VERSION 1.0

OCTOBER 28, 2014

ROBOTIC PLANT MEASUREMENT SYSTEM

DESIGN DOCUMENT

DYLAN GRANSEE

ROBERT LARSEN

ALBERTO DI MARTINO

IAN MCINERNEY

AARON PEDERSON

ROHIT ZAMBRE

FENGXING ZHU

OUTLINE

TERMINOLOGY

DEFINITIONS

ACTUATOR

A motor or some means of providing force into a system.

GIMBAL

A pivoted support that allows the rotation of an object about a single axis. In the context of this project, a gimbal is a mechanical structure that allows the sensing and actuation of the movement of the instrumentation arm.

INERTIAL MEASUREMENT SYSTEM

A sensor system consisting of a gyroscope, accelerometer and magnetometer to provide data about the current position of a physical system.

KALMAN FILTER/SENSOR FUSION

An algorithm to combine current sensor measurements (with sensor noise) with past system states to create an estimate of the current system state.

MIXING MATRIX

An algorithm to combine a set of controller outputs (pitch, roll, yaw, altitude, etc..) into commands to send to individual actuators.

PID CONTROLLER

Proportional Integral Derivative Controller, mechanism used to calculate the a system input based upon the error between the target position and the measured position.

- *Proportional (P)*: Setting that controls how quickly the system will move from the current position to the target position using the current error.
- *Integral (I)*: Setting that uses the past system error to force the system steady-state error to zero.
- *Derivative (D)*: Setting that uses the predicted system error to limit the rate-of-change of the system error.

ACRONYMNS

ESC

Electronic Speed Controller

FPGA

Field-programmable gate array

GUI

Graphical User interface

MICROCART

Microcontroller Controlled Aerial Robotic Team

PCB

Printed Circuit Board

PPM

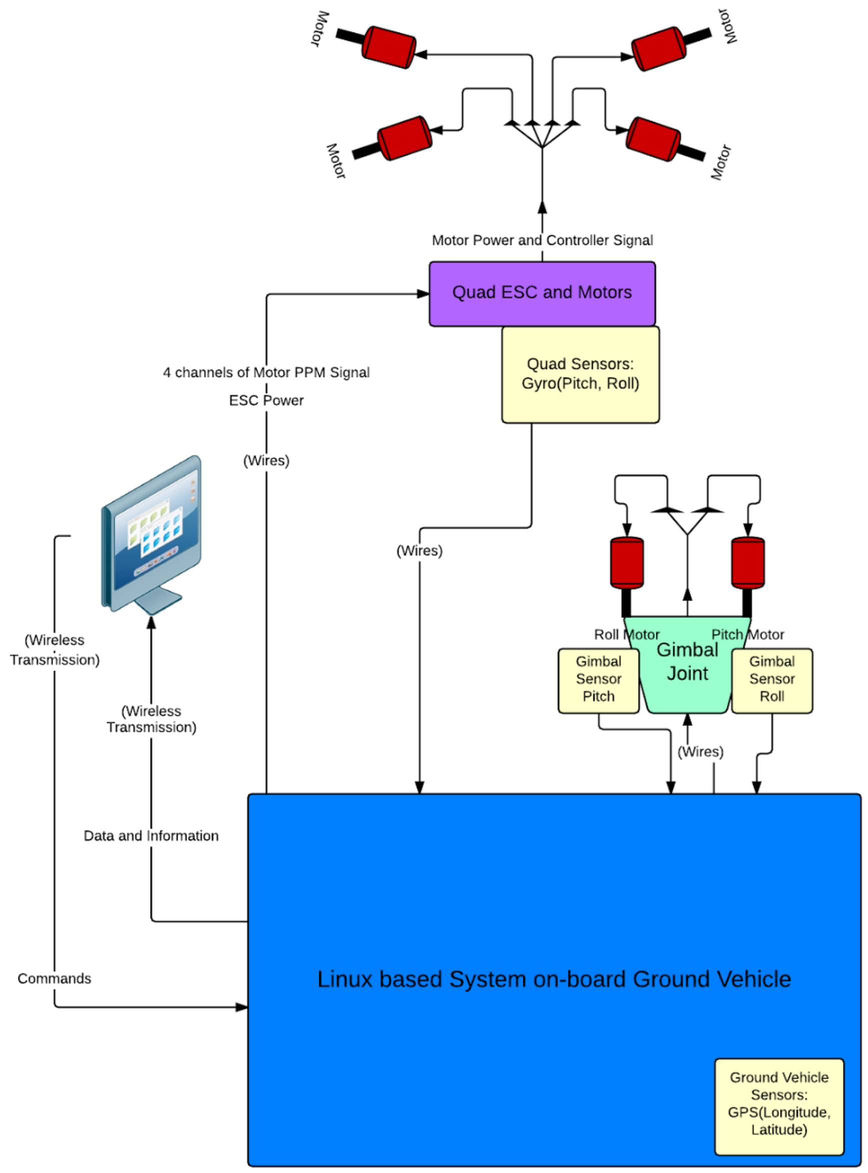
Pulse Position Modulation

VHDL

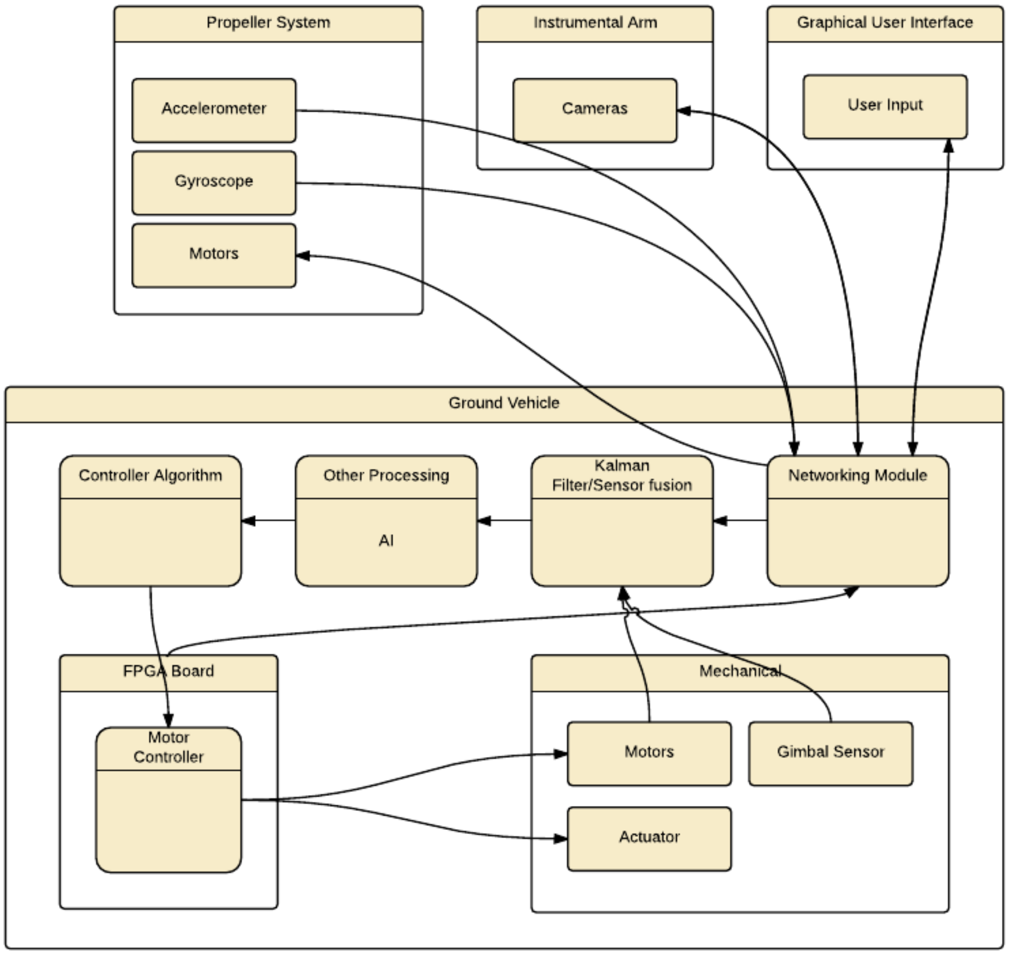
VHSIC Hardware Description Language

SYSTEM DESIGN

BLOCK DIAGRAM



SYSTEM DIAGRAM



SYSTEM REQUIREMENTS

FUNCTIONAL

The high-speed camera system is required for the current MicroCART system since the instrumentation arm controller needs to receive the data from the camera system and process it quickly to generate an adequate response. Eventually an independent sensing system (Gimbal and Inertial Measurement Unit, or IMU) would be installed on the ground robot to transmit data to the instrumentation arm controller.

NON-FUNCTIONAL

- The GUI interface should be visually appealing to use and have multiple options to control the system.

- The entire system must be reliable to perform accurate measurements and transmit signals without considerable time delay.

- Measurement must be taken to ensure the safety of those present during the operation of the system.

SYSTEM INTEGRATION

The hardware and software systems interact very closely and are highly dependent on each other. In the current system configuration, the position and orientation of the propeller system are tracked by the camera system. The tracked data is sent to the Linux machine, which hosts the GUI, via WiFi. The GUI processes the tracked data and input signals from the user (if any) and uses the PID controller to generate the signal that needs to be sent to the actuators. These output signals are then sent to the FPGA which converts them into PPM signals used for transmission. These PPM signals are transmitted to the receiver connected to the propeller system via a RC transmitter. In the final implementation of the system, all the computation currently located on the Linux machine will be shifted to the ground robot. A combination of gimbal sensing, GPS positioning and IMU measurements will be used to track the orientation of the inverted pendulum.

DETAILED SYSTEM

HARDWARE SYSTEM

GROUND ROBOT

The ground robot serves as the main processing platform for the system. It contains a linux based system that handles the interpretation of data from various sensors contained within the system. Using the data, it calculates the required control needed in order to maintain the state of the system based on user input (commands to move). The robot contains two subcomponents: a processing system and a mechanical system.

a) **Processing System**
The processing system consists of the PCB boards for handling input from sensors and computing response. Its primary components are a PC board running linux, an FPGA board used for motor control and sensor input, and a battery protection board. The system will communicate with the GUI for taking in a specific set of user commands.
Inputs: Gimbal sensor data, User commands, Positioning data
Outputs: Control commands, Mixing Matrix PPM

b) **Mechanical System**
The mechanical system of the robot consists of wheels for motion and motors to actuate a gimbal (on which the instrumentation Arm is mounted). The gimbal component senses its position and gives that feedback to the processing system, to be factored in to the control calculations.
Inputs: Control commands
Outputs: Gimbal sensor data

INSTRUMENTATION ARM

The instrumentation arm serves as the device for recording measurements of the experimental crops. The length of the arm may be anywhere between 10 and 14 feet (including ground robot) in real life applications; however, due to limitations in lab, the system must be limited to a length around 6 to 7 feet. The arm is attached to the ground robot via an actuated gimbal and has the system of propellers attached directly to the top.
Inputs: User commands
Outputs: Imaging data

SYSTEM OF PROPELLERS

The propeller system serves to control the position of the Instrumentation Arm and the data logging components (camera, temperature sensor, etc) of the system by actuating the top of the arm. This unit is composed by 4 individual DJI 940 KV propellers, that are being powered by the main power source located at the ground robot.
Inputs: Mixing Matrix PPM
Outputs: Positioning data.

MOUNT OF PROPELLER SYSTEM

The propeller system must be mounted on a frame. This frame is the DJI F450 quadcopter frame. This frame must be modified to fulfill the needs of our system. The frame will be fixed to the instrumentation arm by a mount. In addition to that, the motors must be mounted at an angle of 90 degrees to enable sufficient actuation. The drawing of the frame and add-on parts are below.

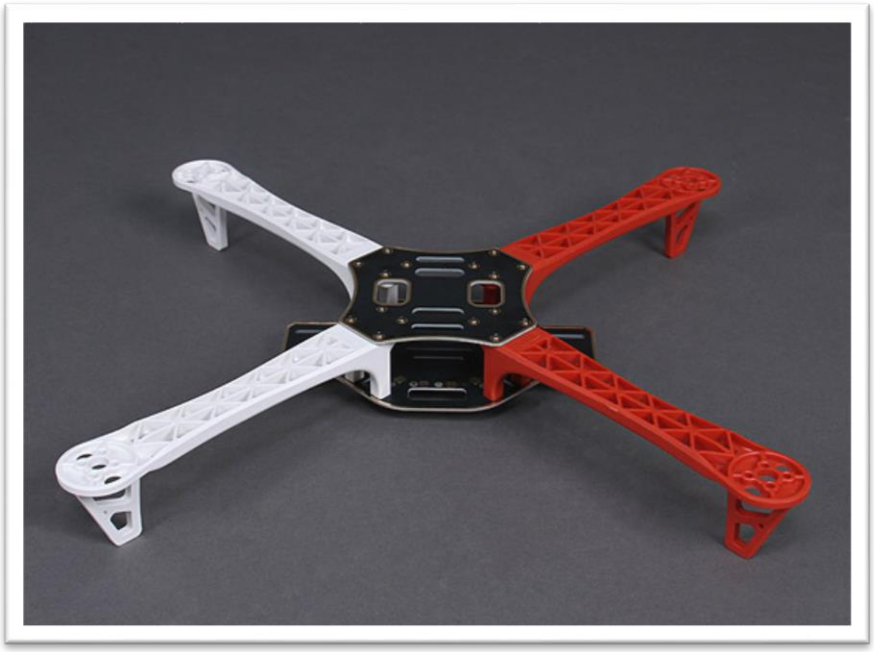


FIGURE 1: DJI F450 FRAME

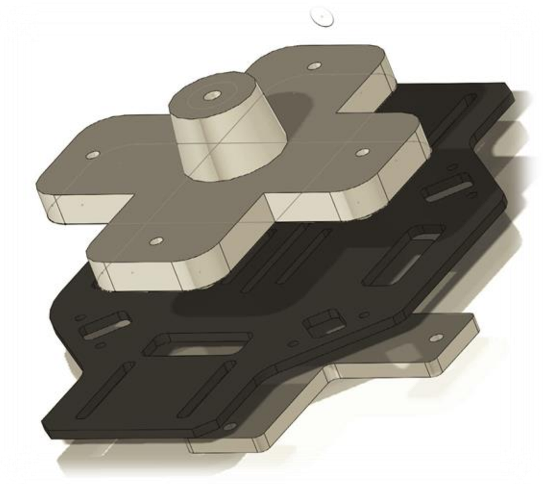


FIGURE 2: BOTTOM MOUNT FOR POLE

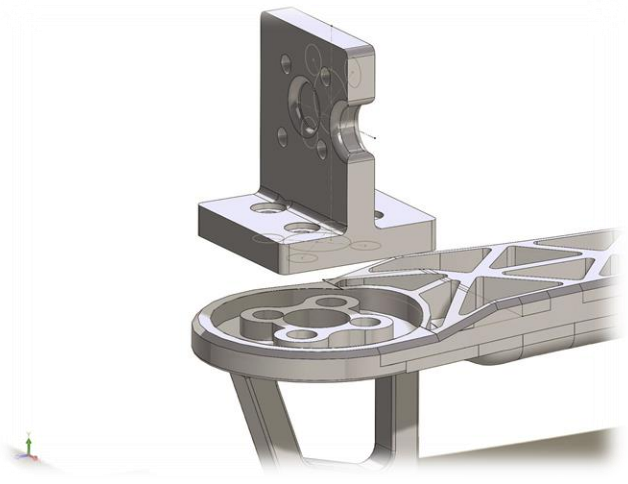


FIGURE 3: MOTOR MOUNT

SOFTWARE SYSTEM

GROUND ROBOT

The ground robot will be running a linux operating system that will act as the base of our system. For the final system, our code will be written and compiled on a separate linux machine, and then transferred to the ground vehicle either through serial cable or SSH into the machine. The most important modules are the networks module, kalman filter/sensor fusion, controller algorithm, and AI processing.

a) **Networks Module**
The networks module is meant to communicate between the devices. Throughout the testing phase, we will be using wireless communications to talk back and forth between the propeller system, instrumentation arm, the ground robot, and the GUI. When we are using the wireless system, there will be a large amount of communications between the devices that is required, but we plan on adapting networking code from the previous Omnibot project in order to support all new devices.
The ideal final system is completely wired together (with exception of the GUI controls). Since our system will be connected through the instrumentation arm, we should be able to wire the system very easily, and thus we can simplify the networks module to only communicate with the GUI
Inputs: Sensor data and user commands
Outputs: Control commands

b) **Kalman Filter/Sensor Fusion**
The Kalman Filter/Sensor Fusion algorithm will take all of the data from every sensor and combine them to produce a single estimate of the pitch and roll of the instrumentation arm. This will allow for more precise tracking of the pitch and roll angles and better sensor noise resilience.
Once the data is transformed by the algorithm, it is ready to be sent to the PID loop as the current position information.
Inputs: Sensor data
Outputs: Combined/transformed sensor data

c) **Controller Algorithm**
The Controller Algorithm takes the current position and the desired position, then it converts those into a command signal that will be sent to the motors.
Inputs: Sensor data
Outputs: Propeller command signal

d) **AI processing**
The system should be automated in the movement through rows of experimental crops, so based on the current state of the ground robot, the AI will decide what direction it needs to move next in order to stay on course.
Inputs: User commands
Outputs: Speed, Heading

DATA ANALYSIS

The data analysis tool is used to analyze runtime data and to graphically display the values that are being sent between different parts of the whole system. The values are first logged by the GUI to a text file. A MATLAB script then parses the file and generates graphical plots of the data. The MATLAB script will offer the following options to the user:

1. Choose a single data value to be plotted.
2. Choose 4 data values to be plotted on a 2x2 grid.
3. Plot multiple data in one MATLAB figure.

The current data analysis tool doesn't offer the above configuration options to the user. It is sufficiently limited to analyze test experiments of the 1-D inverted pendulum system. Some of the configuration options will be implemented before testing of the 2-d inverted pendulum system will begin.
Inputs: Text file consisting of logged data with headers
Outputs: MATLAB plots of logged data.
Interface: GUI (to log data) and MATLAB (to plot data)

CONTROL SYSTEM

INSTRUMENTATION ARM

The overall instrumentation arm control system can be seen in the following diagram.

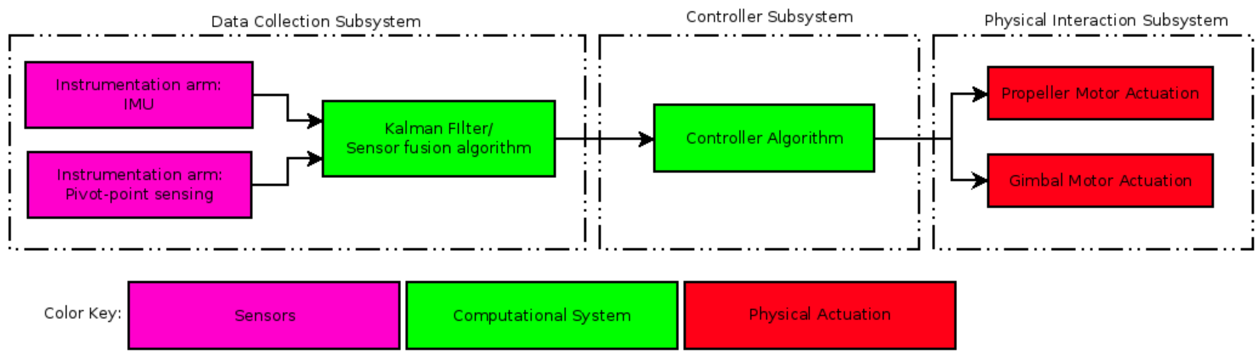

Data Collection Subsystem
Controller Subsystem
Physical Interaction Subsystem
Instrumentation arm: IMU
Instrumentation arm: Pivot-point sensing
Kalman Filter/ Sensor fusion algorithm
Controller Algorithm
Propeller Motor Actuation
Gimbal Motor Actuation
Color Key:
Sensors
Computational System
Physical Actuation

FIGURE 4: CONTROLLER SYSTEM

The controller is composed of two distinct subsystems: Data Collection Subsystem, and Controller Subsystem

DATA COLLECTION SUBSYSTEM

In the final design for the instrumentation arm system there will be two different methods of collecting data about the current position of the instrumentation arm.

a) **Inertial Measurement Unit (IMU)**
   An IMU will be located on the top of the instrumentation arm. The IMU will consist of a 3-axis accelerometer, a 3-axis gyroscope, and a 3-axis magnetometer. These sensors will be able to supply measurements at the rate of ~1kHz to the controller.
   Input: Physical movement of the system
   Output: 16-bit values of the current measurements
   Interface: I2C communications interface to the controller

b) **Angular Measurement Devices**
   Two angular measurement devices will be located at the pivot point, one to provide angular measurement for the pitch direction, the other providing angular measurement for the roll direction.
   Input: Physical movement of the system
   Output: Angular measurement
   Interface: Analog voltage input to the controller

All of these sensor readings will be combined using a Kalman filter sensor fusion algorithm to produce a single estimate of the instrumentation arm's pitch and roll position.
Input: Measurements from the IMU system and the angular measurement system.
Output: State estimation of the current angular position of the instrumentation arm.

CONTROLLER SUBSYSTEM

The instrumentation arm controller will consist of a modularized controller interface that will take the following inputs and outputs:
Input: Angular position of the instrumentation arm (pitch and roll measurements). Requested movement of the ground robot.
Output: Motor commands for the four propellers located on the top of the instrumentation arm. Motor commands for the actuators at the pivot point of the instrumentation arm.

The overall controller will be a modular system so that the control algorithm can be changed without modifying the rest of the system. This will allow for experimentation with control algorithms. The possible control algorithms explored in this design are PID controllers and LQR controllers.

Since this instrumentation arm system is a non-linear dynamic system, each controller for this system will be linearized around the balance point (when the instrumentation arm is vertical). This linearization may present issues when the instrumentation arm position varies greatly from vertical. In order to reduce these issues, there may be two pieces to the controller.

a) **Non-linear stabilization system**
This system will take the current angle from vertical and provide large actuation inputs to the propeller system to overcome the gravitational force experienced by the leaning instrumentation arm. This will move the instrumentation arm back into the linearization region for the main controller.
Input: Instrumentation arm angle from vertical
Output: Actuation inputs to the propeller system

b) **Linear stabilization system**
This system will take the current angle and provide small actuation inputs to the gimbal-mounted actuators to maintain the instrumentation arm's position near vertical.
Input: Instrumentation arm angle from vertical
Output: Actuation inputs to the gimbal-mounted actuators

SOFTWARE TOOLS

GUI

Purpose: Control the desired settings and profiles of the propeller system. Also allows command line input to change the position of the system.
Setup:

- If needed, configure the desired constants of the PID controllers in the Profiles/GAUIquadPID.profile file.
- Run the MicroCART executable on the local machine terminal:
    - ./MicroCART (in the terminal)

MATLAB

Purpose: Parse text files containing logged data and generate plots for the parsed data corresponding to user's configuration options.
Setup: Open up the Configuration.m script, set the required variables, and run the script. Make sure the text file with the logged data is saved with the right name.

SIMULINK

Simulink will be used during the design of the controller to model the overall system and simulate the response of the system to different control algorithms. A differential equation model for the non-linear system dynamics of the instrumentation arm will be created and then placed into a Simulink block-diagram. Then, a model of the desired control algorithm will be created in Simulink. This model will then be paired with the system model, and the overall system response to the controller will be measured.

This analysis will assist in the design and identification of possible controllers for the instrumentation arm. In addition, the detailed simulations will provide early insight into possible issues that may be faced in the physical system, and allow for possible solutions to be integrated into the design faster.

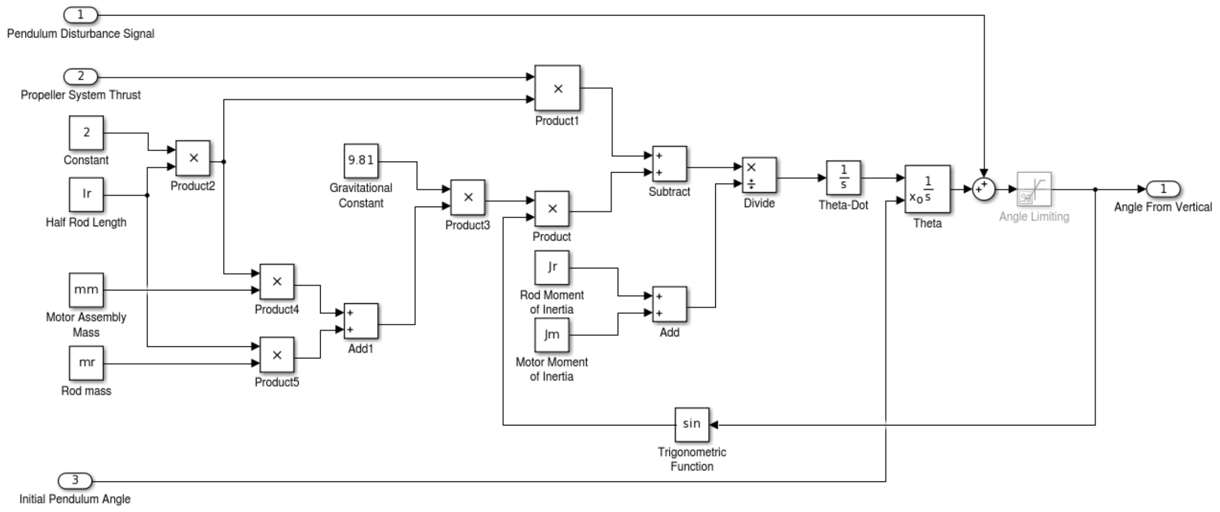


FIGURE 5: SIMULINK BLOCK DIAGRAM SHOWING NON-LINEAR DIFFERENTIAL EQUATION

HARDWARE TOOLS

XILINX ISE

The Xilinx ISE is used to write VHDL files and generate BIT files that can be programmed onto the FPGA on the ground robot.

LINUXCNC

An open source software used to program the 4i68 Mesa board currently being used for the system. Different from ISE in that this uses pre-made drivers as a base for editing.