SSA 1

Matlab & Github

Thomas Driessen & Mihai-Dragos Ungureanu

February 8, 2021

Goal

- Get to know how to use Github.
- Look into the given training data for Matlab.
- Begin on working on an Matlab script for this project.

Conclusion

- Github is made by Mihai.
- The basics of using Github are known .
- The given training scripts and data are analysed en runned.

Problems

- The first problem for Thomas was the fact that he had to install the Signal Processing Toolbox to run the given examples. So for every member, check if you have this toolbox!
- In this SSA most of the time (for Thomas since Mihai focussed on Github) was spent on understanding all the matlab script.
- Also it took a lot of time to define how the theoretical model will be written in a proper way.
- A way to integrate Github directly into Canvas Student was researched (that will automatically push every file into Canvas) but no such program exist.

Follow up Steps

- Finnishing the theoretical thermodynamic model in Matlab. Making a full script which can read measured data with the same variables as the theoretical model.

Work Division

- Mihai was creating Github and worked this out.
- Thomas looked at the given example data and tried to find a way to begin with the Matlab script.

Time Division

- Understanding, running and explaining (in this document) the example scripts. (2 hours) (Thomas)
- Running and understanding the script of Vito, as this was also usefull for understanding further stages of writing the matlab script. (2 hour) (Thomas)
- Looking into some examples on the internet on how to run en write thermodynamics into Matlab, such as plotting a PV-diagram. (1 hour Thomas)
- Create the Github Repository and add all the files already available on Canvas. (1 hour) (Mihai)

- Make the Github and Trello integrations into Gitkraken and creating a synced Trello board with the Github repository. (2.5 hours) (Mihai)
- Create the Installation guide for other Team members (1 hour)
- Making a RPC list (0.5 hour) (Thomas and Mihai each person)

Overleaf Link

1 The training data and script

The training data for Matlab consist of three different scripts, which all have different applications. One script called "ConvertFiles" is written in such a way to to convert text data into an useful data set for Matlab. This script will be also used in the final script for the experiment, as there can be assumed that the measuring data will be collected as a text file which uses comma's in data. Much applications for Matlab can only be used when first a text file with data is converted, since Matlab itself is not able to read such data in a text file in a proper way. One thing which will change when converting the text file, is that a number with an "," in it will be change to an ".".

The second Matlab script is defined as "NasaUseExample". This script uses a database in which thermal properties of certain types of molecules can be found, determined by NASA. In this example, first all alcoholic molecules are selected with the use of a for loop. After this the C_p and the C_v values of these alcoholic species will be plotted over time. These plots can be seen in figure 1. The dotted lines in the plot are the C_v values. Also γ is plotted which is defined as $\frac{C_p}{C_v}$. In the plot there can be seen and concluded that the C_p and C_v values will increase at higher temperatures. The ratio between C_p and C_v will decrease at higher temperatures. These variables, C_p,C_v and γ will be used in the final theoretical Matlab script for certain thermodynamic processes.

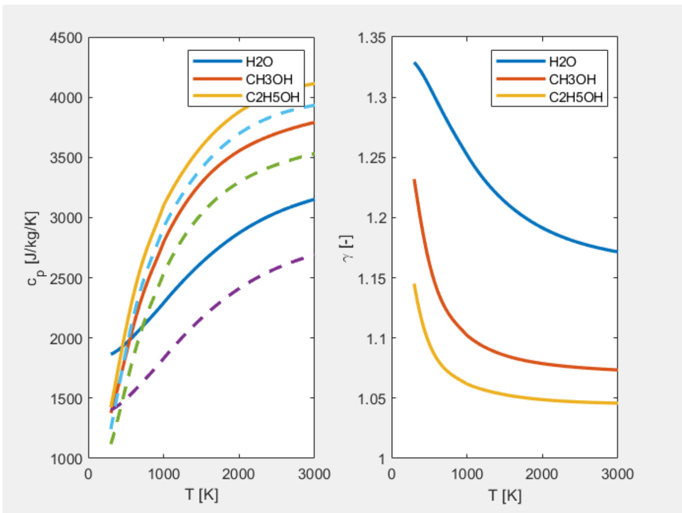

Figure 1: C_p, C_v and γ values of the example data.

The third given example script is defined as "ReadCaseDataExample". This script will read the converted data text file. Three different data sets are given; full load, half load and no load. The data in the text file exists of the; time [s], pulse[V], sensor pressure signal [V]. A subscript called "ImportData4GB10" is used to import this data in such a way that only full cycles are used. For each data set the pulse and sensor pulse is plotted against the time. Also the double tooth moments are indicated in these plots. These moments are determined such that the differential of the pulses between two peaks is the highest. These plots can be seen in figure 3.

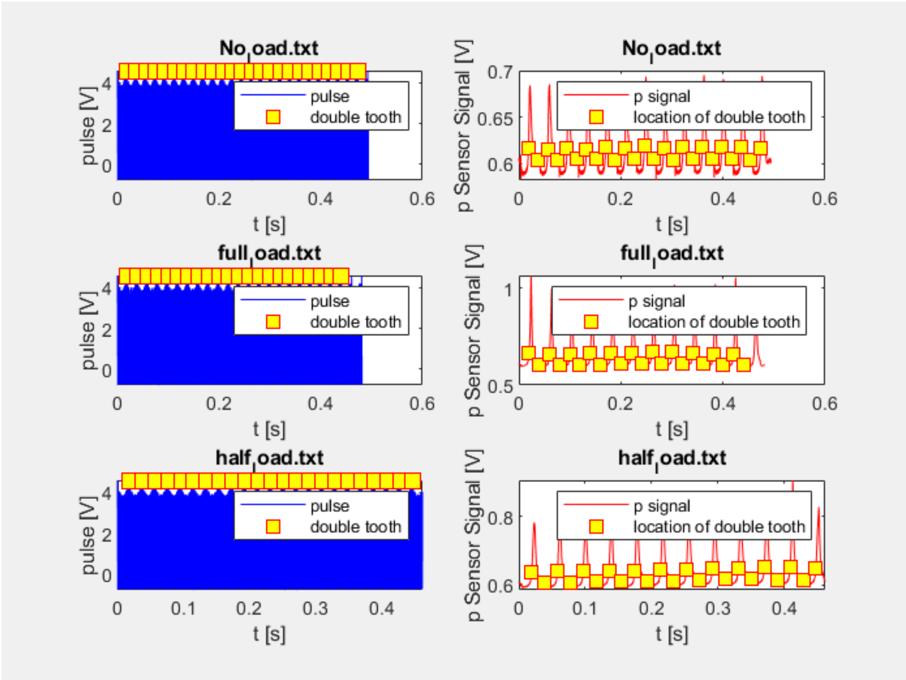

Figure 2: The pulses with the double tooth moments plotted against the time.

The training data gives some useful insights in how to collect measured data and analyse this. One important aspect which was found when running the code for the first time is the fact that the "Signal Processing Toolbox" is required. This toolbox can be downloaded and installed on MatWorks.

1.1 Further steps for Matlab

Vito already uploaded a great script where a lot of variables for the theoretical script are defined. After discussing with Mihai about the Matlab script we concluded that it would be useful to write script which can plot the PV-diagram of the system theoretically. This is decided because of the fact that when plotting the theoretical PV-diagram and the PV-diagram out of the measurements, there can be easily seen how accurate the theoretical script is. To realise this the four different stages of the cycly will be plotted. This are two adiabatic processes and two isochoric processes. First with thermodynamics relations for these processes will be formulated in such a way that these are functions for the pressure in terms of the volume, since this is what will be plotted into the PV-diagram.

An example of this can be seen below, this is a very simplic version but this can give some insights on how to work further on the script. In the refereces some examples can be found.

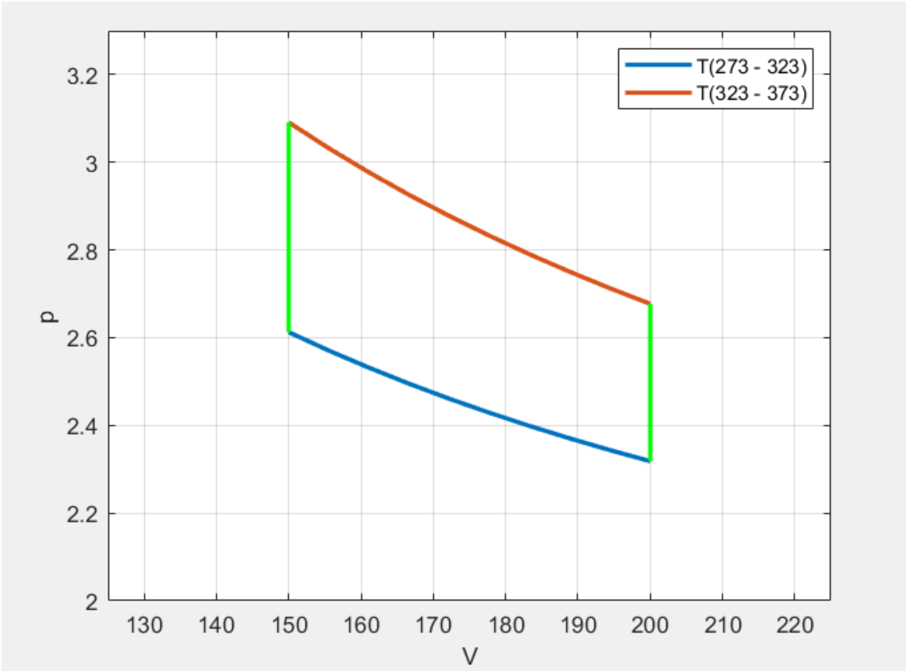

Figure 3: An example of a PV-diagram of an Otto cycle written in Matlab.

2 Github

2.1 Setting up Github

During the previous meeting it has been decided to use Github in order to have a better workflow between the members of the team regarding the Matlab Code. Therefore, we have implemented the Github repository [1] and added all the members to it. As Github is quite easy to use and also to update files on it, we have decided to keep the same structure as in Canvas and instead of adding just the Matlab code to it, to add instead all our files in there.

2.2 Trello $->$ Github Integration = GitKraken

In order to have a better structure in our project it has been decided to use Trello. The Trello board has been implemented by Mats and Dolf. In order to have everything in one app (both the plans and the code) it has been decided to integrate both Trello and Github into Gitkraken. Gitkraken app works in the same principle as Github Desktop (PUSH & PULL) but it has extra features such as a Trelo board integration that can directly be synced with the Github by using the Issues tab, see figure 4. More onn this will be explained at the next sub-section. Right now, the GitKraken has implemented both the Trello boards and Github repository.

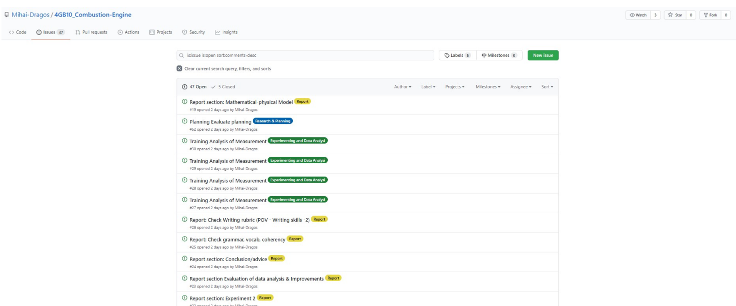

Figure 4: Issue tab in Github directly synced with Trello board

2.3 Gitkraken Features

Gitkraken has many features compared to only using the Github Desktop app. This section will mainly describe the advantages of this platform and why we think it will be better to use it for better workflow.

Feature List:

- Easy Trello & Github integration

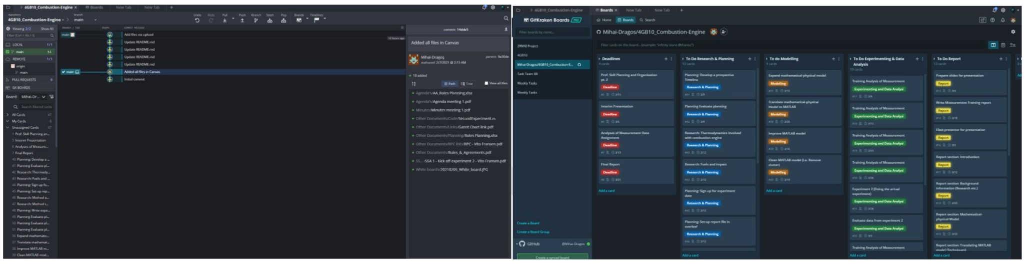

Figure 5: Github and Trello, GitKraken Integration

- Possibility to display the Trello task into a Calendar or Timeline

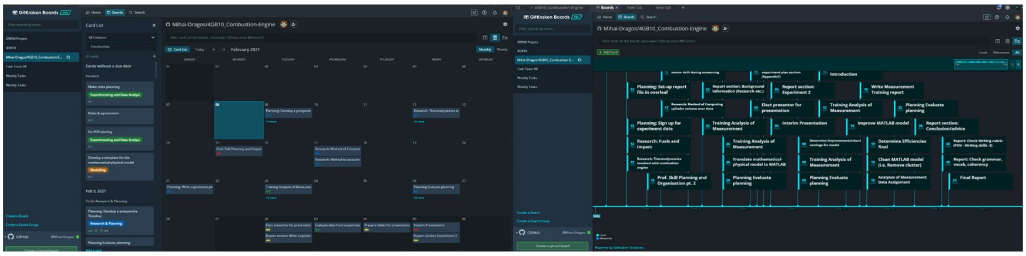

Figure 6: Trello Tasks show in Calendar and Timeline

- Possibility to link a Trello task to a code that will be pushed on the server
- All in one application (everything is in one application)

Due to all these benefits that this app will bring we have thought that it will help us a lot in working together on the code, planning the project and futures tasks while also keeping a good collaboration between each others and assigning peoples to certain tasks.

2.4 Installation Guide

In order to make the installation as quickly as possible, a guide has been created in order to show to the other team members how to install GitKraken, add the Github repository to this app and integrate the Trello boards. The guide is available on $Canvas- > OtherDocuments- > Guide$

3 RPC lists

3.1 RPC list Thomas

Requirements
The model must give a PV-diagram within an accuracy of 10%.
The model must give a thermodynamic efficiency within an accuracy of 10%.
The model must give an emission value within an accuracy of 10%.
Themodel must be able to implement measured data the same variables as thetheoretical description.
The model must be able to hold the requirements above for gasoline and bioethanol.
Preferences
The model can show the theoretical PV-diagram with the measured diagram within the same graph.
The variables of the theoretical model can be easily adjust in the model.
A list of different fuels can be selected to simulate in the model.
Constrains
The physical experiment can only take two hours.
The physical experiment can only be done by two persons.
The model must written in Matlab.

3.2 RPC list Mihai

Requirements
At least 2 different type of fuels will be analyzed during this project.
The CO_2 emission must be reduced by 15%.
An efficiency of around 25% must be achieved for the motor which is closer to reality.
The model must not contain any error and the Matlab code should run without any fault.
Preferences
The model should be as close as possible to the actual real engine.
Different variables should be implemented in a way that could be adjustable by the user.
A GUI (Graphical User Interface) might be implemented in order to facilitate the change of certain parameters.
The model should have the ability to plot 3 different graphs and export the data as a .CSV file.
Constrains
The fuel types that could be used are limited (only gasoline and ethanol so no diesel).
Only 6 weeks to implement everything on the model and 7 weeks and a few days to write the final report.
Only specific time slots to do the experiments and get experimental data.

References

[1] https://github.com/Mihai-Dragos/4GB10_Combustion-Engine

[2] https://nl.mathworks.com/matlabcentral/answers/489174-plotting-a-p-v-diagram

[3] https://skill-lync.com/projects/OTTO-CYCLE-with-P-V-diagram-using-Matlab-43652