Optimally Controlling Hybrid Electric Vehicles using Path Forecasting

by

Georgia-Evangelia Katsargyri

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2008

Author .....
Department of Electrical Engineering and Computer Science
May 23, 2008

Certified by .....
Munther A. Dahleh
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by .....
Terry P. Orlando
Chairman, Department Committee on Graduate Students

Optimally Controlling Hybrid Electric Vehicles using Path Forecasting

by

Georgia-Evangelia Katsargyri

Submitted to the Department of Electrical Engineering and Computer Science
on May 23, 2008, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

Hybrid Electric Vehicles (HEVs) with path-forecasting belong to the class of fuel-efficient vehicles, which use external sensory information and powertrains with multiple operating modes in order to increase fuel economy. Their main characteristic is that the decision to charge and discharge the battery is made in part by using a prediction of future road conditions. The increasing presence of GPS navigational systems in the standard feature sets of the modern vehicles suggests that path predictive methods applied to HEVs constitute one of the most promising directions towards the solution of serious problems of our era, such as the energy problem, the increasing cost of oil, and the greenhouse gas emissions. In the current project we are given a route and an HEV simulation model, and we aim to minimize the fuel consumption of the vehicle along the route. Towards this direction, we adopt a novel way of decomposing the route into a series of route segments connected to each other and linking the origin to the destination. For each route segment, the road grade, the segment length, and the nominal speed are available. Then, the main idea of our method is to prescribe those set-points of the state of charge of the battery for each road segment, that result in the most fuel efficient travel between the origin and the destination.

Thesis Supervisor: Munther A. Dahleh
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

The author would like to acknowledge Ilya Kolmonovsky, John Michelini, Ming Kuang and Anthony Phillips of Ford Motor Company, for their collaboration on this work. This project is supported in part by Research and Innovation Center and Sustainable Mobility Technologies, Ford Motor Company, Dearborn, MI. Acknowledgements also to the author's advisor, Prof Munther Dahleh, for the constant support and attention. Finally, the author would like to acknowledge the substantial help and feedback of Michael Rinehart.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The increasing global need of energy, the decreasing sediments of oil, and the greenhouse gas emissions are three of the most alarming problems of the contemporary world. Combining the internal combustion engine of a conventional vehicle with the battery and electric motor of an electric vehicle, Hybrid Electric Vehicles could be a drastic solution to all the above concerns, given that they are environmentally-friendly vehicles that can be two or three times more fuel-efficient than conventional vehicles. At the same time they offer low emissions, with the power, range, and convenient fueling of conventional vehicles, and they never need to be plugged into an external battery charger. Instead, to recharge the battery different methods are employed, such as converting kinetic into electric energy during regenerative braking or storing any excessive power during efficient engine operation. The inherent flexibility and the adequate performance of HEVs make them convenient for both fleet and personal transportation, which could facilitate massive usage of HEVs towards a drastic solution to the problems of our era.

An important trend in the design of fuel-efficient vehicles is the use of external sensory information and powertrains with multiple operating modes, a trend that promises significant gains in fuel economy while meeting both performance and resource constraints. HEVs with path-forecasting are a special case of this class of systems, whereby the decision to charge and discharge the battery is made in part by using a prediction of future road conditions. The increasing presence of GPS nav-

igational systems in the standard feature sets of the modern vehicles suggests that path predictive methods applied on HEVs constitute one of the most promising directions towards the solution of the energy problem, the increasing cost of oil, and the greenhouse gas emissions.

Much of the recent research in HEV control has focused on the use of road-condition classifiers and model predictive controllers, that estimate the upcoming road conditions in order to decide whether to charge or discharge the battery. Though this route-predicting information can be incorporated into the state-of-the-art HEV control systems, the amount of computational power and memory that would be required for these systems to effectively utilize it, may be impractical. For instance, a classifier system would require a sizable set of classifications to sufficiently represent all of the possibilities of road characteristics over a long horizon, and a predictive controller would require a significant amount of computational speed in order to make decisions within an acceptable time frame. Though on-board computing power is growing, it is still not sufficiently fast enough to address the above challenges.

In this project, we seek to reduce the computational complexity of controlling HEVs by using Bellman's Dynamic Programming (BDP) and make the on-board procedure quicker by realizing most of the control offline [11], [10]. Specifically we are given an HEV simulation model and we consider a route which is decomposed into a series of route segments connected to each other and linking the origin to the destination. We present a novel way of optimally segmenting the route into segments of unequal length in order to increase the efficiency of our algorithm. For each route segment, the road grade, the segment length, and the nominal speed are available. Our main objective is to prescribe, using dynamic programming, those set-points of the state of charge of the battery for each road segment, that result in the most fuel efficient travel between the origin and the destination.

The rest of the current thesis is organized as follows:

- In chapter 2 we discuss a range of existing approaches for optimally controlling hybrid electric vehicles.

- In chapter 3 we give the necessary background on dynamic programming.
- In chapter 4 we present the simulation model we used and we describe our method in detail.
- In chapter 5 we present some experiments with their results.
- Finally, in chapter 6 we summarize the conclusions and we suggest future work.

Chapter 2

A Range of Existing Approaches to Optimally Controlling Hybrid Electric Vehicles

Responding to the signs of the era, the global auto industry is showing an increasing turn towards the development and improvement of Hybrid Electric Vehicles [4]. HEVs are typically powered by two energy sources [5], an energy conversion unit, such as a conventional internal combustion engine (ICE) or a fuel cell and an energy storage device, such as batteries or ultracapacitors [15],[1]. The seamless integration of these two powertrains and their corresponding components is essential in order to preserve the vehicle's performance and at the same time achieve multiple design objectives, such as high fuel economy and low emissions. To achieve these objectives, the hardware configuration and the power control strategy are designed together. The hardware configuration dictates to some extent what control strategies make sense. Although comparing to the conventional vehicles these strategies add complexity to HEVs , they achieve an optimal integrative operation of all the components with undisputable fuel savings. In the following sections several existing HEV hardware configurations and power control strategies are presented.[8]

2.1 Hybrid Electric Vehicle Configurations

The biggest distinction among hybrid configurations is whether the internal combustion engine and the electric drive operate in parallel, series, or a combination of the two. The last one is also called a split or dual configuration.

2.1.1 Series Configuration

In a *series* design [6], the primary ICE engine does not directly drive the wheels, but it is connected to a generator that produces electricity and charges the battery. Then the battery powers the wheels through an electric motor. It is clear that series HEVs have no mechanical connection between the hybrid powertrain unit and the wheels. The path that the power follows can be described as the following sequential energy conversion: chemical (engine) - mechanical (generator) - electrical (battery) - mechanical (electric motor - wheels). Despite some benefits, such as the facts that the engine can continuously operate in its most efficient region and that we may need no transmission, the series HEVs have some serious drawbacks, such as the fact that they require large battery packs, the engine works hard to maintain battery charge and the multiple energy conversions cause inefficiency.

2.1.2 Parallel Configuration

In a *parallel* design [5], [17], both the engine unit and the battery unit are connected directly to the vehicle's wheels. Usually the engine unit is used alone for highway driving and until a power threshold is reached, while the battery unit adds power during high-demand periods, such as hill climbs, acceleration, and other. The parallel configuration corrects the disadvantages of the series configuration. In addition, the vehicle has more power because both the engine and the motor supply power simultaneously and a separate generator is usually not needed because the motor charges the batteries.

2.1.3 Series-Parallel Configuration

A third type which combines the benefits of both above options is the *series-parallel* or *split* or *dual* configuration. The series-parallel configuration allows the engine to directly drive the wheels and at the same time charge the energy storage device through a generator, if needed. Both the engine and the battery can drive the car alone, depending on the conditions. A series-parallel configuration is described in detail in section 4.1.

2.2 General Rule-Based Control Methods

The general rule-based control strategies do not use any past or future route knowledge, but they only rely on components maps and fuzzy intuitive control techniques, in order to manage how much power is flowing to or from each component [16]. The state of charge (SOC) of the battery, namely the stored energy divided by the maximum energy which can be stored in it, has to take values within a narrow region of operation, in order to avoid fast corruption of the battery and the high cost for its replacement. Very high or very low states of charge would have life-reducing impacts on the battery. The above techniques set rules on both charging and discharging the battery, subject to that constraint of keeping the SOC within that narrow range which is considered safe for the preservation of the battery. For instance, in general rule-based techniques the controller tends to bias the battery operation towards the midpoint of the allowed SOC region, while a strategy which uses future route information might allow complete discharging down to the lower SOC bound, because the future road conditions, e.g. an upcoming downhill, are expected to offer recharging. In other words, because of lack of route information, this SOC biasing within a region of safe operation is applied similarly on all the different types of cycles, although some knowledge of the cycle of interest might suggest less conservative, but still safe charging or discharging for maximum efficiency.

2.2.1 Charge-Sustaining and Charge-Depleting Operation

An important distinction among general rule-based contol strategies is whether they adopt the charge-sustaining or the charge-depleting operation.

Charge-sustaining is a mode of operation, which is based on the above described biasing around the midpoint of the safe SOC region. Specifically, the SOC fluctuates usually within the narrow range $40\% - 60\%$, having as general target the midpoint 50%. Discharge patterns consist of many shallow discharges which utilize only a portion of the battery pack's chemical energy. Since charge-sustaining HEVs seldom drive far on electric power alone, the battery pack can be relatively small, making the vehicle more affordable, and improving efficiency by reducing weight. In a charge sustaining HEV, the hybrid powertrain is capable of providing sufficient energy, independent of the storage device, to drive the vehicle just like it was a conventional vehicle. The engine has to be adequately sized to meet the average power load and, if operated under the expected conditions, will be able to maintain adequate electrical energy storage reserves indefinitely. As long as the hybrid has fuel for the engine, the vehicle will operate.

Charge-depleting refers to a mode of vehicle operation that is dependent on energy from the battery pack. It is the mode of operation of electric vehicles. Charge-depleting vehicles allow their batteries to become depleted and cannot recharge them at the same rate they are being discharged. Some HEVs (mostly plug-in HEVs) start with a charge-depleting strategy, and switch to charge-sustaining mode after the battery has reached its minimum SOC threshold. The hybrid power source in a charge-depleting HEV is only able to provide recharging energy and cannot supply the necessary energy to drive the vehicle by itself. If a charge-depleting HEV requires instantaneous power to accelerate and the hybrid power source is not capable of providing the whole amount of the needed power, the enginegenerator cannot produce the required energy necessary to accelerate the vehicle. This system must have additional energy from the battery to meet the power needs of the vehicle.

2.2.2 Local Optimization

Local optimization is a more systematic method which has still no-cycle knowledge, but takes into account the instantaneous road power demands. The main idea of the method is to assign an expected replacement cost to the battery usage and decide how to derive for every operating point the total needed power from the two sources in order to minimize their total cost. The total cost can be expressed as the sum of the expected replacement cost of the energy of the battery and the cost of the fuel that is consumed by the engine at the specific operating point. A technique to roughly predict the fuel cost of recharging the battery could take into account the instantaneous vehicle speed, which represents the kinetic energy that can be recaptured later through regenerative breaking by the battery. Although this method can be more efficient, it does not avoid completely the SOC biasing, because the prediction of the battery recharging cost includes important uncertainties.

2.3 Route-Based Control Methods

The route-based control methods are the main object of the most recent research on optimally controlling the HEVs. They are the most promising methods for achieving maximization of fuel economy.

2.3.1 Pattern Recognition Approach

This approach consists of two procedures. The first procedure is an off-line classification of all the possible driving patterns, during which an optimal tuning of significant parameters is chosen after many simulations to represent each one of those patterns. The parameters that are employed vary from method to method, but they usually include the target SOC, a nominal power demand threshold and an SOC threshold to trigger engine operation. The second procedure realizes on-board pattern recognition based on the current and previous driving conditions [9]. Then, assuming that the specific driving pattern will continue, the vehicle applies the optimal control tuning

that was off-line computed for the assumed driving pattern. The weakness of the specific method is that the upcoming cycle may not coincide with the past pattern. As a result, the current approach offers modest improvement.

2.3.2 Pattern Learning Approach

This method is also based on optimal parameter-tuning and on-board pattern recognition. The difference here is that the off-board classification of driving patterns is realized using statistical pattern learning through repetitive simulations [13], [18]. Specifically, multiple simulations are realized in order to characterize statistically the different driving patterns. Then, a new component is added which combines the driving environment, the style of the driver and the operating mode of the vehicle in order to identify a driving situation, based on its statistical properties. Although this approach is more fuel efficient than the pattern recognition method, its main drawback is that it requires a large amount of repeating simulations in order to obtain representative statistical features and achieve reliable pattern learning.

2.3.3 Route Prediction Approach

This is the most recent and promising research direction towards maximizing fuel economy of HEVs. While predicting the future driving conditions was very uncertain some years ago given that it could only rely on doubtful and intuitive heuristics, it has recently become possible thanks to the development of powerful systems, such as GPS, ITS and GIS, which can provide the vehicle with the essential geographic and traffic data. Specifically, information such as road grade, stop signs, traffic lights, speed limits and traffic flow data can be offered to the vehicle and allow it to efficiently predict the future route. Then the predicted cycle can be used for global optimization of the whole route. Most route prediction methods use route classifiers, statistical traffic models [14], [8] and some form of segmentation of the entire route. The segmentation is sometimes based on creating nearly equal-distance segments and sometimes on forming segments with the same road characteristics or segments that

belong to the same route class. Some of the route prediction approaches realize multiple parameter tunings in order to achieve the optimal parameter combinations, while some other employ dynamic programming techniques [19], [3], [12], [7], which succeed in significant reduction of computational cost. Dynamic programming, which was abandoned in the past because of the weakness to predict the future route, can now be reemployed as one of the most suitable methods to solve the global optimization problem.

Chapter 3

Dynamic Programming - An Essential Tool

Dynamic programming is an exact technique, which can be applied to a specific class of problems, whose objective function represents an additive cost. Specifically, the idea is to split the additive problem of interest into subproblems and use the optimal solutions of the subproblems in order to compute the global optimal solution of the initial problem [2]. Consequently, dynamic programming could be characterized as a "divide and conquer" method. The general dynamic programming process can be described in steps:

1. The problem is divided into subproblems. Every subproblem can be considered as a stage of a procedure which leads sequentially to a feasible solution of the overall problem. In other words, we can view every feasible solution of the primary problem as a sequence of decisions taken at each stage. In the same sense, the total cost of a feasible solution will be the sum of the costs of the partial stage decisions that led to the specific feasible solution.

2. Every stage can be seen as a "moment" at which a new decision has to be made. We can define the *state* as a summary of all relevant past decisions. The state of every stage will affect the decisions of all the next stages.

3. Possible state transitions from stage to stage are determined. Let the cost of each state transition be the cost of the corresponding stage decision.

4. A recursive cost function is defined, which starts from the terminal state and moves towards the previous states (backward recursion). This function can be called "cost-to-go" function, because it expresses the cost needed to move from the n_{th} state to the terminal state N+1. The "cost-to-go" function for the terminal state is fixed.

Let us consider the following optimization problem

$$\min_{u \in U^N} F(u) \tag{3.1}$$

where u the decision variable over which we wish to minimize the additive cost function F. The feasible set U^N can be either discrete or continuous and the cost function F either linear or nonlinear. Let us assume that the problem is physically divided in N stages (Fig. 3-1), at each one of which a new quantity is added to the cost function, and let u be an N dimensional vector whose components $u_1, u_2, ..., u_N \in U$ represent the decisions that should be made at each stage. According to the idea of the Dynamic Programming we can consider N subproblems based on the N stages. Then the optimal solutions of the subproblems will be used sequentially in order to compute the optimal vector u.

To be more specific, we define a recursive cost function

$$J^*(x_n) = \min_{u_n \in U} \{J^*(x_{n+1}) + w_{n \to n+1, u_n}\} \tag{3.2}$$

$$\Downarrow$$

$$J^*(x_n) = \min_{u_n \in U} \{J^*(f(x_n, u_n)) + w_{n \to n+1, u_n}\} \tag{3.3}$$

where $x_{n+1} = f(x_n, u_n)$ the state of stage $n+1$ which summarizes all the already taken decisions (past decisions) in stages $n+1, n+2, ...N$, $u_n \in U$ the control variable, namely the decision that should be taken at stage n, f a function that gives us the state of stage $n+1$ if we know the state x_n of the previous stage n and the decision u_n taken there and finally $w_{n \to n+1, u_n}$ the cost to move from stage n to stage $n+1$ when the decision taken at n is u_n. The sum of all the costs $w_{n \to n+1, u_n}$, $n = 1, 2, ..., N$, is the value of the objective function F. The recursion that Equation (3.3) represents is

$$J^*(x_n) = \min_{u_n \in U} \{J^*(f(x_n, u_n)) + w_{n \to n+1, u_n}\}$$

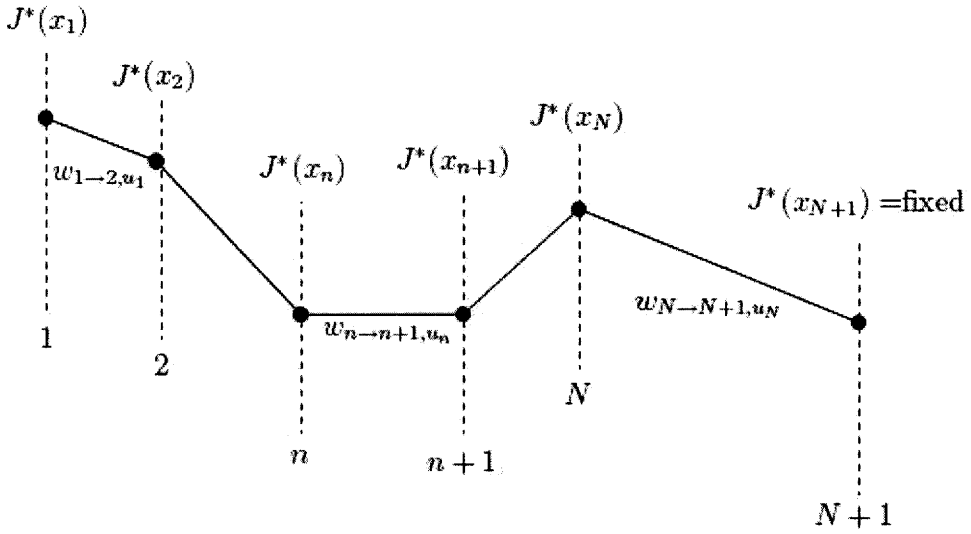


Figure 3-1: Description of Bellman-Ford algorithm.

called Bellman−Ford algorithm. Explaining the above formulas in words, the optimal cost $J^*(x_n)$ of every subproblem and every stage n is computed by minimizing over all the sums of the optimal cost-to-go function $J^*(x_{n+1})$ at state $n+1$ plus the cost to move from state n to state $n+1$, for all the possible decisions u_n that can be taken at stage n. It is clear that the optimal solution u of the primary problem, that is the one that results in the minimal cost to travel from the initial to the final state, will be the sequence of the decisions $u_n, n = 1, 2...N$ that led to the value $J^*(x_1)$.

In case that the feasible set is continuous, its quantization is necessary in order to make the state-space finite and tractable.

Chapter 4

Detailed Description of the Method

The review presented in chapter 2 indicates some of the weaknesses of the methods that have already been developed to optimally control hybrid electric vehicles. As we saw there, general rule-based control strategies offer noticeable but still modest fuel efficiency results. In pattern recognition methods, the usage of past driving conditions as a precursor of the future is often misleading and can reduce their efficiency. On the other hand, pattern learning approaches are more fuel efficient, but a considerable computational price is paid. Route prediction approaches are the most recently studied and most effective HEV controlling methods so far. We seek to develop a method of this kind, which applies the dynamic programming algorithm on a conveniently segmented route. The contribution of our method is exactly this novel segmentation, which is implemented in a way that increases the effectiveness of dynamic programming.

4.1 Necessary Background

Before we proceed to the formulation of our problem, it is necessary to familiarize with the configuration of the hybrid electric vehicle whose operation we are interested in optimizing, and relate this configuration to the scope of our problem.

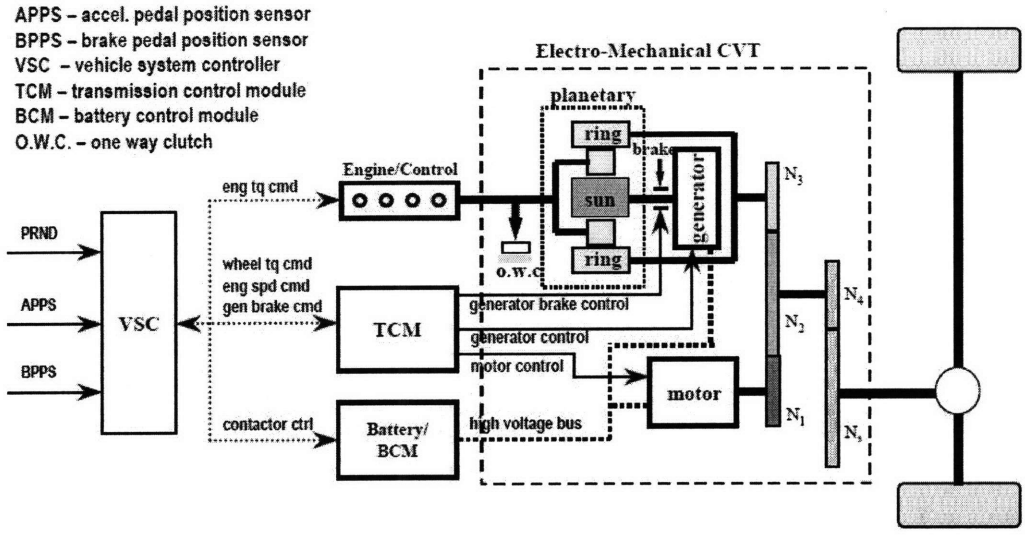


Figure 4-1: Series-parallel powertrain system.

In Fig. 4-1 we can see a typical series-parallel powertrain configuration with its control system. The basic components of this configuration are the engine, the battery, a power split device called planetary gear set, a generator and a motor. The engine is connected to the generator through the planetary gear set, while the battery is connected to the generator and the motor and can be recharged or discharged by both of them. The planetary gear set splits the power produced by the engine and transfers part of it to drive the wheels, and the rest to the generator to either provide electric power to the motor or to recharge the battery.

In order to capture all possible operating modes and integrate the two power sources (engine, battery) to work together seamlessly and optimally, and meet the driver demand, a hierarchical and sophisticated vehicle control system is needed (VSC). Inherent in this controller is a logical structure to guide the vehicle through its various operating modes and a dynamic control strategy associated with each operating mode to specify the vehicle demands to each subsystem. As shown in Fig. 4-2, the controller takes as inputs the environmental conditions, the driver's demands and the current state of the vehicle, and gives as outputs the torque and speed commands for the several components. Then, the operation of the powertrain system under these commands must result in optimal fuel efficiency and desired performance.

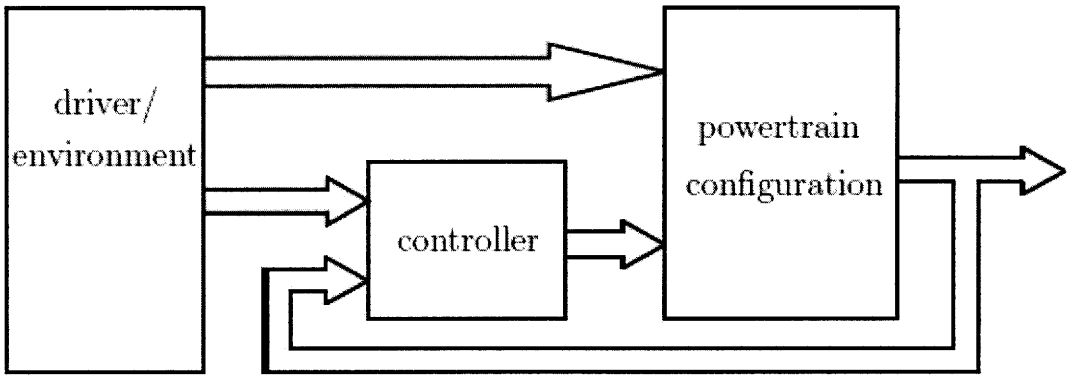


Figure 4-2: HEV simulation model.

In the case of our controller, the driver's demand and the environmental conditions inputs are included and summarized into a speed model and a value for the road grade. Both will be inserted to the controller externally. On the other hand the current state of the vehicle passes directly from the powertrain outputs back into the controller and can be adequately described by the current SOC, the current engine speed and engine torque, and the current motor speed and motor torque. In order to achieve optimal fuel economy, what we mainly want to control are the transmissions from charging to discharging mode and the durations of charging and discharging periods during the trip. Ideally, we would like to prescribe the SOC of the battery at every moment of the trip. The first challenge of this problem is that the trip is continuous and the moments infinitely many, so it is impossible to prescribe completely the SOC trajectory of the trip. Consequently, we choose to decompose the route into segments and try to control the SOC at the end of each one of those segments.

4.2 Brief Statement of the Problem

4.2.1 Notation

We are given a route between an origin (O) and a destination (D), which is decomposed into a series of route segments connected to each other and linking the origin to the destination (Fig.4-3). Let the route be decomposed into N segments. For

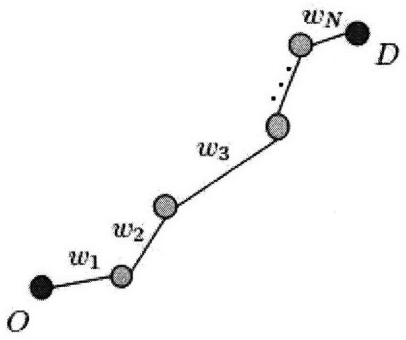


Figure 4-3: General problem - route segmentation.

each road segment i, $i = 1, 2, ..., N$, the average road grade g_i, the average speed v_i and the segment length l_i are available. We assume a predicting model $v_i(t)$ of the vehicle speed trajectory along the route based on the speed limits, the length l_i and the road type. Let w_i be the fuel consumption along the i_{th} segment, in other words the segment's cost. Let also SOC_i and SOC_{i+1} be the state of charge of the battery at the beginning and the end of the i_{th} segment respectively. It is obvious that w_i depends on the length l_i, the grade g_i, the vehicle speed $v_i(t)$ and the SOC at the edges of the segment (SOC_i, SOC_{i+1}). It also depends though on the current vehicle state, which includes the engine speed and torque ($\omega_{eng}(i)$, $\tau_{eng}(i)$) and the motor speed and torque ($\omega_m(i)$, $\tau_m(i)$) at the entrance of the i_{th} segment. Hence, w_i can be written as a complicated nonlinear function of all these parameters: $w_i\big(l_i, g_i, v_i(t), \omega_{eng}(i), \tau_{eng}(i), \omega_m(i), \tau_m(i), SOC_i, SOC_{i+1}\big)$.

4.2.2 Goal

Our goal is to minimize the total fuel consumption along the route:

$$\min_{SOC_i} J = \sum_{i=1}^{N} w_i\big(l_i, g_i, v_i(t), \omega_{eng}(i), \tau_{eng}(i), \omega_m(i), \tau_m(i), SOC_i, SOC_{i+1}\big), \qquad (4.1)$$

$$\text{subject to } SOC_{min} \leq SOC_i \leq SOC_{max}$$

where J will be the objective function, SOC_{i+1}, $i = 2, 3, ..., N+1$, the control variables of our optimization problem, as explained in section 4.1, and SOC_{min}, SOC_{max} the minimum and the maximum value that the state of charge can take (section 2.2). Therefore, our main objective is to create a controller capable of prescribing the set-points for the SOC at the edges of each road segment, which result in the most fuel efficient travel between the origin and the destination.

The second challenge of our problem is the fact that J is a very complicated non-linear function. Consequently, we have to deal with a difficult nonlinear constrained optimization problem, for which we need to apply one of the most powerful tools of nonlinear optimization, namely dynamic programming.

4.2.3 Controller

As already mentioned in section 4.1, we need to build a hierarchical and sophisticated controller (Fig.4-4). The controller is composed of two levels: a high-level controller which plans the trip, by prescribing the SOC at the end of each road segment and a low-level controller that uses the predicted trip to compute an optimal charging/discharging strategy to make the trip fuel efficient. The road segments may not correspond to uniform units such as miles or yards, in order to provide more efficient aggregation of the relevant road conditions.

Within each road segment, we employ the low-level controller (section 4.3). The low-level controller is fixed and is implemented with a high-fidelity HEV model available within PSAT (Powertrain System Analysis Toolkit). We use it like a black box, giving it inputs and taking its outputs. Specifically the PSAT model takes as inputs the vehicle state $(\omega_{eng}(i), \tau_{eng}(i), \omega_m(i), \tau_m(i), SOC_i)$, the grade g_i, the vehicle speed $v_i(t)$ and a desired value for the SOC at the end of the segment (SOC_{i+1}). Using the cycle information and the state of the vehicle, the low-level controller applies an energy management strategy (sections 2.1,2.2) in order to efficiently track the target SOC value (SOC_{i+1}) at the end of the current segment. The output of the controller is the consumed fuel along the segment w_i. In other words the low-level controller can give us the expected fuel consumption w_i along a segment i for every possible

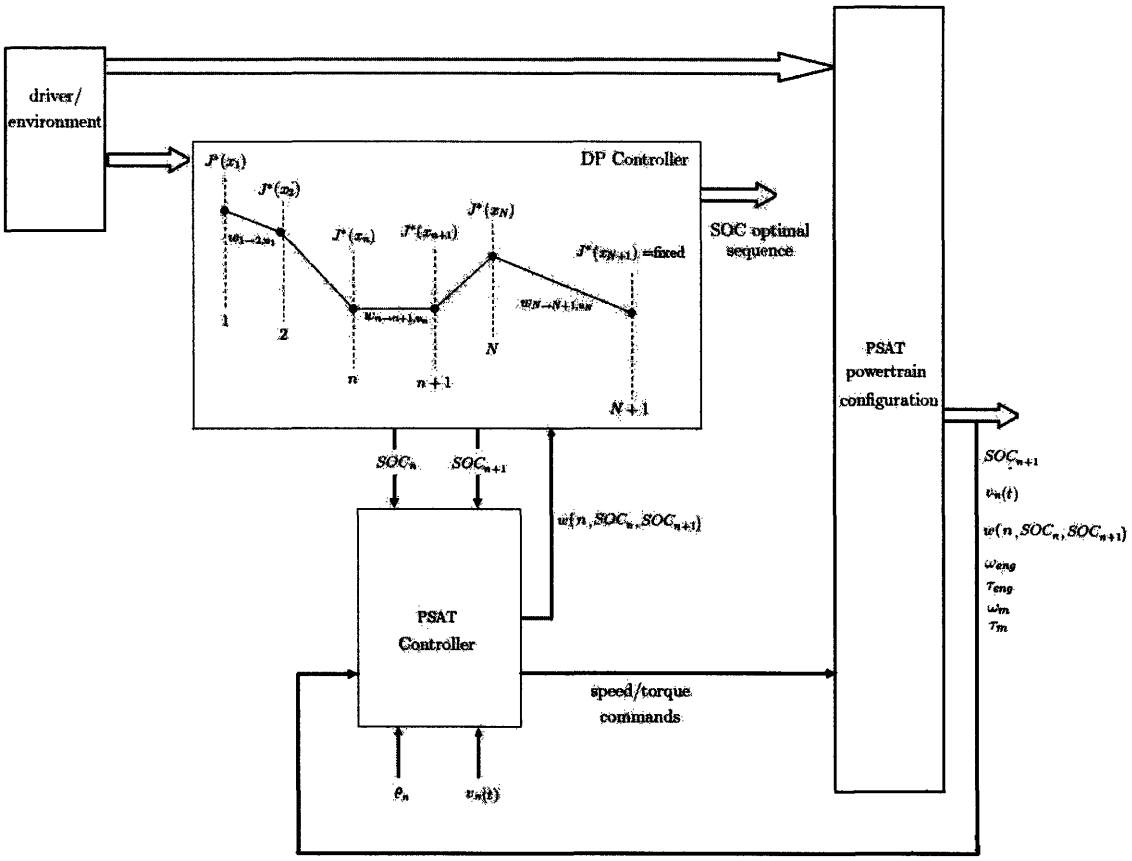


Figure 4-4: Controller.

pair (SOC_i, SOC_{i+1}).

The high-level controller (Sec. 4.4) can be represented by the Bellman-Ford algorithm (Fig.3-1). It can be called "planner" because it plans the trip, by prescribing the SOC at the end of each road segment. If we imagine the given route as a series of route segments connected to each other with nodes and linking the origin to the destination, then the control will be updated at every node. Let i be the current node, $i = 1, 2, ..., N+1$ (counting from the origin to the destination), where $i = 1$ and $i = N + 1$ represent the origin and the destination respectively. We assume that the segment starting from node i takes the same index i. The controller has two domains for its state $x(i)$, the node i itself and the state of charge $SOC(i)$ at the same node. Furthermore, at node i the controller has one control input, the desired SOC for the

next node ($SOC_d(i)$). So, the dynamics of the controller are given by:

$x(i+1) = f\big(x(i), SOC_d(i)\big)$, where $x(i) = \begin{pmatrix} i \\ SOC(i) \end{pmatrix}$ and f a nonlinear function, which generates every state from its precedent state.

After this brief analysis, it is easy to express our objective in the form of Bellman's Equation:

$$\begin{aligned} J^*(x) &= \min_{SOC_d} \{J^*\big(f(x, SOC_d)\big) + w\} \\ J^*(x_f) &= 0 \end{aligned} \tag{4.2}$$

where w is the expected fuel consumption if we travel from node i to node $i+1$ and will be computed using the low-level controller. The latter equation stands for the initial condition of our DP problem and it represents the fact that at the final state $x_f = x(N+1)$, that is the state at the destination, we do not need to move any further, so the expected fuel consumption to the destination and thus the cost function at x_f are 0. Hence, we seek to construct the optimal control law $SOC^*(x)$ that is the minimizer of Eq (4.2).

A considerable challenge of the current dynamic programming problem is the fact that the state-space contains all possible values of (n, SOC) which are infinite, because SOC is a continuous variable. But nonlinear optimization over an infinite state space may be not tractable. As a solution to this problem we choose to quantize the values of SOC in order to reduce the problem into a finite state space. Suppose the quantization is $SOC_i \in \{SOC^1, SOC^2, ..., SOC^n\}$ with $SOC^1 \leq SOC^2 \leq ... \leq SOC^n$, then every node i is associated with all possible quantization values as shown in Fig. 4-5. As a consequence, the number of all possible values that the expected fuel consumption for each segment can take, is equal to the amount of all possible combinations (SOC_i, SOC_{i+1}), with SOC_i and SOC_{i+1} quantized. The number of all these possible combinations is obviously n^2 and thus the expected fuel consumption can take n^2 different values $\{w_i^1, w_i^2, ..., w_i^{n^2}\}$.

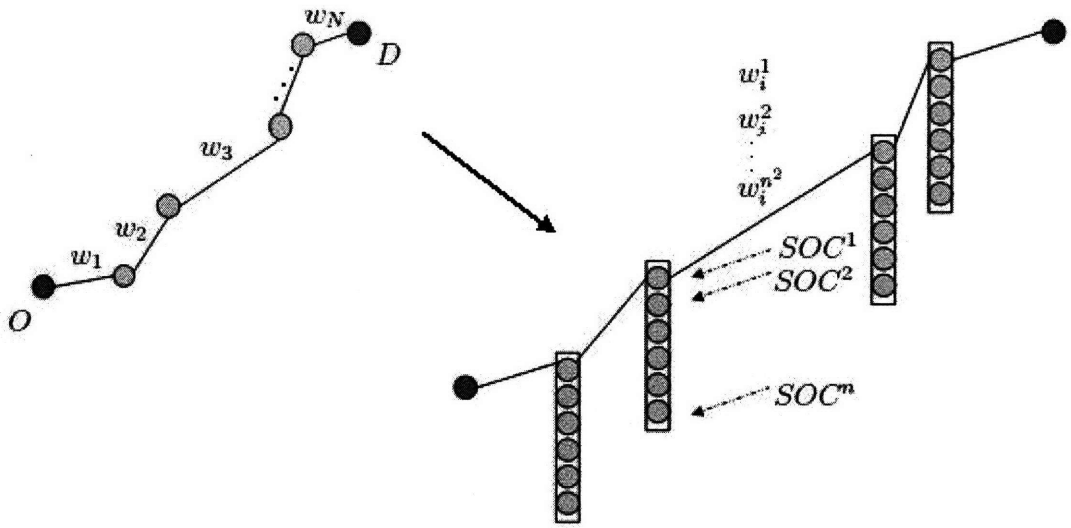


Figure 4-5: SOC quantization.

4.3 Implementation of Low-Level Controller

As already mentioned in the previous section, the low-level controller that we used is a high-fidelity HEV model available within PSAT. PSAT takes as inputs the average grade g_i, the vehicle speed $v_i(t)$, the SOC at the beginning of the current segment (SOC_i) and a desired value for the SOC at the end of the segment (SOC_{d_i}) and gives the expected fuel consumption of the segment as an output. While the average grade is a known constant and the SOC can only take n specific values, the speed trajectory that the vehicle will follow along the segment is not fixed. Consequently, it is our responsibility to create a reliable predictive speed model, based on the knowledge we have about the length and the average speed of the segment. In this section we first describe the PSAT model that we utilized and then we explain how we came up with a predictive speed model.

4.3.1 PSAT Model

PSAT stands for Powertrain System Analysis Toolkit. It is an easy-to-use simulation tool developed by Argonne under the direction of Ford, General Motors, and Daim-

lerChrysler and with the support of the U.S. Department of Energy. PSAT simulates vehicle fuel economy and performance in a realistic manner, taking into account transient behavior and control system characteristics. A driver model follows any given driving cycle, sending a power demand to the vehicle controller, which, in turn, sends a demand to the propulsion components. Component models react to the demand and feed their status to the vehicle controller, and the process iterates on a sub-second basis to achieve the desired result, similar to the operation of a real vehicle controller.

Specifically for our problem, we used the model shown in Fig. 4-6, which is based on the Toyota Prius model. It simulates a light HEV with series-parallel configuration (Sec. 2.1).

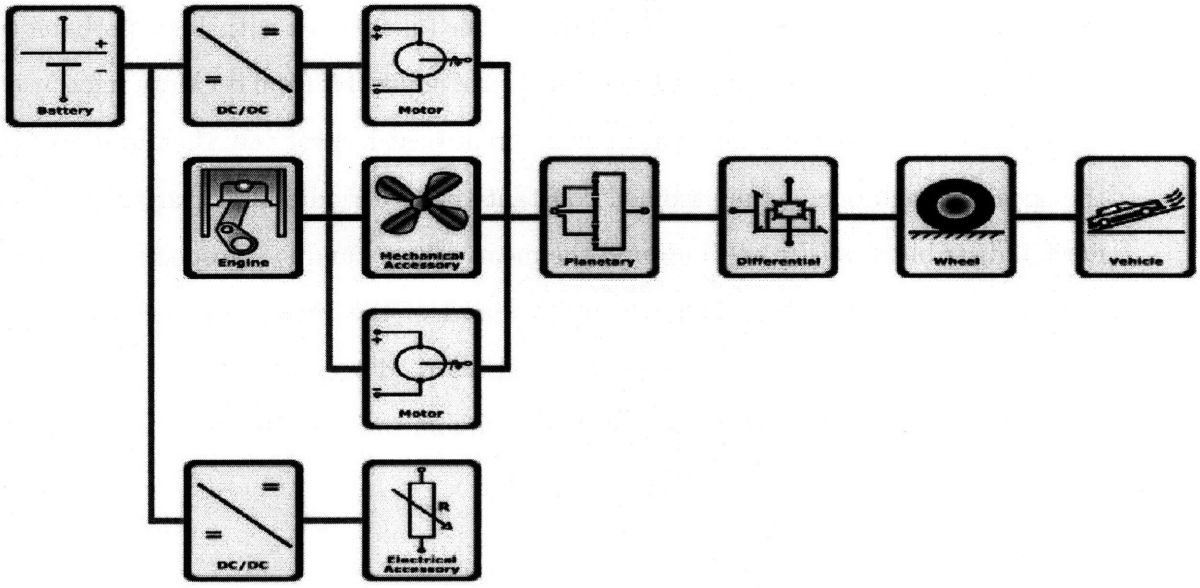


Figure 4-6: PSAT model.

4.3.2 Predictive Speed Model

As we already pointed out, one of the inputs we feed into the PSAT controller, the speed transient $v_i(t)$, is not directly available. The only relevant information we have at our disposal is the length and the average/nominal speed of each road segment. Therefore, we had to invent a reliable model, capable of predicting the speed behavior

of the vehicle. What is important in our case is not the vehicle speed itself, but the fuel that we expect to be consumed, independently of any rational deviation from the predicted speed. In other words, the predictive model that we created does not aim to accurately predict the speed curve. Something like that would anyway be impossible, given that in real-time the speed is influenced by multiple external factors, such as the traffic, the mood and personality of the driver, the time, the day and many other. We wanted our model to be useful in the sense that, when we use its prediction as an input to the low level controller, no matter how realizable this prediction is, it should result in correct values for the expected fuel consumption on each road segment.

The main idea of our prediction method follows. We assume that for each segment the speed trajectory consists of successive constant-speed periods of short but random duration. During these short periods, speed can take values with high probability around the nominal speed v_{avg} and with lower probability far from it. More rigorously, consider two random variables t and v which represent respectively the duration and the speed of each one of the assumed constant-speed periods. Both variables follow normal distributions (Fig. 4-7) around the mean values $\bar{t}$ and v_{avg} respectively, where $\bar{t}$ the average time interval over which the speed of the vehicle can be considered almost constant and v_{avg} the average speed of the road segment. In case the normal distribution generates negative values for our random variable, we reject all the values till a positive value comes up again.

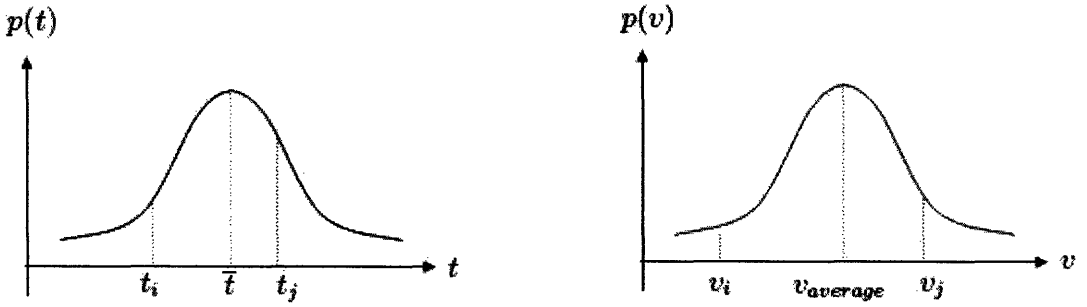


Figure 4-7: Normal distributions out of which the duration and the speed of constant-speed periods are drawn.

Furthermore, we assume that the speed trajectory of each segment begins with

either the speed of the preceding road segment or with speed equal to zero, if it is the first segment of the route. The generation of constant-speed periods stops when the total length of the segment is exceeded. In that case, we reject the last generated period and we replace it with a new period whose constant speed is equal to the speed of the rejected period and its duration is derived as the time needed to traverse the remaining length with the latter speed.

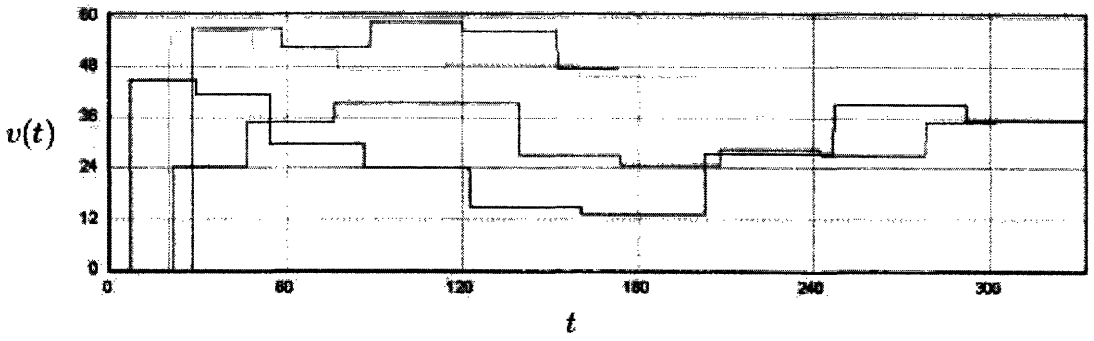
So, let (t_i, v_i) be the i_{th} pair generated from the two gaussian distributions of a specific road segment, representing the i_{th} steady-speed period. Let also $L_i = \sum v_i \cdot t_i$ be the traversed length by the end of the i_{th} period. If $L_i > L$, where L the total length of the segment, then the pair (t_i, v_i) is replaced by $(\frac{L-L_{i-1}}{v_i}, v_i)$. It is obvious that this last pair leads to the end of the segment, since $L_i = L_{i-1} + \frac{L-L_{i-1}}{v_i} \cdot v_i = L$

Hence, creating periods of random duration and random constant speed around the nominal speed of the segment, we generate a random and fictitious speed model for the specific segment. If we repeat this procedure many times and compute with PSAT the fuel that is consumed for each one of the generated models along the specific road segment, then we can find the expected fuel consumption of the segment, by computing the average fuel of all the models. In Fig. 4-6 two sets of 5 randomly generated speed models for two different road segments are depicted. The first segment is the initial segment on its route because it starts from standstill (Fig. 4-8(a)), while the second follows a segment of nominal speed $60 miles/h$ (Fig. 4-8(b)).

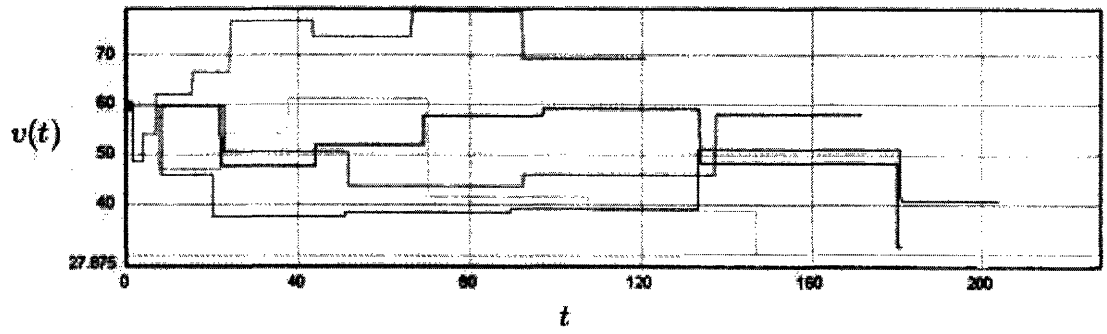
Although the steep changes from one value to another of the produced speed trajectories render them non feasible in real-time, the way they are generated and the fact that the expected fuel consumption can be derived as the average of many different possible routes shows that, although the route prediction does not represent accurately a single possible cycle, it can give the reliable value for the expected fuel consumption on any road segment.

4.3.3 Off-Line Fuel Consumption Table

In the previous subsections we saw how we can compute the expected fuel consumption of a road segment, using a speed predicting method and the PSAT model. Specifi-



(a) Starting from standstill, $v_{avg} = 30miles/h, L = 1.5miles.$



(b) Following a segment with average speed $60miles/h$,$v_{avg} = 45miles/h, L = 2.4miles.$

Figure 4-8: Examples of predictive speed models.

cally, we explained how we generate several random cycles based on the characteristics of the segment and how by averaging the fuel consumption of those cycles, we can get the expected fuel consumption for any possible cycle on this segment. This way we create off-line a table including the expected fuel consumption for every segment i and for every possible combination (SOC_i, SOC_{i+1}), that is a table that contains all the values $\{w_i^1, w_i^2, ..., w_i^{n^2}\}$ for $i = 1, 2, ..., N$. This off-line procedure is essential for the high-level controller implementation, which needs the values $\{w_i^1, w_i^2, ..., w_i^{n^2}\}$, $i = 1, 2, ..., N$ in order to optimize quickly and globally the total fuel consumption.

In Fig. 4-9 we give as an example the off-line fuel consumption table of a specific road segment. We assume a three-level quantization of the SOC, that is $SOC_i \in \{SOC^1, SOC^2, SOC^3\} = \{40\%, 50\%, 60\%\}$, where $40\% - 60\%$ the allowed range of the SOC values. The rows of the table represent the initial SOC of the segment i (SOC_i), while the columns stand for the different values of the SOC target at the end of the road segment (SOC_{i+1}). So, for example, for initial SOC 50% and final SOC 40% the expected fuel consumption is $0.07kg$.

target SOC

initial SOC	40	50	60
40	0.15	0.21	[illegible]
50	0.07	0.16	0.21
60	0.02	0.07	0.16

Figure 4-9: Example of the off-line fuel consumption table for a specific road segment.

It is interesting to notice how the numbers are related to each other. First of all, we can see that the numbers increase as we move from bottom-left to top-right. This is also highlighted by the increasing darkness on the increasing direction of the numbers. This is something we expected, because for example in order to change the charge of the battery from 40% to 60% the vehicle needs much more fuel ($0.25kg$) and effort than in the case where the SOC diminishes from 60% to 40%, where less effort from the engine and less fuel is needed ($0.02kg$). Another interesting relationship among the numbers is the fact that they are diagonally almost equal, where by diagonally this time we mean from top-left to bottom-right. This shows that for similar changes

in the SOC, e.g. in case of increase or decrease by 10% or in case of no change, the fuel consumption values are also similar, independently of the individual values of SOC_i and SOC_{i+1}. The only thing that matters is the difference of these two quantities. This fact indicates that for similar SOC changes the strategy of the vehicle is the same.

Finally, we see that for a three-level quantization of the SOC ($n = 3$), the possible values that the expected fuel consumption can take on each segment are indeed $n^2 = 9$.

4.4 Implementation of High-Level Controller

As we discussed in section 4.2, for the implementation of the high-level controller we employ the Bellman-Ford algorithm (Fig.3-1). We also saw that in order to apply dynamic programming to our problem, we needed to build the off-line tables which contain the fuel values $\{w_i^1, w_i^2, ..., w_i^{n^2}\}$ for all the segments $i = 1, 2, ..., N$ of the route. Having those tables ready, the dynamic programming process can lead really quickly to the optimal solution of our problem, that is to the optimal sequence of SOC_i's that will result in the minimum fuel consumption.

Let us describe in detail the Bellman-Ford algorithm as we applied it on our problem. In order to facilitate our description we use as example a simple route consisting of only 3 road segments. The route is shown in Fig. 4-10.

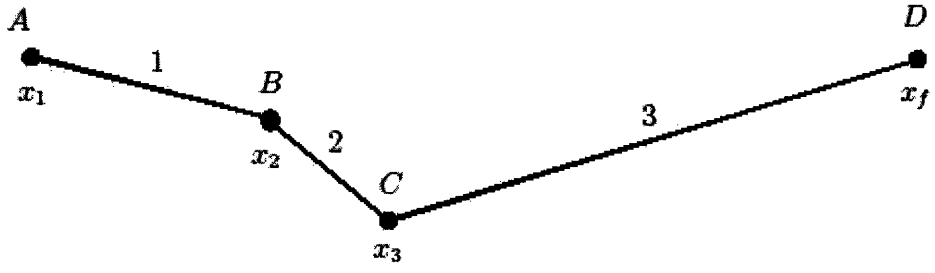


Figure 4-10: Example of a 3-segment route.

The tables of the expected fuel consumptions for the three segments are given in Fig. 4-11. They were built using PSAT and the predictive speed model that we

discussed earlier.

Segment1			Segment2			Segment3		
0.11	0.18	0.24	0.16	0.22	0.26	0.35	0.42	0.46
0.01	0.11	0.18	0.04	0.16	0.19	0.31	0.3	0.42
0	0.01	0.1	0	0.31	0.13	0.25	0.31	0.36

Figure 4-11: Expected fuel consumption tables for the 3 segments.

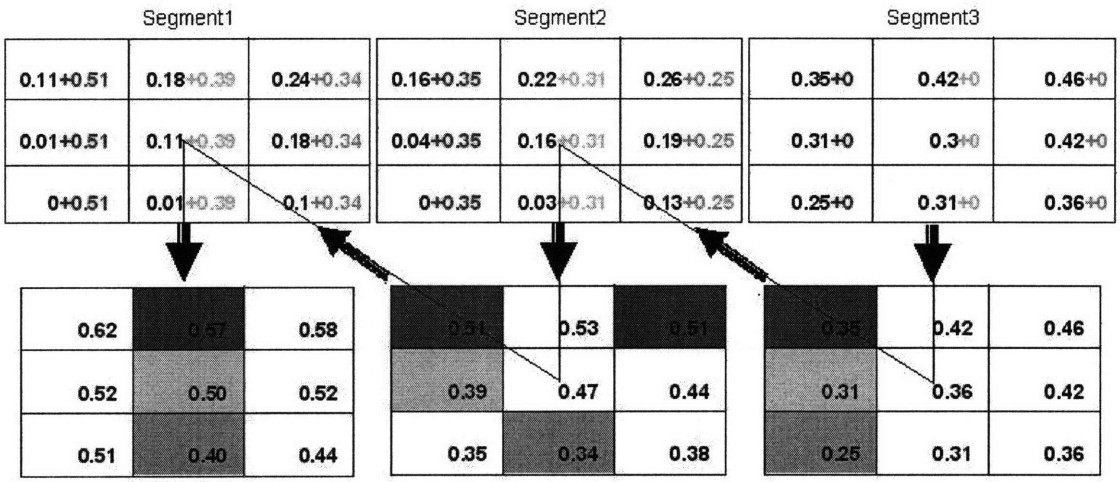
The dynamic programming process that we used consists of two parts. The first part can be characterized as a backward procedure, because it travels through the nodes starting from the destination and finishing at the origin (Fig. 4-12(a)). Similarly, the second part is a forward procedure which traverses the nodes starting from the origin and moving towards the destination (Fig. 4-12(b)).

Let us describe the whole process analytically. Starting from the destination we apply the Bellman-Ford algorithm. As we already mentioned, the value of the cost function at the destination is known and equal to zero, because we are already at the final state $x_f = x(N+1)$, where the expected fuel consumption to reach the destination is clearly equal to zero. Moving backwards, we compute which is the optimal fuel consumption to travel from each one of the nodes C, B and A to the destination D, ignoring every time all the nodes of smaller index. In other words, we solve 3 subproblems of the initial problem, where every time a different node is considered to be the origin, and all the nodes lying on the left of the current origin are ignored (Fig. 4-10).

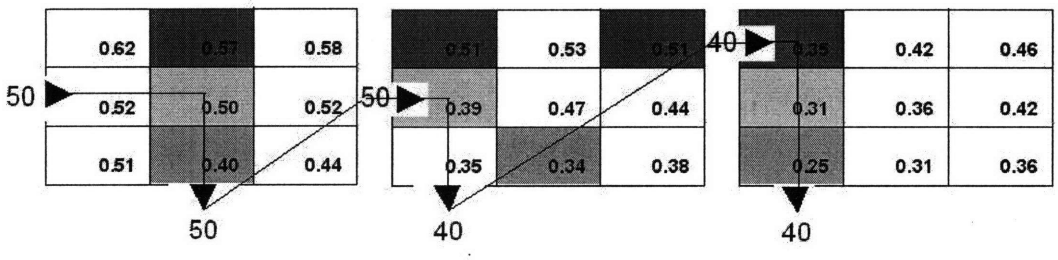
Let us describe in detail Fig. 4-12(a). As a reminder, the rows of each table represent the initial SOC of the segment i (SOC_i), while the columns stand for the different values of the SOC target at the end of the road segment (SOC_{i+1}).

Considering node C as the origin of our trip, the minimum expected fuel consumption is computed with the expression:

$$J^*(x_3) = \min_{SOC_D} \{J^*(x_f) + w_{C \to D, SOC_D}\} \Rightarrow \tag{4.3}$$



(a) First part of dynamic programming procedure - backward.



(b) Second part of dynamic programming procedure - forward.

Figure 4-12: Dynamic programming procedure - example.

$$J^*(C, SOC_C) = \min_{SOC_D} \{J^*(D, SOC_D) + w_{C \to D, SOC_D}\} \tag{4.4}$$

But we know that $J^*(x_f = 0)$, so

$$J^*(C, SOC_C) = \min_{SOC_D} \{w_{C \to D, SOC_D}\} \tag{4.5}$$

which means that, for every possible SOC_C we only need to find the final state of charge SOC_D which results in the minimum fuel consumption. Referring to Fig. 4-12(a), for every row of the rightmost-bottom table, we choose the cell with the minimum value. The corresponding column of the chosen cell represents the SOC_D that minimizes the fuel consumption for the trip $C \to D$ and the SOC_C of the examined row. So, for example, if $SOC_C = 50\%$, the optimal SOC target is $SOC_D = 40\%$ and the minimum fuel cost is $w_3 = 0.31$, which is obvious from the highlighted cell.

Moving to node B, we revise the same procedure. The difference here is that in the expression

$$J^*(B, SOC_B) = \min_{SOC_C} \{J^*(C, SOC_C) + w_{B \to C, SOC_C}\} \tag{4.6}$$

$J^*(x_3) = J^*(C, SOC_C)$ has now different values for each possible SOC target (SOC_C). These are the values of $J^*(x_3)$ that we already computed in the previous step. So, for every SOC_C we add to the corresponding column of the upper-middle table the optimal value $J^*(C, SOC_C)$ that we computed in the previous step, that is the value of the colored cell of the corresponding row of the previously examined table. So, for example, for any SOC_C ($40\%, 50\%$ or 60%), we add to the first column of the upper-middle table the value 0.35, which represents the minimum fuel consumption if we travel from C to D, starting with $SOC_C = 40$. This way, we can compute the minimum fuel consumption on the path $B \to C$, without recomputing anything on segment 3. All the information we needed from segment 3 is included in the values $J^*(x_3) = J^*(C, SOC_C)$. Now we understand the role of the state as a summary of all past decisions.

The lower middle table just computes the sums of its above table. As we did before, from each row we choose and highlight the cells with the minimum values, in order to use them at the next and final step of the backward procedure. Applying exactly the same idea we compute the minimum fuel consumptions for the path $A \rightarrow B$ and for every possible SOC_A.

Given that A is also the origin of our primary problem, the value of $J^*(A, SOC_A)$ is the optimal fuel consumption for the whole route, that is the minimum value of the objective function J (Eq. 4.1). So, the only thing that is left to do, is to go back and pick from all the nodes the SOC values that compose the optimal solution. As mentioned before, this is a forward process, given that this time we start from the origin and move towards the destination. Fig. 4-12(b) will help us understand this procedure. Let us assume that the state of charge of the origin A is known and equal to 50%. We give this value as an input to the leftmost table. The row that corresponds to $SOC_A = 50$ is the second row of the latter table. Using the highlights from the previous procedure we notice that the minimum value for the specific row is met in the middle cell (0.50). This dictates the value 50% as the optimal SOC_B. Taking the value 50 as an output of the left table, we apply it as an input to the middle table. With the same logic we keep $SOC_C = 40$ and $SOC_D = 40$ as optimal. The result of the whole procedure is the sequence $[50, 50, 40, 40]$, which represents the SOC values of the four nodes that result in minimum fuel consumption along the whole route, when the state of charge at the origin is known to be $SOC_A = 50$.

Chapter 5

Experiments-Results

After the familiarization with the techniques of our method, we are ready to present and analyze some of its potentials. In the current chapter we present some experiments in order to study how the grade information availability, the level of route segmentation and the level of SOC quantization can affect the effectiveness of our method. The results indicate that the method can be quite promising.

5.1 Grade Information Necessity

In this section we present a couple of experiments which prove how important it is to employ road grade information in order to achieve higher fuel efficiency. Although the existing digital maps do not include this information, the fact that the largest digital map vendors (Zenrin, Tele Atlas, NavTeq) are considering to add soon grade information to their digital maps, shows that the necessity of having this information available is generally realized.

5.1.1 Grade Experiment 1

Let us consider the route of Fig. 5-1, where the length L is measured in *miles*, the speed v in $miles/h$ and the grade g in %. While for segments 1 and 3 we consider grade equal to zero, for segment 2 we try five different values of grade ($g = 0, 1, 2, 5, 10$).

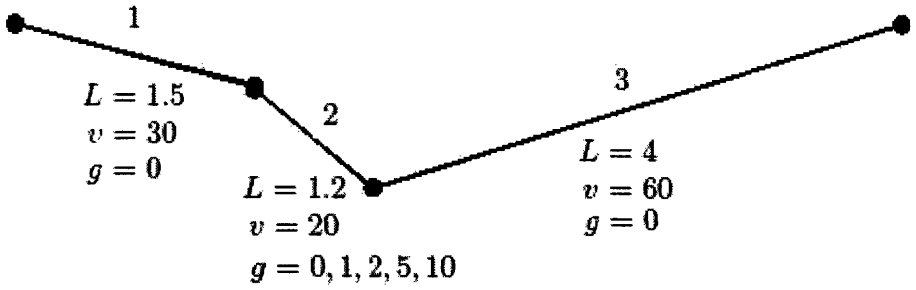


Figure 5-1: Grade experiment route.

In Fig. 5-2 we give the fuel consumption tables of segment 2 for the five different values of its grade. As we saw in chapter 4, the increasing darkness of the cells of each table shows at which direction the fuel consumption augments. Also, the same convention as before holds for the rows and the columns of every table, that is they represent the initial and the final SOC of the segment respectively.

The five tables are diagonally lined up and emphasized with an arrow in order to make their relationship obvious. We notice that, as expected, the fuel consumption increases with the grade for all the correspondent cells. Let us try to quantify this fuel increase. Tables 5.1 and 5.2 present in percentages the increase of the fuel consumption of the segment for the five different values of its grade, using as reference the consumption for $g = 0$ and for two cases of SOC change along the segment (40→60 and 50→40). The interpretation of these tables will reveal the importance of these results. It is more than obvious that the increase rate in table 5.2 is much larger than in table 5.1. For example, for $g = 10\%$ the increase of fuel consumption is 173% for the case of 40→60 and 1300% for the case of 50→40. The significance of this is that, although for $g = 0$ the path 50→40 is much more fuel efficient than 40→60 ($0.04 \ll 0.26$), when the grade increases the latter path becomes much more competitive in relevance to 50→40 ($0.56 < 0.71$). In other words, the bigger the grade gets, the more likely it is for 40→60 path to compete with 50→40 during the process of our optimization algorithm. For $g = 0$, the ratio of the consumed fuel for the two

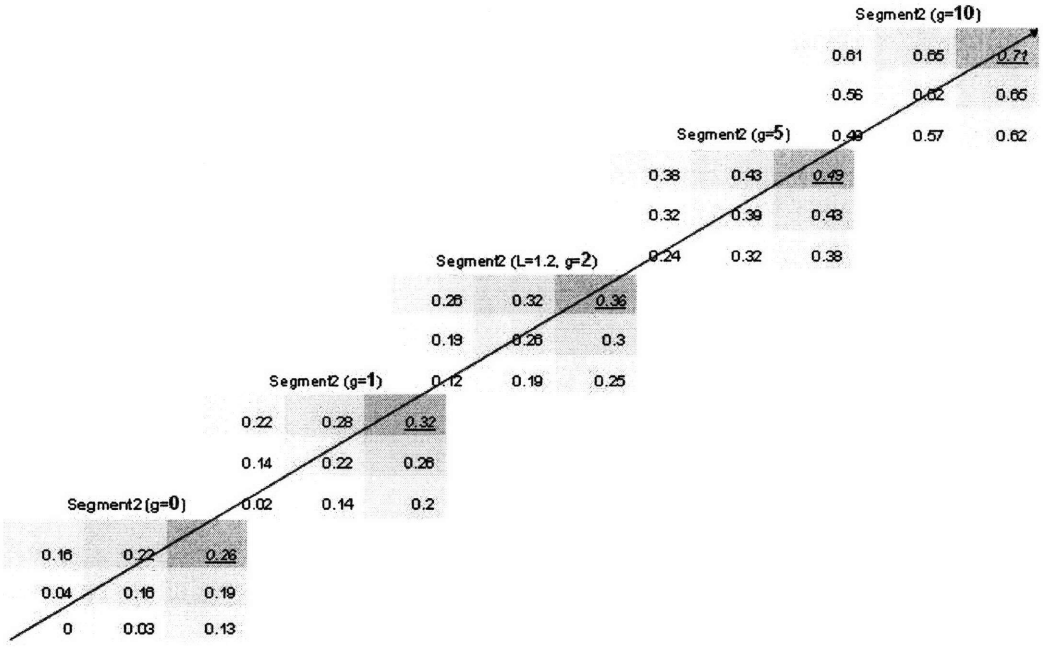


Figure 5-2: Grade increase experiment - fuel consumption tables.

paths is $\frac{0.04}{0.26} = 0.15$ while it is $\frac{0.56}{0.71} = 0.79$ for $g = 10$. Although both ratios are smaller than one, the second ratio is much closer to it than the first one, which shows again that the difference of the two fuel consumptions diminishes with the grade, and the two values become almost equivalently competitive.

The impact of this result on our method is that, if the grade is taken into account, we may increase fuel efficiency by considering paths which become more competitive exactly because of the presence of a grade. In other words, if we ignore the grade pretending that it is zero, probably the chosen paths will actually not be optimal. The following section proves exactly that such a case can happen.

Table 5.1: Grade and fuel consumption interaction for SOC change 40→60.

40→60			
	grade	**fuel consumption**	**increase**
	0	0.26	(0%)
	1	0.32	23%
	2	0.36	38%
	5	0.49	88%
	10	0.71	173%

Table 5.2: Grade and fuel consumption interaction for SOC change 50→40.

50→40			
	grade	**fuel consumption**	**increase**
	0	0.04	(0%)
	1	0.14	250%
	2	0.19	375%
	5	0.32	700%
	10	0.56	1300%

5.1.2 Grade Experiment 2

In this experiment, we try to measure the impact on our method's efficiency of ignoring the grade, pretending that it is zero. Let us continue with the route of the previous example (Fig. 5-1). In Fig. 5-3 we can see the fuel consumption tables of all the three segments for two values of the grade of segment 2, $g = 0$, $g = 1$. Using these two tables and the method described in section 4.4, we compute the optimal SOC sequence and the minimum expected fuel consumption along the route for the two different values of the grade of segment 2 and for initial SOC=50. Specifically, for $g = 0$ we found the optimal SOC sequence $[50, 50, 40, 40]$ and the minimal fuel consumption 0.5, while for $g = 1$ we got $[50, 60, 40, 40]$ and the minimal fuel consumption 0.55. Let us assume that, although the real value of the grade at segment 2 is 1%, we ignore it and we consider $g = 0$ for all the segments. Then, the dynamic programming algorithm will give us the results of the upper part of Fig. 5-3, which means that the SOC sequence that the method will recommend us is $[50, 50, 40, 40]$. Let us compute the fuel consumption of this sequence, using the correct grade's fuel values, namely the lower table of Fig. 5-3. The bold numbers of the later table represent exactly the sequence $[50, 50, 40, 40]$ and the expected fuel consumption for this sequence is their sum: $0.11 + 0.14 + 0.35 = 0.6$. So the error that emerges because of the grade ignorance is a 9.0909% increase of the minimum fuel consumption as computed in Fig. 5-3. Consequently, if we ignore the real grade of segment 2 and we set it zero, we will recommend to the HEV an SOC sequence which increases the fuel consumption

9.0909% with respect to its optimal value. In connection to the previous section, we see that even a small increase in the grade of segment 2 renders the seemingly less efficient choice 50→60 on segment 1 more competitive than 50→50.

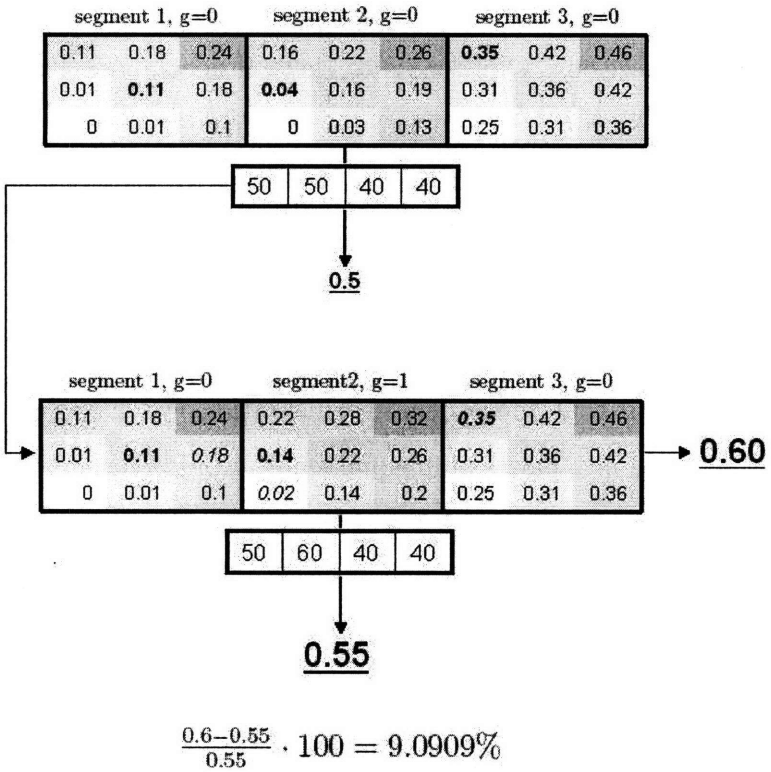


Figure 5-3: Grade ignorance experiment.

It is obvious from the last two subsections that the availability of grade information makes our method much more efficient.

5.2 Route Segmentation

In this section we present a series of experiments, which study different levels and ways of route segmentation and evaluate their impact on our method's efficiency. Based on the results of these experiments we can suggest some rules on how to optimally decompose a specific route and create the novel segmentation method that we have

already mentioned from the beginning of this thesis.

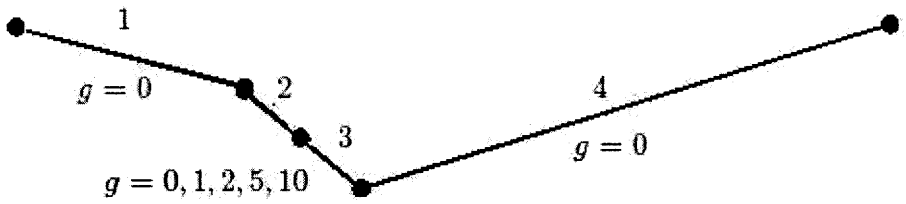
5.2.1 Route Segmentation - Part 1

The idea of this experiment is to check how the impact of a further segmentation of a segment changes, when the grade of the segment increases. Let us consider segment 2 from the previous sections. Our goal is to check if decomposing the segment itself into multiple smaller segments would increase the fuel efficiency that our algorithm achieves. The intuition suggests that the answer to this question probably changes for different segment grades. Consequently, we repeat the experiment for five different values of the grade of segment 2 ($g = 0, 1, 2, 5, 10$).

Segmentation-in-two: We start by splitting segment 2 into two new segments of equal length. Given that the original segmentation was based on common road characteristics, cutting into pieces of unequal length would not offer anything special. Considering now the route of Fig. 5-1 and the new segmentation, we get the same route with one supplementary control point (Fig. 5-4(a)). Computationally, this means that instead of computing 9 values of expected fuel consumption for segment 2, we compute $2 \cdot 9 = 18$ values corresponding to the two new segments. Using these values we compute the optimal fuel consumption of the entire route and we compare the results of the two segmentations in table 5-4(b). Specifically, this table presents the total fuel consumptions of the newly segmented route for all the grade values of segment 2, as percentages of the respective fuel consumptions for the initial three-segment route. We see that with the increase of the grade, the percentages diminish. For example, for g=1 on segment 2, the fuel consumed on the finer decomposed route (Fig. 5-4(a)) is slightly improved ($\approx 95\%$) but still close to the consumption of the initial route shown in Fig. 5-1. The same percentage is much smaller ($\approx 75\%$) in the case of g=10. Therefore, we realize that the bigger the grade of segment 2 is, the more beneficial it is to further segment it. For g=0 the percentages are sometimes slightly above and sometimes slightly below 100%. These small deviations are a consequence of the fact that the expected fuel consumptions are computed stochastically, so statistical errors are expected to appear. The errors become almost invisible for

bigger grades, showing that the improvement that the further segmentation provides is large enough to stay unaffected by the statistical inaccuracies.

Concluding with the segmentation-in-two, although for small values of the grade the tradeoff between the fuel efficiency improvement and the additional computational cost may suggest to avoid making the segmentation finer, when the grade becomes large the fuel efficiency improvement is significant and probably the additional computational cost of the finer segmentation is a worth-paying price.



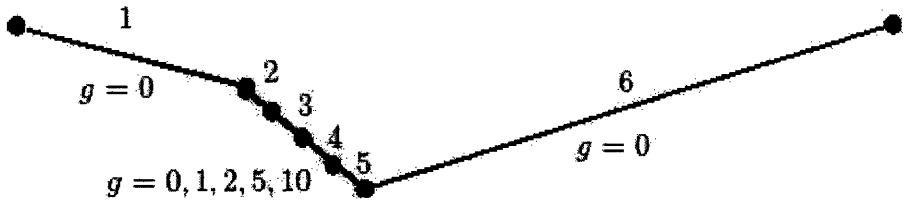
(a) Segmentation of segment 2 in two pieces.

%	g_2=0	g_2=1	g_2=2	g_2=5	g_2=10
40	103	100	90	86	76
50	98	93	87	82	73
60	107	94	85	82	77
	103	**95**	**88**	**83**	**75**

(b) Improvement table for segmentation in two pieces. Each row represents a different SOC at the origin of the route.

Figure 5-4: Segmentation-in-two.

Segmentation-in-four: Exactly similarly to the previous experiment, we keep splitting segment 2 into smaller segments of equal length. The new route we get includes two new control points (Fig. 5-5(a)). Computationally, this means that we now have to compute $4 \cdot 9 = 36$ fuel consumption values corresponding to the four new segments. Using these values we compute the optimal fuel consumption of the entire route and we compare the results in table 5-5(b). As expected, the results are



(a) Segmentation of segment 2 in four pieces.

%	g_2=0	g_2=1	g_2=2	g_2=5	g_2=10
40	104	93	69	72	65
50	98	85	64	68	61
60	113	98	68	72	63
	105	**92**	**67**	**71**	**63**

(b) Improvement table for segmentation in four pieces. Each row represents a different SOC at the origin of the route.

Figure 5-5: Segmentation in four.

on the same direction as before. Here also, the increase of the grade is combined with fuel efficiency improvement and particularly the improvement is clearly bigger compared to segmentation-in-two. For example, for g=10 on segment 2, the fuel consumption reduces here by $\approx$ 63%, while the correspondent percentage is $\approx$ 75% when segmenting by two. Table 5.3 shows the improvement that segmentation-in-four offers in comparison to segmentation-in-two.

Table 5.3: From segmentation-in-two to segmentation-in-four

g_2	0	1	2	5	10
	102%	97%	69%	82%	80%

Summarizing the current section, we conclude that the bigger the grade of a segment is, the finer the segmentation should be, in order to maximize the potentials of our method. Nevertheless, we should never forget that increasing the segmenta-

tion adds significant computational cost to generating the fuel consumption tables. Specifically, for an n-level quantization of the SOC, every new segment adds n^2 new computations of the expected fuel consumption along that segment. It is, hence, of great importance to be able to define the optimal tradeoff between the fuel efficiency improvement and the additional computational cost that a finer segmentation would cause in the case of a segment with a large grade. We may need to employ grade thresholds, the surpassing of which would allow us to move to the next level of segmentation.

5.2.2 Route Segmentation - Part 2

In the previous section we studied how the grade of a segment can indicate whether it is useful to further decompose that segment or not. In the current section we examine how the changes of the grade and the nominal speed along the whole route should affect the level of segmentation. In other words, the two sections present two different approaches of the same concern, which is to define some rules according to which we will be deciding how fine the route segmentation should be.

Employ Average Grade: With this experiment we check how much our fuel efficiency deteriorates when we do not segment our route based on the changes of grade, but instead we assume unique grade equal to the average grade of the whole route. We try to track how much our results are affected by choosing this grade uniformity.

Let us consider once more the route of Fig. 5-1, assuming though common nominal speed for the three segments. Our goal is to check if the efficiency of our algorithm decreases when we do not segment the route according to the grade changes, but we consider instead one average value for the grade of the whole route. The average grade of this route is

$$\frac{\sum_{i=1}^{3} L_i \cdot g_i}{\sum_{i=1}^{3} L_i} \tag{5.1}$$

where L_i, g_i the length and the grade of the i_{th} road segment, $i = 1, 2, 3$. Using our

method we optimize the fuel consumption along the whole route considering its grade uniform and equal to g (Fig. 5-6). According to equation 5.1, for $g_2 = 1, 2, 5, 10$ the average grade of the whole route is $g = 0.2, 0.4, 0.9, 1.8$ respectively.

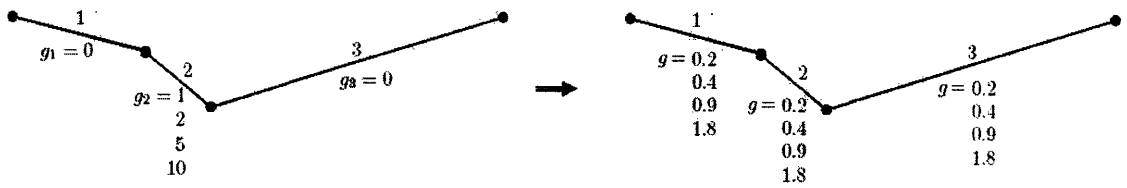


Figure 5-6: Non-segmented route - uniform grade of average value.

Realizing the experiment we noticed that the adoption of a uniform value for the grade does reduce the efficiency of our algorithm. Furthermore, this efficiency deterioration increases with the grade. Table 5.4 summarizes the results of the experiment. Specifically, it presents the total fuel consumptions of the route for all the average values of the grade, as percentages of the respective fuel consumptions for the case where we keep the real grade values on each segment. We see that for small grade of segment 2 (e.g. $g_2 = 1 \Rightarrow g = 0.2$) the computed fuel consumption remains the same (100%), while if we increase the grade, the percentages increase too. For example, for $g_2 = 10$ we have an increase of 14% in the optimal fuel consumption compared to the consumption that we would get if we considered separately the real grades of the three segments.

Table 5.4: Non-segmented route - uniform grade of average value.

g_2	**1**	**2**	**5**	**10**
g	0.2	0.4	0.9	1.8
	100%	108%	120%	114%

The essence of these results is that, if along a part of our route the grade takes generally small values, then we can assume uniform average grade for that part, without serious loss of efficiency. On the other hand, if the grade presents serious fluctuations and reaches higher values, then it is recommended to segment the route

according to the grade changes locations.

Employ Average Nominal Speed: With this experiment we check how much our fuel efficiency deteriorates when we segment our route based only on the changes of the grade and not on the changes of the nominal speed. In other words, we assume unique nominal speed equal to the average nominal for the whole route and we track how much our results are affected by employing this speed uniformity.

Let us consider the route of Fig. 5-1 with the difference that we assume constant grade g for the whole route. Our goal is to check if the efficiency of our algorithm decreases when we do not segment the route according to the nominal speed changes, but we consider instead one average value for the speed of the whole route. The average speed of this route is

$$\frac{\sum_{i=1}^{3} L_i \cdot v_i}{\sum_{i=1}^{3} L_i} \tag{5.2}$$

where L_i, v_i the length and the speed of the i_{th} road segment, $i = 1, 2, 3$. Using our method we optimize the fuel consumption along the whole route considering its nominal speed uniform and equal to v (Fig. 5-7). According to equation 5.2, the average speed of the whole route is $v = 46.1194$. We repeat the experiment for two values of grade, $g = 0$ and $g = 2$.

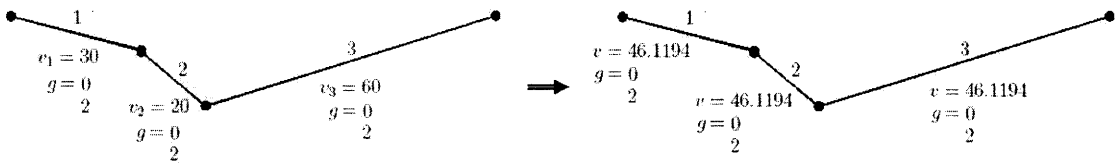


Figure 5-7: Non-segmented route - average nominal speed.

Realizing the experiment we noticed that the adoption of a uniform value for the nominal speed does reduce the efficiency of our algorithm. Furthermore, this efficiency deterioration increases with the grade. Table 5.5 summarizes the results of the experiment. Specifically, it presents the total fuel consumptions of the route for both $g = 0$ and $g = 2$, as percentages of the respective fuel consumptions for the case

where we keep the real values of nominal speed on each segment. We see that for zero grade of the route the computed fuel consumption deteriorates slightly (102%), while even for small increase of the grade of the route, the percentage increases significantly (113%).

Table 5.5: Non-segmented route - average nominal speed.

g_2	**0**	**10**
g	0	1.8
	102%	113%

The essence of these results is that, if along a part of our route where the grade remains constant we do not segment that part according to the speed changes locations, but instead we assume a uniform average speed, then we notice some loss of efficiency. This loss increases with the increase of the grade and it remains greater than zero even for $g = 0$. Hence, we conclude that it is prudent to always segment our route whenever the nominal speed changes significantly.

Summarizing the current section, we conclude that whenever we meet significant changes of the grade or the nominal speed along our route, it is meaningful to base our segmentation on the locations of those changes. In other words, it is highly recommended to add an SOC control point on our route wherever a speed or grade change is located. Especially when the route grade takes generally large values, the adoption of the above segmentation becomes even more important.

5.3 SOC Quantization

The goal of this experiment is to study how the SOC quantization affects the efficiency of our method. Let us consider the left route of Fig. 5-6. We compute the optimal fuel consumption for two different levels of the SOC quantization, that is for $SOC_i \in \{40, 50, 60\}$ and for $SOC_i \in \{40, 45, 50, 55, 60\}$. We realize the experiment for two values of the route grade, $g = 0$ and $g = 2$. The results are shown in Fig. 5-8.

$g = 0$

40	0.57	0.57
45		0.54
50	0.5	0.5
55		0.45
60	0.4	0.4

$g = 2$

40	0.92	0.89
45		0.86
50	0.87	0.84
55		0.8
60	0.79	0.76

Figure 5-8: SOC quantization

The second column of each one of the tables shows the expexted fuel consumption of the route for the three-level quantization ($SOC_i \in \{40, 50, 60\}$) and the third column represents the five-level quantization ($SOC_i \in \{40, 45, 50, 55, 60\}$). We see that for grade $g = 0$ nothing really changes for the two quantizations, but when the grade of the route increases to $g = 2$ it is obvious that our method becomes more efficient. For example, for SOC at the origin equal to 50, the expected fuel consumption reduces from 0.87 to 0.84 when we use the five-level instead of the three-level quantization. Generally, we notice a $3\% - 4\%$ improvement of the fuel consumption when we compare the five-level to the three-level quantization.

Therefore, with this experiment we verify as expected that a finer quantization of the SOC values leads to higher fuel efficiency, especially for big values of the grade. Still, there is again a concern which reminds us of section 5.2.1. Increasing the SOC quanta from three to five, we add $5^2 - 3^2 = 16$ new computations of the expected fuel consumption along each one of the segments, that is for the specific route $3 \cdot 16 = 48$ additional computations. It is, hence, of great importance to be able to define the optimal tradeoff between the fuel efficiency improvement and the additional computational cost that a finer SOC quantization would cause in the case of a route that contains parts of significant grade. We may even choose to adopt a mixed SOC segmentation, such as a three-level segmentation for parts of the route

with small grade and five-level segmentation, for the higher grade parts.

Chapter 6

Conclusions and Future Work

In this work, we presented a new approach to optimally controlling hybrid electric vehicles. The main advantages of the hereby presented method are that it reduces the computational complexity of the problem by applying Bellman's dynamic programming and it increases fuel efficiency by introducing new rules to optimally segmenting the route. These rules can be summarized below:

- Add on the route of interest a new control point at each location of significant grade or nominal speed change.
- Apply further segmentation on segments of large grade.
- Unify several segments in one, only in case of small speed changes and small grade values. This way the computational cost may be considerably diminished, without violating optimality.

Exactly this novel way of segmenting the route of interest is the main contribution of the current work. Additionally to the above rules, other rules concerning the SOC quantization can be considered, in order to boost the potentials of our algorithm. Finally, the widely realized but still not less important result that the availability of grade information is essential for the efficiency of the route-based control methods, is one more small contribution of the current thesis, in the sense that the multiple proof of a single conclusion by many different works makes the conclusion stronger.

Future work should include the following:

- Define specific grade thresholds, the surpassing of which would allow moving to the next level of segmentation of a single segment.
- Define specific grade thresholds, the surpassing of which would allow moving to the next level of SOC quantization.
- Define specific grade and speed change thresholds, below which several segments can be unified.
- Develop an improved model of vehicle speed prediction.
- Introduce a feedback system to ensure the high level controller's robustness.

Apart from the above, many different directions of future work are recommended, considering the potentials of the broader field of route-based control methods.

Bibliography

[1] B. M. Baumann, G. Washington, B. C. Glenn, and G. Rizzoni. Mechatronic design and control of hybrid electric vehicles. *Mechatronics, IEEE/ASME Transactions on*, 5(1):58–72, 2000.

[2] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control.* 3rd edition, 2007.

[3] A. Brahma, Y. Guezennec, and G. Rizzoni. Optimal energy management in series hybrid electric vehicles. In *American Control Conference, 2000. Proceedings of the 2000*, volume 1, pages 60–64 vol.1, 2000.

[4] C. C. Chan. The state of the art of electric and hybrid vehicles [prolog]. *Proceedings of the IEEE*, 90(2):245–246, 2002.

[5] S. Delprat, J. Lauber, T. M. Guerra, and J. Rimaux. Control of a parallel hybrid powertrain: optimal control. *Vehicular Technology, IEEE Transactions on*, 53(3):872–881, 2004.

[6] M. Gokasan, S. Bogosyan, and D. J. Goering. Sliding mode based powertrain control for efficiency improvement in series hybrid-electric vehicles. *Power Electronics, IEEE Transactions on*, 21(3):779–790, 2006.

[7] Qiuming Gong and Yaoyu Li. Trip based optimal power management of plug-in hybrid electric vehicle with advanced traffic modeling. April 2008.

[8] Dirk Helbing. Derivation and empirical validation of a refined traffic flow model, May 1998.

[9] Soo-Il Jeon, Ki-Back Kim, Sung-Tae Jo, and Jang-Moo Lee. Driving simulation of a parallel hybrid electric vehicle using receding horizon control. In *Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on*, volume 2, pages 1180–1185 vol.2, 2001.

[10] Jun M. Kang, Ilya Kolmanovsky, and J. W. Grizzle. Dynamic optimization of lean burn engine aftertreatment. *Journal of Dynamic Systems, Measurement, and Control*, 123(2):153–160, 2001.

[11] Ilya Kolmanovsky, Michiel van Nieuwstadt, and Jing Sun. Optimization of complex powertrain systems for fuel economy and emissions. *Nonlinear Analysis: Real World Applications*, 1(2):205–221, June 2000.

[12] M. Koot, J. T. B. A. Kessels, B. de Jager, W. P. M. H. Heemels, P. P. J. van den Bosch, and M. Steinbuch. Energy management strategies for vehicular electric power systems. *Vehicular Technology, IEEE Transactions on*, 54(3):771–782, 2005.

[13] R. Langari and Jong-Seob Won. Intelligent energy management agent for a parallel hybrid vehicle-part i: system architecture and design of the driving situation identification process. *Vehicular Technology, IEEE Transactions on*, 54(3):925–934, 2005.

[14] Louis A. Pipes. An operational analysis of traffic dynamics. *Journal of Applied Physics*, 24(3):274–281, 1953.

[15] B. K. Powell, K. E. Bailey, and S. R. Cikanek. Dynamic modeling and control of hybrid electric vehicle powertrain systems. *Control Systems Magazine, IEEE*, 18(5):17–33, 1998.

[16] N. J. Schouten, M. A. Salman, and N. A. Kheir. Fuzzy logic control for parallel hybrid vehicles. *Control Systems Technology, IEEE Transactions on*, 10(3):460–468, 2002.

[17] A. Sciarretta, M. Back, and L. Guzzella. Optimal control of parallel hybrid electric vehicles. *Control Systems Technology, IEEE Transactions on*, 12(3):352–363, 2004.

[18] Jong-Seob Won and R. Langari. Intelligent energy management agent for a parallel hybrid vehicle-part ii: torque distribution, charge sustenance strategies, and performance results. *Vehicular Technology, IEEE Transactions on*, 54(3):935–953, 2005.

[19] U. Zoelch and D. Schroeder. Dynamic optimization method for design and rating of the components of a hybrid vehicle. *International Journal of Vehicle Design*, 19:1–13, 1998.