Nicholas Ha

JAN 2019

CONTENTS

Introduction

- Background
- Goals

Project A: Work at LANL's particle accelerator: (2 summers)

Project B: LQG-based control design for jet (class final assignment)

Intro

Well-rounded and curious, desire to work on challenging problems, with interest and expertise in control systems and autonomous systems.

Education

- MSEE: Intelligent Systems, Robotics and Control
- BSEE: controls, signal processing, machine learning
- Also 2 years as ME, took Statics, Dynamics, Solid Mechanics, Thermo, and Numerical Methods
- Goal: **broad knowledge base** and **mathematical ability**

Design projects

- Arduino based: Vibration Table, Grand PrIEEE, Quadcopter, cycling computer
- Other: Hyperloop Pod, Intro MAE design class

Industry experiences:

- Developing missile launchers, improving lasers for photolithography, and steering particle accelerators for isotope production

Hands-on, passionate, and driven

Voluntary Projects:

High-school

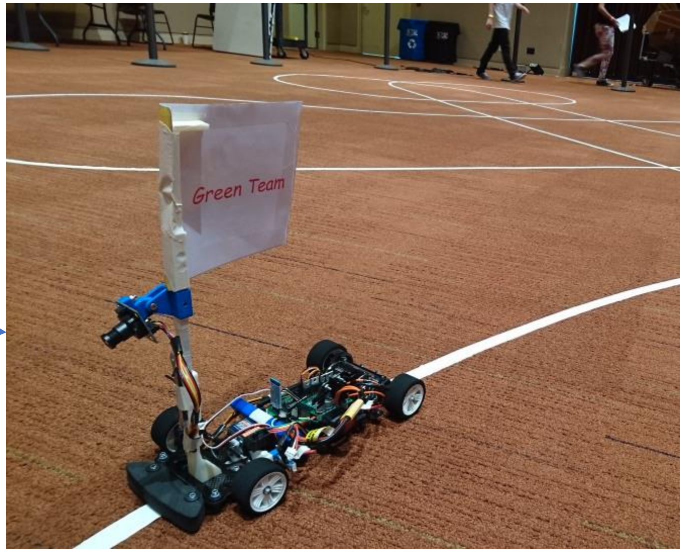
- RC Blimp, Science Olympiad

Undergrad:

- Cycling computer
- Quadcopter
- Hyperloop Pod Team
- Grand PrIEEE

'Hands-on' grad classes

"Most Spectacular Failure ($500)" – *San Jose Tech Challenge 2012*



Grand PrIEEE

Career Preferences/Goals

Long-term goal: conduct impactful work in aerospace, transportation, sustainability

Short-term goals:

- Work on an ambitious team
- Wear different hats to learn many skills
- Get familiar with standard platforms and procedures, gain more domain knowledge

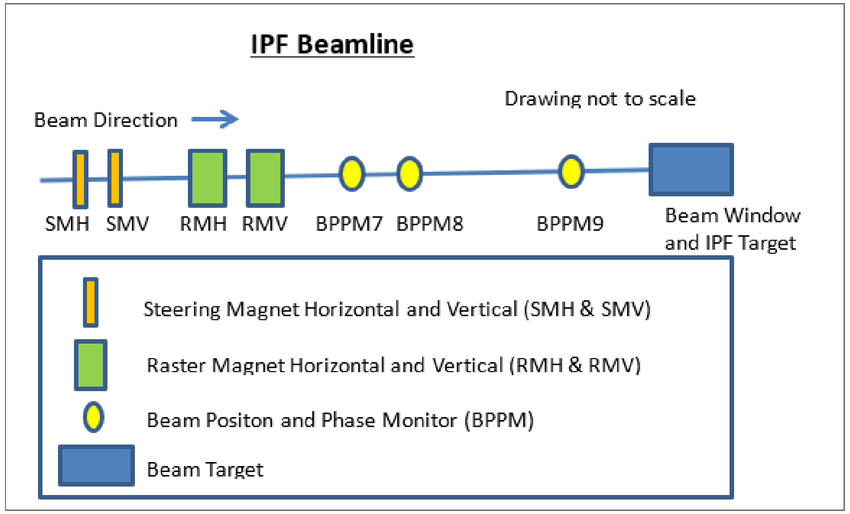
Project A: LANL

Particle accelerator bombards circular target with pulses of protons following raster sequence of concentric circles

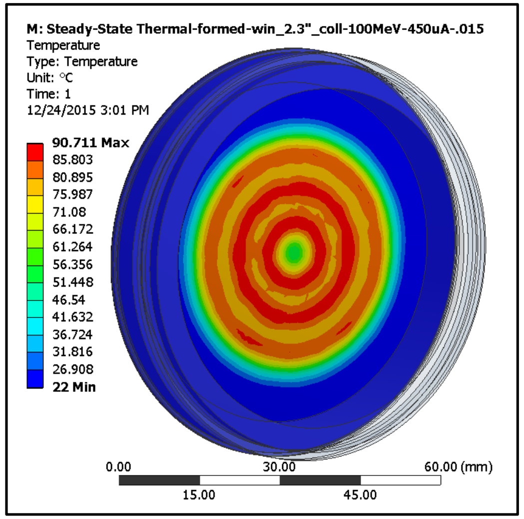
Goals:

1. Ensure beam follows reference behavior (estimation and control design)
2. Determine the ideal reference behavior: (optimization problem)

Make these improvements with minimal interference to accelerator operation



Main IPF beamline components



Temperature simulation on beam window

Project Outcomes Overview

Before:	Goal	Results
Fault Detector: based on actuator inputs	Fault Trigger based on actual failure mode	Accurately simulated temperature to use as a fault condition
Monitoring: multiple sensors giving ***(x,y)*** and intensity data	Capture key characteristics of beam	MLE algorithm for size and center location of each "circular pulse" of beam
Control: is done open-loop → performance can slowly drift away from nominal	Closed loop control of circle size and center location	Circle size: control by updating the estimate of system parameters. Center location: PI (Proportional Integral) controller
Optimization: Hand chosen beam parameters for baseline	Better choice of beam parameters	Method for improving baseline parameters

Proof-of-concept in Simulink completed

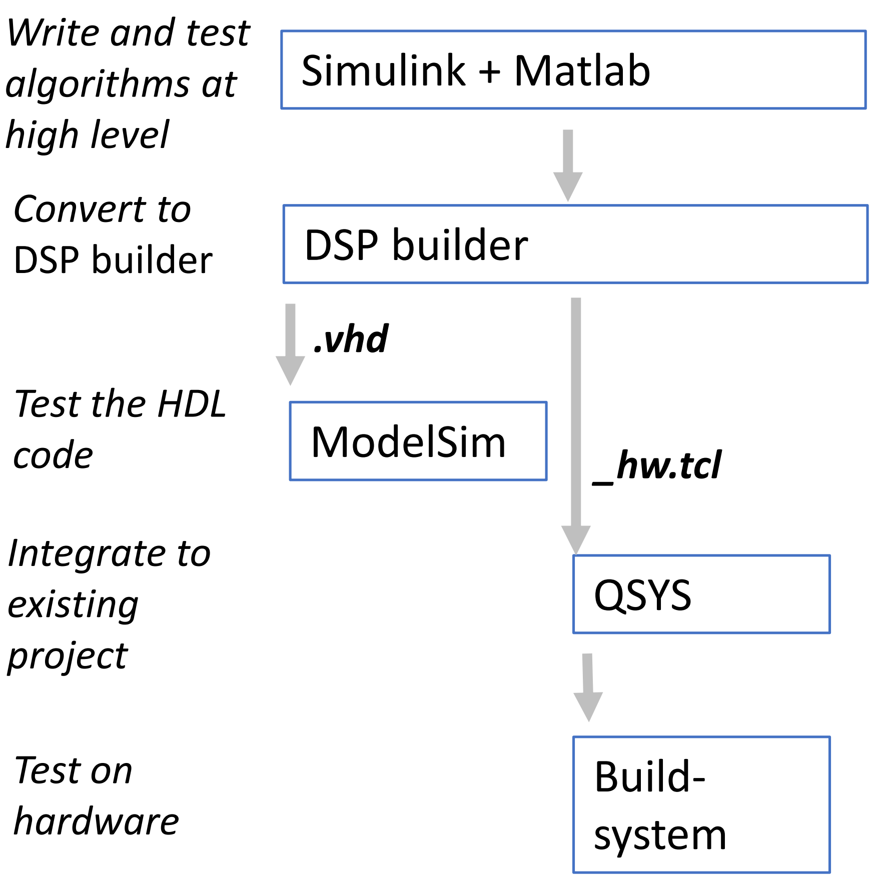
Subject of MS Thesis

Fast Protect, Monitoring, And Control

APPROACH

Status:

- Proof-of-concept for all algorithms in Simulink/Matlab
- Test-benches for Simulink and ModelSim are ready



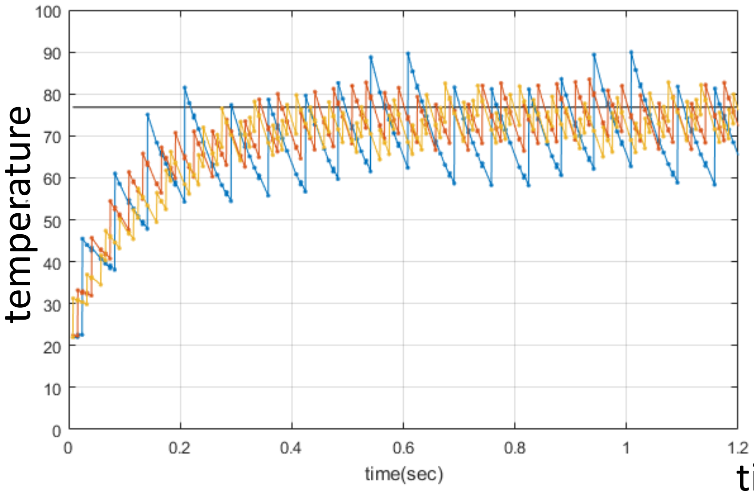
Fast-protect Trigger

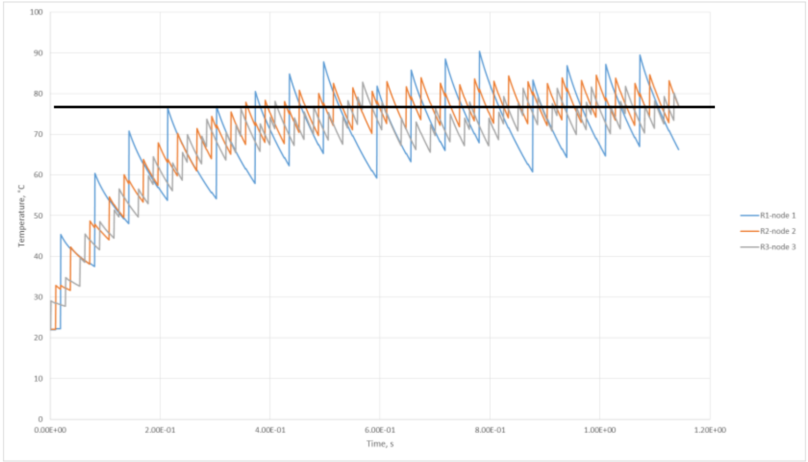
Goal: *prevent overheating of target window*

My simplified model: beam on - dT/dt linear w.r.t. current
beam off - Newtons law of cooling

Performance:

temperature





time

My model

Full simulation

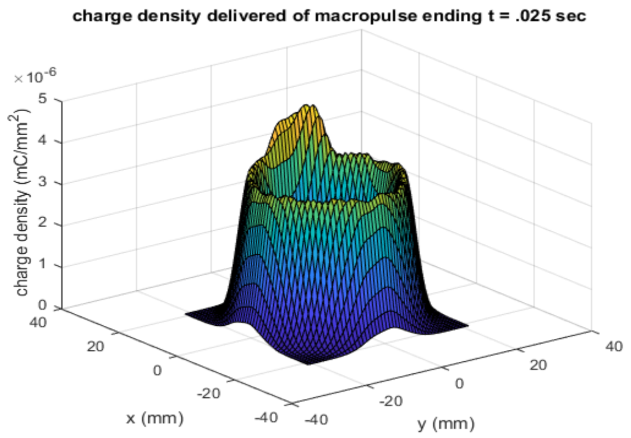
Monitoring

Problem:

- Given ~ 62 (x, y) samples
- Estimate circle parameters: size and center location

Approach: least squares MLE

Performance: With 0.1mm noise in (x, y) per sample and 62 points: accuracy in A < 0.02 mm



$$y = A\sin(\omega x + \psi) + C = A\sin\omega x\cos\psi - A\cos\omega x\sin\psi + Cx + \text{noise}$$

$$\mathbb{Y} = A\,\mathbb{X}, \text{ then } \mathbb{X} = (A^TA)^{-1}A^T\,\mathbb{Y}$$

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} \sin\omega x_1 & -\cos\omega x_1 & 1 \\ \sin\omega x_2 & -\cos\omega x_2 & 1 \\ \vdots & \vdots & \vdots \\ \sin\omega x_n & -\cos\omega x_n & 1 \end{pmatrix} \begin{pmatrix} A\cos\psi \\ A\sin\psi \\ C \end{pmatrix}; \text{ then solve for } \mathbb{X} \text{ using}$$

then $A = \sqrt{\mathbb{X}(1)^2 + \mathbb{X}(2)^2}$

$\psi = \cos^{-1}(\mathbb{X}(1)/A)$

$C = \mathbb{X}(3)$

Setup for least squares MLE

Control

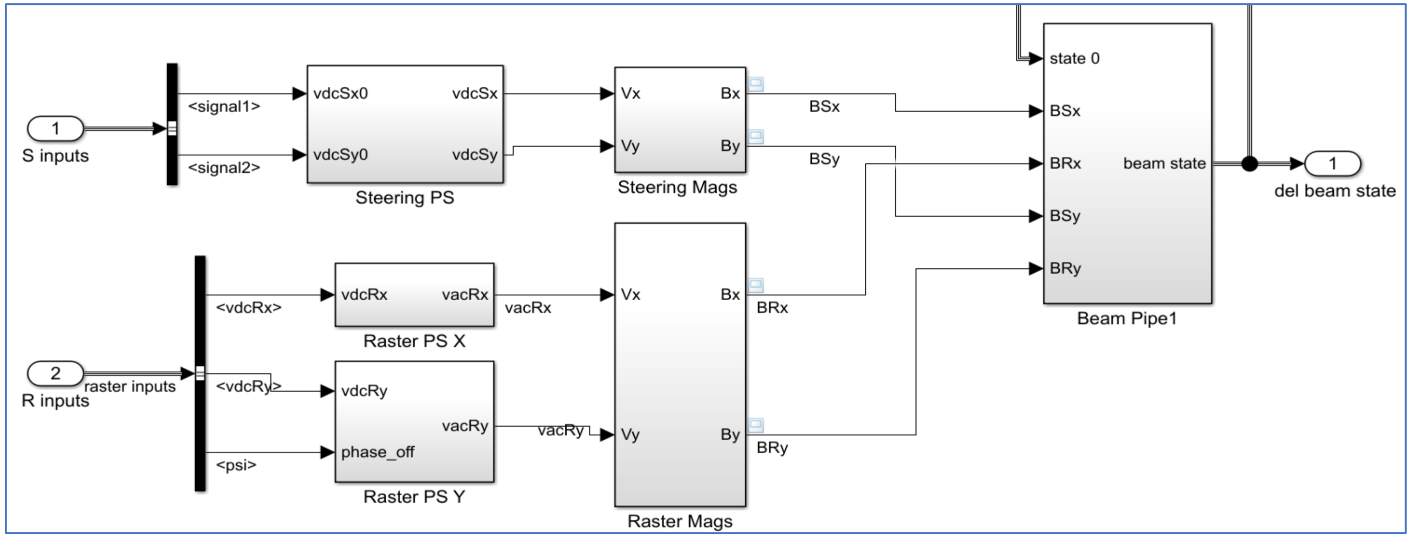
Goals: minimize steady state error for

1. circle size
2. center location

Approach: note that transients are negligible

1. Retain OL design but adaptively correct the gain
2. PI controller

Results:
Steady state errors minimized



Plant Model

Project B: Control design for jet

Problem: design yaw rate damper while preserving a certain normal flight characteristic

- damping ratio ***zeta > 0.35***, with
- natural frequency ***Wn < 1.0*** rad/s

Approach: structured LQG design

Results: vs Matlab demo using root locus,

- Better damping
- More complex yet more structured and generalizable approach

The LTI system

$$\dot{x}(t) = \begin{bmatrix} -.0558 & -.9968 & .0802 & .0415 \\ .598 & -.115 & -.0318 & 0 \\ -3.05 & .388 & -.4650 & 0 \\ 0 & 0.0805 & 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} .00729 & 0 \\ -0.475 & 0.00775 \\ 0.153 & 0.143 \\ 0 & 0 \end{bmatrix} u(t)$$

is a simplified trim model of a Boeing 747 during cruise flight. The four states are

x_1: side-slip angle,

x_2: yaw rate,

x_3: roll rate, and

x_4: bank angle.

The two inputs are

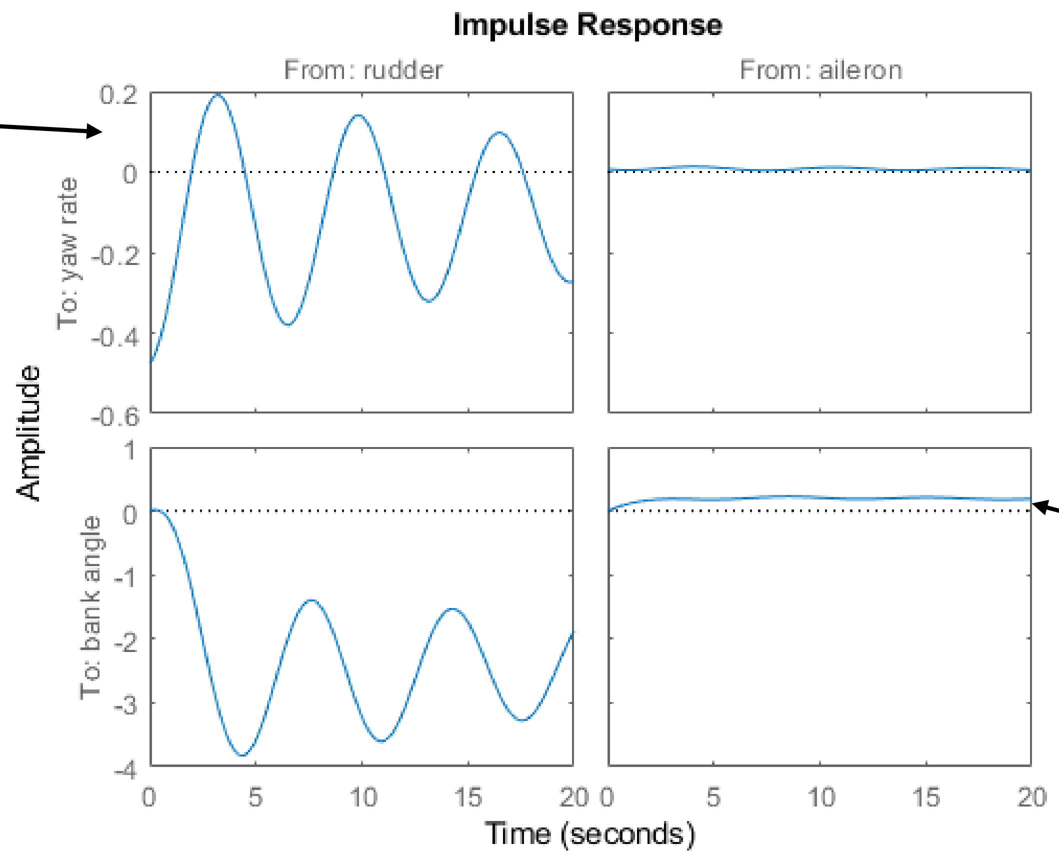
u_1: rudder deflection, and

u_2: aileron deflection.

State space model

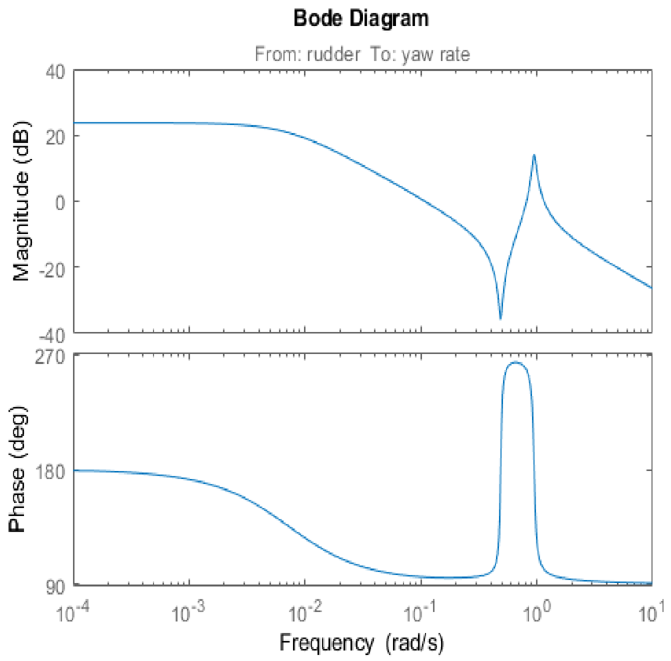
Open-loop responses

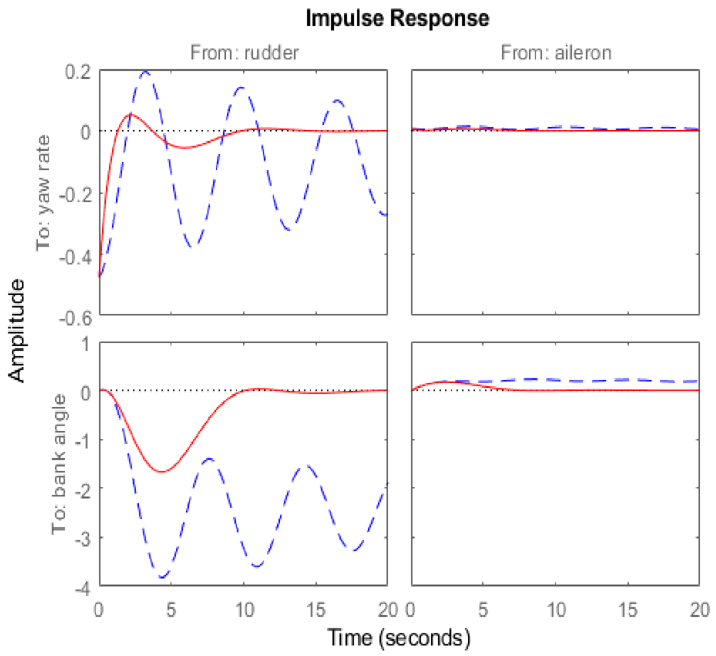
underdamped

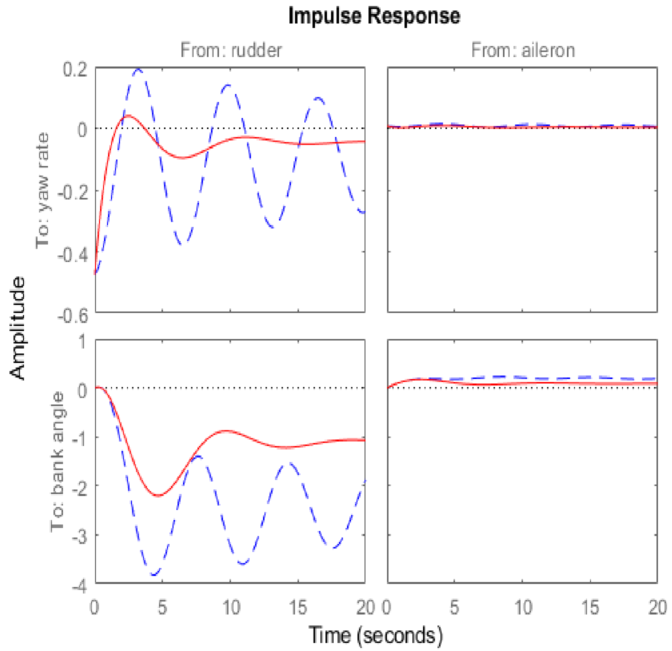


Some steady state error is good

Classical approach







1. Pick a feedback path to try
2. Design a pure gain controller via root locus
3. Design a washout filter to cascade with pure gain controller

Blue = open loop

LQG approach

1. Select same feedback path
2. Put into desired form
3. Design basic controller
4. If not sufficient, design controller that's forced to have a zero
    1. Design for plant with added zero, then transfer it to the controller

Given the LTI system

$$\begin{aligned} \dot{x} &= Ax + B_u u + B_w w, \\ y &= C_y x + D_{yw} w, \\ z &= C_z x + D_{zu} u, \end{aligned}$$

and the *observer-based controller*

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + B_u u + F(\hat{y} - y), \\ \hat{y} &= C_y \hat{x}, \\ u &= K\hat{x}. \end{aligned}$$

compute (K, F) that stabilize the closed loop system and minimize the cost function

$$J := \lim_{t\to\infty} E\left[z(t)^T z(t)\right].$$

LQG problem formulation

LQG design

1. Put into desired form
    1. A and B are given. Determine the other matrices
2. Select which terms of these matrices will affect damping
3. Determine **q/r** ratio to achieve desired damping ratio

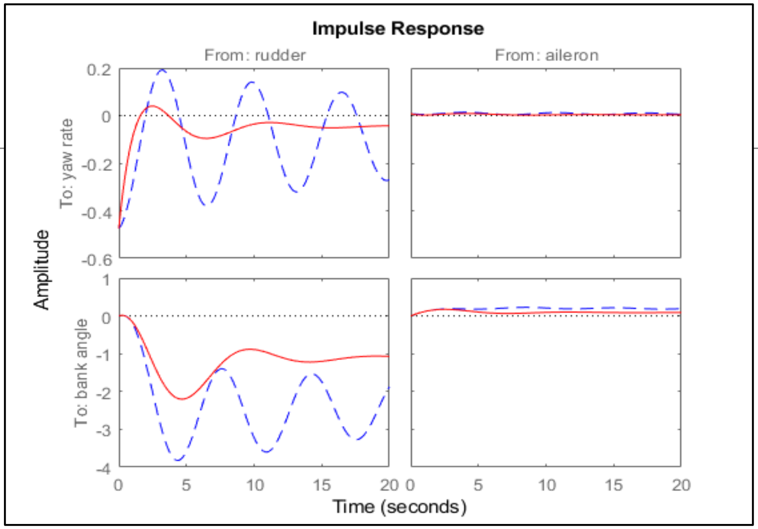
$$B_u = B[:,1] \qquad B_w = \begin{bmatrix}0 & 0\\ 1 & 0\\ 0 & 0\\ 0 & 0\end{bmatrix}$$

$$C_y = [0 \ 1 \ 0 \ 0] \qquad D_{yw} = [0 \ 1]$$

$$C_z = \begin{bmatrix}0 & q & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix} \qquad D_{zu} = \begin{bmatrix}0\\ r\end{bmatrix} \qquad \begin{matrix} q \geq 0 \\ r > 0 \end{matrix}$$

Results

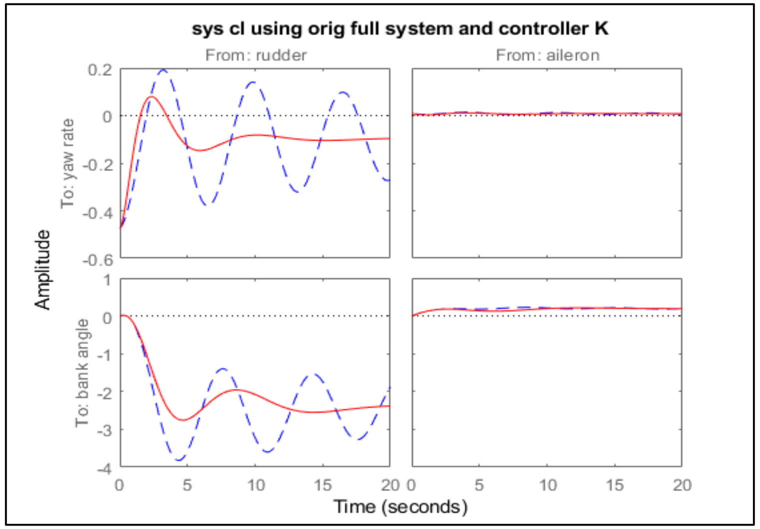
demo



Controller Transfer Function

$$\frac{2.34\ s}{(s+0.2)}$$

LQG



$$\frac{0.0011824\ s\ (s+3496)\ (s+0.5786)\ (s+0.009039)}{(s+2.572)\ (s+0.4954)\ (s^2 + 0.3473s + 0.1327)}$$

SUMMARY

Strong foundation + willingness to learn makes me highly qualified for many teams

Willing to use any of my knowledge and skills to get the job done

Enthusiastic and passionate about doing challenging and impactful work