NTNU – Trondheim
Norwegian University of
Science and Technology

Stabilization Control of an Autonomous Bicycle

Modeled as an Acrobot with Angular Limitation

Jørgen Herje Nilsen

Master of Science in Cybernetics and Robotics
Submission date: June 2014
Supervisor: Amund Skavhaug, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF ENGINEERING CYBERNETICS

MASTER THESIS

Stabilization Control of an Autonomous Bicycle

Modeled as an Acrobot with Angular Limitation

Author:

Jørgen Herje NILSEN

Supervisor:

Asoc.Prof. Amund SKAVHAUG

June 2014

NTNU – Trondheim
Norwegian University of
Science and Technology

Preface

This report is my master thesis conducted during my 10th semester, in the spring of 2014, at the Department of Engineering Cybernetics at NTNU Trondheim.

Throughout the semester many interesting and challenging problems has occurred. The assignment has given me the challenge of working with advanced and independent problem solving, providing great learning and experience within scientific research. By use of relevant academic literature and discussions with professional experienced associates, the occurring problems has given a great experience of structured work, cooperation and documentation, a great lesson for my coming work career.

Trondheim, 9. June 2014

Jørgen Herje Nilsen

Acknowledgment

First and utmost I want to thank my supervisor, associate Professor Amund Skavhaug, for giving me the opportunity of working on his precious autonomous bicycle, with a new approach. By his idea of modeling the system as an inverted double pendulum, his optimistic attitude has given me the possibility of approaching the problem as I want, resulting in a very interesting assignment in my point of view. I also want to thank Roy Nilsen for the assistance throughout the semester, giving me the possibility of continuously discussing the system analysis for improving results. In addition, the NTNU subscription on IEEE should be thanked, making it possible to find relevant literature on the Internet without making me commit personal bankruptcy. Finally yet importantly, I want to thank Wärtsilä Norway AS for giving me the experience of working with electrical power converters and motor drives during my work as summer assistant the last four years.

Summary

With an inverted pendulum mounted on the bicycle frame, the system is corresponding to a bicyclist who applies balancing torque from the hip. This thesis present a mathematical system model of the autonomous bicycle, modeled as an inverted double pendulum with actuation at the joint connecting the two system links, better known as an Acrobot. The Acrobot is a well-known underactuated robot manipulator, which implies that only the mounted inverted pendulum can obtain instantaneous acceleration in arbitrary direction by use of the system actuator. The goal is to stabilize the autonomous bicycle around the upright unstable equilibrium by use of the control torque applied from the DC motor mounted to the bicycle frame and the inverted pendulum. With this single control torque, the underactuated system introduces a challenging control-problem. By use of the system model in the authors own work[12], angular limitation of the bicycle tilt angle and the inverted pendulum angle are included into the derived Acrobot model, to get a complete mathematical model equivalent to the real life bicycle system. As the goal is to develop a stabilization controller, the thesis further investigates the most convincing controllers presented in [12] - Lai et al. [11] and Kobayashi et al. [9]. These articles suggests to defined controller subspaces by dividing the system control into balance and swing-up control. The balance controller is designed by use of system linearization around the upright unstable equilibrium, where the linear system matrices are used to create an LQR balance controller for optimal state feedback control by pole placement. The suggested controllers utilizes nonlinear energy based controllers for system swing-up. As the controller-designs increases system energy by use of the bicycle outside of the angular limitations, the resulting swing-up control is unable to move the system states within the balance subspace. Thereby the controller implementations are verified as unsuccessful stabilization controllers for the bicycle system, through simulation. Further on a new controller strategy is investigated, where the partially feedback linearization is utilized to cancel out the nonlinearities of the actuated system angle. By dividing the stabilization controller into balance and swing-up control, the partially feedback linearized system representation is utilized to design linear PD-controller for system swing-up. A new LQR controller was designed by linearization around the upright equilibrium of the partially feedback linearized system. Through simulation the swing-up PD-controller is tuned and tested to achieve the desired system functionality by utilization of the inverted pendulum and counteracting torque onto the bicycle. With torque disturbance applied onto the bicycle, the stabilization controller functionality is tested and analyzed. The complete controller-design is verified to successfully stabilize the bicycle, from initial positions at the limit, at its upright position by use of an LQR balance controller and swing-up PD-controller. Without exceeding the maximum torque able to apply from the system actuator, the complete controller satisfies the requirements of the autonomous bicycle stabilization.

Sammendrag

Med en invertert pendel montert på sykkelrammen er sykkelsystemet tilsvarende en syklist som balanserer med rotasjon i hoften. Denne oppgaven presenterer en matematisk modell av den autonome sykkelen, modellert som en dobbel-invertert pendel med pådrag påført i leddet hvor sykkelen og den inverterte pendelen knyttes. Dette er bedre kjent som en Acrobot, en underaktuert robotmanipulator - som resulterer i at kun den påmonterte inverterte pendelen kan oppnå momentan akselerasjon i virkårlig retning. Ettersom at målet er å balansere sykkelsystemet ved det oppreiste ustabile likevekts-punktet ved bruk av DC motoren som pådragsorgan, vil det bli et utfordrende kontrollproblem da det kun er ett pådragsorgan på systemet. Med bruk av den matematiske modellen utviklet i forfatterens eget arbeid[12] er fysiske vinkelbegrensinger inkludert i Acrobot-modellen for å gi en systemmodell som er ekvivalent med det fysiske sykkelsystemet. Videre blir de mest overbevisende kontrollerne, presentert i [12], undersøkt. Kontrollerdesignene fra Lai m. fl.[11] og Kobayashi m. fl.[9] foreslår og dele robot-manipulatorens arbeidsområde inn i et balanseområde og oppsvingsområde, hvor henholdsvis balansekontroller og oppsvings-kontroller blir benyttet. Balansekontrolleren er designet med bruk av linearisering om det ustabile likevekts-punktet, hvor de lineære systemmatrisene er brukt for å lage en LQR-kontroller for optimal tilstandskontrollering ved hjelp av pol-plassering. Disse foreslåtte kontrollerne utnytter ulineære energibaserte kontrollere for oppsving. Siden disse energibaserte kontrollerne øker systemets energi ved bruk av sykkel-leddet utenfor de definerte vinkelbegrensningene, er de foreslåtte kontroll-designene ute av stand til å bevege systemet inn i balanseområdet for balansering av sykkelsystemet. Gjennom simulering er disse kontroller-implementasjonene verifisert som mislykkede stabiliserings-kontrollere for den autonome sykkelen. Videre blir en ny kontrollerstrategi undersøkt. Ved bruk av delvis tilbakekoblings-linearisering vil ulinearitetene knyttet til den inverterte pendelen bli kansellert. Inndelingen av balansekontroll og oppsvings-kontroll er videre utnyttet, hvor den delvise lineariseringen er brukt til å ha en lineær PD-kontroller i oppsvingsområdet. En ny LQR-kontroller er også utviklet, basert på linearisering om det ustabile likevekts-punktet av det delvis tilbakekoblings-lineariserte systemet. Gjennom simulering blir PD-kontrolleren justert og testet, for å oppnå ønsket kontrollerfunksjonalitet, hvor den inverterte pendelen blir utnyttet for å påføre et motvirkende moment på sykkelen. Robustheten blir også testet ved å påføre en forstyrrelse til systemet. Det endelige kontroll-designet, med LQR balansekontroller og PD oppsvings-kontroller, blir videre verifisert som en vellykket stabiliseringskontroller for sykkelsystemet, med initial posisjoner på vinkelgrensene. Uten å overgå det maksimale pådraget til pådragsorganet, oppfyller kontroll-designet systemkravene til det autonome sykkelsystemet.

Acronyms

- IMU - Inertial Measurement Unit
- LQR - Linear Quadratic Regulator
- MIMO - Multiple Input Multiple Output
- DOF - Degree Of Freedom
- DC - Direct Current
- P - Proportional
- PI - Proportional Integrating
- PD - Proportional Derivative
- PID - Proportional Integrating Derivative

List of Figures

Contents

Chapter 1

Introduction

In this thesis balancing of an autonomous bicycle system will be investigated and developed. By use of an inverted pendulum mounted on the bicycle frame, controllers will be analyzed and tested for stabilization of the complete bicycle system by use of the actuated inverted pendulum. These controllers will be developed for the autonomous bicycle system designed by Dag Christan Ånnestad at NTNU in 2011.

1.1 Motivation

In 2011 Dag Christian Ånnestad, [14], succeeded to develop a simple, but functioning system, by designing hardware and software for driving the autonomous bicycle and control the steering. Based on the idea of Amund Skavhaug an inverted pendulum is mounted to stabilize the bicycle, to make it balance around the upright position at both zero and nonzero velocity. Thus the bicycle does not need support for stabilization at the start-up phase and while driving at low velocities. The goal is to utilize the authors own work done in the project thesis [12] to develop a stabilizing controller for the bicycle system. By use of the mathematical system model and the controllers presented in [12], further research of the stabilization controller will be conducted. With the inverted pendulum representing the upper body of a cyclist, the goal is to investigate various controller strategies for system stabilization within defined angular limits. As the control torque applied at the joint connecting the bicycle frame and the mounted inverted pendulum, the system will be represented a robot manipulator better known as an Acrobot, with system limitations. Physical limitations on angles and angular velocity will be included in the system model to make the model equivalent to a real life humanoid cyclist. Control theory and physics will be applied to analyze and understand the system for further investigation and development of the stabilization controller for the bicycle used in recruitment and motivation of students at the Department of Engineering Cybernetics at NTNU.

Problem Formulation

By utilizing the mathematical model derived in the authors previous work in [12], investigation of controllers will be conducted through analysis and simulation of the bicycle system. Through system simulation, the controllers will be tested and verified for the limited Acrobot system for future system implementation. In this thesis, Acrobot-controllers from various literature presented in the authors own work [12] will be investigated, to verify if the controllers works sufficiently for the limited Acrobot system. If these controllers are not satisfying the system requirements, further research will be conducted to obtain knowledge and understanding of the control problem to present new controller strategies. By use of these controllers, the goal is to stabilize the system around the upright equilibrium and balance the bicycle.

- Investigate relevant controllers for system stabilization.
- Design controllers for system simulation.
- Analyze controllers through simulation.
- Obtain knowledge and understanding of the control problem.

For readability, previously derived equations in [12] and solutions of the occurring problems will be discussed throughout the thesis.

Literature Survey

The articles presenting the Acrobot control, shows how the system stabilizing control is designed by two types of controllers: balancing and swing-up controllers. It is shown by Lai et al. in [10] and [11] how the Acrobot stabilization problem can be solved by defining system subspaces. The system controllers are designed and applied in their corresponding subspaces. Balancing control is designed as an LQR controller, while the swing-up controllers are energy based controllers, defined by either system energy or suggested Lyapunov functions. Kobayashi et al. [9] shows that the swing-up and balancing control can not be designed by a single feedback controller, due to excessively large feedback gain. The control torque is designed by one LQR balancing control and energy based swing-up control, as in [10] and [11]. Among the Acrobot literature this system control design often occur. With the video [1] and the corresponding paper [2], Andersen et al. present various nonlinear controller, both with and without angular limitations for Acrobot balancing. This video shows a successful attempt of balancing the Acrobot at the upright equilibrium, which increases the knowledge and understanding of the system behavior. The controller presented in [2] are presented with system time response and shows that some of the controllers are inapplicable on the bicycle system, as the system angles

moves outside the defined angular limitations. In [18] energy based control was applied as well. In addition, this article perform partial feedback linearization, canceling out some of the nonlinearities of the Acrobot system, to achieve less complex stabilization controllers.

What Remains to be Done?

As the system model is derived and stabilization controllers are developed, the suggested control theories can be utilized for future design and implementation of the complete bicycle system stabilization controller. The controller can then be implemented and tested on the real bicycle system to see how well the theoretical controller theories verified by system simulation works on the physical system.

1.2 Objectives

The main objectives of this Master's project are

1. Analyze and discuss system actuation by DC motor
2. Utilize system model with limitations for simulation of bicycle system
3. Introduce stabilization controllers for system stabilization
4. Analyze and test the stabilization controllers through simulation
5. Test robustness of stabilization controller by perturbation

1.3 Limitations

In this study, the research is limited to analyze and investigate the controller performance of the mathematical system model through simulation. The previous work on the autonomous bicycle model will not be tested until a complete controller is developed. The focus will be on the mathematical approach and physical understanding. Controller research will be conducted where control theory will be applied for investigation of possible designs for future implementation of a complete stabilization controller for the real life bicycle system.

1.4 Approach

The system will be represented as an inverted double pendulum with the control torque placed at the link connecting-joint. This system is better known as an Acrobot in various literature. As the goal is to create a controller for balancing the bicycle system around the upright position, the author will utilize his own previous work in [12] with a mathematical system model including angular limitation and suggestions to possible stabilization controllers. These controllers will be further investigated and tested. If these controllers are not sufficient, further research will be conducted to develop a complete system controller. The mathematical system model is extend the with angular limitations such that the model is equivalent to the real life bicycle system. Due to the electrical DC motor, limitation of the controller torque must be taken into account for the system actuator. Equations for the electrical DC motor actuator-system will derived and analyzed.

1.5 Structure of the Report

The rest of the report is structured as following:

- Chapter 2: Gives an introduction to the physical bicycle system with description of the angular limitations, sensors, parameters, gear box and system actuator (DC motor).
- Chapter 2: Presents modifications and illustrations of the real physical bicycle system. The bicycle is shown to provide a clear understanding of the relation between the real system and the system model derived in [12] and presented in Chapter 3
- Chapter 3: Contains the derivation of the system model of the bicycle system represented as an Acrobot with angular limitations.
- Chapter 4: Present system controllers for the system model from Chapter 3, where linear and nonlinear control is presented, investigated and analyzed.

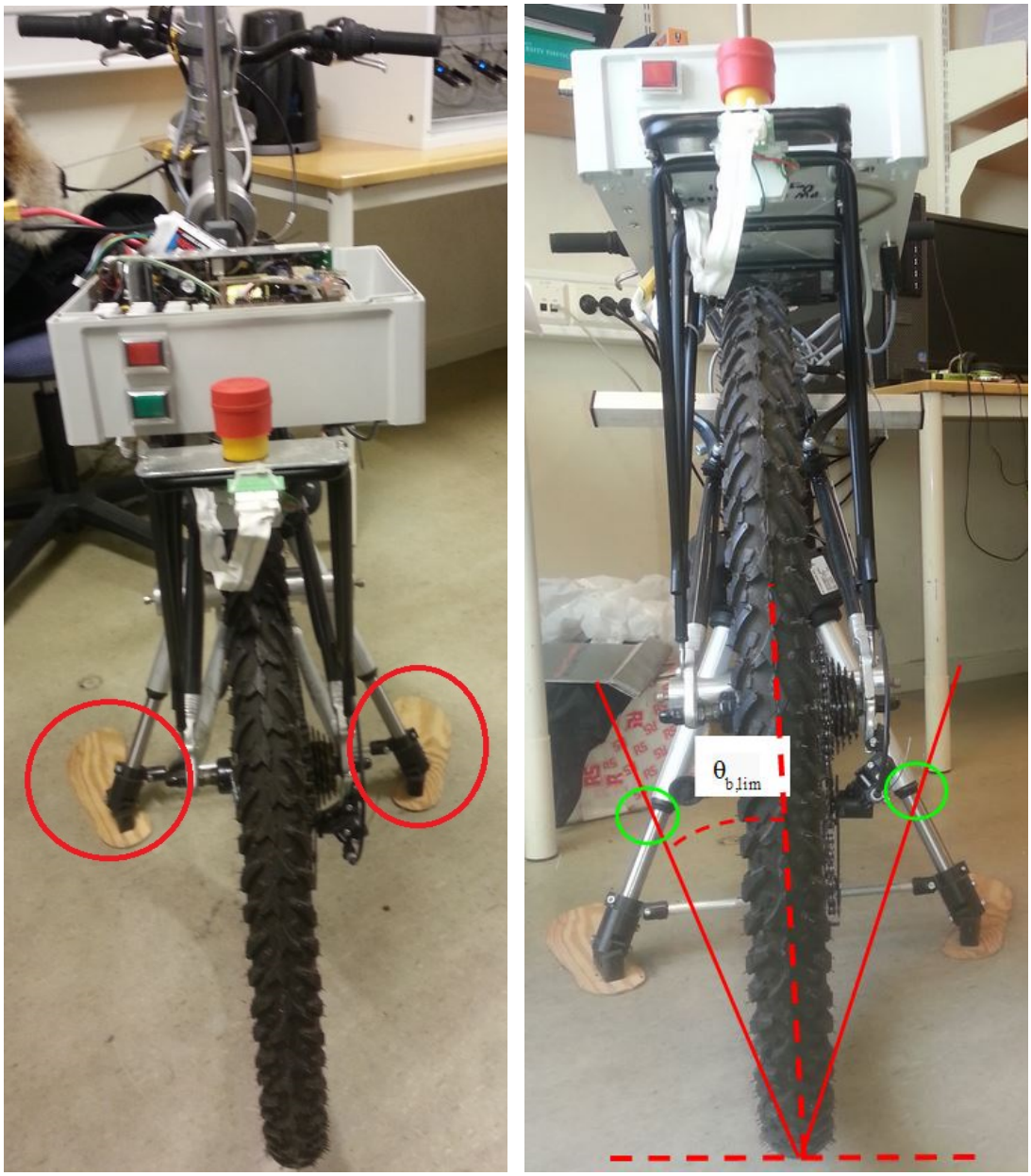
Chapter 2

Bicycle System

In this chapter hardware and software of the autonomous bicycle system, constructed through previous thesis' at NTNU, is described. With the goal of stabilizing the bicycle system at the upright position, this chapter will present specifications and details of the bicycle design to give an understanding of the control problem. As the inverted pendulum was not included in the previous model designs of the physical bicycle, the specifications presented in this chapter are utilized in the derivation of the new mathematical system model, presented in Chapter 3, and based on the authors own work in [12].

2.1 Bicycle With Limitation

The goal is to stabilize the autonomous bicycle system by use of the inverted pendulum mounted on the bicycle. When the system is modeled as an Acrobot, the model of the physical bicycle system has to include angular limitations. Both the bicycle tilt angle and the inverted pendulum angle is limited due to the fact that neither of the two links are able to make a full rotation about their rotational axis. The pendulum is limited by the bicycle frame, making it able to rotate about $\pm 156°$. The bicycle tilt angle is limited to approximately $\pm 30°$ when the mechanical feet, illustrated in Figure 2.1(a), are elevated. In the implementation made by Ånnestad, the pendulum limitation is set to $\pm 60°$, as an assumed maximum angle to avoid damage to the pendulum motor. Throughout this thesis, the system simulations will be performed with angular limitations of: $\theta_{b,lim} = \pm 30°$ and $\theta_{p,lim} = \pm 110°$, such that the system is able to perform the stabilization. Thereby the limitation implemented by Ånnestad has to be modified before complete controller implementation is conducted.



(a) Bicycle tilt angle limitation due to mechanical feet, marked red

(b) Illustration of angle measurement, lifted feet marked green

Figure 2.1: Bicycle tilt angle limitation due to mechanical feet

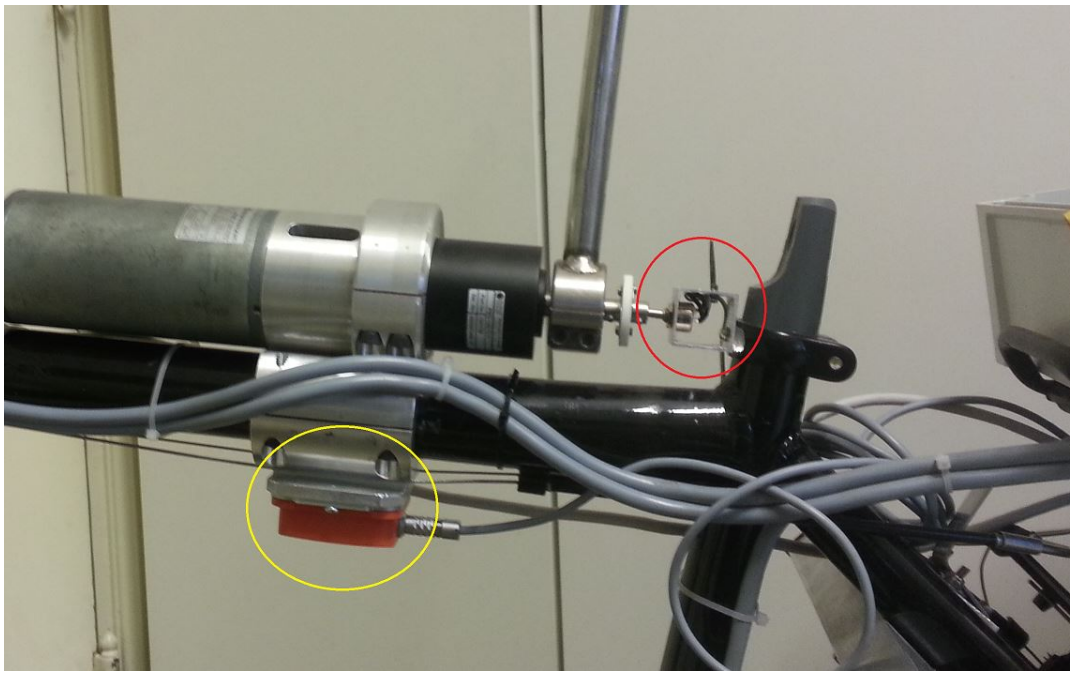


Figure 2.2: Sensor placement - IMU marked yellow and Potentiometer marked red

2.2 Sensor Placement

The potentiometer measuring the position of the inverted pendulum is mounted on the bicycle frame, as shown in Figure 2.2. The angle output from the sensor gives a pendulum angle relative to the bicycle frame, resulting in an angle relative to the bicycle tilt angle. The IMU placed on the bicycle frame gives two important output measurements: The bicycle frame acceleration in (x, y, z)-coordinates and the roll, pitch, yaw angles of the IMU. These measurements can be used to calculate the bicycle tilt angle, θ_b, which is defined in Section 3.2. When using the roll, pitch, yaw representation from the IMU, the measured roll angle correspond to the tilt angle, θ_b, of the bicycle, as shown in Figure 2.3. The of angle ϕ correspond to the roll angle measured with the IMU, which states how many degrees the IMU is rotated about its z-axis. The roll angle, ϕ is measured, and by the geometry of the system, the bicycle tilt angle is:

$$\phi = \frac{\pi}{2} - (\frac{\pi}{2} - \theta_b) = \theta_b \tag{2.1}$$

As seen in Eq. 2.1, the roll angle of the IMU correspond to the tilt angle of the bicycle, which is intuitive. The tilt angle can also be calculated with the (x, y, z)-coordinates from the IMU. To increase the reliability of the sensor output, the angle can be calculated with both methods and compared. Note that the (x, y, z)-coordinates are derived by double differentiating the (x, y, z)-acceleration measured with the IMU, and thereby the unreliability increases as the system noise can be enlarged.

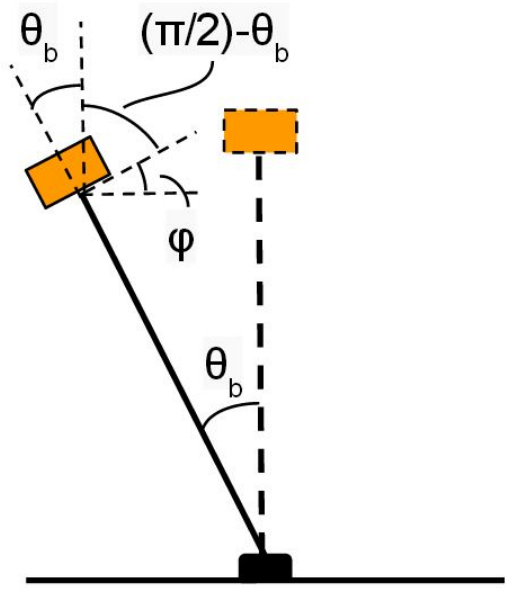


Figure 2.3: Bicycle tilt angle represented with roll, pitch, yaw output from IMU

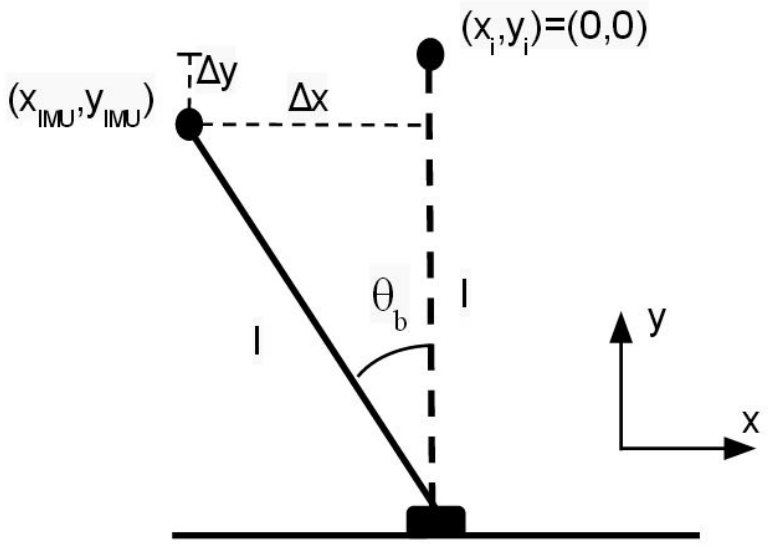


Figure 2.4: Bicycle tilt angle based on (x,y,z)-acceleration from IMU

By use of Figure 2.4 the bicycle tilt angle can be calculated. With the initial IMU position $(x_i, y_i) = (0,0)$, the measured IMU-position, (x_{IMU}, y_{IMU}), and positive angle defined counterclockwise, two equations can be derived:

$$\theta_b = arcsin\left(\frac{\Delta x}{l}\right) \quad , \quad where\, \Delta x = x_i - x_{IMU} \tag{2.2a}$$

$$\theta_b = arccos\left(\frac{l-\Delta y}{l}\right) , \quad where\, \Delta y = y_i - y_{IMU} \tag{2.2b}$$

where l is the height of the bicycle.

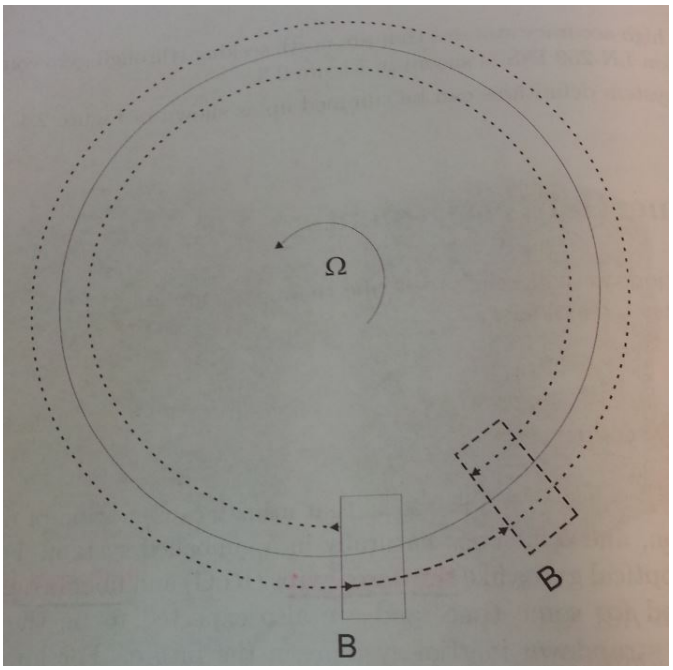


Figure 2.5: Illustration of the optical gyro functionality by Vik [23]

2.2.1 IMU

In Vik [23] describes the IMU as an inertial measurement unit, which consists of an assembly of inertial sensors (ISA), hardware to interface the ISA and low level software for down-sampling, temperature calibration and vibration compensation. The INS can be divided into two classes - the Gimbal and strapdown. As the IMU on the bicycle system is attached to the frame and follows the movements, the system is equipped with a strapdown IMU. The most common inertial instruments are the gyroscope and accelerometer.

The classical gyro is a Gimbal system where the spinning wheel that utilizes the momentum to detect rotation. For the strapdown, as the IMU on the bicycle, the optical gyros known as the ring laser gyro (RLG) and fiber optic gyro (FOG) are the most common designs due to high accuracy. Vik [23], states: "The optical gyros detects a path between two optical beams going in opposite direction and interfering with each other. A light source sends light through a beam splitter (B), which sends the light in opposite directions around the circular path. If the interferometer is non-rotating, the light beam will meet at point B, where an interference pattern is detected. If the interferometer rotates counter clockwise with an angular velocity of Ω, the counter clockwise beam will have to travel further to reach point B, while the other beam will travel a shorter path. The result is that the interference pattern will move at the detector." Figure 2.5 illustrates this functionality described by Vik [23].

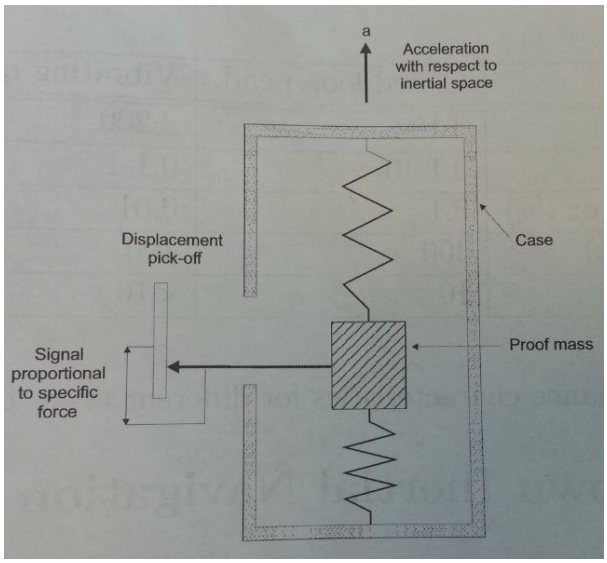


Figure 2.6: Illustration of the accelerometer functionality by Vik [23]

In [23] Vik presents the most important accelerometers, the mechanical and the vibratory. The mechanical accelerometers are based on Newton's second law of motion, $F = ma$. A mass is balanced by the tension of the mounted springs, illustrated in Figure 2.6, under steady state conditions. When the instrument is subjected to acceleration along its sensitive axis, the mass tends to resist the change of movement due to its own inertia. The mass displacement due to acceleration results in extension of the springs which is proportional with the accelerating force. By measuring the displacement, the acceleration can be calculated and given as an output from the accelerometer. The vibratory accelerometers are based on measurement of frequency shits due to increased or decreased tension on a string. Vik [23] states: "When the accelerometer mass attached to a quartz beam is loaded, the frequency of the quartz beam increases. The difference in frequency is measured, and is proportional to the applied acceleration."

The position of the bicycle, measured with the IMU, can be inaccurate as the system angles are derived by adding detected changes to its previously calculated positions. If errors occur when the system operates, the calculation of the system angles at the next time sample is based on the previous measurement containing an error. This is a phenomenon known as drift, which is a common disadvantage in the IMU. In navigational systems, the IMU is often calibrated by use of a GPS. As the IMU is utilized to find the angles around the rotational axis of the bicycle system, this sort of calibration is inapplicable, and the system will suffer from the drifting IMU-measurements. The uncertainty of the angular measurement must be taken into account when the stabilization controller is implemented on the real life bicycle system, where sensitivity analysis can be performed. As this thesis will focus on investigating controllers for system stabilization through simulation and analysis, this uncertainty is neglected and ideal angular measurements are assumed throughout the thesis.

Parameter	Value	Description
m_b	31.12	Bicycle mass [kg]
m_p	5.0	Pendulum mass [kg]
r_b	0.4	Distance to bicycle center of mass [m]
r_p	0.2	Distance to pendulum center of mass [m]
l	0.72	Height of bicycle [m]

Table 2.1: Bicycle parameter values

2.3 Measured System Parameters

In the work of Ånnestad [14], system parameters were determined and measured. All these parameters were measured for the derivation of a complete model for the bicycle system. As this thesis presents the bicycle system as an inverted double pendulum known as an Acrobot, some of the parameters will change when the bicycle is presented as a manipulator link. As the work has been directed towards the mathematical modeling of the system, the parameters in Table 2.1 are approximated values, based on the parameter values from [14], utilized in the system simulation.

Note that the parameter value of the bicycle mass, m_b, is a lot larger than the inverted pendulum mass, m_p. Comparing this with a real life bicyclist, the inverted pendulum mass is a lot less than the upper body weight of the cyclist. Thereby the leaning angle required to balance the bicycle system must be larger when the moment of inertia of the inverted pendulum is less than the moment of inertia of the bicycle.

2.4 GearBox

The system is equipped with a gear between the DC motor and the mounted inverted pendulum. With the gear ratio of $R = 72$, the system is able to operate with an inverted pendulum torque, τ_p, 72 times as high as the motor torque, τ_m, as stated in Section 2.5.1. By use of the gear, the system is able to accelerate the inverted pendulum and create a desirable torque to stabilize the bicycle at the upright position. For the gearbox data sheet see Appendix C

		SI unit
Nominal voltage, V_{an}	24	V
Nominal speed, N_n	3350	min^-1
Nominal torque, τ_n	27	Ncm
Nominal current, I_{an}	4,9	A
Demagnetization current	33	A
No load speed, N_0	3650	min^-1
No load current, I_0	0,4	A
Starting torque	211	Ncm
Efficiency	80	%
Moment of inertia	750	gcm^2
Weight	1,7	kg

Table 2.2: Motor performance data from data sheet

2.5 Motor Model For Pendulum System

In this section the model of the system actuator controlling the mounted inverted pendulum is presented, an audited version of the authors own work in Chapter 6 in [12]. The desired control torque applied at the link-connecting joint of the bicycle system is generated by a DC motor. When designing the balancing controller it is important that the motor control is designed as wanted. The pendulum motor performance will determine quality of the complete bicycle system control, which is equipped with a "Dunkermotoren D.C. motor GR 63x55" motor. The following motor equations are derived from [13] by use of parameters from the motor data sheet listed in Table 2.2. For the motor data sheet see Appendix C.

2.5.1 DC Motor Model

The bicycle system has a torque input to the system, as stated in Eq. 3.8, thus the goal for the motor model is to control the output torque:

$$\tau_m = \tau_e - \tau_{loss} = \phi_m \cdot I_a - k \cdot \omega \tag{2.3}$$

Where τ_e is the ideal magnetic air gap torque. τ_{loss} represent the iron losses (eddy current and magnetic hysteresis losses), winding and friction losses. The DC motor drive can be described by the following equation:

$$V_a = R_a \cdot I_a + L_a \cdot \frac{dI_a}{dt} + \omega \cdot \phi_m \tag{2.4}$$

Where L_a is the motor inductance, ω is the rotational velocity and ϕ_m is the magnetic motor flux, which is constant. Assuming L_a to be 0,4 mH, the resulting time constant becomes $T_a = \frac{L_a}{R_a} \approx 0.9$ ms, R_a is given in Eq. 2.12. Note that the rotational velocity, ω, correspond to the angular velocity of the

pendulum link of the bicycle system with gear ratio $R = 72$, i.e. $72 \cdot \dot{\theta}_p = 72 \cdot \Omega_p$. Assuming no loss in the gear, the power of the motor is equal to the power of the inverted pendulum:

$$\omega \cdot \tau_m = \Omega_p \cdot \tau_p \tag{2.5}$$

Thereby the torque of the inverted pendulum is:

$$\tau_p = \frac{\omega}{\Omega_p} \cdot \tau_m = R \cdot \tau_m \tag{2.6}$$

With the nominal torque, τ_n, of the DC motor from Table 2.2, the nominal torque of the inverted pendulum, $\tau_{p,n}$, is given by:

$$\tau_{p,n} = R \cdot \tau_n = 72 \cdot \tau_n = 19.44[Nm] \tag{2.7}$$

The motor is able to perform with 3-4 times the nominal torque in periods significantly shorter than the thermal time constant, which is assumed to be within the range of minutes for such a small DC motor. Thereby the maximum torque of the inverted pendulum can be $50 - 100$[Nm], as the control torque in the system stabilization is applied within the range of seconds. By potentiometer measurement of θ_p and calculation of the time derivative, the velocity ω can be found . At steady state the armature voltage in Eq. 2.4 can be rewritten as:

$$V_a = R_a \cdot I_a + \omega_0 \cdot \phi_m \tag{2.8}$$

At no load steady state, the armature voltage is equal the induced motor voltage:

$$V_a = R_a \cdot I_0 + \omega \cdot \phi_m \tag{2.9}$$

The rotation velocity is given by: $\omega = \frac{\pi}{30} \cdot N$, where N is the speed [rpm]. The no load expression can be expressed with the given motor no load speed, N_0:

$$\omega_0 = \frac{\pi}{30} \cdot N_0 = 382,23 \ [rad/s] \tag{2.10}$$

The nominal operation is given by:

$$V_{an} = R_a \cdot I_{an} + \omega_n \cdot \phi_m \tag{2.11}$$

By use of Eq. 2.9 and 2.11 the resistance; R_a, and flux, ψ_m, can be calculated by the set of two equations, no load and nominal load, respectively:

$$V_{an} = R_a \cdot I_0 + \omega_0 \cdot \phi_m \tag{2.12a}$$

$$V_{an} = R_a \cdot I_{an} + \omega_n \cdot \phi_m \tag{2.12b}$$

Eq. 2.12 can solved by the following matrix equation:

$$\begin{bmatrix} R_a \\ \phi_m \end{bmatrix} = \begin{bmatrix} I_0 & \omega_0 \\ I_{an} & \omega_n \end{bmatrix}^{-1} \cdot \begin{bmatrix} V_{an} \\ V_{an} \end{bmatrix} = \begin{bmatrix} 0,435 \\ 0,062 \end{bmatrix} \tag{2.13}$$

The nominal current $I_{an} = 4,9$ [A] and no load current $I_0 = 0,4$ [A] are given by the data sheet. $\omega_n = \frac{\pi}{30} \cdot N_n$.

2.5.2 Motor Losses

At the nominal operation, the system power is expressed as:

$$P_r = R_a \cdot I_{an}^2 = 10,45\ [W] \tag{2.14}$$

The power loss is given by:

$$P_{total\,loss} = V_{an} \cdot I_{an} - \tau_n \cdot \omega_n = 22,87\ [W] \tag{2.15}$$

where τ_n is $0,27$ [Nm], given by Table 2.2. The total loss is then further given by:

$$P_{total} = P_{total\,loss} - P_r = 12,43\ [W] \tag{2.16}$$

By Eq. 2.3 the motor torque loss is given as: $\tau_{loss} = k \cdot \omega$, giving:

$$k = \frac{P_{loss}}{\omega_n^2} = 1,01 \cdot 10^{-4}\ [Nms] \tag{2.17}$$

Thus, when k is defined, the motor output torque, τ_m, can be derived by Eq. 2.3. The complete system without balancing control is shown in Figure 2.7 where the inverted double pendulum model, based on Eq. 3.8, is the system load. The inverted double pendulum output, q, is the system angles θ_b and θ_p, i.e. $q = \left[\theta_b\ ,\ \theta_p\right]^T$.

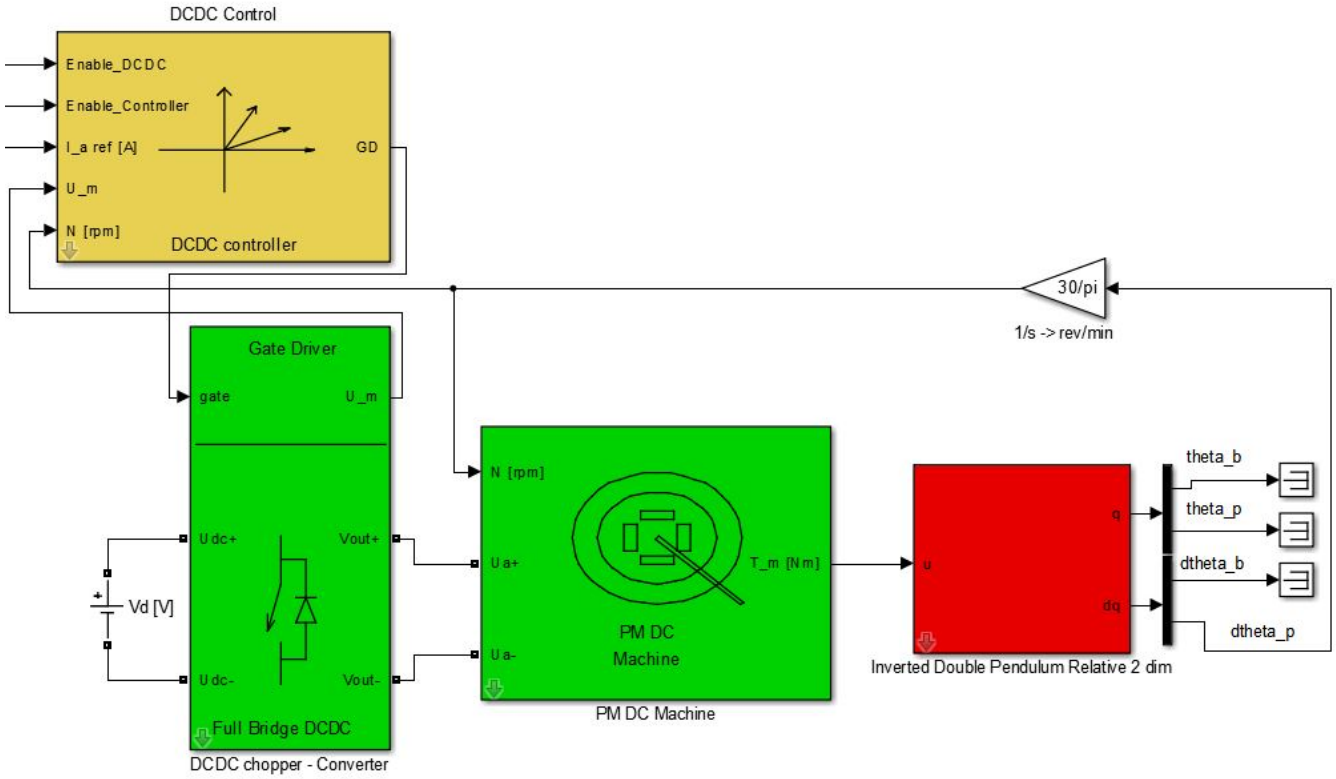


Figure 2.7: Complete system model without balancing control, from [13]

2.5.3 Motor Control and Limitation

The converter can be designed with current control or open-loop control. By designing an inner current control loop, the current reference can be calculated based on the required torque, as shown below:

$$I_{a,ref} = \frac{\tau_{ref}}{\phi_m} \tag{2.18}$$

Thus by limiting τ_{ref} the current in the motor and converter can be limited. The current controller is usually selected as a PI-Controller where the parameter values are derived på modulus optimum, see Table 9.4 row 2.3 − PI control in [3], where K_p is selected to half of the given value. The relative damping will then be $0,7$.

To improve the dynamic performance for small inertia, feed-forward of the induced voltage, $\omega \cdot \phi_m$, is usually implemented. The step response is shown in Figure 2.8, where the torque and current response is within the [ms] area. If current sensors are not available, open-loop control has to be implemented. The current reference is calculated based on the torque reference as shown above in Eq.2.18. The armature voltage can the be calculated as:

$$V_{a,ref} = R_{a,mod} \cdot I_{a,ref} + \omega \cdot \phi_{m,mod} \tag{2.19}$$

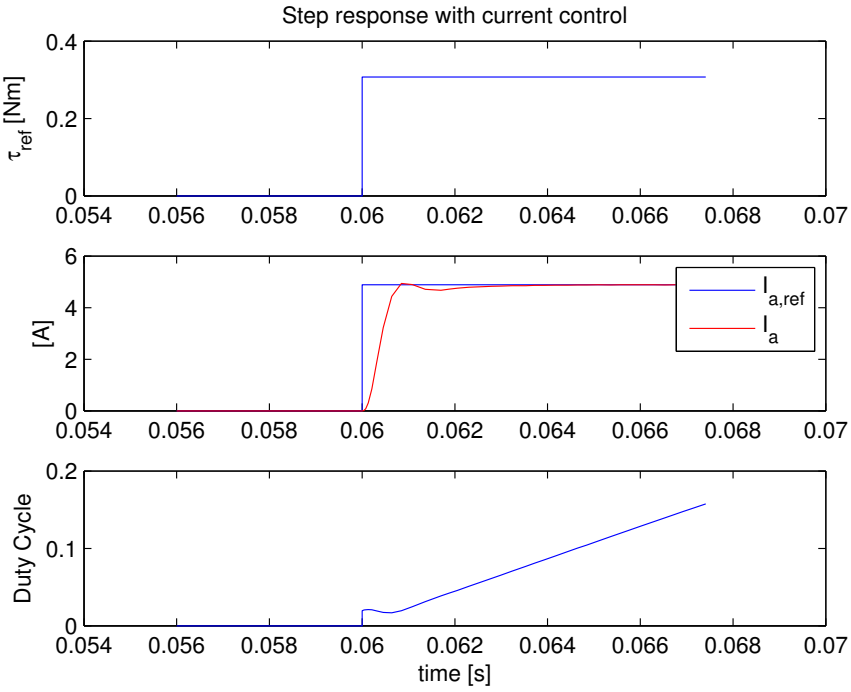


Figure 2.8: Step response of system with current controller, with model from [13]

Incorrect values of $R_{a,mod}$ and $\phi_{m,mod}$ will heavily influence the current control accuracy. Simulations with correct parameters are shown in Figure. 2.9. In this case measured speed is used in the controller. If the measured speed is not available, it is not possible to predict the induced voltage. Thereby the controller can give large currents. By rearranging Eq. 2.12, the steady state armature current of the motor can be expressed by:

$$I_a = \frac{V_a - \omega \cdot \phi_m}{R_a} \tag{2.20}$$

With a battery capacity of 24 [V] the worst case scenario would be when torque is applied with negative angular velocity, negative ω. With a nominal armature voltage, V_{an}, of 24 [V], and armature resistance, R_a, of 0.435 [Ω]the power converter has to handle 120 [A]:

$$I_{a,max} = \frac{V_{an} - (-\omega) \cdot \phi_m}{R_a} = \frac{24+24}{0,435} \approx 110\,[A] \tag{2.21}$$

Eq. 2.21 shows that the converter has to handle a current of 110 [A], which is approximately $22 \times I_{an}$. This is a current much larger than the demagnetization current of 33 [A], which will demagnetize the motor magnets. To protect the motor, a current controller is preferable.

Due to the system masses, the thermo capacity is quite large. The thermal time constants of the machine is within the range of minutes, which means that a current can be up to $2-3$ times the nominal current for periods of time shorter than this. The maximum voltage V_a is equal to the battery voltage, V_{batt}. If the $\omega \cdot \phi_m$ becomes equal to the battery voltage, the armature current I_a becomes zero. This

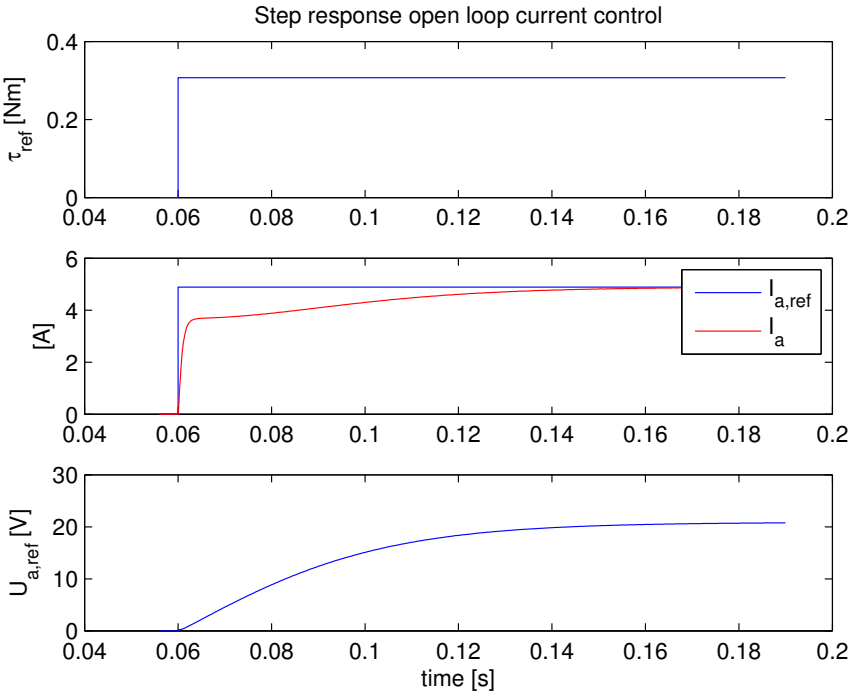


Figure 2.9: Open loop step response, with model from [13]

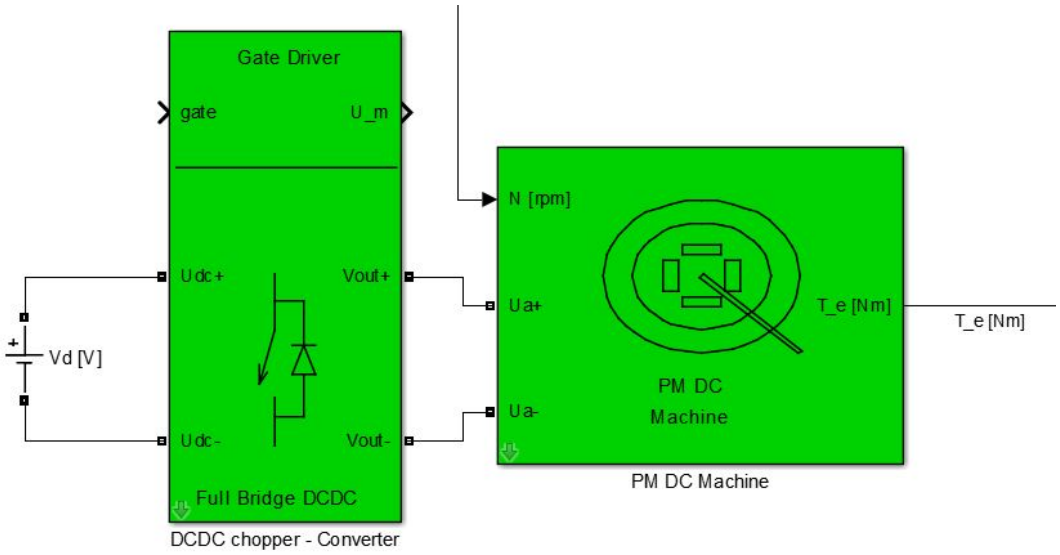


Figure 2.10: Power converter and DC motor, from [13]

means there is an upper speed limit for which the current can be controlled, and has to be taken into consideration when selecting motor and gear box for the actual bicycle system. The complete motor model with a DCDC converter is shown in Figure 2.10, where the battery voltage V_d is fed to the power converter, giving the motor voltage for the bicycle system actuator. The motor output torque, τ_m is the control torque of the bicycle system.

2.5.4 Power Converter

The converter has very short thermal time constants. This means the converter has to be rated for maximum motor current, given by the maximum allowed current in the current controller. In addition, the converter should preferably be able to braking/generator mode. If for instance the DC-link voltage of the converter is supplied by a battery, this is possible. However, if the input to the DC-link is a diode rectifier, the braking power will increase the capacitor voltage in the DC-link, then a braking chopper transistor can be used to discharge the capacitor to the correct level. If such a braking resistor is not installed, the breaking torque reference has to be limited to zero, if the capacitor voltage is too high. This will influence the motor control performance of the inverted pendulum system.

2.6 Specified Software

When the autonomous bicycle system was implemented in Matlab by Ånnestad, as in [14], the setup was made with Matlab 32-bit version. When developing the system on the physical bike, it is necessary to have the right software installed at the computer. The different type of necessary software is listed:

- Matlab 32-bits version with Windows Real-Time toolbox.
- QNX650 32-bit version.
- Windows terminal and file transfer interface, e.g. Putty

The Matlab software has to be the 32-bits version as the model uses S-Functions, programmed with simple C-code. By use of MEX ("Matlab Executable") the system is able to use the MEX S-Functions to run the system. These MEX-files are made as 32-bits versions and are therefore not compatible with the 64 bit version of Matlab.

The manual for "How to Start The Bicycle" listed in Appendix E of the thesis by Ånnestad [14] is not entirely correct. The section for checking data transmitting should be performed as following:

- Check that the MTi and analog card is transmitting data(returning a signal with noise):
  - "cat /dev/mt/orientation/roll".
  - "cat /dev/dmm32at/analog/in/ad0".

Where the "\" in [14] has been replaced with "/".

Chapter 3

System Model

In this chapter the mathematical system model is presented. The bicycle with the inverted pendulum will be represented as a limited Acrobot system. This mathematical model was derived in the authors' previous work done in [12], and is presented in Appendix A.

The Acrobot, shown in Figure 3.1, is a two-link robot manipulator with control torque applied at the second, link-connecting, joint. This system is an underactuated system as no torque is applied at the first joint, resulting in a passive joint. This underactuated degree of freedom joint results in a system with fewer control inputs than configuration variables, making the passive joint angles unable to follow arbitrary trajectories in the configuration space. For the system presented in Eq. 3.7, an underactuated system is defined by:

$$rank(\boldsymbol{F}(\boldsymbol{\theta})) < dim(\boldsymbol{\theta}) \tag{3.1}$$

The number of actuated degrees of freedom is less than the system degree of freedom, and the actuator can only create instantaneous acceleration in the active joint angles. Due to this system property, the control becomes more challenging as the bicycle tilt angle, θ_b, is a passive joint angle. The Acrobot is based on the idea of a robot representation of an acrobatic gymnastic, where the second joint correspond to the hip of the gymnastic. By swinging the second link, corresponding to the gymnastics legs, the Acrobot will increase the systems kinetic energy, creating motion in the robot manipulator. The goal is to swing up the system to the upright position and balance around the unstable equilibrium. By use of this robot manipulator, the goal is to introduce system angular limitation and present the bicycle system as a limited Acrobot.

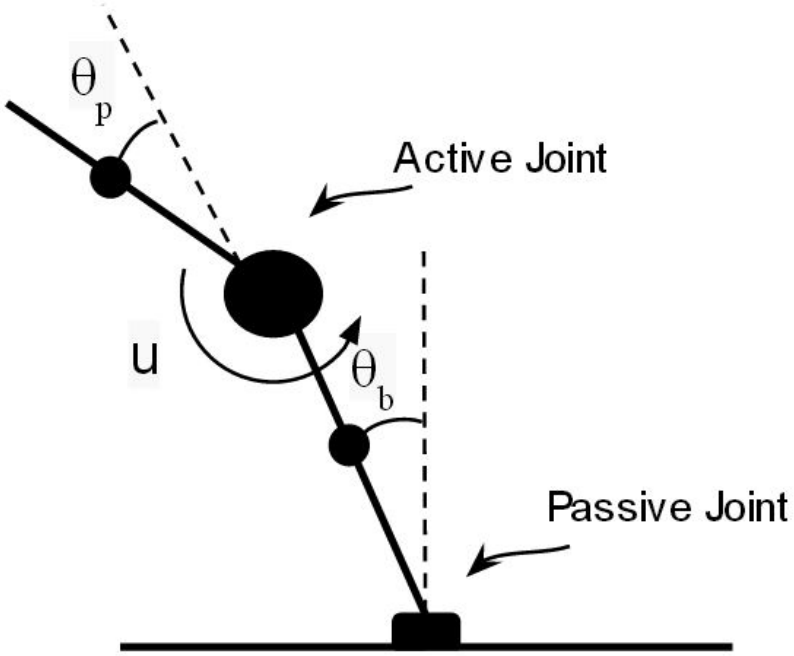


Figure 3.1: Illustration of an Acrobot

3.1 Assumptions

To derive a mathematical model of the bicycle system, some assumptions has to be made:

- Both wheels of the bicycle has contact with the ground at all time, with no slide.
- The ground is level, i.e. no ground inclination in the two dimensional workspace, such that the gravity vector is parallel with the y-axis of the reference frame (x_i, y_i) presented in Figure 3.2.
- A leaning torque from rider is applied at the hip, represented as the pendulum motor.
- Bicycle can be modeled as a manipulator link.

3.2 System Coordinates

With the system represented as an Acrobot, the system model derived in Section 3.5 is presented with relative coordinates, meaning θ_p is relative to θ_b. For equations see Appendix A.1. The angles θ_b and θ_p are corresponding to the bicycle tilt angle and the mounted inverted pendulum angle, respectively. Figure 3.2 shows the angles relative to the reference frame (x_i, y_i) and the corresponding (x, y)-coordinates of the respective angles. The number of generalized coordinates is given as the degree of freedom (DOF).

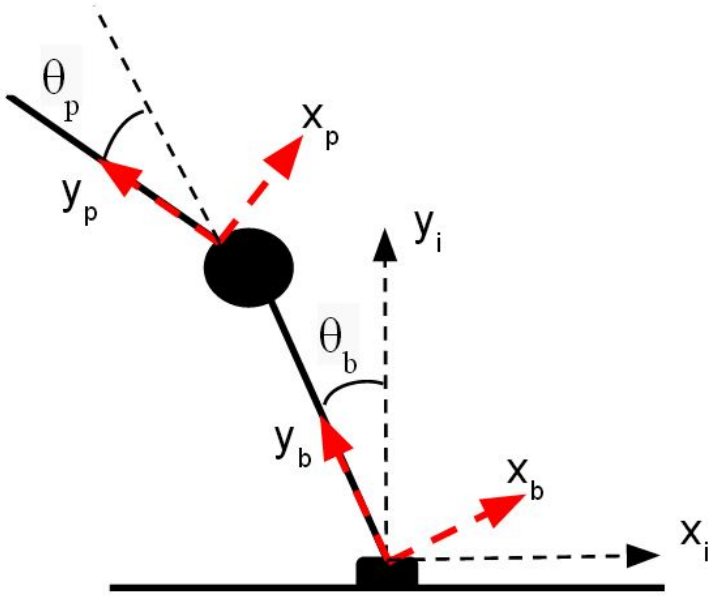


Figure 3.2: Inverted Double Pendulum Relative Coordinates

3.3 Inverted Double Pendulum Representation

The inverted double pendulum represented in Figure 3.3 illustrates the system parameters used to present the Acrobot manipulator system. The illustrated parameters are described as:

- m_b is the bicycle mass, i.e. mass of the first link.
- m_p is the pendulum mass, i.e. mass of the second link.
- r_b is the distance from the first joint to the center of mass of the first link.
- r_p is the distance from the second joint to the center of mass of the second link.
- l is the length of the first link, i.e. height of bicycle.
- θ_b and θ_p are the tilt angles of the bicycle and inverted pendulum, respectively.

Based on the geometrical shape of the links, each link of the manipulator has the moment of inertia, I_i. With n number of links, I_1, I_2,...,I_n represents the moment of inertia about the center of mass of each link. As the system in Eq. 3.8 and 3.9 is derived from energy of the point masses in the Lagranges equation of motion, Eq. 3.5, the moments of inertia are not included in the equations derived in [12].

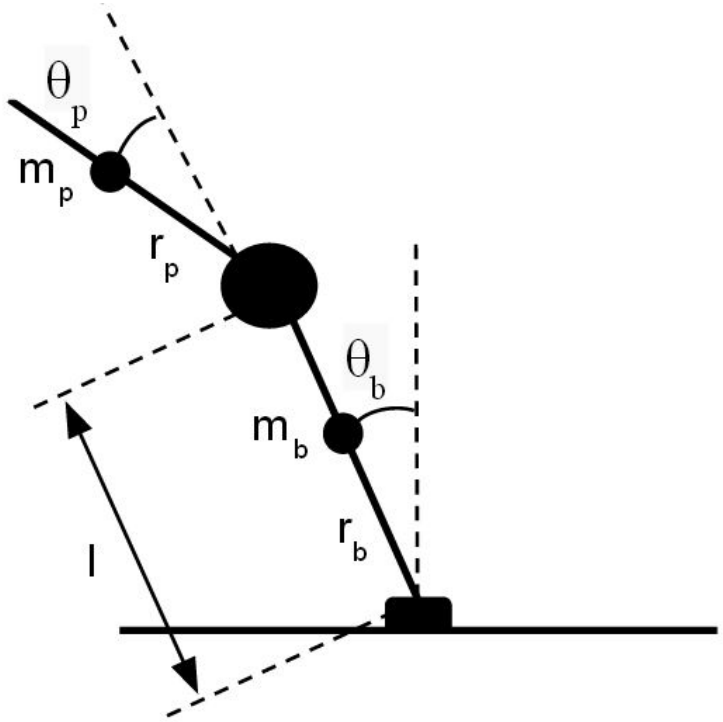


Figure 3.3: Inverted Double Pendulum Parameter Illustration

3.4 System Energy

In [12], the system equation of motion is derived by use of the Lagranges equation of motion, Eq. 3.5. By use of the equations of position and velocity of the point masses, m_b and m_p, in Appendix A.2, the system energy given in Eq. 3.2 was derived. For derivation of these equations see Appendix A.3. T_b and T_p is the kinetic energy of bicycle and pendulum and U_b and U_p is the potential energy of the bicycle and pendulum, respectively.

$$T_b = \frac{1}{2} m_b (r_b \dot{\theta}_b)^2 \tag{3.2a}$$

$$T_p = \frac{1}{2} m_p [(l\dot{\theta}_b)^2 + 2 l r_p \dot{\theta}_p (\dot{\theta}_b + \dot{\theta}_p) cos\,\theta_p + r_p^2 (\dot{\theta}_b + \dot{\theta}_p)^2] \tag{3.2b}$$

$$U_b = m_b r_b g\, cos\,\theta_b \tag{3.2c}$$

$$U_p = m_p g (l\, cos\,\theta_b + r_p\, cos(\theta_b + \theta_p) \tag{3.2d}$$

The total kinetic and potential energy is then given by Eq. 3.3.

$$\begin{aligned} T &= T_b + T_p \\ &= \frac{1}{2}(m_b r_b^2 + m_p l^2)\dot{\theta}_b^2 + m_p r_p l \dot{\theta}_b(\dot{\theta}_b + \dot{\theta}_p) cos\,\theta_p + \frac{1}{2} m_p r_p^2 (\dot{\theta}_b + \dot{\theta}_p)^2 \end{aligned} \tag{3.3a}$$

$$\begin{aligned} U &= U_b + U_p \\ &= (m_b r_b + m_p l) g\, cos\,\theta_b + m_p r_p g\, cos(\theta_b + \theta_p) \end{aligned} \tag{3.3b}$$

Note that the total kinetic energy of the rotational system can be expressed by the inertia matrix presented in Section 3.5 and $\dot{\boldsymbol{\theta}} = [\dot{\theta}_b, \dot{\theta}_p]^T$:

$$T = \frac{1}{2}\dot{\boldsymbol{\theta}}^T \boldsymbol{M}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \tag{3.4}$$

3.5 Inverted Double Pendulum System - Acrobot

The system equations of motion is derived by use of the Lagranges equation of motion, which is defined by:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = u_i \tag{3.5}$$

Where u_i represent the external torque, τ, applied to the rotational system. L is the Lagrangian function defined with T and U as the kinetic and potential energy, respectively.:

$$L = T - U \tag{3.6}$$

By use of the Lagranges equation of motion, Eq. 3.5, and the system energy given in Eq. 3.3, the system model for an Acrobot was derived in the authors own work in [12], and is shown in Appendix A.4. The underactuated manipulator with relative state coordinates $\boldsymbol{\theta} = [\theta_b, \theta_p]^T$ is given as:

$$\boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \boldsymbol{G}(\boldsymbol{\theta}) = \boldsymbol{F}(\boldsymbol{\theta})\tau \tag{3.7}$$

The matrix $\boldsymbol{M}(\boldsymbol{\theta})$ is the system inertia matrix and $\boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ is the Coriolis and centrifugal forces. The $\boldsymbol{G}(\boldsymbol{\theta})$-matrix is the force of gravity and $\boldsymbol{F}(\boldsymbol{\theta})$ represent the system actuators. This 2 DOM system can be presentes as:

$$\begin{bmatrix} m_{11} & m_{12} \\ m_{12} & m_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_b \\ \ddot{\theta}_p \end{bmatrix} + \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_b \\ \dot{\theta}_p \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \tau \tag{3.8}$$

Where:

$$\boldsymbol{M}(\boldsymbol{\theta}) = \begin{bmatrix} m_b r_b^2 + m_p l^2 + m_p r_p^2 + 2 m_p r_p l \cos\theta_p & m_p r_p^2 + m_p r_p l \cos\theta_p \\ m_p r_p^2 + m_p r_p l \cos\theta_p & m_p r_p^2 \end{bmatrix} \tag{3.9a}$$

$$\boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \begin{bmatrix} -2 m_p r_p l \dot{\theta}_p \sin\theta_p & -m_p r_p l \dot{\theta}_p \sin\theta_p \\ m_p r_p l \dot{\theta}_b \sin\theta_p & 0 \end{bmatrix} \tag{3.9b}$$

$$\boldsymbol{G}(\boldsymbol{\theta}) = \begin{bmatrix} -(m_b r_b + m_p l) g \sin\theta_b - m_p r_p g \sin(\theta_b + \theta_p) \\ -m_p r_p g \sin(\theta_b + \theta_p) \end{bmatrix} \tag{3.9c}$$

$$\boldsymbol{F} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3.9d}$$

The rewritten state space representation is given by:

$$\ddot{\boldsymbol{\theta}} = \boldsymbol{M}^{-1}(\boldsymbol{\theta}) \left[-\boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \dot{\boldsymbol{\theta}} - \boldsymbol{G}(\boldsymbol{\theta}) + \boldsymbol{F}(\boldsymbol{\theta}) u \right] \tag{3.10}$$

Note that the moment of inertia of the manipulator links, mentioned in Section 3.3, are neglected. As the system is derived from the Lagranges equation of motion, Eq. 3.5, the inertia matrix, $\boldsymbol{M}(\boldsymbol{\theta})$, shown in Eq. 3.9 only consist of the inertia of the point masses m_b and m_p. As described in Section 3.3 the moment of inertia is dependent of the geometrical shape of the manipulator links. When the moment of inertia is taken into account and included into the system model, all the system matrices except from the inertia matrix, $\boldsymbol{M}(\boldsymbol{\theta})$, remains unchanged. Eq. 3.11 shows the resulting inertia matrix where I_b and I_p are denoted as the moment of inertia about the center of mass of the bicycle and the mounted inverted pendulum, respectively.

$$\boldsymbol{M}(\boldsymbol{\theta}) = \begin{bmatrix} I_b + I_p + m_b r_b^2 + m_p l^2 + m_p r_p^2 + 2 m_p r_p l \cos\theta_p & I_p + m_p r_p^2 + m_p r_p l \cos\theta_p \\ I_p + m_p r_p^2 + m_p r_p l \cos\theta_p & I_p + m_p r_p^2 \end{bmatrix} \tag{3.11}$$

Note that I_b and $m_b r_b^2$ are only included in m_{11} of the $\boldsymbol{M}(\boldsymbol{\theta})$-matrix in Eq. 3.8, thus the bicycle inertia is only multiplied with the bicycle tilt angles acceleration, $\ddot{\theta}_b$, to get the bicycle torque. The matrix in in Eq. 3.11 is verified with the system model of an Acrobot derived by Spong in [18].

In this system, the manipulator joints are the DC motor and the connecting point between the ground and tires, respectively. The friction between the ground and tires are not significant in the rotational axis. With the frictionless DC motor friction in both system joints are neglected, resulting in a system model without joint friction and energy dissipation.

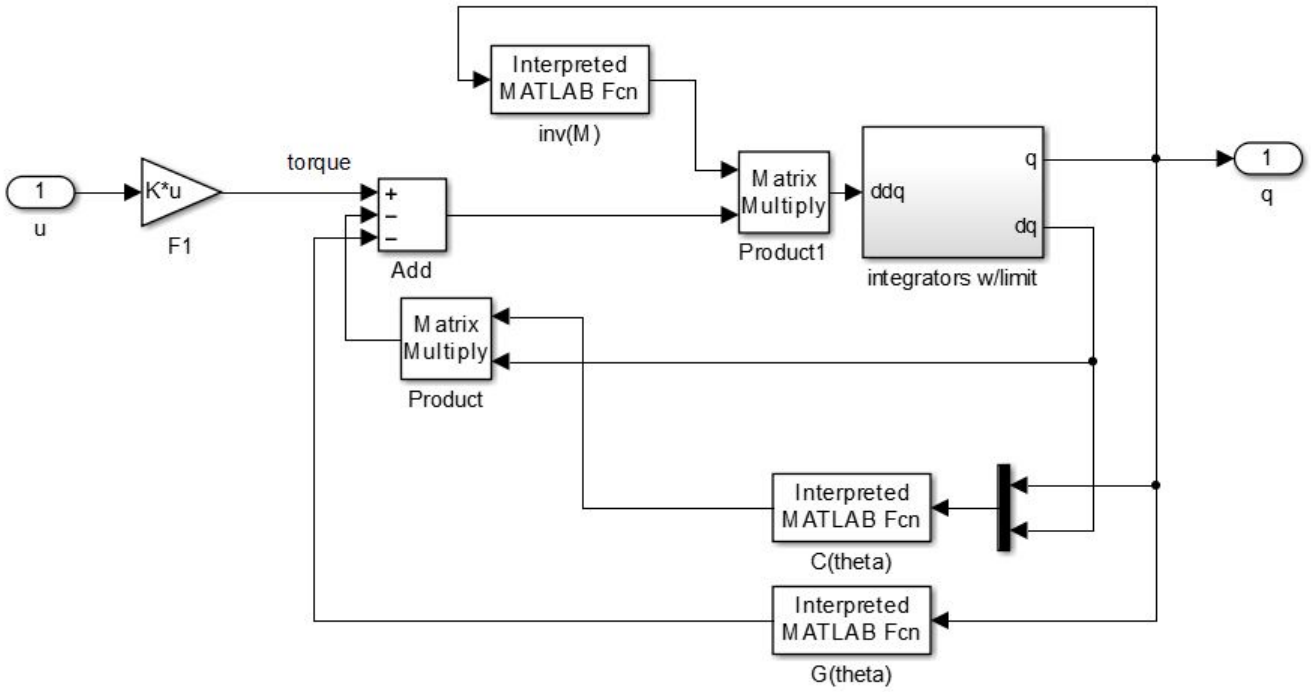


Figure 3.4: System model in Simulink with limitation, from [12]

3.6 Model Angular Limitations

To complete the system model of the bicycle system, angular limitation has to be included. Figure 3.4 shows Eq. 3.10 implemented in Mathworks Simulink®, where the block "*integrators w/limit*" is the integrators with logical angular limitation. This is the most important detail that separates the bicycle system from a regular Acrobot. The model in Eq. 3.8 derived from the Lagranges equation of motion, Eq. 3.5, is based on an Acrobot with no state limitations, i.e. the angles of θ_b and θ_p can utilize the whole workspace of the manipulator construction. By including the physical system limitation described in Section 2.1, the mathematical model of the bicycle system is complete. Thereby the angular limitation gives the possibility of presenting the system as an Acrobot within these limits, to further investigate various controller strategies for system stabilization.

In [12], the author designed integrators with logical limitation, as shown in Figure 3.5, where one limited integrator design is applied to the bicycle tilt angle acceleration, $\ddot{\theta}_b$, and another to the inverted pendulum acceleration, $\ddot{\theta}_p$. This design is inside the sub block "integrators w/limit" shown in Figure 3.4. The second cascade integrator, shown in Figure 3.5, gives the system angles and a signal which is set to 1 if the angles is at the limit. These limits are either upper or lower angular limits. This can not be detected by the integrator limit signal, thus the hysteresis control is utilized to detect whether the link positions are at the upper or lower limit. If the system angles are at the lower limit, the LL hysteresis gives a signal of 1 while the UL hysteresis gives a signal of 0, and vice versa. The logical

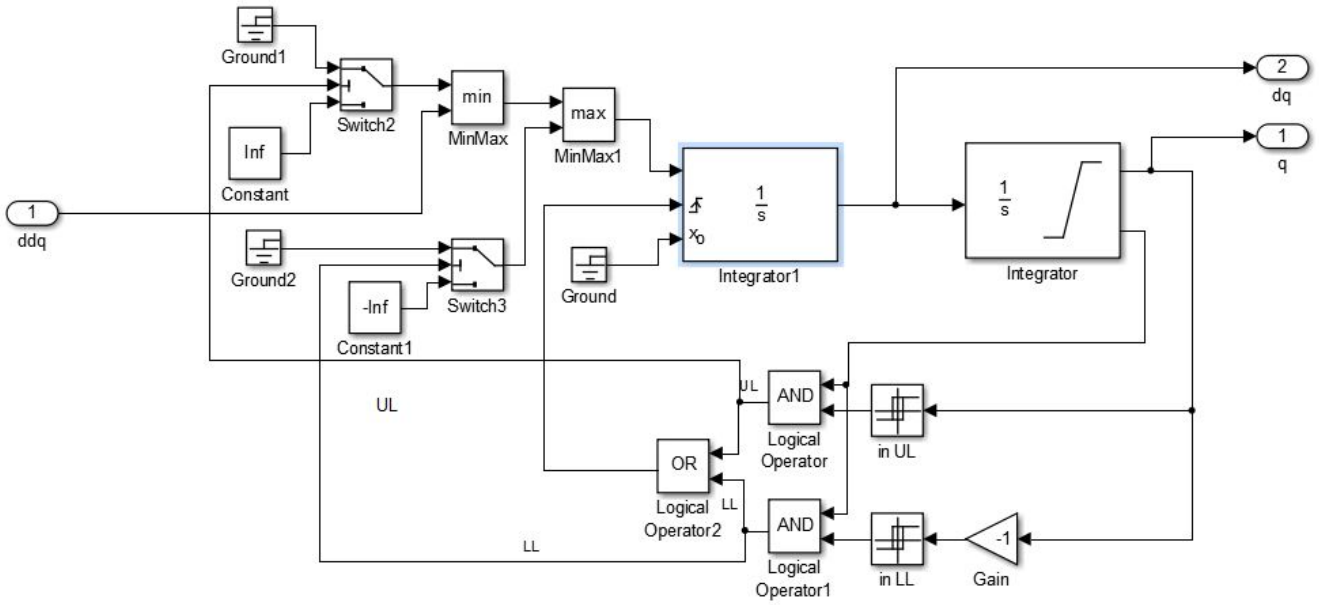


Figure 3.5: Integrators with limitation, from [12]

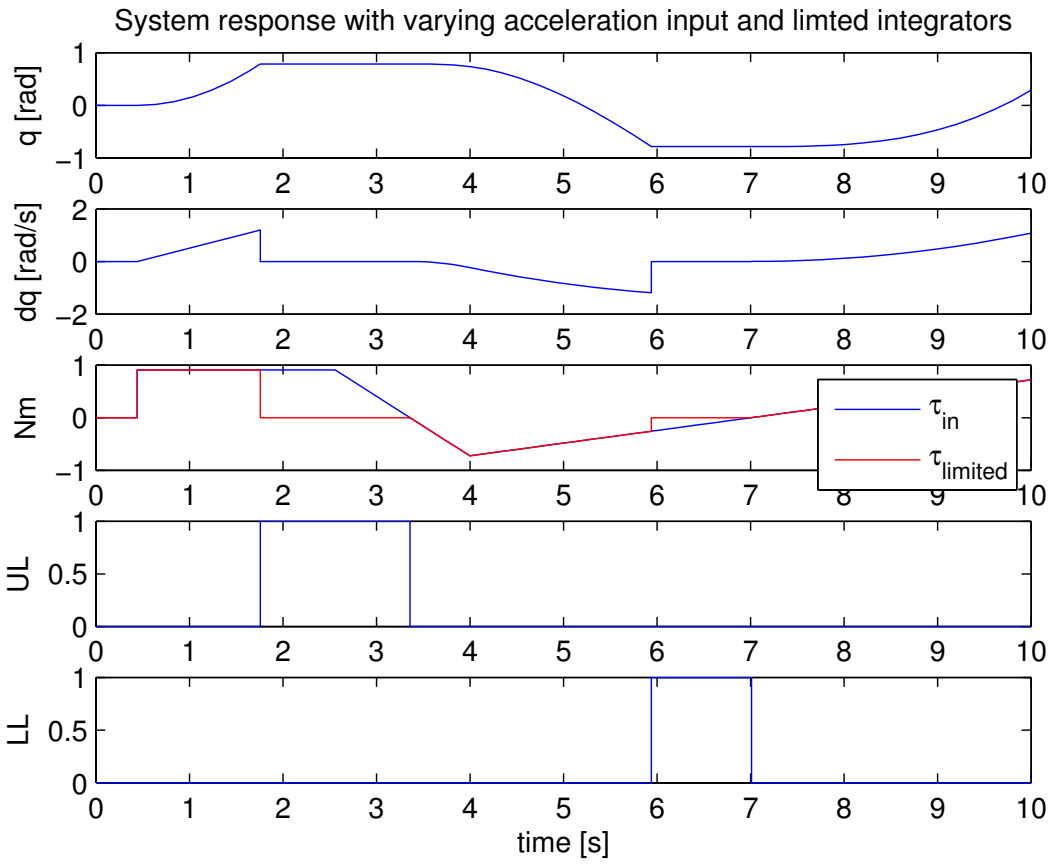


Figure 3.6: Plot of integrator performance with limitation

"AND" operators gives a positive signal, of 1, when the integrator detects system limitation and the hysteresis control gives an output of 1. The UL control set to switch on 0.01 [rad] before the upper limit, and switch off 0.02 [rad] under this limit. The LL control works with the opposite signals, thereby multiplied with -1. These logical outputs has two functionalities. First, the "OR" block is used to reset the first integrator, setting the angular acceleration to zero (ground), preventing torque appliance and system wear and tear. Second, the limits are used to prevent the system actuator to apply torque in the direction of system limitation. The only torque allowed is the torque moving the system away from the limit, e.g. at lower limit, positive torque is the only system actuation allowed. At upper limit, negative torque is the only actuation allowed. Figure 3.6 shows the described functionality of the integrators with limitation, where a system acceleration is applied to activate both upper and lower limit detection. $\tau_{limited}$ shows the torque applied due to the system limitation handled by the integrator structure in Figure 3.5.

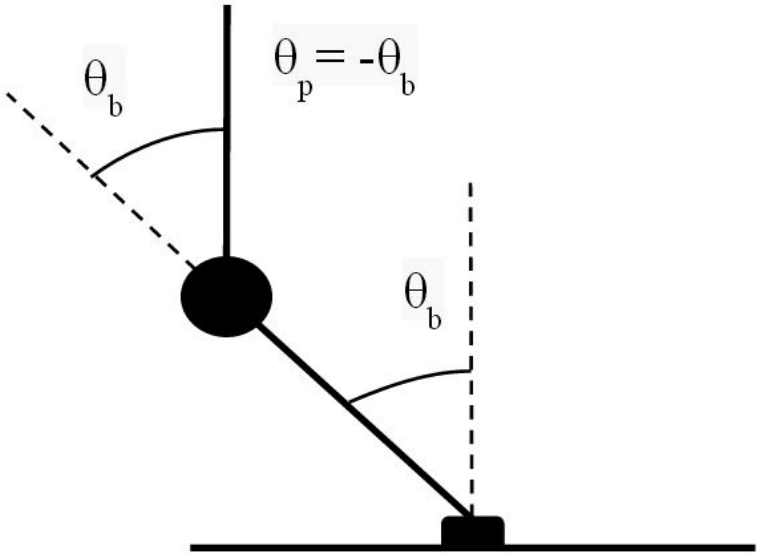


Figure 3.7: Stability of second link

3.7 System Stability

When balancing the bicycle system, one has to balance around the upright unstable equilibrium of the nonlinear dynamical system. The underactuated system is a highly unstable nonlinear system, which implies that small perturbations will result in the states move away from their unstable equilibriums. Assuming a fixed bicycle tilt angle, θ_b, the mounted inverted pendulum will have an unstable equilibrium at $\theta_p = -\theta_b$, see Figure 3.7. The bicycle link has infinite possible angles when $\theta_b \in [-\pi, \pi)$, thus the second link will have infinite amount of equilibrium points as it has two equilibrium points, $(\pi - \theta_b, \theta_b)$, for each angle of θ_b, when $\theta_b \in [-\pi, \pi)$. When the system is balancing around the upright equilibrium, the time derivatives of the system angles and control torque, u, are set to zero:

$$\dot{\theta}_b \equiv \ddot{\theta}_b \equiv \dot{\theta}_p \equiv \ddot{\theta}_p \equiv 0 \tag{3.12a}$$

$$u = 0 \tag{3.12b}$$

From the system equations in Eq. 3.9, the system equilibrium points are given by:

$$\boldsymbol{G}(\boldsymbol{\theta}) = \begin{bmatrix} -(m_b r_b + m_p l) g \, sin\theta_b - m_p r_p g \, sin(\theta_b + \theta_p) \\ -m_p r_p g \, sin(\theta_b + \theta_p) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.13}$$

From Eq. 3.13 and the physical angular limitation presented in Section 2.1 and 3.6 the trivial solution is:

$$\theta_b \equiv \theta_p \equiv 0 \tag{3.14}$$

The angles of θ_b and θ_p given in Eq. 2.1 correspond to the upright unstable equilibrium of the uncontrolled bicycle system.

3.8 State Space Model

In [12] the model presented in Eq. 3.8 was presented with change of variables to reduce the number of system integrators. With $\boldsymbol{\theta}$ and $\dot{\boldsymbol{\theta}}$ defined as:

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_b \\ \theta_p \end{bmatrix}$$

$$\dot{\boldsymbol{\theta}} = \begin{bmatrix} \dot{\theta}_b \\ \dot{\theta}_p \end{bmatrix}$$

The state vector $\boldsymbol{x}$ is introduced, and defined as:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{\theta} \\ \dot{\boldsymbol{\theta}} \end{bmatrix} \tag{3.16}$$

With the change of variables, the state equation in Eq. 3.8 becomes a four dimensional system shown in Eq. 3.17:

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}(\boldsymbol{\theta}) \end{bmatrix} \dot{\boldsymbol{x}} + \begin{bmatrix} \boldsymbol{0} & -\boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{G}(\boldsymbol{\theta}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{F}(\boldsymbol{\theta}) \end{bmatrix} u \tag{3.17}$$

The corresponding rewritten state space representation, is written as:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, u) = \begin{bmatrix} \dot{\boldsymbol{\theta}} \\ \boldsymbol{M}^{-1}(\boldsymbol{\theta}) \left[\boldsymbol{F} u - \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \dot{\boldsymbol{\theta}} - \boldsymbol{G}(\boldsymbol{\theta}) \right] \end{bmatrix} \tag{3.18}$$

where $\boldsymbol{f}(\boldsymbol{x}, u) = \left[f_1(\boldsymbol{x}, u),\ f_2(\boldsymbol{x}, u),\ f_3(\boldsymbol{x}, u),\ f_4(\boldsymbol{x}, u) \right]^T$.

3.8.1 State Space Differential Equations

By multiplying the system matrices in the equations of motion in Eq 3.8, the system can be presented as two differential equations. To increase the readability, the denominator of the inverse matrix $\boldsymbol{M}^{-1}(\boldsymbol{\theta})$ is introduced as an inertial term defined as:

$$J_{tot} = m_b r_b^2 + m_p l^2 sin^2\theta_p \tag{3.19}$$

By use of the notation where $\dot{\theta} = \Omega$ and $\ddot{\theta} = \frac{d\Omega}{dt}$, the system equations of motion is given as:

$$\begin{aligned} J_{tot}\frac{d\Omega_b}{dt} = {} & m_p r_p l sin\,\theta_p \left((\Omega_b+\Omega_p)^2 + \frac{l}{r_p} cos\,\theta_p\, \Omega_b^2\right) \\ & + g m_p l \left(\left(1+\frac{m_b r_b}{m_p l}\right) sin\,\theta_b - cos\,\theta_p\, sin(\theta_b+\theta_p)\right) \\ & - \left(1+\frac{l}{r_p} cos\,\theta_p\right)\tau \end{aligned} \tag{3.20a}$$

$$\begin{aligned} J_{tot}\frac{d\Omega_p}{dt} = {} & - m_p r_p l sin\,\theta_p \left((\Omega_b+\Omega_p)^2 + \left(\frac{m_b r_b^2}{m_p r_p^2} + \left(\frac{l}{r_p}\right)^2\right)\Omega_b^2 + \frac{l}{r_p} cos\,\theta_p (\Omega_b^2 + (\Omega_b+\Omega_p)^2)\right) \\ & - g m_p l \left(\left(1+\frac{m_b r_b}{m_p l}\right)\left(1+\frac{l}{r_p} cos\,\theta_p\right) sin\,\theta_b - \left(\frac{m_b r_b^2}{m_p r_p l} + \frac{l}{r_p} + cos\,\theta_p\right) sin(\theta_b+\theta_p)\right) \\ & + \left(1+\frac{m_b r_b^2}{m_p r_p^2} + \left(\frac{l}{r_p}\right)^2 + 2\frac{l}{r_p} cos\,\theta_p\right)\tau \end{aligned} \tag{3.20b}$$

With these differential equations the simplicity of systematic analysis is increased, as the equations are sorted with the terms representing Coriolis and centrifugal torques, gravity related torques and actuator torque, respectively. This derivation has been verified with Wolfram Mathematica ©.

3.8.2 Partially Feedback Linearized State Space Representation

As the bicycle system is an underactuated system the nonlinearities can not be completely canceled by feedback linearization. However, when θ_p can gain instantaneous acceleration in arbitrary direction from the control torque, partial feedback linearization can be applied to the system. This feedback cancel out the nonlinear terms of the mounted inverted pendulum, simplifying the nonlinear control problem. With the definition $u = \ddot{\theta}_p = \frac{d\Omega_p}{dt}$, the feedback control torque is given as:

$$\begin{aligned}\tau = & \frac{1}{1+\frac{m_b r_b^2}{m_p r_p^2}+\left(\frac{l}{r_p}\right)^2+2\frac{l}{r_p}cos\,\theta_p}\cdot \\ & \left[J_{tot}u + g m_p l\left(\left(1+\frac{m_b r_b}{m_p l}\right)\left(1+\frac{l}{r_p}cos\,\theta_p\right)sin\,\theta_b - \left(\frac{m_b r_b^2}{m_p r_p l}+\frac{l}{r_p}+cos\,\theta_p\right)sin(\theta_b+\theta_p)\right)\right. \\ & \left. + r_p m_p l\, sin\,\theta_p\left((\Omega_b+\Omega_p)^2+\left(\frac{m_b r_b^2}{m_p r_p^2}+\left(\frac{l}{r_p}\right)^2\right)\Omega_b^2+\frac{l}{r_p}\,cos\,\theta_p\left(\Omega_b^2+(\Omega_b+\Omega_p)^2\right)\right)\right]\end{aligned} \tag{3.21}$$

With this partially feedback linearization, the state space representation defined in Eq. 3.20 becomes simplified as the differential equation for the bicycle tilt angle, Eq. 3.20a, has the torque, τ, as given in Eq. 3.21. Thus, by use of Eq. 3.8, the set of differential equations for the bicycle system becomes:

$$\frac{d\theta_b}{dt} = \Omega_b \tag{3.22a}$$

$$\frac{d\theta_p}{dt} = \Omega_p \tag{3.22b}$$

$$\frac{d\Omega_b}{dt} = \frac{-c_{11}\Omega_b - c_{12}\Omega_p - g_1 - m_{12}u}{m_{11}} \tag{3.22c}$$

$$\frac{d\Omega_p}{dt} = u \tag{3.22d}$$

where the full expression for $\ddot{\theta}_b \equiv \frac{d\Omega_b}{dt}$ is given as:

$$\frac{d\Omega_b}{dt} = \frac{-(m_p r_p^2 + m_p r_p l cos\theta_p)u + g(m_p l + m_b r_b)sin\theta_b + m_p r_p\left(2l\Omega_b\Omega_p sin\theta_p + l\Omega_p^2 sin\theta_p + g sin(\theta_b+\theta_p)\right)}{m_p l^2 + m_b r_b^2 + m_p r_p^2 + 2m_p r_p l\, cos\,\theta_p}$$

Thus the state space representation of the Acrobot system is given by the set of differential equations in Eq. 3.22, where the nonlinear representation is defined as:

$$\boldsymbol{f}(\boldsymbol{x},u)=\begin{bmatrix} f_1(\boldsymbol{x},u)\\ f_2(\boldsymbol{x},u)\\ f_3(\boldsymbol{x},u)\\ f_4(\boldsymbol{x},u)\end{bmatrix}=\begin{bmatrix} \frac{d\theta_b}{dt}\\ \frac{d\theta_p}{dt}\\ \frac{d\Omega_b}{dt}\\ \frac{d\Omega_p}{dt}\end{bmatrix} \tag{3.23}$$

where $\boldsymbol{x}=(\theta_b,\,\theta_p,\,\Omega_b,\,\Omega_p)^T$.

3.9 System Linearization

When investigating the nonlinear model, system linearization can be applied. By linearizing the nonlinear model, one can analyze the stability properties of the system around the equilibrium points. The eigenvalues of the linearized system can be analyzed to show the system properties and see the system response when small perturbations occur around the equilibrium points of the nonlinear system. With a linearized model, linear controllers can be designed, as the LQR controller utilized in the bicycle stabilization presented in Chapter 4. The linear time-invariant (LTI) system with n states is presented with input, u, states, x, and output, y:

$$\dot{\boldsymbol{x}}(t)=\boldsymbol{A}\boldsymbol{x}(t)+\boldsymbol{B}\boldsymbol{u}(t)\ ,\quad \boldsymbol{A}\in\mathbb{R}^{n\times n} \tag{3.24a}$$

$$\boldsymbol{y}(t)=\boldsymbol{C}\boldsymbol{x}(t)+\boldsymbol{D}\boldsymbol{u}(t) \tag{3.24b}$$

3.9.1 Linearization by Taylor Expansion

In [12] a linear model linearized around the upright unstable equilibrium $(\theta_b,\theta_p)=(0,0)$ was derived. With change of variables, as presented in Eq. 3.18, the bicycle system was linearized by Taylor expansion around the fixed point $(\boldsymbol{x}^*,u^*)$:

$$\dot{\boldsymbol{x}}=\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})\approx\boldsymbol{f}(\boldsymbol{x}^*,\boldsymbol{u}^*)+\left[\frac{\partial\boldsymbol{f}}{\partial\boldsymbol{x}}\right]_{\boldsymbol{x}=\boldsymbol{x}^*,\,\boldsymbol{u}=\boldsymbol{u}^*}(\boldsymbol{x}-\boldsymbol{x}^*)+\left[\frac{\partial\boldsymbol{f}}{\partial\boldsymbol{u}}\right]_{\boldsymbol{x}=\boldsymbol{x}^*,\,\boldsymbol{u}=\boldsymbol{u}^*}(\boldsymbol{u}-\boldsymbol{u}^*) \tag{3.25}$$

At the equilibrium of $\boldsymbol{x}^* = (\theta_b^*\ ,\ \theta_p^*\ ,\ \dot{\theta}_b^*\ ,\ \dot{\theta}_p^*)^T = (0,0,0,0)^T$, with control torque $u^* = \tau^* = 0$, the state derivatives, $\dot{\boldsymbol{x}}$, of Eq. 3.18 becomes:

$$\boldsymbol{f}(\boldsymbol{x}^*, u^*) = \boldsymbol{0}$$

From the Taylor expansion the linear state-space representation becomes:

$$\dot{\boldsymbol{x}} \approx \boldsymbol{A}(\boldsymbol{x} - \boldsymbol{x}^*) + \boldsymbol{B}(u - u^*)$$

Note that $\boldsymbol{x}^* = u^* = 0$, thus the state representation $(\boldsymbol{x} - \boldsymbol{x}^*)$ and $(u - u^*)$ equals $\boldsymbol{x}$ and u. In [12] the system matrices were derived as [1]:

$$\begin{aligned}\boldsymbol{A} = \left[\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right]_{\boldsymbol{x}=\boldsymbol{x}^*,\ \boldsymbol{u}=\boldsymbol{u}^*} &= \begin{bmatrix} \frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{f}_1}{\partial \dot{\boldsymbol{\theta}}} \\ \frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{f}_2}{\partial \dot{\boldsymbol{\theta}}} \end{bmatrix}_{\boldsymbol{x}=\boldsymbol{x}^*,\ u=u^*} \\ &= \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{M}^{-1}\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{\theta}} & -\boldsymbol{M}^{-1}\boldsymbol{C} \end{bmatrix}_{\boldsymbol{x}=\boldsymbol{x}^*,\ u=u^*}\end{aligned}$$

$$\begin{aligned}\boldsymbol{B} = \left[\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right]_{\boldsymbol{x}=\boldsymbol{x}^*,\ \boldsymbol{u}=\boldsymbol{u}^*} &= \begin{bmatrix} \frac{\partial \boldsymbol{f}_1}{\partial u} \\ \frac{\partial \boldsymbol{f}_2}{\partial u} \end{bmatrix}_{\boldsymbol{x}=\boldsymbol{x}^*,\ u=u^*} \\ &= \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{M}^{-1}\boldsymbol{F} \end{bmatrix}_{\boldsymbol{x}=\boldsymbol{x}^*,\ u=u^*}\end{aligned}$$

Where $\boldsymbol{G} = [g_1\ ,\ g_2]^T$ and:

$$\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial \boldsymbol{G}(\boldsymbol{\theta})}{\partial \theta_b} & \frac{\partial \boldsymbol{G}(\boldsymbol{\theta})}{\partial \theta_p} \end{bmatrix} = \begin{bmatrix} \frac{\partial g_1}{\partial \theta_b} & \frac{\partial g_1}{\partial \theta_p} \\ \frac{\partial g_2}{\partial \theta_b} & \frac{\partial g_2}{\partial \theta_p} \end{bmatrix}$$

The system matrices, given in Eq. 3.9, linearized around the upright equilibrium are:

$$\boldsymbol{M}(\boldsymbol{\theta})\big|_{\boldsymbol{\theta}=0} = \begin{bmatrix} m_b r_b^2 + m_p l^2 + m_p r_p^2 + 2 m_p r_p l & m_p r_p^2 + m_p r_p l \\ m_p r_p^2 + m_p r_p l & m_p r_p^2 \end{bmatrix} \tag{3.27a}$$

$$\boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\big|_{\boldsymbol{\theta}=\dot{\boldsymbol{\theta}}=0} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{3.27b}$$

$$\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}=0} = \begin{bmatrix} -g(m_b r_b + m_p l + m_p r_p) & -m_p r_p g \\ -m_p r_p g & -m_p r_p g \end{bmatrix} \tag{3.27c}$$

[1] Note that [12] has a sign error in the $\boldsymbol{B}$-matrix, which is corrected

The matrices in Eq. 3.28 are the system matrices in Eq. 3.26, derived symbolically with Maplesoft ©.

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{g}{r_b} & -\frac{m_p l g}{m_b r_b^2} & 0 & 0 \\ -\frac{g(r_p - r_b + l)}{r_b r_p} & -\frac{g(m_b r_b^2 + m_p l^2 + m_p r_p l)}{m_b r_p r_b^2} & 0 & 0 \end{bmatrix} \tag{3.28a}$$

$$\boldsymbol{B} = \begin{bmatrix} 0 \\ 0 \\ -\frac{l + r_p}{m_b r_p r_b^2} \\ \frac{m_b r_b^2 + m_p l^2 + m_p r_p^2 + 2 m_p l r_p}{m_b m_p r_b^2 r_p^2} \end{bmatrix} \tag{3.28b}$$

Linear System Eigenvalues

With system parameters presented in Table 2.1 the eigenvalues of the linearized system is found by use of the Matlab® function:

$$\boldsymbol{\lambda} = eig(A) = \begin{bmatrix} -9.4193 \\ -4.1809 \\ 9.4193 \\ 4.1809 \end{bmatrix} \tag{3.29}$$

As the system has two eigenvalues where $Re(\lambda_i) > 0$, the system is unstable and is a proof of why a system controller has to be developed.

3.9.2 Partially Feedback Linearized System Linearization

As a result of the partially feedback linearization presented in Section 3.8.2, the nonlinear system representation is given with $\ddot{\theta}_p = u$. By use of a feedback controller u, the systems nonlinear terms are simplified and the linearized representation of Eq. 3.22 can be presented as:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{\partial f_3(\boldsymbol{x})}{\partial \theta_b} & \frac{\partial f_3(\boldsymbol{x})}{\partial \theta_p} & \frac{\partial f_3(\boldsymbol{x})}{\partial \Omega_b} & \frac{\partial f_3(\boldsymbol{x})}{\partial \Omega_p} \\ 0 & 0 & 0 & 0 \end{bmatrix}_{\boldsymbol{x}=\boldsymbol{x}^*,\, u=u^*} \tag{3.30a}$$

$$\boldsymbol{B} = \begin{bmatrix} 0 \\ 0 \\ \frac{\partial f_3(\boldsymbol{x})}{\partial u} \\ 1 \end{bmatrix}_{\boldsymbol{x}=\boldsymbol{x}^*,\, u=u^*} \tag{3.30b}$$

where the partial derivatives are:

$$\frac{\partial f_3(\boldsymbol{x})}{\partial \theta_b} = \frac{g\left((m_p l + m_b r_b)\, cos\,\theta_b + m_p r_p\, cos(\theta_b + \theta_p)\right)}{m_b r_b^2 + m_p r_p^2 + m_p l^2 + 2 m_p r_p l\, cos\,\theta_p}$$

$$\begin{aligned}\frac{\partial f_3(\boldsymbol{x})}{\partial \theta_p} = & \frac{m_p r_p \left[\left(l\Omega_p\left(2\Omega_b + \Omega_p\right) + g\, cos\,\theta_b\right)\left(2 m_p r_p l + \left(m_b r_b^2 + m_p r_p^2 + m_p l^2\right)\, cos\,\theta_p\right)\right]}{\left(m_b r_b^2 + m_p r_p^2 + m_p l^2 + 2 m_p r_p l\, cos\,\theta_p\right)^2} \\ & + \frac{m_p r_p \left[l\left(m_b r_b^2 - m_p r_p^2 + m_p l^2\right) u - g\left(-m_p l^2 - 2 m_b r_b l + m_b r_b^2 + m_p r_p^2\right)\, sin\,\theta_b\right]\, sin\,\theta_p}{\left(m_b r_b^2 + m_p r_p^2 + m_p l^2 + 2 m_p r_p l\, cos\,\theta_p\right)^2}\end{aligned}$$

$$\frac{\partial f_3(\boldsymbol{x})}{\partial \Omega_b} = \frac{2 m_p r_p l \Omega_p\, sin\,\theta_p}{m_b r_b^2 + m_p r_p^2 + m_p l^2 + 2 m_p r_p l\, cos\,\theta_p}$$

$$\frac{\partial f_3(\boldsymbol{x})}{\partial \Omega_p} = \frac{2 m_p r_p l (\Omega_b + \Omega_p) sin\,\theta_p}{m_b r_b^2 + m_p r_p^2 + m_p l^2 + 2 m_p r_p l\, cos\,\theta_p}$$

$$\frac{\partial f_3(\boldsymbol{x})}{\partial u} = -\frac{m_{12}}{m_{11}} = -\frac{m_p r_p (r_p + l\, cos\,\theta_p)}{m_b r_b^2 + m_p r_p^2 + m_p l^2 + 2 m_p r_p l\, cos\,\theta_p}$$

With these expressions for the linearized state space representation, the system can be linearized around a desirable operating point. The system states can be set, and the linearized state space representation around the upright equilibrium is given by Eq. 3.32, with $\boldsymbol{x}^* = (\theta_b^*\ ,\ \theta_p^*\ ,\ \Omega_b^*\ ,\ \Omega_p^*)^T = (0,0,0,0)^T$ and $u^* = 0$.

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{g(m_b r_b + m_p r_p + m_p l)}{m_b r_b^2 + m_p r_p^2 + m_p l^2 + 2 m_p r_p l} & \frac{m_p r_p g}{m_b r_b^2 + m_p r_p^2 + m_p l^2 + 2 m_p r_p l} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.32a}$$

$$\boldsymbol{B} = \begin{bmatrix} 0 \\ 0 \\ -\frac{m_p r_p^2 + m_p r_p l}{m_b r_b^2 + m_p r_p^2 + m_p l^2 + 2 m_p r_p l} \\ 1 \end{bmatrix} \tag{3.32b}$$

where the property of $cos\,\theta \approx 1$ and $sin\,\theta \approx \theta$ for $\theta \approx 0$, is utilized.

Linear System Eigenvalues

With system parameters presented in Table 2.1 the eigenvalues of the linearized system is found by use of the Matlab® function:

$$\boldsymbol{\lambda} = eig(A) = \begin{bmatrix} 4.2610 \\ -4.2610 \\ 0 \\ 0 \end{bmatrix} \tag{3.33}$$

As the system has one eigenvalue where $Re(\lambda_i) > 0$, the system is unstable and a controller for system stabilization must be implemented.

3.9.3 System Controllability

With the state-space matrix $A \in \mathbb{R}^{n\times n}$ of the linearization around the upright unstable equilibrium, $(\boldsymbol{x}^*, u^*) = (\mathbf{0}, 0)$, the number of linear independent rows are equal to the state dimension, n. By use of Maplesoft ©, the A-matrix has proven full rank, i.e. $Rank(\boldsymbol{A}) = 4$. As stated in the introduction of this chapter, the nonlinear system is underactuated, indicating the linear system to be underactuated as well. This shows how an underactuated system can be controllable and is able to move from one initial state to a final state in finite time, but not with arbitrary trajectories.

Linear System Controllability

[5] states: A linear system, as Eq 3.24 or the pair $(\boldsymbol{A}, \boldsymbol{B})$ is said to be controllable if an external input can move the internal initial state $x(t = 0) = x_0$ to any final state x_1 in a finite time interval.

$$\mathcal{C} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{AB} & \boldsymbol{A}^2\boldsymbol{B} & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{B} \end{bmatrix} \ , \ \mathcal{C} \in \mathbb{R}^{n\times np} \tag{3.34}$$

The n-dimensional system is controllable if the controllability matrix $\mathcal{C}$ has the rank equal the dimension, $rank(\mathcal{C}) = n$, which both linearized systems with system matrices in Eq. 3.28 and 3.32 fulfills.

3.9.4 System Observability

From [5] observability is defined as: "The state equation Eq. 3.24 is said to be observable if for any unknown initial state $\boldsymbol{x}(0)$, there exists a finite $t_1 > 0$ such that the knowledge of the input u and output y over $[0,\ t_1]$ suffices to determine uniquely the initial state $\boldsymbol{x}(0)$. Otherwise, the equation is said to be unobservable."

The bicycle system is observable, as the system states can be measured by the potentiometer and the IMU mounted on the bicycle frame. Thus the angular positions of the bicycle and the mounted inverted pendulum can be obtained, and the velocities can be derived through low-pass filtering the measurements and derivate with respect to time. As the system is nonlinear, Kalman filter is not suitable as a state estimator. The low pass filter can be applied where:

$$\Omega_p = \frac{d\theta_p}{dt} \tag{3.35}$$

With the angular measurements, the torque can be measured by angular incrementation during one time sample:

$$\hat{\Omega}_p = \frac{\theta_p(k) - \theta_p(k-1)}{T_{samp}} \tag{3.36}$$

By low pass filtering this estimate, the angular velocities can be derived and used in the state space model:

$$\Omega_{p,used} = \frac{\hat{\Omega}_p}{1 + T_f \cdot s} \tag{3.37}$$

where s is the complex Laplace operator.

Chapter 4

System Control

In this chapter further research of the system stabilization controllers suggested in the authors own work in [12], is conducted. Based on previous research, the most relevant controllers from literature for Acrobot-control is investigated and tested. The goal is to investigate the functionality of these controllers and further on develop a controller stabilizing the bicycle system. These controllers are designed with swing-up control to move the system into a defined subspace where a balance controller is activated. By use of this structure the author will analyze the controller for further implementation and simulation. With the results, the controller functionality will be verified for the limited Acrobot system, shown in Figure 4.1. Based on the results of the controller functionality of the controllers designed by Lai et al. [11] and Kobayashi et al. [9] further research is conducted.

The swing-up controllers designed in [10], [11] and [9] are utilized to move the unlimited Acrobot from the straight down stable equilibrium and to the upright unstable position. The goal is to utilize the controller design within the defined angular limits, described in Section 3.6, and move system angles within the defined subspace where the balance controllers are activated to stabilize the system. Based on the results from simulation and analysis, controller modifications are made. In [9] it has been shown that swing-up and balance control can not be handled by one single control law, as the feedback gain becomes too large. With this physical impossible feedback gain on the LQR controller, separate swing-up and balance controllers has to developed. The goal is to utilize the nonlinear control to move the system angles back within the balance subspace when perturbations occur.

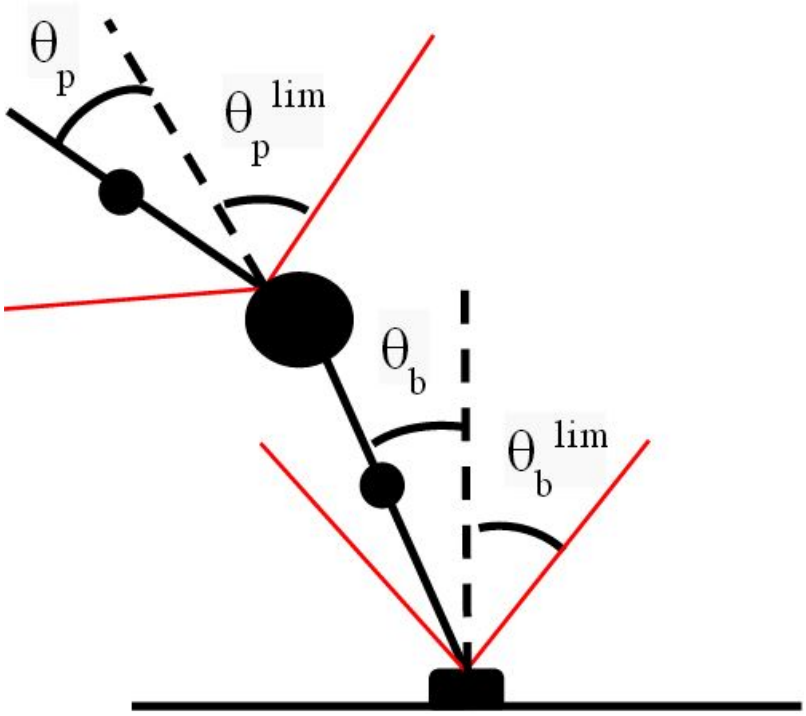


Figure 4.1: Illustration of system angular limitation, marked red

4.1 System Controller Based on Lai et al.

In [12] the controller designed by Lai et al. in [10] and [11] was presented. This controller design is based on an energy-based controller functioning in defined subspaces in the Acrobot manipulator workspace. This swing-up controller is utilized to increase system energy and move the system states within an attractive area, defined as the balance subspace. In this subspace, an optimal linear controller is utilized to stabilize the system around the upright unstable equilibrium. The goal is to investigate if this controller design can be utilized on the bicycle system presented as a limited Acrobot.

4.1.1 System Subspaces

In [12] the controller designed by Lai et al. was presented by two published articles. In both articles, the system subspace were defined and improved.

System Subspace Based on [10]

In [10] the subspaces are defined as:

- Swing-up subspace:
  $|\theta_b| > \lambda_1$ or $|\theta_b + \theta_p| > \lambda_2$

- Balance subspace:
  $|\theta_b| \leq \lambda_1$ and $|\theta_b + \theta_p| \leq \lambda_2$

Where λ_1 and λ_2 are small positive values. This theory is further developed in [11], where the idea of attractive area is introduced. Note that the angular velocity is not included in the criteria functions.

System Subspace Based on [11]

By defining the whole motion space as Σ, an attractive area, where the balance control is activated, is given by:

- Σ_3: $\left|mod\left(\frac{\theta_b}{2\pi}\right)\right| \leq \beta_1$ and $\left|mod\left(\frac{\theta_p}{2\pi}\right)\right| \leq \beta_2$

Where $mod(\bullet)$ is the residue modulus 2π. The swing-up subspace is then defined as: $\Sigma - \Sigma_3$. Due to singularity in one of the energy based controllers, Lai et al. designed two swing-up controllers. One applied in the subspace Σ_1 and a second in Σ_2. Thus the total swing-up subspace can be expressed as: $\Sigma_1 + \Sigma_2$.

In the system simulation presented in Section 4.1.5 and code implementation presented in Section 4.1.7, β_1 and β_2 is set to $\frac{\pi}{6}$ and $\frac{\pi}{4}$, respectively. The parameters are utilized as a maximum limit for the balance control, as the system angles moves outside the LQR region of attraction when the balance controller is activated, as shown in Figure 4.4. The LQR region of attraction is defined in Section 4.1.3.

4.1.2 Energy Based Swing-Up Control

The swing-up control presented in [10] and [11] is based on the total mechanical energy of the Acrobot system. With kinetic energy, T, as given in Eq. 3.4 with $\boldsymbol{M}(\boldsymbol{\theta})$ from Eq. 3.11 and potential energy, U, from Eq. 3.3, the energy function is expressed as:

$$\begin{aligned} E(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = T + U &= \frac{1}{2}\dot{\boldsymbol{\theta}}^T \boldsymbol{M}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} + U \\ &= \frac{1}{2}(I_b + m_b r_b^2 + m_p l^2)\dot{\theta}_b + m_p r_p l \dot{\theta}_b(\dot{\theta}_b + \dot{\theta}_p)\, cos\,\theta_p + \frac{1}{2}(I_p + m_p r_p^2)(\dot{\theta}_b + \dot{\theta}_p)^2 \\ &+ (m_b r_b + m_p l) g\, cos\,\theta_b + m_p r_p g\, cos(\theta_b + \theta_p) \end{aligned} \tag{4.1}$$

In 2004 Lai et al. designed the first controller in [10]. The design was based on the property of a positive semi-definite time derivative of the system energy, Eq. 4.1:

$$\dot{E}(\boldsymbol{\theta}\,,\,\dot{\boldsymbol{\theta}}) \geq 0 \tag{4.2}$$

Thereby the system energy has the property of being non-decreasing at all time when the states are within the swing-up subspace. The time derivative in Eq. 4.1 is derived as:

$$\dot{E}(\boldsymbol{\theta}\,,\,\dot{\boldsymbol{\theta}}) = \dot{\theta}_p \tau \tag{4.3}$$

where [10] has chosen the torque control swing-up to be:

$$\tau = sgn(\dot{\theta}_p) \cdot \upsilon\,,\;\; \upsilon \geq 0 \tag{4.4}$$

The constant υ is the control amplitude and can bee chosen arbitrary. To get smooth behavior of the Acrobot system, the control variable should be large when the system energy is low, and decrease as the system energy increases. Thus the actuation will be large when the system is far away from the upright unstable equilibrium, and more soft as it comes closer to the balance subspace.

In 2005 Lai et al. presented a further developed energy based control method in [11]. This theory is based on defining an attractive area and utilize an energy control based on a non-smooth Lyapunov function, which is a combination of two separate control laws. To achieve quick approach into the attractive area, the first control law is designed to quickly increase the system energy and make the state angles and the respective velocities approach zero. By introducing change of variables, as in [11], the system representation of Eq. 3.18 can be rewritten, with $\boldsymbol{x} = [\theta_b,\, \theta_p,\, \dot{\theta}_b,\, \dot{\theta}_p]^T = [\theta_b,\, \theta_p,\, \Omega_b,\, \Omega_p]^T$:

$$\dot{\theta}_b = \Omega_b \tag{4.5a}$$

$$\dot{\theta}_p = \Omega_p \tag{4.5b}$$

$$\dot{\Omega}_b = f_\mu(\boldsymbol{x}) + b_\mu(\boldsymbol{x})\tau \tag{4.5c}$$

$$\dot{\Omega}_p = f_\eta(\boldsymbol{x}) + b_\eta(\boldsymbol{x})\tau \tag{4.5d}$$

Note that τ is equivalent with u in Eq. 3.18 and 3.20, and:

$$\begin{bmatrix} f_\mu(\boldsymbol{x}) \\ f_\eta(\boldsymbol{x}) \end{bmatrix} = -\boldsymbol{M}^{-1}(\boldsymbol{\theta}) \left[\boldsymbol{C}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \boldsymbol{G}(\boldsymbol{\theta}) \right] \tag{4.6a}$$

$$\begin{bmatrix} b_\mu(\boldsymbol{x}) \\ b_\eta(\boldsymbol{x}) \end{bmatrix} = \boldsymbol{M}^{-1}(\boldsymbol{\theta})\boldsymbol{F}(\boldsymbol{\theta}) \tag{4.6b}$$

With $\boldsymbol{f}(\boldsymbol{x}) = [\Omega_b\ ,\ \Omega_p\ ,\ f_\mu(\boldsymbol{x})\ ,\ f_\eta(\boldsymbol{x})]^T$ and $\boldsymbol{b}(\boldsymbol{x}) = [0\ ,\ 0\ ,\ b_\mu(\boldsymbol{x})\ ,\ b_\eta(\boldsymbol{x})]^T$, the vectorial representation of Eq. 4.5 is given as:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{b}(\boldsymbol{x})\tau \tag{4.7}$$

The first and second Lyapunov function in [11], for energy based swing-up control, are listed with the corresponding control torque:

$$V_1(\boldsymbol{x}) = \frac{1}{2}(k_{p1}\theta_p^2 + k_{d1}\Omega_p^2 + k_{e1}[E(\boldsymbol{\theta}\ ,\ \dot{\boldsymbol{\theta}}) - E_0]) + \Delta_1 \tag{4.8a}$$

$$\tau = -\frac{k_{p1}\theta_p + k_{d1}f_\eta(\boldsymbol{x}) + \lambda_1 sat(\frac{\Omega_p}{\phi_1})}{k_{d1}b_\eta(\boldsymbol{x}) + k_{e1}(E(\boldsymbol{\theta}\ ,\ \dot{\boldsymbol{\theta}}) - E_0)} \tag{4.8b}$$

$$V_2(\boldsymbol{x}) = \frac{1}{2}(k_{p2}\theta_p^2 + k_{d2}\Omega_p^2) + \Delta_1 \tag{4.8c}$$

$$\tau = \dot{\tau} - \lambda_2 sat(\frac{\Omega_p}{\phi_2}), \quad \text{where } \dot{\tau} = -\frac{k_{p2}\Omega_p + k_{d2}f_\eta(\boldsymbol{x})}{k_{d2}b_\eta(\boldsymbol{x})} \tag{4.8d}$$

Where $sat(\bullet)$ is defined as:

$$sat(x) = \begin{cases} sgn(x), & if|x| \geq 1 \\ x, & if|x| < 1 \end{cases}$$

E_0 is the potential energy at the upright position, i.e. the unstable equilibrium. The constants: k_{p1}, k_{d1}, k_{e1}, λ_1, ϕ_1, k_{p2}, $k_{d2} > 0$. Δ_1 is a constant guaranteeing a non-smooth Lyapunov function. λ_2 is defined as:

$$\lambda_2 = \lambda_\alpha(1 + r), \quad -1 < r < 1 \tag{4.9}$$

where $\lambda_\alpha > 0$ is a constant. A fuzzy controller is designed to regulate the parameter r. $V_1(\boldsymbol{x})$ ensures that $\dot{V}_1(\boldsymbol{x}) \leq 0$ when the angles are in the swing-up subspace, guaranteeing the energy to increase as the second link straightens out, see [11] for more details. This controller is not suitable when $E(\boldsymbol{\theta}\ ,\ \dot{\boldsymbol{\theta}}) < E_0$, as singularity occurs when:

$$E(\boldsymbol{\theta}\ ,\ \dot{\boldsymbol{\theta}}) = E_0 - \frac{k_{d1}}{k_{e1}}b_\eta(\boldsymbol{x}) \tag{4.10}$$

Thus the second energy based control law, with Lyapunov function $V_2(\boldsymbol{x})$, is designed.

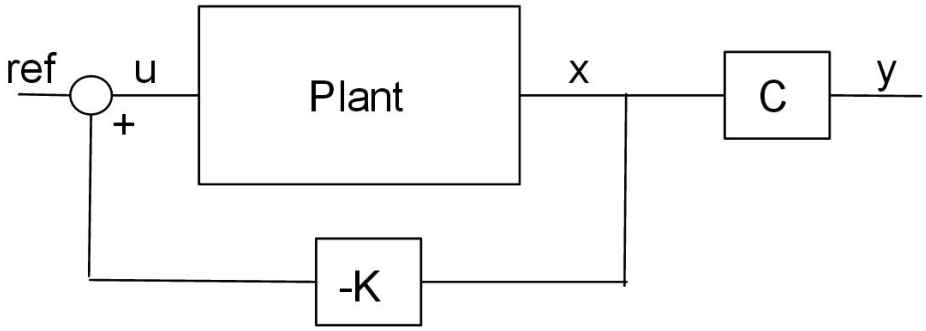


Figure 4.2: Block Diagram LQR controller with state feedback $u = -Kx$

4.1.3 LQR Balance Control

To stabilize the system around the upright equilibrium, Lai et al. used an LQR controller for balance control. In [12] the author designed an optimal linear state feedback controller, as shown in Figure 4.2, where stabilization is performed with pole placement. The MIMO LQR controller was designed for the linearized state-space system presented in Eq. 3.24 with the linear system matrices from Eq. 3.28. The LQR controller known as an infinite horizon LQ controller, uses optimal feedback theory for controlling the dynamical system at minimum cost by optimizing the gain matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ of the quadratic cost function, J:

$$J = \int_0^\infty \left(\boldsymbol{x}^T(t)\boldsymbol{Q}\boldsymbol{x}(t) + \boldsymbol{u}^T(t)\boldsymbol{R}\boldsymbol{u}(t)\right) dt \tag{4.11}$$

The weighted matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ are real, symmetric and positive definite, i.e. $\boldsymbol{Q} = \boldsymbol{Q}^T$, $\boldsymbol{R} = \boldsymbol{R}^T$ and $\boldsymbol{Q} > 0$, $\boldsymbol{R} > 0$. With the system output $\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}$, the term $\boldsymbol{y}^T\boldsymbol{C}\boldsymbol{y} = \boldsymbol{x}^T(\boldsymbol{C}^T\boldsymbol{Q}\boldsymbol{C})\boldsymbol{x}$ is nonnegative. Minimizing its integral forces $y(t)$ to approach zero as time goes to infinity. With $\boldsymbol{R} > 0$ the term $\boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u}$ is positive for $\boldsymbol{u} \neq \boldsymbol{0}$. Minimizing its integral will force $\boldsymbol{u}(t)$ to remain small. The relative values of $\boldsymbol{Q}$ and $\boldsymbol{R}$ will decide the amount of control action and speed of the response. The larger value of $\boldsymbol{Q}$, the more aggressive controller and the larger value of $\boldsymbol{R}$, the stricter system actuation. With the system matrices linearized around the upright equilibrium, $(\theta_b, \theta_p) = (0,0)$, the state feedback control is given by:

$$u = -\boldsymbol{K}\boldsymbol{x} \tag{4.12}$$

where $\boldsymbol{K} = \boldsymbol{R}\boldsymbol{B}^T\boldsymbol{P}$ and $\boldsymbol{P}$ is the symmetric positive definite solution of the algebraic Riccati equation:

$$\boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P} + \boldsymbol{M}^T\boldsymbol{Q}\boldsymbol{M} = 0 \tag{4.13}$$

When the LQR controller is developed, the weight matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ has to be defined. By use of the system parameters from Table 2.1, the following matrices were found through simulation of the linearized system:

$$\boldsymbol{Q} = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} \tag{4.14a}$$

$$R = 1 \tag{4.14b}$$

Notice that the weighted R-matrix, penalizing the system control input u, is scalar. This is intuitive as there is only one system actuator, giving the dimension of the control input: $u \in \mathbb{R}^1$. The following Matlab® function was used to get the optimal feedback gain matrix, $\boldsymbol{K}$:

$$[K, P, EIG] = lqr(A, B, Q, R) \tag{4.15}$$

By use of Eq. 4.15 the resulting state feedback gain matrix, $\boldsymbol{K}$, becomes:

$$\boldsymbol{K} = \begin{bmatrix} -975,63 & -57,24 & -227,34 & -19,74 \end{bmatrix} \tag{4.16}$$

By use of the Matlab® function in Eq. 3.29, the closed-loop system poles becomes:

$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} = \begin{bmatrix} -10.99 \\ -8.09 \\ -4.20 \\ -4.16 \end{bmatrix} \tag{4.17}$$

Showing the linear system is stable with $Re(\lambda_i) < 0$. Figure 4.3 shows the system time response with initial state angles, in radians, of $(\theta_b^i,\ \theta_p^i) = (0.05, 0.05)$.Figure 4.4 shows the system time response of the nonlinear system with the state feedback LQR controller. From simulation the maximum initial angles for the system with parameters from Table 2.1, is:

$$(\theta_b^i,\ \theta_p^i) = \left(\frac{0.6}{180}\pi, \frac{1.2}{180}\pi\right)[rad] \tag{4.18}$$

With absolute values of the angles larger than this, the system becomes unstable, making the angles fall to the angular limit. Thus, the region within the absolute angles given in Eq. 4.18 is defined as the LQR region of attraction.

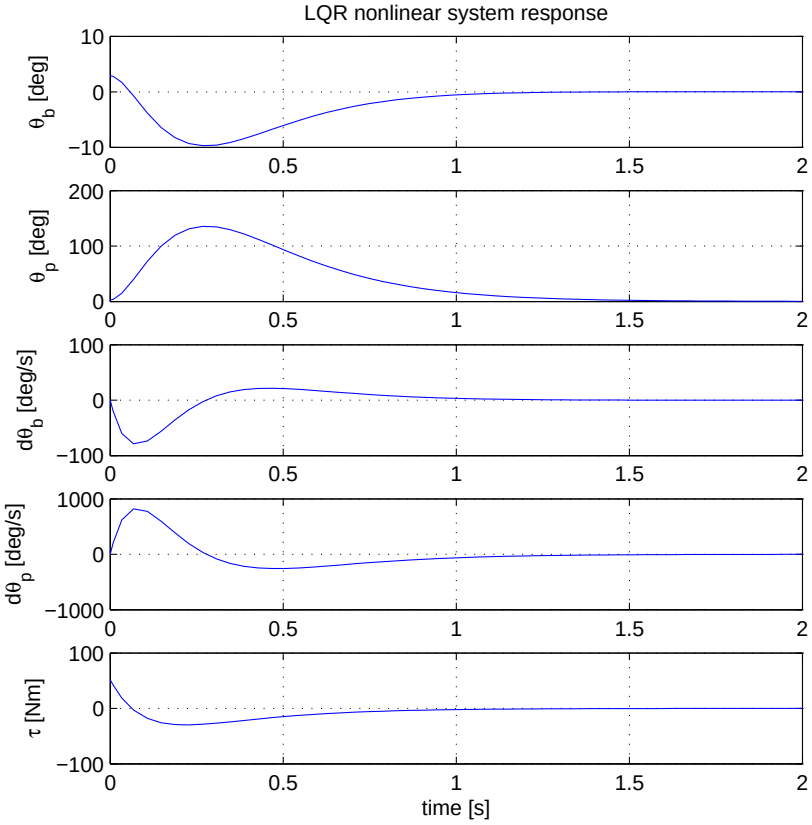


Figure 4.3: Linearized system time response with LQR control

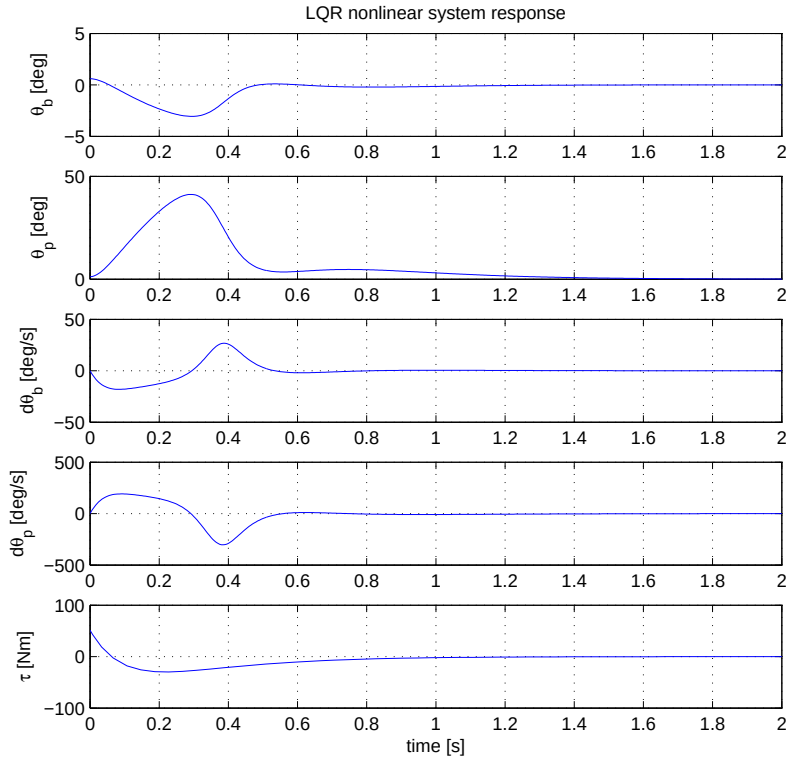


Figure 4.4: Nonlinear system time response with LQR control

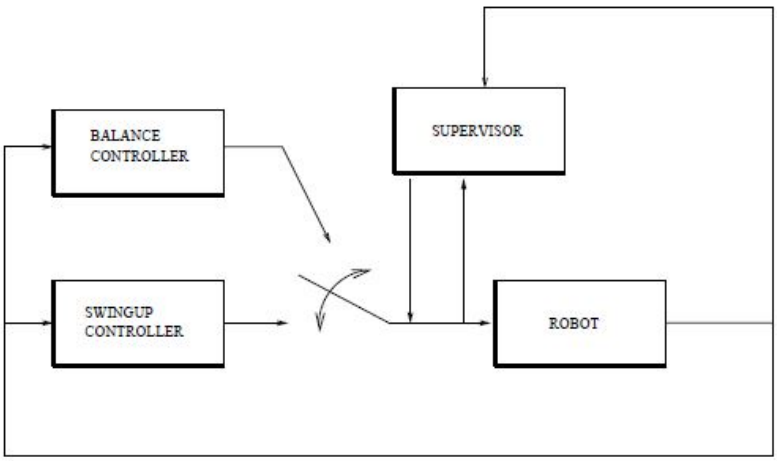


Figure 4.5: Illustration of switching control mode, from [17]

4.1.4 Control Mode Switching

Figure 4.5 shows an idea of the controller design, where the system switches between balance and swing-up control. Each controller is activated dependent on which subspace the system states are in. With two controllers for swing-up control a switching condition is introduced. Eq. 4.19 shows the condition for switching between the two control torques, dividing the swing-up area into the two subspaces Σ_1 and Σ_2.

$$k_{d1} b_\eta(\boldsymbol{x}) + k_{e1}[E(\boldsymbol{\theta}\,,\,\dot{\boldsymbol{\theta}}) - E_0] \leq \zeta \tag{4.19}$$

where $\zeta < 0$ is a constant. When Eq. 4.19 is satisfied, the control torque τ, of Lyapunov function $V_1(\boldsymbol{x})$, switches to the second control torque based on $V_2(\boldsymbol{x})$. Thus the first control law is used in subspace Σ_1 and the second control law is used in Σ_2. The angle $x_2 = \theta_p$ and angular velocity $x_4 = \dot{\theta}_p$ are driven and attracted toward zero if $\dot{V}_2(\boldsymbol{x}) < 0$.

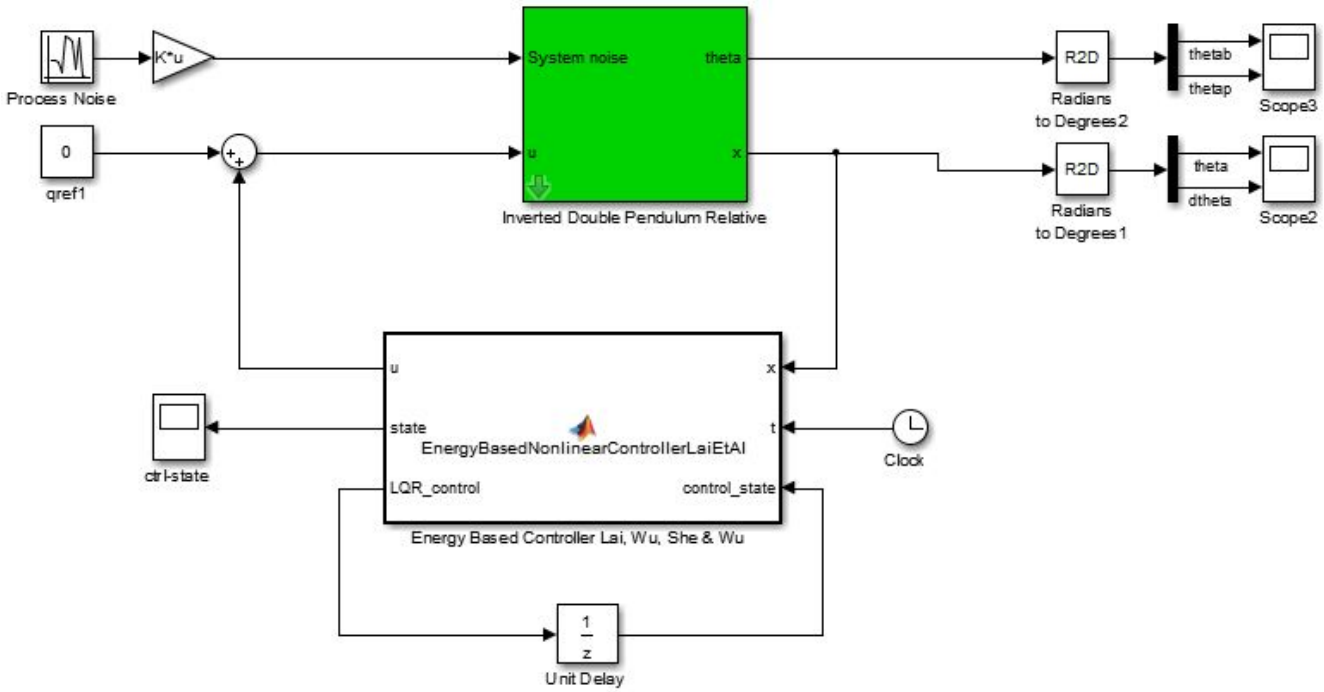


Figure 4.6: Illustration of System Implementation of Energy Based Controller by Lai et al.

4.1.5 Controller Analysis

In this section the controller functionality is analyzed and investigated through simulation, to verify whether the controller can be utilized on the bicycle system or not. With the controller design developed by Lai et al. in [10] and [11], the energy based swing-up control is combined with the LQR controller designed in Section 4.1.3 to give a complete controller, stabilizing the system. Figure 4.6 illustrates the design of the controller where the system has to switch between balance and swing-up. As mentioned in Section 4.1.3 the LQR is able to stabilize when $|\theta_b| \leq 0.6°$ and $|\theta_p| \leq 1.2°$. In [10] and [11] Lai et al. designed a controller moving the Acrobot system from the straight down stable equilibrium up to the upright unstable equilibrium. As the bicycle system is not able to reach the straight down stable equilibrium of $(\theta_b, \theta_p) = (180°, 0°)$, all system simulations is simulated with initial angular position within the angular limits presented in Section 3.6 and illustrated in Figure 4.1. Note that the limitation of the first link is relative to the reference frame, whereas the limitation of the second link is relative to the y_b-axis of the first link. The axis representation is presented in Section 3.2 and Figure 3.2.

As mentioned in the Section 4.1.2, the controller parameter r in Eq. 4.9 is regulated by a fuzzy controller. As the bicycle system can not reach the straight down stable equilibrium point due to the angular limitation, the controller will be implemented without fuzzy control. In [10] and [11] the fuzzy control was utilized to regulate parameter λ_2 of the second control law, resulting in regulation of the gain in the swing-up subspace. With the angular limitations reducing the swing-up subspace and initial position within these limits, the goal is to neglect fuzzy control and present λ_2 as a constant. The idea is to simulate the bicycle system without angular limitation to investigate the controller functionality. As the system has initial position within the angular limits, there will be nonzero velocity at the straight down equilibrium. Thus the fuzzy control is not needed to create system energy, as for the unlimited Acrobot system with initial position at the straight down equilibrium. When the unlimited Acrobot has initial position at the straight down equilibrium, the fuzzy control is utilized to perturb the system and increase system energy, such that the energy controllers can move the system states towards the upright unstable equilibrium.

From Eq. 4.8 one can see that the expression for τ of $V_1(x)$ and $V_2(x)$ contains the gains k_{p1}, k_{d1}, k_{p2} and k_{d2}. These system gains has are proportional and derivative gains. The proportional gains are multiplied with the system angles, θ_b and θ_p, whereas the derivative gains are multiplied with the expressions of $f_\eta(\boldsymbol{x})$ and $b_\eta(\boldsymbol{x})$. Thus the energy based swing-up control has the structure similar to the linear PD-controller, where the derivative gains are used to damp the system oscillations. The proportional gain multiplied with θ_p is used to increase the control torque, dependent on the system angle.

First the system simulation is performed without angular limitation, to verify the controller functionality on an unlimited Acrobot. The initial angle positions is set to $(\theta_b, \theta_p) = (2^\circ, -2^\circ)$, close to the upright unstable equilibrium, as the goal is to utilize the swing-up control to move the angles back into the LQR region of attraction. With the angular limitation presented in Section 3.6 and Figure 3.5 deactivated, the resulting time response is given in Figure 4.7. With initial angle position outside the LQR region of attraction and controller parameters as presented in [11] the system is able to utilize the energy based swing-up and switch to linear balance control. The controller parameters presented in [11] are:

$$\begin{cases} \beta_1 = \frac{\pi}{4}, & \beta_2 = \frac{\pi}{6}, & \lambda_1 = 38, & \phi_1 = 10 \\ k_{p1} = k_{d1} = 1, & k_{e1} = 0.2, & \zeta = -2 & \\ k_{p2} = k_{d2} = 1, & \phi_2 = 5, & \lambda_a = r = 0.5 & \end{cases}$$

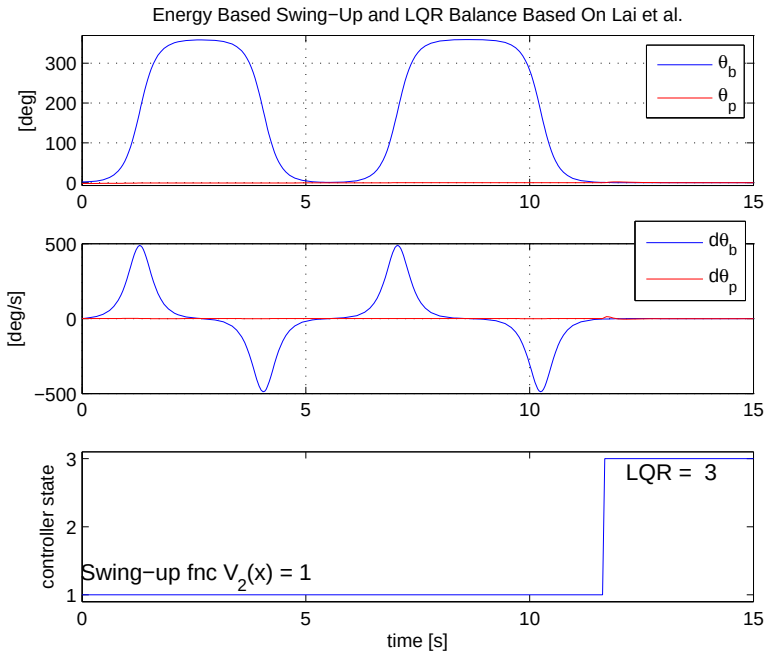


Figure 4.7: Time Response of Energy Based Controller by Lai et al. $(\theta_b^{init}, \theta_p^{init}) = (2°, -2°)$

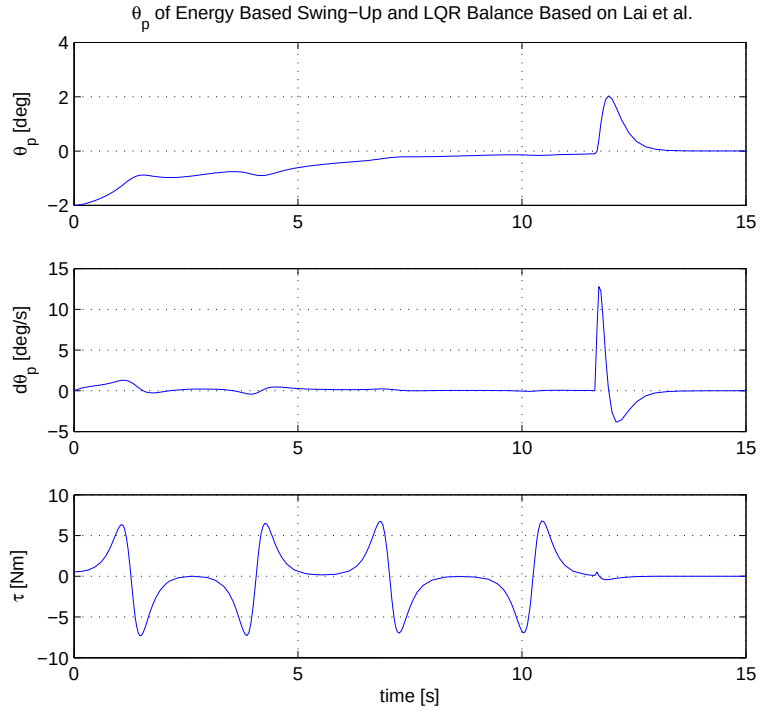


Figure 4.8: Time Response of θ_p, $\dot{\theta}_p$ and τ of Energy Based Controller by Lai et al. $(\theta_b^{init}, \theta_p^{init}) = (2°, -2°)$

Figure 4.7 illustrates how the system swings up by use of the first Lyapunov function, denoted with controller state 1. As singularities does not occur, the second Lyapunov function, denoted with controller state 2, is never activated by the controller. The figure also illustrates how the LQR is activated as the link angles are within the region of attraction. The LQR balancing control, denoted with controller state 3, is successful and the system stabilizes at the upright equilibrium. Figure 4.8 shows the

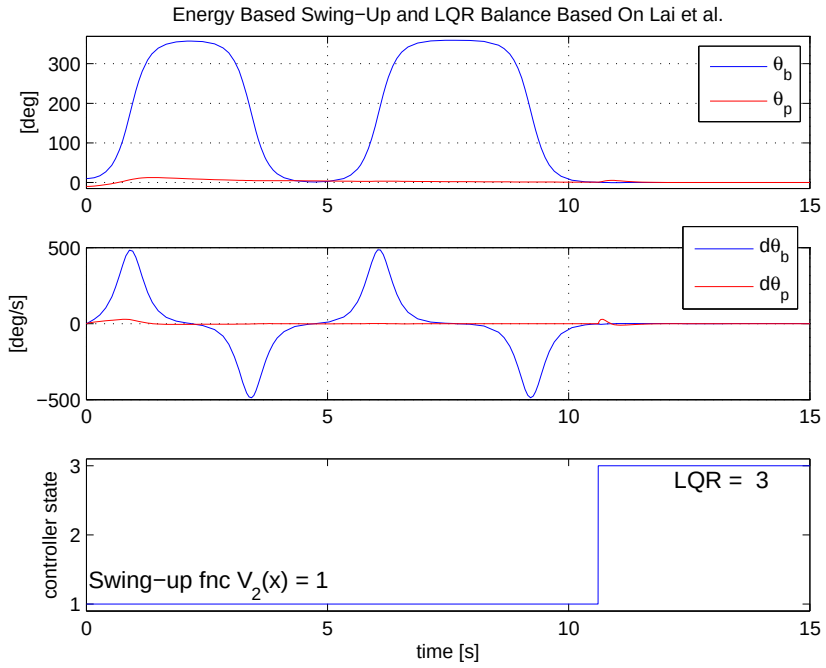


Figure 4.9: Time Response of Energy Based Controller by Lai et al. $(\theta_b^{init}, \theta_p^{init}) = (10°, -10°)$

control torque applied to the system. It is clear that the energy based controllers utilize the kinetic energy generated as the system falls towards the straight down equilibrium. As the actuator only creates instantaneous acceleration in θ_p, the small change of the angle is not enough to counteract the oscillatory behavior of θ_b shown in Figure 4.7. The bicycle link uses four swings before the system angles reaches the LQR region of attraction and switches to balance control. As the system oscillates, the angle of θ_p illustrated in Figure 4.8 attracts towards zero, making it possible for the system to utilize the LQR controller. When the bicycle link angle, θ_b, reaches the LQR region of attraction, the controller has already controlled θ_p towards zero, making both angles fulfill the initial angle requirements for the LQR controller, stated in Section 4.1.3. Simulation of initial positions further away from the upright unstable equilibrium is also performed. Figure 4.9 shows how the system is able to stabilize with initial angle position of $(\theta_b,\ \theta_p) = (10°, -10°)$. As for the simulation with initial positions closer to the upright equilibrium, the system is able to utilize the energy based swing-up controller and swing the system back and forth until both states are within the LQR region of attraction. When the system angles are within the balance subspace, the balance controller is activated, and the system is able to stabilize around the upright unstable equilibrium. Figure 4.10 shows the time response of θ_p, where the angle is controlled towards zero throughout the simulation. Thus the controller is able to move both system angles within the LQR region of attraction to activate the LQR balance control, as shown in Figure 4.9.

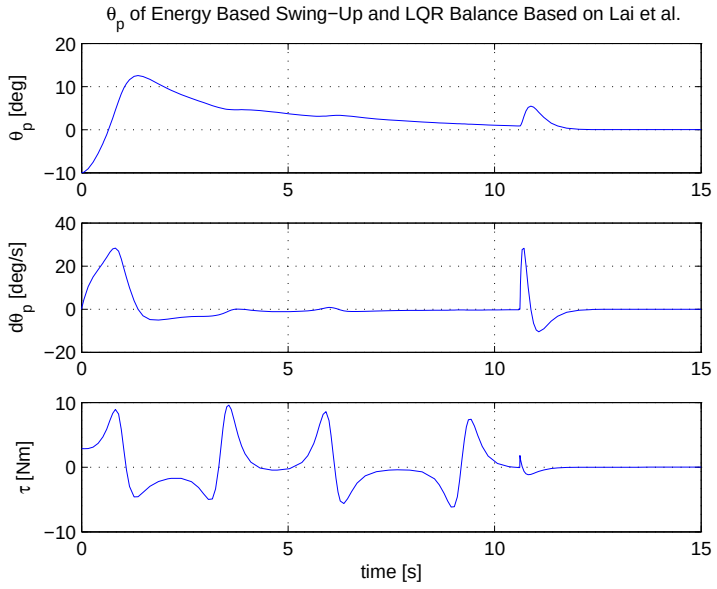


Figure 4.10: Time Response of θ_p, $\dot{\theta}_p$ and τ of Energy Based Controller by Lai et al. $(\theta_b^{init}, \theta_p^{init}) = (10°, -10°)$

In Figure 4.7 it is shown that the angles moves outside the angle limits for the bicycle system, thus the controller parameters has to be tuned. By tuning the controller parameters, the goal is to verify whether or not the controller can perform system stabilization within the angular limits. By activating the angular limitation presented in Section 3.6 the model becomes a limited Acrobot, equivalent to the bicycle system. With these limits defining the workspace of the bicycle system, simulation will be conducted to verify if the controller is satisfying the requirements - utilizing the swing-up control to move the system within the LQR region of attraction.

Through multiple simulations and tuning of controller parameters, the controller is unsuccessful to stabilize the bicycle system. From Figure 4.7 this can be intuitively described. As the bicycle angle, θ_b, is outside of the LQR region of attraction, the controller does not apply torque to move the bicycle link towards the upright equilibrium, but it allows the system to fall down and swing up on the opposite side. As the bicycle links swings, the control torque is utilized to increases system energy to oscillate the bicycle angle and move both system states within the LQR region of attraction. Thus the energy based swing-up controller is not designed to apply control torque and create instant acceleration towards the upright equilibrium, when the bicycle angle, θ_b, starts to fall. The controller applies torque to swing up the system from the initial position of $(\theta_b, \theta_p) = (2°, -2°)$, moving counter clockwise towards the upright equilibrium of $(\theta_b, \theta_p) = (360°, 0°)$, which is equivalent to $(\theta_b, \theta_p) = (0°, 0°)$ in the rotational system. As mentioned in the introduction of this chapter, the Acrobot is based on the idea of an acrobat gymnastic. The controller designed by Lai et al. has the functionality of a real life

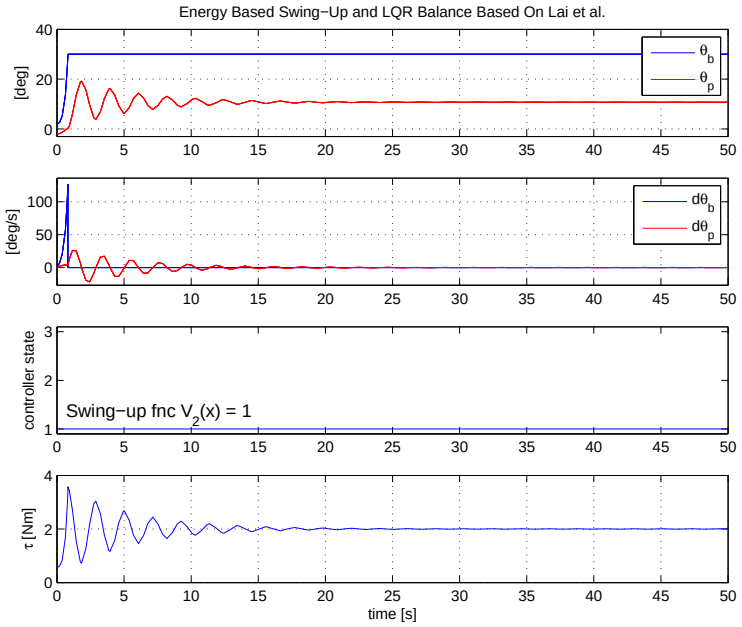


Figure 4.11: Time Response of θ_p, $\dot{\theta}_p$ and τ of Energy Based Controller by Lai et al. $(\theta_b^{init}, \theta_p^{init}) = (2°, -2°)$

gymnastic, where he swings up from the straight down position by swinging back and forth until he reaches the upright position. This is further described in Section 4.3. Thus the control torque applied from the controller is not able to stabilize the bicycle system within the angular limits.

As shown in Figure 4.11 the bicycle tilt angle, θ_b, falls to the limit, and the controller is not able to stabilize the system. The control torque stabilizes the inverted pendulum angle, θ_p, at approximately $11°$, with the required input torque, τ, of approximately 2[Nm]. Thus the energy based swing-up controller is not able to control the angle of θ_b or the angle θ_p within the angular limits. A lot of time was spent on parameter tuning, without success. As the swing-up control is not able to move the system within the LQR region of attraction, the balance control is never activated. With the controller functionality described above, the energy based swing-up controller designed by Lai et al. does not satisfy the requirements for the bicycle system. The simulation in Figure 4.11 was simulated with initial positions of $(\theta_b, \theta_p) = (2°, -2°)$ and the following system parameters:

$$\begin{cases} \beta_1 = \frac{\pi}{4}, & \beta_2 = \frac{\pi}{6}, & \lambda_1 = 38, & \phi_1 = 10 \\ k_{p1} = 50, & k_{d1} = 10, & k_{e1} = 0.2, & \zeta = -2 \\ k_{p2} = 50, & k_{d2} = 20, & \phi_2 = 5, & \lambda_a = r = 0.5 \end{cases}$$

4.1.6 Concluding Remarks

As the controller does not satisfy the system requirements of stabilizing the bicycle system within the angular limitations, the energy based swing-up controller designed by Lai et al. is not applicable as a stabilizing controller. Thus other controller methods is investigated to derive a functional stabilizing controller for the bicycle system presented as a limited Acrobot. However, the controller shows how the inverted pendulum angle, θ_p, has to be controlled towards zero simultaneously as the bicycle angle, θ_b, is moved towards the LQR region of attraction to apply the LQR balance controller. This property is utilized further on in the controller investigation in this thesis.

4.1.7 Controller Implementation

The following code is controller implementation of the energy based swing-up control, where functions "*SysChangeOfVar*" and "*hysteresis*" are designed for the complete controller functionality. Description of the controller implementation is given after each Matlab® function presented. The controller parameters were tuned and tested through simulation, and are presented with the respective figures in Section 4.1.5.

EnergyBasedNonlinearCOntrollerLaiEtAl:

```
function [u, LQR_control, state]=EnergyBasedNonlinearControllerLaiEtAl(x,t,...
    control_state)
l = 0.72;     % [m] approximately
rb = 0.4;     % [m] approximately
rp = 0.2;     % [m] approximately
mb = 31.118; % [kg] approximately
mp = 5;       % [kg] approximately
g = 9.81;     % [m/s^2] gravity acceleration
Ib = 0;       % [Nm^2] Moment of Inertia of bicycle
Ip = 0;       % [Nm^2] Moment of Inertia of pendulum

M =[Ib+Ip+mb*rb^2+mp*l^2+mp*rp^2+2*mp*rp*l*cos(x(2)) Ip+mp*rp^2+mp*rp*l*cos(x(2));
           Ip+mp*rp^2+mp*rp*l*cos(x(2))                        Ip+mp*rp^2         ];

C = [-2*mp*rp*l*x(4)*sin(x(2)) -mp*rp*l*x(4)*sin(x(2)) ;
       mp*rp*l*x(3)*sin(x(2))              0           ];

G = [-(mb*rb+mp*l)*g*sin(x(1))-mp*rp*g*sin(x(1)+x(2)) ;
                  -mp*rp*g*sin(x(1)+x(2))             ];

F = [0 1]';

% Energy at Upright eq.pt (potential energy)
E0 = (mb*rb+mp*l)*g+mp*rp*g;
% Kinetic Energy
T = 1/2*(Ib+mb*rb^2+mp*l^2)*x(3)^2+mp*rp*l*x(3)*(x(3)+x(4))*cos(x(2))+...
    1/2*(Ip+mp*rp^2)*(x(3)+x(4))^2;
% Potential Energy
U = (mb*rb+mp*l)*g*cos(x(1))+mp*rp*g*cos(x(1)+x(2));
% Total System Energy
```

```
31 E = T+U;
32
33 [f, b] = SysChangeOfVar(x,M,C,G,F);
34
35 f_mu  = f(3);
36 f_eta = f(4);
37 b_mu  = b(3);
38 b_eta = b(4);
39
40 % Controller Constants:
41 lambda_1 = 38;  % > 0
42 phi_1    = 10;  % > 0
43 k_p1     = 1;   % > 0
44 k_d1     = 1;   % > 0
45 k_e1     = 0.2; % > 0
46
47 lambda_a = 0.5;
48 r = 0.5; % -1 < r < 1
49 lambda_2 = lambda_a*(1+r);
50 phi_2    = 5; % > 0
51 k_p2     = 1; % > 0
52 k_d2     = 1; % > 0
53
54 beta_1 = pi/8;
55 beta_2 = pi/6;
56
57 % LQR limitations:
58 LQR_lim_x1 = 0.6/180*pi;
59 LQR_lim_x2 = 1.2/180*pi;
60
61 limit = [beta_1 beta_2 LQR_lim_x1 LQR_lim_x2];
62 if t == 0
63     [x_lim, LQR_control] = hysteresis(x,limit,false);
64 else
65     [x_lim, LQR_control] = hysteresis(x,limit,control_state);
66 end
67
68 x1_lim = x_lim(1);
69 x2_lim = x_lim(2);
70
71 if mod(abs(x(1)),2*pi)<=x1_lim && mod(abs(x(2)),2*pi)<=x2_lim && LQR_control==1
```

```
72      %Control law 3
73      u = -[-975.6281  -57.2364 -227.3441  -19.7419]*x;
74      state = 3;
75  elseif E == E0-k_d1/k_e1*b_eta %if singularity occurs
76      %Control law 2
77      du = -(k_p2*x(2)+k_d2*f_eta)/(k_d2*b_eta);
78      u = (du-lambda_2*sat(x(4)/phi_2));
79      state = 2;
80  else
81      %Control law 1
82      u = -(k_p1*x(2)+k_d1*f_eta+lambda_1*sat(x(4)/phi_1))/...
83          (k_d1*b_eta+k_e1*(E-E0));
84      state = 1;
85  end
```

In the function "*EnergyBasedNonlinearCOntrollerLaiEtAl*", the system energy in Eq. 4.1 is calculated. The function "*SysChangeOfVar*" is applied to calculate the values of f_μ, f_η, b_μ and b_η for use in the energy based control torque, given in Eq. 4.8b and 4.8d. The controller implementation utilizes the function "*hysteresis*" to set the limitations of the LQR control in real time as the simulation is running. The limits are either defined as the LQR region of attraction or the maximum angles needed for the LQR to stabilize the system. If the system is within the LQR region of attraction, the LQR control is activated, and the angular limits are set to maximum. As long as the system angles are within these limits after the LQR is activated, the state feedback controller is utilized. If the system moves outside the angular controller limits, the LQR is deactivated and the system switches to energy based swing-up control.

SysChangeOfVar:

```
function [f, b] = SysChangeOfVar(x, M, C, G, F)

[rowq, columnq] = size(x);
if columnq ~= 1 && rowq ~= 0
    disp('state vector consist of mutiple columns')
    return;
else
    if columnq ~= 1 && rowq == 1
        x = x'; % if q is row vector, make q column vector
    end
    f  = zeros(length(x),1);
    b  = zeros(length(x),1);

    f(1:length(x)/2,1) = x(length(x)/2+1:end,1);

    %Make the row vector C(q,q')*x'
    [~, columnC] = size(C);
    Cq = C;
    if columnC > 1
        Cq = C*[x(3) x(4)]';
    end

    CG = -(Cq+G);
    f(length(x)/2+1:end,1) = M\CG;
    b(length(x)/2+1:end,1) = M\F;
end
```

The output of "*SysChangeOfVar*" is the rewritten system state vectors $f(\boldsymbol{x})$ and $b(\boldsymbol{x})$ presented in Eq. 4.6, where the input vector $\boldsymbol{x}$ is presented in Eq. 3.16 and the input matrices M, C, G and F are the system matrices presented in Eq. 3.10. Note that the notation of q is the system states before change of variables, as in Eq. 3.16, i.e. $\boldsymbol{q} = \boldsymbol{\theta}$.

hysteresis:

```
function [x_lim, LQR_control] = hysteresis(x,limit,LQR_state)

x1_max_lim = limit(1);
x2_max_lim = limit(2);
x1_LQR_lim = limit(3);
x2_LQR_lim = limit(4);

if mod(abs(x(1)),2*pi) <= x1_LQR_lim && mod(abs(x(2)),2*pi) <= x2_LQR_lim
    x_lim = [x1_max_lim x2_max_lim];
    LQR_control = 1;
elseif mod(abs(x(1)),2*pi)<=x1_max_lim && mod(abs(x(2)),2*pi)<=x2_max_lim...
       && LQR_state==true
    x_lim = [x1_max_lim x2_max_lim];
    LQR_control = 1;
elseif mod(abs(x(1)),2*pi) > x1_max_lim || mod(abs(x(2)),2*pi) > x2_max_lim
    x_lim = [x1_LQR_lim x2_LQR_lim];
    LQR_control = 0;
else
    x_lim = [x1_LQR_lim x2_LQR_lim];
    LQR_control = 0;
end
```

"*hysteresis*" is designed to give the desired angular limitation for use in the LQR control. As the system reaches the region of attraction of the LQR, the function changes the system limits to β_1 and β_2 to enlarge the controller workspace when the LQR stabilization is activated. The function defines the angular limits to maximum such that the LQR control can operate outside the angles of $\theta_b \leq 0.6$ and $\theta_p \leq 1.2$ to perform the balance control, as shown in Figure 4.4. The function also returns the control state: $true$ if the LQR control is activated and $false$ otherwise. $true$ and $false$ are denoted 1 and 0, respectively. The use of $mod(abs(x,2\pi))$ is used to check whether the absolute angular value is between 0 and 2π.

4.2 System Controller Based on Kobayashi et al.

In the authors previous work in [12] the controller design by Kobayashi et al. in [9] was presented. As stated in the introduction of this chapter, Kobayashi et al. has proven that swing-up and balance control can not be handled by one single control law, due to excessively large feedback gain. Thus the controller presented is, as the controller by Lai et al., an energy based swing-up controller. The controller is utilized to increase system energy and move the system states of the Acrobot towards the upright equilibrium. As the system moves towards the upright position, the controller switches to balance control when the states reaches the attractive area to stabilize the system. The goal is to investigate whether or not the control design can be utilized on the bicycle system presented as a limited Acrobot.

4.2.1 Energy Based Swing-Up Control

The swing-up controller presented in [9] is based on system energy, as the design by Lai et al. Thus the energy function is given as:

$$\begin{aligned} E(\boldsymbol{\theta}\,,\,\dot{\boldsymbol{\theta}}) &= T + U \\ &= \frac{1}{2}(I_b + m_b r_b^2 + m_p l^2)\dot{\theta}_b + m_p r_p l \dot{\theta}_b(\dot{\theta}_b + \dot{\theta}_p)\, cos\,\theta_p + \frac{1}{2}(I_p + m_p r_p^2)(\dot{\theta}_b + \dot{\theta}_p)^2 \\ &+ (m_b r_b + m_p l) g\, cos\,\theta_b + m_p r_p g\, cos(\theta_b + \theta_p) \end{aligned} \tag{4.20}$$

By defining an energy function $E_z(\boldsymbol{\theta}\,,\,\dot{\boldsymbol{\theta}})$ Kobayashi et al. defined a swing-up controller to move the system states towards the upright unstable equilibrium:

$$E_z(\boldsymbol{\theta}\,,\,\dot{\boldsymbol{\theta}}) = E(\boldsymbol{\theta}\,,\,\dot{\boldsymbol{\theta}}) - E_0 \tag{4.21}$$

where E_0 is the potential energy at the upright inverted equilibrium point, i.e. $U|_{\theta_b=\theta_p=0}$. Passivity of the two-link Acrobot yields as $u^T y = \dot{V}$, see Definition 6.3. in Khalil[8]:

$$\dot{E} = [0\ \ \tau] \begin{bmatrix} \dot{\theta}_b \\ \dot{\theta}_p \end{bmatrix} = \dot{\theta}_p \tau \tag{4.22}$$

Note time derivative of the system energy is equal to Eq. 4.3. The defined system energy function $E_z(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ is considered by:

$$\dot{E}_z(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = -\mu E_z(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \tag{4.23}$$

where $\mu > 0$.

By Eq. 4.23 the property of $E_z(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \rightarrow 0$ is obtained, thereby the system will move towards the upright unstable equilibrium. By Eq. 4.21 and the negative semi-definite time derivative of $E_z(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ the energy function in Eq. 4.21 goes to zero, as the system energy $E(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ is equal to the upright equilibrium point potential energy, E_0. Thereby the energy based control function from [9] in Eq. 4.24 is able to move the system towards the upright unstable position based on a negative semi-definite time derivative of the Lyapunov function, $E_z(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$. From Eq. 4.22 and 4.23 the control torque is given by:

$$\tau = -\frac{\mu E_z(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})}{\dot{\theta}_p + \epsilon} \tag{4.24}$$

where $\epsilon \ll 1$ and $\dot{\theta}_p \equiv \Omega_p$.

One can see from Eq. 4.24 that the system control torque does not contain any proportional or derivative gains as the control torques in Eq. 4.8. Thus there are no gains directly coupled with the system states to control the angles with a PD-structure and create controllable oscillations of θ_p to counteract the movement of θ_b.

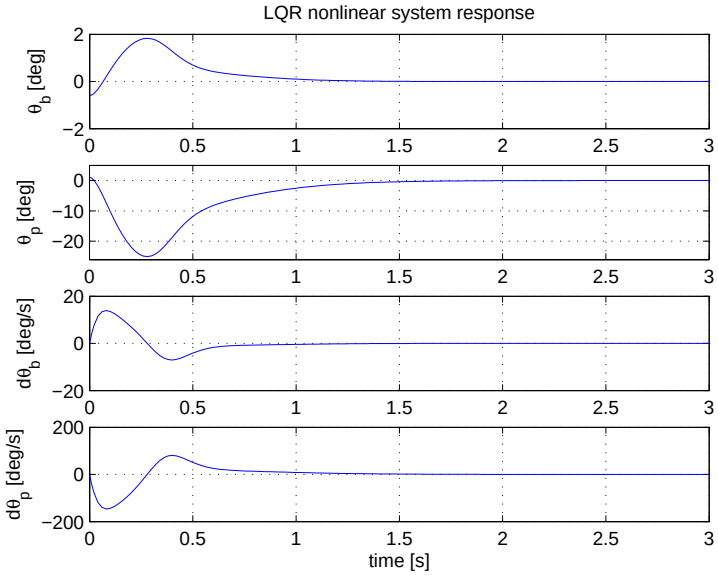


Figure 4.12: Nonlinear system time response with LQR control

4.2.2 LQR Balance Control

As for the energy based swing-up controller by Lai et al. the controller designed in [9] utilize the LQR controller. When the swing-up controller moves the system states within the LQR region of attraction, the balance control is activated. The LQR controller uses the optimal feedback gain, $\boldsymbol{K}$, in Eq. 4.16, derived by use of the Matlab® function given in Eq. 4.15:

$$[K, P, EIG] = lqr(A, B, Q, R)$$

$$\boldsymbol{K} = \begin{bmatrix} -975,63 & -57,24 & -227,34 & -19,74 \end{bmatrix}$$

The LQR controller is equivalent to the controller derived in Section 4.1.3, as both controllers are derived from system linearization around the upright unstable equilibrium of (θ_b, θ_p) = (0, 0) and weight matrices as given in Eq. 4.14. With the state feedback controller $u = -\boldsymbol{K}x$, the time response of the nonlinear system with initial positions (θ_b, θ_p) = ($-0.6°$, $1.2°$) is shown in Figure 4.12. With the LQR activated through the entire simulation, the mounted inverted pendulum is utilized to counteract the falling bicycle angle of θ_b. The bicycle angle is moved past the upright equilibrium and the inverted pendulum angle θ_p is moved slowly up towards the upright unstable equilibrium from maximum negative excursion. As the inverted pendulum is accelerated slowly in positive direction, the counteracting torque moves the bicycle angle in negative direction, stabilizing both system angles around the upright unstable equilibrium of (θ_b, θ_p) = ($0°$, $0°$).

4.2.3 Control Mode Switching

In [9] the energy based swing-up controller is utilized for both balance and swing-up control. As the bicycle system has initial positions around the upright equilibrium, the goal is to utilize the optimal linear controller for system stabilization, when the state angles are within the LQR region of attraction. If perturbations occur, the system states might move further away from the upright position, and energy based swing-up control is activated. Thus the goal is to combine the energy based swing-up and LQR controller in a complete controller for stabilization. Note that Kobayashi et al. designed a swing-up controller based on linear system representation, where a pole assigning controller was designed to move the system angles away from the initial straight down position of the unlimited Acrobot. This controller is not satisfactory as the pole assignment is used to create an unstable equilibrium and perturb the system from the initial straight down position. Thus the swing-up control designed by pole assignment is not applicable as the non-smooth system behavior is undesirable to utilize for moving the states within the LQR region of attraction.

When the system angles are within the LQR region of attraction, the balance control is activated. The maximum controller angles for the LQR controller is then set to β_1 and β_2, as for the controller by Lai et al., such that the system is able to stabilize, as shown in Figure 4.12. If the system angles moves outside these limits, the swing-up controller is activated to move the system states within the LQR region of attraction, shown in Eq. 4.18.

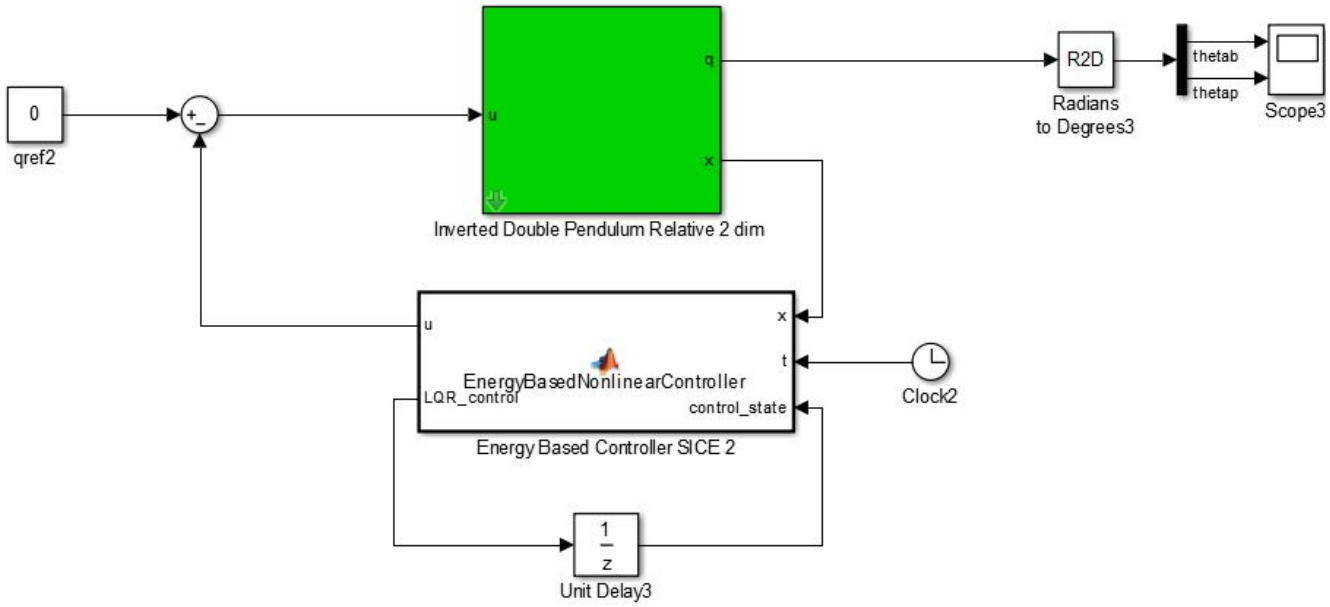


Figure 4.13: Illustration of System Implementation of Energy Based Controller by Kobayashi et al.

4.2.4 Controller Analysis

Throughout the semester, the author spent a lot of time on the design by Kobayashi et al., shown in Figure 4.13, without successfully implementing a functionally controller. Through simulations of the unlimited Acrobot it has been verified that the controller implementation is unable to stabilize the system around the unstable upright equilibrium. Multiple attempts of parameter tuning were performed, without success. The controller design by Kobayashi et al. is based on the energy function $E_z(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ without state feedback in the controller. There are no gains coupled with the system state angles in the control design, resulting in a controller which is unable to apply a desired trajectory of θ_p to counteract the repelling bicycle angle, θ_b, and move the states within the LQR region of attraction.

By looking at Eq. 4.24, the control output of the energy based swing-up controller can be analyzed. With the angular velocity of the second link, $\dot{\theta}_p$, in the denominator, the control torque becomes larger as the velocity decreases. At very small values, $\dot{\theta}_p \ll 1$, the system control output becomes excessively large, especially at the initial position of the simulations, when $\dot{\theta}_p = 0$. Thus the controller gives a physical impossible control torque to the bicycle system. As mentioned in Section 4.2.1, the swing-up controller does not contain any derivative gains coupled with the system states. Thus the controller only utilizes Eq. 4.21, where the property of $E_z(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \rightarrow 0$ is obtained. The controller is the based on the negative definite time derivative of the energy function. This verifies that the system is able to stabilize, but is not any guarantee of system stability within a desirable time range. As none of the system states are coupled in the feedback control, the controller is unable to control each system state as desired with respect to the reference angles. Without gains directly coupled with the system states in the

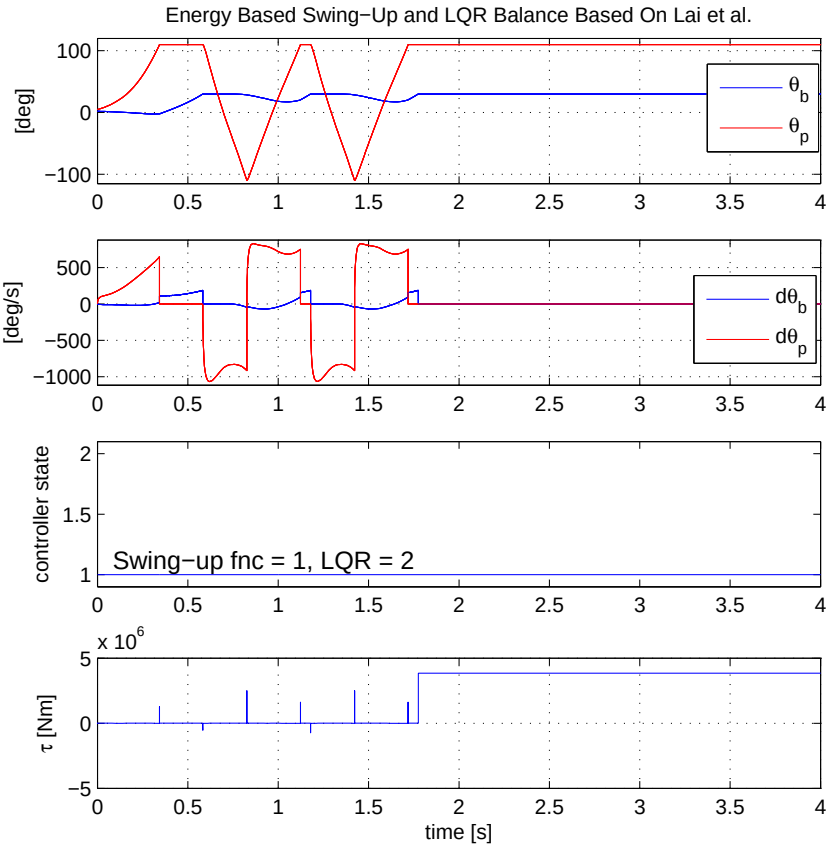
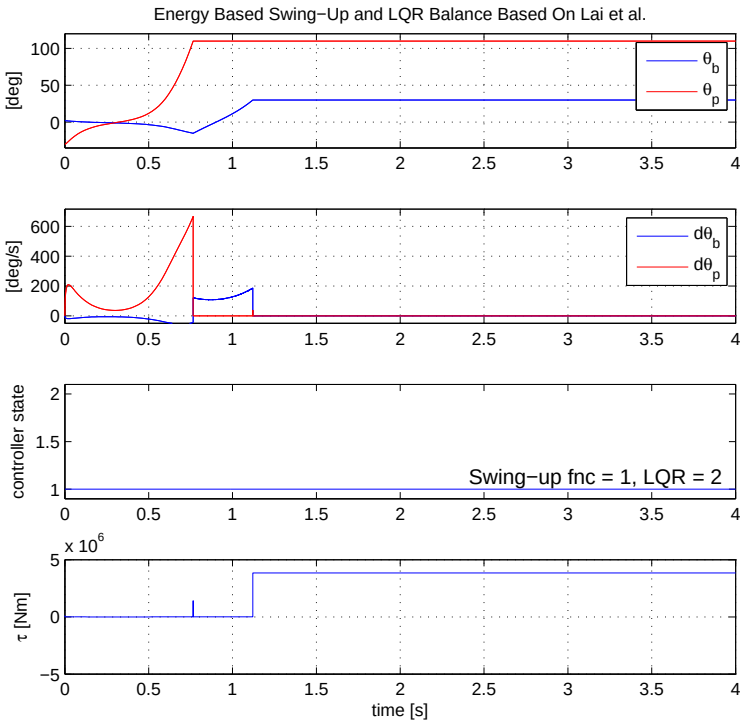
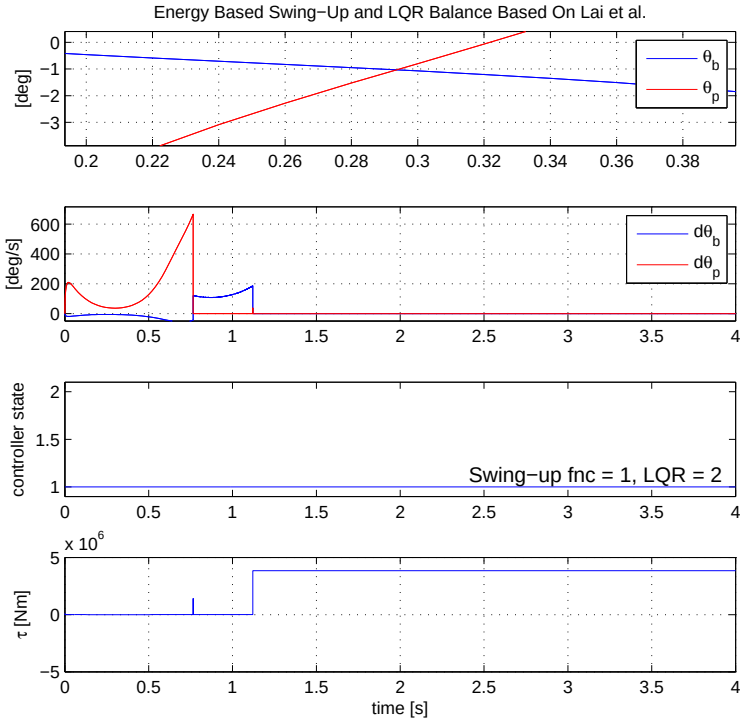


Figure 4.14: Time Response of Energy Based Controller by Kobayashi et al. $(\theta_b^{init}, \theta_p^{init}) = (2°, -5°)$

controller, the system is unable to create instantaneous acceleration in the inverted pendulum angle, θ_p, to counteract the repelling bicycle angle, θ_b. The design by Lai et al. shows how the system states can be utilized to control the angle of θ_p in the swing-up subspace such that both system angles are within the LQR region of attraction when the angle of θ_b reaches the balance subspace. As the design by Kobayashi et al. does not utilize the system states it is unable to control the angle of θ_p towards zero throughout the system swing-up within the angular limits. Based on the property of negative semi-definite time derivative energy function, the design only utilizes the mathematical property of system stability based on the Lyapunov function without consideration of state trajectories and a desirable time range for system swing-up. Thus the author has not been able to tune the design by Kobayashi et al. to be a well functioning controller for the bicycle system, as it is unable to utilize the inverted pendulum to counteract the repelling bicycle tilt angle and move the states back towards the upright equilibrium within the angular limitations. As the controller is not able to control both system angles, θ_b and θ_p, towards zero and into the balance subspace, the implementation based on Kobayashi et al. does not meet the requirements of the bicycle system.



(a) Full System Time Response



(b) Time Response Zoom-in Angles θ_b and θ_p

Figure 4.15: Time Response of Energy Based Controller by Kobayashi et al. $(\theta_b^{init}, \theta_p^{init}) = (2°, -30°)$

Figure 4.14 illustrates how the controller is unable to stabilize the bicycle system around the upright unstable equilibrium, with initial angular positions of $(\theta_b, \theta_p) = (2°, -5°)$. As the controller is not able to accelerate the inverted pendulum angle, θ_p, enough to counteract the falling bicycle tilt angle, θ_b, both system angles falls to the limits. Thus the controller is unable to move the system states within the LQR region of attraction for balancing control. The figure shows how the inverted pendulum tries to move the bicycle tilt angle towards the upright equilibrium without success. The torque output becomes excessively large and gives a physical impossible control torque to the system. Even though the controller is not able to stabilize the system within the angular limits, it shows an important property. The inverted pendulum is utilized to move the bicycle tilt angle towards the upright position of $\theta_b = 0$. By applying positive acceleration to the inverted pendulum, the bicycle angle is moved with negative acceleration from the positive angle limit back towards the upright equilibrium. By applying acceleration to the angle of θ_p, the bicycle angle is accelerated in the opposite direction. Thus the inverted pendulum is utilized to create torque and lift the bicycle up towards the upright position, which is an essential property utilized in the further investigation of the limited Acrobot presented in Section 4.3. Figure 4.15 shows the time response with initial positions of $(\theta_b, \theta_p) = (2°, -30°)$. In Figure 4.15(b) it is shown how the system angles are close to the LQR region of attraction at $-1°$. As the controller does not utilize the system states counteract the repelling bicycle angle, it is not able to control each system angle towards the upright equilibrium of $(\theta_b, \theta_p) = (0°, 0°)$ and into the LQR region of attraction. From Figure 4.14 and 4.15 it is shown how different the controller performance is based on the initial angular positions. None of the system angles in Figure 4.14 are close to the LQR region of attraction at the same time sample, whereas for time response in Figure 4.15 the inverted pendulum is accelerated in positive direction to counteract the falling bicycle angle θ_b. With the initial positions of $(\theta_b, \theta_p) = (2°, -30°)$ the inverted pendulum is accelerated to move θ_b back towards the upright unstable equilibrium, without being able to apply the linear balance controller.

4.2.5 Concluding Remarks

With no PD-structure in the energy based swing-up controller, the design by Kobayashi et al. is not able to control the system states into the balance subspace. As no gains are directly coupled with the system state angles and their respective time derivatives, the controller is unable to control the system states towards the upright equilibrium within a desired time range for the unlimited Acrobot. As illustrated in Figure 4.14 and 4.15, the controller performance is varying dependent on the initial positions of the system angles. Thus when the controller only utilize the property of a negative definite time derivative of the energy function, and does not control the system states directly, it is not able to stabilize the bicycle system within the angular limits. Thereby the controller implementation based on Kobayashi et al. is not satisfying the requirements for system stabilization. However, the design shows how the acceleration of θ_p moves the bicycle angle, which is desirable. This idea is utilized further on in the controller investigation in this thesis.

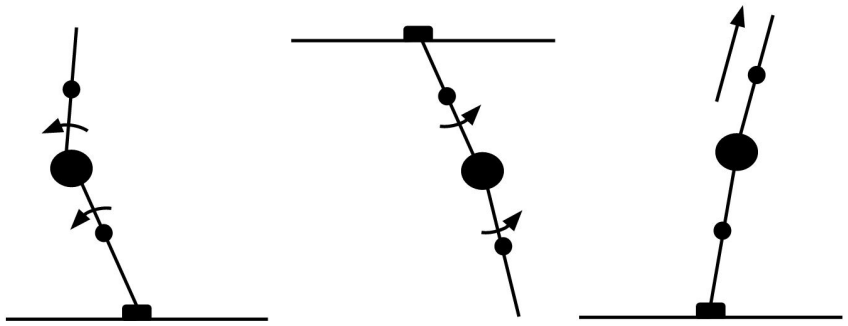
4.2.6 Controller Implementation

The following code is controller implementation of the energy based swing-up control, where function "*hysteresis*", shown in Section 4.1.7, is utilized in to set the limits of the LQR balance controller. Description of the controller implementation is given after the Matlab® function presented. The parameters were tuned and tested through system simulation.

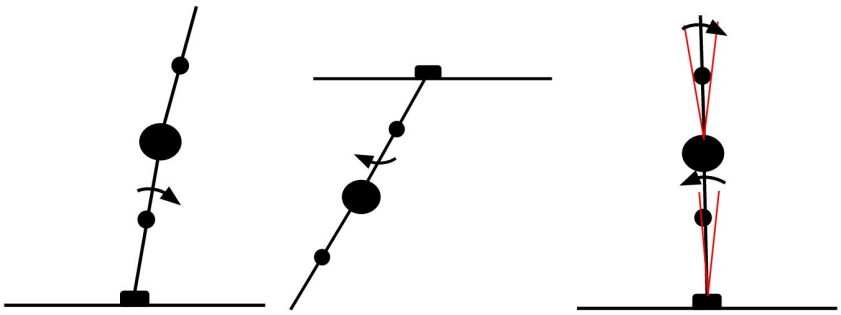
```
function [u, LQR_control, state] = EnergyBasedNonlinearController(x,t,...
    control_state)
l = 0.72;     % [m] approximately
rb = 0.4;     % [m] approximately
rp = 0.2;     % [m] approximately
mb = 31.118; % [kg] approximately
mp = 5;       % [kg] approximately
g = 9.81;     % [m/s^2] gravity acceleration
Ib = 0;       % [Nm] Moment of Inertia of bicycle
Ip = 0;       % [Nm] Moment of Inertia of pendulum

% States:
thetab  = x(1);
thetap  = x(2);
dthetab = x(3);
dthetap = x(4);

% Energy at Upright eq.pt (potential energy)
E0 = (mb*rb+mp*l)*g+mp*rp*g;

% Total System Energy
E = 1/2*(Ib+mb*rb^2+mp*l^2)*x(3)^2+mp*rp*l*x(3)*(x(3)+x(4))*cos(x(2))+1/2*...
    (Ip+mp*rp^2)*(x(3)+x(4))^2+(mb*rb+mp*l)*g*cos(x(1))+mp*rp*g*cos(x(1)+x(2));

Ez = E-E0;

% Controller Constants:
mu = 100; % mu > 0
epsilon = 0.001; % epsilon << 1


beta_1 = pi/8;
beta_2 = pi/6;
```

```
34
35 LQR_lim_x1 = 0.6/180*pi;
36 LQR_lim_x2 = 1.2/180*pi;
37
38 limit = [beta_1 beta_2 LQR_lim_x1 LQR_lim_x2];
39 if t == 0
40     [x_lim, LQR_control] = hysteresis(x,limit,false);
41 else
42     [x_lim, LQR_control] = hysteresis(x,limit,control_state);
43 end
44 x1_lim = x_lim(1);
45 x2_lim = x_lim(2);
46
47
48 % Controller:
49 if mod(abs(x(1)),2*pi)<=x1_lim && mod(abs(x(2)),2*pi)<=x2_lim && LQR_control==1
50     u = -[-975.6281  -57.2364 -227.3441  -19.7419]*x; % LQR
51     state = 2;
52 else
53     u = -mu*Ez/(dthetap+epsilon); %Energy Swing-Up Controller
54     state = 1;
55 end
```

The controller calculates the system energy given in Eq. 4.20 and the potential energy at the upright equilibrium, E_0. With the energy function defined in Eq. 4.21, the nonlinear energy based swing-up controller is defined by Eq. 4.24. The function "*hysteresis*" is utilized to defined the angular limitations of the balance subspace. When the system is outside the LQR region of attraction and the previous controller state was not in LQR mode, the limits are set to $(|\theta_b|, |\theta_p|) = (0.6°, 1.2°)$ as stated in Section 4.1.3. If the system states are within the LQR region of attraction, the balance control is activated, and the limits are set to β_1 and β_2, respectively.



(a) Falls from initial state (b) Swings past stable eq. pt. (c) Reaches highpoint

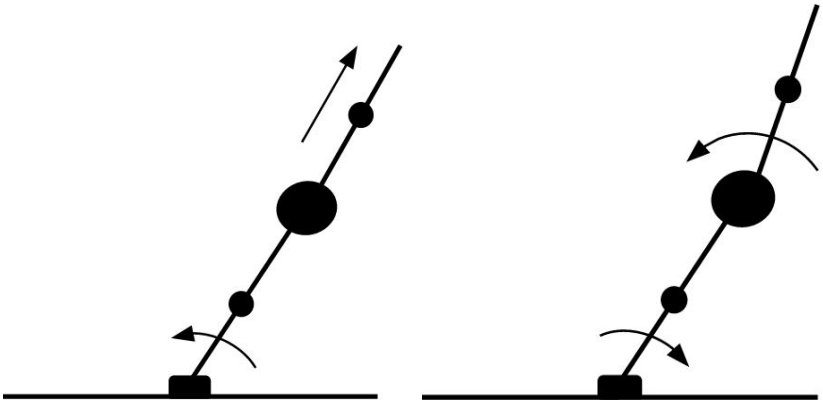


(d) Falls down from highpt. (e) Swings past stable eq. pt. (f) Swings into LQR ROA

Figure 4.16: Illustration of Energy Based Swing-Up Control by Lai et al. on unlimited Acrobot

4.3 System Controller for Feedback Linearized System

As shown in the previous sections, the controllers by Lai et al. and Kobayashi et al. were not able to stabilize the bicycle system within the angular limits. As the controllers were originally designed for an unlimited Acrobot, the energy based swing-up controllers were not able to satisfy the requirements for the bicycle system. From Figure 4.7 and 4.11 it is shown how the controller designed by Lai et al. makes the system fall away from the upright equilibrium when the initial position is outside the LQR region of attraction. As shown in Figure 4.16, this energy based controller utilizes the increasing kinetic energy as the angular velocities increases when the system falls towards the straight down equilibrium of the unlimited Acrobot. As the system swings, the control torque is applied to increase the kinetic energy and move the system angles closer to the LQR region of attraction for each swing. This behavior is similar to the real life acrobat gymnastic. When the acrobat swings up from the straight down position, he starts by swinging his legs back and forth, as the Acrobot would do from the same initial position. By applying torque from the hip, the acrobat is able to swing his legs and increase the kinetic energy and move his body closer to the upright position.



(a) Without actuation (b) With actuation giving $\dot{\Omega}_p > 0$

Figure 4.17: Illustration of Negative Torque Contribution of Inverted Pendulum Onto Bicycle

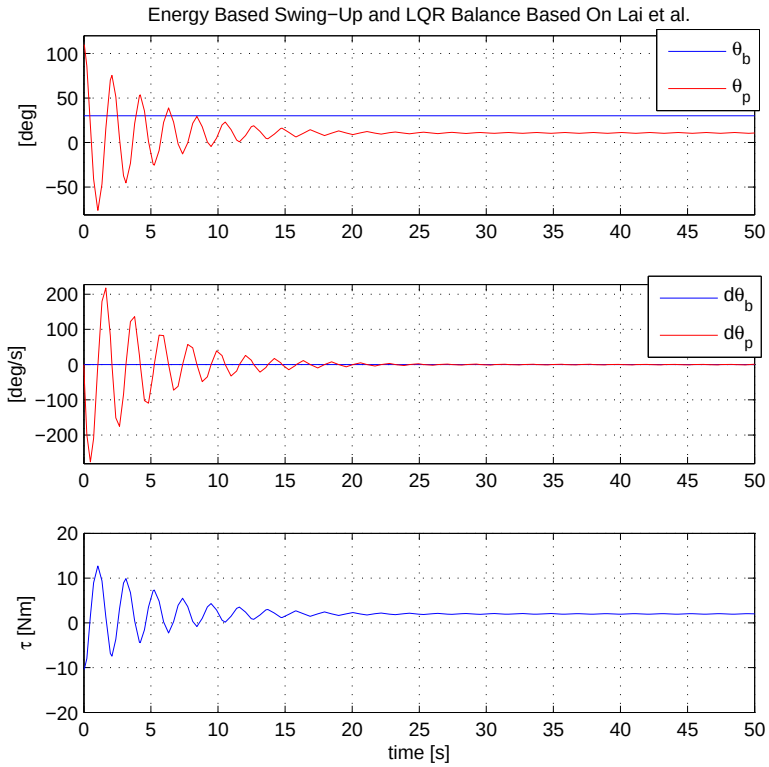
When the acrobat reaches the highpoint of each swing, he stops the movement of the legs, which is equivalent to having $\Omega_p = 0$ on the bicycle system, to maximize the swing-up. This is to prevent the legs to push the upper body downwards from the highpoint. From Eq. 3.20a it is shown how the acceleration of θ_b is dependent of the inverted pendulum angle velocity, Ω_p. At the highpoint of each swing, both velocities are close to zero, giving:

$$J_{tot}\frac{d\Omega_b}{dt} = +gm_pl\left(\left(1+\frac{m_br_b}{m_pl}\right)sin\,\theta_b - cos\,\theta_p\,sin(\theta_b+\theta_p)\right) - \left(1+\frac{l}{r_p}cos\,\theta_p\right)\tau$$

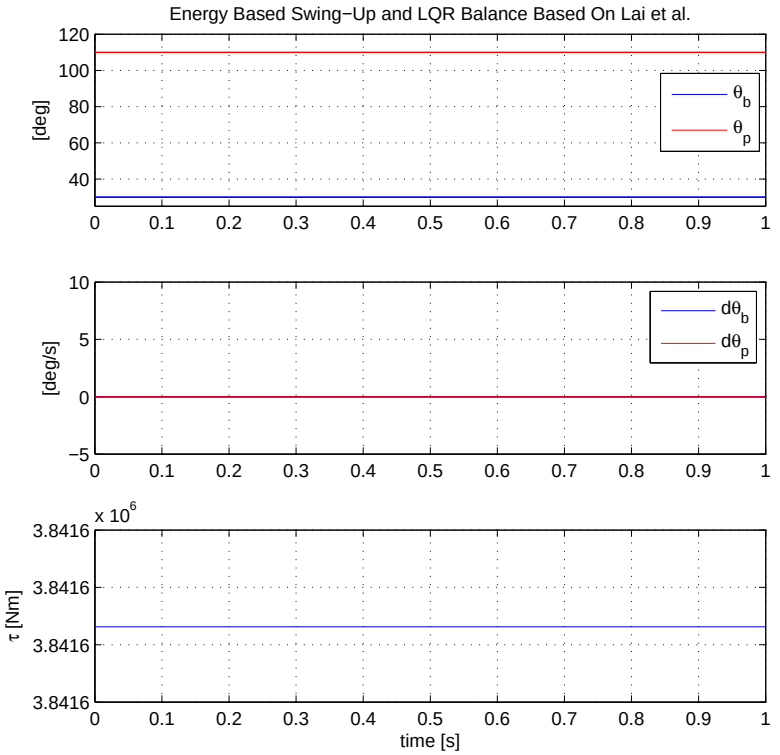
With the legs parallel to the upper body, the acrobat will have $\theta_p \approx 0$ and the equation for $\frac{d\Omega_b}{dt}$ is mainly dependent on the gravitation force and the torque applied from the system actuator. The gravity forces divided on the inertia-term J_{tot} will reduce the angular velocity of the bicycle as it moves up towards the upright equilibrium. By analyzing the torque-term of the equation, it is shown how the actuator torque contributes with negative acceleration to the bicycle angle, θ_b, with respect to its direction of movement. The increasing angle of θ_b contributes with negative torque as long as $\theta_p = cos^{-1}\left(-\frac{r_p}{l}\right) > 0$, which decreases the acceleration of θ_b. With a swing-up controller utilizing this functionality, the angle of θ_p is kept close to zero at the highpoint of each swing and the balance controller can be applied when the angle of θ_b reaches LQR region of attraction. Figure 4.17 illustrates how the inverted pendulum contributes with negative torque onto the bicycle.

The most important property that separates the limited and unlimited Acrobot system is the angular velocities at the angle limits. When the controller by Lai et al. is able to stabilize the unlimited Acrobot, it utilizes the increasing system energy. By applying torque from the system actuator, the Acrobot is able to move the angles closer to the upright equilibrium for each swing. When the system angles θ_b and θ_p moves past the respective limit angles of ± 30 and ± 110, the angular velocities are unequal to zero. As the energy based swing-up controller utilizes the angular velocity of Ω_b, it is able to increase the system energy and move the states towards the balance subspace by controlling the angle of θ_p towards zero. Unlike the unlimited Acrobot, the system with angular limitations is not able to utilize this property. As the angular velocities are zero at the limits, the bicycle system is unable to utilize the angular velocity of Ω_b as the energy based swing-up controllers do. To move the system angles towards the upright equilibrium, the controller has to accelerate the inverted pendulum to increase the kinetic energy of the bicycle system. From Eq. 3.20a and Figure 4.7 it is clear how the velocities in squared affects the acceleration of the bicycle angle, θ_b. The figure also shows how large the angular velocity of the bicycle angle is relative to the velocity of the inverted pendulum. Thus the main difference from the swing-up controller for the limited and the unlimited Acrobot, is that the limited Acrobot has to increase the system energy with the inverted pendulum instead of the bicycle link, to move the system angels towards the upright position. Thereby the challenge is to create a controller which is able to utilize the inverted pendulum to stabilize the bicycle angle, θ_b, at its upright position as it simultaneously controls the angle of θ_p towards zero.

Figure 4.18(a) and 4.18(b) shows the time response of the limited Acrobot with the controller design by Lai et al. and Kobayashi et al., respectively. The initial positions are at the angular limitation of $(\theta_b, \theta_p) = (30°, 110°)$. As for the simulation of the limited Acrobot in Figure 4.11, the controller by Lai et al. is unable to move the bicycle angle away from the limit and towards the upright position. The controller oscillates the angle of θ_p which stabilizes at the approximate value of 11° with the corresponding torque of approximately 2 [Nm]. The control design is unable to utilize the oscillatory behavior of the inverted pendulum to move the bicycle from the limit. The implementation based on Kobayashi et al. also fail to move the system towards the upright equilibrium. As the controller does not utilize the system states directly with controller gains, it is unable to utilize the inverted pendulum to create torque, moving the bicycle towards its upright position. Due to zero angular velocity, $\Omega_b = \Omega_p = 0$ at the limits, the control torque becomes excessively large, as the denominator of Eq. 4.24 is equal to ϵ, pushing the angles towards the limit.



(a) Swing-up by Lai et al.

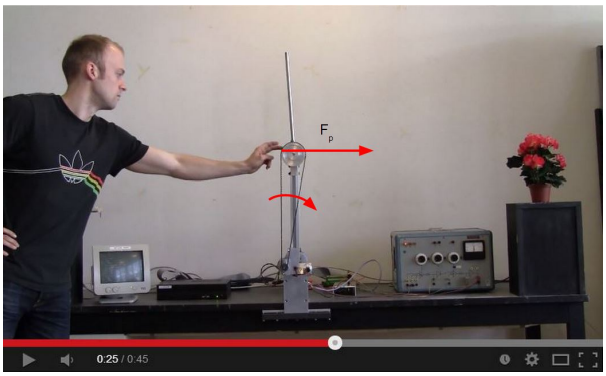


(b) Swing-up by Kobayashi et al.

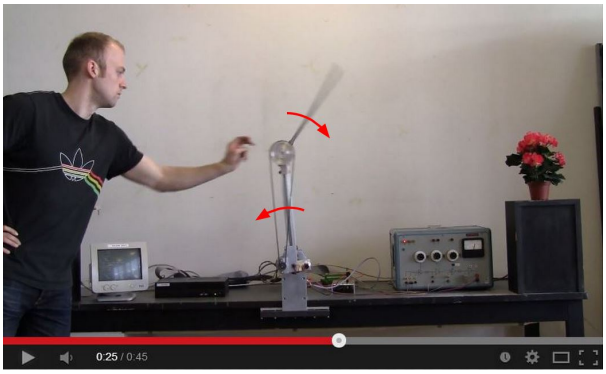
Figure 4.18: Time response of Energy Based Swing-up Controllers with initial positions at limits

4.3.1 Swing-Up Control with Feedback Linearization

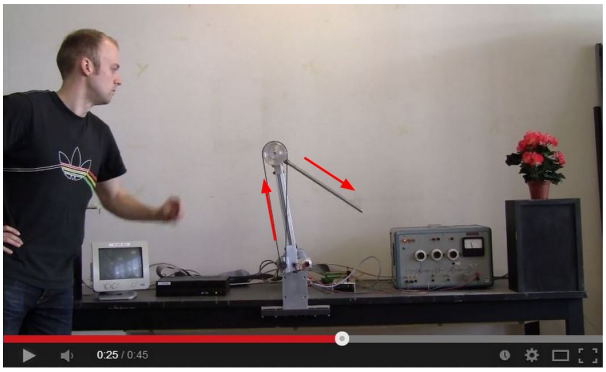
During the semester of working with the project thesis, the author came across a video [1] of a functional Acrobot. Throughout the semester of working with the master thesis, the author was able to find the corresponding paper [2] by Andersen, Skovgaard and Ravn, which presents several nonlinear controllers, both with and without angular limitation, for stabilization of the underactuated unlimited Acrobot. With the system parameters used in [2], Andersen et al. presents the corresponding time responses of the unlimited Acrobot, with the respective controllers. All the nonlinear controllers moves outside the angular limitations of the bicycle system, whereas the LQR balance controller is able to stay within the limits. It is not specified which controller is applied in the video, but Figure 4.19 illustrates the system behavior when perturbation occur at the upright position. After swing-up and system stabilization, the inner link, corresponding to the bicycle, is perturbed, as shown in Figure 4.19(a), and the outer link, corresponding to the mounted inverted pendulum, is utilized to stabilize the inner link. By accelerating the outer link in the same direction of the falling inner link, the actuator and outer link contributes with a counteracting torque onto the inner link, pushing it back towards the upright equilibrium. When the angular velocity Ω_b reaches zero, the outer link angle θ_p has reached maximum negative excursion, as shown in Figure 4.19(c). An important property, is that the angle of θ_b is not set to zero, but is moved past the upright equilibrium in positive direction to balance the system before the angle of θ_p is the slowly moved towards its upright equilibrium as well. When the angle of θ_p is accelerated in positive direction, the counteracting torque onto the inner link moves the angle of θ_b in negative direction, as shown in Figure 4.19(d). When the angle of θ_p reaches the maximum positive excursion, the inner link angle is placed slightly past the upright equilibrium at a negative angle, as shown in Figure 4.19(e), to balance the system. Thus, as shown in Figure 4.19(f), the angle of θ_p is accelerated in negative direction and the counteracting torque onto the inner link moves the angle of θ_b in positive direction. Thereby both angles are moved towards the upright equilibrium of $(\theta_b, \theta_p) = (0^\circ, 0^\circ)$ as shown in Figure 4.19(g). The balance control performance by Andersen et al. [2] illustrated in Figure 4.19 has the same system behavior of the LQR balance control described in Section 4.2.2.



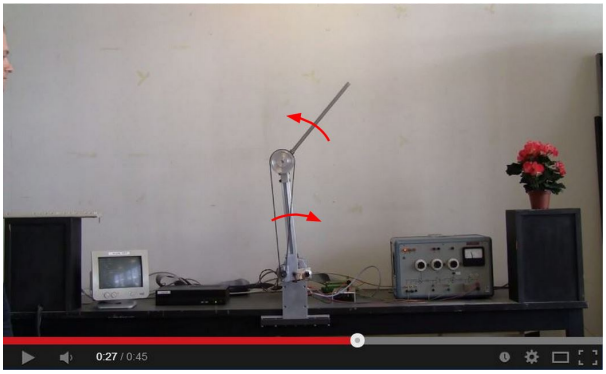
(a) System perturbation, F_p



(b) Counteracting acceleration of θ_p onto θ_b



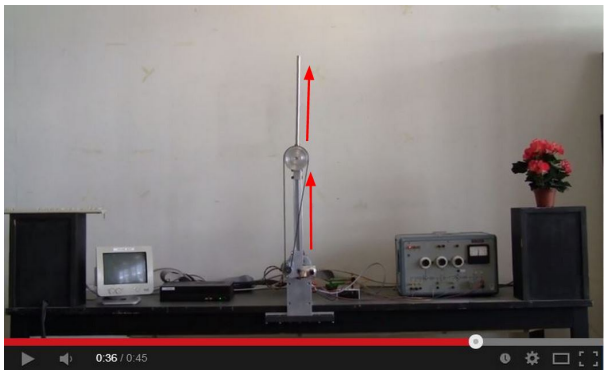
(c) θ_p at maximum negative excursion



(d) Moving θ_p towards upright eq. pt.

(e) θ_p at maximum positive excursion

(f) Moving θ_p towards upright eq. pt.



(g) System balancing around urpight eq. pt.

Figure 4.19: Illustration of time response of system balance with controller by Andersen et al. [2] from video [1]

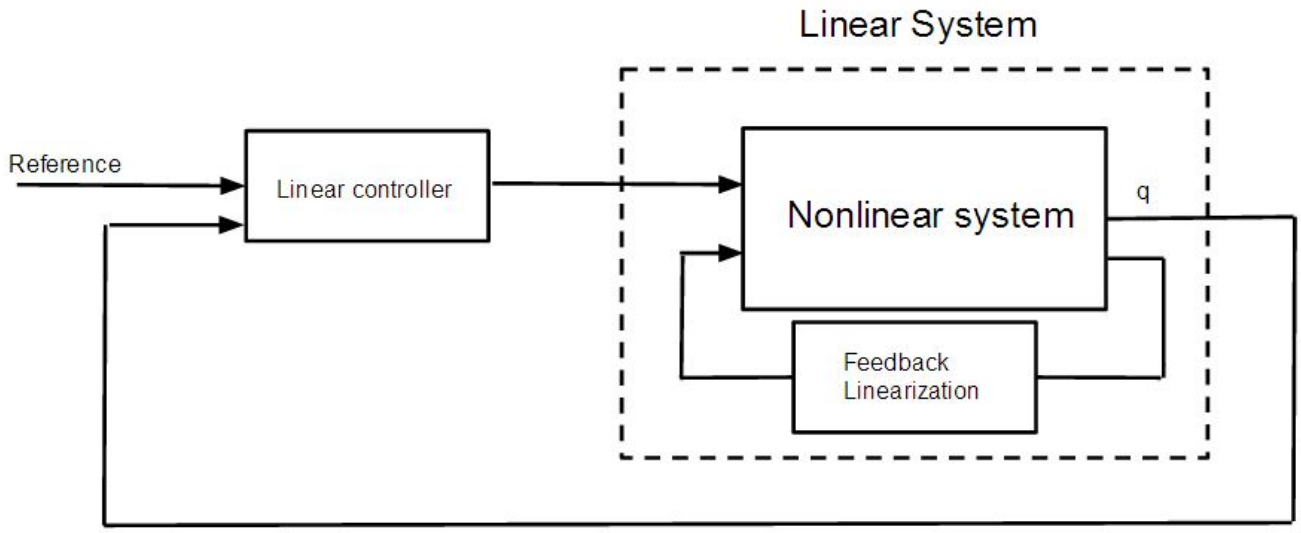


Figure 4.20: Illustration of feedback linearization in a nonlinear system

Based on the theory of the acrobat gymnastic presented above and the results of the controller performance of the controllers designed by Lai et al. [11], Kobayashi et al. [9] and Andersen et al. [2] a new controller structure for the limited Acrobot system representation can be developed and investigated. When controlling an actuated nonlinear system, a feedback linearizing inner controller, which cancels the nonlinear terms of the dynamical system, can be applied. From Khalil [8], the nonlinear system in Eq. 3.18 can be presented as:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u} \tag{4.26a}$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) \tag{4.26b}$$

The goal is to develop a state feedback control, $\boldsymbol{u}$:

$$\boldsymbol{u} = \alpha(\boldsymbol{x}) + \beta(\boldsymbol{x})\,v$$

where $\alpha(\boldsymbol{x})$ is the function canceling the nonlinearities in the system, and $\beta(\boldsymbol{x})$ is the function giving the new input v to the system. Figure 4.20 shows how the inner loop feedback linearization gives a linear system where an outer loop linear controller can be designed. As the bicycle system is underactuated, it is not feedback linearizable [1] due to the fact that the actuator can not create instantaneous acceleration in arbitrary system angle, as mentioned in the introduction of Chapter 3. Thus the actuated subsystem of the underactuated manipulator system can be linearized by use of partial feedback linearization, as Spong et al. [22]. The goal is to simplify the nonlinearities of the system by use of the

[1] The set $\{\boldsymbol{g}, \boldsymbol{ad_f}(\boldsymbol{g}), \boldsymbol{ad_f}^2(\boldsymbol{g})\}$ is involutive when the term $\boldsymbol{ad_f}^k(\boldsymbol{g})$ denotes the iterative Lie Bracket $\left[\boldsymbol{f}, \boldsymbol{ad_f}^{k-1}(\boldsymbol{g})\right]$, where the Lie Bracket of $\boldsymbol{f}$ and $\boldsymbol{g}$ is defined as: $[\boldsymbol{f},\boldsymbol{g}] = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\boldsymbol{f} - \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\boldsymbol{g}$, see Khalil [8]

actuated joint, creating a feedback linearized subsystem coupled with the remaining non-linearized terms of the original system representation. Due to the simplified nonlinearities, a less complicated controller can be designed in the complete system control. The state space representation presented in Section 3.8.2 and 3.9.2 is derived based on the theory of partial feedback lineraization of the equations of motion on matrix form by Spong in [18], [19], [20] and [17]. From Spong [20] the partial feedback linearized system with matrix elements of Eq. 3.17 can be written as:

$$m_{11}\ddot{\theta}_b + m_{12}\ddot{\theta}_p + c_1 + g_1 = 0 \tag{4.27a}$$

$$m_{12}\ddot{\theta}_b + m_{22}\ddot{\theta}_p + c_2 + g_2 = \tau \tag{4.27b}$$

where $c_1 = c_{11}\dot{\theta}_b + c_{12}\dot{\theta}_p$ and $c_2 = c_{21}\dot{\theta}_b$ of Eq 3.8. By rearranging Eq. 4.27a it can be solved for $\ddot{\theta}_b$:

$$\ddot{\theta}_b = -m_{11}^{-1}\left(m_{12}\ddot{\theta}_p + c_1 + g_1\right) \tag{4.28}$$

and substitution of Eq. 4.28 into Eq. 4.27b obtains:

$$\overline{m}_{22}\ddot{\theta}_p + \overline{c}_2 + \overline{g}_2 = \tau \tag{4.29}$$

where the terms of $\overline{m}_{22}$, $\overline{c}_2$ and $\overline{g}_2$ are given by:

$$\overline{m}_{22} = m_{22} - m_{12}m_{11}^{-1}m_{12}$$

$$\overline{c}_2 = c_2 - m_{12}m_{11}^{-1}c_1$$

$$\overline{g}_2 = g_2 - m_{12}m_{11}^{-1}g_1$$

The partial feedback linearizing controller can thereby be defined for Eq. 4.29:

$$\tau = \overline{m}_{22}u + \overline{c}_2 + \overline{g}_2 \tag{4.31}$$

where $u \in R^n$ is an additional control input, giving the complete system representation:

$$m_{11}\ddot{\theta}_b + c_1 + g_1 = -m_{12}u \tag{4.32a}$$

$$\ddot{\theta}_p = u \tag{4.32b}$$

Thus the acceleration of the inverted pendulum angle θ_p is set equal to the controller output, u, giving a linear second order system in Eq. 4.32b. By applying the partial feedback linearizing controller on the bicycle system, the nonlinearities are simplified and the actuated joint angle, θ_p, is directly cou-

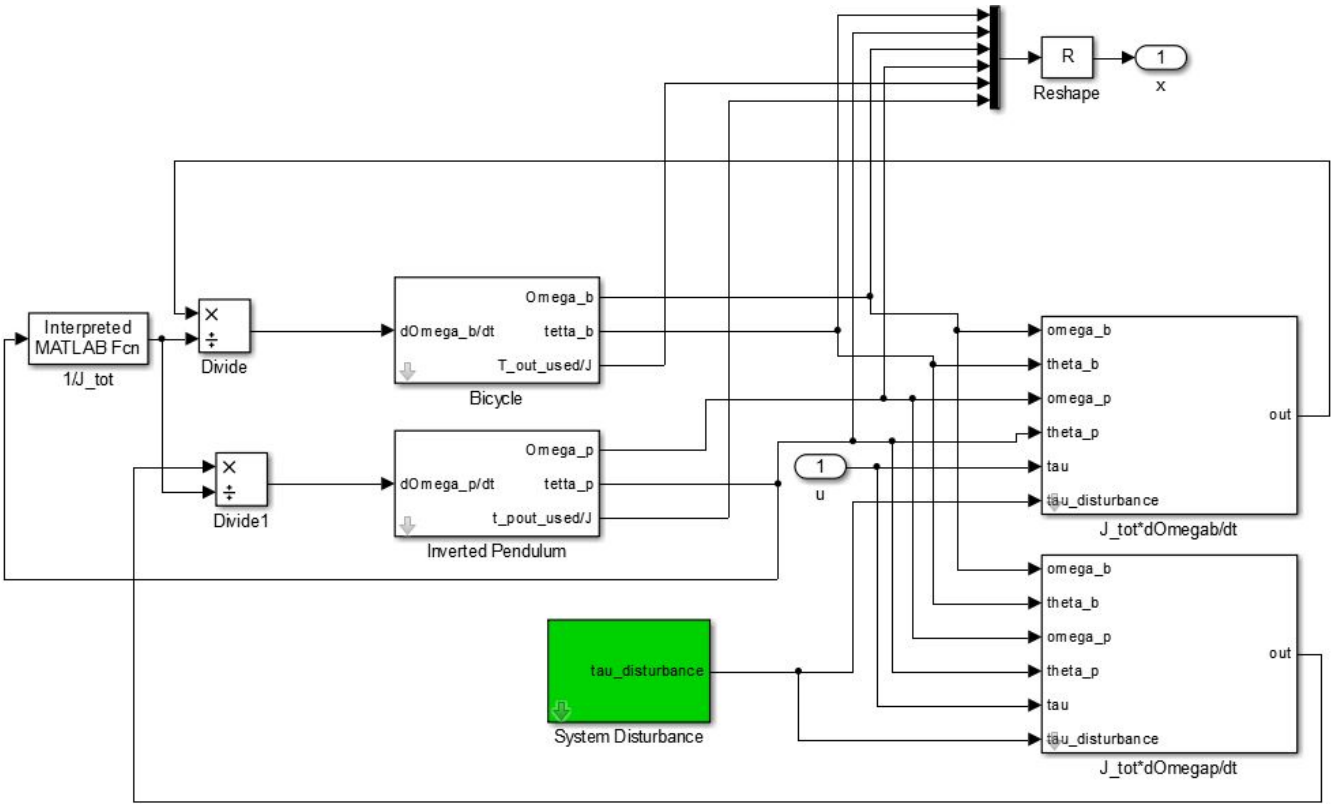


Figure 4.21: Implemented State Space Representation of Bicycle System, Eq. 3.20, in Simulink®

pled with the controller output, as the inverted pendulum acceleration, $\ddot{\theta}_p$, is equal to the controller output, u. This implies that the actuated system angle, θ_p, is utilized to move the bicycle angle, θ_b, towards its upright position as it simultaneously is controlled towards zero. Note that the system representation in Eq. 4.32 is equivalent to the state space representation in Eq. 3.22, with control torque in Eq. 4.31 equivalent to Eq. 3.21 and angular velocity Ω equivalent to $\dot{\theta}$ in Eq. 4.27. To improve the readability for the system analysis, the author will further on present the equations of motion as a set of differential equations, equivalent to the state space representation in Eq. 3.22 with Eq. 3.21. Thus a new system model was implemented in Simulink®, as shown in Figure 4.21. With the state space representation given in Eq. 3.20, the categorizing structure is utilized to analyze the inertial, centrifugal, Coriolis and gravity related torques in the system time response. The blocks of "*J_tot*dOmegab/dt*" and "*J_tot*dOmegap/dt*" is the right hand side of the differential equations in Eq. 3.20a and 3.20b, respectively. The implementation of the sub-blocks are shown in Figure B.1 and B.2 in Appendix B.1. The blocks "*Bicycle*" and "*Inverted Pendulum*" are the integrator structure with logical limitation, presented in Section 3.6 and illustrated in Figure 3.5.

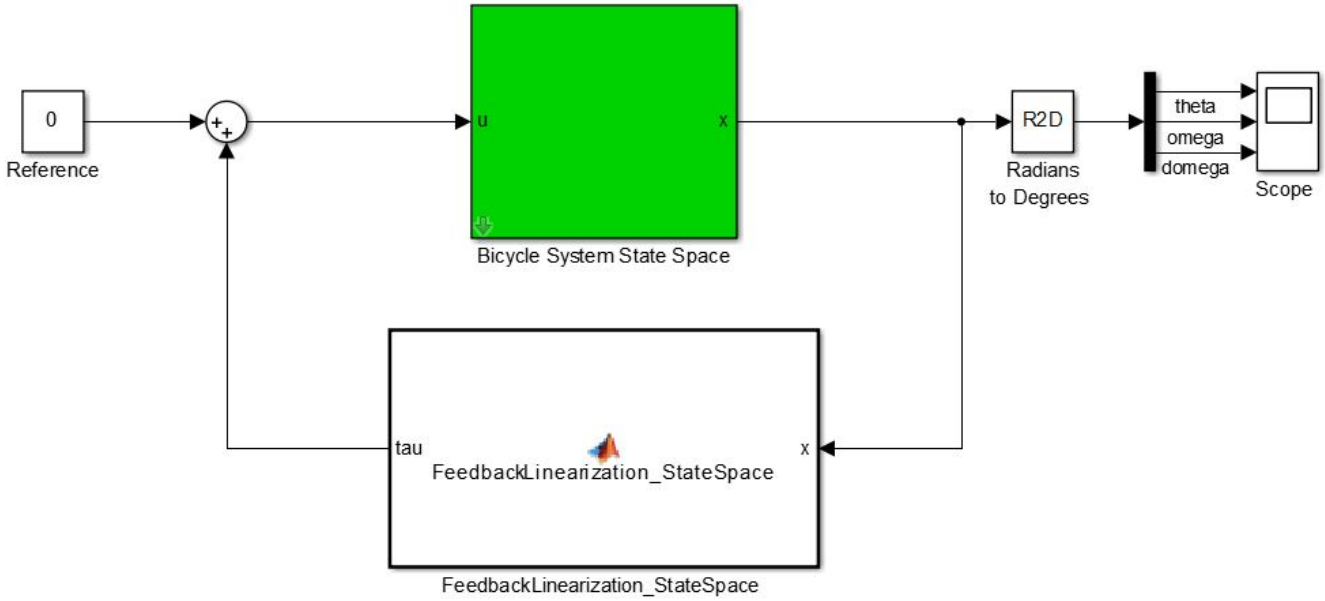


Figure 4.22: Implemented State Space Representation of Bicycle System in Simulink®

Figure 4.22 shows the complete feedback linearized state space representation of the bicycle system implemented in Simulink®, where the sub-block "*FeedbackLinearization_StateSpace*" is the feedback linearizing controller implemented with code in Matlab®, as shown in Section 4.3.4. The controller given in Eq. 3.21 can be presented as:

$$\tau = \frac{J_{tot}u - T_{pc} - T_{pg}}{K_{p\tau}} \tag{4.33}$$

The torque terms T_{pc}, T_{pg} and $T_{p\tau}$ are given by the differential equations in Eq. 3.20a and 3.20b presented as:

$$J_{tot}\frac{d\Omega_b}{dt} = T_{bc} + T_{bg} + T_{b\tau} + T_{bd} \tag{4.34a}$$

$$J_{tot}\frac{d\Omega_p}{dt} = T_{pc} + T_{pg} + T_{p\tau} + T_{pd} \tag{4.34b}$$

The terms T_{bc} and T_{pc}, T_{bg} and T_{pg}, $T_{b\tau}$ and $T_{p\tau}$ and T_{bd} and T_{pd} represent the Coriolis and centrifugal, gravity, actuator related torques of the bicycle and inverted pendulum and disturbance torque due to perturbation, respectively:

$$
\begin{aligned}
T_{bc} &= m_p r_p l sin\,\theta_p \left((\Omega_b + \Omega_p)^2 + \frac{l}{r_p} cos\,\theta_p\, \Omega_b^2 \right) \\
T_{bg} &= g m_p l \left(\left(1 + \frac{m_b r_b}{m_p l} \right) sin\,\theta_b - cos\,\theta_p\, sin(\theta_b + \theta_p) \right) \\
T_{b\tau} &= -\left(1 + \frac{l}{r_p} cos\,\theta_p \right) \tau \\
T_{bd} &= \tau_d
\end{aligned}
$$

$$
\begin{aligned}
T_{pc} &= -m_p r_p l sin\,\theta_p \left((\Omega_b + \Omega_p)^2 + \left(\frac{m_b r_b^2}{m_p r_p^2} + \left(\frac{l}{r_p} \right)^2 \right) \Omega_b^2 + \frac{l}{r_p} cos\,\theta_p (\Omega_b^2 + (\Omega_b + \Omega_p)^2) \right) \\
T_{pg} &= -g m_p l \left(\left(1 + \frac{m_b r_b}{m_p l} \right) \left(1 + \frac{l}{r_p} cos\,\theta_p \right) sin\,\theta_b - \left(\frac{m_b r_b^2}{m_p r_p l} + \frac{l}{r_p} + cos\,\theta_p \right) sin(\theta_b + \theta_p) \right) \\
T_{p\tau} &= K_{p\tau} \tau = \left(1 + \frac{m_b r_b^2}{m_p r_p^2} + \left(\frac{l}{r_p} \right)^2 + 2 \frac{l}{r_p} cos\,\theta_p \right) \tau \\
T_{pd} &= -\left(1 + \frac{l}{r_p} cos\,\theta_p \right) \tau_d
\end{aligned}
$$

By use of the partial feedback linearizing controller in Eq. 4.33 the control input, u, is utilized to achieve the controller requirements of the bicycle system. Note that the perturbation torque, τ_d, is set to zero in simulation of the bicycle system without disturbance.

PD-Controller

A linear state feedback PD-structured control input is utilized to create oscillation of the inverted pendulum angle, θ_p, and move the bicycle tilt angle towards the upright equilibrium of $(\theta_b,\ \theta_p) = (0°, 0°)$, as the equation of $\ddot{\theta}_p$ in Eq. 4.32b is a linear second order system with u_0 as angular reference input:

$$\begin{aligned} u &= -k_p\theta_p - k_d\dot{\theta}_p + u_0 \\ &= -k_p\theta_p - k_d\Omega_p + u_0 \end{aligned} \tag{4.35}$$

With the PD-controller structure in the control input, u, the gains k_p and k_d can be chosen to achieve oscillatory behavior of the inverted pendulum angle, θ_p. The idea is to utilize the proportional gain, k_p, to accelerate the inverted pendulum and move the bicycle angle, θ_b towards the upright equilibrium. The derivative gain, k_d, is utilized to obtain a desired damping of the system oscillations to create a controlled counteracting torque from the inverted pendulum onto the bicycle. Unlike the energy based swing-up controller presented in Section 4.1 and 4.2, the feedback linearizing controller is able to utilize the inverted pendulum angle instead of the bicycle angle, to create system energy. Note that the controller does not utilize the states θ_b and Ω_b, which implies that the controller does not have information of the bicycle angle. The goal is to utilize $\ddot{\theta}_p$ from the PD-controller to control Ω_b such that $\theta_b = 0$.

When the control input, u, is designed, the reference input, u_0, has to be designed. In [20] Spong utilized the saturation function, to give the steady state reference input, whereas in [18] and [19] Spong utilized the "*atan*" function give the steady state reference input. The design of the reference input u_0 is presented in Section 4.3.2 with the simulation results. The transfer function from θ_p to u is given as:

$$\theta_p(s) = \frac{u_0(s)}{s^2 + k_d s + k_p} = \frac{u_0(s)}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{4.36}$$

Thereby the proportional gain, k_p, is equal the undamped resonance frequency, ω, squared. The derivative gain, k_d, is given by the proportional gain and relative damping factor, ζ:

$$k_d = 2\zeta\omega_0 = 2\zeta\sqrt{k_p}$$

Throughout the thesis, the proportional gain, ω_0^2, and the term of 2ζ, will be tuned for the PD-controller. Increased resonance frequency will give larger bandwidth, thereby a faster system response. See Egeland et al.[3] Chapter 4.7.2 for more information. Note that the steady state value of u_0 has to be zero to achieve $\theta_p = 0$ in steady state.

State Space LQR Balance Control

As for the controllers by Lai et al. and Kobayashi et al. in Section 4.1 and 4.2, an LQR controller is utilized for balance control. With the linearized system matrices of Eq. 3.28, the Matlab® function utilized in Eq. 4.15, the state feedback gain matrix, K, is equivalent to the previous LQR controllers presented:

$$[K, P, EIG] = lqr(A, B, Q, R) \tag{4.37a}$$

$$\boldsymbol{K} = \begin{bmatrix} -975,63 & -57,24 & -227,34 & -19,74 \end{bmatrix} \tag{4.37b}$$

With this feedback controller, the LQR controller is equivalent to the controller derived in Section 4.1.3 and 4.2.2, as the controllers are derived from system linearization around the upright unstable equilibrium of (θ_b, θ_p) = (0, 0) and weight matrices as given in Eq. 4.14. With the state feedback controller $u = -\boldsymbol{K}x$, the time response of the nonlinear system with initial positions (θ_b, θ_p) = ($-0.6°$, $1.2°$) is shown in Figure 4.4. This feedback controller places the eigenvalues of the nonlinear dynamical system into the left half plane, giving a closed loop stable system. With the Matlab® function "*eig*" and the system matrices A and B in Eq. 3.28, the eigenvalues are given as:

$$eig(A - BK) = \begin{bmatrix} -10.99 \\ -8.09 \\ -4.20 \\ -4.16 \end{bmatrix} \tag{4.38}$$

Partially Feedback Linearized LQR Balance Control

With the linearization of the partially feedback linearized state space representation, as presented in Section 3.9.2, an LQR balance controller is designed based on the system matrices linearized around the upright position of the bicycle system. With linearization around the upright equilibrium of $(\theta_b, \theta_p) = (0°, 0°)$, the linear system matrices in Eq. 3.32a and 3.32b is utilized in the Matlab® function from Section 4.1.3. Through simulation of the nonlinear system with the LQR controller, the wight matrices Q and R were tuned to get the LQR region of attraction as large as possible:

$$\boldsymbol{Q} = \begin{bmatrix} 1000 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 \\ 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 1000 \end{bmatrix} \tag{4.39a}$$

$$R = 1 \tag{4.39b}$$

The resulting state feedback gain matrix, $\boldsymbol{K}$, is found by the Matlab® function "*lqr*" with the linear system matrices A and B of the linearized partially feedback system and the weight matrices Q and R in Eq. 4.39:

$$[K, P, EIG] = lqr(A, B, Q, R) \tag{4.40a}$$

$$\boldsymbol{K} = \begin{bmatrix} -918,05 & -57,02 & -215,48 & -17,39 \end{bmatrix} \tag{4.40b}$$

Note that the tuned weight matrix $\boldsymbol{Q}$ gives a feedback gain matrix, $\boldsymbol{K}$, approximately equal to the gain matrix of the LQR designed for the linearization of the state space representation in Eq. 3.18, presented in Eq. 4.37. And that an LQR gain matrix is equivalent to a PD-controller where $u_0 = -k_{pb}\theta_b - k_{db}\Omega_b$, which implies that the controller utilizes the angle and angular velocity of the bicycle in a linear state feedback controller.

The goal of having the gain matrices as similar as possible is to obtain the functionality of angle measurement noise affecting the system behavior uniformly for both LQR controllers. With nominal actuation torque of 19.44[Nm], as stated in Section 2.5.1, and feedback gains coupled with the system states and their respective measurement noise, the controller gains must be taken into account. With a proportional gain of $-918,05$ in the K matrix, the angle measurement of θ_b can have an error of $0.063°$ resulting in a torque error of 1[Nm], approximately 5% of the nominal torque of the system actuator. When the system is implemented, the measurements should be filtered, but nevertheless the system gains must be taken into account to ensure that the controller gains are not physically unobtainable. Through simulation the largest LQR region of attraction, of this LQR controller designed by the linear system matrices of the linearization of the partially feedback linearized system in Eq. 3.22, is verified:

$$(\theta_b^i, \theta_p^i) = \left(\frac{0.8}{180}\pi, \frac{1.4}{180}\pi\right)[rad] \tag{4.41}$$

Figure 4.23 illustrates how the controller implementation, presented in Section 4.3.4, stabilizes the nonlinear system with initial angular positions at the maximum angles in the LQR region of attraction. The figure also shows how the partially feedback linearized LQR controller is slower than the LQR controller designed on linearization of the state space representation, presented above and illustrated in Figure 4.4. As the LQR region of attraction of the partially feedback linearized LQR controller is not significantly improved, the choice of which LQR controller to be applied is not critical. With the Matlab® function "*eig*" and the system matrices A and B in Eq. 3.9.2, the eigenvalues are given as:

$$eig(A-BK) = \begin{bmatrix} -31.767 \\ -1.001 \\ -4.253 + j0.088 \\ -4.253 - j0.088 \end{bmatrix} \tag{4.42}$$

With the poles further into the left half plane and relative damping, the system is able to stabilize from angles further away from the upright unstable equilibrium, as given in Eq. 4.41, than the previous LQR controller.

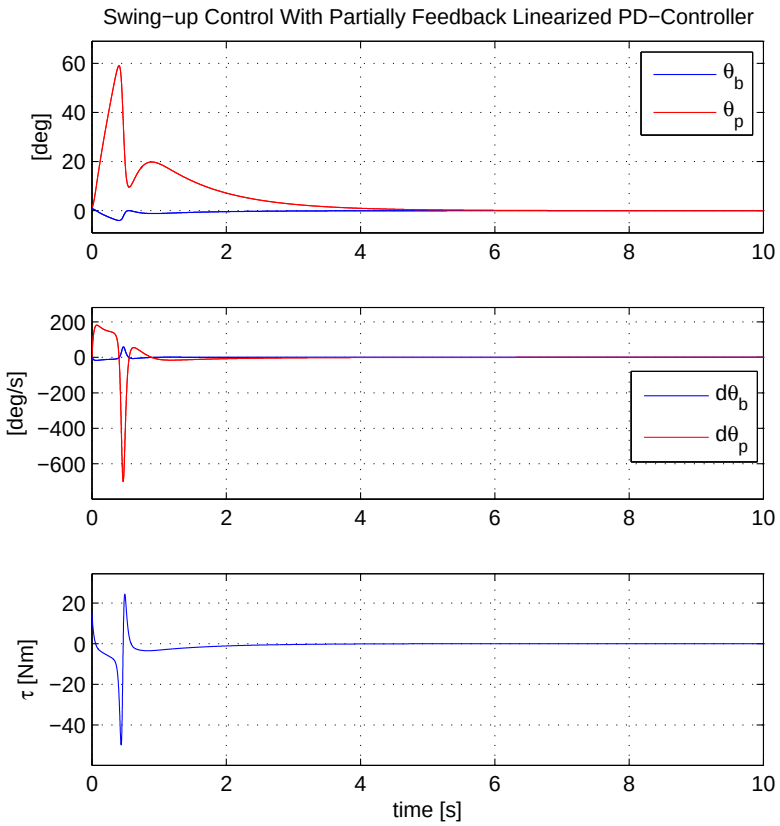


Figure 4.23: Time Response of Partially Feedback Linearized LQR Balance Control, $(\theta_b^{init}, \theta_p^{init}) = (0.8°, 1.4°)$

4.3.2 Controller Analysis

PD-Controller with $u_0 = 0$

With the state space model presented in Eq. 4.34 and the partial feedback linearizing controller in Eq. 4.33, the additional PD-controller, u, in Eq. 4.35 has to be tuned. By choosing the control parameters, k_p and k_d, and angular reference, u_0, the controller performance can be analyzed through simulation. As the partial feedback linearization gives a linear second order system as in Eq. 4.32, the PD-controller, u, is utilized to control the inverted pendulum angle, θ_p, and move the bicycle angle, θ_b, towards its upright position. Note that the controller in Eq. 4.35 only utilizes the states of θ_p and Ω_p, which implies that the controller is designed to stabilize the angle of θ_p around a desired angular value of zero, independently of the bicycle angle, θ_b. Thereby the controller gains of the PD-controller are tuned to utilize the angular oscillation of θ_p and move the bicycle towards its upright position as a result of the counteracting torque. At first the reference input u_0 is set to zero, where Figure 4.24 shows the time response of the feedback linearized system, with initial angle positions at the angular

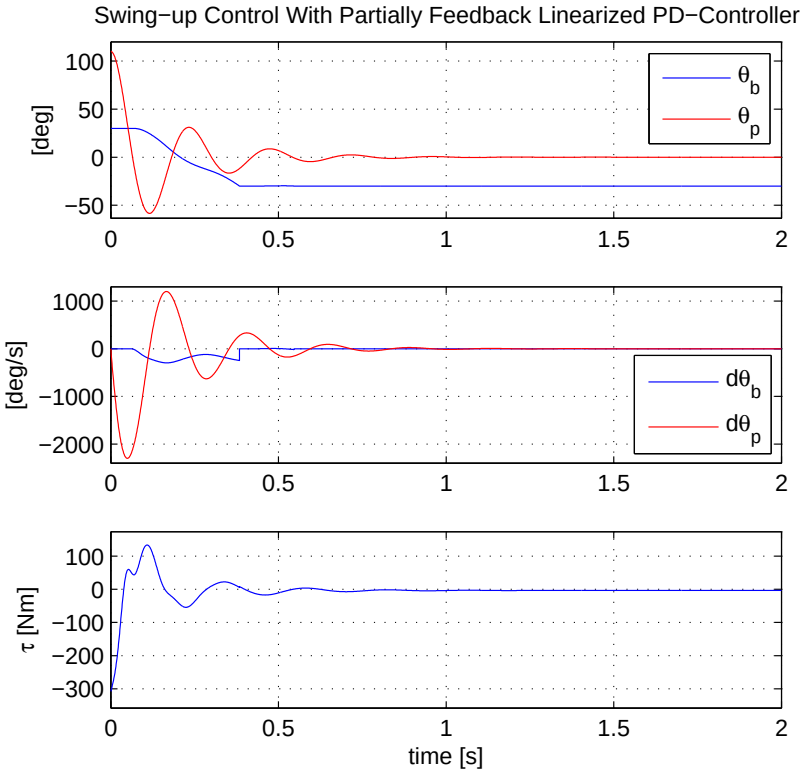


Figure 4.24: Time Response of Partially Feedback Linearized PD-Controller, $k_p = 700$, $(\theta_b^{init}, \theta_p^{init}) = (30°, 110°)$

limits, $(\theta_b^{init}, \theta_p^{init}) = (30°, 110°)$ and control parameters:

$$k_p = 700 \tag{4.43a}$$

$$k_d = 0.2 \cdot 2\sqrt{k_p} \tag{4.43b}$$

$$u_0 = 0 \tag{4.43c}$$

The simulation illustrates how the control torque oscillates the angle of θ_p to move the bicycle angle, θ_b. First the inverted pendulum is accelerated in negative direction to approximately $-58°$, before it is accelerated in positive direction. As the control torque accelerates the angle θ_p in positive direction the counteracting torque applied onto the bicycle accelerates the angle of θ_b i negative direction. As shown in Figure 4.24, the angle of θ_b is moved from the positive angular limit of 30° to the negative limit of $-30°$. Thereby the simulation verifies that the controller is able to move the bicycle tilt angle towards its upright position by use of the inverted pendulum when the initial angle positions are at the limits. Note that the control torque, τ, is more active throughout the system response compared with the energy based swing-up controllers by Lai et al. in Figure 4.7, as the inverted pendulum is utilized to increase system energy.

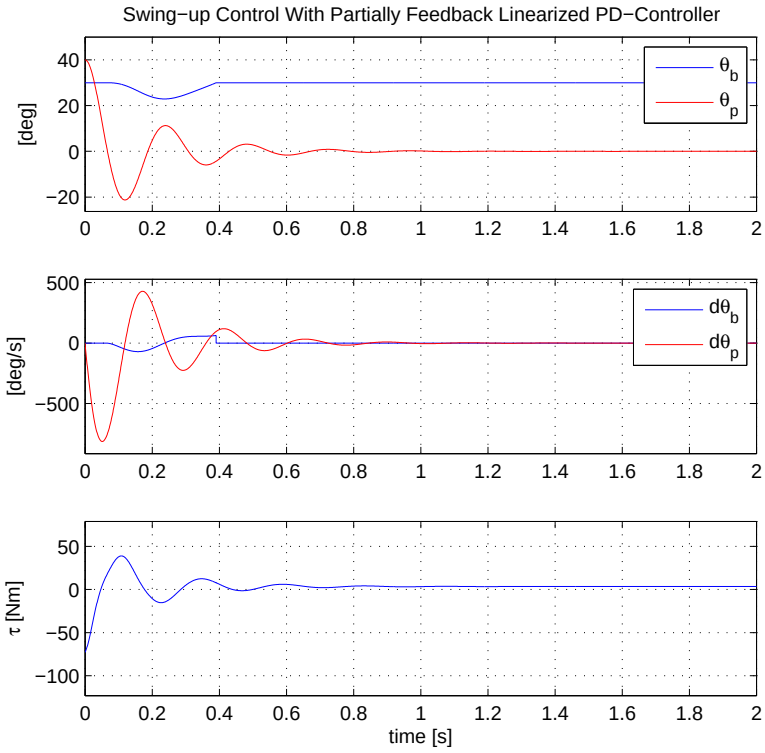


Figure 4.25: Time Response of Partially Feedback Linearized PD-Controller, $k_p = 700$, $(\theta_b^{init}, \theta_p^{init}) = (30°, 40°)$

As mentioned, the PD-controller only utilizes the states θ_p and Ω_p, which implies that the controller is stabilized when the angle of θ_p reaches the desired angle of zero. Note that when θ_p and Ω_p are stabilized the output of the PD-controller is zero, thus no input for controlling the angular velocity of the bicycle is generated. Thereby the controller, with the presented gains, is unable to apply counteracting torque from the inverted pendulum onto the bicycle and move the angle of θ_b towards the upright unstable equilibrium, when the initial angular position of the inverted pendulum is too close to zero. As shown in Figure 4.25, the presented gains are unable to move the bicycle towards its upright position, with initial angles of $(\theta_b, \theta_p) = (30°, 40°)$. As the bicycle angle is dependent of the counteracting torque applied from the inverted pendulum, it is desirable to have a controller which is able to achieve this performance from any initial angular position. The figure shows the time response of the system with the PD-controller simulated in Figure 4.24, where the parameters are tuned for initial angular position at the limits: $(\theta_b, \theta_p) = (30°, 110°)$. Thereby the proportional gain, k_p, and derivative gain, k_d, results in a controller which is unable to apply enough counteracting torque onto the bicycle, and move it towards its upright position. The proportional gain is unable to give enough acceleration to the inverted pendulum whereas the derivative gain damps the oscillations too much, resulting in a controller which is unable to create a desired counteracting torque onto the bicycle. Thus the parameters utilized in these simulations are not applicable for every initial angle position in the workspace.

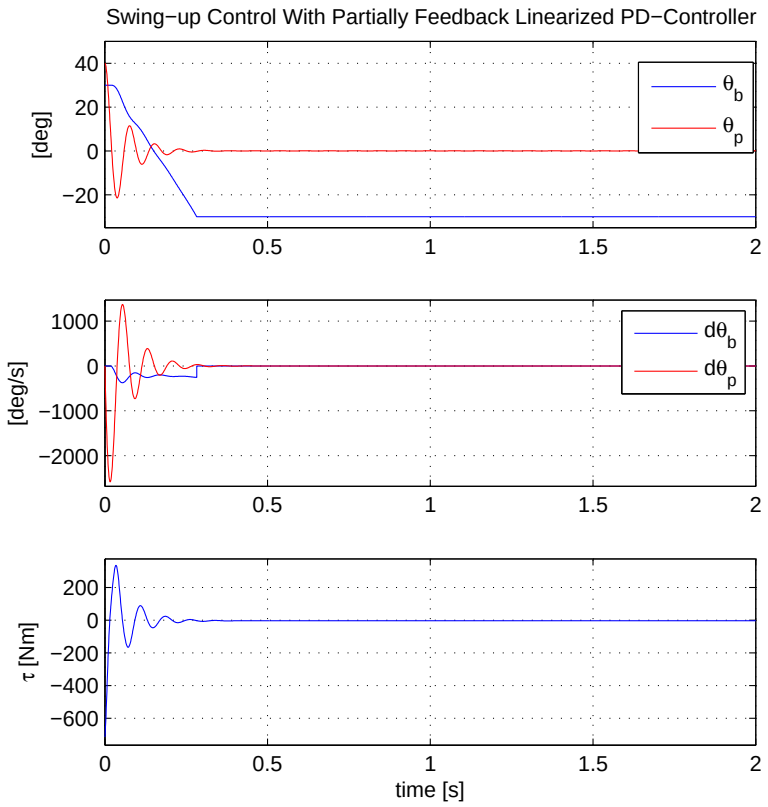


Figure 4.26: Time Response of Partially Feedback Linearized PD-Controller, $k_p = 7000$, $(\theta_b^{init}, \theta_p^{init}) = (30°, 40°)$

Even though the PD-controller is unable to move the bicycle angle towards the upright position, Figure 4.25 shows that the resulting control torque is within the maximum torque which can be applied from the DC motor actuator.

Figure 4.26 shows the time response of the system with initial positions closer to the upright equilibrium, $(\theta_b^{init}, \theta_p^{init}) = (30°, 40°)$, and new controller parameters:

$$k_p = 7000 \tag{4.44a}$$

$$k_d = 0.2 \cdot 2\sqrt{k_p} \tag{4.44b}$$

$$u_0 = 0 \tag{4.44c}$$

The time response illustrates how the increased proportional gain is able to move the bicycle angle, θ_b, from the positive to the negative limit unlike the simulation with the system parameters in Figure 4.25. This shows how the linear PD-controller performance depends on the controller parameters when the initial positions are varying. By adjusting the controller parameters the acceleration of the inverted pendulum becomes larger, as shown in Figure 4.26, resulting in increased counteracting torque onto the bicycle. Thus it is shown that the PD-controller depends on the controller parame-

ters and the initial angular positions, when the inverted pendulum is utilized to create counteracting torque onto the bicycle. The figure also illustrates how the control torque exceeds the maximum possible torque for the DC motor, which gives a challenge in the design of a stabilization controller. Also there is no guarantee that the controller is able to control the bicycle towards the upright position as it is unable to detect the angle of θ_b, when the states of θ_b and Ω_b are not utilized in the state feedback PD-controller. Thereby the controller is not suitable to control the bicycle angle towards its upright position as it simultaneously controls the angle of θ_p towards zero, from arbitrary initial positions in the workspace.

The goal is to utilize the controller when the system states moves outside the LQR region of attraction due to perturbation. With the PD-controller, the fixed system gains the controller is only able to move the bicycle towards the upright equilibrium from a smaller subset of initial positions in the system workspace. If the initial positions are too close to the desired angles the proportional gain is unable to create enough acceleration of θ_p to give the desired counteracting torque onto the bicycle and move it towards the upright position of $\theta_b = 0$, as illustrated in Figure 4.25. The time response presented in Figure 4.24 shows how the swing-up controller is able to move the system states from the limits up towards the upright equilibrium. The PD-controller stabilizes the angle of θ_p at its desired angle, while the bicycle angle θ_b falls to the negative angular limit. As the goal is to swing the system up towards the upright equilibrium and activate the LQR controller when the states reaches the LQR region of attraction, the angular velocities must be close to zero. As both LQR controllers in Eq. 4.37b and 4.40b are designed by linearization around the upright equilibrium with $\boldsymbol{x}^* = (\theta_b^*, \theta_p^*, \Omega_b^*, \Omega_p^*)^T = (0,0,0,0)^T$ and $u^* = 0$, the angular velocities Ω_b and Ω_p has to be close to zero to apply the linear balance controller. Thereby the PD-controller is tuned to get the system states close to zero simultaneously, to activate the LQR balance controller. As the PD-controller controller controls the angle of θ_p towards its desired angle independently of the bicycle angle θ_b, the controller parameters must be tuned to give the desired system response from the various initial positions. Through simulation, the controller parameters is tuned to obtain angular velocity close to zero when the system angles reaches the LQR region of attraction for initial values at the limit, $(\theta_b^{init}, \theta_p^{init}) = (30°, 110°)$:

$$k_p = 700 \tag{4.45a}$$

$$k_d = 0.4598 \cdot 2\sqrt{k_p} \tag{4.45b}$$

$$u_0 = 0 \tag{4.45c}$$

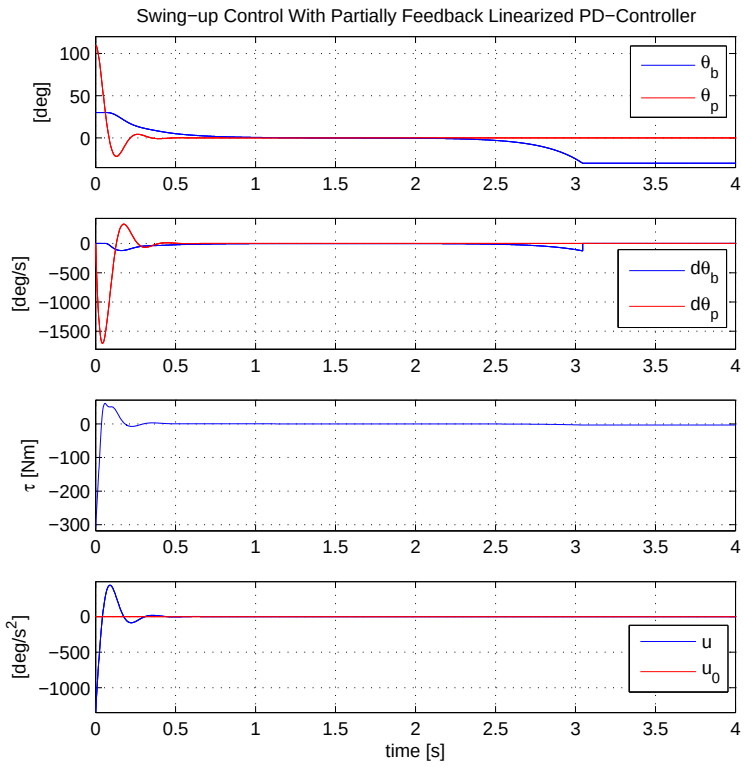


Figure 4.27: Time Response of Partially Feedback Linearized PD-Controller, $u_0 = 0$, $(\theta_b^{init}, \theta_p^{init}) = (30°, 110°)$

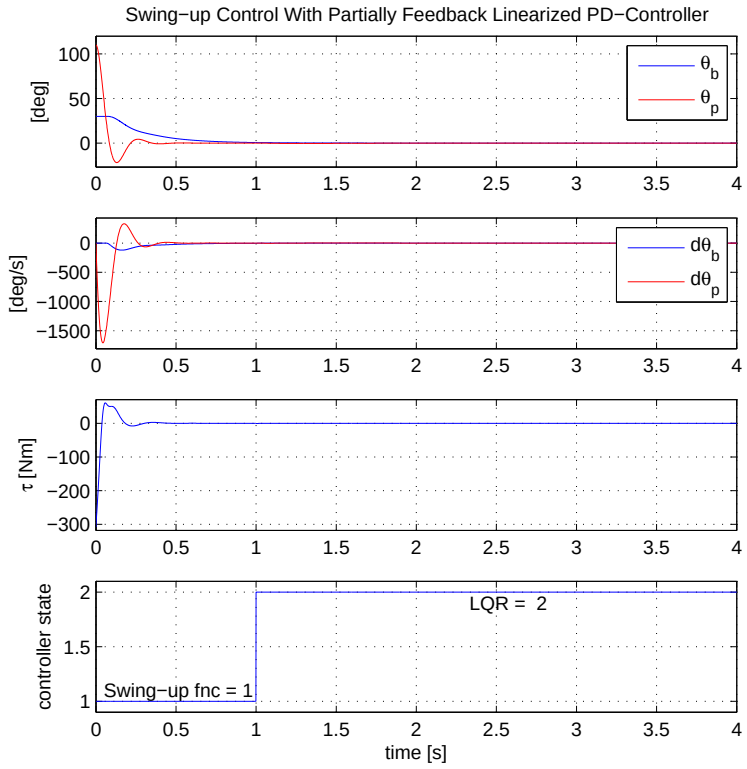


Figure 4.28: Time Response of PD-Controller Swing-Up and LQR Balance Control, $u_0 = 0$

Figure 4.27 shows how the PD-controller is able to move the bicycle system to the upright position, with angular velocities equal zero. As the PD-controller is unable to verify that the bicycle angle, θ_b, has reached the upright position the controller is unable to balance the system, and the bicycle falls to the negative angular limit. Thus the LQR controller can be activated for system balancing around the upright unstable equilibrium. Note that the angle of θ_p reaches zero before the bicycle angle reaches the upright position of $\theta_b = 0$. This implies that the counteracting torque applied from the positive acceleration of the inverted pendulum is enough to move the bicycle in negative direction towards the upright position after the first oscillation. As the PD-controller consist of damped oscillations, the amplitude of each oscillation of θ_p decreases throughout the time response. Thereby the controller must be able to move the bicycle towards its upright position after one oscillation, as shown in Figure 4.27. If the oscillations are too large, the positive acceleration will move the bicycle towards the upright position before the negative acceleration will push the bicycle in positive direction back towards the initial limit angle. As the property of zero angular velocity is obtained, the LQR balance controller is applied to the system controller. As the PD-controller is able to reach the LQR region of attraction of $(\theta_b, \theta_p) = (0.6°, 1.2°)$, the LQR controller presented in Eq. 4.37 selected due to more rapid system response. Figure 4.28 shows how the PD-controller is able to move both system angles within the LQR region of attraction while simultaneously controlling the angular velocities towards zero and activating the LQR balance controller. Thus the controller is able to move the system angles from the maximum excursion at the angular limits and up to the upright position, where the balance control is activated to stabilize the bicycle system around the upright unstable equilibrium of $(\theta_b, \theta_p) = (0°, 0°)$. The controller implementation is shown in Section 4.3.4.

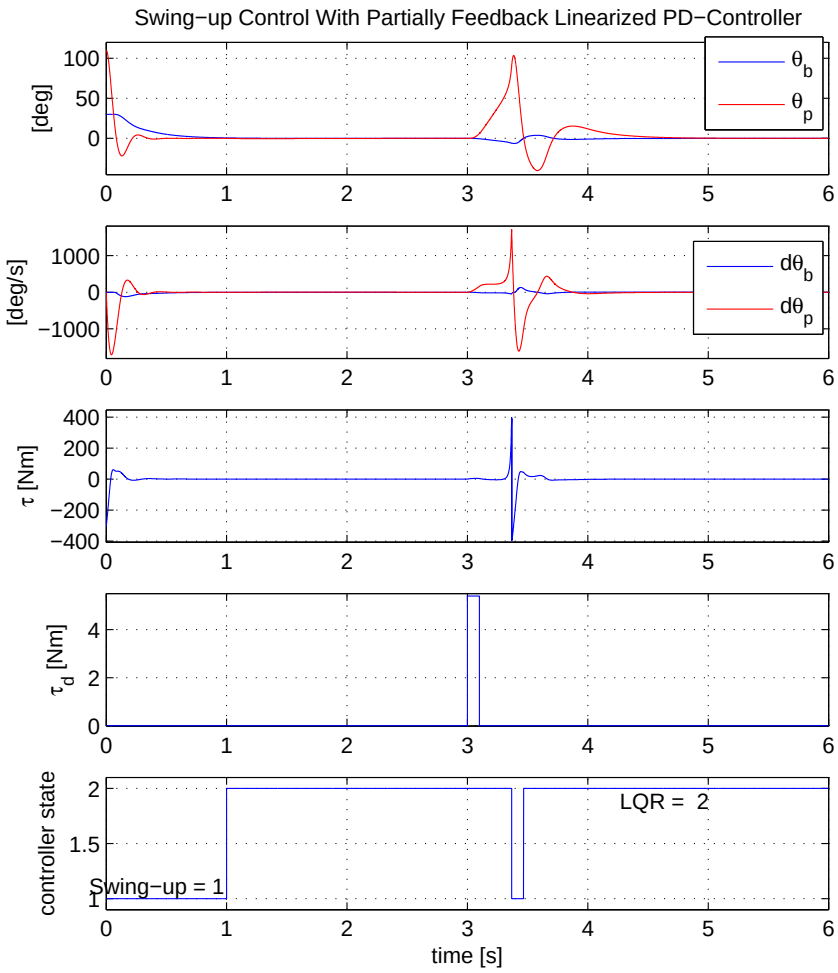


Figure 4.29: Time Response of PD-Controller Swing-Up and LQR Balance Control w/ $\tau_d = 5.4[\text{Nm}], u_0 = 0$, $\beta_1 = \frac{\pi}{6}$, $\beta_2 = \frac{\pi}{2}$

Further on the robustness of the LQR controller is tested. By applying a torque disturbance onto the system after the LQR balance control has stabilized the bicycle system, the time response is analyzed through simulation. A push onto the bicycle, performed by a person, is assumed to last in 1 [ms]. The LQR balance controller limits of β_1 and β_2, presented in Section 4.1.5 and 4.1.7, are tuned to enlarge the operating workspace of the LQR balance controller:

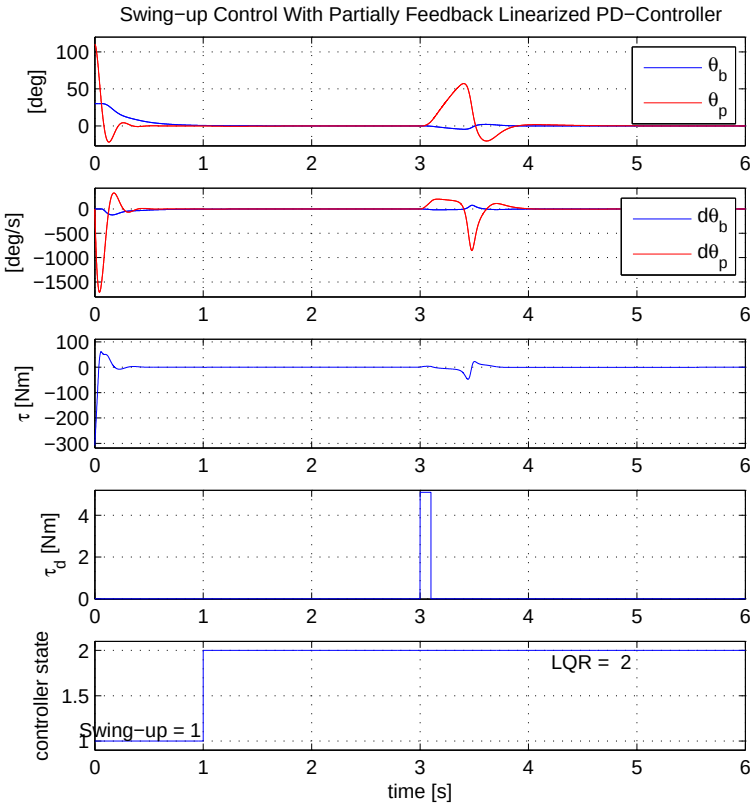
$$\beta_1 = \frac{\pi}{6}, \quad \beta_2 = \frac{\pi}{2} \tag{4.46}$$

Note that the limits of β_1 and β_2 correspond to a limitation of $\theta_b = \beta_1 = 30°$ and $\theta_p = \beta_2 = 90°$. Through simulation it is verified that the system is able to perform successful stabilization by use of the PD-controller and LQR balance controller, with LQR workspace limits in Eq. 4.46, as long as the perturbation torque is: $\tau_d \leq 5.4[\text{Nm}]$. Figure 4.29 shows the time response of the system with swing-up by the PD-controller with parameters from Eq. 4.43 and the LQR balance controller from Eq. 4.37. The time

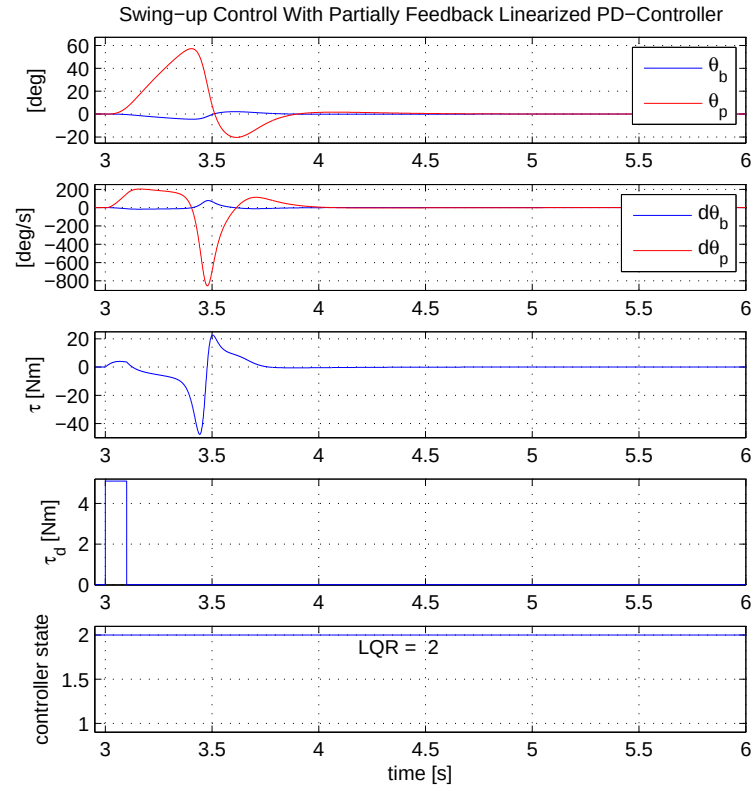
response illustrates how the perturbation torque moves the bicycle angle, θ_b, resulting in acceleration of the inverted pendulum angle, θ_p, to counteract the falling bicycle angle which is moved to a maximal negative excursion of $-6.2°$. When the angle of θ_p exceeds the maximum limit of β_2 for the LQR control, the PD-controller is applied. As the angle is close to $110°$ the system is able to utilize the PD-controller and control all the system states towards zero simultaneously. The controller accelerates θ_p to create counteracting torque onto the bicycle, to move both system angles within the LQR region of attraction, by use of the fixed controller parameters in Eq. 4.45. As both system angles are moved within the LQR region of attraction while the angular velocities Ω_b and Ω_p are controller towards zero, the LQR balance controller is applied to stabilize the bicycle system. The balance controller gives a maximum positive excursion of $4°$ on the bicycle angle, before it is moved towards zero. The figure also shows that the control torque exceeds the maximum torque which can be applied from the system actuator, presented in Section 2.5.1. Thereby the LQR balance controller limits, β_1 and β_2 are tuned further increase the operating workspace of the LQR balance controller to prevent exceeding control torque from the PD-controller:

$$\beta_1 = \frac{\pi}{6}, \quad \beta_2 = \frac{100}{180}\pi \tag{4.47}$$

This implies that the LQR workspace limitation of $\theta_b = \beta_1 = 30°$ and $\theta_p = \beta_2 = 100°$. Through simulation, shown in Figure 4.30(a), it is verified that the controller performs successful system stabilization by use the LQR balance controller, with LQR workspace limits of Eq. 4.47. When the perturbation torque is: $\tau_d \leq 5.1$[Nm], the system is able to utilize the LQR controller throughout the whole stabilization operation. Thereby the PD-controller is not utilized, and the control torque is kept within the maximum torque able to be applied from the DC motor actuator. Figure 4.30(b) is zoomed in when the perturbation occurs, and shows the acceleration of the inverted pendulum angle, θ_p, applied to create the counteracting torque onto the bicycle angle, θ_b, for system stabilization. The resulting counteracting torque from the inverted pendulum onto the bicycle, moves the angle of θ_b to the maximum negative excursion of $-4.5°$ before it is moved to $2°$ and further stabilized at the upright equilibrium of $0°$. The perturbation of 5.1[Nm] results in a counteracting controller torque of -47.5[Nm] before it is controlled to 22.5[Nm], which is approximately equal the nominal actuator torque, $\tau_{p,n} = 19.44$[Nm]. As these controller torques are within maximum torque able to be applied from the system actuator, which is 3-4 times the nominal torque $\tau_{p,n}$, the controller simulated in Figure 4.30(a) can be utilized on the real life bicycle system.



(a) Full time response with perturbation



(b) Zoomed LQR balance, when perturbation occur

Figure 4.30: Time Response of PD-Controller Swing-Up and LQR Balance Control w/ $\tau_d = 5.1[\text{Nm}], u_0 = 0, \beta_1 = \frac{\pi}{6}, \beta_2 = \frac{100}{180}\pi$

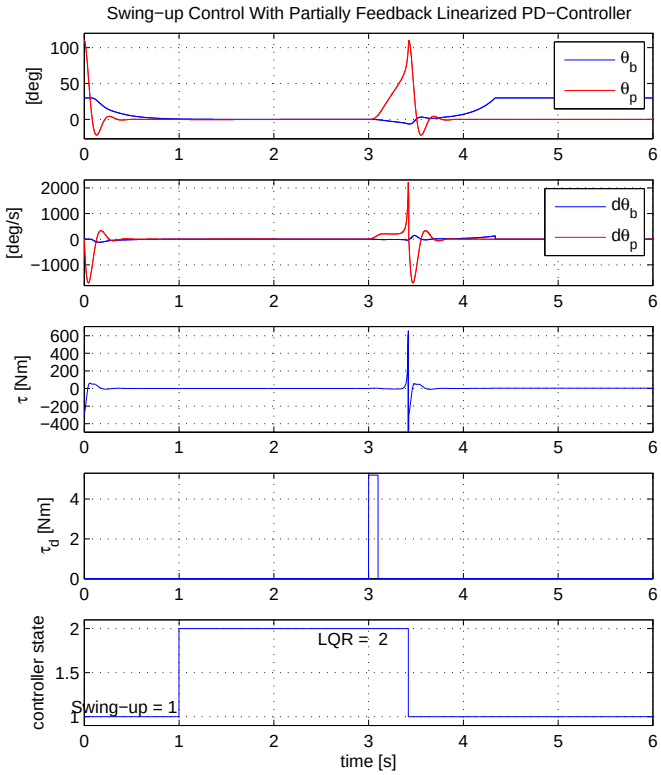


Figure 4.31: Time Response of PD-Controller Swing-Up and LQR Balance Control w/ $\tau_d = 5.2[\text{Nm}], u_0 = 0$, $\beta_1 = \frac{\pi}{6}$, $\beta_2 = \frac{100}{180}\pi$

As mention previously in this section, the parameters of the PD-controller are dependent of the initial angular positions and system parameters. Figure 4.31 shows how the controller is unable to stabilize the bicycle system when the perturbation torque becomes too large. With torque perturbation of $\tau_d = 5.2[\text{Nm}]$ and LQR workspace limits from Eq. 4.47, the system is moved outside of the workspace of the LQR balance controller. The PD-controller, with fixed controller parameters of Eq. 4.45, is unable to swing the bicycle system back into the LQR region of attraction, while simultaneously controlling the angular velocities toward zero. Thereby the bicycle angle falls to the angular limit of 30°, while the PD-controller stabilizes the angle of θ_p at zero. This verifies that the controller design does not guarantee that the PD-controller, with fixed controller parameters in Eq. 4.45, is able to move the system angles back into the LQR region of attraction when perturbation occur. Thus one must allow the bicycle system to fall closer to the angular limits to apply the PD-controller with fixed controller gains of Eq. 4.43, to successfully stabilize the system around the upright equilibrium, as shown in Figure 4.30(a). The swing-up PD-controller can be improved by implementing functionality which adjust the controller parameters dependent of the system states, to utilize $\ddot{\theta}_p$ from to control Ω_b such that $\theta_b = 0$. As the system is four dimensional, this approach is a very complicated and time consuming. Thereby the author will further investigate how the input reference, u_0, can be utilized to achieve the desired swing-up functionality, throughout the thesis.

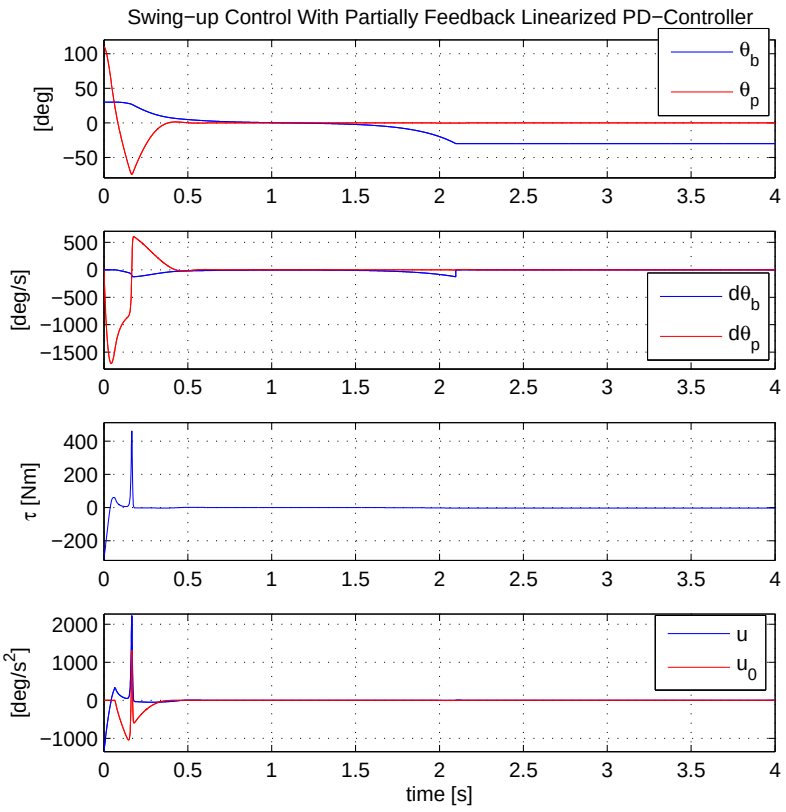


Figure 4.32: Time Response of Partially Feedback Linearized PD-Controller, $u_0 = sat$, $(\theta_b^{init}, \theta_p^{init}) = (30°, 110°)$

PD-Controller with u_0 as Saturation Function

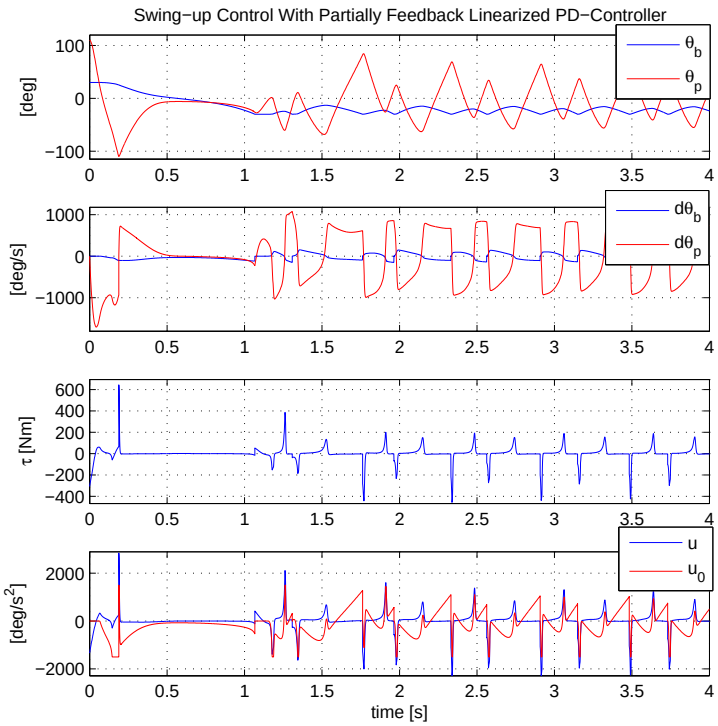
As mentioned, the PD-controller with controller gains in Eq. 4.43 is unable to create counteracting torque onto the falling bicycle angle when θ_p is close to zero. Thus the PD-controller with fixed controller gains from Eq. 4.43 does not guarantee that the wing-up controller is able to move the system angles back into the LQR region of attraction when large perturbations occur, as shown in Figure 4.31. Thereby the reference input, u_0, is further investigated to see if the controller is able to utilize the system states θ_b and Ω_b to obtain an improved system response. Spong [20] utilizes a saturation function, equivalent to the implemented Matlab® function "*sat2*"[2]:

$$u_0 = 97 \cdot sat2\left(E_z(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\Omega_b\right) \tag{4.48}$$

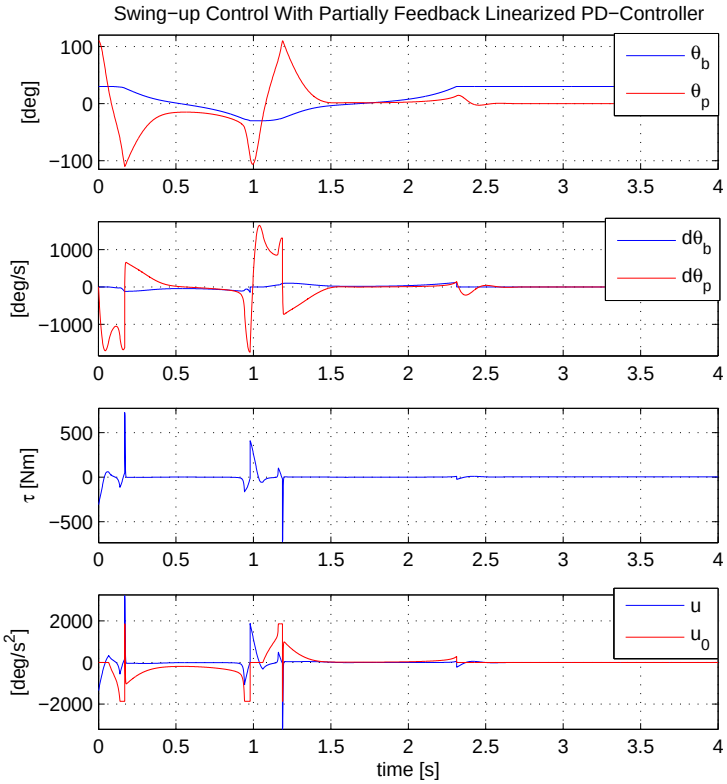
Where the energy function, $E_z(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = E(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) - E_0$, is defined in Eq. 4.21, with total system energy, $E(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$, and potential energy at the upright equilibrium, E_0. The function is multiplied with 97 to increase the magnitude of the reference input and give a larger impact in the controller output, u. This magnitude factor is tuned through simulation, to achieve angular velocity close to zero as the system angles moves towards LQR region of attraction, for further utilization of the LQR balance controller.

[2]Note that "*sat2*" is a modified version of the function "*sat*" utilized by the controller implementation in Section 4.1.7

Figure 4.32 shows the time response of the bicycle system, with u_0 from Eq. 4.48 and controller parameters of the PD-controller in Eq. 4.45. The figure shows how the controller is able to move the system states towards zero simultaneously, before the bicycle angle, θ_b, falls to the negative limit of $-30°$ when the balance control is deactivated. It is clear how the choice of u_0 results in a more aggressive controller, as the positive acceleration moves the angle of θ_p to approximately $-75°$, before the inverted pendulum is accelerated in negative direction and contributes with counteracting torque onto the bicycle, moving it towards the upright position. Note that the resulting control torque, τ, thereby exceeds the maximum torque possible to apply from the system actuator, presented in Section 2.5.1. By scaling the magnitude of the saturation function in Eq. 4.48 to 100 instead of 97, the controller performance becomes even more aggressive. Figure 4.33(a) illustrates how the inverted pendulum angle, θ_p is unable to stabilize around the upright position before the bicycle angle, θ_b, reaches its upright position. Thereby the bicycle angle falls to the negative angular limit of $-30°$. However, the choice of u_0 shows how the controller continuously tries to move the bicycle angle towards the upright position, without success. By increasing the magnitude factor to 124.022, the controller is able to move the bicycle angle from the negative angular limit up towards the upright position by accelerating the inverted pendulum from the negative to the positive angular limit, as shown in Figure 4.33(b). This is undesirable as it can ruin the system actuator when the inverted pendulum can hit objects mounted on the bicycle. Note that when the inverted pendulum is accelerated from the negative limit of $-110°$ to the positive limit of $110°$, the resulting torque becomes excessively large. Thereby this choice of the reference input magnitude is physical impossible to utilize with the given system actuator. The bicycle angle must be controlled by a counteracting torque from the inverted pendulum, as the PD-controller simulated in Figure 4.28, where the control torque does not exceed the maximum actuation torque given in Section 2.5.1.



(a) magnitude:100



(b) magnitude:124.022

Figure 4.33: Time Response of Partially Feedback Linearized PD-Controller, $u_0 = sat$, adjusted magnitude

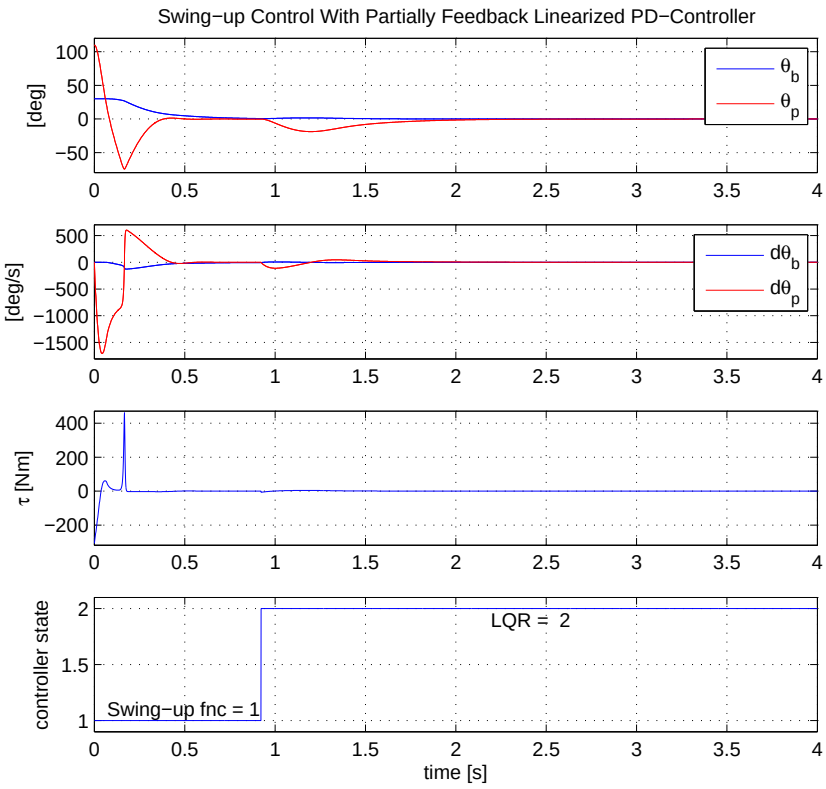


Figure 4.34: Time Response of PD-Controller Swing-Up and LQR Balance Control, $u_0 = sat$

By utilizing the reference input, u_0, given in Eq. 4.48, the system time response in Figure 4.34 shows how the swing-up and LQR balance controller is able to stabilize the system around the upright unstable equilibrium. As the swing-up controller is able to move both angles towards zero, while simultaneously controlling the angular velocities, Ω_b and Ω_p, towards zero, the LQR balance control can be activated. However, as the control torque, τ, exceeds the maximum torque possible to apply from the system actuator, the reference input, u_0, in Eq. 4.48 is not applicable for the real life bicycle system.

PD-Controller with u_0 as atan Function

In [18] and [19] Spong utilizes the "*atan*" function in the reference input, u_0. The idea is to utilize a reference input which includes the angular velocity, Ω_b, of the bicycle to investigate whether the performance of the PD-controller is improved or not. The reference input is given as Spong in [18] and [19]:

$$u_0 = k_p \alpha \cdot tan^{-1}(\Omega_b)$$

The parameter α is chosen less than β when the second link of an Acrobot is constrained to lie in an interval of $\theta_p \in [-\delta, \delta]$, where the value of δ is equivalent to the angular limitation of θ_p in radians. The idea is to utilize the "*atan*" function to give a reference depending on the bicycle link. By utilizing the angular velocity of the bicycle, the resulting PD-controller is given as:

$$u = -k_p \left(\theta_p - \alpha \cdot tan^{-1}(\Omega_b)\right) - k_d \Omega_p \tag{4.49}$$

Note that the smaller the amplitude, α, is, the less affect the reference input, u_0, will affect the system. If the amplitude is too small, the input reference will not contribute to the output from the PD-controller, giving a swing-up controller similar to the PD-controller with control parameters from Eq. 4.45. Through simulation, the amplitude was tuned to $\alpha = \frac{105}{180}\pi$, which is within the angular limitation of $\delta = \frac{110}{180}\pi$, giving:

$$u_0 = k_p \frac{105}{180}\pi \cdot tan^{-1}(\Omega_b) \tag{4.50}$$

As shown in Figure 4.35, the PD-controller with input reference, u_0, of Eq. 4.50 is verified to successfully control the system states towards zero, simultaneously. Thereby the angles are moved close to the upright position as the angular velocities are controlled towards zero. As the LQR balance control is not applied, the PD-controller is unable to stabilize the system around the upright unstable equilibrium, and bicycle angle, θ_b, falls back to the positive angular limit of 30°. The improved controller is then able to accelerate the inverted pendulum from the positive limit in negative direction to thereby accelerate in positive direction and move the bicycle angle towards the upright position, as a result of counteracting torque. This system response is similar to the input reference presented in Eq. 4.48 with magnitude of 124.022, but smaller control torque. When the bicycle angle, θ_b, is moved to the upright position, the inverted pendulum angle, θ_p, is moved slowly towards the desired angle of zero. Thereby both system angles are within the LQR region of attraction, while the angular velocities, Ω_b and Ω_p, are controlled towards zero. As the LQR balance controller is not applied, the system is unable to stabilize and the bicycle angle moves away from the upright equilibrium once more.

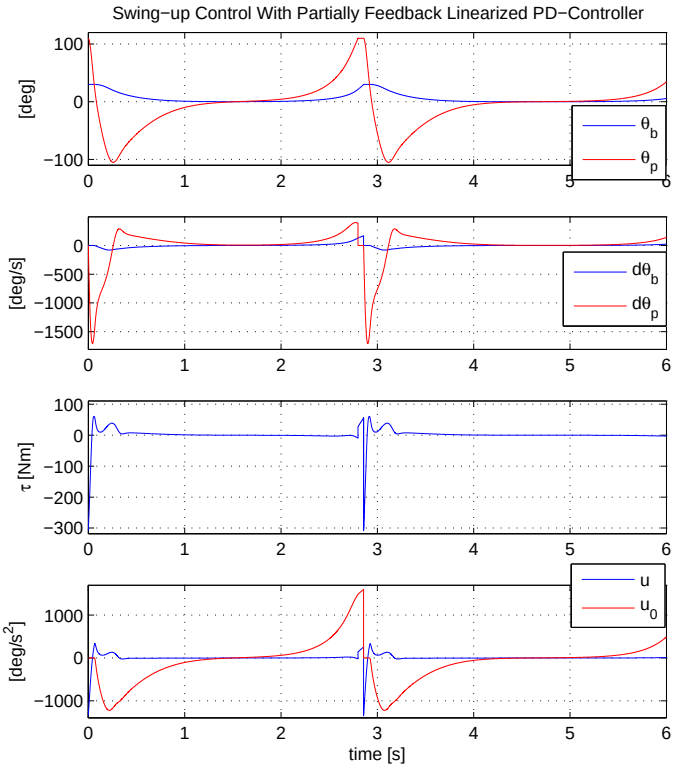


Figure 4.35: Time Response of Partially Feedback Linearized PD-Controller, $u_0 = atan$, $\alpha = \frac{105}{180}\pi$, $(\theta_b^{init}, \theta_p^{init}) = (30°, 110°)$

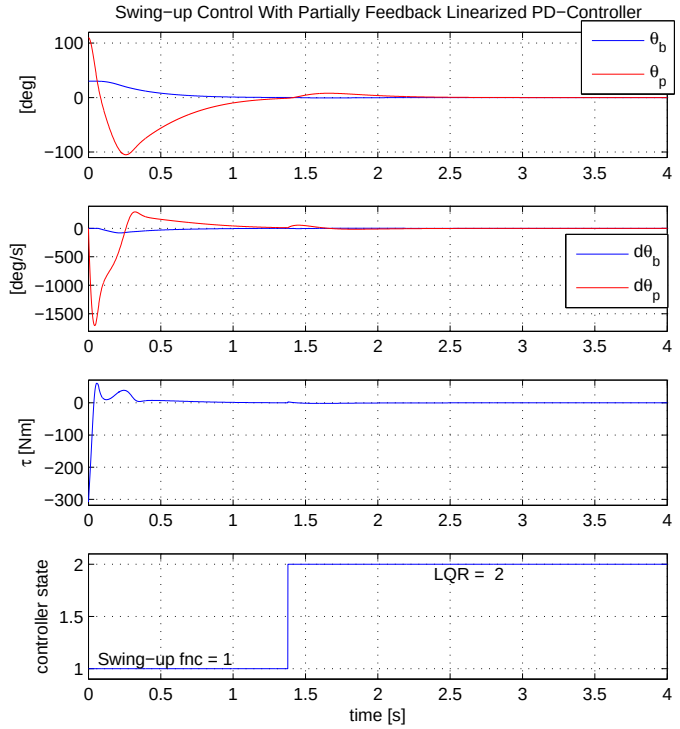


Figure 4.36: Time Response of Partially Feedback Linearized PD-Controller and LQR Balance Control, $u_0 = atan$, $\alpha = \frac{105}{180}\pi$, $(\theta_b^{init}, \theta_p^{init}) = (30°, 110°)$

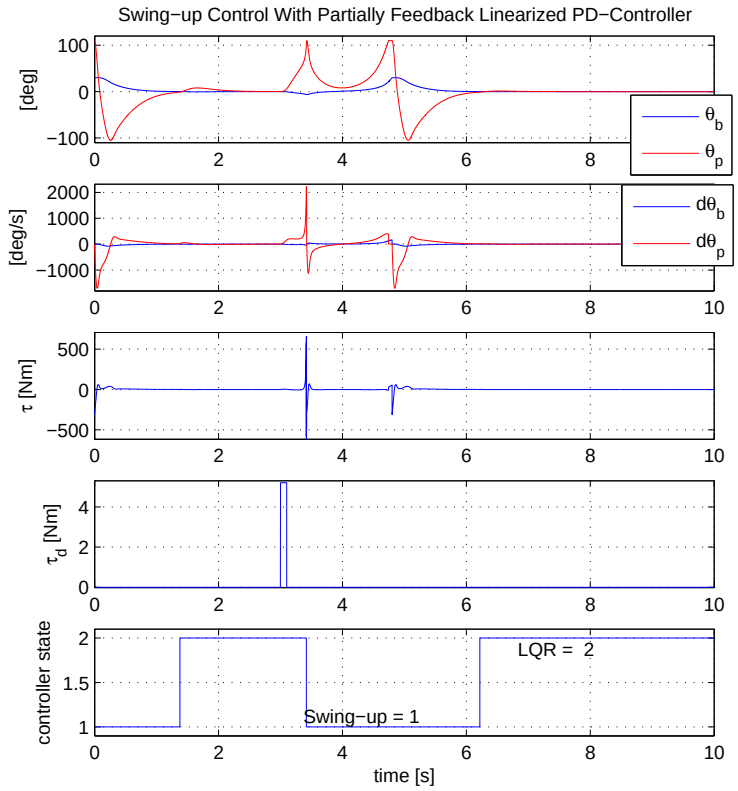


Figure 4.37: Time Response of Partially Feedback Linearized PD-Controller and LQR Balance Control w/ $\tau_d = 5.2$[Nm], $u_0 = atan$, $\alpha = \frac{105}{180}\pi$, $(\theta_b^{init}, \theta_p^{init}) = (30°, 110°)$

As the goal is to improve the performance of the stabilization controller by utilization of the reference input, u_0, the LQR balance controller is activated. Figure 4.36 shows how the PD-controller is able to move the system angles within the LQR region of attraction, while simultaneously controlling the angular velocities towards zero. Thereby the LQR balance controller is applied and the system stabilizes around the upright equilibrium.

In Figure 4.30(a) the LQR controller is proven to be able to stabilize the system when the perturbation torque, τ_d, is less than 5.1[Nm]. Thus a disturbance pulse of 5.2[Nm] is applied for 1[ms] at $t = 3$[s] to move the system outside of the LQR workspace and test the functionality of the reference input in Eq. 4.50. Figure 4.37 shows how the system is able to utilize the reference input, u_0, when the system moves outside the LQR balance controller workspace, with limits from Eq. 4.47. The controller is able to accelerate the inverted pendulum angle, θ_p, from the angular limitation of 110° in negative direction to thereby accelerate in positive direction, and move bicycle angle towards its upright position due to the counteracting torque. As the angle of θ_b is moved slowly towards the upright position, the PD-controller slowly controls the angle of θ_p towards the desired angle of zero. Thereby the angular velocities, Ω_b and Ω_p, are moved towards zero and the LQR balance controller is activated when the system angles reaches the LQR region of attraction. However, as the figure shows that the con-

trol torque becomes excessively large, and exceeds the maximum torque possible to apply from the system actuator, the reference input, u_0, in Eq. 4.50 is not applicable for the real life bicycle system.

4.3.3 Concluding Remarks

As the swing-up design only utilizes the states of θ_p and Ω_p in the PD-controller of Eq. 4.35, the controller is designed to stabilize the angle of θ_p around a desired angular value of zero, independently of the bicycle angle, θ_b. Thus the PD-controller is not designed to control θ_p dependent of the bicycle angle. Rather the gains are tuned to utilize the angular oscillations of θ_p and move the bicycle towards its upright position as a result of the counteracting torque. First the PD-controller, with input reference $u_0 = 0$, is utilized as a swing-up controller, where the parameters are tuned based on an initial positions at the angular limitations of $(\theta_b^{init}, \theta_p^{init}) = (30°, 110°)$ as in Eq. 4.45. As illustrated in Figure 4.28, the system is able to utilize the PD-controller and move the system states within the LQR region of attraction and activate the LQR balance controller for system stabilization. Further on the system is able to stabilize when perturbation occur. By enlarging the LQR workspace limits, β_1 and β_2, as presented in Eq. 4.47, the system is able to utilize the LQR balance controller to stabilize the system angles when perturbation occur. As shown in Figure 4.30(a) the system stabilization is performed without exceeding the maximum torque of the DC motor actuator, when the perturbation torque, τ_d, is less or equal to 5.1[Nm]. If the perturbation torque becomes larger, the system is unable to utilize the LQR controller for system stabilization, and the angles are moved outside the LQR workspace. As the system angles are too far away from the angular limits, the PD-controller with fixed controller parameters of Eq. 4.45 is unable to move the system angles towards the LQR region of attraction, while simultaneously controlling the angular velocities Ω_b and Ω_p towards zero. Thereby the system controller is unable to stabilize the state angles at the upright position, as shown in Figure 4.31. By utilizing a more strict workspace limit of θ_p, as in Eq. 4.46, the controller is able to utilize the PD-controller and move the system states towards zero to activate the LQR balance controller when perturbation occur, as shown in Figure 4.29. The disadvantage of this controller functionality is that the control torque exceeds the maximum torque possible to apply from the system actuator. Thereby the larger workspace limit of θ_p, as given in Eq. 4.47, has to be utilized on the real life bicycle system. This implies that the system must be allowed to fall to the angular limits and obtain angular velocities $\Omega_b = \Omega_p = 0$ to reinitialize the PD-controller with $(\theta_b^{init}, \theta_p^{init}) = (30°, 110°)$, when the angles moves outside of the workspace of the LQR balance controller. Therefrom the PD-controller is activated, with controller parameters from Eq. 4.45, to move the angles back into the LQR region of attraction, while simultaneously controlling the angular velocities towards zero, to activate the LQR balance controller for system stabilization.

As the PD-controller oscillates the inverted pendulum angle, θ_p, independently of the bicycle angle, θ_b, to move the system angles within the LQR region of attraction, the input reference, u_0, was investigated to improve the controller performance. By use of the saturation function of Eq. 4.48, the input reference is able to utilize the large inverted pendulum acceleration and move the bicycle towards its upright position. As the control torque exceeds the maximum possible actuation torque, the controller design and tuning is not applicable for the bicycle system with angular limitation.

Further on the "*atan*" function, of 4.50, is utilized, including the angular velocity of the bicycle. With the idea of utilizing the reference input function when perturbation occur, the robustness is tested. The system simulation in Figure 4.37 illustrates how the swing-up controller is able to move the system states back within the LQR region of attraction when perturbation occur. With the slow outer control-loop, the angle of θ_p is following the reference of u_0, as shown in Figure 4.35. This property should be taken into account when a new input reference controller, u_0, is designed. Similar to the saturation function, the "*atan*" reference input function results in large acceleration of the inverted pendulum to move the bicycle angle towards the upright position. Thereby the resulting control torque exceeds the maximum actuation torque. Thus, to improve the stabilization controller by use of the reference inputs of Eq. 4.48 and 4.50, the system actuator has to be upgraded to give the desired control torque required for these controller designs.

4.3.4 Controller Implementation

Partially Feedback Linearized LQR Controller:

"*controller_FeedbackLinearizedLQR*" is the implementation of the LQR controller designed by linearization around the upright equilibrium of the partially feedback linearized system, presented in Eq. 4.40 and Section 4.3.1.

```
function tau  = FeedbackLinearization_LQR(x)
l = 0.72;     % [m] approximately
rb = 0.4;     % [m] approximately
rp = 0.2;     % [m] approximately
mb = 31.118; % [kg] approximately
mp = 5;       % [kg] approximately
g = 9.81;     % [m/s^2] gravity acceleration
Ib = 0;       % [Nm^2] Moment of Inertia of bicycle
Ip = 0;       % [Nm^2] Moment of Inertia of pendulum

% System Energy
m11 = Ib+Ip+mb*rb^2+mp*l^2+mp*rp^2+2*mp*rp*l*cos(x(2));
m12 = Ip+mp*rp^2+mp*rp*l*cos(x(2));
m22 = Ip+mp*rp^2;

T = 1/2*m11*x(3)^2+m12*x(3)*x(4)+1/2*m22*x(4)^2;
U = (mb*rb+mp*l)*g*cos(x(1))+mp*rp*g*cos(x(1)+x(2));
E = T+U;
E0 = (mb*rb+mp*l)*g+mp*rp*g;
E_c = E-E0;

J_tot = mb*rb^2+mp*l*sin(x(2))^2;

% Torque parts of the diff.eq. of Omegap, J_tot*dOmegap/dt = T_pc+T_pg+Tptau*tau:
T_pc   = rp*mp*l*sin(x(2))*((x(3)+x(4))^2+((mb*rb^2)/(mp*rp^2)+(l/rp)^2)*...
         (x(3)^2)+(l/rp)*cos(x(2))*(x(3)^2+(x(3)+x(4))^2));
T_pg   = -g*mp*l*((1+(mb*rb)/(mp*l))*(1+(l/rp)*cos(x(2)))*sin(x(1))-...
         ((mb*rb^2)/(mp*rp*l)+l/rp+cos(x(2)))*sin(x(1)+x(2)));
T_ptau = 1+(mb*rb^2)/(mp*rp^2)+(l/rp)^2+2*(l/rp)*cos(x(2));

% Controller u:
u = -1000*[-9.1805   -0.5702   -2.1548   -0.1739]*[x(1) x(2) x(3) x(4)]';
tau = (J_tot*u-T_pc-T_pg)/T_ptau;
```

Partially Feedback Linearized Stabilization Controller w/ input reference u_0:

"*FeedbackLinearization_StateSpace*" is the implementation of the feedback linearized stabilization controller, with LQR balance controller and swing-up PD-controller. Note that the controller implementation shows the different choices of reference input, u_0, which can be chosen in simulation. The function "*hysteresis*" is equivalent to the function presented in the controller implementation in Section 4.1.7, and is utilized to set the limits of the LQR balance controller in real time.

```
function [tau, LQR_control] = FeedbackLinearization_StateSpace(x,t,control_state)

l = 0.72;     % [m] approximately
rb = 0.4;     % [m] approximately
rp = 0.2;     % [m] approximately
mb = 31.118; % [kg] approximately
mp = 5;       % [kg] approximately
g = 9.81;     % [m/s^2] gravity acceleration
Ib = 0;       % [Nm^2] Moment of Inertia of bicycle
Ip = 0;       % [Nm^2] Moment of Inertia of pendulum

% System Energy
m11 = Ib+Ip+mb*rb^2+mp*l^2+mp*rp^2+2*mp*rp*l*cos(x(2));
m12 = Ip+mp*rp^2+mp*rp*l*cos(x(2));
m22 = Ip+mp*rp^2;

T = 1/2*m11*x(3)^2+m12*x(3)*x(4)+1/2*m22*x(4)^2;
U = (mb*rb+mp*l)*g*cos(x(1))+mp*rp*g*cos(x(1)+x(2));
E = T+U;

E0 = (mb*rb+mp*l)*g+mp*rp*g;
E_c = E-E0;

% Inertia-term J_tot:
J_tot = mb*rb^2+mp*l*sin(x(2))^2;

% Torque parts of the diff.eq. of Omegap, J_tot*dOmegap/dt = T_pc+T_pg+Tptau*tau:
T_pc   = -rp*mp*l*sin(x(2))*((x(3)+x(4))^2+((mb*rb^2)/(mp*rp^2)+(l/rp)^2)*...
         (x(3)^2)+(l/rp)*cos(x(2))*(x(3)^2+(x(3)+x(4))^2));
T_pg   = -g*mp*l*((1+(mb*rb)/(mp*l))*(1+(l/rp)*cos(x(2)))*sin(x(1))-...
         ((mb*rb^2)/(mp*rp*l)+l/rp+cos(x(2)))*sin(x(1)+x(2)));
T_ptau = 1+(mb*rb^2)/(mp*rp^2)+(l/rp)^2+2*(l/rp)*cos(x(2));
```

```
33
34 % PD-Controller parameters:
35 kp = 700; % w0^2
36 kd = 0.4598*2*sqrt(kp); % 2*zeta*sqrt(kp)
37
38 % LQR Region of attraction:
39 LQR_lim_x1 = 0.6/180*pi;
40 LQR_lim_x2 = 1.2/180*pi;
41
42 %Set LQR balance controller workspace limits:
43 beta_1 = pi/6;
44 beta_2 = 100/180*pi;
45 limit = [beta_1 beta_2 LQR_lim_x1 LQR_lim_x2];
46 if t == 0
47     [x_lim, LQR_control] = hysteresis(x,limit,false);
48 else
49     [x_lim, LQR_control] = hysteresis(x,limit,control_state);
50 end
51
52 % CONTROLLER:
53 x1_lim = x_lim(1);
54 x2_lim = x_lim(2);
55 % LQR balance controller:
56 if mod(abs(x(1)),2*pi)<=x1_lim && mod(abs(x(2)),2*pi)<=x2_lim && LQR_control == 1
57     tau = -[-975.6281  -57.2364 -227.3441  -19.7419]*[x(1) x(2) x(3) x(4)]';
58     state = 2;
59 % Swing-up PD-controller:
60 else
61     u0 = 0;
62
63     % Uncomment if u0 with saturation is to be utilized:
64 %     u0 = 97*sat2(E_c*x(3));
65
66     % Uncomment If u0 with atan is to be utilized:
67 %     alpha = 105/180*pi;
68 %     u0 = kp*alpha*atan(x(3));
69
70     u = -kp*x(2)-kd*x(4)+u0;
71     tau = (J_tot*u-T_pc-T_pg)/T_ptau;
72     state = 1;
73 end
```

Chapter 5

Discussion and Further Work

5.1 Discussion

In this thesis, the bicycle system is presented as an Acrobot with angular limitations. The system equations of motion are derived by use of Lagrange, without joint friction and energy dissipation as the system joints does not contribute with friction in their respective rotational axis. Utilization of Lagranges equation of motion is based on the property of a structured set of operations for derivation of system equations. Further on the moments of inertia, I_b and I_p, can be included into the inertia matrix $(M)(\boldsymbol{\theta})$ for the complete design of the stabilization controller, if required.

With system actuation from the DC motor mounted on the bicycle frame, Chapter 2.5 shows how the system response is highly improved when the motor is equipped with a current controller. Not only is the system response improved with the current controller, but the motor can be protected to prevent demagnetization of the permanent magnets and a smaller converter can be used. This motor investigation shows preliminaries for the motor system, needed to give a reliable actuator for the physical bicycle system.

When the system is modeled as an Acrobot, limitations has to be included into the system model to get the complete model of the autonomous bicycle with a mounted inverted pendulum. These limits are set by the physical angular limitations of the bicycle system to obtain an equivalent mathematical system model. As the goal is to model the real bicycle system as an Acrobot within the angular limitation region, this is a very important part of the system model as the inverted pendulum systems presented in the various literature has system angles in the range of $[-\pi, \pi)$ and utilizes full swing-up control. To move the bicycle angle, θ_b, away from the limit of 30°, a negative acceleration of θ_p has to be applied for thereafter apply a torque accelerating θ_p in the positive direction, generating a coun-

teracting torque onto the bicycle. This counteracting torque moves the bicycle from the angular limit towards the upright position. The requirement of using the mounted inverted pendulum to increase the system energy and move the bicycle results in a challenging control design.

When the control theory was applied, the system was linearized and the LQR controller was designed. This controller was designed for balancing control around the upright unstable system equilibrium. Due to the sing-input multiple-output system, the LQR controller was designed due to the fact that it is an optimal pole placing controller, giving a more suitable control design than the single-input single-output controllers, as P-, PI- and PID-controllers. The system could also be linearized around the measured system angles in real time, for utilization of the LQR as a swing-up controller. This was not performed in this thesis as the time was spent on investigation of nonlinear controllers for system swing-up.

Due to the angular limitations the nonlinear energy-based Acrobot swing-up controllers, presented in Section 4.1 and 4.2, does not satisfy the system requirements of bicycle stabilization within the limits. The idea was to utilize the designs by Lai et al. and Kobayashi et al. to obtain a functional stabilization controller. As the controllers are designed to rotate outside of the bicycle system limited angular region, by swinging back and forth, the control torque is applied to increase the energy by use of the bicycle angle. By moving the angle of θ_b closer to the upright equilibrium for each swing while simultaneously controlling the angle of θ_p towards zero, the stabilization designs applies the LQR balance controller when the angles are moved within the LQR region of attraction for system stabilization. Thereby the controllers are designed for an unlimited Acrobot, and does not satisfy the system requirement of utilizing the inverted pendulum to increase system energy and move the states towards zero within the bicycle angular limits.

As the stabilization control is unsuccessful for the controllers presented by Lai et al. and Kobayashi et al., the partially feedback linearization is introduced. The feedback linearized system gives a linear second order system of the inverted pendulum acceleration, $\ddot{\theta}_p$, which can simplify the controller design for the bicycle system. By implementing a swing-up PD-controller, the goal is to utilize a controlled oscillatory system response of the inverted pendulum angle, θ_p, to give a counteracting torque onto the bicycle link and move the system states. As the PD-controller oscillates the angle of θ_p, the goal is to utilize the controller output, $u = \ddot{\theta}_p$, to control the bicycle angular velocity, Ω_b, such that $\theta_b = 0$. The control input, u_0, was investigated to improve the controller performance and include the states of θ_b and Ω_b into a nonlinear input reference function. As these input references is not applica-

ble to the bicycle system, due to a control torque exceeding the maximum actuation torque, the input reference of $u_0 = 0$ must be utilized. When the input reference of $u_0 = 0$ is utilized, the system must be allowed to fall to the limits and obtain zero angular velocities and reinitialize the PD-controller, when the system moves outside of the LQR balance controller workspace. Otherwise the controller can be further investigated by improved functions of u_0, or by finding new swing-up controllers. Another solution is to utilize the current swing-up controller and further investigate an improved balance controller with larger region of attraction.

5.2 Further Work

Improve Balance Controller:

Design an improved balance controller which has a larger region of attraction. Thereby the existing PD-controller can be utilized to move the system angles within the region of attraction of the balance controller, when the states are moved outside of the balance workspace.

Improved Swing-Up Controller:

By designing an improved swing-up controller, the complete stabilization controller can utilize the existing LQR balance controller. The swing-up controller can be improved by further investigation of the PD-controller input reference, u_0, or by looking at a new controller strategy for system swing-up.

Sensitivity Analysis on System and Controller Parameters:

It is a large uncertainty to the controller parameters as they are tuned based on the approximate system parameter values. The PD-controller parameters, k_p and k_d, were finely tuned through simulation, thereby a sensitivity analysis should be performed to analyze and verify the controller functionality before the complete controller is implemented on the real life bicycle system.

Limitation modification:

As the system limitations are defined in the implementation made by Ånnestad in [14], as presented in Section 2.1, the system has been proven to utilize state angles outside the limits assumed by Ånnestad. Thereby the simulations of the nonlinear control verifies that these controller limits on the system implemented by Ånnestad in [14] has to be changed, before complete controller implementation.

Controller implementation

When the complete stabilization controller is derived, the controller should be implemented for use in the real bicycle system. The implementation should be included in the bicycle system implementation from [14] done in Simulink.

Filter theory and measurements:

The required accuracy of the position and speed measurements has to be further investigated. Different types of filter theory can be applied. The system consist of sensor measurements, which contains noise and can be filtered by use e.g. a first order low-pass filter, as presented in Section 3.9.4. Further research can be conducted, and observers can be included. Due to the nonlinear system the Extended Kalman filter might be applicable.

Testing complete controller implementation on the physical bicycle system

When the complete controller is implemented in the bicycle system model designed by Ånnestad, testing should be performed. These test results will verify if the complete stabilizing controller satisfies the requirements for stabilization of the physical system. The results can be compared with results from system simulation, and analyzed.

Chapter 6

Conclusion

The goal of this thesis was to investigate and develop a stabilization controller for the bicycle system, by use of the previously mathematical system model in the authors own work [12]. By presenting the system as an Acrobot with angular limitations, the two controller strategies of Acrobot control, presented in [12] has been investigated. Through simulation, the controller implementations by Lai et al. and Kobayashi et al. has been verified as inapplicable stabilization controllers for the bicycle system. The design by Lai et al., presented in Section 4.1, utilizes the energy-based swing-up controller with PD-structure to swing up the system states to the upright unstable equilibrium. As illustrated in the simulation in Figure 4.7, the controller utilizes the angular velocity of the bicycle, Ω_b, to increase the system energy, and thereby moves the angles outside the angular limits of the bicycle. By applying a small control torque, the controller moves the system states closer to the upright position by each swing. The PD-structure is utilized to give the desired control torque and control the inverted pendulum angle towards zero, for utilization of the LQR balance controller. Neither the implementation based on Kobayashi et al. was able to stabilize the system within the angular limitations. This design did not utilize the system states in the same way as Lai et al. The controller was based on the property of a negative semi-definite time derivative of the energy function, $E_z(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$, and was unable to move the system states within the balance subspace in simulation. However, both controllers illustrated how the increased system energy has to be applied from the inverted pendulum angle, θ_p, instead of the bicycle angle. As the angular velocities are zero when the angles are at the limits of the bicycle system, the inverted pendulum must be utilized to increase the system energy and move the bicycle angle towards its upright position.

When the controller designs presented in the authors previous work [12], was verified as inapplicable stabilization controller for the bicycle system, further research was conducted. The partially feedback linearized system was derived, resulting in a linear second order system coupled with a nonlinear

system, as the presentation in Eq. 4.32. A simplified swing-up controller was designed by use of a PD-controller, presented in Eq. 4.35. As the swing-up design only utilizes the states of θ_p and Ω_p, the controller was designed to stabilize the angle of θ_p around a desired angular value of zero, independently of the bicycle angle, θ_b. Thus the gains are tuned to utilize the angular oscillations of θ_p and move the bicycle towards its upright position as a result of the counteracting torque. First the PD-controller, with input reference $u_0 = 0$, was utilized as a swing-up controller, where the parameters are tuned based on an initial positions at the angular limitations of $(\theta_b^{init}, \theta_p^{init}) = (30°, 110°)$, as in Eq. 4.45. As illustrated in Figure 4.28, the system was able to utilize the PD-controller and move the system states within the LQR region of attraction and activate the LQR balance controller for system stabilization. Further on the controller robustness was tested. As shown in Figure 4.30(a) the system stabilization was performed without exceeding the maximum torque of the DC motor actuator, when the perturbation torque, τ_d, was less or equal to 5.1[Nm]. If the perturbation torque became larger, the system was unable to utilize the LQR controller for system stabilization, and the angles are moved outside the LQR workspace. As the system angles were too far away from the angular limits, the PD-controller with fixed controller parameters of Eq. 4.45 was unable to move the system angles towards the LQR region of attraction, while simultaneously controlling the angular velocities Ω_b and Ω_p towards zero. Thereby the system controller was unable to stabilize the state angles at the upright position, as shown in Figure 4.31. The system must be allowed to fall to the angular limits and obtain angular velocities $\Omega_b = \Omega_p = 0$, when the angles moves outside of the workspace of the LQR balance controller. From the limits, the PD-controller can be reinitialized, with controller parameters from Eq. 4.45, to move the angles back into the LQR region of attraction, while simultaneously controlling the angular velocities towards zero, to activate the LQR balance controller for system stabilization.

As the PD-controller oscillates the inverted pendulum angle, θ_p, independently of the bicycle angle, θ_b, to move the system angles within the LQR region of attraction, the input reference, u_0, was investigated to improve the controller performance, without success. The reference input functions presented in Eq. 4.48 and 4.50 was utilized to include bicycle angle, θ_b, and the angular velocity of the bicycle, Ω_b, for improved swing-up control. The reference inputs were unable to improve the system performance, as the input reference gave a controller torque which exceeded the maximum actuation torque when perturbation occurred. When the disturbance torque was applied onto the system, the resulting controller torque gave an excessively large acceleration of the inverted pendulum. Thus, to improve the stabilization controller by use of the reference inputs from Eq. 4.48 and 4.50, the system actuator has to be upgraded to give the desired control torque required for these controller designs.

Thereby the swing-up PD-controller with $u_0 = 0$ is the only stabilization controller which successfully stabilizes the bicycle system without exceeding the maximum actuation torque. To improve the complete stabilization controller for system implementation, the future work should focus on utilization of the LQR balance controller and improve the swing-up controller. Another method is to focus on utilization of the current PD swing-up controller and further investigate a balance controller with larger region of attraction.

Appendix A

Appendix A

In this appendix, the derivation and calculations of the system model presented in Chapter 3 will be shown. The model was derived in the authors own work in [12].

A.1 System Coordinates

In Section 3.2 the system coordinates was presented. The relative generalized coordinates used in the model derivation are illustrated in Figure 3.2, where (x_i, y_i, z_i) is the initial frame and θ_p is relative to θ_b. The rotation is around the z_i-axis, as shown in Figure A.1, with respect to the right-hand rule. The coordinate frames of the bicycle and inverted pendulum are denoted (x_b, y_b, z_b) and (x_p, y_p, z_p), respectively. The two coordinate systems are defined as:

$$\begin{aligned}
\vec{x}_b &= cos\,\theta_b \cdot \vec{x}_i + sin\,\theta_b \cdot \vec{y}_i \\
\vec{y}_b &= -sin\,\theta_b \cdot \vec{x}_i + cos\,\theta_b \cdot \vec{y}_i \\
\vec{z}_b &= \vec{z}_i
\end{aligned} \tag{A.1}$$

$$\begin{aligned}
\vec{x}_p &= cos(\theta_b + \theta_p) \cdot \vec{x}_i + sin(\theta_b + \theta_p) \cdot \vec{y}_i \\
\vec{y}_p &= -sin(\theta_b + \theta_p) \cdot \vec{x}_i + cos(\theta_b + \theta_p) \cdot \vec{y}_i \\
\vec{z}_p &= \vec{z}_i
\end{aligned}$$

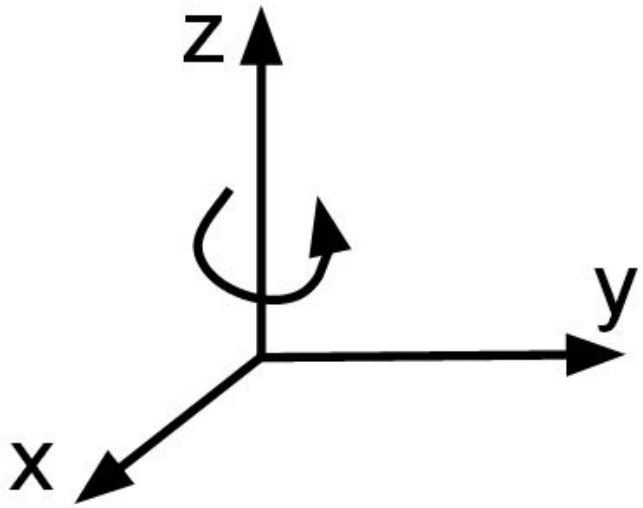


Figure A.1: Rotation About The z-axis

A.2 Vectorial Length and Velocity

When the system masses are presented as point masses, the coordinates system presented in Section 3.2 are presented as vectorial, giving the elements both size and direction. The system energy was derived by use of vectorial notation, and use of Eq. (6.400) in [7]:

$$\begin{aligned} \vec{v}_a &\equiv \frac{^i d}{dt}\vec{r}_a \\ &= \vec{v}_0 + \frac{^a d}{dt}\vec{r}_a + \vec{\omega}_{ia} \times \vec{r}_a \end{aligned} \tag{A.2}$$

where subscript a denotes a reference coordinate frame relative to the initial reference frame denoted i. Note that $\vec{v}_0$ is zero in the upcoming equations, derived in [12]. The system parameters are illustrated in Figure 3.3.

A.2.1 System Mass m_b

The position of m_b, denoted $\vec{r}_b$, is placed along the defined y-axis of the bicycle coordinate frame:

$$\vec{r}_b = r_b \cdot \vec{y}_b \tag{A.3}$$

By use of Eq. A.2, the resulting vectorial velocity was derived as:

$$\begin{aligned}\vec{v}_b &= \frac{^i d}{dt}\vec{r}_b \\ &= \frac{^b d}{dt}(r_b \cdot \vec{y}_b) + \vec{\omega}_{ib} \times (r_b \cdot \vec{y}_b)\end{aligned}$$

$\frac{^b d}{dt}$ represents time derivative with respect to the bicycle coordinate frame, b. As neither r_b or $\vec{y}_b$ changes in time with respect to the bicycle frame, the equation becomes:

$$\vec{v}_b = \vec{\omega}_{ib} \times (r_b \cdot \vec{y}_b)$$

Where $\vec{\omega}_{ib}$ is the angular velocity of the bicycle mass center, m_b, rotating around the defined z_b-axis in Eq. A.1. This gives: $\vec{\omega}_{ib} = -\dot{\theta}_b \cdot \vec{z}_b$, resulting in:

$$\begin{aligned}\vec{v}_b &= (-\dot{\theta}_b \cdot \vec{z}_b) \times (r_b \cdot \vec{y}_b) \\ &= -(\dot{\theta}_b r_b)\vec{z}_b \times \vec{y}_b \\ &= r_b \dot{\theta}_b \cdot \vec{x}_b\end{aligned} \tag{A.4}$$

A.2.2 System Actuator Joint

As the system actuator joint is not a mass point, the joint does not contribute with system energy. Nonetheless, in [12], the vectorial length and velocity was derived for use in the derivation of the length and velocity of the system mass m_p. The joint position is placed along the defined y-axis of the bicycle coordinate frame, with length l:

$$\vec{r}_j = l \cdot \vec{y}_b \tag{A.5}$$

By use of Eq. A.2, the time derivative of the joint position gives the vectorial velocity:

$$\begin{aligned}\vec{v}_j &= \frac{^b d}{dt}(l \cdot \vec{y}_b) + \vec{\omega}_{ib} \times (l \cdot \vec{y}_b) \\ &= \vec{\omega}_{ib} \times (l \cdot \vec{y}_b) \\ &= -(\dot{\theta} \cdot \vec{z}_b) \times (l \cdot \vec{y}_b) \\ &= l\dot{\theta}_b \cdot \vec{x}_b\end{aligned} \tag{A.6}$$

A.2.3 System Mass m_p

With the pendulum mounted on the system actuator joint, the position $\vec{r}_p$ was derived as:

$$\begin{aligned} \vec{r}_p &= \vec{r}_j + r_p \cdot \vec{y}_p \\ &= l \cdot \vec{y}_b + r_p \cdot \vec{y}_p \end{aligned} \tag{A.7}$$

As for the mass point m_b, Eq. A.2 was applied, giving:

$$\begin{aligned} \vec{v}_p &= \vec{v}_{joint} + \frac{^i d}{dt}\vec{r}_p \\ &= \vec{v}_{joint} + \frac{^p d}{dt}(r_p \cdot \vec{y}_b) + \vec{\omega}_{ip} \times (r_p \cdot \vec{y}_p) \end{aligned} \tag{A.8}$$

Where $\frac{^p d}{dt}$ represents time derivative with respect to the pendulum coordinate frame, p. As for the bicycle, neither r_p or $\vec{y}_b$ changes in time with respect to the pendulum frame. The velocity is then written as:

$$\vec{v}_p = \vec{v}_{joint} + \vec{\omega}_{ip} \times (r_p \cdot \vec{y}_p)$$

$\vec{\omega}_{ip}$ is the angular velocity of m_p rotating around the defined z_p-axis in Eq. A.1. This gives: $\vec{\omega}_{ip} = -(\dot{\theta}_b + \dot{\theta}_p) \cdot \vec{z}_p$, resulting in:

$$\begin{aligned} \vec{v}_p &= \vec{v}_{joint} - (\dot{\theta}_b + \dot{\theta}_p) \cdot \vec{z}_p \times (r_p \cdot \vec{y}_p) \\ &= l\dot{\theta}_b \cdot \vec{x}_b + r_p(\dot{\theta}_b + \dot{\theta}_p) \cdot \vec{x}_p \end{aligned} \tag{A.9}$$

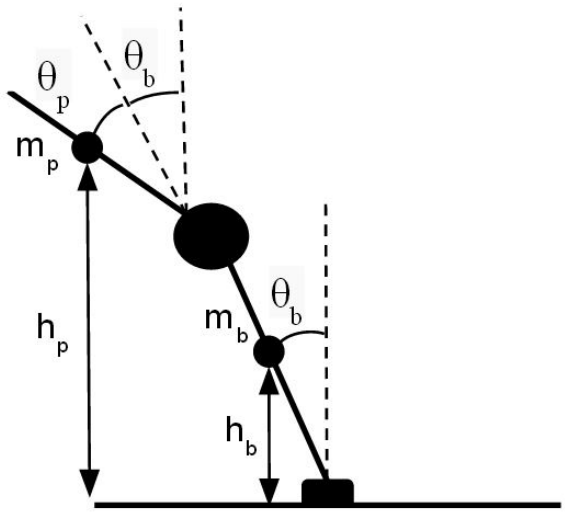


Figure A.2: Defined heights for system potential energy

A.3 System Energy

As the position and velocity of the system masses has been derived, the potential and kinetic energy of the point masses was derived. These energy expressions are utilized in the Lagranges equation of motion, Eq. 3.5, for derivation of the mathematical equation of motion of the bicycle system presented in Section 3.5.

A.3.1 Potential Energy

Figure A.2 illustrates the defined heights for the bicycle system, giving the potential energy of the system masses m_b and m_p, with heights h_b and h_p defined as:

$$h_b = r_b \cos\theta_b$$
$$h_p = l \cos\theta_b + r_p \cos(\theta_b + \theta_p)$$

The corresponding potential energy of the system masses becomes, as shown in Section 3.4:

$$\begin{aligned} U_b &= m_b g h_b \\ &= m_b r_b g \cos\theta_b \end{aligned}$$

$$\begin{aligned} U_p &= m_p g h_p \\ &= m_p g (l \cos\theta_b + r_p \cos(\theta_b + \theta_p) \end{aligned}$$

A.3.2 Kinetic Energy

By use of the vectorial velocity equations derived in [12] and shown in Appendix A.2, the kinetic energy of the point masses was derived. For the system mass m_b, the kinetic energy is given by:

$$\begin{aligned} T_b &= \frac{1}{2} m_b \vec{v}_b \cdot \vec{v}_b \\ &= \frac{1}{2} m_b [-r_b \dot{\theta}_b \cdot \vec{x}_b] \cdot [-r_b \dot{\theta}_b \cdot \vec{x}_b] \\ &= \frac{1}{2} m_b (r_b \dot{\theta}_b)^2 \end{aligned} \tag{A.12}$$

where the dot product of $\vec{x}_b$ is: $\vec{x}_b \cdot \vec{x}_b = |\vec{x}_b||\vec{x}_b| cos\,\theta_{x_b x_b} = 1$.

The kinetic energy of the system mass m_p is a bit more complex. By use of the vectorial velocity equation of $\vec{v}_p$, the derivation in [12] was:

$$T_p = \frac{1}{2} m_p \vec{v}_p \cdot \vec{v}_p \tag{A.13a}$$

$$= \frac{1}{2} m_p [l\dot{\theta}_b \cdot \vec{x}_b + (\dot{\theta}_b + \dot{\theta}_p) r_p \cdot \vec{x}_p] \cdot [l\dot{\theta}_b \cdot \vec{x}_b + (\dot{\theta}_b + \dot{\theta}_p) r_p \cdot \vec{x}_p] \tag{A.13b}$$

$$= \frac{1}{2} m_p [(l\dot{\theta}_b)^2 \cdot \vec{x}_b \cdot \vec{x}_b + l r_p \dot{\theta}_b (\dot{\theta}_b + \dot{\theta}_p) \cdot \vec{x}_b \cdot \vec{x}_p + l r_p \dot{\theta}_b (\dot{\theta}_b + \dot{\theta}_p) \cdot \vec{x}_p \cdot \vec{x}_b + r_p^2 (\dot{\theta}_b + \dot{\theta}_p)^2 \cdot \vec{x}_p \cdot \vec{x}_p] \tag{A.13c}$$

$$= \frac{1}{2} m_p [(l\dot{\theta}_b)^2 + 2 l r_p \dot{\theta}_b (\dot{\theta}_b + \dot{\theta}_p) \vec{x}_b \cdot \vec{x}_p + r_p^2 (\dot{\theta}_b + \dot{\theta}_p)^2] \tag{A.13d}$$

$$= \frac{1}{2} m_p [(l\dot{\theta}_b)^2 + 2 l r_p \dot{\theta}_p (\dot{\theta}_b + \dot{\theta}_p) cos\,\theta_p + r_p^2 (\dot{\theta}_b + \dot{\theta}_p)^2] \tag{A.13e}$$

Where the rule of vectorial dot product has been utilized, giving: $\vec{x}_b \cdot \vec{x}_b = 1$, $\vec{x}_p \cdot \vec{x}_p = 1$ and $\vec{x}_b \cdot \vec{x}_p = \vec{x}_p \cdot \vec{x}_b$ in Eq. A.13c.

Further on the dot product of $\vec{x}_b$ and $\vec{x}_p$ in Eq. A.13d was calculated. Using Eq. A.1 and the fact that the perpendicular basis vectors $\vec{x}_a$ and $\vec{y}_a$ has the property: $\vec{x}_a \cdot \vec{y}_a = \vec{y}_a \cdot \vec{x}_a = 0$, the dot product is:

$$\begin{aligned} \vec{x}_b \cdot \vec{x}_p &= [cos\,\theta_b \cdot \vec{x}_i + sin\,\theta_b \cdot \vec{y}_i] \cdot [cos(\theta_b + \theta_p) \cdot \vec{x}_i + sin(\theta_b + \theta_p) \cdot \vec{y}_i] \\ &= cos\,\theta_b\, cos(\theta_b + \theta_p) + cos\,\theta_b\, sin(\theta_b + \theta_p) \cdot \vec{x}_i \cdot \vec{y}_i + sin\,\theta_b\, cos(\theta_b + \theta_p) \cdot \vec{y}_i \cdot \vec{x}_i + sin\,\theta_b\, sin(\theta_b + \theta_p) \\ &= cos\,\theta_b\, cos(\theta_b + \theta_p) + sin\,\theta_b\, sin(\theta_b + \theta_p) \end{aligned}$$

Using the trigonometric addition formula: $cos(\alpha - \beta) = cos\,\alpha\, cos\,\beta + sin\,\alpha\, sin\,\beta$, the dot product can be written as:

$$\vec{x}_b \cdot \vec{x}_p = cos(\theta_b - (\theta_b + \theta_p)) = cos(-\theta_p) = cos\,\theta_p$$

Giving the final expression for the kinetic energy of m_p in Eq. A.13e.

A.4 The System Lagrange Equation of Motion

The equation of motion of the bicycle system, presented as an Acrobot, in Section 3.5, was derived by use og the Lagrangian function in Eq. 3.6 and the Lagranges equation of motion, Eq. 3.5. The Lagrangian function is defined as:

$$\begin{aligned} L &= T - U \\ &= \frac{1}{2}(m_b r_b^2 + m_p l^2)\dot{\theta}_b^2 + m_p r_p l \dot{\theta}_b(\dot{\theta}_b + \dot{\theta}_p) cos\,\theta_p + \frac{1}{2} m_p r_p^2 (\dot{\theta}_b + \dot{\theta}_p)^2 \\ &- (m_b r_b + m_p l) g\, cos\,\theta_b - m_p r_p g\, cos(\theta_b + \theta_p) \end{aligned} \tag{A.14}$$

The two degrees of freedom $\boldsymbol{q} = [\theta_b, \theta_p]^T$ in Eq. 3.5 gives the set of Lagrangian Equations of motion:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_b}\right) - \frac{\partial L}{\partial \theta_b} = 0 \tag{A.15a}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_p}\right) - \frac{\partial L}{\partial \theta_p} = \tau \tag{A.15b}$$

Where:

$$\frac{\partial L}{\partial \theta_b} = (m_b r_b + m_p l) g\, sin\,\theta_b + m_p r_p g\, sin(\theta_b + \theta_p) \tag{A.16a}$$

$$\frac{\partial L}{\partial \dot{\theta}_b} = (m_b r_b^2 + m_p l^2)\dot{\theta}_b + m_p r_p l\, cos\,\theta_p (2\dot{\theta}_b + \dot{\theta}_p) + m_p r_p^2 (\dot{\theta}_b + \dot{\theta}_p) \tag{A.16b}$$

$$\frac{\partial L}{\partial \theta_p} = -m_p r_p l \dot{\theta}_b (\dot{\theta}_b + \dot{\theta}_p)\, sin\,\theta_p + m_p r_p g\, sin(\theta_b + \theta_p) \tag{A.16c}$$

$$\frac{\partial L}{\partial \dot{\theta}_p} = m_p r_p l \dot{\theta}_b\, cos\,\theta_p + m_p r_p^2 (\dot{\theta}_b + \dot{\theta}_p) \tag{A.16d}$$

By use of the equations Eq. A.15 and Eq. A.16 the equation of motion for θ_b has been derived as:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_b}\right) - \frac{\partial L}{\partial \theta_b} = \frac{d}{dt}\left((m_b r_b^2 + m_p l^2)\dot{\theta}_b + m_p r_p l\, cos\,\theta_p(2\dot{\theta}_b + \dot{\theta}_p) + m_p r_p^2(\dot{\theta}_b + \dot{\theta}_p)\right) \tag{A.17a}$$

$$- (m_b r_b + m_p l) g\, sin\,\theta_b + m_p r_p g\, sin(\theta_b + \theta_p)$$

$$= (m_b r_b^2 + m_p l^2)\ddot{\theta}_b + 2m_p r_p l\ddot{\theta}_b\, cos\,\theta_p - 2m_p r_p l\dot{\theta}_b\dot{\theta}_p\, sin\,\theta_p$$

$$+ m_p r_p l\ddot{\theta}_p\, cos\,\theta_p - m_p r_p l\dot{\theta}_p^2\, sin\,\theta_p + m_p r_p^2(\ddot{\theta}_b + \ddot{\theta}_p) \tag{A.17b}$$

$$- (m_b r_b + m_p l) g\, sin\,\theta_b - m_p r_p g\, sin(\theta_b + \theta_p)$$

$$= (m_b r_b^2 + m_p l^2 + m_p r_p^2 + 2m_p r_p l\, cos\,\theta_p)\ddot{\theta}_b + (m_p r_p^2 + m_p r_p l\, cos\,\theta_p)\ddot{\theta}_p$$

$$- 2m_p r_p l\dot{\theta}_b\dot{\theta}_p\, sin\,\theta_p - m_p r_p l\dot{\theta}_p^2\, sin\,\theta_p - (m_b r_b + m_p l) g\, sin\,\theta_b \tag{A.17c}$$

$$- m_p r_p g\, sin(\theta_b + \theta_p)$$

For the mounted inverted pendulum, θ_p, the equation of motion has been derived as:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_p}\right) - \frac{\partial L}{\partial \theta_p} = \frac{d}{dt}\left(m_p r_p l\dot{\theta}_b\, cos\,\theta_p + m_p r_p^2(\dot{\theta}_b + \dot{\theta}_p)\right) + m_p r_p l\dot{\theta}_b(\dot{\theta}_b + \dot{\theta}_p)\, sin\,\theta_p \tag{A.18a}$$

$$- m_p r_p g\, sin(\theta_b + \theta_p)$$

$$= m_p r_p l\ddot{\theta}_b\, cos\,\theta_p - m_p r_p l\dot{\theta}_b\dot{\theta}_p\, sin\,\theta_p + m_p r_p^2(\ddot{\theta}_b + \ddot{\theta}_p) \tag{A.18b}$$

$$+ m_p r_p l\dot{\theta}_b(\dot{\theta}_b + \dot{\theta}_p)\, sin\,\theta_p - m_p r_p g\, sin(\theta_b + \theta_p)$$

$$= (m_p r_p^2 + m_p r_p l cos\,\theta_p)\ddot{\theta}_b + m_p r_p^2\ddot{\theta}_p + m_p r_p l\dot{\theta}_b^2\, sin\,\theta_p \tag{A.18c}$$

$$- m_p r_p g\, sin(\theta_b + \theta_p)$$

From Eq. A.15, A.17c and A.18c the final system equation of motion was derived, as presented in Eq 3.8 and 3.9.

Appendix B

Appendix B

In this appendix, some of the system implementations for controllers in Chapter 4 are presented. The implementations are developed in Matlab®and Simulink®.

B.1 Implementation of State Space Representation in Simulink

Figure B.1 shows the implementation of right hand side of the differential equation in Eq. 3.20a. The terms are categorized in centrifugal and Coriolis, gravity and actuator related torques for system analysis.

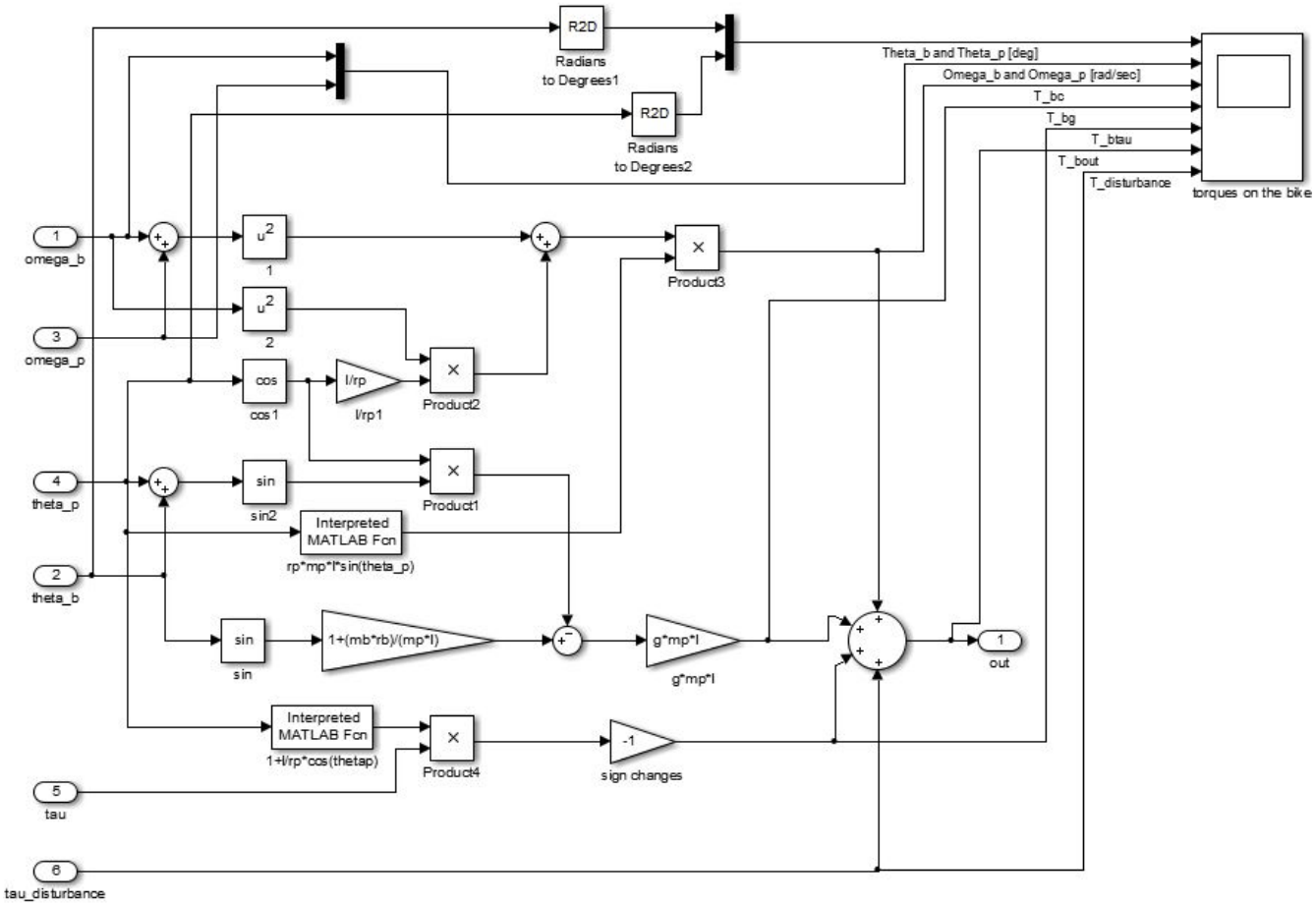


Figure B.1: Implemented State Space Representation of Bicycle System in Simulink®

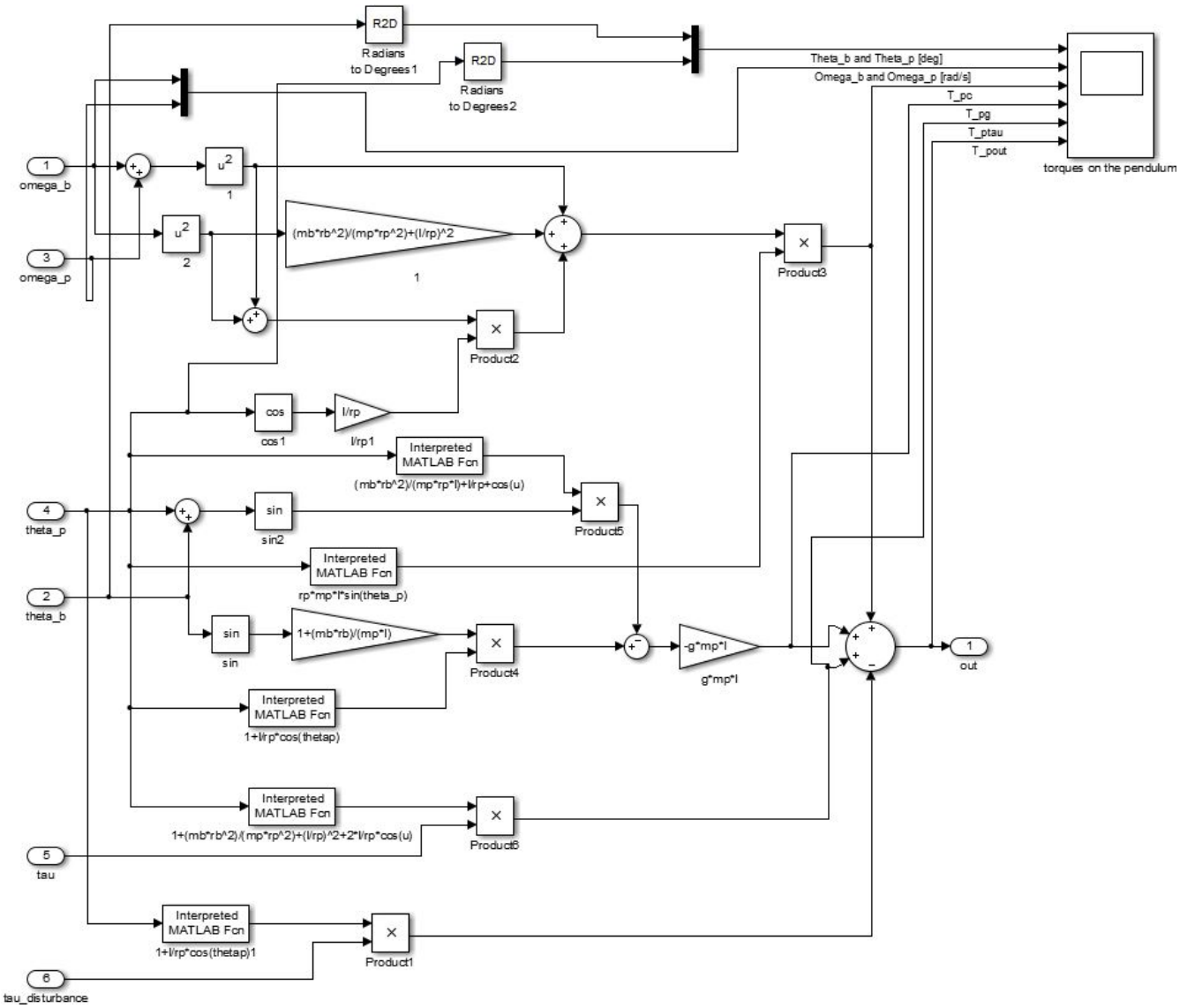


Figure B.2: Implemention of Right Hand Side of Eq. 3.20b in Simulink®

Figure B.2 illustrates the implementation of the right hand side of the differential equation in Eq. 3.20b, where the terms are categorize in centrifugal and Coriolis, gravity and actuator related torques, as the implementation in Figure B.1. Both implementations are utilized in the complete system model of Eq. 3.20 presented in Figure 4.21.

Appendix C

Appendix C

This appendix presents the data sheets for the gearbox and system actuator DC motor, presented in Section 2.4 and 2.5.

C.1 Harmonic Drive AG PMG-14A-72-S

On the next page, the data sheet of the gearbox is presented. The gearbox is produced by Harmonic Drive AG, and is the model PMG-14A-72-S, which is a PMG series gearbox of size 14A and gear ratio 72. The mark S implies that the gearbox is equipped with input shaft.

Bestellbezeichnungen und Technische Daten
Ordering Code and Technical Data

Bestellbezeichnungen

Tabelle / Table 109.1

Baureihe Series	Baugröße Size	Untersetzung Ratio						Version Version	Sonderausführung Special design
Mini-Getriebeboxen Baureihe PMG PMG series precision gearboxes	5A	50		80		100		M mit Eingangsnabe für Motoradaption Close coupled model for motor adaption	Nach Kundenanforderung According to customer requirements
	8A	50	72			100			
	11A	50	72			100		S mit Eingangswelle with input shaft	
	14A	50	72		88	100	110		

Bestellbezeichnung Ordering Code										
PMG	**–**	**8A**	**–**	**100**		**–**	**M**	**–**	**SP**

Leistungsdaten

Rating Table

Tabelle / Table 109.2

PMG Baugröße PMG size	Untersetzung Ratio i	Grenze für wiederholbares Spitzendrehmoment Limit for repeated peak torque T_R [Nm]	Grenze für Durchschnittsdrehmoment Limit for average torque T_A [Nm]	Nenndrehmoment Rated output torque T_N [Nm]	Grenze für Kollisionsdrehmoment Limit for momentary peak torque T_M [Nm]	Nenndrehzahl [min^{-1}] Rated input speed [rpm]	Maximale Antriebsdrehzahl Fettschmierung [min^{-1}] Max. input speed grease lubrication [rpm]	Grenze für mittlere Antriebsdrehzahl [min^{-1}] Limit for Average Input Speed [rpm]	Abtriebswelle Output shaft Max. radiale Last[1] / Max. radial load[1] [N]		Antriebswelle Typ S Input shaft type S Max. radiale Last[1] / Max. radial load[1] [N]		Massenträgheitsmoment[2] Moment of inertia[2] Typ M / Type M [x 10^{-4} kgcm^2]		Gewicht Weight Typ M / Type M [kg]	
									Max. radiale Last[1] / Max. radial load[1]	Max. axiale Last / Max. axial load	Max. radiale Last[1] / Max. radial load[1]	Max. axiale Last / Max. axial load	Typ M / Type M	Typ S / Type S	Typ M / Type M	Typ S / Type S
5	50	0,3	0,3	0,2	0,4	4500	10000	4900	59	29	8	5	2,5	2,5	0,03	0,031
	80	0,45	0,45	0,3	0,6	4500										
	100	0,55	0,55	0,3	0,7	4500										
8	50	1,9	1,9	1,5	2,5	3500	6000	3500	196	98	10	5	30	30	0,12	0,125
	72	2,4	2,3	2,0	3,1	3500										
	100	2,7	2,7	2,0	3,8	3500										
11	50	5,0	4,7	2,5	6,8	3500	5000	3500	245	196	20	10	120	140	0,25	0,27
	72	5,6	5,4	4,0	8,8	3500										
	100	7,9	7,6	4,0	10,8	3500										
14	50	9,8	7,0	5,4	14,0	3500	5000	3500	392	392	29	10	330	340	0,42	0,495
	72	11,8	9,0	7,8	16,0	3500										
	88	12,7	11,0	7,8	18,0	3500										
	100	14,7	11,0	7,8	20,0	3500										
	110	14,7	11,0	7,8	20,0	3500										

[1] Die maximale radiale Last bezieht sich auf die Wellenmitte der Antriebs- bzw. Abtriebsseite.
[2] Massenträgheitsmoment auf der Antriebsseite.

[1] The radial load is based on a force applied at the midpoint of the shaft extension.
[2] The moment of inertia is measured at the input of the gearbox.

Siehe „Erläuterungen zu Technischen Daten" im Kapitel „Projektierung mit Harmonic Drive Getrieben".

Please refer to the notes on "Understanding the Technical Data" in section "Engineering Data for Harmonic Drive Gears".

C.2 Dunkermotoren DC Motor GR 63x55 Performance Data Sheet

On the next page, the performance data sheet of the system actuator DC motor is attached. The system parameters applied in Chapter 2.5 is with the nominal battery voltage of 24 [V]. These parameters from the data sheet are listed in Table 2.1.

D.C. Motors

Series GR 63 x 25 and GR 63 x 55

The type GR 63 motors have permanent magnets and high power at small volume and thus especially suited for industrial applications, such as
– Computer and office machines
– Pumps and compressors
– Industrial blowers
– Welding technics
– General machine construction
– Medical equipment
– Door operating drives

Design
These motors are rigidly built and need no maintenance during their lifetime.

The rotor runs in ball bearings. The bearing plates are of die cast zinc.

Depending on case-length two versions are available with max. continuous torque of 14 Ncm resp. 28 Ncm.

The motors can be combined with different gears and actual value encoders resp. generators.

The GR 63 series motors can be delivered with brakes and/or actual value encoders resp. generators.

Standard version without second drive straft and without connector.

Features
– Mount-on dimensions according to DIN 42 016.
– Independent of mounting position.
– Clockwise and counter-clockwise rotation.
– Insulation according to VDE 0530, insulation class E.
– Surface protection.

Standard program
Motors with nominal voltages 12 V, 24 V, 40 V and 60 V represent our standard motor program and should preferably be used.

Load characteristics
The characteristics are examples for the standard program with the possible winding configurations of the motors, type GR 63.

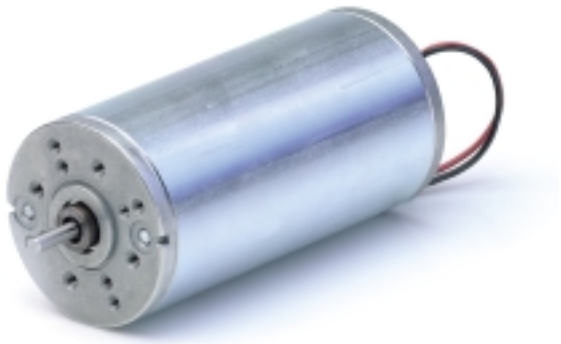
Angle dimensions and angle offset
See page 2

Performance data of motor GR 63 x 25					
Nominal voltage	**12**	**24**	**40**	**60**	V
Nominal speed 2)	3100	3300	3500	3300	min^{-1}
Nominal torque 2)	13,7	14	13,3	14,5	Ncm
Nominal current 2)	5,2	2,7	1,65	1,1	A
Demagnetization current 1)	50	24	16	9,5	A
No load speed 1)	3600	3600	3800	3600	min^{-1}
No load current 1)	0,6	0,36	0,205	0,135	A
Starting torque 1)	82	108	118	116	Ncm
Efficiency 2)	71	74	74	76	%
Moment of inertia	400	400	400	400	gcm^2
Weight	1,2	1,2	1,2	1,2	kg

Performance data of motor GR 63 x 55					
Nominal voltage	**12**	**24**	**40**	**60**	V
Nominal speed 2)	3000	3350	3450	3350	min^{-1}
Nominal torque 2)	24	27	27	28,3	Ncm
Nominal current 2)	8,7	4,9	2,95	2,0	A
Demagnetization current 1)	66	33	20	13	A
No load speed 1)	3500	3650	3600	3600	min^{-1}
No load current 1)	0,8	0,4	0,28	0,2	A
Starting torque 1)	202	211	210	200	Ncm
Efficiency 2)	80,5	80	82	82	%
Moment of inertia	750	750	750	750	gcm^2
Weight	1,7	1,7	1,7	1,7	kg

Shaft load capacity axial max.	150 N
Shaft load capacity radial max.	150 N applied 20 mm from mounting surface

All output data are referred to 1) $\partial_R = 20\,°C$ resp. 2) $\Delta\partial_w = 100\,K$



Appendix D

Appendix D

The simulation models designed throughout the semester is included in the zip-file, where the controller implementations in Matlab®/Simulink® are sorted into folders as following:

Controller by Kobayashi et al.:

The folder includes the controller implementation based on Kobayashi et al., presented in Section 4.2. This controller implementation is simulated with the bicycle system model with matrix representation, as given in Eq. 3.8 and 3.9.

Controller by Lai et al.:

The folder includes the controller implementation based on Lai et al., presented in Section 4.1. This controller implementation is simulated with the bicycle system model with matrix representation, as given in Eq. 3.8 and 3.9.

Partially Feedback Linearized System Controllers:

The folder includes the controller implementation of the partially feedback linearized system controllers presented in Section 4.3. In the controller implementation one can choose between the different reference input functions of u_0, presented in Section 4.3.2. This controller implementation is simulated with the State Space Differential Equation system, presented in Eq. 3.20. Note that the LQR-controller utilized in the stabilization controller is the balance controller presented in Eq. 4.37. The implementation of the partially feedback linearized LQR-controller of Eq. 4.40 is given in the additional system model in the Simulink®-file.

State Space Differential Equations System Model

This folder includes the system model presented in Section 3.8.1, with the set of differential equations utilized in Section 4.3.1.

System Model With Matrix Representation

This folder includes the system model on matrix form, presented in Section 3.5, and utilized in the simulations of the controller implementations based on Lai et al. and Kobayashi et al., in Section 4.1 and 4.2.

A "*readme.txt*"-file is included for description of how to run the system simulations. execution

Bibliography

[1] Nils A Andersen, Lars Skovgaard, and Ole Ravn. Dtu acrobot. Video on YouTube ©, Technical University of Denmark, 2010. https://www.youtube.com/watch?v=sMZRnE3q72c.

[2] Nils A Andersen, Lars Skovgaard, and Ole Ravn. Control of an under actuated unstable nonlinear object. In *Experimental Robotics VII*, pages 481–490. Springer, 2001.

[3] Jens G. Balchen, Trond Andresen, and Bjarne A. Foss. *Reguleringsteknikk*. Institutt for Teknisk Kybernetikk, NTNU, Trondheim, Norway, 5th edition, 2003.

[4] Lasse Bjermeland. Modeling, simulation and control system for an autonomous bicycle. Master's thesis, NTNU - Norwegian University of Science and Technology, Trondheim, June 2006.

[5] Chi-Tsong Chen. *Linear System Theory And Design*. Oxford University Press, Inc., Madison Avenue, New York, 3rd edition, 2009.

[6] Alessandro De Luca and Giuseppe Oriolo. Motion planning under gravity for underactuated three-link robots. In *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference*, volume 1, pages 139–144, Oct. and Nov. 2000.

[7] Olav Egeland and Jan Tommy Gravdahl. *Modeling and Simulation for Automatic Control*. Marine Cybernetics AS, Trondheim, Norway, 1st edition, 2002.

[8] Hassan K. Khalil. *Nonlinear Systems*. Prentice Hall, Inc., Upper Saddle River, New Jersey, 3rd edition, 2002.

[9] T Kobayashi, T Komine, S Suzuki, M Iwase, and K Furuta. Swing-up and balancing control of acrobot. In *SICE 2002. Proceedings of the 41st SICE Annual Conference*, volume 5, pages 3072–3075. IEEE, 2002.

[10] Xu-zhi Lai, Min Wu, and Jin-hua She. Control of acrobot based on lyapunov function. *Journal of Central South University of Technology*, 11(2):210–215, 2004.

[11] Xu-zhi Lai, Yun-xin Wu, Jin-hua She, and Min Wu. Control design and comprehensive stability analysis of acrobots based on lyapunov functions. *Journal of Central South University of Technology*, 12(1):210–216, 2005.

[12] Jørgen Herje Nilsen. Stabilization of autonomous bicycle modeled as an inverted double pendulum. Technical report, NTNU - Norwegian University of Science and Technology, Trondheim, December 2013.

[13] Roy Nilsen. *Kompendium TET4120 Elektriske Motordrifter*. NTNU - Norwegian University of Science and Technology, Institutt for Elkraftteknikk, 2004.

[14] Dag Christian Ånnestad. Autonomous bicycle - the first self balanced ride. Master's thesis, NTNU - Norwegian University of Science and Technology, May 2011.

[15] Giuseppe Oriolo and Yoshihiko Nakamura. Control of mechanical systems with second-order nonholonomic constraints: underactuated manipulators. In *Decision and Control, 1991., Proceedings of the 30th IEEE Conference*, volume 3, pages 2398–2403, December 1991.

[16] Jin-Hua She, Xu-Zhi Lai, Xin Xin, and Li-Li Guo. A rewinding approach to motion planning for acrobot based on virtual friction. In *Industrial Technology (ICIT), 2010 IEEE International Conference*, pages 471–476, March 2010.

[17] Mark. W Spong. Underactuated mechanical systems. In *Control Problems in Robotics and Automation*, pages 135–150. Springer-Verlag.

[18] Mark. W Spong. Partial feedback linearization of underactuated mechanical systems. In *Intelligent Robots and Systems '94. 'Advanced Robotic System and the Real World', IROS '94. Proceedings of the IEEE/RSJ/GI International Conference*, volume 1, pages 314–321, September 1994.

[19] Mark W. Spong. The swing up control problem for the acrobot. *Control Systems, IEEE*, 15(1):49–55, Feb 1995.

[20] Mark. W Spong. Energy based control of a class of underactuated mechanical systems. In *IFAC World Congress*, pages 431–435, 1996.

[21] Mark. W Spong and Daniel J. Block. The pendubot: a mechatronic system for control research and education. In *Decision and Control, 1995., Proceedings of the 34th IEEE Conference*, volume 1, pages 555–556, December 1995.

[22] Mark W. Spong, Seth Hutchinson, and Mathukumalli Vidyasagar. *Robot Modeling And Control*. John Wiley & Sons, Inc., Hooboken, New Jersey, 1st edition, 2006.

[23] Bjørnar Vik. *Integrated Satellite and Inertial Navigation Systems*. NTNU - Norwegian University of Science and Technology, Department of Engineering Cybernetics, 2012.