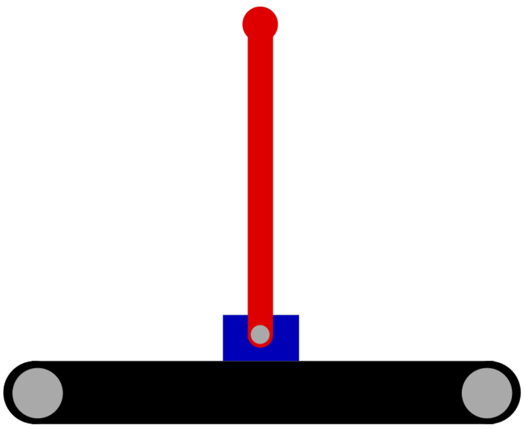
The Inverted Pendulum

Semesterproject in Control and

Simulation of autonomous systems



Project group 5

Alex Ellegaard - `aelle20` Anders Lind-Thomsen - `andli20`

Simon Christensen - `simch20` Peter Frydensberg - `pefry20`

Thomas Therkelsen - `ththe20` Victoria Jørgensen - `vijoe20`

Supervisor

Christian Schlette

SYDDANSK UNIVERSITET

BEng in Robot Systems

TEK MMMI

University of Southern Denmark

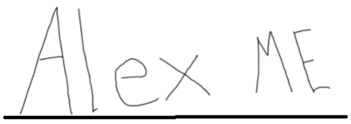
May 21st 2022

Abstract

Preface

Special thanks should be given to Associate Professors Aljaz Kramberger & Christoffer Sloth, for exemplary guidance and councelling regarding the control systems design in this project.

Alex Ellegaard



Simon Bork

Anders Lind-Thomsen

Figures/anders.underskrift.png

Thomas Therkelsen

Victoria Jørgensen

Peter Frydensberg

Contents

1 Introduction

The fourth industrial revolution, as some refer to it, is constantly evolving, as can be seen in the automation of many industrial and everyday processes. One of the many advantages of automating a process is that the mistakes that humans make during the manual process are eliminated. In light of these things, a wide range of robots are constructed for all kinds of purposes.

In this project, the robot is going to be based upon a PLC which will be controlling an inverted-pendulum-cart-system, which is inherently unstable. This is marginally resemblant to the classic kids' game of trying to keep a pencil upright on the palm, by adjusting for the movement of the pencil by moving your hand around. This is a case in which you can notice the robot eliminating human error. This is due to humans being unable to accurately measure and react to the movement of the pencil. - Whereas the robot can quantify the error precisely and adjust for it in a short amount of time.

Stabilizing an inverted pendulum on a cart, an inherently unstable system is a textbook control problem. This is one that many will face in their journey through the world of control systems. This version of the problem is taken one step further, as it's not just an inverted pendulum on a cart; It's an inverted pendulum on a cart, strapped to a conveyor belt, moved by a DC motor, and controlled by a PLC.

1.1 Problem

From the given project description, a problem is formulated from which a fully fledged control system can be developed.

- How is the dynamical model of the system as well as a control system mathematically modelled as well as simulated such that it can be implemented on the physical system?
- How is the PLC used to implement the control system model on the physical system, to be able to control it?
- How is the physical system controlled using the implemented control system, and how will it need to be tweaked to optimize performance?

1.2 Specifications

1.3 Report structure

2 Related Work

This chapter surveys and compares previous work in the field of controlling the classic control systems problem of an inverted pendulum on a cart.

2.1 Modelling of the system

The problem is commonly approached using one of two methods of modelling the system being either **Newton–Euler**-based modelling or **Euler–Lagrange**-based modelling. Given that the system is inherently unstable and it consisting of non-linear elements, a linearized model of the system is desirable. This is usually done using a first order **Taylor approximation**, which is either applied on the equations of motion of the system, or the State-Space model of the system. As well as a linearization of the system model, some approximations such as $\sin(\theta) \approx \theta$, $\cos(\theta) \approx \theta$ and $\dot{\theta}^2 \approx 0$ all for sufficiently small θ.

2.2 Controlling the system

2.2.1 Classical Control

There are a few common approaches to designing a control system for the inverted pendulum. The most simple way of controlling an inverted-pendulum-cart-system is by making just a single PID controller that corrects the pendulum angle.

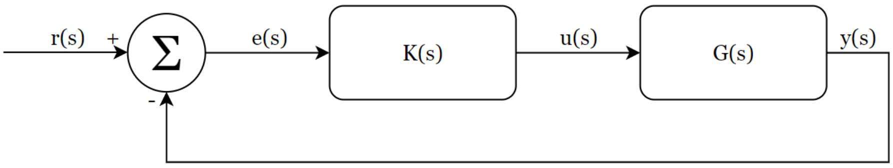


Figure 1: Example of PID Control structure

Another way to do it is to design a parallel control structure using two PID controllers with one correcting the pendulum angle and the other correcting the cart position.

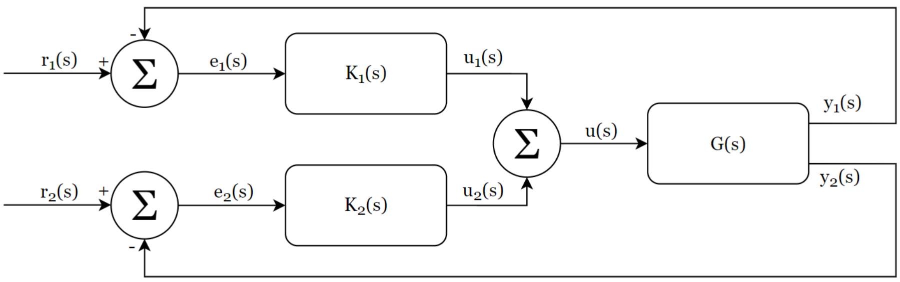


Figure 2: Example of Parallel PID Control structure

If a parallel structure is not desirable for ones system, a cascade control system can be implemented as seen below.

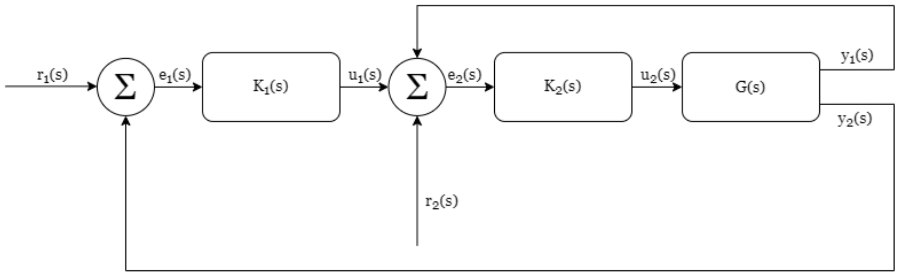


Figure 3: Example of Cascade PID Control structure

2.2.2 Modern Control

A modern control system can be designed to control the pendulum angle and cart position.

2.3 Our contribution

Our contribution to this problem will be based upon Euler–Lagrange modelling, which a few approaches will be made to determine the one with the best performance.

3 Mathematical Modelling

3.1 Euler-Lagrange Modelling

This project revoles around controlling a SIMO - system (single input, multiple outputs) - a cart with a pendulum attached to it. The goal of the control- system is to position the pendulum in an upright position, and keeping it there for a couple of seconds (MORE PRECISE). The cart can move on one axis(x), and the pendulum can move in two dimensions (x, y). The only input to the system is from the cart, which comes from a motor. A graph of the system and a box of parameters can be seen here:

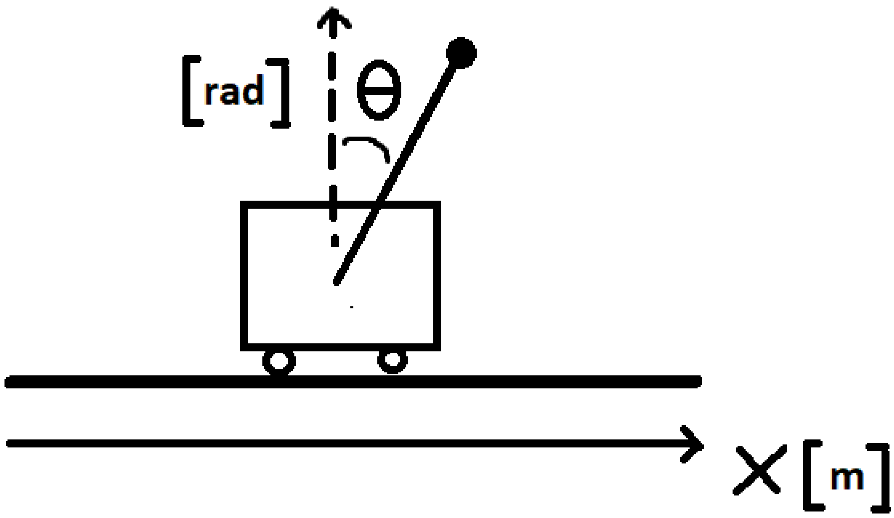


Figure 4: Model of the system

The parameters describing the system are

Name	Symbol	Value	Unit
Mass of cart	M	0.5	kg
Mass of pendulum	m	0.166	kgm
Damping between cart and track	b_c	5	N/(m/s)
Damping between pendulum and bearings	b_p	0.0012	Nm/(rad/s)
Gravitational pull	g	9.82	m/s^2
length of the pendulum arm	l	0.35	m
Position of cart	x_c	variable	m
Velocity of cart	$\dot{x}_c$	variabe	m/s
Acceleration of cart	$\ddot{x}_c$	variable	m/s^2
Angle of pendulum	θ	variable	rad
Angular velocity of pendulum	$\dot{\theta}$	variable	rad/s
Angular acceleration of pendulum	$\ddot{\theta}$	variable	rad/s^2
Input force on cart	u	variable	N

In order to control the system, a model of the system needs to be determined. This can be done by Euler-lagrange modelling or THE NEWTONIAN METHOD. Here, Euler-lagrange modelling is utilized. The equations of motions are derived from the following formulars:

$$\mathscr{L} = E_{kin} - E_{pot} \tag{1}$$

Where $\mathscr{L}$ is the lagrangian.

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathscr{L}}{\partial \dot{q}} - \frac{\partial \mathscr{L}}{\partial q} = Q \tag{2}$$

Where q is a vector of the generalized coordinates $q = [x_c, \theta]^T$, and Q is a vector of generalized forces, in this case friction and input force $Q = [-b_c \cdot \dot{x}_c + u, -b_p \cdot \dot{\theta}]^T$.

The kinetic energy of the system is given by:

$$E_{kin} = \frac{1}{2} \cdot (M+m) \cdot \dot{x}^2 - m \cdot \dot{x} \cdot l \cdot \dot{\theta} \cdot \cos(\theta) + \frac{1}{2} \cdot m \cdot l^2 \cdot \dot{\theta}^2 \tag{3}$$

This consists of the kinetic energy from the translational velocity along the x-axis of the pendulum and cart ($\frac{1}{2} \cdot (M+m) \cdot \dot{x}^2$), the kinetic energy from the pendulum's velocity along the y-axis $(-m \cdot \dot{x} \cdot l \cdot \dot{\theta} \cdot \cos(\theta))$ and lastly the kinetic energy coming from the angular velocity of the pendulum ($\frac{1}{2} \cdot m \cdot l^2 \dot{\theta}^2$) The potential energy of the system is given by the gravitational force on the pendulum:

$$E_{pot} = m \cdot g \cdot l \cdot \cos\theta \tag{4}$$

By using equation **??**, 2, 3 and 4 the equations of motion written as second order equations on vector-form become:

$$\begin{bmatrix} (M+m) \cdot \ddot{x} - m \cdot l \cdot \cos(\theta) \cdot \ddot{\theta} + m \cdot l \cdot \sin(\theta) \cdot \dot{\theta}^2 \\ m \cdot l^2 \cdot \ddot{\theta} - m \cdot l \cdot \cos(\theta) \cdot \ddot{x} - m \cdot g \cdot l \cdot \sin(\theta) \end{bmatrix} = \begin{bmatrix} u - b_c \cdot \dot{x} \\ -b_p \cdot \dot{\theta} \end{bmatrix} \tag{5}$$

These equations are non-linear because of the trigonometric functions and '$\dot{\theta}^2$'. They therefore have to be linearized, before modern control and classical control can be used on the system.

In order to linearize, we use the Jacobian matrix with linearization around an equilibrium point. The equilibrium point is a system state, around which, we want our system to move. In our case, this point corresponds to the upright pendulum-position, and an arbitrary cart-position. This method also brings the system directly on state space form which is used for modern control-strategies. Before linearizing the system, the general state space notation is described, since it play a big role in the linearization.

The state space form is a way of describing a linear time-invariant system. The general state space structure is as follows:

$$\dot{x} = A \cdot x + B \cdot u \tag{6}$$

$$y = C \cdot x + D \cdot u \tag{7}$$

Here, x is the state space vector, B is the input matrix, y describes the output of the measurements, C is the output matrix and D is the direct feedthrough matrix, which in many cases is set to a matrix of zeros. In our case, the state vector x is equal to $x = [x_c, \dot{x}_c, \theta, \dot{\theta}]^T$.

In order to linearize with the Jacobian matrix, an equilibrium point needs to established. This point is given by: $(\bar{u}, \bar{x}_c, \dot{\bar{x}}_c, \bar{\theta}, \dot{\bar{\theta}}) = (0, 0, 0, 0, 0)$, since the pendulum should be in upright position ($\bar{\theta}$) with ideally no angular velocity ($\dot{\bar{\theta}}$). In this pendulum-position, the cart should be standing still ($\dot{\bar{x}}_c, \bar{u}$) in an arbitrary position, here set to 0 for simplicty ($\bar{x}_c$). In order to use the Jacobian, the 2 non-linear second order equations of motion need to be reformulated into 4 first order equations. Matlab is here used to isolate the different state-variables from equation 5:

$$\begin{bmatrix} v \\ \dot{v} \\ \omega \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ -\frac{l \cdot m \cdot \sin(\theta)^2 + b_p \cdot \cos(\theta) - u + b_c \cdot v - g \cdot l \cdot m \cdot \cos(\theta) \cdot \sin(\theta)}{-l \cdot m \cdot \cos(\theta)^2 m + M} \\ \dot{\theta} \\ -\frac{b_p \cdot \omega \cdot m + b_p \cdot \omega \cdot m - u \cdot l \cdot m \cdot \cos(\theta) - g \cdot l \cdot m^2 \cdot \sin(\theta) + \omega^2 \cdot l^2 \cdot m^2 \cdot \cos(\theta) \cdot \sin(\theta) + b_c \cdot v \cdot l \cdot m \cdot \cos(\theta) - g \cdot l \cdot m \cdot M \cdot \sin(\theta)}{-l^2 \cdot m^2 \cdot \cos(\theta)^2 + M \cdot l \cdot m + m^2 \cdot l} \end{bmatrix} \tag{8}$$

These equations are now first order equations and are ready to be used in the Jacobian. In order to use the jacobian, some more notation is presented. The first element of the righthandvector of equation 8 is called 'eq1', the second element of the righthandvector of equation 8 is called 'eq2', the third element is called 'eq3' and the same procedure with the fourth element which is called 'eq4'. The formular is as follows:

$$A = \begin{bmatrix} \frac{\partial eq1}{\partial x} & \frac{\partial eq1}{\partial \dot{x}} & \frac{\partial eq1}{\partial \theta} & \frac{\partial eq1}{\partial \dot{\theta}} \\ \frac{\partial eq2}{\partial x} & \frac{\partial eq2}{\partial \dot{x}} & \frac{\partial eq2}{\partial \theta} & \frac{\partial eq2}{\partial \dot{\theta}} \\ \frac{\partial eq3}{\partial x} & \frac{\partial eq3}{\partial \dot{x}} & \frac{\partial eq3}{\partial \theta} & \frac{\partial eq3}{\partial \dot{\theta}} \\ \frac{\partial eq4}{\partial x} & \frac{\partial eq4}{\partial \dot{x}} & \frac{\partial eq4}{\partial \theta} & \frac{\partial eq4}{\partial \dot{\theta}} \end{bmatrix} \Bigg|_{(u,x,\dot{x},\theta,\dot{\theta}) = (\bar{u},\bar{x},\dot{\bar{x}},\bar{\theta},\dot{\bar{\theta}})} \tag{9}$$

$$B = \begin{bmatrix} \frac{\partial eq1}{\partial u} \\ \frac{\partial eq2}{\partial u} \\ \frac{\partial eq3}{\partial u} \\ \frac{\partial eq4}{\partial u} \end{bmatrix} \Bigg|_{(u,x,\dot{x},\theta,\dot{\theta}) = (\bar{u},\bar{x},\dot{\bar{x}},\bar{\theta},\dot{\bar{\theta}})} \tag{10}$$

This operation outputs these matrices:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{b_c}{M+m-l\cdot m} & \frac{g\cdot l\cdot m}{M+m-l\cdot m} & -\frac{b_p}{M+m-l\cdot m} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{b_c}{M+m-l\cdot m} & \frac{g\cdot l\cdot m^2+g\cdot l\cdot M\cdot m}{l\cdot m\cdot(M+m-l\cdot m)} & -\frac{b_p\cdot M+b_p\cdot m}{l\cdot m\cdot(M+m-l\cdot m)} \end{bmatrix} \tag{11}$$

$$B = \begin{bmatrix} 0 \\ \frac{1}{M+m-l\cdot m} \\ 0 \\ \frac{1}{M+m-l\cdot m} \end{bmatrix} \tag{12}$$

Lastly, we need the C-matrix in order to obtain the state space form. Here the angle of the pendulum and the position of the cart is being measured, and therefore we get:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{13}$$

These equations can now be used to describe the system via state space, and modern control can therefore be utilized. The final structure becomes:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{I\cdot b_c+b_c\cdot l\cdot m}{G} & \frac{g\cdot l^2\cdot m^2}{G} & -\frac{b_p\cdot l\cdot m}{G} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{b_c\cdot l\cdot m}{G} & \frac{g\cdot l\cdot m^2+g\cdot l\cdot M\cdot m}{G} & -\frac{b_p\cdot M+b_p\cdot m}{G} \end{bmatrix} \cdot \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+l\cdot m}{G} \\ 0 \\ \frac{l\cdot m}{G} \end{bmatrix} \cdot \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} \tag{14}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} \tag{15}$$

In order to do classical control, the transfer functions of the system need to be determined. This can easily be done with matlab. REFERENCE HER TABER!. The transfer functions become:

$$G_P(s) = \frac{\theta(s)}{u(s)} = \frac{1.606 \cdot s}{s^3 + 9.004 \cdot s^2 - 8.881 \cdot s - 78.85} \tag{16}$$

$$G_C(s) = \frac{x(s)}{u(s)} = \frac{1.793 \cdot s^2 + 0.06554 \cdot s - 15.77}{s^4 + 9.004 \cdot s^3 - 8.881 \cdot s^2 - 78.85 \cdot s} \tag{17}$$

The system has now been modelled with euler-lagrange-modelling and linearized with the jacobian matrix. The state-space matrices and the transfer functions have also been determined, so it is now possible to do classical and modern control - strategies.

4 Control System Design

4.1 PID Control

4.1.1 Structure

To control the system we have chosen to implement a PID-controller with feedback on the system. The desired goals when implementing a PID-controller is typically fast rise time, low overshoot and close to no steady state error. For this system, the following time-domain performance specifications have been set.

- M_p is overshoot. We have choosen an overshoot of less than 25%.
- t_r is rise time. We have choosen a rise time of 0.5 seconds.
- t_s is settling time. We have chosen a 1% settling time of 1 second.

In figure 5 below, the different symbolic values of the specification can be seen.

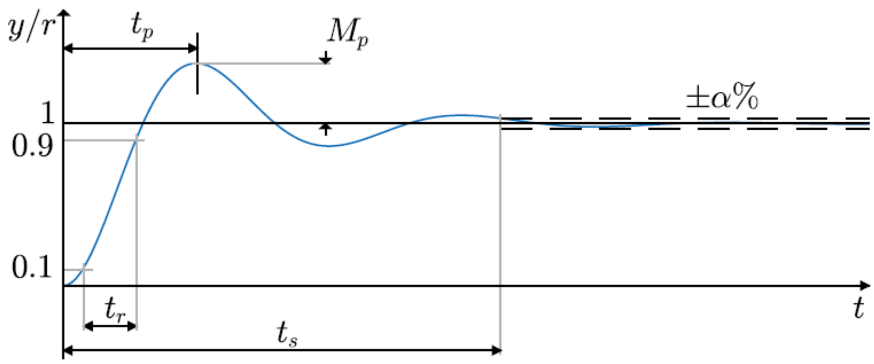


Figure 5: Time domain specification graph

Once the PID controller has been created and tuned, the performance can be compared to the previously defined performance specifications in time domain. To design a PID controller, one must first consider the transfer function for it.

PID controller on ideal form

$$K(s) = K_p \cdot \left(1 + \frac{1}{s \cdot T_i} + T_d\right) \tag{18}$$

PID controller on ideal form with low-pass filter in the differential term

$$K(s) = K_p \cdot \left(1 + \frac{1}{s \cdot T_i} + \frac{s \cdot T_d}{1 + \frac{s \cdot T_d}{N}}\right) \tag{19}$$

Where K_p is the proportional gain, T_i is the integral time constant, T_d is the differential time constant and N is the filter order.

4.1.2 Tuning

The transfer function for the open-loop response, given by the angle of the pendulum is shown in Equation 16 and is defined as G_P.

The loop gain and closed loop transfer function are derived for the system.

$$L_P(s) = G_P(s) \cdot K_P(s) \tag{20}$$

$$H_P(s) = \frac{L_P(s)}{1 + L_P(s)} \tag{21}$$

To illustrate how closed-loop poles move in the s-plane when the gain K_p is changed, a root locus plot of ones system can be drawn as seen below, and analyzed to pick the best gain.

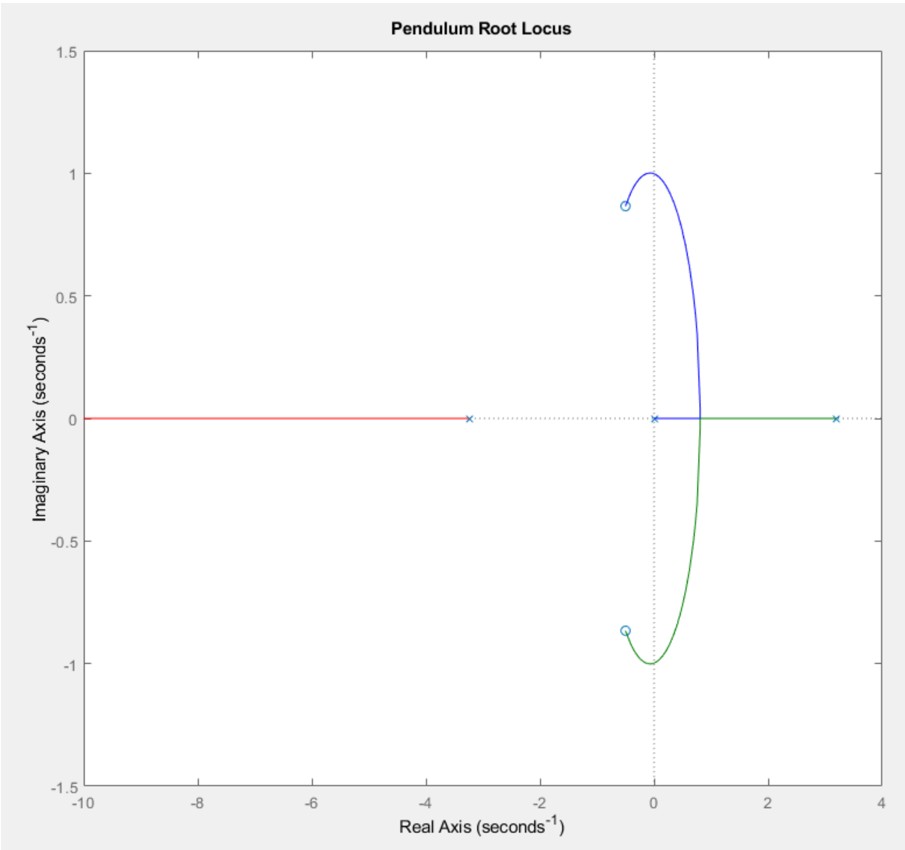


Figure 6: Pendulum Root Locus plot

By analyzing the root locus plot and doing some parameter finetuning afterwards, the following controller gains are picked. There is no filter on the derivative term.

$K_P(s)$ gains

$$\begin{aligned} K_p &= 617 \\ T_i &= 3.36 \cdot 10^3 \\ T_d &= 28.2 \end{aligned}$$

Resulting in the following responses for the pendulum.

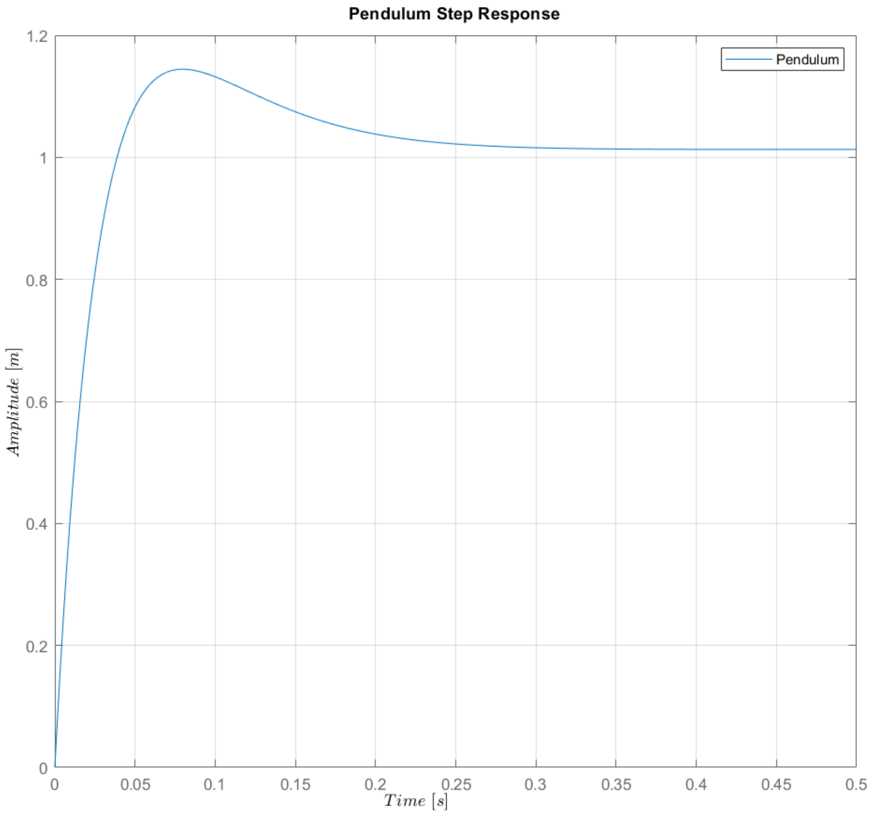


Figure 7: Pendulum step response

The impulse responses show that the pendulum system recovers from the δ impulse within $300\ [ms]$. The step response shows that the system converges on its' reference with a slight steady state error of amplitude$1.5 \cdot 10^{-2}$. Thus it can be concluded that this performance is acceptable.

There is just the problem of the cart; On the real life cart system, there are hardware limits of $\pm 0.86\ [m]$. Meaning, a new approach must be taken to solve this problem.

4.2 Parallel Control

4.2.1 Structure

While a single PID controller could do the job in a context with no bounds on how far the cart can move, which is not the case in this project given the hardware limits on the motor-conveyor-cart system. Based on the previously defined PID control system, which only adjusts the pendulum angle, a parallel PID control system can be designed. A parallel structure is chosen because it is desired to not only control the pendulum angle, but also the cart position at the same time. - And because of its ability to automatically adjust for control signals that differ. This structure is seen below in figure 8

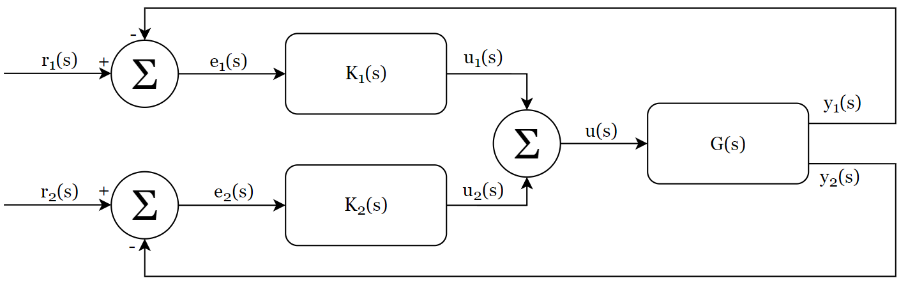


Figure 8: Example of Parallel PID Control structure

4.2.2 Tuning

When dealing with parallel structure, one method to design the system is to just design each controller independently of eachother. In our case, the pendulum controller is already designed, so the cart controller is missing. This is utilizable due to the previously mentioned property of the parallel structure which is that you simply add the control signals and let it handle the differences internally.

The transfer function for the cart was defined in 16. Then the loop gain and closed loop transfer function is derived for the cart

$$L_C = G_C(s) \cdot K_C(s) \tag{22}$$

$$H_C(s) = \frac{L_C(s)}{1 + L_C(s)} \tag{23}$$

The root locus plot looks as follows

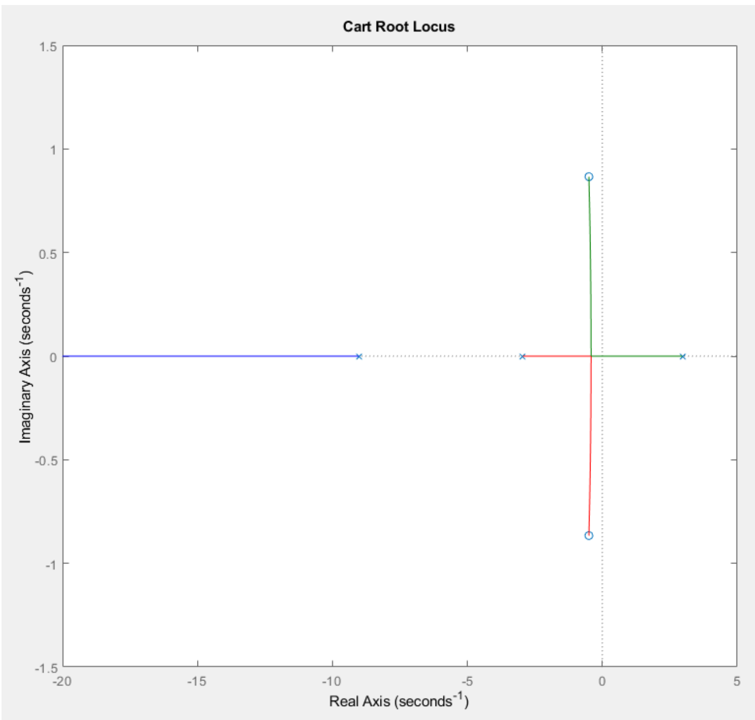


Figure 9: Cart Root Locus plot

The root locus is analyzed, and the controller gains are picked as

$K_C(s)$ gains

$$K_p = 2.16 \cdot 10^6$$
$$T_i = 5.14 \cdot 10^8$$
$$T_d = 2.26 \cdot 10^3$$

Which gives the following system responses.

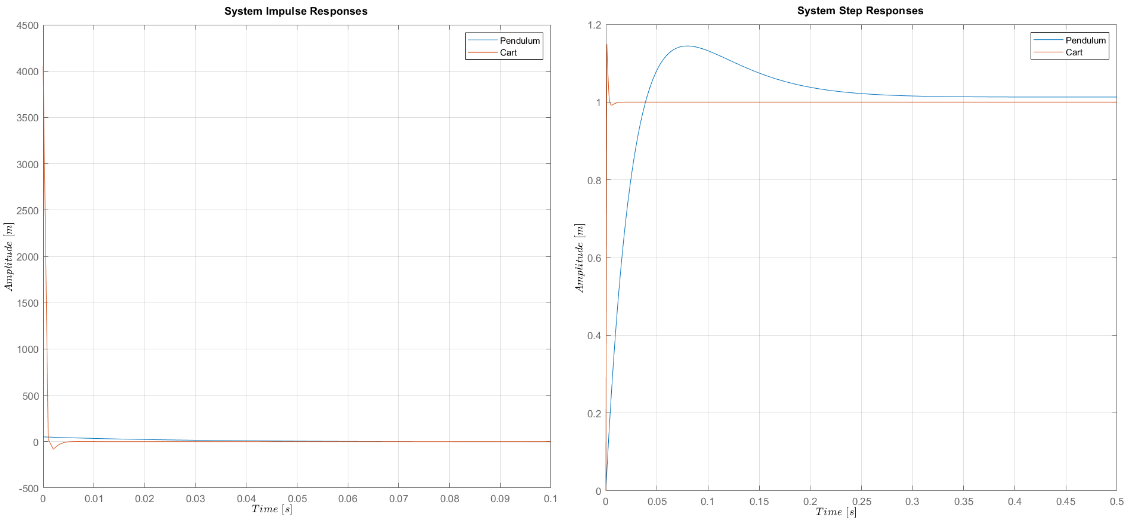


Figure 10: Parallel System Responses

While the responses are desirable, the gains of the cart controller are completely unrealistic, and cause the control signal to be many thousand newtons of force. This is unoptainable for the motor, resulting in the need for yet another new strategy to control the system.

4.3 Cascade Control

4.3.1 Structure

In a cascade control system, there are more than one controllers where the output of the master controller is subtracted added to the error of the slave controller, which then computes the final control signal as the input to the plant. In the figure below, a basic example of a cascade control system is shown.

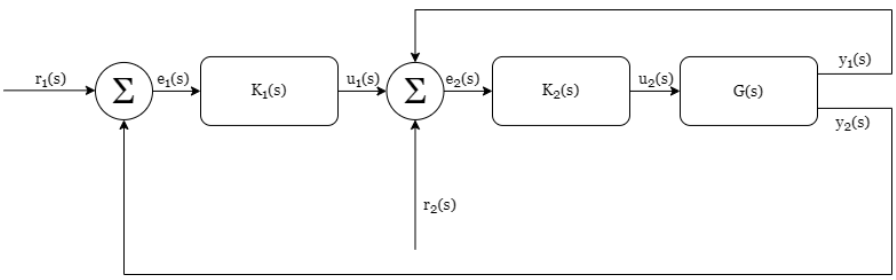


Figure 11: Example of Cascade PID Control structure

Here the master controller K_1 controls the outer loop and gives the input to controller K_2. This inner loop controller is the slave controller which computes a control signal rejecting disturbances before passing on to the plant G. It is essencial for the inner loop to respond much faster than the outer loop, a good rule of thumb is that it must be at least ten times faster than the outer loop, for the cascade control system to be well functioning.

4.3.2 Tuning

While the previous systems were tuned using root locus and some manual tweaking, this system is tuned in a slightly different manner; Using the built in PID Tuner from the PID Block in Simulink. The final gains for the two controllers are seen below.

$\mathrm{K_P}(s)$ gains

$$K_p = 695$$
$$T_i = 3.8 \cdot 10^3$$
$$T_d = 31.8$$

$\mathrm{K_C}(s)$ gains

$$K_p = 4.69 \cdot 10^{-3}$$
$$T_i = 3.36 \cdot 10^3$$
$$T_d = 28.2$$

Since this system is designed in Simulink, random white noise and big noise spikes as seen below in figure 12

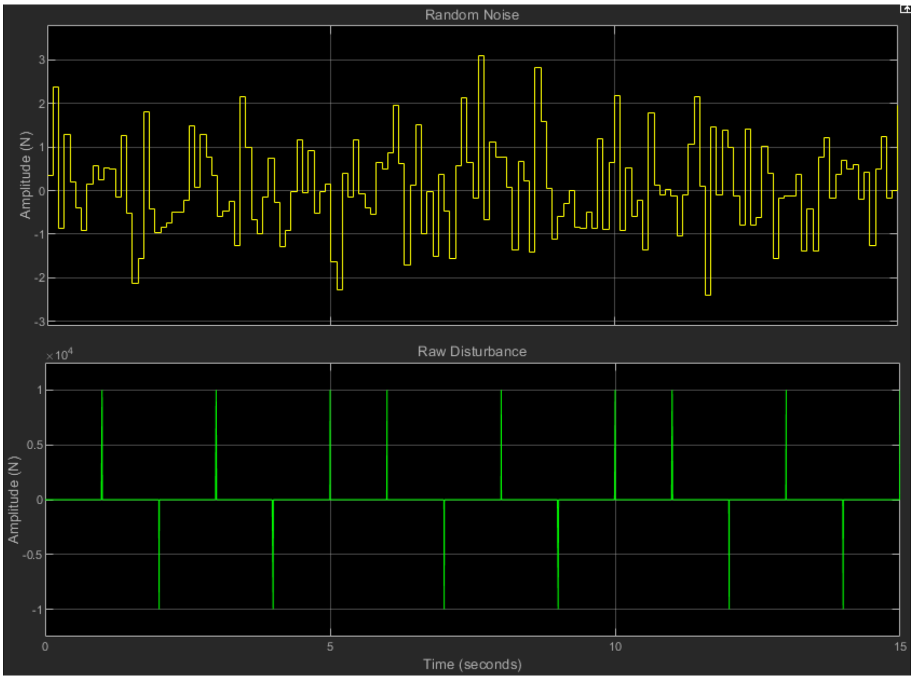


Figure 12: Disturbance Graphs

Using the previously defined gains in the cascade control system, the following performance is obtained.

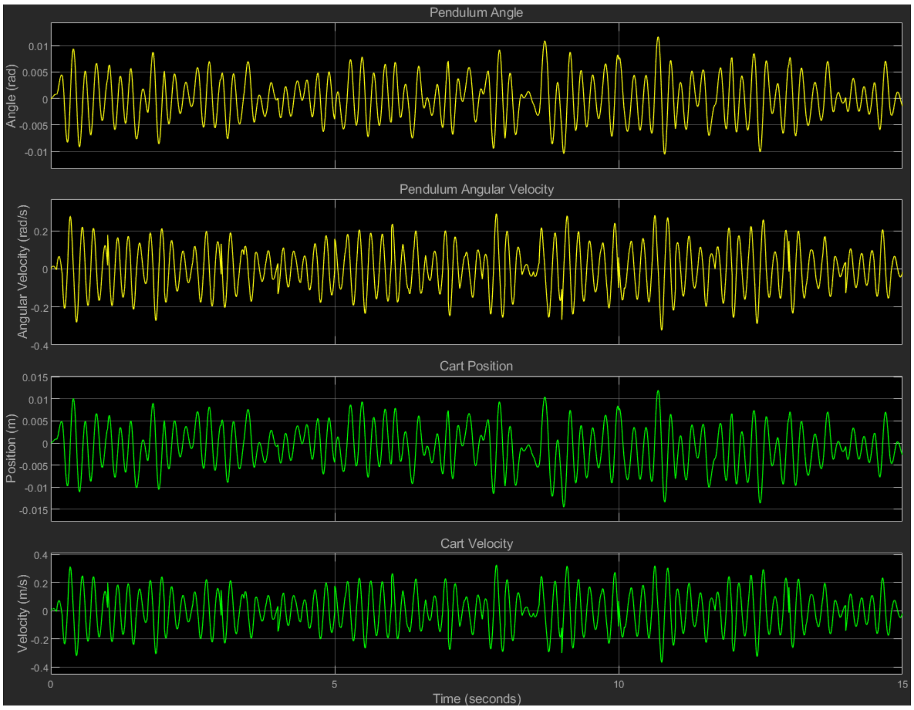


Figure 13: Cascade Control System Performance

5 Simulated Modelling and Control Systems Design

5.1 Rigidbody Model

To get an idea of how the system behaves under force control, a model of the system as well as a few control systems are designed in Simulink, using a combination of Simscape and Simulink components. A good deal of inspiration is taken from the CTMS website[1], which is changed quite a bit as their system is different than ours. In our model, the motor and belt are both omitted for simplification's sake.

5.1.1 Parameters

The parameters of the rigidbody system is given by

Name	Symbol	Value	Unit
Length of cart	L	0.203	m
Length of pendulum	l	0.350	m
Width of cart	W	0.0038	m
Width of pendulum	w	0.006	m
Mass of cart	M	0.5	kg
Mass of pendulum	m	0.0084	kg
Cart damping	b_c	5	$\mathrm{N}/(\mathrm{m}/\mathrm{s})$
Pendulum damping	b_p	0.0012	$\mathrm{N}/(\mathrm{m}/\mathrm{s})$

Using these parameters, a simplified model of the system is created using a few different Simscape components. A visualisation of this model is shown below.

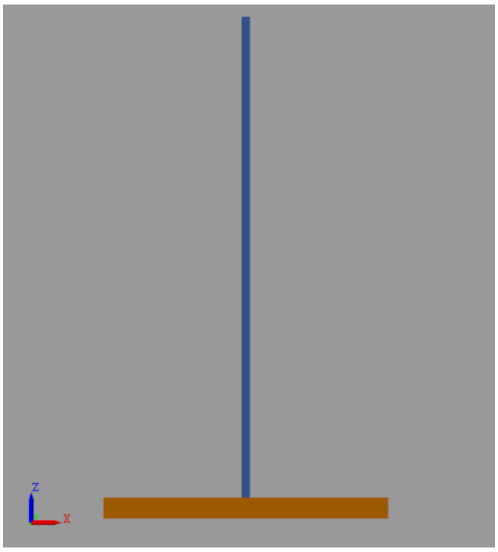


Figure 14: Rigidbody Model

5.2 System Model

This model is then hooked up to some converters to/from Simulink signals, to allow for control systems design to commence.

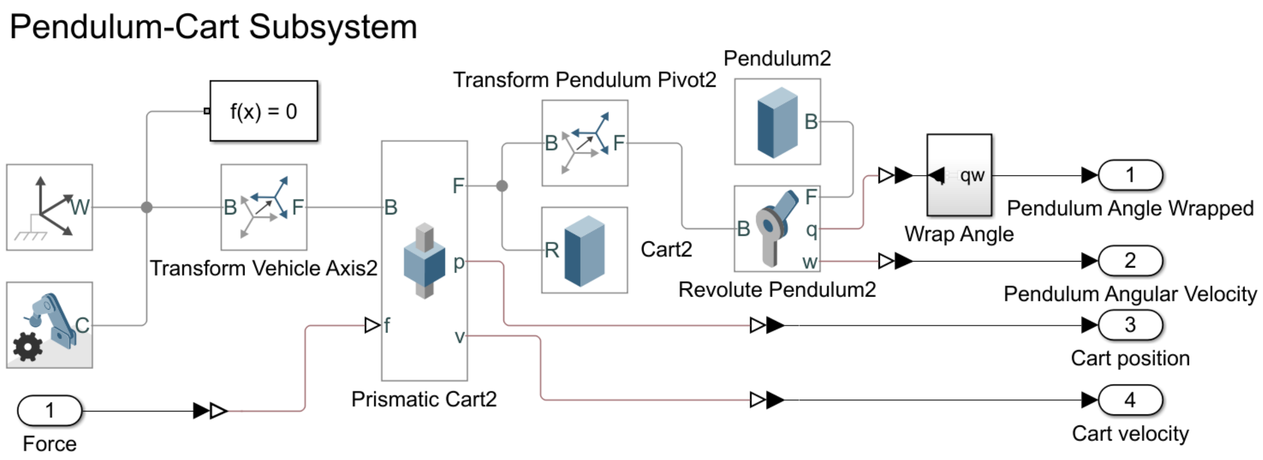


Figure 15: Pendulum-Cart Simscape Subsystem

Before doing any control systems design, it should be worked out whether the model is accurate. As seen in figure 16, the pendulum starts at 0 [rad] (pointing upright) and after around 15 [s] falls and converges to π [rad] (pointing straight down), which affects the cart making it move to one side.

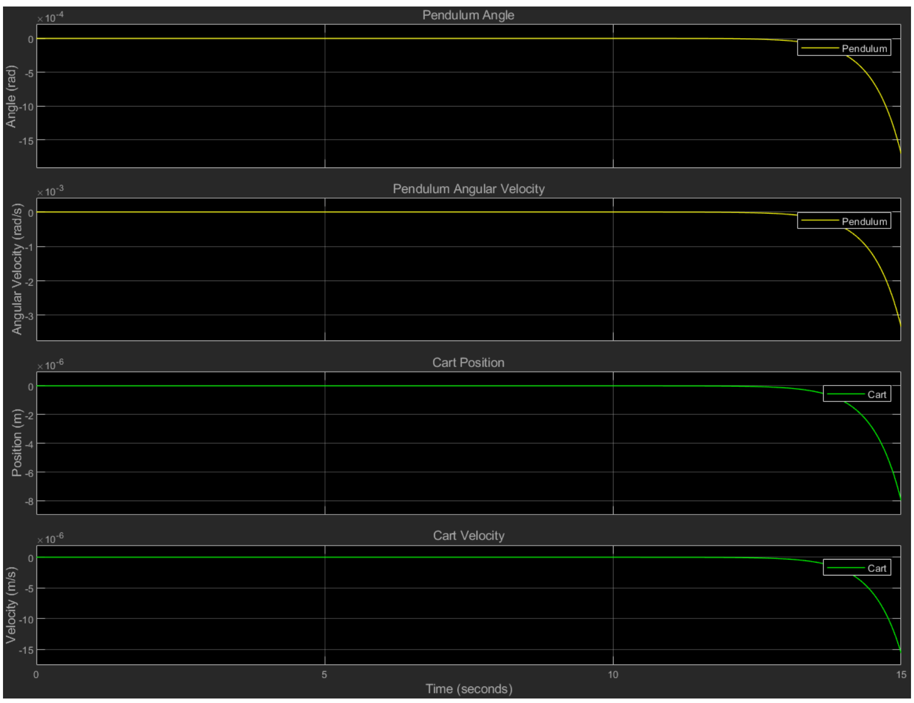


Figure 16: Rigidbody System Disturbanceless

5.3 Parallel PID Control

A parallel PID controller system is made using the previously defined Pendulum-Cart Subsystem as the Plant, and a few Simulink components, the main one being the built in PID block. Furthermore, a whole disturbance setup is made to comply with the project description.

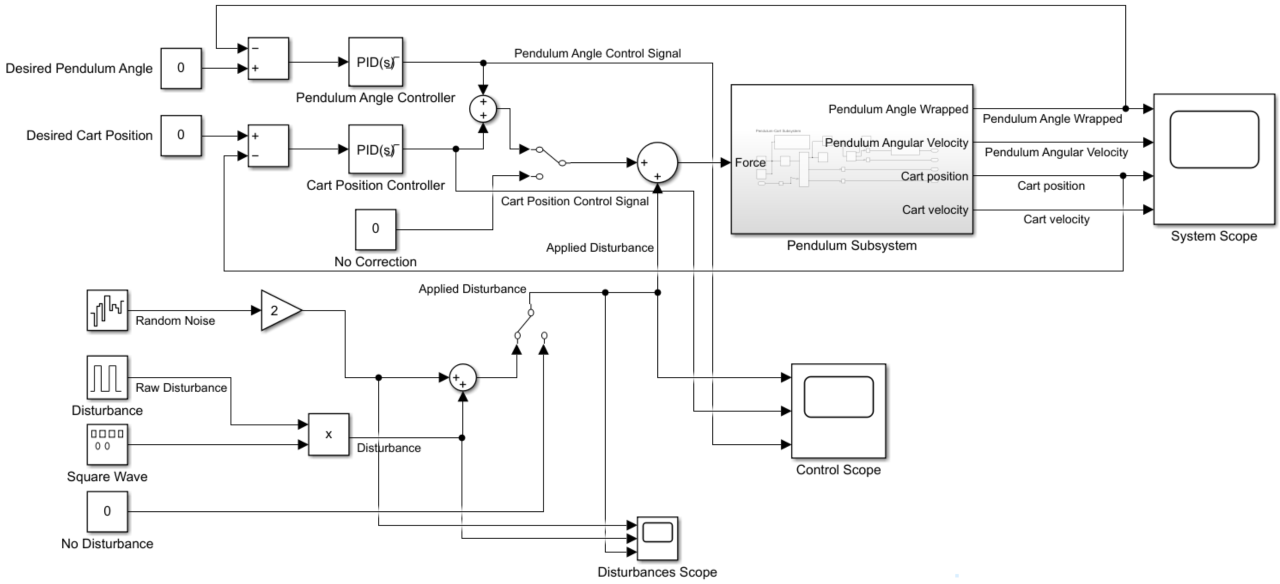


Figure 17: Simscape/Simulink Parallel Control System

A pulse generator is generating a disturbance of amplitude 10 [kN], which which is then multiplied by a square wave with an amplitude of 1 and then some random white noise is added. The former results in a pseudo-random disturbance with amplitude between -10 and 10 [kN] as seen in figure 18.

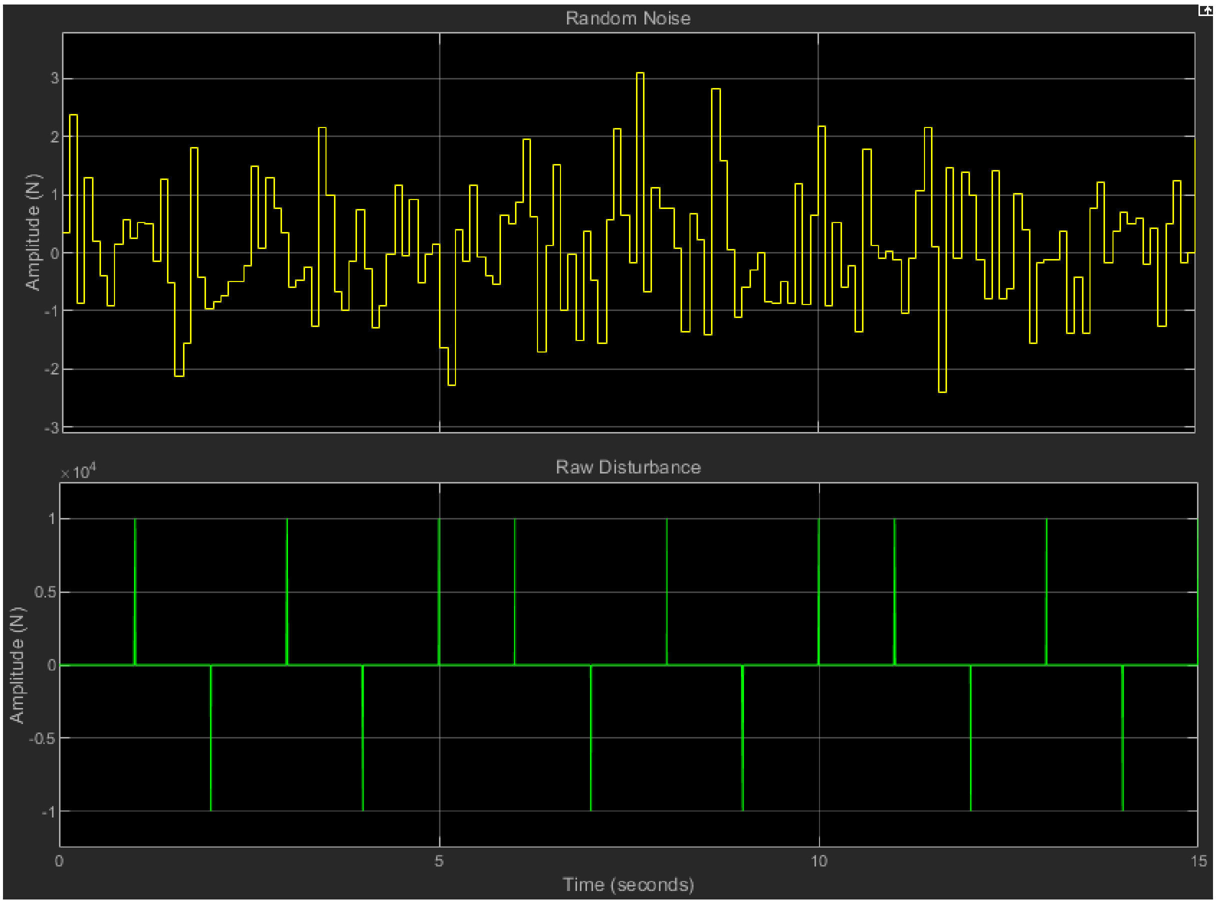


Figure 18: Disturbance Graphs

The PID-Controllers used in the parallel control structure are tuned independently of eachother, done by disconnecting one controller while the other is being tuned, and vice versa. The controllers have built in anti-windup of type back-calculation with a coefficient of 1, and saturation with the bounds -5 and 5. It may be worth noting, that the pendulum angle is wrapped to be between $-\pi$ and π [rad]. The simulated control system's performance can be seen below. Using the parallel control system to reject the previously defined disturbances, a maximum divergence from the reference (0 [rad]), of 0.1 [rad] is reached, as seen in the figure below. Furthermore, the cart has a mazimum divergence from its reference (0 [m]) of 0.115 [m].

5.4 Cascade PID Control

While the parallel system's performance looks fine, a cascade PID controller system tested as well. It is made using the previously defined Pendulum-Cart Subsystem as the Plant, and a few Simulink components.

The same disturbance is applied as on the parallel system, seen in figure 18. While in the parallel system you tune the controllers by disconnecting the other controller, in a cascade system you first tune the inner loop and then the outer loop. When tuning the controllers, a good rule of thumb to keep in mind to ensure good performance, is that the inner loop should be at least 10 times faster than the outer loop. Additionally, the controllers have the same anti-windup and saturation as in the parallel system.

Applying the previously mentioned disturbances to the defined control system, the performance is seen below in figure 21. As seen, a maximum divergence from the reference (0 [rad]), of magnitude $7.5 \cdot 10^{-2}$ [rad] is

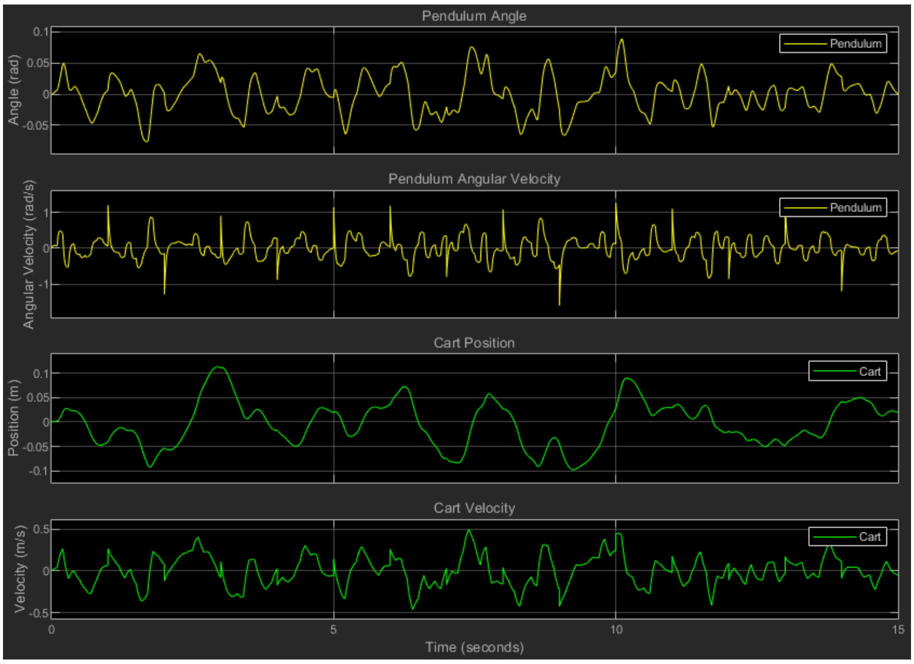


Figure 19: Parallel Control System Response

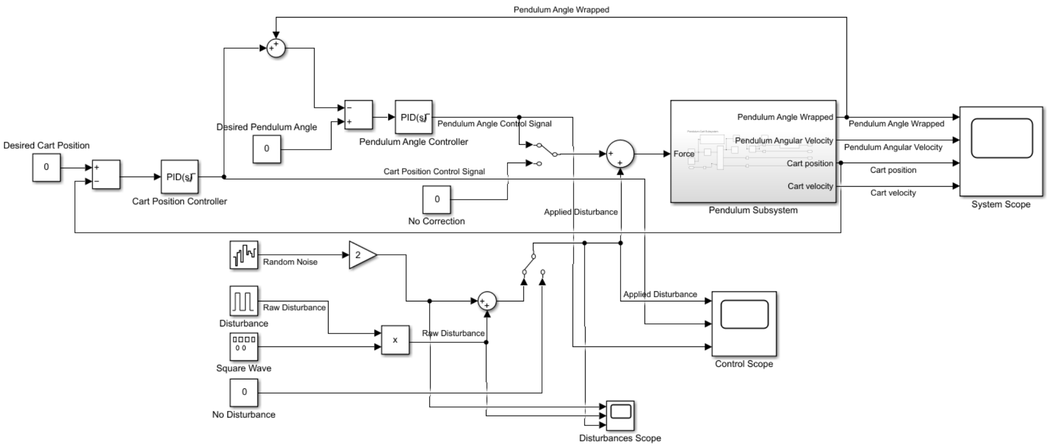


Figure 20: Simscape/Simulink Cascade Control System

reached, as seen in the figure below. This is quite similar to the parallel system, but the cart's performance is much better. This has a maximum error of 0.045 [m] which is twice as good as in the parallel system.

In conclusion, it is expected that cascade control will perform better than parallel control, on the physical system.

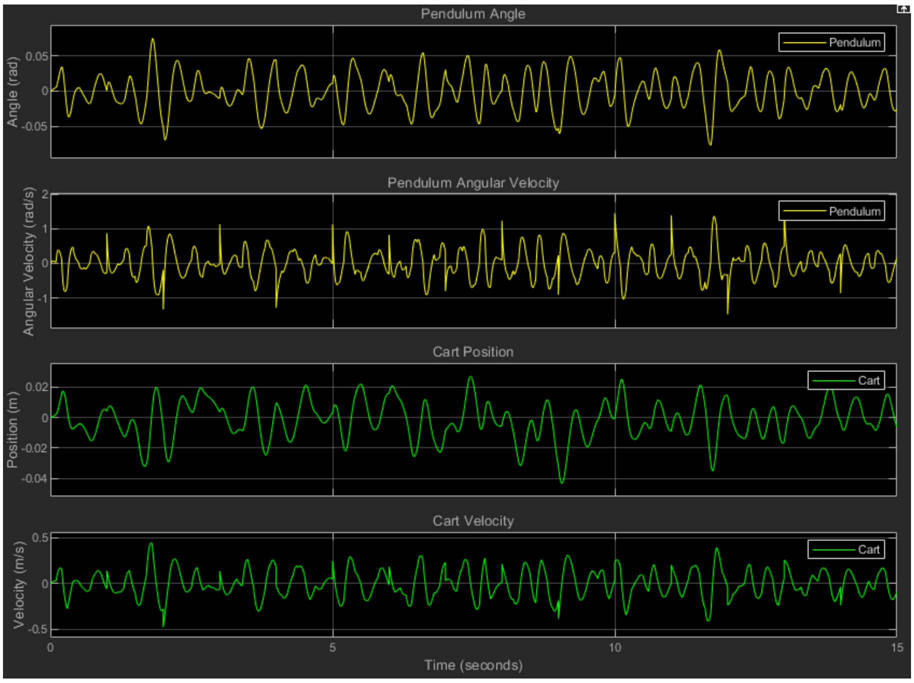


Figure 21: Cascade Control System Response

6 Discussion

7 Future Work

8 Conclusion

9 References

[1] Control Tutorials for MATLAB and Simulink. *Inverted Pendulum: Simscape Modeling*. URL: `https://bit.ly/3r3Glm7`.

10 Appendix

Appendix A