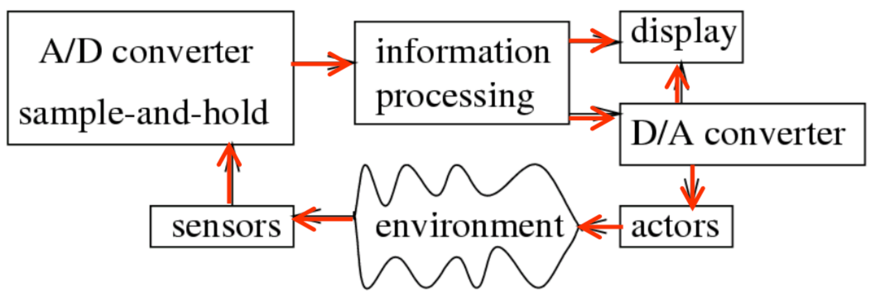
Sensors/Actuators, A/D & D/A and Control

Outline

- Sensors and actuators
- Analog-to-Digital converter
- Digital-to-Analog converter
- Digital signal measurement
- Device interfaces
  - Motors
  - LED
  - Keyboard
- Control

Embedded Systems w. Sensors/Actuators

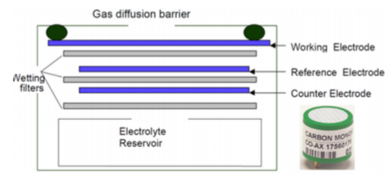


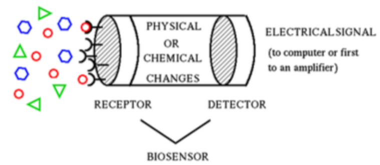
Sensors and Actuators

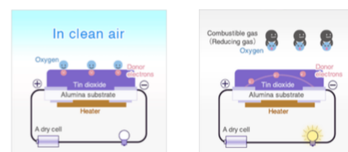
- Sensors
  - Perceive & capture physical stimulus
  - Convert to electrical signals
  - Analog sensors: generate a voltage/current
    - ✓E.g., ambient light
  - Digital: sensors generate a digital value;
    - ✓E.g., GPS
- Actuators
  - Create physical stimulus
  - E.g., Pneumatic systems, IR, thermal, motors MEMS
- ADC and DAC are needed

External Sensor Examples

- Electrochemical
  - Active electrode exposed to gas (or liquid)
  - Change in current → concentration of gas etc.
- Biosensor
  - Receptor (ligand) binds to target molecules → electrical/chemical charge
- MOS sensor
  - Clean air causes adsorption of donor electrons
  - Other gases → increase current

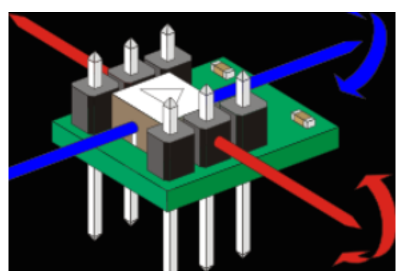






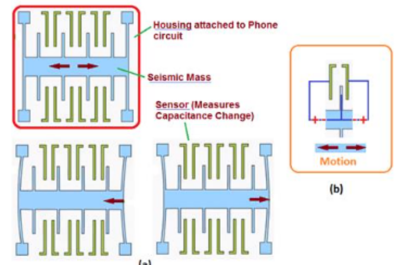
Example Sensors in Smartphone

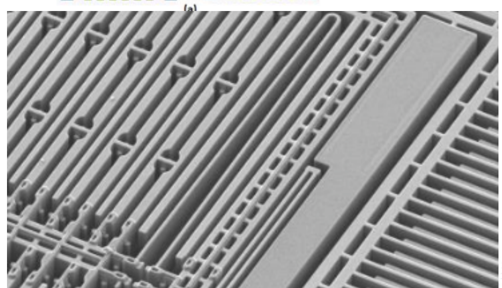
- Camera
- GPS, A-GPS
- Ambient light
- Proximity sensor
- Capacitive/resistive touch
- RFID/NFC
- Gyroscope: reference direction for navigation etc.
- Environment – air temperature, pressure, humidity



Onboard MEMS Accelerometer

- Microelectromechanical device
  - Motion trigger displacement of mass → change in capacitance
  - Sensor measures the change in capacitance
  - Multiple sensors in different orientations determine different axes

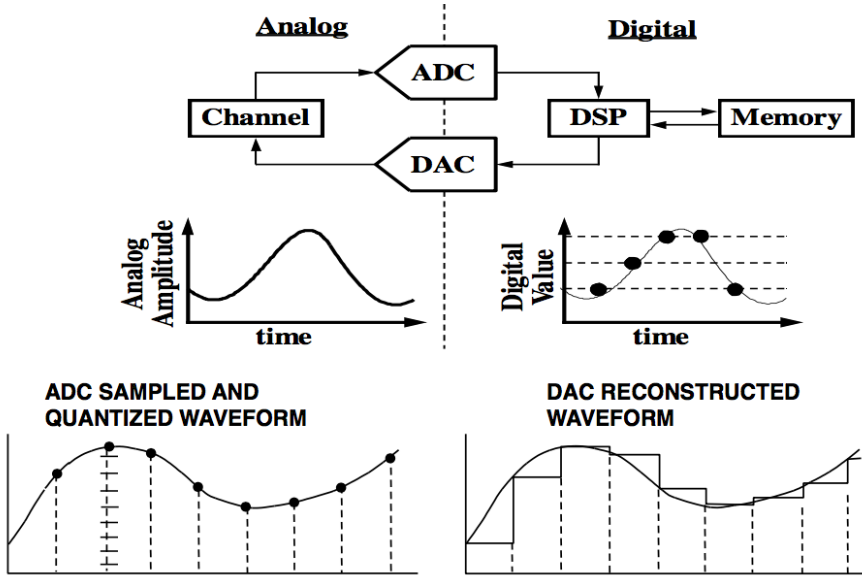




Issues in Sensors

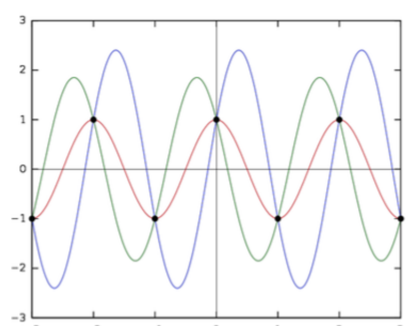
- Multi-property sensitivity (e.g., temperature)
- Processing overhead
- Drift
- Noise
- Power
- Accuracy
- Latency

Sampled Data w. ADC and DAC



Sensors: Quantization and Sampling

- Quantization: digital representation of analog value
  - Quantization error
  - Data format
  - Resolution
  - A/D conversion overhead
- Sampling: convert analog f(t) into time-series sequence
  - Discretization
  - Sampling interval
  - Nyquist rate/aliasing
  - Power/accuracy tradeoff



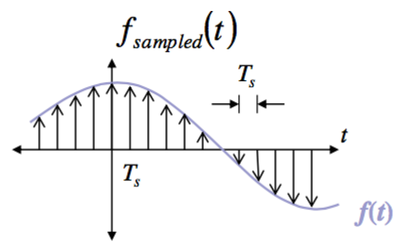
Sampling: Time Domain

- Signals normally originate in continuous-time
  - Such as talking/voice on cell phone; or music
- By ideally sampling a continuous-time signal at isolated, equally-spaced points in time, we obtain a sequence of numbers

  f[n] = f(n*T_s)

  n = { …, -2, -1, 0, 1, 2, …}

  T_s : is the sample period



Sampled analog waveform

$$f_{sampled}(t) = f(t) \underbrace{\sum_{n=-\infty}^{\infty} \delta(t - nT_s)}_{\textit{impulse train}}$$

CS 4833: Embedded Systems 11

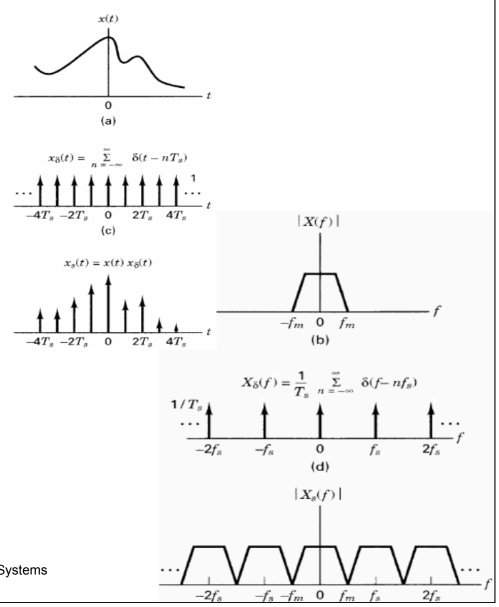
Signal Sampling

- Sampling converts a continuous time signal into a discrete time signal
  - It replicates spectrum of continuous-time signal at integer multiples of sampling frequency
- Categories
  - Impulse (ideal) sampling
  - Natural sampling
  - Sample and Hold operation

CS 4833: Embedded Systems 12

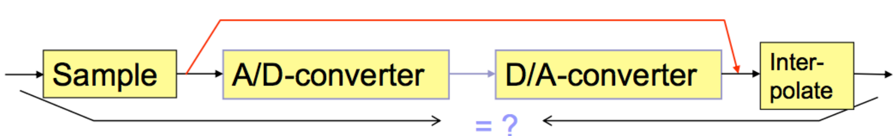
Impulse Sampling

- Impulse train spaced at T_s multiplies the signal x(t) in **time domain**, creating
  - Discrete time, continuous amplitude signal $x_s(t)$
- Impulse train spaced at f_s convolutes the signal X(f) in **frequency domain**, creating
  - Repeating spectrum $X_s(f)$, spaced at f_s



Nyquist Theorem

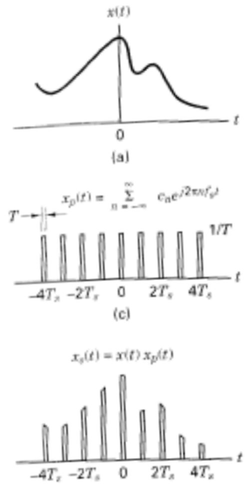
- Analog input can be precisely reconstructed from its output, provided that the sampling proceeds at >= double of the highest frequency found in the input [Nyquist 1928]



Does not capture effect of quantization:
Quantization noise prevents precise reconstruction.

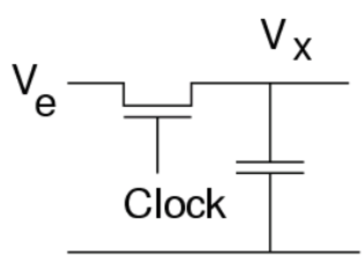
Natural Sampling

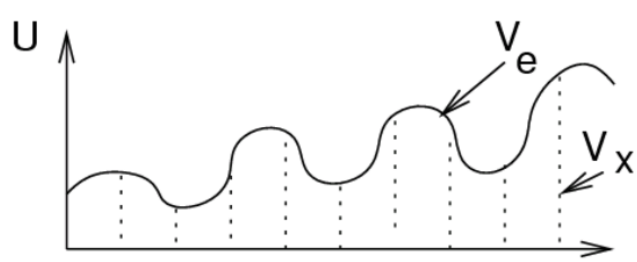
- Sampling pulse train has a finite width t
- Sampled spectrum will repeat with a 'Sinc' envelope
- More realist modeling
- Distortion after recovery depends on t/T_s



Sample and Hold

- V_e is analog input signal
- V_x is digital sampled signal





Different Sampling Models

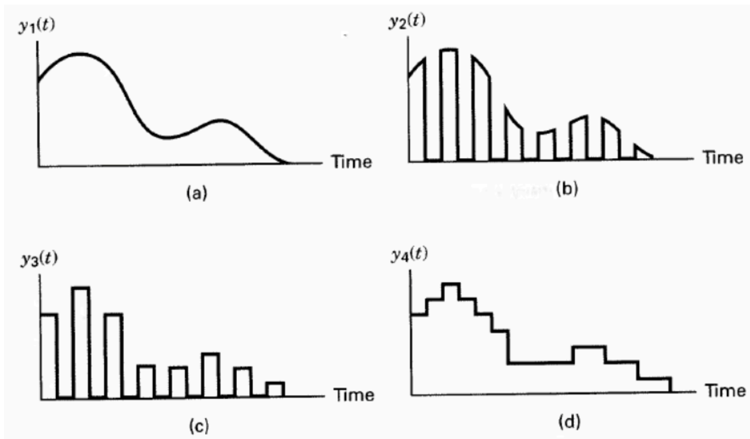
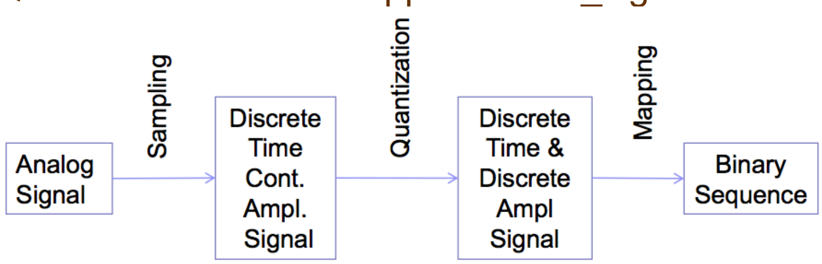
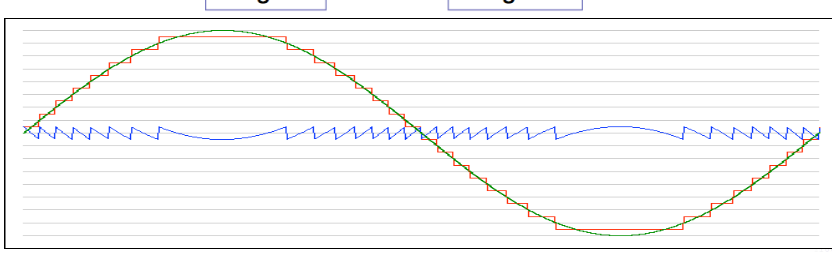


Figure 2.14 Amplitude and time coordinates of source data. (a) Original analog waveform. (b) Natural-sampled data. (c) Quantized samples. (d) Sample and hold.

Quantization and Noise

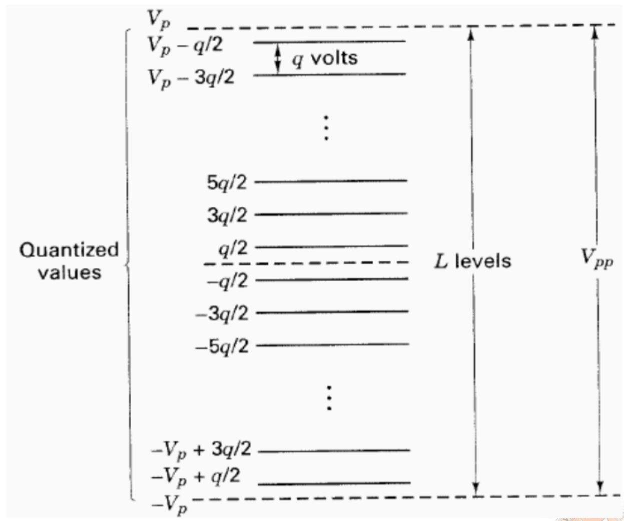
- The process of making the signal amplitude **discrete**
- Quantization **noise**: = approx – real_signal



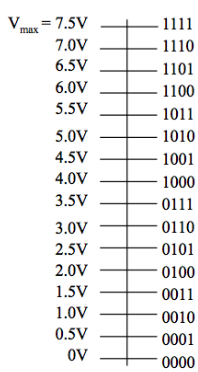


Linear Quantization

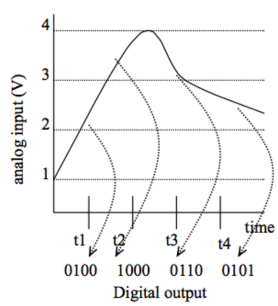
- L levels
  - $(L-1)*q = 2V_p = V_{pp} \rightarrow$ large L, $L*q = V_{pp}$



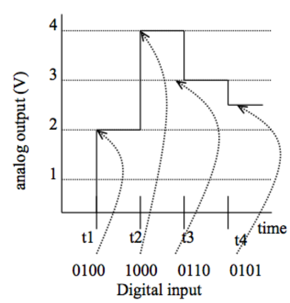
Linear Pulse Code Modulation



proportionality



analog to digital

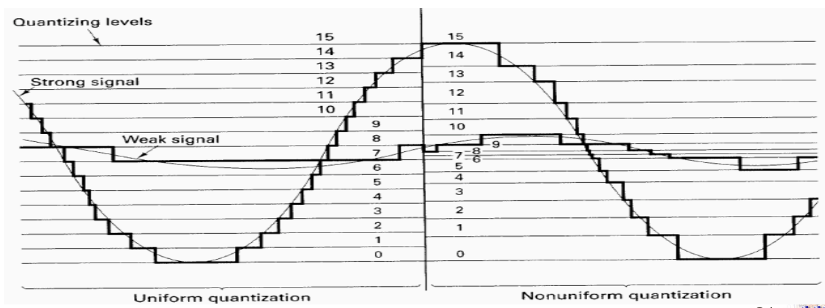


digital to analog

- Defined by sampling rate & bit depth L (total number of values to be represented)

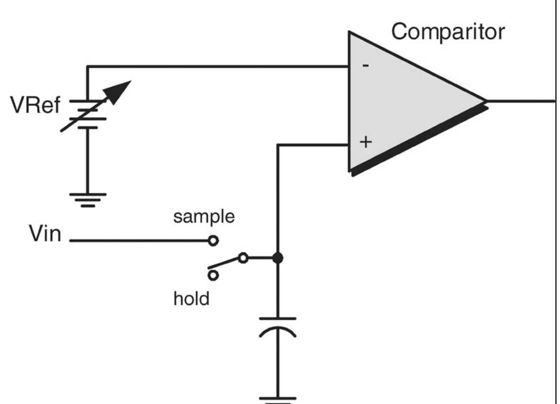
Non-Uniform Quantization

- Voice signals, low speech volumes predominate
  - ➢ Exceeds RMS (Root-mean-square) value → only 15%
- Low level signals are under represented with uniform quantization: same noise, but low signal power
  - ➢ Non-Uniform quantization



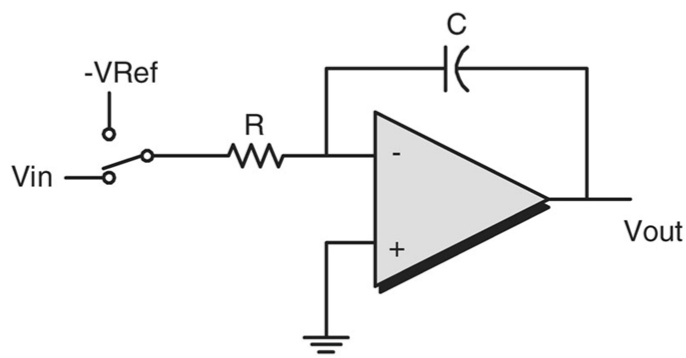
Analog to Digital Conversion

- Analog-to-digital conversion requires specialized circuits
- Most circuits based on analog **comparators**
- Dual slope A/D
- Successive approximation A/D



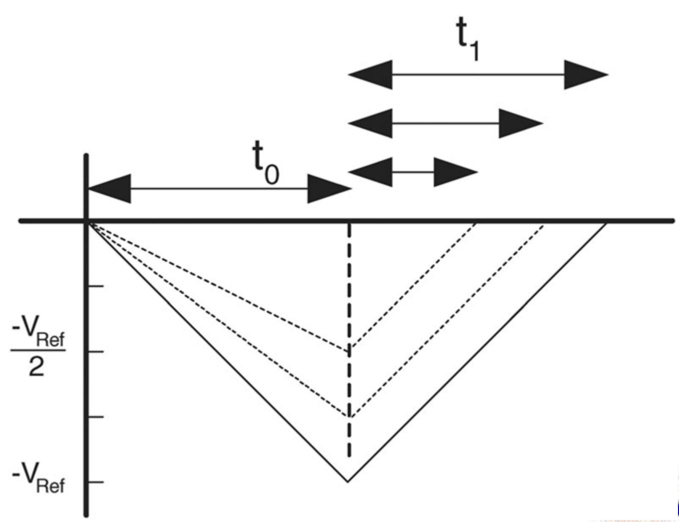
Dual-Slope A/D Conversion

- One reference input
- Integrator RC
- Measure time to charge/re-charge



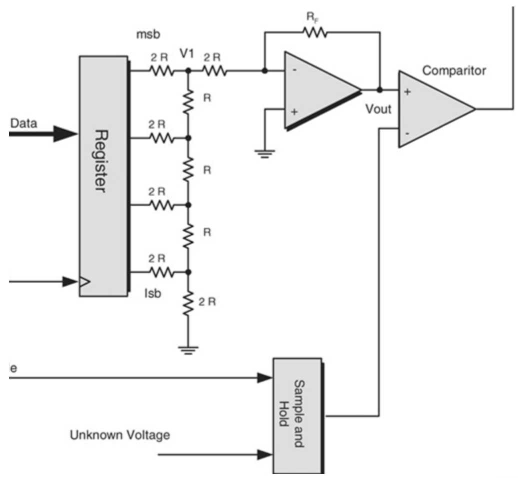
Dual-Slope A/D Conversion (cont.)

- Based on t0, t1 and Vref → input value Vin



Successive Approximation

- Register to store different values for different Vref
- **Binary** search to get desired values
- Comparitor charge/ re-charge time → speed



ADC with Successive Approximation

Given an analog input signal whose voltage should range from 0 to 15 volts, and an 8-bit digital encoding, calculate the correct encoding for 5 volts. Then trace the successive-approximation approach to find the correct encoding.

15 V → 255 (11111111); 5V → (????????)

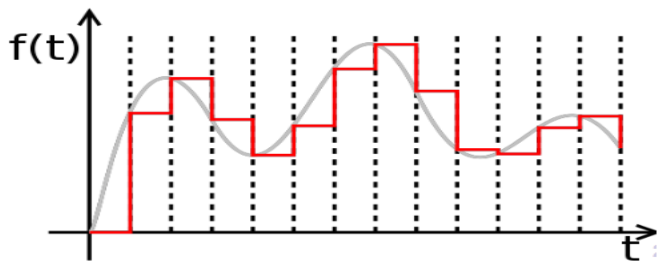
5/15 = d/255 → d = 85

Successive-approximation method

½($V_{max} – V_{min}$) = 7.5 volts V_{max} = 7.5 volts.	0 0 0 0 0 0 0 0
½(7.5 + 0) = 3.75 volts V_{min} = 3.75 volts.	0 1 0 0 0 0 0 0
½(7.5 + 3.75) = 5.63 volts V_{max} = 5.63 volts	0 1 0 0 0 0 0 0
½(5.63 + 3.75) = 4.69 volts V_{min} = 4.69 volts	0 1 0 1 0 0 0 0
½(5.63 + 4.69) = 5.16 volts V_{max} = 5.16 volts.	0 1 0 1 0 0 0 0
½(5.16 + 4.69) = 4.93 volts V_{min} = 4.93 volts.	0 1 0 1 0 1 0 0
½(5.16 + 4.93) = 5.05 volts V_{max} = 5.05 volts.	0 1 0 1 0 1 0 0
½(5.05 + 4.93) = 4.99 volts	0 1 0 1 0 1 0 1

Digital to Analog Converters

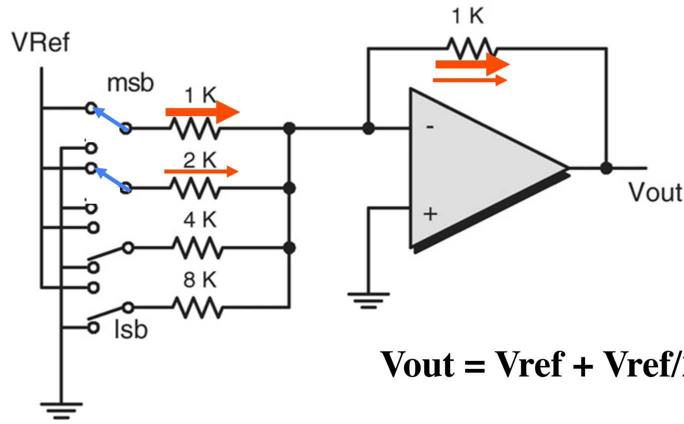
- Ideal sampling would allow to reconstruct the original signal perfectly with a sequence of impulses
- Use zero-order hold circuit to create analog output
  - Holding each sample value for one sample interval causes multiple harmonics above the Nyquist frequency



DAC: Key Parameters

- Resolution: n bit DAC gives 2^n levels
- Maximum sampling rate: defined by Nyquist theorem
- Dynamic range
  - difference of largest & smallest signals
- Monotonicity: move in the same direction
- Total harmonic distortion and noise (THD+N)
  - Measurement of distortion and noise: Percentage of total power of unwanted harmonic distortion & noises
- Example: 4-bit word
  - 0001 → 1mv ; 1111 → 15mv
  - 0001 → 1uv; 1111→15uv

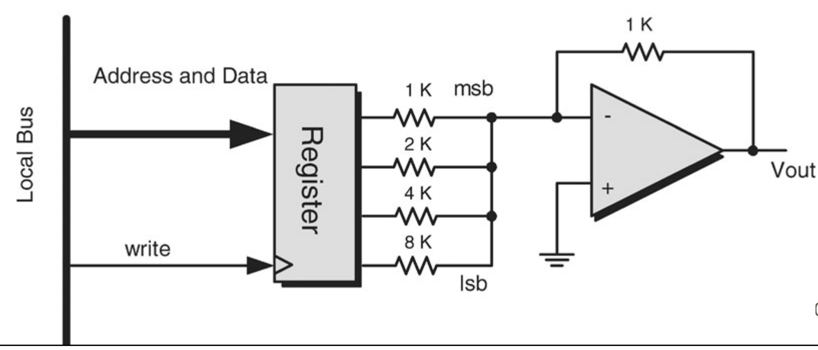
Example: Binary Weighted D/A Converter



Vout = Vref + Vref/2

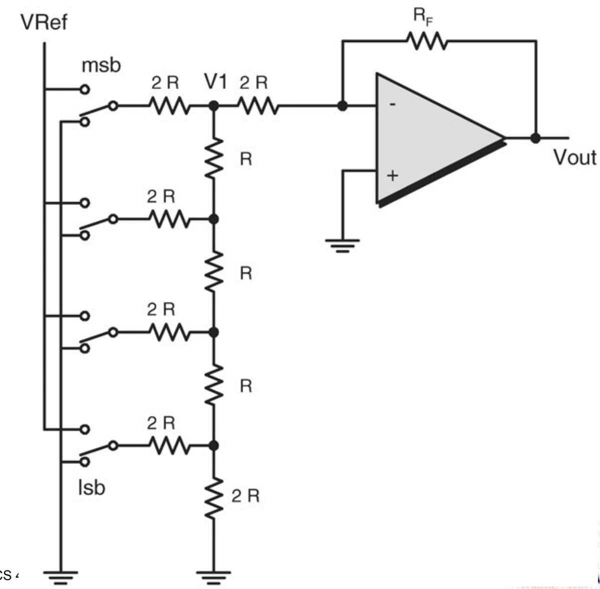
Interface to Microprocessor

- Issues to consider
  - CMOS register: input Vref = 4.95 - 5.0 volts; not precise
  - Use register output to select calibrated voltage: high accuracy and precision with a switch
  - Resistors accuracy



Better D/A: R/R2 Ladder

- Use only two different values of resistors
- Only the ratio matters: better precision
- Easy to expand



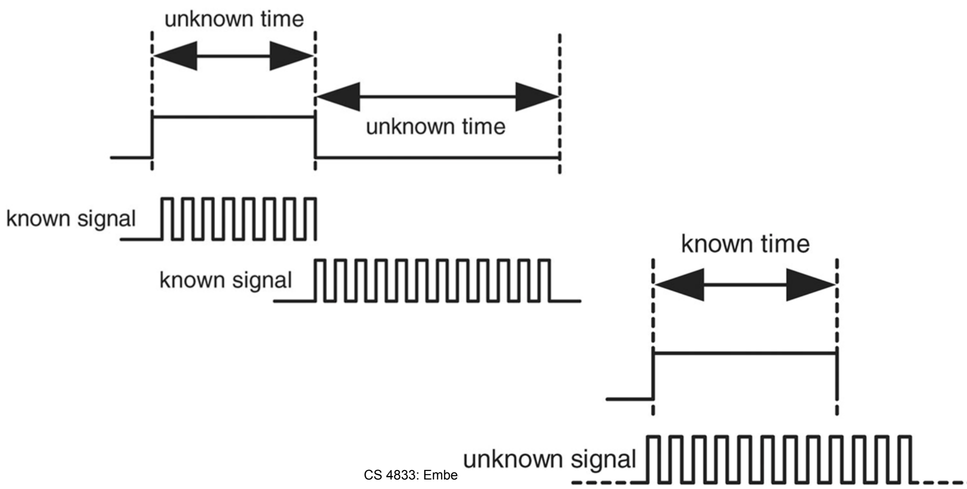
CS 4

Digital Signals: Other Measurements

- Time domain
  - Period of periodic signals
  - Duration of a signal
  - Elapsed time between events
- Frequency domain
  - Frequency of periodic signals
  - Number of events within an interval

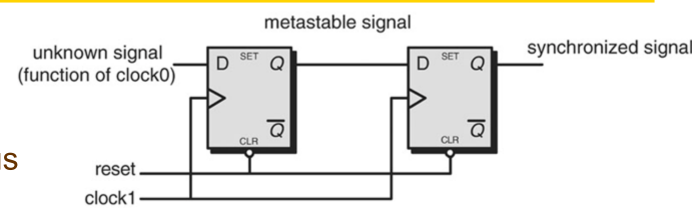
Basic Approach

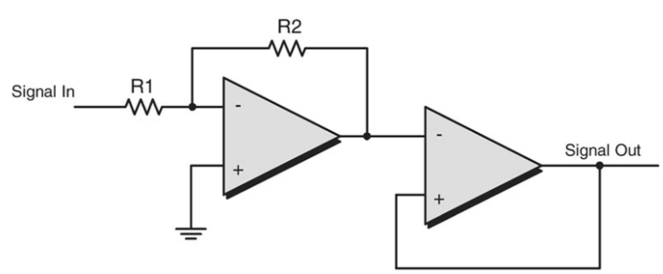
- Use **known** signals to measure **unknown** signals (time, period, interval)



Other Issues

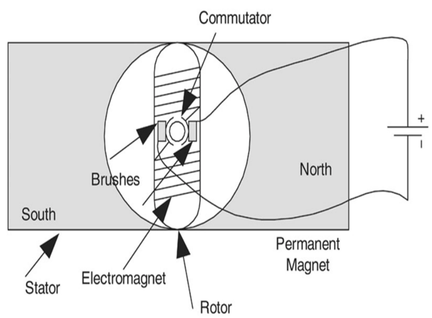
- Synchronize and Buffer input signals
  - Asynchronous signal vs. internal clock
  - **Different signal levels** (3, 5, 12 V)
- Internal vs. external implement
  - Timer/counter





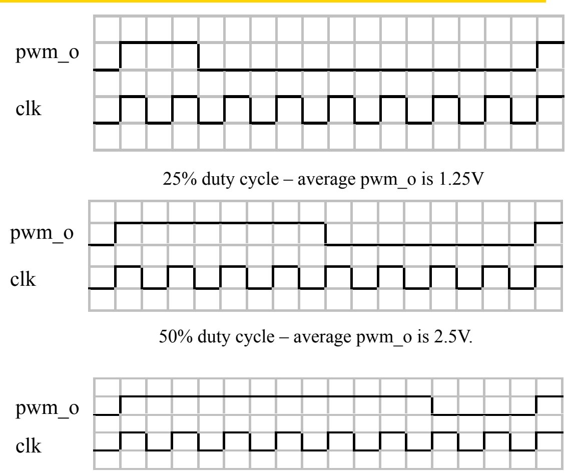
Motors and the Control

- DC motors
  - Can turn in 360 degree
  - Speed/power proportional to supply voltages
- Use PWM (pulse width modulation) to control speed



Pulse Width Modulator (PWM)

- Generates pulses with specific high/low times
- **Duty cycle**: % time high
  - Square wave: 50% duty cycle
- Common use: control average voltage to electric device
  - Simpler than DC-DC converter or digital-analog converter
  - DC motor speed, dimmer lights
- Another use: encode commands, receiver uses timer to decode



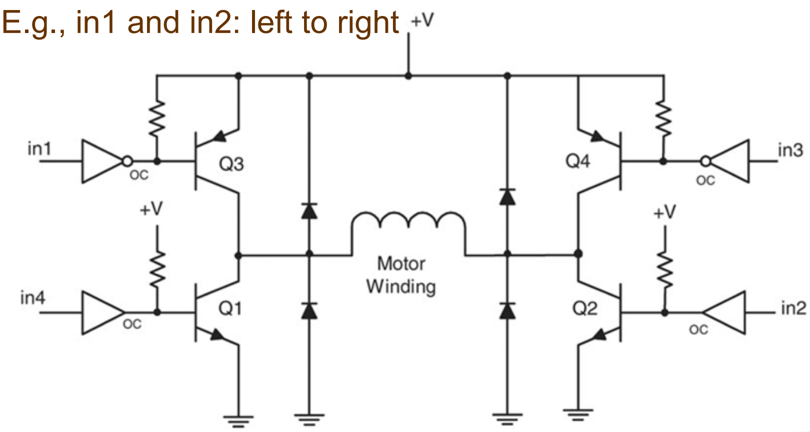
25% duty cycle – average pwm_o is 1.25V

50% duty cycle – average pwm_o is 2.5V.

75% duty cycle – average pwm_o is 3.75V.

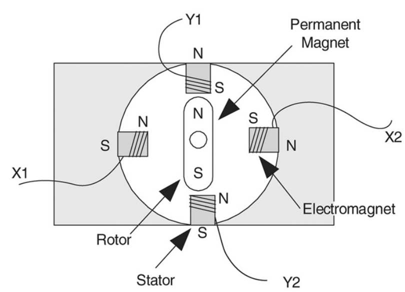
Motor Drive Circuitry

- High current requirements, not supported by CMOS or TTL gates
- H Bridge Motor Driver: bi-directional
  - E.g., in1 and in2: left to right



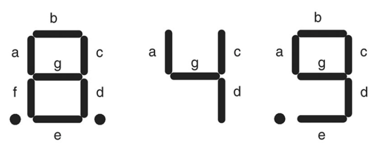
Other Motors

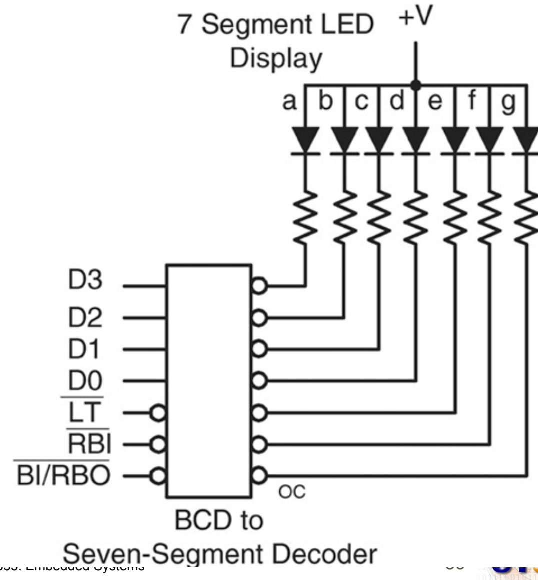
- Servo motors
  - Certain limited degree
  - Accurate positioning, close loop control
- Stepper motors
  - Fixed positions



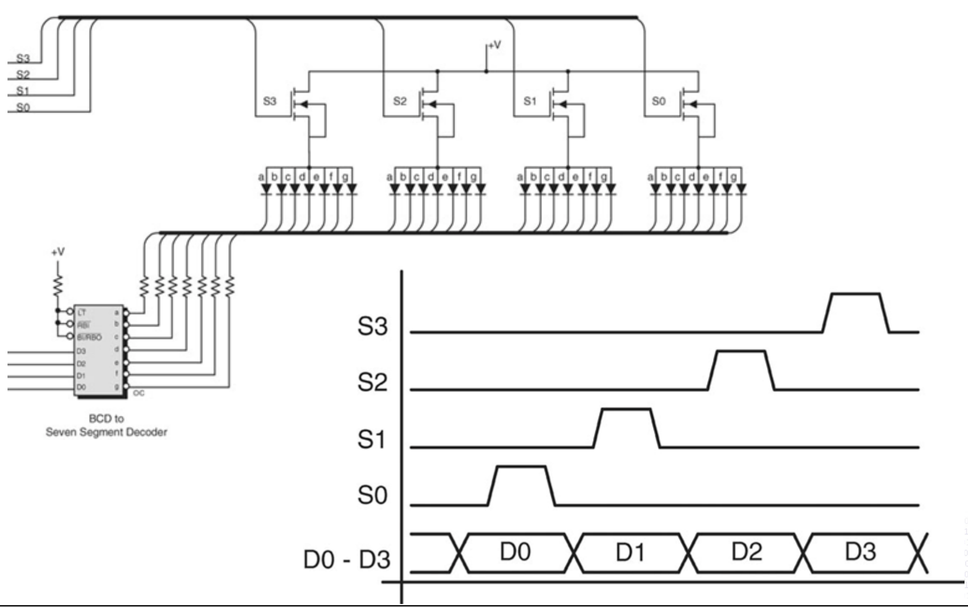
LED Display and Control

- LED: light emitting diodes
- Seven-Segment LED and decoder
  - Common **anode**
  - Common cathode

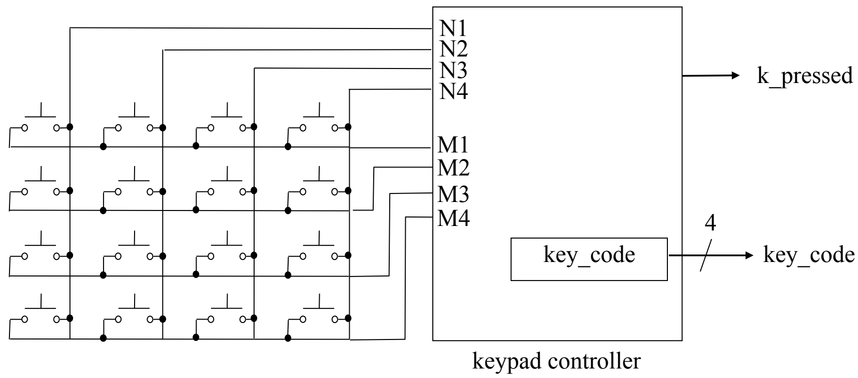




An Example: 4-Digital LED Display



Keypad Controller



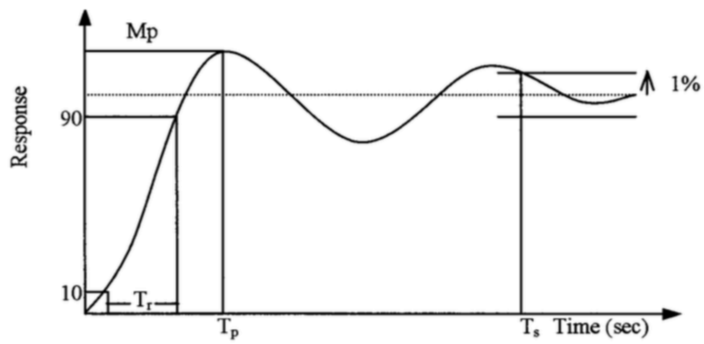
N=4, M=4

Control Basics

- Objective: tracks a reference even in presence of measurement noise, model error & disturbance
- Metrics
  - Stability : output remains bounded
  - Performance: how well an output tracks the reference
  - Disturbance rejection : tolerate outside error sources
  - Robustness : ability to tolerate modeling error of plant
- Software gives commands to meet a set-point, system responds trying get to the set-point
  - e.g., thermostat, car cruise control, aircraft altitude control

Control Performance

- Rise time: time taken from 10% to 90%
- Overshoot: percentage of peak over target
- Setting time: time to reach 1% of target

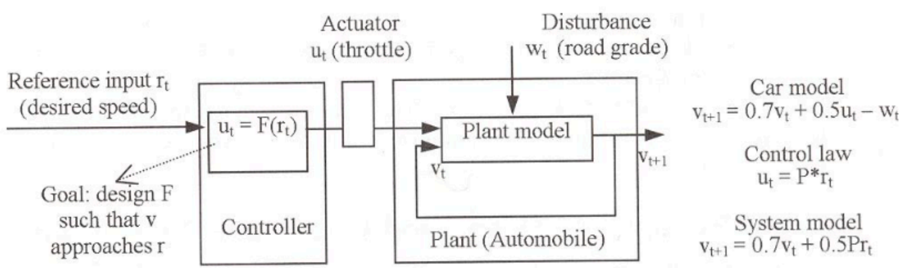


Open-Loop Control Systems

- Plant: physical system to be controlled
  - Car, airplane, heater etc.
- Reference/Setpoint: value of interest (target)
  - Desired speed, height, desired temperature
- Actuator: device to control the plant
  - Gas throttle, wing flap, furnace/AC
- Output: targeted aspect of the physical system
  - Speed, altitude, temperature etc.
- Controller: designed product to control the plant
- Disturbance: uncontrollable input from environment
  - Road slope, wind, opening door etc.

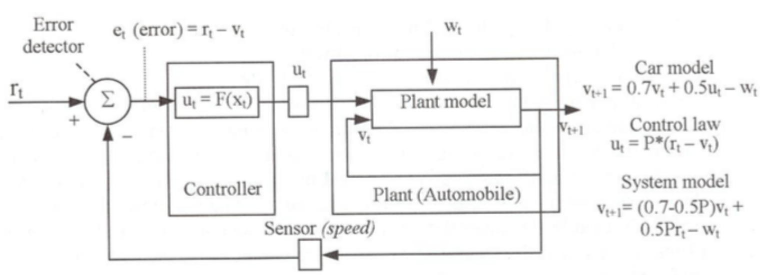
Example: Cruise Control in Cars

- Set target speed



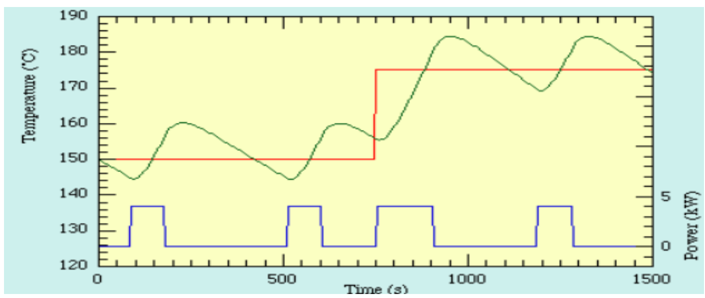
Closed-Loop Control Systems

- Sensor: measure the output
- Error detector: minimize tracking error
- Feedback control systems



Controller Design: On-Off

- If below a set point, turn-on the controlled actuator; otherwise, turn-off the controlled actuator
- Hysteresis H: difference between on and off values
  - Large enough to prevent noise from switching control rapidly and unnecessarily when near the set-point
- Thermostats: normally On-Off control



47

Proportional Control

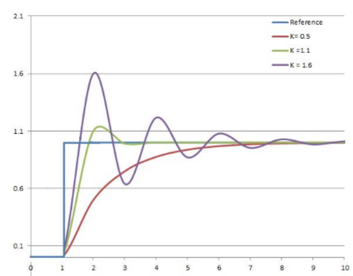
- Good alternative to on-off control
- Signal becomes proportional to the error
  - P (setpoint – output)
  - E.g., gas throttle level → cruise control for cars
- Need to find out value of constant P
  - Tuning the controller is hard
  - What if the values of P is too low? Or too high?
- Typically, large values of P decrease response time so it quickly gets to the setpoint, but can lead to increased overshoot.

CS 4833: Embedded Systems

48

P Controller Design

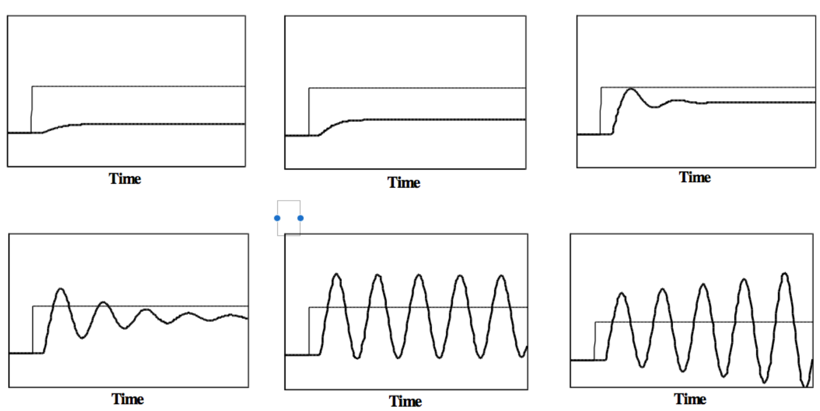
- Make a change to the output that is proportional to the current error
  - e: setpoint – current value
  - K_p: proportional gain
  - P_{out}: proportional term of output
- P controller often has a permanent offset from setpoint
  - Retains error depends on Kp and process gain
- System can become unstable when K_p is too large



$$P_{\text{out}} = K_p\, e(t)$$

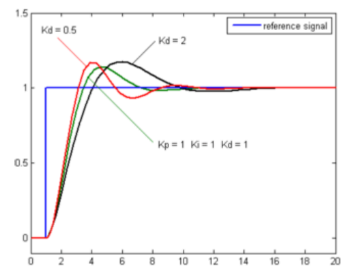
P-Only Control vs. K_p

- As Kp increases, an open loop process dynamics
  - overdamped, critically damped, oscillatory, ringing, sustained oscillations, & unstable oscillations



Adding Derivative Control

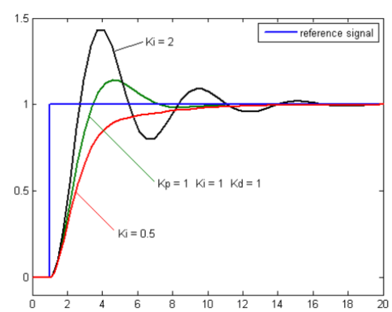
- To reduce overshoot/ripple: take into account how fast approaching the setpoint
  - If too fast, may overshoot → reduce signal by P-controller
  - If too slow, may not work → increase signal
  - Change is proportional to the **derivative** of the error
  - Large K_d decreases overshoot, but amplifies the noise
- Proportional-Derivative controller
  - Slower than P-controller
  - Smaller overshoot/ripple
  - Less oscillation



$$D_{\text{out}} = K_d \frac{de}{dt}$$

Integral Control

- Output is close to setpoint → small error
  - Proportional: very small, discretization provides no change
  - Derivative: no change as well since output is close
- Integral control
  - proportional to both magnitude & duration of error
  - Take sum of errors over time: small error adds up
  - Larger Ki eliminates steady state errors, but can cause overshoot



$$I_{out} = K_i \int_0^t e(\tau)\, d\tau$$

Controller Design: PID

- Combine Proportional, Integral and Derivative control to change manipulated variable (MV)
  - Use P to control the amount of disturbance (error)
  - Use I to ensure steady state convergence & rate
  - Use D to control the speed of error reduction
- No guarantee on optimality, stability, & not adaptive

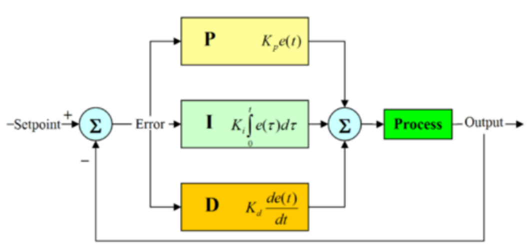
$$\mathrm{MV(t)} = P_{\mathrm{out}} + I_{\mathrm{out}} + D_{\mathrm{out}}$$

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

Controller Design: PID

- Effects of increasing parameters independently

Parameter	Rise time	Overshoot	Settling time	Steady-state error	Stability
K_p	Decrease	Increase	Small change	Decrease	Degrade
K_i	Decrease	Increase	Increase	Eliminate	Degrade
K_d	Minor change	Decrease	Decrease	No effect in theory	Improve if small



Summary

- Sensors and actuators
- Analog-to-Digital converter
- Digital-to-Analog converter
- Digital signal measurement
- Device interfaces
  - Motors
  - LED
  - Keyboard
- Control