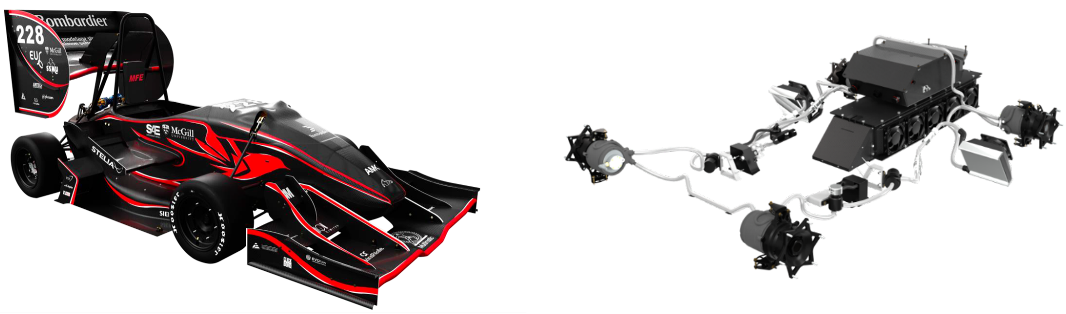
MECH 513 - Control Systems
Prof. James Richard Forbes ©
Course Project - DE.3

(Last Updated: November 14, 2025)

Motivation

The McGill Formula Electric (MFE) 2025 car is shown in Figure 1a. The car features all-wheel drive where electric motors drive each wheel. The electric motors heat up significantly during operation. The motor cooling system shown in Figure 1b extracts heat from the motors that is then rejected through a radiator. In more detail, water is pumped through motor cooling jackets where heat is transferred from the motors to the water. The water is then pumped to a radiator where the heat in the water is rejected to the atmosphere.



(a) MFE 2025 car. (b) Cooling loop.

Figure 1: MFE 2025 car and cooling loop.

The water pumps consume electrical energy from the car's low-voltage battery. Continuous operation of the water pumps, which is the existing solution, necessitates a large battery, both in terms capacity and mass. A large battery results in a larger car mass. A larger car mass means reduced car accelerations. Reduced car accelerations means a longer lap time. Naturally, the goal of MFE is to minimize the lap time. Therefore, working backwards, if the capacity and mass of the battery can be reduced, lap times can be further minimized. To reduce the capacity and mass of the battery, a means for the water pumps to use *less* energy is sought.

To reduce the energy consumed by the pumps, rather than simply having the pumps circulate water continuously, *feedback control* can be used to change the water flow rate given the changing motor temperatures in real time. Roughly speaking, less water flow is required at low motor temperatures, while more water flow is requires at high motor temperatures. The total energy consumed by the pumps will be reduced when water is pumped "as needed" rather than "all the time".

The problem at hand boils down to controlling the flow rate of water using the pumps. Focus will be placed on controlling one pump, as shown in Figure 2. In order to control the flow rate of water, a model of the pump is needed, and then a controller needs to be designed. These two tasks will be completed in **Parts 1** and **2**. Specific deliverables associated with Parts 1 and 2 are listed in the subsections below.

Reporting and Submission

Completion of the deliverables associated with Parts 1 and 2 must be communicated form of two reports, a report for Part 1, and another report for Part 2. Each report has the same format as outlined in Appendix A. When submitting Part 2 do not just append it to your Part 1 submission. Treat the reports for Part 1 and Part 2 as separate reports/documents.

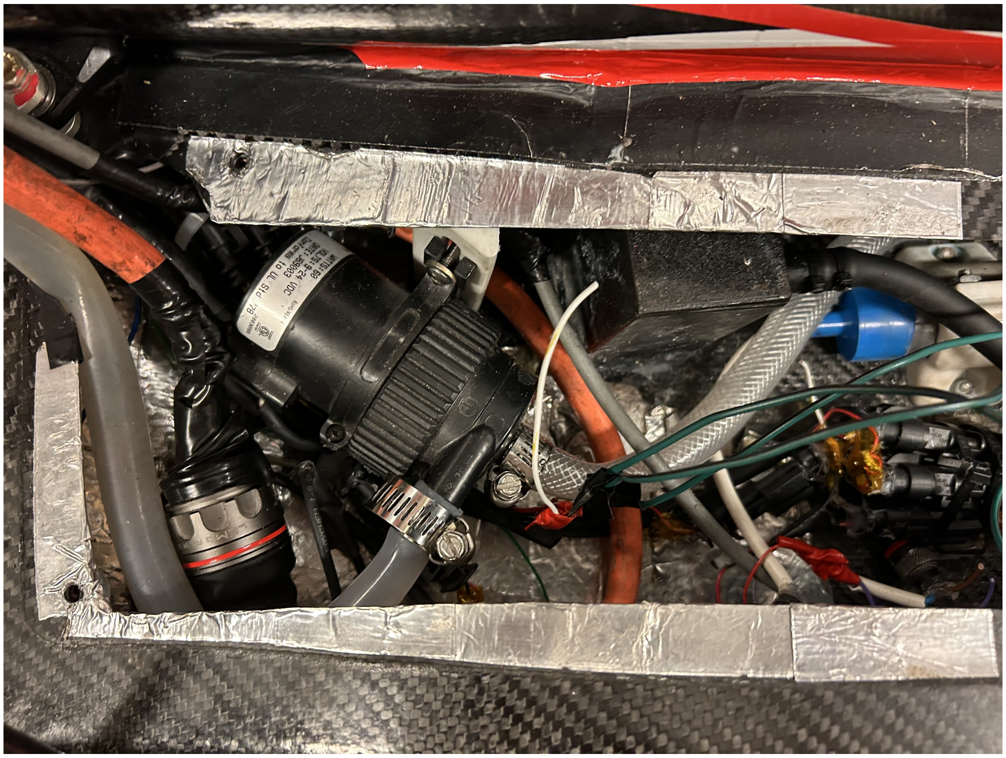


Figure 2: Pump, hoses, and other wiring, within the MFE 2025 car.

The reports associated with Parts 1 and 2 will be submitted for evaluation sequentially. Part 1 will be completed and submitted roughly 1/2 to 2/3 through the semester, then Part 2 will be completed and submitted at the end of the semester. However, Part 2 builds on Part 1.

Each report will be submitted as *one* pdf file on Crowdmark *as a group*. Groups must be groups of 2 students. Students can form their own groups.

Make sure plots are easily readable. Any frequency response plot (e.g., a Bode plot) must use units of Hz on the x axis.

Python code must be submitted on myCourses (under "Assignments") as ONE zip file by ONE group member. Specifically, the one who's last name is alphabetically first, is required to submit code on myCourses.[1]

1 System Identification and Uncertainty Characterization

Deriving a first principles model of the pump is impractical for many reasons. As such, 4 datasets have been collected that you will use to system ID a model. This model will used for controller design in Part 2. Each dataset includes time, the input voltage to the pump in units of volts (V), and a measure of of the water flow rate in units of liters per minute (LPM). The maximum input voltage is 5 (V). Each each dataset corresponds to a different input voltage range. The pump does not pump water when the input voltage is less than approximately 1.5 (V). (For the purposes of this project, this fact will be ignored.)

You must use the 4 input-output datasets to identify a nominal plant model, $P_0(s)$, and to characterize the uncertainty of the model in the frequency domain. Your results will be communicated in the form of a short report. Your report should include, at a minimum, the following.

1. A brief (roughly one paragraph) introduction to the problem.

[1] For example, say the group members last names were Haddad, Ferraro, and Kowalski. Listing these last names alphabetically results in Ferraro, Haddad, Kowalski. Thus, the person with the last name Ferraro would be the person to submit the code on myCourses.

2. A step-by step discussion as to how you identified a continuous-time nominal model, $P_0(s)$, using the discrete-time input-output data. In particular, what is the system ID approach taken? How did you use the 4 input-output datasets to come up with one nominal model? What metrics did you use to assess the quality of the ID'ed model? How did you validate the model? How did you pick a numerator and denominator order, that is, how did you pick m and n in

$$P_0(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0}. \tag{1}$$

   Plots comparing the error between the measured output and the system ID'ed output versus time, the relative error versus time, or any other plot(s) you think are worth communicating, can be included.

3. A step-by-step discussion as to how you characterized the model uncertainty in terms of the transfer function $W_2(s)$. In particular, what are the "off nominal plants", $P_k(s)$, $k = 1, 2, 3, \ldots$ used? What uncertainty model did you choose and why? Provide a bode magnitude plot of the nominal plant and the off nominal plants. How did you pick the order n of $W_2(s)$ where

$$W_2(s) = \frac{b_n s^n + b_{n-1} s^{n-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0}. \tag{2}$$

   Include a bode magnitude plot of $W_2(s)$ bounding the residuals between the nominal plant $P_0(s)$ and the off nominal plants $P_k(s)$.

4. Justify why, or why not, MFE should continue to work with you for Part 2.

Advice and Considerations

- You may use any `numpy`, `scipy`, or `control` commands within `python`. Additionally, you may reuse (and are encouraged to reuse) `python` code from MECH 513 assignments. However, you many not use any system ID libraries found on git, generated purely/exclusively using GenAI (e.g., ChatGPT, Claude, co-pilot, gemini, etc.), or found anywhere else. You must write your own system ID code.
- Rather than using

$$P(\jmath\omega_m) = \frac{\hat{u}_m^{\mathsf{H}} \hat{y}_m}{|\hat{u}_m|^2}, \qquad m = 0, 1, \ldots, N/2 - 1, \tag{3}$$

  to compute $P(\jmath\omega_m)$, it is suggested to use [1, Sec. 7.7]

$$P(\jmath\omega_m) = \frac{P_{uy}(\jmath\omega_m)}{P_{uu}(\jmath\omega_m)}, \qquad m = 0, 1, \ldots, N/2 - 1, \tag{4}$$

  where $P_{uy}(\jmath\omega_m) = \frac{2}{N f_s} |\tilde{u}_m^{\mathsf{H}} \tilde{y}_m|$ is the cross power spectral density (PSD) between $y(\jmath\omega)$ and $u(\jmath\omega)$, $P_{uu}(\jmath\omega) = \frac{2}{N f_s} |\tilde{u}_m|^2 = \frac{2}{N f_s} \tilde{u}_m^{\mathsf{H}} \tilde{u}_m$ is the PSD of $u(\jmath\omega)$, and f_s is the sampling frequency. Equation (4) can be derived directly from (3). Starting from (3),

$$P(\jmath\omega_m) = \frac{\hat{u}_m^{\mathsf{H}} \hat{y}_m}{|\hat{u}_m|^2} = \frac{\frac{2}{N f_s} \hat{u}_m^{\mathsf{H}} \hat{y}_m}{\frac{2}{N f_s} |\hat{u}_m|^2} = \frac{P_{uy}(\jmath\omega_m)}{P_{uu}(\jmath\omega_m)}, \qquad m = 0, 1, \ldots, N/2 - 1. \tag{5}$$

  The reason (4) is used over (3) is that, when coding in `python`, the cross PSD function (and PSD function) `signal.csd` from the `signal` package can be used. The function `signal.csd` has built-in windowing capabilities. Windowing smooths out the cross PSD computation, enabling a better estimate of the frequency response [1, Sec. 7.5.3]. The specific windowing used within `signal.csd` is called a Hanning window. To enable windowing, use

```
f, Puy = signal.csd(u, y, fs=fs, window='hann')
f, Puu = signal.csd(u, u, fs=fs, window='hann')
```

where `fs` is the sampling frequency the data is collected at.

- Once a model has been ID'ed using *training data*, it's important to test how good the ID'ed model predicts outputs given a different input using *test data*. The training and test data cannot be the same. Assume a model has been ID'ed. Let u_k and y_k be test input-output data. Let y_k^{ID} be the predicted output of the ID'ed system given the test input u_k. The error between the predicted output and the test output is $e_k = y_k^{\mathrm{ID}} - y_k$. The percent relative error at time t_k can be defined as $e_{k,\mathrm{rel},y_{\max}} = \frac{|e_k|}{y_{\max}} \times 100$ where $y_{\max} = \max_{k=1,\ldots,N} |y_k|$, or as $e_{k,\mathrm{rel},\sigma_y} = \frac{|e_k|}{\sigma_y} \times 100$ where $\sigma_y = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(y_k - \bar{y})^2}$.
- Let

$$\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \cdots & y_{N-1} & y_N \end{bmatrix}^{\mathsf{T}}, \quad \mathbf{y}^{\mathrm{ID}} = \begin{bmatrix} y_1^{\mathrm{ID}} & y_2^{\mathrm{ID}} & \cdots & y_{N-1}^{\mathrm{ID}} & y_N^{\mathrm{ID}} \end{bmatrix}^{\mathsf{T}}, \quad \mathbf{e} = \mathbf{y}^{\mathrm{ID}} - \mathbf{y}. \tag{6}$$

  The sum of squared errors (SSE) is defined as

$$\mathrm{SSE} = \frac{1}{N}\sum_{k=1}^{N}(y_k^{\mathrm{ID}} - y_k)^2 = \frac{1}{N}(\mathbf{y}^{\mathrm{ID}} - \mathbf{y})^{\mathsf{T}}(\mathbf{y}^{\mathrm{ID}} - \mathbf{y}) = \mathrm{var}(\mathbf{y}^{\mathrm{ID}} - \mathbf{y}) = \mathrm{var}(\mathbf{e}). \tag{7}$$

  The *percent variance accounted for* (%VAF) is defined as

$$\%\mathrm{VAF} = \left(1 - \frac{\mathrm{var}(\mathbf{e})}{\mathrm{var}(\mathbf{y})}\right) \times 100. \tag{8}$$

  When %VAF is closed to 100% the ID'ed model is "good".
- The fit ratio (FIT) is defined as

$$\mathrm{FIT} = \left(1 - \frac{\sqrt{\frac{1}{N}\sum_{k=1}^{N}(y_k^{\mathrm{ID}} - y_k)^2}}{\sigma_y}\right) \times 100 \tag{9}$$

  where

$$\bar{y} = \frac{1}{N}\sum_{k=1}^{N} y_k, \qquad \sigma_y = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(y_k - \bar{y})^2}. \tag{10}$$

  The root mean square error (RMSE) is $\sqrt{\frac{1}{N}\sum_{k=1}^{N}(y_k^{\mathrm{ID}} - y_k)^2}$. Thus, FIT equals 1 minus the normalized RMSE (NRMSE) where the normalization constant is σ_y.

  When FIT is closed to 100% the ID'ed model is "good". Generally, %VAF $>$ FIT. For instance, if %VAF $\approx 90\%$ expect FIT $\approx 70\%$.
- You can find sample uncertainty bound code here:

  `https://github.com/jrforbes/mech_412_code/tree/main/python/uncertainty_bound`

2 Controller Design

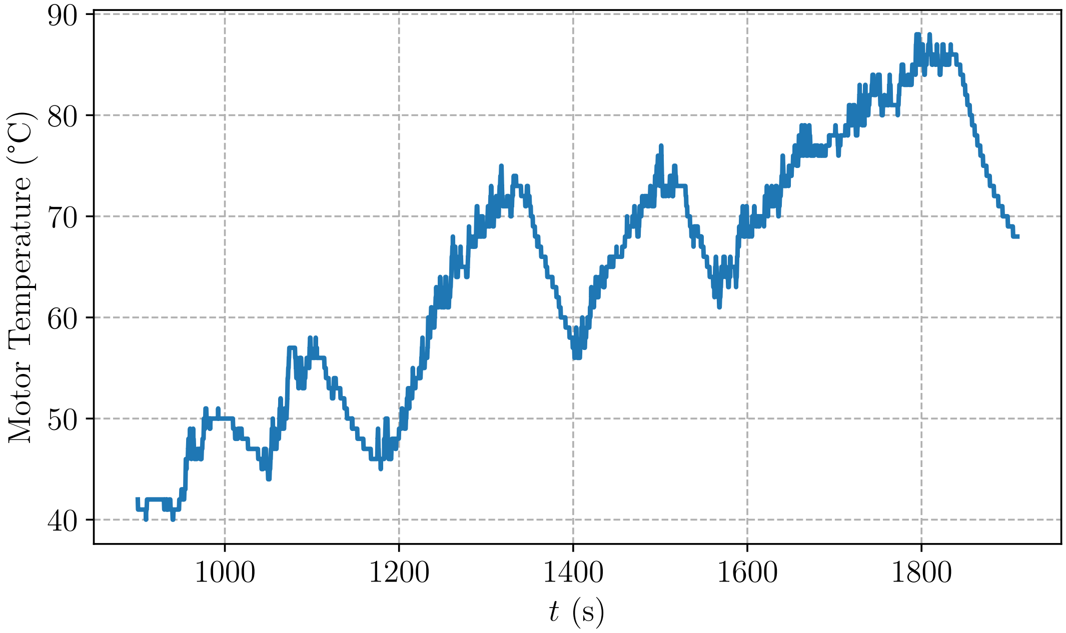


Figure 3: Motor temperature versus time over $t \in [900, 1900]$ (s).

The temperature of the motor(s) is measured in real-time during an endurance test. The motor temperature during an endurance test, $T_{\text{motor}}(t)$, is given in Figure 3. The minimum motor temperature is $T_{\text{motor,min}} = 40$ (°C) and the maximum motor temperature is $T_{\text{motor,max}} = 88 \approx 90$ (°C).

When $T_{\text{motor}} = T_{\text{motor,min}}$ the pump should not pump water. On the other hand, when $T_{\text{motor}} = T_{\text{motor,max}}$ the pump should be pumping the maximum amount of water possible. Thus, $T_{\text{motor}}(t)$ can be used to compute a water flow-rate reference for the control system to track. You must design a control system able to track a desired water flow rate.

The control system must meet the following requirements.

- The tracking error, which is a water flow-rate error, should always be less than $\pm 5\%$ of the maximum water-flow rate. Briefly exceeding $\pm 5\%$ due to a rare yet large noise value (e.g., $\pm 2\sigma_n$ or $\pm 3\sigma_n$), where σ_n is the standard deviation of the noise, is acceptable.
- The control input voltage should not exceed ± 5 (V). Briefly exceeding ± 5 (V) due to a rare yet large noise value (e.g., $\pm 2\sigma_n$ or $\pm 3\sigma_n$), is acceptable.
- The control system must be robust to model uncertainty.

Using the identified model and its uncertainty characterization from Part 1, you are to design a control system to meet these requirements. Your approach to control design must be documented in a short report. Your report should include, at a minimum, the following.

1. A brief introduction to the problem.
2. You are given the temperature data in (3). Explain how you used this data to determine $r(t)$, the command to track. Explain how you used this data to find f_r (that is, ω_r, just in Hz). Use data within $t \in [900, 1900]$ (s).
3. A series of voltage to water-flow rate step responses are given. Explain how you used this data to characterize the noise corrupting the water-flow rate measurement. What's the standard deviation of the noise, σ_n?

4. Discuss how to formulate the generalized plant to be used in DK-iteration. What scaling, if any, did you use? What weights, if any, did you use? How does $\mathbf{u}_\Delta(s)$ and $\mathbf{y}_\Delta(s)$ interact with the generalized plant? What are the exogenous inputs, $\mathbf{w}(s)$? What are the performance outputs, $\mathbf{z}(s)$? What are the inputs to the controller, $\mathbf{y}$? Clearly write out the generalized plant in transfer matrix form.

5. A sample code has been provide on myCourses that you must build on. Building on the sample code, synthesize a controller using DK-iteration.

6. Within the sample code, be sure to replace the "dummy" plant, controller, reference, σ_n, etc. with your nominal plant, your controller, your reference, your σ_n, etc. Notice that time is over $t \in [900, 1900]$ (s). Provide plots of the error $e(t) = r(t) - y(t)$ (LPM) versus time, $r(t)$ and $y(t)$ (LPM) versus time, and $u(t)$ (V) versus time, where $t \in [900, 1900]$ (s). Provide the mean of $e(t)$ as a percent of maximum water-flow rate (i.e., $\mathrm{mean}(e(t))/\max \mathrm{LPM}$), the standard deviation of $e(t)$ as a percent of maximum water-flow rate, the max ideal error as a percent of maximum water-flow rate, and the max voltage as a percent of maximum voltage, only using $t \in [900, 1900]$ (s). Comment on if or if not your control design meets the performance requirements.

7. Recall that power is $P(t) = v(t)i(t)$ where $v(t)$ is the voltage and $i(t)$ is the current. Given that $v(t) = i(t)R$ where R is a "lumped" motor resistance, a proxy for power is simply $P(t) = v(t)^2$.

   The prior MFE cooling solution was to simply run the plumps at their max voltage, resulting in the max flow rate, for the entire endurance test. This consumes $E_{\mathrm{baseline}} = \int_{t_1}^{t_2} P(t)\,\mathrm{d}t \approx 25000$ ($\mathrm{V}^2 \cdot \mathrm{s}$) of energy, where $t_1 = 900$ (s), $t_2 = 1900$ (s). If E_{fb} is the energy consumed when using feedback control, compute the energy savings as a percent relative to the baseline, that is $(E_{\mathrm{baseline}} - E_{\mathrm{fb}})/E_{\mathrm{baseline}} \times 100$. Comment on the energy savings, if any, when using feedback control relative to the baseline solution.

8. Justify why, or why not, MFE should have you proceed to the next step, which would be to implement the control system on the car.

Advice and Considerations

- To ensure plots are not too small, your Part 2 report can be up to 10 pages of "main text" (e.g., not including title page, table of contents, and reference).
- You may use any `numpy`, `scipy`, or `control` commands within `python`. Additionally, you may reuse (and are encouraged to reuse) `python` code from MECH 412 assignments. However, other than code provided by Prof. Forbes, you many not use any control libraries found on git, code generated purely/exclusively using GenAI (e.g., ChatGPT, Claude, co-pilot, gemini, etc.), or code found anywhere else. You must write your own code.
- The fact the pump "get stuck" when the voltage is below 1.5 (V) will be ignored. Just assume the pump starts moving for any non-zero voltage input.
- The maximum input voltage is 5 (V). You can use the DC gain of $P_0(s)$ to find the maximum water-flow rate. In reality, the max voltage can be slightly exceeded, but only for very short durations.
- Your solution must use the tools taught in MECH 513. That is, a "hand tuned" controller, à la MECH 412, will not be accepted.
- By far the hardest part of this is formulating the generalized plant properly. Carefully think about normalization, weights, how uncertainty enters into the plant, etc.
- If you cannot get DK-iteration to work, attempt a weighted mixed sensitivity design, or a $\mathcal{H}_\infty$ or $\mathcal{H}_2$ design. At the end of the day, you need to use *at least one* controller design method taught in MECH 513.
- When formulating the generalized plant, use the `control` packages `interconnect` function. Reuse the sample code provided from the assignment(s).
- Within the sample code is a means to simulate the plant, which is *not* linear, and the controller, which *is* linear. The simulation realizes the interconnection shown in Figure 4.

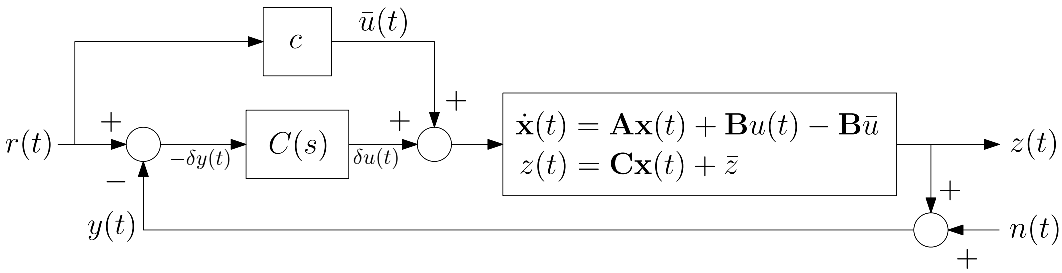


Figure 4: Feedback loop involving an affine plant.

3 Acknowledgement

A big "thank you" to Philip Becker who collected the pump data used in this project. Additionally, "thanks" to MFE for supporting this project by providing pictures and additional data.

Appendices

A Report Requirements

Specific Report Requirements

- 7 pages or less, including main text, figures, tables, equations, etc., but excluding front and back matter such as the title page, table of contents, and references.
- No appendix is permitted. All text, plots, tables, must be in the main report.
- 11 point font.
- Single spaced.
- 1” (≈25 mm) margins.
- 8.5” × 11” paper.

Tips and Hints

- **Justify** every design decision you make with analysis. For example, as part of the control design part of this project, if you present results that satisfy the design requirements (e.g., the voltage is less than 5 (V)), but at no point do you **justify** the design decisions you made in order to meet the design requirements, zero marks will be allotted.

- The report is not a “code guide” where you explain how the code you wrote works. The point of the report is to justify **justify** why/how the analysis, your design decisions, your results, etc. are correct.

- The report is short. Get to the point. Communicate the most important, the most critical findings. “More text” does not equal “a better report”. My advise when writing the report is as follows.
  - Create a “skeleton” of the report that includes each section/subsection title you think you need.
  - Put in the plots in the appropriate section(s) before writing anything. The plots act as a lighthouses; you need to structure your report such that, in the end, the reader understands the plots.
  - Before writing the main body of text, in each section/subsection, write two to three key bullet points that must be communicated to the reader. If you can’t, what’s the point of the section/subsection?
  - Before writing the main body of text, revise your skeleton, your plots, your bullet points as appropriate.
  - Write the main body of text first; write the introduction and conclusion last.
  - Explain in the text the story any plot or table is telling. If a plot of table is not telling a story, what’s the point of the plot or table?

- Make sure plots and tables are appropriately sized and legible. If you think a plot (or anything in your report, for that matter) might not be legible when viewed on crowdmark, then it’s too small.

- Obviously, given that your code will be read by the grader, it should be neat, professional, well structured, and well commented code.

General Advice when Writing Reports

When writing any report, “**think like the customer**”. That is, think about what the customer, which is both MFE and the grader, would want to see in the report, what questions they’re going to have after reading the report, and what questions they’re going to ask you at a meeting when you present the report results (if one were to occur).[2] If you “think like the customer” you will likely be able to answer almost all potential questions before they’re even asked. As you write your report, ask yourself questions like “Will the customer know what I’m talking about, or is some background information needed?”, “Will this be clear to the customer?”, “Is the customer going to understand this plot? What do I need to write to *make them* understand the importance of this plot?”, “Is this plot clear/readable?”, “Will the customer care about this?”. If you ask these sorts of questions, and similar questions, as you write your report will be much better than if you write a report with a “I just got to get this done for my boss” type attitude.

[2] This is, in fact, a very important life skill. If you show up to a meeting with your team, with your boss, or with a customer, not having thought through what questions are going to be asked of you, the meeting is not going to go very well for you.