Patrick Marlatt; Austin LaFever; Frederick Peterson; Jonathan Wozny

FAMU-FSU College of Engineering  2525 Pottsdamer St. Tallahassee, FL. 32310

Team 513: Improving Engine Performance, Fuel Economy, and Emissions Through MIMO Engine Airpath Control

11/6/2020



Table of Contents

[Chapter One: EML 4551C 4](#_Toc55562546)

[1.1 Project Scope 4](#_Toc55562547)

[Project Description 4](#_Toc55562548)

[Key Goals 4](#_Toc55562549)

[Primary Market 4](#_Toc55562550)

[Secondary Market 4](#_Toc55562551)

[Assumptions 5](#_Toc55562552)

[Stake Holders 5](#_Toc55562553)

[1.2 Customer Needs 5](#_Toc55562554)

[Method 5](#_Toc55562555)

[Question-Answer-Need Table 5](#_Toc55562556)

[Synthesis 6](#_Toc55562557)

[1.3 Functional Decomposition 6](#_Toc55562558)

[Graphics 7](#_Toc55562559)

[Explanation of Functional Relationships 8](#_Toc55562560)

[1.4 Targets and Metrics 9](#_Toc55562561)

[Target Summary 9](#_Toc55562562)

[Critical Targets and Metrics 11](#_Toc55562563)

[Target Validation 12](#_Toc55562564)

[1.5 Concept Generation 12](#_Toc55562565)

[High Fidelity: 14](#_Toc55562566)

[Medium Fidelity: 14](#_Toc55562567)

[Appendices i](#_Toc55562568)

[Appendix A – Code of Conduct i](#_Toc55562569)

[Code of Conduct i](#_Toc55562570)

[Appendix B – Customer Needs iii](#_Toc55562571)

[Appendix C – Functional Decomposition iv](#_Toc55562572)

[Appendix D – Targets and Metrics v](#_Toc55562573)

[Appendix E – Concept Generation vii](#_Toc55562574)

# Chapter One: EML 4551C

## 1.1 Project Scope

### Project Description

The goal of this project is to create a control system that corrects the air-fuel ratio by moving the throttle and wastegates to improve engine performance.

### Key Goals

The key goal of this project is to create a 2 input, 2 output Model Predictive Control System to replace the current control system Mathworks and others use. Our MPC system is to be validated with GTPower engine analytics to ensure the air-fuel ratio agrees with the simulated value. The final goal is seamlessly integrating the system into the powertrain blockset dynamometer.

### Primary Market

The first primary market for our program would be engine manufacturers trying to optimize their engines. Our control system would allow the manufacturers to optimize engine performance. We can feel confident that this is a strong market, because General Motors has already attempted to implement this Model Predictive Control.

Our second market would be emissions testing agencies, because our control system will reduce fuel usage and pollution emissions, so they can use the program to test the benchmark.

A third market for our program would be GTPower and similar engine simulation technologies. They have state of the art engine analytics and virtual dynamometers, so if they find interest in our control system, the implementation into their simulation program could improve the quality of their software.

Our final target market is students and researchers. They may be working to find ways to

reduce emissions or improve performance, and our software package is a strong start to that. It’s a relatively cheap way to improve an engine, and they could use our control system as a start to designing a new one, or use it as a better way to test their new equipment.

### Secondary Market

Our secondary market could include any pedestrians. The optimized fuel consumption and reduced emissions would improve air quality, and reduce fossil fuel use, resulting in a better environment for everyone.

### Assumptions

Our first assumption is that everything else in the powertrain blockset is perfect. This reduces the need for a physical dynamometer, which reduces cost. It also reduces the need for testing of other components in the blockset.

Our second assumption is temporary, but will be used to establish a base for our control system. We will assume the system is single input, single output, and then add the additional input and output as our system functions correctly. This will reduce the load of things we need to learn at a time, while still getting a product prepared.

Our final assumption will be that an MPC is the best possible control system we can use. This will reduce research time, and allow us to dive straight into our design.

### Stake Holders

There are multiple stakeholders who will benefit from our successful completion of the project. A primary stakeholder for our project is Mathworks. Correct implementation of an MPC system would result in a more modernized and accurate model, making their Powertrain Blockset more appealing to auto manufacturers and Mathworks customers.

Secondly, Roberto Valenti and Peter Maloney are stakeholders as they are putting in their time to advise and assist us with our project. Additionally, ours is the pilot project for a new Mathworks program, headed by Valenti, that sponsors educational research projects, like ours, that are mutually beneficial to both Mathworks and students working on the projects.

Our advisors, Shayne McConomy and Koroush Shoele are stakeholders as well, since they will also be investing time into this project. Finally, FSU and FAMU are stakeholders. As a team, we represent the schools, and the schools therefore are impacted positively or negatively based on the outcome of the project. Successful completion will bring more status to the colleges. It may similarly result in continued project sponsorship by Mathworks and possibly an increase in the number of projects sponsored for the college.

## 1.2 Customer Needs

### Method

To get the needed information, we looked over the project statement and created a list of questions we felt were necessary to begin work on the project. We met our points of contact Roberto Valenti and Peter Maloney from our Mathworks, our sponsor, over Zoom for a “meet and greet.” They answered our questions, and we developed interpreted needs accordingly.

### Question-Answer-Need Table

|  |  |  |  |
| --- | --- | --- | --- |
| # | Question | Answer | Interpreted Need |
| 1 | Is it required to use model predictive control, or can we use controls that we are more familiar with, like PID, table-based, etc.? | We have been looking into MPC recently, and from our research, MPC seems to be the very best approach to solving the problem. | Use MPC control for project approach. |
| 2 | What will we be using as inputs and outputs to our control system? | A good start place to get familiar with simulink and the control system is the wastegate and throttle. If you start there, we can see what else you can add. | The control system implements the throttle and wastegate. Additional factors can be included. |
| 3 | How would we know if our control system is accurate? | Running the program produces a graph of desired intake vs actual intake. The closer the two lineup, the better your system. | The control system only demands what the system can provide. The blockset will produce a graph to ensure accuracy. |
| 4 | What is the main goal of this project? | The main goal is to integrate and simulate an MPC controller block in the powertrain block control set. The MPC should control the throttle and wastegate to get the ideal boost for the system. The new MPC controller designed by the group will replace the existing control system. | Create an MPC controller that will control the throttle and wastegate inputs to replace the existing outdated controller. |
| 5 | What is a good place to get started? | Play around with the power train blockset. Go through it, see how it all works; change the inputs and see what happens. |  |
| 6 | What can we do while we wait for the blockset? | There are apps on the market to record your car’s information [ on the throttle and wastegate], that information will help get started | Our model results can be compared to physical systems |
| 7 | What is the best way to simulate our controller within the system? | The powertrain blockset and Simulink can be used to simulate the system. We will look into getting the group access to GT Power as it has more specific engine models and GT Power is used the most in the industry for these types of simulations. | Use powertrain blockset and Simulink for simulations and perhaps GT Power later on in the project. |

### Synthesis

From our question and answer session we learned that the end-goal of this project is an MPC controller that regulates the throttle and wastegate. The controller simulation is to be done with Mathwork’s Powertrain Blockset, and the model’s results can be compared to physical systems and possibly GTPower simulations to test system efficiency.

## 1.3 Functional Decomposition

In order to understand and quantify the base functions that our project needs to fulfill in order to be a success, we created a functional decomposition of the project. To define each function, we tried to answer the question, "what does this project have to do?" We took each answer to that question and defined them as functions, then further broke them down into the lower level subsystems. Each subsystem relates to the main function under which it is defined. Each item is a quantifiable task; as each basic function is completed, the subsystem and the project as a whole become one step closer to a success. When every task is defined as successful, then the entire project will be considered a success.

### Graphics

Below is the functional decomposition of our Mathworks engine project. The first main function is control, as we have to manipulate the desired input values and design a controller that outputs values that align with the desired value. Secondly, we have to simulate the controller and model the system to show that the output torque aligns with the desired torque input, and measure time lag between input throttle/wastegate command and output throttle/wastegate position.

We generated these functions for our functional decomposition by discussing the project with our sponsor as well as amongst ourselves. Looking through the powertrain blockset gave us insight into how the system works and how the controller would fit into the system.

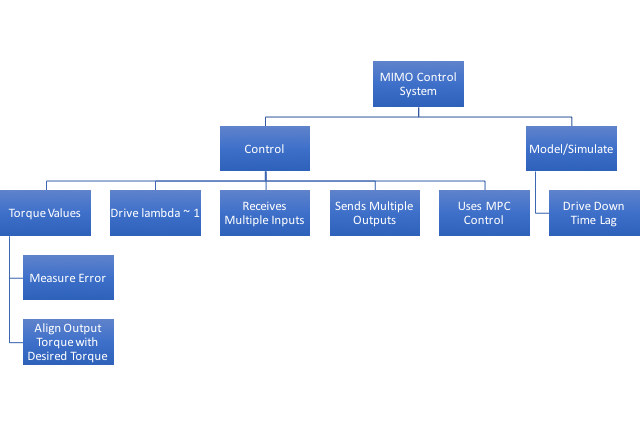


Figure 1: FD Hierarchy Chart for MIMO Control

The table below details the overlap between secondary sub-systems and how each relates to each function. While some may only relate to one function, there is opportunity to satisfy multiple functions through each subsystem completion.

Table 2

*Functional Decomposition Cross-Reference Table for MIMO Control*

|  |  |  |
| --- | --- | --- |
| **Function** | Control | Simulate/Model |
| Measure torque error | x |  |
| Align output torque with desired torque | x | x |
| Drive Lambda ~ 1 | x |  |
| Recieves multiple inputs | x | x |
| Sends multiple outputs | x |  |
| Use an MPC controller | x |  |
| Drive down time lag | x | x |

Because of the nature of the control system, many of the aspects overlap with the physical system. Many of the control systems boundaries are defined by the physics of the system, discussed further below. Additionally, our control system strives to improve the simulated models systems, by reducing time lag and error of torque output.

### Explanation of Functional Relationships

The overall transfer function of our control system will relate the desired torque to the output torque of the plant. The RPM of the engine is a limiting factor to our control system, and the current table based system asks too much of the plant when the engine is at a low RPM. Our model will improve the response of the system and demand appropriate torque from the system. Our control input and output will be the throttle and wastegate, and fluids equations for mass flow will be derived to be included in either the transfer function, or in a state-space set of equations.

The entire system can be broken down into our control system and the plant, or the simulation since it is fully virtual.

The control system will be multiple input, multiple output. Our initial inputs/outputs will be the desired throttle and wastegate angle, and the actual angle, respectively. In order to maximize accuracy and reduce error from external forces, a model predictive control (MPC) system is used. The MPC “looks into the future” to see what can quickly satisfy the requirements, and it is constantly checking itself. This makes it great at handling forces not included in the control system, and at driving the output to the desired value.

The simulated plant is where our boundaries come from. One of our primary goals in the plant is to drive lambda, the ratio of fuel to air, to 1. The plant also establishes our upper bound of torque, which our controller will have to respect. The plant’s driving equations for our control system will be the fluid flow through the throttle and wastegate.

The most important function is the control function. The controller will decide how the throttle and wastegate behave, and subsequently change how the torque output responds. In order to get the output torque to align with the desired output, the ‘control’ function must be fulfilled. As our functional decomposition cross-reference table shows, every single subsystem has a hand in the control side of the project, indicating further that the control function is the most important of the two.

## 1.4 Targets and Metrics

To create targets and metrics for each of our functions, the functional decomposition was modified (shown in Appendix C). We first re-labeled the ‘Control’ system as ‘Air Flow Control’ for specificity. The ‘Model/Simulate’ system was removed and two new systems were added in its place: ‘Integration’ and ‘Estimation Improvement’. Many of the old functions were kept, but they were broken down further and/or moved into a different system after further definition of our project.

After the necessary functions of the project were determined through the functional decomposition, targets and metrics were created for each function. A target is the measure that a certain function is designed to meet. The metric is a qualitative value that describes the target. For example, a target of 2 feet might have a metric of length, width, or height. Targets and metrics were created for each function in the new functional decomposition shown in in Appendix B and were placed in Table A1 in Appendix A. Some targets are termed critical targets and have been deemed more crucial to the project’s completion; they are presented below in Table 1.

### Target Summary

During the functional decomposition, the functions were grouped into three different systems: ‘Air Flow Control’, ‘Integration’, and ‘Estimation Improvement’.  Another system was added in Table  A1 called ‘Miscellaneous’; the functions under this system were additional needs not addressed in the functional decomposition.  These were also given targets to be satisfied.

There were ten functions under the system called ‘Air Flow Control’. These functions dealt with the specific components that were involved in controlling air flow into the engine. The project requires that our controller be multi-input and multi-output. The two components that can be used to satisfy this requirement are the throttle and the wastegate positions, which will be both the outputs and the inputs of our controller. As the goal is to simply include these functions in the controller, the target is whether they are inputs and outputs or not and will be represented as a Boolean; their state of satisfaction will be written as either a ‘1,’ meaning yes, or ‘0,’ meaning no.

The Mathworks Powertrain Blockset SI (Spark Ignition) Dynamometer is an application in Matlab that can be used to simulate and model different engines with varying parameters operating at different conditions. Our controller will be added into this program. The Powertrain Blockset allows the user to “resize” the engine being simulated to fit a desired engine’s parameters. There are multiple parameters, such as desired power or displacement and the number of cylinders in the engine that the user would like to model. The controller needs to operate for any parameter changes that the system allows. Since the evaluation is also a yes/no, this function metric is also Boolean.

There was only one function under the system ‘Integration’: works in conjunction with current feed forward controls. The Engine Dynamometer operates using feed forward table based controls, and it is required that our control system be a drop in modification rather than changing the current structure and controls. The target for this function is also a Boolean value. This function will be discussed further below since it is a critical target and metric.

The final system ‘Estimation Integration’ contains three functions with non-Boolean metrics. The functions are as follows: ‘Align Output Torque with Desired’, ‘Measure Error’, and ‘Drive Lambda ~1’. These three functions have non-Boolean values as their targets can be classified as goals of improvement for the simulation results of the current system. The metric for the output torque function is a percentage. It is required that the output torque aligns better with the desired; this cannot be done with a single torque value, so it is examined over a large range of torque values over the simulation and the average error will be used for comparisons. The function ‘Measure Error’ will be achieved in a similar way and will go hand in hand with the validation of the previously discussed function. For the function ‘Drive Lambda ~1’, the Greek letter lambda in this case represents the ratio of the actual air fuel ratio to the ideal stoichiometric air fuel ratio of 14.7 kg of Air to 1 kg of Fuel used for powering an engine. The ideal lambda value is 1, meaning there was exactly enough air to combust the fuel. Any value within the range between 0.8 to 1.2 is acceptable for this system, but our goal is to achieve a consistent lambda as close to 1 as possible. This range was determined by evaluating the average air fuel ratio of the current engine dynamometer controls, which was found to be 0.82. Any number closer to 1 than 0.82 for this specific simulation can satisfy this target.  While the average lambda is within our range, the goal is to create a system with a more constant lambda value close to 1.

The fourth function under ‘Estimation Integration’ is ‘Improve Time Response’. The metric for ‘Improve Time Response’ is also a Boolean evaluation, with the target being ‘1’, an improved time response. A better time response correlates with better control of the system. The time response of the system will be calculated with the desired torque as a steady state response for our system, and we will evaluate the time the transient output torque takes to reach the steady state torque value. As the time response is variable depending on the simulation and chosen parameters, and thus, varying time responses, a Boolean value was again chosen as a target. An improved time response will result in a successful function, and yield a ‘1’.

There are three other targets not included in the functional decomposition which are that the engine controller will work in both CI (diesel) and SI (gasoline) engine models in Simulink, that the controller will work in other engine simulation software and that the controller will also drive down error in the other engine simulation models. For the first target, the Boolean metric will be determined based on whether or not the simulation will run without errors for both the CI and SI engine models once our controller has been added to the models. The target will be evaluated as a ‘1’ for yes, and ‘0’ for no. This will be the same target for when the controller will be implemented in the other simulation software engine models as well. For the final target, the metric will be percent error between command torque and output torque of the engine model in the other simulation software. The target is that our controller will be able to improve error by 50%, similar to the target of the Powertrain Blockset. To determine how our controller affects the error between command and output torque, Matlab’s error rate functions will be used.

Table 1 below lists the targets and metrics whose satisfaction are most critical to the successful completion of the project.

#### Table 3: Critical Targets and Metrics

|  |  |  |  |
| --- | --- | --- | --- |
| **System** | **Function** | **Metric** | **Target** |
| Estimation Improvement | Align Output Torque with Desired | Percentage | 1.75% error |
| Integration | Works in conjunction with current feed forward controls | Boolean | 1 (unitless) |
| Air Flow Control | Multiple Inputs | Integer Count | 2 inputs |
| Air Flow Control | Multiple Outputs | Integer Count | 2 outputs |

### Critical Targets and Metrics

Our first critical target describes the accuracy of the torque with our added system, which results in several beneficial byproducts. This includes improving the fuel efficiency and emissions, paving the way for future year’s iterations of our design to remove the look-up tables, and reduce time for engine calibration by removing the table corrections.

The critical target we’ve set is used to validate the improvement in the accuracy of the current system. The current error values for a calibrated engine simulation can be found by running the Matlab engine simulation. Data values for the measured and commanded torque can be extracted from the simulation results; these values were used to calculate the current system’s error to average 3.5%, this will be done after the table values have been manipulated to cause error. This will be tested by taking the desired torque, comparing it to the torque our controller outputs, and comparing that error to the error the original system produced. The target was set at 1.75%. Our sponsors asked for “significant” improvement, and half the current error value was decided as satisfying the condition “significant” while not overcommitting and attempting to achieve the impossible.

The second critical target is for the function ‘Works in Conjunction with Current Feed-Forward Controls’. Satisfying this target is crucial for the project’s success. The current feed forward table-based controls in the Powertrain Blockset do a good job already at controlling the torque; our controller will be added in to make the simulation even more accurate, especially under varying conditions. If the controller is unable to operate with the current controls, then there is little value in dropping it into the Powertrain Blockset. The reasoning used to come up with this target was simple: either it works with the current feed forward controls or it does not, and therefore, a Boolean value of ‘1’ for yes, or ‘0’ for no, were used to quantify satisfaction of this need.

The third and fourth critical targets and metrics are the requirements of controlling multiple inputs and multiple outputs. These were chosen over their respective lower level functions of controlling specifically both the throttle and the wastegate because controlling multiple inputs and multiple outputs were the actual demands of our sponsors, even though there were four possible components to control: throttle, wastegate, and intake and exhaust cams (for timing of letting air enter the engine cylinders). The value of 2 was chosen as a target because the purpose of this is to switch from single input single output to multi-input multi-output, and two inputs and two outputs is the minimum number of inputs and outputs to be considered “multiple.”

### Target Validation

In brief, our goal is to create a control system to append to the table system in the Powertrain Blockset. Our primary metrics of determining the success of our system is the error of our system and if the system works when added to the current system. The latter is a Boolean value, being that it works or it does not work in conjunction with the current system.  Validating the functions that involve the input/output controller requirements will be done by simply going through the list and checking off whether the controller operates for these requirements. The functions of differing parameters will be validated through two ways. First, it will be tested to see if the controller can operate when these parameters are changed. Second, the parameters will be changed for the original Engine Dynamometer and then for the Engine Dynamometer with our dropped-in controller. The output torques will be compared qualitatively to see if similar torque values were created.

Validation of whether the new controller works with the current feed forward controls will be done in a similar manner to the input/output functions described previously. The measure of our controller’s performance can be determined by the error of the system compared to the error of the old system. Because the system is entirely virtual, measurement of values is built in. We can attach scopes to various points of the system to get basic measurements, and have Matlab display more specific values on high accuracy targets. The Engine Dynamometer simulation will be done with and without our controller, and the error between the commanded and output torque will be calculated using Matlab’s error rate functions. The time response of the system will be calculated using the commanded torque values as steady state values and evaluating the transient time response of the engine’s torque output. This will be done using Matlab’s MIMO time response evaluation tools.

## 1.5 Concept Generation

A variety of methods were used to create concepts in our concept generation. The first method used was a morphological chart. For each individual concept, we made one selection from each column, and the resulting four choices became a single concept. It is important to note for the morphological chart in Table 1 below, that the last two columns were used interchangeably based on what controller type was selected. For MPC, there were different types of programs to choose from, but for the PID and feed forward controller types, there is just an option on whether to linearize or not. Each concept is created by choosing different options in each column as you move through the table. As this method is a structured way for creating a useful and wide variety of control approaches, it was our main source of ideas for the one-hundred required concepts, totaling fifty ideas.

#### Table 4: Morphological Chart

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input/Output** | **Controller Type** | **Location in Blockset** | **Program**  (for MPC) | **Linearize**  (for non-MPC) |
| SISO | MPC | Before Table | Linear | yes |
| MIMO | PID | After Table | Quadratic |
| SIMO | Feed forward | Replace tables | Nonlinear | no |
| MISO |

Another method that provided a few ideas was creating and solving multiple “Anti-Problems.”  We came up with this by considering how we would reduce control over the engine simulation. Solutions to this were changing the engine settings and changing the table values (used for Mathworks’ current table-based control). Two ideas that came from these were to improve the response to variation and get new table values with our own experimentation. About five to ten ideas were ideated using this exercise.

A technique called “Battle of Perspectives” was used. In this method, the team was broken up into groups of two from arbitrary categorizations. The following are examples of groupings: cat vs dog people, cars vs non-car members, and controls vs non-controls students. There are only four team members, so there were not too many combinations of group members that could be done. Using this method, about twenty concepts were created.

The fourth and final method of ideation was through how one might generally think of ideation. This involved individual thought as well as group discussion, presenting any random or interesting ideas to the rest of the group and recording them. This filled in the remaining gaps with about twenty to twenty-five, adding up to about one hundred ideas in total.

To narrow our one-hundred concepts to just eight, we first qualitatively eliminated ideas based on what would be possible or reasonable in our timeline. We further narrowed ideas down by examining and analyzing the remaining concepts using our customer needs. If a concept did not include the correct amounts of inputs and outputs, we removed that concept. Secondly, we removed any concepts that were unrealistic, required equipment or a budget that was not possible to attain, or would take an unreasonable amount of time.

Through this processing of narrowing down the ideas, high fidelity and medium fidelity concepts were selected. These are each described below, and reasoning for placing them in each category is explained.

The concept behind each idea is fairly similar but with added complications and different placements in the current controls. The placement labeled “replacing tables” would involve replacing the current throttle and wastegate controls with our controller, while the one called “after tables” would be a controller taking the output throttle and wastegate positions and letting these values run through our controller to clean up control. This is more in line with what our sponsor would want.

Non-linear control would involve the most accurate models of our system. While moving down to quadratic and then linear modeling would decrease the simulation run time and computational load, but would decrease the overall accuracy.

**NOTE:** Before we delve into the different Model Predictive Control (MPC) options, the definitions should be noted. The MPCs are defined by the governing characteristics, which are the equations it uses to calculate, and the constraints. If both are linear, it is a linear system. If the functions are quadratic, it is quadratic. Any other combination is considered a non-linear system. An overall benefit of MPC is its ability to operate with given constraints in an optimal fashion.

**NOTE:** As our project is completely virtual and solutions only involve mathematical and control models, there were no sketches used to represent the high and medium fidelity concepts.

### High Fidelity:

Local Trajectory Optimization - A Mathematical calculation to minimize the output of a function. Very quick, but only finds the local minimum, which is not always the absolute best answer. It does not have the capability to make calculations regarding future changes, and as such won’t adjust. A major advantage of this is the speed that it operates at.

Quadratic MIMO MPC After Tables - A MIMO MPC satisfies the explicit request of the customer. The category of quadratic means that we linearize the constraints of the function. This reduces the calculation time at a slight cost of accuracy. Placing it after the tables allows it to be a “drop-in” modification with better satisfies the customer’s request.

Non-linear MIMO MPC Replacing Tables - in non linear, no simplifications are made. This should improve accuracy to the  highest degree, but at the cost of significantly slower run times. This idea involves replacing some of the current controls for the throttle and wastegate completely.

### Medium Fidelity:

Linear MIMO MPC Replacing Tables - Linearizing both the constraints and the functions speeds up the program, at a cost of less accuracy.

Linear MIMO MPC After Tables - Similar as previous linear MPC, but adding them after the tables satisfies another request from the customer. Adds more difficulty to the project.

Quadratic MIMO MPC Replacing Tables - Quadratic will be slower but more accurate than linear control. Placing the controller after the table better satisfies customer request

Non-linear MIMO MPC After Tables - Non-linear is the slowest but most accurate. “After tables” better satisfies customer request.

Global Trajectory Optimization - Due to the difficulty of finding the absolute lowest value of a function,  brute force is required. This takes longer, but produces more accurate results. However, sometimes local minima already produce the best result, so this is placed lower in fidelity because the cost may outweigh the reward.

# Appendices

## Appendix A – Code of Conduct

### Code of Conduct

**Mission Statement**

The members of team 513 will strive to create work that the members, the FAMU FSU College of Engineering, and the sponsor, Mathworks, will be proud to have their names on. Work will be completed with quality and timeliness.

**Ethics**

Team 513 will adhere completely to the National Society of Professional Engineers Code of Ethics. All assignments, decisions, issues, and discussions will be carried out with honesty and fairness. All commitments made will be honored and carried out with integrity. Any violations will be brought to the entire team and any repercussions will be decided as a whole.

**Team Roles**

* Austin LaFever - Controls & Simulink Engineer
  + Design the system inputs and outputs in Simulink
* Create a preliminary control system for the project
* Find a controls professor to assist in MIMO design
* Frederick Peterson - Matlab Engineer
  + Develop code for control system for the project
* Take attendance at team meetings
* Jonathan Wozny - Matlab and Thermal Fluids Engineer
  + Develop code
* Complete any necessary calculations for design involving thermal fluids
* Patrick Marlatt - Project Engineer
  + Review and turn in work, main contact with sponsor, help with code and controls
* Miscellaneous work will be assigned by the project engineer on a case-by-case basis depending on the current workload of the group members.

**Communication**

* GroupMe and Basecamp will be the main methods of communication. Github will be the main method for sharing and collaborating on the Matlab code.
* Zoom will be utilized as the main method of conducting meetings. In-person meetings may occur occasionally only for those who are okay with it and who have no COVID-19 symptoms. COVID-19 precautions will be followed at all meetings.
* Documents and assignments that will be turned in will be shared and tracked through Basecamp.
* Acknowledgement of any messages must be done within 24 hours. Failure to do so will result in team discipline and/or spontaneous combustion.

**Project Submission**

All work to be submitted must be given to the project engineer a minimum of 48 hours in advance of the due date. If any problems occur, the project engineer must be notified ASAP. If for any reason the project engineer is unable to submit the assignment, Jonathan will submit the assignment in his place.

Once the assignment has been submitted, the project engineer will notify the group via GroupMe, as well as send an image of the receipt saying it is turned in to confirm submission.

**Dress Code**

The dress code for the various meeting types are as follows:

* Group meetings: Casual attire
* Group meetings (in person): Casual attire w/ mask
* Meetings with advisors/sponsors: Collared shirts
* Presentations: Dress shirts, as uniform as possible among the team

**Attendance Policy**

* Group meetings will be held every Tuesday and Thursday during the extra time in Senior Design. There will be additional meetings on Wednesdays at a time decided that week. If needed, a Friday meeting will be called and set the same way.
* Attendance to all meetings is required unless the member lets the group know at least 24 hours in advance.
* Attendance of group meetings will be recorded by Frederick and added into an excel sheet that is available for everyone to see in Basecamp.
* Meetings with the group advisor will be held weekly on Wednesdays.
* If for any reason a team member misses more than three team meetings, the project engineer will reach out and see if there is anything that can be done to make it easier to attend team meetings, such as adjusting certain meeting times. If the offences continue, then the team will handle it as outlined in the conflict resolution section.

**Decision-Making**

All members will participate in the decision-making process. If there are any conflicts of interest, team members must state this and disqualify themselves from the decision-making process. The entire decision-making process is outlined below:

* Define the problem
* Propose solutions
* Research
* Testing
* Presenting results to the group

The solutions will then be evaluated and a vote will occur. If there is still indecision, a tie, etc., the problem will be presented to the project advisor.

**Conflict Resolution**

All conflict will be documented and signed by the project engineer and the team member(s) involved. Each member has the right to present their case to the rest of the group without any interruptions. All conflicts will be solved by majority rule. Any conflict that cannot be resolved internally will be brought to Dr. McConomy, who has the final decision.

**Amending the Code of Conduct**

If any additions or changes are required for Code of Conduct, a team meeting will be held and the changes will be discussed and administered if agreed upon by every member. The Code of Conduct document will be amended and then re-uploaded.

**Statement of Understanding**

By signing the lines below, I agree to and will abide by the above guidelines of this code of conduct.

Name:\_\_Austin LaFever\_\_\_\_\_\_ Date: \_\_9/11/2020\_\_

Name: \_\_Frederick Peterson\_\_\_ Date: \_\_9/11/2020\_\_

Name: \_\_Jonathan Wozny\_\_\_\_\_ Date: \_\_9/11/2020\_\_

Name: \_\_Patrick Marlatt\_\_\_\_\_\_ Date: \_\_9/11/2020\_\_

## Appendix B – Customer Needs

Table B1: Question-Answer-Needs Chart

|  |  |  |  |
| --- | --- | --- | --- |
| # | Question | Answer | Interpreted Need |
| 1 | Is it required to use model predictive control, or can we use controls that we are more familiar with, like PID, table-based, etc.? | We have been looking into MPC recently, and from our research, MPC seems to be the very best approach to solving the problem. | Use MPC control for project approach. |
| 2 | What will we be using as inputs and outputs to our control system? | A good start place to get familiar with simulink and the control system is the wastegate and throttle. If you start there, we can see what else you can add. | The control system implements the throttle and wastegate. Additional factors can be included. |
| 3 | How would we know if our control system is accurate? | Running the program produces a graph of desired intake vs actual intake. The closer the two lineup, the better your system. | The control system only demands what the system can provide. The blockset will produce a graph to ensure accuracy. |
| 4 | What is the main goal of this project? | The main goal is to integrate and simulate an MPC controller block in the powertrain block control set. The MPC should control the throttle and wastegate to get the ideal boost for the system. The new MPC controller designed by the group will replace the existing control system. | Create an MPC controller that will control the throttle and wastegate inputs to replace the existing outdated controller. |
| 5 | What is a good place to get started? | Play around with the power train blockset. Go through it, see how it all works; change the inputs and see what happens. |  |
| 6 | What can we do while we wait for the blockset? | There are apps on the market to record your car’s information [ on the throttle and wastegate], that information will help get started | Our model results can be compared to physical systems |
| 7 | What is the best way to simulate our controller within the system? | The powertrain blockset and Simulink can be used to simulate the system. We will look into getting the group access to GT Power as it has more specific engine models and GT Power is used the most in the industry for these types of simulations. | Use powertrain blockset and Simulink for simulations and perhaps GT Power later on in the project. |

## Appendix C – Functional Decomposition

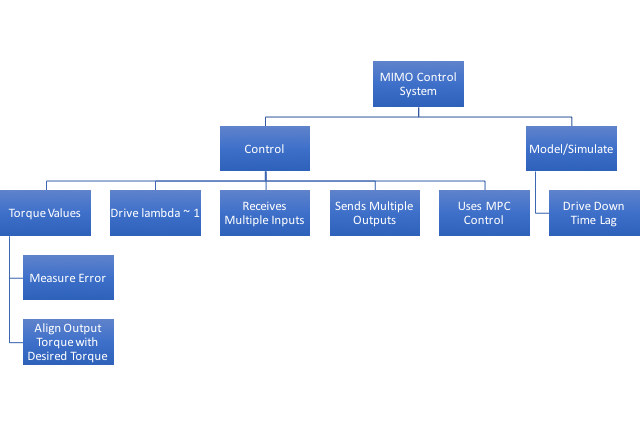


Figure C1: Previous Functional Decomp. Hierarchy Chart

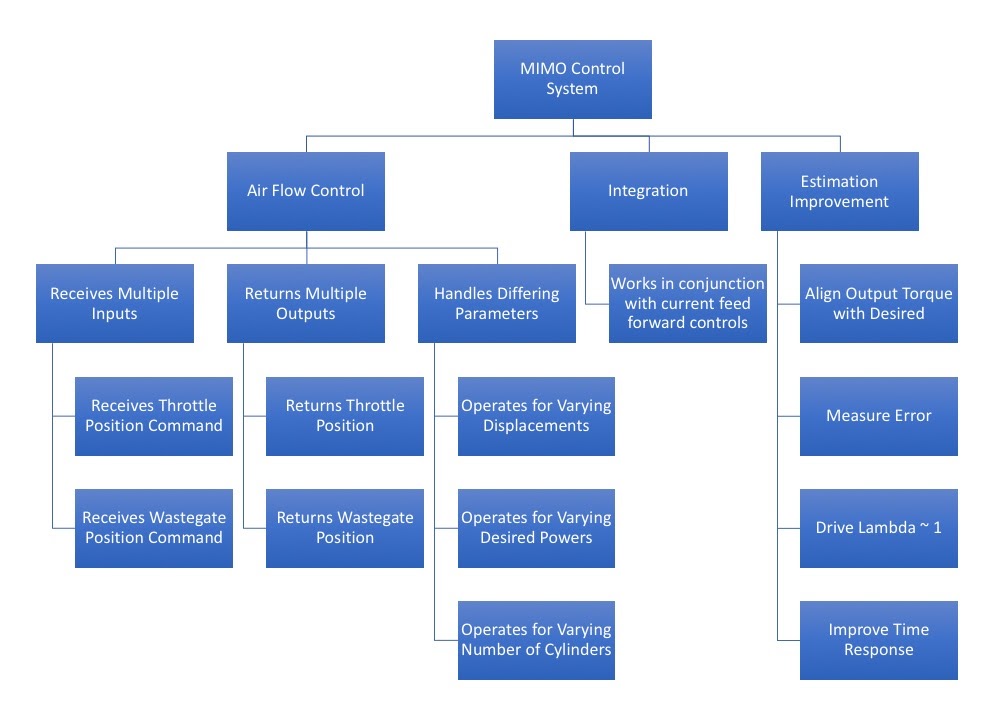


Figure C2: New Functional Decomposition Hierarchy Chart (as of 10/30/2020)

## Appendix D – Targets and Metrics

Table D1: Target and Metric Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **System** | **Item Number** | **Function** | **Metric** | **Target** |
| Integration | 1 | Works in Conjunction with Feed-forward | Boolean | 1 |
| Estimation Improvement | 2 | Improve Time Response | Boolean | 1 |
| 3 | Reduce Torque Error | Percentage | 50% Error Improvement |
| 4 | Align Output Torque with Desired | Percentage | 1.75% |
| 5 | Drive Lambda ~1 | Unitless ratio | 1 +/- 0.18  (kgAir/kgFuel)/(kgAir/kgFuel) |
| Air Flow Control | 6 | Recieves Multiple Inputs | Integer Count | 2 inputs |
| 7 | Receives Throttle Position | Boolean | 1 |
| 8 | Receives Wastegate Position | Boolean | 1 |
| 9 | Returns Multiple Outputs | Integer Count | 2 outputs |
| 10 | Returns Throttle Position | Boolean | 1 |
| 11 | Returns Wastegate Position | Boolean | 1 |
| 12 | Parameters can be Changed | Boolean | 1 |
| 13 | Varying Displacement | Boolean | 1 |
| 14 | Varying Desired Torque | Boolean | 1 |
| 15 | Varying Cylinders | Boolean | 1 |
| Miscellaneous | 16 | Works with Both SI and CI Engine Models | Boolean | 1 |
| 17 | Controller Works in Other Simulation Model | Boolean | 1 |
| 18 | Drives Down Error in Other Simulation Models | Percent Error | 50% Error Improvement |

## Appendix E – Concept Generation

Table E1: Morphological Chart

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input/Output** | **Controller Type** | **Location in Blockset** | **Program**  (for MPC) | **Linearize**  (for non-MPC) |
| SISO | MPC | Before Table | Linear | yes |
| MIMO | PID | After Table | Quadratic |
| SIMO | Feed forward | Replace tables | Nonlinear | no |
| MISO |