HYBRID VEHICLE CONTROL STRATEGY IMPLEMENTATION ANALYSIS

Zihao Xu
Purdue University
School of Mechanical Engineering

Spring 2020

ME 49800

Table of Contents

ABSTRACT

Control strategies for hybrid vehicles have been developed and tested in simulation. The implementation of control strategies – ECMS and Regression – on the vehicle failed because of motor oscillations that occurred during implementation. Study of the result of successful PSOC experiments provides a clear view of what the relationship between simulation and implementation should be, which enables us to use the simulation result as a benchmark for implementation result. By tracking the signals showing oscillations in the implementation of ECMS and Regression will then lead to the reason of motor oscillation. Checking the same signals in PSOC implementation can make sure if the oscillations are normal or not.

1 INTRODUCTION

1.1 Control strategies for hybrid vehicle

While the transportation sector is still one of the largest contributors to the global greenhouse gas (GHG) emissions, hybrid vehicles and electric vehicles have become the most popular solutions to reduce greenhouse gas emissions since they use alternative energy sources other than fossil fuels. On one hand, powering the vehicle by electricity enables the vehicles to operate with less carbon emission and higher efficiency. On the other hand, the range of vehicles only powered by electricity is limited because of the cost and limited capacity of batteries. As a result, a hybrid vehicle becomes the best solution when driving range and efficiency are both taken into consideration on account of its two different main power sources, engine powered by fuels and electric motor powered by batteries.

When these two different main power sources provide the vehicle with a lot of possibilities, it's essential to find out a best real-time optimization-based power management strategy to tell the vehicle when to use the electric motor only, when to use the engine only and when to use both of them and how to make them cooperate with each other. Some preliminary work was carried out in the past few years and a variety of power management strategies have been studied, including Dynamic programming, Equivalent Consumption Minimization Strategy (ECMS), Proportional State of Charge Algorithm (PSOC) and Regression Modeling [1].

1.2 Implementation of control strategies

After the simulation of these control strategies gave satisfactory results, some trials for implementation were carried out to see if these strategies can work well on real hybrid vehicles. Sauradeep in 2019 was the first to design and operate those experiments for implementation [2]. He tried to implement the PSOC, ECMS and Regression control strategies on the real vehicle. The vehicle used for testing is a 2013 Chevrolet Malibu, which was developed in the EcoCAR2 competition in 2014. It has a 1.7L turbo diesel engine and a 100-kW electric motor from Magna. It was tested on a dynamometer to get accurate states of the vehicle such as wheel torques and speeds.

According to his report, implementation of PSOC worked well while both implementations of Regression and ECMS failed because of motor fluctuations that occurred when testing, as shown in figure 1-1 and figure 1-2.

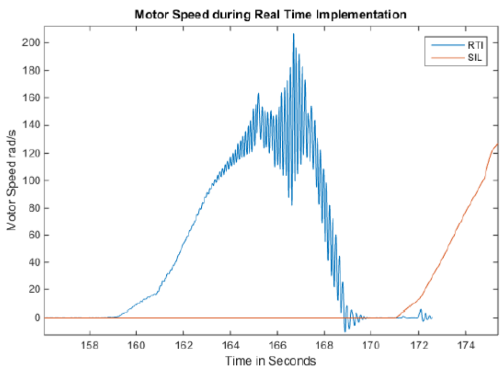

Figure 1-1 Oscillations in Motor Speed for Regression [2]

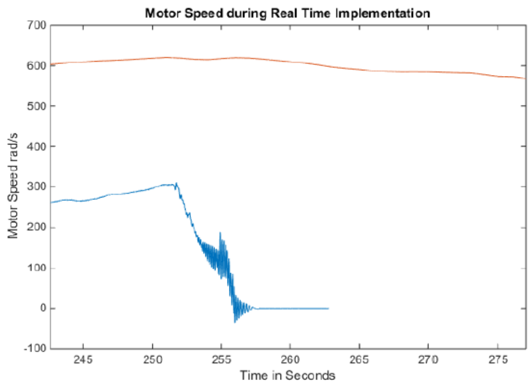

Figure 1-2 Oscillations in Motor Speed for ECMS [2]

To avoid any unwanted damage to the car, the implementation was stopped as soon as the oscillations were observed. The reason for these oscillations was noted as unknown in the report. However, it is essential to figure out where these oscillations come from if a successful implementation of ECMS or Regression is still wanted. This will be figured out in the next few chapters of this report.

1.3 Distribution of report content

This report consists of five chapters:

Chapter 1 is the introduction to the purpose and background of this report. It contains the introduction to hybrid vehicles and its control strategy as well as the introduction to the result of implementation trials for the control strategies.

Chapter 2 shows the relationship between the simulation result and implementation result to give a definition of a successful implementation. Since only PSOC worked well in implementation, this chapter will be based on the comparison between the results of PSOC simulation and implementation.

Chapter 3 gives a more detailed description of the implementation results of ECMS and PSOC, including all the signals that are useful. By plotting out all the unusual signals the reason for the oscillation that occurred can be found.

Chapter 4 provides a conclusion on possible reasons and some recommendations for future work.

2 RELATIONSHIP BETWEEN SIMULATION AND IMPLEMENTATION

Since it was unclear how the Simulink codes were implemented on the hybrid vehicle, it makes sense to check whether different codes or drive cycles were used in implementation, which could probably explain these oscillations. Therefore, it is necessary to check if there is any difference between the model in simulation and the reality in implementation.

2.1 Relationship between the codes

Several versions of implementation Simulink model can be found on the laptop for the experiments. By taking an overview of its whole system, it is not hard to find that the simulation version code is just a simplified version of the 'supervisory' part of the whole system for implementation, as shown in figure 2-1 and figure 2-2.

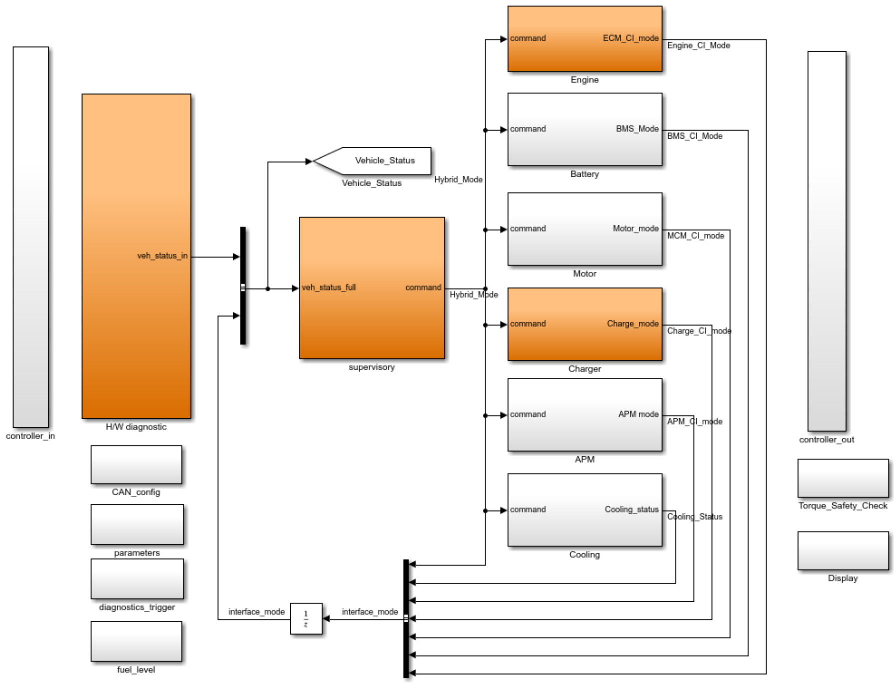

Figure 2-1 Whole system of the implementation Simulink model

And here is the whole system of the simulation version.

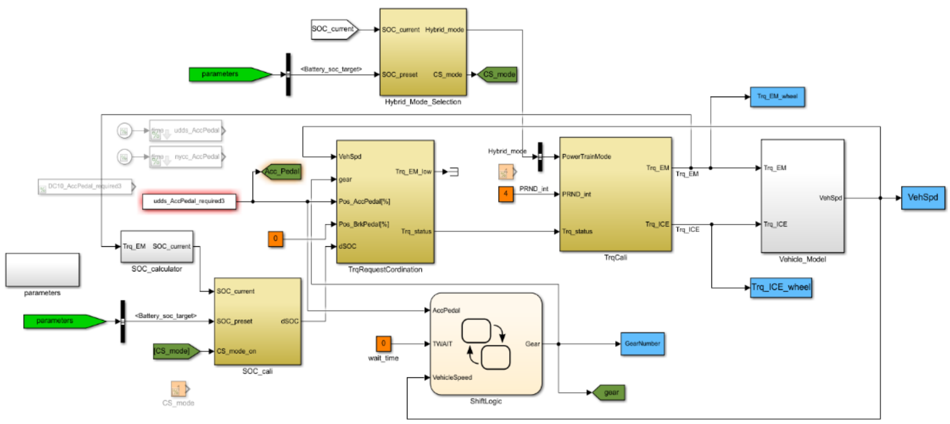

Figure 2-2 Overview of the simulation Simulink model

Figure 2-3 is the overview of the 'supervisory' block in the implementation version of Simulink model.

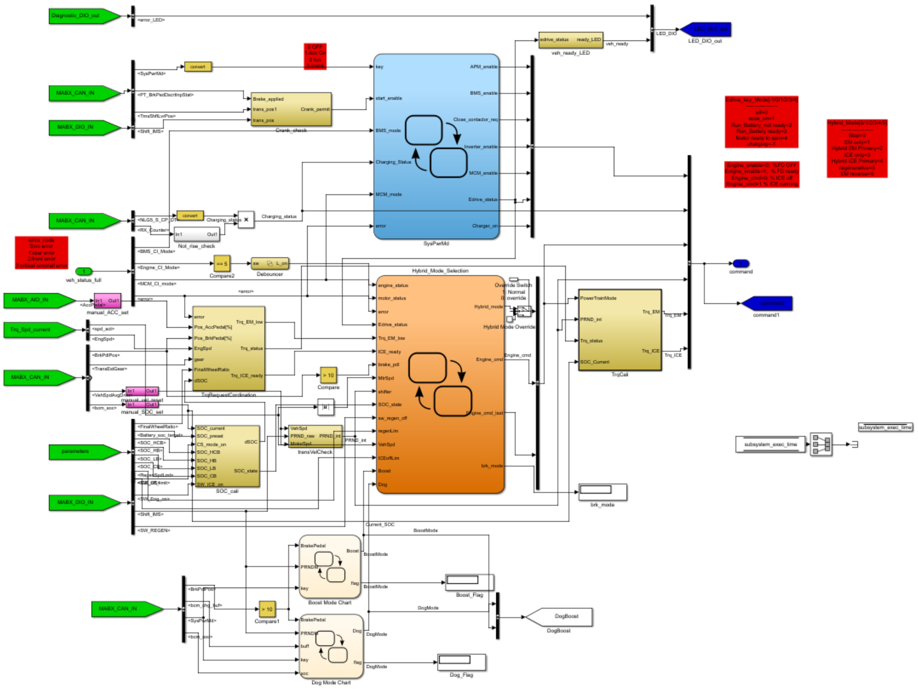

Figure 2-3 Overview of 'supervisory' block in implementation

Comparing the whole system of simulation model and the 'supervisory' block of implementation model, it is obvious that the state of vehicle is simplified to several data set in the simulation while those parameters would need to be obtained from the vehicle directly in implementation model. The core is still the 'TrqCali' block where the control strategy was placed.

When the block 'TrqCali' is expanded, it turns out that this block is almost the same in implementation version and simulation version except for the way some signals are input into this block, as shown in figure 2-4 and figure 2-5.

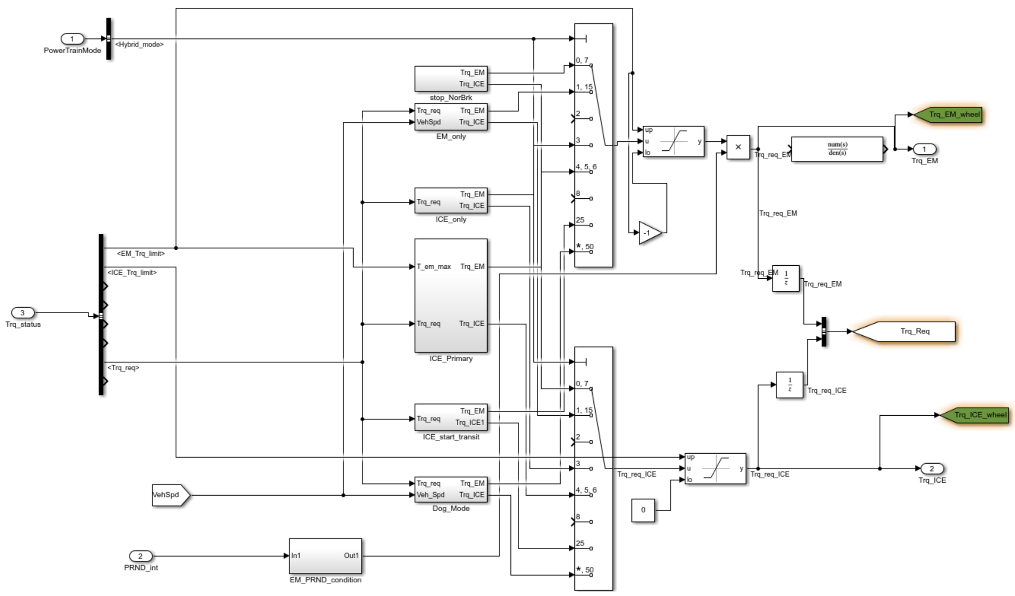

Figure 2-4 'TrqCali' block of the simulation model

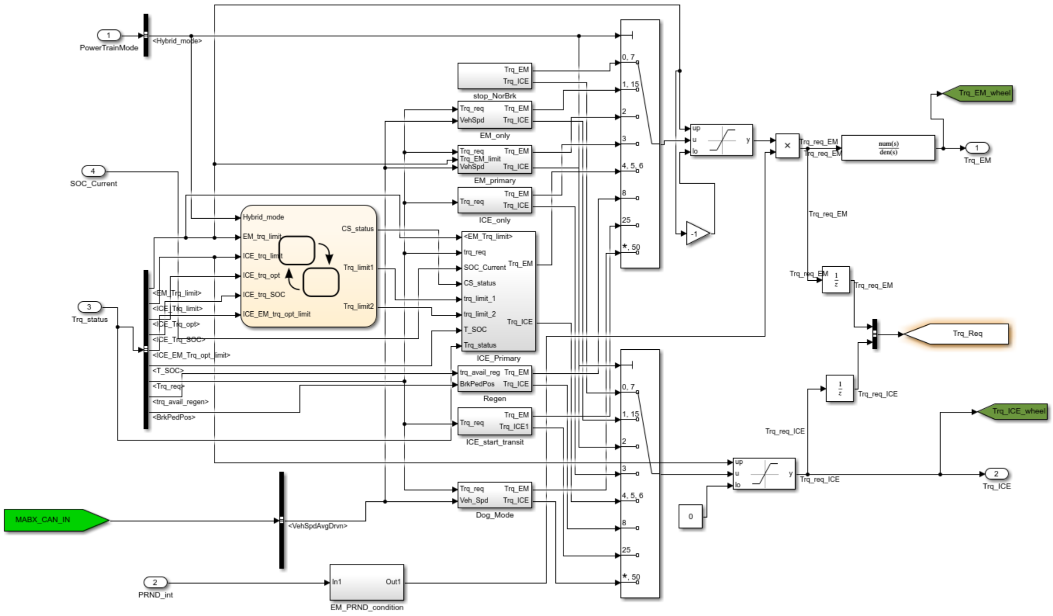

Figure 2-5 ‘TrqCali’ block of the implementation model

To make sure that the codes inside are exactly the same, a comparison between the ‘ICE_Primary’ blocks in implementation and simulation is made. The control strategy, such as PSOC and ECMS, is placed here as a controller which will be activated when the vehicle wants to use both the engine and the electric motor. The results show that all three strategies are placed in one Simulink model in implementation while the three strategies are separated into three independent Simulink models in simulation, as shown in figure 2-6. According to figure 2-7 and figure 2-8, no other great differences can be found in the codes for each control strategy, including the functions used inside the blocks.

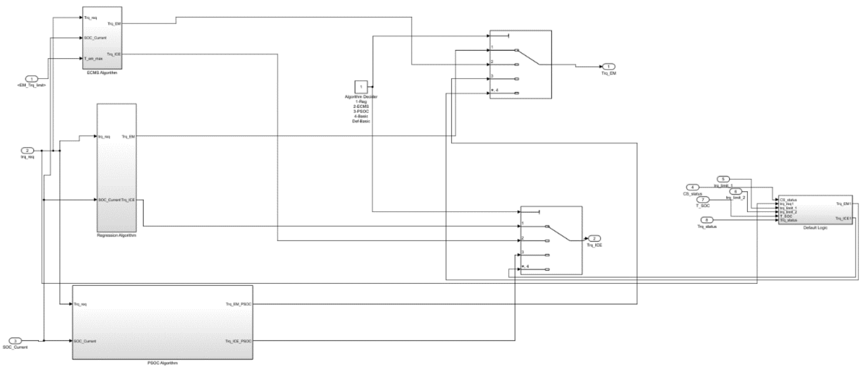

Figure 2-6 ‘ICE_Primary’ block

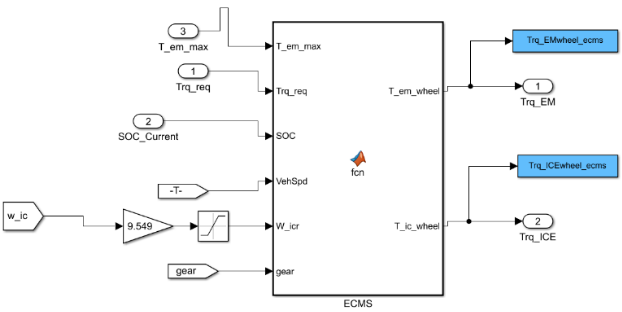

Figure 2-7 ECMS codes in implementation

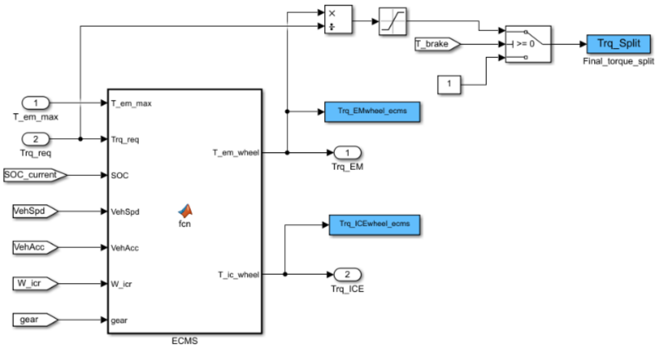

Figure 2-8 ECMS codes in simulation

That is to say, the Simulink model for implementation shows no great difference with the Simulink model for simulation. Since the implementation of PSOC is successful, there is no reason to doubt the implementation method or the Simulink model for implementation.

2.2 Relationship between the results

To conduct a comparison between the results of simulation and implementation, the data collected from a successful implementation is necessary. According to Sauradeep [2], only the implementation of PSOC succeeded without any motor oscillations. Therefore, the results of implementation and simulation of PSOC are plotted out to find out any possible relationship.

The first factor that needs to be checked is the drive cycle. In simulation, the UDDS standard drive cycle is used. According to Sauradeep in his report, the drive cycle is expected to be as close to the UDDS standard drive cycle as possible [2], as shown in figure 2-9.

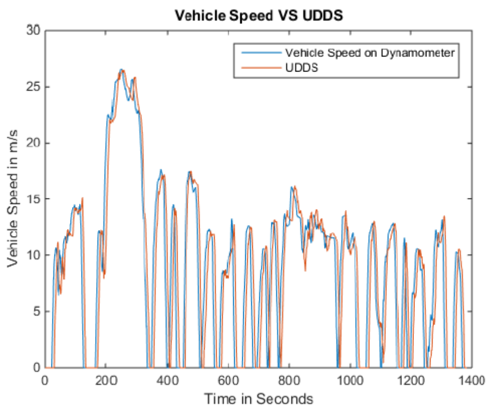

Figure 2-9 Vehicle speed in PSOC implementation comparison with UDDS

The second factor that needs to be checked is the total torque. If the simulation model matches with the real car state well, a similar drive cycle will certainly lead to a similar total torque profile. It needs to be mentioned that the torque profile consists of two parts, calculation result of the control strategy and regenerative braking. The calculation result of control strategy from the 'ICE_Primary' block will always keep positive since the car needs to be pushed forward while the regenerative braking will always show negative because it is actually storing energy to the battery.

Figure 2-11, a plot from the collected data shows that the total torque profile is almost the same, which means a total torque profile similar to simulation result should be expected in a successful implementation where the drive cycle is close to a standard UDDS drive cycle, as shown in figure 2-10.

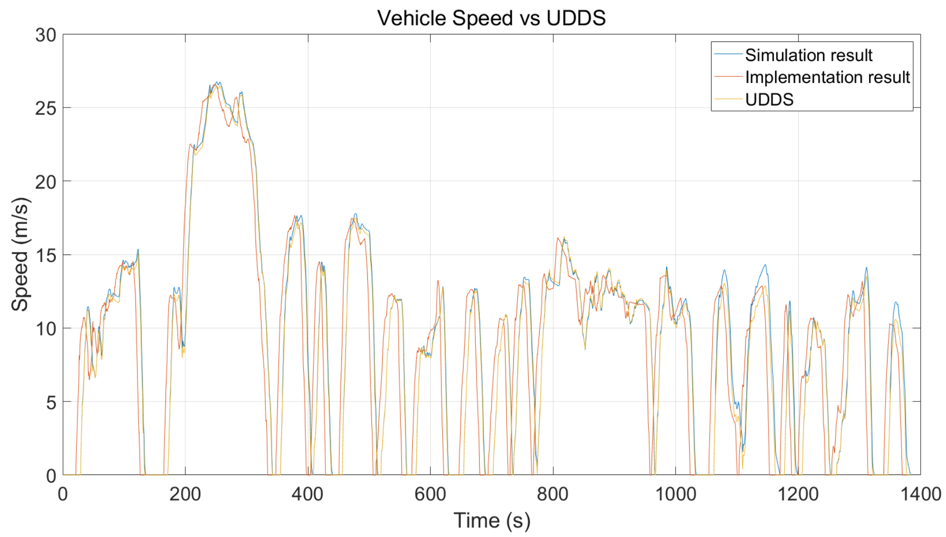

Figure 2-10 Comparison of vehicle speed to standard UDDS drive cycle.

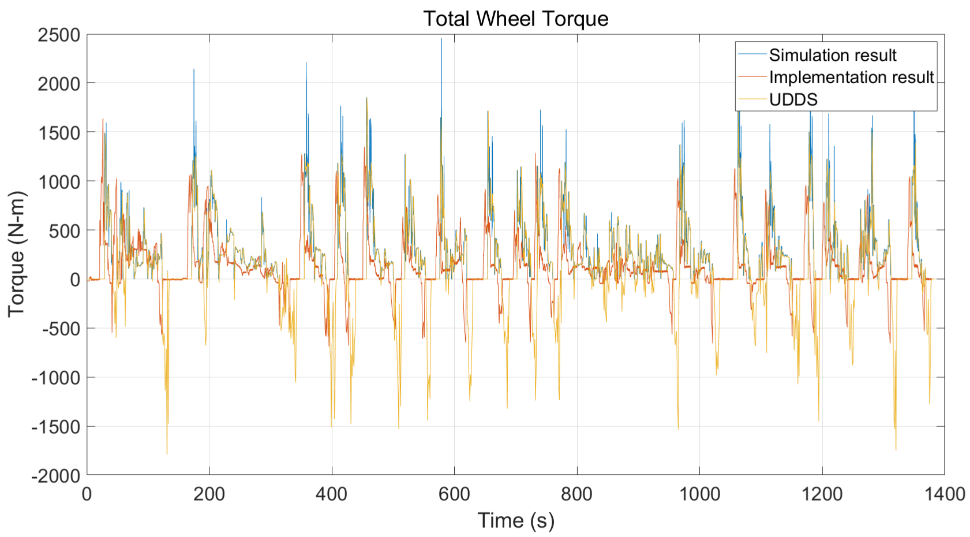

Figure 2-11 Comparison of total wheel torque to standard UDDS drive cycle

2.3 Conclusion on the relationship

It is easy to see that there is no great difference between the codes of implementation and the codes of simulation. Also, if the drive cycle is input to the test correctly, a result similar to the simulation result is expected in implementation. Therefore, the codes should not be a possible reason for the oscillations, which have never occurred in simulation.

In the meanwhile, knowing what the expectation for a successful implementation should be is also probably useful when looking into the data collected in experiments since the simulation results can provide a benchmark for the implementation results.

3 REASON FOR THE OSCILLATIONS

After making sure the implementation codes work well, more detailed study of the experiment data was conducted to find out possible reasons for the oscillation. One basic concept is to find out all the signals showing oscillations to track the origin of the oscillations. Since the regression and ECMS showed similar problems, the assumption could be made that they are caused by the same reason. Then we can focus on the ECMS data and check all the signals in the data set.

As mentioned above, the motor showed great oscillations when operating ECMS in implementation. The oscillations in motor speed are actually caused by the oscillations in the torque, as shown in figure 3-1. In this figure, EM and ICE tells where the torque comes from and ICE Primary means the signal is the calculation result of the control strategy, instead of the actual torque.

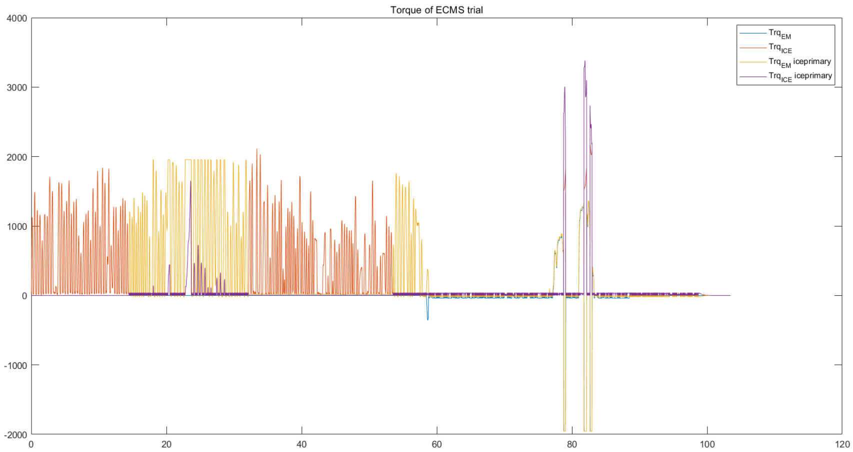

Figure 3-1 Oscillations in torque profile of ECMS implementation

As shown in figure 3-2, similar oscillations are observed in the input of a saturation block in the 'TrqCali' block (see figure 2-5) . The location of the block is shown in figure 3-3. Due to the powertrain mode filter, some oscillations disappeared before the signal finally goes to the output. This means the oscillation occurred after the calculation of control strategy.

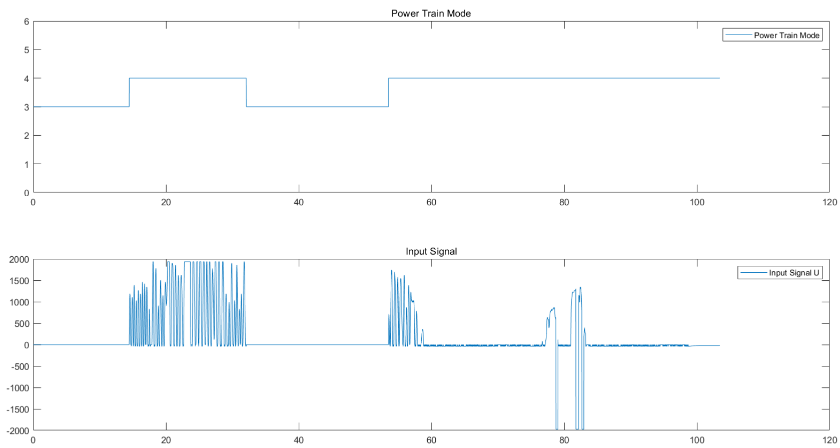

Figure 3-2 Oscillations filtered by the power train mode controller

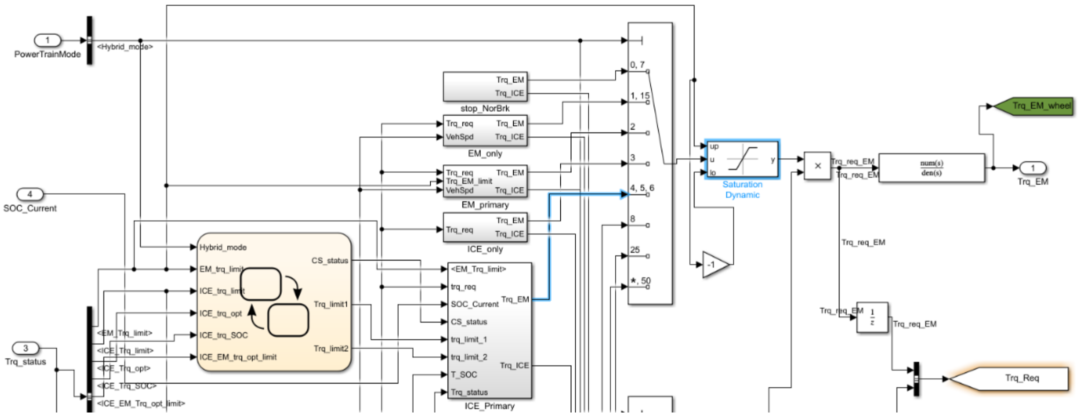

Figure 3-3 Input of the saturation block and the data flow

Since the conclusion has been reached that the codes would not produce such oscillations, the oscillations must come from the input to this ‘ICE_Primary’ block. After checking all the data in the data set, another signal with similar oscillations was found among the inputs of the ‘ICE_Primary’ block, as shown in figure 3-4. It is labelled with ‘Pos_AccPedal’ and its meaning is the pedal position. This signal is located in the ‘Torque Request Coordination’ block, as shown in figure 3-5.

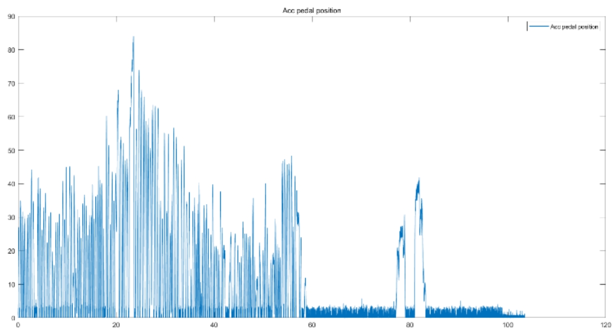

Figure 3-4 Similar oscillations in the signal of pedal position

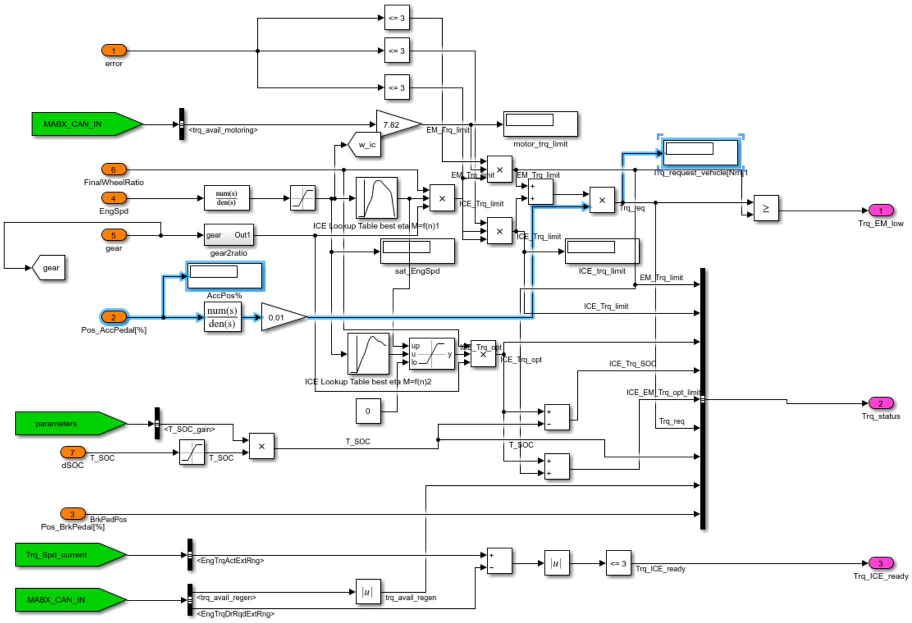

Figure 3-5 ‘Torque Request Coordination’ block and signal location

It is obvious that this signal gives the required torque after calculation. Plotting the required torque shows similar oscillation while this required torque is a direct input to the controller, as shown in figure 3-6.

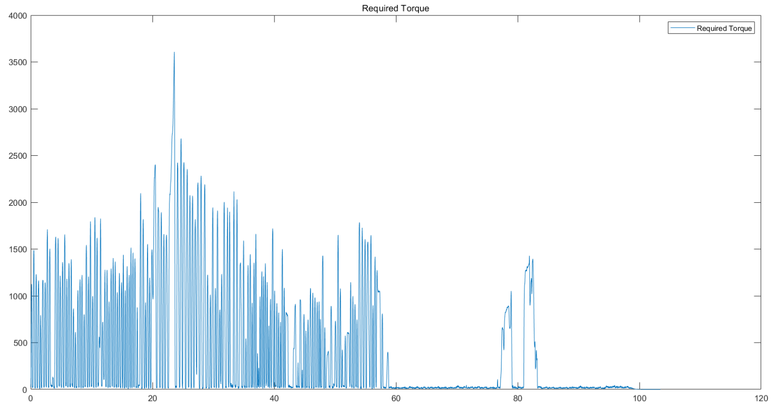

Figure 3-6 Similar oscillations in the signal of required torque

It seems that the oscillations in motor speed are caused by the oscillations in the signal of pedal position. Then the questions is where this signal comes from. By looking into overview of the Simulink model (see figure 2-3), it turns out that this signal is completely an independent input to this system, as shown in figure 3-7.

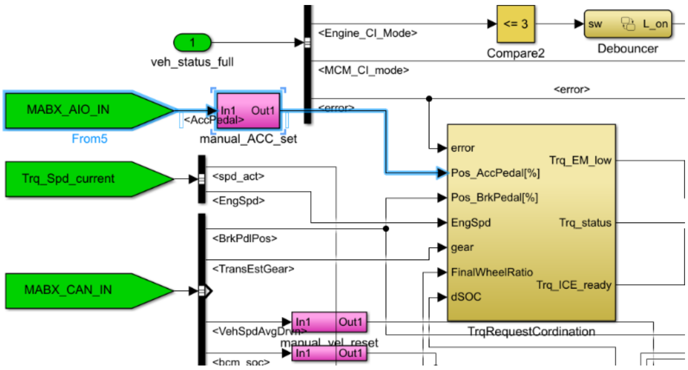

Figure 3-7 Pos_AccPedal comes from an independent input

That is to say, the oscillations in motor speed are totally caused by the oscillations in this analog input. To make sure this conclusion is right, check the successful PSOC experiments and see how these signals should be in a successful implementation, as shown in figure 3-8, 3-9 and 3-10. To get a better view of how the normal signal is, a small-time region 0-120s is focused, as shown in figure 3-11, 3-12 and 3-13.

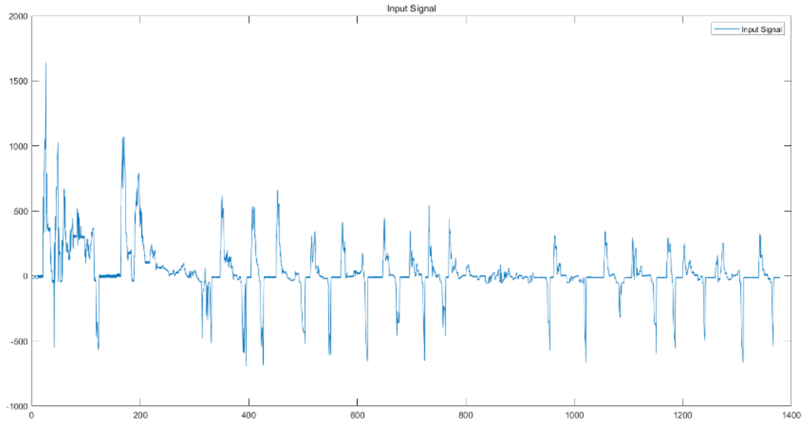

Figure 3-8 Input signal in PSOC experiment, 0-1400s

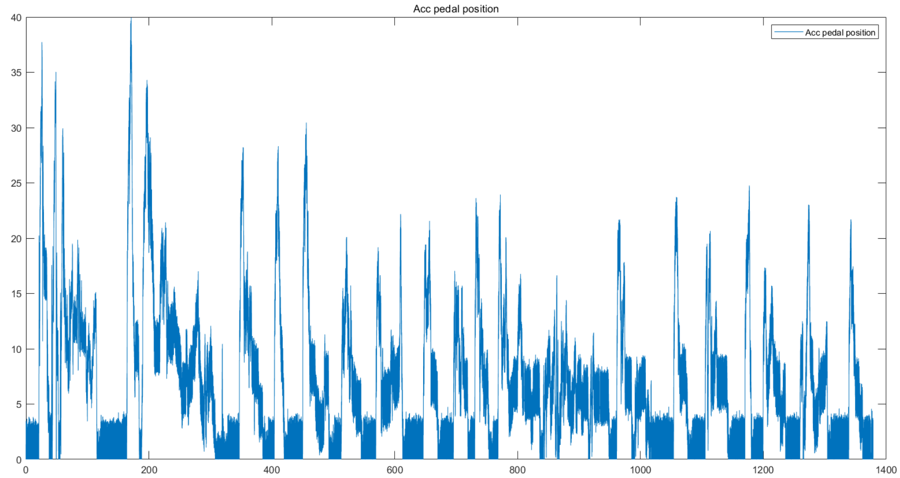

Figure 3-9 Signal of pedal position in PSOC experiment, 0-1400s

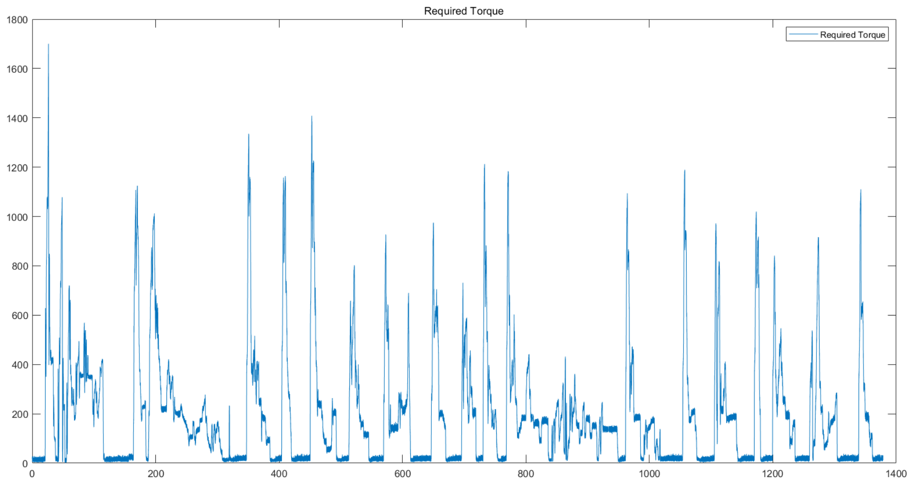

Figure 3-10 Signal of required torque in PSOC experiment, 0-1400s

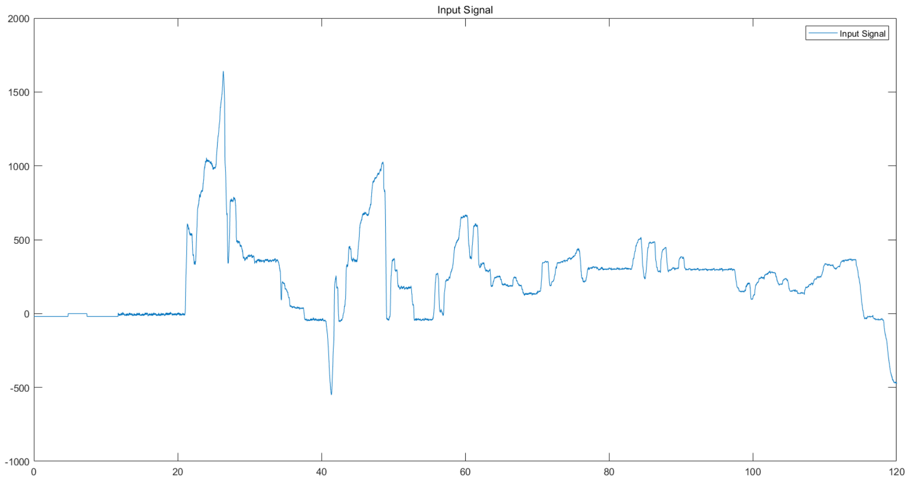

Figure 3-11 Input signal in PSOC experiment, 0-200s

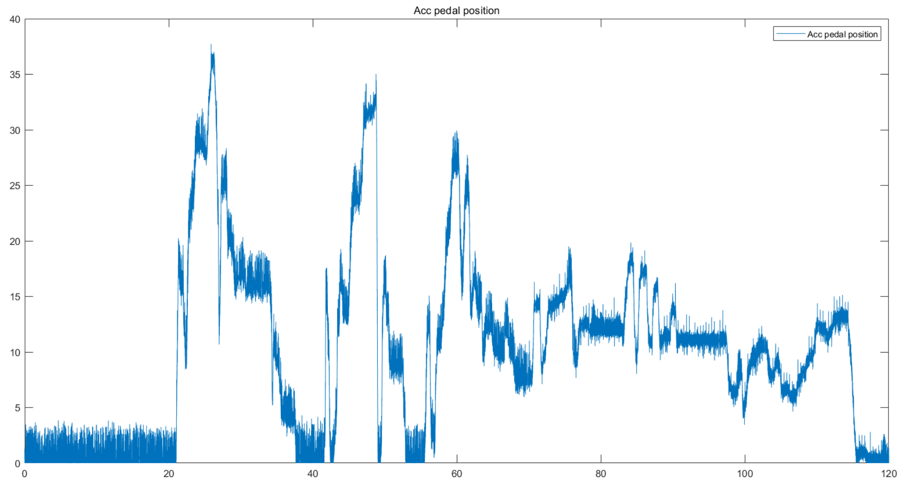

Figure 3-12 Signal of pedal position in PSOC experiment, 0-200s

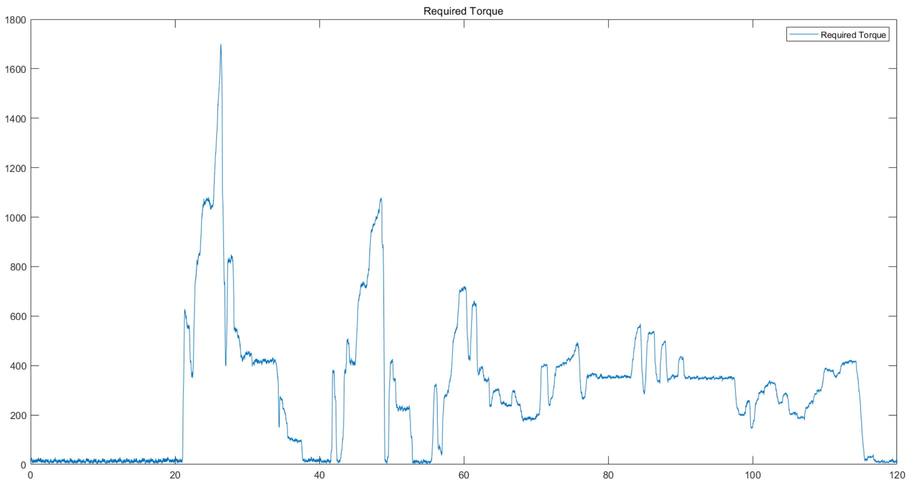

Figure 3-13 Signal of required torque in PSOC experiment, 0-200s

It is obvious that it was the unusual oscillations in the analog input that caused the motor fluctuation. In the successful PSOC experiment, no oscillations occurred in the analog input and everything worked well. However, strange oscillations occurred in the analog input of accelerator pedal position, which finally caused the oscillation in motor speed and made the implementation fail.

4 CONCLUSIONS AND FUTURE WORK

4.1 Conclusion

Since the codes for implementation and simulation are the same, there is nothing wrong with the implementation codes. By plotting the signals one by one, it turns out that the most possible reason for the failure of ECMS implementation is the oscillation in the analog signal of the accelerator pedal position. This oscillation caused the oscillation in the required torque of vehicle, which is a direct input into the control strategy, thus finally producing oscillations in the motor speed.

4.2 Future work

On the one hand, since the oscillation is mainly caused by the input in implementation, further investigation is needed to check how this signal is produced and transmitted and why it showed such a huge oscillation in the implementation experiments. This could be a small problem such as something went wrong with the sensor while it could also be a very complicated problem related to the pedal structure. We cannot reach a conclusion on what the problem exactly is until someone gets to the lab and tests the particular part of pedal position detection.

On the other hand, the conclusion reached from the ECMS data set is not enough to say that the problem in Regression implementation must be the same. However, because the data of Regression implementation where oscillations occurred is missing, no test or check can be conducted to see the signals of Regression implementation while it’s essential for further experiments. Someone must find the missing data set and check if the failures of ECMS and Regression are caused by the same reason.

5 REFERENCES

[1] Rohinish Gupta, "Modeling and control of a parallel through-the-road plug-in hybrid vehicle", Purdue University, West Lafayette, Indiana, 2016.

[2] Sauradeep Samanta, "Real-time implementation of power management strategies in a hybrid electric vehicle", Purdue University, West Lafayette, Indiana, 2019.