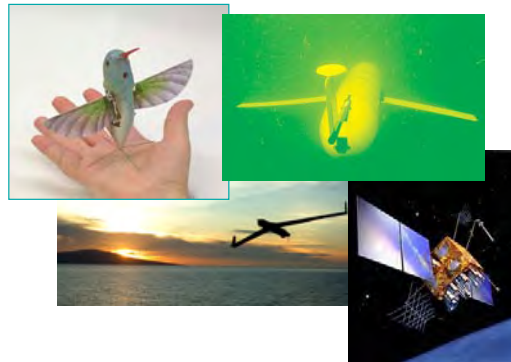


Flying and Swimming Robots

Robert Stengel

Robotics and Intelligent Systems MAE 345,
Princeton University, 2017

- Aircraft
- Aquatic robots
- Space robots
- Quaternions
- Simulink/Simscape/
SimMechanics



Copyright 2017 by Robert Stengel. All rights reserved. For educational use only.
<http://www.princeton.edu/~stengel/MAE345.html>

1

Bio-Inspiration for Flying



Hummingbird

<http://www.youtube.com/watch?v=D8vjYTXglJw&feature=related>



Eagle vs. Eagle

http://www.youtube.com/watch?v=tufnqWNP9AA&feature=video_response



Birds Flying

<http://www.youtube.com/watch?v=l5GbFgk-EPw>



Moth Flying

<https://www.youtube.com/watch?v=hD2BjAsvIbl>



Lady Bug

<http://www.youtube.com/watch?v=fjZobEZJYBc>

2

Biomimetic UAVs



Markus Fisher at TED
http://www.youtube.com/watch?v=Fg_JcKSHUtQ



Aerovironment Nano Hummingbird
<https://www.youtube.com/watch?v=xolH02Zba04>

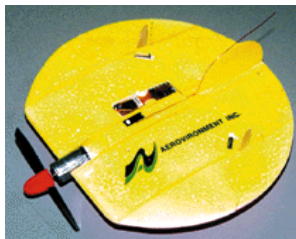


Festo Air Ray Dirigible
<http://www.youtube.com/watch?v=UxPzodKQays>



Harvard Robo-Flies
http://www.youtube.com/watch?v=2lQcKr0A_7c

3



Uninhabited Air Vehicles (UAV)



4

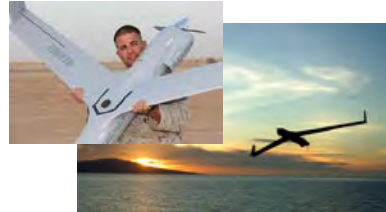
Uninhabited Aircraft

Tad McGeer, '79

Aerosonde
First UAV Transatlantic Crossing,
1998



**Boeing (InSitu)
ScanEagle**

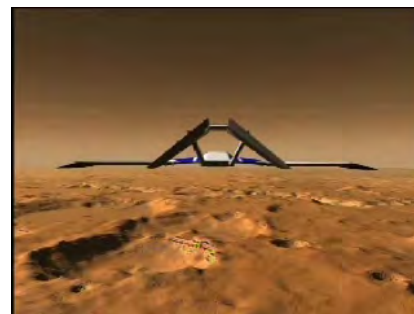
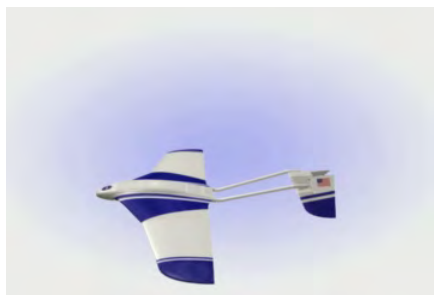


Aerovel Flexrotor



5

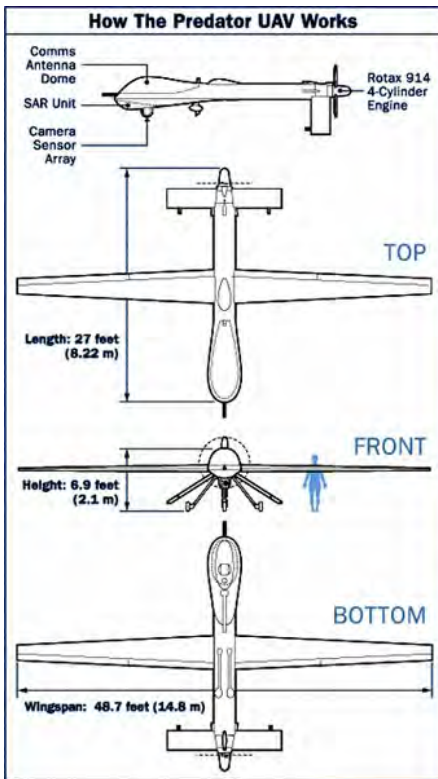
Mars Aerial Regional-Scale Environmental Survey (ARES) Research Airplane Concept, ~2008



<https://www.youtube.com/watch?v=8YutbpJuFil>

<https://www.youtube.com/watch?v=wAOTomGFs5M>

6



Uninhabited Aircraft



MQ-9 Reaper
<http://www.youtube.com/watch?v=kSpOYZR0kIA>

Aggressive Quadrotor UAV Maneuvers
<http://www.youtube.com/watch?v=MvRTALJp8DM>

Multi-Copters



Autonomous Air Taxi *Volocopter*

18 rotors
30-min flying time

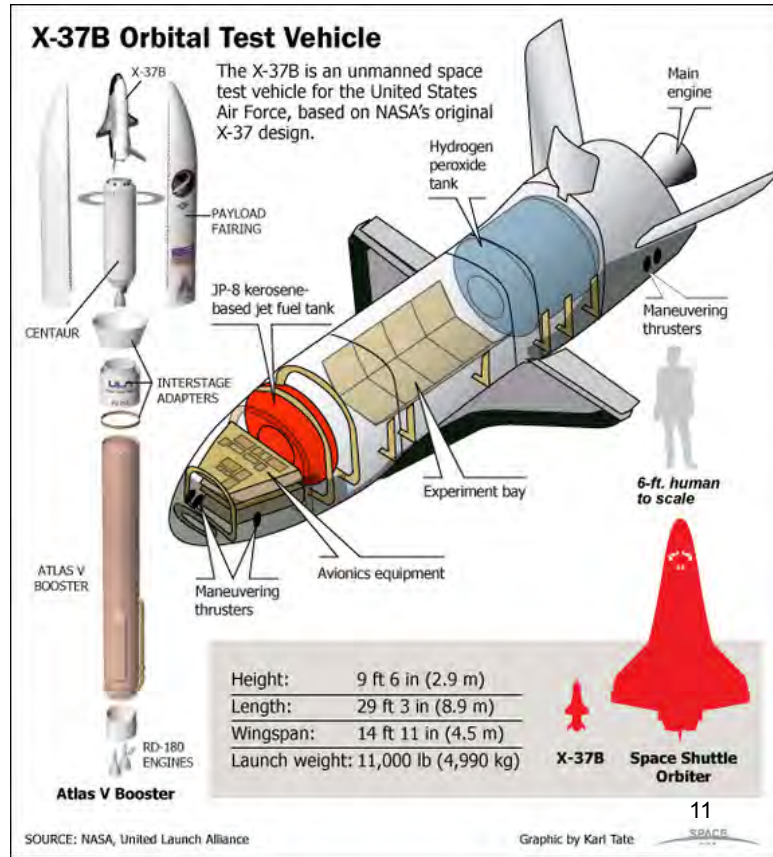


9

Space Robots

X-37B

- Reusable experimental/operational vehicle
- Unmanned “mini-Space Shuttle”
- Orbital maneuvering
- Highly classified project
- 1st 4 missions: 224, 469, 675, & 717 days in orbit
- 5th mission on-going



Expendable (Rocket) Launch Vehicles

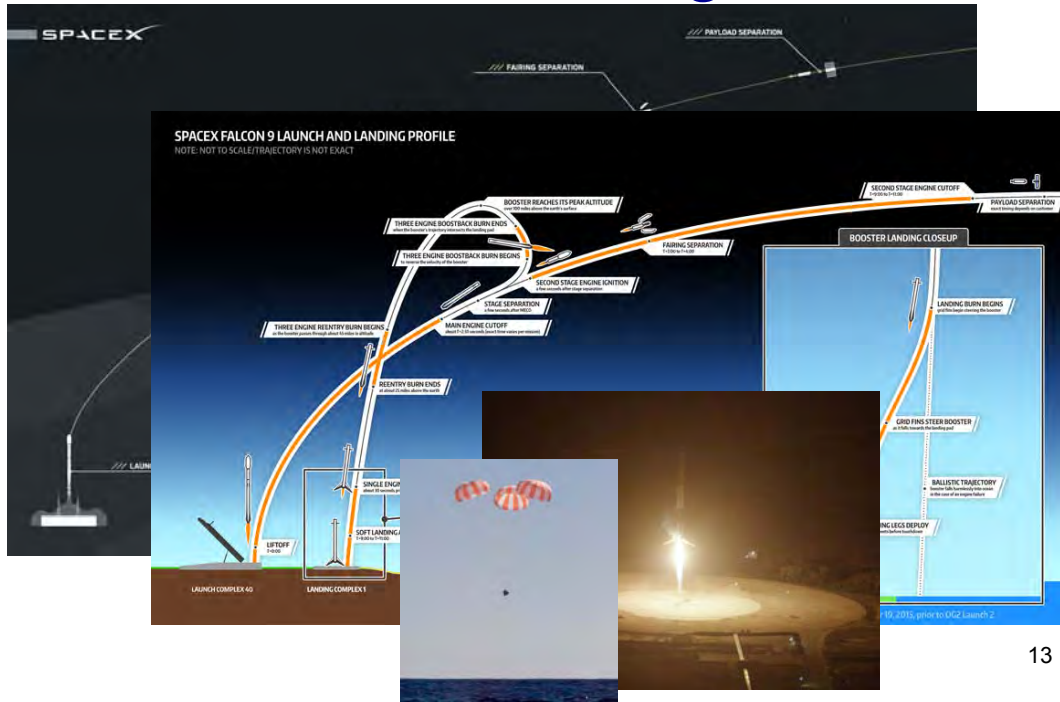
Current space launch vehicles are largely autonomous



Atlas V

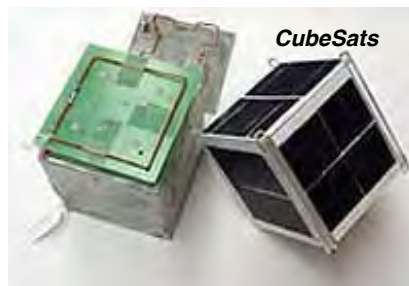
<http://www.youtube.com/watch?v=KxQbex7LJwg>

Reusable Launch/Reentry Vehicles Falcon 9/Dragon



13

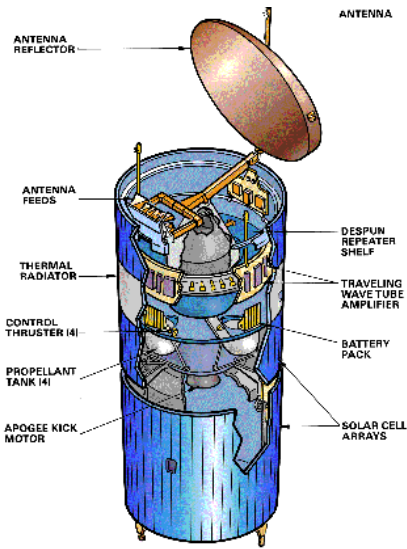
Uninhabited Spacecraft



14

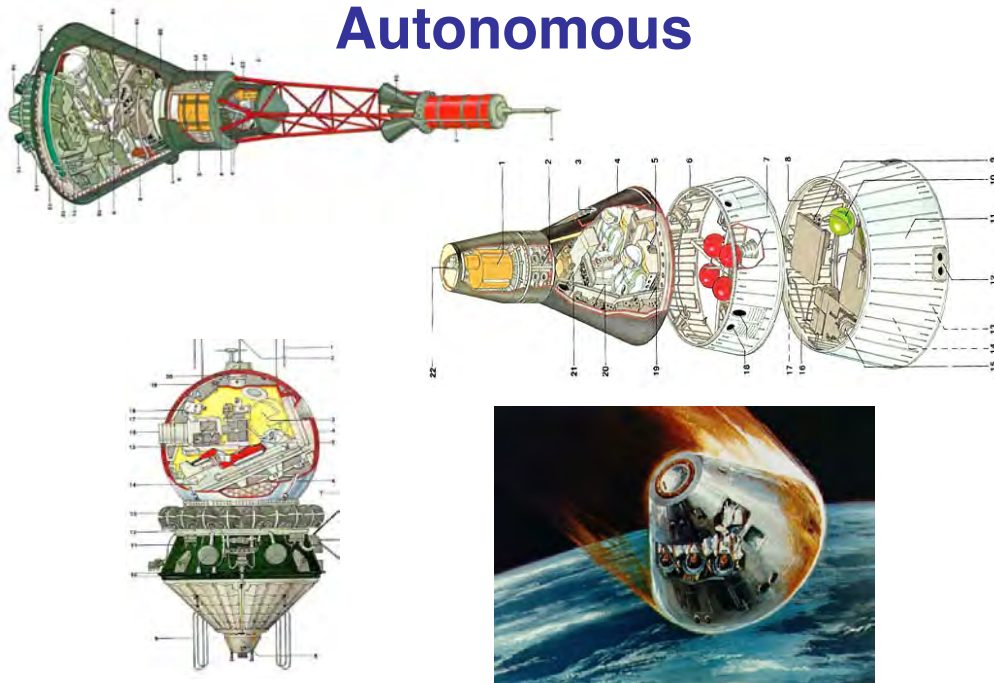


Uninhabited Spacecraft



BOEING 376 SPACECRAFT CONFIGURATION 15

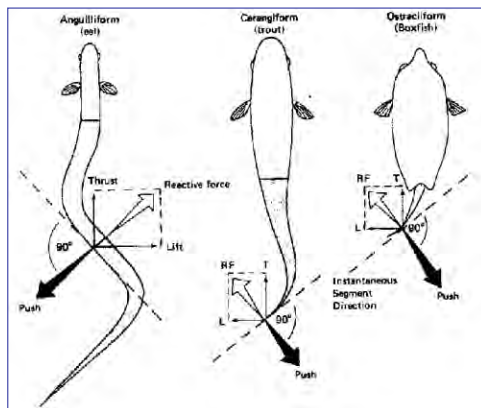
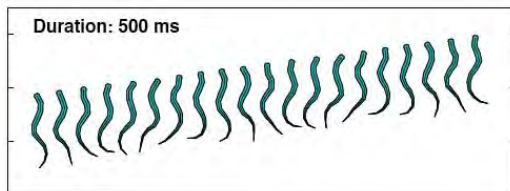
Manned Re-Entry Vehicles Largely Autonomous



Undersea Robots

17

Swimming Gaits



Anguilliform locomotion

Long, slender fish, e.g., lamprey
Amplitude of flexion wave along body ~ constant

Sub-carangiform locomotion

Increase in wave amplitude along the body
Most work done by rear half of fish body
Higher speed, reduced maneuverability

Carangiform locomotion

Stiffer and faster-moving, e.g., trout
Majority of movement rear of body and tail
Rapidly oscillating tails

Thunniform locomotion

High-speed long-distance swimmers, e.g. tuna, shark
Virtually all lateral movement in the tail
Tail itself is large and crescent-shaped

18

Swimming

- Lift, drag, and vorticity
- Schooling behavior



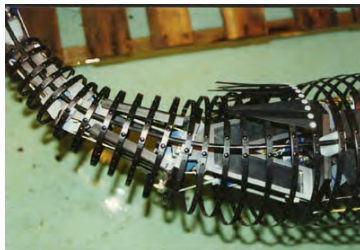
Human Swimming

<http://www.youtube.com/watch?v=ClzBaSiWdRA>

Fish Swimming

http://www.youtube.com/watch?v=U_VJ_0wORbM

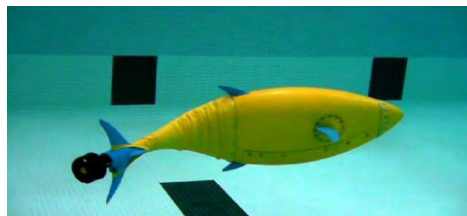
19



RoboTuna (Olin/MIT)

Autonomous Underwater Vehicles

RoboLobster (Northeastern)



<https://www.youtube.com/watch?v=pDitxrXeYnA>



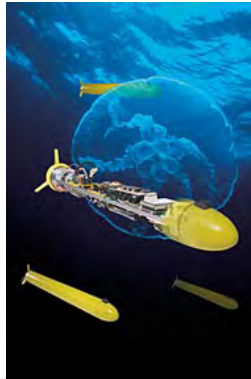
20

Autonomous Submarines

Autonomous Benthic Explorer



VPI concept



Oberon (U Sydney)

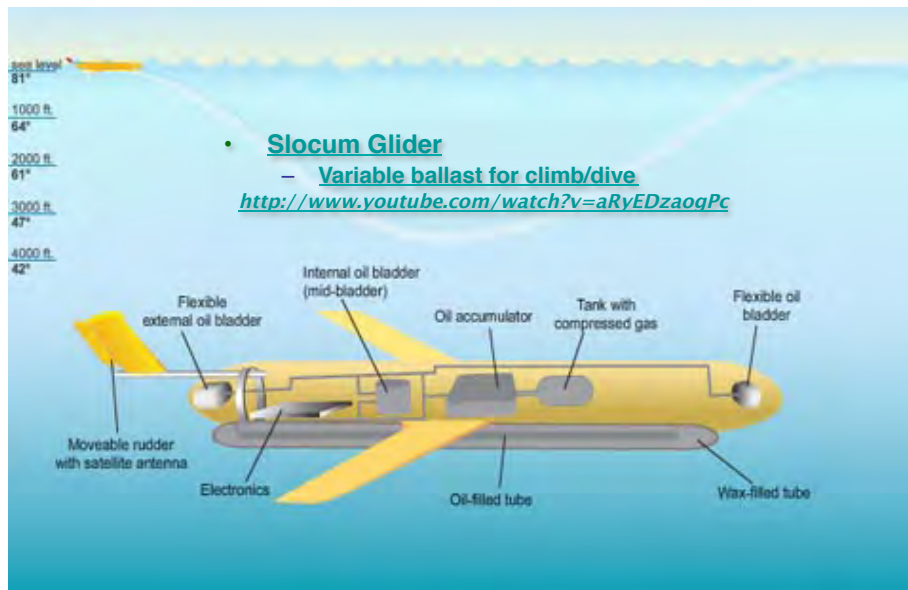


AQUA
http://www.youtube.com/watch?v=9Vm-gQ9_H9I&feature=related

AQUA encounters a lobster
<https://www.youtube.com/watch?v=FCOZFwzMiU8>

21

Autonomous Underwater Gliders



Slocum Glider
– Variable ballast for climb/dive
<http://www.youtube.com/watch?v=aRvEDzaogPc>

https://en.wikipedia.org/wiki/Underwater_glider

22

Avoiding the Euler Angle Singularity

23

Inverse Transformation for Euler-Angle Rates

$$(\mathbf{L}_I^B)^{-1} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix}^{-1} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix}$$

Euler-angle rates from body-axis rates

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \mathbf{L}_B^I \boldsymbol{\omega}_B$$

24

Avoiding the Euler Angle Singularity at $\theta = \pm 90^\circ$

Alternatives to Euler angles

- 1) Direction cosine (rotation) matrix
- 2) Quaternions

Propagation of direction cosine matrix (9 parameters)

$$\frac{d[\mathbf{H}_I^B(t)]}{dt} = -\tilde{\omega}_B(t)\mathbf{H}_I^B(t) = -\begin{bmatrix} 0 & -r(t) & q(t) \\ r(t) & 0 & -p(t) \\ -q(t) & p(t) & 0(t) \end{bmatrix}_B \mathbf{H}_I^B(t)$$

Initialize with Euler Angles

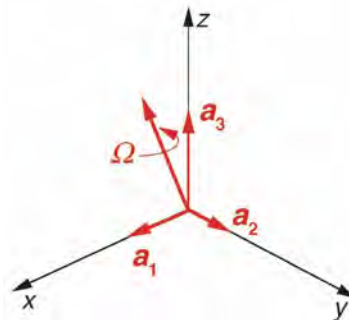
$$\mathbf{H}_I^B(0) = \mathbf{H}_I^B(\phi_0, \theta_0, \psi_0)$$

25

Avoiding the Euler Angle Singularity at $\theta = \pm 90^\circ$

Propagation of quaternion vector: single rotation
from inertial to body frame (4 parameters)

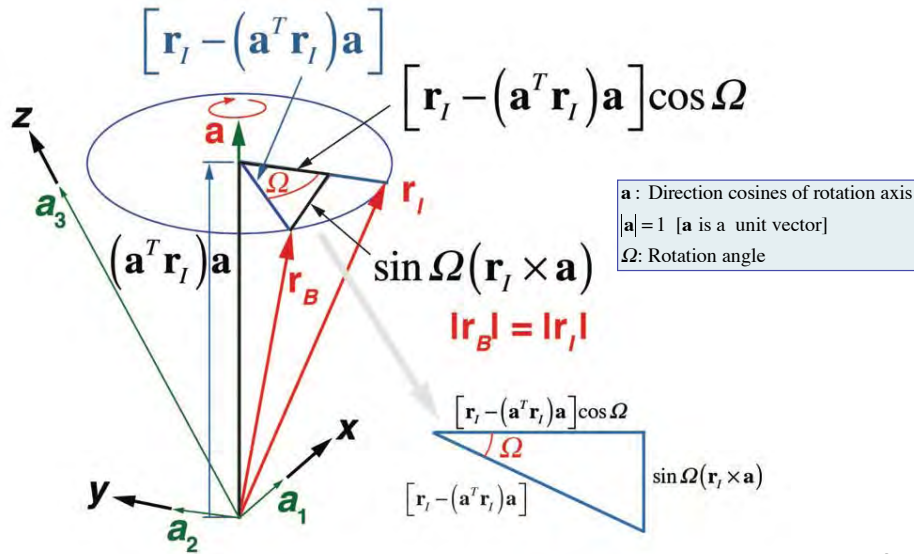
- Rotation from one axis system, I , to another, B , represented by
 - Orientation of axis vector about which the rotation occurs (3 parameters of a unit vector, \mathbf{a} : a_1 , a_2 , and a_3)
 - Magnitude of the rotation angle, Ω , rad



26

Begin with Euler Rotation of a Vector

Rotation about axis, \mathbf{a} , of a vector, \mathbf{r}_I , to a new orientation, \mathbf{r}_B



27

Development of Theorem

Defined vector is given a different orientation

$$|\mathbf{r}_B| = |\mathbf{r}_I| \quad \mathbf{r}_B = \mathbf{H}_I^B \mathbf{r}_I$$

Transformation involves addition of 3 vectors

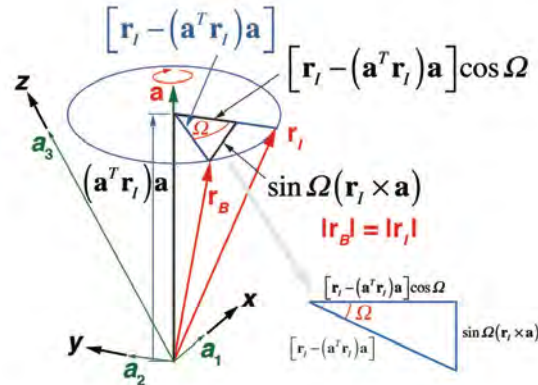
$(\mathbf{a}^T \mathbf{r}_I) \mathbf{a}$ $[\mathbf{r}_I - (\mathbf{a}^T \mathbf{r}_I) \mathbf{a}] \cos \Omega$ $\sin \Omega (\mathbf{r}_I \times \mathbf{a})$	Along axis of rotation \perp to \mathbf{a} and through \mathbf{r}_I \perp to \mathbf{a} and \mathbf{r}_I
---	--

Scaled by rotation angle, Ω , to produce \mathbf{r}_B

28

Development of Theorem

$$\begin{aligned}
 \mathbf{r}_B &= \mathbf{H}_I^B \mathbf{r}_I && \text{Combine terms} \\
 &= (\mathbf{a}^T \mathbf{r}_I) \mathbf{a} + \left[\mathbf{r}_I - (\mathbf{a}^T \mathbf{r}_I) \mathbf{a} \right] \cos \Omega + \sin \Omega (\mathbf{r}_I \times \mathbf{a}) && \text{Reverse cross-product order} \\
 &= \cos \Omega \mathbf{r}_I + (1 - \cos \Omega) (\mathbf{a}^T \mathbf{r}_I) \mathbf{a} - \sin \Omega (\mathbf{a} \times \mathbf{r}_I)
 \end{aligned}$$



29

Rotation Matrix Derived from Euler's Formula

$$\mathbf{r}_B = \mathbf{H}_I^B \mathbf{r}_I = \cos \Omega \mathbf{r}_I + (1 - \cos \Omega) (\mathbf{a}^T \mathbf{r}_I) \mathbf{a} - \sin \Omega (\tilde{\mathbf{a}} \mathbf{r}_I)$$

Identity

$$(\mathbf{a}^T \mathbf{r}_I) \mathbf{a} = (\mathbf{a} \mathbf{a}^T) \mathbf{r}_I$$

Cancel like terms

$$\left[\mathbf{H}_I^B \right] \cancel{\mathbf{r}_I} = \left[\cos \Omega + (1 - \cos \Omega) (\mathbf{a} \mathbf{a}^T) - \sin \Omega \tilde{\mathbf{a}} \right] \cancel{\mathbf{r}_I}$$

Rotation matrix

$$\mathbf{H}_I^B = \cos \Omega \mathbf{I}_3 + (1 - \cos \Omega) \mathbf{a} \mathbf{a}^T - \sin \Omega \tilde{\mathbf{a}}$$

30

Quaternion Derived from Euler Rotation Angle and Orientation

Quaternion vector

4 parameters based on Euler's rotation formula

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \triangleq \begin{bmatrix} \mathbf{q}_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \sin(\Omega/2) \mathbf{a} \\ \cos(\Omega/2) \end{bmatrix} = \begin{bmatrix} \sin(\Omega/2) \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \\ \cos(\Omega/2) \end{bmatrix} \quad (4 \times 1)$$

4-parameter representation of 3 parameters;
hence, a **constraint** must be satisfied

$$\begin{aligned} \mathbf{q}^T \mathbf{q} &= q_1^2 + q_2^2 + q_3^2 + q_4^2 \\ &= \sin^2(\Omega/2)(a_1^2 + a_2^2 + a_3^2) + \cos^2(\Omega/2) \\ &= \sin^2(\Omega/2) + \cos^2(\Omega/2) = \mathbf{1} \end{aligned}$$

31

Rotation Matrix Expressed with Quaternion

Euler's formula expressed with quaternion

$$\mathbf{H}_I^B = \left[q_4^2 - (\mathbf{q}_3^T \mathbf{q}_3) \right] \mathbf{I}_3 + 2\mathbf{q}_3 \mathbf{q}_3^T - 2q_4 \tilde{\mathbf{q}}_3$$

Terms of rotation matrix from quaternion elements

$$\mathbf{H}_I^B = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix}$$

32

Initial Quaternion Expressed from Elements of Rotation Matrix

Initialize $\mathbf{q}(0)$ from Direction Cosine Matrix or Euler Angles

$$\mathbf{H}_I^B(0) = \begin{bmatrix} h_{11}(=\cos\delta_{11}) & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} = \mathbf{H}_I^B(\phi_0, \theta_0, \psi_0)$$

$$q_4(0) = \frac{1}{2} \sqrt{1 + h_{11}(0) + h_{22}(0) + h_{33}(0)}$$

Assuming that $q_4 \neq 0$

$$\mathbf{q}_3(0) \triangleq \begin{bmatrix} q_1(0) \\ q_2(0) \\ q_3(0) \end{bmatrix} = \frac{1}{4q_4(0)} \begin{bmatrix} [h_{23}(0) - h_{32}(0)] \\ [h_{31}(0) - h_{13}(0)] \\ [h_{12}(0) - h_{21}(0)] \end{bmatrix}$$

33

Quaternion Vector Kinematics

$$\dot{\mathbf{q}} = \frac{d}{dt} \begin{bmatrix} \mathbf{q}_3 \\ q_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_4 \boldsymbol{\omega}_B - \tilde{\boldsymbol{\omega}}_B \mathbf{q}_3 \\ -\boldsymbol{\omega}_B^T \mathbf{q}_3 \end{bmatrix} \quad (4 \times 1)$$

(4 x 4) skew-symmetric matrix

$$\frac{d\mathbf{q}(t)}{dt} = \begin{bmatrix} \dot{q}_1(t) \\ \dot{q}_2(t) \\ \dot{q}_3(t) \\ \dot{q}_4(t) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & r(t) & -q(t) & p(t) \\ -r(t) & 0 & p(t) & q(t) \\ q(t) & -p(t) & 0 & r(t) \\ -p(t) & -q(t) & -r(t) & 0 \end{bmatrix}_B \begin{bmatrix} q_1(t) \\ q_2(t) \\ q_3(t) \\ q_4(t) \end{bmatrix}$$

Digital integration to compute $\mathbf{q}(t_k)$

$$\mathbf{q}_{\text{int}}(t_k) = \mathbf{q}(t_{k-1}) + \int_{t_{k-1}}^{t_k} \frac{d\mathbf{q}(\tau)}{dt} d\tau$$

34

Rigid Body Equations of Motion Using Quaternion

$$\begin{aligned}
 \dot{\mathbf{r}}_l(t) &= \mathbf{H}_B^l[\mathbf{q}(t)]\mathbf{v}_B(t) \\
 \dot{\mathbf{v}}_B(t) &= \frac{1}{m}\mathbf{f}_B(t) - \tilde{\boldsymbol{\omega}}_B(t)\mathbf{v}_B(t) \\
 \dot{\mathbf{q}}(t) &= \frac{1}{2} \begin{bmatrix} 0 & r(t) & -q(t) & p(t) \\ -r(t) & 0 & p(t) & q(t) \\ q(t) & -p(t) & 0 & r(t) \\ -p(t) & -q(t) & -r(t) & 0 \end{bmatrix} \mathbf{q}(t) \\
 \dot{\boldsymbol{\omega}}_B(t) &= \mathbb{I}_B^{-1}[\mathbf{m}_B(t) - \tilde{\boldsymbol{\omega}}_B(t)\mathbb{I}_B\boldsymbol{\omega}_B(t)]
 \end{aligned}$$

<http://www.princeton.edu/~stengel/FDcodeB.html>

35

Euler Angles Derived from Quaternion

- **atan2**: generalized arctangent algorithm, 2 arguments
 - returns angle in proper quadrant
 - avoids dividing by zero
 - has various definitions, e.g., (MATLAB)

$$\text{atan2}(y,x) = \begin{cases} \tan^{-1}\left(\frac{y}{x}\right) & \text{if } x > 0 \\ \pi + \tan^{-1}\left(\frac{y}{x}\right), -\pi + \tan^{-1}\left(\frac{y}{x}\right) & \text{if } x < 0 \text{ and } y \geq 0, < 0 \\ \pi/2, -\pi/2 & \text{if } x = 0 \text{ and } y > 0, < 0 \\ 0 & \text{if } x = 0 \text{ and } y = 0 \end{cases}$$

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \text{atan2}\left\{2(q_1q_4 + q_2q_3), [1 - 2(q_1^2 + q_2^2)]\right\} \\ \sin^{-1}\left[2(q_2q_4 - q_1q_3)\right] \\ \text{atan2}\left\{2(q_3q_4 + q_1q_2), [1 - 2(q_2^2 + q_3^2)]\right\} \end{bmatrix}$$

36

Solving Math Problems Computationally

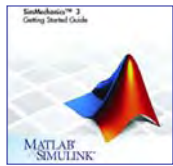
Task : Calculate $x_1(t)$ and $x_2(t)$ for $t = 1$ to 10 sec

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ -x_1(t) - x_2(t) \end{bmatrix}$$

with initial conditions

$$\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 10 \end{bmatrix}$$

37



MATLAB Models of Dynamic Systems

Systems are described by *instructions*

Main Script

```
% Linear 2nd-Order Example
clear
tspan = [0 10];
xo = [0, 10];
[t,x] = ode23('Lin',tspan,xo);

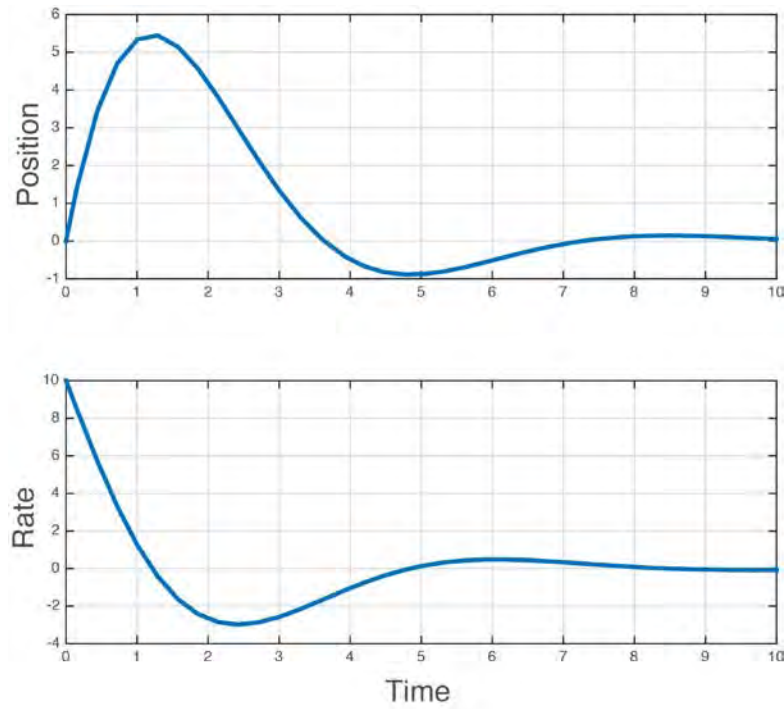
subplot(2,1,1)
plot(t,x(:,1))
ylabel('Position'), grid
subplot(2,1,2)
plot(t,x(:,2))
xlabel('Time'), ylabel('Rate'), grid
```

Function

```
function xdot = Lin(t,x)
% Linear Ordinary Differential Equation
% x(1) = Position
% x(2) = Rate
xdot = [x(2)
        -x(1) - x(2)];
```

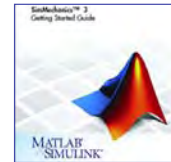
38

MATLAB Initial-Condition Output



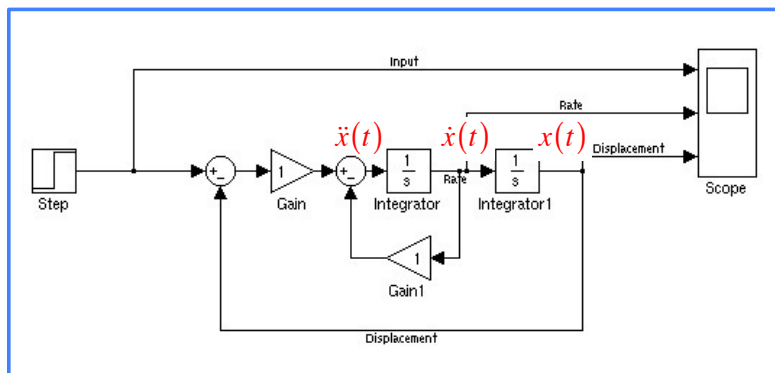
39

Simulink Models of Dynamic Systems



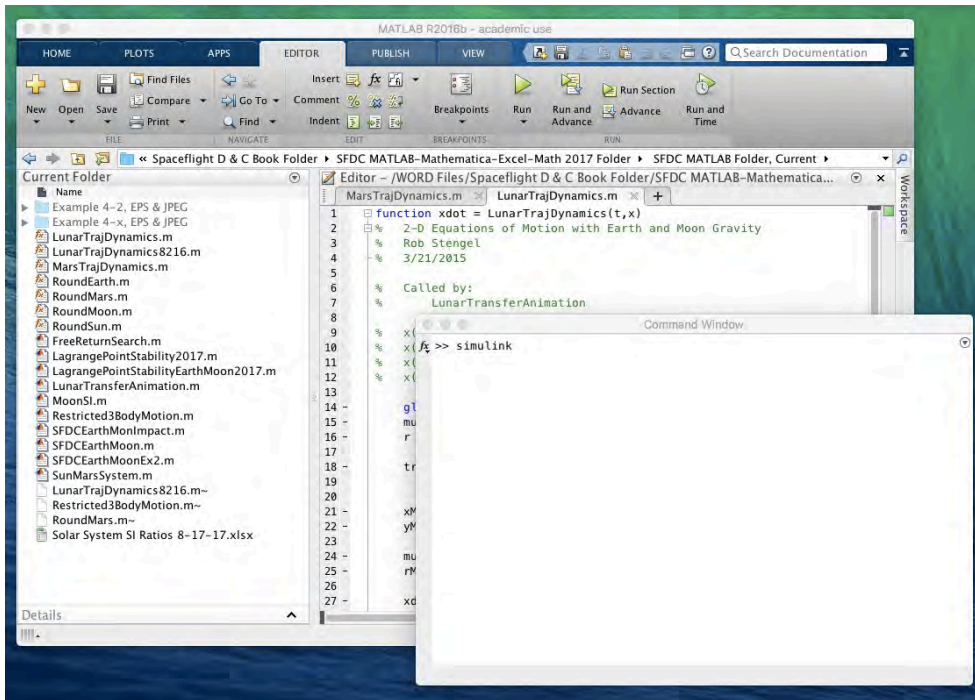
Systems are described by **block diagrams**

$$\frac{d^2x(t)}{dt^2} = \ddot{x}(t) = -x(t) - \dot{x}(t) + u(t)$$



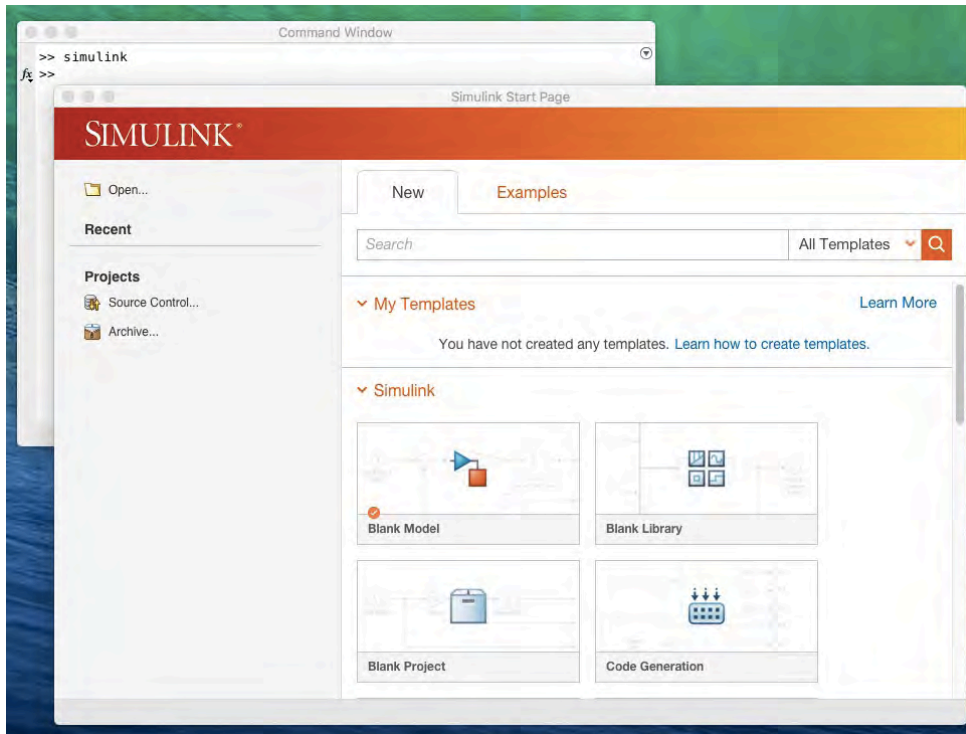
40

Accessing Simulink from MATLAB



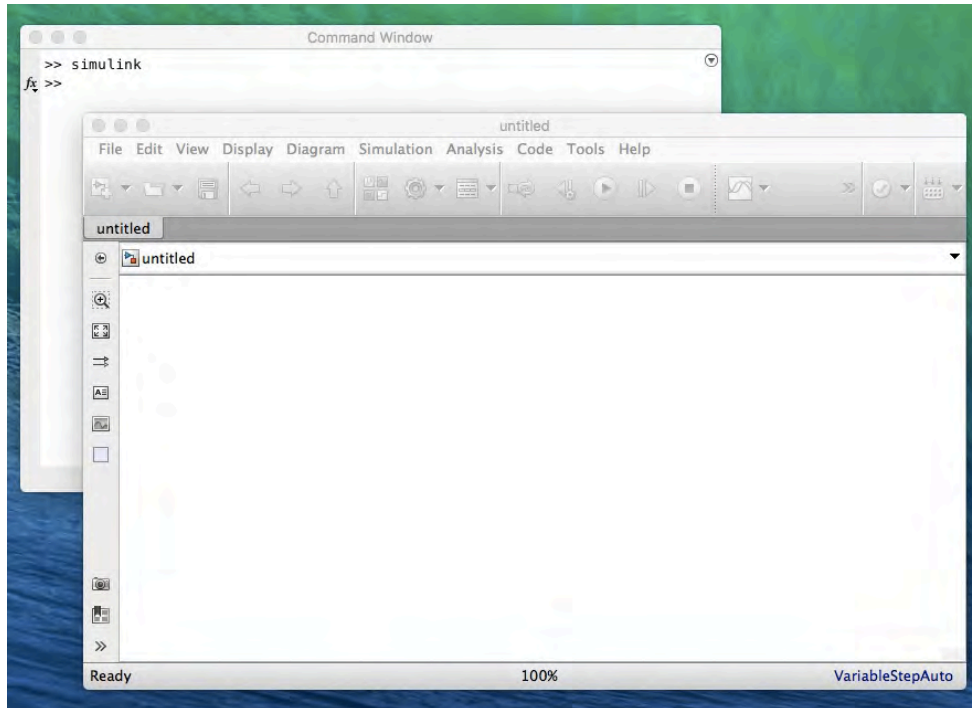
41

Accessing Simulink from MATLAB



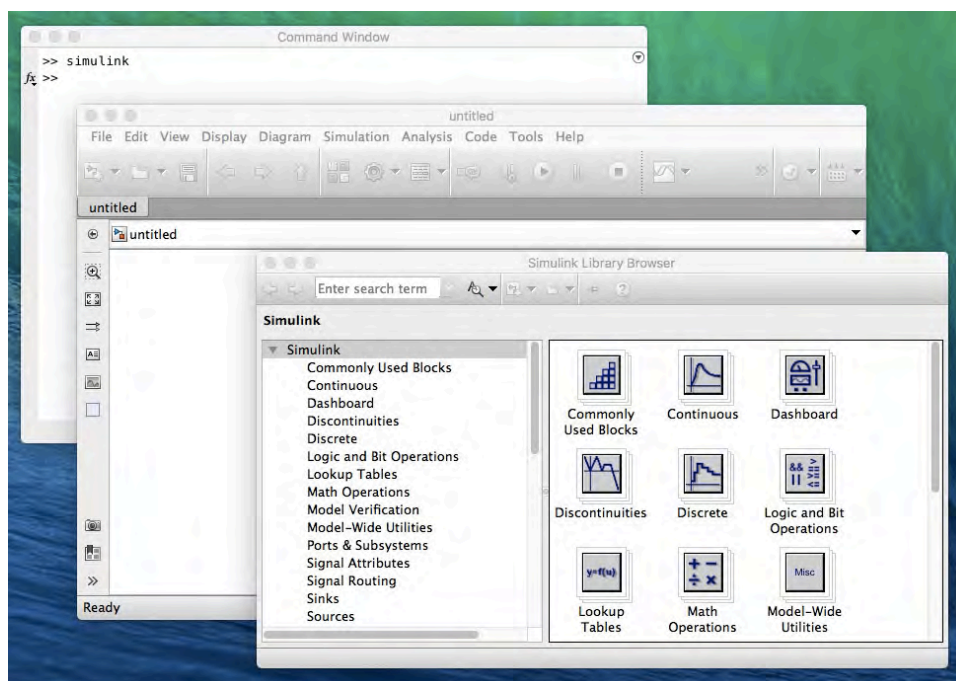
42

Open Simulink Blank Model



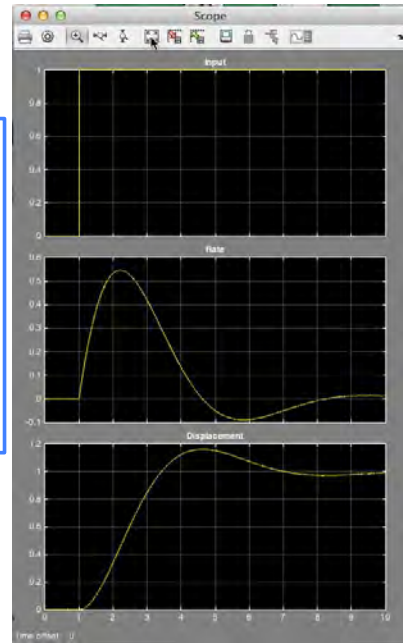
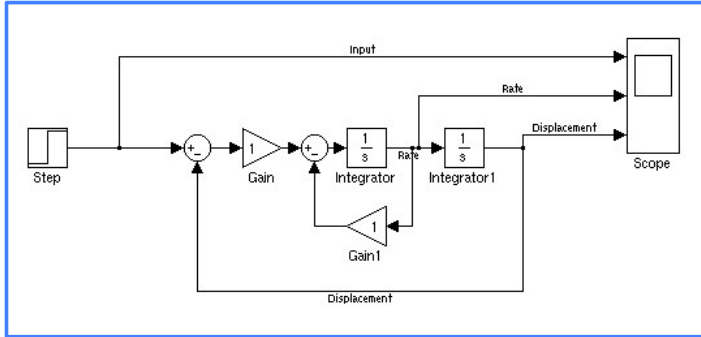
43

Open Simulink Library Browser for Function Blocks



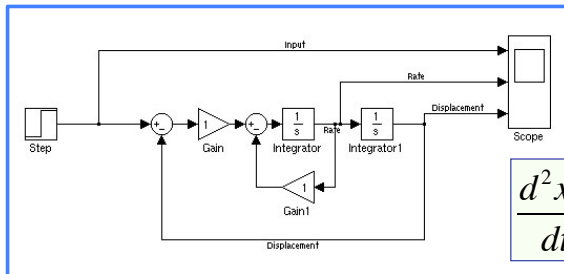
44

Simulink Step Response



45

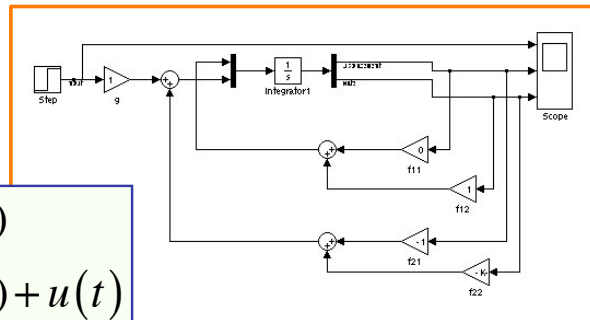
Alternative Simulink Models of 2nd-Order Systems



Single 2nd-order model, with step input and damping

$$\frac{d^2x(t)}{dt^2} = \ddot{x}(t) = -x(t) - \dot{x}(t) + u(t)$$

State-space model (two 1st-order equations), with step input and damping



$$\dot{x}_1(t) = (0)x_1(t) + (1)x_2(t)$$

$$\dot{x}_2(t) = -(1)x_1(t) - Kx_2(t) + u(t)$$

46

Simulink Autocoding

Automatic Code Generation

Pires, NASA Ames

```
142 /* Gain: '<S1E>/Gain1' incorporated:
143 * Constant: '<S1E>/Constant3'
144 * Constant: '<S1E>/Constant3'
145 * Gain: '<S1E>/Gain1'
146 * Sum: '<S1E>/Sum'
147 * Sum: '<S1E>/Sum3'
148 */
149 rtb_gain1_o = (5.0 - (10.0 - rtb_h_cs) * 0.4) * -1.0;
150 rtb_switch_g_idx = 0.1 - rtb_gain1_o;
151 rtb_switch_g_idx = rtb_gain1_o - 1.0;
```

- Graphic modeling of dynamic systems
- Library of functions
- Generation of C and C++ code

47

SimMechanics is Mechanical Subset of SimScape Library

Simulink Library Browser

File Edit View Help

Enter search term

Library: Simscape/Foundation Library/Mechanical/Mechanical Sensors Search Results: (none)

Libraries

- Simulink
- Control System Toolbox
- Instrument Control Toolbox
- Neural Network Toolbox
- Real-Time Workshop
- Signal Processing Blockset
- Simscape
 - Foundation Library
 - Electrical
 - Hydraulic
 - Mechanical
 - Mechanical Sensors**
 - Mechanical Sources
 - Mechanisms
 - Rotational Elements
 - Translational Elem...
 - Physical Signals
 - Thermal

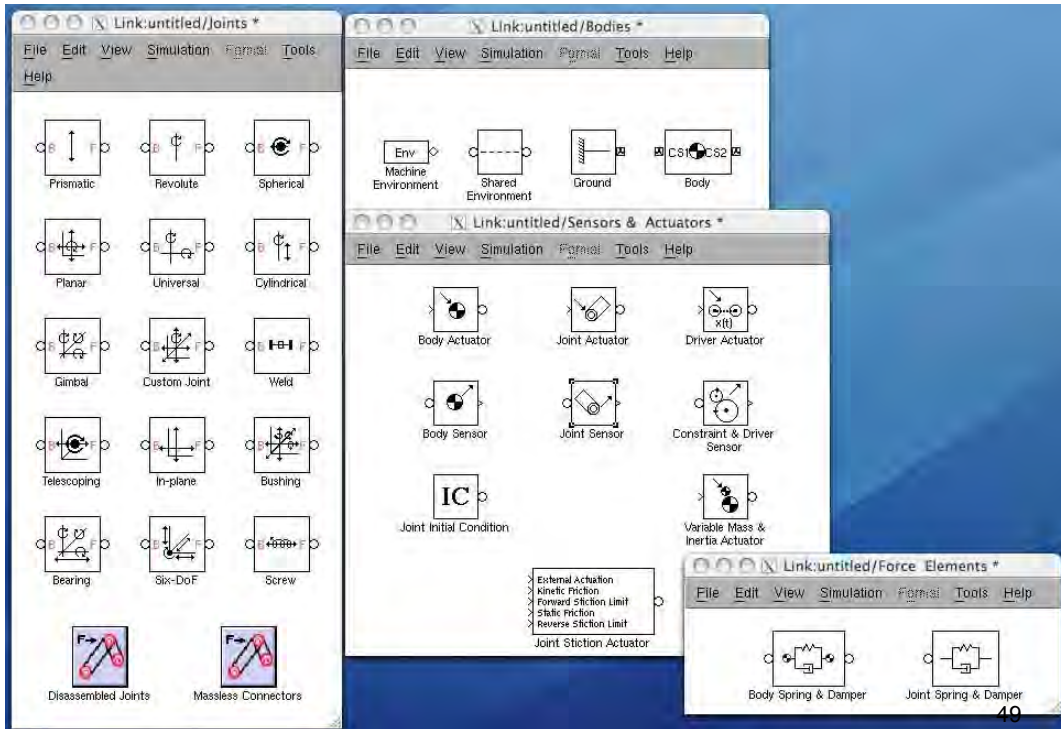
Block Description

Simscape/Foundation Library/Mechanical/Mechanical Sensors/Ideal Force Sensor: The block represents an ideal force sensor, that is, a device that converts a variable passing through the sensor into a control signal proportional to the force with a specified coefficient of proportionality. The sensor is ideal since it does not account for inertia, friction, delays, energy consumption, and so on.

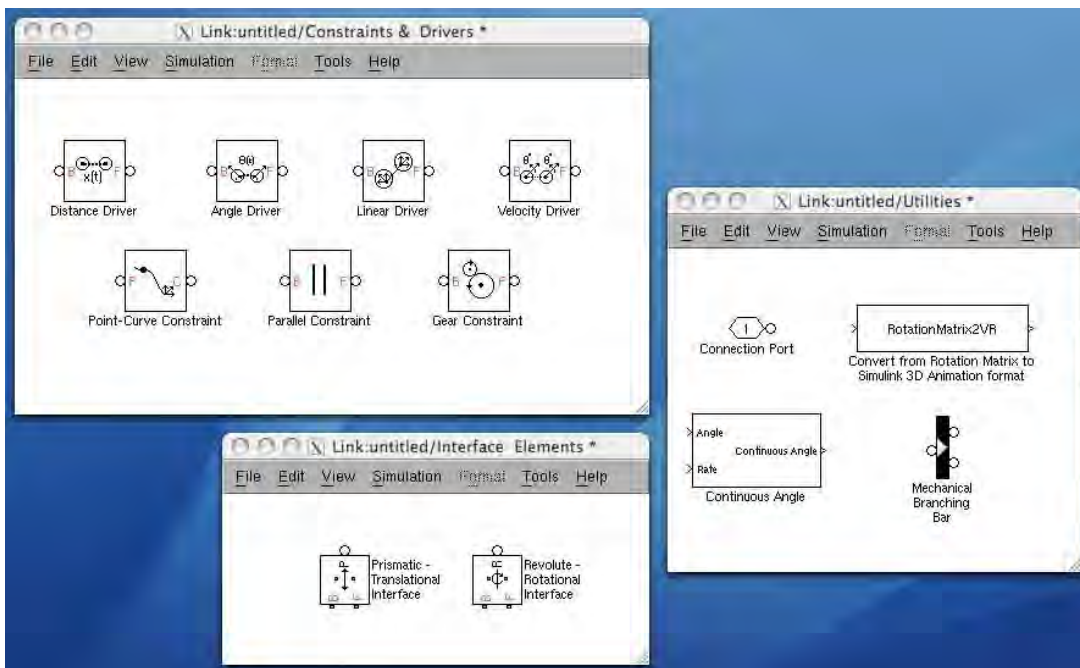
Showing: Simscape/Foundation Library/Mechanical/Mechanical Sensors

48

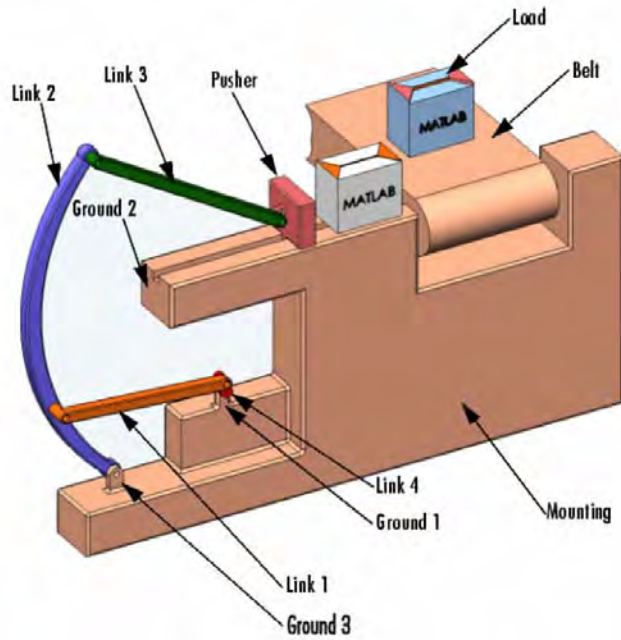
SimMechanics Library - 1



SimMechanics Library - 2

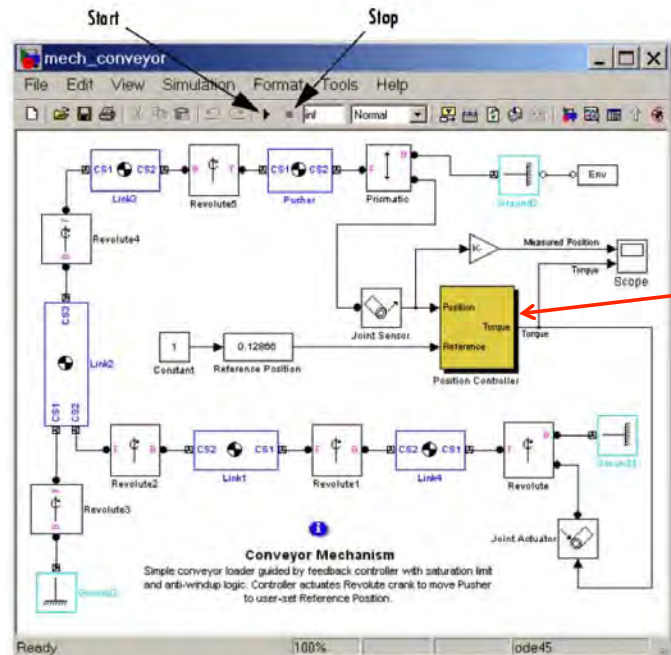


Conveyer-Loader Demonstration



51

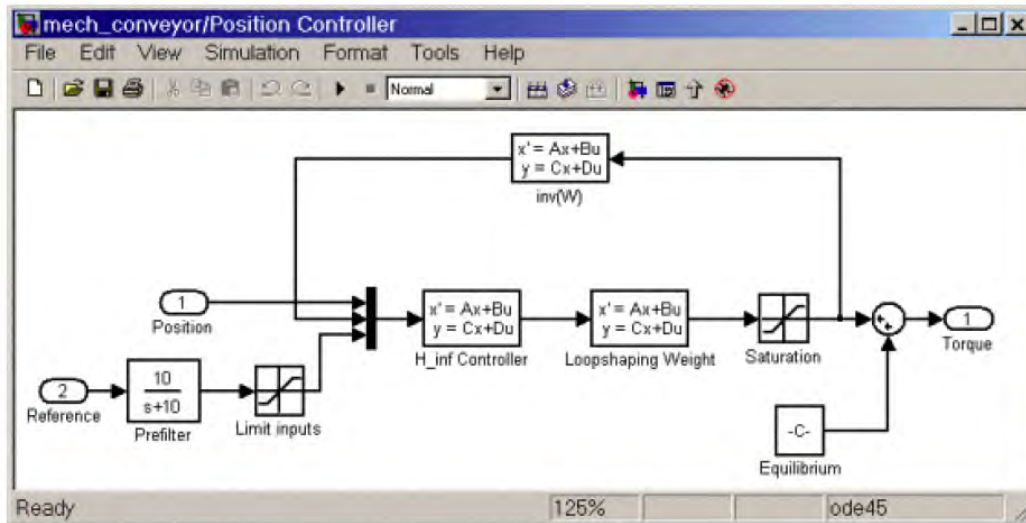
Conveyer-Loader Demonstration



Controller specified within box

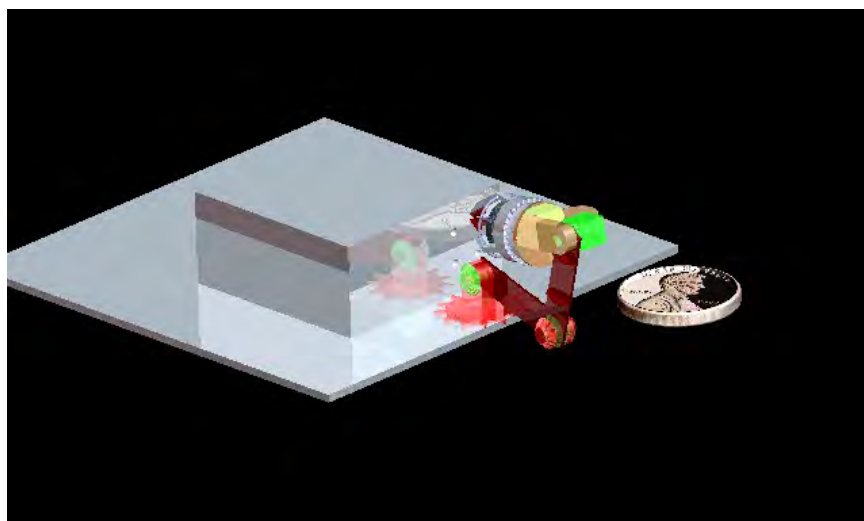
52

Position Controller for Conveyor-Loader Demonstration (Simulink)



53

Simulink Demonstration of 1-Inch Robot (MAE 345 Mid-Term Project, 2009)



54

***Next Time:
Articulated Robots***

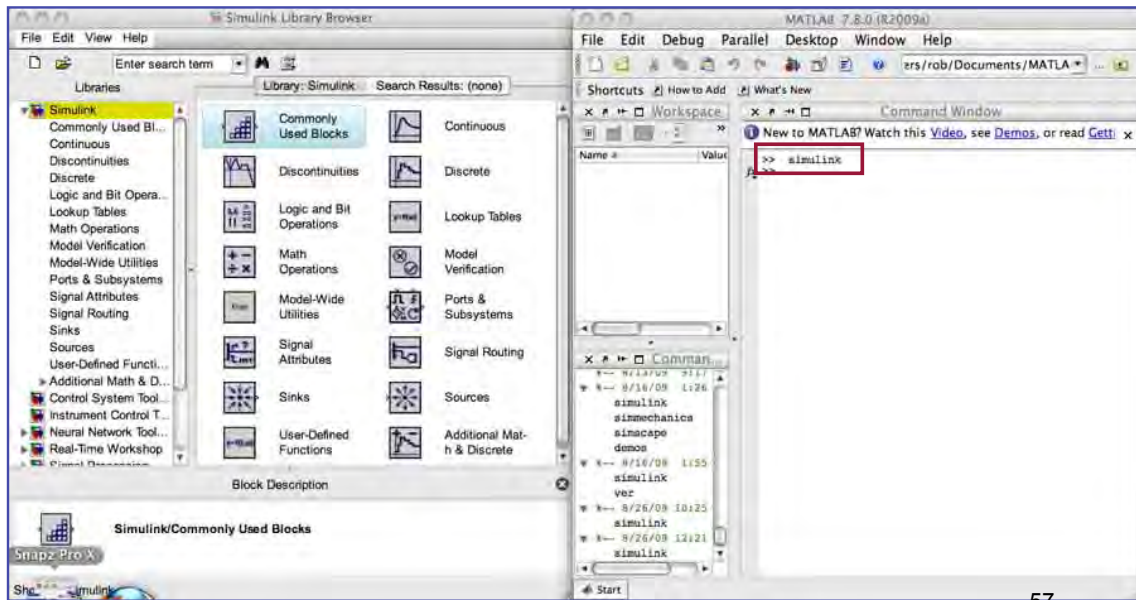
55

***Supplemental
Material***

56

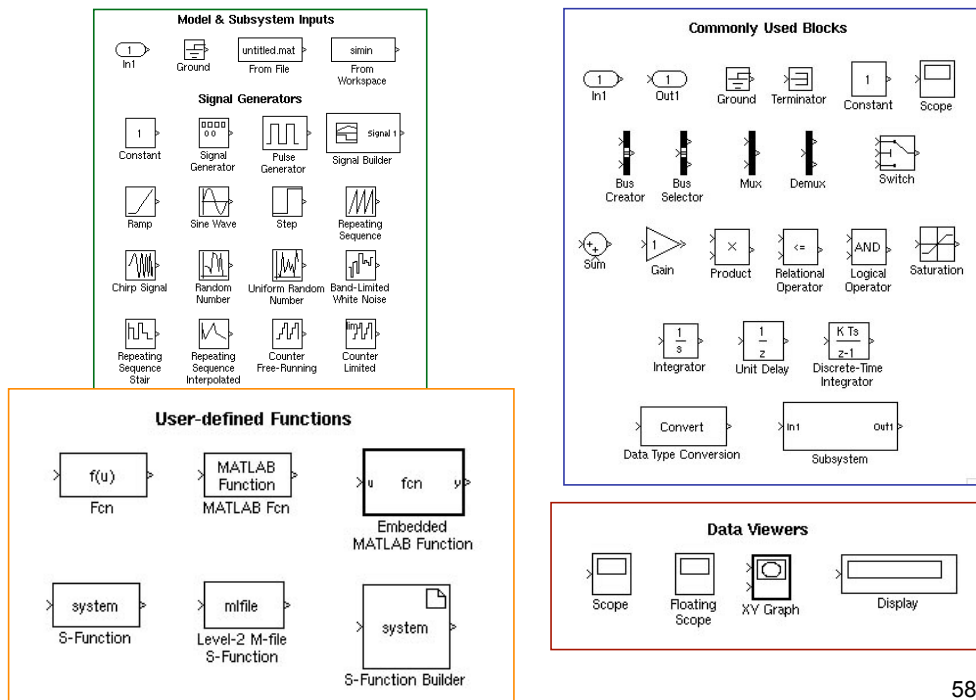
Simulink

Library of blocks, sources, and sinks



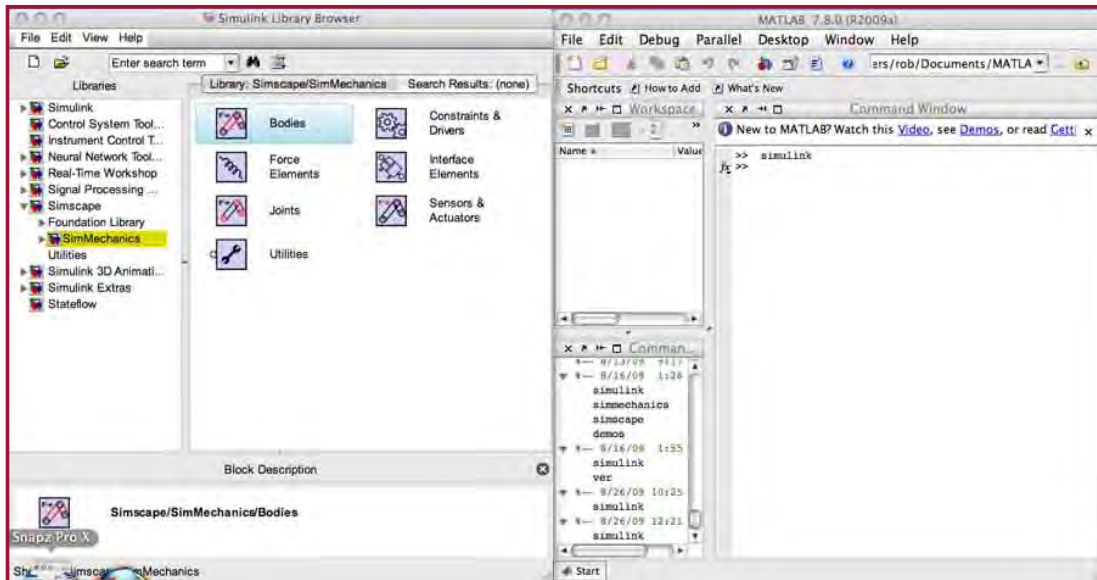
57

Simulink Blocks

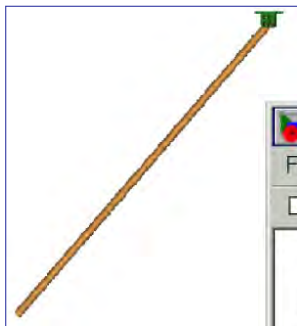


58

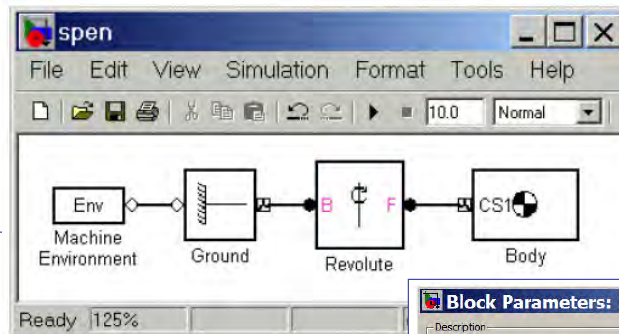
SimMechanics Called from Simulink



59



Simple Pendulum



Specifying Body Coordinate System

Position	Orientation	Visualization				
Show Port	Port Side	Name	Orientation Vector	Units	Relative CS	Specified Using Convention
<input checked="" type="checkbox"/>	Right	CG	[0 0 0]	deg	World	Euler X-Y-Z
<input checked="" type="checkbox"/>	Right	CS1	[0 0 0]	deg	World	Euler X-Y-Z

Block Parameters: Machine Environment

Description: Defines the mechanical simulation environment for the machine to which the block is connected: gravity, dimensionality, analysis mode, constraint solver type, tolerances, linearization, and visualization.

Parameters | Constraints | Linearization | Visualization

Analysis mode: Type of solution for machine's motion.
Tolerances: Maximum permissible misalignment of machine's joints.

Gravity vector: [0 -9.81 0] m/s²

Input gravity as signal

Machine dimensionality: Auto-detect

Analysis mode: Forward dynamics

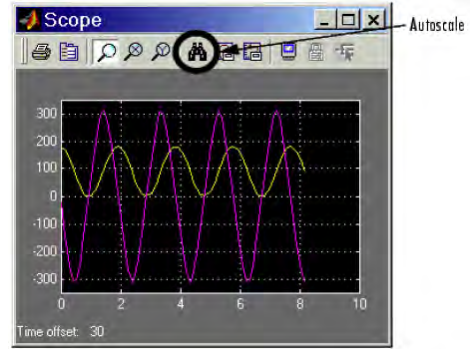
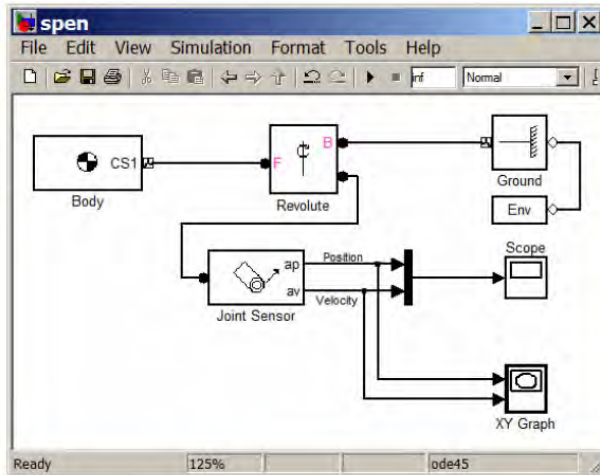
Linear assembly tolerance: 1e-3 m

Angular assembly tolerance: 1e-3 rad

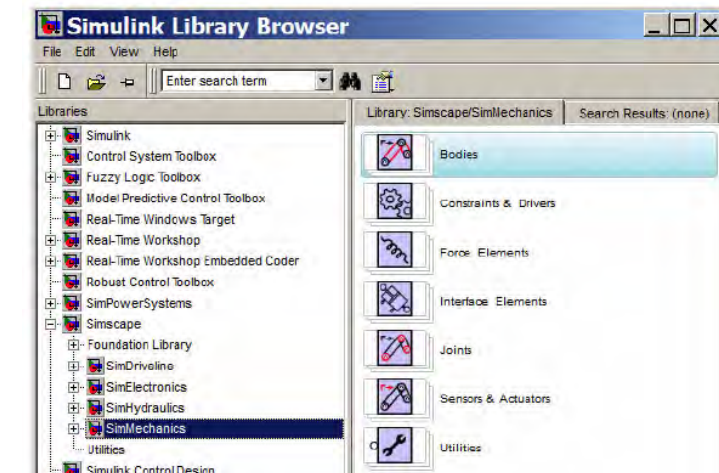
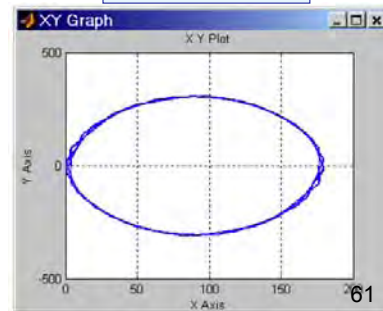
Configuration Parameters...

OK Cancel Help **60** Apply

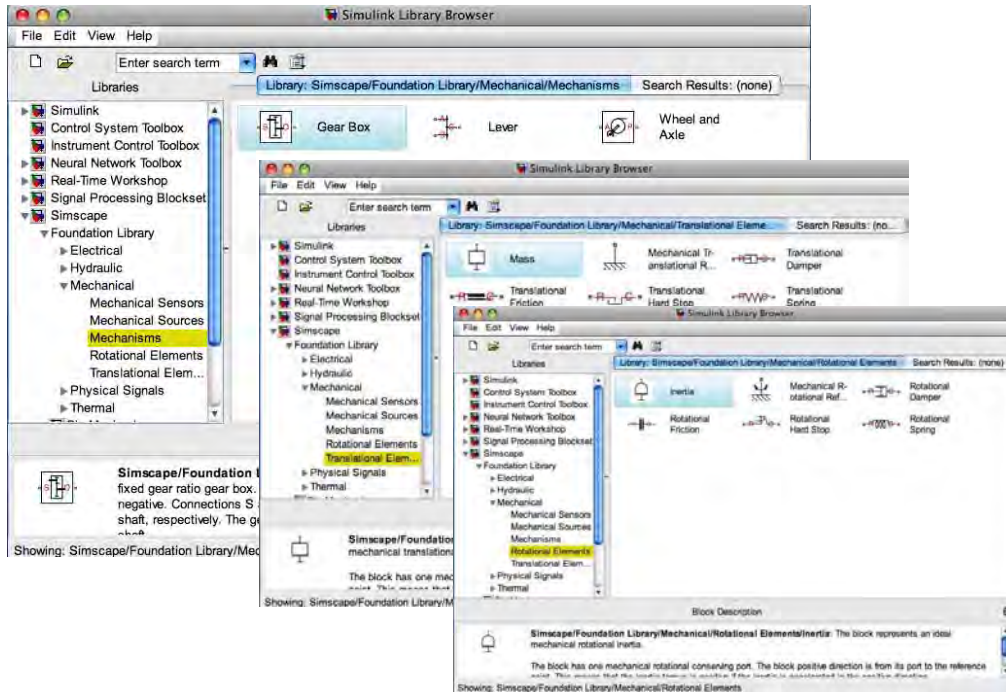
Simple Pendulum with Scope and XY Graph



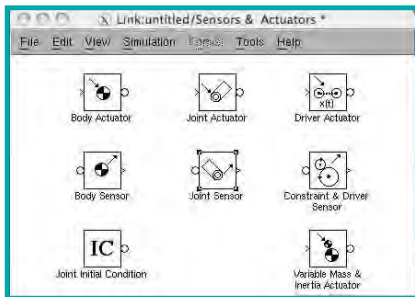
Phase-Plane Plot
(Rate vs. Displacement)



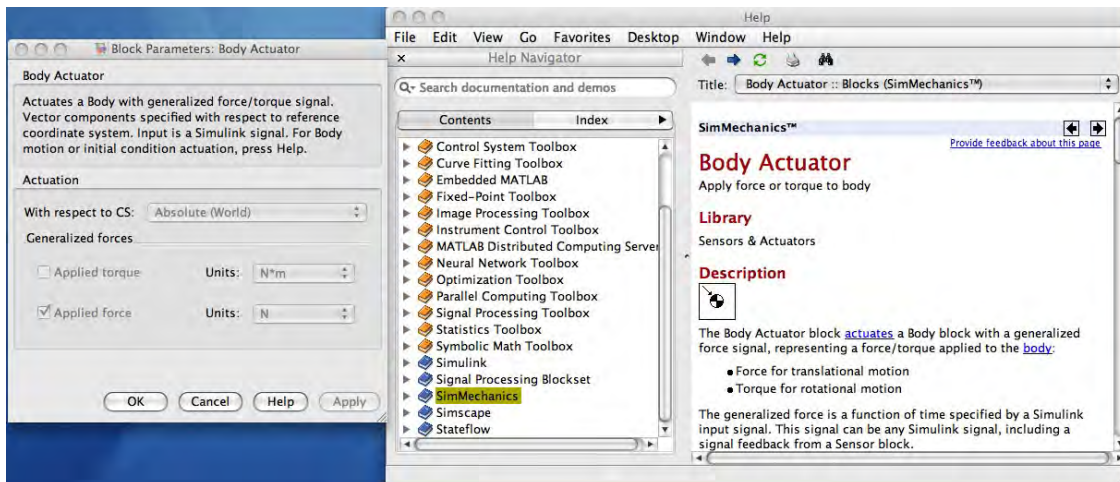
SimScape Mechanism Models



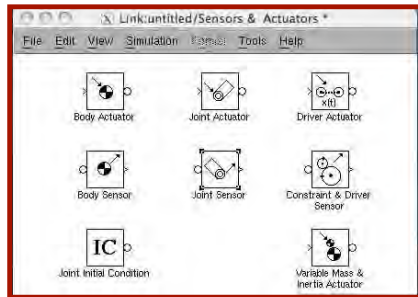
63



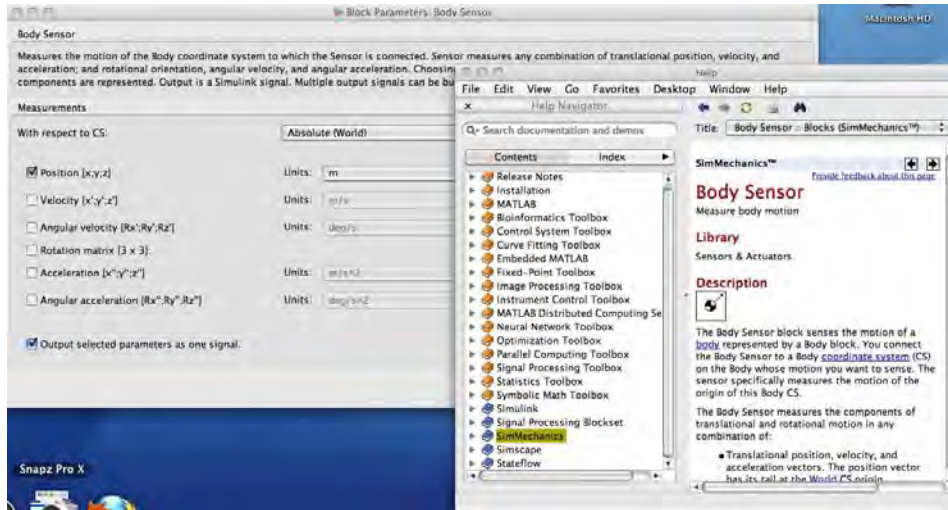
SimMechanics Body Actuator



64



SimMechanics Body Sensor



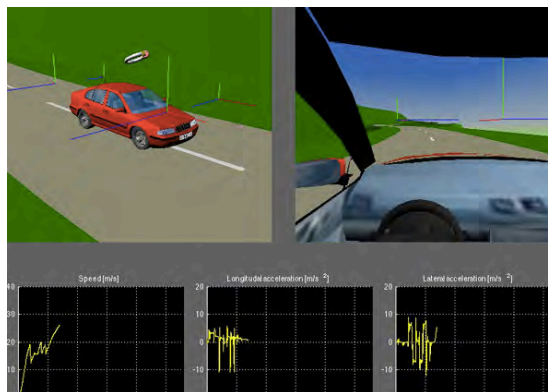
65

SimMechanics, Simulink 3D Animation 'Product Help' Demos

Robotic Manipulator



Vehicle Dynamics



<http://www.mathworks.com/products/simmechanics/demos.html>