Gear Selection Strategy Development

Dzheylan S. Myumyun

Master thesis

Institute of Automotive Engineering
Graz University of Technology

Supervisors

DI(FH) Thomas Frühwirth

Assoc.Prof.Dipl.-Ing.Dr.techn. Mario Hirz

2018

Contents

Abstract

Day by day, the car manufacturers try to improve fuel consumption levels to be able to achieve emission regulations. In this thesis, predictive road data information is integrated to the gear selection algorithm to be able to achieve better fuel consumption levels. Similar designs were already used in heavy duty vehicles.

Sensors in the vehicle can observe the coming conditions and controllers can adapt vehicle reactions more efficiently with the predictive road data information. Additionally, since gear selection effects engine operation points, proper selection of the gears in the transmission has a significant impact on the fuel consumption levels. Moreover, with the enhancement of transmisson hardware architecture of the Dedicated Hybrid Transmissions (DHT) area and hybrid vehicles, it will not be possible anymore to use the conventional shift maps in gear selection strategies. That's why a different approach for gear selection strategy with the consideration of inertia forces, transmission losses and engine efficiencies should be developed that is able to consider the predictive road data information.

The proposed gear selection strategy architecture, which is the main goal of this thesis, will be designed for different kinds of powertrain and transmission designs. Since predictive road data information will be also considered during the design of the architecture, this can then also be used in autonomous vehicles.

Engine torque and speed targets will be calculated with respect to predictive vehicle velocity profile and with the consideration of powertrain losses. Gear selection will be performed according to engine efficiency and predictive clutch losses during the gear shift. During selection of the desired gear, predictive engine efficiencies and predictive engine limitations will be also taken into consideration. Designed architecture will be implemented in Simulink for Dual Clutch Transmission (DCT) conventional vehicle and simulated in AVL VSM™. The results will be compared with the currently developed DCT software project. The expected results will show that with the proposed gear selection strategy, it is possible to achieve less fuel consumption levels.

Kurzfassung

Tag für Tag versuchen die Automobilhersteller den Kraftstoffverbrauch zu verbessern, um die neu aufkommenden Emissionsvorschriften einzuhalten. In dieser Arbeit werden prädiktive Streckendateninformationen in den Gangwahlalgorithmus integriert, um bessere Kraftstoffverbrauchswerte zu erreichen. Ähnliche Konzepte finden bereits in Lastkraftfahrzeugen Anwendung.

Durch prädiktive Streckendaten kann die Gangwahl effizienter gestalltet werden, indem die Sensoren im Fahrzeug die aufkommenden Straßenbedingungen erfassen und darauf reagieren. Da auch die Gangwahl den Motorbetriebspunkt beeinflusst, hat die richtige Wahl der Gänge einen wesentlichen Einfluss auf die Kraftstoffverbrauchswerte. Darüber hinaus wird es durch die Erweiterung von Fahrzeugen, mit Dedicated Hybrid Transmissions (DHT) und Hybridisierung, nicht mehr möglich sein, die bis jetzt genutzen Schaltkennfelder in den aktuellen Gangauswahlstrategien zu verwenden. Aus diesem Grund sollte eine neue Gangauswahlstrategie unter Berücksichtigung von Trägheitskräften, Übertragungsverlusten und Motorwirkungsgraden unter Einbeziehung der prädiktiven Streckendaten entwickelt werden.

Durch den vorgeschlagenen Gangwahlalgorithmus sollte für verschiedene Arten von Antriebssträngen und Getrieben eine Funktionsarchitektur entwickelt werden. Da prädiktive Streckendateninformationen auch beim Entwickeln der Funktionsarchitektur berücksichtigt werden, können diese dann auch in autonomen Fahrzeugen verwendet werden.

Motordrehmoment- und -drehzahl werden in Bezug auf das voraussichtliche Fahrzeuggeschwindigkeitsprofil und unter Berücksichtigung von Antriebsstrangverlusten berechnet. Die Gangwahl sollte entsprechend dem Motorwirkungsgrad und den vorhergesagten Kupplungsverlusten während des Schaltvorgangs durchgeführt werden. Bei der Auswahl des gewünschten Ganges werden auch prädiktiv der Motorwirkungsgrad und die Einschränkungen berücksichtigt. Die daraus entworfene Funktionsarchitektur sollte in Simulink für konventionelle Fahrzeuge mit Doppelkupplungsgetriebe (DCT) implementiert und in AVL VSM™ simuliert werden. Die Ergebnisse werden mit dem aktuell laufenden DCT-Softwareprojekt verglichen. Die erwarteten Werte zeigen, dass mit der vorgeschlagenen Gangauswahlstrategie eine Reduktion des Kraftstoffverbrauchs erzielbar ist.

Acknowledgment

First of all, I would like to thank my manager Muammer Yolga for giving me the opportunity working with AVL and supporting me while my supervisor Thomas Frühwirth was not available. Thanks to my professor Mario Hirz for accepting me as a master student. I want to thank also my supervisor Thomas Frühwirth for the reviews and solving my problems in his lack of time.

Finally, special thanks go my family for supporting me every moment in my life and my best friend Emrah Erdoğan for encouraging me from far away in my whole study time in Graz. The last thanks to love of my life Daniel Reiterer as a first reviewer of my thesis, supporting me morally and technically.

1 Introduction

1.1 Description

Mankind always try to predict the future in their social life, in economy, in social sciences, etc. It plays important role in engineering life as well. In these days, engineers try to predict the conditions coming in the future and adapt the systems accordingly. If controllers in the vehicle are able to know driver needs, upcoming road, vehicle velocity in the future, they can calculate the most efficient control strategy to satisfy drivers intent. It would also make the gear selection more effective. The idea of this thesis is coming up from this point.

In the future, the emission regulations will become more strict because of legislation. As an illustrate, the EU6d Emission Regulation implements Real Driving Emissions (RDE) as an additional type approval requirement within the 2017-2020 timeframe. [3]. The RDE legislation requires engines to be clean under all operating conditions. Because of this reason, car manufacturers try to improve emission levels of the production cars by hybrid architecture, efficient engine operations or reducing losses in overall vehicle. Moreover, people are willing to buy environment-friendly vehicles, which exhaust less emissions. To be able to achieve optimal engine efficiency, gear selection strategies play an important role, since gear selection provides engine operation in optimal points.

According to today's developments, gear selection in non manual transmissions is handled by parameterizable maps, so called shift maps, based on the accelerator pedal position and vehicle speed. There are different shift maps for different conditions, like uphill, downhill, sport mode, cold start, etc. There are big calibration efforts not only because of the number of shift maps, but also switching between shift maps and interpolation during switching. Frühwirth [13], interpreted the accelerator pedal position by changing hierarchy, which helps in reducing calibration effort by not using conventional shift strategies so called shift maps. Moreover, with more complex hybrid transmissions such as AVL Future Hybrid 7 and 8 Mode [2], shift maps are not usable anymore, because of ECVT (Electronic Continuously Variable Transmission) driving modes. ECVT driving modes have to be used in conjunction with pure electric driving gears as well as parallel hybrid driving gears, which need to be considered by future gear selection/ mode selection systems.

1.2 Objective

In this thesis, in addition to Frühwirth's [13] development, new approach for the gear selection strategy with predictive road data will be suggested. By the usage of predictive road data, the controller can predict/ calculate future conditions, like uphill, downhill, curve, traffic lights, etc. The presented architecture can be used for hybrid and conventional vehicles, future development can also be included easily.

In addition to the design of the architecture, software model will be implemented for conventional 7 speed DCT (Dual Clutch Transmission) for SUV (Sport Utility Vehicle). The developed software will be tested in the simulation environment and results will be compared with the gear selection strategy of currently developed project. The proposed software architecture, which will be explained in detail in 2.Method Description section, is not implemented completely. Namely, predictive road data is not processed, which is not the scope of this thesis, but the proccessed data will be taken as an input.

2 Method Description

2.1 Current Technology

In the current production cars, predictive road data information is used in adaptive driver assistant systems, for example Audi introduced this in the Q7 model [6]. It slows down based on the speed limits on the road by detecting the traffic signs, as well as when curve or downhill is detected in the coming road. Moreover, it is also used in gear selection algorithms in heavy duty vehicles. Banerjee et.al. [8] showed in the development in ZF Friedrichshafen AG, by including GPS (Global Positioning System) data information in gear selection algorithm, that heavy duty vehicles can react better for road gradient changes. By recognizing road gradient before it comes, controller in the heavy duty vehicle can prepare engine acceleration capacity for driving on the uphill in advanced. There are more research topics about predictive road data usage in gear selection strategies for heavy duty vehicles. To illustrate, Reghunath et.al. [20] developed a gear shift strategy for heavy duty vehicles to be able to overcome to coming hill conditions by using predictive road data information. Additionally, Schuler [21] and Terwen [22] analyzed road slope, vehicle mass, vehicle movement and driver wish for trucks in their PhD studies and used these information during the gear selection.

The challenging part in the proccessing of predictive road data is calculating the road trajectories out of processed GPS data. Road trajectory calculation depends on not only GPS data, it also depends on psychological conditions of the driver, such as driver thinking, driver behaviors in the specific traffic situations, driving style, like aggressive, or relax. In the literature, there are several researches made to be able to calculate road trajectories. Panahandeh [19] suggested route and destination prediction by analyzing the history of driving for individuals. Lorenzo et.al. [10] used the sensors provided information about the driver's mood, attention span, as well as interaction with the car. For example, in addition to the navigation data, acceleration and braking, climate control and seat position are used during the prediction of trajectories. Tran et.al. [23] suggested a prediction of a driver behavior by foot gesturing, namely they analyzed a foot movement by using vision-based foot behavior analysis. Wang et.al. [24] analyzed driver distraction based on brain activity patterns. Mabuchi et.al. [17] tried to predict drivers stop or go at yellow traffic signal from vehicle behavior. Keller et.al. [16] studied the pedestrian movements to be able to predict moving behaviors.

As can be understood from all these papers, there are so many researches going on about proccessing the predictive road data, like studies of Banerjee et.al. [8], Reghunath [20]. This information should be also included in gear selection strategies, since there is still room for improvement related to fuel efficiency/emissions but also driver satisfaction.

If autonomous vehicles are brought into consideration, which might even communicate with their environment, this will open up even more prediction strategies. For example, cars might be able to plan their stay at electric charging stations or battery exchanges. Therefore, these information can be also used in their driving and gear selection strategy.

Currently, gear selection in vehicles with automatic transmissions are calculated with a conventional shift maps. Figure 2.1 shows a state of the art shift map with upshift and downshift lines with constraints of vehicle velocity and accelerator pedal. When solid lines are crossed, upshift is performed. In the same way, when the dashed lines are crossed, downshift is performed, as it is shown in the figure. There is a hysteresis between upshift and downshift. There are various parameterizable maps possible for more efficient drive and required drivability, which brings too much calibration effort not only because of the several number of shift maps for the different situations like sport, uphill, downhill, cold start,etc. but also because of switching between them.

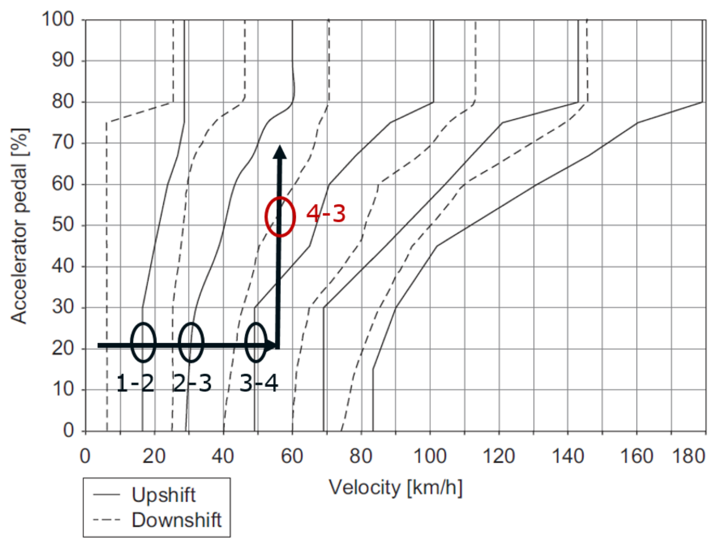


Figure 2.1: Conventional shift map with upshift and downshift lines [11]

In literature, there are several researches for improving these conventional shift maps. As Fofana et.al. [12] stated in their research, a gear shift map is designed in two stages. First, initial shift point is obtained by using analytical methods. Calibration is then used to refine the initial shift points and generate a set of gear shift maps based on expected driving conditions and vehicle uses. They aimed to automate this process by using dynamic programming to obtain first shift point based on driveability, then applied a statistical analysis to define the final shift point. Ngo et.al. [18] developed a gear shift map design methodology, which takes advantage of the optimal-based gear shift strategy and the statistical theory to construct an optimized gear shift map, which is consistent, robust and real time implementable. Ha et.al. [14] designed a flexible gear shift pattern instead of static gear shift maps. They used normalized radial basis function neural network for generating flexible shift maps to satisfy the driver demands including comfort and fuel consumption.

Vehicle developments progress through hybrid architectures, where electric motor is used in addition to the engine. DHTs (Dedicated Hybrid Transmission) are one type of these hybrid architectures. Figure 2.2 shows a DHT type gearbox, where electric motor is mounted inside transmission. Engine can be separated with C0 clutch completely from the transmission and pure electric drive with two different discrete gear ratios are possible. Additionally, ECVT (Electronic Continuously Variable Transmission) modes provide variable gear ratio instead of fixed gear ratio, with planetary gear structure. With the variable gear ratio flexibility, engine can be operated at its most efficient load point to either propel the car or charge the battery. If engine is not needed, it is also possible to shut down the engine and drive only with electric motor. One of the drawback of state of the art shift maps is usability in DHTs. DHTs open up new areas for optimization also considering battery state with ECVT modes. Velocity and accelerator pedal are not the only constraints in gear selection in DHTs. Torque split between engine and electric motor, as well as battery charging level should be also considered. Different gears can be selected for the same velocity and accelerator pedal position, since torque splits may differ in every time. With variable gear ratio in ECVT modes, it is even harder to decide correct gear ratio just by considering these two constraints.

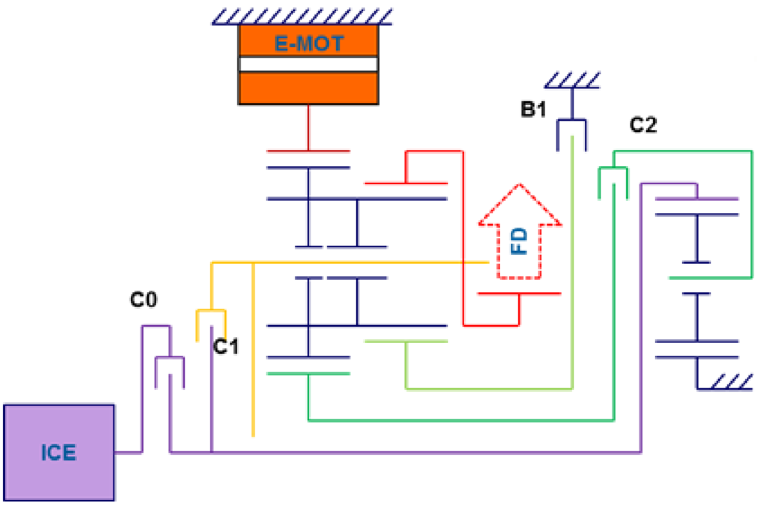


Figure 2.2: DHT type Gearbox [2]

2.2 Ideas of Concept

In this study, new architecture will be suggested for the gear selection strategy. This architecture includes the usage of predictive road data for the passenger cars. The architecture is applicable for both hybrid and conventional passenger cars. Gear selection will be calculated based on the vehicle velocity profile. Vehicle velocity profile calculation highly depends on predictive road data information. Additionally, dynamic driver behavior and road changes will be analyzed and considered in the gear selection like fast off, heavy brake, NVH, ABS/ESP activations. This part of the architecture is taken from thesis of Frühwirth [13] and extended by the predictive road data information.

The perfect conditions for these calculations would be provided by autonomous vehicles, where the route is calculated in the beginning of the drive. Prediction of the driver behaviors, road planning algorithms, learning algorithms are still in development as can be seen from the research papers, which are provided in the chapter above. In the current technology, it is possible to read the road information for the coming road like uphill, downhill and curve, which are important factors in gear selections. Road data information is collected and processed by some companies like Continental [1]. eHorizon project of Continental integrates topographical and digital map data with sensor data, namely GPS receivers for predictive control of vehicle systems. Future events, such as the uphill inclination after the next corner, are exploited at an early stage in order to optimize the vehicle's response. eHorizon interprets map and sensor data, which can be adapted to the vehicle control easily. Additionally, this system provides weather-, vehicle and time dependent speed instructions, which are not identified by the camera systems.

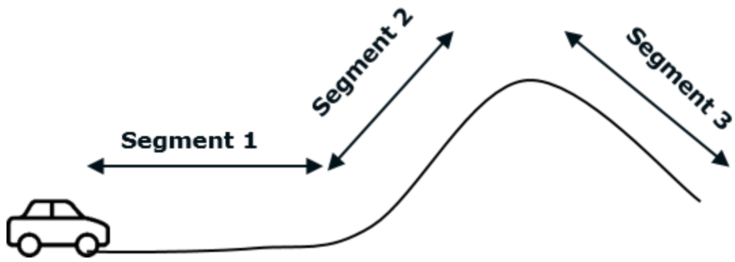


Figure 2.3: A schematic of a segmentational splitting of predictive data

These information is combined with the map and sensor data, then proccessed data is provided as segments, see Figure 2.3. Segment is defined as the part of the road, which has same characteristics, such as weather, road gradient, velocity limits, etc. It supplies also dynamic data via cloud, like traffic information or changes on the route.

The other aim of this thesis is designing a new software for the gear selection by not using conventional shift maps. There is a thesis about a different gear selection strategy without using of conventional shift maps, as it was mentioned before [13]. In addition to this, aim of this thesis is to improve this approach by considering predictive road data information, transmission efficiency and inertia losses. Therefore, gear selection is done with physical model by considering efficiencies and losses instead of velocity and acceleration pedal dependency.

2.3 Architecture

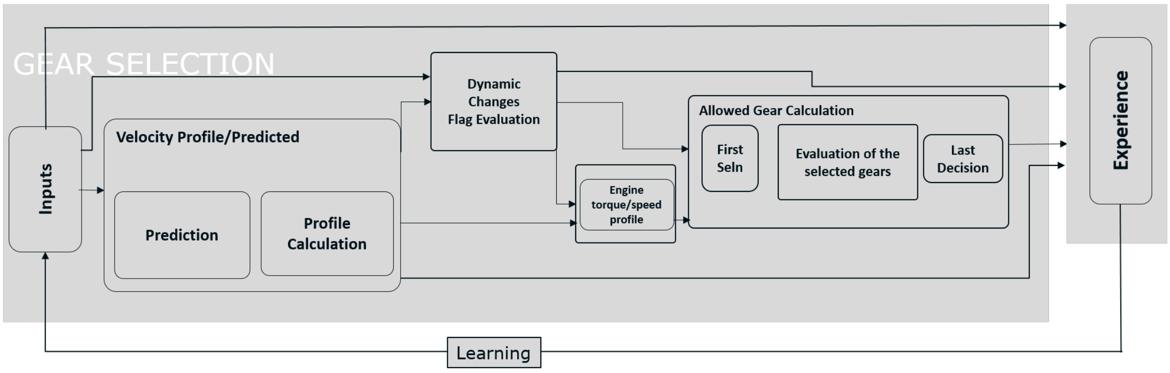


Figure 2.4: The architecture of the software

Self proposed architecture can be seen in Figure 2.4. Figure shows relevant subsystems of Gear Selection, which will be described in detail in the coming sections. Signal flows are represented by the arrows. The subsystems are placed according to calculation order, where Inputs is the first executed subsystem, which is followed by Vehicle velocity profile/predicted and so on. In this figure, only Gear Selection part is implemented, where Experience and Learning systems will not be treated in this thesis but might be an improvement suggestion. As discussed in chapter C. Simulation and Testing Environment, pure gear selection part is already showing acceptable results.

In this thesis, Gear Selection itself is called as system, whereas the other blocks are subsystems of Gear Selection system or subsystems of other systems. Gear Selection System consists of the following subsystems, which will be discussed in the following subsections in detail.

- Inputs
- Velocity Profile/Predicted
- Dynamic Changes Evaluation
- Engine Torque/Speed Profile
- Allowed Gear Calculation
- Experience

2.3.1 Inputs

In this subsystem of Gear Selection, required input data will be taken from ECU (Engine Control Unit), Hardware, ESP (Electronic Stability Program), HCU (Hybrid Control Unit) and HMI (Human Machine Interface). These signals will be processed to be used inside the gear selection software.

Inputs from HCU:

HCU is the main control component of hybrid vehicles. According to the input signals such as accelerator pedal position, requested gear and brake pedal position, the HCU can calculate the engine, electric motor driving and generation torque.

If this system is applied for the hybrid vehicles, HCU should give some efficiency calculations to the TCU (Transmission Control Unit). TCU is the control unit for the automatic transmission, which uses sensors from the vehicle as well as data provided by ECU (Engine Control Unit) to calculate how and when to change gears in the vehicle for optimum performance, fuel economy and shift quality. Gear shifting, hydraulic and clutch controls are also performed by TCU. When it is a hybrid vehicle, HCU is the master of engine control. HCU decides the possible hybrid mode and accordingly engine speed and torque requests are sent to the ECU. Afterwards ECU controls the engine to be able to fulfill the HCU requests. Because of this reason, HCU is able to calculate overall efficiency more effectively, since HCU can calculate which electric motor/engine speed and torque are possible according to possible hybrid mode. Out of this information, HCU can select the optimum HCU driving mode such as boosting or generation and provide efficiencies accordingly to TCU. By designing this architecture, it is assumed that HCU will influence overall vehicle efficiency according to hybrid mode and engine requests. Additionally, in hybrid vehicles, torque split is an important factor for deciding the desired gear since it effects the engine and electric motor efficiencies and in parallel fuel consumptions. To be able to get desired output shaft request, torque source can be partially engine, partially electric motor or only electric motor or only engine. Proportional torque between engine and electric motor is defined as torque split. SOC (State of Charge) level of the battery and hybrid modes are factors that torque split depends on, that's why torque splits should be calculated in HCU. With the predictive road data information, it is possible to simulate recuperation and drive capabilites for future conditions and battery management system can be planned more efficiently. For example, driver goes in flat road, battery needs to be charged and downhill road is coming. There is no need to find another strategy for charging the battery but braking during the downhill drive can easily be used for charging the battery.

Hardware:

Following signals will be taken from hardware interface from the sensors to be able to calculate desired gear.

- Output shaft speed is read from sensor as a unit of rad/s.
- Output shaft direction is read from sensor and sensor delivers the information about forward or backward turning of the output shaft.

HMI:

HMI can be defined as interface between controller and a user. It provides a graphic based visualization of a controllers. User can influence the controllers in the vehicle with HMI or vehicle controllers can also communicate with the user by showing malfunctioning in the car or delivering some informations such as fuel consumption, vehicle velocity, current gear, etc. Following input signals are delivered by driver via HMI.

- Gear lever position is influenced by a driver and sent through the HMI. Typical gear lever positions are Park, Neutral, Drive, Manual and Sport. When Drive is engaged, possible gear is defined by the gear selection system in the TCU. When Manual is selected, desired gear is defined by the user by Tip Up/Down buttons and gear shifting is performed by TCU. Sport mode is like Drive mode but in this mode, sportive drive can be performed, namely acceleration capacity is much more according to Drive mode.
- Vehicle mode can be also influenced by driver via HMI, where Comfort, Sport, Economy, etc. modes can be selected.

Prediction related:

Static and dynamic route data is delivered via this module. Following inputs are used in gear selection strategy.

- Processed static route data, namely segment matrix is delivered via this module.
- Processed dynamic data, like traffic information, constraction on the way, etc. are delivered via this module.

ESP controller:

ESP controller provides vehicle stability incase of loss of steering control and applies brakes in the wheels individually for bringing the driver the steering that driver intents.

In case of ESP is activated, following signals are taken as an input and processed inside the gear selection.

- In loss of steering detection, ABS/ESP Flags are raised.
- When ESP is activated, wheel speeds are controlled by ESP system.
- Wheel dynamic radius is also provided by ESP system.
- When ESP is activated, it applies brakes to the wheels individually, so that brake pedal pressure is also provided by this system.

ECU:

ECU controls the actuators inside the combustion engine and optimizes the engine performance. Following signals are provided via ECU to the TCU.

- Acceleration pedal position in percentage
- Engine torque in Nm
- Engine idle speed in rpm
- Engine speed in rpm

Moreover, vehicle speed will be calculated from output shaft speed. Vehicle acceleration will be calculated by taking the derivative of the vehicle speed. Engine synchronized speed for each gear will be calculated by multiplication of gear ratio with output shaft speed. Engine torque limitations will be calculated based on parameterizable maps with respect to engine synchronized speed. Parameterizable maps used for engine limitations are static maps, where influencing factors like ambient temperature or engine temperature are not considered. Actual gear will be calculated in other part of the TCU software based on rail states or clutch conditions.

Experience:

This system should be implemented separately from the gear selection strategy. This is a different system for observing the behavior of vehicle according to output of the gear selection strategy. The idea of Experience system is to generate data and behavior of the vehicle according to these data can be monitored easily. Out of this information, parameter fitting can be implemented by using the input information against the actual outputs. In the end, weight factor should be generated and given as a feedback to the Input subsystem of Gear Selection. Afterwards, this information will be evaluated under 'Evaluation of the selected gears' subsystem of 'Allowed Gear Calculation'. By the information coming from Experience system, gear selection strategy can be adapted and improved overtime.

2.3.2 Velocity Profile/Predicted

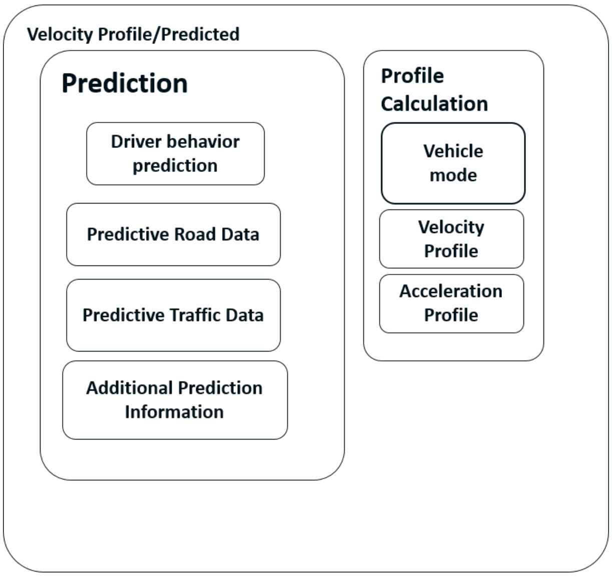


Figure 2.5: The subsystem of Velocity Profile/Predicted Calculation

Velocity and acceleration profile prediction will be calculated in this subsystem of Gear Selection, which can be seen in the Figure 2.5. Subsystems are shown in the figure in the calculation orders, where Prediction will be processed first, then Profile is calculated. Implementation of this subsystem is not the scope of this thesis. During the testing of the software, predicted velocity profile will be given as an input to the output shaft torque profile calculation.

Driver behavior prediction subsystem is used for predicting the driver needs. There are different kind of researches going on. Some of them was also mentioned in the previous sections. For example, Lorenzo et.al. [10] tried to predict drivers mood, Tran et.al. [23] analyzed foot gesturing or Wang et.al. [24] analyzed brain activity pattern. All those information can be used for predicting the driver needs and adapting vehicle velocity profile accordingly.

Predictive road data subsystem of Prediction analyze the static data like uphill, downhill, curve, velocit limits, etc. Route information comes as segments, which was explained in the Figure 2.3.

Predictive traffic data subsystem of Prediction contains the traffic information, which changes dynamically. Such as car behavior in front, pedestrian, traffic lights, construction area. Predictive traffic data can overwrite the predictive road data by generating emergency changes flag and distance to the object, such as sudden brake in front of the car. It is possible to add any predicted information coming in the future here.

During the velocity profile calculation, vehicle mode has a big effect, which is implemented as a subsystem of Profile calculation. It contains the driver mode request information like sport, comfort, economy, etc. Vehicle mode information comes from HMI as driver choice and is combined with driver behavior prediction from the Prediction subsystem before. According to the vehicle mode, different parameterizable parameters should be assigned to be used in the coming subsystems during vehicle velocity and acceleration calculations.

Calculating the predictive velocity profile requires quite a lot calculation works, which will not be done in this thesis. The perfect condition for this calculation might be possible in an autonomous car such that the road is known in advance and velocity profile is calculated in a most efficient way. The most challenging part here is predicting the driver behavior. It is like to procast the drivers wishes: whether driver prefers slow or aggressive drive, whether driver prefers press accelerator pedal more to be able to overcome traffic lights, when it is blinking or just stop. What is the behavior of the driver? Is it more impulsive or relaxed? Does the driver like to go high speeds or low speeds generally? They are all constraints in the prediction. It is possible to use learning algorithm to be able to analyze and adapt software according to driver behavior over time, introduced e.g. in the research of Panahandeh [19].

2.3.3 Dynamic Changes Evaluation

Dynamic changes during the drive, which effect gear selection, will be evaluated and flags will be calculated out of this information to be used afterwards during Allowed Gear Calculation subsystem of Gear Selection, which can be seen in the Figure 2.6. Subsystems are shown in the calculation order, where Emergency changes will be processed first, then it is further evaluated under Heavy Brake, Fast Off and Kickdown subsystems. The other subsystems are calculated in paralel and send their own output information seperately from the Dynamic changes evaluation to the other subsystems in the Gear Selection system.

- Heavy Brake: Heavy brake is a state, which pressing the brake pedal suddenly and hard. Brake pedal pressure will be checked against a parameterizable map with vehicle velocity input. If it is greater than a parameter, Heavy brake flag will be activated. Heavy brake flag will be reseted after a parameterizable time, if Heavy Brake flag is disabled.

- Fast Off: Fast off is a state, which removing foot from the acceleration pedal. Accelerator pedal position will be checked with a parameterizable map. If it is released too fast, Fast off flag will be activated. Fast off flag will be reseted after a parameterizable time, if Fast off flag is disabled.

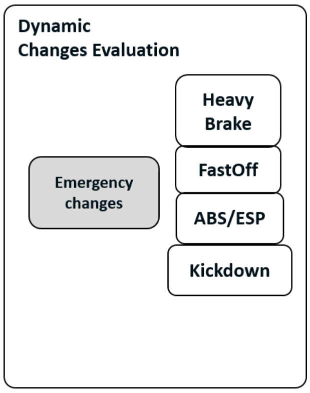


Figure 2.6: The Subsystem of Dynamic Changes Evaluation.

- ABS/ESP: If ABS/ESP is triggered, which comes from ABS/ESP directly, this flag will be activated. This flag information may change depending on ESP system. ESP can completely disable gear shift or request upshift for reducing the output shaft speed or request clutch separation. Required action will be evaluated in the coming subsystem, Evaluation of the selected gears.

- Kickdown: Kickdown is a state, which pressing the accelerator pedal suddenly and hard. If accelerator pedal position gradient is higher than a parameterizable maps value according to accelerator pedal position, Kickdown flag will be activated. Kickdown flag will be reseted after a parameterizable time, if Kickdown flag is disabled. For the hybrid vehicles, this information will come from HCU directly.

- Emergency Changes: This information comes from 'Prediction Traffic Data' as a flag and a distance to the object. Object can be car, which makes emergency brake or a pedestrian, who jumps to the road unexpectedly or traffic lights that turn on red. Following actions can be taken during this condition:

  - For stopping before crashing to the object, heavy brake can be applied. In that case, Heavy Brake flag will be activated.

  - When vehicle speed is low and distance to the object is not that short, just removing foot from accelerator pedal can be enough. In that case, Fast off action can be enough for overcoming the target and Fast off flag will be activated.

  - When front car applies brake and side road is available, take over can be performed. In this time, Kickdown flag will be activated.

  - If vehicle velocity is low, normal brake can be enough, in this case none of the flag will be activated and normal drive will be performed.

2.3.4 Engine Torque/Speed Profile

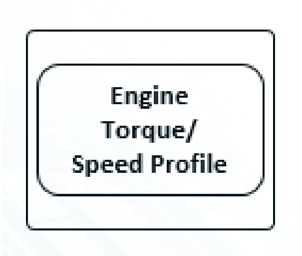


Figure 2.7: The Subsystem of Engine Torque/Speed Profile

Predictive vehicle velocity profile will be calculated out of predictive road data information, which in this thesis will be given as an input to the simulation but not calculated. By the vehicle velocity profile, predictive engine torque and speed will be calculated under subsystem Engine Torque/Speed Profile of Gear Selection, see Figure 2.7, so that it can be used during decision of the most appropriate gear. For this calculation, the engine power equation is used at output shaft level, where P is the Power, F is the general Force and v is the speed, which is shown by Equation 2.1.

$$P = F * v \tag{2.1}$$

It is a known fact that forces acting at the vehicle for constant velocity is shown in the Figure 2.8. The figure shows the gravity force related force F_G, air resistance force F_L, rolling resistance force F_R and total equilibrium force to these forces F_k.

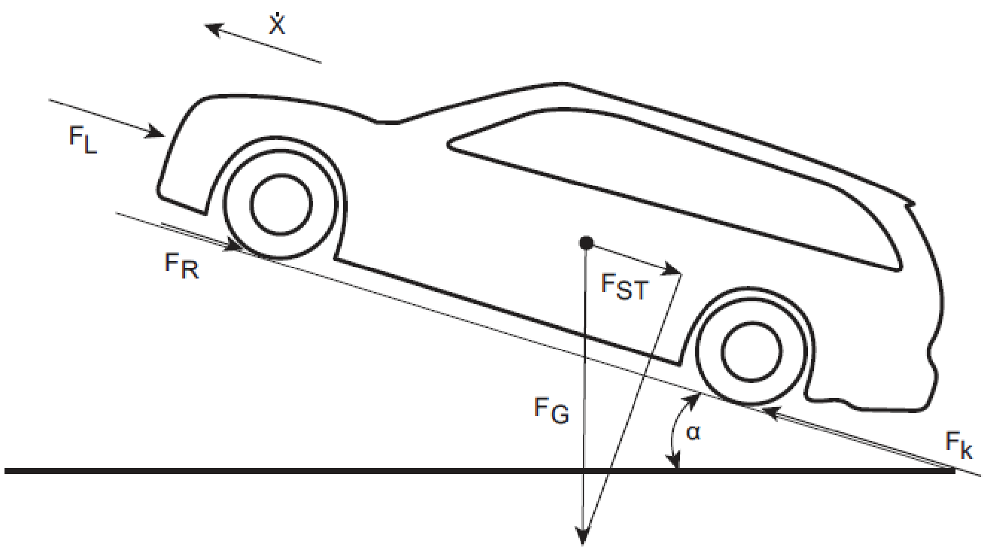


Figure 2.8: Forces on vehicle model in constant velocity [11]

The rolling resitance can be estimated by the Friction force F_R with the mass m, acceleration of gravity constant g, the friction coefficent f_R and the angle α, which can be seen in Equation 2.2 [11]:

$$F_R = m * g * f_R * cos\alpha \tag{2.2}$$

The air resistance against the vehicle will be concluded by air resistance force F_L with air drag force c_D, air density ρ_A, cross sectional area of the vehicle A_F and vehicle velocity v, which can be seen in Equation 2.3 [11]:

$$F_L = \frac{1}{2} * c_D * \rho_A * A_F * v * |v| \tag{2.3}$$

Climbing resistance force F_{ST} where road gradient will be inclueded in the calculations is calculated by Equation 2.4 [11]:

$$F_{ST} = m * g * sin\alpha \tag{2.4}$$

Forces F_k to keep the vehicle in the constant velocity calculation can be seen in Equation 2.5 [11]:

$$F_k = F_R + F_L + F_{ST} \tag{2.5}$$

For an overall forces F on the vehicle, engine acceleration effect will be also included with vehicle mass m_{veh}, vehicle acceleration a, which is shown in Equation 2.6.

$$F = m_{veh} * a + F_k \tag{2.6}$$

There is inertia losses in the vehicle, which will be taken into consideration during engine torque profile calculation from engine power in output shaft level. In this thesis FWD (Front Wheel Drive) vehicle will be used during implementation of the software, whose inertias can be seen in Figure 2.9.

Where I_E refers to inertia of engine, I_G refers to inertia of transmission, i_A refers to axle drive ratio, i_G refers to discrete gear ratio, I_{AF} refers to inertia of front axles, I_{AR} refers to inertia of rear axles.

Total inertia in front axles I_F will be calculated with the Equation 2.7 [15]:

$$I_F = I_{AF} + {i^2}_A * (I_G + {i^2}_G * I_E) \tag{2.7}$$

Total inertia in rear axles I_R will be calculated with the Equation 2.8 [15]:

$$I_R = I_{AR} \tag{2.8}$$

In the end, overall torque loss M_{Jout} because of inertias with acceleration a and wheel

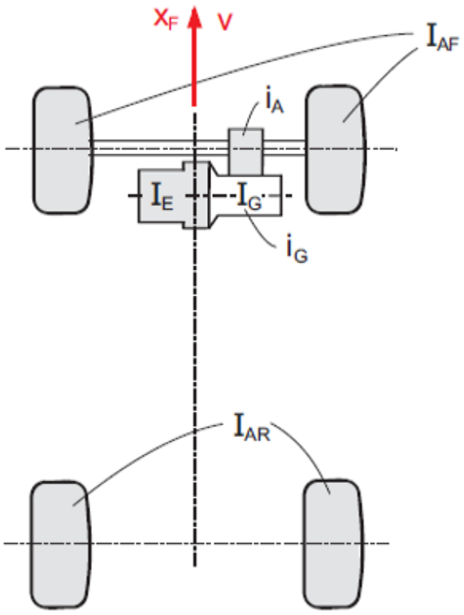


Figure 2.9: Moments of Inertias for a FWD Vehicle [15]

dynamic radius r will be calculated with the Equation 2.9 [15]:

$$M_{Jout} = a * \frac{I_F + I_R}{r^2} \tag{2.9}$$

Power loss because of inertias will be calculated with the Equation 2.10:

$$P_{Jout} = M_{Jout} * \omega_{Eng} \tag{2.10}$$

Engine speed in gear ω_{Eng} is a function of vehicle velocity v, dynamic tire radius r and total gear ratio i, as can be seen in Equation 2.11:

$$\omega_{Eng} = \frac{v}{r} * i; \tag{2.11}$$

Transmission losses was measured in the testbed under different conditions with the following steady state conditions: torque and load, rotational speed or velocity, temperature, time or duration of operation, engaged gear. Depending on the variation of load and speed, the propotional amounts of losses vary. While in low speed and torque situations the base losses are predominant, the torque and speed dependent losses are significant for the description of the efficiency at high speed or load condition. By permutation of the testing parameters, an efficiency map was formed with respect to input speed and input torque in percentage of the maximum input torque that describes losses over all conditions.

All in all, engine torque target for each gear M_{Eng} will be calculated by engine power P, engine speed in gear ω_{Eng}, transmission efficiency η and power loss because of inertias P_{Jout} with the following Equation: 2.12

$$M_{Eng} = \frac{P + P_{Jout}}{\omega_{Eng} * \eta} \tag{2.12}$$

2.3.5 Allowed Gear Calculation

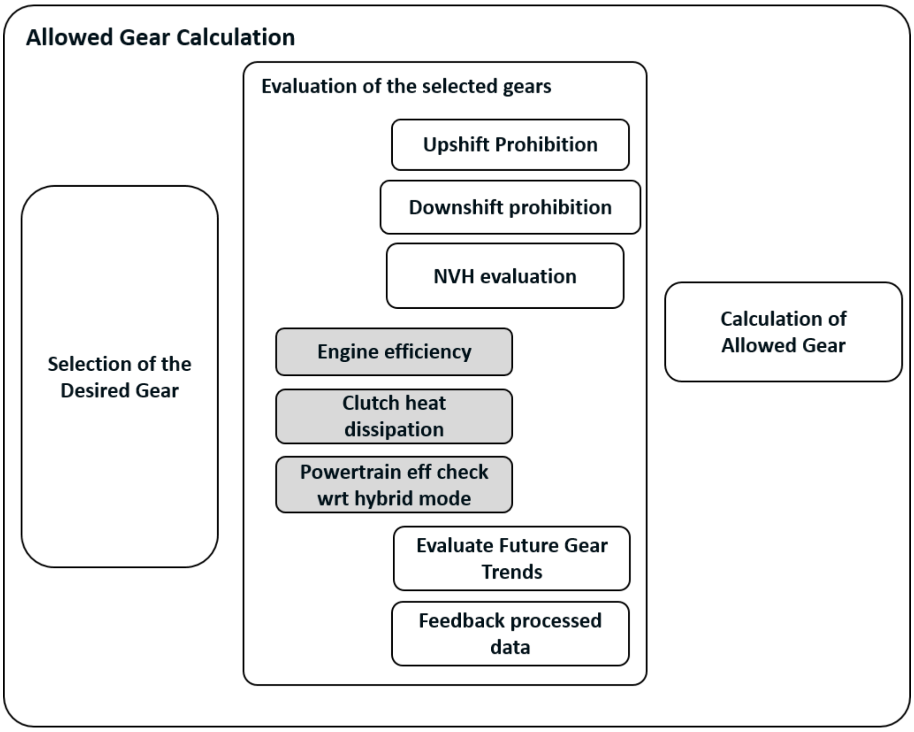


Figure 2.10: The Subsystem of Allowed Gear Calculation

This is the final step of the Gear Selection strategy, which is shown in Figure 2.10. The figure shows subsystems in the calculation order where Selection of the desired gear will be executed first, then Evaluation of the selected gears and in the end Calculation of allowed gear. The subsystem under Evaluation of the selected gears are calculated in paralel. The subsystems, which are colored grey are processed further in Evaluation future gear trends. These grey subsystems do not send any output to from Evaluation of the selected gears to Calculation of allowed gear. That's the reason why these subsystems are placed back from the other subsystems under Evaluation of the selected gears.

First of all, desired gear/s will be selected according to engine allowances. Afterwards, selected gears will be evaluated with respect to engine and powertrain efficiencies. In this stage clutch heat dissipation during shift will be also considered. Moreover, future gear trends will be evaluated with respect to efficiencies. Dynamic changes flags will be evaluated under Upshift, Downshift prohibition subsystems of Evaluation of the selected gears. All in all evaluations will be finalized and the most desired gear will be selected in Calculation of Allowed Gear subsystem of Allowed gear calculation.

Selection of the Desired Gear:

In this subsystem of Allowed gear calculation, desired gear will be selected with respect to engine and electric motor limitations. For the current thesis, where a conventional car is simulated, following strategy will be implemented:

Engine speed synchronization for each gear calculation will be taken from 'Input' subsystem. Engine target torque and speed will be taken from 'Engine Torque/Speed Profile' subsystem of Gear Selection. Synchronized engine speeds will be checked, whether they are in engine parameterized speed limitations. If they are in the limits, they will be assigned as allowed gear/s. Engine torque limits will be calculated in 'Inputs' subsystem with respect to engine synchronized speed, which will be used for checking whether engine torque targets are in these limits. If engine target torques are within these limits, it will be assigned as allowed gear/s. In the end, selected desired gear will be the common gear/s from these limitation checks. There can be a condition where none of the gear is sufficient for the engine target torque. In this case, the gear which gives the highest engine torque within the engine speed limits will be selected as an allowed gear.

For the hybrid drive, where both engine and electric motor is running, both engine and electric motor limitations should be considered. If it is hybrid architecture, torque splits will be also an important variant.

For the hybrid drive, when only electric motor is running, logic can be updated according to both electric motor limitations and hydraulic needs. Logic would be changed according to hydraulic architecture. For example, if mechanical pump is positioned in the input shaft and it turns with the input shaft speed, following conditions can be considered:

- If flow request increases above the actual flow, minimum input shaft speed will not be interfered.
- If pressure request increases above available pressure, minimum input shaft speed will not be interfered.
- Otherwise; current gear will be kept during electric drive.

Evaluation of the selected gears:

Selected desired gear under subsystem 'Selection of the Desired Gear' will be evaluated under this subsystem.

Upshift prohibition: In this subsystem of Evaluation of the selected gears, dynamic flags which were calculated under 'Dynamic Changes Evaluation' will be evaluated with respect to upshift prohibition.

- If ABS/ESP flag is activated, necessary action will change according to ESP signal content. Some ESP systems inhibit gear shifts, whereas some ESP systems might request upshift to reduce output torque.
- If Fast off flag is activated, action will be taken according to synchronized engine speed. If synchronized engine speed is exceeded, gears until the last desired one plus one upper gear will be allowed. So that higher engine speed will be prevented. Otherwise; gears until the last desired one will be allowed.
- If Kickdown flag is activated, gears higher than the last desired gear and last desired gear will be disabled, to be able to force vehicle to downshift. The reason behind, when downshift is performed, there will be more engine speed and requested acceleration performance will be achieved.
- If Heavy brake flag is activated, new engine speed for each gear to be able to get during brake actions should be calculated. If this new engine speed for each gear is greater than a synchronized engine speed for each gear, then relevant gear will be disabled. The idea behind is triggering the shift earlier, so that having higher acceleration capabilites afterwards.

Additionally, current gear will be evaluated, according to future engine speed limitations. Current selected gear/s which were calculated under 'Selection of the Desired Gear' will be evaluated according to predicted engine speed. UpShift duration is estimated according to current engine speed. Current selected gears will be checked with respect to future maximum engine speed limitations. If after a parameterizable shifting duration, this gear is not under maximum engine speed limit, then it should be disabled.

Downshift prohibition: current gear will be evaluated according to future engine speed limitations. Current desired gear/s which were calculated under 'Selection of the Desired Gear' will be evaluated according to predicted engine speed. Downshift duration is estimated according to current engine speed. Current selected gears will be checked with respect to future minimum engine speed limitations. If after a parameterizable shifting duration, this gear is not above minimum engine speed limit, then it should be disabled.

NVH evaluation: NVH behavior will be considered in gear selection with a parameterizable map with engine speed and mean pressure according to engine torque target at engine level constraints. Output of this maps will indicate good or bad NVH behavior. These maps can be gear dependent. There can be specific gears, in specific conditions that cause specific bad noises in specific engine speed/torque areas. Observing of NVH will be more challenging for DHTs as powerflow through the transmission is continuously variable. In this case, torque splits and ECVT gears should be also constraints for NVH evaluation.

Engine efficiency: Engine efficiency for interfered, as well as for predicted engine efficiency values will be evaluated by a map with respect to average mean pressure and engine speed. Output of this map will be a fuel consumption in g/kWh.

Results of this map will be processed further in Evaluation of future gear trends.

Clutch heat dissipation [11]: The temperature increase in the clutch is determined by the power loss. The power loss in the clutch is a result of clutch torque and speed differences between the clutch sides. The greatest power loss occurs in the beginning of a shift and drops to zero toward the end of the synchronization phase. Clutch torque in the engine side is applied by the engine, which can be reduced by engine torque intervention control. The other side of the torque decelerates because of the rotating masses. At constant torque during shift, the heat loss is determined by a work loss. Figure 2.11 shows the curve of the dissipated power (P_V) for a powershift and the associated integral that equals the energy converted into heat namely work loss (W_V).

As can be understood from the upper explanations, heat dissipation in clutch occurs during the shifting phase, namely where clutch slip occurs. In slipping phase, friction between the clutches results in heat dissipation, respectively power loss. This value highly depends on the torque handover functionality in the transmission software, where shifting from one gear to an other gear is performed. This makes clutch heat dissipation, in other words, power losses estimation, a complicated task. In this software, parameterizable map will be constructed with respect to clutch slip, clutch torque and shift time. Power losses during shifting is estimated as a power unit of kW. In the future, this functionality can be improved with the interface connection with torque handover, such as torque handover can deliver an estimated time during speed and torque phase and these maps can be constructed dynamically.

Power losses during shifting will be processed further in Evaluation of future gear trends.

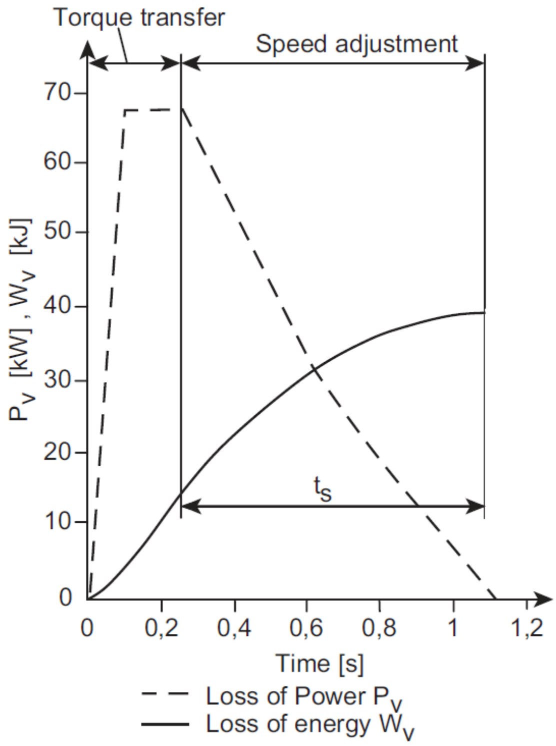


Figure 2.11: Power and energy losses during torque speed sequence shift [11]

Powertrain efficiency check with respect to hybrid mode: In HCU, powertrain efficiency was calculated for each gear with respect to selected hybrid mode. During the calculation of the losses, electric motor and inverter efficiencies should be considered. There will be also losses during the transformation of kinetic energy into chemical energy for charging the battery and during using the stored energy. For the predictive selection, powertrain efficiency should also be predicted according to engine and electric motor torque/speed distributions with the consideration of torque splits.

Output of this subsystem of Evaluation of the selected gears will be processed further in Evaluation of future gear trends.

Since conventional vehicle is used for the validation of the architecture, this subsystem will not be implemented in the software.

Evaluation of future gear trends: Under this subsystem of Evaluation of the selected gears, evaluation of engine efficiency, clutch heat dissipation and powertrain efficiencies with respect to hybrid mode will be processed further. First of all, some gears will be inhibited by checking the engine efficiency for the predicted gears. If predicted most efficient gear is different than the current gear, engine efficiency improvement will be checked. If engine efficiency is not changed more than a parameterizable value, current gear will be kept in this time step.

The second evaluation will consider shift losses because of clutch heat dissipation. When predictive calculation gives the scenario in the Figure 2.12, two possibility will be evaluated. First scenario calculates the power losses during shifting. For example, calculation of power losses during shifting from gear 2 to gear 3, engine losses during driving in gear 3 and power losses during shifting from gear 3 to gear 2 will be added. Second scenario calculates engine losses during driving in the actual gear, in this figure gear 2. For the same time interval, losses are compared and most efficient scenario is selected.

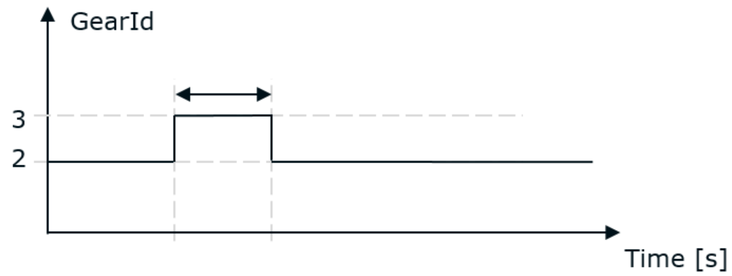


Figure 2.12: Possible Shift Scenario, change of a target gear

Feedback processed data: The idea behind treats the question, if it is possible to monitor the data that is generated, and according to generated data, responses will be monitored. In the end of the Gear Selection, under 'Experience' system, see Figure 2.4, gear request and efficiency of the responses will be monitored. In the end, weight factor will be generated and evaluated here. This is a curve fitting or so called learning algorithm. As an example, after the desired gear is engaged, driver torque demand will be checked whether it is fulfilled. How was the behavior? Was it the correct gear? In the end weighting factor will be calculated and used during evaluation of the gear afterwards under this subsystem. The other example can be, if early shift was occured several times, weighting factor will be changed, so that there will not be early shift next time.

Calculation of allowed gear:

After the evaluation of the possible factors in Gear Selection in the transmission software, last calculation will be done under this subsystem of Calculation of allowed gear for defining the desired gear and sending as an output from the Gear selection strategy.

Calculated desired gear will be combined here with the gear lever position and dynamic flags. Transmissions have general 5 gear lever conditions, which are manual, drive, reverse, park and neutral. If park or neutral gear lever is engaged, neutral gear will be selected. If reverse gear lever is engaged, reverse gear will be selected. If manual gear lever mode is requested, calculated gear in the before subsystems will be disabled and gear selection will be done according to driver manual gear selection. Finally, if drive mode is selected, evaluated gears according to predicted values as well as current values will be combined under this subsystem and sent as a desired gear. Dynamic changes have always priority over the other factors. If there is upshift/downshift prohibition or selected gear is not valid under NVH point of view, calculated gear with respect to efficiencies, driver requests will be cancelled. In this time, gear selection will be done according to dynamic changes limits.

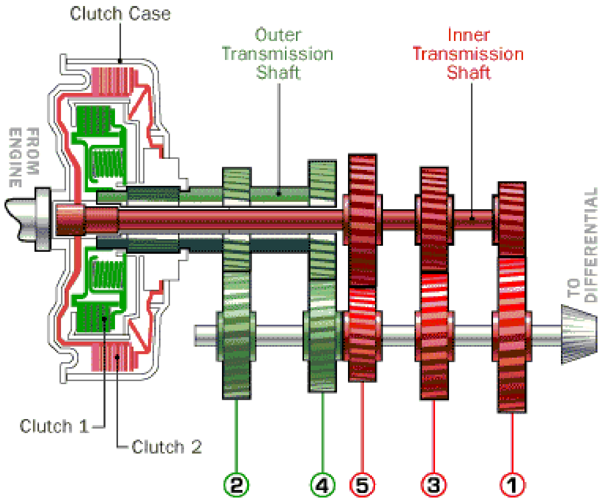


Figure 3.1: 5 speed DCT Architecture [5]

3 Software Description

The proposed architecture in chapter: 2.Method Description is applicable for the different kinds of powertrain architecture and transmission types like conventional, hybrid, AT (Automatic Transmission), DHT, whereas in the present work, software is implemented for 1.2 liter inline 3 TGDI (Turbocharged Gasoline Direct Injection) engine with a wet clutch 7 speed DCT in SUV. An example of a DCT transmission can be seen in Figure 3.1. DCT has two separate shafts, where even and odd gears are separated onto these shafts. One clutch is used just for engaging odd gears and other clutch is used just for engaging even gears. With this two shaft architecture, it is possible to preselect the coming gear. For example, when the vehicle speeds up, controller in the car automatically detect the gear change point and will preselect the upper gear. In the same way, when the vehicle speeds down, controller will preselect a lower gear. When gear change is requested, currently engaged clutch disengages and the other preselected clutch engages.

Software implementation is done in Matlab/Simulink environment [4]. Simulink, developed by MathWorks, is a graphical programming environment for modeling, simulating and analyzing multidomain dynamical systems. It's primary interface is a graphical block diagramming tool and a customizable set of block libraries.

Implemented software according to explained architecture in section 2. Method Description can be seen in Figure 3.2. Under this section, software implementation will be explained. Moreover, it should be noted down that 2_Velocity/ProfilePrediction is an empty subsystem where predicted vehicle velocity is calculated. For the validation of the software, vehicle velocity profile vector is given as an input to the system and processed in subsystem 3_EngineTorque/SpeedProfile.

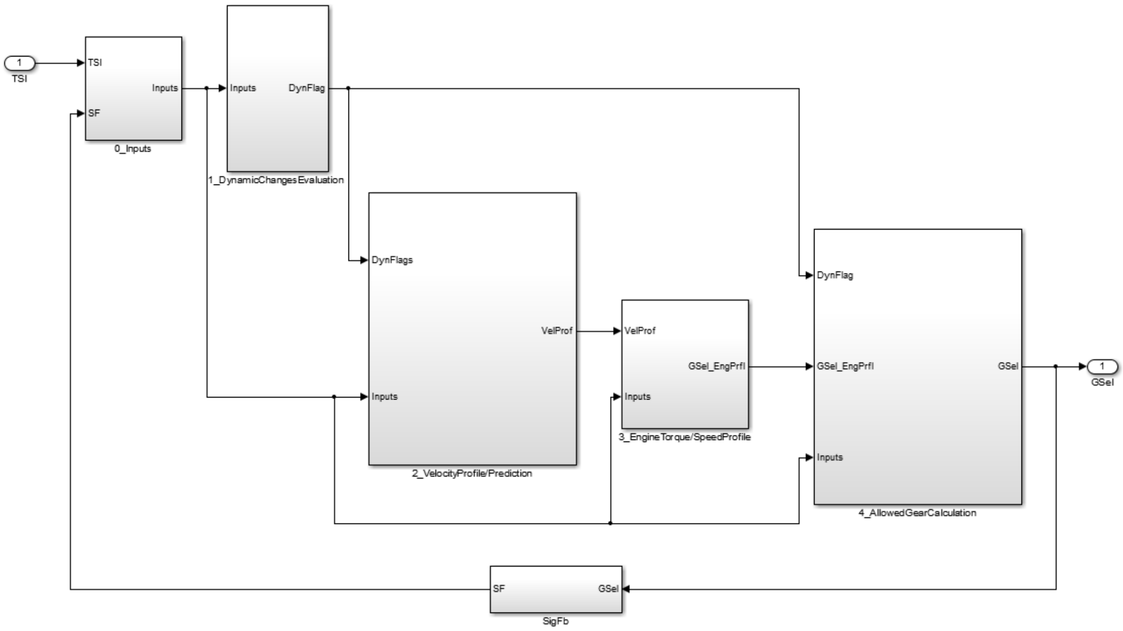


Figure 3.2: Implemented Software in Matlab/Simulink Environment

The figure 3.2 shows the subsystems in execution order from left to right. The arrows represent signal flows. Input port TSI takes the required interfaces from AVL VSM and output port GSel sends required outputs from gear selection system to the AVL VSM. Integration with AVL VSM will be explained more in the coming section C.Simulation and Testing Environment.

Please see Appendix A.Legend for the Simulink Blocks for the functionalities of the simulink blocks.

3.1 0_Inputs

Inputs are processed and gathered to the software from this subsystem, which can be seen in Figure 3.3. Signal naming is done according to names in the project to be compared. Explanations of the input signals can be seen in Table 3.1.

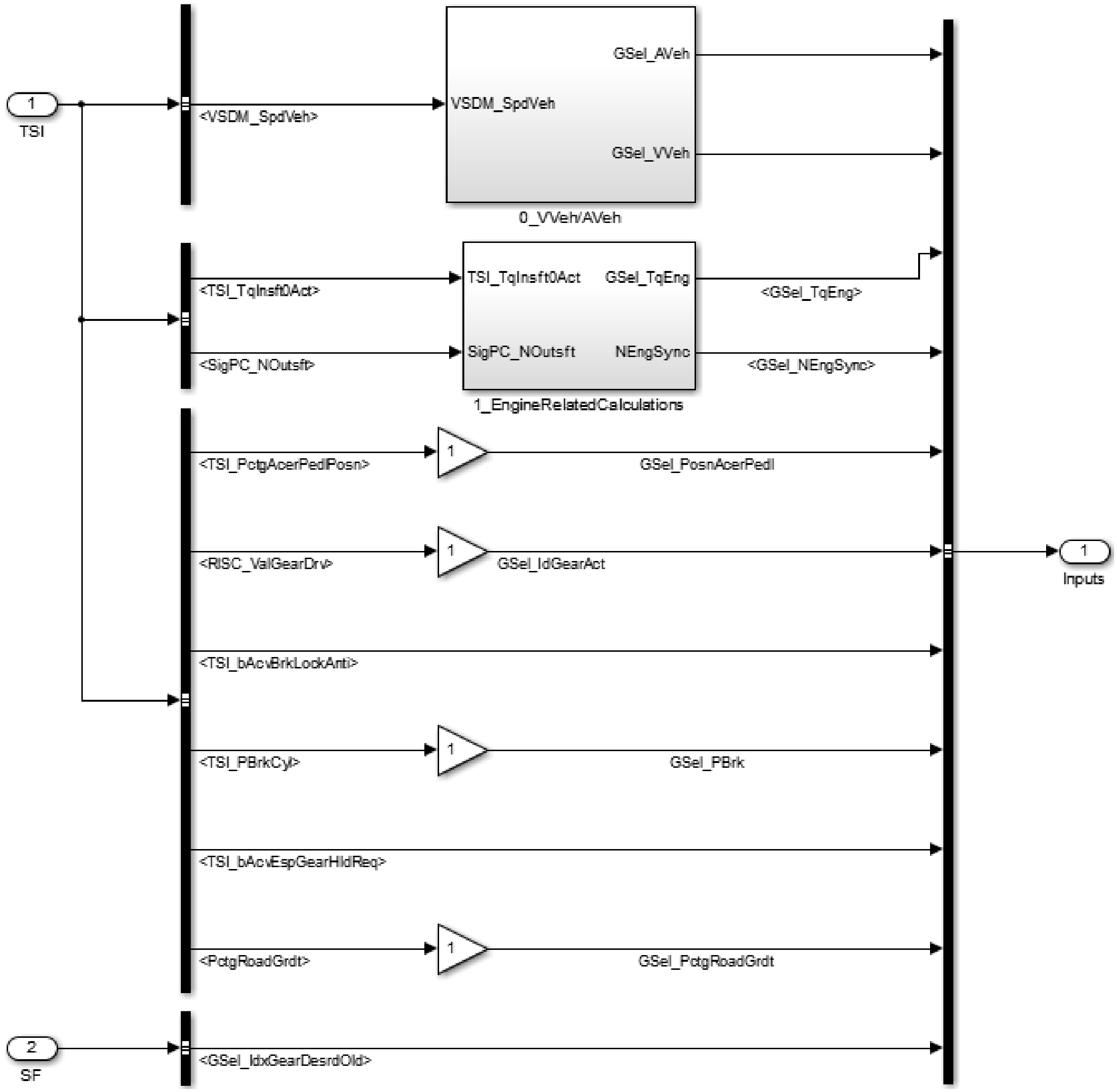


Figure 3.3: Input subsystem implementation in Matlab/Simulink Environment

Table 3.1: Input list of the sofware

Signal name	Description
Signal name	**Description**
VSDM_SpdVeh	Vehicle velocity in km/h
TSI_TqInsft0Act	Engine torque in Nm
SigPC_NOutsft	Output shaft speed in rpm
TSI_PctgAcerPedlPosn	Acceleration pedal position in percentage
RlSC_ValGearDrv	Actual gear value
TSI_bAcvBrkLockAnti	ABS activation flag
TSI_PBrkCyl	Brake cylinder pressure in bar
TSI_bAcvEspGearHldReq	ESP activation that indicates actual gear shall be hold
PctgRoadGrdt	Road gradient input as a percentage
GSel_IdxGearDesrdOld	One sampled delayed value of the requested gear

3.1.1 0_VVeh/AVeh

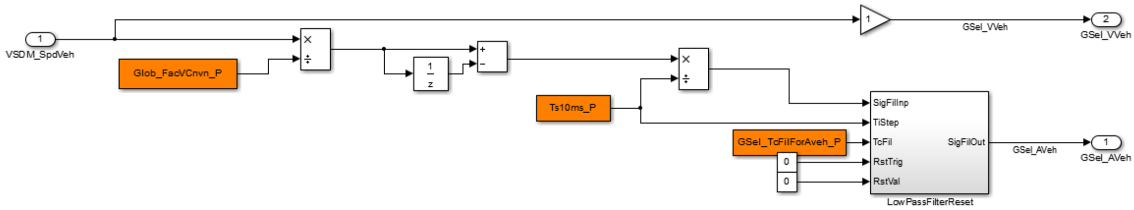


Figure 3.4: Calculation of Vehicle Acceleration in Matlab/Simulink Environment

Figure 3.4 shows the calculation of the vehicle acceleration from the vehicle velocity input under subsystem '0_VVeh/AVeh'.

Input and Output signals description can be seen in Table 3.1 and 3.3 respectively. Parameter descriptions can be seen in Table 3.2.

Vehicle velocity is feed through with changing the signal name. For the vehicle acceleration calculation: first of all unit conversion from km/h to m/s is applied by division with parameter, afterwards derivative operation is applied. Since the simulation step is 10 ms, subtraction is divided to the 10 ms for getting acceleration unit as m/s. In the end of the calculation, Lowpass filter is implemented by library, incase of noises in the vehicle velocity information. Please see Appendix Figure B.3 for detailed implementation and description of the low pass filter.

Table 3.2: Calibrateable parameters used in 0_Inputs subsystem

Parameter name	Description	Value
Glob_FacVCnvn_P	Unit conversion factor from km/h to m/s	3,6
GSel_TcFilForAVeh_P	Vehicle acceleration filtering, low pass filter time constant	0,01
Ts10ms_P	Sample time in sec	0,01

Table 3.3: Output list of subsystem 0_VVeh/AVeh

Signal name	Description
GSel_VVeh	Vehicle velocity in m/s
GSel_AVeh	Vehicle acceleration in m/s^2

3.1.2 1_EngineRelatedCalculations

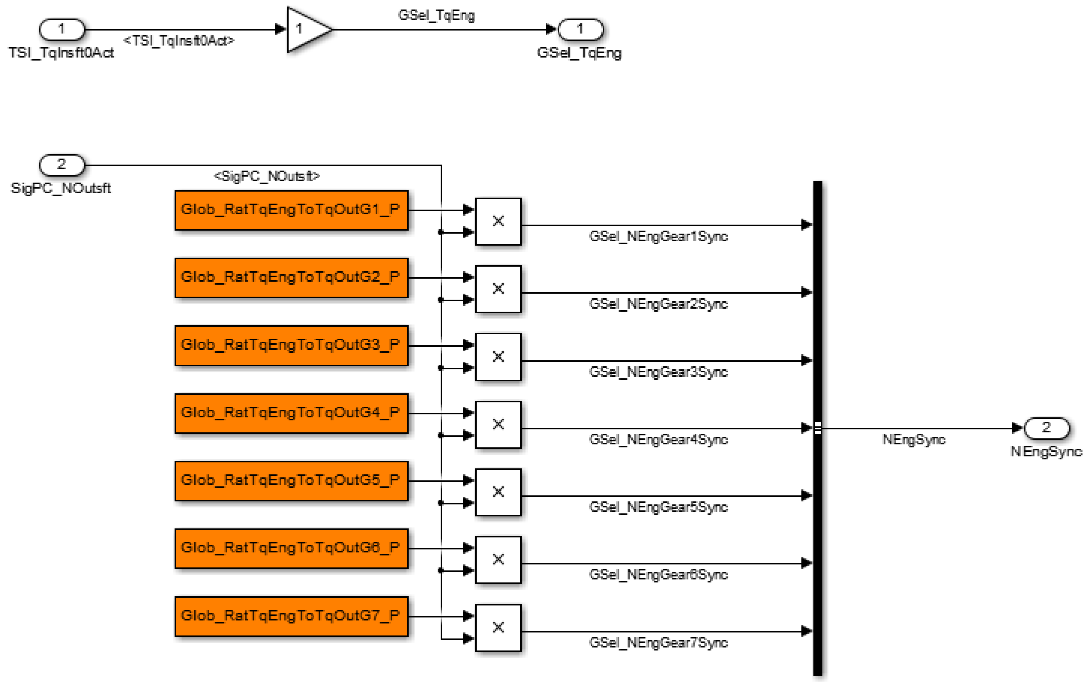


Figure 3.5: Calculation of synchronized engine speeds in Matlab/Simulink Environment

Figure 3.5 shows the synchronized engine speed calculation out of output shaft speed with respect to gear ratios. Gear ratios are defined from engine level to output shaft level. Input and output signals explanation can be seen in Table 3.1 and 3.4 respectively. Please note that final drive ratio is also included in these parameters, where detailed explanations can be seen in Table 3.5.

Table 3.4: Output list of subsystem 0_EngineRelatedCalculations

Signal name	**Description**
GSel_TqEng	Engine torque in Nm
GSel_NEngGearXSync	Engine synchronized speed for gear X in rpm

Table 3.5: Calibrateable parameters used in 1_EngineRelatedCalculations subsystem

Parameter name	**Description**	**Value**
Glob_RatTqEngToTq-OutG1_P	Gear ratio of gear 1, final drive ratio is included	16,800
Glob_RatTqEngToTq-OutG2_P	Gear ratio of gear 2, final drive ratio is included	9,444
Glob_RatTqEngToTq-OutG3_P	Gear ratio of gear 3, final drive ratio is included	6,323
Glob_RatTqEngToTq-OutG4_P	Gear ratio of gear 4, final drive ratio is included	4,718
Glob_RatTqEngToTq-OutG5_P	Gear ratio of gear 5, final drive ratio is included	3,498
Glob_RatTqEngToTq-OutG6_P	Gear ratio of gear 6, final drive ratio is included	2,776
Glob_RatTqEngToTq-OutG7_P	Gear ratio of gear 7, final drive ratio is included	2,386

3.2 1_DynamicChangesEvaluation

Figure 3.6: Calculation of Dynamic Changes Flags in Matlab/Simulink Environment

Dynamic changes are evaluated under this subsystem, where dynamic change types are grouped under different subsystems. Detailed explanations will be presented in the coming pages.

- 0_HeavyBrakeFlag: Heavy brake detection flag calculation
- 1_FastOffFlag: Fast off detection flag calculation
- 2_ABS/ESPFlag: ABS/ESP flag detection
- 3_KickDownFlag: Kickdown detection flag calculation

3.2.1 0_HeavyBrakeFlag

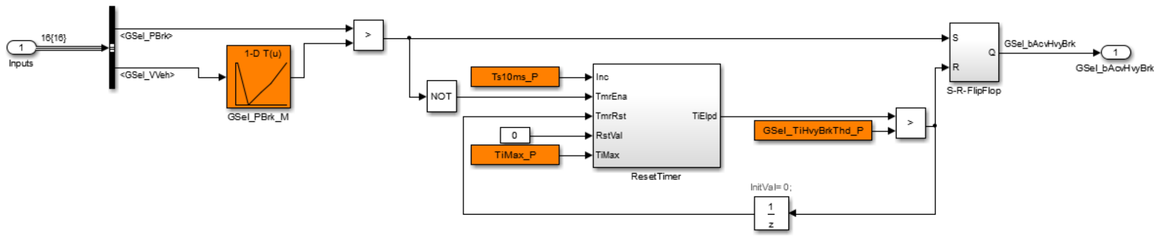


Figure 3.7: Calculation of Heavy Brake Flag in Matlab/Simulink Environment

Inputs/Outputs description can be seen in Table 3.6. Parameter descriptions can be seen in Table 3.7.

Brake pedal pressure is compared against a parameterizable map, which depends on a vehicle velocity. If brake pedal pressure is greater than a parameterizable map output, heavy brake flag is set(S-R-FlipFlop library block, see Figure B.1). When brake pedal pressure is not greater anymore from the output of the parameterizable map, timer is started (ResetTimer library block, see Figure B.2). After a parameterizable time, flag is reseted.

Table 3.6: Input/Output signals used in 0_HeavyBrakeFlag subsystem

Signal name	**Description**
GSel_PBrk	Brake pedal pressure in *bar* comes from simulation environment
GSel_VVeh	Vehicle velocity in km/h comes from simulation environment
GSel_bAcvHvyBrk	Output signal of heavy brake flag calculation. Flag is raised, when heavy brake is detected.

Table 3.7: Calibrateable parameters used in 0_HeavyBrakeFlag subsystem

Parameter name	**Description**	**Value**
TiMax_P	Maximum timer count value in 10 ms	1000
Ts10ms_P	Sample time in sec	0,01
GSel_VVehForBrkP_A	Axes for determining the heavy brake with the GSel_PBrk_M	[0 5 25 50 70 100]
GSel_PBrk_M	Brake pressure threshold value in *bar* with respect to vehicle velocity to be able to determine heavy brake	[150 150 50 80 100 150]
GSel_TiHvyBrkThd_P	In heavy brake flag calculation, timer counter output comparison threshold in sec	0.5

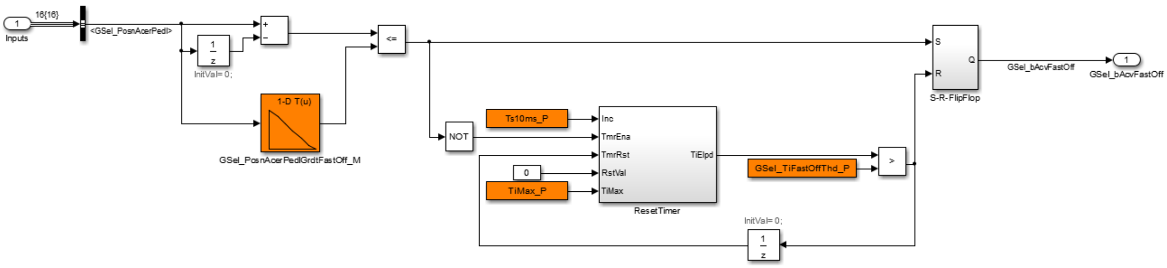


Figure 3.8: Calculation of Fast off Flag in Matlab/Simulink Environment

3.2.2 1_FastOffFlag

Inputs/Outputs description can be seen in Table 3.8. Parameter descriptions can be seen in Table 3.9.

If accelerator pedal position change is lower or equal to the parameterizable map depends on accelerator pedal, Fast off flag is set(S-R-FlipFlop block, see Figure B.1). When this comparison is not correct anymore, timer is started (ResetTimer block, see Figure B.2). After a parameterizable time, flag is reseted.

Table 3.8: Input/Output signals used in 1_FastOffFlag subsystem

Signal name	**Description**
GSel_PosnAcerPedl	Accelerator pedal position in *pct* comes from simulation environment
GSel_bAcvFastOff	Output signal of fast off flag calculation. Flag is raised, when fast off is detected.

Table 3.9: Calibrateable parameters used in 1_FastOffFlag subsystem

Parameter name	**Description**	**Value**
TiMax_P	Maximum timer count value in 10 ms	1000
Ts10ms_P	Sample time in sec	0,01
GSel_PosnAcerPedlGrdt-FastOff_M	Acceleration pedal gradient threshold value in $pct/10ms$ with respect to acceleration pedal position to be able to determine fast off	[-12,5 -17,5 -22,5 -27,5 -32,5 -37,5 -40]
GSel_PosnAcerPedlFast-Off_A	Axes for determining the fast off with GSel_PosnAcerpedlGrdtFastOff_M	[0 20 35 50 70 85 101]
GSel_TiFastOffThd_P	In fast off flag calculation, timer counter output comparison threshold in sec	0,5

3.2.3 2_ABS/ESPFlag

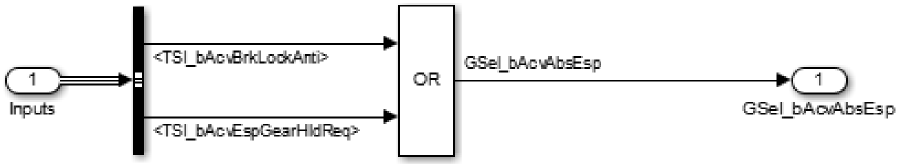


Figure 3.9: Calculation of ABS/ESP Flag in Matlab/Simulink Environment

Inputs/Outputs description can be seen in Table 3.10.

If ABS is activated or ESP gear hold request is activated, ABS/ESP activation flag is activated as well. This implementation can be vary depending on ESP interfaces. In the current project, ESP gives a gear hold request, but there may be also situation, that ESP can request upshift for reducing the output shaft torque. In that time, this part should be updated.

Table 3.10: Input/Output signals used in 2_ABS/ESPFlag subsystem

Signal name	Description
TSI_bAcvEspGearHldReq	ESP activation flag that indicates actual gear shall be hold, comes from simulation environment
TSI_bAcvBrkLockAnti	ABS activation flag comes from simulation environment
GSel_bAcvAbsEsp	Output signal of ABS/ESP flag calculation. Flag is raised, when ABS or ESP is activated.

3.2.4 3_KickDownFlag

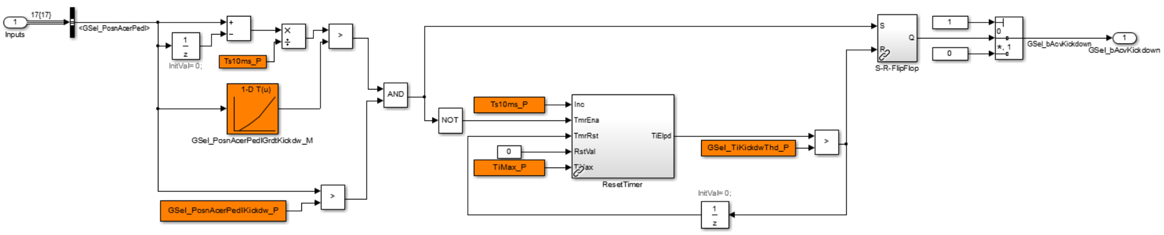


Figure 3.10: Calculation of Kickdown Flag in Matlab/Simulink Environment

Inputs/Outputs description can be seen in Table 3.11. Parameter descriptions can be seen in Table 3.12.

If accelerator pedal position change is higher than the parameterizable map depends on accelerator pedal position and accelerator pedal position is greater than a parameterizable value, Kickdown flag is set (S-R-FlipFlop block, see Figure B.1). When this comparison is not correct anymore, timer is started (ResetTimer block, see Figure B.2). After a parameterizable time, flag is reseted. Kickdown flag calculation logic can be overwritten by a multiport switch. If it is as seen in the Figure 3.10, first input is one, third input is given as output of the multiport switch.

Table 3.11: Input/Output signals used in 3_KickDownFlag subsystem

Signal name	**Description**
GSel_PosnAcerPedl	Accelerator pedal position in *pct* comes from simulation environment
GSel_bAcvKickdown	Output signal of Kickdown flag calculation. Flag is raised, when kickdown is detected.

Table 3.12: Calibrateable parameters used in 3_KickDownFlag subsystem

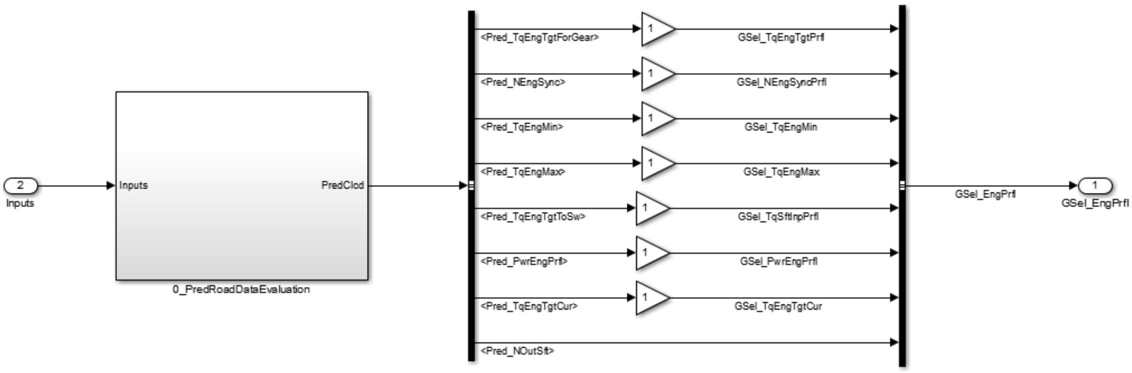
Parameter name	**Description**	**Value**
TiMax_P	Maximum timer count value in 10 ms	1000
Ts10ms_P	Sample time in sec	0,01
GSel_PosnAcerPedlGrdt-Kickdw_M	Acceleration pedal gradient threshold value in $pct/10ms$ with respect to acceleration pedal position to be able to determine kick down	[7 7 15 25 50]
GSel_PosnAcerPedl-Kickdw_A	Axes for determining the kick down with GSel_PosnAcerPedlGrdtKickdw_M	[5 15 35 55 85]
GSel_PosnAcerPedl-Kickdw_P	Accelerator pedal position threshold for kickdown determination in percentage	90

3.3 3_EngineTorque/SpeedProfile

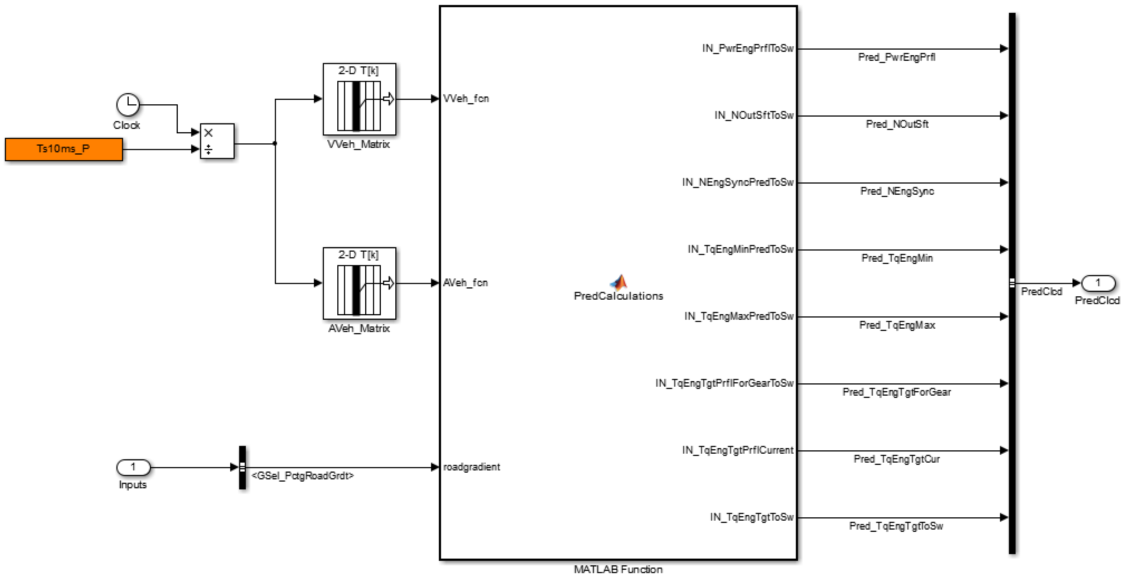
Under this subsystem, engine profile calculation is implemented by matlab function block, see Figure 3.11. In each sample time, 10 ms, vehicle velocity and acceleration vector are updated. Number of elements of these vectors depend on predicted time span, which can be parameterized for this simulation.

The other input is road gradient in percentage. For the simulation, road gradient is taken as a constant value. That's the reason, it is not implemented as a vector like vehicle velocity and acceleration. When road gradient is calculated from predictive road data properly, this value should be also taken as a vector, since it can change in the defined time span.

Inputs/Outputs description can be seen in Table 3.13. All parameters used in this subsystem are summarized in Table 3.18



(a) Prediction Road Data Evaluation Subsystem



(b) Prediction Road Data Evaluation shown with Matlab function

Figure 3.11: Calculation of Engine Torque and Speed Profile in Matlab/Simulink Environment

Table 3.13: Input/Output signals used in 3_EngineTorque/SpeedProfile subsystem

Signal name	**Description**
VVeh_fcn	Predicted vehicle velocity profile given to the software as an input in m/s
AVeh_fcn	Predicted vehicle acceleration profile given to the software as an input in m/s^2
GSel_PctgRoadGrdt	Road gradient given to the software as an input in $pctg$
Pred_PwrEngPrfl	Engine power profile in the parameterizable time interval in $Watt$. It is a vector, which elements represent the value for each sample time
Pred_NOutSft	Output shaft speed profile in the parameterizable time interval rpm. It is a vector, which elements represent the value for each sample time
Pred_NEngSync	Synchronized engine speed profile matrix in rpm, which each column represents gear id and rows are value for each sample time
Pred_TqEngMin	Minimum engine torque profile matrix in Nm, which each column represents gear id and rows are value for each sample time
Pred_TqEngMax	Maximum engine torque profile matrix in Nm, which each column represents gear id and rows are value for each sample time
Pred_TqEngTgtForGear	Engine torque target profile for each gear in parameterizable time interval in Nm. It is a matrix , which each column represents gear id and rows are value for each sample time
Pred_TqEngTgtCur	Engine torque target for each gear in present sample time in Nm
Pred_TqEngTgtToSw	Engine torque target profile in parameterizable time interval in Nm. It is a vector, which elements represent the value for each time

```
%% Inputs assignment
IN_VVehProfile = VVeh_fcn; %vehicle velocity vector is assigned to a local variable
IN_AVehProfile = AVeh_fcn; %vehicle acceleration vector is assigned to a local variable
%% Engine power profile calculation
%Road gradient calculation
alpha = atan(roadgradient/100);
% %Air resistance
Rd = (0.5*Glob_CoeffAirDrag_P*Glob_ArVehRef_P*Glob_RhoAir_P).*IN_VVehProfile.*IN_VVehProfile;
% %Climbing resistance force
Rc = GSel_MVehDrvtr_P*Glob_AGrv_P*sin(alpha);
% %Rolling resistance
Rr = GSel_MVehDrvtr_P*Glob_AGrv_P*cos(alpha)*0.03;
PwrEngMinThdForDecel = 0;
IN_PwrEngPrfl = (((IN_AVehProfile.*GSel_MVehDrvtr_P.*GSel_FacMVehCorrn_P +...
                  Rd + Rc + Rr).*IN_VVehProfile));%unit; (kg*m^2/s^3)watt
```

Figure 3.12: Calculation of Engine Power by Matlab Function

In 2. Method Description part, related formulas were introduced for calculation of the engine power. In Figure 3.12, engine power calculation for the defined prediction span can be seen. Descriptions of the calculated values are summarized in the Table 3.14.

Table 3.14: Calculated values in Figure 3.12

Signal name	**Description**
IN_VVehProfile	Predicted vehicle velocity to be used in script in m/s
alpha	road gradient angle in rad
IN_AVehProfile	Predicted vehicle acceleration to the used in script in m/s^2
Rd	Air resistance force N.
Rc	Climbing resistance force rpm.
Rr	Rolling resistance force rpm.
IN_PwrEngPrfl	Engine power profile in the parameterizable time interval to be used in script in $Watt$. It is a vector by [redictiontimespan]

```
%% Engine gear synchronized speed calculation
IN_NOutSft = ((IN_VVeh_m_s*60)./(2*Glob_ValPi_P*Glob_RdVehWhlDyn_P)); %unit:rpm
IN_NEngGear1Sync = IN_NOutSft.*Glob_RatTqEngToTqOutG1_P;
IN_NEngGear2Sync = IN_NOutSft.*Glob_RatTqEngToTqOutG2_P;
IN_NEngGear3Sync = IN_NOutSft.*Glob_RatTqEngToTqOutG3_P;
IN_NEngGear4Sync = IN_NOutSft.*Glob_RatTqEngToTqOutG4_P;
IN_NEngGear5Sync = IN_NOutSft.*Glob_RatTqEngToTqOutG5_P;
IN_NEngGear6Sync = IN_NOutSft.*Glob_RatTqEngToTqOutG6_P;
IN_NEngGear7Sync = IN_NOutSft.*Glob_RatTqEngToTqOutG7_P;
IN_NEngSync = [IN_NEngGear1Sync IN_NEngGear2Sync IN_NEngGear3Sync IN_NEngGear4Sync...
    IN_NEngGear5Sync IN_NEngGear6Sync IN_NEngGear7Sync];
```

Figure 3.13: Calculation of Synchronized Engine Speed by Matlab function

In Figure 3.13, output shaft and engine synchronized speed for each gear for the defined predicted time span is calculated. IN_NEngSync is a matrix, which each column represents gear id and rows are value for each sample time. Descriptions of the calculated values are summarized in the Table 3.15

Table 3.15: Calculated values in Figure 3.13

Signal name	**Description**
IN_NOutSft	Calculated output shaft speed vector in rpm, which elements represent the value for each sample time
IN_NEngGearXSync	Engine synchronized speed vector for gear X rpm, which elements represent the value for each sample time
IN_NEngSync	Calculated engine synchronized speed matrix [predictiontimespan,7] in rpm

```matlab
%% Engine torque calculation
%Inertia loss as torque in output shaft level
GSel_TqJLossTotForG1 = IN_AVehProfile.*(2*GSel_JWhl_P + GSel_JEng_P*Glob_RatTqEngToTqOutG1_P^2 ...
                            + GSel_JTrsmG1Out_P)./Glob_RdVehWhlDyn_P^2;
GSel_TqJLossTotForG2 = IN_AVehProfile.*(2*GSel_JWhl_P + GSel_JEng_P*Glob_RatTqEngToTqOutG2_P^2 ...
                            + GSel_JTrsmG2Out_P)./Glob_RdVehWhlDyn_P^2;
GSel_TqJLossTotForG3 = IN_AVehProfile.*(2*GSel_JWhl_P + GSel_JEng_P*Glob_RatTqEngToTqOutG3_P^2 ...
                            + GSel_JTrsmG3Out_P)./Glob_RdVehWhlDyn_P^2;
GSel_TqJLossTotForG4 = IN_AVehProfile.*(2*GSel_JWhl_P + GSel_JEng_P*Glob_RatTqEngToTqOutG4_P^2 ...
                            + GSel_JTrsmG4Out_P)./Glob_RdVehWhlDyn_P^2;
GSel_TqJLossTotForG5 = IN_AVehProfile.*(2*GSel_JWhl_P + GSel_JEng_P*Glob_RatTqEngToTqOutG5_P^2 ...
                            + GSel_JTrsmG5Out_P)./Glob_RdVehWhlDyn_P^2;
GSel_TqJLossTotForG6 = IN_AVehProfile.*(2*GSel_JWhl_P + GSel_JEng_P*Glob_RatTqEngToTqOutG6_P^2 ...
                            + GSel_JTrsmG6Out_P)./Glob_RdVehWhlDyn_P^2;
GSel_TqJLossTotForG7 = IN_AVehProfile.*(2*GSel_JWhl_P + GSel_JEng_P*Glob_RatTqEngToTqOutG7_P^2 ...
                            + GSel_JTrsmG7Out_P)./Glob_RdVehWhlDyn_P^2;
GSel_PwrJLossTotForG1 = GSel_TqJLossTotForG1.*(2*Glob_ValPi_P.*(IN_NOutSft/60));
GSel_PwrJLossTotForG2 = GSel_TqJLossTotForG2.*(2*Glob_ValPi_P.*(IN_NOutSft/60));
GSel_PwrJLossTotForG3 = GSel_TqJLossTotForG3.*(2*Glob_ValPi_P.*(IN_NOutSft/60));
GSel_PwrJLossTotForG4 = GSel_TqJLossTotForG4.*(2*Glob_ValPi_P.*(IN_NOutSft/60));
GSel_PwrJLossTotForG5 = GSel_TqJLossTotForG5.*(2*Glob_ValPi_P.*(IN_NOutSft/60));
GSel_PwrJLossTotForG6 = GSel_TqJLossTotForG6.*(2*Glob_ValPi_P.*(IN_NOutSft/60));
GSel_PwrJLossTotForG7 = GSel_TqJLossTotForG7.*(2*Glob_ValPi_P.*(IN_NOutSft/60));
%Engine torque target
IN_TqEngTgt = IN_PwrEngPrfl./(2*Glob_ValPi_P.*(IN_NOutSft/60));  %output shaft torque
IN_PctgEngTqGear1 =  IN_TqEngTgt./Glob_RatTqEngToTqOutG1_P.*(100)./(350);
IN_PctgEngTqGear2 =  IN_TqEngTgt./Glob_RatTqEngToTqOutG2_P.*(100)./(350);
IN_PctgEngTqGear3 =  IN_TqEngTgt./Glob_RatTqEngToTqOutG3_P.*(100)./(350);
IN_PctgEngTqGear4 =  IN_TqEngTgt./Glob_RatTqEngToTqOutG4_P.*(100)./(350);
IN_PctgEngTqGear5 =  IN_TqEngTgt./Glob_RatTqEngToTqOutG5_P.*(100)./(350);
IN_PctgEngTqGear6 =  IN_TqEngTgt./Glob_RatTqEngToTqOutG6_P.*(100)./(350);
IN_PctgEngTqGear7 =  IN_TqEngTgt./Glob_RatTqEngToTqOutG7_P.*(100)./(350);
%Transmission gear efficiency
GSel_EffTrsmGear1 = interp2(GSel_PctgEngTqForTrsmEff_A,GSel_NEngForTrsmEff_A,GSel_EffTrsmGear1_M,...
    IN_PctgEngTqGear1,IN_NEngGear1Sync, 'linear', GSel_EffTrsmGear1_M(1,1))/100;
GSel_EffTrsmGear2 = interp2(GSel_PctgEngTqForTrsmEff_A,GSel_NEngForTrsmEff_A,GSel_EffTrsmGear2_M,...
    IN_PctgEngTqGear2,IN_NEngGear2Sync, 'linear', GSel_EffTrsmGear2_M(1,1))/100;
GSel_EffTrsmGear3 = interp2(GSel_PctgEngTqForTrsmEff_A,GSel_NEngForTrsmEff_A,GSel_EffTrsmGear3_M,...
    IN_PctgEngTqGear3,IN_NEngGear3Sync, 'linear', GSel_EffTrsmGear3_M(1,1))/100;
GSel_EffTrsmGear4 = interp2(GSel_PctgEngTqForTrsmEff_A,GSel_NEngForTrsmEff_A,GSel_EffTrsmGear4_M,...
    IN_PctgEngTqGear4,IN_NEngGear4Sync, 'linear', GSel_EffTrsmGear4_M(1,1))/100;
GSel_EffTrsmGear5 = interp2(GSel_PctgEngTqForTrsmEff_A,GSel_NEngForTrsmEff_A,GSel_EffTrsmGear5_M,...
    IN_PctgEngTqGear5,IN_NEngGear5Sync, 'linear', GSel_EffTrsmGear5_M(1,1))/100;
GSel_EffTrsmGear6 = interp2(GSel_PctgEngTqForTrsmEff_A,GSel_NEngForTrsmEff_A,GSel_EffTrsmGear6_M,...
    IN_PctgEngTqGear6,IN_NEngGear6Sync, 'linear', GSel_EffTrsmGear6_M(1,1))/100;
GSel_EffTrsmGear7 = interp2(GSel_PctgEngTqForTrsmEff_A,GSel_NEngForTrsmEffGear7_A,GSel_EffTrsmGear7_M,...
    IN_PctgEngTqGear7,IN_NEngGear7Sync, 'linear', GSel_EffTrsmGear7_M(1,1))/100;
%Engine torque target for each gear
IN_TqEngGear1Tgt = max(PwrEngMinThdForDecel,(IN_PwrEngPrfl+GSel_PwrJLossTotForG1))./...
    (2*Glob_ValPi_P.*(IN_NEngGear1Sync/60).*GSel_EffTrsmGear1);
IN_TqEngGear2Tgt = max(PwrEngMinThdForDecel,(IN_PwrEngPrfl+GSel_PwrJLossTotForG2))./...
    (2*Glob_ValPi_P.*(IN_NEngGear2Sync/60).*GSel_EffTrsmGear2);
IN_TqEngGear3Tgt = max(PwrEngMinThdForDecel,(IN_PwrEngPrfl+GSel_PwrJLossTotForG3))./...
    (2*Glob_ValPi_P.*(IN_NEngGear3Sync/60).*GSel_EffTrsmGear3);
IN_TqEngGear4Tgt = max(PwrEngMinThdForDecel,(IN_PwrEngPrfl+GSel_PwrJLossTotForG4))./...
    (2*Glob_ValPi_P.*(IN_NEngGear4Sync/60).*GSel_EffTrsmGear4);
IN_TqEngGear5Tgt = max(PwrEngMinThdForDecel,(IN_PwrEngPrfl+GSel_PwrJLossTotForG5))./...
    (2*Glob_ValPi_P.*(IN_NEngGear5Sync/60).*GSel_EffTrsmGear5);
IN_TqEngGear6Tgt = max(PwrEngMinThdForDecel,(IN_PwrEngPrfl+GSel_PwrJLossTotForG6))./...
    (2*Glob_ValPi_P.*(IN_NEngGear6Sync/60).*GSel_EffTrsmGear6);
IN_TqEngGear7Tgt = max(PwrEngMinThdForDecel,(IN_PwrEngPrfl+GSel_PwrJLossTotForG7))./...
    (2*Glob_ValPi_P.*(IN_NEngGear7Sync/60).*GSel_EffTrsmGear7);
IN_TqEngTgtPrfl = [IN_TqEngGear1Tgt IN_TqEngGear2Tgt IN_TqEngGear3Tgt IN_TqEngGear4Tgt...
    IN_TqEngGear5Tgt IN_TqEngGear6Tgt IN_TqEngGear7Tgt];
IN_TqEngGear7Max = interp1(GSel_NEngMax_A,GSel_TqEngMax_M,IN_NEngGear7Sync,'linear', 0);
IN_TqEngMax = [IN_TqEngGear1Max IN_TqEngGear2Max IN_TqEngGear3Max IN_TqEngGear4Max...
    IN_TqEngGear5Max IN_TqEngGear6Max IN_TqEngGear7Max];
```

Figure 3.14: Calculation of Engine Torque Target and Engine Limitation by Matlab Function

In Figure 3.14, engine torque related calculations in matlab function can be seen. Descriptions of the calculated values are summarized in the Table 3.16. Equations, transmission efficiency calculations that are implemented in this code, were explained in 2. Method Description part.

In the first lines, inertia loss as torque in output shaft level is calculated. Calculated inertia losses are converted to the power for each gear.

Afterwards, engine torque target is calculated from engine power and output shaft speed. Percentage engine torque is calculated to be used in transmission efficiency maps from this engine torque target. It should be noted down that, percentage engine torque calculation is not accurate. Since calculated engine torque target does not include any losses. But in this calculation it is necessary for transmission efficiency calculation, since percentage engine torque is one input of transmission efficiency maps. Then, transmission efficiency calculation for each engaged gear by maps are implemented by *interp*(...) matlab function. This is interpolation function, where related output of the transmission efficiency map is calculated by linear interpolation according to axes.

Finally, engine target torque is calculated with the effects of transmission and inertia losses. Inertia losses as power is added to the engine power value, whereas transmission efficiency value is divided. In this stage, engine target torque is limited by a parameter 'PwrEngMinThdForDecel'. This parameter represents engine power minimum for deceleration, which has a default 0 value. The reason is engine efficiency maps are applicable just for the positive engine torque. For the deceleration, namely negative engine torques, different engine maps should be used for gear selection, which is evaluated in the coming parts of the software. Engine target torque is a matrix, which each column represents gear id and rows are value for each sample time.

Table 3.16: Calculated values in Figure 3.14

Signal name	**Description**
GSel_TqJLossTotForGX	Calculated inertia loss vector, which elements represent the value for each sample time, as torque in output shaft level for gear X in Nm.
IN_TqEngTgt	Engine target torque vector, which elements represent the value for each sample time in Nm.
IN_PctgEngTqGearX	Calculated percentage engine torque vector, which elements represent the value for each sample time for gear X in *pct*
GSel_EffTrsmGearX	Calculated transmission efficiency vector, which elements represent the value for each sample time for gear X
IN_TqEngGearXTgt	Calculated engine target torque vector, which elements represent the value for each sample time for gear X in Nm.
IN_TqEngTgtPrfl	Calculated engine target torque matrix [predictiontimespan,7] in Nm

```
% Maximum engine torque
IN_TqEngGear1Max = interp1(GSel_NEngMax_A,GSel_TqEngMax_M,IN_NEngGear1Sync,'linear', 0);
IN_TqEngGear2Max = interp1(GSel_NEngMax_A,GSel_TqEngMax_M,IN_NEngGear2Sync,'linear', 0);
IN_TqEngGear3Max = interp1(GSel_NEngMax_A,GSel_TqEngMax_M,IN_NEngGear3Sync,'linear', 0);
IN_TqEngGear4Max = interp1(GSel_NEngMax_A,GSel_TqEngMax_M,IN_NEngGear4Sync,'linear', 0);
IN_TqEngGear5Max = interp1(GSel_NEngMax_A,GSel_TqEngMax_M,IN_NEngGear5Sync,'linear', 0);
IN_TqEngGear6Max = interp1(GSel_NEngMax_A,GSel_TqEngMax_M,IN_NEngGear6Sync,'linear', 0);
IN_TqEngGear7Max = interp1(GSel_NEngMax_A,GSel_TqEngMax_M,IN_NEngGear7Sync,'linear', 0);
IN_TqEngMax = [IN_TqEngGear1Max IN_TqEngGear2Max IN_TqEngGear3Max IN_TqEngGear4Max...
    IN_TqEngGear5Max IN_TqEngGear6Max IN_TqEngGear7Max];
%Minimum engine torque
IN_TqEngGear1Min = interp1(GSel_NEngMin_A,GSel_TqEngMin_M,IN_NEngGear1Sync,'linear', -10.7);
IN_TqEngGear2Min = interp1(GSel_NEngMin_A,GSel_TqEngMin_M,IN_NEngGear2Sync,'linear', -10.7);
IN_TqEngGear3Min = interp1(GSel_NEngMin_A,GSel_TqEngMin_M,IN_NEngGear3Sync,'linear', -10.7);
IN_TqEngGear4Min = interp1(GSel_NEngMin_A,GSel_TqEngMin_M,IN_NEngGear4Sync,'linear', -10.7);
IN_TqEngGear5Min = interp1(GSel_NEngMin_A,GSel_TqEngMin_M,IN_NEngGear5Sync,'linear', -10.7);
IN_TqEngGear6Min = interp1(GSel_NEngMin_A,GSel_TqEngMin_M,IN_NEngGear6Sync,'linear', -10.7);
IN_TqEngGear7Min = interp1(GSel_NEngMin_A,GSel_TqEngMin_M,IN_NEngGear7Sync,'linear', -10.7);
IN_TqEngMin = [IN_TqEngGear1Min IN_TqEngGear2Min IN_TqEngGear3Min IN_TqEngGear4Min...
    IN_TqEngGear5Min IN_TqEngGear6Min IN_TqEngGear7Min];

%% Signals to be sent to output
IN_PwrEngPrflToSw = IN_PwrEngPrfl; %Sending engine profile
IN_NEngSyncPredToSw = IN_NEngSync; %Sending engine synchronized speed
IN_NOutSftToSw = IN_NOutSft; % Sending output shaft speed
IN_TqEngMaxPredToSw = IN_TqEngMax; %Sending maximum engine torque
IN_TqEngMinPredToSw = IN_TqEngMin; %Sending minimum engine torque
IN_TqEngTgtPrflForGearToSw = IN_TqEngTgtPrfl;  %Sending engine torque target for each gear
IN_TqEngTgtPrflCurrent = IN_TqEngTgtPrfl(1,1:7);  %Sending engine torque target for each gear
IN_TqEngTgtToSw = IN_TqEngTgt;  %Sending engine torque target
```

Figure 3.15: Sending the related outputs from Matlab Function to Simulink Environment

Figure 3.15 shows engine torque limitation calculations and assignment of related output values to be sent to the Simulink environment. Descriptions of the calculated values are summarized in the Table 3.17.

Engine minimum and maximum torques are calculated for each gear. They are implemented by a parameterizable maps according to engine characteristics, depends on synchronized engine speed. Engine torque minimum and maximum limitations are a matrix, which each column represents gear id and rows are value for each sample time.

Table 3.17: Calculated values in Figure 3.15

Signal name	Description
IN_TqEngGearXMax	Calculated maximum engine torque vector [predictiontimespan] for gear X in Nm.
IN_TqEngMax	Calculated maximum engine torque matrix [predictiontimespan,7] in Nm
IN_TqEngGearXMin	Calculated minimum engine torque vector [predictiontimespan], which elements represent the value for each sample time for gear X in Nm.
IN_TqEngMin	Calculated minimum engine torque matrix [predictiontimespan,7] in Nm
IN_PwrEngPrflToSw	Engine power profile in the parameterizable time interval to be used in script in $Watt$. It is a vector by [redictiontimespan]
IN_NEngSyncPredToSw	Calculated engine synchronized speed matrix in rpm, which each column represents gear id and rows are value for each sample time
IN_NOutSftToSw	Calculated output shaft speed vector, which elements represent the value for each sample time in rpm.
IN_TqEngMaxPredToSw	Calculated maximum engine torque matrix in Nm, which each column represents gear id and rows are value for each sample time
IN_TqEngMinPredToSw	Calculated minimum engine torque matrix in Nm, which each column represents gear id and rows are value for each sample time
IN_TqEngTgtToSw	Engine target torque vector, which elements represent the value for each sample time in Nm.
IN_TqEngTgtPrflCurrent	Calculated engine target for each gear in current time in Nm
IN_TqEngTgtPrflFor... GearToSw	Calculated engine target torque matrix in Nm, which each column represents gear id and rows are value for each sample time

Table 3.18: Calibrateable parameters used in 3_EngineTorque/SpeedProfile subsystem

Parameter name	Description	Value
Glob_AGrv_P	Gravitational acceleration on earth in m/s^2	9,81
Glob_CoeffRRollFric_P	Rolling resistance friction coefficient	0,009
Glob_RdVehWhlDyn_P	vehicle dynamic wheel radius in m	0,335
Glob_CoeffAirDrag_P	Air drag coefficient	0,3
Glob_ArVehRef_P	Reference vehicle area to be used in air drag calculation in m^2	2,35

Continued on next page

Table 3.18 – continued from previous page

Parameter name	Description	Value
Glob_RhoAir_P	Air density kg/m^3	1,225
Glob_ValPi_P	Value pi	3,141592654
GSel_MVehDrvtr_P	Vehicle weight in kg	1582
GSel_NEngMax_A	Maximum engine rotational speed in rpm axes for maximum engine torque calculation	[0 750 1000 1500 2000 2500 3000 3500 4000 4500 4650 5000 5500 6000]
GSel_TqEngMax_M	Maximum engine torque in Nm table value for maximum engine torque calculation	[0 75 120 210 210 210 210 210 210 210 210 194 177 120]
GSel_NEngMin_A	Minimum engine rotational speed in rpm axes for minimum engine torque calculation	[750 1000 1500 2000 2500 3000 3500 4000 4500 4650 5000 5500 6000]
GSel_TqEngMin_M	Minimum engine torque in Nm table value for minimum engine torque calculation	[-10,7 -11 -12.2 -12.8 -14.3 -16.1 -18 -20,7 -23,8 -24,73 -26,9 -30,1 -33,55]
Moment of Inertia related parameters		
GSel_JEng_P	Engine inertia in kgm^2	0,173
GSel_JTrsmG1Out_P	Moment of inertia in transmission output for gear 1 in kgm^2	0,127680228
GSel_JTrsmG2Out_P	Moment of inertia in transmission output for gear 2 in kgm^2	0,045754253
GSel_JTrsmG3Out_P	Moment of inertia in transmission output for gear 3 in kgm^2	0,018090944
GSel_JTrsmG4Out_P	Moment of inertia in transmission output for gear 4 in kgm^2	0,005309165
GSel_JTrsmG5Out_P	Moment of inertia in transmission output for gear 5 in kgm^2	0,011912576
GSel_JTrsmG6Out_P	Moment of inertia in transmission output for gear 6 in kgm^2	0,003950731

Continued on next page

Table 3.18 – continued from previous page

Parameter name	Description	Value
GSel_JTrsmG7Out_P	Moment of inertia in transmission output for gear 7 in kgm^2	0,00554048
GSel_JWhl_P	Moment of inertia in wheel in kgm^2	1,1
Transmission efficiency related parameters		
GSel_PctgEngTqForTrsm-Eff_A	Engine load in percentage with respect to maximum load	[10 20 30 40 50 60 70 80 90 100]
GSel_NEngForTrsmEff_A	Engine speed in rpm	[1000 1250 1500 1750 2000 2250 2500 2750 3000 3500]
GSel_EffTrsmGear1_M	Transmission efficiency for gear 1	[91.9 94.4 95.3 95.6 95.6 95.7 95.6 0.0 0.0 0.0 91.0 94.1 95.0 95.4 95.5 95.6 95.6 95.7 95.6 0.0 90.5 93.8 94.8 95.3 95.5 95.6 95.6 95.6 95.6 95.6 89.4 93.3 94.5 95.0 95.4 95.5 95.6 95.6 95.7 95.7 88.4 93.0 94.3 94.7 95.5 95.3 95.7 95.5 95.7 95.7 87.5 92.4 94.0 94.7 95.0 95.3 95.5 95.5 95.6 95.6 86.3 91.9 93.8 94.5 94.8 95.2 95.4 95.5 95.6 95.5 84.8 91.3 93.3 94.2 94.7 95.0 95.2 95.4 95.4 95.5 84.2 90.7 92.9 93.9 94.5 94.9 95.1 95.3 95.4 95.5 81.7 89.5 92.3 93.5 94.3 94.7 94.9 95.2 95.3 95.3]
GSel_EffTrsmGear2_M	Transmission efficiency for gear 2	[92.7 94.8 95.4 96.0 96.0 96.2 96.2 0.0 0.0 0.0 91.3 94.2 95.4 95.8 96.1 96.1 96.2 96.3 96.3 0.0 90.6 94.0 95.2 95.7 96.0 96.2 96.3 96.3 96.4 96.3 90.1 93.8 95.0 95.6 95.9 96.0 96.2 96.2 96.3 96.3 88.1 93.6 94.9 95.1 95.6 96.1 96.1 96.1 96.3 96.4 88.1 92.9 94.4 95.2 95.6 95.9 96.0 96.1 96.2 96.3 87.1 92.3 94.1 95.0 95.4 95.7 95.9 96.0 96.1 96.2 86.0 91.8 93.8 94.7 95.3 95.6 95.8 96.0 96.1 96.2 84.7 91.3 93.4 94.5 95.0 95.4 95.7 95.8 96.1 96.1 83.1 90.3 92.9 94.1 94.8 95.3 95.6 95.8 95.9 96.0]
GSel_EffTrsmGear3_M	Transmission efficiency for gear 3	[93.0 95.5 96.1 96.3 96.4 96.4 96.6 0.0 0.0 0.0 92.1 94.8 95.7 96.2 96.4 96.5 96.5 96.6 96.7 0.0 91.6 94.6 95.5 95.9 96.2 96.4 96.5 96.6 96.6 96.6 90.8 94.1 95.3 95.8 96.2 96.4 96.5 96.6 96.6 96.6 90.5 94.4 95.5 96.0 96.3 96.5 96.6 96.7 96.8 96.8 89.3 93.4 94.8 95.6 96.0 96.2 96.4 96.5 96.6 96.6 88.3 92.9 94.5 95.3 95.8 96.1 96.3 96.5 96.5 96.6 87.1 92.3 94.1 95.1 95.6 96.0 96.2 96.3 96.5 96.6 85.8 91.9 93.9 94.9 95.5 95.8 96.1 96.4 96.5 96.5 84.1 91.1 93.5 94.6 95.3 95.7 96.0 96.2 96.3 96.4]

Continued on next page

Table 3.18 – continued from previous page

Parameter name	Description	Value
GSel_EffTrsmGear4_M	Transmission efficiency for gear 4	[92.5 95.0 95.9 96.4 96.6 96.7 96.9 0.0 0.0 0.0 91.2 94.5 95.6 96.2 96.5 96.7 96.9 96.9 97.0 0.0 90.3 94.1 95.5 96.1 96.4 96.7 96.8 97.0 97.0 97.0 89.8 93.8 95.3 95.9 96.3 96.5 96.7 96.9 96.9 97.0 89.3 93.4 94.7 95.7 96.3 96.4 96.5 96.8 97.0 96.9 87.6 93.0 94.7 95.5 96.0 96.3 96.6 96.7 96.8 96.9 86.4 92.3 94.2 95.2 95.9 96.2 96.4 96.6 96.7 96.8 85.3 91.8 93.9 95.0 95.6 96.0 96.3 96.5 96.7 96.8 84.6 91.4 93.6 94.7 95.5 95.9 96.2 96.5 96.7 96.8 82.5 90.4 93.1 94.4 95.2 95.7 96.1 96.3 96.5 96.6]
GSel_EffTrsmGear5_M	Transmission efficiency for gear 5	[91.5 95.1 95.9 96.5 96.8 96.9 97.0 0.0 0.0 0.0 90.7 94.5 95.6 96.3 96.7 96.9 97.0 97.1 97.1 0.0 89.5 94.0 95.5 96.1 96.5 96.8 96.9 97.1 97.1 97.2 88.9 93.6 95.1 95.9 96.4 96.7 97.0 97.1 97.1 97.2 88.2 93.4 94.7 95.7 96.5 96.6 96.7 97.0 97.2 97.2 86.1 92.5 94.5 95.5 96.1 96.4 96.7 96.8 97.0 97.0 85.6 92.1 94.3 95.3 95.9 96.4 96.6 96.8 96.9 97.0 84.3 91.6 93.8 95.1 95.7 96.2 96.5 96.7 96.8 97.0 83.2 91.1 93.5 94.8 95.5 96.0 96.4 96.6 96.8 96.9 81.5 90.0 93.0 94.4 95.2 95.8 96.2 96.4 96.6 96.8]
GSel_EffTrsmGear6_M	Transmission efficiency for gear 6	[89.8 94.0 95.3 96.1 96.5 96.7 96.9 0.0 0.0 0.0 88.4 93.4 95.0 95.8 96.3 96.6 96.8 97.0 97.1 0.0 87.5 93.0 94.8 95.7 96.3 96.6 96.9 97.0 97.1 97.2 86.7 92.6 94.5 95.5 96.1 96.4 96.7 96.9 97.0 97.1 83.4 92.2 93.9 95.4 95.8 96.3 96.6 96.7 97.0 97.0 84.1 90.6 93.8 95.0 95.7 96.2 96.5 96.7 96.8 96.9 83.2 90.8 93.4 94.7 95.5 96.0 96.3 96.6 96.7 96.9 81.9 90.2 93.0 94.4 95.3 95.8 96.2 96.5 96.7 96.8 80.7 89.6 92.7 94.2 95.1 95.7 96.2 96.4 96.7 96.8 78.8 88.7 92.1 93.8 94.8 95.5 95.9 96.2 96.5 96.6]
GSel_EffTrsmGear7_M	Transmission efficiency for gear 7	[89.4 93.4 95.3 96.1 96.6 96.9 97.0 0.0 0.0 0.0 88.4 93.1 95.0 95.8 96.4 96.6 96.8 97.0 97.1 0.0 87.2 92.8 94.7 95.6 96.1 96.5 96.8 96.9 97.0 97.1 85.8 92.1 94.2 95.3 96.0 96.4 96.7 96.9 97.1 97.2 84.0 88.7 93.0 94.8 95.5 96.1 96.4 96.7 96.8 96.9 83.5 90.7 93.4 94.8 95.5 96.0 96.4 96.6 96.7 96.9 82.3 90.2 93.1 94.5 95.4 95.8 96.2 96.5 96.8 96.9 80.7 89.7 92.7 94.1 95.1 95.7 96.1 96.4 96.6 96.8 79.4 88.8 92.2 93.9 94.9 95.5 96.0 96.3 96.5 96.7]

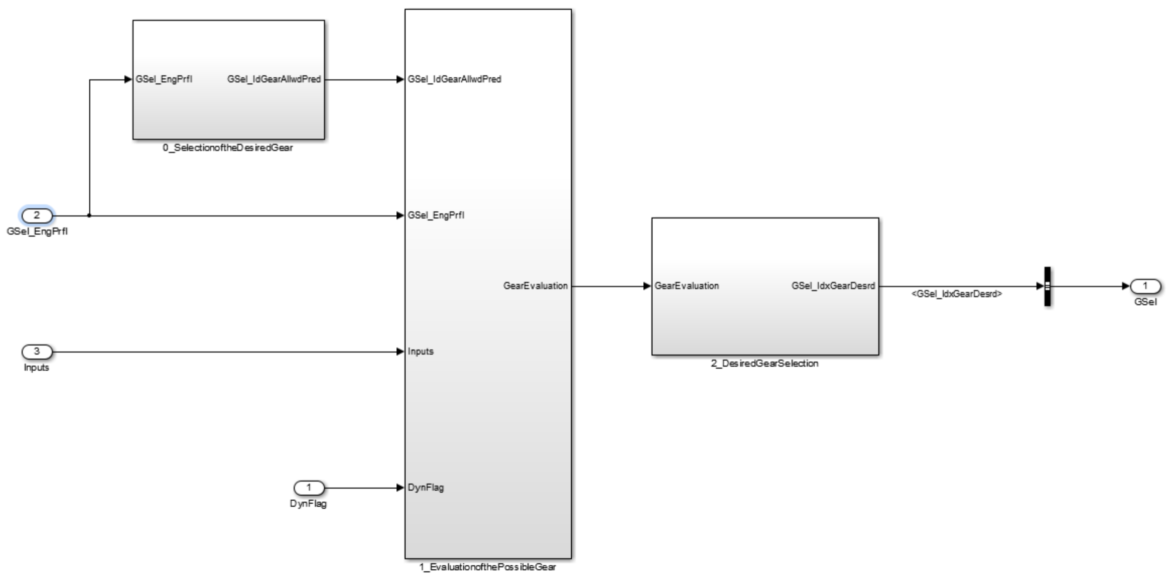


Figure 3.16: Calculation of Allowed Gear in Matlab/Simulink Environment

3.4 4_AllowedGearCalculation

After the definiton of engine torque and speed according to vehicle velocity and acceleration profile, allowed gear according to these values are calculated under this subsystem. First selection will be done in subsystem 0_SelectionoftheDesiredGear according to engine limitations. Then, in subsystem 1_EvaluationofthePossibleGear, selected gears are evaluated. Finally, desired gear is defined under subsystem 2_DesiredGearSelection.

3.4.1 0_SelectionoftheDesiredGear

First selection of the desired gear/s are evaluated under this subsystem by matlab function, Figure 3.17. Inputs/Outputs description can be seen in Table 3.20. Parameter descriptions can be seen in Table 3.19.

Firstly, synchronized engine speed matrix for each gear is evaluated with respect to engine speed limitations and 'bEnaEngSpeed' boolean matrix, which each columns represent gear id and rows are value for each sample time. If 4. gear is allowed, 1 is assigned to fourth column in respective row. The same operation is repeated for the engine torques and 'bEnaEngTq' boolean matrix is calculated. Selection gear will be the intersection of these matrix.


```
%% Engine speed comparison
bEnaEngSpd = ones(length(IN_TqEngMax),7);
for i = 1:length(IN_TqEngMax)
    bEnaEngSpd(i,1:7) = (IN_NEngSync(i,1:7)<GSel_NEngMaxInGear_P) & (IN_NEngSync(i,1:7)>GSel_NEngMinInGear_P);
end
%% Engine torque comparison
bEnaEngTq = ones(length(IN_TqEngMax),7);
for i = 1:length(IN_TqEngMax)
    bEnaEngTq(i,1:7) = (IN_TqEngTgtPrfl(i,1:7)<IN_TqEngMax(i,1:7)) & (IN_TqEngTgtPrfl(i,1:7)>IN_TqEngMin(i,1:7));
end
bEnaGear = bEnaEngSpd& bEnaEngTq;
%% Modify selected gear (bEnaGear) with respect to TqEngTgtPrfl vs TqEngTgtMax so that look only the gear
TqEngMax_Modified = ones(length(IN_TqEngMax),7);
for i = 1:length(IN_TqEngMax)
    if all(IN_TqEngMax(i,1:7)==0) %standstill
        bEnaGear(i,1:7) = [1 0 0 0 0 0 0];
    elseif all(bEnaGear(i,1:7)==0) %engine torque cannot be satisfied, select the gear which has the highest engine torque
        TqEngMax_Modified(i,1:7) = IN_TqEngMax(i,1:7).*bEnaEngSpd(i,1:7);
        MaxTqId = find((TqEngMax_Modified(i,1:7)- IN_TqEngMax(i,1:7))==0);
        for j = 1:length(MaxTqId)
            bEnaGear(i,MaxTqId(j)) = 1;
        end
    else
        bEnaGear(i,1:7) = bEnaGear(i,1:7);
    end
end

%% Assign the IN_IdGearAllowed signal
[row,column] = find(bEnaGear==1);
IN_IdGearAllowed = zeros(length(bEnaGear),7);
for i = 1:length(column)
    IN_IdGearAllowed(row(i),column(i)) = bEnaGear(row(i),column(i));
end
```

Figure 3.17: Calculation of Desired Gear Selection in Matlab/Simulink Environment

It should be highlighted that bEnaEngSpeed assigns at least one gear for the calculated engine synchronized speeds, in other words, there will be always at least one gear selected. But this is not valid for bEnaEngTq. Since engine torque target can be calculated such that it is not possible to fulfill the output shaft torque request. So that, bEnaEngTq matrix row/s may all results as 0. In this case, the gear which gives the highest engine torque within the engine speed limits will be selected as an allowed gear.

In the end, selected gear matrix for each time step, for each gear 'bEnaGear' is assigned to the output signal IN_IdGearAllowed.

Table 3.19: Calibrateable parameters used in 0_SelectionoftheDesiredGear subsystem

GSel_NEngMaxInGear_P	Maximum engine speed for each gear in *rpm*	[6300 6300 6300 6300 6300 6300 6300]
GSel_NEngMinInGear_P	Minimum engine speed for each gear in *rpm*	[-1000 1200 1200 1200 1200 1200 1200]

Table 3.20: Input/Output signals used in 0_SelectionoftheDesiredGear subsystem

Signal name	Description
simtime	simulation time in 10 ms
GSel_NEngSyncPrfl	Synchronized engine speed profile for each gear in parameterizable time interval in rpm. It is a matrix, which each column represents gear id and rows are value for each sample time
GSel_TqEngMin	Minimum engine torque profile for each gear in parameterizable time interval in Nm. It is a matrix, which each column represents gear id and rows are value for each sample time
GSel_TqEngMax	Maximum engine torque profile for each gear in parameterizable time interval in Nm, It is a matrix, which each column represents gear id and rows are value for each sample time
GSel_TqEngTgtPrfl	Engine torque target profile for each gear in parameterizable time interval in Nm. It is a matrix, which each column represents gear id and rows are value for each sample time
GSel_IdGearAllwdPred	Allowed gear with respect to engine allowances in parameterizable time interval. It is a matrix, which each column represents gear id and rows are value for each sample time

3.4.2 1_EvaluationofthePossibleGear

Under this subsystem, Figure 3.18, selected gears are evaluated.

- 0_UpshiftProhibition: Dynamic changes flags are evaluated according to upshift allowances.
- 1_DownshiftProhibition: Dynamic changes flags are evaluated according to downshift allowances.
- 2_NVH: NVH behavior of the engine is evaluated with respect to gear.
- 3_EngineEfficiencyCheck: Engine efficiency is evaluated with respect to gear.
- 4_ClutchHeatDissipationCheck: Clutch heat dissipation during the possible shifts are evaluated.
- 5_FutureGearTrendCheck: Future gear trend is evaluated and selected gear is optimized accordingly.

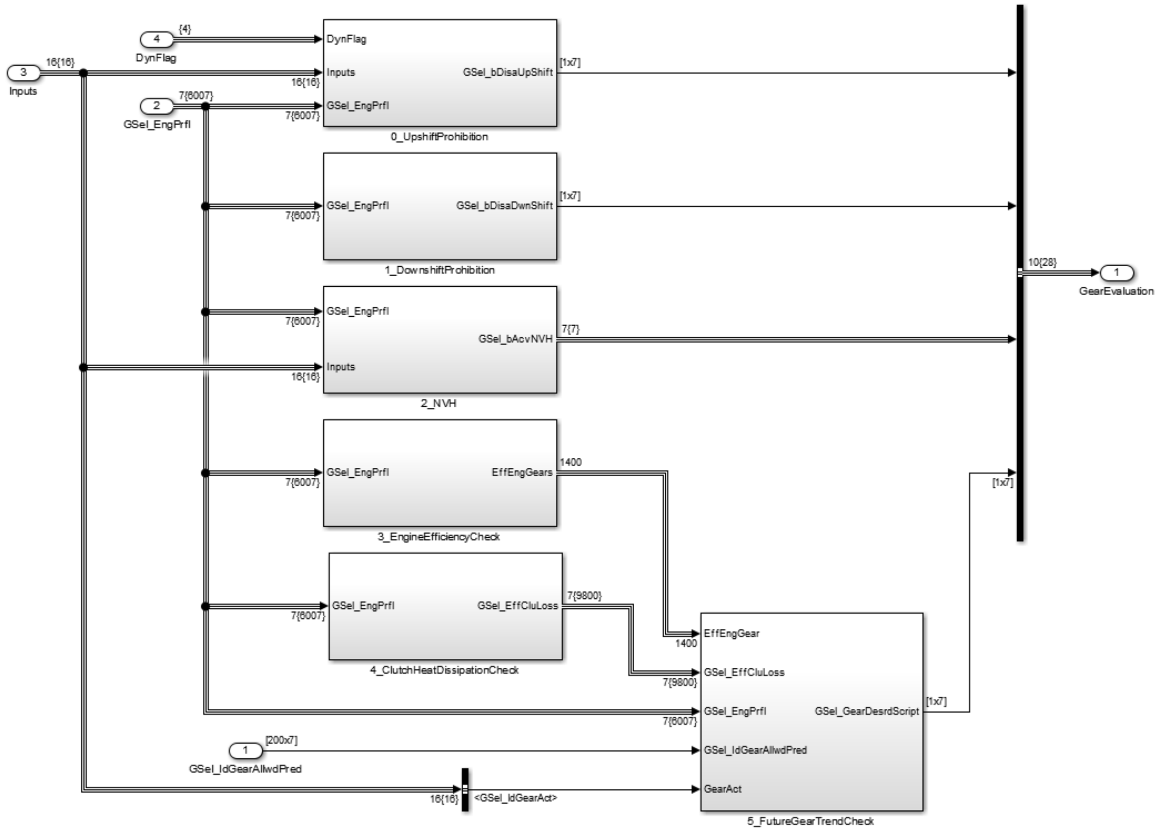


Figure 3.18: Calculation of Selected Gear Evaluation in Matlab/Simulink Environment

3.4.2.1 0_UpshiftProhibition :

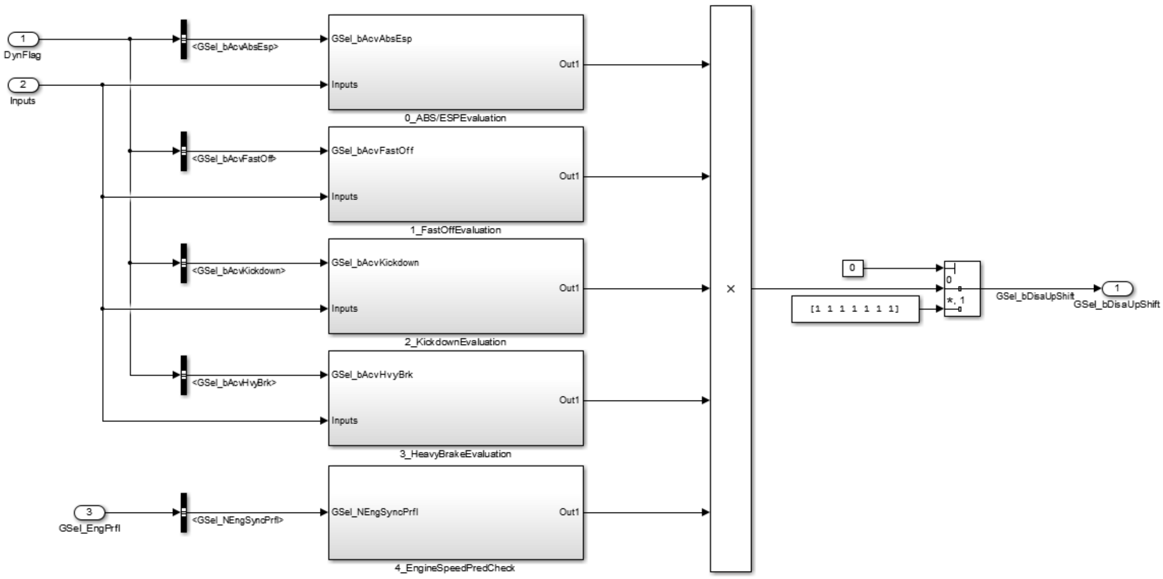


Figure 3.19: Calculation of Upshift Disable Evaluation in Matlab/Simulink Environment

Upshift prohibition flag according to dynamic changes flag is calculated under this subsystem, Figure 3.19. Output signal description can be seen in Table 3.21. Subsystems can be descriped shortly as the following. Detailed explanations will be provided in the coming pages.

- 0_ABS/ESPEvaluation: ABS/ESP dynamic flag is evaluated according to upshift allowances.
- 1_FastOffEvaluation: Fast off dynamic flag is evaluated according to upshift allowances.
- 2_KickdownEvaluation: Kickdown dynamic flag is evaluated according to upshift allowances.
- 3_HeavyBrakeEvaluation: Heavy brake dynamic flag is evaluated according to upshift allowances.
- 4_EngineSpeedPredCheck: Engine predicted speed allowances is evaluated according to upshift allowances.

Table 3.21: Input/Output signals used in 0_UpshiftProhibition subsystem

Signal name	**Description**
GSel_bDisaUpShift	Enabled/Disabled gears with respect to upshift prohibition. If 1 is assigned, gear is allowed, otherwise not allowed.

3.4.2.1.1 0_ABS/ESPEvaluation

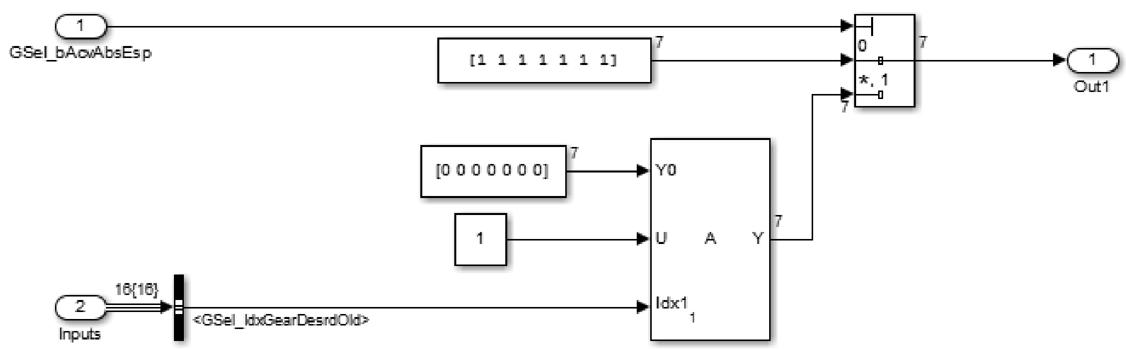


Figure 3.20: Evaluation of the ABS/ESP Flag

If ABS/ESP dynamic changes flag is activated, upshift is not allowed and gears from the current desired gear are disabled until 7.gear. This logic may change depending on ESP interfaces. In the current simulation, ESP interfaces deactivates the upshift, that's why, it is implemented like this. Inputs description can be seen in Table 3.22

Table 3.22: Input/Output signals used in 0_ABS/ESPEvaluation subsystem

Signal name	Description
GSel_bAcvAbsEsp	ABS/ESP activated flag, calculated in 1_DynamicChangesEvaluation subsystem
GSel_IdxGearDesrdOld	Calculated desired gear feedback signal

3.4.2.1.2 1_FastOffEvaluation

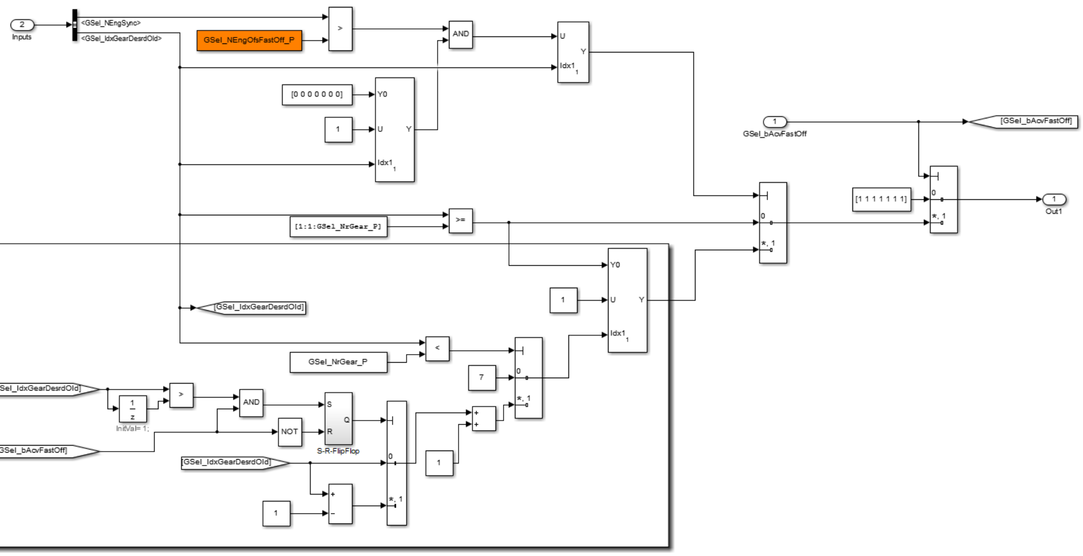


Figure 3.21: Evaluation of the Fast Off Flag

Input signals description can be seen in Table 3.23. If synchronized engine speed is higher than a parameterizable value, see Table 3.24, gears will be allowed until actual gear and one gear upper, to be able to satisfy engine speed request. Otherwise, gears will be allowed until actual one.

Bordered simulink block calculation will be sent to the output incase, fast off flag is activated and synchronized engine speed is greater than a parameterizable value. This calculation also make sure that during the fast off only one uphift happens. For example, if desired gear is two gear upshift according to old desired gear, desired gear id will be decreased one gear down.

Table 3.23: Input/Output signals used in 1_FastOffEvaluation subsystem

Signal name	Description
GSel_bAcvFastOff	Fast off activated flag, calculated in 1_DynamicChanges-Evaluation subsystem
GSel_IdxGearDesrdOld	Calculated desired gear feedback signal
GSel_NEngSync	Synchronized engine speed for each gear in *rpm*

Table 3.24: Calibrateable parameters used in 1_FastOffEvaluation subsystem

Parameter name	**Description**	**Value**
GSel_NEngOfsFastOff_P	Engine speed offset for fast of evaluation in *rpm*	3500

3.4.2.1.3 2_KickdownEvaluation

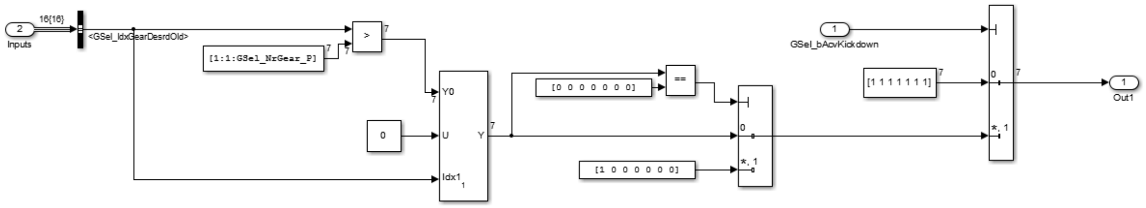


Figure 3.22: Evaluation of the Kickdown Flag

Under this logic, upper gears with actual gear are disabled to force the software downshift. Since in the high torque demands, engine speed should be kept high with a low gear selection. There is also precaution incase of an old desired gear is 1. In this case, '>' relational operation will be false for all gears, then output of the Assignment block [0 0 0 0 0 0 0] will be replaced by a vector [1 0 0 0 0 0 0], which means, only 1 is allowed.

Inputs description can be seen in Table 3.25

Table 3.25: Input/Output signals used in 2_KickdownEvaluation subsystem

Signal name	**Description**
GSel_bAcvKickdown	Kickdown activated flag, calculated in 1_DynamicChanges-Evaluation subsystem
GSel_IdxGearDesrdOld	Calculated desired gear feedback signal

3.4.2.1.4 3_HeavyBrakeEvaluation

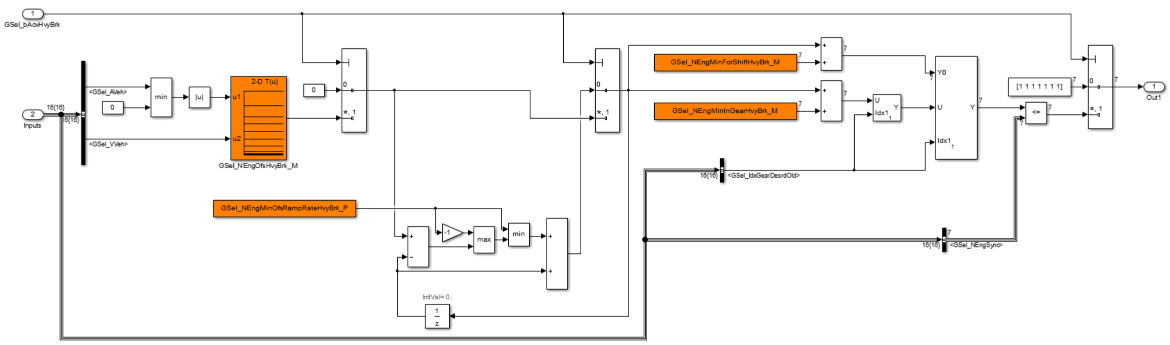


Figure 3.23: Evaluation of the Heavy Brake Flag

Heavy brake pedal state evaluation is handled under this component, see Figure 3.23. Firstly minimum engine offset speed is calculated by a parameterizable map with respect to vehicle velocity and deceleration of a vehicle. The map and axes values can be seen in Table 3.26. This offset is summed by the parameterizable values for engine minimum speed for shifting and for driving. Element, which id is old desired gear in vector goes to Y0 input is replaced by a value of U, which is the minimum engine value for the driven gear. The updated engine minimum parameterizable vector is compared with synchronized engine speed. If the parameterizable vector element is larger than a synchronized engine speed, the gear, which has this element id is disabled. Since it means, synchronized engine speed cannot satisfy the driving with these gears. If heavy brake flag is not activated, all the gears are allowed by giving all 1, 7 element vector to the output.

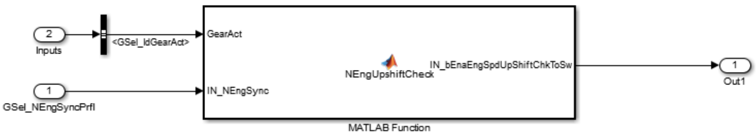
Table 3.26: Calibrateable parameters used in 3_HeavyBrakeEvaluation subsystem

Parameter name	**Description**	**Value**
GSel_VVehHvyBrk_A	Vehicle velocity supporting points for heavy brake situation evaluation in km/h	[0 20 35 50 70 85 100 110 130 145 160 190 220 250]
GSel_AVehHvyBrk_A	Vehicle acceleration supporting points for heavy brake situation evaluation	[1 1.5 2 2.5 3 3.5 4 5 6 7 8 10 12 14]
GSel_NEngOfsHvyBrk_M	Minimum engine speed offset over absolute vehicle deceleration and vehicle velocity when brake is applied	-
GSel_NEngMinOfsRamp-RateHvyBrk_P	Ramp of minimum engine speed offset over absolute vehicle deceleration and vehicle velocity when brake is released:ramp of engine rotational speed offset in rpm	25
GSel_NEngMinOfsRamp-LimnHvyBrk_P	Ramp of minimum engine speed offset over absolute vehicle deceleration and vehicle velocity when brake is released:ramp of engine rotational speed offset in rpm	10000
GSel_NEngMinInGear-HvyBrk_M	Heavy brake minimum engine rotational speed in each gear in rpm	[1400 1400 1440 1600 1650 1700 1700]
GSel_NEngMinForShift-HvyBrk_M	Heavy brake minimum engine rotational speed to shift into gear in rpm	[1200 1300 1400 1450 1500 1550 1600]

Table 3.27: Input/Output signals used in 3_HeavyBrakeEvaluation subsystem

Signal name	Description
GSel_bAcvHvyBrk	Heavy Brake activated flag, calculated in 1_DynamicChanges-Evaluation subsystem
GSel_IdxGearDesrdOld	Calculated desired gear feedback signal
GSel_NEngSync	Synchronized engine speed for each gear in *rpm*
GSel_AVeh	Vehicle acceleration in m/s^2
GSel_VVeh	Vehicle velocity in m/s

3.4.2.1.5 4_EngineSpeedPredCheck




```
%% Upshift/Downshift allowance check according to predictive engine speeds
GSel_NEngMaxInGear_P = [6300 6300 6300 6300 6300 6300 6300]; %rpm'.'maximum engine speed in each gear (1 to 7)';
GSel_TiUpShift_P = 100; % put 1sec according to VSM model; %unit: ms
IN_bEnaEngSpdUpShiftChk = ones(length(IN_NEngSync),7);
if GearAct ~= 7
    for i = 1:length(IN_NEngSync)-GSel_TiUpShift_P
        IN_bEnaEngSpdUpShiftChk(i,1:7) = IN_NEngSync(i+GSel_TiUpShift_P,1:7)<GSel_NEngMaxInGear_P;
    end
end
IN_bEnaEngSpdUpShiftChkToSw = IN_bEnaEngSpdUpShiftChk(1,1:7);
```

Figure 3.24: Evaluation of the Engine Speed Prediction for Upshift Prohibition

Engine speed limit are checked for the current gear after a parameterizable shifting time, see Figure 3.24. If after a parameterizable shifting time, engine speed will still allow the gear/s that are selected in present, there will not be change in the selected gears. If engine speed does not allow after a parameterizable shifting time the current selected gear/s, since engine maximum limit is violated, then the violating gear should be disabled.

This check is executed, when actual gear is not equal to 7, since otherwise it will not be logical. There is not a higher gear than 7.

Used parameters and input/output signals can be seen from the Table 3.29 and Table 3.28

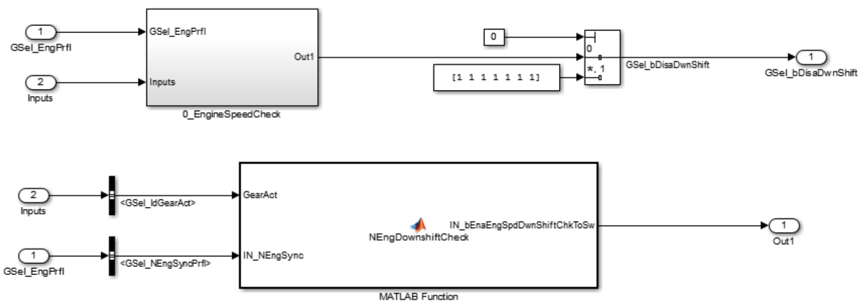
Table 3.28: Calibrateable parameters used in 4_EngineSpeedPredCheck subsystem

Parameter name	Description	Value
GSel_NEngMaxInGear_P	Maximum engine speed in each gear in *rpm*	[6300 6300 6300 6300 6300 6300 6300]
GSel_TiUpShift_P	Calibrateable upshift time 10*ms*	30

Table 3.29: Input/Output signals used in 4_EngineSpeedPredCheck subsystem

Signal name	**Description**
GSel_NEngSync	Synchronized engine speed profile for each gear in parameterizable time interval in *rpm*. It is a matrix , which each column represents gear id and rows are value for each sample time
IN_bEnaEngSpdUp-ShiftChkToSw	Engine speed check after a parameterizable upshift time for checking, if the selected gear is still in the engine speed limitations. It is a matrix by [1,7]

3.4.2.2 1_DownshiftProhibition :




```
%% Upshift/Downshift allowance check according to predictive engine speeds
GSel_NEngMinInGear_P = [-1000 1200 1200 1200 1200 1200 1200]; %'rpm'.'minimum engine speed in each gear (1 to 7)';
GSel_TiDownShift_P = 200; % put 2sec according to VSM; %unit: 10ms
IN_bEnaEngSpdDwnShiftChk = ones(length(IN_NEngSync),7);
if GearAct ~= 1
    for i = 1:length(IN_NEngSync)-GSel_TiDownShift_P
        IN_bEnaEngSpdDwnShiftChk(i,1:7) = IN_NEngSync(i+GSel_TiDownShift_P,1:7)>GSel_NEngMinInGear_P;
    end
end
IN_bEnaEngSpdDwnShiftChkToSw = IN_bEnaEngSpdDwnShiftChk(1,1:7);
```

Figure 3.25: Calculation of Downshift Disable Evaluation in Matlab/Simulink Environment

Engine speed limit are checked for the current gear after a parameterizable shifting time is evaluated, see Figure 3.25. If after a parameterizable shifting time, engine speed will still allow the gear/s that are selected in present, there will not be change in the selected gears. If engine speed does not allow after a parameterizable shifting time the current selected gear/s, since engine minimum limit is violated, then the violating gear should be disabled.

This check is executed, when actual gear is not equal to 1, since otherwise it will not be logical. There is not a lower gear than 1.

Used input/output signals and parameters can be seen from the Table 3.29 and Table 3.31

Table 3.30: Input/Output signals used in 1_DownshiftProhibition subsystem

Signal name	Description
GSel_NEngSync	Synchronized engine speed profile for each gear in parameterizable time interval in *rpm*. It is a matrix, which each column represents gear id and rows are value for each time
IN_bEnaEngSpdDown-ShiftChkToSw	Engine speed check after a parameterizable upshift time for checking if the selected gear is still allowed in engine speed allowances point of view. It is a matrix by [1,7]
GSel_bDisaDwnShift	Enabled/Disabled gears with respect to downshift prohibition, calculated in 1_DownshiftProhibition subsystem

Table 3.31: Calibrateable parameters used in 1_DownshiftProhibition subsystem

Parameter name	Description	Value
GSel_NEngMinInGear_P	Minimum engine speed in each gear in *rpm*	[-1000 1200 1200 1200 1200 1200 1200]
GSel_TiDownShift_P	Calibrateable upshift time 10*ms*	30

3.4.2.3 2_NVH :

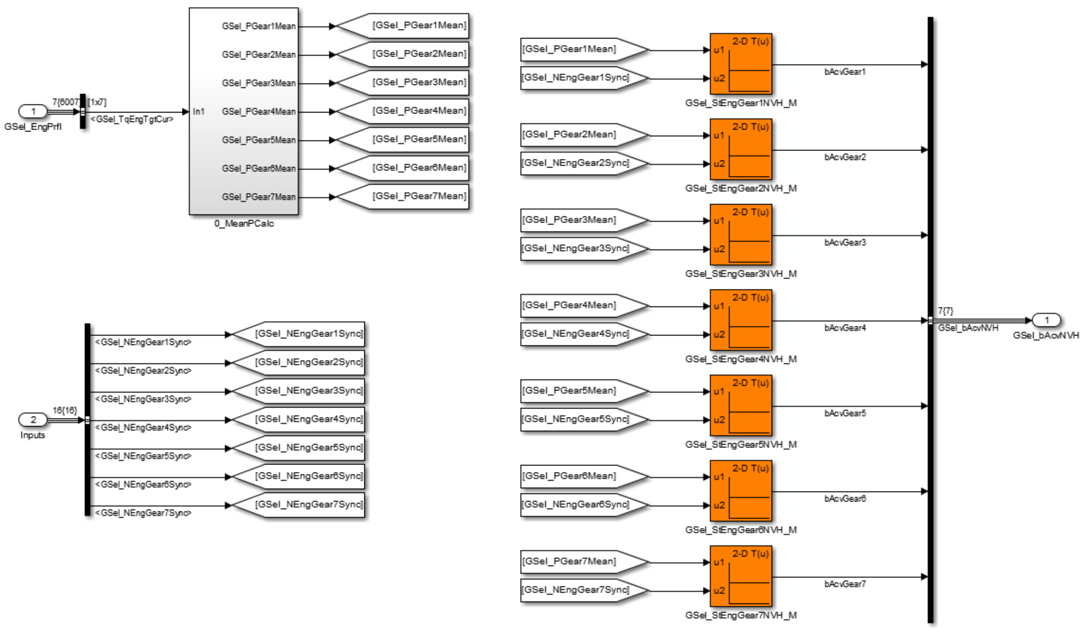


Figure 3.26: Calculation of NVH in Matlab/Simulink Environment

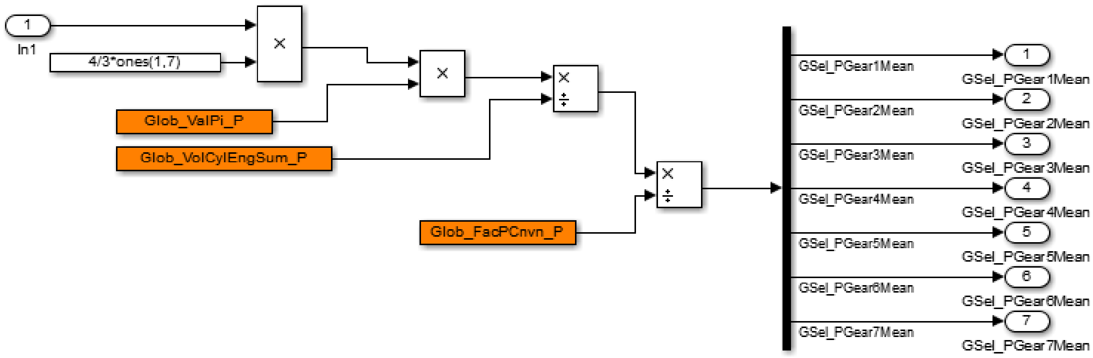


Figure 3.27: Calculation of Mean Effective Pressure in Matlab/Simulink Environment

Under this subsystem, NVH behavior of the engine is calculated for each gear and current data by a parameterizable map. The output 1 of the map means gear is allowed for the current mean effective pressure and engine synchronized speed. The output 0 of the map means gear is not allowed for the current mean effective pressure and engine synchronized speed. Inputs definitions can be seen in Table 3.32. Parameter values and explanations can be seen in Table 3.33.

Mean effective pressure calculation to be sent to the first input of the map can be seen in Figure 3.27. Calculation was implemented based on the following equation 3.1, where P_m is mean effective pressure, M_{Eng} is engine torque, V_{cyl} is volume of a piston cylinder.

$$P_m = \frac{4 * M_{Eng} * \pi}{3 * V_{cyl}} \tag{3.1}$$

This calculation handled by a vector, which each element represents value for one gear. In first operation, engine target torque vector is divided by value 3 in vector form which is the number of cylinder in the engine and it is multiplied with 4 since it is a four stroke engine. By the parameter 'Glob_FacPCnvn_P', unit conversion from *bar* to N/m^2 is implemented. In the end, vector is splitted to his elements by 'Demux' simulink block. Namely, calculated values for each gear is sent seperately to be able to evaluated in the maps.

It should be highlighted that related NVH data cannot be found for the current project, thats why all maps are parameterized by value 1, namely all gears are allowed.

Table 3.32: Input/Output signals used in 2_NVH subsystem

Signal name	Description
GSel_NEngGearXSync	Engine synchronized speed for gear X in *rpm*
GSel_PGearXMean	Mean effective pressure for gear X in *bar*
GSel_TqEngTgtCur	Engine torque target for the current data for each gear in Nm.

Table 3.33: Calibrateable parameters used in 2_NVH subsystem

Parameter name	Description	Value
Glob_ValPi_P	Value pi in vector form for each gear value	3,141592654
Glob_FacPCnvn_P	Conversion factor from bar to N/m^2 in vector form for each gear value	1e5
Glob_VolCylEngSum_P	Engine cylinder capacity in vector form for each gear value in m^3	0,00039942
GSel_PEngForNVH_A	Mean effective pressure supporting points for NVH table in *bar*	[...]
GSel_NEngForNVH_A	Engine rotational speed supporting points for NVH table in *rpm*	[...]
GSel_StEngGearX-NVH_M	Possible areas of mean effective pressure and rotational speed according to NVH behavior for gear X	[...]

3.4.2.4 3_EngineEfficiencyCheck :

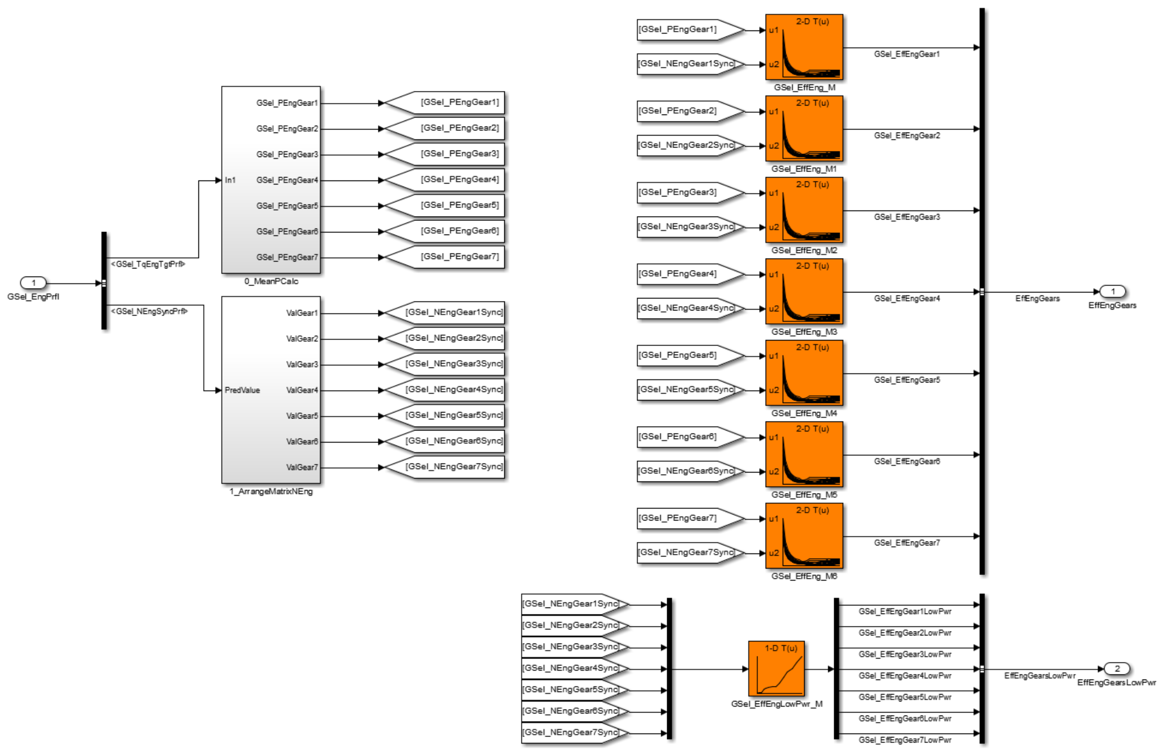


Figure 3.28: Calculation of Engine Efficiencies in Matlab/Simulink Environment

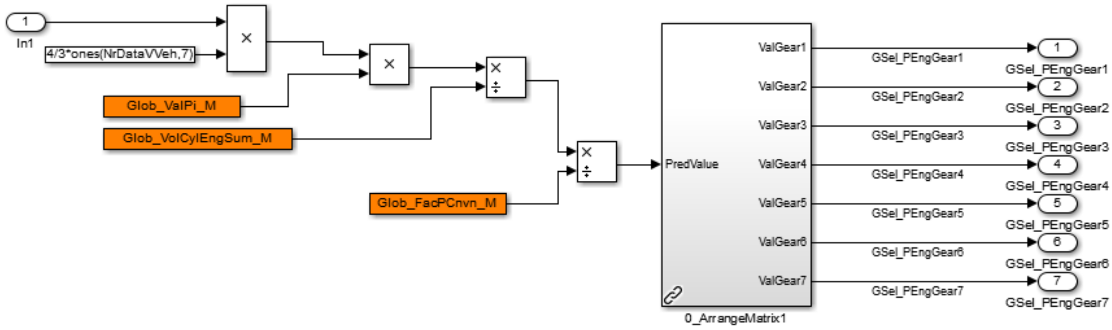


Figure 3.29: Calculation of Mean Effective Pressure in Matlab/Simulink Environment

Under this subsystem, engine efficiency for each gear is calculated, see Figure 3.28. Inputs definitions can be seen in Table 3.34. Parameter values and explanations can be seen in Table 3.35.

Calibrateable engine efficiency map is used with constraints engine speed and mean effective pressure in the piston cylinder. Maps output is a fuel consumption in g/kWh. Mean effective pressure calculation to be sent to the first input of the map can be seen in Figure 3.29. Calculation was implemented based on the following equation 3.2, where P_m is mean effective pressure, M_{Eng} is engine torque, V_{cyl} is volume of a piston cylinder..

$$P_m = \frac{4 * M_{Eng} * \pi}{3 * V_{cyl}} \tag{3.2}$$

This calculation handled by a matrix, which each column represents gear id and rows are value for each sample time. In first operation, engine target torque matrix is multiplied by value 4 in matrix form which is the engine stroke and it is divided by 3, which is the number of cylinder. By the parameter 'Glob_FacPCnvn_M', unit conversion from bar to N/m^2 is implemented. Afterwards, matrix is splitted to his columns by a library 'ArrangeMatrix, Figure B.4'. Namely, calculated values for each gear is sent as a vector seperately, to be able to evaluated in the maps. From Figure 3.28, same split operation can be seen for the synchronized engine speeds as well.

GSel_EffEngLowPwr_M map is used for the low engine power and when engine decelerates, namely when engine torque has a negative value. In this time, only engine speed is used as a map input.

Table 3.34: Input/Output signals used in 3_EngineEfficiencyCheck subsystem

Signal name	Description
GSel_NEngSyncPrfl	Synchronized engine speed profile for each gear in parameterizable time interval in rpm. It is a matrix, which each column represents gear id and rows are value for each sample time
GSel_TqEngTgtPrfl	Engine torque target profile for each gear in parameterizable time interval in Nm. It is a matrix, which each column represents gear id and rows are value for each sample time

Table 3.35: Calibrateable parameters used in 3_EngineEfficiencyCheck subsystem

Parameter name	Description	Value
Glob_ValPi_M	Value pi in matrix form	3,141592654
Glob_VolCylEng-Sum_M	Engine cylinder capacity in matrix form in m^3	3,141592654
Glob_FacPCnvn_M	Conversion factor from bar to N/m^2 in matrix form	1e5
GSel_PEngForEng-Eff_A	Mean effective pressure for engine efficiency map in bar	[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22]
GSel_NEngForEng-Eff_A	Engine speed for engine efficiency map in rpm	[800 1000 1250 1500 1750 2000 2400 2800 3200 3600 4000 4500 5000 5500]
GSel_EffEngLow-Pwr_M	Engine BSFC efficiency map in g/kWh in low power	[560 562 565 575 580 582 585 587 600 620 640 655 675 700]

Continued on next page

Table 3.35 – continued from previous page

Parameter name	Description	Value
GSel_EffEng_M	Engine BSFC efficiency map in g/kWh	[628.2 594.1 574.8 578.2 549.7 582.1 635.1 ... 587.2 640.6 653.6 637.4 672.1 730.4 801.8 423.1 402.7109 405.0 374.6 378.81 387.9228 402.6616 ... 409.2663 417.9404 431.7763 450.0657 468.5180 473.2 525.5 354.4682 338.5840 328.9000 322.3233 318.3098 322.7262 332.3945 ... 339.8782 349.9596 359.8806 371.6048 385.2477 400.4624 421.1 319.0462 307.0821 292.8286 290.3292 289.2902 291.9550 302.2127 ... 307.4907 314.5663 323.2909 331.8385 342.6308 352.9194 367.0000 292.4143 285.7394 276.8647 272.9913 274.1029 275.4505 282.8286 ... 287.4059 293.7475 299.9228 305.6846 314.1546 324.6025 336.8165 283.6010 272.0000 264.7317 262.0757 261.5000 264.8495 270.3643 ... 274.1971 278.8086 282.9930 288.3252 294.3245 303.5192 315.4509 279.5524 266.2857 255.7973 254.5982 254.1604 256.7519 260.7396 ... 264.1453 266.5229 271.1750 275.5272 280.5509 288.7270 300.5453 276.3316 265.4673 255.7064 250.5811 251.3676 251.9406 255.6389 ... 258.6736 259.5757 262.8696 265.8761 270.4990 277.9162 286.7257 280.0158 266.9583 255.5082 248.8807 247.7220 248.9857 251.2870 ... 253.7000 254.6854 256.5623 258.0000 263.6578 269.2288 279.9382 298.6364 274.2226 255.2204 248.7053 246.4340 245.9306 248.3491 ... 250.5224 251.1894 252.7887 254.3434 258.1115 265.3550 273.7618 314.6500 294.3965 267.2075 248.1165 245.2122 243.6154 244.6154 ... 246.1424 247.8333 249.3385 250.4292 253.6538 259.0991 270.7212 314.6500 297.4315 264.9811 252.0000 244.0835 241.2130 240.7170 ... 242.3288 244.6796 246.4325 248.1019 250.2649 255.6187 269.6264 314.6500 298.6551 278.0513 252.2970 243.9000 239.1600 238.1532 ... 239.9000 241.5315 243.6392 245.9925 248.2981 257.3375 270.2876 314.6500 298.6551 276.3991 258.1296 245.0321 239.3839 236.7489 ... 237.1807 238.8661 241.2864 244.2000 246.1000 254.9992 273.6129 314.6500 298.6551 282.5040 263.5000 244.7350 239.2380 237.5554 ... 236.6263 237.6217 239.2845 242.7014 246.3019 257.0510 278.5957 314.6500 298.6551 286.9164 270.9042 247.1066 241.0747 238.6146 ... 239.1375 237.8875 239.1495 241.4685 247.7648 261.6514 285.5485 314.6500 298.6551 285.4500 271.3244 251.9126 246.5037 239.7718 ... 238.3302 238.4708 240.2870 242.0614 250.1980 268.2245 295.4000 314.6500 298.6551 287.9500 275.5000 263.1739 256.0975 245.5552 ... 239.8396 240.7870 244.9472 245.1748 254.1182 280.6774 319.3000 314.6500 298.6551 287.9500 276.9443 266.8979 265.9195 249.5214 ... 241.5615 242.5500 249.5072 252.6505 270.4552 309.6217 319.3000 314.6500 298.6551 287.9500 284.8212 270.4452 266.9125 252.9638 ... 245.8306 245.6733 254.1217 267.0545 286.3752 309.6217 319.3000 314.6500 298.6551 287.9500 286.7621 275.3122 270.8512 261.8833 ... 249.2955 256.5429 266.4539 288.9988 299.4636 309.6217 319.3000 314.6500 298.6551 287.9500 286.7621 277.0000 272.8024 261.8833 ... 249.2955 256.5429 266.4539 288.9988 299.4636 309.6217 319.3000]

3.4.2.5 4_ClutchHeatDissipationCheck :

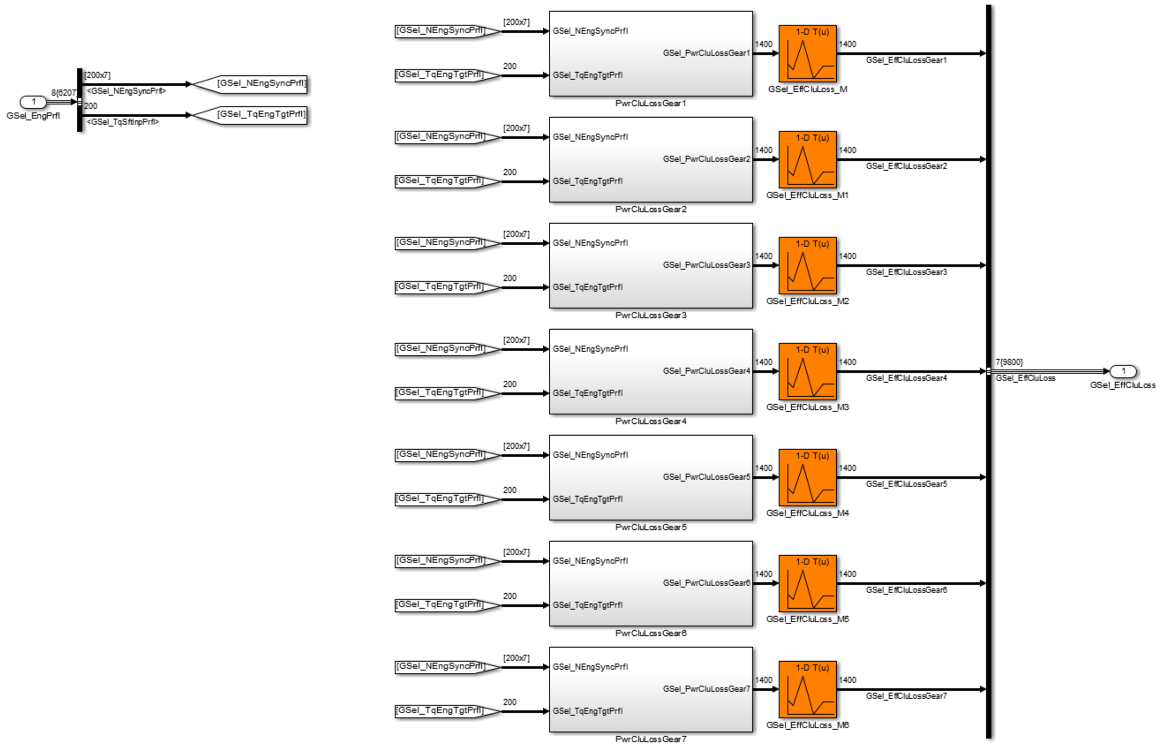


Figure 3.30: Calculation of Clutch Heat Dissipation in Matlab/Simulink Environment

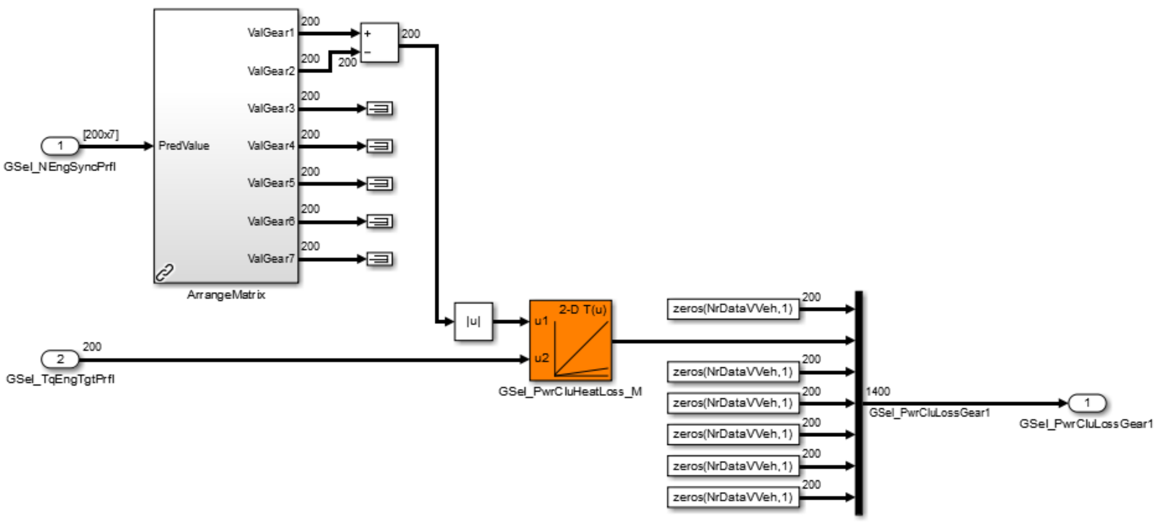


Figure 3.31: Calculation of power loss in clutches during shift in Matlab/Simulink Environment

Theoretical background was introduced in chapter '2.Method Description' for the power losses in the clutch during the shift. In this subsystem, according to this theoretical background some assumptions were made and maps were created. Maps are linear, where minimum and maximum clutch torques are taken as 50 and 350Nm, minimum and maximum clutch slips are taken as 50 and 1000rpm and shifting time is taken as 500ms. Clutch slip is defined by the speed difference between each side of the clutch. As it can be seen in Figure 3.31, clutch slip shifting from gear 1 to gear 2 is calculated by the engine synchronized speed difference. This figure shows only one calculation for shifting from gear 1 to gear 2. This calculation is repeated for the each possible shifts. In this software, upshift is allowed sequentially, where downshift can be handled by skipping the gears. For example, upshift from gear 5 will be allowed only to gear 6, whereas downshift from gear 5 can be allowed to 4. gear as well as 3, 2 or 1. Odd gear to odd gear or even gear to even gear shift is harder than odd to even or vice versa shift according to control point of view. Although shifting takes longer and hard to controlling clutches and rails, it is possible.

During driving, there will be no clutch slip, namely speed differences between the each side of the clutch will be zero. If the shift is requested, engine speed should be changed from engine synchronized speed at 1. gear to engine synchronized speed at 2. gear, which creates clutch slip. In Figure 3.31, input engine synchronized matrix is splitted to its column to get the value for each gear seperately by library ArrangeMatrix, see Figure B.4. After the calculation of clutch slip by substraction the speeds, absolute value is taken. Since negative or positive sign in the clutch slip just represents, which side has the highest speed value. But this information is not required during power loss calculation. The important thing is magnitude of speed difference in each side of the clutch.

Output of the map in Figure 3.31 is in unit $Watt$. To be able to compare this result with engine efficiencies in g/kWh in the coming subsystem 5_FutureGearTrendCheck, the parameterizable conversion factor is implemented by a map before sending to the output, which can be seen in Figure 3.30. This map is calibrated according to simulation measurements. Fuel consumption is measured for a specific engine power and approximate value is assigned to the map.

Detailed explanations of the each signals and parameters used in this subsystem can be seen in Table 3.36 and 3.37 respectively.

Table 3.36: Input/Output signals used in 4_ClutchHeatDissipationCheck subsystem

Signal name	Description
GSel_NEngSyncPrfl	Synchronized engine speed profile for each gear in parameterizable time interval in *rpm*. It is a matrix, which each column represents gear id and rows are value for each sample time
GSel_TqSftInpPrfl	Engine torque target profile in parameterizable time interval in *Nm*. It is a vector, which elements represent the value for each sample time
GSel_EffCluLossGearX	Clutch calculated power loss during shifting from gear X to other 7 gears, it is a vector with a number of gear, 7 times predicted time span.

Table 3.37: Calibrateable parameters used in 4_ClutchHeatDissipationCheck subsystem

Parameter name	Description	Value
GSel_DifSyncdEngSpd_A	Synchronized engine speed difference axis between current gear and allowed shifting gear in *rpm*	[50 2000]
GSel_TqSftInp_A	Current input shaft torque axes for clutch heat loss in *Nm*	[50 350]
GSel_PwrCluHeatLoss_M	Factor for comparing the power loss in the clutch during shift because of the heat dissipation in *Watt*	[65,4498 458,1489 2618,0 18326]
GSel_PwrCluLoss_A	Clutch heat dissipation power to engine efficiency conversion factor axes of the map	[5 10 20 30 40 50]
GSel_EffCluLoss_M	Clutch heat dissipation power to engine efficiency conversion factor map	[280 268 340 245 275 275]
GSel_NrGear_P	number of gear parameter	7

3.4.2.6 5_FutureGearTrendCheck :

Desired gear will be calculated according to predictive road data. Background information was explained in chapter 2.Method Description. The input signals were taken from 3_EngineEfficiencyCheck and 4_ClutchHeatDissipationCheck subsystems and these values are evaluated with the matlab function block, as can be seen in Figure 3.32.

Engine efficiency and clutch heat dissipation signals are initialized inside the matlab function as can be seen in Figure 3.33. Clutch heat dissipation signals GSel_EffCluLoss-GearX are a vector, which can be divided 7 equal layers. First layer represents power loss during shifting from 1 to 1, second layer represents shift from 1 to 2, and so on. The X in the name represents the actual gear. In the Figure 3.33, in first lines, clutch heat dissipation signals are structured to be able to use in the script.

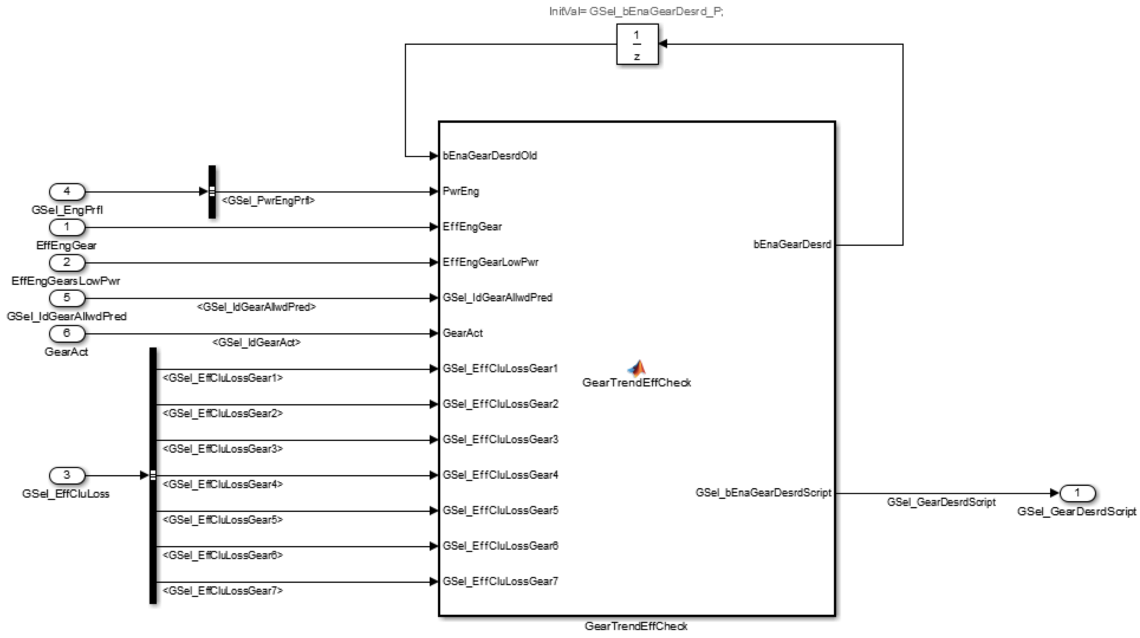


Figure 3.32: Calculation of Future Gear Trend in Matlab/Simulink Environment

```
%% Arrange GSel_PwrCluLoss
NrData = length(PwrEng);
GSel_EffEng = ones(NrData,7);
GSel_EffEngLowPwr = ones(NrData,7);
EffCluLossGear1 = ones(NrData,7);
EffCluLossGear2 = ones(NrData,7);
EffCluLossGear3 = ones(NrData,7);
EffCluLossGear4 = ones(NrData,7);
EffCluLossGear5 = ones(NrData,7);
EffCluLossGear6 = ones(NrData,7);
EffCluLossGear7 = ones(NrData,7);
GSel_EffClu = ones(NrData,7,7); %ones(NrData,Reqd,Act);

%% Arrange clutch efficiency signals
for i = 1:7
    EffCluLossGear1(:,i) = GSel_EffCluLossGear1((1+(i-1)*NrData):i*NrData);
    EffCluLossGear2(:,i) = GSel_EffCluLossGear2((1+(i-1)*NrData):i*NrData);
    EffCluLossGear3(:,i) = GSel_EffCluLossGear3((1+(i-1)*NrData):i*NrData);
    EffCluLossGear4(:,i) = GSel_EffCluLossGear4((1+(i-1)*NrData):i*NrData);
    EffCluLossGear5(:,i) = GSel_EffCluLossGear5((1+(i-1)*NrData):i*NrData);
    EffCluLossGear6(:,i) = GSel_EffCluLossGear6((1+(i-1)*NrData):i*NrData);
    EffCluLossGear7(:,i) = GSel_EffCluLossGear7((1+(i-1)*NrData):i*NrData);
end
GSel_EffClu(:,:,1) = EffCluLossGear1;%shift from 1to2
GSel_EffClu(:,:,2) = EffCluLossGear2;%shift from 2to1, 2to3
GSel_EffClu(:,:,3) = EffCluLossGear3;%shift from 3to2, 3to4
GSel_EffClu(:,:,4) = EffCluLossGear4;%shift from 4to1, 4to3, 4to5
GSel_EffClu(:,:,5) = EffCluLossGear5;%shift from 5to2, 5to4, 5to6
GSel_EffClu(:,:,6) = EffCluLossGear6;%shift from 6to1, 6to3, 6to5, 6to7
GSel_EffClu(:,:,7) = EffCluLossGear7;%shift from 7to2, 7to4, 7to6

%% Arrange Engine efficiency signals
for i = 1:7
    GSel_EffEng(:,i) = EffEngGear((1+(i-1)*NrData):i*NrData);
end

for i = 1:7
    GSel_EffEngLowPwr(:,i) = EffEngGearLowPwr((1+(i-1)*NrData):i*NrData);
end
```

Figure 3.33: Initialization of the used signals in the future gear trend evaluation script.

Out of clutch heat dissipation signals, 3 dimensional matrix is created, where column represents to be shifted gear, row represents time span and 3. dimension represents actual gear. The part 'Arrange Engine Efficiency' assigns the engine efficiency signals to variable GSel_EffEng and to variable GSel_EffEngLowPwr for low engine power to be used inside the script. This variable is a matrix, where each column represents the desired gear and rows represents each sample time.

In the Figure 3.34, engine efficiency evaluation for the coming gears are implemented. By the engine efficiency map, fuel consumption in g/kWh is calculated. This map output is multiplied with an engine power to be able to get fuel consumption in g/h, which is used for the selection of the most efficient gear. If gear is not selected, engine efficiency of this gear will be zero. Adding the high value like 70000, keeps these not selected gears out of evaluation. In the gear efficiency evaluation, minimum value means minimum fuel consumption, that's why value with a minimum engine efficiency will be selected as the most desired gear. The same calculation is done one time for the general gear efficiency and one time for the gear efficiency in low engine power.

```
%% Evaluation of engine efficiency
%If engine efficiency will not change
%too much after new gear, keep the current one
% PwrEng in Watt
PwrEng_M = ones(NrData,7);
for i=1:7
    PwrEng_M(:,i) = PwrEng/1000; %Matrix form of PwrEng in kW
end
GearEval = GSel_IdGearAllwdPred.*GSel_EffEng.*PwrEng_M + ~(GSel_IdGearAllwdPred).*70000.*ones(NrData,7);
GearEvalLowPwr = GSel_IdGearAllwdPred.*GSel_EffEngLowPwr + ~(GSel_IdGearAllwdPred).*70000.*ones(NrData,7);
GearEval_min = ones(NrData,1);
GearEvalLowPwr_min = ones(NrData,1);
bEnaGearDesrd = ones(NrData,7);
IdGearOld = ones(NrData,1);
IdGearDesrd = ones(NrData,1);
for i = 1:NrData
    if PwrEng(i)<3699.5
        GearEvalLowPwr_min(i,1) = min(GearEvalLowPwr(i,1:7));
        for j = 1:7
            bEnaGearDesrd(i,j) = ~(GearEvalLowPwr(i,j)-GearEvalLowPwr_min(i,1));
        end
    else
        GearEval_min(i,1) = min(GearEval(i,1:7));
        for j = 1:7
            bEnaGearDesrd(i,j) = ~(GearEval(i,j)-GearEval_min(i,1));
        end
        IdGearOld(i) = find(bEnaGearDesrdOld(i,1:7));
        if ((GearEval(i,IdGearOld(i))-GearEval_min(i,1))<100)
            IdGearDesrd(i) = find(bEnaGearDesrd(i,1:7));
            if IdGearOld(i)~=IdGearDesrd(i)
                bEnaGearDesrd(i,1:7) = bEnaGearDesrdOld(i,1:7);
            end
        end
    end
end
```

Figure 3.34: Evaluation of the engine efficiency. If engine efficiency does not change more than a parameterizable value after the new desired gear, keep the old desired gear.

Afterwards, used variables in the for loop is initialized. For loop turn in number of sample time in defined times span. Explanation of these variables are presented in Table 3.38.

In the for loop, there are two main conditions; firstly low engine power is checked. If engine power is lower than a threshold, engine efficiency map for low engine power is used for gear selection. This condition includes low engine power condition as well as deceleration. The second condition is normal driving condition, which engine power has a value higher than a threshold. In this time, Desired gear vector (bEnaGearDesrd) is filled according to minimum engine efficiency. If the selected gear is different than the old desired gear 'IdGearOld', engine efficiencies are compared for old and new selected gear. If the engine efficiency difference is not greater than 100 g/h, which is a parameterizable value, old desired gear will be kept, otherwise new desired gear is selected.

In Figure 3.35 power loss during the shift is evaluated. In the first 'for loop', variables are initialized, where values are from the calculation in Figure 3.34.

```
%% %Powerloss During shift
%fuel consumption changes
for i = 1:NrData
    IdGearDesrd(i) = find(bEnaGearDesrd(i,1:7));
    IdGearOld(i) = find(bEnaGearDesrdOld(i,1:7));
end
if ((IdGearDesrd(1)~=GearAct) && (IdGearOld(1)==GearAct))%Desired gear Id is different than actual
    for i = 1:NrData
        if (IdGearDesrd(i)==GearAct)% Coming GearId is again actual one after some time
            DeltaTime = i;
            GSel_TiShift_P = 100;
            if (DeltaTime-2*GSel_TiShift_P)>0
                EffEngDrive_Sum = GSel_EffEng(1,GearAct);
                for j=1:DeltaTime %Condition stay in the same gear-> sum gear efficiencies
                    EffEngDrive_Sum = EffEngDrive_Sum+GSel_EffEng(j,GearAct);
                end
                %Condition shift the new gear and shift back to actual one
                EffClu_Sum = GSel_EffClu(1,IdGearDesrd(i-1),GearAct);
                for l = 2:GSel_TiShift_P  %first shift
                    EffClu_Sum = EffClu_Sum+GSel_EffClu(l,IdGearDesrd(i-1),GearAct);
                end
                EffEngShift_Sum = GSel_EffEng(GSel_TiShift_P+1,IdGearDesrd(i-1));
                for m = (GSel_TiShift_P+2):(DeltaTime-GSel_TiShift_P)     %Drive in new gear
                    EffEngShift_Sum = EffEngShift_Sum+GSel_EffEng(m,IdGearDesrd(i-1));
                end
                EffCluBack_Sum = GSel_EffClu(DeltaTime-2*GSel_TiShift_P,GearAct,IdGearDesrd(i-1));
                for k = (DeltaTime-2*GSel_TiShift_P+1):DeltaTime   %second shift, go back to actual one
                    EffCluBack_Sum = EffCluBack_Sum+GSel_EffClu(k,GearAct,IdGearDesrd(i-1));
                end
                if EffEngDrive_Sum<(EffClu_Sum+EffEngShift_Sum+EffCluBack_Sum)
                    bEnaGearDesrd(1,1:7) = [0 0 0 0 0 0 0];
                    bEnaGearDesrd(1,GearAct) = 1;
                end
            end
        end
    end
end
GSel_bEnaGearDesrdScript = bEnaGearDesrd(1,1:7);
```

Figure 3.35: Comparison between shifting and driving in the current gear losses

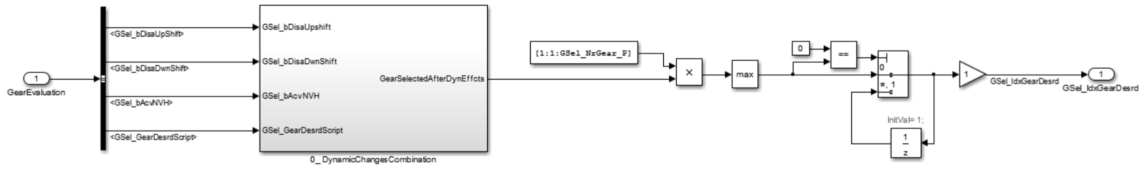
In the first 'if', it is checked whether the desired gear is different than the actual one, namely gear shift is requested by the engine efficiency calculation. Then the time interval is defined for the condition, where actual gear will be requested after some time again. If the calculated time interval, 'DeltaTime' is more than 2 shifting times, this calculation will be skipped. Since the time span which is defined in the beginning of the software is not sufficient for this scenario check.

If the prediction time span is enough, following calculations will be done. Overall engine efficiency is calculated for the condition keeping the actual gear. Result is written to the variable 'EffEngDrive_Sum'. Then, shifting to the desired gear, driving in desired gear and come back to the actual gear losses are calculated and written to the variables EffClu_Sum, EffEngShift_Sum and EffCluBack_Sum respectively. In the end, these scenarios are compared in the losses point of view. If the driving in the actual gear has less fuel consumption compare to the shifting scenario, actual gear will be kept. Namely, element id, which has the actual gear is set in the vector bEnaGearDesrd. Otherwise, bEnaGearDesrd vector is kept same and desired gear is updated with the new gear id, like calculated in Figure 3.34. In the end of the line, output from script assignment is implemented.

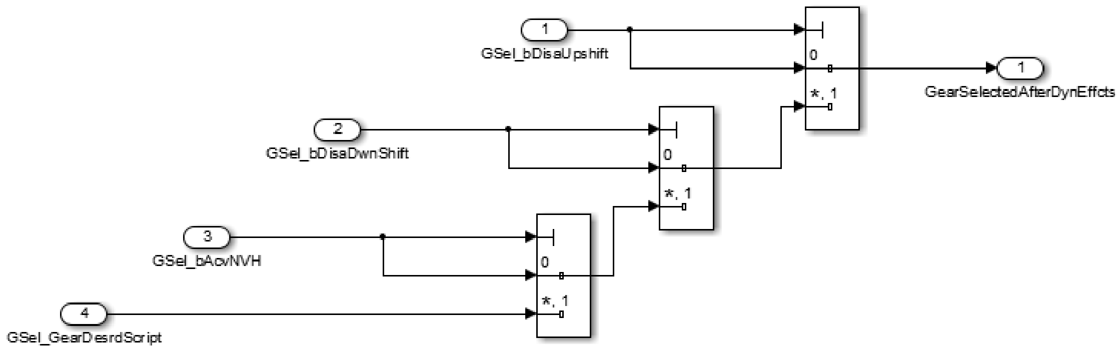
Table 3.38: Input/Output signals and variables used in 5_FutureGearTrendCheck subsystem and inside the matlab function.

Signal name	**Description**
PwrEng	Engine power profile in the parameterizable time interval in $Watt$. It is a vector by [redictiontimespan]
GSel_IdGearAllwdPred	Allowed gear with respect to engine allowances in parameterizable time interval. It is a matrix, which each column represents gear id and rows are value for each sample time
EffEngGear	Engine gear efficiencies for each gear, calculated in 3_EngineEfficiencyCheck subsystem
GSel_IdGearAct	Actual gear value
GSel_EffCluLossGearX	Clutch power loss calculated during shifting from gear X to other 7 gears in 4_ClutchHeatDissipationCheck subsystem,it is a vector with a number of element 7 times predicted time span.
GSel_GearDesrdScript	Desired gear, calculated in 5_FutureGearTrendCheck subsystem
GearEval_min	minimum engine efficiency value for the specified time between selected gears
bEnaGearDesrd	this is a 7 element vector. When X gear is selected, X. element of a vector assigned 1
IdGearOld	Old desired gear Id
IdGearDesrd	Desired gear Id
EffEngDrive_Sum	Overall engine efficiency in g/h, for the scenario, come back to actual gear after some time
EffClu_Sum	First shifting power loss in the clutch in g/kWh, for the scenario, come back to actual gear after some time
EffEngShift_Sum	Engine efficiency in g/h between two shifts, for the scenario, come back to actual gear after some time
EffCluBack_Sum	Second shifting power loss in the clutch in g/h, for the scenario, come back to actual gear after some time

3.4.3 2_DesiredGearSelection



(a) Desired Gear Calculation Subsystem



(b) Combination of Evaluated Gears from 1_EvaluationofthePossibleGear Subsystem

Figure 3.36: Calculation of Desired Gear in Matlab/Simulink Environment

Under this subsystem, evaluated gear values from 1_EvaluationofthePossibleGear subsystem is combined. All input signals are vector, which has 7 element which represents one gear value. If the value is 1, it means, this gear is allowed. If the value is 0, it means, this gear is not allowed. Combined form of input signals can be seen in Figure 3.36.b. Within all this signals Upshift prohibition flag 'GSel_bDisaUpShift' has the highest priority. Then in a sequence, Downshift prohibition flag 'GSel_bDisaDwnShift', NVH evaluation flag 'GSel_bAcvNVH' and gear id calculated in 5_FutureGearTrendCheck subsystem GSel_GearDesrdScript has a higher priority. Which means, if one of the upshift prohibition flag is raised, the other inputs are not considered but gear is selected according to 'GSel_bDisaUpShift' flag. If none of the flags are activated, gear selection will be done with calculation under 5_FutureGearTrendCheck script.

Afterwards, the selected gear Id will be defined, see Figure 3.36.a. The gear has a higher id will be selected. Ideally, in this level of the component, only one gear should be requested. Vector elements going as an input to the max block, has one value between 1 to 7 and other values are all zero. In the end of the calculation, precaution is taken, incase, there will be no gear selected. In this case, old desired gear is kept.

Table 3.39: Input/Output signals used in 2_DesiredGearSelection subsystem

Signal name	**Description**
GSel_bDisaUpShift	Enabled/Disabled gears with respect to upshift prohibition in vector for each gear, calculated in 0_UpshiftProhibition subsystem
GSel_bDisaDwnShift	Enabled/Disabled gears with respect to downshift prohibition in vector for each gear, calculated in 1_DownshiftProhibition subsystem
GSel_bAcvNVH	Enabled/Disabled gears information with respect to NVH behavior in vector for each gear, calculated in 2_NVH subsystem
GSel_GearDesrdScript	Desired gear in vector for each gear, calculated in 5_FutureGearTrendCheck subsystem
GSel_IdxGearDesrd	Desired gear which is sent as an output from Gear Selection Strategy component to other components in the TCU

4 Simulation and Testing Environment

Under this section, simulation environment and test results will be presented. AVL VSM™ is used for creating the simulation environment for functional testing. AVL VSM™ is a simulation package of tools that supports engineers in the field of concept development, calibration and functional validation [7]. By this tool it is possible to simulate longitudinal and lateral dynamic behaviors of the vehicle.

4.1 Testing Environment

VSM provides to set drivetrain, engine, tyre, suspension, etc. characteristics. There are also different kinds of transmission models like AT, MT (Manual Transmission), CVT (Continuously Variable Transmission).

VSM uses Matlab Simulink environment for the execution of the tests. For each required part of the vehicle such as Drivetrain, ECU Module, Engine Module, Steering Module, etc. and road, there are S-functions. S-function is a computer language description of a Simulink block written in Matlab, C, C++ or Fortran. Signal flow between VSM and gear selection strategy is represented in Figure 4.1. VSM is the input generator and provides all required interfaces to the Gear Selection Strategy. Gear selection strategy executes the inputs and provides desired gear to the VSM.

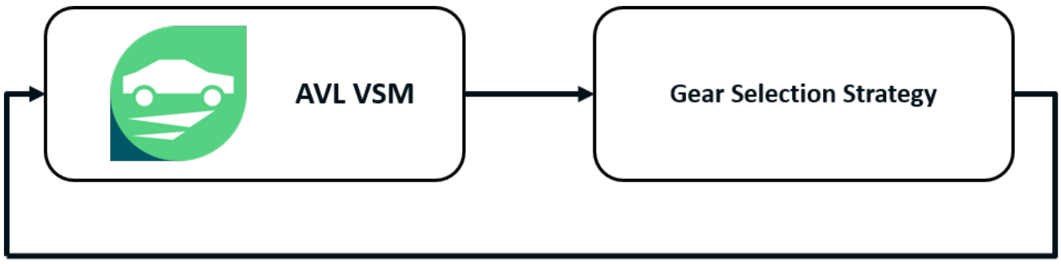


Figure 4.1: Communication between Gear Selection Strategy and AVL VSM™

Figure 4.2 shows intefaces between VSM and gear selection strategy in Simulink Environment. The yellow subsystem in the Figure 4.2 is a function call for initialization of the parameters and objects used in software. The green subsystem is the developed software within this thesis. AVL VSM requires desired gear which is provided by first output Gear. When gear change is active, which is an information coming from AVL VSM, desired gear is not updated, when shifting is not active, calculated desired gear is sent. The second output is requested by AVL VSM is Gear Change, which is provided by second output. This signal contains information whether upshift or downshift is requested. When upshift is requested, 1 is sent. When downshift is requested, -1 is sent. When desired gear is the same with the current gear, namely no shift is requested, 0 is sent.

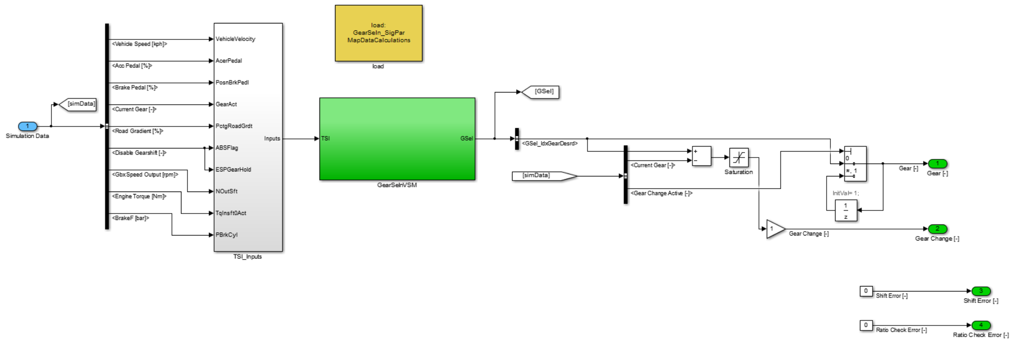


Figure 4.2: Interface connections between developed gear selection strategy and AVL VSM™

Parameter settings for the drivetrain, engine, vehicle geometry are parameterized through the VSM GUI , see Figure 4.3. Front differential configuration is selected as drivetrain configuration.

Engine torque map, fuel consumption map, transmission efficiency map, inertias, aerodynamic constans are assigned via the VSM GUI according to project vehicle specifications. For a driver, brakes, clutches, suspension, pedal maps typical mid-size vehicle default values are used, which is already exist in VSM. For the both simulation, same VSM GUI parameterization is used.

Vehicle velocity and road gradient vectors are also given from there.

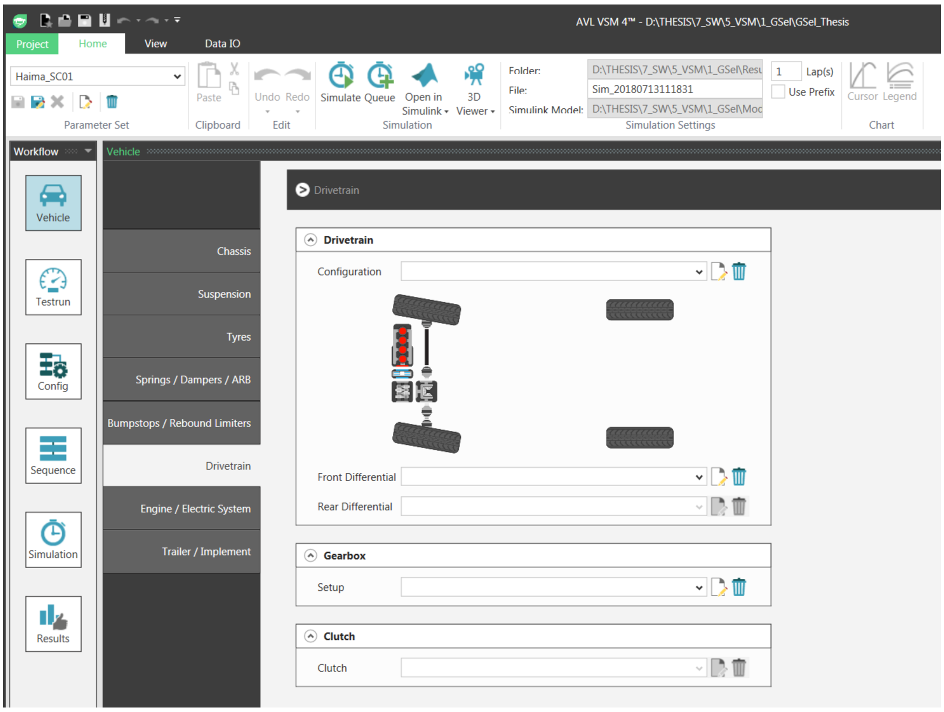


Figure 4.3: AVL VSM™ GUI

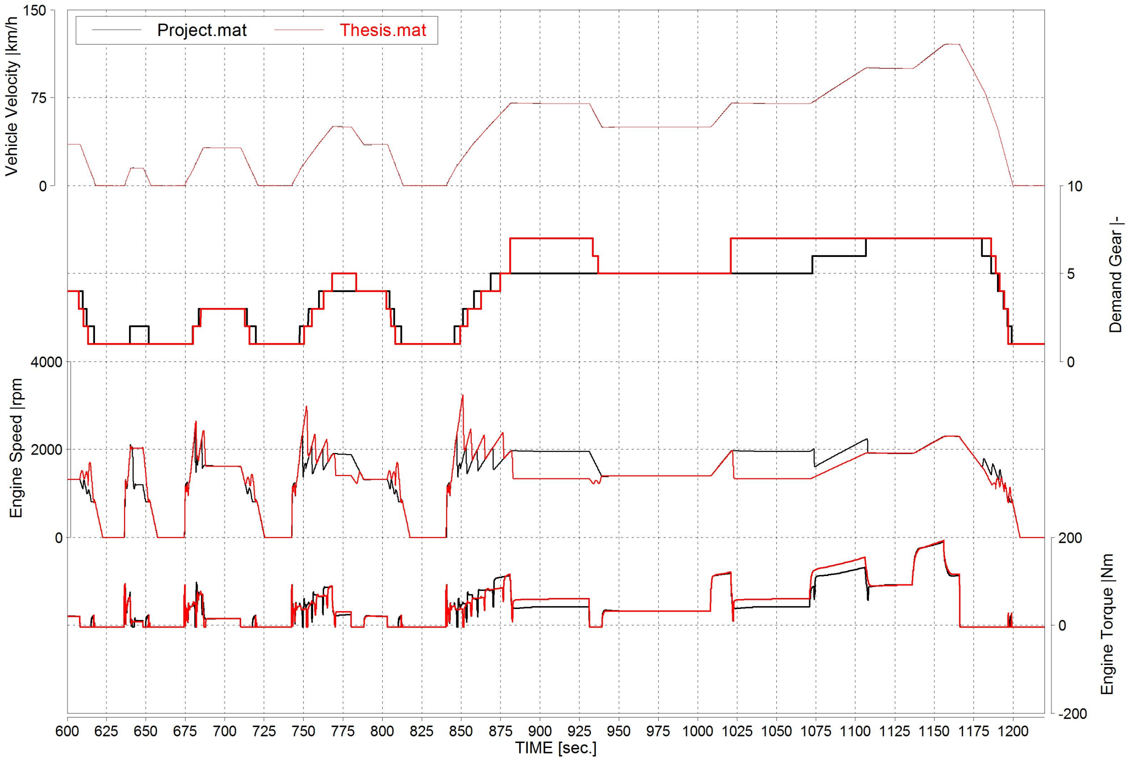
4.2 Test Results

Functional tests were run with the initial calibration of the project, which has been calibrated at a test track. This means, gear selection strategy works in normal conditions properly on a flat road test track and the project is close to finishing phase. Project uses state of the art shift maps as a gear selection strategy, which was explained under 2.1 Current Technology section. Software developed in this study is calibrated according to simulation model in VSM.

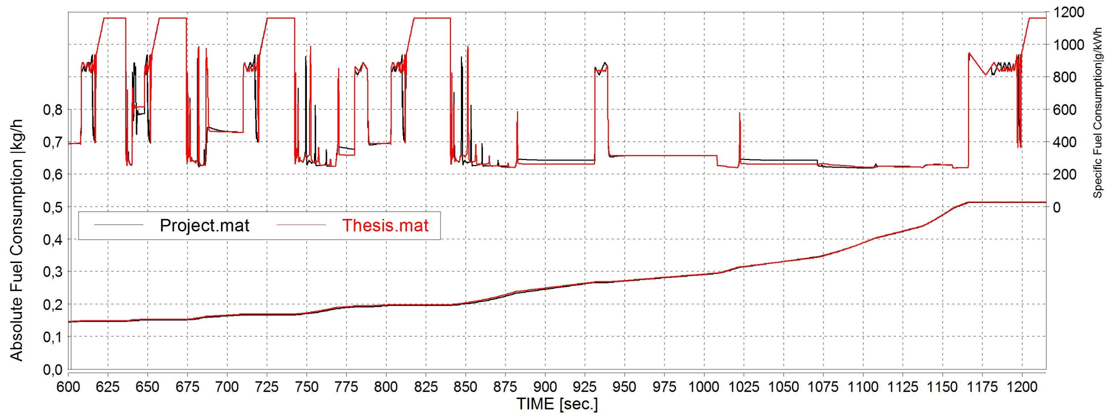
Same parameterization is used for both project and thesis simulation environment in VSM GUI. Overall, 3 different testcases were created for functional testing. In first test, NEDC (New European Driving Cycle) [9] is given as a vehicle velocity profile for the flat road and simulated for the gear selection developed in project and in this study. NEDC is a driving cycle, designed to assess the emission levels of engines and fuel economy in passenger cars. The results can be seen from Figure 4.4. For the better visibility, first repeated part of the test results were cutted and results are shown after 600 seconds. Results until 600. seconds can be seen from Appendix C.

As it is indicated in legend of the graphs, black lines are the results for the component developed in project and red lines are the results for the component developed in this study. Absolute fuel consumption levels are slightly different, see Figure 4.4.b. Differences in emissions can be seen better from specific fuel consumption graph. Project gear selection selects the gears according to vehicle velocity and accelerator pedal position. Calibration engineers calibrated these maps to be able to get better fuel consumption levels. Whereas gear selection developed in this study decides desired gear according to engine efficiency maps with the consideration of drivetrain losses. Around 640. second, while project gear selection demands second gear, thesis gear selection stays in the same gear. In this part, project gear selection efficiency performance is better with respect to specific fuel consumption. Around 775. seconds, from 880. seconds to 937. seconds and from 1020. seconds to 1072. seconds according to different gear demands better specific fuel consumption is reached by developed gear selection. As it was expected in the beginning of the study, better fuel consumption levels were not reached completely but in some parts in the simulation better fuel consumption levels can be observed. In the end of the test cycle, absolute fuel consumption levels difference is around 1 g/h.

It should be noted down that, during the standstill, requesting the neutral gear logic was not implemented, that’s the reason in this time first gear is demanded and specific fuel consumption has a highest value.



(a) Vehicle velocity, gear demand, engine torque and speed results



(b) Fuel consumption levels

Figure 4.4: Test Results for NEDC test cyle in flat road after 600 seconds

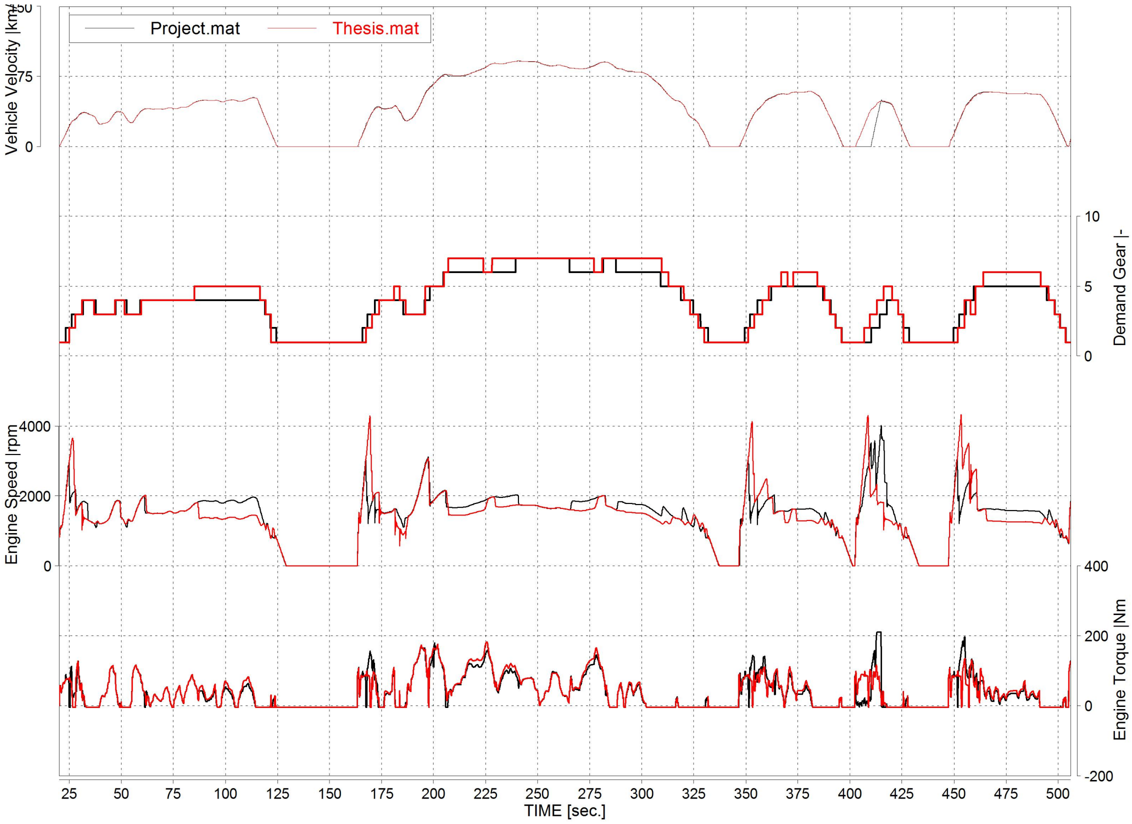
In second test, cold start phase of FTP-75 [9] driving cycle is given as a vehicle velocity profile for the flat road and simulated for the gear selection developed in project and in this study. FTP-75 (United States Environmental Protection Agency Federal Test Procedure) is a city driving cycle to measure tailpipe emissions and fuel economy of passenger cars. The results can be seen in Figure 4.5.

Velocity profile in this cycle is more aggressive than the NEDC cycle. In this case, clutch heat dissipation checks calculation, which is explained under section 3.4.2.6 5_FutureGearTrendCheck, plays an important role. When there is velocity change occurs just a short time, maybe there is no need to demand gear change but keeping the same gear is more efficient. This logic is needed, since gear selection is done in this study based on output shaft torque. In FTP-75, vehicle velocity changes aggressively and this results also aggressive changes in output shaft torque, namely engine torque. Without this logic, gear demand would be changed always according to engine torque changes. Aggresive vehicle velocity changes are overcome with the hysteresis implementation in the shift maps in project gear selection, which was explained under 2.1 Current Technology section.

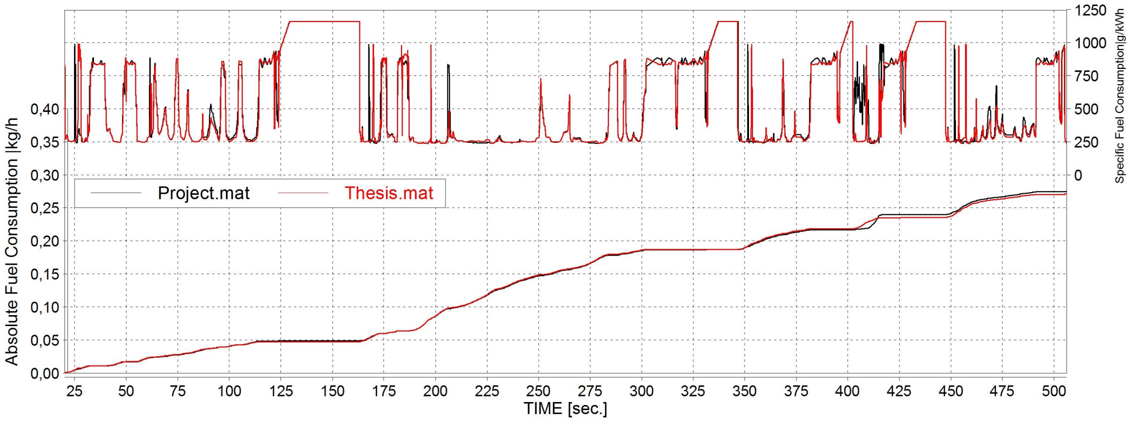
In this study, it was possible to get improvement in some parts of the cycle in specific fuel consumption, which can be seen in Figure 4.5.b. After 415. seconds, it is clearly seen that absolute fuel consumption levels are also better in developed gear selection results. The other remark about the results are around 410. seconds, project is not able to follow vehicle velocity, namely it cannot fulfill the driver demand, which is also an important requirements that should be fulfilled during the vehicle development.

As expected in the beginning of the study, better fuel consumption levels were not able to reached completely but improvements can be seen in some parts of the simulation. In the end, absolute fuel consumption level difference is measured as 3 g/h. Gear selection for the future can be improved more with the preselection logic for DCTs. In this way, coming gear can be preengaged before the selection. This strategy can provide better shift quality and less losses during shifting and this results in a better fuel consumption levels.

It should be noted down that, during the standstill, requesting the neutral gear logic was not implemented, that's the reason in this time first gear is demanded and specific fuel consumption has a highest value.



(a) Vehicle velocity, gear demand, engine torque and speed results



(b) Fuel consumption levels

Figure 4.5: Test Results for FTP-75 test cyle in flat road

In third test, gear selection strategy developed in this study was compared flat road vs 5% road gradient condition by giving the last cycle of the NEDC as a velocity profile. The aim of this test, showing how software works for road gradient conditions. The results can be seen from Figure 4.6.

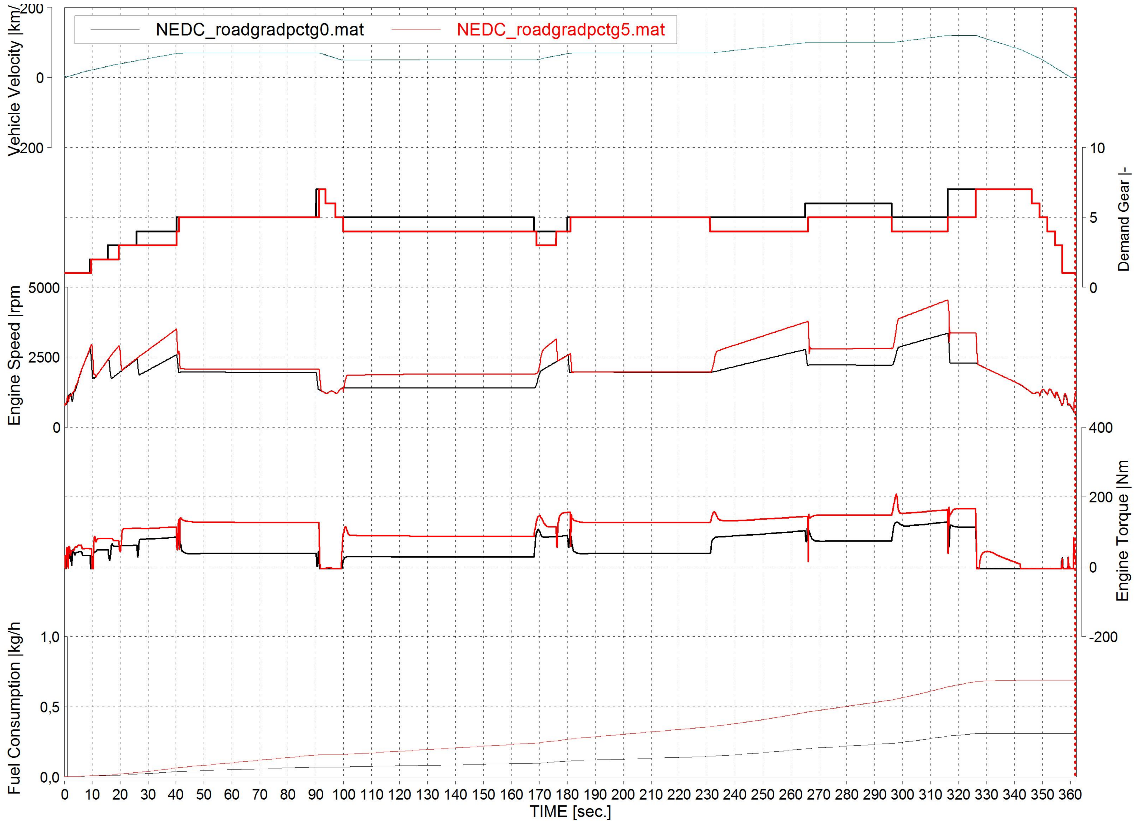


Figure 4.6: Test results for NEDC in 5% road gradient

As it is indicated in the legend of the graph, black lines show results in flat road, and red lines show results in 5% road gradient. When road gradient was 5%, shiftings happened later than flat road tests, which was an expected result. During uphill driving, more torque is needed in the wheels, that's why to be able to get this higher torque, it is required to stay in lower gear longer to be able to get higher engine speed. Around 230.sec, when velocity start to increase from around 70 km/h to 100 km/h, one downshift is requested in 5% road gradient testing. The reason is engine does not have enough torque to be able to deliver this desired torque to output shaft. That's why one downshift is performed, engine accelerates and afterwards, previous gear is engaged again. This can be improved by including torque reserve logic. While output shaft torque is calculated, torque reserve can be added to the output shaft. Torque reserve represents the power reserve of the vehicle for the coming situatons. Gear selection is done according to updated output shaft torque, in other words, engine operation points are evaluated with the updated output shaft torque. In this way, it may be possible to drive without downshifting, since current gear can already deliver required acceleration levels. As an improvement to the current implementation, driver acceleration request in the future can be predicted and this value can be reserved during output shaft torque calculation. In this way, when acceleration is requested, there is no need to fulfill this request by downshifting, since this torque is already available in engine.

For fuel consumption point, fuel consumption is higher when it is driven in uphill road. Since more torque is needed in output shaft and engine torque is higher in this time for giving the higher torque to the wheels.

5 Conclusion and Future Work

As can be seen from test results, it was possible to get better fuel consumption levels partially with the developed gear selection strategy. In the end of the test cycles, absolute fuel consumption levels were calculated quite same. It should be noted down that, developed software requires less calibration work compared to the gear selection strategy developed in the project, since during the gear selection, engine efficiency maps are used instead of several conventional shift maps. For 7 gears transmission, each shift map contains 6 upshift and 6 downshift lines, 14 supporting points per line, 20 shift maps for different conditions. Totally, there is 12*14*20 = 3360 parameters. Instead of 3360 parameters, gear selection in this study uses engine efficiency map, which has 308 parameter inside. Moreover, developed gear selection requires less implementation effort, since instead of 20 shift maps, there is just one engine efficiency map. There is also not extra implementation effort for selection of the proper shift map.

By the third test cases, it was also shown that developed gear selection strategy is applicable for the road gradient conditions. Unfortunately, performance could not be compared with the project gear selection strategy, since calibration was not completed for driving uphill and downhill conditions.

There is still room for improvement in the developed gear selection strategy. Shift sequencing can be considered, which depends on the applied transmission type. Since it is possible to get the coming road conditions and velocity, shift sequencing can be calculated in advance. During deciding shift sequencing, efficiency as well as driver feeling can be considered. Maybe shifting from 7 to 1 gear is the most efficient option, however how driver will feel during this shift is also important. Moreover, for the DHT transmission, there will be several kinds of shift possibilities in consideration of torque splits. This shift sequencing is considered partially in this thesis. The implementation can be seen under section 3.4.2.6. 5_FutureGearTrendCheck. This logic can be improved further for hybrid and different kinds of transmission types.

Technology is developed more in hybrid drive, since the strict emission regulations. Transmission types are also improved through the DHT. In DHT, it is not possible to use conventional shift maps, since there is ECVT drives. ECVT driving modes are used in conjunction with pure electric driving gears as well as parallel hybrid driving gears, which makes gear selection with shift maps almost impossible. Hybrid vehicle architecture and ECVT driving modes bring torque splits into account, which should be also considered, during developing the gear selection for hybrid vehicles.

In the future, sailing function can be included, which is an energy efficient driving operating mode, that turning off the engine during vehicle is still moving. In the same way, coasting can be also included. Coasting is defined as decoupling the engine from the vehicle and running engine in idle mode. These modes can be used during decelerating for the standstill position. For example, since vehicle velocity profile is known by prediction calculations, when vehicle is decelerating, and it is seen that vehicle has enough power to be able to come to the predicted standstill position, sailing or coasting can be activated.

Torque reserve calculation can be also included as it is mentioned in the results of driving in uphill road. With the torque reserve logic, it is possible to get sportive drive without downshift.

In DCT, even and odd gears are separated onto two separate shafts. One clutch is used just for engaging odd gears and other clutch is used just for engaging even gears. With this two shaft architecture, it is possible to preselect the coming gear. For example, when the vehicle speeds up, controller in the car automatically detect the gear change point and will preselect the upper gear. In the same way, when the vehicle speeds down, controller will preselect a lower gear. When gear change is requested, currently engaged clutch disengages and the other preselected clutch engages. Currently, this logic is not implemented in the thesis software. This logic should be also included for DCT control software. When this is introduced in the current architecture, there should be one more output for preselect gear, namely future gear. This gear is already calculated under 3.4.2.6. 5_FutureGearTrendCheck subsystem but not given to the output. With the improvement of gear sequencing calculation, this logic can be also improved.

Power losses during the shifting in clutches can be also improved. In this calculation, shifting time is given as a constant value. This value highly depends on torque handover functionality, which handles the torque shifting from one clutch to other. In the future, torque handover part of the software can send shifting time according to the hardware limitations. By this, it is possible to get more accurate shifting loss values. For this, additional research should be done for analyzing the clutch losses during shift. The effect of the developed gear selection strategy would not be too much. After the output of the clutch losses simulation, current clutch losses parameters used in the software should be updated according to research output.

As it was described in the architecture description of the software, learning algorithm can be added after the gear selection strategy system. This algorithm can improve gear selection overtime also it helps learning the driver needs. This system can be parameter fitting approach, so that it can update parameter used in the software overtime, for better gear selection quality.

Instead of considering NVH just for the current data, it can be considered as further development during the prediction calculation. Since engine speed and torque can be calculated for future, NVH bahavior can be checked for the future conditions. In this way, even the selected gear is the best gear in efficiency point of view, it can result a bad noises or jerks, which is not desirable by the driver. The effect of the software would be using the same parameterizable map implementation with vector, namely predicted engine torque and speed inputs. Afterwards, the output of the maps should be evaluated under subsystem 3.4.2.6. 5_FutureGearTrendCheck in the software, which was explained under 3. Software Description chapter.

Symbols

F: Force on the vehicle$[N]$

m_{veh}: Vehicle mass$[kg]$

v: vehicle velocity$[m * s]$

a: Vehicle acceleration$[m/s^2]$

F_R: Rolling resistance$[N]$

f_R: Coefficient of friction F_L: Air resistance$[N]$

F_{ST}: Climbing resistance$[N]$

g: acceleration of gravity constant $[9.81 m/s^2]$

c_D: Air drag force$[N]$

ρ_A: Air density $[kg/m^3]$

A_F: Cross sectional area of the vehicle$[m^2]$

P: engine power$[Watt]$

I_E: Inertia of engine $[kgm^2]$

I_G: Inertia of transmission$[kgm^2]$

I_{AF}: Inertia of front axes $[kgm^2]$

I_{AR}: Inertia of rear axes $[kgm^2]$

I_F: Total inertia in front axes $[kgm^2]$

I_R: Total inertia in rear axes $[kgm^2]$

M_{Jout}: Total inertia torque loss in output shaft level with respect to engaged gear $[Nm]$

P_{Jout}: Total inertia power loss in output shaft level with respect to engaged gear $[Watt]$

r: Vehicle dynamic radius $[m]$

i_A: axle drive ratio

i_G: discrete gear ratio

i: total gear ratio includes discrete gear ratio and axle drive ratio

ω_{Eng}: Engine synchronized speed for each gear $[rad/s]$

M_{Eng}: Engine target torque for each gear $[Nm]$

η: transmission efficiency

P_v: Loss of power

W_v: Loss of energy

P_m: Mean effective pressure $[bar]$

V_{cyl}: Volume of an engine cylinder $[m^3]$

Acronyms

ABS: Anti-lock Braking System
AT: Automatic Transmission
BMEP: Brake mean effective pressure in engin piston cylinder
BSFC: Brake Specific Fuel Consumption
CAN: Controller Area Network
CVT: Continuously Variable Transmission DCT: Dual Clutch Transmission
DHT: Dedicated Hybrid Transmission
ECVT: Electronic Continuously Variable Transmission ECU: Engine Contol Unit
ESP: Electronic Stability Program
FWD: Front Wheel Drive
FTP75: US Environmental Protection Agency Federal Test Procedure
GPS: Global Positioning System
GUI: Graphical User Interface
HCU: Hybrid Control Unit
HMI: Human Machine Interface
MT: Manual Transmission
NEDC: New European Driving Cycle
NVH: Noise, Vibration and Harshness
RDE: Real Driving Emissions
SOC: State of Charge
SUV: Sport Utility Vehicle
TCU: Transmission Control Unit
TGDI: Turbocharged Gasoline Direct Injection

A Legend for the Simulink Blocks

Table A.1: simulink blocks, which used in the software explanations

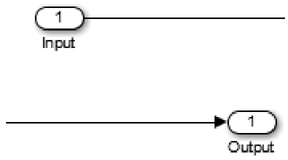
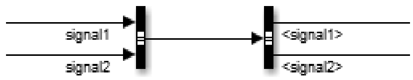
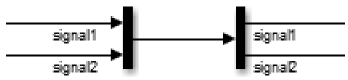
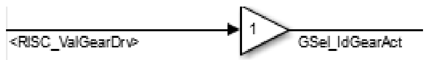
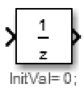
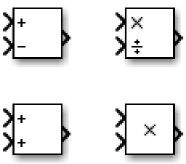
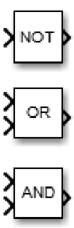
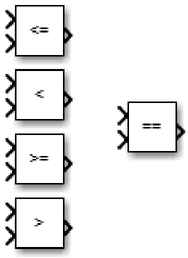
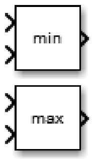
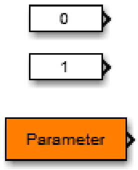
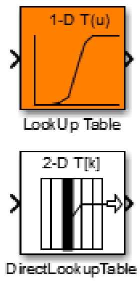
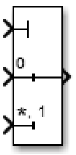
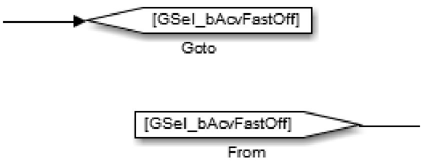
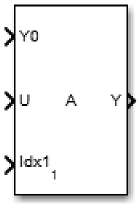
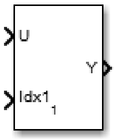
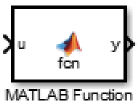
Simulink Block	Description
	Input/Output port: Inputs are the links from outside a system into the system. Outport blocks are the links from a system to a destination outside the system.
	Bus Creator/Selector: The Bus Creator block combines a set of signals into a bus. The Bus Selector block outputs a specified subset of the elements of the bus at its input.
	Mux/Demux: Mux block combines input signals of same data type and numeric type into virtual vector. Demux block extracts and outputs elements of virtual vector signal
	GainBlock: It multiples input by constant. In this software, it is used for changing the signal name.
	Unit Delay: It delays signal one sample period.
	Mathematical Operations: They are used for summation, subtraction, multiplication and division operations.
	Logical Operators: They perform specified logical operation on input
	Continued on next page

Table A.1 – continued from previous page

Simulink Block	Description
	Relational Operations: They perform specified relational operation on inputs.
	Min/Max Block: They output minimum or maximum input value.
	Constant Block: They generate constant value. When this constant value such as 'Parameter', is defined in workspace, it is colored orange in this software to show, it is a parameter, which can be parameterized from workspace.
	Lookup Tables: LookUp table approximates one/two-dimensional function. Direct Lookup Table index into n-dimensional table to retrieve element, vector, or 2-D matrix.
	Multiport Switch: it chooses between multiple block inputs. In this picture, first input is the control input, when it is 0, second input is given as an output, otherwise, third input is given as an output.
	Goto/From: Goto block passes block input to From blocks. From block accept input from Goto block.

Continued on next page

Table A.1 – continued from previous page

Simulink Block	Description
	Assignment: It assigns values to specified elements of signal. Input Y0 is a vector. Input Idx1 is the Id of a vector Y0. Y0 element value, where id is defined from Idx1 input is replaced by U.
	Selector: It selects input elements from vector, matrix, or multidimensional signal, where U is the vector input, Idx1 is the Id of a vector element.
	Matlab Function: It is a function defined with Simulink block, where you can write matlab script inside as a function and this function is called in every sample time.

B Legend for the Simulink Libraries

In Simulink, library is defined as reusable libraries of blocks and subsystems. Some libraries were created to be used during the implementation, where logic is repeated several times.

B.1 S-RFlipFlop

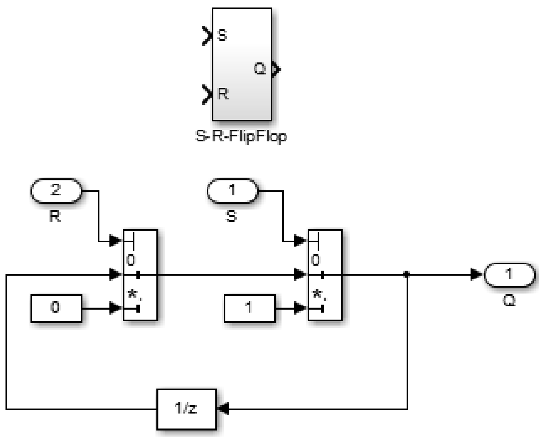


Figure B.1: S-RFlipFlop library subsystem in Matlab/Simulink Environment

SR Flip Flop holds the flag Set 'S' until it is reseted by input Reset 'R'. For the truth table, please see TableB.1

Table B.1: Truth table of SR Flip Flop

S	**R**	**Q**
1	0	1
0	0	1 (after S=1, R=0)
0	1	0
0	0	0 (after S=0, R=1)
1	1	1

B.2 ResetTimer

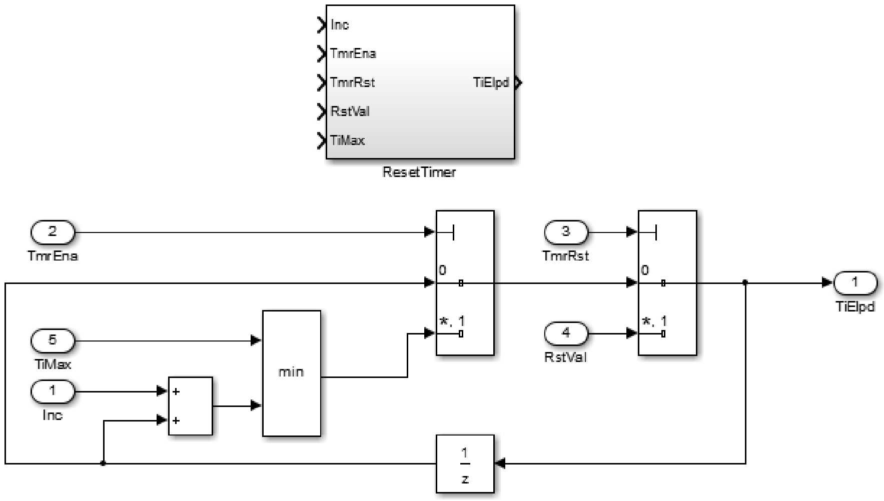


Figure B.2: ResetTimer library subsystem in Matlab/Simulink Environment

ResetTimer library counts the time by adding the Increment 'Inc' to the last value of Time Elapsed 'TiElpd' in every calculation time step, when it is triggered by Enable Timer 'TmrEna'. Limitation by Maximum time 'TiMax' can be also set for not counting to the infinity.

B.3 LowPassFilterReset

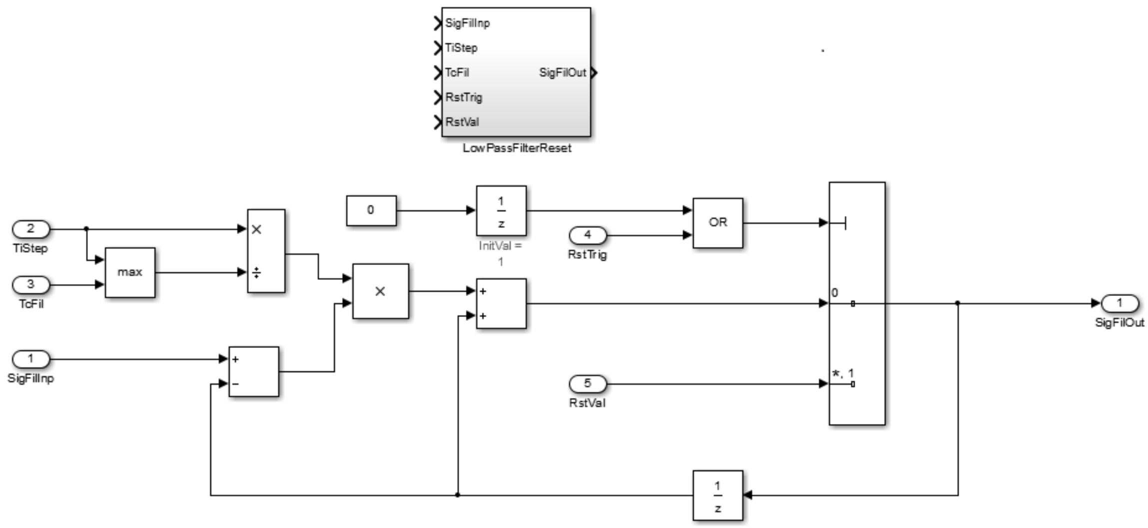


Figure B.3: LowPassFilterReset library subsystem in Matlab/Simulink Environment

LowPassFilterReset library is a filter that passes signals with a frequency lower than a certain cutoff frequency 'TcFil', and attenuates signals 'SigFilInp' with frequencies higher than the cutoff frequency 'TcFil'. Filter is initialized by Reset Value 'RstVal' in the first time sample of the simulation. This filter can be reseted by the input Trigger Reset 'RstTrig' in any time during the simulation.

B.4 ArrangeMatrix

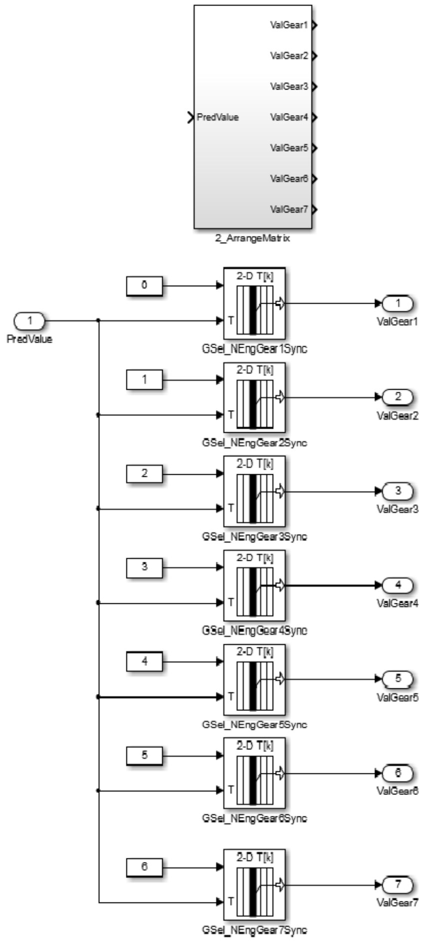
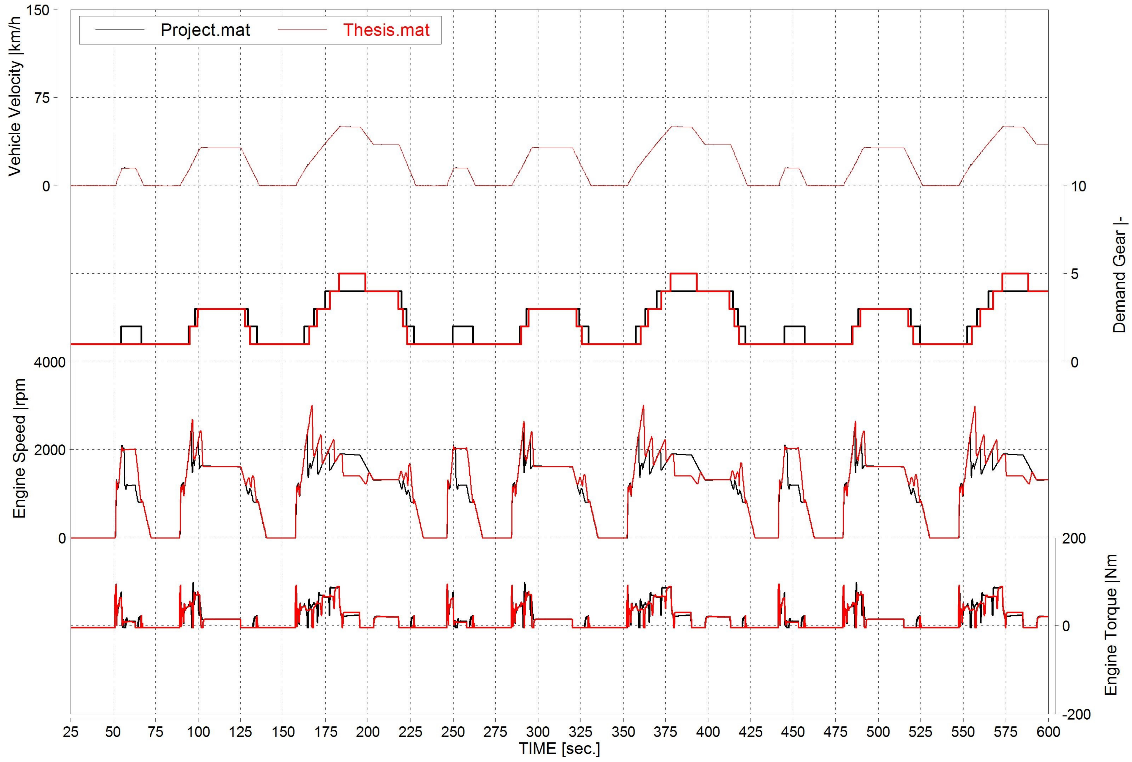


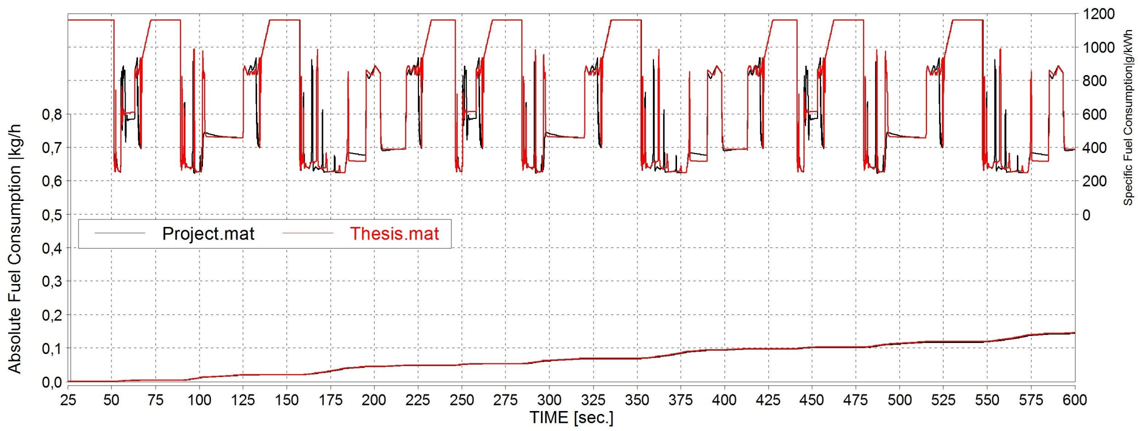
Figure B.4: ArrangeMatrix library subsystem in Matlab/Simulink Environment

ArrangeMatrix library is split the columns of a matrix which has 7 columns. In the software, it is used for splitting the prediction value matrix to its column and get the values for each gear as a vector. For example, 'ValGear1' is the calculated prediction value for the Gear 1.

C Test Results Extension



(a) Vehicle velocity, gear demand, engine torque and speed results



(b) Fuel consumption levels

Figure C.1: Test Results for NEDC test cyle in flat road before 600 seconds

List of Figures

List of Tables

References

[1] Dynamic eHorizon, 2017 (accessed Aprily 20, 2018). https://www.continental-automotive.com/en-gl/Passenger-Cars/Interior/Software-Solutions-and-Services/eHorizon/Dynamic-eHorizon.

[2] AVL Future Hybrid 7 and 8 Mode, 2018 (accessed April 22, 2018). https://www.avl.com/documents/10138/2699442/Solution+Sheet+Future+Hybrid.

[3] Real Driving Emissions (RDE), 2018 (accessed Aprily 22, 2018). https://www.avl.com/-/real-driving-emissions-rde-.

[4] Matlab Homepage, 2018 (accessed August 24, 2018). https://de.mathworks.com/?s_tid=gn_logo.

[5] How Dual-clutch Transmissions Work, 2018 (accessed July 15, 2018). https://s.hswstatic.com/gif/dual-clutch-transmission-12.gif.

[6] Audi Q7 predictive efficiency assistant, 2018 (accessed July 8, 2018). https://www.audi-technology-portal.de/en/electrics-electronics/driver-assistant-systems/audi-q7-predictive-efficiency-assistant.

[7] AVL VSM 4, 2018 (accessed July 8, 2018). https://www.avl.com/-/avl-vsm-4-.

[8] Alexander Banerjee, Maik Würthner, and Cosmin Tudosie. Prevision gps die schaltstrategie für das getriebesystem traxon. ATZ-Automobiltechnische Zeitschrift, 115(6):490–493, 2013.

[9] Tim J BARLOW, S Latham, IS McCrae, and PG Boulter. A reference book of driving cycles for use in the measurement of road vehicle emissions. TRL Published Project Report, 2009.

[10] Giusy Di Lorenzo, Fabio Pinelli, Francisco C Pereira, Assaf Biderman, Carlo Ratti, Charles Lee, and Chuhee Lee. An affective intelligent driving agent: Driver's trajectory and activities prediction. In Vehicular Technology Conference Fall (VTC 2009-Fall), 2009 IEEE 70th, pages 1–4. IEEE, 2009.

[11] Robert Fischer, Ferit Küçükay, Gunter Jürgens, Rolf Najork, and Burkhard Pollak. The Automotive Transmission Book. Springer, 2015.

[12] Adama Fofana, Olivier Haas, Vince Ersanilli, Keith Burnham, Joe Mahtani, Chris Woolley, and Karen Vithanage. Multi-objective genetic algorithm for an automatic transmission gear shift map. IFAC-PapersOnLine, 49(3):123–128, 2016.

[13] Thomas Frühwirth. Entwicklung einer schaltstrategie zur reduktion des kalibrationsaufwandes, January 2014.

[14] Sang-Hyung Ha and Hong-Tae Jeon. Development of intelligent gear-shifting map based on radial basis function neural networks. International Journal of Fuzzy Logic and Intelligent Systems, 13(2):116–123, 2013.

[15] Wolfgang Hirschberg. Script for the lecture automotive engineering, February 2014.

[16] Christoph G Keller and Dariu M Gavrila. Will the pedestrian cross? a study on pedestrian path prediction. IEEE Transactions on Intelligent Transportation Systems, 15(2):494–506, 2014.

[17] Ryuki Mabuchi and Keiichi Yamada. Prediction of driver's stop or go at yellow traffic signal from vehicle behavior. In Intelligent Vehicles Symposium (IV), 2013 IEEE, pages 1161–1166. IEEE, 2013.

[18] Viet Dac Ngo, Theo Hofman, Maarten Steinbuch, and Alex Serrarens. Gear shift map design methodology for automotive transmissions. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 228(1):50–72, 2014.

[19] Ghazaleh Panahandeh. Driver route and destination prediction. In Intelligent Vehicles Symposium (IV), 2017 IEEE, pages 895–900. IEEE, 2017.

[20] Sreenath K Reghunath, Deepak Sharma, and Ashwini S Athreya. Optimal gearshift strategy using predictive algorithm for fuel economy improvement. Technical report, SAE Technical Paper, 2014.

[21] Ralf Schuler. Situationsadaptive Gangwahl in Nutzfahrzeugen mit automatisiertem Schaltgetriebe, volume 34. expert verlag, 2007.

[22] Stephan Terwen. Vorausschauende längsregelung schwerer lastkraftwagen, volume 6. KIT Scientific Publishing, 2010.

[23] Cuong Tran, Anup Doshi, and Mohan Manubhai Trivedi. Modeling and prediction of driver behavior by foot gesture analysis. Computer Vision and Image Understanding, 116(3):435–445, 2012.

[24] Shouyi Wang, Yiqi Zhang, Changxu Wu, Felix Darvas, and Wanpracha Art Chaovalitwongse. Online prediction of driver distraction based on brain activity patterns. IEEE Transactions on Intelligent Transportation Systems, 16(1):136–150, 2015.