Experiments on 2-DOF Helicopter Using Approximate Dynamic Programming

Anthony Birge Andrew Fandel
Advisor: Dr. Suruz Miah

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

Tuesday, November 28, 2017

BRADLEY University

Outline I

Outline II

Introduction

- Unmanned Aerial Vehicles (UAVs) offer a small-form solution to many aerial tasks and provide many benefits such as cost effectiveness and the lack of a pilot.
- One of the most challenging tasks in control applications is to model a system, such as the quadcopter, half-quadcopter, and helicopter considered in this project, using a set of ordinary nonlinear differential equations.
- There are many applications of quadcopters, half-quadcopters, and helicopters.

- Quadcopter
  - Standard quadcopter
  - Industrial drone
  - Personal drone
- Half-Quadcopter
  - Chinook helicopter
  - Concrete power screeder
- Helicopter
  - Standard helicopter
  - Hobby helicopter

- Understand the Quanser AERO and methods of controlling it which will allow for development of teaching materials for future classes.
- Implement the proposed approximate dynamic programming algorithm in [7] to both the half-quadcopter and helicopter configuration of the Quanser AERO (seen on next slide).
- Implement other controllers to the Quanser AERO to measure the effectiveness of the proposed approximate dynamic programming control algorithm.

Introduction

Quanser AERO

Figure: Quanser AERO configured as a 2-DOF helicopter.

- Approximate dynamic programming is the strategic method of dividing a complex problem into simpler problems and then solving those simpler problems with use of applicable algorithms.
- General steps proposed in [7]
  - Divide the total time of control into periods of T seconds.
  - Collect error data every τ seconds in each period T.
  - Use an actor-critic neural network as the data is collected to find the optimal inputs to minimize the error.
  - Use the updated optimal inputs for the next period T.

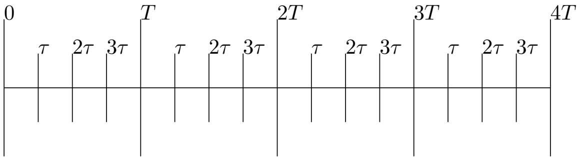

Figure: Timing in the approximate dynamic programming algorithm.

Various control techniques have been proposed for the quadcopter including the following:

- PD Controller [6]
- PID Controller [3]
- PI Control [6]

Various control techniques have been proposed for the 2-DOF helicopter like the following:

- Sliding mode control [1]
- Fuzzy model-based control [10, 5]
- Data-driven adaptive optimal control [2]
- Neural control [4]
- Reinforcement learning optimal control [8]
- Augmented baseline LQR [9]

These control techniques use complicated, advanced mathematics. We also found that few techniques use approximate dynamic programming for real-time implementation.

High-Level System Block Diagram

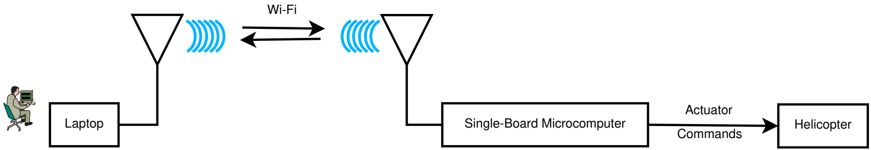

Figure: High-level system block diagram.

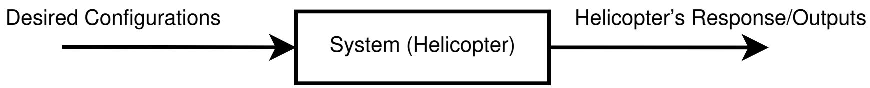

Figure: Helicopter subsystem block diagram

Preliminary Work

- Researching derivations of the quadcopter state-space model
- Developing a method in MATLAB to plot the quadcopter (use a standard PI controller to verify the results)
- Simulating a quadcopter in V-REP for a visual representation
- Goal is to extend these projects to the approximate dynamic programming control algorithm

The quadcopter free-body diagram is shown below.

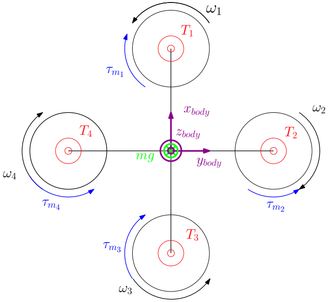

Figure: Free-body diagram in body frame - top view.

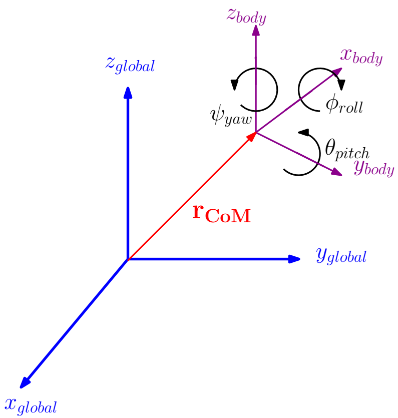

Figure: Global frame, body frame, and position vector.

The variables used in the quadcopter state-space model are as follows:

- l_q is the length of a quadcopter arm.
- m is the mass of the quadcopter.
- g is the gravitational constant.
- b is the drag coefficient for the propellers.
- $\tau_{m_i} = b\omega_i^2$ is the yaw processional torque from a single propeller.
- F_d is the drag force coefficient.
- c_t is the thrust coefficient.
- ω_i is the propeller (1-4) angular speed.
- $T_i = c_t\omega_i^2$ is the single motor (1-4) thrust.
- $T_B = \sum_1^4 c_t\omega_i^2$ is the body frame thrust.

- x is the quadcopter's center of mass X coordinate in the global frame.
- $\dot{x}$ is the quadcopter's center of mass X velocity in the global frame.
- $\ddot{x}$ is the quadcopter's center of mass X acceleration in the global frame.
- y is the quadcopter's center of mass Y coordinate in the global frame.
- $\dot{y}$ is the quadcopter's center of mass Y velocity in the global frame.
- $\ddot{y}$ is the quadcopter's center of mass Y acceleration in the global frame.
- z is the quadcopter's center of mass Z coordinate in the global frame.
- $\dot{z}$ is the quadcopter's center of mass Z velocity in the global frame.
- $\ddot{z}$ is the quadcopter's center of mass Z accleration in the global frame.

- ϕ is the roll (about X).
- $\dot{\phi}$ is the angular velocity about the body frame X.
- $\ddot{\phi}$ is the angular acceleration about the body frame X.
- θ is the pitch (about Y).
- $\dot{\theta}$ is the angular velocity about the body frame Y.
- $\ddot{\theta}$ is the angular acceleration about the body frame Y.
- ψ is the roll (about Z).
- $\dot{\psi}$ is angular velocity about the body frame Z.
- $\ddot{\psi}$ is angular acceleration about the body frame Z.

Quadcopter

Mathematical Modeling - Rotational Constants and R-Matrix

- I_{xx} is the moment of inertia of quadcopter about body x axis.
- I_{yy} is the moment of inertia of quadcopter about body y axis.
- Note that $I_{xx} = I_{yy}$ by symmetry.
- I_{zz} is the moment of inertia of quadcopter about body z axis.
- R is the matrix that transforms the body frame to the global frame

First, we introduce the following shorthand:

$$c(\cdot) = \cos(\cdot) \tag{1a}$$
$$s(\cdot) = \sin(\cdot) \tag{1b}$$

$$R = \begin{bmatrix} c(\psi)c(\theta) & c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi) & c(\psi)s(\theta)c(\phi) + s(\psi)s(\phi) \\ s(\psi)c(\theta) & s(\psi)s(\theta)s(\phi) + c(\psi)c(\phi) & s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi) \\ -s(\theta) & c(\theta)s(\psi) & c(\theta)c(\phi) \end{bmatrix} \tag{2}$$

$$x_1 = \begin{bmatrix} x & y & z \end{bmatrix}^T \tag{3a}$$

$$x_2 = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T \tag{3b}$$

$$x_3 = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T \tag{3c}$$

$$x_4 = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T \tag{3d}$$

$$q = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T \tag{3e}$$

$$\dot{x_1} = x_2 \tag{4a}$$

$$\dot{x_2} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \frac{1}{m} RT_B + \frac{1}{m} F_d \tag{4b}$$

$$\dot{x_3} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\theta\sin\phi \end{bmatrix}^{-1} x_4 \tag{4c}$$

$$\dot{x_4} = \begin{bmatrix} \frac{1}{I_{XX}} & 0 & 0 \\ 0 & \frac{1}{I_{YY}} & 0 \\ 0 & 0 & \frac{1}{I_{ZZ}} \end{bmatrix} \left(\begin{bmatrix} l_q c_t(-\omega_2^2 + \omega_4^2) + \dot{\theta}\dot{\psi}(I_{yy} - I_{zz}) \\ l_q c_t(-\omega_1^2 + \omega_3^2) + \dot{\phi}\dot{\psi}(I_{zz} - I_{xx}) \\ \sum_{i=1}^{4} \tau_{m_i} \end{bmatrix} \right) \tag{4d}$$

Assumptions - very small $\dot{\phi}, \dot{\theta}, \dot{\psi}$ and deviations in ϕ, θ

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \\ \dot{z} \\ \ddot{z} \\ \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{F_d}{m} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{F_d}{m} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{F_d}{m} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \\ \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \end{bmatrix} +$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{c_t}{m} & \frac{c_t}{m} & \frac{c_t}{m} & \frac{c_t}{m} & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{-lc_t}{I_{xx}} & 0 & \frac{lc_t}{I_{xx}} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{-lc_t}{I_{yy}} & 0 & \frac{lc_t}{I_{yy}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ b & -b & b & -b & 0 \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \\ g \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \\ \dot{z} \\ \ddot{z} \\ \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ -\frac{F_d}{m} \\ \dot{y} \\ -\frac{F_d}{m} \\ \dot{z} \\ -\frac{F_d}{m} \\ \dot{\phi} \\ \dot{\theta}\dot{\psi}(I_{yy} - I_{zz}) \\ \dot{\theta} \\ \dot{\phi}\dot{\psi}(I_{zz} - I_{xx}) \\ \dot{\psi} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \frac{c_t T_x}{m} & \frac{c_t T_x}{m} & \frac{c_t T_x}{m} & \frac{c_t T_x}{m} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{c_t T_y}{m} & \frac{c_t T_y}{m} & \frac{c_t T_y}{m} & \frac{c_t T_y}{m} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{c_t T_z}{m} & \frac{c_t T_z}{m} & \frac{c_t T_z}{m} & \frac{c_t T_z}{m} & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{-lc_t}{I_{xx}} & 0 & \frac{lc_t}{I_{xx}} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{-lc_t}{I_{yy}} & 0 & \frac{lc_t}{I_{yy}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ b & -b & b & -b & 0 \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \\ g \end{bmatrix} \quad (5)$$

BRADLEY University

Where:

$$T_x = (\sin\phi \sin\psi + \cos\psi \sin\theta \cos\phi) \tag{6a}$$

$$T_y = (-\sin\phi \cos\psi + \sin\psi \sin\theta \cos\phi) \tag{6b}$$

$$T_z = (\cos\psi \cos\phi) \tag{6c}$$

We were able to implement the non-linear model for the quadcopter in MATLAB.

- *Step 1* - Use thrusts to generate lateral acceleration.
- *Step 2* - Use thrusts to generate angular accelerations.
- *Step 3* - Keep a running total of velocity by adding instantaneous accelerations multiplied by sampling time to perform running total integration.
- *Step 4* - Keep a running total of position by adding instantaneous velocity multiplied by sampling time to perform running total integration.
- *Step 5* - Rotate struct of points locating motor mounts, and shift into position by adding position vector, as shown in Equation 9 and presented graphically next.

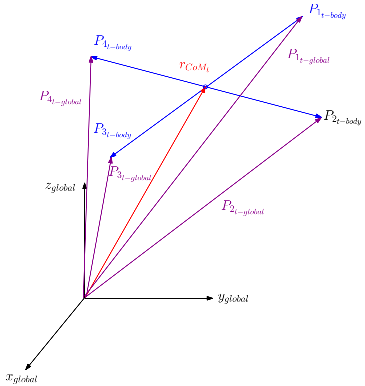

Figure: Vectors locating motor mounts at time t.

$$\mathbf{P} = [x_i \quad y_i \quad z_i]^T \tag{7a}$$

$$\mathbf{r_{CoM_{init}}} = [0 \quad 0 \quad 0]^T \tag{7b}$$

$$\mathbf{P_{1_{init}}} = [\frac{l_a}{2} \quad 0 \quad 0]^T \tag{7c}$$

$$\mathbf{P_{2_{init}}} = [0 \quad \frac{l_a}{2} \quad 0]^T \tag{7d}$$

$$\mathbf{P_{3_{init}}} = [-\frac{l_a}{2} \quad 0 \quad 0]^T \tag{7e}$$

$$\mathbf{P_{4_{init}}} = [0 \quad -\frac{l_a}{2} \quad 0]^T \tag{7f}$$

The location of the motor mounts in the body frame can then be rotated as follows

$$\mathbf{P_{i_{t-body}}} = \mathbf{R_t P_{i_{init}}} \tag{8}$$

And then shifted into the global frame by position vector for the quadcopter's center of mass:

$$\mathbf{P_{i_{t-global}}} = \mathbf{r_{CoM_t}} + \mathbf{P_{i_{t-body}}} \tag{9}$$

For i = 1, 2, 3, 4, where $\mathbf{r_{CoM_t}}$ is the location of the center of the quadcopter at time instant t.

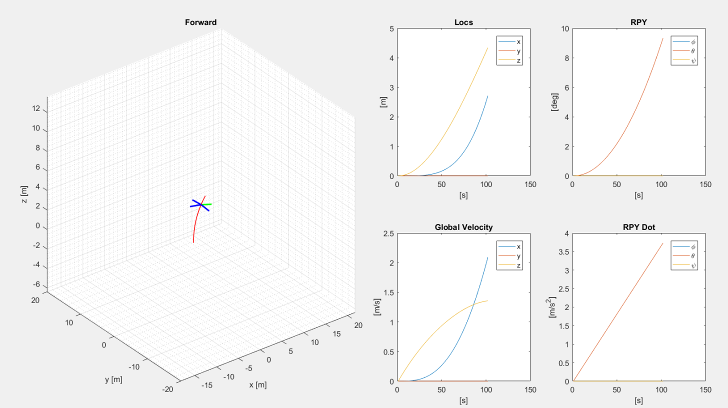

Figure: Quadcopter moving forward.

Quadcopter
MATLAB Simulations

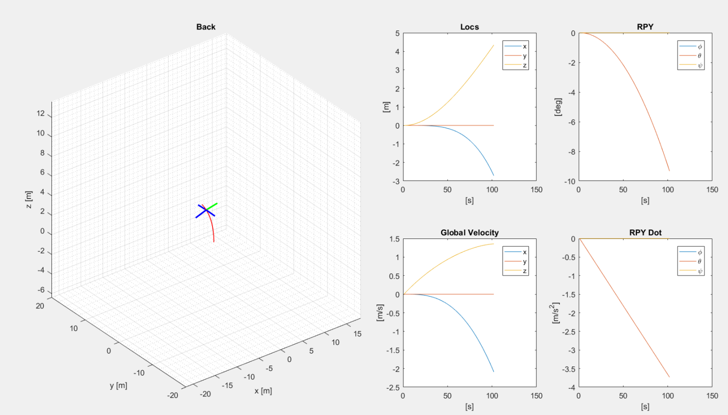

Figure: Quadcopter moving backwards.

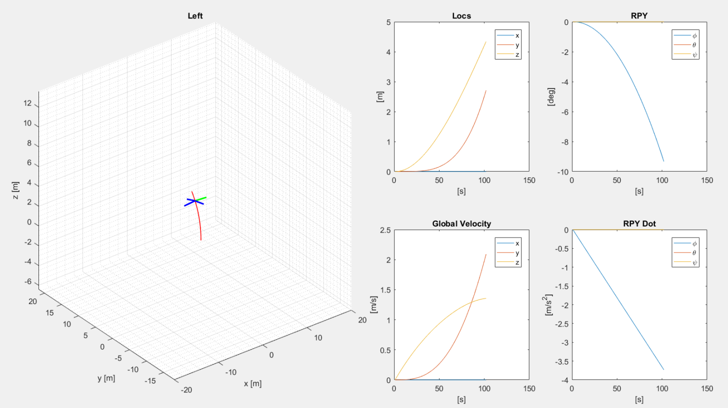

Figure: Quadcopter moving left.

Quadcopter

MATLAB Simulations

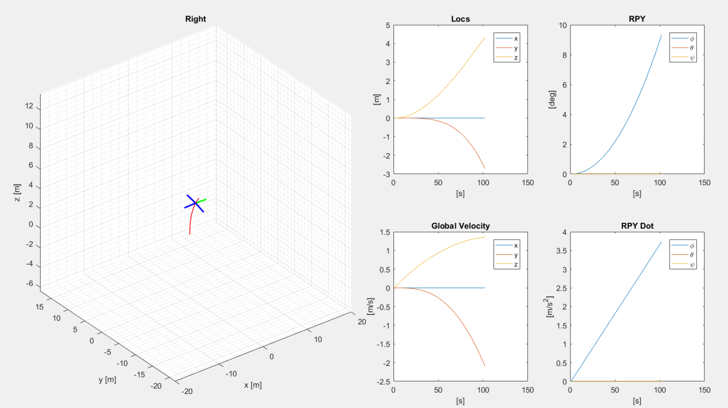

Figure: Quadcopter moving right.

The proposed error model is

$$\dot{\mathbf{e}} = \mathbf{A}\mathbf{e} - \mathbf{B}\mathbf{u} - \mathbf{A}\mathbf{x}^d \tag{10}$$

The error vector will be expressed as:

$$\mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \\ e_9 \\ e_{10} \\ e_{11} \\ e_{12} \end{bmatrix} = \begin{bmatrix} x^d - x \\ y^d - y \\ z^d - z \\ \dot{x}^d - \dot{x} \\ \dot{y}^d - \dot{y} \\ \dot{z}^d - \dot{z} \\ \phi^d - \phi \\ \theta^d - \theta \\ \psi^d - \psi \\ \dot{\phi}^d - \dot{\phi} \\ \dot{\theta}^d - \dot{\theta} \\ \dot{\psi}^d - \dot{\psi} \end{bmatrix} = \mathbf{x}^d - \mathbf{x} \tag{11}$$

Because we want the quadcopter to hover on a point, we assume that ϕ^d, θ^d and ψ^d are 0 as well as $\dot{\phi}^d, \dot{\theta}^d$, and $\dot{\psi}^d$. Further, we can assume $\dot{x}^d, \dot{y}^d$, and $\dot{z}^d$ to be zero as well. Thus the error vector is:

$$\mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \\ e_9 \\ e_{10} \\ e_{11} \\ e_{12} \end{bmatrix} = \begin{bmatrix} x^d - x \\ y^d - y \\ z^d - z \\ -\dot{x} \\ -\dot{y} \\ -\dot{z} \\ -\phi \\ -\theta \\ \psi^d - \psi \\ -\dot{\phi} \\ -\dot{\theta} \\ -\dot{\psi} \end{bmatrix} = \mathbf{x}^d - \mathbf{x} \tag{12}$$

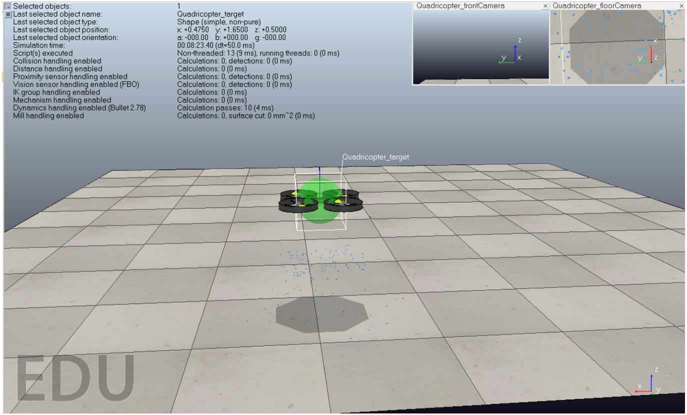

Figure: Quadcopter hovering in V-REP.

Quadcopter

V-REP Simulations

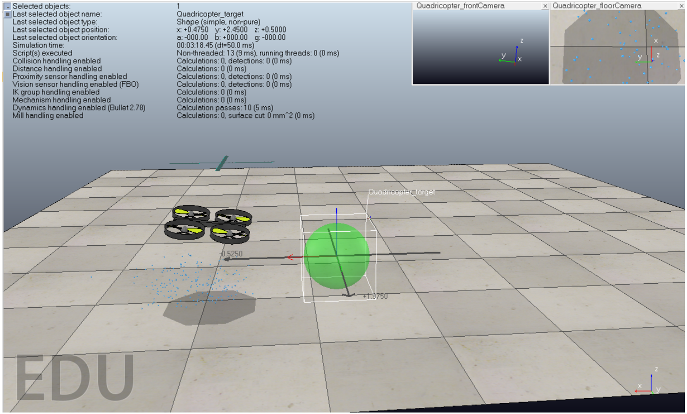

Figure: Quadcopter approaching a point in V-REP.

Preliminary Work

- Researching the derivation of the 2-DOF state-space model
- Researching the proposed approximate dynamic programming algorithm in [7]
- Simulating the proposed approximate dynamic programming algorithm in [7] for the 2-DOF helicopter
- Configuring the Quanser software to operate the Quanser AERO

Complete state-space model of the 2-DOF helicopter was derived.

$$\begin{bmatrix} \dot{\theta} \\ \dot{\psi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{R_p \beta_p}{I_p} & 0 \\ 0 & 0 & 0 & -\frac{R_y \beta_y}{I_y} \end{bmatrix} \begin{bmatrix} \theta \\ \psi \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{K_{p,main}}{I_p} & 0 \\ 0 & \frac{K_{y,tail}}{I_y} \end{bmatrix} \begin{bmatrix} V_p \\ V_y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{G_p}{I_p} - \frac{R_c m_{body} g sin(\theta)}{I_p} \\ -\frac{G_y}{I_y} \end{bmatrix} \tag{13}$$

- θ is the pitch.
- ψ is the yaw.
- V_p is the voltage applied to the main rotor.
- V_y is the voltage applied to the tail rotor.

BRADLEY University

The variables used here are as follows for the Quanser AERO:

- R_p is the distance from the pitch rotor to the fork.
- R_y is the distance from the tail rotor to the fork.
- R_c is the distance from the fork to the center of mass of the body.
- β_p is the damping coefficient associated with the main rotor.
- β_y is the damping coefficient associated with the tail rotor.
- I_p is the rotational inertia of the main rotor.
- I_y is the rotational inertia of the tail rotor.
- $K_{p,main}$ is the thrust constant associated with the main rotor.
- $K_{y,tail}$ is the thrust constant associated with the tail rotor.
- G_p is the nonlinear coupling on the main rotor due to the tail rotor.
- G_y is the nonlinear coupling on the tail rotor due to the main rotor.
- m_{body} is the mass of the body.
- g is the acceleration due gravity.

We then linearized the 2-DOF helicopter state-space model.

Assumptions

- $sin(\theta) \approx \theta$
- $G_p = K_{p,tail} V_y$
- $G_y = K_{y,main} V_p$

$$\begin{bmatrix} \dot{\theta} \\ \dot{\psi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{R_c m_{body} g}{I_p} & 0 & -\frac{R_p \beta_p}{I_p} & 0 \\ 0 & 0 & 0 & -\frac{R_y \beta_y}{I_y} \end{bmatrix} \begin{bmatrix} \theta \\ \psi \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{K_{p,main}}{I_p} & \frac{K_{p,tail}}{I_p} \\ -\frac{K_{y,main}}{I_y} & \frac{K_{y,tail}}{I_y} \end{bmatrix} \begin{bmatrix} V_p \\ V_y \end{bmatrix} \quad (14)$$

Using measurements provided by Quanser, the state-space model for the Quanser AERO becomes Equation 15.

$$\begin{bmatrix} \dot{\theta} \\ \dot{\psi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1.7442 & 0 & -0.3307 & 0 \\ 0 & 0 & 0 & -0.9283 \end{bmatrix} \begin{bmatrix} \theta \\ \psi \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.0512 & 0.0977 \\ -0.1139 & 0.0928 \end{bmatrix} \begin{bmatrix} V_p \\ V_y \end{bmatrix} \quad (15)$$

Our derived state-space model matches the state-space model provided by Quanser within rounding.

- Using the state-space model of the 2-DOF helicopter, we are currently trying to simulate the proposed algorithm in [7].
- The proposed algorithm in [7] uses an actor-critic neural network to find the optimal inputs to minimize the error model.
- The proposed error model is

$$\dot{\mathbf{e}} = \mathbf{Ae} - \mathbf{Bu} - \mathbf{Ax}^d \tag{16}$$

where the error vector is

$$\mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} = \begin{bmatrix} \theta^d - \theta \\ \psi^d - \psi \\ -\dot{\theta} \\ -\dot{\psi} \end{bmatrix} = \mathbf{x}^d - \mathbf{x} \tag{17}$$

- We are currently researching the regulator control problem.
- The initial states are arbitrary, and the desired states are all zero.
- Linear quadratic regulator (LQR) control is often used with the 2-DOF helicopter, so we are using this as a baseline controller.

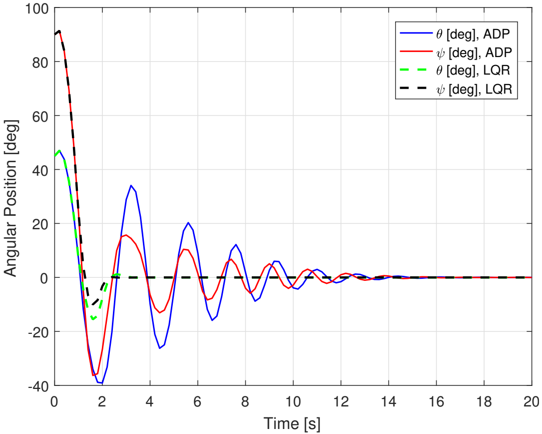

Figure: Angular position using the approximate dynamic programming algorithm and LQR.

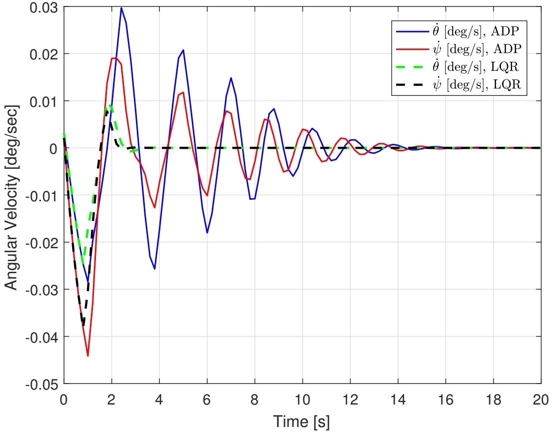

Figure: Angular velocity using the approximate dynamic programming algorithm and LQR.

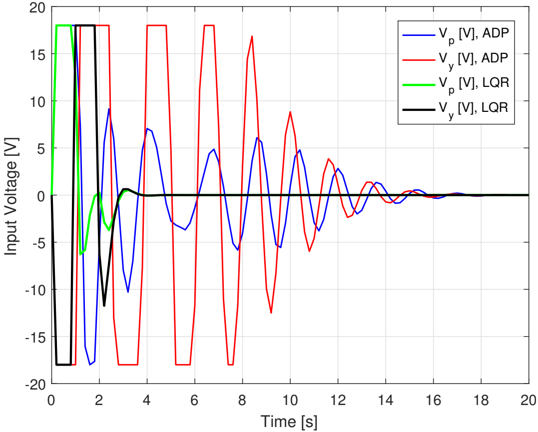

Figure: Motor voltage inputs using the approximate dynamic programming algorithm and LQR.

- In theory, the approximate dynamic programming algorithm should provide a better response than LQR control.
- Further research is needed to determine tuning parameters to make the approximate dynamic programming algorithm better than LQR.

Parts List

- Quanser AERO (Dr. Miah currently has one.)
- Software
  - MATLAB (Purchased for other classes)
  - V-REP (Free)
  - QUARC (Free)
- Raspberry Pi/BeagleBone (Dr. Miah currently has some.)

Anthony

- Quadcopter Mathematical Modeling
- Quadcopter MATLAB Simulations
- Quadcopter V-REP Simulations
- Half-Quadcopter MATLAB Simulations
- Half-Quadcopter V-REP Simulations
- Simulink Integration of Quanser Half-Quadcopter

Andrew

- Helicopter Mathematical Modeling
- Helicopter MATLAB Simulations
- Quanser Software Setup
- Simulink Integration of Baseline Controllers
- Simulink Integration of Helicopter
- Single-Board Microcomputer Integration

Schedule for Completion

Fall 2017

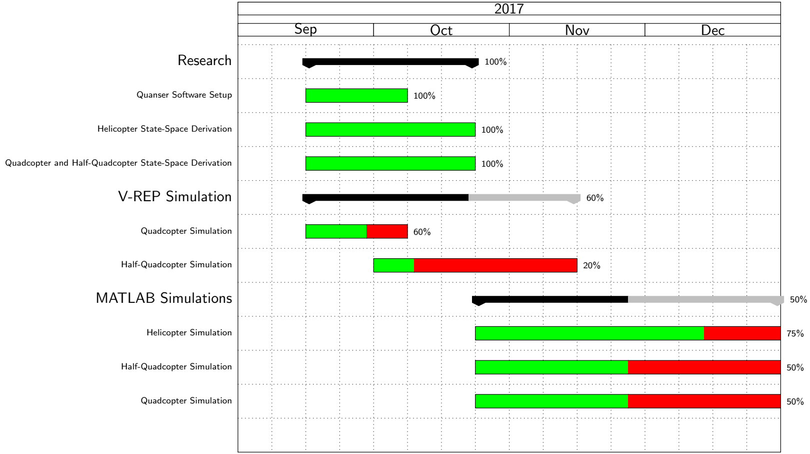

Figure: Fall 2017.

Schedule for Completion
Spring 2018

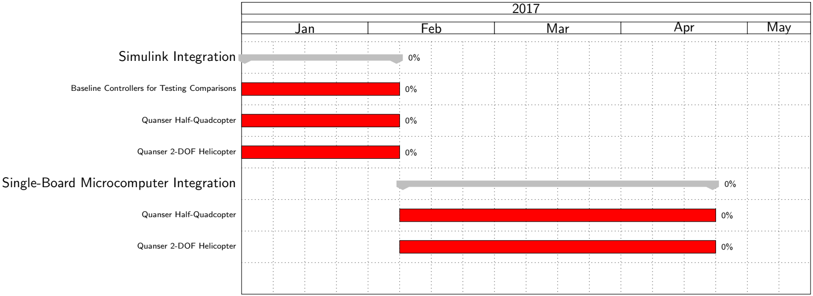

Figure: Spring 2018.

Future Directions

- Research the approximate dynamic programming algorithm in order to make it better than LQR control
- Finish both the V-REP and MATLAB simulations
- Implement the tuned algorithm into Simulink to control the Quanser AERO (next semester)
- Implement the single-board microcomputer as a cost-effective solution (next semester)

References I

Q. Ahmed et al. “2-sliding mode based robust control for 2-DOF helicopter”. In: *2010 11th International Workshop on Variable Structure Systems (VSS)*. June 2010, pp. 481–486. DOI: 10.1109/VSS.2010.5544531.

W. Gao and Z. P. Jiang. “Data-driven adaptive optimal output-feedback control of a 2-DOF helicopter”. In: *2016 American Control Conference (ACC)*. July 2016, pp. 2512–2517. DOI: 10.1109/ACC.2016.7525294.

Eswarmurthi Gopalakrishnan. “QUADCOPTER FLIGHT MECHANICS MODEL AND CONTROL ALGORITHMS”. In: *Czech Technical University - Department of Control Engineering*. May 2016. DOI: 10.1109/CDC.2011.6160521.

References II

- M. Hernandez-Gonzalez, A.Y. Alanis, and E.A. Hernandez-Vargas. "Decentralized discrete-time neural control for a Quanser 2-DOF helicopter". In: *Applied Soft Computing*. Feb. 2012, pp. 2462–2469. DOI: 10.1016/j.asoc.2012.02.016.
- E. Kayacan and M.A. Khanesar. "Recurrent interval type-2 fuzzy control of 2-DOF helicopter with finite time training algorithm". In: *IFAC-PapersOnLine*. July 2016, pp. 293–299. DOI: 10.1016/j.ifacol.2016.07.977.
- Teppo Luukkonen. "Modelling and control of quadcopter". In: *2014 American Control Conference*. Vol. Aalto School of Science. June 2011.
- S. Miah. "Value iteration based approximate dynamic programming for mobile robot trajectory with persistent inputs". In: ().

BRADLEY University

References III

K. Subbarao, P. Nuthi., and G. Atmeh. "Reinforcement learning based computational adaptive optimal control and system identification for linear systems". In: *Annual Reviews in Control*. Sept. 2016, pp. 319–331. DOI: 10.1016/j.arcontrol.2016.09.021.

R.G. Subramanian and V.K. Elumalai. "Robust MRAC augmented baseline LQR for tracking control of 2-DOF helicopter". In: *Robotics and Autonomous Systems*. Aug. 2016, pp. 70–77. DOI: 10.1016/j.robot.2016.08.004.

J.H. Moon W. Chang and H.J. Lee. "Fuzzy Model-Based Output-Tracking Control for 2 Degree-of-Freedom Helicopter". English. In: *Journal of Electrical Engineering Technology* 12.00.1 (2017). Quanser product(s): 2 DOF Helicopter, pp. 1921–1928. ISSN: 1975-0102. URL: http://www.jeet.or.kr/LTKPSWeb/uploadfiles/be/201705/290520170957344003750.pdf.

BRADLEY University