MATLAB EXPO 2016

Introduction to Simulink & Stateflow

GianCarlo Pacitti, MathWorks
Topics we will address this session

- Why do organisations use Simulink and Stateflow?
- Getting to grips with the basics of Simulink and Stateflow through a worked example
Why model a system?
Modelling & Simulation gives you insight
Traditional Development Process

**RESEARCH**
- Mechanical Components
- Electrical Components

**SPECIFICATIONS**
- Requirement Documents
  - Difficult to analyze
  - Difficult to manage as they change
- Paper Specifications
  - Easy to misinterpret
  - Difficult to integrate with design
- Physical Prototypes
  - Incomplete and expensive
  - Prevents rapid iteration
  - No system-level testing
- Manual Coding
  - Time consuming
  - Introduces defects and variance
  - Difficult to reuse
- Traditional Testing
  - Design and integration issues found late
  - Difficult to feed insights back into design process
  - Traceability

**DESIGN**
- EDA
  - Electrical Components
- Algorithm Design
  - Embeddable Algorithms
- MCAD/MCAE
  - Mechanical Components

**IMPLEMENTATION**
- C/C++
  - Embedded Software

**INTEGRATION AND TEST**
Model-Based Design

- Requirements Definition
- Desktop Modeling and Simulation
- Code Generation
- Design

- Control System
- Physical System or Process

- Realization
- Validation
- Hardware-in-the-Loop (HIL)
- Rapid Control Prototyping (RCP)
Why use Simulink?
1. Common environment for all disciplines

2. Integrated workflow from requirements to code

3. Integration with MATLAB
Using Simulink & Stateflow
Model-Based Design Application

- Rotate a camera to track an object
- Computer vision application
- Closed-loop motor control
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…?
Introduction to Simulink

- Block-diagram environment
- Model, simulate, and analyze multidomain systems
- Design, implement, and test:
  - Control systems
  - Signal processing systems
  - Communications systems
  - Other dynamic systems
- Platform for Model-Based Design
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**
Decision Flows and State Machines
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

What are Flow Charts?

- Represent an algorithm or process

E.g. Fault Management
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
- Simulate the model
- Validate and integrate the image processing algorithm
- Deploy the control model to hardware

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