DIGITAL CONTROLLER FOR A SMALL VTOL UAV

Gregory Kravit 6.111 Fall 2014 Final Project

Project Checklist

I. Goals

A. Quadcopter Exhibits Stabilized Hover

Pass Criteria: The quadcopter is able to maintain a stabilized hover without crashing in the lab environment

B. Quadcopter Can Hold Selected Altitude

Pass Criteria: The quadcopter is able to hold a desired height in the stabilized hover without crashing in the lab environment.

II. Core Functionality

A. Controller Design

1. Quadrotor dynamics model

Pass Criteria: A usable dynamics model of the quadrotor will be researched and derived in order to accurately model the quadrotor in simulation for developing the controller.

2. Derive PID controller input equations

Pass Criteria: A PID controller scheme will be designed to generate inputs that will stabilize the quadrotor dynamics model and later be deployed on the real physical quadrotor.

3. MATLAB simulation and Simulink models

Pass Criteria: MATLAB simulations and Simulink block diagrams will be written to model the expected responses of the system and test the system in simulation before testing the actual quadrotor.

4. Estimate PID parameters

Pass Criteria: PID parameters will be derived through simulation tuning or optimization before testing on the physical quadcopter. The PID parameters will then be refined to correctly stabilize the physical quadrotor.

B. Flight Controller Module

1. Arithmetic modules/PID Control Logic

Pass Criteria: A series of arithmetic modules will perform the multiplication and trigonometry necessary to compute the control inputs (which is the desired RPM^2) for each motor. Testing will be performed in ModelSim.

2. Convert RPM^2 setting to Throttle Setting

Pass Criteria: The module will use a lookup table to convert the RPM^2 to an 8 bit throttle output that is in the range of 0-255. Testing will be performed in ModelSim.

C. Motor Controller Module

1. Convert Throttle Setting to Pulse Width Signal Length

Pass Criteria: The module will use a lookup table to convert the 8 bit throttle setting to pulse width time that is in the range of 1080 μs to 1860 μs. Testing will be performed in ModelSim.

2. Generate Pulse Width Signal

Pass Criteria: The module will take the pulse width signal length time and emit a PWM signal to the desired servo pin. Testing will demonstrate the operation of a motor at various throttle settings.

D. IMU Sensor Module

1. I2C Master Module

Pass Criteria: A master I2C module will be implemented that will operate at the correct baud rate to retrieve information from the desired register address of the MPU-6050. Testing will be performed in ModelSim using an I2C slave module from opencores.org.

2. Initialization of DSP and Setup of MPU-6050

Pass Criteria: Reverse-engineered Arduino packages will be translated to synthesizable Verilog. The module will be able to setup the MPU-6050 and initialize the digital signal processing (DSP) unit. Testing will be performed on the LabKit in order to use the hex displays for visually checking correct register and data outputs.

3. Retrieve sensor packets from MPU-6050

Pass Criteria: The I2C module will read the 42 bit sensor data packet from the FIFO queue on the chip. Testing will be performed on the LabKit in order to use the hex displays for visually checking that the correct data is returned.

4. Convert quaternions to angles

Pass Criteria: The quaternions retrieved from the 42 bit sensor data packet will be converted to roll, pitch and yaw angles for use by the flight control module. Testing will be performed in ModelSim.

E. Ultrasonic Altitude Ranger Module

1. Trigger Ultrasonic Pulse and Receive Echo

Pass Criteria: This module will initiate the ultrasonic signals to measure distance from an object information. This module will be tested at different calibrated heights to make sure the reader is operating properly.

2. Counter determines length of echo signal in microseconds

Pass Criteria: The UAR counter correctly determines the length of the echo signal in microseconds. Testing and verification will be performed in ModelSim.

III. Interfaces

A. System Level State Machine

Pass Criteria: The system correctly moves through each state in the system as expected. Testing will be performed onboard the quadcopter before first flight.

B. Ground Control Station

Pass Criteria: A system or framework needs to be able to communicate with the Mojo FPGA for desired inputs. Testing will be needed to have the FPGA connected to the ground station and correctly receive desired inputs.

C. Build Quadcopter Kit

Pass Criteria: All components are correctly built, connectors for power and ESCs are soldered on, and the quadcopter is physically ready to fly.

D. Integrate Sensors onto the Quadcopter and FPGA

Pass Criteria: Sensor breakout boards are attached properly onboard the quadcopter.

E. Pipeline sensor state data to flight controller

Pass Criteria: Flight controller can read the sensor state data despite operating at different response rates (400 compared to ~200 Hz and ~30 Hz for the sensors).