Zumobot case study

Control Theory Module

Javier Reyes

Hari Kumar Venkatesh

Group No.

Homework assignment

Fachhochschule Dortmund
University of Applied Sciences and Arts

Master Embedded Systems for Mechatronics

Dortmund University of Applied Sciences and Arts

Germany

August 10, 2017

Contents

List of Figures

Introduction

In the control theory field, there are several case studies with different characteristics that allow the modeling and testing of different control concepts and strategies. One of this common study cases is the Inverted Pendulum, as it presents an unstable open-loop characteristic but it is also possible to stabilize it on a closed-loop configuration. The inverted pendulum system, as its name can lead, is typically a wheeled body that is kept in an unstable position from which, without any external signal or command, the body will inevitably fall down. The equilibrium point of the body should be maintained by means of an external element, that actuates over the body, commanded by a programmatic logic.

In the present document the typical control flow is presented, where the goal is initially to define the physical characteristics of a test device, a small robot 32U4 from Pololu. Then a mathematical model that represents the dynamic behavior of the robot when maintained at its highest position is obtained. From this model, a frequency domain representation is obtained, from which finally a controller can be designed that provides the adequate signals to the system in order to maintain the robot in its unstable equilibrium position.

The Zumo 32U4 robot is a complete and versatile robot controlled by a microcontroller and additional circuitry. The Zumo 32U4 robot can be programmed to carry out a constant task, allowing a digital implementation of a PID control of the angle of the robot, by actuating on the motors of the robot.

The result obtained showed some mathematical differences between the standard models found in the literature and the specific robot used. Some control limitations were also stated when the real execution was compared with the simulated response. Some future work and considerations for better alternatives are made.

1 System Analysis

An inverted pendulum is basically a consistent mass on the ground (usually wheeled), connected through a frictionless joint to a pendulum, so that the pendulum can freely rotate around the joint and fall down by its own weight. The goal in this system is to provide an horizontal force to the mass on the ground with a direction contrary to the inclination of the pendulum with respect to the vertical axis, so that the pendulum holds in its highest position.

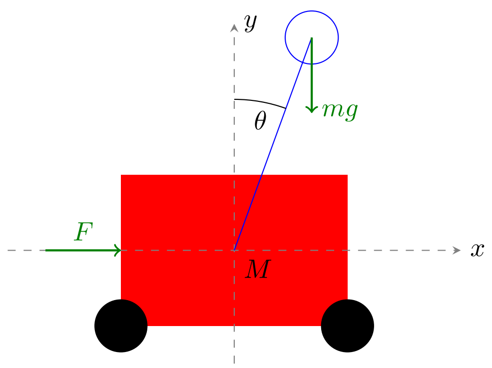

Figure 1.1: Physical representation of the inverted pendulum

1.1 Zumobot case study

The case study in this work is based on a commercial Zumobot 32U4 (See figure 1.2) robot from Pololu manufacturer. It consist on a rigid case that holds 2 brushless DC motors, an Arduino-compatible ATmega32U4 microcontroller and a set of batteries to power up the system. The motors are connected to a belt that rotates between to sets of wheels, sufficiently high to make the Zumobot able to stay in a vertical position without any contact between the case and the ground. The stucture is well suited for an inverted pendulum configuration, considering its physical characteristics and its simple programmability.

The goal position of the Zumobot is a vertical position with respect to the vertical axis. The Zumobot needs to be previously adjusted (frontal sensors holder dismounted), as the frontal part will face the ground, and the posterior part will be held up.

The device includes several helpful sensors and peripherals:

- AVR ATmega 32U4 microcontroller, with 16MHz crystal oscillator.
- Two micrometal gearmotors, driven by on-board TI DRV8838 motor divers.
- 3-axis accelerometer + 3-axis gyroscope

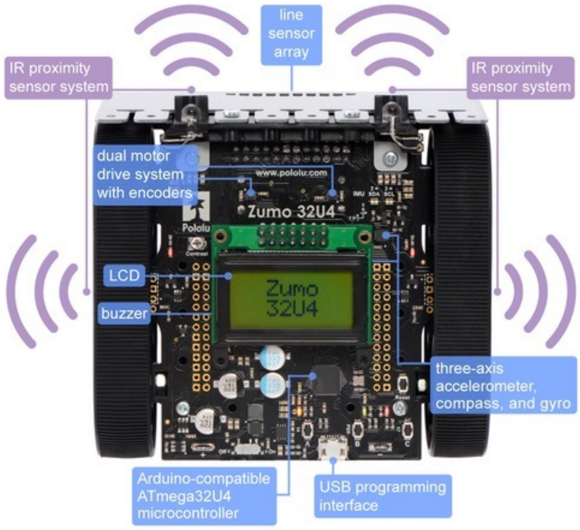

Figure 1.2: Pololu Zumobot

To power the motors, the microcontroller generates a PWM digital signal and a direction bit for each wheel (left and right). The Arduino environment provides a library to configure and control the PWM value with low effort. Along with the corresponding firmware, the whole actuator part of the system is complete.

The sensor part of the system uses the accelerometer and gyroscope to provide a value of the angle in the y axis, which is the axis that shows the angle θ defined in figure 1.1. The firmware in the Arduino controller gets the value from the gyroscope, and performs an adjustment of the value based on the accelerometer reading. It should be noted that the manufacturer of the robot states in the technical documentation[Pol] that the accelerometer and gyroscope readings are likely to be influenced by external noise from the DC motors and the batteries, and the values obtained should only be considered for rough estimation.

1.2 Control system

The desired control system is then defined as shown in the figure 1.4.

The dynamics of the system can be divided into the following elements:

- DC motor behavior to convert a DC voltage to a force
- Robot behavior defined by the physical characteristics

With this clarification, the control diagram of the system becomes:

To design the controller for the system, the dynamics of the motor and robot need to be defined.

Figure 1.3: Stabilized Zumobot position

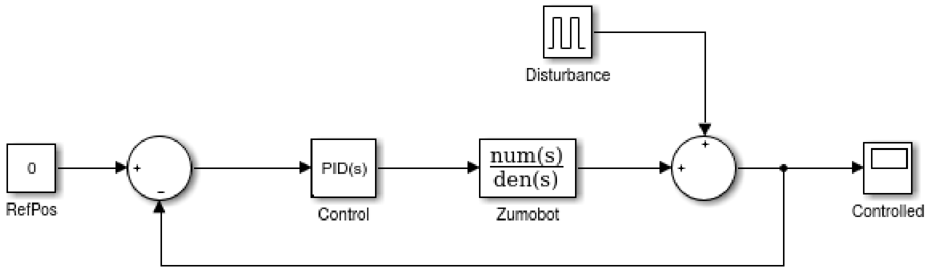

Figure 1.4: Control diagram of the inverted pendulum

This is done in the next chapter by two different approaches.

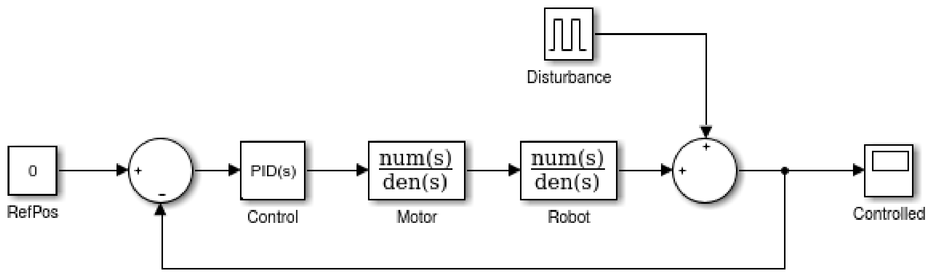

Figure 1.5: Control diagram of the inverted pendulum - analysis elements

2 Mathematical Development

Through the numerous literature it is possible to find several approaches to obtain a model for the Inverted Pendulum elements. For this work the focus is mostly on the dynamic behavior of the robot, as it is the one that influences the most the response of the system. To completely model the dynamic behavior of the robot, we need to consider the equations that govern the movement as a rigid body.

2.1 First approach: Forces analysis

The first approach considered here is described in [Sul03]. The methodology uses traditional dynamic physics to obtain the equations of motion. The final equation is then linearized and transformed into complex frequency domain.

2.1.1 Dynamic behavior

The equations of motion are obtained from the sum of forces in the cart for the horizontal direction (see figure 2.1).

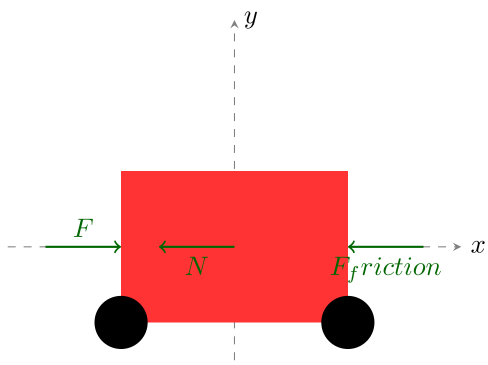

Figure 2.1: Cart forces

$$F - b \cdot \dot{x} - N = M \cdot \ddot{x} \tag{2.1}$$

Now considering the pendulum itself, the force applied in the horizontal direction due to the momentum of the pendulum is determined as:

$$\tau = r \cdot F = I \cdot \ddot{\theta} \tag{2.2}$$

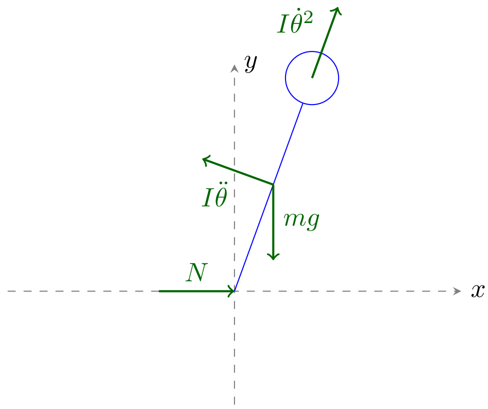

Figure 2.2: Pendulum forces

Given the fact that the moment of inertia of a pendulum of mass m is defined as $I = m \cdot L^2$, the previous equation can be rewritten as:

$$F = \frac{I \cdot \ddot{\theta}}{r} = \frac{m \cdot l^2 \cdot \ddot{\theta}}{l} = m \cdot l \cdot \ddot{\theta} \tag{2.3}$$

Obtaining the component of the force defined in 2.3 in the horizontal direction:

$$F = m \cdot l \cdot \ddot{\theta} \cdot \cos\theta \tag{2.4}$$

Now, the component of the centripetal force acting on the pendulum is similar to the one in 2.3, but the horizontal component of this force is:

$$F = m \cdot l \cdot \dot{\theta}^2 \cdot \sin\theta \tag{2.5}$$

Summing the defined forces present in the horizontal direction of the pendulum in 2.4 and 2.5, we obtain the following expression:

$$N = m \cdot \ddot{x} + m \cdot l \cdot \ddot{\theta} \cdot \cos\theta - m \cdot l \cdot \dot{\theta}^2 \cdot \sin\theta \tag{2.6}$$

Now we can substitute 2.6 into 2.1, we obtain the first equation of motion:

$$F = (M + m)\ddot{x} + b \cdot \dot{x} + m \cdot l \cdot \ddot{\theta} \cdot \cos\theta - m \cdot l \cdot \dot{\theta}^2 \cdot \sin\theta \tag{2.7}$$

To get the second equation of motion, we sum the forces perpendicular to the pendulum. The vertical components of this forces are considered here to get:

$$P \cdot \sin\theta + N \cdot \cos\theta - m \cdot g \cdot \sin\theta = m \cdot l \cdot \ddot{\theta} + m \cdot \ddot{x} \cdot \cos\theta \tag{2.8}$$

To get rid of the P and N terms, sum the moments around the center of gravity of the pendulum:

$$-P \cdot l \cdot \sin\theta - N \cdot l \cdot \cos\theta = I \cdot \ddot{\theta} \tag{2.9}$$

Summing up the equations 2.8 and 2.9, we obtain the second equation of movement:

$$(I + m \cdot l^2)\ddot{\theta} + m \cdot g \cdot l \cdot \sin\theta = -m \cdot l \cdot \ddot{x} \cdot \cos\theta \tag{2.10}$$

The obtained equations are non-linear, so they are linearized around the operating point, defined as the top vertical position or πrad from the stable equilibrium position. We need also to define a small angle deviation from the top vertical position, so that $\theta = \pi + \phi$.

Under this circumstances, we can deduce that $\cos\theta \approx -1$, $\sin\theta \approx -\phi$, and $\dot{\theta}^2 \approx 0$. Applying this relations into our movement equations, we obtain:

$$F = (M + m)\ddot{x} + b \cdot \dot{x} - m \cdot l \cdot \ddot{\phi} \tag{2.11}$$

$$(I + m \cdot l^2)\ddot{\phi} - m \cdot g \cdot l \cdot \phi = m \cdot l \cdot \ddot{x} \tag{2.12}$$

To obtain the transfer function of the linearized system of equations analytically, we perform the Laplace transform of the system equations:

$$F(s) = (M + m)s^2 \cdot X(s) + b \cdot s \cdot X(s) - m \cdot l \cdot s^2 \cdot \Phi(s) \tag{2.13}$$

$$(I + m \cdot l^2)s^2 \cdot \Phi(s) - m \cdot g \cdot l \cdot \Phi(s) = m \cdot l \cdot s^2 \cdot X(s) \tag{2.14}$$

To unify the equations, we solve 2.13 for $X(s)$ and then replace it into 2.14, obtaining:

$$\frac{\Phi(s)}{F(s)} = \frac{m \cdot l \cdot s}{q \cdot s^3 + b(l + m \cdot l^2)s^2 - m \cdot g \cdot l(M + m)s - b \cdot m \cdot g \cdot l} \tag{2.15}$$

Where:

$$q = (M + m)(l + m \cdot l^2) - (m \cdot l)^2 \tag{2.16}$$

Assuming a coefficient of friction as zero, we can represent the equation as:

$$\frac{\Phi(s)}{F(s)} = \frac{K_p}{\frac{s^2}{A_p^2} - 1} \tag{2.17}$$

With:

$$K_p = \frac{1}{(M+m)g}; A_p = \pm\sqrt{\frac{(M+m)m \cdot g \cdot l}{(M+m)(l + m \cdot l^2) - (m \cdot l)^2}} \tag{2.18}$$

Considering that the Zumobot is a unique mass element, where a differentiation of cart and pendulum masses is not feasible, an adjustment of the equation 2.18 is made, so that the mass considered is the total mass of the robot. This is possible as the actuator system should face the entire mass of the robot, and the pendular mass is also corresponding to the complete body.

$$K_p = \frac{1}{mg}; A_p = \pm\sqrt{\frac{m^2 \cdot g \cdot l}{m(l + m \cdot l^2) - (m \cdot l)^2}} \tag{2.19}$$

2.1.2 Actuator system

The actuation mechanism consist on a DC motor that drives a belt system around two wheels. The overall transfer function of the actuation mechanism will depend upon the motor and the belt system.

The torque to be delivered by the motor is:

$$T_L = (M+m)r^2\dot{\omega} \tag{2.20}$$

The relation between Torque and Force can be expressed as:

$$T_L \propto r^2; F \propto r \tag{2.21}$$

The motor dynamics can be represented with the well known transfer function, as in [Sul03] [Zac]:

$$\omega(s) = K_m \frac{V(s)}{\tau s + 1} \tag{2.22}$$

Where τ is the time constant and depends on the load, and K_m is the steady-state gain of the motor.

$$\frac{T(s)}{E(s)} = K_m \frac{(M+m)r \cdot s}{\tau s + 1} \tag{2.23}$$

2.1.3 Simulation

The obtained model is analyzed with computational tools (MATLAB), to validate its behavior and calculate an appropriate controller. As shown in figure 2.3, the system has a pole in the right part of the complex plane, fitting in the definition of an unstable system.

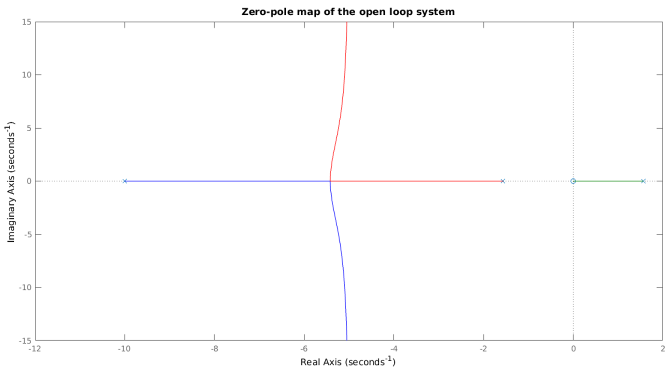

Figure 2.3: Zero-pole diagram for the open-loop system

The open-loop response shown in figure 2.4 confirms the unstable behavior.

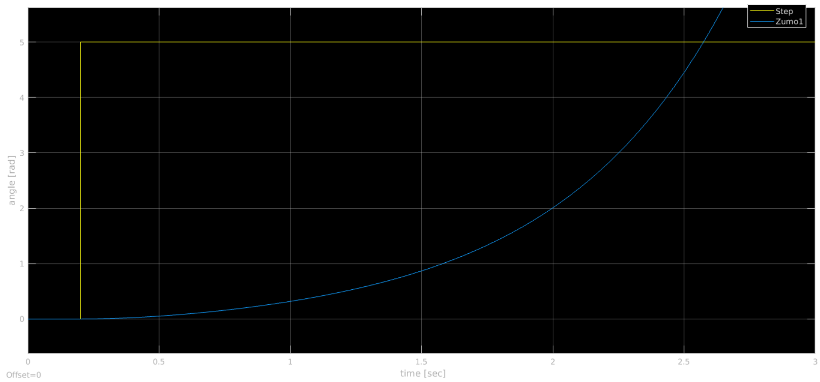

Figure 2.4: Open-loop step response of the system

2.2 Second approach - Euler-Lagrange

2.2.1 Dynamic behavior

To represent the dynamics of the system as in [Jer12] and [Lun02], the initial definition is the natural form of the Lagrangian in classical mechanics:

$$\mathcal{L} = E_k - E_p \tag{2.24}$$

Where $E_k = \frac{1}{2}mv^2$ and $E_p = mgh$. Equation 2.24 can then be rewriten as:

$$\mathcal{L} = \frac{1}{2}mv^2 - mgh \tag{2.25}$$

The Euler-Lagrange equation states that:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}}\right) = \frac{\partial \mathcal{L}}{\partial \theta} \tag{2.26}$$

The pendulum is a stiff bar of lenght L which is supported at one end by a frictionless pin. From the figure 2.5 we can state that:

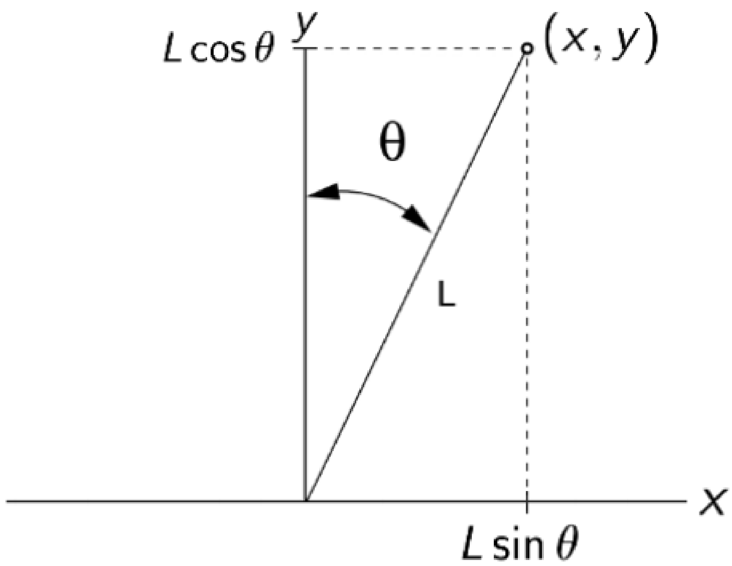

Figure 2.5: Pendulum position

$$\begin{aligned} x &= L\sin\theta \quad \dot{x} = L\cos\theta\dot{\theta} \\ y &= L\cos\theta \quad \dot{y} = -L\sin\theta\dot{\theta} \end{aligned} \tag{2.27}$$

Velocity is a vector representing the change in the position in the coordinates x and y. Hence:

$$v^2 = \dot{x}^2 + \dot{y}^2 \tag{2.28}$$

With the coordinates found in 2.27, we can substitute in 2.28 to obtain:

$$
\begin{aligned}
v^2 &= L^2 \cos\theta^2\dot{\theta}^2 + L^2 \sin\theta^2\dot{\theta}^2 \\
v^2 &= L^2\dot{\theta}^2
\end{aligned}
\tag{2.29}
$$

Substituting 2.27 and 2.29 into 2.25, we obtain:

$$
\mathcal{L} = \frac{1}{2}mL^2\dot{\theta}^2 - mgL\cos\theta \tag{2.30}
$$

To perform the Euler-Lagrange equation presented in 2.26, we need to compute the partial derivatives. First we compute the left part:

$$
\frac{\partial\mathcal{L}}{\partial\theta} = mgL\sin\theta \tag{2.31}
$$

Now we compute the inner part of the right element:

$$
\frac{\partial\mathcal{L}}{\partial\dot{\theta}} = mL^2\dot{\theta} \tag{2.32}
$$

And now we can compute the outer derivative of 2.32:

$$
\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{\theta}}\right) = mL^2\ddot{\theta} \tag{2.33}
$$

Now that we have all the terms, we can write the equation:

$$
\begin{aligned}
\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{\theta}}\right) &= \frac{\partial\mathcal{L}}{\partial\theta} \\
mL^2\ddot{\theta} &= mgL\sin\theta \\
\ddot{\theta} &= \frac{g}{l}\sin\theta
\end{aligned}
\tag{2.34}
$$

Following the same procedure, we obtain the expression for the angular acceleration produced by the cart:

$$
\ddot{\theta}_x = \frac{\ddot{x}}{l}\cos\theta \tag{2.35}
$$

The total accelerations present on the pendulum can then be stated as:

$$
\ddot{\theta} = \ddot{\theta}_g + \ddot{\theta}_x = \left(\frac{g}{l}\right)\sin\theta - \left(\frac{\ddot{x}}{l}\right)\cos\theta \tag{2.36}
$$

The obtained equation is non-linear, so we apply an approximation based on the fact that the operation point implies an angle $\theta \approx 0$. It means that $\sin\theta \approx \theta$ and $\cos\theta \approx 1$. Applying this on 2.36, we obtain:

$$l\ddot{\theta} - g\theta = -\ddot{x} \tag{2.37}$$

To obtain the transfer function, we perform the Laplace transform on 2.37:

$$ls^2\Theta(s) - g\Theta(s) = -s^2X(s) \tag{2.38}$$

Now we solve for the variables Θ and X:

$$\frac{\Theta(s)}{X(s)} = \frac{-s^2}{ls^2 - g} \tag{2.39}$$

2.2.2 Actuator system

For the motor modeling, a recommended approach [Lun02] is used. The motor is diven by a voltage proportional to the angle:

$$\frac{X(s)}{V(s)} = \frac{k_M}{s(\tau_M s + 1)} \tag{2.40}$$

2.2.3 Simulation

With the obtained model, a simulation in open-loop condition is run. As expected, the system contains a pole in the right part of the complex plane.

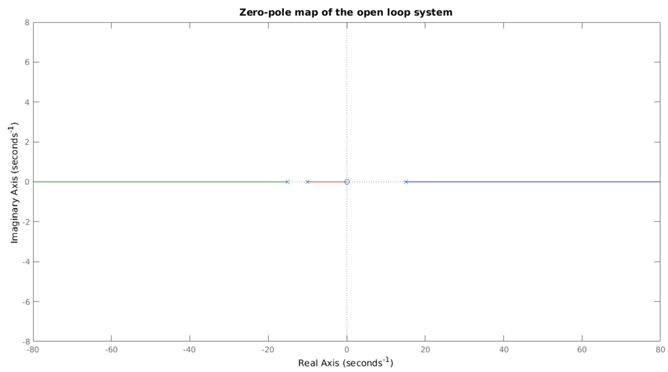

Figure 2.6: Zero-pole diagram for the open-loop system

The system response to a step input shows total unstability, as in figure 2.7.

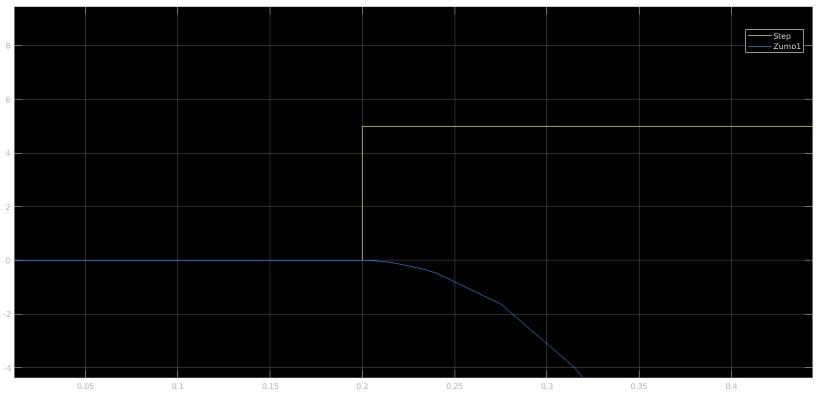

Figure 2.7: Open-loop step response of the system

3 Control Design

From the obtained model, it is clear that the system needs a controller to govern the signal control, proportionally to the error (the pole on the right part needs to be canceled). For that reason, a PID parallel controller is the first step in the control work flow.

3.1 Simulated controller

With help of the computational tool (MATLAB), it is nowadays easy to tune a controller with desired specifications. The current version includes a GUI tool that allows to move around the different control specifications for the selected system, checking continuously the result. It certainly reduces time and effort in the calculations and design.

Given the

3.2 Implemented controller

3.3 Results analysis

The results of this work were not as satisfactory as expected, due to the bad behavior obtained from the Zumobot with the different designed controllers. It is clear that several reasons contribute to this factor, some of them can be stated as follows.

3.3.1 Insufficient model

3.3.2 Insufficient control technique

3.3.3 subsection name

Bibliography

[1] *Pololu zumo 32u4 robot's user guide*, Pololu Corporation, 2001.

[2] K. Sultan, *The inverted pendulum, analysis, design and implementation*, 2003.

[3] L. Zaccarian. (). Dc motors: Dynamic model and control techniques, [Online]. Available: `http://homepages.laas.fr/lzaccari/seminars/DCmotors.pdf`.

[4] F. Jeremic, *Derivation of equations of motion for inverted pendulum problem*, 2012.

[5] K. Lundberg, *The inverted pendulum system*, 2002.

[6] M. Hasan, C. Saha, M. M. Rahman, M. R. I. Sarker, and S. K. Aditya, "Balancing of an inverted pendulum using pd controller," 2012.

[7] A. Castro, "Modeling and dynamic analysis of a two wheeled inverted pendulum," Master Thesis, Georgia University of Technology, 2012.

[8] H. Hellman and H. Sunnerman, "Two-wheeled self-balancing robot," Bachelor Thesis, 2015.

[9] K. G. Libbrecht and V. de Oliveira Sannibale, *Classical mechanics - the inverted pendulum*, 2012.

[10] D. Deley. (). Controlling the inverted pendulum: An example of a digital feedback control system, [Online]. Available: `http://daviddeley.com/pendulum/pendulum.htm`.

[11] Dorf and Bishop. (). Dc motor transfer functions, [Online]. Available: `http://edge.rit.edu/edge/P14453/public/Research/DC%20Motor%20Transfer%20Function%20Example.pdf`.

[12] R. L. Robles and Y. A. Shardt. (). Linear motion inverted pendulum - derivation of the state-space model, [Online]. Available: `http://jtjt.pl/www/pages/odwrocone-wahadlo/LMIP.pdf`.

[13] N. Instruments. (2016). Modeling dc motor position, [Online]. Available: `http://www.ni.com/tutorial/6859/en/`.

[14] T.-B. Nguyen. (2014). Segbot: Modeling, [Online]. Available: `http://www.arxterra.com/segbot-modeling/`.