# Performance Comparison Between LQR, PD, MPC for Inverted Pendulum with Actuator Dynamics & Constraints

Seung Yun (Leo), Song
*Mechanical Science and Engineering*
*University of Illinois, Urbana-Champaign*
Urbana IL, USA
ssong47@illinois.edu

Joao Ramos
*Mechanical Science and Engineering*
*University of Illinois, Urbana-Champaign*
Urbana IL, USA
jlramos@illinois.edu

***Abstract*—Recently, a major interest has emerged for controlling underactuated mechanical systems, specifically inverted pendulum on a cart system, in many practical applications. However, studies used different and unrealistic models of the system, so the comparison between different controllers from different studies was difficult. Thus, the goal of this study was to design and analyze the performances of three controllers (i.e., LQR, PD, MPC) to achieve the control objective on a virtual inverted pendulum on cart system with actuator dynamics and constraints. The dynamics equations of motion were determined using Euler-Langrangian equations and linearized at the point of equilibrium to generate a linear time invariant state space model. Tracking performances (e.g., settling time, rise time), energy usage (e.g., cost of transportation), and computational time were calculated for each controller. The results indicated that MPC had the best tracking performance (e.g., shortest settling time of 11.9s and 0.1s for cart position and pendulum) and energy usage (e.g., 80% less cost of transportation for cart position) but longest computational time (e.g., 4.12 s), while LQR and PD controller displayed similar tracking performances and energy usage but shortest computational time. The LQR design process was the most intuitive due to its ability to directly tune the most relevant control parameters such as the weights on tracking performance (Q) and control effort (R).**

***Keywords—cart-pole, inverted pendulum, LQR, PID, MPC***

## I. INTRODUCTION

In the past couple decades, the control and analysis of fully actuated robot manipulators have been studied extensively. Many control methods based on feedback linearization, Lyapunov theory, learning, etc. were created for the fully actuated cases (i.e., systems with equal number of actuators as degrees of freedom) [1].

Recently, a major interest has emerged for stabilizing algorithms in underactuated mechanical systems (i.e., number of independent actuators < degrees of freedom) due to the need for control of such systems in many practical applications [2]–[4]. The interest comes from the need to stabilize systems such as ships, helicopters, aircraft, satellites, and walking robots, which may be underactuated by design or become underactuated due to actuator failure. The control techniques used in fully actuated robot manipulators do not apply directly to underactuated non-linear mechanical systems.

The inverted pendulum system is used as a valuable benchmark for control engineers to demonstrate effectiveness of different control approaches since the system imposes many challenges for the engineer [2]–[5]: 1) the system is highly non-linear and unstable open-loop system with only one equilibrium point (i.e., pole perpendicular to the ground in upward position), and 2) an underactuated system. The control issue requires not only stabilization of the pendulum in the upward equilibrium position but also the position of the cart, increasing the complexity of control design. For controlling nonlinear underactuated mechanical systems, many control designs were developed starting from simple controllers such as proportional-derivative (PD) controllers and linear quadratic regulator (LQR) method to more advanced controllers such as Lyapunov direct method, sliding mode control, and model predictive controllers [6]–[9].

However, these studies' systems were not realistic due to lack of consideration of the actuator dynamics and constraints (e.g., maximum allowable input) [6]–[9]. Thus, controllers with unrealistic and extreme magnitudes of gain were used to satisfy the control design requirements. To select and analyze a practical controller, these realistic actuator constraints and dynamics must be included. Also, comparison between various controllers developed by different studies was difficult since the system setup and dynamics were different for each study. For example, some papers assumed to neglect the inertia of the pendulum [5] and friction of the pendulum while others incorporated them into the system dynamics [9]. Therefore, a common system setup needs to be used to analyze and compare different controllers.

Thus, there are two objectives of this study: 1) to achieve desired control objectives after implementing realistic actuator dynamics and constraints, and 2) to implement and compare three common control methods (LQR, PD, MPC) in terms of performance, computation time, ease of development. Also, basic design guidelines for tuning parameters for these three controllers were discussed. We virtually created an inverted pendulum on a cart system using MATLAB and Simulink. The dynamic equations of motion governing the behavior of mechanical systems with holonomic constraints were determined from the Euler-Lagrangian equations. Then, the system was linearized at equilibrium point to obtain a state space model representation of the system. The control objective was to move the cart to a desired position while stabilizing the mechanism around the unstable equilibrium point. The tracking performance (i.e., settling time, rise time, overshoot, steady state error), and energy usage (i.e., average power consumption, cost of transportation) and computation time were computed.

## II. METHODS

### A. Modeling of Inverted Pendulum on Cart System

The system consisted of a pendulum pivotable on a cart that moved horizontally and actuated by a horizontal force, generated by a motor-belt-pulley system (Fig. 1). The system parameters were given in the Appendix. The assumptions for the system were that 1) pendulum pole had zero mass and inertia, 2) all friction including cart and ground friction, transmission friction, and air resistance were ignored, 3) the system starts from the equilibrium position, and 3) driven and drive pulley have the same radius.

To solve for the equations of motions, the nonlinear model of the system was obtained through Lagrangian equation (1). The Lagrangian equation requires computation of the total kinetic (K) and potential (V) energy in the system (2 – 5).

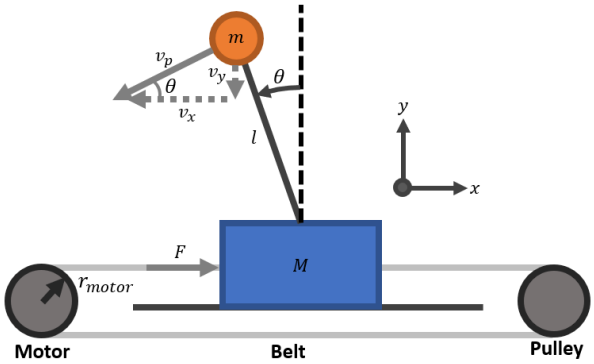


Fig. 1. Inverted pendulum on a cart driven by motor-belt-pulley system.

$$L = K - V \tag{1}$$

$$K = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m(v_x^2 + v_y^2) \tag{2}$$

$$v_x = \dot{x} - \dot{\theta}l\cos\theta \quad v_y = \dot{\theta}l\sin\theta \tag{3}$$

$$V = mgl\cos\theta \tag{4}$$

$$L = \frac{1}{2}(M+m)\dot{x}^2 + \frac{1}{2}ml\left(l\dot{\theta}^2 - 2\dot{\theta}\dot{x}\cos\theta - g\cos\theta\right) \tag{5}$$

Using the Euler-Langrangian formulation (6), the two equations of motion, one for each degree of freedom (i.e., horizontal movement, rotation about pendulum pivot), can be derived (7, 8).

$$F_i = \frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \left(\frac{\partial L}{\partial q_i}\right) \tag{6}$$

$$F = \frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{x}}\right) - \left(\frac{\partial L}{\partial x}\right) = (M+m)\ddot{x} - ml\left(\ddot{\theta}\cos\theta - \dot{\theta}^2\sin\theta\right) \tag{7}$$

$$0 = \frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \left(\frac{\partial L}{\partial \theta}\right) = l\ddot{\theta} - \ddot{x}\cos\theta - g\sin\theta \tag{8}$$

The Euler-Langrangian formulation can be organized into a generic form (9) in order to obtain a non-linear differential equation (10).

$$\begin{bmatrix} F \\ 0 \end{bmatrix} = (D(q) + J_{rotor})\ddot{q} + C(q,\dot{q}) + G(q) + B_{damp}(\dot{q}) = B_e u, \tag{9}$$

$$\ddot{q} = \left(D^{-1}(q) + J_{rotor}\right)\left(B_e u - C(q,\dot{q}) - G(q) - B_{damp}\dot{q}\right) \tag{10}$$

where, $q = \begin{bmatrix} x \\ \theta \end{bmatrix}$.

To implement the three control techniques, the nonlinear equation was linearized at the equilibrium point using Taylor expansion. The linearized model was organized into a state space model for easier computation (11-14). See Appendix for $A, B, C, D$.

$$\mathbf{x} = \begin{bmatrix} x\ \theta\ \dot{x}\ \dot{\theta} \end{bmatrix}^T \quad \mathbf{u} = \text{motor voltage} \tag{11}$$

$$\mathbf{x}^* = [0\ 0\ 0\ 0]^T \quad \mathbf{u}^* = 0 \tag{12}$$

$$\dot{\mathbf{x}} \approx f(\mathbf{x}^*, \mathbf{u}^*) + \frac{\partial f}{\partial \mathbf{x}}\Big|_{\substack{\mathbf{x}=\mathbf{x}^* \\ \mathbf{u}=\mathbf{u}^*}} (\mathbf{x} - \mathbf{x}^*) + \frac{\partial f}{\partial \mathbf{u}}\Big|_{\substack{\mathbf{x}=\mathbf{x}^* \\ \mathbf{u}=\mathbf{u}^*}} (\mathbf{u} - \mathbf{u}^*) \tag{13}$$

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \tag{14}$$

$$\mathbf{y} = C\mathbf{x} + D\mathbf{u}$$

To ensure that the mode's linearized system can be controlled and observable, we verified that the ranks of the controllability matrix ($C(A,B)$) was equivalent to the number of states (= 4). Since the ranks of $C(A,B)$ were equal to the number of states, the system was controllable, allowing implementation of controllers. To understand the performances of all types of controllers, we chose three different controllers: two classical commonly used controllers in industry (LQR, PD) for linear time invariant systems (LTI) and one modern and nonlinear controller (MPC) for nonlinear time variant system.

### B. Linear Quadratic Regulator (LQR)

Linear Quadratic Regulator (LQR) is a full state feedback controller that optimally finds the gain $K_{LQR}$ given the weights on performance (i.e., tracking of the states) and control effort (i.e., magnitude of the actuator input) as well as system dynamics (Fig. 2) [10]. The weights on performance and control effort were represented in the form of a square and semi-positive definite matrix Q ($n \times n$), and square and positive definite matrix R ($m \times m$), respectively, where $n, m$ were number of states and inputs, respectively. The magnitude of each diagonal element in Q and R represented the amount of weight for achieving the minimizing the tracking error for Q, the control effort for R. For the study, we used the following values of Q and R (15).

$$Q = \begin{bmatrix} 1000 & 0 & 0 & 0 \\ 0 & 4000 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad R = 1 \tag{15}$$

The internal mechanism behind LQR involved finding a set of solutions by solving cost function that used the values of Q and R while satisfying the system dynamics (16) [10].

$$(\rho) \quad \min J = \int_0^\infty (\mathbf{x}^\mathbf{T} Q\mathbf{x} + \mathbf{u}^\mathbf{T} R\mathbf{u})\, dt\,, \quad \mathbf{u} \in \mathbb{R}^\mathbf{m} \tag{16}$$
$$s.t. \qquad \dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$$
$$\mathbf{y} = C\mathbf{x} + D\mathbf{u}$$

A set of solutions (S) to the cost function can be solved using the algebraic Riccati Equation (17). Each solution can represent a set of gains ($K_{LQR}$) (18) [10].

$$A^T S + SA - SBR^{-1}B^T S + Q = 0 \quad (17)$$

$$K = R^{-1}B^T S \quad (18)$$

where, $S$ is the solution to (23). Since multiple $S$ can exist, multiple set of $K_{LQR}$ can exist. However, only one set of $K_{LQR}$ can yield the system to be stable, which can easily be done by checking the eigen values ($\lambda$) of the characteristic equation ($A - BK$). If $\lambda$ are all negative, then such set of $K_{LQR}$ can make the system stable. Note that, since LQR was a state feedback controller, a pre-compensator ($\bar{N}$) was necessary to compute what the steady-state value of the states should be. Thus, $\bar{N}r$ became our new reference to track. In our study, we used the following gain and pre-compensation (19,20).

$$K_{LQR} = [-31.6228 \quad 113.3629 \quad -240.1784 \quad 33.5693] \quad (19)$$

$$\bar{N} = -31.6228 \quad (20)$$

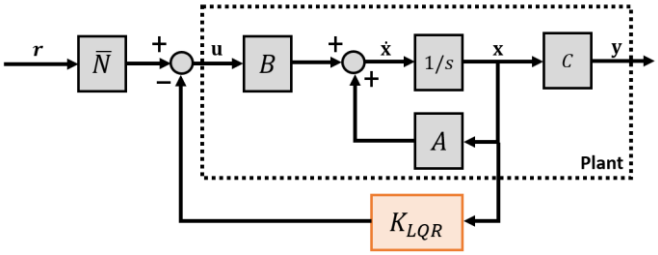


Fig. 2. Control diagram for a generic LQR implementation on a linear, time invariant (LTI) system.

The benefits of LQR were that the easiness of tuning the controller since the tuning parameters were directly related to intuitive and physical parameters of the system (e.g., performance and control effort) rather than relying on arbitrary pole placement, which was not as intuitive. Also, the controller enables the weighting of individual states on more important states such as cart position and pendulum angle than cart speed or pendulum angular speed. However, one potential drawback of LQR is the limitation on linear models and does not update the gains online. So, when nonlinearities or disturbances are present, the LQR may not perform well.

### C. Proportional-Derivative (PD) Controller

Proportional-Derivative (PD) controller is a classical feedback controller widely used in the industry [9]. The controller continuously calculates the error ($\mathbf{e} = \mathbf{x} - \mathbf{r}$) and rate of error ($\dot{\mathbf{e}} = \dot{\mathbf{x}} - \dot{\mathbf{r}}$) between the desired reference point and measured output (Fig. 3.). The input is simply the sum of the product of the gain ($K_p$) and $\mathbf{e}$ and the product of derivative gain ($K_d$) and $\dot{\mathbf{e}}$ (21) [11].

$$\mathbf{u} = K_p\mathbf{e} + K_d\dot{\mathbf{e}} \quad (21)$$

For controlling multiple reference points and output, multiple PD controllers can be used to control for each output. In our study, two PD controllers were used with the following gains, and the input values from the two controllers were summed (22, 23) [9].

$$K_{p1}, K_{d1} = -50, -400 \qquad K_{p2}, K_{d2} = 150, 100 \quad (22)$$

$$\mathbf{u} = K_{p1}\mathbf{e_x} + K_{d1}\dot{\mathbf{e}}_\mathbf{x} + K_{p2}\mathbf{e_\theta} + K_{d2}\dot{\mathbf{e}}_\boldsymbol{\theta} \quad (23)$$

The pros of PD controller were the ease of development and setup. Also, the computational cost is extremely low, especially with modern computers. The cons of PD controller are that they are difficult to tune when multiple PD controllers are involved. If an input of one controller is coupled to multiple outputs (which happens to be the case in this study), the tuning process becomes more tedious. Also, PD controllers do not provide optimal control since the gains are predefined, making the overall performance is reactive and a compromise. Finally, PD controllers give poor performance in the presence of nonlinearity and process changes.

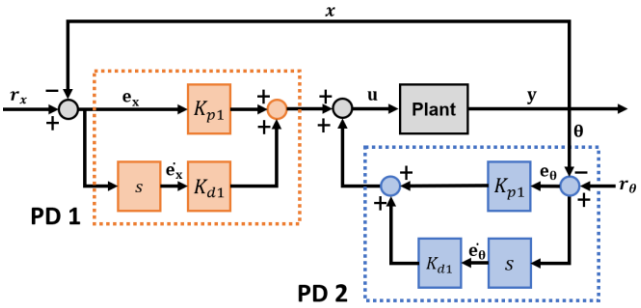


Fig. 3. Control diagram implementing two PD controllers on an LTI system.

### D. Model Predictive Controller (MPC)

Model Predictive Controller (MPC) is a relatively modern feedback controller that optimally finds the actuator input by running multiple predictions of the plant model for various actuator values for a short period of time in the future (prediction horizon) (Fig. 4.)[12]. The controller strategically computes the actuator input values for a brief period (control horizon) using an optimizer by minimizing cost function while satisfying the input and output constraints (24). The control input of the next immediate time stamp is used, and this cycle of prediction and optimization is repeated as the horizon shifts to the next time stamp.

$$\begin{aligned} (\rho) \quad & \min J = |\mathbf{y_t} - \mathbf{y}| \\ s.t. \quad & \dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \\ & \mathbf{y} = C\mathbf{x} + D\mathbf{u} \\ & \mathbf{u_{low}} < \mathbf{u} < \mathbf{u_{high}} \\ & \mathbf{y_{t_{low}}} < \mathbf{y_t} < \mathbf{y_{t_{high}}} \end{aligned} \quad (24)$$

where, $\mathbf{y_t}$ and $\mathbf{y}$ were predicted and measured output, respectively, and $\mathbf{u_{low}}, \mathbf{u_{high}}, \mathbf{y_{t_{low}}}, \mathbf{y_{t_{high}}}$ were lower bound of input, upper bound of input, lower bound of predicted output, and higher bound of predicted output, respectively.

The tunable parameters for MPC are sampled time ($T_s$), length of prediction ($P_h$) and control horizon ($C_h$), weight on inputs and outputs ($W$), and constraints on input and output ($\mathbf{u_{low}}, \mathbf{u_{high}}, \mathbf{y_{t_{low}}}, \mathbf{y_{t_{high}}}$). A $T_s$ too small would increase the computational cost excessively, while $T_s$ too large would

make the controller difficult to handle disturbances. A $P_h, C_h$too small would generate a solution that is not optimal or too slow to react, while $P_h, C_h$ too large would increase the computational cost. In this study, the following parameters were used for MPC (25 – 27).

$$T_s = 0.001s \quad P_h = 4000 \quad C_h = 10 \tag{25}$$

$$\mathbf{u_{low}} = -36V \quad \mathbf{u_{high}} = 36V \tag{26}$$

$$\mathbf{y}_{\mathbf{t}_{\mathbf{low}}} = \begin{bmatrix}\mathbf{x}\\ \boldsymbol{\theta}\end{bmatrix} = \begin{bmatrix}-\infty\\ -0.05\,rad\end{bmatrix} \quad \mathbf{y}_{\mathbf{t}_{\mathbf{high}}} = \begin{bmatrix}\infty\\ 0.05\,rad\end{bmatrix} \tag{27}$$

The input was set as hard constraint (i.e., constraint that must be satisfied), while outputs were set as soft constraints (i.e., constraints that MPC attempts to satisfy but would allow minimal violation of the constraints for convergence). The single input ($\mathbf{u}$) and two outputs ($\mathbf{x}, \boldsymbol{\theta}$) were equally weighed.

The advantages of MPC are that control designers can explicitly impose constraints on inputs and outputs (e.g., motor input voltage between -12V to 12V). Also, the controller works for non-linear systems. In addition, MPC can handle Multi-Input-Multi-Output (MIMO) systems with input and output coupling, which is especially useful for our case. The main disadvantage of MPC is the expensive computational cost, since high end processor and large memory is required for successfully predicting the future model response and computing the optimal solution for each time stamp.

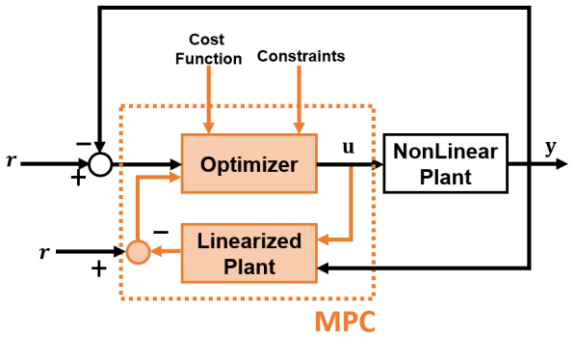


Fig. 4. Control diagram implementing MPC on a nonlinear system.

### E. Experimental Setup and Data Analysis

MATLAB™ (Mathworks, Natick MA, USA) and Simulink™ (Mathworks, Natick MA, USA) were used to simulate and implement the system, since many control toolboxes and numerical solvers such as LQR and MPC control toolbox and `ode45` were available for efficient tuning process and solving the system dynamics. Three experiment trials were run: each trial using one of the tuned controllers (LQR, PD, MPC) on a system with same parameters (Table 1.), initial conditions ($x, \theta, \dot{x}, \dot{\theta}, \mathbf{u} = 0$), references ($r_x = 0.5\text{m}, r_\theta = 0$), and simulation time ($\text{t}_{\text{final}}$) of 25 seconds. The control objectives were shown in Table 1.

Actuation input for all controllers were saturated at $\pm$ 12V. However, since a short burst of overvoltage is typically allowed and observed in real life system, we allowed the actuation input to exceed $\pm$12V for a maximum of 1.5 seconds. If the actuation input exceeded $\pm$12V for longer than 1.5 seconds or reached absolute maximum voltage ($\pm$36V), the actuation input was saturated at $\pm$12V. We used an existing commercial brushed DC motor with planetary gearbox to reference motor specification (5202-0002-0027, goBilda Winfield, KS USA).

For each trial, tracking performance (settling time ($t_{settle}$), rise time ($t_{rise}$), overshoot ($OS$), steady-state error ($e_{ss}$),), energy usage (average power consumption ($\bar{P}$), and cost of transportation ( $COT$ )), and computational time ($t_{comp}$) were analyzed. The definitions for these performance parameters were given in Appendix. The computational time was computed using `tic toc` function in MATLAB.

TABLE I. CONTROL OBJECTIVES

| | $t_{settle}$ $(s)$ | $t_{rise}$ $(s)$ | $OS$ | $e_{ss}$ |
|---|---|---|---|---|
| $x$ (m) | 16 | 10 | 0.1 | 0.02 |
| $\theta$ (rad) | 10 | - | 0.1 | 0.05 |

## III. RESULTS AND DISCUSSIONS

### A. Tracking Performance and Computation Time

The state ($x, \theta, \dot{x}, \dot{\theta}$) responses and control effort ($\mathbf{u}$) of the inverted pendulum on cart system across time were plotted for each controller (Fig. 5 – 7) to visually investigate the tracking performance and control effort. All controllers met the control objective with input saturation. Also, all controllers met the control objectives. Low steady state errors were observed for all states from all controllers. Finally, the computation time were all below five seconds. While the controllers were able to maneuver the cart position and stabilize the pendulum, many performance and computational time differences for each controller were observed.

The MPC exhibited best tracking performance and energy efficiency, but longest computation time. All tracking performance metrics as well as cost of transportation for MPC were significantly lower than other controllers for all states. In comparison to the other two controllers, MPC displayed a more rapid change in the control input, resulting in faster tracking performances for all states (Fig. 7). The settling time and rise time for all states except $\dot{x}$ was almost twice faster than LQR and PD. The steady state error was almost zero for all states. Also, the energy efficiency of MPC was far superior than the other controllers due to low average power consumption and fast settling time. The drawback of MPC was the long computation time (almost 640 times of LQR and 230 times of PD controller). The long computation time was expected since MPC required higher computational resources to perform prediction of the system model and optimization of the actuation input for every loop.

The PD controller and LQR had minor performance differences in terms of tracking, energy usage, and computational time. In terms of tracking, the LQR method exhibited lower $\text{t}_{\text{settle}}$ for all states, especially $\theta$ (almost four times faster).$t_{rise}, OS, e_{ss}$ for LQR and PD controller were similar for all states. However, in terms of energy usage, the COTs of all states, except for $\theta$, for PD controller were almost two times lower than LQR, indicating that PD controller demonstrated higher efficiency than LQR method. While both LQR and PD controller demonstrated a smoother state tracking for all states and control effort compared to MPC, the PD controller exhibited smoother state tracking than LQR since $\dot{x}, \dot{\theta}$ were smaller in magnitude compared to LQR (Fig. 5 – 7). Thus, the smoother state transitions and control effort

of the PD controller decreased the amount of power consumed from the motor, reducing the cost of transportation. In addition, the LQR and PD controller moved the cart in negative $x$ direction first to allow the pendulum to swing towards the positive $x$ direction. Then, the cart was moved to positive $x$ direction to reach the desired cart position (Fig. 5, 6). Thus, the LQR and PD controller's $\theta$ changed in terms of direction from negative to positive, which was the opposite case of MPC. Finally, the times for computation for LQR and PD controller were significantly lower than MPC: 1.7%, and 4.9% of MPC's, respectively. This was expected since LQR and PD controller do not perform any prediction and online optimizations for each time stamp.

### *B. Controller Design Guidelines*

The design and tuning process was most intuitive for LQR method. This was due to LQR's ability to directly tune the most relevant and intuitive control parameters such as the weights on tracking performance and control effort. Also, when tuning LQR, the designer need not worry about the coupling effect between inputs and outputs. Unlike LQR, tuning the PD controller took a much longer time since changing the gains (e.g., $K_{d2}$) for tracking one output (e.g., $\theta$) would affect the response of another output (e.g., $x$), complicating the tuning process. While the MPC allowed control of key control parameters such as input constraints, there were too many control parameters that were dependent to each other. For example, changing the sampled time would require the designer to change the prediction horizon, control horizon, etc. Also, the weights on inputs and outputs were dependent on soft and hard constraints, so change in one weight required changes on the soft and hard constraints. Without the control toolbox offered by MATLAB, tuning the MPC would have taken a long time.

It is difficult to conclude which controller performed best, since all the controllers have pros and cons. For controlling physical systems with online control, fast computing time is necessary, especially if computational resources are limited. Thus, LQR may be more favorable than MPC, sacrificing tracking performance for practical implementation. However, if the control was done offline and tracking performance was critical, MPC may be a better choice.

### *C. Limitations and Future Work*

There were some limitations of the study due to simplification of the system. The study used a linearized model of the system to allow use of control techniques such as LQR. However, the true dynamics of the system involved a highly nonlinear system, so deviation of $\theta$ outside of $\pm 20°$ would render the linearized plant inaccurate. Also, certain physics such as friction were ignored. These effects may be nonlinear, rendering the system dynamics to be more complex.

Thus, future work may include using a nonlinear model of the system taking realistic physical parameters such as friction, inertia of pendulum, disturbances in the input, noise in the output into account. This would require use of more advanced and nonlinear controller such as adaptive MPC and Lyapunov direct method as well as state estimators such as Kalman Filter [6]. Also, the analysis of robustness of the system using disk margins can be done.

TABLE II. PERFORMANCE OF LQR, PD, MPC

| | | $t_{settle}$ $(s)$ | $t_{rise}$ $(s)$ | $OS$ | $e_{ss}$ | $\bar{P}$ $(W)$ | $COT$ | $t_{comp}$ $(s)$ |
|---|---|---|---|---|---|---|---|---|
| **LQR** | $x$ | 15.3 | 10.6 | 0.02 | 0.02 | 18.0 | 80.0 | 0.12 |
| | $\theta$ | 3.8 | - | 0.04 | 0.005 | | 20.1 | |
| | $\dot{x}$ | 0.2 | - | 0.05 | -0.001 | | 1.04 | |
| | $\dot{\theta}$ | 1.4 | - | 0.06 | -0.002 | | 7.4 | |
| **PD** | $x$ | 15.4 | 10.6 | 0.008 | 0.008 | 7.05 | 32.6 | 0.33 |
| | $\theta$ | 11.1 | - | 0.05 | 0.004 | | 22.9 | |
| | $\dot{x}$ | 0.1 | - | 0.05 | 0.00 | | 0.5 | |
| | $\dot{\theta}$ | 3.2 | - | 0.05 | -0.001 | | 6.5 | |
| **MPC** | $x$ | 7.2 | 4.8 | 0.0 | 0.00 | 14.4 | 30.3 | 76.9 |
| | $\theta$ | 0.2 | - | 0.005 | 0.00 | | 0.7 | |
| | $\dot{x}$ | 3.6 | - | 0.2 | 0.00 | | 15.2 | |
| | $\dot{\theta}$ | 0.47 | - | 0.4 | 0.00 | | 2.0 | |

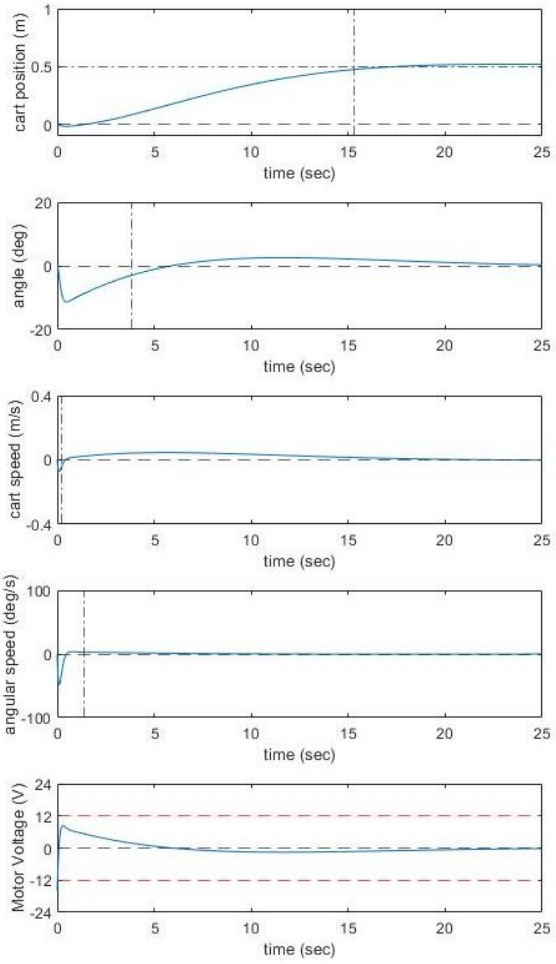


Fig. 5. Response and actuator input of inverted pendulum on cart system using LQR controller. The vertical dashed line indicates the $t_{settle}$ for top four plots. The red horizontal dashed line represents the input saturation limit of $\pm 12$V.

## IV. CONCLUSIONS

In this study, three controllers (LQR, PD, MPC) were designed to satisfy the control objectives of a virtual inverted pendulum on cart system that incorporated realistic actuator dynamics and constraints. The equations of motions were determined by the Euler-Langrangian equations and linearized at the equilibrium point to generate a linear time invariant state space model of the system. Tracking performance of the states, actuator efficiency, and computation time were analyzed. MPC exhibited the best

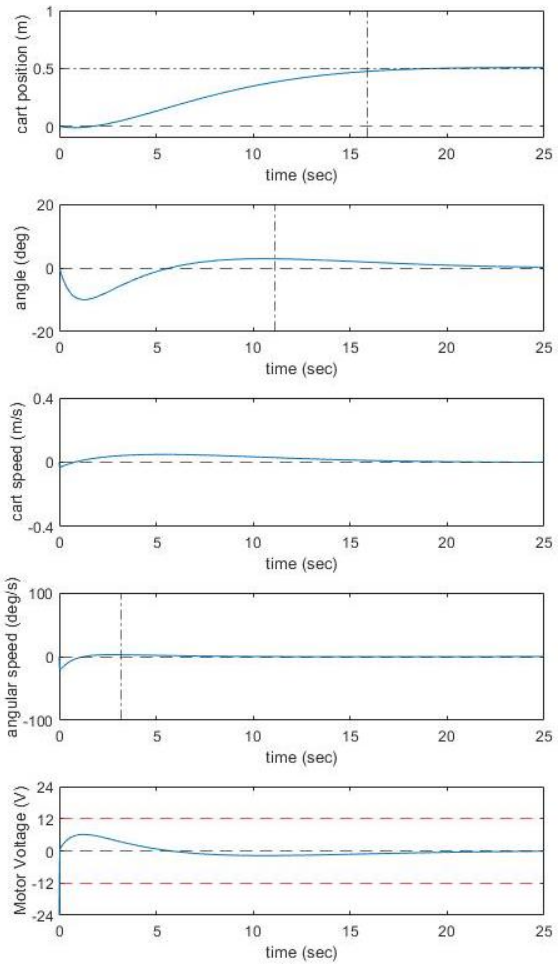


Fig. 6. Response and actuator input of inverted pendulum on cart system using PD controller. The vertical dashed line indicates the $t_{settle}$ for top four plots. The red horizontal dashed line represents the input saturation limit of $\pm$12V.

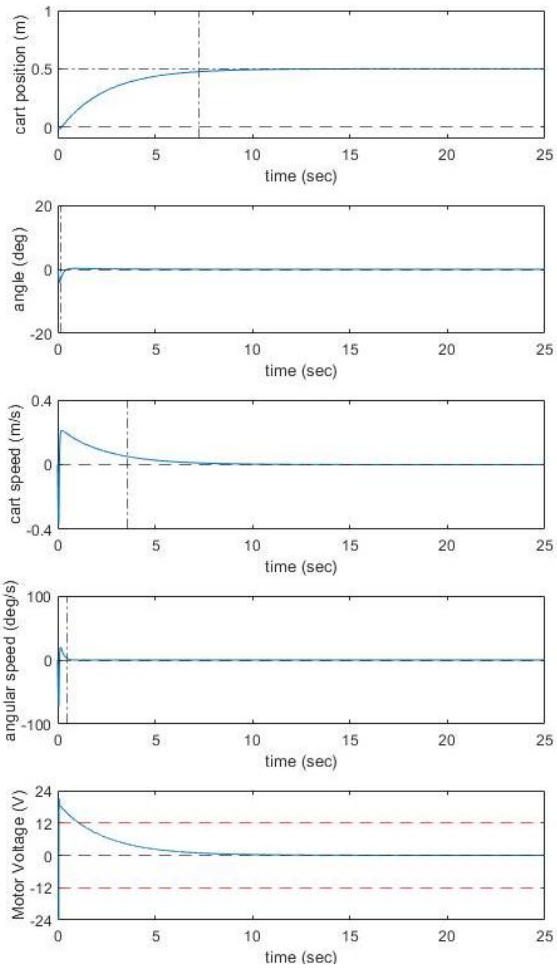


Fig. 7. Response and actuator input of inverted pendulum on cart system using MPC. The vertical dashed line indicates the $t_{settle}$ for top four plots. The red horizontal dashed line represents the input saturation limit of $\pm$12V.

tracking performance and actuator efficiency but longest computation time, while LQR and PD controller displayed faster computation times.The design of LQR was most intuitive due to direct tuning of relevant control parameters such as performance tracking and control effort. Future work may include using nonlinear and realistic model, nonlinear controllers, and state estimators.

## ACKNOWLEDGEMENT

I would like to thank all my ECE team members (Shaoyu Meng, Ningyuan Du, Samir Wadhwa, Deonna Flanagan) as well as the instructors for ECE 489 class (Professor Ramos, Dan Block, Yanran Ding, and Mengchao Zhang) for their help.

## REFERENCES

[1] R. M. Murray, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 2017.

[2] M. W. Spong, "Underactuated mechanical systems," in *Control Problems in Robotics and Automation*, 1998, pp. 135–150.

[3] Fantoni I , Author and, Lozano R , Author, and Sinha SC , Reviewer, "Non-linear Control for Underactuated Mechanical Systems," *Appl. Mech. Rev.*, vol. 55, no. 4, pp. B67–B68, Jul. 2002, doi: 10.1115/1.1483350.

[4] R. Ortega, M. W. Spong, F. Gomez-Estern, and G. Blankenstein, "Stabilization of a class of underactuated mechanical systems via interconnection and damping assignment," *IEEE Trans. Automat. Contr.*, vol. 47, no. 8, pp. 1218–1233, Aug. 2002, doi: 10.1109/TAC.2002.800770.

[5] H. Yu, Y. Liu, and T. Yang, "Closed-loop tracking control of a pendulum-driven cart-pole underactuated system," *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.*, vol. 222, no. 2, pp. 109–125, 2008, doi: 10.1243/09596518JSCE460.

[6] W. N. White, M. Foss, and X. Guo, "A direct Lyapunov approach for a class of underactuated mechanical systems," in *2006 American Control Conference*, 2006, pp. 8-pp.

[7] M. Nikkhah, H. Ashrafiuon, and K. R. Muske, "Optimal sliding mode control for underactuated systems," in *2006 American Control Conference*, 2006, p. 6 pp., doi: 10.1109/ACC.2006.1657461.

[8] Y. Han*, "A robust algorithm for model-following control of under-actuated systems, and its application to a non-holonomic robot and an aircraft," *Int. J. Syst. Sci.*, vol. 36, no. 6, pp. 341–356, 2005.

[9] A. N. K. Nasir, M. A. Ahmad, and M. F. Rahmat, "Performance comparison between LQR and PID controllers for an inverted pendulum system," in *AIP Conference Proceedings*, 2008, vol. 1052, no. 1, pp. 124–128.

[10] R. E. Kalman, "Contributions to the theory of optimal control."

[11] N. Minorsky., "DIRECTIONAL STABILITY OF AUTOMATICALLY STEERED BODIES," *J. Am. Soc. Nav. Eng.*, vol. 34, no. 2, pp. 280–309, May 1922, doi: 10.1111/j.1559-3584.1922.tb04958.x.

[12] C. R. Cutler and B. L. Ramaker, "Dynamic matrix control - A computer control algorithm," in *joint automatic control conference*, 1980, no. 17, p. 72.

## APPENDIX

TABLE III. SYSTEM PARAMETERS

| Parameter | Units | Value |
|---|---|---|
| $M$ | Mass of cart | 0.5 $kg$ |
| $m$ | Mass of pole-end | 0.3 $kg$ |
| $l$ | Length of pole | 0.3 $m$ |
| $K_v$ | Motor speed constant | 0.0186 $\frac{rad}{s \cdot V}$ |
| $K_t$ | Motor torque constant | 0.0135 $\frac{Nm}{A}$ |
| $I_r$ | Rotor inertia | 0.000007 $kgm^2$ |
| $N_r$ | Gear ratio | 26.9 |
| $R_w$ | Winding resistance | 1.3 Ω |
| $r_m$ | Radius of motor shaft | 0.006 $m$ |

Equations (28–31) explain the matrices for Euler-Lagrangian Formulation. $D, C, G, J_{rotor}, B_{damp}, B_e$ were the mass-inertia, Coriolis and centripetal, gravitational, rotor-inertia, damping, and actuator matrices, respectively.

$$D(q) = \begin{bmatrix} M+m & -mlcos\theta \\ -cos\theta & l \end{bmatrix} \tag{28}$$

$$C(q,\dot{q}) = \begin{bmatrix} ml\dot{\theta}^2 sin\theta \\ 0 \end{bmatrix} \quad G(q) = \begin{bmatrix} 0 \\ -gsin\theta \end{bmatrix} \tag{29}$$

$$J_{rotor} = \begin{bmatrix} \frac{I_r N_r^2}{r_m^2} & 0 \\ 0 & 0 \end{bmatrix} \quad B_{damp} = \begin{bmatrix} \frac{K_t K_v N_r^2}{R_w r_m^2} & 0 \\ 0 & 0 \end{bmatrix} \tag{30}$$

$$B_e = \begin{bmatrix} \frac{K_t N_r}{R_w r_m} \\ 0 \end{bmatrix} \qquad q = \begin{bmatrix} x \\ \theta \end{bmatrix} \tag{31}$$

Equations (32 – 40) explain the linearized time invariant state space matrices of the system.

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & A_{32} & A_{33} & 0 \\ 0 & A_{42} & A_{43} & 0 \end{bmatrix} \tag{32}$$

$$A_{32} = \frac{R_w gm - R_w lm}{R_w(I_r m N_r^2 r_r^2 + M)} \tag{33}$$

$$A_{33} = -\frac{K_t K_v N_r^2 m}{R_w(I_r m N_r^2 r_r^2 + M)} \tag{34}$$

$$A_{42} = \frac{I_r R_w lm N_r^2 r_r^2 + M R_w g + R_w gm - R_w lm}{R_w l(I_r m\, N_r^2 r_r^2 + M)} \tag{35}$$

$$A_{43} = -\frac{K_t K_v N_r^2 m + K_t K_v M N_r^2}{R_w l(I_r m N_r^2 r_r^2 + M)} \tag{36}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ B_{31} \\ B_{41} \end{bmatrix} \tag{37}$$

$$B_{31} = \frac{K_t N_r m}{R_w(I_r m N_r^2 r_r^2 + M)} \tag{38}$$

$$B_{41} = \frac{K_t M N_r + K_t N_r m}{R_w l(I_r m N_r^2 r_r^2 + M)} \tag{39}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{40}$$

Equations (41 – 48) explain the definition of tracking performance (41 – 45) and energy usage (46 – 48).

$$\mathbf{y_{final}}\text{: steady-state response } = \begin{bmatrix} 0.5 \text{ m} \\ 0 \text{ rad} \end{bmatrix} \tag{41}$$

$$t_{settle}\text{: time for error } |\mathbf{y} - \mathbf{y_{final}}| \text{to fall within 5\%} \tag{42}$$

$$t_{rise}\text{: time it takes for } \mathbf{y} \text{ from 10\% to 90\% of } \mathbf{y_{final}} \tag{43}$$

$$OS\text{: } \max(\mathbf{y}) - \mathbf{y_{final}} \tag{44}$$

$$e_{ss}\text{: } \mathbf{y}(\mathrm{t_{final}}) - \mathbf{y_{final}} \tag{45}$$

$$\bar{P} = \frac{\sum_{t=0}^{t_{final}} \mathbf{u}(t) * \mathbf{i}(t)}{t_{final}} \tag{46}$$

$$\mathbf{i}(t) = \frac{u(t) - K_v \omega(t)}{R_w}, \qquad \omega(t) = \frac{\dot{x}(t)}{r_m} \tag{47}$$

$$COT = \frac{\bar{P}}{g(M+m)(\frac{r_{\mathbf{x}}}{t_{settle}})}, \quad r_{\mathbf{x}} = [r_x, r_\theta, r_{\dot{x}}, r_{\dot{\theta}}] \tag{48}$$