MCHE 470: Robotics

Fall 2013 – Final Exam

Friday, December 13

Name: Answer Key **CLID:** ____________

Directions: Complete the attached problems making sure to clearly indicate your answer, show your work, and list any assumptions that you have made (with justification for them, if necessary). If you need extra space for any question, you may attach additional sheets of paper.

Academic Honesty (just a reminder):

An essential rule in every class of the University is that all work for which a student will receive a grade or credit be entirely his or her own or be properly documented to indicate sources. When a student does not follow this rule, s/he is dishonest and s/he defeats the purpose of the course and undermines the goals of the University.

______ Problem 1 - 10 Points

a. What is the primary difference between analog and digital signals?

b. You are attempting to read an analog signal that can vary continuously between 0 and 5VDC. Using the 10bit ADC on your RedBoard, what analog signal voltage corresponds to an ADC value of 256?

c. What is the effective output voltage for the three PWM commands shown in Figure 1.1?

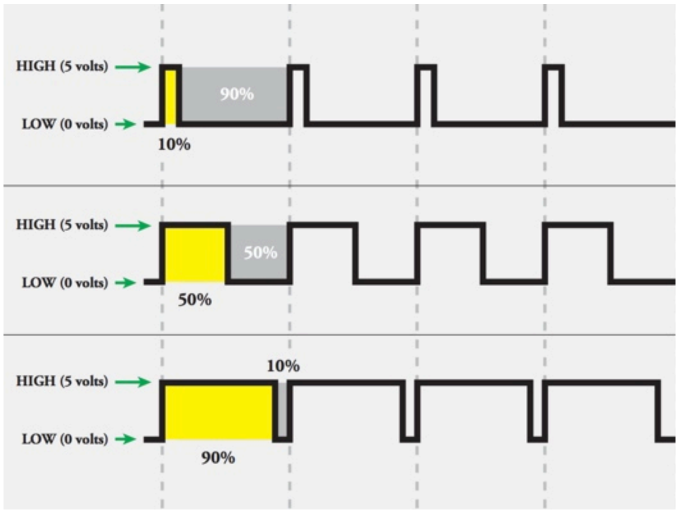


Figure 1.1: PWM Commands
(Image from SparkFun SIK Guide)

a. Analog signals are continuously variable. Digital signals are discrete (on/off, High/Low)

b. Use:

$$\frac{\text{Resolution of ADC}}{\text{System Voltage}} = \frac{\text{ADC value}}{\text{Analog Voltage}}$$

The 10-bit RedBoard ADC has a resolution of 2^10 = 1024. The values range 0-1023.

$$\frac{1023}{5V} = \frac{256}{\text{Analog V}} \rightarrow \boxed{\text{Analog V} = \left(\frac{5}{1023}\right)256 \approx 1.25V}$$

c.

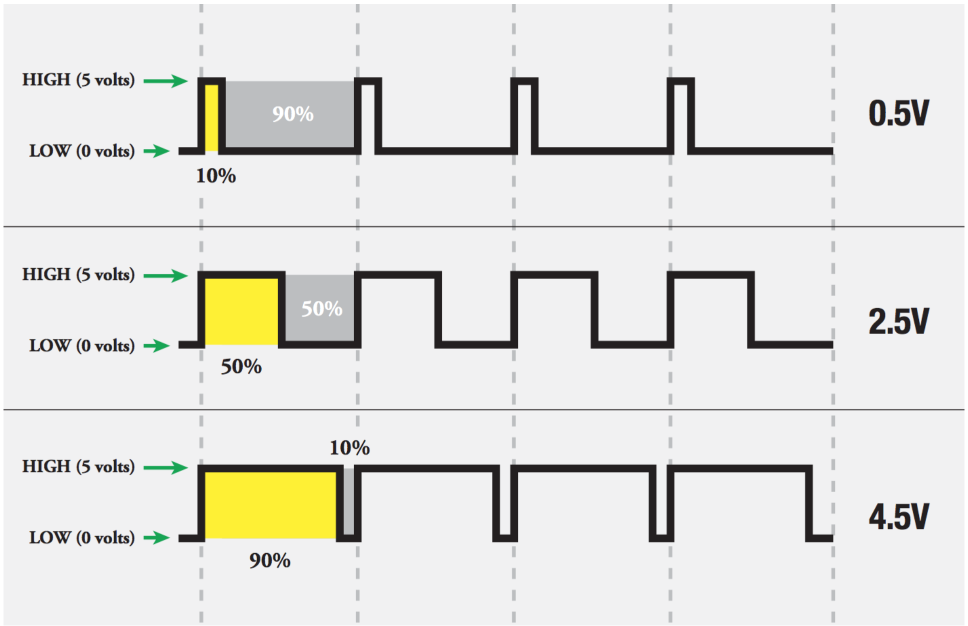


Image from SparkFun SIK Guide

______ Problem 2 - 15 Points

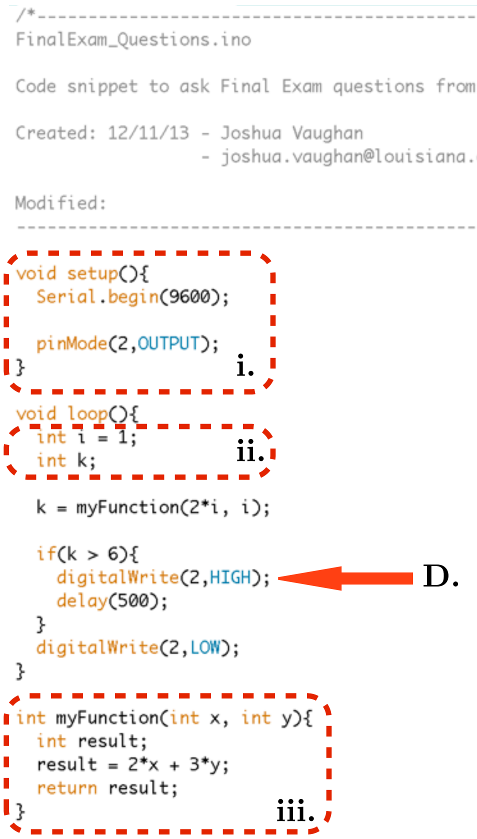
a. What two functions are needed in every Arduino sketch?

b. What is the purpose of each of these functions?

c. What is the purpose/functionality of each of the sections labeled **i.** - **iii.** in Figure 2.1?

d. Does the code marked by **D.** ever run? Why or why not?

e. Write an Arduino sketch (pseudo-code is okay, but be sure to include all the elements of a proper Arduino sketch) that will print a value to the Serial Monitor indicating the condition of two switches. The number printed should equal the number of switches currently pressed. An LED should be lit if both switches are pressed simultaneously.



```
/*------------------------------------------------------------
FinalExam_Questions.ino

Code snippet to ask Final Exam questions from

Created: 12/11/13 - Joshua Vaughan
                  - joshua.vaughan@louisiana.edu

Modified:
-------------------------------------------------------------
void setup(){
  Serial.begin(9600);

  pinMode(2,OUTPUT);
}
void loop(){
  int i = 1;
  int k;

  k = myFunction(2*i, i);

  if(k > 6){
    digitalWrite(2,HIGH);
    delay(500);
  }
  digitalWrite(2,LOW);
}
int myFunction(int x, int y){
  int result;
  result = 2*x + 3*y;
  return result;
}
```

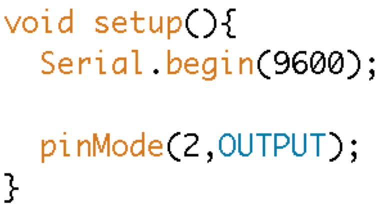
a. setup() and loop()

b. setup() contains an initialization code that must be run before the "main" program starts. It is run only once.

loop() runs continuously once setup() is finished. It is the "main" part of the sketch.

c.

i. This is the setup function.



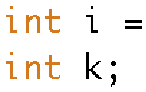
```
void setup(){
  Serial.begin(9600);

  pinMode(2,OUTPUT);
}
```

Initializes serial comm. at 9600 bps

Initializes digital pin 2 as an output

ii.



```
int i = 1;
int k;
```

Declares variable i as an integer and makes it equal to 1

Declares variable k as an integer

Problem 2 (cont.)

c.

iii.
```
int myFunction(int x, int y){
  int result;
  result = 2*x + 3*y;
  return result;
}
```

This is a user-defined/written function (i.e. not setup() or loop()). It takes two integers as input the returns the result of the equation:

2*(the 1st integer input) + 3*(the 2nd integer input)

d.
```
void loop(){
  int i = 1;
  int k;

  k = myFunction(2*i, i);

  if(k > 6){
    digitalWrite(2,HIGH);
    delay(500);
  }
  digitalWrite(2,LOW);
}

int myFunction(int x, int y){
  int result;
  result = 2*x + 3*y;
  return result;
}
```

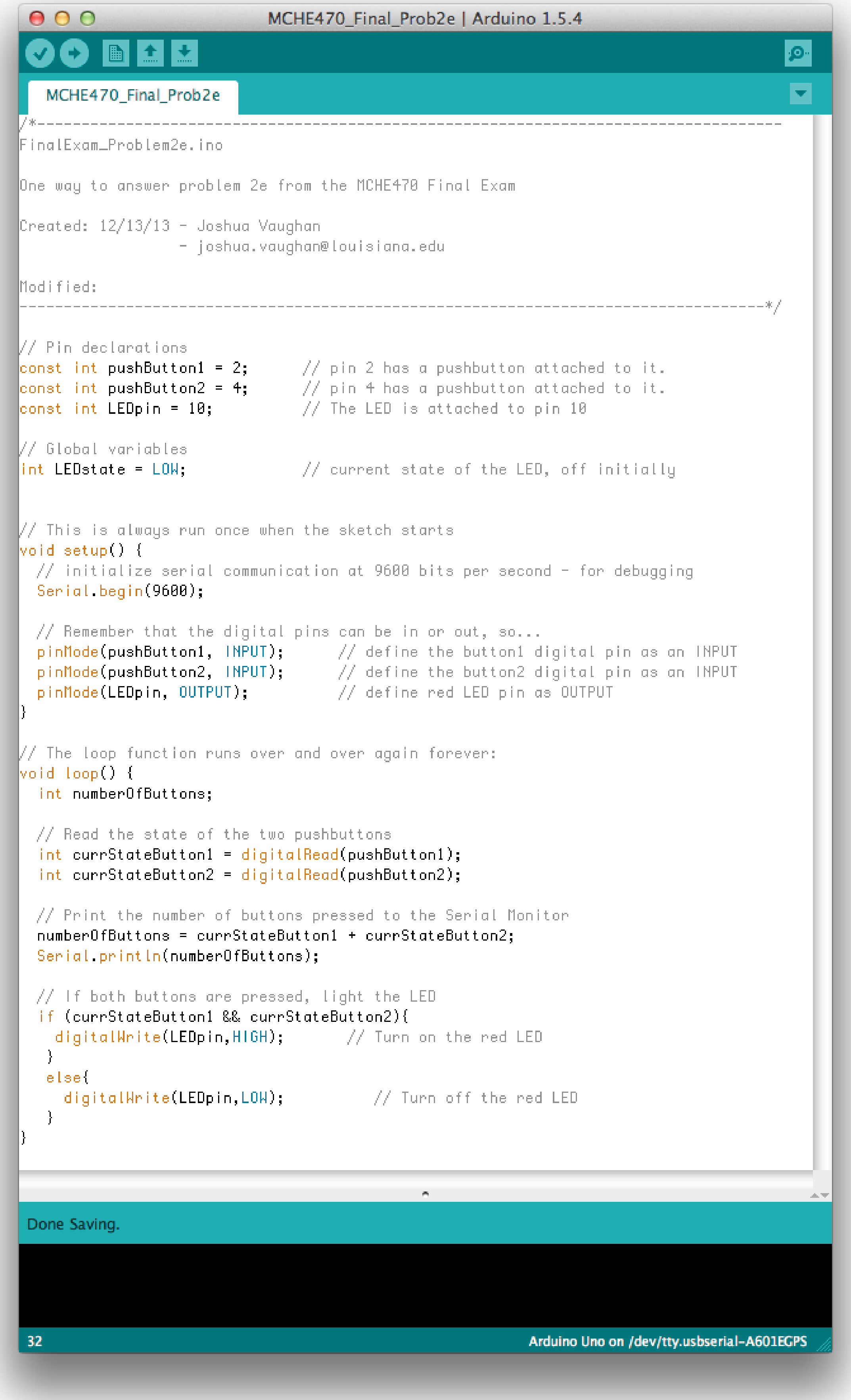
This line calls the function myFunction, with a input-values of 2 and 1. The function therefore returns 7 (2*2 + 3*1). This value is stored in variable k.

This line is the key to whether or not the line marked D (digitalWrite(2,HIGH) gets run. It checks the value of k. If k is greater than 6, then the code inside the loop is run.

So, in this case, the code inside the loop (marked D) in the question is run.

Problem 2 (cont.)

e.



```
/*------------------------------------------------------------------------------------
FinalExam_Problem2e.ino

One way to answer problem 2e from the MCHE470 Final Exam

Created: 12/13/13 - Joshua Vaughan
                  - joshua.vaughan@louisiana.edu

Modified:
------------------------------------------------------------------------------------*/

// Pin declarations
const int pushButton1 = 2;      // pin 2 has a pushbutton attached to it.
const int pushButton2 = 4;      // pin 4 has a pushbutton attached to it.
const int LEDpin = 10;          // The LED is attached to pin 10

// Global variables
int LEDstate = LOW;             // current state of the LED, off initially


// This is always run once when the sketch starts
void setup() {
  // initialize serial communication at 9600 bits per second - for debugging
  Serial.begin(9600);

  // Remember that the digital pins can be in or out, so...
  pinMode(pushButton1, INPUT);      // define the button1 digital pin as an INPUT
  pinMode(pushButton2, INPUT);      // define the button2 digital pin as an INPUT
  pinMode(LEDpin, OUTPUT);          // define red LED pin as OUTPUT
}

// The loop function runs over and over again forever:
void loop() {
  int numberOfButtons;

  // Read the state of the two pushbuttons
  int currStateButton1 = digitalRead(pushButton1);
  int currStateButton2 = digitalRead(pushButton2);

  // Print the number of buttons pressed to the Serial Monitor
  numberOfButtons = currStateButton1 + currStateButton2;
  Serial.println(numberOfButtons);

  // If both buttons are pressed, light the LED
  if (currStateButton1 && currStateButton2){
    digitalWrite(LEDpin,HIGH);       // Turn on the red LED
   }
   else{
     digitalWrite(LEDpin,LOW);          // Turn off the red LED
   }
}
```

Done Saving.

32   Arduino Uno on /dev/tty.usbserial-A601EGPS

_____ **Problem 3 - 10 points**

For the proposed control system in Figure 3.1:

a. What is the primary purpose of each control block (G_{CG} and G_{FB})? What is G_P?

b. Assume G_{FB} is a PID controller:

  i. What is its goal? For full credit, write this goal using an equation expressed in the terms from Figure 3.1.

  ii. Complete Table 3.1, which summarizes the effect of increasing each of the gains on various performance measures.

c. What are some potential problems of the PID controller?

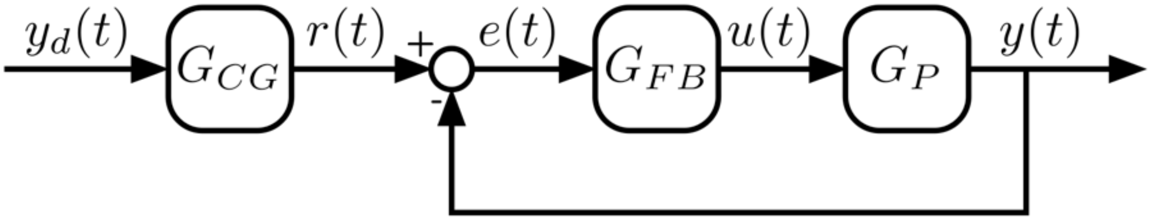


Figure 3.1: Control System Block Diagram

Table 3.1: PID Gain Influence

Gain	Rise Time	Overshoot	Settling Time	Steady-State Error
k_p				
k_d				
k_i				

a. G_CG is the command generation block. It creates a reference signal, r(t), from the desired states of the system, yd(t).

G_FB is a feedback controller. It acts on the error between the current system state and the reference command. Using this error, e(t), an input to the system is generated.

G_P is the "plant". It is the actual system to be controlled.

b.

i. The goal of the feedback controller is to minimize the error between the current states of the system and the reference command (which is often the desired states of the system).To do this, the PID controller generates a input, u(t), to the system based on:

$$u(t) = k_p e(t) + k_i \int e(t)dt + k_d \dot{e}(t)$$

Problem 3 (cont.)

b. ii.

Gain	Rise Time	Overshoot	Settling Time	Steady-State Error
k_p	$\downarrow$	$\uparrow$	$\approx$	$\downarrow$
k_d	$\approx$	$\downarrow$	$\downarrow$	$=$
k_i	$\downarrow$	$\uparrow$	$\uparrow$	$\rightarrow 0$

c. General problems of the PID controller include:

- difficulty sensing the needed states (velocity is often tough to sense)
- noise in the sensing can result in noisy inputs to the system
- potential instability
- incompatibility with human operators

______ Problem 4 - 10 points

a. Briefly answer the questions below:

i. What are some applications of Machine Learning?

ii. What is the difference between supervised and unsupervised learning?

iii. What problem does overfitting training data cause?

b. Given the data in Figure 4.1, explain how to use the k-Nearest Neighbor algorithm to classify the ⚾. (Hint: Use figures to support your explanation.)

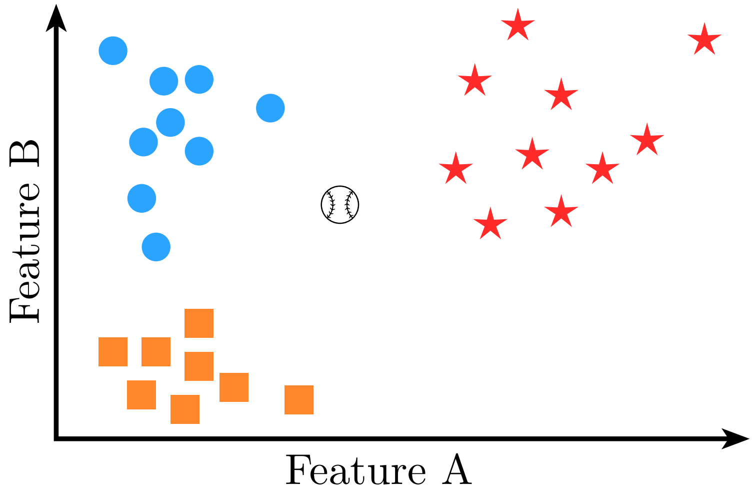


Figure 4.1: Three Categories of Data

a.

i. • Classification

- An object is red and round, is it an apple?
- An object has _______ properties, is it _______?

• Text categorization

- Spam filtering
- Grammar guides
- “Threat” analysis
- Author/Plagiarism detection

• Recommendation engines (iTunes, Netflix, etc.)

• Face recognition

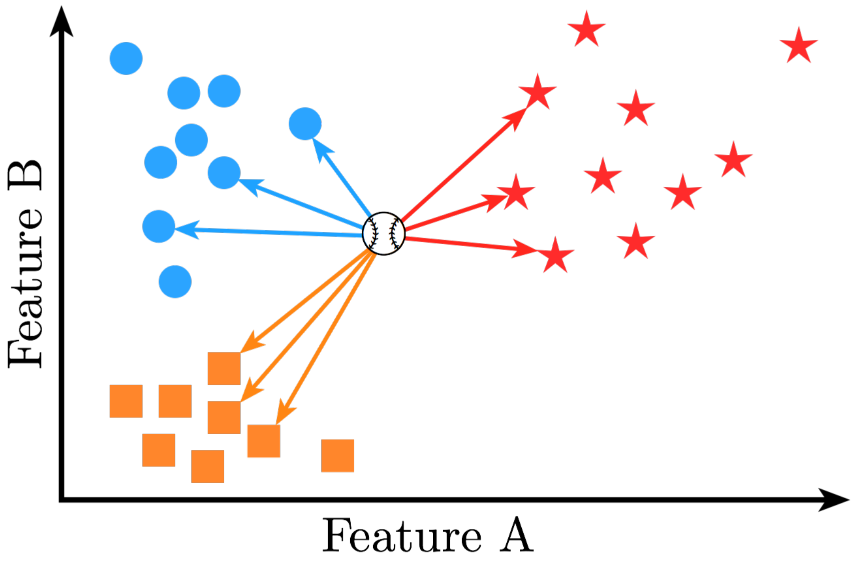
• Election/sports analysis (Moneyball, Nate Silver)

• Diagnostics, medical and otherwise

ii. Supervised learning uses some set of data that has the "truth" and involves human interaction. Unsupervised learning has no such "truth" set of data and has much less human interaction (if any at all once the processing begins)

iii. Overfitting training data can reduce the ability of the developed classification alogrithm to generalize.

Problem 4 (cont.)



b. The simplified k-Nearest Neighbor algorithm:

1. Decide k (k=3 in the figure above)
2. Find the k-Nearest points in each of the possible categories
3. Calculate the distance from the unknown point to each of these
4. The category with the minimum average distance to the test point is the category that it belongs in.

_____ **Problem 5 - 15 points**

Consider the simple mass-spring system in Figure 5.1. In this model, x_d represents the desired location of mass m and is the input to the system.

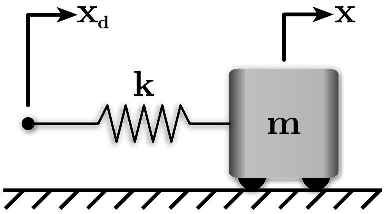


Figure 5.1: A Simple Mass-Spring Model

a. What problem may arise if we don't correctly choose x_d?

b. What are two ways (there are many more) that the vibration of x can be limited?

c. Figure 5.2 shows the impulse response of a flexible system. Where could a second impulse be placed to eliminate vibration in this system?

d. A system can not be driven by impulses. So, how could the impulse sequence from part c. be used to design commands for real systems? Demonstrate this process using the impulse sequence you created in part c of this problem. (Hint: Use a step command as the original vibration-inducing reference command.)

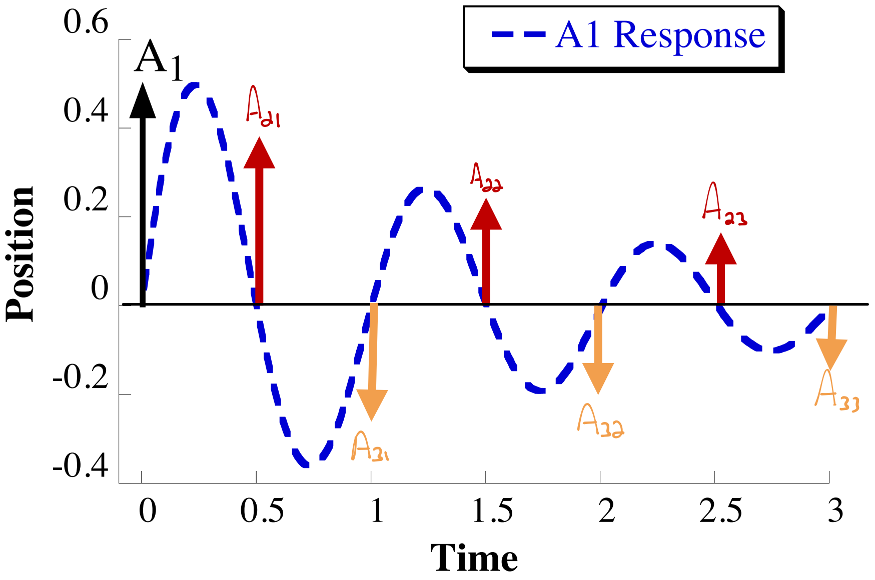


Figure 5.2: Impulse Responses of a Flexible System

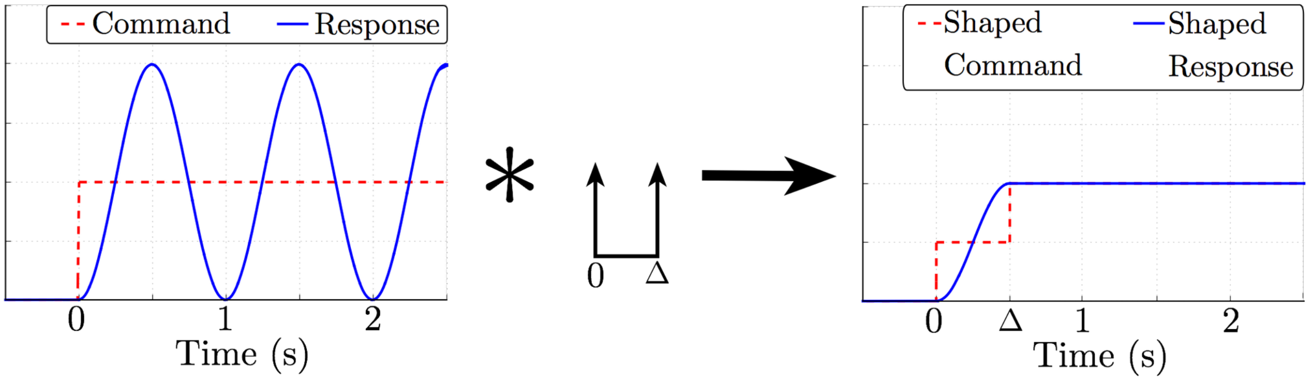
a. Vibration of x(t) (which can lead to many other problems).

b. To limit vibration of x, you can:

- Move very slow
- Add physical damping
- Lowpass filter the input (which is very similar to just moving slower)
- Use input shaping

c. See Figure 4.2 above for three positive-amplitude options, labeled A21, A22, and A23. Any one of these alone could be added to eliminate the vibration. Another option is to apply a negative amplitude impulse at one of A31, A32, or A33.

Problem 4 (cont.)



d. The impulse sequence should be convolved with the original reference command. The vibration-reduction properties of the sequence are transferred to the resulting, shaped command.

______ **Problem 6 - 15 points**

a. Answer the following true/false statements about the design process, by *clearly* circling T or F.

i. **T** | [✓ **F**] – The design process is linear.

ii. **T** | [✓ **F**] – The only customer of a design is the consumer or end-user.

iii. **T** | [✓ **F**] – Once completed, the design tools (House of Quality, Spec. Sheet, etc.) never need to be revisited.

iv. **T** | [✓ **F**] – A Morph Chart lists main, high-level functions and possible ways to fulfill them.

v. **T** | [✓ **F**] – Once a concept is selected and built, the design process is finished.

b. Figure 6.1 shows a portion of a Function Tree. Identify three of the proposed functions that are incorrect (there are more) and why they are so. (Hint: Some of them are examples of a *very* common mistake in Function Trees.)

c. Create the first iteration of a House of Quality for a new 400-level robotics class.

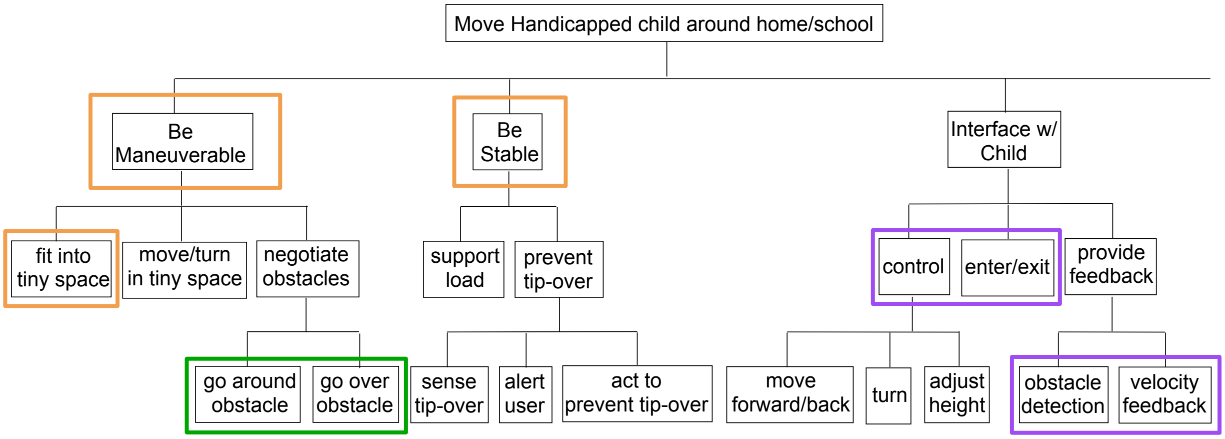


Figure 6.1: A Portion of an Example Function Tree

b. The problem with the items boxed in orange is that they are specifications, not functions.

The problem with the items boxed in purple is that they do not represent things that the machine has to DO.

The problem with the items boxed in green is that they are possible solutions to the function which they support, negotiate obstacles. Go around or go over are potential solutions, not sub-functions.

Problem 6 (cont.)

		Characteristics						
	Importance	Syllabus contents	Curriculum comparison	Student Survey Results	Professor Survey Results	ABET review	Price of implementation	% of material that's hands-on
Customer Requirements								
Creates/promotes interest in robotics	10	◆		■	■			◆
Teaches core robotics concepts	8	■	■			■		◆
Uses robotics as "Trojan Horse"	6	■		◆	◆	◆	●	■
Is fun	10		◆	■	■		◆	■
Addresses "real world" issues	7	■	■	●	■	●	◆	■
Still allows professor time for research	5	■		◆	■			■
Fills 3 hours of class time per week	7	■						■
Is promotable outside the dept.	6	■	◆	■	◆	●	●	■
Includes some basic programming	8	■		◆	◆	◆		◆
Fits into department curriculum	6	■		●	■	◆		◆
Costs are inline with "normal" class	9	■	●		●		■	■
Contains hands-on component	9	■		■	■			■

Relationship

Strong	■
Medium	◆
Weak	●

______ **Problem 7 - 15 points**

Your boss has asked your team to design a command-generation algorithm for a system that can be represented by the block diagram in Figure 7.1. Your goal is to make the system output, *y(t)*, track the desired trajectory, $y_d(t)$, by properly choosing the contents of the $\boxed{?}$ block.

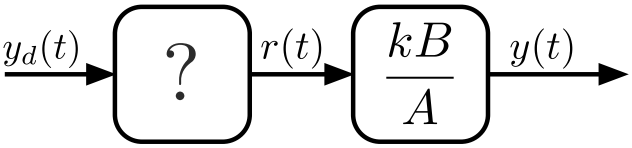


Figure 7.1: A Tracking Problem

a. Your co-worker suggests using plant/model inversion. Using the terms in Figure 7.1, explain his idea and why he might think it would work.

b. What are the problems with this plant/model inversion idea?

c. Suggest and *thoroughly* explain an alternative.

d. If $y_d(t)$ is a periodically-repeated trajectory, how could the tracking algorithm be further improved?

a. If $G_p = \frac{kB}{A}$ then the coworker is proposing filling $\boxed{?}$ with $1/G_p$ or $\frac{A}{kB}$

will result in:

$$r(t) = \frac{A}{kB} y_d(t)$$

$$y(t) = \left(\frac{\cancel{kB}}{\cancel{A}}\right)\left(\frac{\cancel{A}}{\cancel{kB}} y_d(t)\right)$$

so

$$y(t) = y_d(t) \leftarrow$$

The output exactly matches the desired state.

b. Problems with plant/model inversion include:

- The resulting reference command can be one that is not achievable by the system.
- The model will never match the system exactly, so the exact cancellation above will never happen in practice
- For nonminimum-phase systems, the RHP zeros will become RHP poles, creating an unstable system input.

Problem 7 (cont.)

c. There are many. A "direct" alternative is the Zero Phase Error Tracking Controller (ZPETC). It is a partial model inversion, only inverting the parts of the system that will not cause the problems outlined in the part b.

An input shaper may also improve the tracking, particularly for a vibratory system. The input shaper will remove vibration, reducing the tracking error.

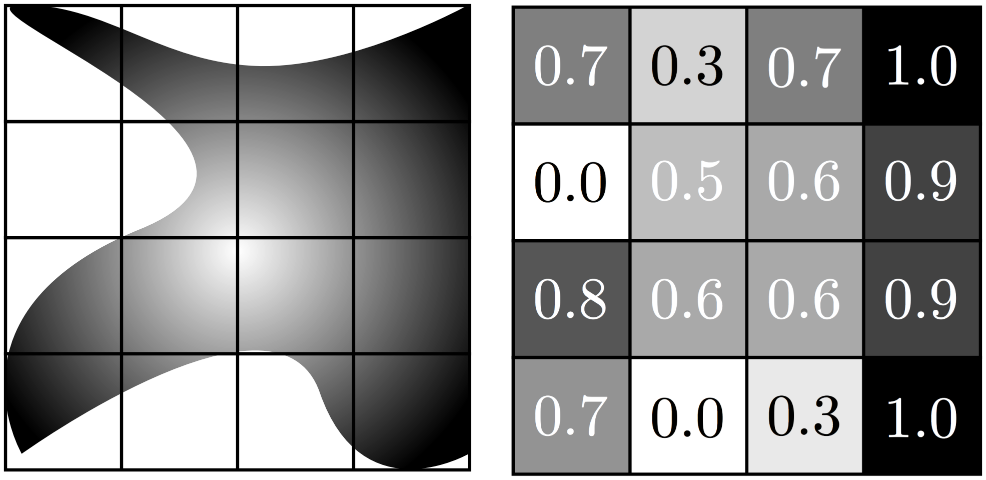
d. If the trajectory is repeated, then error between sequential reptetions of the trajectory can be used in a learning controller. The reference command for the next cycle becomes the reference command from the current cycle plus some corrective term based on the output error.

______ **Problem 8 - 10 points**

a. What are the three main components that influence the exposure of an image? Explain how each influences the exposure.

b. How would the image in Figure 8.1 be processed by a vision sensor with the resolution of the grid overlay? (*i.e.* What would the value of each "pixel" be and/ or what matrix could represent this image?)

c. What is the function the the box filter in Figure 8.2? How does it do this?



0.7	0.3	0.7	1.0
0.0	0.5	0.6	0.9
0.8	0.6	0.6	0.9
0.7	0.0	0.3	1.0

Figure 8.1: A Simple Image and Sensor Array

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Figure 8.2: A Simple Box Filter

a.
- ISO: How sensitive the sensor is to light
- Aperture: How big the lens opening is
- Shutter Speed: How long the shutter stays open

b. See the image above. Each cell/pixel is given a value that best approximate its contents. The numbers inside the cells are a low precision matrix approximation of the image, with each cell given an entry in the 4x4 matrix.

c. This box filter will blur the image. It assigns each pixel a new value based on the average of it and the surrounding pixels (itself and 8 surrounding pixels, in this case).

Possibly Useful Equations and Information

$$\bar{f} = m\bar{a}$$

$$I_o\bar{\alpha} = \sum \bar{M}_o$$

$$\sin(a \pm b) = \sin(a)\cos(b) \pm \cos(a)\sin(b)$$

$$\cos(a \pm b) = \cos(a)\cos(b) \mp \sin(a)\sin(b)$$

$$x(t) = ae^{i\omega_n t} + be^{-i\omega_n t}$$

$$x(t) = a\cos\omega_n t + b\sin\omega_n t$$

$$x(t) = e^{-\zeta\omega_n t}\left[a\cos(\omega_d t) + b\sin(\omega_d t)\right]$$

$$e^{\pm i\omega t} = \cos(\omega t) \pm i\sin(\omega t)$$

$$\int u\ dv = uv - \int v\ du$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$\delta_{oc}V \equiv \forall \sum$$

$$\begin{bmatrix} A_i \\ t_i \end{bmatrix} = \begin{bmatrix} \frac{1}{1+K} & \frac{K}{1+K} \\ 0 & \frac{\tau_d}{2} \end{bmatrix}$$

$$\begin{bmatrix} A_i \\ t_i \end{bmatrix} = \begin{bmatrix} \frac{1}{1+2K+K^2} & \frac{2K}{1+2K+K^2} & \frac{K^2}{1+2K+K^2} \\ 0 & \frac{\tau_d}{2} & \tau_d \end{bmatrix}$$

$$K = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}}$$

$$\omega_n = \sqrt{\frac{k}{m}} \qquad \frac{c}{m} = 2\zeta\omega_n \qquad f = \frac{\omega_n}{2\pi}$$

$$\frac{\text{Resolution of ADC}}{\text{System Voltage}} = \frac{\text{ADC value}}{\text{Analog Voltage}}$$

$$i \equiv \sqrt{-1}$$

$$2 + 2 = 2 \times 2 = 2^2 = \texttt{0b0010} = \texttt{0x2}$$

$$V(\omega,\zeta) \equiv e^{-\zeta\omega t_n}\sqrt{[C(\omega,\zeta)]^2 + [S(\omega,\zeta)]^2}$$

$$C(\omega,\zeta) = \sum_{i=1}^{n} A_i e^{\zeta\omega t_i}\cos(\omega t_i\sqrt{1-\zeta^2})$$

$$S(\omega,\zeta) = \sum_{i=1}^{n} A_i e^{\zeta\omega t_i}\sin(\omega t_i\sqrt{1-\zeta^2})$$

$$x(t) = \int_0^t f(\tau)h(t-\tau)d\tau$$

$$x(t) = \int_0^t f(t-\tau)h(\tau)d\tau$$

$$f(t) = \sum_{n=0}^{\infty} a_n\cos(n\omega_0 t) + \sum_{n=1}^{\infty} b_n\sin(n\omega_0 t)$$

$$a_n = \frac{\omega_0}{\pi}\int_0^{\frac{2\pi}{\omega_0}} f(t)\cos(n\omega_0 t)dt$$

$$b_n = \frac{\omega_0}{\pi}\int_0^{\frac{2\pi}{\omega_0}} f(t)\sin(n\omega_0 t)dt$$

$$a_0 = \frac{\omega_0}{2\pi}\int_0^{\frac{2\pi}{\omega_0}} f(t)dt$$