

Controlling the Reaction Wheel Pendulum

Geoffrey Chauveau, Damien Chazal, Daisuke Nakayama
Erik Olsen, Staffan Palm

May 25, 2005

Abstract

The goal of this project is to control a complicated mechanical plant. The plant is an inverted pendulum freely swinging around its axis. A DC-motor is attached at its outer end, running a heavy wheel which is freely spinning in the air. The only way to apply torque to the pendulum is to accelerate this wheel.

The motor and the wheel are too weak to swing the pendulum straight to its upright position, that is why energy must be accumulated over several swings to get it upright.

A model of the plant is derived from laws of physics and by doing tests and measurements. The model is verified by simulating step responses and comparing those to step responses of the real plant.

A hybrid automaton switches between two states depending on the position of the pendulum, the swinging state which swings the pendulum to the upright position and the balancing state which keeps the pendulum upright.

The swinging state uses a bang-bang energy controller to ensure the pendulum reaches the upright position with the right amount of speed.

The plant is time-continuous and the controller needs to be implemented in a time-discrete form. Two different approaches were used for the design of the balancing controller, depending on where the discretization was realized.

Linear Quadratic (LQ) design of controllers has been used for both of these approaches.

The controller designed for a continuous plant works fine as long as the sampling frequency is higher than about 15 times the highest natural frequency of the plant. For lower sampling frequencies the discrete designed controllers gave better results. They worked well up to a sampling frequency of about 10 times the highest natural frequency, which fits well with classical rules of thumb.

Contents

Introduction	1
1 The Plant	2
1.1 Description	2
1.2 Input and Output Signals	2
1.3 An Analytical Model	5
1.4 Change of the State-Space Variables	6
1.5 Analysis of the System	7
1.5.1 Investigation of Poles and Zeroes	7
1.5.2 Controllability and Observability	8
1.6 Validating the Analytical Model	8
2 Control System	10
2.1 The Hybrid Automaton	10
2.2 The Swinging State	10
2.2.1 Simple Swing Up	10
2.2.2 Energy Controlled Swing Up	11
2.3 The Balancing State	12
2.3.1 Preliminary Analysis of the Control System	12
2.3.2 Design in Continuous Time	14
2.3.3 Design in Discrete Time	20
3 Implementation and Development Environment	24
3.1 Implementation	24
3.2 User Interface	25
3.2.1 LCD	25
3.2.2 Switches	26
Conclusion	27
Bibliography	29
Appendix	30

A	Tests	30
A.1	Measurement of mgI	30
A.2	Measurement of J_p	30
A.3	Measurement of Pendulum Friction	31
A.4	Measurement of J_w	31
A.5	Motor Driver A3952 Test 1	31
A.6	Motor Driver A3952 Test 2	32
A.7	Wheel Friction Test	32
A.8	Relation Between Output "u" and Voltage to the Motor	34
B	Filtering of Data	35

Introduction

This project is part of the project course in automatic control of the S3 department of the Royal Institute of Technology (KTH) in Stockholm. The goal of this school project is to control a complicated mechanical plant.

The plant is an inverted pendulum freely spinning around its axis moving thanks to a reaction wheel at its outer end.

Our aim is to swing the pendulum to the upright position and keep it stable.

Sensors give information about the angles of the pendulum and the wheel, and the controller is implemented on a Texas Instruments digital signal processor.

The final result will be a completely embedded system, which only needs power supply to run.

One industrial application of reaction wheels is to adjust the angles of satellites in space. This phenomenon is also observed in motorcycling: pilots use the acceleration of their rear wheels to tilt the angle of the motorcycle when they jump.

Chapter 1

The Plant

This chapter describes the pendulum and derives a mathematical model for it.

1.1 Description

The plant is an inverted pendulum freely spinning around its axis. A DC-motor is attached at its outer end. This motor runs a heavy wheel which is freely spinning in the air. The only way to apply a torque to the pendulum is to accelerate this wheel.

The motor and the wheel are too weak to swing the pendulum to its upright position in one swing. They need to add energy to the system with many swings to get it upright.

The system is based on a TI Digital Signal Processor that calculates the output, and an I/O daughter card that makes measurements using the two encoders and controls the DC motor using a PWM motor driver. The control system is built in C language, compiled and downloaded to a DSP memory.

Figure 1.1 shows the plant and some of the parameters.

Following is a complete list of the parameters (see Table 1.1).

1.2 Input and Output Signals

The output from the DSP, u , is a Pulse Width Modulated (PWM) signal. It means that the proportion of time when the signal is high is manipulated. This signal is sent to a motor driver chip (A3952) which transforms it to an analog signal. If this signal is a specific voltage or a specific current has been a subject of confusion. The chip is primarily designed to be a current control chip, though, after studying the circuitry and doing measurements of the output (see A.5 and A.6), it has clearly been shown that the controllable quantity is the voltage.

The relation between the voltage and the PWM signal is unfortunately nonlinear. A nonlinear transfer function has been derived to calculate an PWM signal

θ	The angle of the pendulum with respect to ground, counted from upright position
$\tilde{\theta}$	The angle of the pendulum with respect to ground, counted from downward position
φ	The angle of the wheel with respect to ground
φ_r	The angle of the wheel with respect to the pendulum
ω	The angular velocity of the wheel with respect to ground
ω_r	The angular velocity of the wheel with respect to the pendulum
e	The back emf voltage from the motor
f	The torque between the wheel and the pendulum due to wet friction, ~ 0 Ns/rad
g	The gravity constant, 9.81 m/s^2
i	The current through the motor
J_p	The moment of inertia of the pendulum, 0.0045 kg.m^2
J_w	The moment of inertia of the wheel and rotor, $2.13 \cdot 10^{-5} \text{ kg.m}^2$
k_E	The back emf constant of the motor, $27.4 \cdot 10^{-3} \text{ V.s/rad}$
k_T	The torque constant of the motor, $27.4 \cdot 10^{-3} \text{ N.m/A}$
l	The radius of the center of mass on the pendulum, including wheel, ($mgl = 0.344 \text{ N.m}$)
L	The inductance of the motor, 6.27 mH
m	The mass of the pendulum, including wheel, ($mgl = 0.344 \text{ N.m}$)
R_T	The resistance of through the motor 12.1Ω
t	Time
T_g	The torque on the pendulum due to gravity
T_m	The torque applied by the reaction wheel to the pendulum
u	The output from the DSP, -10 to 10 (corresponds to the duty cycle of the PWM signal)
v	The voltage applied to the motor

Table 1.1: Variables and parameters of the project

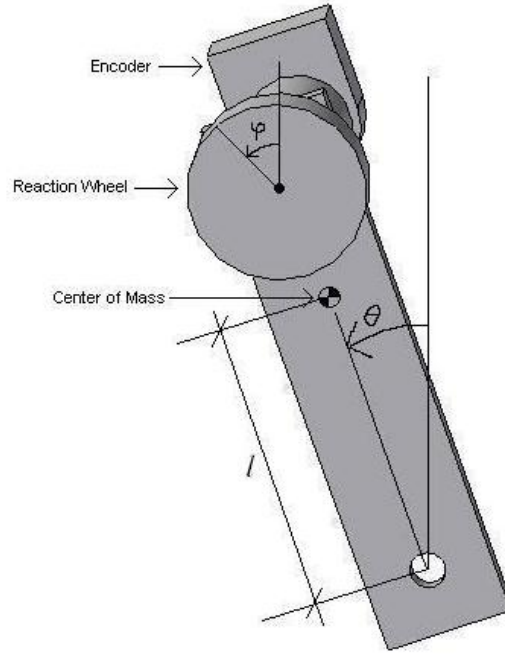


Figure 1.1: The different parameters of the plant.

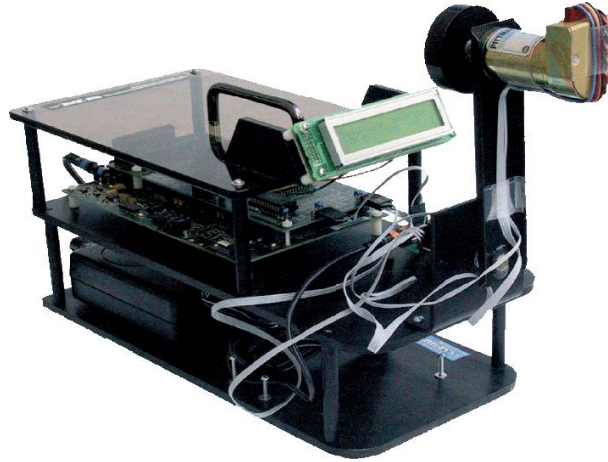


Figure 1.2: A picture of the whole plant.

from a demanded voltage (see A.8). The output from the controller should then be a voltage.

There are two output signals from the plant. The angle of the pendulum (θ) and the angle of the wheel (φ). The signals we are interested in for our controller are the angle of the pendulum (θ), the angular velocity of the pendulum ($\dot{\theta}$) and the angular velocity of the wheel ($\omega = \dot{\varphi}$).

The angular velocities are derived by differentiating the angles. The Euler backwards method is used for this. The angular velocities need to be filtered to reduce noise. The filter design is presented in Appendix B.

The angles of the wheel φ_r and the pendulum θ can be measured from two embedded encoders: the encoder on the motor can measure the angle of the wheel φ_r , and the optical shaft encoder of the pendulum measures its angle θ .

- DC Servo Motor
 - Reference Voltage: 24 V
 - Encoder Resolution: 1000 CPR
- Optical Shaft Encoder
 - Encoder Resolution: 1000 CPR

1.3 An Analytical Model

The motor is a DC-motor whose equations are:

$$v = R_T i + L \frac{di}{dt} + e \quad (1.1)$$

$$e = k_E(\omega - \dot{\theta}) \quad (1.2)$$

$$-T_m = k_T i \quad (1.3)$$

The time constant for this inductance-resistor system equals to

$$\tau_L = \frac{L}{R} = 0.518 \text{ ms} \quad (1.4)$$

which is very much faster than the other dynamics in the plant. The fastest pole of the remaining system lies, according to 1.5.1, in -8.7473, which corresponds to a time constant of 114 ms. Hence this inductance is neglected.

The motion of the wheel and the pendulum is given by the torque equation:

$$J_w \ddot{\varphi} = J_w \dot{\omega} = -T_m - f(\omega - \dot{\theta}) \quad \text{for the wheel} \quad (1.5)$$

$$J_p \ddot{\theta} = T_m + T_g + f(\omega - \dot{\theta}) \quad \text{for the pendulum with} \quad (1.6)$$

$$T_g = mgl \sin \theta \quad (1.7)$$

and the friction between the pendulum and ground is neglected according to A.3.

With the state variables isolated the equations become

$$\dot{\omega} = - \left(\frac{f}{J_w} + \frac{k_E k_T}{J_w R_T} \right) \omega + \left(\frac{f}{J_w} + \frac{k_E k_T}{J_w R_T} \right) \dot{\theta} + \frac{k_T}{J_w R_T} v \quad (1.8)$$

$$\ddot{\theta} = \left(\frac{f}{J_p} + \frac{k_E k_T}{J_p R_T} \right) \omega + \frac{lm g}{J_p} \sin \theta - \left(\frac{f}{J_p} + \frac{k_E k_T}{J_p R_T} \right) \dot{\theta} - \frac{k_T}{J_p R_T} v \quad (1.9)$$

The system can be linearized and described in state-space form with state $x_a = (\omega, \theta, \dot{\theta})^T$ and input voltage v .

$$\dot{x}_a = A_a x_a + B_a v \quad (1.10)$$

where

$$B_a = \begin{pmatrix} \frac{k_T}{J_w R_T} \\ 0 \\ -\frac{k_T}{J_p R_T} \end{pmatrix} \quad (1.11)$$

and A_a depends on the linearization point. Linearizations around the two equilibrium points give:

$$\begin{array}{l} \theta \text{ around } 0 \\ (\sin \theta \sim \theta) \end{array} : A_a = \begin{pmatrix} -\frac{f}{J_w} - \frac{k_E k_T}{J_w R_T} & 0 & \frac{f}{J_w} + \frac{k_E k_T}{J_w R_T} \\ 0 & 0 & 1 \\ \frac{f}{J_p} + \frac{k_E k_T}{J_p R_T} & \frac{lm g}{J_p} & -\frac{f}{J_p} - \frac{k_E k_T}{J_p R_T} \end{pmatrix} \quad (1.12)$$

$$\begin{array}{l} \theta \text{ around } \pi \\ (\sin \theta \sim -(\theta - \pi)) \\ \theta \text{ substituted to } \tilde{\theta} \text{ according to} \\ \tilde{\theta} = \theta - \pi \end{array} : A_a = \begin{pmatrix} -\frac{f}{J_w} - \frac{k_E k_T}{J_w R_T} & 0 & \frac{f}{J_w} + \frac{k_E k_T}{J_w R_T} \\ 0 & 0 & 1 \\ \frac{f}{J_p} + \frac{k_E k_T}{J_p R_T} & \frac{-lm g}{J_p} & -\frac{f}{J_p} - \frac{k_E k_T}{J_p R_T} \end{pmatrix} \quad (1.13)$$

Different tests are performed to find out the values of the parameters, see A

1.4 Change of the State-Space Variables

The linearized model has been derived with absolute angles. However, to simplify the design of the controller with respect to the measured signals from the sensors, the use of the relative speed of the wheel is more relevant. Thus a change of the state variable ω must be done: $\omega_r = \omega - \dot{\theta}$.

If the state equation is denoted in this way:

$$\begin{pmatrix} \dot{\omega} \\ \dot{\theta} \\ \ddot{\theta} \end{pmatrix} = A_a \begin{pmatrix} \omega \\ \theta \\ \dot{\theta} \end{pmatrix} + B_a v$$

then the modified state space equation with ω_r is:

$$\begin{pmatrix} \dot{\omega}_r \\ \dot{\theta} \\ \ddot{\theta} \end{pmatrix} = PA_a \begin{pmatrix} \omega_r \\ \theta \\ \dot{\theta} \end{pmatrix} + PB_a v \quad \text{with } P = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

New variables are defined according to $x = (\omega_r, \theta, \dot{\theta})^T$, $A = PA_a$ and $B = PB_a$.

The system is now described by

$$\dot{x} = Ax + Bv \quad (1.14)$$

1.5 Analysis of the System

1.5.1 Investigation of Poles and Zeroes

Our system has three poles and one zero. Figure 1.3 shows the location of the poles and zeroes for the linearization around the unstable equilibrium point. The pole in the right half plane shows that the system is unstable. The poles are located in (8.7164, -8.7473, -2.9093) and the zero is in the origin.

The two poles that are symmetrical around the origin are due to the inverted pendulum and the third pole (-2.9093) is due to the reaction wheel.

The zero can be explained by the fact that the wheel needs to be accelerated to get a torque on the pendulum, which requires changes in the voltage.

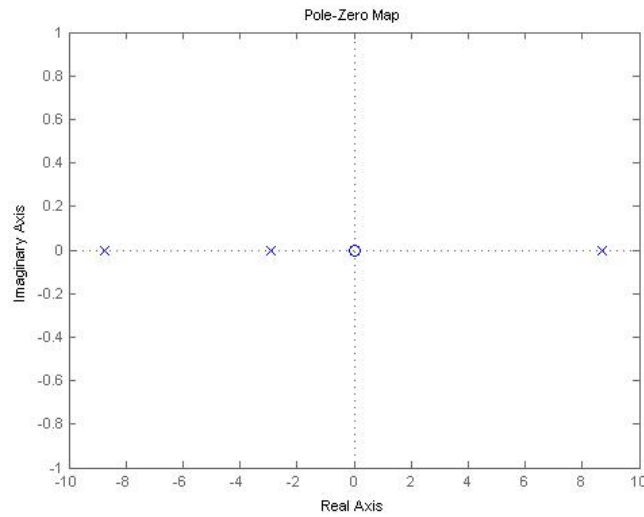


Figure 1.3: Poles and Zeros of the linearized model in the upright position.

1.5.2 Controllability and Observability

To be able to fully control the system it needs to be controllable. Therefore, the controllability matrix Q_C is calculated.

$$Q_C = [B \quad AB \quad A^2B] = \begin{pmatrix} 106.8041 & -314.0222 & 961.4548 \\ 0 & -0.5013 & 1.4739 \\ -0.5013 & 1.4739 & -42.5101 \end{pmatrix}$$

Since $\det Q_C \neq 0$, the process is fully controllable. All the states can then be controlled with the input signal v .

θ is measured directly with an encoder. The two other states, $\dot{\theta}$ and ω_r are derived by differentiation (and filtering) of the sensor signals. Obviously our plant is observable as well.

1.6 Validating the Analytical Model

The analytical model derived in 1.3 will be verified by comparing its step response to a step response from the plant. Figure 1.4 shows the output signals θ and ω_r for both the model and the process for a 10 volt step. Figure 1.5 shows the same data for a 5 volt step.

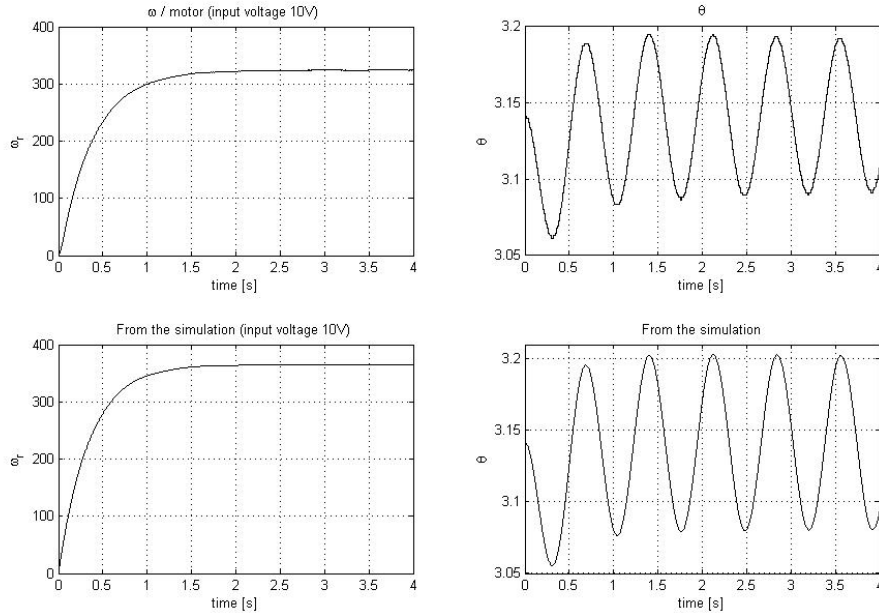


Figure 1.4: Output signals from model and process for a step of 10 volts.

The behaviours of the systems are very similar. The steady state angular velocity of the wheel ω_r is always about 40 rad/s lower in the process than in

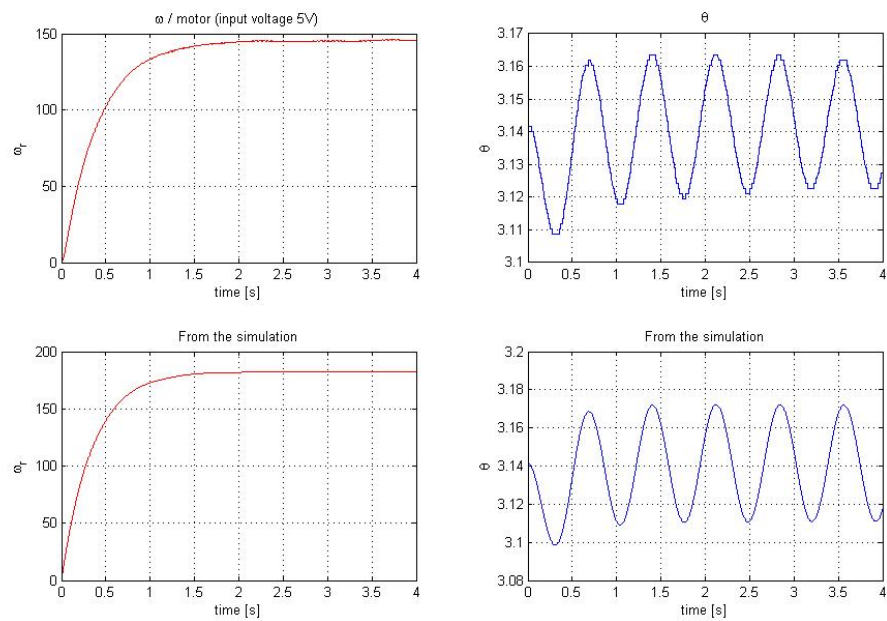


Figure 1.5: Output signals from model and process for a step of 5 volts.

the model, independent of the level of the input voltage, probably because of the neglected dry friction.

Chapter 2

Control System

The control system switches between different states. The hybrid automaton and the different controllers will be described in this chapter.

2.1 The Hybrid Automaton

The controller is implemented as an hybrid automaton that switches between two states during normal operation; the swinging state to swing the pendulum up and the balancing state to keep the pendulum in the upright position.

As Figure 2.1 shows, the active state depends on the position of the pendulum. The swinging state is active from the initial downward position until the pendulum enters the catch area around the upper position, where the balance controller is activated instead. The possible size of the catch area depends on the strength of the balance controller and the precision of the swing up function.

Tests showed that a catch area starting 8 degrees from the top position worked fine.

Extra features can also be seen as states, for example the brake to return the pendulum to the downward position or to prevent the pendulum from damaging the cables.

2.2 The Swinging State

This chapter describes two ways of swinging up the pendulum.

2.2.1 Simple Swing Up

The aim of this controller is to make the pendulum go up as fast as possible. A solution is to give maximum torque to the pendulum to increase its speed in the direction it is going. The following controller was built:

$$u = -9 \quad \text{when } \dot{\theta} \geq 0$$

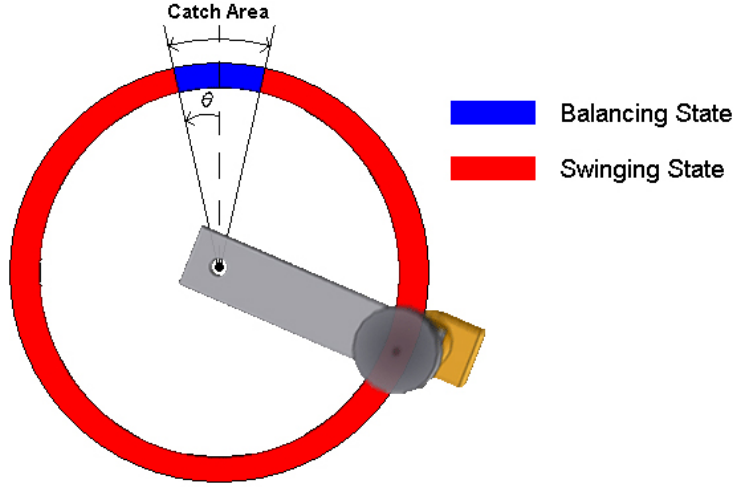


Figure 2.1: The two states depending on pendulum angle.

$$u = 9 \quad \text{when } \dot{\theta} < 0$$

It works pretty good but the angular velocity of the pendulum is often too high when it enters the catch area.

2.2.2 Energy Controlled Swing Up

To be able to make the pendulum arrive at its upright position with the right amount of speed, the total energy of the pendulum is monitored. For every sample, the energy of the pendulum is calculated as $E = \frac{1}{2}J_p\dot{\theta}^2 + mgl \cos \theta$.

To validate the energy calculations, the total energy of a freely swinging pendulum was observed. Figure 2.2 shows the result. As expected, it was almost constant, decreasing slowly due to friction.

The target value for the total energy is the potential energy of the pendulum at the top position

$$E_0 = mgl$$

The controller is a simple bang-bang controller, accelerating when the energy is below the target value and braking when the energy is too high.

The final swing controller is:

	$E < E_0$	$E \geq E_0$
$\dot{\theta} \geq 0$	u=-9	u=9
$\dot{\theta} < 0$	u=9	u=-9

The high braking values are useful when the pendulum is put to higher energy levels because of external disturbances.

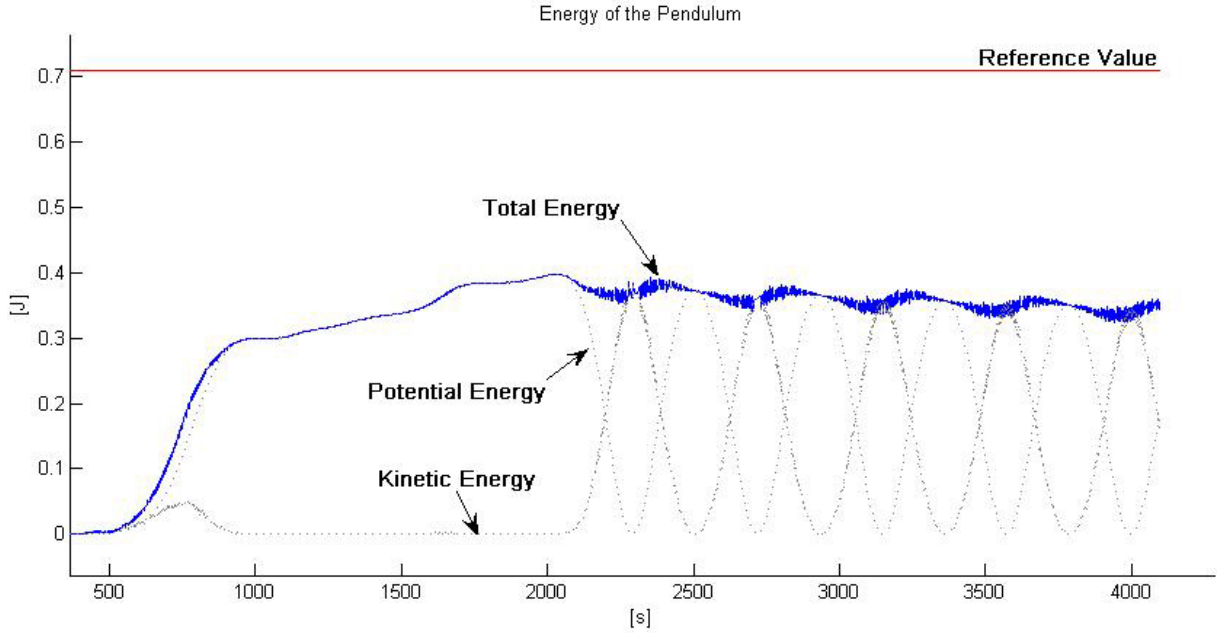


Figure 2.2: Energies of the pendulum when lifted and then released to swing freely.

2.3 The Balancing State

After being moved by the swinging process to its upright position, the pendulum is supposed to be in the linearized region around $\theta \sim 0$. Thus in the following section, the plant refers to the linear model that was derived previously for the upright position.

2.3.1 Preliminary Analysis of the Control System

Analysis of the Closed-Loop System

In our case, the plant is time-continuous and the implemented controller must be time-discrete since it is based on the sampled measurements of the system outputs. The control system of the plant can be represented by the block diagram on Figure 2.3.

The input signal is the voltage v and the output of the plant is $y = (\phi_r, \theta)^T$. The measurement of y consists of a uniform sampling at the sampling frequency $\frac{1}{h}$. In the feedback loop the controller computes a discrete command signal $v[k]$ from the discrete outputs up to $y[k]$. Finally, a continuous input signal $v(t)$ is generated after a zero-order hold circuit.

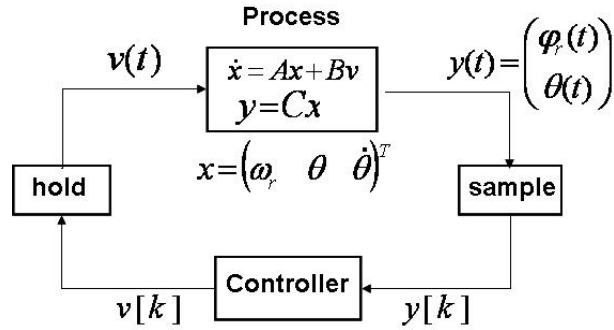


Figure 2.3: General block diagram of the controlled plant

Two Approaches of Controller Design

Two different methods are then available for the design of the discrete controller (see figure 2.4): a *continuous-time design* and a *discrete-time design*. The first one consists in building a continuous controller from the continuous model of the plant, and then discretizing it. The second one requires a preliminary discretization of the plant in order to build a completely discrete controller. These two approaches will be studied and compared in the next parts.

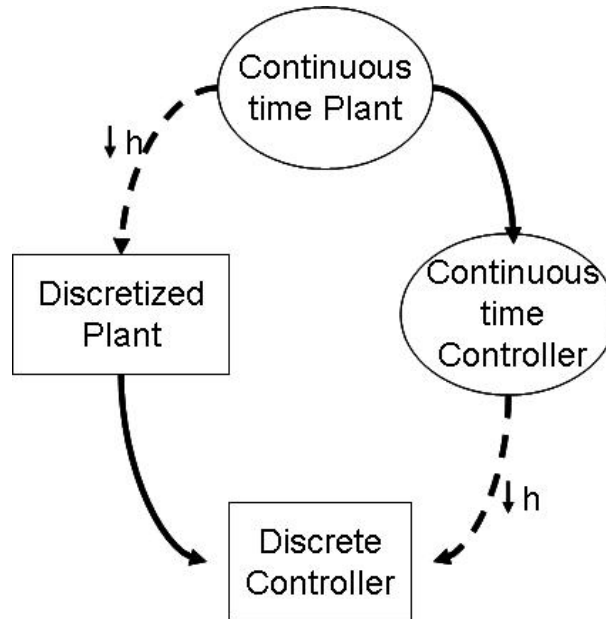


Figure 2.4: Design in continuous time and in discrete time.

2.3.2 Design in Continuous Time

In this part a controller based on the continuous plant is studied. This controller calculates the command signal v from the two outputs of the system ϕ_r and θ . However, an efficient controller should also depend on ω_r and $\dot{\theta}$. Therefore, state feedback controllers have been studied in this project.

The State Feedback Controller

In order to estimate the state variables $x = (\omega_r, \theta, \dot{\theta})^T$, the angles ϕ_r and θ have to be differentiated in the feedback loop. This operation is represented by the transfer function $F(s)$ called a *state observer*, even though it is not really an observer. The feedback controller is called $C(s)$ and is a linear combination of the state variables:

$$v = k_\omega \omega_r + k_\theta \theta + k_{\dot{\theta}} \dot{\theta}$$

Figure 2.5 shows a block diagram of the continuous-time control system. The part that will be implemented on the real system corresponds to the dashed-line rectangle. The discretization is realized by approximating the derivatives in $F(s)$ with an Euler backward approximation: $s = \frac{z-1}{zh}$. The discrete controller is represented in the block diagram in Figure 2.6.

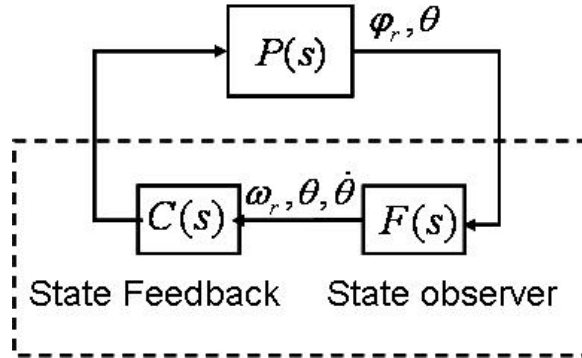


Figure 2.5: Block diagram of the continuous-time state feedback control system.

PD Controller

Since the variable of interest is θ , it is natural to investigate a controller in this form: $v = k_\theta \theta + k_{\dot{\theta}} \dot{\theta}$, which is similar to a PD controller of the angle θ .

In order to determine the coefficients $(k_\theta, k_{\dot{\theta}})$, a pole placement analysis is performed. First, let us consider the transfer function $\frac{\Theta(s)}{V(s)}$, from the input voltage to the pendulum angle:

$$\frac{\Theta(s)}{V(s)} = \frac{-0.5013s}{s^3 + 2.94s^2 - 76.15s - 221.8}$$

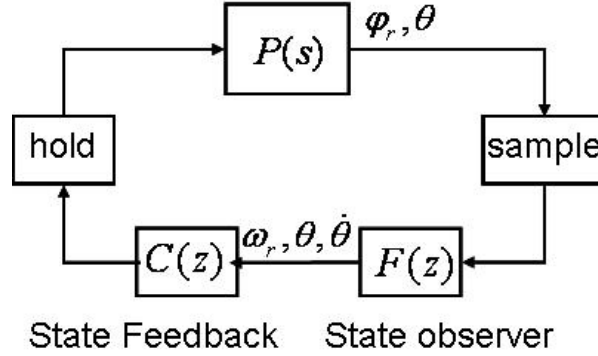


Figure 2.6: Block diagram of the real state feedback control system.

Generally speaking, since it is a single-input system, the linearized system around the upright position can lead to three separated transfer functions from the input voltage v to each state variable ω_r , θ and $\dot{\theta}$.

The plant can now be considered as a single-input single-output (SISO) system. The derivative part of the controller ($k_{\dot{\theta}}$) introduces a zero on the real axis. If $k_{\dot{\theta}}$ is fixed, it is possible to plot the root locus of the closed-loop system, that is to say the closed-loop pole trajectories as a function of k_{θ} . Figure 2.7 shows the root locus when $k_{\dot{\theta}} = 6.6$. The controller's zero is represented by a triangle whereas the poles and zeroes of the plant are respectively crosses and circles. The three squares are the three closed-loop poles showing the location of the three different pole trajectories. The root locus shows that wherever the PD zero is placed, the closed-loop system will remain unstable due to the closed-loop pole on the positive real axis.

Complete State Feedback Controller

A more complete feedback controller must be used, using a feedback on the speed of the wheel ω_r . Such a controller can be written as follows: $v = k_{\omega}\omega_r + k_{\theta}\theta + k_{\dot{\theta}}\dot{\theta}$

In order to keep on analyzing the SISO system represented by the transfer function $\frac{\Theta(s)}{V(s)}$, the feedback on ω_r is included in the state space representation:

$$\dot{x} = (A - BL)x + Bv, \text{ with } L = (k_{\omega} \quad 0 \quad 0)$$

It is then possible to plot the root locus when $k_{\omega} = 0.1$ and $k_{\dot{\theta}} = 6.6$ (see Figure 2.8). The closed loop system is now stable because all the closed-loop poles can be moved to the left part of the real axis. It is important to notice that the plant now has two unstable poles, which is due to the feedback loop on ω_r .

The following state feedback controller will be evaluated regarding stability and speed:

$$v = 0.1\omega_r + 400\theta + 60\dot{\theta}$$

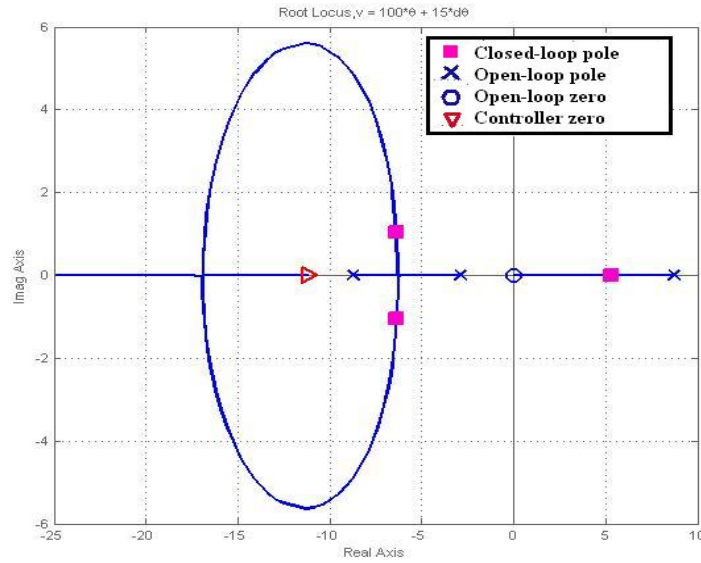


Figure 2.7: Root locus of the closed-loop system using a PD controller, $k_{\dot{\theta}} = 6.6$.

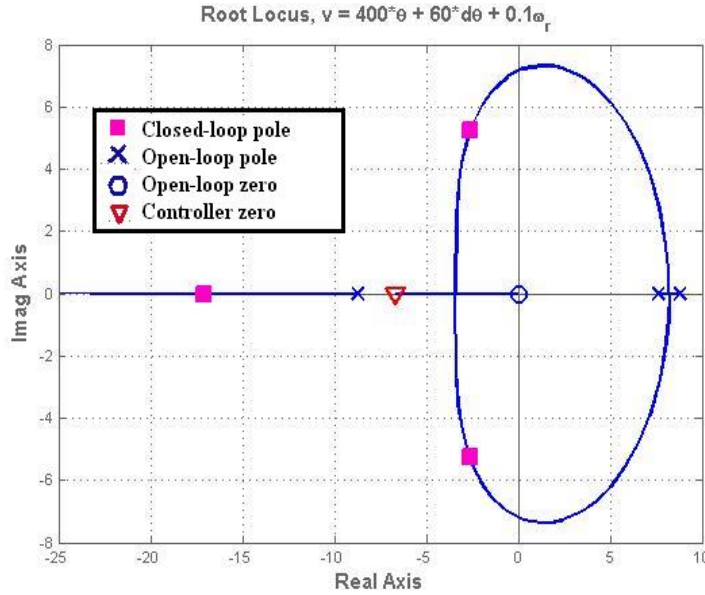


Figure 2.8: Root locus of the closed-loop system using a feedback controller, $k_{\dot{\theta}} = 6.6$ and $k_{\omega} = 0.1$.

First of all, the Nyquist diagram of the open-loop is illustrated in Figure 2.9. According to the Nyquist criterion, the closed loop system is stable since there are two turns around -1, which is the exact number of positive real-part poles. Furthermore, the gain and the phase margins are respectively $G_m = 3.93dB$ and

$$\phi_m = 41.2^\circ.$$

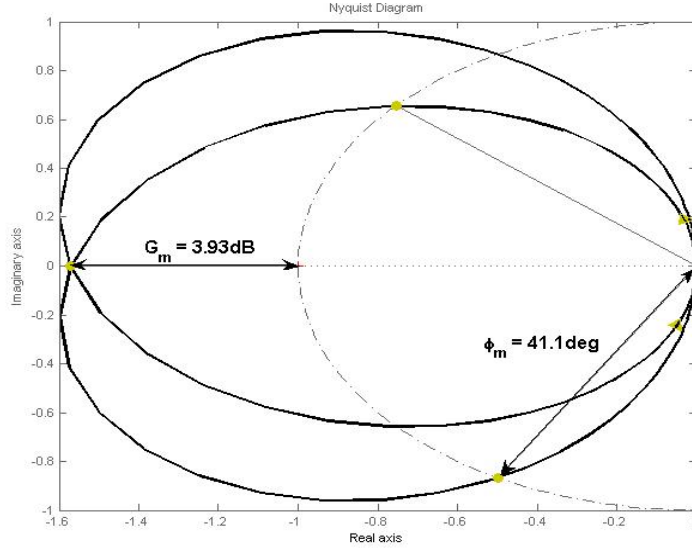


Figure 2.9: Nyquist diagram of the state feedback controller: $v = 0.1\omega_r + 400\theta + 60\dot{\theta}$.

In the following acquisitions, the sampling period h is 1 ms. In Figure 2.10, the response to a step disturbance of 15V is illustrated. This experiment is done when the pendulum is stabilized around $\theta = 0$. A disturbance signal of 15V is then added to the control signal, making the pendulum move away from its equilibrium point. The simulated and measured responses have the same behaviour except for the final overshoot which is much higher for the real system. This is probably due to unmodelled nonlinearities. The conclusion is that in order to increase stability, it is necessary to avoid overshoots, since they become higher in reality.

LQ Controller

A method to design the coefficients $(k_\omega, k_\theta, k_{\dot{\theta}})$ for optimal performance is to use a Linear Quadratic (LQ) controller. Such a control law v is derived by minimizing a cost function depending on the weights Q , R and N :

$$J(v) = \int_0^\infty (x^T Q x + v^T R v + 2x^T N v) dt$$

where x and v are linked to the state space equation : $\dot{x}(t) = Ax(t) + Bv(t)$. Q and R are chosen in this way:

$$Q = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad R = 100 \quad N = 0$$

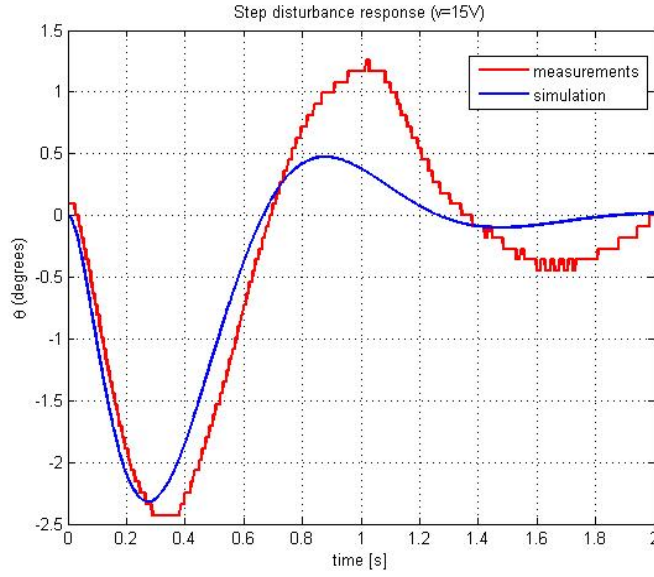


Figure 2.10: Step disturbance response (15V) for $v = 0.1\omega_r + 400\theta + 60\dot{\theta}$

The choice of Q lies on the fact that more emphasis should be given to θ and ω . R is just a scaling factor and is set to 100 in order to avoid having a too high command voltage v that could saturate. Finally, N is equal to 0 because the cost due to the correlation between x and v has been neglected. For linear systems, there is a unique solution $v = -Kx$ to this minimization which is expressed as the solution to the Riccati equation:

$$K = R^{-1}B^TS \quad \text{with} \quad SA + A^TS + Q - SQR^{-1}B^TS = 0$$

The numerical solution is then:

$$v = 0.1311\omega + 688.1082\theta + 78.9278\dot{\theta}$$

The Nyquist diagram is shown in Figure 2.11. The gain margin $G_m = 6dB$ is much larger in this case and the phase margin $\phi_m = 60^\circ$ is a characteristic value for LQ controllers. The simulated and measured responses to a step disturbance of 15V illustrate this improved stability of the system (see Figure 2.12). The sampling period is still $h = 1ms$.

The LQ-controller that has been designed from the continuous plant leads to stable results as long as the sampling period is very small (\approx up to 50 ms as shown in Figure 2.13). This is due to the fact that the coefficients ($k_\omega, k_\theta, k_{\dot{\theta}}$) do not depend on the sampling period, making the control system unadapted to every h . A way to tackle this problem is to take into account the transfer function of the zero-order-hold circuit ($\frac{1-e^{-sh}}{sh}$) (see Figure 2.3) for the design of the continuous-time controller. Then $C(s)$ should be based on $P(s)\frac{1-e^{-sh}}{sh}$

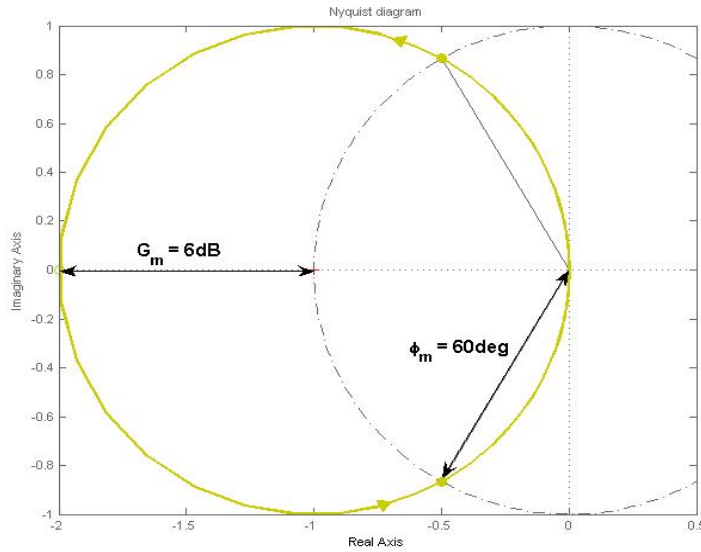


Figure 2.11: Nyquist diagram for the LQ-controller.

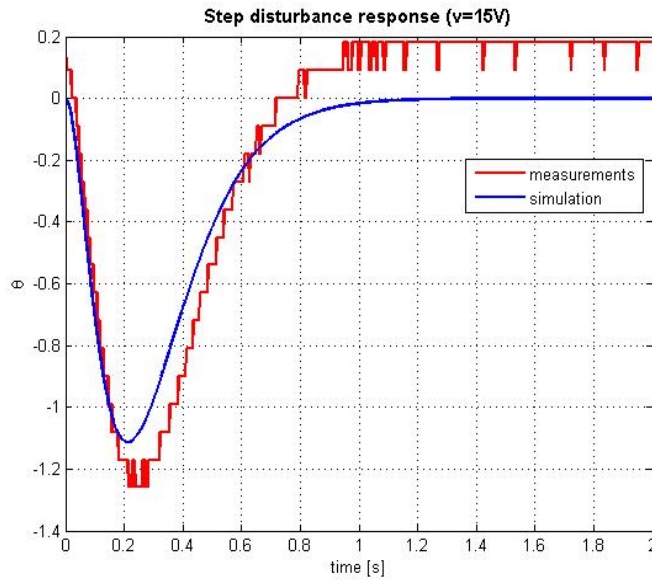


Figure 2.12: Step disturbance response (15V) for the LQ-controller.

instead of $P(s)$. Unfortunately, this is not possible in our multivariable case because a single transfer function for the whole system $P(s)$ can not be extracted. Consequently, the discrete approach of controller design will be investigated in order to handle different sampling periods.

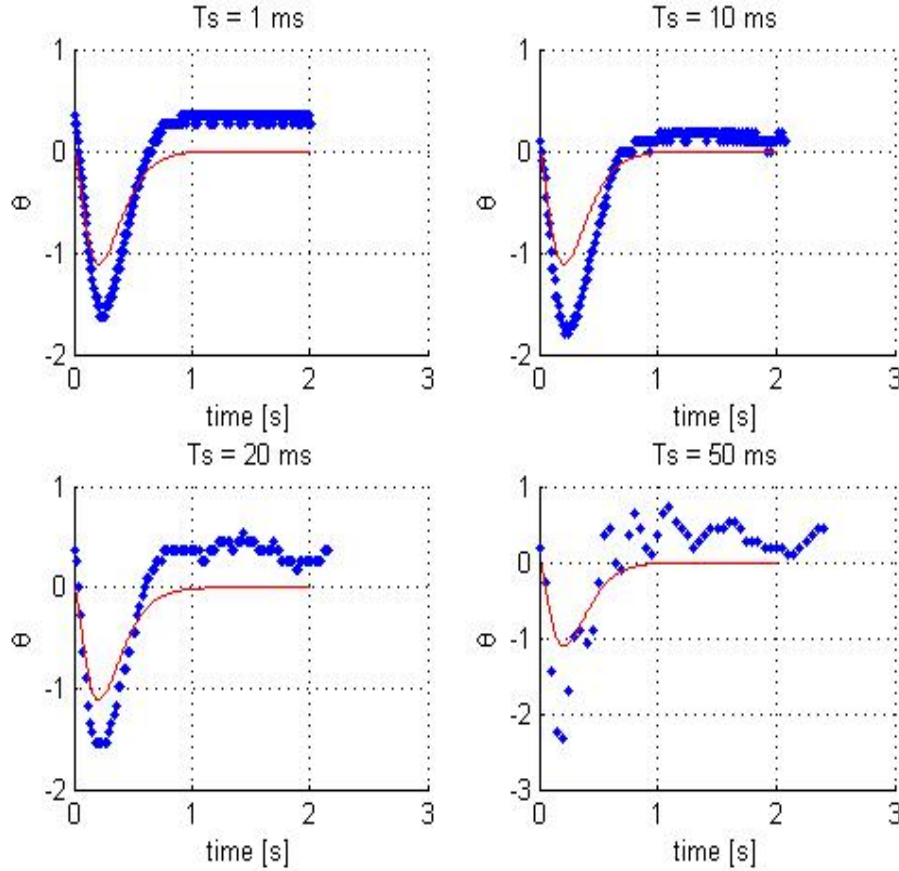


Figure 2.13: Step disturbance responses (15V) for the LQ-controller for different sampling periods.

2.3.3 Design in Discrete Time

Instead of estimating a discrete controller from a continuous one, another approach to control the pendulum is to build a discrete controller directly from the discrete model of the pendulum.

Discretized Plant

Around the equilibrium point, the system is almost linear, with the state-space equation $\dot{x} = Ax + Bv$. The plant can be sampled with a zero-order hold as input (since the computed input sent by Code Composer Studio is constant during the sampling period) and the resulting discrete system is:

$$x[k+1] = A_d(h)x[k] + B_d(h)v[k]$$

with $A_d(h) = e^{Ah}$, $B_d(h) = \int_0^h e^{As}Bds$ and h the sampling period

$h[\text{ms}]$	k_{ω_r}	k_θ	$k_{\dot{\theta}}$
1	-0,12977	-682,35	-78,267
10	-0,11873	-633,95	-72,712
50	-0,083278	-476,62	-54,655
70	-0,071475	-423,13	-48,515
80	-0,066614	-400,83	-45,957
90	-0,062329	-381	-43,681
100	-0,058544	-363,34	-41,654
120	-0,052235	-333,47	-38,228
150	-0,045205	-299,38	-34,315

Table 2.1: Coefficients of the LQ controller for different sampling periods.

Discrete LQ Controller

After discretizing the plant, a discrete LQ controller can be designed. This controller is such that the cost function

$$J_d(v) = \sum_{n=1}^{\infty} x[n]^T Q_d x[n] + v[n]^T R_d v[n] + 2x[n]^T N_d v[n]$$

is minimized. Since we want the same properties as the continuous LQ-controller, the matrix Q_d , R_d and N_d are obtained by discretizing the matrix Q , R and N of the previous controller:

$$\begin{pmatrix} Q_d & N_d \\ N_d^T & R_d \end{pmatrix} = \int_0^h \begin{pmatrix} A_d(\tau) & 0 \\ B_d(\tau) & I \end{pmatrix} \begin{pmatrix} Q & N \\ N^T & R \end{pmatrix} \begin{pmatrix} A_d(\tau) & B_d(\tau) \\ 0 & I \end{pmatrix} d\tau$$

With Matlab, different controllers $v = -\left(k_{\omega_r}\omega_r + k_\theta\theta + k_{\dot{\theta}}\dot{\theta}\right)$ are easily obtained for different sampling times (see Table 2.1).

After implementing this controller for different sampling periods, we observed that it works quite good up to 80 ms and begins to be unable to control the pendulum after a sampling period of 120 ms. Then, for higher sampling periods, the limit of the sampling is reached: a rule of thumb says indeed that h should be chosen such that $0.2 < \omega_0 h < 0.6$, with ω_0 the natural frequency of the system (here $\omega_0 = 8.75$ rad/s). Therefore, we should not use sampling periods higher than 69 ms, which can explain the problems we have with high sampling periods.

Then, some tests were done to investigate the stability of these controllers: a step of 5V is added to the calculated input of the system. We clearly observed that the system is less stable and oscillates more for high sampling periods (see Figure 2.14). The amplitude of the oscillations seems to be about proportional to the sampling time for such low sampling frequencies.

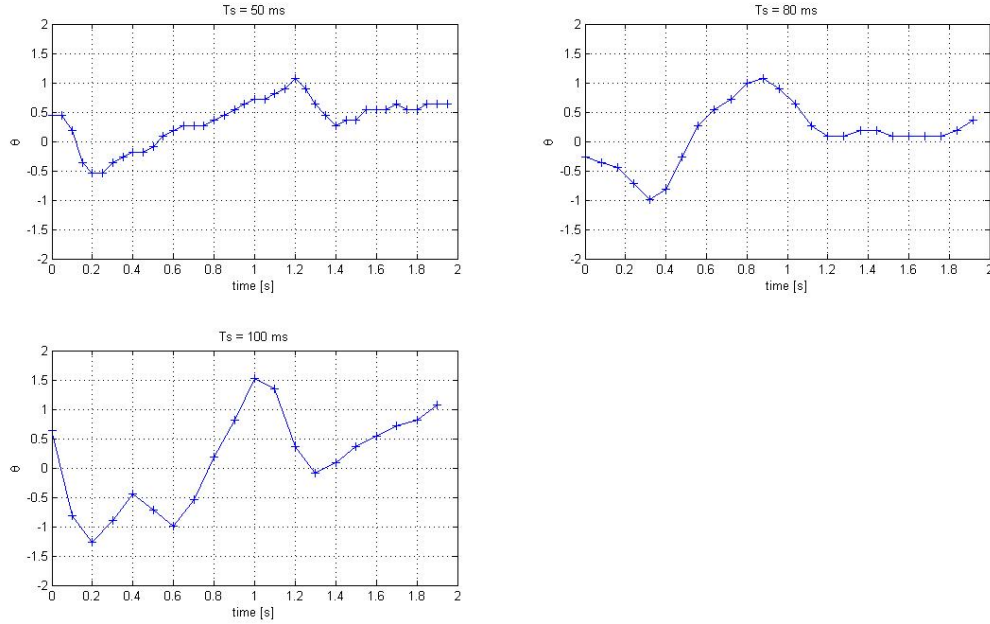


Figure 2.14: Stability of different discrete LQ controllers with sampling periods of 50ms, 80ms and 100ms.

Deadbeat Controller

Another idea was to build a completely discrete controller that has no equivalent in the continuous time-domain. The Deadbeat controller is a discrete controller that drives the state to the origin in at most n steps, with n the number of states variables (here $n = 3$). The controller is a state feedback $v[k] = -Lx[k]$. The state space equation then becomes:

$$x[k+1] = (A_d - B_d L)x[k] = A_c(L)x[k]$$

If the characteristic polynomial of $A_c(L)$ is $P(X)$, then we have

$$P(A_c(L)) = 0_{\mathcal{M}_3(\mathbb{R})} \quad (\text{Cayley-Hamilton theorem})$$

The DeadBeat controller is such that all closed-loop poles are put at the origin (ie $P(X) = X^3$). Therefore, $(A_c(L))^3 = 0$ and

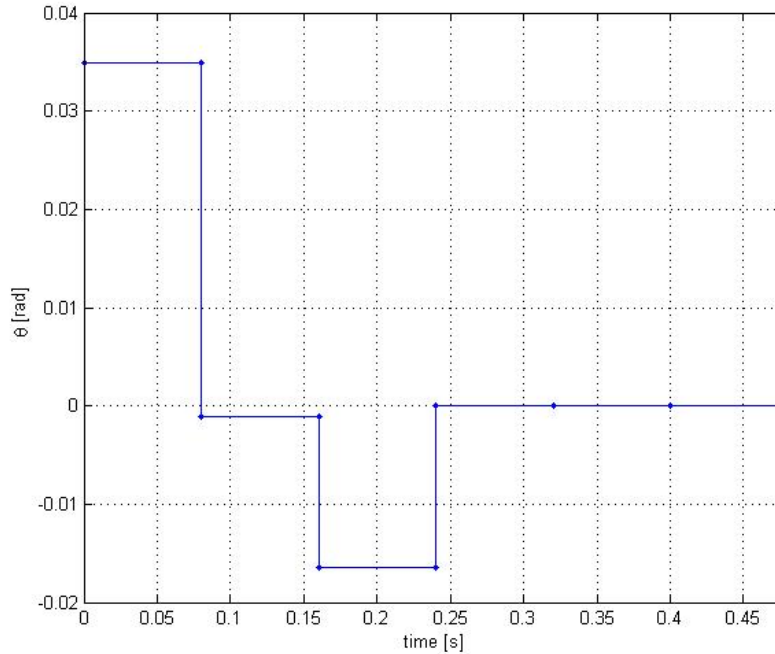
$$x[n] = (A_c(L))^n x[0] = 0 \quad \text{if } n \geq 3$$

With Maple and Matlab, different matrices $L = (l_{\omega_r}, l_{\theta}, l_{\dot{\theta}})^T$ were derived depending on the sampling period h (see Table 2.2).

A simulation is done for $h = 80\text{ms}$ and the controller seems to work perfectly: θ goes to zero in three steps (see Figure 2.15). However the coefficients for low

$h[\text{ms}]$	l_{ω_r}	l_{θ}	$l_{\dot{\theta}}$
1	$-1,2371 \cdot 10^5$	$-3,9951 \cdot 10^6$	$-2,636 \cdot 10^7$
10	-125,3	-40564	-27060
50	-1,0733	-1838,3	-301,96
70	-0,4137	-1028,5	-141,17
80	-0,28741	-829,4	-108,06
90	-0,21064	-692,39	-86,94
100	-0,16126	-594,13	-72,674
120	-0,10489	-465,84	-55,195
150	-0,066718	-360,94	-41,877

Table 2.2: Coefficients of the Deadbeat controller for different sampling periods.

Figure 2.15: Simulation of the Deadbeat controller with $h=80\text{ms}$.

sampling periods are not realistic: the input voltage must reach several hundreds of volts to stabilize the system rapidly (in a few milliseconds). In practice, a sampling period of 80ms is the minimum for the input signal not to saturate. The problem is that the pendulum is not stable for a long time: after a few seconds, the pendulum falls down. This comes from the fact that this controller is very fast and therefore not very stable. Moreover, the sampling period is certainly too high to have an efficient controller ($h > 69 \text{ ms}$).

Chapter 3

Implementation and Development Environment

3.1 Implementation

The implementation of the controller has been realized in Code Composer Studio. This software is used to compile a C-language program and then load it on a DSP card. We were given a collection of basic functions used for the interface with the daughter card but except these functions, everything has been implemented right from the beginning.

All the control algorithms are performed in a periodic function. This periodic function runs every millisecond and has the following structure:

- Tasks in the Periodic Function
 1. Set the previous values of Theta and Phi
 - $\text{ThetaPrevious} = \text{Theta}$
 - $\text{PhiPrevious} = \text{Phi}$
 2. Get data theta and phi from encoders
 - $\text{ReadFromEncoders}(\text{Theta}, \text{Phi})$
 3. Calculate angular velocities
 - $\text{DTheta} = (\text{Theta} - \text{ThetaPrevious}) / h$
 - $\text{DThetaFiltered} = \text{filter}(\text{DTheta})$
 - $\text{DPhi} = (\text{Phi} - \text{PhiPrevious}) / h$
 - $\text{DPhiFiltered} = \text{filter}(\text{DPhi})$
 4. Calculate the output
 - $\text{Output} = \text{controller}(\text{DphiFiltered}, \text{Theta}, \text{DThetaFiltered})$
 5. Send the output signal to the motor
 6. Write information on the LCD

The sampling period is increased artificially by making the periodic function enter an idle loop for a certain time. The reason is that it is then possible to use different sampling periods only by pressing a switch. Of course the periodic function can be specified to run every sampling period in order to study the real effects of the sampling rate on computation time. However, since the CPU-load was always below 5% , it was useless to consider such limitations.

3.2 User Interface

An interaction with the system was made possible thanks to a set of switches and LEDs whose actions were parameterized in the software. Moreover, several buffers and log files have been used for data acquisitions since they were not performed in real time.

3.2.1 LCD

The display is the main part of the user interface and has also been used for debugging. The state of the hybrid automaton, the controller name, the angle of the pendulum, the output voltage and the sampling period are shown on the LCD (see Figure 3.1)

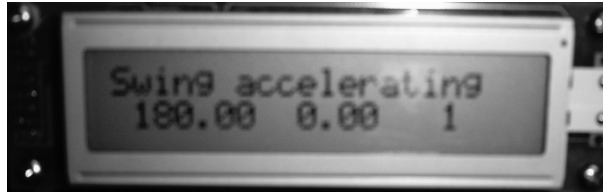


Figure 3.1: User interface: controller, angle θ , input voltage v , sampling period [ms].

3.2.2 Switches

Three switches are used to choose a controller and initialize encoders. As shown below, four types of experiments can be done.

SW1	SW2	SW3	Controller
OFF	OFF	OFF	No Control
ON	OFF	OFF	Continuous LQ Controller (Sampling time: 1[ms])
ON	ON	OFF	Continuous LQ Controller (Sampling time: 80[ms])
OFF	ON	OFF	Discrete LQ Controller (Sampling time: 80[ms])
OFF	ON	ON	Deadbeat Controller (Sampling time: 80[ms])
ON	ON	ON	Initialize two encoders

Conclusions

During this project, two different approaches were used to design controllers: estimate a discrete controller from a continuous one or create a discrete controller directly from a discretized model of the plant. The second approach gave better results, since LQ-controllers created this way were especially designed for particular sampling times. The Deadbeat controller seemed to be a good idea, but it was not possible to implement it properly.

The controllers that were obtained worked well and were quite efficient, which shows that our modelling of the system was quite accurate. The energy controller swings the pendulum up rapidly and makes it arrive in the catch area with low speed; both continuous and discrete LQ-controllers stabilize the pendulum quickly. Moreover, the balancing controllers were really robust to disturbances (a step of 20 volts could be added to the continuous LQ-controller at high sampling frequencies without problem). For lower sampling frequencies, when the discrete LQ-controllers were used, the effects of the disturbance on control signals seemed to be about proportional to the sampling time.

However, what was not expected in this project was all the problems encountered with hardware and software. The biggest issue was handling Code Composer Studio. Most of the problems occurred because of this software and it took several weeks to do all that we wanted with it. That is why this was the most difficult part of the project. The other problems came from the sensor of the wheel that did not work properly, the LCD controller that was difficult to implement and other side tasks. Therefore, most of the time was not spent on automatic control but on solving hardware and software problems.

But the fact that we began from a point where almost nothing was implemented in the software gave us control on everything we did: the use of periodic functions, the acquisition of data from sensors, the implementation of controllers... This allowed us to avoid many unexpected results and deal with problems more easily when they occurred.

Of course, this project was limited in time and some points could have been improved. For example, the modelling could have been done by using the System Identification Toolbox of Matlab. This could have given us a more accurate model, especially for all disturbances and frictions that were neglected. However,

we were able to build controllers from our model that were satisfying. More important, an observer can be used to derive the state variables we cannot measure directly ($\dot{\theta}$ and ω_r), instead of calculating them by Euler approximation. This gives often better results. Finally, other kinds of controllers than state-feedback ones can be used: some built from a non-linear model (like exact feedback linearization) or other independent of the model (like a fuzzy controller).

This project was the opportunity for us to experience the topic of embedded systems. This requires a knowledge of many different fields such as modelling, signal processing, control theory and computer science. Consequently, it was an asset (quite unexpected) to have team members with different technical backgrounds. This course helped us to become familiar with project management and team work for a long period. Finally, we learnt a lot during these two months and had a pleasure doing this project.

Bibliography

- *Reglerteori - Flervariabla och olinjära metoder* - Torkel Glad and Lennart Ljung
- *Hybrid and Embedded Control Systems* - Lecture notes - Karl Henrik Johansson (KTH)
- *Automatique 2003* - Compendium - Collectif automatique (Suplec)
- *Nonlinear Control* - Lecture notes - Karl Henrik Johansson and Bo Wahlberg (KTH)

Appendix A

Tests

A.1 Measurement of mgl

With the pendulum horizontal, a scales was placed under the wheel to measure the weight of the pendulum. The radius where the scales was located was also measured. Using the measured values, the torque mgl can be calculated.

$$m = 0.27 \text{ kg}, \quad l = 0.13 \text{ m}, \quad g = 9.81 \text{ m.s}^{-2}$$
$$mgl = 0.344 \text{ N.m}$$

A.2 Measurement of J_p

The moment of inertia J_p can be calculated by using the pendulum equation (1.9) without frictions or running the motor, linearized around the lower equilibrium point.

$$\ddot{\theta} + \frac{mgl}{J_p}\theta = 0 \tag{A.1}$$

Solving this differential equation and isolating J_p gives

$$J_p = \frac{mgl}{\left(\frac{2\pi}{T}\right)^2} \tag{A.2}$$

Where T is the period time for small pendulum oscillations, measured to be 0.72s. We will find that

$$J_p = 0.0045 \text{ kg.m}^2$$

A.3 Measurement of Pendulum Friction

The friction is calculated from the time constant for the decrease in pendulum amplitude. This is done by measuring the time it takes for the pendulum amplitude to decrease from a known initial value to a predetermined end value.

$$f_p = 5 \cdot 10^{-5} \text{ kg.m}^2\text{s}^{-1}$$

In comparison with the torque from gravity, $T = 1.55 \text{ N.m}$, the torque due to friction will be very small for all reasonable angular velocities. Hence f_p will be neglected.

A.4 Measurement of J_w

The wheel has the shape of a cylinder, whose inertia can be derived as

$$J_{cylinder} = \frac{mR^2}{2} \quad (\text{A.3})$$

The inertia of the rotor in the DC-motor then needs to be added to get the total inertia.

$$J_w = J_{cylinder} + J_{rotor} \quad (\text{A.4})$$

and it resulted in

$$J_w = \frac{mR^2}{2} + J_{rotor} \quad (\text{A.5})$$

The measurements and data are

$$m = 0.065 \text{ kg}, \quad R = 0.025 \text{ m}, \quad J_{rotor} = 9.89 \cdot 10^{-7} \text{ kg.m}^2$$

which results in

$$J_w = 2.13 \cdot 10^{-5} \text{ kg.m}^2$$

A.5 Motor Driver A3952 Test 1

The voltage and the current was measured at different PWM outputs from the DSP, both when the wheel was running at constant speed and when it was held still. Following are the results:

The voltage is almost constant when the speed of the wheel is slowed down by hand. The current on the other hand increases dramatically. It is clearly shown that the controllable quantity is the voltage.

A linear function

$$v = k_O u \quad (\text{A.6})$$

Input	PWM duty cycle	Running wheel	Stopped Wheel
-10	100%	22 V	21 V
-5	74%	9 V, 65 mA	8.4 V, 720 mA
-2.5	62%	3.4 V, 58 mA	2.8 V, 230 mA

	54.7 Ω		104.9 Ω	
Command	Current [mA]	Voltage [V]	Current [mA]	Voltage [V]
10	400	21.64	211	21.81
5	202	11.05	107	11.19
2	81	4.36	45	4.46

Table A.1: Motor Driver Test 2

from output u to voltage v is to be derived. A mean value of all the points will be used. We will find that

$$k_O = -2.0V \quad (\text{A.7})$$

A more accurate nonlinear function is derived in A.8.

A.6 Motor Driver A3952 Test 2

A variable resistor is coupled to the driver circuit instead of the motor. To find out if the motor driver controls the voltage or the current, the voltage and current are measured for different resistances, and different control signals.

Two different resistances and three different PWM output signals is used. Results are presented in Table A.6.

The voltage is almost constant when the resistance is changed. The current is almost halved. The controllable quantity is apparently the voltage.

A.7 Wheel Friction Test

The steady state angular velocity of the wheel is notated for different constant outputs from the DSP. The result is shown in Figure A.1.

According to Equation 1.8, the dynamics of the wheel is described by

$$\dot{\omega} = -\left(\frac{f}{J_w} + \frac{k_E k_T}{J_w R_T}\right)\omega + \left(\frac{f}{J_w} + \frac{k_E k_T}{J_w R_T}\right)\dot{\theta} + \frac{k_T}{J_w R_T}v \quad (\text{A.8})$$

In steady-state, $\dot{\omega}$ and $\dot{\theta}$ will be zero and we will have

$$\left(\frac{f}{J_w} + \frac{k_E k_T}{J_w R_T}\right)\omega = \frac{k_T}{J_w R_T}v \quad (\text{A.9})$$

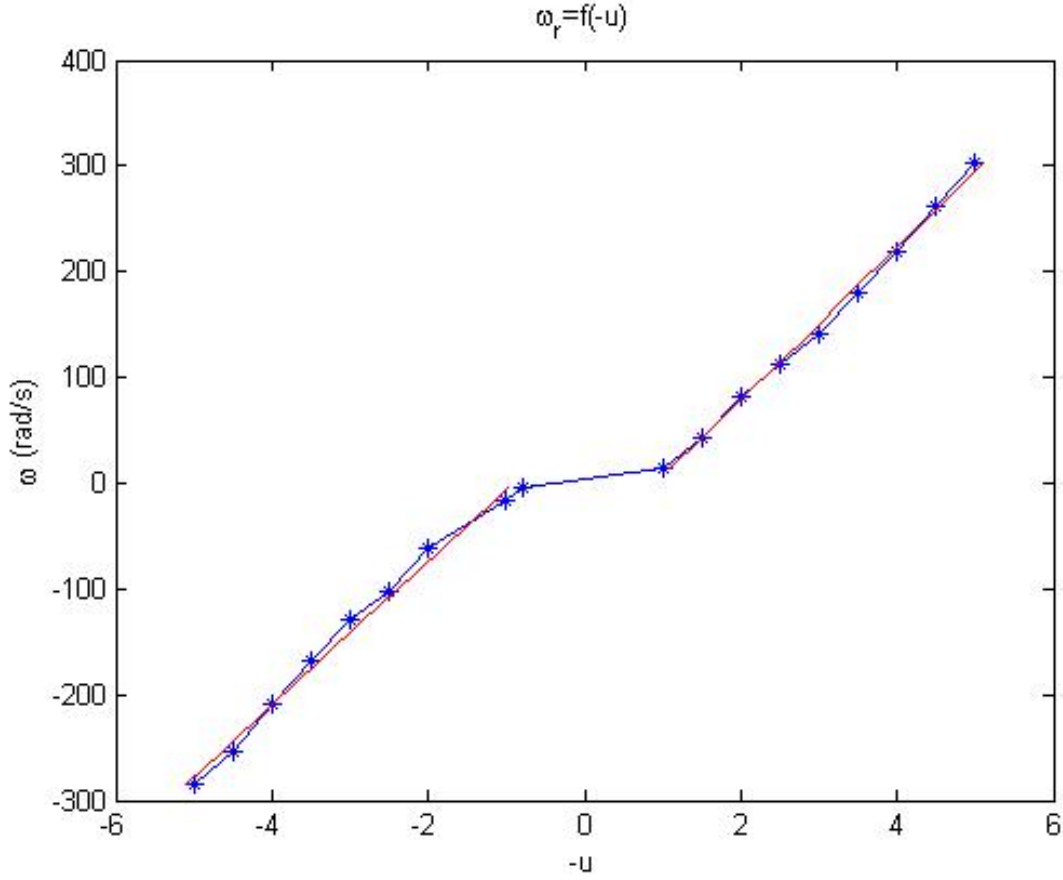


Figure A.1: The resulting angular velocity of the wheel for different outputs from the DSP (not voltages)

which, with equation (A.6), can be rewritten as

$$f = \frac{k_T}{R_T} \left(\frac{k_O u}{\omega} - k_E \right) \quad (\text{A.10})$$

First of all we notice that there is a dead zone between -1 and 1, probably because of dry friction. The derivative $\frac{du}{d\omega}$ for the line in the upper right corner will be used as the u/ω factor. The value is -0.0139 , and we get that

$$f = 9.06 \cdot 10^{-7} \text{ Nm.s.rad}^{-1} \quad (\text{A.11})$$

The friction f is always compared to the parameters

$$\frac{k_E k_T}{R_T} = 6.2 \cdot 10^{-5} \text{ Nm.s.rad}^{-1} \quad (\text{A.12})$$

which is almost hundred times higher. The conclusion is that f is negligible.

A.8 Relation Between Output "u" and Voltage to the Motor

In A.5, a linear relation between the PWM output signal and the voltage to the motor was derived. It has been shown though, that this relation is highly nonlinear. To get rid of this nonlinearity a function $u = f(v)$ needs to be derived.

The points in Figure A.2 shows results from measurements of the voltage for different PWM outputs.

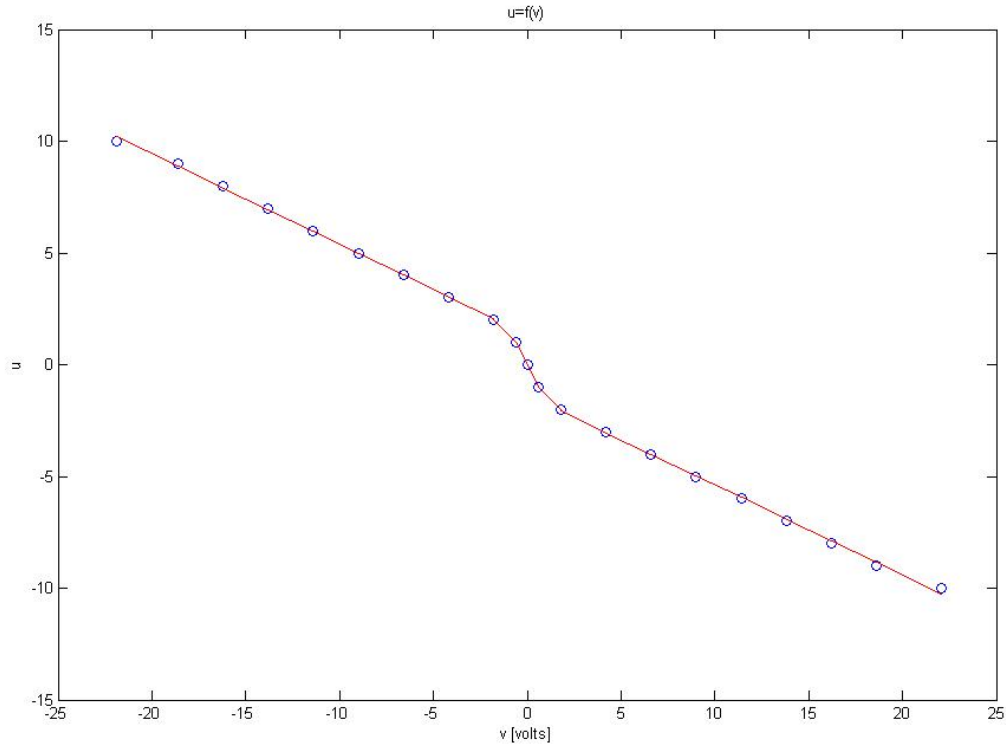


Figure A.2: The non linear function from voltage to the motor to the PWM output signal u .

A function is derived to fit to these measured points. It is divided into three different domains. The middle domain (around zero) is a third order polynomial fitted with the least square method. The two other parts are linear regressions.

The function is defined as

$$u = \begin{cases} 1.3317 - 0.4059v & v \leq -1.8 \\ -1.7361v + 0.1929v^3 & -1.8 < v < 1.8 \\ -1.3510 - 0.4035v & 1.8 \leq v \end{cases} \quad (\text{A.13})$$

Figure A.2 shows that this function fits the points well.

Appendix B

Filtering of Data

First order low-pass filters will be used to filter the angular velocities. Such a filter can be described in continuous time by

$$Y_f(s) = \frac{1}{Ts + 1} Y(s) \quad (\text{B.1})$$

where T is the time constant for the filter.

Using Euler backward estimation for the derivative according to

$$s = \frac{z - 1}{zh} \quad (\text{B.2})$$

where h is the sampling time. After some simple derivation it will be shown that

$$zY_f(z) = \frac{T}{h} \frac{Y_f(z)}{1 + \frac{T}{h}} + \frac{zY(z)}{1 + \frac{T}{h}} \quad (\text{B.3})$$

Transformed back to time domain and shifted one step back we get

$$y_f(k) = \frac{T}{h} \frac{y_f(k-1)}{1 + \frac{T}{h}} + \frac{y(k)}{1 + \frac{T}{h}} \quad (\text{B.4})$$

The time constants for the filters need to be chosen. Setting them too low will not suppress noise enough, too high will distort and flatten the signal. Experiments show that $T_\omega = 0.020$ and $T_{\dot{\theta}} = 0.004$ suppress noise in a good manner. The time constants of the corresponding systems are about 20 times larger, so the "real signals" should not be affected.

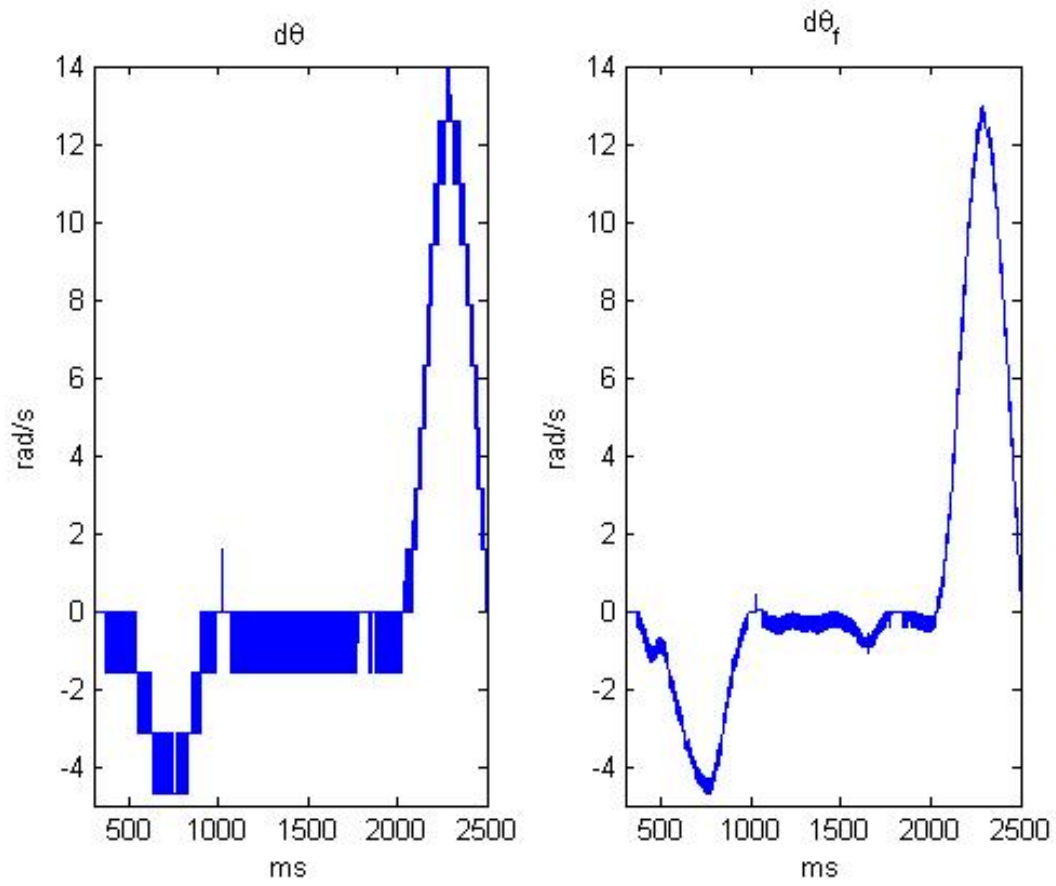


Figure B.1: The angular velocity of the pendulum before (left) and after (right) filtering.