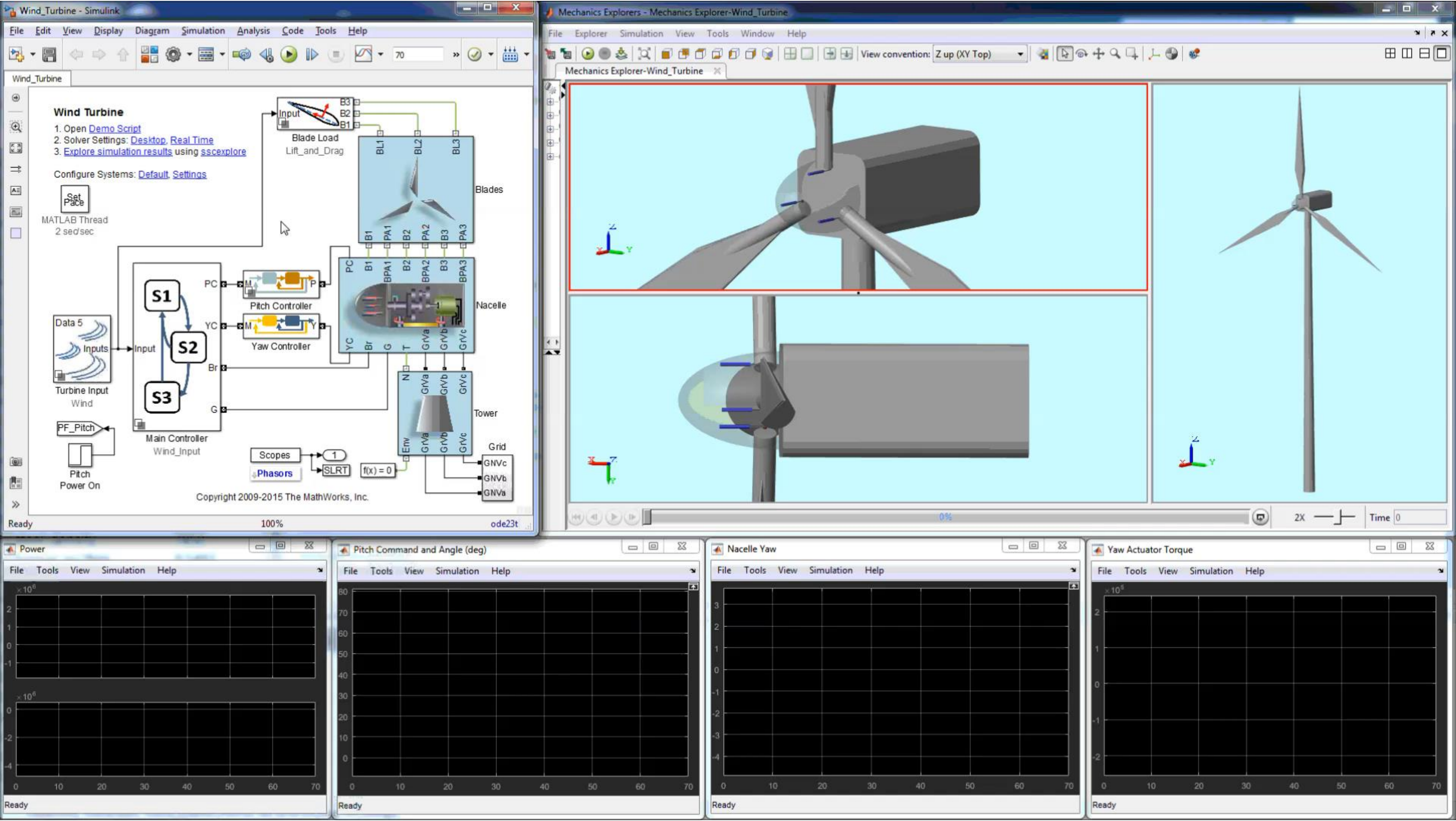
MATLAB EXPO 2017

Introduction to Simulink & Stateflow

Jonathan Agg


Wind_Turbine - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
Wind_Turbine
Wind Turbine
1. Open Demo Script
2. Solver Settings: Desktop, Real Time
3. Explore simulation results using sscexplore
Configure Systems: Default, Settings
Set Pace
MATLAB Thread
2 sec/sec
Blade Load
Lift_and_Drag
Blades
Nacelle
Tower
Grid
Pitch Controller
Yaw Controller
Main Controller
Wind_Input
Turbine Input
Wind
PF_Pitch
Pitch
Power On
Scopes
Phasors
SLRT
f(x) = 0
Copyright 2009-2015 The MathWorks, Inc.
Ready
100%
ode23t
Mechanics Explorers - Mechanics Explorer-Wind_Turbine
File Explorer Simulation View Tools Window Help
View convention: Z up (XY Top)
Mechanics Explorer-Wind_Turbine
Time
Power
Pitch Command and Angle (deg)
Nacelle Yaw
Yaw Actuator Torque

Topics we will address this session

- Why model a system?

- Why use Simulink?

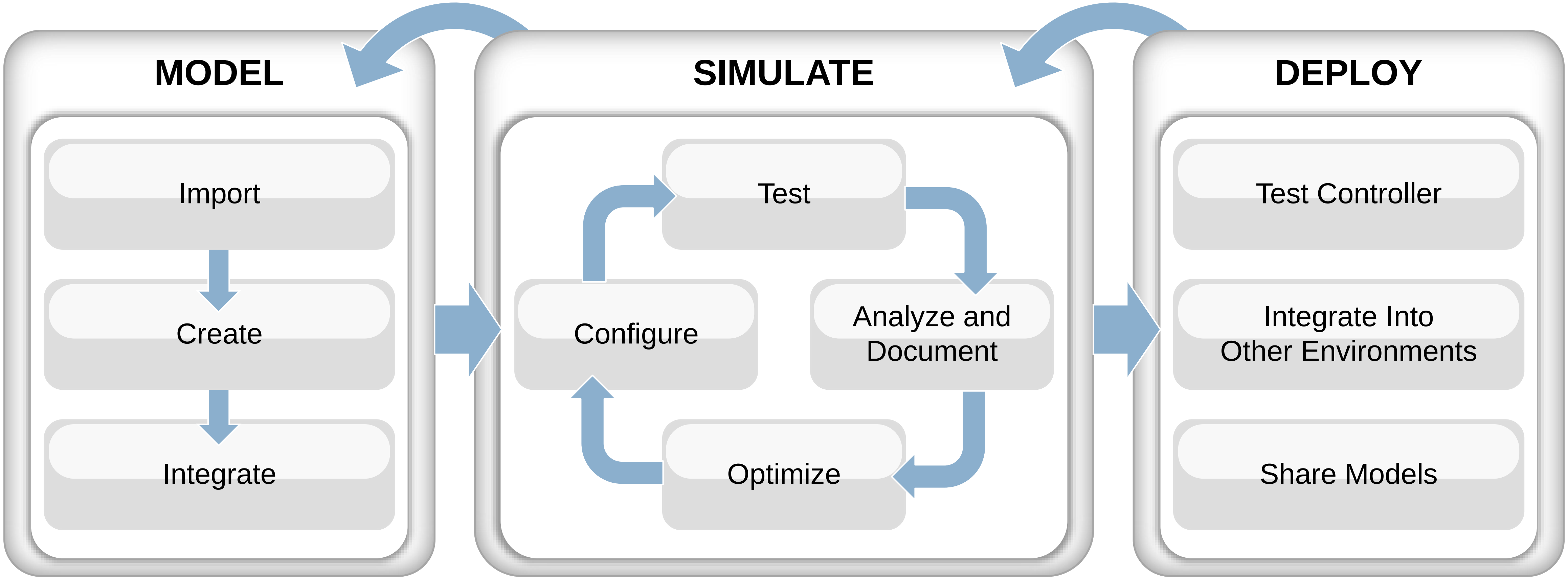
- Getting to grips with the basics of Simulink and Stateflow through a worked example

Why model a system?

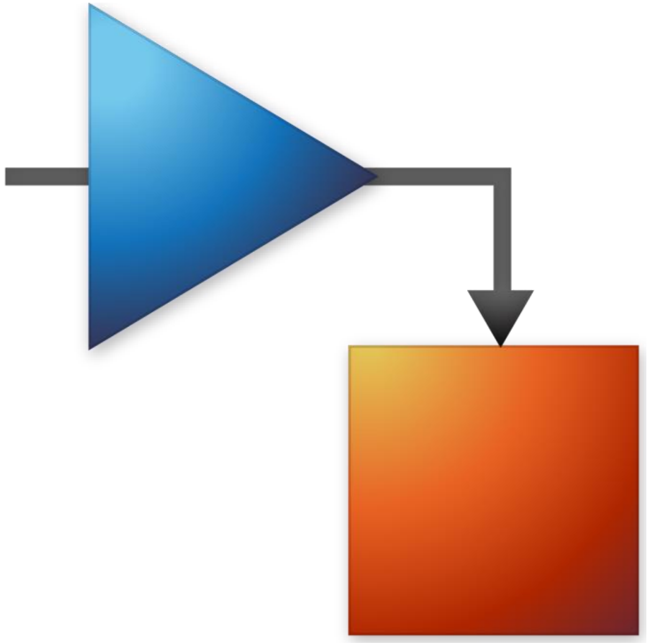
Modelling & Simulation gives you insight

Image credit: McLaren

Image credit: Peter Gronemann | Wikipedia

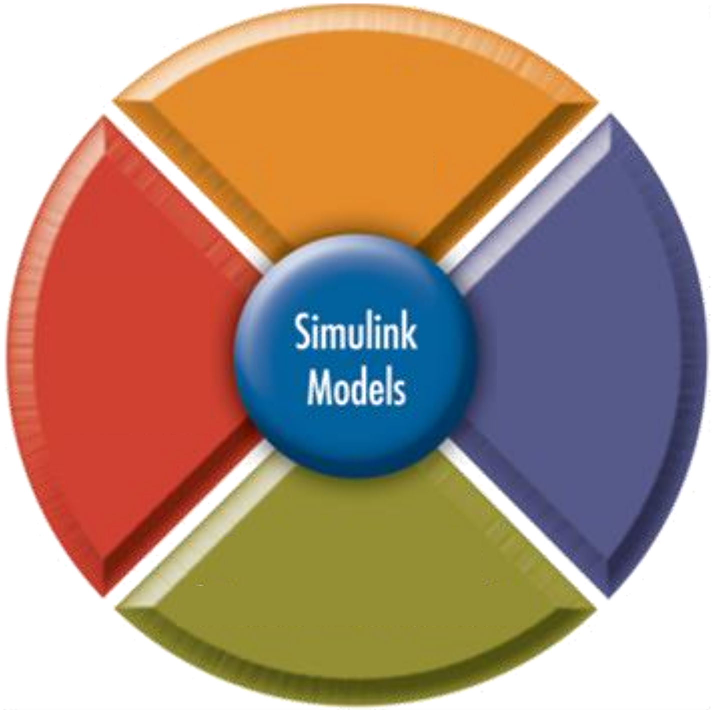

MODEL
Import
Create
Integrate
SIMULATE
Test
Configure
Analyze and Document
Optimize
DEPLOY
Test Controller
Integrate Into Other Environments
Share Models

Why use Simulink?

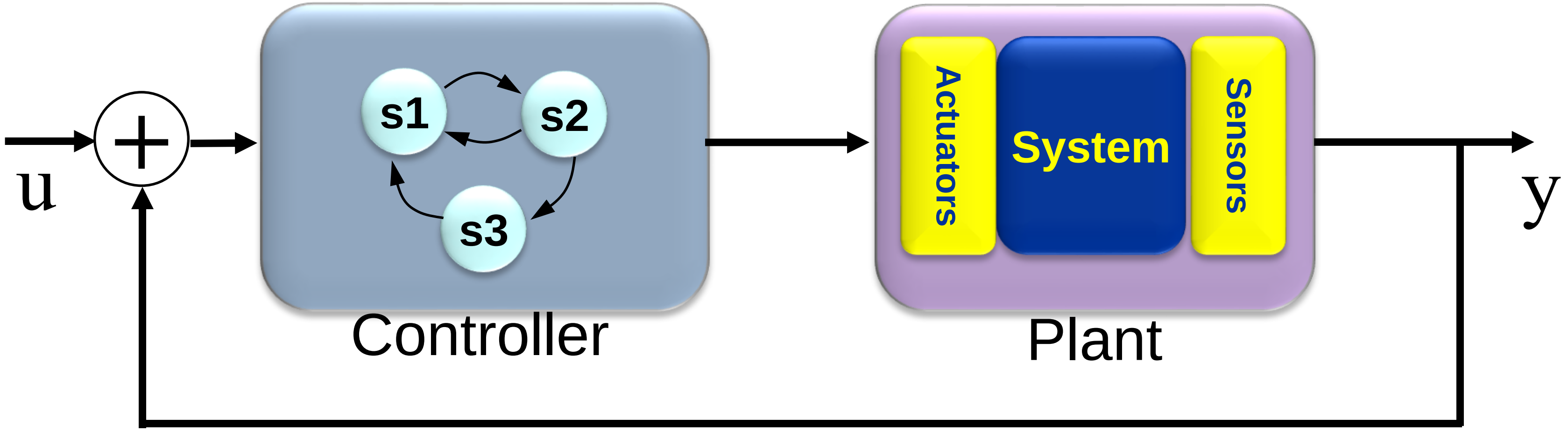


Model Based Design with Simulink

- Modelling and simulation
  - Multidomain Dynamic Systems
  - Nonlinear Systems
  - Continuous-time, Discrete-time, Multi-Rate systems
- Plant and Controller Design
  - Select/optimise control architecture and parameters
  - Rapidly model “what-if” scenarios
  - Communicate design ideas
  - Embody performance specifications
- Implementation
  - Automatic code generation
    - Embedded systems, FPGAs, GPUs
  - Rapid prototyping for HIL, SIL, PIL
  - Verification and validation

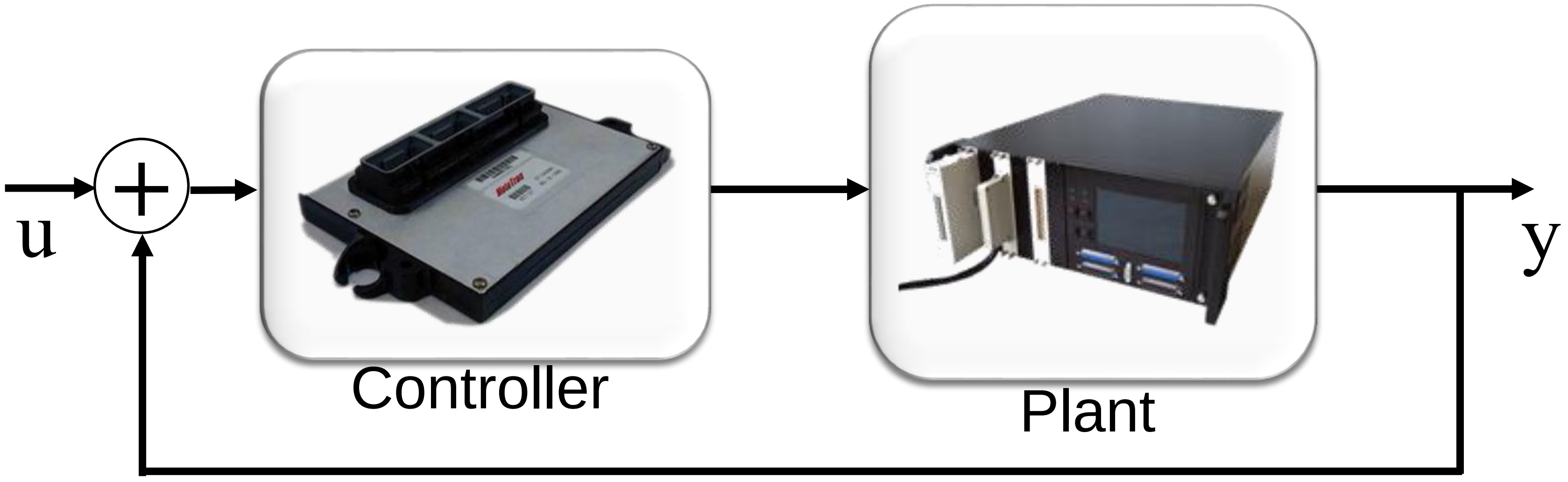


Optimise System-Level Performance



- Simulating plant and controller **in one environment** allows you to **optimize system-level performance**.
  - Automate tuning process using optimization algorithms
  - Accelerate process using parallel computing

Detect Integration Issues Earlier



- Controls engineers and domain specialists can work together to **detect integration issues in simulation**
  - Convert plant models to C code for hardware-in-the-loop tests
  - Share models with other internal users
  - Share models with external users while protecting IP

Modelling Approaches

First Principles Modelling | **Data-Driven Modelling**

Simulink

MATLAB

Simscape

Simulink
Design Optimization

System
Identification
Toolbox

Statistics and
Machine
Learning Toolbox

Tools for Modelling Dynamic Systems

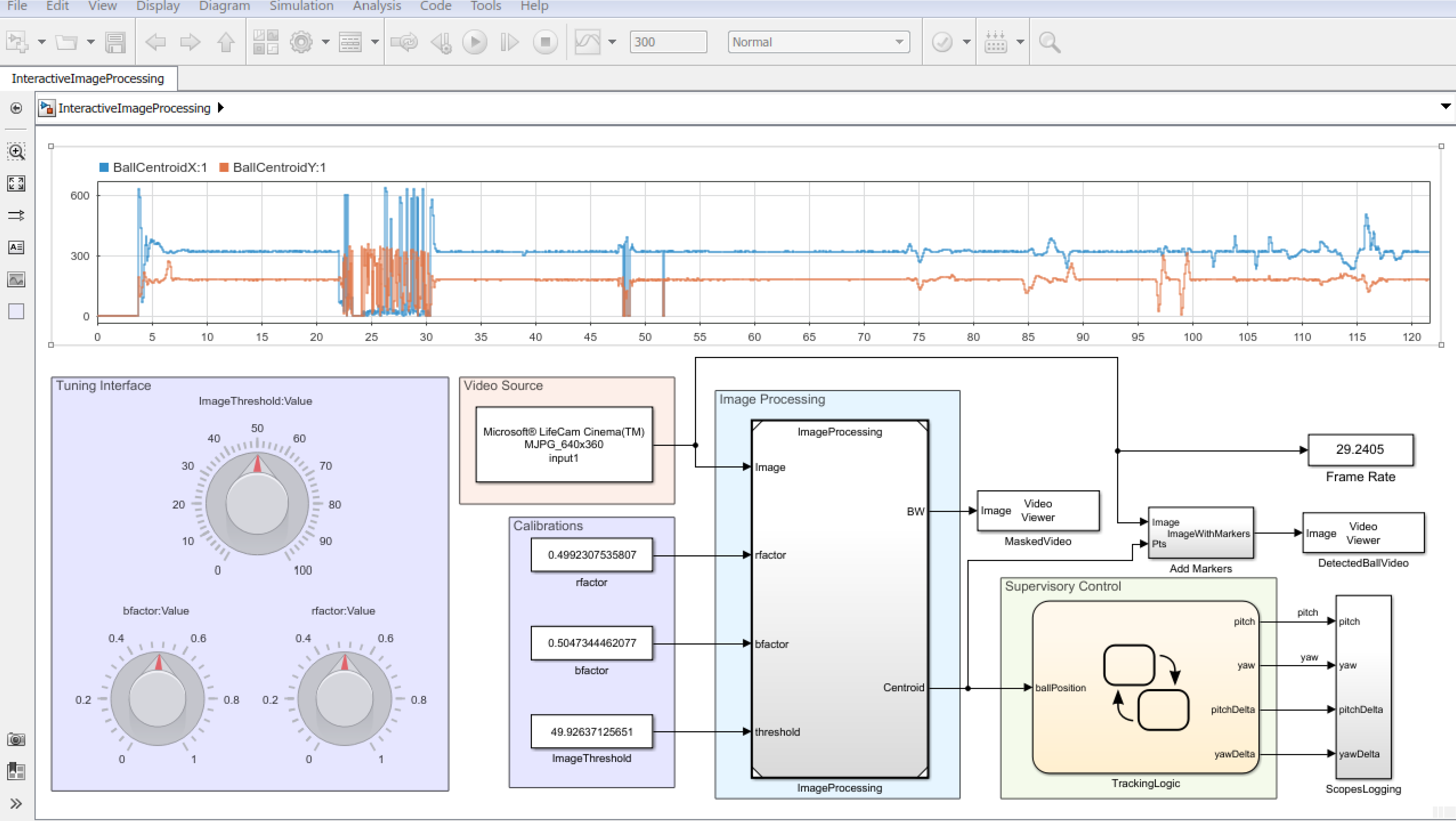
Using Simulink & Stateflow

Model-Based Design Application

+

+

- Rotate a camera to track an object
- Computer vision application
- Closed-loop motor control

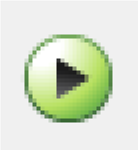

File Edit View Display Diagram Simulation Analysis Code Tools Help
300
Normal
InteractiveImageProcessing
BallCentroidX:1
BallCentroidY:1
Tuning Interface
ImageThreshold:Value
bfactor:Value
rfactor:Value
Video Source
Microsoft® LifeCam Cinema(TM)
MJPG_640x360
input1
Calibrations
0.4992307535807
rfactor
0.5047344462077
bfactor
49.92637125651
ImageThreshold
Image Processing
ImageProcessing
Image
rfactor
bfactor
threshold
BW
Centroid
Video Viewer
MaskedVideo
29.2405
Frame Rate
ImageWithMarkers
Pts
Add Markers
DetectedBallVideo
Supervisory Control
ballPosition
pitch
yaw
pitchDelta
yawDelta
TrackingLogic
ScopesLogging

What questions do we want to answer?

- Can I get the closed loop response I need?
- What current will my motor draw during operation?
- Does my system still work if component values change?
- What if…?

Steps in the process

1. Model the motor
2. Model the speed controller
3. Refine the motor model using measured data
4. Model the supervisory logic
5. Validate and integrate the image processing algorithm
6. Deploy the control model to hardware

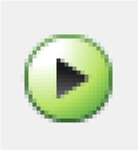
At each stage: **Simulate the model** 

Steps in the process

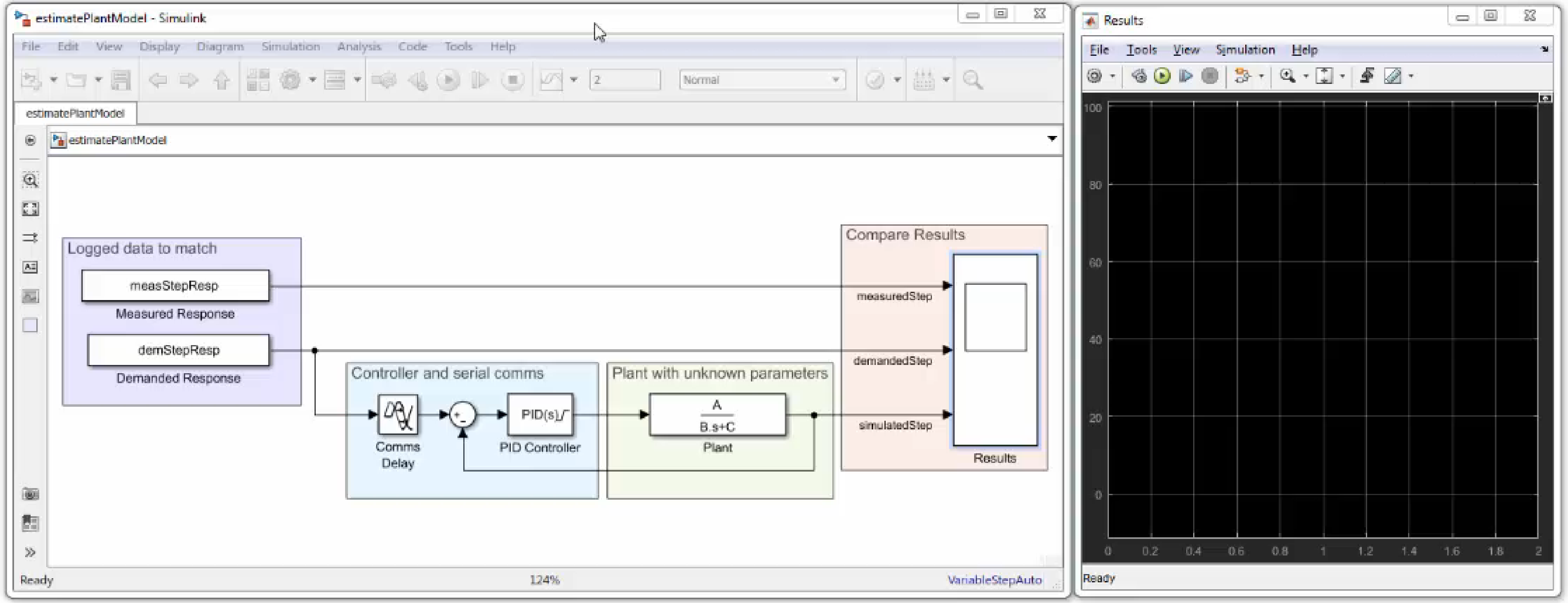
- ✓ Model the motor
- ✓ Model the speed controller

3. Refine the motor model using measured data
4. Model the supervisory logic
5. Validate and integrate the image processing algorithm
6. Deploy the control model to hardware

At each stage: **<u>Simulate the model</u>**



Parameter Estimation

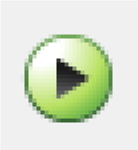


Steps in the process

✓ Model the motor
✓ Model the speed controller
✓ Refine the motor model using measured data

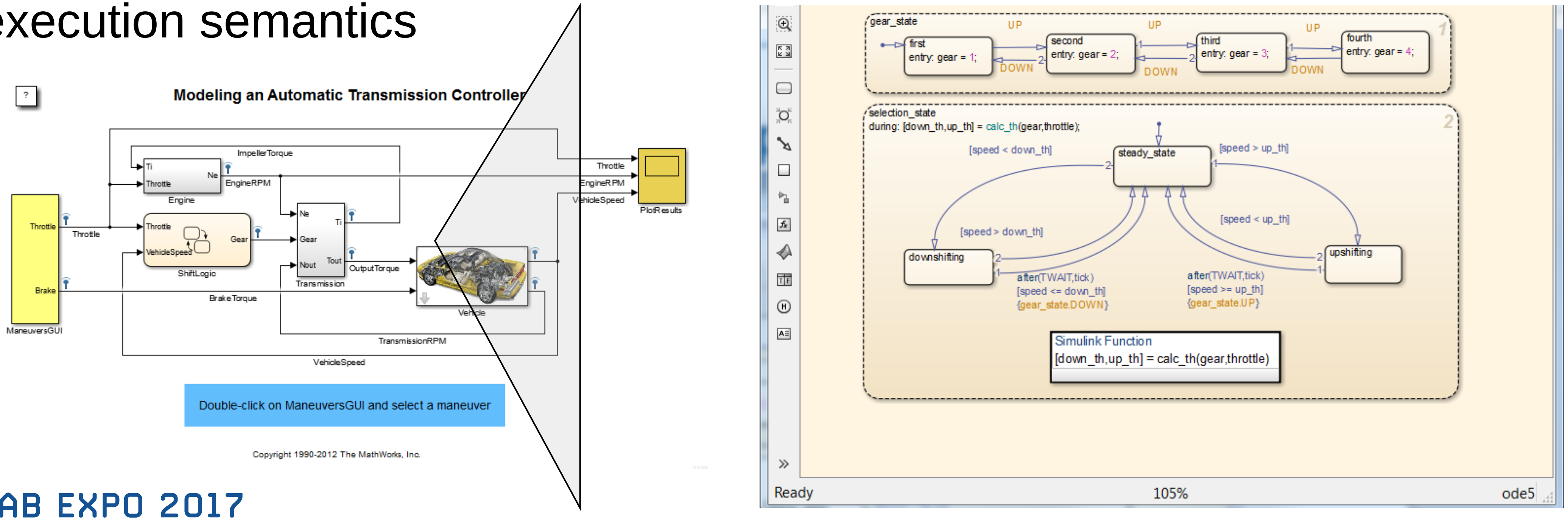
4. Model the supervisory logic
5. Validate and integrate the image processing algorithm
6. Deploy the control model to hardware

At each stage: **Simulate the model**



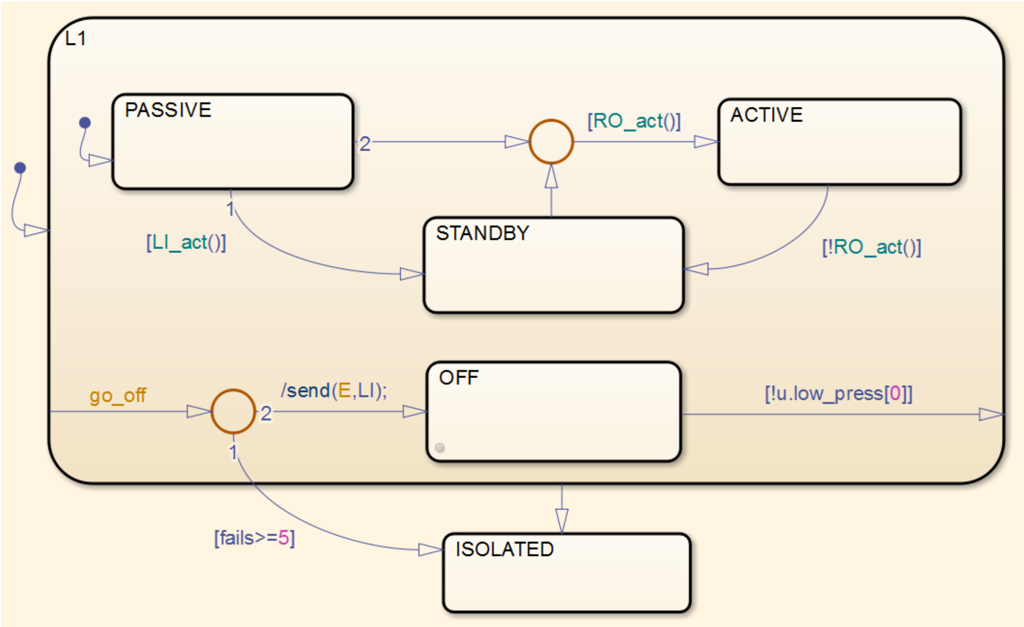
Stateflow Overview

- Extend Simulink with a design environment for developing state machines and flow charts
- Design systems containing control, supervisory, and mode logic
- Describe logic in a natural and understandable form with deterministic execution semantics



What are State Machines?

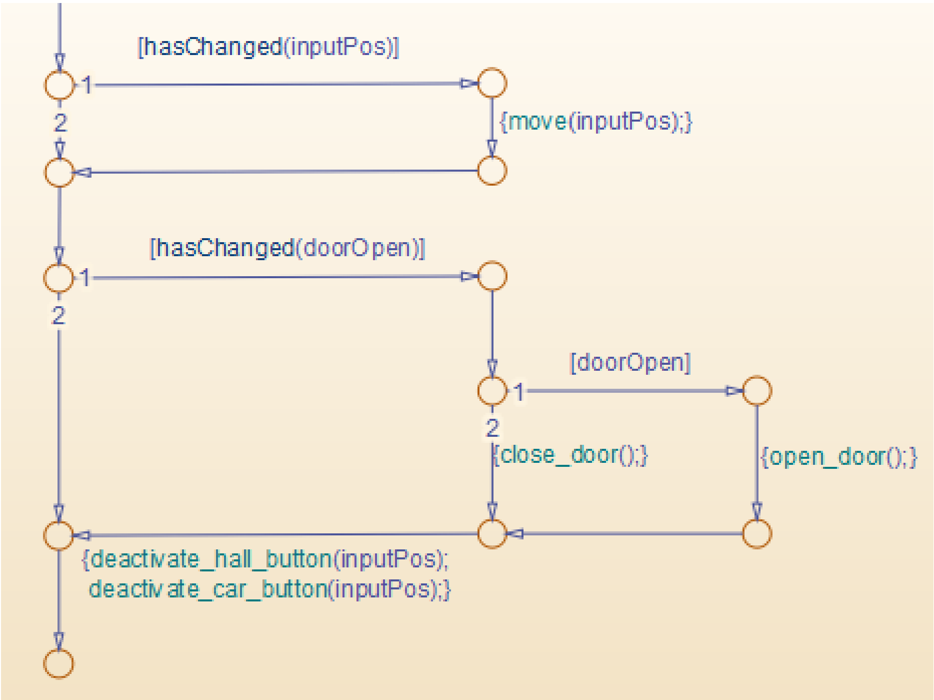
- Represent reactive systems that have states or modes
- States change based on defined conditions and events



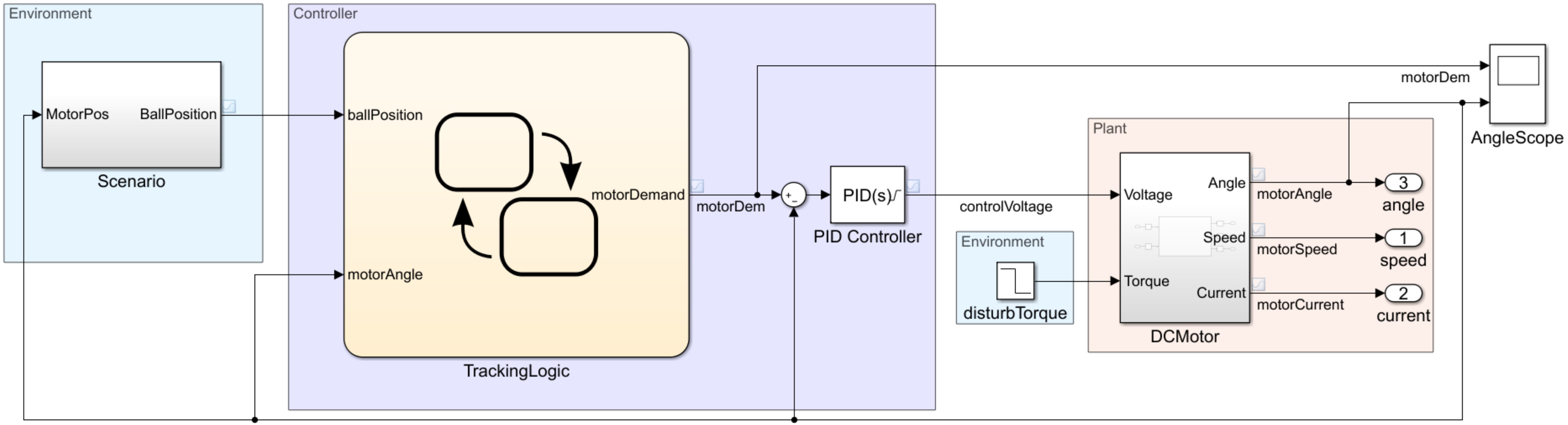
E.g. Fault Management

What are Flow Charts?

- Represent an algorithm or process



Modelling the system with Simulink and Stateflow



Next steps in the process

- ✓ Model the motor
- ✓ Model the speed controller
- ✓ Refine the motor model using measured data
- ✓ Model the supervisory logic

Visit the Demo Stations!

5. Validate and integrate the image processing algorithm
6. Deploy the control model to hardware

- ✓ Simulate the model

Conclusions

- Modelling and simulation gives you insight to make smarter decisions, earlier
- Simulink allows you to model the complete system in a single environment
- Accelerate your simulation work with the power of MATLAB

Solar Impulse Develops Advanced Solar-Powered Airplane

- Key design decisions made early
- Vital pilot training enabled
- Models reused and shared throughout development

"Simulations with MATLAB and Simulink were essential to assessing feasibility and evaluating broad design tradeoffs as well as making detailed design decisions—like the size of control surfaces and the vertical tail—that directly affect aircraft dynamics and handling qualities."

–Ralph Paul, Solar Impulse