Smart Control Algorithm for 2-DOF Helicopter

Glenn Janiak Kenneth Vonckx Advisor: Dr. Suruz Miah

Department of Electrical and Computer Engineering
Bradley University
1501 W. Bradley Avenue
Peoria, IL, 61625, USA

Saturday, May 4, 2019

Outline

Introduction

- Helicopter are important for short-distance travel
  - air-sea rescue
  - fire fighting
  - traffic control
  - tourism
- Purpose of control system
  - resistance to turbulence
  - enable use of mobile device
- Which is better?
  - Optimal Control (Linear Quadratic Regulator)
  - Optimal Noise Resistant Control (Linear Quadratic Gaussian)
  - Machine Learning (Approximate Dynamic Programming)

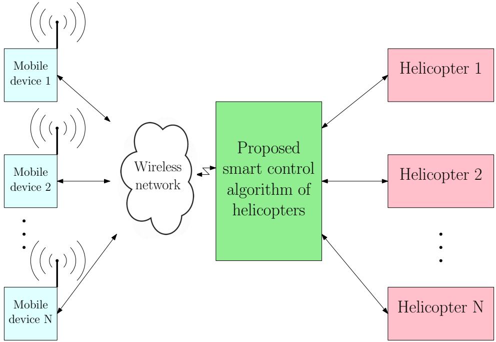


Figure 1: General High-Level System Architecture

Introduction

- This project will:
  - use a pair of 2-DOF (2-degrees-of-freedom) testing platforms
  - implement control algorithms on embedded system
  - use mobile device for user control
  - encourage research
  - serve as an educational tool

Outline

Various control techniques have been proposed for 2-DOF helicopters such as:

- Sliding mode control [?]
- Fuzzy Logic control [?] [?] [?]
- Data-driven Adaptive Optimal Output-feedback control [?]
- Decentralized discrete-time neural control [?]

These control techniques employ advanced mathematics that are difficult to implement on embedded systems.

Outline

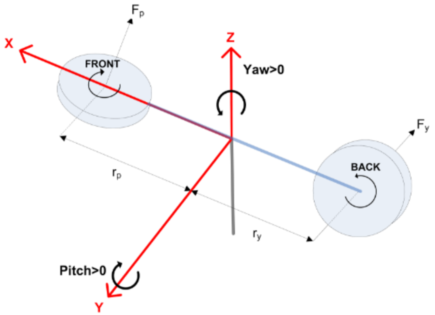


Figure 2: Model of a 2-DOF Helicopter

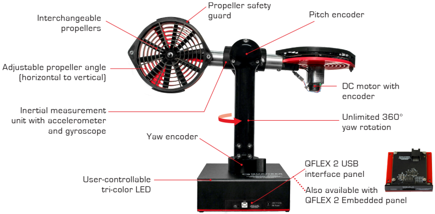


Figure 3: Quanser Aero

- Characterized by fixed base
  - Can change 2 of 3 possible orientations...
    - Pitch (θ)
    - Yaw (ψ)
    - *Not Roll*
  - and cannot change position
    - x direction
    - y direction
    - z direction

- Motors are attached to the propellers to create thrust due to air resistance
  - Main - changes pitch angle
  - Tail - changes yaw angle
- Torque due to rotation also creates a force on opposite axes

Due to the efficiency of the Quanser Aero, we can create a linearized system model:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \text{ such that} \tag{1}$$

$$\begin{bmatrix} \dot{\theta} \\ \dot{\psi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -K_{sp}/J_p & -D_p/J_p & 0 \\ 0 & 0 & 1 & -D_y/J_y \end{bmatrix} \begin{bmatrix} \theta \\ \psi \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ K_{pp}/J_p & K_{py}/J_p \\ K_{yp}/J_y & K_{yy}/J_y \end{bmatrix} \begin{bmatrix} V_p \\ V_y \end{bmatrix}$$

- K_{sp} - being the stiffness of the axes
- K_{pp} - pitch motor thrust constant
- K_{py} - thrust constant acting on the pitch angle from the yaw motor
- K_{yp} - thrust constant acting on the yaw angle from the pitch motor
- K_{yy} - yaw motor thrust constant
- J_p - moment of inertia about pitch axis
- J_y - moment of inertia about yaw axis
- D_p - viscous damping of the pitch axis
- D_y - viscous damping of the yaw axis

Outline

Background Study

Prior Work

- extensive modeling & simulations
- implementation of two motion control algorithms (LQR & ADP)
- one helicopter

Outline

Subsystem Level Functional Requirements
Block Diagram

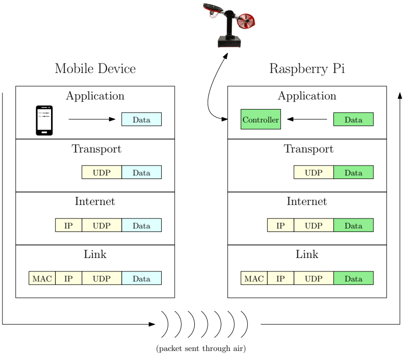


Figure 4: Communication Model

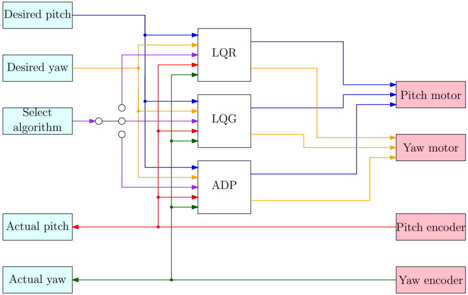
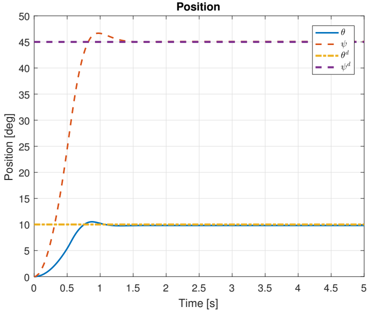
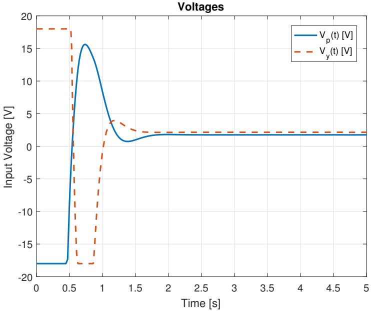


Figure 5: Low Level Smart Control Diagram

Outline



(a)



(b)

Figure 6: Optimal Control (P Controller) Simulation (a) Position and (b) Voltage w/ Step Input

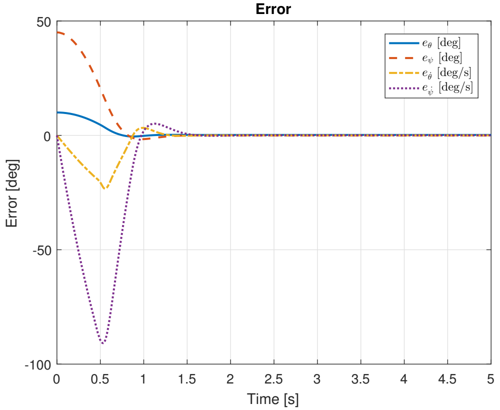
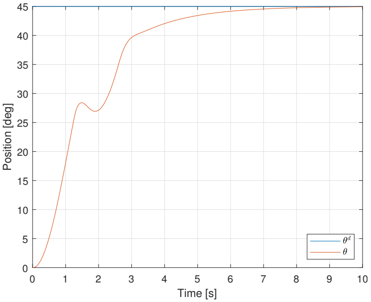
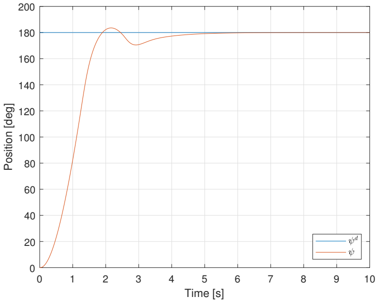


Figure 7: Optimal Control (P Controller) Simulation w/ Constant Signal

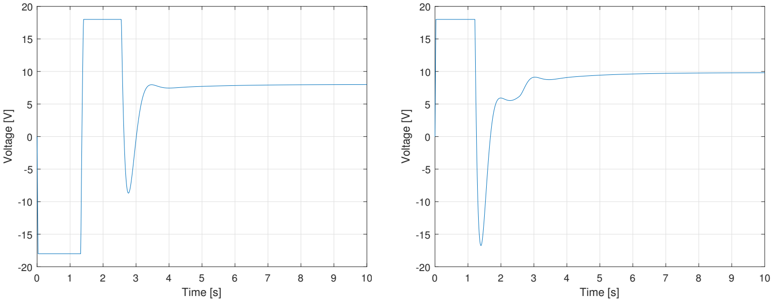


(a)



(b)

Figure 8: Optimal Control (PI Controller) Simulation (a) Pitch Position and (b) Yaw Position w/ Step Input

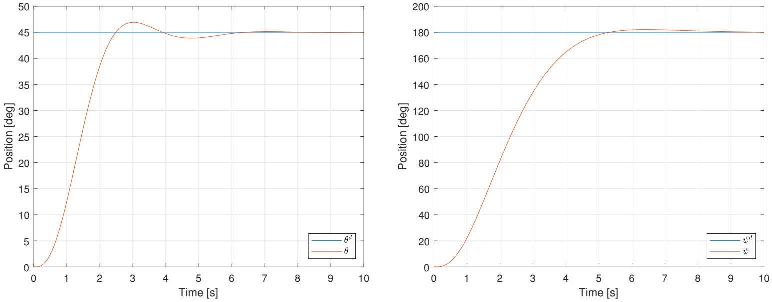


(a) (b)

Figure 9: Optimal Control (PI Controller) Simulation (a) Pitch Voltage and (b) Yaw Voltage w/ Step Input

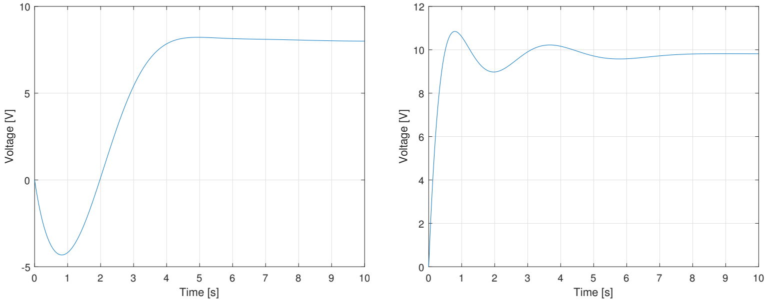
Outline

Optimal Noise Resistant Control (PI Controller) Simulation



(a) (b)

Figure 10: Optimal Noise Resistant Control (PI Controller) (a) Pitch Position and (b) Yaw Position w/ Step Input



(a) (b)

Figure 11: Optimal Noise Resistant Control Simulation (PI Controller) (a) Pitch Voltage and (b) Yaw Voltage w/ Step Input

Outline

Machine Learning and Optimal Control (P Controller) USB

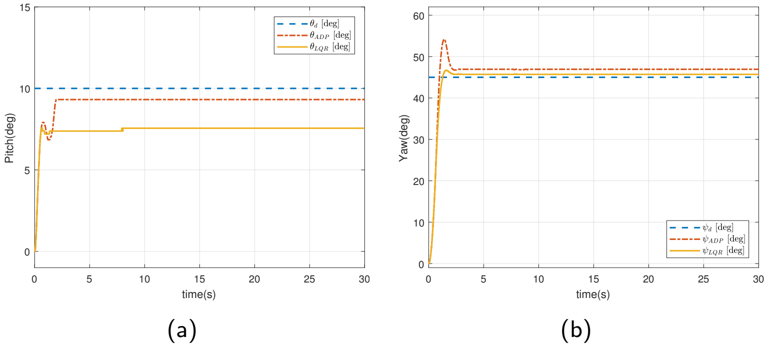


Figure 12: USB Implementation comparison between Machine Learning and Optimal Control (P Controller) for (a) Pitch and (b) Yaw orientations w/ Step Input

Outline

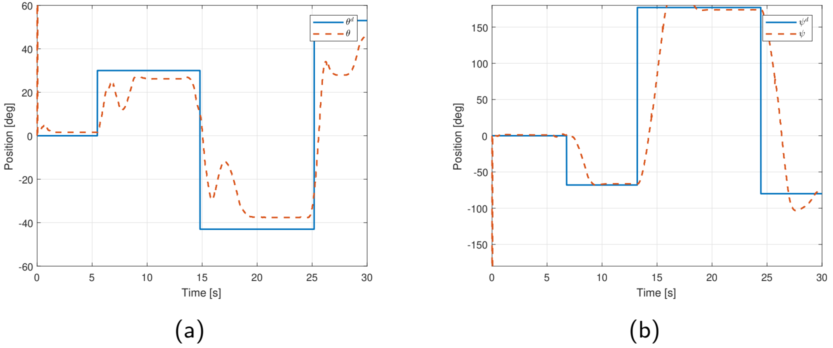
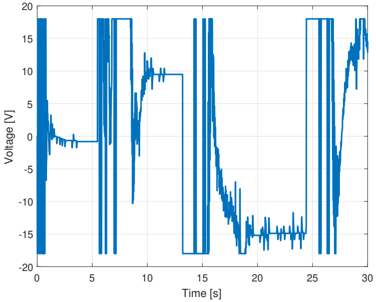
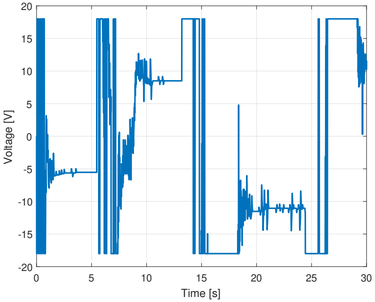


Figure 13: Optimal Control (P Controller) (a) Pitch Position and (b) Yaw Position w/ input from Mobile Phone

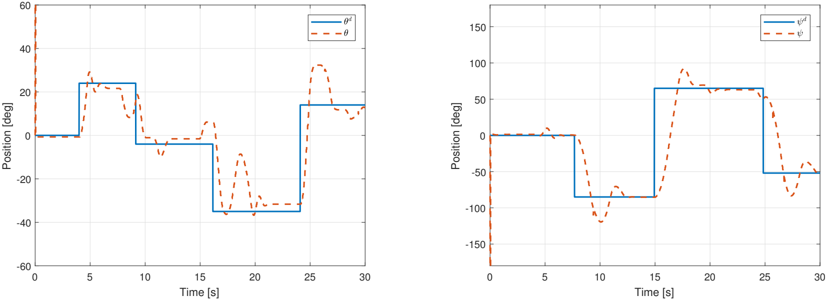


(a)



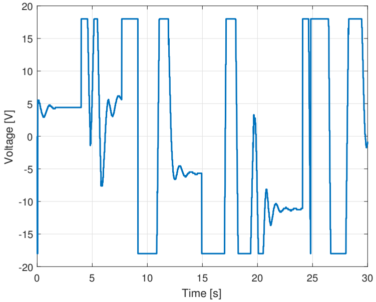
(b)

Figure 14: Optimal Control (P Controller) (a) Pitch Voltage and (b) Yaw Voltage w/ input from Mobile Phone

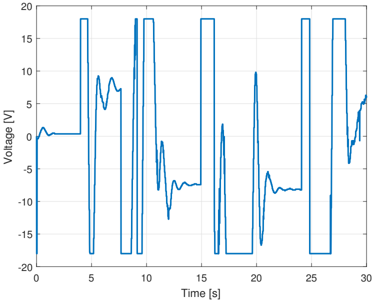


(a) (b)

Figure 15: Machine Learning (a) Pitch Position and (b) Yaw Position w/ input from Mobile Phone



(a)



(b)

Figure 16: Optimal Control (P Controller) (a) Pitch Voltage and (b) Yaw Voltage w/ input from Mobile Phone

Outline

Demonstration

- Discretization of System
- Digital Compass
- Enhanced Smart Control

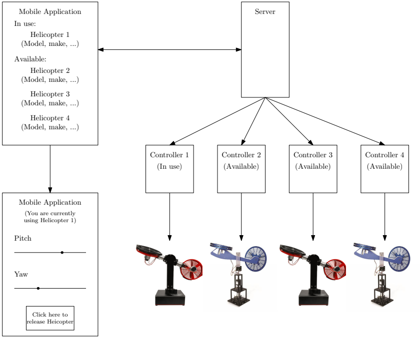


Figure 17: Enhanced Smart Control

- Implmentation on 6-DOF Helicopter

Summary

- Embedded implementation of control algorithms
- Mobile interface
- PI control improves steady-state error
- Machine Learning is best when system parameters are unknown or time-varient
- Add table for RMSE?

For Further Reading I