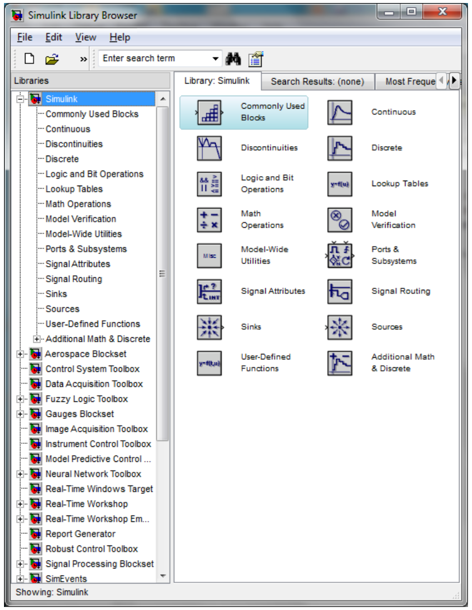
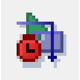
# Simulation Analysis of Nonlinear Systems



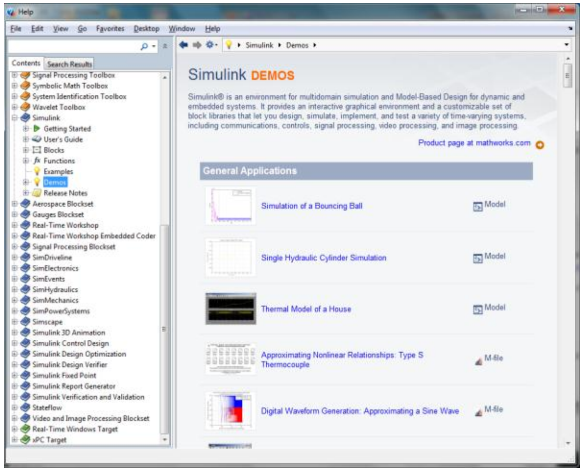

## An introduction to Simulink

### To start Simulink

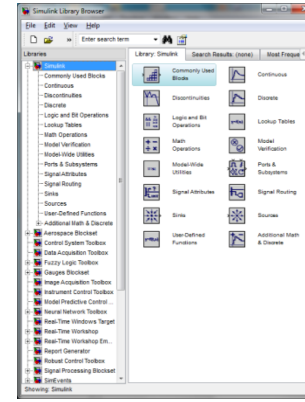
- Start Matlab
  - Click on the Simulink icon , or
  - Type “simulink” in Matlab workspace

## SIMULINK online tutorials

- Interactive video tutorials and recorded webinars are at MathWorks website: http://www.mathworks.com/academia/student_center/tutorials/
  - Registration is required, but is free
- Demo models are available in MATLAB help:
  - Help → Simulink → Demos

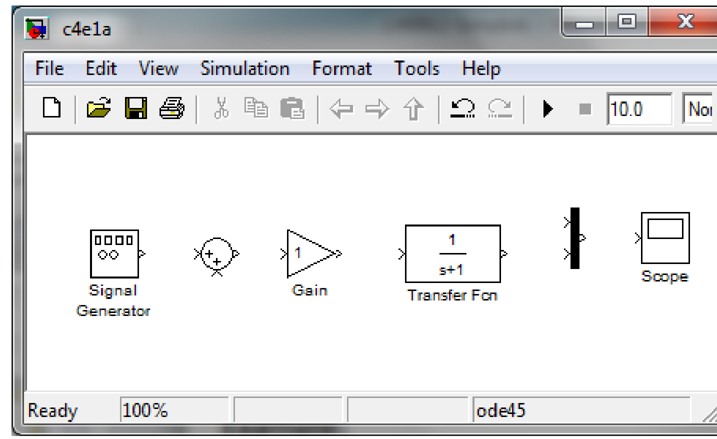


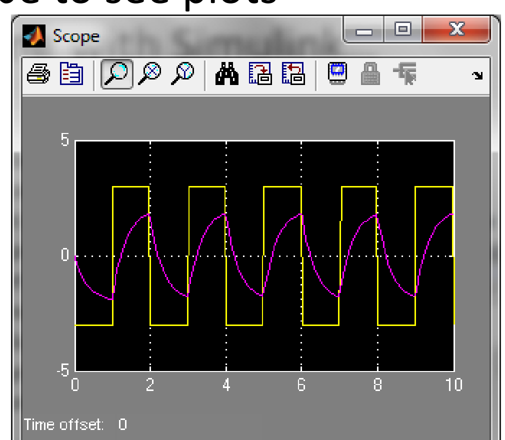
# Modeling with Simulink…

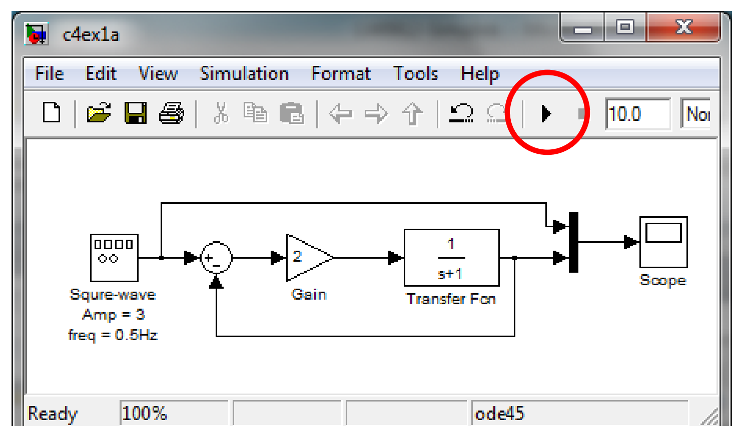


## <u>Example-1:</u>

- Start Simulink
- Select File → New → Model (to open a new model window)
    - Or, click on the 'New Model' icon
- Drag and drop blocks from the Library Browser in the new model window
- Connect the blocks appropriately
- Double-click blocks and set their parameters
- Select simulation solver and set parameters
    - Simulation > Configuration Parameters…
- Start simulation and see results on scope
    - Click 'Start Simulation' icon or press 'Ctrl+T'
    - Double-click on scope to see plots



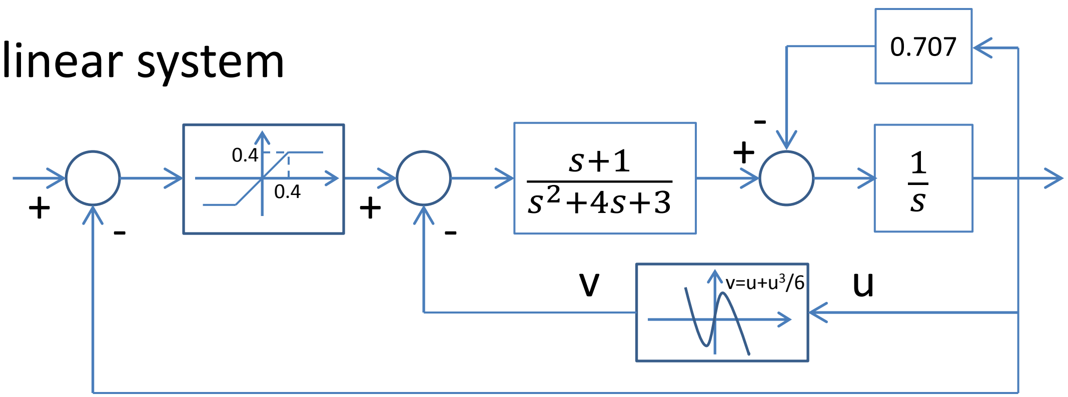



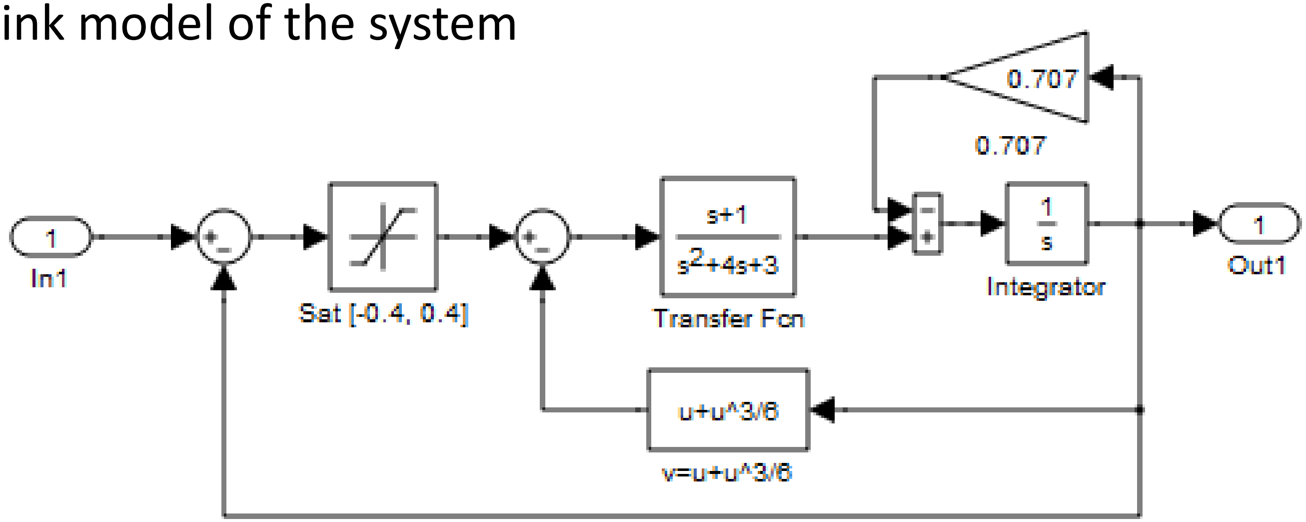
# Modeling Nonlinear Systems

## Example-2:

- Block diagram of a nonlinear system




- Equivalent Simulink model of the system

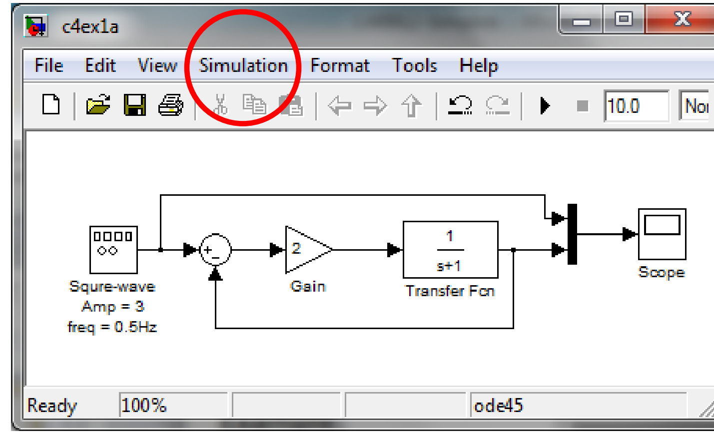


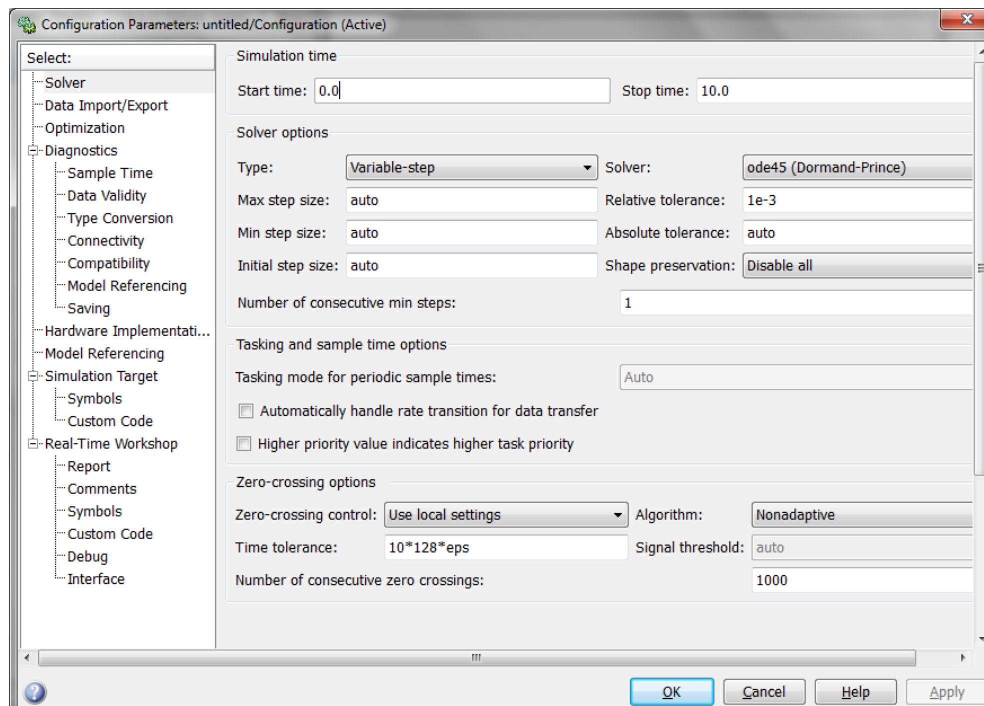
# Specifying Simulation Algorithm and Solver Parameters

## Example-2 continued:

Click on the “Simulation” tab

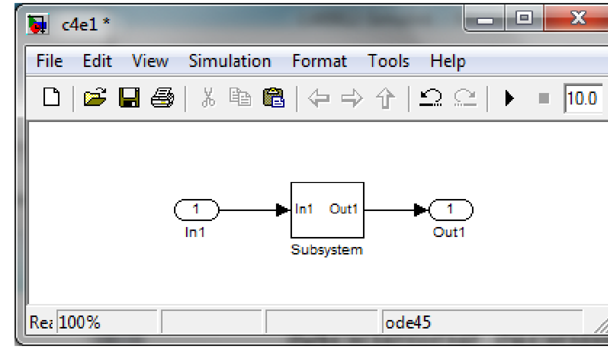
- In the “Configuration Parameters” window specify
  - Start and stop times
  - Solver algorithm and options
    - Variable or fixed step
    - Ode45 (Dormand-Prince)
    - Ode25s (stiff/NDF)
  - Simulation accuracy
    - Absolute and relative tolerance
  - Minimum and maximum step size
  - Warning and error messages



# Simulating Nonlinear Systems

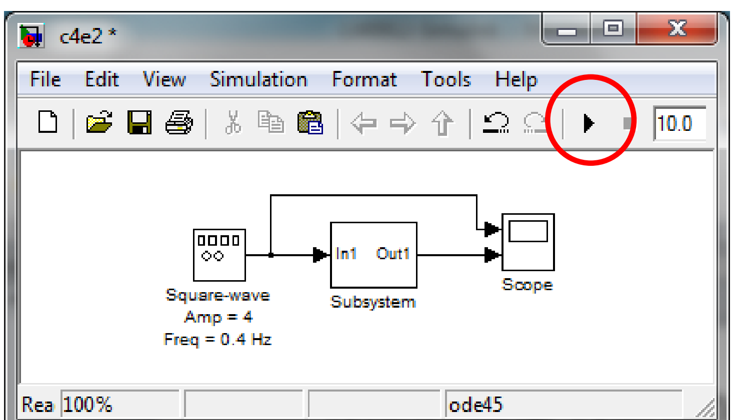
## Example-2 continued

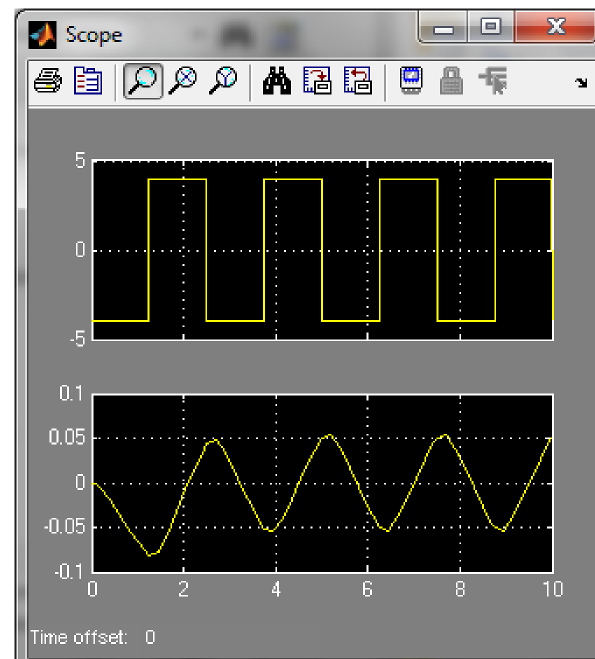



- Create subsystem
  - Highlight system blocks
  - Select: Edit > Create Subsystem, or press ‘Ctrl+G’
- Add ‘Signal Generator’ and ‘Scope’ (double-axis)
  - Add a ‘To Workspace’ block to export signals to Matlab workspace
    - The exported vector is named ‘simout’ (can be re-named)
- Start simulation (▶) and observe results on scope
  - You can also start simulation using sim() function
    - [t,x,y]=sim(model_name,tf,options)

      where options can be set as:

      options=simset(property1,parameter1,property2,parameter2,…)

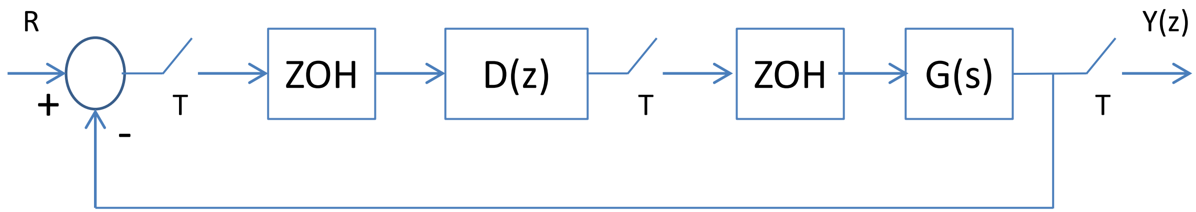


# Modeling Hybrid Systems

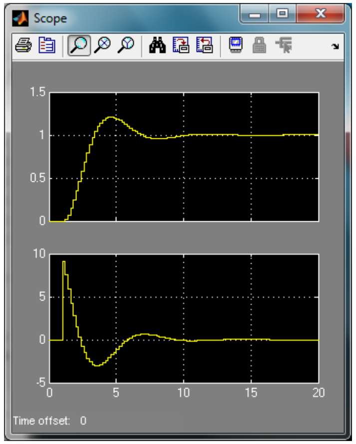
## Example-3:

- Block diagram of a hybrid (continuous + discrete) system

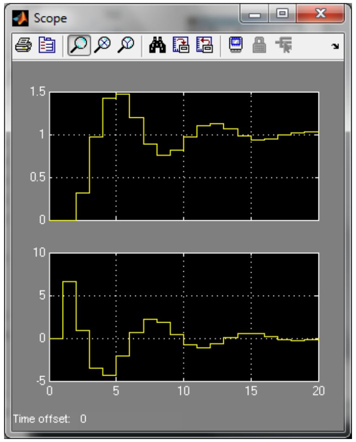


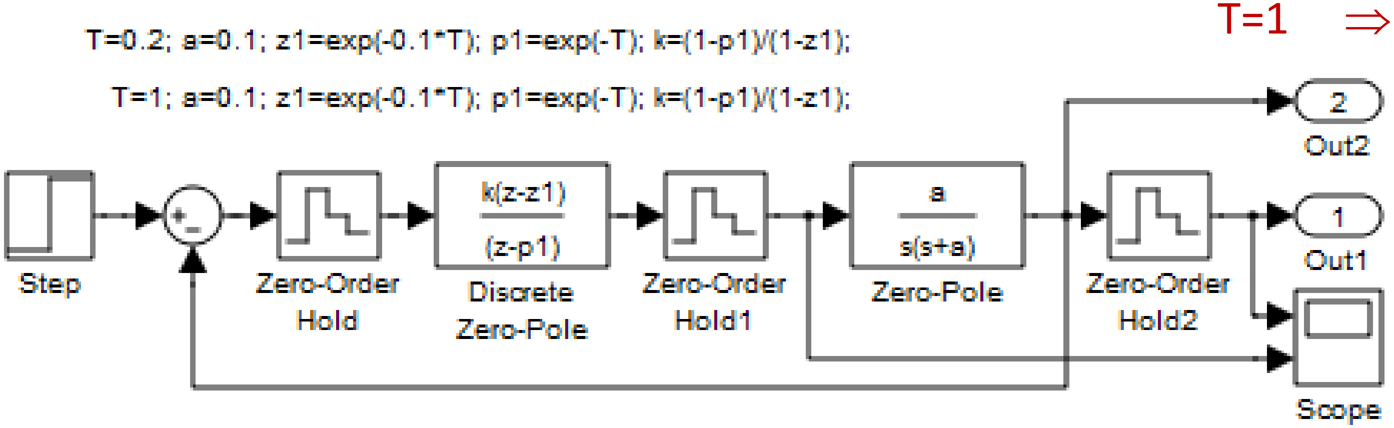

- Equivalent Simulink model of the system

T=0.2 ⇒



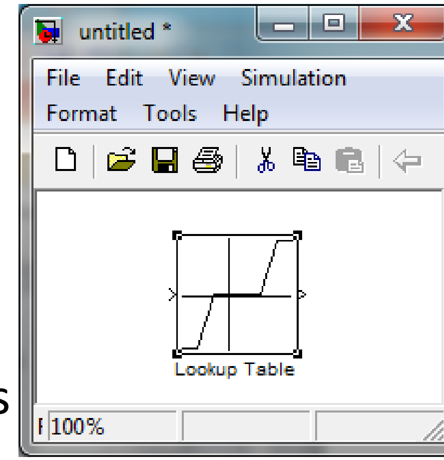

T=1 ⇒

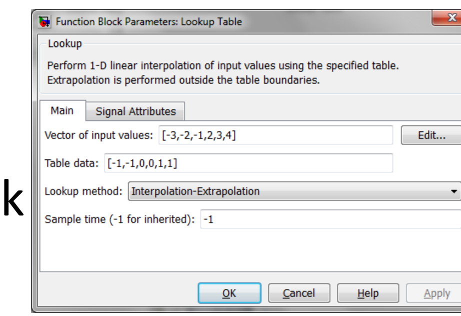


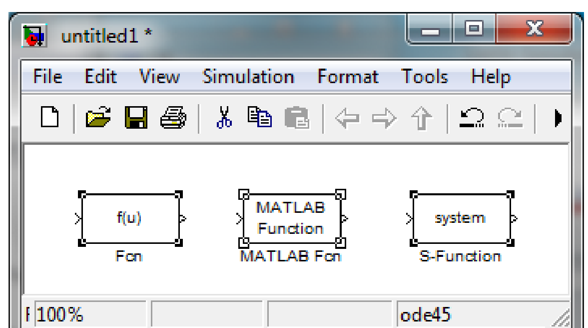



# Nonlinear Elements Modeling

- Modeling piecewise linear nonlinearities
  - The one-dimensional “Look-up Table” block can be used to represent piecewise-linear nonlinearities
  - Other smooth nonlinearities can also be represented
    - For example: y=tanh(x)
  - Also “2-D Look-up Table” and “n-D Look-up Table” blocks
- User-defined functions
  - “Fcn” block can define mathematical operations
  - “Matlab Fcn” block embeds Matlab functions in Simulink
  - “S-Function” is to embed “C” programs in Simulink



# Linearization of Nonlinear Models

## Linearization of nonlinear systems at an equilibrium point

### Equilibrium point of a system

- Consider the system: $\begin{cases} \dot{x} = f(x,u,t) \\ y = h(x,u,t) \end{cases}$

  where

  $x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$, $u = \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix}$, $y = \begin{bmatrix} y_1 \\ \vdots \\ y_p \end{bmatrix}$, $f(x,u,t) = \begin{bmatrix} f_1(x_1 \dots x_n, u_1 \dots u_m, t) \\ \vdots \\ f_n(x_1 \dots x_n, u_1 \dots u_m, t) \end{bmatrix}$, and $h(x,u,t) = \begin{bmatrix} h_1(x_1 \dots x_n, u_1 \dots u_m, t) \\ \vdots \\ h_p(x_1 \dots x_n, u_1 \dots u_m, t) \end{bmatrix}$

- An equilibrium-point is a point $(x_e, u_e)$ where $\dot{x} = 0$
- To find the equilibrium-points / operating-points (OP) of a system:
  - Solve $f(x,u,t) = 0$ for $(x,u)$

  In **Matlab** use the command 'trim()'

  Syntax: [x,u,y,z]=trim(model_name,x0,u0)

  - x0,u0 are the initial guess
  - x,u,y are the returned operating points
  - z is the returned value of dx/dt at the estimated OP

# Linearization of Nonlinear Systems

Linearizing system model at an OP $(x_e, u_e)$

- Expand $f(x,u,t)$ and $h(x,u,t)$ about $(x_e, u_e)$
  - Use Taylor series expansion
  - For time-invariant systems, t is not considered

$$f(x,u) = \underbrace{f(x_e,u_e)}_{0} + \left.\frac{\partial f}{\partial x}\right|_{\substack{x=x_e\\u=u_e}}(x-x_e) + \left.\frac{\partial f}{\partial u}\right|_{\substack{x=x_e\\u=u_e}}(u-u_e) + \underbrace{\cdots}_{h.o.t.}$$

$$h(x,u) = \underbrace{h(x_e,u_e)}_{0} + \left.\frac{\partial h}{\partial x}\right|_{\substack{x=x_e\\u=u_e}}(x-x_e) + \left.\frac{\partial h}{\partial u}\right|_{\substack{x=x_e\\u=u_e}}(u-u_e) + \underbrace{\cdots}_{h.o.t.}$$

Let $\tilde{x} = x - x_e$, $\tilde{u} = u - u_e$, and $\dot{\tilde{x}} = \dot{x} - \underbrace{\dot{x}_e}_{0}$. Then, neglecting the h.o.t.'s, we get:

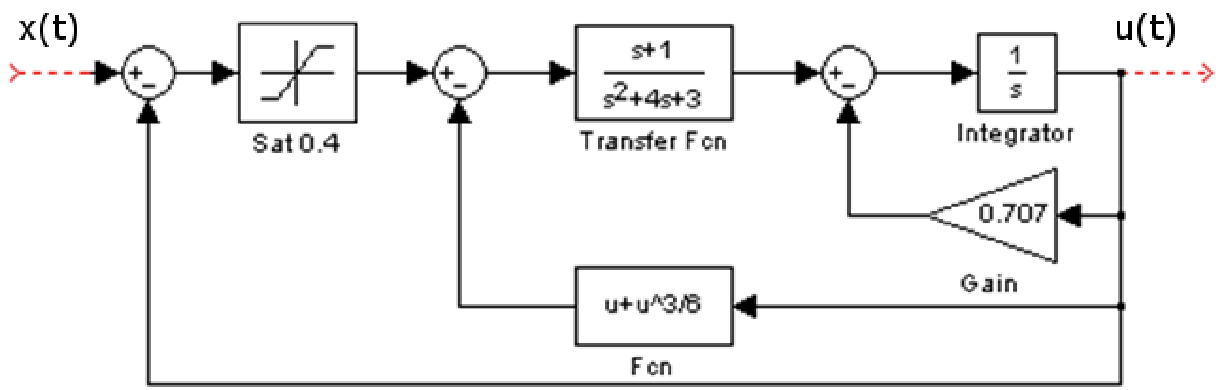
$$\begin{cases} \dot{\tilde{x}} = \underbrace{\left(\left.\frac{\partial f}{\partial x}\right|_{\substack{x=x_e\\u=u_e}}\right)}_{A}\tilde{x} + \underbrace{\left(\left.\frac{\partial f}{\partial u}\right|_{\substack{x=x_e\\u=u_e}}\right)}_{B}\tilde{u} \\ y = \underbrace{\left(\left.\frac{\partial h}{\partial x}\right|_{\substack{x=x_e\\u=u_e}}\right)}_{C}\tilde{x} + \underbrace{\left(\left.\frac{\partial h}{\partial u}\right|_{\substack{x=x_e\\u=u_e}}\right)}_{D}\tilde{u} \end{cases} \Rightarrow \begin{cases} \dot{\tilde{x}} = A\,\tilde{x} + B\,\tilde{u} \\ y = C\,\tilde{x} + D\,\tilde{u} \end{cases}$$

- In **MATLAB**, use the command 'linmod()'

# Linear Approximation with Matlab/Simulink

## Example

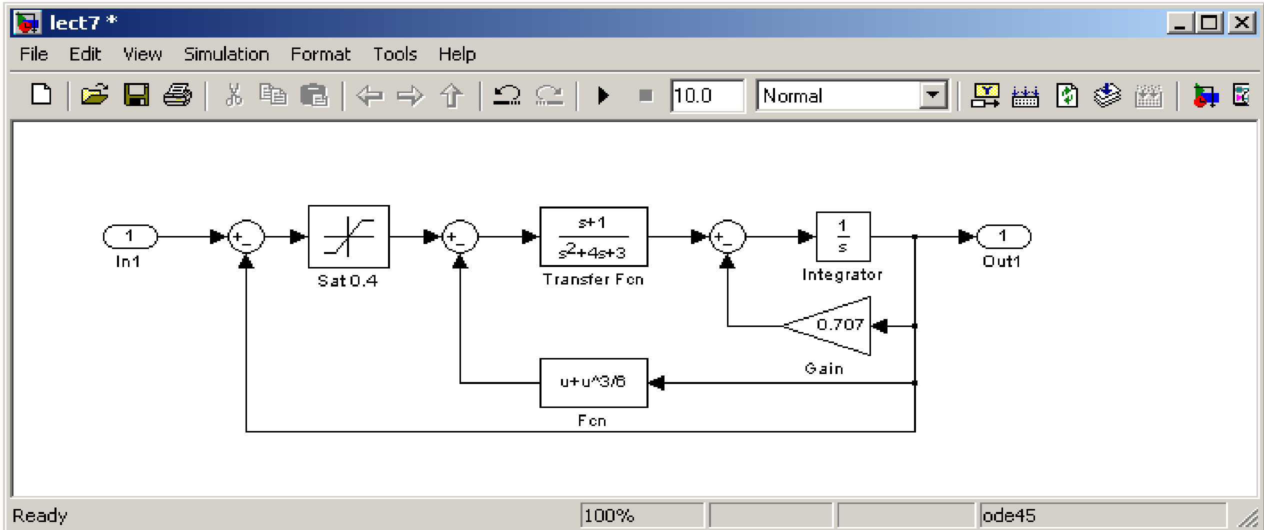
Consider the nonlinear system shown (chapter 4, example 1)




How can we find the transfer function of this system?

- A transfer function exists only for LTI systems, so we must find a linear approximation of this model

# System Linearization of Simulink Models



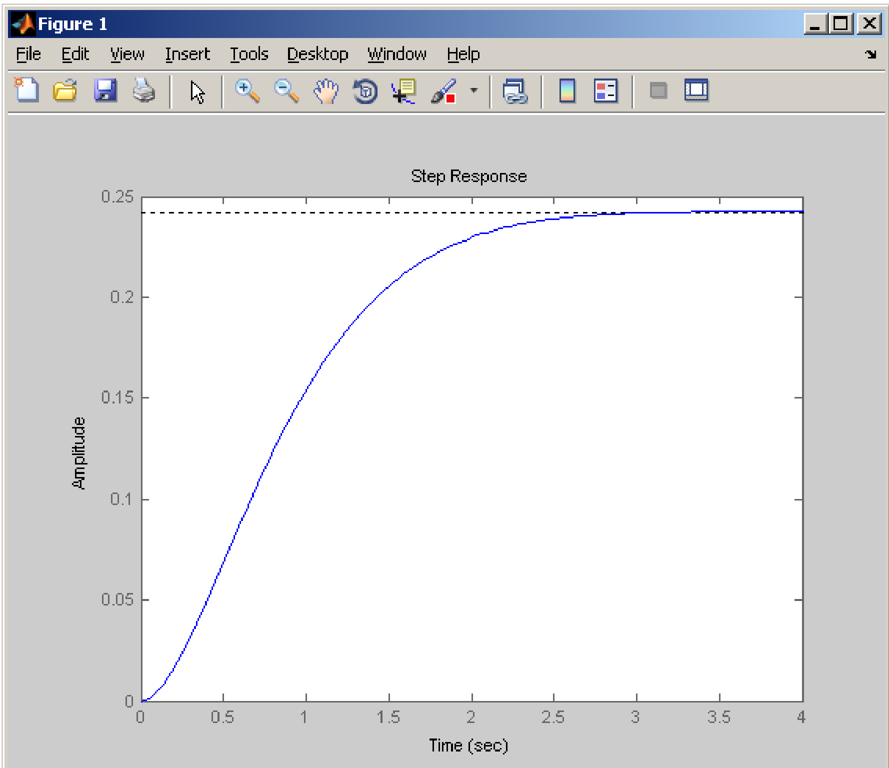

Given a model in Simulink

- Use Matlab commands trim and linmod to find the operating point (OP) and linearize the system model at that OP
- The linearized model can then be manipulated and analyzed in MATLAB

## MATLAB code:

```
[xe,ue,ye,dxe]=trim('lect7',[],1);
[A,B,C,D]=linmod('lect7',xe,ue);
sys=ss(A,B,C,D);
G=tf(sys);
stepplot(G);
```

# System Linearization …




- From the step response, it appears that the system is stable
- This could also be verified using isstable() command or by checking if the real parts of all its eigenvalues are negative.

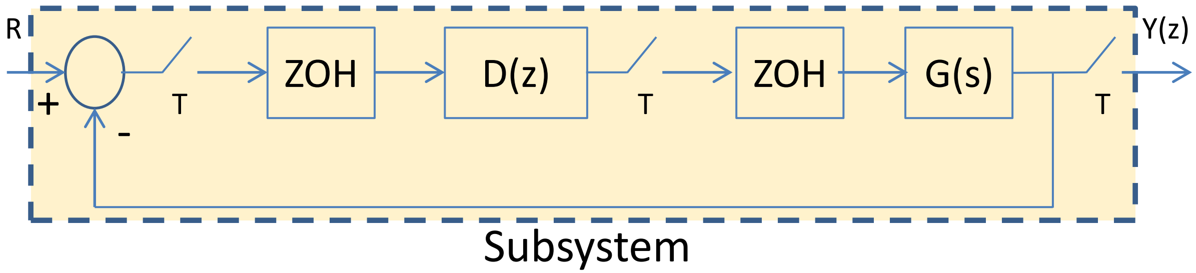
```
>> isstable(sys)
ans =
     1

>> eig(sys)
ans =
  -1.8535 + 0.8329i
  -1.8535 - 0.8329i
  -1.0000
```

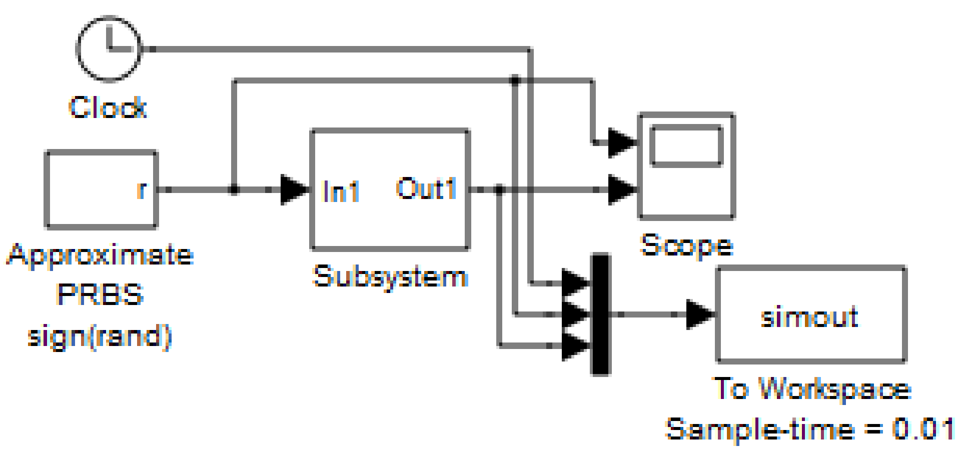
# Linear Identification of Nonlinear Systems

Example (same as example 3):

- Block diagram of the system



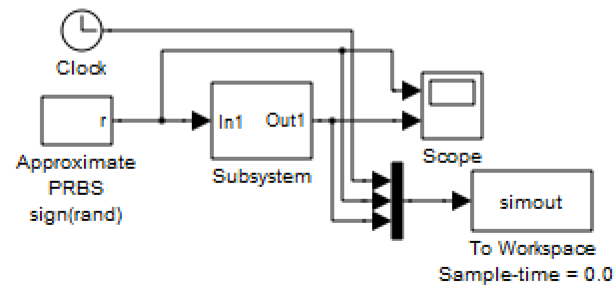

- Add other blocks for input and output collection

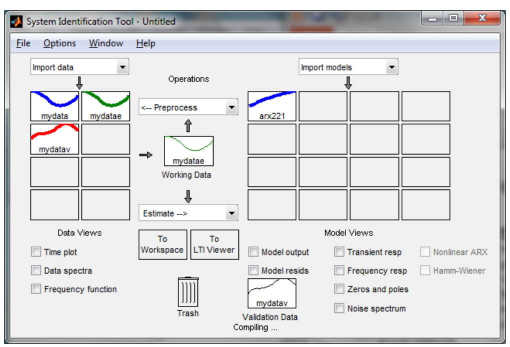


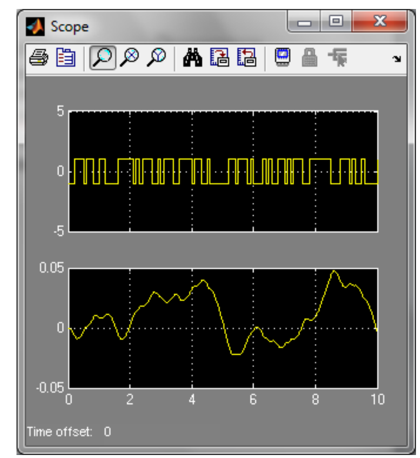
# Linear Identification of Nonlinear Systems

## Example continued

- Simulate system with a 'sufficiently rich' input
- Export sampled I/O data to Matlab workspace
  - Use 'To Workspace' block and set its sample-time, T=0.01
- In Matlab, from simout, construct uk and yk
- Use 'ident' to find a linear model of the system
- Verify the accuracy of the approximate linear model




**Identified linear model:**

$$G(s) \cong \frac{0.4}{s^2+3.707s+3.121}$$

**Verify linear model accuracy:**

```
tk=simout(:,1);
uk=simout(:,2);
yk=simout(:,3);
% Run ident and find
% System model arx221
G221=tf(arx221);
Ghz=G221(1);
Ghs=d2c(Ghz)
yhk=lsim(Ghs,uk,tk);
plot(tk,yk,'r'), hold on,
plot(tk,yhk,'bo'); hold off,
```