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
Traditional Development Process

- **RESEARCH**
- **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**

MATLAB EXPO 2018
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)
Model-Based Design

**MODEL**
- Import
- Create
- Integrate

**SIMULATE**
- Test
- Configure
- Analyze and Document
- Optimize

**DEPLOY**
- Test Controller
- Integrate Into Other Environments
- Share Models

MATLAB EXPO 2018
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
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…?
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**
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
Modelling the system with Simulink and Stateflow
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