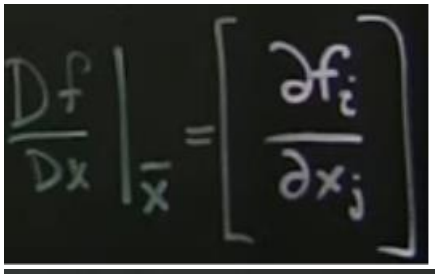
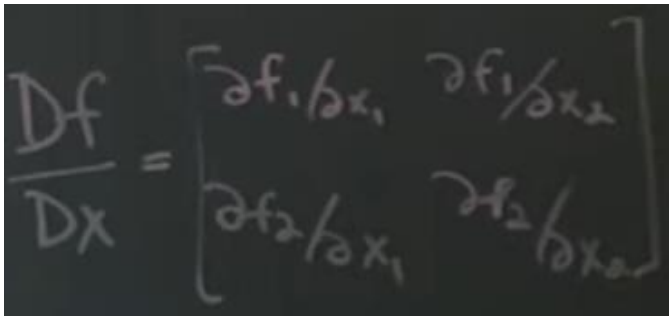
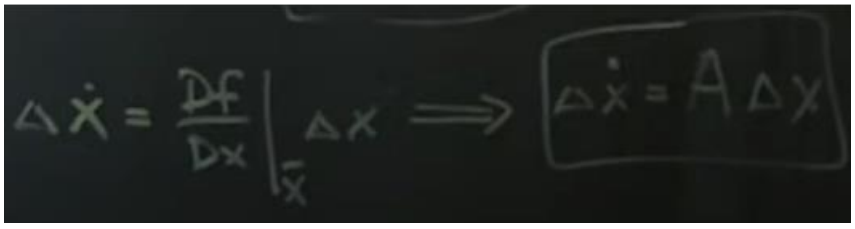
LQR BALANCER

- CONTROL SYSTEM PLAYLIST
  - LINEAR SYSTEM X.=AX
  - We find the eigen values by [A-λI] method (this we will use ahead in pole placement method)
  - If the system is non linear then
    1) Find a fixed point $\bar{x}$
    2) Linearize the equation around $\bar{x}$ by





This gives us linearized equation



Note that we consider these things in a very close proximity of the fixed point so we can consider assumptions like sinθ=θ

  - Here we can use that A as that is linearized
  - CONTROLLABLITY

    The equation we use is

$$\dot{x}=Ax+Bu$$

- Here the u is the actuator

LAGRANGIAN

THIS FORMULATION IS A METHOD TO DERIVE THE EQUATIONS OF MOTION FOR A SYSTEM USING THE POTENTIAL ENERGY AND KINETIC ENERGY.

THIS IS ALSO KNOWN AS LAW OF LEAST ACTION HENCE THE SYSTEM WILL TAKE THE LEAST ENERGY PATH

L=KE-PE

THEN THE EQUATION

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = 0$$

THEN WE GET EQUATION FOR qi..

HERE IF WE EQUATE THE LHS TO u INSTEAD OF 0 THEN WE GET THE ACTUATOR EQUATION

WE USE THIS IN A MATRIX AND THEN SOLVE IT TO USE THE EFFORT PROVIDED BY US IN THE SYSTEM

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = \tau$$

POLE PLACEMENT METHOD

Pole placement, also known as pole assignment, is a method used in control theory to design controllers that place the poles of a closed-loop system at specific, desired locations in the complex plane. This technique is crucial in shaping the system's response characteristics, such as stability, speed, and damping.

- In control systems, the poles of a transfer function or the eigenvalues of the system matrix in state-space representation determine the stability and dynamic response of the system.
- For a continuous-time system, poles in the left half of the complex plane indicate a stable system, while poles in the right half indicate instability.
- In a discrete-time system, poles must be inside the unit circle in the complex plane for stability.

Controllability: Before applying pole placement, it's essential to check that the system is controllable. This ensures that it's possible to place the poles at the desired locations using state feedback. A system is controllable if the controllability matrix has full rank.

HOW TO CHECK IF THE SYSTEM IS CONTROLLABLE OR NOT??

This is a system in state space form

$$\dot{x}(t) = Ax(t) + Bu(t)$$

There is a controllability matrix C

$$\mathcal{C} = [B\ AB\ A^2B\ \cdots\ A^{n-1}B]$$

Here n is the number of state variables

Now if Rank(A)= n …then the system is controllable

And

Rank(A) < n …then the system is not controllable

LQR THEORY (LINEAR QUADRATIC REGULATOR)

- It aims to balance system performance and control effort by minimizing a quadratic cost function.

In our project, the system under control is modelled using state-space representation, where:

- **State Equation:**

$$\dot{x}(t) = Ax(t) + Bu(t)$$ INPUT

Here, x(t) represents the state vector of the system, u(t) is the control input, A is the system matrix, and B is the input matrix.

- **Output Equation:**

where y(t) is the output vector, C is the output matrix, and D is the feedforward matrix.

$$y(t) = Cx(t) + Du(t)$$ OUTPUT

Cost Function

The LQR algorithm minimizes a cost function that is designed to balance performance and control effort. The cost function is given by:

$$J = \int_0^\infty (x^T Q x + u^T R u)\, dt$$

where:

- Q is a positive semi-definite matrix that penalizes deviations from the desired state. It reflects how much we care about the state variables.
- R is a positive definite matrix that penalizes the control effort. It reflects how much we care about the amount of control input used.

Optimal Control Law

The LQR algorithm provides an optimal control law of the form:

$$u(t) = -Kx(t)$$

where K is the state feedback gain matrix. This control law minimizes the cost function J by adjusting the control input u(t) based on the current state x(t).

Calculation of Gain Matrix K

To determine the optimal gain matrix K, we solve the Algebraic Riccati Equation (ARE):

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

where P is a positive definite matrix. Once P is found, the gain matrix K is calculated as:

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

On MATLAB we implement this as following

```
Q = [5000 0; 0 10000]; % State cost matrix
R = 70;          % Control effort cost
K = lqr(A, B, Q, R); % LQR gain
```

EFFECT OF R IN THE SYSTEM

1. Increasing the Elements of R :

-Higher Weight on Control Effort: Increasing the diagonal elements of R increases the cost associated with using large control inputs. The LQR controller will be more conservative in its use of control inputs, aiming to minimize control effort.

- Less Aggressive Control: With a higher R, the controller becomes less aggressive, applying smaller control actions to avoid large penalties associated with control effort. This may result in slower convergence to the desired state or reduced control precision.

2. Decreasing the Elements of R :

-Lower Weight on Control Effort: Decreasing the diagonal elements of R reduces the penalty for using control inputs. The controller can afford to use larger control inputs to quickly correct deviations from the desired state.

- More Aggressive Control: A lower R leads to more aggressive control actions, as the cost of control effort is less emphasized. This can result in faster response times and more precise control but may also lead to higher energy consumption or actuator wear.

3. Balancing Control and State Costs:

-Trade-Off Considerations: The choice of R reflects a trade-off between minimizing the state error and minimizing control effort. A well-chosen R balances the need for efficient control with the desire to minimize deviations from the desired state.

- Impact on Stability and Performance: While a lower R can enhance control precision, it may also increase the risk of instability or excessive control activity, especially in the presence of noise or model inaccuracies. Conversely, a higher R can lead to more stable but potentially less responsive control.

4. Design Considerations:

- Physical Constraints and Costs: In practical systems, the selection of R often reflects physical constraints (e.g.,

maximum actuator force) or operational costs (e.g., energy consumption). The LQR design allows these considerations to be incorporated into the control law.

EFFECT OF Q ON SYSTEM

The Q matrix determines how much importance is placed on minimizing the state errors. It is a weighting matrix that can emphasize certain components of the state vector over others. Here's how changing the Q matrix affects the controller and system behavior:

1. Increasing the Elements of Q :

-Higher Weight on State Errors: Increasing the diagonal elements of Q associated with certain state variables (like angle or velocity) increases the cost associated with deviations in those states. The LQR controller will act more

aggressively to minimize errors in those specific state variables.

- More Aggressive Control: A higher Q value often leads to a more aggressive control action, as the controller prioritizes minimizing the state error more strongly.

2. Decreasing the Elements of Q:

- Lower Weight on State Errors: Decreasing the diagonal elements of Q reduces the importance of minimizing deviations in the corresponding state variables. The controller will be less aggressive in correcting errors in these states.

- Less Aggressive Control: Lower values in Q lead to less aggressive control actions, potentially allowing for smoother but slower convergence to the desired state.

3. Different Weights for Different States:

- Balancing Priorities: By adjusting individual elements of Q , you can prioritize certain aspects of the system's behaviour. For example, you might set a higher weight on the angular position error than on the angular velocity error if reaching a specific position is more critical than minimizing velocity.

4. Effect on System Stability and Performance:

- Stability Considerations: While Q can be used to fine-tune the system's response, it is important to ensure that the chosen weights do not lead to instability or undesired oscillatory behavior. Typically, a balance between Q and R is needed to ensure both effective control and efficient use of control effort.

In summary, the Q matrix in the LQR design allows you to shape the desired behavior of the control system by specifying how much to penalize deviations in the state variables. Adjusting Q changes the trade-off between state error minimization and control effort, impacting the system's responsiveness, stability, and overall performance.

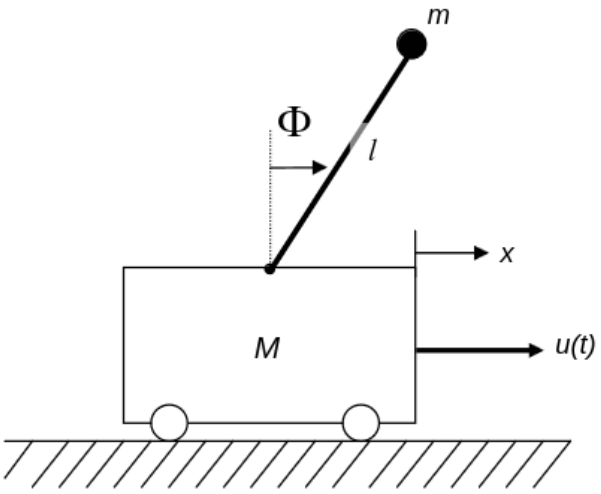
IMPLEMENTATION OF LQR IN INVERTED PENDULUM

- Linear Quadratic Regulator - as the name suggests minimises the quadratic cost function for a linear system. The key idea is to balance the trade-off between state regulation and control effort.
- $\dot{x}=Ax+Bu$ is a state space equation of a system. Which explains the dynamics of a system
- The objective of LQR is to find the control law $u=-Kx$ that minimises the cost function

$$J = \int_0^\infty \left(x^T Q x + u^T R u + 2x^T N u\right) dt$$

- Q: Penalises deviations from the desired state. R: Penalises the control effort
- K is mathematically derived from algebraic Riccati equation - $A^T P + PA - PBR^{-1}B^T P + Q = 0$
- But matlab provides a single command which gives gain matrix K

Implementation in inverse pendulum in MATLAB-

- 

1. Write state space equations

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{-mgx_3 - u}{M} = \frac{-mg}{M}x_3 + \frac{u}{M}$$

$$\dot{x}_3 = \dot{x}_4$$

$$\dot{x}_4 = \frac{Mgx_3 + mgx_3 - u}{Ml} = \frac{(M+m)g}{Ml}x_3 - \frac{u}{Ml}$$

2. Find A and B matrix

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-mg}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{(M+m)g}{Ml} & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{-1}{Ml} \end{bmatrix} u$$

3. Define Q and R

4. K = lqr(A, B, Q, R); # Get K

5. odeFunc = @(t, y) setSystem(y, -K * (y - y_final)); #Solve ODE