University of Pennsylvania
Department of Electrical and System Engineering
Digital Audio Basics

ESE150, Spring 2021 Final Solutions Wednesday, May 5

See exam as given for figures and question details.

1. **Q1** Assume the servo can be rotated 0 to 360 degrees with 7b worth of control, what is maximum absolute error between an intended angle and the angle you can express to the servo? Express your answer in degrees.

   $\frac{360^\circ}{2^7 \times 2} \approx 1.4^\circ$.

   We divide by 2 because the question asks for absolute error. It never gets more then 1.4° from the closest of the specifiable points.

2. **Q2** How do you extract the loudness of frequency components at the microphone using a microcontroller like the Arduino/ItsyBitsy used in lab?

   Perform a dot product of the time sample sequence with the sine and cosine time samples of the frequency in question. Take the magnitude of the two dot product results to get the loudness of a frequency. Take the min of the loudness of the frequencies in the set.

   Will also take Take the Fourier Transform of the time samples to get the loudness of the frequency components. Take the min of the frequency components in the frequency set.

3. **Q3** Assume the ItsyBitsy take 10 microseconds to read one analog value from one of its inputs.

   Just based on this read time and the 20KHz sampling rate, what is the upper bound on the number of sensors the single ItsyBitsy support?

   20KHz means 1/20KHz = 50μs per sample. So, we can read $\frac{50\mu S}{10\mu S} = 5$ sensors within the sample period.

4. **Q4** Assume we sample at 20KHz and operate on 25ms windows for frequency detection. (assume analog filtering of input is set at the appropriate level to prevent aliasing.) Per frequency whose loudness we would like to extract, how many multiply operations will be required to compute the magnitude of the frequency?

We have $\frac{25{,}000\mu S}{50\mu S} = 500$ samples in the 25 ms window. To extract one frequency, we perform dot products with both the cosine samples and the sine samples. Each dot product requires 500 multiplies. So, we need 1000 multiplies.

5. **Q7** Assuming the whistle continues and the robot continues to orient its head as outline above, will the robot dog be effective at moving closer to the whistle source? why or why not?

   This will be very effective. The control will continually re-orient the head toward the source. So, if the robot dog wanders a bit off course, each new sample window will update the orientation to be closer to the desired destination. Over many samples, the effects of limited resolution will be address. Slight off-course will eventually become measureable and the orientation will correct.

6. **Q8** Continue to assume the ItsyBitsy take 10 microseconds to read one analog value from one of its inputs. Further assume the ItsyBitsy runs at 100MHz and can perform a single multiplication and the other (non-multiply) operations associated with each multiple in 100 cycles. If we need to extract the amplitude of 4 tones from each microphone, how many microphone sensors can a single ItsyBitsy support?

   $$50\mu s = N \times (10\mu s + 4 \times 2 \times 100 \times 0.01\mu s)$$

   $$N = \frac{50\mu S}{18\mu S} \approx 2.8$$

   So, a single ItsyBitsy can only handle 2 microphones.

7. **Q9** Once oriented, it should only be necessary to listen to the position 0 microphone.

   Using the same analogRead time (10 microseconds) and processing assumptions (100MHz operation, 100 cycles for all the work associated with each multiply), how many distinct tones, T, can the ItsyBitsy extract from a single microphone?

   $$50\mu s = 10\mu s + (T \times 2 \times 100 \times 0.01\mu s)$$

   $$T = \frac{40\mu S}{2\mu S} = 20$$

8. **Q12** Provide multiplexer logic for control to orient the front wheel (microphone position 0) to face the whistle sound for an N=8 design.

   There are two components of this solution.

   Fine:

```
tmpplus=mux(currentplus6,current_position,current_position_plus);
tmpminus=mux(currentminus6,tmpplus,current_position_minus);
```

Coarse:

```
t_0_0=mux(loudest[0],tmpminux,16);
t_0_1=mux(loudest[0],32,48);
t_0_2=mux(loudest[0],64,80);
t_0_3=mux(loudest[0],96,112);

t_1_0=mux(loudest[1],t_0_0,t_0_1);
t_1_1=mux(loudest[1],t_0_2,t_0_3);

new_position=mux(loudest[2],t_1_0,t_1_1);
```

For the course component, we turn by the microphone quantization so that microphone 0 will now be where the microphone that previously heard the loudest was. Units are based on the 7b specification, so 128b in the full 360 degrees.

(no one got the the coarse components)

As stated above, this assumes it started at oriengation 0. A more general coarse orientation might be:

```
t_0_0=mux(loudest[0],tmpminux,16+current_position);
t_0_1=mux(loudest[0],32+current_position,48+current_position);
t_0_2=mux(loudest[0],64+current_position,80+current_position);
t_0_3=mux(loudest[0],96+current_position,112+current_position);
```

We should have given you the additons above as input, as well.

9. **Q14** Assume you want to work with a set of at most 256 robot dogs in audible range of the whistle, describe how you form a packet from the T frequencies identified.

   Use each frequency as a bit in the packet. This gives us 20b packets. Use the first 8b to encode the robot dog selection (the destination) and the remaining 12b as payload.

10. **Q15** For your identified packet, how many distinct payload commands can be encoded?

    $2^{12} = 4096$

11. **Q18** How could a robot dog (all robot dogs) relay whistle packets to extend the communication range?

    If the packet is not for the dog that hears it, it could turn around and repeat the command. Simplest is to turn around 180 degrees and repeat it. A more complex version might repeat it at several angles, not just 180 degrees.

[not required] For the case of multiple dogs potentially hearing it, it would be better to wait a random interval of time before repeating the packet. This will reduce the likelihood multiple dogs repeat it at the same time (or, at least, overlapped with each other).

12. Q19 Rank from most usable (1) to least usable (4).

    A: 4, B: 2, C: 1, D: 3

13. Q20 Identify good and bad aspects of each interface.

Interface	Good	Bad
A		Unwieldy. Human with 10 fingers cannot necessarily operate it with 20 buttons to control. High cognitive load for user to convert dog and command to binary to control.
B	Abstracts away binary, allowing the user to select by dog and command name on wheel. For common case of same dog (or same command), just need to turn command (dog) wheel.	probably big and bulky to acomnodate 4096 commands
C	Abstracts away binary, allowing the user to select by dog and command name on wheel. For common case of same dog (or same command), just need to turn command (dog) wheel. Groups common commands together to reduce time to change commands.	requires battery power, which may be exhausted
D	Abstracts away binary, allowing the user to select by dog and command by tube compnents. For common case of same dog (or same command), just need to change one component.	Unwieldy to carry 4,352 components, or even just 4097 for a single dog

14. **Q21** Another smart phone app (either independently or part of the logic in the smart phone app option above), can convert whistle/bark packets back to logical dog and response payloads. Assuming each robot dog is programmed to acknowledge whistle/bark commands, how does the addition of this response interpreter change the user experience?

    This will improve the user experience. This will allow the user to know when the command was heard by the dog. This help the user know when it is safe to go on to other commands, and help the user avoid sending a command multiple times unintentionally. Generally, it will promote peace of mind knowing what commands were or weren't received.

15. **Q23** All this whistling and barking could get very noisy. How could we keep the basic design with less noise pollution for humans?

    Exploit the fact that humans cannot hear above 25 KHz. So, move all the tones used for robot dog control up above 25 KHz.

16. **Q24** For your solution above and keeping the frequency sample window at 25ms, what is the implication on processing power needed? The answer should be quantitative.

    It will be necessary to sample at a higher rate. Sampling at a higher rate, means more samples in each 25ms window, so more computations per second.

    Nyquist tells us we must sample at twice the highest frequency we want to extract. So, if we operate with usable tones between 25KHz and 50KHz, we will need to sample at 100KHz. Sampling at 100KHz instead of 20KHz, means we will need to perform 5× as many multplications per second as we did when sampling at 20KHz.

    It's possible to tighten this and operate at some lower rate than 50KHz, but the sample rate will still be above 50KHz to accommodate frequencies above 25KHz.

17. **Q25** How could you support the new processing power requirement identified in the previous question?

    We could use more ItsyBitsys. Two would have enough computating capacity, but there would need to be some communication between them.

    alternately

    We could select or design a more powerful processor that was able of performing each multiplication and its associated operations in fewer cycles (20 cycles rather than 100).

alternately

Use a faster procesor – assuming everything speeds up uniformly with the clock speed, 500 MHz instead of 100 MHz.

18. **Q27** How much longer does it take to send a command with the sequential scheme (give a ratio Tsequential/Tsimultaneous)

    We will now need to 5 sequential sets of 4 tones to communicate 20b, instead of one set, so it will take 5 times as long.

19. **Q28** How much less processing power does the sequential version require? (ratio: multiplies-per-second-simultaneous/multiplies-per-second-sequential)

    We will now need to only identify 4 rather than 20 frequencies, so we perform one-fifth the number of multiplies or the simultaneous takes 5 times as many multiplies.

20. **Q31** Assuming you implement the robot dog and the smart phone apps, there are parts of your design that are copyrightable.

    True

21. **Q32** Explain why or why not. (if true, identify what might be copyrightable; if false, detail why no component can be copyrighted.)

    The software code that runs on the microcontroller can be copyrighted. Software code for any UI app would also be copyrightable.

22. **Q33** Ignoring prior art issues, you can patent the description in the first box.

    False

23. **Q34** Why or why not?

    The first box describes an abstract idea or concept without description of how it could be implemented.

24. **Q35** Ignoring prior art issues, if you built a prototype based on your answers to this quiz and demonstrated that it worked, you can patent the design.

    True

25. **Q36** Why or why not?

    You have reduced the idea to practice and can detail the implementation in the patent description.