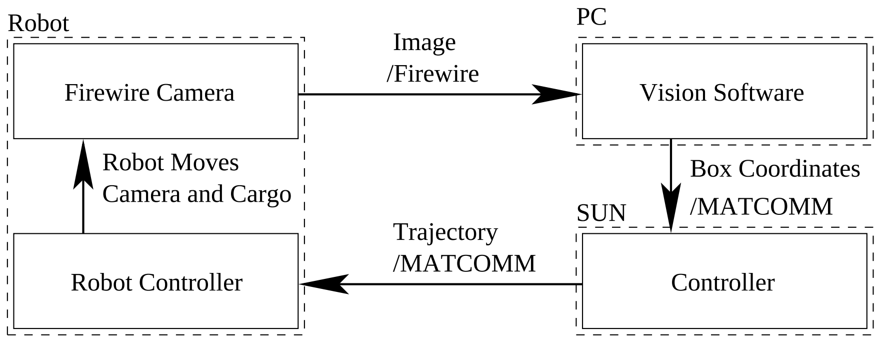

Figure 3.1: Flow chart.

3.4 Hardware-Software Environment

The system has been developed in a mixed environment. Visual Studio on a IBM PC, running Windows, has been used together with Unibrain firewire soft and hardware to implement the vision system. MATLAB on a SUN workstation, running Solaris, has been used for controller implementation and trajectory generation. A library called MATCOMM, developed

[2]Explained in Appendix A and Chapter 4.

by Department of Automatic Control has been used for communication between the different computers and the robot. Naturally, various other programs and utilities were used throughout development like text editors and imaging tools.

Chapter 4

Implementation

This chapter gives a detailed description of system implementation, hardware and software.

As described in Chapter 3, a firewire camera has been used. Every image captured was analyzed with various algorithms to determine where the coordinates of the cargo were located. A Kalman filter was applied to track cargo movement and compensate for any error in image analysis. The coordinates were then transferred to the controller which calculated the correct control signals subsequently to be issued to the robot.

4.1 Environment

This section will describe the environment used for development in great detail.

4.1.1 Software

Different software tools have been used throughout development in either windows or unix environment. The main tools consist of:

Visual studio. A software development tool by Microsoft mainly focused on C++. The version used in this thesis is 6.0. It has been used only for development of the vision software.

FireI. A library from the company Unibrain. Together with their firewire hardware it provided easy control of the camera and receive images. The version used in this thesis was 1.61. It was only used by the vision software which depend on it for receiving images.

MATLAB. An advanced math and simulation tool by Mathworks. It was the basis for all simulation and testing as well as controller development. The version used in this thesis was 6.0. It has been used extensively throughout development of the system.

Matcomm. A library developed at the Department of Automation and Control. It was used by both the vision software and the controller in order to communicate with each other and the robot over a TCP/IP network.

4.1.2 Hardware

The system consists of different hardware. Some hardware has been used only for development purposes but mainly everything is used in the final system.

IBM PC. A few different PCs running Windows have been used throughout development. The PC running the vision software in the final system uses Windows 2000.

SUN workstation. Many different workstations has been used. The system acting as a controller and trajectory generator is a Sparc Ultra 10 running Solaris.

Sony DFW-V300. A firewire camera capable of taking high quality images at 30 fps. It is used by the vision software.

Firewire. Unibrain firewire cards have been used together with a generic firewire repeater to communicate with the camera.

Robot. An industrial robot by ABB, described in Chapter 2, acts as the crane in the system. It uses specially modified hardware to enable fast control of the robot.

4.2 Vision Software

The main task of the vision software is to locate and track the cargo. In this thesis the cargo is represented by a colored box as seen in Fig. 4.1. Different approaches were considered but after short consultation with the supervisor the choice was made to try to locate the corners of the box and use those 4 pairs of coordinates to control the system.

The first idea was to simply analyze each image to find all the corners and then figure out which ones where from the box. The system would then repeat. In order to perform this task in real time, however, dedicated DSP[1] hardware would be needed. A system using less computational effort was then conceived.

The main parts of the vision software is described in the following subsections.

[1]Digital Signal Processing.

4.2.1 Initial corner detection

An algorithm called "Harris corner detection", described in [3], was used to locate all the corners in an image. The algorithm works by multiplying the image derivatives in x and y axis. This results in three matrices X^2, Y^2 and XY. Those derivatives are then filtered with a gaussian filter and a corner "strength" image generated by the following per element formula.

$$\frac{(X_{ij}^2 \cdot Y_{ij}^2 - XY_{ij} \cdot XY_{ij})}{(X_{ij}^2 + Y_{ij}^2 + \epsilon)}$$

The ϵ in the formula is to protect from division by zero. The corner "strength" image is then dilated, meaning that local maxima replaces the surrounding values and then compared with the original "strength" image. The points that are equal, meaning they are a local maxima, and also greater than a threshold value are corners. The algorithm is illustrated in sequence in Fig. 4.2.

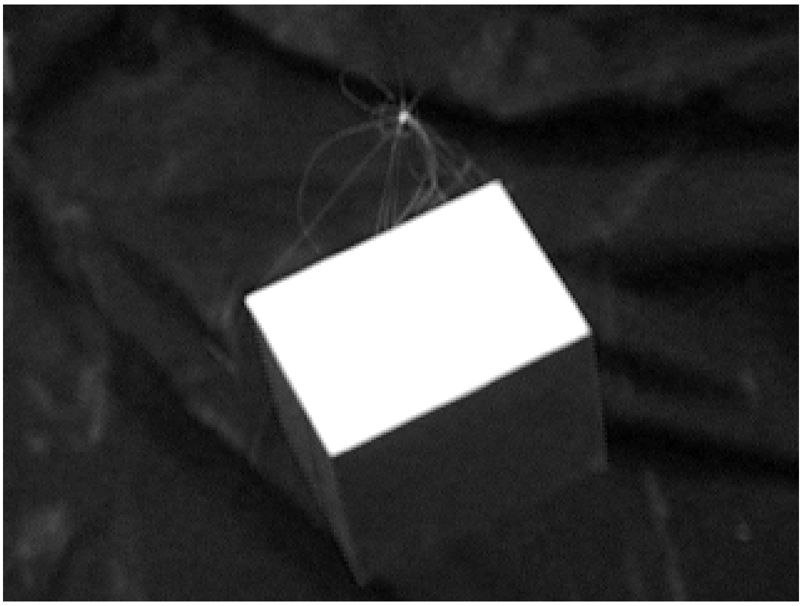

Figure 4.1: The box used to simulate cargo.

4.2.2 Locating the Corners that Represent the Box

In order to locate the four important corners from the image, which might contain several corners, an algorithm is needed. Various fast algorithms trying to sort the corners in special ways where tried, but none found successful. These were based around painting the center of the box in a special

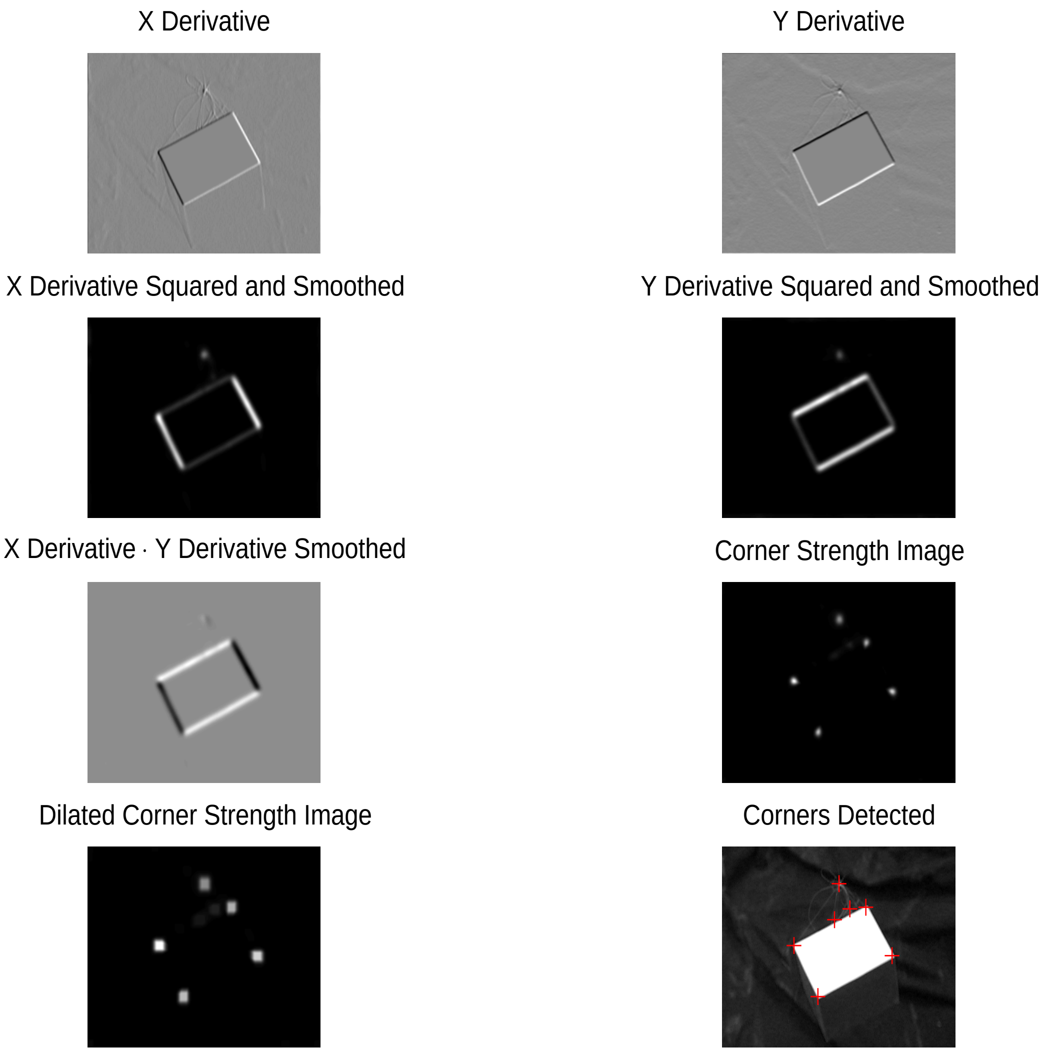

Figure 4.2: The steps of the corner detection algorithm.

color and then determining the center of the box in the image. Using the known center all corners were then sorted by distance, since all corners are equidistant they should be in order. As previously mentioned all algorithms based around that assumption failed, mainly because extra corners detected might have similar distance to the center of the box as the real corners, thus disrupting the order.

The other option at hand was to iterate through all corners to find the box. In order to select the four corners a measurement for how close a group of four corners are to aligning as a box was needed. This was performed by calculating a value for how parallel the edges between the corners are and how close they are to fulfilling Pythagoras theorem. The size of the box is checked as well in order to avoid small clusters of corners from disrupting the algorithm. The group that fulfilled these criteria the best were chosen as the box. The value for checking how parallel the lines were was calculated by adding the absolute values of the difference of lengths of the opposing edges in x and y axis. The value for checking Pythagoras theorem was calculated by subtracting the actual length of the diagonal from the reconstructed length.

4.2.3 Tracking the Corners

As previously discussed, performing full image corner detection was much too slow. One of the main operations in the corner detection algorithm is two dimensional convolution, and it is also the step consuming the majority of computation resources. To try speed up the convolution, fast fourier transformation (FFT) was considered, since in the frequency domain normal convolution is performed by simple multiplication. The problem with this approach is that the two images being convoluted would need to be zero padded to final convolution size, then transformed and subsequently per element multiplication has to be done. This operation may require more computation than the direct approach should one of the images be considerably smaller than the other, as is the case in this thesis. The FFT algorithm also looses efficiency if the size of the input is not a power of two, which again is the case in this thesis. The workings and efficiency of the FFT algorithm is explained in [8] and [7].

MATLAB experiments, using the same image and filter size as this thesis, proved that direct convolution was faster in almost every case. Together with the fact that direct convolution also was easier to implement that method was chosen for this thesis. To optimize performance only small areas, corner windows, from the image were used. After determining the initial four corners these windows were used to locate the same corner in the next image, thus tracking it. A Kalman filter was applied to the coordinates to maximize the use of the corner windows.

To increase algorithm stability, methods to recover from losing corners were conceived. They enable the program to reconstruct one or even two corners[2] if the vision software somehow lost them. The integrity of the box is also checked to protect from foreign objects ”stealing” the corners. This means that a foreign object may have a corner located above or near the true corner. The algorithm might then be tricked into believing that the foreign object holds the true corner and track that corner instead. However, the other corners would sooner or later move away from that object and the integrity check would fail. When this occurs the program will then reconstruct the corner most likely to not be apart of the box and thus start tracking the real one again.

[2] In order to reconstruct two corners they must not be opposing each other.

4.3 Controller

A controller that has servo and regulation properties was constructed for the system. This section contains how the physical system was mathematically modelled and describes the controller used. System identification and a simulation was done to verify that the theory works.

4.3.1 Model

The system modelled consists of an overhead crane travelling in a plane perpendicular to the gravity vector dragging a container in a wire device beneath. The model of the system that was tried out was to view the container hanging from the crane as a spherical pendulum. In order to derive the differential equations needed Lagrange mechanics was used[3]. In the first subsection the spherical pendulum is separated into two decoupled planar pendulums. The second subsection derives the differential equations for a planar pendulum attached to an overhead crane.

Spherical Pendulum

In Fig. 4.3 a spherical pendulum has been projected onto the x-z and y-z plane. These two projection is two coupled planar pendulums but as long

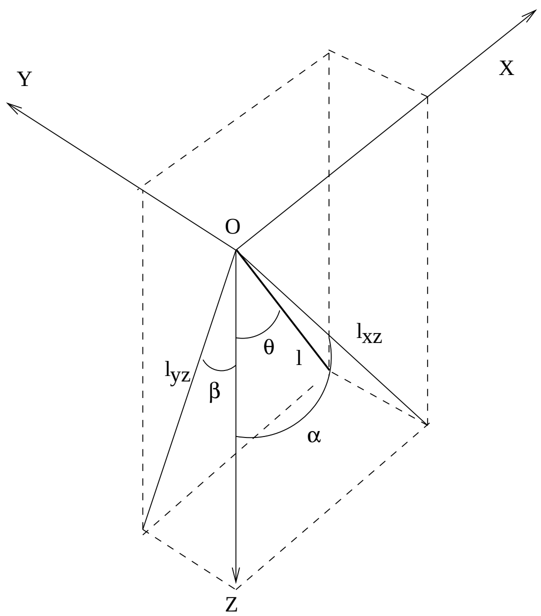

Figure 4.3: Spherical pendulum.

[3]Lagrange Mechanics can be studied in [5] and [6].

as θ is small they can be viewed as two decoupled planar pendulums[4].

Lagrange mechanics is going to be used in order to separate the system into two decoupled planar pendulums. By defining $q = [\alpha, x, \beta, y]^T$ as generalized coordinates of the spherical pendulum system the kinetic energy can be separated into two parts T_{xz} and T_{yz}, where T_{xz} is the kinetic energy projected onto the x-z plane and T_{yz} is the part projected onto the y-z plane. The potential energy is a function of θ and can not be separated and projected onto the planes. The length of the projected pendulum on the x-z plane and y-z plane is

$$l_{xz} = \frac{1}{1 + \frac{\cos^2\alpha}{\cos^2\beta}\sin^2\beta} l$$

$$l_{yz} = \frac{1}{1 + \frac{\cos^2\beta}{\cos^2\alpha}\sin^2\alpha} l.$$

If the angles α and β remain small[5] the approximations $\sin\alpha = \sin\beta = 0$ and $\cos\alpha = \cos\beta = 1$ can be used. l_{xz} and l_{yz} are then approximately l. This means that θ does not have any influence over the length of the projection. The potential energy projected is then a function of α for the x-z plane pendulum and β for the y-z plane pendulum.

Under these assumptions the kinetic energy and the potential energy of the spherical pendulum can be separated and projected onto two orthogonal planes. This results in two decoupled planar pendulums which can be controlled separately.

Planar Pendulum

A planar pendulum attached to an overhead crane can be viewed in Fig. 4.4. The system consists of a pendulum bob of mass m and a cart of mass M. The main interest is the differential equations describing the position of the bob thus the Lagrangian for the system has to be constructed. By using X and θ as generalized coordinates the following is derived. The kinetic energy of the pendulum bob is

$$T_{bob} = \frac{M}{2}\dot{X}^2$$

and the kinetic energy of the cart is

$$T_{cart} = \frac{m}{2}(\dot{X}^2 + 2l\dot{X}\dot{\theta}\cos\theta + l^2\dot{\theta}^2).$$

The potential energy of the pendulum bob and the whole system is

$$V = mgl(1 - \cos\theta).$$

[4]The following proof of this is based on [10], p. 6-9.
[5]$\pm 10^o$.

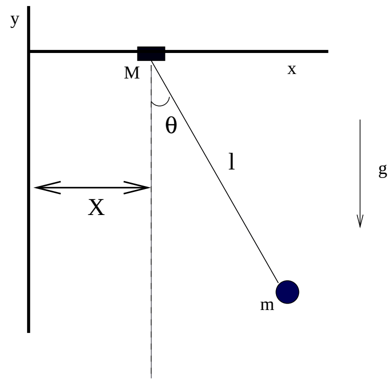

Figure 4.4: Pendulum hanging from an overhead crane.

The lagrangian is $T_{tot} - V$.

$$\mathcal{L} = T - V = \frac{M}{2}\dot{X}^2 + \frac{m}{2}(\dot{X}^2 + 2l\dot{X}\dot{\theta}\cos\theta + l^2\dot{\theta}^2) - mgl(1-\cos\theta)$$

The Euler-Lagrange equation for θ is

$$\frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{\theta}} - \frac{\partial\mathcal{L}}{\partial\theta} = 0$$

which simplifies to

$$l\ddot{X}\cos\theta + l^2\ddot{\theta} + gl\sin\theta = 0. \tag{4.1}$$

This is the needed differential equation describing the angle of the planar pendulum. This equation is not linear but the assumption that the angle will remain small[6] the following approximation $\sin\theta \approx \theta$ and $\cos\theta \approx 1$ will then be valid. The linearized version of Equation 4.1 is

$$l\ddot{X} + l^2\ddot{\theta} + gl\theta = 0. \tag{4.2}$$

By taking the laplace transform of Equation 4.2 the transfer function from position of the overhead crane to the angle can be derived as

$$\frac{\theta(s)}{X(s)} = -\frac{1}{l}\frac{s^2}{s^2 + \frac{g}{l}}. \tag{4.3}$$

State-Space Representation

By using: $x_1 = \theta$, $x_2 = \dot{x}_1$, $x_3 = X_{overheadcrane}$, $x_4 = \dot{x}_3$ as states and introduce the acceleration of the overhead crane as input signal the resulting system will be that of Fig. 4.5. An expression for $\dot{x}_2$ is needed. From

[6]This should be the case if proper precautions are taken during controller design.

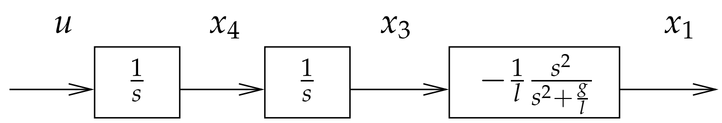

Figure 4.5: The model over the plant.

Fig. 4.5 the following relationship is obtained.

$$-x_3 \frac{s^2}{ls^2 + g} = x_1$$

From this equation $\dot{x}_2$ can be deduced as

$$\dot{x}_2 = -\frac{g}{l}x_1 - \frac{1}{l}u.$$

Now we can set up the system in state-space form as:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{-g}{l} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{-1}{l} \\ 0 \\ 1 \end{pmatrix} u$$

$$y = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}.$$

The poles of the system are located at $\pm\frac{\sqrt{-lg}}{l}$ and a double pole in 0.

4.3.2 System Identification

The transfer function from the position of the overhead crane to the angle of the attached pendulum[7] is

$$\frac{\theta(s)}{X(s)} = -\frac{1}{l}\frac{s^2}{s^2 + \frac{g}{l}}. \tag{4.4}$$

The parameter $\frac{g}{l}$ has to be identified. This was done by simulating a step. The simulation of a step was done by moving the overhead crane very fast to a new position. The plot of the step response can be found in Fig. 4.6. The upper diagram shows the position of the crane and the lower diagram shows the position of the cargo hanging 0.9m below the crane. The pendulum oscillated with a period time of 1.9s which gave an angular frequency of $\omega = 2\pi/T$. Equation 4.4 tells that ω equals $\sqrt{g/l}$ and the identification of the crucial parameter was completed.

[7]Where the pendulum bob represents the container hanging below the crane.

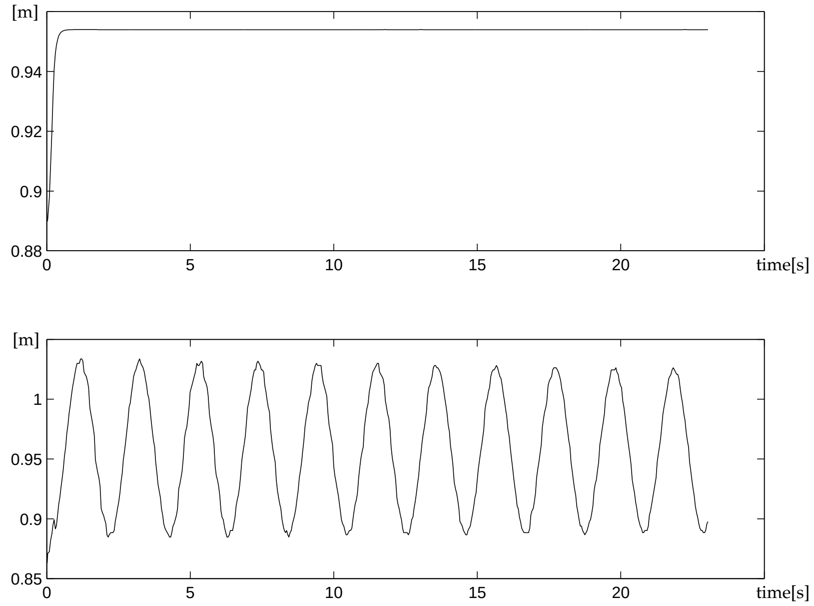

Figure 4.6: Step response, upper diagram shows pos. of the crane and the lower diagram pos. of the cargo.

4.3.3 Two Degree of Freedom Controller

It is highly desirable that the system has both servo and regulation properties and therefore a two degree of freedom structure on the controller was used. The structure of the controller can be found in Fig. 4.7. The state feedback is used to eliminate process disturbances and model errors. The feedforward is used to incorporate the servo property into the system. The task is to find the $G_{ff}(s)$ transfer function that makes the system behave like $G_m(s)$. $G_{ff}(s)$ equals $G_m(s)/G_{plant}(s)$ hence the model can not be chosen arbitrary. The pole excess of the model must match the plant and all unstable zeros in open-loop must be present in the model. In a SISO[8] system where $G_m(s)$ has the same order and same zeros as $G_{plant}(s)$ it is very easy to calculate $G_{ff}(s)$. If $G_m(s) = \lambda B_m/A_m$ then $G_{ff}(s)$ equals $\lambda A_{plant}/A_{model}$.[9]

The setpoint generation for the state feedback loop has to be solved. The method used was to drive the feedforward through a model of the plant and use the states of the model as setpoints.

[8]Single input single output.
[9]The two degree of freedom controller is further described in [2], p. 150-152.

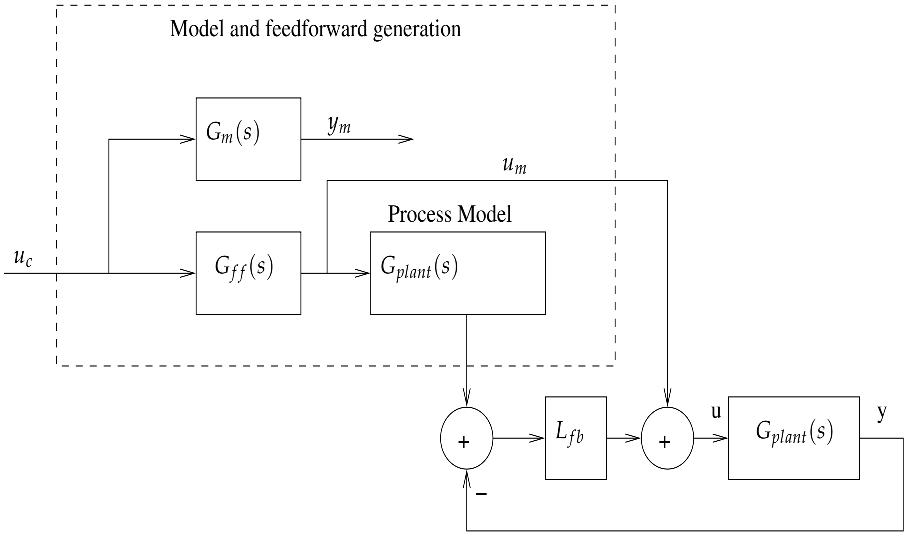

Figure 4.7: Two degree of freedom controller with a linear response model.

Trajectory Generation

The following model is used

$$G_m(s) = \lambda \frac{B_{plant}(s)}{(s+\omega_m)^4}.$$

with λ chosen so that $G_m(0) = 1$. It is the position of the container[10] a trajectory is going to be generated for and not the crane position or the angle and therefore the transfer function from acceleration of the cart to the position of the container has to be deduced. The following transfer functions are known.

$$G_{cart}(s) = \frac{X_{cart}(s)}{U(s)} = \frac{1}{s^2}$$

$$G_{angle}(s) = \frac{\theta(s)}{U(s)} = \frac{-1.115}{s^2 + 10.94}$$

Then the position of the container is

$$G_{container}(s) = \frac{X_{container}(s)}{U(s)} = G_{cart}(s) + l \cdot \sin(G_{angle}(s)).$$

For small angles this can be approximated with

$$G_{container}(s) = \frac{X_{container}(s)}{U(s)} = G_{cart}(s) + l \cdot G_{angle}(s).$$

Now when the open-loop transfer function for the container position is known the theory for the two degree of freedom controller is applicable.

[10] The pendulum bob.

Non Linear Response Model

The feedforward signal can not be generated by driving the reference signal through a linear filter if the desired response model is non linear. In this thesis a non linear response model is highly desirable because a real crane has an upper limit when it comes to acceleration. If cargo is to be moved time optimal between two points a non linear response model is needed. The solution to this problem is to use the structure seen in Fig. 4.8. This calls for an implementation of the inverse process in the controller.

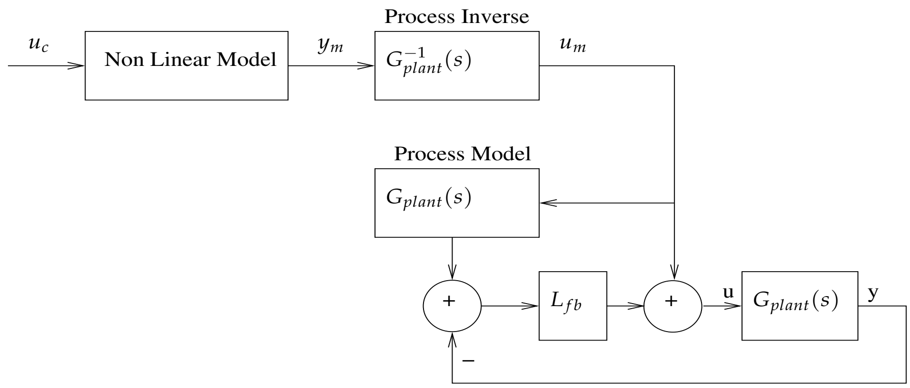

Figure 4.8: Two degree of freedom controller with a non linear response model.

4.3.4 Simulation

The plant simulated is the one shown in Fig. 4.5 save the fact that the pendulum block has been replaced with a non linear block describing the pendulum with the same angular frequency as our real system identified in section 4.3.2. The feedback matrix is an LQ controller with the weight matrices Q and R chosen as: Q=diag(1,1,1,1), R=1[11]. The system is simulated with $\omega_m = 2$ in MATLAB.

Results

The results can be viewed in Fig. 4.9. Model output is the resulting trajectory from the model when a step is applied. The position of the container is found in the upper right diagram. The control signal is the acceleration of the overhead crane. The angle denoted θ in Fig. 4.4 is shown in the middle right diagram. In the error plot the deviation from the trajectory can be studied.

[11] Linear Quadratic control can be studied in [2], p. 408-429.

Comments

The simulation shows that the method described in Section 4.3.3 works very well in theory. The control signal is not too large and the angle is never greater than one degree.

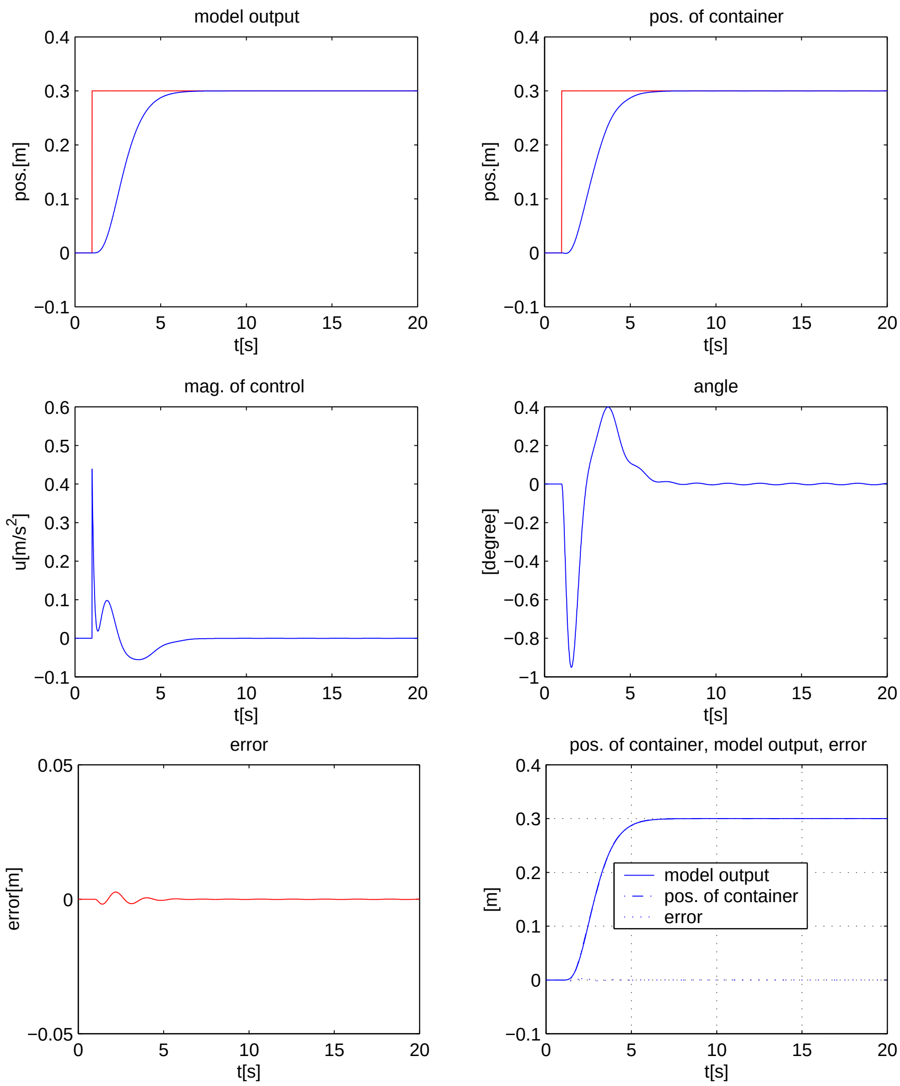

Figure 4.9: Step response from model and simulated output from the controlled plant, $\omega_m = 2$.

Chapter 5

Experiments

This chapter describes how the system has been tested to validate that the main goal of this thesis, described in Section 3.1, has been accomplished. The system's performance and limitations are also discussed.

5.1 Closed-Loop System

5.1.1 Control Procedure

1. *Identification of physical parameters.* System identification of the physical parameters in the transfer functions has been done as described in Section 4.3.1.

2. *State observation.* The position of the overhead crane was directly measurable. The velocity of the crane was obtained by using backward difference. The angle from the crane to the pendulum bob was obtained by visual tracking and a Kalman filter.

3. *Control.* To control the plant an LQ regulator was used and the weight matrix Q and R were as follows: Q=diag(1,1,1,1), R=1. The poles of the closed-loop system are:

   $$-0.5555 \pm 3.2586\jmath, -0.8650 \pm 0.5025\jmath.$$

   The control law yields the acceleration of the overhead crane and the calculation of the trajectory to send to the robot is described in Appendix A. The dynamics of the robot is considered negligible and therefore not present in the control law.

5.1.2 Results

The system has been run for an extensive time and no signs of instability have been detected. A step response from the closed-loop system can be viewed in Fig. 5.1. The upper right diagram shows the position of the overhead crane and the lower right diagram shows the angle θ[1].

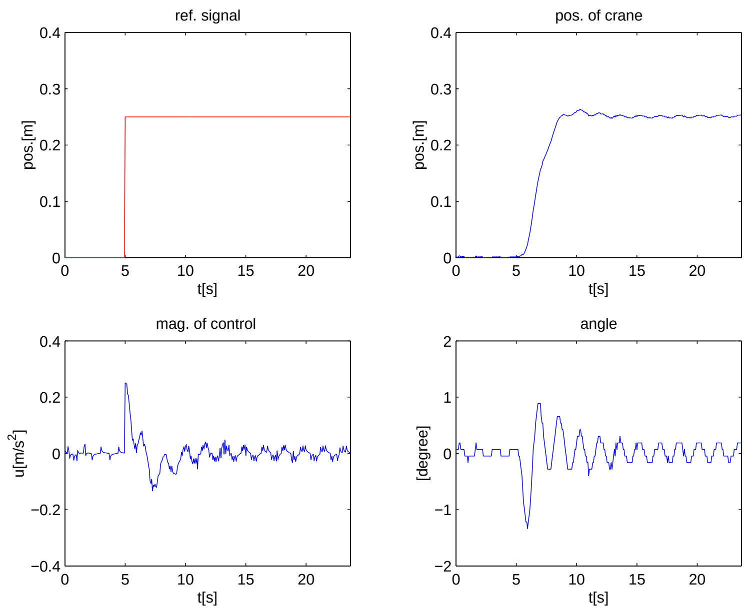

Figure 5.1: A 0.25m step to the position reference was introduced after 5 seconds. An LQ regulator was used to stabilize the system.

5.1.3 Comments

During testing some problems arose but were corrected and a few comments on the result is in place.

1. There is a quantification effect seen in the lower right diagram in Fig. 5.1. It stems from that a pixel in the capture from the camera corresponds to 1.8385mm in the x-y plane where the box is. The angle is then calculated with this information and the knowledge that the box is 0.9m from the camera.

2. Due to the delay inherited when the image is analyzed with the vision software the prediction from the Kalman filter is used in the control law.

5.2 Two Degree of Freedom Controller

The configuration of the system is the same as in Section 5.1 save the fact that the controller has been replaced with the two degree of freedom con-

[1]The angle θ from Fig. 4.4.

troller described in Section 4.3.3.

5.2.1 Results

Testing the two degree of freedom controller involved two tests. The linear response model has $\omega_m = 2$ and a static gain of one in both tests.

1. A 0.3m step in the x-axis position reference was introduced after 6 seconds and a 0.3m step in the y-axis position reference was introduced after 7 seconds.

2. A sinus signal with an amplitude of 0.5 and a frequency of 1.57 rad/s was added to the x-axis position reference after 5 seconds and a 0.3m step in the y-axis position reference was introduced after 7 seconds.

The plots for the step-step run can be viewed in Fig. 5.3 and Fig. 5.4. The plots for the second run can be found in Fig. 5.5 and Fig. 5.6. The plots in the figures is explained in Section 4.3.4.

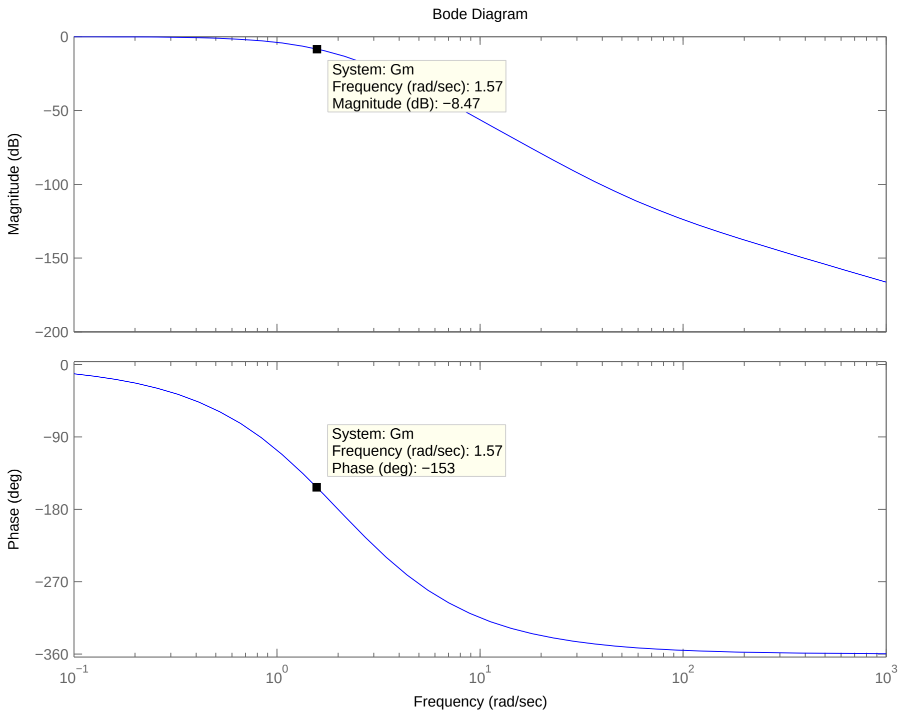

Figure 5.2: Bode diagram of the linear response model. The frequency of the sinus signal introduced in test two has been marked.

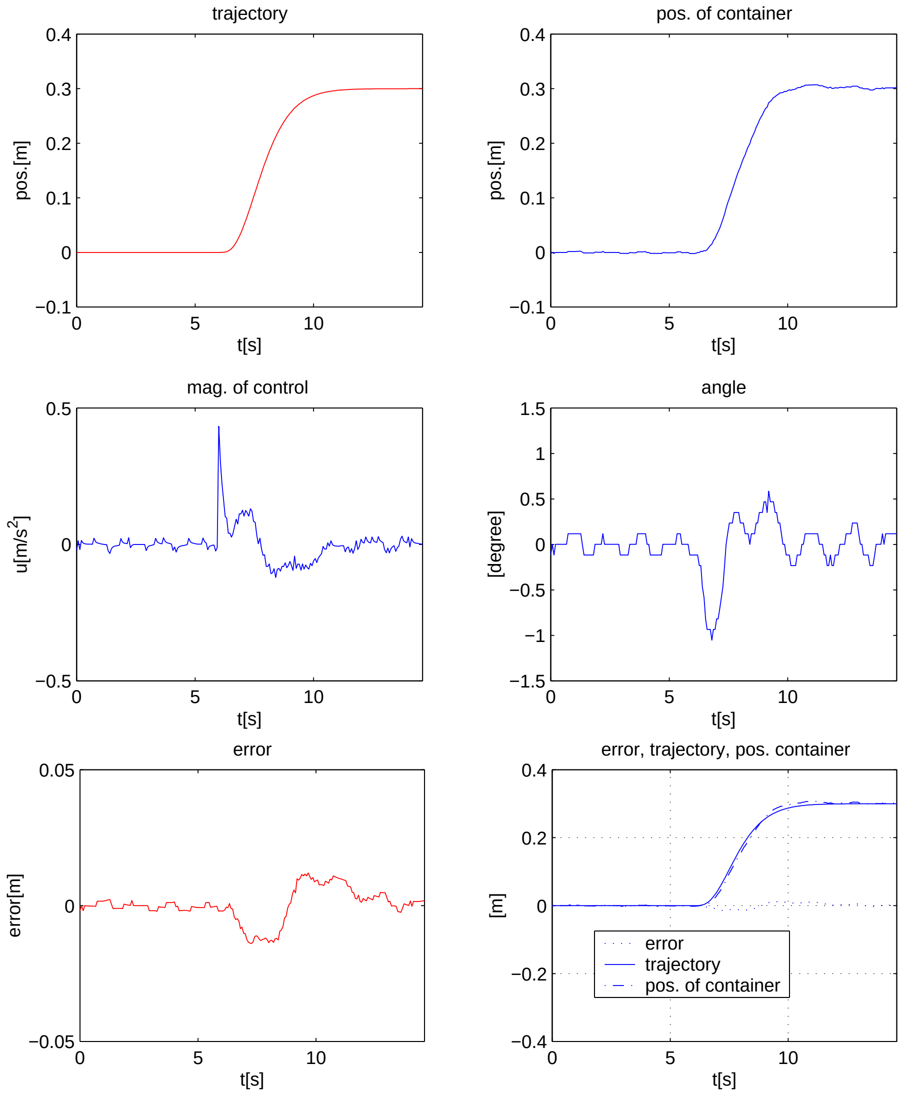

Figure 5.3: Test number one, x-axis response. A 0.3m step in the position reference was introduced after 6 seconds. The two degree of freedom controller was used with $\omega_m = 2$.

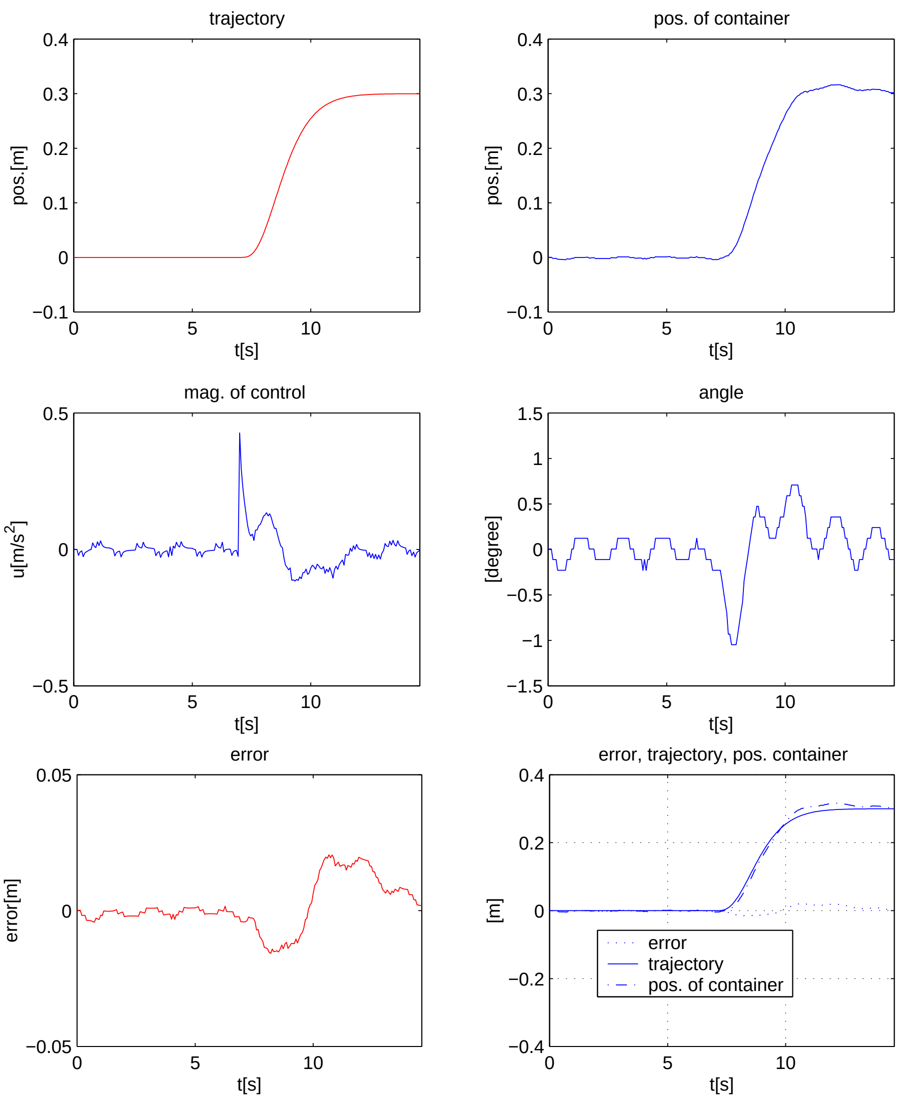

Figure 5.4: Test number one, y-axis response. A 0.3m step in the position reference was introduced after 7 seconds. The two degree of freedom controller was used with $\omega_m = 2$.

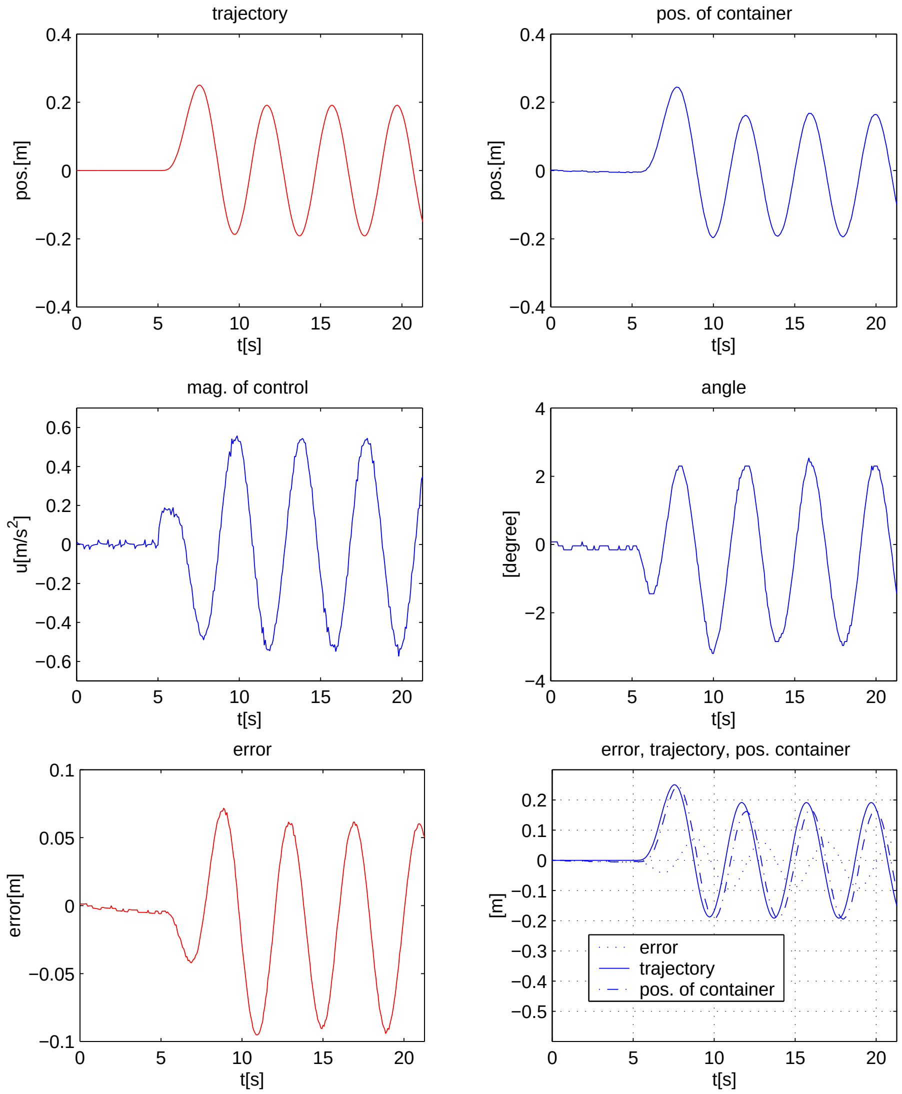

Figure 5.5: Test number two, x-axis response. A sinus signal with amplitude 0.5 was added to the position reference after 5 seconds. The two degree of freedom controller was used with $\omega_m = 2$.

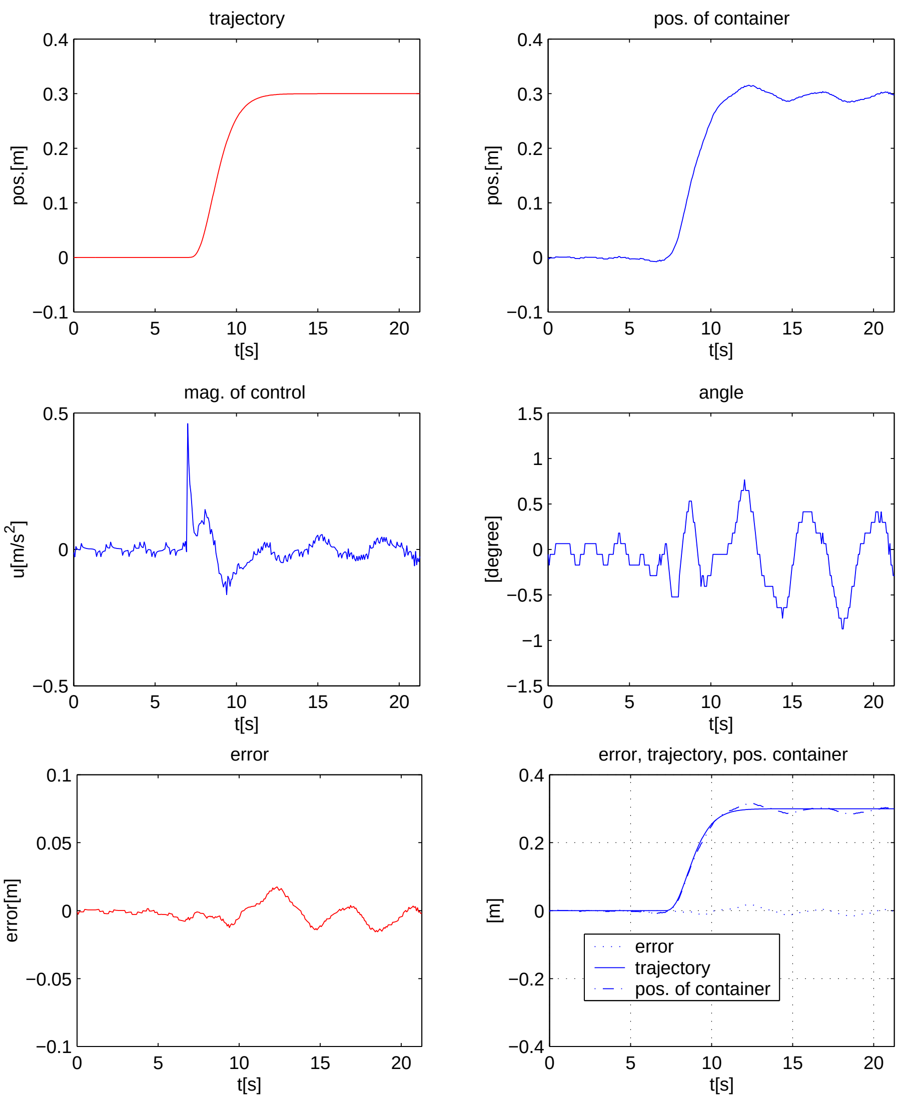

Figure 5.6: Test number two, y-axis response. A 0.3m step in the position reference was introduced after 7 seconds. The two degree of freedom controller was used with $\omega_m = 2$.

5.2.2 Comments

These two tests directly show that the goal of the thesis has been reached. By looking at the first run and compare the result with the simulation in Fig. 4.7 it is clear that the system works well and it follows the trajectory generated by the model. In the second run when a sinus reference is used in the x-axis it is clear from Fig. 5.5 that model errors are not eliminated by the state feedback. The position of the container suffers from phase loss in comparison with the trajectory generated by the model. To compensate for model errors a faster LQ controller should be constructed.

5.3 System Performance and Capabilities

A working system has been constructed and tested. By using the two degree of freedom controller described in section 4.3.3 it is possible to follow trajectories generated by a model. The main goal of this thesis is thereby achieved.

5.4 System Limitations

As described in this chapter, the system developed in this work solves the task posed in Section 3.1. The system is not ready for real world deployment, but it shows that the basic theory works.

Of course there is always a trade off between computational efficiency and programming structure. But since the system operates in real-time it is adequately optimized.

The system was developed considering dynamics and characteristics of the industrial robot described in Chapter 2 and needs modifications to be adapted to an industrial crane. For instance a model of the crane is needed, as well as incorporation of adaptive features to handle movement while raising or lowering the cargo.

Chapter 6

Conclusions and Further Work

In this chapter, the conclusions of this work are presented followed by suggestions on how to continue development.

6.1 Conclusions

The main goals of this thesis was to stabilize cargo moved with an overhead crane and enable it to follow trajectories. The problem was scaled down due to cost and to make it possible to test the system in our institution. The work was divided into two fields; control and vision.

The control part of the problem needs a model describing an overhead crane with cargo. An LQ controller was constructed by assuming that the cargo hanging below the crane could be modelled as a two degree of freedom spherical pendulum. During this work the spherical pendulum was decoupled into two planar pendulums by projection onto two orthogonal planes. To incorporate regulation properties into the system a two degree of freedom controller was used, described in Section 4.3.3.

The problem of tracking the cargo[1] was solved using Harris corner detection algorithm. This algorithm was used to detect the corners in the captures from the camera. The corners of the box were then monitored with Kalman filters.

6.2 Further Work

The software available after this thesis work is able to track the corners of a container and feeding the coordinates to a MATLAB program that contains the two degree of freedom controller. Further work that will make use of the software packages developed will be discussed in the following two sections.

[1]In this thesis simplified to a box.

6.2.1 Vision

Further work on the software would include testing the vision package developed under real life conditions. Objects moving in the background and other perturbations are likely to occur if the camera is attached to a real crane. A more flexible user interface where the operator could mark the corners of a container and thus make it possible to adapt the vision software to different containers and wire devices is also desirable.

6.2.2 Control

The controller used in this thesis was a simple LQ controller and all work was focused on creating a working system. No tuning of the LQ controller took place after sufficient pole placement had been done. The LQ controller used was able to cancel out model errors and guarantee stability despite this. An interesting approach would be to use a non linear control structure and thus the non linear differential equations could be used.

In this thesis the container was modelled as a pendulum, but a more realistic model of the container would enhance the performance of the system. More important is what happens to the model used when the problem no longer is scaled down?

The problem of incorporating regulation properties, thus making it possible to follow trajectories, was solved using a two degree of freedom structure. This made it possible to follow trajectories generated from a model transfer function. However, it is also desirable to be able to follow non linear trajectories and this calls for an implementation of an inverted plant in the source if the control structure in this thesis is to remain.

Appendix A

Robot Trajectory

The purpose of this appendix is to describe, in detail, the actual method of transferring control signals to the robot.

A.1 Moving the Robot

The robot used throughout this thesis, as described in Chapter 2, is modified by the Department of Automation and Control to enable access to the robot controller's inner loop. This enables an external system to directly control the angular reference signal for each of the robot's six joints. The physical trajectories are transferred to the robot controller in chunks containing five[1] collections of reference signals for each joint, describing movement for 1/15 of a second. One such chunk is referred to as a trajectory.

A.2 Implementing Trajectory Generation

In this thesis the movement of the cargo is calculated in cartesian space. The robot however, requires control signals in joint space. To accommodate this a MATLAB program, referred to as the movement system. was developed. The movement system uses the robot acceleration, calculated by the controller which is also implemented in Matlab, to determine where the robot should move to. Since the controller operates in cartesian space the destination coordinates is in the same space. A MATLAB function converts the destination coordinates to joint space by using inverse kinomatics[2]. The end destination, in joint space, together with the previous end

[1]It is possible to have either fewer or more collections, but information provided by the Department of Automation and Control declared that more than five collections did not improve performance.

[2]The interested reader can learn kinematics in [9].

destination is used to generate a trajectory for the robot. Finally the trajectory is transferred, using MATCOMM, to the robot's internal controller.

List of Figures

Bibliography

[1] Torkel Glad, Lennart Ljung, *Reglerteknik Grundläggande teori*, Studentlitteratur Lund, 1989.

[2] Karl J. Åström, Björn Wittenmark, *Computer-Controlled systems*, Prentice Hall, 1997.

[3] A. Heyden, K. Åström, G. Sparr, F. Kahl, *Lecture notes in computer vision*, Center for Mathematical Sciences, Lund Institute of Technology, 2002.

[4] Young, Freedman, *University Physics*. Addison-Wesley, 1996.

[5] Tai L. Chow, *Classical Mechanics*, Wiley, 1995.

[6] Goldstein, *Classical Mechanics*, Addison-Wesley, 1980.

[7] C. Führer, A. Schroll, *Numerical Analysis—An Introduction*, Third Edition, Center for Mathematical Sciences, Lund University, 2001.

[8] Michael T. Heath *Scientific Computing—An Introductory Survey*, Second Edition, Mc Graw Hill, 2001.

[9] Craig, *Introduction to Robotics Mechanics&Control*, Addison-Wesley, 1986.

[10] Miss. Yang Rong, *Geometric Techniques for Control of a 2-DOF Spherical Inverted Pendulum*, Master Thesis, Honk Kong University of Science and Technology, 2000.