Institute for Cognitive Systems
Technische Universität München
Prof. Gordon Cheng

Humanoid Sensors and Actuators
Tutorial 5 – Sensors and signal processing

In this tutorial we refresh some basics of signal processing, including correlation, FFT and filtering. All tasks are solvable with Matlab, Octave or Scilab with a slightly varying syntax.

1. Sound Source Localization

R1.1: Find the expression to determine the azimuth angle of a sound source for a system with two microphones. (Probe the equations shown in the slides of Lecture 4). **(3 points)**

R1.2: Find the expression to determine the velocity of a target from the pulse duration difference of a radar sensor. (Probe the equations shown in the slides of Lecture 4) **(3 points)**

R1.3: What kind of waves is better to measure the distance to a target? **(1 point)**

R1.4: What kind of waves is better to measure the speed of a moving target? **(1 point)**

2. Fast Fourier Transform

a) Create a signal consisting of the sum of two sine waves using a sample frequency of 1000Hz, one with amplitude of 1 and frequency of 50 Hz and the other with amplitude 0.5 and frequency 120 Hz. Plot the signal over a time slot of 2s.

b) Run a FFT (fast Fourier transform) on the signal from (2.a) and plot it. Normalize the output to 1 and only show positive frequencies.

c) Add random noise to the signal from (2.a) and plot the signal again.

d) Run the FFT on the noisy signal from (2.c) and plot it.

R2.1: Is it possible to implement FFT to an online data streaming? Why? **(2 points)**

R2.2: How can you use FFT in signal processing? **(2 points)**

R2.3: Deliver the code used to generate the signals and plots? **(4 points)**

3. Audio Correlation

a) Load the 'chimes.wav' file into the workspace and isolate one of its channels.
   - Plot the signal with a reasonable time scale.
   - If you have speakers/headphones: try to output the audio signal.

b) Run an FFT on the audio signal from (3.a) and plot it.

c) Generate a left and a right channel from the signal of (3.a) and add a delay and scaling factor to one of them. Make the delay and scaling factors variable.

d) Plot the two signals in the same figure.

e) If you have speakers/headphones: listen to the signals and find parameters at which stereo localization works for you.
f) Run cross-correlation on both signals and plot the correlation against delay.
g) Add noise with a normal distribution to both channels.
h) Run cross-correlation on the noisy signals and plot.

R3.1: Is it possible to implement the cross-correlation to an online data streaming? Why? **(1 point)**
R3.2: If you answered "no" to R3.1, how would you work around to use it to identify the interaural time delay? **(1 point)**
R3.3: Deliver the code to generate the signals and the plots. **(4 points)**

4. Signal Filtering

a) Create a signal consisting of two sine waves, one with amplitude of 1 and a frequency of 50 Hz and one with amplitude 0.5 and frequency of 500 Hz using a sample frequency of 10,000 Hz.
b) Plot the signal over a time slot of 0.1s.
c) Design an analog low-pass passive filter with a cut-off frequency of 50 Hz
d) Design and implement a first order discrete low-pass filter with cut-off frequency of 50 Hz.
e) Apply the filter from (4.d) to the signal created in (4.a). Plot again the signal after filtering.
f) Design an analog high-pass passive filter with a cut-off frequency of 500 Hz
g) Design and implement a first order discrete high-pass filter with cut-off frequency of 500 Hz.
h) Apply the filter from (4.g) to the signal created in (4.a). Plot again the signal after filtering.
i) Create another sine wave signal with a frequency of 1,000 Hz using a sample frequency of 10,000 Hz. Then, add it to the signal created in (a).
j) Design an analog band-pass active filter to recover the 500 Hz signal.
k) Design and implement a discrete band-pass filter to recover the 500 Hz signal from the superposed signal.
l) Apply the filter from (4.k) to the signal created in (4.i). Plot again the signal after filtering.

R4.1: Detail the design process of the filter in (4.c). **(2 points)**
R4.2: Detail the design process of the filter in (4.d). **(2 points)**
R4.3: Detail the design process of the filter in (4.e). **(2 points)**
R4.4: Detail the design process of the filter in (4.f). **(2 points)**
R4.5: Detail the design process of the filter in (4.j). **(2 points)**
R4.6: Detail the design process of the filter in (4.k). **(2 points)**
R4.7: What is the difference between passive and active filters? **(1 point)**
R4.8: What is the "order" of a discrete filter? **(1 point)**

R4.9: How could you implement a continuous-time filter in a robotic system? **(1 point)**

R4.10: How could you implement a discrete-time filter in a robotic system? **(1 point)**

R4.11: What are the advantages and disadvantages of analog and digital filters in robotic systems? **(1 point)**

R4.12: Is it possible to make a 1000 Hz Low-Pass filter in a digital system with a sampling rate of 1000 Hz? **(1 point)**

R4.13: Deliver the code to generate the signals and the plots. **(6 points)**