RBE 502 Final
Cart-Pole Controller Evaluation

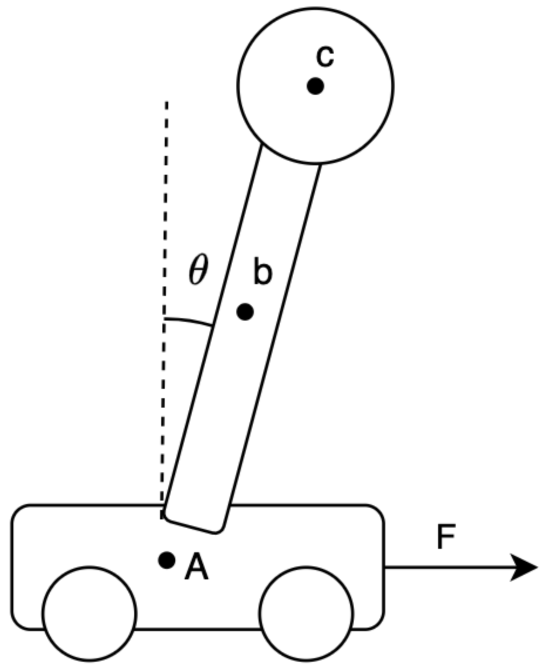
Paul Crann

Presentation Summary

- System Overview
- Related Literature
- Assumptions
- Dynamical Modeling
- Simulation Overview
- Controllers
- State Estimation
- Results
- Future Work / Learnings

System Overview

- Cart-Pole System
- 2 Degrees of freedom
- Underactuated
- Challenges:
  - Naturally unstable
  - Moving one direction requires first moving the opposite
- Real-world examples
  - Segways
  - Self Balancing Robots (evoBOT)



https://www.amazon.com/Segway-Self-Balancing-Electric-Hoverboard-Compatible/dp/B07LFD7FMF?th=1

https://newatlas.com/robotics/self-balancing-evobot-grasps-cargo/

Literature Review

- Control Design and Implementation of an Inverted Pendulum on a Cart [1]

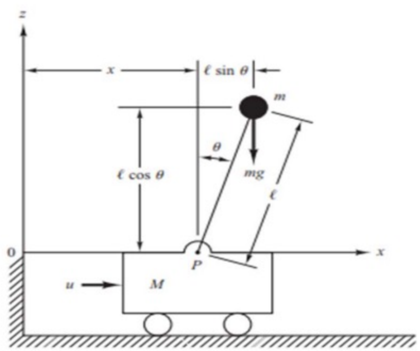


Fig. 2 Simple Pendulum on a cart [9]

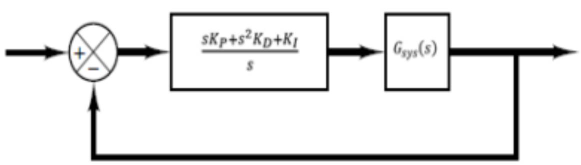


Fig. 3 PID Controller Block Diagram

- Swing-up Control of a Single Inverted Pendulum on a Cart With Input and Output Constraints [2]

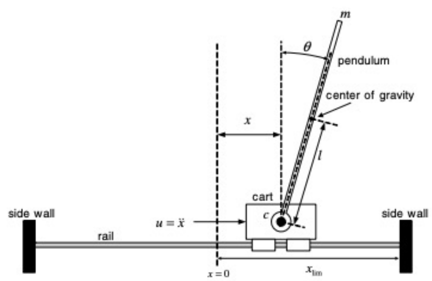


Figure 1: The conceptual diagram of a cart pendulum system.

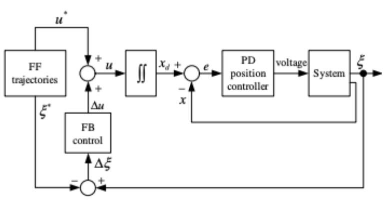


Figure 7: The 2-DOF control scheme for the swing-up.

Assumptions

- Ignore motor dynamics. Input force directly on body A.
- Input force limited by F_{range}.
- Kinetic friction about both degrees of freedom.
- Zero-mean additive Gaussian sensor (Q) and process (R) noise.

$$\boldsymbol{Q} = \begin{bmatrix} (1mm)^2 & 0 \\ 0 & (0.1deg))^2 \end{bmatrix}$$

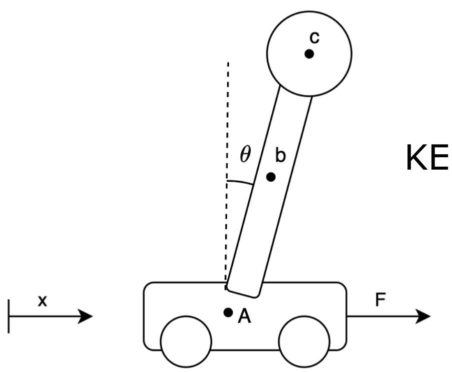
$$\boldsymbol{R} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & (1\frac{mm}{s})^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & (0.1\frac{deg}{s}))^2 \end{bmatrix}$$

Parameter	Value	Units
m_a	1.0	kg
m_b	0.2	kg
m_c	0.5	kg
l_b	0.2	m
l_c	0.4	m
I_b	0.0107	km^2
I_c	0.08	km^2
f_a	0.01	$N/\frac{m}{s}$
f_b	0.01	$N/\frac{rad}{s}$
g	9.81	m/s^2
F_{range}	[-25, 25]	N

TABLE I
SYSTEM PARAMETERS

Dynamical Modeling

- Euler Lagrangian method is used to obtain the equations of motion.
- Obtain the Lagrangian by calculating the potential, translational, and rotational energies of each mass
- Generalized forces dependent on input force and frictions



PE = PE_b + PEe_c
KE = TE_a + TE_b + TE_c + RE_b + RE_c
L = KE - PE

$$M = m_a + m_b + m_c$$

$$a = m_b l_b + m_c l_c$$

$$b = m_b l_b^2 + m c l_c^2 + I_b + I_c$$

$$L = \frac{1}{2} M \dot{x}^2 + \frac{1}{2} b \dot{\theta}^2 + a \dot{x} \dot{\theta} cos(\theta) - agscos(\theta)$$

$$\frac{\partial}{dt}(\frac{\partial L}{\partial \dot{x}}) - \frac{\partial L}{\partial x} = F - f_a \dot{x}$$

$$\frac{\partial}{dt}(\frac{\partial L}{\partial \dot{\theta}}) - \frac{\partial L}{\partial \theta} = -f_b \dot{\theta}$$

Dynamical Modeling Cont.

- Euler Lagrangian equation yields Xdd and Thetadd as functions of each other
- Algebraic manipulation to obtain Xdd and Thetadd independent of each other
- Addition of Xd and Thetad, we have the nonlinear dynamical system

$$\ddot{x} = \frac{F - f_a\dot{x} - (m_b l_b + m_c l_c) * (\ddot{\theta}cos(\theta) - \dot{\theta}^2 sin(\theta))}{m_a + m_b + m_a} \quad (5)$$

$$\ddot{\Theta} = \frac{-f_b\dot{\theta} - (m_b l_b + m_c l_c)(\ddot{x}cos(\theta) - \dot{x}\dot{\theta}sin(\theta) - gsin(\theta))}{m_b l_b^2 + m_c l_c^2 + I_b + I_c} \quad (6)$$

$$\dot{x} = f(x, u)$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix}$$

$$\dot{x}_1 = x_2 \qquad \dot{x}_2 = \frac{-absin(x_3)x_4^2 - bu + bf_a x_2}{a^2cos(x_3)^2 - Mb} + \frac{bcos(x_3)(agsin(x_3) - f_b x_4 + ax_2x_4sin(x_3))}{a^2cos(x_3)^2 - Mb}$$

$$\dot{x}_3 = x_4 \qquad \dot{x}_4 = \frac{-aMgsin(x_3) + Mf_b x_4 - aMx_2x_4sin(x_3)}{a^2cos(x_3)^2 - Mb} + \frac{acos(x_3)(asin(x_3)x_4^2 + u - f_a x_2)}{a^2cos(x_3)^2 - Mb}$$

Dynamical Modeling Cont.

- Nonlinear model not suitable for Kalman Filter, linear controller techniques(ex LQR), and to check controllability and observability
- Region of interest is the upright position
- Linearize nonlinear system about theta = 0, thetad = 0

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & \frac{-f_a b}{a^2 - Mb} & \frac{-abg}{a^2 - Mb} & \frac{b f_b}{a^2 - Mb}\\ 0 & 0 & 0 & 1\\ 0 & \frac{a f_a}{a^2 - Mb} & \frac{aMg}{a^2 - Mb} & \frac{-f_b M}{a^2 - Mb}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix} + \begin{bmatrix}0\\ \frac{b}{a^2 - Mb}\\ 0\\ \frac{-a}{a^2 - Mb}\end{bmatrix}\mathbf{u} \quad \begin{bmatrix}y_1\\ y_2\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix} + 0$$

Dynamical Modeling Cont.

- Observability: All states can be observed (obviously)

$$O = [C; CA; CA^2; CA^3]$$

$$O = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1.0000 \\ 0 & -0.0071 & -1.6613 & 0.0071 \\ 0 & 0.0084 & 13.9349 & -0.0592 \\ 0 & 0.0001 & 0.1102 & -1.6617 \\ 0 & -0.0006 & -0.8395 & 13.9385 \end{bmatrix}$$

$$rank(O) = 4 = n$$

- Controllability: Possible for system to be driven to any desired state from any initial state

$$C = [B, AB, A^2B, A^3B]$$

$$C = \begin{bmatrix} 0 & -0.7063 & 0.0109 & -1.3900 \\ -0.7063 & 0.0109 & -1.3900 & 0.1843 \\ 0 & 0.8364 & -0.0555 & 11.6590 \\ 0.8364 & -0.0555 & 11.6590 & -1.4753 \end{bmatrix}$$

$$rank(C) = 4 = n$$

Simulation Overview

- Custom MATLAB script to run simulations
- At each time step:
  - Nonlinear dynamics to update truth position according to past state, control input, and process noise
  - Takes a measurement according to the current state and sensor noise
  - Performs state estimation to update the current estimation
  - Computes next control input according to specified controller
- Other simulator options considered: ode45, Simulink, and PyBullet
- Custom MATLAB script chosen for customizability and convenience

Controllers - LQR

- Linear Quadratic Regulator – computes the optimal feedback matrix K to minimize the cost function J
- Uses MATLABs built in lqr() function
- LQR 1– Larger weight given to position x
  - Goal: Quicker movement to desired x position
- LQR 2– Larger weight given to angle theta
  - Goal: Smaller theta deviations
- Multiplies this computed K value by the current state error

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

$$Q1 = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R = 1, N = 0$$

$$Q2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R = 1, N = 0$$

```
function u = inputLQR(x, xd, K)
    u = K*(x-xd);
end
```

Controllers – 2 Step PD Controller

- The goal of this controller is to use two steps of PD control
- A first PD controller calculates an instantaneous desired theta angle that would result in the systems state converging on the desired goal state
- The second step PD controller is used to calculate a control input that will drive the systems angle towards the desired angle
- Between two states, theta desired will be saturated between the limits [-3, 3] degrees to avoid too large angle commands

Stage 1:

$$\theta_{desired} = P_1 * (x_{desired} - x) - D_1 * \dot{x}$$

State 2:

$$u = -P_2 * (\theta_{desired} - \theta) + D_2 * \dot{\theta}$$

P1 = 10 D1 = 30 P2 = 65 D2 = 45

```
function u = inputPD2(x_est, xd, pd2)
    thetaD = pd2.Pt*(xd(1)-x_est(1)) - pd2.Dt*(x_est(2));
    if thetaD < deg2rad(-3); thetaD = deg2rad(-3);
    elseif thetaD > deg2rad(3); thetaD = deg2rad(3); end
    u = -pd2.Pu*(thetaD - x_est(3)) + pd2.Du*x_est(4);
end
```

State Estimation – Kalman Filter

- Sensors receive noisy x and theta measurements

```
function x_measure = measureState(x_truth, p , C)
    x_measure = C*x_truth + sqrt(p.noise.sensor)*randn([2,1]);
end
```

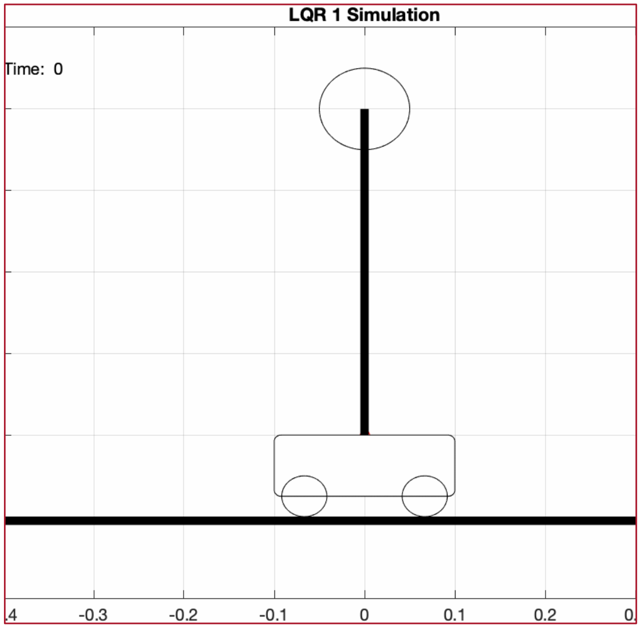
- Kalman filter uses these measurements along with a model of the system to produce a more optimal estimate of the systems states

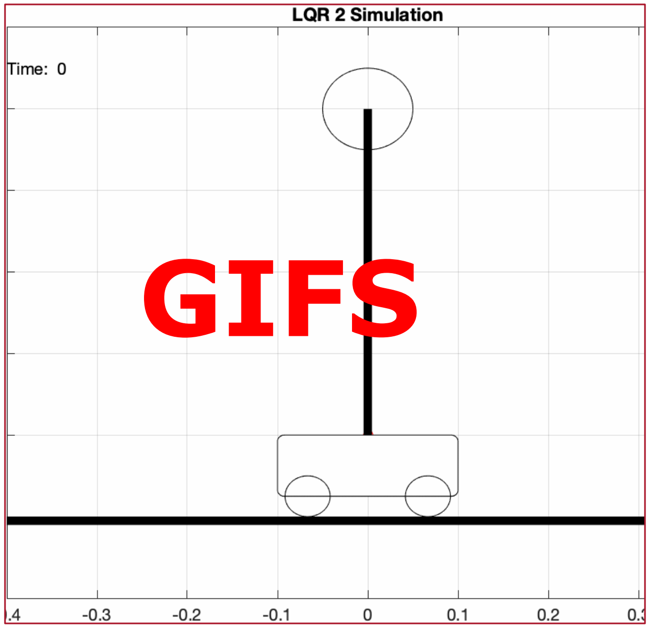
```
pred = stateTransition(x_est_old, u_old, p, dt, false);
P = A*P*A.'+ R;
K = P*transpose(C)/(C*P*C.' + Q);
x_est = pred + 10*K*(z - C*pred);
P = (eye(4) - K * C) * P;
```

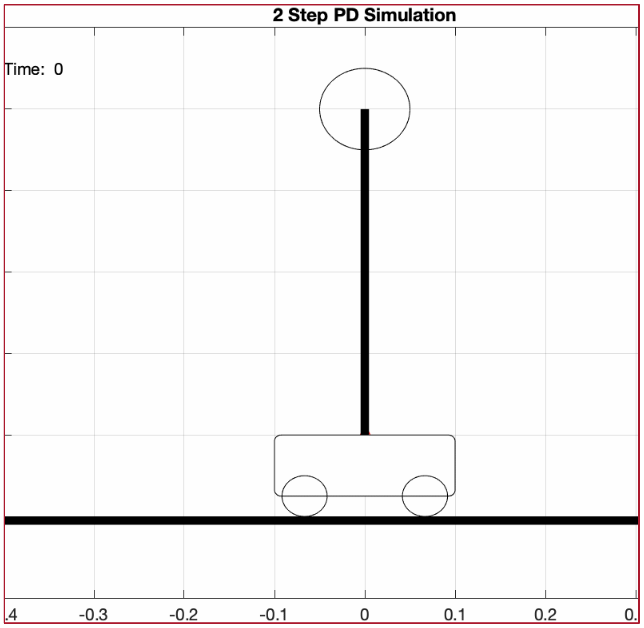
$$\boldsymbol{Q} = \begin{bmatrix} (1mm)^2 & 0 \\ 0 & (0.1deg))^2 \end{bmatrix}$$

$$\boldsymbol{R} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & (1\frac{mm}{s})^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & (0.1\frac{deg}{s}))^2 \end{bmatrix}$$

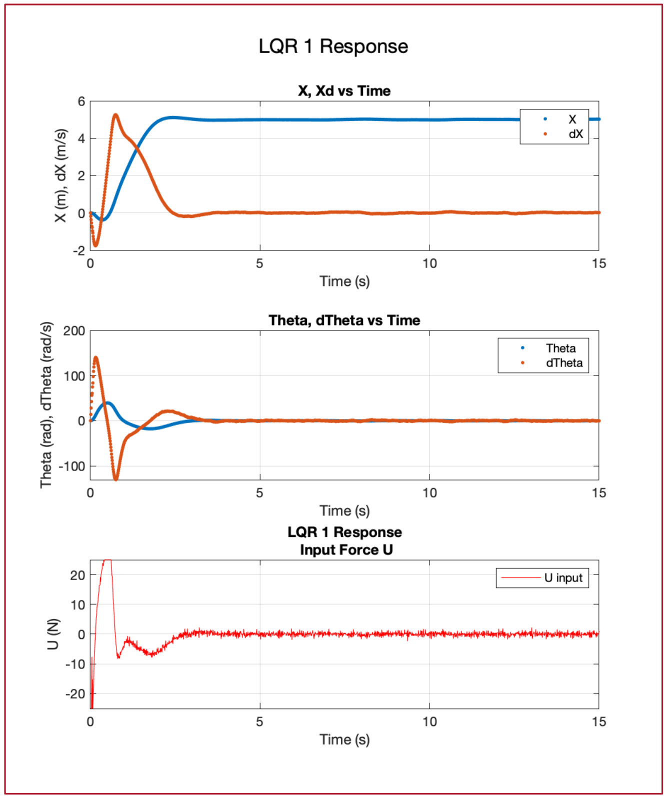
Results

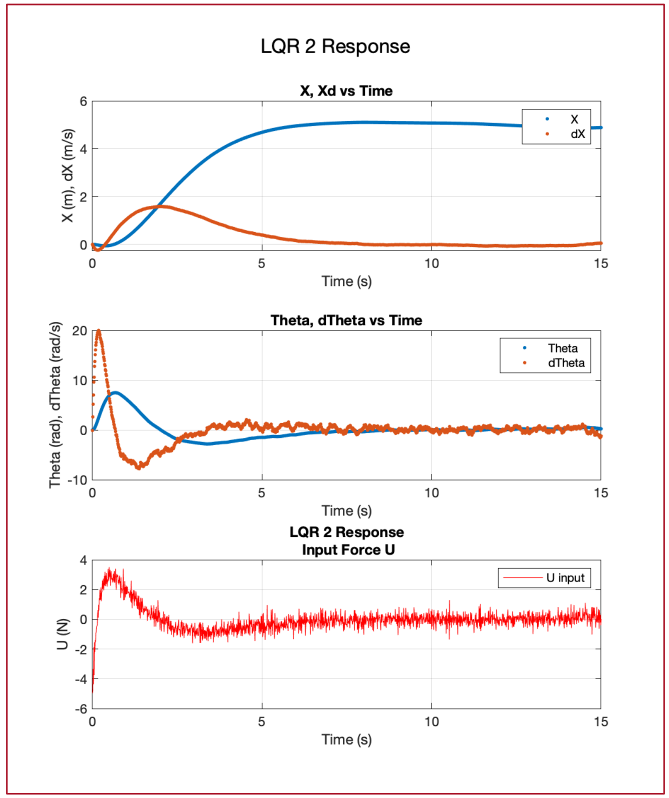

LQR 1 Simulation
Time: 0
-0.3
-0.2
-0.1
0
0.1
0.2

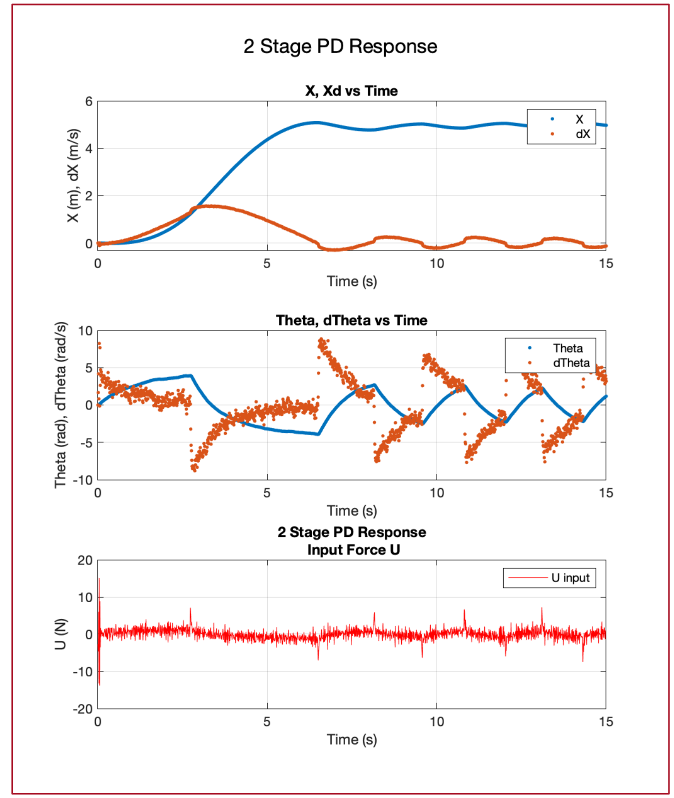

LQR 2 Simulation
Time: 0
GIFS
-0.3
-0.2
-0.1
0
0.1
0.2


2 Step PD Simulation
Time: 0
-0.3
-0.2
-0.1
0
0.1
0.2

Results Cont.

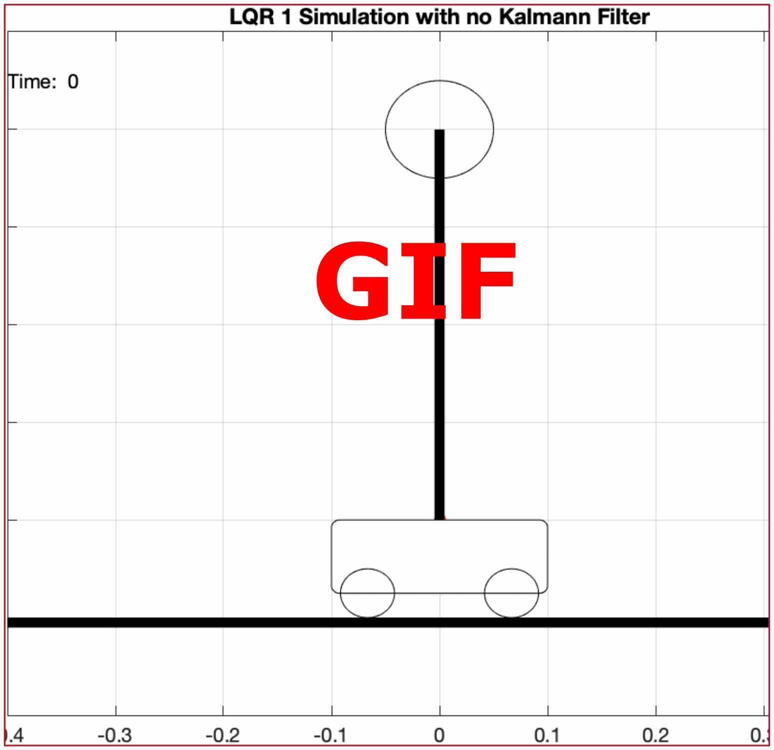


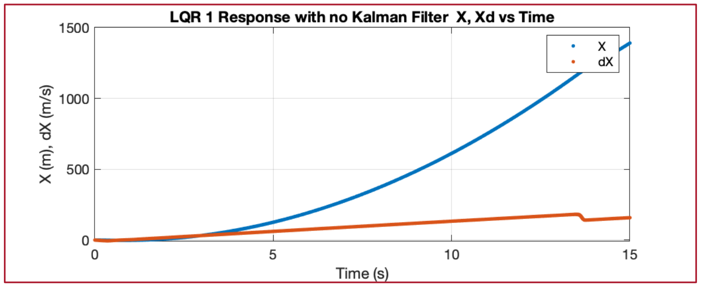


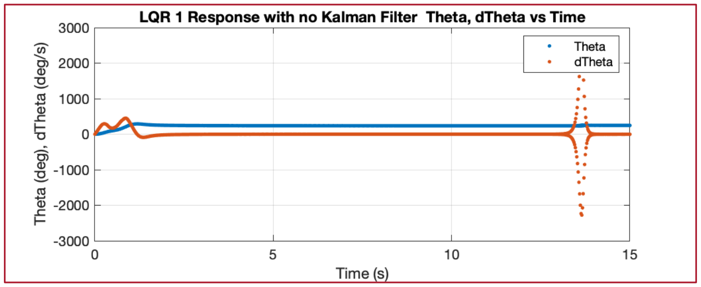


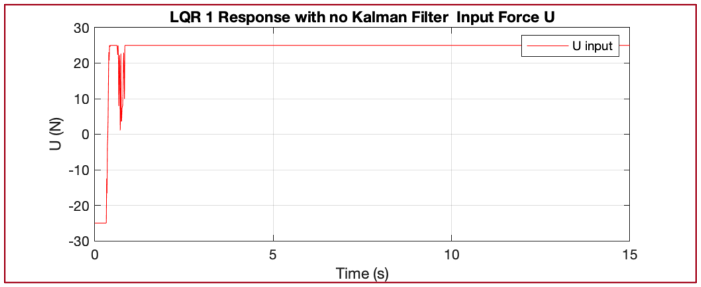
Results Cont.

Response with no Kalman Filter









Results Cont.

- Monte Carlo Evaluation
  - 100 Simulations of each controller
  - Assessed on 6 different metrics

Controller type	LQR 1	LQR 2	2 Stage PD
% Success	100	100	100
5% Settling Time (s)	1.89	5.17	8.11
x Overshoot (mm)	94.1	84.1	66.3
Max $\dot{x}$ (m/s)	5.36	1.55	1.55
Max θ (deg)	39.62	7.37	3.83
Max $\dot{\theta}$ (deg/s)	141	19.6	9.54

Future Work / Learning Outcomes

Future Work

- Evaluating controllers on additional tasks
  - Trajectory Following, Disturbance Rejection
- Further tunning of controllers for more optimal results
  - Reduce oscillations in 2 Stage PD controller
- Testing additional controllers on this system
- Robust or adaptive control with unknown load on top of inverted pendulum

Learning Outcome

- Practice taking a system through dynamical modeling, controller design, and simulation
- Designing LQR's and tuning a PD controller
- Implementing Kalman Filters for state estimation
- If I were to do it again:
  - Implement system in a physics-based 3D-simulator for higher fidelity model verification

References

- [1] M. Owais, A. Ul-Haque, H. A. Rahim, S. Aftab and A. A. Jalal, "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, 2019, pp. 1-6, doi: 10.1109/ICETAS48360.2019.9117388.
- [2] M. Tum, G. Gyeong, J. H. Park and Y. S. Lee, "Swing-up control of a single inverted pendulum on a cart with input and output constraints," 2014 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO), Vienna, Austria, 2014, pp. 475-482, doi: 10.5220/0005018604750482.

Thank You!