

**CONCURRENT DESIGN OF ENERGY
MANAGEMENT AND VEHICLE STABILITY
CONTROL ALGORITHMS FOR A
PARALLEL HYBRID VEHICLE USING
DYNAMIC PROGRAMMING**

A THESIS

SUBMITTED TO THE DEPARTMENT OF MECHANICAL
ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By

Halil İbrahim Dokuyucu

August, 2012

I certify that I have read this thesis and that in my opinion it is fully adequate,
in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Melih akmakcı (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate,
in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. İlker Temizer

I certify that I have read this thesis and that in my opinion it is fully adequate,
in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Hitay Özbay

Approved for the Graduate School of Engineering and
Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

CONCURRENT DESIGN OF ENERGY MANAGEMENT AND VEHICLE STABILITY CONTROL ALGORITHMS FOR A PARALLEL HYBRID VEHICLE USING DYNAMIC PROGRAMMING

Halil İbrahim Dokuyucu

M.S. in Mechanical Engineering

Supervisor: Asst. Prof. Melih Çakmakcı

August, 2012

Concurrent design of controllers for a vehicle equipped with a parallel hybrid powertrain is studied. Our work focuses on simultaneously solving two automotive control problems, energy management and vehicle stability, which are traditionally considered separately. The optimal actions for the controllers are obtained by applying dynamic programming using pre-determined drive cycles. By analyzing these actions rule-based controllers are designed so that the results can be implemented on real vehicle controllers. These control algorithms calculate the desired values for the state-of-charge and the wheel slip for the vehicle and this information together with the actual data are used to supervise the subsystem controllers. Our control strategy is based on minimizing the fuel consumption and the wheel slip concurrently. The controller design problems are solved separately also and compared to the concurrent solution. Results show that promising benefits can be obtained from the concurrent approach for designing hybrid vehicles which display better fuel economy and vehicle stability.

Keywords: Concurrent Controllers, Hybrid Electric Vehicles.

ÖZET

PARALEL HİBRİD ARAÇLAR İÇİN ENERJİ YÖNETİMİ VE ARAÇ DENGİ KONTROLÇÜ ALGORİTMALARININ EŞ ZAMANLI OLARAK DİNAMİK PROGRAMLAMA YARDIMIYLA TASARLANMASI

Halil İbrahim Dokuyucu

Makina Mühendisliği, Yüksek Lisans

Tez Yöneticisi: Asst. Prof. Melih Çakmakcı

Ağustos, 2012

Paralel hibrid araçların kontrolcülerinin eş zamanlı tasarlanması üzerine çalışılmıştır. Çalışmamız esas olarak, geleneksel olarak ayrı ele alınan enerji yönetimi ve araç denge kontrolcü algoritmalarının eş zamanlı olarak tasarlanması üzerine yoğunlaşmıştır. Optimum sonuçlar dinamik programlama kullanılarak önceden belirlenen sürüş çevrimlerinde elde edilmiştir. Bu sonuçların analiz edilmesiyle gerçek araç kontrolcülerine uygulanabilecek kurala dayalı kontrolcüler tasarlanmıştır. Kontrolcüler, araç için istenen batarya şarj durumu ve araç tekerleği kaymasını hesaplamakta ve hesaplanan bu değerler ile gerçek batarya şarj durumu ve araç tekerleği kayma değerlerini kullanarak alt sistem kontrolcülerini gözetmektedir. Kontrol stratejimizin hedefi, araç yakıt tüketimi ve araç tekerleği kayma değerlerinin minimize edilmesi olarak belirlenmiştir. Kontrol problemleri ayrı olarak da çözülmüş olup eş zamanlı olarak çözülen kontrolcü ile karşılaştırma yapılmıştır. Elde ettiğimiz sonuçlar eş zamanlı kontrolcüler yaklaşımının daha iyi hibrid araçların tasarlanmasında faydalar sağlayabileceğini göstermektedir.

Anahtar sözcükler: Eş Zamanlı Kontrolcüler, Hibrid Elektrik Araçlar.

Acknowledgement

First, I would like to thank my advisor Asst. Prof. Melih akmacı. He was very helpful and kind during my graduate studies. He took a great role in this thesis. I would also like to thank our department chair Prof. Dr. Adnan Akay. He gave us great support in our academic research.

I would like to thank all the people from the Department of Mechanical Engineering at Bilkent University for the great environment of academic life.

Finally I would like to thank my family. They always believed in me and gave their very special support.

Bilkent, August 2012
Halil İbrahim Dokuyucu

Contents

Acknowledgements	v
List of Figures	ix
List of Tables	xii
1 Introduction	1
1.1 Motivation	1
1.2 Background	2
1.3 Objectives and Problem Statement	5
1.4 Contributions	6
2 Parallel Hybrid Electric Vehicle Model	7
2.1 Parallel Hybrid Electrical Vehicle	13
2.2 Simplified Model Used In Dynamic Programming	15
2.2.1 Vehicle Dynamics Model	16
2.2.2 Powertrain Model	18

2.2.2.1	Transfer Case Model	18
2.2.2.2	Engine Model	20
2.2.2.3	Battery Model	20
2.2.2.4	Electric Motor Model	21
2.2.2.5	Gearbox Model	22
2.3	Complex Nonlinear Model Used In Simulations	23
2.3.1	Vehicle Dynamics Model	23
2.3.2	Powertrain Model	26
2.3.2.1	Transfer Case Model	26
2.3.2.2	Engine Model	26
2.3.2.3	Battery Model	27
2.3.2.4	Electric Motor Model	29
2.3.2.5	Gearbox Model	29
3	Dynamic Programming	32
3.1	Dynamic Programming Overview	32
3.2	Problem Formulation	33
3.3	Implementation in MATLAB	34
3.4	Application to Automotive Control Problems	37
3.4.1	The Concurrent Problem	38
3.4.2	The Energy Management Problem	40

3.4.3	The Vehicle Stability Problem	45
4	Controller Development	51
4.1	The Vehicle Stability Controller	52
4.2	The Energy Management Controller	56
4.3	The Concurrent Controller	60
5	Conclusions and Future Work	68
5.1	Conclusions	68
5.2	Future Work	70
5.2.1	Real Time Application Aspects	70
5.2.2	Improvement of The Problem Formulation	73
	Bibliography	76
A	Codes	79

List of Figures

1.1	Automotive Controller Systems Network.	3
2.1	Series and Parallel Hybrid Vehicle Layouts.	8
2.2	Volvo V70, a series hybrid vehicle offered by Volvo.	9
2.3	Toyota Prius, a parallel hybrid vehicle offered by Toyota.	10
2.4	Toyota Corolla Equipped with VSC System.	11
2.5	Configuration of Parallel Hybrid Vehicle.	13
2.6	Integrated Transfer Case Unit.	14
2.7	Bicycle model used in [16].	16
2.8	Simulation Model.	24
2.9	Vehicle Dynamics Model.	25
2.10	Inside the Transfer Case Block.	27
2.11	Thermal Model of the Engine.	28
2.12	SOC Algorithm of the Battery Model.	29
2.13	Maximum Torque Selection of the Electric Motor.	30

2.14	Speed Calculation of the Gearbox Model.	31
3.1	Graphical representation of linear interpolation [5].	36
3.2	Interaction Between Control Problems.	39
3.3	FTP 75 Drive Cycle.	42
3.4	SOC Behavior of Energy Management Controller.	43
3.5	Optimal Operating Points of EM vs Concurrent Controllers. . . .	44
3.6	RRSR Behavior of Vehicle Stability Controller.	47
3.7	Optimal Operating Points of VSC vs Concurrent Controllers. . . .	48
3.8	Wheel Slip Comparison of Vehicle Stability and Concurrent Con- trollers.	49
3.9	Fuel rate comparison of concurrent and EM controllers.	50
4.1	Optimal Traces provided by DP process.	52
4.2	Wheel Slip Calculation of the Vehicle Stability Controller.	53
4.3	Control Signals of Vehicle Stability Controller.	54
4.4	Relationship Between Torque Demand and Wheel Slip.	54
4.5	Wheel Slip Behavior of the Vehicle with Vehicle Stability Controller.	55
4.6	Energy Management Controller Add-on Unit.	57
4.7	Relationship Between Torque Demand and SOC.	58
4.8	SOC Behavior of the Vehicle with Energy Management Controller.	59
4.9	Extracted rules for concurrent controller when wheel slip is high. .	61

4.10	Extracted rules for concurrent controller when wheel slip is low. . .	62
4.11	SOC behavior of the concurrent controller.	63
4.12	Wheel Slip Behavior of the Concurrent Controller.	64
4.13	Fuel rate behavior of the concurrent controller with UDDS.	66
4.14	Fuel rate behavior of the concurrent controller with Indian Highway. .	67
5.1	Controller Strategy Approach.	69
5.2	Signal Process of Concurrent Controller.	71
5.3	Signal Processing of the Concurrent Controller.	71
5.4	Onboard Control Scheme of the Concurrent Controller.	72

List of Tables

2.1	Vehicle Parameters.	12
2.2	Parameters of the Electric Motor.	22
3.1	Working Modes of Powertrain.	42
3.2	Working Modes of Powertrain.	47
3.3	Fuel Consumption and Wheel Slip Comparison over FTP75 Cycle.	48
4.1	Energy Management Controller Rules.	58
4.2	Critical Values of SOC and Wheel Slip.	60
4.3	Fuel Consumption and Wheel Slip Comparison over UDDS and Indian Highway.	66

To my beloved friend Evrim Kimyonok . . .

Chapter 1

Introduction

1.1 Motivation

In parallel with the rapid increase in population around the world, the need for personal mobility and transportation has reached high levels. Although vehicles make our daily life easier, the pollution caused by them is one of the major problems of the big cities and has overall adverse effects to the environment [1]. Automotive companies try to find alternative ways of operating vehicles in cleaner and more efficient ways in order to cope with the strict environmental regulations by the governments. For this purpose engineers have been working on developing promising technologies such as hybrid electric and fuel cell vehicles. However these new vehicles usually result in more complex systems compared to the vehicles equipped with a conventional powertrain and managing the complexity of such systems with improving the vehicle performance is an ongoing objective for researchers in the automotive field. For example by using hybrid powertrains, which combine two or more power sources in a single system, provides significant improvements in fuel efficiency and reduces the emissions until zero emission vehicle (ZEV) technologies are commercially feasible. However, the inclusion of the new power sources and the accompanying energy storage systems increase the complexity of the system extensively. The operation of today's vehicles involves

many different controller systems working together with each other in an efficient manner. In Fig.1.1, examples of these controller systems are shown with possible interactions among each other. The principal contribution of this research is the development of a control method that uses the interaction between energy management and vehicle stability controllers. As the interaction between these two problems grows, significant improvements in terms of fuel consumption and wheel slip can be achieved when optimization problems of these controllers are solved concurrently. The strong interaction between the two control problems is presented in Chapter 3. Rules are extracted based on the optimal traces obtained in the Dynamic Programming, *DP*, process. Results show that when energy management controller is deciding on torque split ratio the information about wheel slip provides more efficient fuel consumption behavior. For instance, the energy loss caused by wheel slip is converted to usable energy by regenerative braking. In addition to this, information about energy management reduces wheel slip values of the vehicle.

1.2 Background

Traditionally, energy management strategies for hybrid electric vehicles are developed considering powertrain dynamics only [9, 6, 4]. Our research shows the benefits of the interaction among the two controller problems which gives us better results when we consider vehicle stability when determining the energy management of the hybrid powertrain or vice versa. We propose concurrent design of two controllers communicating with each other by means of controller area network (CAN) units.

DP is performed in order to obtain the optimal trace of controller outputs given the reference set-point data for the system [15]. The vehicle parameters in this model are updated according to a parallel hybrid vehicle configuration which we have also developed a complex and nonlinear simulation model based on actual vehicle data to be used in the controller development section of our research. Also vehicle longitudinal dynamics of our model is updated according to a bicycle

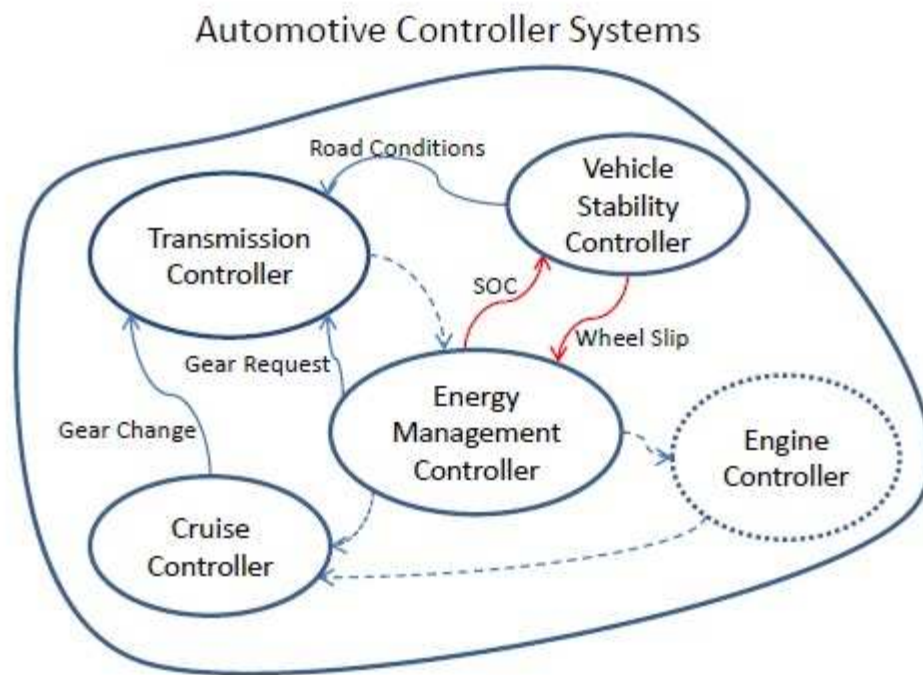


Figure 1.1: Automotive Controller Systems Network.

model, which involves longitudinal dynamics only, including torque split device between front and rear axles in order to be used in developing vehicle dynamics controller algorithm.

There are many studies in literature on the design and performance of energy management and vehicle dynamics controllers. In [8], optimal energy-management strategies for parallel hybrid powertrain of urban buses are studied. The parallel hybrid powertrain model is developed with a one way clutch and automatic transmission. Heuristic control techniques such as logic threshold power split strategy is offered regarding the instantaneously optimized algorithm. The controller proposed in this research tries to minimize the fuel consumption of the vehicle and to make the state-of-charge (SOC) of the battery to stay balanced. The designed controller is mentioned in two steps. First, simple rules arranging the power split between the engine and the electric motor are embedded into the controller architecture and then instantaneously optimization algorithm is applied in order to maximize the controller performance.

Optimal control theory is used when developing the control strategy in the research presented in [6]. Two vehicle models are used in this study. First a general complex model is constructed in order to be used in the controller tests over predefined drive cycles. This model is simplified in order to be used in the optimization process. The optimal control theory is applied and an optimization algorithm is obtained. This algorithm is improved by integrating a more accurate battery modeling regarding long trips of the vehicle in which the battery may diverge from its nominal values. Computation time for the optimal solution is tried to be reduced in this study. Adaptive control strategy is offered in [17]. Stochastic dynamic programming is used in the development of the control strategy and predictive algorithms are proposed in order to use the controller in real-time applications. The hybrid vehicle is modeled as a stochastic system and the powertrain operation is modeled as a stochastic process. The problem is formulated such that the controller decides on the power split between the engine and the electric motor under the uncertainty conditions. The Markov Decision Process is used to solve the stochastic problem. The controller results are compared to the ones using the rule-based algorithms.

A vehicle stability enhancement control algorithm is proposed in [9]. The vehicle is modeled as a four-wheel drive hybrid electric vehicle in which there exist two separated electric motors at the rear and the front axles. The benefits of eliminating the transfer case unit which splits the total torque between the rear and the front axles are obtained by using separated electric motors. The rear axle is driven only by the electric motor whereas the front axle is driven by both the engine and the electric motor. In the proposed system rear motor control and electrohydraulic brake control is used in order to enhance the vehicle stability during cornering. Also regenerative braking is used in order to improve the fuel economy of the vehicle. Fuzzy logic rules are generated in the proposed control algorithm.

In the thesis presented in [16], simultaneous controller optimization of traction controller for electric vehicles is proposed. Magic Formula is used when modeling the tire longitudinal force which is a function of wheel slip. The controller problem is formulated including continuous and discrete optimization variables so that a genetic algorithm using mixed encoding is proposed in the control strategy development. The optimization process is performed in order to minimize the energy consumed and completion time of a specified drive cycle which has varying surface conditions in the sense of coefficient of friction.

Several methods are offered for developing control strategies of energy management and vehicle stability controllers. In this thesis we have chosen rule based controllers including optimization by using dynamic programming for our control strategy due to its simplicity and successful examples in many areas.

1.3 Objectives and Problem Statement

In the thesis presented here, the main objective is to show the possible benefits of considering the interaction between the two controllers working in the same physical plant during the design phase. Rule extraction methods are used to obtain nearly optimum hybrid electric vehicle control algorithms compared to

the optimum ones obtained by DP.

Although communication of different controllers via CAN units in vehicle control architecture is well known in literature, our method differs by proposing the problem definition in which we show the interaction between the two control problems by means of the parameter's selection, i.e. state variables of concurrent controller problem are used in the constant parameters set when defining the single controller problems.

1.4 Contributions

The contributions of this thesis can be listed as follows:

1. An analytical method of solving two automotive control problems concurrently via DP and developing rule based control algorithms based on the optimal traces gained in DP.
2. A DP for obtaining the optimal traces of the vehicle stability controller and showing that DP works properly in applications of long drive cycles, in contrast to other studies using short profiles with constant road conditions, which have varying road conditions.
3. The integration of a wheel slip model into the hybrid powertrain simulation model which is based on actual vehicle data. Wheel slip and SOC signals network providing the communication between the two control algorithms.

Chapter 2

Parallel Hybrid Electric Vehicle Model

Hybrid electric vehicles can be classified in two basic powertrain configurations: series hybrid and parallel hybrid. Other configurations which combine series and parallel hybrid features in one powertrain can also be defined. In Fig.2.1, both series (right) and parallel (left) hybrid layouts are described.

In series hybrid vehicles the energy for the electric motor is generated by the engine. The electric motor provides the propelling of the vehicle. As there is no mechanical coupling between the engine and the wheels the optimal operating range of the engine for fuel economy can be achieved most of the time. On the other hand the energy conversion steps between the engine, the electric motor and the wheels result in energy losses of the powertrain. The series hybrid is more efficient during urban driving conditions where the engine speed is expected to vary most of the time. In Fig.2.2, a series hybrid vehicle introduced by Volvo in 2012 is shown. This is a plug-in series model which the battery of the vehicle can be charged off duty by plugging it into any electric source. In the low states of the battery the engine starts to provide power for the vehicle.

In parallel hybrid vehicles the engine and the electric motor are both coupled with the wheels. This coupling is provided by the series of clutches and geartrain,

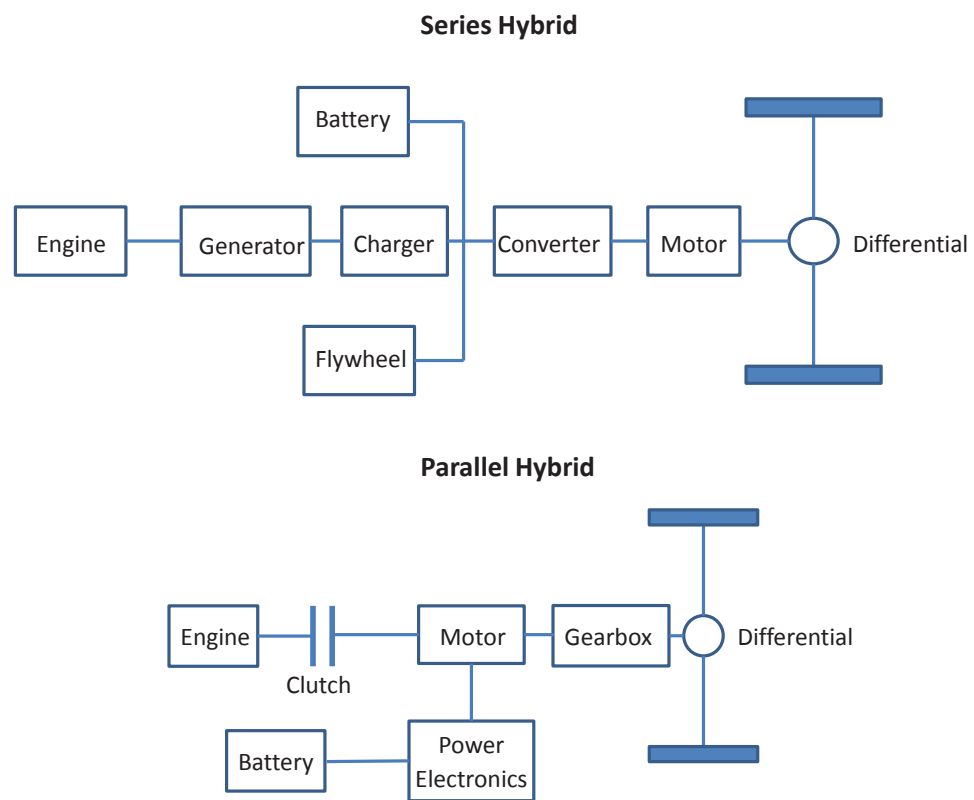


Figure 2.1: Series and Parallel Hybrid Vehicle Layouts.



Figure 2.2: Volvo V70, a series hybrid vehicle offered by Volvo. Available from <http://www.carpages.co.uk/volvo/volvo-25-09-09.asp>.

popularly termed as the power split hardware. They provide the propelling of the vehicle and also assist each other. In low torque demands of the driver the engine has the option to work in high torque level which is more efficient. The extra energy is used to charge the battery and used later. The electric motor has the capability of working as a generator in the braking mode of the vehicle by regenerative braking. The fuel economy benefits of the parallel hybrid configuration are achieved mostly in highway driving conditions. In Fig 2.3, one of the most recognized parallel hybrid electric vehicles commercially available today, Toyota Prius is shown. Toyota introduced the first generation of Prius in 2003. In Toyota Prius continuously variable transmission (CVT) is used in the torque coupling mechanism of the engine and the electric motor as well as in the mechanism of torque transfer to the wheels. Toyota promises to achieve 3.7

and 4.7 [L/100 km] fuel economy values in urban and highway driving conditions respectively. Also 1748 [kg/year] carbon dioxide (CO₂) emission is achieved.



Figure 2.3: Toyota Prius, a parallel hybrid vehicle offered by Toyota. Available from <http://www.mibz.com/10064-kbbs-green-cars-list-vw-golf-tdi-chevy-tahoe-hybrid-and-others.html>.

By definition it is in fact possible to use a parallel hybrid vehicle in series hybrid mode [12]. Researchers try to benefit from the advantages of series and parallel hybrid powertrains at the same time by designing new hybrid powertrain configurations in which series and parallel hybrid mode is selected by the driver due to the driving condition. When the driver selects the city driving mode the powertrain works as a series hybrid and parallel hybrid is activated by the selection of highway driving mode. Seven different parallel hybrid powertrain configurations were studied during the design stage of GM Parallel Hybrid Truck explained in [7]. The design objectives are examined on all powertrain configurations and one specific configuration which shows the best performance is selected.

Vehicle stability control systems are first introduced in the late 1980s by BMW in the name of Electronic Stability Control (ESC). This control system uses the



Figure 2.4: Toyota Corolla Equipped with VSC System. Available from <http://www.autoreview2u.com/toyota-corolla-2011-with-new-interior-car-design/>.

engine torque delivery system by reducing the torque provided by the engine in critical conditions. The vehicle stability control systems were taken into account by the automotive producers and rapidly commercialized since then.

In Fig.2.4 new model of Toyota Corolla which uses the latest vehicle stability control (VSC) system developed by Toyota is shown. Skidding condition is detected by the skid sensors of the vehicle and braking or accelerating command is sent through the wheels which need to be adjusted. VSC system also shows control actions during the cornering maneuvers of the vehicle. It prevents the vehicle to get into understeer or oversteer situations.

For the research presented here, we have chosen to work on the parallel hybrid powertrain configuration because of its generality. Since we are studying the coupling effects between energy management and vehicle stability controllers we expect that our results will be more general when the parallel hybrid powertrain configuration is chosen. To this end, we developed a parallel hybrid powertrain

Table 2.1: Vehicle Parameters.

Component	Component Parameters
Internal Combustion Engine (SI)	4 cylinders 2.2 liters 84 kW (peak power) 250 kg (mass)
Permanent Magnet Electric Motor (brushless)	53 kW (peak power) 248 N.m (peak torque)
Nimh Ovonic Battery	28 Ah (capacity) 50 (number of modules) 6 V/module (nominal voltage) 48.6 Wh/kg (Energy density) 444.4 W/kg (Power density)
Manual Gearbox	5 Speed, GR: 3.45, 1.94, 1.29, 0.97, 0.75
Vehicle	Body mass: 800 kg

model in Simulink based on actual vehicle data and a typical powertrain configuration. This is a complex nonlinear plant model based on empirical data from actual vehicles driven by realistic control algorithms which we used as our verification model as our rule-based vehicle control strategy is developed. This model is presented in section 2.3.

For the DP procedure, a simplified model based on [15] is developed using our complex simulation model vehicle parameters. Longitudinal dynamics of the vehicle are modeled using the bicycle model ignoring the lateral dynamics and the transfer case model in [9] is used to distribute the total torque between front and rear axles. Vehicle component parameters used for our research for both the complex nonlinear and the simplified models are given in Table 2.1. The simple model is suitable for studying both the energy management and vehicle stability problems.

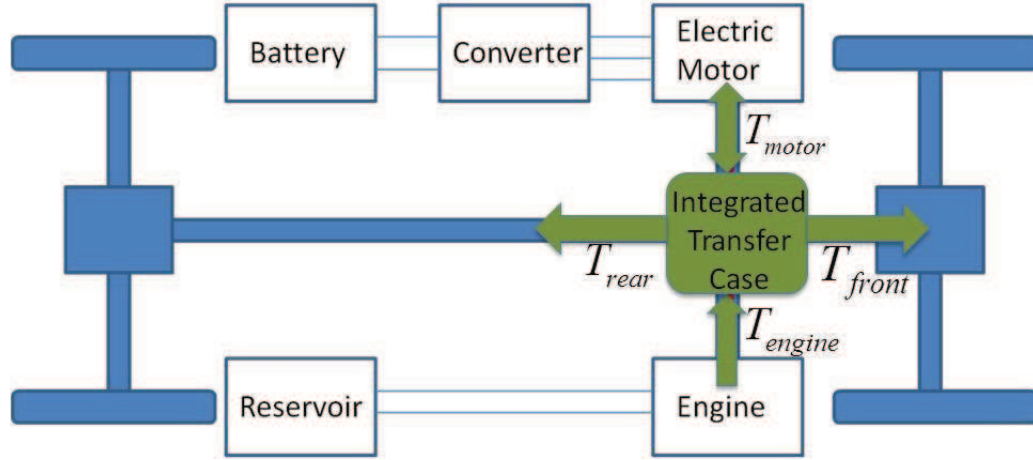


Figure 2.5: Configuration of Parallel Hybrid Vehicle.

2.1 Parallel Hybrid Electrical Vehicle

We used a typical parallel hybrid electric vehicle configuration with an integrated transfer case as shown in Fig.2.5. The power provided by the electric motor and the power provided by the engine are mechanically coupled in the integrated transfer case unit.

Typical transfer case units split the power between the rear and the front axles. We also integrated the power split mechanism between the electric motor and the engine into the transfer case unit. Total torque demand of the vehicle is met by the electric motor and the engine. Torque Split Ratio (TSR) is the control signal which determines the power split between the electric motor and the engine. Lower level clutch and gearbox control dynamics which handles the transients of the integrated transfer case is assumed to be negligible. That is for example it is assumed that coupling is achieved with no delays.

The integrated transfer case unit also distributes the total power between rear and front axles. Torque Split Factor (TSF) is the control signal which determines the power split between rear and front axles. The torque split between rear and front axles is used to provide vehicle stability during acceleration and braking. In the integrated transfer case unit total torque is mechanically split and transferred to

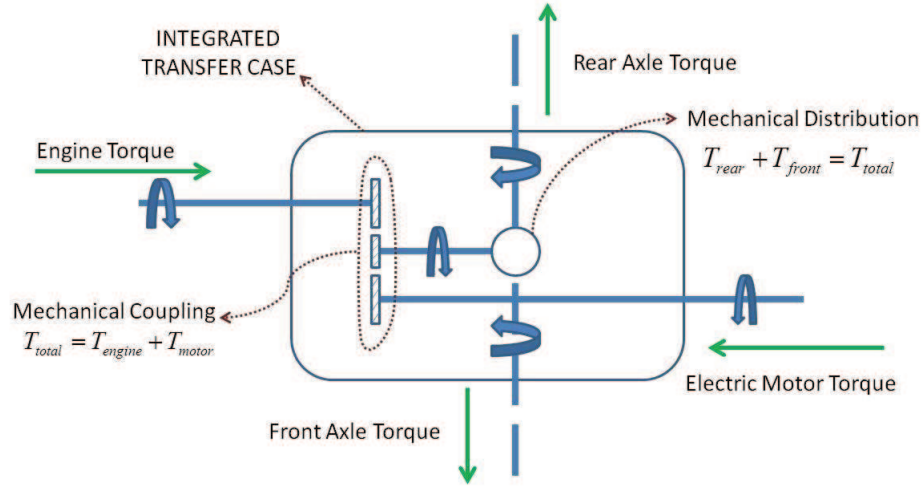


Figure 2.6: Integrated Transfer Case Unit.

both axles. The physical plant of the integrated transfer case is complicated and has some energy losses during its operation. In Fig.2.6, the integrated transfer case and power split mechanisms are explained.

As shown in Fig.2.6 there are two split mechanisms in the powertrain. Power split between electric motor and engine is determined by energy management controller and torque distribution between front and rear axles is determined by vehicle stability controller. In our proposed controller system the target split levels are controlled at the same time by the concurrent supervisory controller. The engine can operate in more efficient ranges with the help of the electric motor. At high loads the electric motor provides assistance to the engine whereas in low loads the engine can be shut off or works as a generator for the electric motor. In addition, the supervised distribution of the total torque provided to the wheels between the rear and the front axles by the transfer case reduces the wheel slip of the vehicle. Our powertrain model allows us to control the power split between the electric motor and the engine and torque distribution between the rear and the front axle at the same time.

Other configurations can be used as an alternative to the power split mechanisms

used here. One possible configuration is that the rear axle is propelled by the engine while the front axle is propelled by the electric motor. In this configuration, the power split decision between the engine and the electric motor needs to interact with the power split decision between the rear axle and the front axle in such a manner that control signals of the power split mechanisms are not separated from each other.

For another possible configuration, there are two electric motors mounted on each axles and the engine providing power for each axle. In this configuration, transfer case unit is used for only the power split of the engine between rear and front axles while two mechanical coupling units in each axle are needed for the power split mechanisms between the engine and the electric motor.

In our research, we use more general powertrain configuration rather than specific configurations. The results of our research presented in this thesis are generic enough so that they can be extended to specific powertrain configurations easily.

2.2 Simplified Model Used In Dynamic Programming

Our simulation model is based on empirical data from actual vehicles which has many numbers of dynamic states and high level of complexity. This model is a useful tool for calculating the validity of the controllers using heuristic control techniques such as rule extraction [16].

As mentioned earlier dynamic programming (DP) is also used in the research presented in this thesis. The performance of DP is highly related with the accuracy and the number of states of the model used for DP. Nonlinear models result in inaccuracy and the computation burden of the process is high.

For the DP procedure, a simple but functional mathematical model is preferred rather than a complex nonlinear model. The dynamic states of the complex nonlinear model are degraded based on the simplification methods introduced in

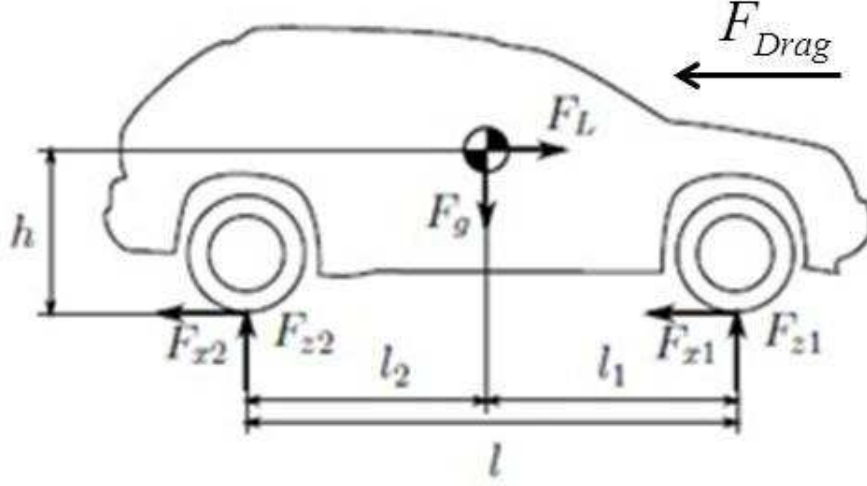


Figure 2.7: Bicycle model used in [16].

[15]. The simplified model is left with two dynamic states, namely SOC and the wheel slip. Based on the characteristics of the DP, which is explained in detail in Chapter 3, these dynamic states are used individually or together in the model.

The components that take part in the simplified hybrid vehicle model are explained in the sections 2.2.1-2.2.2.

2.2.1 Vehicle Dynamics Model

The vehicle dynamics are modeled using the bicycle model without the lateral dynamics. Dynamic weight transfer between front and rear axles is considered due to the vehicle acceleration. The model used in [16] is followed. The bicycle model used in this study is shown in Fig.2.7.

The inputs of the vehicle model are vehicle velocity, v_{veh} , and vehicle acceleration, a_{veh} . Predetermined drive cycle data in forms of vehicle velocity and acceleration is used for simulations.

As this is a bicycle model including the longitudinal dynamics only, vehicle mass is modeled as the half of the total vehicle mass. At the contact points between tires and the road there are the reaction forces, F_{z1} and F_{z2} , due to the gravitation, g :

$$F_g = mg = F_{z1} + F_{z2} \quad (2.1)$$

where

$$F_{z1} = mg \frac{l_1}{l}, \quad F_{z2} = mg \frac{l_2}{l}. \quad (2.2)$$

Longitudinal tire forces are produced with propulsion or braking action of the vehicle. There is a linear relationship, shown in equation (2.3), between the tire normal forces, obtained in equation (2.2), and maximum tire longitudinal forces $F_{xi,max}$, which limits the tire friction forces. The road friction coefficient μ , is assumed to be uniform. In equation (2.3), F_{xi} denotes the actual friction forces between the tire and the road.

$$|F_{xi}| \leq |F_{xi,max}| = \mu F_{zi} \quad \text{for } i = 1, 2 \quad (2.3)$$

The drag force as a function of vehicle velocity, v , and drag coefficient, C , can be described as

$$F_{drag} = Cv^2. \quad (2.4)$$

By using Newton's second law we can find the relationship between the vehicle acceleration and the longitudinal tire force.

$$a_{veh} = \frac{F_{x1} + F_{x2}}{m} = \frac{F_x}{m} \quad (2.5)$$

where F_x is the net external longitudinal tire force and it is limited by the front and rear maximum longitudinal tire forces as

$$|F_x| \leq |F_{x1,max}| + |F_{x2,max}| = \mu(F_{z1} + F_{z2}) = \mu F_g. \quad (2.6)$$

With equations (2.5) and (2.6) we can obtain a limit for the acceleration. The condition for this limitation is given as

$$|a_{veh}| \leq \mu g. \quad (2.7)$$

Before analyzing the tire normal forces the longitudinal dynamics can be written as a static system by considering d’Alambert’s Principle

$$F_x + F_L = 0 \quad (2.8)$$

where F_L denotes the inertial force associated with accelerating/decelerating status of the vehicle in the above equation. By using equations (2.5) and (2.8) F_L is found as

$$F_L = -ma_{veh}. \quad (2.9)$$

Weight transfer under vehicle acceleration is modeled as

$$F_{z1} = F_g \frac{l - l_1}{l} + \frac{F_L \times h}{l}, \quad (2.10)$$

$$F_{z2} = F_g \frac{l_1}{l} - \frac{F_L \times h}{l} \quad (2.11)$$

where F_{z1} and F_{z2} denote the vehicle traction forces of the front and the rear wheels respectively. F_g represents the weight of the vehicle and F_L represents the inertial forces of the vehicle. l is used for the distance between the axles of the vehicle whereas l_1 is used for the distance between the center of gravity of the vehicle and the front axle. h is used for the height of the center of gravity of the vehicle. In equations (2.10)-(2.11) first terms on the right hand represent the static weight distribution and second terms represent the dynamic weight distribution.

2.2.2 Powertrain Model

2.2.2.1 Transfer Case Model

As discussed before rear and front axle torque values can be different from each other while providing the vehicle traction stability. In order to distribute the torque between front and rear axles we need to use a mechanical accessory like a center differential. There exists complex transfer case models such as those used in [4] but for simplicity we used a zero order model as in [9]. The inputs of the

model are the total torque produced, inertia, rotational speeds of the front and rear axles. The outputs are the torque values of front and rear axles.

Output torque is calculated as

$$T_{out} = ratio \times T_i. \quad (2.12)$$

Front and rear torque values are determined via factor of torque split, (K), as shown below:

$$T_{out,front} = T_{out} \times K_{front}, \quad (2.13)$$

$$T_{out,rear} = T_{out} \times K_{rear}. \quad (2.14)$$

Factor of torque split, K_{rear} , is a function of rear rotational speed ratio (RRSR), $\omega_{ratio,rear}$ which is calculated as

$$\omega_{ratio,rear} = 0.5 + \frac{\omega_{rear} - \omega_{front}}{0.5 \times (\omega_{rear} + \omega_{front})}. \quad (2.15)$$

In equation (2.15), ω_{rear} and ω_{front} denote the rear axle and front axle rotational speeds respectively. In practice, the control law uses wheel slip instead of $\omega_{ratio,rear}$. It should be noted that wheel slip is a function of $\omega_{ratio,rear}$:

$$wheel\ slip = \omega_{ratio,rear} - 0.5. \quad (2.16)$$

The function, $f(\omega_{ratio,rear})$, is to be the control law of the vehicle stability controller.

$$K_{rear} = f(\omega_{ratio,rear}) \quad (2.17)$$

RRSR is the dynamic state of the model and it depends on the speed difference of front and rear axles. This ratio is the function of the wheel slip of the vehicle and the objective of the vehicle stability controller is to make the RRSR value at 0.5, i.e. zero slip.

The split factors of rear and front axles sum up to unity:

$$K_{front} = 1 - K_{rear}. \quad (2.18)$$

The output rotational speed is calculated as

$$\omega_{in} = ratio \times \omega_{out}. \quad (2.19)$$

ratio denotes the reduction in the transfer case model and it is unity, i.e. the rotational speeds of the electric motor and the engine are assumed to be the same.

The mean value of the rotational speeds of front and rear axles, ω_{out} , which is defined in equation (2.20), which appears in equation (2.19).

$$\omega_{out} = \frac{(\omega_{rear} + \omega_{front})}{2} \quad (2.20)$$

2.2.2.2 Engine Model

A quasi-static model is used for the engine. The static map obtained from the actual vehicle data determines fuel consumption rate.

Quasi-static model consists of a steady state model to which an equivalent dynamical model of the system is added. The output torque of the engine is modeled as

$$T_{engine,out} = T_{eng} - \frac{P_{loss,eng}}{\omega_{eng}}. \quad (2.21)$$

Here $P_{loss,eng}$ represents the frictional losses of the engine. $P_{loss,eng}$ is assumed to be a constant average value whereas in complex model it is assumed to be varying with respect to the engine speed. Constant fuel rate is assumed during the idling.

2.2.2.3 Battery Model

The open circuit voltage of the battery, V_{oc} , is governed as

$$V_{oc} = \frac{q_{batt}}{C_{batt}}. \quad (2.22)$$

Here q_{batt} represents the charge of the battery. The capacitance of the battery, C_{batt} , depends on the internal temperature and current of the battery.

SOC of the battery is modeled as a normalized value, which represents the charge capacity of the battery

$$SOC = \frac{V_{oc} - V_{min}}{V_{max} - V_{min}} \quad (2.23)$$

where V_{min} and V_{max} represent the minimum and maximum allowable voltage of the battery. These minimum and maximum values are selected as 250V and 400V respectively based on the electric motor used in the modeling.

The provided voltage of the battery can be shown as

$$V_{out} = V_{oc} - i_{batt}R_{batt} \quad (2.24)$$

It should be noted that transients and thermal effects are neglected. So the only dynamic state is SOC value of the battery.

2.2.2.4 Electric Motor Model

Electric motor dynamics are faster than battery dynamics. The motor model developed does not have a dynamic state. The losses of the motor are taken into account considering the output torque and speed.

The power needed by the motor is a mapped function of output torque, T_{mot} , and speed, ω_{mot}

$$P_{batt} = P_{map,motor}(T_{mot}, \omega_{mot}). \quad (2.25)$$

The maximum allowable current of the motor, $i_{max,mot}$, limits the maximum electric power, $P_{max,mot}$, supplied from the motor

$$P_{max,mot} = i_{max,mot}V_{out}. \quad (2.26)$$

The mechanical torque limit of the motor, $T_{max,mech}$, is given as

$$T_{max,mech} = T_{peak}(\omega_{mot}) - HI[T_{peak}(\omega_{mot}) - T_{cont}(\omega_{mot})]. \quad (2.27)$$

Table 2.2: Parameters of the Electric Motor.

Parameter	Value
$i_{max,mot}$	475 A
$P_{max,mot}$	53kW
$\omega_{max,mot}$	8000 rpm

Here peak torque value of the motor, $T_{peak}(\omega_{mot})$, and the continuous torque value of the motor, $T_{cont}(\omega_{mot})$, are the mapped functions of rotational speed of the motor, ω_{mot} . The heat index HI appeared in the above equation arranges the available torque between the peak and the continuous torques during its operation.

The maximum allowable electrical power available to the motor and the motor speed also limit the output torque. Maximum allowable output torque of the motor limited electrically, $T_{max,elec}$, is

$$T_{max,elec} = P_{map,motor}^{-1}(P_{max,motor}, \omega_{mot}) \quad (2.28)$$

where $P_{map,motor}^{-1}$ represents the inverse map of the motor. In Table 2.2 limitation parameters of the electric motor are given.

2.2.2.5 Gearbox Model

Gear ratio, G_T , and loss terms, $T_{loss,gb}$, are considered when modeling the gearbox. The torque passing through the gearbox, T_{gb} , can be described as

$$T_{gb} = G_T \times (T_{out,tc} - T_{loss,gb}) \quad (2.29)$$

where $T_{out,tc}$ represents the output torque of the transfer case. The gearbox losses, $T_{loss,gb}$, are assumed to be constant.

Wheel torque, T_{wheel} , is described with final drive ratio, g_f , output torque of the gearbox, $T_{out,gb}$, and loss terms of the final drive, $T_{loss,fd}$, included as

$$T_{wheel} = g_f \times T_{out,gb} - T_{loss,fd}. \quad (2.30)$$

2.3 Complex Nonlinear Model Used In Simulations

In our controller development stage of research we developed a complex and nonlinear simulation model in Matlab\Simulink environment based on actual data. For developing this model Autonomie simulation software which is based on the several actual vehicle and component tests of Argonne National Laboratory are taken as the reference point when developing our simulation model which is driven by realistic controllers. We configured our model by selecting the appropriate model blocks from the libraries of Autonomie. The original controller blocks are upgraded according to the control algorithms we developed which will be explained in detail in the controller development section.

In Fig.2.8, the high level model blocks representation of our complex nonlinear simulation model is given. Two power paths provided by the engine and the electric motor are separated. The total torque is obtained by summing up the torques provided by the engine and the electric motor. The total torque is then distributed to the rear and the front axles via the integrated transfer case unit. The model blocks of the rear and the front axles are separated from each other as shown in the above figure. The outputs of the powertrain are collected by signal processing blocks and feed the controller blocks.

2.3.1 Vehicle Dynamics Model

Vehicle dynamics model is separated to rear and front axle dynamics including the bicycle model ignoring the lateral dynamics. Vehicle dynamics model is divided into three subblocks, namely differential block, wheel block and axle block. Output torque of the transfer case is the input and the vehicle speed is the output of the vehicle dynamics model. The system is fed by the control signals of energy management and vehicle stability controllers. These signals are processed by the wheel model.

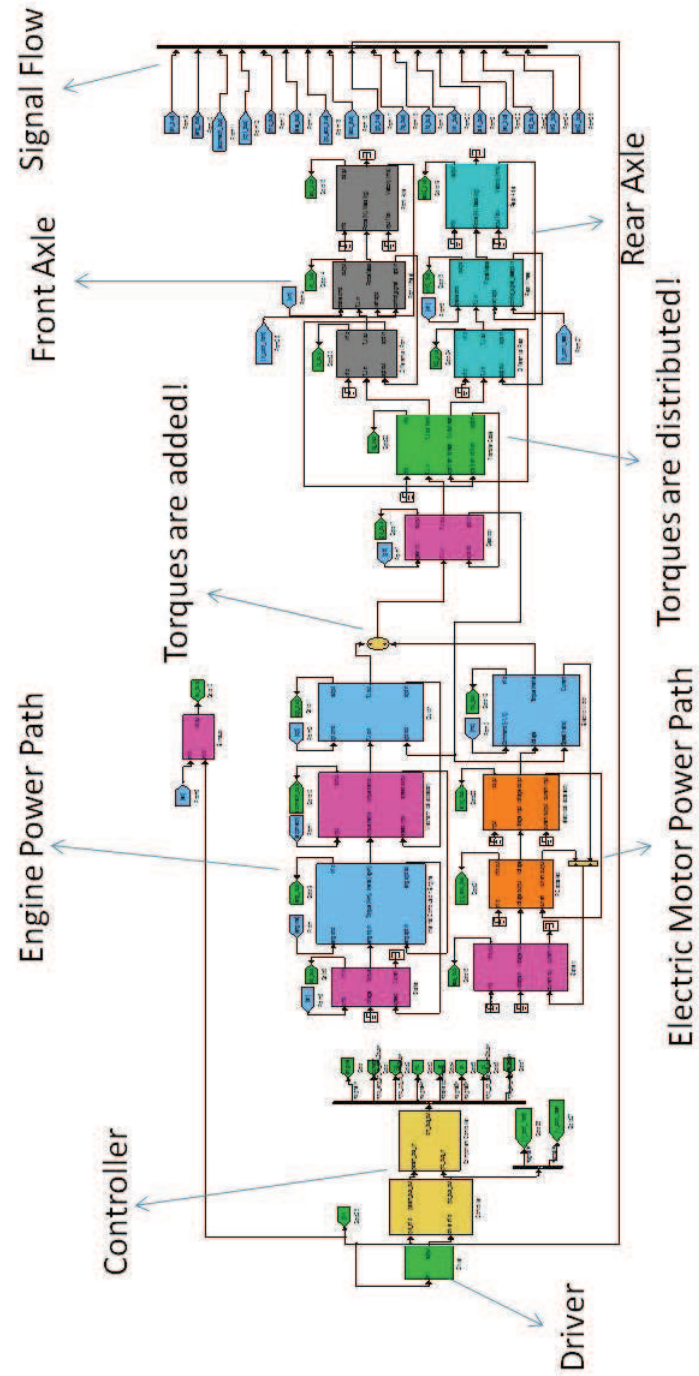


Figure 2.8: Simulation Model.

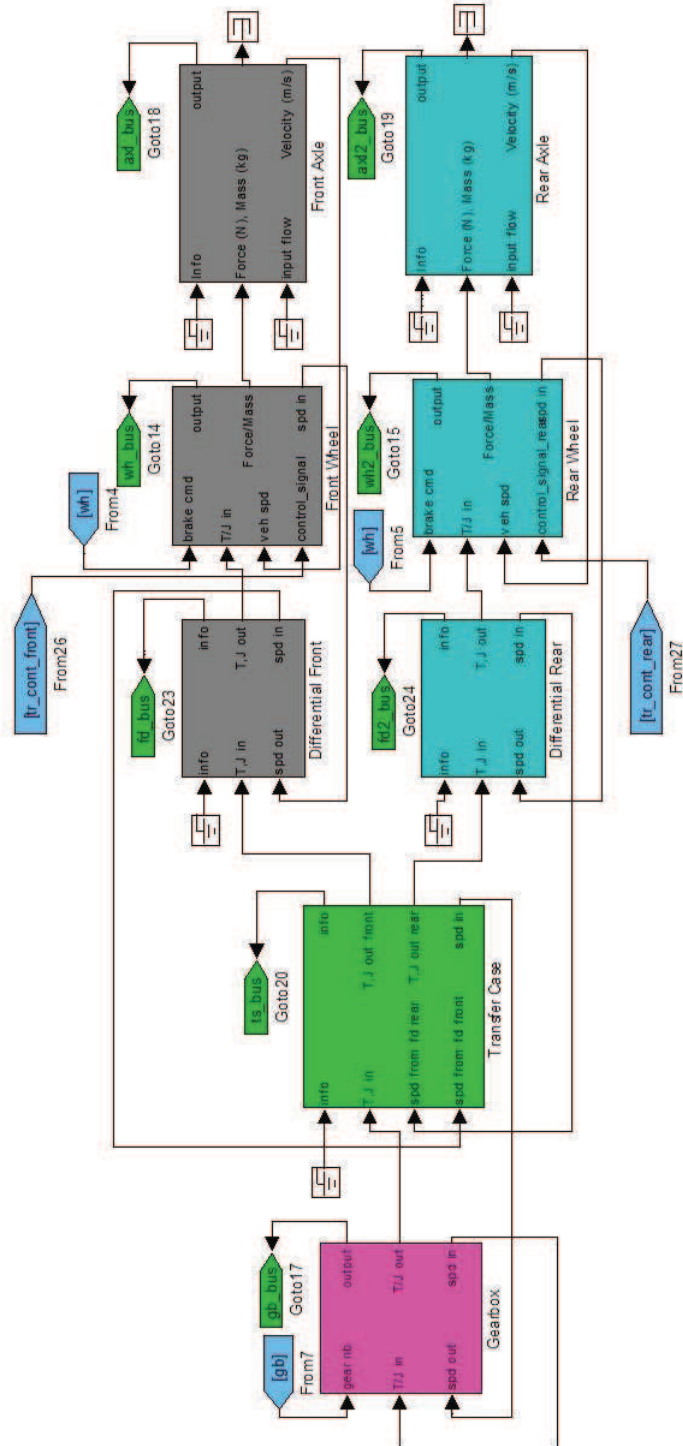


Figure 2.9: Vehicle Dynamics Model.

In Fig.2.9, the model blocks providing the vehicle dynamics are shown. Since the vehicle longitudinal dynamics are modeled only steering angle is neglected in the wheel model. The tire forces generated by cornering are excluded in the model. It should be noted that control signals are used in the wheel models. In wheel model a feedback traction controller is embedded using the control signals. This traction controller will be explained in detail in section 4.3.

2.3.2 Powertrain Model

Powertrain model includes two power paths adding up in the transfer case. As this is a mathematical model summation is done before the gearbox block in our simulation model. Controller blocks feed the powertrain during its operation.

2.3.2.1 Transfer Case Model

The transfer case is modeled as having an input of the total output provided by the electric motor and the engine and giving an output of the distributed torque to the axles. Front and rear wheel speeds values drive the mechanism of determining the power split between rear and front axles.

Fig.2.10 shows the model block representation of the integrated transfer case unit. The weighted speed of the rear and the front axles is calculated by taking the mean value of the sum of the rear and the front axle speeds. Rear rotational speed ratio is calculated via a specific function. The rear and the front torque ratio values are calculated by a look-up table embedded into the integrated transfer case unit.

2.3.2.2 Engine Model

When modeling the engine Autonomie map functions are heavily used in calculation steps of engine torque, fuel rate, exhaust emissions. Thermal model of the engine is also embedded in the engine model.

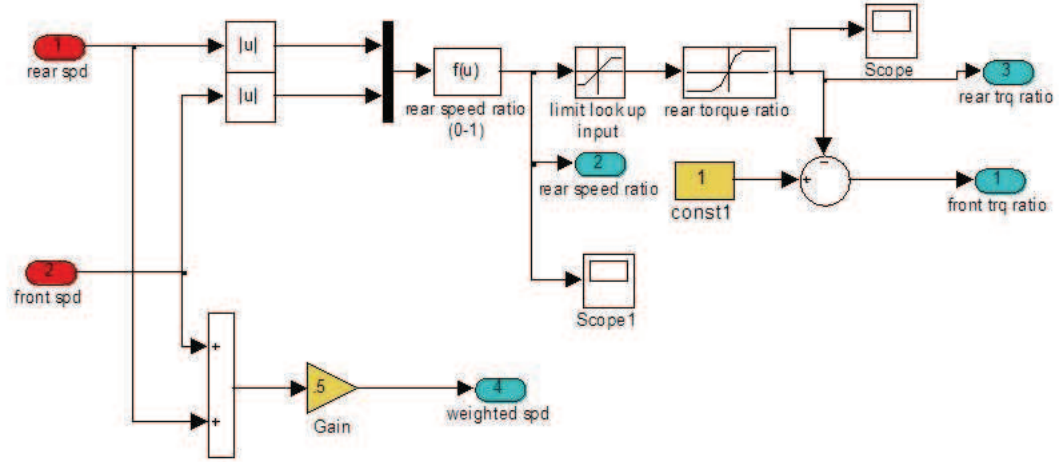


Figure 2.10: Inside the Transfer Case Block.

In Fig.2.11, the model block representation of the thermal model of the engine is given. The heat rejection characteristics of the engine is calculated based on the actual values of the engine speed, the engine torque and the fuel rate.

2.3.2.3 Battery Model

Battery model consists of voltage and SOC calculation based on mapped functions. Temperature effects are also included in the model. SOC algorithm highly depends on the temperature effects of the battery.

The calculation process of SOC value of the battery is shown in Fig.2.12. A look-up table determines the maximum power capacity of the battery with respect to the temperature value of the battery. This maximum power capacity and the consumed power values are used in the calculation process of the SOC.

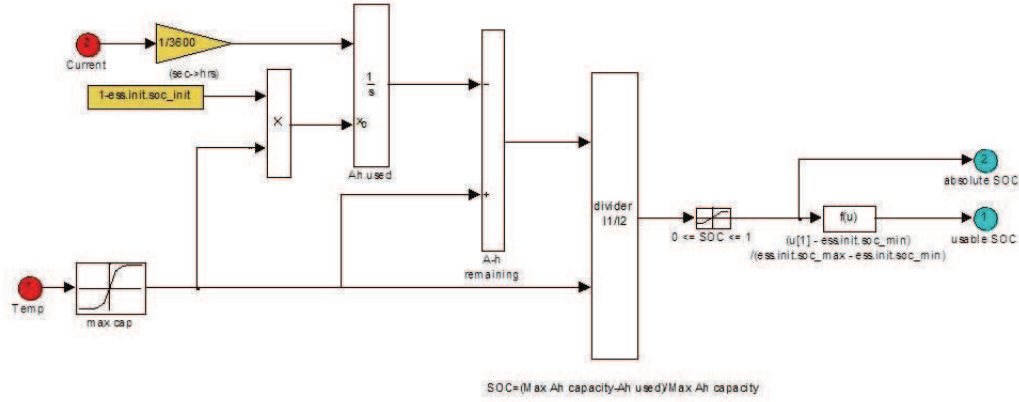


Figure 2.12: SOC Algorithm of the Battery Model.

2.3.2.4 Electric Motor Model

Battery voltage is the main input to the electric motor model. Torque provided by the electric motor which is driven by the control signal of power split between the electric motor and the engine is the output. In Fig.2.13 the maximum torque selection model block is represented. Peak and continuous torque values for propelling and the regenerative modes of the electric motor are determined based on the look-up tables using the actual speed of the electric motor. There is a switch mechanism between propelling and the regenerative modes based on the control signal of the electric motor. Heat index HI is calculated in subsystem using the actual and the continuous torque values of the electric motor. The maximum torque selection process decides on whether the torque is limited mechanically or electrically.

2.3.2.5 Gearbox Model

Gearbox model gets the gear number information from the original controller blocks of Autonomie including the gear shifting strategy. As the appropriate



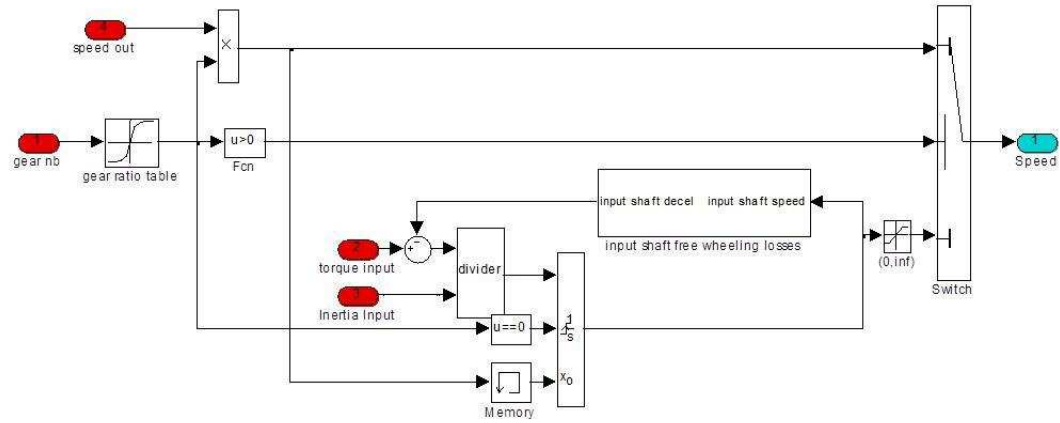


Figure 2.14: Speed Calculation of the Gearbox Model.

gear number is determined speed and torque calculation are done in the model. Gear table based on our vehicle parameters is entered into the model. Speed calculation process of the gearbox is shown in Fig.2.14. It should be noted that the free wheeling losses of the gearbox is modeled and used in the model when the gear number of the vehicle is zero.

Chapter 3

Dynamic Programming

Dynamic Programming (DP) is a design optimization tool which can be applied to systems with defined design objectives. It is applied in a number of areas such as information theory, bioinformatics, computer science and control theory [3]. In control theory, DP is used for obtaining the optimal outputs of controllers to be designed [13].

3.1 Dynamic Programming Overview

Bellman's principle of optimality constitutes the basics of DP. According to Bellman's principle, the optimality of subsequent control actions included in the main optimization problem should be validated on the entire space defined for the problem with respect to the resulting state of the first decision of the sequence. This optimal behavior is independent of the initial conditions of the system [2]. The optimal trajectories of subproblems build up the optimal trajectory of the main problem. The performance of the DP process depends highly on the dynamic states defined for the problem. The number of the states directly affects the computation burden and the memory needed for the DP process.

3.2 Problem Formulation

In order to formulate the problem to be solved with DP, continuous time models should be discretized first as shown in (3.1).

$$x_{k+1} = f(x_k, u_k, k) \quad (3.1)$$

Here x_k represents the state vector of the system, u_k represents the control actions of the system and k represents the time instant of the DP process. After the model is discretized for numerical optimization, a general DP problem is stated [2, 5]:

A general DP problem

given an objective J , solve

$$\min J = \sum_{k=0}^N L(x_k, u_k, k) \quad (3.2)$$

where

$$x_{k+1} = f(x_k, u_k, k) \quad (3.3)$$

subject to

$$x \in X_k \subset R^n, u \in U_k \subset R^m. \quad (3.4)$$

In 3.2-3.4 N denotes the length of the drive cycle. L is defined as the cost function of a single subsequent stage whereas J represents the total cost of the system. Constraints on system states, X_k , and control actions, U_k , can also be defined as shown in equation (3.4). The constraints of the system are stated as shown in equation (3.5).

$$X_k : x_{min} \leq x_k \leq x_{max} \quad (3.5)$$

$$U_k : u_{min} \leq u_k \leq u_{max}$$

x_{min} and x_{max} represent the minimum and maximum allowable values of the system states respectively. u_{min} and u_{max} represent the minimum and maximum allowable values of the control actions respectively.

The general DP problem defined in equations (3.2)-(3.4) can be solved by numerical methods according to the Bellman's principle of optimality [11].

In equations (3.6)-(3.7), J_k represents minimum cost function between the stage k and stage N . The problem formulation is shown as if it is starting from the state of x_k .

$$\begin{aligned}
J_k(x_k) &= \min_{u_k, \dots, u_N} \left[\sum_{j=k}^N L(x_j, u_j, j) \right] \\
&= \min_{u_k} \left[L(x_k, u_k, k) + \min_{u_{k+1}, \dots, u_N} \left[\sum_{j=k+1}^N L(x_j, u_j, j) \right] \right] \\
&= \min_{u_k} [L(x_k, u_k, k) + J_{k+1}(x_{k+1}, k+1)]
\end{aligned} \tag{3.6}$$

where

$$J_N(x_N) = \min_{u(N)} [L(x_N, u_N, N)] \tag{3.7}$$

3.3 Implementation in MATLAB

In our research we used the DP algorithm outlined in [14] with appropriate modifications based on our parallel hybrid vehicle model. In this section the DP algorithm in [14] is summarized. As the result of the research presented in [14] a generic Matlab function which solves deterministic optimization problems specifically for energy management strategy of parallel hybrid vehicles is developed. First the optimization problem is defined as outlined in equations (3.2)-(3.4). Continuous time model is discretized according to the equation (3.1).

Control policy, π , is defined as a function of control actions, μ_i , as

$$\pi = \mu_0, \mu_1, \dots, \mu_{N-1}. \tag{3.8}$$

The discretized cost function is rearranged as shown in equation (3.8) based on the control policy defined in equation (3.7) where initial state is defined as, $x(0) = x_0$.

$$J_\pi(x_0) = g_N(x_N) + \varphi_N(x_N) + \sum_{k=0}^{N-1} h_k(x_k, \mu_k(x_k)) + \varphi_k(x_k) \tag{3.9}$$

In equation (3.9) $g_N(x_N) + \varphi_N(x_N)$ stands for the final cost in which an additional penalty function is defined within as, $\varphi_N(x_N)$, which has an effect on the final constraints. The function of, $h_k(x_k, \mu_k(x_k))$, is defined as the cost function of the

control actions of $\mu_k(x_k)$. Another penalty function, $\varphi_k(x_k)$, which has an effect on system state constraints.

An optimal control policy is defined as shown in equation (3.10).

$$J^0(x_0) = \min_{\pi \in \Pi} J_\pi(x_0) \quad (3.10)$$

Here the control policy is minimized over a finite horizon. Π stands for the all possible control policies.

The DP algorithm calculates the optimal cost-to-go function at every stage going backwards based on the Bellman's principle of optimality as discussed earlier.

$$J_k(x^i) = \min_{u_k \in U_k} [h_k(x^i, u_k) + \varphi_k(x^i) + J_{k+1}(F_k(x^i, u_k))] \quad (3.11)$$

Linear interpolation is used to calculate the cost-to-go function, $J_{k+1}(F_k(x^i, u_k))$. The interpolation procedure is represented graphically in Fig.3.1.

The optimal control signal map is obtained as the result of the discretized cost function defined in equation (3.8). The optimal trajectory of the controller is obtained using this optimal signal map during the forward simulation of the model. It should be noted that the optimal cost-to-go function defined in (3.10) is calculated on the discretized points of the specified state space. However the function $F_k(x^i, u_k)$ which is included in the last term of the optimal cost-to-go function is a continuous function. When the output of this function stays between any two nodes on the state space the optimal cost-to-go function should be calculated with suitable approximation methods. Linear interpolation is used for the control signal when actual state does not match the discrete points.

Detailed description on how to use the “dpm function” which solves the discretized optimal control problem is given in [14] and also in the appendix. The structure of the function is given as:

$$[res\ dyn] = dpm[fun, par, grd, prb, options]$$

Here *fun* stands for handling the dpm function, *par* is the parameter structure defined by the user, *grd* is the grid structure, which is constituted by the nodes on

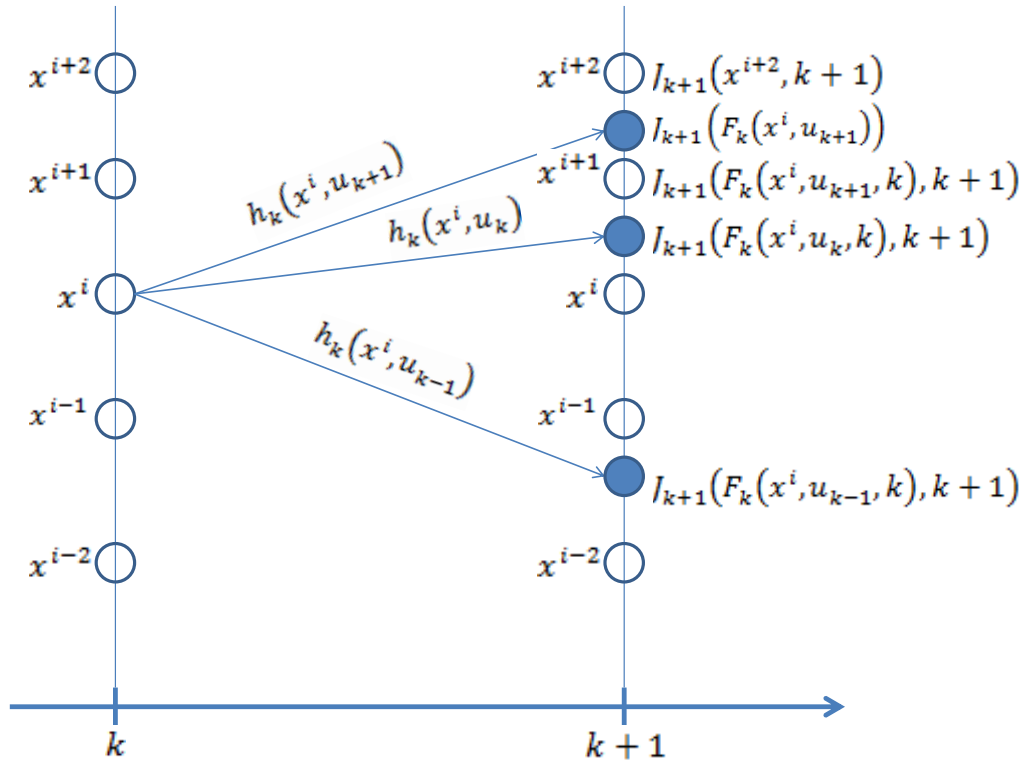


Figure 3.1: Graphical representation of linear interpolation [5].

which the discretized optimal cost-to-go function is solved, defined by the user, *prb* is the problem structure and *options* is the option structure selected by the user. As the output of the *dpm* function, two variable structures are created: output of the DP procedure and the optimal signal trajectory calculated using the control signal map.

3.4 Application to Automotive Control Problems

In the research presented in [14] an automotive optimization control problem is given as an example. A Quasi-static discrete-time model of an hybrid vehicle is used to define the fuel consumption of the vehicle. SOC of the battery is chosen as the only state variable of the model. The details of the mathematical modeling of the vehicle components are given in [14].

The discretized model is defined based on the model equations as described in equation (3.12).

$$x_{k+1} = f(x_k, u_k, v_k, a_k, i_k) + x_k \quad (3.12)$$

In equation (3.12) x_k denotes the SOC of the battery which is the only dynamic state of the model, u_k denotes the control signal of the model which is defined as the torque split between the engine and the electric motor, v_k , a_k , i_k , denote the vehicle speed, vehicle acceleration and the gear number of the vehicle respectively at the instance of k . As vehicle speed, v_k , vehicle acceleration, a_k , and the gear number, i_k , are known by the drive cycle defined for the system, the discretized model defined in equation (3.12) can be reduced to a simplified model:

$$x_{k+1} = f(x_k, u_k) + x_k, \quad k = 0, 1, 2, \dots, N - 1. \quad (3.13)$$

In equation (3.13), k denotes the instance of the calculation steps. The simulation time is set up by defining the end point of instance which is denoted as N .

Optimization control problem of the hybrid vehicle

Given an objective J , solve

$$\min_{u_k \in U_k} J = \sum_{k=0}^{N-1} \Delta m_f(u_k, k) \cdot T_s \quad (3.14)$$

where

$$x_{k+1} = f(x_k, u_k) + x_k \quad (3.15)$$

subject to

$$x_0 = 0.55, x_N = 0.55, x_k \in [0.4 \ 0.7] \quad (3.16)$$

and

$$N = \frac{660}{T_s} + 1 \quad (3.17)$$

Through the equation (3.14)-(3.17), the fuel consumption of the vehicle is denoted by the function, $\Delta m_f(u_k, k) \cdot T_s$. Time step, T_s is selected as $T_s = 1s$. The state space of the state variable is given in equation (3.16). The number of calculation step is calculated as considering the period of the drive cycle. In the research presented in [14] Japanese 10-15 (J1015) drive cycle is used.

The optimal control problem formulated in equations (3.14)-(3.17) is solved using *dpm* function.

In sections 3.4.1-3.4.3 the automotive control system design problems for our research using DP are described.

3.4.1 The Concurrent Problem

Since the concurrent control system is the main focus in our research, control systems for energy management and vehicle dynamics are presented concurrently first. In Fig.3.2 the interaction between the two control problems are explained. The two dynamics states, SOC and RRSR, are used together in concurrent problem formulation. These dynamic states communicate with each other during the DP procedure. The optimal solution of energy management problem at each

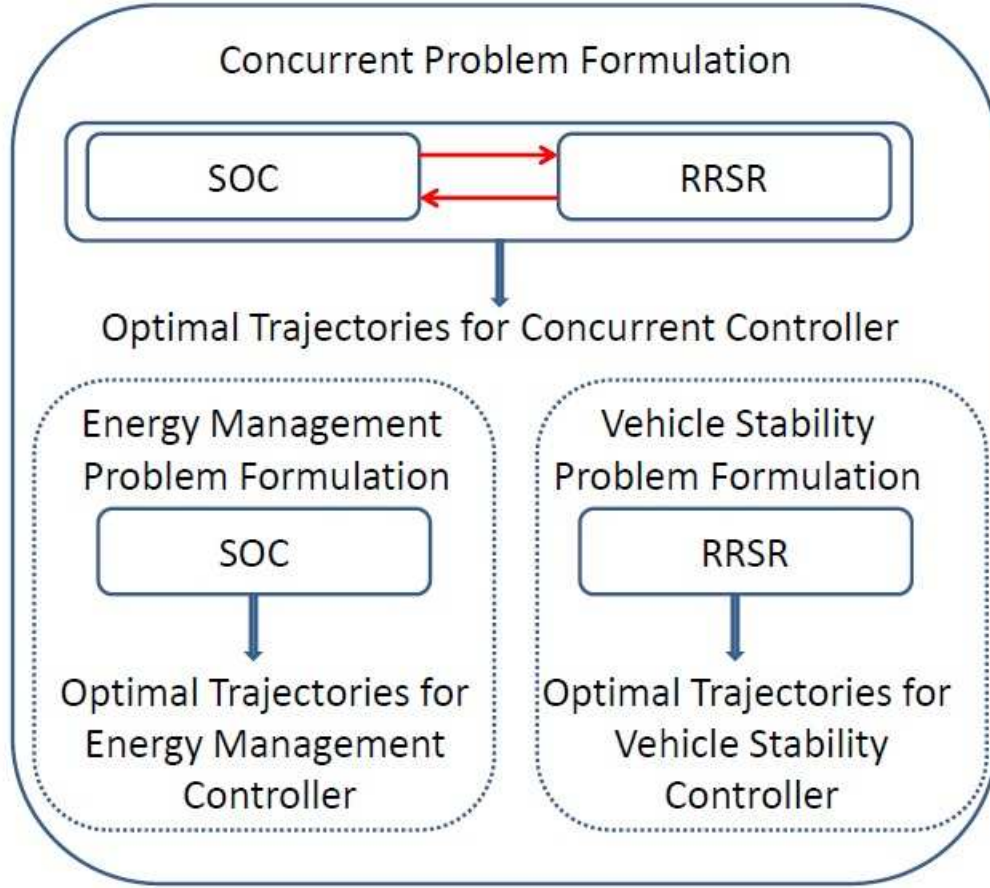


Figure 3.2: Interaction Between Control Problems.

instance is used when the vehicle stability is solved and vice versa.

This relation outlined in Fig.3.2 is stated in equations (3.18)-(3.20) in which the optimization control problem for concurrent controller is formulated for two dynamic states, namely SOC and RRSR.

The concurrent DP problem

Given an objective J , solve

$$\min_{TSR_k \in U_k^1(x_k^1), TSF_k \in U_k^2(x_k^2)} J = \sum_{k=1}^{K-1} \{f_{HEV,k}(x_k^1, TSR_k), f_{vehicle,k}(x_k^2, TSF_k), k\} \quad (3.18)$$

where

$$\begin{aligned} x_{k+1}^1 &= f(x_k^1, u_k^1) + x_k^1, \quad k = 0, 1, \dots, N-1 \\ x_{k+1}^2 &= f(x_k^2, u_k^2) + x_k^2, \quad k = 0, 1, \dots, N-1 \end{aligned} \quad (3.19)$$

subject to

$$\begin{aligned} x_k^1 &\in X_k^1 \subset [0.4 \ 0.7], u \in U_k^1 \subset [-1 \ 1] \\ x_k^2 &\in X_k^2 \subset [0.3 \ 0.7], u \in U_k^2 \subset [0 \ 1] \end{aligned} \quad (3.20)$$

3.4.2 The Energy Management Problem

The objective of the energy management controller problem is to minimize the fuel consumption of the vehicle over a predefined drive cycle. A quasi-static discrete-time model is used to define the fuel consumption of the vehicle. Our energy management DP formulation is similar to the one presented in [14]. The SOC is the only dynamic state in the model. And torque split ratio between internal combustion engine and electric motor is the control signal. In [14], the discrete model is firstly defined as

$$x_{k+1}^1 = f(x_k^1, u_k^1, v, a, i) \quad (3.21)$$

where x_k^1 stands for SOC, u_k^1 stands for torque split ratio between internal combustion engine and electric motor, v stands for vehicle velocity, a stands for vehicle acceleration and i stands for gear number.

For our DP analysis the discrete model in equation (3.21) is simplified:

$$x_{k+1}^1 = f(x_k^1, u_k^1), \quad k = 0, 1, \dots, N-1 \quad (3.22)$$

where

$$x_k^1 \in S_k^1 \wedge u_k^1 \in C_k^1 \quad (3.23)$$

with

$$S_k^1 = [0.4 \ 0.7] \wedge C_k^1 = [-1 \ 1] \quad (3.24)$$

N denotes the number of calculation steps in the DP procedure based on the length of the drive cycle defined for DP procedure. S_k^1 and C_k^1 are defined as the state space and input space for the dynamic programming algorithm respectively in equation (3.24). S_k^1 limits the dynamic state of the model, and C_k^1 limits the control signal of the controller. Here it is assumed that driving cycle is known in advance. In our study FTP75 drive cycle is used for all simulations in order to have a fixed basis when comparing different controller schemes. Fig.3.3 shows the velocity profile of the FTP75 drive cycle.

The optimization problem for energy management controller is formulated as below.

The energy management DP problem

Given an objective J , solve

$$\min_{TSR_k \in U_k^1(x_k^1)} J = \sum_{k=1}^{K-1} f_{HEV,k}(x_k^1, TSR_k) \quad (3.25)$$

where

$$x_{k+1}^1 = f_{HEV}(x_k^1, TSR_k) \quad (3.26)$$

subject to

$$x_k^1 \in X_k^1 \subset [0.4 \ 0.7], TSR_k \in U_k^1 \subset [-1 \ 1]. \quad (3.27)$$

In equation (3.25), $f_{HEV}(x_k^1, TSR_k)$ is the fuel consumption function of the HEV model as a cost of the system. $f_{HEV}(x_k^1, TSR_k)$ calculates the fuel consumption of the vehicle as

$$f_{HEV}(x_k^1, TSR_k) = \Delta m_f(TSR_k, k) \cdot T_s. \quad (3.28)$$

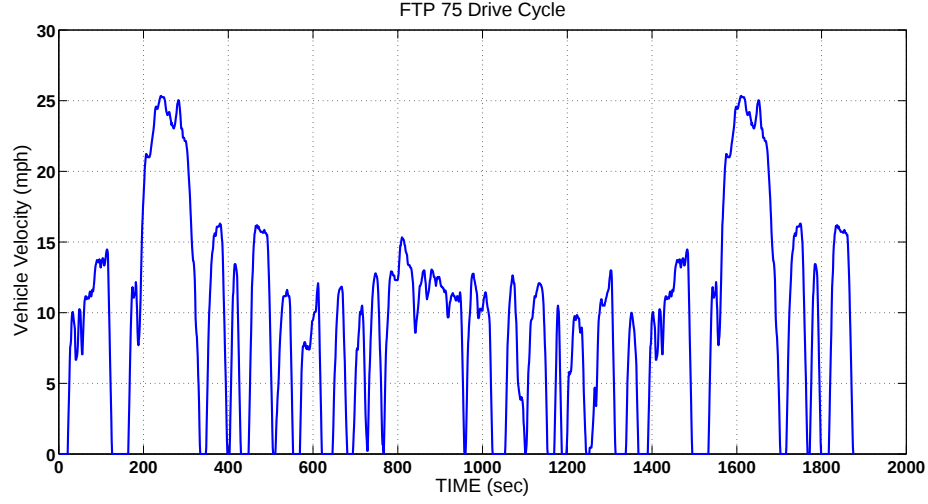


Figure 3.3: FTP 75 Drive Cycle.

Table 3.1: Working Modes of Powertrain.

TSR RANGE	WORKING MODE
$TSR = 0$	Electric Motor Only Mode
$0 < TSR < 1$	Torque Assist Mode
$TSR = 1$	Engine Only Mode
$1 < TSR$	Battery Charging Mode

Dynamic state, x_k^1 , is SOC and control signal, TSR_k , is torque split ratio between internal combustion engine and electric motor. The objective of the DP algorithm is to minimize the cost function.

TSR is defined as

$$TSR = \frac{\text{Engine Torque Calculated at the Wheels}}{\text{Total Torque Calculated at the Wheels}}. \quad (3.29)$$

The working modes of the powertrain are given in Table 3.1

Initial SOC is taken as 0.5 and final SOC is between 0.5 and 0.51. For our DP analysis the dpm function outlined in [14] is used. We obtain the optimal torque split ratio trace by taking the argument which minimizes the cost function given

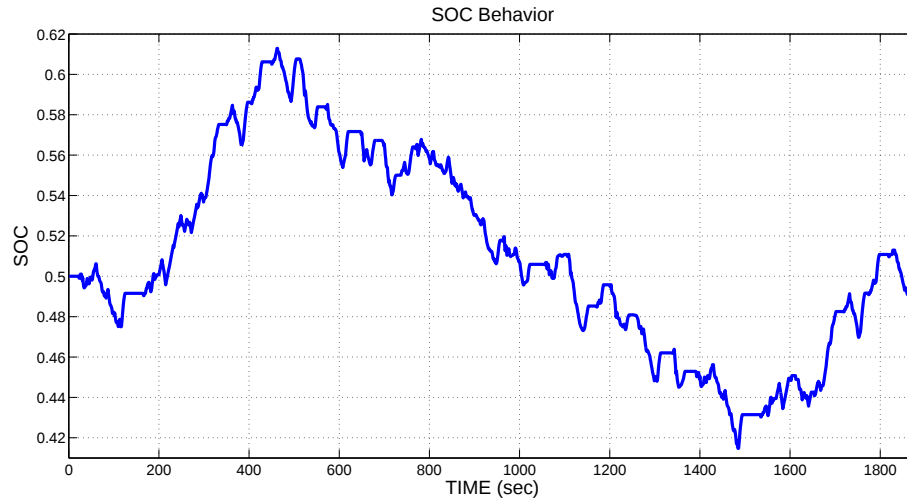


Figure 3.4: SOC Behavior of Energy Management Controller.

in equation (3.25). In Fig.3.4, SOC behavior is given and in Fig.3.5, the optimal operating trace of the energy management controller is given.

It can be seen in the results in Fig.3.4 and Fig.3.5 that the vehicle is working in the electric motor only mode in the low torque demand range when vehicle is launched. Optimal trace of the controller shows that in the low torque demand range except vehicle launch, recharging mode is preferred. Engine only mode is dominant in the middle torque demand range, and torque assist mode is preferred in the high torque demand mode. In Fig.3.5, optimum trace shows that our hybrid electric model works like a typical parallel hybrid electric vehicle [14].

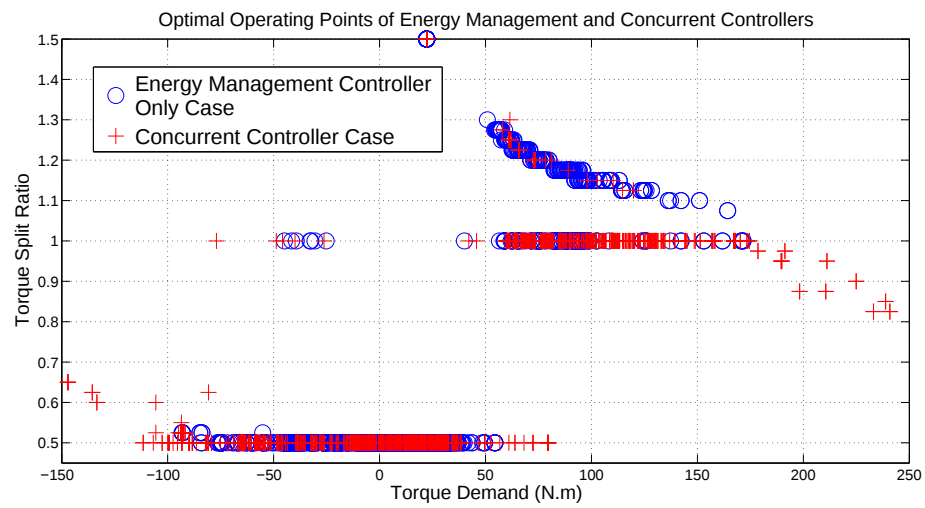


Figure 3.5: Optimal Operating Points of EM vs Concurrent Controllers.

3.4.3 The Vehicle Stability Problem

The objective of the vehicle stability controller problem is to minimize the wheel slip of the vehicle over a predefined drive cycle. A quasi-static discrete-time model is used to define the wheel slip of the vehicle.

The algorithm outlined in [14] is also used for DP analysis of vehicle dynamics control system individually. Here the vehicle is assumed to be non-hybrid so the battery and the electric motor are removed from the system. The only dynamic state is the rear rotational speed ratio (RRSR), $\omega_{ratio,rear}$. The torque split factor between front and rear axles is the control signal.

The discrete model is firstly defined as

$$x_{k+1}^2 = f(x_k^2, u_k^2, v, a, i, \mu) \quad (3.30)$$

where x_k^2 stands for RRSR, u_k^2 stands for torque split factor between front and rear axles, v stands for vehicle velocity, a stands for vehicle acceleration, i stands for gear number and μ stands for the friction coefficient between tire and road.

In 3.2-3.4 N denotes the length of the drive cycle. L is defined as the cost function of a single subsequent stage whereas J represents the total cost of the system. Constraints on system states, X_k , and control actions, U_k , can also be defined as shown in equation (3.4). The constraints of the system are stated as shown in equation (3.5). For our DP analysis the discrete model in equation (3.30) is simplified as given below.

$$x_{k+1}^2 = f(x_k^2, u_k^2), \quad k = 0, 1, \dots, N - 1 \quad (3.31)$$

where

$$x_k^2 \in S_k^2 \wedge u_k^2 \in C_k^2 \quad (3.32)$$

with

$$S_k^2 = [0.3 \ 0.7] \wedge C_k^2 = [0 \ 1] \quad (3.33)$$

N denotes the number of calculation steps in the DP procedure based on the length of the drive cycle defined for DP procedure. S_k^2 and C_k^2 are defined as the

state space and input space for the dynamic programming algorithm respectively in equation (3.33). S_k^2 limits the dynamic state of the model and C_k^2 limits the control signal of the controller.

When simplifying the discrete model we have to know the friction coefficient between road and tire as well as vehicle speed, vehicle acceleration and gear number. For traction controller studies, the most common way is to make simulations for short distances. In this study we need to use long drive cycles in order to provide the coherence between the two control problems. It is assumed that friction coefficient is given for the drive cycle. This is specified based on the limitation of the vehicle acceleration given in the equation (2.6).

The vehicle stability DP problem

Given an objective J , solve

$$\min_{TSF_k \in U_k^2(x_k^2)} J = \sum_{k=1}^{K-1} f_{vehicle,k}(x_k^2, TSF_k) \quad (3.34)$$

where

$$x_{k+1}^2 = f_{vehicle}(x_k^2, TSF_k) \quad (3.35)$$

subject to

$$x_k^2 \in X_k^2 \subset [0.3 \ 0.7], TSF_k \in U_k^2 \subset [0 \ 1] \quad (3.36)$$

In equation (3.34) $f_{vehicle}(x_k^2, TSF_k)$, is the wheel slip function of the vehicle model as a cost of the system. $f_{vehicle}(x_k^2, TSF_k)$ calculates the wheel slip of the vehicle as

$$f_{vehicle}(x_k^2, TSF_k) = \Delta wheel\ slip_f(TSF_k, k) \cdot T_s. \quad (3.37)$$

Dynamic state, x_k^2 , is RRSR and control signal, TSF_k , is torque split factor between front and rear axles. The aim of the DP algorithm is minimizing the wheel slip while maximizing the tractive force. TSF is defined as

$$TSF = \frac{Front\ Axle\ Torque\ Calculated\ at\ Wheels}{Total\ Torque\ Calculated\ at\ Wheels}. \quad (3.38)$$

The working modes of the powertrain are given in Table 3.2.

Table 3.2: Working Modes of Powertrain.

TSF RANGE	WORKING MODE
$TSF = 0$	<i>Rear Axle Only Mode</i>
$0 < TSF < 1$	<i>Front and Rear Mixing Mode</i>
$TSF = 1$	<i>Front Axle Only Mode</i>

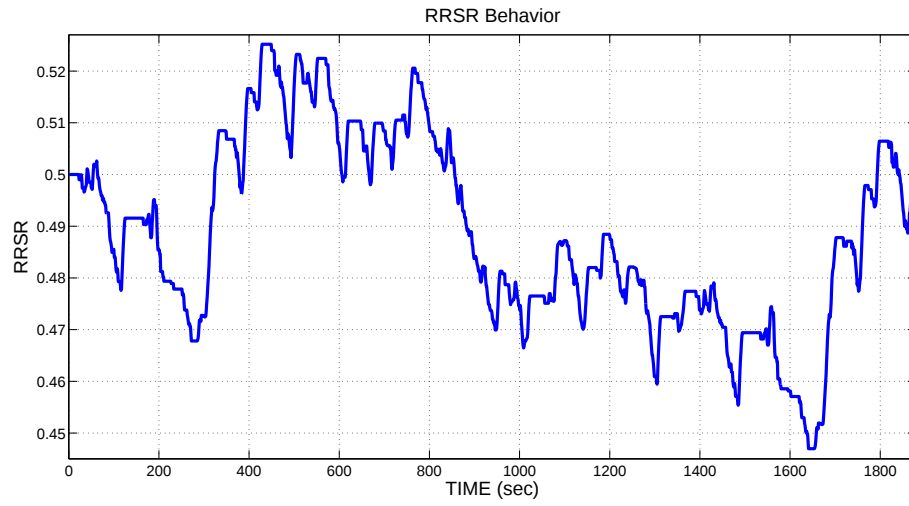


Figure 3.6: RRSR Behavior of Vehicle Stability Controller.

Initial RRSR is chosen as 0.5. The final RRSR value is between 0.5 and 0.51. RRSR behavior is given in Fig.3.6. Fig.3.7 shows the optimal operating trace of the controller. Optimal trace of the controller shows that in the low speed range rear axle only mode is preferred. Front and rear axle mixing mode is dominant in the middle and high speed range. There are transitions between front and rear axle when the crankshaft speed is about 250 rad/s. Fig.3.7 (plus signs) shows the optimal traces of the concurrent controller. Torque assist mode got more dominant in the high torque demand range. Transitions between front and rear axles took place between 200 rad/s and 350 rad/s range.

It should be noted that we need a complex transfer case model in order to provide front/rear axle only modes. The main objective of applying DP is to obtain the optimal traces. Working mode of powertrain should also be considered when

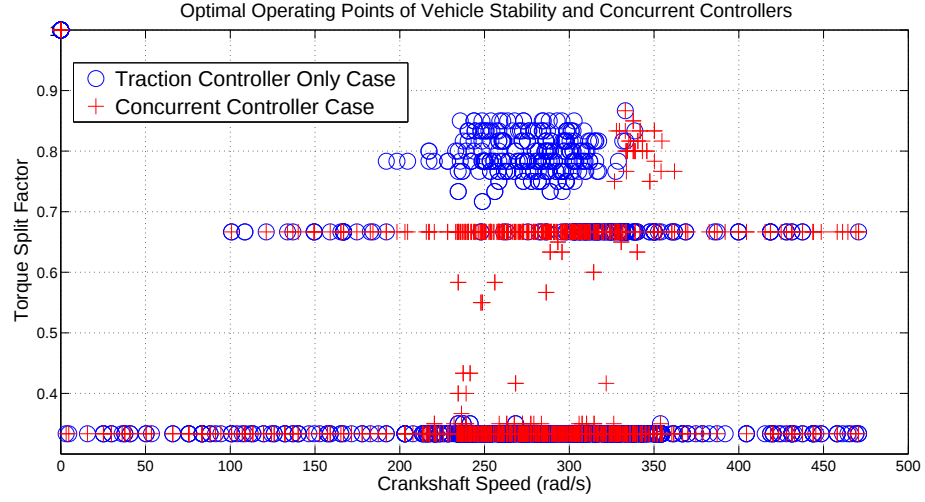


Figure 3.7: Optimal Operating Points of VSC vs Concurrent Controllers.

Table 3.3: Fuel Consumption and Wheel Slip Comparison over FTP75 Cycle.

	Fuel Consumption ($l/100km$)	Average Wheel Slip (%)	Improvement	
			Fuel Consumption	Average Wheel Slip
EM Only Case	8.3			
VSC Only Case		3.6		
Concurrent Controller	7.5	3.4	9.63%	5.5%

analyzing the results. In the torque assist mode the decrease in wheel slip provides decrease in energy loss of the vehicle.

It should also be noted that the concurrent problem formulation reduces to the energy management problem formulation when the state variable of vehicle stability controller is kept fixed or vice versa. The interaction between the state variables of energy management and vehicle stability controllers provide us better results in concurrent controller optimization. Optimization characteristics of the hybrid vehicle problem are enhanced by using two level optimization algorithms. However these optimization levels are not separated as shown in our problem formulations.

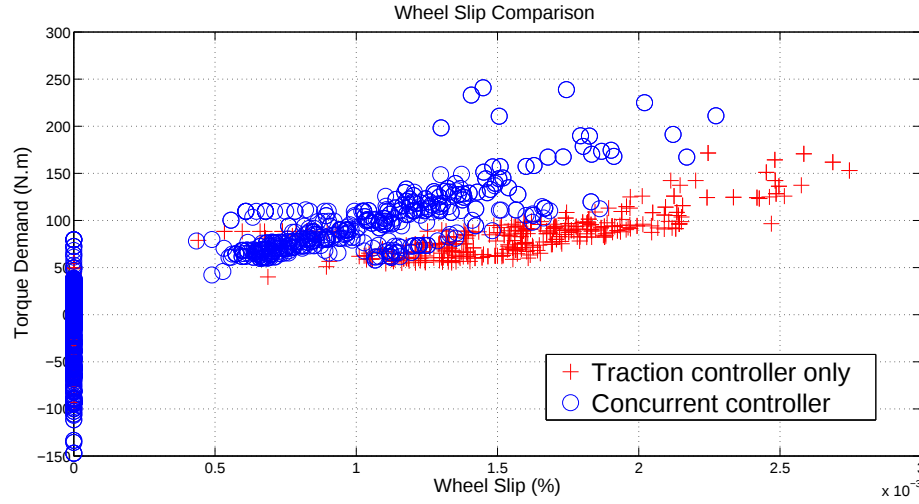


Figure 3.8: Wheel Slip Comparison of Vehicle Stability and Concurrent Controllers.

In Fig.3.8 and Fig.3.9, the fuel rate and wheel slip comparisons are illustrated between individual cases and concurrent. Profiles obtained show that fuel rate is decreased when concurrent controller is used since energy loss due to the slipage is eliminated by hybrid energy management strategies such as regenerative braking. The profile of concurrent controller stays around an optimum fuel rate line with low fluctuations. This tells us that power consumption of electric motor gets higher by being dominant in the torque assist mode of the powertrain and helping the internal combustion engine to operate in the fuel efficient range. The fuel rate and wheel slip profiles are integrated over FTP75 drive cycle given in Table 3.3. The results indicate that we can obtain high levels of fuel efficiency in the long range driving conditions. Hard acceleration and braking ranges are outlined where the difference is significant. It is also observed that less wheel slip is obtained for the same torque demand when concurrent controller is used since torque adjustment of wheel slip controller is stabilized by the energy management strategy. As electric motor assistance is improved the contribution of electric motor gets higher. This makes the torque transitions of the powertrain stable.

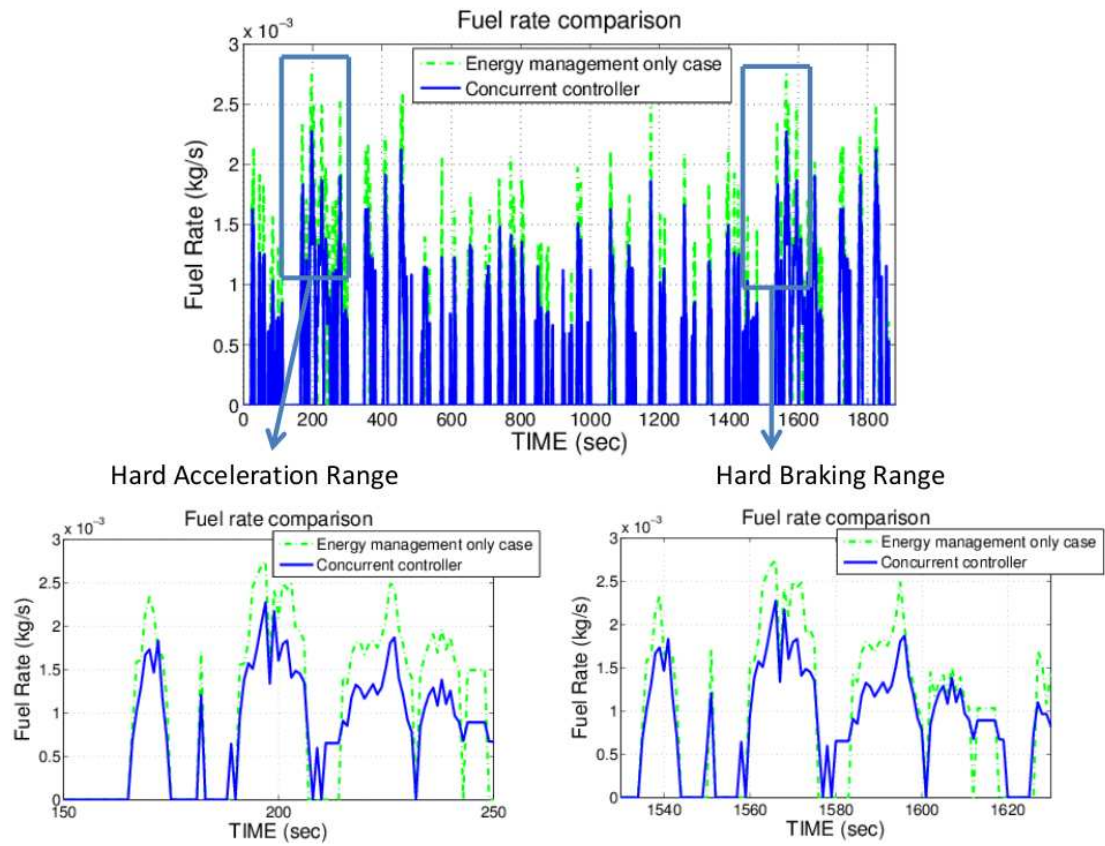


Figure 3.9: Fuel rate comparison of concurrent and EM controllers.

Chapter 4

Controller Development

When optimal control actions are obtained in the DP process the next step is to design controllers which will command actions similar to the optimal actions in similar cases. These designed energy management and vehicle stability control algorithms will be added to our complex nonlinear vehicle model discussed in Chapter 2 so that the algorithms can be tested and the control design process is completed.

Optimal control actions are used as reference set-points when constructing the control algorithms. The relationships between SOC and torque demand, wheel slip and torque demand are used in individual cases whereas the relationship between SOC, wheel slip and torque demand as shown in Fig.4.1 is used in the concurrent case. These relationships give us desired values of SOC and wheel slip for various operating points.

Based on these relationships look-up tables are constructed in order to be used in the control algorithms. Look-up tables provide the desired values of SOC and wheel slip in the controller architecture based on the cases mentioned. In individual cases, one dimensional look-up tables are used such that the actual torque demand determines the desired values of SOC and wheel slip values. However in the concurrent case two look-up tables which are two dimensional are used such that the actual torque demand and SOC values determine the desired wheel slip

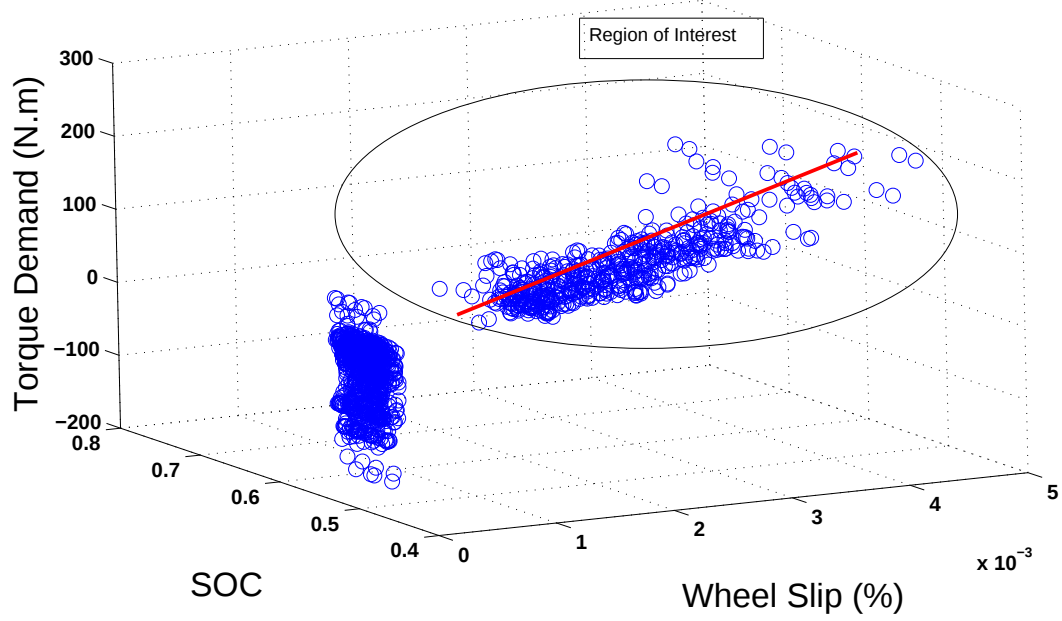


Figure 4.1: Optimal Traces provided by DP process.

of the vehicle and the actual torque demand and wheel slip values determine the desired SOC of the vehicle.

4.1 The Vehicle Stability Controller

A simple feedback controller is designed for vehicle stability. The objective of the controller is to minimize the wheel slip. The controller consists of two model blocks which are embedded in to the controller block and the wheel model blocks discussed in Chapter 2. In the wheel model a quarter car model is used to calculate the actual wheel slip value based on the controller command sent to the wheel models as shown in Fig. 4.2. Empirical tire model is used in the model. The control signal and the vehicle speed are the inputs whereas the wheel speed and the wheel slip are outputs of the model. PI controller is used in the controller block as shown in Fig.4.3. The control signal is the difference between the actual and the desired wheel slip values. Desired wheel slip value is taken from look-up

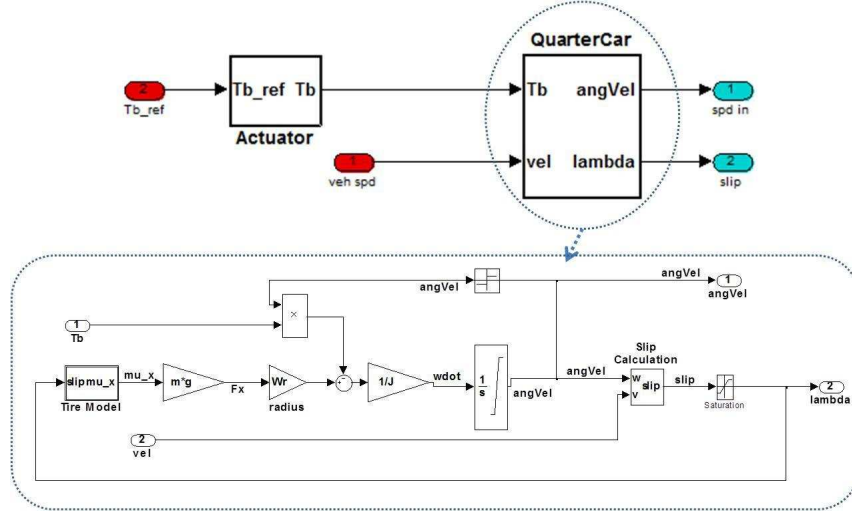


Figure 4.2: Wheel Slip Calculation of the Vehicle Stability Controller.

table constructed having input as torque demand and giving output as desired wheel slip. The relationship between torque demand and optimal wheel slip provided by DP process is shown in Fig. 4.4. It is observed that in high torque demand region there is more wheel slip. We can say that the vehicle model is achieving a good transfer of traction torque to the wheels in low torque demand region. However in high torque demand region, as our vehicle model includes different torque values on the rear and the front axles, traction torque transfer to the wheels is not so good compared to the low torque demand range.

We can conclude that the weight transfer to the rear and the front axles in acceleration and decelerating situations consequences to more wheel slip of the vehicle model. Our controller tries to reduce this high wheel slip as shown in Fig. 4.5.

The vehicle model is simulated using the Urban Dynamometer Driving Schedule, UDDS, as the input drive cycle. Fig. 4.5 shows the wheel slip behavior of the vehicle. It can be seen that wheel slip difference follows a decreasing trend.

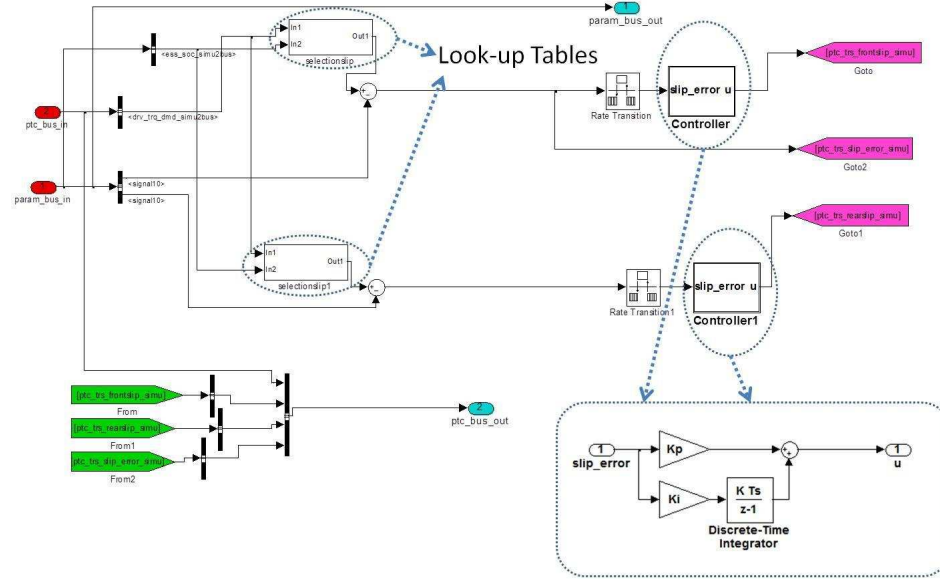


Figure 4.3: Control Signals of Vehicle Stability Controller.



Figure 4.4: Relationship Between Torque Demand and Wheel Slip.

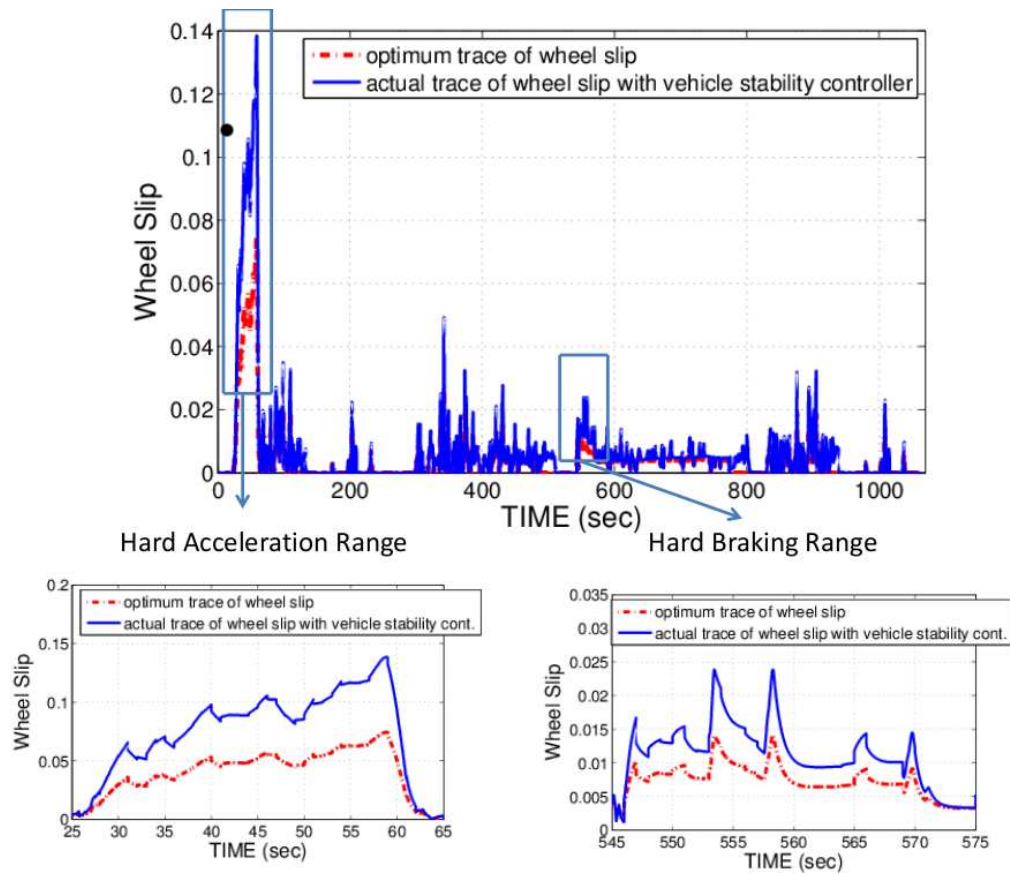


Figure 4.5: Wheel Slip Behavior of the Vehicle with Vehicle Stability Controller.

4.2 The Energy Management Controller

In energy management controller development process our controller is integrated to the state flow chart of the existing realistic control algorithm as an add-on controller function. In Table 4.1 the conditions for the control algorithm are given. The controller decides on the working mode of the powertrain based on these conditions. The signal path of the add-on unit is shown in Fig. 4.6. The control signal is the difference between actual and desired SOC values which is provided by the look-up table embedded in the controller. In Fig. 4.7, the relationship between torque demand and SOC provided by the DP process is given.

The rules about energy management strategy extracted are given in Table 4.1. The controller enforces rules in its operations by sending status control commands. $SOC_{diff,cr}$ represents the critical value of the difference between actual and desired SOC values. In energy management control algorithm this critical value is determined as 0.15 as shown in Table 4.2. The critical value is obtained by tuning the controller in various cases. The tuning procedure is based on simulations made in order to compare the controller behavior. As summarized in Table 4.1, the controller tries to make the battery stay in the optimum SOC range. The system is forced to stay in the discharging mode when SOC difference is higher than a specified threshold. Likewise the system is forced to stay in charging mode when SOC difference is lower than a specified threshold. If the SOC difference is in the allowable range the existing realistic controller algorithm is used. In addition if the SOC difference is higher than a specified threshold, the system is forced to use only mechanical brakes instead of regenerative braking.

System is simulated with respect to UDDS. The SOC behavior of the vehicle as shown in Fig. 4.8 shows that the SOC change is stabilized with the help of the developed controllers. It is also seen that in hard acceleration and braking regions the system cannot successfully trace the optimum SOC values where the controller seems to be increasing the SOC value as much as possible which leads the system deviating from the optimum SOC range. In this study it is shown that communicating with the vehicle stability controller providing information

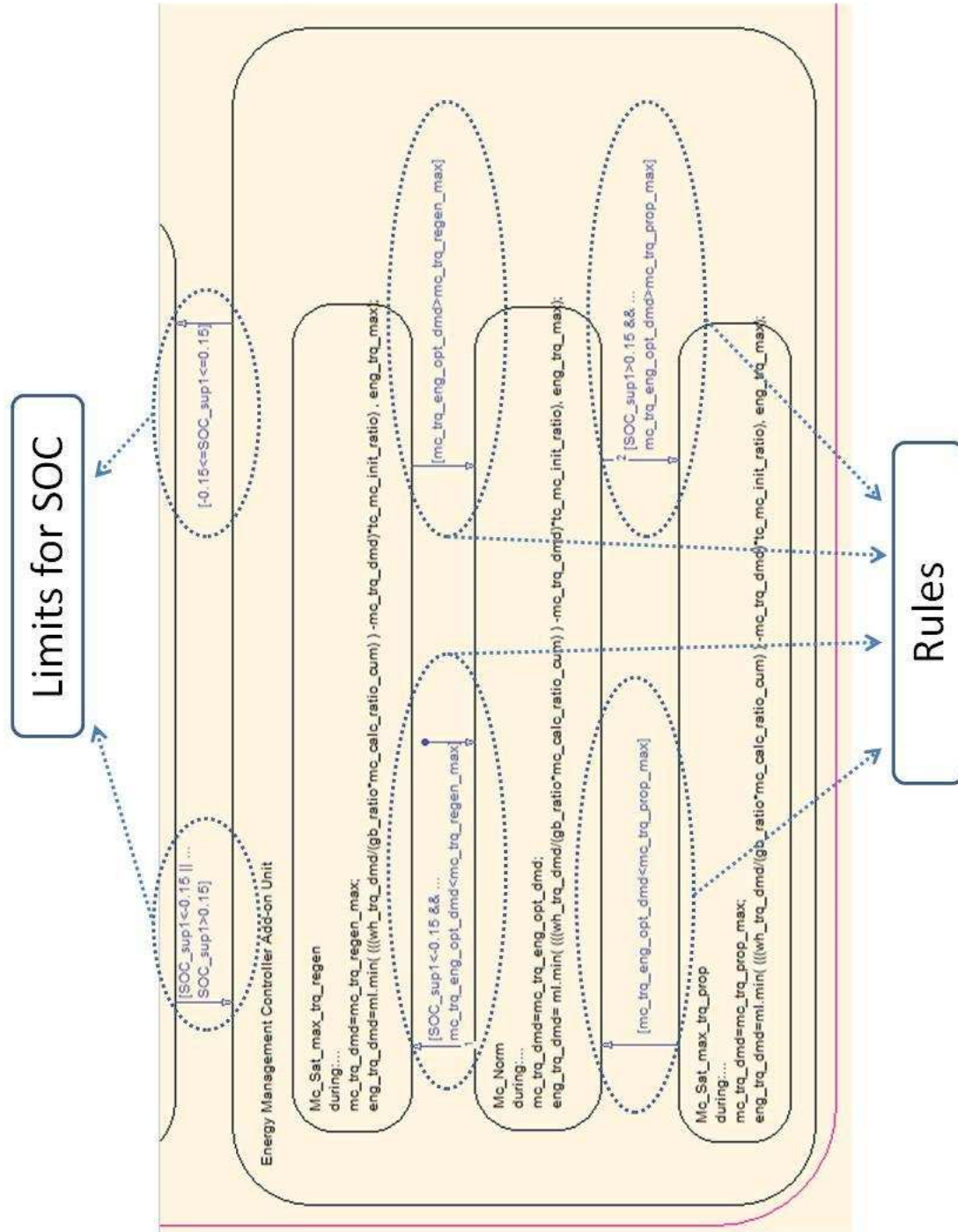


Figure 4.6: Energy Management Controller Add-on Unit.

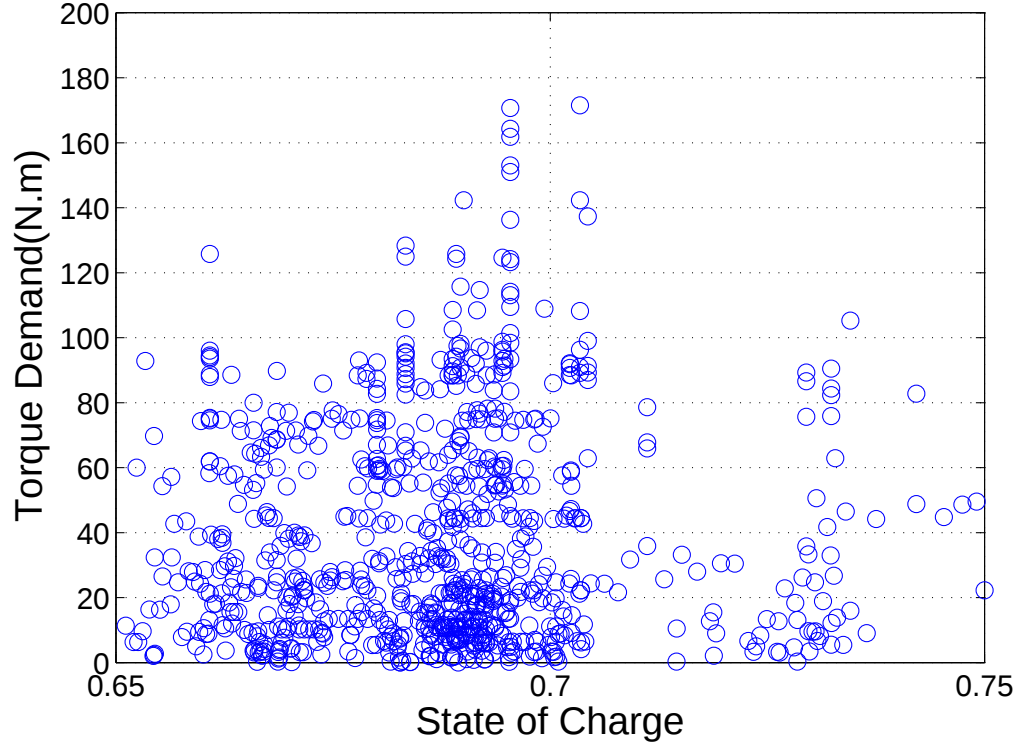


Figure 4.7: Relationship Between Torque Demand and SOC.

Table 4.1: Energy Management Controller Rules.

CONDITION	OPERATING MODE
$if SOC_{diff} < SOC_{diff,cr}$	<i>Normal Mode</i>
$if SOC_{diff} < -SOC_{diff,cr}$	<i>Charging Mode</i>
$if SOC_{diff} > SOC_{diff,cr}$	<i>Discharging Mode</i>
$if SOC_{diff} > SOC_{diff,cr}$	<i>Mechanical Braking</i>

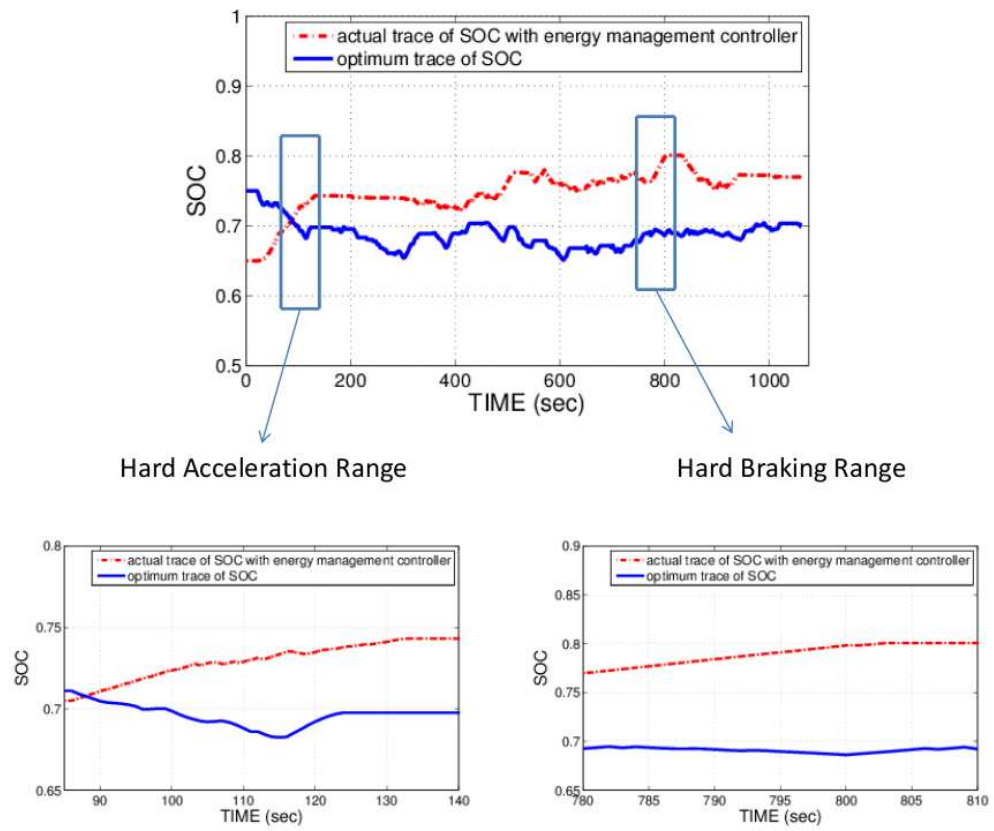


Figure 4.8: SOC Behavior of the Vehicle with Energy Management Controller.

Table 4.2: Critical Values of SOC and Wheel Slip.

$wheel\ slip_{cr}$	0.03
$SOC_{diff,cr}$	0.15
$SOC_{diff,cr1}$	0.15
$SOC_{diff,cr2}$	0.13

about the road conditions, helps the energy management controller to supervise the powertrain systems stay in the optimum SOC range. It should be noted that supervising actions eliminate the deviations in SOC behavior.

4.3 The Concurrent Controller

In concurrent controller development the two developed controllers are coupled by means of new rules and control signal network. It should be noted that the desired values of SOC and wheel slip are taken from the 2-D look-up table constructed having input as torque demand and (i) SOC when determining the desired wheel slip, and (ii) wheel slip when determining the desired SOC. These new rules are integrated to the model via a subsystem added to the existing realistic control algorithm again as such an add-on controller unit. Vehicle stability controller remains the same whereas the signal it takes differs from the individual case.

New rules extracted for concurrent controller are given in Figs. 4.9-4.10. The sequence of these are the same whereas the critical values of the SOC difference, $SOC_{diff,cr1}$ and $SOC_{diff,cr2}$, change with respect to the condition of wheel slip, whether it is higher than the critical wheel slip value, $wheel\ slip_{cr}$, or not. The critical values of the SOC and the wheel slip are obtained by tuning the controller for the optimum SOC and wheel slip behavior of the concurrent controller. Here by defining a critical value, we claim that the developed rules using this critical values work in the most efficient way compared to any other value. Table 4.2 shows the critical values.

The system is simulated with respect to UDDS, and Indian Highway. The SOC

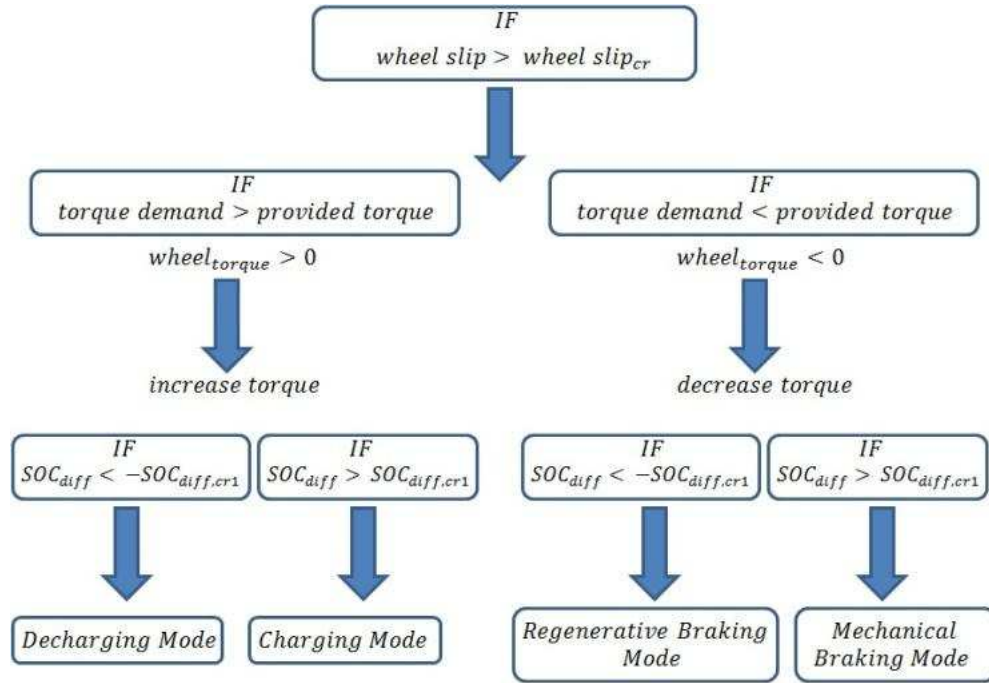


Figure 4.9: Extracted rules for concurrent controller when wheel slip is high.

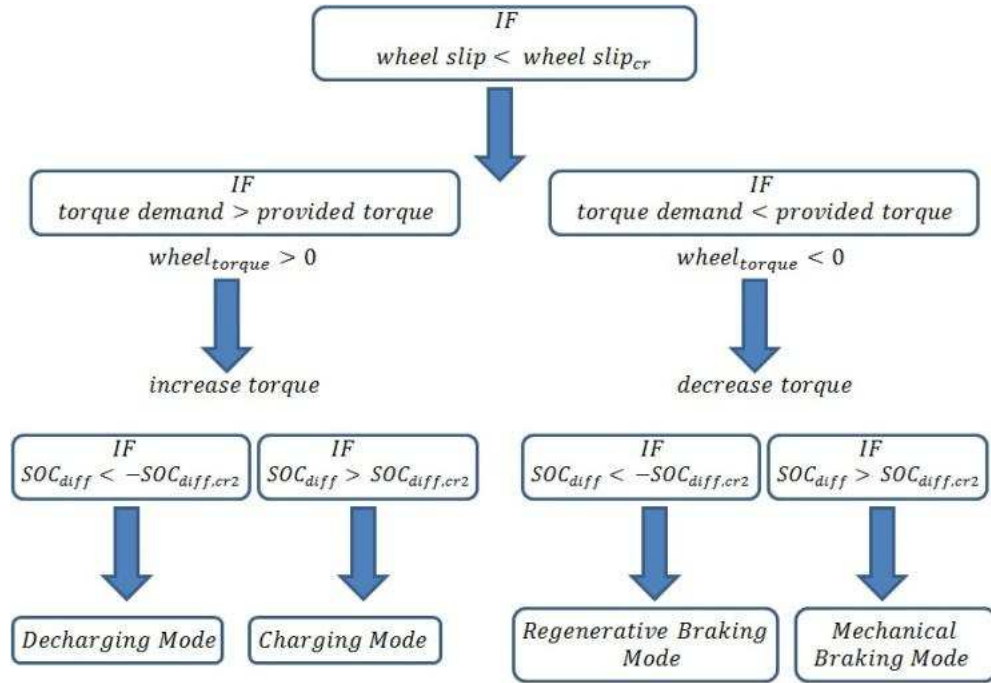


Figure 4.10: Extracted rules for concurrent controller when wheel slip is low.

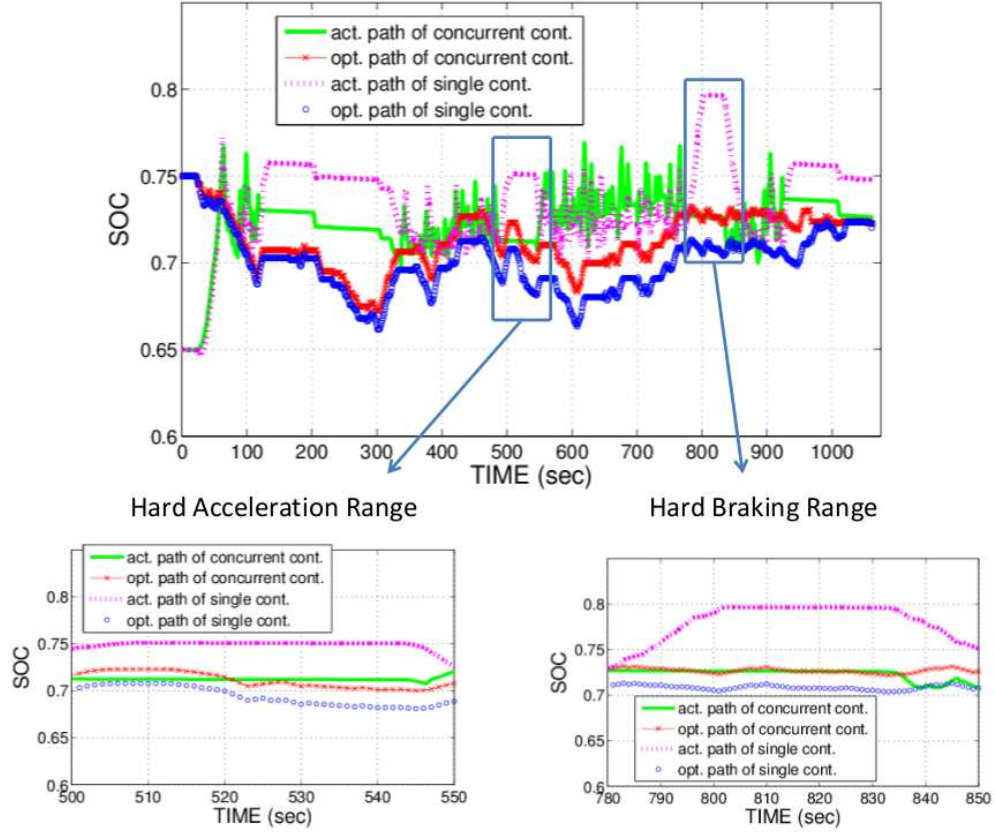


Figure 4.11: SOC behavior of the concurrent controller.

behavior of the vehicle as shown in Fig. 4.11 shows that the actual SOC value is stabilized and merged to desired SOC value with the help of the interaction between the two control algorithms. Wheel slip difference is controlled by the torque adjustments. If energy management controller knows about torque adjustment of the vehicle stability controller before its action, it will take the advantage of giving more efficient decisions. For instance if the vehicle is under the high wheel slip range and vehicle should decrease the torque provided, torque demand will be less and the vehicle will be able to operate in electric motor only mode in order to make the internal combustion engine to stay in efficient range. Regenerative braking can also be applied in such a situation which will provide efficiency to the system.

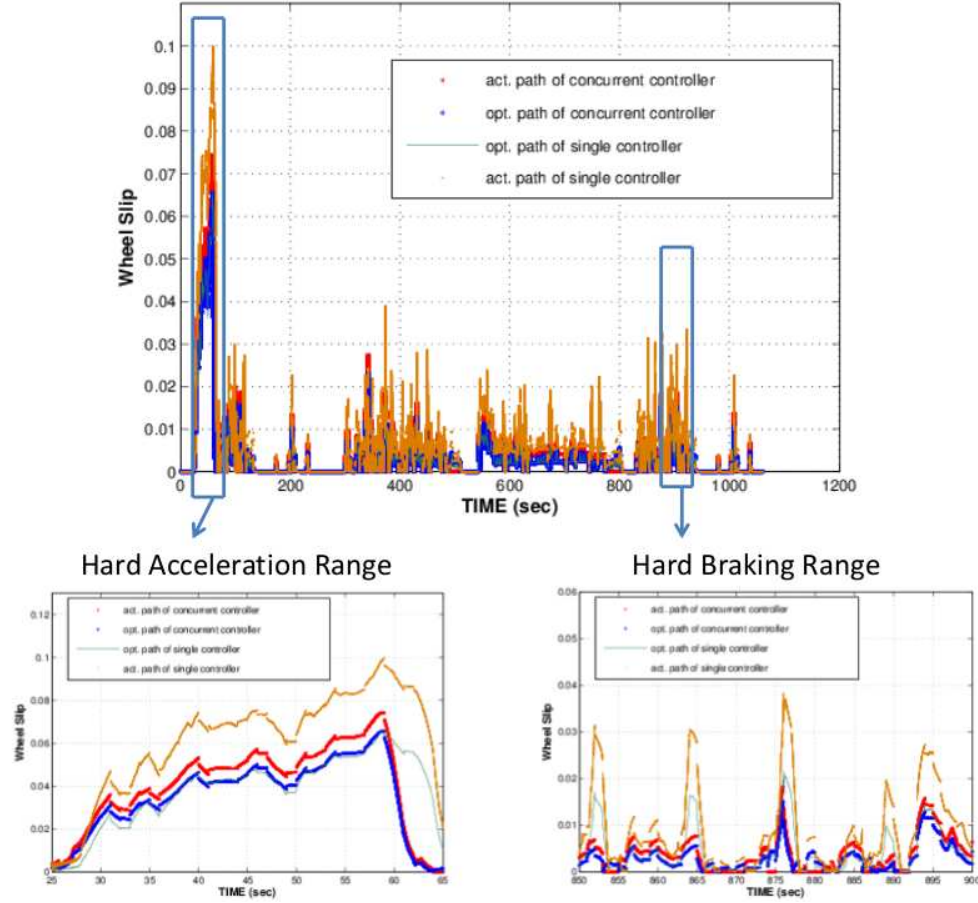


Figure 4.12: Wheel Slip Behavior of the Concurrent Controller.

On the other hand if the vehicle stability controller knows the energy management strategy before its action, this will provide giving more efficient decisions. If the vehicle is operating in the internal combustion engine only mode and vehicle stability controller should decrease the torque, a relevant signal should be sent to the energy management controller to select the electric motor only mode.

Fig. 4.12 shows the wheel slip behavior of the developed controllers. The wheel slip values of the concurrent controller noticeably get closer to the optimum wheel slip range. Interaction between the two controllers makes the concurrent controller work better compared to the single controller. The reason is that torque adjustment actions of the concurrent controller optimized with the help of the

SOC information of the system provided by the energy management controller.

Our concurrent controller provides such an interaction in which the two controllers try to make each other to work in the most efficient range. SOC and wheel slip behavior of the concurrent controller shows that the interaction between the controllers provides flexibility in decision making process. To make the SOC and wheel slip values stay in optimum range also effects the fuel consumption of the vehicle. The fuel rate in different cases for two drive cycles are shown in Figs. 4.13-4.14 for UDDS and Indian Highway drive cycles respectively. It is seen that significant difference of fuel rate between single and concurrent controllers exists in the hard acceleration and braking ranges. This is due to the fact that the SOC and wheel slip behaviors of the concurrent controller are enhanced by the interaction between the two control algorithms.

Nearly optimal behavior in SOC and wheel slip provides better fuel economy as shown in Figs. 4.13-4.14. The reason we are using two different kinds of drive cycle is that we aim to show that the controller algorithm developed in this study is able to compete with all driving conditions. The concurrent controller provides significant improvements especially in hard acceleration and braking ranges as highlighted in Figs. 4.13-4.14. Since the differences between optimal and actual values of SOC and wheel slip increase in such a situation, the concurrent controller takes important role in keeping the vehicle stay in optimal ranges. In Table 4.3 the integrated values over UDDS and Indian Highway driving cycles are given. The two driving conditions are combined when calculating the fuel consumption of the vehicle. We try to reach general fuel economy behavior in order to show more general scope of our controller algorithm. It should be noted that the improvements shown in this study is valid in ideal conditions. Our aim in this study is to show the opportunities in taking into account the interactions between controllers in automotive control systems.

Table 4.3: Fuel Consumption and Wheel Slip Comparison over UDDS and Indian Highway.

	Fuel Consumption ($l/100km$)	Average Wheel Slip (%)	Improvement	
			Fuel Consumption	Average Wheel Slip
EM Only Case	8.5			
VSC Only Case		3.7		
Concurrent Controller	7.1	3.25	16.47%	12.16%

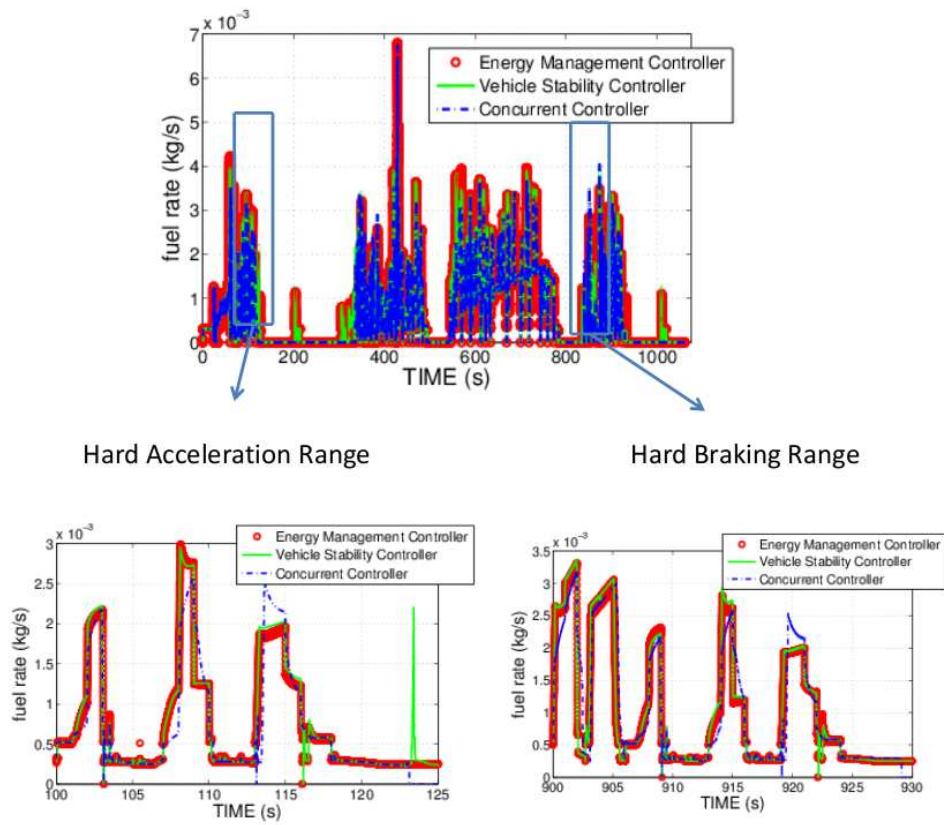


Figure 4.13: Fuel rate behavior of the concurrent controller with UDDS.

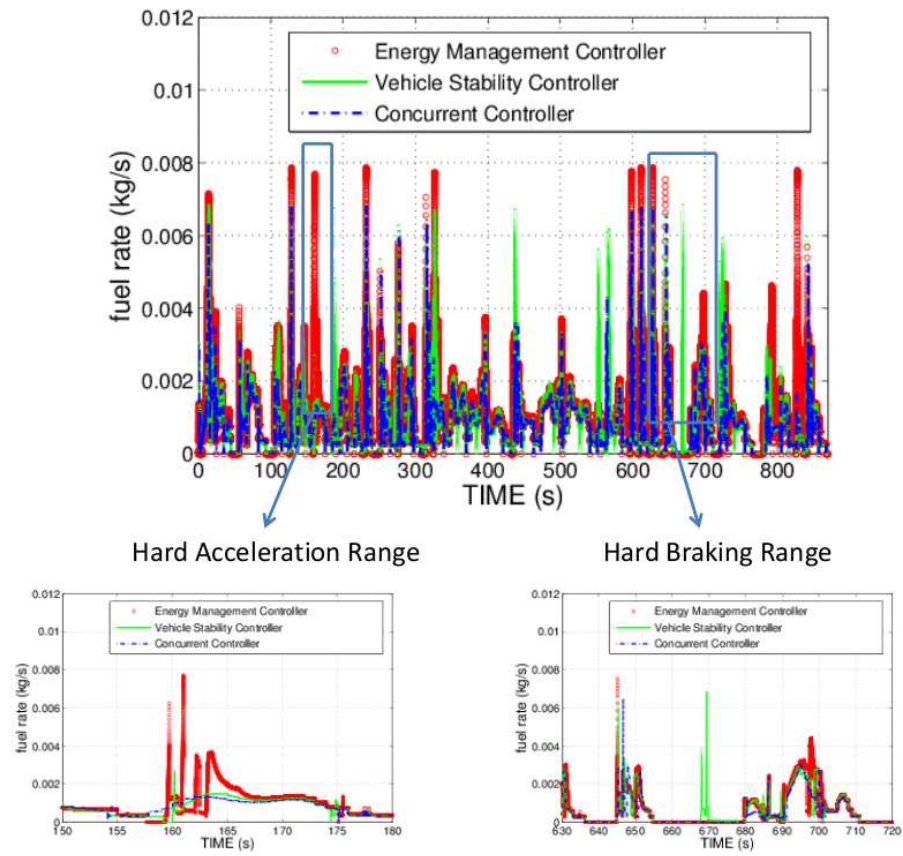


Figure 4.14: Fuel rate behavior of the concurrent controller with Indian Highway.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In our research we studied the possibility of designing a better control algorithm by considering the interaction between the energy management and the vehicle stability controller design problems. The optimal traces obtained by DP constitute the reference point of the developed controller algorithms. Dynamic programming cannot be directly used in developing a controller strategy because it requires a prior known drive cycle. In Fig.5.1, our approach of controller strategy is outlined. We used heuristic control techniques supported with global optimization. We tried to use the advantages of both heuristic control techniques and global optimization. Our control algorithm is based on logic threshold control strategy based on instantaneous optimization mechanisms. The results of DP gave us a chance of comparing the results of our developed controller algorithms.

The performance of the concurrent controller is compared against energy management and vehicle stability controllers individually. Under the same driving conditions the concurrent controller is 16.47% more efficient than energy management controller by means of fuel consumption and 12.16% more efficient than vehicle stability controller by means of wheel slip.

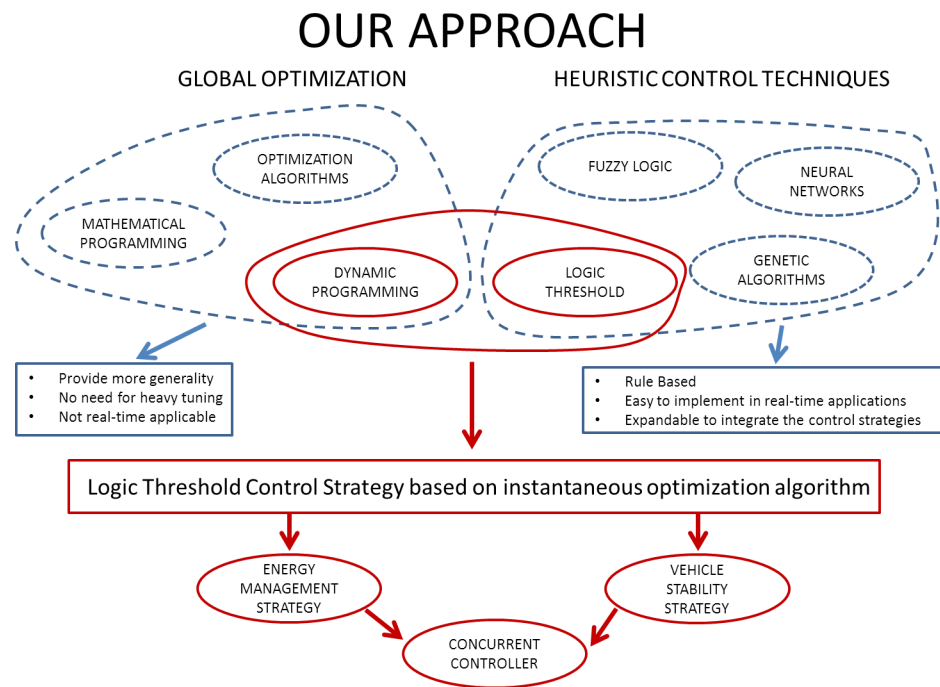


Figure 5.1: Controller Strategy Approach.

In general, energy management controllers using DP are known to be fuel efficient [15]. The main idea pointed out in this study is that the interaction between energy management and vehicle stability controllers can enhance the fuel efficiency of the vehicle. Additionally the vehicle stability controllers proven as reducing the wheel slip values of a vehicle [9] can be enhanced individually by using DP, also they can be more efficiently operated when interacted with energy management controller.

As a result we conclude that allowing the interaction between the two control algorithms gives us better fuel economy with simulations based on mathematical models. The future work for this research is to develop real time implementation of the controllers developed here. Benefits obtained in this study will be examined considering the frequency characteristics of the two controllers due to the fact that traction controller has faster dynamics compared to energy management controller.

5.2 Future Work

5.2.1 Real Time Application Aspects

In this section real time implementation aspects of our research is presented. The optimal traces of controllers obtained in DP are embedded in the controller algorithm via look-up tables. In the operation process of the powertrain, the system is fed by the calculated values of our control signals, namely SOC and wheel slip. These actual values are compared with the optimal traces. The difference between these actual and optimum values drives our controller system. Torque demand of the system is also used in look-up tables as a parameter. In our proposed concurrent controller algorithm optimum SOC is determined by actual torque demand and wheel values and optimum wheel slip value is determined by actual torque demand and SOC value. The first stage of concurrency is provided in the selection of optimum values.

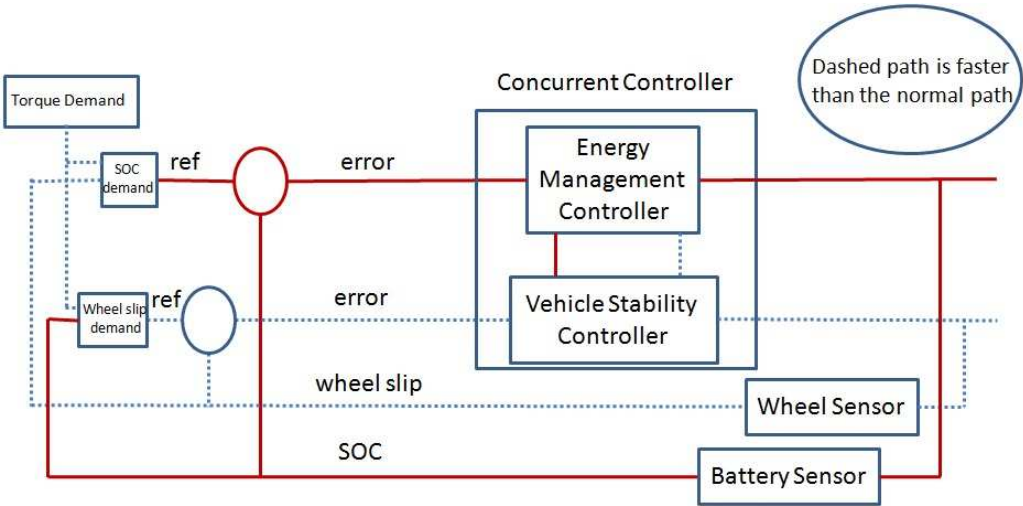


Figure 5.2: Signal Process of Concurrent Controller.

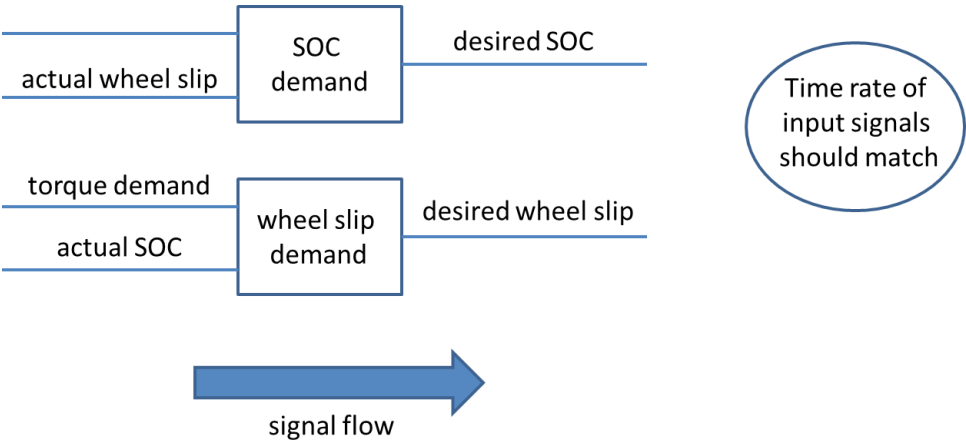


Figure 5.3: Signal Processing of the Concurrent Controller.

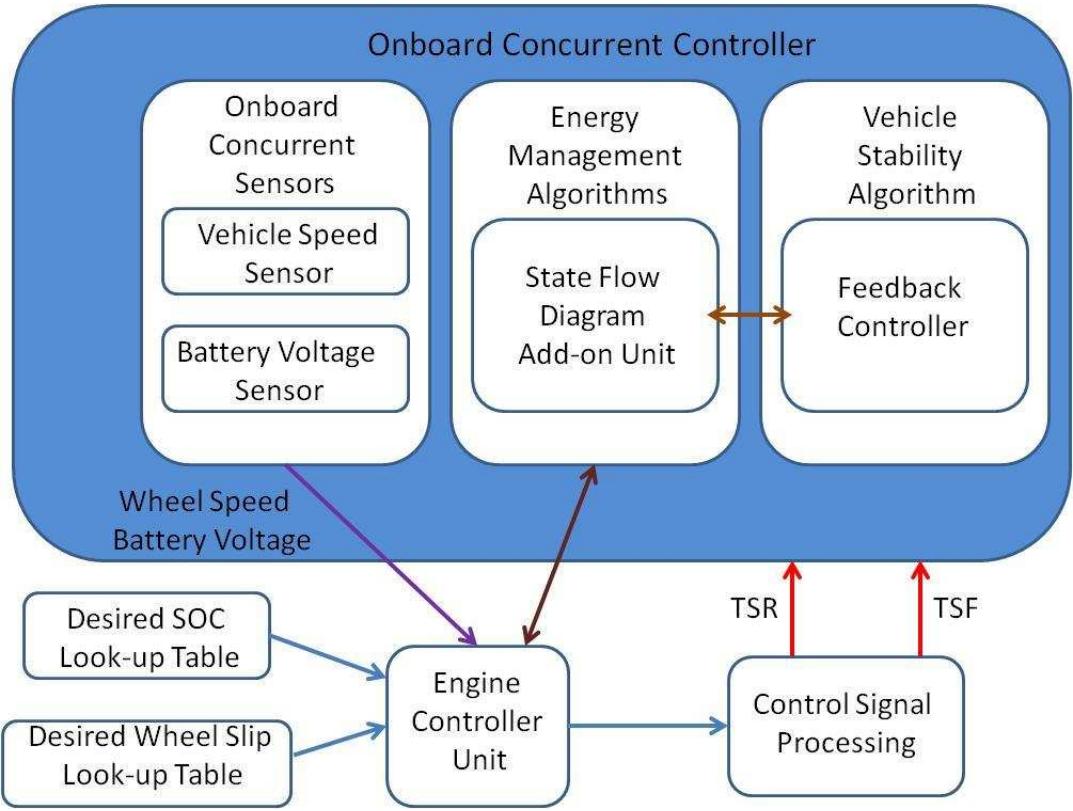


Figure 5.4: Onboard Control Scheme of the Concurrent Controller.

Fig. 5.2 presents a block diagram which describes how our system can be implemented on a real vehicle controller. The error signals of SOC and wheel slip are used in the concurrent controller system at the same time. These error signals are heavily used in our extracted rules in order to compare the actual data with specified thresholds. As we propose that energy management and vehicle stability working concurrently we should mention the time characteristics of these two controllers. Vehicle stability controller dynamics is faster than energy management controller dynamics. As the time characteristics of the two controllers do not match, real time simulations should be done in order to validate the applicability of our proposed concurrent controller algorithm which is planned to be the future work of our research. The signal rates should match in the operation of the controller as outlined in Fig. 5.3.

Our concurrent controller is designed as an add-on unit in the state flow diagram of the original controller. This block is to be placed in the engine control unit (ECU) of the vehicle. The vehicle stability controller block is detached in the controller architecture of the vehicle. Signal sharing between energy management and the vehicle stability controller is provided by the CAN units. The look-up tables providing the desired SOC and the wheel slip values work in the ECU plant. Onboard controller sensors feed the system with the actual data. Actual SOC and the actual wheel slip values are calculated instantaneously by using the sensor outputs battery voltage and the wheel speed respectively. ECU gives the control signals of TSR and TSF as outputs. The scheme of the concurrent controller system integration into the vehicle controller area is shown in Fig. 5.4.

5.2.2 Improvement of The Problem Formulation

By obtaining the optimal trajectories in DP process we continued our research into heuristic control techniques such as rule extracting. Simulations based on predefined driving cycles are performed and results are tried to be generalized with the consideration of urban and highway driving characteristics together.

On the other hand stochastic optimal control policies can also be applied to our

hybrid vehicle model in order to improve the level of optimality of the control problem. The research presented in [10] offers stochastic methods to develop the controllers for hybrid vehicles. The developed controller is optimized on a family of random drive cycles. The optimization problem is defined on an infinite horizon so that time invariant control strategy is developed. The vehicle model has three state variables, power demand, SOC and the wheel speed.

The infinite horizon optimization problem is given as

$$J_{\pi}(x_0) = \lim_{N \rightarrow \infty} E_{\omega} \left[\sum_{k=0}^{N-1} \gamma^k g(x_k, \pi(x_k)) \right], \quad (5.1)$$

where $g(x_k, \pi(x_k))$ is the instantaneous cost function defined for the problem which has a discount factor of γ , $\pi(x_k)$ is the control policy and $J_{\pi}(x_0)$ is the expected cost with the system starting point of the state x_0 . As it is seen in equation (5.1), N is defined as going to the infinity instead of a specified value as in our research. Power demand, P_{dem} , of the vehicle is modeled as discrete-time stochastic dynamic process. The details of stochastic modeling can be found in [15].

Control signal is defined as the engine power P_e . The motor power, P_{mot} , becomes a dependent variable:

$$P_{e,k} + P_{mot,k} = P_{dem,k} \quad (5.2)$$

where k denotes the calculation instance.

The control policy iteration algorithm is used to solve the stochastic dynamic programming problem based on the Bellman's Optimality Principle. The iteration process continues until the convergence point of the optimal cost function. The stochastic method offers a generalized solution independent of time. So that it is implemented directly to the real-time applications and vehicle tests. Since our research focuses on the coupling effects between two automotive control problems working in the same physical plant we studied the deterministic controller development techniques in order to see the interaction of the controllers better. It should be noted that deterministic methods give better results than stochastic methods on a specified drive cycle. As a future work stochastic methods are

planned to be used in the controller development process for our hybrid vehicle model.

Bibliography

- [1] Anonymous. European environment agency, transport and environment: on the way to a new common transport policy. in EEA Report, no:1/2007.
- [2] R. E. Bellman and R. E. Kalaba. *Dynamic Programming and Modern Control Theory*. Academic Press, 1965.
- [3] B. Bhowmik. Dynamic programming its principles, applications, strengths and limitations. *International Journal of Engineering Science and Technology*, 2(9):4822–4826, 2010.
- [4] F. Borelli, A. Bemporad, M. Fodor, and D. Hrovat. An mpc/hybrid system approach to traction control. *IEEE Trans. on Con. Sys. Technol.*, 14(3), May 2006.
- [5] A. P. de Madrid, S. Dormido, and F. Morilla. Reduction of the dimensionality of dynamic programming: a case study. In *Proceedings of the American Control Conference*, volume 4 of *American Control Conference*, San Diego, USA, 1999. ACC.
- [6] S. Delprat, J. Lauber, T. M. Guerra, and J. Rimaux. Control of a parallel hybrid power train: Optimal control. *IEEE Trans. Veh. Technol.*, 53(3):872–881, 2004.
- [7] D. G. Evans, M. E. Polom, S. G. Poulos, K. D. van Maanen, and T. H. Zarger. Powertrain architecture and controls integration for GM’s hybrid full-size pickup truck. *SAE*, 2003. SAE Technical Paper 2003-01-0085 , 2003, doi:10.4271/2003-01-0085.

- [8] Y. Huang, C. Yin, and J. Zhang. Optimal torque distribution control strategy for parallel hybrid electric buses. *WSEAS Transactions on Systems*, 7(6), June 2008.
- [9] D. Kim, S. Hwang, and H. Kim. Vehicle stability enhancement of four-wheel-drive hybrid electric vehicle using rear motor control. *Vehicular Transactions on Vehicular Technology*, 57(2), 2008.
- [10] C. Lin, H. Peng, and J. W. Grizzle. A stochastic control strategy for hybrid electric vehicles. In *Proceedings of 2004, American Control Conference*, volume 5 of *American Control Conference*, pages 4710 – 4715. American Control Conference, 2004.
- [11] J. Ottosson. *Energy Management and Control of Electrical Drives in Hybrid Electrical Vehicles*. PhD thesis, Lund University, 2007.
- [12] D. Porto, A. Martinez, and S. Scimone. A hybrid engine control system based on genetic algorithms. *WSEAS*, pages 417–423, July 2007.
- [13] R. D. Robinett, III, D. G. Wilson, G. R. Eisler, and J. E. Hurtado. *Applied Dynamic Programming for Optimization of Dynamical Systems*. Society for Industrial and Applied Mathematics, 2005.
- [14] O. Sundström and L. Guzzella. A generic dynamic programming Matlab function. In *Proceedings of the 18th IEEE International Conference on Control Applications*, IEEE International Conference, Saint Petersburg, Russia, 2009. IEEE.
- [15] O. Sundström, L. Guzzella, and P. Soltic. Optimal hybridization two parallel hybrid electric vehicles using dynamic programming. In *Proceedings of The 17th IFAC World Congress, 2008*, volume 17 of *IFAC World Congress*, Seoul, Korea, 2008. IFAC.
- [16] K. Tong. *Simultaneous Plant/Controller Optimization of Traction Control for Electric Vehicle*. PhD thesis, University of Waterloo, 2007.

- [17] S. G. Wirasingha, S. M. Lukic, F. Rodriguez, A. I. Antoniou, and A. Emadi. Adaptive control for hybrid electric vehicles. *International Journal of Electric and Hybrid Vehicles*, 3(2):872–881, 2011.

Appendix A

Codes

The Main file running dpm function used in [14] is presented here:

```
% load driving cycle
load JN1015

% create grid
grd.Nx{1}    = 61;
grd.Xn{1}.hi = 0.7;
grd.Xn{1}.lo = 0.4;

grd.Nu{1}    = 21;
grd.Un{1}.hi = 1;
grd.Un{1}.lo = -1; % Att: Lower bound may vary with engine size.

% set initial state
grd.X0{1} = 0.55;

% final state constraints
grd.XN{1}.hi = 0.56;
grd.XN{1}.lo = 0.55;
```

```
% define problem
prb.W{1} = speed_vector; % (661 elements)
prb.W{2} = acceleration_vector; % (661 elements)
prb.W{3} = gearnumber_vector; % (661 elements)
prb.Ts = 1;
prb.N = 660*1/prb.Ts + 1;

% set options
options = dpm();
options.UseLine = 1;
options.SaveMap = 1;
options.MyInf = 1000;
options.Iter = 5;
options.InputType = 'c';
options.FixedGrid = 0;
[res dyn] = dpm(@hev,[],grd,prb,options);
```