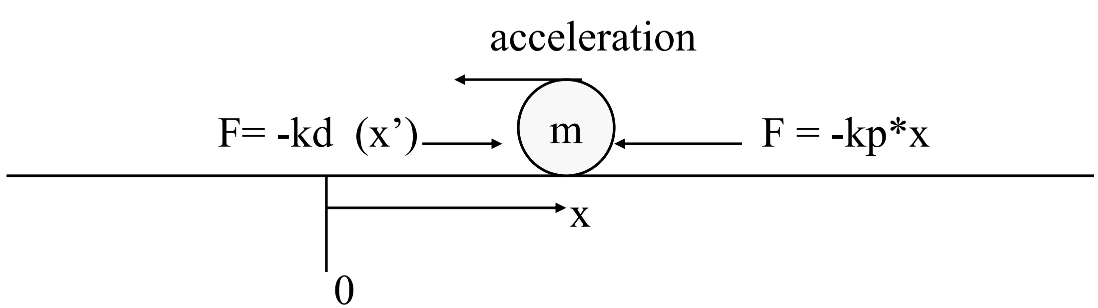
Announcement

- Last 4 lectures:
  - basic concepts of signals and filters
  - basic concepts of PID controllers

- Today
  - Using Matlab for signal processing and control
  - Combined Homework and MATLAB Exercises will be posted later. If you go to EWS at the Grainger Library during the lab sessions next week, Dan will help you with MATLAB.

Intuition

- When the position error, x, is positive, proportional control force is negative. It pushes the marble back to the origin (setpoint) at 0. What will be the force and the speed when marble reaches origin if we use only proportional control?
- When the marble is moving towards setpoint, the velocity is negative. So the force due to derivative control -kd*x' is positive. This counters the proportional force and slows down the marble's motion towards the origin.
- In summary, proportional control is like your car's gas pedal that moves the car towards the setpoint (where you want it to be) while derivation control is like your brake.



Modern State Space Control

- We have learned the effect of control is to change the eigenvalues, e.g., under proportional control of the marble, the real-part is zero. By adding the derivative term, the eigenvalues turn negative.

- Matrix is far more powerful tool to manipulate a linear system's eigenvalues. The use of the matrix method is called the state space method.

- The state of a dynamic system is its position and velocities along all the axes.

- (Recall that in the absence of external force, an object will keep its velocity. If it is rest, it will stay rest and if it is moving, it will keep moving. This is known as __________ law)

Marble on a Table in Matrix Form

The general form of linear control is: $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$,

where $\mathbf{x}$ is state and we let x_1 =position error, and x_2= error derivative

A is the coefficients of the PDE without control

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$ without external force, $\dot{x}$(i.e., x_2) unchange and $\ddot{x}$ (i.e. $\dot{x}_2$)$= 0$

$\mathbf{u} = -\mathbf{k}\mathbf{x}$, the control is a linear combination error and error derivative

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \begin{bmatrix} k_p & k_d \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\dot{\mathbf{x}} \quad = \quad \mathbf{A} \qquad \mathbf{x} \; - \; \mathbf{B} \qquad \mathbf{k} \qquad \mathbf{x}$$

where **B** maps external control into velocity and acceleration

Since $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}$, the effect of $\mathbf{u}$ is to change the system matrix from **A** to **(A − BK)**

A describes how the states will change when there is no control and **(A - BK)** tells us how the states will change with feedback control.

Using Matlab LQR

- One of the powerful and practical tools in control is the Linear Quadratic Regulator

LQR is designed to minimize the cost function

$$J = \int (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt$$

where $\mathbf{x}^T \mathbf{Q} \mathbf{x}$ is the square of errors weighted by $\mathbf{Q}$

$\mathbf{u}^T \mathbf{R} \mathbf{u}$ is the square of control signal weighted by $\mathbf{R}$

Using LQR to design a Marble controller

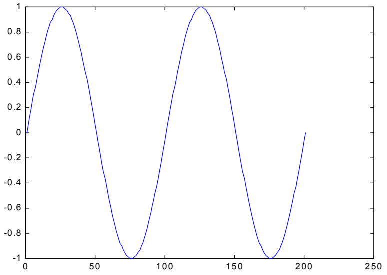
```
>>m = 0.01;  A = [0,1;0,0]; B = [0; 1/m];  Q=[1,0;0,1]; R = 1;
>>[K, S, E] = lqr(A, B, Q, R] %where k is the gains,  E are the eigenvalues
% Matlab will give us K = 1.0000, 1.0100 and E = -1.0001,   -99.9950
```

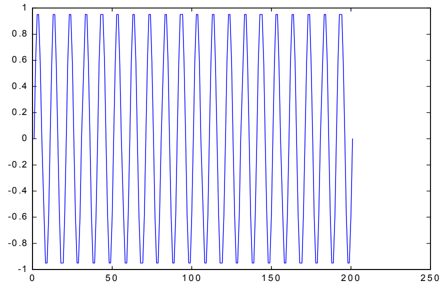
Signal + High Frequency Noise

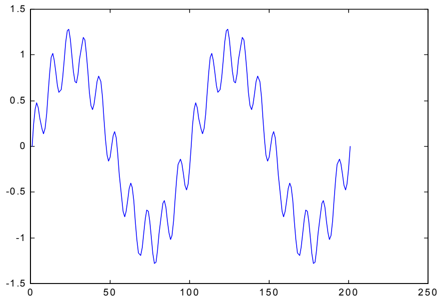
- Getting help: Go to Help/HelpDesk (HTML). Read getting started.

- Start Matlab program

- >>% duration of 2 seconds, using 0.01 sec step. Sampling rate is ______ Hz
- >>t = 0:0.01:2;   % semicolon means don't print the value of t
- >>y1   = sin(2*pi*t); % 1 Hz
- >>y10 = sin(20*pi*t); %
- >>plot(y1)
- >>plot(y10)
- >>z = y1+ 0.3*y10;
- >>plot(z)

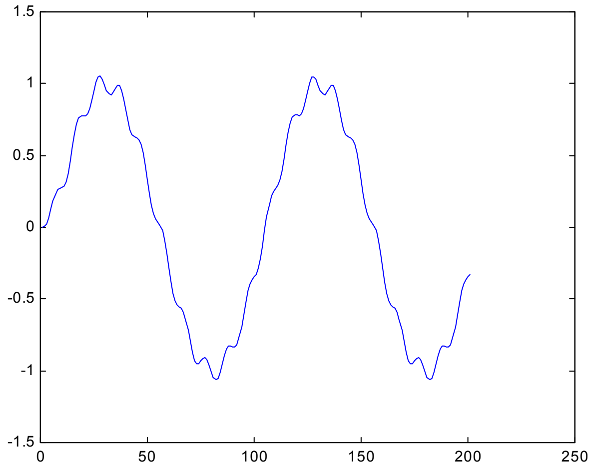


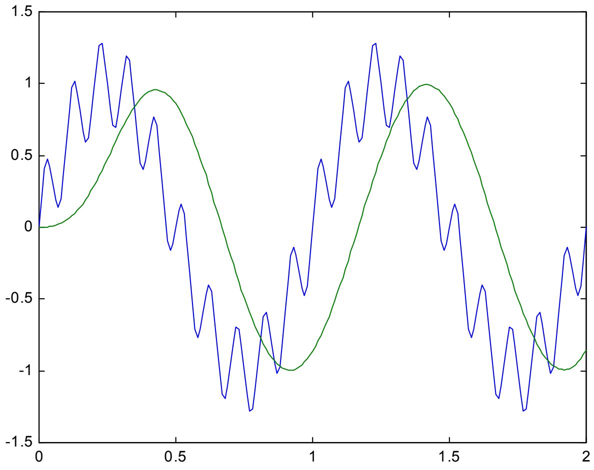




Low Pass Filter

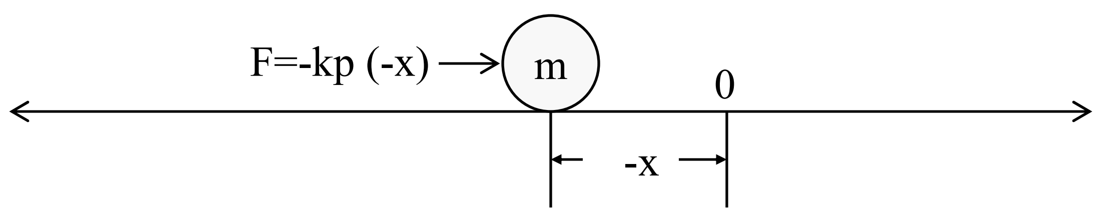
- >> [b, a] = butter(_____, _____)   % 2nd order, cutoff at 5 Hz
- >>filz = filter(b, a, z); % filtering z using the filter that we have just designed.
- >>plot(t, filz); % plotting filtered z using the format (x, function(x))
- >> [c, d] = butter(3, 2/50);
- >>filz2 = filter(c, d, z);
- >>plot(t, z, t, filz2) % putting two graphs on the same figure



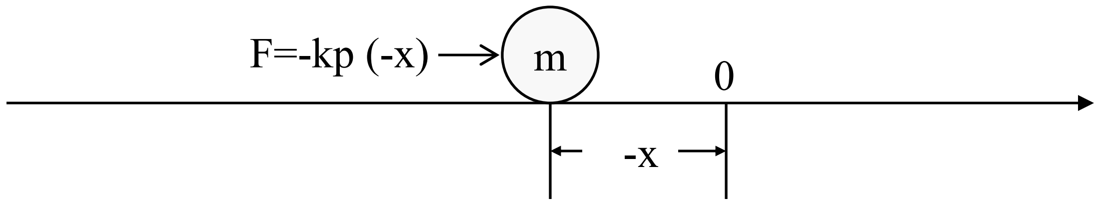


Simulate A Simple System with Proportional Control

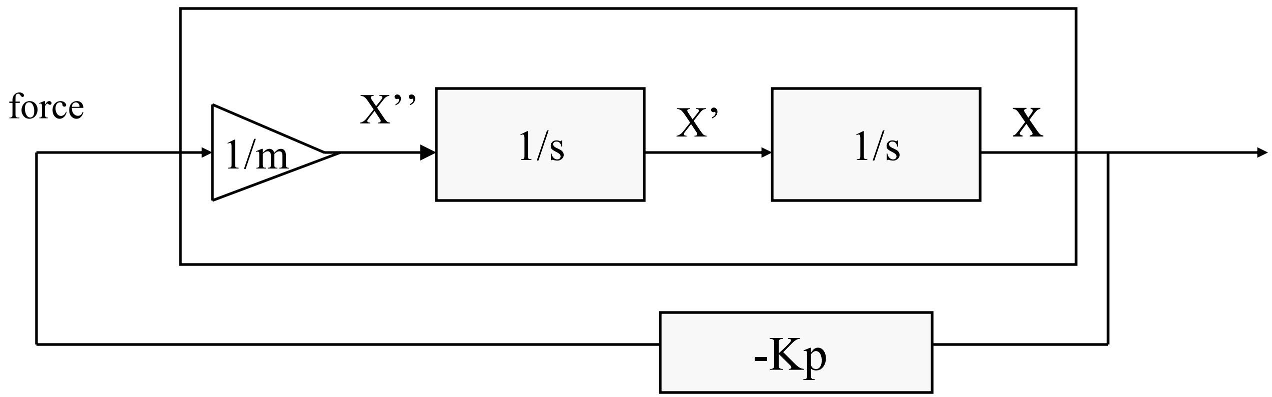
- Consider a marble on a flat and perfectly leveled table again.
  - Any point can be an equilibrium point (just pick one)
  - Its motion can be described by Newton's law F = ma, or x'' = F/m
  - suppose that we want to keep the marble at x = 0, by applying proportional control: $F = -k_p x$.
    - The feedback is negative since if the marble *position error* is negative, it pushes with a positive force and vice versa.
    - K_p is a positive integer known as proportional control constant.



Basic Simulation Diagram Concepts: Marble Under Proportional Control



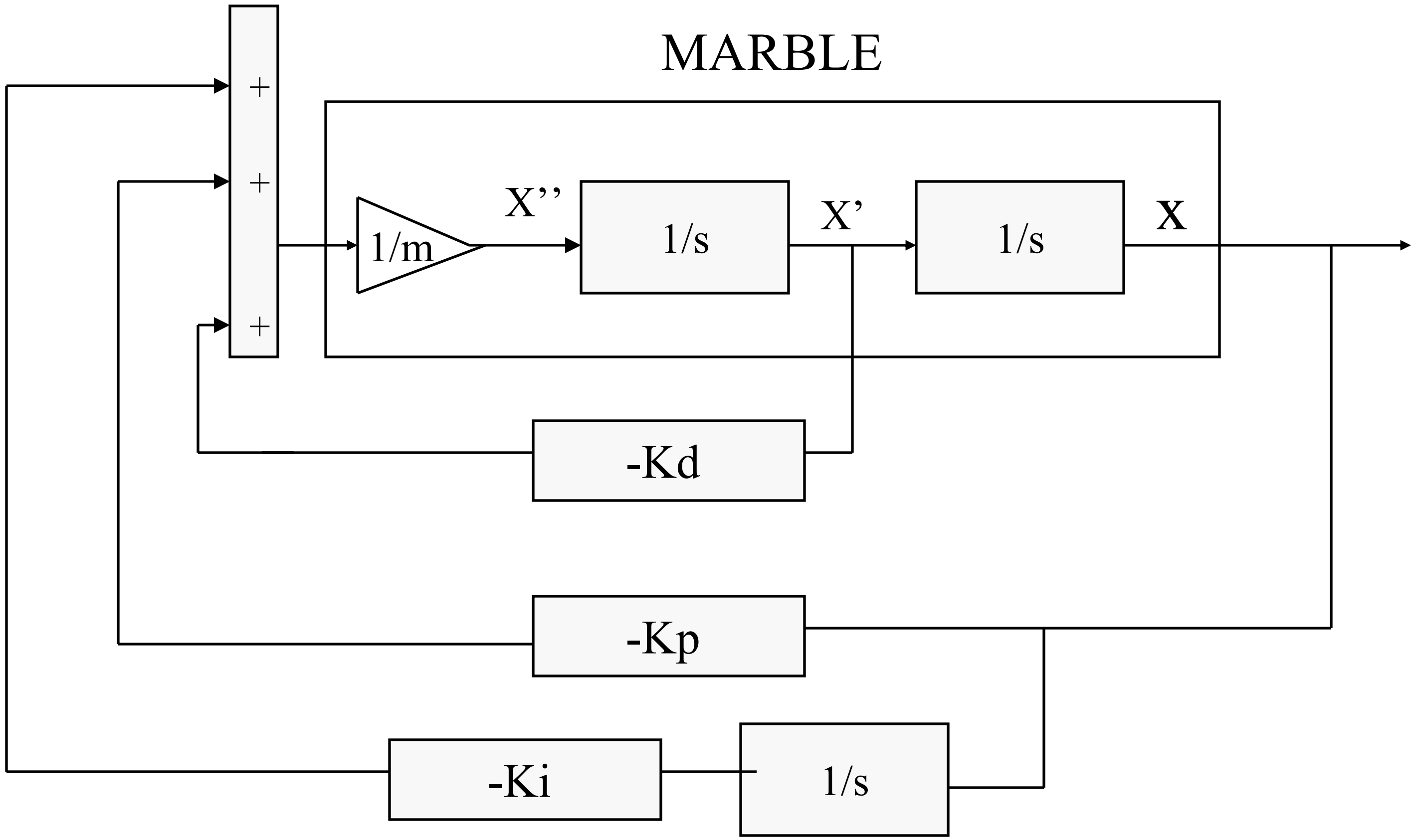
MARBLE



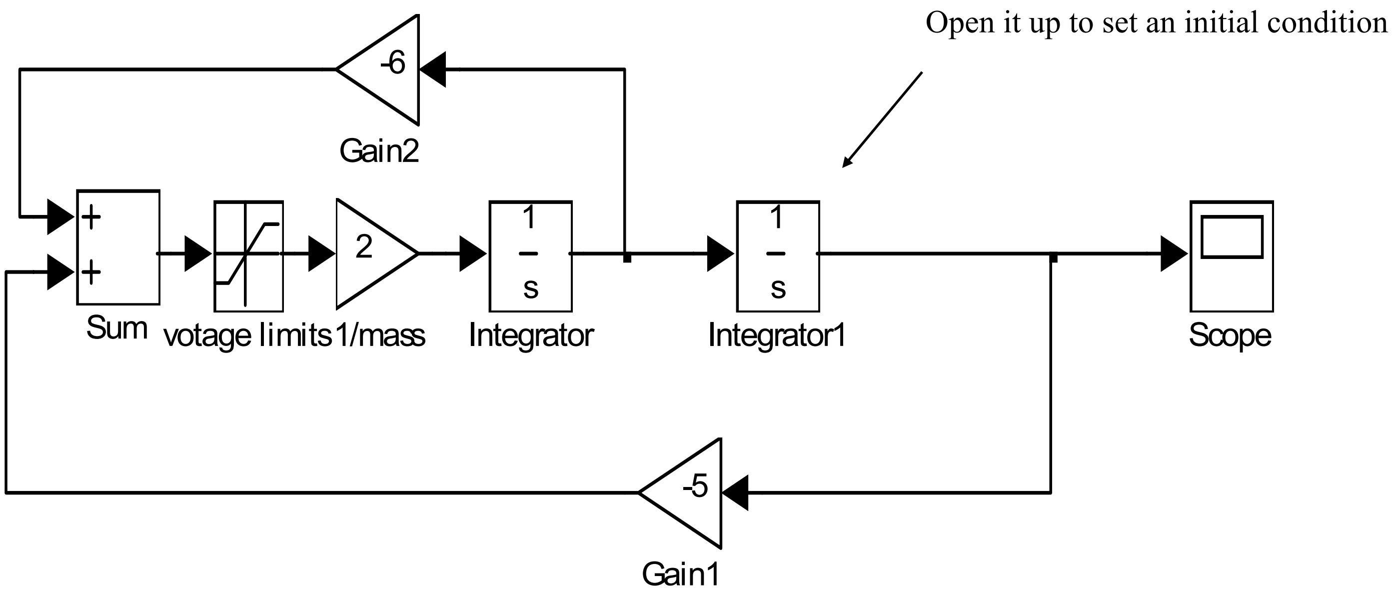
Mass is a double integrator with weight 1/m, which transforms force to acceleration, acceleration to velocity, and then velocity to position

1/s denotes integration

Marble Under PID Control



MATLAB Simulation Diagram

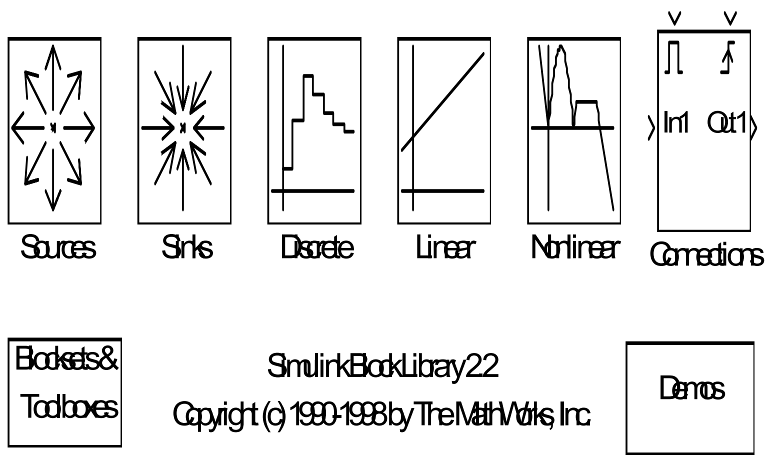


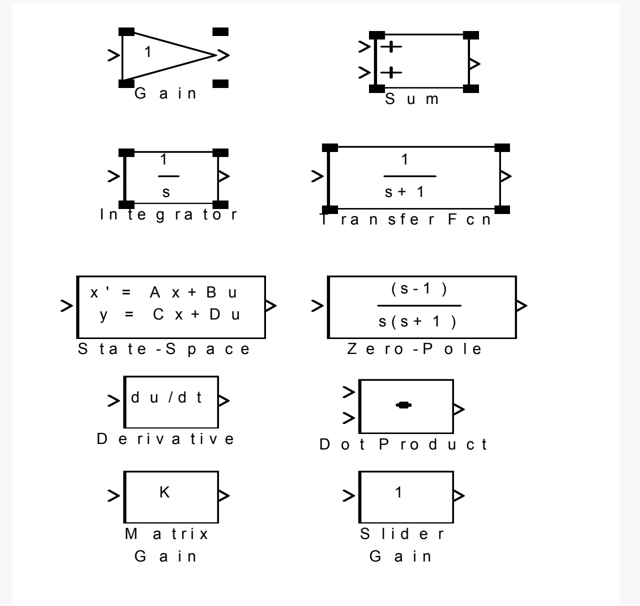
What type of controller is this?

What is Gain 1 called ? What is Gain 2 called?

Matlab Simulation

- >>simulink





Linear box

•Scope is in sink box

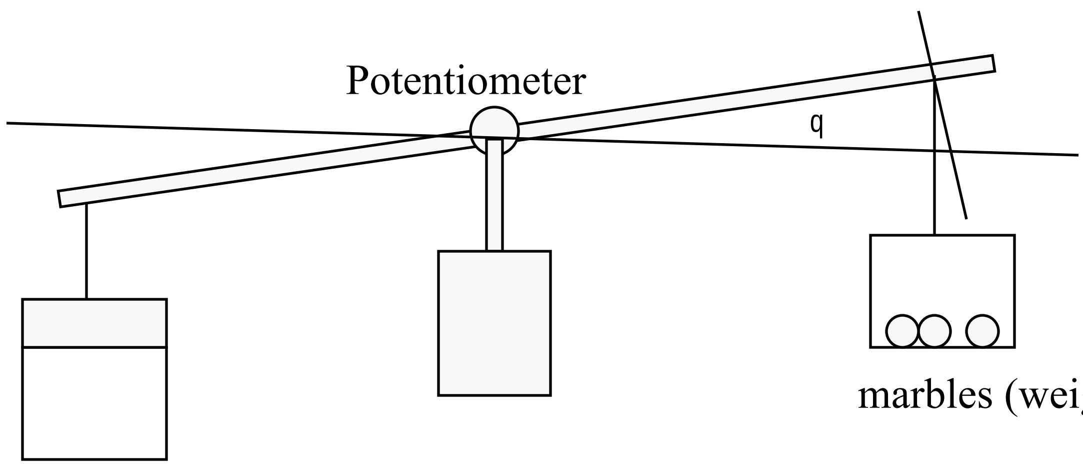
•saturation is in non-linear box etc

•double click to open a box to set parameters

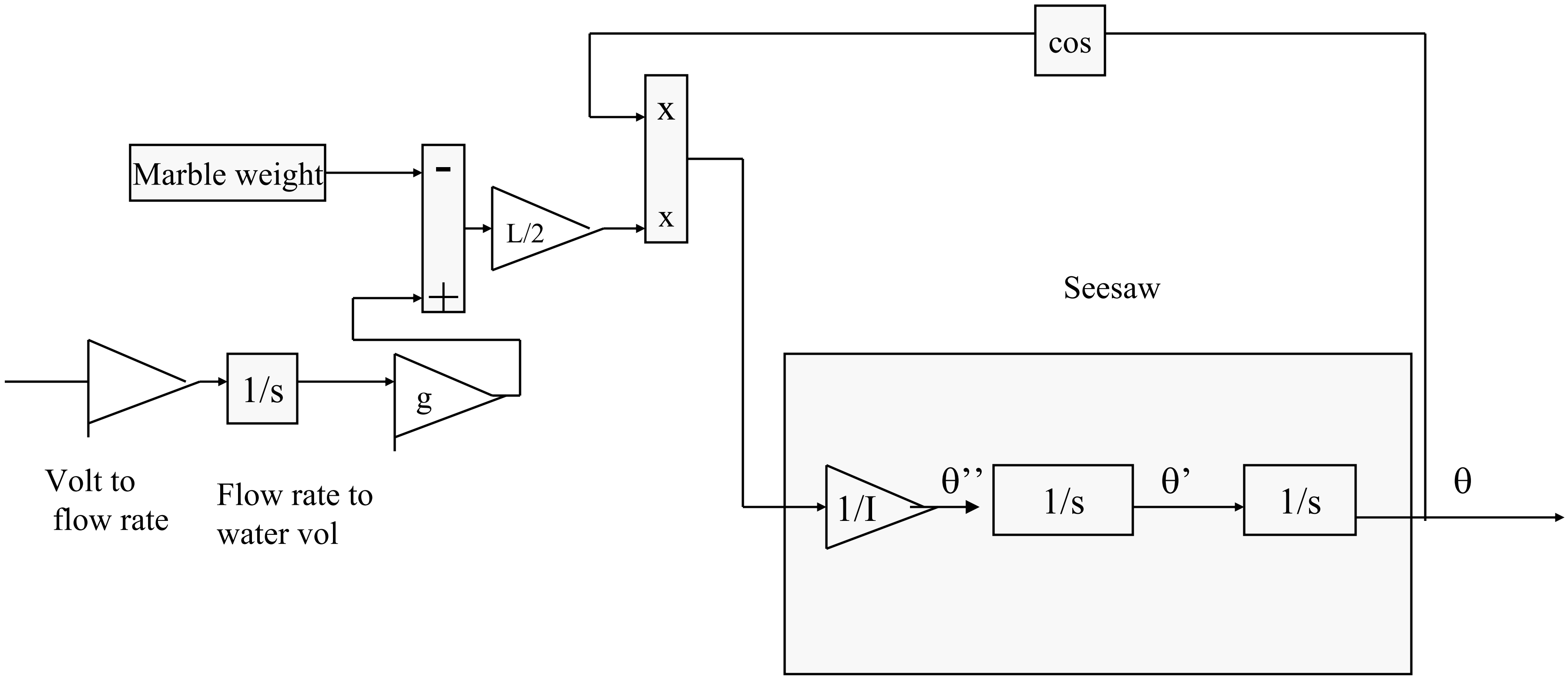
Modeling The Water Seesaw

It is self-evident that the the water see-saw is a highly unstable device. It is a triple integrator

- control output -> flow rate
- flow rate integration -> water in the container
- water weight - marble weight -> torque -> angular acceleration
- angular acceleration integration -> angular velocity
- angular velocity integration -> angular position



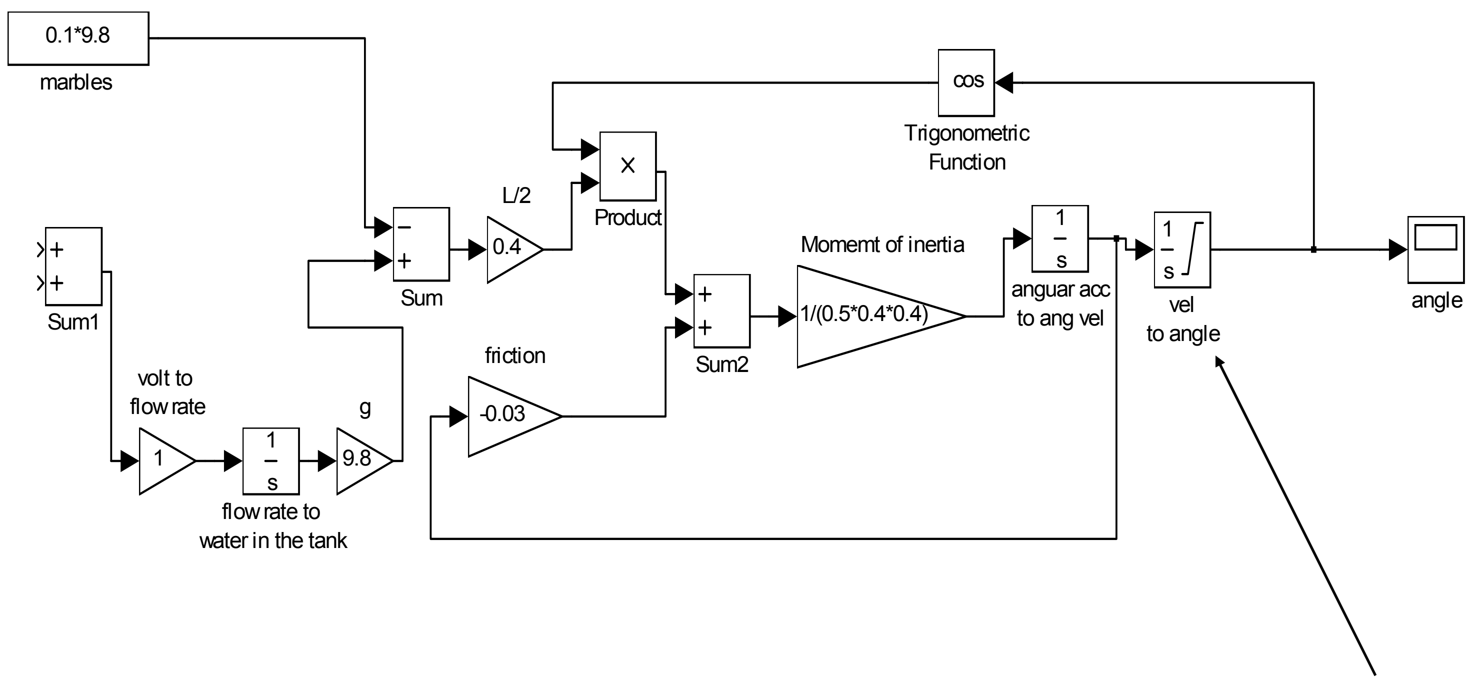
Water Seesaw Simulation Diagram



[A, B, C, D] = linmod('filename') will give you the *open loop* linear model, eig(A) will tell you that it is unstable.

Matlab Simulink Model

Where should we add the controls in this diagram?



Note that the angle has a limit due to the table that restricts the beam's movement

Summary

- We have
  - reviewed the basic concepts of control
  - examined P+D control
  - introduced the simulation approach
  - some basics of MATLAB
  - How control affects a system’s behavior

- Next Lecture
  - Serial I/O