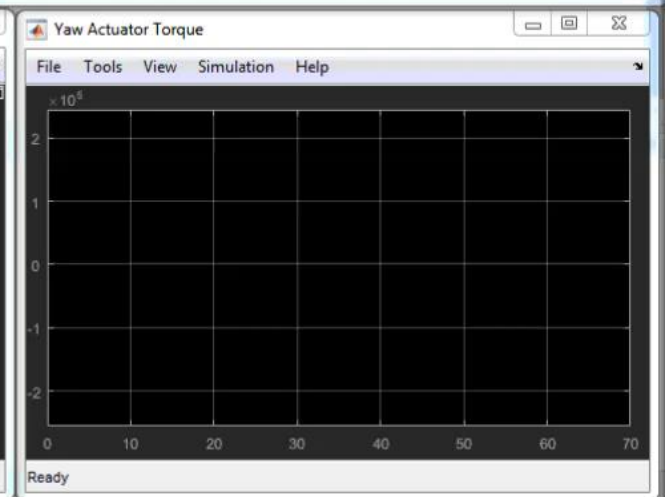
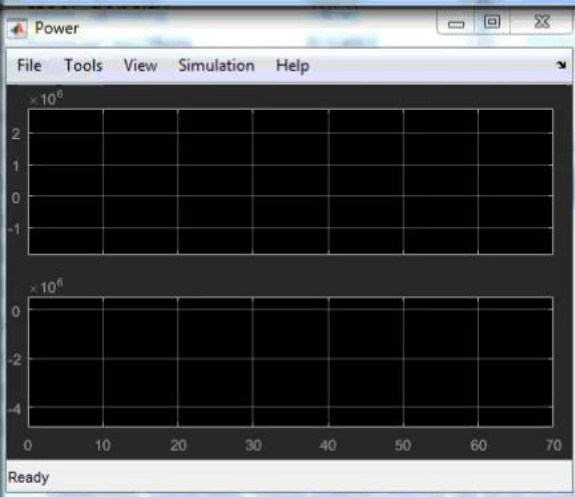
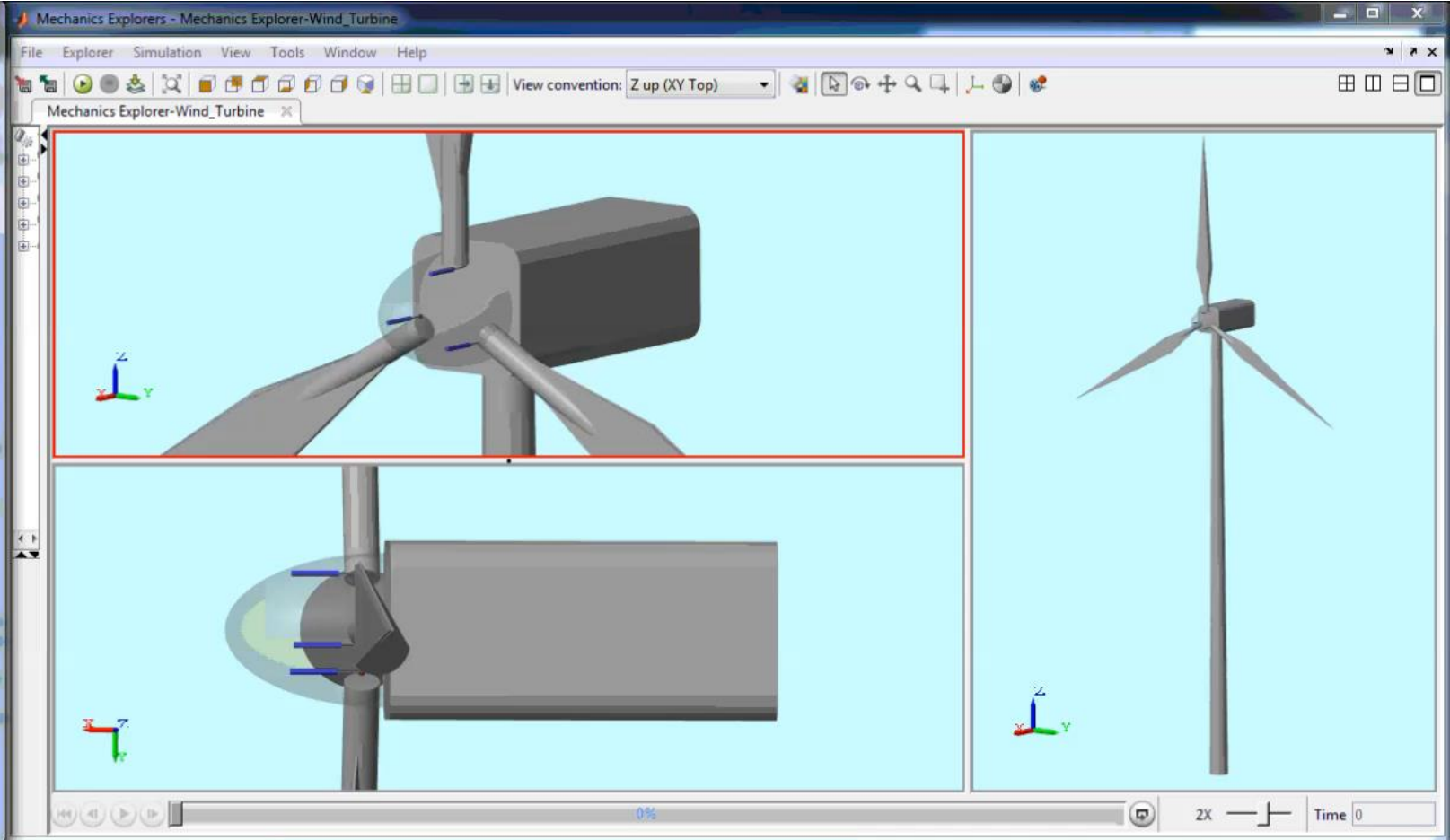
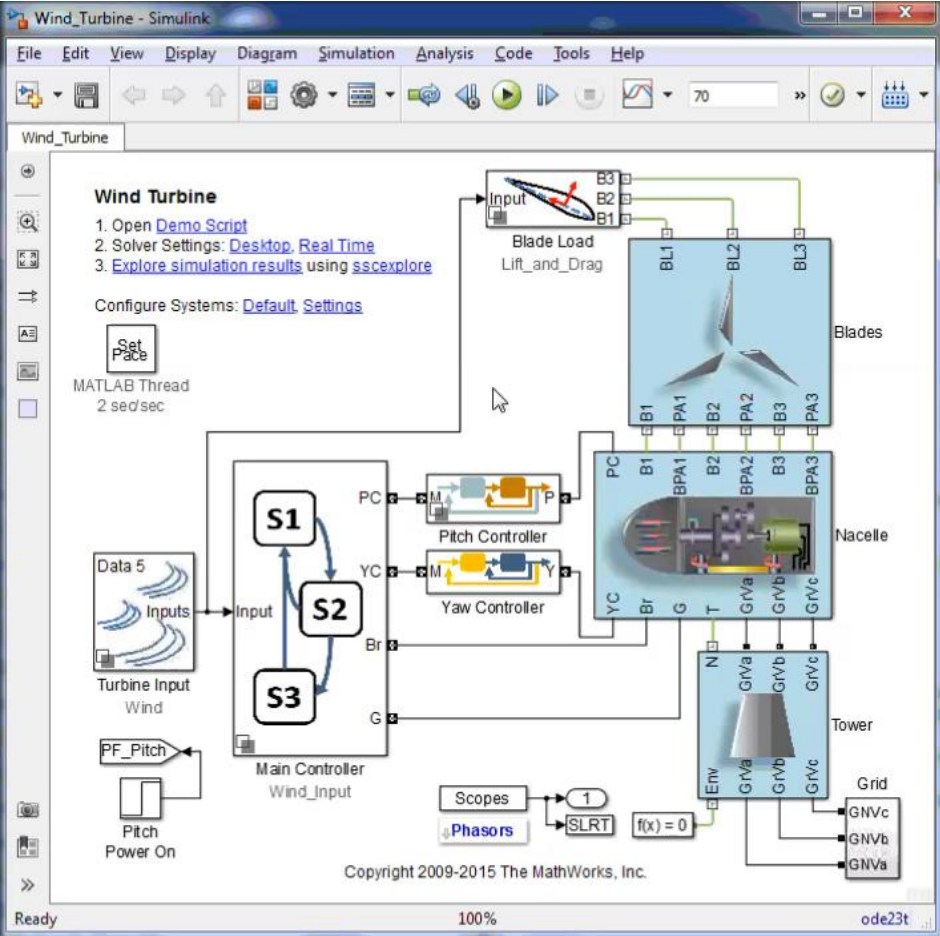


The background features a dark blue field on the left and a lighter blue field on the right, separated by a diagonal line. In the upper right, there are white, stylized waveforms. In the lower right, there is a 3D wireframe mesh with a color gradient from yellow at the top to blue at the bottom. Faint blue circuit-like patterns are visible in the bottom right corner.

MATLAB EXPO 2017

Introduction to Simulink & Stateflow

Jonathan Agg



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?

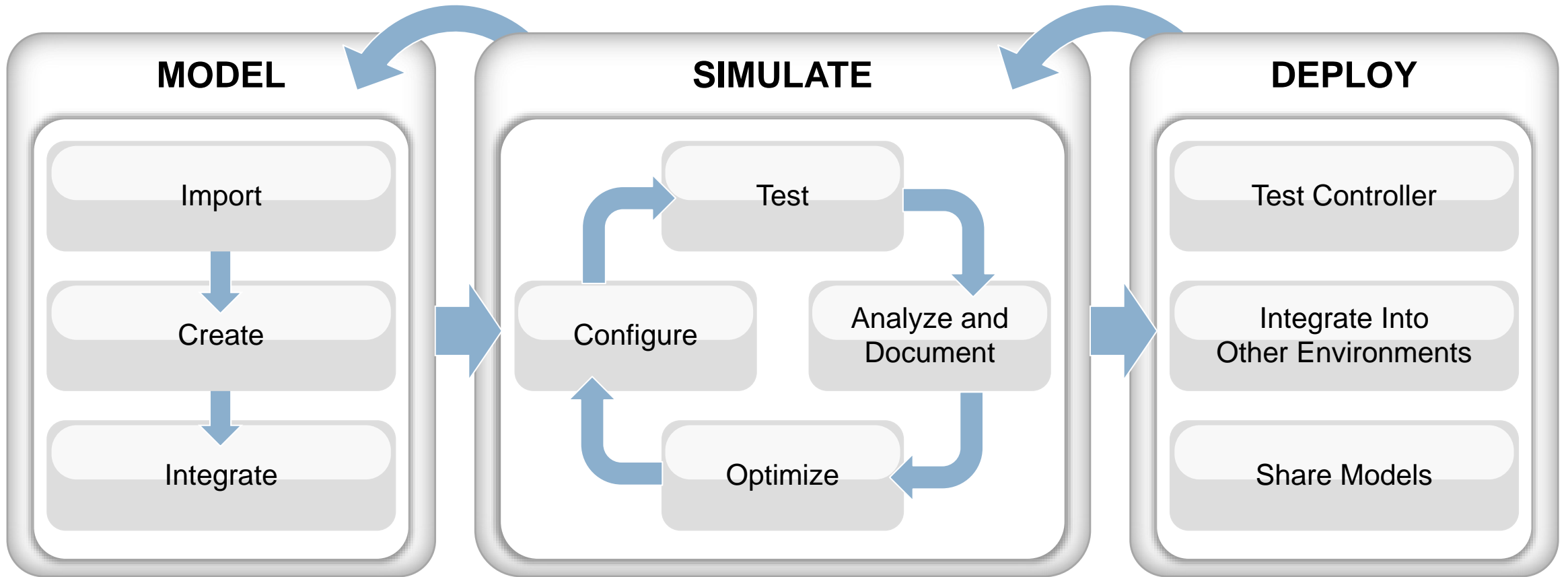
Modelling & Simulation gives you insight



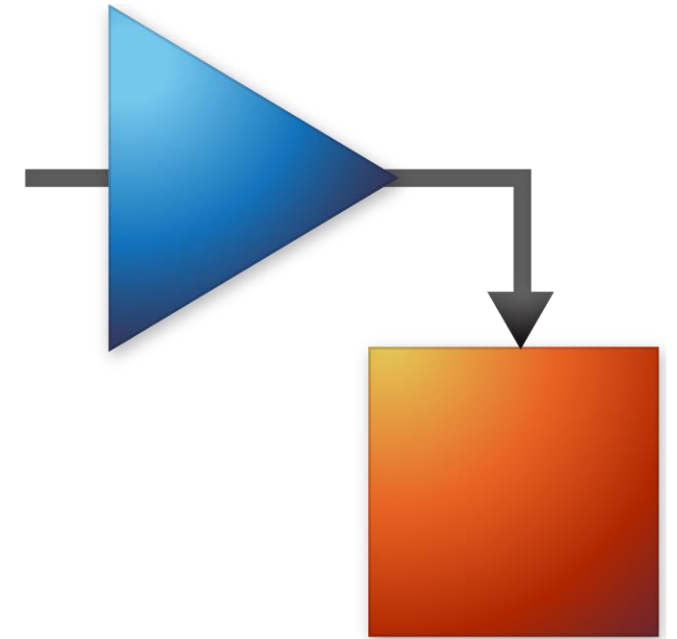
Image credit: McLaren



Image credit: Peter Gronemann | Wikipedia

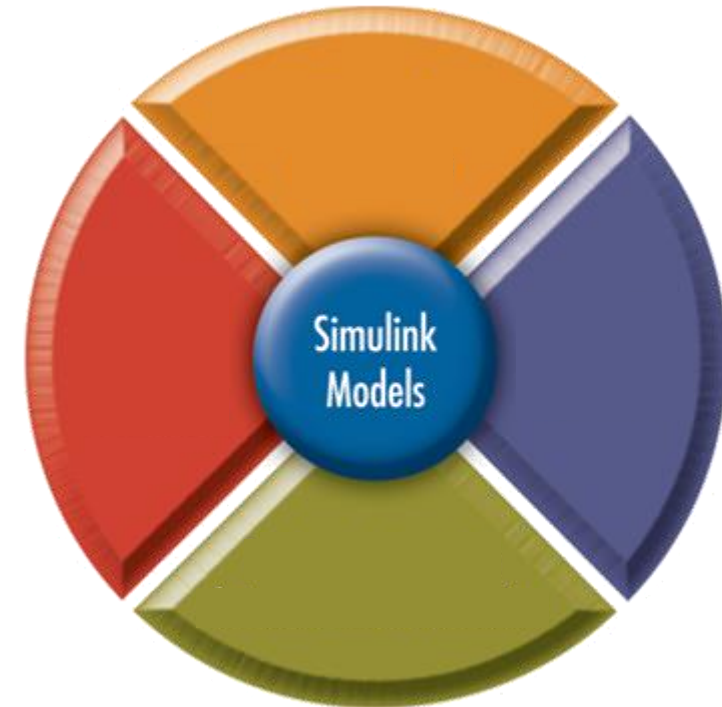


Why use Simulink?

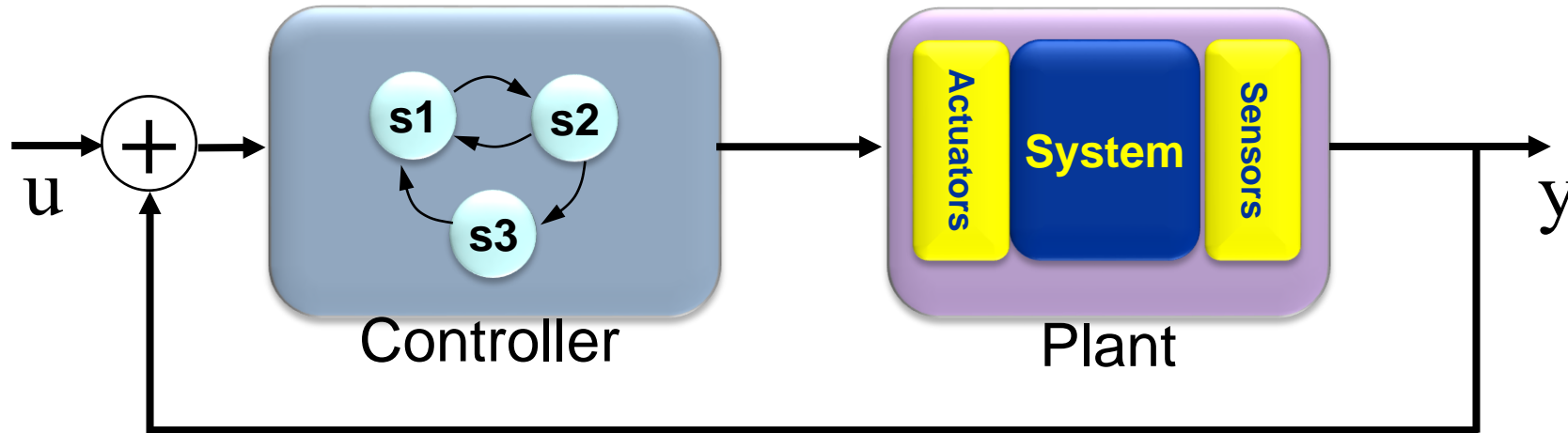


Model Based Design with Simulink

- Modelling and simulation
 - Multidomain Dynamic Systems
 - Nonlinear Systems
 - Continuous-time, Discrete-time, Multi-Rate systems
- Plant and Controller Design
 - Select/optimize 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

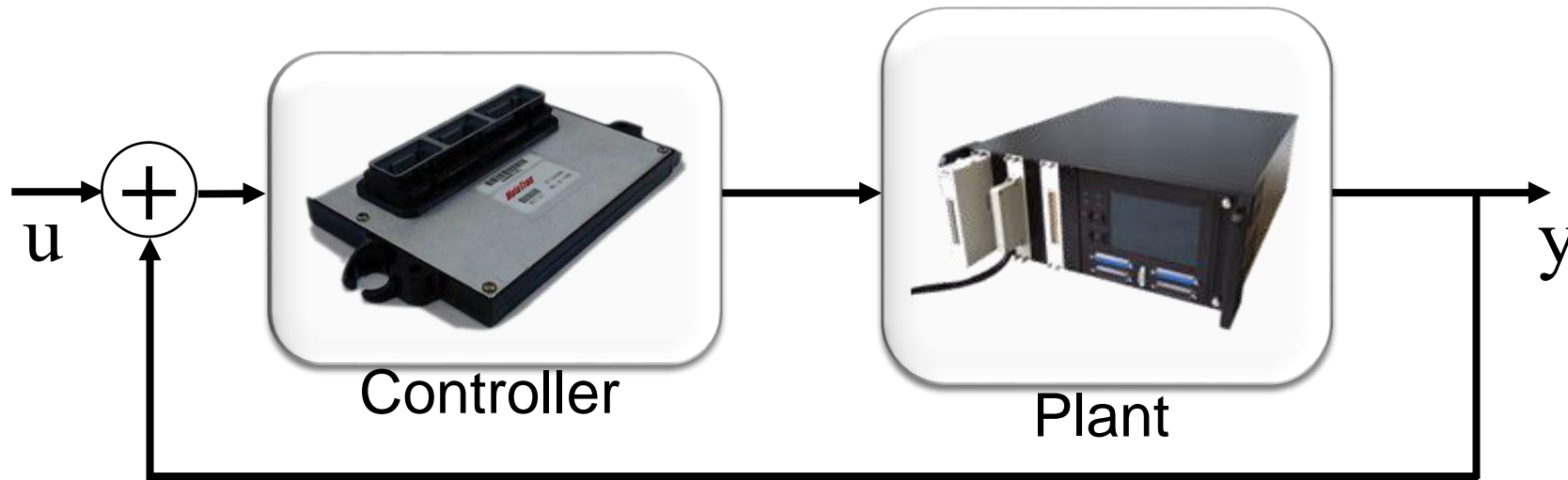


Optimise System-Level Performance



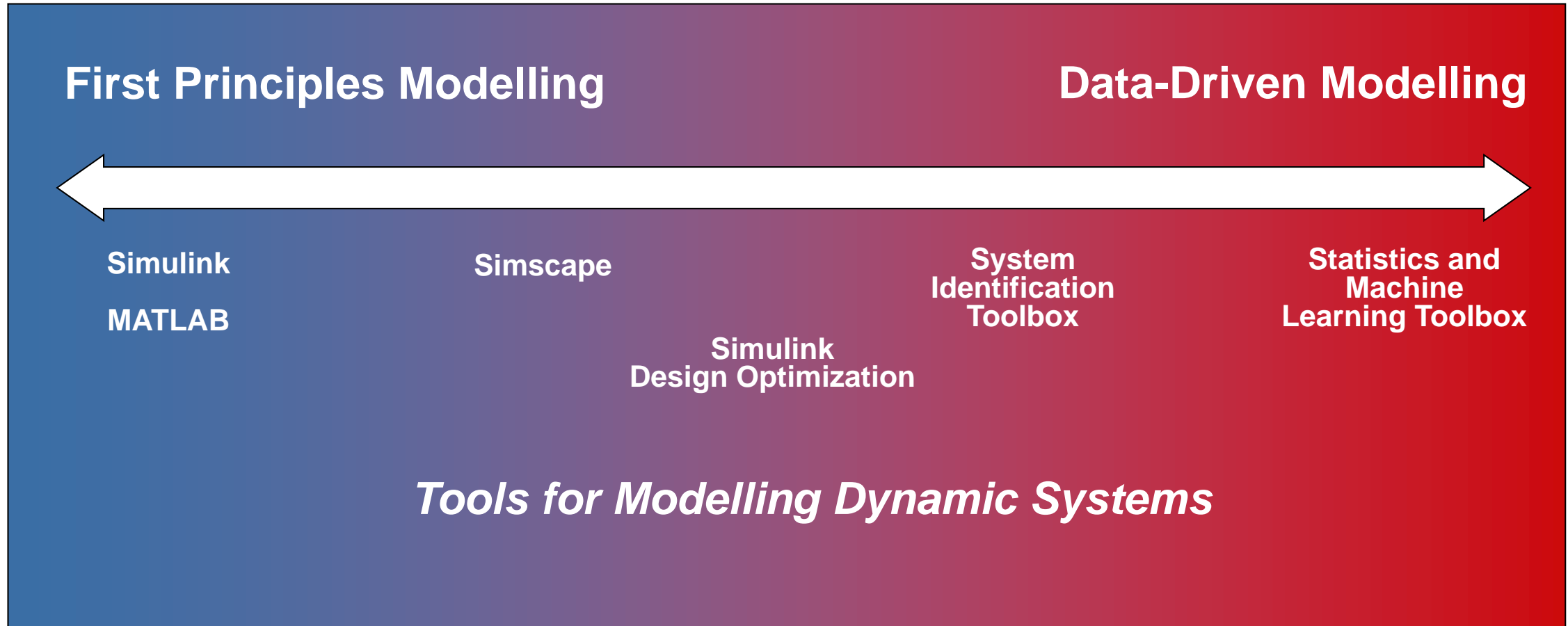
- Simulating plant and controller **in one environment** allows you to **optimize system-level performance**.
 - Automate tuning process using optimization algorithms
 - Accelerate process using parallel computing

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

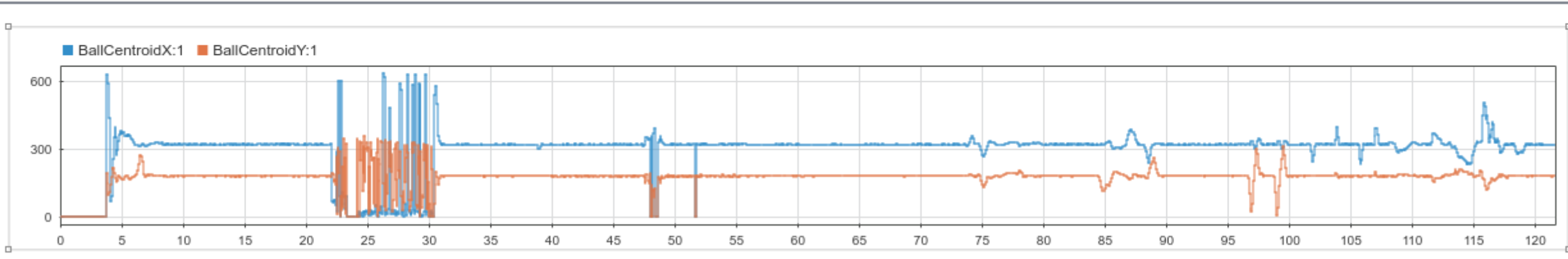


Using Simulink & Stateflow

Model-Based Design Application



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



Tuning Interface

ImageThreshold:Value

0 10 20 30 40 50 60 70 80 90 100

bfactor:Value

0.2 0.4 0.6 0.8 1

rfactor:Value

0 0.4 0.6 0.8 1

Video Source

Microsoft® LifeCam Cinema(TM)
MJPEG_640x360
input1

Calibrations

0.4992307535807
rfactor

0.5047344462077
bfactor

49.92637125651
ImageThreshold

Image Processing

Image Processing

Image

ImageProcessing

BW

Centroid

ImageProcessing

Supervisory Control

Image Video Viewer
MaskedVideo

Image ImageWithMarkers
Add Markers

Image Video Viewer
DetectedBallVideo

29.2405
Frame Rate

pitch

yaw

pitchDelta

yawDelta

ScopesLogging

TrackingLogic

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**



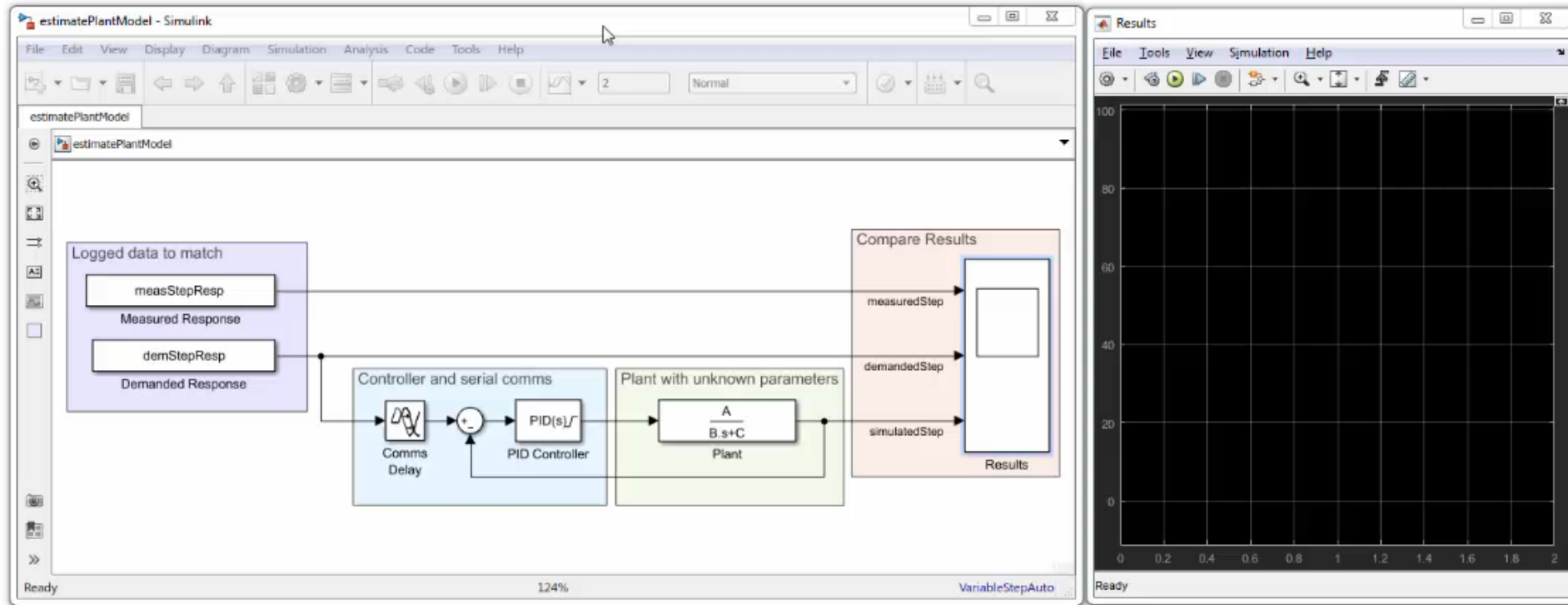
Steps in the process

- ✓ 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: **Simulate the model**



Parameter Estimation



Steps in the process

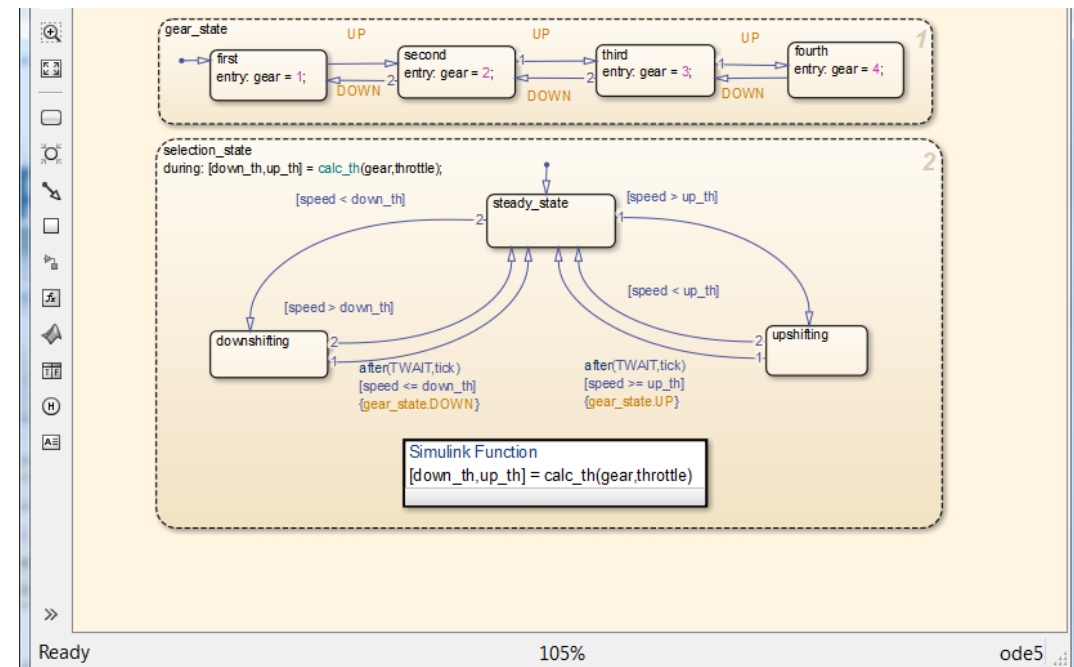
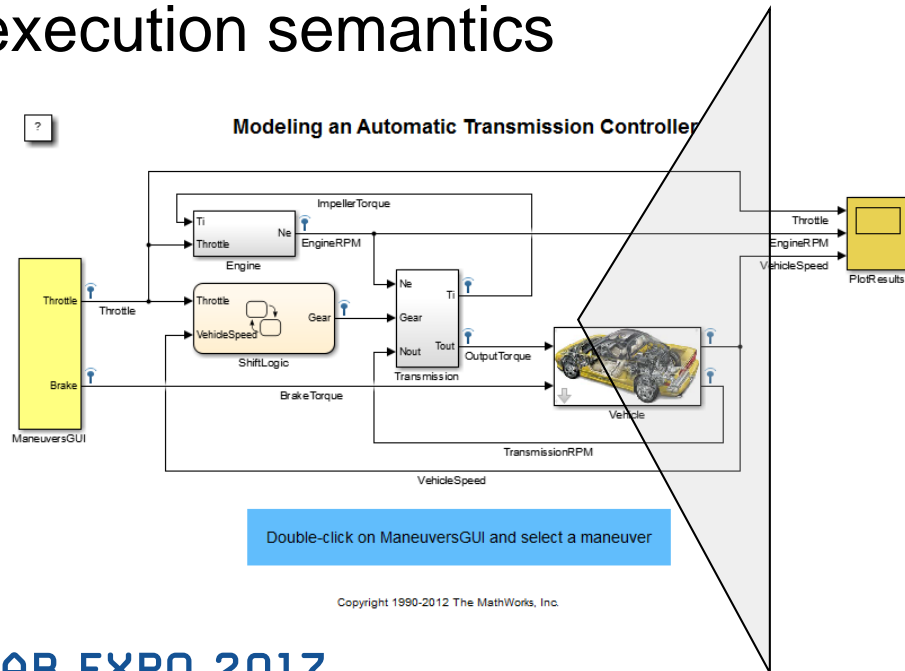
- ✓ Model the motor
- ✓ Model the speed controller
- ✓ Refine the motor model using measured data
- 4. Model the supervisory logic
- 5. Validate and integrate the image processing algorithm
- 6. Deploy the control model to hardware

At each stage: **Simulate the model**



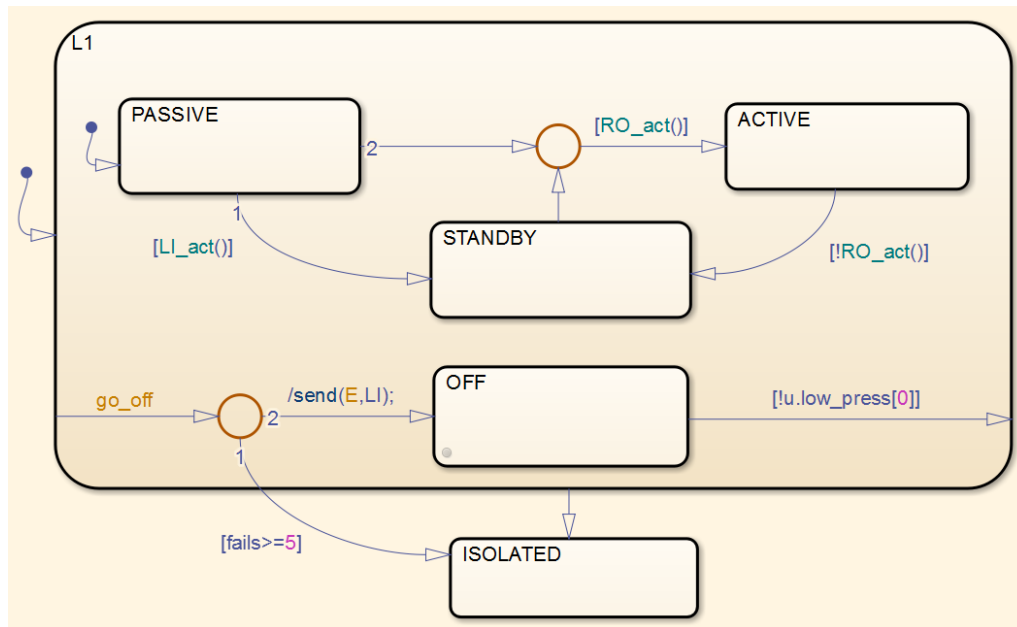
Stateflow Overview

- 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?

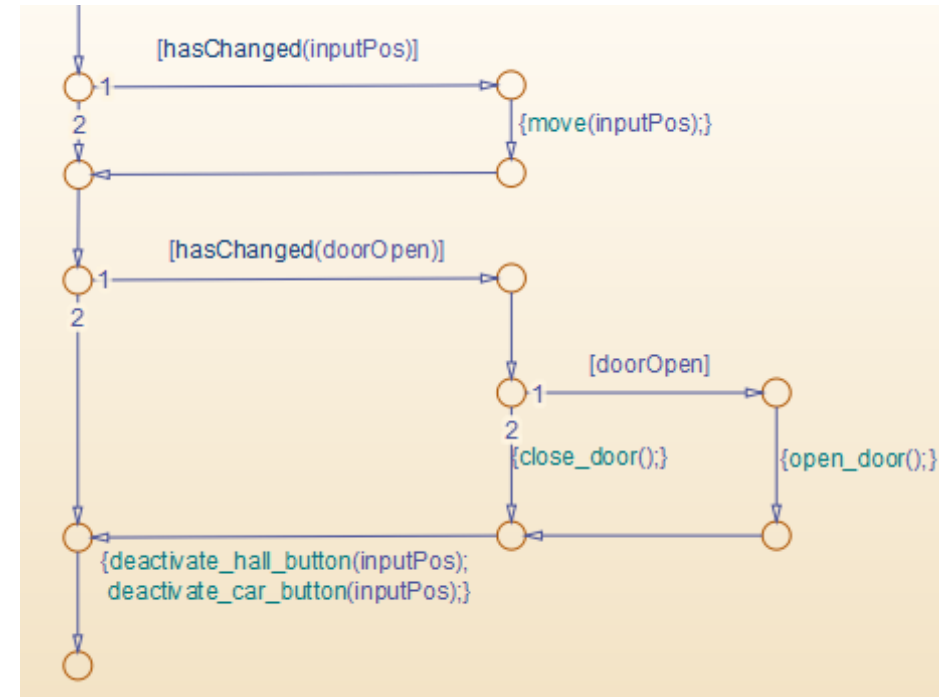
- Represent reactive systems that have states or modes
- States change based on defined conditions and events



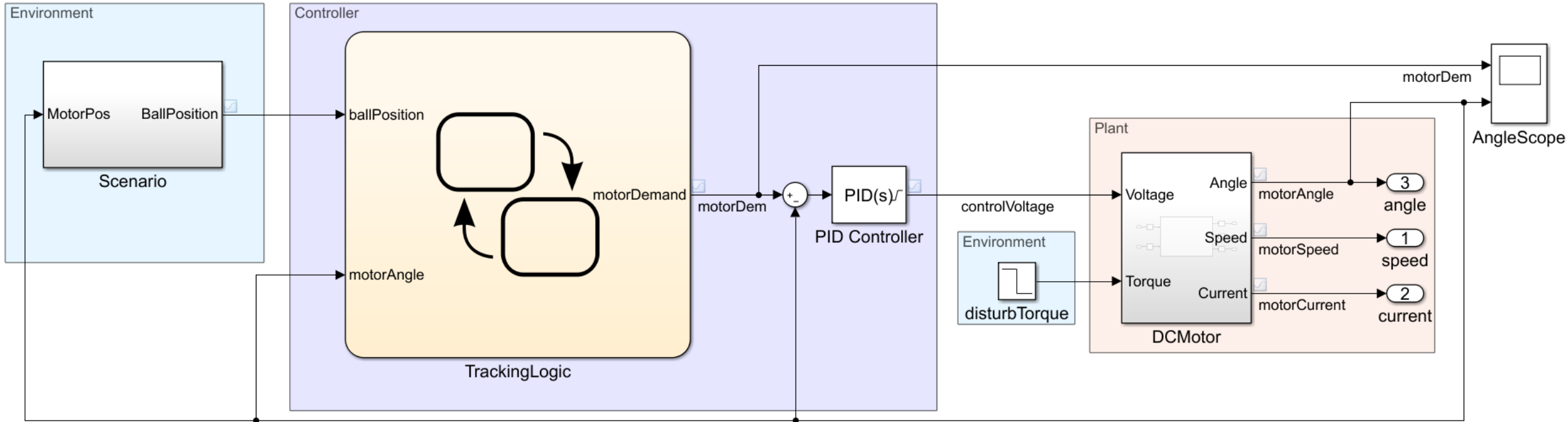
E.g. Fault Management

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

Visit the
Demo
Stations!

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