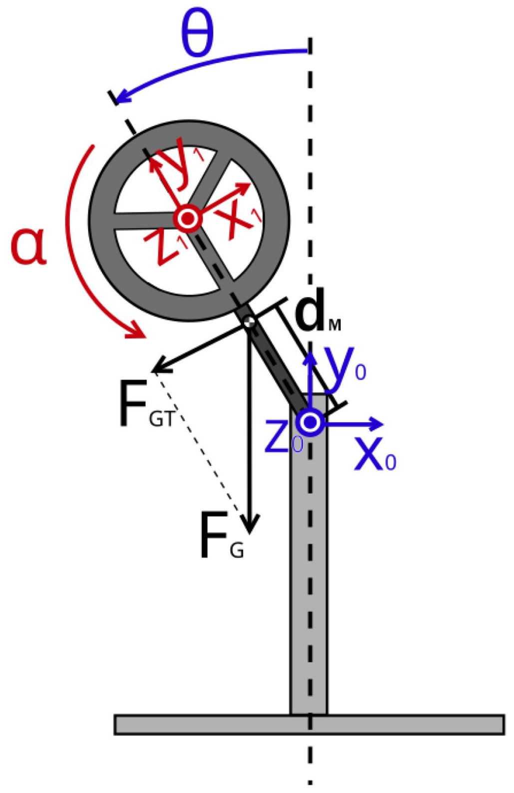
Linear Quadratic Regulator

IITB SSP GNC Subsystem Mini-Project

Aim of the Project

To model an inverted pendulum controlled by a single reaction wheel and control it using an LQR algorithm.

For given parameters of the system and required final orientation, I had to apply lqr to figure out the input that had to be provided to actuate the reaction wheel.



Intro to Project

The Linear Quadratic Regulator (LQR) is a well-known method that provides optimally controlled feedback gains to enable the closed-loop stable and high performance design of systems.

It is a control algorithm that can be used to provide the input to actuate the reaction wheel on a satellite. This will enable the satellite to control its attitude and hence perform its tasks with the proper orientation. LQR is a very powerful algorithm which combined with the Kalman Filter to estimate the state space can have wide-ranging applications in control.

Timeline

Timeline

Week 1	Week 2	Week 3
Understanding Eigenvalues and eigenvectors, Laplace transforms, transfer functions, Linearization, State-Space Representation, first-order systems, second-order systems, poles, zeros, system response, Routh-Hurwitz criterion	Understanding controllability and reachability, bang-bang principle, a brief overview of linear ODEs, Riccati equations, learning MATLAB and Simulink	Modelling the inverted pendulum system on MATLAB/Simulink and demonstrating its optimal control to the desired state.

Let’s Get into the thick of it

Classical Control Theory

Classical control theory is a branch of control theory that deals with the behavior of dynamical systems with inputs, and how their behavior is modified by feedback, using the Laplace transform as a basic tool to model such systems.

To overcome the limitations of the open-loop controller, classical control theory introduces feedback. A closed-loop controller uses feedback to control states or outputs of a dynamical system.

Classical control theory uses the Laplace transform to model the systems and signals. The Laplace transform is a frequency-domain approach for continuous time signals irrespective of whether the system is stable or unstable.

$$F(s) = \int_0^\infty e^{-st} f(t)\, dt$$

Optimal Control Theory

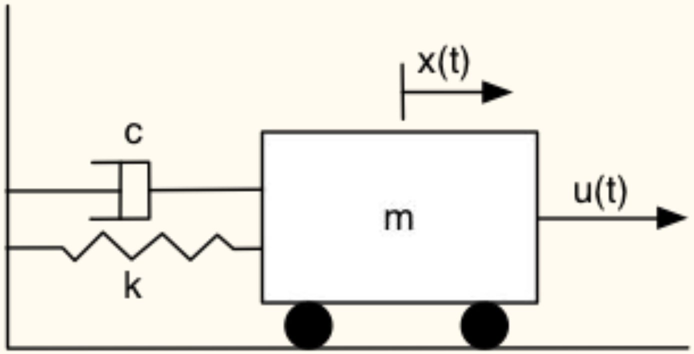
Optimal control theory is a branch of control theory that deals with finding a control for a dynamical system over a period of time such that an objective function is optimized.

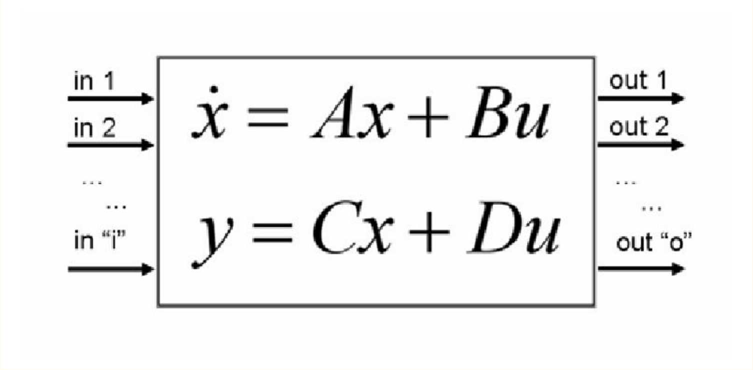
Optimal control deals with the problem of finding a control law for a given system such that a certain optimality criterion is achieved. A control problem includes a cost functional that is a function of state and control variables. An optimal control is a set of differential equations describing the paths of the control variables that minimize the cost function.

State Space Representation

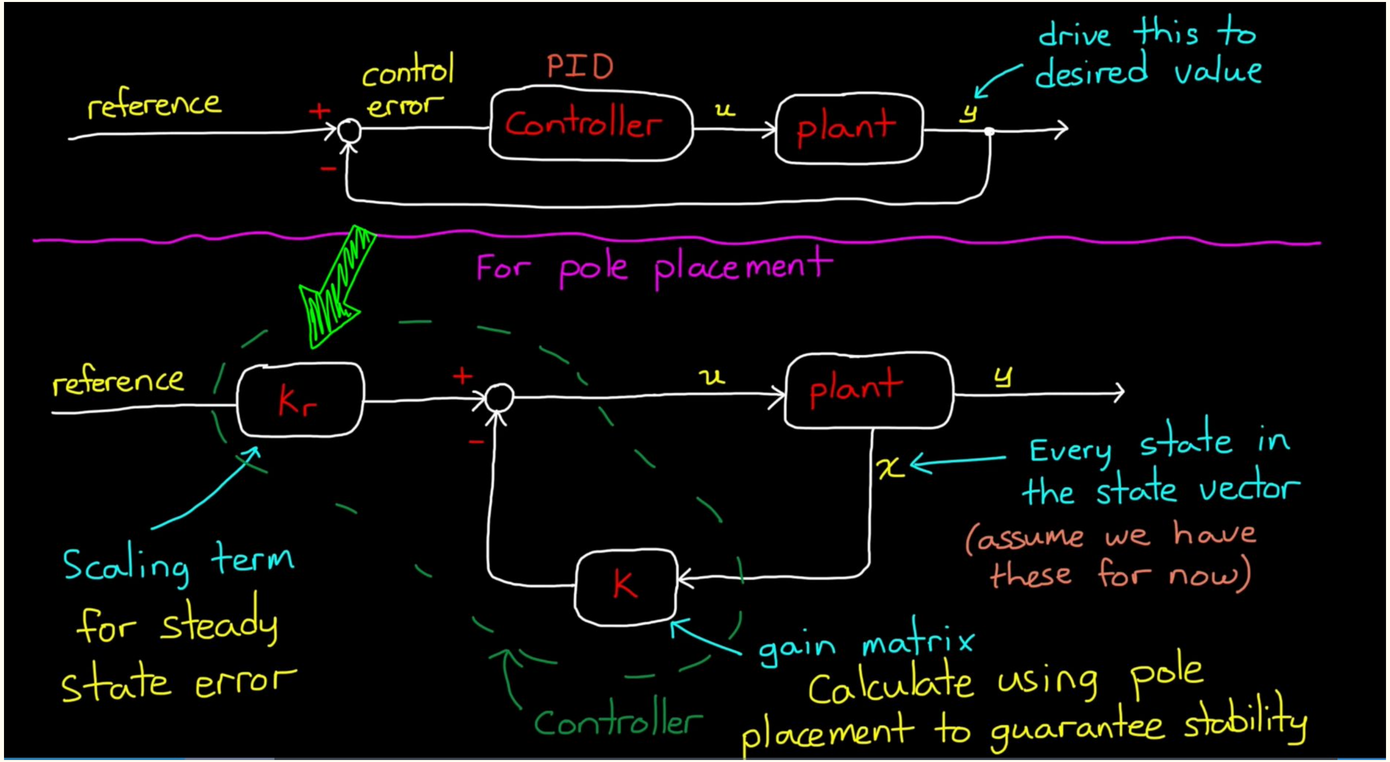
A state-space representation is a mathematical model of a physical system specified as a set of input, output, and variables related by first-order differential equations or difference equations.

The least number of convenient variables that can entirely describe the dynamics of the system at any given moment describe the state space of the system.





Full State Feedback



Controllability

Is it controllable? Is it observable?

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

It's controllable if...

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

$$\text{rank}(\mathcal{C}) = n$$

How controllable? Look at eigenvectors/eigenvalues of

$$W_c = \int_0^\infty e^{A\tau} BB^T e^{A^T\tau} \, d\tau$$

It's observable if...

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}, \quad \text{rank}(\mathcal{O}) = n$$

How observable? Look at eigenvectors/eigenvalues of

$$W_o = \int_0^\infty e^{A^T\tau} C^T C e^{A\tau} \, d\tau$$

Matrices

A, B, C, D, Q, R, K, S

```
A = [0 1 0 0;
    a/Inet -b/Inet 0 0;
    0 0 0 1;
    0 0 0 0];

B = [0 -1/Inet 0 1/Ir]';

C = eye(4);

D = 0;
```

```
% Q and R matrices

Q = [0.5 0 0 0;
    0 5 0 0;
    0 0 0.1 0;
    0 0 0 0.5];

R = 0.001;
```

Closed Loop Control Law

The closed loop control law takes the form u = -Kx where K is obtained from solving the algebraic Riccati Equation

Infinite-horizon, continuous-time [edit]

For a continuous-time linear system described by:

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

with a cost function defined as:

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

the feedback control law that minimizes the value of the cost is:

$$u = -Kx$$

where K is given by:

$$K = R^{-1}(B^T P + N^T)$$

and P is found by solving the continuous time algebraic Riccati equation:

$$A^T P + PA - (PB + N)R^{-1}(B^T P + N^T) + Q = 0$$

This can be also written as:

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

with

$$\mathcal{A} = A - BR^{-1}N^T \qquad \mathcal{Q} = Q - NR^{-1}N^T$$

Cost function for LQR

The general form of the cost function for LQR is given by:

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

Where:

- J is the total cost or performance index.
- $x(t)$ represents the state vector of the system.
- $u(t)$ is the control input.
- Q is the state weighting matrix (positive semi-definite).
- R is the control weighting matrix (positive definite).

LIGO

Other LQR Cost Function Forms

Other forms you may see for the cost function

- 'Infinite time horizon'. The simplest, and most common form. The one we use here. Ref [1].

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

- 'Finite time horizon'. More general form. Used in the most rigorous LQR derivations. Results in a time variant feedback law, i.e. *K = K(t)*. Ref [2].

$$J = \int_0^{t_f} \left(x^T Q x + u^T R u\right) dt + x^T(t_f) P x(t_f)$$

- You may also see this form, which is used by Matlab and also frequency domain versions of LQR.

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

In all cases, the weighting matrices are assumed to have these properties

$$Q = Q^T \quad 0, \; R = R^T > 0, \; P = P^T > 0$$

Riccati Equations

An **algebraic Riccati equation** is a type of nonlinear equation that arises in the context of infinite-horizon optimal control problems in continuous time or discrete time.

A typical algebraic Riccati equation is similar to one of the following:

the continuous time algebraic Riccati equation (CARE):

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

or the discrete time algebraic Riccati equation (DARE):

$$P = A^T PA - (A^T PB)(R + B^T PB)^{-1}(B^T PA) + Q.$$

P is the unknown n by n symmetric matrix and A, B, Q, R are known real coefficient matrices, with Q and R symmetric.

Though generally this equation can have many solutions, it is usually specified that we want to obtain the unique stabilizing solution, if such a solution exists.

Usually solvers try to find the unique stabilizing solution, if such a solution exists. A solution is stabilizing if using it for controlling the associated LQR system makes the closed loop system stable.

For the CARE, the control is

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

and the closed loop state transfer matrix is

$$A - BK = A - BR^{-1} B^T P$$

which is stable if and only if all of its eigenvalues have strictly negative real part.

For the DARE, the control is

$$K = (R + B^T PB)^{-1} B^T PA$$

and the closed loop state transfer matrix is

$$A - BK = A - B(R + B^T PB)^{-1} B^T PA$$

which is stable if and only if all of its eigenvalues are strictly inside the unit circle of the complex plane.

Demonstration

Editor - C:\Users\amrut\Documents\MATLAB\gvv.m

```
close all
% Parameters
Mm = 0.3; % Mass of motor
Mr = 0.7; % Mass of reaction wheel
Mb = 0.5; % Mass of pendulum
R = 0.25; % Radius of reaction wheel
lr = 0.5; % Distance of centre of reaction wheel from fixed base point, O
lb = 0.25; % Distance of centre of mass of pendulum from fixed base point, O

Imo = Mm*lr^2; % Moment of inertia of motor wrt O
Ir = Mr*R^2; % Moment of inertia of reaction wheel wrt its centre
Iro = Ir+ Mr*lr^2; % Moment of inertia of reaction wheel wrt O
Ibo = (Mb*lr^2)/3; % Moment of inertia of pendulum wrt O
Inet = Ibo + Iro + Imo; % Net moment of inertia of system wrt O

b = 0.2; % coefficient of viscous friction axle of pendulum
g = 9.8; % acceleration due to gravity

% initial points
theta_0 = 0;
theta_R_dot_0 = 0;

% set point

theta_req = 0.25;

% define the desired state
xd=[0.4;0;0.3;0];
```

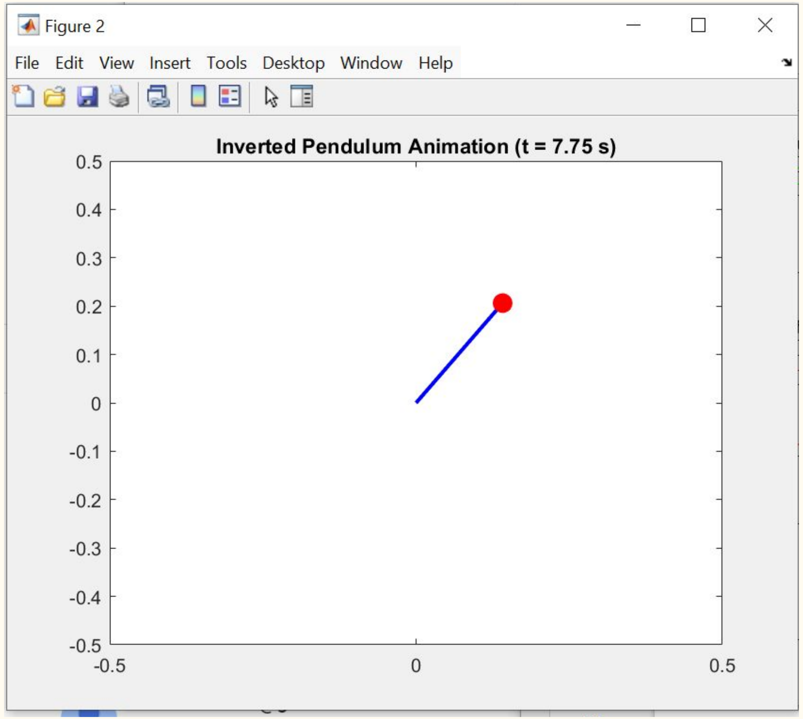


Figure 2
File Edit View Insert Tools Desktop Window Help
Inverted Pendulum Animation (t = 7.75 s)
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.5
0
0.5

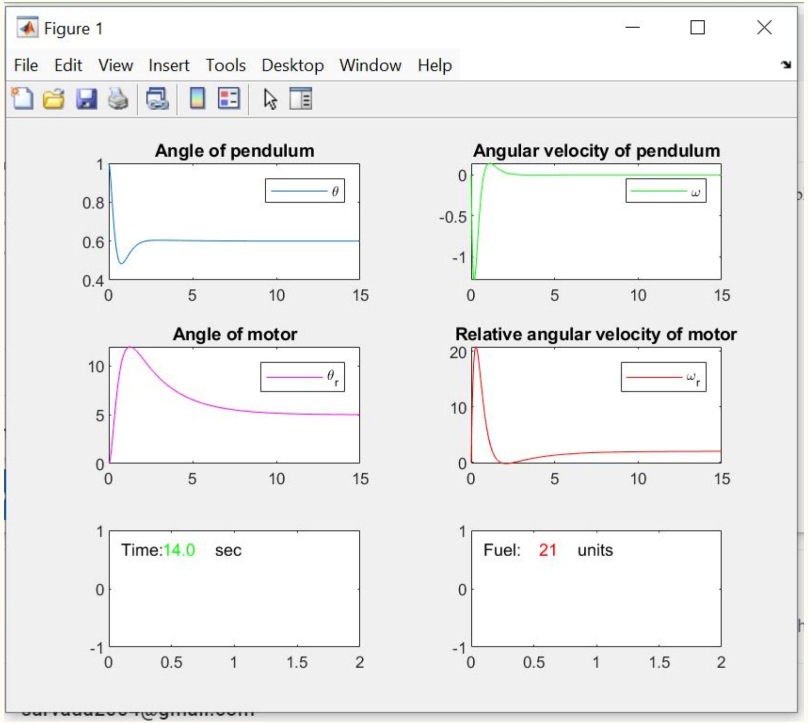


Figure 1
File Edit View Insert Tools Desktop Window Help
Angle of pendulum
θ
Angular velocity of pendulum
ω
Angle of motor
θr
Relative angular velocity of motor
ωr
Time:14.0 sec
Fuel: 21 units

Conclusion


MY WORK HERE
IS DONE
memegenerator.net