Autonomous Hover Quad Rotor

In conjunction with the Distributed [Multi-Vehicle] Space System Lab

MS5

Andrew Nelson
Justin Palm
Andy Bradford

Overview

- Customer's Requirements (MS1)
- System Modeling (MS2 – MS3)
  - Plant
  - Controller
  - Simulation
- Implementation (MS4)
  - Hardware
  - Software

The Customer

- The Distributed Space Systems Lab (DSSL) is interested in controls such as:
  guidance
  navigation
  control of single/multi platform aerial systems
  (quadrotor that has only flown under remote control)

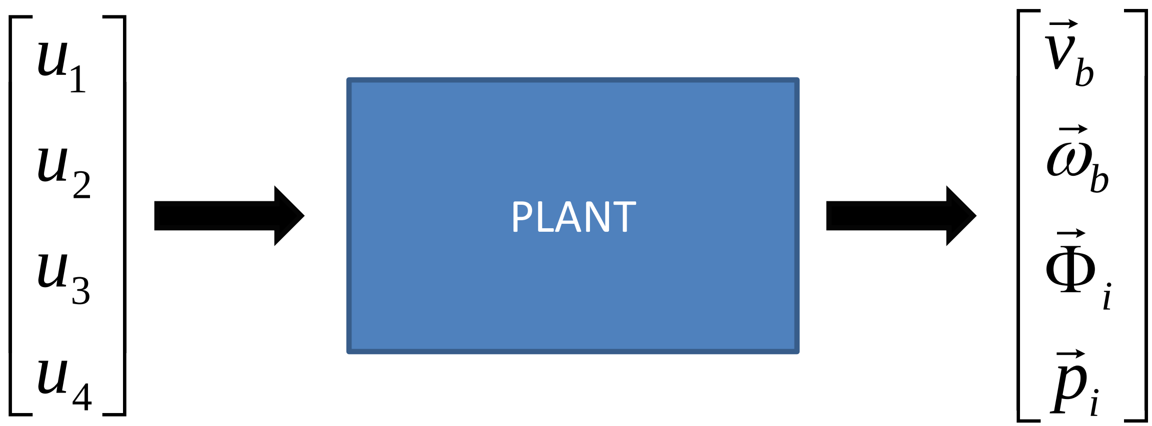
- The DSSL requirement for this project: hover and position control

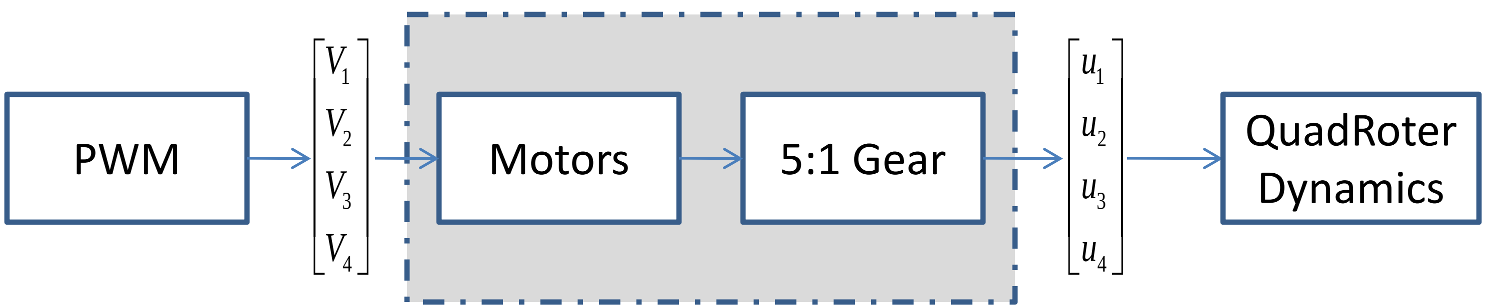
System Modeling

- Plant Identification (MS2)

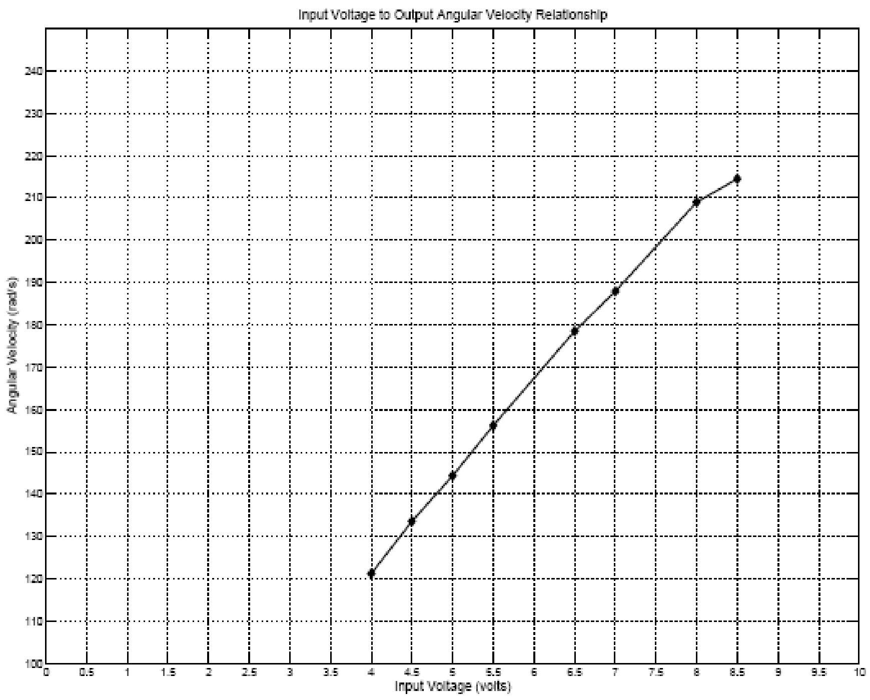
- Controller Design (MS3)

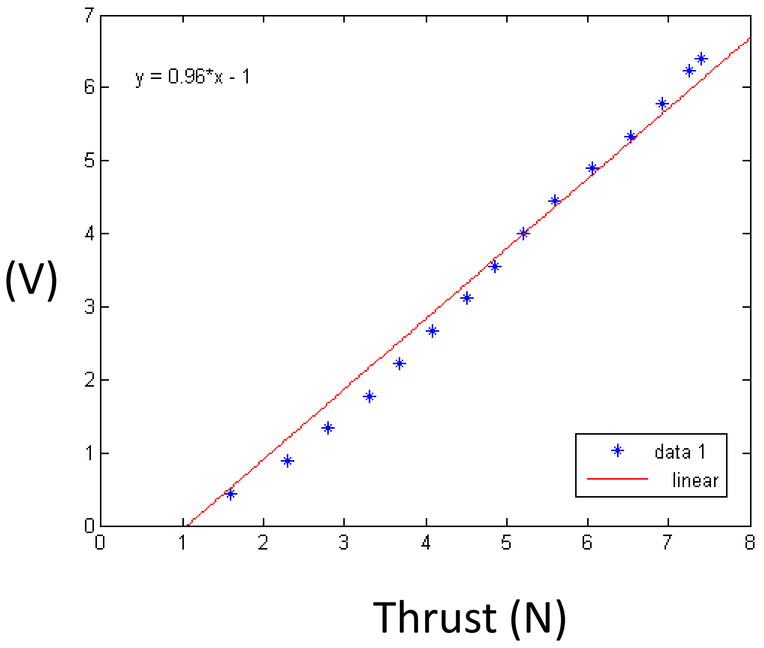
Plant Identification (MS2)





Angular Velocity and Thrust Curves*





*Verification in Progress

Linearized State-Space Equations

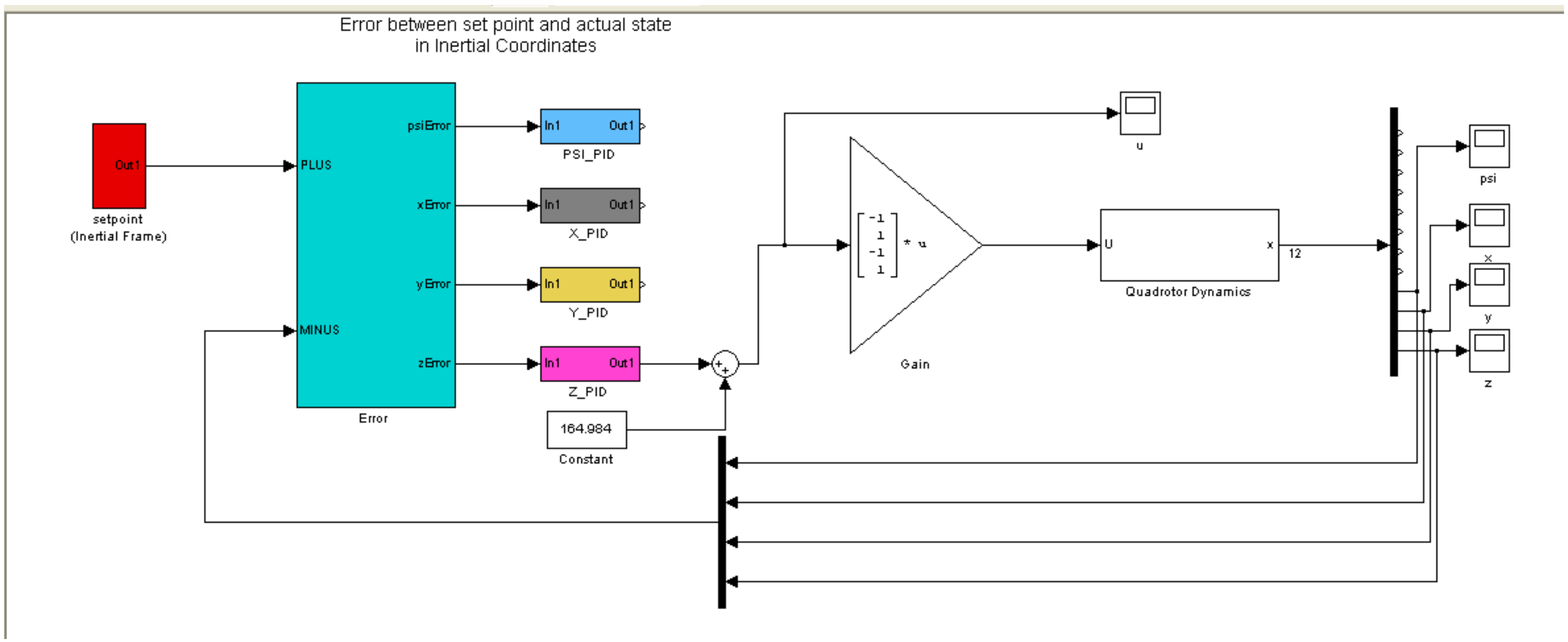
$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9.81 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -9.81 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ .0297 & -.0297 & .0297 & -.0297 \\ 0 & -.5066 & 0 & .5066 \\ -.4954 & 0 & .4954 & 0 \\ -.2470 & -.2470 & -.2470 & -.2470 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$D = 0$$

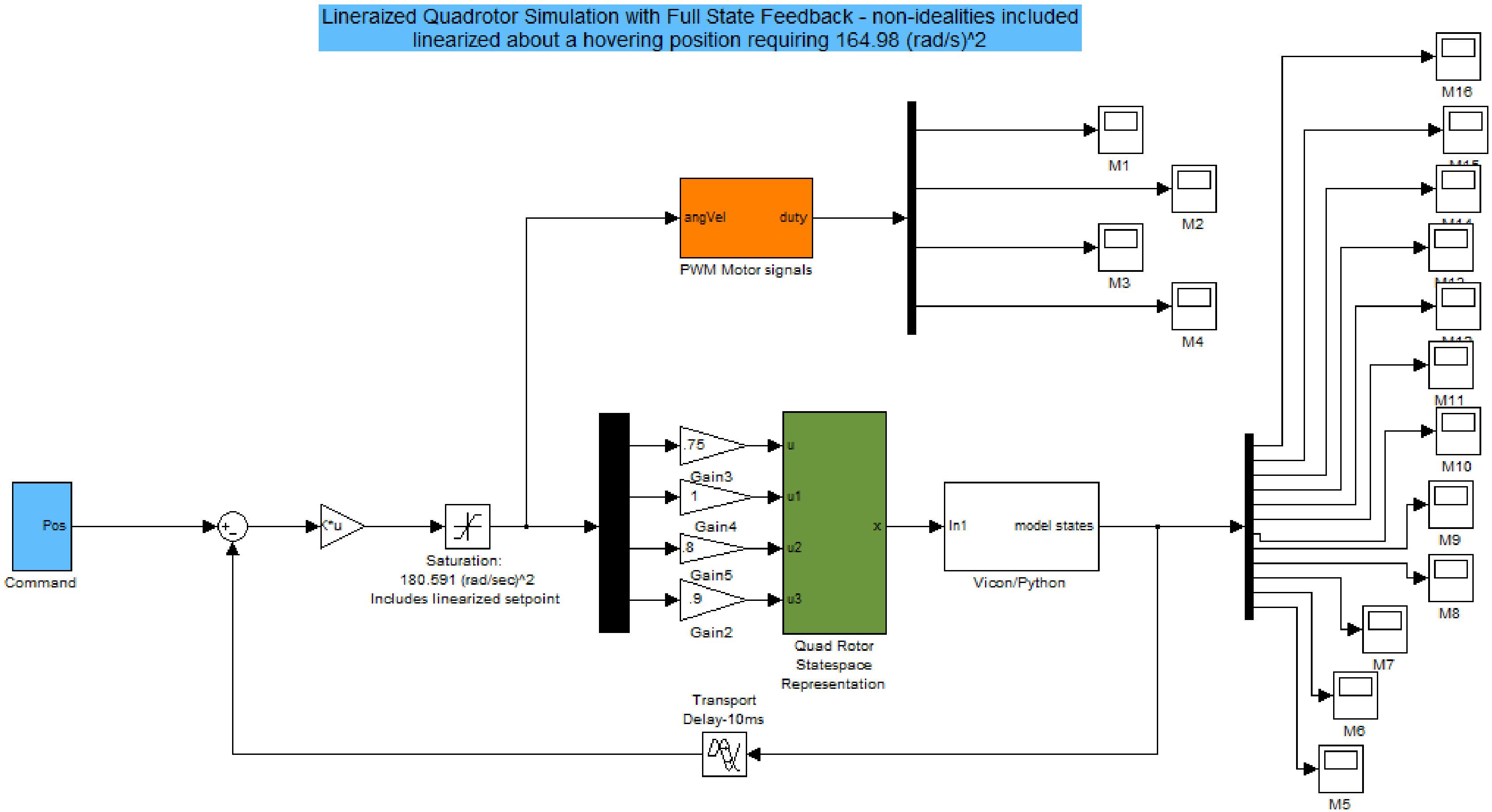
Controller Design (1st attempt)



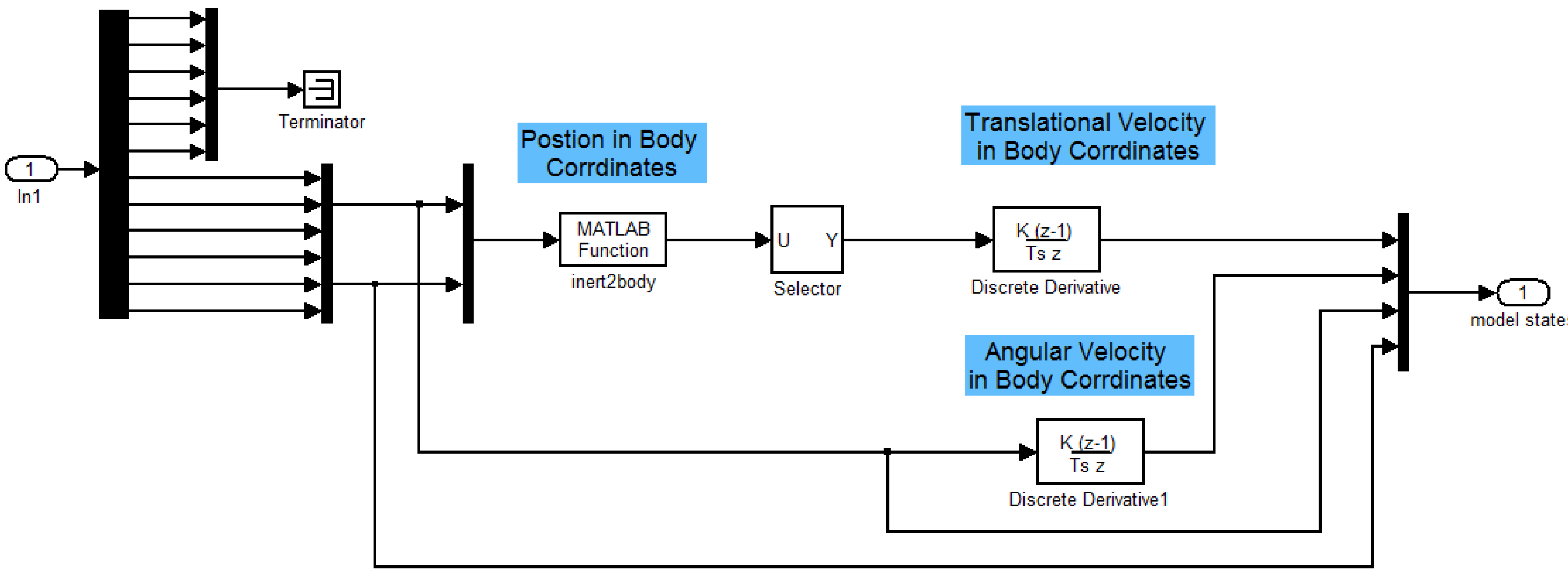
Controller Design (LQR)

- Used Matlab to calculate the Gain Matrix for Proportional Control
- lqrd(A,B,eye(12),0.1*eye(4),1/120)
  - Discrete time linear quadratic regulator
  - Minimzes cost function that balances control effort with the state error

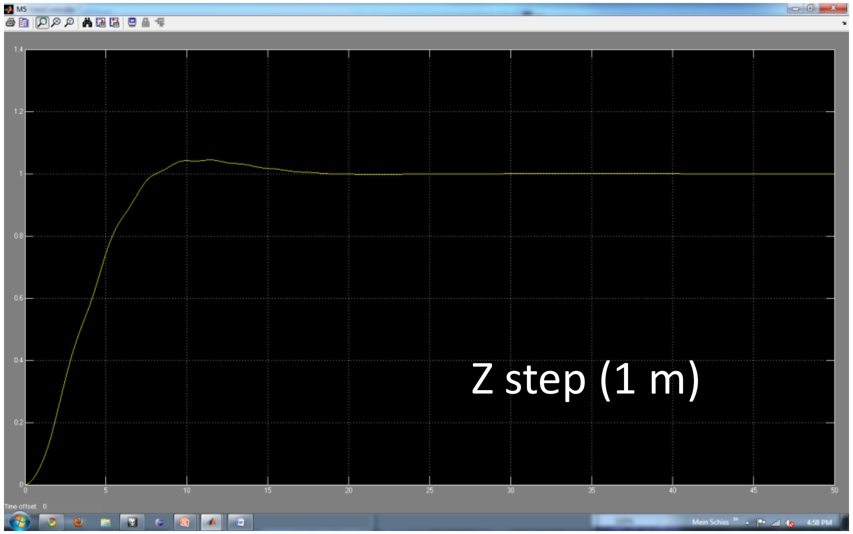
Matlab Simulation

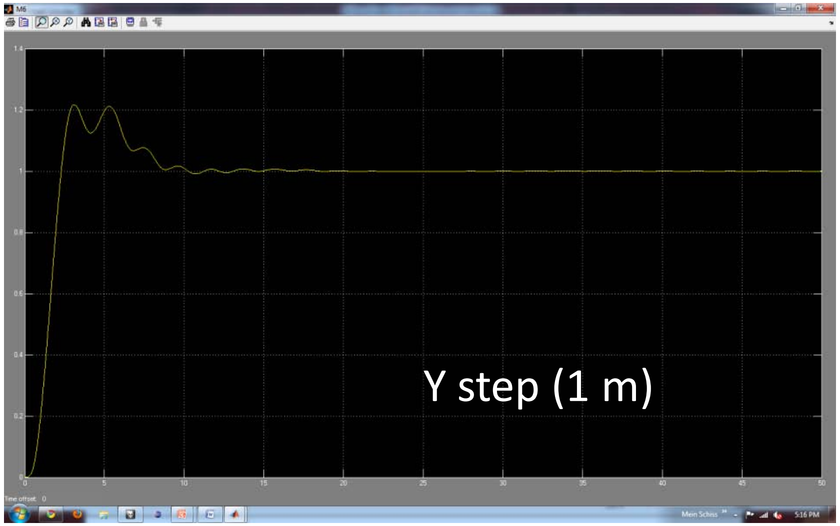


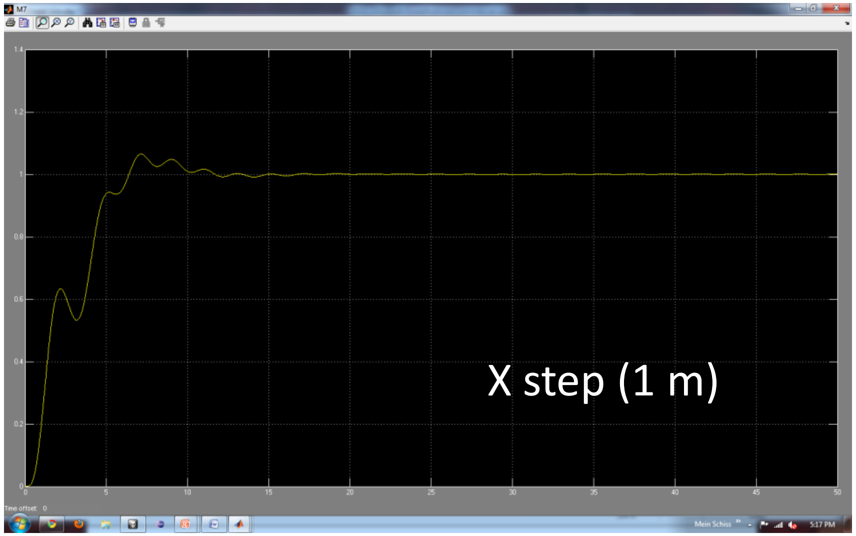
State Estimator



Simulation Results







Implementation

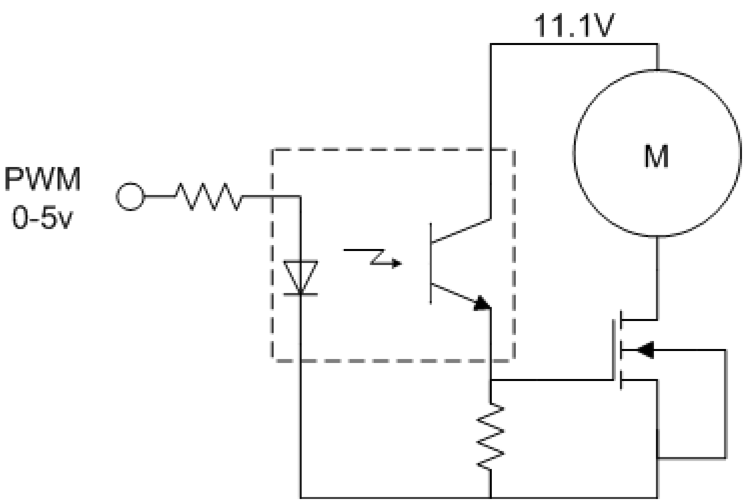
- Hardware
  - Motor Control & Power
  - Communication
- Software
  - Python
  - Vicon API (limited)

Hardware- Motor Control & Power

MOSFETs – MTP3055
- •12A max current
- •0.15 Ω on Resistance
- •Ideal for switching

Opto-Isolator – LTV-847 (Lite-On)
- •Quad-package
- • 4µs rise-time (25x base freq.)
- •Cheap! ($0.75)

•Battery –Li-Polymer 3 Cell
- •2100 mAhr (~40 min of flight)



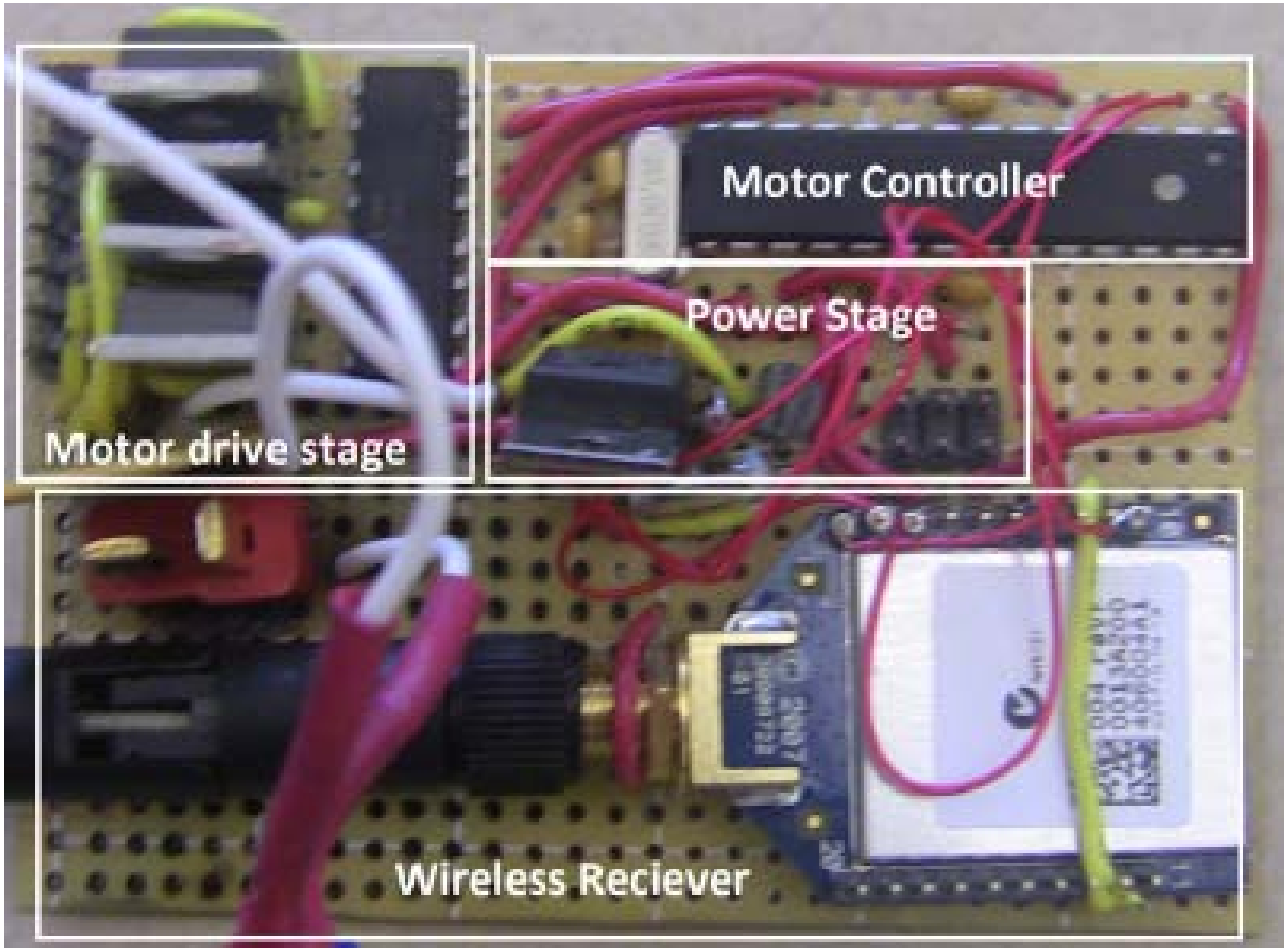
Hardware - Communication

- Vicon Camera System
  - “0.1mm” accuracy
  - 120fps
  - Expensive – DSSL lab already has

- Laptop Running Matlab
- ZigBee Wireless
  - 900MHz
  - Easy to Implement
  - High Baud Rate (100kBaud)

- Network Link
- Atmel ATMEGA328 microcontroller
  - Familiarity
  - Relatively cheap
  - Has all the functionality
    - PWM (we’re using 10Khz)
    - Analog in
    - Already had programmer and working development Environment

PC Board Layout



Old Software Plan

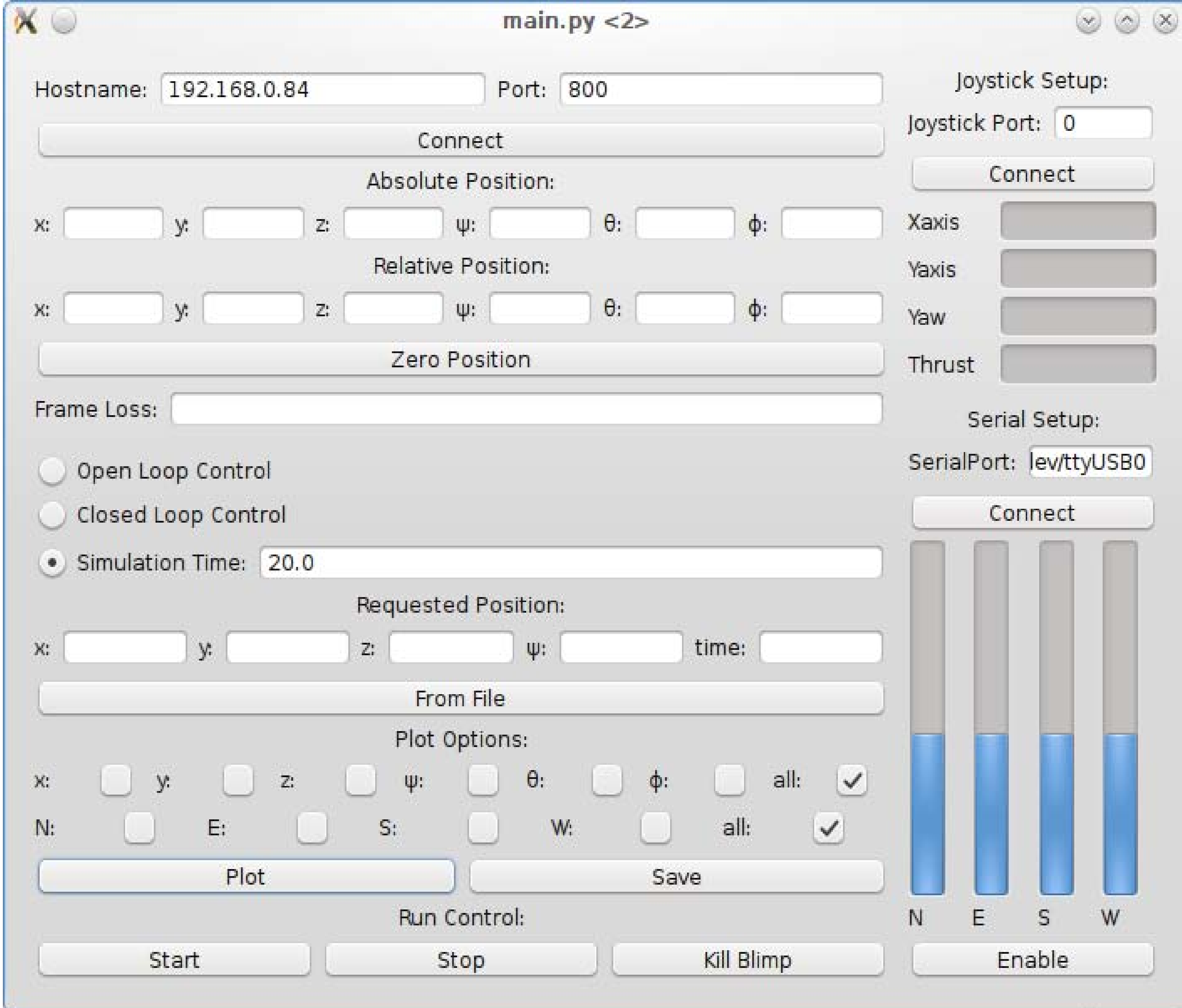
- Use Simulink for feedback loop
  - Allows for rapid prototyping
  - Has built in simulation to verify implementation
- Send motor commands over serial link
- Have Simulink pull data from Vicon
  - Required writing custom S-functions
  - No easy way to trigger loop when data comes in
- No easy way to stop quadrotor

New Software Requirements

- Should be easy to implement and make changes
- Ability to verify implementation
- Real time diagnostics and data logging capabilities
- Have a safe way of stopping the quadrotor

Current Software

- Written in Python
- Scipy and Numpy for numerical computation
  - Gives the power of Matlab to Python
  - Able to import and export data as .mat files
- GUI written with the PyQT4 interface
- PyQT4 network stack used for Vicon interface
- Feedback control loop uses dedicated thread
- Fully cross platform compatible


main.py <2>
Hostname: 192.168.0.84
Port: 800
Connect
Absolute Position:
x: y: z: ψ: θ: ϕ:
Relative Position:
x: y: z: ψ: θ: ϕ:
Zero Position
Frame Loss:
Open Loop Control
Closed Loop Control
Simulation Time: 20.0
Requested Position:
x: y: z: ψ: time:
From File
Plot Options:
x: y: z: ψ: θ: ϕ: all:
N: E: S: W: all:
Plot
Save
Run Control:
Start
Stop
Kill Blimp
Joystick Setup:
Joystick Port: 0
Connect
Xaxis
Yaxis
Yaw
Thrust
Serial Setup:
SerialPort: lev/ttyUSB0
Connect
N E S W
Enable

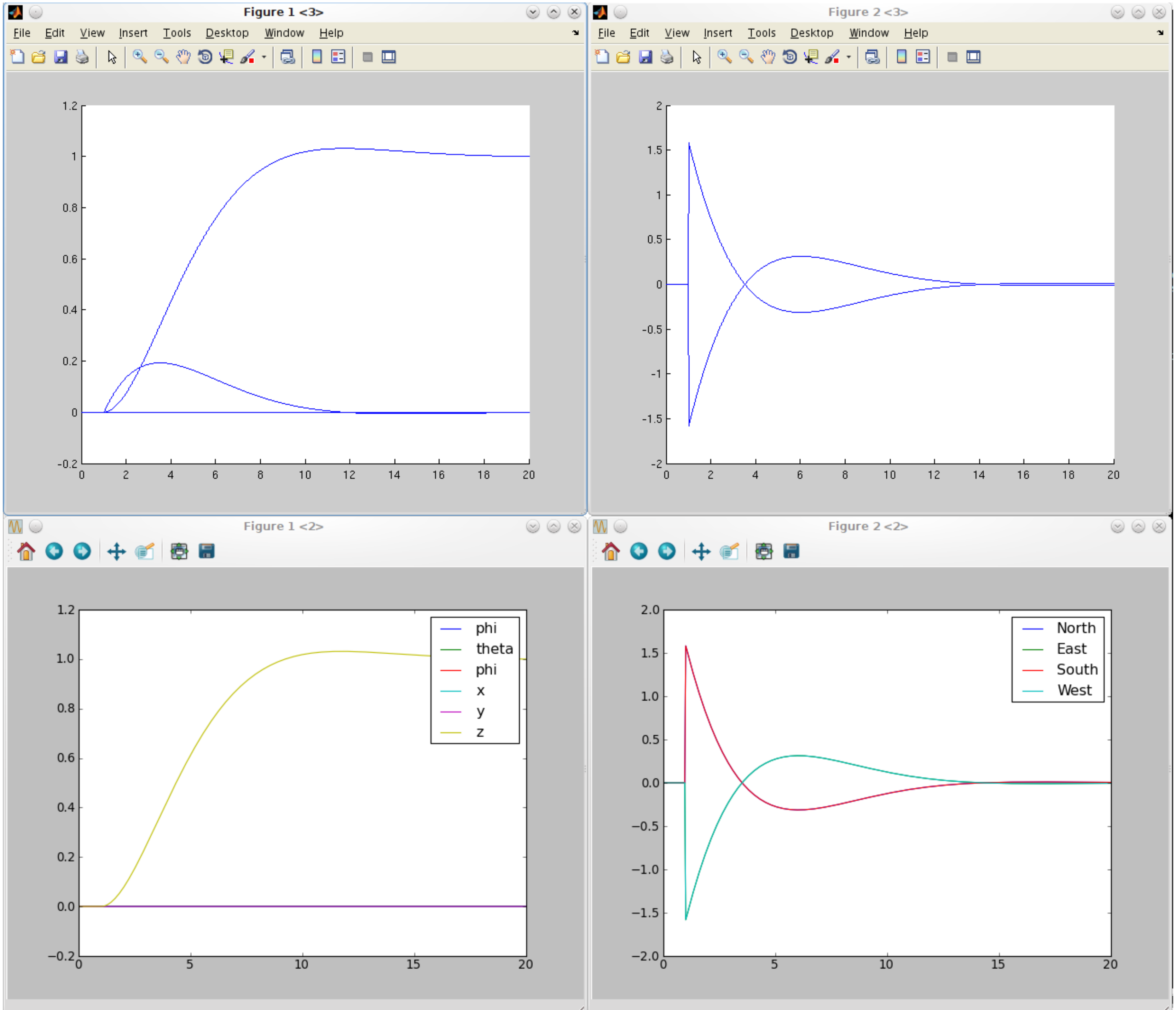


Figure 1 <3>
File Edit View Insert Tools Desktop Window Help
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0 2 4 6 8 10 12 14 16 18 20
Figure 2 <3>
File Edit View Insert Tools Desktop Window Help
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0 2 4 6 8 10 12 14 16 18 20
Figure 1 <2>
phi
theta
phi
x
y
z
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0 5 10 15 20
Figure 2 <2>
North
East
South
West
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0 5 10 15 20

Bibliography

B. Heemstra. (2010) *Linear Quadratic Methods Applied to Quadrotor Control.* Unpublished Masters thesis. University of Washington.

C. Balas. (2007) *Modeling and Linear Control of a Quadrotor*. Unpublished Masters thesis. Cranfield University. https://dspace.lib.cranfield.ac.uk/bitstream/1826/2417/1/Modelling%20and%20Linear%20Control%20of%20a%20Quadrotor.pdf

S. Bouabdallah, A. Noth, and R. Siegwart, “PID vs LQ control techniques applied to an indoor micro quadrotor”, 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings, vol. 3, pp. 1-6.

P. Castillo, A. Dzul, and R. Lozano, “Real-Time Stabilization and Tracking of a Four-Rotor Mini Rotorcraft”, IEEE Transactions on Control Systems Technology, Vol 12, No 4, July, 2004.

McKerrow, P. (2004), "Modelling the Draganflyer four rotor helicopter", *2004 IEEE International Conference on Robotics and Automation, April 2004, New* Orleans, pp. 3596.