Control Systems

Chapter 7

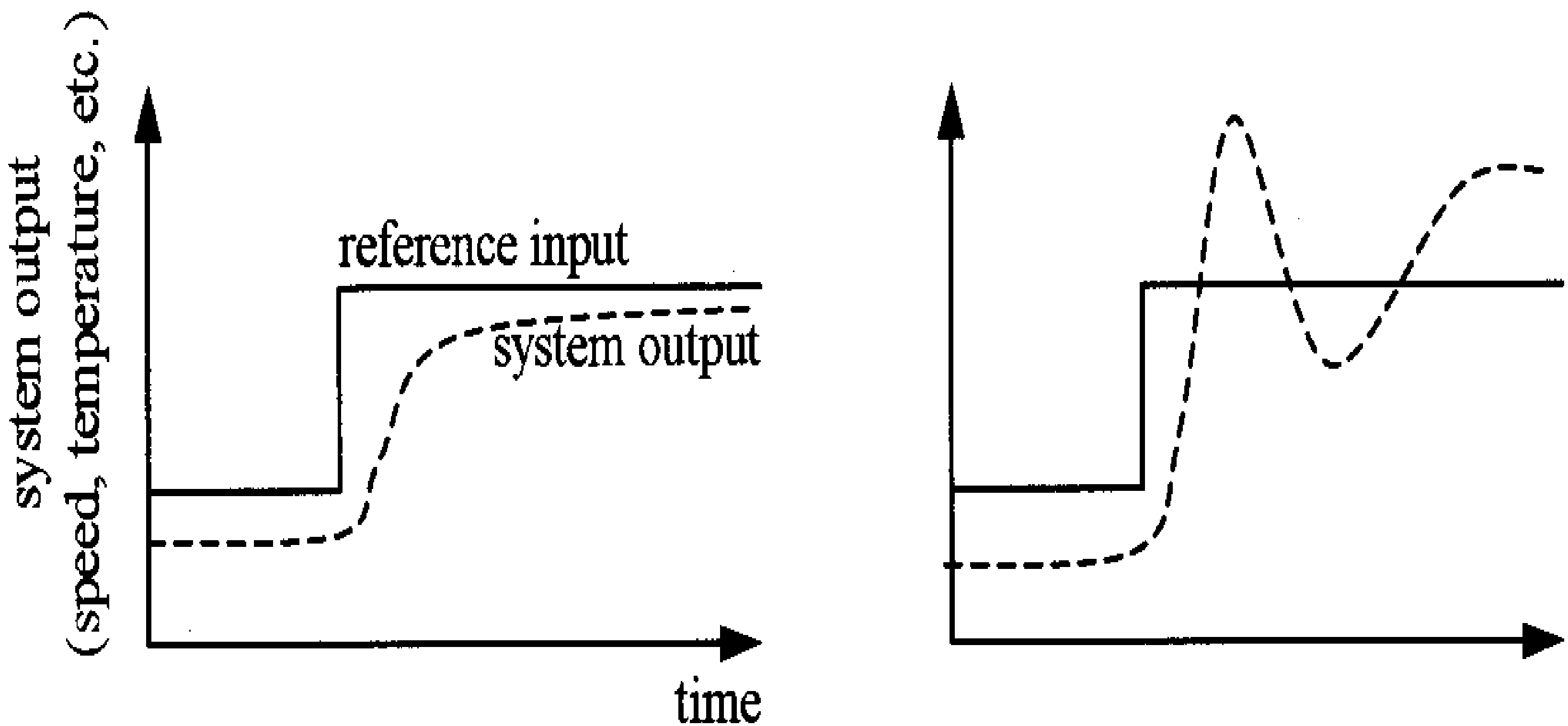
Outline

- Introduction
- Open and Closed Loop Control System Overview
- General Control Systems and PID Controllers
- Software Coding of PID Controller
- PID Tuning
- Practical issues Related to Computer-Based Control
- Benefits of Computer-Based Control Implementations

Introduction

- Control physical system's output
  - By setting physical system's input
- Tracking
- E.g.
  - Air Conditioner, automobiles.
- Difficulty due to
  - Disturbance: wind, road, tire, brake; opening/closing door…
  - Human interface: feel good, feel right

Tracking


system output
(speed, temperature, etc.)
reference input
system output
time

Open-Loop and Closed-Loop Control Systems Overview

- Plant
  - Also known as process
  - Physical system to be controlled
  - Example: Automobiles, fan, heater, disk
- Output
  - The particular physical system aspect we want to control
  - Example: Speed, temperature
- Reference
  - Input value that is desired to be seen at the output

…

- Actuator
  - Device used to control the input of the plant
  - Example: Motors
- Controller
  - System that computes input to the plant so as to achieve desired output from the plant
- Disturbance
  - Additional undesirable inputs

Open Loop Control Systems

- System in which output has no influence on control action of input signal
- Feed-forward control or non feed
- Delay in actual change of the output
- Controller doesn't know how well thing goes
- Simple
- Best use for predictable systems
  - Model is accurate and disturbance effect is minimal

…

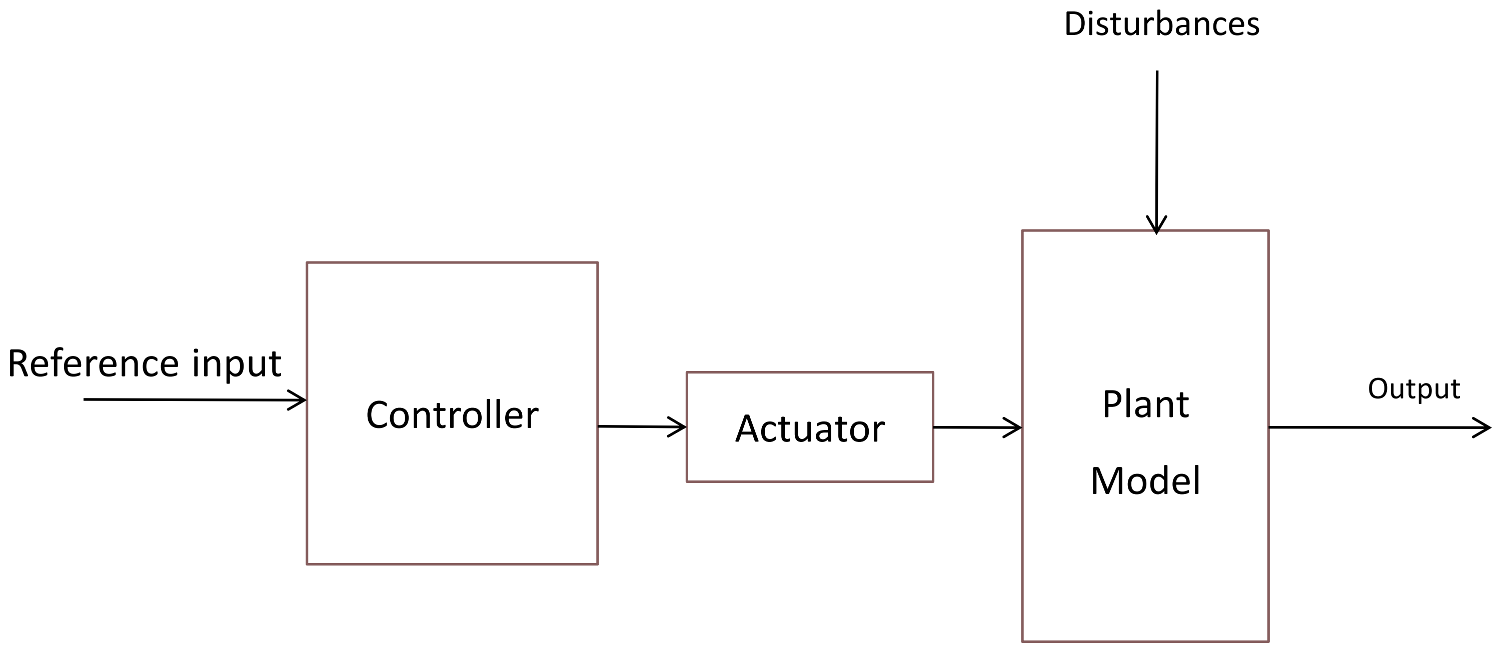


Figure: Simple Block Diagram of Open Loop Systems

Close Loop Control Systems

- Feedback control systems
- Minimize tracking error
- Additional Components
  - Sensor
    - Measure the plant output
  - Error detector
    - Detect Error

...

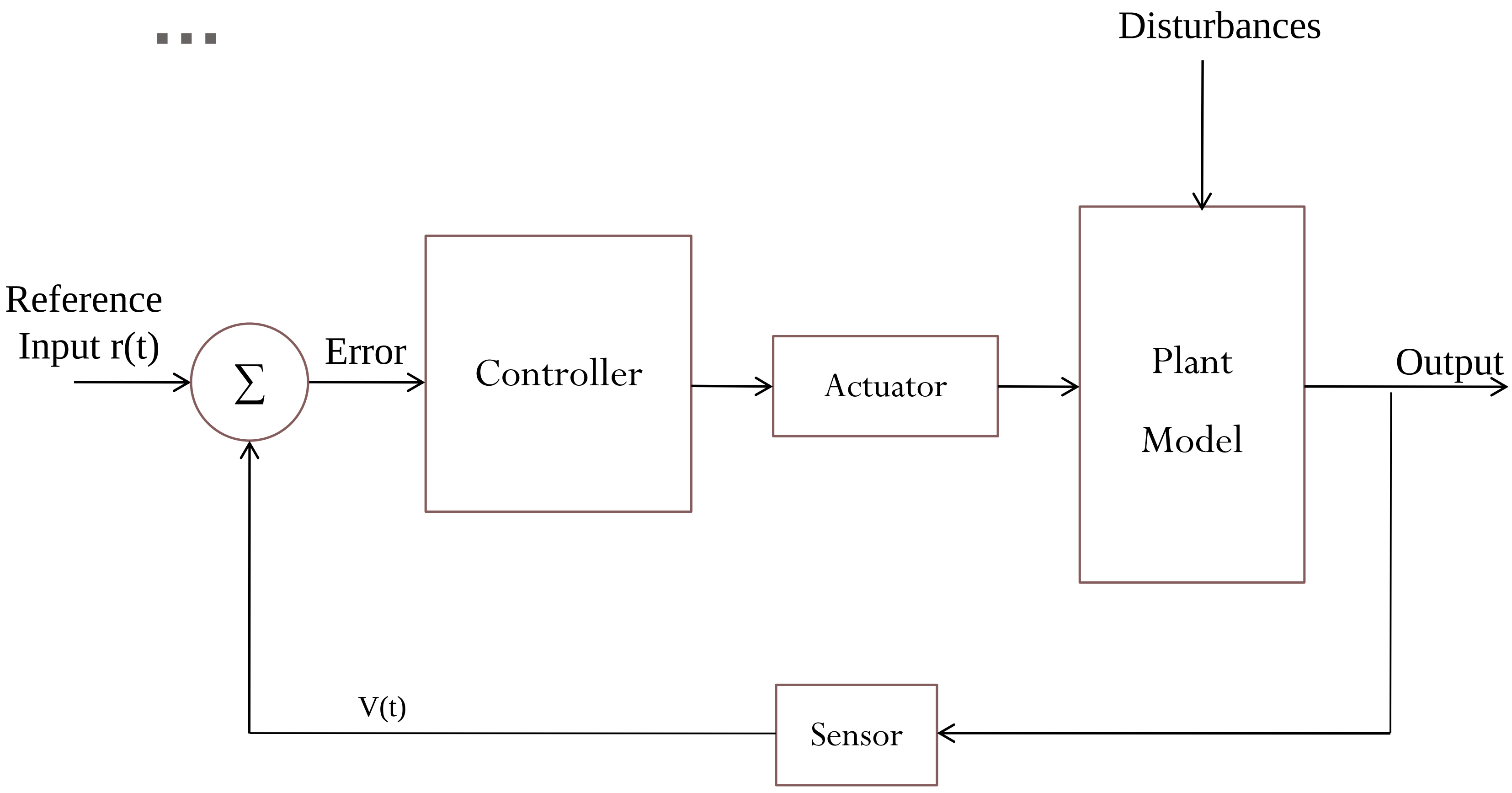


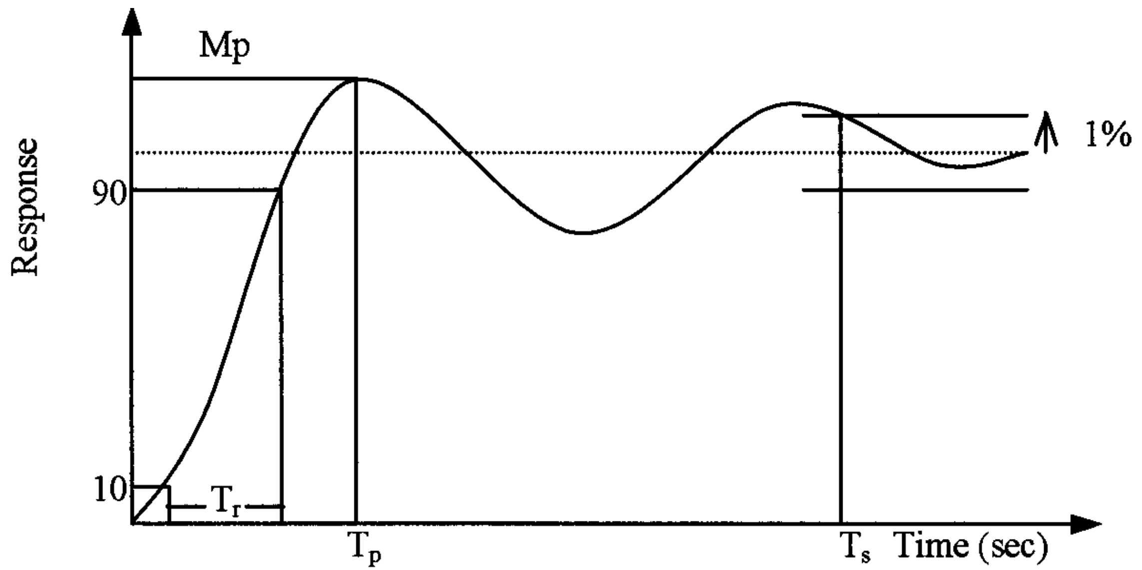
Figure: Simple Block Diagram of Closed Loop Systems

General Control Systems and PID Controllers

- Objective
  - Causing output to track a reference even in the presence of
    - Measurement noise, Model error, Disturbances
- Objectives evaluated through several metrics
  - **Stability**: All variables in the system remain bounded
  - **Performance**: How well an output tracks the reference
  - **Disturbance rejection**: cannot eliminate but can reduce its impact
  - **Robustness**: Ability to tolerate modeling error of the plant

Aspects of Performance

- Rise time: Time it takes to change from 10% to 90%
- Peak time: Time required to reach the first peak
- Overshoot: Percentage by which Peak exceed final value
- Settling time: Time it takes to reach 1% of final value



Modeling Real Physical Systems

- Real Physical Systems
  - Respond as continuous variables and as continuous function of time
    - Plant dynamic model is usually a differential equation
    - Sampling period selection much smaller than reaction time
  - Much more complex
    - Our model may not include all nonlinear effects, all system states, or all state interactons

Controller Design

- Proportional controller
  - A controller that multiplies the tracking error by a constant
    - $u_t = P * (r_t - v_t) = P * e_t$
- P affects
  - Transient response, steady state tracking, disturbance rejection
  - Trade offs
    - Reduce oscillation and improve convergence with worse steady-state error

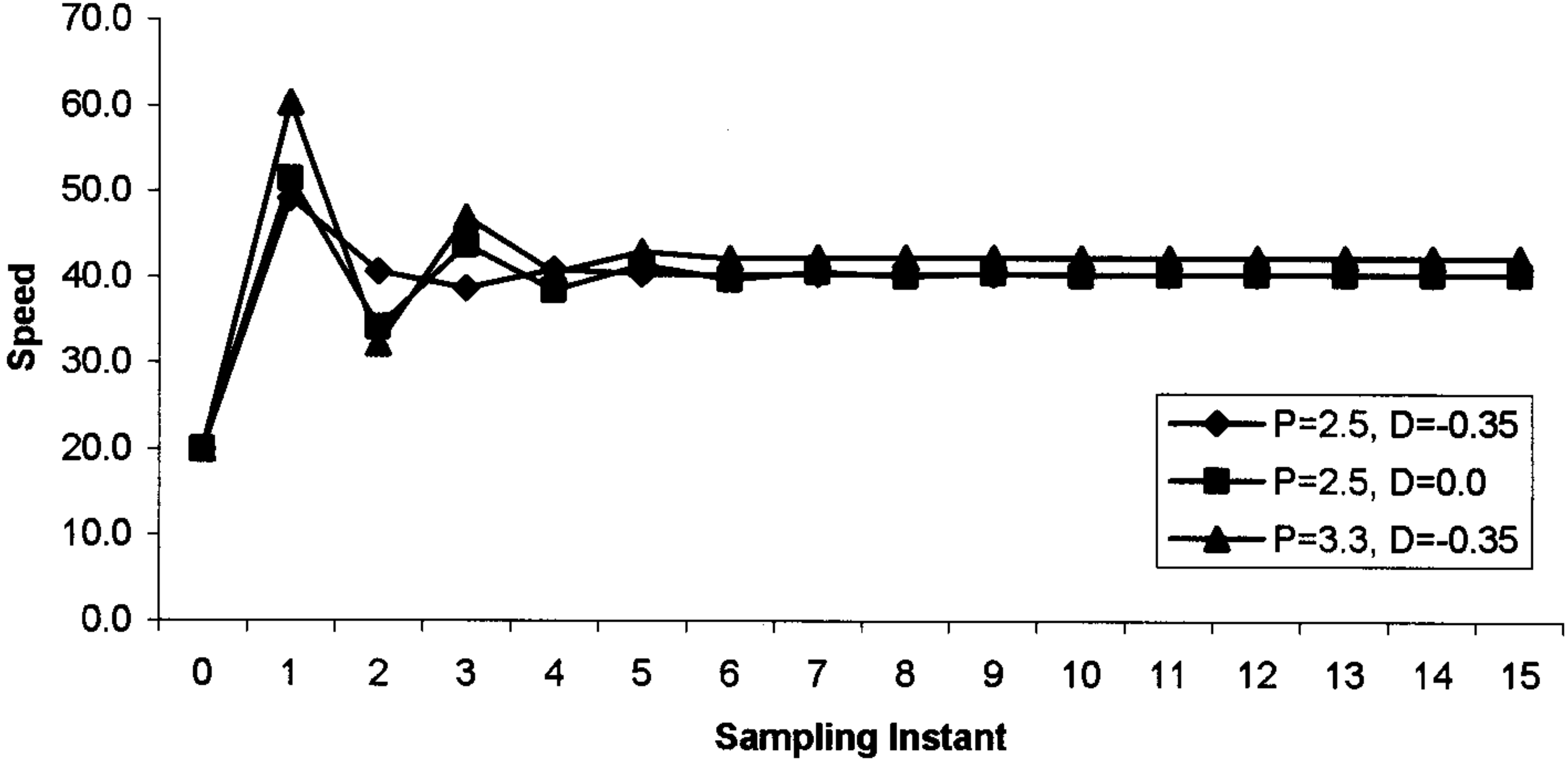
…

- Proportional and Derivative control (PD)
  - $u_t = P * (r_t - v_t) + D * ((r_t - v_t) - (r_{t-1} - v_{t-1})) = P * e_t + D * (e_t - e_{t-1})$
    - P: Proportional Constant
    - D: Derivative Constant
    - u_t: Output
    - e_t: measured error
    - $e_t - e_{t-1}$: derivative of the error
- Allow greater flexibility in the optimization

…

- Derivative term used to predict the future
  - Looks the difference between two successive time instances
- More Complex Controller
  - Need to keep track of error derivative
- PD give more flexibility
  - P term for best tracking and disturbance control
  - D term effects transient response only
    - Control oscillation, overshoot and rate of convergence

…

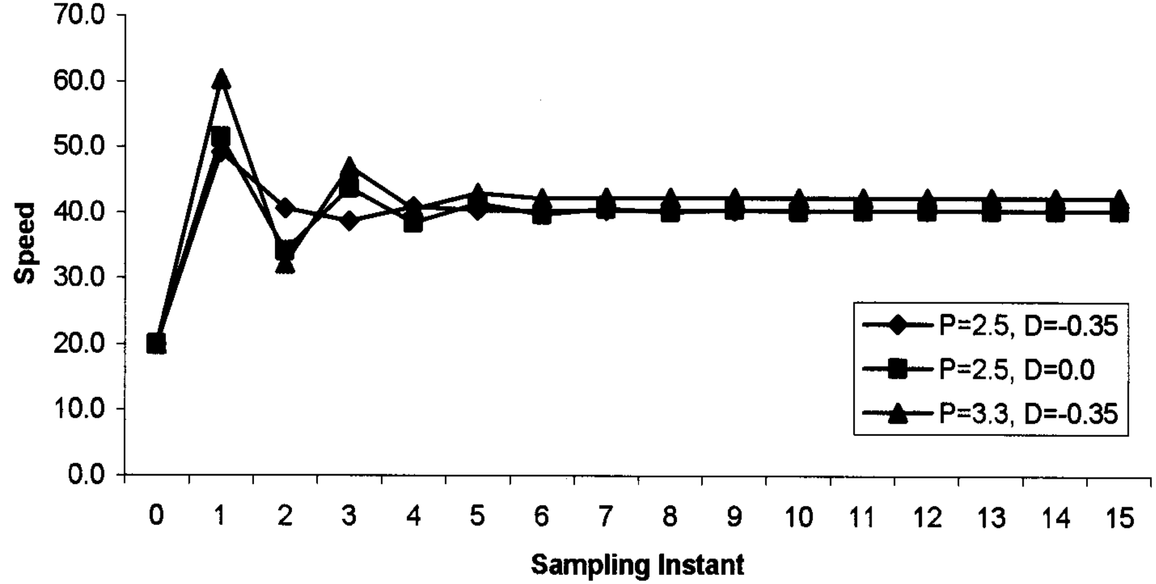


70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
Speed
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Sampling Instant
P=2.5, D=-0.35
P=2.5, D=0.0
P=3.3, D=-0.35

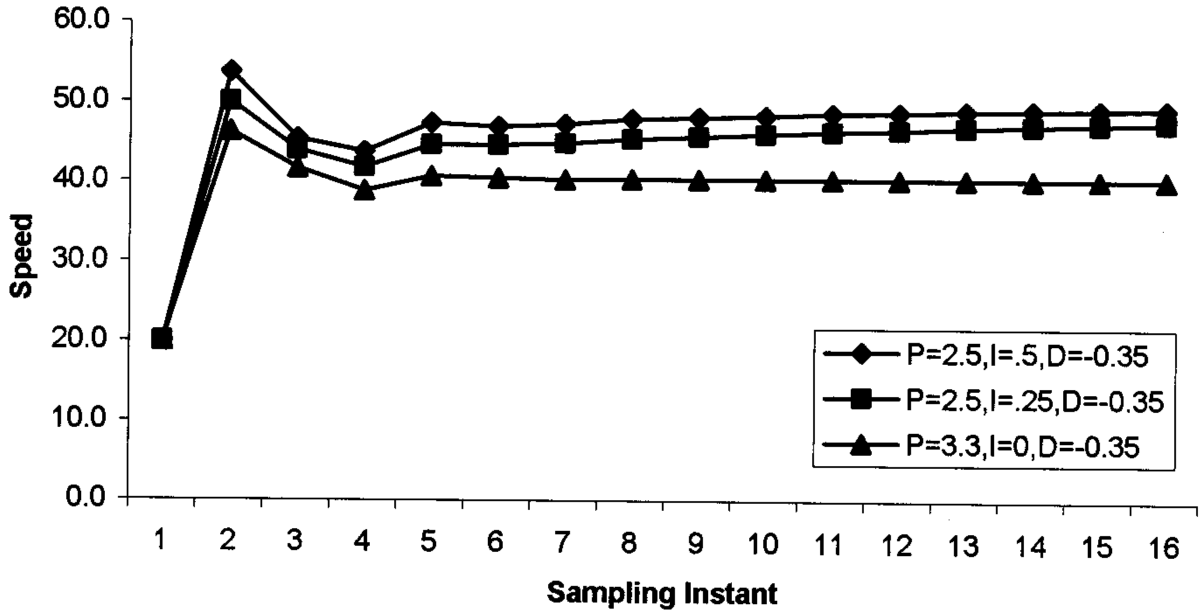
…

- PI Control - Proportional plus Integral Control
  - $u_t = P*e_t + I*(e_0 + e_1 + \ldots + e_t)$
- Sum up error over time
  - Ensure reaching desired output, eventually
  - v_{ss} will not be reached until $e_{ss}=0$
- Use P to control disturbance
- Use I to ensure steady state convergence and convergence rate

...



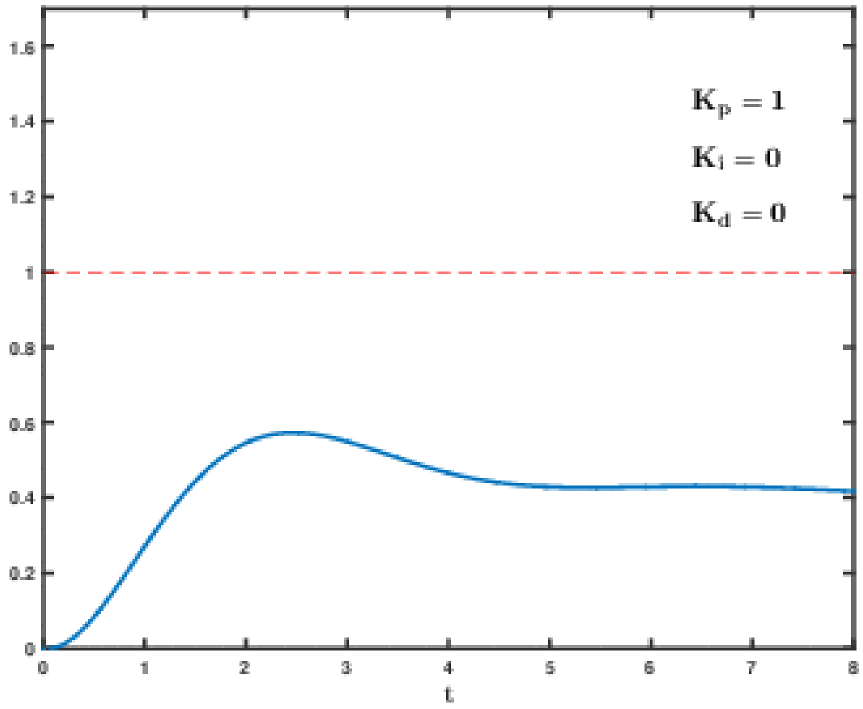
Speed
Sampling Instant
P=2.5, D=-0.35
P=2.5, D=0.0
P=3.3, D=-0.35



Speed
Sampling Instant
P=2.5,I=.5,D=-0.35
P=2.5,I=.25,D=-0.35
P=3.3,I=0,D=-0.35

…

- PID Control
  - Combine Proportional, integral, and derivative control
    - $u_t = P*e_t + I*(e_0 + e_1 + \ldots + e_t) + D*(e_t - e_{t-1})$
- Select the PID gains to achieve the desired stable transient behavior



$K_p = 1$
$K_i = 0$
$K_d = 0$
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0 1 2 3 4 5 6 7 8
t

Software Coding of a PID Controller

- Initialization
  - Initialize P, I, D gain
- Main function loops forever, during each iteration
  - Read plant output sensor
    - May require A2D
  - Read current desired reference input
  - Determine actuator value
  - Set actuator value
    - May require D2A

Pseudo code for design

- Set the values of Pgain, Dgain, Igain.
- Initialize prior error=0 and integral=0
- Repeat following steps
  - sensorValue=getValueFromSensor();
  - refValue=getReferenceValue();
  - Error=sensorValue-refValue;
  - Integral=integral+error*iterationTime
  - Derivative=(error-prior_error)/iterationTime
  - Output=Pgain*error+ Igain*integral+Dgain*derivative
  - setActuator(output)
  - Prior_error=error
  - Wait(iterationTime)

PID tuning

- Values of P, I, and D can be determined through quantitative analysis
- Quantitative analysis not necessary when
  - Safety is not a concern
  - Cost of using plant is not a concern
  - PID values selected using ad hoc process
- Advantages of Ad hoc tuning
  - Model of plant may be too complex to analyze quantitatively
  - Model may not available

…

- Ad hoc method for getting "reasonable" P, I, D
  - Start with a small P, I=D=0
  - Increase D, until seeing oscillation
    - Reduce D a bit by 2 to 4 factor
  - Increase P, until seeing oscillation or excessive overshoot
    - Reduce P a bit
  - Increase I, until seeing oscillation or excessive overshoot
- The above steps are repeated until satisfactory performance is achieved

P, I and D summary

- A proportional controller (Kp) will have the effect of reducing the rise time and will reduce, but never eliminate, the steady-state error.
- An **integral control (Ki)** will have the effect of eliminating the steady-state error, but it may make **the transient response worse**.
- A **derivative control (Kd)** will have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response but little effect on rise time
- A PD Controller could add damping to a system, but the steady-state response is not affected.(steady state error is not eliminated)
- A PI Controller could improve relative stability and eliminate steady state error at the same time, but the settling time is increased(System response sluggish)

Practical Issues with Computer-Based Control

- Quantization
- Overflow
- Aliasing
- Computation Delay

Quantization

- It occurs when machine number is altered to fit the constraints of memory
  - Arithmetic results requiring more precision than original values - 0.50 x 0.25 = 0.125
  - Analog signals from sensors quantized by ADC
- If 0.36 were to be stored as a 4-bit fraction
  - 0.75, 0.50, 0.25, 0.00, -0.25, -0.50, -0.75, -1.00 possible
  - Saved number would be 0.25 with error of 0.11

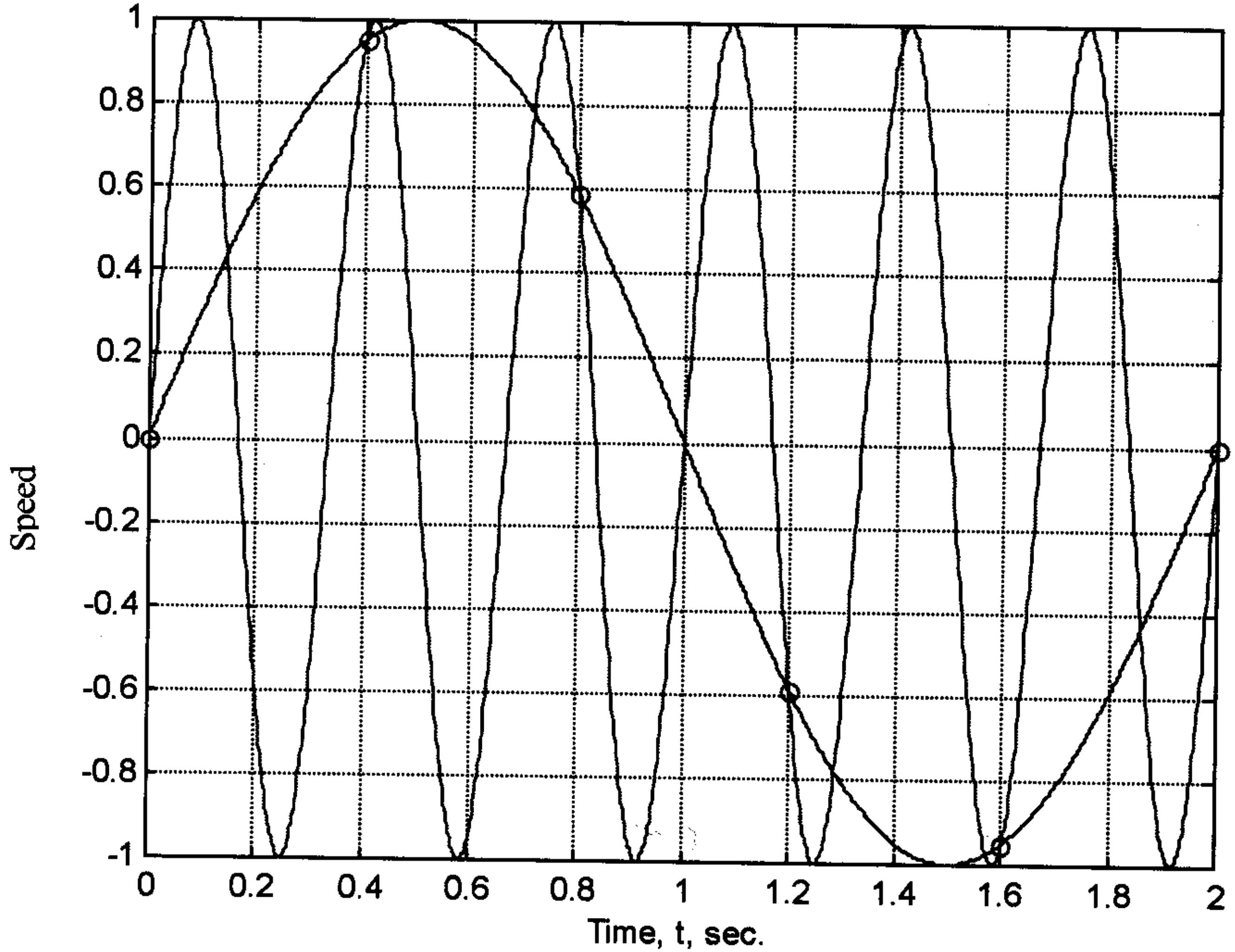
Overflow

- Operation outputs large magnitude number
  - Can’t store $0.75+0.50 = 1.25$ as 4-bit fractional number

Aliasing

- Reconstructed signal different from original signal
- Causes different signals to become indistinguishable
- Example
  - Sampling at 2.5 Hz, period of 0.4, the following are indistinguishable
    - $y(t)=1.0*\sin(6\pi t)$, frequency 3 Hz
    - $y(t)=1.0*\sin(\pi t)$, frequency of 0.5 Hz
  - In fact, with sampling frequency of 2.5 Hz
    - Can only correctly sample signal below Nyquist frequency $2.5/2 = 1.25$ Hz

…



Speed
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Time, t, sec.

Computation Delay

- Delay results in control signal being applied later than desired
- Too much delay results in performance degradation
- Need to characterize implementation delay to make sure it is negligible
- Hardware delay is usually easy to characterize
  - Synchronous design
- Software delay is harder to predict
  - Should organize code carefully so delay is predictable and minimized
  - Write software with predictable timing behavior (be like hardware)

Benefit of Computer Control

- Repeatability, Reproducibility and stability
  - Analog circuits more prone to aging, temperature and manufacturing tolerance effects – results may vary with time
  - Digital systems will compute identical results
- Programmability
  - Allows advanced features to be implemented easily
    - Adaptive behavior, data storage, on-line performance evaluation and so on