ESC204 – Skill-Building Assignment

(Created in respond to the COVID-19 situation)

Due: April 9th (2020-04-09) 22:00 on Quercus	**Weight:** 10% individual

Introduction

This individual assignment aims to provide students an opportunity to develop a specific engineering design skill of your choice, which could benefit you in your future engineering career.

You could choose 1 out of the 11 tasks provided below. We are also open to you creating your own tasks with similar level of difficulty. Please submit a description (in similar format as presented in this document, include a description, submission type, and what you think the expectations should be) on Discourse for approval (on a rolling basis).

You may have already completed some components of these tasks during your design process, which you are welcome to finish and submit, but what you submit must be **YOUR OWN** work.

Submission

The submission should consist a written summary of up to 5 pages describing the task you completed and the results. An additional video (up to 5 minutes) file could be included (not required) if it helps with demonstrating the results of the task. In the detailed description of each task below, there is a section outlining the suggested submission type.

Your effort on this assignment should be focused on doing the actual task rather than the written component. Thus, we are asking for a "written summary" for submission instead of a detailed "report".

The grading of this assignment consists three levels: below, meet, and exceed expectation, transferring to 30%, 60% and 100% of your grade. Each task has included a very brief description of the expectation for each level as a guideline.

You will be given two opportunities to submit this individual assignment, so you could receive feedback on the first submission and improve for the second one. If you'd like to take this opportunity, the first (draft) submission is due on **2020-04-05 22:00**. The second (final) submission is due on **2020-04-09 22:00**.

Supports

Some tasks require additional documents which can be found on Quercus.

If you have any questions about specific tasks, please post them on Discourse. A sub-topic has been created under which each task has an individual thread.

TA office hours will be provided through BB Collaborate. The schedule for each TA has been posted on Quercus.

CONTENTS

1 MECHANICAL

1.1 ENGINEERING DRAWING

Primary Support/Grading TAs: Francisco and Chris

Description

Submit proper engineering drawings of your current project and/or custom components. The engineering drawings should be clear such that there should be only one interpretation for manufacturing the specific component. At least 2 critical/custom components should be submitted separately, and an additional full assembly drawing should be submitted as well. Each component submission should be an orthographic multi-view drawing. For each component, the following information is required on the page:

- Datums/References
- Material used for part
- Fabrication steps for part (I.e. Cut part out from wood, 3d print components, laser cut from a sheet of steel)
- Critical length/measurements
- Tolerances
- Assembly steps required (I.e. join parts A and B with screws, Align parts B and C along an edge)
- Proper labelling of holes
- Surface finishes
- Proper sectioning for critical interior details

Fabrication and assembly should be concise and detailed but does not need specific values.
For assembly drawings, an exploded view should be submitted along with a Bill of Materials.
For guidance and reference please follow these links:
Component Drawings: https://ocw.mit.edu/courses/mechanical-engineering/2-007-design-and-manufacturing-i-spring-2009/related-resources/drawing_and_sketching/
Exploded View Drawings: https://www.designingbuildings.co.uk/wiki/Exploded_view

Submission Type

Submission Type 1 (Computer generated Engineering drawings): Submit three separate components on individual pages and then one drawing of the fully assembled robot.
Submission Type 2 (Hand Drawn): Submit four component engineering drawings.

Engineering drawings should follow a similar format found on SolidWorks Engineering Drawing Sheet Templates. Should the page not be enough to include any information, you may use a page of the submission as a supplementary page. If you use the supplementary page, please use proper referencing to specific parts of the supplementary page in the engineering drawing descriptions.

Expectation

Below Expectations	Meets Expectations	Exceeds Expectations
Submission is missing critical/custom component drawings Submitted drawing lacks information such that the part cannot be duplicated Drawing is sloppy and unclear	At least 3 engineering drawings are submitted Submissions are drawn to appropriate scale and specifications and can be used to replicate component	At least 4 drawings are submitted Submitted drawings have detailed information for each component

	Proper engineering drawing template is used	Components have appropriate sectioning for critical interior details

1.2 Computer Aided Design (Conceptual): Robot Arm Design

Primary Support/Grading TAs: Francisco and Chris

Description

Submit a robot arm design to accomplish the task of plugging in the charger into the fixture. Use the same dimensions as the project proposal. The robotic arm design needs to be able to theoretically plug the charger in at least 6 of the 9 locations identified in Milestone 1 and 2 and cannot have a mobile base, you do not need to account for z variation and yaw angles. The design should be made through a CAD software of your choice (e.g. SolidWorks, AutoDesk Fusion 360, FreeCAD).

Submissions must prove that the design can achieve all the locations through calculations/images/theory. Designs are required to incorporate any actuators that drive the motion of the arms, but wires and circuitry do not need to be included in the CAD.

For some guidance on this task follow this link or refer to the optional seminar (Mechanical Design): https://www.societyofrobots.com/robot_arm_tutorial.shtml

Submission Type

Submissions should be no longer than 5 pages: 2-3 pages for figures/calculations/images and 2 pages for written submission. The written portion should include details to describe your design process and theory such as: choice in robotic arm degrees of freedom, choice in defined workspace, choice of actuators, potential risks and limitations of your design.

A short video submission can be included to show that your CAD design has proper mate features such that the arm can be manipulated and properly constrained to demonstrate functionality.

Expectation

Below Expectations	Meets Expectations	Exceeds Expectations
Less than 6 locations can be achieved through the robot arm design proposed Design is incomplete and missing key details such as: dimensions, material choice, motor choices, defined workspace and obvious potential risks	At least 6 locations can be achieved through the robot arm design proposed Design has all critical details	All 9 locations can be achieved through the robot arm design proposed CAD model demonstrates that it has proper mate features and can be manipulated freely without breaking Video submission properly demonstrates the animation of the robotic arm and functionality

1.3 Computer Aided Design (Detailed): Additional Function

Primary Support/Grading TAs: Thomas and Armin

Description

Your family just bought a new electric vehicle which uses a different EV charger (.step file can be found on Quercus). Create a design for a shelf to house two separate charger plugs, and modify your robot so that it has a mechanism to pick up and place back the chargers. Your robot can only hold one charger at a time. You can design the shelf in any way you like that can accommodate your original robot design. The design should be made through a CAD software of your choice (e.g. SolidWorks, AutoDesk Fusion 360, FreeCAD).

Submissions must prove that the design can achieve this additional function through calculations/images/theory. Designs are required to incorporate all mechanical components for this mechanism, including actuators and fasteners. Wires and circuitry do not need to be included in the CAD. Theoretical calculations of required motor torque and forces should also be included.

Submission Type

Submissions should be no longer than 5 pages: 2-3 pages for figures/calculations/images and 2 pages for written submission. The written portion should include details to describe your design process and theory such as: a summary of design decisions, additional mechanical testing needed to verify the design, actuator choices and calculations, a description of how the design will move to grab a particular charger and plug it in and then return it, and potential risks of the design with mitigation strategies.

A short video submission can be included to show that your CAD design has proper mate features such that the arm can be manipulated to demonstrate functionality.

Expectation

Below Expectations	Meets Expectations	Exceeds Expectations
The additional functionality cannot be achieved with the proposed redesign Design is incomplete and missing key details such as: dimensions, material choice, force calculations, motor choices, and known potential risks	The additional function of grabbing and returning chargers can be achieved/proven to be possible with the new design Design has all critical details Submission includes a practical “shelf” capable to housing/storing two chargers	CAD model demonstrates that it has proper mate features and can be manipulated freely without breaking Video submission properly demonstrates the animation of the robot interacting with the shelf and proving that it is capable of grabbing and placing the charger plugs

1.4 MECHANICAL FAILURE ANALYSIS

Primary Support/Grading TAs: Harvey and Thomas

Description

With your current robot design, conduct the following three analyses:
1. Develop a comprehensive assessment of the forces found in your teams current or anticipated design. The goal of the assessment is to demonstrate that there is no risk of structural failure (Take the weight of the charger to be 0.25 kg).
2. Assume that the "charger" is now 9 kg.
3. Assume that the charger is back to 0.25kg, but the robot experiences high resistance as it is plugging in the charger (You can estimate it to be 400N).

For all three scenarios, assess where the designed robot will now experience the highest risk of structural failure. The assessment should highlight at least 3 components that are at the highest risk. For the components that experience high compressive or tensile forces, evaluate if the material you have chosen can resist that force using proper calculations and material elastic modulus (utilize a factor of safety of 1.5). Suggest appropriate design change for the last two scenarios if applicable. Each component, as well as the material chosen, should be assessed for their contribution to the forces (use average material densities that can be referenced). For each equation/calculation done, please include a title of what the equation assesses. Use a labeling convention to denote which structural component you are referring to, to save space.

Tip: For the second and third assessments, do not repeat the calculations with the component weights. Instead, evaluate the design as if components are "weightless" and then add the calculated forces from the first assessment. This will save you time and energy.

Helpful Link: https://vention.io/blogs/robot-base-how-to-calculate-their-stability-101

Submission Type

Submissions should be no longer than 5 pages and can be broken down into the following. 1 page for each analysis, which includes appropriate free body diagrams and calculations. An additional page for any calculations that do not fit. 1 page for short summary on each assessment detailing the results of the analysis which highlight structural supports that are at high risk.

Expectation

Below Expectations	Meets Expectations	Exceeds Expectations
Some critical components are not considered in the assessments Oversimplification of robot components/parts makes the assessment irrelevant to the design Material weights are not considered and simplified	All critical components of the mechanical design have been thoroughly assessed Main/critical robot parts are considered for force calculations with proper equations	Practical (weighed using a scale) weight measurements are used All three assessments are completed Different materials densities and weights are considered

		Design changes proposed can address the last two scenarios and are reasonable

2 Electrical

2.1 Printed Circuit Board Design

Primary Support/Grading TAs: Justin and Maryam

Description

As part of this task, you are asked to lay out a 2-layer printed circuit board using Eagle, Circuit Maker, Altium Designer, or some other printed circuit board software of your choice. A list of components, design rule constraints, and a complete schematic will be provided as a basis for this task (see external documents for Bill of Materials and Schematic). The provided schematic is for a maximum power point tracking controller for solar panels. Though no knowledge of solar power systems is necessary for this task to complete it, however.

You are responsible for the layout and placement of each of the parts, as well as creating footprints for any non-standard parts that are included in the design. You should be able to set up your editor with the below design rules, such that they are followed automatically when routing the PCB.

Design Rule Constraints

1. Minimum trace width – 4 mil
2. Minimum trace spacing – 4 mil
3. Minimum hole diameter – 0.2mm
4. Minimum via diameter – 0.25mm + hole diameter
5. Minimum via to trace clearance – 5 mil
6. Via aspect ratio – 6:1
7. Maximum board dimensions – 400mm x 500mm

This manufacturer has some visuals on their website which may help you to understand what each of these rules means. https://jlcpcb.com/capabilities/Capabilities

Submission Type

The submission should be a file of the appropriate type containing the completed printed circuit board, including top and bottom copper layers, substrate thickness, and via locations and sizes. You should also replicate the given schematic within the PCB software's internal schematic creation tool, and link each of the parts to the components on the PCB. If the PCB design tool you wish to use does not include this feature, it is not recommended for use for this task.

A short document or screen capture video should also be included, with a brief explanation of what you've done to create the PCB, showing that you've followed the given design rules. Any video that is submitted **MUST** be a screen capture, no recording of screens with cameras will be accepted.

Expectation

Below Expectations	Meets Expectations	Exceeds Expectations
PCB is incomplete or has parts missing or overlapping each other. Traces are excessively long or narrow to make them	Design includes a PCB of reasonable size, with all parts included. Space on the PCB is	PCB is cleanly designed, and traces of appropriate widths used for each type of connection.

fit on the board. Multiple layers not used efficiently. Custom made footprints are inaccurate to the part they are being used for. Schematic file either not included or incorrectly linked to many or all components. Design rules have been ignored. Circuit is grounded exclusively through traces	well used, without excessively long traces. Custom footprints are reasonably accurate to the parts they represent. Schematic file included but may not be linked to PCB components. Design rules mostly respected. Proper use of ground plane for grounding	All components are linked to a schematic, which is also included. Design rules all respected. Nets used correctly to separate different signal traces on the circuit board.

2.2 AUTONOMOUS SYSTEM POWER BUDGET

Primary Support/Grading TAs: Harvey and Deniz

Description/Scenario

Your robot is to be used in a charging station placed outdoors, and to keep it as green as possible, the robot itself is to be powered via solar power rather than the electrical mains.

1. Using your team's robot design as a baseline, calculate approximately how much power will be consumed by the robot when it runs one plug-in cycle.
    a. This involves estimating the amount of power (Watts) each component (sensors, motors, microcontrollers) draws when it is active/idle, and then determining how long each of these components will be active/idle/off during a plug-in cycle.
    b. If you are using an external computer (i.e. laptop) to do any image processing, you can ignore it for the sake of this task.
2. Then, create a power budget to size a battery and solar panel combination to power the robot. This will involve running a simulation in MATLAB or Python.

    The robot will run once per hour during the day (6:00 – 18:00), and then once every two hours during the night (18:00-6:00), consuming the amount of power you calculated in part 1.

    i. Choose a solar panel size (in m^2) to power the robot, assume the panel has 15% efficiency.
    ii. Choose a battery capacity (in Watt-hours).
    iii. Using the solar irradiance data provided in the excel spreadsheet, calculate how much your battery charges throughout the day due to the solar panel, and how much it discharges due to the robot running. Make sure to respect the upper battery capacity limit. You can use discretized time steps of 15 mins to simplify the calculations and assume that the battery begins 50% charged.
    iv. Modify the solar panel size (from part i) and battery capacity (from part ii) to make sure your battery never falls below 20% of its total capacity for seven days of continuous operation.

Data for this task is taken from Natural Resources Canada
https://www.nrcan.gc.ca/energy/renewable-electricity/solar-photovoltaic/18409

Getting Started

To simulate the solar power system charging and the battery, you should use MATLAB to calculate how much power is in the battery at each time step over the seven-day period. At each time step, the power in the battery will increase by the amount generated by the solar panel based on the irradiance at that time, assuming it is not fully charged. At each time the robot runs a plug-in cycle (indicated by a 1 in the "Robot Runs?" column), the energy in the battery will decrease by the amount you calculated in part 1, to simulate the robot plugging the charger into a car.

Submission

The submission for this task will consist of two deliverables.

The first deliverable, based on your robot design, is a complete power consumption calculation of your robot during one plug-in cycle. This can take the form of an excel spreadsheet, but please ensure that it is clean and clearly labelled.

The second deliverable is a power calculation and battery/solar panel sizing based on the power consumption you've calculated. This submission should include MATLAB or Python code to import the solar irradiance data and calculate the battery state of charge. Plots should also be included showing the amount of energy in the battery over the 7 days that data is provided for, showing that the battery capacity doesn't fall below 20%.

Expectation

Below Expectations	**Meets Expectations**	**Exceeds Expectations**
Significant components of the robot are missing from power calculation Unreasonable active/idle/off times used in calculation Solar power harvested not calculated correctly, or battery exceeds 100% stated capacity, or goes below 0% capacity Solar panel or battery are excessively large for the application	All significant components of the robot are included in the power consumption calculation. Reasonable active/idle/off times have been used to calculate power consumption Solar power harvested is calculated correctly in the simulation Solar panel and battery size are reasonable for the application	Every component of the robot correctly accounted for Inefficiencies/wiring/voltage conversion losses are also included in the power consumption calculation

2.3 Circuit Simulation

Primary Support/Grading TAs: Justin and Maryam

Description

Using a simulation tool of your choice, simulate a boost converter circuit. LTSpice is recommended (a basic instruction guide on how to use LTSpice is provided in the link *LTSpice*), but you are free to use whatever software you like for this if you are familiar with another one. Your converter should be able to accept input voltages of 3, 5, and 12 V DC and output 24 V DC.

The boost converter uses a PWM signal to control a semiconductor switch, switching current between two circuit paths to alternately store energy in an inductor and pump the voltage across a capacitor to a value higher than the source voltage (see *Boost Converter* for more information on how this type of circuit works and an example circuit). A load placed in parallel with the capacitor will then be powered by the higher voltage. The conversion ratio of the input to output voltage is approximately dictated by the formula below, where D is the duty cycle of the PWM signal. The duty cycle should be the only thing you vary in order to switch between voltages in your circuit.

$$D = \frac{Vout - Vin}{Vout}$$

In order to do this, you will need to choose inductance and capacitance values for the converter that will enable proper operation. With these values chosen, simulate the circuit, generating the following plots:

1. Input and output voltages vs time for each of the three input voltages
2. Current through the inductor and the voltage across the capacitor for each case
3. Voltage across the switch
4. Current through the switch

Make sure to simulate for a long enough time such that the output voltage stabilizes and use a switching frequency high enough that the inductor always has current flowing through it. Use a timestep that is smaller than the period of the switching frequency. Your circuit will have some ripple (oscillation around the steady state voltage), this is fine as long as it’s relatively small compared to the steady state voltage.

With the maximum values obtained from plots 1-4, find parts (inductor, capacitor, MOSFET switch) on digikey.ca that correspond to the inductance and capacitance values you specified. If the components are to large or expensive to be placed on a small printed circuit board, the reviewing TA may ask you to tweak your inductance, capacitance, or switching frequency values to find more reasonable ones.

Submission

Submission for this task should consist of

1. LTSpice (or other simulator) files used to generate plates
2. Document or slides containing plots 1-4 for each of the three input voltages, with brief explanation of each
3. Datasheets for each of the chosen parts from digikey (pdf format)

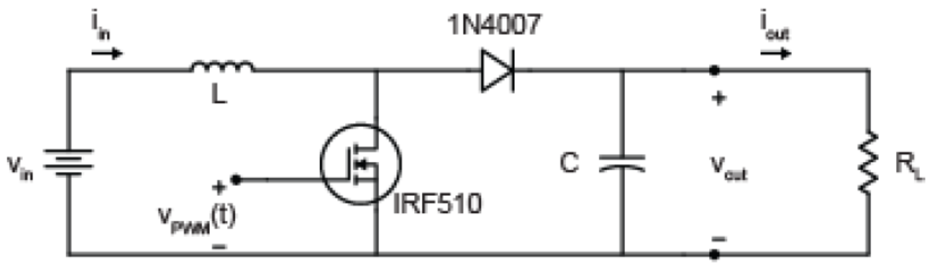
Links

LTSpice

https://www.allaboutcircuits.com/technical-articles/basic-circuit-simulation-with-ltspice/

Boost Converter

https://components101.com/articles/boost-converter-basics-working-design



1N4007 represents a diode, IRF510 represents a switch. You can use a generic diode and MOSFET for these parts in your simulation. Use a 1 k Ω load resistance to start.

Expectation

Below Expectations	Meets Expectations	Exceeds Expectations
Circuit either doesn't convert voltage correctly or has huge ripples or is unstable Incorrect components used in converter Inductances and capacitances are excessively large for the application	Circuit operates properly, converting input voltages to correct output, but ripple may be large All requested plots included, but current/voltage values may be higher than feasible for a small converter Chosen inductance and capacitance values are reasonably sized Parts found on digikey have correct values, but may not have suitable current/voltage limits	Simulation includes losses to model realistic component values Parts found on digikey are reasonably sized and priced, and respect limits on current and voltage found in plots 2,3, and 4 for each input voltage

3 PROGRAMMING

3.1 CLOSED-LOOP CONTROLLER: DRIVING A MOBILE ROBOT STRAIGHT

Primary Support/Grading TAs: Daniel and Yoshiki

Description

The task is to implement a PID controller so the simulated rover, as shown in Figure 1 will drive straight. A short additional reading on control theory and MATLAB starter code for this project are available on Quercus.

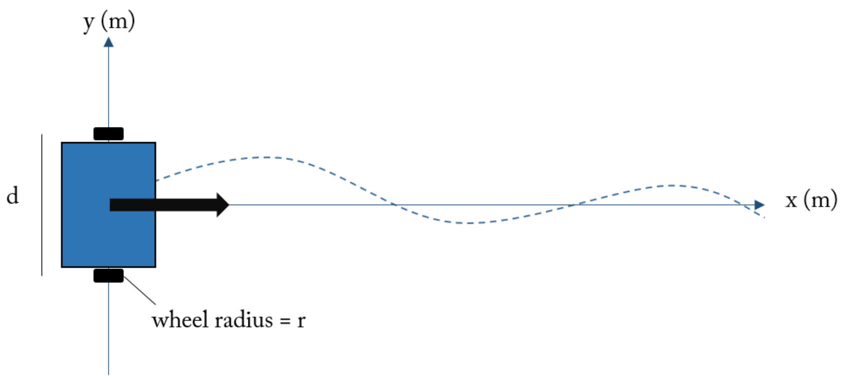


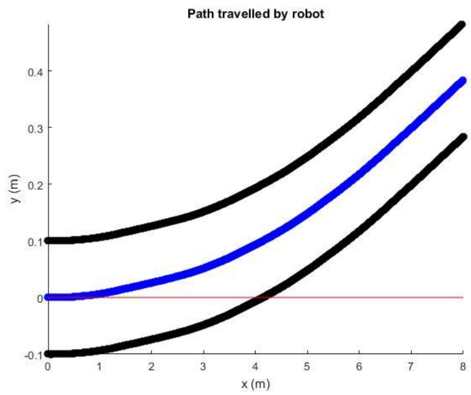
Figure 1 - Schematic of simulated rover to be controlled. Wheels are of radius r, separated by d (both in (m)). The goal of the control scheme is to minimize the error in driving along the x-axis.

Our control signal is the PWM duty cycle (DC) delivered to each motor. In the simulation, there are several factors that prevent the rover from driving straight:

1. The motors are not symmetric (simulated by making one motor drive faster).
2. The surface is rough and uneven (simulated by random disturbances in the robot's movements).

You will attempt to maintain the rover on a straight path by exploring the implementation of a PID controller via the following tasks. Sample code are provided on Quercus as a starting point. There are 3 sections. Sections 1 and 2 are fast, and only require you to use and slightly modify the provided code. Section 3 asks you to implement your own PID, explore the simulation limitations, and discuss the practical issues involved with PID controllers.

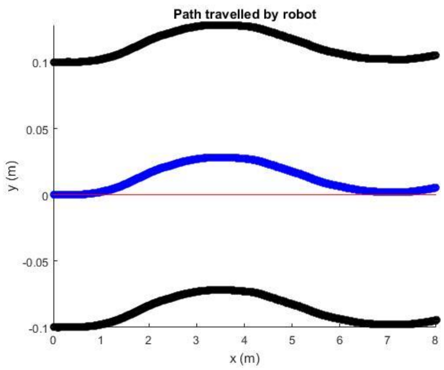
Part I: Setting up.	This section contains a series of short questions designed to get you started with the project. Please briefly answer these in your report and provide figures as necessary. 1. Run the *main_script.m* starter code, as provided. You should produce a figure similar to the one given below, where the wheel paths are the black lines, and the blue path depicts the center of the robot.

<table>
<tr>
<td></td>
<td>


<ol start="2">
Re-run the code several times. Then, pick an example and explain the factors that cause the trajectory of the robot to deviate. Bonus: find where the positional disturbances and motor power offset are implemented in the code.
Simulation parameters: change the parameter N which controls the number of seconds of simulation, and the pps (points-per-second) which controls the resolution. Pick values that work for you, and briefly justify them.
Robot parameters: choose values to create your unique rover; r, wMax, and d. Explain your choices.
Control parameters: look at spd, dT, and ctrl_enable. Explain how these are used, but do not modify them yet. Pay close attention to the relationship between the control interval dT and the simulation time-resolution pps; explain how this could cause problems if not properly matched.
Within the drive_robot.m helper function, look at the sub-functions getDC. Explain how getDC is used to convert from desired speed to a duty cycle.

</td>
</tr>
<tr>
<td>Part II: Closed-loop proportional control</td>
<td>
This short section implements and explores a simple proportional controller.

Change ctrl_enable = 1 and run the main script. You should produce a figure similar to below. What is the value of the proportional coefficient K? Briefly explain why the robot trajectory has changed.



<ol start="2">
In drive_robot, explain what getError does. Is this realistic? Explain (with the help of equations) how you would implement this function using encoder readings instead.
Explore how changing the proportional parameter K affects system behavior. Also explore how changing dT affects control.
Create new figure(s) to help explain errors, settling time, overshoot, etc.

</td>
</tr>
<tr>
<td>Part III: Closed-loop control</td>
<td>By this section, you should have a good understanding of the simulation. You will implement, tune and explore the parameters of a PID controller. Most of your report should deal with this section.</td>
</tr>
</table>

	1. Briefly explain how to discretize the terms of the PID, since in this numerical simulation we cannot use the continuous forms. For this, should we use the resolution of the simulation, or the *dT* controller frequency? Explain. 2. Implement a new control case for Proportional Integral Derivative (PID) control. Please refer to the supplementary material and the control seminar from the beginning of the semester, for background guidance and examples. 3. Explain your approach to tuning the PID parameters. Why is this approach appropriate? Demonstrate (with figures) how the parameters change behavior, with reference to underdamped, overdamped and critically damped systems. 4. Compare the behavior of the system using the PID to other control schemes (at least two of proportional, integral, derivative, or other, or combinations thereof). 5. Given your tuned PID control system, are you able to increase disturbances and noise to “break” your controller? Please demonstrate. Discuss the limitations of this simulation. 6. Discuss some important practical issues (e.g. interrupt timing on a serial processor, windup, derivative noise) that must be considered for real-world implementation. Are there improvements that you could recommend for the PID algorithm?

Submission

Submission to this task shall be a written document of no longer than 5 pages, and your modified MATLAB code in a zip folder. You should include answers and simulation results from all three parts of the exercise, with a focus on Part III. For Part I and II, describe how the parameters changed and answer the questions. For Part III, provide discussion on the open-ended questions. Show the process of tuning the control parameters and attach the corresponding simulation results.

Expectation

Below	Meet	Exceed
Report is incomplete, little evidence of individual learning. Graphs are missing titles, axes labels, and/or legends, or there are no original figures. Implementation errors in code, or code not submitted.	Tasks are completed as described; all results are supported, and discussion contains evidence of research. Graphs are labelled appropriately and referenced in-text. Code is submitted, is easily legible, and is largely free from errors.	As per meet, and: Analytical conclusions are insightful and draw from additional research into the field of control theory. Simulations are extended beyond provided code and reproducible.

3.2 3D Vision with 2D cameras

Primary Support/Grading TAs: Jane and Zain

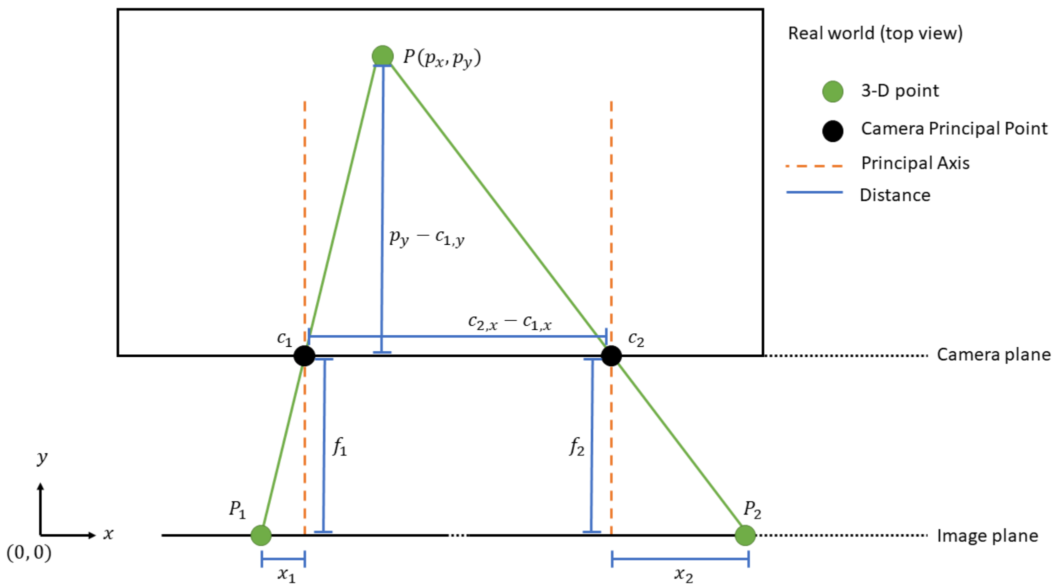
Description

Stereo vision mimics the human eye's method of reconstructing 3D information about the world. You can use two 2-D cameras spaced some distance apart to reconstruct points in 3D space. You are required to implement stereo vision point cloud construction on any of the following sets of images (perf mode recommended):

http://vision.middlebury.edu/stereo/data/scenes2014/

You may choose to use Python or MATLAB for this exercise, but you should apply the stereo image reconstruction method yourself.

1. Create a disparity map for the set of images. First compute the correlation coefficient for each row of pixels to match features. Then compute the pixel distance from one image to the other for each pixel. You may refer to the following resource: https://www.cse.usf.edu/~r1k/MachineVisionBook/MachineVision.files/MachineVision_Chapter11.pdf
2. Compute the real-world 3D point P (px, py, pz) for each 2D pixel (pick one reference image). Indicate the formula for each coordinate based on the diagram below. Use the camera parameters given in the dataset.



3. Plot the reconstructed point in 3D. Show a video if you want!
4. Consider rectification of stereo images. How would this affect the complexity of a mobile rover application? What kind of sensors could you use to inform you about the rover state to help with image rectification?

Submission

Your submission should be a written document ≤ 5 pages. You must include a description of the process you used when performing this task, as well as the 3D reconstructed image. Indicate any formulas you used, and/or assumptions you made. You may include a video to show the 3D point cloud. A discussion about the performance and any limitations of this method should be discussed.

Expectation

Below	**Meet**	**Exceed**
Work is incomplete or answers to questions are vague or surface level. Graphs are missing titles, axes labels, and/or legends.	Tasks are completed as described and all answers are justified with factual basis or real-life relevance. Generated 3D image matches real life objects. Graphs are labelled appropriately and referenced in-text.	As per meet, and: Analytical conclusions are insightful and draw from additional research into the field of computer vision.

3.3 Camera Calibration: Correcting Radial Distortion

Primary Support/Grading TAs: Jane and Zain

Description

Radial distortion of images occurs due to the camera lens' imperfections which distorts the incoming rays of light. Radial distortion can be modelled as a function of distance from the center of the image, hence the term "radial".

1. Artificially distort an image of your choice by applying Brown's distortion model:

$$x_{distorted} = x \cdot (k_1 r^2 + k_2 r^4 + k_3 r^6)$$
$$y_{distorted} = y \cdot (k_1 r^2 + k_2 r^4 + k_3 r^6)$$

   For each pixel in the image, where if (x_c, y_c) is the centre of the image,

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$$

   Formulas adapted from https://www.mathworks.com/help/vision/ug/camera-calibration.html, distortion in camera calibration.

   Try combining different combinations of k_1, k_2, k_3 being positive or negative. Save the results of the images. What is the effect of changing the sign? Magnitude?
2. Now you will take one of your artificially distorted images and undistort it using:
   a. Only the k_1 term
   b. Only the k_1, k_2 terms
   c. All terms, i.e. k_1, k_2, k_3

   Compare the performance. When might you need to use the higher order model? When would the k_1 term be sufficient?
3. In practice, what types of lens would need calibration for radial distortion? What application would this be relevant for?

Submission

Written document and graphs (suggested to be done in MATLAB or Python). The document should be ≤5 pages. You should include sections which describe and depict applying the artificial distortion, the performance of the un-distortion across number of terms considered, and a section discussing the application/usage of such camera calibration.

Expectation

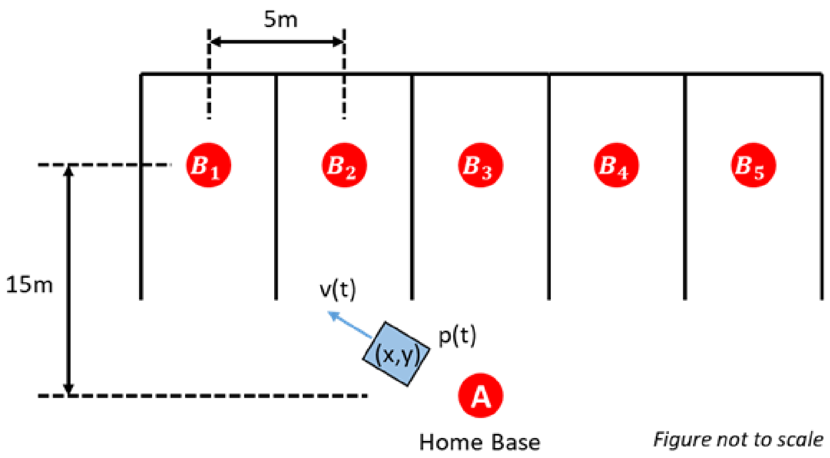
Below	Meet	Exceed
Work is incomplete or answers to questions are vague or surface level. Graphs are missing titles, axes labels, and/or legends.	Tasks are completed as described and all answers are justified with factual basis or real-life relevance. Graphs are labelled appropriately and referenced in-text.	As per meet, and: Analytical conclusions are insightful and draw from additional research into the field of computer vision.

3.4 MOBILE ROBOTICS: IDENTIFYING SOURCES OF ERROR

Primary Support/Grading TAs: Daniel and Yoshiki

Description:

You are designing a mobile charger delivery system for an outdoor parking garage. Your robot will move the battery pack from home base location (A) to each of the parking spots (B1 – B5). You will only be able to carry one battery pack at a time, before coming back to base to pick up the next one. Due to the outdoor nature of this setting, there will not be lines or tracks to follow. You have access to an onboard accelerometer and GPS. You would like to know how well your sensors perform for this application. This exercise will focus on sensor calibration and not path finding or plug insertion.



You are asked to plan and simulate the trajectory of the robot to charge the five vehicles in the parking spots. The trajectory of the robot is given as $p(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}$ over $t \in [0, T]$, with constant time steps Δt. You may use any programming environment of your choice. To simulate the robot motion in discrete time, choose a reasonable velocity and use Euler's method:

$$p(t_i + \Delta t) = p(t_{i+1}) = p(t_i) + v(t_i)\,\Delta t$$

Requirements:

1. Simulate the desired motion of the robot for $t \in [0, T]$. This will be the "ground truth" position information. Plot the trajectory in 2D.
2. On the same figure, overlay the following plots.
   The trajectory of the motion in part (1) if we were to obtain the position of the robot from:
   a. GPS: You may assume the GPS module provides an instantaneous (x, y) position of the robot at each time sample, with added noise in each axis which is sampled from a normal distribution $\sim N(\mu = 0, \sigma^2 = 1m)$.
   b. Accelerometer: Obtain the true acceleration (a_x, a_y) by taking the twice-derivative of the position data from part (1). Your accelerometer measurement is the true acceleration distorted by additive error from two sources: (i) noise sampled from a normal distribution $\sim N(\mu = 0, \sigma_a^2)$; choose σ_a such that 95% of the noise distribution falls within 45% of the maximum distance travelled over any time step and (ii) a small baseline offset to the data. The error from (ii) replicates sensor drift.

Now, numerically integrate the accelerometer reading twice to reconstruct the position.

3. Re-simulate the trajectory, now with the robot using accelerometer data to identify its current position $p(t_i)$ at every time step and re-adjust its velocity $v(t_i)$ towards the destination. Overlay this trajectory over the ground truth.
4. Comment on the sources of error:
    a. Numerical method error
    b. Sensor noise error
    c. Sensor drift error

    Can you distinguish how each factor affects your reconstructed position values? What can you do to reduce the effects of each of the sources of error?
5. Notably, there is a technique called the “Kalman filter”. In your own words, explain what this technique does and why it is useful. How would you use a Kalman Filter for the scenario described in this task? Indicate any sources you use to supplement your understanding.

Submission

The submission should be a written document ≤ 5 pages and must include the required trajectory plots. Specifically, you should include 2-3 pages setting up the problem and show the overlaying trajectories for the robot when taking position values from a) ground truth, b) GPS, c) accelerometer. Another page should be taken to explain and show the trajectories of the robot if it uses the sensor position data to determine its path. The last page should be used for discussion on the errors and possible ways to use Kalman filter for this design.

Expectation

Below	**Meet**	**Exceed**
Work is incomplete OR plotted trajectories do not reflect the nature of the problem OR discussions are vague or surface level. Graphs are missing titles, axes labels, and/or legends.	Tasks are completed as described. All simulated trajectories reflect the effect of sensor error. Graphs are labelled appropriately and referenced in-text.	As per meet, and: Analytical conclusions are insightful and draw from additional research into the field of mobile robotics.