MathWorks Minidrone Competition

Overview

MathWorks

Agenda

- **Competition overview**
- A quick look at a working design
- Recap and Resources
- Appendix
  - Your starting point model
  - Tips and Tricks

MathWorks Minidrone Compeititon

Learn how to develop an autonomous minidrone line follower and develop key skills like *Model-Based Design* relevant to the industry - all while having fun using drones!

Organized by MathWorks

https://www.mathworks.com/academia/student-competitions/minidrones.html

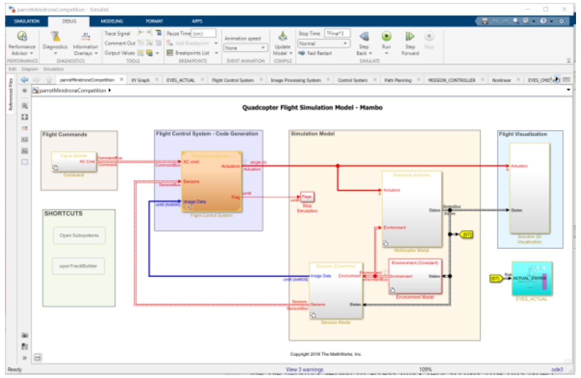
Develop a Minidrone line following algorithm

- Round 1: Simulation Round
  - Edit a starting point model provided by MathWorks
  - Submit your models to MathWorks (deadlines on web)
  - The BEST submissions go through to Round 2
- Round 2:
  - Simulation and Hardware Deployment Round
    or
  - Virtual Round

Depending on your local competition

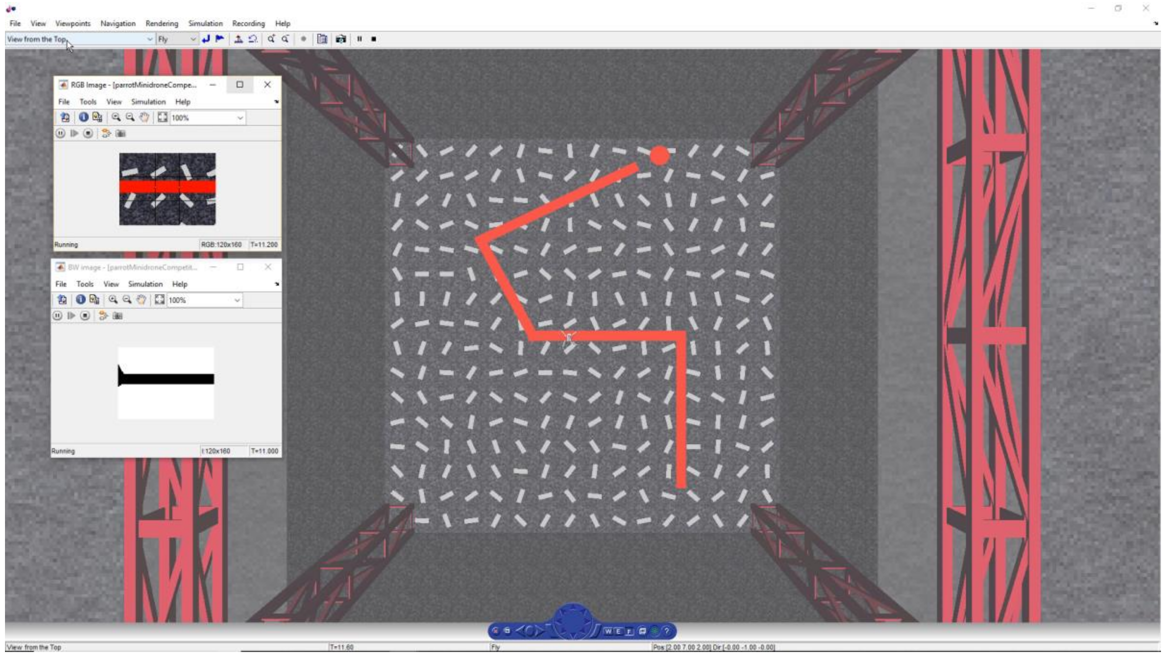
Simulation
(Round 1)

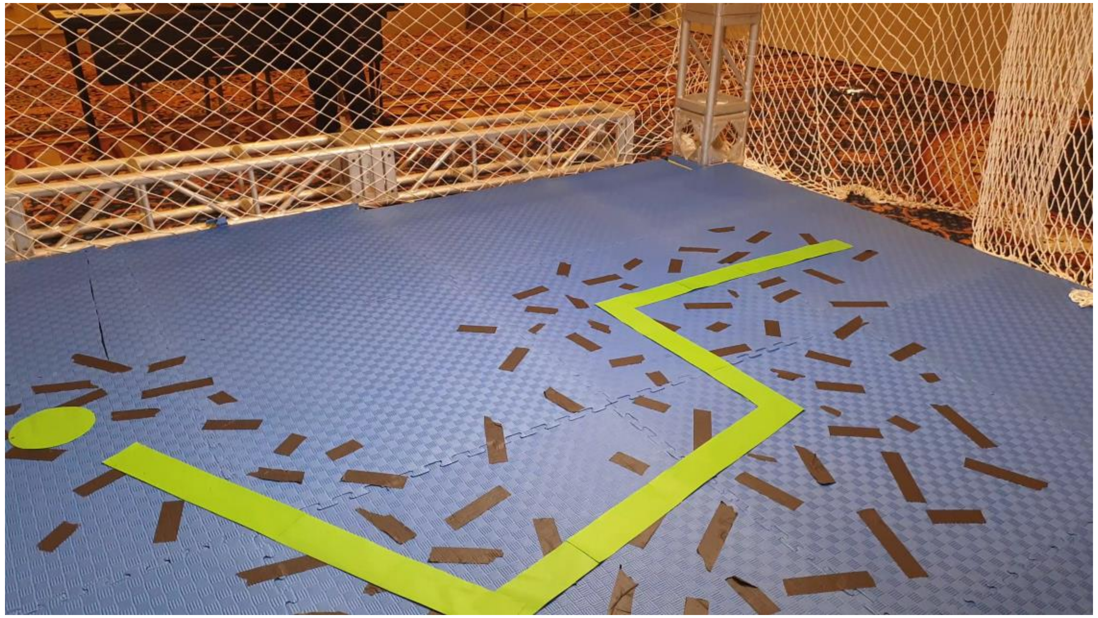
Real system
(Round 2)



The BEST Round 1 submissions will advance to Round 2

Depending on your local competition





You Can Find the Competitions Around the World!

Student Programs

Search MathWorks.com

Overview | Teach | Learn | Research | Student Programs

Competitions Around the World

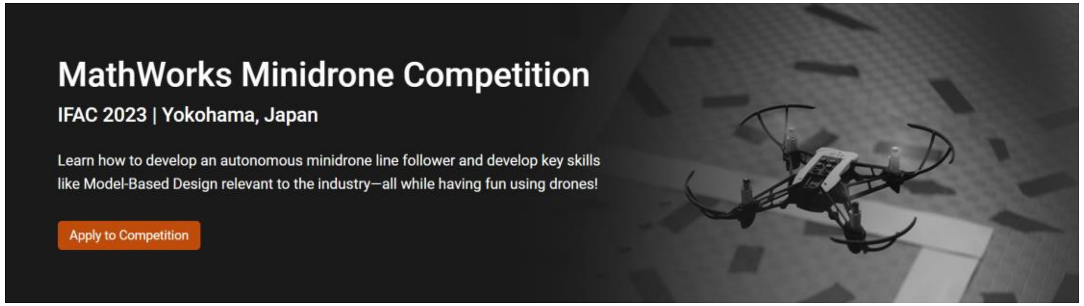
Regional Competition	Format	Venue	Application Deadline	Simulation Submission Deadline
Poland 2023	In Person	Poznan, Poland	December 09, 2022 1:00 p.m. EST	December 23, 2022
Turkey 2023	In Person	Ankara, Turkey	December 09, 2022 1:00 p.m. EST	December 23, 2022
IFAC 2023	In Person	Yokohama, Japan	April 5, 2023 1:00 p.m. EST	April 19, 2023
Australia 2023	In Person	Sydney, Australia	April 12, 2023 1:00 p.m. EST	May 1, 2023

Click on a relevant instance of the competition

https://www.mathworks.com/academia/student-competitions/minidrones.html

Get Ready to Fly

1. Apply to receive all the competition materials.
2. Watch the MathWorks Minidrone Competition video series to better understand the competition details.
3. Review the competition rules and guidelines.
4. Complete MATLAB Onramp, Simulink Onramp, and Stateflow Onramp courses before starting to work on your algorithm.



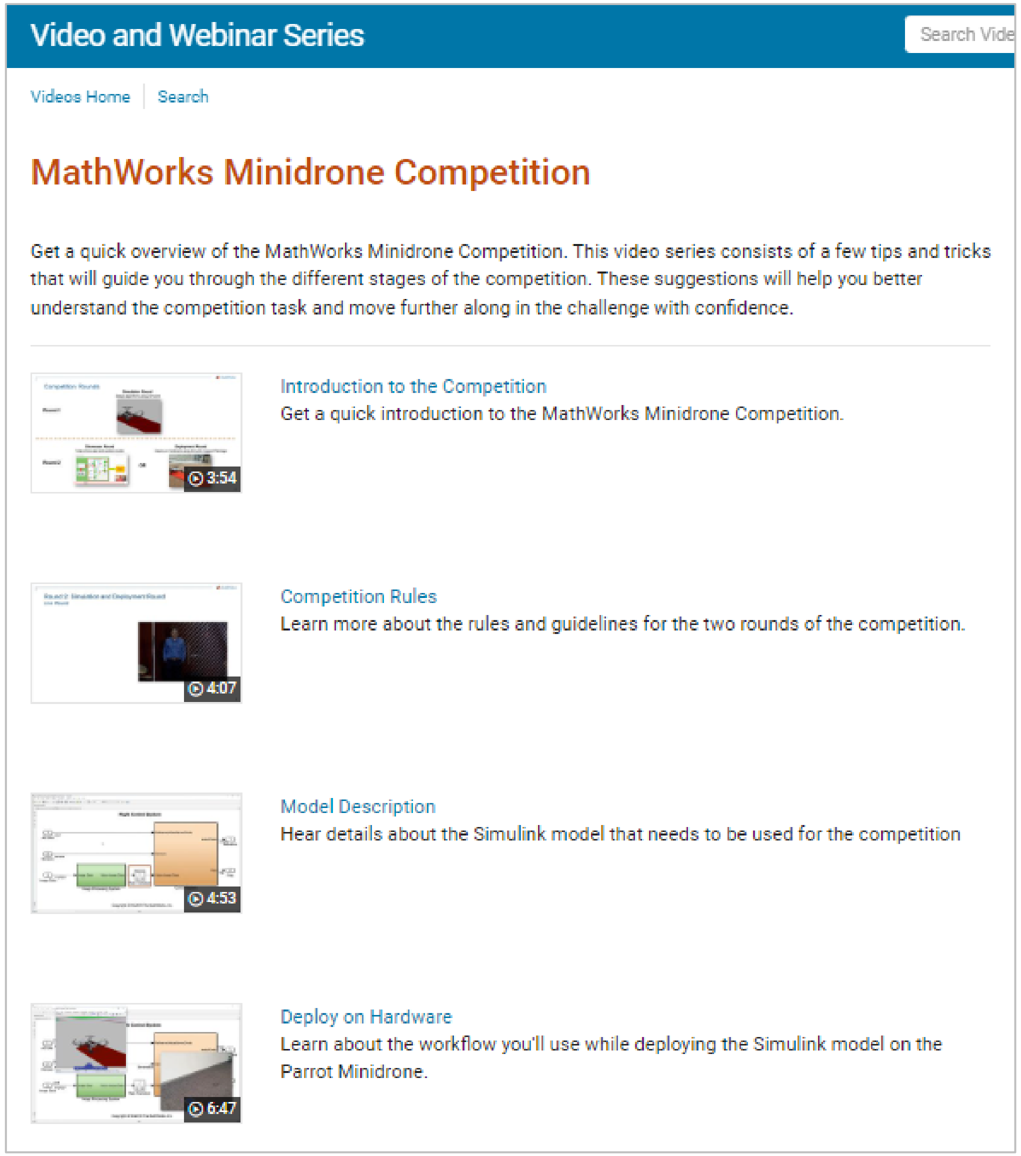
You need to APPLY if you want to enter the specific competition.

Complimentary licenses are sent to teams

Get Ready to Fly

Make sure you watch these short videos

1. Apply to receive all the competition materials.
2. Watch the [MathWorks Minidrone Competition video series] to better understand the competition details.
3. Review the competition [rules and guidelines].
4. Complete [MATLAB Onramp], [Simulink Onramp], and [Stateflow Onramp] courses before starting to work on your algorithm.



Get Ready to Fly

1. Apply to receive all the competition materials.
2. Watch the MathWorks Minidrone Competition video series to better understand the competition details.
3. Review the competition rules and guidelines.
4. Complete MATLAB Onramp, Simulink Onramp, and Stateflow Onramp courses before starting to work on your algorithm.

Familiarise yourself with the rules

MathWorks
Accelerating the pace of engineering and science

MathWorks Minidrone Competition

Rules and Guidelines

Table of Contents

mathworks.com

00000v00 00/00

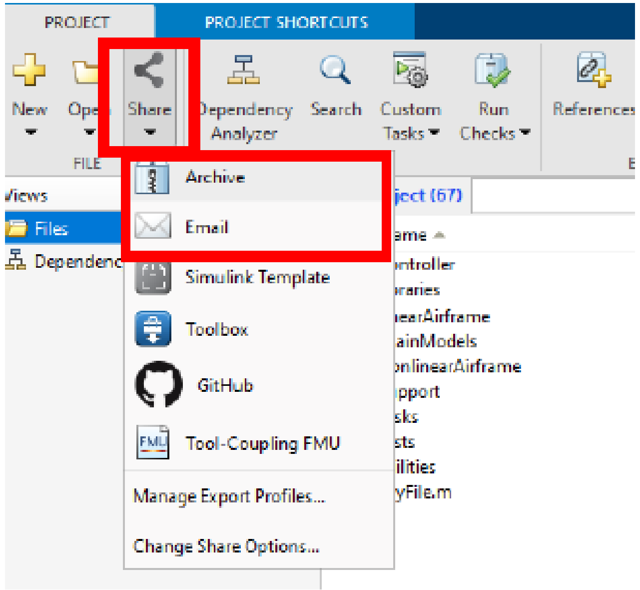
Rules and Guidelines – Submit Your Simulation

***How** do you submit your Simulation at the end of Round 1 ?*

***Who** do you send this to?*

3. Submit the Project Archive

- In the Projects folder, click on Share and then select *Archive* or *E-mail*.



- Select *Export Profile* as 'Whole Project' and name the Project Archive as `<TeamName>` where `<TeamName>` is the name of your team.
- Send the Project Archive (.mlproj) to ***minidronecompetition@mathworks.com*** with the subject as '`<TeamName>` at `<EventName>` `<EventYear>`' where `<EventName>` is the name of the event as found on the web page for the competition (often a location, conference, or university) and `<EventYear>` is the year when the deployment round of the competition is to be held. For example, if your team name is 'Drone Squad' and you are participating in the event at IROS 2019, your email would be titled 'Drone Squad at IROS 2019'.
- The Team Captain should submit the model using their registered email address.

Get the Rules and Guidelines ([PDF](PDF))

Rules and Guidelines – Competition Set-up

C.1.2 Competition Set-Up

The deployment round will be one- or two-day event. The competition will be divided into two parts:

a. **Practice Round:**

- In this round, each team gets **two slots of 15 minutes each** in the Arena to calibrate their model gains and thresholds.
- Performance of the Minidrone during this round will not be considered to declare the winners.

b. **Live Round:**

- Each team will get one **15-minute** assigned slot in the arena. This includes the setup time and the Minidrone flight.
- Each team gets **maximum 7 chances** during the 15 minutes to fly the Minidrone.

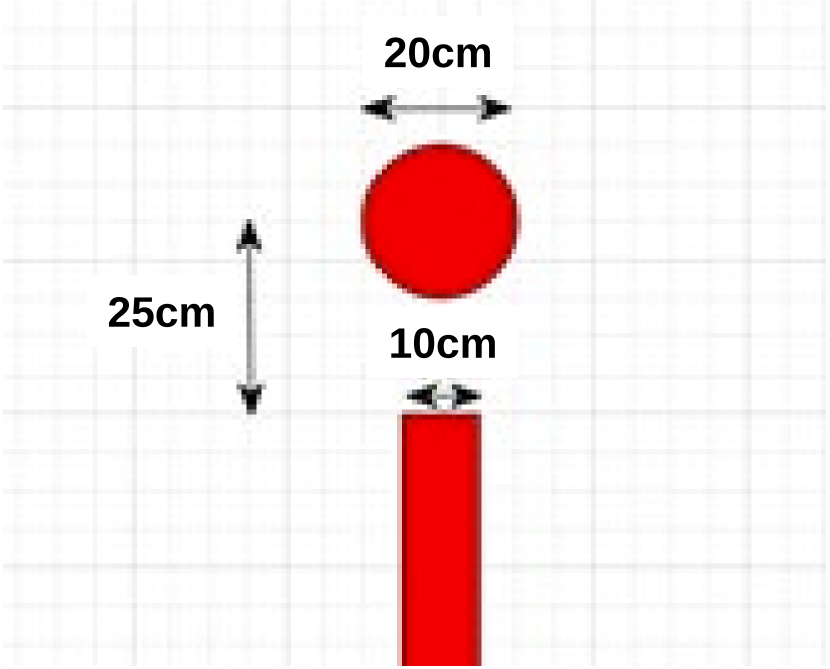
Get the Rules and Guidelines (PDF)

Rules and Guidelines – Arena Details

D. Arena Details

This detail of the arena is valid for Round 1 as well as Round 2. The following are the details about the arena track:

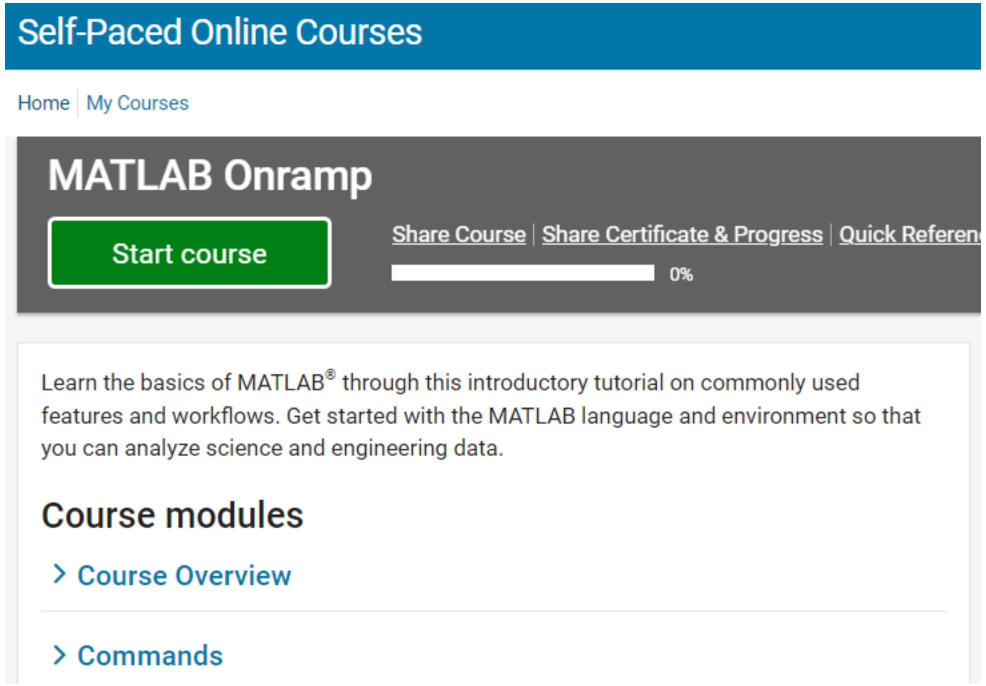
- The arena would be a 4-meter * 4-meter space enclosed by nets on all sides.
- The arena track will be 10 cm in width.
- The landing circular marker will have a diameter of 20 cm.
- The line follower track will consist of connected line segments only and will not have any smooth curves at the connections.
- The angle between any two track sections could have a value between 10 degrees to 350 degrees.
- The track can have anywhere between 1 to 10 connected line segments. The initial position of the drone will always be on the start of the line. However, the mouth of the drone may not always face the direction of the first line on the track.
- The distance from the end of the track to the center of the circle will be 25 cm.
- The background on which the track will be laid may not be a single color and will have texture.
- The color and the track for the in-person round will be disclosed on the day of the competition.
- The track for the Practice Round and the Live Round may be different in case on an in-person round.



Get Ready to Fly

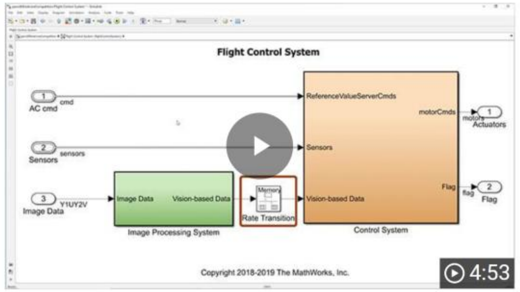
1. Apply to receive all the competition materials.
2. Watch the [MathWorks Minidrone Competition video series] to better understand the competition details.
3. Review the competition [rules and guidelines].
4. Complete [MATLAB Onramp], [Simulink Onramp], and [Stateflow Onramp] courses before starting to work on your algorithm.

Get started with self paced online courses



Getting Started

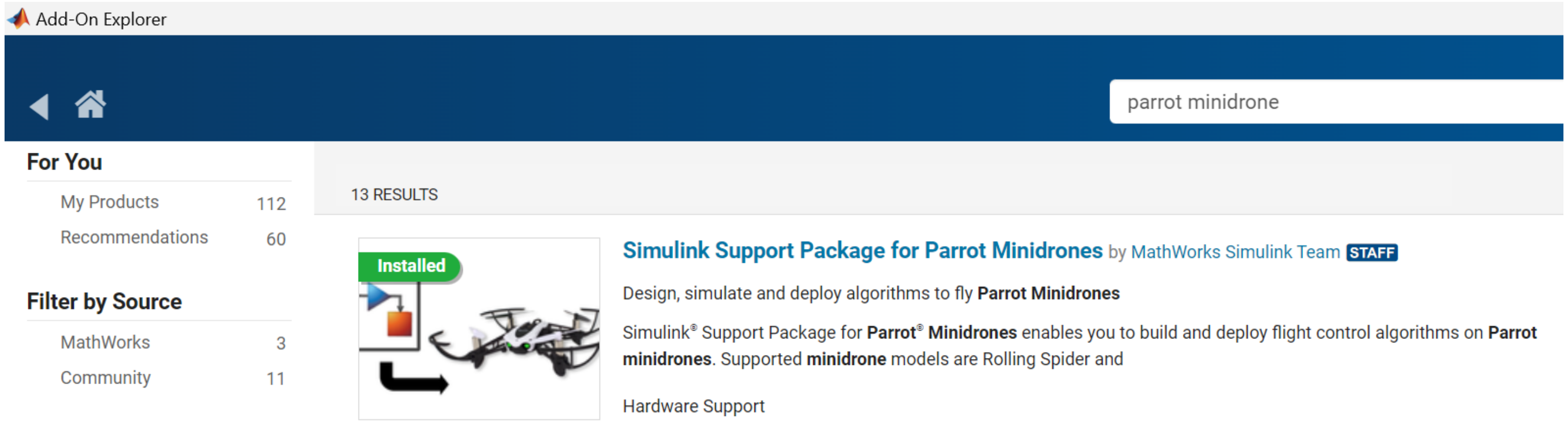
1. Watch this 5-minute video

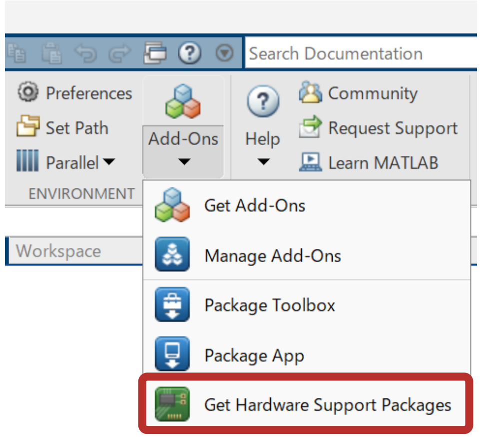


MathWorks Minidrone Competition: Model Description
Get details about the Simulink® model that needs to be used for the MathWorks Minidrone Competition and learn how to use the Simulink Support Package for Parrot® Minidrones.

https://www.mathworks.com/videos/mathworks-minidrone-competition-model-description-1551445160030.html

2. Install the Simulink Support Package for the Parrot Minidrone





File Exchange

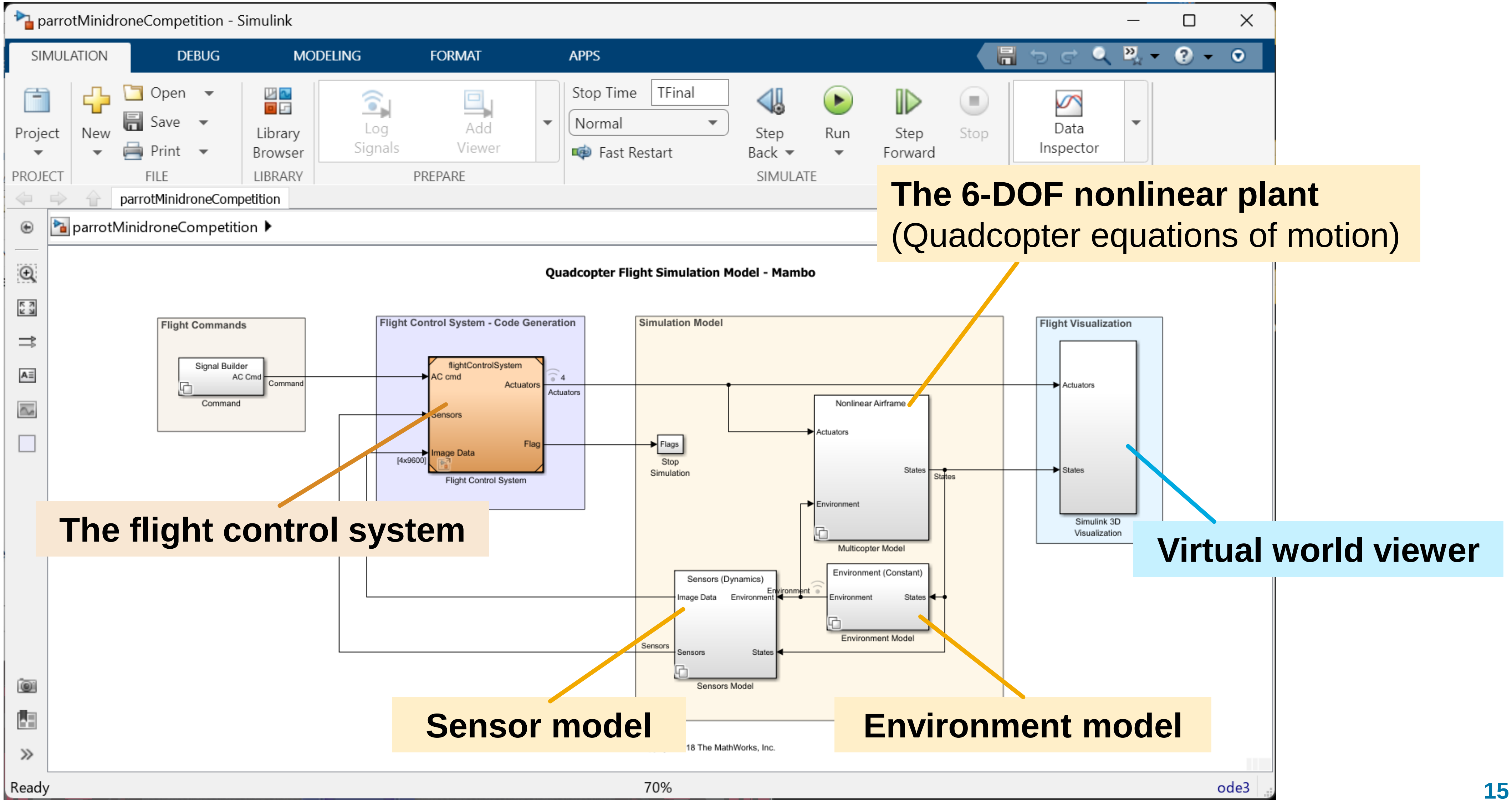
You now have the starting point Simulink model !

Agenda

- Competition overview
- **A quick look at a working design**
- Recap and Resources
- Appendix
  - Your starting point model
  - Tips and Tricks

Here's My Draft of A Working Model

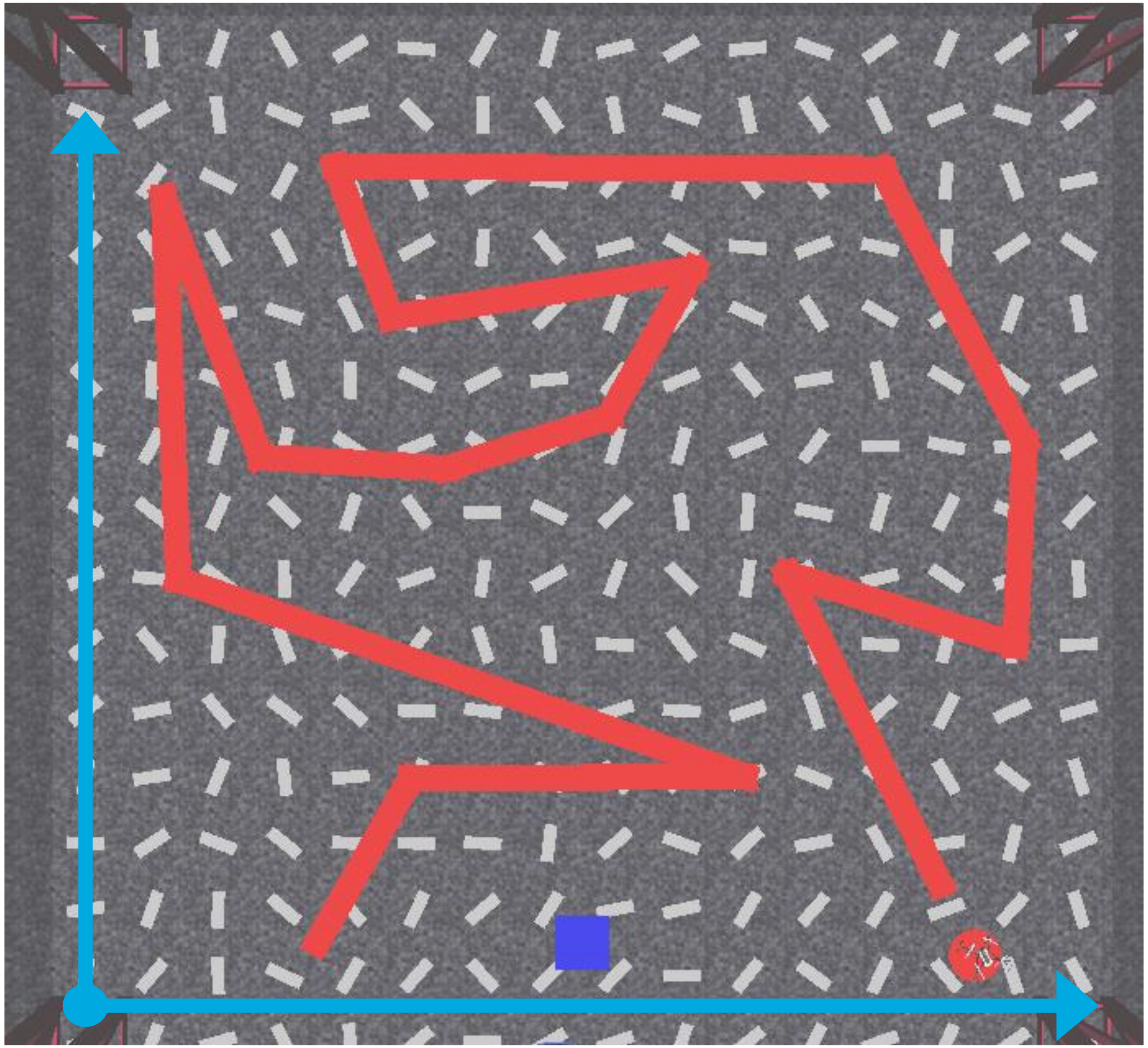
Go to MATLAB and Simulink !



The Inertial G-frame

The Arena is: **4m x 4m x 3m**

North
X



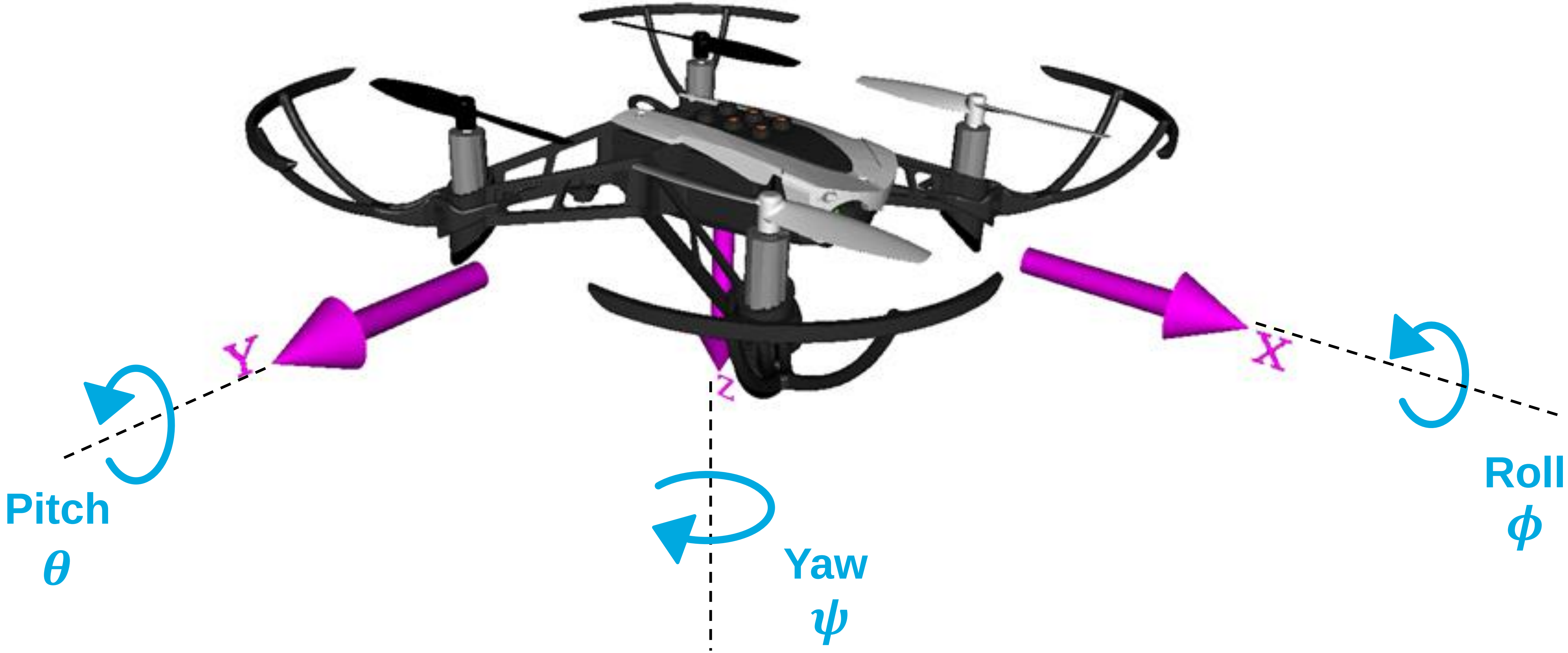
East
Y

And Right hand rule for **Z**, i.e.:

(Z / down) is positive into the ground …. And negative pointing to the sky

Acronym:
North East Down (NED)

The Drone’s Body Fixed B-frame



Euler Rotation Sequence

For a Z-Y-X Euler rotation sequence we get

```
R1Z =
```

$$\begin{pmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```
R2Y =
```

$$\begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{pmatrix}$$

```
R3X =
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{pmatrix}$$

Inertial G-frame → Body fixed B-frame

$$v^B = \left[{}^B R_G\right] . v^G$$

$$= \mathrm{R3}(\phi_X) . \mathrm{R2}(\theta_Y) . \mathrm{R1}(\psi_Z) . v^G$$

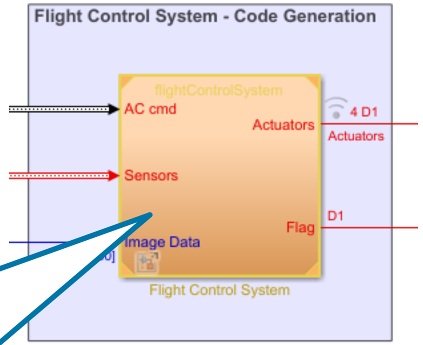
Body fixed B-frame → Inertial G-frame

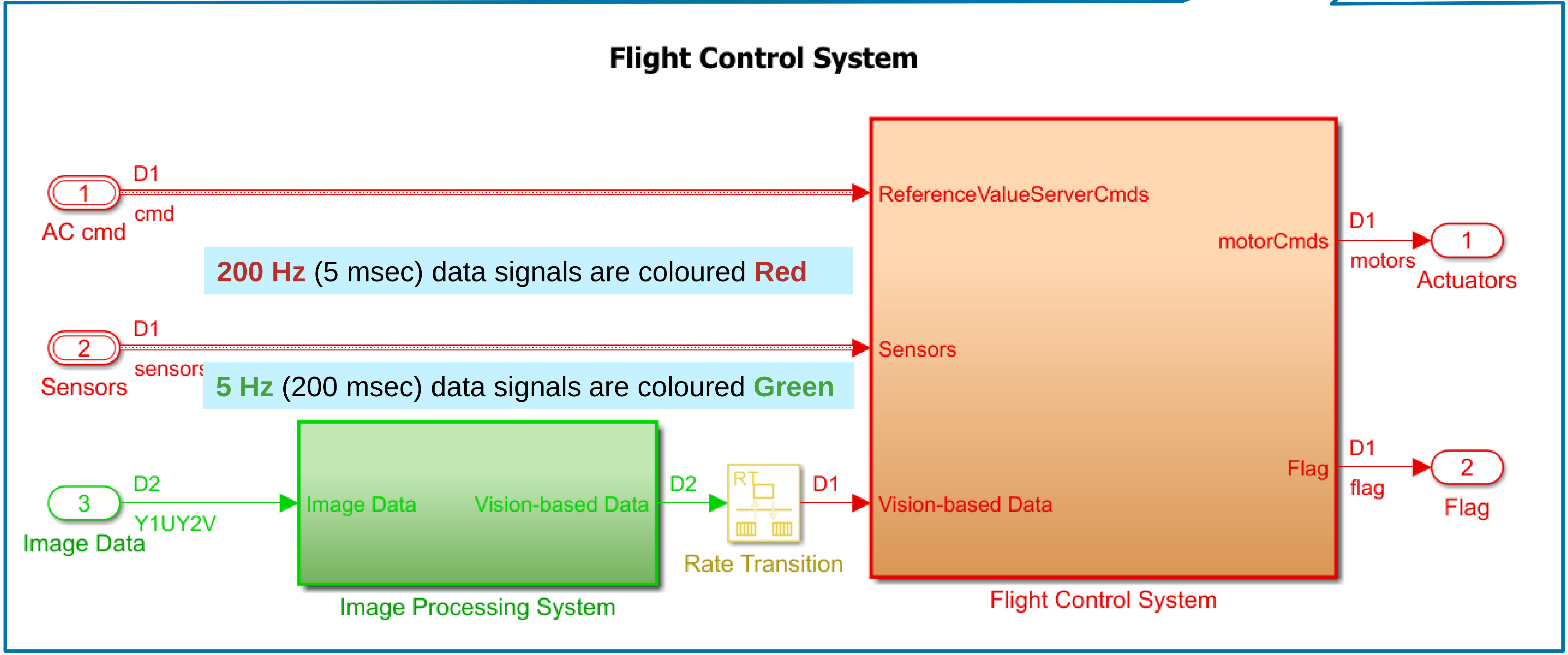
$$v^G = \left[{}^B R_G\right]^{-1} . v^B$$

$$= \left[{}^B R_G\right]^T . v^B$$

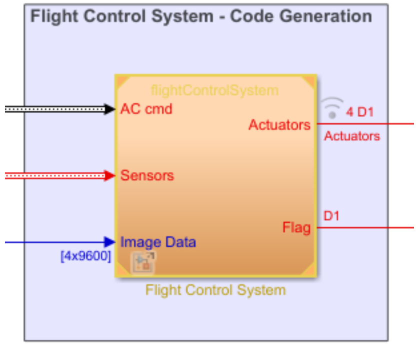
Flight Control System

This is the subsystem that you must edit

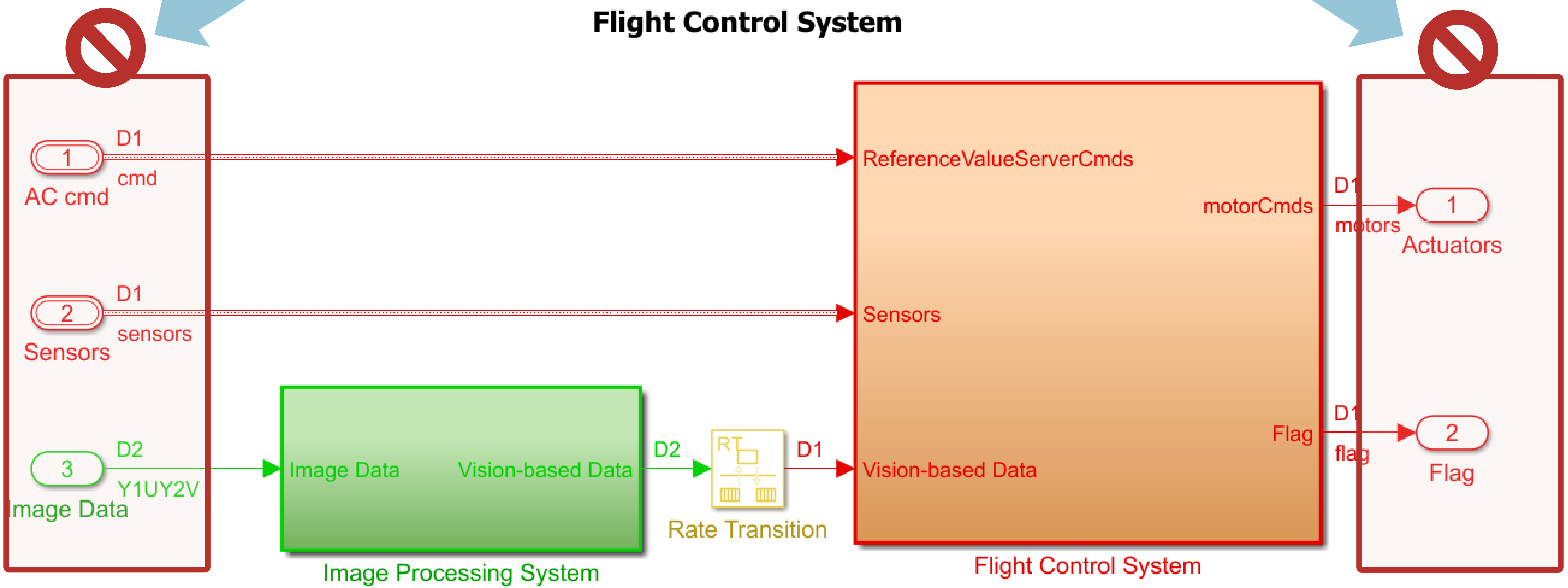




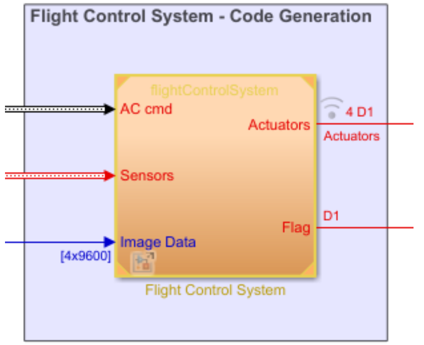
Flight Control System



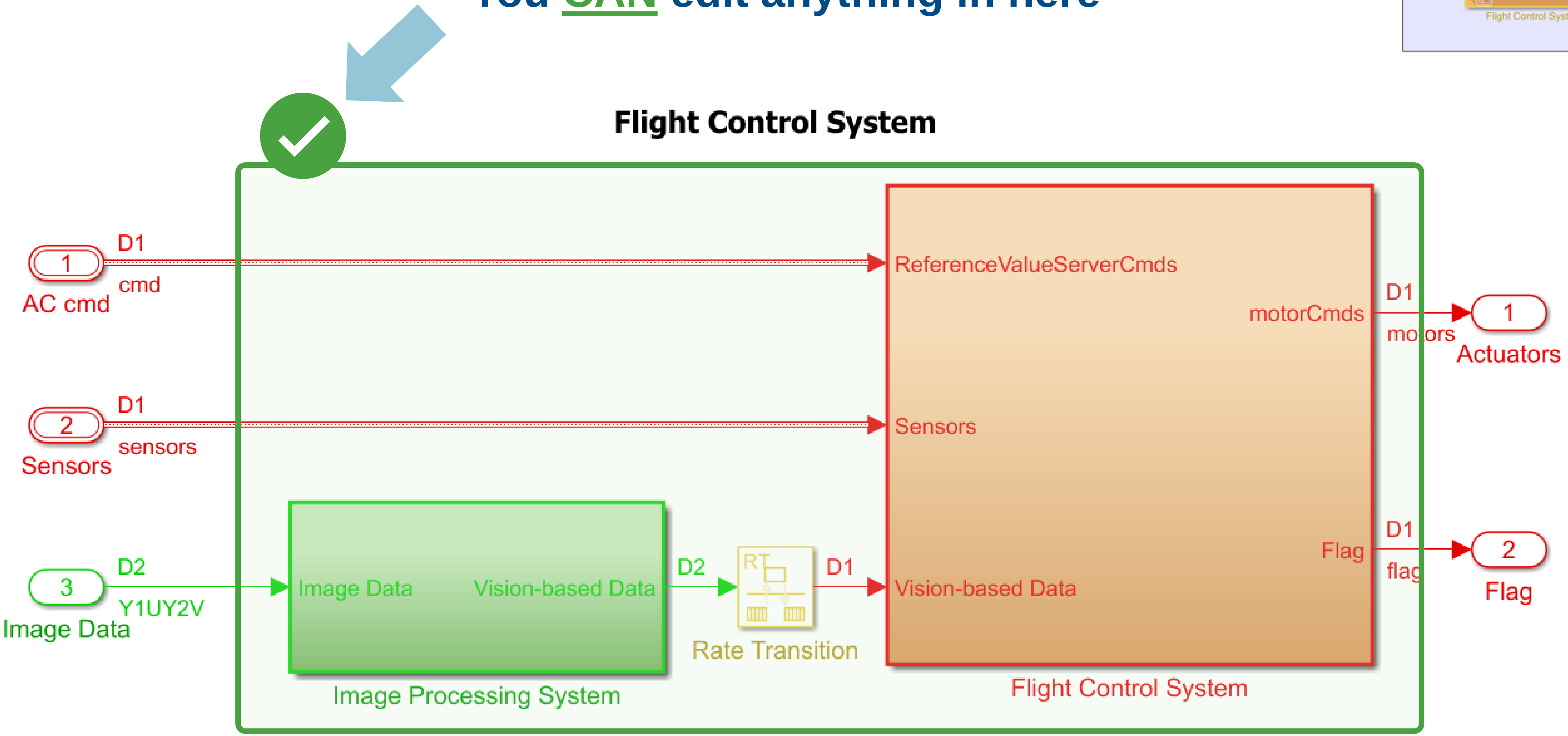
You can <u>NOT</u> edit the interfaces of the flight control system



Flight Control System



You CAN edit anything in here



Agenda

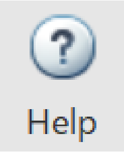
- Competition overview
- A quick look at a working design
- **Recap and Resources**
- Appendix
  - Your starting point model
  - Tips and Tricks

Recap

1. **The MathWorks Minidrone Competition**
   - Develop a control strategy for a line following quadcopter
   - Practice Model-Based Design (MBD)
2. **Register for the competition**
   - Your application and submission due dates and the T&Cs for that specific competition
   - Install your complementary MATLAB license
3. **Review the rules document and watch the getting started video series**
4. **Install the Simulink Support Package for Parrot Minidrone**
   - You get a starting point Simulink model to work on
5. **Solve the problem using the Simulink model**
6. **Round 1**
   - Submit your Simulink model project by the stated due date
7. **Round 2**
   - Simulation and hardware deployment round OR Virtual round, depending on each competition

Resources

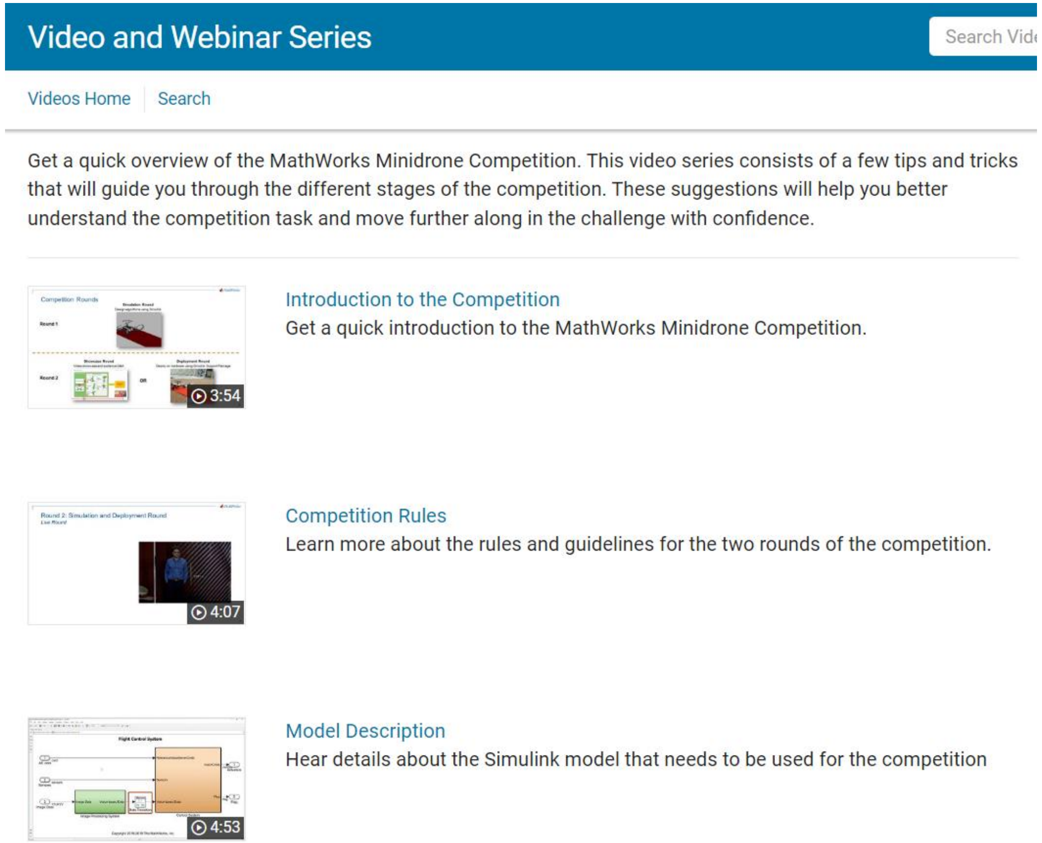
- The MATLAB Help Browser



- The Minidrone Video Series

- Self-Paced Online Courses
  - [MATLAB Academy]
    - [MATLAB Onramp]
    - [Simulink Onramp]
    - [Stateflow Onramp]
    - [Image Processing Onramp]

- Minidrone Virtual Lab Contents
  - [Getting Started with Minidrone Basics Using Virtual Lab Modules]



https://www.mathworks.com/videos/series/mathworks-minidrone-competition.html

MathWorks Minidrone Compeititon

Learn how to develop an autonomous minidrone line follower and develop key skills like *Model-Based Design* relevant to the industry - all while having fun using drones!

Organized by MathWorks

https://www.mathworks.com/academia/student-competitions/minidrones.html

Develop a Minidrone line following algorithm

- Round 1: Simulation Round
  - Edit a starting point model provided by MathWorks
  - Submit your models to MathWorks (deadlines on web)
  - The BEST submissions go through to Round 2
- Round 2:
  - Simulation and Hardware Deployment Round or
  - Virtual Round

Depending on your local competition

Let's have fun with MBD !

We are looking forward to your application !

Agenda

- Competition overview
- A quick look at a working design
- Recap and Resources
- **Appendix**
  - Your starting point model
  - Tips and Tricks

Appendix

- **Your starting point model**
  - Launching the Simulink project
  - The relevant subsystems and data buses
- Tips and Tricks

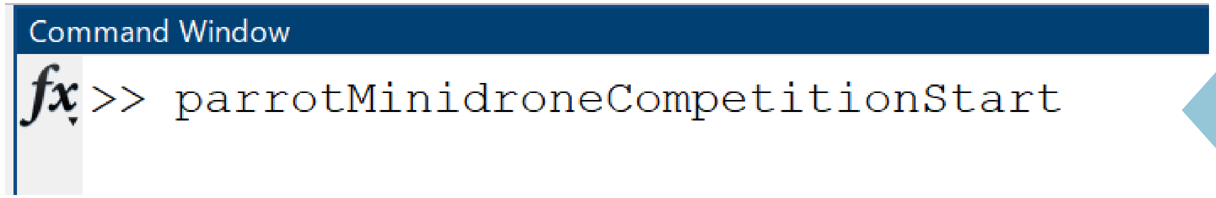
Prerequisites

1. Registered for the Competition
   – Wherever that is around the globe

2. Received your complimentary MATLAB License
   – … and have installed this license (contains all of our products)

3. Installed the Simulink support package for the Parrot Minidrone

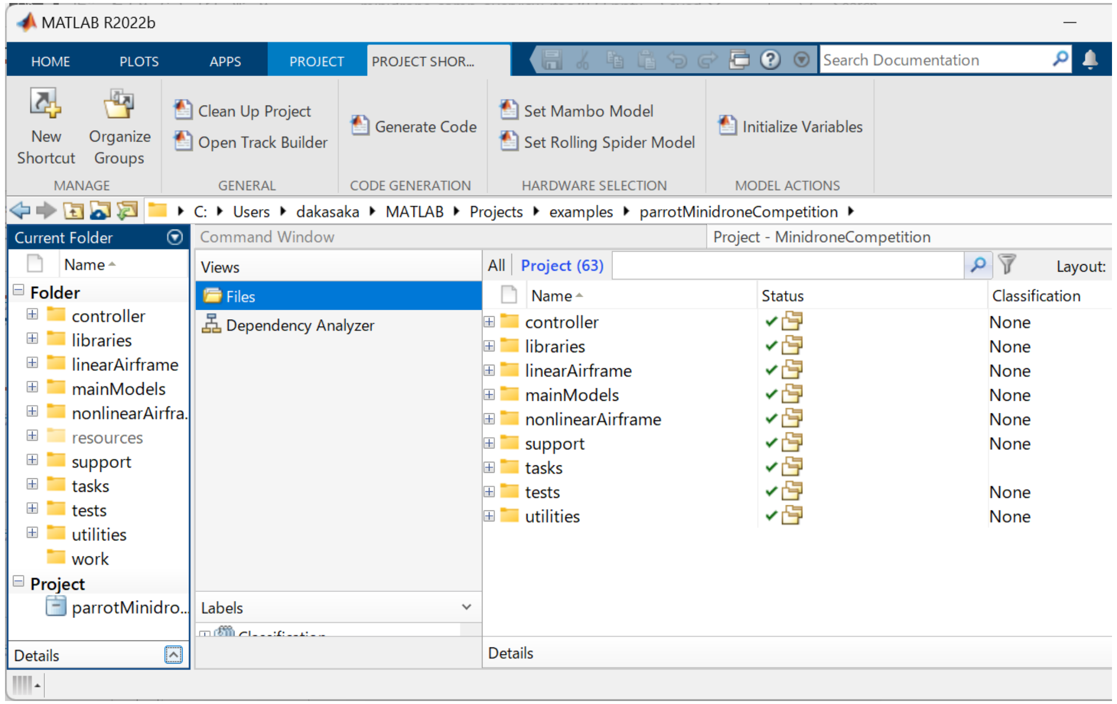
Launching the Simulink Project



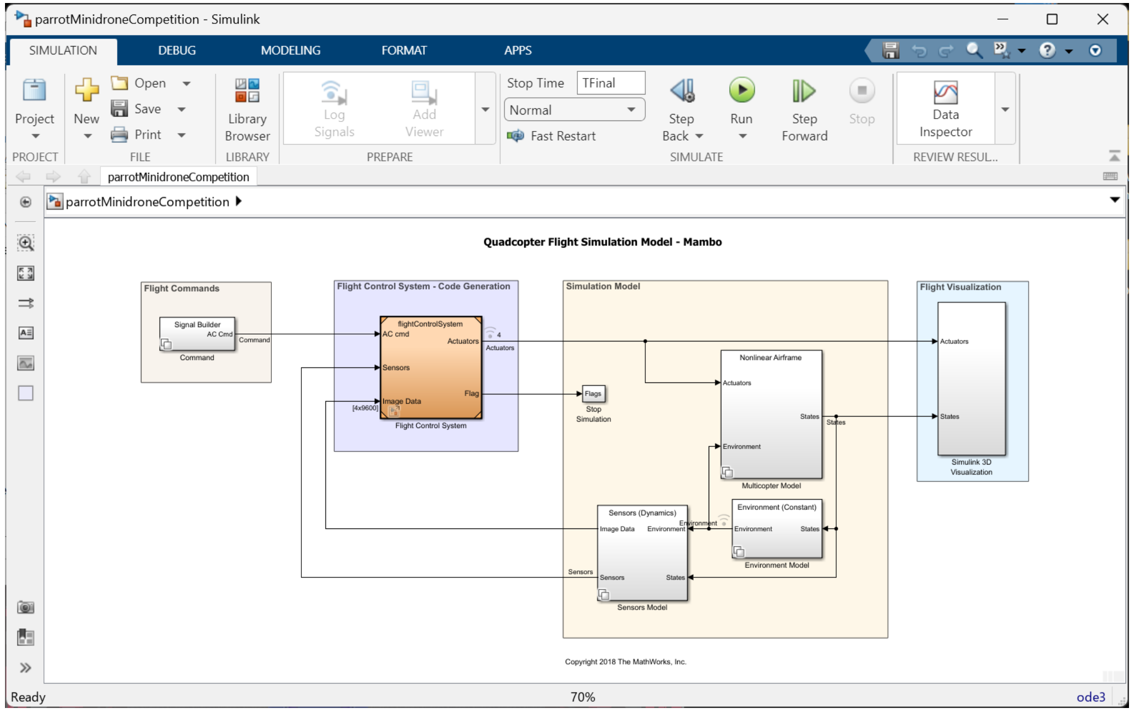
Just type:
`parrotMinidroneCompetitionStart`

Open

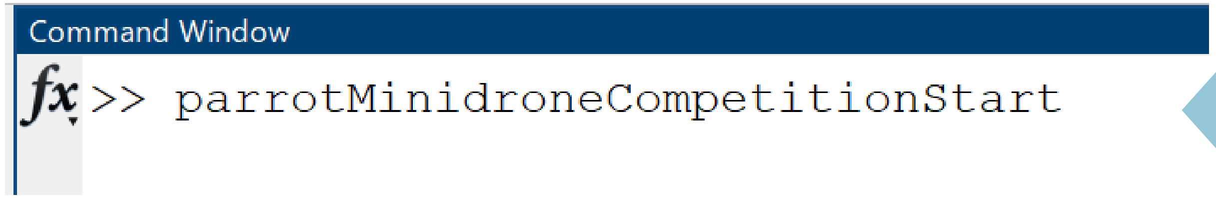
The Simulink project



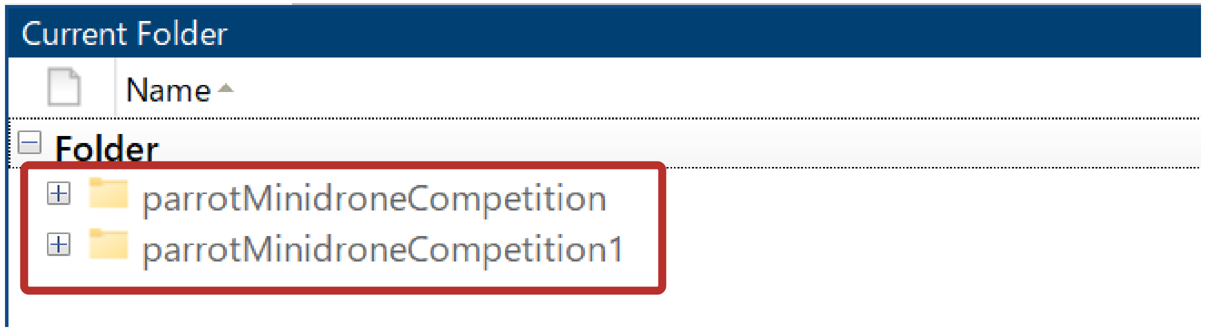
The starting point of Simulink model



Launching the Simulink Project - The 1st Time

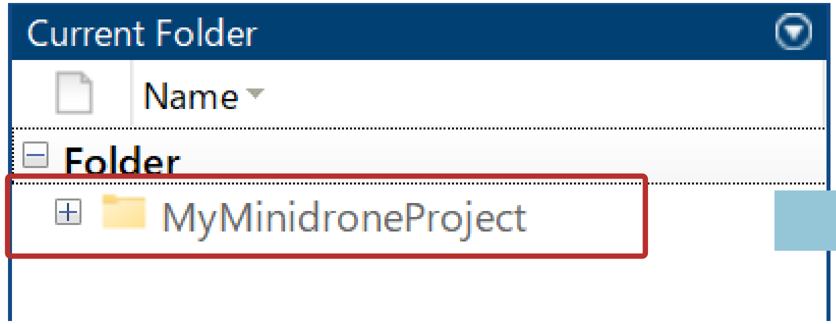


Each time you type this, it will create a NEW folder and project from scratch

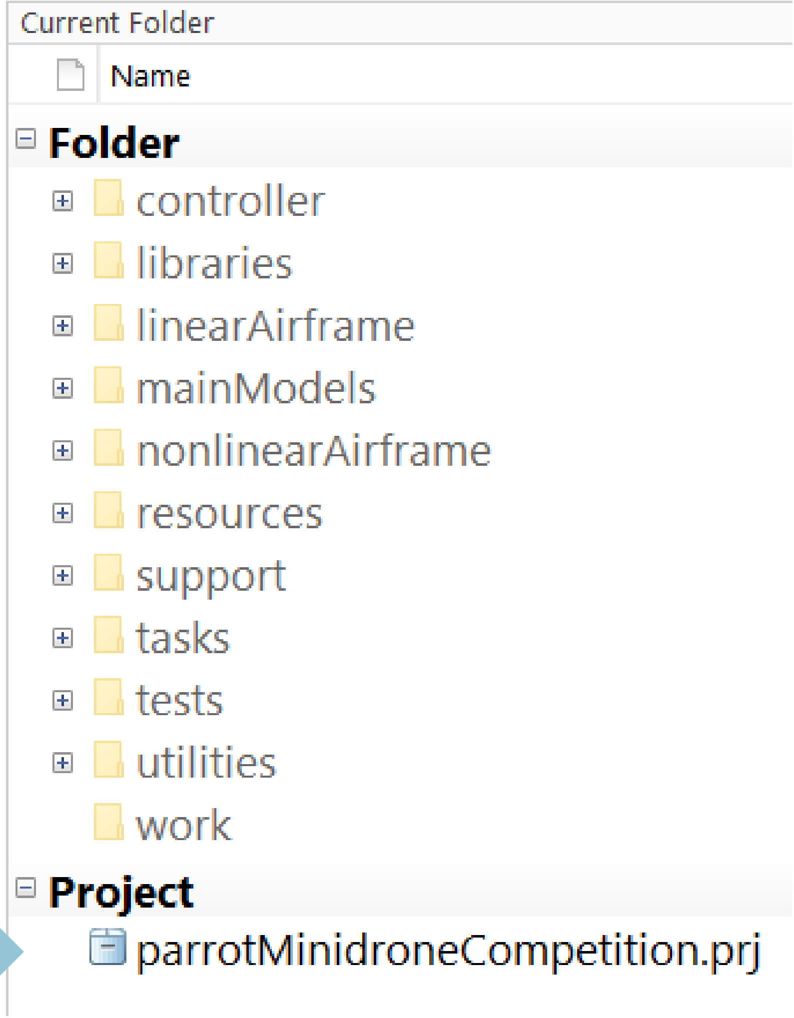


Launching the Simulink Project - The Next Time

Once you start working on this, just rename your root folder, e.g.:



Open

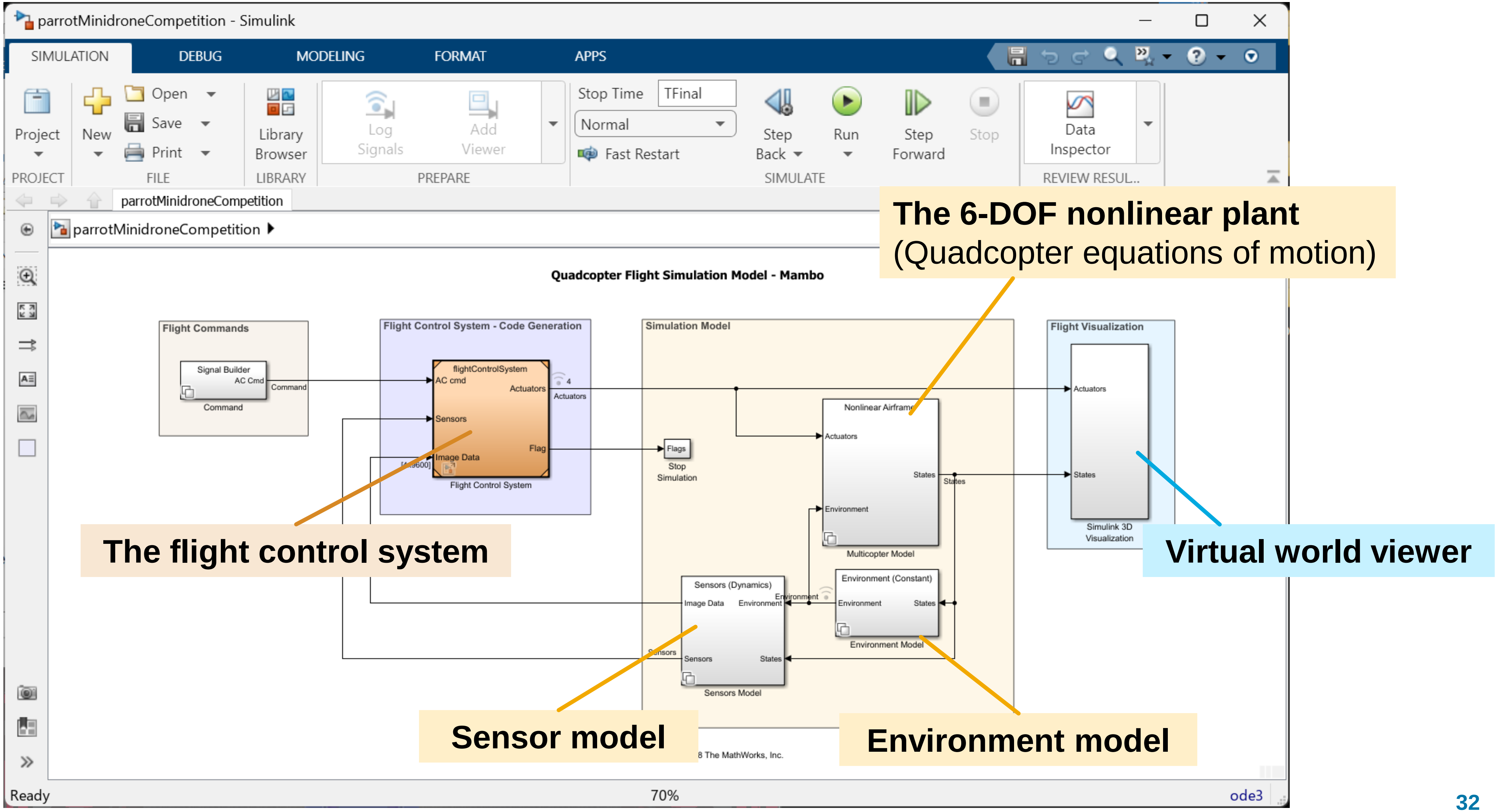


To continue working … just:

- Navigate to this folder
- Double click this to launch the Simulink project

Click

The Starting Point Model

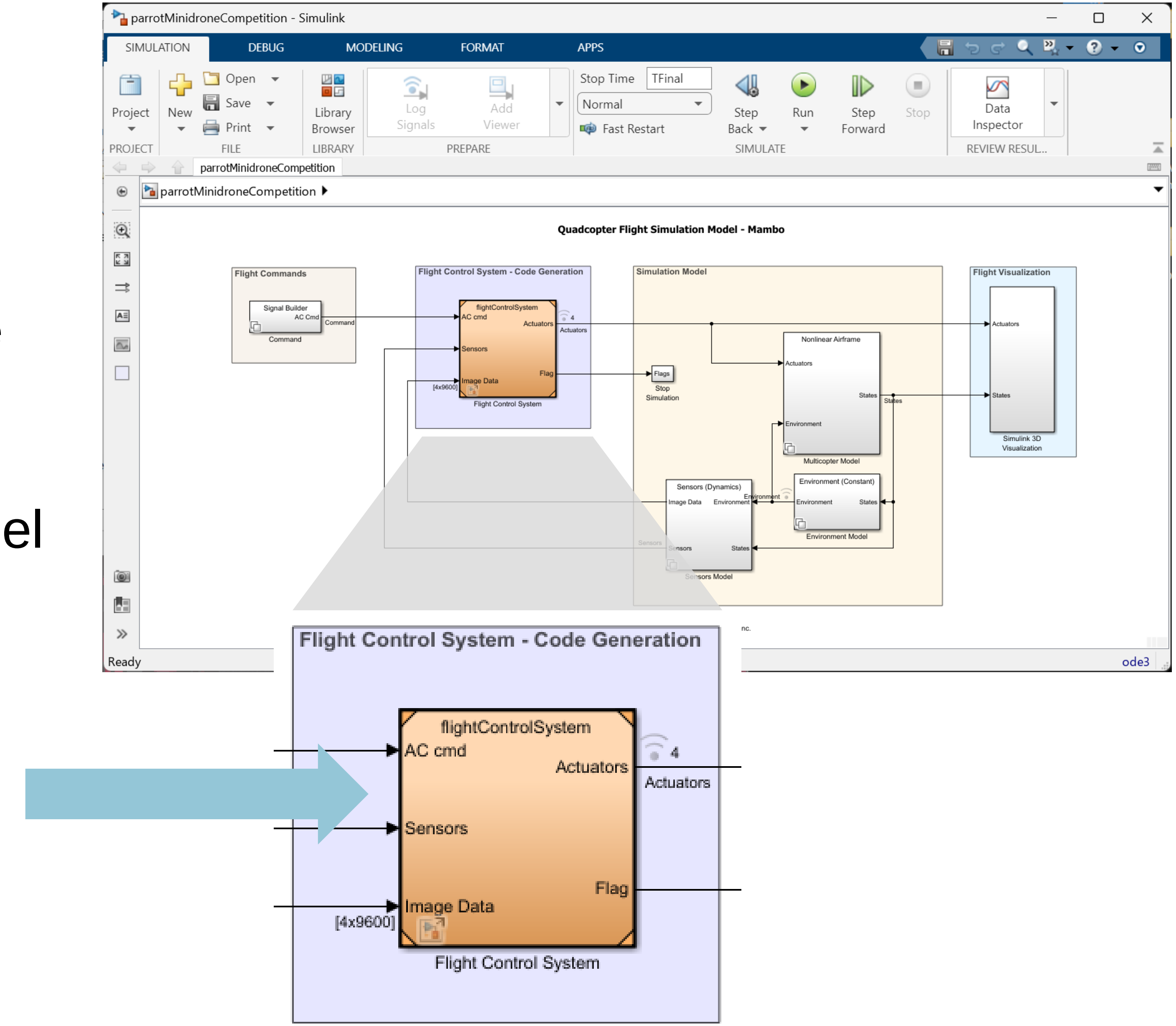


The Starting Point Model

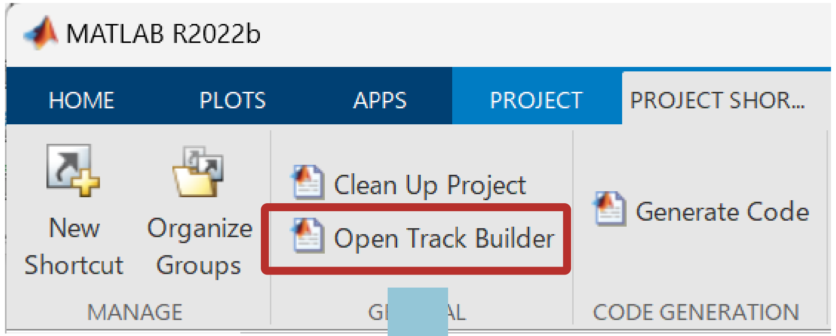
- Is a fully functioning model
  - It makes the quadcopter move towards a blue square placed in the arena

- You are required to edit this model
  - … to make it follow a ground path

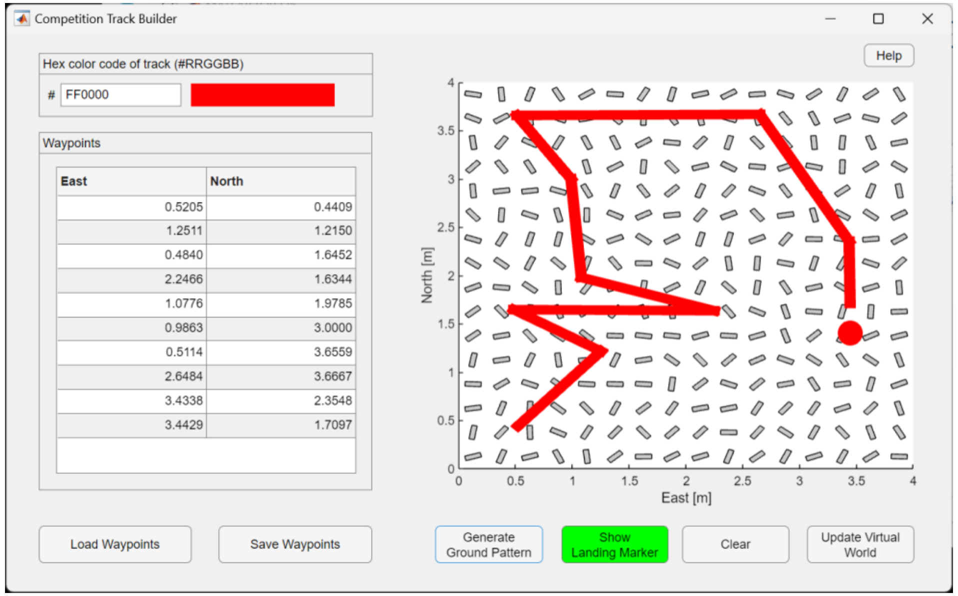
- **Specifically, your edits will be on this subsystem**
  - Flight Control System



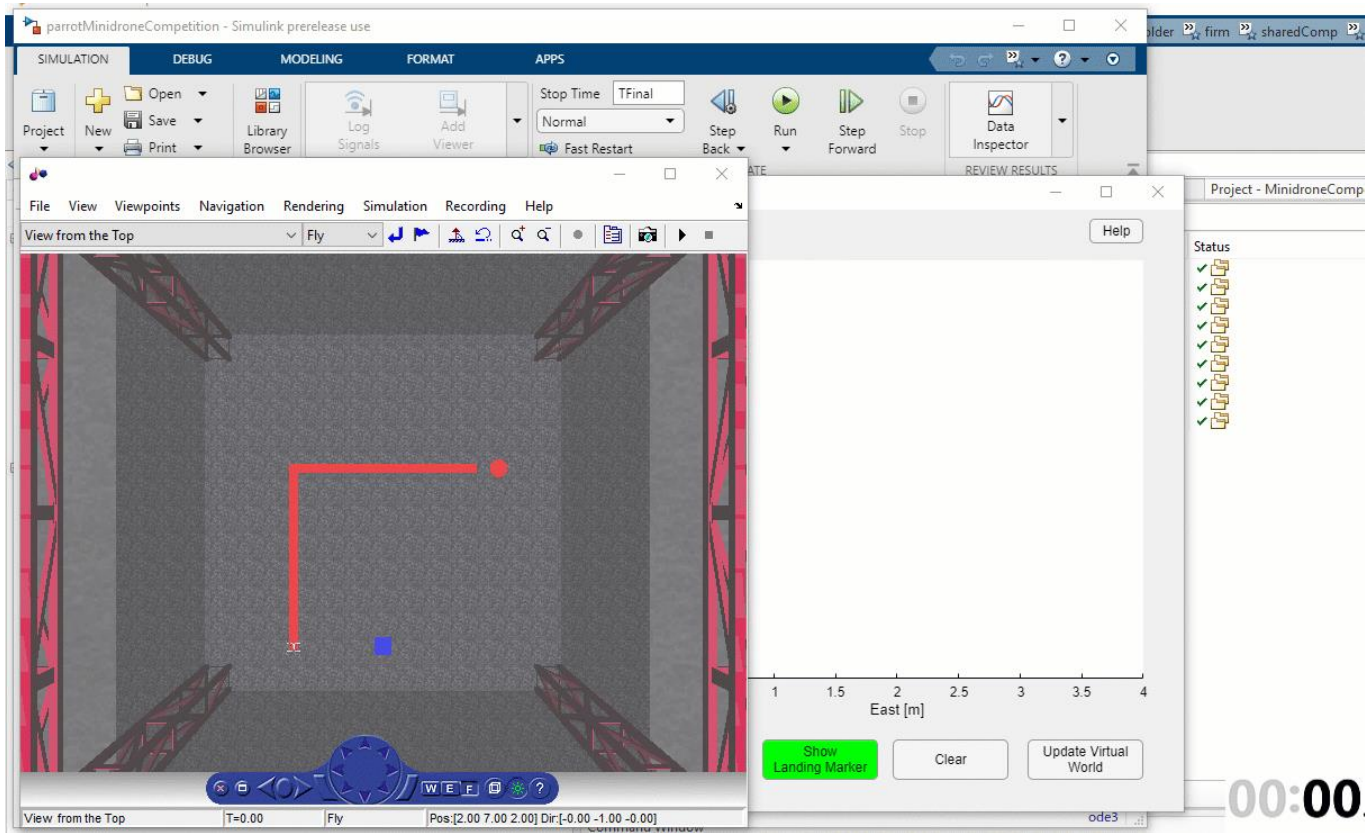
The Simulink Project Gives You A Track Building App



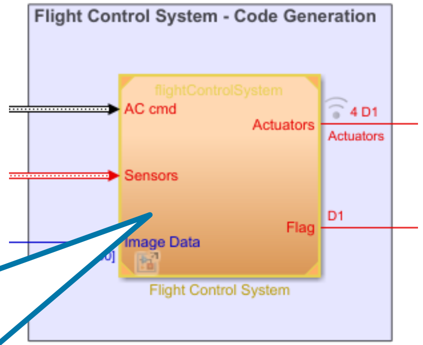
Track Builder App



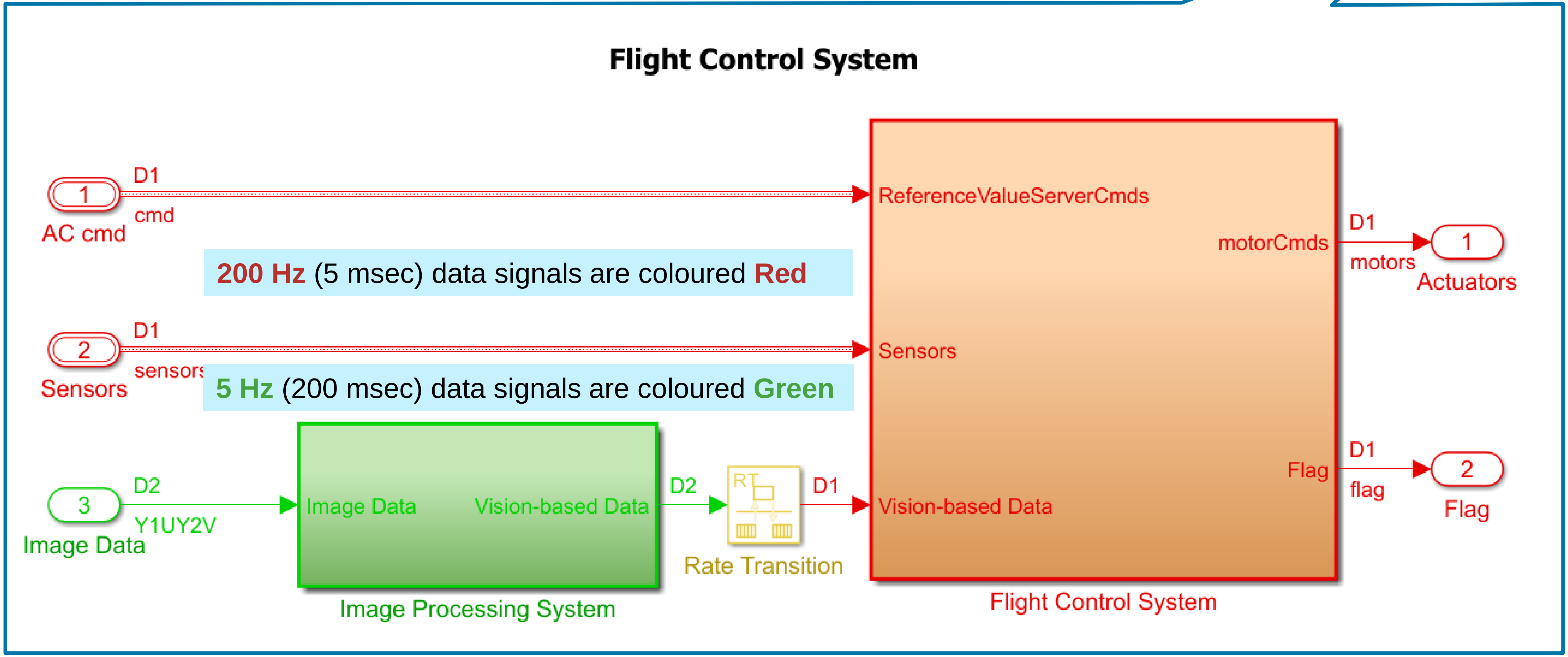
The **Track Builder App** allows you to design new paths to test your design on !



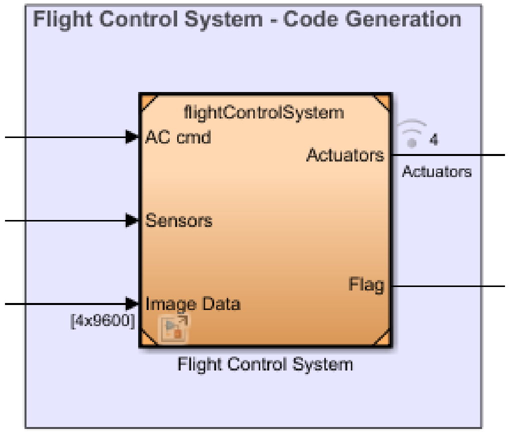
Flight Control System



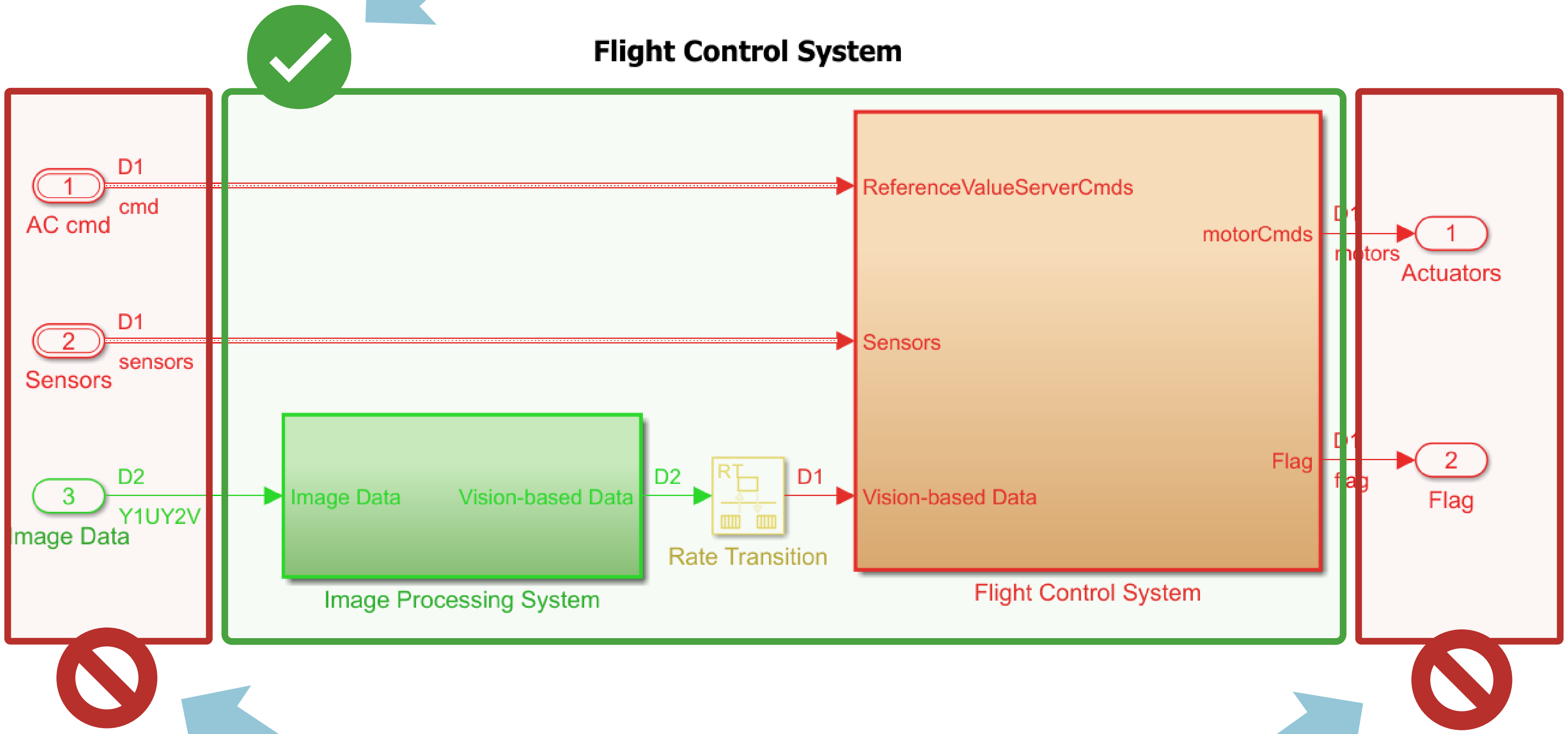
This is the subsystem that you must edit



Flight Control System



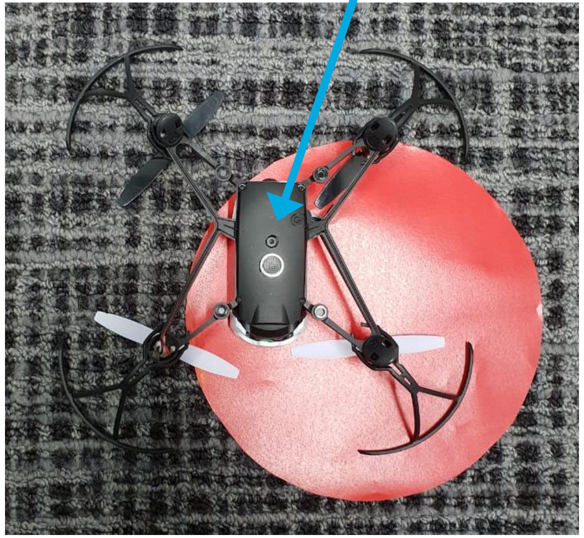
You CAN edit anything in here

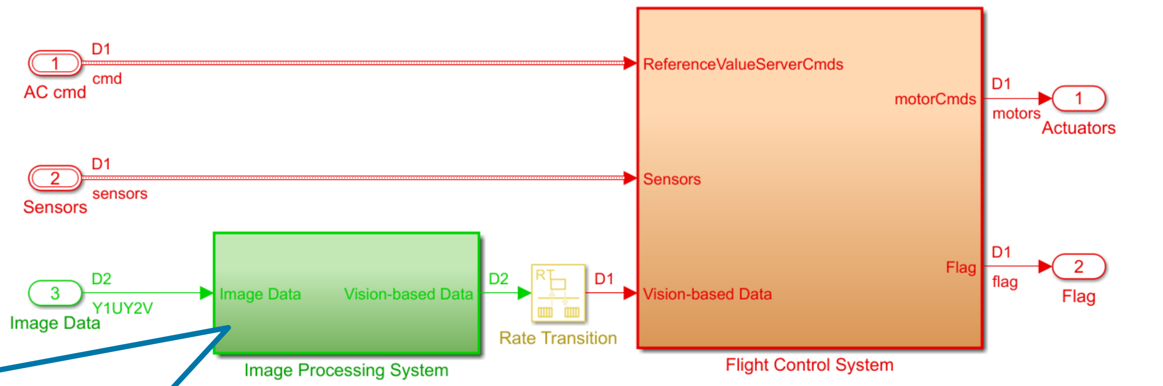


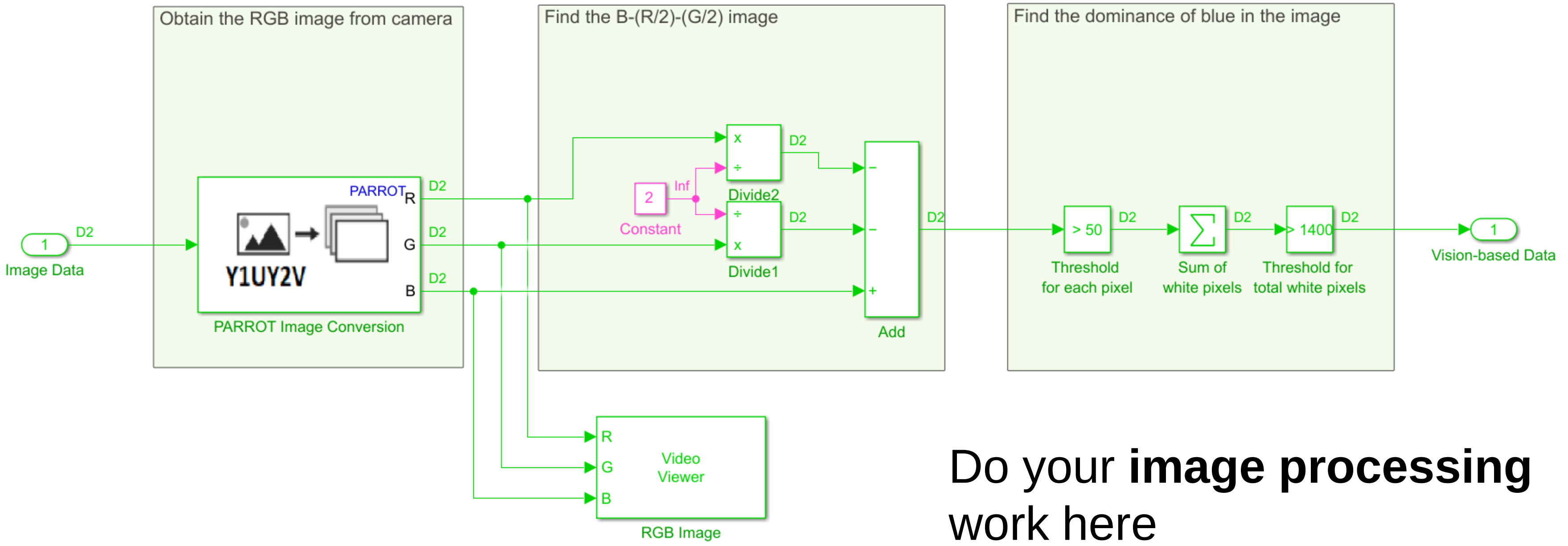
You can NOT edit the interfaces of the flight control system

Relevant Subsystems - Image Processing System

We're processing images from the drones camera, 5 times per second



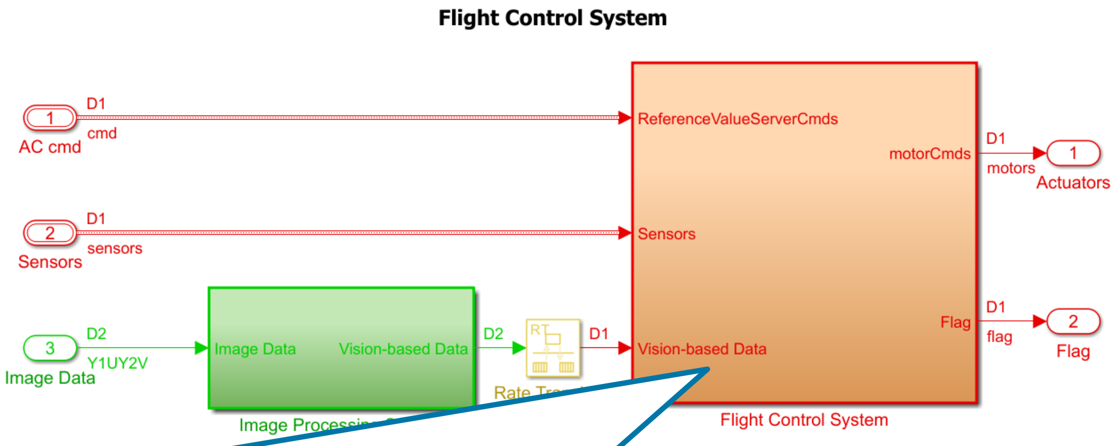


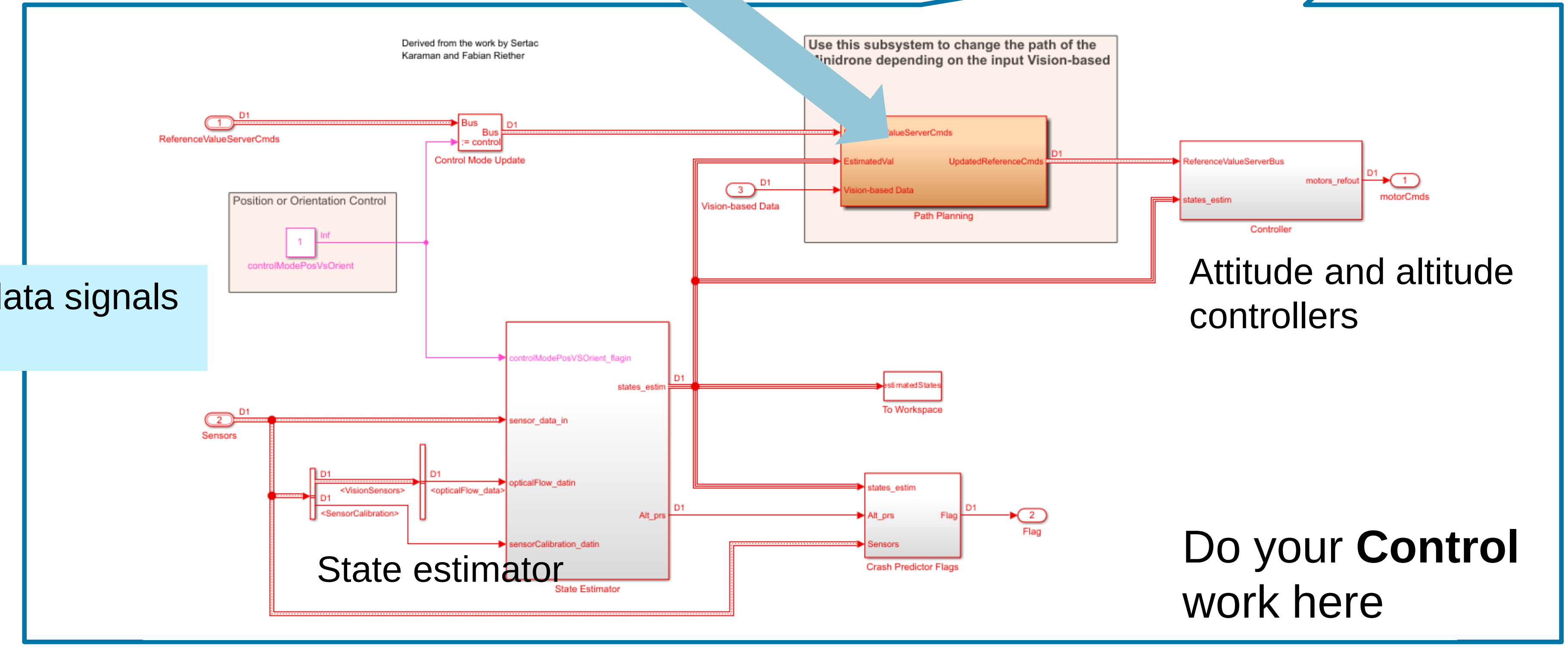


Do your **image processing** work here

Relevant Subsystems - Flight Control System

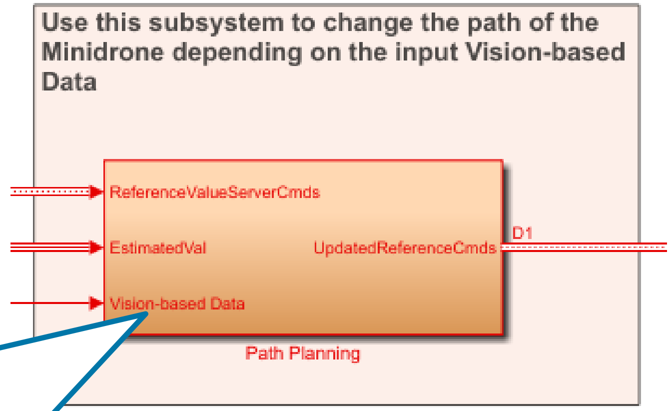
Most likely you'll be editing this **Path Planning** subsystem





200 Hz (5 msec) data signals are coloured **Red**

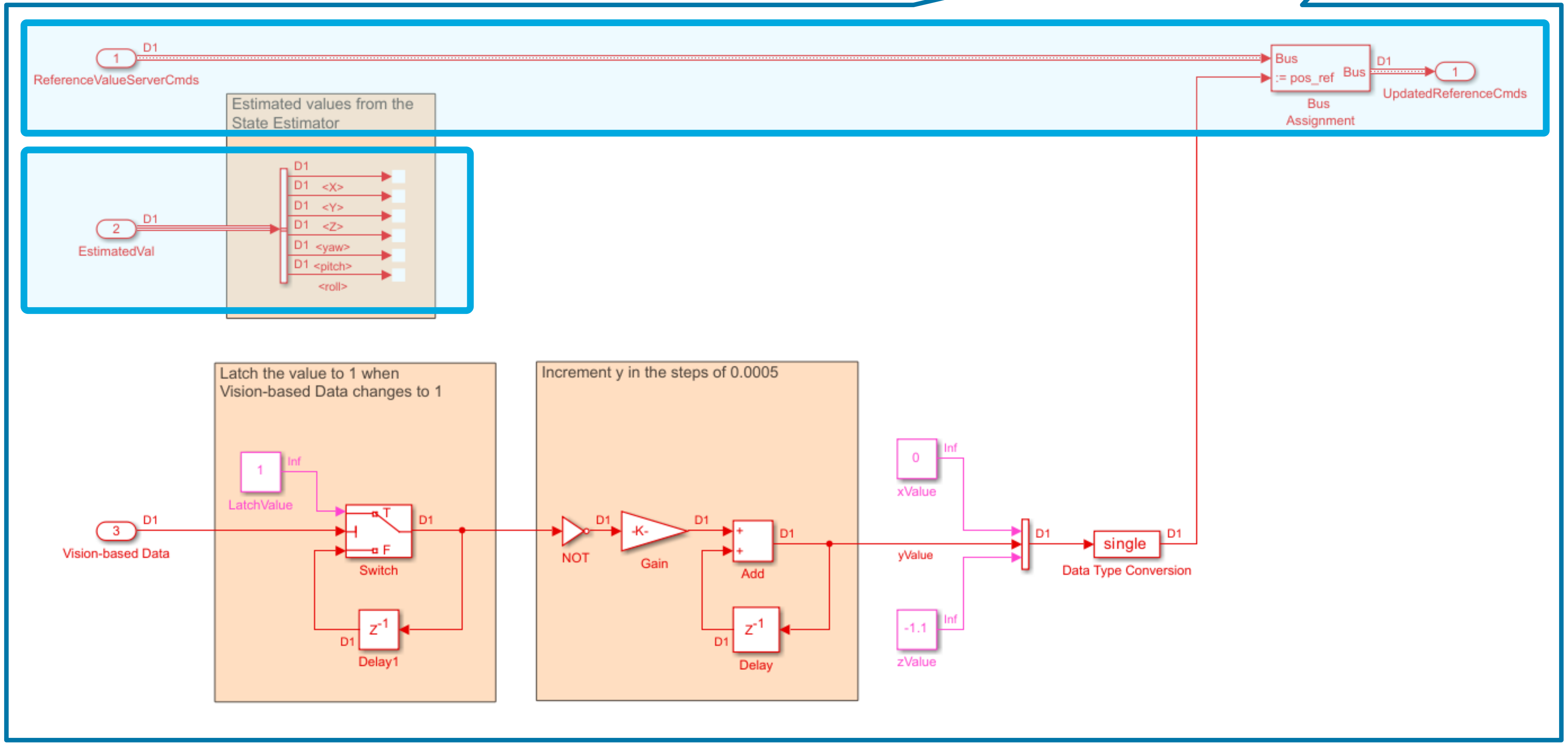
Relevant Subsystems - Path Planning



What are these BUS signals?

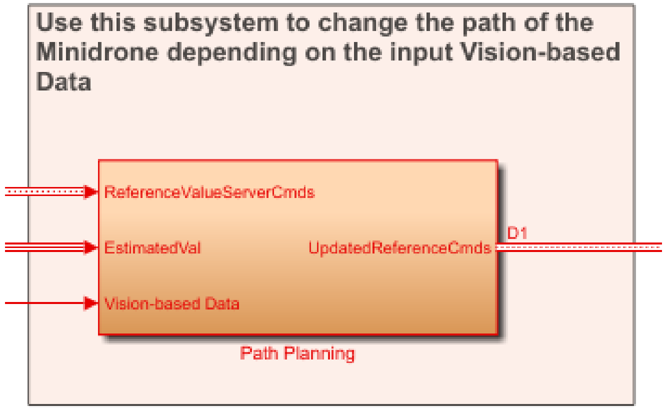
ReferenceValue ServerCmds

EstimatedVal

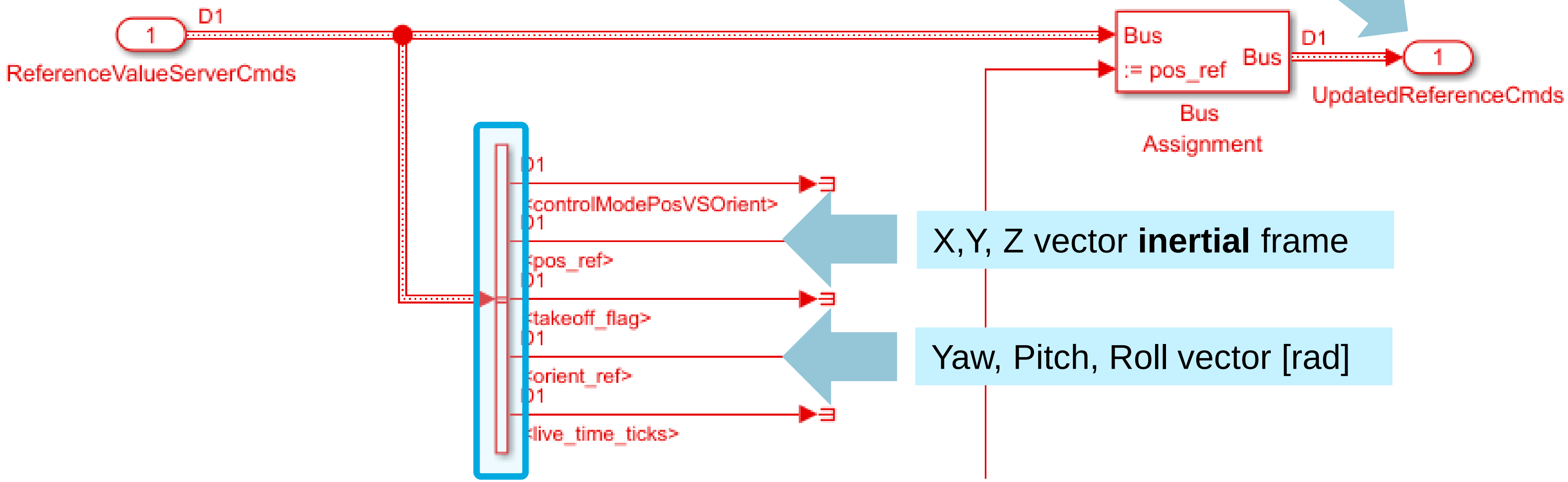


UpdatedReferenceCmds

Relevant Data Buses - Reference Value Commands



Specify the quadcopters **commanded position and orientation** by setting values in this BUS signal

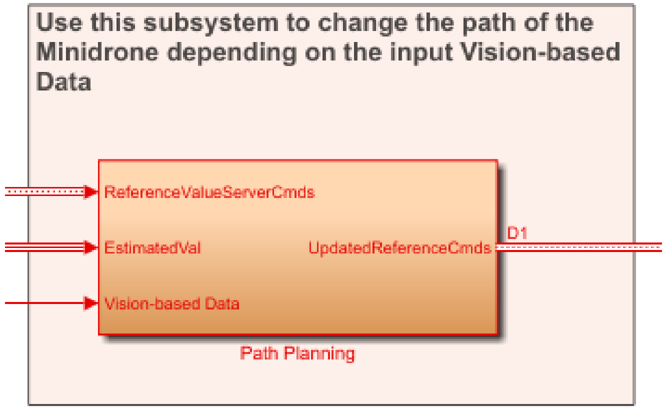


X,Y, Z vector **inertial** frame

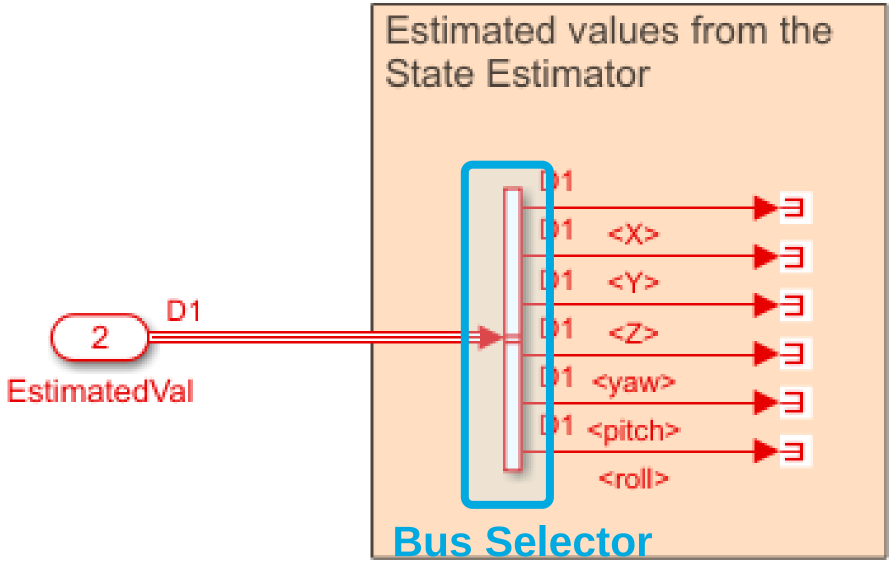
Yaw, Pitch, Roll vector [rad]

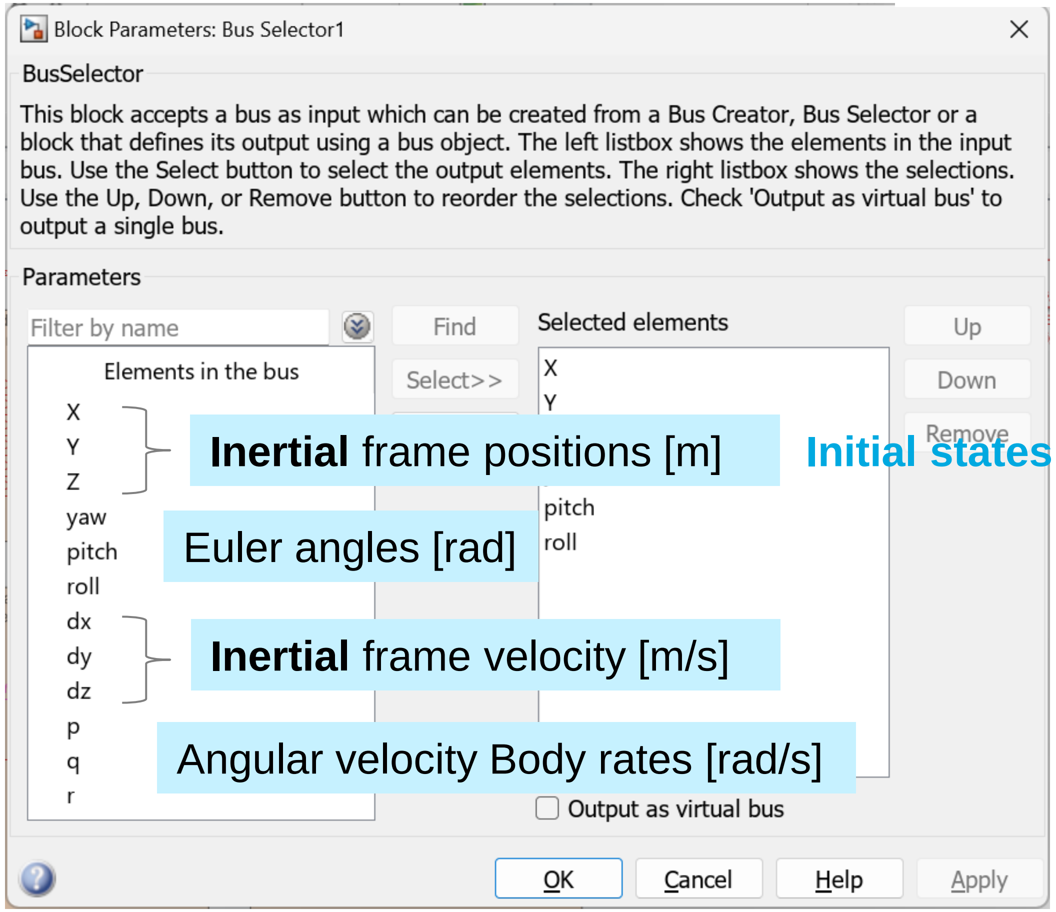
Bus Selector is added to extract each signals

Relevant Data Buses - Estimated State Values



Estimates of the **quadcopters states** are in this BUS signal





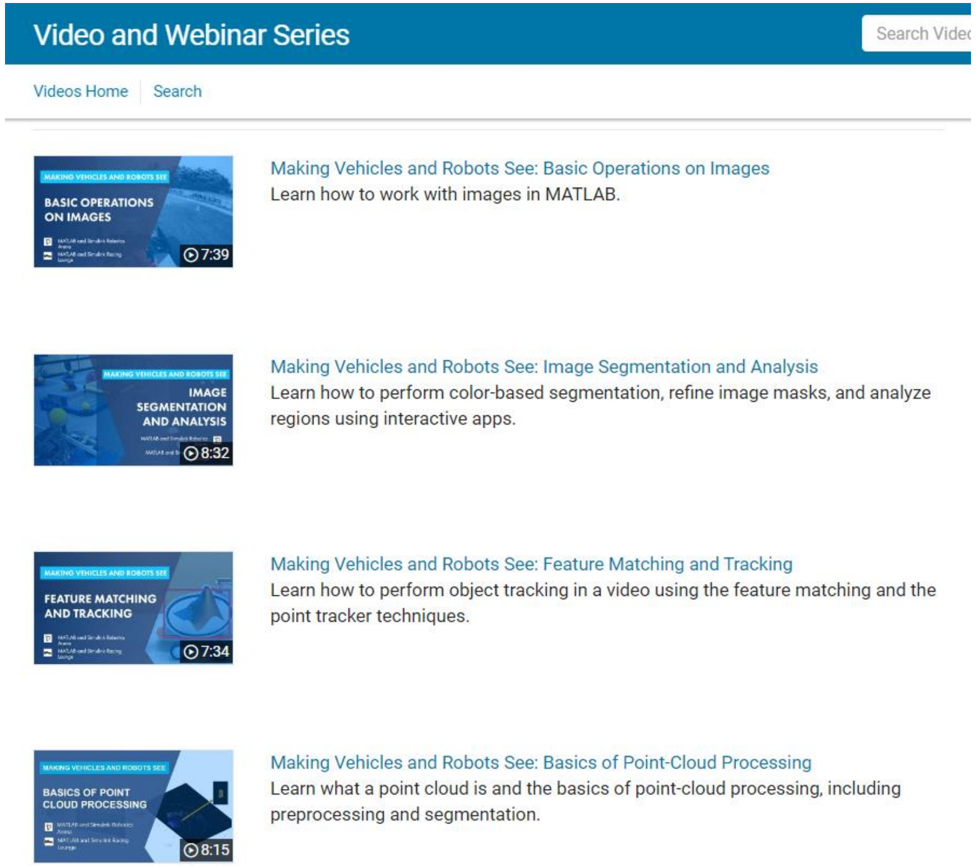
Initial states are zero

Appendix

- Your starting point model
- **Tips and Tricks**
  - First steps towards perception
    - detecting blobs and lines using image processing
  - First steps towards mission control
    - incorporating finite state machines
  - Useful Simulink modelling patterns
    - Bus objects for Stateflow and MATLAB Function blocks

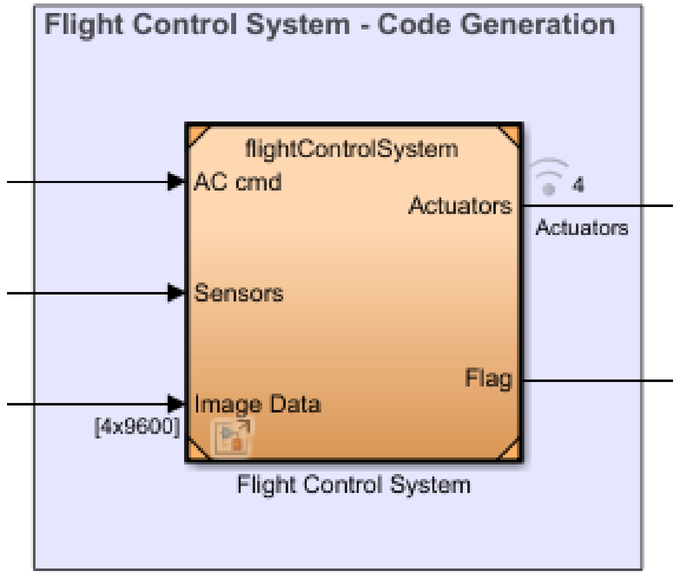
Perception with Image Processing

- Videos
  - Making Vehicles and Robots See
    - Making Vehicles and Robots See: Basic Operations on Images
    - Making Vehicles and Robots See: Image Segmentation and Analysis
- Documentation
  - Computer Vision
    - The Color Thresholder App
  - Blob Analysis
    - Computer Vision Toolbox
    - vision.BlobAnalysis
  - Authoring MATLAB Function Blocks in Simulink
    - Implement MATLAB Functions in Simulink
    - Control memory allocation
    - Create, modify, and manage types, such as bus objects
- Self-Paced Online Training Courses
  - MATLAB Academy
    - Simulink Onramp
    - Image Processing Onramp



MATLAB Functions

- When we generate a code from **Flight Control System**
  - We will NOT allow dynamic memory allocation for variable sized arrays
- You'll need to specify **upper bounds** for your local variables in your MATLAB code
- See the documentation
  - [Specify Upper Bounds for Variable-Size Arrays]()

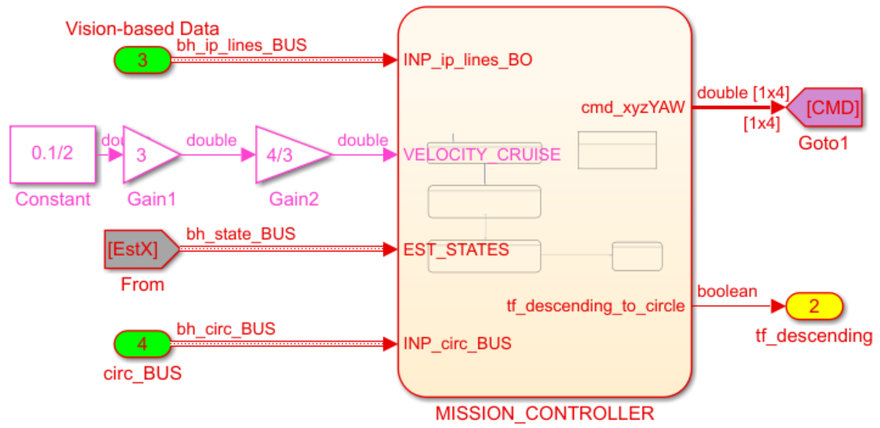


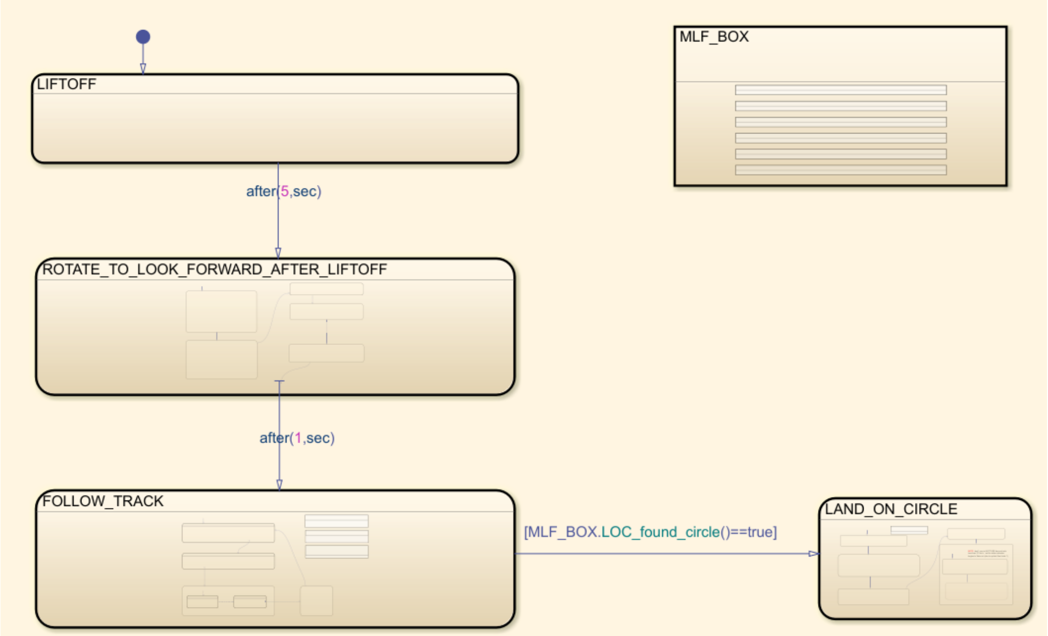

```
assert(NUM_LINES <= 50);

tf_is_line     = false(1, NUM_LINES);
count_of_lines = 0;
for kk = 1:NUM_LINES

    ABS_tmp_deg = abs(theta(kk));

    if( ABS_tmp_deg < 990 )
        count_of_lines = 1 + count_of_lines;
        tf_is_line(kk) = true;
    end
end
```

Mission Planning with Finite State Machines

- Decision logic within Simulink models
  - Is handled by a Simulink library called **Stateflow**

- [Stateflow]()
  - Finite state machines can be implemented using a graphical language





Stateflow

- Stateflow
  - Getting started
  - Semantics
  - Temporal Logic
  - Execution of a Stateflow chart
  - Reuse MATLAB code in your charts
  - Access Bus Signals

- Self-Paced Online Training Courses
  - MATLAB Academy
    - Stateflow Onramp