929 SCE

Signal Conditioning In The Embedded World

Class Objectives

When you walk out, you will know

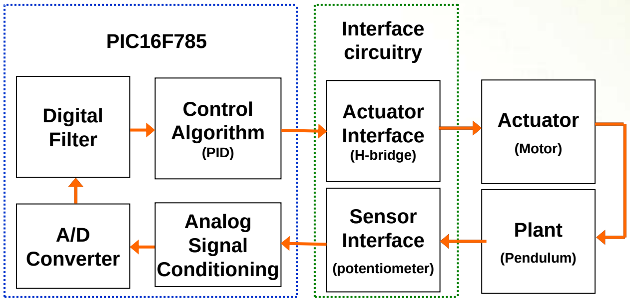
- What makes up the embedded signal chain?
- How do you get the information you need from your signal?
- How can this be applied to a control loop?

Class Agenda

- The Embedded Signal Chain
  - Sensor Interfacing
  - Analog Signal Conditioning
  - Analog to Digital (A/D) Conversion
  - Digital Filtering
  - Control Algorithms
  - Actuator Interfacing

The Embedded Signal Chain

Control Loop for the Inverted Pendulum



PIC16F785

Digital Filter

Control Algorithm (PID)

A/D Converter

Analog Signal Conditioning

Interface circuitry

Actuator Interface (H-bridge)

Sensor Interface (potentiometer)

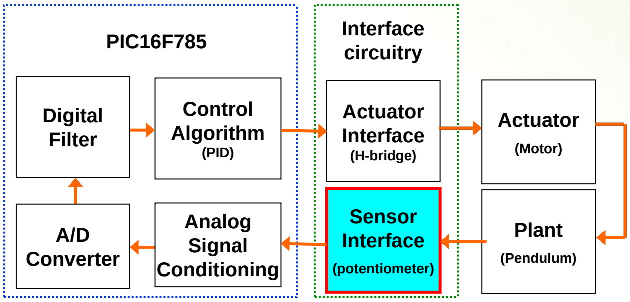
Actuator (Motor)

Plant (Pendulum)

The Embedded Signal Chain

The Embedded Signal Chain

Sensor Interfacing



PIC16F785

Digital Filter

Control Algorithm (PID)

A/D Converter

Analog Signal Conditioning

Interface circuitry

Actuator Interface (H-bridge)

Sensor Interface (potentiometer)

Actuator (Motor)

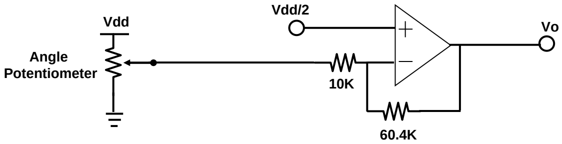
Plant (Pendulum)

Sensor Interfacing

- Sensors generate voltage, resistance, or current
- Must be translated into a variable voltage for analog to digital conversion
- AN990 -Analog Sensor Conditioning Circuits

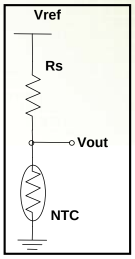
Sensor Interfacing

- A potentiometer provides mechanical feedback, e.g. angle, position



Sensor Interfacing

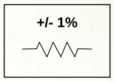
- Interfacing the thermistor with a voltage divider
  - Voltage divider configuration helps with linearity
  - Rs should be equal to thermistor at mid-point of temperature range
  - For ± 25 ^{o}C temperature range ± 1% error
- AN685 - Thermistors in Single Supply Temperature Sensing Circuits



Sensor Interfacing

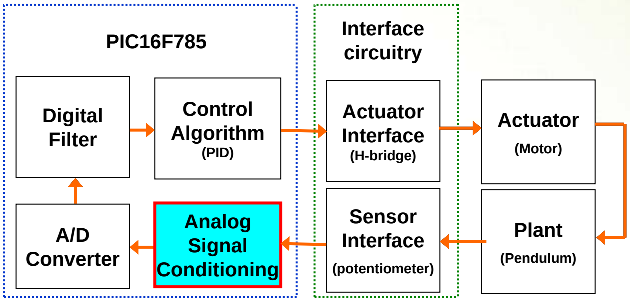
Possible error sources

- Component tolerances
- Electrical Fast Transient (EFT) signals (e.g. high speed digital, high current/power drive circuits)
- Electrostatic Discharge (ESD)



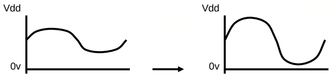
The Embedded Signal Chain

Analog Signal Conditioning

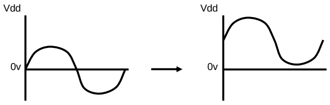


Analog Signal Conditioning

- Why do you need analog signal conditioning?
  - Filter out high frequency noise (e.g. anti-aliasing)
  - Add gain to increase signal resolution



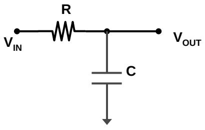
  - Level shifting

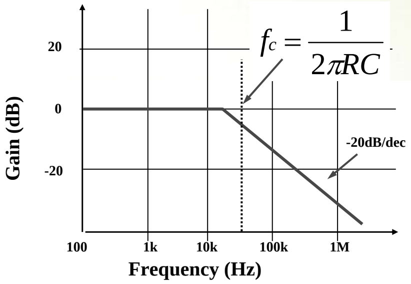


Analog Signal Conditioning

- A single order passive RC filter attenuates by 20 dB/decade beyond f_c or the 3 dB cutoff frequency

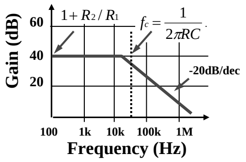
$$\frac{V_{out}}{V_{in}} = \frac{1}{1+sRC}$$



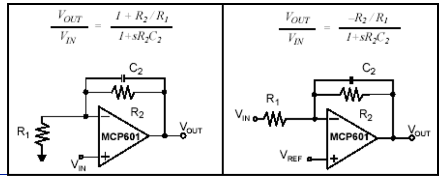


Active Filtering

- Single pole active filters have same frequency response with lower output impedance
- Can gain a signal



Non-inverting or Inverting



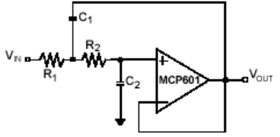
$$\frac{V_{OUT}}{V_{IN}} = \frac{1+R_2/R_1}{1+sR_2C_2}$$

$$\frac{V_{OUT}}{V_{IN}} = \frac{-R_2/R_1}{1+sR_2C_2}$$

Sallen Key Second Order Filter

- High input impedance, non-inverting, unity gain
- Implement Bessel, Chebyshev, Butterworth, Elliptical or other filter types
- Implement lowpass, highpass, bandpass, band stop

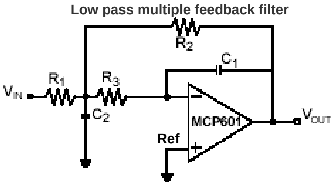
Low pass Sallen Key filter



Multiple Feedback Active Filter

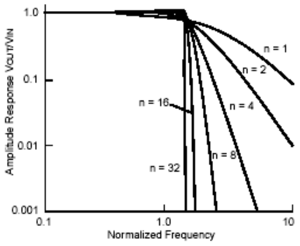
- Lower input impedance than Sallen Key
- Inverting with gain or attenuation of $G = -R_2/R_1$
- Less sensitive than Sallen Key to component tolerances
- Pay attention to reference voltage with single supply op-amps

Low pass multiple feedback filter



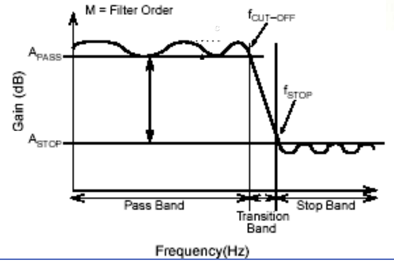
Filter Response

- The higher the order the steeper the response
- Use multiple second order stages to get required results



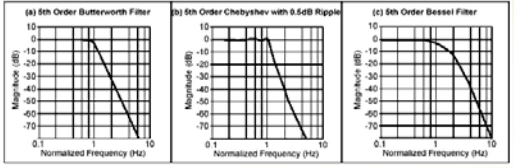
Active Filter Design

- Use a design program such as Filterlab from Microchip
- Specify key filter performance factors
- The op-amp’s open loop bandwidth should be at least 100 times the filters bandwidth at the cutoff frequency



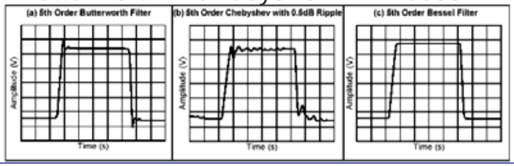
Frequency Response

Butterworth Chebyshev Bessel



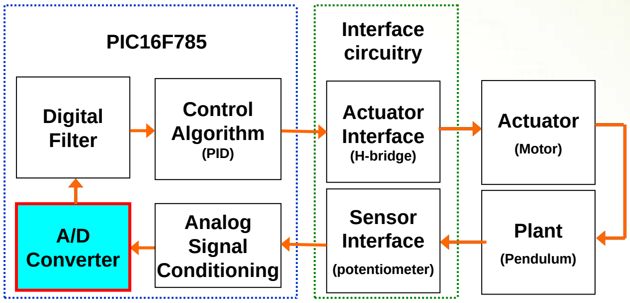
Step Response

Butterworth Chebyshev Bessel



The Embedded Signal Chain

Analog to Digital Conversion



PIC16F785

Digital Filter

Control Algorithm (PID)

A/D Converter

Analog Signal Conditioning

Interface circuitry

Actuator Interface (H-bridge)

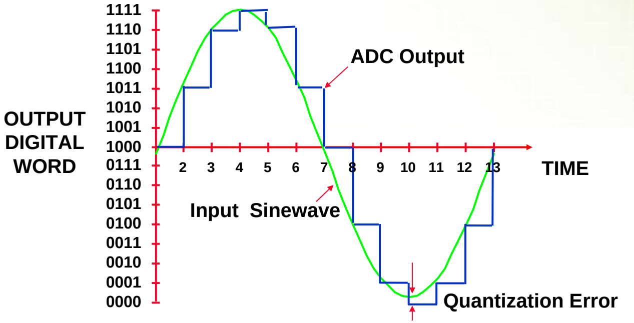
Sensor Interface (potentiometer)

Actuator (Motor)

Plant (Pendulum)

A/D conversion

- Sampling and Quantization



4-bit (16 level) ADC sampling a sinewave input, time domain

A/D conversion

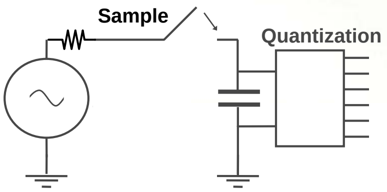
- Ideal signal to noise ratio (SNR) of A/D in dB

$$SNR = 6.02 \cdot N + 1.76 \quad [dB]$$

  - N = A/D resolution in bits
  - Ideal SNR is 62 dB for 10 bit A/D

A/D conversion

- Output impedance



The Alias Effect

- Satisfy Nyquist Sampling Theorem to retain signal information

  $$Fs \geq 2 \cdot Fin$$

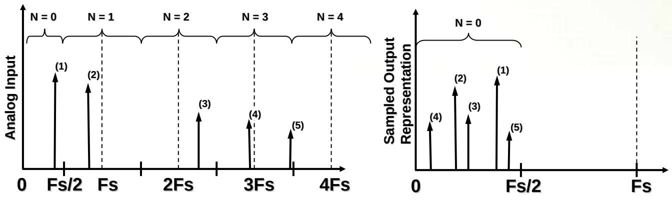
  - Fs - Sample frequency
  - Fin - Input signal frequency
- Get an image Fi if Nyquist conditions are not met

  $$Fi = \left| n \cdot Fs \pm Fin \right|$$

  - n - integer

The Alias Effect

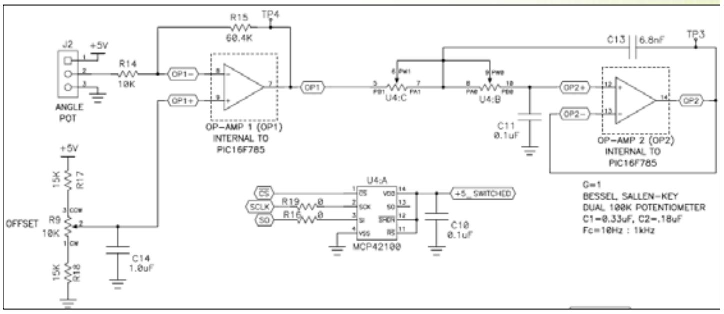
- Input signals above Fs/2 will be folded back (aliased) down to a lower frequency



- Use analog filter with Fc < Fs/2 before A/D to prevent aliasing

---

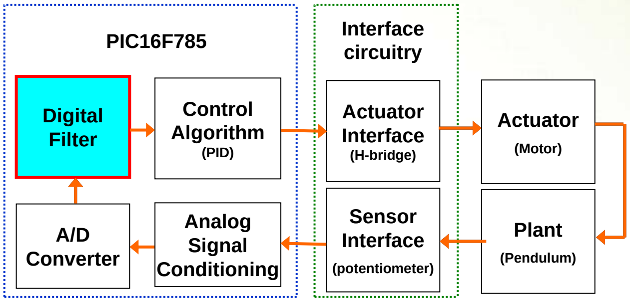
The Alias Effect in action



- Filter:
  - Sallen-Key, Bessel, G=6, Fc = 30 Hz
- What happens without proper anti-aliasing?

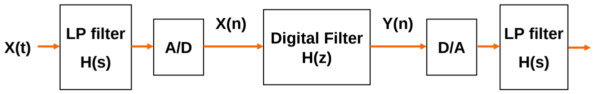
The Embedded Signal Chain

Digital Filtering



Digital Filtering

- Finite Impulse Response (FIR) filters do not have feedback
- Infinite Impulse Response (IIR) filters have feedback
- Efficient digitals filters

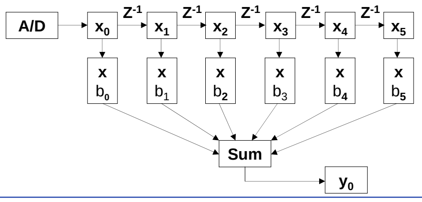


FIR Filter

- Difference equation

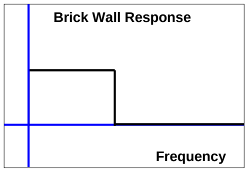
$$y(n) = b_0 x(n) + b_1 x(n-1) + \ldots\ldots + b_m x(n-m)$$

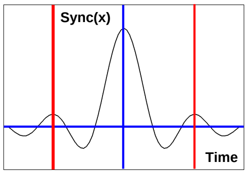
- With order of "m"



Designing a FIR Filter

- Filter coefficients are based on a given window function H[n]
- Different window functions have different performance and filter length requirements





Designing a FIR Filter

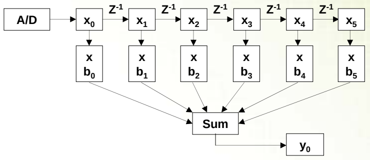
- FIR filter has a constant group delay
- Processor intensive with respect to filter length (order)
- Use a filter coefficient program such as dsPICworks or dsPIC FD Lite from Microchip

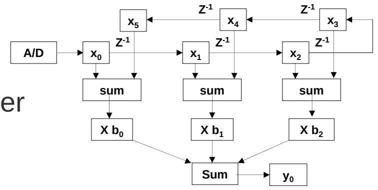
---

Folded FIR Filter

- If coefficients symmetrical

(b_0=b_5 b_1=b_4 b_2=b_3)





- Use folded FIR filter to reduce multiply instructions

Moving Average Filters

- Simply taking the moving average of a number of samples, the filter has M points

$$y(n) = \frac{1}{M}\sum_{j=0}^{M-1} x(n-j)$$

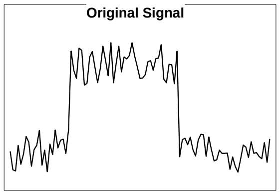
- Very good at removing switching noise and random noise
- Is very fast especially if M is a power of 2, i.e. 2, 4, 8, 16 etc → allows one to shift instead of divide

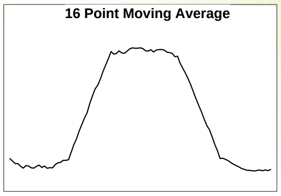
---

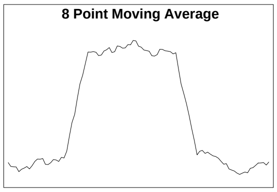
Recursive Implementation

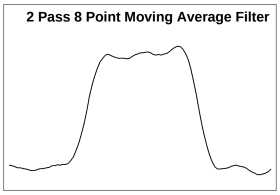
- Two adjacent outputs of a moving average filter
  - y(50)=x(50)+x(49)+x(48)+ x(47)+x(46)+x(45)
  - y(51)=x(51)+x(50)+x(49)+x(48)+ x(47)+x(46)
  - => y(51)=y(50)+x(51)-x(45)
  - General case: y(n)=y(n-1)+x(n)-x(n-M)

Moving Average Filter Examples


Original Signal


16 Point Moving Average


8 Point Moving Average

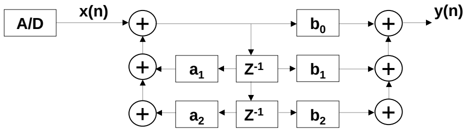

2 Pass 8 Point Moving Average Filter

IIR Filter

- A second order IIR filter uses feedback and it's difference equation is

$$y(n) = b_0x(n) + b_1x(n-1) + b_2x(n-2) + a_1y(n-1) + a_2y(n-2)$$

- The coefficient b_0 gives the DC gain, typically unity
- Use multiple stages to get desired order



IIR Filter Characteristics

- Normally more efficient than FIR
- Can implement Chebyshev, Butterworth, and Elliptic

IIR Filter Design

- Use a filter coefficient program such as dsPICworks or dsPIC FD Lite from Microchip
- High Q stages can lead to instability due to rounding

Efficient Filters

- Keep bits in the multiplier functions to a minimum and implement own multiplier
- Look into different numbering schemes:
  - Fixed Point signed/unsigned
  - Floating point
  - Two's complement
  - Fractional two's complement or Qx
- Double check filter function for saturation and stability - IIR especially

---

Efficient Filters

- Can speed up FIR and IIR filters by rounding off the filter coefficients
  - Round by inspection to allow shift and add instead of multiply
- Use two or three shift and add routines instead of a multiply routine
  - Need to multiply x(n) by 10010101 (dec 149)
  - 10010000 (dec 144) => error = -5 ✓
  - 10100000 (dec 160) => error = +11 x
  - x(n)*10010101 ≈ x(n)<<7 + x(n)<<4 = x(n)*2^7+x(n)*2^4
- To multiply y(n) by 00001011(dec 11)
  - 00001100 (dec 12) => error = +1
  - 00001010 (dec 10) => error = -1
  - In general round down for y(n)

Efficient Filters

- Consider convergent (unbiased) rounding instead of conventional or biased rounding - IIR especially
- Use pointers instead of shifting data
- Will a less expensive microcontroller do the job?

Efficient Filters

- Using unsigned filter coefficients is sometimes beneficial
- For scaled filter coefficients between -128 and 127 convert these to 0 to 255
  - Start with difference equation:

$$\boxed{y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) + a_1 y(n-1) + a_2 y(n-2)}$$

$$\boxed{\begin{aligned} y(n)^1 = {} & (b_0 + 128)x(n) + (b_1 + 128)x(n-1) + (b_2 + 128)x(n-2) \\ & + (a_1 + 128)y(n-1) + (a_2 + 128)y(n-2) \end{aligned}}$$

$$\boxed{X(n) = x(n)\cdot 128 + x(n-1)\cdot 128 + x(n-2)\cdot 128}$$

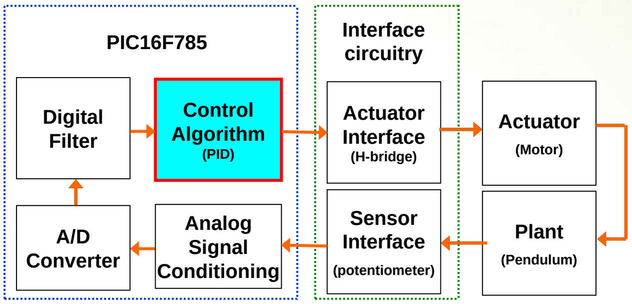
$$\boxed{y(n) = y^1(n) - X(n)}$$

Which to Use, Analog or Digital?

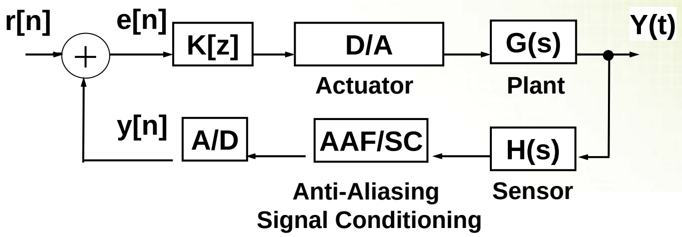
- Digital Filters
  - Highly repeatable and stable
  - Adaptable
  - Always needs power
  - Quantization and rounding noise and errors
- Analog Filters
  - Component variations affects performance
  - Fixed characteristics
  - Can be passive
  - Noise susceptible from conducted, radiated and thermal

The Embedded Signal Chain

Control Algorithm



Microchip in Control



r - reference signal
e[n] - error signal
K[z] - controller transfer function (PID)
G(s) - plant transfer function
H(s) - sensor transfer function
y(t) - output of controlled system

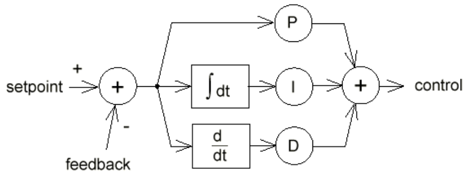
Microchip in Control

- Use a Proportional-Integral-Derivative (PID) loop to control a system
  - Reduce overshoot
  - Improve stability
  - Improved Performance

PID Loop

- The PID controller has a transfer function
  - K_P - proportional gain, K_I - integral gain, K_D - derivative gain, e[z] - error input signal to PID loop and u[z] - PID control signal

$$K[z]=\frac{u[z]}{e[z]}=K_P+\frac{K_I}{1-z^{-1}}+K_D(1-z^{-1})$$

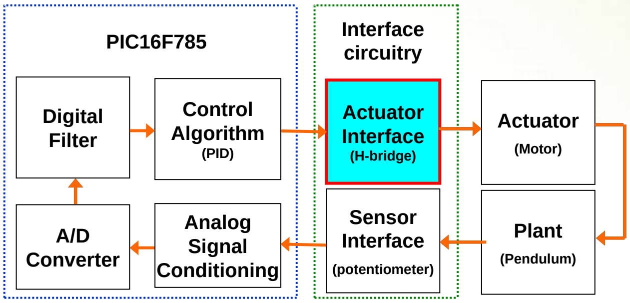


PID Loop

- Tuning the PID parameters
  - Demo
- How does each PID constant affect the System’s stability? Overshoot? Response time?

The Embedded Signal Chain

Actuator Interfacing


PIC16F785

Interface circuitry

Digital Filter

Control Algorithm (PID)

Actuator Interface (H-bridge)

Actuator (Motor)

A/D Converter

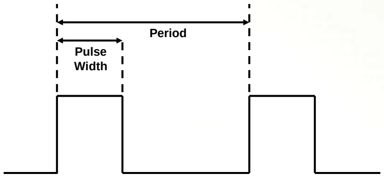
Analog Signal Conditioning

Sensor Interface (potentiometer)

Plant (Pendulum)

Actuator Interfacing

- Use of PWM for motor control

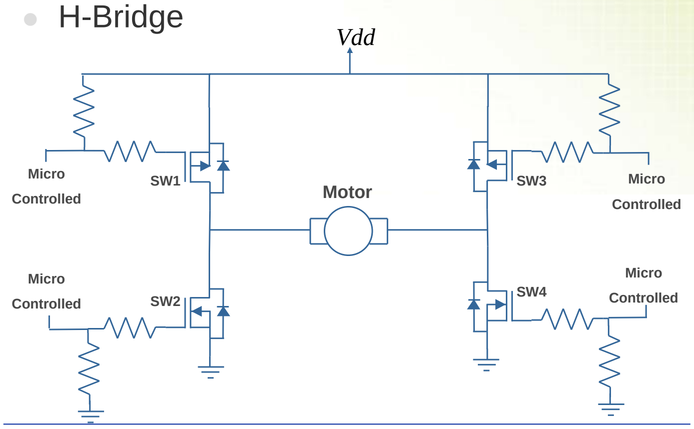


$$Duty\_Cycle = \frac{Pulse\_Width}{Period}$$

$$Motor\_Voltage = Vdd * Duty\_Cycle$$

Actuator Interfacing

- H-Bridge



Summary

- The Embedded Signal Chain
  - Sensor Interfacing
  - Analog Signal Conditioning
  - A/D Conversion
  - Digital Filtering
  - PID Control
  - Actuator Interfacing
- Identify possible performance improvement techniques to get the most out of your signal
- PID controllers can be very powerful tools towards improving embedded control

---

References

- Analysis and Design of Feedback Control Systems, Armando A. Rodriguez, CONTROL3D
- AN699 - Anti-Aliasing Filters for Data Acquisition Systems, Bonnie C Baker
- AN693 - Understanding A/D Converter Performance Specifications, Steven Bowling
- Active Filter Cookbook, Second Edition, Don Lancaster, Newnes
- The Scientist and Engineers Guide to Digital Signal Processing, Steven W Smith, WWW.DSPguide.com
- AN617 - Fixed Point Routines, Frank Testa
- AN526 - PIC16C5X / PIC16CXXX Math Utility Routines, Amar Palacherla
- AN575 - IEEE 754 Compliant Floating Point Routines, Frank Testa
- AN538 - Using PWM to Generate Analog Output, Amar Palacheria

---

References

- AN852 - Implementing FIR and IIR Digital Filters Using PIC18 Microcontrollers, Ananthma Ramu
- AN616 - Digital Signal Processing with the PIC16C74, Darius Mostowfi
- AN823 - Analog Design in a Digital World Using Mixed Signal Controllers, Keith Curtis
- AN685 - Thermistors in Single Supply Temperature Sensing Circuits, Bonnie C Baker
- AN251 - Bridge Sensing with the MCP6S2X PGAs, Bonnie C Baker
- AN990 - Analog Sensor Conditioning Circuits - An overview, Kumen Blake
- AN964 - Software PID Control of an Inverted Pendulum Using the PIC16F684, John Charais/Ruan Lourens
- Discrete-Time Signal Processing, Second Edition, Alan V Oppenheim, Prentice Hall