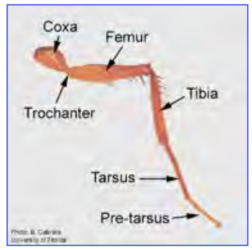
Mobile Robots, Position, and Orientation

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

- **Ground Vehicles**
  - Legged creatures
  - Wheeled and tracked robots
  - Other
- **Frames of Reference and Pose**
- **Translation and Rotation**
- **Homogeneous Transformation**
- **Math Review**

Legged Creatures

Walking, Running, and Jumping

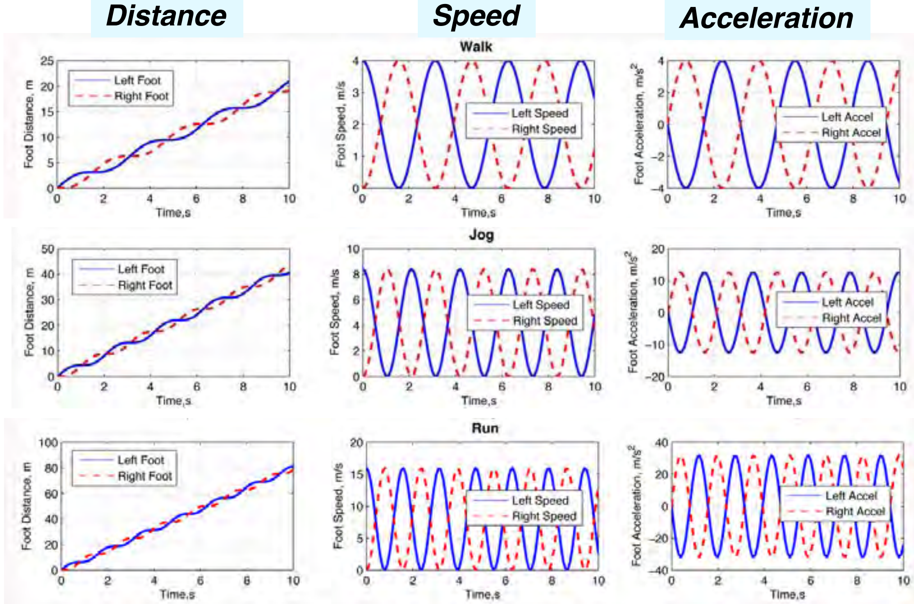



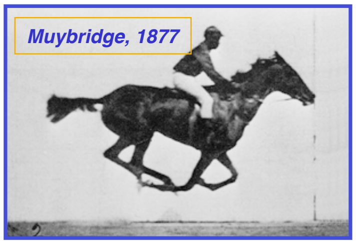


Dynamic Effects Increase with Speed

- Horizontal foot motion ~ sinusoidal oscillation
- Increasing acceleration from walk to jog to run
- Increasing importance of forces and dynamics

Distance ***Speed*** ***Acceleration***

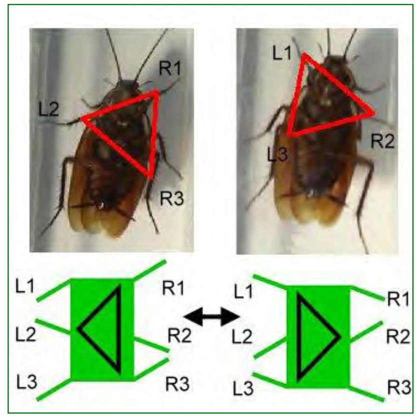




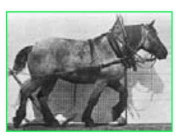
Gaits

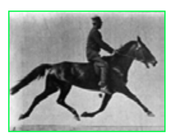
- Biped
- Quadraped
- Hexaped

- Walking
  - Statically stable
  - Statically unstable
- Running

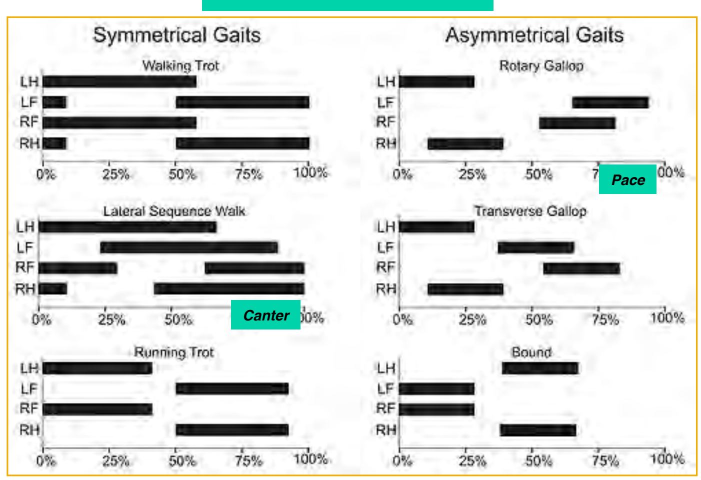


Quadraped Gaits



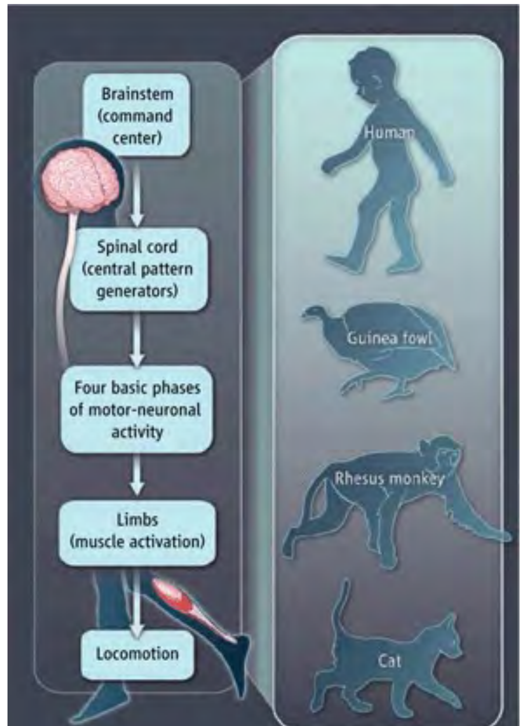


Feet on the Ground



Locomotor Primitives

- Common across legged vertebrate species
- Brain command: ***declarative neural signal***
- Spinal column: ***reflexive central pattern generator***
- Phases of motor-neuronal activity
  - "Toe-off"
  - Flexion
  - Extension
  - Limb alternation
- Muscle activation
- ***Plus sensory feedback***



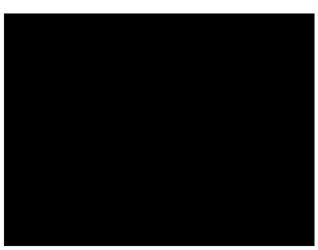
Grillner, *Science*, 2011

Hopping Robots

(Raibert, ~1990)

High inertia of "sprung" mass

2-D (Planar)



3-D

Kangeroo hopping
http://www.youtube.com/watch?v=OpYRIW314sE

Sandia Robot
http://www.youtube.com/watch?v=SDSkqt2xpcc

Walking Robots

Passive Walking
TU Delft

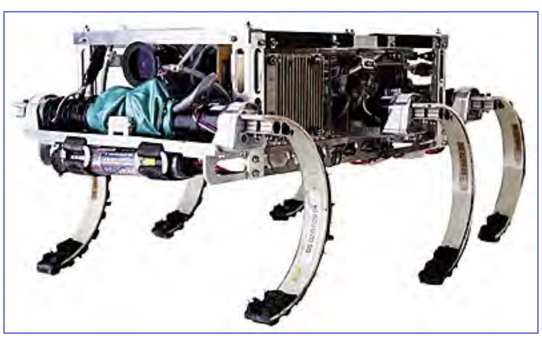
Cytron Kit Robot

Spider Robot
https://www.youtube.com/watch?v=-vVbIGlIMgw

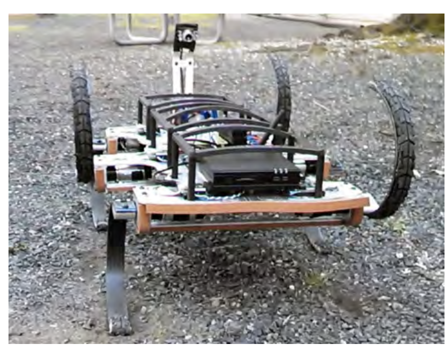
Hexapod Robots

Combined walking and rolling motion
Alternating tripod gate

RHex, Rigid body



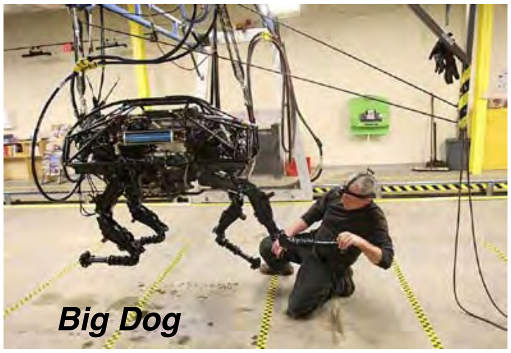
iJus (Princeton '13 IW),
Flexible spine
(3 segments)

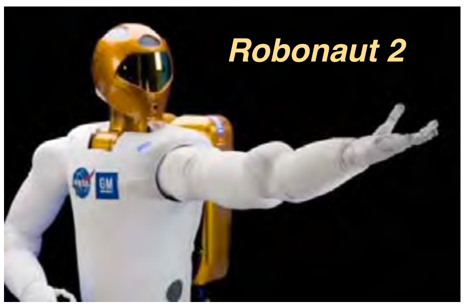


http://www.youtube.com/watch?v=a0NFrA-Nx4Y

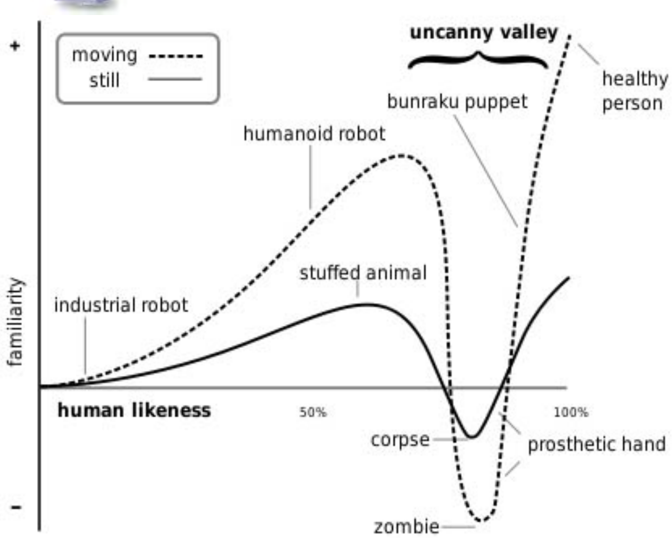
https://www.youtube.com/watch?v=35owx65Ei6g&hd=1

Big Dog, ATLAS, and Robonaut





The Uncanny Valley



Prosthetic Foot/Ankle/Hand

Robot Ankle
http://www.youtube.com/watch?v=HhSVqsHzRI4

3D-Printed Hand
https://www.youtube.com/watch?v=Cl8ijPGEKO8

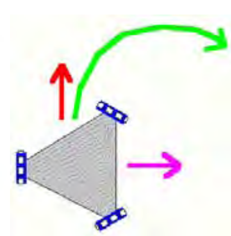
Robotic Exoskeleton

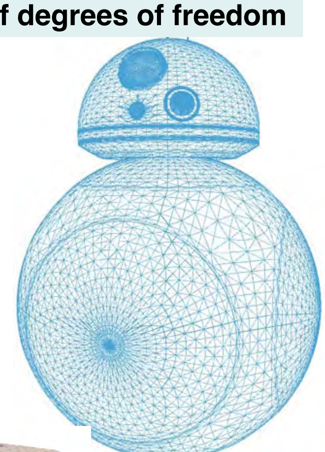
Paraplegic student walks at 2011 UC Berkeley graduation
http://newscenter.berkeley.edu/2011/05/12/paraplegic-student-exoskeleton-graduation-walk/

Rolling Vehicles

Holonomic Robots

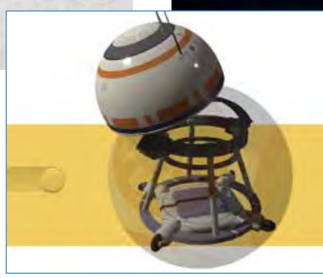
Controllable # of degrees of freedom = Total # of degrees of freedom





Mecanum Wheel

Sphero Ball and BB-8

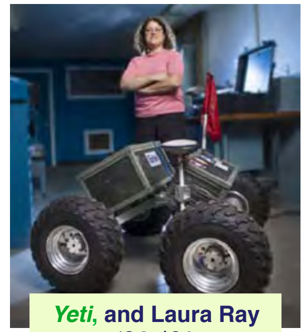


NonHolonomic Robots

Controllable # of degrees of freedom
≠ Total # of degrees of freedom

Wheeled and Tracked Ground Vehicles

- **Vacuum cleaners (*Roomba*)**
- **Military/Emergency robots (*PackBot*)**
- **Exploration robots (*Yeti*)**



Yeti, and Laura Ray '84, *91

mBot

Autonomous Automobiles & Trucks

5 Levels of Autonomy

Level 0:
Human driver controls all: steering, brakes, throttle, power.

The Big Problem With Self-Driving Cars Is People, Rodney Brooks, *IEEE Spectrum*, Jul/Aug 2017

Level 1:
Most functions are still controlled by the driver, but a specific function (like steering or accelerating) can be done automatically by the car.

Level 2:
At least one driver-assistance system is automated. Driver is disengaged from physically operating the vehicle (hands off the steering wheel AND foot off the pedal at the same time).

5 Levels of Autonomy

Level 3:
Driver shifts "safety-critical functions" to the vehicle under certain traffic or environmental conditions.

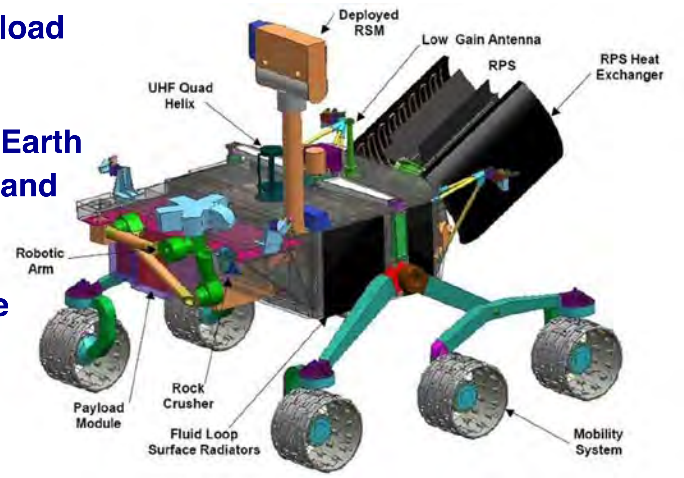
The Big Problem With Self-Driving Cars Is People, Rodney Brooks, *IEEE Spectrum*, Jul/Aug 2017

Level 4:
Fully autonomous vehicles perform all safety-critical driving functions in certain areas and under defined weather conditions.

Level 5:
Fully autonomous system is equal to that of a human driver, in every driving scenario.

Mars Science Laboratory
(Curiosity)

- Transport science payload over Martian surface
  - Rocker-bogie design
- Communications with Earth
- Guidance, navigation, and control
- Power supply
- Support for deployable devices
- Size ~ Mini-Cooper
- Landed, 8/6/12, and operational



Curiosity Trailer
https://mars.nasa.gov/msl/

Math Review

- *Matrix and Transpose*
- *Sums and Multiplication*
- *Matrix Products*
- *Identity Matrix*
- *Matrix Inverse*
- *Transformations*

Matrix and Transpose

- **Matrix**:
  - Usually bold capital or capital: **F** or F
  - Dimension = (m x n)

$$\mathbf{A} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & k \\ l & m & n \end{bmatrix}$$

4 x 3

- **Transpose**:
  - Interchange rows and columns

$$\mathbf{A}^T = \begin{bmatrix} a & d & g & l \\ b & e & h & m \\ c & f & k & n \end{bmatrix}$$

3 x 4

Matrix Products

Matrix-vector product transforms one vector into another

$$\mathbf{y} = \mathbf{A}\mathbf{x} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & k \\ l & m & n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} ax_1 + bx_2 + cx_3 \\ dx_1 + ex_2 + fx_3 \\ gx_1 + hx_2 + kx_3 \\ lx_1 + mx_2 + nx_3 \end{bmatrix}$$

$$(n \times 1) = (n \times m)(m \times 1)$$

Matrix-matrix product produces a new matrix

$$\mathbf{A} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix}; \quad \mathbf{AB} = \begin{bmatrix} (a_1b_1 + a_2b_3) & (a_1b_2 + a_2b_4) \\ (a_3b_1 + a_4b_3) & (a_3b_2 + a_4b_4) \end{bmatrix}$$

$$(n \times m) = (n \times l)(l \times m)$$

Numerical Example 1

$$\mathbf{y} = \mathbf{A}\mathbf{x} = \begin{bmatrix} 2 & 4 & 6 \\ 3 & -5 & 7 \\ 4 & 1 & 8 \\ -9 & -6 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$(n \times 1) = (n \times m)(m \times 1)$$

$$= \begin{bmatrix} (2x_1 + 4x_2 + 6x_3) \\ (3x_1 - 5x_2 + 7x_3) \\ (4x_1 + x_2 + 8x_3) \\ (-9x_1 - 6x_2 - 3x_3) \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

Numerical Example 2

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad ; \quad \mathbf{B} = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

$$\mathbf{AB} = \begin{bmatrix} (5+14) & (6+16) \\ (15+28) & (18+32) \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

$$\mathbf{x}_A = \mathbf{A}\mathbf{x}_B \quad ; \quad \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_B$$

$$\mathbf{x}_B = \mathbf{B}\mathbf{x}_o \quad ; \quad \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_o$$

$$\mathbf{x}_A = \mathbf{A}\mathbf{x}_B = \mathbf{AB}\mathbf{x}_o \quad ; \quad \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_A = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_o$$

Square Matrix Identity and Inverse

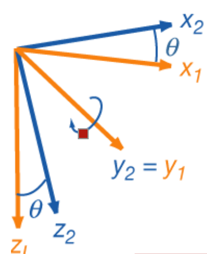
- **Identity matrix: no change when it multiplies a conformable vector or matrix**

$$\mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{x} = \mathbf{I}\mathbf{x}$$

- **A non-singular square matrix multiplied by its inverse forms an identity matrix**

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$$



Matrix Inverse Example

Transformation $\mathbf{x}_2 = \mathbf{A}\mathbf{x}_1$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_2 = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_1$$

Inverse Transformation $\mathbf{x}_1 = \mathbf{A}^{-1}\mathbf{x}_2$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_1 = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_2$$

Consequently, …

$$\mathbf{AA}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$$

$$\mathbf{AA}^{-1} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}^{-1}$$

$$= \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{x}_2 = \mathbf{Ax}_1 = \mathbf{AA}^{-1}\mathbf{x}_2 = \mathbf{x}_2$$

Computation of (*n* x *n*) Matrix Inverse

$$\mathbf{y} = \mathbf{Ax}; \quad \mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$$

$$\dim(\mathbf{x}) = \dim(\mathbf{y}) = (n \times 1); \quad \dim(\mathbf{A}) = (n \times n)$$

$$[\mathbf{A}]^{-1} = \frac{\text{Adj}(\mathbf{A})}{|\mathbf{A}|} = \frac{\text{Adj}(\mathbf{A})}{\det \mathbf{A}} \quad \frac{(n \times n)}{(1 \times 1)}$$

$$= \frac{\mathbf{C}^T}{\det \mathbf{A}}; \quad \mathbf{C} = \textit{matrix of cofactors}$$

Cofactors are signed minors of **A**

ij[th] *minor of* **A** *is the determinant of* **A** *with the i*[th] *row and j*[th] *column removed*

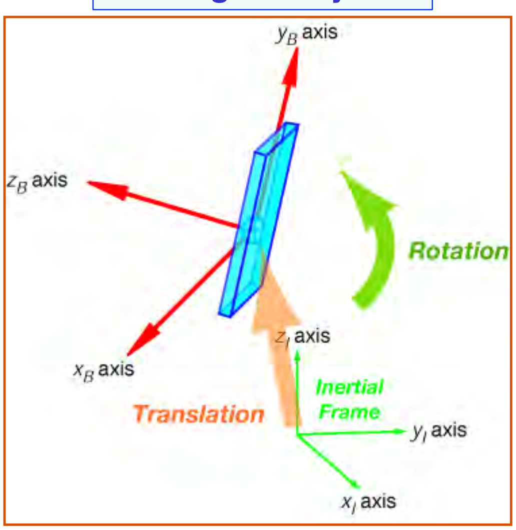
Frames of Reference

33

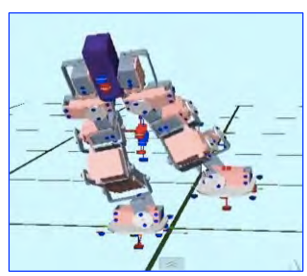
Pose of an Object

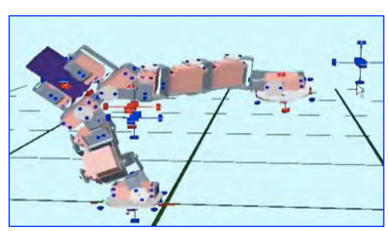
Expression of an object's frame(s) of reference with respect to the original frame

Rigid Body



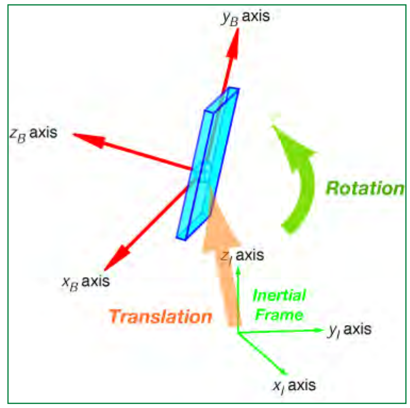
Multi-Link Body



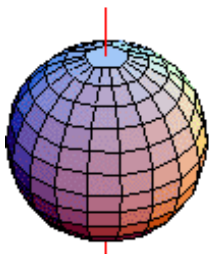


34

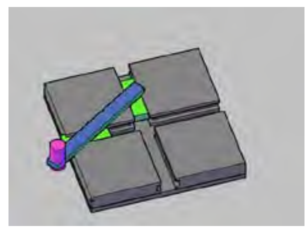
Transformations Between Reference Frames



Rotation



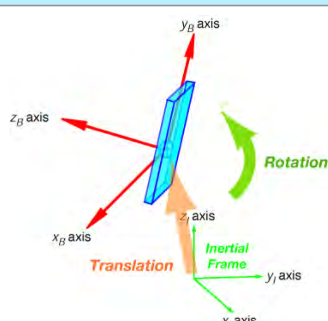
Translation



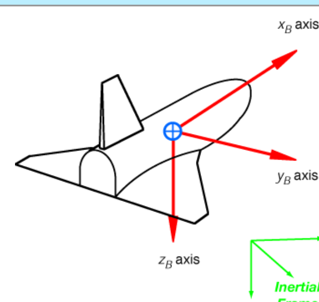
Cartesian Frames of Reference

- Reference frames of interest
  - *I*: Inertial frame (fixed to inertial space, unmoving)
  - *B*: Body frame (fixed to body, moving, non-inertial)
- Translation
  - Linear position of the body frame origin with respect to the inertial frame origin
- Rotation
  - Orientation of the body frame axes with respect to the inertial frame axes

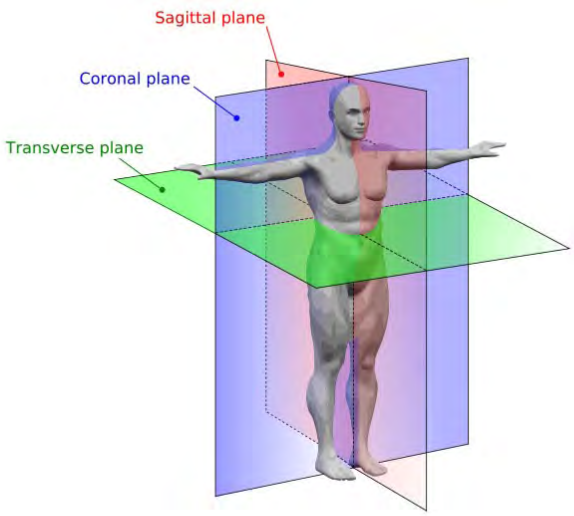
Common convention (*z* up)

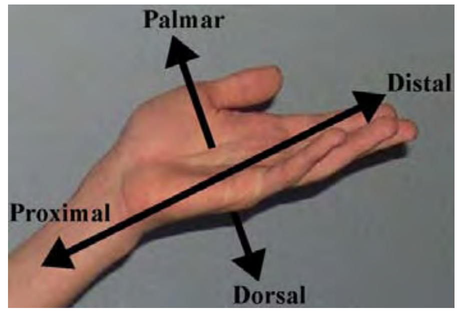


Aircraft convention (*z* down)

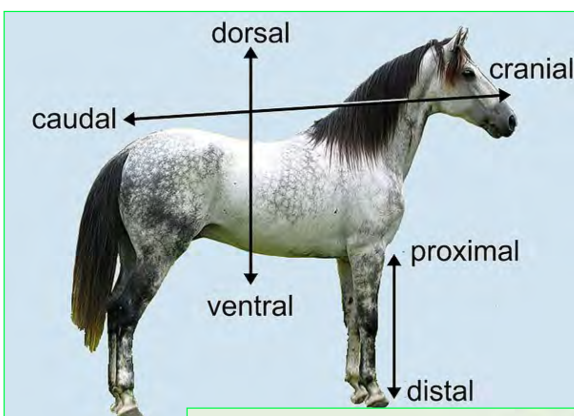


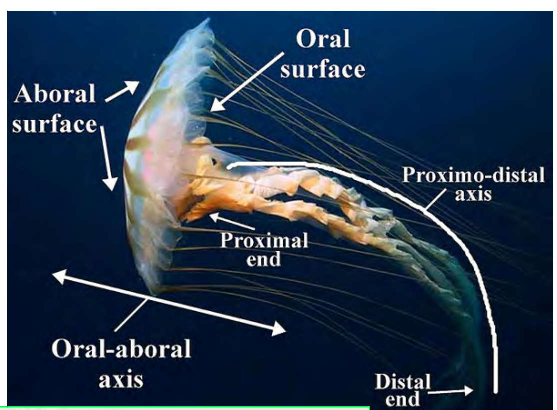
Human Anatomical Coordinates

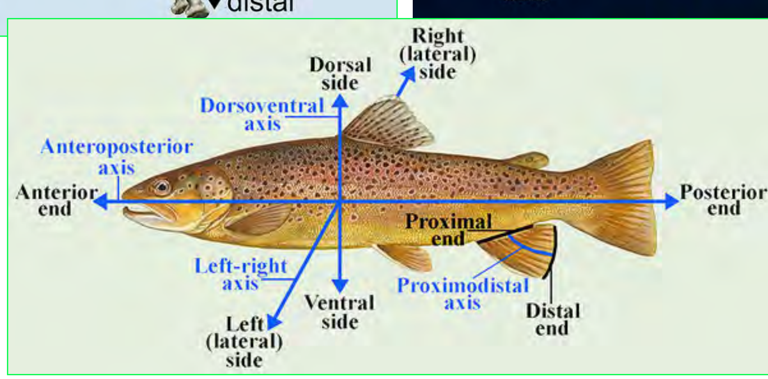




Animal Anatomical Coordinates







Measurement of Position in Alternative Frames - 1

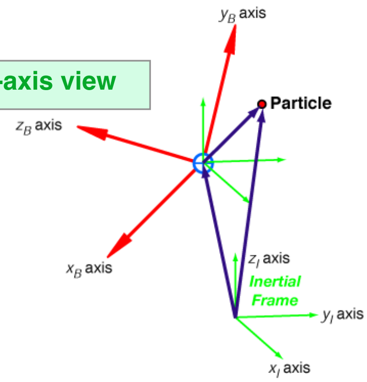
Position vector

$$\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

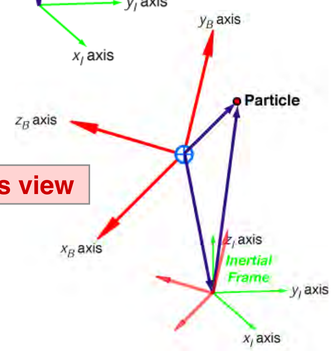
$$\mathbf{r}_{particle} = \mathbf{r}_{origin} + \Delta\mathbf{r}_{w.r.t.origin}$$

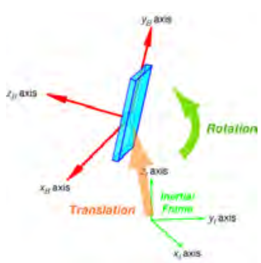
Differences in frame orientations must be taken into account in adding vector components

Inertial-axis view



Body-axis view



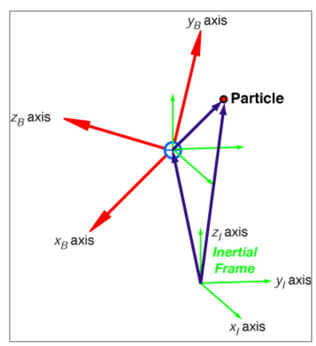


Measurement of Position in Alternative Frames - 2

Inertial-axis view

$$\mathbf{r}_{particle_I} = \mathbf{H}_B^I \mathbf{r}_B + \mathbf{r}_{body\ origin_I}$$

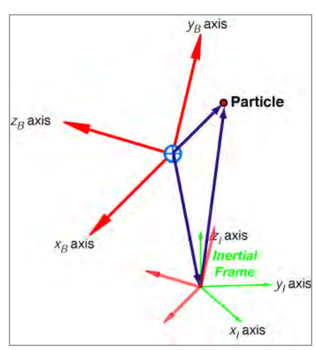
$\mathbf{H}_B^I$: from Body-Axis Vector to Inertial-Axis Vector

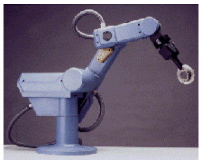


Body-axis view

$$\mathbf{r}_{particle_B} = \mathbf{H}_I^B \mathbf{r}_I + \mathbf{r}_{inertial\ origin_B}$$

$\mathbf{H}_I^B$: from Inertial-Axis Vector to Body-Axis Vector





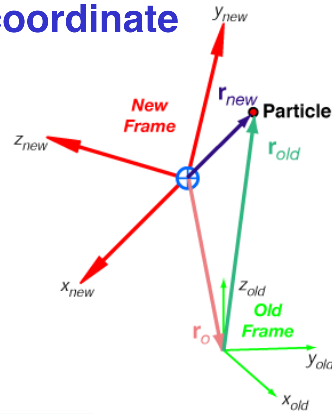
Rotation + Translation
("Forward Kinematics")

- **Expression of a vector in a new coordinate frame**
  - Displaced from old frame
  - Rotated w.r.t. old frame

$$\mathbf{r}_{new} = \mathbf{H}_{old}^{new} \mathbf{r}_{old} + \mathbf{r}_{old_{new}}$$

Rotation matrix

$$\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$



- **Augmented vector**
  - Concatenate a "1" to r

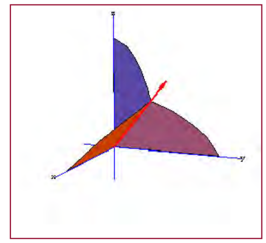
$$\mathbf{s} = \begin{bmatrix} \mathbf{r} \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \equiv$$

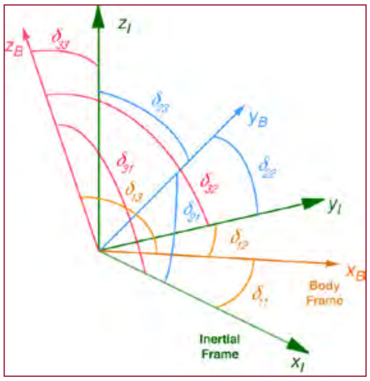
Homogeneous coordinate

Rotational Orientation of a Rigid Body

Direction Cosine Matrix

Orientation of Axes of One Coordinate Frame with Respect to Another





Angles between each *I* axis and each *B* axis

$$\mathbf{H}_I^B = \begin{bmatrix} \cos\delta_{11} & \cos\delta_{21} & \cos\delta_{31} \\ \cos\delta_{12} & \cos\delta_{22} & \cos\delta_{32} \\ \cos\delta_{13} & \cos\delta_{23} & \cos\delta_{33} \end{bmatrix}$$

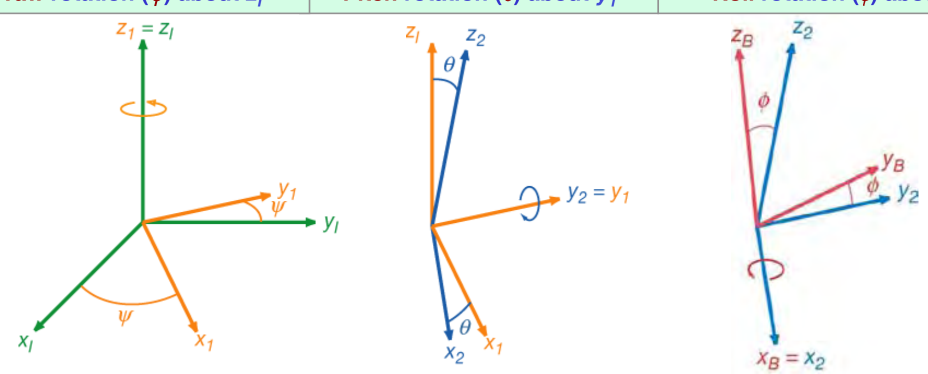
$$\begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} = \begin{bmatrix} \cos\delta_{11} & \cos\delta_{21} & \cos\delta_{31} \\ \cos\delta_{12} & \cos\delta_{22} & \cos\delta_{32} \\ \cos\delta_{13} & \cos\delta_{23} & \cos\delta_{33} \end{bmatrix} \begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix}$$

Orientation of One Frame with Respect to Another

Euler Angles

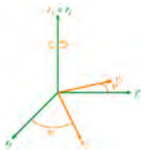
- **Conventional sequence of rotations from inertial to body frame**
  - **Each rotation occurs about a single axis**
  - **Right-hand rule**
  - **Yaw, then pitch, then roll**

Yaw rotation (ψ) about z_I	Pitch rotation (θ) about y_1	Roll rotation (ϕ) about x_2



Effects of Orientation on Vector Transformation

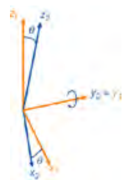
Yaw rotation (ψ) about z_I



$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_1 = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_I = \begin{bmatrix} (x_I \cos\psi + y_I \sin\psi) \\ (-x_I \sin\psi + y_I \cos\psi) \\ z_I \end{bmatrix}$$

$$\mathbf{r}_1 = \mathbf{H}_I^1 \mathbf{r}_I$$

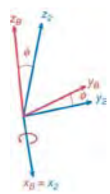
Pitch rotation (θ) about y_1



$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_2 = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_1$$

$$\mathbf{r}_2 = \mathbf{H}_1^2 \mathbf{r}_1 = \mathbf{H}_1^2 \mathbf{H}_I^1 \mathbf{r}_I = \mathbf{H}_I^2 \mathbf{r}_I$$

Roll rotation (ϕ) about x_2



$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_2$$

$$\mathbf{r}_B = \mathbf{H}_2^B \mathbf{r}_2 = \mathbf{H}_2^B \mathbf{H}_1^2 \mathbf{r}_1 = \mathbf{H}_2^B \mathbf{H}_I^2 \mathbf{H}_I^1 \mathbf{r}_I = \mathbf{H}_I^B \mathbf{r}_I$$

The Rotation Matrix

Direction Cosine Matrix expressed in terms of Euler Angles

$$\mathbf{H}_I^B(\phi,\theta,\psi) = \mathbf{H}_2^B(\phi)\mathbf{H}_1^2(\theta)\mathbf{H}_I^1(\psi)$$

$$\mathbf{H}_I^B(\phi,\theta,\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & \sin\phi\cos\theta \\ \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & \cos\phi\cos\theta \end{bmatrix}$$

$$\equiv \begin{bmatrix} \cos\delta_{11} & \cos\delta_{21} & \cos\delta_{31} \\ \cos\delta_{12} & \cos\delta_{22} & \cos\delta_{32} \\ \cos\delta_{13} & \cos\delta_{23} & \cos\delta_{33} \end{bmatrix}$$

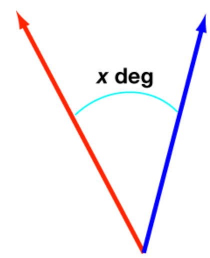
Properties of the Rotation Matrix

- The three-Euler-angle rotation matrix from ***I*** to ***B*** is the product of 3 single-angle rotation matrices

$$\mathbf{H}_I^B(\phi,\theta,\psi) = \mathbf{H}_2^B(\phi)\mathbf{H}_1^2(\theta)\mathbf{H}_I^1(\psi)$$

- The rotation matrix produces an orthonormal transformation
  - Angles are preserved
  - Lengths are preserved

$$|\mathbf{r}_I| = |\mathbf{r}_B| \quad ; \quad |\mathbf{s}_I| = |\mathbf{s}_B|$$
$$\angle(\mathbf{r}_I,\mathbf{s}_I) = \angle(\mathbf{r}_B,\mathbf{s}_B)$$



- With same origins, $\mathbf{r}_o = 0$

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

Orthonormal Rotation

- Inverse relationship: Transformation from ***B*** to ***I***

$$\mathbf{r}_B = \mathbf{H}_I^B\mathbf{r}_I \quad ; \quad \mathbf{r}_I = \left(\mathbf{H}_I^B\right)^{-1}\mathbf{r}_B = \mathbf{H}_B^I\mathbf{r}_B$$

- Because rotation transformation is orthonormal,
  - Inverse = transpose
  - Rotation matrix is always non-singular

$$\mathbf{H}_B^I = \left(\mathbf{H}_I^B\right)^{-1} = \left(\mathbf{H}_I^B\right)^T \quad = \mathbf{H}_1^I\mathbf{H}_2^1\mathbf{H}_B^2$$

$$\mathbf{H}_B^I\,\mathbf{H}_I^B = \mathbf{H}_I^B\mathbf{H}_B^I = \mathbf{I}$$

Orthonormal Transformation of Vector Coordinates

Same vector, different points of view

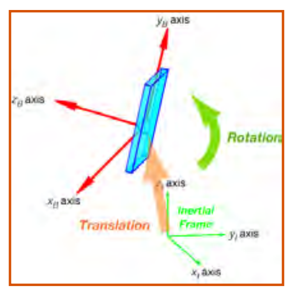
From inertial frame to body frame

$$\begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} = \left[\begin{array}{c:c:c} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \hdashline -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & \sin\phi\cos\theta \\ \hdashline \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & \cos\phi\cos\theta \end{array}\right] \begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix}$$

From body frame to inertial frame

$$\begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} = \left[\begin{array}{c:c:c} \cos\theta\cos\psi & -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi \\ \hdashline \cos\theta\sin\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi \\ \hdashline -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{array}\right] \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix}$$

Homogeneous Transformation Matrix



Express rotation and translation in a single transformation

$$\mathbf{s}_{new} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{new} = \left[\begin{array}{c:c} \begin{pmatrix} \text{Rotation} \\ \text{Matrix} \end{pmatrix}_{old}^{new} & \begin{pmatrix} \text{Location} \\ \text{of Old} \\ \text{Origin} \end{pmatrix}_{new} \\ \hdashline \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} & 1 \end{array}\right] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{old} = \mathbf{A}_{old}^{new}\, \mathbf{s}_{old}$$

$$(4\times1) = \left[\begin{array}{c:c} (3\times3) & (3\times1) \\ \hdashline (1\times3) & (1\times1) \end{array}\right](4\times1) = \left[(4\times4)\right](4\times1) = (4\times1)$$

Homogeneous Transformation

- Rotation and translation can be expressed in terms of homogeneous coordinates
  - Single matrix-vector product produces rotation and transformation

$$\mathbf{s}_{new} = \left[\begin{array}{c:c} \mathbf{H}_{old}^{new} & \mathbf{r}_{old_{new}} \\ \hdashline \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} & 1 \end{array}\right] \mathbf{s}_{old} = \mathbf{A}\,\mathbf{s}_{old}$$

- or

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{new} = \left[\begin{array}{ccc:c} h_{11} & h_{12} & h_{13} & x_o \\ h_{21} & h_{22} & h_{23} & y_o \\ h_{31} & h_{32} & h_{33} & z_o \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{old}$$

Equivalent Scalar Equations for Homogeneous Transformation

$$\mathbf{s}_{new} = \mathbf{A}_{old}^{new}\,\mathbf{s}_{old}$$

Matrix-Vector Multiplication

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{new} = \left[\begin{array}{ccc:c} h_{11} & h_{12} & h_{13} & x_o \\ h_{21} & h_{22} & h_{23} & y_o \\ h_{31} & h_{32} & h_{33} & z_o \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{old}$$

Individual Operations

$$\begin{aligned} x_{new} &= h_{11}x_{old} + h_{12}y_{old} + h_{13}z_{old} + x_o \\ y_{new} &= h_{21}x_{old} + h_{22}y_{old} + h_{23}z_{old} + y_o \\ z_{new} &= h_{31}x_{old} + h_{32}y_{old} + h_{33}z_{old} + z_o \\ & ---- \\ 1 &= 1 \end{aligned}$$

Next Time: Translational and Rotational Dynamics

Supplemental Material

Princeton's Variable-Stability Airplanes

Holonomic and NonHolonomic Airplanes

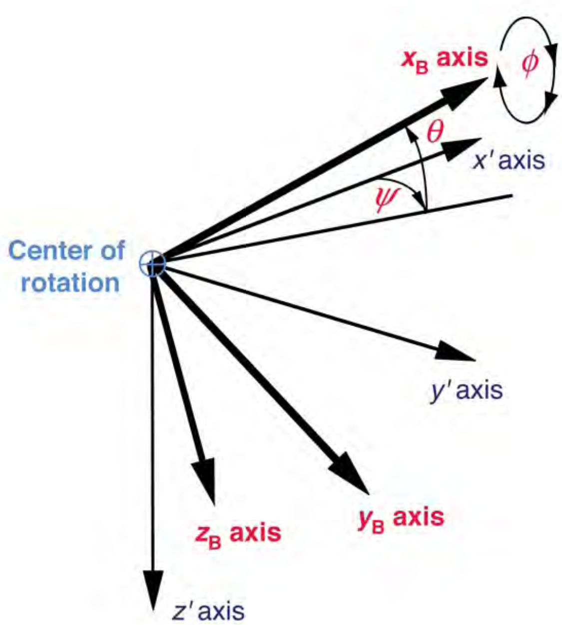
**6 degrees of freedom,
6 independent controls**

Thrust force
Lift force
~~Side force~~

Pitching moment
Yawing moment
Rolling moment

**6 degrees of freedom,
5 independent controls**

Euler Angles (with *z* Axis down)



MATLAB Code for Math Review

Use of Symbolic Variables

```
%   MAE 345 Lecture 2 Math Review
%   Rob Stengel

    clear
    disp('   ')
    disp('=====================================')
    disp('>>>MAE 345 Lecture 2 Math Review<<<')
    disp('=====================================')
    disp('   ')
    disp(['Date and Time are ', num2str(datestr(now))]);
    disp('   ')

%   Matrix
    syms A AT a b c d e f g h k l m n
    A   =   [a b c;d e f;g h k;l m n]   %   Matrix
    AT  =   A'                          %   Matrix Transpose

%   Matrix-Vector Product
    syms x x1 x2 x3 y1 y2 y3 y4
    x   =   [x1;x2;x3]
    y   =   [y1;y2;y3;y4]
    y   =   A * x
```


MATLAB Code for Math Review

```
%   Matrix-Matrix Product
    syms A a1 a2 a3 a4 B b1 b2 b3 b4 AB
    A   =   [a1 a2;a3 a4]
    B   =   [b1 b2;b3 b4]
    AB  =   A * B

%   Example 1
    syms A
    A   =   [2 4 6;3 -5 7;4 1 8;-9 -6 -3]
    y   =   A * x

%   Example 2
    A   =   [1 2;3 4]
    B   =   [5 6;7 8]
    AB  =   A * B

    syms xA xB x0
    x0  =   [x1;x2]
    xA  =   A * xB
    xB  =   B * x0
    xA  =   A * B * x0
```

MATLAB Code for Math Review

```
%   Matrix Identity and Inverse
    I3      =   eye(3)
    x       =   I3 * x
    syms A Ainv
    A       =   [a b c;d e f;g h k]
    Ainv    =   inv(A)
    I3      =   simplify(A * Ainv)
    I3      =   simplify(Ainv * A)

%   Matrix Inverse Example
    syms A Th cTh sTh Ainv
    A       =   [cTh 0 sTh;0 1 0;-sTh 0 cTh]
    Ainv    =   inv(A)
    detA    =   det(A)

    cTh     =   cos(Th)
    sTh     =   sin(Th)
    Th      =   pi / 4
    syms A Ainv
    A       =   [cos(Th) 0 sin(Th);0 1 0;-sin(Th) 0 cos(Th)]
    Ainv    =   inv(A)

%   Consequently, ...
    I3      =   A * Ainv

%   Computation of (n x n) Inverse
    detA    =   det(A)
    AdjA    =   Ainv * detA
```


MATLAB Command Window Output for Math Review

```
======================================
>>>MAE 345 Lecture 2 Math Review<<<
======================================

Date and Time are 03-Sep-2013 13:49:40

A =
[ a, b, c]
[ d, e, f]
[ g, h, k]
[ l, m, n]

AT =
[ conj(a), conj(d), conj(g), conj(l)]
[ conj(b), conj(e), conj(h), conj(m)]
[ conj(c), conj(f), conj(k), conj(n)]

x =
 x1
 x2
 x3

y =
 y1
 y2
 y3
 y4

y =
 a*x1 + b*x2 + c*x3
 d*x1 + e*x2 + f*x3
 g*x1 + h*x2 + k*x3
 l*x1 + m*x2 + n*x3
```

```
A =
[ a1, a2]
[ a3, a4]

B =
[ b1, b2]
[ b3, b4]

AB =
[ a1*b1 + a2*b3, a1*b2 + a2*b4]
[ a3*b1 + a4*b3, a3*b2 + a4*b4]

A =  2     4     6
     3    -5     7
     4     1     8
    -9    -6    -3

y = 2*x1 + 4*x2 + 6*x3
    3*x1 - 5*x2 + 7*x3
      4*x1 + x2 + 8*x3
  - 9*x1 - 6*x2 - 3*x3

A =  1     2
     3     4

B =  5     6
     7     8

AB = 19    22
     43    50
```

```
x0 =
 x1
 x2

xA =
[   xB, 2*xB]
[ 3*xB, 4*xB]

xB =
 5*x1 + 6*x2
 7*x1 + 8*x2

xA =
 19*x1 + 22*x2
 43*x1 + 50*x2

I3 =
     1     0     0
     0     1     0
     0     0     1

x =
 x1
 x2
 x3
```

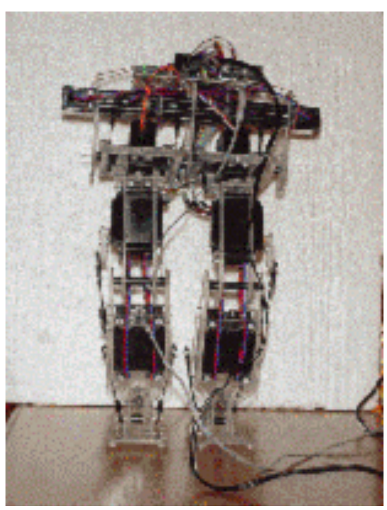
MATLAB Command Window Output for Math Review

```
A =
[ a, b, c]
[ d, e, f]
[ g, h, k]

Ainv =
[  (f*h - e*k)/(a*f*h - b*f*g - c*d*h + c*e*g - a*e*k +
b*d*k), -(c*h - b*k)/(a*f*h - b*f*g - c*d*h + c*e*g -
a*e*k + b*d*k), -(b*f - c*e)/(a*f*h - b*f*g - c*d*h +
c*e*g - a*e*k + b*d*k)]
[ -(f*g - d*k)/(a*f*h - b*f*g - c*d*h + c*e*g - a*e*k +
b*d*k),  (c*g - a*k)/(a*f*h - b*f*g - c*d*h + c*e*g -
a*e*k + b*d*k),  (a*f - c*d)/(a*f*h - b*f*g - c*d*h +
c*e*g - a*e*k + b*d*k)]
[ -(d*h - e*g)/(a*f*h - b*f*g - c*d*h + c*e*g - a*e*k +
b*d*k),  (a*h - b*g)/(a*f*h - b*f*g - c*d*h + c*e*g -
a*e*k + b*d*k), -(a*e - b*d)/(a*f*h - b*f*g - c*d*h +
c*e*g - a*e*k + b*d*k)]

I3 =
[ 1, 0, 0]
[ 0, 1, 0]
[ 0, 0, 1]

I3 =
[ 1, 0, 0]
[ 0, 1, 0]
[ 0, 0, 1]
```

```
A = [  cTh, 0, sTh]
     [    0, 1,    0]
   [ -sTh, 0, cTh]

Ainv =
[ cTh/(cTh^2 + sTh^2), 0, -sTh/(cTh^2 + sTh^2)]
[                   0, 1,                    0]
[ sTh/(cTh^2 + sTh^2), 0,  cTh/(cTh^2 + sTh^2)]

detA = cTh^2 + sTh^2

cTh = cos(Th)

sTh = sin(Th)

Th = 0.7854

A = 0.7071         0    0.7071
         0    1.0000         0
   -0.7071         0    0.7071

Ainv = 0.7071         0   -0.7071
            0    1.0000         0
       0.7071         0    0.7071

I3 = 1     0     0
     0     1     0
     0     0     1

detA = 1

AdjA = 0.7071         0   -0.7071
            0    1.0000         0
       0.7071         0    0.7071
```


American Android All-Terrain Biped

(David Handelman, *89)

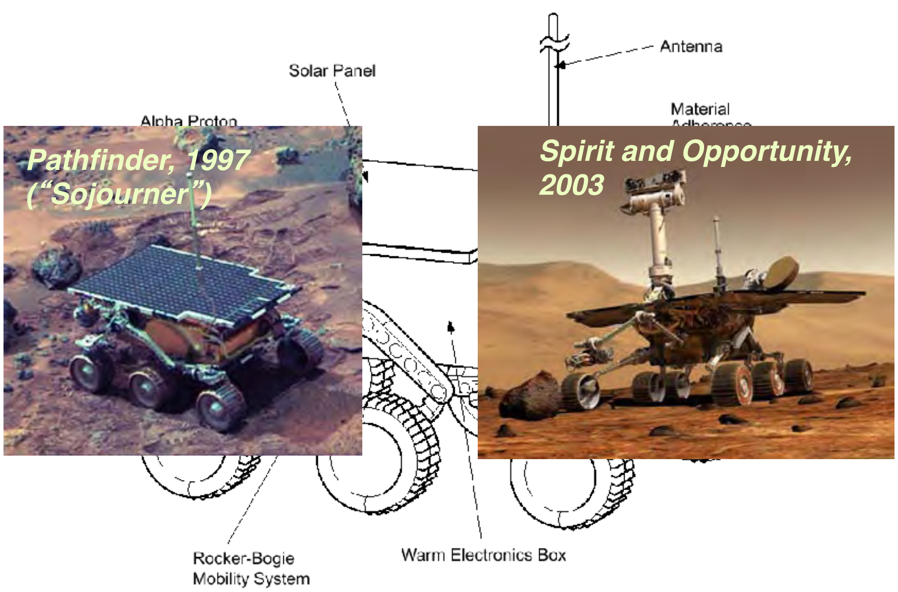


http://www.youtube.com/watch?v=UX0P11wNkcM

Timberjack Logging Robot

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

Mars Exploration Rovers



Personal Assistance

65

Surveillance Robots

SECOM Robot X
http://www.youtube.com/watch?v=0b6izpxj61o

Oculus Robot
http://www.youtube.com/watch?v=Q4L3Ujsclnk

66

Telepresence Robots

iRobot Ava 500 Video Collaboration Robot
http://www.youtube.com/watch?v=hVviDvsBQ78

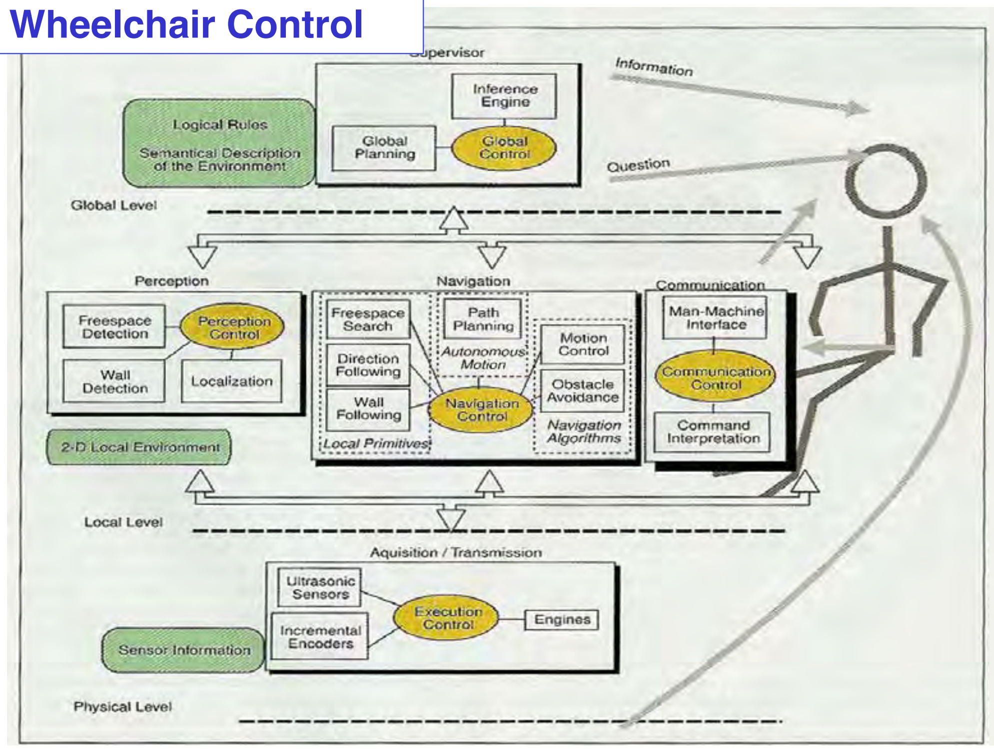
VGo Telepresence Robot
http://www.youtube.com/watch?v=8fdXStgdhEg

iBot and Segway *(DEKA)*

Failure-tolerance Stability

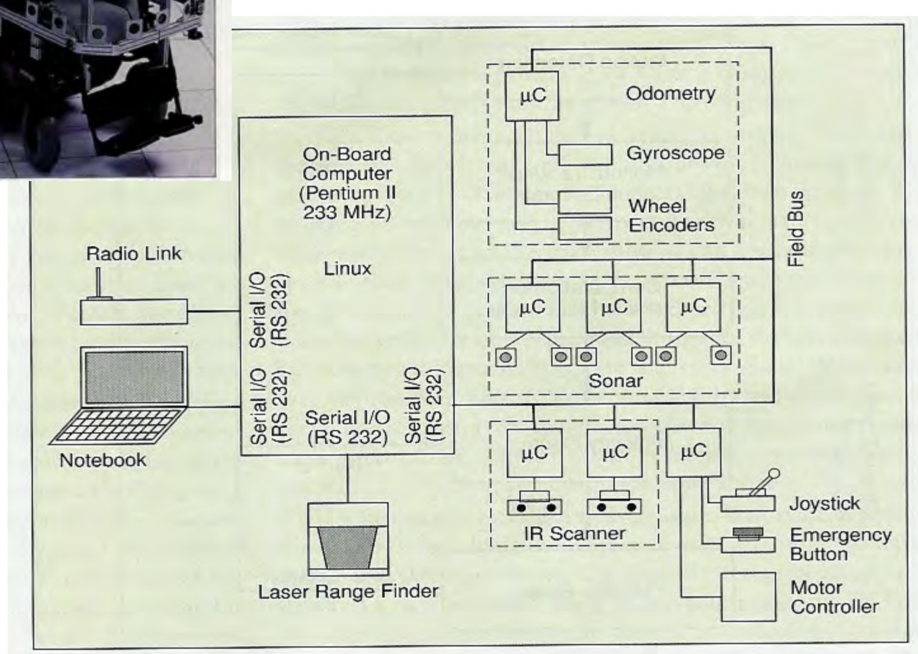
System Redundancy

Hierarchical Model of Wheelchair Control



69

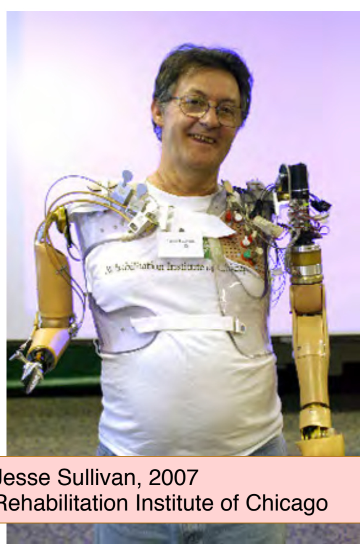
Wheelchair and Control Hardware



70

Personal Assistance Robots

Prosthetic Arms and Hands



Jesse Sullivan, 2007
Rehabilitation Institute of Chicago

Jesse with the DEKA/DARPA Arm, 2009
http://www.youtube.com/watch?v=ddInW6sm7JE

"Terminator Arm"
https://www.youtube.com/watch?v=_qUPnnROxvY

Open Bionics 3D-Printed Hand
https://www.youtube.com/watch?v=Z93Qp69pIWw

Toward the Bionic Man
https://www.youtube.com/watch?v=xBiOQKonkWs

Home-built 3D-Printed Hand
https://www.youtube.com/watch?v=uWL13vvi94s

Robotic Friends for Young and Old

Paro Robotic Seal
http://www.youtube.com/watch?v=Vx8mv87e6wE

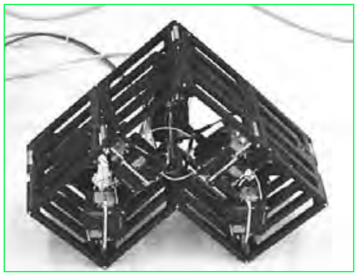
PaPeRo
http://www.youtube.com/watch?v=Z_QKHS3lydA

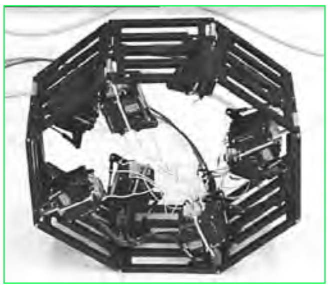
Furby
http://www.youtube.com/watch?v=lAZ8QG8uz9I

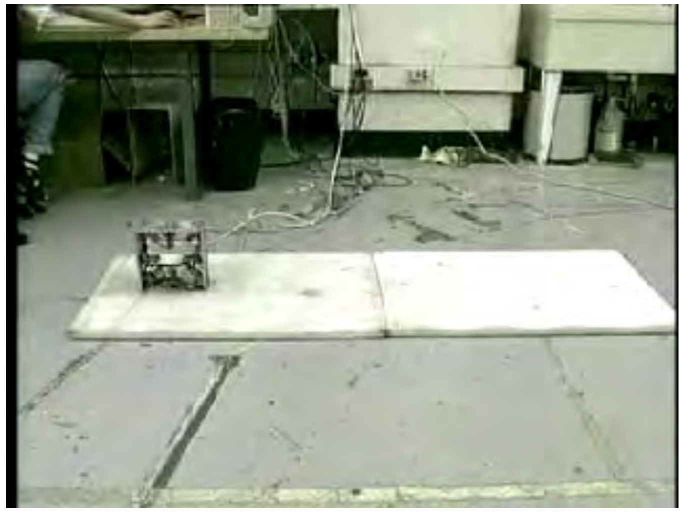
Other

The Blob

(MIT Leg Laboratory, 1995-97)







Meshworm Robot

(Seoul, MIT, Harvard)

Mesh of shape-memory alloy activated by differential heating

http://www.youtube.com/watch?v=EXkf62qGFIl

Snake Robots

Games and Toys

Games

Toys

Toys and A.I.

RQ-HUNO

Zoomer

Robo Me

Rotation of Lunar Module Ascent Stage

LM Ascent Stage from CSM