Controller design for stabilization of an inverted pendulum on cart against disturbances

Deepak Ramesh, Kriish Hate

Background and Motivation :

Carts and pendulum systems have been widely used in engineering applications ranging from industrial automation to robotics. The cart mechanism, often coupled with a pendulum, serves as a fundamental model in understanding and developing control systems. In industrial settings, carts are used for material transportation, autonomously guided cars, and load balancing systems. Their simple yet versatile design allows them to adapt to a range of tasks in various environments (shop floor, warehouse etc..) In robotics, cart-based systems are foundational in creating balance-driven robots, which mimic human locomotion and dynamic stability. These systems are used in humanoid robotics, self balancing scooters and planetary rovers where maintaining balance in dynamic environments is critical.

One of the most well-known cart-based challenges is the inverted pendulum problem. This theory is a simplified model of real-world balancing tasks. For instance, balancing a humanoid robot on uneven terrain, stabilizing a rocket during its ascent, or controlling a robotic arm with high degrees of freedom can all be abstracted to variations of the inverted pendulum. The inverted pendulum on a cart, specifically, introduces an additional layer of complexity: it requires active control to stabilize an inherently unstable equilibrium point. Unlike static systems, where stability can be achieved passively, the inverted pendulum demands continuous adjustments in force and torque to counteract gravitational pull and maintain balance.

The challenge becomes even more pronounced when external disturbances are introduced. In real-world scenarios, systems are rarely isolated; they are subjected to various types of noise, such as random impulses, periodic sinusoidal variations, or external forces from the environment. These disturbances exacerbate the difficulty of stabilization by forcing the system to react dynamically to unpredictable changes. For example, a robotic cart carrying fragile goods in a manufacturing line might encounter vibrations from nearby machinery, requiring precise control to avoid tipping. Similarly, a self-balancing robot operating outdoors could face wind gusts subjected by environmental factors or uneven ground that destabilize the system without advanced control measures.

This project focuses on addressing these challenges through robust controller design for the inverted pendulum on a cart. By modeling the system and integrating external disturbances into the control framework, we aim to explore and compare the effectiveness of three control strategies: PID (Proportional-Integral-Derivative) controllers, Linear Quadratic Regulators (LQR), and Model Predictive Control (MPC). Each of these approaches brings unique strengths to the problem. PID controllers, known for their simplicity and ubiquity, provide a baseline for performance under varying conditions. LQR offers an optimal control solution by minimizing a cost function, balancing stability and control effort. MPC, on the other hand, introduces predictive capabilities, enabling the system to anticipate and adapt to disturbances in real-time.

Our project specifically addresses the problem of stabilizing an inverted pendulum under external disturbances, this project not only addresses a classical control challenge but also provides insights into designing robust systems for practical applications. The outcomes of this study could inform the development of more reliable and adaptable control systems in robotics, automation.

Current state of the art and your main contribution:

The inverted pendulum problem has been widely studied as a benchmark for testing and developing control strategies in dynamic and unstable systems. Traditional techniques such as PID (Proportional-Integral-Derivative) controllers have been extensively used due to their simplicity and effectiveness in stabilizing linear systems. However, PID controllers often struggle to handle nonlinearities and external disturbances effectively. Advanced techniques like Linear Quadratic Regulators (LQR) have gained prominence for providing optimal control by minimizing a defined cost function, balancing stability, and control effort. Model Predictive Control (MPC) has emerged as a promising solution in recent years, offering predictive capabilities that allow the system to anticipate future disturbances and adjust dynamically. While these methods have been successful in various applications, challenges remain in designing controllers that are robust enough to handle real-world disturbances such as impulses, noise, and sinusoidal variations.

Our main contribution lies in integrating and evaluating these three distinct control strategies—PID, LQR, and MPC—on the inverted pendulum cart system under a unified framework that incorporates external disturbances. By systematically modeling and simulating disturbances, we aim to provide a comparative analysis of the controllers' performance, highlighting their strengths and limitations. Additionally, we seek to demonstrate how advanced predictive techniques like MPC can outperform traditional methods in dynamic and noisy environments. This work not only advances the understanding of controller design for the inverted pendulum but also lays the groundwork for real-world applications in robotics, exoskeletons, and balance-driven systems, bridging the gap between theoretical control strategies and practical implementations.

Modelling

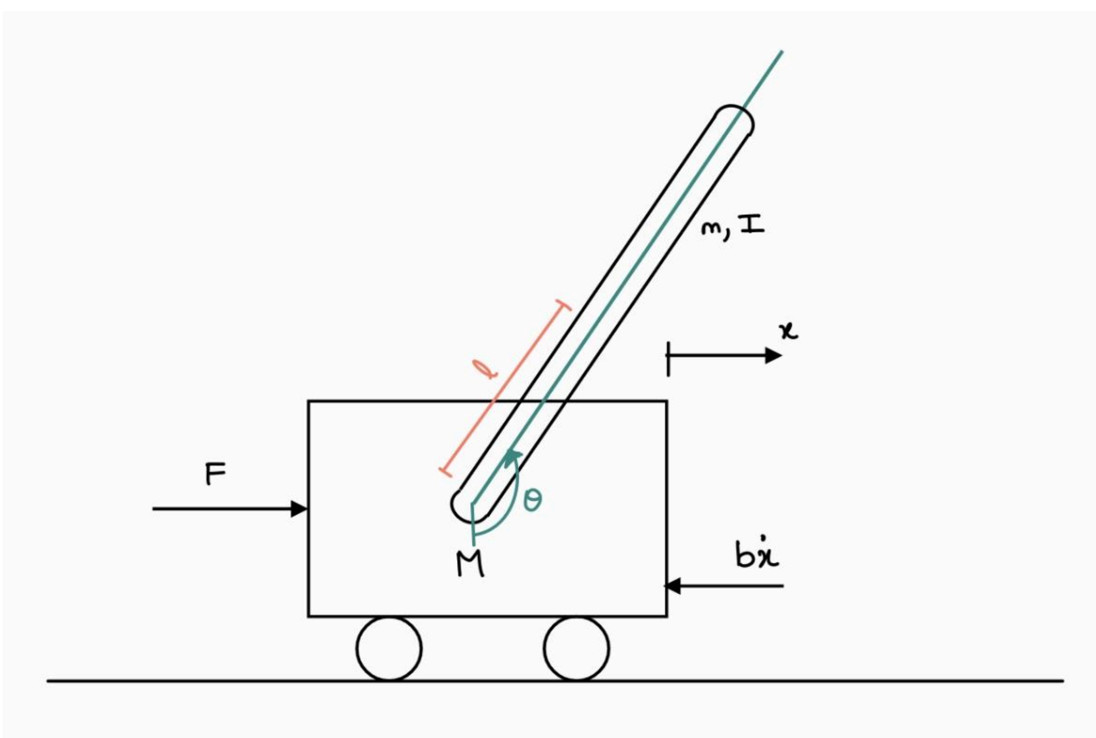

Figure 1: Schematic of an inverted pendulum on a moving cart. The cart has mass (M) and experiences a force (F) with a damping coefficient (b). The pendulum, of mass (m) and length (l), rotates with an angle (theta) from the vertical and has a moment of inertia (I).

(M) - mass of the cart = 8.441 kg

(m) - mass of the pendulum = 0.794 kg

(b) - coefficient of friction for cart = 1 N/m/sec

(l) - length to pendulum center of mass = 0.25 m

(I) - mass moment of inertia of the pendulum = 0.00176 kg.m^2

(F) - force applied to the cart

(x) - cart position coordinate

(θ) - pendulum angle from vertical (down)

The dynamic equations of the system are derived using Newton's equations of motion.

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

$$\left(I+ml^2\right)\ddot{\theta}+mgl\sin(\theta)=-ml\ddot{y}\cos(\theta)$$

Now the set of nonlinear equations need to be linearised around the upward equilibrium $\theta = \pi$.

Let (φ) represent the deviation of the pendulum's position from equilibrium, that is, $\theta = \varphi + \pi$. Again presuming a small deviation (φ) from equilibrium, we can use the following small angle approximations of the nonlinear functions in our system equations:

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

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

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

Now substituting these approximations into the non-linear equations, we obtain a set of two linear equations.

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

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

Transfer function

To obtain the transfer function we take the Laplace transform of the differential equations. As we have two outputs namely, pendulum angle and cart position, we obtain two transfer functions

$$P_{pend}(s) = \frac{\Phi(s)}{U(s)} = \frac{\frac{ml}{q}s}{s^3 + \frac{b(I+ml^2)}{q}s^2 - \frac{(M+m)mgl}{q}s - \frac{bmgl}{q}} \qquad [\frac{rad}{N}]$$

$$P_{cart}(s) = \frac{X(s)}{U(s)} = \frac{\frac{(I+ml^2)s^2 - gml}{q}}{s^4 + \frac{b(I+ml^2)}{q}s^3 - \frac{(M+m)mgl}{q}s^2 - \frac{bmgl}{q}s} \qquad [\frac{m}{N}]$$

State space

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Mml^2} & \frac{m^2gl^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-mlb}{I(M+m)+Mml^2} & \frac{mgl(M+m)}{I(M+m)+Mml^2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ 0 \\ \frac{ml}{I(M+m)+Mml^2} \end{bmatrix} u$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$

Simscape

The CAD model of the system was developed in SOLIDWORKS and the rigid body tree was developed in Simscape. The control input here is a force to move the cart horizontally. The disturbance in the form of an impulse acts on the pendulum. The measured outputs are the states namely cart position, cart velocity, pendulum position and pendulum velocity. The time step of simulation is 0.01s.

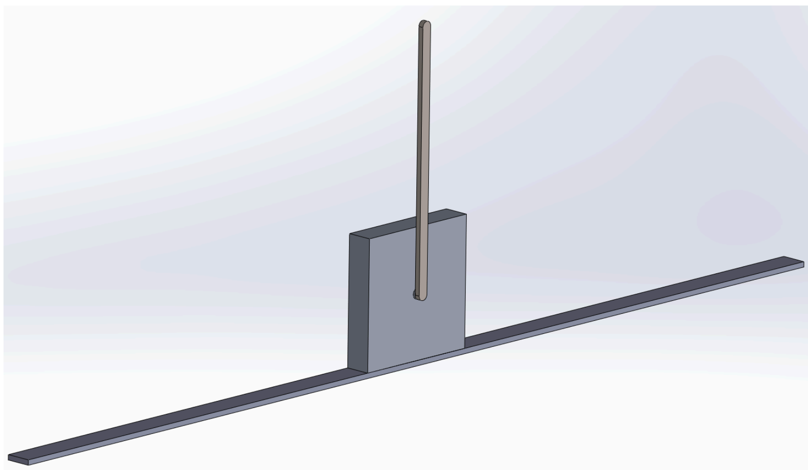

Figure 2 : SOLIDWORKS CAD model of the pendulum cart assembly

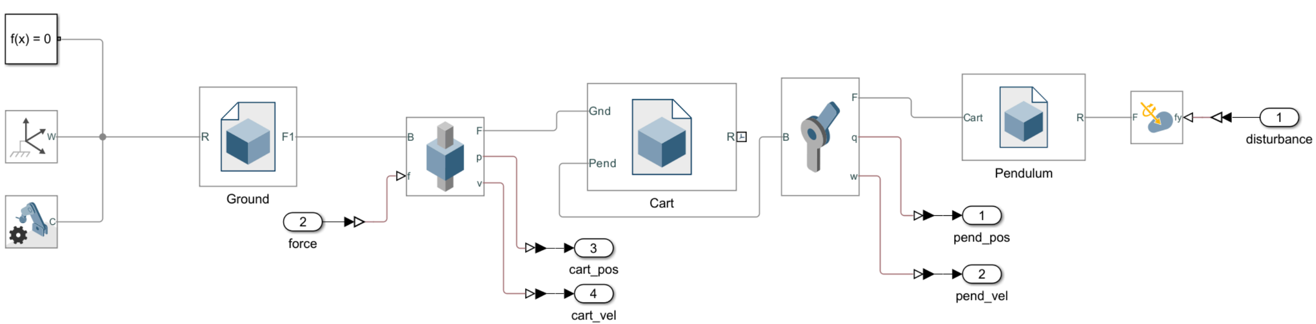

Figure 2: Simulink physical modeling of a pendulum-cart system. The diagram includes components for ground interaction, the cart, and the pendulum, with inputs such as force and disturbances applied to the system. Key outputs include cart position (cart_pos), cart velocity (cart_vel), pendulum position (pend_pos), and pendulum velocity (pend_vel). The disturbance block simulates external forces acting on the system.

Disturbance

The disturbance is assumed to be in the form of an impulse acting on the pendulum. It is assumed to have a sine wave structure with a time period of 50ms and a magnitude of 10 N. The impulse is applied 1s after the start of the simulation.

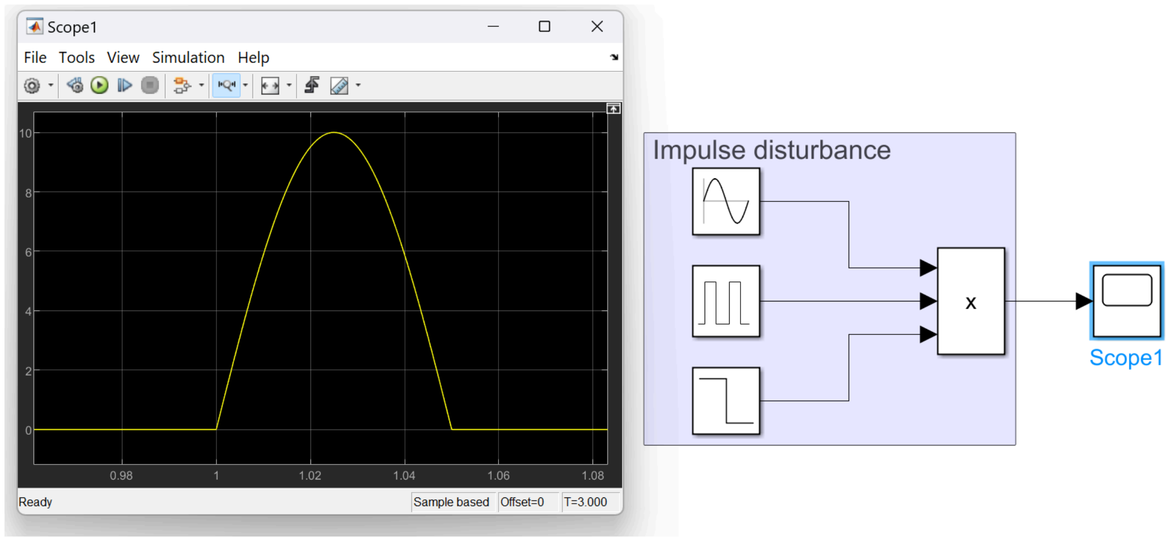

Figure 3: Graphical representation of the impulse being added to the model, The block diagram shows various input signals, such as sine, square, and step waveforms, combined through multiplication, with the resulting output displayed on the Scope1 plot.

Constraints

The pendulum is assumed to have small deviations from the equilibrium of about +/- 20 degrees to maintain the linearised dynamics. The actuator is constrained to have a maximum output of 40 N.

Controller Design

PID Controller

As a PID controller can be used for a SISO system, we need to use two PID controllers in a loop in order to control both the position of the cart and the pendulum. The inner PID loop is used to stabilize the pendulum while the outer loop is used to track the desired reference for the cart position.

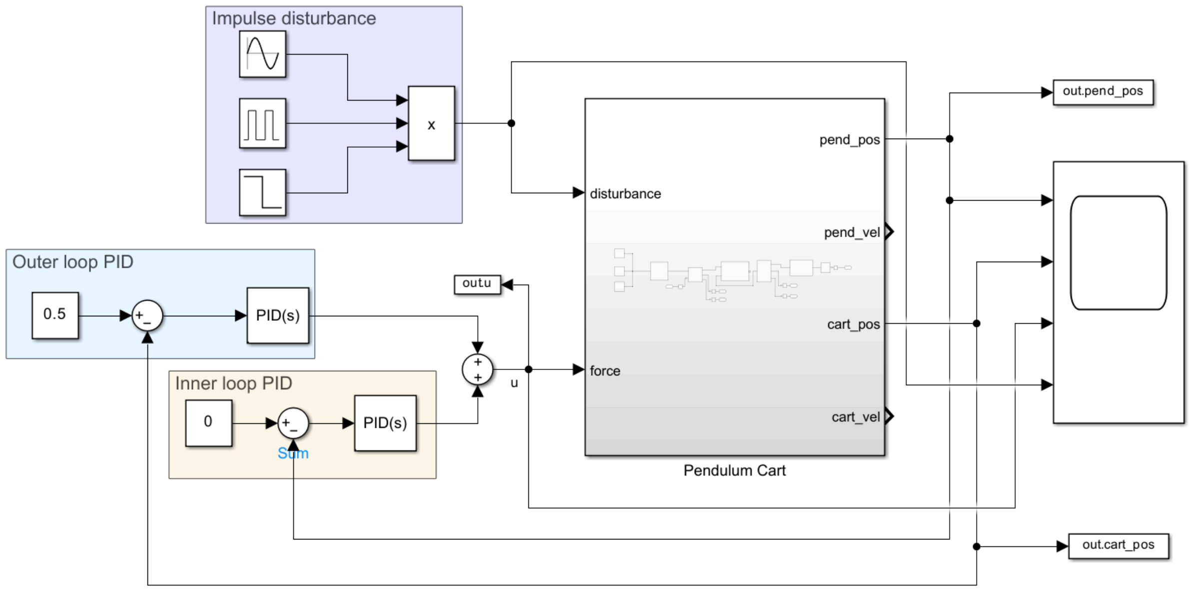

Figure 4: Simulink model of a pendulum cart system with inner and outer loop PID controllers. The diagram incorporates an impulse disturbance block, which introduces external perturbations to the system. The outer loop PID controls the pendulum's position, while the inner loop PID manages the cart's dynamics. Outputs such as pendulum position, velocity, and cart position are visualized in the scope.

The control law of a PID controller is :

$$u(t) = Kpe(t) + Ki\int_0^t e(\tau)d\tau + Kd\frac{de(t)}{dt}$$

Where e(t) = r(t) - y(t) is the error between reference and output y

The controller gains are chosen by a trial and error basis to obtain desired characteristics.

Gains	Open Loop PID	Closed Loop PID
Kp	2.18	289.92
Ki	0.077	0.5
Kd	15.26	22.315

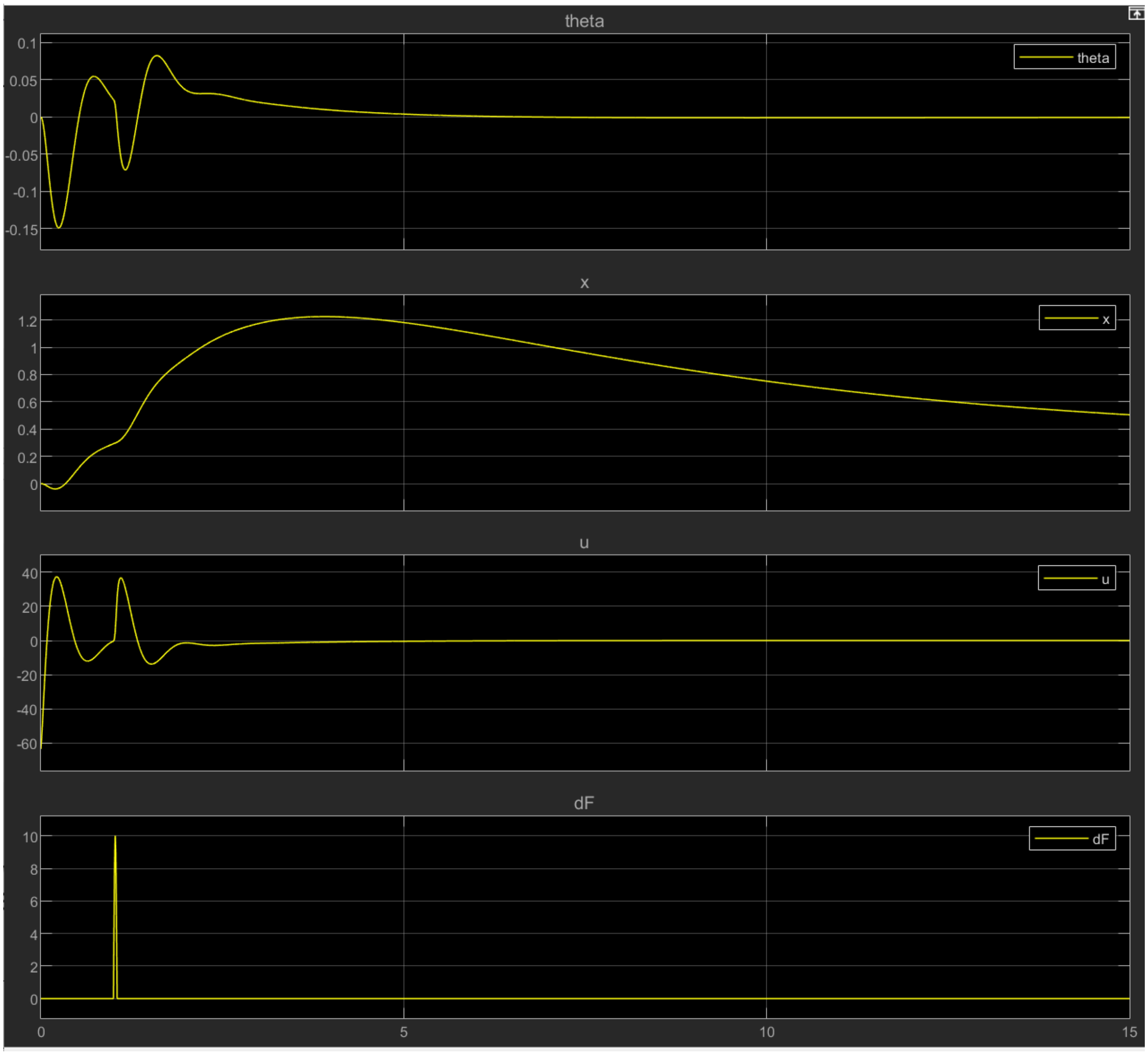

Figure 5: Simulink model of a pendulum-cart system with corresponding output plots. The outputs include theta (θ) (the pendulum angle), x (X) (the cart position), u (U) (the input signal), and dF (dF) (the disturbance). The plots show the system's dynamic response over time, highlighting the behavior of each variable under control and disturbance conditions.

LQR Controller

The LQR controller is a state feedback controller that regulates the system to the equilibrium conditions.

The feedback gain K is calculated by optimizing a quadratic cost function.

$$min\, J = \int_0^{\infty} (X^T Q X + U^T R U) dt$$

Where Q is a (nxn) matrix that defines the weights on tracking the states

R is a (rxr) matrix that defines the weights on the inputs. A set of solutions to the cost function can be solved using the algebraic Riccati Equation.

$$A^T S + SA - SBR^{-1}B^T S + Q = 0$$

$$K = R^{-1}B^T S$$

Or it can be solved using the MATLAB command K = lqr(A,B,Q,R)

The obtained gain matrix K is **K = [-10.0000 -16.7864 228.0862 35.8050]**

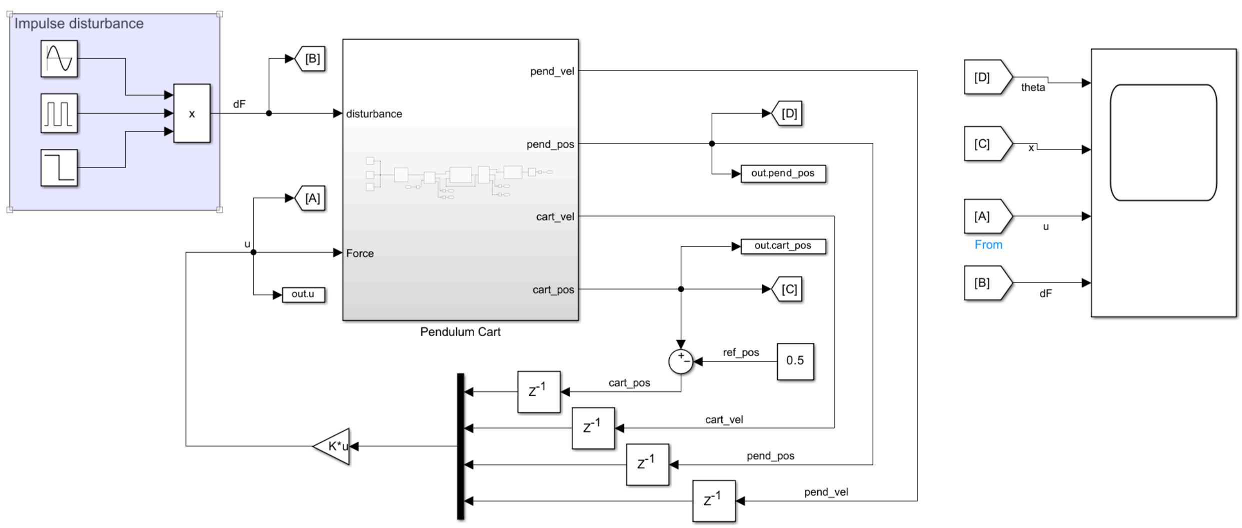

Figure 6: Simulink model of a pendulum cart system incorporating an impulse disturbance block and a feedback control loop. The disturbance is applied to the system, influencing pendulum velocity, position, cart velocity, and cart position. Key outputs such as θ\thetaθ (pendulum angle), x (cart position), u (control force), and dF (disturbance force) are visualized, with feedback loops managing the dynamic response of the system."

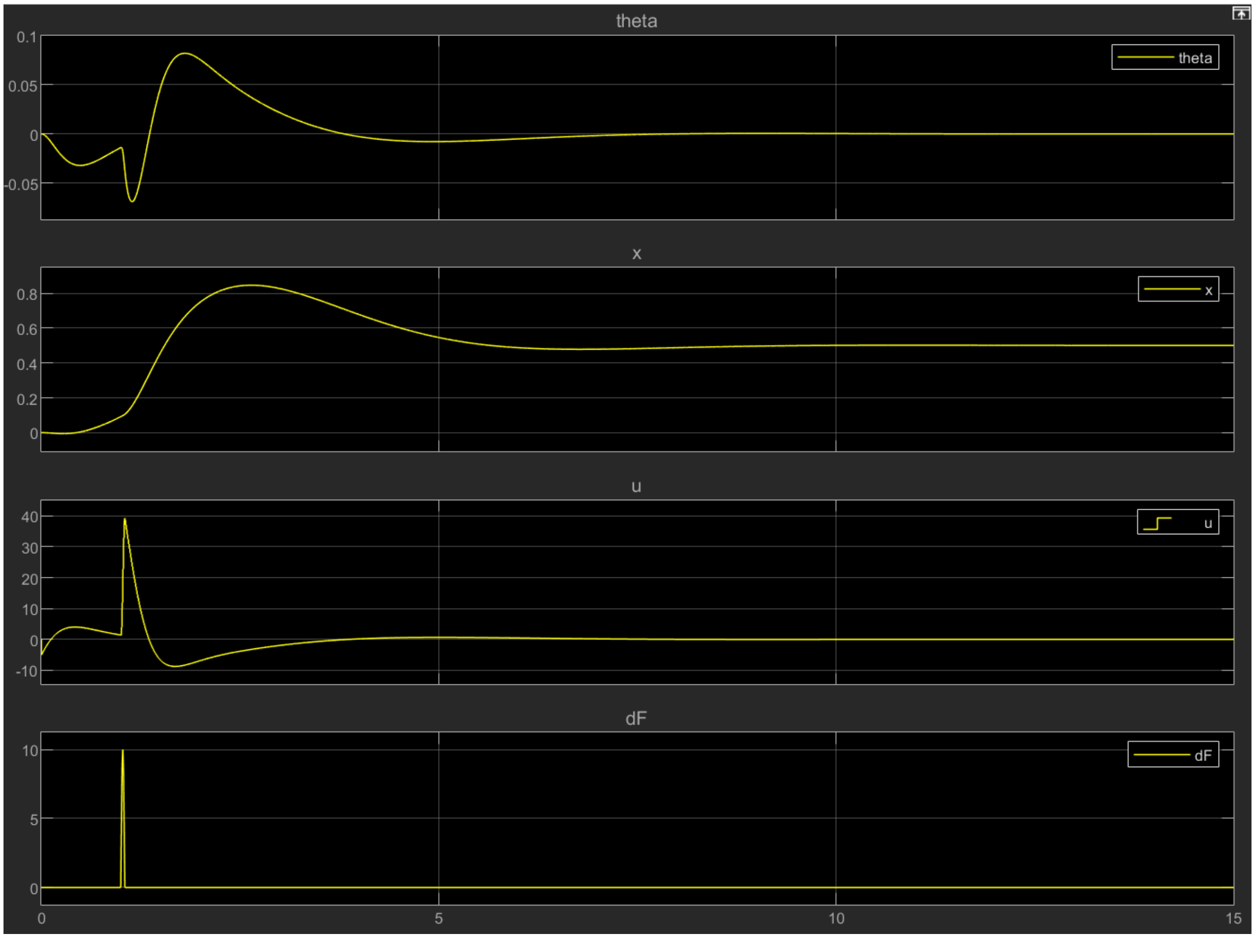

Figure 7: Simulink model of a pendulum-cart system with corresponding output plots. The outputs include theta (the pendulum angle), x (the cart position), u (the input signal), and dF (the disturbance). The plots show the system's dynamic response over time, highlighting the behavior of each variable under control and disturbance conditions.

MPC Controller

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). 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. It comprises a linear plant model, a state estimator and an optimiser to optimise the control input based on the defined cost function. By default a Kalman filter is used as a state estimator.

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

The constraints and weights defined are as follows :

```
%% System definition
pend_cart_ss = ss(A,B,C,D);

pend_cart_ss.InputName = {'u'};
pend_cart_ss.OutputName = {'x'; 'theta'};
pend_cart_ss.StateName = {'x' 'x_dot' 'theta' 'theta_dot'};

pend_cart_ss = setmpcsignals(pend_cart_ss, 'MV',1,'MO',[1 2]);
%% MPC Object creation

mpcobj = mpc(pend_cart_ss,0.01); % Sample time defined

mpcobj.PredictionHorizon = 200;
mpcobj.ControlHorizon = 10;

% Constraint definition
mpcobj.ManipulatedVariables.Min = -40;    % K
mpcobj.ManipulatedVariables.Max = 40;     % K
mpcobj.ManipulatedVariables.RateMin = -1000; % K/step
mpcobj.ManipulatedVariables.RateMax = 1000;  % K/step

% Weights definition
mpcobj.W.ManipulatedVariablesRate = 0.1;
mpcobj.W.OutputVariables = [5 5];

% Noise definition
% mpcobj.Model.Noise = zeros(2, 2); % Zero noise

% get(mpcobj)
% review(mpcobj)

xc = mpcstate(mpcobj);
sim('Simscape_Pend_Cart_MPC.slx');
```

The maximum control input is set to 40N. The weights on the manipulated variable rates determine the rate of change of inputs. The output variable weights are added to eliminate the tracking errors. By default a white noise is assumed on the measured variables by MATLAB.

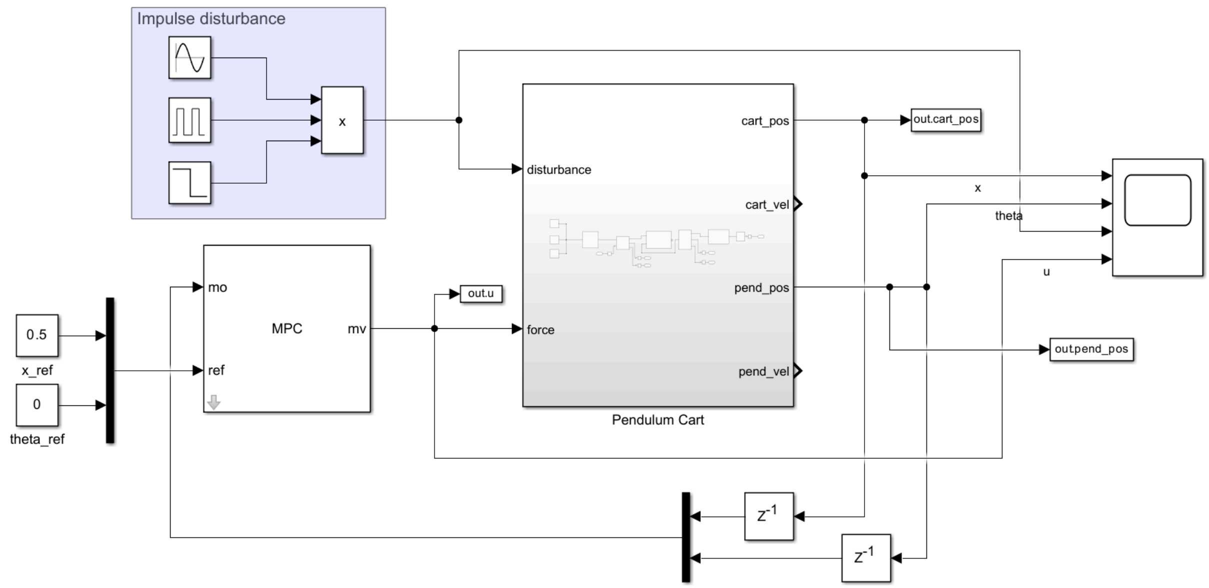

Figure 8 : Simulink model of the MPC controller with reference signals and state feedback

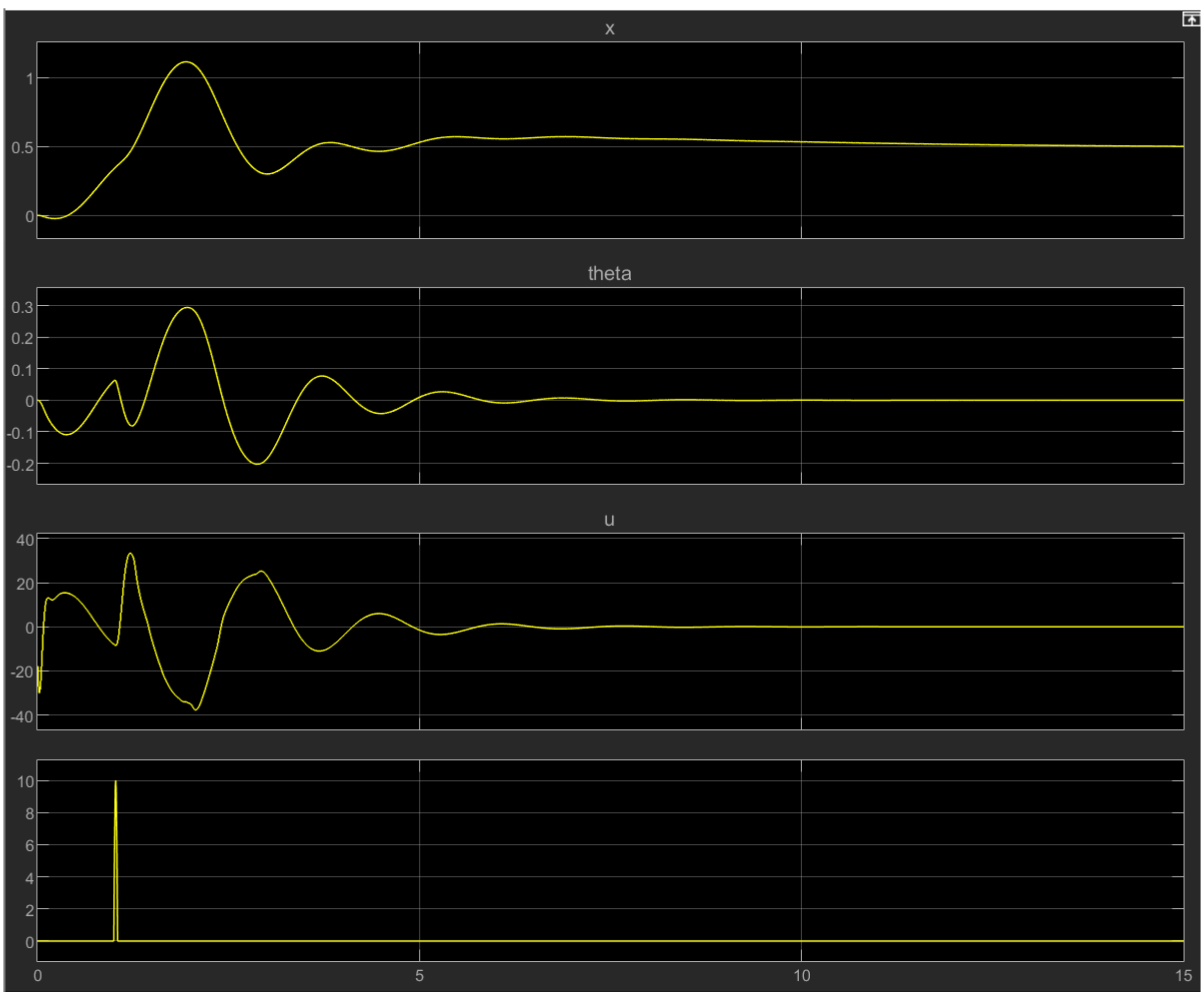

Figure 9: State responses of pendulum angle (theta), cart position (x), control input (u) and impulse disturbance (dF)

Results

Paramters	PID	LQR	MPC
Rise time (s)			
x	0.840	**0.680**	2.930
Settling time (s)			
theta	**14.650**	14.960	14.790
x	14.240	**5.240**	10.720
Percent overshoot (%)			
theta	**14090.13**	811560.00	257660.00
x	143.12	**69.40**	122.41
Max actuator effort (N)	37.40	39.27	**33.35**
Computational time (s)	**2.607**	2.707	3.435

Figure 10: Comparison of control performance metrics for PID, LQR, and MPC controllers. Parameters include rise time (x), settling time (theta and x), percent overshoot (theta and x), maximum actuator effort (N), and computational time (s). Notable values are highlighted, emphasizing differences in performance across control strategies.

PID Controller

Pros : Least computational time; easy to design

Cons : Tuning is time consuming; cannot add constraints on input magnitude, states etc

LQR Controller

Pros : Best performance among the three; relatively less computational time; tuning the weights is very intuitive

Cons : Results in a very high actuator effort

MPC Controller

Pros : Least actuator effort; provision to add constraints on all states; inputs and outputs.

Cons : High computation time; tuning is relatively complex as there are a lot of dependencies on constraints

Conclusion

The simplest controller to design is PID, but it takes a long time to tune, as there are no defined methods to obtain the gains. The trial and error method is time intensive as the max available actuator effort has to be referenced while tuning. LQR has the most intuitive method of tuning, as the weights for performance and input cost can be changed independently. The performance parameters obtained were also the most desirable from LQR. The MPC allows varying the key control parameters individually, but there are a lot of interdependencies on performance outputs and inputs. This makes it relatively complex to tune. Additionally, the computational time is huge in MPC, as it has to evaluate the system over the entire prediction horizon and optimize the control inputs for the entire control horizon for each time step. Overall, if an online controller is preferred, LQR is the most suitable one. If design simplicity is required, PID is the most suitable. And if extreme tracking performance and the least actuator effort are required on an offline computer, then MPC is the most suitable.

a) Major Design Decisions:

Modeling:

Using state-space and transfer function approaches for mathematical modeling.

Including external disturbances in the state-space formulation for robustness analysis.

Controller Selection:

PID for simplicity and widespread applicability.

LQR for optimal control with cost function minimization.

MPC for constraint handling and predictive capability.

Simulation Environment:

MATLAB and simscape simulations for tuning and testing the controllers under varying disturbances.

b) Challenges Encountered and Solutions:

Challenge: Implementation of a controller designed for a linear plant model on a nonlinear system.

Solution: Precise tuning of gains and constraints to achieve robust control

Challenge: Tuning PID gains for stability under varying disturbances.

Solution: Iterative tuning combined with trial-and-error using simulation tools.

Challenge: Integrating disturbance models.

Solution: Adapting state-space equations to account for external inputs.

c) What we learned:

PID works effectively for simple scenarios but struggles with complex disturbances.

LQR offers better stability with optimal control effort but depends on accurate cost matrix selection.

MPC handles disturbances best but introduces computational challenges.

d) Next Steps:

Extend the study to hardware implementation on an actual inverted pendulum system.

Explore hybrid controller designs that combine PID simplicity with MPC robustness.

Investigate adaptive control strategies for systems with time-varying dynamics.

References:

Stabilization of inverted pendulum on cart based on LQG optimal control. (2018, December 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/8821243

State Estimation of an Inverted Pendulum on Cart System by Kalman Filtering and Optimal Control (LQG). (2022, May 23). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/983769

Karl.J.Astrom "PID Controllers : Theory, Design and Tuning" 2nd Edition

Model Predictive Control of an Inverted Pendulum. (2010, January 10) | IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/5412069

State Estimation of an Inverted Pendulum on Cart System by Kalman Filtering and Optimal Contro (LQG)| IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/9837694

Robust control of Inverted pendulum using fuzzy logic controller | IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/6547577

Individual Contributions

Deepak Ramesh:

Deepak contributed to the initial brainstorming of the project and conducted comprehensive literature reviews to establish a solid foundation for the research. He was responsible for identifying the problem areas and collaboratively working on determining the set of suitable controllers needed. Deepak also contributed equally to the implementation and refinement of the controllers. Additionally, he helped in validating the results and ensuring the accuracy of the analysis. In collaboration with Kriish, he worked on developing and refining the controllers, ensuring the robustness and effectiveness of the control system.

Kriish Hate:

Kriish actively participated in the brainstorming sessions and worked alongside Deepak to identify the broader context and problem areas of the project. He leveraged the literature reviews conducted by Deepak to gain a deeper understanding of the current state of the art. Kriish also played a significant role in identifying a suitable project topic and narrowing it down to the specific problem addressed. In collaboration with Deepak, he worked on developing and refining the controllers, ensuring the robustness and effectiveness of the control system.