Modeling and Simulation of using Matlab/Simulink

Modeling Building and Simulation with Simulink

Updated for Spring 2001

Raul G. Longoria

University of Texas at Austin

Department of Mechanical Engineering

Outline for Today

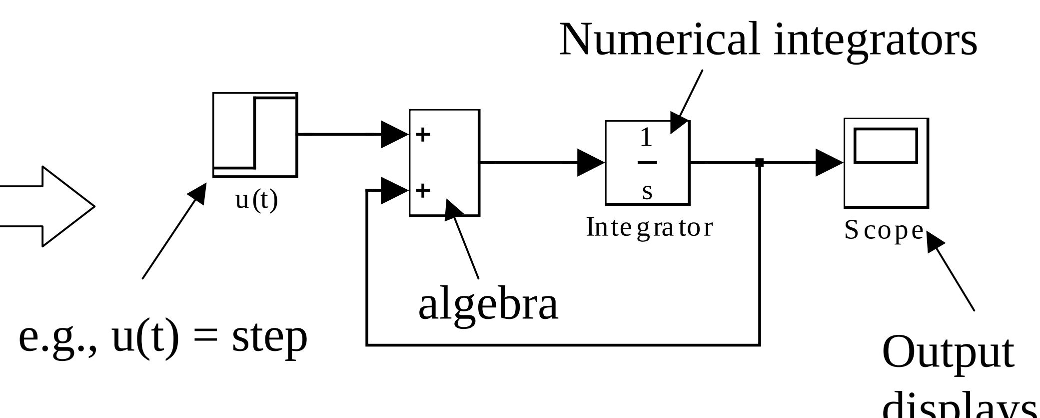
- Quick Intro/Demo
- Physical modeling with block diagrams
- Simulink modeling tools and features
- Analysis Tools in Simulink
  - review available solvers
  - trim
- Tutorials:
  - PMDC motor
  - Angular drive coil

Goal: Feel comfortable building and analyzing basic models in Simulink.

Description of Simulink

- Simulink is a software package that runs within MATLAB, and provides a graphical user interface for building and analyzing system models.
- Simulink can communicate with the MATLAB workspace and functions, as well as with user-written programs in other languages.
- Simulink can be used for modeling, simulating, and analyzing dynamic systems, and has features that can be used to represent linear and nonlinear systems with continuous and/or sampled time characteristics. Systems can also be multirate, i.e., have different parts that are sampled or updated at different rates.
- Simulink uses a block diagram approach to represent mathematical equations.

$$\dot{x} = -x + u(t)$$



Simulink Description

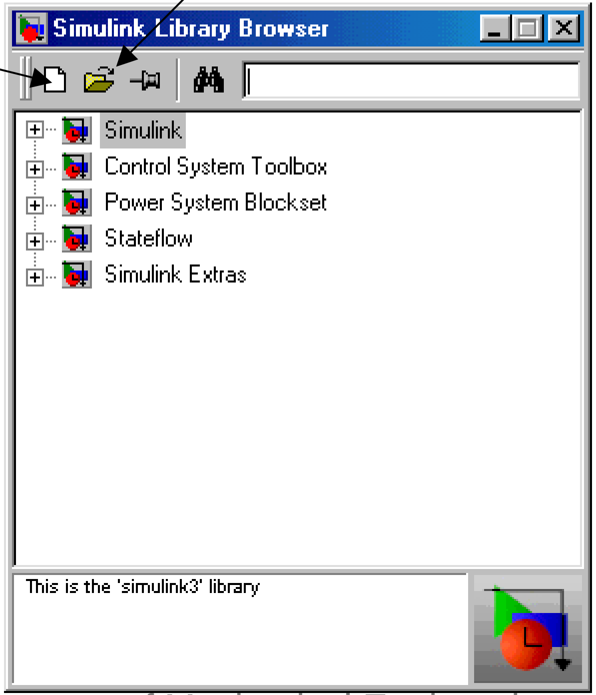
- Models are hierarchical. Model detail can be viewed by “diving into” blocks.
- The block diagram describes the equations, and possibly the interconnection of components.
- You can simulate the model choosing from several integration methods.
- Analysis can be conducted in Simulink or from the MATLAB command window. It depends on whether an interactive or batch mode is desired.
- Display functions are available for direct plotting (while simulation runs).
- The simulation results can also be “sent” to the workspace for postprocessing and visualization using the MATLAB graphical tools.
- There are many analysis tools available, and since Simulink is integrated into MATLAB, you can simulate, analyze, and revise models in either environment at any point.

Starting Simulink

- Simulink can be ‘launched’ from the command line:
  » simulink

Open a model

Create a new model



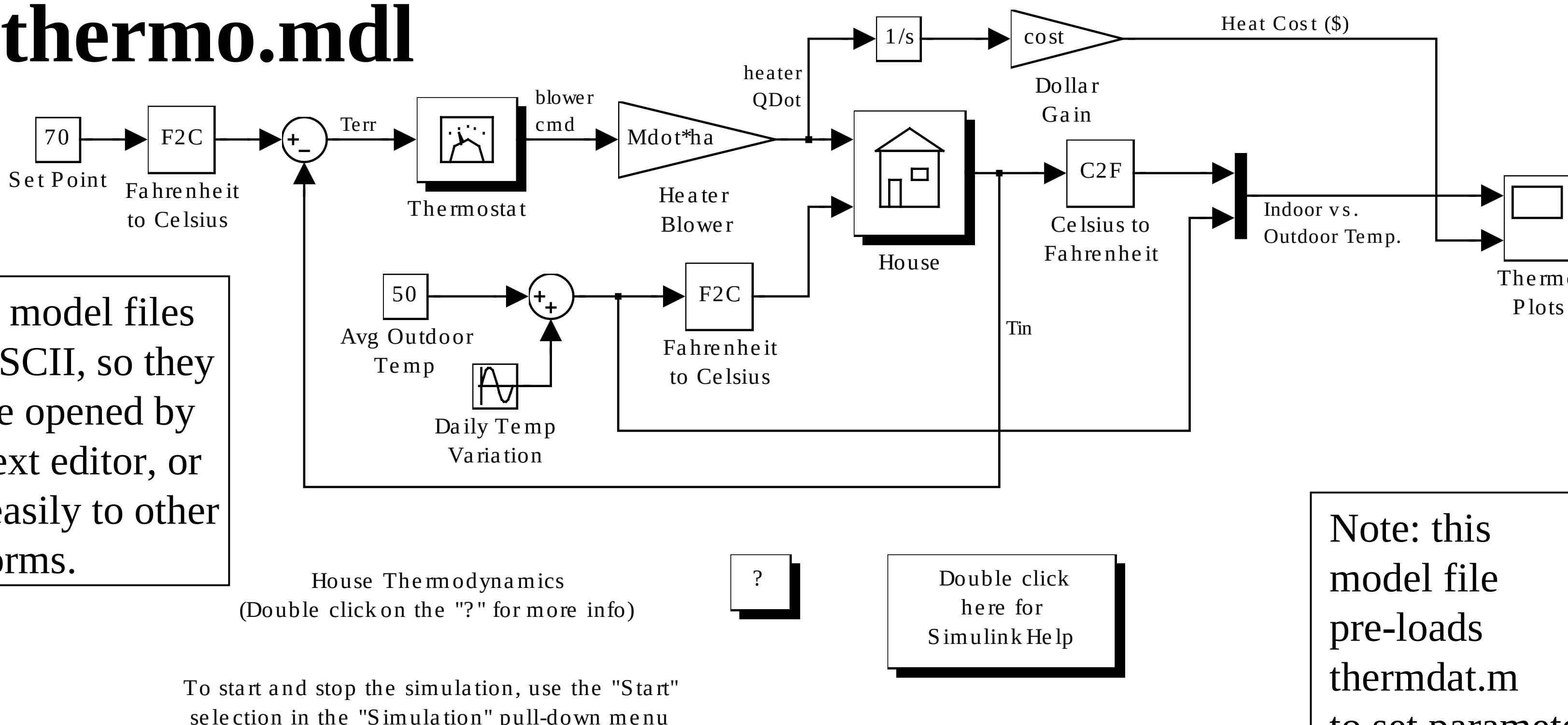
- Or, use the menu bar
- The model browser opens
- *.mdl is the extension used for Simulink models

House Demo

Open:

MATLABR11\toolbox\simulink\simdemos

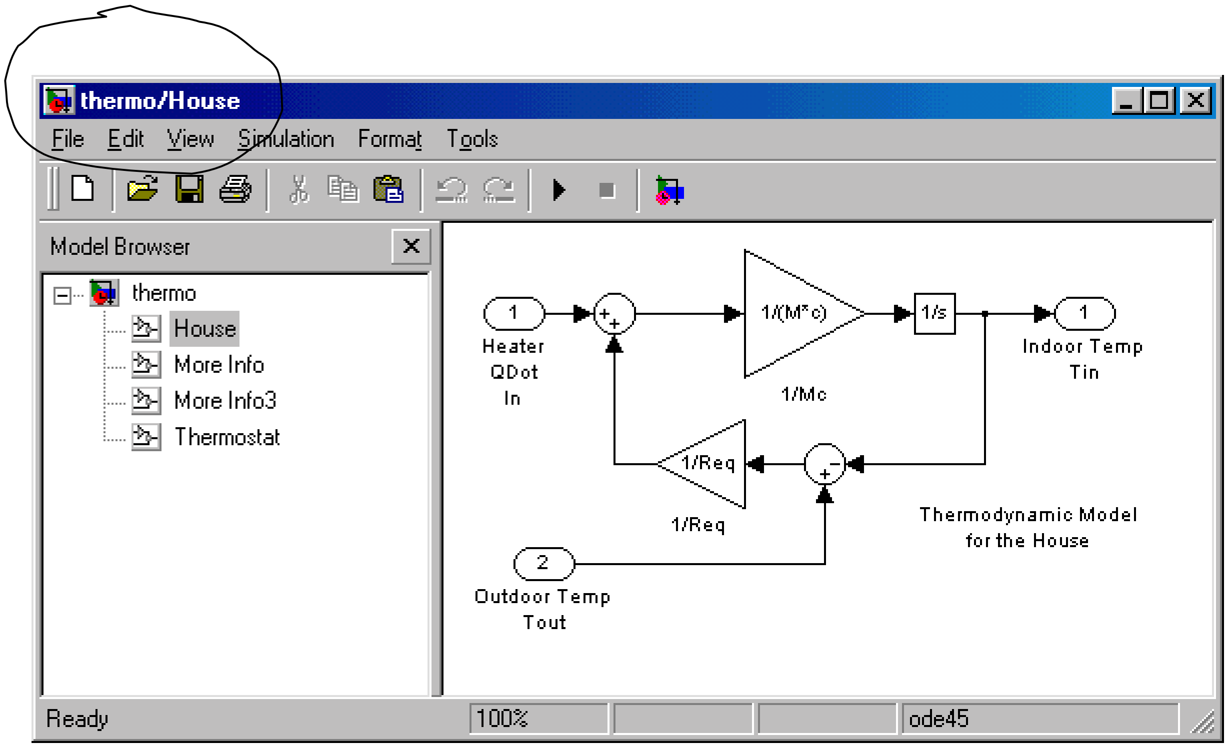
thermo.mdl



Note: model files are ASCII, so they can be opened by any text editor, or sent easily to other platforms.

Note: this model file pre-loads thermdat.m to set parameters.

What’s in the house?

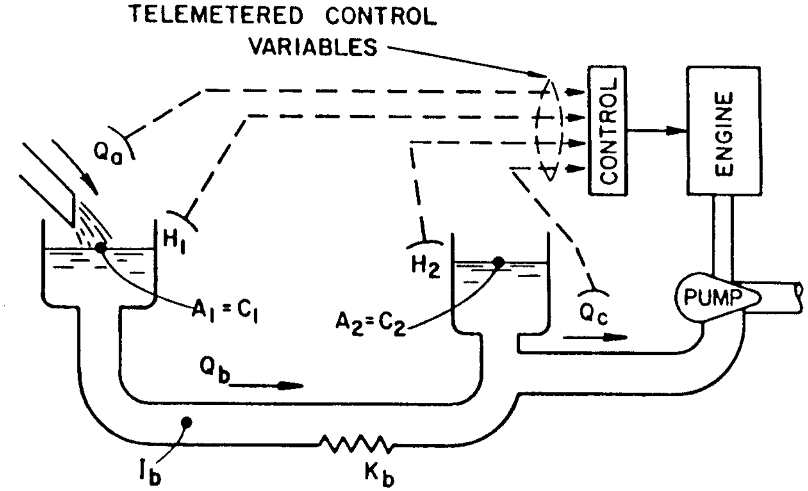


The models are formulated from a set of basic block diagram elements. These can be “grouped” into subsystems, building a multilevel model.

The Simulink environment provides a way to piece together basic elements with “algebraic” elements (e.g., summers), and there are “calculus” functions as well. The physics can either be diagrammed or embedded in function calls.

Other demos

Block Diagrams
for Computer Simulation:
Not a new idea.

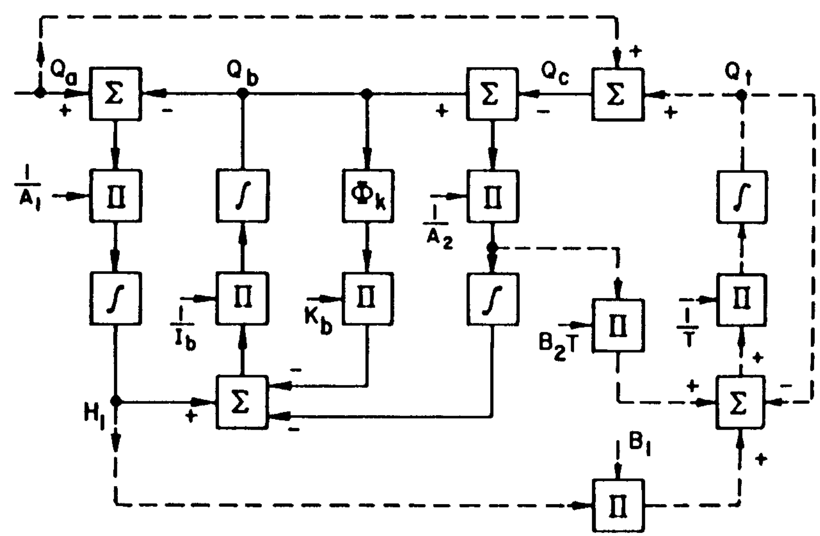


(a) SEWAGE TUNNEL SYSTEM

F.D. Ezekiel and H.M. Paynter, "Computer Representations of Engineering Systems Involving Fluid Transients," Trans. ASME, Nov. 1957.

The block diagram descriptions are effective for modeling dynamic system models.

Example elements:
Σ = summer
Π = product
∫ = integrator
Φ = function (nonlinear type)



(b) COMPUTER BLOCK DIAGRAM

What is to be gained?

- Quicker modeling
- Better communication
- Modularity - can connect subsystems together to make more complex models

Exploring the Simulink Environment

- To learn about the Simulink environment, we will explore the various blocks and features as we build several simple examples.

Transient Response

- Let’s build a simple mass-spring-damper or LRC circuit model responding to a step response.
- Display either:
  – velocity (for mechanical)
  – current (for LRC)
- Add nonlinear effects

Department of Mechanical Engineering
The University of Texas at Austin

Simulink Features to Explore

- Manipulating Blocks, Lines, and Labels
- Block Orientation & Sizing
- Changing Block Properties

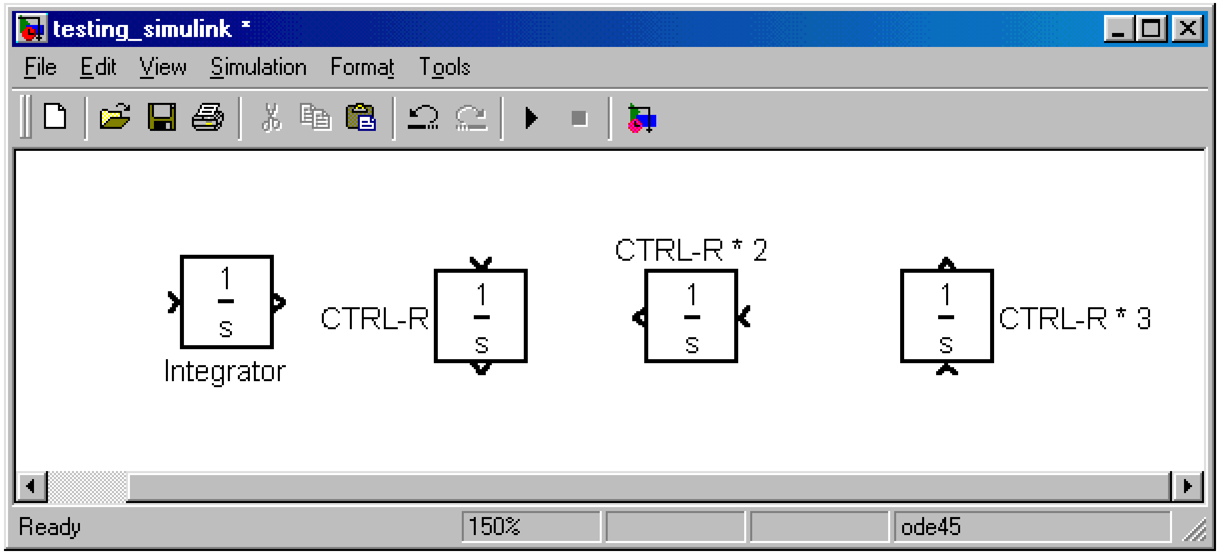
- Display blocks
- Output to workspace

Signal Lines

- Drawing lines
- Changing line directions
- Deleting lines
- Vector lines; displaying line widths
- Inserting blocks in a line
- Line labels (double-click on line)

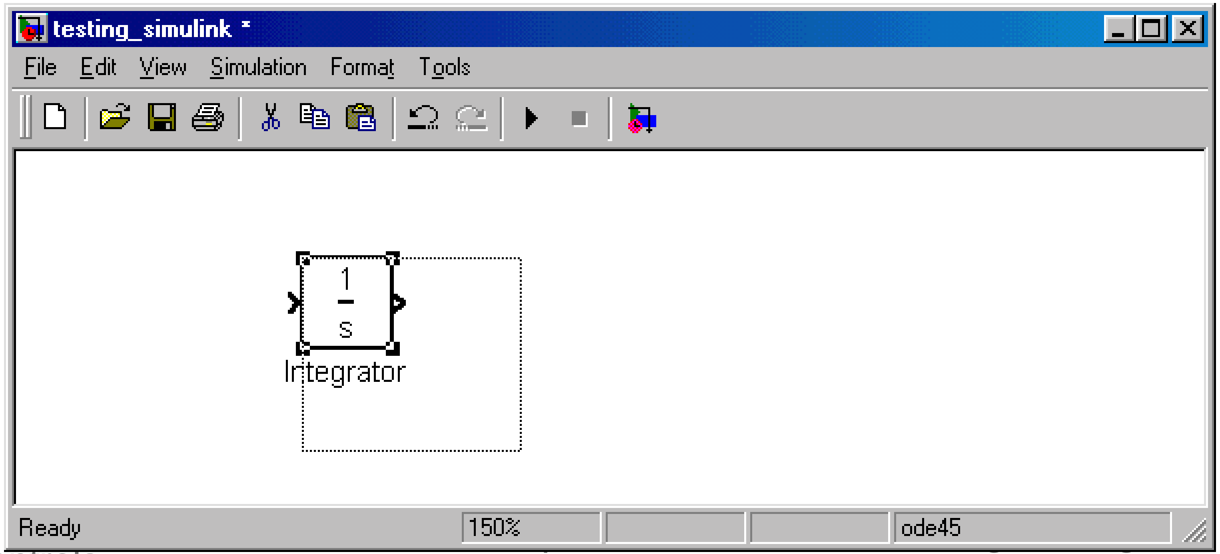
Block Orientation & Sizing

Use CTRL-R to rotate



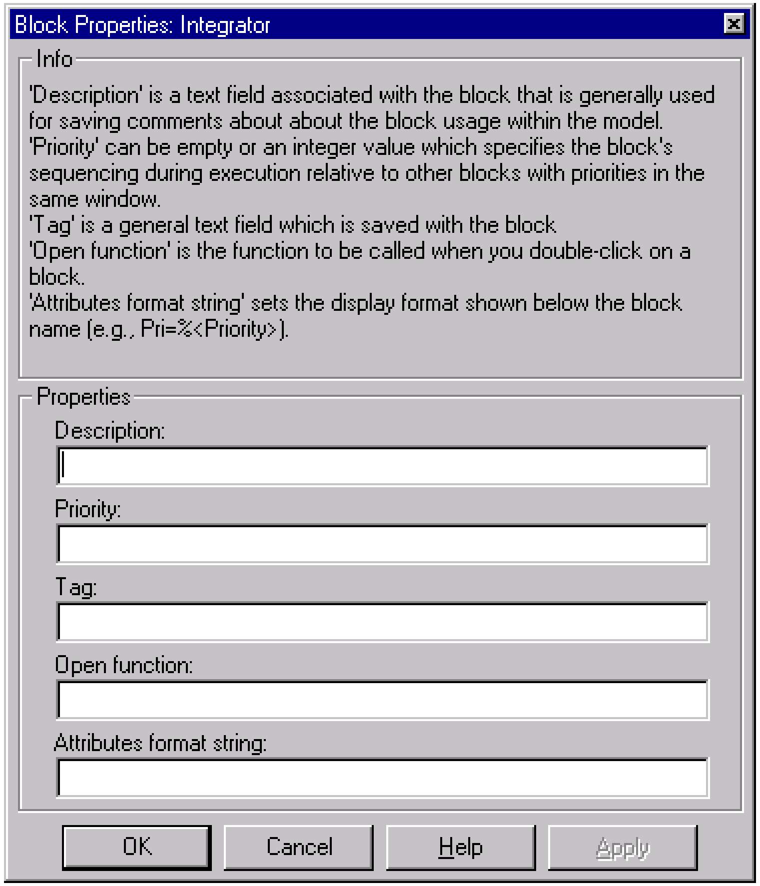
Grab corner and drag to size

(no UNDO!)



Block Properties

For any given block, for example, an integrator, open the Block Properties



Creating a Subsystem

- If a group of elements is to form a component, you can group them into a subsystem.
- Select components, go to Edit menu and select “Create Subsystem”.
- Careful, this is hard to UNDO.

Department of Mechanical Engineering
The University of Texas at Austin

Example: Using Subsystems

- **Example**: Create a subsystem for the mass-spring-damper or LRC system. Illustrate inputs and outputs.

- You can use subsystem as a method for loading parameter data.
  - Or Callback routines (PreLoadFcn)

Simulink Analysis

- Finding model information
  For example,
  » sizes = bicycle1([ ],[ ],[ ],0)
  no. of states (every integrator -> state), etc.
- Running Simulation: interactive vs batch
- Setting parameters (simset)
- Getting parameters (simget)

Parameters for Simulation

- Solver
- RelTol
- AbsTol
- Refine
- MaxStep
- InitialStep
- FixedStep
- OutputPoints

Solver
RelTol
AbsTol
Refine
MaxStep
InitialStep
FixedStep

Using Command Line

- You can use ‘sim’ to run a model from the command

» [t,x,y] = sim(modelname,tspan,options,ut)

» help sim

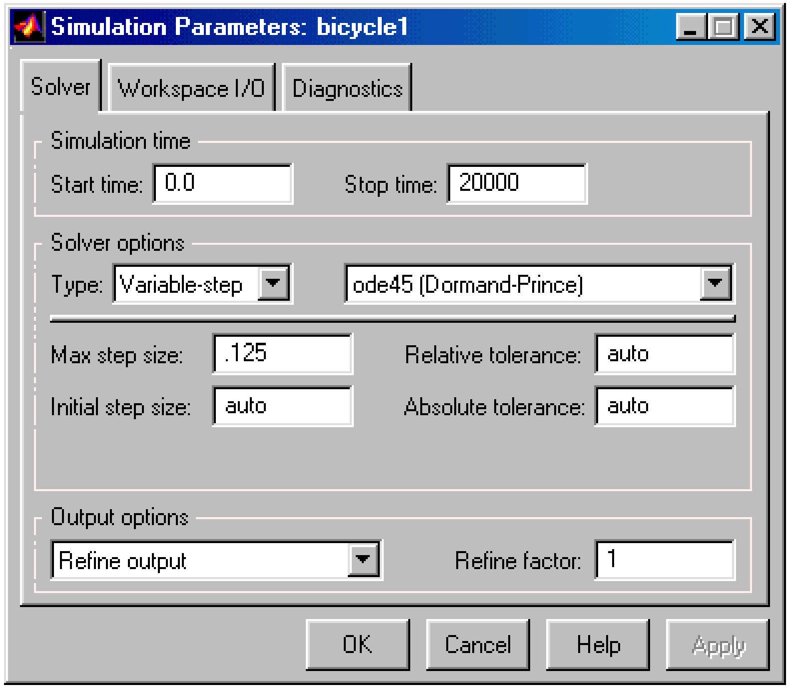
Use simset

SIM Simulate a Simulink model
SIM('model') will simulate your Simulink model using all simulation parameter dialog settings including Workspace I/O options.

[T,X,Y] = SIM('model',TIMESPAN,OPTIONS,UT)

Simulation Parameters

- Under 'Simulation' menu (3 panels)
  - Solver settings
  - Workspace I/O
  - Diagnostics



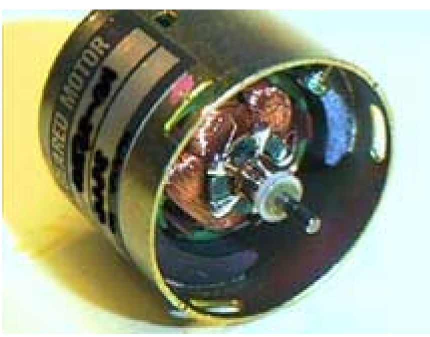
- simset can be used to set **options** from Command line

Department of Mechanical Engineering
The University of Texas at Austin

Model Building Tutorial Examples

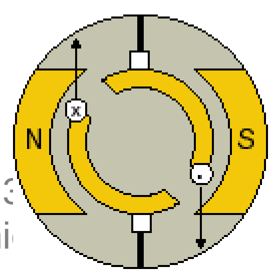
- PMDC Motor
- Revisit Driving Coil Example

Basic PMDC Motor Example



Basic PMDC motor equations

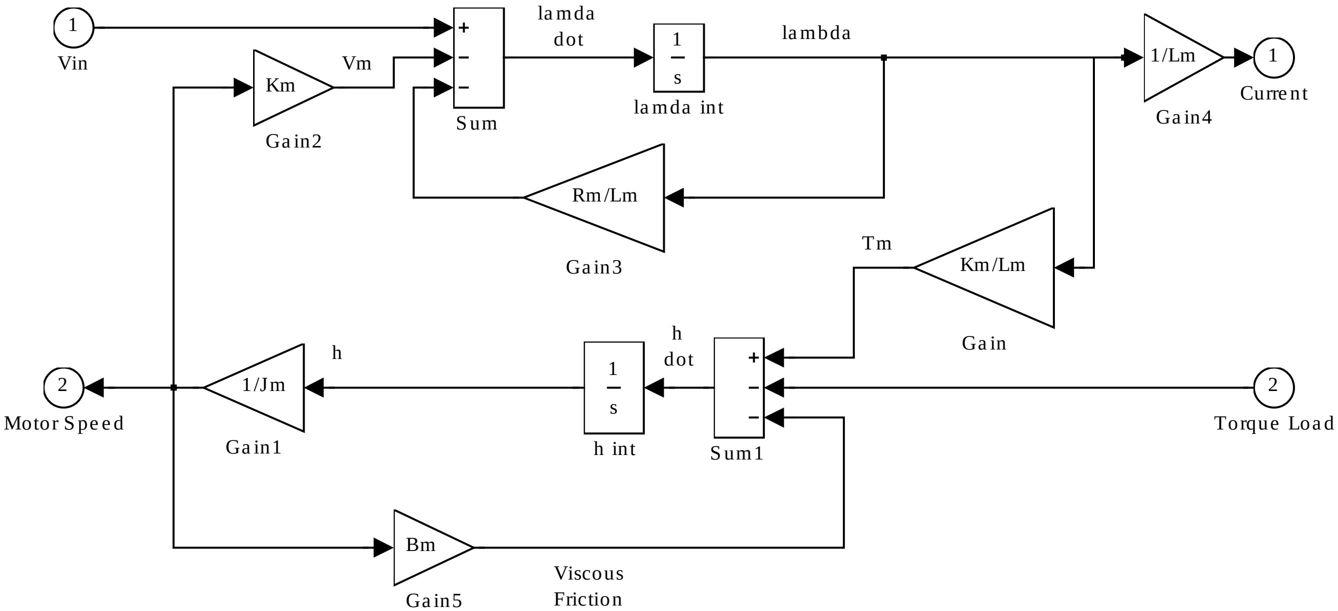
$$\begin{bmatrix}\dot{\lambda}\\ \dot{h}\end{bmatrix} = \begin{bmatrix} -\frac{R}{L_m}\lambda - \frac{k_m}{J_m}h + v_m \\ \frac{k_m}{L_m}\lambda - \frac{B_m}{J_m}h + T_L \end{bmatrix}$$

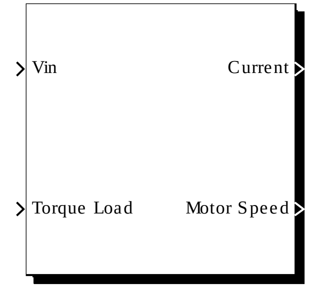


"fixed field"

Require:
- Load system
- Electrical source model

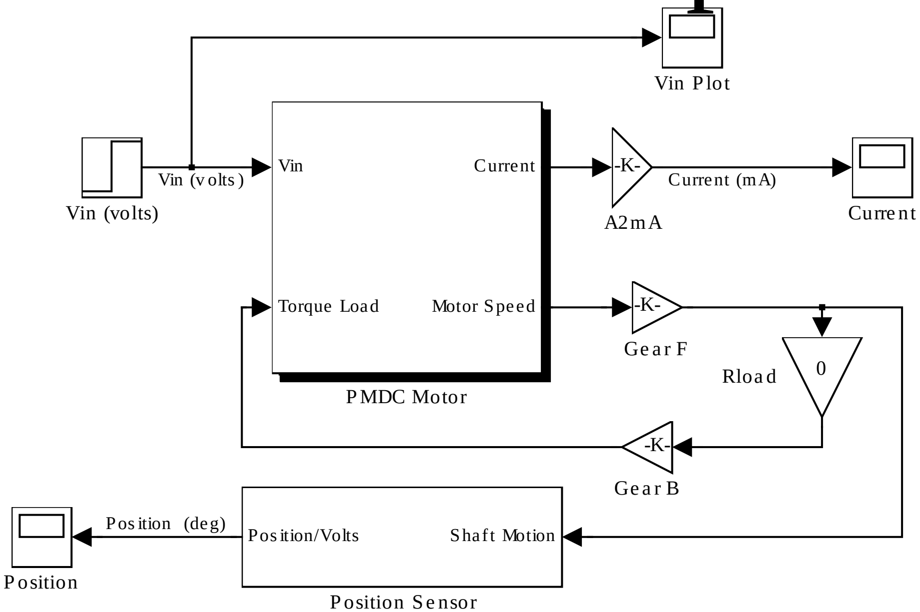
Model of PMDC Motor

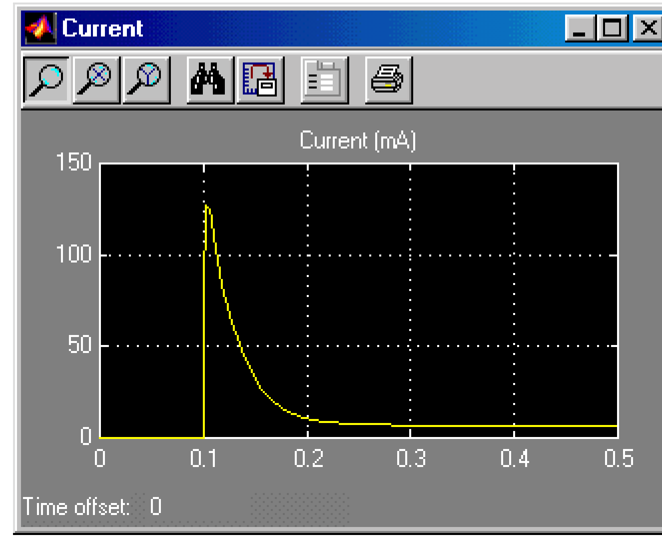


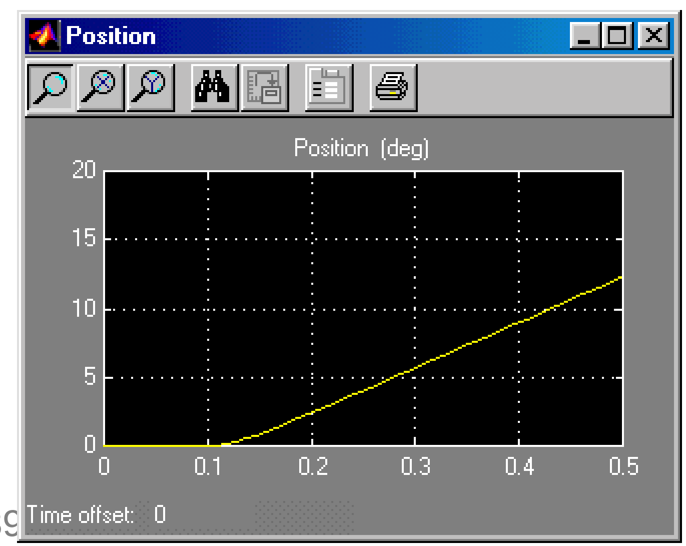


PMDC Motor

pmdc.mdl


Vin Plot
Vin (volts)
Vin (volts)
Vin
Current
-K-
A2mA
Current (mA)
Current
Torque Load
Motor Speed
-K-
Gear F
Rload
0
PMDC Motor
-K-
Gear B
Position
Position (deg)
Position/Volts
Shaft Motion
Position Sensor


Current
Current (mA)
150
100
50
0
0
0.1
0.2
0.3
0.4
0.5
Time offset: 0


Position
Position (deg)
20
15
10
5
0
0
0.1
0.2
0.3
0.4
0.5
Time offset: 0

Adding “Real Effects” in the PMDC Model

- Gear-reduction
- Backlash
- Stiction

Department of Mechanical Engineering
The University of Texas at Austin

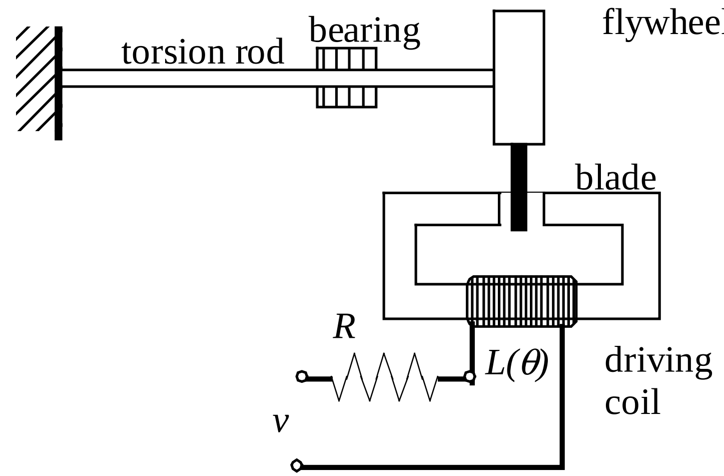
Simulink Modeling of Driving Coil Example

- Recall example from previous session.

- Develop a Simulink model of this system, and compare to m-file version.

Example: Angular Driving-Coil

$$\begin{bmatrix} \dot{\lambda} \\ \dot{\theta} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} v(t) - \dfrac{R}{L(\theta)}\lambda \\ \dfrac{1}{J}h \\ \dfrac{b\lambda^2}{2L(\theta)^2} - \dfrac{B}{J}h - K\theta \end{bmatrix}$$



Adapted from
S. Nasar, "Electric machines and electromechanics"

$J = \text{flywheel inertia} = 0.001\ \text{kg m}^2$

$K = 1.0\ \text{N m/rad} = \text{torsion rod stiffness}$

$B = 0.01\ \text{N sec/rad} = \text{bearing resistance}$

$R = 100\ \Omega$

$L(\theta) = 1 + 0.5\theta$ henry

$f = 10\ \text{Hz}$

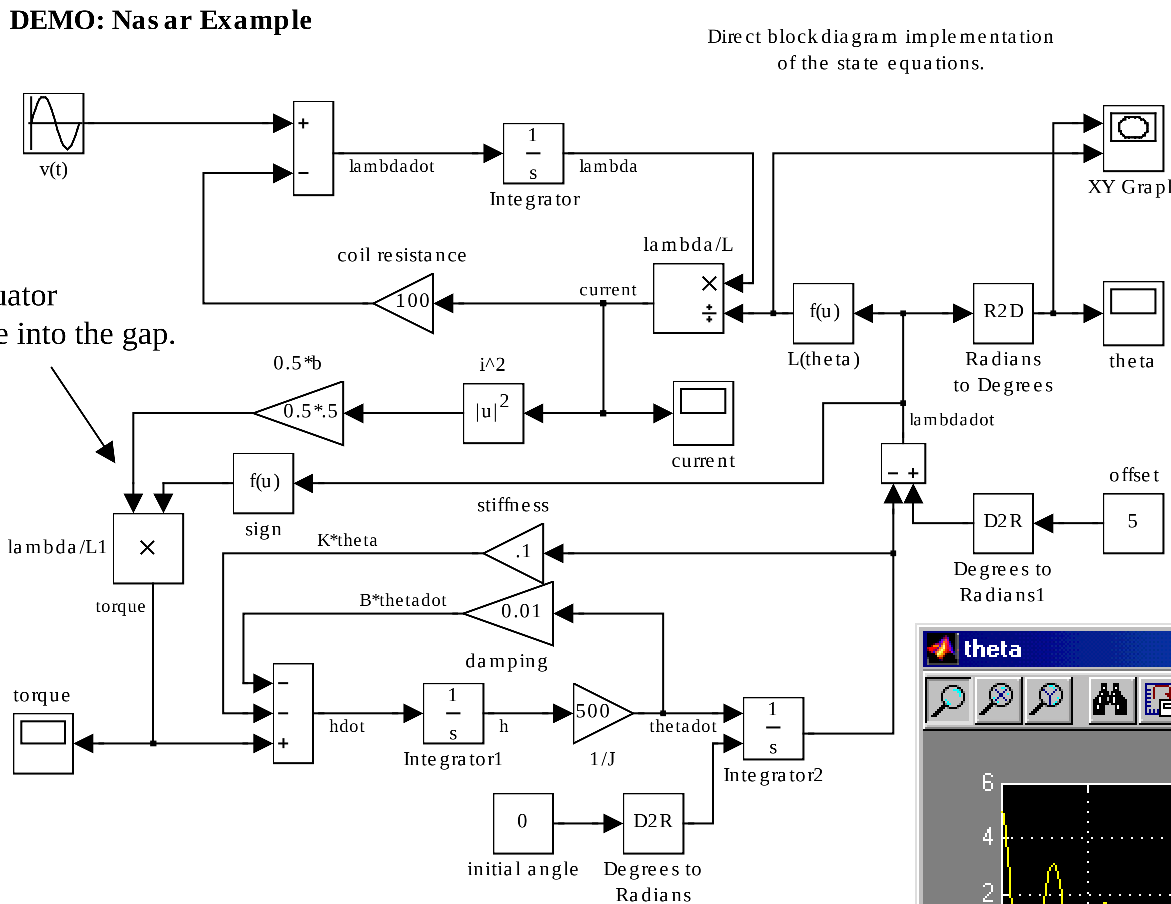
Drive this system with voltage of amplitude 10 volts at frequency *f*.

Building the Simulink Model

- Use a sine wave source as the input
- A function block can be used to represent the dependence of inductance on angle

- Assume initial conditions are such that the blade is sitting outside of the core material, and the coil pulls the blade into the gap

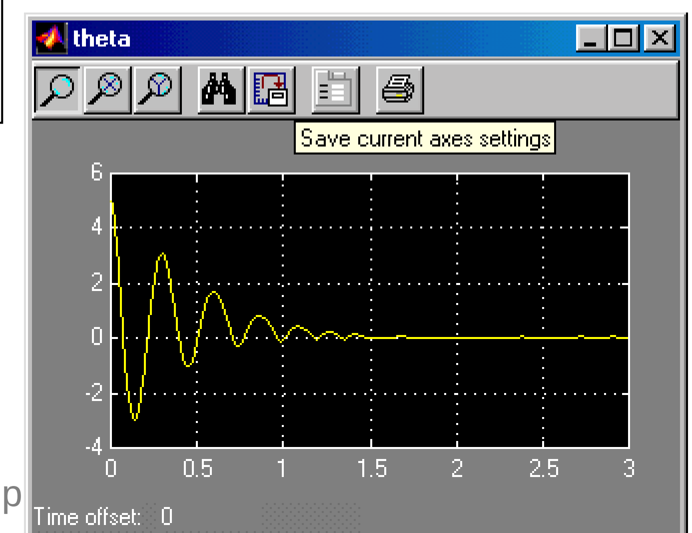
Drive coil example



Note: there has to be a way to have the actuator always drive the blade into the gap. This does it.

The sign is included.

Blade position will be brought to zero when coil is energized.



Equilibrium Point Determination

» help **trim**

TRIM   Finds steady state parameters for a system given a set of conditions.
TRIM enables steady state parameters to be found that satisfy
certain input, output and state conditions.

[X,U,Y,DX]=TRIM('SYS') attempts to find values for X, U and Y
that set the state derivatives, DX, of the S-function 'SYS' to
zero. TRIM uses a constrained optimization technique.

[X,U,Y,DX]=TRIM('SYS',X0,U0) sets the initial starting guesses
for X and U to X0 and U0, respectively.

Example: A Pressure-Compensated Pump System

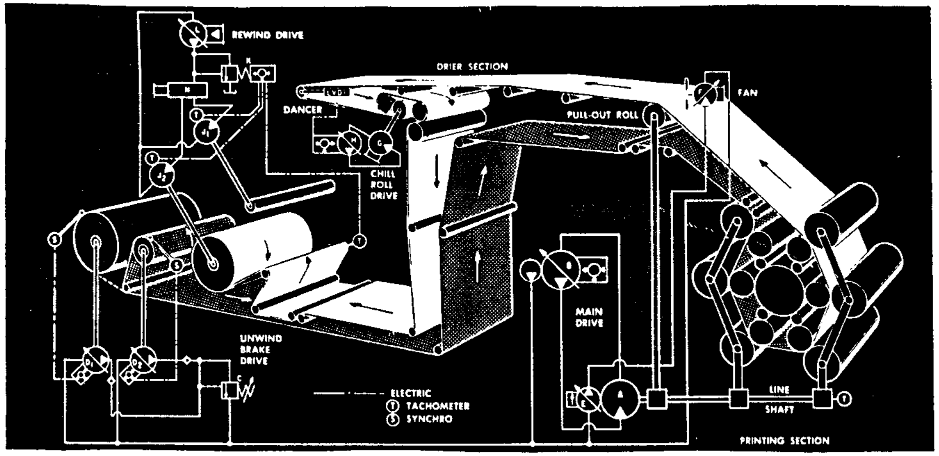


Figure 9.17 Hydraulically controlled printing press.

This example is a hydraulic drive from a printing press.

Ref: E.O. Doebelin, System Response Modeling

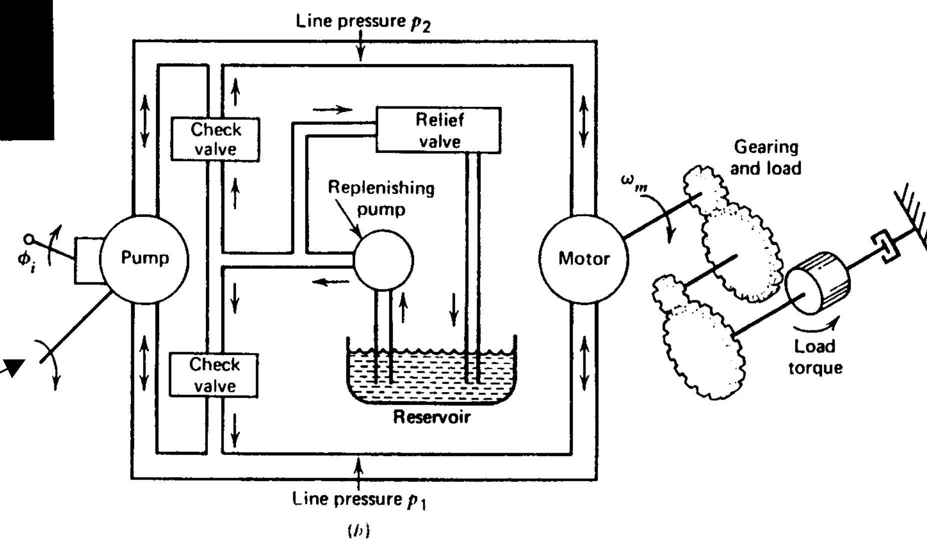
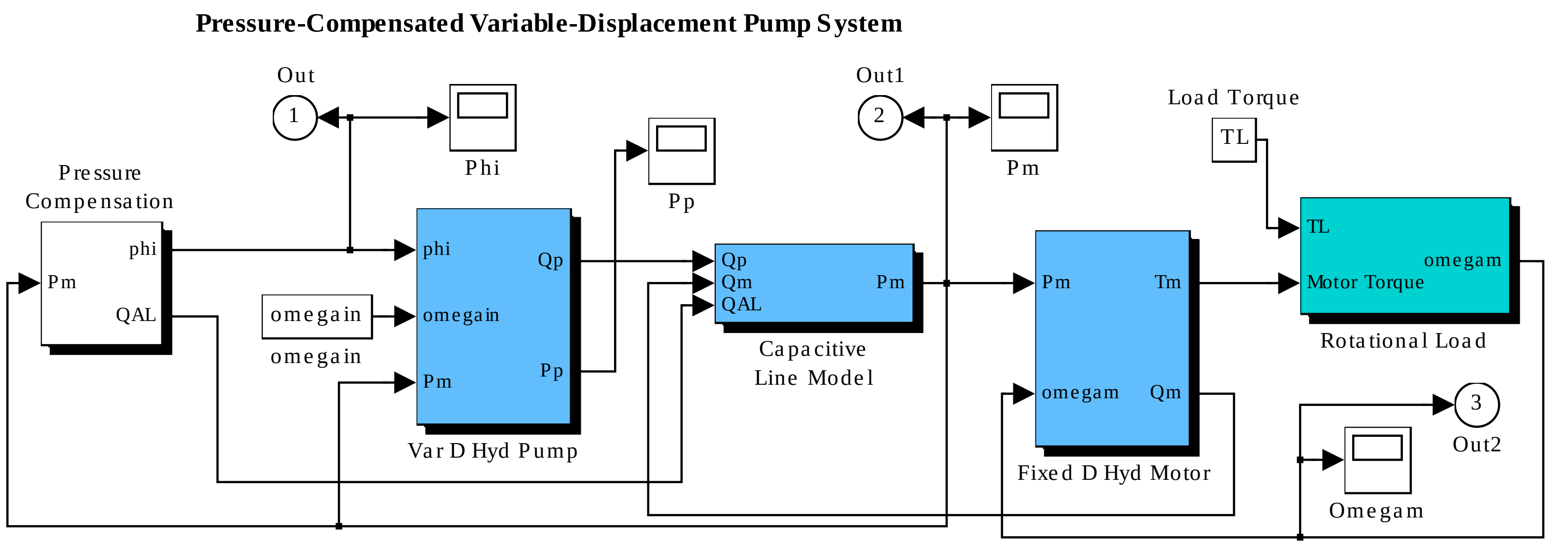


Figure 9.18 Pump/motor drive system.

If the pressure in the line is used to change the yoke angle, the feedback provides a pressure-compensation.

Let's look at a model of this system to demonstrate how you can use trim to find steady-states.

Example: The Pressure-Compensated Pump System



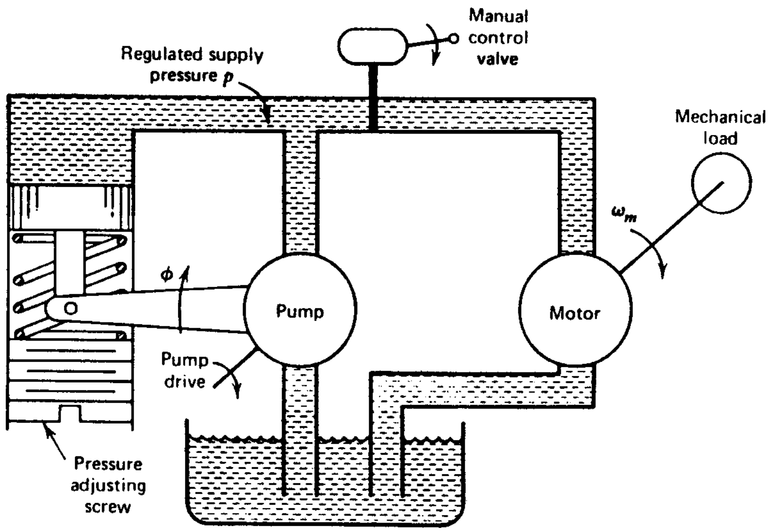


Figure 9.19 **Pressure-compensated variable-displacement pump system.**

R.G. Longoria
May 1998

Parameter file: psetup2.m

This model has multiple levels. We can find information on states as shown next.

Model Information Example

This command will provide information on a system model.

» [sizes, x0, xstord] = pcpump([],[],[],0)

Using this call from MATLAB returns the information shown.

sizes =

4 (# states)
0 (discrete)
3 (outputs)
0 (inputs)
0 (discontinuous roots)
0
2

x0 =

0.0033
0
0
0
(block initial conditions)

Using trim we can find the equilibrium closest to this initial condition.

xstord =

'pcpump/Pressure Compensation/phidot'
'pcpump/Capacitive Line Model/Vdot'
'pcpump/Rotational Load/hldot'
'pcpump/Pressure Compensation/hidot'

A string containing where to find the integrators.

Equilibrium for Pressure-Compensated Pump

Using **trim** to find steady-states

trim uses a sequential quadratic programming algorithm to find trim points. See the Optimization Toolbox User's Guide for a description of this algorithm.

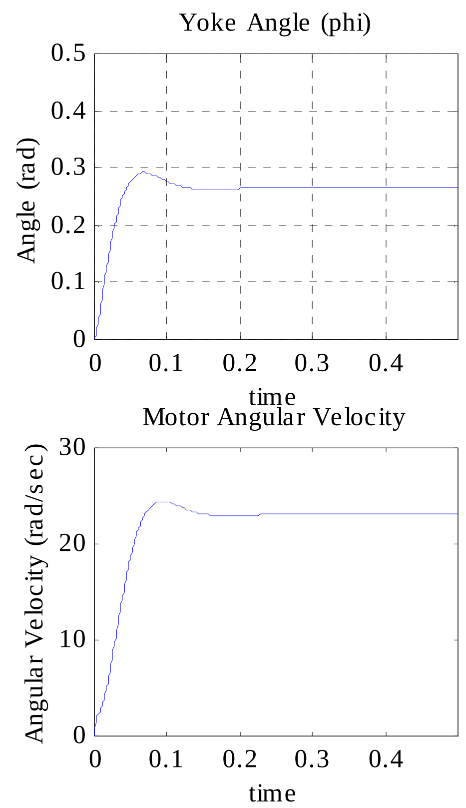
```
» [X,U,Y,DX]=TRIM('pcpump')

X =

    0.2644
    0.9482
    0.9908
    0.0000
```

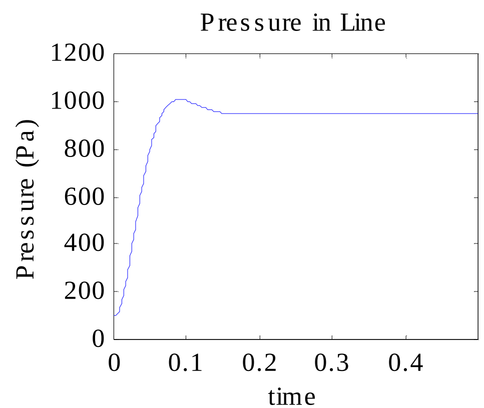
System states

```
U =

   Empty matrix: 0-by-1
```

System outputs

```
Y =

    0.2644    (radians)
  948.2204    (pressure)
   23.0418    (angular velocity)
```





Direct simulation shows the same SS values for the outputs (Y) are reached.

DEE

DEE offers a direct way to integrate nonlinear state-space equations.

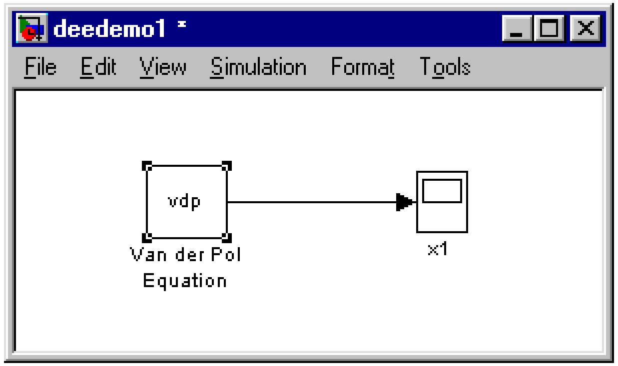
» help dee

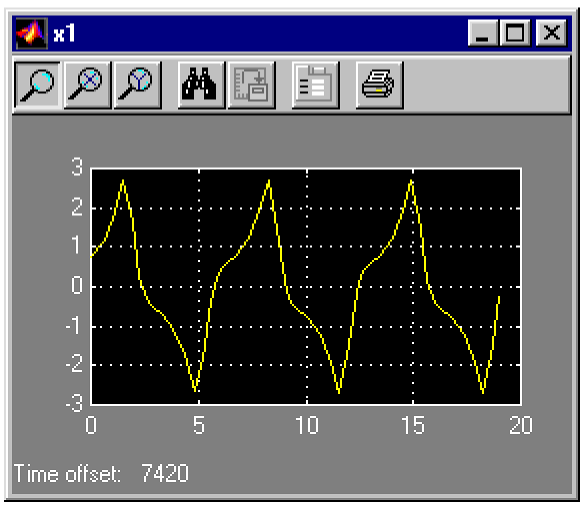
Differential Equation Editor

Demonstrations.

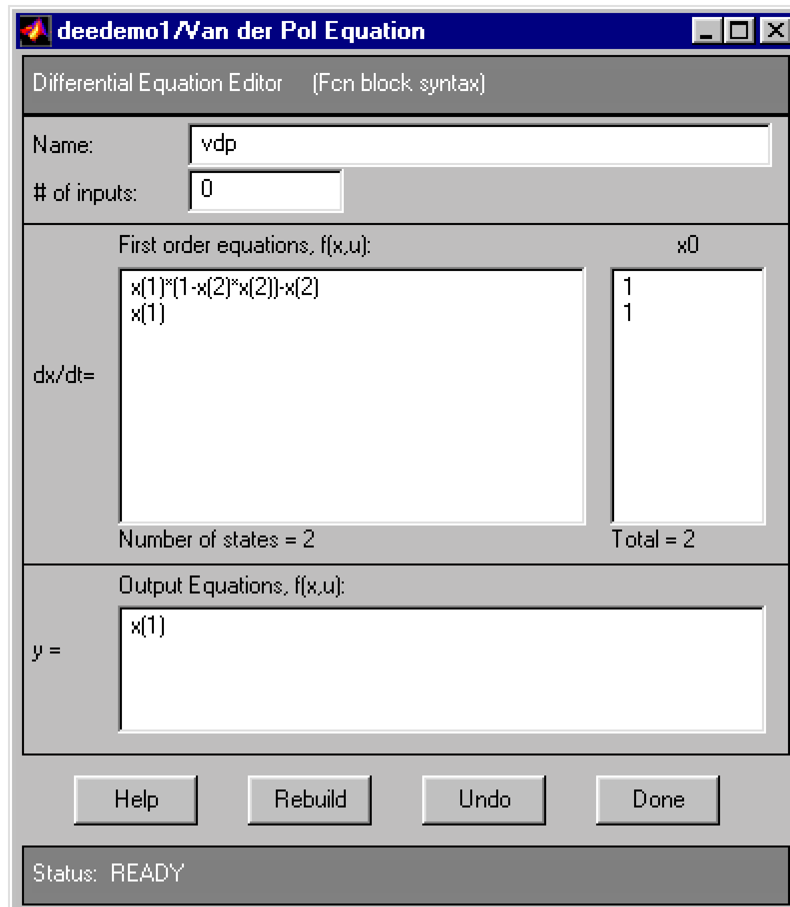
```
dee        - Simulink system containing DEE and associated demos.
deedemo1   - Van der Pol Equation.
deedemo2   - Lorenz Atractor.
deedemo3   - Mass and Spring.
deedemo4   - Coupled Mass and Spring System with Animation.
```

Example: The Classic Van der Pol

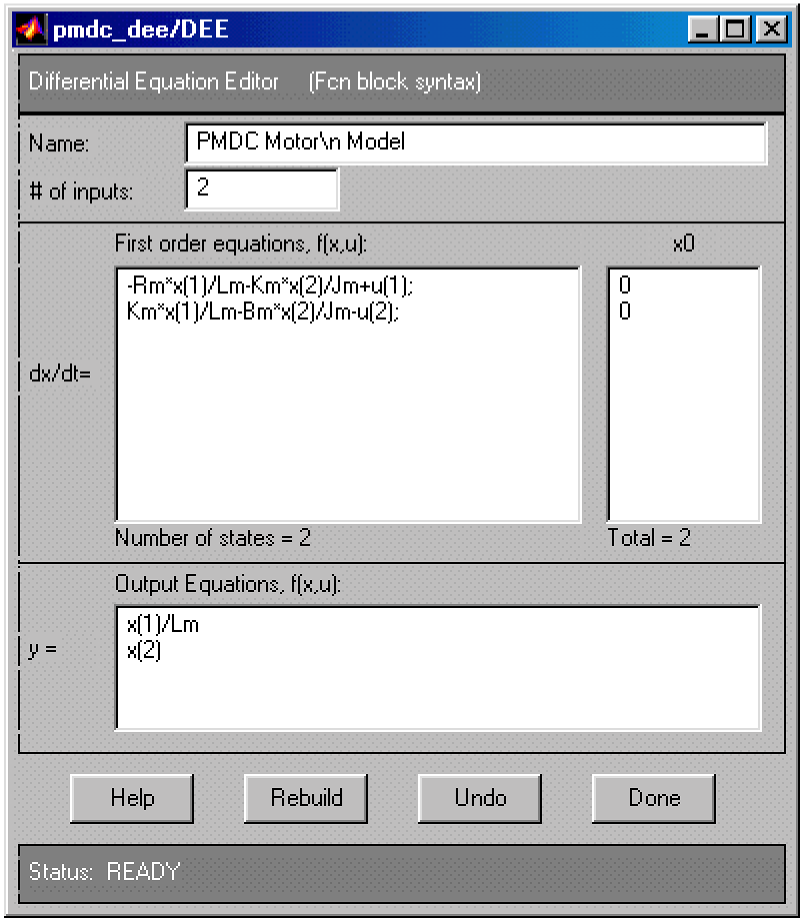


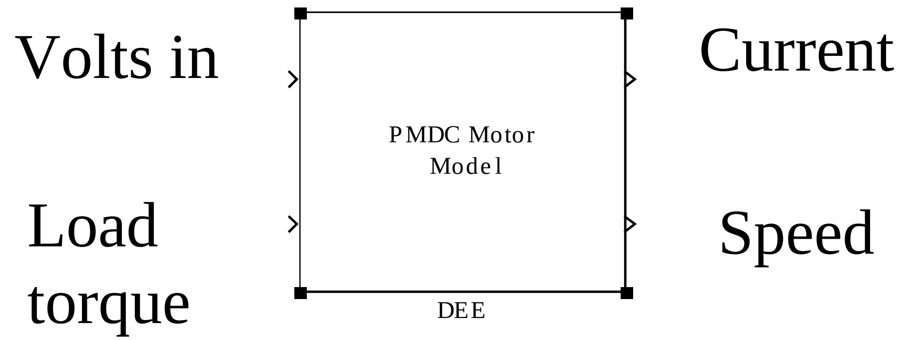


Minimize block diagram construction required.



pmdc using DEE





Rather than "block diagram" a given set of equations, the DEE allows you to insert them to this block. Input and output ports can be defined.

Summary

- We took a hands-on approach to learning how to use Simulink to build basic models.
- Some modeling issues were discussed
- Some of the basic analysis routines were used to conduct simple simulation.

Next time: S-functions, model libraries, more complex examples, difficult simulation cases, toolbox usage