

Feedback Basics

David M. Auslander
Mechanical Engineering
University of California at Berkeley

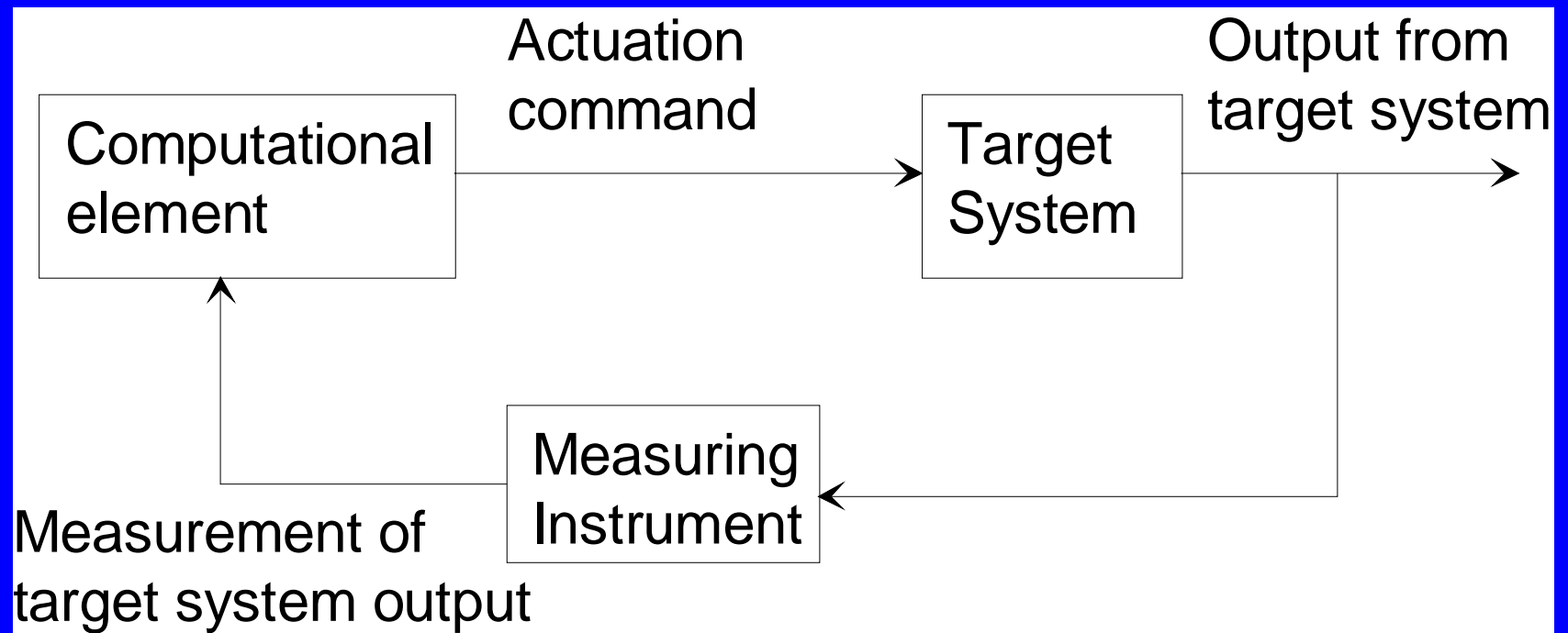
copyright © 1998, D.M. Auslander

I. Feedback Control Context

What is Feedback Control?

- ❖ Measure desired behavior of target system
- ❖ Compare measured to desired behavior
- ❖ Based on comparison, decide on appropriate actuation so measured will move closer to desired

Feedback Control System



Why Use It?

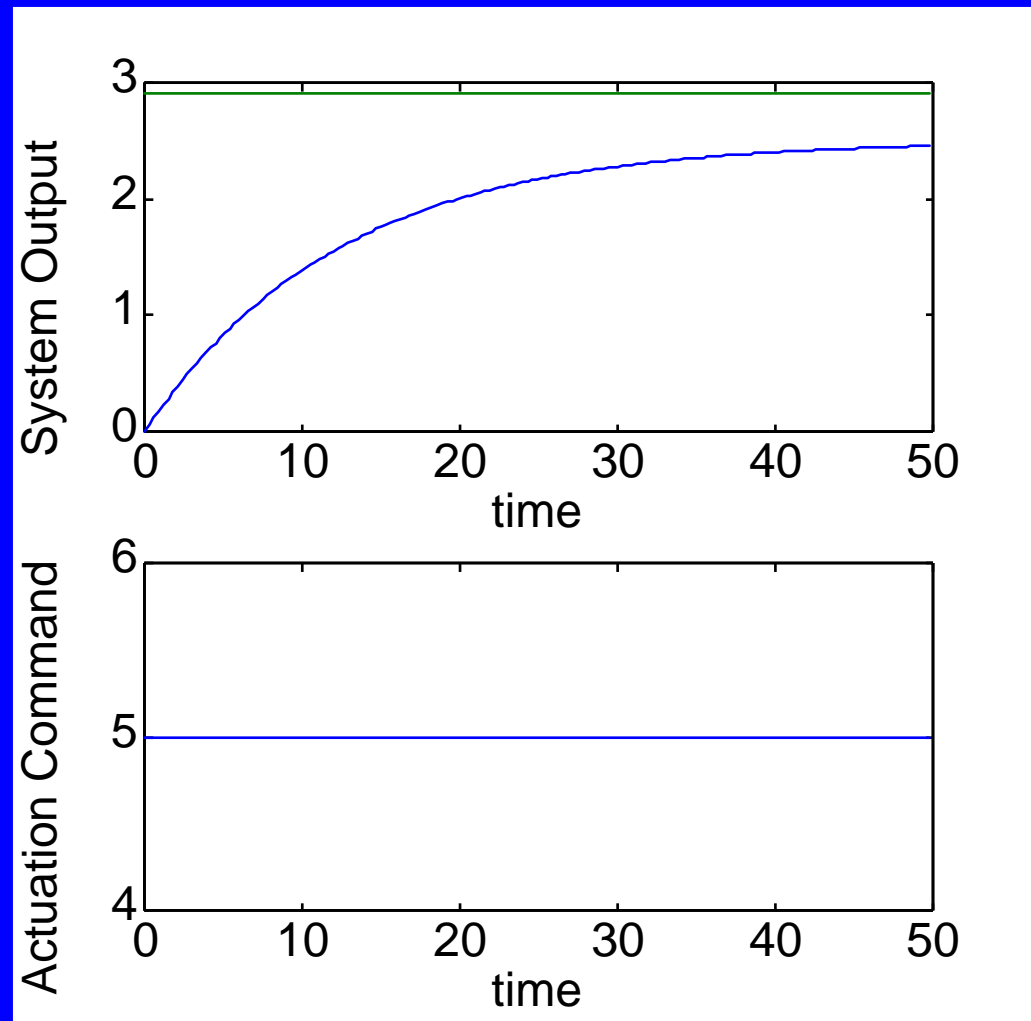
- ❖ To compensate for lack of knowledge
- ❖ With “perfect” knowledge the target system can be made to do anything desired
- ❖ No need to measure - behavior is “known,” i.e. predictable from our knowledge
- ❖ The real world is not so nice! Unknown disturbances and imperfect system models are more the norm

Demonstrations

- ❖ Matlab animations - tank(s) and mass(es)
- ❖ Single tank with drain -
 - IntroTank1.m and IntroTank2.m
 - No feedback (open loop) and with feedback
- ❖ Single mass
 - IntroMass1.m and IntroMass2.m
 - No feedback, with feedback

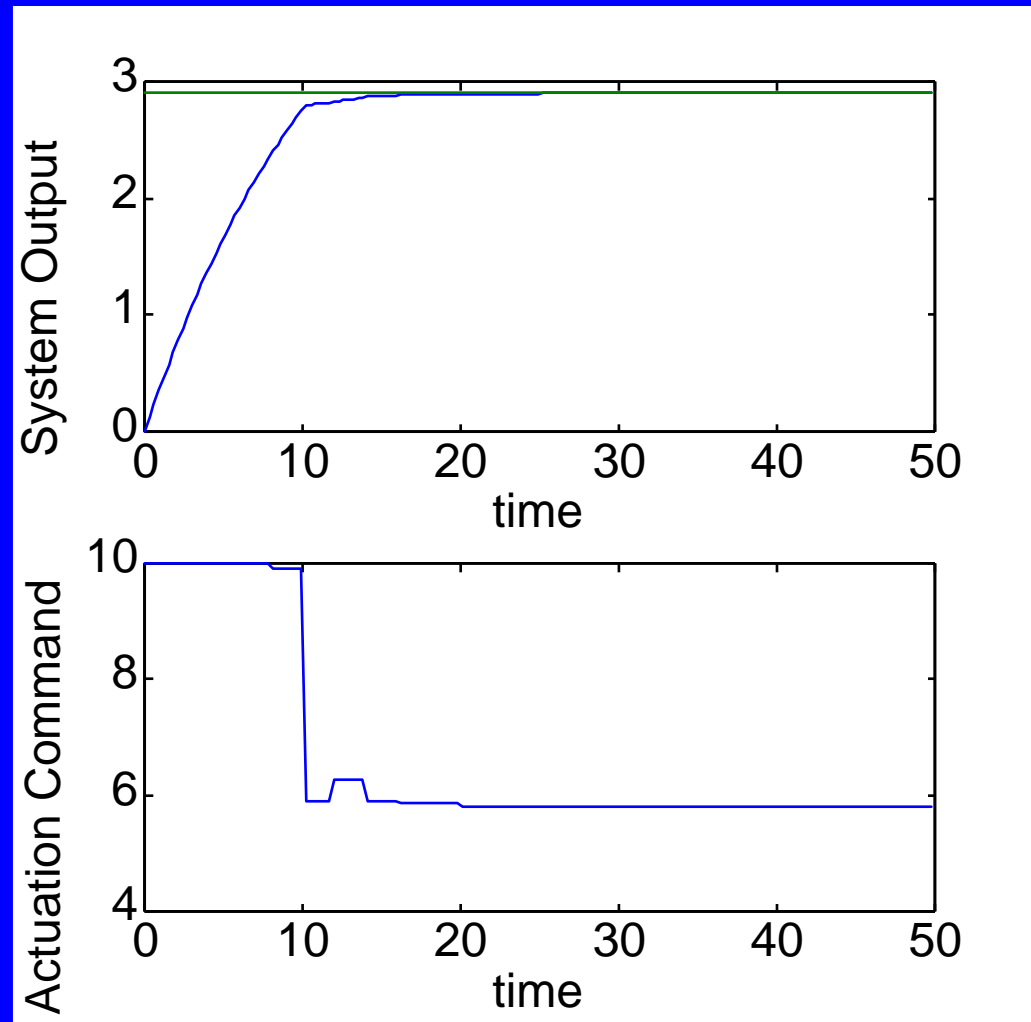
Single Tank, No Feedback

- ❖ Top line (2.9) is desired value
- ❖ System output (liquid height) never quite gets there
- ❖ Actuation constant at 5



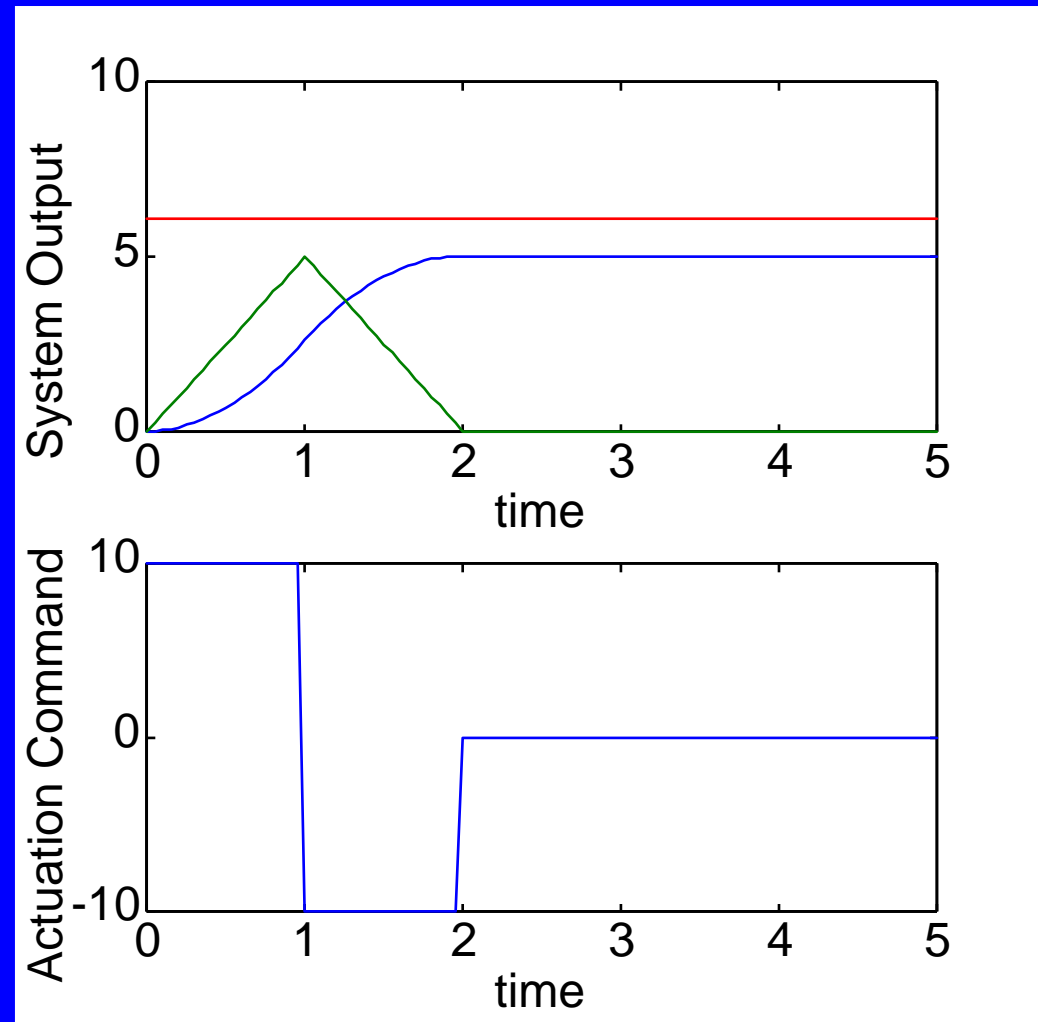
Single Tanks, With Feedback

- ❖ Liquid level reaches desired value
- ❖ Gets there faster
- ❖ Actuation is adjusted based on feedback measurement



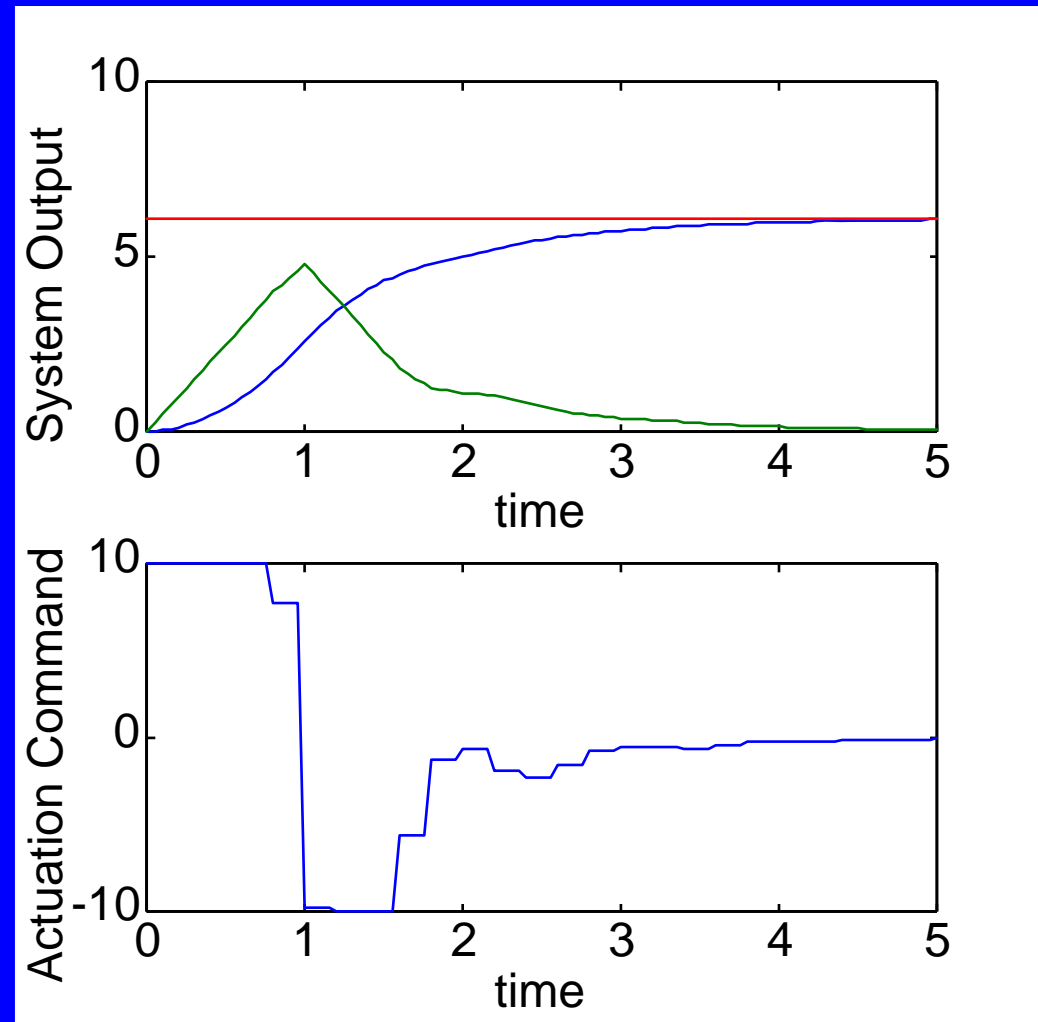
Position a Mass - No Feedback

- ❖ Turn actuation on full blast to accelerate
- ❖ Turn it full negative to decelerate
- ❖ Doesn't quite reach desired position



Position a Mass - With Feedback

- ❖ Gets to desired position
- ❖ Similar actuation pattern
- ❖ Final stage is much more conservative
- ❖ Overall response is slower

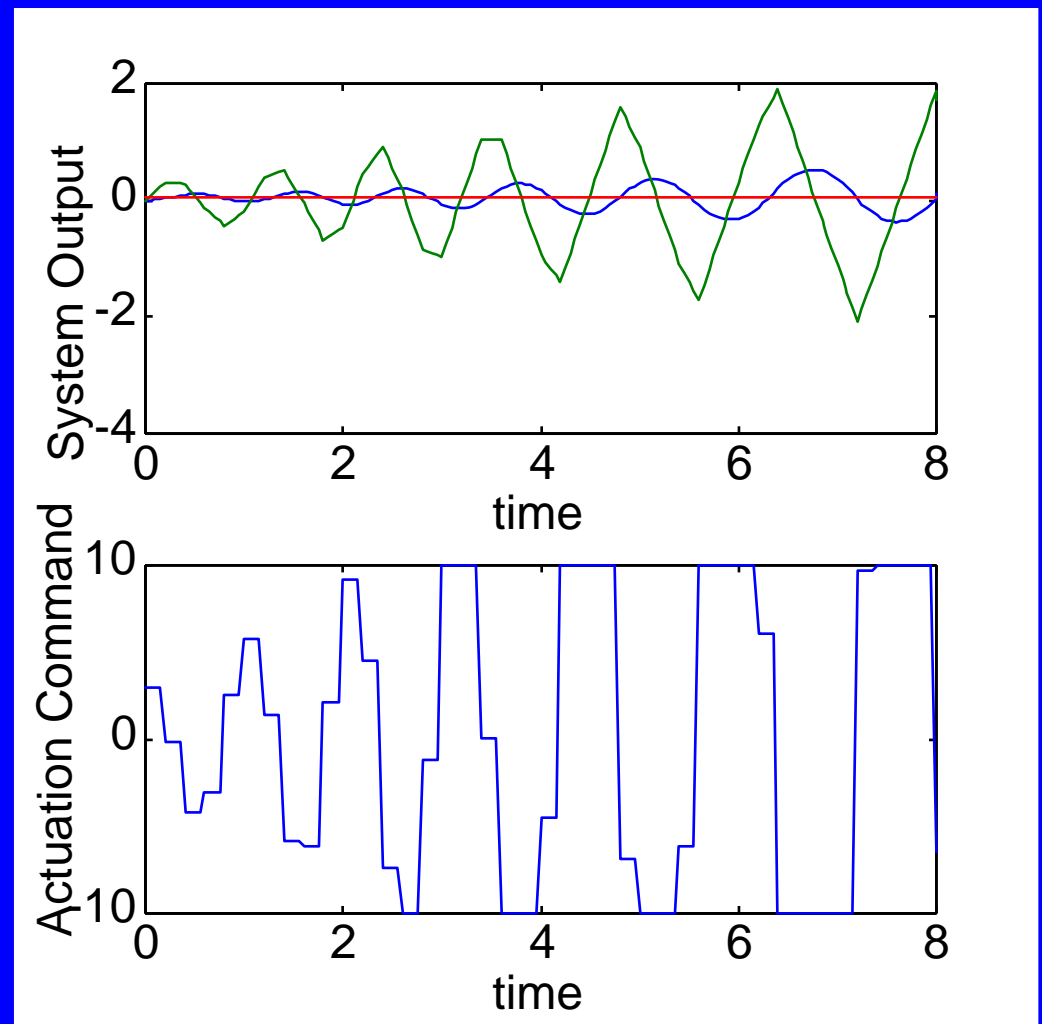


Feedback Control Risks

- ❖ Improper use of feedback can make behavior worse than non-feedback case
- ❖ In these cases, open-loop system is stable
- ❖ Feedback can make it unstable
- ❖ IntroMassUnstable.m

Controller Induced Instability

- ❖ Motion diverges from the desired position (0.05, red)
- ❖ Oscillation levels off when actuator reaches its limit (at ± 10)



Difficulties Making Feedback Work

- ❖ Dynamics
- ❖ Nonlinearity
- ❖ Noise
- ❖ Dimensionality

Dynamics

Definition

- ❖ History dependence
- ❖ Output depends on the history of the input
- ❖ Example, tank
 - Instantaneous value of level depends on recent inflow history
 - Knowledge of current inflow does not say anything about level
 - *Vice-versa*, knowledge of the level does not say anything about current inflow

Energy

- ❖ For physical systems, history is characterized by storage of energy
- ❖ Each independent energy storage is described by a "state variable"
- ❖ Examples
 - Liquid height in tank is potential energy
 - Velocity of mass is kinetic energy
- ❖ History-dependent behavior occurs as stored energy is released

Ordinary Differential Equations

- ❖ Calculus is a mathematics that can be used to describe history-dependent behavior
- ❖ Systems with a finite number of state variables can be described by ordinary differential equations, ODEs (as contrasted with partial differential equations)
- ❖ Another contrast - "finite-state-machines" have state variables that each can only have a finite number of values (usually binary, sequential logic)
- ❖ Energy-based state variables are "real" numbers - infinite precision

Single-Port Elements

- ❖ Each element has a single energy storage mode
- ❖ Characterized by one state variable ("lumped" element)
- ❖ Its connection to other elements (port) is characterized by a pair of "signal" variables
 - Voltage, current for electrical
 - Force, velocity or torque, angular velocity for mechanical
- ❖ Product of signal variables is power ($v * i$)

Accumulation Variables

- ❖ Integration of one of the signal variables gives an accumulation related to energy
- ❖ Example, $q = \int i dt$
- ❖ q , charge, is related to potential energy in a capacitor
- ❖ These are good choices for state variables
- ❖ Note: choice of state variables is not unique

Constitutive Relations

- ❖ Other signal variable is related algebraically to accumulation variable
- ❖ To continue capacitor example,
 - Voltage derived from charge, $v = q / C$
 - More common form, $i = C (dv/dt)$
- ❖ Similar relation for inductor

Dissipative Elements

- ❖ Resistive elements
- ❖ Store no energy
- ❖ Convert mechanical power to heat
- ❖ No accumulation variables
- ❖ Signal variable pair is algebraically connected
 - For electrical resistor, $v = i R$

Modeling

- ❖ Single-port element is an idealization of real physical components
- ❖ Makes derivation of mathematical model easy
- ❖ Works well for many real engineering problems (but certainly not for all!)
- ❖ Each energy storage element has one state
- ❖ Dissipative elements have no states

System Topology

- ❖ Differential equation can be derived by combining single-port relations with interconnect information
- ❖ Physical systems have continuity and conservation laws to describe interconnect
- ❖ Kirchoff loop and node rules for electrical systems, for example

Getting the ODE

- ❖ Get a set of coupled, 1st order ordinary differential equations from:
 - accumulation equations
 - constitutive relations
 - continuity and conservation
- ❖ One equation for each energy storage element

Simulation

- ❖ Numerical solution of resulting equations
- ❖ Provides a prediction of system behavior
- ❖ Use for “what-if” studies
- ❖ Iterate for design studies

Euler's method

- ❖ What does it mean to do "numerical solution"?
- ❖ Euler's method is simplest, illustrates principles
- ❖ First-order example:

$dx/dt = f(x) + u(t)$ -- $f()$ and $u()$ are known functions

Approximate dx/dt with $\Delta x/\Delta t$; $\Delta x = [f(x) + u(t)] \Delta t$

For any value of x and t this gives change in x (Δx)

Then, assume $f(x)$ and $u(t)$ don't change for some short time

So, x at the next time is approximated by,

$$x(t + \Delta t) = x(t) + [f(x(t)) + u(t)] \Delta t$$

Applying Euler's Method

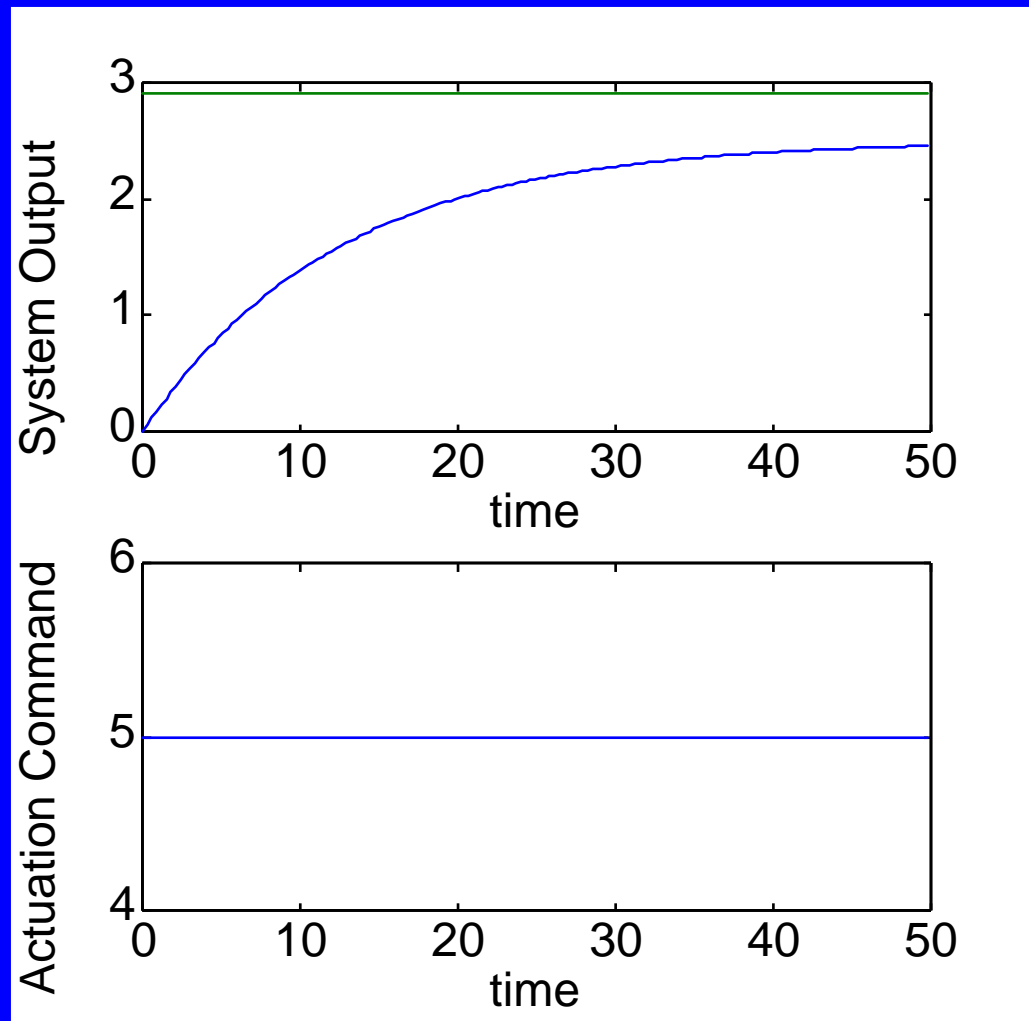
- ❖ Use this approximation over and over again to step out solution
- ❖ If higher than first order, use it on all equations at the same time (compute all stuff on the right, then update all state variables)
- ❖ Each variable on left (x in dx/dt , y in dy/dt , etc.) is a state variable
- ❖ Accuracy depends on making Δt small enough
- ❖ Euler's method is very crude, requires relatively small step size

Demo - Step Size

- ❖ IntroTank1.m -- tank, small step size (repeat)
- ❖ DynEulerStep2.m -- tank, large step size

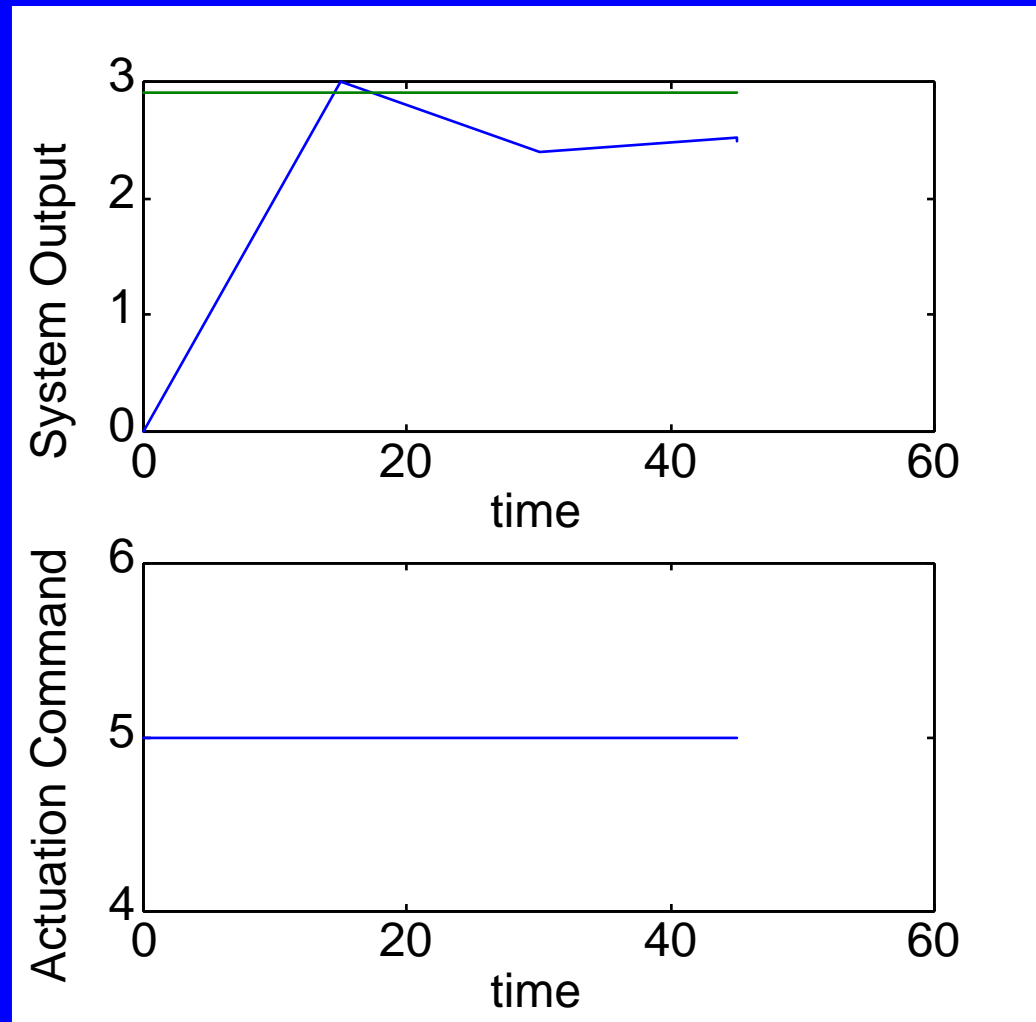
Tank, Small Step Size

- ❖ Repeat of earlier result
- ❖ Smaller step size would not change shape significantly



Tank, Large Step Size

- ❖ Note that character of response has changed
- ❖ This is an artifact
- ❖ Caused by step size too large
- ❖ Fast animation speed!



High-Order Solvers

- ❖ Better, more complex approximations than Euler's can be constructed
- ❖ More efficient for many problems
- ❖ Best known: Runge-Kutta methods

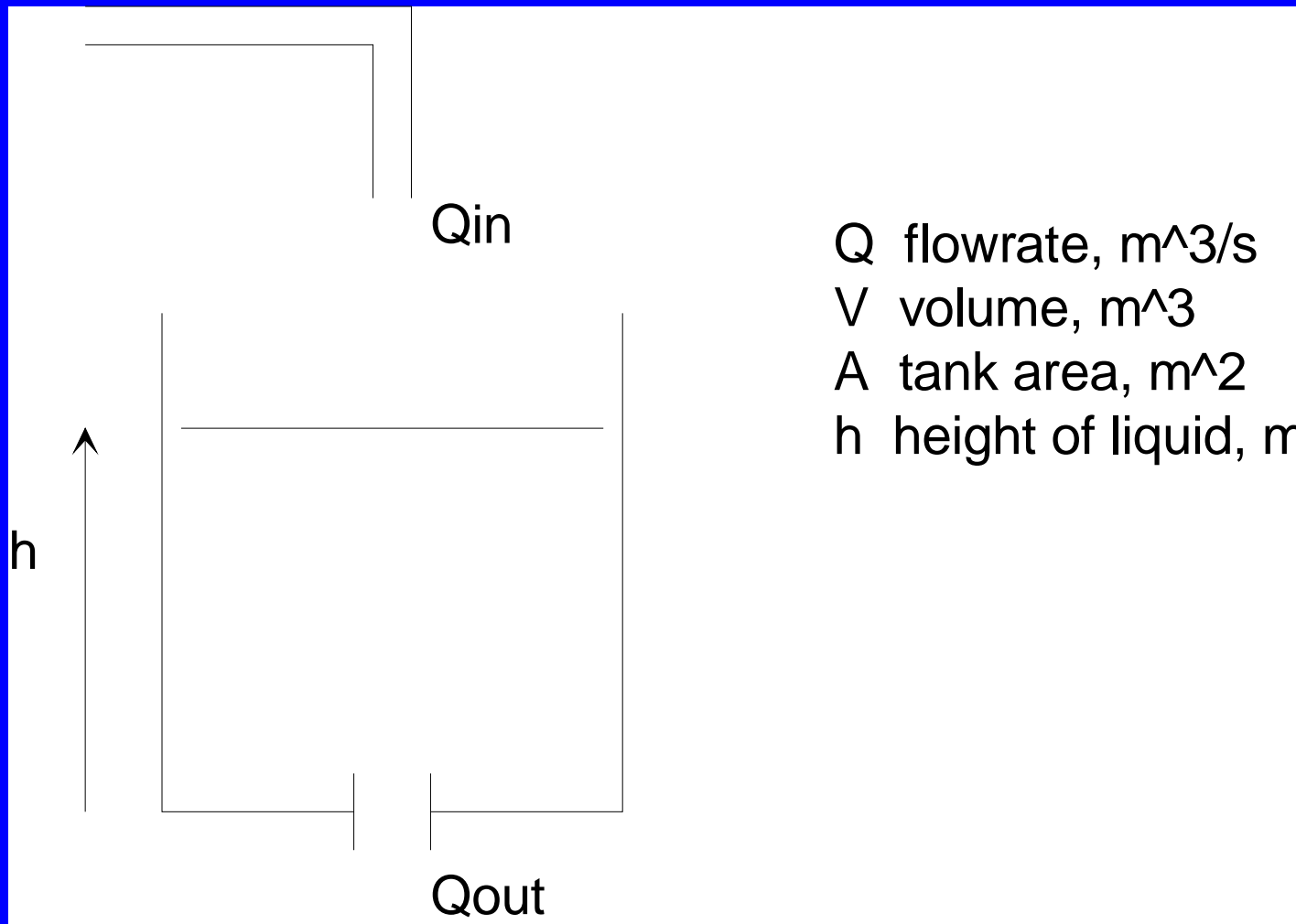
Stiff Problems

- ❖ Systems with widely separated time scales cause severe numerical difficulties
- ❖ "Stiff" systems
- ❖ Standard methods cannot look at long time scale without dealing with the details of the short time scale
- ❖ "Implicit" methods can deal with long term while ignoring details of short term
- ❖ Heavy calculation load per step, but, for stiff systems, very much bigger steps than standard (explicit) methods

Why Simulate?

- ❖ Discipline of modeling
 - Forces rigorous examination of target system
 - Unearths easily overlooked characteristics
- ❖ Complex interactions can produce unanticipated system behaviors
- ❖ Multi-parameter optimization

Example - Tank



Q flowrate, m^3/s
 V volume, m^3
 A tank area, m^2
 h height of liquid, m

Tank Equations

$$\frac{dV}{dt} = Q_{in} - Q_{out}$$

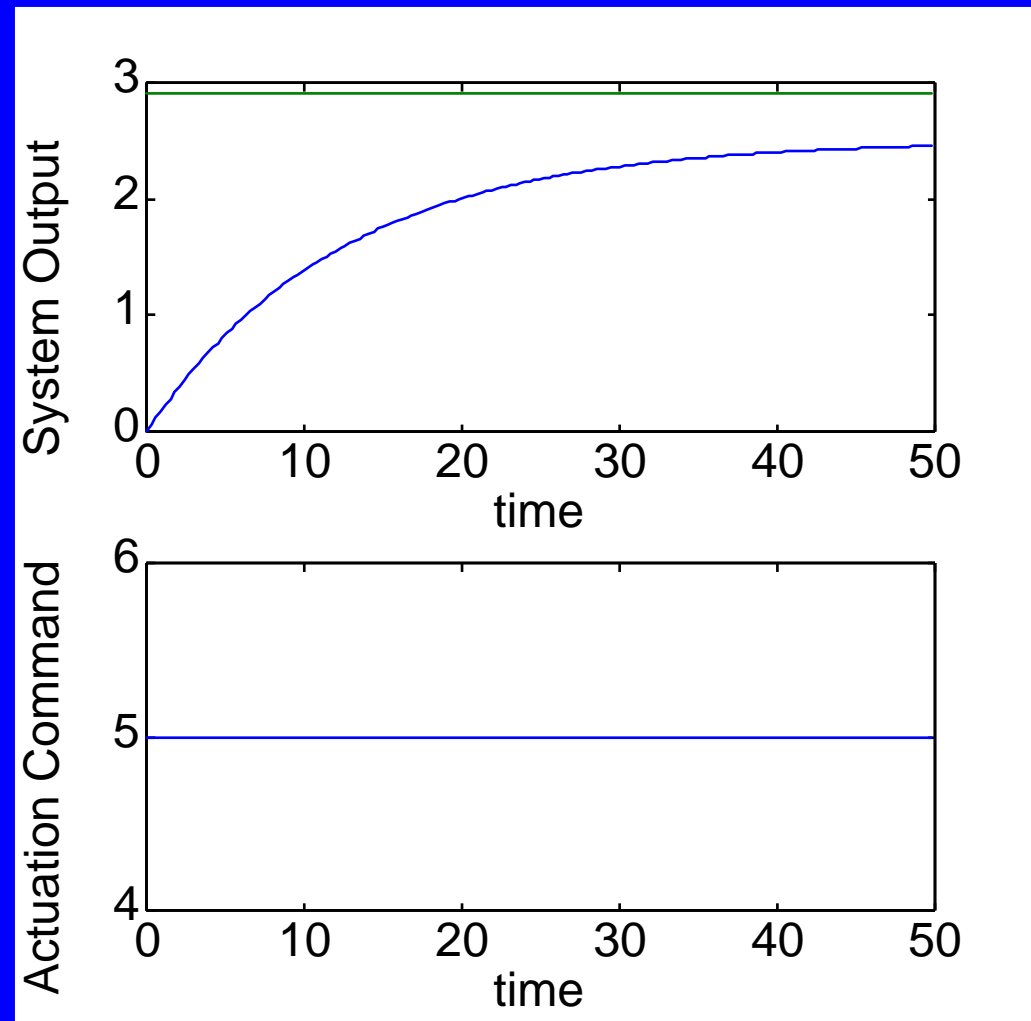
$$Q_{out} = kh \quad (\textit{not very accurate!})$$

$$V = Ah$$

$$\frac{dh}{dt} = \frac{(Q_{in} - kh)}{A}$$

Tank Simulation

- ❖ IntroTank1.m (repeat)
- ❖ Each step computes inflow, outflow
- ❖ Then computes change in height



Dynamics

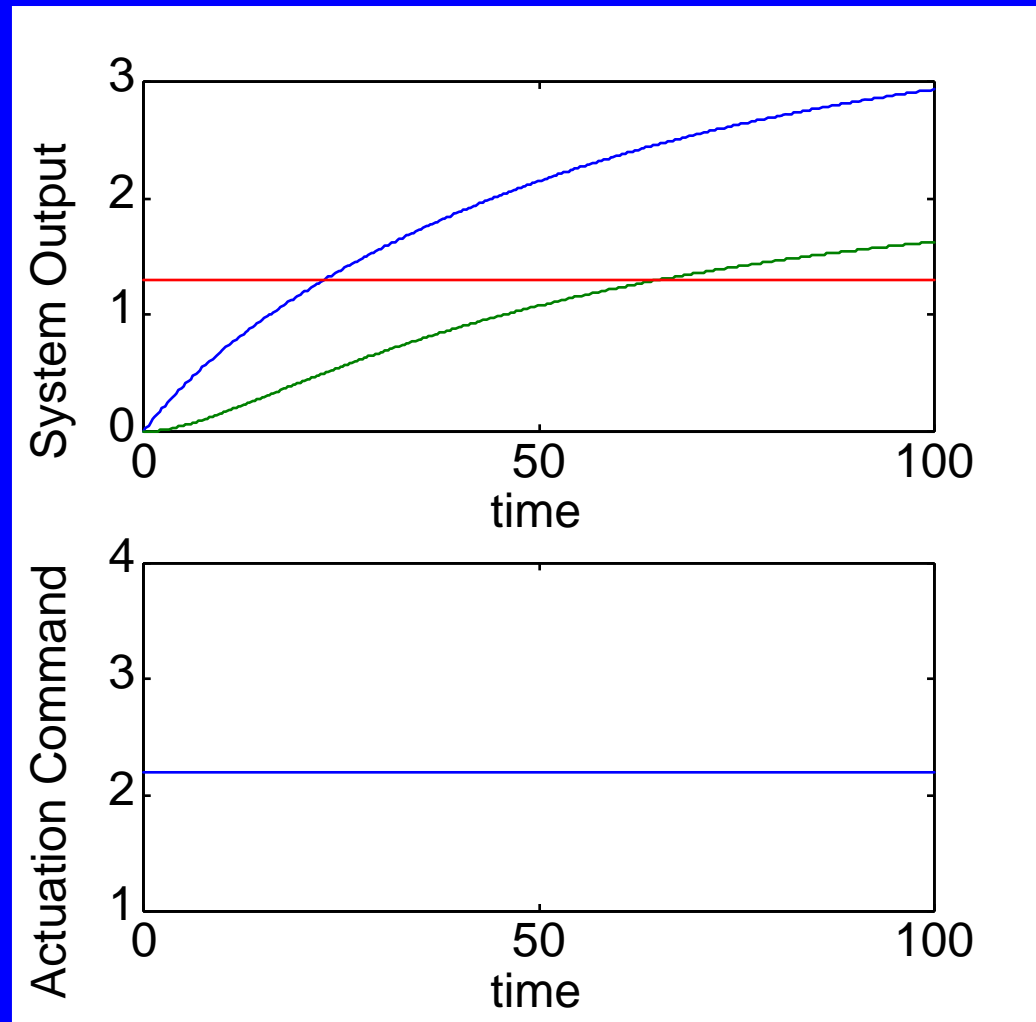
- ❖ Information hiding
- ❖ Internal energy storages, not measured
- ❖ Dynamic information not visible to the system measurement
- ❖ Example - two interconnected tanks
- ❖ Liquid height only measured in one
- ❖ Much more difficult to control

Two-Tank Demo

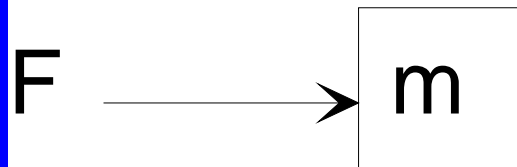
- ❖ Contrast single and double tank systems:
- ❖ IntroTank1.m (back a couple of slides)
- ❖ Two tanks connected with a pipe
- ❖ Inflow to first tank; outflow from second
- ❖ Dyn2Tank.m

Two Tanks, Open Loop

- ❖ Change in first tank (blue) shows up right away
- ❖ Second tank (green) "hides" change in inflow initially



Example - Mass



x, v →

F force, N

x position, m

v velocity, m/s

m mass, kg

H momentum, kg N

Mass, Equations

$$\frac{dH}{dt} = F; \quad H = mv$$

$$\frac{dx}{dt} = v$$

thus,

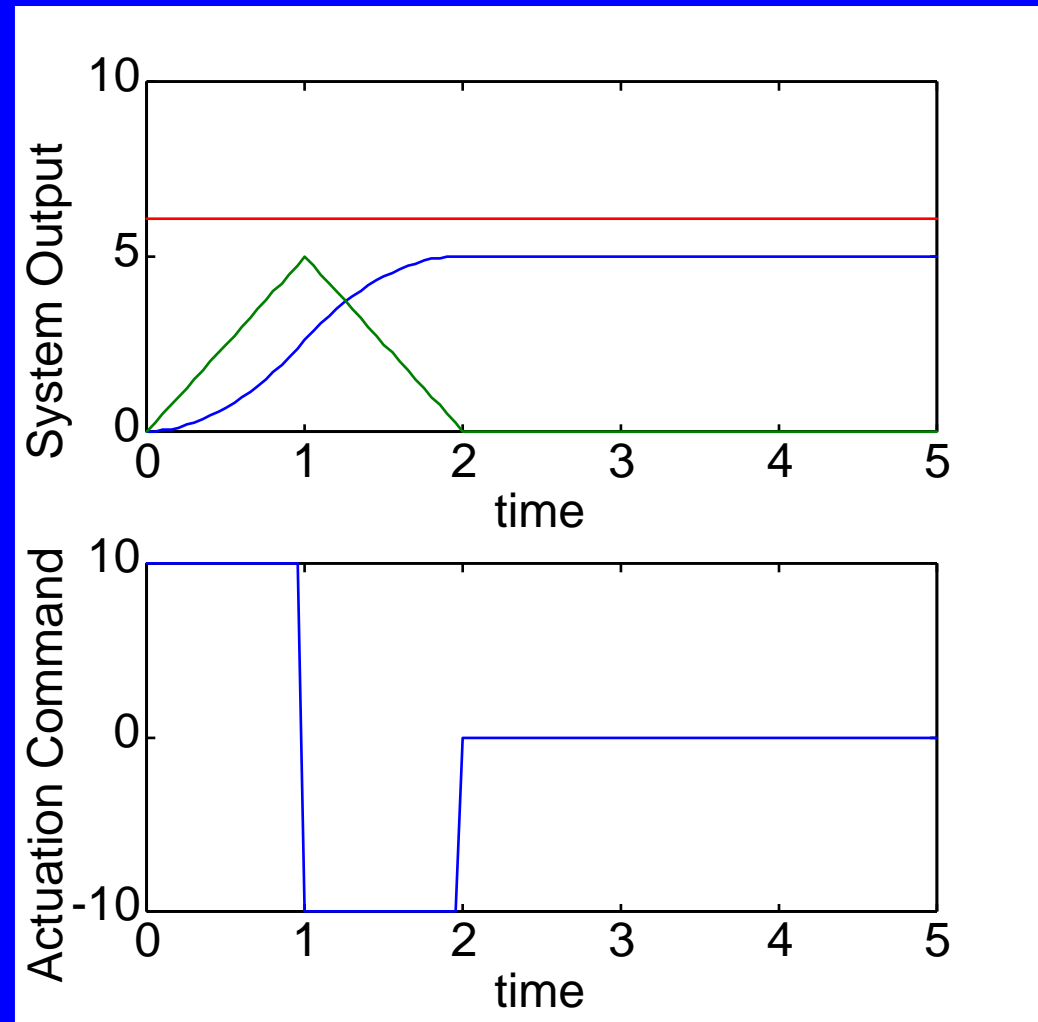
$$\frac{dv}{dt} = \frac{F}{m}; \quad \text{or,} \quad \frac{d^2x}{dt^2} = \frac{F}{m}$$

Simulations

- ❖ Single mass, IntroMass1.m
- ❖ Two masses connected by a spring
- ❖ Energy moves between modes but doesn't dissipate
- ❖ Dyn2Mass.m

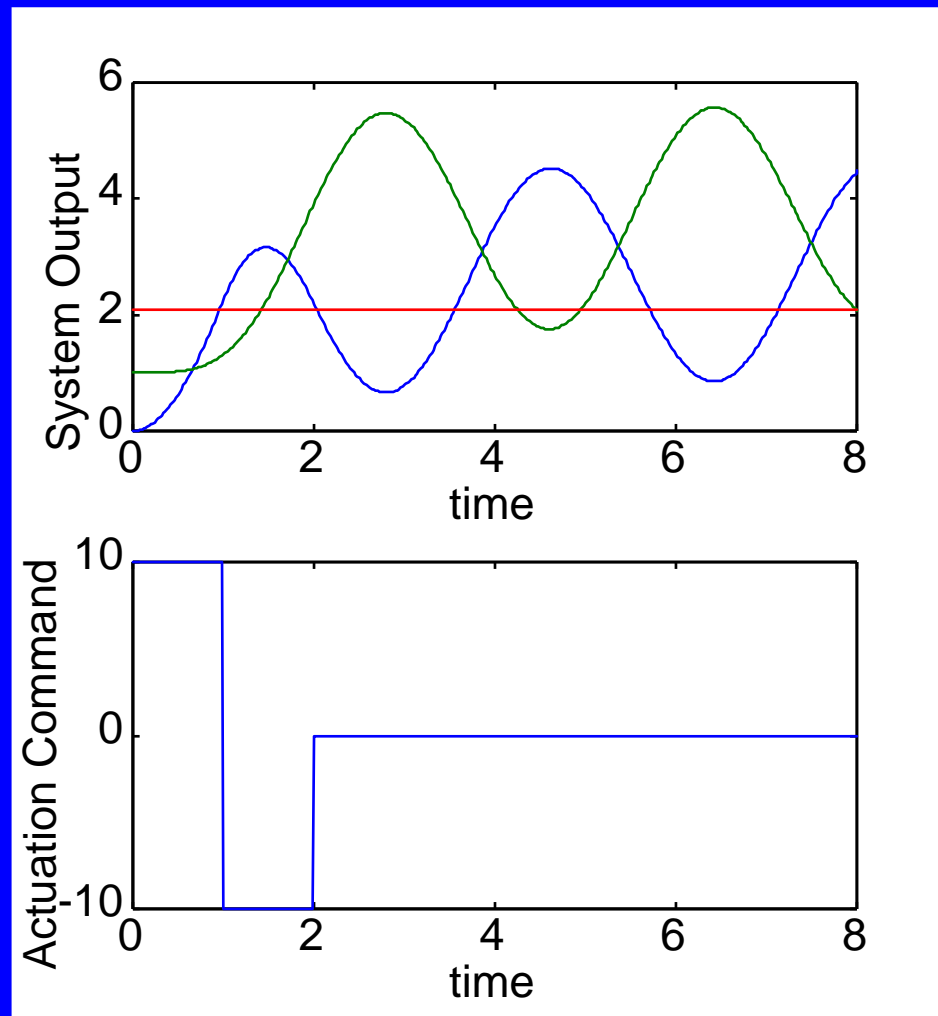
Single Mass (repeated)

- ❖ Velocity slope changes immediately
- ❖ Position (blue) hides change in force



Two Masses, No Feedback

- ❖ Energy stays internal to system
- ❖ Simple open loop control is unacceptable



Equilibrium, Stability

- ❖ State space: the Cartesian space with state variables on each axis
- ❖ Equilibrium: a point in the state space at which none of the state variables are changing
- ❖ Stability: the behavior near an equilibrium point
 - Stable: stays near the equilibrium point
 - Asymptotically stable: converges to the equilibrium point
 - Unstable: does not stay near the equilibrium point

Control and Stability

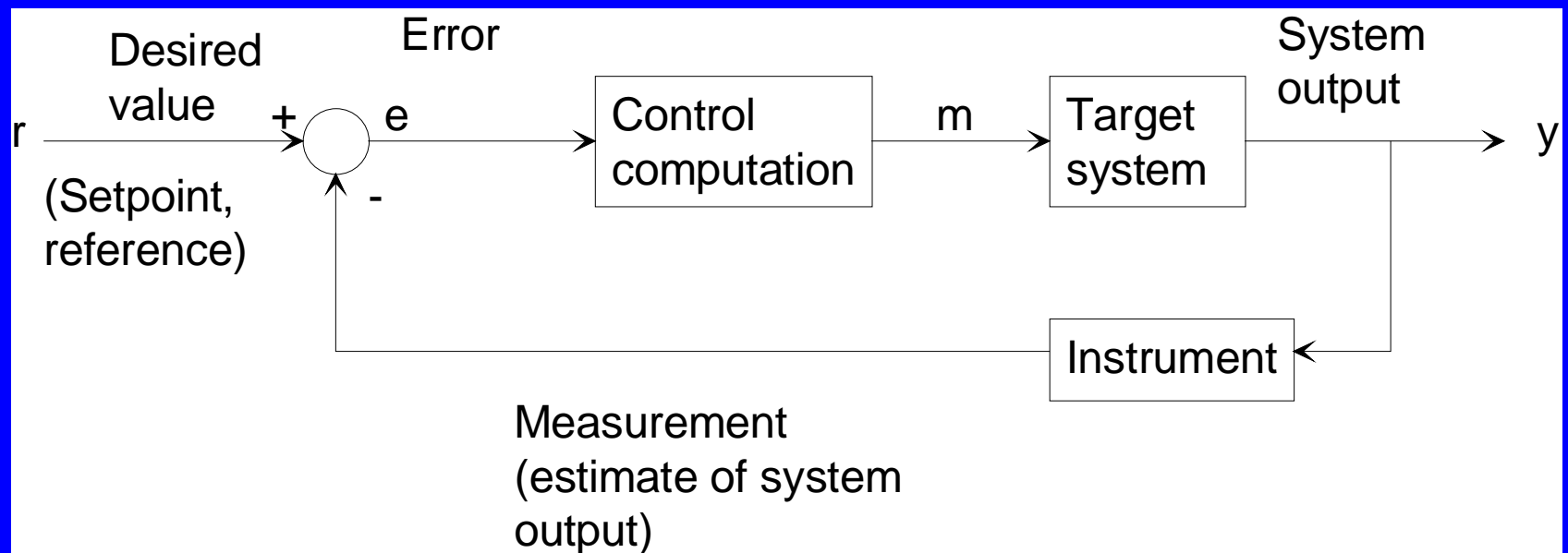
- ❖ Target system can be
 - Open-loop stable
 - Open-loop unstable
- ❖ First goal of feedback control: assure stability of closed-loop system
- ❖ Danger in applying feedback control: produce unstable behavior in case when target system is open-loop stable
- ❖ Feedback can be used to stabilize an open-loop unstable target system

III. Feedback Control Implementation

Single Loop Systems

- ❖ Single-input, single output systems (SISO)
- ❖ One actuator
- ❖ One instrument
- ❖ No limit to internal dynamic complexity
 - But hard to control if internal dynamics get too complicated

Canonical Diagram



Discrete/Continuous Time

- ❖ Various devices can be used for the “control computation”
- ❖ If computer is used its peculiarities must be accommodated
 - Sequential instruction execution
 - Finite precision
- ❖ Sequential execution means computers do only one thing at a time
- ❖ Result: sampled data; discrete time

Information Loss

- ❖ Alternative devices, such as op-amps, are continuous time
- ❖ Discrete time devices ignore information between samples
- ❖ Finite precision - causes noise injection into the control loop
- ❖ Computers make up for these deficiencies by their ability to implement complex computations economically

Computer Control

- ❖ Computers are currently the most common way of implementing feedback control
- ❖ Continuous-time implementations (usually op-amp, but could be dedicated digital logic) still used where computers aren't practical
- ❖ Formulations here will be presented for digital computer implementation

Dynamic Range

- ❖ The ability to handle a wide range of values
- ❖ Important in coping with a variety of situations
- ❖ Example, rapid slewing and slow scanning
- ❖ Hardest to achieve in actuators, then instruments, then computers

Control Algorithm

- ❖ Computes the manipulated variable (amplifier command)
- ❖ Uses error as input,
- ❖ Must figure out value of “m” that will bring the system output to its desired (setpoint) value
- ❖ First problem in designing the control algorithm is the dynamics of the target system ...
- ❖ The measured value of the output only hints at what the system is about to do!

PID

- ❖ Proportional/integral/derivative
- ❖ Most popular
- ❖ Broadly applicable
 - Not limited to any type or class of system
- ❖ Limited in its ability to cope with complicated dynamics

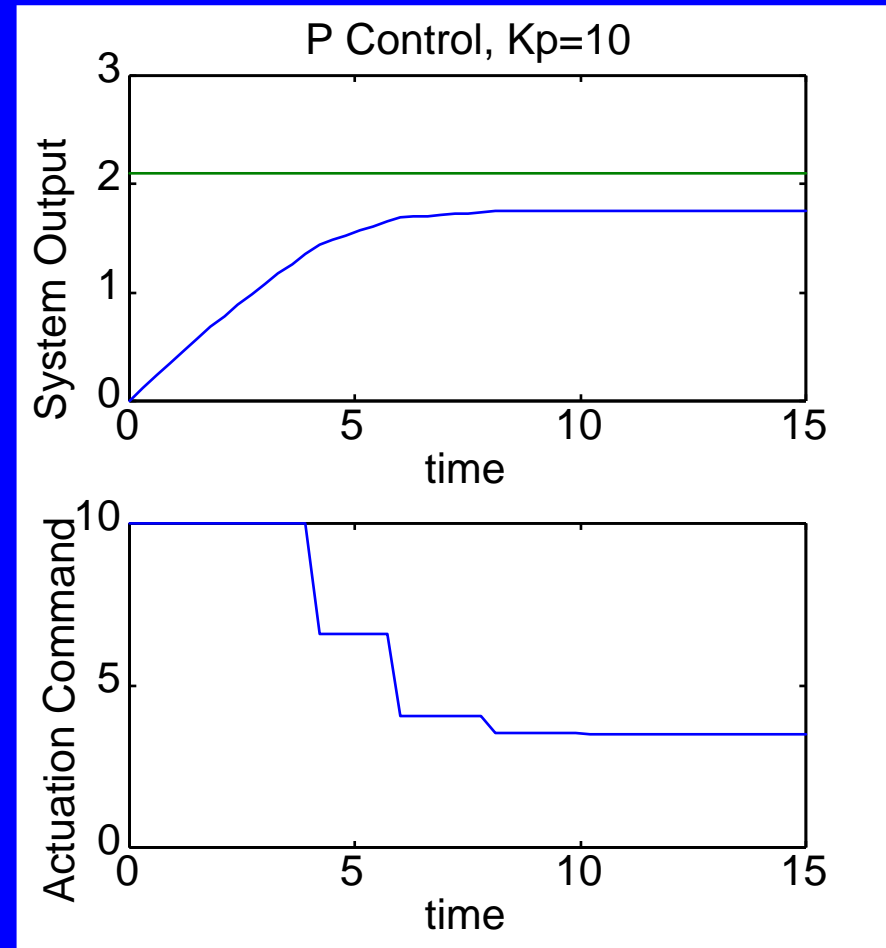
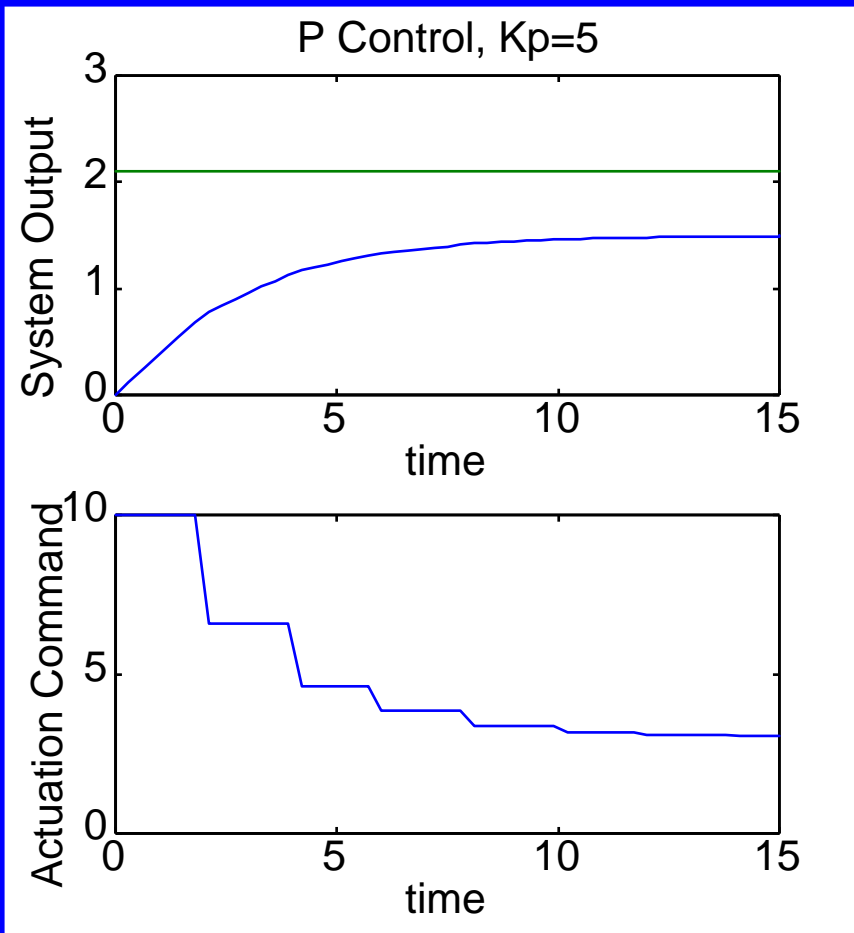
Proportional

- ❖ Strength of actuation proportional to error
- ❖ $m = k_p e$
- ❖ k_p is known as "gain"
- ❖ Simplest PID mode to apply
- ❖ May solve problem by itself

Simulations of P Control

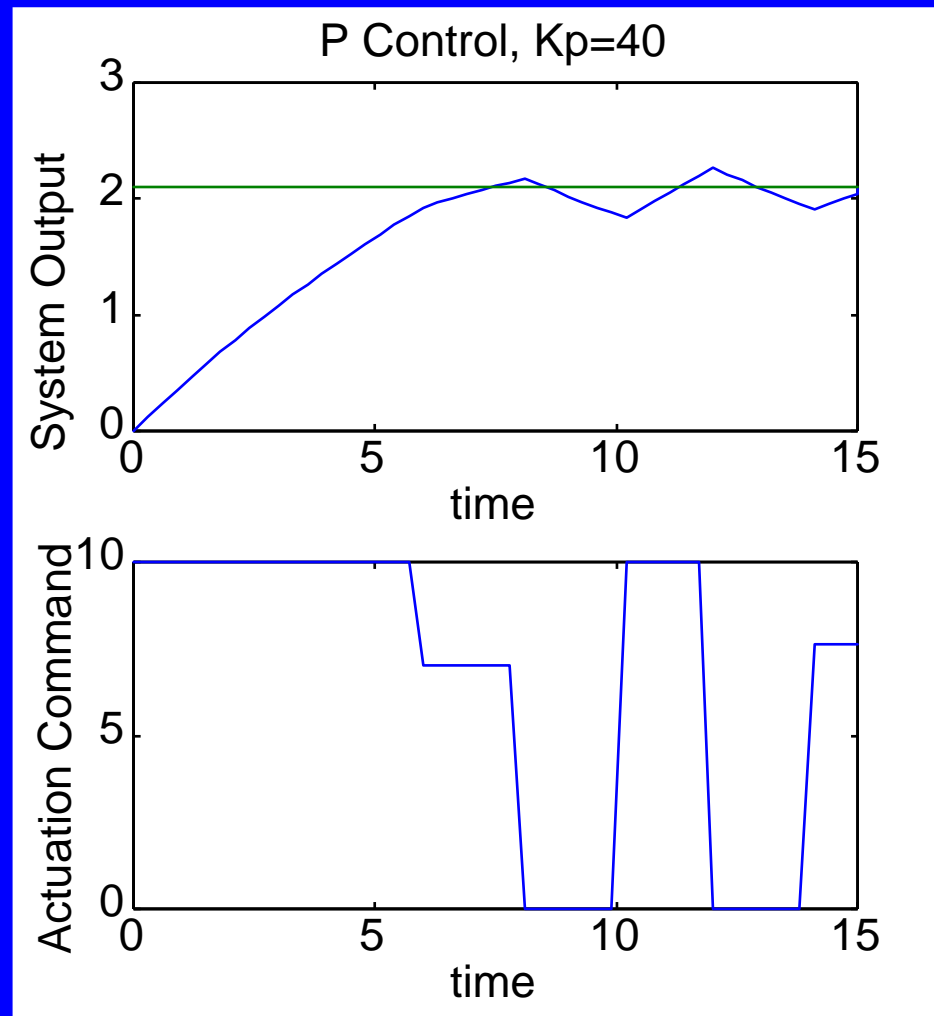
- ❖ Single tank, Prop1Tank.m
- ❖ Higher gain, Prop1TankHiGain.m
- ❖ Still higher gain, Prop1TankTooHiGain.m
- ❖ Slower sampling, Prop1TankSloSamp.m

Tank, Proportional Control



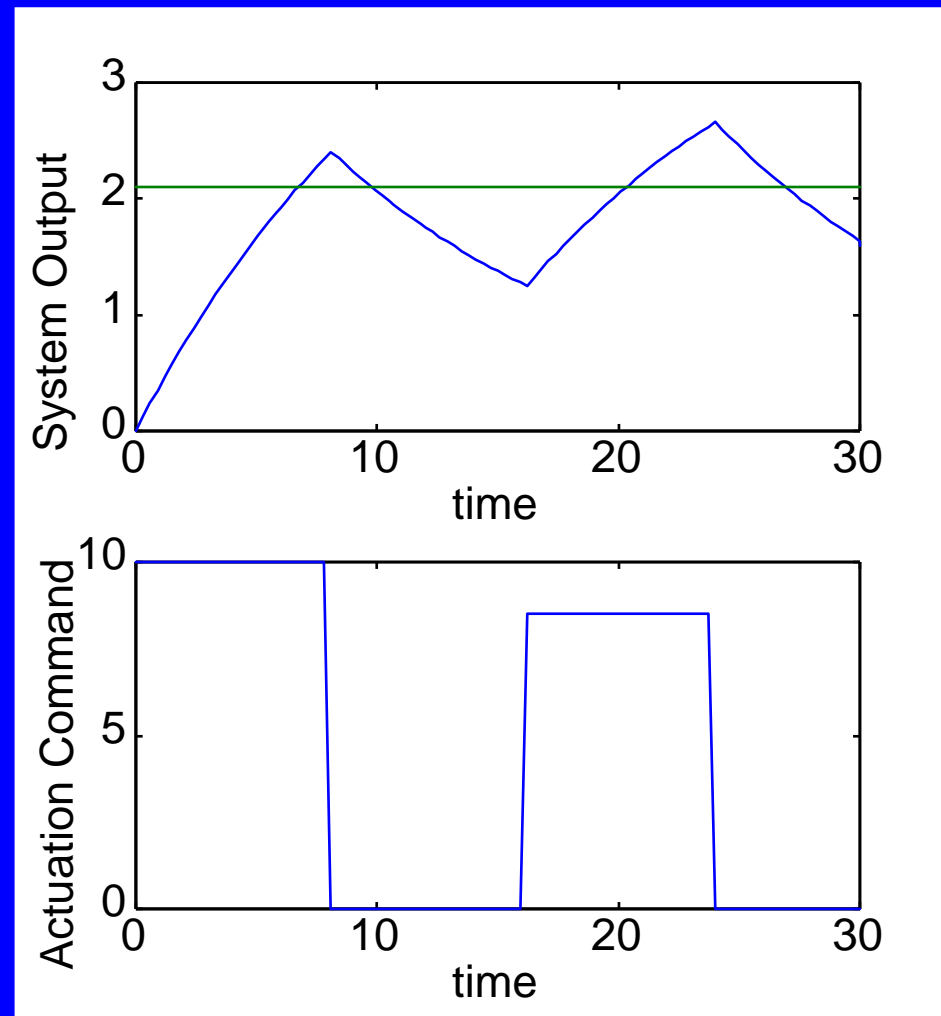
Tank - Very High P Gain

- ❖ Too much of a good thing!
- ❖ Error goes down
- ❖ But behavior is not acceptable



Tank - Slow Sampling

- ❖ Same gain as 2nd case ($k_p=10$)
- ❖ Behavior is much worse
- ❖ Sampling interval from 2 to 8



Integral

- ❖ Improves long-term behavior
- ❖ Note that desired level was never quite reached with just P control
- ❖ Adding I action can fix this
- ❖ "If the error persists, push harder!"

Integral Implementation

- ❖ Integral term formulation

$$I = k_i \int e dt$$

But, computer can only approximate this:

$$I = k_i \sum e \Delta t; \quad \text{or, in operational form:}$$

$$I(t + \Delta t) = I(t) + k_i e \Delta t$$

PI Control

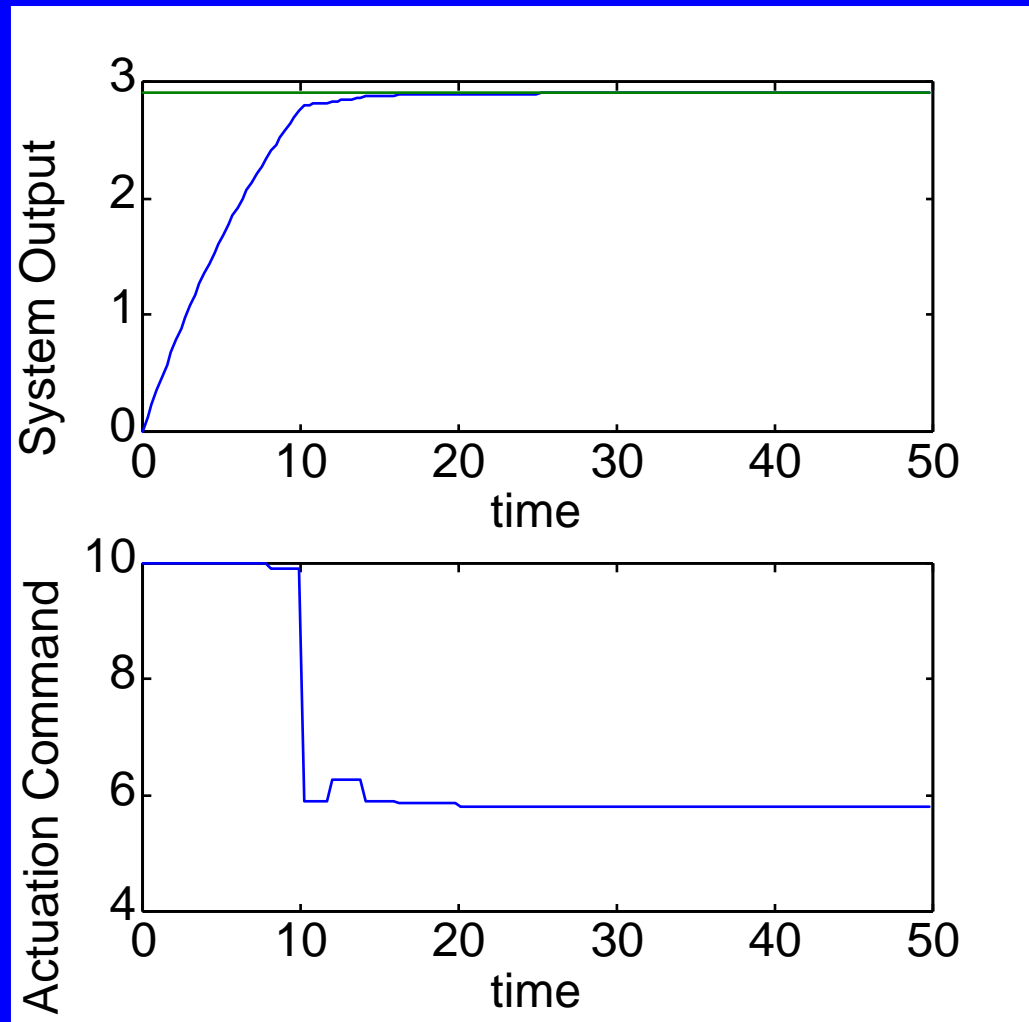
- ❖ Sum the proportional and integral terms
- ❖ The integral action is added for cases in which the long-term behavior is not satisfactory
- ❖ "Steady-state error"
- ❖ Adding the I term has a destabilizing effect
- ❖ Proportional gain may have to be lowered a bit

Simulation - Tank

- ❖ Single tank, effects of I gain: IntroTank2.m
- ❖ Higher I gain: PI-1TankHiGain.m

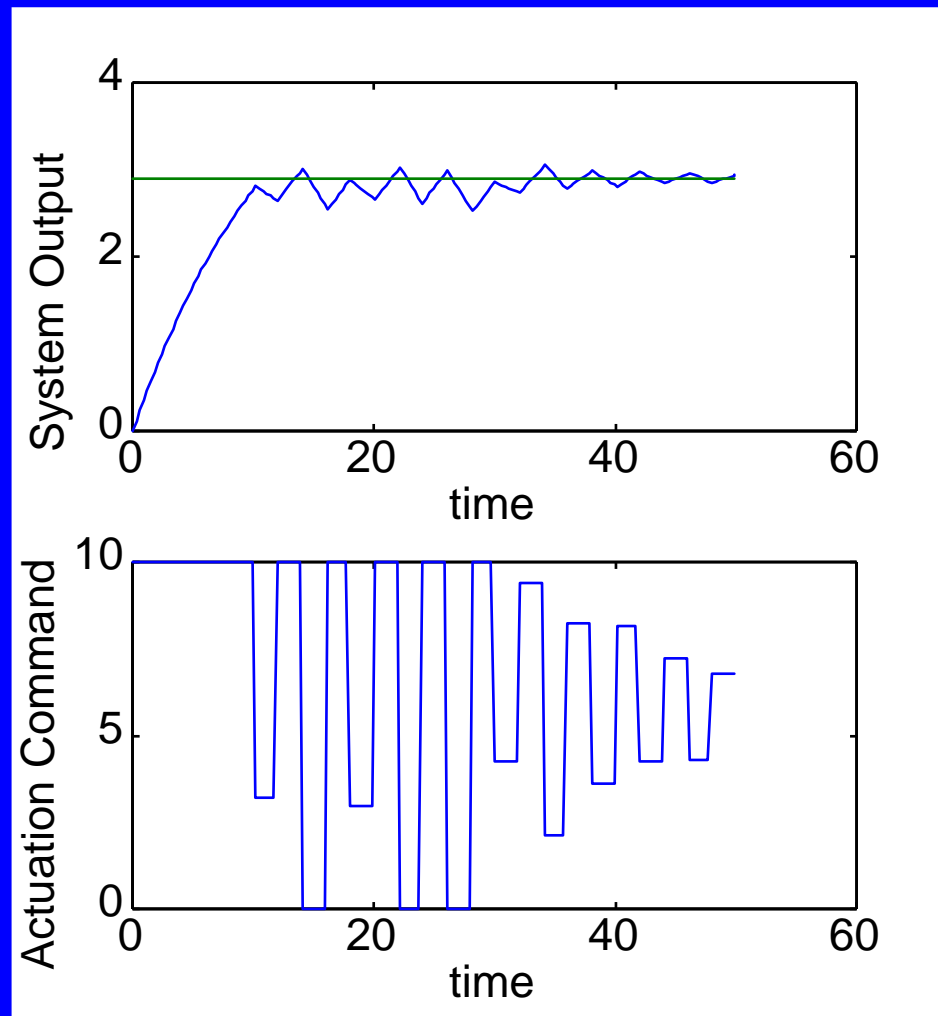
Tank - PI Control

- ❖ With integral action added the steady state error disappears



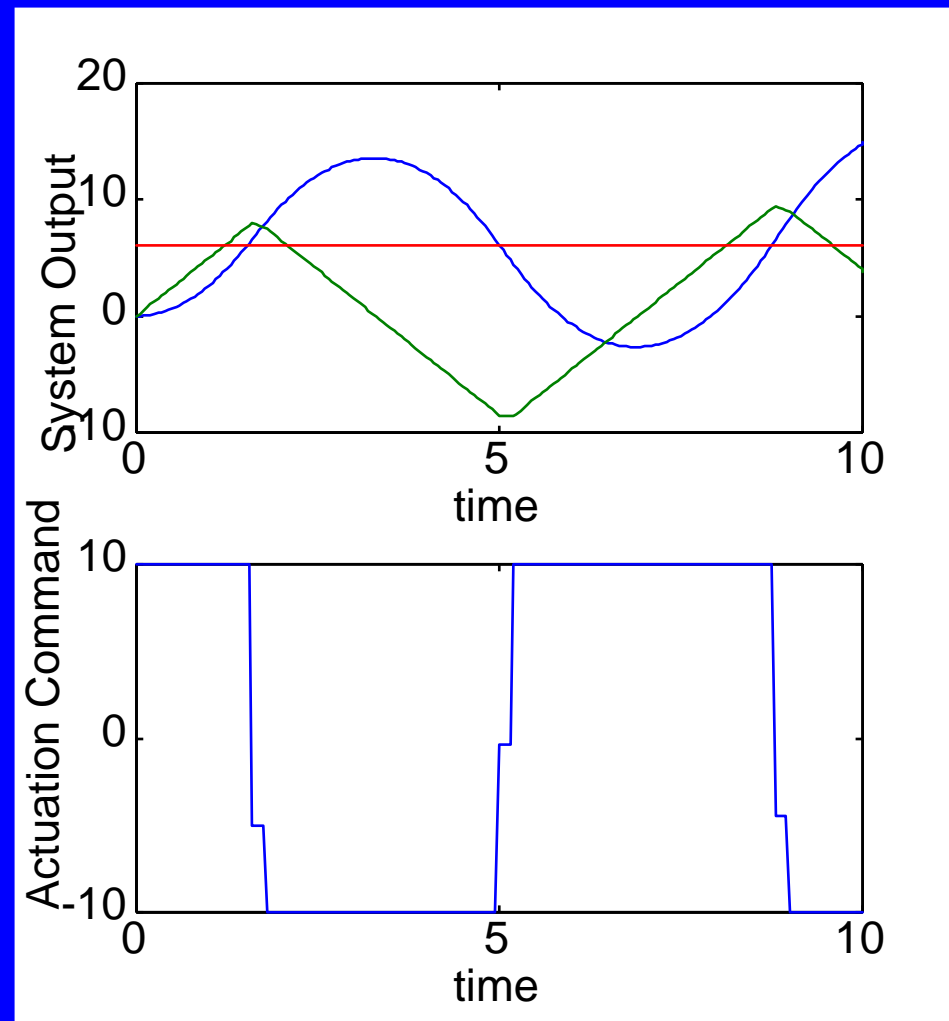
Tank - I Control, High Gain

- ❖ Raising the integral gain causes instability



Simulation - Position Control

- ❖ Single mass, P control:
P1Mass.m
- ❖ Unacceptable behavior with just P control



Derivative Action

- ❖ In this case, short term behavior is the problem
- ❖ Adding feedback makes the system marginally stable
- ❖ Derivative action can fix that
- ❖ Improves stability

Derivative Implementation

- ❖ Derivative action “anticipates”:

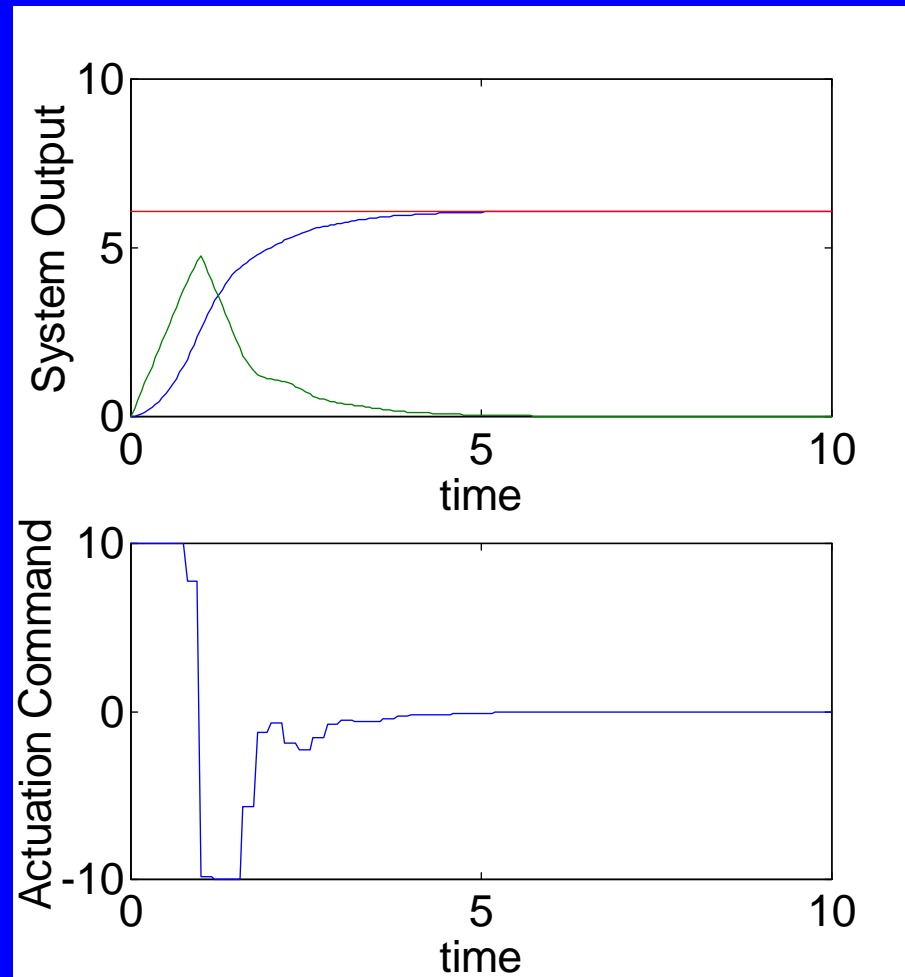
$$D = k_d \frac{de}{dt}$$

Computer approximation:

$$D = k_d \frac{\Delta e}{\Delta t}$$

Simulation PD

- ❖ Single mass, PD control:
PD1Mass.m
- ❖ D action stabilizes



Control Difficulty - Dynamics

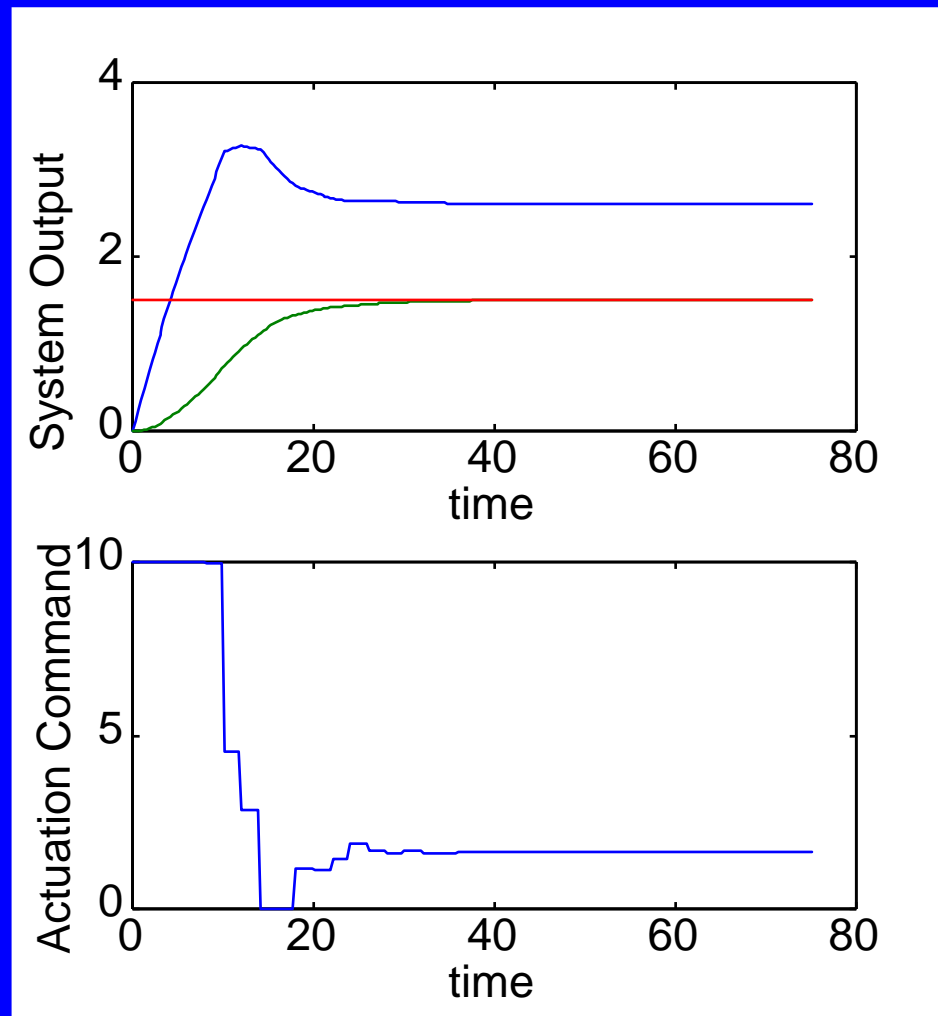
- ❖ Control gets much more difficult as internal dynamics increase
- ❖ Much more information is hidden so decisions based on feedback have to be more conservative
- ❖ Extreme example: pure time delay
- ❖ For example, controlling something on the moon with a controller on earth

Simulations - Dynamics

- ❖ Two tanks, connected by a pipe
- ❖ Control target is downstream tank
- ❖ Tank2PID.m
- ❖ Two masses, connected by a spring
- ❖ Control target is first mass (at the motor)
- ❖ Mass2.m

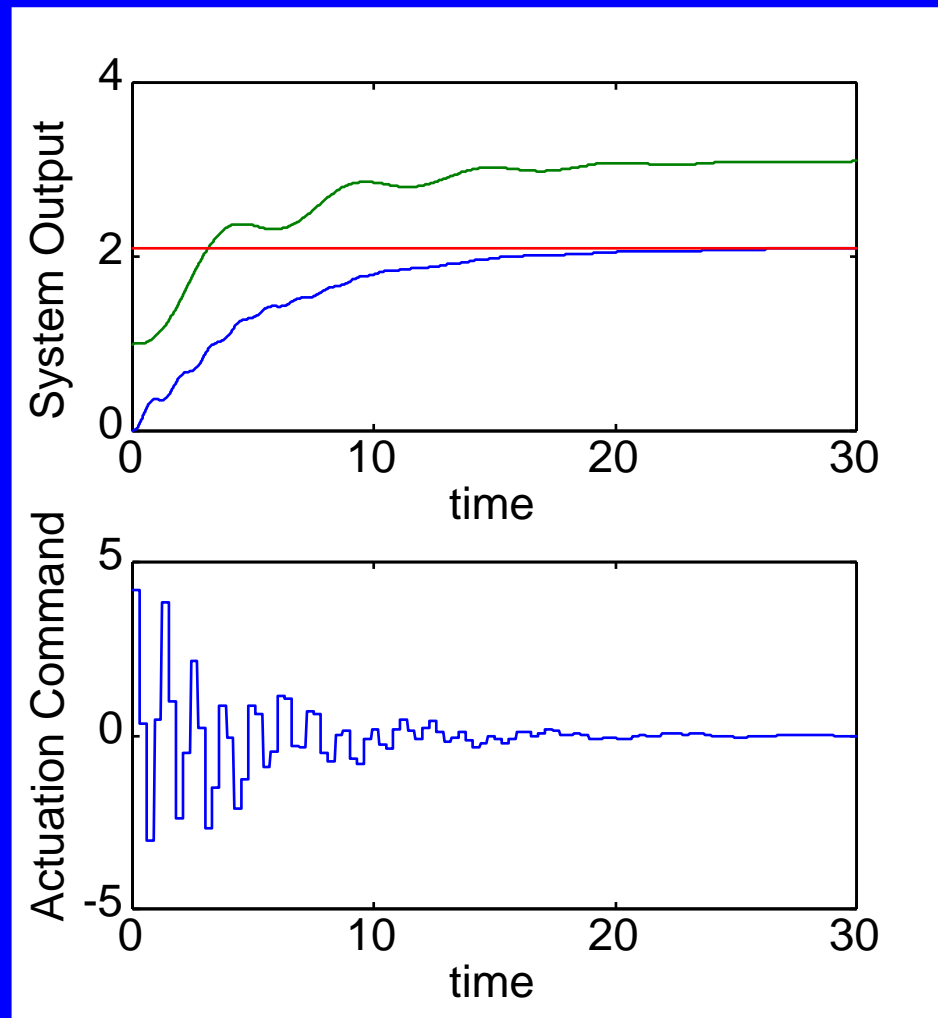
Two Tanks - with Feedback

- ❖ Needs full PID to work
- ❖ Settling time is double single tank
- ❖ Overshoot in first tank; slow rise in second



Two Masses

- ❖ Very difficult for controller to take energy out of the system
- ❖ Highly oscillatory behavior
- ❖ Setpoint target is mass that starts at 0 (blue)

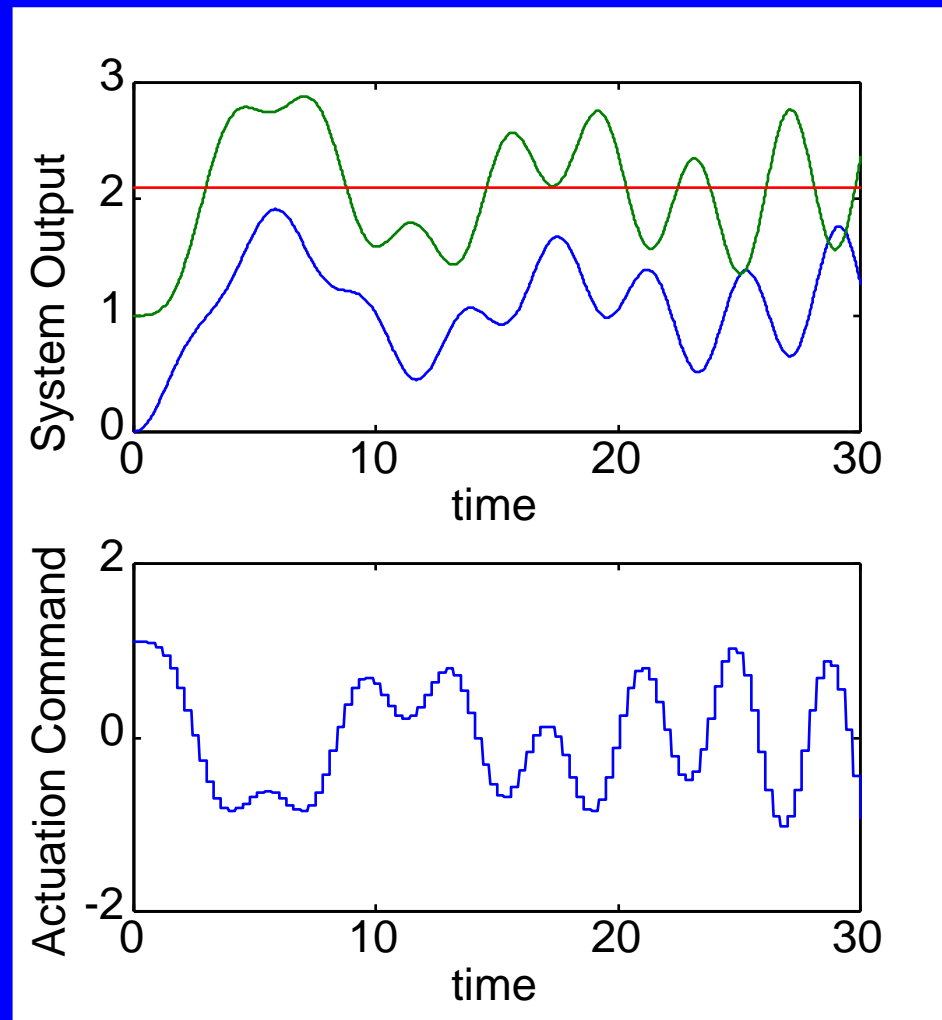


Point-of-Control

- ❖ Separating actuation and measurement makes control more difficult
- ❖ Example - applying force to first mass but using second mass as control target
- ❖ Mass2Split.m

Two Masses - Target #2

- ❖ Even with greatly reduced gains, system is not stable
- ❖ Splitting actuation and control makes life difficult!



Control Difficulty - Nonlinearity

- ❖ Linear: everything happens in proportion
- ❖ Double the input, the output doubles, etc.
- ❖ Mathematical: the dependent variables (states) in the differential equation appear only in linear form
- ❖ Mathematical: Superposition of responses
- ❖ Superposition: Given response to one input and response to another, the response to the sum of the inputs is the sum of the individual responses

Linear Mathematics

- ❖ Superposition is very powerful
- ❖ Only applies to linear systems
- ❖ Huge body of mathematical technique can be applied to linear systems
- ❖ If system is nonlinear, much more difficult to get general result
- ❖ Real world is mostly nonlinear!
- ❖ Lots of problems “forced” into linear formulation

Saturation

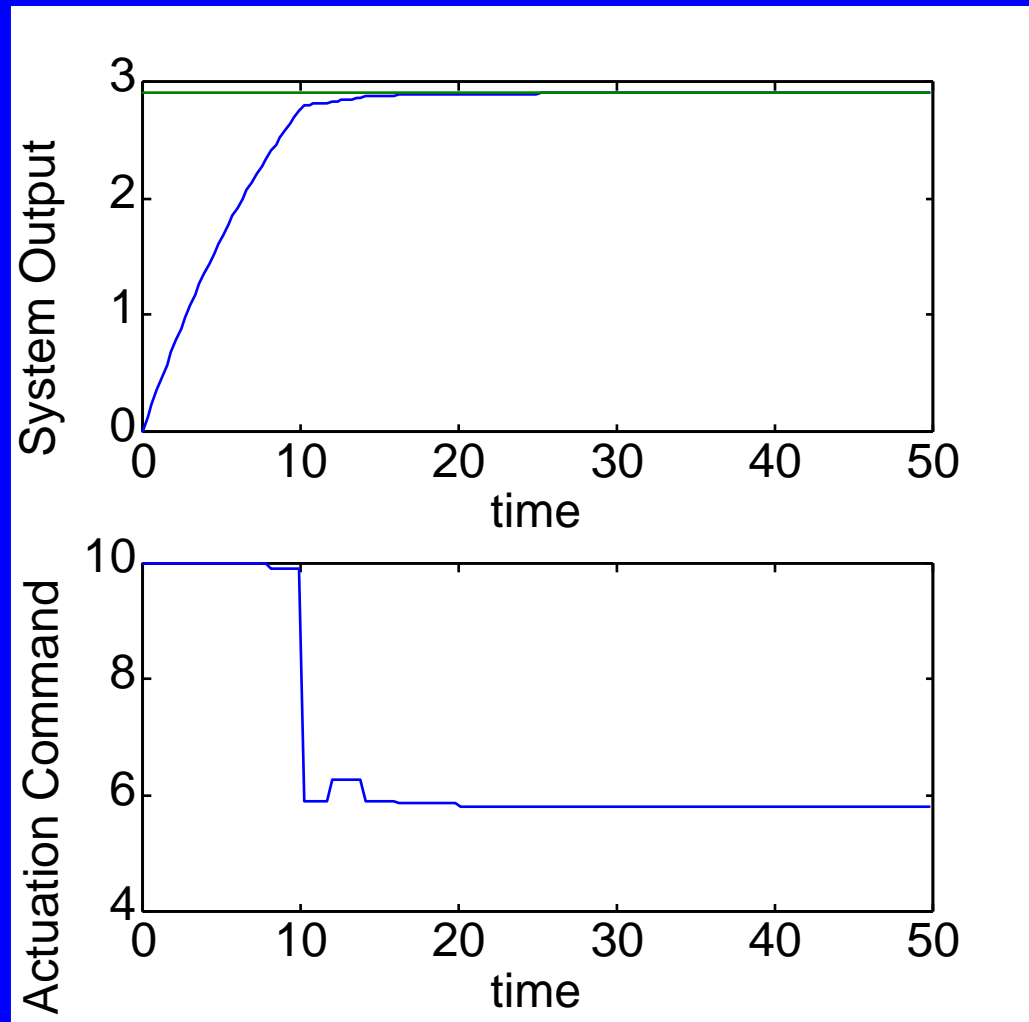
- ❖ Saturation: a limiting value for a variable
- ❖ Can be for an actuator
 - Limit of available power
 - Physical limit
- ❖ Or for a state
 - Safety limit, example - motor velocity limit
 - Physical limit, example - tank overflow
- ❖ Most common nonlinearity

Simulation - Saturation

- ❖ Sometimes saturation is benign
- ❖ Single tank: IntroTank2.m
- ❖ Sometimes dangerous: state variable saturation
- ❖ Internal state, not measured
- ❖ Two-tank overflow: OverFlow2Tank.m

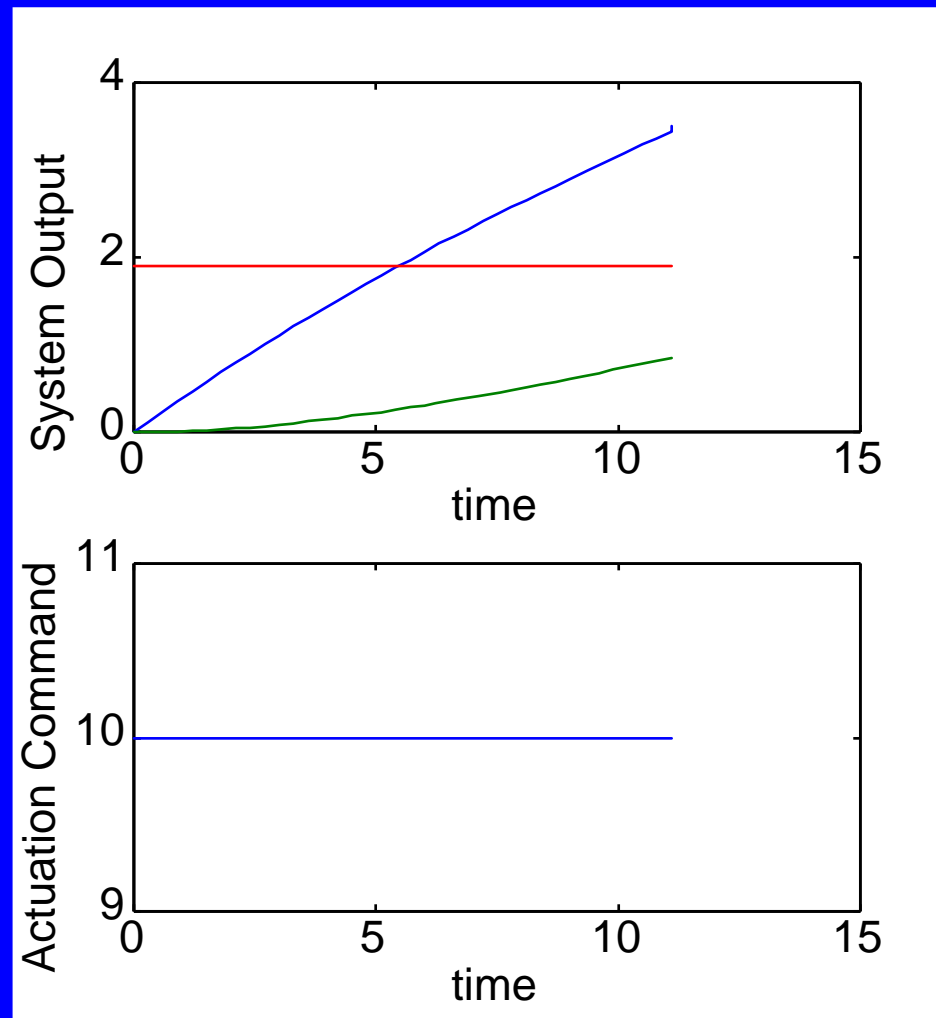
Actuator Saturation - Single Tank

- ❖ Actuator saturates early in response
- ❖ No harm done!



Two Tanks - Overflow!

- ❖ First tank overflows -- simulation is stopped
- ❖ Actuation saturated full time



Keep the Error Small: Setpoint Profiling

- ❖ Solution - keep the error small
 - Rule #1 of feedback control!
 - Hides a multitude of sin!
- ❖ Do this by making only realistic demands
- ❖ Setpoint profiling
- ❖ Rate-of-change of setpoint never greater than target system is capable of following

Keep the Error Small: Disturbances

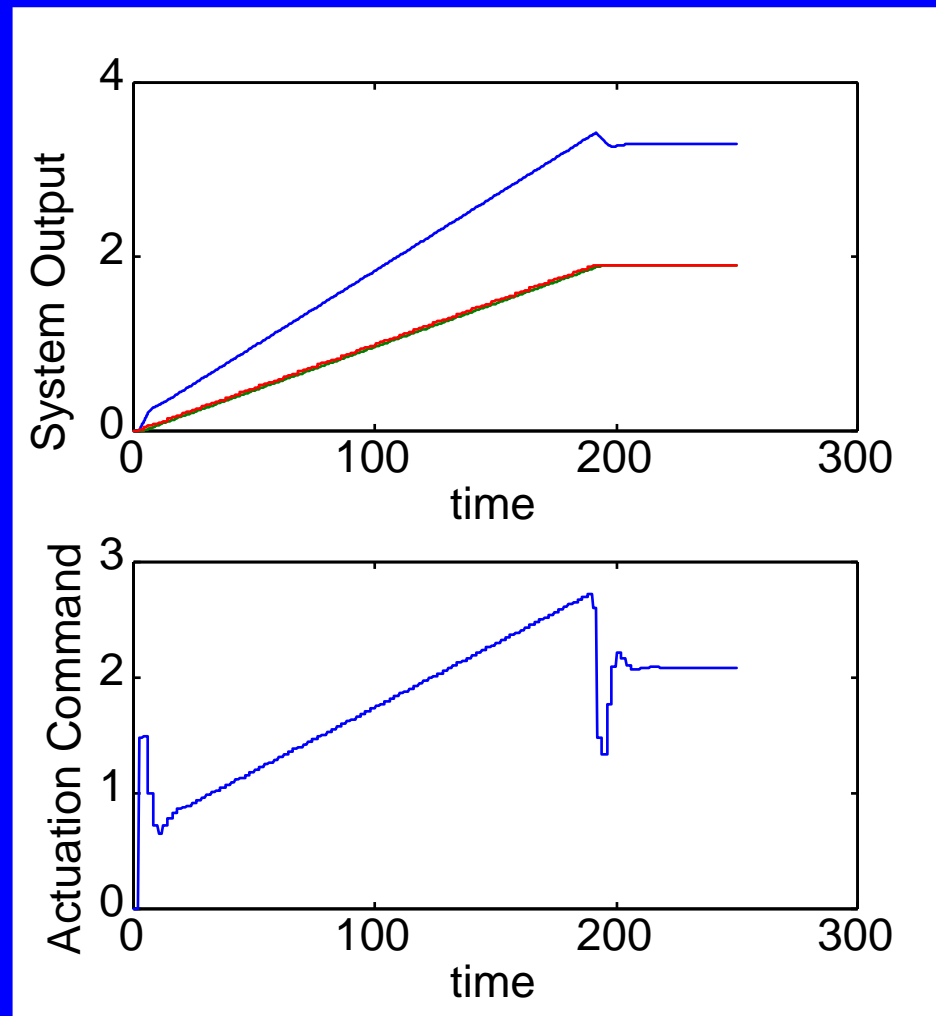
- ❖ The error can get too large because of disturbances
- ❖ Pushes response away from profile
- ❖ “Chase the system” to keep the error small
 - That is, change the setpoint to follow the system
 - Bring it back to the profile when disturbance subsides
- ❖ If system is open-loop unstable, saturation of the control means disaster!

Simulations - Profile

- ❖ Solving the tank overflow: Profile2Tank.m
- ❖ Position control of mass - velocity can get too high
- ❖ IntroMass2.m
- ❖ Use a profile to keep velocity in limit
- ❖ Profile1Mass.m

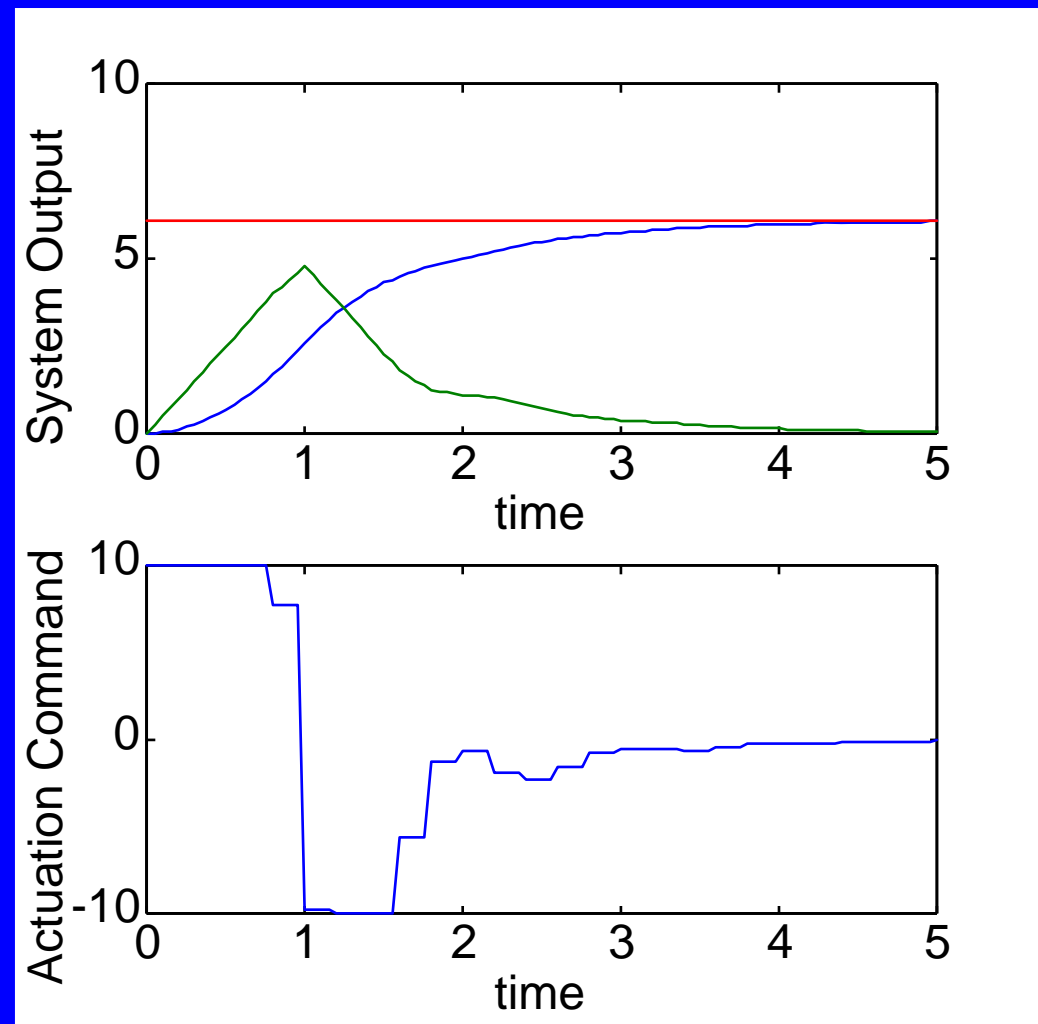
Two Tanks - with Profiled Setpoint

- ❖ Using a very slow profile keeps the overshoot to almost zero



Single Mass, Feedback (repeated)

- ❖ Velocity could be too high
- ❖ The larger the motion, the higher the velocity will go



Single Mass - with Profile

- ❖ Velocity is kept to a limit
- ❖ Maximum velocity reached will not change with size of move

