Intelligent Embedded Systems Laboratory

Implementation of a selected signal processing algorithm

One of the following measurement tasks should be selected and solved in consultation with the instructor during the two lab sessions. It is also possible to implement a task not listed below, chosen freely by the group if agreed in advance with the instructor.
Most of the tasks require the setting of certain parameters - this can be done using the four push buttons on the development card or using UART communication with the computer keyboard. Data can also be displayed via UART communication.

1. **LMS algorithm implementation.** The LMS algorithm is used to identify an unknown system. The reference signal is applied to one input of the development board and to the input of the system to be identified. The output of the unknown system is fed to the other input of the development board. One of the outputs of the card is the error signal of the algorithm, which can be observed using an oscilloscope.

2. **Phase-locked loop (PLL) implementation.** To solve this problem, a C library function is provided which returns the sine of an arbitrary angle. The task is to control the frequency of the sine signal generated by using this function, so that its frequency is equal to the frequency of the periodic signal given as input.

3. **Implementation of a signal generator.** The task is to implement a signal generator capable of generating sine, sawtooth, square wave and triangular signals of adjustable frequency and amplitude. The library function mentioned in the PLL task above can be used to generate the sine signal, the other signals mentioned can be generated either by Fourier decomposition or by using a counter. The signal generator should be able to generate linear and logarithmic frequency sweeps in continuous operation. The sweep start and end frequency and sweep time shall be adjustable.

4. **Implementation of a median filter.** The task is to implement a median filter with length N=2K+1 (integer K) using different ordering algorithms. This implies on the one hand the sorting of the entire data set of length N using e.g. bubble or quick sort algorithms, and on the other hand an efficient implementation that takes into account that in each filtering step one element (the oldest data) is deleted and the new data is inserted into the remaining sorted list. A MATLAB generated sine signal corrupted by noise pulses can be used as a test signal.

5. **FFT Analyzer.** Calculate the FFT of the input data block using the FFT library function and send the amplitude of the spectrum to the output in a form that can be displayed by an oscilloscope.

6. **BFSK (binary frequency shift keying) decoder.** Based on a variable frequency sine wave signal generated in MATLAB, the binary signal stream is generated, converted to ASCII code and sent out on UART.

7. **Implementation of an oscilloscope.** The periodic signal led to the input of the DSP board must be displayed on the screen of the computer by means of UART communication: in the 80 * 25 character window of the terminal, a "*" character should be drawn according to the signal shape.

8. **Implementation of additive synthesis.** Generating musical sound from the computer keyboard or from a pre-programmed melody. The musical sound should be produced as a sum of sine signals, where the amplitude of the sine components, represented in dB, is a straight line whose slope is adjustable. The amplitude of each note is controlled by an envelope (ADSR) generator.

9. **Implementation of virtual analog synthesis.** Generate musical sound from a computer keyboard or from a pre-programmed melody. The musical sound is produced by low-pass filtering of simple waveforms (square wave and sawtooth) similarly to old analogue synthesizers. The duty cycle of the square wave and the symmetry of the sawtooth signal shall be adjustable. This parameter is modulated by a low frequency oscillator (LFO).

10. **Implementation of wavetable synthesis.** Generate musical sound from a computer keyboard or from a pre-programmed melody. The musical sound is produced by the periodic playback of a waveform. The wavetable is drawn in a drawing program (e.g. Paint) and then the image is converted to a C array in MATLAB. Linear interpolation is applied between each sample point during playback.

11. **Implementation of an equalizer (EQ).** The task is to implement a three-band EQ using three FIR filters connected in parallel. The FIR filters shall be designed so that the resulting transfer function is unity at zero dB settings.

12. **Implementation of a delay-based effect.** Implement a feedback delay line for delay effects and modulate the delay line length for vibrato, chorus and flanger effects. The delay line shall be implemented with a circular buffer, in case of modulation the delay line signal shall be linearly interpolated.

13. **Implementation of hard clipping distortion.** An implementation of a distortion effect that clips the signal above and below a certain signal level. To reduce aliasing, the input signal is interpolated to a higher sampling frequency and then decimated after nonlinear distortion.