MATLAB EXPO 2019

Model-Based Design for Autonomous Aerial Systems

Naga Chakrapani Pemmaraju

Applications

Inspection

Monitoring

Surveillance

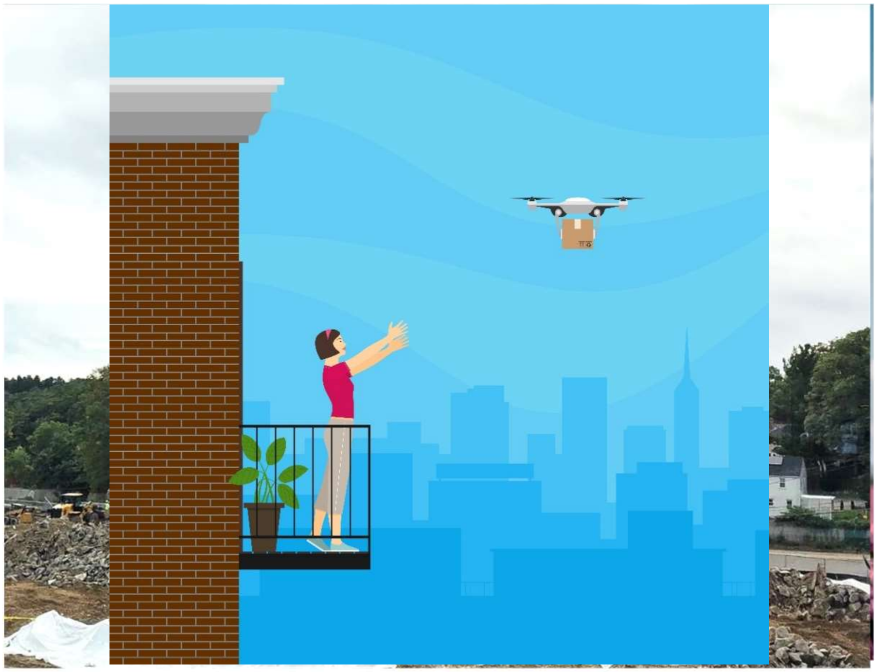
Mapping

Search & Rescue

Surveying

Aerial Photography

Package Delivery



Industries

Agriculture

Mining

Construction

Oil & Gas

Power Grid

Forest & Wildlife

Archaeology

Real Estate

E-commerce

Korean Air Speeds UAV Flight Control Software Development and Verification with Model-Based Design

Challenge

Develop and verify flight control software for unmanned aerial vehicles

Solution

Use Model-Based Design to design and simulate flight control laws and operational logic, generate and verify production code, and conduct HIL tests

Results

- 100% of run-time errors in handwritten code identified and eliminated
- Development effort reduced by 60%
- Costly flight tests minimized

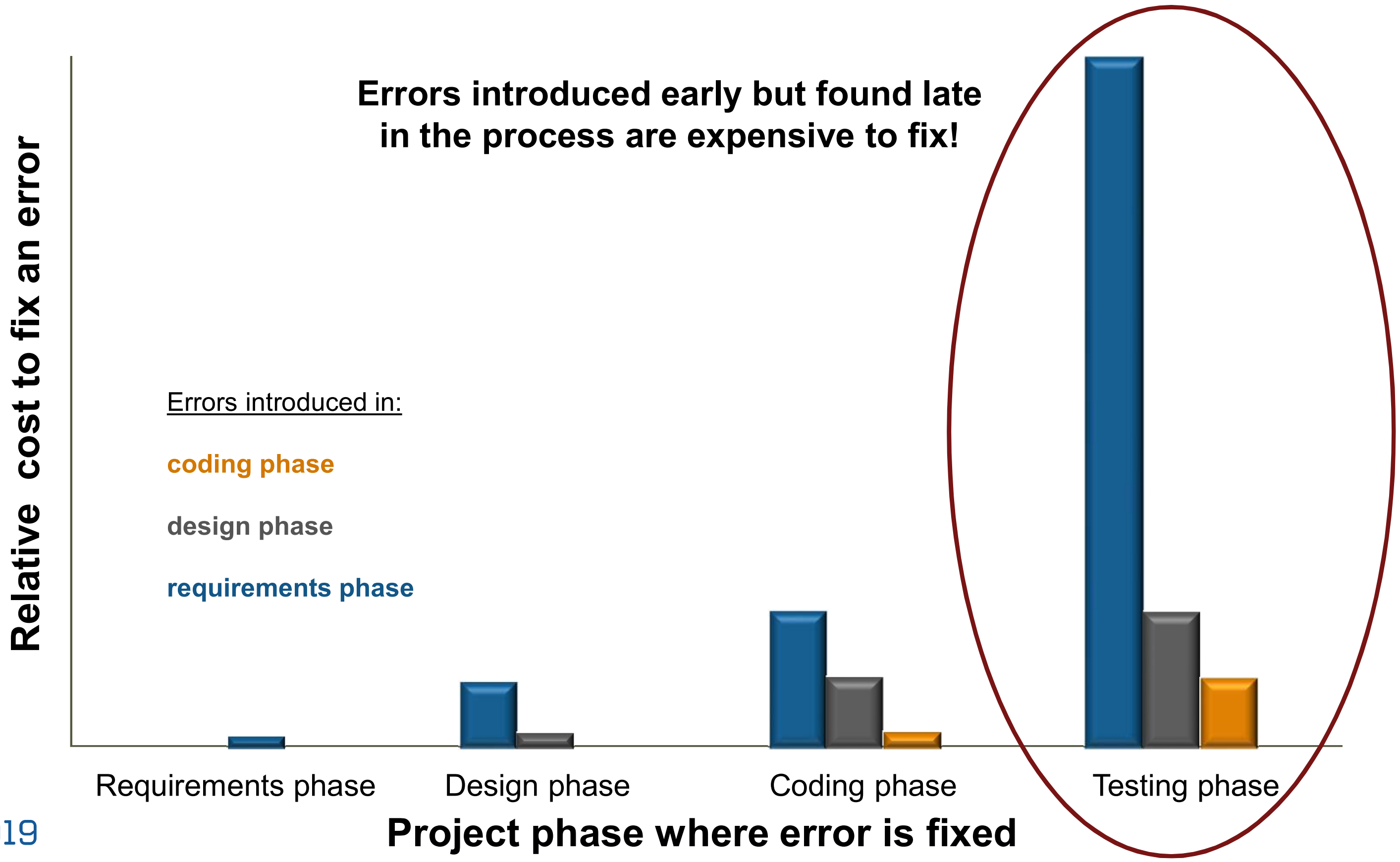
A Korean Air unmanned aerial vehicle.

> *"The model reuse and efficiency improvements enabled by MATLAB and Simulink save time and lower costs. We estimate a time savings of more than 50% is achievable with Model-Based Design compared with hand-coding, and the advantages of Model-Based Design increase with the complexity of the project."*
>
> *- Jungho Moon, Korean Air*

Link to user story

Avoiding simulation before deployment can be dangerous

Fixing bugs late is very expensive



Source: Return on Investment for Independent Verification & Validation, NASA, 2004.

What are we discussing today?

Algorithms Development

- Sensing & Perception
- Planning & control

Simulation

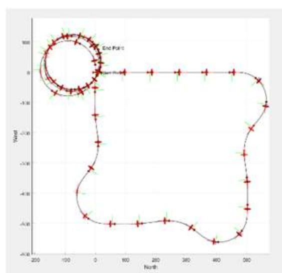
- Tuning Parameters
- Visualization

Deployment

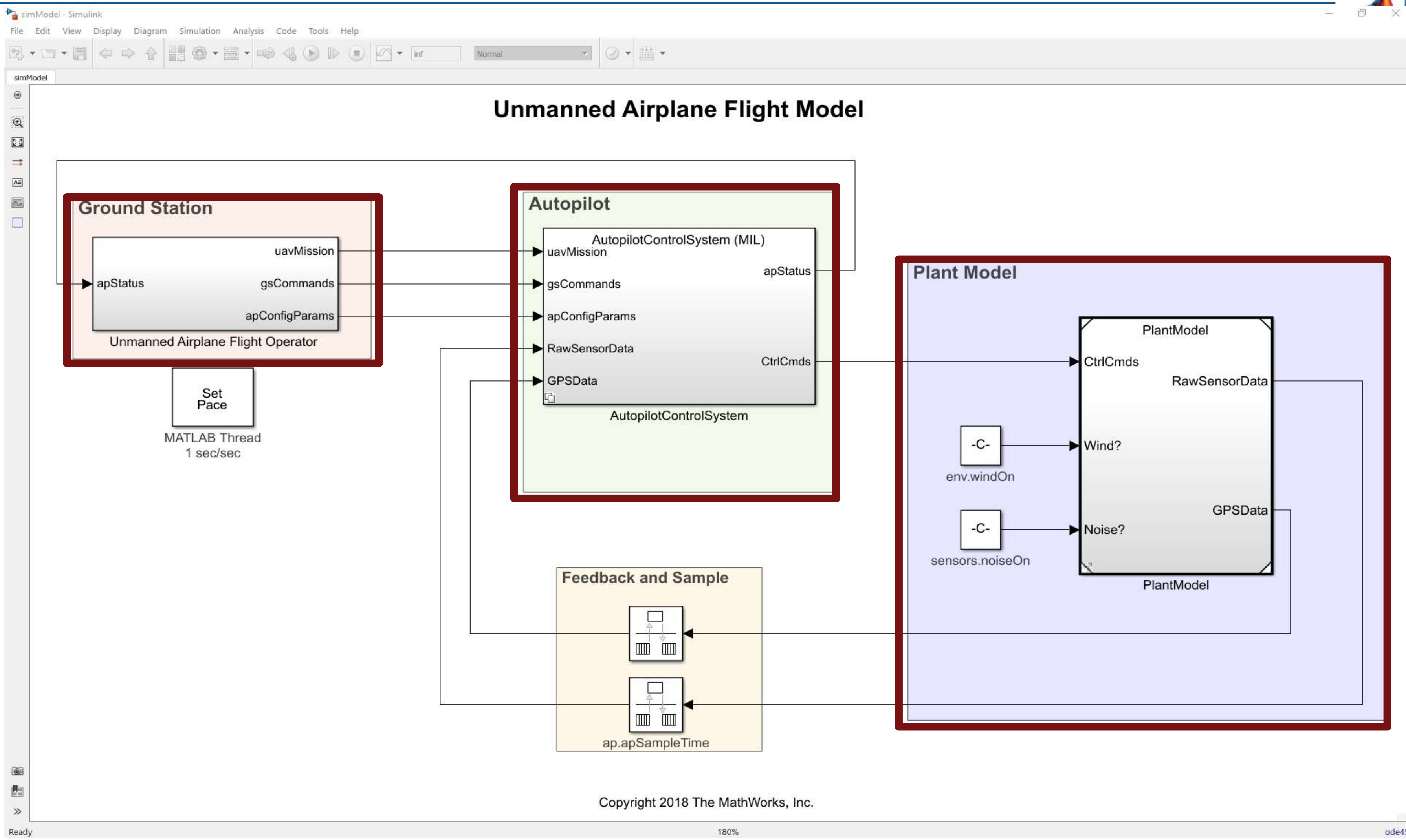
- Code Generation
- Hardware and Flight Tests

Data Import

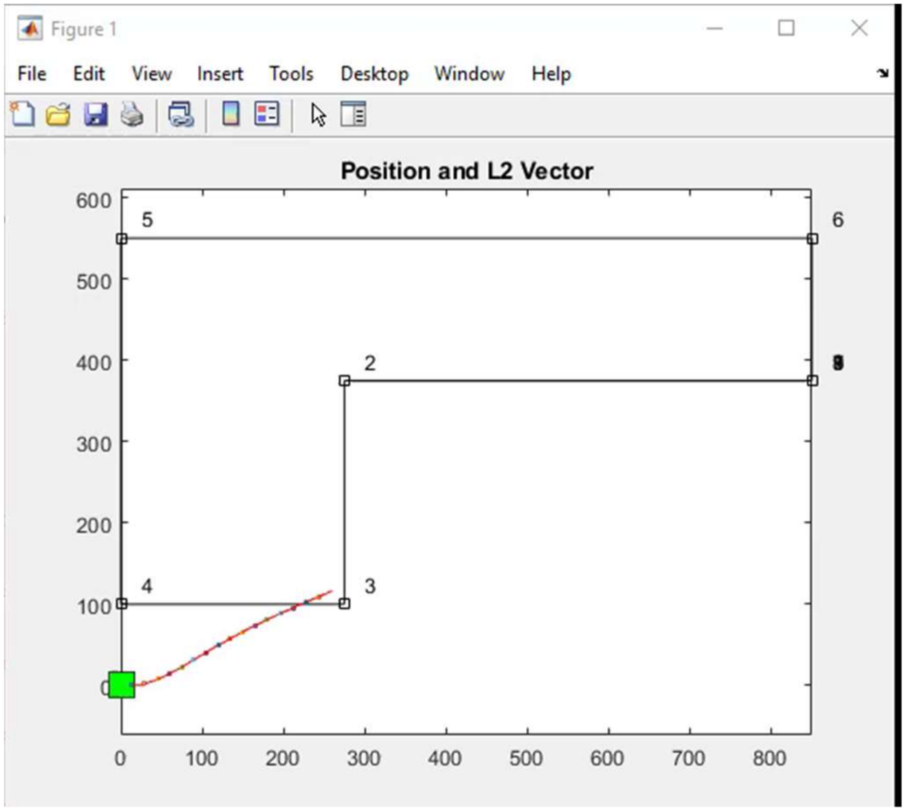
- Data Logging & Processing
- Interactive Plots



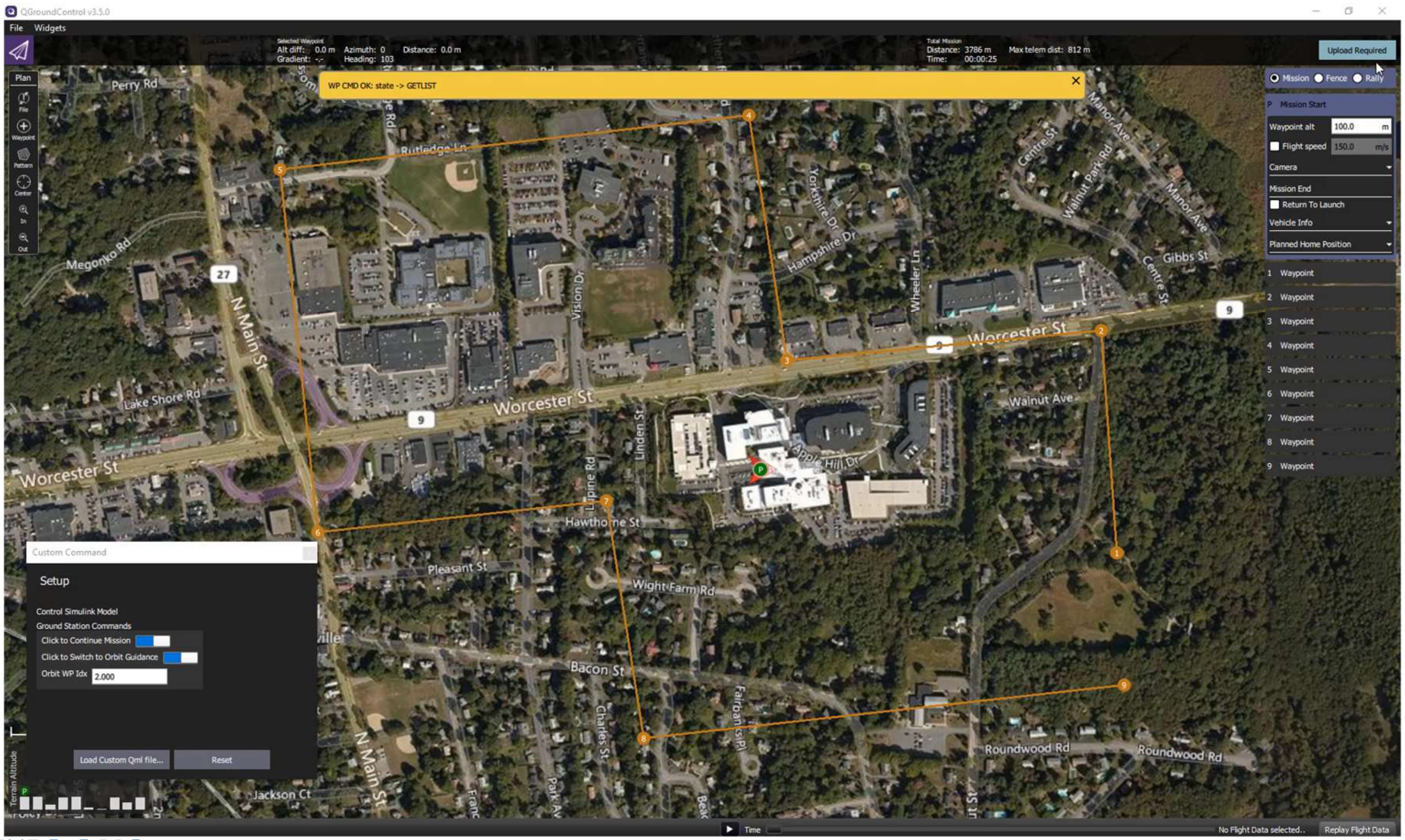
Waypoint following


simModel - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
Unmanned Airplane Flight Model
Ground Station
uavMission
apStatus
gsCommands
apConfigParams
Unmanned Airplane Flight Operator
Set Pace
MATLAB Thread
1 sec/sec
Autopilot
AutopilotControlSystem (MIL)
uavMission
apStatus
gsCommands
apConfigParams
RawSensorData
CtrlCmds
GPSData
AutopilotControlSystem
Plant Model
PlantModel
CtrlCmds
RawSensorData
-C-
Wind?
env.windOn
GPSData
-C-
Noise?
sensors.noiseOn
PlantModel
Feedback and Sample
ap.apSampleTime
Copyright 2018 The MathWorks, Inc.
Ready
180%
ode45

Desktop Simulation



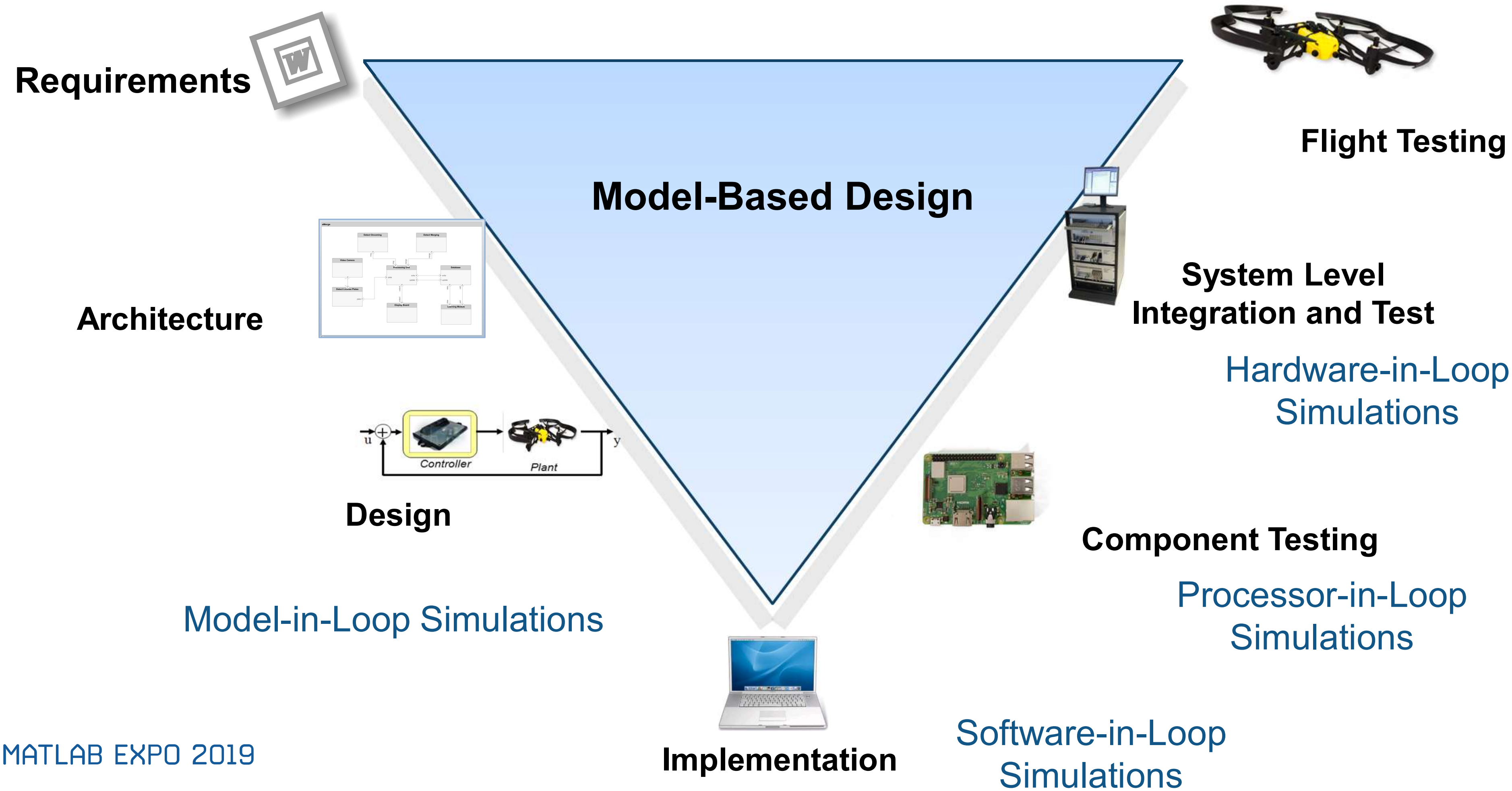
Co-Simulation with QGroundControl



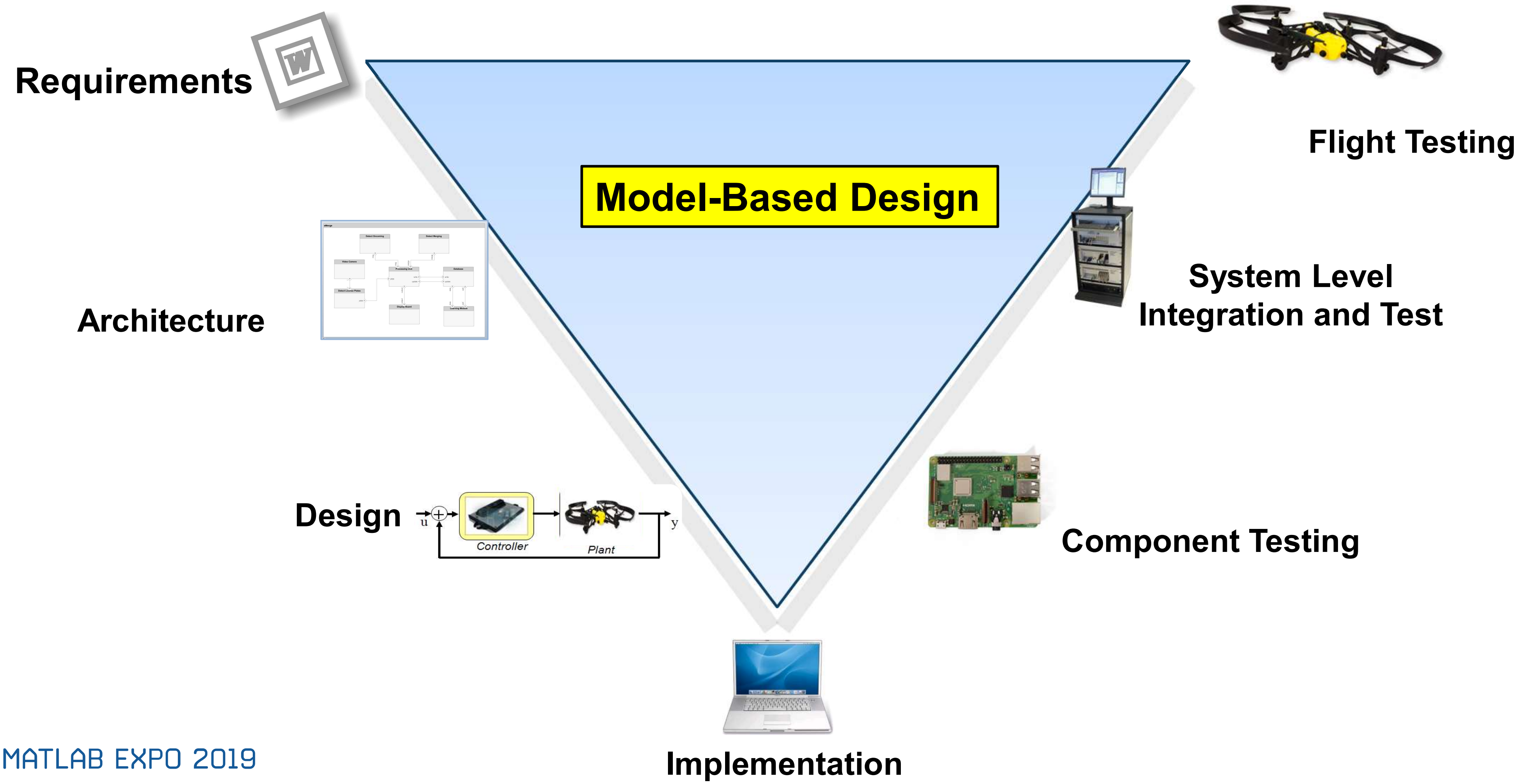
Key takeaways

- Simulink as a platform for System level design of UAV
- Model environmental effects and 6DOF aircraft simulations
- Design autopilot and test its performance under simulated flight conditions
- Deploy and test correctness of Flight Controller's generated code

V Development Cycle

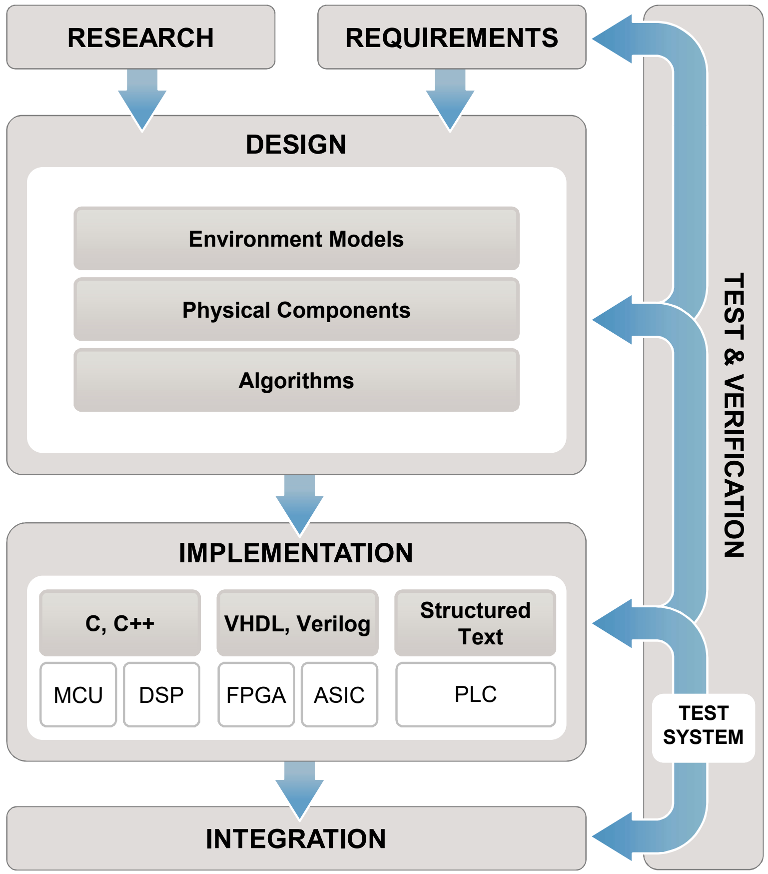


V Development Cycle



Model-Based Design with MATLAB and Simulink

- **Cost**
  - Minimize prototypes and rework
  - Facilitates design reuse
- **Schedule**
  - Shortens time-to-market
  - Enhances team communication
- **Performance**
  - Fosters innovation
  - Improves quality



Airnamics Develops Unmanned Aerial System for Close-Range Filming with Model Based Design

Challenge

Design and develop an unmanned aerial camera motion system for close-range aerial filming

Solution

Use Model-Based Design with MATLAB and Simulink to accelerate the design, debugging, and implementation of the vehicle's fly-by-wire and flight management system software

Results

- Development time reduced by one year or more
- Coding errors eliminated
- 80% model reuse achieved

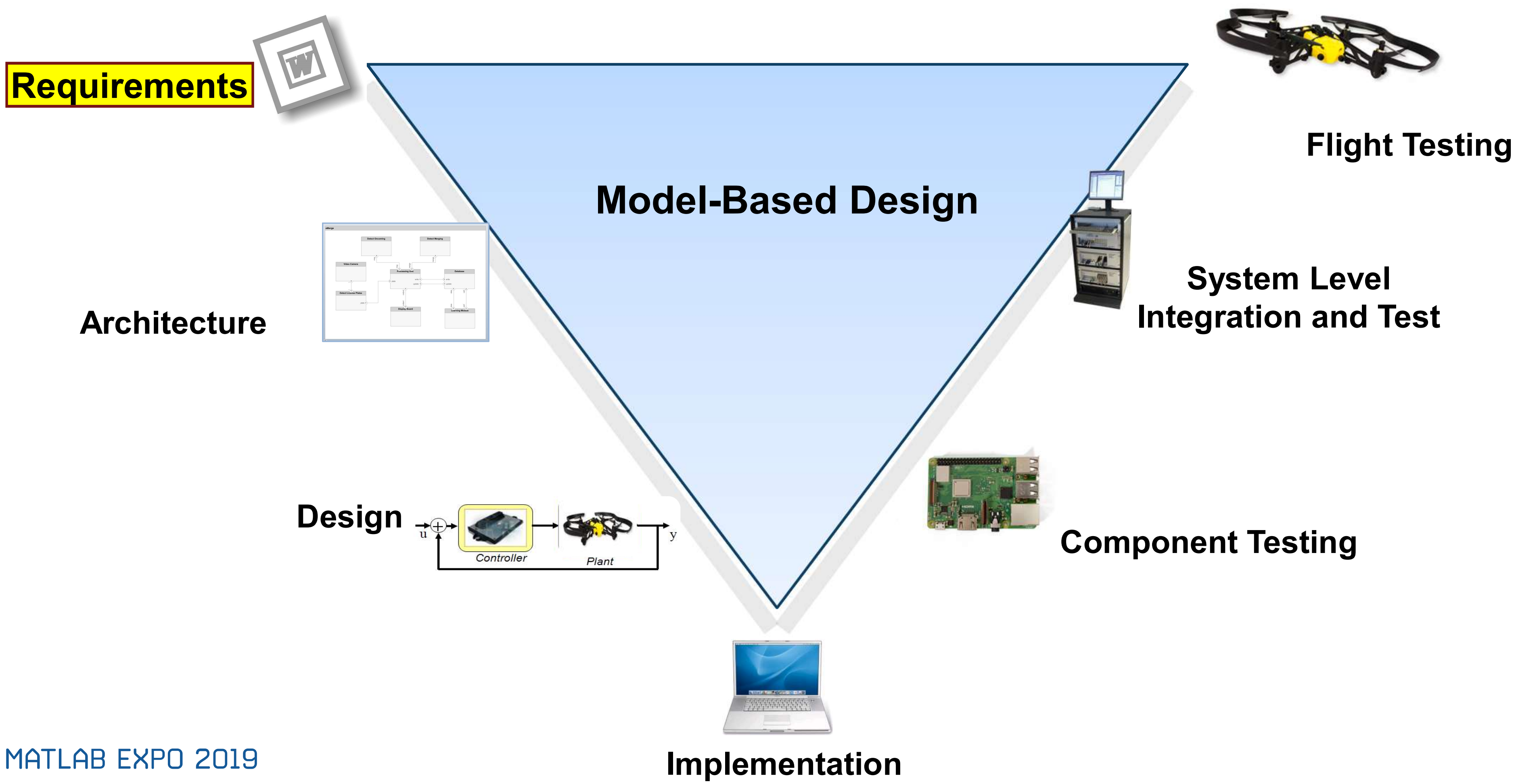
Airnamics co-founders Marko Thaler and Zoran Bjelić with the R5 MSN1 prototype after its first flight.

"With Model-Based Design our three-engineer team found more than 95% of control software bugs before the first flight. We used the test flights to increase our Simulink models' fidelity and isolate remaining bugs with high precision. The result is a safer, more reliable, and higher-quality product."

- Marko Thaler, Airnamics

Link to user story

V Development Cycle

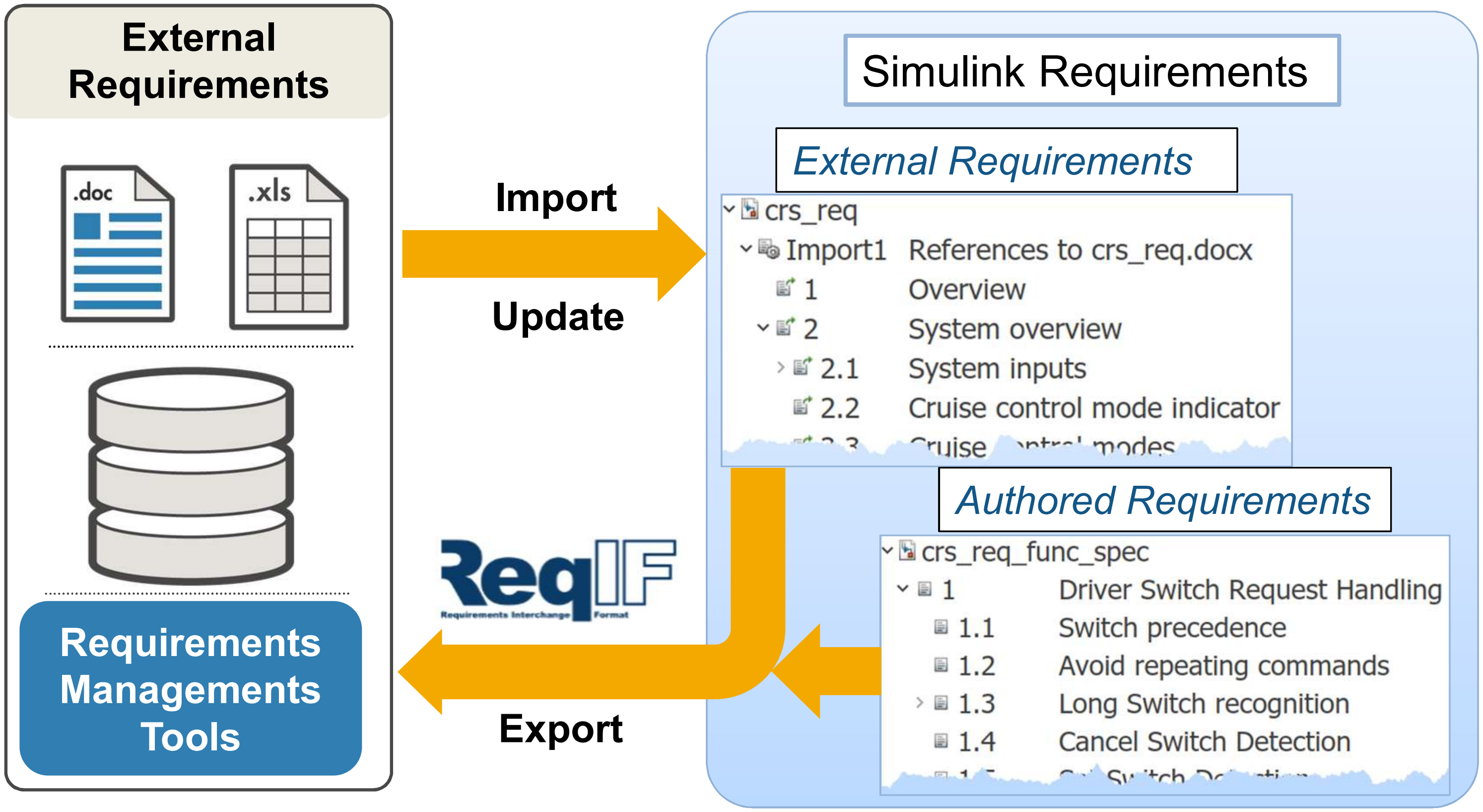


Requirements

Sample Requirements for the flight controller

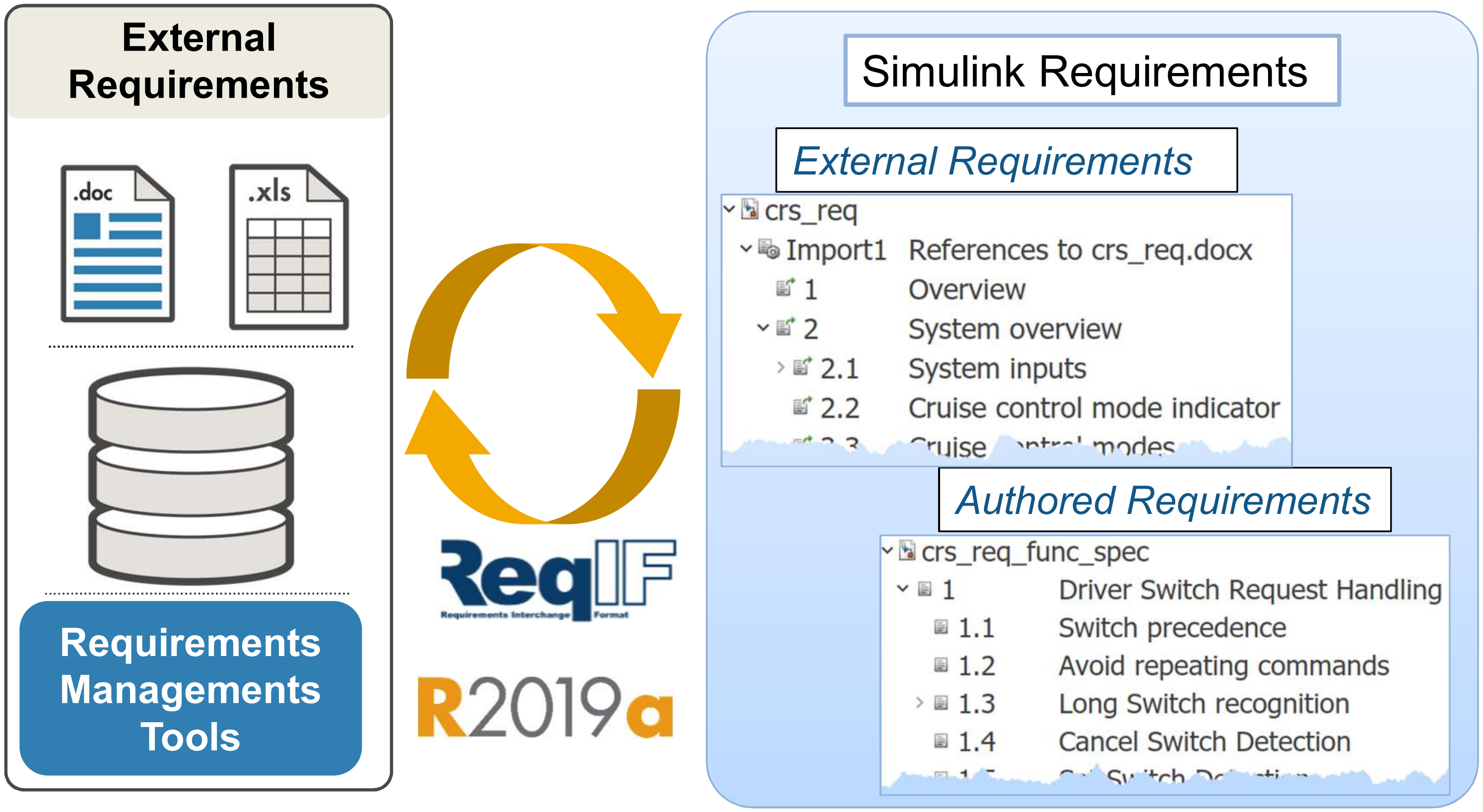
Requirement	Description
Flight Controller	Aircraft shall provide an off-the-shelf computer to work as a flight controller to autonomously control the aircraft from launch to recovery.
Sensors: Temperature Compensation	All sensor drivers shall account for temperature-induced bias in their readings.
Guidance: Waypoint Traversal	Aircraft shall follow a given set of way-points provided by an operator via the ground control station software in triads of latitude, longitude, and height.
Control: Airspeed Controller	Aircraft shall be able, within its physical limits, to reach and hold a commanded air-speed from an outside component.

Integrate with requirements tools and author requirements



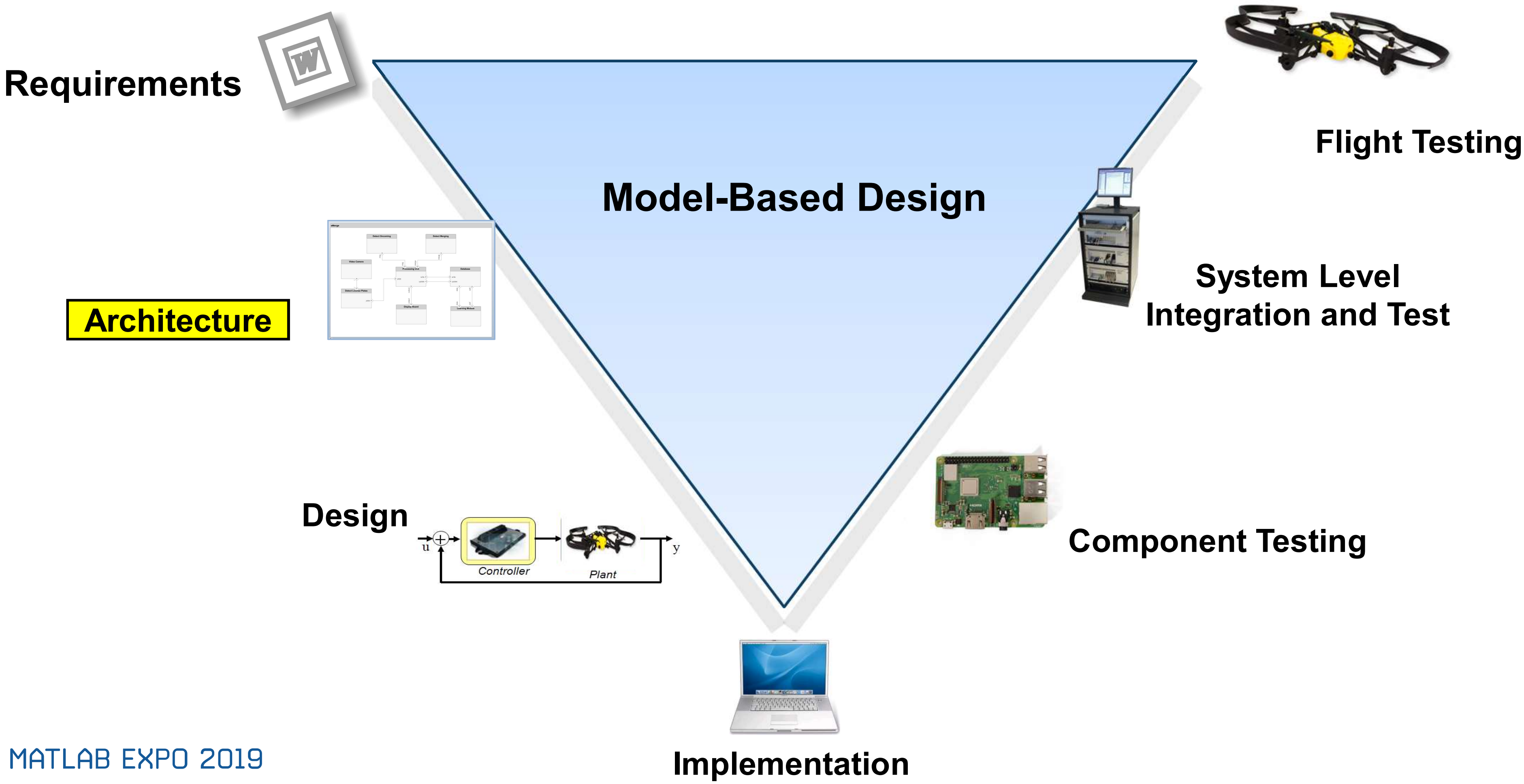
- Import from:
  - Word / Excel
  - IBM® Rational® DOORS®
  - ReqIF™ standard
- Update synchronizes changes from source
- Edit and add further details to import R2019a
- Author requirements
- Export ReqIF R2019a
  - Enables roundtrip with external tools

Roundtrip workflow with external tools thru ReqIF



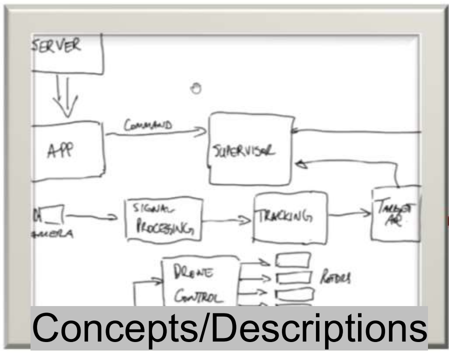
- Import from:
  - Word / Excel
  - IBM® Rational® DOORS®
  - ReqIF™ standard
- Update synchronizes changes from source
- Edit and further details to import R2019a
- Author requirements
- Export ReqIF R2019a
  - Enables roundtrip with external tools

V Development Cycle



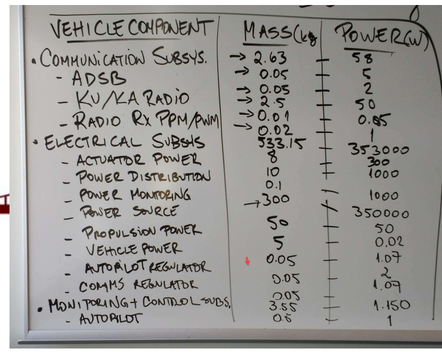
What goes into the bridge?

Be Intuitive

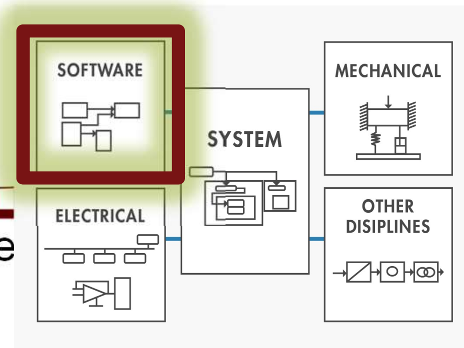


Concepts/Descriptions

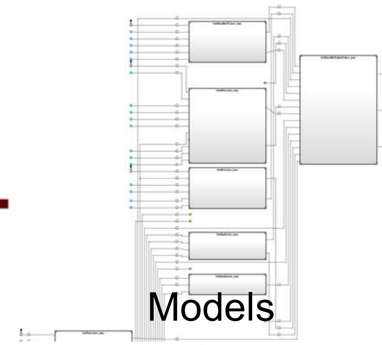
Facilitate Analysis



Tackle Complexity



Enable Implementation



Models

Digital Thread for Traceability

1. Functional Requirements

1.1. Normal Mode of Operation

During the normal mode of operation, the Fault Tolerant Fuel Control System shall determine the fuel rate which is injected at the valves.

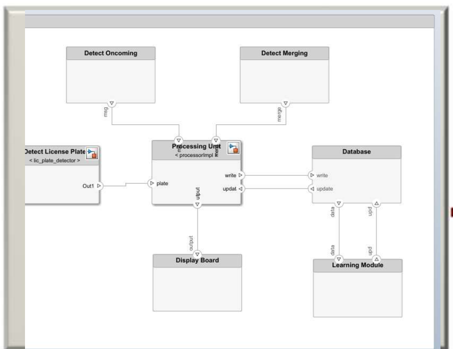
1.1.1. Stoichiometric mixture ratio

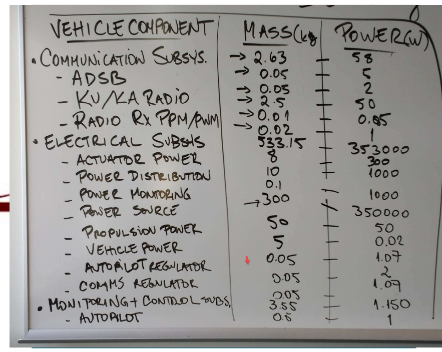
During normal model of operation, the System shall maintain the stoichiometric mixture target ratio of 14.6.

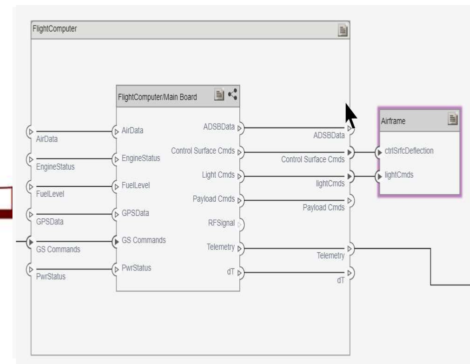
1.1.2. Oxygen Sensor (EGO)

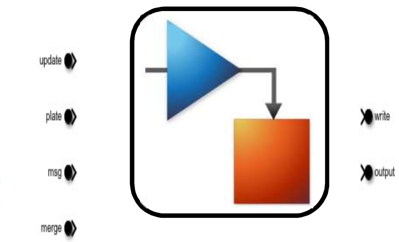
MathWorks Solution: System Composer R2019a and

Be Intuitive ✓Facilitate Analysis ✓Tackle Complexity ✓Enable Implementation



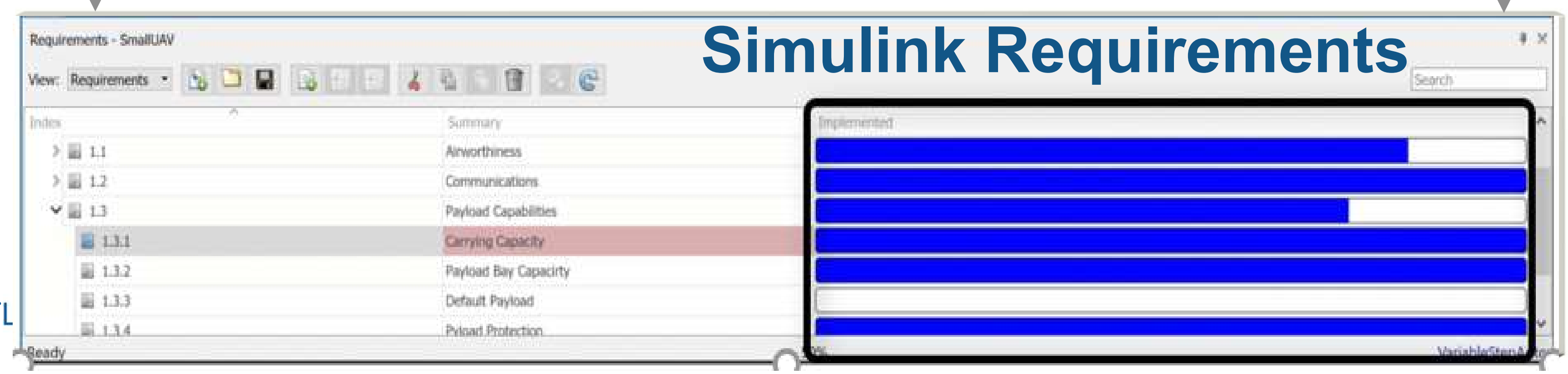






Simulink

Requirements Coverage Reporting and Impact Analysis



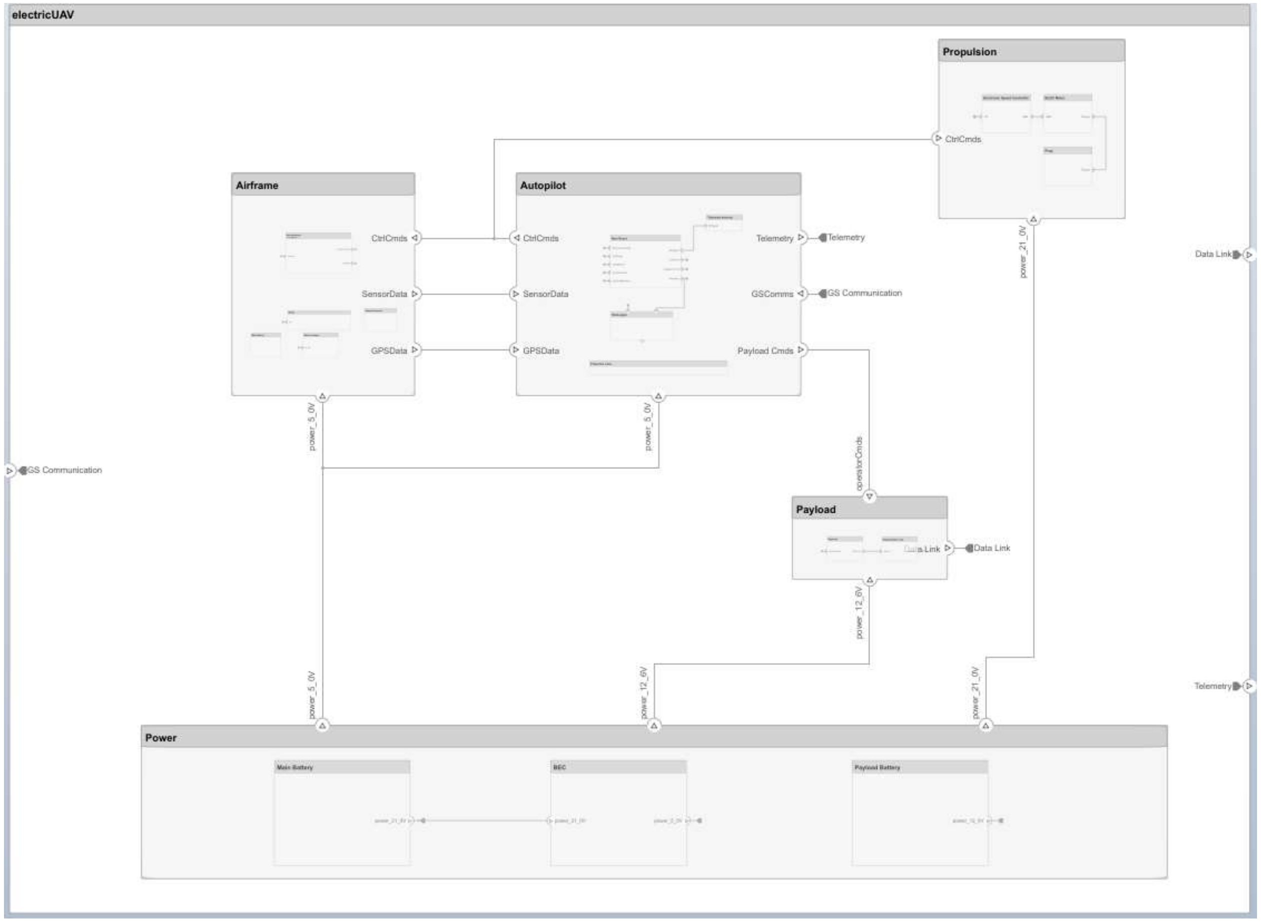
Simulink Requirements

MATL

Electric UAV Architecture modeling using System Composer

Intuitively design system and software architectures

R2019a

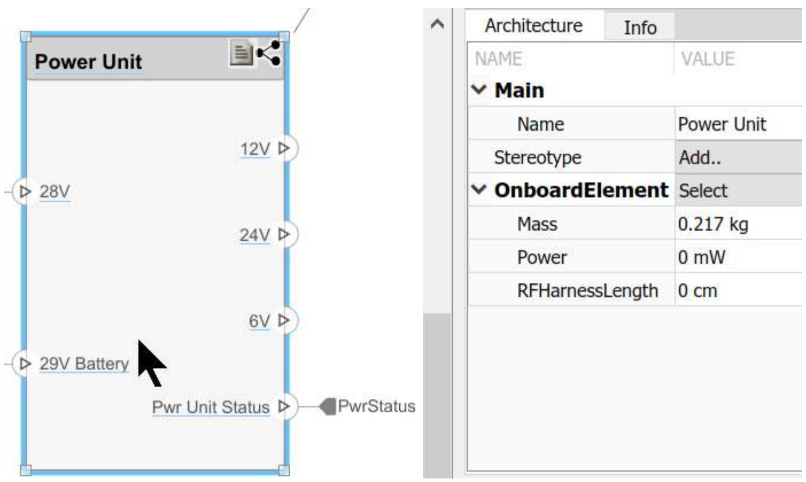


System Composer

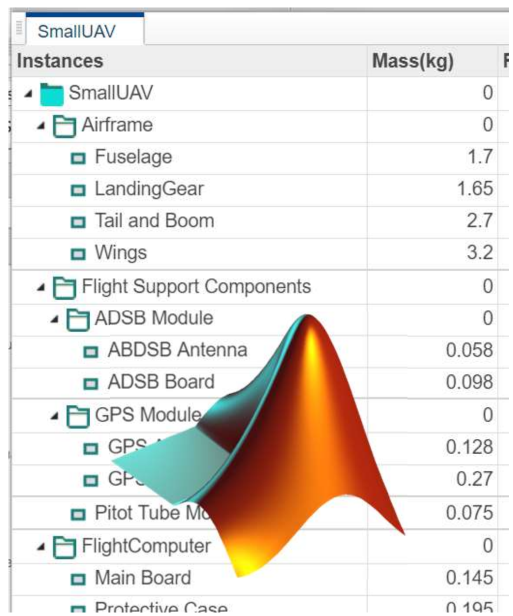
R2019a

Perform trade studies based on data driven analysis to optimize architectures

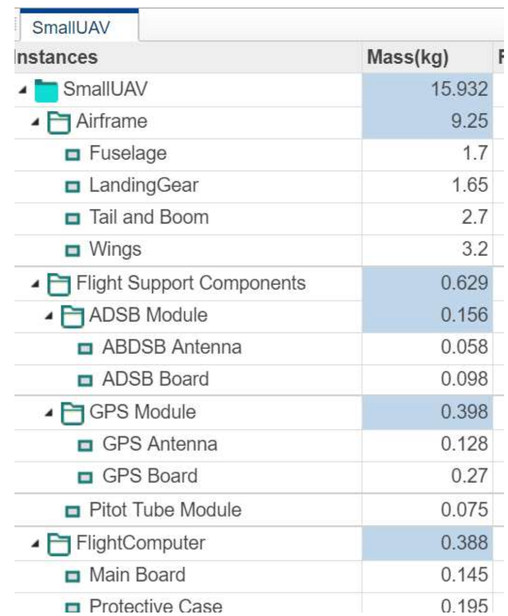
Add custom data

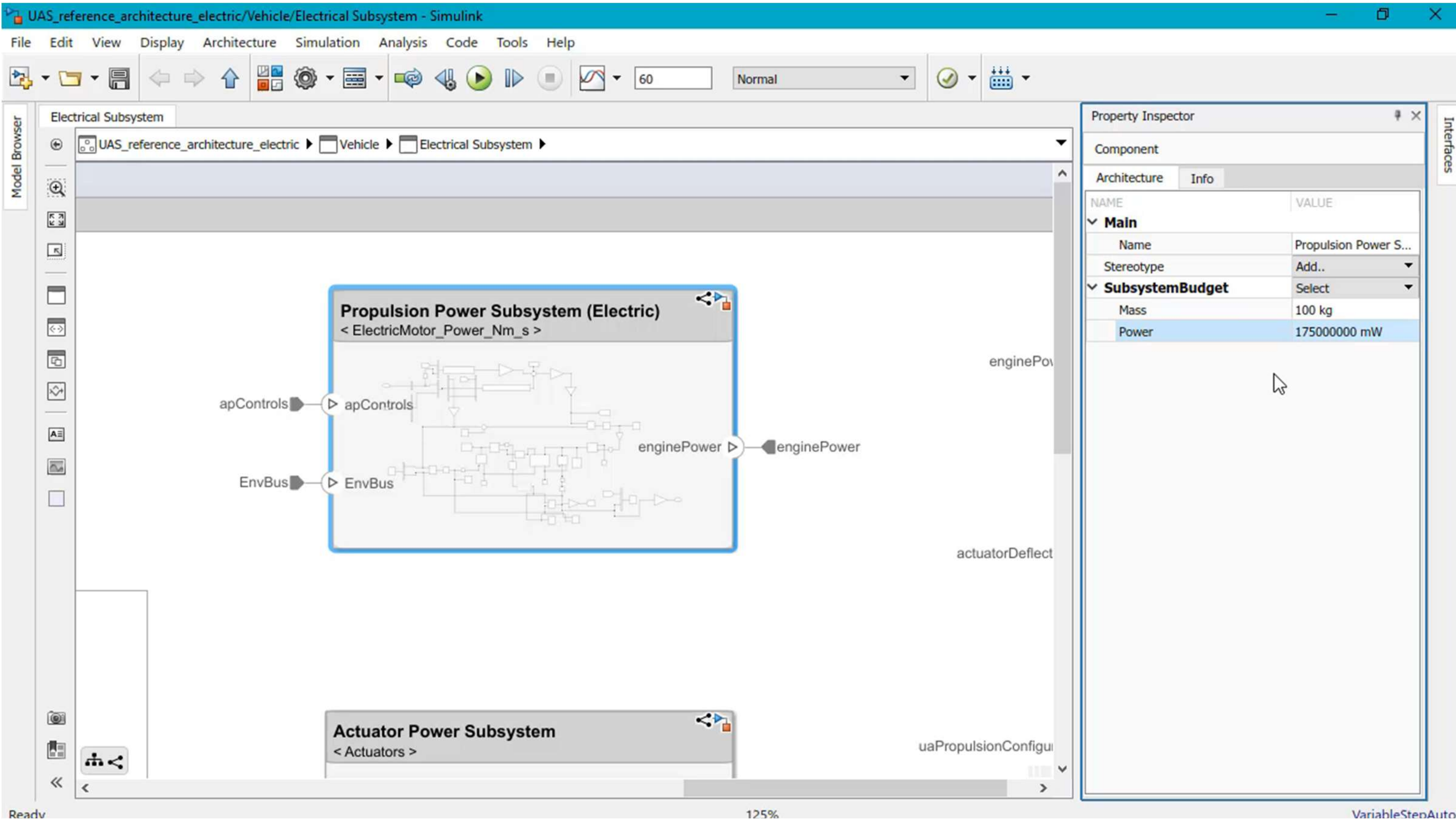


Create analysis model

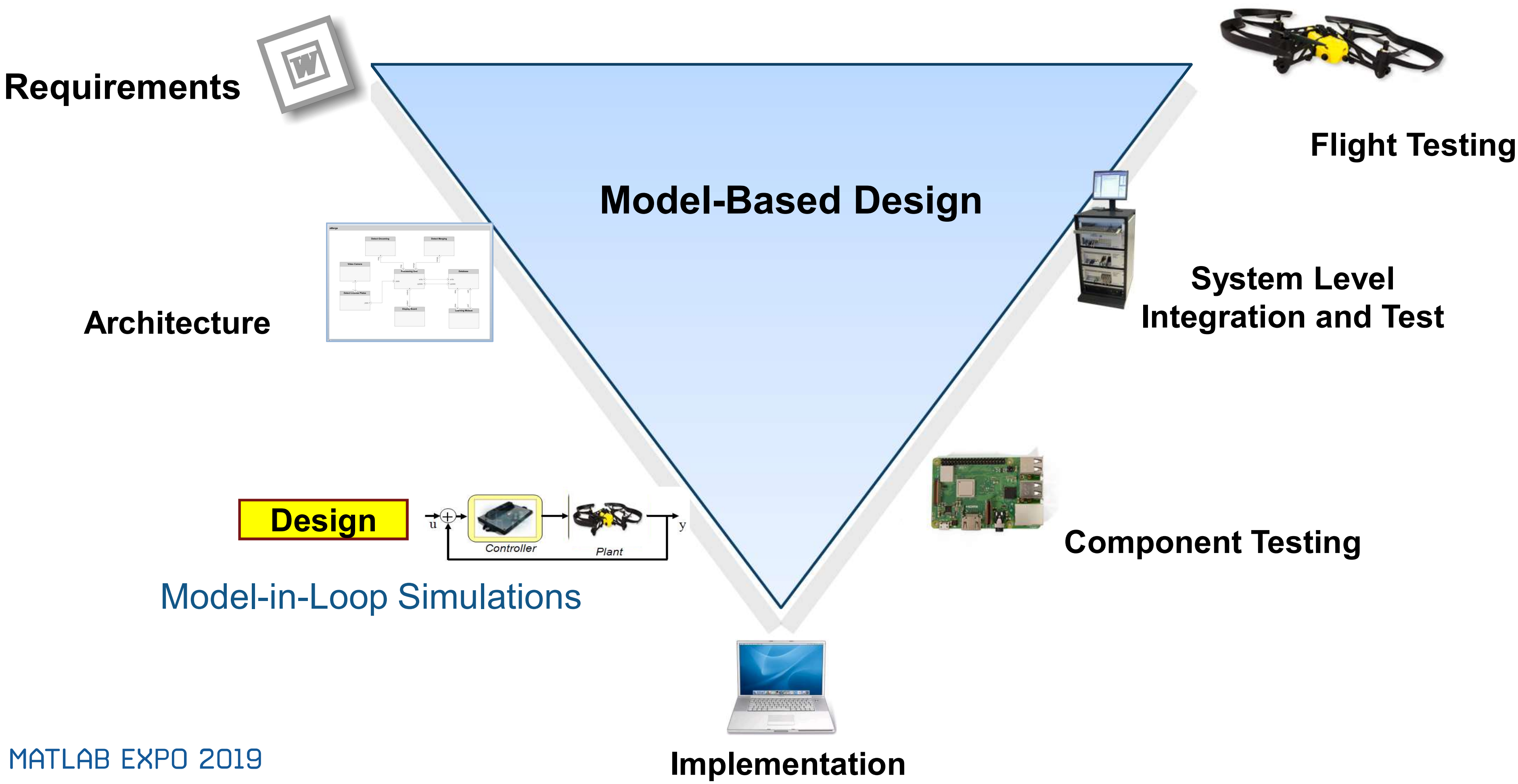


Calculate mass roll-up data




UAS_reference_architecture_electric/Vehicle/Electrical Subsystem - Simulink
File Edit View Display Architecture Simulation Analysis Code Tools Help
60
Normal
Electrical Subsystem
UAS_reference_architecture_electric
Vehicle
Electrical Subsystem
Model Browser
Propulsion Power Subsystem (Electric)
< ElectricMotor_Power_Nm_s >
apControls
apControls
EnvBus
EnvBus
enginePower
enginePower
enginePow
actuatorDeflect
Actuator Power Subsystem
< Actuators >
uaPropulsionConfigu
Property Inspector
Component
Architecture
Info
NAME
VALUE
Main
Name
Propulsion Power S...
Stereotype
Add..
SubsystemBudget
Select
Mass
100 kg
Power
175000000 mW
Interfaces
Ready
125%
VariableStepAuto

V Development Cycle



Design using Simulink

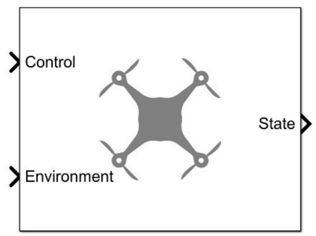
Representing a UAV with a kinematic model

↓

Controlling and tuning the model using a waypoint following algorithm

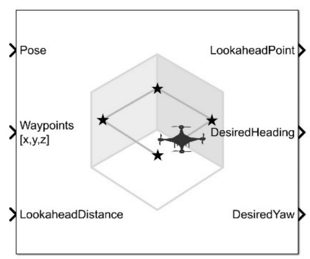
↓

Visualizing the flight behavior and analyzing the flight data



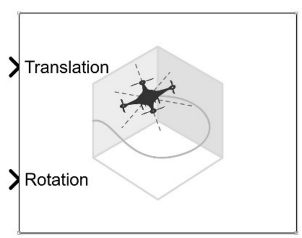
Representing a UAV with a kinematic model

UAV Guidance Model Block



Controlling and tuning the model using a waypoint following algorithm

Waypoint Follower Block



Visualizing the flight behavior and analyzing the flight data

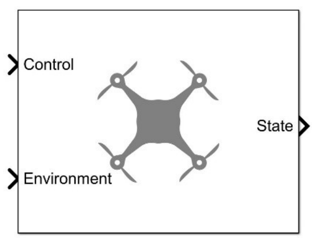
UAV Animation Block

Two approaches to Modeling

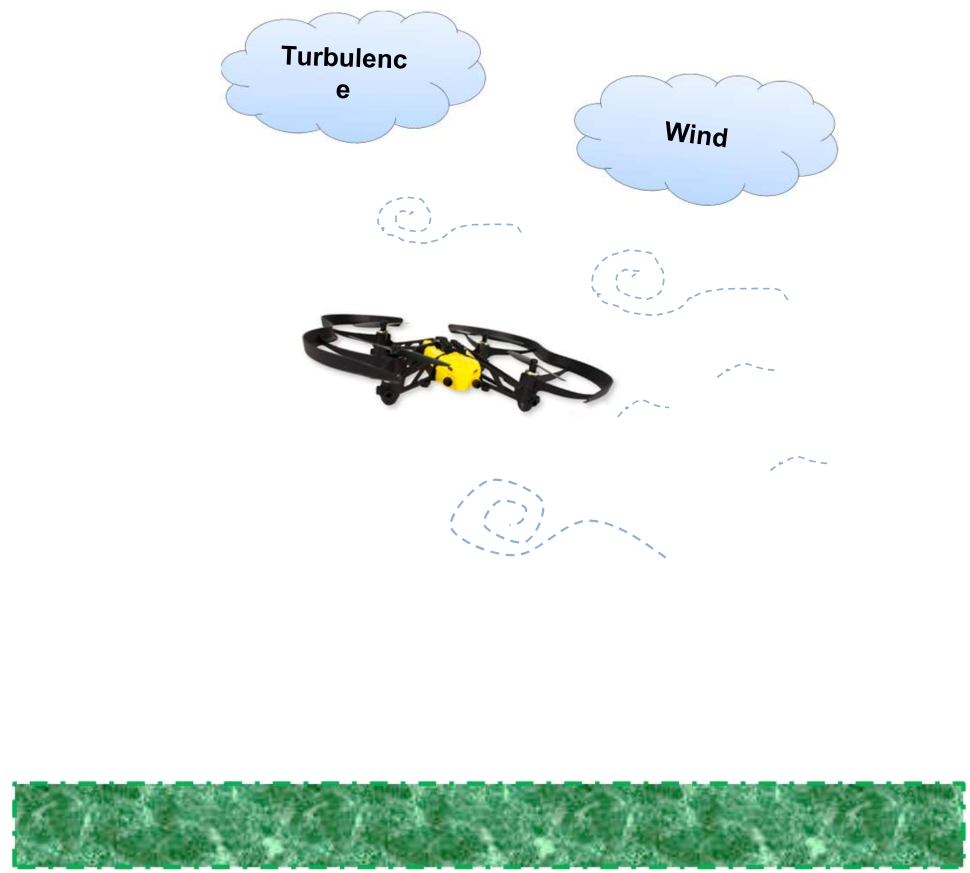
Low Fidelity Simulation ⟷ **High Fidelity Simulation**

Low Fidelity Simulation

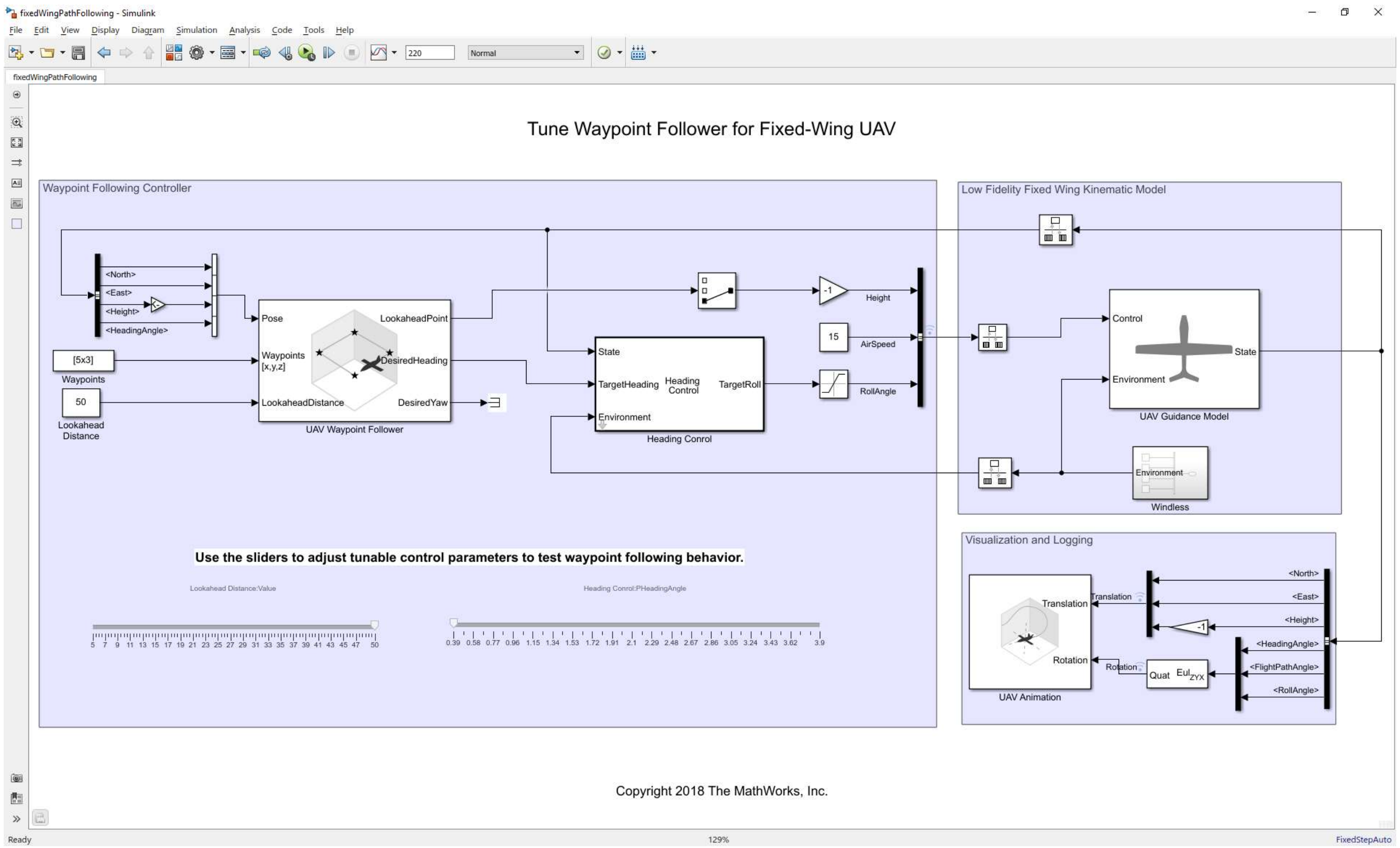
UAV Guidance Model Block

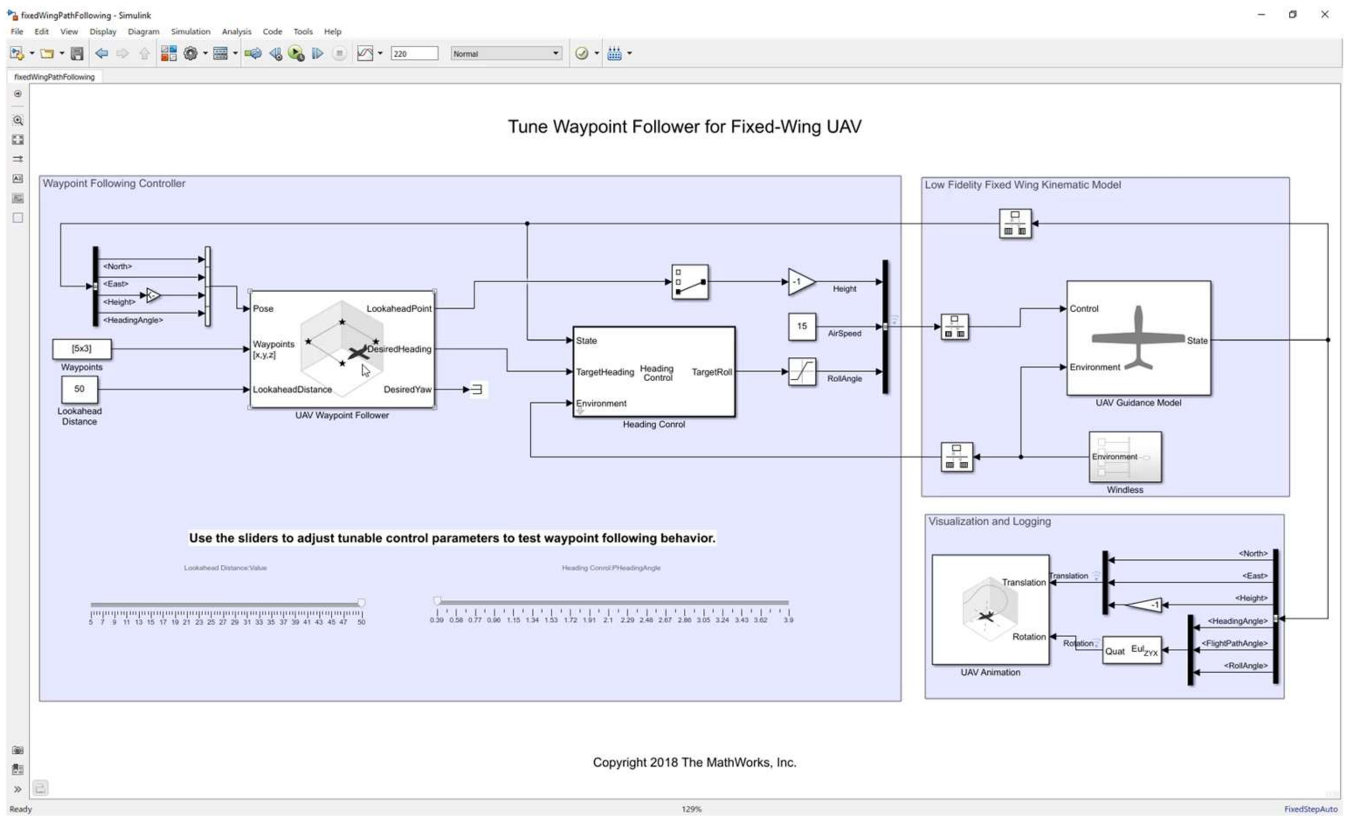


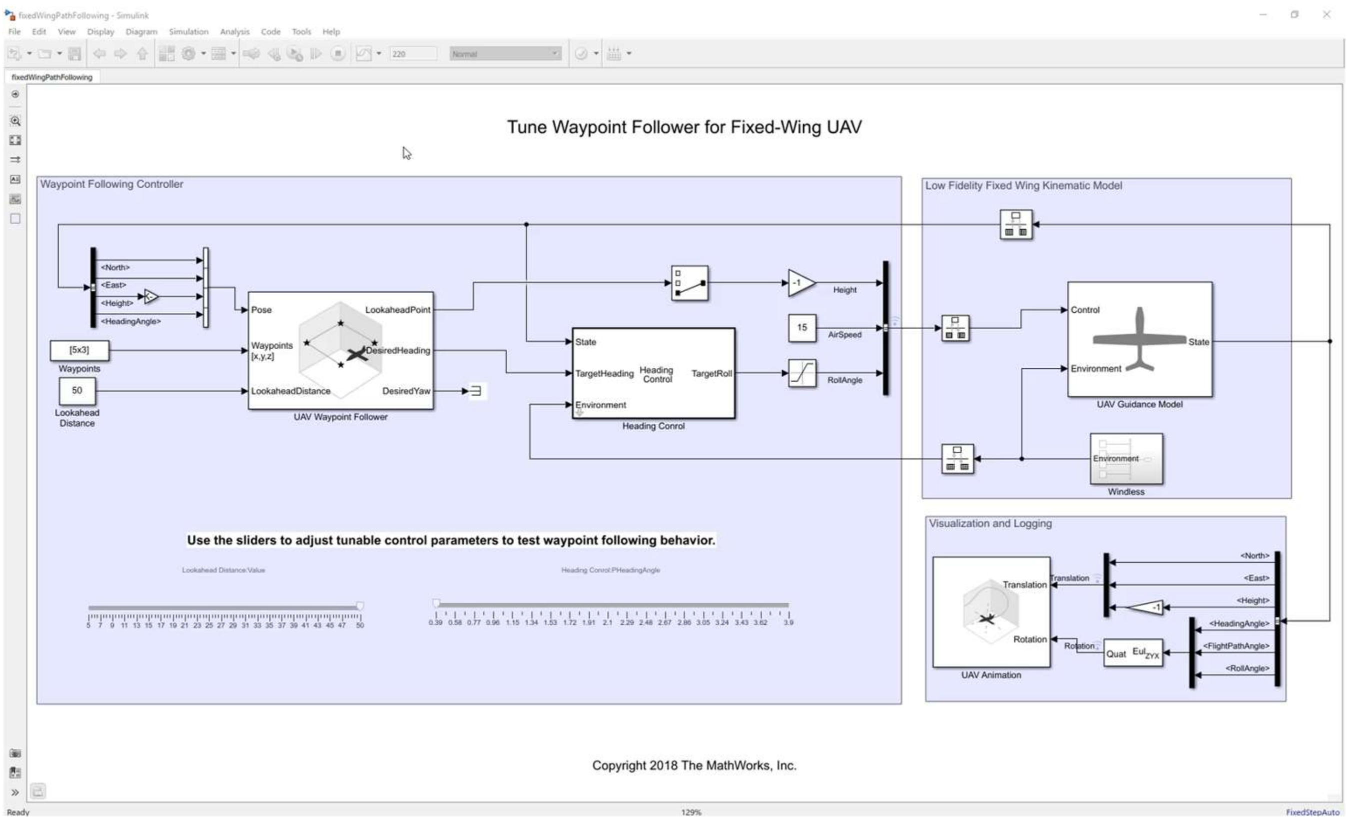
Quick tuning and simulation results



Low Fidelity Simulation for Fixed-Wing UAV

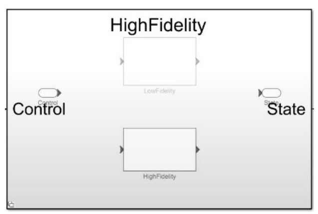



fixedWingPathFollowing - Simulink
Tune Waypoint Follower for Fixed-Wing UAV
Waypoint Following Controller
Low Fidelity Fixed Wing Kinematic Model
UAV Waypoint Follower
Heading Control
UAV Guidance Model
Windless
Visualization and Logging
UAV Animation
Use the sliders to adjust tunable control parameters to test waypoint following behavior.
Copyright 2018 The MathWorks, Inc.

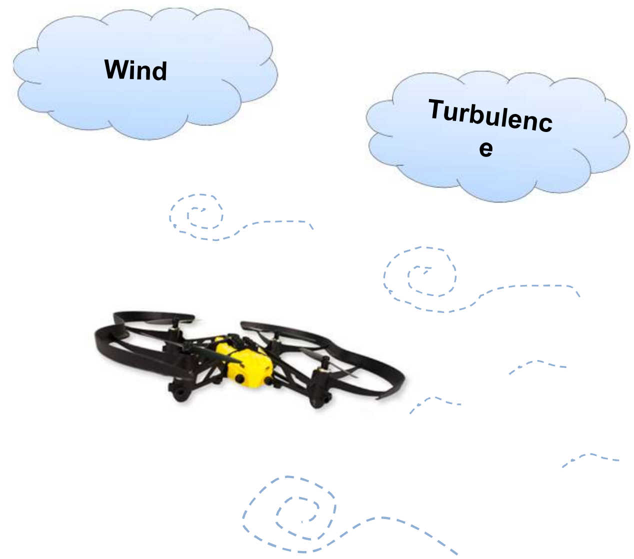

fixedWingPathFollowing - Simulink
Tune Waypoint Follower for Fixed-Wing UAV
Waypoint Following Controller
Low Fidelity Fixed Wing Kinematic Model
Pose
Waypoints [x,y,z]
LookaheadDistance
LookaheadPoint
DesiredHeading
DesiredYaw
UAV Waypoint Follower
[5x3]
Waypoints
50
Lookahead Distance
State
TargetHeading
Environment
Heading Control
TargetRoll
Heading Conrol
Height
AirSpeed
RollAngle
Control
Environment
State
UAV Guidance Model
Windless
Use the sliders to adjust tunable control parameters to test waypoint following behavior.
Visualization and Logging
Translation
Rotation
UAV Animation
Copyright 2018 The MathWorks, Inc.

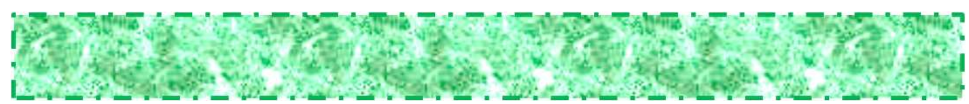
Low Fidelity Simulation

High Fidelity Simulation



Test the parameters with environmental conditions





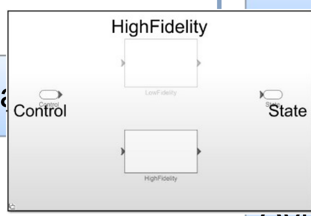
Built with Aerospace Blockset

High Fidelity Simulation

High Fidelity Model

UAV Dynamics model

Pl



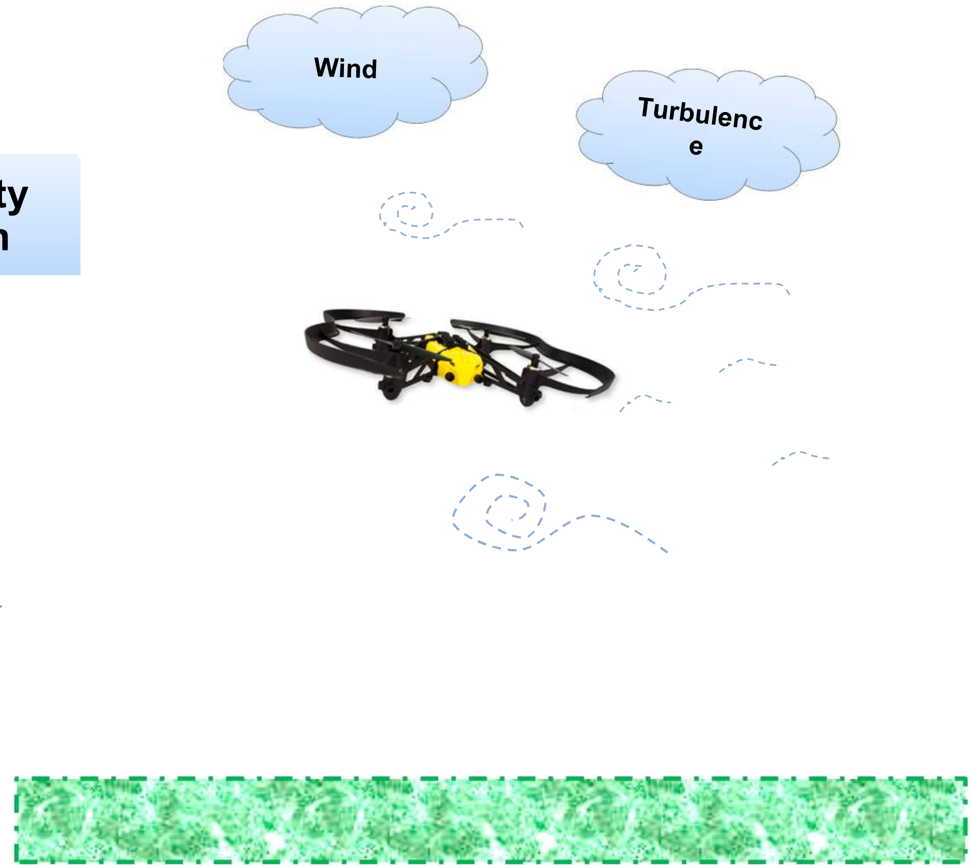
ather Model

ctuator Model

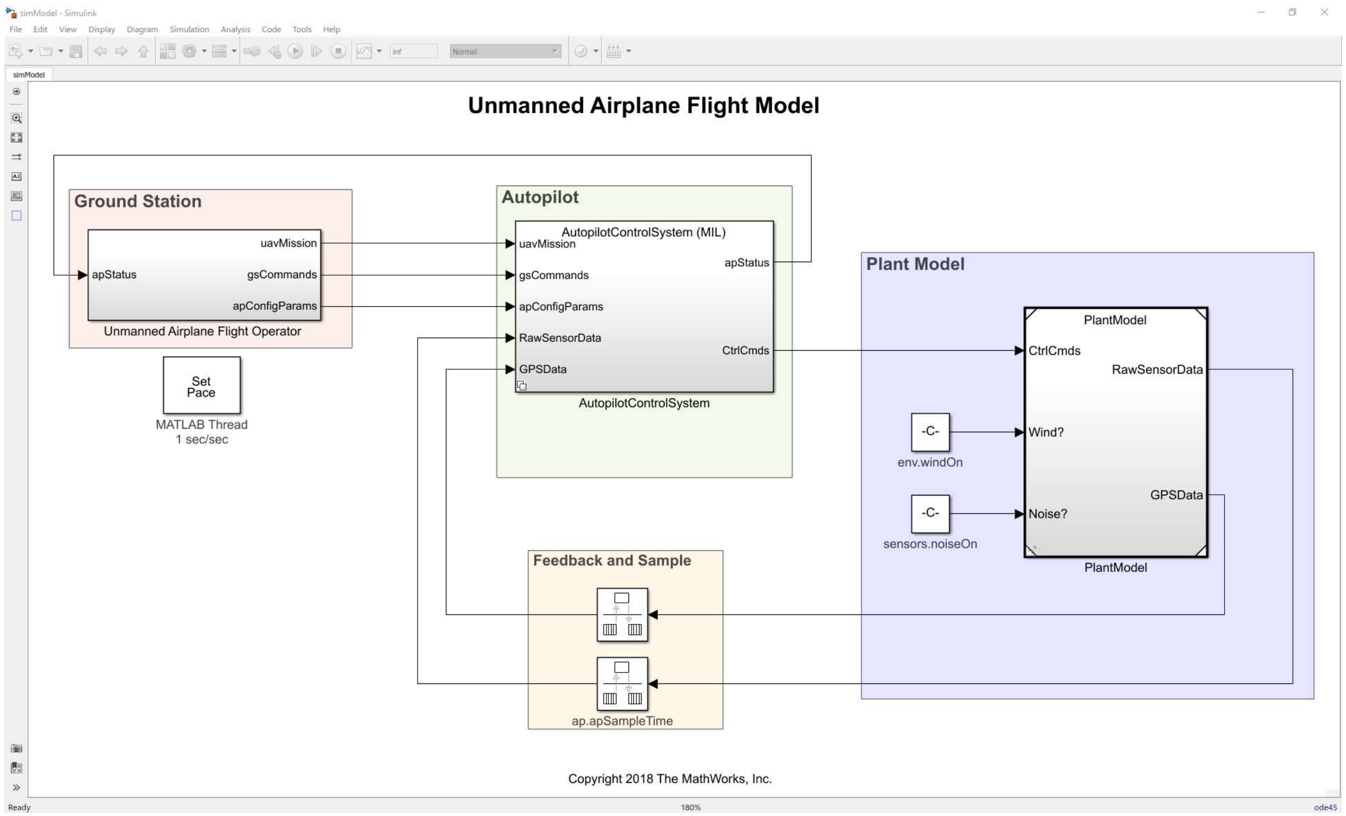
trol

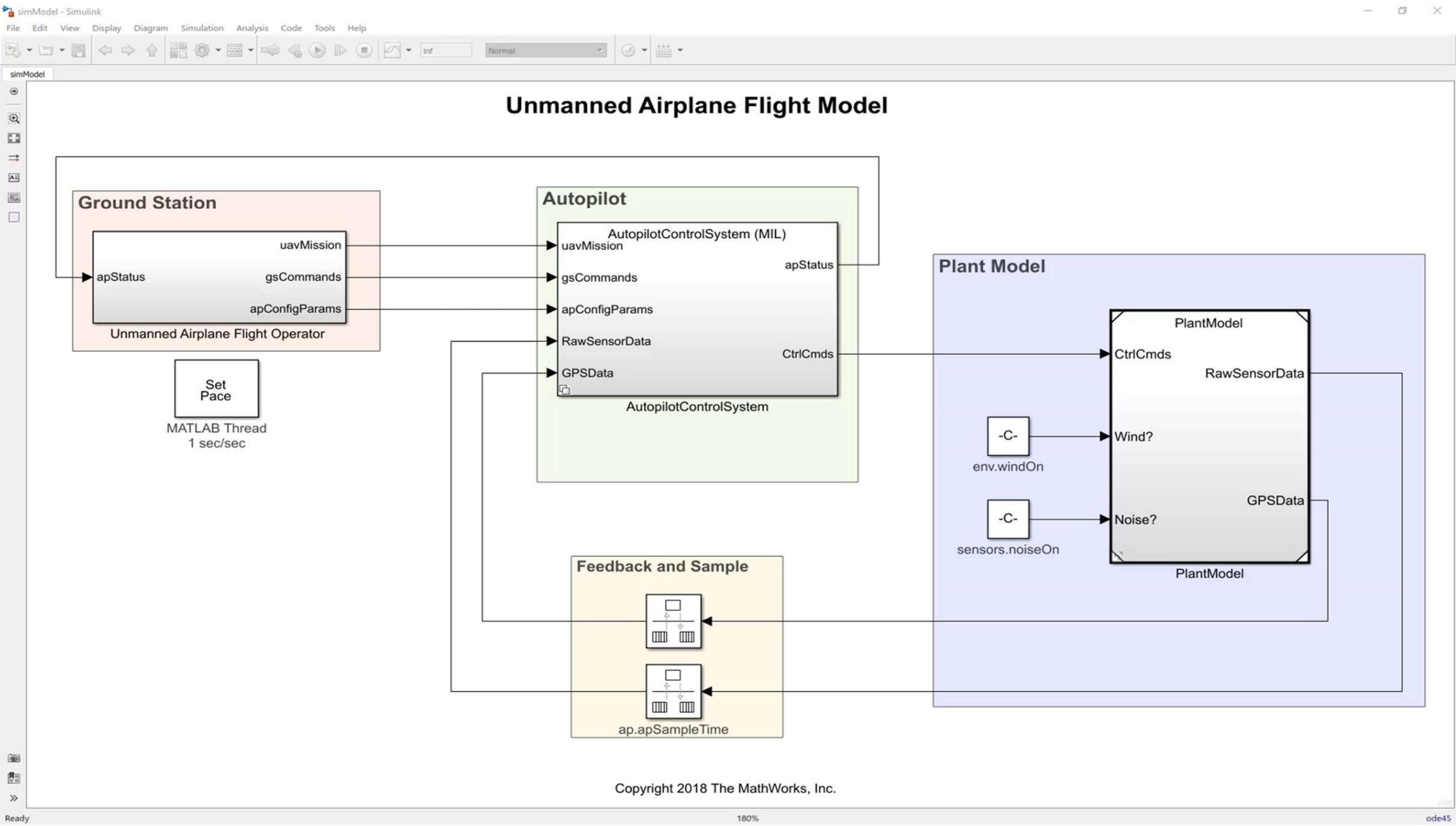
Longitudinal Control

Test the parameters with environmental conditions

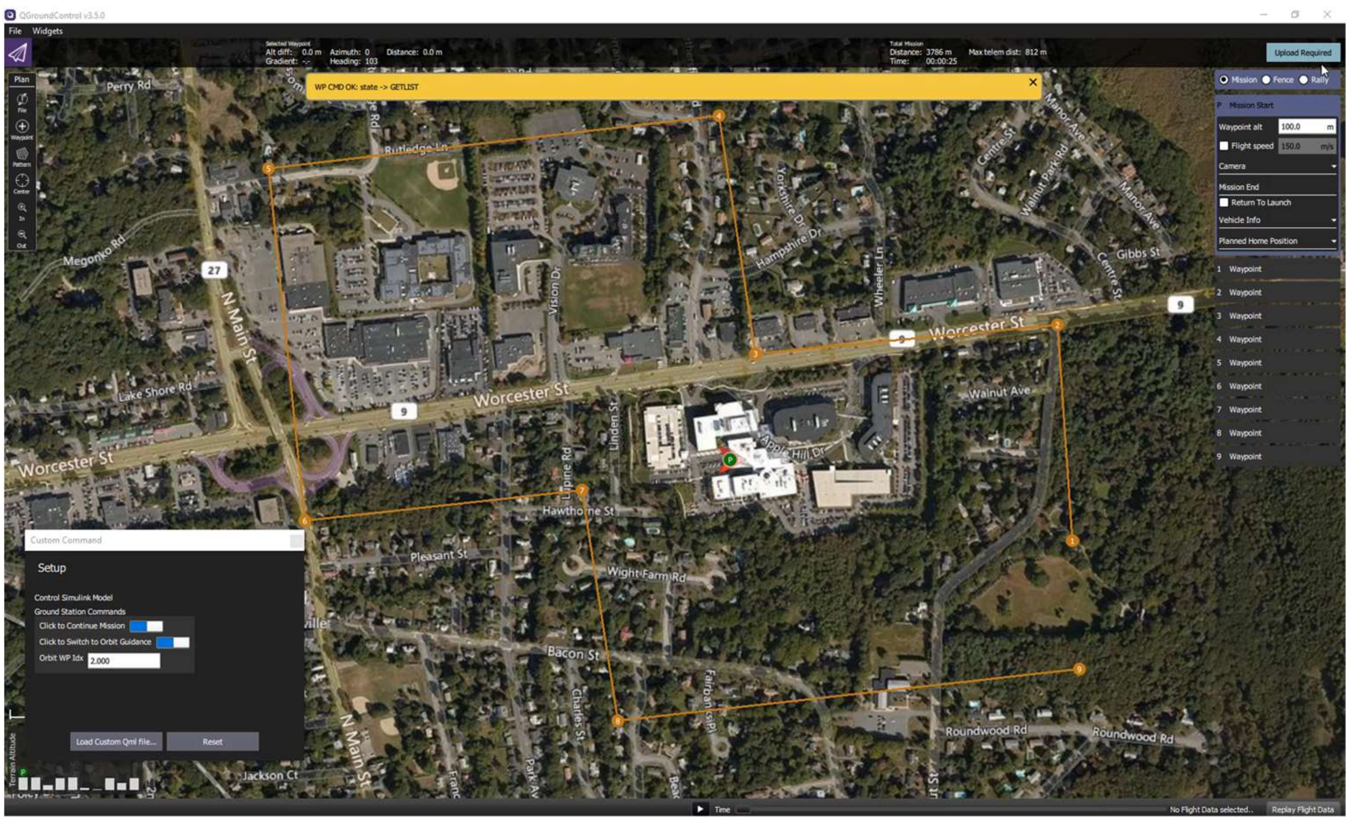


High-Fidelity Simulation Model




simModel - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
simModel
Unmanned Airplane Flight Model
Ground Station
uavMission
apStatus
gsCommands
apConfigParams
Unmanned Airplane Flight Operator
Set Pace
MATLAB Thread
1 sec/sec
Autopilot
AutopilotControlSystem (MIL)
uavMission
gsCommands
apConfigParams
RawSensorData
GPSData
apStatus
CtrlCmds
AutopilotControlSystem
Plant Model
PlantModel
CtrlCmds
RawSensorData
Wind?
GPSData
Noise?
-C-
env.windOn
-C-
sensors.noiseOn
PlantModel
Feedback and Sample
ap.apSampleTime
Copyright 2018 The MathWorks, Inc.
Ready
180%
ode45

Co-simulation with QGroundControl



Intel Creates Dynamic Simulation Environment for Testing GNC Algorithms for Multirotor UAVs

Challenge

Develop and test advanced control algorithms for existing and next-generation multirotor UAVs

Solution

Use MATLAB and Simulink to model multirotor UAV dynamics, verify control algorithms via simulation, and evaluate control design ideas

Results

- Complex calculations verified before flight testing
- Design ideas evaluated in days, not weeks
- Design iterations and testing time reduced

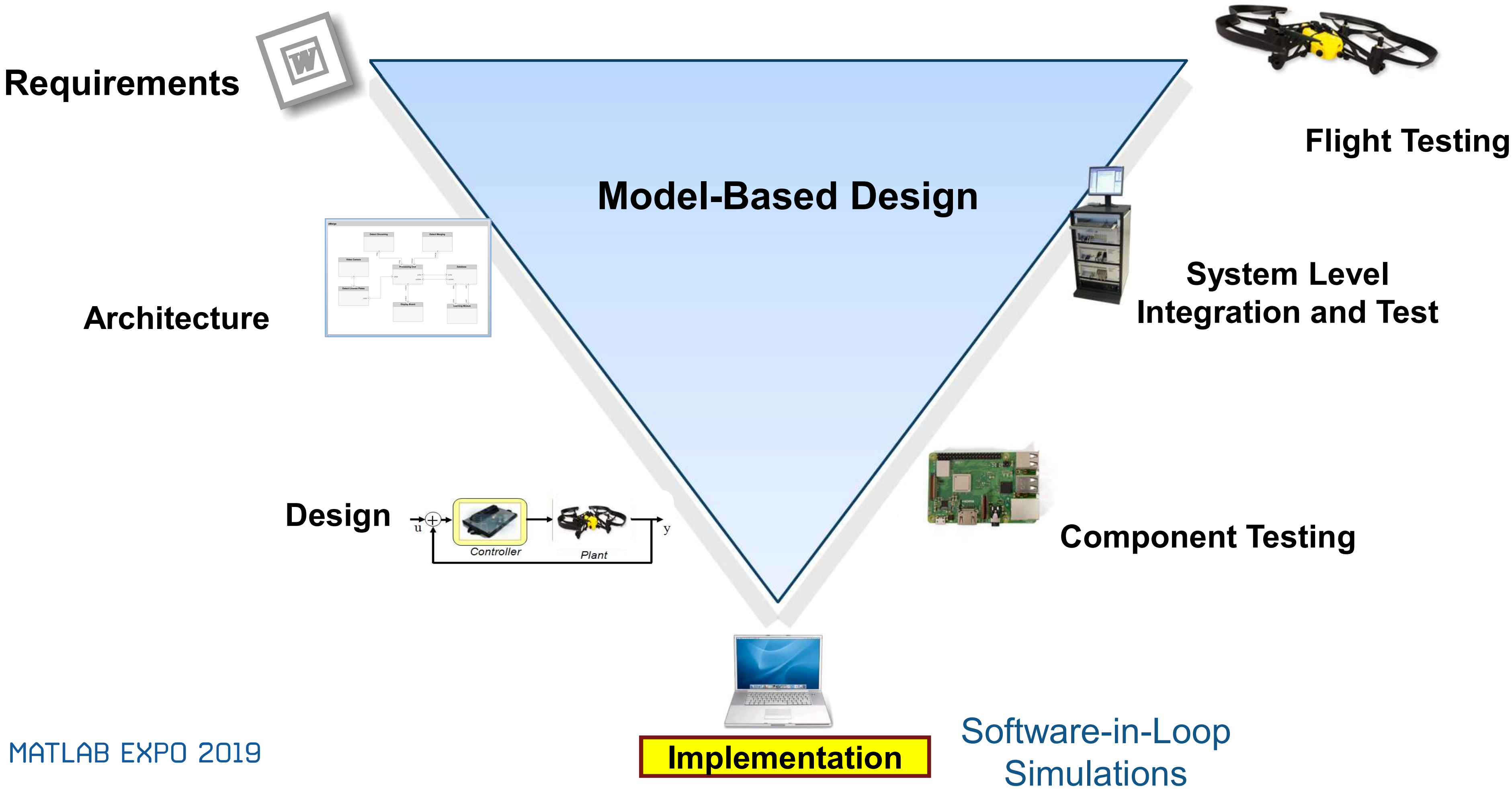
The eight-rotor Intel® Falcon™ 8+.

"Modeling and simulation with Simulink is the only way that we can get the results we need with the speed and quality that's expected in our industry today. If we had to do everything by hand and rely solely on flight testing, we would require more bug fixing iterations and need more testing time per iteration. The problem would grow intractable. There's no other way."

- Jan Vervoorst, Intel

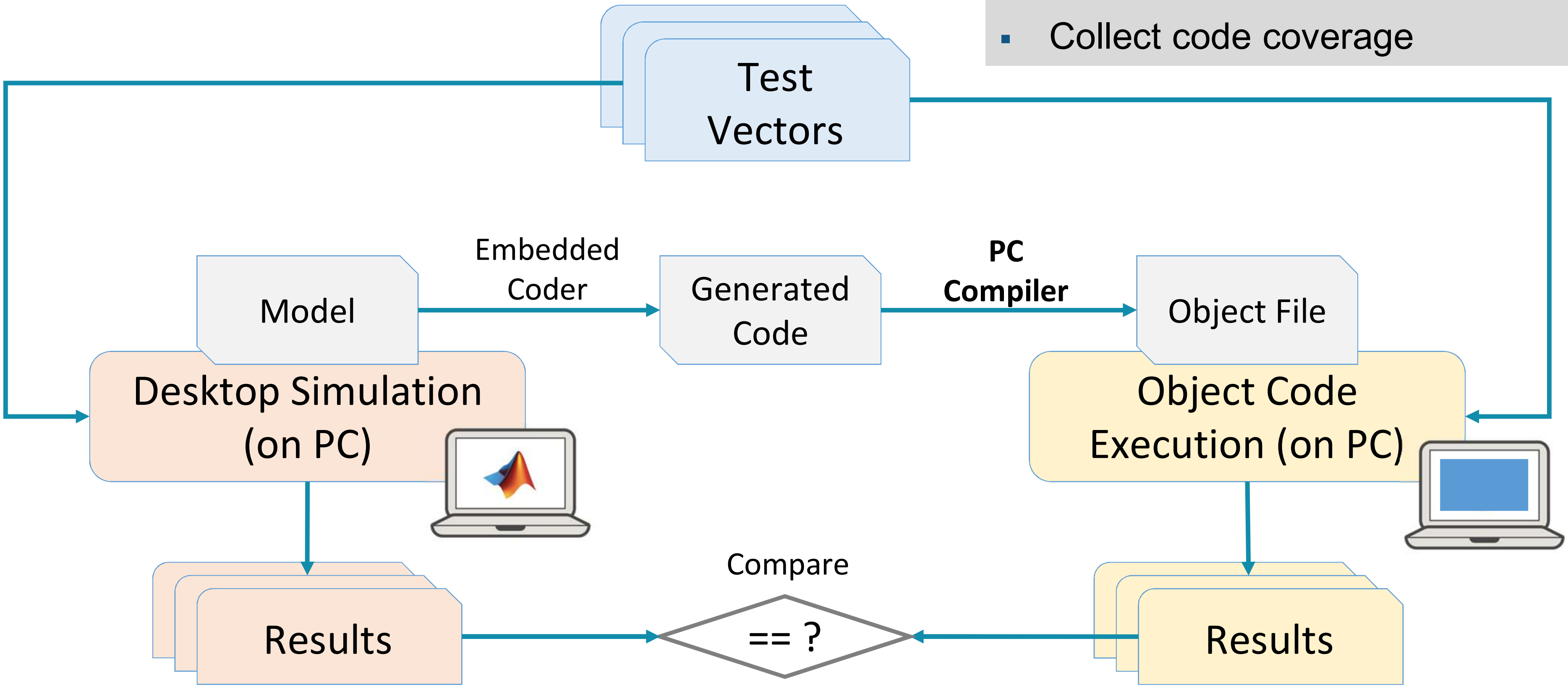
Link to user story

V Development Cycle

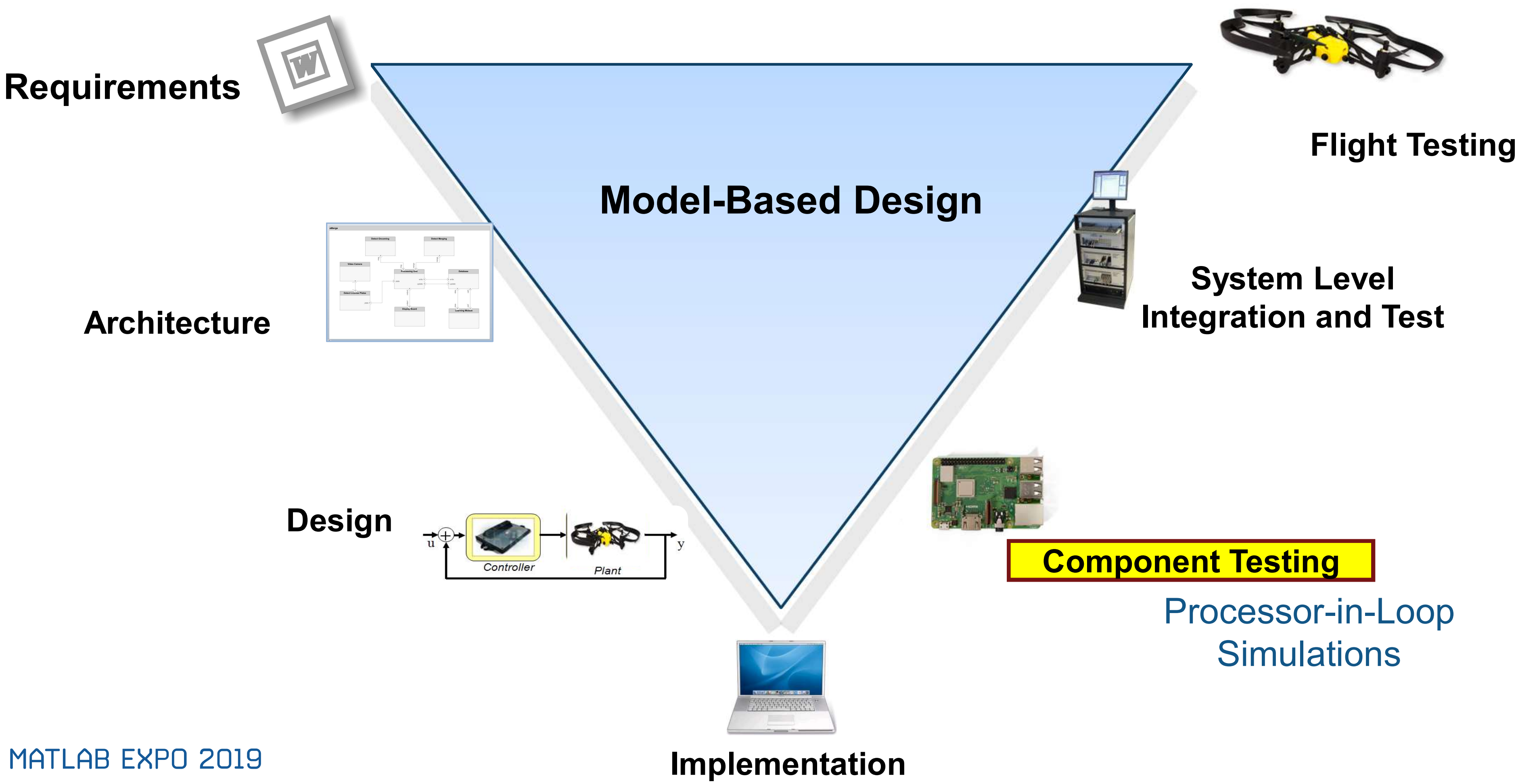


Software In the Loop (SIL) Testing

- Show equivalence, model to code
- Assess code execution time
- Collect code coverage

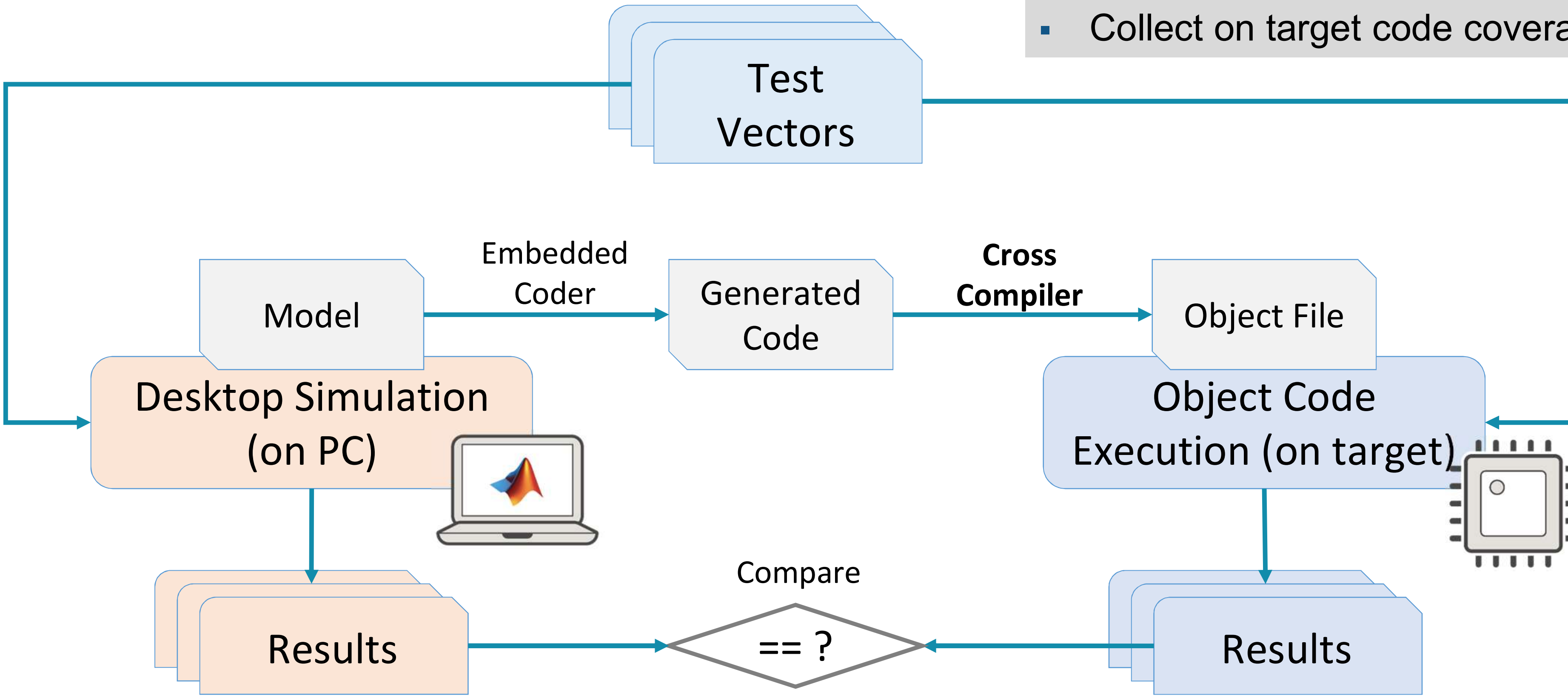


V Development Cycle



Processor In the Loop (PIL) Testing

- Verify numerical equivalence
- Assess target execution time
- Collect on target code coverage



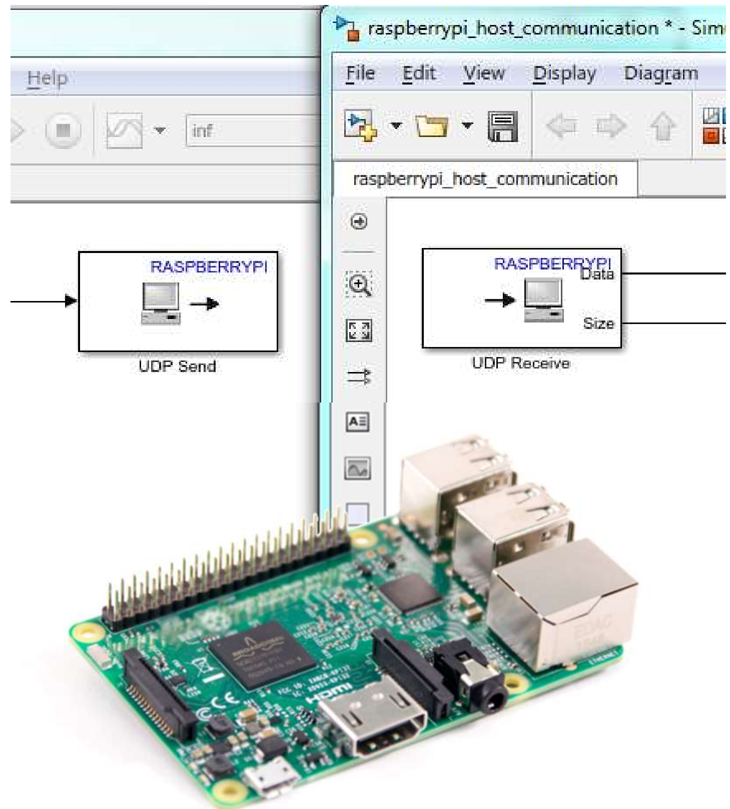
Connection to Hardware
Raspberry Pi 3 Support

Run Simulink models on Raspberry Pi 3 hardware

- Raspberry Pi 3 Model B offers 10 times the performance of the first Raspberry Pi
- Simulink support is now available for Windows, Mac OSX and Linux
- UDP Send and UDP Receive blocks let you communicate over Raspberry Pi 3's integrated 802.11n wireless LAN
- Simulink support now makes use of the Jessi version of the Raspbian OS

» **`raspberrypi_communication`**

» **`raspberrypi_host_communication`**



Embedded Coder Support Package for PX4 Autopilots

Generate, build, and deploy Simulink models on Pixhawk flight controllers

- Integration of the Pixhawk CMAKE build system with Embedded Coder
- Sensor/peripheral block library for inertial measurements, GPS, vehicle estimation, PWM output, and ADC and serial Rx/Tx
- View signal values and tune parameters in real time to interactively test behavior of generated code

Pixhawk 4 hardware

Connection to Hardware

Simulink Support Package for Parrot Minidrones

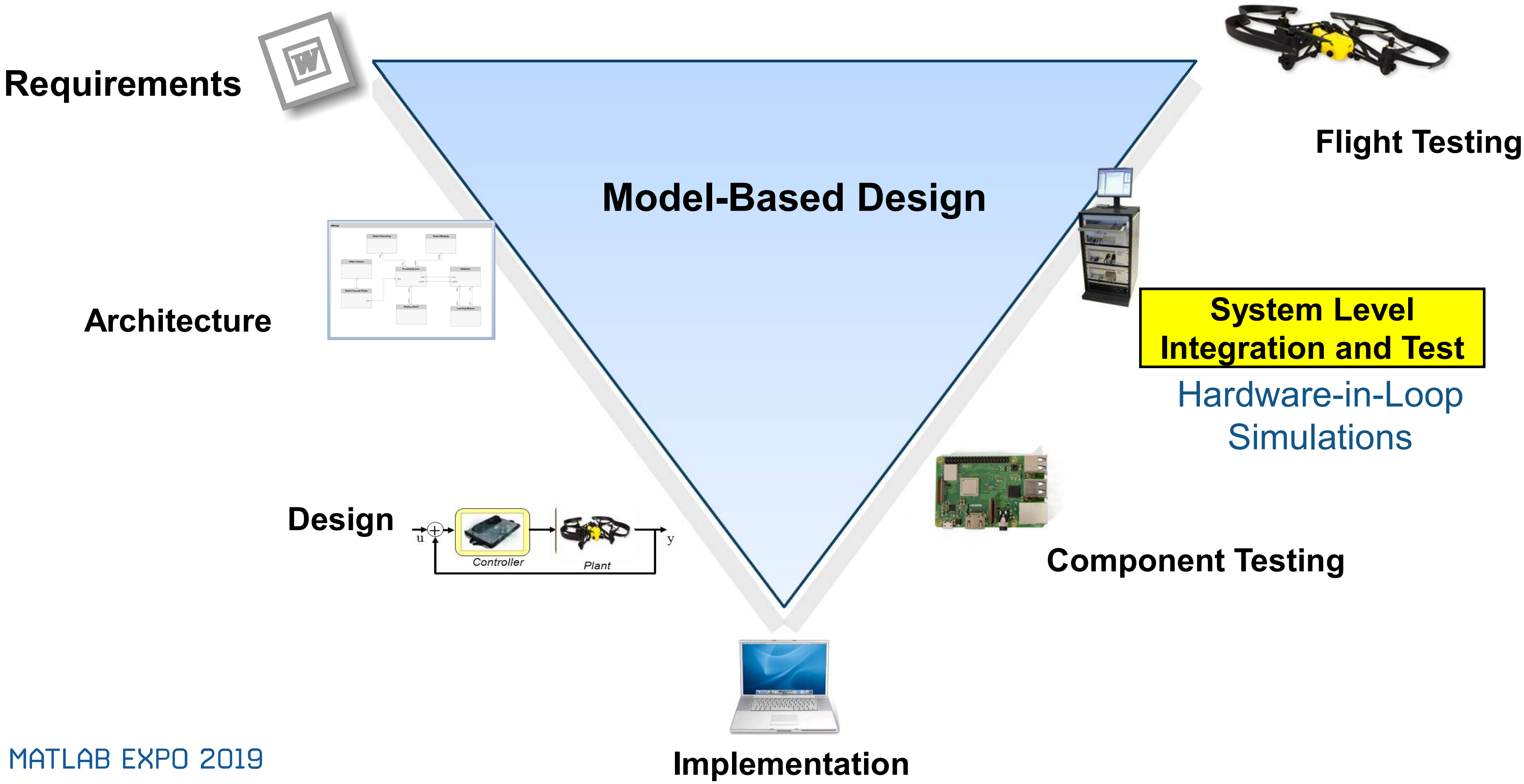
Design, simulate, and deploy algorithms to fly Parrot Minidrones

- Lets you program a low-cost palm-sized reliable minidrone that is available worldwide
- Lets you learn more about Model-Based Design workflows from modeling to hardware deployment
- Aerospace Blockset provides a ready-to-fly example that you can use to get started in minutes
- Simulink Coder gives you access to C code generated for the flight controller

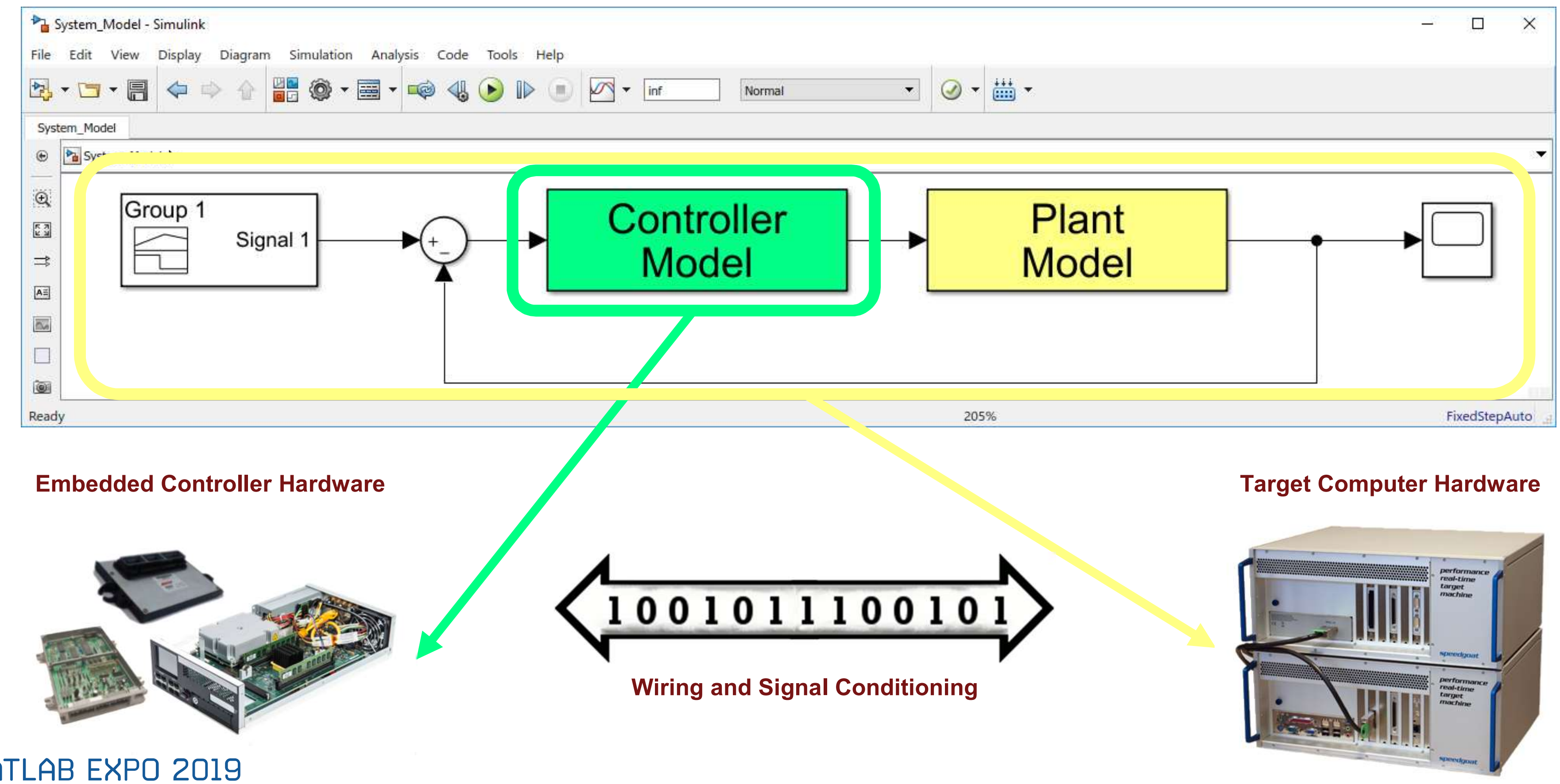
- Introductory video (also on YouTube)
- Hardware Catalog Page: parrot Minidrones Support from Simulink

R2017b

V Development Cycle



Hardware-in-the-loop (HIL) Simulation

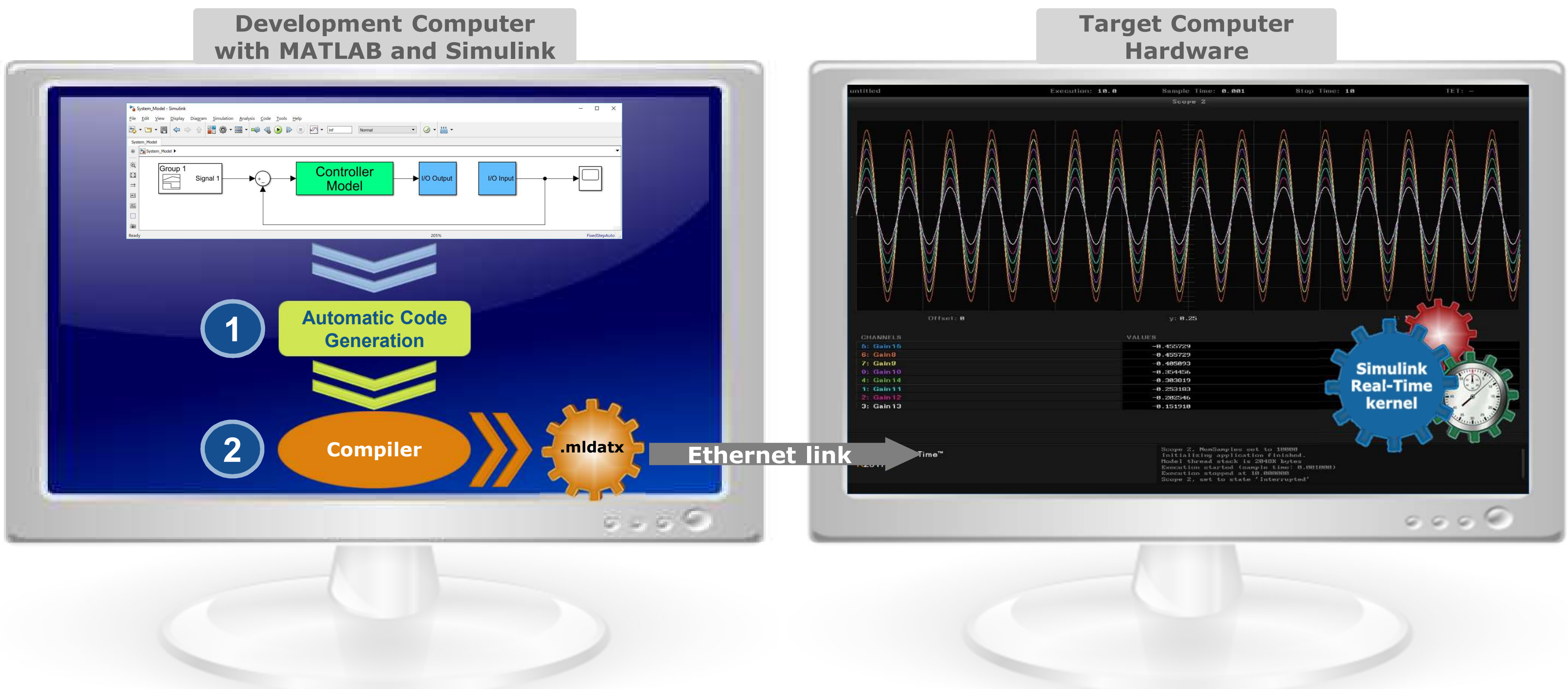


Why HIL Simulations?

- Enables repeatable testing
- Avoid mechanical wear and tear and test boundary/extreme conditions
- Simulate faults and test fault tolerance of control system
- Can also ease path of certification by conducting exhaustive flight tests

Simulink Real-Time

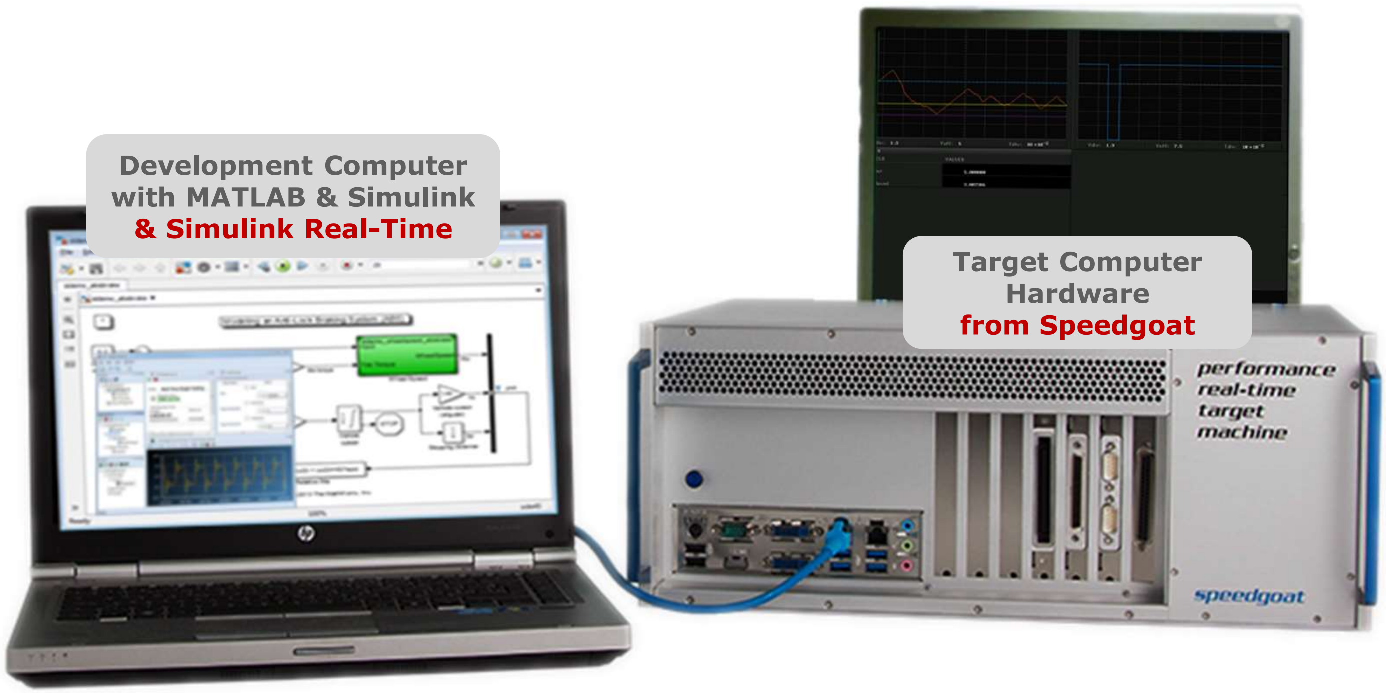
From Desktop Simulation to Real-Time Execution



1 Code Generation 2 Compile and Link 3 Download and Ready to Run

Simulink Real-Time with Speedgoat

Real-time software environment + real-time target computer



Real-Time Simulation and Testing

Complete Solution

MathWorks Simulink Real-Time

MathWorks instrumentation

Toolboxes

Simscape/SimMechanics/Sim PowerSystems

HDL Coder

MathWorks Kernel

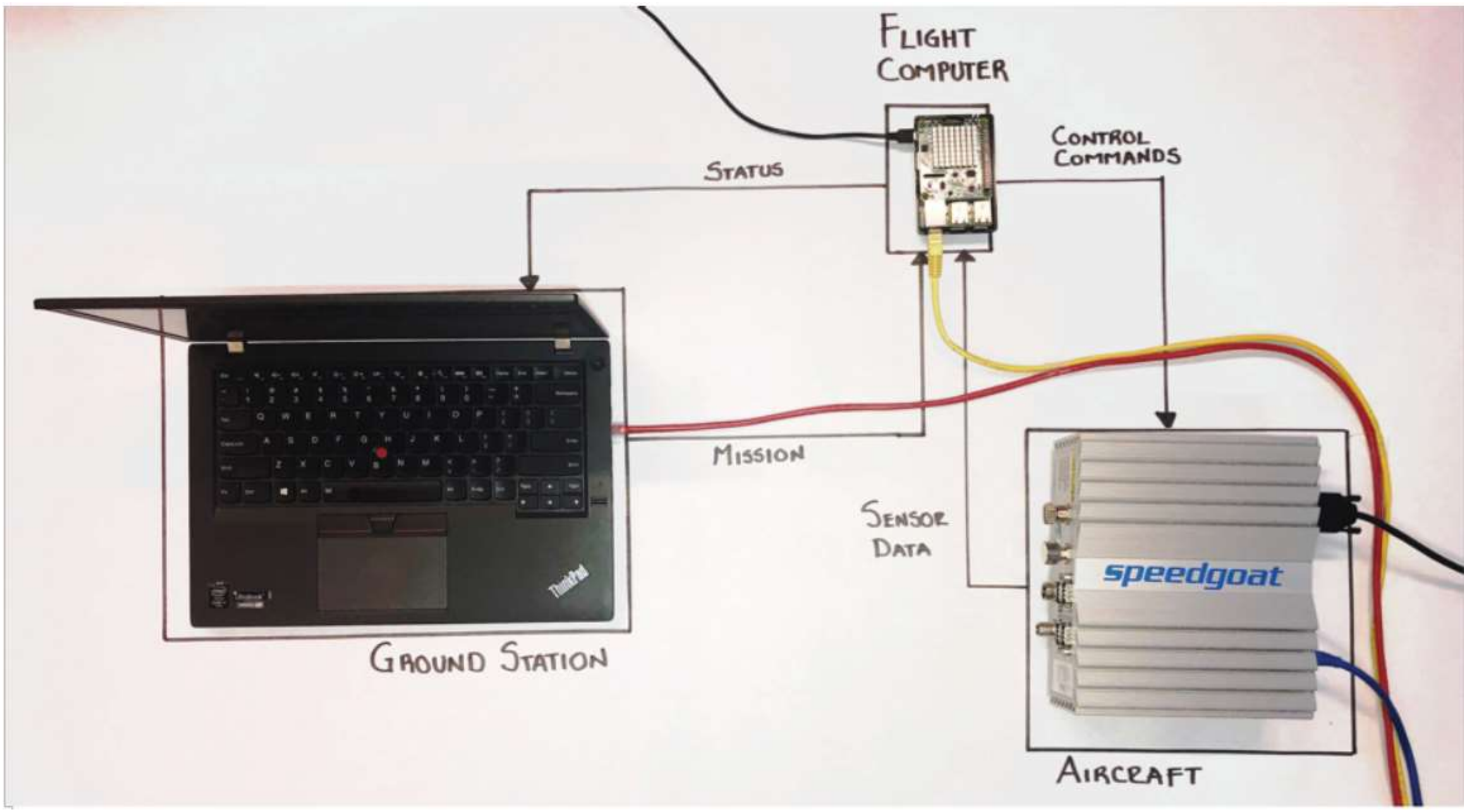
Speedgoat real-time target machines

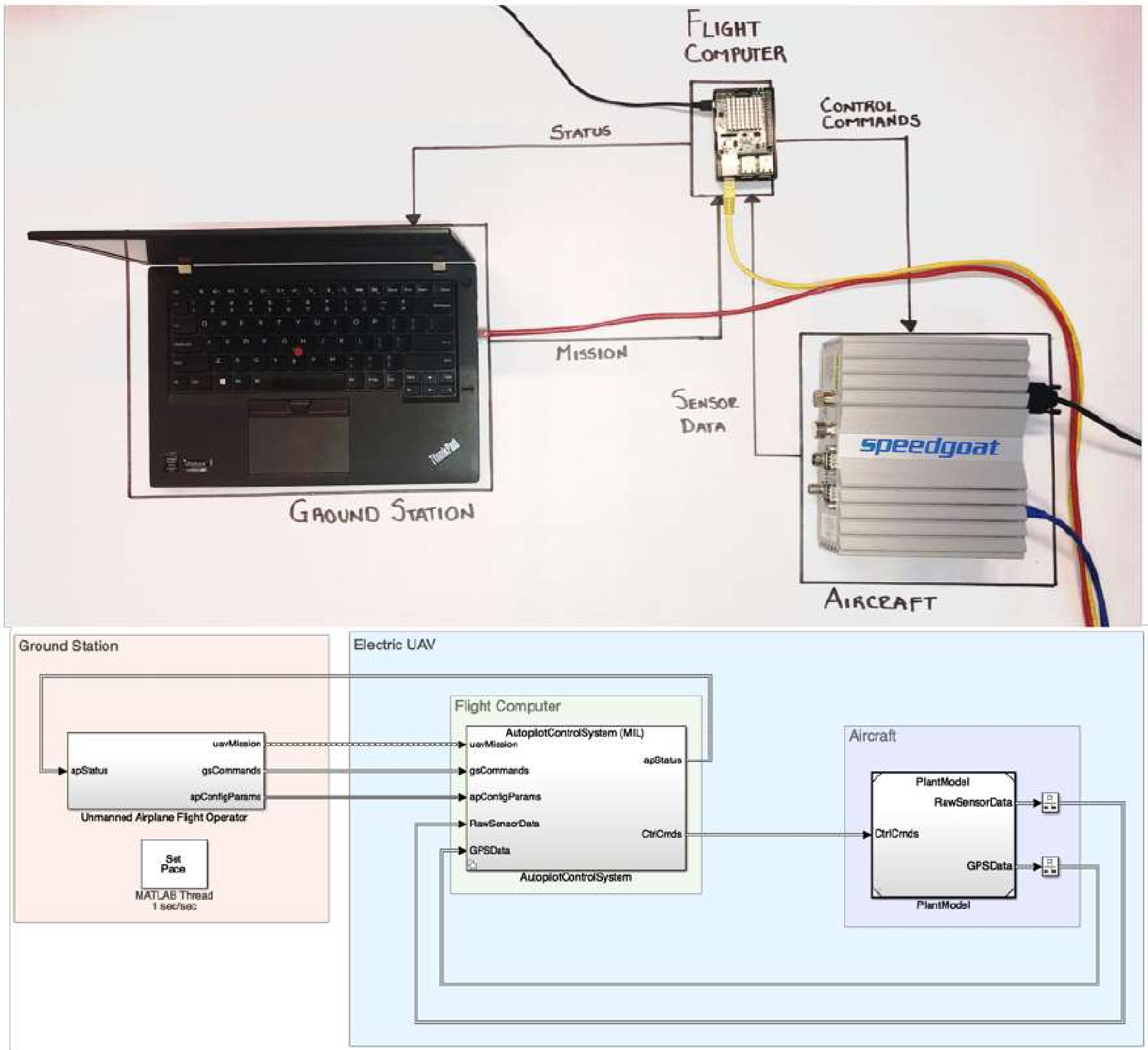
Speedgoat I/O Modules and protocol support

Speedgoat driver library

FPGA-based solutions

HIL Simulations - setup





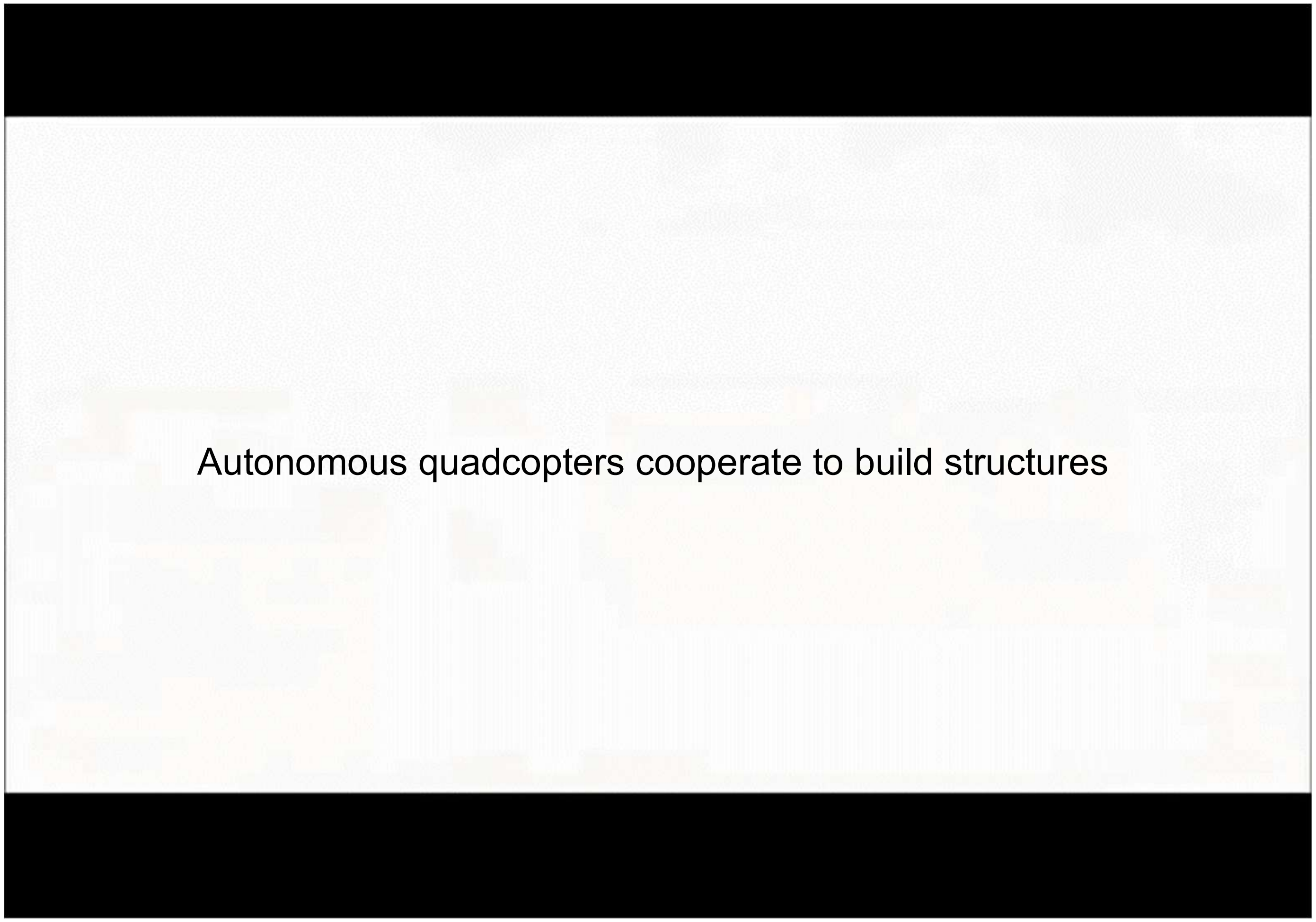
FLIGHT COMPUTER
STATUS
CONTROL COMMANDS
MISSION
SENSOR DATA
GROUND STATION
AIRCRAFT
speedgoat
Ground Station
Electric UAV
Flight Computer
AutopilotControlSystem (MIL)
uavMission
gsCommands
apConfigParams
RawSensorData
GPSData
apStatus
CtrlCmds
AutopilotControlSystem
apStatus
Unmanned Airplane Flight Operator
Set Pace
MATLAB Thread
1 sec/sec
Aircraft
PlantModel
RawSensorData
GPSData
CtrlCmds

HIL Simulations Video

Key takeaways

- Simulink as a platform for System level design of UAV
- Model environmental effects and 6DOF aircraft simulations
- Design autopilot and test its performance under simulated flight conditions
- Deploy and test correctness of Flight Controller’s generated code

Where is this UAV development heading?


Autonomous quadcopters cooperate to build structures

So, What's your mission?

Training Services

Exploit the full potential of MathWorks products

Flexible delivery options:

- Public training available in several cities
- Onsite training with standard or customized courses
- Web-based training with live, interactive instructor-led courses

More than 30 course offerings:

- Introductory and intermediate training on MATLAB, Simulink, Stateflow, code generation, and Polyspace products
- Specialized courses in control design, signal processing, parallel computing, code generation, communications, financial analysis, and other areas

www.mathworks.in/training

Thank you!

MATLAB EXPO 2019

Email: Naga.Pemmaraju@mathworks.in

LinkedIn: https://www.linkedin.com/in/n-pemmaraju/