

**Middle East Technical University**

**Department of Electrical & Electronics Engineering**

EE 498 Special Topics: Control System Design and Simulation

Term Project Report

Mehmet KILIÇ

2232262

Table of Contents

[Introduction 3](#_Toc76656620)

[System Model 3](#_Toc76656621)

[Controller Design Method Discussion 8](#_Toc76656622)

[LQR Design 9](#_Toc76656623)

# Introduction

# System Model

In this section, mathematical model of the inverted pendulum on a cart system is derived. A simplified version of the system is given below. In this system, a pendulum having the mass is placed on a cart and its position (i.e., the angle ϴ) is controlled by using an actuator which drives the cart. The system is unstable without control. This means that the pendulum will fall if not controlled. When the position of the pendulum starts to deviate from the reference (let us say the point ϴ=π), actuator moves the cart so that the pendulum keeps its position. In the figure below, the force F shows the resulting force that actuator applies to the cart. In this derivation, actuator side is omitted, and system dynamic equations are derived in terms of the force F.

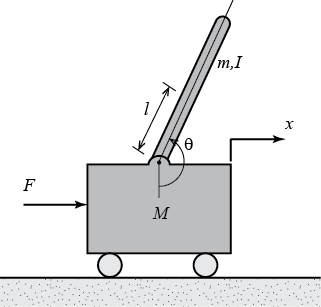


Figure 1. Inverted Pendulum on a Cart

M: Mass of the cart

m: Mass of the pendulum

b: Friction coefficient

l: Distance between the cart and the pendulum center of mass

I: Moment of inertia of the pendulum

F: Force applied to the cart by actuator

x: Cart’s position

ϴ: Counterclockwise angle between pendulum and the vertical axis.

Now, let us draw the free body diagram and write the force equations of the system.

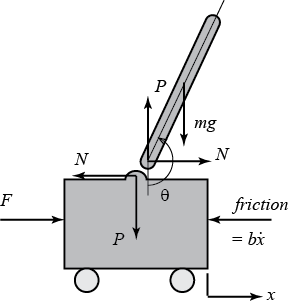


Figure 2. Free body diagram of the system

Where P and N shows the action-reaction forces that the pendulum and cart apply to each other. First, the force equation applied to the cart in horizontal direction is written.

$$ M\ddot{x}+b\dot{x}+N = F $$ (1)

Second, forces acting on the pendulum in the horizontal direction are summed up.

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

Finally, N in the equation (2) is put into the equation (1) and following equation is obtained.

 (3)

After obtaining this equation, forces acting on the pendulum in the perpendicular direction of the pendulum are summed up.

$$P\sin\theta+N\cos\theta-mg\sin\theta=ml\ddot{\theta}+m\ddot{x}\cos\theta$$ (4)

Also, moment equation around the center of the pendulum is written.

$$-Pl\sin\theta-Nl\cos\theta=I\ddot{\theta}$$ (5)

When the equations (4) and (5) are combined, following equation is obtained.

$$(I+ml^2)\ddot{\theta}+mgl\sin\theta=-ml\ddot{x}\cos\theta $$ (6)

Now we have the equations (3) and (6), which are required to obtain the continuous time state space model of the system. Note that both equations include sine and cosine terms, so they are nonlinear. To analyze the system and design a controller for the system, the equations (3) and (6) should be linearized. To do that, the angle that we want to keep the pendulum must be determined. Let us define that the reference angle ϴ = π. This means that we want to control the pendulum angle such that it does not deviate from this point much. So, we can linearize the trigonometric functions around this point. If we define the deviation angle ϕ, we can approximate the pendulum angle ϴ as ϴ = π ± ϕ.

$$ \cos \theta = \cos(\pi + \phi) \approx -1 $$ (7)

$$ \sin \theta = \sin(\pi + \phi) \approx -\phi $$ (8)

$$ \dot{\theta}^2 =  \dot{\phi}^2 \approx 0 $$ (9)

By substituting the equations (7), (8) and (9), final linearized system dynamics equations are obtained. Note that the letter u is used to represent the force instead of the F because it represents the input.

$$ (I+ml^2)\ddot{\phi}-mgl\phi=ml\ddot{x} $$ (10)

$$ (M+m)\ddot{x}+b\dot{x}-ml\ddot{\phi}=u $$ (11)

Finally, we are ready to obtain the state-space model. To completely control the system, state vector is chosen as follows.

By using this state vector, both carts and pendulums position can be controlled. Otherwise, we could control the pendulum angle with the cart’s position diverging to the infinity or vice versa. However, we can control both carts and pendulum’s position now. Also, the output vector y is determined as follows.

Finally, continuous time state space model of the system is obtained by using equations (10) and (11).

$$
\left[{\begin{array}{c}
  \dot{x}\\ \ddot{x}\\ \dot{\phi}\\ \ddot{\phi}
\end{array}}\right] =
\left[{\begin{array}{cccc}
  0&1&0&0\\
  0&\frac{-(I+ml^2)b}{I(M+m)+Mml^2}&\frac{m^2gl^2}{I(M+m)+Mml^2}&0\\
  0&0&0&1\\
  0&\frac{-mlb}{I(M+m)+Mml^2}&\frac{mgl(M+m)}{I(M+m)+Mml^2}&0
\end{array}}\right]
\left[{\begin{array}{c}
  x\\ \dot{x}\\ \phi\\ \dot{\phi}
\end{array}}\right]+
\left[{\begin{array}{c}0\\
  \frac{I+ml^2}{I(M+m)+Mml^2}\\
  0 \\
  \frac{ml}{I(M+m)+Mml^2}
\end{array}}\right]u$$

$${\bf y} =
\left[{\begin{array}{cccc}
  1&0&0&0\\0&0&1&0
\end{array}}\right]
\left[{\begin{array}{c}
  x\\ \dot{x}\\ \phi\\ \dot{\phi}
\end{array}}\right]+
\left[{\begin{array}{c}
  0\\0
\end{array}}\right]u$$

Figure 3. Continuous Time State Space Model

After having the continuous time state-space model, discrete time model is obtained by using MATLAB. Before continuing to the controller design, plant properties are observed. First, stability of the plant is checked by calculating the eigenvalues of matrix A in the discrete time state-space model. As shown in the figure 4, there are two eigenvalues that outside of the unit circle, which makes the system asymptotically instable. This makes sense because if no control is implemented, pendulum would fall. This means that if the system is started without control, output will diverge. In other words, zero input solution does not converge to zero for large times.

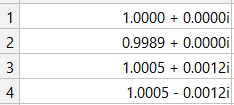


Figure 4. Eigenvalues of matrix A

Then, ranks of the controllability and observability matrices are calculated. Both have the rank=4, Which means that the plant is completely controllable and observable. Transfer functions are also obtained. Since two outputs are defined in the system definitions, two transfer functions can be obtained for both outputs. These transfer functions are obtained in MATLAB.

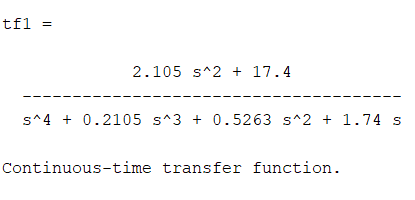


Figure 5. Transfer Function with respect to output x(t)

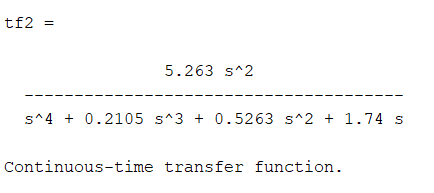


Figure 6. Transfer function with respect to output theta(t)

Stability of both transfer functions are checked by using the *isstable()* function of the MATLAB and it turns out that both transfer functions are instable as expected. Having full information about the system model, now we can discuss the suitable controller design methods.

# Controller Design Method Discussion

By the end of the last section, DTSS model of the system is obtained. System is asymptotically instable, controllable, and observable. Also, has no plant integrators for both transfer functions. Although the complexity of the system and having multiple state variables call for the methods like LQR and MPC, suitability of other methods is also discussed.

This system is too complicated to use a simple design method like bode plot design. Another possible method is the symmetric optimum. Having a stable plant is a prerequisite for symmetric optimum method. It is obvious that this method is not suitable for this system. Another possible method is the pole placement. Plant structure is suitable for using pole placement method. Therefore, it is suitable for such a system. However, since we have multiple state variables that we want to control, the most suitable design methods are the LQR and MPC. No input or state constraints are defined in the system definition. Therefore, LQR design seems safe.

# LQR Design

In this part, first, a finite horizon LQR algorithm is developed with different cost matrices. Their differences are compared, and relevant matrices are determined by considering design specifications and control aim. Then, an infinite horizon solution is implemented with the same cost matrices and results are compared. After solving the problem in MATLAB using linearized system model and ignoring the non-idealities such as saturation and disturbances, both linear and non-linear models are implemented in Simulink and different cases are simulated. Effect of the non-idealities, accuracy of the linearization and success of the designed LQR is discussed.

Before starting to the finite horizon LQR design, first the priorities of the control design are considered. Note that different cost matrices lead us to different optimal solutions. Therefore, weights of the input, intermediate states and final states should be adjusted such that the controller would behave suitably to the control aim. Note that LQR may become vulnerable in case of saturation limits on the input or states. Let us assume that our main consideration is this: magnitude of the input signal is more important from the settling time or the overshoot. This verbal definition is a good point to start. We may start choosing cost matrices as shown below and tune them to enhance the performance of the controller.