Controlling an Inertia Wheel Pendulum

Project Intelligent Control

Document name	Controlling an Inertia Wheel Pendulum	
Version	0.1	
Place & Date	Enschede	March 6, 2018
Created by:	Brian Armstrong	331934
	Edwin van Emmerik	407184

Contents

Foreword

Eddwin van Emmerik and Brian Armstrong would like to take this moment to thank Abeje Mersha for the opportunity provided to realize this project as well as the extra guidance and explanations required to get this far. Although full completion of the project has not been achieved upon writing this document the Intelligent-Control project has proven to be one of the most educational endeavors we have taken part in during our time at Saxion. It really opened our eyes to the mathematics and control theory beyond the curriculum given at Saxion and motivates us to continue developing our control theory skills.

Abbreviations

IWP Inertia Wheel Pendulum

Introduction

The inertia wheel pendulum system is an under-actuated single actuator inverse pendulum with 2 degrees of freedom. The way it works is by quickly accelerating an inertia wheel mounted on the end of the pendulum. The toque applied to the actuated inertia wheel generates torque on the inertia wheel as well as an opposite torque response on the pendulum. Therefore acceleration of the pendulum as well as inertia wheel can be achieved by actuating only the inertia wheel. The behavior of the system is drastically effected by the moment of inertia of the pendulum and inertia wheel.

Motivation: This system allows for a variety of interesting applications. The classic applications for this device are attitude control for satellites, telescopes and rockets but the inertia wheel is finding newer applications in robotics to balance uni/bi -cycle and -ped type robots. Spending time studying/realizing and controlling this system is something we believe could prove beneficial both for the further development of our knowledge in control theory as well as serve as a useful tool that we can use in the future.
Inertia wheel pendulums are well documented systems as they are the subject of many university theses. Although many of these systems are for high-tech purposes they can also be realized using budget components thus making this system feasible to realize within the context of this project.

Problem Analysis

To get a more in depth understanding of the system some introduction to system and the control problem will be given in this chapter.

2.1 Inertia Wheel Pendulum

The inertia wheel pendulum is a combination of two systems; inverted pendulum and a inertia wheel that is actuated at the end of the pendulum. The actuator needs to control the inertia wheel to apply a reacting torque to the inverted pendulum, hence the actuator can control some of the states of the pendulum. The system is shown in figure 2.1.

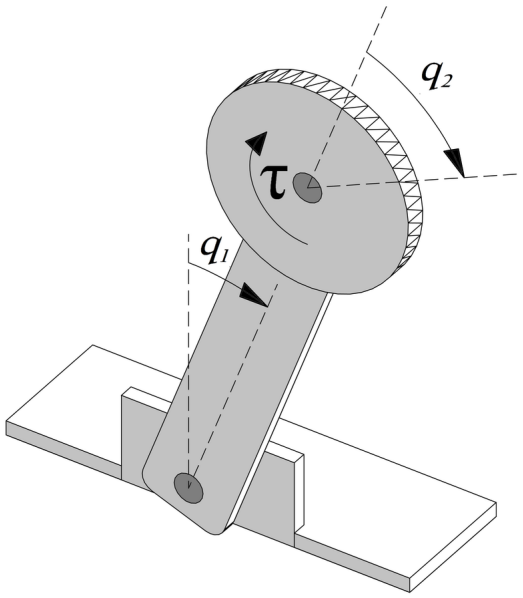

Figure 2.1: Inertia Wheel Pendulum

2.2 Control problem

The system has to be linearized due to the non-linearity of the inverted pendulum sub-system to use any of the linear control strategies. the inertia wheel pendulum has two degrees of freedom and only one actuator and thus it is an under-actuated system. Also, the system loses controllability if the angle of the pendulum and the y-axis is too large.

Problem Approach

In this chapter the approach of the problem will be given. Hence, this will be the main guidance through the rest of this document.

3.1 Linearization

The system its center of mass is not located on the revolute axis of the inverted pendulum. Thus, as stated in the paragraph 2.2 the system is only controllable within a certain angle parallel to the y-axis. This gives the opportunity to use linearization around an equilibrium point using the Taylor Series Expansion instead of full state feedback linearization using Lie algebra.

To determine what order of the Tayler Series Expansion is needed to linearize around the equilibrium point, the Taylor Series Expansion up to the third order is shown in figure 3.1. Where the blue line (y = sin(x)) needs to be linearized. By removing the sin() function this results in the linear orange line (y = x), where the linear range to the blue line is approximately $[-0.5, 0.5]rad$. By taking the third order Taylor expansion gives the purple line, where the linear range to the blue line is approximately $[-1.2, 1.2]rad$.

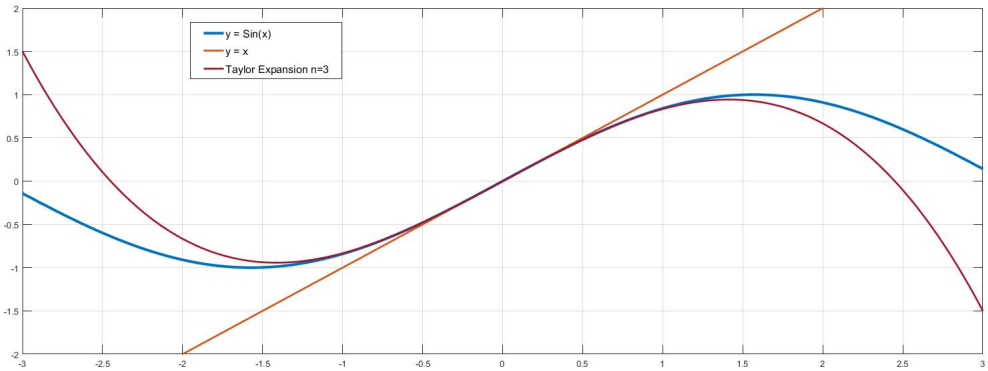

Figure 3.1: Taylor Expansion

Using this knowledge, the system will be controllable with a linear controller within the range of $[-0.5, 0.5]rad$ without actually linearizing the system.

3.2 Control strategies

The above research gave more insight on what kind of controller strategies should be implemented. Because the system can be controlled by a linear controller as stated above, a pid-controller can be designed to controll the system to its setpoint within the linear range.

The next step is to design an energy shaping swingup controller that controls the system to the linear-controllable range and then swap to the linear controller, this is possible because the non-linearity of the system does not influence the energy shaping swingup controller. Using this gain scheduling strategy the system will have adaptive control based on where the system behaves linear or non-linear and is more likely to reach the goal if both the controllers are tuned properly.

Modelling an inertia wheel pendulum

In this chapter the modeling of the system will be shown and explained in four steps; the schematic system desription, the system characterization based on Lagrangian mechanics and Euler-Lagrange equations, constructing the non-linear state space representation and finally the simulation of the uncontrolled system.

4.1 Schematic system description

For now lets assume that the pendulum is rigid and that the pendulum and wheel axis have no friction. The system schematics and the parameter table are shown below.

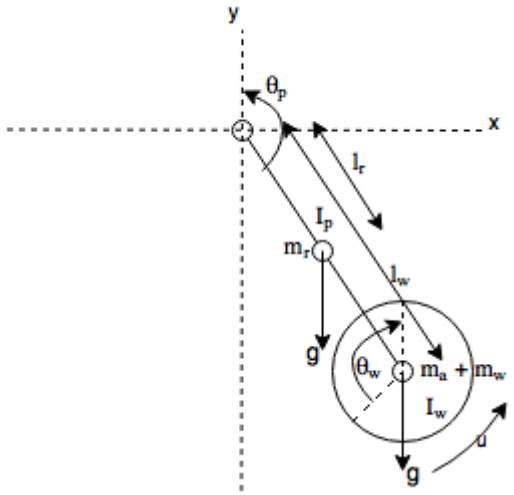

Figure 4.1: Inertia Wheel Pendulum schematics

m_r	Mass of the rod
m_w	Mass of the wheel
m_a	Mass of the actuator
l_r	Distance to the center of mass of the rod
l_w	Distance to the center of mass of the wheel
I_p	Moment of inertia of the pendulum
I_w	Moment of inertia of the wheel and actuator
u	Controller action [torque]
g	Gravity

Table 4.1: Schematics parameters

Now that the schematic description of the system is clear. In the next section the system will be mathematically characterized.

4.2 System characterization

Using the system schematics the Lagrangian equation of energy can be found using Lagrangian mechanics and the Lagrangian equation is shown in 4.1.

$$L = KE - PE \tag{4.1}$$

Using this equation the total kinetic energy of the system KE is a sum of the kinetic energy in the pendulum [KE_p] and the kinetic energy of the wheel [KE_w], where the kinetic energy of the wheel is the sum of the kinetic energy of the wheel and the actuator.

$$KE = KE_p + KE_w \tag{4.2}$$

Where the kinetic energy of the pendulum is shown in equation 4.3.

$$KE_p = \frac{1}{2} m_r l_p^2 \dot{\theta}_p^2 + \frac{1}{2} I_r \dot{\theta}_p^2 \tag{4.3}$$

And the kinetic energy of the wheel is shown in equation 4.4.

$$KE_w = \frac{1}{2}(m_a + m_w) l_w^2 \dot{\theta}_w + \frac{1}{2} I_w (\dot{\theta}_w + \dot{\theta}_p)^2 \tag{4.4}$$

Hence the total kinetic energy of the system:

$$KE = \frac{1}{2}\left(m_r l_p^2 \dot{\theta}_p^2 + I_r \dot{\theta}_p^2 + (m_a + m_w) l_w^2 \dot{\theta}_w + I_w (\dot{\theta}_w + \dot{\theta}_p)^2 \right) \tag{4.5}$$

Now that the total kinetic energy of the system has been found the same can be done for the total potential energy of the system.

$$PE = PE_p + PE_w \tag{4.6}$$

Where the potential energy in the pendulum is equal to:

$$PE_p = m_r l_p g\big(1 + cos(\theta_p)\big) \tag{4.7}$$

And the potential energy in the wheel is equal to:

$$PE_w = (m_a + m_w) l_w g\big(1 + cos(\theta_p)\big) \tag{4.8}$$

Hence the total potential energy of the system can be found.

$$PE = \big(m_r l_p + (m_a + m_w) l_w\big) g\big(1 + cos(\theta_p)\big) \tag{4.9}$$

Now the kinetic energy equation (equation 4.5) and the potential energy equation (equation 4.9 have been found these can be substituted into equation 4.1, hence lagrangian equation is shown below.

$$\begin{aligned} L = \frac{1}{2}\Big(& m_r l_p^2 \dot{\theta}_p^2 + I_r \dot{\theta}_p^2 + (m_a + m_w) l_w^2 \dot{\theta}_w + I_w (\dot{\theta}_w + \dot{\theta}_p)^2 \Big) \\ & - \Big(m_r l_p + (m_a + m_w) l_w \Big) g\big(1 + cos(\theta_p)\big) \end{aligned} \tag{4.10}$$

Before continuing some dissipative energy has to be introduced to the system as friction on the axis of the pendulum [DE_p] and the wheel[DE_w].

$$DE_p = \frac{1}{2} b_p \dot{\theta}_p^2 \tag{4.11}$$

$$DE_w = \frac{1}{2} b_w \dot{\theta}_w^2 \tag{4.12}$$

This results in the dissipative equation 4.13.

$$DE = DE_p + DE_w = \frac{1}{2}\left(b_p \dot{\theta}_p^2 + b_w \dot{\theta}_w^2 \right) \tag{4.13}$$

Now that the total energy of the system has been characterized using Lagrangian mechanics, the Euler-Lagrange equations can be constructed for the pendulum (equation 4.14) and the wheel (equation 4.15).

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_p} \right) - \frac{\partial L}{\partial \theta_p} + \frac{\partial DE}{\dot{\theta}_p} = 0 \tag{4.14}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_w} \right) - \frac{\partial L}{\partial \theta_w} + \frac{\partial DE}{\dot{\theta}_w} = u \tag{4.15}$$

Equations 4.10 and 4.13 can now be substituted into equation 4.14 and this results in 4.16. After some re-orginizing the Euler-Lagrange equation of motion for the pendulum(equation 4.17) is found.

$$\begin{aligned} &\frac{d}{dt}\Big(m_r l_p^2 \dot{\theta}_p + I_r \dot{\theta}_p + \big(m_a + m_w\big) l_w^2 \dot{\theta}_p + I_w \big(\dot{\theta}_w + \dot{\theta}_p\big) \Big) \\ &\quad - \Big(m_r l_p + \big(m_a + m_w\big) l_w \Big) g \sin(\theta_p) + b_p \dot{\theta}_p = 0 \end{aligned} \tag{4.16}$$

$$\Big(m_r l_p^2 + I_r + \big(m_a + m_w\big) l_w^2 + I_w \Big) \ddot{\theta}_p + I_w \ddot{\theta}_w - \Big(m_r l_p + \big(m_a + m_w\big) l_w \Big) g \sin(\theta_p) + b_p \dot{\theta}_p = 0 \tag{4.17}$$

Then again equations 4.10 and 4.13 can now be substituted in equation 4.15 to find the Euler-Lagrange equation of motion for the wheel (equation 4.19).

$$\frac{d}{dt}\Big(I_w \big(\dot{\theta}_p + \dot{\theta}_w\big) \Big) + b_w \dot{\theta}_w = u \tag{4.18}$$

$$I_w \big(\ddot{\theta}_p + \ddot{\theta}_w\big) + b_w \dot{\theta}_w = u \tag{4.19}$$

4.3 Non-linear state space representation

Using these two Euler-Lagrange equations of motion (4.17 and 4.19), the state variables $\ddot{\theta}_p$ (equation 4.20) and $\ddot{\theta}_w$ (equation 4.21) can be found using algebra.

$$\ddot{\theta}_p = \frac{\Big(m_r l_p + \big(m_a + m_w\big) l_w\Big) g \sin(\theta_p) + b_p \dot{\theta}_p - I_w \ddot{\theta}_w}{m_r l_p^2 + I_r + \big(m_a + m_w\big) l_w^2 + I_w} \tag{4.20}$$

$$\ddot{\theta}_w = \frac{u - b_w \dot{\theta}_w - I_w \ddot{\theta}_p}{I_w} \tag{4.21}$$

To find the pure state variable $\ddot{\theta}_p$ (equation 4.24), where the other state variable $\ddot{\theta}_w$ is not used within the same equation, equation 4.21 can be substituted in equation 4.20.

$$\ddot{\theta}_p = \frac{\Big(m_r l_p + \big(m_a + m_w\big) l_w\Big) g \sin(\theta_p) + b_p \dot{\theta}_p - I_w \frac{u - b_w \dot{\theta}_w - I_w \ddot{\theta}_p}{I_w}}{m_r l_p^2 + I_r + \big(m_a + m_w\big) l_w^2 + I_w} \tag{4.22}$$

$$\ddot{\theta}_p = \frac{\Big(m_r l_p + \big(m_a + m_w\big) l_w\Big) g \sin(\theta_p) + b_p \dot{\theta}_p + b_w \dot{\theta}_w + I_w \ddot{\theta}_p - u}{m_r l_p^2 + I_r + \big(m_a + m_w\big) l_w^2 + I_w} \tag{4.23}$$

$$\ddot{\theta}_p = \frac{\Big(m_r l_p + \big(m_a + m_w\big) l_w\Big) g \sin(\theta_p) + b_p \dot{\theta}_p + b_w \dot{\theta}_w}{m_r l_p^2 + I_r + \big(m_a + m_w\big) l_w^2} - \frac{1}{m_r l_p^2 + I_r + \big(m_a + m_w\big) l_w^2} u \tag{4.24}$$

Hence to find the pure state variable $\ddot{\theta}_w$ (equation 4.27), where $\ddot{\theta}_p$ is not within the same equation, equation 4.20 can be substituted into equation 4.21.

$$\ddot{\theta}_w = \frac{u - b_w \dot{\theta} - I_w \frac{\Big(m_r l_p + \big(m_a + m_w\big) l_w\Big) g \sin(\theta_p) + b_p \dot{\theta}_p - I_w \ddot{\theta}_w}{m_r l_p^2 + I_r + \big(m_a + m_w\big) l_w^2 + I_w}}{I_w} \tag{4.25}$$

$$\ddot{\theta}_w = \frac{u - b_w \dot{\theta}_w}{I_w} - \frac{\Big(m_r l_p + \big(m_a + m_w\big) l_w\Big) g \sin(\theta_p) + b_p \dot{\theta}_p - I_w \ddot{\theta}_w}{m_r l_p^2 + I_r + \big(m_a + m_w\big) l_w^2 + I_w} \tag{4.26}$$

$$\ddot{\theta}_w = -\frac{\Big(m_r l_p + \big(m_a + m_w\big) l_w\Big) g \sin(\theta_p) + b_p \dot{\theta}_p}{m_r l_p^2 + I_r + \big(m_a + m_w\big) l_w^2 + 2I_w} - \frac{\frac{b_w \dot{\theta}_w \Big(m_r l_p^2 + I_r + \big(m_a + m_w\big) l_w^2 + I_w\Big)}{I_w}}{m_r l_p^2 + I_r + \big(m_a + m_w\big) l_w^2 + 2I_w} + \frac{\frac{m_r l_p^2 + I_r + \big(m_a + m_w\big) l_w^2 + I_w}{I_w}}{m_r l_p^2 + I_r + \big(m_a + m_w\big) l_w^2 + 2I_w} u \tag{4.27}$$

Now that the system state variables $\ddot{\theta}_p$ and $\ddot{\theta}_w$ are found, the system its states can be determined as:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \theta_p \\ \dot{\theta}_p \\ \theta_w \\ \dot{\theta}_w \end{bmatrix} \tag{4.28}$$

Because the system is non-linear the following system model representation can be used for the found equations (4.24 and 4.27).

$$\dot{x} = f(x) + g(x)u \tag{4.29}$$

Hence, the state variables need to be derived over time one more time:

$$\dot{x} = \begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = \begin{bmatrix} \dot{\theta}_p \\ \ddot{\theta}_p \\ \dot{\theta}_w \\ \ddot{\theta}_w \end{bmatrix} \tag{4.30}$$

Then the equations 4.24 and 4.27 can be divided into two parts. One that contains information about the f(x) system behaviour and one contains information about the g(x) input behaviour. This results in equation 4.31 for f(x) and equation 4.32 for g(x).

$$f(x) = \begin{bmatrix} x_2 \\ \dfrac{\left(m_r l_p + \left(m_a + m_w\right) l_w\right) g \sin(\theta_p) + b_p \dot{\theta}_p + b_w \dot{\theta}_w}{m_r l_p^2 + I_r + \left(m_a + m_w\right) l_w^2} \\ x_4 \\ -\dfrac{\left(m_r l_p + \left(m_a + m_w\right) l_w\right) g \sin(\theta_p) + b_p \dot{\theta}_p - \frac{b_w \left(m_r l_p^2 + I_r + \left(m_a + m_w\right) l_w^2 + I_w\right)}{I_w}}{m_r l_p^2 + I_r + \left(m_a + m_w\right) l_w^2 + 2I_w} \end{bmatrix} \tag{4.31}$$

$$g(x) = \begin{bmatrix} 0 \\ \dfrac{1}{m_r l_p^2 + I_r + \left(m_a + m_w\right) l_w^2} \\ 0 \\ \dfrac{\frac{m_r l_p^2 + I_r + \left(m_a + m_w\right) l_w^2 + I_w}{I_w}}{m_r l_p^2 + I_r + \left(m_a + m_w\right) l_w^2 + 2I_w} \end{bmatrix} \tag{4.32}$$

4.4 Simulation of the uncontrolled system

Now that the system model is found, a simulation can be done. But for this the system parameters need to be determined. The mass parameters are weight, the length parameters are measured to the estimated center of mass, the inertia parameters are calculated and the friction parameters are determined using the friction of the. These values for the system parameters can be found in table 4.2. The simulation is shown in figure 4.2.

m_r	0.5
m_w	0.34
m_a	0.070
l_r	-0.08
l_w	0.2
I_p	0.020650
I_w	0.001224
b_p	0.01
b_w	0.1
u	0
g	9.81

Table 4.2: Filled system paramters

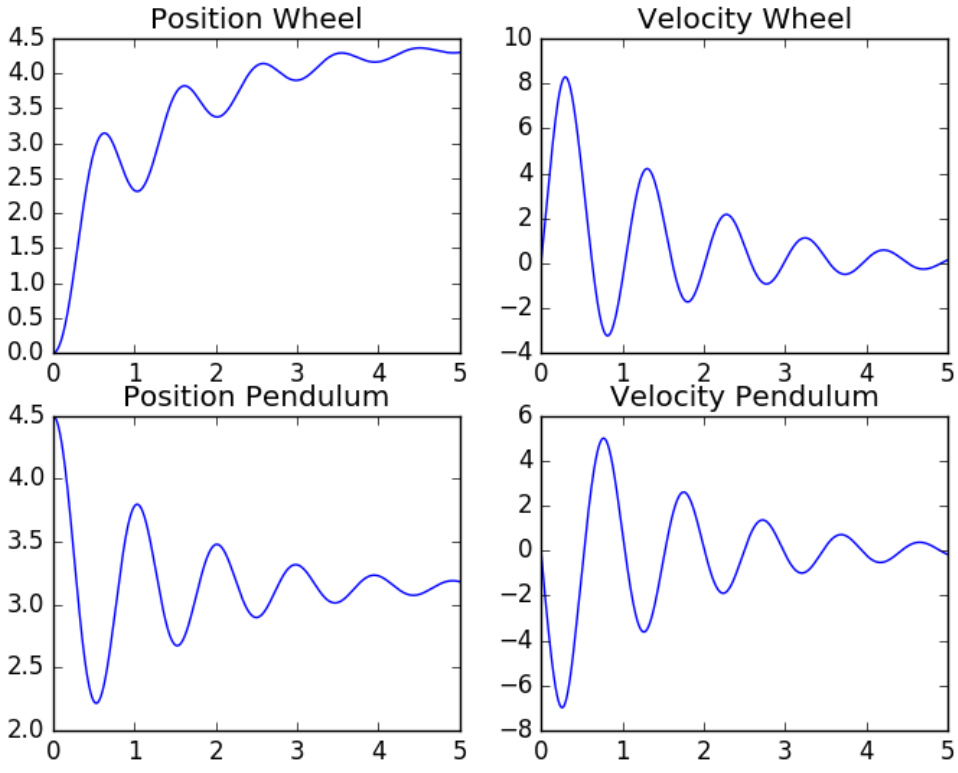

Figure 4.2: Uncontrolled system; $\theta_p = 0.4[rad]$,$\dot{\theta}_p = 0.0[rad/s]$

However it is still unknown if this is an accurate representation of the real system. That is why the same start state will be applied on the real system and the data from the encoder in the pendulum axes is saved and plotted (figure 4.3).

From the simulated model (figure 4.2-Position Pendulum) and the real system (figure 4.3) the frequency at which the system oscilates can be determined. For the simulated model this is approximately 0.9 Hz and for the real system this is also approximately 0.9 Hz. The next thing that can be analyzed is the damping. The simulated model shows that the system oscilates around 5 times before it reaches its natural equilibrium, while the real system shows that the system oscilates around 4 times before it reaches its natural equilibrium.

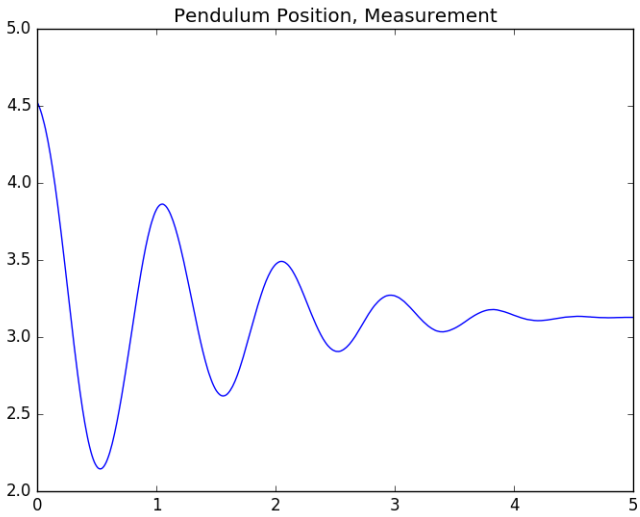

Figure 4.3: Measured pendulum position on real system

Now using these analysis there can be concluded that the determined system parameters are an accurate representation of the system. If this results in making the real system uncontrollable the system parameters can be slightly tuned intuitively.

Controlling an inertia wheel pendulum

In this chapter the controller choices, tuning methodology and architecture is motivated.

5.1 Adaptive Control

The adaptive controller chosen for this system is a gain scheduler. The reason this controller is chosen is because the behavior of the controller near it's equilibrium point $\theta_p = 0$ is drastically different than near the equilibrium point $\theta_p = \pi$. Thus 2 separate controllers must be realized and toggled accordingly to make the system behave as intended. When the pendulum is at $\theta_p = \pi$ the desired controller behavior is swing-up action, for maintaining the position at $\theta_p = 0$ PID controllers will be realized.
The chosen range for PID control is $-0.3 \leq \theta_p \leq 0.3$ all other values for pendulum position should result in swingup behavior.

The gain scheduler is realized using the following logic statement.

if $(sin(0.3)^2 < sin(\theta_p)^2$ or $cos(\theta_p) <= 0)$
is true -> Swingup Control
if not true -> PID control

$sin(0.3)^2 < sin(\theta_p)^2$ accounts for the region in which the PID controller is intended to function.
$cos(\theta_p) <= 0$ makes sure the swingup controller remains on when θ_p is near π as well.

5.2 Multiple Input Single output PID control

Pendulum Position PID controller design The pendulum position PID controller is only active in the region of $-0.3 \leq \theta_p \leq 0.3$. Using the knowledge from figure 2.1 the controller can achieve desired behavior when tuned for a linearized system model where $sin(\theta_p) := \theta_p$.

The linearized state space model has the following matrices:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{m_c g}{I_t - I_w} & \frac{-b_p}{I_t - I_w} & 0 & \frac{-B_p}{I_t - I_w} \\ 0 & 0 & 0 & 1 \\ \frac{-m_c g}{I_t + I_w} & \frac{-b_p}{I_t + I_w} & 0 & \frac{-b_w I_t (I_t + I_w)}{I_w} \end{bmatrix} \quad (5.1)$$

$$B = \begin{bmatrix} 0 \\ \frac{-1}{I_t - I_w} \\ 0 \\ \frac{I_t / I_w}{I_t + I_w} \end{bmatrix} \quad (5.2)$$

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.3)$$

$$D = 0 \quad (5.4)$$

$$I_t = m_r l_p^2 + (m_a + m_w) l_w^2 + I_r + I_w \quad (5.5)$$

To tune the system Nyquist and bode plots of the above state space model are constructed. The close loop instability of the system is quite obvious as the -1 + 0j point is widely encircled by the path in the Nyquist plot.

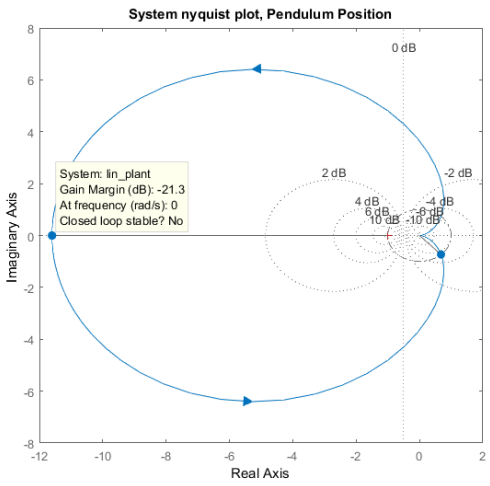

Figure 5.1: Nyquist plot of the linearized plant pendulum position

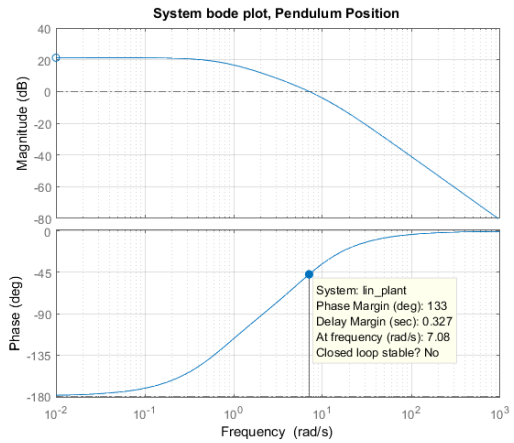

Figure 5.2: Bode plot of the linearized plant pendulum position

Designing a PID controller proved to be problematic using the Linear System Analysis and design tools normally used. The Control System Designer in Matlab was unable to provide any suitable PID controller parameters without producing unachievable high controller action. And the impulse response plots would often exceed the region in which the system linearizion was shown to be accurate.
Which highlights the next problem, the actuator on the Inertia wheel takes -5 to 5 volt reference. But the controller action would often exceed these limits. Attempts were made to model the actuator

saturation into Matlab to tune the PID controller, but these attempts were not met with success. Ultimately the chosen method for tuning the PID controller was to do it based on intuition. The simulation model written in python does contain actuator saturation and provided accurate plots, thus acceptable controlled system behavior can be achieved by trial and error by adjusting the parameters as seen fit.

The starting state of the system was set:

$$x = \begin{bmatrix} \theta_p \\ \dot{\theta}_p \\ \theta_w \\ \dot{\theta}_w \end{bmatrix} = \begin{bmatrix} 0.3 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.6}$$

First K_p is scaled up to the point that the controller can overcome the force of gravity to the point that the system comes close to reaching $\theta_p = 0$.
Then K_i is incremented until overshoot is present, the overshoot is increased as much as possible without making the pendulum fall on the opposing side.
Now K_d is introduced, it serves for 2 purposes.
1. The derivative action is increased to decrease the overshoot, but the overshoot must remain present as it is beneficial for decelerating the wheel near the equilibrium point, if the derivative action is too large the pendulum never reaches $\theta_p = 0$ and falls backwards.
2. The derivative action provides preventive controller action if the pendulum starts falling. So it must remain large to help prevent the pendulum from falling after the pendulum has reached the equilibrium point.

No matter what is used for the control parameters the pendulum will fall if enough time passes. This is solved by creating another controller and combining the controller actions.

Maintaining System Controllability As can be seen in 5.3 if the system manages to reach the equilibrium point $\theta_p = 0$ the pendulum will inevitably fall if enough time passes.
The cause for this controllers inability to maintain the equilibrium point is that the wheel velocity is not controlled. As the wheel speeds up it reaches it's max RPM. Near the max rpm the wheel cannot be accelerated in both directions anymore. In other words if the inertia wheel nears the max RPM the system approaches an uncontrollable system configuration and falls.

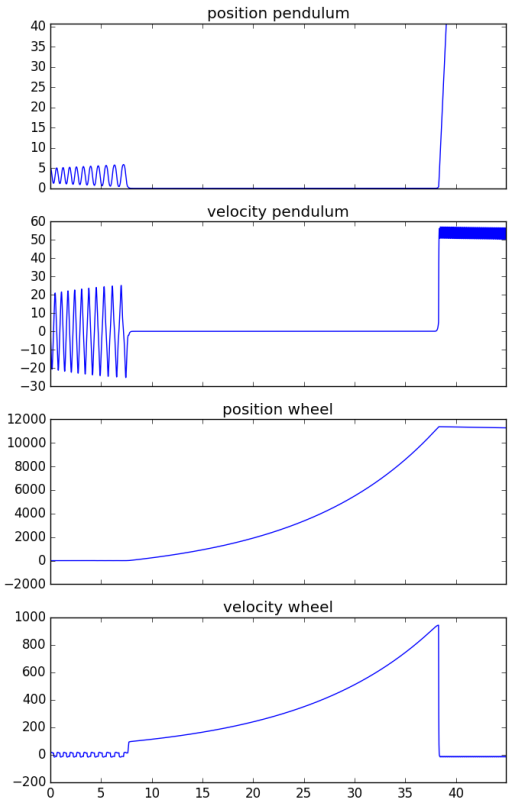

Figure 5.3: PID controlled pendulum, wheel uncontrolled

To solve this problem another PID controller is added, this second PID controller regulates the speed of the wheel back down to zero preventing the wheel from reaching high velocities, to tune this controller integrator action is added until the wheel velocity decelerates to 0 rad/s instead of accelerating till it falls. The output of the second PID controller is U_w. The pendulum position controller action is given the name U_p.
Since the main task of the controllers is to regulate pendulum position the wheel velocity controller must contribute a proportionally smaller controller action than U_p. The implementation of these controllers can be seen in 5.4. To provide an easy method to control the PID controller actions without significantly effecting the controller outputs U_w can be scaled with a weighting factor before the controller actions are combined.

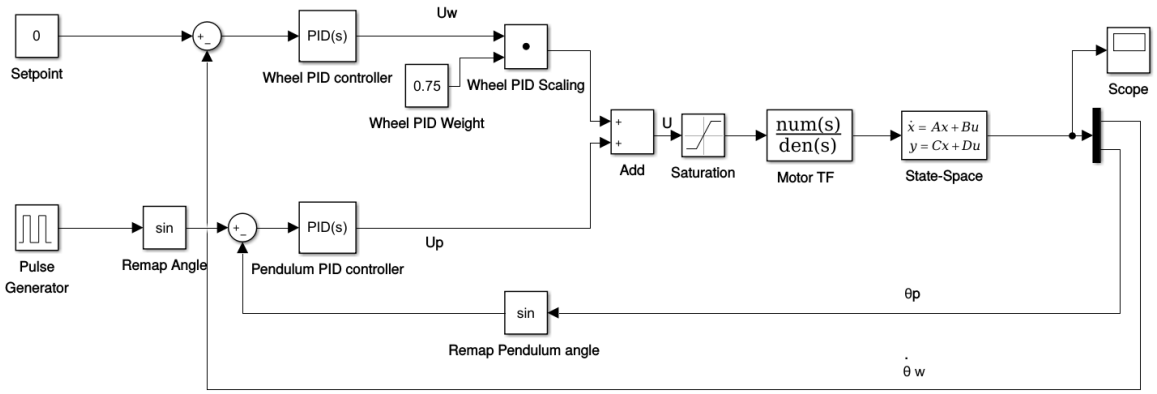

Figure 5.4: Simulink model of MISO controller, Parallel PID controllers with combined output

5.3 Energy Shaping Controller

The inverse pendulum is repulsed by the equilibrium point at $\theta_p = 0$ and attracted to $\theta_p = \pi$. Because of this behavior a swing-up controller must be realized to overcome the force of gravity. For this task an energy shaping controller is realized. As indicated by the name an energy-shaping controller regulates the energy in a system to a set reference energy.
The reason why an energy controller works well for this task is that the direction of rotation is not relevant for the sum of energies. Only the magnitude of the rotational velocity.
The sum of energy in a real physical system such as the inertia wheel pendulum is:
$KE + PE \geq 0$. Where $KE = f(\dot{\theta_p})$ and $PE = f(\theta_p)$
Swingup action, which requires constant directional changes can therefore easily be realized using the following controller strategy:

$$u = -K\dot{\theta_p}(E - E_{sp}) \tag{5.7}$$

To simplify the controller design the inertial wheel pendulum is evaluated for a rigid pendulum with a single mass. This mass m_t is the total mass in the system located at distance l_{mt} from the axis of rotation. Where the total moment of inertia in the system is I_t. Thus the potential and kinetic energy functions can be derived:

$$PE = m_t g l_{mt}(1 + cos(\theta_p)) \tag{5.8}$$

$$KE = \frac{1}{2} I_t \dot{\theta_p}^2 \tag{5.9}$$

Where the total energy is:

$$E = KE + PE \tag{5.10}$$

The chosen energy setpoint (E_{sp}) is equal to the potential energy of the system when it is at it's equilibrium point $\theta_p = 0$. This is the max potential energy the physical system can have.

$$E_{sp} = PE(0) = m_t g l_{mt}(1 + cos(0)) = 2 m_t g l_{mt} \tag{5.11}$$

Equation 5.7 can now be filled in to get the controller equation:

$$u = -K\dot{\theta_p}(\frac{1}{2} I_t \dot{\theta_p}^2 + m g l_{mt}(1 + cos(\theta_p)) - 2 m_t g l_{mt}) \tag{5.12}$$

Tuning the Energy Shaping Controller The energy shaping controller has three values that need to be calculated, derived and tuned. The first value that needs to be determined is l_{mt}, which can be approximated by giving the pendulum a small swing ($\theta_p \leq 1 radian$) and plotting the output. The length of the pendulum arm is approximated using the peak to peak time T_{pp} and the following function:

$$l_{mt} \approx (\frac{T_{pp}}{2\pi})^2 g \tag{5.13}$$

The controller is intended to swing the pendulum up, once in close proximity of $\theta_p = 0$ the PID controllers take over. Since the maximum controller output is defined by the max reference voltage which is 5v. The value K can be set quite high to cause the controller to aggressively regulate the system to it's setpoint. For this system K is set to be 1 until m_t is found. Then it is scaled up to decrease the time to reach the set-point, no matter how high K becomes the controller output will remain max 5 to min -5 volt.

The last parameter to find is m_t. This can be done easily by tuning the parameter. Since the

system is under actuated not all system configurations can be maintained. The only positions in the phase $0 \leq \theta_p < 2\pi$ for theta that can be maintained are $\theta_p = 0$ and $\theta_p = \pi$. Therefore any E_{sp} that does not result in $KE = 0$ at $\theta_p = 0$ or $\theta_p = \pi$ causes the system to oscillate. When plotting the position of the pendulum while it is oscillating the peak has no velocity thus no kinetic energy. Using this knowledge tuning m_t can be done by increasing the total mass until the system oscillates with the peaks near $\theta_p = 0$ and $\theta_p = 2\pi$. This method gives an intuitive way to evaluate the total mass with respect to E_{sp}, because the angle θ_p at the desired equilibrium is known to be 0 or 2π.

5.4 Controller realization and Analysis

The controlled system is plotted and the resulting plots of the energy error and system states are analyzed in this chapter.

Energy Controller The energy controller is the focus of this paragraph, using the plots of the energy error and energy controlled system states conclusions will be drawn about the behavior of the controller.

All the parameters that needed to be found for the energy controller are listed in the table below. These values are acquired by following the steps described prior.

$l_m t$	0.23	m	Distance to center of mass
m_t	0.1	kg	Approximated weight
K	6	-	Controller gain

Table 5.1: All the energy controller parameters

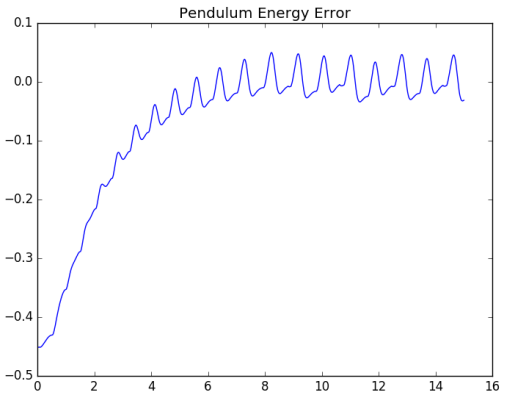

Figure 5.5: The Energy error of the energy controlled system, x axis is the time in seconds, the y axis is the error in Joules

In 5.4 on the right the resulting energy error in the system can be seen while the controller is on. The line in this figure seems to follow a (over)critically damped step response, however once the error starts become very small peaks in the graph start appearing.
The line in this figure seems to follow a critically damped step response, however as the error approaches 0 ripples start appearing. The energy-shaping controller that was designed does not pay attention to the kinetic energy of the inertia wheel, it assumes the pendulum is a rigid body rotating about it's axis.
The decision to realize the controller as a rigid pendulum was conscious, it was assumed that an acceptable swing up behavior could be achieved without modeling the Inertia wheel energy into the system.

The system states are also plotted while the energy controller is turned on, see 5.6. These are the system states plotted after the energy-shaping controller parameters are tuned. The important trait of this plot is the position of the pendulum which clearly oscillates from roughly 0.5 rad to 6.1 rads. Because $2\pi - 6.1 < 0.3$ the pid controller can take over, and because θ_p remains within the range of 0 to 2π the PID controllers can reliably take over the controller action.

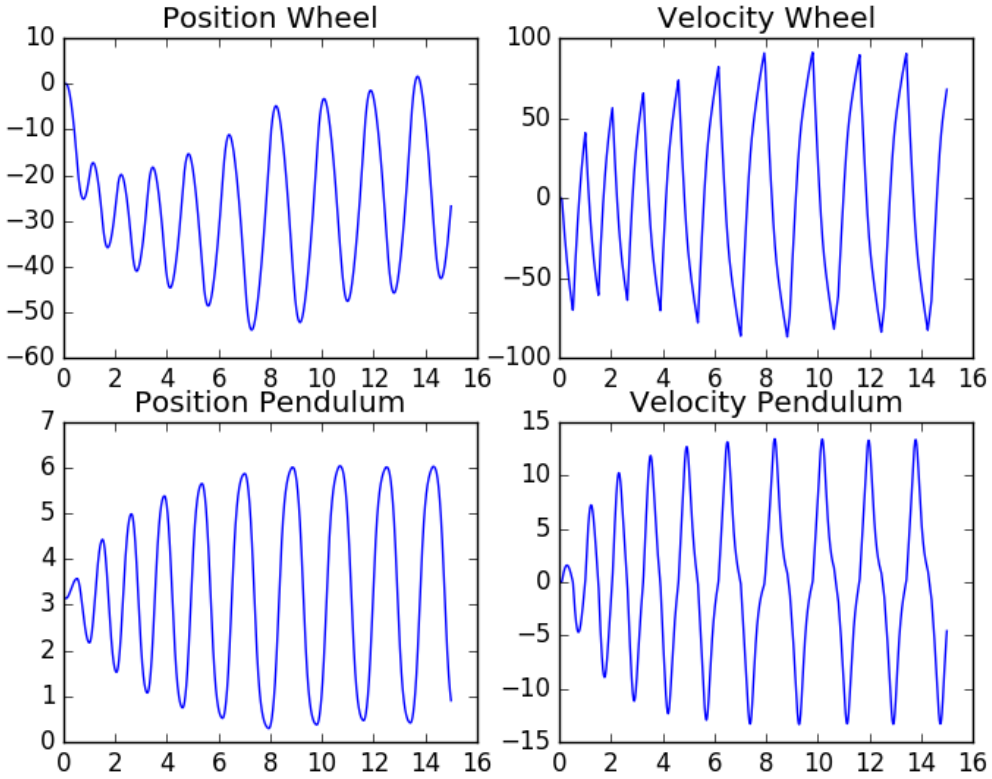

Figure 5.6: The controlled system states, for all graphs the x axis is time in seconds, the y axis is in radians for position graphs and rad/s for velocity graphs

Combined controllers All controller behaviors are combined in this paragraph and plots are made of the resulting simulated system behavior. Within the range of $-0.3 \leq \theta_p \leq 0.3$ the parallel PID controllers are producing the control action, for all other values of θ_p the energy controller is functioning.

Pendulum position controller parameters.

Input: θ_p	
K_p	50
K_i	100
K_d	2

Wheel velocity controller parameters.

Input: $\dot{\theta}_w$	
K_p	0.01
K_i	0.1
K_d	0

The wheel PID weight factor, as can be seen in 5.4, that is used is 0.75.

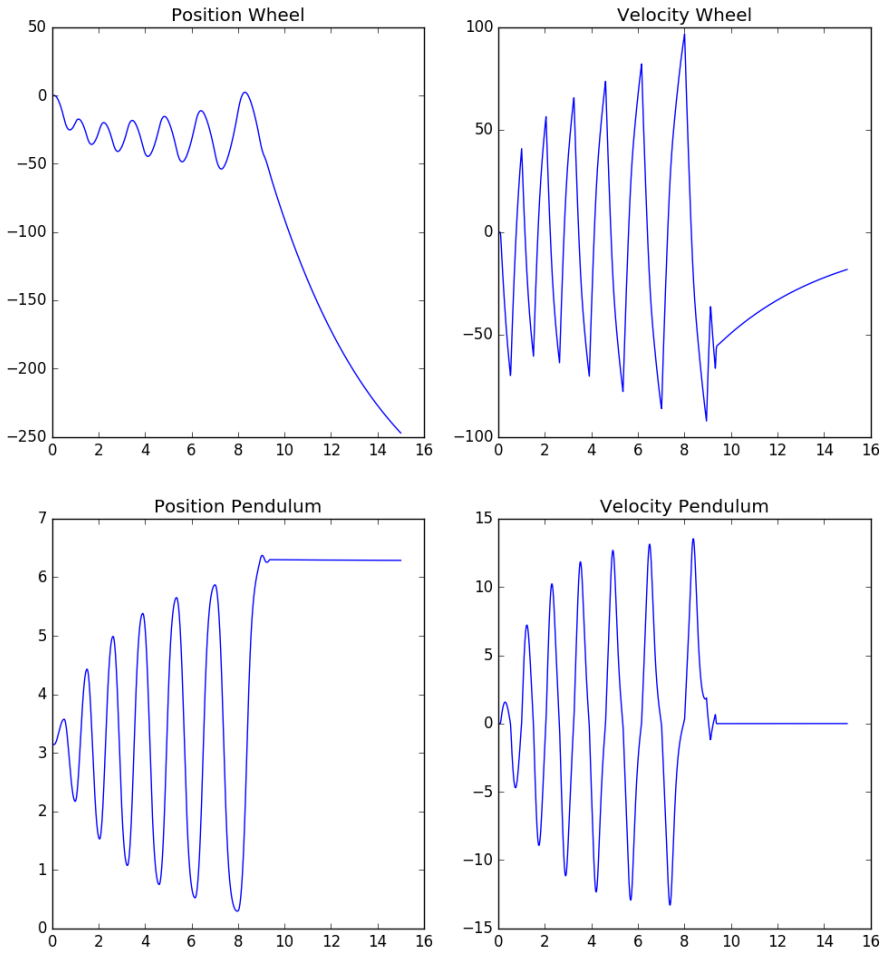

Figure 5.7: The controlled system states, for all graphs the x axis is time in seconds, the y axis is in radians for position graphs and rad/s for velocity graphs

In the figure above, 5.7, the behavior of the fully controlled system can be seen.
Determining that the fully realized controller functions as intended is done by evaluating the following points.
1. Wheel velocity is driven to 0 as time passes. See right top plot in figure 5.7
2. Pendulum position is maintained at $sin(\theta_p) = 0$. See left bottom plot in figure 5.7
3. Energy fluctuations are also removed from energy error. See figure 5.4
4. PID controllers take over in the range $-0.3 \leq \theta_p \leq 0.3$. See left bottom plot in figure 5.7

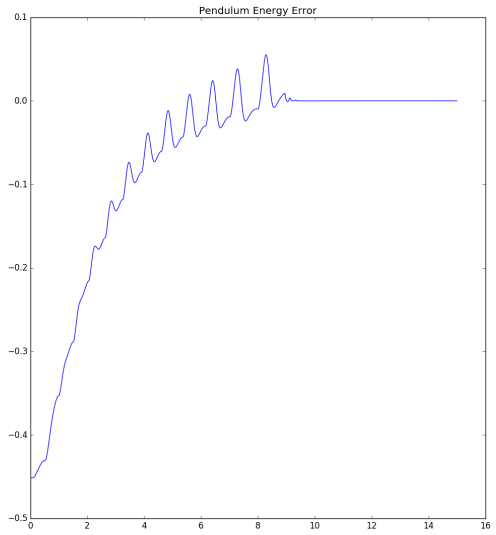

Figure 5.8: The energy error of the fully controlled system, x axis is the time in seconds, the y axis is the error in Joules

In the energy error plot on the left it can be seen that the ripples in energy are not causing enough disturbance to hinder the PID controllers from successfully taking over control.
It can therefore be concluded that the controllers successfully regulate the system with respect to the set controller requirements.

5.5 Implementation of controllers on real system

The controllers have not yet been implemented on the real system since the brush-less DC motor driver has not yet run reliably with the motor. Therefore it is impossible to actuate the physical system.

Conclusions

The system modelling was troubling in the beginning, due the complexity of the inertia wheel pendulum system the free-body-diagram was possible but not intuitive. Then a study about Lagrangian mechanics helped to see the more intuitive manner to approach these kind of complex systems based on energy.

The design of the linear controller went very fast once the system was properly modelled. The swingup controller was approached intuitively, however a different kind of approach is still optional. Regarding the adaptive controller, the gain scheduling strategy was by far the best option because both the because both the linear controller and energy shaping swingup controller could just be swapped depending of the pendulum angle of the system.

Finally, the realization of the inertia wheel pendulum was difficult. Due to problems with the BLDC-motor, combined with hall sensors and a BLDC-driver, the system seemed to have a lot of bouncing issues. In the next coming months the work on the physical system will be continued. For now the modelling and controlling of the physical inertia wheel pendulum got very close but not yet finished and this a motivation to keep working on the physical system until it is finished.