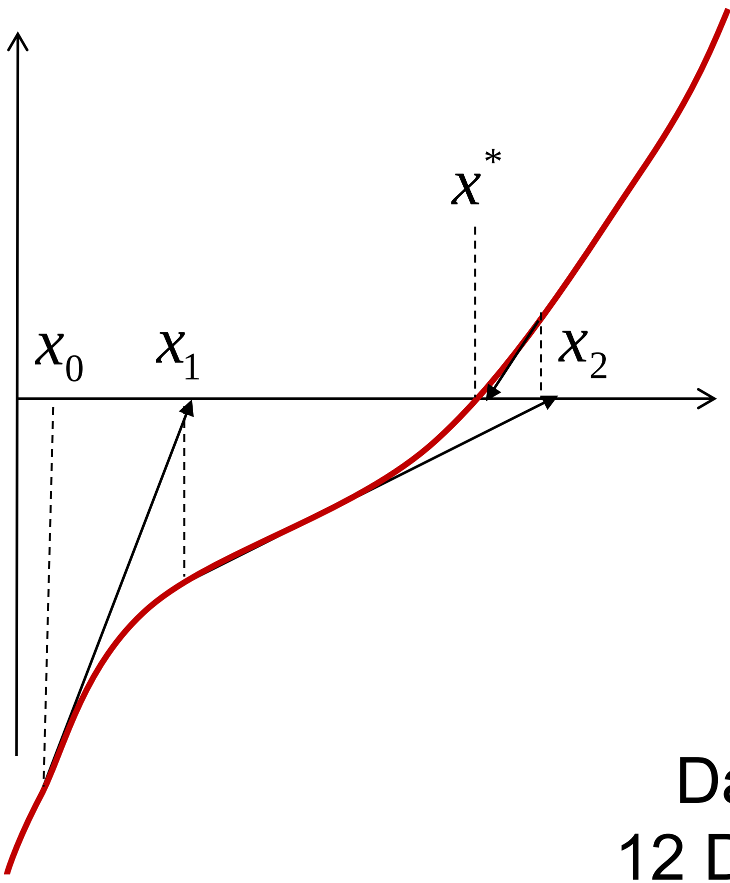
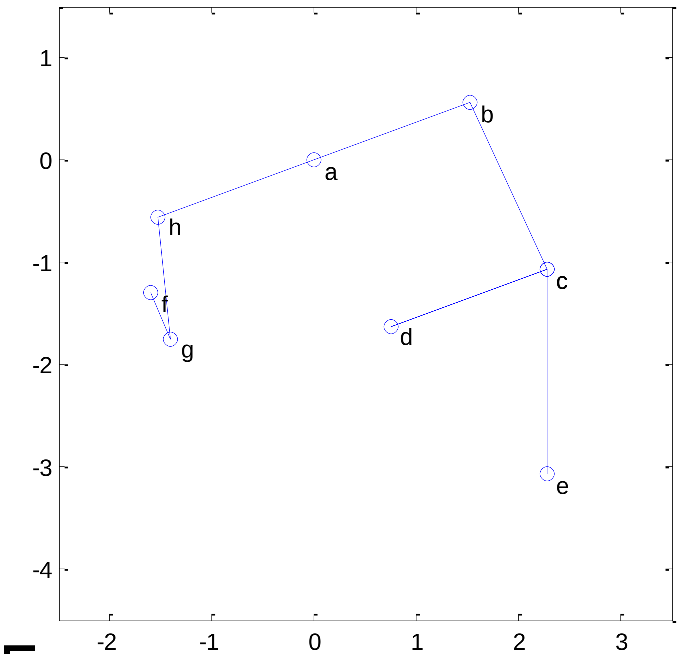
# 2.086 – Numerical Computation for Mechanical Engineers

# Bridging to

# 2.007 –Design and Manufacturing I




Dan Frey

12 DEC 2012

# 2.086 Objectives

- Knowledge
  - An understanding of the basic “canon” of numerical approaches and methods
  - sources of error and uncertainty
  - An understanding of the basic MATLAB architecture/environment, data types, syntax

# 2.086 Objectives (cont.)

- Skills
  - The ability to formulate an engineering problem in a mathematical form …
  - The ability to test and use (or reject) third-party numerical programs with confidence.
  - The ability to solve mechanical engineering problems by numerical approaches …
- Attitudes and Professional Values
  - A commitment to always providing … some indication of error and uncertainty …

# 2.007 Learning Objectives

After taking this subject students should be able to:

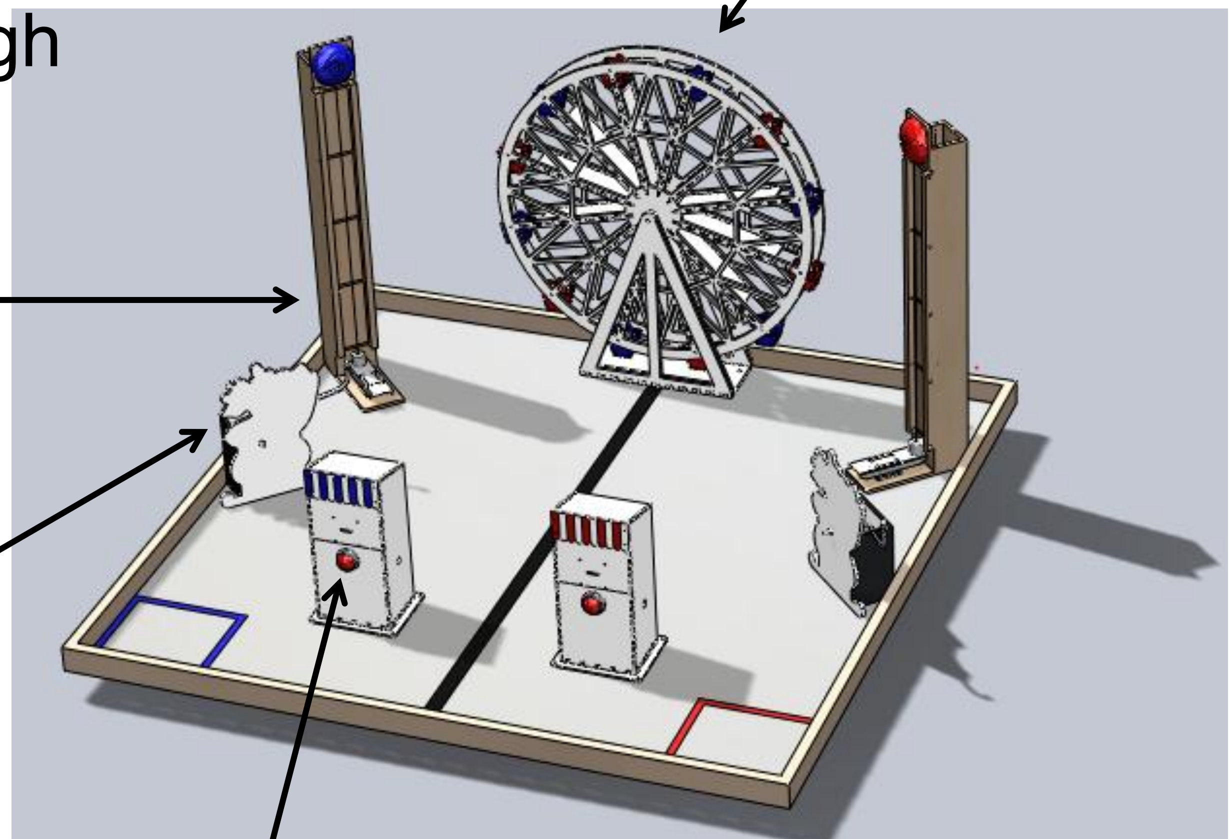
- Generate, analyze, and refine the design of electro-mechanical devices making use of physics and mathematics
- For common machine elements including fasteners, joints, springs, bearings, gearing, belts, chains, shafts, sensors, and electronics
  - Describe the function of the element
  - List common uses in mechanical systems and give examples
  - Analyze its performance and failure modes
  - Describe how they are manufactured and the implications of the alternatives
  - Select an element for a specific use based on information such as that typically available in a manufacturer's catalog
- Apply experimentation and data analytic principles relevant to mechanical design
  - Consider the effects of geometric variation on a design
  - …
- Communicate a design and its analysis (written, oral, and graphical forms)
  - Read and interpret mechanical drawings of systems with moderate complexity
  - …

# Ways to Score

You can multiply your score by up to 3X by rotating the Ferris wheel

You earn points based on how high you can get the slug to go on the “high striker”

You score one point for each liter of inflation your balloon



You score 1 point per ticket dispensed and removed

# Operation!

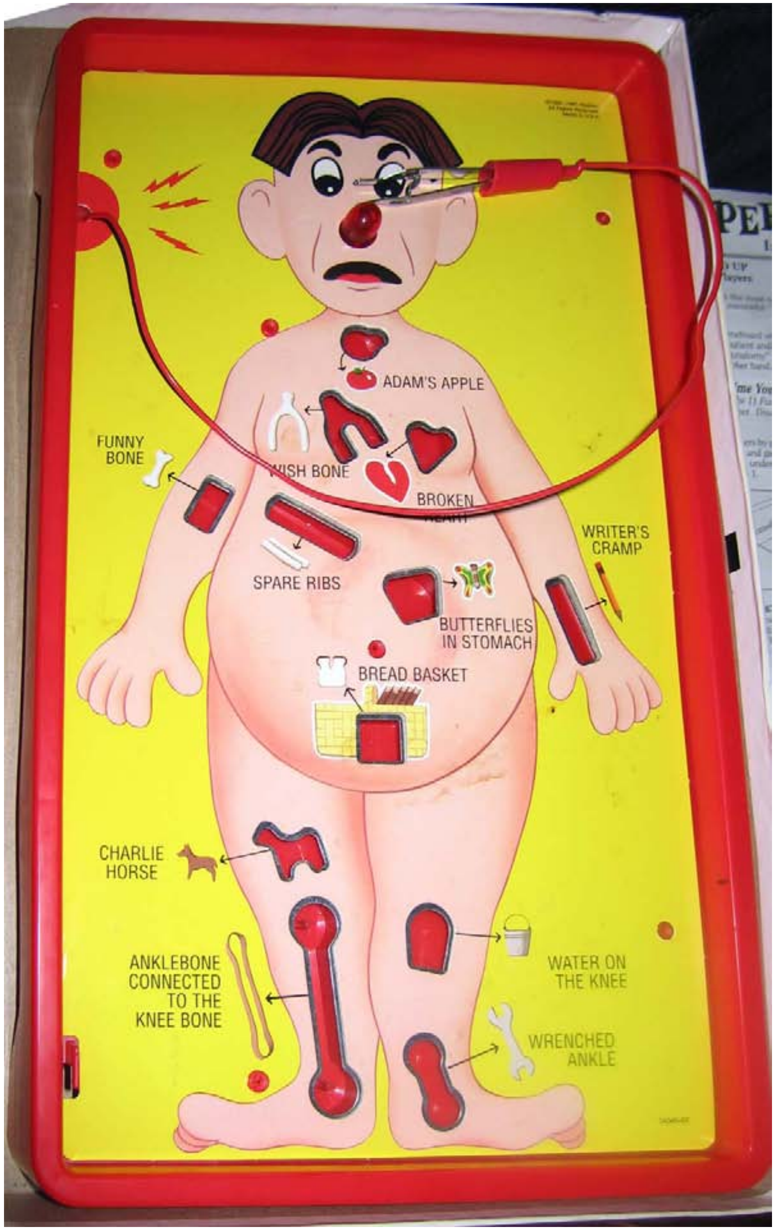


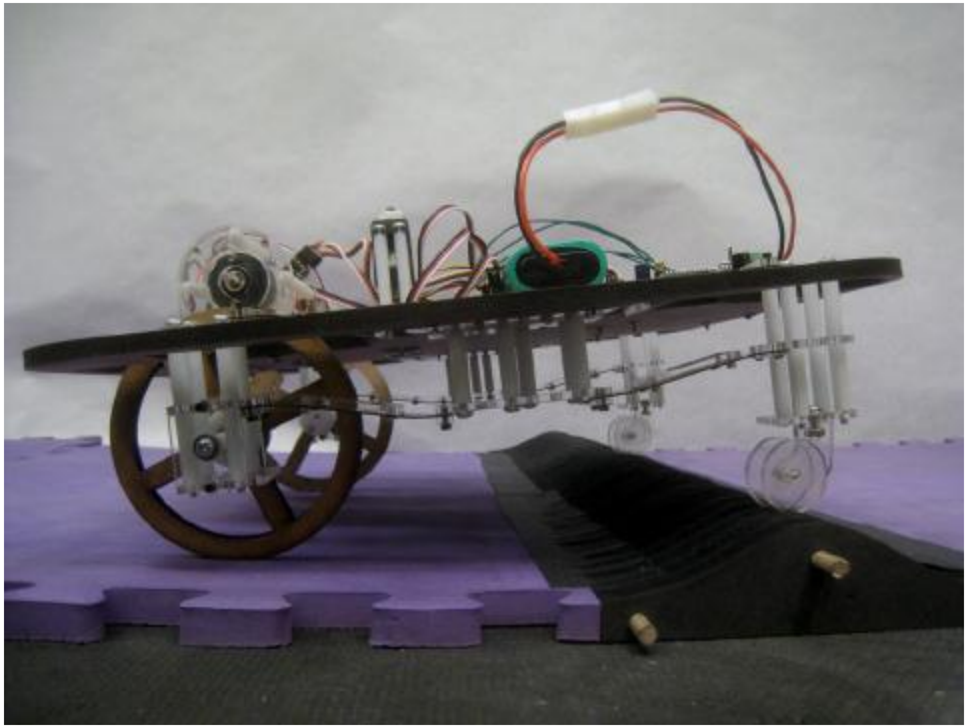

Photo courtesy of watz on Flickr.

# 2.086 Recitation 8

```
delt = 0.01;
J = 1/delt;
u_exac = (exp(-2*1)+1)/2;
up = 1;   %initialization
for i=1:J
    un = up + (-2*up+1)*delt;
    up = un;
end
```

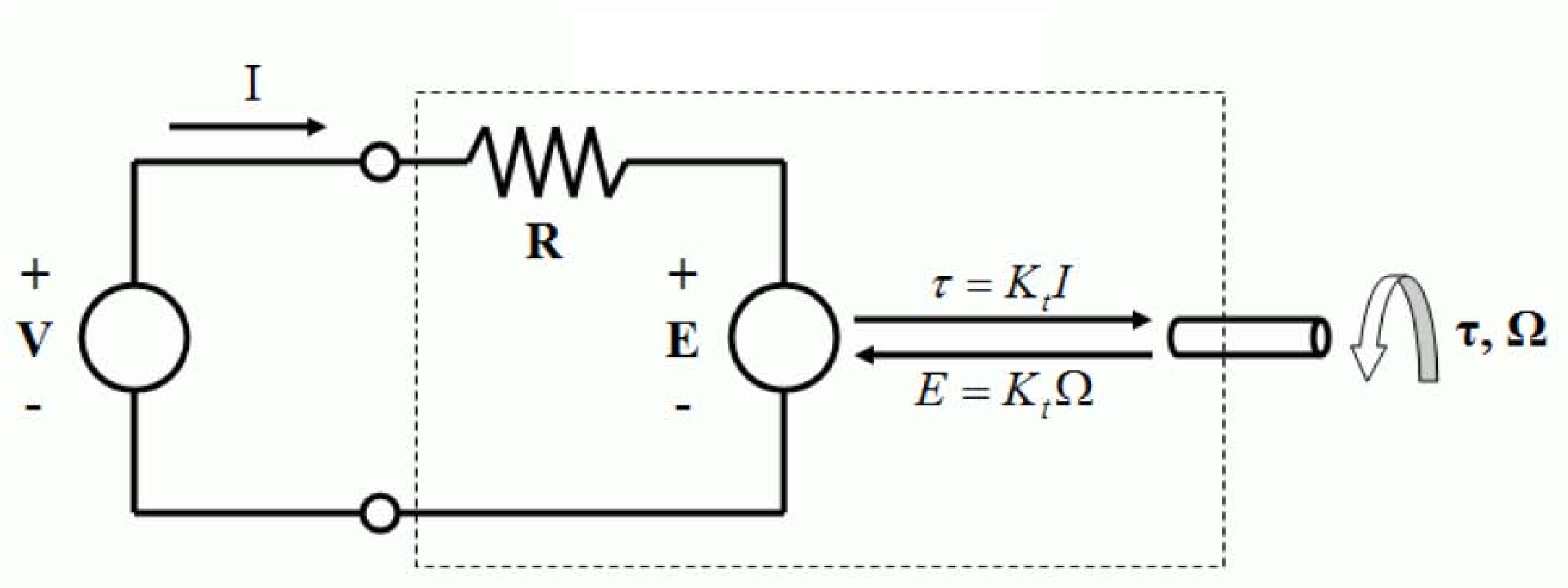
# Apply this to the Macro-Me Robot

- How long would it take to cross the 2.007 contest field (8 feet = 2.4m)?



Courtesy of James Penn. Used with permission.

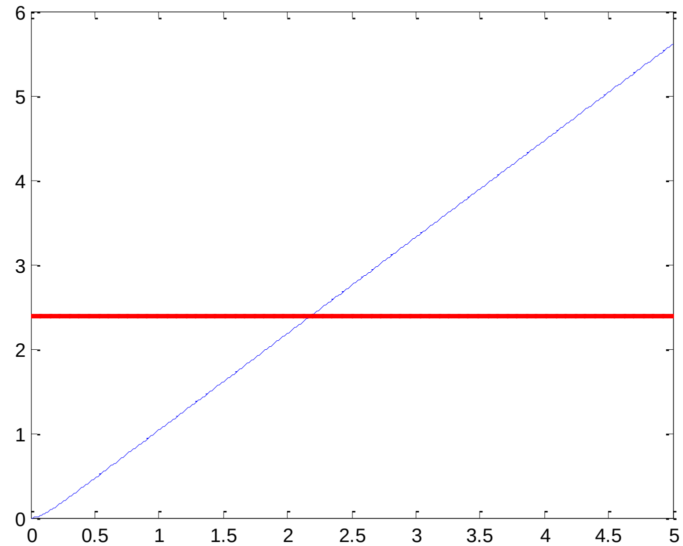
# A Model of a Motor



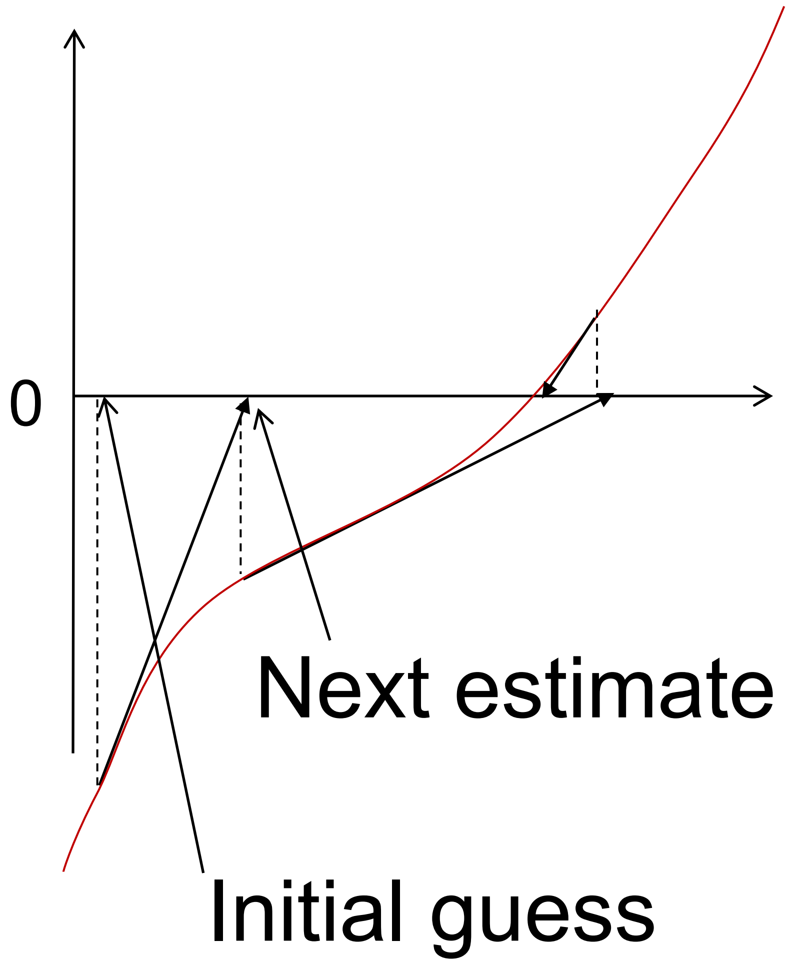

$$\tau(\Omega) = (K_t I) = \left( K_t \frac{(V - E)}{R} \right) = \left( \frac{K_t V_0}{R} - \frac{K_t^2}{R} \Omega \right)$$

# 2.086 Recitation 8 (Adapted)

```
delt = 0.01;
J = 5/delt;

Kt=0.21; Rm=1;
Vb=4.8;  m=1.456; Rw=0.05;

omega(1) = 0; dist(1) = 0;  %initialization
for i=2:J
    omega(i) = omega(i-1) + (((Kt*Vb/Rm)-((Kt^2)*omega(i-1)/Rm))/(m*(Rw^2)))*delt;
    dist(i)=dist(i-1)+omega(i-1)*Rw*delt;
end

time=delt*min(find(dist>2.4))

figure(1)
plot(delt*(1:J),dist)
hold on
plot(delt*(1:J),2.4,'r--')
```



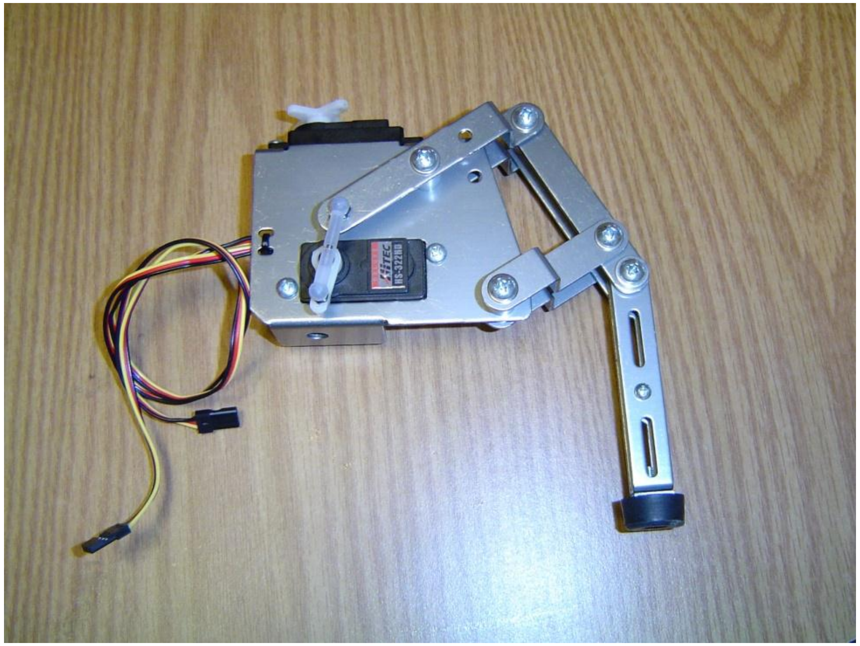
# Newton-Raphson Method




- Make a guess at the solution
- Make a linear approximation of a function by e.g., finite difference
- Solve the linear system
- Use that solution as a new guess
- Repeat until some criterion is met

# Example Problem

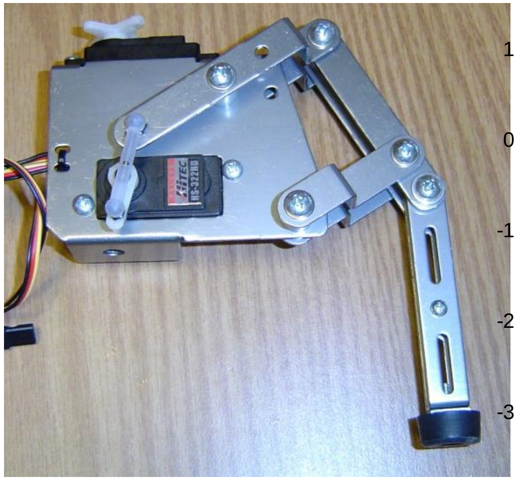
- Here is a leg from a simple robot
- If the servo motor starts from the position shown and rotates 45 deg CCW
- How far will the “foot” descend?

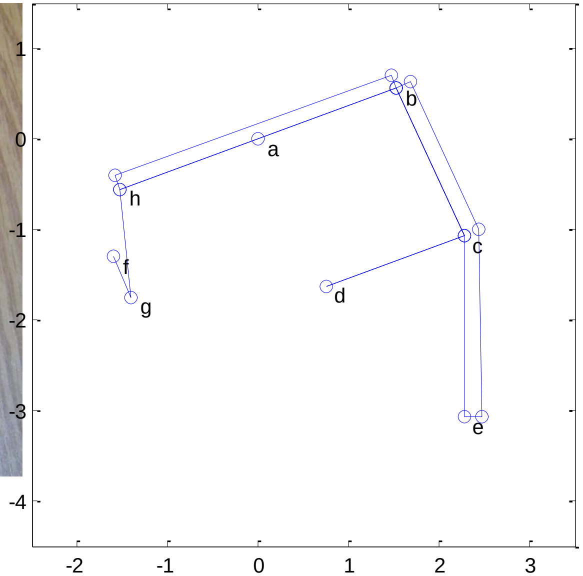


# Define a Few Functions

```
R=@(theta) [cos(theta) -sin(theta) 0 0;
             sin(theta) cos(theta)  0 0;
             0              0       1 0;
             0              0       0 1];

T=@(p) [1 0 0 p(1);
        0 1 0 p(2);
        0 0 1 p(3);
        0 0 0  1];

Rp=@(theta,p) T(p)*R(theta)*T(-p);
```

# Representing the Geometry

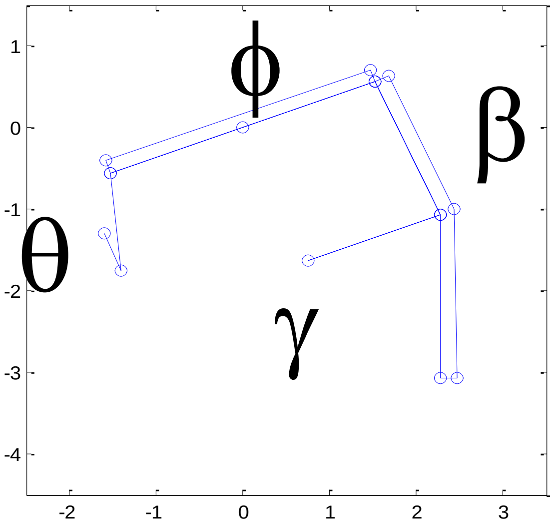





```
a=[0 0 0 1]';
b=[1.527 0.556 0 1]';
c=[2.277 -1.069 0 1]';
d=[0.75 -1.625 0 1]';
e=[2.277 -3.069 0 1]';
f=[-1.6 -1.3 0 1]';
g=[-1.4 -1.75 0 1]';
h=[-1.527 -0.556 0 1]';
leg=[f g h a b c b b+0.05*Rp(-pi/2,b)*(h-b)
h+0.05*Rp(pi/2,h)*(b-h) h b c d c e e+0.1*Rp(-pi/2,e)*(c-e)
c+0.1*Rp(-pi/2,c)*(b-c) b+0.1*Rp(pi/2,b)*(c-b) b];
names=char('f','g','h','a','b','c','d','e');
plot(leg(1,:),leg(2,:),'o-b')
axis equal
axis([-2.5 3.5 -4.5 1.5]);
loc=[1 2 3 4 5 6 13 15];
for i=1:8
    text(leg(1,loc(i))+0.1, leg(2,loc(i))-0.1, names(i))
end
```

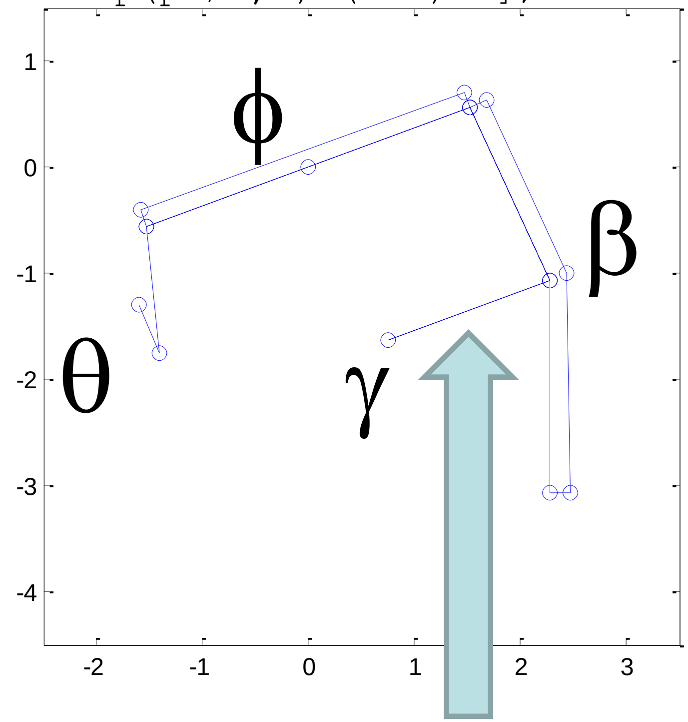
# Animate the Leg Mechanism

```
instant = 0.0001;  % pause between frames
leg=[f g h a b c b b+0.05*Rp(-pi/2,b)*(h-b) h+0.05*Rp(pi/2,h)*(b-h) h b c d c
e e+0.1*Rp(-pi/2,e)*(c-e) c+0.1*Rp(-pi/2,c)*(b-c) b+0.1*Rp(pi/2,b)*(c-b) b];
p = plot(leg(1,:),leg(2,:),'o-b',...
    'EraseMode', 'normal');
axis equal
axis([-2.5 3.5 -4.5 1.5]);
options = optimset('Display','off');
for theta=0:0.5*pi/180:210*pi/180
  g2=Rp(theta,f)*g;
  link1=@(phi) norm(g-h)-norm(g2-Rp(phi,a)*h);
  phi=fzero(link1,0);
  h2=Rp(phi,a)*h;
  b2=Rp(phi,a)*b;
  link2=@(gamma) norm(b-c)-norm(b2-Rp(gamma,d)*c);
  gamma=fzero(link2,0);
  c2=Rp(gamma,d)*c;
  link3=@(beta) norm(c2-Rp(beta,b2)*T(b2-b)*c);
  beta=fsolve(link3,0,options);
  e2=Rp(beta,b2)*T(b2-b)*e;
 leg=[f g2 h2 a b2 c2 b2 b2+0.05*Rp(-pi/2,b2)*(h2-b2) h2+0.05*Rp(pi/2,h2)*(b2-
h2) h2 b2 c2 d c2 e2 e2+0.1*Rp(-pi/2,e2)*(c2-e2) c2+0.1*Rp(-pi/2,c2)*(b2-c2)
b2+0.1*Rp(pi/2,b2)*(c2-b2) b2];
  set(p,'XData',leg(1,:), 'YData',leg(2,:))
  pause(instant)
end
```

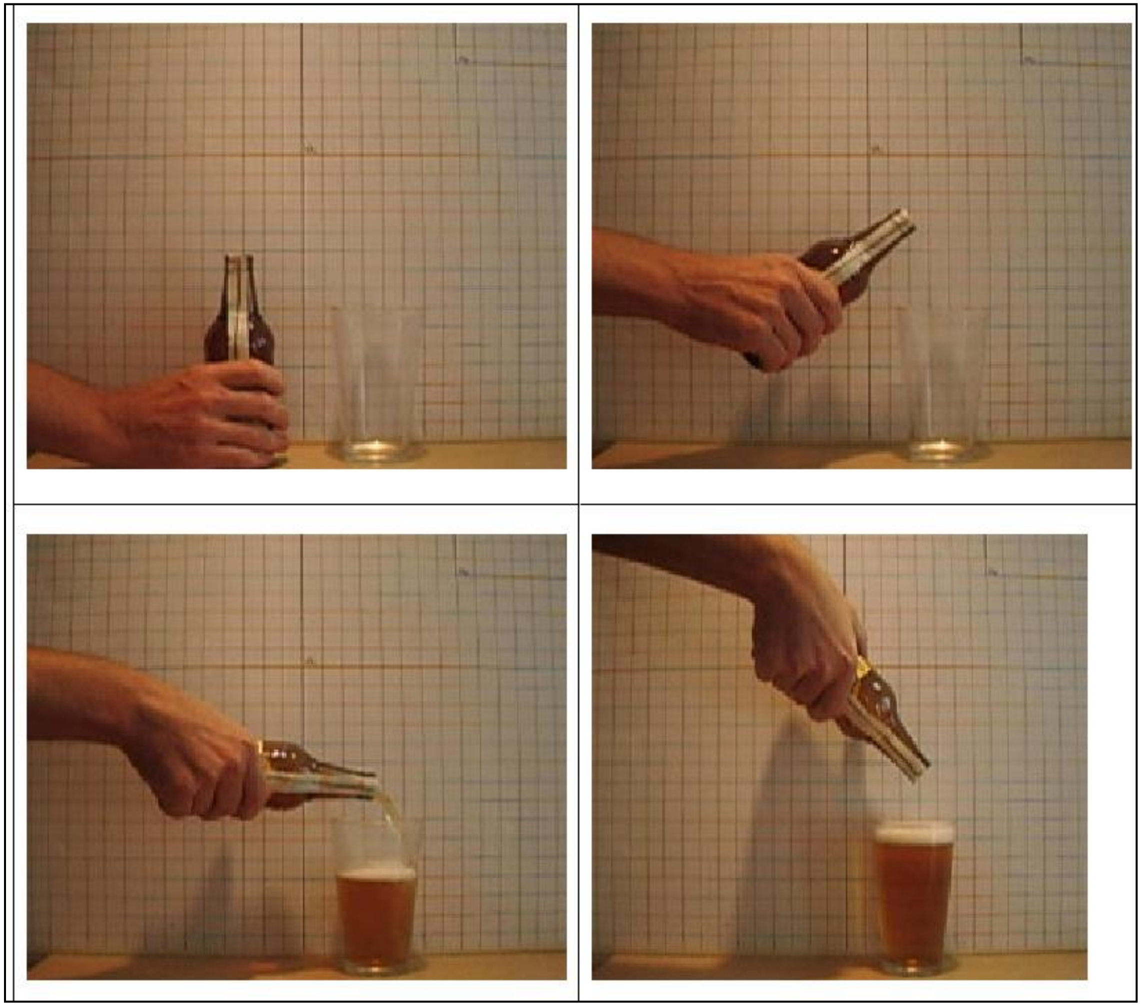


# Back-Drive the Leg with Link cd

```
instant = 0.0001;  % pause between frames
leg=[f g h a b c b b+0.05*Rp(-pi/2,b)*(h-b) h+0.05*Rp(pi/2,h)*(b-h) h b c d c
e e+0.1*Rp(-pi/2,e)*(c-e) c+0.1*Rp(-pi/2,c)*(b-c) b+0.1*Rp(pi/2,b)*(c-b) b];
p = plot(leg(1,:),leg(2,:),'o-b',...
    'EraseMode', 'normal');
axis equal
axis([-2.5 3.5 -4.5 1.5]);
for gamma=0:-0.5*pi/180:-50*pi/180
  c2=Rp(gamma,d)*c;
  link1=@(phi) norm(b-c)-norm(Rp(phi,a)*b-c2);
  phi=fzero(link1,0);
  b2=Rp(phi,a)*b;
  h2=Rp(phi,a)*h;
  link2=@(theta) norm(g-h)-norm(Rp(theta,f)*g-h2);
  theta=fzero(link2,0);
  g2=Rp(theta,f)*g; leg=[f g2 h2 a b2 c2 d c2 e2];
  link3=@(beta) norm(c2-Rp(beta,b2)*T(b2-b)*c);
  beta=fsolve(link3,0,options);
  e2=Rp(beta,b2)*T(b2-b)*e;
 leg=[f g2 h2 a b2 c2 b2 b2+0.05*Rp(-pi/2,b2)*(h2-b2) h2+0.05*Rp(pi/2,h2)*(b2-
h2) h2 b2 c2 d c2 e2 e2+0.1*Rp(-pi/2,e2)*(c2-e2) c2+0.1*Rp(-pi/2,c2)*(b2-c2)
b2+0.1*Rp(pi/2,b2)*(c2-b2) b2];
  set(p,'XData',leg(1,:), 'YData',leg(2,:))
  pause(instant)
end
```



# A Proper Pour

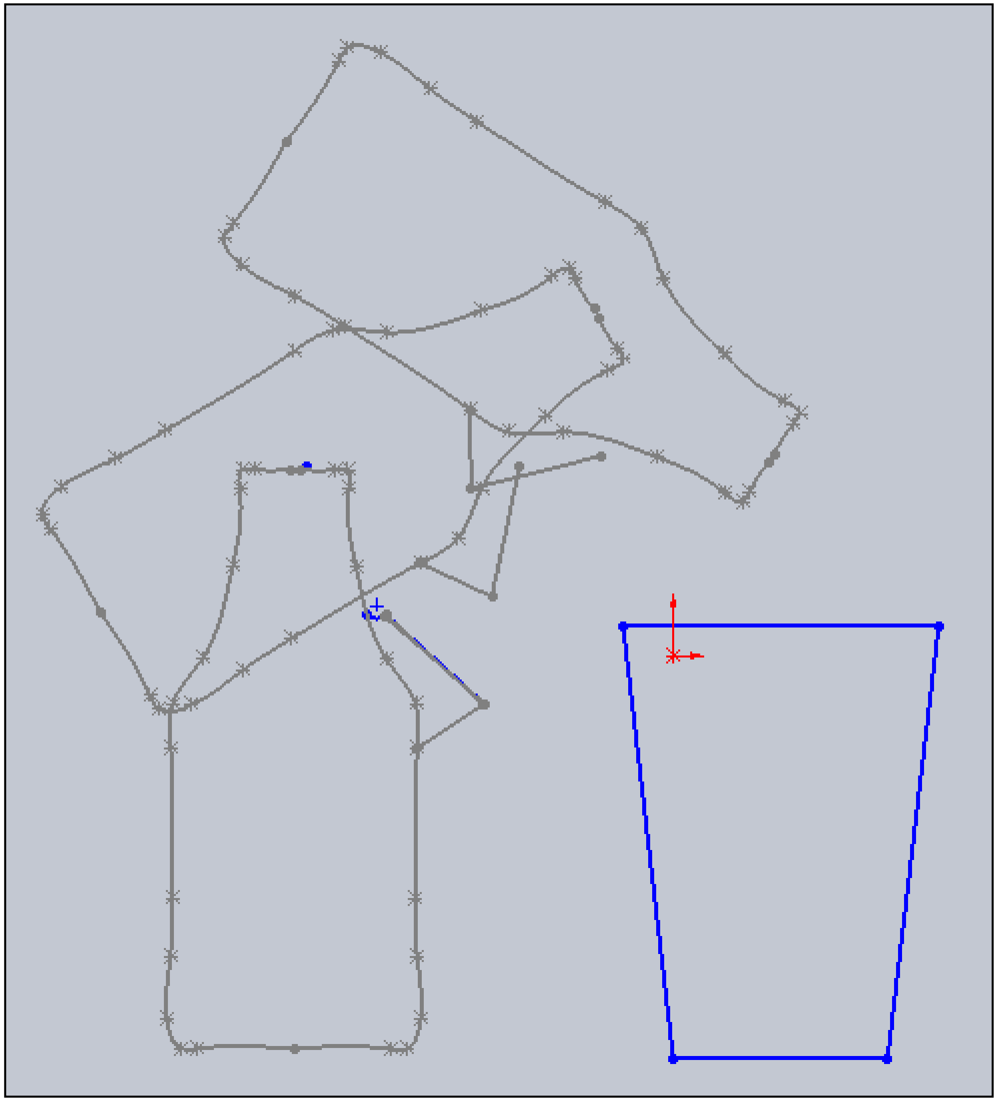




Source: Ridley, P. "Teaching Mechanism Design." *Proceedings of the 2006 Australasian Conference on Robotics & Automation*, December 6 - 8, Auckland, New Zealand.

# 3 Position Synthesis

- Say we want a mechanism to guide a body in a prescribed way
- Pick 3 positions
- Pick two attachment points
- The 4 bar mechanism can be constructed graphically



# Import the Desired Motions into Matlab

```
% dxf2coord 1.1 matrix
% author: lukas wischounig, innsbruck, austria (dept. of geology,
% university innsbruck), email: csad0018@uibk.ac.at
% date: may 2005
% filename: dxf2coord_11_matrix.m


figure(1)
hold on
axis equal

x=2;y=3;
for i=1:7
   plot(polylines(find(polylines(:,1)==i),x),polylines(find(polylines(:,1)==i),y))
end
plot(points(:,x)+0.2,points(:,y)-0.2,'ro')

names=char('1a', '2a', '3a','3c',  '2b', '1b');
for i=1:6
   text(points(i,x), points(i,y), names(i,:))
end
```

# Use a Function Minimizer to Make the Mechanism More Compact

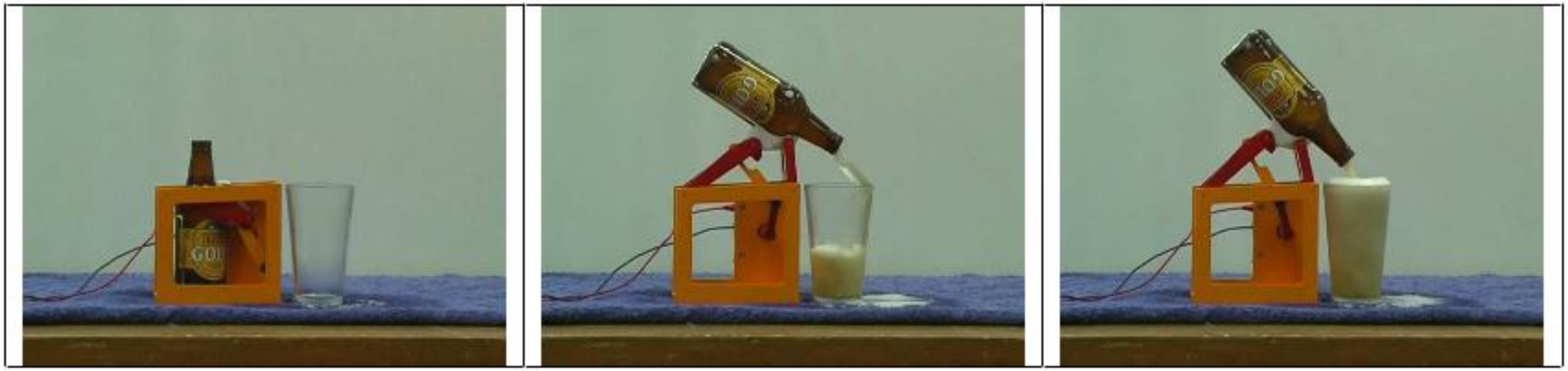
```
x_link=-6;
sum_sqr_err=@(input) sum([(input(2)-norm([x_link input(1)]-points(1,x:y)))^2 ...
    (input(2)-norm([x_link input(1)]-points(3,x:y)))^2 ...
    (input(2)-norm([x_link input(1)]-points(3,x:y)))^2]);
best=fminsearch(sum_sqr_err,[-1 4]);
bext_y=best(1);
best_rad=best(2);
```

Force the joint to lie along this line

Find the
y location and radius
of a circle that fits the
commanded points
1a 2a 3a in a least
squares sense.

# Automating the Task

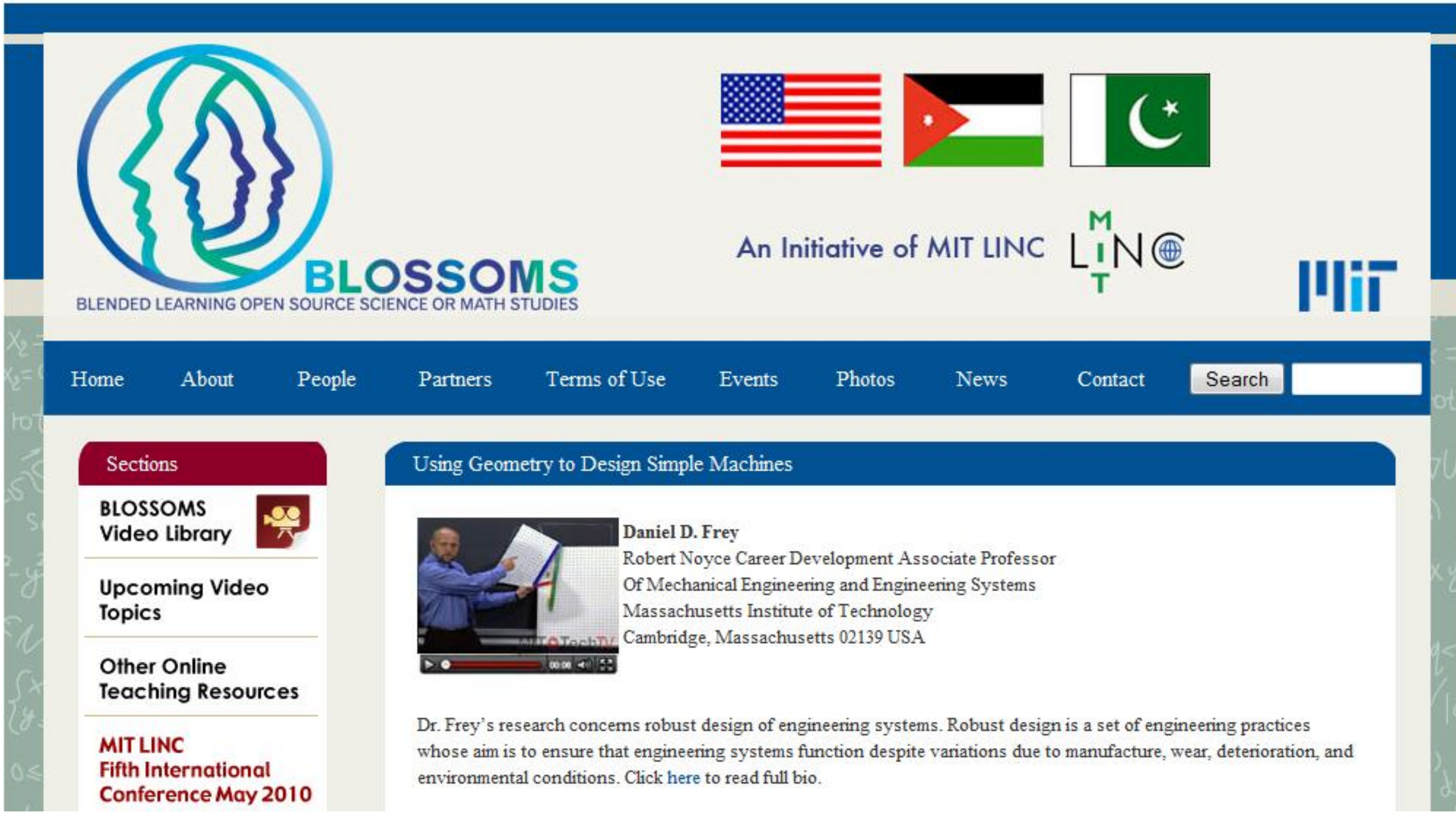
- Very simple mechanically
- Nicely compact
- Springs could allow the servo to be loaded uniformly on the up and down stroke





Source: Ridley, P. "Teaching Mechanism Design." *Proceedings of the 2006 Australasian Conference on Robotics & Automation*, December 6 - 8, Auckland, New Zealand.

# A Video on 3 Pos'n Synthesis

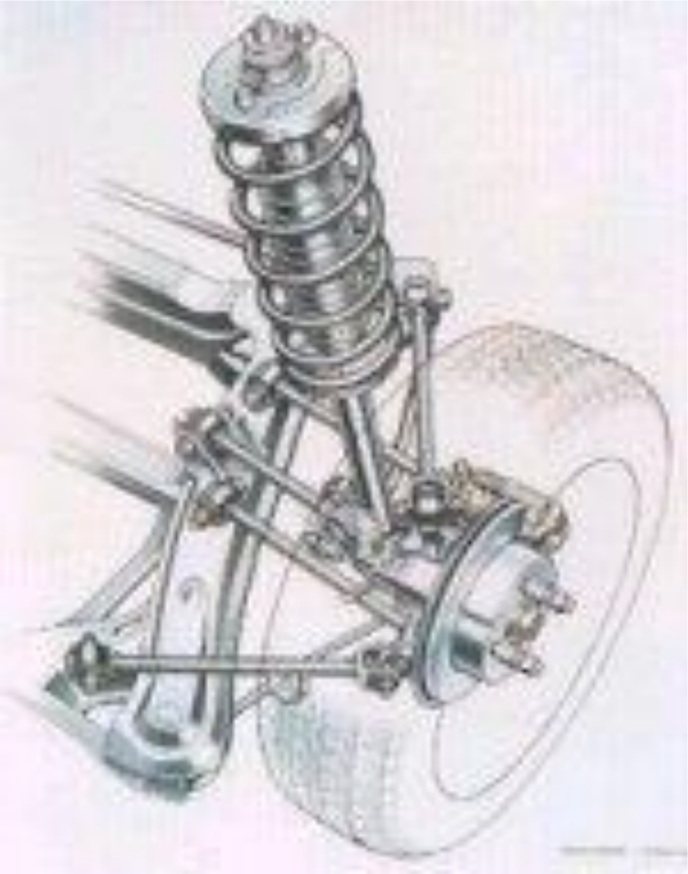



http://blossoms.mit.edu/video/frey.html

# What about 3D Mechanisms?

I the past, we didn't emphasize
3D mechanisms in the subject 2.007

But thanks to 2.086, now maybe we can!



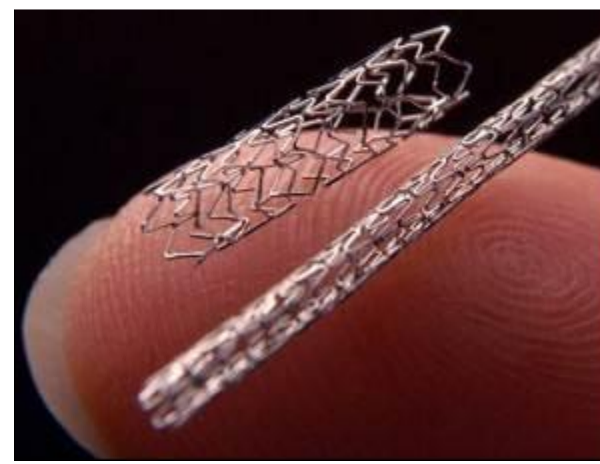


Rear suspension of a Honda Accord

# An Idea for a 2.007 "Medical" Scoring Challenge

- Angioplasty -- mechanically widening narrowed or obstructed arteries (to correct effects of atherosclerosis)

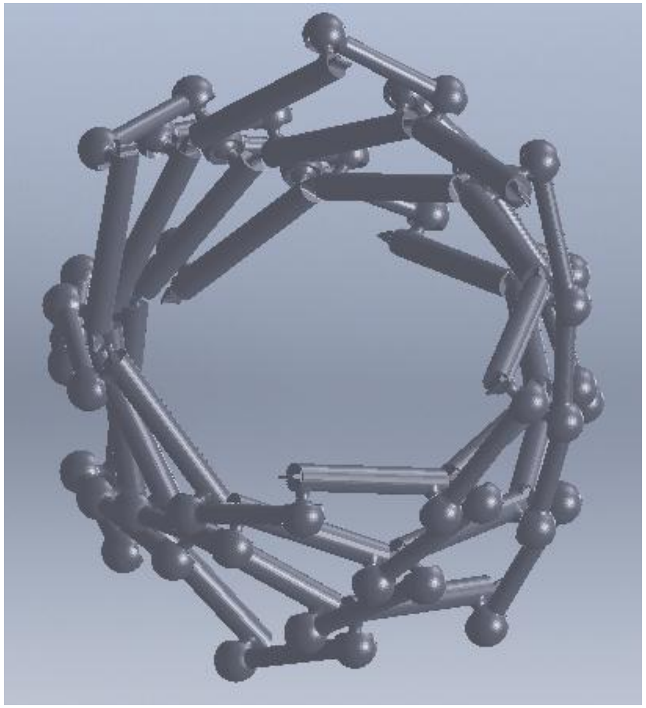
- A balloon on a guide wire is placed and then inflated
- The balloon crushes the fatty deposits
- A stent may be used so the vessel remains open

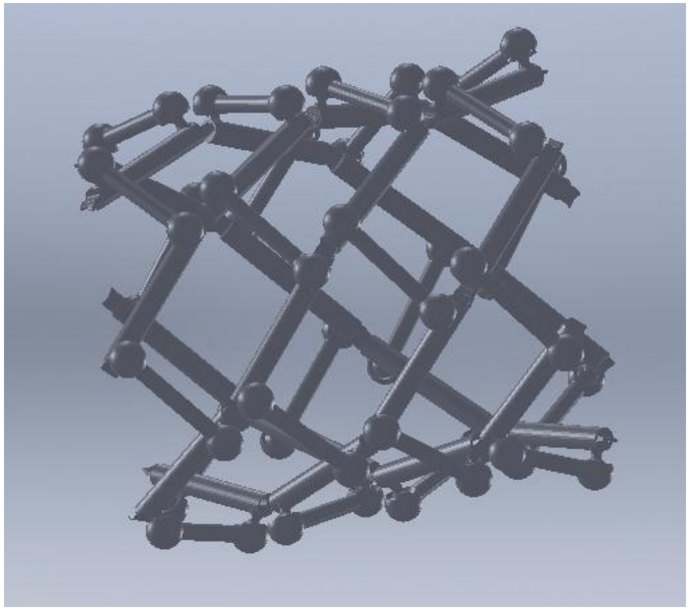


# What I Want to Design

- Scaled up stent
- Starts narrow and can be expanded with moderate pressure
- Kinematic freedom similar to actual stent
- Relatively thin in radial direction
- Stays open after expansion
- Can be reused many times
- Teach about 3D mechanism design

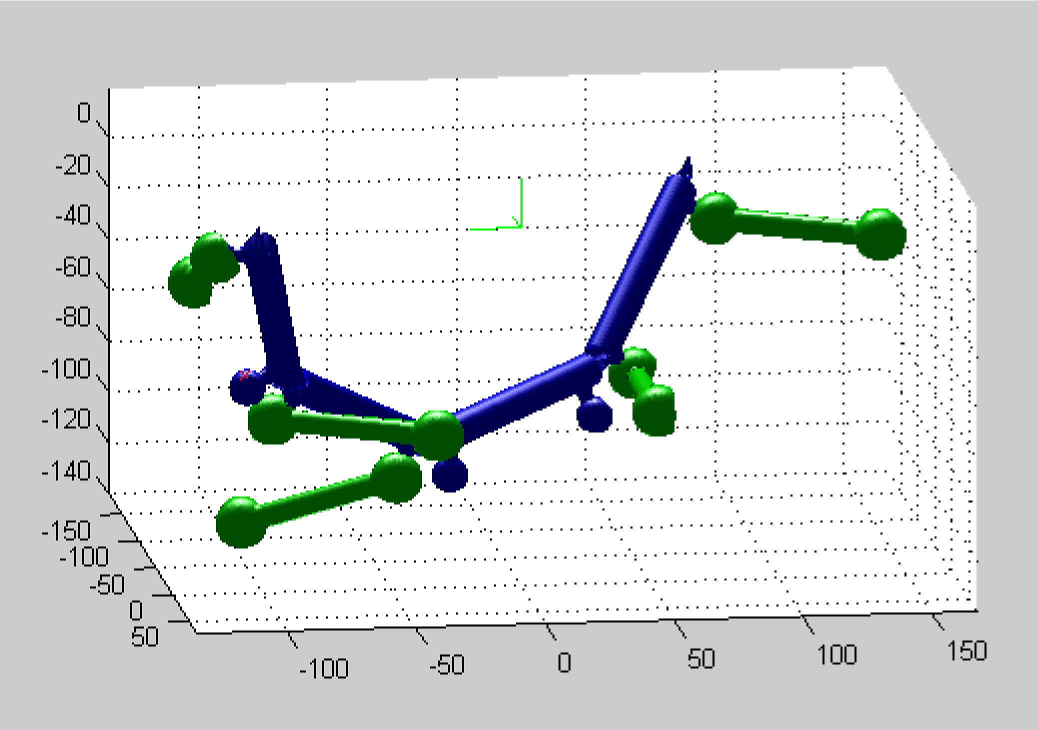
# First Cut at the 3D Mechanism





# Using fsolve to Animate the Stent

- Can I import the 3D CAD into Matlab?
- Can I control the motions of the objects?



# Questions?