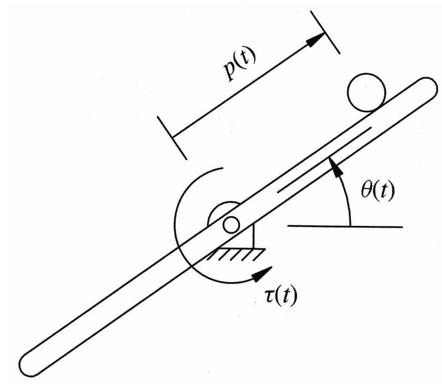
The project described below is an optional assignment that can be used to supplement your final course grade. Although completion of the project is optional, I hope that you will learn something through the process of completing it, and so encourage you to work on it even if you do not plan to turn it in for credit.

Complete each of the items listed below, then summarize your results and analysis in a report. The report is to be typed in 12-pt font single-spaced and should be *no longer than 3 pages*, not counting figures and Matlab scripts. Your score will be determined based on the completeness of this report as well as on the quality of your controller design and analysis. The final report will be due at the beginning of the final exam on Monday, December 18.

The system to be studied in this project is a ball rolling along a slotted beam, as shown. The goal will be to control the position of the ball on the slotted rotating beam by applying a torque $\tau(t)$ to the beam.



The equations of motion for this system are given by,

$$\begin{aligned} \left[\frac{J_b}{r^2} + m\right]\ddot{p}(t) + mg\sin\theta(t) - mp(t)\dot{\theta}(t)^2 &= 0 \\ \left[mp(t)^2 + J + J_b\right]\ddot{\theta}(t) + 2mp(t)\dot{p}(t)\dot{\theta}(t) + mgp(t)\cos\theta(t) &= \tau(t) \end{aligned}$$

in which $p(t)$ is the ball position, $\theta(t)$ is the beam angle, and $\tau(t)$ is the applied torque. In addition, g is the gravitational acceleration constant, J is the mass moment of intertia of the beam, and m, r, and J_b are the mass, radius, and mass moment of inertia of the ball, respectively.

1. Express the nonlinear equations of motion in state-space form with input $u(t) = \tau(t)$, output $y(t) = p(t)$, and the state variables $x_1(t) = p(t)$, $x_2(t) = \dot{p}(t)$, $x_3(t) = \theta(t)$, and $x_4(t) = \dot{\theta}(t)$.
2. Linearize the state-space system realization in (1) about the nominal trajectory,

$$\begin{aligned} x_1^o(t) &= v_o(t - t_o) + p_o \\ x_2^o(t) &= v_o \\ x_3^o(t) &= 0 \\ x_4^o(t) &= 0 \\ u^o(t) &= mgx_1^o(t), \end{aligned}$$

which corresponds to a steady level beam and a ball with constant velocity v_o and initial position p_o at initial time t_o.

For the remainder of the project, consider a special case of the nominal trajectory with $v_o = 0$ and the following parameter values:

Parameter	**Value**	**Units**	**Name**
L	1	m	Beam length (rotates about center)
J	0.0676	kg-m^2	Beam mass moment of inertia
m	0.9048	kg	Ball mass
r	0.03	m	Ball radius
J_b	0.000326	kg-m^2	Ball mass moment of inertia
g	9.81	m/s^2	Gravitational acceleration

3. Simulate and plot the open-loop state variables in response to an impulse torque input $\tau(t) = \delta(t)$-Nm and $p(0) = p_o = 0.25$-m with zero initial conditions on all other state variables. Simulate long enough to demonstrate the steady-state behavior. What are the system eigenvalues? Based on these eigenvalues and the physical system, explain the system response to these initial conditions.

4. Assess the observability, controllability, and minimality of the linearized system realization. Justify your responses.

5. Based on your knowledge of the ball-beam system and the response of linear systems, determine a "good" set of closed-loop eigenvalues that should be targeted to achieve a desirable closed-loop response. Explain the various factors that led you to your final decision.

6. Based on your choice of desired closed-loop eigenvalues in (5), design a full-state feedback control law to achieve the desired response. Evaluate your design: Simulate and plot the closed-loop response to an impulsive input $\tau(t) = \delta(t)$ and zero initial conditions. Plot this along with the open-loop response in a single figure. If the closed-loop response is not as desired, iterate on the design and update your response to (5).

7. Design an observer-based feedback controller based on the full-state feedback control law designed in (6). Set the observer eigenvalues to be 10 times the full-state feedback eigenvalues. Evaluate your design: Simulate and plot the closed-loop output response to an impulsive input $\tau(t) = \delta(t)$. Plot this along with the open-loop and full-state feedback responses in a single figure. Initialize the observer such that the error is non-zero; otherwise, the response will be identical to the full-state feedback controller. Report your initialization.

8. In the final part of the project, you will design a full-state feedback controller and an observer-based feedback controller using linear quadratic optimal control/estimation techniques. Specifically, use the command `lqr` to synthesize an optimal control gain for use in full-state feedback. Further, synthesize an optimal observer using the command `lqe`, and combine with the optimal full-state feedback controller to yield an observer-based feedback controller. In a single figure, compare these system responses with the responses in (6) and (7). In a separate figure, plot the input versus time for the optimal control laws and the full-state and observer-based feedback laws in (6) and (7). "Tune" both the controller and the observer weighting matrices to achieve a desirable output response. Explain how you came to your final design, including any trades you made during the process (e.g., quick response versus low actuator effort).

Position Control of the Ball on Slotted Rotating Beam

Jyot Buch

December 16, 2017

Abstract

In this project, we consider the ball and beam system, in which the beam is supported in the center by the fulcrum. The objective is to analyze the system from the controls viewpoint and design a observer-based full state feedback controller using pole placement techniques. This work also compares the ordinary pole placement observer based state feedback controller with the Linear Quadratic Gaussian (LQG) control law, which is a combination of the Linear Quadratic Regulator (LQR) optimal control and Kalman filter based optimal observer/estimator. The results and analysis are summarized in this report.

1 Introduction

Consider the ball and the beam system as shown in the figure 1. The goal of this project is to investigate the control policy to manipulate the position of the ball on the slotted rotating beam by applying a torque $\tau(t)$ to the beam. Here, $p(t)$ represents ball position, $\theta(t)$ is the

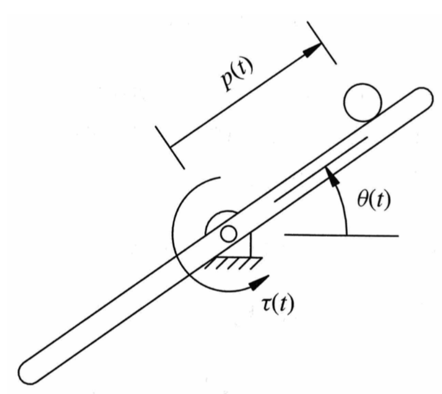


Figure 1: Ball and Beam System

beam angle. This system is inherently exhibits a nonlinear behavior. In what follows, we will first look at the nonlinear dynamics of this system in section 2. Linearized dynamics at given nominal trajectory are represented in section 3. Further assessment on linearized system eigenvalues, controllability, observability, minimality has been done in section 4. MATLAB was used to design a observer based full state feedback controller.

2 Nonlinear System Dynamics

2.1 Equations of Motion

Dynamics of the system shown in the figure 1 can be derived from the Euler-Lagrange equations of motion.

$$\begin{aligned} \left[\frac{J_b}{r^2} + m\right]\ddot{p}(t) + mgsin(\theta(t)) - mp(t)\dot{\theta}(t)^2 = 0 \\ [mp(t)^2 + J + J_b]\ddot{\theta}(t) + 2mp(t)\dot{p}(t)\dot{\theta}(t) + mgp(t)cos(\theta(t)) = \tau(t) \end{aligned} \tag{1}$$

Where, g is the gravitational acceleration constant, J is the mass moment of inertia of the beam, m, r, and J_b are the mass, radius, and mass moment of inertia of the ball, respectively.

2.2 Nonlinear State Space

Above dynamical equations can also be written in general nonlinear state-space as shown in the equation 4.

$$\begin{aligned} \dot{x} &= f(x(t), u(t), t) \\ y &= g(x(t), u(t), t) \end{aligned} \tag{2}$$

The input is given by $u(t) = \tau(t)$ and output is $y(t) = p(t)$. Where, the state vector is defined by,

$$\underline{x} = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} = \begin{bmatrix} p(t) \\ \dot{p}(t) \\ \theta(t) \\ \dot{\theta}(t) \end{bmatrix} \tag{3}$$

$$\begin{aligned} \underline{\dot{x}} = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4 t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ \left(mx_1(t)x_3^2(t) - mgsinx_3(t)\right)\frac{r^2}{J_b+mr^2} \\ x_4(t) \\ \left(\tau(t) - 2mx_1(t)x_2(t)x_4(t) - mgx_1(t)cosx_3(t)\right)\frac{1}{mx_1^2+J+J_b} \end{bmatrix} = f(x,u) \\ y(t) = x_1(t) = g(x,u) \end{aligned} \tag{4}$$

3 Linearizion

Linearizing the dynamical equation 4, about the following nominal trajectory, which corresponds to a steady level beam and a ball with constant velocity v_o and initial position p_o at initial time t_o.

$$
\begin{aligned}
x_1^o(t) &= v_o(t - t_o) + p_o \\
x_2^o(t) &= v_o \\
x_3^o(t) &= 0 \\
x_4^o(t) &= 0 \\
u^o(t) &= mgx_1^o(t)
\end{aligned} \tag{5}
$$

In a symbolic form, we get

$$
\begin{aligned}
\Delta\underline{\dot{x}}(t) &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-mgr^2}{J_b+mr^2} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-mg}{mx_1^2+J+J_b} & 0 & 0 & \frac{-2mx_1v_0}{mx_1^2+J+J_b} \end{bmatrix} \Delta\underline{x} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{mx_1^2+J+J_b} \end{bmatrix} \Delta u \\
\Delta y &= \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \Delta\underline{x}
\end{aligned} \tag{6}
$$

Parameter	Value	Units	Name
L	1	m	Beam length (rotates about center)
J	0.0676	kg-m^2	Beam mass moment of inertia
m	0.9048	kg	Ball mass
r	0.03	m	Ball radius
J_b	0.000326	kg-m^2	Ball mass moment of inertia
g	9.81	m/s^2	Gravitational acceleration

Figure 2: System Parameters

Using the given parameters in figure 2, $p_o = 0.25$ and $v_o = 0$ we get the linearized system as shown in the MATLAB script.

4 Analysis

4.1 Eigenvalues

Looking at the eigenvalues, we find that ball-beam system has 2 purely real eigenvalues and 2 purely imaginary eigenvalues. One of the real eigen value is positive, i.e. system is unstable. Moreover, due to imaginary eigenvalues, the system exhibits undamped oscillatory modes. Simulation result shows that system response blows up.

$$
\underline{\lambda} = \begin{bmatrix} -4.7276 \\ 4.7276 \\ -4.7276i \\ +4.7276i \end{bmatrix} \tag{7}
$$

From the dynamical system, since we have L = 1m, the ball will escape the beam in order to react towards impulse input. Due to physical length L constraint, the system will ultimately be categorized as unstable system, since now ball does not exist in the system boundary.

4.2 Impulse Response

Since, the system is unstable, we get an unbounded increase of output in impulse response. Please refer to the attached simulation results for more details.

4.3 Observability

Observability matrix for this system is given by,

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -7.005 & 0 \\ 0 & 0 & 0 & -7.005 \end{bmatrix} \tag{8}$$

We can clearly see that observability matrix is full rank, thus, given system is full state observable.

4.4 Controllability

Controllability matrix for this system is given by,

$$\mathcal{C} = \begin{bmatrix} B & AB & A^2B & A^3B \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -56.28 \\ 0 & 0 & -56.28 & 0 \\ 0 & 8.03 & 0 & 0 \\ 8.03 & 0 & 0 & 0 \end{bmatrix} \tag{9}$$

We can clearly see that controllability matrix is full rank, thus, given system is full state controllable.

4.5 Minimality

Since, given system is both controllable and observable, it is in the minimal realization form (verified from MATLAB Command `minreal`) In other words, adding a new state to this system does not change the rank of the system matrix. Also, we either loose the controllability or observability of the system.

5 Control Design

5.1 Full State Feedback Controller

Since, no explicit control behavior was specified, in order to design a state feedback controller using pole placement technique, let's consider following requirement.

- The closed loop pole or eigenvalues of the controlled system needs to be on left hand side of the s-plane, i.e. $Re(\lambda) < 0$
- The controlled system should not violate the constraints for the beam length, i.e. the impulse response output should not overshoot 1 m length.

In order to realize the above requirement for the controller, we place the closed loop poles at λ_{des} = [-5, -4, -3, -2] location. The full order state feedback control law can be written as,

$$
\begin{aligned}
u &= r - K\underline{x} \\
\dot{\underline{x}} &= A\underline{x} + Bu = (A - BK)\underline{x} + Br \\
y &= C\underline{x}
\end{aligned}
\tag{10}
$$

An iterative design approach was performed to obtain the overshoot less then 1.

5.2 Observer-based full state feedback controller

Observer-based feedback controller was designed by placing the eigenvalues at 10 times of the controller eigenvalues, i.e. $\lambda_{observer}$ = [-50, -40, -30, -20], because it is desired that observer dynamics are 5-10 times faster than controller dynamics, i.e. we want the estimates to be computed faster enough so that the controller can timely propagate the state in the feedback. Since, we are feeding estimated states to the controller, the control law now becomes $u = r - K\hat{\underline{x}}$. The augmented dynamics can be written as,

$$
\begin{aligned}
\begin{bmatrix} \dot{\underline{x}} \\ \dot{\hat{\underline{x}}} \end{bmatrix} &= \begin{bmatrix} A & -BK \\ GC & A - GC - BK \end{bmatrix} \begin{bmatrix} \underline{x} \\ \hat{\underline{x}} \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} r \\
y &= \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} \underline{x} \\ \hat{\underline{x}} \end{bmatrix}
\end{aligned}
\tag{11}
$$

5.3 LQG Control

Linear Quadratic Control(LQR) combined with Kalman Filter (Optimal State Observer) known as Linear Quadratic Gaussian controller. From, separation principle we can design a optimal observer and optimal controller separately and incorporate together in design.

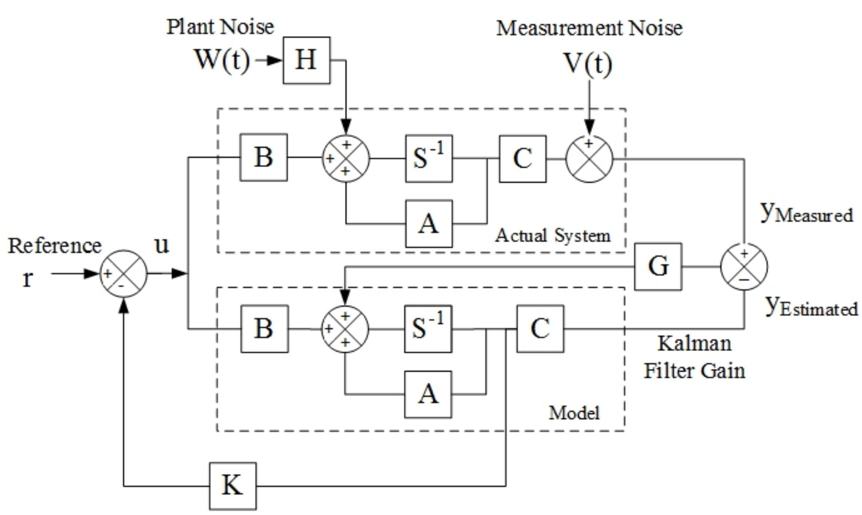


Figure 3: LQG Controller

Using Iterative approach, to tune the controller, Q was chosen to be diag([100 10 10 10]) to penalize the states and R was chosen to be 10 as an input cost. First element of Q was selected to be 100 which is higher than other elements in the diagonal matrix, to emphasis on position control more. For Kalman filter tunning, since we are not adding any noise to the system, the plant uncertainties and measurement uncertainties can be assumed to be low. For simplicity $Q_{est} = 0.01$*eye(4), $R_{est} = 0.01$.

5.4 Comparison of Results & Discussions

- The initial conditions for actual states and estimated states were off, so as to replicate the initial condition errors.

$$\begin{aligned} \underline{x}_0 &= \begin{bmatrix} 0.25 & 0 & 0 & 0 \end{bmatrix}^T \\ \underline{\hat{x}}_0 &= \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \end{bmatrix}^T \end{aligned} \tag{12}$$

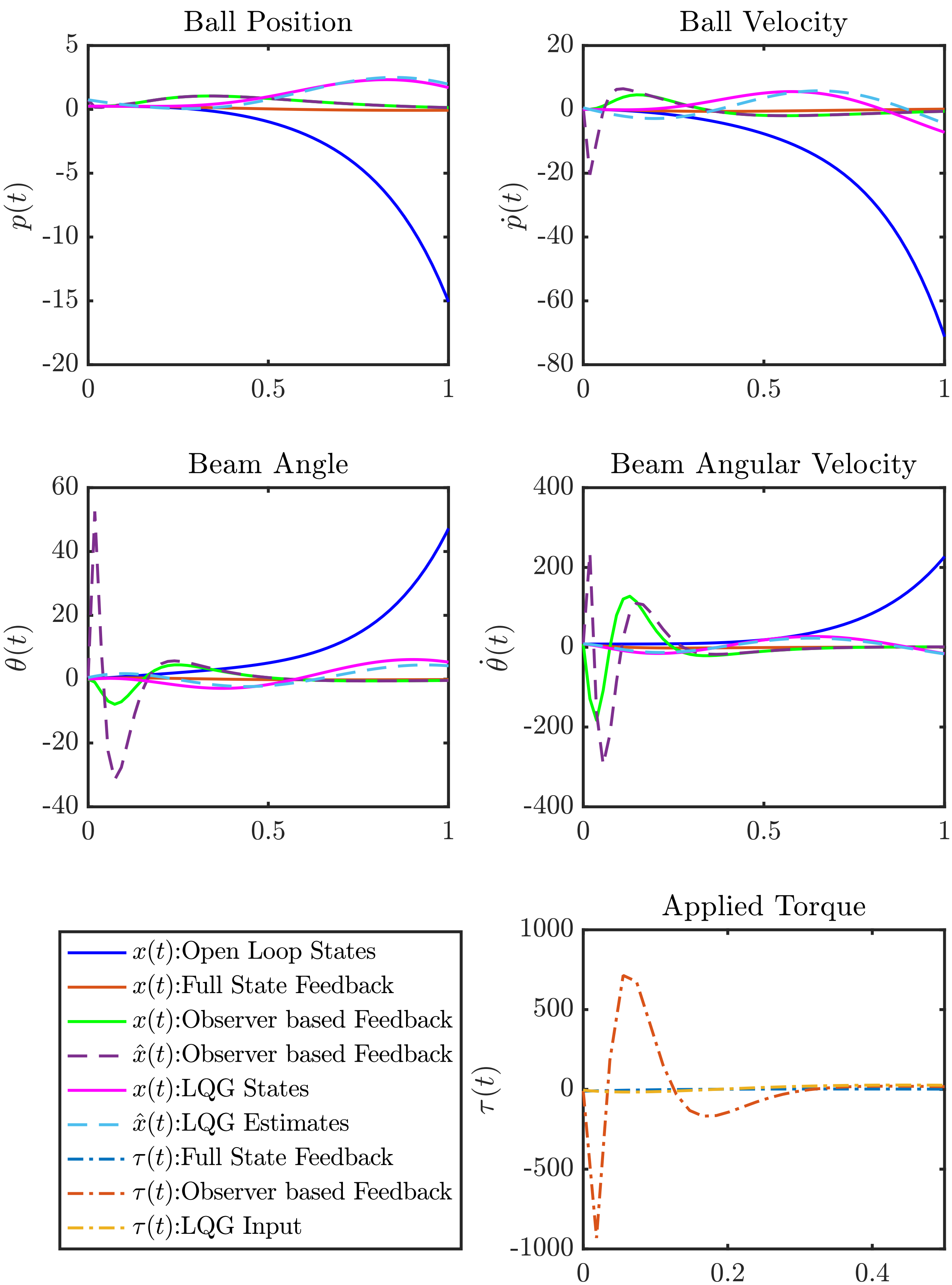
  It was observed that initially the estimation error $\underline{x} - \underline{\hat{x}}$ is high, but as time progresses it approaches to zero for both observer based feedback controller and LQG. Although it was observed that LQG response errors were slowly decreasing compared to observer based feedback, that is due to the tuning parameters that we choose in the design.

- Looking at the actuation effort response, it was found that Observer based feedback was having very high input effort. Since, LQR considered the input cost in the objective it minimizes the input cost. The main objective of the design in this project was to minimize the actuation effort and still have good tracking behavior.

- It is possible for the observer based feedback that if initial condition errors are higher than some threshold, the impulse response show that the ball may leave the contact with the beam of length L = 1 m. Thus, tuning was done such that ball stays in the boundary of the system.

- During the controller design a trade off was observed between actuation effort and quick response. Hence, if we desire quick response, the input energy required for it is huge, where as for minimum energy input we get sluggish tracking. As mentioned earlier, the objective of this project was to minimize the actuator effort and still get the better position tracking performance.

- Length of the beam impose constraint to the system behavior, i.e. constraint on the position of the ball, so that it can not exceed the length of the beam. But, LQG does not allow us to add constraints to the system parameters. Since, we know the model of the system, in this case an alternative approach like Model Predictive Control (MPC) can be considered to further improve the performance. In MPC, we predict the behavior of the system for finite time horizon and try to minimize the actuation effort over it, but we implement only few steps and recalculate the policy. One of the great advantage for MPC is, its ability to handle state constraints. Moreover, it is more robust to disturbance rejection.


Ball Position
Ball Velocity
Beam Angle
Beam Angular Velocity
Applied Torque
$p(t)$
$\dot{p}(t)$
$\theta(t)$
$\dot{\theta}(t)$
$\tau(t)$
$x(t)$:Open Loop States
$x(t)$:Full State Feedback
$x(t)$:Observer based Feedback
$\hat{x}(t)$:Observer based Feedback
$x(t)$:LQG States
$\hat{x}(t)$:LQG Estimates
$\tau(t)$:Full State Feedback
$\tau(t)$:Observer based Feedback
$\tau(t)$:LQG Input

6 Conclusion

- The comparison of LQG vs observer based feedback vs pure state feedback controller was presented in this project report. The study shows that the controller design depends on various factors and tuning parameters for the controller and observer. Having some statistics of the measurement noise and modeling uncertainties would greatly enhance the controller performance.
- Alternative control strategy like MPC, Reinforcement Learning approaches, robustness of the controller, effect of the measurement noise and disturbance rejection properties of the controller can be considered for future investigation.

7 MATLAB Code

Contents

Clear Workspace

```
clear;close all; clc;
```

Part-3: Linearized System Dynamics

```
L = 1;
J = 0.0676;
m = 0.9048;
r = 0.03;
Jb = 0.000326;
g = 9.81;

% Initial conditions
p0 = 0.25;
x1 = p0;
v0 = 0;
u0 = m*g*p0;
x0 = [x1;v0;0;0];

A = [0 1 0 0;
```

```
0 0 -m*g*r^2/(Jb + m*r^2) 0;
0 0 0 1;
-m*g/(m*x1^2+J+Jb) 0 0 -2*m*x1*v0/(m*x1^2+J+Jb)];
B = [0;0;0;1/(m*x1^2+J+Jb)];
C = [1 0 0 0];
D = 0;

% Eigenvalues
[V,Di]=jordan(A);
fprintf('Eigenvalues: \n\n');
disp(diag(Di));

% Impulse Response
tfinal = 10;
sysC = ss(A,B,C,D);
[~,t,x] = impulse(sysC,tfinal);
x(:,1) = x(:,1) + p0*ones(size(x(:,1)));
figure(1);

ylabelArray = {'$p(t)$','$\dot{p}(t)$','$\theta(t)$','$\dot{\theta}(t)$'};
titleArray = {'Ball Position','Ball Velocity','Beam Angle','Beam Angular Velocity'};
for i = 1:4
sp = subplot(3,2,i);
plt1 = plot(t,x(:,i),'LineStyle','-','Color','b');
set(gca,'XLim',[0 0.6]);
ylabel(ylabelArray{i});
title(titleArray{i});
end

% Legend Plot
sh=subplot(3,2,5);
p=get(sh,'position');
lh=legend(sh,plt1,'Open Loop');
set(lh,'position',p);
axis(sh,'off');
```

```
Eigenvalues:

-4.7276 + 0.0000i
4.7276 + 0.0000i
0.0000 - 4.7276i
0.0000 + 4.7276i
```

Part-4: System Analysis

```
sysObsv = obsv(A,C);
sysCtrb = ctrb(A,B);
fprintf('Rank of Observability Matrix: %d \n\n',rank(sysObsv));
fprintf('Rank of Controllability Matrix: %d \n\n',rank(sysCtrb));
fprintf('Since the system is both controllable and observable, it is minimal.\n\n');
```

```
Rank of Observability Matrix: 4

Rank of Controllability Matrix: 4

Since the system is both controllable and observable, it is minimal.
```

Part-5: State Feedback Controller Design

```
P = [-5 -4 -3 -2];
K = acker(A,B,P);
Aaug = A-B*K;
sysAug = ss(Aaug,B,C,D);
[~,t,x] = impulse(sysAug,tfinal);
x(:,1) = x(:,1) + p0*ones(size(x(:,1)));
figure(1);

% Verify Eigenvalues
fprintf('Eigenvalues of augmented system with full state feedback:\n\n');
disp(eig(Aaug));
```

```
Eigenvalues of augmented system
with full state feedback:

-5.0000
-4.0000
-3.0000
-2.0000
```

Part-6: State Feedback Controller Design Plot

```
for i = 1:4
subplot(3,2,i)
```

```
hold on;
plt2 = plot(t,x(:,i));
set(gca,'XLim',[0 0.6]);
ylabel(ylabelArray{i});
title(titleArray{i});
end
hold off;

% Actuation effort
subplot(3,2,6);
u = -K*x';
tau_plt1 = plot(t,u,'LineStyle','-.');
title('Applied Torque');ylabel('$\tau(t)$');

% Legend Plot
sh=subplot(3,2,5);
p=get(sh,'position');
lh=legend(sh,[plt1 plt2 tau_plt1],'Open Loop','Closed Loop',...
'$\tau(t)$ Full State Feedback');
set(lh,'position',p);
axis(sh,'off');
```

Part-7: Observer based Feedback Control Design

```
% Observer poles 10 times the controller poles
O = 10*P;
G = place(A',C',O)';
Aaug = [...
A -B*K;
G*C A-G*C-B*K
];
Baug = [B;B];
Caug = [C zeros(size(C))];
x0_observer = 0.5*ones(4,1);

% Verify Eigenvalues
fprintf('### Eigenvalues of augmented system controller dynamics with observer based fee
disp(eig(A-B*K));
fprintf('### Eigenvalues of augmented system controller dynamics with observer based fee
disp(eig(A-G*C));

sysAug = ss(Aaug,Baug,Caug,D);
x0Aug = [0;0;0;0;x0_observer];
[~,timp,ximp] = impulse(sysAug,tfinal);
[~,tinit,xinit] = initial(sysAug,x0Aug,timp);
```

```
xadd = ximp + xinit;
xadd(:,1) = xadd(:,1) + p0*ones(size(xadd(:,1)));
xadd(:,5) = xadd(:,5) + p0*ones(size(xadd(:,1)));
figure(1);
for i = 1:4
subplot(3,2,i)
hold on;
plt3 = plot(tinit,xadd(:,i),'Color','g');
plt4 = plot(tinit,xadd(:,4+i),'--');
set(gca,'XLim',[0 0.6]);
ylabel(ylabelArray{i});
title(titleArray{i});
end
hold off;

% Actuation effort
subplot(3,2,6);
hold on;
u = -K*xadd(:,5:8)';
tau_plt2 = plot(t,u,'LineStyle','-.');

% Legend Plot
sh=subplot(3,2,5);
p=get(sh,'position');
lh=legend(sh,[plt1 plt2 plt3 plt4 tau_plt1 tau_plt2],...
'$x(t)$:Open Loop States',...
'$x(t)$:Full State Feedback','$x(t)$:Observer based Feedback',...
'$\hat{x}(t)$:Observer based Feedback','$\tau(t)$:Full State Feedback',...
'$\tau(t)$:Observer based Feedback');
set(lh,'position',p);
axis(sh,'off');


### Eigenvalues of augmented system
controller dynamics with observer based feedback:

-5.0000
-4.0000
-3.0000
-2.0000

### Eigenvalues of augmented system controller dynamics
with observer based feedback:

-50.0000
```

```
-40.0000
-30.0000
-20.0000
```

Part-8: Optimal State Feedback Controller Design

```
Q = diag([100 10 10 10]);
R = 10;
[K,S,e] = lqr(sysC,Q,R);

Qest = 0.01*diag([1 1 1 1]);
Rest = 0.01;
G = eye(4);
[LGain,P,E] = lqe(A,G,C,Qest,Rest);

Aaug = [...
A -B*K;
LGain*C A-LGain*C-B*K
];
Baug = [B;B];
Caug = [C zeros(size(C))];
sys_aug = ss(Aaug,Baug,Caug,D);
x0_observer = 0.5*ones(4,1);

% Verify Eigenvalues
fprintf('*** Eigenvalues of augmented system controller dynamics with LQG:\n\n');
disp(eig(A-B*K));
fprintf('*** Eigenvalues of augmented system observer dynamics with LQG:\n\n');
disp(eig(A'-G'*C'));

sysAug = ss(Aaug,Baug,Caug,D);
x0Aug = [0;0;0;0;x0_observer];
[yimp,timp,ximp] = impulse(sysAug,tfinal);
[yinit,tinit,xinit] = initial(sysAug,x0Aug,timp);
xadd = ximp + xinit;
xadd(:,1) = xadd(:,1) + p0*ones(size(xadd(:,1)));
xadd(:,5) = xadd(:,5) + p0*ones(size(xadd(:,1)));
figure(1);
for i = 1:4
subplot(3,2,i)
hold on;
plt5 = plot(tinit,xadd(:,i),'Color','m');
plt6 = plot(tinit,xadd(:,4+i),'--');
```

```
set(gca,'XLim',[0 1]);
ylabel(ylabelArray{i});
title(titleArray{i});
end
hold off;

% Actuation effort
subplot(3,2,6);
hold on;
u = -K*xadd(:,5:8)';
tau_plt3 = plot(tinit,u,'LineStyle','-.');
set(gca,'XLim',[0 0.5]);

% Legend Plot
sh=subplot(3,2,5);
p=get(sh,'position');
lh=legend(sh,[plt1 plt2 plt3 plt4 plt5 plt6 tau_plt1 tau_plt2 tau_plt3],...
'$x(t)$:Open Loop States',...
'$x(t)$:Full State Feedback','$x(t)$:Observer based Feedback',...
'$\hat{x}(t)$:Observer based Feedback','$x(t)$:LQG States',...
'$\hat{x}(t)$:LQG Estimates','$\tau(t)$:Full State Feedback',...
'$\tau(t)$:Observer based Feedback','$\tau(t)$:LQG Input');
set(lh,'position',p);
axis(sh,'off');
print('-fillpage','Report/fig1','-dpdf')
```

```
*** Eigenvalues of augmented system controller dynamics with LQG:

-8.7549 + 0.0000i
-1.5774 + 3.8380i
-1.5774 - 3.8380i
-3.5179 + 0.0000i

*** Eigenvalues of augmented system observer dynamics with LQG:

-4.8742 + 0.0000i
-0.3321 + 4.7737i
-0.3321 - 4.7737i
4.5385 + 0.0000i
```

8 Attachments

- Manuscript for the linearizion of dynamics

(1) $x_1(t) = p(t)$ $\qquad u(t) = \tau(t)$

$x_2(t) = \dot{p}(t)$ $\qquad y(t) = p(t)$.

$x_3(t) = \theta(t)$

$x_4(t) = \dot{\theta}(t)$

↳ Nonlinear state space equations,

$$\left.\begin{aligned}\dot{\underline{x}}(t) &= f(\underline{x}(t), u(t)) \\ y(t) &= g(\underline{x}(t), u(t))\end{aligned}\right\} \text{standard form.}$$

↳ $\dot{x}_1(t) = \dot{p}(t) = x_2(t)$

$$\dot{x}_2(t) = \ddot{p}(t) = \left(m\underbrace{p(t)}_{x_1(t)}\underbrace{\dot{\theta}(t)}_{x_4(t)}^2 - mg\sin\underbrace{\theta(t)}_{x_3(t)}\right)\left(\frac{r^2}{J_b + mr^2}\right)$$

$\dot{x}_3(t) = \dot{\theta}(t) = x_4(t)$

$$\dot{x}_4(t) = \ddot{\theta}(t) = \left[\underbrace{\tau(t)}_{u(t)} - 2m\underbrace{p(t)}_{x_1(t)}\underbrace{\dot{p}(t)}_{x_2(t)}\underbrace{\dot{\theta}(t)}_{x_4(t)} - mg\underbrace{p(t)}_{x_1(t)}\cos\underbrace{\theta(t)}_{x_3(t)}\right]\left(\frac{1}{m\underbrace{p(t)}_{x_1(t)}^2 + J + J_b}\right)$$

↳ Standard nonlinear state space,

$$\dot{\underline{x}}(t) = \begin{bmatrix}\dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t)\end{bmatrix} = \begin{bmatrix} x_2 \\ (m x_1 x_4^2 - mg\sin x_3)\dfrac{r^2}{J_b + mr^2} \\ x_4 \\ (u - 2m x_1 x_2 x_4 - mg x_1 \cos x_3)\dfrac{1}{m x_1^2 + J + J_b}\end{bmatrix} = f(\underline{x}, u) = \begin{bmatrix} f_1(\underline{x}, u) \\ f_2(\underline{x}, u) \\ f_3(\underline{x}, u) \\ f_4(\underline{x}, u)\end{bmatrix}$$

NOTE: states are functions of time, for simplisity its written as x_1 instead of $x_1(t)$

$y(t) = x_1(t)$

(2) Linearization about the nominal trajectory.

$$A = \left[\frac{\partial f}{\partial \underline{x}}\right]_{\substack{x_0\\u_0}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \frac{\partial f_1}{\partial x_4} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \frac{\partial f_3}{\partial x_4} \\ \frac{\partial f_4}{\partial x_1} & \frac{\partial f_4}{\partial x_2} & \frac{\partial f_4}{\partial x_3} & \frac{\partial f_4}{\partial x_4} \end{bmatrix}_{(\underline{x}_0, \underline{u}_0)}$$

$$\frac{\partial f_1}{\partial x_2} = 1$$

$$x_1^0(t) = v_0(t-t_0) + p_0$$
$$x_2^0(t) = v_0$$
$$x_3^0(t) = 0$$
$$x_4^0(t) = 0$$
$$u^0(t) = mg\, x_1^0(t)$$

$$\left[\frac{\partial f_2}{\partial x_1}\right]_{(x_0,u_0)} = \frac{m x_4^2 r^2}{J_b + m r^2} = 0$$

$$\left[\frac{\partial f_2}{\partial x_3}\right]_{(x_0,u_0)} = \frac{-mg\cos x_3 \cdot r^2}{J_b + m r^2} = \frac{-mg r^2}{J_b + m r^2}$$

$$\left[\frac{\partial f_2}{\partial x_4}\right]_{(x_0,u_0)} = 2m x_1 x_4 \cdot \frac{r^2}{J_b + m r^2} = 0$$

$$\left[\frac{\partial f_3}{\partial x_4}\right]_{(x_0,u_0)} = 1$$

$$\left[\frac{\partial f_4}{\partial x_1}\right]_{(x_0,u_0)} = \frac{(m x_1^2 + J + J_b)(-\overset{\nearrow 0}{2m x_2 x_4} - mg\cos x_3) - (u - 2m x_1 \overset{\nearrow 0}{x_2 x_4} - mg x_1 \cos x_3)(2m x_1)}{(m x_1^2 + J + J_b)^2}$$

$$= \frac{(m x_1^2 + J + J_b)(-mg) - 2m x_1 \overset{\nearrow 0}{(u - mg x_1)}}{(m x_1^2 + J + J_b)^2} = \frac{-mg}{m x_1^2 + J + J_b}$$

$$\left[\frac{\partial f_4}{\partial x_2}\right]_{(x_0,u_0)} = \frac{-2m x_1 x_4}{m x_1^2 + J + J_b} = 0$$

$$\left[\frac{\partial f_4}{\partial x_3}\right]_{(x_0,u_0)} = \frac{-m g x_1 \sin x_3}{m x_1^2 + J + J_b} = 0$$

$$\left[\frac{\partial f_4}{\partial x_4}\right]_{(x_0,u_0)} = \frac{-2m x_1 x_2}{m x_1^2 + J + J_b} = \frac{-2m\left[v_0^2(t-t_0) + P_0 v_0\right]}{m\left(v_0(t-t_0)+P_0\right)^2 + J + J_b}$$

$$B = \left[\frac{\partial f}{\partial u}\right]_{u_0} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dfrac{1}{m\left(v_0(t-t_0)+P_0\right)^2 + J + J_b} \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \dfrac{-mgr^2}{J_b + mr^2} & 0 \\ 0 & 0 & 0 & 1 \\ \dfrac{-mg}{m\left(v_0(t-t_0)+P_0\right)^2 + J + J_b} & 0 & 0 & \dfrac{-2m\left[v_0^2(t-t_0) + P_0 v_0\right]}{m\left(v_0(t-t_0)+P_0\right)^2 + J + J_b} \end{bmatrix}$$