Synchronous Machine Modelling Using Simscape

Peenki Rani
MATLAB EXPO 2019 – Silverstone, UK
2nd October 2019
Agenda

- Who We Are
- Product Validation Challenges
- Project Introduction
- Why MATLAB? Why Simscape?
- Approach
- Execution of a System
- Outcome
- Future of Modelling
- How to Model Complex Systems
- Tips
- Conclusion
Who We Are

Engine Business

Power Generation Business

Components Business

Distribution Business

Power Generation

Emission Solutions

Distribution

Generator Technologies

Turbo Technologies

Fuel Systems

Filtration

4 – 11,200 kVA
Who We Are

- G-Drive Engine
- Emission Solutions
- Air handling system
- Turbo Technologies
- Turbocharger
- Control systems
- Steelwork/brackets
- Generator/Alternator
- Baseframe/enclosures
- Filtration
- Filtration system

4 – 11,200 kVA
Who We Are
Applications

Prime Power:
Supplying continuous power 24/7 for seven years supporting construction of one of the world’s largest natural gas projects

Marine:
Diving support vessel for saturation and air diving support work

Mining:
Alternators required for 58 MW power plant at a remote iron ore mining site
Product Validation Challenges

- Costly and time consuming experimental testing methods
- Remote location testing requirement
- Challenging applications and fault investigation
Project Introduction

Challenge:
- Time consuming and expensive
- Remote locations applications

Solution:
- Simscape for plant
- Simulink for the controls
- MATLAB to validate and automate
- Appdesigner to deploy

Benefit:
- Reduced commission time
- Application validation
- Fault simulations
- Customer enquiries

Therefore, enhancing simulation capabilities.
Why MATLAB? Why Simscape?

- Multiple physical domains
- Pre-validated model blocks
- Design optimisation
- Flexible environment
- Cummins adopted software package
Approach

Learned MATLAB
Replicate Test
Define scope
Design & build
Validated
Deploy
Approach

Learned MATLAB

Replicate Test

Define scope

Design & build

Validated

Deploy

STAMFORD S7
Execution of a System
Outcome

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Test Results</th>
<th>Model Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xd direct axis synchronous reactance</td>
<td>2.36</td>
<td>2.3288</td>
</tr>
<tr>
<td>X'd direct axis transient reactance</td>
<td>0.16</td>
<td>0.15655</td>
</tr>
<tr>
<td>X&quot;d direct axis sub transient reactance</td>
<td>0.13</td>
<td>0.14864</td>
</tr>
<tr>
<td>T'd Transient Time Constant</td>
<td>0.164</td>
<td>0.16351</td>
</tr>
<tr>
<td>T&quot;d Sub-Transient Time Constant</td>
<td>0.0076</td>
<td>0.010014</td>
</tr>
<tr>
<td>Ta armature time constant</td>
<td>0.037</td>
<td>0.030489</td>
</tr>
</tbody>
</table>
### Outcome

**MATLAB Simulation Data**

**Tech Centre Test Data**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Test Results</th>
<th>Model Results</th>
<th>Accuracy %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xd direct axis synchronous reactance</td>
<td>2.36</td>
<td>2.3288</td>
<td>96.88</td>
</tr>
<tr>
<td>X'd direct axis transient reactance</td>
<td>0.16</td>
<td>0.15655</td>
<td>99.66</td>
</tr>
<tr>
<td>X&quot;d direct axis sub transient reactance</td>
<td>0.13</td>
<td>0.14864</td>
<td>98.14</td>
</tr>
<tr>
<td>T'd Transient Time Constant</td>
<td>0.164</td>
<td>0.16351</td>
<td>99.95</td>
</tr>
<tr>
<td>T&quot;d Sub-Transient Time Constant</td>
<td>0.0076</td>
<td>0.010014</td>
<td>99.76</td>
</tr>
<tr>
<td>Ta armature time constant</td>
<td>0.037</td>
<td>0.030489</td>
<td>99.35</td>
</tr>
</tbody>
</table>
Future of Modelling

- Tool for everyone
- Lays foundation for digital twin
- Implement multi discipline Genset
- Extracting variables from FEA
- Create project library
- Training
How Not to Model Complex Systems

- Model the full system
- Build subsystems individually
- Finally, connect the subsystems to complete the model
- Run
- Oops…. What went wrong?
How to Model Complex Systems

- Understand science behind your system
  - Inputs and outputs
  - Expected performance of system
  - Physical boundaries of model

- Breakdown your system

- Understand its safety features

- Customise the model
Subsystems

- Identify correct result for a subsystems
- Don’t re-invent the wheel
- Define the inputs and outputs
- Consider how to set the initial states
- Testing for physical boundaries
- Test the subsystem
Testing Subsystems

- Understand expected result
- Test the subsystem using real and validated test data
- Input incorrect variables
- Select suitable solver
  - Start with variable-step
  - Consider if appropriate to move to fixed-step
- Update on live document
  - Keep track of model updates
  - Easy for others to take on use of the model and understand the modeling process

- **ode45**
  - For linear electrical models
- **ode15s**
  - For nonlinear electrical numerically stiff models
  - Simulation inefficiently
- **ode23tb**
  - For nonlinear electrical models
  - Improves Simulation performance
Summary of Model-Based Design Process

- Connect all the subsystems
- Have you checked:
  - Physical boundaries
  - Initial conditions - Machine initialisation and load flow analysis
- Do not ignore any warnings or error
- Save the initialising states
Conclusion

- Science behind your system
- Difficult problem to solve
- Modelling and Simulation benefits on projects
- Commissioning time reduced
- Saved cost
Thank You