ISSN 0280-5316
ISRN LUTFD2/TFRT--5679--SE

Construction and Control of an Inverted Pendulum

Carina Hansen
Cecilia Svensson

Department of Automatic Control
Lund Institute of Technology
July 2000

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

Document name
MASTER THESES

Date of issue
July 2000

Document Number
ISRN LUTFD2/TFRT--5679--SE

Author(s)
Carina Hansen
Cecilia Svensson

Supervisor
Karl-Erik Årzén, LTH
Håkan Andersson, Saab Ericsson Space AB

Sponsoring organization

Title and subtitle
Construction and control of an inverted pendulum. (Design och reglering av en inverterad pendel).

Abstract

The aim of this master thesis is to develop a laboratory process that Saab Ericsson Space AB (SE) can use to demonstrate their fault tolerant computer systems at exhibitions. The process is an inverted pendulum shaped as a rocket to symbolize that the company develops fault tolerant control systems for satellites and carrier rockets e.g. for Ariane 5.

The inverted pendulum is a classical control problem and our task is to develop the mechanical equipment and to control it. The pendulum is made in two dimensions instead of the common one-dimension version. In order to make the model of the pendulum simpler a universal joint was used. That means that the pendulum falls in two separate directions at the same time. This solution makes it possible to split the controller into two and to get the measurements of the fall angles through potentiometers on the axes of the joint. The angle velocity is computed from the angle using an analog filter board.

To move the pendulum, a cross table consisting of two linear actuators is used. The cross table is driven by two step motors and can move the pendulum with smooth motions. The simulations of the system were done in a modelling tool called Modelica. A program written in C-code was made to control the complete system. The control-loop is built upon state-space theory. To make the computer and the mechanical process communicate, an I/O-board was developed by SE.

To compare the computer and a person it will also be possible in the future to control the pendulum with a joystick. The mind of a human being works with a frequency around 10 Hz. This can then be compered with the computer’s control frequency of 200 Hz.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
33

Recipient’s notes

Security classification

The report may be ordered from the Department of Automatic Control or borrowed through:
University Library 2, Box 3, SE-221 00 Lund, Sweden
Fax +46 46 222 44 22 E-mail ub2@ub2.se

Outline

- *Introduction* The first chapter presents the problem and goal of this master thesis and gives the background to what Saab Ericsson Space will use the equipment for.

- *The Equipment* Chapter 2 brings up the physical equipment to discussion. How the pendulum should be built and which motor that is to prefer, are examples of questions answered in this part. The communication between the equipment and the computer is also described.

- *Simulation* The modelling and simulation is treated in Chapter 3. Force analysis and system equations are derived and the simulation tool is presented. The results are explained and conclusions are made for demands upon the equipment.

- *Control Design* In Chapter 4 the state-space controller is derived and compared with the PD-controller used in the simulation.

- *Real-Time Implementation* Here is an overview of the program and modules presented. The computer hardware is also described.

Acknowledgements

We would like to thank Niclas Åberg at Östergrens Elmotor AB for untiring answered all our questions about motors and linear actuators. We also want to thank Christer Houbaer at Elwia for helping us finding a joystick that matches our demand for the pendulum.

Thanks to section DD and DP for your support and for listen to our whining. Finally we will thank our supervisors Karl-Erik Årzen at the Department of Automatic Control, Lund Institute of Technology, Håkan Andersson at Saab Ericsson Space AB.

Table of Contents

1. INTRODUCTION

1.1 Background

When the project is complete a laboratory equipment will show a fault tolerant system. A fault tolerant system often contains several computers. The computers work in parallel to minimize the risk of system breakdown, even if one of the computers should stop working. To visualize the fault tolerance the computers shall control an inverted pendulum in two dimensions.

This master thesis is the first of two. It should treat how to develop the inverted pendulum and how to control it. The pendulum will have the shape of a rocket to symbolize that Saab Ericsson Space AB (SE) develops fault tolerant control systems for satellites and carrier rockets e.g. for Ariane 5.

1.2 Problem description

When the development of the inverted pendulum started, we got free hands to solve the problem and few ideas how to do it. We started to search the market for possible solutions. Based on this, a model for simulations of the pendulum was created in Modelica. Simulation results together with true values (from Östergrens Elmotor AB) shall result in an order of the equipment. A control-loop to balance the pendulum will be developed out of this solution. The controller will then be implemented in the ATB-computer.

1.3 Objective

The goal of this master thesis is to develop complete equipment with a working control system for an inverted pendulum. This includes making a good description and technical report for the next master thesis, so that they can modify our solution to a fault tolerant system.

1.4 Status

Each part functions separate from each other, but not as a unit. It is hard to tell if the control design works satisfactory without testing it on the real system. From our calculations and simulations the system should work.

2. Physical Process Description

The total system consists of a pendulum, cross table, drive units, motors, two boards and an ATB-computer. The I/O-board (SCORPIO) is the link between the ATB-computer and the rest of the equipment. SCORPIO also have some control functions. The filter-board (AVM) makes it possible to calculate the fall velocity for the pendulum. The cross table consists of two linear actuators. The motors move the linear actuators. The motors are supplied with right voltage and effect by the drive units.
The pendulum, cross table, motors and the boards will be discussed in this chapter while the ATB-computer will be explained in Chapter 5.

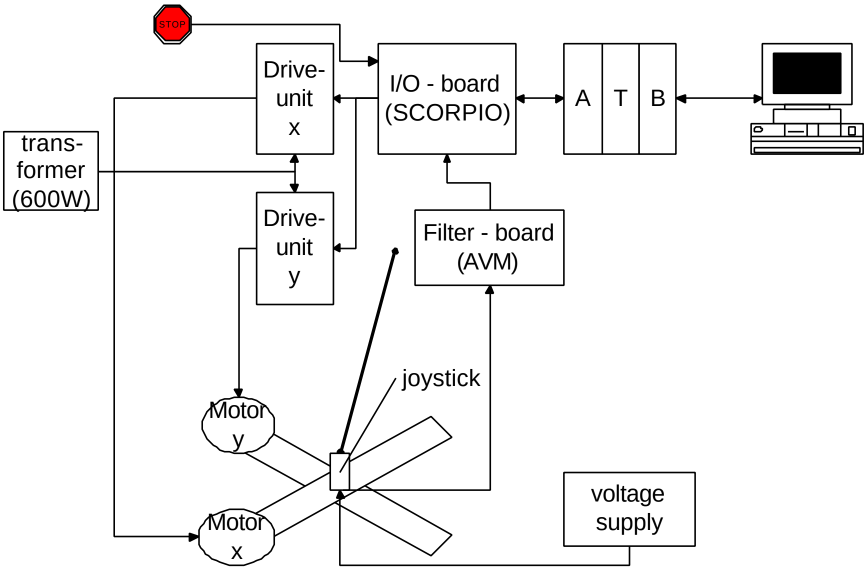


Figure 2.1 Total system design

2.1 The Pendulum

To simplify the model of the pendulum the assumption was made that the movement of the pendulum could be split into two perpendicular planes. This assumption makes it necessary to choose a universal joint instead of a common ball-joint. A universal joint consists of two perpendicular joints compounded in one plane (Figure 2.2). Most joysticks are constructed with this type of joint.

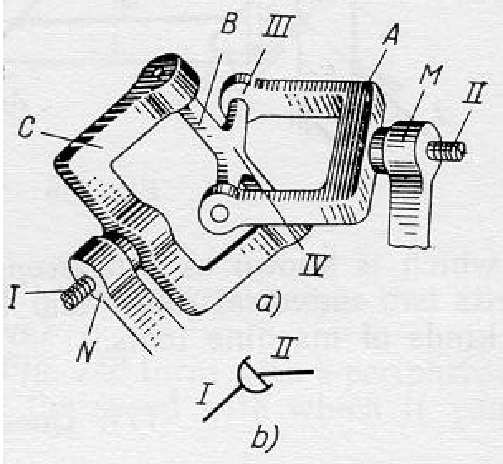
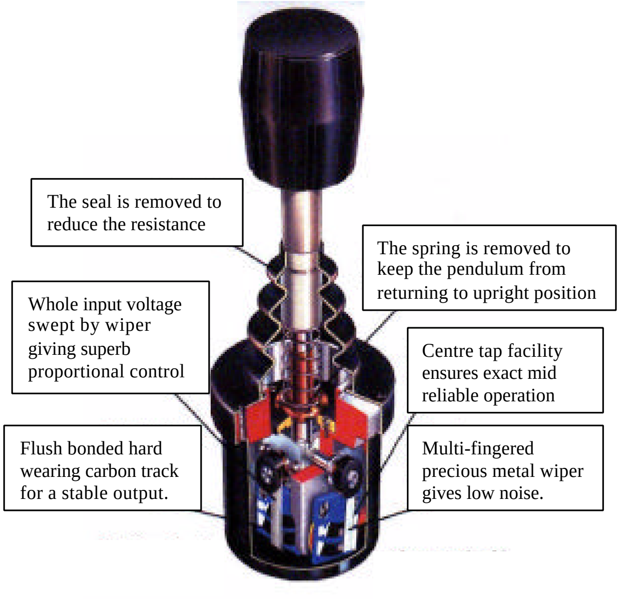


Figure 2.2 Universal joint.

To measure the fall angle a joystick is a good solution since the sensors already are attached to the axes. The only problem of this idea is to find a joystick without a built-in backlash around the upright position. The joystick used in this application is shown in Figure 2.3. Technical details are available in Appendix A.



The seal is removed to reduce the resistance

The spring is removed to keep the pendulum from returning to upright position

Whole input voltage swept by wiper giving superb proportional control

Centre tap facility ensures exact mid reliable operation

Flush bonded hard wearing carbon track for a stable output.

Multi-fingered precious metal wiper gives low noise.

Figure 2.3 The joystick for the pendulum.

The top of the pendulum has to be above the base of the pendulum in order to balance it. The length of the stroke (see Figure 2.4) will therefore, together with the maximum fall angle, give the upper limit for the pendulum's length through the following relation.

$$L = \frac{length\_of\_stroke}{\sin \Theta}$$

Choosing the total motion to a maximum of 75×75-cm gives a stroke-length at approximately 37 cm. The maximum fall angle of the joystick is ±18° and this will limit the length of the pendulum to 119 cm. This will be discussed later in the simulation part; the pendulum length will also be much shorter for other reasons.

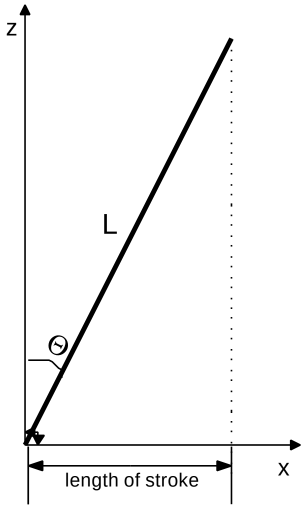


Figure 2.4 Length of stroke.

Design constraints for the pendulum:
Max fall angle = 18°
Max angle velocity ~ 2.6 rad/s ~ 149°/s
Max length = 1.19 m

2.2 Cross table

The ideas of how to balance the pendulum were numerous and varying. Everything from pneumatic control in different shapes to the final solution with linear actuators was discussed.

There are two available linear actuators one with belt-driven unit and the other with ball-screw-driven unit, Figure 2.5. The belt-driven unit can reach higher speed, acceleration and a greater distance. The ball-screw-driven unit has higher repeatability but can not reach as high acceleration as the belt-driven unit.

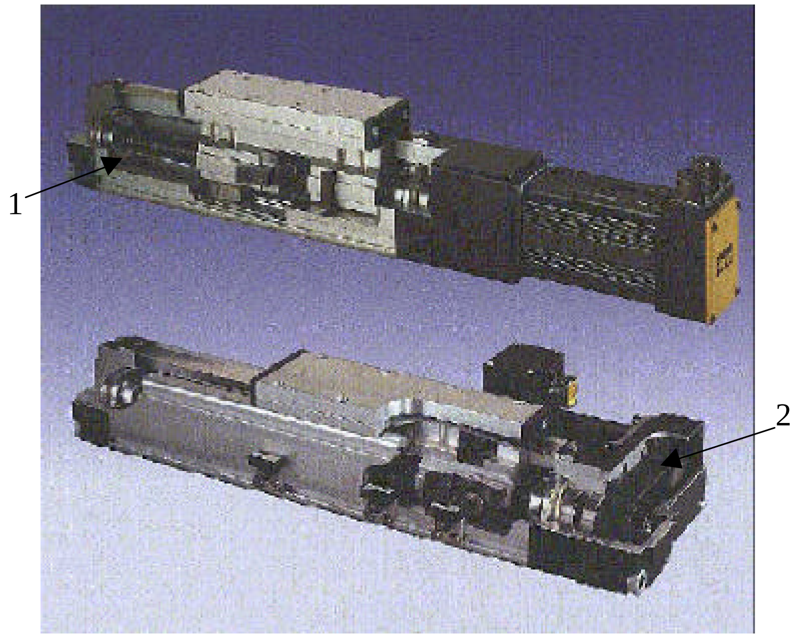
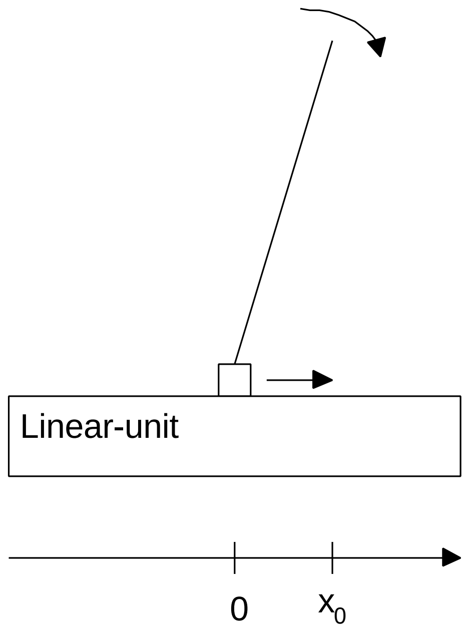
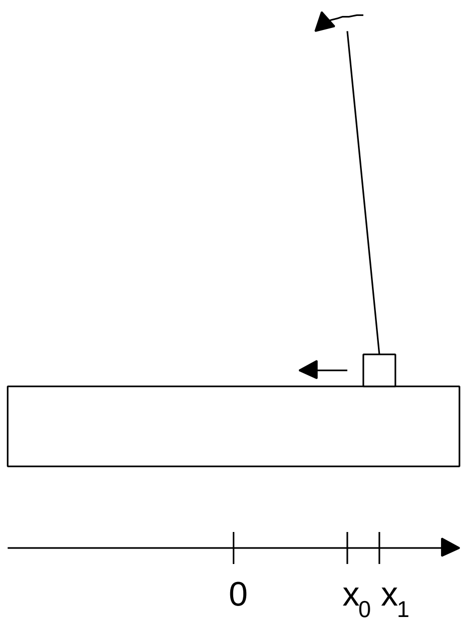


Figure 2.5 The upper picture is the screw-driven unit (1) and the other is the belt-driven unit (2) that is used in this solution.

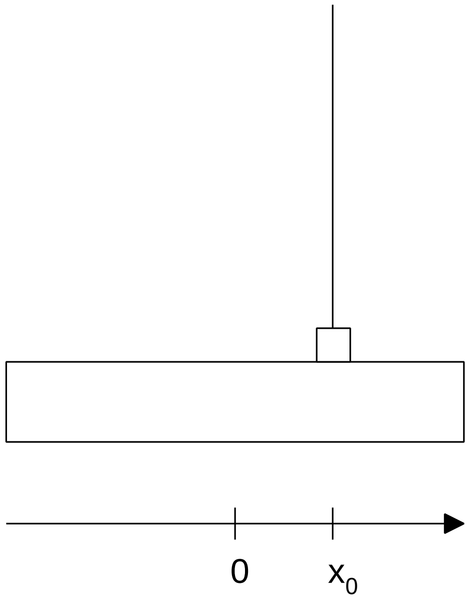
The belt-driven linear actuator was chosen since high acceleration for the table is a critical requirement. The acceleration is necessary to create an overcompensated motion i.e. the table needs to accelerate faster and reach a higher speed than the pendulum's falling motion in the same direction. This higher motion makes the pendulum fall a little less in the opposite direction. If this happens repeatedly the pendulum will reach a balanced state. See Figure 2.6.



Figur 2.6 a) The pendulum has started to fall. The plate on the linear unit is moving in the same direction trying to counteract the pendulums motion.



Figur 2.6 b) The plate has made an overcompensated movement and the pendulum has changed direction.The plate is following the pendulum again until the fall-motion is close to zero.



Figur 2.6 c): The pendulum has been balanced which results in that the top no longer has any motion.

2.2.1 Construction and Function

To create a two-dimension motion two linear actuators have to be mounted on top of each other in a cross shape. The moving part of the actuator is only a small plate on top of the unit. The bottom unit has to be more robust and it has to be driven by a stronger motor to bear the weight of the upper unit (see Appendix B). Each axle has its own motor and drive-unit so they are controlled separately from each other.

To check that the plate has not come to close to any edge there are four sensors on the table, one sensor at each edge. When a sensor is set there are only two centimetres left to the edge. The motor then has to be stopped before it hits the edge and become overheated. To control if the plate on the cross table is centred in the middle of the table, there are two centre sensors applied to the middle. The home sensors are set when the plate is in the exact right position. This is necessary since dead reckoning is used. More about the sensor use is discussed in the I/O-part, see Chapter1.4.

2.3 Motor

There are a lot of motors to choose among which all reach the demand of speed and repeatability. The two motors discussed in the following chapter are common, and used in many applications. They also have the advantage of being easy to control.

2.3.1 Step motor vs. Servomotor

Servo motor

With a servomotor it is possible to control position, number of revolutions and the braking. By using higher frequencies compared to the step motor, high acceleration is achieved. A change of current direction will only affect the acceleration of the motor, not the direction of the motion. In this case, a feedback signal of velocity and position to control the motor is needed. To get this feedback a sensor is used on the motor, making it more expensive and complicated compared to a simple step motor.

Step motor

The step motor consists of one stator and one rotor. When one or more of the windings have current drift, a magnetic field is generated. This field will turn the rotor until balance arises. The motor has now taken one step and found its new position. By sending in pulses to the motor, the current drift will move from one winding to the next. This results in the rotor moving step by step, one at a time.

To change the motor’s speed or direction, the pulse frequency or the direction signal to the drive-unit that controls the motor is changed. The motor can only take a step if a pulse has been sent into it. This makes the motor easy to control since no feedback signals are needed as e.g. for the servomotor. The step motor was chosen for this application due to the easy control and the low cost. (Appendix C)

2.3.2 Drive-unit

This is a high-performance motion control equipment, capable of producing rapid movement and very high forces. It can be powered by $48V_{dc}$ or $70V_{ac}$ and 5A supplies (appendix D). The pulse train and direction signal for the driver will come from the SCORPIO. The driver then creates the corresponding electrical pulse train to the motor.

2.3.3 Limits for the step motor and the linear actuator

In the simulations (Chapter 3.3) it was found that the max velocity needed is 0.6 m/s, to stabilise the pendulum when the fall angle is large (~18°). The max velocity is reached in 4 cm and has a linear progress as seen in figure 2.7. In the drive-unit it is possible to change the step length for the motor. The step length is depending on how exact the movements have to be. In this case it does not have to be so exact. It is high velocity that is needed. The lead on the linear unit has a circumference of 70mm for the small unit and 100 mm for the large unit.

$$Step\_length = \frac{linear\_lead\_circumference}{step/revolution}$$

step/revolution	400	800	2000	4000
Step length x (mm)	0.1750	0.0875	0.0335	0.0175
Step length y (mm)	0.2500	0.1250	0.0500	0.0250

Table 2.1: Step length

The reach the same velocity with a shorter step length SCORPIO has to send out pulses with higher frequencies to the motors. That can cause disturbances that are not wanted. The lowest precision, 400 step/revolution, is therefore chosen.

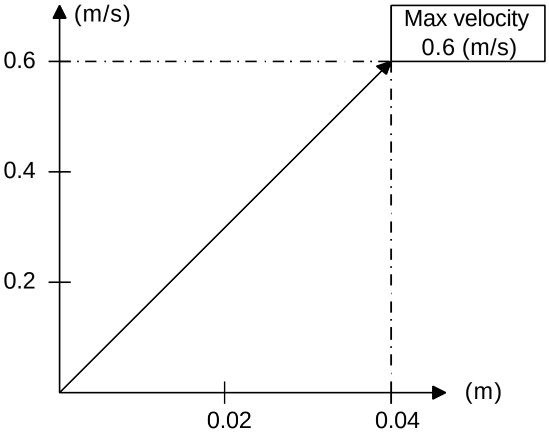


Figure 2.7 Acceleration from 0 to 0.6 in 4 cm.

Based on this, some important values can be calculated.

Limits for the actuator:
Position = 0.37 m
Max velocity = 0.6 m/s
Max acceleration = 4.5 m/s^2

***Limits for the motor**:*
Step length x = 0.175 mm
Step length y = 0.250 mm
Sample time = 4.992 msec
Max number of steps per sample (x-direction) = 17 step/sample
Max number of steps per sample (y-direction) = 12 step/sample
Sample to reach max velocity = 0.13 / 4.99m = 26 samples

The connections to the motor must be using a high quality braided-screen cable. This to prevent cross talk disturbances between control signals.

2.4 Boards

To control the step motor drivers a specified I/O board, called SCORPIO, has been developed by SE. It interfaces the signals from four end-sensors and two centre-sensors. It also takes care of all the signals coming from the inverted pendulum. The angle signals are conditioned to provide a better measurement and filtered to provide the angle velocity. The filter board (AVM) is developed by LTH (Department of Automatic Control). Specifications for this board are found in Appendix E. These signals are connected to an A/D-converter on SCORPIO.

SCORPIO should also count all steps that have been sent out, making it possible to update the position counting and velocity of the table in the program. The properties of the counter are seen in Figure 2.8. Every time the pendulum passes the center sensor the counter is reset and restarted. Signals from the program out to SCORPIO determine how many steps the motors should move and in which direction. If the counter passes the limit an overflow/underflow bit is set. This bit will not be cleared until the plate passes the center sensor. The specification of SCORPIO is found in Appendix F.

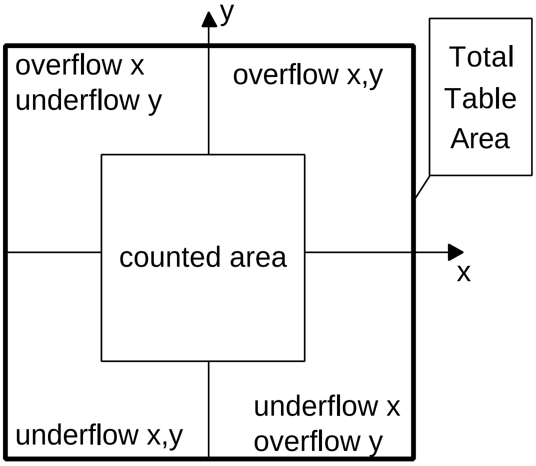


Figure 2.8 Count area on the cross table.

3. Modelling and Simulation

A model of the equipment is used for the control design for the real system. The dynamics of the model is calculated to be as close as possible to the real process. Through simulations it is possible to extract the requirements for the equipment such as the length of the pendulum and requirements upon the velocity and acceleration of the table. Also the impact of noise and sample rates on the measurements and stability of the system is studied.

3.1 Modelling an inverted pendulum

Due to the assumption that the pendulum could be split into two perpendicular parts, the equations can be derived for one direction at a time.

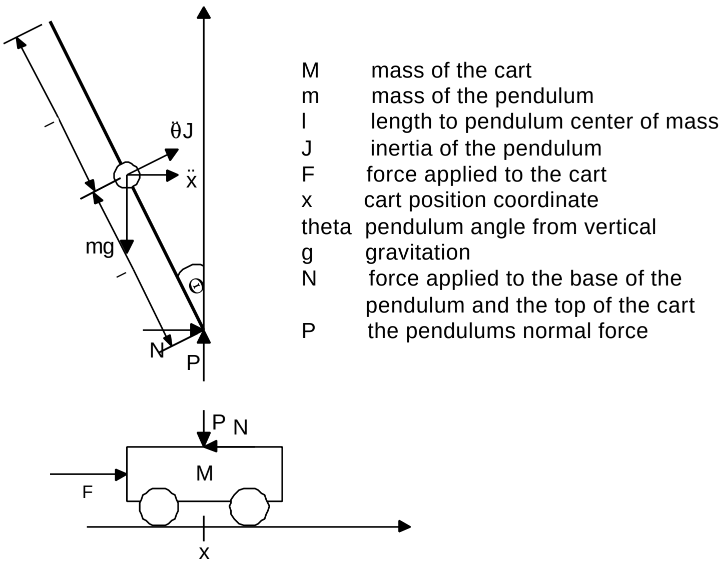

θJ
ẍ
mg
Θ
N
P
P
N
M
F
x
M mass of the cart
m mass of the pendulum
l length to pendulum center of mass
J inertia of the pendulum
F force applied to the cart
x cart position coordinate
theta pendulum angle from vertical
g gravitation
N force applied to the base of the pendulum and the top of the cart
P the pendulums normal force

Figure 3.1 Model of the pendulum

Force analysis and system equations

Summing the forces of the cart in the horizontal direction give the following equation of motion

$$M\ddot{x} + N = F$$

For the pendulum in the same direction an equation for N is achieved.

$$N = m\ddot{x} + ml\ddot{\Theta}\cos\Theta - ml\dot{\Theta}^2\sin\Theta$$

If these two equations are put together it results in the first equation of motion for this system.

$$(M+m)\ddot{x} + ml\ddot{\Theta}\cos\Theta - ml\dot{\Theta}^2\sin\Theta = F \qquad (1)$$

Summing the forces perpendicular to the pendulum gives the equation

$$P\sin\Theta + N\cos\Theta - mg\sin\Theta = ml\ddot{\Theta} + m\ddot{x}\cos\Theta$$

To get rid of N and P, the moments around the centroid of the pendulum are summed. The equation found is

$$-Pl\sin\Theta - Nl\cos\Theta = J\ddot{\Theta}$$

Combining these last two equations gives the second equation of motion shown below.

$$\left(J + ml^2\right)\ddot{\Theta} + mgl\sin\Theta = -ml\ddot{x}\cos\Theta \qquad (2)$$

The two equations of motion are then linearized about $\Theta = 0$.

$$\begin{aligned} &(1) \quad (\mathrm{M}+\mathrm{m})\ddot{\mathrm{x}} - \mathrm{ml}\ddot{\Theta} = \mathrm{u} \\ &(2) \quad \left(J + ml^2\right)\ddot{\Theta} - \mathrm{mgl}\,\Theta = \mathrm{ml}\,\ddot{\mathrm{x}} \end{aligned}$$

These two system equations can be presented in state-space form. The matrices A and B represent the inputs and the process dynamics, while C and D give an expression for the output signals.

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\Theta} \\ \ddot{\Theta} \end{bmatrix} = Aq(t) + Bu(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \dfrac{m^2gl^2}{p} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \dfrac{mgl(M+m)}{p} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \Theta \\ \dot{\Theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{J+ml^2}{p} \\ 0 \\ \dfrac{ml}{p} \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} x \\ \Theta \end{bmatrix} = Cq(t) + Du(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \Theta \\ \dot{\Theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u(t)$$

Since a computer controls the system, the model of the system has to be sampled. This to be able to use the full potential of computer control. This transformation could be done in many ways. One way is to make it numerically in Matlab®. The matrices Φ and Γ are the corresponding discrete-time matrices to A and B. The matrices C and D are not affected. The continues state space model is sampled with the sample-time 4.992 ms and gives the discrete-time system described by

$$q(k+1) = \Phi_{top}q(k) + \Gamma_{top}u(k) = \begin{bmatrix} 1 & 0.00512 & 0.000022 & 0 \\ 0 & 1 & 0.008812 & 0.000022 \\ 0 & 0 & 1.000185 & 0.005120 \\ 0 & 0 & 0.072297 & 1.000185 \end{bmatrix} q(k) + \begin{bmatrix} 0.000050 \\ 0.019612 \\ 0.000061 \\ 0.024016 \end{bmatrix} u(k)$$

$$q(k+1) = \Phi_{bottom}q(k) + \Gamma_{bottom}u(k) = \begin{bmatrix} 1.0 & 0.00512 & 0.000001 & 0 \\ 0 & 1.0 & 0.000504 & 0.000001 \\ 0 & 0 & 1.000159 & 0.005120 \\ 0 & 0 & 0.062124 & 1.000159 \end{bmatrix} q(k) + \begin{bmatrix} 0.000003 \\ 0.001122 \\ 0.000004 \\ 0.001375 \end{bmatrix} u(k)$$

3.2 Modelica

Modelica is a language for physical system modelling, visualisation and interaction. The benefits of Modelica are that the program contains predefined equations, and the animation tool makes it possible to see how the system moves in 3D (Figure 3.2). The equations in Modelica are non-causal i.e. no particular variable needs to be solved for manually. A Modelica tool will have enough information to decide that automatically.

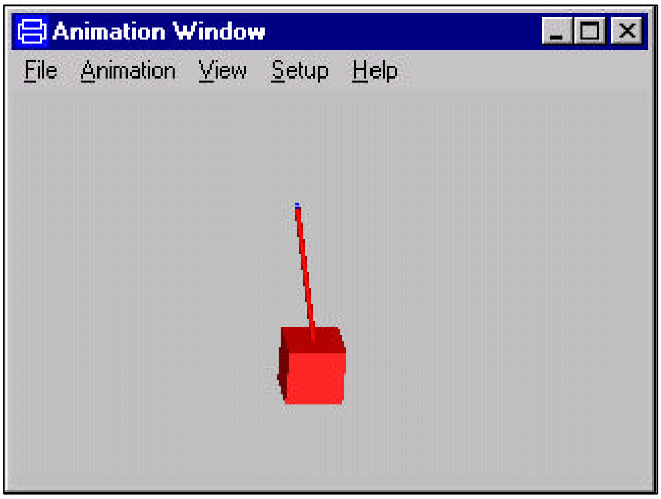


Figure 3.2 Animation of pendulum

Simulink is a modelling tool often used for modelling processes and controllers. Both Modelica and Simulink use the idea of building models with blocks. The major difference is that Modelica has already defined equations for a fixed block while Simulink´s blocks are built upon mathematical operators and functions. The user interface in Simulink is easy to understand. It is easy to modify a Simulink model e.g. to limit outputs or add noise. In this case however, when not all of the equations for e.g. step motors and linear actuators were known a great effort had been necessary to get a good model in Simulink. For this reason Modelica is chosen instead of Simulink. Appendix G contains a description of how to build a model in Modelica.

3.2.1 Model and equation

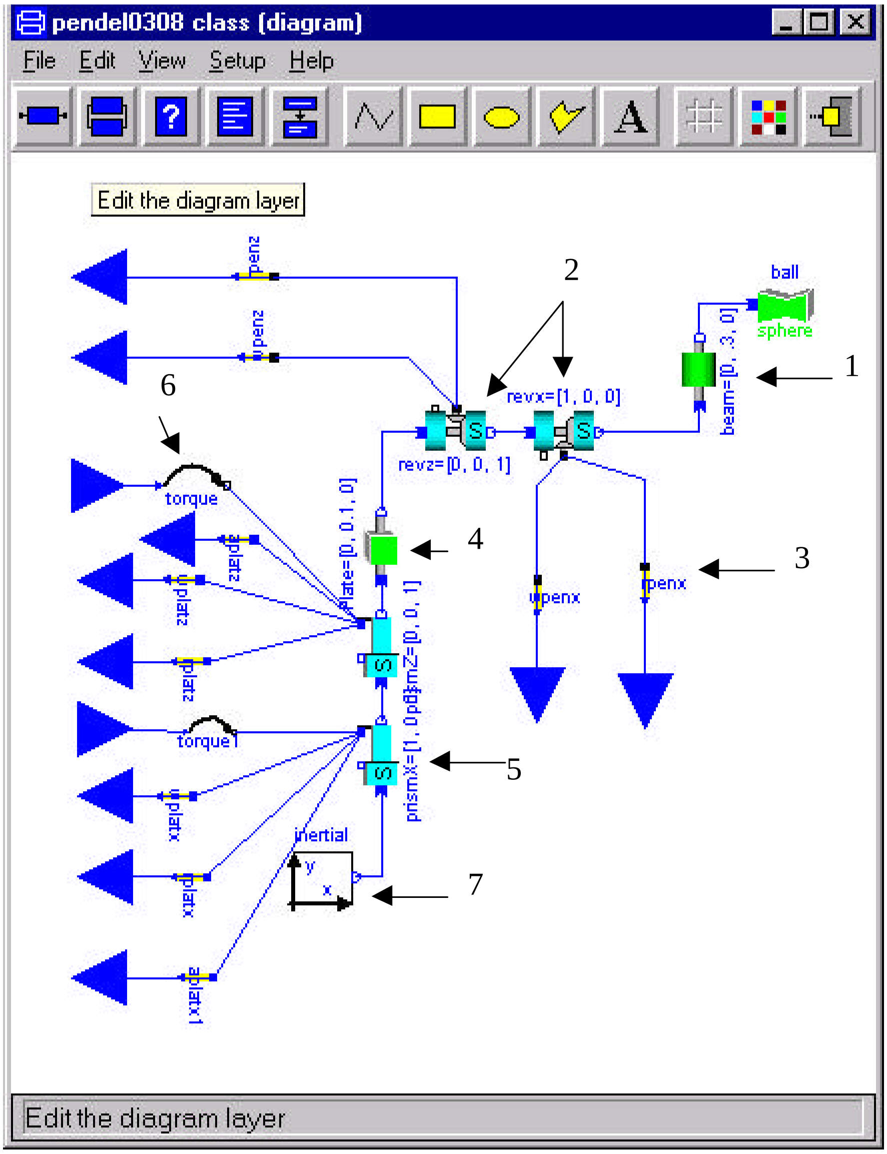
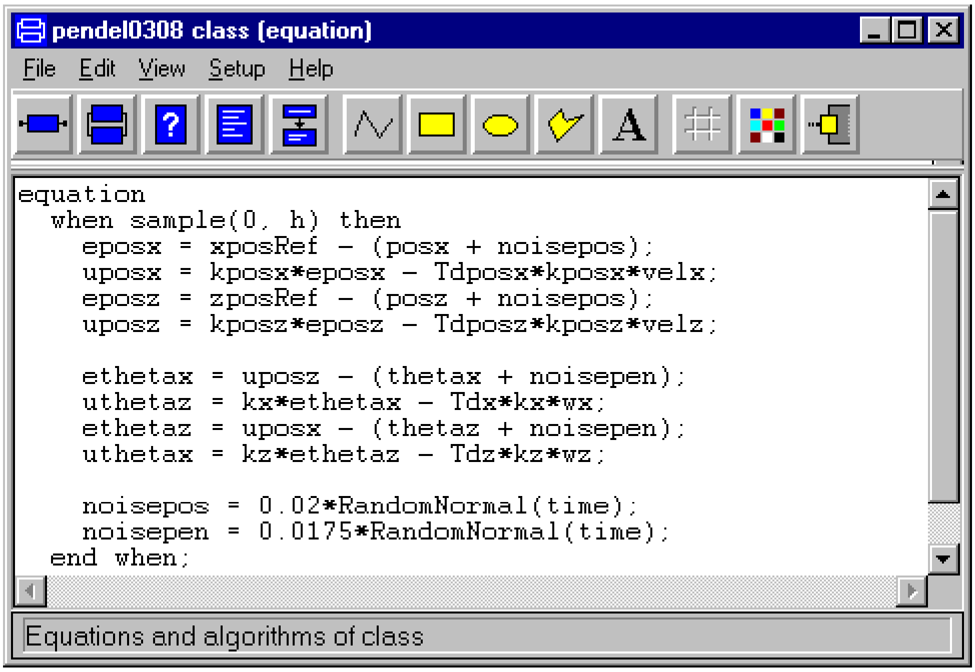


Figure 3.3: Model of the pendulum made in Modelica

This model (Figure 3.3) will describe the whole equipment from the pendulum to the motors. Symbol 1 in figure 3.3 represents the pendulum. Symbol 2 is equivalent to the universal joint that gives the pendulum its dynamic properties.

Symbols 5 represent the linear actuators (including motors) that move the plate (symbol 4) where the pendulum is applied. There has to be an in-signal that control the linear actuators and in this case it is torque (symbol 6) for the motors. To make the symbols move as the real equipment is planned to, an initiation of the co-ordinate system (symbol 7) is made. All signals are then defined from this initiation.

To make own equations in the program there are sensors (symbol 3) used to extract information from the joint and the linear actuators. The information from symbol 2 is the fall angles and angle velocity. This is the same information that is possible to extract from the real joint. The velocity signals that the sensors take from symbol 5 are the same as the in signals to the real step motors.


pendel0308 class (equation)
File
Edit
View
Setup
Help
Equations and algorithms of class

```
equation
  when sample(0, h) then
    eposx = xposRef - (posx + noisepos);
    uposx = kposx*eposx - Tdposx*kposx*velx;
    eposz = zposRef - (posz + noisepos);
    uposz = kposz*eposz - Tdposz*kposz*velz;

    ethetax = uposz - (thetax + noisepen);
    uthetaz = kx*ethetax - Tdx*kx*wx;
    ethetaz = uposx - (thetaz + noisepen);
    uthetax = kz*ethetaz - Tdz*kz*wz;

    noisepos = 0.02*RandomNormal(time);
    noisepen = 0.0175*RandomNormal(time);
  end when;
```

Figure 3.4: The equations that controls the system model.

The equation in figure 3.4 describes two cascade coupled PD-controlers with the sample time set to 5 ms. **RandomNormal** is a function make random sample between 0 and 1. This is used to simulate disturbances. The variable **Noisepos** is set to 0.02m to symbolize an error in position that can appear at the real cross table. The variable **Noisepen** is set to 0.0175 rad (=1°) that is the same as the error that can affect the real fall angle. The noise values are explained in Chapter 4.5.

3.3 Results

To get a picture of the dynamics of the pendulum three different simulations were made. Note that the start angle in x-direction is larger than the real pendulum could reach. This was not known when the simulations were made. A diagram of a simulation can be seen in Figure 3.6.

To see if the mass of the pendulum has any effect at the measurements the density were changed. The different densities tested correspond to real material. Wood has a density of about 0.4 – 0.8 kg/dm^3, plastic's density is 1.2 kg/dm^3 and aluminium 2.7 kg/dm^3 and many common metals have a density of about 8 kg/dm^3.

Initial value:

Radius	0.005	(m)
Length	0.5	(m)
Initial anglex	0.35	(rad) (~20°)
Initial angley	0.2	(rad) (~18°)
Sample time	0.005	(sec)

Rho (kg/dm^3)	Mass (kg)	xpos(max) (m)	ypos(max) (m)	xvel(max) (m/s)	yvel(max) (m/s)	Moment (Nm)	Setting time (sec)
0.4	0.16	0.37	0.21	0.95	0.49	35, 20	7.5
0.8	0.31	0.37	0.21	0.95	0.48	35, 20	7.5
1.2	0.47	0.37	0.21	0.95	0.47	35, 20	7.5
2.7	1.06	0.37	0.21	0.94	0.46	35, 20	7.5
8	3.14	0.38	0.21	0.94	0.47	35, 20	7.5

Table 3.1 Change of density.

Xvel in Table 3.1 is the velocity that was needed to make the pendulum fall in opposite direction. Xpos is the distance where the plate has passed the top of the pendulum and turns over. As seen in Table 3.1 small changes in density does not affect the max position and the velocity.

The absolute maximum limit of the length of the pendulum is set already in Chapter 2.1 where the pendulum is discussed. Here the consequences of changing the length are verified more closely.

Initial value:

Density	2.7	(kg/dm^3)
Radius	0.005	(m)
Initial anglex	0.35	(rad)
Initial angley	0.2	(rad)

Length (m)	**Mass** (kg)	**Dev. position** (max) (m)		**velocity** (max) (m/s)		**x-acc.** (max) (m/s^2)	**Settling time** (sec)	**Moment** (Nm)		**Sample time** (sec)
		x	**y**	**x**	**y**			**x**	**y**	
1.0	0.21	0.47	0.25	1.20	2.40	96.0	5.0	45	20	0.01
0.8	0.17	0.42	0.23	0.85	1.80	72.0	4.8	35	20	0.01
0.6	0.13	0.39	0.22	0.81	1.41	56.4	4.9	35	20	0.01
0.6	0.13	0.39	0.22	0.65	1.18	47.2	7.0	35	20	0.005
0.5	0.11	0.37	0.21	0.47	0.94	37.6	7.0	35	20	0.005
0.4	0.08	0.36	0.20	0.44	0.76	30.4	7.0	35	20	0.005
0.35	0.07	0.36	0.20	0.44	0.60	12.0	7.5	35	20	0.005
0.3	0.06	0.35	0.20	0.40	0.60	12.0	7.5	35	20	0.005

Table 3.2 Change of length.

A small fall angle will give a shorter turning point than a larger fall angle. That explains the big difference between the maximum movement in x-direction and y-direction respectively. Since the max deviation of the real pendulum is 18° the conclusions for the equipment was drawn from the x-direction simulation.

To keep the table size down the length of the pendulum was reduced as seen in cross Table 3.2. Between 0.3 m and 0.6 m the difference in position deviation is not very large. The table size will be about 0.75 m. The corresponding velocity is of vital importance to decide which length to use. A length of 0.3 m only requires a velocity of 0.6 m/s while a length of 0.6 m requires 1.18 m/s. Choosing the shorter pendulum makes it possible to use step motors. The physical limits of the motor are found in Chapter 2.3.

Even if the shorter pendulum implies both lower velocity and acceleration, the accelerations are still very high. A simulation with a bounded acceleration should be preferred but problems modifying the Modelica model made that impossible. Instead some assumptions were made.

The fall angle will never reach its maximum. A smaller angle gives a smaller position deviation. This gives some extra space and time to counteract the falling motion of the pendulum making it possible to balance the pendulum with a limited acceleration.

The sample time is reduced because of ringing problems, Figure 3.5. A short pendulum requires a short sample time to maintain stability and performance.

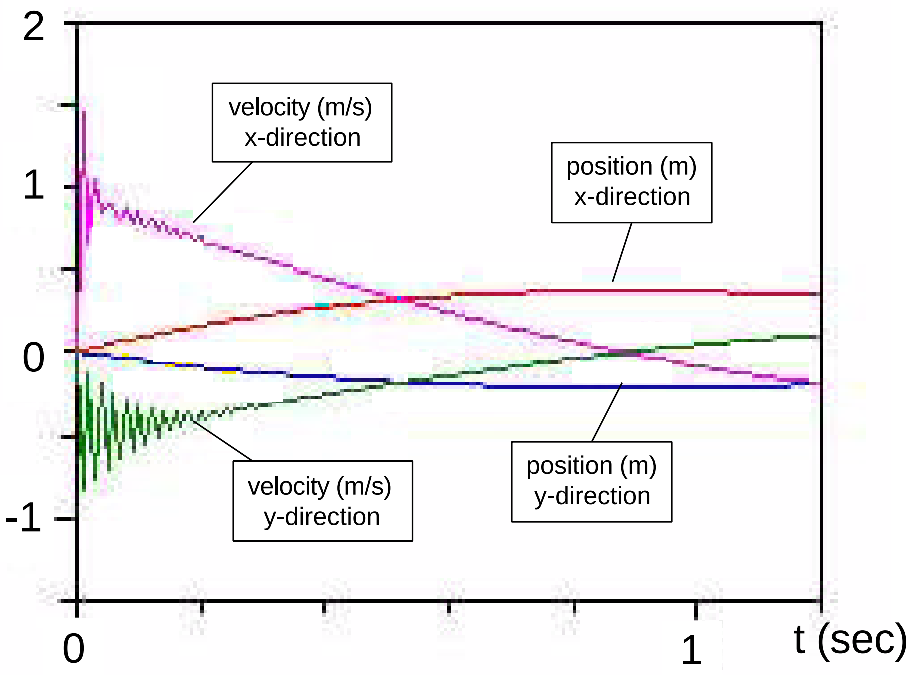


Figure 3.5 The simulation is done without noise for the following values: l = 0.6 m, anglex = 20°, angley = 12°, h = 0.01 sec.

Noise added to the measured signals did not influence the stability or the settling time considerable, compare Figure 3.6 and 3.7.

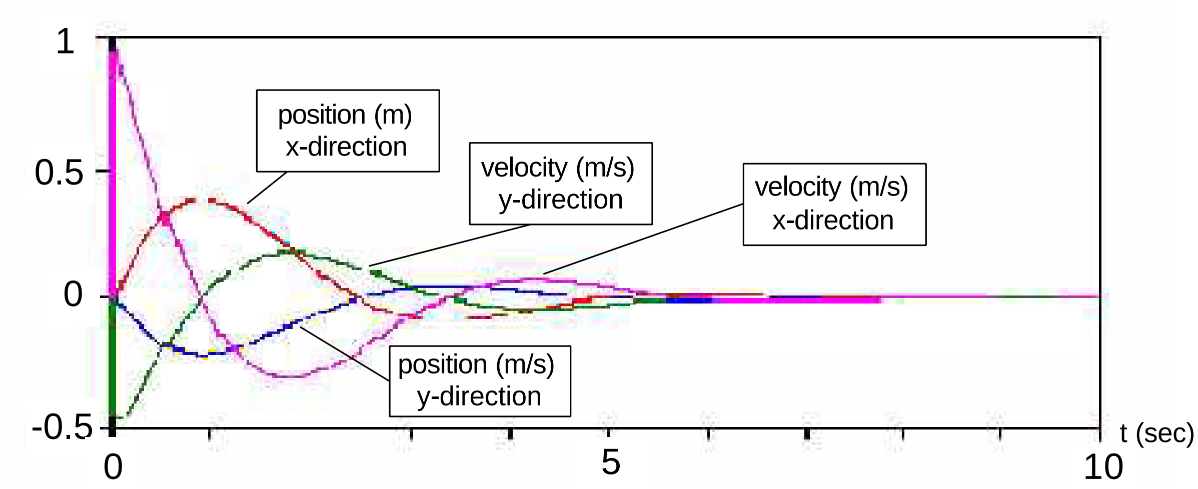


Figure 3.6 Simulation with: l = 0.5 m, start angles 0.35 and 0.2 rad., h = 0.005 without noise

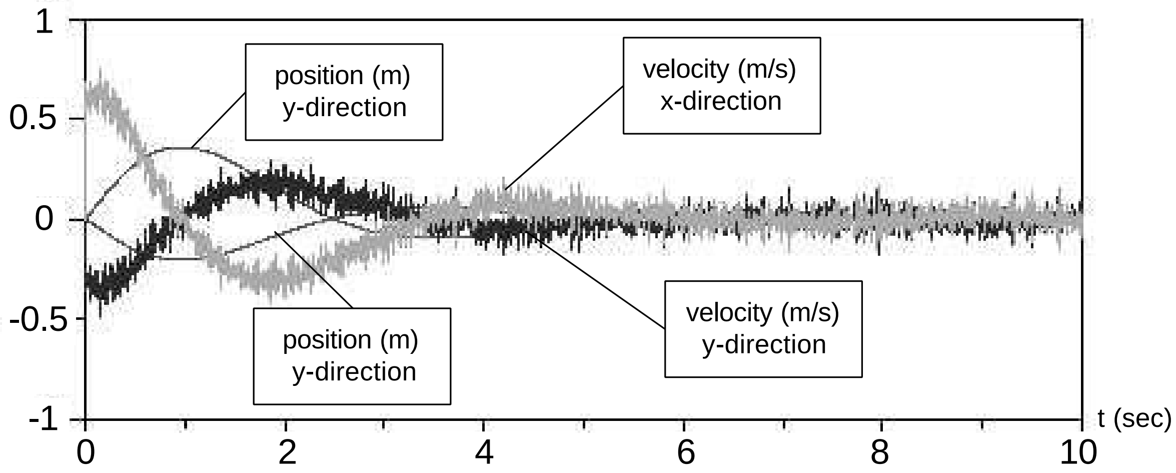


Figure 3.7 Simulation with: l = 0.5 m, start angles 0.35 and 0.2 rad., h = 0.005 with noise

4. Control Design

Modelling the dynamic properties of a process makes it possible to calculate a controller for the process. Sometimes process uncertainty, disturbances and time delays have to be taken into account. However as shown later this controller does not have to involve these problems.

4.1 Control Structure

With all states available regulation by state feedback is to prefer. The control structure is shown in figure 4.1. When the properties of the system are specified of its closed-loop poles, it is natural to require that the feedback is linear. This feedback is then of the form.

$$u(k) = -Lq(k) \qquad \text{where } L = \begin{bmatrix} l_1 & l_2 & l_3 & l_4 \end{bmatrix}$$

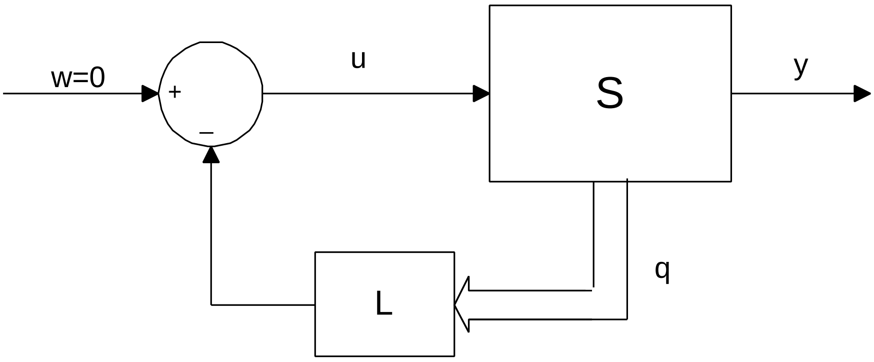


Figure 4.1 Regulation by state feedback.

4.2 Pole-Placement

The continuos poles are placed in a circle pattern as seen in figure 4.2. The distance between the origo and the poles influence the rapidity of the systems step response. The damping ζ of the system is dependent of the angle from the negative axle to the pole as

$$V = \cos j$$

The continuos poles are transformed into discrete time. The reason why the poles are placed in continuos time at first is that the properties of the system are easy to connect to concrete values in the picture.

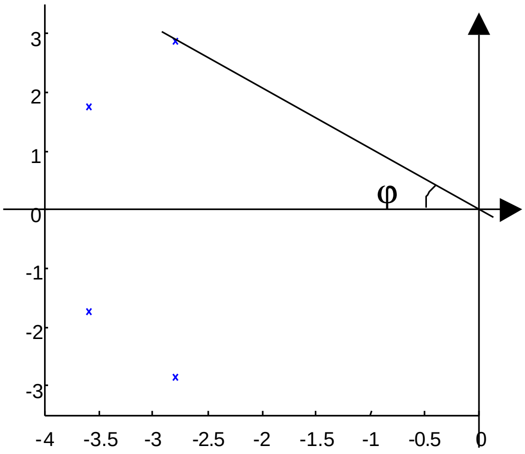


Figure 4.1 Pole-Placement

4.3 Calculation of the feedback vector

The feedback law together with the discrete state space model of the process derived in the chapter of Modelling and Simulation, give following equations

Equation 4.1

$$q(k+1)=(\Phi-\Gamma L)q(k)=\begin{bmatrix} 0 & \Phi_{12} & 0 & 0 \\ -\Gamma_2 l_1 & -\Gamma_2 l_2 & \Phi_{23}-\Gamma_2 l_3 & -\Gamma_2 l_4 \\ 0 & 0 & 0 & \Phi_{34} \\ -\Gamma_4 l_1 & -\Gamma_4 l_2 & \Phi_{43}-\Gamma_4 l_3 & -\Gamma_4 l_4 \end{bmatrix} q(k)$$

$$y=\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} q(k)$$

From this equation the characteristic equation is derived. The characteristic equation (equation 4.2) and the pole-placement equation give an equation system with the feedback constants (L) as unknowns.

Equation 4.2 The characteristic equation

$$\det(lI-(\Phi-\Gamma L))=\begin{vmatrix} l & -\Phi_{12} & 0 & 0 \\ \Gamma_2 l_1 & l+\Gamma_2 l_2 & -\Phi_{23}+\Gamma_2 l_3 & \Gamma_2 l_4 \\ 0 & 0 & l & -\Phi_{34} \\ \Gamma_4 l_1 & \Gamma_4 l_2 & -\Phi_{43}+\Gamma_4 l_3 & l+\Gamma_4 l_4 \end{vmatrix}=$$

$$=l^4+(\Gamma_2 l_2+\Gamma_4 l_4)l^3+\Gamma_2 l_1 l^2+(\Gamma_4 l_3-\Phi_{43})l+(-\Phi_{43}\Gamma_2+\Phi_{23}\Gamma_4)(l_1+l_2)$$

4.4 Process uncertainty

The A/D-converter on SCORPIO needs 10µs to sample and place the measurements into the registers in the ATB. Measuring the calculation time in the program give a worst case of 18.75µs. These two delays added to the delay to set out the control signal gives a total delay of 29µs. The delay is then about 0.58 % of the sample time and does not need to be taken into consideration.

Many different sources can contribute to the total error in the angle measurement. The voltage supplies over the potentiometers in the joystick e.g. can variate. To prevent this a reference generator is used. The reference is placed on the AVM. The signals from the joystick pass through the AVM can be affected by disturbances. These disturbances are reduced to about 100mV through screened cables. The value 100mV corresponds to about 0.396°. The truncation error that appears in the A/D-converter is 0.154°. If the pendulum fall with a velocity of 149°/s and the time delay is 29µs a angle error of 0.004° arise. The total angle error is 0.55°. The simulations managed an error of 1°, which is nearly twice the calculated error.

The mass of the pendulum does not influence the control as seen in the simulations. However, the mass matters when the pendulum changes direction. The breakout friction is easier overcome with a heavy pendulum.

The repeatability was of minor importance compared to the velocity in the choice of linear unit. To prevent errors in the control calculations a function in the program adjusts the position (chapter 5.1). In the simulations the control has been successful with an error of 2cm in position. With a step length of 0.25 mm, this results in a maximum error of up to 80 steps without disturbing the control.

5. Programming

The program was edited in Borland C++ on a PC with NT-platform and compiled by a free GCC (gnu c-compiler) in UNIX-environment. The choice of language is based on the environment of the ATB-computer. The ATB, computer is part of a fault tolerant computer system designed by SE. The ATB was used to demonstrate the fault tolerant computer concept for the Automatic Transfer Vehicle (ATV), in the frame of the International Space Station (ISS). The ATB is based on an ERC32, a 32-bits Sparc processor, specifically designed for space applications. The processor ERC32 makes it possible to use C or ADA. C has been chosen since it is easier and faster to learn.

5.1 Structure

The structure of the program is found in figure 5.1. The **Poll**-function in **Main** is the one that does the repeatedly check of different events. The events can be a key that has been pushed down or signals from sensors on the cross table. The **Poll**-function also makes the control loop regular, which is important since the system is in real time. The program is built on two options, **Regul** and **Joy**. In **Regul** it is the computer that controls the pendulum and in **Joy** it is possible to control the pendulum by a joystick (not implemented). **Regul** and **Joy** can only be used independent of each other. **Regul** calculates new output values (number of steps the motor should take) from the in signals (angle, angle velocity, plate position and plate velocity). The function **Adjust_Pos** in **Regul** is used to adjust the plate position counted in the program to the actual position counted by SCORPIO. This is done because of the uncertainty of the repeatability of the linear actuators. **Joy** function will convert the operator's motion on the joystick to a number of steps to the motors.

The program assumes that the table is in the middle when it starts control the joystick. Before **Main** can call either of **Regul** or **Joy** it is necessary to center the table. So this is be done in **Center** every time the program is started and when the pendulum has fallen. When neither of **Regul** or **Joy** is active it is possible to change parameters for **Regul** in **Change_Par**. This is an important function if the control function does not work as expected. When the program is working satisfactory the possibility to change parameters will be removed. That will make the program more user friendly for the person who makes the demonstration. The only thing the operator has to choose among is to start **Regul**, **Joy** or to shut down the program.
The program is found in appendix H.

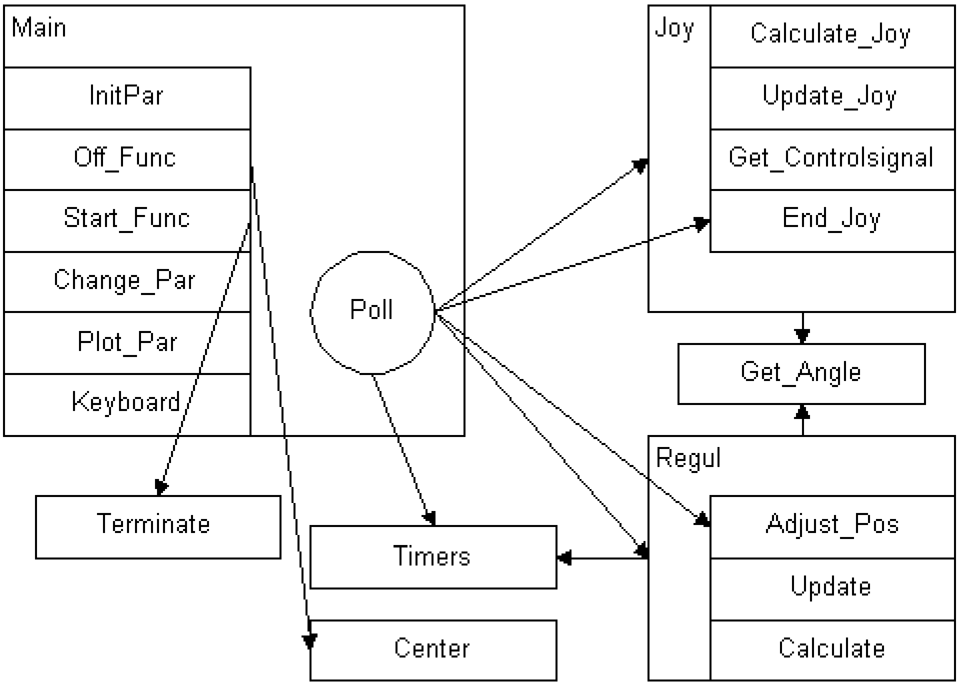


Figure 5.1 Program structure

5.2 Polling

Interrupt handling will not be necessary since the execution time of the program does not exceed 5 ms for each loop, which correspond to the sample time for the program. In every loop events are checked e.g. if the operator has given any command, if the pendulum has passed the center or reached any edge of the table.

SCORPIO will change the output frequency to the motors in the beginning of every new loop. The new value has to lay on the board a short time before next clock pulse arrives, otherwise the board will generate the same frequency as last time. This is a solution to never let the motor stop before a stop signal has been received. If the motor should make a short stop the pendulum is going to fall.

To make the program loop work adequate, the program clock has to be synchronised with SCORPIO's loop clock (figure 5.2). The program will always wait for SCORPIO's loop clock signal to arrive before starting a new loop. If the program loop and the board loop drift apart the pendulum is going to gradually fall.

The program is going to check for events a fixed time before **Regul** is called (figure 5.2). Then can the calculations start as soon as the angle and angel velocity has been read for both directions. The calculations in **Regul** do not take more than 5ms. The velocities and angles for the pendulum should be read as late as possible before the calculation, so that the calculations in **Regul** use as correct values as possible.

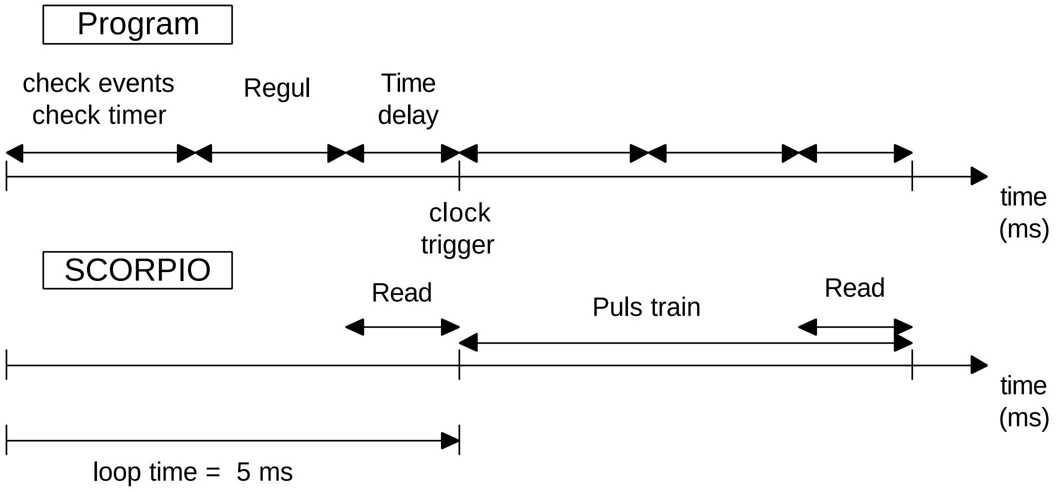


Figure 5.2: Polling loop

6. Conclusion

The main goal with this master thesis was to develop an equipment from specification to a final commission. The total system is not finished because of delays in delivering of the mechanical equipment and problems with the development of SCORPIO.

At this date all the parts of the system are working separately. The simulations have given the demands of the equipment. Modelica has been chosen to simulate the system since e.g. the step motor is very difficult to make model of in Simulink. To simplify the calculations and simulations a universal joint has been chosen instead of a ball joint. A ball joint can be affected by a torque in contrast to the universal joint. That makes the model equations for the ball joint very complicated. The AVM gives out the angle and angle velocity from the pendulum with very little noise. The estimated total noise and time delay lay within the limits.

So the only things that remains are to put all parts together and do the final tests and fine adjustments.

7. Abbreviations

ATB	Avionia Test Bench
ATV	Automatic Transfer Vehicle
AVM	Angle and Velocity Measurements
ISS	International Space Station
LTH	Lund Institute of Technology
SCORPIO	Step motor Controller and Rocket Position IO board
SE	Saab Ericsson Space (AB)
e.g	for example (Latin: exempli gratia)
i.e	that is (Latin: id est)
vs.	versus
Ariane 5	Carrier rocket

8. REFERENCES

Karl J. Åström, Björn Wittenmark; Computer-Controlled Systems, Prentice Hall, 1997.

Ulf Bilting, Jan Skansholm; Vägen till C, Studentlitteratur, 1987.

L.J. Meriam, L.G. Kraige; Engineering Mechanics, Wiley, 1993.

Department of industrial electrical engineering and automation; Elmaskinsystem, 1997.

Östergrens Elmotor AB; Produktkatolog 2000.

Modelica; http://www.dynasim.se, http://www.simulink.org

Bill Messner, Dawn Tilbury, http://tech.buffalostate.edu/ctm/index.html, 1996.

9. Appendix.