Inverted pendulum

Anton Prokopev

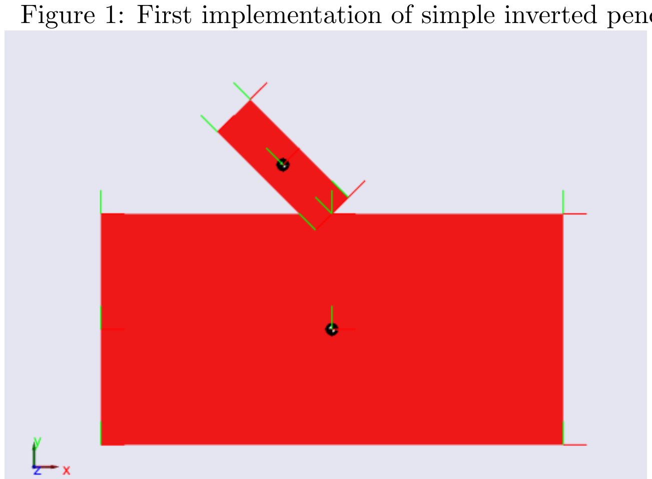
May 2018

1 Introduction

Inverted pendulum one of the most interesting and widespread task in Control Theory. An inverted pendulum is a pendulum that has its center of mass above its pivot point. It is unstable and without additional help will fall over. It can be suspended stably in this inverted position by using a control system to monitor the angle of the pole and move the pivot point horizontally back under the center of mass when it starts to fall over, keeping it balanced. The inverted pendulum is a classic problem in dynamics and control theory and is used as a benchmark for testing control strategies[3].

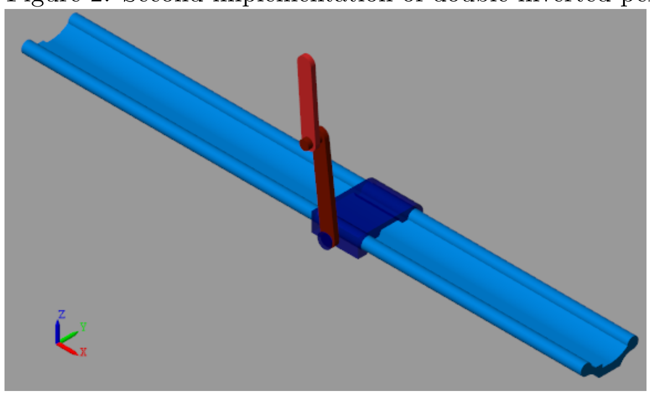
For this task I decided to use Matlab R2017a with Simulink v8.9 and Simscape v1 and v2 for simulation of pendulum. As a model a chose the inverted pendulum on a cart. There are several restrictions because of the simulator. There is no friction force in double inverted pendulum. And cart can move only along the one axis. First model was rectangular and flat and used Simscape Multibody v1.

Figure 1: First implementation of simple inverted pendulum



For the second part I used more convenient Simscape Multibody v2. It has some advantages: measurement angle is not limit by [-180,180] degrees. Simpler execution and simulation process.

Figure 2: Second implementation of double inverted pendulum



2 Goals

There are 4 main goals for this coursework. Implement:

1. Inverted pendulum
2. Swing up for inverted pendulum
3. Double inverted pendulum
4. Swing up for double inverted pendulum

And show how inverted pendulum can be controlled.

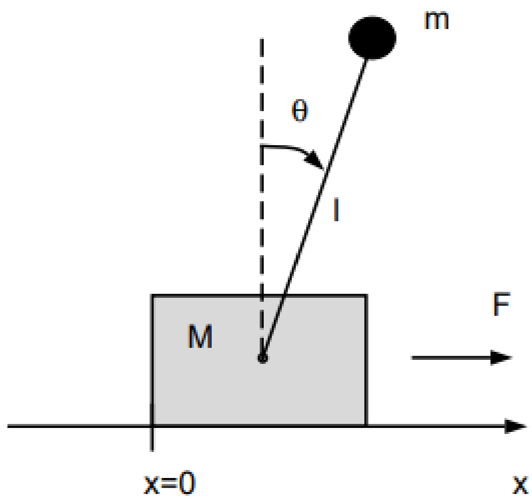
3 Inverted pendulum on a cart

3.1 Equations of motion

The equations of motion of inverted pendulums are dependent on what constraints are placed on the motion of the pendulum. Inverted pendulums can be created in various configurations resulting in a number of Equations of Motion describing the behavior of the pendulum.

Description of an inverted pendulum in a figure below.

Figure 3: Description of an inverted pendulum



With the notation x – cart position, θ – pendulum angle and F – applied force, the system can be described with the equations

The net torque of the system must equal the moment of inertia times the angular acceleration:

$$(M+m)\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F$$

$$l\ddot{\theta} + \ddot{x}\cos\theta - g\sin\theta = -f_\theta\dot{\theta}$$

where M and m denotes the cart and pendulum mass, respectively, l the pendulum length, g the gravitational constant and f_θ the friction coefficient for the link where the pendulum is attached to the cart.

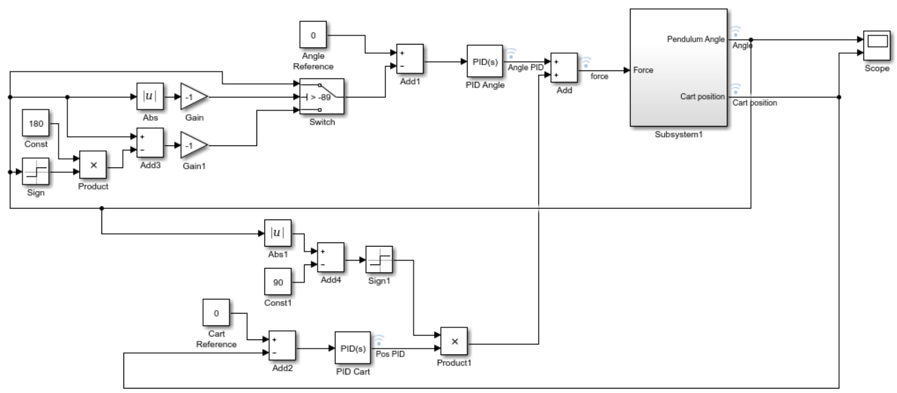
Let's build a nonlinear state space model:

$$x_1 = x, \quad x_2 = \dot{x}, \quad x_3 = \theta, \quad x_4 = \dot{\theta}$$

$$\begin{aligned}
\dot{x_1} &= x_2 \\
\dot{x_2} &= \frac{-mg\sin x_3\cos x_3 + mlx_4^2\sin x_3 + f_\theta x_4\cos x_3 + F}{M + (1-\cos {x_3}^2)m} \\
\dot{x_3} &= x_4 \\
\dot{x_4} &= \frac{(M+m)(g\sin x_3 - f_\theta x_4) - (lmx_4^2\sin x_3 + F)\cos x_3}{l(M + (1-\cos {x_3}^2)m)}
\end{aligned}$$

3.2 Model

Figure 4: Simulink model

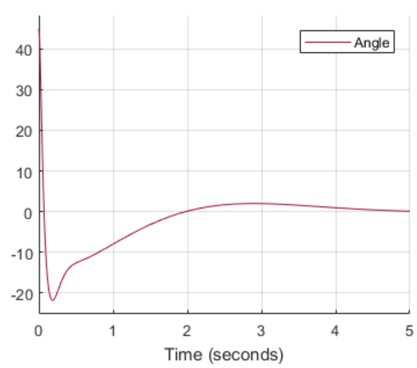


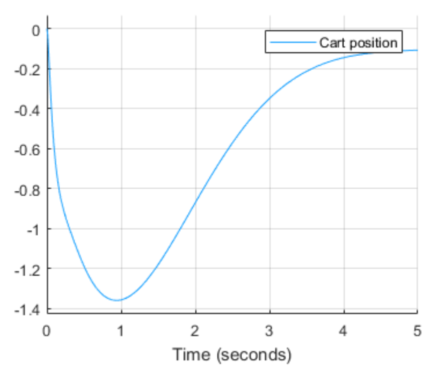
Main part of the model is "Subsystem1" - block of simulation of the inverted pendulum. It was taken from one of the tutorials. Also, this block was modified to provide more wide range of measurement angles. It was extended from $[-180, 180]$ to $[-360, 360]$. This changes are necessary for stability at point of -180 or 180 degree.

System is stabilized by two PID controllers. One for Angle, one for position of the cart. PID angle has $= 1, 1, 0.1$ coefficients for P, I, D respectively. And for position's PID values $= 10, 3, 3$.

For the initial conditions of

$$x = 0, \quad , \dot{x} = 0, \quad \theta = 45^\circ, \quad \dot{\theta} = 0$$





As we can see, the system with one joint can be controlled perfectly with simple PID controller.

To stabilize at the point with $\theta = 180°$, there is sign changer for PID coefficient. So system can be controlled at any initial point.

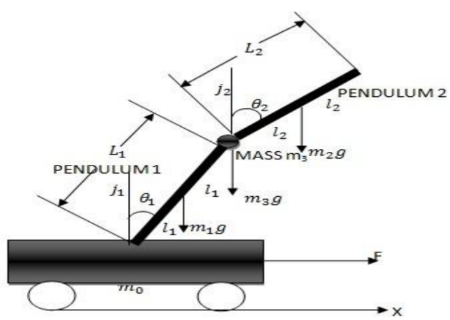
4 Double inverted pendulum

I assumed that swing up for pendulum with one joint is simple task and moved on toward the Double inverted pendulum (with swing up and swing a half up).

4.1 Math model

To control this system, its dynamic behavior must be analyzed first. The dynamic behavior is the changing rate of the status and position of the double inverted pendulum proportionate to the force applied. This relationship can be explained using a series of differential equations called the motion equations ruling over the pendulum response to the applied force. The double inverted pendulum is shown below.[7]

Figure 5: Schematic diagram of Double Inverted Pendulum



$M(m_1, m_2, m_3)$	Mass of the cart (first pole, second pole, joint)
θ_1, θ_2	The angle between pole 1(2) and vertical direction (degree)
$L_1(l_1), L_2(l_2)$	Length of pendulum first($2l_1$) and length of second pendulum ($2l_2$)
g	Center of gravity
F	Force applied to the cart

Table 1: Description of double inverted pendulum

To derive its equations of motion, one of the possible ways is to use Lagrange equations[5]:

$$\frac{d}{dt}\frac{dL}{d\dot{q_i}} - \frac{dL}{dq_i} = Q_i$$

Where L = T - V is a Lagrangian, Q is a vector of generalized forces (or moments) acting in the direction of generalized coordinates q and not accounted for in formulation of kinetic energy T and potential energy V. Kinetic and potential energies of the system are given by the sum of energies of cart and pendulums.

$$\boldsymbol{T} = \frac{1}{2}(m_0+m_1+m_2+m_3)\dot{x}^2 + (\frac{2}{3}m_1l_1^2+2m_2*l_1^2+2m_3l_1^2)\dot{\theta_1}^2 + \frac{1}{6}m_2l_2^2\dot{\theta_2}^2 + (m_1l_1+2m_2l_1+2m_3l_1)\dot{x}\dot{\theta_1}\cos\theta_1 + m_2l_2\dot{x}\dot{\theta_2} + 2m_2l_1l_2\cos(\theta_1-\theta_2)\dot{\theta_1}\dot{\theta_2}$$

$$\boldsymbol{V} = m_1gl_1\cos\theta_1 + 2m_3gl_1\cos\theta_1 + m_2g(2l_1\cos\theta_1 + l_2\cos\theta_2)$$

$$\boldsymbol{L} = \frac{1}{2}(m_0+m_1+m_2+m_3)\dot{x}^2 + (\frac{2}{3}m_1l_1^2+2m_2*l_1^2+2m_3l_1^2)\dot{\theta_1}^2 + \frac{1}{6}m_2l_2^2\dot{\theta_2}^2 + (m_1l_1+2m_2l_1+2m_3l_1)\dot{x}\dot{\theta_1}\cos\theta_1 + m_2l_2\dot{x}\dot{\theta_2} + 2m_2l_1l_2\cos(\theta_1-\theta_2)\dot{\theta_1}\dot{\theta_2} - m_1gl_1\cos\theta_1 - 2m_3gl_1\cos\theta_1 - m_2g(2l_1\cos\theta_1 + l_2\cos\theta_2)$$

Differentiating the Lagrangian by and $\dot{\theta}$ and θ yields Lagrange equation as:

$$\frac{d}{dt}\frac{dL}{d\dot{\theta_1}} - \frac{dL}{\theta_1} = 0$$

$$\frac{d}{dt}\frac{dL}{d\dot{\theta_2}} - \frac{dL}{\theta_2} = 0$$

The stationary point of the system is $(x, \theta_1, \theta_2, \dot{x}, \dot{\theta_1}, \dot{\theta_2}) = (0, 0, 0, 0, 0, 0, 0)$, introduce small deviation around a stationary point and Taylor series expansion; the stable control process of the Double Inverted Pendulums are usually $\cos(\theta_1 - \theta_2) \cong 1$, $\sin(\theta_1 - \theta_2) \cong 0$, $\cos\theta_1 \cong 1$, $\cos\theta_2 \cong 1$, $\sin\theta_1 \cong \theta_1$, $\sin\theta_2 \cong \theta_2$.

For the linearization I used a built-in matlab function (*linmod*) and simulation (*sim*). Result equation:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta_1} \\ \ddot{\theta_1} \\ \dot{\theta_2} \\ \ddot{\theta_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3.2670 & 0 & -0.1331 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 90.2007 & 0 & -17.9553 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -120.8744 & 0 & 143.2933 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta_1 \\ \dot{\theta_1} \\ \theta_2 \\ \dot{\theta_2} \end{bmatrix} + \begin{bmatrix} 0 \\ 1.5153 \\ 0 \\ 10.4519 \\ 0 \\ -14.0062 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} u(t)$$

4.2 Analysis of double inverted pendulum

4.2.1 Stability

The real part of eigenvalues of matrix A should be less than 0. The matrix has eigenvector:

$$\lambda = \begin{bmatrix} 0 \\ 0 \\ -13.0524 \\ -7.9453 \\ 13.0524 \\ 7.9453 \end{bmatrix}$$

It is obvious that matrix is not stable.

4.2.2 Controllability

The matrix Q_c should have $rank = n$.

$$Q_c = [B \quad AB \quad ... \quad ... \quad A^{n-1}B]$$

$$Q_c = \begin{bmatrix} 0 & 1.5153 & 0 & 36.0097 & 0 & 4336.77 \\ 1.5153 & 0 & 36.0097 & 0 & 4336.77 & 0 \\ 0 & 10.4519 & 0 & 1194.2524 & 0 & 166442.56 \\ 10.4519 & 0 & 1194.2524 & 0 & 166442.56 & 0 \\ 0 & -14.0062 & 0 & -3270.3555 & 0 & -612974.62 \\ -14.0062 & 0 & -3270.3555 & 0 & -612974.62 & 0 \end{bmatrix}$$

This matrix has $rank = 6 = n$

4.2.3 Observability

The matrix O_c should have $rank = n$.

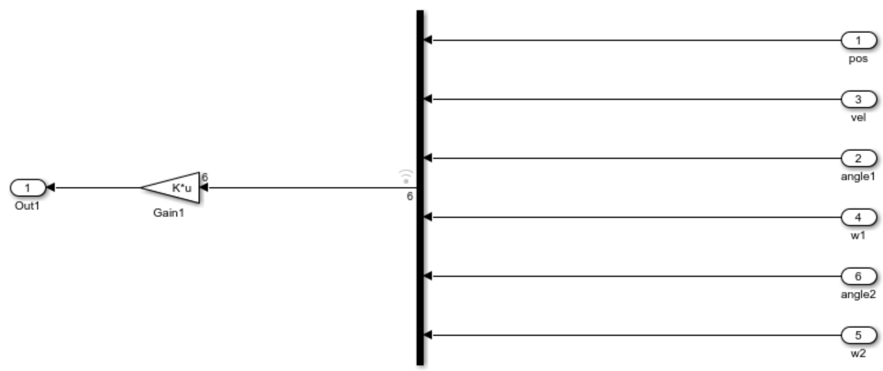
$$O_c = \begin{bmatrix} C \\ CA \\ ... \\ ... \\ CA^{n-1} \end{bmatrix}$$

Matrix will have size of $n * m \times n$. For this case it is 18×6, by obvious reasons I will not put it here but rank of this matrix equals 6. So, the system is fully observable.

4.3 Design of PD controller

A Proportional-Derivative (PD) controller is a control loop feedback mechanism used in process control type industrial. A PD controller calculates an "error" value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error by adjusting the process input. PD controller calculation involves two parameters values proportional (P) and derivative (D). Proportional values is determines the reaction to the current error, derivative values determines the reaction rate at which the error has been changing.

Figure 6: PD controller



Gain has matrix form and implements PD component. w_1, w_2 are the $\dot{\theta_1}$ and $\dot{\theta_2}$ respectively. I assumed that desired closed poles are [-11, -10, -10, -10, -10, -10]. We can obtain parameter of PD controller by Ackermann's formula. [1][2] By calculations coefficients are

$$K = \begin{bmatrix} 0.7052 & 0.4167 & -1.5266 & -0.4232 & -3.8166 & -0.3251 \end{bmatrix}$$

But in gain we should use $-K$ because we are finding error.

Worth to note that this values work only near upper bound of stability ($\theta_1 = \theta_2 = 0°$). In different places different coefficients, i.e. in the lowest bound of stability ($\theta_1 = \theta_2 = 180°$) gain coefficient equals

$$K = \begin{bmatrix} 0.8180 & 0.1800 & 0.9666 & -0.0770 & -0.0289 & -0.0235 \end{bmatrix}$$

In this work I have 3 different coefficient for 3 different states.

4.4 Swing up, swing down

Swing up is the process moving the pendulum from lower bound stability point to the upright position. This problem can not be linearized during the whole process. Swing up can be done by applying the right sequence of impulse to the cart. Same problem for swing down algorithms. There are no explicit (adequate)solutions.

Because of that I implemented two types of swing up sequences and one for swing down. First moves from lowest bound of stability to the upright position. Second moves from lowest bound of stability to $\theta_1 = 0°, \quad \theta_2 = 180°$. And the swing down gets down from upright to $\theta_1 = -180°, \quad \theta_2 = -180°$.

4.5 Simulation and results

Simulation for initial values of $x = 7, \quad \theta_1 = 20°, \quad \theta_2 = 0°$. Worth to note that $q1 = \theta_2 - \theta_1$ and measures relative angle.

Figure 7: Upright balancing

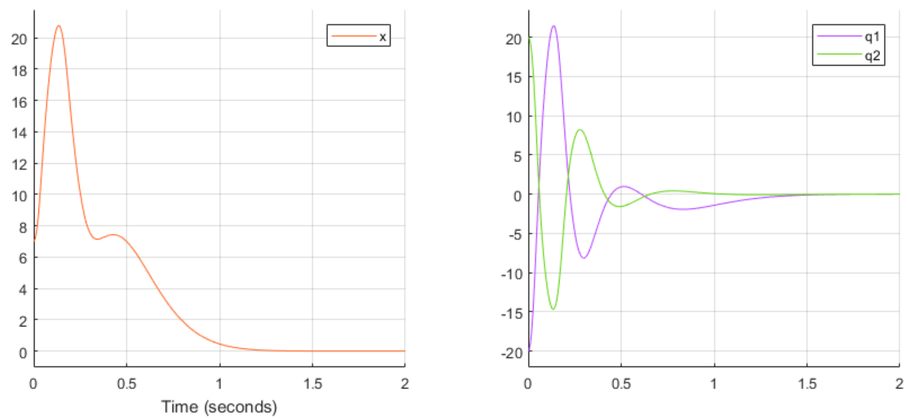
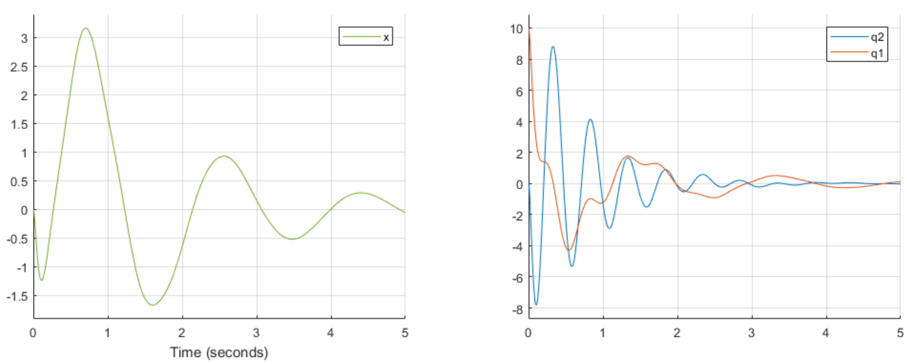


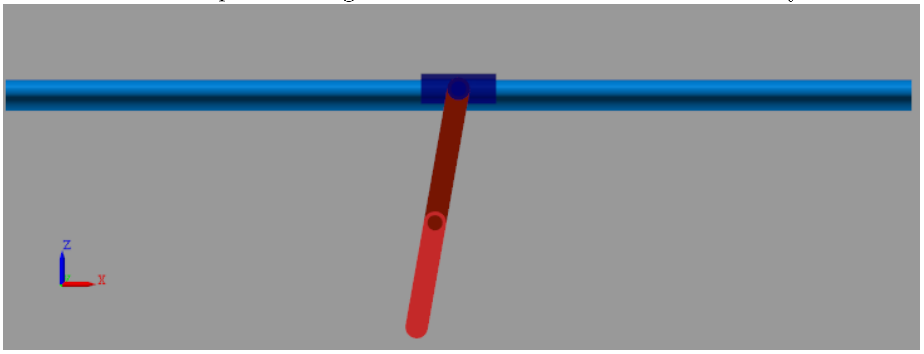
Figure 8: Upside down balancing



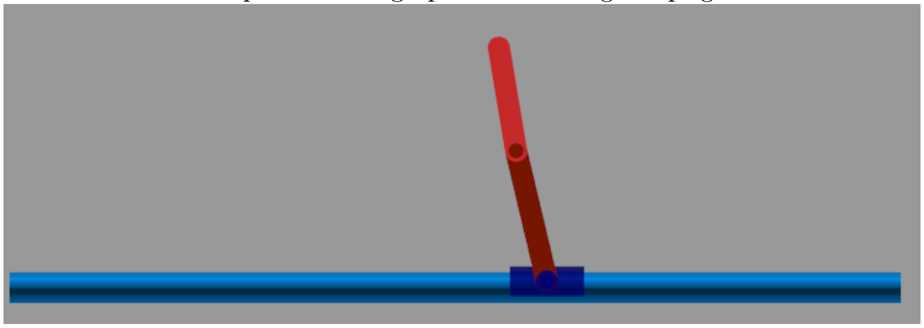
4.6 Sequence of actions

Also I have implemented sequence of actions to prove algorithms.

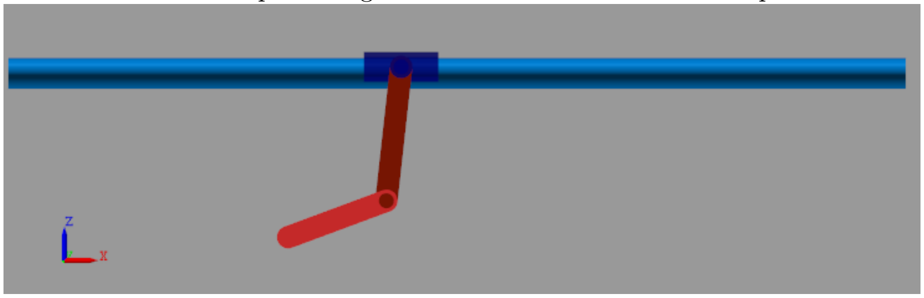
First step is the alignment to the lowest bound of stability



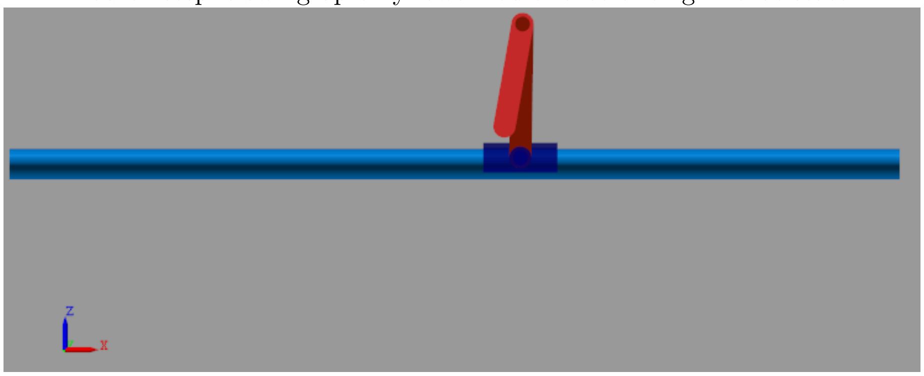
Second step is the swing up and balancing in upright state



Third step is swing down to the state of the first step



Fourth step is swing up only lower rod and balancing in that state



5 Conclusion and Discussion

This results and code can be found in GitHub page [4]. Also, there are plenty of thing which can be improved. The first thing is I want to do is to implement LQR[6] controller which seems more stable than PD. Also the problem of swing up/down actions can be solved by advanced techniques which are working with nonlinear systems. One of the minor improvements can be the way to do the sequence of actions (now it works through several switches and timers). Improvements can be added to the algorithms for swing down (should be smoother).

References

[1] Ackermann's formula. "`https://en.wikipedia.org/wiki/Ackermann%27s_formula`".

[2] Feedback control. "`http://web.mit.edu/16.31/www/Fall06/1631_topic13.pdf`".

[3] Inverted pendulum. "`https://en.wikipedia.org/wiki/Inverted_pendulum`".

[4] Inverted pendulum on a cart. "`https://github.com/prokop7/Control-Theory-Coursework`".

[5] A. Bogdanov. Optimal control of a double inverted pendulum on a cart. *Oregon Health and Science University, Tech. Rep. CSE-04-006, OGI School of Science and Engineering, Beaverton, OR*, 2004.

[6] L. Moysis. Balancing a double inverted pendulum using optimal control and laguerre functions. *Aristotle University of Thessaloniki, Greece*, 2016.

[7] N. Singh and S. K. Yadav. Comparison of lqr and pd controller for stabilizing double inverted pendulum system. *International Journal of Engineering Research and Development*, 1:69–74, July 2012.