Overview

- Last lecture: **basics of control theory, stability, proportional control and transfer function.**

- **Today:**
  - Proportional + Derivative + Integral Control
  - Matlab and Simulink

- **A basic tutorial** on control theory is available on the web: see https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID on PID controllers.

Integral Control

- A system often has friction or changing workloads that we may not be able to model in advance:
  - In auto-cruise control, we cannot know how many passengers will be in the car
  - Frictions may change due to machine conditions
- Unmodeled heavy load often results in steady-state error, the system will settle near rather than at the set-point.

What Does Integral Control Do?

- Integral control adds up (integrates) the past errors and then gives a force that is proportional to the cumulative errors.
  - So if the marble gets stuck near a set-point due to some friction, the position error adds-up over time, eventually generate a force large enough to help get the marble going toward the set-point.
  - So if the car has a heavier load and the velocity settles on a speed lower than the set-point for a while, the error adds up and the integral control leads to increased throttle.

The Dark Side of Integral Control

- Integral control acts on cumulative errors. It takes a while to reach a large sum and it will take time to reduce the sum. Consider the following case:
  - the marble is stuck on the left side of the set-point
  - After 10 sec, the integral control is large enough to help get the marble moving toward the set-point
  - The integral will keep increasing until the marble crosses the origin
  - It will take a while to “wash out” the cumulative error.
- <u>Overdose of integral control is a common source of overshoot, oscillations and even instability</u>.

Using Integral Control

- As a rule of thumb, start from zero integral control and use it lightly!
- Check the eigenvalues of the system to make sure that all of them are sufficiently negative.

- $X'' = F/m$, where $F = -K_p x - K_d x' - K_i \int^* x\, dt$

- $s^2 + (K_d / m) s + K_p /m + (K_i /m) 1/s = 0$, ➔ Laplace transform of an integral is 1/s

- $s^3 + (K_d /m) s^2 + (K_p /m) s + K_i /m = 0$

- As we can see, the effect of integral is to add an order to the system. Large value of K_i could lead to positive eigenvalue.

- We can solve the equation to see if the real part of the eigenvalues are still negative. This is best done by using tools such as Matlab. A subject that we will overview in next lecture.

How to Tune a Simple PID Controller

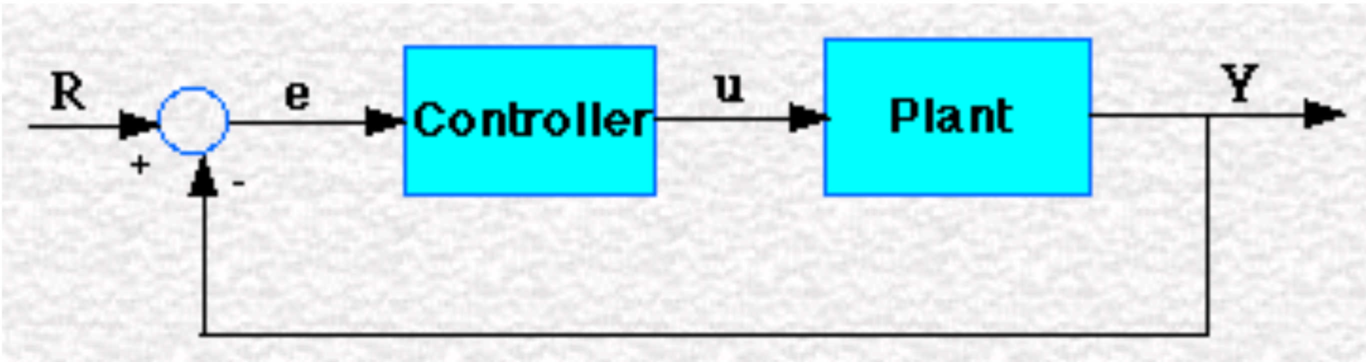
- Experimental tuning procedure
  - first set derivative gain and integral gain to zero.
  - Slowly increase the proportional gain until the device oscillates around the set point with a speed that is satisfactory. At this point,
    - the real part of the eigenvalue is _________
    - the imaginary part of the eigenvalue determines the __________ of oscillation
  - Slowly increase the derivative gain until the device settles down at the setpoint. At this point, the real part of the eigenvalue is ____________
  - If there is steady-state error, slightly increase the integral gain until the steady state error is corrected and yet not causing serious oscillations. This means that
    - the real part of the eigenvalues is still _________.

- Fine tuning
  - If the motion movement towards setpoint is too slow, we can __________ the proportional gain or ___________ derivative gain, don't play with integral gain!
  - If there is steady-state error, we can add a little of __________ gain
  - If the motion overshoots the setpoint and oscillates, we can __________ the derivative gain or reduce __________ gain and __________ gain

Transfer function of PID controller

- Quiz: what is the transfer function of PID controller?

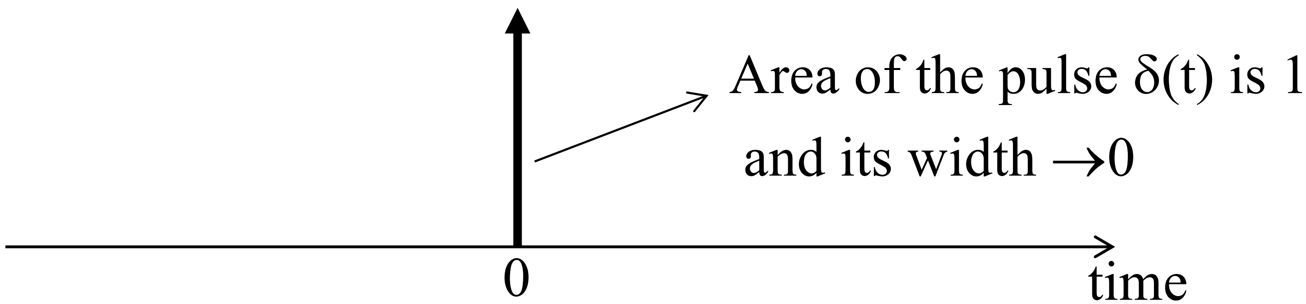
$$u(t) = K_p e(t) + K_d \frac{d}{dt} e(t) + K_i \int_0^t e(\tau) d\tau$$



Basic input signals

- **<u>Delta function:</u>**

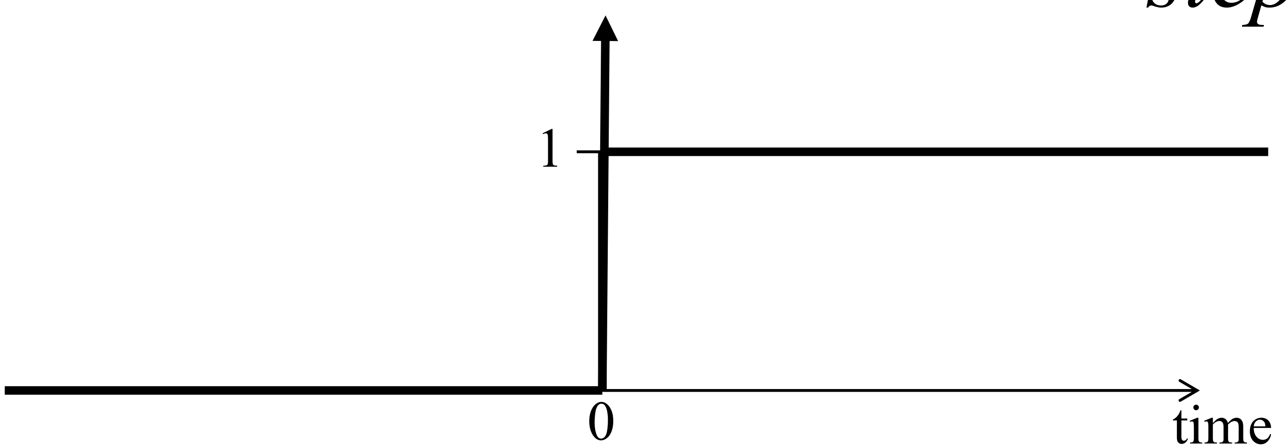
$$\delta(t) \quad \Leftrightarrow \quad 1$$



Laplace transform

- **<u>Step function:</u>**

$$step(t) \quad \Leftrightarrow \quad \frac{1}{s}$$

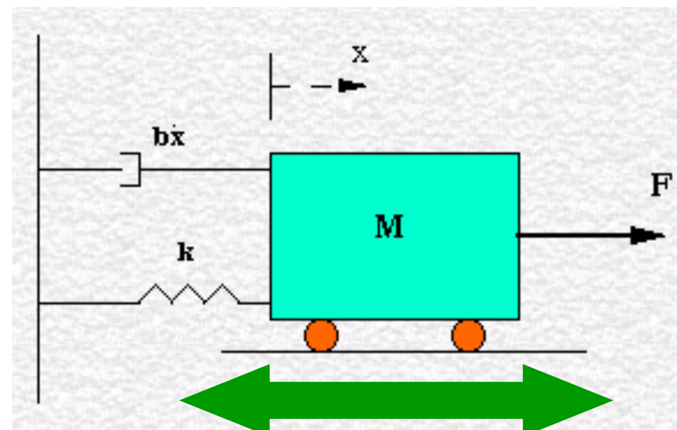


Class exercise

- Consider a physical system which has mass, spring, and damper

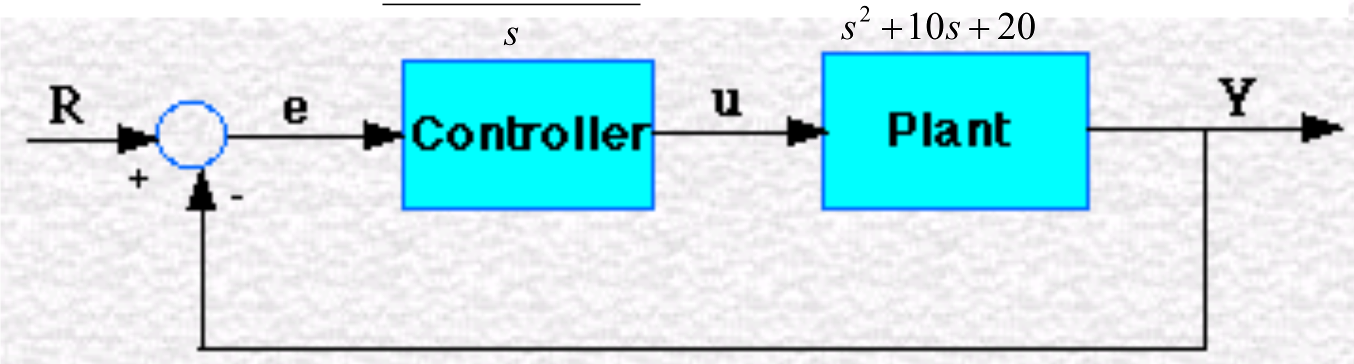
- We want to apply a position control; that is, r(t) will describe the trajectory to be followed by the mass.

- What is the transfer function of the closed-loop system?



Position control

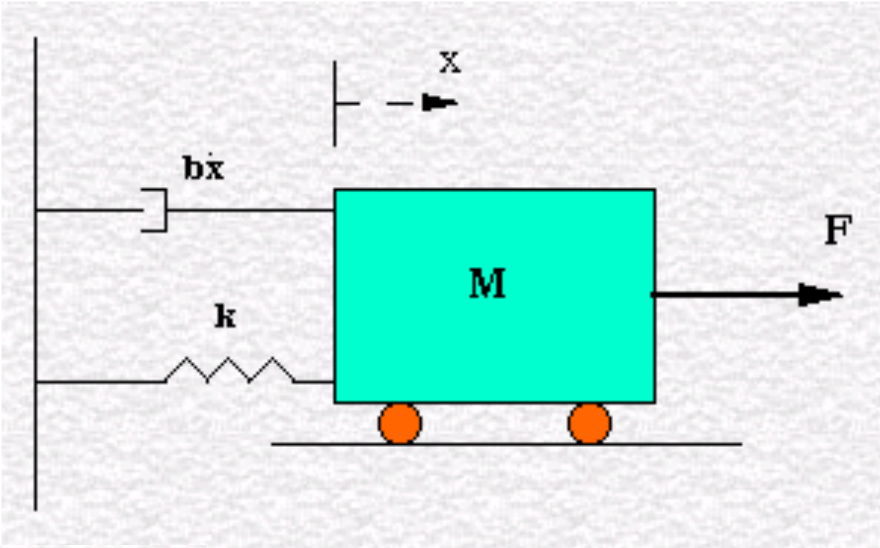
$$\frac{K_d s^2 + K_p s + K_i}{s} \qquad \frac{1}{s^2 + 10s + 20}$$



Example of mass, spring and damper problem

- Suppose to have a simple mass, spring, and damper problem.

Assume that motion's equation is given!



- $M = 1\text{kg}$
- $b = 10\ \text{N.s/m}$
- $k = 20\ \text{N/m}$

$$M\ddot{x} + b\dot{x} + kx = F$$

Laplace transform

$$Ms^2X(s) + bsX(s) + kX(s) = F(s)$$

$$\frac{X(s)}{F(s)} = \frac{1}{Ms^2 + bs + k}$$

Transfer Function

$$\frac{X(s)}{F(s)} = \frac{1}{s^2 + 10s + 20}$$

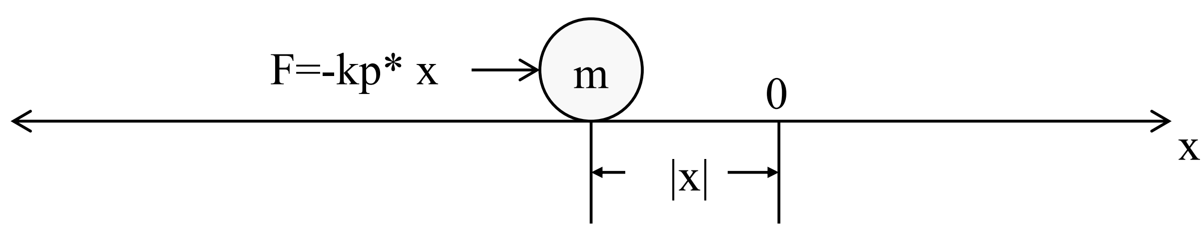
•More details are available at https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID

Tool: Matlab and Simulink

- **A basic tutorial** on simulink is available as additional material: see **simulink_quickstart.pdf**

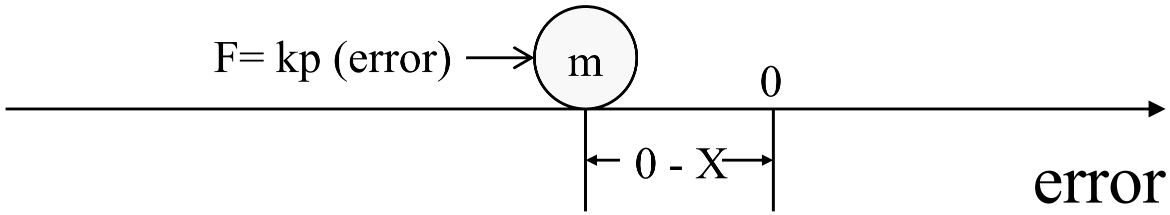
Simulate A Simple System with Proportional Control

- 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 *<u>position error</u>* is negative, it pushes with a positive force and vice versa.
    - K_p is a positive integer known as proportional control constant.

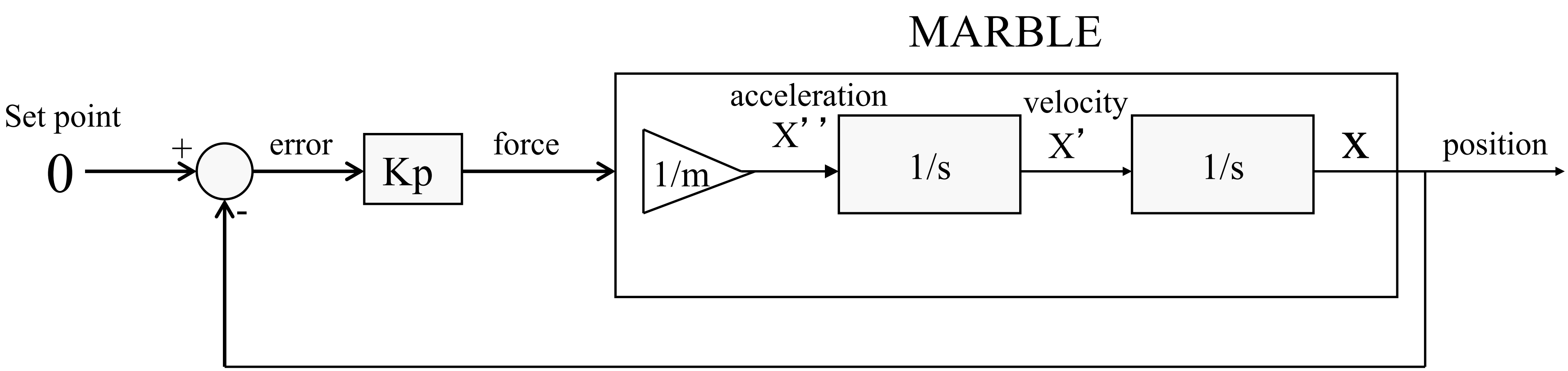


Basic Simulation Diagram Concepts: Marble Under Proportional Control

Equal to zero!



error = Set_point - X

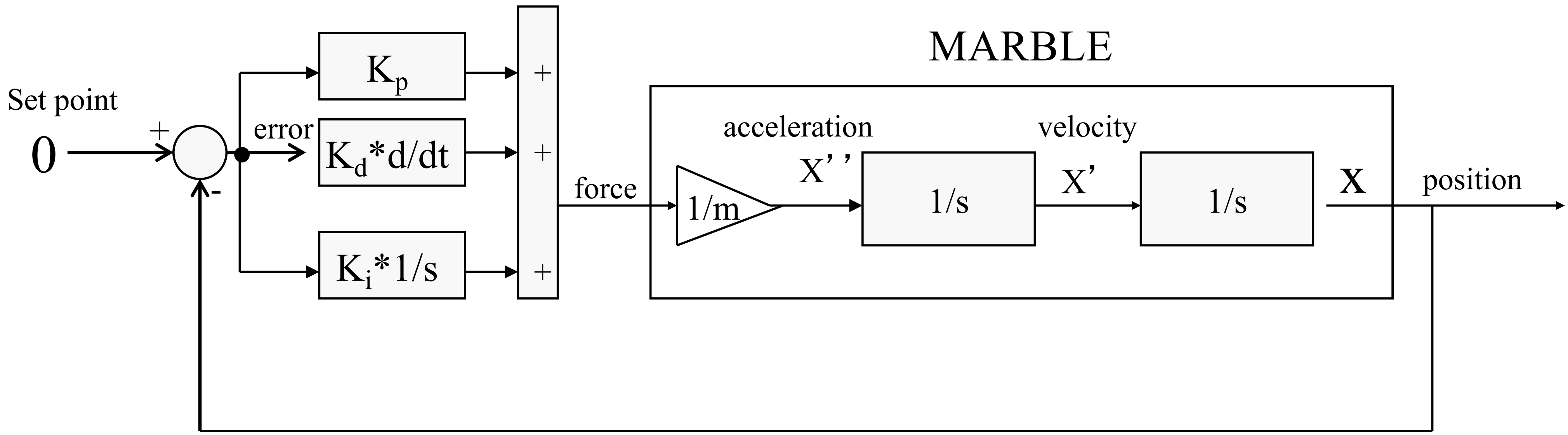


The double integrator with weight 1/m transforms force to acceleration, acceleration to velocity, and then velocity to position

1/s denotes integration (see Laplace transform of integral!)

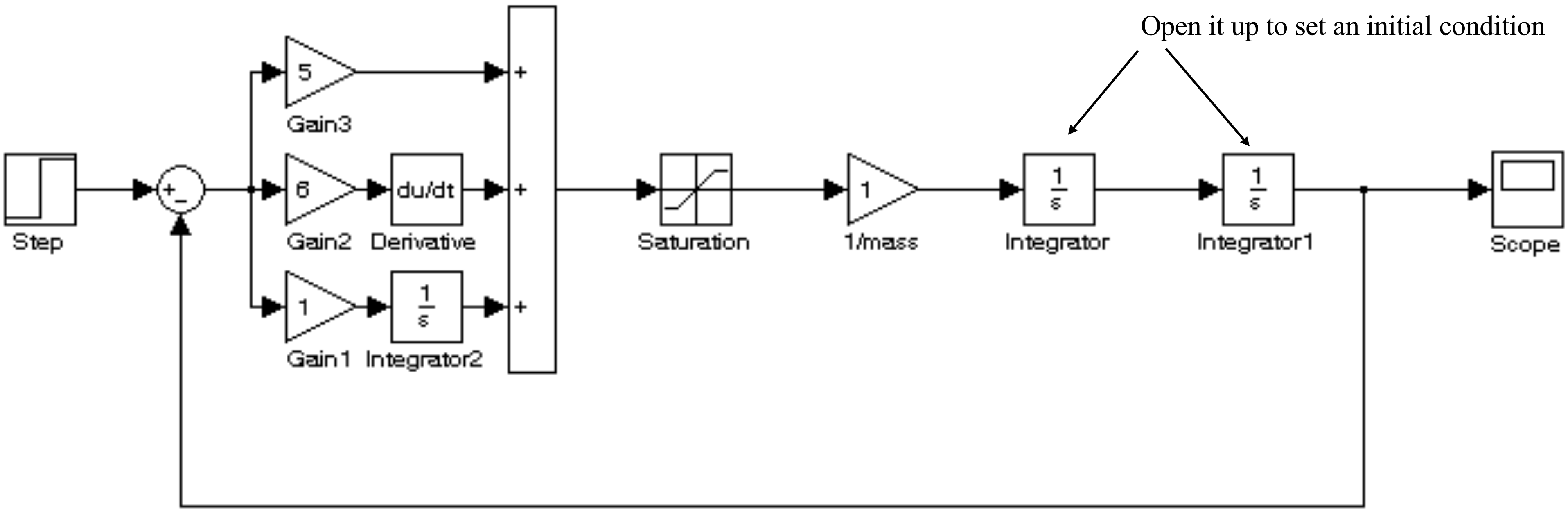
Block diagram of Marble Control

Block diagram of Marble control: you can plot it with simulink and run simulations of the system!



•Quiz: What type of controller is this?

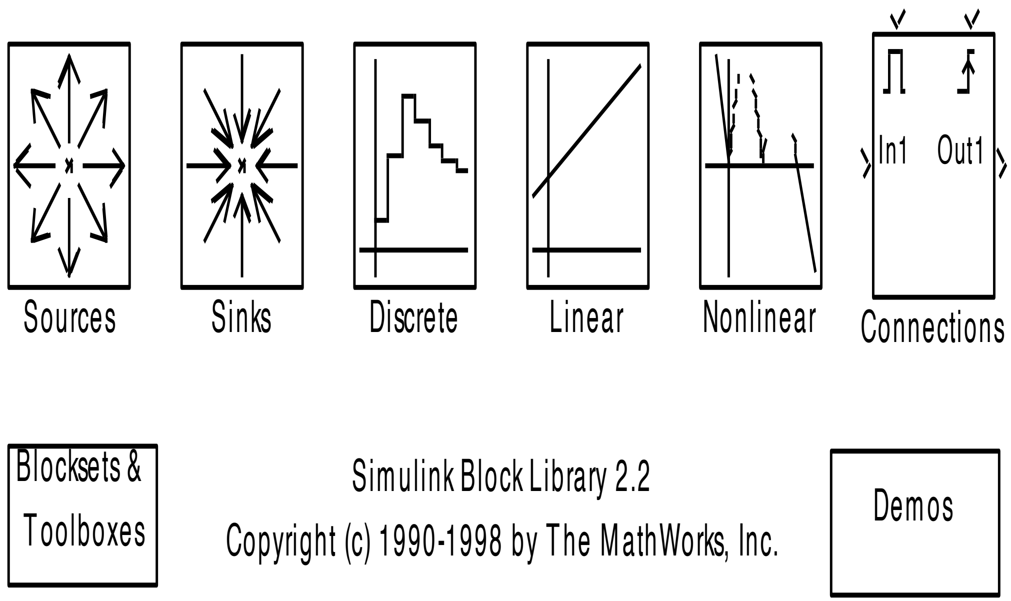
Simulink Simulation Diagram

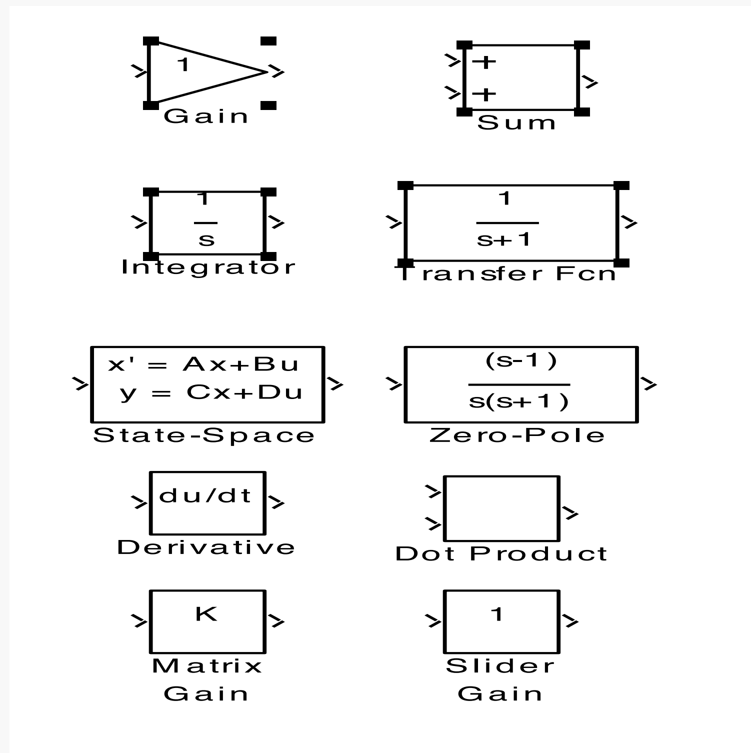


•Quiz: What are Gain1 Gain2 and Gain3 called?

Matlab Simulink

- >>simulink



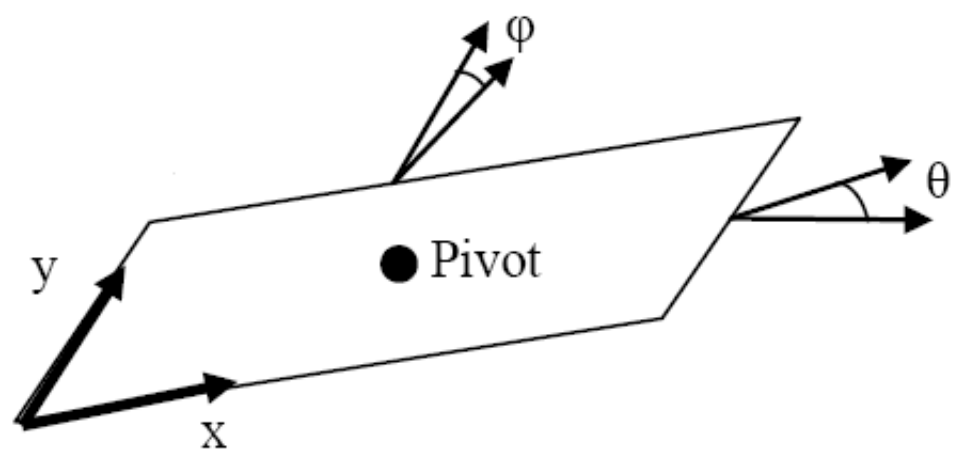


Linear box

- Scope is in sink box
- saturation is in non-linear box etc
- double click to open a box and to set parameters
- **See Simulink Quick Start guide** (simulink_quickstart.pdf)

Modeling the amazing ball system (simplified)

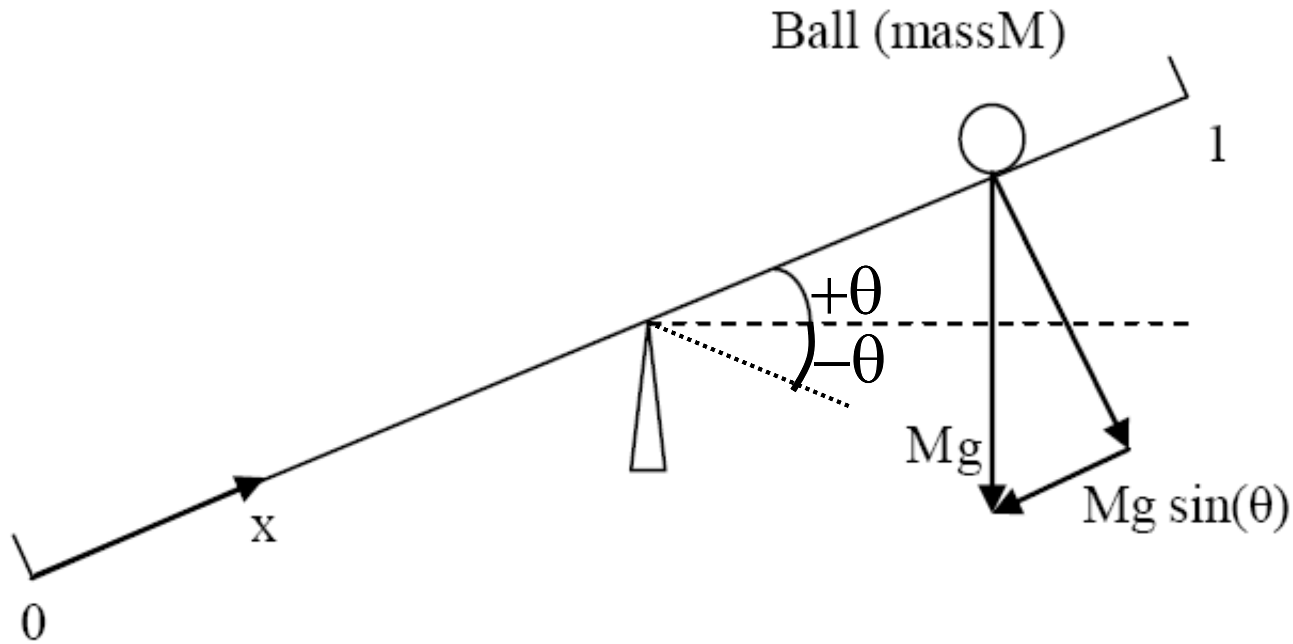
How can we model the amazing ball system?
- Neglect friction and rolling effects ➔ model amazing ball as inclined plane



Modeling the amazing ball system (simplified)

It is easy to model the amazing ball system as inclined plane by using Newton's second law F=m*a

- control variable = angle of servo
- change of servo's angle ➔ affects the (parallel) component of weight force applied to ball
- M*g*sin(θ) ➔ parallel component of weight force applied to the ball ➔ acceleration x''
- Integration of acceleration ➔ velocity x'
- Integration of velocity ➔ position x



1-Dimensional Free Body Diagram

Matlab Simulink Model: amazing ball system

Class exercise: Where should we add the PID controller in this diagram?