Exam 1 - Wed. 10/4/23, 7:00pm-9:00pm

Name: Aidan Chin Student ID #: 33803321

NOTES:

- You **MUST** start RPNow first, i.e., before you download the exam on Moodle.
- You **MUST** be able to receive email notifications, and you should read anything you receive from Prof. Duarte, in case he needs to make a broadcast announcement about the test or contact you individually about something you are doing.
- You may visit only **STATIC** web pages. You may NOT use Chegg or any interactive site.
- If you are running out of time generating your PDF submission, email Prof. Duarte **BEFORE** your time runs out. (You should not need to email any files to Prof. Duarte.)

This exam consists of three parts on four pages, including this page. Part 1 is worth 60 points. Part 2 is worth 30 points. Part 3 is worth 110 points. You should spend more than half of your time on Part 3. Therefore, you might want to do as much of Parts 1 and 2 as you can quickly, then move on to Part 3. You can go back to Parts 1 and 2 once you are finished with Part 3.

You are required to use RPNow during this exam. You must have a functioning microphone and a functioning webcam. You must complete the authentication process before you can start the exam, so try to get settled before the exam opens and begin the authentication. This includes a Microphone Check, an ID check, a Room Scan, and a User Photo. (You can pre-pay for the service and practice the authentication using "Computing Exam" in RPNow.) If you have used RPNow for another course, or for another version of ECE 202, you must use a different email address, so that the system prompts you to pay another $40.

During the exam, you *must* choose "Computing Exam".

You may not call, text, email, or otherwise communicate with anyone other than Prof. Duarte.
If you have a question during the exam, you should email Prof. Duarte.

You may use your notes from class, your old scripts, and lecture notes. You may use the internet to search for answers to any questions you might have, but you may NOT use any interactive resources such as Live Chat or ChatGPT. However, keep in mind that it is very inefficient to look up answers to questions. It is much better to have prepared completely, as time is limited.

You may not use headphones. You may have music playing in the background.

The time limit is 2 hours. The exam will become available at 7:00pm on Wednesday, 10/4/23, and it will close at 9:15pm to allow you time to collect and submit your work as a single PDF file via Gradescope.

Part 1. (60 points)

Consider the following MATLAB script, and use it for all questions in this part.

```
Marco Duarte
9/3/23
ECE 202, Exam 1
Predicting the final velocities of two carts that collide elastically
clear
----- givens -----
m1 = 100;
m2 = 200;
total_M = 300; % (total mass of m1 and m2)
v1i = 40;
v2i = 20;
----- calculation -----
answer to part (a)...
cart1v_final = ((m1 - m2)*v1i) / (total_M)+((2 m2)*20) / (total_M);
answer to part (b)...
cart2v_final = ((2*m1)*v1i)/(total_M) + ((m2 - m1)*20)/(total_M);
----- check momentum conservation -----
p0 = (m1*v1i) + (m2*v2i);
pf = m1*cart1v_final + m2*cart2v_final;
pcheck = pf - p0 % (p0 should be equal to pf)
```

For questions 1–4, you need to "fix" certain lines of code, as listed below. By "fix", I mean that you need to think about: (1) how to get the script to compile properly, (2) how to make the script robust and efficient, (3) how to make the script relatively easy to read and understand, and (4) how to make sure the output is appropriate for passing in, according to the guidelines you have been given so far in this course. In each case, re-write the given code as "fixed", then describe what's wrong and everything you changed. NOTE: There can be more than one "mistake" in each line of code. For full credit, you must explain everything that is wrong.

(a) Fix line 8 (and explain what you fixed):

```
total_M = 300;   (total mass of m1 and m2)
```

M = m1+m2; % total mass of m1 and m2 in grams
added %, made variable name more consise
made dependent on variables instead of hard coded
added units, assumed SI units

(b) Fix line 10 (and explain what you fixed):

```
v1i = 40;
```

v1i = 40; % initial velocity of cart 1 in m/s
added units and description of variable assumed SI units

(c) Fix line 12 (and explain what you fixed):

```
----- calculation -----
```

% ------ Calculation --------
added % to make a comment, capatalized to make neater

(d) Fix line 14 (and explain what you fixed):

```
cart1v_final = ((m1 - m2)*v1i) / (total_M)+((2 m2)*20) / (total_M);
```

v1f = ((m1-m2)*v1i + 2 * m2 * 20) / M; % calculate resultant velocity of cart 1 in m/s
added describing note, simplified expression, changed variable total_M to M, changed cart1v_final to simply v1f, removed unnessessary (), added nessessary ()
added units

(e) What is missing from near the beginning of the script? (HINT: As written, the script would not run.)

% on each line is missing making the text at the top a comment

(f) Is the momentum check appropriate? Why or why not? If not, rewrite the momentum check.

the momentum check is missing a few things
18 pi = m1*v1i + m2*v2i; % calculate initial momentum
19 pf = m1*v1f + m2*v2f; % calculate resultant momentum
20 pcheck = pf - pi % find difference of momentum(pi should be equal to pf, output 0)
21 %checks concervation of momentum
changed p0 to pi to remain in pattern, removed unnessesary () added expected output,
added describing comments

Part 2. (30 points)

Consider the following lines from a MATLAB script, starting in line 7...

```
7   r = [10 20 30 40 50]
8   v = [12 8 4 2 1]
9
10  i = v / r;   (current in each resistor)
11  p = r * i * i;   (total energy dissipated per resistor)
12
13  ptot = m(1) + m(2) + m(3) + m(4) + m(5);   (total dissipated energy)
```

Like Part 1, you need to think about how to get these lines to compile properly, how to make them robust and efficient, how to make them relatively easy to read and understand, and how to make sure the output is appropriate for passing in, according to the guidelines you have been given so far in this course. Write the "fixed" code in the text boxes below, and explain everything you fixed and why. As before, there can be more than one "mistake" in each line of code. For full credit, make sure to explain why something is wrong.

(a) Fix line 8 (and explain what you fixed):

```
v = [12 8 4 2 1]
```

v = [12 8 4 2 1]; % voltages across corresponding resistors in V
added ; to suppress output, added comment describing what the variable is, added units

(b) Fix line 11 (and explain what you fixed):

```
p = r * i * i;   (total energy dissipated per resistor)
```

p = r .* i^2; % (total energy dissipated per resistor)
added % to make the comment actually a comment
made i*i i^2 to make simpler
made r*i^2 to r.*i^2 because matrix needs .* operation

(c) Fix line 13 (and explain what you fixed):

```
ptot = m(1) + m(2) + m(3) + m(4) + m(5);   (total dissipated energy)
```

ptot = sum(m); % (total dissipated energy)
added % to make comment actually a comment
exchanged for sum() more efficient way to write that expression

Part 3. (110 points) Writing your own scripts. Note that there is a base problem (a) and additional subproblems (b), (c), (d) that build upon each other and will increase your score. You only need to submit one script and its output.

Consider first the identity: $\sin(3\theta) = 3\sin\theta - 4\sin^3\theta$.

Write a short MATLAB script to show that $\sin(3\omega t)$ is equal to $3\sin(\omega t) - 4\sin^3(\omega t)$.

In other words, construct two functions in the range $-\pi/\omega \le t \le \pi/\omega$, then create a check that is one number, output to the Command Window, that convincingly shows that this identity is true, point by point.

Make sure the script is efficient, combining expressions and avoiding intermediate variables where possible. In your script, use lower-case "w" to represent the angular frequency ω.

Print your M file and the command window output to PDFs and append them into this PDF. (If you can't make PDFs, instead copy/paste the text from your script/.m file into the corresponding box on page 6; there will be a penalty for missing PDF files)

Clear the Command Window before your last run. Print your script/.m file and Command Window to PDF, then append them into this PDF. (If you can't make PDFs, instead copy/paste the text from your script/.m file and the text from your Command Window into the corresponding boxes on page 7; there will be a 10-point penalty for missing PDF files).

(a) (75 points) Set the angular frequency $\omega = \pi$ radians/second as a fixed value in your script. You can simplify the math involved if you like.

(b) (15 points) Plot these two functions on one figure with one pair of axes; use a continuous line for the first plot and a dashed line for the second, with different colors, so that we can see that the two lines fall on top of one another. Make sure there is a proper legend, a meaningful title, and complete axis labels on your figure. Make sure that the legend does not overlap with the plot by leaving enough space for it. Make sure the font sizes are appropriate to the size of the figure. Use at least 1000 points in time.

Create a JPG or PNG file of the figure, e.g., using File | Save As... .PNG or .JPG, then append them to this PDF. You may also take screen shots of the figures with a small penalty, if that is easier for you to manage. Include the plotted functions in the legend of the figure.

(c) (10 points) Instead of hard-coding the value of ω, ask the user to set ω. Make sure that the selected value is reflected in the time axis of the plot. Execute the script while setting $\omega = 2\pi$.

(d) (10 points) When we make ω large, the range of times being plotted may be very small. Using IF/ELSE statements, set the plot so that if its time range is smaller than [-0.1 sec, 0.1 sec], then the time axis unit is milliseconds. The axis label should state the corresponding units. Execute the script while entering $\omega = 600$ (note that there's no π there on purpose).

Part 3 Script/.m file

Part 3 Command Window Output

```matlab
%{
Aidan Chin Midterm 1 part 3 A 10/4/23
ECE 202 MATLAB Midterm 1
the goal of this script is to show that sin(3wt) is equal to 3sin(wt)
-4sin^3(wt)
%}

% *** Prepare workspace ***
clear % clear variables to remove chance of error
clf % clear figures to make the graph window clear

% *** Givens ***
w = pi; % angular momentum in radians/second
tmin = -pi/w; % minimum t value
tmax = pi/w; % maximum t value
N = 1000; % number of steps to be made between min and max
t = linspace(tmin,tmax,1+N);%create array of numbers between tmin
                                                              % and tmax

% *** Calculation ***
a = w.*t; % define a so avoid writing many times later
f1 = sin(3.*a); % given sine formula
f2 = 3*sin(a) - 4*sin(a).^3; % given sine subtraction formula

% *** Check ***
d = abs(f2) - abs(f1); % check to make sure the 2 formulas are equivalent
check = sum(d) % print single number check, should be close to 0

% the reason we use abs intead of soley sum is because you want to find the
% distance of the 2 graphs from eachother, and distance is always positive

% *** Graphing ***
plot(t,f1,t,f2,':','LineWidth',3) %plot both formulas on graph with f1 being
%solid line, f2 being dotted and linewidth of 3
title('ECE 202 Midterm 1 Part 3 B', ...
    'Proving that sin(3wt) \equiv 3sin(wt) - 4sin^3(wt)')
%make title and description of figure
legend('f1 = sin(3wt)','f2 = 3sin(wt) - 4sin^3(wt)') % make legend for figure
ylabel('Angular Momentum (radians/second)')% change y axis label
xlabel('Time (s)') % change x axis label
ylim([-1.5,1.5]) % modify the y axis limits so no overlap on legend
grid on % enable the grid on the graph
```

```
>> MIDTERM1

check =

   2.6753e-14

>>
```

```matlab
%{
Aidan Chin Midterm 1 part 3 B 10/4/23
ECE 202 MATLAB Midterm 1
the goal of this script is to show that sin(3wt) is equal to 3sin(wt)
-4sin^3(wt)
%}

% *** Prepare workspace ***
clear % clear variables to remove chance of error
clf % clear figures to make the graph window clear

% *** Givens ***
w = pi; % angular momentum in radians/second
tmin = -pi/w; % minimum t value
tmax = pi/w; % maximum t value
N = 1000; % number of steps to be made between min and max
t = linspace(tmin,tmax,1+N);%create array of numbers between tmin
                                                              % and tmax

% *** Calculation ***
a = w.*t; % define a so avoid writing many times later
f1 = sin(3.*a); % given sine formula
f2 = 3*sin(a) - 4*sin(a).^3; % given sine subtraction formula

% *** Check ***
d = abs(f2) - abs(f1); % check to make sure the 2 formulas are equivalent
check = sum(d) % print single number check, should be close to 0

% the reason we use abs intead of soley sum is because you want to find the
% distance of the 2 graphs from eachother, and distance is always positive

% *** Graphing ***
plot(t,f1,t,f2,':','LineWidth',3) %plot both formulas on graph with f1 being
%solid line, f2 being dotted and linewidth of 3
title('ECE 202 Midterm 1 Part 3 B', ...
    'Proving that sin(3wt) \equiv 3sin(wt) - 4sin^3(wt)')
%make title and description of figure
legend('f1 = sin(3wt)','f2 = 3sin(wt) - 4sin^3(wt)') % make legend for figure
ylabel('Angular Momentum (radians/second)')% change y axis label
xlabel('Time (s)') % change x axis label
ylim([-1.5,1.5]) % modify the y axis limits so no overlap on legend
grid on % enable the grid on the graph

```

```
>> MIDTERM1

check =

   2.6753e-14

>>
```

```
%{
Aidan Chin Midterm 1 part 3 C 10/4/23
ECE 202 MATLAB Midterm 1
the goal of this script is to show that sin(3wt) is equal to 3sin(wt)
-4sin^3(wt)
%}

% *** Prepare workspace ***
clear % clear variables to remove chance of error
clf % clear figures to make the graph window clear

% *** Givens ***
w = input('Input value of w: '); % angular momentum in radians/second
tmin = -pi/w; % minimum t value
tmax = pi/w; % maximum t value
N = 1000; % number of steps to be made between min and max
t = linspace(tmin,tmax,1+N);%create array of numbers between tmin
                                                              % and tmax

% *** Calculation ***
a = w.*t; % define a so avoid writing many times later
f1 = sin(3.*a); % given sine formula
f2 = 3*sin(a) - 4*sin(a).^3; % given sine subtraction formula

% *** Check ***
d = abs(f2) - abs(f1); % check to make sure the 2 formulas are equivalent
check = sum(d) % print single number check, should be close to 0

% the reason we use abs intead of soley sum is because you want to find the
% distance of the 2 graphs from eachother, and distance is always positive

% *** Graphing ***
plot(t,f1,t,f2,':','LineWidth',3) %plot both formulas on graph with f1 being
%solid line, f2 being dotted and linewidth of 3
title('ECE 202 Midterm 1 Part 3 B', ...
    'Proving that sin(3wt) \equiv 3sin(wt) - 4sin^3(wt)')
%make title and description of figure
legend('f1 = sin(3wt)','f2 = 3sin(wt) - 4sin^3(wt)') % make legend for figure
ylabel('Angular Momentum (radians/second)')% change y axis label
xlabel('Time (s)') % change x axis label
ylim([-1.5,1.5]) % modify the y axis limits so no overlap on legend
grid on % enable the grid on the graph
```

```
%{
Aidan Chin Midterm 1 part 3 C 10/4/23
ECE 202 MATLAB Midterm 1
the goal of this script is to show that sin(3wt) is equal to 3sin(wt)
-4sin^3(wt)
%}

% *** Prepare workspace ***
clear % clear variables to remove chance of error
clf % clear figures to make the graph window clear

% *** Givens ***
w = input('Input value of w: '); % angular momentum in radians/second
tmin = -pi/w; % minimum t value
tmax = pi/w; % maximum t value
N = 1000; % number of steps to be made between min and max
t = linspace(tmin,tmax,1+N);%create array of numbers between tmin
                                                              % and tmax

% *** Calculation ***
a = w.*t; % define a so avoid writing many times later
f1 = sin(3.*a); % given sine formula
f2 = 3*sin(a) - 4*sin(a).^3; % given sine subtraction formula

% *** Check ***
d = abs(f2) - abs(f1); % check to make sure the 2 formulas are equivalent
check = sum(d) % print single number check, should be close to 0

% the reason we use abs intead of soley sum is because you want to find the
% distance of the 2 graphs from eachother, and distance is always positive

% *** Graphing ***
plot(t,f1,t,f2,':','LineWidth',3) %plot both formulas on graph with f1 being
%solid line, f2 being dotted and linewidth of 3
title('ECE 202 Midterm 1 Part 3 B', ...
    'Proving that sin(3wt) \equiv 3sin(wt) - 4sin^3(wt)')
%make title and description of figure
legend('f1 = sin(3wt)','f2 = 3sin(wt) - 4sin^3(wt)') % make legend for figure
ylabel('Angular Momentum (radians/second)')% change y axis label
xlabel('Time (s)') % change x axis label
ylim([-1.5,1.5]) % modify the y axis limits so no overlap on legend
grid on % enable the grid on the graph
```

```matlab
%{
Aidan Chin Midterm 1 part 3 C 10/4/23
ECE 202 MATLAB Midterm 1
the goal of this script is to show that sin(3wt) is equal to 3sin(wt)
-4sin^3(wt)
%}

% *** Prepare workspace ***
clear % clear variables to remove chance of error
clf % clear figures to make the graph window clear

% *** Givens ***
w = input('Input value of w: '); % angular momentum in radians/second
tmin = -pi/w; % minimum t value
tmax = -1 * tmin; % maximum t value
N = 1000; % number of steps to be made between min and max
t = linspace(tmin,tmax,1+N);%create array of numbers between tmin
                                                              % and tmax

% *** Calculation ***
a = w.*t; % define a so avoid writing many times later
f1 = sin(3.*a); % given sine formula
f2 = 3*sin(a) - 4*sin(a).^3; % given sine subtraction formula

% *** Check ***
d = abs(f2) - abs(f1); % check to make sure the 2 formulas are equivalent
check = sum(d) % print single number check, should be close to 0

% the reason we use abs intead of soley sum is because you want to find the
% distance of the 2 graphs from eachother, and distance is always positive

% *** Graphing ***
if tmin <= -.1 %check if time values are larger than .1 to see if should be in ms
    plot(t,f1,t,f2,':','LineWidth',3) %plot both formulas on graph with f1 being
    %solid line, f2 being dotted and linewidth of 3
    xlabel('Time (s)') % change x axis label
else
    tms = t.*1000;
    plot(tms,f1,tms,f2,':','LineWidth',3) %plot both formulas on graph with f1 being
    %solid line, f2 being dotted and linewidth of 3
    xlabel('Time (ms)') % change x axis label
end

title('ECE 202 Midterm 1 Part 3 D', ...
    'Proving that sin(3wt) \equiv 3sin(wt) - 4sin^3(wt)')
%make title and description of figure
legend('f1 = sin(3wt)','f2 = 3sin(wt) - 4sin^3(wt)') % make legend for figure
ylabel('Angular Momentum (radians/second)')% change y axis label
ylim([-1.5,1.5]) % modify the y axis limits so no overlap on legend
grid on % enable the grid on the graph

```

```
>> MIDTERM1
Input value of w: 600

check =

   5.7697e-15

>>
```