Swing-Up, Stabilization, and Observer-Based State Estimation for the Inverted Pendulum on a Cart

Abhinab Sharma
Mechanical Engineering 24250005
IIT Gandhinagar

Parth Jayeshbhai Joshi
Mechanical Engineering 24310044
IIT Gandhinagar

Samriddhi Dubey
Mechanical Engineering 24250080
IIT Gandhinagar

***Abstract*—This study implements a dual-mode control strategy for an inverted pendulum-cart system, combining energy-based swing-up control with LQR stabilization. To address sensor noise and unobservable states, Luenberger and Kalman filter observers are designed for state estimation. Simulations demonstrate improved control accuracy and robustness using observers compared to noisy direct measurements, underscoring their practical value in real-world dynamic systems.**

***Index Terms*—Inverted pendulum, observer-based control, Luenberger observer, Linear Quadratic Regulator (LQR), energy-based control, sensor noise, state estimation, robust control.**

I. Introduction

The inverted pendulum on a cart is a classic control problem, valued for testing advanced control strategies due to its unstable nature and sensitivity to disturbances. While controlling it in ideal conditions is well understood, real-world scenarios introduce challenges like sensor noise and unmeasurable states, especially velocities, which can severely affect system performance.

Accurate state feedback is essential for balancing the pendulum, but relying directly on noisy or incomplete measurements can lead to poor control and instability. To address this, observer-based control techniques are often used in practical systems. An observer estimates unmeasured or noisy states using available data and a system model, improving noise rejection and reducing dependence on expensive sensors.

In this work, a Luenberger observer is integrated with an energy-based swing-up controller and an LQR for stabilization. The system's performance is then evaluated under simulated sensor noise to demonstrate how the observer improves robustness and control reliability.

II. System Modeling and nomenclature

A. Nonlinear Dynamics

The position coordinates of the pendulum mass are expressed as:

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} x + l\sin\theta \\ -l\cos\theta \end{bmatrix} \tag{1}$$

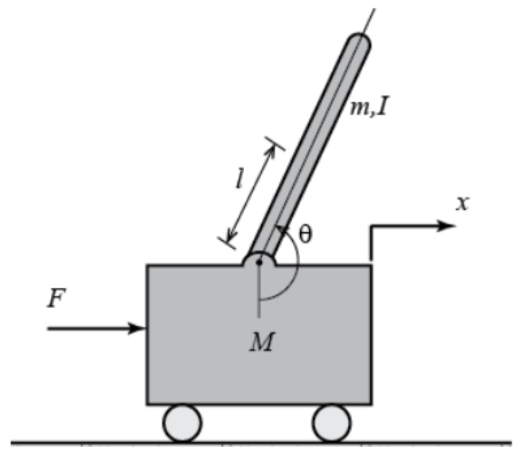

Fig. 1. Free Body Diagram of Inverted Pendulum

The Lagrangian for a system of particles in classical mechanics is given by the difference between the total kinetic and potential energies:

$$\mathcal{L} = T - U \tag{2}$$

The potential energy U associated with the pendulum is:

$$U = -mgl\cos\theta \tag{3}$$

Description	Variable (Matlab)	Value with Units
Mass of pendulum rod	M	0.1 kg
Mass of cart	Mc	0.135 kg
Length to pendulum C.G.	L	0.2 m
Motor rotor inertia	Jm	3.26×10^{-8} kg.m^2
Motor armature resistance	Rm	12.5 Ω
Motor back emf constant	Kb	0.031 V/rad/sec
Motor torque constant	Kt	0.031 N.m/A
Motor pinion radius	R	0.006 m
Pivot viscous damping	B	0.000078 N.m/rad/sec
Cart friction coefficient	C	0.63 N/m/sec
Pendulum inertia about pivot	I	0.00072 kg.m^2
Total mass (cart+motor)	M	0.136 kg
Gravitational acceleration	G	9.81 m/sec^2

TABLE I
System Parameters for Inverted Pendulum on Cart

Total Kinetic Energy of the System

The total kinetic energy T consists of the translational kinetic energy T_t and the rotational kinetic energy T_r:

$$T = T_t + T_r \tag{4}$$

The translational kinetic energy T_t is the sum of the kinetic energy of the cart T_{t_c} and that of the pendulum mass T_{t_p}:

$$T_t = T_{t_c} + T_{t_p} \tag{5}$$

The kinetic energy due to the translation of the cart is:

$$T_{t_c} = \frac{1}{2} M \dot{x}_c^2 \tag{6}$$

The translational kinetic energy of the pendulum mass is given by:

$$T_{t_p} = \frac{1}{2} m \left(\dot{x}_p^2 + \dot{y}_p^2 \right)$$

Substituting the expressions from equation (1), we have:

$$T_{t_p} = \frac{1}{2} m \left[\left(\frac{d}{dt}(x + l \sin\theta) \right)^2 + \left(\frac{d}{dt}(-l \cos\theta) \right)^2 \right]$$

Simplifying the derivatives results in:

$$T_{t_p} = \frac{1}{2} m \left(\dot{x}^2 + l^2 \dot{\theta}^2 + 2 l \dot{x} \dot{\theta} \cos\theta \right) \tag{7}$$

By combining equations (5), (6), and (7), the total translational kinetic energy becomes:

$$T_t = \frac{1}{2} M_c \dot{x}_c^2 + \frac{1}{2} m \dot{x}^2 + \frac{1}{2} m l^2 \dot{\theta}^2 + m l \dot{x} \dot{\theta} \cos\theta \tag{8}$$

The rotational kinetic energy T_r due to the pendulum's angular motion is:

$$T_r = \frac{1}{2} I \dot{\theta}^2 \tag{9}$$

Therefore, the total kinetic energy of the combined system is expressed as:

$$T = \frac{1}{2} M_c \dot{x}_c^2 + \frac{1}{2} m \dot{x}^2 + \frac{1}{2} m l^2 \dot{\theta}^2 + m l \dot{x} \dot{\theta} \cos\theta + \frac{1}{2} I \dot{\theta}^2 \tag{10}$$

Using equations (2), (3), and (10), the Lagrangian for the system is:

$$\mathcal{L} = \frac{1}{2} M_c \dot{x}_c^2 + \frac{1}{2} m \dot{x}^2 + \frac{1}{2} m l^2 \dot{\theta}^2 + m l \dot{x} \dot{\theta} \cos\theta + \frac{1}{2} I \dot{\theta}^2 + m g l \cos\theta \tag{11}$$

Equations of Motion

As the system possesses two degrees of freedom, two Lagrangian equations of motion are formulated:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{x}} \right) - \frac{\partial \mathcal{L}}{\partial x} = F - c\dot{x} \tag{12}$$

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}} \right) - \frac{\partial \mathcal{L}}{\partial \theta} = -b\dot{\theta} \tag{13}$$

Substituting the expression from equation (11) into equations (12) and (13) provides the following nonlinear coupled equations of motion for the cart-pendulum system:

$$(M + m)\ddot{x} + m l \ddot{\theta} \cos\theta - m l \dot{\theta}^2 \sin\theta = F - c\dot{x} \tag{14}$$

$$(I + m l^2)\ddot{\theta} + m l \ddot{x} \cos\theta + m g l \sin\theta = -b\dot{\theta} \tag{15}$$

B. *Linear State-Space Representation of the Inverted Pendulum System*

We start with the non-linear equations of motion for the system:

$$(M + m)\ddot{x} + m l \ddot{\theta} \cos\theta - m l \dot{\theta}^2 \sin\theta = F - c\dot{x} \tag{14}$$

$$(I + m l^2)\ddot{\theta} + m l \ddot{x} \cos\theta + m g l \sin\theta = -b\dot{\theta} \tag{15}$$

From equation (15), we can solve for $\ddot{x}$:

$$\ddot{x} = \frac{-b\dot{\theta} - m g l \sin\theta - (I + m l^2)\ddot{\theta}}{m l \cos\theta} \tag{16}$$

Next, by substituting this expression for $\ddot{x}$ into equation (14), we obtain the following equation for $\ddot{\theta}$:

$$\ddot{\theta} = \frac{-\left(F m l \cos\theta - c m l \dot{x} \cos\theta + m^2 l^2 \dot{\theta}^2 \sin\theta \cos\theta + P \right)}{m^2 l^2 \sin^2\theta + M m l^2 + (M + m) I} \tag{17}$$

where,

$$P = (M + m) \left(b\dot{\theta} + m g l \sin\theta \right)$$

Similarly, solving equation (14) for $\ddot{\theta}$, we get:

$$\ddot{\theta} = \frac{F - c\dot{x} - (M + m)\ddot{x} + m l \dot{\theta}^2 \sin\theta}{m l \cos\theta} \tag{19}$$

By substituting this expression for $\ddot{\theta}$ into equation (15), we obtain:

$$\ddot{x} = \frac{b m l \dot{\theta} \cos\theta + m^2 l^2 g \sin\theta \cos\theta + (I + m l^2)(F - c\dot{x} + m l \dot{\theta}^2 \sin\theta)}{m^2 l^2 \sin^2\theta + M m l^2 + (M + m) I} \tag{20}$$

To express the system in state-space form, we introduce the following state variables:

$$x_1 = x, \quad x_2 = \theta, \quad x_3 = \dot{x}, \quad x_4 = \dot{\theta} \tag{21}$$

We can then write the accelerations $\ddot{x}$ and $\ddot{\theta}$ as:

$$\dot{x}_3 = \frac{bmlx_4 \cos x_2 + m^2 l^2 g \sin x_2 \cos x_2 + Q}{Mml^2 + (M+m)I + m^2 l^2 \sin^2 x_2} \tag{22}$$

$$\dot{x}_4 = \frac{-\left(Fml \cos x_2 - cmlx_3 \cos x_2 + R\right)}{Mml^2 + (M+m)I + m^2 l^2 \sin^2 x_2} \tag{23}$$

Where R and Q are given by:

$$R = m^2 l^2 x_4^2 \sin x_2 \cos x_2 + (M+m)\left(bx_4 + mgl \sin x_2\right)$$

$$Q = \left(I + ml^2\right)\left(F - cx_3 + mlx_4^2 \sin x_2\right)$$

Thus, the final non-linear state-space representation of the inverted pendulum system is:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_3 \\ x_4 \\ \text{(from equation 22)} \\ \text{(from equation 23)} \end{bmatrix} \tag{24}$$

The output equation is:

$$\mathbf{Y} = \mathbf{CX} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} \tag{25}$$

To linearize the system around the upright stationary point, we simply linearize the non-linear system.

The non-linear function is:

$$\mathbf{f}(\mathbf{X}, \mathbf{U}) = \begin{bmatrix} x_3 \\ x_4 \\ \text{(non-linear expression for } \dot{x}_3\text{)} \\ \text{(non-linear expression for } \dot{x}_4\text{)} \end{bmatrix} \tag{26}$$

*1) **Linear Approximation using Taylor's series and Jacobian Matrix**:* We begin with the nonlinear system:

$$\dot{X} = f(X, U)$$

Our goal is to linearize this system around the equilibrium point (X_0, U_0). To achieve this, we express the variables as deviations from their nominal values:

$$X = X_0 + \delta X, \quad U = U_0 + \delta U$$

The dynamics of the perturbed state become:

$$\delta\dot{X} = f(X_0 + \delta X, U_0 + \delta U)$$

By applying a Taylor Series expansion, we approximate:

$$\delta\dot{X} = f(X_0, U_0) + \frac{\partial f}{\partial X}\bigg|_{(X_0, U_0)} \delta X + \frac{\partial f}{\partial U}\bigg|_{(X_0, U_0)} \delta U + (H.O.T.)$$

H.O.T = Higher Order Terms

Assuming that at the operating point, $f(X_0, U_0) = 0$, we can neglect the higher-order terms, leading to the linearized system:

$$\delta\dot{X} = A\delta X + B\delta U$$

The output equation becomes:

$$Y = C\delta X$$

Where:

$$A = \frac{\partial f}{\partial X}\bigg|_{(X_0, U_0)}, \quad B = \frac{\partial f}{\partial U}\bigg|_{(X_0, U_0)}$$

Now, linearizing the nonlinear model around the reference point gives:

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_2}{\partial x_1} & \frac{\partial f_3}{\partial x_1} & \frac{\partial f_4}{\partial x_1} \\ \frac{\partial f_1}{\partial x_2} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_4}{\partial x_2} \\ \frac{\partial f_1}{\partial x_3} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_3}{\partial x_3} & \frac{\partial f_4}{\partial x_3} \\ \frac{\partial f_1}{\partial x_4} & \frac{\partial f_2}{\partial x_4} & \frac{\partial f_3}{\partial x_4} & \frac{\partial f_4}{\partial x_4} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{m^2 l^2 g}{\alpha} & \frac{-\left(I + ml^2\right)c}{\alpha} & \frac{-bml}{\alpha} \\ 0 & \frac{mgl(M_c + m)}{\alpha} & \frac{-mlc}{\alpha} & \frac{-b(M_c + m)}{\alpha} \end{bmatrix}$$

Here, the force F is treated as the input U:

$$U = F$$

The linearized state-space model, with force F as input U, is:

$$\dot{X} = AX + BU$$

Where:

$$B = \begin{bmatrix} \frac{\partial f_1}{\partial F} \\ \frac{\partial f_2}{\partial F} \\ \frac{\partial f_3}{\partial F} \\ \frac{\partial f_4}{\partial F} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{I + ml^2}{\alpha} \\ \frac{ml}{\alpha} \end{bmatrix}$$

Thus, the linear state-space model is:

$$\dot{x} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{m^2 l^2 g}{\alpha} & \frac{-\left(I + ml^2\right)c}{\alpha} & \frac{-bml}{\alpha} \\ 0 & \frac{mgl(M_c + m)}{\alpha} & \frac{-mlc}{\alpha} & \frac{-b(M_c + m)}{\alpha} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{I + ml^2}{\alpha} \\ \frac{ml}{\alpha} \end{bmatrix} F \tag{20}$$

2) Motor Dynamics: Force to voltage conversion: Finally, the force F applied on the cart is generated by the PMDC motor. The relation between F and the applied voltage V_m is given by:

$$F = \frac{k_t V_m r - k_t k_b \dot{x}}{R_m r^2} \tag{21}$$

Substituting this into the linear model, the final linear state-space model becomes:

$$\dot{X} = AX + BV_m$$

$$\dot{x} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{m^2l^2g}{\alpha} & \frac{-(I+ml^2)c}{\alpha} & \frac{-bml}{\alpha} \\ 0 & \frac{mgl(M_c+m)}{\alpha} & \frac{-mlc}{\alpha} & \frac{-b(M_c+m)}{\alpha} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{(I+ml^2)K_t}{\alpha R_m r} \\ \frac{mlK_t}{\alpha R_m r} \end{bmatrix} V_m$$

$$\dot{X} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 5.51 & -18.29 & -0.002 \\ 0 & 64.9 & -77.53 & -0.026 \end{bmatrix} \begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 2.73 \\ 11.59 \end{bmatrix} V_m$$

The output equation is:

$$Y = CX = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix}$$

III. Stability, Controllability and Observability

A. Stability Test

To check the stability of the system, we analyze the eigenvalues of the system's state matrix A. These eigenvalues correspond to the poles of the system's transfer function. The system will be stable if all the eigenvalues have negative real parts, meaning they lie in the left half of the complex plane.

The eigenvalue equation is:

$$A\mathbf{v} = \lambda \mathbf{v}$$

where λ represents the eigenvalues, and $\mathbf{v}$ is the corresponding eigenvector.

To determine the stability, we compute the eigenvalues of matrix A, and a system is stable if the real part of all eigenvalues is negative. Matlab Code fro stability check can be found in Appedices.

MATLAB Output:

```
Eigenvalues (Poles):
   0.0000
 -19.6327
   6.9178
  -5.6006

The system is unstable.
```

From the output, we can see that the system has one eigenvalue with a positive real part (6.9178), which indicates that the system is unstable.

B. Controllability

Complete state controllability (or simply controllability in the absence of any further context) refers to the capacity of an external input to transition the system's internal state from any initial condition to any final state within a finite time interval.

Controllability Matrix

For linear time-invariant (LTI) systems, the system is considered reachable if and only if the controllability matrix, denoted by ζ, has full row rank equal to p, where p is the dimension of matrix A, and $p \times q$ is the dimension of matrix B.

$$\zeta = [B \ AB \ A^2B \ \dots \ A^{p-1}B] \in \mathbb{R}^{p \times pq}$$

The system is controllable if the rank of the system matrix A equals p, and the rank of the controllability matrix satisfies:

$$\text{Rank}(\zeta) = \text{Rank}(A^{-1}\zeta) = p$$

In MATLAB, you can easily generate the controllability matrix using the `ctrb` function. To compute the controllability matrix ζ, simply type:

```
>> zeta = ctrb(A, B);

>> Rank = rank(zeta);
```

To verify the controllability of the system, we calculate the rank of the controllability matrix ζ using the `ctrb` command in MATLAB. Matlab Code to find the controllability check can be found in Appendices. **Output:**

```
Rank of Controllability Matrix: 4
The system is CONTROLLABLE.
```

Since the rank of the controllability matrix is equal to $p = 4$ (the number of states), the system is fully controllable.

C. Observability Test

To determine whether the given system is observable, we construct the observability matrix using the MATLAB `obsv` command. The system is observable if the rank of the observability matrix is equal to the number of states in the system.

The state-space matrices used are:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 5.51 & -18.29 & -0.002 \\ 0 & 64.9 & -77.53 & -0.026 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The MATLAB code used to check observability is provideed in Appendices

***Result:**:* The output obtained from MATLAB is:

```
Rank of Observability Matrix: 4
The system is observable.
```

Hence, since the rank of the observability matrix is equal to the number of states (4), **the system is observable**.

IV. CONTROLLER DESIGN

POLE PLACEMENT CONTROL

A. *Introduction to Pole Placement*

Pole placement is a control design method that assigns specific eigenvalues (or poles) to the closed-loop system. By modifying the state feedback matrix K, we can place the poles in desired locations, affecting the system's stability and transient response.

The state-space system is represented as:

$$\dot{X} = AX + BV_m$$

with state feedback:

$$V_m = -KX$$

The closed-loop dynamics are:

$$\dot{X} = (A - BK)X$$

where the goal is to compute K to achieve desired eigenvalues for the system.

The MATLAB code provided in Appendices calculates the gain matrix K.

B. *Output from Pole Placement*

The output for the state feedback gain matrix K is:

$$K = [185.1640, 219.1039, -7.8073, 6.6095]$$

This matrix stabilizes the system by placing the poles at the desired locations, ensuring desired transient and stability properties.

C. *Why Do We Need LQR?*

Although pole placement is effective, it has some limitations:

- LQR minimizes a cost function that balances state errors and control effort, optimizing system performance and efficiency.
- LQR automatically computes the optimal gain matrix K, making it more efficient for complex systems.

LQR BALANCE

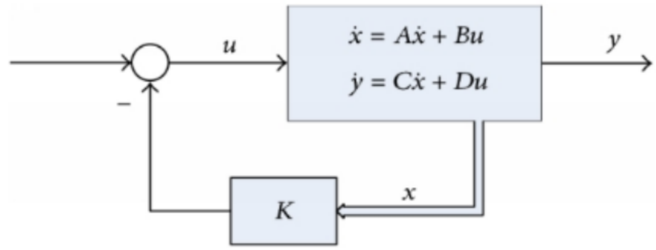

Fig. 2. Block Diagram of LQR

Linear Quadratic Regulator: Definition and Functionality

The Linear Quadratic Regulator (LQR) is an optimal control strategy used in modern control systems to regulate the state of a linear dynamic system to the origin or to track a desired trajectory with minimal effort. It achieves this by minimizing a quadratic cost function that penalizes deviations from the desired state as well as excessive control inputs.

The objective is to design a control law of the form:

$$V_m(t) = -KX(t)$$

which minimizes the cost function:

$$J = \int_0^\infty \left(X^T Q X + V_m^T R V_m\right) dt$$

Optimal Full State Feedback Control Law

For a linear time-invariant (LTI) system:

$$\dot{X} = AX + BV_m$$

the optimal full state feedback control law is:

$$V_m = -KX$$

where the feedback gain matrix K is given by:

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

and P is the unique, symmetric, positive semi-definite solution to the Algebraic Riccati Equation (ARE):

$$PA + A^T P + Q - PBR^{-1}B^T P = 0$$

Given System Parameters

$$Q = \begin{bmatrix} 1200 & 0 & 0 & 0 \\ 0 & 1500 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad R = 0.035$$

MATLAB Implementation

The following MATLAB code provided in Appendices defines the system dynamics matrices A and B, cost matrices Q and R, and computes the optimal gain matrix K:

LQR Gain Result

Running the above code yields the following optimal gain matrix:

$$K = \begin{bmatrix} 185.1640 & 219.1039 & -7.8073 & 6.6095 \end{bmatrix}$$

Interpretation

This gain matrix K is used to apply the control input:

$$V_m = -KX$$

RESULTS AND DISCUSSION

The system uses a state-space representation with four state variables:

- Cart position (x)
- Pendulum angle (θ)
- Cart velocity ($\dot{x}$)
- Pendulum angular velocity ($\dot{\theta}$)

1) ***LQR Controller Design****:* The LQR controller is designed with these key characteristics:

- State weighting matrix $Q = \text{diag}([1200, 1500, 0, 0])$, heavily prioritizing position and angle
- Control effort weighting $R = 0.035$, allowing relatively high control effort
- The control objective is to stabilize the pendulum in the upright position ($\theta = \pi$) with the cart at the origin

Based on the provided graphs in figure 3, we can analyze the performance of the LQR controller implemented for the inverted pendulum system:

Cart Position (x):

The graph shows the cart starting at the initial position of 0.2m and quickly converging to the desired position (origin). The response exhibits excellent damping characteristics with minimal undershoot around the 1-second mark. The cart stabilizes at the target position within approximately 2 seconds, demonstrating effective position control.

Cart Velocity ($\dot{x}$):

The velocity response shows a brief spike to approximately 3-4 m/sec in the initial phase as the controller aggressively moves to stabilize the system. The velocity then quickly returns to zero with excellent damping. After the initial transient response, the velocity remains at zero, indicating stable control with no oscillations or drift.

Pendulum Angle (θ):

Starting from the initial angle of 160 degrees, the pendulum rapidly approaches the desired upright position (180 degrees). The angle response shows a slight undershoot to about 177 degrees around the 1-second mark before smoothly converging to 180 degrees. This demonstrates the controller's ability to balance the pendulum in the naturally unstable inverted position.

Pendulum Angular Velocity ($\dot{\theta}$):

The angular velocity graph shows minimal movement throughout the control period. The y-axis scale (-200 to 600 deg/sec) is much larger than the actual values, indicating that the angular velocity remains close to zero. This suggests excellent damping of rotational oscillations and stable control.

V. ENERGY-BASED SWING-UP CONTROLLER DESIGN

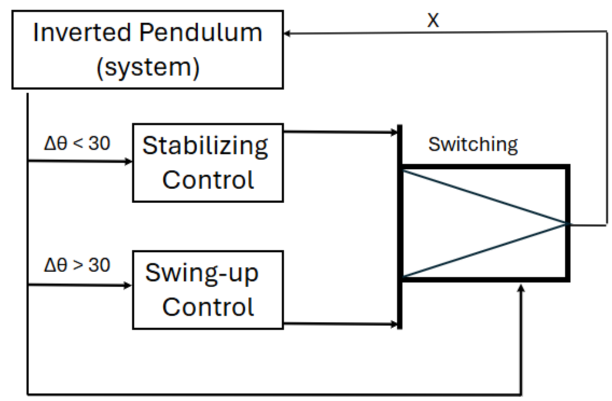

Fig. 4. Control Scheme of Inverted Pendulum

The swing-up controller is typically applied after the inverted pendulum has been stabilized in the upright position using the LQR controller. The LQR ensures the pendulum remains upright under small disturbances, stabilizing the system. However, when the pendulum is far from the upright position (in the downward state), the LQR struggles to bring the pendulum up due to the high control effort needed to overcome gravitational forces.

Thus, after stabilization, the **Swing-Up Controller** is introduced to lift the pendulum from the downward to the upright position.

A. Why the Swing-Up Controller is Necessary

In the downward position, the pendulum is unstable, and small control inputs are insufficient to lift it. While LQR is effective near the upright position, it cannot swing the pendulum up. A swing-up controller injects the required energy to move the pendulum upright, after which LQR efficiently stabilizes it with minimal effort.

B. Transition from Inverted Pendulum to Swing-Up Controller

The system undergoes two phases:

- **LQR Stabilization (Upright Position)**:
  - The pendulum is in the upright position ($\theta = \pi$).
  - The LQR controller ensures balance by correcting any small disturbances in angle or velocity.
- **Swing-Up (Bringing Pendulum to Upright)**:
  - The pendulum starts from the downward position, and the goal is to use the swing-up controller to inject energy into the system.

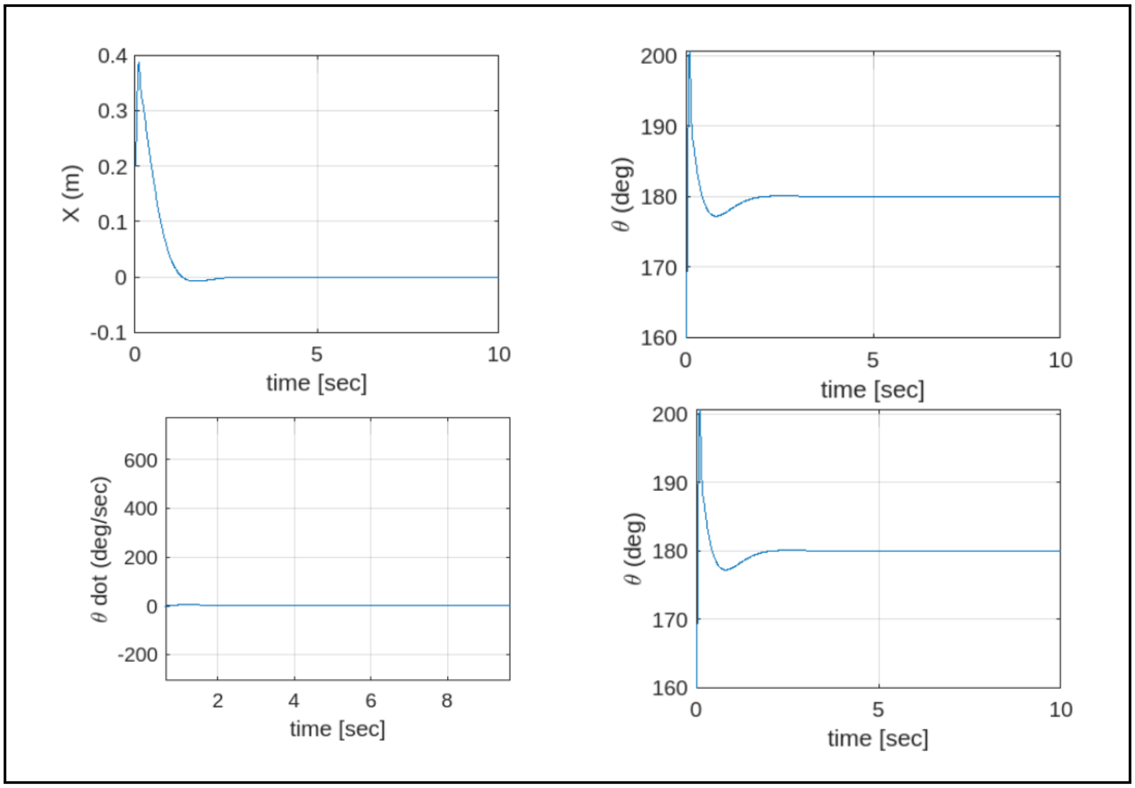

Fig. 3. Cart position, velocity and pendulum position, velocity vs. time for LQR-controlled inverted pendulum.

Swing Up Control Phase

C. Collocated Partial Feedback Linearization (PFL)

We begin by simplifying Equation (15) for $\ddot{\theta}$:

$$\ddot{\theta} = \frac{-b\dot{\theta} - mlx\ddot{d}\cos\theta - mgl\sin\theta}{I + ml^2} \tag{22}$$

Substituting $\ddot{\theta}$ from Equation (22) into Equation (14), we obtain the nonlinear equation for the control force:

$$F = (M + m)\,\ddot{x}_d + c\,\dot{x} - m\,l\,\dot{\theta}^2\sin\theta$$
$$-m\,l\left(\frac{b\,\dot{\theta} + m\,l\,\ddot{x}_d\cos\theta + m\,g\,l\sin\theta}{I + m\,l^2}\right)\cos\theta$$

For an appropriate control law, energy shaping control suggests that:

$$\ddot{x}_d = u$$

Substituting $\ddot{x}_d = u$ into Equation (22), we get:

$$\ddot{\theta} = \frac{-b\dot{\theta} - mlu\cos\theta - mgl\sin\theta}{I + ml^2} \tag{24}$$

The total energy of the simple pendulum is given by:

$$E = \frac{1}{2}(I + ml^2)\dot{\theta}^2 + mgl(1 - \cos\theta) \tag{25}$$

The desired energy, which corresponds to the energy at the desired fixed point, is:

$$E_r = mgl(1 - \cos\pi) = 2mgl \tag{26}$$

If we define the error dynamics as $E_t = E - E_r$, we differentiate with respect to time:

$$\dot{E}_t = \dot{E} = (I + ml^2)\dot{\theta}\ddot{\theta} + mgl\sin\theta\dot{\theta}$$

Substituting the value of $\ddot{\theta}$ from Equation (24) into the above equation, we get:

$$\dot{E}_t = -b\dot{\theta}^2 - mlu\dot{\theta}\cos\theta$$

Now, if we design a controller of the form:

$$u = k\dot{\theta}\cos\theta E_t, \quad k > 0$$

The resulting error dynamics are:

$$\dot{E}_t = -b\dot{\theta}^2 - mlk\dot{\theta}^2\cos^2\theta E_t$$

If the damping term b is very small, it can be neglected. The error dynamics imply exponential convergence:

$$E_t \to 0, \quad \text{except for states where } \dot{\theta} = 0.$$

The essential property of this system is that when $E > E_r$, we remove energy from the system (damping), and when $E < E_r$, we add energy (negative damping). A tunable gain k is multiplied by the control law, and the controller is saturated at the maximum acceleration deliverable by the motor, u_{max}:

$$u = \text{sat}_{\max}\Big(k(E - E_r)\,\text{sign}(\dot{\theta}\cos\theta)\Big) \tag{27}$$

From Equation (27) and Equation (23), we derive the Collocated Partial Feedback Linearization (PFL) control law, which represents the appropriate nonlinear control law in terms of force.

Results and Discussion

Analysis of Swing-Up Control Implementation for Inverted Pendulum

Figure 5 presents the two-phase control strategy: swing-up followed by LQR stabilization. MATLAB code is included in the appendices.

D. Swing-Up Phase (0–5 seconds)

- **Angular Velocity ($\dot{\theta}$):** Large oscillations (±700 deg/s) add energy to the system, increasing swing amplitude.
- **Pendulum Angle (θ):** Increasing oscillations show energy build-up. The 210° red line marks the LQR activation boundary.
- **Cart Position (x):** The cart shifts to 0.55 m with small oscillations, creating a moving pivot for better energy transfer.
- **Cart Velocity ($\dot{x}$):** A peak at 1.8 m/s is followed by oscillations, helping generate pendulum momentum.

LQR Stabilization Phase (5–10 seconds)

- **Angular Velocity ($\dot{\theta}$):** Rapidly settles to zero, confirming successful damping.
- **Pendulum Angle (θ):** Quickly stabilizes at 180° with minimal overshoot, showing precise control.
- **Cart Position (x):** Smoothly returns to origin, indicating effective balance and minimal displacement.
- **Cart Velocity ($\dot{x}$):** A brief negative spike is followed by quick settling, demonstrating good motion control.

VI. Extension of the Problem: Limited State Measurement

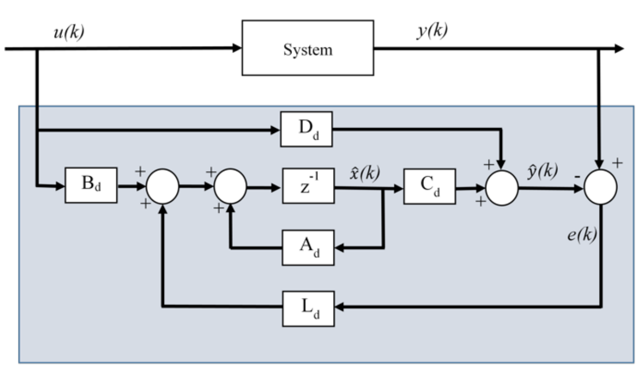

Fig. 6. Block Diagram of Luenberger Observer

In the classical inverted pendulum problem, the complete state vector is defined as:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \text{Cart Position} \\ \text{Cart Velocity} \\ \text{Pendulum Angle} \\ \text{Pendulum Angular Velocity} \end{bmatrix}$$

However, in practical scenarios, it is often difficult or expensive to measure all four states directly. In this problem, we assume that only two of the four states are measurable:

- Cart position x_1
- Pendulum angle x_3

The cart velocity x_2 and the pendulum's angular velocity x_4 remain unmeasured.

Need for a State Observer

Since only partial state feedback is available, we cannot directly apply the Linear Quadratic Regulator (LQR) designed for full state feedback. To estimate the complete state vector from the available measurements, we need a **state observer**.

The measured output can be represented as:

$$y = Cx$$

where C is the measurement matrix:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Observer Design

We employ a **Luenberger Observer** to estimate the unmeasured states. The observer equation is structured as:

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x})$$

where:

- $\hat{x}$ is the estimated state vector.
- u is the control input.
- L is the observer gain matrix.

The term $L(y - C\hat{x})$ represents the correction based on the difference between the measured and estimated outputs. The observer gain L is selected to place the poles of the observer at locations that ensure fast convergence of the estimated states to the true states.

The observer poles are often chosen to be significantly faster than the closed-loop system poles, for example:

$$\text{Observer Poles} = 3 \times \text{Closed-Loop Poles}$$

This ensures rapid error decay and accurate state reconstruction for use in the feedback controller.

The final control input is computed using the estimated state:

$$u = -K\hat{x}$$

where K is the LQR gain matrix designed for full state feedback.

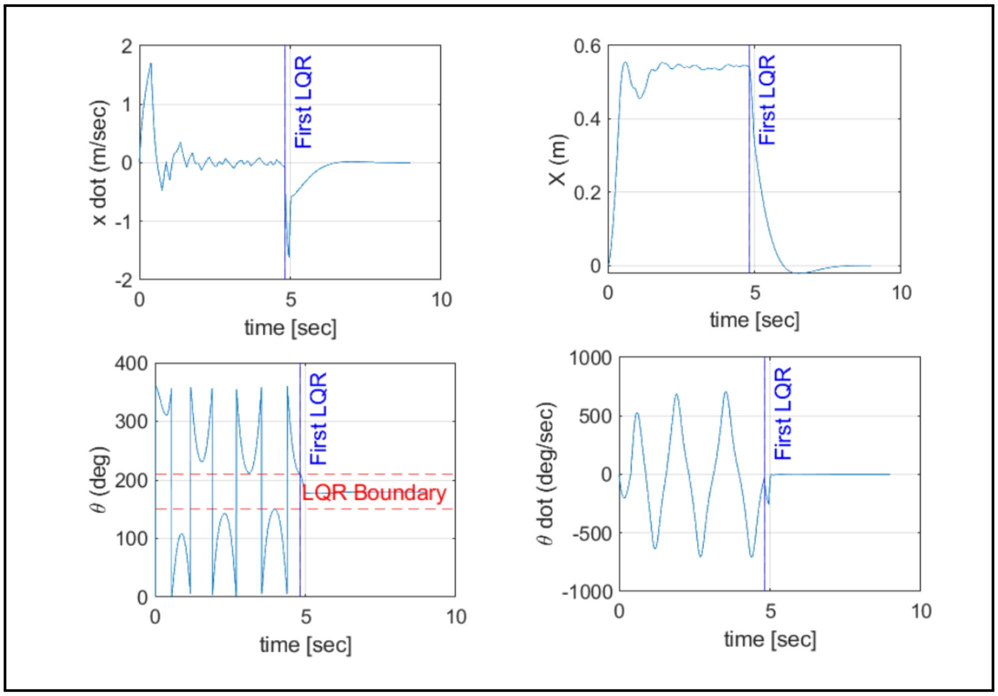

Fig. 5. Cart position, velocity and pendulum position, velocity vs. time for Swing-Up and LQR-controlled inverted pendulum.

RESULTS AND DISCUSSION

The analysis in Figure 7 evaluates the performance of a Luenberger observer paired with an LQR controller for an inverted pendulum system. The graphs compare the actual system states (blue solid lines) with the estimated states (red dashed lines) over a 10-second simulation. MATLAB code is included in the Appendices.

Observer Performance Analysis

- **Cart Position** (x)**:**
  The observer estimate, initialized at zero, quickly converges to the true position of 0.1 m, with minor undershoot around $t = 1$ s. Both signals stabilize at zero, confirming accurate estimation without direct velocity feedback.
- **Cart Velocity** ($\dot{x}$)**:**
  Starting from zero, the observer rapidly matches the true velocity (initially -0.25 m/s) within 0.7 seconds. Both signals converge smoothly to zero with minimal error.
- **Pendulum Angle** (θ)**:**
  Initially at 0.1 radians, the angle estimate converges to the true signal within 0.8 seconds. Both stabilize at $\theta = 0$, verifying the observer's accuracy and the LQR's effectiveness.
- **Angular Velocity** ($\dot{\theta}$)**:**
  Despite starting at zero, the observer accurately captures the initial spike (-0.4 rad/s) and aligns with the true state within 0.8 seconds. Both signals then stabilize at zero, indicating excellent state reconstruction.

VII. State Estimation for Inverted Pendulum using Kalman Filter

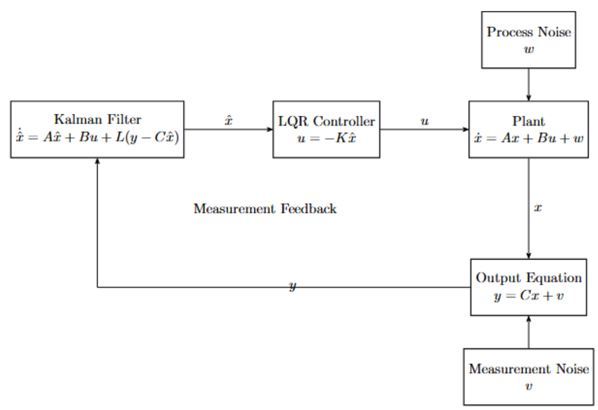

Fig. 8. Block diagram of an LQR-Kalman Filter (LQG) control system.

VIII. State Estimation using Kalman Filter for Inverted Pendulum on a Cart

A. *Objective*

The objective of this simulation is to stabilize an inverted pendulum mounted on a moving cart using a Linear Quadratic Regulator (LQR) controller and to estimate the full system state using a Kalman Filter-based observer. Additionally, a Luenberger observer is implemented for comparison with the

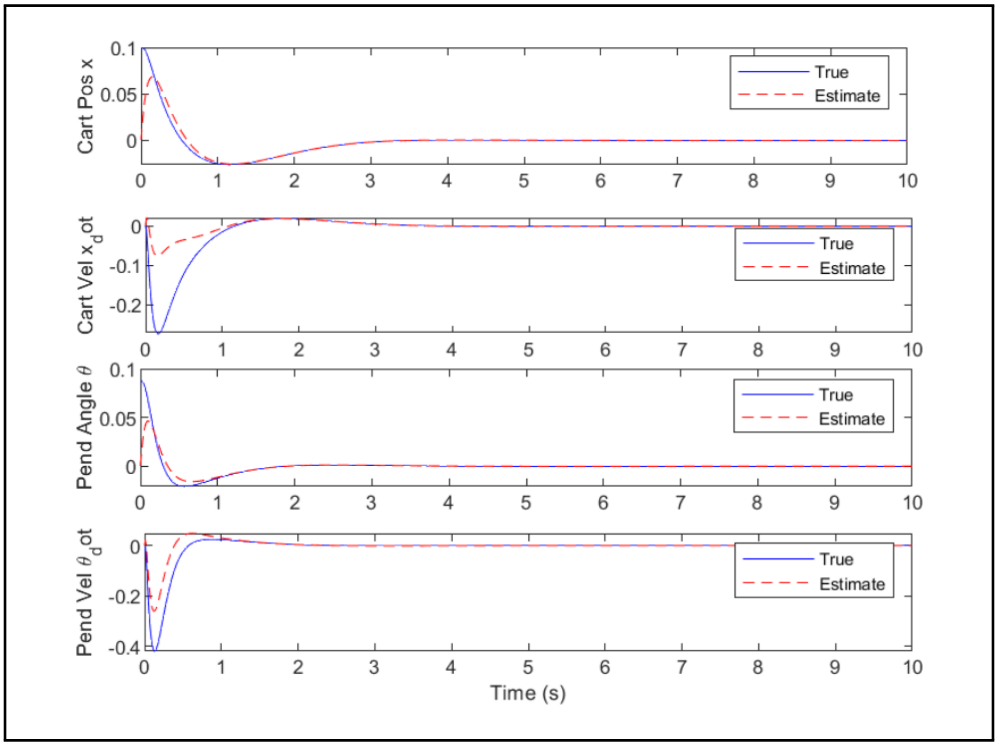

Fig. 7. Cart position, velocity and pendulum position, velocity vs. time With Luenberger Observer.

Kalman Filter in the presence of process and measurement noise.

B. System Overview

The system under consideration is a classical inverted pendulum on a cart, which is a standard benchmark problem in control theory due to its inherently unstable nature. The system is modeled using a linearized state-space representation around the upright equilibrium point.

The state vector is defined as:

$$\mathbf{x} = \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix}$$

where x is the cart position, $\dot{x}$ is the cart velocity, θ is the pendulum angle from vertical, and $\dot{\theta}$ is the pendulum angular velocity.

The control input is the force applied to the cart:

$$u = \text{force on the cart}$$

The output vector $\mathbf{y}$ consists of only two directly measurable states:

$$\mathbf{y} = \begin{bmatrix} x \\ \theta \end{bmatrix} = C\mathbf{x}$$

C. Motivation for Using the Kalman Filter

In real-world scenarios, all system states are not always measurable, and measurements are often corrupted by noise. The Kalman Filter is a powerful optimal estimation algorithm that:

- Estimates the full system state from noisy and incomplete measurements.
- Provides statistically optimal estimates under Gaussian noise assumptions.
- Adapts the estimation based on the known covariances of process and measurement noise.

In this simulation, both process noise $\mathbf{w}(t)$ and measurement noise $\mathbf{v}(t)$ are modeled as zero-mean Gaussian noise with covariance matrices Q_n and R_n respectively:

$$\mathbf{w}(t) \sim \mathcal{N}(0, Q_n), \quad \mathbf{v}(t) \sim \mathcal{N}(0, R_n)$$

Given that only two out of the four states are measurable, the Kalman Filter is justified and necessary for reconstructing the full state vector used in feedback control.

D. Design Methodology

1) LQR Controller: The LQR controller is designed using the standard approach:

$$u = -K\hat{\mathbf{x}}_k$$

where K is the optimal gain matrix computed using the solution to the Algebraic Riccati Equation (ARE). The controller uses the estimated state from the observer (Kalman Filter or Luenberger) instead of the true state.

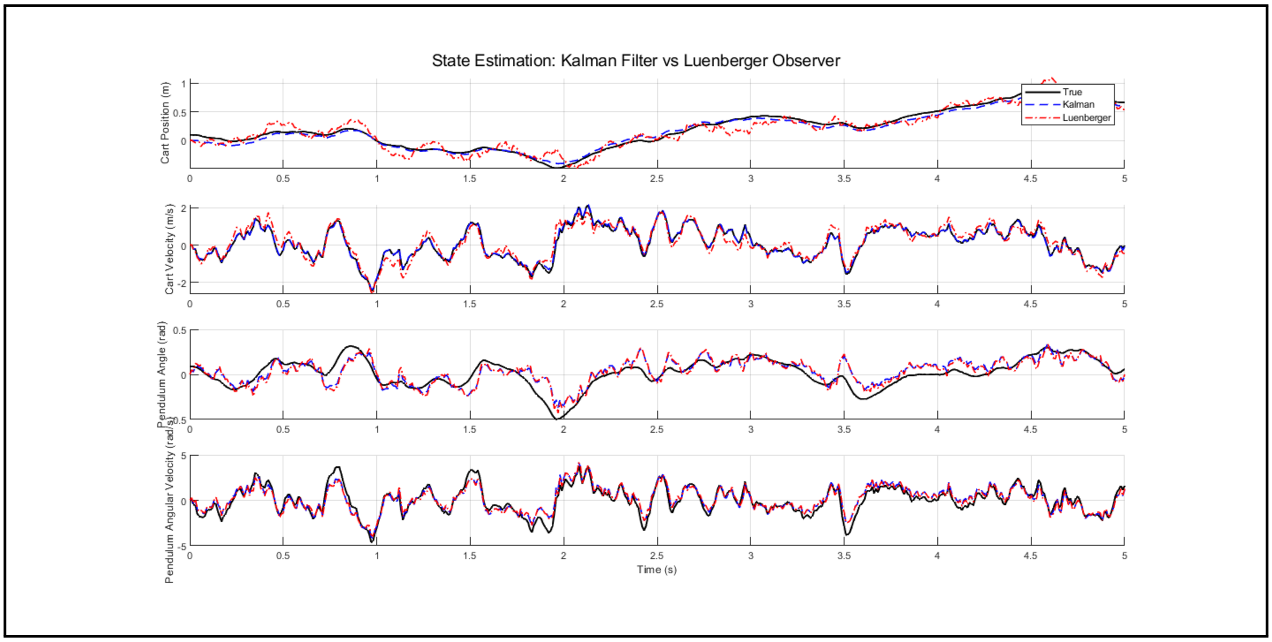

Fig. 9. Comparison of true states, Kalman filter estimates, and Luenberger observer estimates for the inverted pendulum system

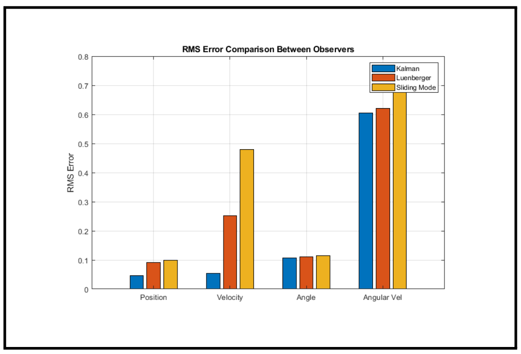

Fig. 10. RMS error comparison between the observers used in the work

*2) **Kalman Filter**:* The continuous-time Kalman Filter (equivalently, the Linear Quadratic Estimator, LQE) is implemented with the gain L obtained from:

$$\dot{\hat{\mathbf{x}}}_k = A\hat{\mathbf{x}}_k + Bu + L(\mathbf{y} - C\hat{\mathbf{x}}_k)$$

The gain L is designed to minimize the estimation error covariance under the noise model defined by Q_n and R_n.

*3) **Luenberger Observer**:* For comparison, a Luenberger observer is designed using pole placement. While it does estimate the state, it does not incorporate knowledge about noise statistics, making it less robust in noisy environments.

E. Simulation Setup

The system is simulated over 20 seconds with a sampling interval of 0.01 seconds. Gaussian noise is added to both the process dynamics and the measurements. The control input at each step is computed using the LQR feedback from the Kalman estimate.

F. Results and Discussion

The performance of both the Kalman Filter and the Luenberger observer as shown in Figure 9 is evaluated by comparing their estimated states against the true system states. The results clearly show that:

- The Kalman Filter provides smoother and more accurate state estimates, especially in the presence of noise.
- The Luenberger observer performs adequately in noise-free conditions but becomes less reliable as noise increases.
- The use of the Kalman Filter is fully justified due to its statistical treatment of noise and partial observability of the system.

The animation and plots further illustrate the close agreement between the true and Kalman-estimated trajectories of the cart and pendulum. Matlab code is provided in Appendices.

RMS Error Comparison Between Observers

The RMS error comparison across Position, Velocity, Angle, and Angular Velocity in 10reveals that the Kalman observer consistently achieves the lowest errors, particularly in position and velocity estimation. While all observers perform similarly for angle, Kalman outperforms both Luenberger and Sliding Mode in the remaining states. Sliding Mode generally exhibits the highest errors, especially in velocity and angular velocity, whereas Luenberger provides moderate accuracy. Overall, the Kalman observer emerges as the most reliable choice for accurate state estimation in noisy environments.

IX. Simulation Video Link

The complete project simulations are available at the following link: Google Drive Folder.

X. CONCLUSION

This study presents a robust hybrid control strategy for stabilizing an inverted pendulum on a cart, combining energy-based swing-up control with LQR for fine stabilization. Switching logic ensures smooth transitions and continuous stability. To address sensor noise and partial state availability, both Luenberger and Kalman observers were implemented, with the Kalman filter outperforming due to its noise modeling capabilities. The system demonstrated reliable performance, achieving fast convergence even from large initial deviations. This work underscores the effectiveness of observer-based control and suggests potential for further enhancement through adaptive or nonlinear observer techniques.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to Prof. Madhu Vadali, Mechanical Engineering, IIT Gandhinagar and Mr.Suraj Borate, research scholar and teaching assistant for the course for their invaluable support and guidance throughout this project. Their insights, encouragement, and assistance were instrumental in the successful completion of this work. We are deeply thankful for their willingness to share their expertise and for the time and effort they dedicated to helping us at every stage of the project.

This project was undertaken with the primary objective of enhancing our understanding and skills in Modern Control Theory (MCT). We explored various theoretical concepts and applied them through simulations, making thoughtful use of AI tools. We sincerely thank our instructors for their guidance and support throughout this enriching learning experience.

REFERENCES

[1] Control Tutorials for MATLAB and Simulink. (n.d.). *Inverted Pendulum: System Modeling*. Retrieved from https://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum&section=SystemModeling

[2] Owais, M., Ul-Haque, A., Rahim, H. A., Aftab, S., & Jalal, A. A. (2019). Control design and implementation of an inverted pendulum on a cart. *2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, Kuala Lumpur, Malaysia, 1-6. https://doi.org/10.1109/ICETAS48360.2019.9117388

[3] Brogan, W. L. (1991). *Modern control theory*.

[4] MathWorks. (n.d.). *Control of an inverted pendulum on a cart*. Retrieved from https://www.mathworks.com/help/control/ug/control-of-an-inverted-pendulum-on-a-cart.html

[5] Swing-Up and stabilization of an inverted pendulum. (n.d.). Retrieved from https://drive.google.com/file/d/1W2v3wKXBVW4FohB33kTv8iBEiOFgoS8d/view