Modellierung physikalischer Systeme in der Lehre – Technologische Ansätze und deren Didaktik

Dr. Mischa Kim, Sebastian Malack

MATLAB EXPO 2015
DEUTSCHLAND
How do you model this system?
Presentation roadmap

- Traditional modeling: MATLAB, Simulink
- PhysMod modeling: SimMechanics
- Face-off 1: Model complexity
- Face-off 2: Visualization, animation
- Summary and conclusions
Modeling approach 1: Traditional MATLAB and Simulink
System modeling (using pen and paper)
Modeling process with MATLAB: the pen and paper approach

**Kinematics**
\[ \dot{q}_i = g(q_j, \dot{q}_j), \quad \dot{\theta}_i = h(q_j, \dot{q}_j) \]

**Lagrangian**
\[ \mathcal{L} = T - V \]

**Euler-Lagrange equation**
\[ \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} + \frac{\partial R}{\partial \dot{q}_i} = Q_i \]

**Differential equations**
\[ \dot{X} = f(X, u) \]

**Deployment**

**Application**

**Analysis**

**Animation**

Euler-Lagrange tool on File Exchange
Modeling approach 2: PhysMod
SimMechanics
Modeling process with SimMechanics

SimMechanics models on File Exchange

Deployment

Application

Analysis

Animation
Face-off #1: Model complexity
Modeling process with MATLAB

**Kinematics**

\[ \ddot{r}_i = g(q_j, \dot{q}_j), \quad \dot{\theta}_i = h(q_j, \dot{q}_j) \]

**Lagrangian**

\[ \mathcal{L} = T - V \]

**Euler-Lagrange equation**

\[ \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} + \frac{\partial R}{\partial \dot{q}_i} = Q_i \]

**Differential equations**

\[ \ddot{X} = f(X, u) \]
Face-off #2: Visualization, animation
Animation is Verification (and instantaneous feedback)
Summary, conclusions
## Modeling approach comparison

<table>
<thead>
<tr>
<th>Philosophy</th>
<th>MATLAB, Simulink</th>
<th>SimMechanics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>white-box: open</td>
<td>black-box: closed</td>
</tr>
<tr>
<td>Focus (teaching)</td>
<td>modeling concepts</td>
<td>applications (e.g. Controls)</td>
</tr>
<tr>
<td>Animation</td>
<td>Simulink 3D Animation</td>
<td>built-in</td>
</tr>
<tr>
<td>Tool ramp-up</td>
<td>steep</td>
<td>moderate</td>
</tr>
</tbody>
</table>
Typical engineering BA curriculum

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physics</td>
<td>Math 1</td>
<td>Math 2</td>
<td>Math 3</td>
<td></td>
</tr>
<tr>
<td>Programming</td>
<td>Statics</td>
<td>Dynamics</td>
<td>Controls</td>
<td>Project/design</td>
</tr>
<tr>
<td>Intro to engineering</td>
<td>EE 1</td>
<td>EE 2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Traditional**
- **PhysMod**
Resources: fetch the models and get started

1. Go to: MATLAB Central, File Exchange
2. Search for
   - Rotary pendulum bundle [contains all presentation code]
   - Euler-Lagrange tool [derive differential equations]

The presentation will be made available post-event.