POLITECNICO
DI TORINO

A Master's Thesis on

Vehicle Longitudinal Dynamics With Electric Powertrain Experimentally Measured Efficiency & Losses

USMAN SALIM

A Thesis submitted for the fulfillment for the

Master of Science Programme In

Automotive Engineering at Politecnico di Torino , Italy

Author

Usman Salim

Matricola: S231400

Ms. Automotive Engineering

DIMEAS (Department of Mechanical and Aerospace Engineering)

Supervisor

Stefano Carabelli

DAUIN (Department of Control and Computer Engineering)

Abstract

The vehicle dynamics model are important tools for research and development in automotive Industry . In today's fast paced industry, the use of software-based simulation plays a vital role in testing and verification of products. The vehicle longitudinal dynamics considers the behavior of different forces acting on the vehicle during its longitudinal motion. The resistance forces in turn may affect the performance of the vehicle powertrain , pushing the propulsion system to provide more better performance to overcome the resistance. This leads to lowering of efficiency or increase in power loss. To observe the vehicle behavior in real life , it may certainly require costly experimentations or simulations , on the other hand software-based approach can lead to a less time-consuming situation.

The Aim of this thesis is to provide a model that is like a vehicle with an electric powertrain and analyzing the efficiency and losses by experimental techniques followed by using software-based approach.Hence, a software package or toolbox that can perform different tasks related to vehicle dynamics and performance shall be presented. The Thesis was carried out using Octave software that features a high-level programming language, primarily intended for numerical computations.

The Results obtained from the simulation of the software are in line with real world behavior of a vehicle with electric powertrain. However , it is certain that such models are used for providing a bottom line and a perspective in real driving conditions.

Keywords: Electric Powertrain , Vehicle longitudinal dynamics , Electric machine power losses, Efficiency of Electric motors, vehicle over driving cycle.

Table of Contents

List of Figures

List of Tables

Document History:

Version	Description	Date
1.0	(Software development) Vehicle longitudinal dynamic simulation using PID Driver	01-Jun-18
2.0	Thesis writing and Power losses efficiency models development	06-Oct-18
3.0	Final review (Structure of software, Installation Manual and Code architecture)	11-Feb-19

Table 1- Document History

1 Introduction

1.1 Background

The vehicle dynamic models are used in today's automotive industry in the research and development of vehicle development. In simple words a vehicle dynamic model is a model of a vehicle during motion . Such models provide a better understanding on the parameters that may influence the vehicle behavior in real life. Moreover , it may be obvious to state that such models are virtual prototypes.

The main purpose of vehicle modeling using software-based techniques is that developers can test , evaluate and optimize the control algorithms used in vehicle development phase and it provides as a degree of freedom to the developers with decreasing the time required for verification and testing.

The are many applications where vehicle models may be needed such as vehicle range-evaluation, homologation, autonomous driving control algorithms etc. but the degree of freedom is dependent on the type of application.

This thesis provides an insight into a how a software can perform performance evaluation of the longitudinal dynamics of a vehicle (equipped with electric powertrain) and evaluation of the E-powertrain (Efficiency , Losses and Operating area) using a high level programming(Gnu Octave) .

The program was established to study and simulate the behavior of a vehicle model considering only longitudinal dynamics of the vehicle whilst following a reference speed. The vehicle is driven by a PID (Proportional-Integral-Derivative) driver that nearly follows the reference speed which is provided as a driving cycle.

The vehicle operating points are hence evaluated using the software simulation. The losses and efficiency of the electric powertrain can be evaluated under different operating conditions provided by the user. The software based approach gives an approximate result of real-world driving situation using test cycles.

1.2 Problem Definition

The electric motor map is provided by a manufacturer and vehicle parameters are provided by the vehicle design consideration. The necessity of a vehicle model (Algorithm) that can present the electric powertrain in the vehicle and evaluate it in real world behavior to an approximate level arises. This problem leads to a development of a virtual prototype software-based model that creates the purpose of this thesis.

1.3 Research Questions

The following are some brief research questions that shall be answered using simulation of the software.

- Does the provided electric machine satisfy the traction requirement posed by the vehicle motion during the test cycle?
- What is the operating area of vehicle , while considering longitudinal motion?
- What are the power losses due to the electric machine and how it may affect the tractive power?
- How to provide the approximate model (Algorithms) for EV Powertrain?

1.4 Deliverables

The software which is developed can be modified to present results based on models (algorithms)

Electric Vehicle Model 1 (Algorithm)

- The first type of model is based on the percentage of pedal movement by the driver that consider the travel of the pedal in forward and backward direction. **The control parameter is the amount of pedal percentage**

Electric Vehicle Model 2 (Algorithm)

- The second type of model takes the force input into the electric machine.**The control parameter is the tractive force required to give the specific speed.**

Map of Efficiencies and Losses (Data Set)

- The Output of the simulation is given in the form of a map of the operating area of the vehicle having the map of efficiencies and losses.

Important points:

- The vehicle models can be used to study the vehicle longitudinal dynamics.
- The map of efficiencies and losses are obtained using a electric vehicle model and studying the behavior of the E-powertrain using experimental equations.

1.5 Tools

There can be several tools available for the development of a vehicle longitudinal model such as MATLAB/Simulink, MSC ADAMS (Real-time)or Modelica. However, the software used to carry out this simulation is Gnu Octave. The primary reason is that is high level programming language and an **opensource software** available under the GNU General Public License. It has compatibility with MATLAB if the code is carefully written.

Each methodology may have pros and cons. Initially this programming methodology is time consuming. However, GNU Octave has a text-based editor where our program will be written to evaluate the vehicle longitudinal dynamics using a structured approach where the main code has several functions to overcome the need of a vehicle dynamics library . This also may give a better readability and user-friendly interface. To perform the simulation, the following software are used:

No.	Software	Version	Type	Purpose
1.	Gnu Octave	4.4.0	Open Source	Coding
2.	Microsoft Excel	2016	Student	Spreadsheet

Table 2-Softwares Utilized

1.6 Outline

The Following figure gives the Outline of the thesis

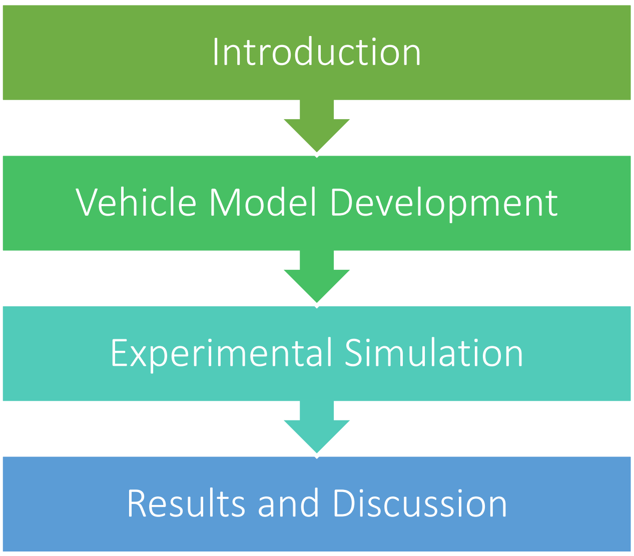


Figure 1-Thesis Outline

2 Vehicle Model Development

In this chapter, the focus is to understand the main concepts behind our developed code and how the process is performed. Every model has some physics which governs the process and outputs. Therefore, it is necessary to initially give a sketch of the simulation , which will be presented in the form of models.

2.1 Vehicle Modelling

It is necessary to present a model to assess the vehicle performance over the test cycle. The model types can be classified into the following types based on how the system may react to the test cycle reference speed.

2.1.1 Kinematic Model

The kinematic model features a strictly following reference speed and an open-loop control approach. This model provides a bottom line for assessment of achievable performance under nominal situation. Such a model is not of our interest as it doesn't feature any driver's behavior.

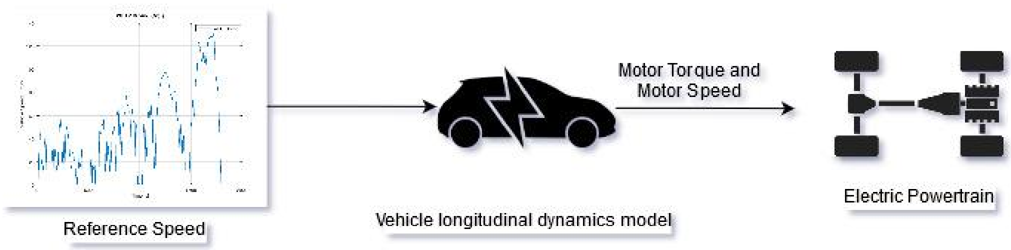


Figure 2-Kinematic Model

2.1.2 Dynamic Model

The dynamic model follows a closed-loop control approach. This makes the driver charge of the vehicle speed and acting on the wheels. The speed difference is calculated and a PID controller simulates the driver behavior to model the proportional, predictive and memory contributions.

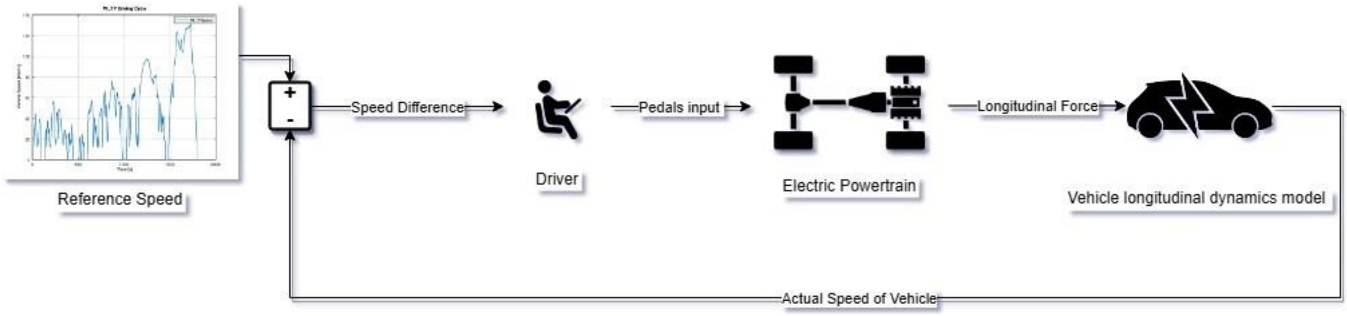


Figure 3-Dynamic Model

2.2 Model Parameters

To perform the simulation some data is required initially. Moreover, this data is to be verified with some test conditions. The following are some initial data used in our model.

2.2.1 Vehicle Related Parameters

Name	Symbol	Value	Codename	Unit
Mass of the vehicle	m_{veh}	1000	mass	$[kg]$
Aerodynamic Drag Coefficient	C_x	0.41	C_x	[-]
Coefficient of Resistance	K	$6.5e-6$	f_2	[-]
Rolling Resistance Coefficient	C_{rr}	0.0148	f_0	[-]
Vehicle Frontal Area	A_f	1.7	Af	$[m^2]$
Air Density	ρ	1.184	Rho	[-]
Gravitational Acceleration	g	9.81	g	$\left[\frac{m}{s^2}\right]$
Radius of Wheel	R_{wheel}	0.2806	r_wheel	$[m]$
Number of wheels	N_{wheels}	4	n_w	[-]
Angular Inertia of wheels	J_w	0.707	J_w	$[kg * m^2]$
Number of E-motor	N_{motors}	1	n_m	[-]
Angular Inertia of E-motor	J_m	0.0033	J_m	$[kg * m^2]$
Total Transmission Ratio	$Tot_{Trans_{ratio}}$	30.8	tot_trans_ratio	$[m^{-1}]$
Single gear transmission Ratio	$i_{gear_{tot}}$	8.67	i_gear_tot	[-]
Angular Inertia of transmission	J_{tau}	0.015	J_t	$[kg * m^2]$
Rotational mass	m_{rot}	39.258	M_rot	$[kg]$
Equivalent mass of vehicle	m_{equiv}	1039.3	M_tot	$[kg]$
Transmission efficiency	η_{trans}	0.98	Eta_trans	[-]

Table 3-Vehicle Related Parameters

2.2.2 Coast down coefficients

When the vehicle is in motion there are certain resistance force on the vehicle that effect the speed. These effects can be studied by coast down test. The coast down coefficients are evaluated by using a test in which a vehicle is left to decelerate by disconnecting the propulsion source to calculate the resistance forces on the vehicle.

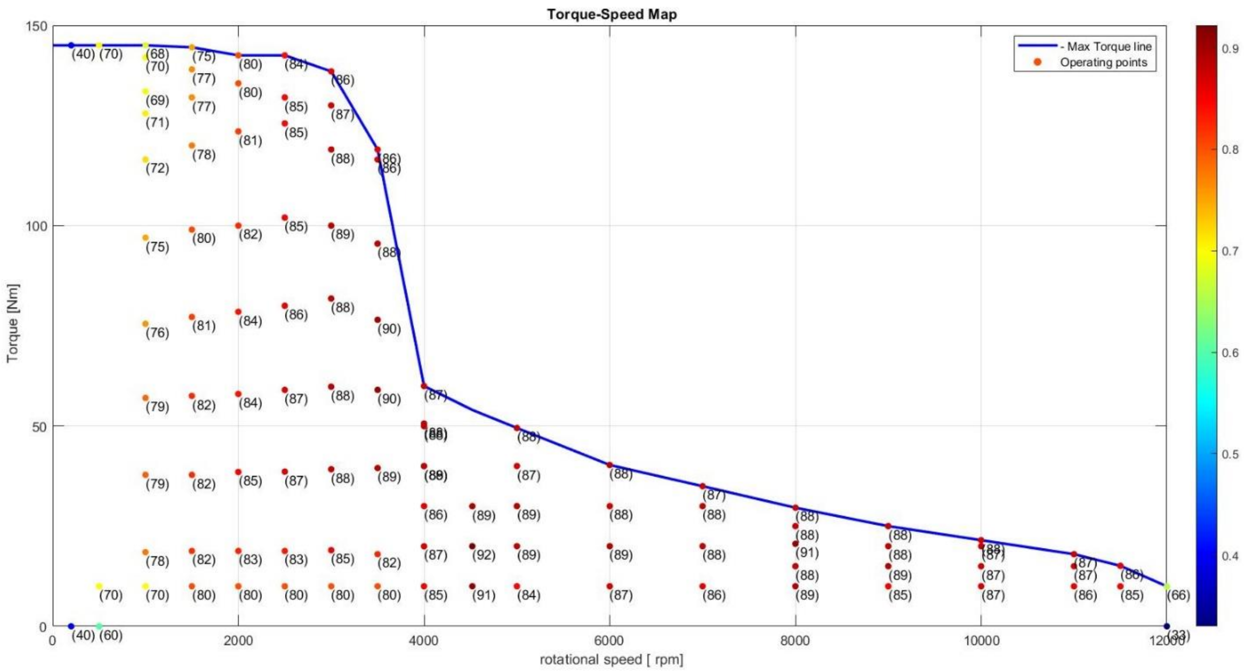
Coefficient	Value	Unit
C_0	145.0911	[N]
C_1	0	[kg/s]
C_2	0.4764	[kg/m]

Table 4-Coast Down Coefficients

2.2.3 Torque-Speed Map

The Torque-Speed Map is provided in the form of a spreadsheet where the data concerns for torque at the given speed and efficiency of the E-Motor at that specific speed. This Map is read using command in Octave and then assigned to variables. The Following Figure provides the data distribution of the map an efficiency information at the operating points.

Figure 4- E-Machine Torque-Speed Map

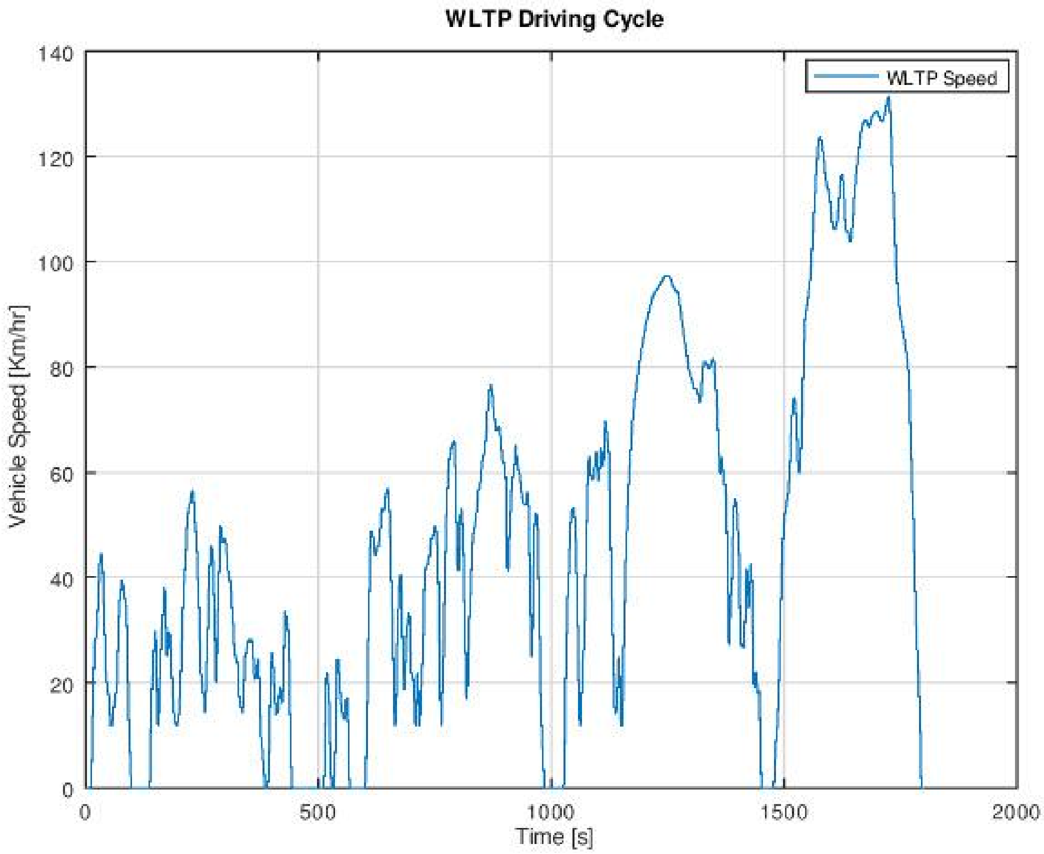


2.2.4 Test Cycle

Different driving cycles are presented by various regions throughout the world , the most popular in Europe is the NEDC (New European driving cycle) which is widely criticized due to its incapability to address the real-world driving behavior. The WLTP (Worldwide Harmonized Light Vehicle Test Procedure) gives a more dynamic behavior of the vehicle speed variation.

The driving cycle provides the information regarding the vehicle speed profile with time . The driving cycle defines the vehicle speed variation and the behavior of the driver. In this simulation the driving cycle is considered as a reference or target speed that the driver must follow.

To provide the input to the program , the driving cycle is provided through the spreadsheet. Which is read by the program by using specific functions embedded in the GNU Octave software . These values are hence stored in velocity and time variables.



Reference Speed

Figure 5- Test Cycle

2.3 Flowcharts

The Following flowchart gives a general overview of the program. In order to get a deeper understanding one can refer to **Appendix**.However, the vehicle model algorithms vary only in the implementation of PID control parameter.

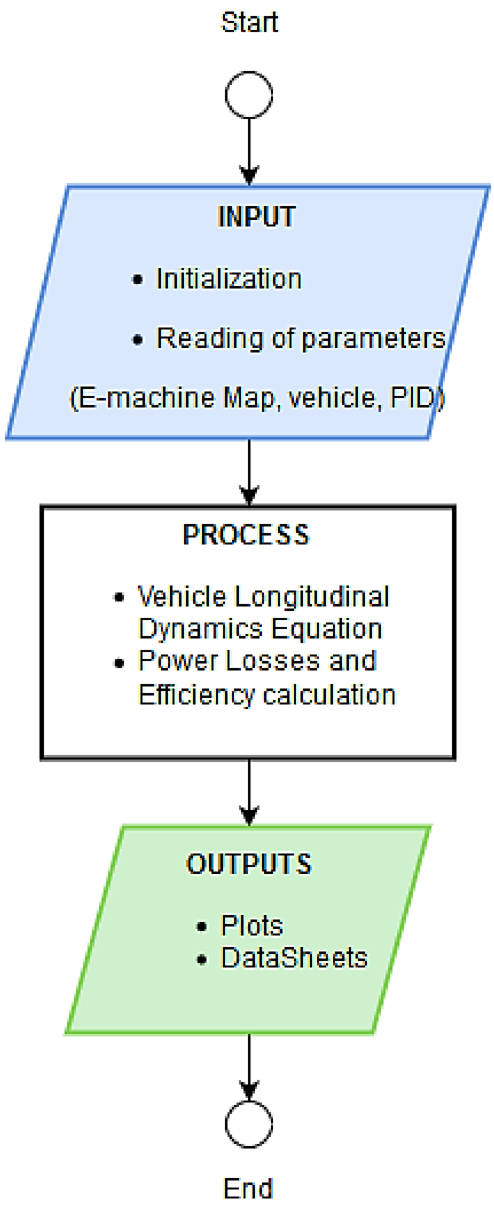


Figure 6-Program Flowchart

2.3.1 EV Model 1

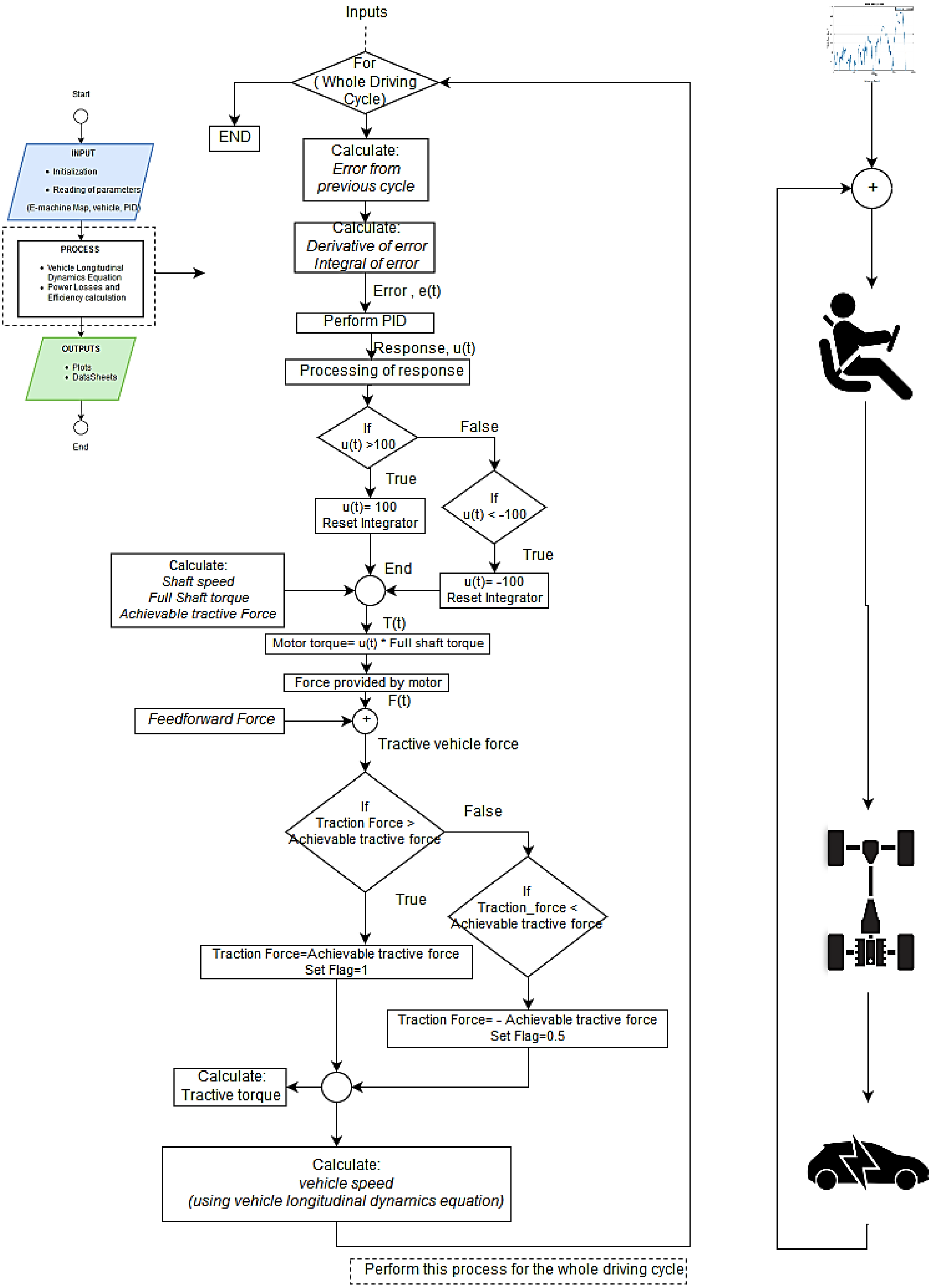


Figure 7-EV Model 1 Flowchart

2.3.2 EV Model 2

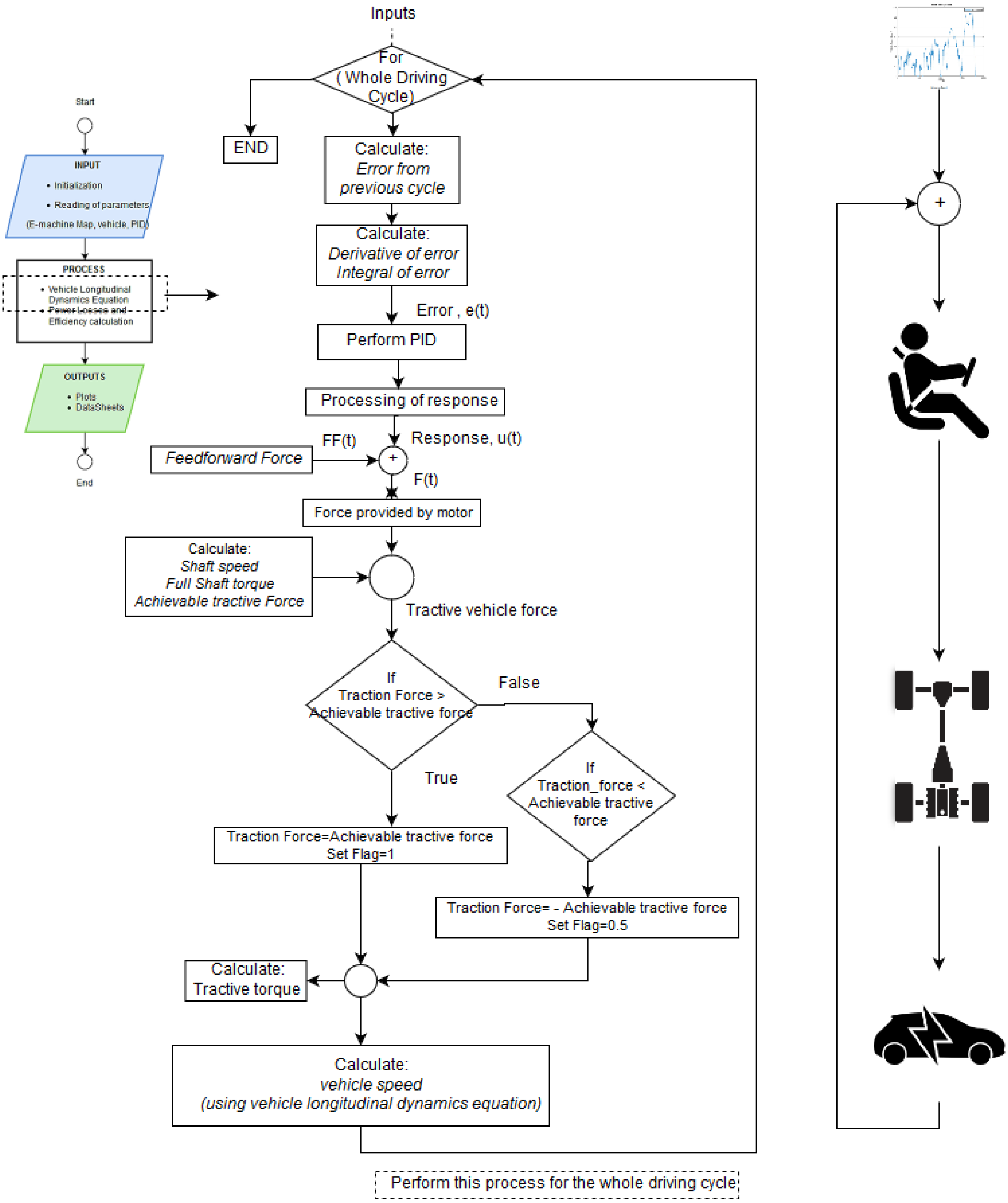


Figure 8-EV Model 2 Flowchart

2.3.3 Power Loss and Efficiency Flowchart

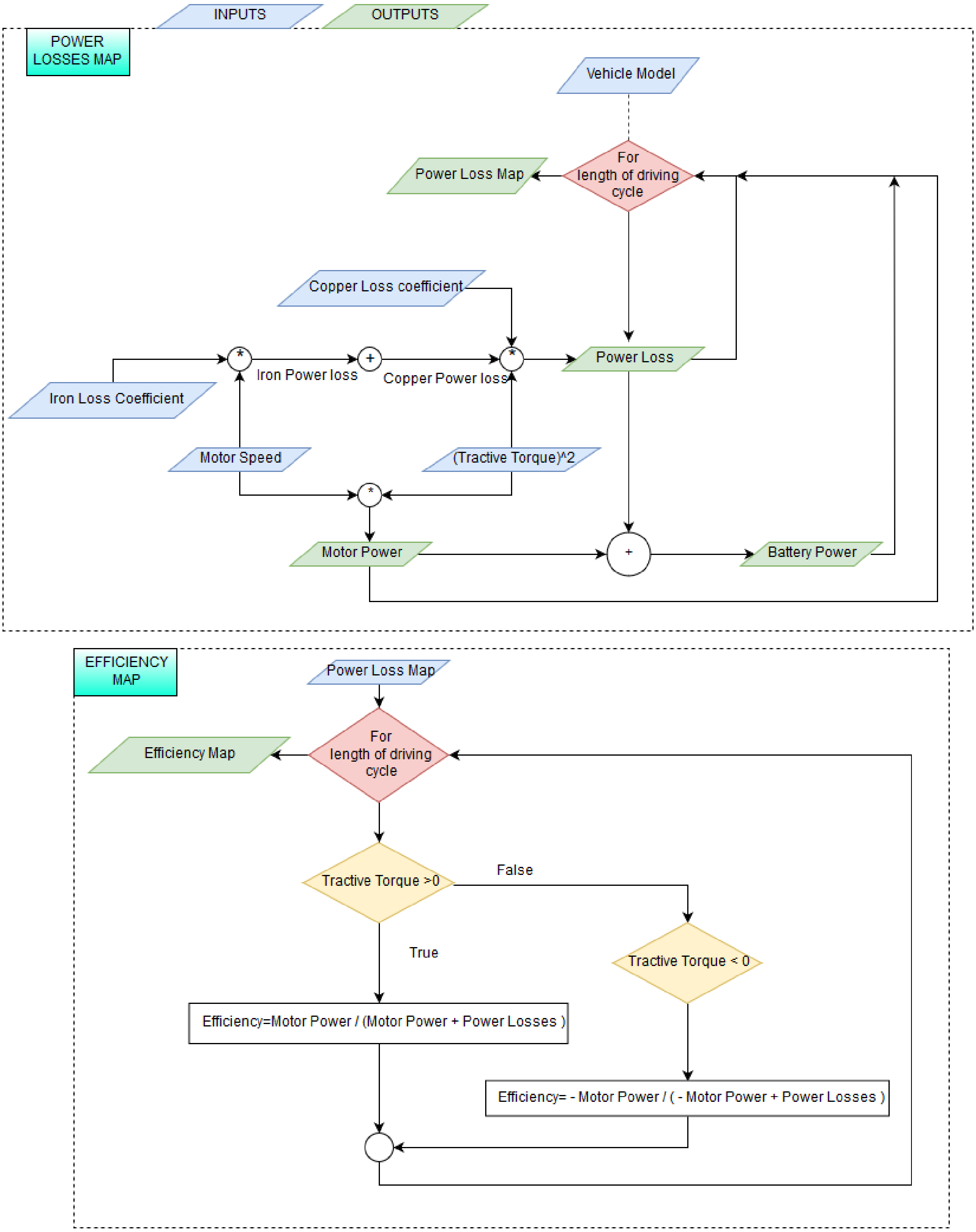


Figure 9- Power losses and Efficiency flowchart

2.4 Concepts and Equations

2.4.1 PID Driver approximation:

The PID loop is composed of Proportional , Integral and Derivative control strategies. To have a good approximation of the driver it is necessary to select the proper coefficients for Kp , Ki and Kd (That are the magnitude of proportional, integral and derivative control).These coefficients are either provided by the manufacturer for a controller or can be found using Trial and Error methods or can be evaluated by using tuning methodologies.

The following figure gives an idea of what is inside the Driver Model represented previously.

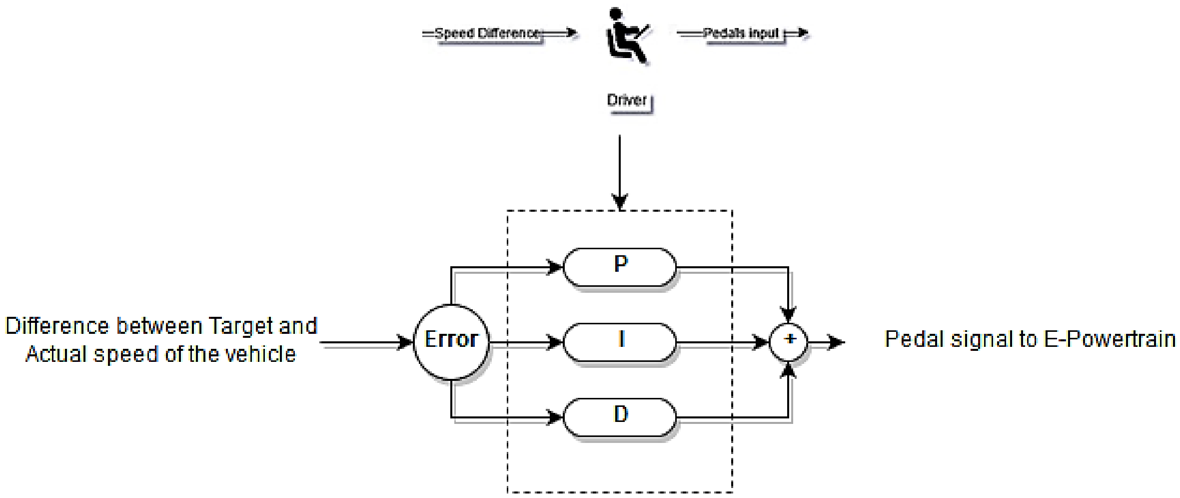


Figure 10- PID Driver

The target or reference speed for the vehicle to follow is the driving cycle. The error between the reference and actual speed of the vehicle is given by the following formula.

$$error = V_{ref} - V_{act} \quad \left[\frac{m}{s}\right]$$

Whereas the Pedal (Response) signal is provided to the electric powertrain , may be of acceleration or deceleration. Which is controlled by the PID control using the following formulation.

$$pedal = P + I + D$$

In this equation:

The Term **P** is proportional to the current value of error.

The Term **I** consider the past values of error and integrates it over time.

The Term **D** provides an estimation of the future trend of the error , based on its current rate of change.

So, the following formula which is most commonly known as the conventional PID can be presented.

$$pedal(t) = K_p * error(t) + K_i \int_0^{t_{cycle}} error(t)\, dt \; + \; K_d \frac{d}{dt} error(t)$$

2.4.2 Transmission Model

The pedal signal which is provided by the PID driver is provided to the E-Powertrain which is composed of the gearbox and the electric machine.

The following figure gives an overview of the scheme

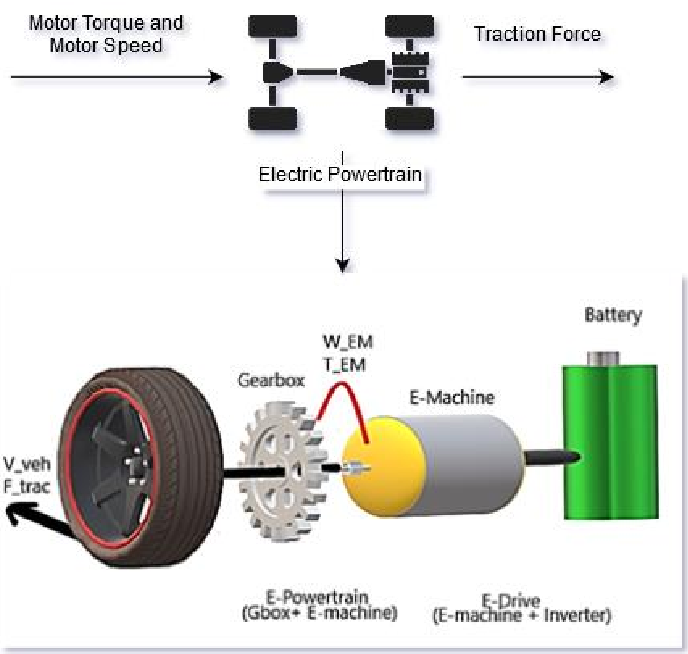


Figure 11- Electric Powertrain

The efficiency map of the e- machine includes the torque and speed (angular speed of the motor shaft) as presented before. As the E-machine is integral with the gearbox and the gearbox provides available tractive force to the wheel to facilitate vehicle motion. In our case we are using a single transmission gear having the final transmission ratio as given in the

Table 3-Vehicle Related ParametersVehicle Related Parameters. It is necessary then to have a transmission model that gives the idea of how the torque is converted into tractive forces and the angular speed of the E-machine into vehicle speed.

The following figure shows the transmission model

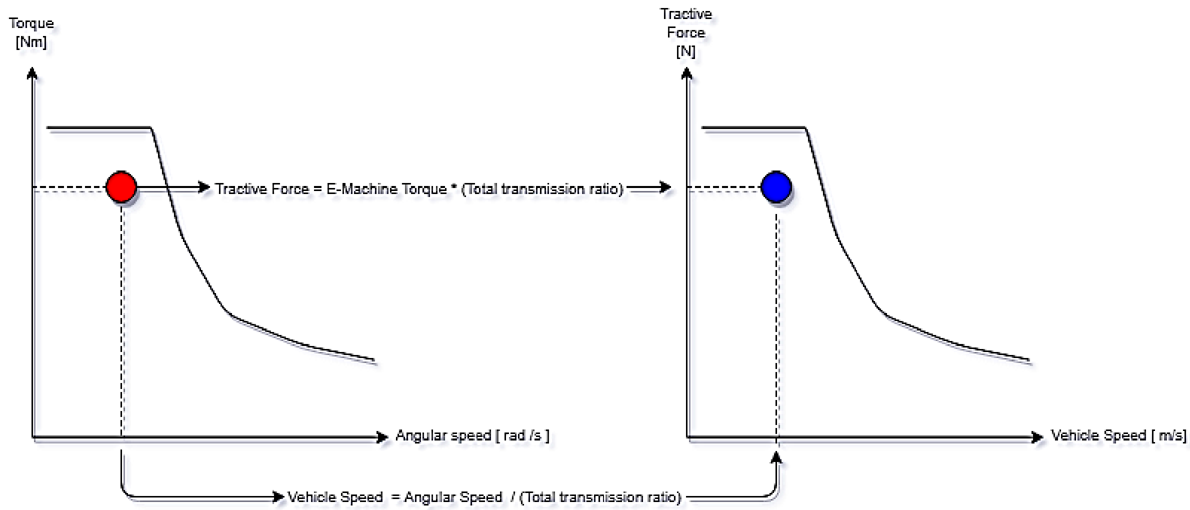


Figure 12-Transmission Model

Using the above-mentioned transmission model , the value of electric machine angular shaft speed is calculated using the following formulation

$$W_{em_{shaft}(t)} = V_{act}(t) * \left(Tot_{trans_{ratio}}\right) * (rad\ to\ rpm) \qquad [RPM]$$

When the angular speed of the shaft is calculated , the maximum torque at the provided speed is then calculated by using linear interpolation.

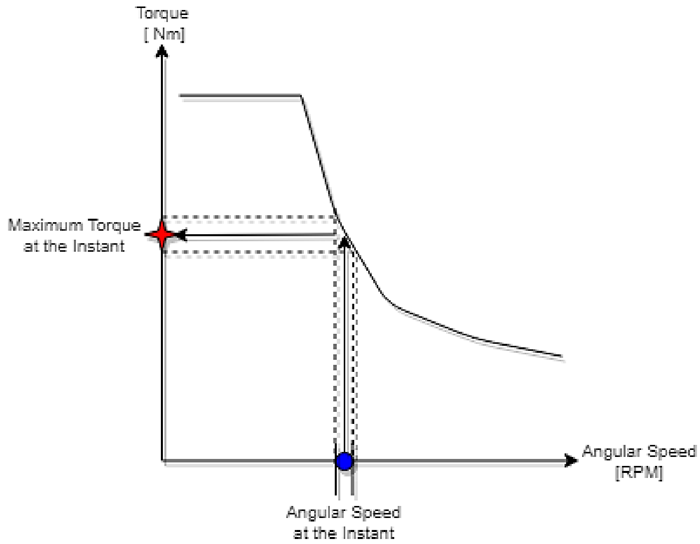


Figure 13-Maximum torque calculation method

The torque given is the maximum torque available at the given speed. In order to calculate the traction force the transmission efficiency is necessary to be considered and given as

$$F_{trac}(t) = T_{em_{shaft}}(t) * \left(Tot_{trans_{ratio}}\right) * (\eta_{trans}) \qquad [N]$$

The tractive force is the amount of force provided by the E-machine to provide the required speed for the vehicle.

2.4.3 Application of Feed Forward control

The application of feed forward control is necessary to reduce the error in the pedal signal. The vehicle when in motion is subjected to resistive forces that is not considered by the feedback control . The following algorithm flow chart gives an overview of its application.

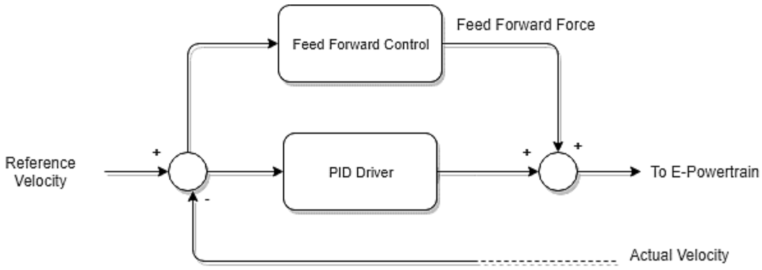


Figure 14-Feed forward control

The main Idea to feedforward is to recover the loss in traction due to resistive force. Moreover, the PID driver tries to push the pedal more to recover the resistance due to rolling and aerodynamic effects and follow the reference speed. The feedforward tries to cancel the known force to lessen the burden on the feedback loop.

2.4.4 Vehicle Longitudinal dynamics

The vehicle longitudinal dynamics means the motion of the vehicle in one direction. Such motion is based on Newtons law of motion given as

$$Force = mass * \ accelaration$$

$$Force_{net} = mass * \frac{dv}{dT}$$

Moreover, the net force is given as follows

$$F_{net}(t) = F_{trac}(t) + F_{res}(t)$$

The Tractive force is provided by the propulsion source (in our case the e-machine) and can be calculated using the transmission model discussed in the previous section. The resistance force is calculated using the coast-down test mentioned before and from the **Coast down coefficients** we can calculate the resistance force.

$$F_{res(t)} = C_0 + C_1 + C_2 * \left(velocity(t)\right)^2 \qquad [N]$$

These coefficients can be found using the aforementioned vehicle related parameters. The Following figure give a more understandable view of main concept behind the resistance force.

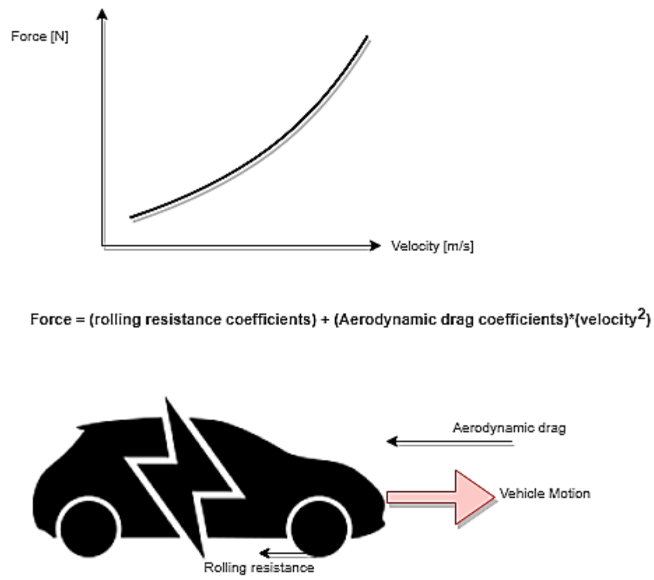


Figure 15-Resistance forces on the vehicle

The final equation which will be used in the simulation to perform the closed-loop control (dynamic model)

$$Velocity_{(t+1)} = \left[\frac{F_{trac}(t) - F_{res}(t)}{mass}\right] * dT + Velocity(t) \quad \left[\frac{m}{s}\right]$$

2.4.5 Torque and Power Limit representation

In order to present the operating regime, it is necessary to limit the torque according to limits

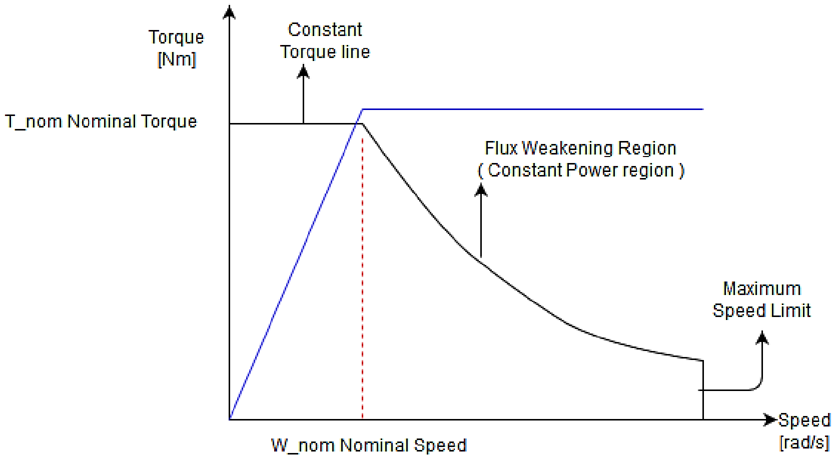


Figure 16- Torque and Power representation

According to the figure, the calculation of torque is done according to the following formulas

The nominal speed and the nominal power at the maximum torque can be hence found as

$$W_{at\_max\_Torque} = abs\left(\frac{P_{max}}{T_{max}}\right)$$

$$P_{nom} = T_{max} * W_{at\_max\_Torque}$$

- In the constant Torque region, the torque can be calculated as

$$T_{line(t)} = T_{\max(t)}$$

$$T_{line(t)} = \min(T_{\max(t)}, \max(-T_{\max(t)}, T_{line(t)}))$$

Also, the power can be calculated as

$$P_{line(t)} = T_{line(t)} * W_{rad(t)}$$

- In the Flux weakening region , where the Power is constant, and the speed of the Electric machine is greater than the nominal speed

$$T_{line(t)} = \frac{P_{nom}}{W_{rad(t)}}$$

$$P_{line(t)} = P_{nom}$$

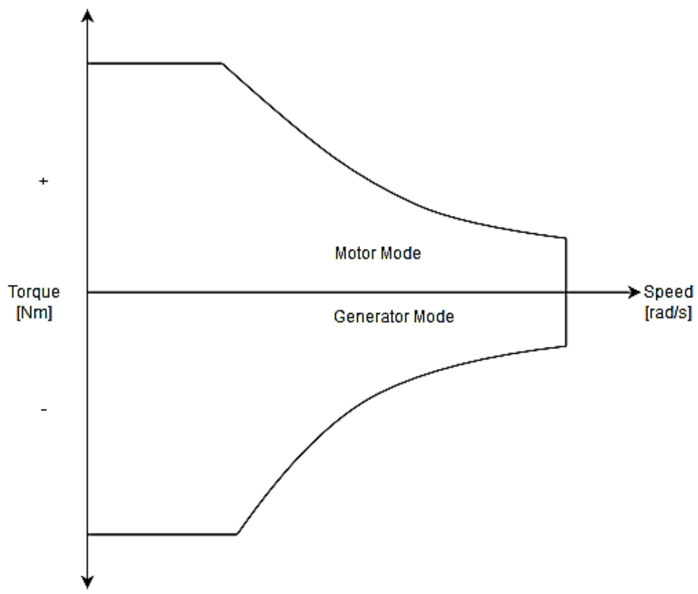
- In the region where the speed limit is reached is given as

$$T_{line(t)\,@\,W\_max} = 0$$

2.4.6 Efficiency of E-Machine

The Efficiency of the Electric machine is a necessary parameter and its calculation is dependent on the mode of operation of the motor. The following figure shows the formulations to be used the certain regimes.

Efficiency is the ratio between the output to input.



- In Motor Mode

$$Efficiency\ (t) = \frac{P_{motor(t)}}{P_{battery(t)}}$$

- In Generator Mode

$$Efficiency\ (t) = \frac{P_{battery(t)}}{P_{motor(t)}}$$

Figure 17-Torque operating regimes

2.4.7 Single Efficiency Measurement (Mapping of Losses)

It is possible to evaluate the efficiency of the E-machine using the equations in previously mentioned or already given by the technical data sheet with the E-machine. However, this efficiency measure at a point by point basis (Torque and Speed) can help in mapping the loss of the E-machine throughout the test cycle. Considering the following equations.

- In Motor Mode (+ve Torque Region)

$$P_{batt}(t) = \frac{P_{mot}(t)}{Efficiency\ (t)}$$

- In Generator Mode (-ve Torque Region)

$$P_{batt}(t) = -\frac{P_{mot}(t)}{Efficiency(t)}$$

Therefore, the Power dissipated, or losses can be hence given as

$$P_{diss}(t) = \big(1 - Efficiency(t)\big) * P_{batt}(t)$$

Such a Loss can then be shown or provided a map of torque and speed. The power losses (using SEM) can hence be compared with the power losses using experimental equation to evaluate the loss coefficients for the electrical machine.

2.4.8 Power loss experimental equations

The Power Losses are necessary to be calculated because it is further processed to evaluate the efficiency of the motor. The Machine used in this project is **IPM (Interior Permanent Magnet) Motor**.

The Power loss is the combination of the following

- **Base Power Loss** are the losses that are present even at zero Torque and Speed

$$P_{base}$$

- **Copper Power Loss** is the term that accounts for the heat produced by the electrical currents in the electric machine windings (if present) . Where K_T is the coefficient for the specific type of machine.

$$P_{Loss_{Copper(t)}} = K_T * T^2_{trac(t)}$$

- **Iron Power Loss (Core Losses)**, are losses present in the electric machine that are dependent on the magnetic properties of the material used for the construction of the core. Where K_I is the coefficient for the specific machine. The index "*i*" accounts for the type of machine used e.g. for *Induction machine* (IM) or *Permanent Machine* (PM) *P_Loss=0*. For a Synchronous Machine (SM) i=1.

$$P_{Loss_{Iron}(t)} = K_I * W^i_{rad(t)}$$

- **Windage Power loss** , accounts for the Losses sustained by the machine due to the resistance by air and rotation of the motor shaft. In our type of machine, we have neglected this type of Losses

$$P_{Loss_{Windage(t)}} = K_W * W^3_{rad(t)}$$

So, the Final Equation Becomes as following

$$\boldsymbol{P_{Loss}(t)} = P_{base} + P_{Loss_{Copper}}(t) + P_{Loss_{Iron}}(t) + P_{Loss_{Windage}}(t)$$

This Equation can be used for calculation of the power losses throughout the test cycle.

2.4.8.1 Loss coefficients

The loss coefficients can be calculated using several techniques such as:

- Curve fitting tools
- Least squares method(LSQ)
- Using single point efficiency

The first two methodologies may require a more technical background for performing it manually.However, software based methods implemented in softwares such as Matlab can be utilized to perform the calculation of loss coefficients. For curve fitting it is recommended to the curve fitting toolbox. For the latter LSQ method , pre-written functions can be used.

For what concerns the operating regimes of the E-machine , it shall be known that the constant power and constant torque regions may have a different loss coefficients to properly address the power losses. Therefore higher order polynomials (upto 3 degree : for w and T) shall be required and separate functions for both the constant torque and constant power region.

However, in this thesis a more simpler approach is utilized i.e Single point efficiency method . Power losses calculated as in Single Efficiency Measurement (Mapping of Losses) and using Power loss experimental equations. A graphical fit is used to observe where both the maps meet while incrementing/decrementing the loss coefficients. Using this methodology we can evaluate the loss coefficients.

3 Experimental Simulation

Simulation of a process is the imitation of a real-world situation. A Computer Simulation is not directly linked to a physical interaction between the machine it is run on and the physical system put under investigation. However , an experimental simulation is related to the study of the approximated model and testing it with experiments.

This chapter provides a understanding of the source code and how it can be implemented to evaluate Electric powertrain performance . The code will be explained according to the flowcharts presented in Vehicle Model Development. However, to have knowledge about the variables,functions and structure of the program it is necessary to refer to the **Appendix**

3.1 Program

3.1.1 Inputs

- These lines are used to clear all the values previously stored values , close all the programs in workspace and clear the command window.

```
clear all
close all
clc
```

- In order to work in Gnu Octave, it is necessary to have loaded previously some packages. So here we are loading the Input-Output Stream package to help the flow of data from spreadsheets and write onto files. Secondly, unlike other programs Octave doesn't initially allow copy and paste functionality on windows OS therefore the next line helps in introducing this feature to the workspace.

```
pkg load io
pkg load windows io
```

- Similarly, the need of Initialization of values is necessary such that the variable shall not contain some random data from the memory. The function **Initialization** is presented separately not to confuse it as a core part of the program. All values are

```
[t_cycle,v_cycle,v_cycle_ms,integ,deriv_err,err,v_act,
F_res_tot,u,P,I,D,F_achievable,limited,T_line,P_batt,
P_mot,P_Loss_Copper,P_Loss_Iron,P_Loss_Windage,
P_LOSS_EQ,E_in_batt,Efficiency] = initialization();
```

```
function
[t_cycle,v_cycle,v_cycle_ms,integ,deriv_err,err,v_act,F_res
_tot,u,P,I,D,F_achievable,limited,T_line,P_batt,P_mot,P_Los
s_Copper,P_Loss_Iron,P_Loss_Windage,P_LOSS_EQ,E_in_batt,Eff
iciency] = initialization()

cycle=xlsread('driving_cycle_wltp.xlsx','wltp');
t_cycle=cycle(:,1);
v_cycle=cycle(:,2);
v_cycle_ms=v_cycle/3.6;
integ(length(t_cycle+1))=0;
deriv_err(length(t_cycle+1))=0;
err(length(t_cycle+1))=0;
v_act(length(t_cycle+1))=0;
F_res_tot(length(t_cycle+1))=0;
u(length(t_cycle+1))=0;
P(length(t_cycle+1))=0;
I(length(t_cycle+1))=0;
D(length(t_cycle+1))=0;
F_achievable(length(t_cycle+1))=0;
t_cycle(1)=0;
limited = zeros(1,1801);
T_line(length(t_cycle+1))=0;
P_batt(length(t_cycle+1))=0;
P_mot(length(t_cycle+1))=0;
P_Loss_Copper(length(t_cycle+1))=0;
P_Loss_Iron(length(t_cycle+1))=0;
P_Loss_Windage(length(t_cycle+1))=0;

P_LOSS_EQ(length(t_cycle+1))=0;

E_in_batt(length(t_cycle+1))=0;
Efficiency(length(t_cycle+1))=0;
end
```

assigned zeros to the length of the test cycle. Also, the test cycle is called from the spreadsheet in this function

- The **Vehicle Related Parameters** are provided in a structured function. This function is called to evaluate the coast-down coefficients

```
v = veh_variables();
```

```
function [veh]=veh_variables(veh)
% Equivalent mass calculation
%-------------------------------------------------
veh.mass_vehicle=700;                            % Vehicle mass [kg]
veh.mass_passenger=300;                          % Passenger mass [kg]
veh.mass=veh.mass_vehicle+veh.mass_passenger;% Combined mass of vehicle +
passenger
veh.r_wheel=(13/2)*0.0254 +(165/1000*0.7);  %  Radius of wheel (R13 165)[m]
veh.n_w=4;                                      % Number of wheels [-]
veh.J_w=0.707;                                % Angular Inertia of wheels [kg*m^2]
veh.n_m=1;                                    % Number of Electric motors [-]
veh.J_m=0.0033;                               % Angular Inertia of motors [kg*m^2]
veh.i_gear_tot=8.67;                       % Total Transmission ratio [-] (or tau)
veh.J_t=0.0150;                       % Angular Inertia of transmission [kg*m^2]
veh.mass_rot=(veh.n_w*veh.J_w + veh.n_m*(veh.J_m*veh.i_gear_tot^2 +
veh.J_t))/veh.r_wheel^2;                      % Rotational mass [kg]
veh.m_tot=veh.mass+veh.mass_rot;              % Equivalent mass [kg]

% Coast-down coefficients calculation
%-------------------------------------------------
veh.slope=0;                                    % Road slope [0 degree ] no slope!
veh.g=9.81;                % Gravitational acceleration constant [m/s^2]

%Rolling resistance related
veh.f_2=6.5e-6;                       % Coeficcient of resistance [s^2/m^2]
veh.f_RRC=10.6;                       % RRC based on EU Label (F) [kg/ton]
veh.f_0=(18.0/1000)-(veh.f_2*(80/3.6)^2); % Rolling resistance coefficient

% Drag resistance related
veh.C_x=0.41;                                   % Aerodynamic coefficient
veh.Af=1.7;                                     % Vehicle frontal area [m^2]
veh.rho=1.184;                                  % Air density [kg/m^3] @ 25°C

% where F_tractive(t)=0 and slope=0 so longitudinal dynamic equation simplifies
to
% m_e*V_dot(t) + c0 + c1*V(t) + c2*V^2(t) =0

veh.C_0=veh.f_0*veh.mass*veh.g;   % Coast down coefficient c0
veh.C_1=0;                         % Coast down coefficient c1 [kg/s]
veh.C_2=veh.f_2*veh.mass*veh.g + (0.5*veh.rho*veh.C_x*veh.Af);
% Coast down coefficient c2 [kg/m]

veh.eta_trans=0.98;                % Transmission efficiency [-]
veh.tot_trans_ratio=veh.i_gear_tot/veh.r_wheel;%Total transmission ratio[1/m]

end
```

- The Electric Motor Map is read from the spreadsheet and assigned to the variables , also the maximum of the torque and speed is taken from the data which will be further used

```
[[W_em_max,T_em_max,P_em_max,Eta_em_max,W_em,T_em,Eta_em] =
electric_motor_map();
```

```
function [W_em_max,T_em_max,W_em,T_em,Eta_em] =
electric_motor_map()

eff_map=xlsread('S3M3_efficiency.xlsx','map');
W_em=eff_map(2:124,1);
T_em=eff_map(2:124,2);
Eta_em=eff_map(2:124,3);
mech_em=xlsread('S3M3_efficiency.xlsx','mechanical
behaviour');
W_em_max=mech_em(2:21,1);
T_em_max=mech_em(2:21,2);
P_em_max=mech_em(2:21,3);
Eta_em_max=mech_em(2:21,4);

end
```

- This function provides parameters for the proportional, derivative and integral control

```
[Kp,Ki,Kd,Ti,Td] = pid_parameters_method1();
```

```
function [Kp,Ki,Kd,Ti,Td] =pid_parameters_method1()

Kp=10*2;
Ki=10*2;
Kd=1*2;
Ti=Kp/Ki;
Td=Kd/Kp;
end
```

3.1.2 Process

3.1.2.1 Vehicle Longitudinal dynamics equation utilization

3.1.2.1.1 EV MODEL 1 (Process)

The Process Contains all the concepts and equations that are mandatory for evaluation of the vehicle longitudinal dynamics as well as the power losses on the motor.

The code will be presented in small blocks and will be explained according to the reference model that has been provided in **Vehicle Model Development**.

- First of all, we need to provide an iteration loop that can perform the process throughout the cycle, so we use a for loop

```
 for i=2:(length(v_cycle))
   -----PROCESS------
end
```

- The Time difference between the two instants is given as

```
dT=(t_cycle(i)-t_cycle(i-1));
```

- In order to design and give a **PID Driver approximation**: the following code is used, which calculates the error between the setpoint and process variable , hence performing a control by using the pedal signal i.e. "**u(t)**"

```
err(i)          =   (v_cycle_ms(i))-v_act(i);
deriv_err(i)    =   (err(i)-err(i-1))/dT;
integ(i)        =   integ(i-1) + err(i)*dT;

        P(i)=Kp*err(i);
        I(i)=(Kp/Ti)*integ(i);
        D(i)=(Kp*Td)*deriv_err(i);
        u(i)= (P(i) + I(i) + D(i));
```

- The Pedal Signal shall be saturated between the maximum and minimum of the pedal travel. The Integral shall be reset to prevent overshoot.

```
if u(i) > 100
     u(i) = 100;
 integ(i) = ( 100 - P(i) - D(i) )*Ti/Kp;
   elseif u(i) < -100
     u(i) = -100;
 integ(i) = ( -100 - P(i) - D(i) )*Ti/Kp;
     end
```

- The next step is to calculate the amount of torque the that shall be provided by the motor to sustain the speed and keep error at least as possible. The concepts are based on the **Transmission Model** previously presented. Thus , the Torque the given speed is calculated using interpolation function.

```
W_em_shaft(i)=((v_act(i))*v.tot_trans_ratio*rad_to_rpm);
W_em_shaft(i) = min(12000, max(0, W_em_shaft(i)));
T_full_shaft(i)=interp1(W_em_max,T_em_max,W_em_shaft(i),'nearest');
```

- The maximum torque available at the respective rotational speed of the motor is converted into the achievable tractive force. Moreover, the available torque is based on the pedal signal as percentage of the maximum torque available. The **Application of Feed Forward control** gives the amount of tractive force at wheels.

```
F_achievable(i)=T_full_shaft(i)*v.tot_trans_ratio*v.eta_trans;
T_trac(i)=(u(i)/100)*(T_full_shaft(i));
F_trac_veh(i)=T_trac(i)*(v.tot_trans_ratio)*v.eta_trans;
FF = C_0+(C_2*(v_act(i))^2);
F_trac_veh(i) = F_trac_veh(i) + FF;
```

- The Limit shall be placed on the Tractive force available by the motor if the requirement exceeds the limit and a flag shall be given to show the saturation of the motor at these points and present that the powertrain cannot follow the test cycle at these points.

```
if (F_trac_veh(i) >  F_achievable(i))
             F_trac_veh(i) = F_achievable(i);
             limited(i) = 1;
         elseif (F_trac_veh(i) < -F_achievable(i))
             F_trac_veh(i) = -F_achievable(i);
             limited(i) = 0.5;
         end
```

- So, the Tractive Torque shall be again re-calculated due the limitation placed on it. The **Coast down coefficients** are given in the resistance forces that complete the **Vehicle Longitudinal dynamics** equation. Here it can be observed that speed is saved to the next instant of the iteration loop ; the reason is that we can compare it with the same instant of the reference speed to follow to reduce the error.

```
T_trac(i) = (F_trac_veh(i))/ (v.tot_trans_ratio*v.eta_trans);
F_res_tot(i)=v.C_0+(v.C_2*(v_act(i))^2);

        if(t_cycle(i)<1800)
v_act(i+1)=((F_trac_veh(i) -F_res_tot(i))*(1/v.m_tot)*dT)+ v_act(i);
v_act(i+1)= min(38, max(0, v_act(i+1)));
        end
```

The code till this portion concludes the calculation of the operating points of the E-machine (Torque, Speed, Power). The tracking speed and Tractive force is also calculated. Now that we have the data on our variable we can proceed the process to evaluate the Map of efficiencies and Losses throughout the test cycle.

3.1.2.1.2 EV Model 2 (Process)

The EV Model 2 code has a difference in few areas such as there is no pedal scaling and the PID parameters are different however two parts that are given below

- The PID parameters are increased by the transmission ratio such that the "u(t)" changes from shaft torque to wheel torque.

```
[Kp_2,Ki_2,Kd_2,Ti_2,Td_2] = pid_parameters_method2();
```

```
function[Kp_2,Ki_2,Kd_2,Ti_2,Td_2]=pid_parameters_method2()

Kp_2=15*2*30.8981;
Ki_2=20*1*30.8981;
Kd_2=1*1* 30.8981;
Ti_2=Kp_2/Ki_2;
Td_2=Kd_2/Kp_2;
end
```

- The Following part is just different as compared to the 1st type of model
- The rest of the whole process Is same as EV Model 1

```
% PID Driver Pedal Input
P(i)=Kp_2*err(i);              % Proportional control
I(i)=(Kp_2/Ti_2)*integ(i);     % Integral control
D(i)=(Kp_2*Td_2)*deriv_err(i); % Derivative control
% The Pedal Signal in this controls the vehicle tractive force
u(i)= (P(i) + I(i) + D(i));    % PID driver
% Utilisation of Feed Forward Control
FF = v.C_0+(v.C_2*(v_act(i))^2);  % [N]
% Total force at wheel, FB+FF
F_trac_veh(i) = u(i)+FF;                % [N]
% Motor Shaft Speed based to provide required vehicle speed
W_em_shaft(i)=((v_act(i))*v.tot_trans_ratio*rad_to_rpm); % [RPM]
% Shaft Speed Saturation
W_em_shaft(i) = min(12000, max(0, W_em_shaft(i)));
% To interpolate the maximum torque available at the shaft
T_full_shaft(i)=interp1(W_em_max,T_em_max,W_em_shaft(i),'nearest');
% Achievable wheel force
F_achievable(i)=T_full_shaft(i)*v.tot_trans_ratio*v.eta_trans; %[N]
```

3.1.2.2 Calculation of Torque and Power Limit

- The Tractive torque shall be presented in way, such that there is a limitation on the maxima's and minima. A Torque and Power **Limit representation** provided gives an understanding. Initially we can place the data from the workspace into variables.

```
P_max=max(P_mot);
T_max=max(T_trac);
W_max=max(W_rad);
W_at_max_T=abs(P_max/T_max);
P_nom=T_max*W_at_max_T;
```

- The following loop calculates the limits of torque as well as power line.

```
for i=2:(length(v_cycle))
  if (W_rad(i)<= W_at_max_T)
          T_line(i)     = T_max;
          T_line(i)     = min(T_max, max(-T_max,T_line(i)));
          P_line(i)     = T_line(i)*W_rad(i);
     elseif((W_rad(i)> W_at_max_T) && (W_rad(i)~=W_max))
          T_line(i)=(P_nom)/W_rad(i);
          P_line(i)=P_nom;
     elseif(W_rad(i)==W_max)
          T_line(i)=0;
     end

     if(T_trac(i)>T_line(i))
          T_trac(i)=T_line(i);

     elseif(T_trac(i) < -T_line(i))
          T_trac(i)=T_line(i);
     end

      if(P_mot(i)>P_line(i))
     P_mot(i)=P_line(i);
     elseif(P_mot(i)<-P_line(i))
     P_mot(i)=P_line(i);
     end

 end
```

3.1.2.3 E-Machine Power Losses calculation

- The Following function is used for evaluation of the Electric machine power losses and is called to the main workspace for utilization. Power loss experimental equations are used for this calculation.

```
% Map of Power Losses
[P_LOSS_TOT,P_Loss_Copper,P_Loss_Iron,P_Loss_Windage]
=loss_map(W_rad,T_trac,v_cycle,v_act,P_Loss_Copper,P_
Loss_Iron,P_Loss_Windage,P_LOSS_EQ);
```

```
%% POWERLOSS MEASUREMENT BASED ON EXPERIMENTAL EQUATION (LUT)
function [P_LOSS_TOT,P_Loss_Copper,P_Loss_Iron,P_Loss_Windage] =
loss_map(W_rad,T_trac,v_cycle,v_act,P_Loss_Copper,P_Loss_Iron,P_Loss_Win
dage,P_LOSS_EQ)
% Function called for Evaluating the losses
[P_base,K_T,K_I,K_W] = Electric_machine_Loss_parameters();
```

```
function [P_base,K_T,K_I,K_W] =
Electric_machine_Loss_parameters()
% Electric Machine Loss Parameters
P_base=300;      % Losses Present at (T and W)=0
K_T=0.35;        % Copper Loss Coefficient
K_I=1.5;         % Iron Loss Coefficient
K_W=0.0;         % Windage Loss Coefficient
end
```

```
for i=2:(length(v_cycle))

    if(v_act(i)~=0)
        %Indices
        a=1;                % Unity
        b=2;                % Squared
        c=3;                % Cubic

        P_Loss_Copper(i)=(K_T*(T_trac(i))^b);
        P_Loss_Iron(i)=(K_I*(W_rad(i))^a);
        P_Loss_Windage(i)=(K_W*(W_rad(i))^c);

        %Power Loss Measured using single efficiency measurement
   P_LOSS_EQ(i)=P_base
               + P_Loss_Copper(i)
               + P_Loss_Iron(i)
               + P_Loss_Windage(i);
    end
P_LOSS_TOT=P_LOSS_EQ(:);

end
```

3.1.2.4 E-Machine Efficiency calculation

- The **Efficiency of E-Machine** can be calculated based on the following code. Where the already calculated power-losses are utilized and the following code is used to perform the efficiency calculation. The function used is provided below

```
% Map of Efficiencies
[ETA_MOT] =
efficiency_map(T_trac,W_rad,v_cycle,Efficiency,P_LOSS_TOT);
```

```
function
[ETA_MOT]=efficiency_map(T_trac,W_rad,v_cycle,Efficiency,P_LOSS)

for i=2:(length(v_cycle))

    % Efficiency Look Up Table
    if (T_trac(i)>0)
        Efficiency(i)=
(T_trac(i)*W_rad(i))/((T_trac(i)*W_rad(i))+ P_LOSS(i));

    elseif T_trac(i)<0
        Efficiency(i)=
(-T_trac(i)*W_rad(i))/((-T_trac(i)*W_rad(i)) + P_LOSS(i));

    end

    Efficiency(i)=Efficiency(i)*100;
end

ETA_MOT=Efficiency(:);

end
```

3.1.2.5 Battery Power Calculation

It is certain that the battery power input is contributed to the motor power and power losses. The following line gives the function and next block gives the methodology to calculate the battery power (Input Power)

```
% Battery Power ( based on Map of Losses )
[P_batt] = Battery_Power(P_mot,P_LOSS_TOT,v_cycle);
```

```
function [P_batt] = Battery_Power(P_mot,P_LOSS_TOT,v_cycle)
for i=1:length(v_cycle)
% Calculation of Power of the battery
P_batt(i)=P_mot(i)+ P_LOSS_TOT(i); % [W]
end
end
```

3.1.2.6 Single Point Efficiency Measurement

If the efficiency is already provided along with the rotational speed and torque as a 1D. The following function gives overview on Single Efficiency Measurement (Mapping of Losses) . This helps the checking of equations.

```
% SEM Methodology
[P_dissipated,ETA_MOT]=Single_Point_Efficiency_Method(P_mot,P_batt);
```

```
function[P_dissipated,ETA_MOT]=Single_Point_Efficiency_Method(P_mot,P_batt)
% Use the following function for measuring the efficiencies based on
% SINGLE POINT EFFICIENCY METHOD to create a map of LOSSES/DISSIPATED POWER
[W_EFF,T_EFF,ETA_MOT] = efficiency_map_read();
for i=2:(length(W_EFF))

      if (T_EFF(i)>=0)
        P_batt(i)=P_mot(i)/(ETA_MOT(i)/100);
      elseif( T_EFF(i)<0)
        P_batt(i)=(-P_mot(i))/(ETA_MOT(i)/100);
      end
      P_dissipated(i)=(1-(ETA_MOT(i)/100))*P_batt(i);

end
end
```

This concludes the process part of our developed code.

3.1.3 Outputs

The Outputs are the code is presented in the form of plots and datasheets. These Results will be discussed in the next chapter. The following functions are used for this purpose.

3.1.3.1 Plot Functions

These functions are used for plots

```
%---------------PLOTS------------------------%

% Tracking Speed Plot Function
plot_tracking_speed(t_cycle,v_act,v_cycle_ms);

% Tractive Force Plot Function
plot_traction_force(t_cycle,F_trac_veh);
plot_traction_vs_acheivable_force(t_cycle,F_trac_veh,F_achievable);

% Machine Torque Plot Functions
plot_torque_speed(T_line,W_rad,T_trac);

% Power Plot Functions
plot_power(t_cycle,P_batt,P_mot);
plot_Power_Speed(P_line,v_act,P_mot);

% Efficiency Plot Functions
plot_efficiency(t_cycle,ETA_MOT);
plot_efficiencies_map(W_rad,T_line,T_trac,ETA_MOT);

% Power Losses Plot Functions
plot_power_losses(t_cycle,P_LOSS_TOT,P_Loss_Copper,P_Loss_Iron);
plot_losses_copper_map(W_rad,T_line,T_trac,P_Loss_Copper);
plot_losses_Iron_map(W_rad,T_line,T_trac,P_Loss_Iron);
plot_losses_map(W_rad,T_line,T_trac,P_LOSS_TOT);
%------------------------------------------------%
```

3.1.3.2 Datasheet Functions

These functions are used to write the efficiency and losses onto spreadsheet files.

```
%-------DATASHEETS------------------------------
% Use Following Function to Write the Efficiency map Into a file
efficiency_map_write(T_trac,W_rad,ETA_MOT);
% % Use Following Function to Write the Loss map Into a file
loss_map_write(W_rad,T_trac,P_LOSS_TOT);
%----------------------------------------------
```

4 Results and Discussion

In this chapter the results obtained by processing the code mentioned in the previous chapter will be discussed.

The software that has been developed can evaluate the performance of the electric vehicle models (presented as EV model 1 and EV model 2) and give an approximation of the power losses and efficiency of the electric machine.

The results are presented by considering vehicle models and power losses of both algorithms developed.This creates the ease to observe the collective results of whole code developed.

4.1 Results from using EV Model 1 (algorithm)

- The vehicle longitudinal dynamics equation provides information about several variables after performing the **Dynamic Model** closed loop control. A variance in outputs can be observed by changing the algorithm from changing the algorithm. This can be observed in the initial tracking speed.

4.1.1 Tracking Speed

The Following result

- It can be observed from the figure that the algorithms used here occurs a saturation of the motor torque such that the traction torque is not enough to satisfy the last acceleration. This can be seen by looking at the flag variable colored in red. However, in this model there is also a saturation during the deceleration region in the mid cycle.

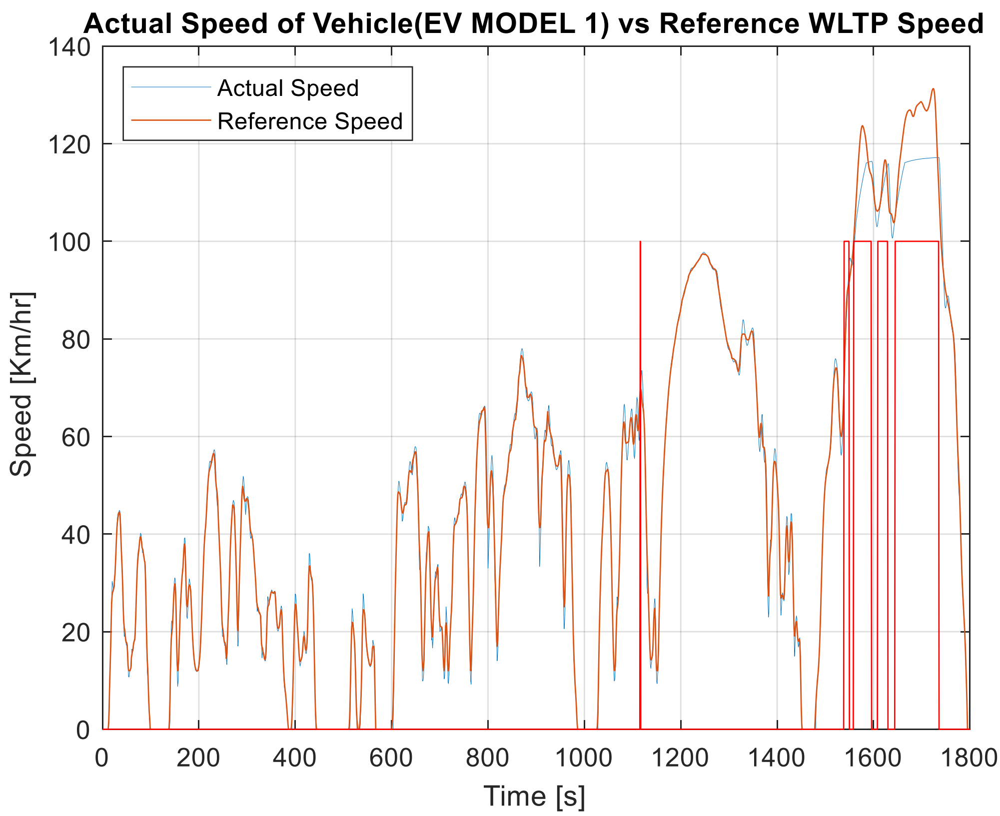


Figure 18-Tracking speed of EV Model 1

4.1.2 Tractive force and Achievable traction by vehicle

The tractive force is calculated based on **Transmission Model.** The vehicle operates both in the positive as well as negative traction force region. This means the E-machine is working in both motor mode and generator mode.

- The circled area in the figure shows where the limitation on the force or moreover on torque occurs and it can be clearly understood that the required traction force exceeds the achievable traction and therefore the in last acceleration E-machine saturates.

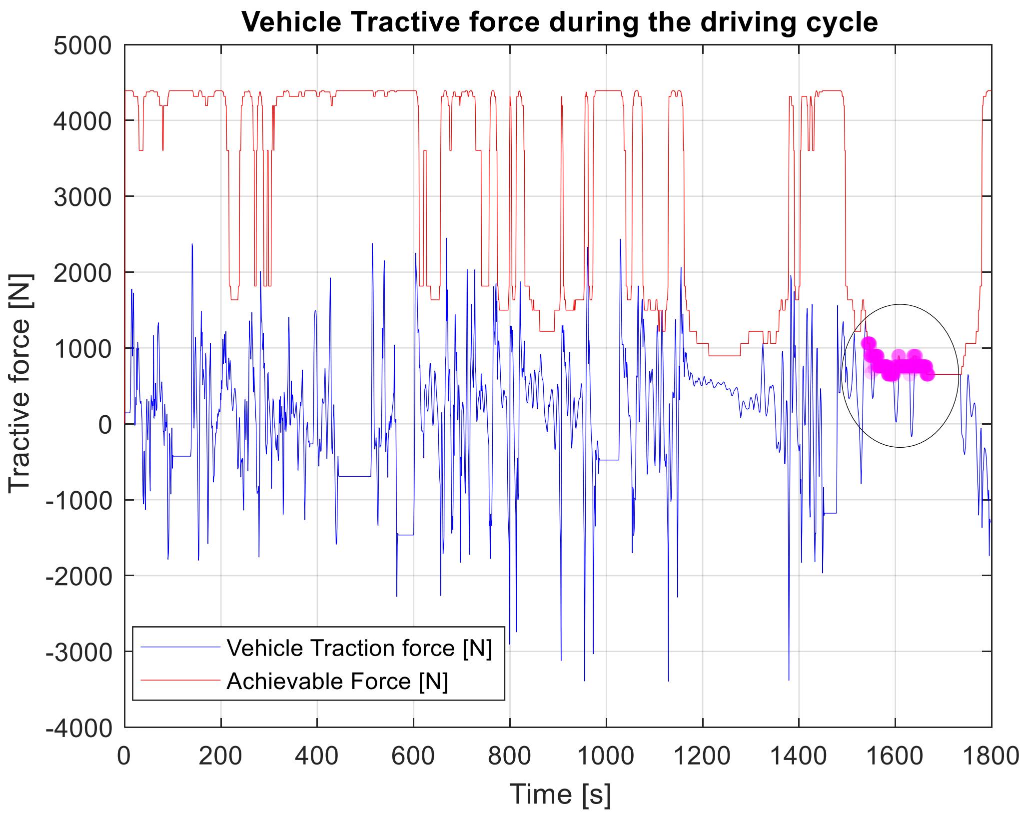


Figure 19-Traction force vs achievable force of EV Model 1

4.1.3 Torque-Speed Map

The torque-speed map gives a better understanding on the previous results.

- The operating points gives the idea in which region of the test cycle is the E-machine working. The E-machine operates more in the range of [-20,20 (Nm)] range. The last acceleration where the limitation occurs can be observed in blue. However, Torque and Power **Limit representation** priorly explained helps to get all the points below the curve and ignores the data points higher than the limit. It shall be noted that this map is presented for the continuous operation of the E-machine.

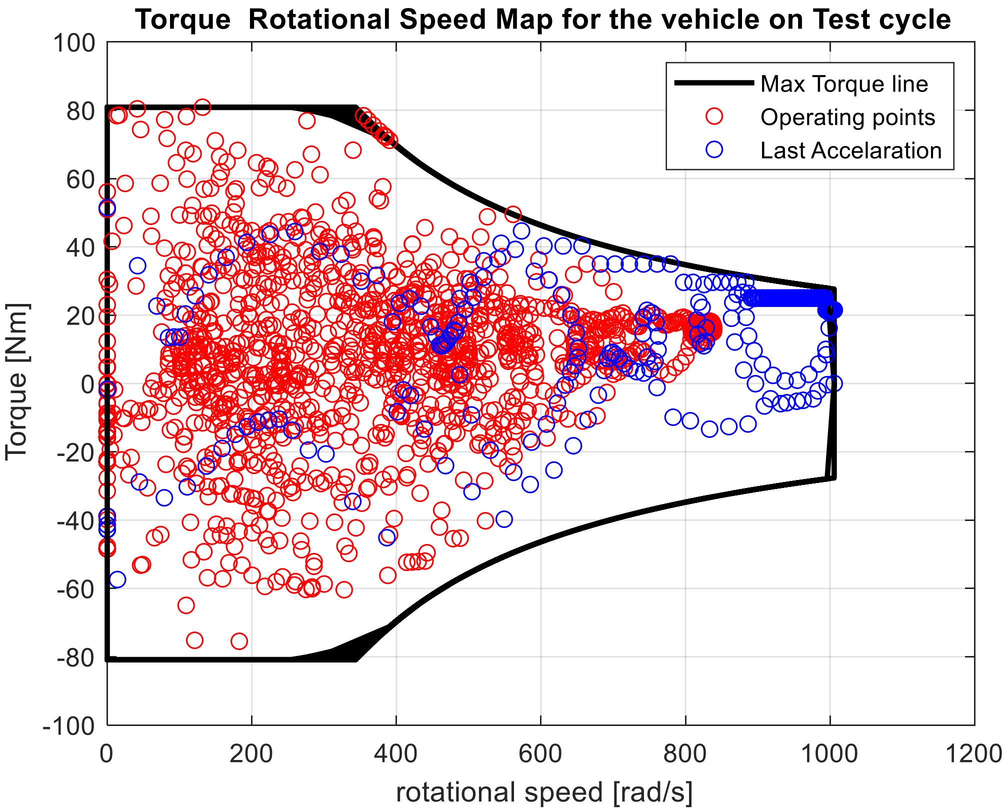


Figure 20- Torque Speed map for EV Model 1

4.1.4 Power of E-machine and battery

The Power is provided as an input by the battery and E-machine operates. However, there are some losses present during the conversion of electrical to mechanical energy. This mainly is dependent on the type of machine. The E-machine operates both in motor and regeneration mode (which power can be utilized for further use or recharging the battery).

- In the following figure it can be observed that the E-machine while operating in the negative region, the power input vs output is lower such that the E-machine is working at a higher conversion efficiency.

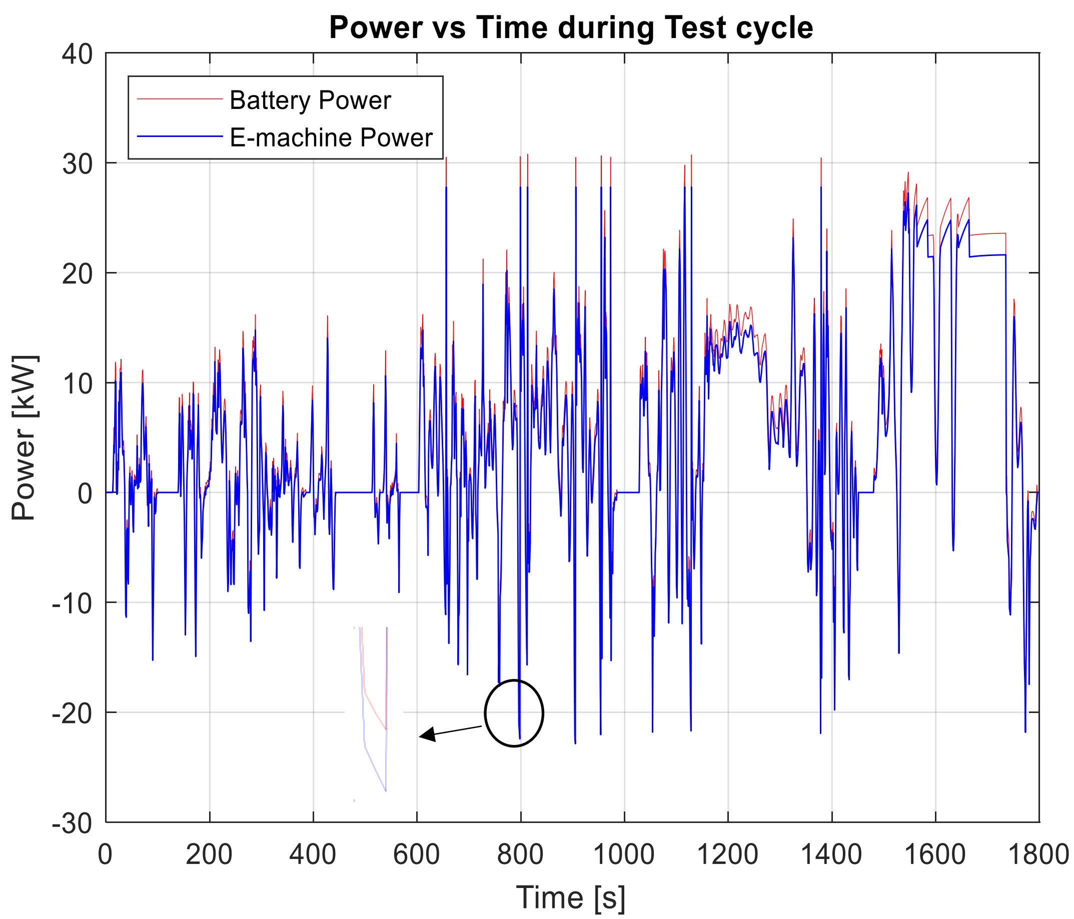


Figure 21-Battery power & E-machine power vs time for EV Model 1

- The Power-vehicle speed map gives the operating points of the test cycle. This gives an information that during the test cycle most of the operation occurs in motor mode at higher speed. Hence, we can conclude that our results are in line with the equations (we can know from the driving cycle).

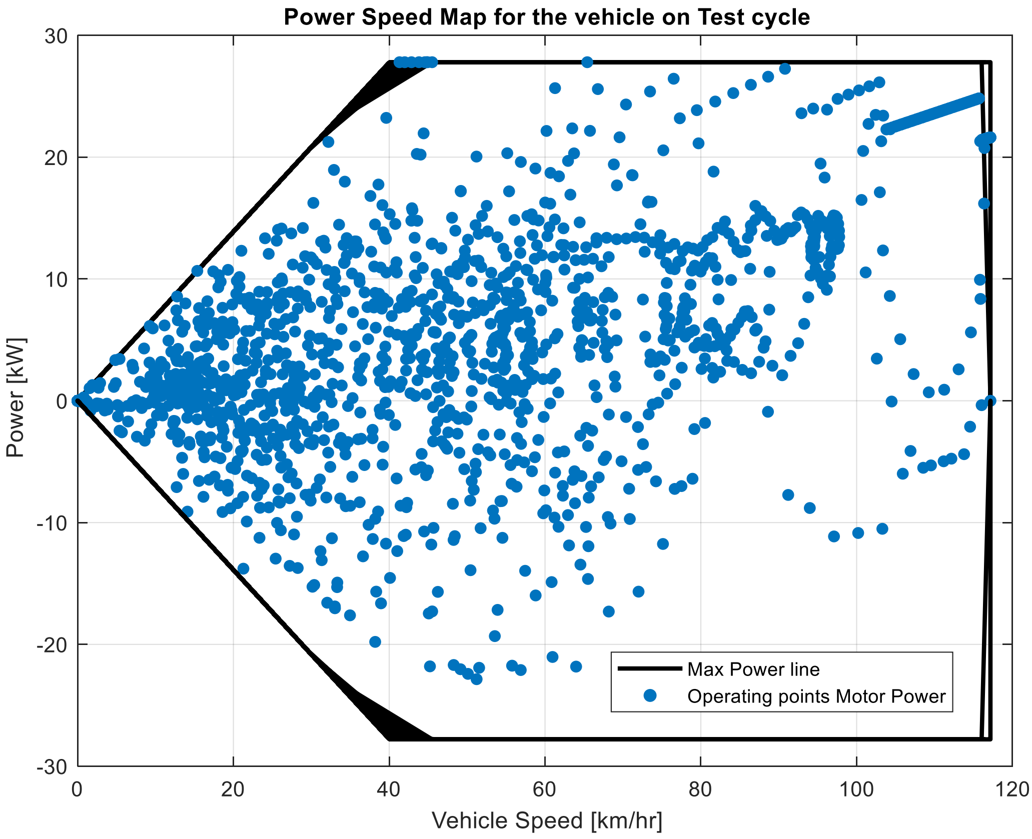


Figure 22-Power-Speed Map for EV Model 1

4.1.5 Efficiency of E-machine throughout test cycle

- The Following figure gives information on how the E-machine converts the electric energy into mechanical energy during the test cycle. The last part shows the highest efficiency operating region.

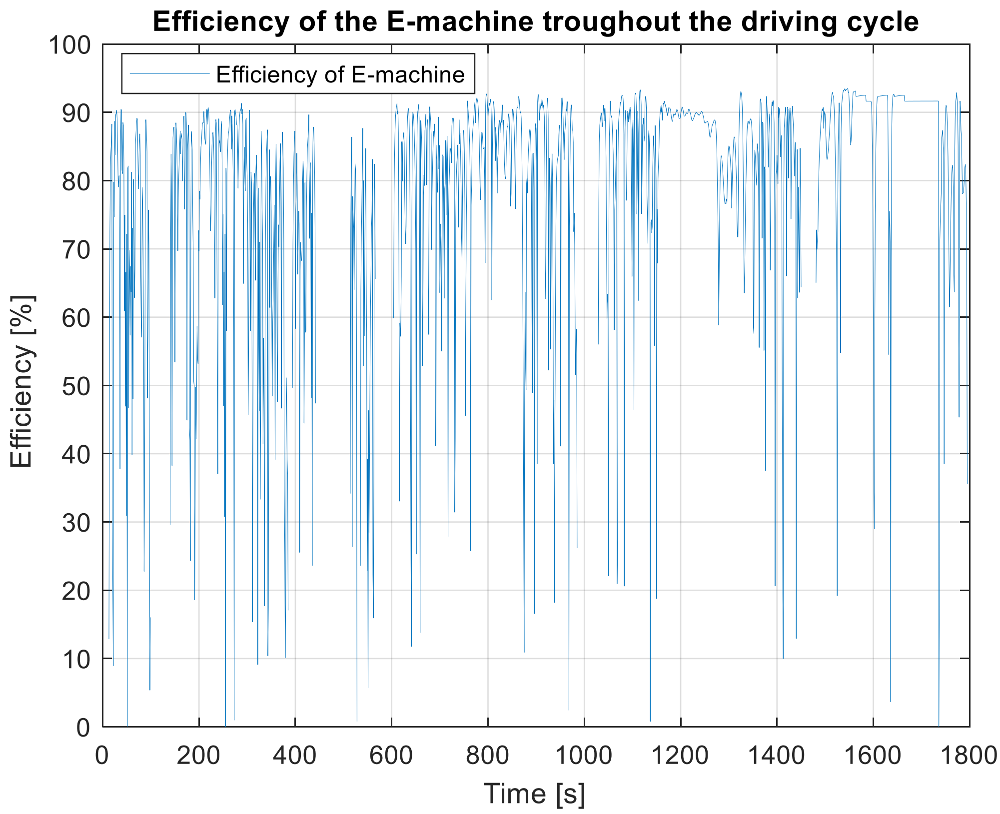
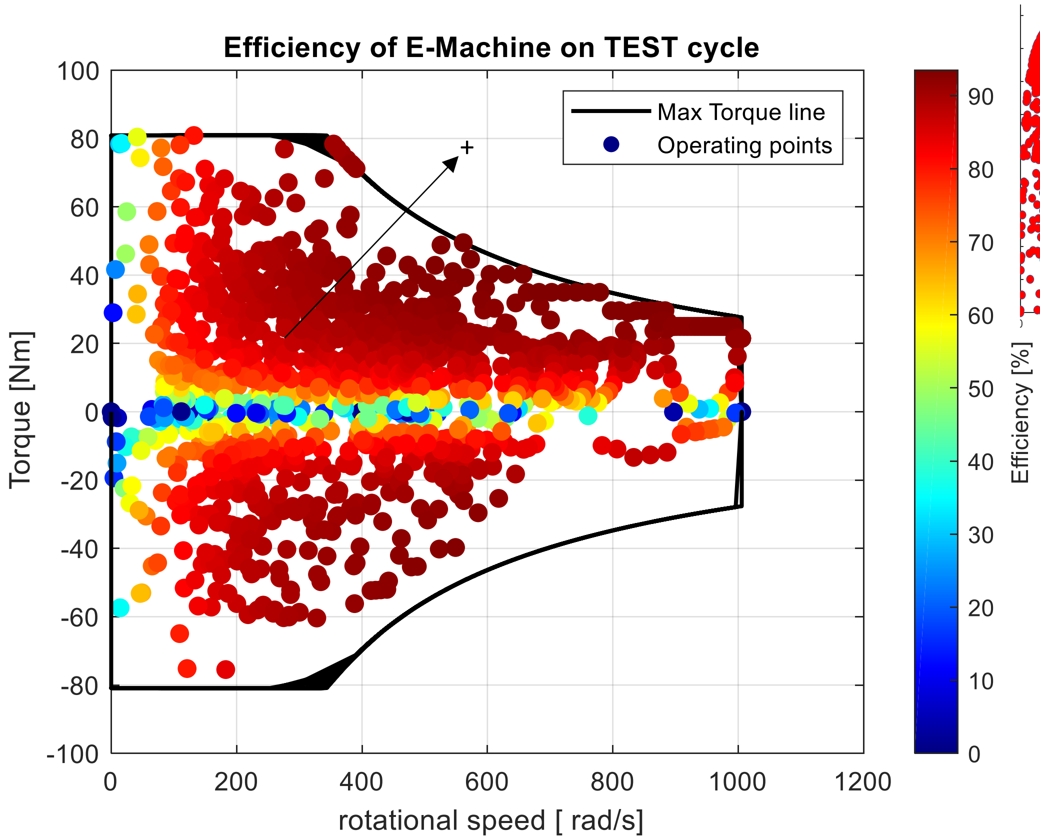


Figure 23- Efficiency of E-machine vs time for EV Model 1

- The Efficiency map of the E-machine gives the information on the distribution of efficiency. It can be observed at lower torque and high rotational speed the efficiency is lower. The higher torque and low speed the efficiency is lower. In the mid-range of torque and speed the efficiency is the highest. The small figure in right shows how the efficiency is increasing with the rotational speed.



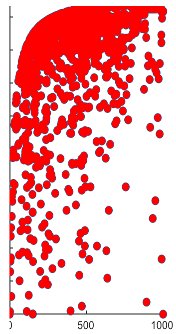


Figure 24- Efficiency Map for the E-machine for EV Model 1

4.1.6 Power Losses of E-machine

The **Power loss experimental equations** are used to get the following results.

- The Copper loss term mainly has higher losses in the constant torque region but little effect in the constant power region. Hence, in the lower speed region there are more Cu-losses than when the speed increases past the rated speed.
- The Iron loss has a higher effect in the constant power region; this is a characteristic in the IPM type of machines. However, it has a little effect in the constant torque region. The higher the torque the more the effect of this type of losses is lower.
- The Effect of Torque and rotational speed can be seen in the lower figure, where the torque effects the copper losses in a hyperbolic fashion and the speed is linear with the iron losses.

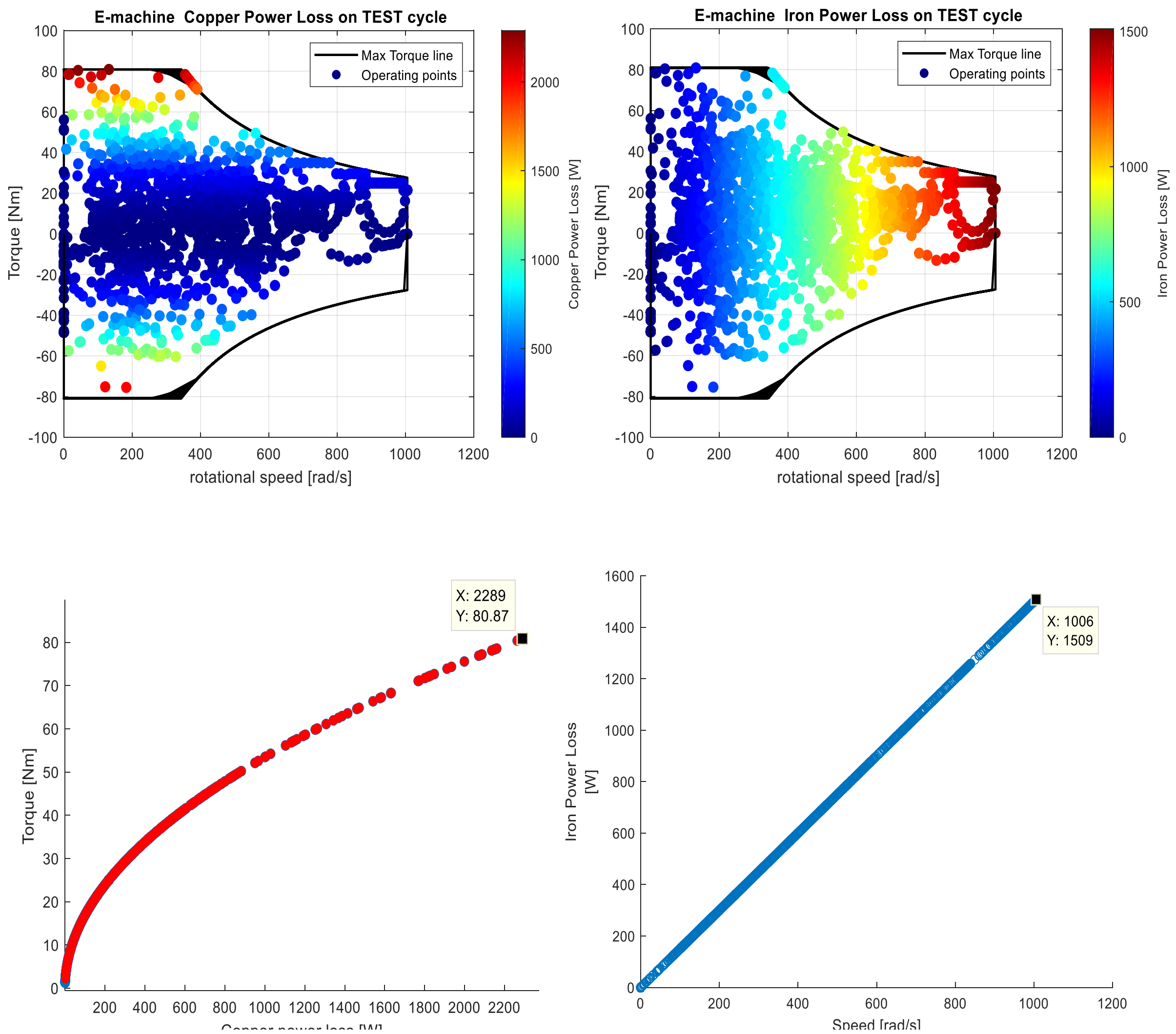


Figure 25- Copper and Iron Power Losses for EV Model 1

- The Combined losses (considering also the base power loss) along the test cycle provides the information that the higher losses occur in the constant torque region where is this behavior decreases in the flux weakening area.
- Another observation is that the copper loss gives more contribution as compared to the iron losses and the higher the torque the more the losses are and vice versa.

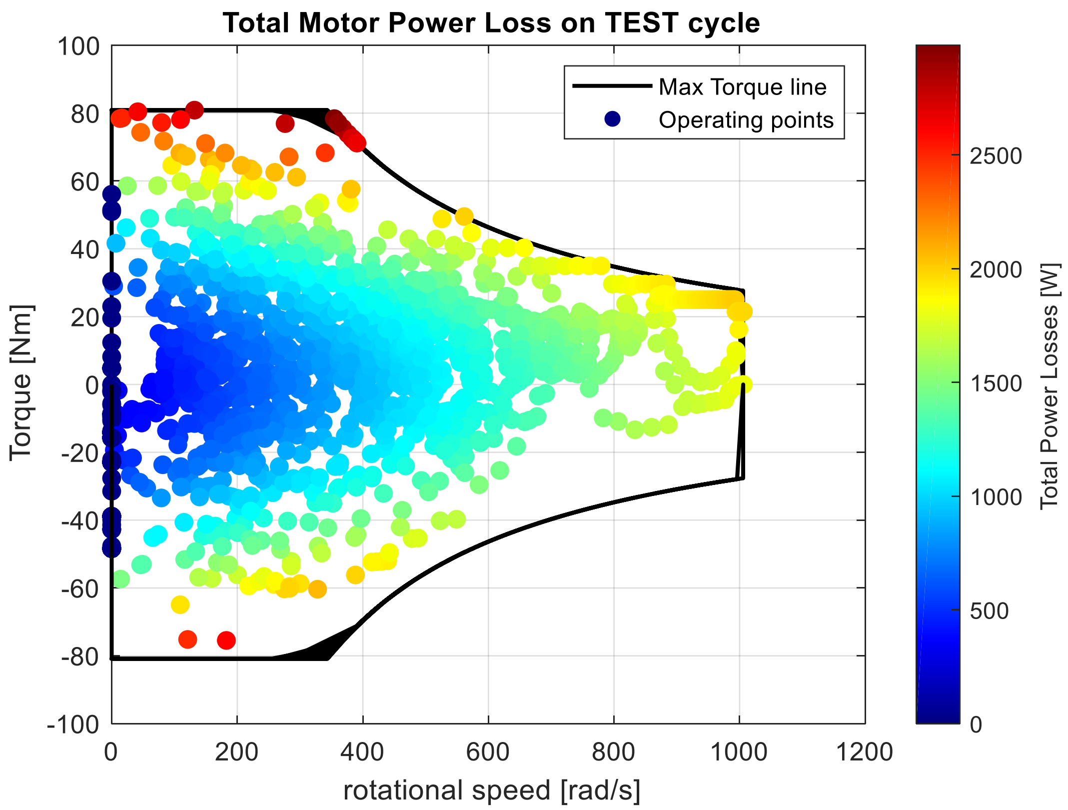


Figure 26-Power Loss Map for EV Model 1

4.2 Results from using EV Model 2 (algorithm)

4.2.1 Tracking Speed

The EV Model 2 algorithm follows the reference speed better as compared to the latter one due to approximate PID parameters in the entire range except the last acceleration. A comparison can be made by plotting the error of both algorithms however this is out of our main interest.

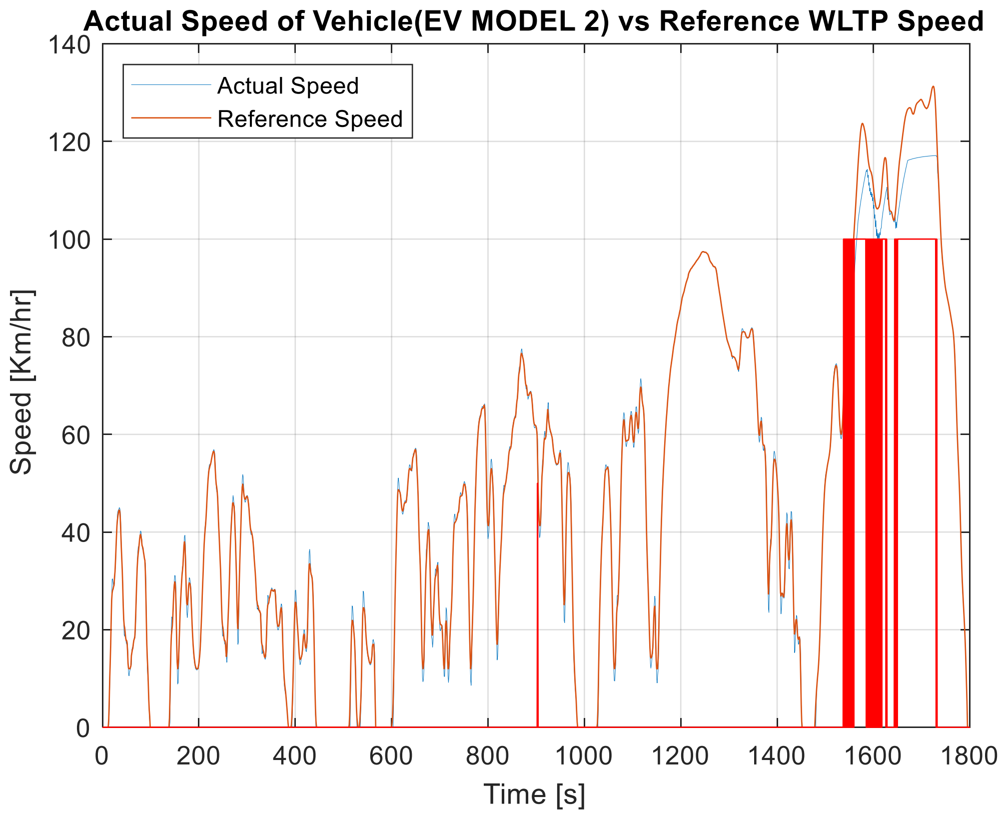


Figure 27-Tracking speed for EV Model 2

4.2.2 Tractive force and Achievable traction by vehicle

The circled area in the figure shows where the limitation on the force or moreover on torque occurs. and it can be clearly understood that the required traction force exceeds the achievable traction and therefore the in last acceleration E-machine saturates. This is almost the same result we already obtained in the latter type of algorithm.

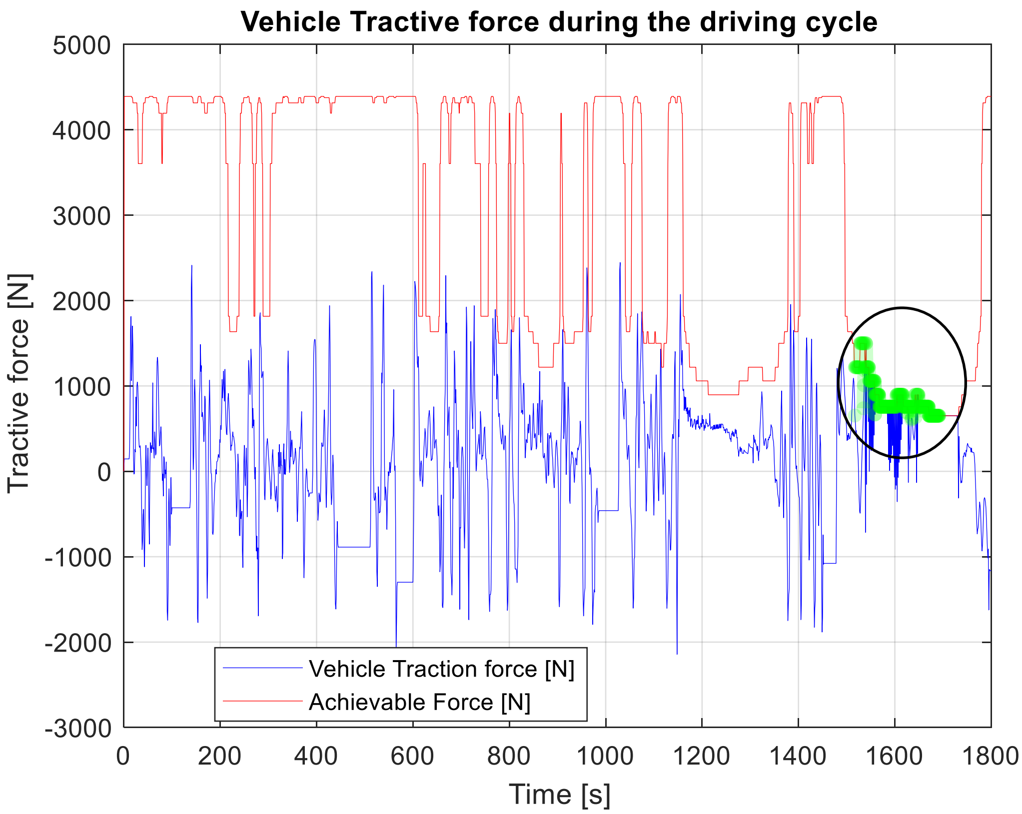


Figure 28-Tractive vs achievable force for EV Model 2

4.2.3 Torque-Speed Map

The torque-speed map gives a better understanding on the previous results.

- The operating points gives the idea in which region of the test cycle is the E-machine working. It can be observed that the last acceleration differs than the latter type of algorithm and the scattered data points (Torque, Speed) are slightly different.

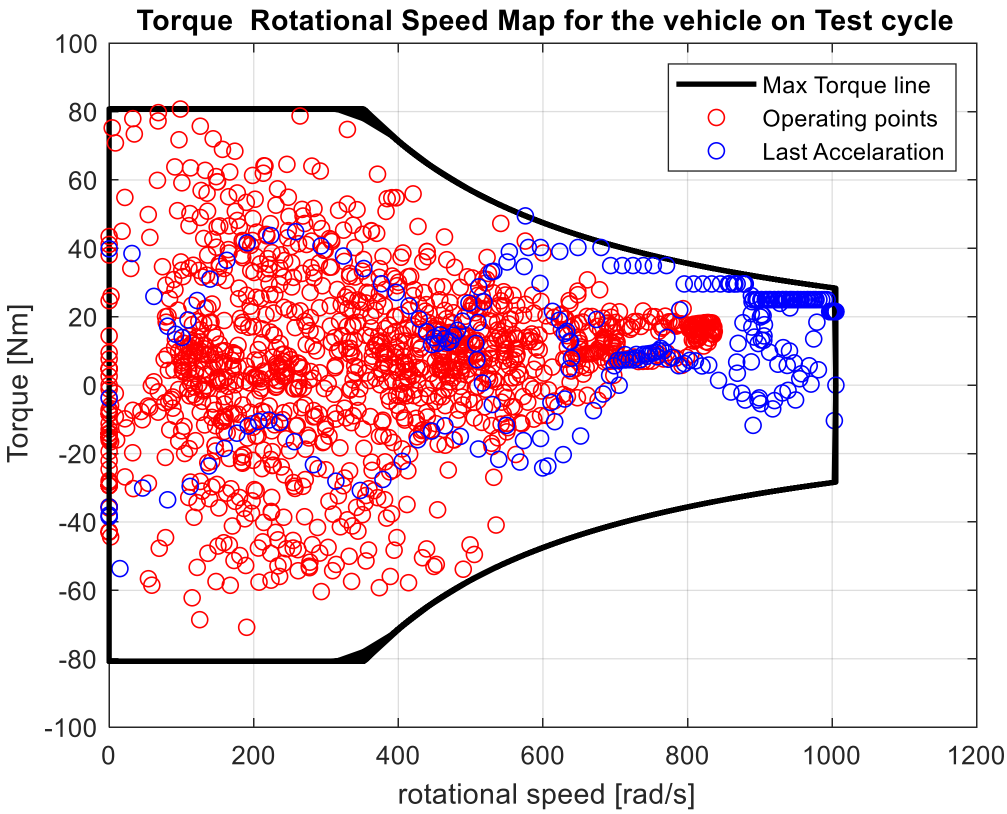


Figure 29-Torque Speed map for EV Model 2

4.2.4 Power of E-machine and battery

The noise in the last part of the cycle occurs since the E-machine saturates.

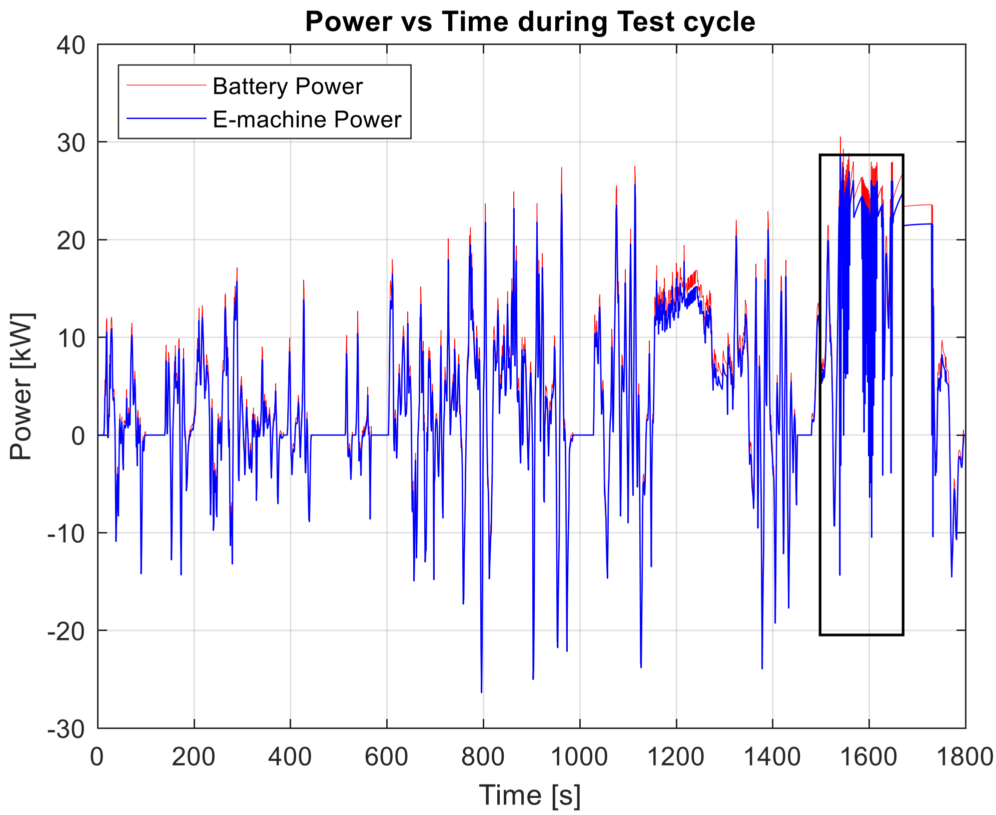


Figure 30-Battery power & E-machine power vs time for EV Model 2

- The Power-vehicle speed map here gives us the idea that the operating points mostly occur in the lower speed region (<40).

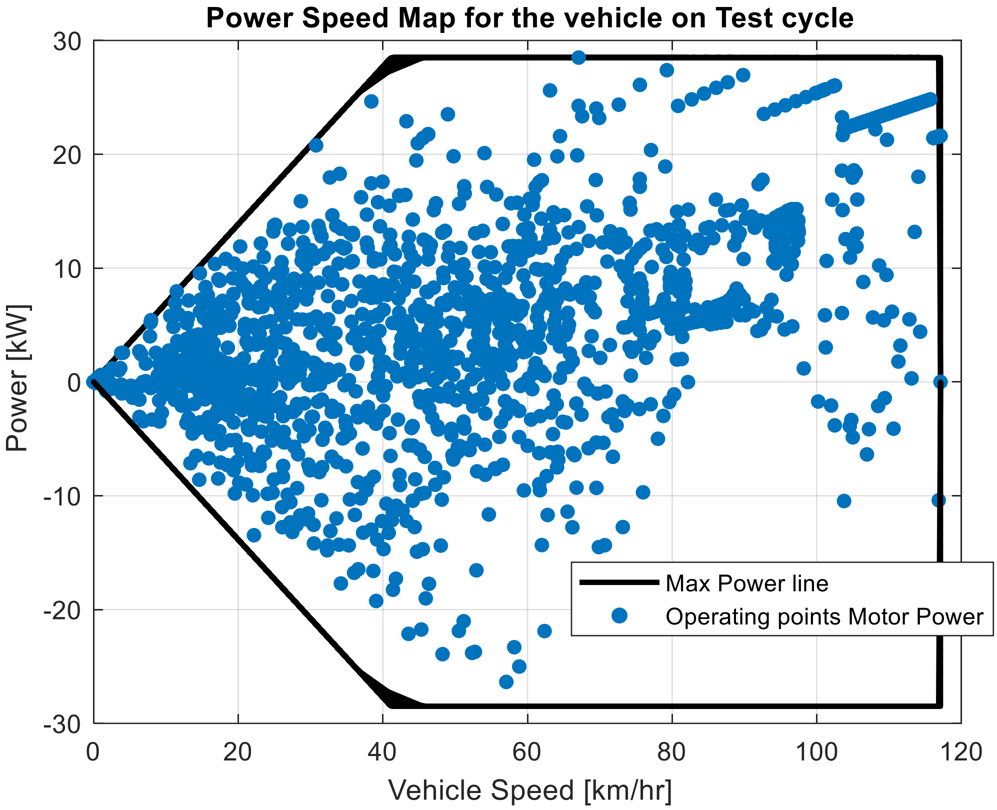


Figure 31-Power-Speed map for EV Model 2

4.2.5 Efficiency of E-machine throughout test cycle

- The Efficiency of the E-machine throughout the test cycle is given below. The last part shows that the efficiency is constant for a while, this behavior is the result of the saturation of the E-machine such that our code limits it to a specific point.

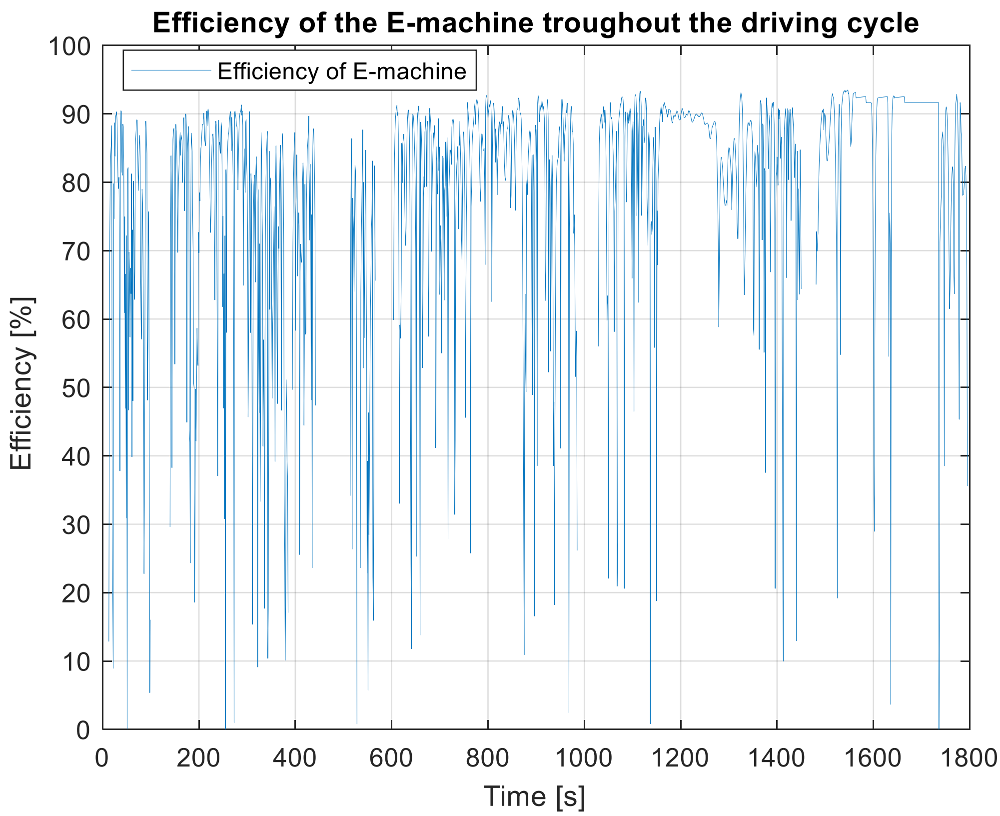


Figure 32-Efficiency of E-machine vs time for EV Model 2

- The Efficiency map of the E-machine gives the information on the distribution of efficiency. The same type of variation in the efficiency with respect to the torque-speed profile can be given.

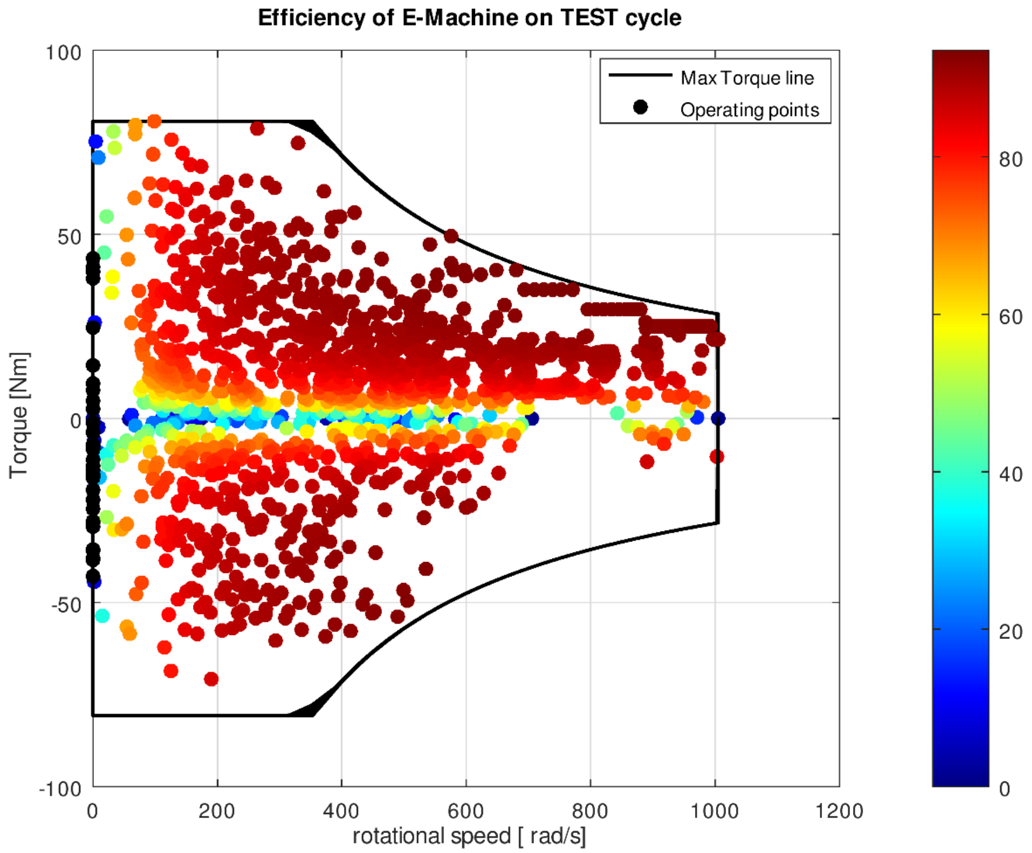


Figure 33-Efficiency map for EV Model 2

4.2.6 Power Losses of E-machine

The **Power loss experimental equations** are used to get the following results.

- The Results and observations are like those presented in the EV Model 1 algorithm. The highest values are same in both cases

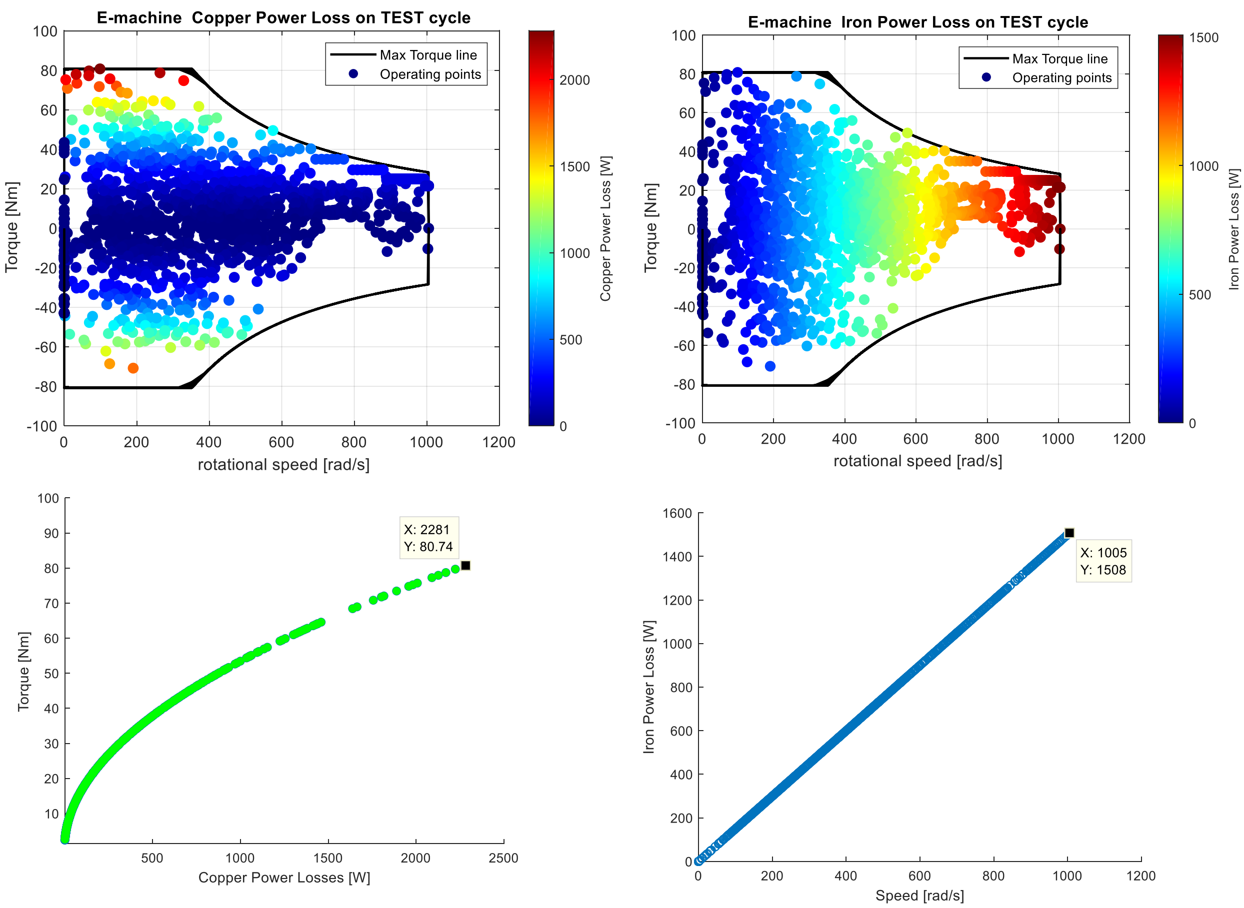


Figure 34-Copper and Iron Power Losses of E-machine for EV Model 2

- The Combined losses (considering also the base power loss) along the test cycle gives the same type of distribution as in the previous model algorithm

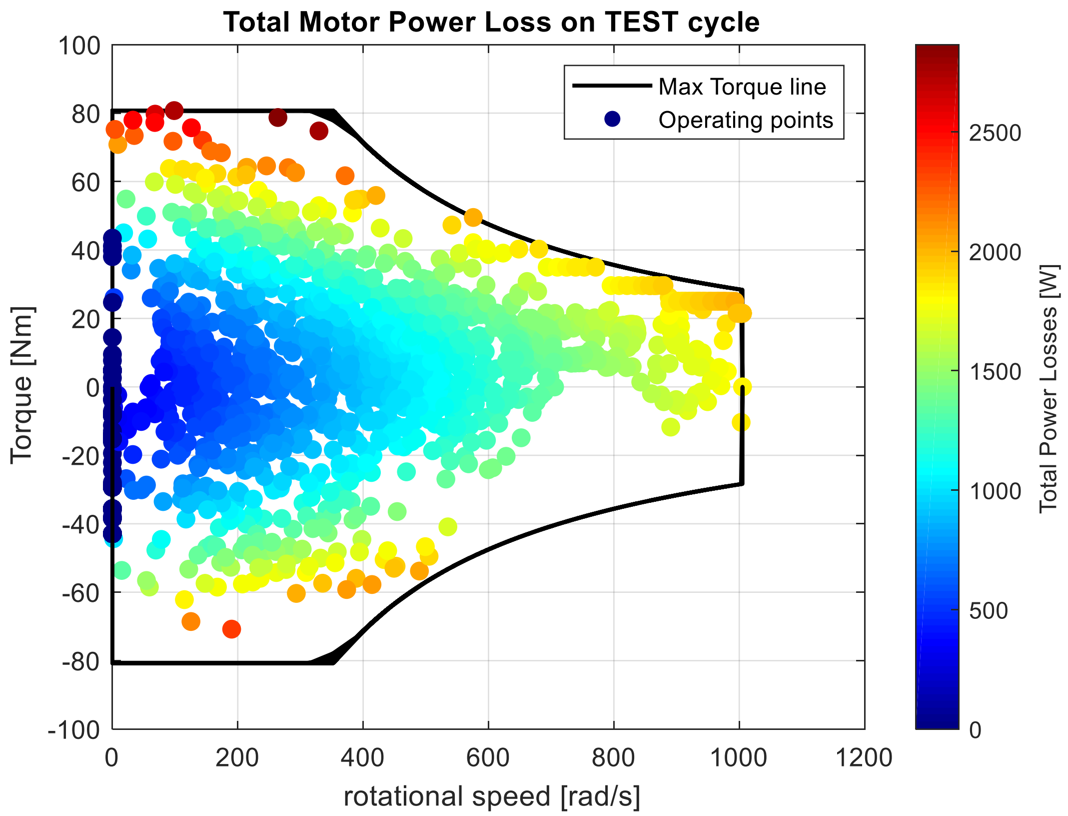


Figure 35-Total Power Losses of E-machine for EV Model 2

- Another observation is that the copper loss gives more contribution as compared to the iron losses and the higher the torque the more the losses are and vice versa.

4.3 Important numerical results

The Important observations from using both type of models is presented in the form of a table given below

No	Parameter	EV Model 1	EV Model 2	Unit
1	Vehicle speed (Maximum)	117	117	[Km/hr.]
2	PID (error_max)	3.9399	4.06	[-]
3	Maximum Efficiency of E-machine	93.49	93.49	[%]
4	Maximum Vehicle tractive force	2.4489e+03	2.4447e+03	[N]
5	Maximum Tractive torque	80.87	80.73	[Nm]
6	Maximum Battery Power	3.0776e+04	3.0517e+04	[W]
7	Maximum Motor Power	2.7795e+04	2.8496e+04	[W]
8	Maximum Copper Loss	2.2892e+03	2.2814e+03	[W]
9	Maximum Iron Loss	1.5086e+03	1.5078e+03	[W]
10	Maximum Total Power	2.9818e+03	2.8647e+03	[W]
11	Copper Loss Coefficient, K_T	0.35	0.35	[-]
12	Iron Loss Coefficient, K_I	1.5	1.5	[-]
13	Base Power Losses	300	300	[W]

Table 5-Comparison and Important results

- Motor_Efficiency_map (datasheet from the developed code)
- Power_loss_map (datasheet from the developed code)

The following are some brief of the research questions that shall be answered.

Q) **Does the provided electric machine satisfy the traction requirement posed by the vehicle motion during the test cycle?**

- Yes, but not in the last acceleration of the test cycle.

Q) **What is the operating area of vehicle, while considering longitudinal motion?**

- It's presented in the Torque-Speed Map and Power-Speed map.

Q**) What are the power losses due to the electric machine and how it may affect the tractive power?**

- The power losses are presented in previous comments with the important comments.

Q) **How to provide the approximate model (Algorithms) for EV Powertrain?**

- Both the presented models provide equivalent information. However, a more simplified algorithm such as in EV Model 2 can provide approximated results for evaluation of EV Powertrain.

4.4 Future Work

- The PID parameters are fixed due to the test cycle. However, if we change the driving cycle every time there will be a need of changing the Kp, Ki and Kd parameters. A solution to this problem is the use of a supervised machine learning algorithm or may be the use of a genetic based neural network such that it can calculate the optimal parameters to minimize the error and evaluate the whole developed code according to the given test cycle.
- There is a possibility to make a GUI based on the developed code, where the user can provide the test cycle, electric machine map and vehicle parameters and the user is provided the end results. This creates a more ease on the user end where the understanding of code is difficult.
- For what concerns the calculation of coefficients of copper, Iron and windage losses. In this thesis a graphical fit method using increment/decrement of the loss coefficients such that power losses of E-machine map using experimental equation and single point efficiency based power losses intersect has been used.However, a more advanced methodology using least squares or curve fitting can be used also.
- It is possible to add a Fuel Stack or Battery Model to the given code and evaluate the state of charge present in the battery after the driving cycle and calculate the consumption.

The ideas presented above were tried to be performed but as it isn't the scope of this thesis therefore it has been presented in the future work section.

References

- LEV Project -Powertrain (Vehicle Consumption and range) – Confidential (Provided by Supervisor)
- Vehicle longitudinal dynamics (Provided by Supervisor)
- Electric Motor Map (Provided by Supervisor)

Appendix

Installation Manual:

In order to install the software developed it is necessary to present the overview of requirements.

Software package name: **EVLD_Program** *EVLD stands for Electric Vehicle Longitudinal Dynamics

Hardware Requirements: No Specific hardware component is requested for this program.

Software Requirements: GNU Octave 4.4.0 or above

OS Tested on : Windows 10

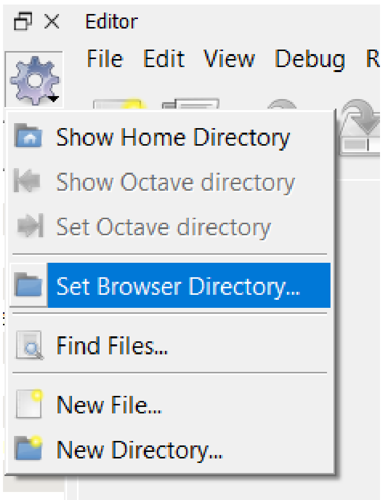
Download Link: https://www.gnu.org/software/octave/download.html

Gnu Package to be installed: No! , already present in program

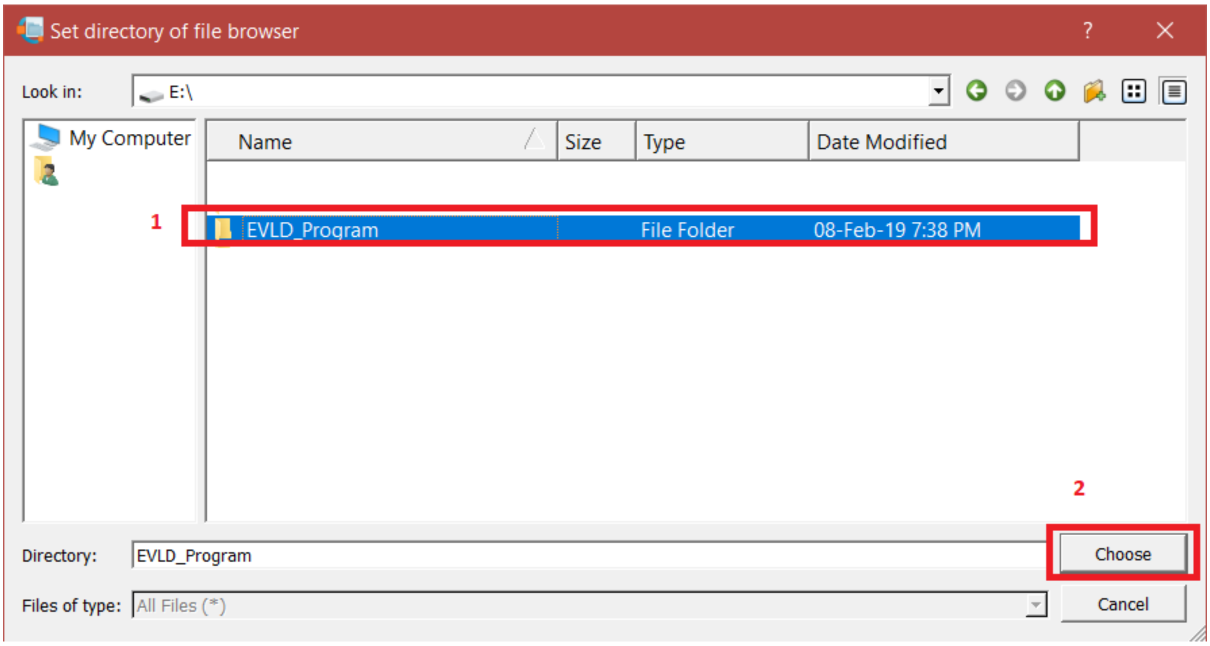
Installation steps:

The following steps are useful for getting started with the EVLD Program.

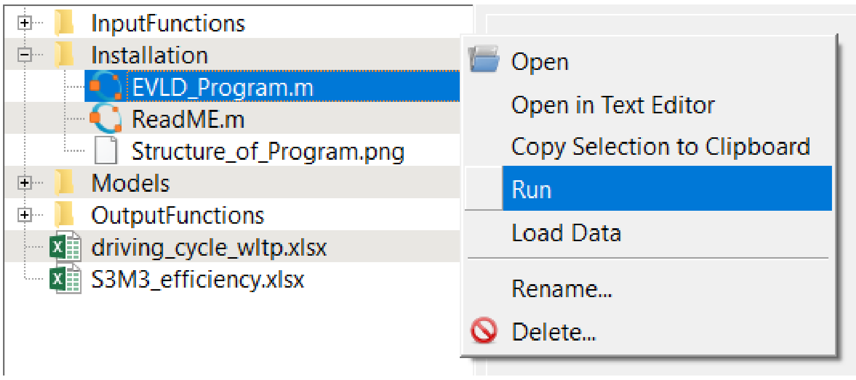
1) Copy/Download the folder "EVLD_Program.zip" on your computer.
2) Extract to folder "EVLD_Program"
3) Open GNU Octave 4.4.0 (or above version) and click "**Set Browser Directory**".



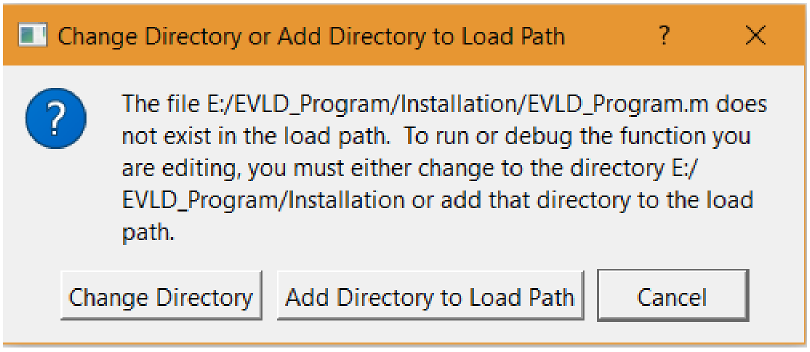
4) Goto the folder where the program has been extracted and choose “EVLD_Program”



5) Goto “Installation” folder and Right click on "**EVLD_Program.m**" and select "Run"



6) A window will appear. Select "Add Directory to Load Path"

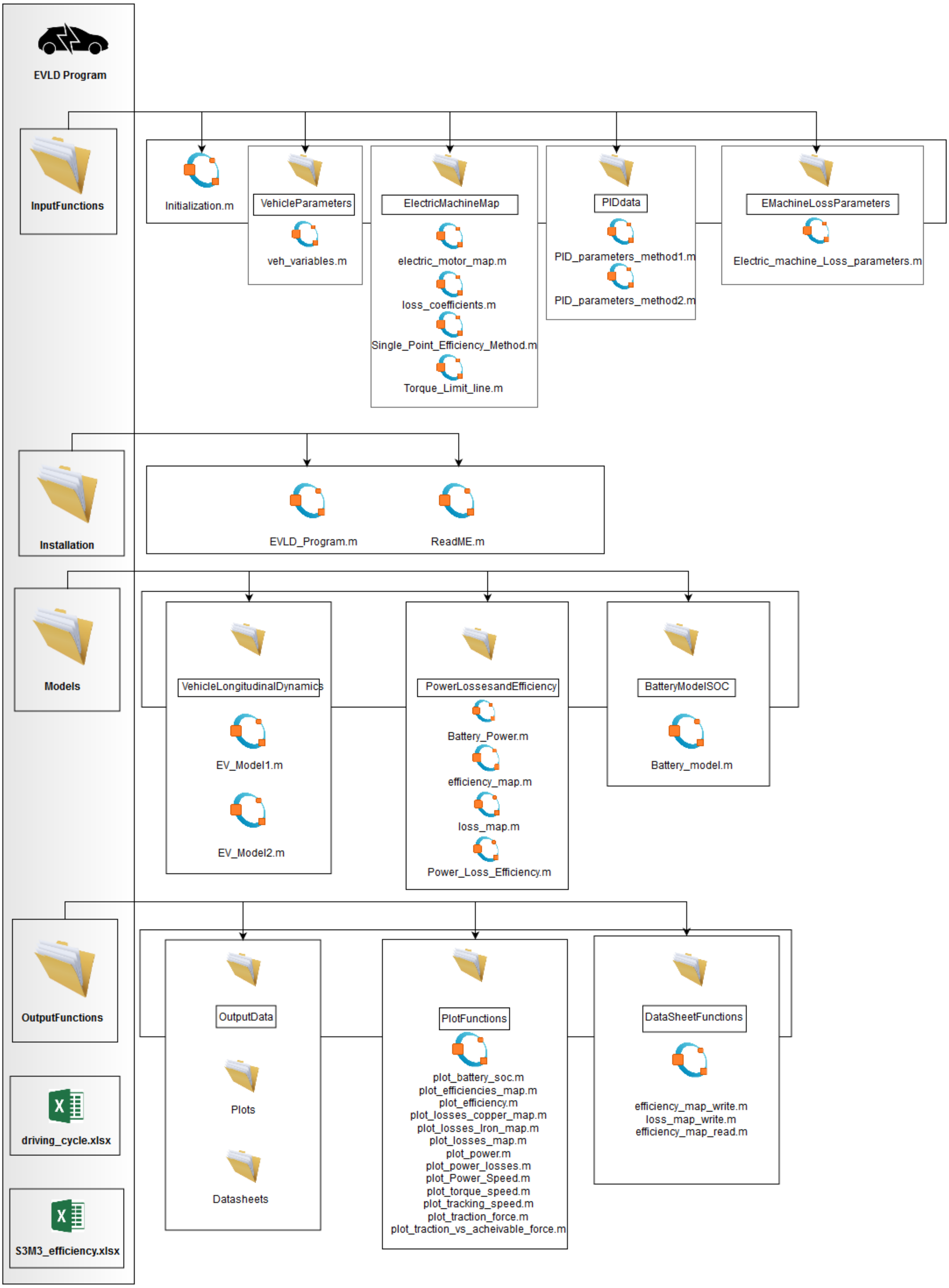


7) In Command Window write "**run EVLD_Program**" (case sensitive) as shown in the figure and press **Enter.**

8) This procedure loads all the files in the EVLD_Program directory and hence we can use the software now.

Structure of the Program:

The following figure shows the structure of the program in a graphical view. However, the functions performed by each file is also presented.



Functions performed by different files (Overview):

EVLD_Program... Main Folder (PARENT DIRECTORY)

- *InputFunctions..Necessary functions that provide support to "MODELS"*
  - *initialization.m ....For Initialization of all values*
  - *VehicleParameters..........................Vehicle data as a structure*
    - *veh_variables.m (The vehicle data i.e mass,coast-down coefficients)*
  - *EMachineLOSSCoefficients ....Coefficients for Power loss (K_I,K_T,K_W)*
    - *Electric_machine_Loss_parameters.m (The electric machine loss coefficients)*
  - *PIDdata...................................... Coefficients of PID Control(Kp,Ki,Kd)*
    - *Pid_parameters_method1.m (PID coefficients for pedal based algorithm)*
    - *Pid_parameters_method2.m (PID coefficients for force input algorithm)*
  - *ElectricMachineMap..................E-Machine Torque-Speed Map*
    - *Electric_motor_map.m (Function file used for importing E-Machine map)*
    - *Loss_coefficients.m (The Calculation of Loss coefficients for E-Machine)*
    - *Single_point_efficiency_method.m (Utilisation of SEM for efficiency)*
    - *Torque_limit_line.m (To limit Torque and Speed to its maximum)*
- *Installation..Program Installation related data*
  - *EVLD_Program.m.....................To start program and include all files to path*
  - *ReadMe.m................................Reference for Installation (Includes this list)*
- *Models..Models to show how program can do different tasks*
  - *VehicleLongitudalDynamics......Vehicle longitudinal dynamics model*
    - *EV_Model1.m (The Vehicle Model based on Pedal scaling algorithm)*
    - *EV_Model2.m (The Vehicle Model based on Input Force algorithm)*
  - *PowerLossesandEfficiency........Evaluator of E-machine performance*
    - *Battery_Power.m (Function controlled by Process to get battery power)*
    - *Efficiency_map.m (Function controlled by Process to get efficiency map)*
    - *Loss_map.m (Function controlled by Process to get power loss map)*
    - *Power_Loss_Efficiency.m (Power Loss and Efficiency evaluation Model)*
  - *BatteryModelSOC....................(EXTRA)Battery SOC estimation after Test cycle*
    - *Battery_model.m (Battery Model to find consumption , presented as example)*
- *OutputFunctions....................................Functions that present Results of the Models*
  - *PlotFunctions............................The functions to plot useful data and relationships*
  - *DataSheetFunctions..................For saving data to files*
  - *OutputData..............................The necessary data to present results*
    - *Plots............................The plots stored in .png format here*
    - *DataSheet...................The datasheet output from models are saved here*
- *driving_cycle.xlsx..................................TEST Driving Cycle*
- *S3M3_efficiency.xlsx.............................Electric Machine Torque Speed Efficiency Map*

Nomenclature:

Notations	Variable	Code name	Unit
T_{cycle}	*Cycle time*	t_cycle	[s]
V_{cycle}	*Cycle speed*	v_cycle	[km/hr]
$V_{cycle_{ms}}$	*Cycle speed in m/s*	v_cycle_ms	[m/s]
$Error$	*Error*	err	[m/s]
$Integral$	*Integral*	integ	[m]
Derivative	*Derivative of error*	deriv_err	[m/s2]
dT	*Cycle time*	dT	[s]
P	*Proportional term*	P	[m/s]
I	*Integral term*	I	[m/s]
D	*Derivative term*	D	[m/s]
K_p , K_{p_2}	*Proportional gain*	Kp , Kp_2	[-]
K_I , K_{I_2}	*Integral gain*	Ki, Ki_2	[-]
K_d , K_{d_2}	*Derivative gain*	Kd , Kd_2	[-]
T_I , T_{I_2}	*Controller Reset time*	Ti , Ti_2	[s]
T_d, T_{d_2}	*Controller Derivative time*	Td , Td_2	[s]
u	*PID Response*	u	[m/s]
$W_{em_{shaft}}$	*Motor Shaft speed*	W_em_shaft	[rpm]
V_{act}	*Vehicle actual speed*	v_act	[m/s]
τ	*Transmission Ratio*	v.tot_trans_ratio	[1/m]
m	*Mass of Vehicle*	v.mass	[kg]
C_x	*Aerodynamic Coefficient*	v.C_x	[-]
C_{rr}	*Rolling resistance Coefficient*	v.f_0	[-]
K	*Coefficient of resistance*	v.f_2	[s^2/m^2]
A_f	*Frontal area of vehicle*	v.Af	[m^2]
ρ	*Air density*	v.rho	[kg/m^3]
g	*Gravitational acceleration*	v.g	[m/s^2]
$i_{gear_{tot}}$	*Total transmission ratio*	v.i_gear_tot	[-]
η_T	*Transmission efficiency*	v.eta_trans	[-]
r	*Radius of wheel*	r.wheel	[m]
C_0	*Coast-down coefficient C0*	v.C_0	[N]
C_1	*Coast-down coefficient C1*	v.C_1	[kg/s]
C_2	*Coast-down coefficient C2*	v.C_2	[kg/m]
T_{full}	*Full Shaft Torque*	T_full_shaft	[Nm]
$W_{em}\ max$	*Maximum motor speed*	W_em_max	[rpm]
$T_{em}max$	*Maximum torque speed*	T_em_max	[Nm]
$F\_achievable$	*Achievable wheel Force*	F_achievable	[N]
FF	*Feed Forward*	FF	[N]
T_{trac}	*Tractive Torque*	T_trac	[Nm]
$F_{trac_{veh}}$	*Tractive wheel Force*	F_trac_veh	[N]
$Limit$	*Limit Flag*	Limited	[-]
$F_{res_{tot}}$	*Total resistance Force*	F_res_tot	[N]
W_{rad}	*Rotational Shaft speed*	W_rad	[rad/s]
P_{te}	*Tractive Power*	P_te	[W]

$\boldsymbol{P_{mot}}$	*Motor Power*	P_mot	[W]
$\boldsymbol{P_{mot,max}}$	*Maximum Motor Power*	P_max	[W]
$\boldsymbol{T\_max}$	*Maximum Motor Torque*	T_max	[Nm]
$\boldsymbol{W_{nominal}}$	*Nominal Speed*	W_at_max_T	[rad/s]
$\boldsymbol{P_{nominal}}$	*Nominal Power*	P_nom	[W]
$\boldsymbol{Torque_{line}}$	*Torque line*	T_line	[Nm]
$\boldsymbol{Power_{line}}$	*Power line*	P_line	[W]
$\boldsymbol{K_T}$	*Copper loss coefficient*	K_T	[-]
$\boldsymbol{K_I}$	*Iron loss coefficient*	K_I	[-]
$\boldsymbol{P_{Loss_{TOT}}}$	*Total Power Loss*	P_LOSS_TOT	[W]
$\boldsymbol{\eta_m}$	*Efficiency of Motor*	ETA_MOT	[%]
$\boldsymbol{P_{batt}}$	*Battery Power*	P_batt	[W]

Table 6- Nomenclature of variables

Functions I/O Stream:

Name of function	Inputs	Outputs
Initialization	()	[all variables used in program]
Electric_motor_map	()	(W_em_max,T_em_max,P_em_max,Eta_em_max, W_em,T_em,Eta_em)
Single_point_efficiency_method	(P_mot,P_batt)	[P_dissipated,ETA_MOT]
Torque_limit_line	(T_x,P_x)	[T_line,P_line]
Limit_line_nested	(W_rad,W_max,W_at_max_T,P_nom,T_max,T_trac,v_cycle)	[T_x,P_x]
Electric_machine_loss_parameters	()	[P_base,K_T,K_I,K_W]
Pid_parameters_method1	()	[Kp,Ki,Kd,Ti,Td]
Pid_parameters_method2	()	[Kp_2,Ki_2,Kd_2,Ti_2,Td_2]
Veh_variables	Structure with vehicle data [input]	v.variablename
Battery_power	(P_mot,P_LOSS_TOT,v_cycle)	[P_batt]
Efficiency_map	(T_trac,W_rad,v_cycle,Efficiency,P_LOSS)	[ETA_MOT]
Loss_map	(W_rad,T_trac,v_cycle,v_act,P_Loss_Copper,P_Loss_Iron,P_Loss_Windage,P_LOSS_EQ)	[P_LOSS_TOT,P_Loss_Copper,P_Loss_Iron,P_Loss_Windage]
Battery_model	(P_mot,P_batt,P_LOSS_TOT,v_cycle,E_in_batt,dT)	[SOC,P_batt_loss,P_batt_int,P_batt]
Efficiency_map_read	()	[W_EFF,T_EFF,ETA_MOT]
Efficiency_map_write	(T_trac,W_rad,ETA_MOT)	[]
Loss_map_write	(W_rad,T_trac,P_LOSS)	[]

Table 7-Functions and stream of variables

Code Architecture:

This section provides the overview of examples that can be simulated using the developed software. For this purpose it's necessary to provide the architecture of the code to provide the understanding of the simulation performed.

- Vehicle longitudinal dynamics

The vehicle model is used to simulate the vehicle longitudinal dynamics.

  - Flowchart:

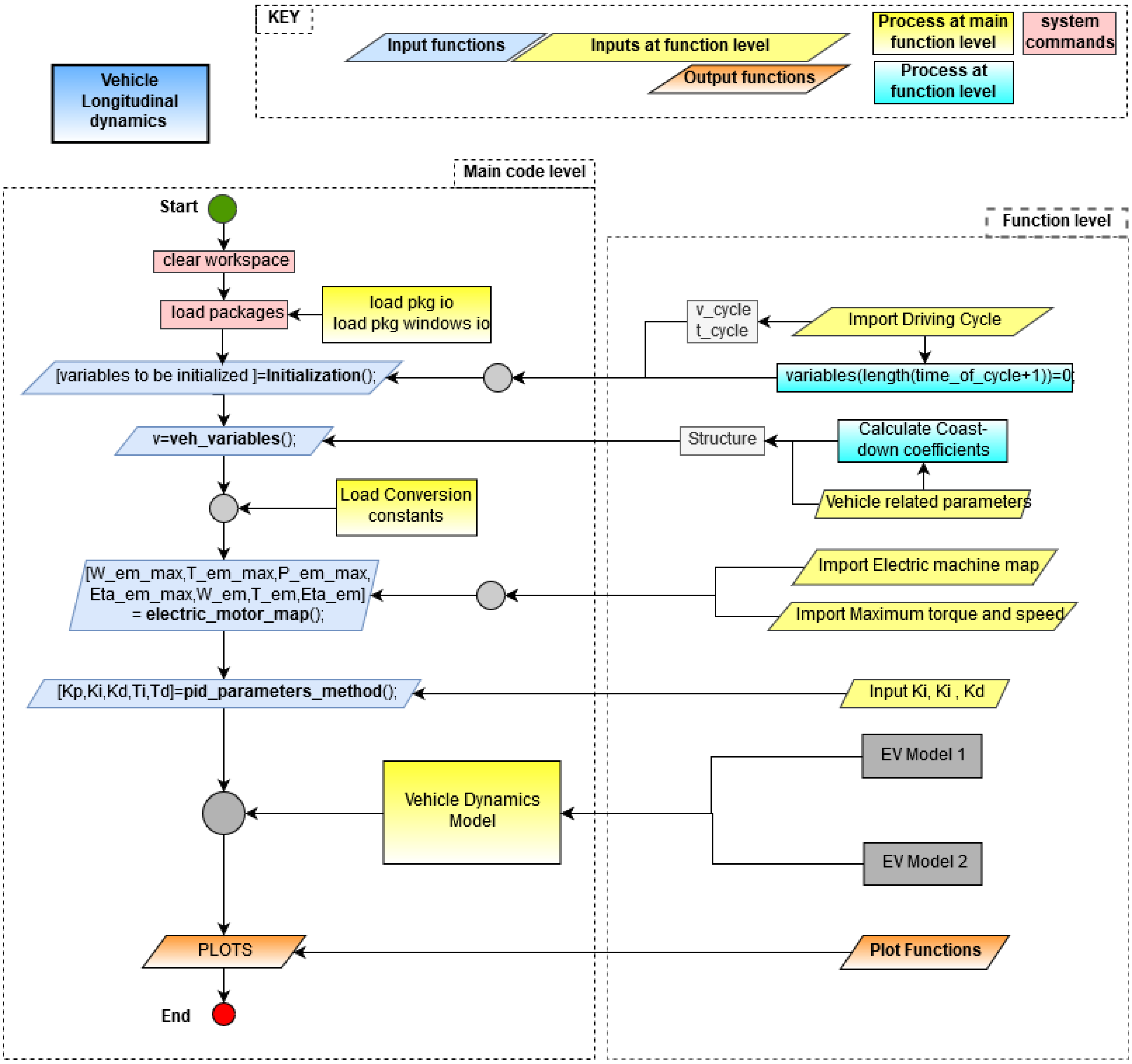


Figure 36- Functional flowchart of vehicle longitudinal dynamics simulation

- Architecture:

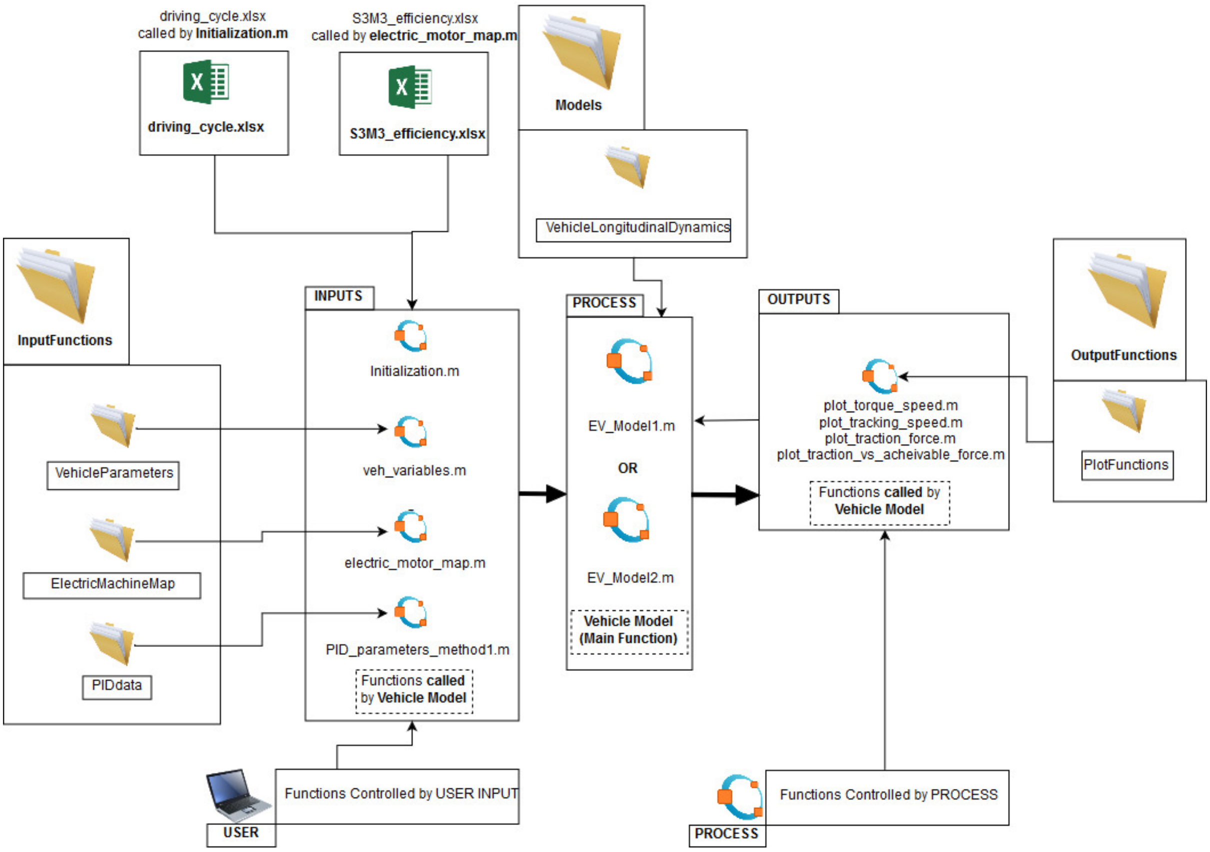


Figure 37-Architecture of vehicle longitudinal dynamics

- Power losses and efficiency

The power losses for an electric machine and it's efficiency can be simulated using a vehicle model.

  - Flowchart:

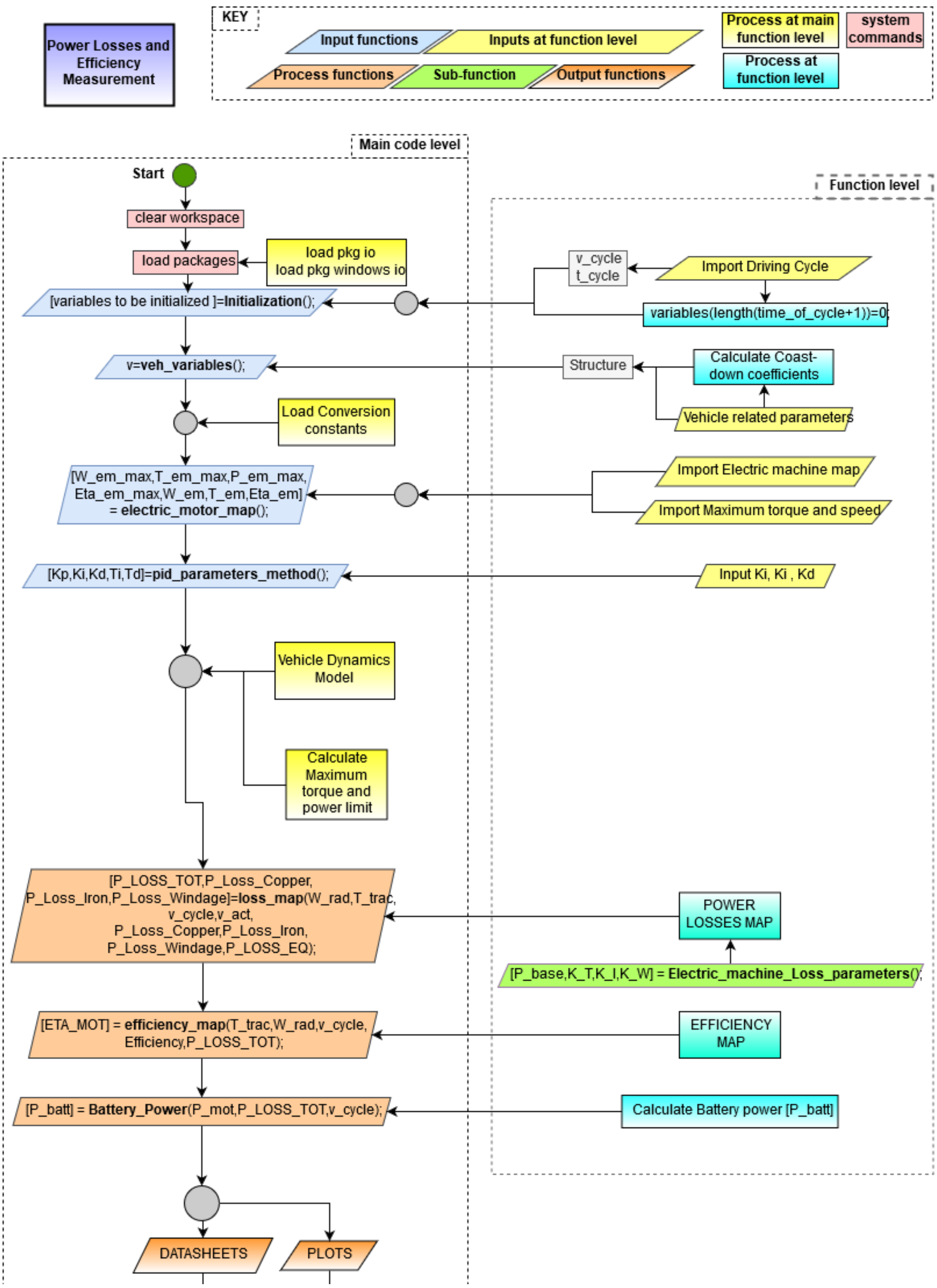


Figure 38-Functional flowchart of power losses and efficiency measurement simulation

- Architecture:

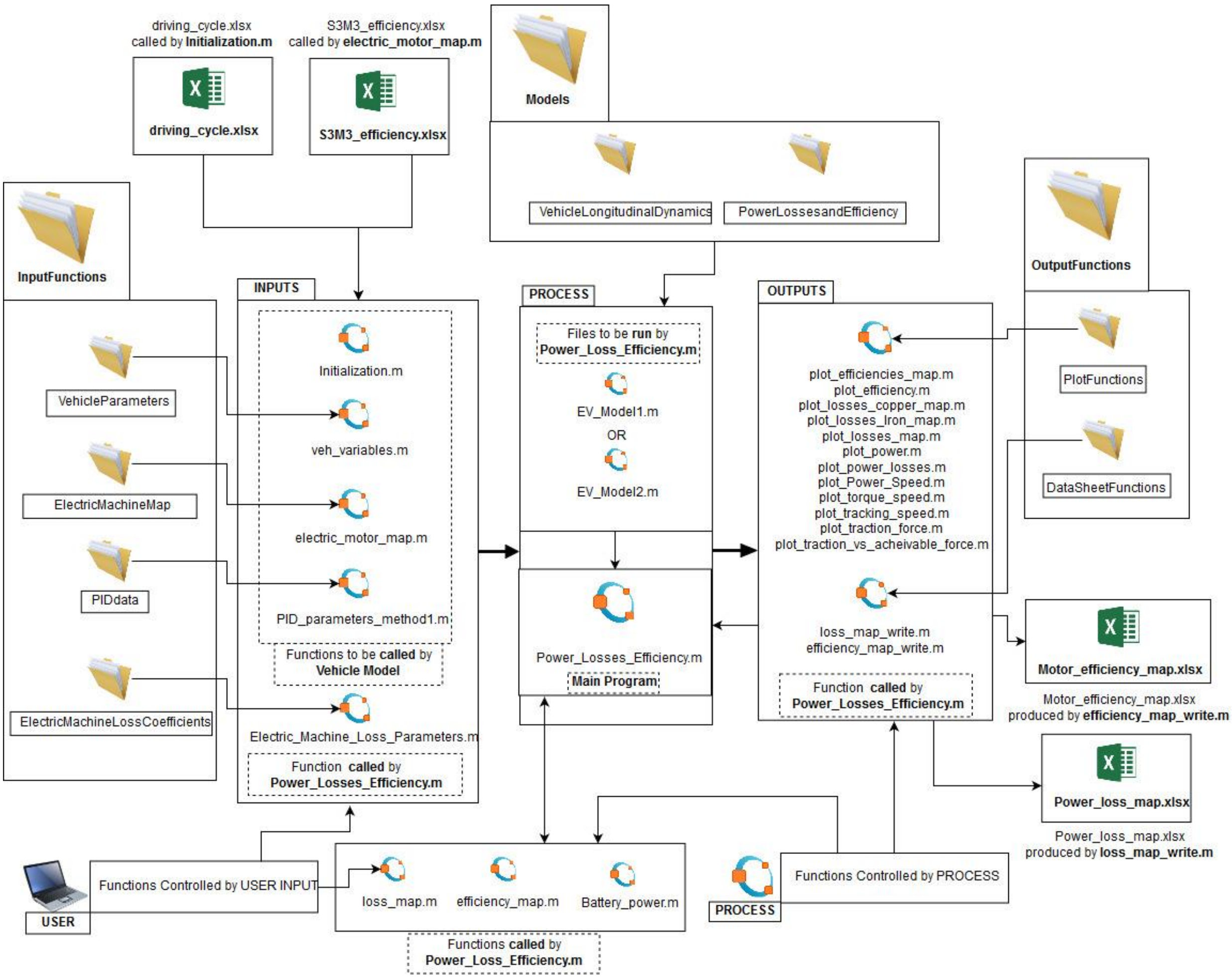


Figure 39-Architecture of power losses and efficiency measurement simulation

End of Document