ME477, Lab 6

Ali Jones

2/25/2022

Description:

Like last week's lab, this program demonstrates the use of an interrupt thread, a subfunction that is external to the microcomputer and not directly called but can briefly override the main program functionality. In this lab, however, there is no 'main' functionality that is interrupted; instead, the interrupt runs on a timer. It reads in an analog signal, applies a difference equation (the discretized version of a differential equation) and outputs the transformed signal.

```
main
|_ Irq_RegisterDiIrq()     - reserves interrupt, configures DI and IRQ
|_ pthread_create()         - creates new thread to service interrupt
|_ Irq_UnregisterDiIrq()    - unregisters interrupt
|_ wait()                   - waits for 5ms
|
Timer_Irq_Thread            - performs function in response to interrupt
|_ printf_lcd()             - prints to LCD screen
|_ Irq_Wait()               - waits for IRQ number or ready signal
|_ Aio_Read()               - reads an input channel
|_ Aio_Write()              - writes to an output channel
|
cascade                     - calculates the difference equation with a
                                                              biquad cascade
```

Testing:

1. Run the program
2. With "emulate.h" included:
    a. Wait for a few seconds, then press "delete" on the keypad
    b. The MATLAB output will show a square wave input and transformed output (see below)
3. Without "emulate.h" included:
    a. The program fails with a 'segmentation fault' error

Results:

While this program almost functions completely, I ran into some problems in lab. I first tested its functionality without the biquad, passing the input waveform directly to the output. The oscilloscope displayed both waves at the input frequency. Then I included the `cascade()` function and tested it with the emulator. The output was as expected; the output matched the input square wave with some ripple. When I plugged in to the oscilloscope, however, the code would not run and gave me a *segmentation fault* error. Additionally, when I tried passing the input directly into the output like before, I received the same error.

This program could be improved by fixing the error that stopped the waveforms from displaying on the oscilloscope. It would also be convenient to adjust transfer function parameters while the program is running to see real-time how the response could be improved.

Figure 1. Oscilloscope showing matching input and output waves

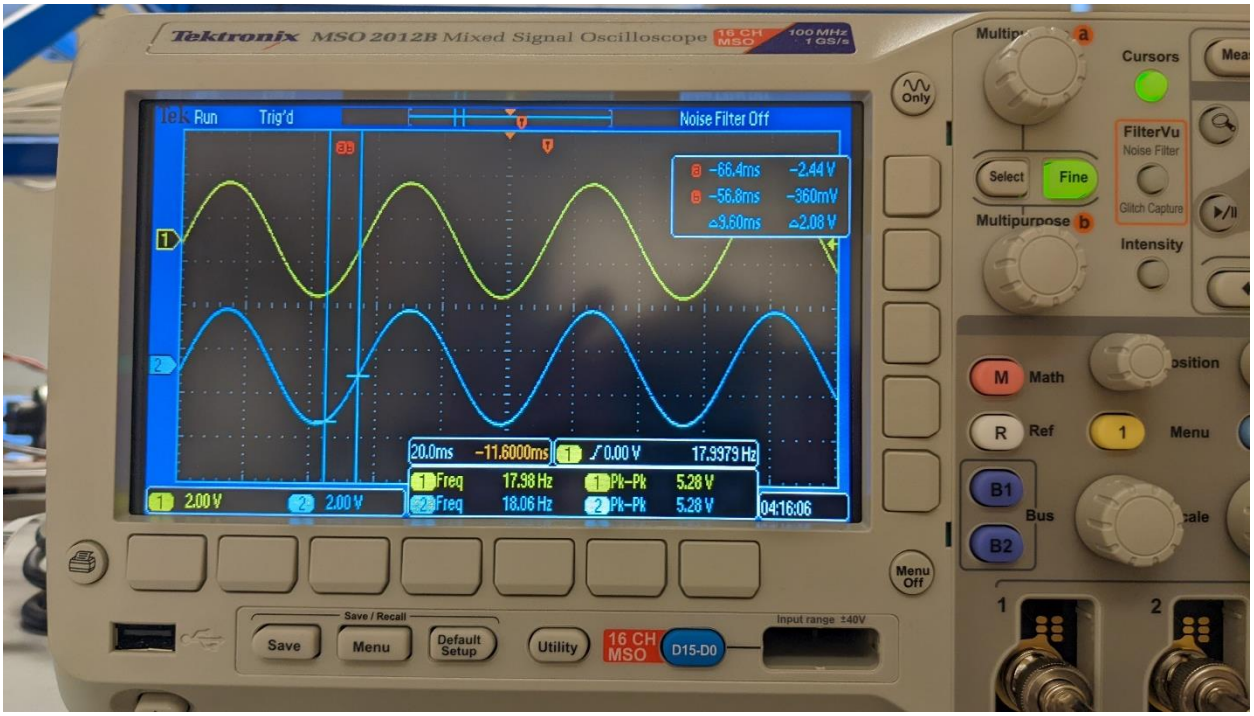

Figure 2. Emulated input and output waveforms