Embedded Systems 2010/2011
Harmonic Oscillator & MATLAB Tutorial

Hans-Jörg Peter
Reactive Systems Group
http://react.cs.uni-saarland.de

October 26th, 2010
Assignments and Tutorials

Assignments
- Handout / return: Tuesday, before the lecture
- Teams are allowed (at most 3 students per team)
- Box is available

First assignment
- Handout: Now, also available online
- Return: Tuesday, 2\textsuperscript{nd} November 2010 (in one week)

Tutorials
- Will start on Wednesday, 3\textsuperscript{rd} November
- Submit and come to any tutorial you like
- But be prepared to change
Course Registration

- Do the assignments
- Come to the tutorials
- Subscribe to the mailing list
- HISPOS registration only for the midterm exam needed
Produced by Mathworks
Used for simulation and numerical computation
No (Maple-like) symbolical solving
Industrial standard tool for developing embedded systems
MATLAB Structure

- MATLAB core: IDE for the MATLAB language
- Simulink: Graphical environment for continuous simulation
- Stateflow: Statecharts for Simulink
- Many other add-ons available...
Some problems do not have a closed-form solution
Approximative numerical solutions often suffice
Simulation of the physical world
Starting MATLAB

1. http://sunray1.studcs.uni-sb.de
2. Log in
3. Click on MatLab

Alternatively:

1. Log onto a cip workstation
2. Execute /installer/arch/bin/matlab
MATLAB IDE

1. Current directory
2. Directory explorer
3. Workspace
4. Command history
5. Command window
The MATLAB Language

- Simplified C-like syntax
- Case sensitive
- Interactive shell: command window
- User defined functions: m-files
- Many built-in commands:
  - `lookfor <keyword>`
  - `help <function>`
  - `sprintf (<format str>, v1, v2, ...)`
  - `disp (<object>)`
  - `plot (Y)`
  - `plot (X, Y)`
  - `...`
Variables

- Each numerical variable is a matrix
- Scalars = $1 \times 1$ matrices
- No explicit declarations / dynamic typing
- Polymorphism
- Removing variables:
  - `clear <variable>`
  - `clear`
Working with Matrices

- $a = 4$
- $b = \begin{bmatrix} 4 & 8 & 15 \\ 16 & 23 & 42 \\ 1 & 2 & 3 \end{bmatrix}$
- $c = b'$
- $d = \text{ones}(4)$
- $e = \text{eye}(3)$
- $f = b\times b$
- $g = b.*b$
- $h = 0:10$
- $i = 0:0.01:2*\pi$
Script Files

- So called m-files
- Must be located in
  - the current directory or
  - the global search path
- Can be executed from the command window
- Can also define functions
Control Structures

- **Conditional**
  
  ```
  if <cond> 
  <statements> 
  [else
  <statements>] 
  end
  ```

- **While loop**
  
  ```
  while <cond> 
  <statements> 
  end
  ```

- **For loop**
  
  ```
  for v = <from>: [<step>]: <to> 
  <statements> 
  end
  ```
Example: Computing $\pi$

- Monte Carlo method for computing $\pi$

\[
\frac{\text{points inside}}{\text{points total}} \approx \frac{\pi}{4}
\]
Simulink
Hooke’s Law: $F = -ky$

- $F$: restoring force
- $k$: positive constant that characterizes the oscillator
- $y$: amplitude or displacement
Harmonic Oscillator (2)

- $m$: mass constant
- $k$: spring constant
- $y_0$: initial displacement
- $y$: current displacement
- $v = \dot{y}$: current velocity
- $a = \dot{v} = \ddot{y}$: current acceleration

$$F = ma = -ky$$

$$\Leftrightarrow \quad ma + ky = 0$$

$$\Leftrightarrow \quad m\ddot{y} + ky = 0$$

$$\Leftrightarrow \quad m\dot{v} + ky = 0$$
Harmonic Oscillator in Simulink
Damped Harmonic Oscillator

- \( m \): mass constant
- \( R \): damper constant
- \( k \): spring constant
- \( y_0 \): initial displacement
- \( y \): current displacement
- \( v = \dot{y} \): current velocity
- \( a = \ddot{v} = \ddot{y} \): current acceleration

\[
\begin{align*}
    m\ddot{y} + R\dot{y} + ky &= 0 \\
    \iff m\dot{v} + Rv + ky &= 0
\end{align*}
\]
Damped Harmonic Oscillator in Simulink