MATLAB EXPO 2018

Progettazione meccatronica per sistemi avionici

Aldo Caraceto
Key Points

- Simulating the system in one environment enables better design of higher quality controls.

- Testing different actuator designs, having different levels of detail, in one environment saves time and encourages innovation.

- Plant model supports the entire development process.
Agenda

- Example: Flight actuation system
  - Benefits of Model-Based Design

- Actuator design
  - Link requirements and design
  - Modeling the mechanical system
  - Determining actuator requirements
  - Tradeoff studies

- Optimizing system performance
  - Tune controller automatically

- Model deployment
  - HIL testing
  - Protecting IP
Example: Aileron Actuation System

- **System**
  
  Desired Angle → Controller → Actuator Force → Measured Angle

- **Simulation goals**
  1. Determine requirements for actuation system
  2. Test actuator designs
  3. Optimise system performance
  4. Run simulation on real-time hardware for HIL tests
Aileron Actuation System – Simulink Model
Traditional Design Process

**REQUIREMENTS**

- Cannot validate design against requirements
- Cannot test or optimize fully integrated design
- Can only find problems using hardware prototypes
- Manual coding is slow, buggy, and hard to verify

**DESIGN**

- Control
- Mechanical
- Electrical

**IMPLEMENTATION**

- Emb. Code

**INTEGRATION AND TEST**
Model-Based Design

**Requirements**
- Detect errors right away with continuous verification

**System Level Design**
- Optimize design in a single simulation environment

**Implementation**
- Lower costs using HIL tests
- Save time/increase quality by automatically generating embedded code

**Integration and Test**
- Simscape

**Test & Verification**
- Cannot validate design against requirements
- Manual coding is slow, buggy, and hard to verify
- Can only find problems using hardware prototypes

**Matlab Expo 2018**
Agenda

- Example: Flight actuation system
  - Benefits of Model-Based Design

- Actuator design
  - Link requirements and design
  - Modeling the mechanical system
  - Determining actuator requirements
  - Tradeoff studies

- Optimizing system performance
  - Tune controller automatically

- Model deployment
  - HIL testing
  - Protecting IP
**Link Specification and Design**

**Situation:**

![Diagram of an aileron system with part numbers labeled b and d]

**Problem:** Difficult to check design against specification.

**Solution:** Link design and specification using Simulink Requirements.

---

MATLAB EXPO 2018
Modeling the Mechanical System

**Problem:** Model the mechanical system within Simulink

**Solution:** Import the mechanical model from CAD into Simscape Multibody

MATLAB EXPO 2018
Modeling the Mechanical System

System:

Problem: Model the mechanical system within Simulink

Solution: Import the mechanical model from CAD into Simscape Multibody

MATLAB EXPO 2018
Determining Actuator Requirements

**Problem:** Determine the requirements for an aircraft aileron actuator

**Solution:** Use Simscape Multibody to model the aileron and use inverse dynamics to determine the required force.

MATLAB EXPO 2018
Determining Actuator Requirements
Determining Actuator Requirements

Problem: Determine the requirements for an aircraft aileron actuator

Solution: Use Simscape Multibody to model the aileron and use inverse dynamics to determine the required force
Testing Electrical and Hydraulic Designs

Problem: Select type of actuator based on system-level requirements

Solution: Use Simscape Fluids and Simscape Electronics to model the actuators, and variant subsystems to test them

MATLAB EXPO 2018
Testing Electrical and Hydraulic Designs
Problem: Select type of actuator based on system-level requirements

Solution: Use Simscape Fluids and Simscape Electronics to model the actuators, and variant subsystems to test them.
Adjusting Fidelity Using Simscape Electronics Components

- Semiconductors, Motors, Sensors, Op-Amps and Logic, Passive Devices

- Switching and signal amplification
  - Parameterize with data sheets
  - Simple and detailed variants

- Thermal effects
  - Effect on behavior
  - Heat transfer to environment

- Measure power losses

```
>> elec_getPowerLossSummary(solar_converter_simlog)

<table>
<thead>
<tr>
<th>LoggingNode</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>'elec_solar_converter.D1'</td>
<td>0.96137</td>
</tr>
<tr>
<td>'elec_solar_converter.MOS1'</td>
<td>16.173</td>
</tr>
<tr>
<td>'elec_solar_converter.MOS3'</td>
<td>21.834</td>
</tr>
</tbody>
</table>
```
Adjusting Fidelity Using Simscape Electronics Components
Semiconductors, Motors, Sensors, Op-Amps and Logic, Passive Devices

- Translational and rotational actuators
  - Parameterize with data sheets or with data from FEM software
  - Specify electrical losses

- Thermal effects
  - Temperature dependent behavior
  - Heat transfer to environment

- Include or neglect switching effects
Adjusting Fidelity Using Simscape Electronics Components
Semiconductors, Motors, Sensors, Op-Amps and Logic, Passive Devices

- Includes electronic, thermal, and mechanical sensors
  - Analog and digital
  - Parameterization options
  - Include or neglect sensor bandwidth

- Test effects of sensor damage or failure on system performance
Adjusting Fidelity Using Simscape Electronics Components
Semiconductors, Motors, Sensors, Op-Amps and Logic, Passive Devices

- Behavioral models for fast simulation
  - Similar behavior to models with transistor implementation
  - Enables testing of larger circuits in less time

- Use models to perform high-level design
  - Avoid nonlinear effects during normal circuit operation

MATLAB EXPO 2018
Adjusting Fidelity Using Simscape Electronics Components
Semiconductors, Motors, Sensors, Op-Amps and Logic, Passive Devices

- Linear and nonlinear devices
  - Enable physical effects
- Specify operating limits and tolerances
  - Model realistic behavior
- Test effects of component failure on system level performance
  - Fault modeling
Agenda

- Example: Flight actuation system
  - Benefits of Model-Based Design

- Actuator design
  - Link requirements and design
  - Modeling the mechanical system
  - Determining actuator requirements
  - Tradeoff studies

- Optimizing system performance
  - Tune controller automatically

- Model deployment
  - HIL testing
  - Protecting IP
Optimizing System Performance

**Model:**

- Speed Control
- Angle
- Current

**Problem:** Optimize the speed controller to meet system requirements

**Solution:** Tune controller parameters with Simulink Design Optimization

![Graph showing Aileron Angle and Actuator Force over time with tuned controller parameters]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_p$</td>
<td>0.62</td>
</tr>
<tr>
<td>$K_i$</td>
<td>0.29</td>
</tr>
</tbody>
</table>
Optimizing System Performance
Optimizing System Performance

**Problem:** Optimize the speed controller to meet system requirements

**Solution:** Tune PID parameters with Simulink Control Design

**Model:**

```
<table>
<thead>
<tr>
<th>Speed Control</th>
<th>Current Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \omega )</td>
<td>( i )</td>
</tr>
</tbody>
</table>
```

**Diagram:**

- **Aileron Angle**
  - Command and Measured
- **Actuator Force**
  - Time (s)

**Parameters:**

- \( K_p = 0.62 \)
- \( K_i = 0.29 \)
Agenda

- Example: Flight actuation system
  - Benefits of Model-Based Design

- Actuator design
  - Link requirements and design
  - Modeling the mechanical system
  - Determining actuator requirements
  - Tradeoff studies

- Optimizing system performance
  - Tune controller automatically

- Model deployment
  - HIL testing
  - Protecting IP
Configuring a Hydraulic Actuator for HIL Testing

Problem: Configure solvers to minimize computations and convert to C code for real-time simulation

Solution: Use Simscape local solvers on stiff physical networks and Simulink Coder™ to generate C code
Sharing Models and Protecting Intellectual Property

**Situation:**

- Physical System
- Model using Simscape and add-on products
- Share

**Problem:** Share a component or library with others that does not expose the source code.

**Solution:** Use the Model Reference Protected Mode from Simulink to protect intellectual property.

- Simulate
- Change parameter values
- Does not require licenses for Simscape add-on products
- Source code protected

**Situation:**

- Ref_Model.slx
- Model.slx
- Ref_Model.slx
- Simulink.ModelReference.protect

**Component Protected**
Key Points

- Simulating the system in one environment enables to design higher quality controls.

- Testing different actuator designs, having different levels of detail, in one environment saves time and encourages innovation.

- Plant model supports the entire development process.
MATLAB EXPO 2018

Progettazione meccatronica per sistemi avionici

Aldo Caraceto