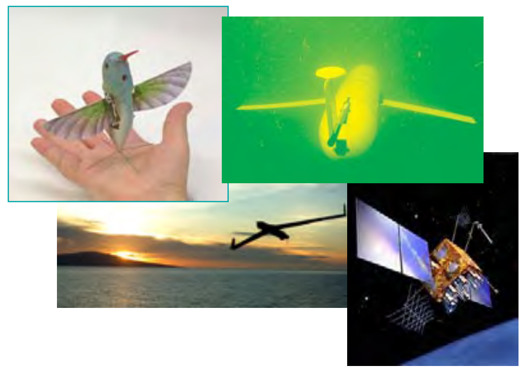
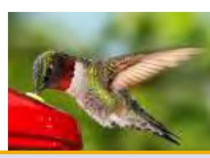
Flying and Swimming Robots

Robert Stengel
Robotics and Intelligent Systems MAE 345,
Princeton University, 2017

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



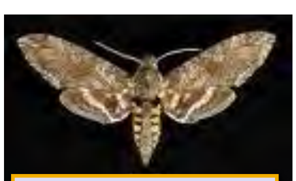
Bio-Inspiration for Flying



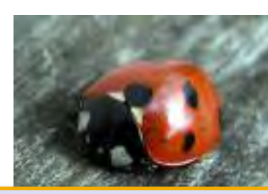
Hummingbird
http://www.youtube.com/watch?v=D8vjYTXglJw&feature=related

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

Birds Flying
http://www.youtube.com/watch?v=I5GbFgk-EPw



Moth Flying
https://www.youtube.com/watch?v=hD2BjAsvlbI



Lady Bug
http://www.youtube.com/watch?v=fjZobEZJYBc

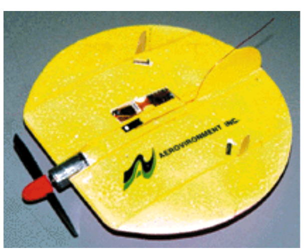
Biomimetic UAVs

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



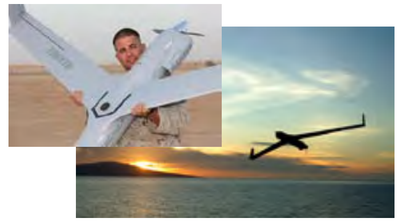
Uninhabited Air Vehicles (UAV)

Uninhabited Aircraft

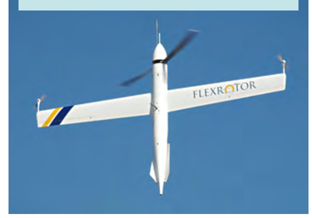
Tad McGeer, ‘79

Aerosonde
First UAV Transatlantic Crossing, 1998

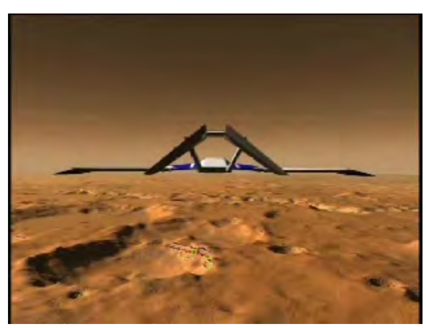
Boeing (InSitu)
ScanEagle



Aerovel *Flexrotor*

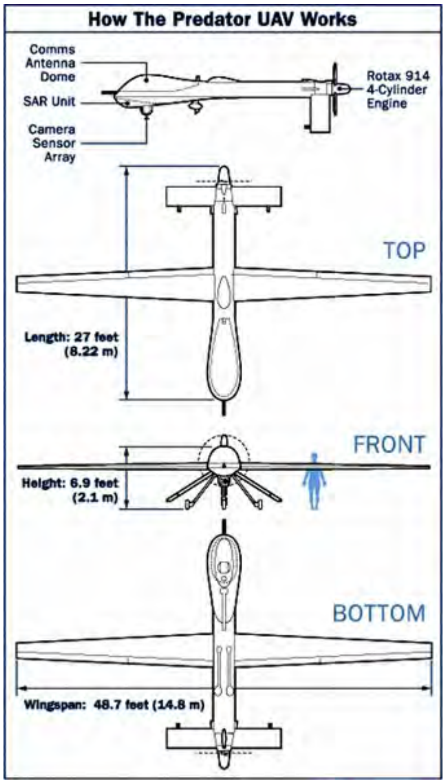


Mars Aerial Regional-Scale Environmental Survey (*ARES*) Research Airplane Concept, ~2008



https://www.youtube.com/watch?v=8YutbpJuFiI

https://www.youtube.com/watch?v=wAOTOmGFs5M



Uninhabited Aircraft

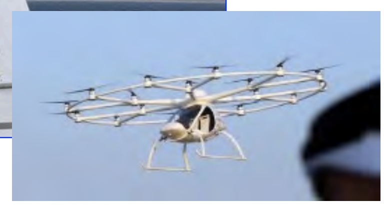
MQ-9 Reaper
http://www.youtube.com/watch?v=kSpOYZR0kIA

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

Multi-Copters

Autonomous Air Taxi
Volocopter

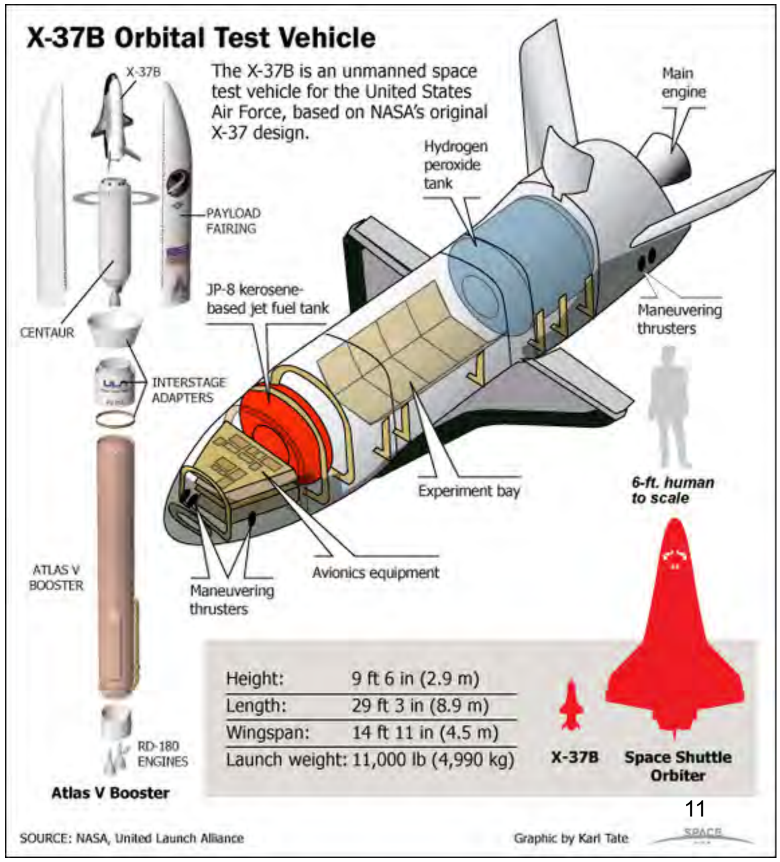
18 rotors
30-min flying time



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



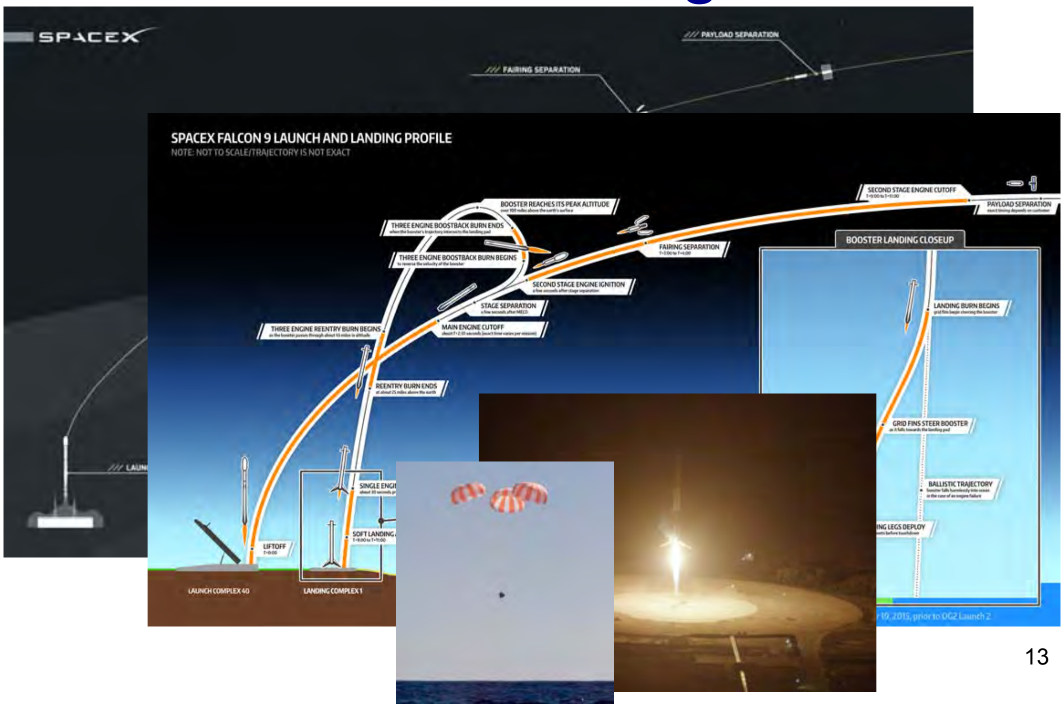
Expendable (Rocket) Launch Vehicles

Current space launch vehicles are largely autonomous

Atlas V

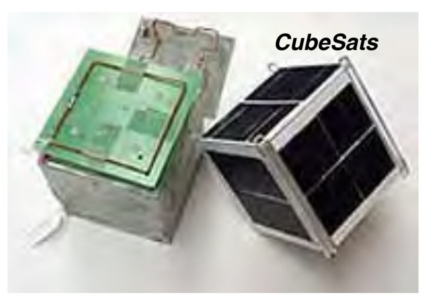
http://www.youtube.com/watch?v=KxQbex7LJwg

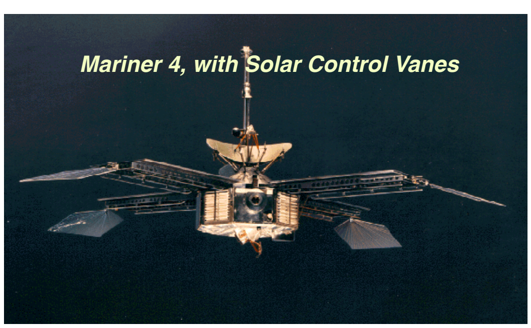
Reusable Launch/Reentry Vehicles
Falcon 9/Dragon



Uninhabited Spacecraft

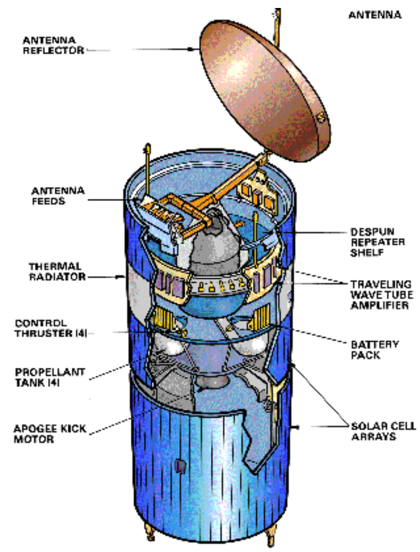

Giotto


CubeSats


Mariner 4, with Solar Control Vanes

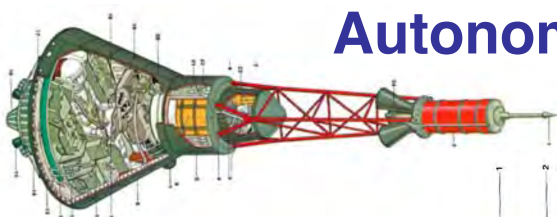

GPS III Satellite

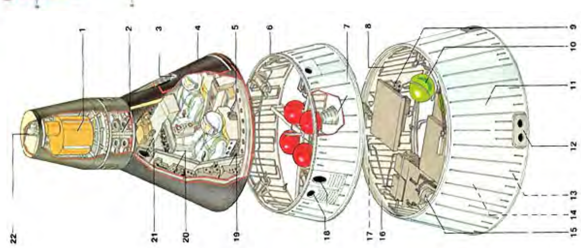
Uninhabited Spacecraft

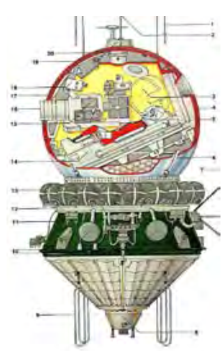


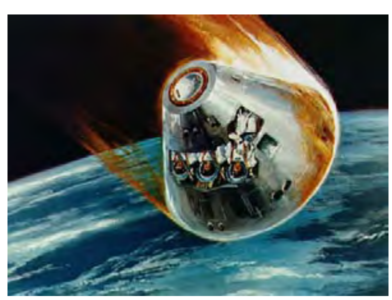
BOEING 376
SPACECRAFT CONFIGURATION

Manned Re-Entry Vehicles Largely Autonomous



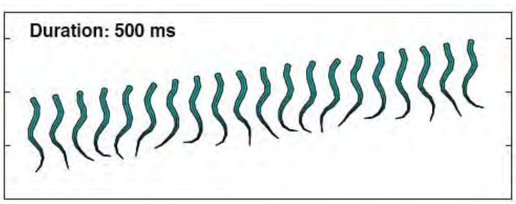


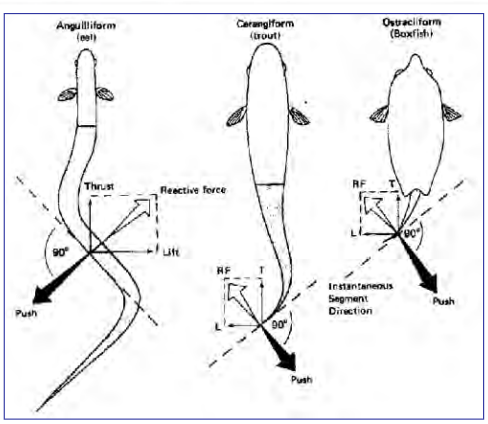




Undersea Robots

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

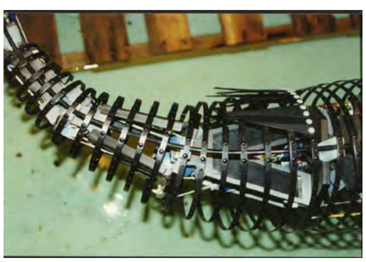
Thunniform locomotion
High-speed long-distance swimmers, e.g. tuna, shark
Virtually all lateral movement in the tail
Tail itself is large and crescent-shaped

Swimming

- Lift, drag, and vorticity
- Schooling behavior

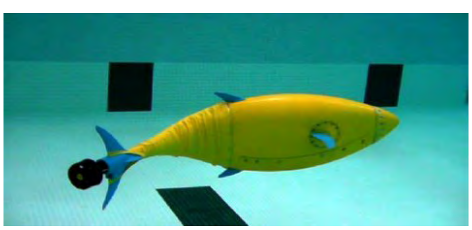
Human Swimming
http://www.youtube.com/watch?v=CIzBaSiWdRA

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



Autonomous Underwater Vehicles

RoboTuna (Olin/MIT)



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

RoboLobster (Northeastern)

Autonomous Submarines

Autonomous Benthic Explorer

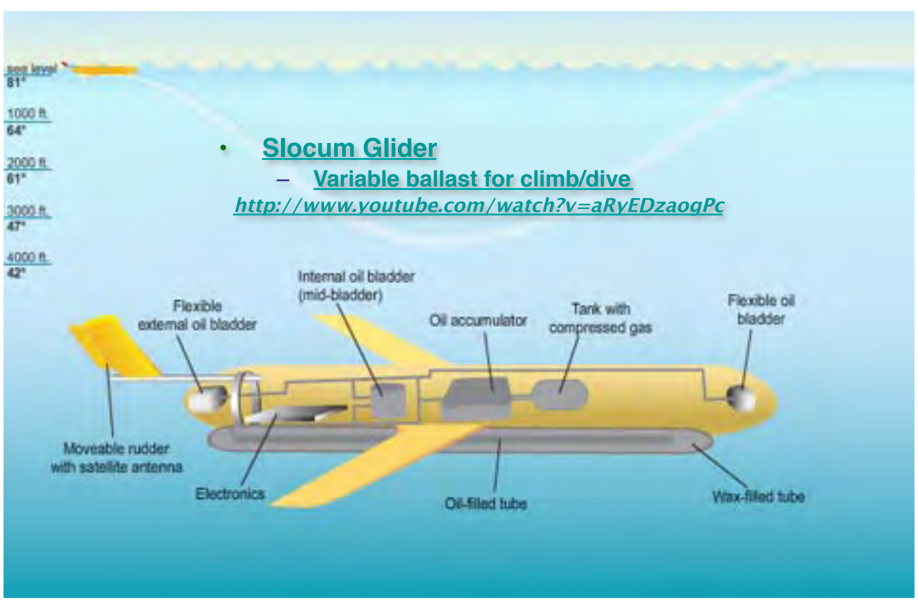
VPI concept

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

Autonomous Underwater Gliders



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

Avoiding the Euler Angle Singularity

Inverse Transformation for Euler-Angle Rates

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

Euler-angle rates from body-axis rates

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

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

Alternatives to Euler angles

1) Direction cosine (rotation) matrix

2) Quaternions

Propagation of direction cosine matrix (9 parameters)

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

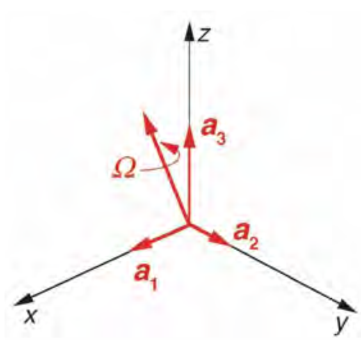
Initialize with Euler Angles

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

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

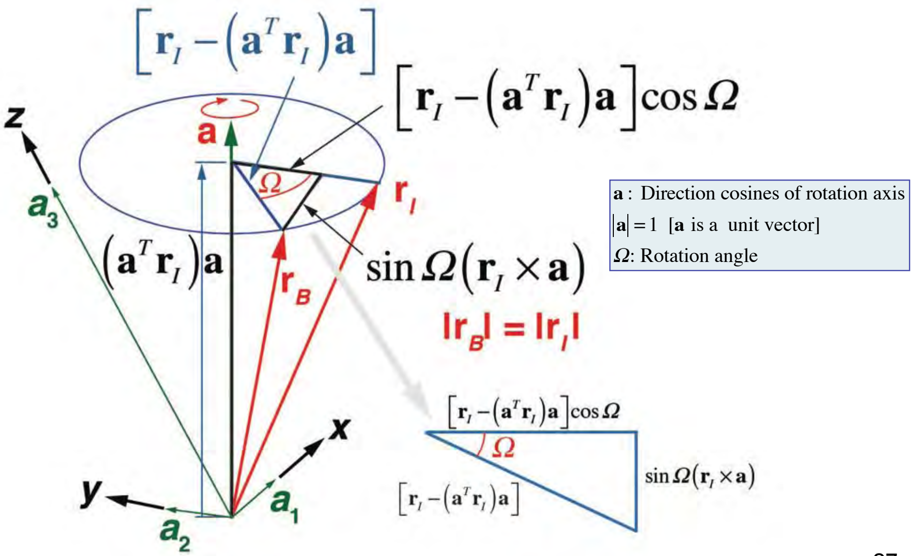
Propagation of quaternion vector: single rotation from inertial to body frame (4 parameters)

- 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, **a**: a_1, a_2, and a_3)
  - Magnitude of the rotation angle, Ω, rad



Begin with Euler Rotation of a Vector

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



Development of Theorem

Defined vector is given a different orientation

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

Transformation involves addition of 3 vectors

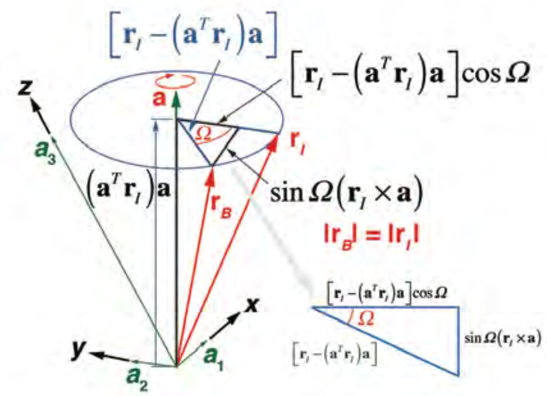
$\left(\mathbf{a}^T\mathbf{r}_I\right)\mathbf{a}$	Along axis of rotation
$\left[\mathbf{r}_I - \left(\mathbf{a}^T\mathbf{r}_I\right)\mathbf{a}\right]\cos\Omega$	$\perp$ to $\mathbf{a}$ and through $\mathbf{r}_I$
$\sin\Omega\left(\mathbf{r}_I \times \mathbf{a}\right)$	$\perp$ to $\mathbf{a}$ and $\mathbf{r}_I$

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

Development of Theorem

Combine terms
Reverse cross-product order

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



Rotation Matrix Derived from Euler's Formula

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

Identity

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

Cancel like terms

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

Rotation matrix

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

Quaternion Derived from Euler Rotation Angle and Orientation

Quaternion vector

4 parameters based on Euler's rotation formula

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

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

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

Rotation Matrix Expressed with Quaternion

Euler's formula expressed with quaternion

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

Terms of rotation matrix from quaternion elements

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

Initial Quaternion Expressed from Elements of Rotation Matrix

Initialize 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 $q_4 \neq 0$

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

Quaternion Vector Kinematics

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

(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 q(t_k)

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

Rigid Body Equations of Motion Using Quaternion

$$\dot{\mathbf{r}}_I(t) = \mathbf{H}_B^I\left[\mathbf{q}(t)\right]\mathbf{v}_B(t)$$

$$\dot{\mathbf{v}}_B(t) = \frac{1}{m}\mathbf{f}_B(t) - \tilde{\boldsymbol{\omega}}_B(t)\mathbf{v}_B(t)$$

$$\dot{\mathbf{q}}(t) = \frac{1}{2}\begin{bmatrix} 0 & r(t) & -q(t) & p(t) \\ -r(t) & 0 & p(t) & q(t) \\ q(t) & -p(t) & 0 & r(t) \\ -p(t) & -q(t) & -r(t) & 0 \end{bmatrix}\mathbf{q}(t)$$

$$\dot{\boldsymbol{\omega}}_B(t) = \mathbb{I}_B^{-1}\left[\mathbf{m}_B(t) - \tilde{\boldsymbol{\omega}}_B(t)\mathbb{I}_B\boldsymbol{\omega}_B(t)\right]$$

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

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)

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

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \text{atan2}\left\{2\left(q_1 q_4 + q_2 q_3\right), \left[1 - 2\left(q_1^2 + q_2^2\right)\right]\right\} \\ \sin^{-1}\left[2\left(q_2 q_4 - q_1 q_3\right)\right] \\ \text{atan2}\left\{2\left(q_3 q_4 + q_1 q_2\right), \left[1 - 2\left(q_2^2 + q_3^2\right)\right]\right\} \end{bmatrix}$$

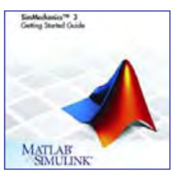
Solving Math Problems Computationally

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

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



MATLAB Models of Dynamic Systems

Systems are described by *instructions*

Main Script

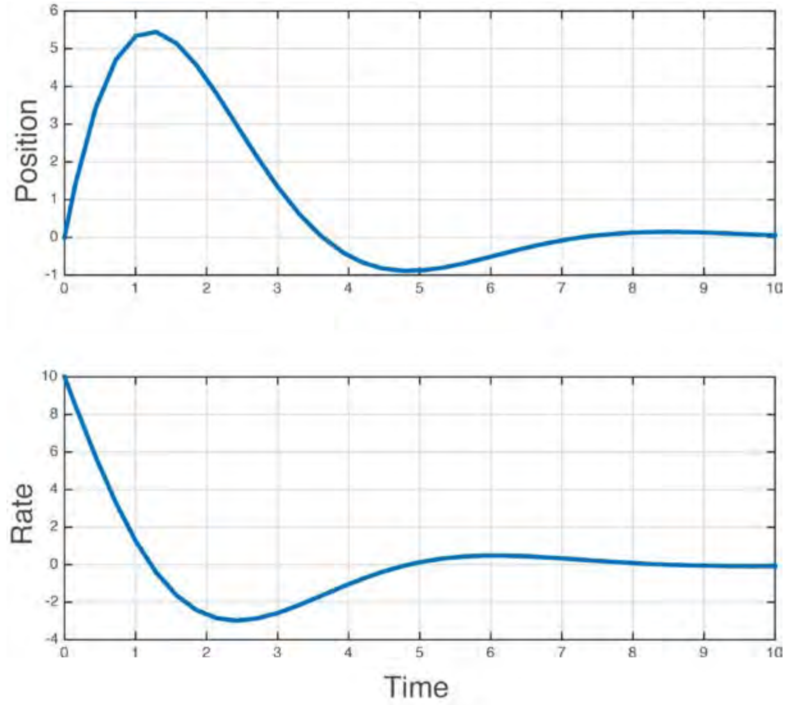
```
% Linear 2nd-Order Example
   clear
   tspan  =  [0 10];
   xo = [0, 10];
   [t,x] = ode23('Lin',tspan,xo);

   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)];
```

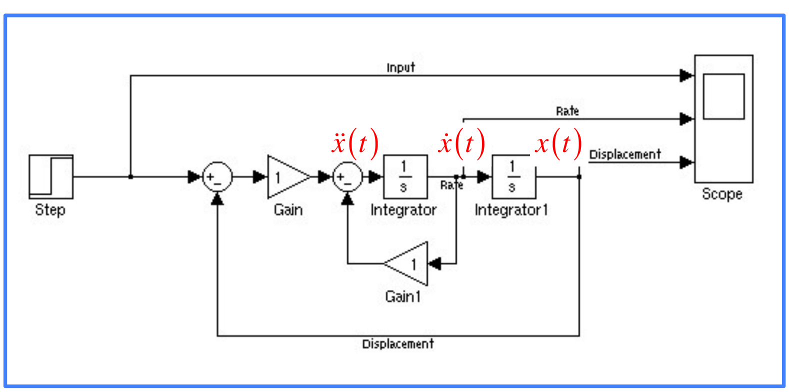
MATLAB Initial-Condition Output



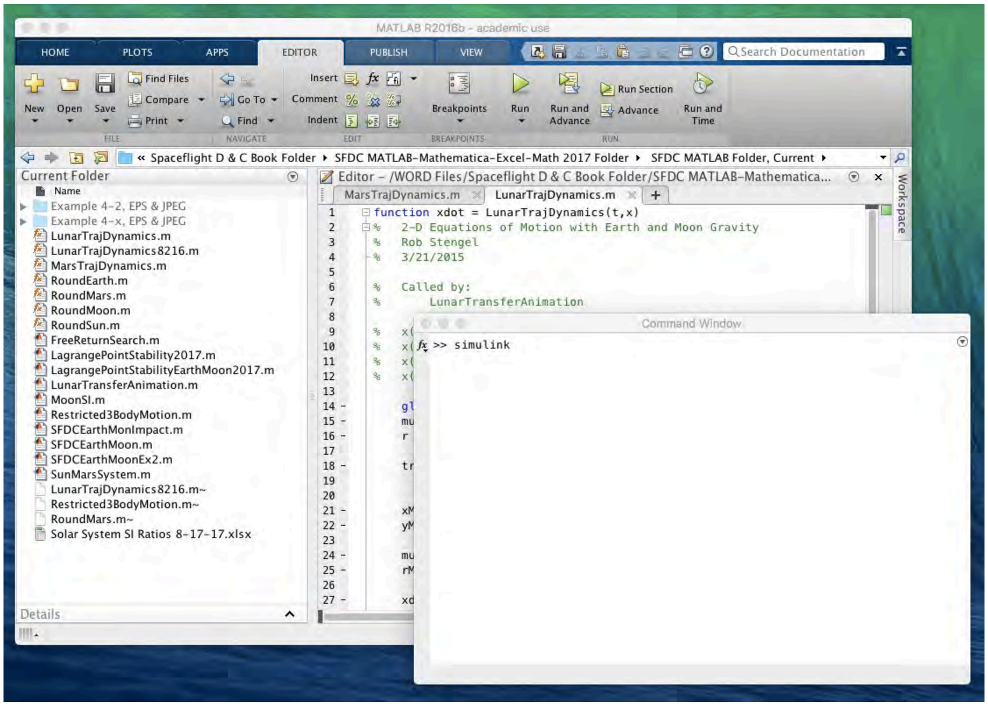
Simulink Models of Dynamic Systems

Systems are described by *block diagrams*

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

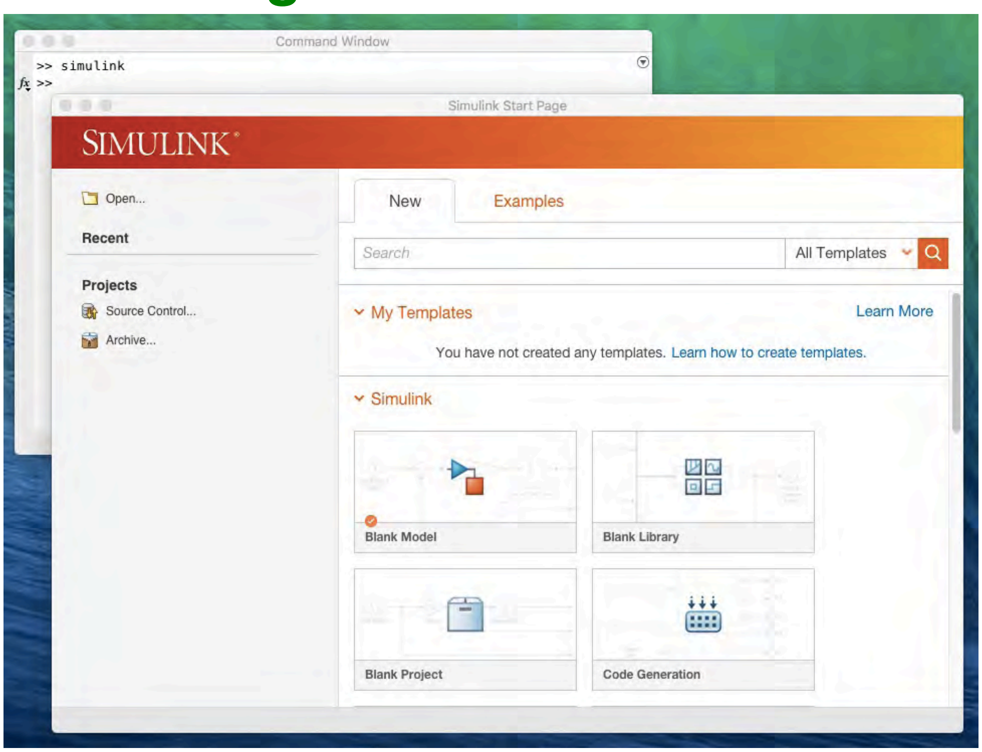


Accessing Simulink from MATLAB



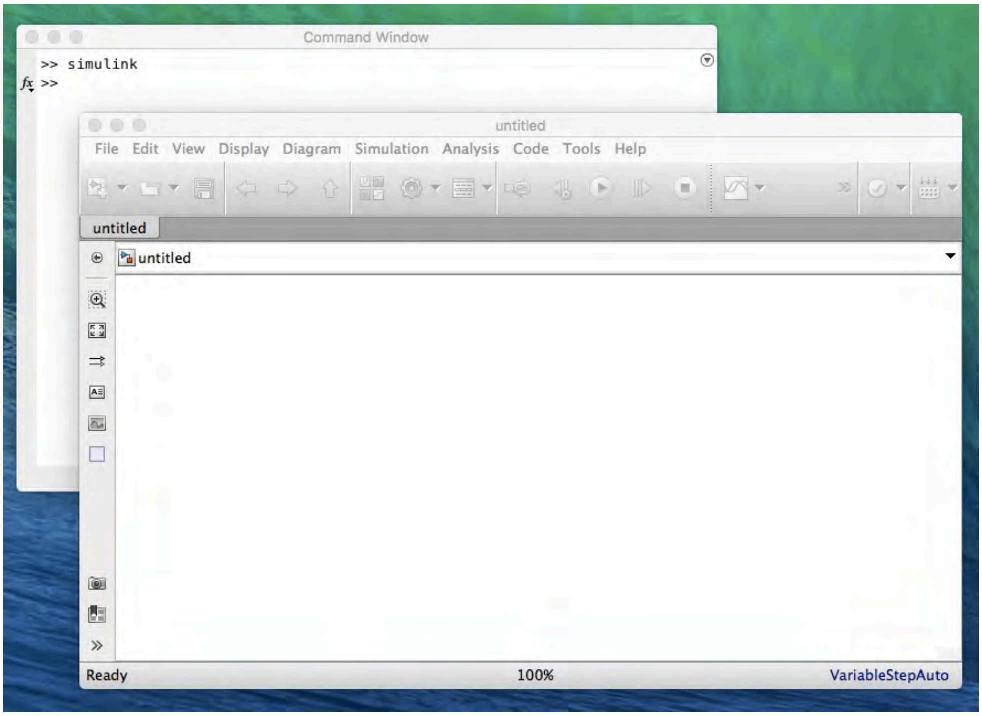
41

Accessing Simulink from MATLAB

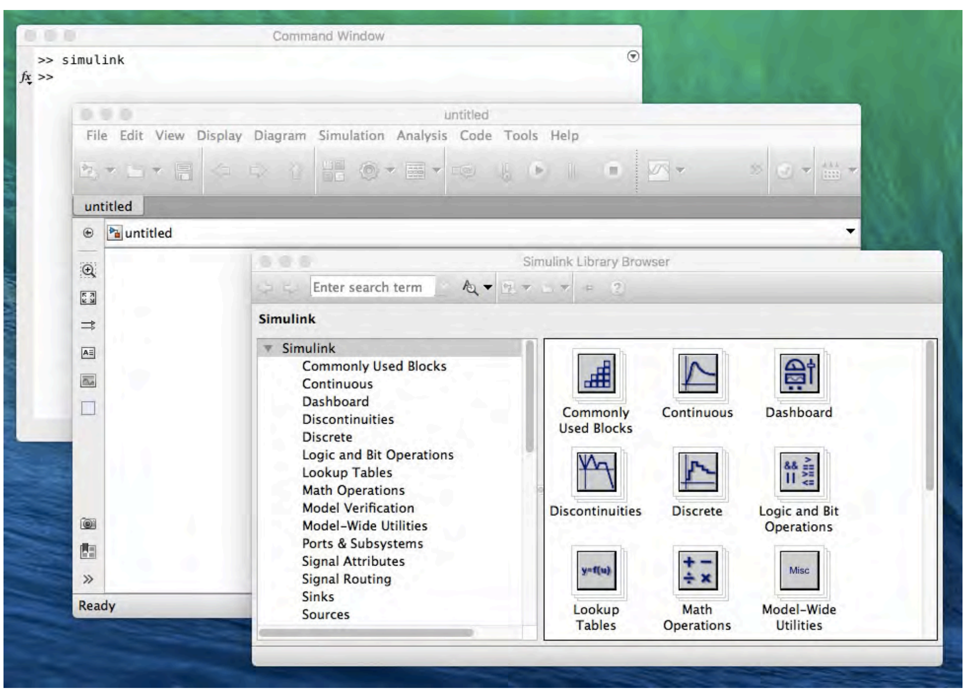


42

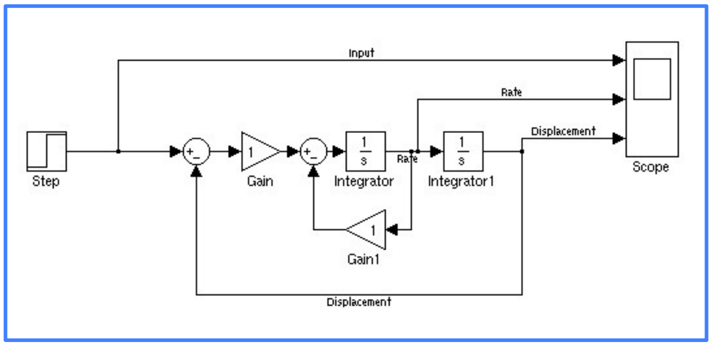
Open Simulink Blank Model

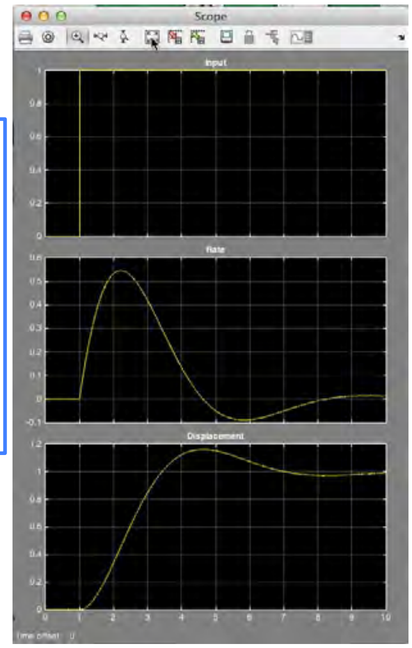


Open Simulink Library Browser for Function Blocks

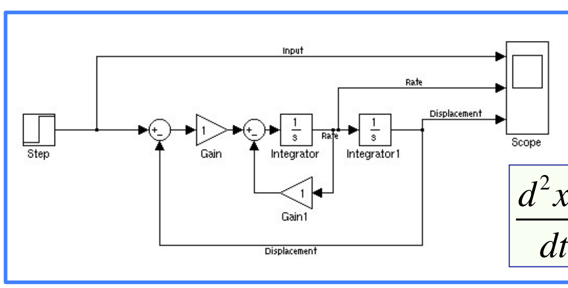


Simulink Step Response





Alternative Simulink Models of 2nd-Order Systems



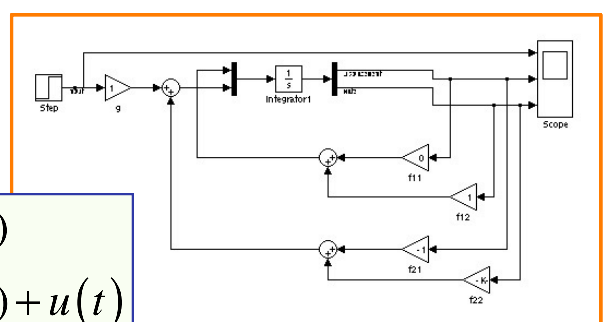
Single 2nd-order model, with step input and damping

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

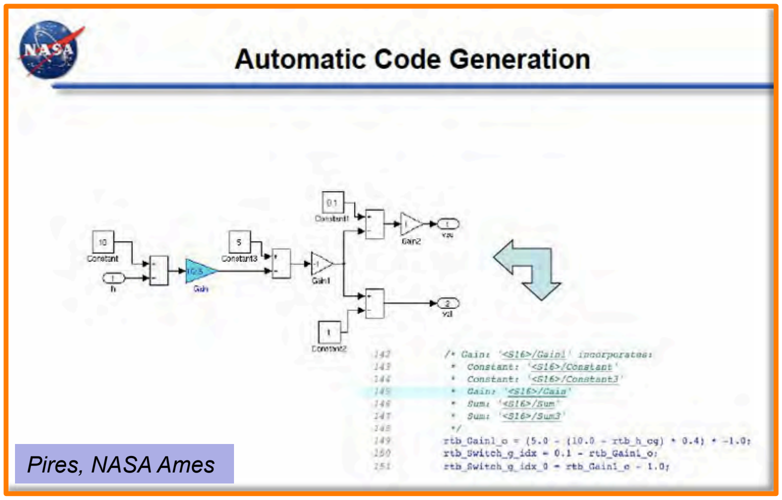
State-space model (two 1st-order equations), with step input and damping

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

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

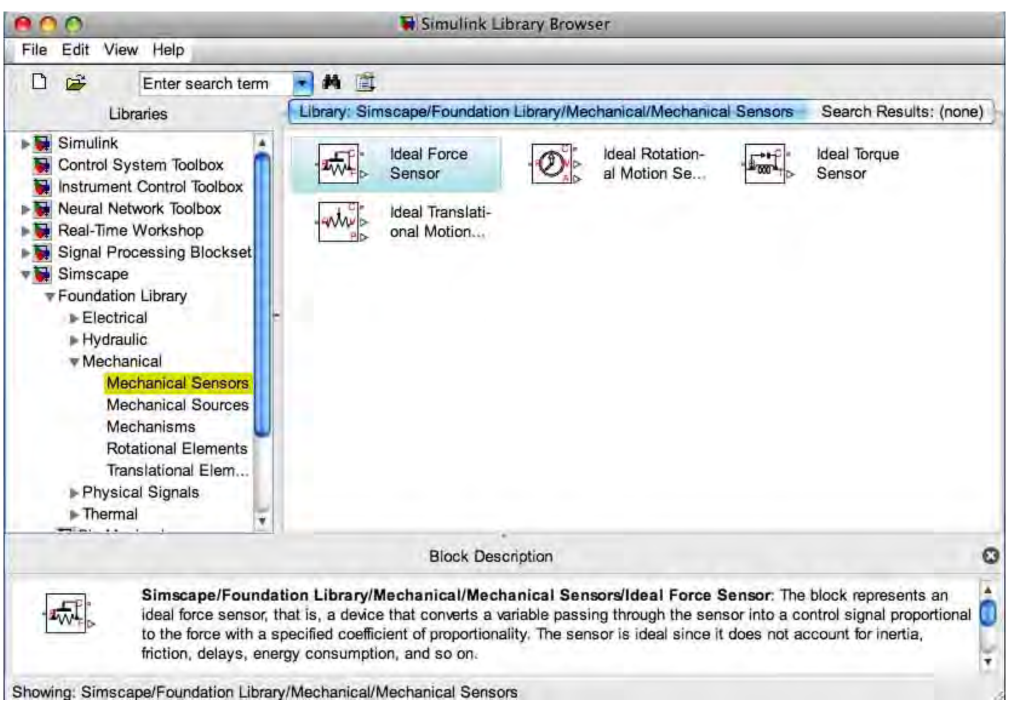


Simulink Autocoding

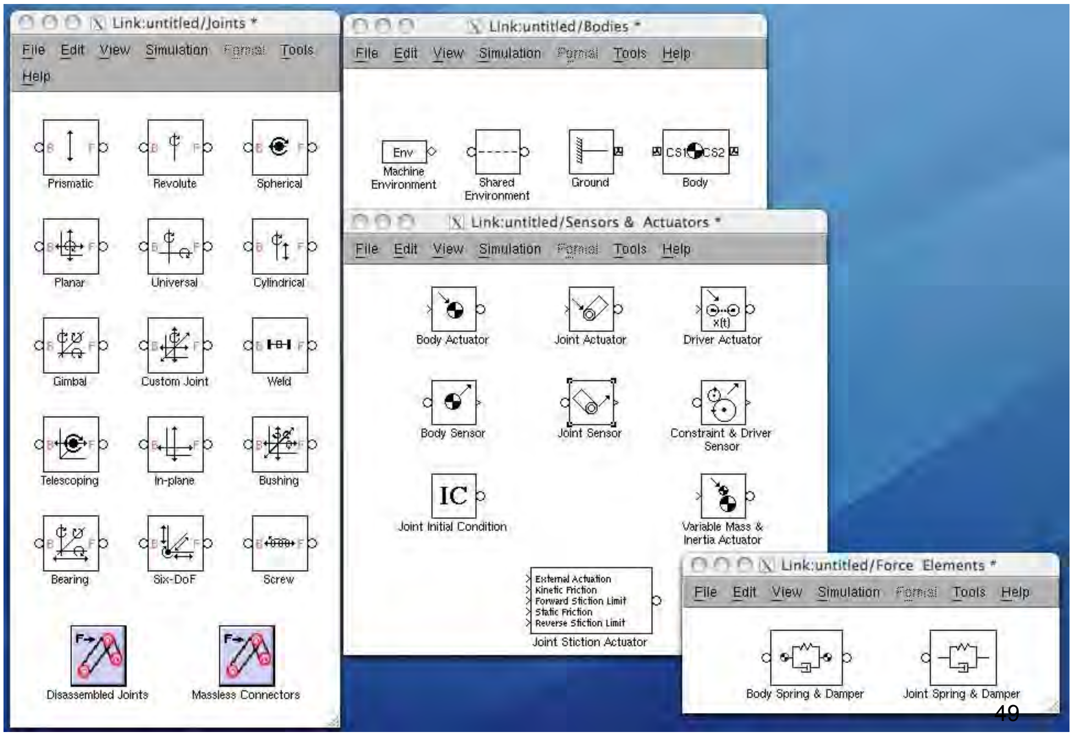


- **Graphic modeling of dynamic systems**
- **Library of functions**
- **Generation of C and C++ code**

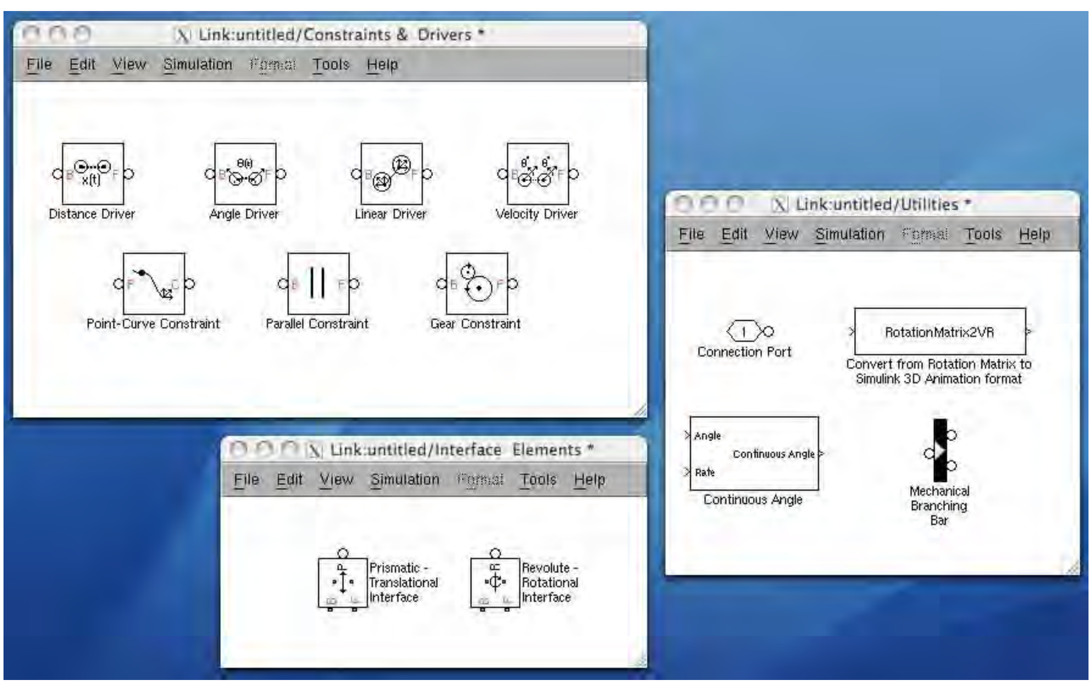
SimMechanics is Mechanical Subset of *SimScape* Library



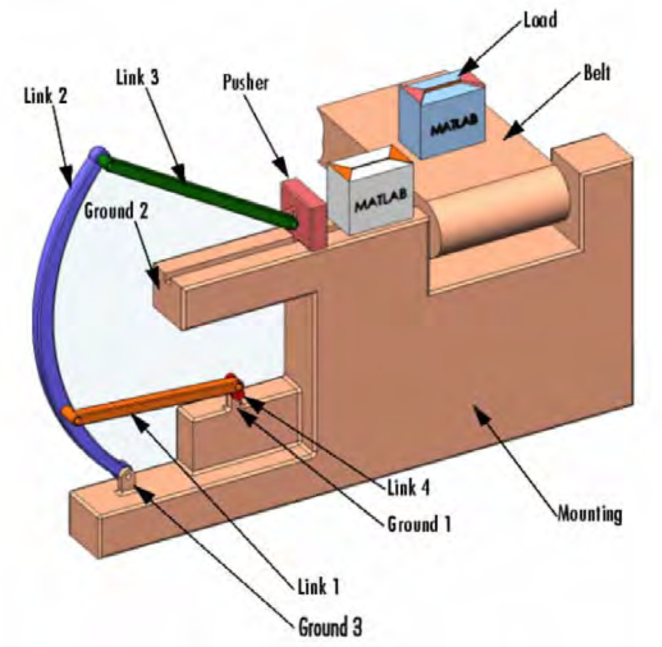
SimMechanics Library - 1



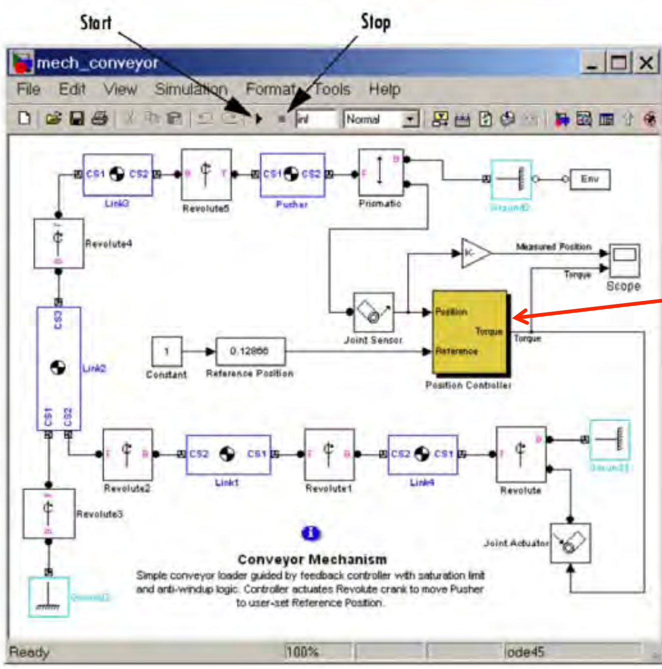
SimMechanics Library - 2



Conveyer-Loader Demonstration

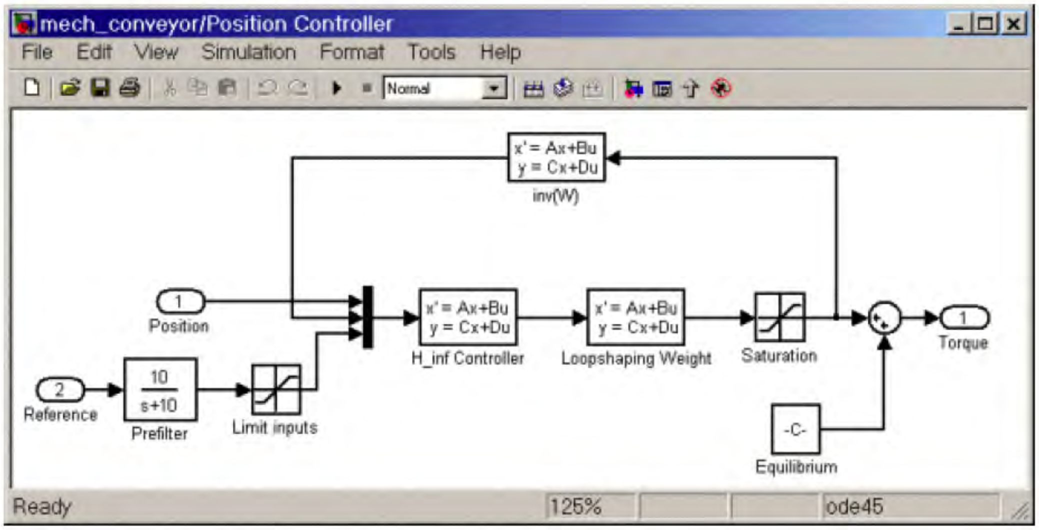


Conveyer-Loader Demonstration

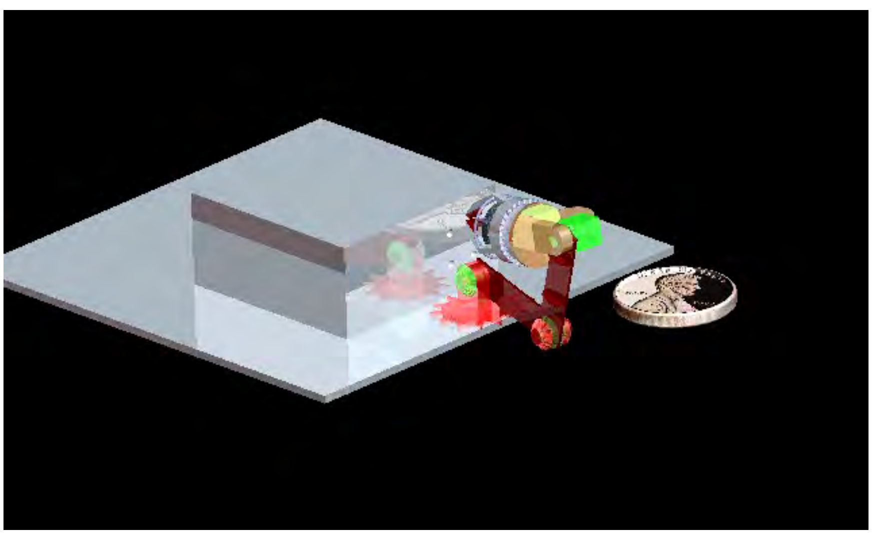


Controller specified within box

Position Controller for Conveyor-Loader Demonstration
(Simulink)



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

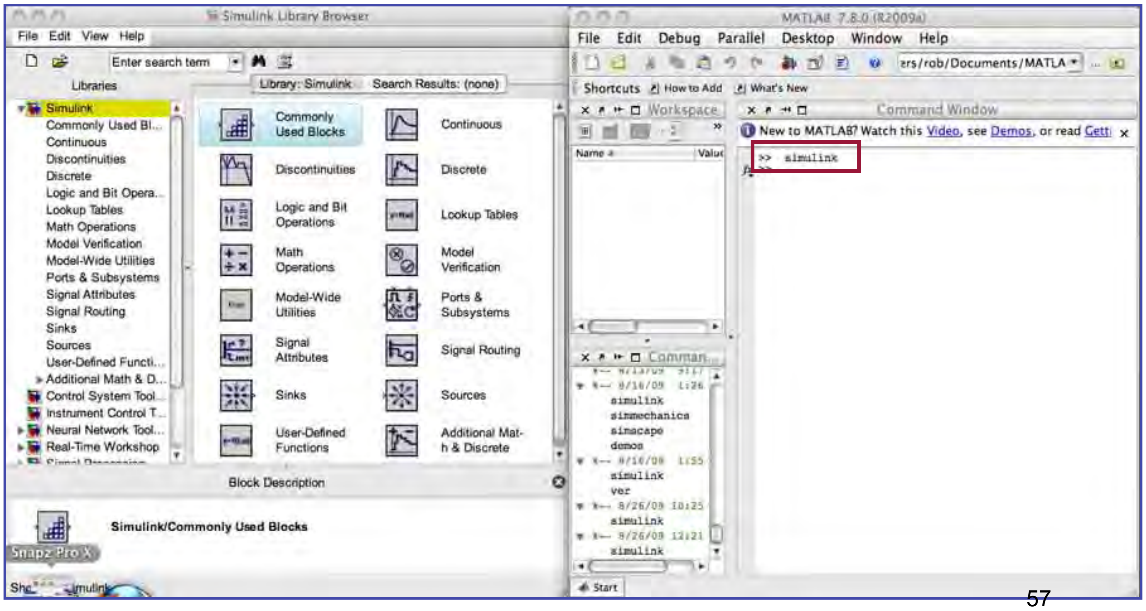


Next Time: Articulated Robots

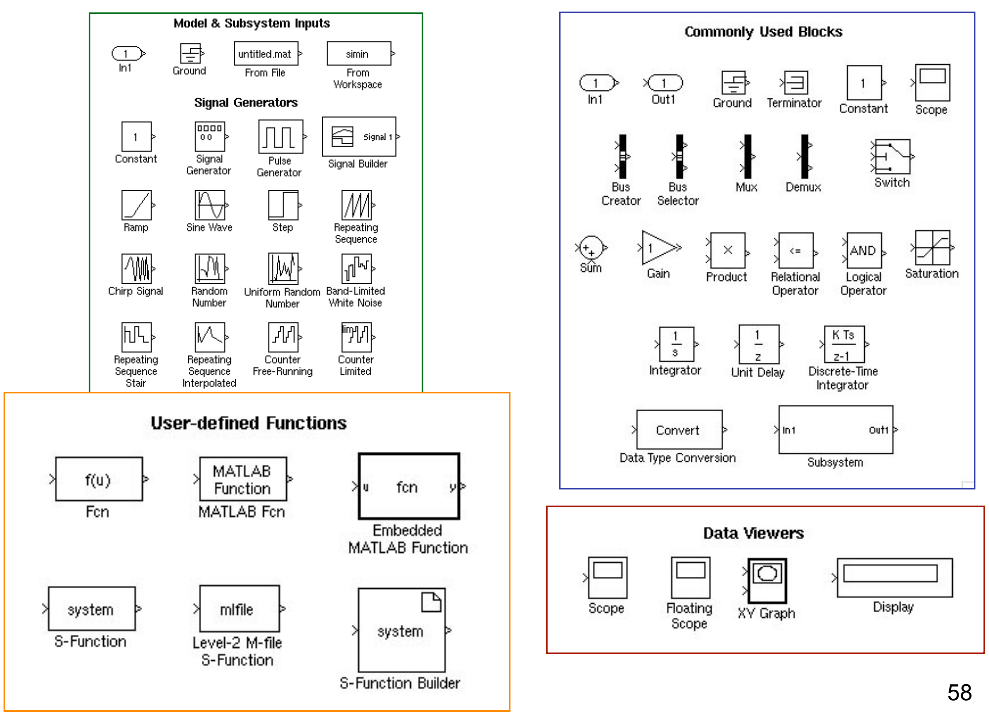
Supplemental Material

Simulink

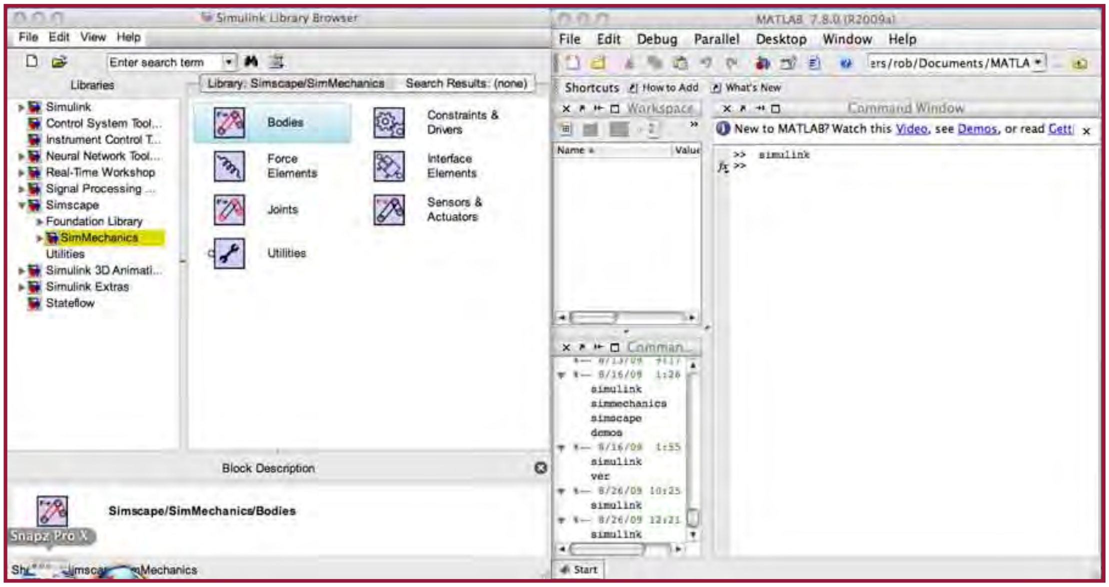
Library of blocks, sources, and sinks



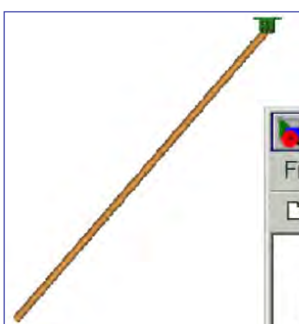
Simulink Blocks

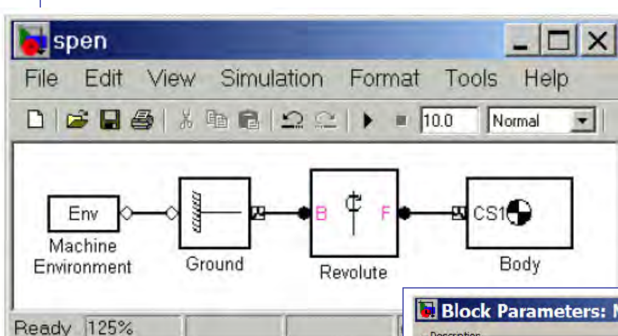


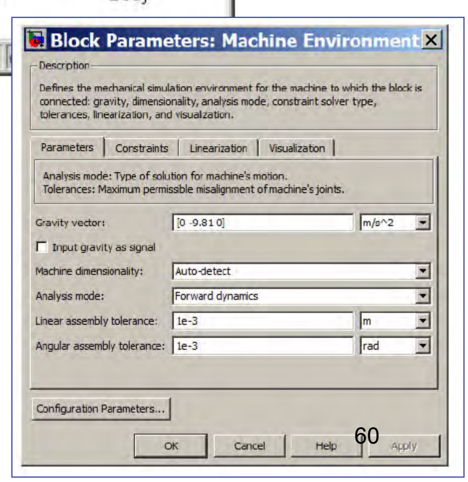
SimMechanics
Called from Simulink



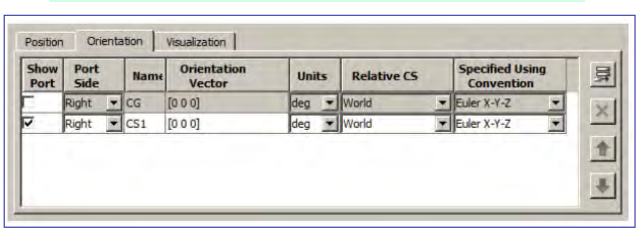
Simple Pendulum







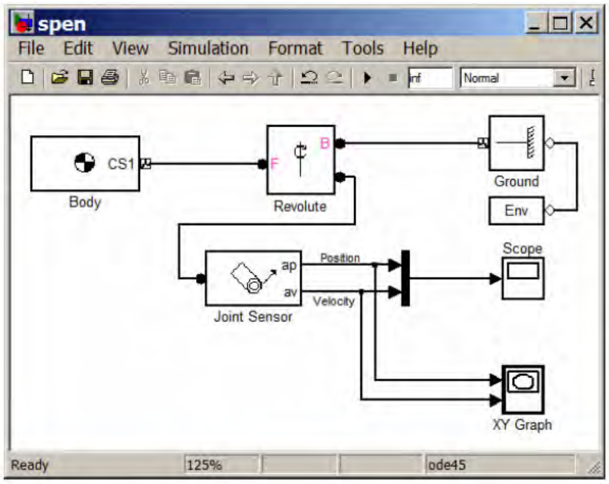
Specifying Body Coordinate System

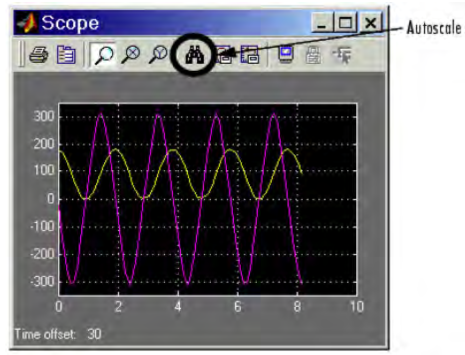


Position | Orientation | Visualization

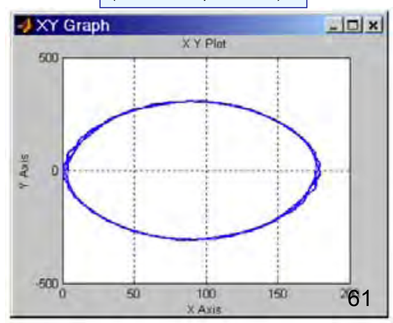
Show Port	Port Side	Name	Orientation Vector	Units	Relative CS	Specified Using Convention
☐	Right	CG	[0 0 0]	deg	World	Euler X-Y-Z
☑	Right	CS1	[0 0 0]	deg	World	Euler X-Y-Z

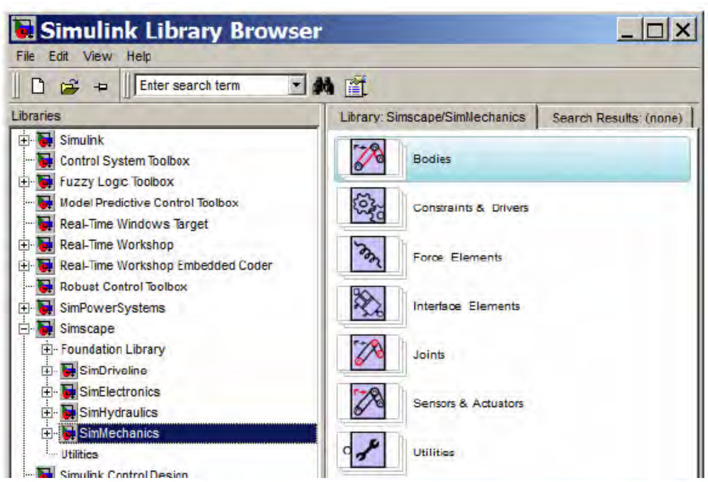
Simple Pendulum with Scope and XY Graph



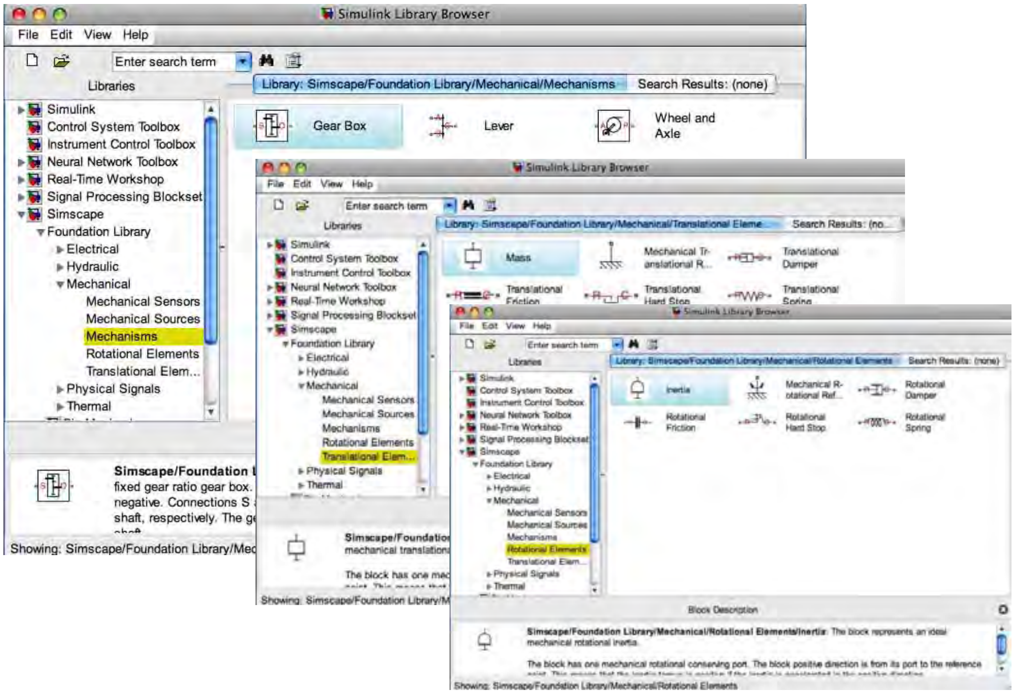


Phase-Plane Plot
(Rate vs. Displacement)

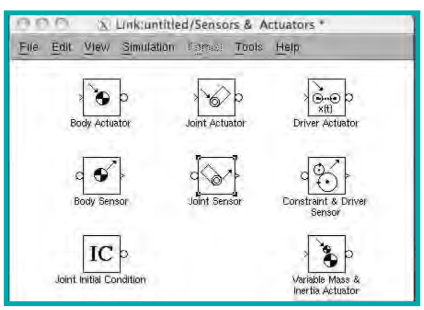


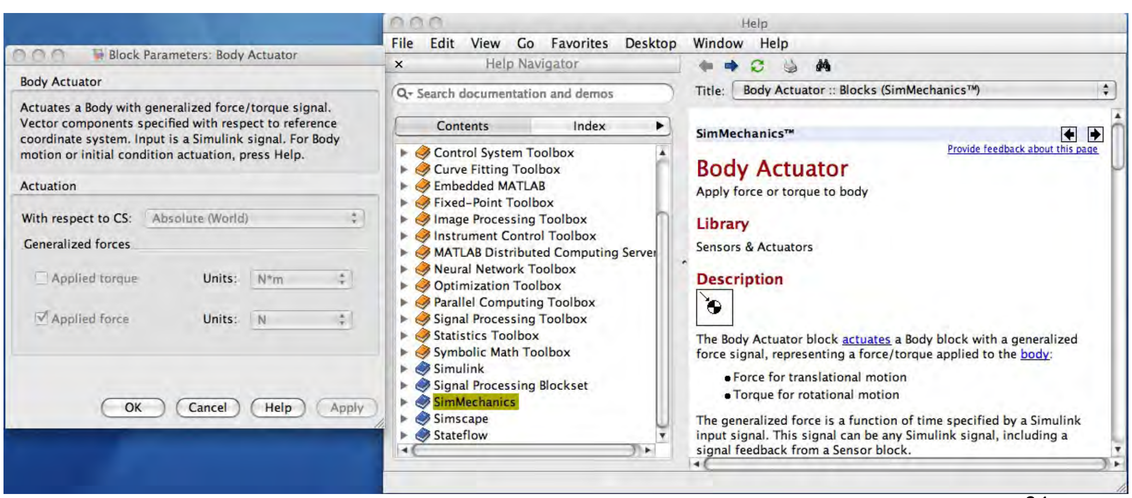


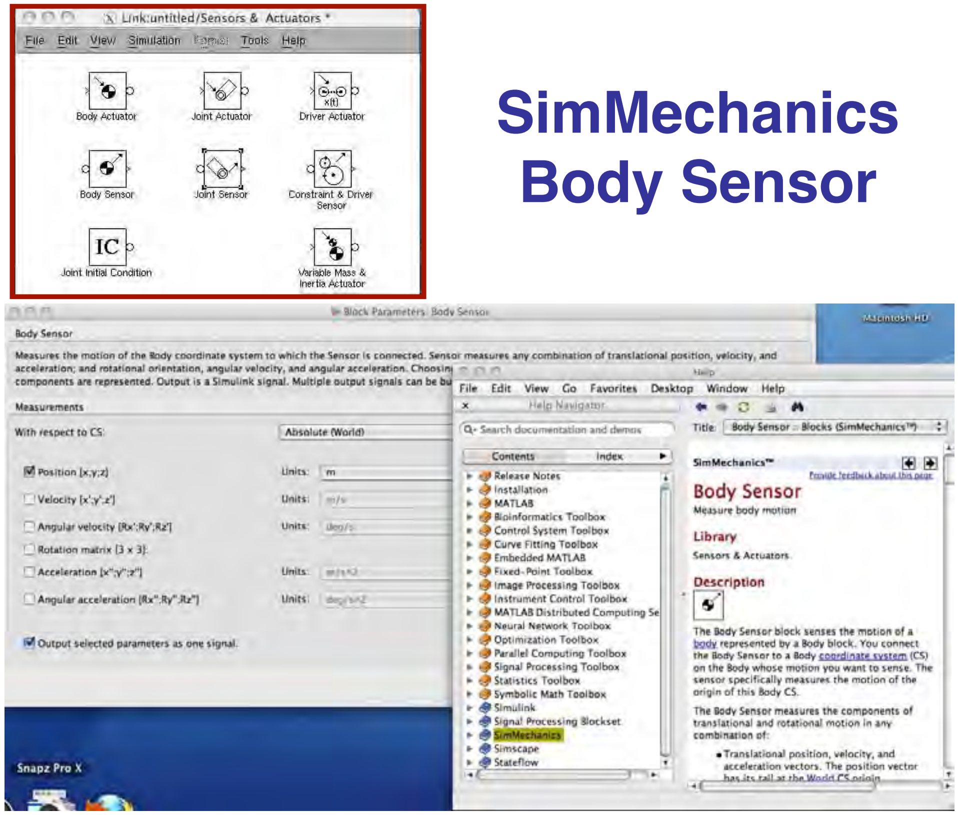
SimScape Mechanism Models



SimMechanics Body Actuator



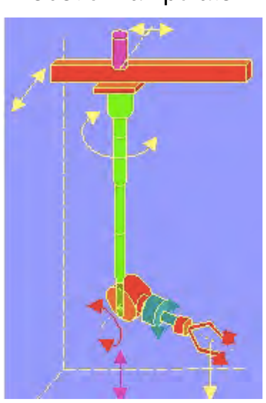




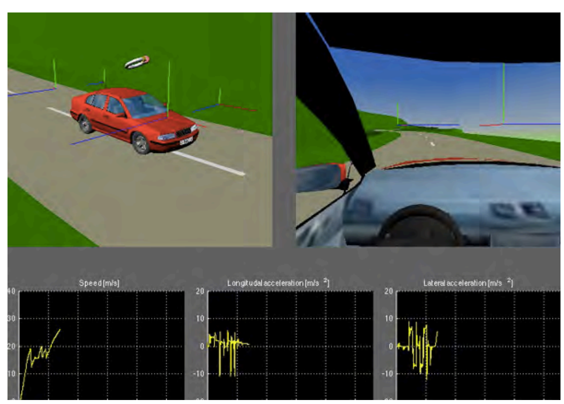
SimMechanics Body Sensor

SimMechanics, Simulink 3D Animation 'Product Help' Demos

Robotic Manipulator



Vehicle Dynamics



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