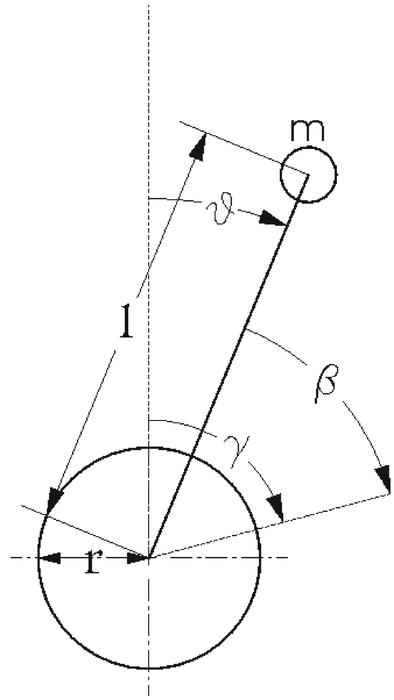
# Balancing the Segway

Charles Masenas
cmasenas@alum.rpi.edu
http://nxtotherway.webs.com

## Objective

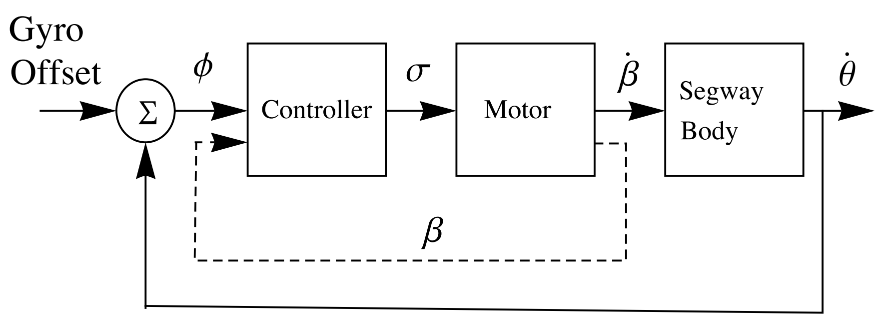
**To demonstrate that using available software to solve systems of differential equations, it is not necessary to know elaborate control theory to solve the two wheel balancing problem.**

## Segway Body




- **Point mass, m**
- **Tilt angle $\theta$ from vertical reference**
- **Wheel angle $\beta$ from arm reference**

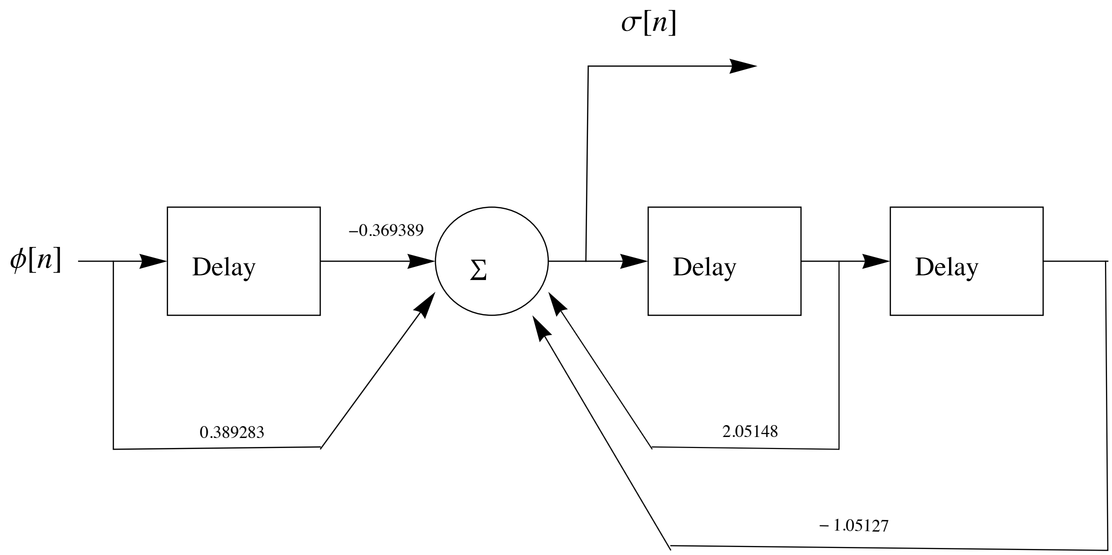
# Loop Model




- **Strategy for Solution**
  - Model the Motor and the Segway Body
  - Design the Controller

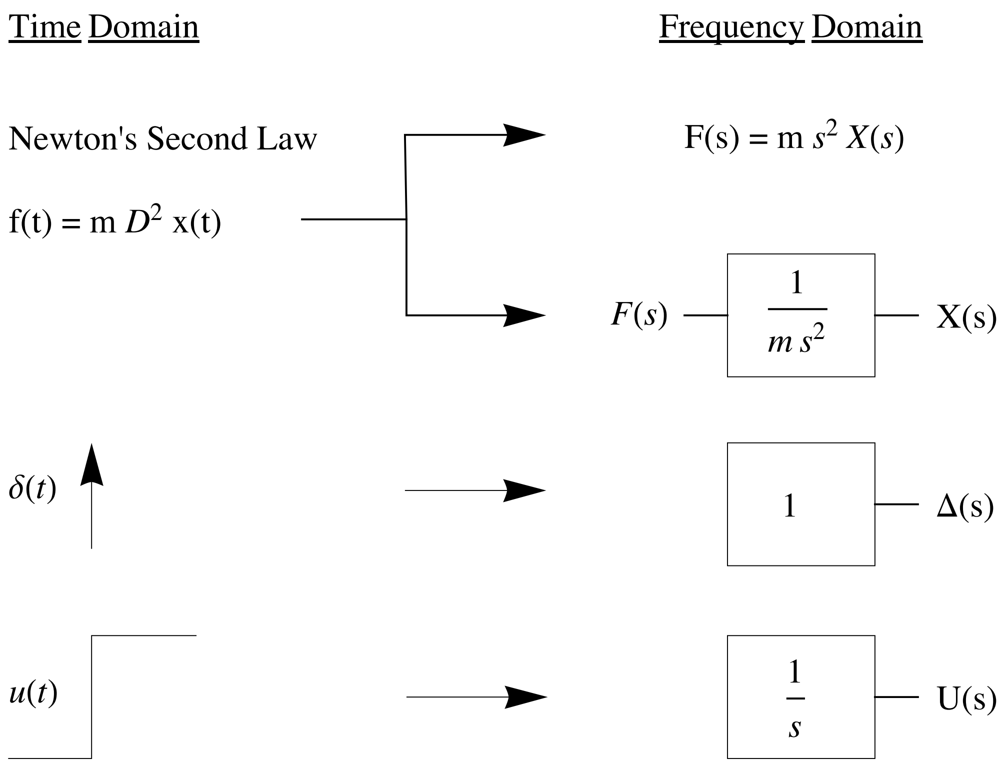
## Software Implementation of the Controller

```
while ( balancing  )
 {
  PHI[2] = ReadGyro( )   ;    // read the gyro, tilt angle change per time, deg/sec

  // This line implements the controller as a digital filter
  SIGMA[2]  =  −0.369389 PHI[0] +  0.389283  PHI[1] - 1.05127 SIGMA[0] + 2.05148 SIGMA[1] ;

 // shift the registers in preparation for the next cycle
  SIGMA[0] = SIGMA[1] ;
  SIGMA[1] = SIGMA[2] ;
  PHI[0] = PHI[1] ;

  SetMotorSpeed( SIGMA[2]  );

  Wait( 12 milliseconds);
 }
```



$\sigma[n]$
$\phi[n]$
Delay
−0.369389
Σ
Delay
Delay
0.389283
2.05148
− 1.05127

## 5 Minute Frequency Domain Speed Date

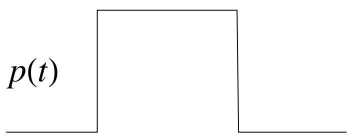


# 5 Minute Frequency Domain Speed Date (Continued)

$$g(t-T) \longrightarrow G(s)\, e^{-s\,T}$$

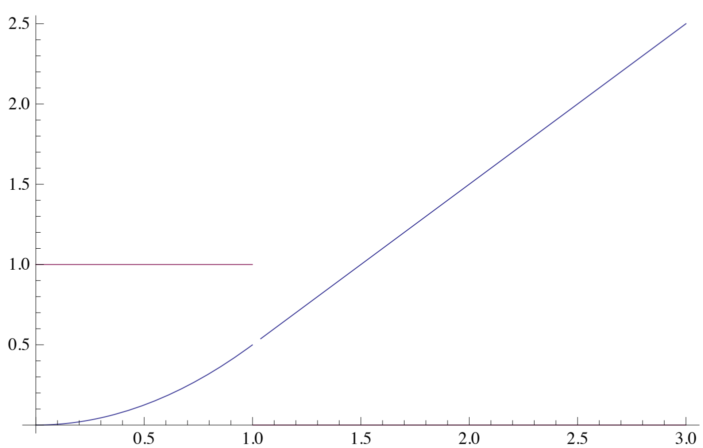
Note: In the discrete-time frequency domain, $z \equiv e^{s\,T}$ is the independent variable.

$$g(n\,T) \longrightarrow g_n\, z^{-n}$$

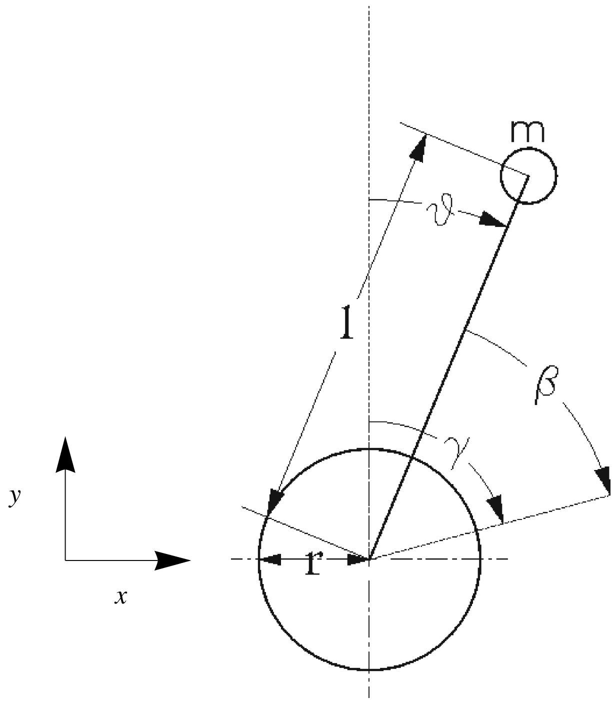
$p(t)$  $\longrightarrow$ $\boxed{\dfrac{1-e^{-\mathrm{sT}}}{s}}$ — P(s)

$\boxed{\dfrac{1-e^{-\mathrm{sT}}}{s}}$ — $\boxed{\dfrac{1}{m\,s^2}}$ — X(s)

In[26]:=
```
Plot[{InverseLaplaceTransform[(1 - E^(-s 1))/s 1/(1 s^2), s, t],
  InverseLaplaceTransform[(1 - E^(-s 1))/s, s, t]}, {t, 0, 3}]
```

Out[26]=


## Modeling of Segway Body
### (Lagrangian Mechanics)




Coordinates of point mass “m”...

$$x(t) = r\,[\theta(t) + \beta(t)] + l\,\sin(\theta(t))$$

$$y(t) = l\,\cos(\theta(t))$$

Kinetic energy, T...

$$T = \frac{1}{2}\,m\left(\left[\frac{d}{dt}x(t)\right]^2 + \left[\frac{d}{dt}y(t)\right]^2\right)$$

Potential energy, V...

$$V = m\,g\,y(t)$$

## Modeling of Segway Body
### (Lagrangian Mechanics)

- **Lagrangian, L**
  - $L = T - V$
  - $L = \frac{1}{2} m \left( [l\, \theta'(t) \sin(\theta(t))]^2 + [l\, \theta'(t) \cos(\theta(t)) + r\,(\beta'(t) + \theta'(t))]^2 \right) - m\, g\, l \cos(\theta(t))$
- **Lagrangian equation for each coordinate ($\theta$, $\beta$).**
  - $\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} = \tau_\theta$
  - $\frac{d}{dt} \frac{\partial L}{\partial \dot{\beta}} - \frac{\partial L}{\partial \beta} = \tau_\beta$
  - Torque on arm = - Torque on wheel
  - $\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} = -\left( \frac{d}{dt} \frac{\partial L}{\partial \dot{\beta}} - \frac{\partial L}{\partial \beta} \right)$
- **Result**

this equation describes the Segway body (relating $\theta$(t) to $\beta$(t))

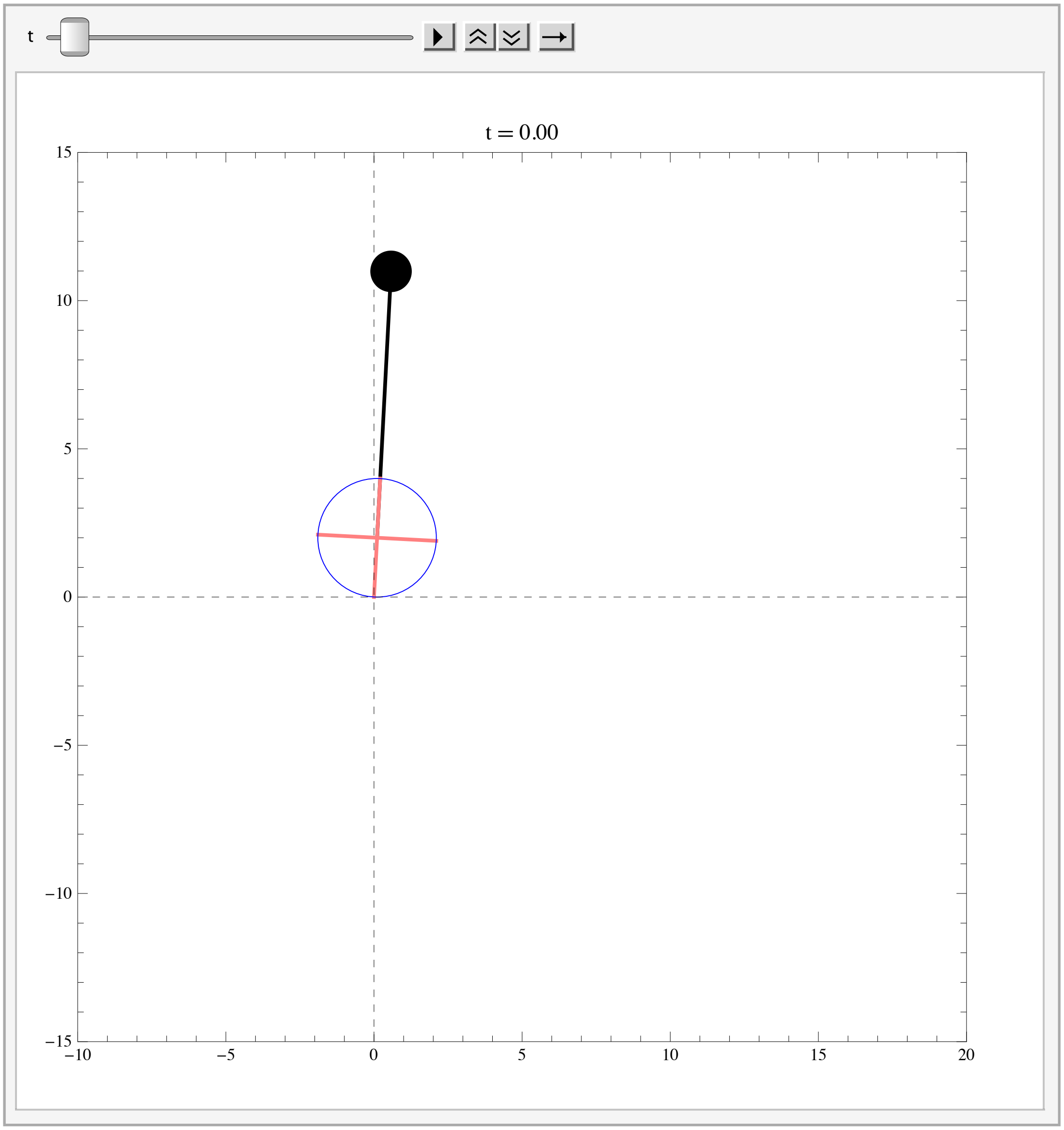
$$l \sin(\theta(t)) \left( g + 2\, r\, \theta'(t)^2 \right) = \theta''(t) \left( l^2 + 3\, l\, r \cos(\theta(t)) + 2\, r^2 \right) + r\, \beta''(t)\, (l \cos(\theta(t)) + 2\, r)$$

- **Linearized Result**

$$\frac{\Theta(s)}{\mathrm{B}(s)} = \frac{-\frac{r}{l+r}\, s^2}{s^2 - \frac{g\, l}{(l+r)\,(l+2\, r)}}$$

# Simulation of Body Model (no feedback control)

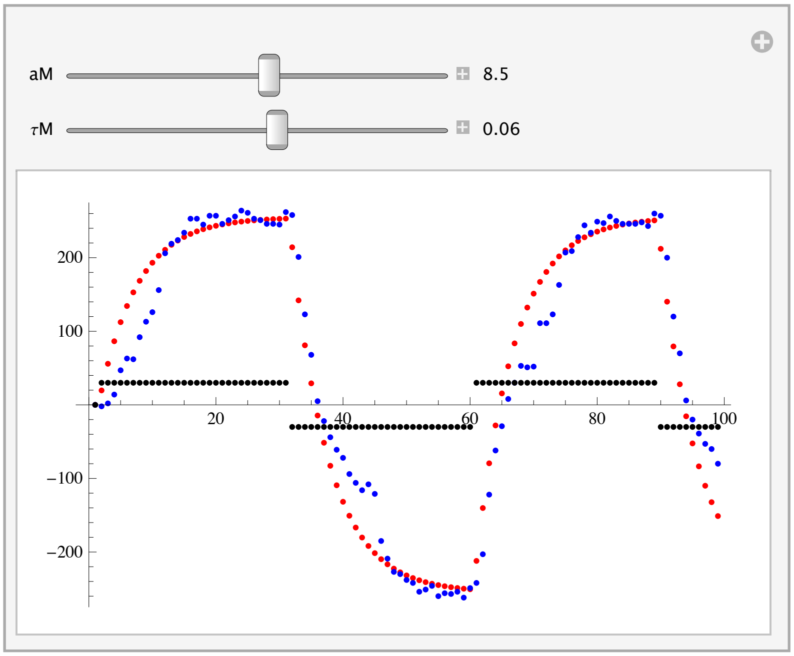
Out[28]=



# Modeling of NXT Motor

- **Assume single pole response**
  - $\beta'(t) + \tau \mathrm{M}\, \beta''(t) = \mathrm{aM}\, \sigma(t)$ (this equation describes the motor)

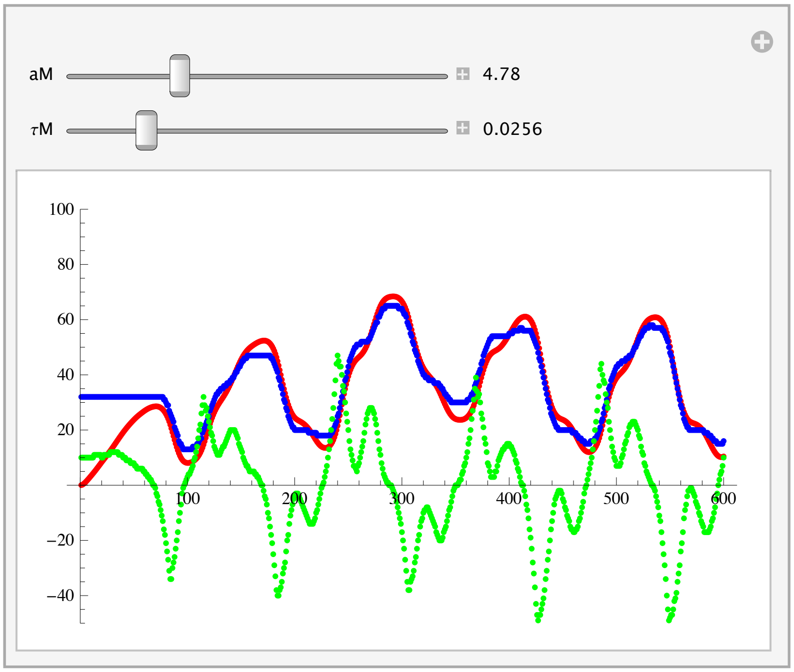
Out[29]=




Unloaded motor output, $\dot{\beta}$ (deg/sec), 10 ms interval
Blue: Data
Red: Model
Black: Motor input, $\sigma$ (30 unit step)

## Modeling of NXT Motor (continued)

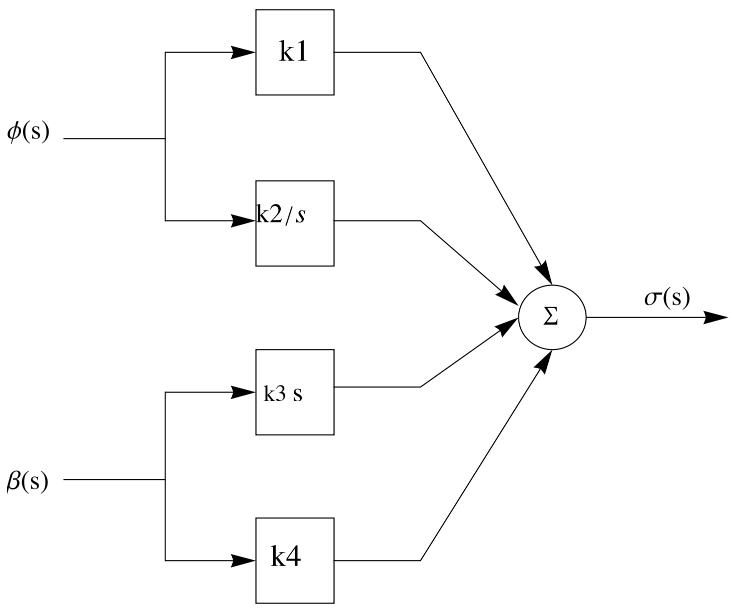
Out[30]=




Motor output, $\beta$ (deg), measurement under actual balancing conditions
Blue: Data
Red: Model
Green: Input, $\sigma$

# Controller Design

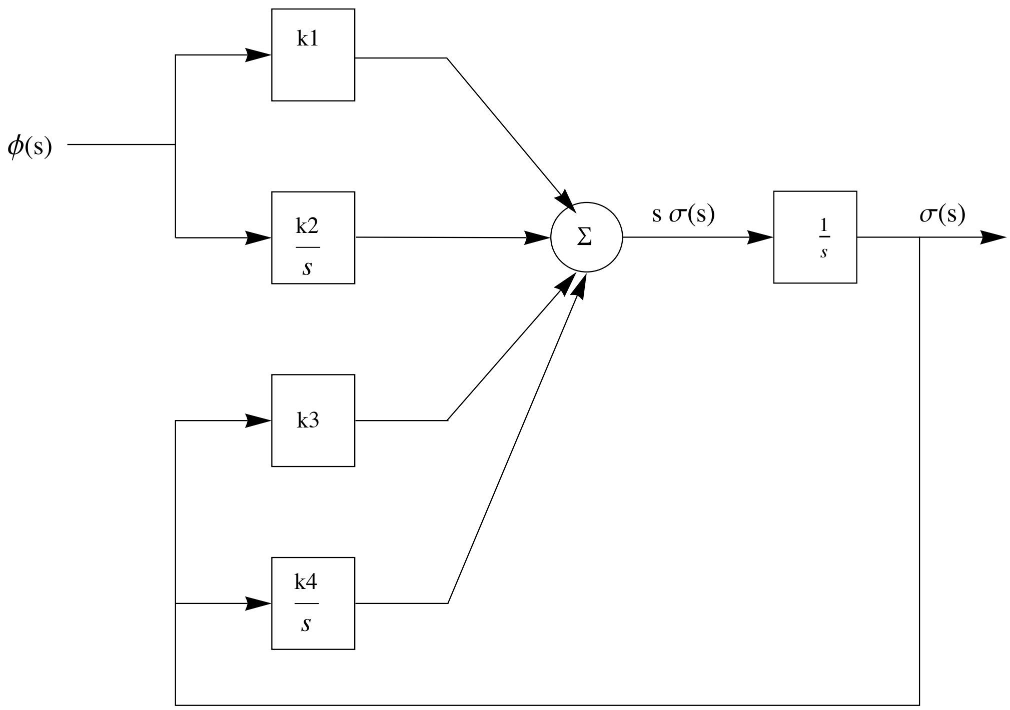
- **“NXTWay”**




- $\sigma(\mathrm{s}) = \mathrm{k1}\ \phi(\mathrm{s}) + \frac{\mathrm{k2}}{s}\ \phi(\mathrm{s}) + \mathrm{k3}\ s\ \beta(s) + \mathrm{k4}\ \beta(\mathrm{s})$

# Controller Design

- **“AnotherWay”**



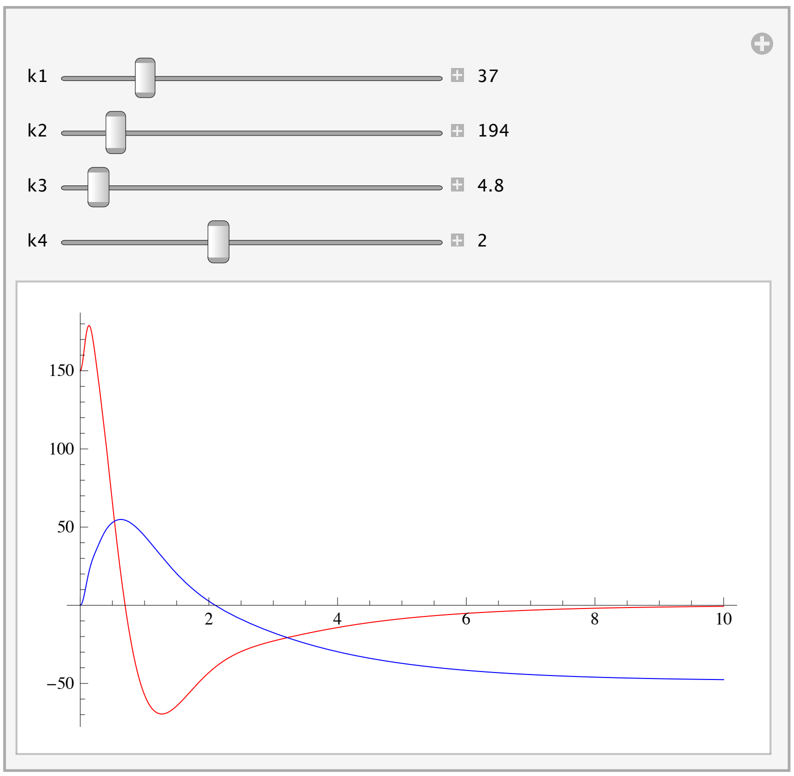

- $s\,\sigma(s) = k1\,\phi(s) + \frac{k2}{s}\,\phi(s) + k3\,\sigma(s) + \frac{k4}{s}\sigma(s)$

## Controller Design (Continued)

- Simulation of differential equations describing the system...

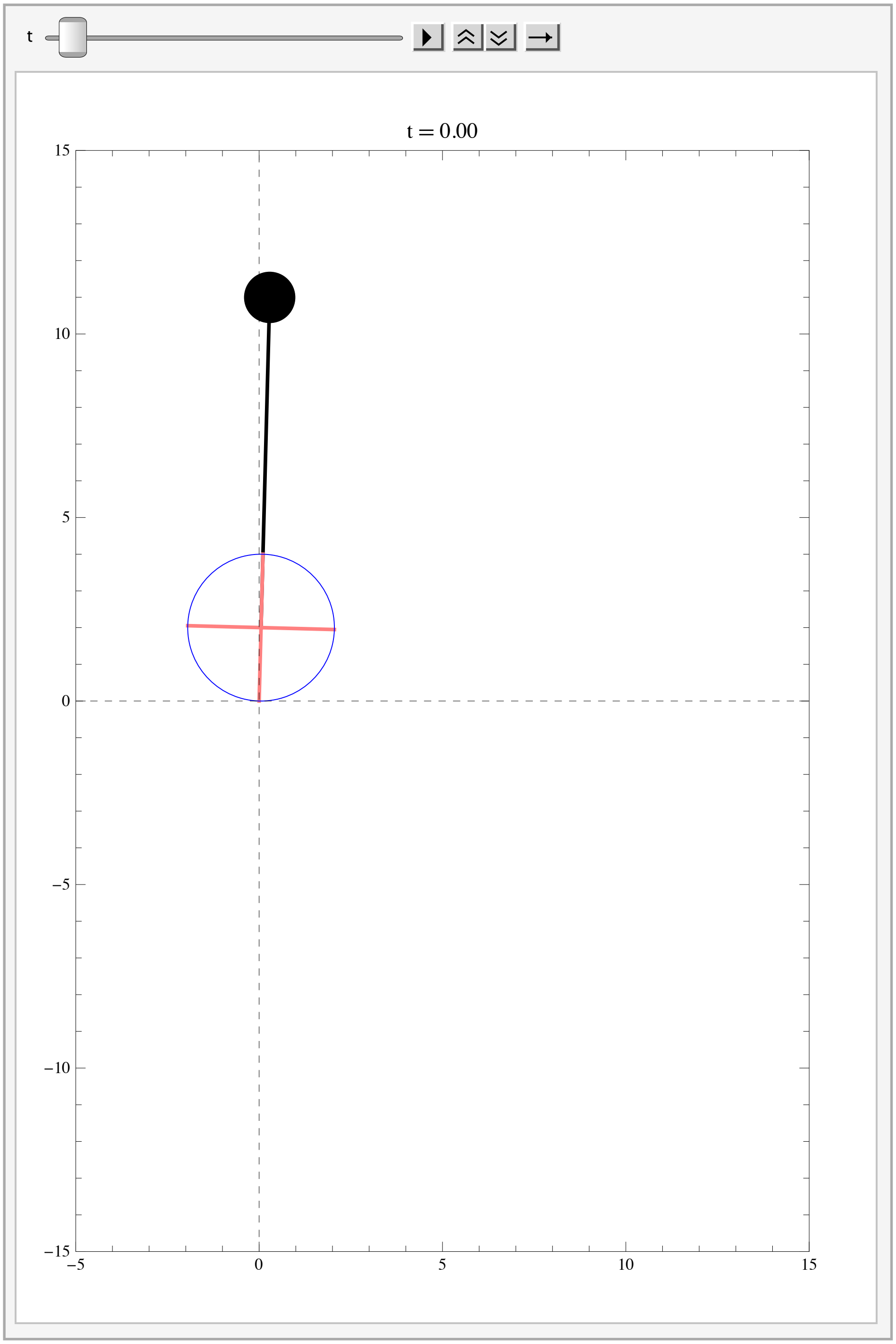
```
l Sin[θ[t]] (g + 2 r θ′[t]^2) =
   r (2 r + l Cos[θ[t]]) β′′[t] + (l^2 + 2 r^2 + 3 l r Cos[θ[t]]) θ′′[t];
β′ (t) + τM β′′(t) = aM σ (t);
ϕ (t) = θ′[t] + offset ;
σ′′[t]  = k1 ϕ′[t] + k2  ϕ[t] + k3 σ′[t] + k4 σ[t];
```

Out[31]=



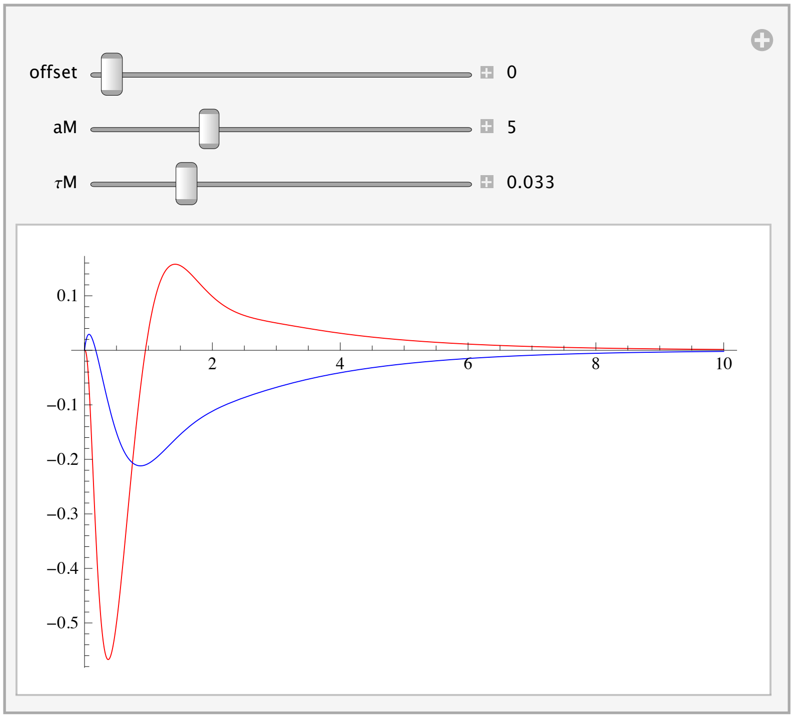

Continuous time simulation, .5deg/sec gyro offset, -1.5 deg/sec $\dot{\theta}$ offset
Red:  Tilt angle, $\theta$ (x 100)
Blue:  Controller output, $\sigma$

## Continuous Time Simulation



# Parameter Sensitivity Simulation

Out[33]=




Red: Tilt angle, $\theta$ (x 100)

Blue: Controller output, $\sigma$

## Digital Controller Implementation

- Differential Equation

$$\sigma''[t] = 194\,\phi[t] + 37\,\phi'[t] + 2\,\sigma[t] + 4.8\,\sigma'[t]$$

- Frequency Domain Gain

$$\frac{\sigma(s)}{\phi(s)} = \frac{194 + 37\,s}{-2 - 4.8\,s + s^2}$$

- Use software to convert to discrete time model
  - Zero Order Hold method is good (also called step invariant transform)
  - Mathematica function "ToDiscreteTimeModel", Matlab function "c2d"

$$\frac{\sigma(z)}{\phi(z)} = \frac{-0.369389 + 0.389283\,z}{1.05127 - 2.05148\,z + z^2}$$

- Solve for latest output term

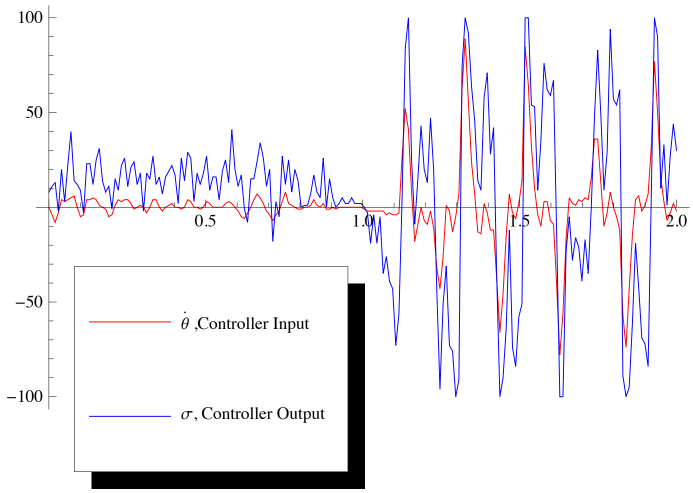
$$\sigma(z)\,z^2 = -0.369389\,\phi(z) + 0.389283\,z\,\phi(z) - 1.05127\,\sigma(z) + 2.05148\,z\,\sigma(z)$$

- Convert to time domain

This equation is implemented in the software

$$\sigma[n+2] = -0.369389\,\phi[n] + 0.389283\,\phi[n+1] - 1.05127\,\sigma[n] + 2.05148\,\sigma[n+1]$$

# NXTWay Balancing



## Conclusions

- **A controller using positive feedback is a feasible solution for balancing a two wheeled robot.**
- **Changing motor direction causes significant nonlinearity and noise.**
- **Sampling time had to be increased from the design value to achieve balance due to probable inaccuracy from approximations.**