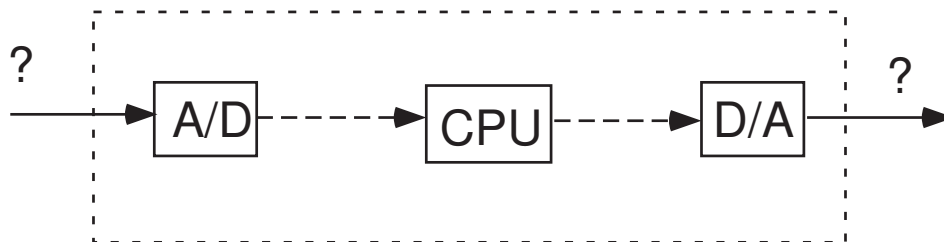


# Interfacing a Microprocessor to the Analog World

In many systems, the embedded processor must interface to the non-digital, analog world.

The issues involved in such interfacing are complex, and go well beyond simple A/D and D/A conversion.



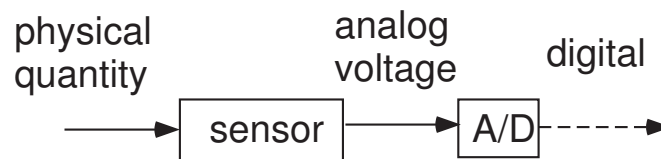
Two questions:

1. How do we represent information about the analog world in a digital microprocessor?
2. How do we use a microprocessor to act on the analog world?

We shall explore each of these questions in detail, both conceptually in the lectures, and practically in the laboratory exercises.

# Sensors

- Used to measure physical quantities such as
  - position
  - velocity
  - temperature
  - sound
  - light
- Two basic types:
  - sensors that measure an (analog) physical quantity and generate an analog signal, such as a voltage or current



- \* tachometer
- \* potentiometer
- sensors that directly generate a digital value



- \* digital camera
- \* position encoders

# Sensor Interfacing Issues

- Shall focus on issues that involve
  - *loss* of information
  - *distortion* of information
- Such issues include
  - quantization
  - sampling
  - noise
- Fundamental difference between quantization and sampling errors:
  - *Quantization errors* affect the precision with which we can represent a *single analog value* in digital form.
  - *Sampling errors* affect how well we can represent an *entire analog waveform* (or time function) digitally.

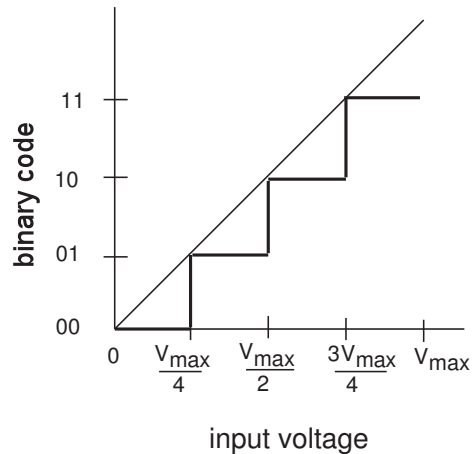
# Quantization

Digital representation of an analog number [2, 3, 6]

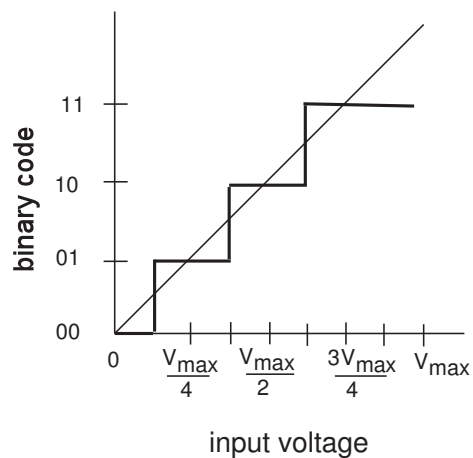
- Issue:
  - an analog voltage can take a continuum of values
  - a binary number can take only finitely many values
- Binary representation of (unsigned or signed) real number
  - unipolar coding
  - unipolar coding with centering
  - offset binary coding
  - two's complement
- Resolution [2, 3]
  - Idea: two analog numbers whose values differ by  $< 1/2^n$  may yield the same digital representation
  - an  $n$ -bit A/D converter has a resolution equal to  $2^{-n}$  times the input voltage range,  $v \in [0, V_{max}]$
  - least significant bit (LSB) represents  $V_{max}/2^n$

## Quantization: Example

- Suppose we quantize an analog voltage in the range  $(0, V_{\max})$  using a two bit binary number.
- The LSB thus represents  $V_{\max}/4$

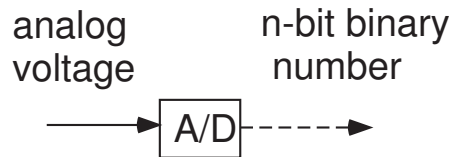


- Quantization error: from 0 to 1 LSB (e.g., 01 represents any voltage from  $V_{\max}/4$  to  $V_{\max}/2$ )
- For 11 to uniquely represent  $V_{\max}$ , divide voltage range into  $2^n - 1$  intervals.  $\text{LSB} = V_{\max}/(2^n - 1)$
- Centering: 01 represents  $V_{\max}/8$  to  $3V_{\max}/8$



- Quantization error:  $\pm \frac{1}{2} \text{LSB}$

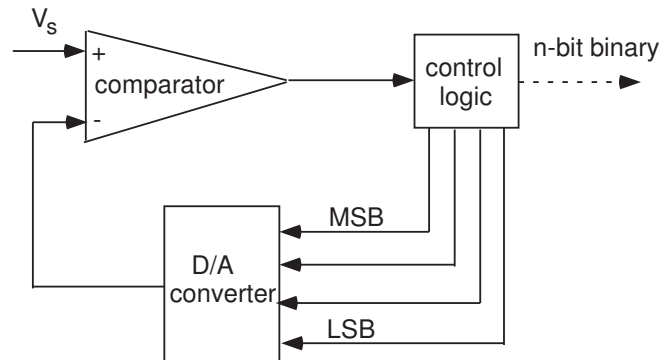
# A/D Conversion



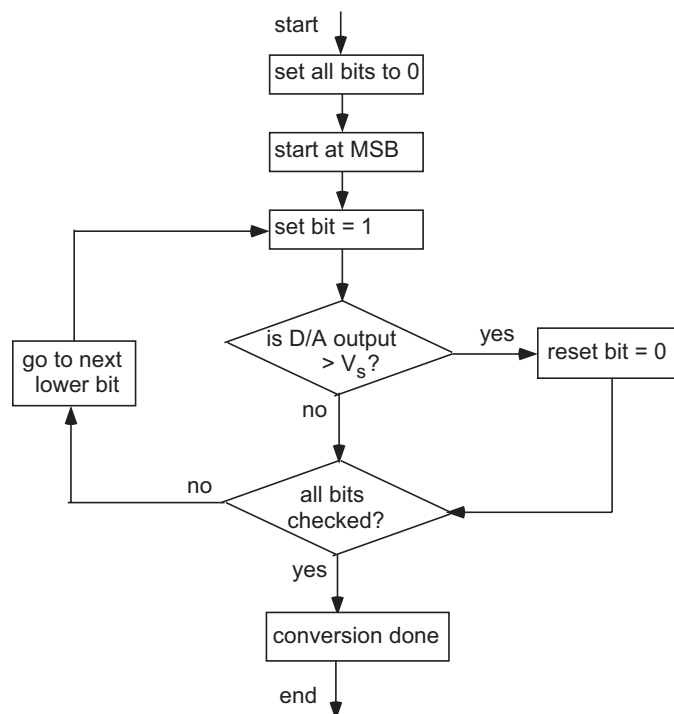
- Types of A/D converters [2]:
  - flash
  - successive approximation (MPC555)
  - single-slope (or dual-slope) integration
  - sigma-delta converters
  - redundant signed digit (RSD) [5] (MPC5553)
- Design issues
  - precision
  - accuracy
  - speed
  - cost
  - relative amount of analog and digital circuitry
- Performance Metrics [4]
  - quantization error
  - offset and gain error
  - differential nonlinearity
  - monotonicity
  - missing codes
  - integral nonlinearity

# Successive Approximation A/D Converter

- Used on the Freescale MPC555
- Bits set in succession, from most to least significant



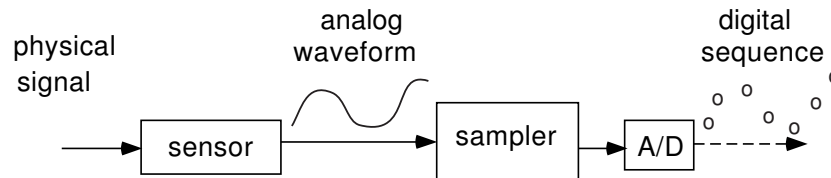
- Control logic [1]



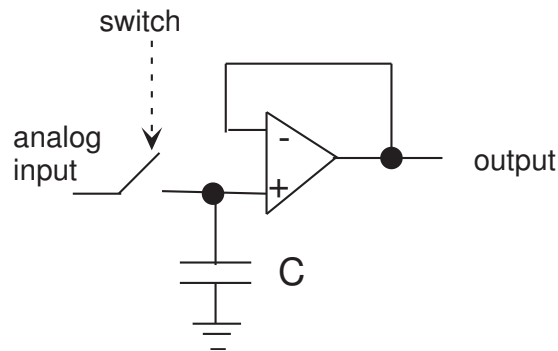
- Issues
  - timing (bits set one at a time)
  - signal to noise ratio (lower bits based on small signals)
  - cannot correct for wrong decisions on a given bit

# Sampling

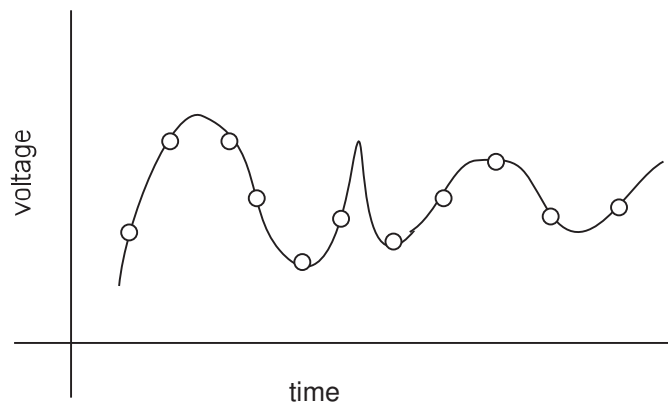
- Convert an analog function of time into a sequence of binary numbers



- sampler [3]

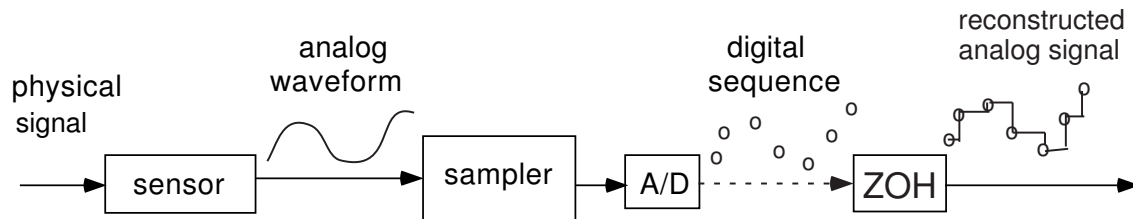


- *Information loss* in representing an analog function as a discrete sequence [2, 3, 8, 6]

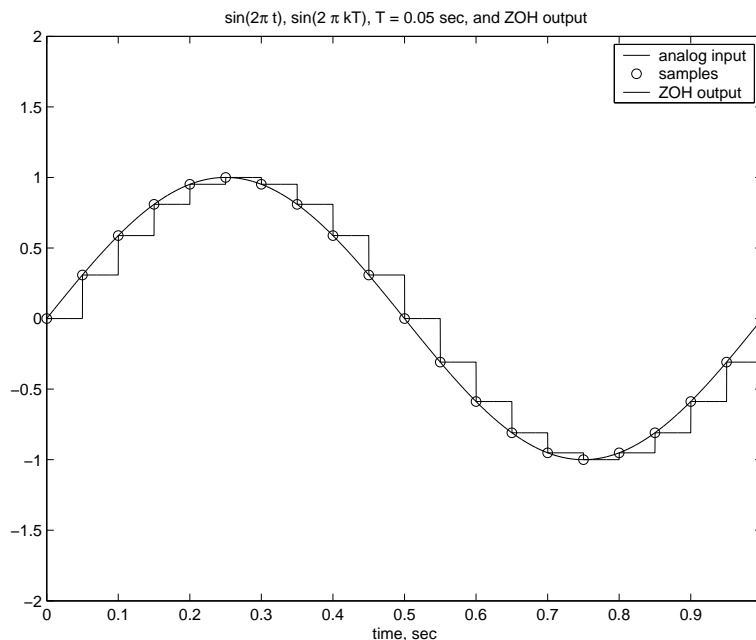


# Information Loss in Sampling

- How to describe information loss?
- Idea: Try to reconstruct the analog signal from its digital representation. This may be done by a D/A converter.



- “Staircase” output<sup>1</sup>:  $\sin(2\pi t)$  sampled with sampling period  $T = 0.05$  seconds (sampling frequency  $f = 20$  Hz)

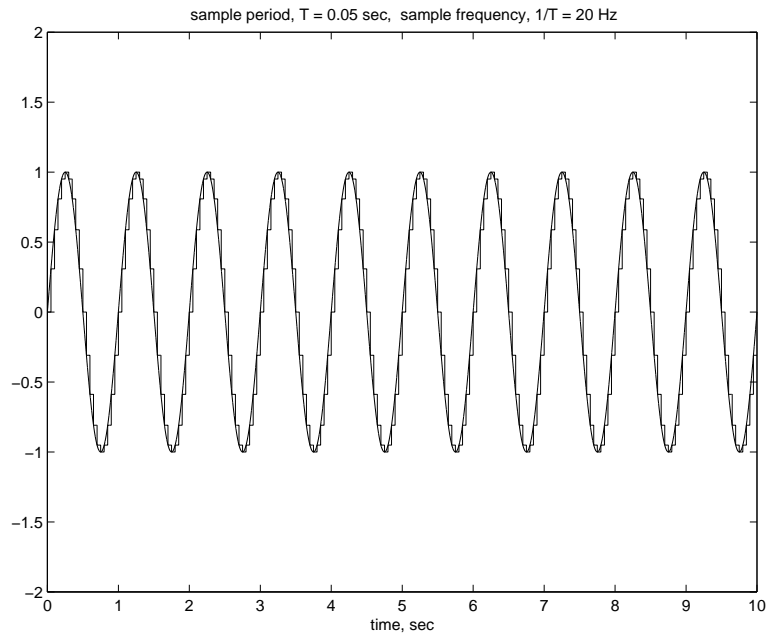


- a staircase approximation of the input delayed by  $T/2$  seconds

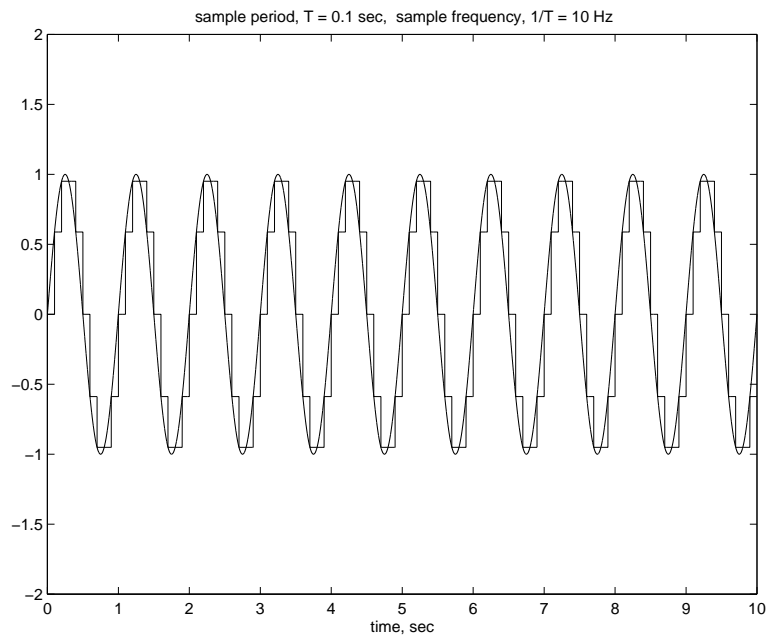
<sup>1</sup>created with MATLAB files staircase\_approx.m and simulate\_ZOH.mdl

# Fast Sampling

- Compare fast and slow sampling<sup>2</sup>
- Input: a 1 Hz sinusoid,  $\sin(2\pi t)$ , sampled at 20 Hz



- Input: a 1 Hz sinusoid,  $\sin(2\pi t)$ , sampled at 10 Hz

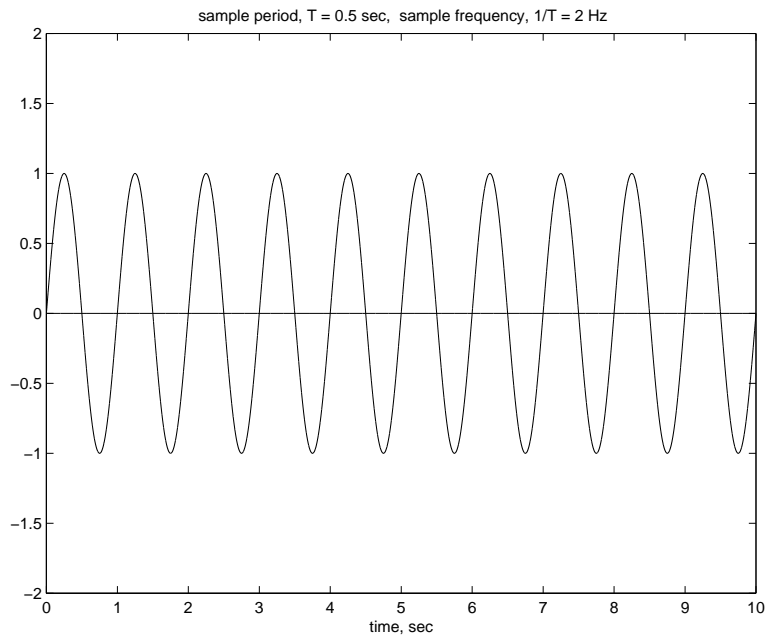


---

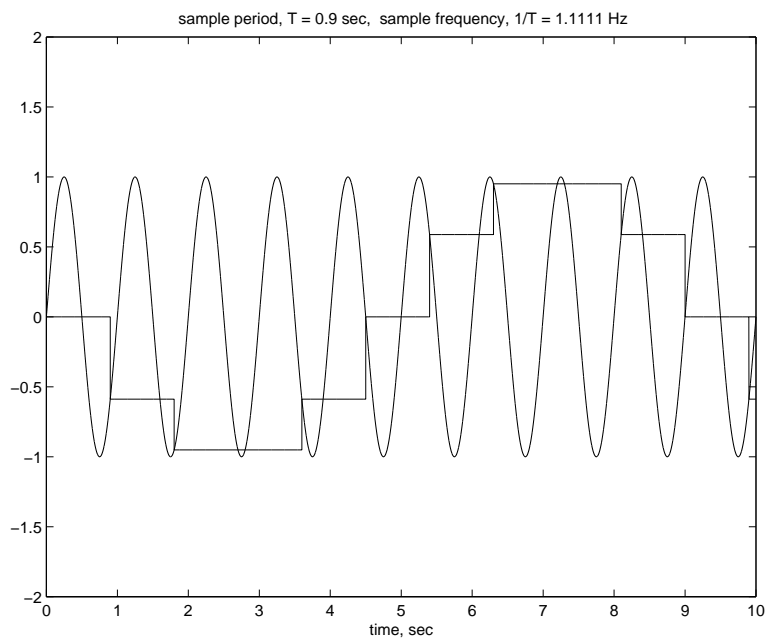
<sup>2</sup>MATLAB m-file fast\_slow\_sampling.m

# Slow Sampling

- Input: a 1 Hz sinusoid,  $\sin(2\pi t)$ , sampled at 2 Hz



- Input: a 1 Hz sinusoid,  $\sin(2\pi t)$ , sampled at 1.11 Hz

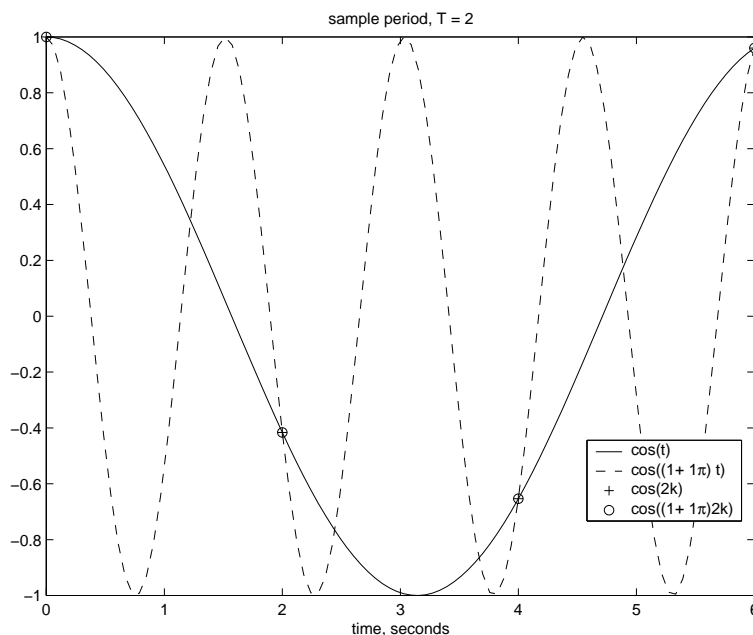


# Observations on Sampling

- If sampling period is fast with respect to period of the signal, then the reproduced signal approximates the original signal.
  - slight staircase effect
  - slight time delay
- If sampling period is relatively slow, then there are the reproduced signal may differ significantly from the original signal.
  - It may equal zero!
  - It may look like a periodic signal of equal amplitude but longer period.
- Other issues[6]
  - irregular sampling interval
  - synchronizing sampling with the signal

# Aliasing

- Suppose we have two analog signals whose values are identical at the sample points. *Then their digital representations will also be identical.*
  - Impossible to reconstruct original signal from its digital representation.
  - Any algorithm on the CPU will be unable to distinguish between signals.
  - Especially problematic when sampling noisy analog signals.
  - $\sin(2\pi t)$  sampled at  $0, 0.5, 1, 1.5, \dots$  seconds is indistinguishable from zero!
  - $\cos(t)$  and  $\cos((1 + \pi)t)$  are identical at  $0, 2, 4, \dots$  seconds<sup>3</sup>!



- A higher frequency signal that “masquerades” as a low frequency signal after sampling is said to be *aliased*.

---

<sup>3</sup>MATLAB m-file aliasing.m

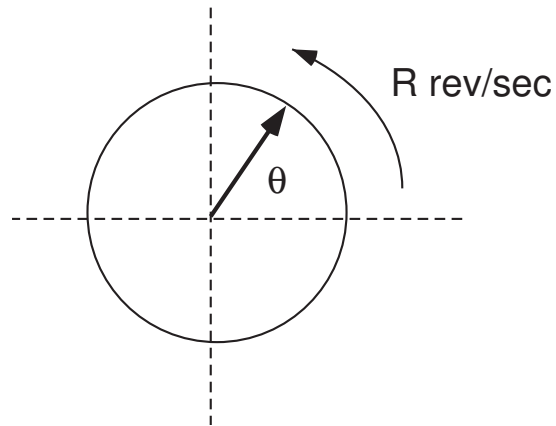
# Effects of Aliasing

- Aliasing is a type of information distortion that results from *undersampling*.
- Questions:
  1. How fast must one sample an arbitrary signal to avoid aliasing?
  2. When is aliasing likely to be a problem in sensor interfacing?
  3. How does one minimize the effects of aliasing?
- We shall return to these questions after an example and a review of some ideas from signals and systems.

Example: Consider a video of a rotating wheel marked with an arrow, and made with a camcorder at a rate of 30 frames/second [8]...

## Aliasing and the Wheel,I

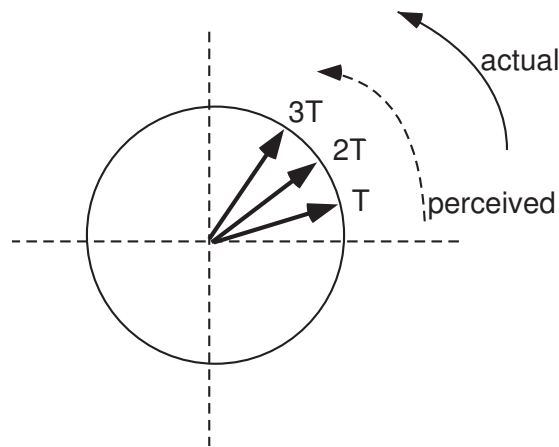
- The effects of aliasing can be striking...
- Consider a wheel rotating counterclockwise (CCW) at  $R$  rev/seconds.
- Suppose we
  - View the wheel with a strobe light every  $T$  seconds, or
  - Use a camcorder to make a video with one frame every  $T$  seconds.



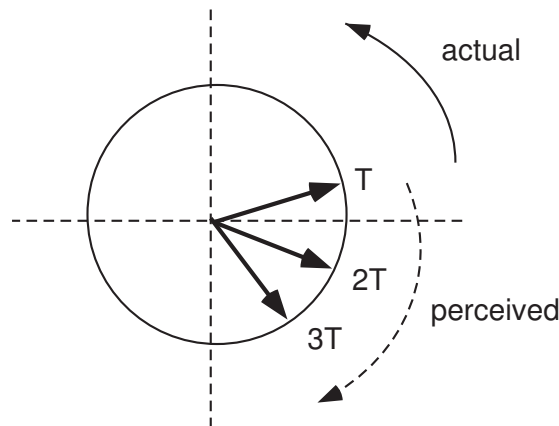
- Depending upon the relative values of  $T$  and  $R$ , the wheel may appear to be
  - rotating CCW – as we expect to see
  - stationary – not moving!
  - rotating clockwise (CW) – backwards!

## Aliasing and the Wheel, II

- We visually determine the direction of motion by noting the difference between consecutive measurements of the position of the arrow.
- If  $T$  is *fast* with respect to the speed of rotation, then motion appears to be CCW:

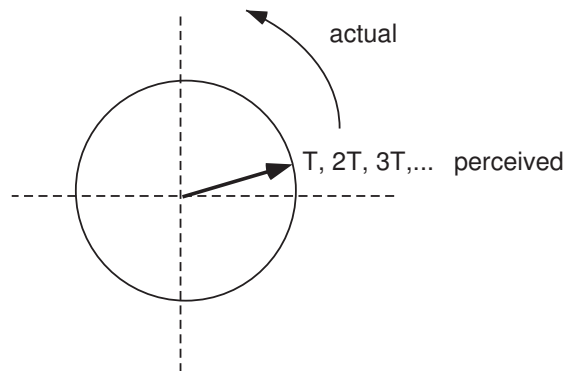


- If  $T$  is *slow* with respect to the speed of rotation, then motion appears to be CW:

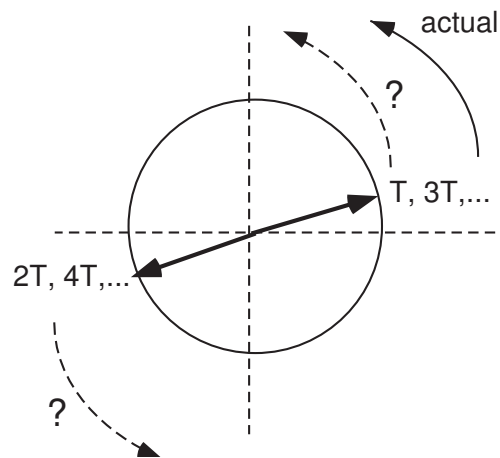


## Aliasing and the Wheel, III

- At an even slower value of  $T$ , wheel appears to be stationary:

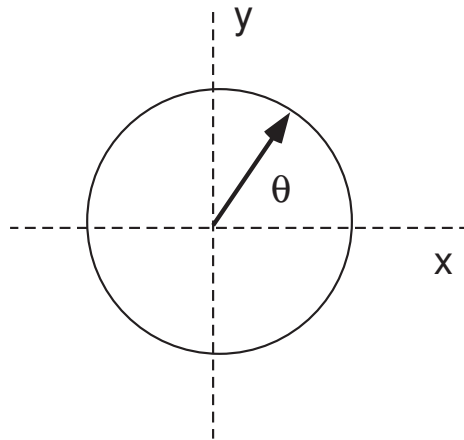


- At an intermediate value of  $T$ , we are only confused:



## Aliasing and the Wheel, IV

- Suppose the wheel rotates CCW at a fixed rate  $R$  rev/sec. Can we determine the maximum value of  $T$  so that the wheel always seems to be rotating (and rotating CCW)?
- Terminology
  - sample period,  $T$  seconds
  - sampling frequency,  $f = 1/T$  Hz or  $\omega_s = 2\pi/T$  rad/sec
  - rotation rate,  $R$  rev/sec, or  $2\pi R$  rad/sec
- Position of wheel in  $(x, y)$  coordinates is given by



$$x(t) = \cos(2\pi Rt)$$

$$y(t) = \sin(2\pi Rt)$$

$\Rightarrow$  Taking a picture of the wheel every  $T$  seconds is equivalent to “sampling” a sine wave every  $T$  seconds

## Aliasing and the Wheel, V

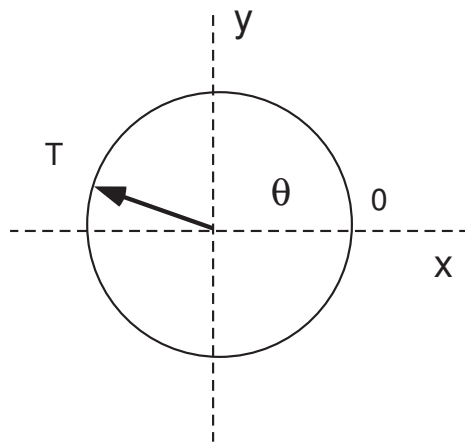
- It takes  $1/R$  seconds for the wheel to make a complete revolution.
- Suppose that initially  $\theta(0) = 0^\circ$ . Hence if we sample at  $T = 1/R$  samples/second, then the position coordinates at the sample times  $kT$ ,  $k = 1, 2, 3\dots$  satisfy

$$x(kT) = \cos(2\pi k) = x(0) = 1$$

$$y(kT) = \sin(2\pi k) = y(0) = 0$$

$\Rightarrow$  the wheel looks as though it were stationary

- To determine the correct direction of rotation, we need to take *at least one* sample before it reaches the halfway point:



- The wheel reaches  $\theta = 180^\circ$  in  $1/2R$  seconds, hence we require
  - sample period  $T < 1/2R$  sec
  - sample frequency  $\omega_s > 4\pi R$  rad/sec ( $f > 2R$  Hz)
- Later we shall rederive this result from the *Shannon sampling theorem* [6, 8]

# Fourier Series

- Consider a periodic time signal  $f(t)$ ,  $t \geq 0$ , with period  $T$ :  
 $f(t) = f(t + kT)$ ,  $k = 0, 1, 2, \dots$
- Examples:
  - sine wave
  - square wave
  - sawtooth wave
- Then  $f(t)$  may be expressed as a sum of (possibly infinitely many) sines and cosines.
- Terminology
  - $T$ : period of signal
  - $\omega_0 = 2\pi/T$ : frequency in rad/sec
  - $f = 1/T$ : frequency in Hz
- Then

$$f(t) = a_0 + \sum_{n=1}^{\infty} (a_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t))$$

- More terminology
  - Fourier coefficients:  $a_i, b_i$
  - DC term:  $a_0$
  - fundamental:  $n = 1$ , sinusoids of frequency  $\omega_0$
  - harmonics:  $n > 1$ , sinusoids of frequency  $> \omega_0$

## Examples of Fourier Series

- Example: A sine wave with period  $T$

$$f(t) = \sin\left(\frac{2\pi}{T}t\right)$$

is its own Fourier series expansion

- Unit amplitude square wave with period  $T$  has Fourier expansion

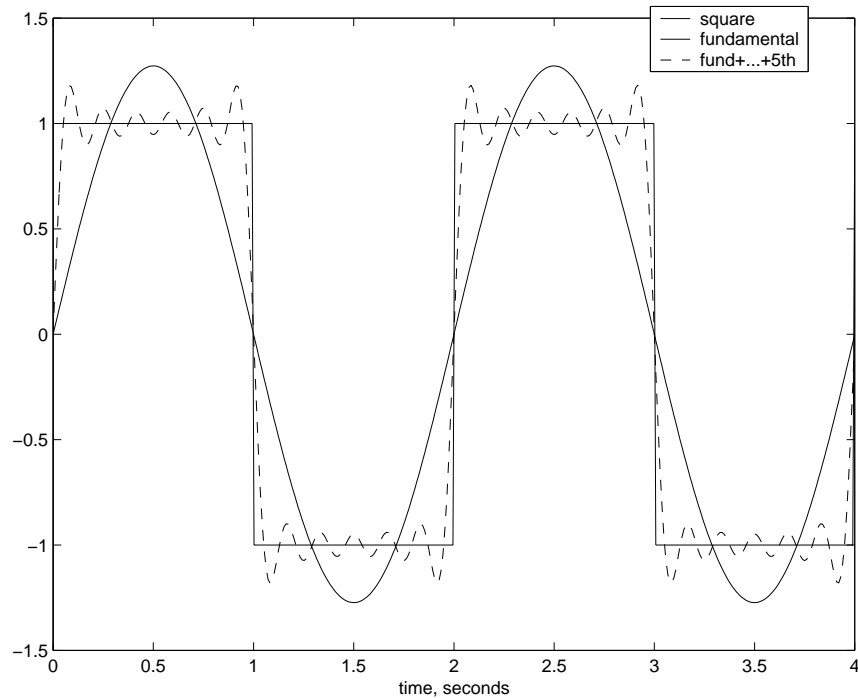
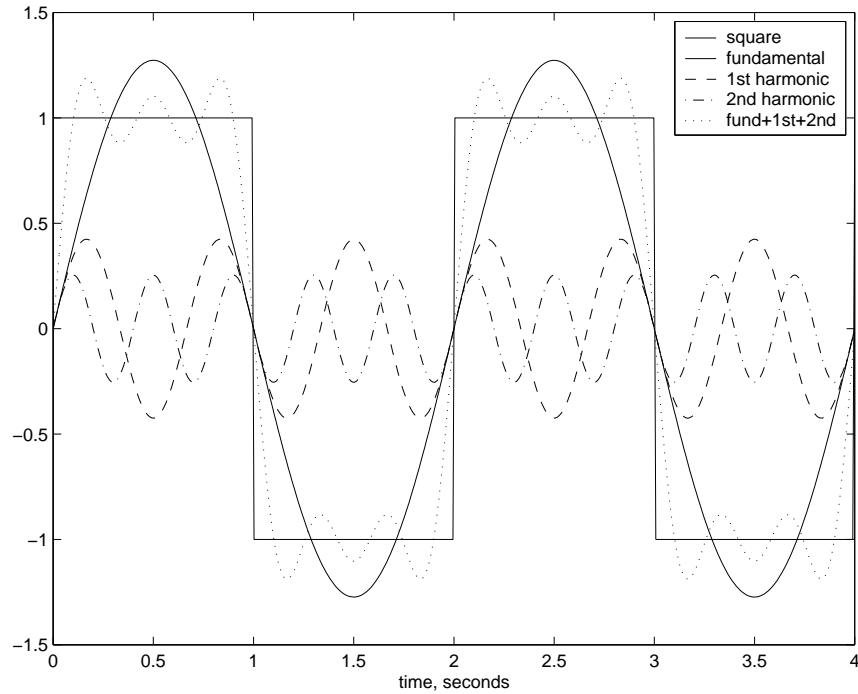
$$f(t) = \sum_{\substack{n=1 \\ n, \text{ odd}}}^{\infty} \frac{4}{n\pi} \sin(n\omega_0 t)$$

where  $\omega_0 = 2\pi/T$  is the frequency of the square wave in rad/sec ( $f = 1/T$  is the frequency in Hz)

- Fundamental:  $n = 1$ ,  $\frac{4}{\pi} \sin(\omega_0 t)$
- 1st harmonic:  $n = 3$ ,  $\frac{4}{3\pi} \sin(3\omega_0 t)$
- 2nd harmonic:  $n = 5$ ,  $\frac{4}{5\pi} \sin(5\omega_0 t)$

## More Terms $\Rightarrow$ Better Approximation

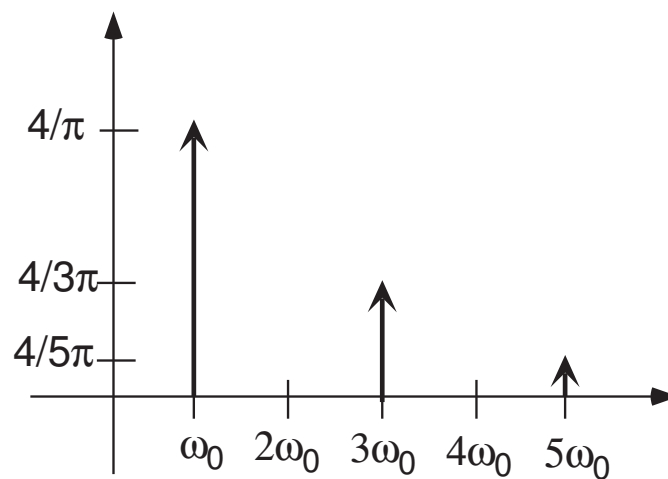
Fourier series of a square wave with period  $T = 2$  seconds<sup>4</sup>.



<sup>4</sup>Matlab m-file sq\_wave.m

## Frequency of a Signal

- Consider a periodic signal, such as a square wave, that has “sharp corners”.
- In general, many high frequency terms are needed to construct such “sharp corners”. In fact, any signal with relatively abrupt changes will contain high frequencies, even if the changes are not discontinuous.
- It is useful to sketch the location, and relative amplitude, of the various frequency components of a signal
- Example: unit amplitude square wave

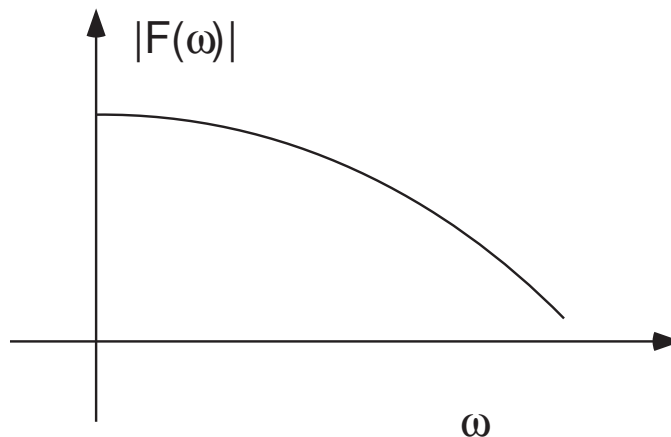


# Fourier Transform

- Most signals are *not* periodic. Nevertheless, it is possible to think of “almost any” signal as the sum of sines and cosines of *all* frequencies.
- Fourier transform [7]: Under certain conditions, we can write

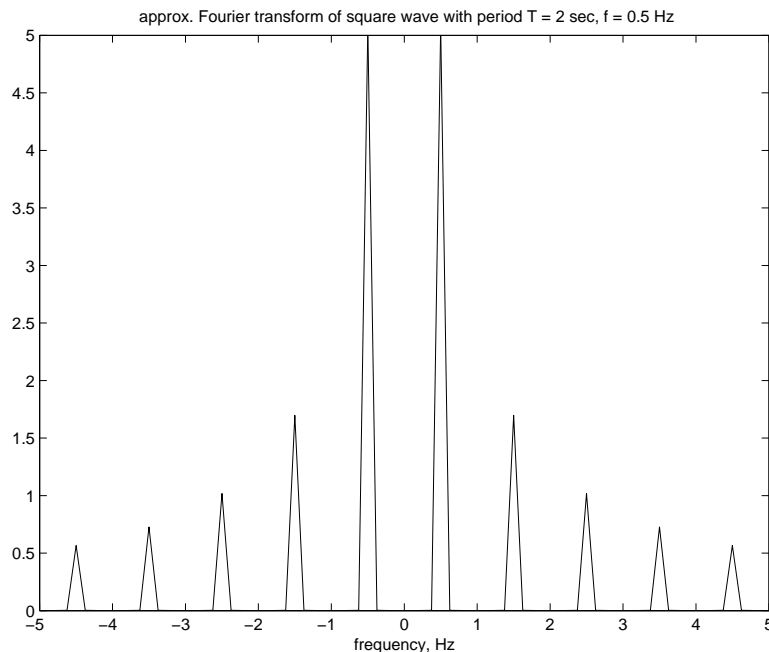
$$\begin{aligned} F(\omega) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} f(t) e^{j\omega t} dt \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} f(t) \cos(\omega t) dt + \frac{j}{2\pi} \int_{-\infty}^{\infty} f(t) \sin(\omega t) dt \end{aligned}$$

- We will not need any of the details of the Fourier transform. However, it is important to remember that time signals may be given a frequency representation.
- Can visualize the frequency content of a signal by plotting  $F(\omega)$  as a function of frequency:



# Fourier Transform of a Periodic Signal

- The Fourier transform of a sinusoid of frequency  $f$  Hz consists of two “delta” functions located at frequencies  $\pm f$  Hz.
- The frequency response of a square wave consists of “delta” functions corresponding to all frequency components of the Fourier series expansion of the square wave.
- Example<sup>5</sup>: Square wave of period  $T = 2$  seconds,  $f = 0.5$  Hz has frequency components at  $\pm f, \pm 3f, \pm 5f, \dots$ . The Fourier transform of a square wave may be approximated using algorithms from [7]



---

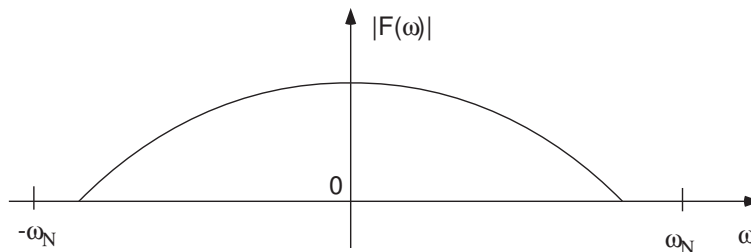
<sup>5</sup>MATLAB m-file sq\_wave.m

# Frequency Response in Embedded Systems Applications

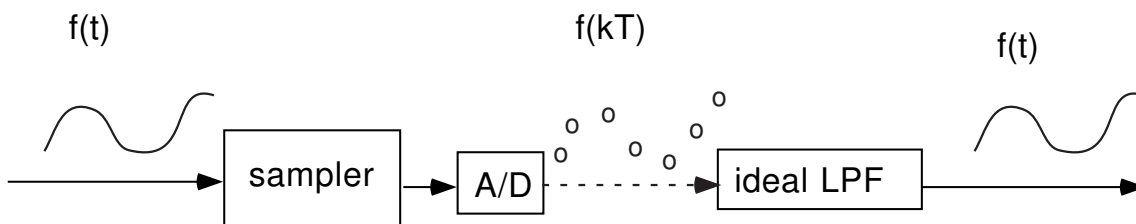
- Many embedded systems for control, communications, and signal processing applications – and anything to do with audio or video – require knowledge of *frequency content* of signals.
- an important class of embedded processors – DSP chips – has a special architecture that allows rapid computation of the frequency response of a signal using the Fast Fourier Transform (FFT) algorithm.
- Knowledge of frequency content is needed to design the interface electronics for an embedded system. For example, circuits that implement lowpass filters to remove unwanted high frequencies.
- Frequency response ideas arise in the study of sampling and aliasing, and in the use of Pulse Width Modulation (PWM) to drive a DC motor.

# Shannon Sampling Theorem

- Recall Question 1: How fast must we sample to avoid aliasing?
- Shannon's Theorem [6, 8]
  - Consider a signal  $f(t)$  with frequency response  $F(\omega)$ .
  - Suppose we sample  $f(t)$  periodically, with period  $T$  sec, and define the *Nyquist frequency*  $\omega_N = \pi/T$  radians/second ( $f_N = 1/2T$  Hz).



- If  $F(\omega) = 0$ , for  $|\omega| \geq \omega_N$ , then it is possible to reconstruct  $f(t)$  *exactly* from its samples  $f(kT)$ .
- Reconstruction requires an *ideal lowpass filter*:



- In practice, reconstruction can only be done approximately, because perfect reconstruction requires *all* samples of the signal, even those in the future!
- Nevertheless, this result tells how fast we must, in principle, sample to avoid aliasing: at least twice as fast as the highest frequency in the signal!

## Aliasing and the Wheel, VI

- Suppose that the wheel rotates at  $R$  rev/sec, or  $2\pi R$  rad/sec.
- Then position coordinates

$$x(t) = \cos(2\pi Rt)$$

$$y(t) = \sin(2\pi Rt)$$

are sinusoids with frequency  $\omega_0 = 2\pi R$ .

- Nyquist says that to avoid aliasing we sample fast enough that

$$\omega_0 < \omega_N = \frac{\pi}{T} \text{ rad/sec} \quad \Rightarrow \quad T < \frac{1}{2R} \text{ sec}$$

- Same result as we derived before!

## Nyquist and Embedded System Applications

- A frequency analysis is done of each analog signal that must be measured with a sensor and represented in digital form.
- Although the signals will have energy at all frequencies, usually the “information” lies in some low frequency range, say  $\omega < \omega_0$ .
- If possible, set the sample period  $T$  so that the Nyquist and sampling frequencies satisfy

$$\omega_N = \frac{\pi}{T} > \omega_0 \quad \Leftrightarrow \quad \omega_s = \frac{2\pi}{T} > 2\omega_0$$

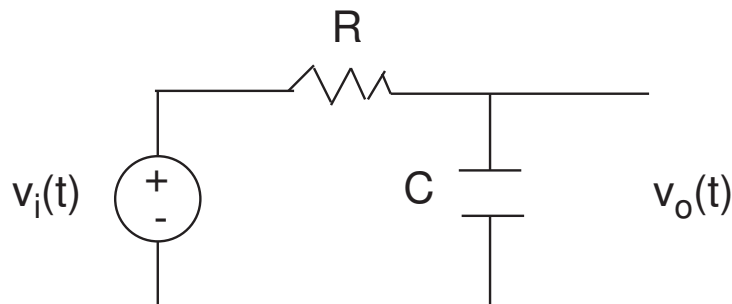
(usually, we set sampling frequency  $\omega_s > (5 - 10)\omega_0$ , twice as fast is only the theoretical limit)

## Problems with Aliasing

- When is aliasing likely to be a problem?
- Almost all signals are corrupted by noise
  - 60 Hz hum
  - EMI from spark ignition
  -
- Often the noise is at a higher frequency than the information contained in the signal. If the noise is at a sufficiently high frequency, it will get “aliased” to a lower frequency, and corrupt the signal we are trying to measure.
- How to resolve?

# Frequency Response Functions

- a linear filter has a frequency response that determines how it responds to periodic input signals
- Example: RC circuit



- frequency response function

$$H(j\omega) = \frac{1}{1 + j\omega RC}$$

- magnitude, or gain

$$|H(j\omega)| = \frac{1}{\sqrt{1 + \omega^2 R^2 C^2}}$$

- phase

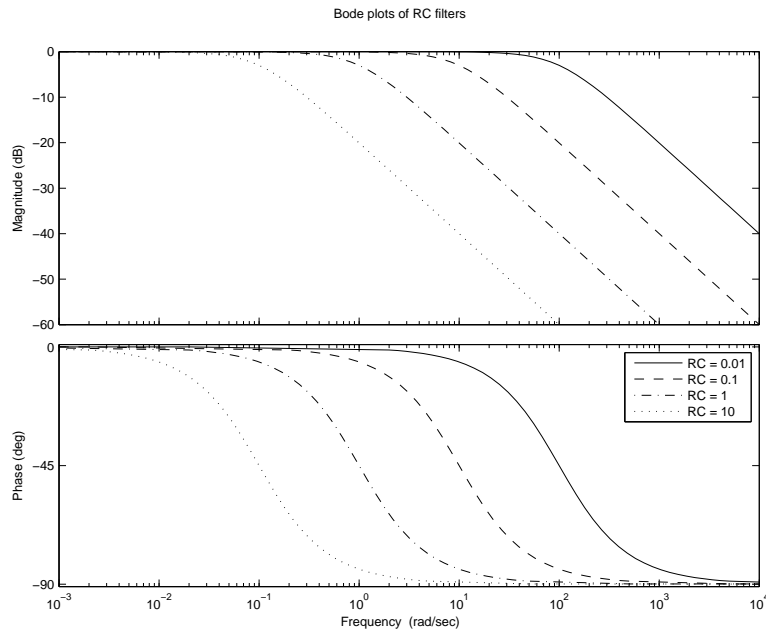
$$\angle H(j\omega) = -\tan^{-1}(\omega RC)$$

- After transients die out, the steady state response of the filter to a sinusoid is determined by its frequency response function:

$$v_i(t) = \sin(\omega_0 t) \Rightarrow v_o(t) \rightarrow |H(j\omega_0)| \sin(j\omega_0 t + \angle H(j\omega_0))$$

# Gain and Phase Plots

- Bode plots: gain and phase vs frequency<sup>6</sup>



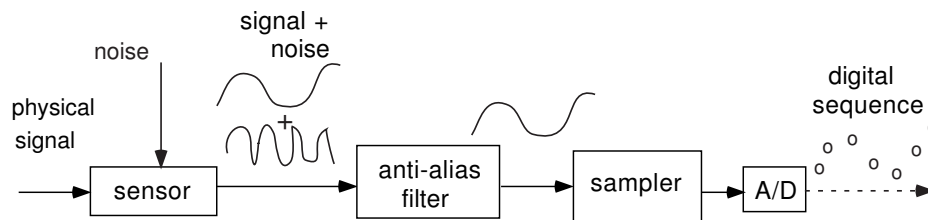
- Lowpass filter
  - passes low frequencies
  - attenuates high frequencies
  - introduces phase lag
- Bandwidth of  $RC$  filter proportional to  $1/RC$

---

<sup>6</sup>MATLAB m-file RC\_filter.m

# Anti-Aliasing Filters

- Potential solution to aliasing problem: “anti-aliasing filters” that are inserted before the sampler to remove high frequencies



- Commercial devices often have an AA filter built in, but may need to build another one to configure the frequency response for the application.
- Problems:
  - may not have frequency separation between signal and noise
  - phase lag in control applications

## References

- [1] <http://hyperphysics.phy-astr.gsu.edu/hbase/electronic/adc.html#c3>.
- [2] D. Auslander and C. J. Kempf. *Mechatronics: Mechanical Systems Interfacing*. Prentice-Hall, 1996.
- [3] W. Bolton. *Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering, 2nd ed.* Longman, 1999.
- [4] J. Feddeler and B. Lucas. *ADC Definitions and Specifications*. Freescale Semiconductor, Application Note AN2438/D, February 2003.
- [5] M. Garrard and P. Ryan. *Design, Accuracy, and Calibration of Analog to Digital Converters on the MPC5500 Family*. Freescale Semiconductor, Application Note AN2989, July 2005.
- [6] S. Heath. *Embedded Systems Design*. Newness, 1997.
- [7] E. Kamen and B. Heck. *Fundamentals of Signals and Systems using MATLAB*. Prentice Hall, 1997.
- [8] J. H. McClellan, R. W. Schafer, and M. A. Yoder. *DSP First: A Multimedia Approach*. Prentice-Hall, 1998.