Design and Evaluation of a 3D Road Geometry Based Heavy Truck Fuel Optimization System

by

Wei Huang

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
August 9, 2010

Keywords: truck fuel optimization, nonlinear programming, 3D road geometry, terrain, map-matching system

Approved by

David M. Bevly, Chair, Associate Professor of Mechanical Engineering
David Beale, Professor of Mechanical Engineering
Song-yul Choe, Associate Professor of Mechanical Engineering
Andrew Sinclair, Assistant Professor of Aerospace Engineering

Abstract

This dissertation develops a 3D road geometry based optimal powertrain control system in reducing the fuel consumption of heavy trucks. The real-time optimal control (OC) system, solving a constrained nonlinear programming problem, is designed to predict and command the optimal throttle, brake torque (generated from the engine retarder), gear shifting, and velocity trajectory to minimize the fuel consumption and travel time. Throttle and gear shifting are continuous and discrete control inputs respectively, which is a mixed-discrete nonlinear programming problem (MDNLP). The optimization solver developed is an interior-point algorithm plus a rounding-off method, where all discrete gear ratios are handled as continuous variables and optimized, and then the optimal discrete gear ratios are obtained by rounding up each continuous gear ratio to the nearest discrete value.

Simulation and experimental tests of a Class 8 truck are conducted with GIS 3D road geometries. Test results show that the optimal control system is able to reduce fuel consumption up to 3.0% with small travel time increases on level and rolling terrains when compared to the defined baseline. When on the highly mountainous terrain with steep crest slope and long slope length, the OC could only save a small amount of fuel. Thus, it is found that the gain in fuel economy is directly effected by the change in terrain type.

Additionally, sensitivity analyses for the terrain and road geometry are conducted to investigate how the change of the terrain and the errors in the terrain data effect the gain in fuel economy and the system performance. For the terrain sensitivity, it is found that the gain in fuel economy is directly related to the change of road grade (both the magnitude and frequency) and slope length. For the terrain error sensitivity, the road error largely effects both the fuel economy and the system performance. This research reveals that the impact from the absolute error (a shift in the slope position) on the fuel savings is not as evident as that from the relative error (a different slope value). The impact from both absolute and relative errors on the system performance is significant, which shows the requirement to apply a high accuracy road map in real road tests as well as in production uses.

Acknowledgments

First and foremost, I would like to thank all of people who encouraged and supported me during the undertaking of this research.

I would present my deep appreciation to my supervisor Prof. David Bevly, for giving me the precious opportunity to join in the GPS and Vehicle Dynamics Lab and conduct such an interesting Ph.D. dissertation project. He, with his strong scientific personality, guided and discussed with me in many aspects.

I would like to express my sincere gratitude to my Ph.D. program committee members, Prof. David Beale, Song-yul Choe, Andrew Sinclair, and Prof. John Y. Hung, for their guidance and advice. They have been a constant source of inspiration, encouragement and assistance during this dissertation work, both in my academic studies and more.

Special thanks to my parents, relatives and all my friends for their continuous support and stimulation of my perseverance in my research activities. Without them this work would have been monumentally harder to carry out.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and objective

Nowadays, heavy trucks account for a high percentage of the US's highway fuel usage. In the 2006, US motor vehicles used 535,809,195,978 liters of petroleum-based motor fuel, 15-20% of which was used by heavy trucks. Manufacturers, therefore, are interested in making trucks more fuel efficient. The major fuel losses of moving trucks are from air drag, rolling resistance, and road slope. The fuel losses resulting from the road slope can be significant in heavy vehicles such as long haul trucks.

The research was performed in collaboration with Eaton and Intermap Corp., to quantify the fuel savings for heavy trucks using Eaton's test truck and Intermap's 3D road geometry. The initial research objectives were:

- Design a 3D road geometry based optimal powertrain control system to predict and command the optimal throttle, brake torque, gear shifting, and velocity trajectory in reducing the fuel consumption and travel time of heavy trucks.

- Investigate how the change of the terrain and the errors in the terrain road geometry effect the gain in fuel economy and the system performance.

1.2 Background and literature survey

Several related research directions are surveyed in this section. Prior work focusing on the relation between the road profile and vehicle fuel economy is introduced first, which is followed by the theoretical study of solving the trajectory optimization problem and the design of the real-time optimal control system.

1.2.1 Road profile based fuel economy research

The interaction between terrain variation and vehicle fuel consumption has gained more and more research interest since the 1970s. The conducted work is generally within two areas: fuel optimal vehicle trajectory optimization and real-time fuel optimal control system design.

Fuel optimal vehicle trajectory optimization

The original trend of this research was to find the optimal speed profile that minimizes vehicle fuel consumption in traversing different paths or routes. The results were then used to support general practice in guideway design for long distance transportation to achieve the best fuel efficiency, e.g., setting speed limit, training new drivers, and generating advice to regulate driving style.

Some work, stemming from [1], formulated an optimal control problem for a vehicle model to minimize fuel consumption and the results suggested that fuel consumption is minimized by operation at a constant speed [2], [3]. In the early work [1], authors developed a feedback algorithm for driving a highway vehicle for minimum fuel consumption. The problem was derived from the Pontryagin maximum principle to provide a mathematically optimal performance subject to the driver's choice of a trip time limit. Authors concluded that the optimal velocity for a level road is a constant velocity. However, they considered the speed limit only at 55 mph and their analytical solution technique obligated them to model the car's fuel consumption rate with a highly simplified function, which may cause a loss of generality.

Authors in [2] addressed the question of what speed profile will minimize fuel consumption of a land transport vehicle based on the general analysis of various vehicle models and driving conditions. Their derivation relied on some assumptions such as the inherent resistance of a vehicle having the usual quadratic form and the approximate proportionality between fuel consumption and propulsive work. However, most engine fuel maps are highly nonlinear related to both engine speed and power, and therefore fuel consumption can not be simply represented as an affine function of the produced work. Authors in [3] used a nonlinear fuel map and conducted tests on short road sections with small road grades.

Given these analyses, driving under the constant speed limit may only be fuel optimal on level roads with small but not significant road grades. In [4], the author found that optimal varied speed driving could gain substantial fuel saving compared to the constant speed driving based on the analysis of a wide range of vehicle models and large road grades. This study used fuel consumption simulators for 15 late-model automobiles to determine how one ought to drive to maximize fuel economy. Dynamic programming was used to determine the optimal way to accelerate from rest to cruising speed, to drive a block between stop signs, and to cruise on hilly terrain while maintaining a given average speed. The dependence of fuel economy on a constant speed was also characterized for various road grades. However in this paper only the single crest, sag, and hills were considered, and therefore the result can not be applied generally to long road sections with combinations of different hills.

Real-time fuel optimal control system design

With the rapid growth of computational power, the 3D road map database, and global positioning system (GPS) technology, the new research trend, starting from [5], is to design real-time predictive powertrain controllers to reduce fuel consumption by using road map data in combination with GPS. Different from the solving of the optimization problem to find the fuel optimal speed trajectory as described in Section 1.2.1, the fast growing research has been aimed to design real-time look ahead control systems which command truck behavior on-line based on the road map and current truck states to minimize the truck fuel consumption and travel time.

In [5], DaimlerChrysler researchers developed a Predictive Cruise Control (PCC) system to reduce fuel consumption of heavy trucks. The PCC basic structure is shown in Figure 1.1, where the position estimation system determines the current vehicle position, the optimization algorithm calculates the optimal engine speed, the 3D road map finds the predictive road grade, and a look-up table stores the map-matched optimal solution. Based on 3D elevation information and a predictive algorithm, the PCC outputs the requested engine speed to ECU and controls the vehicle speed to vary within the bounds of the set speed to reduce fuel consumption. For a designated road profile, the PCC could achieve up to 4% fuel reduction for a selected vehicle.

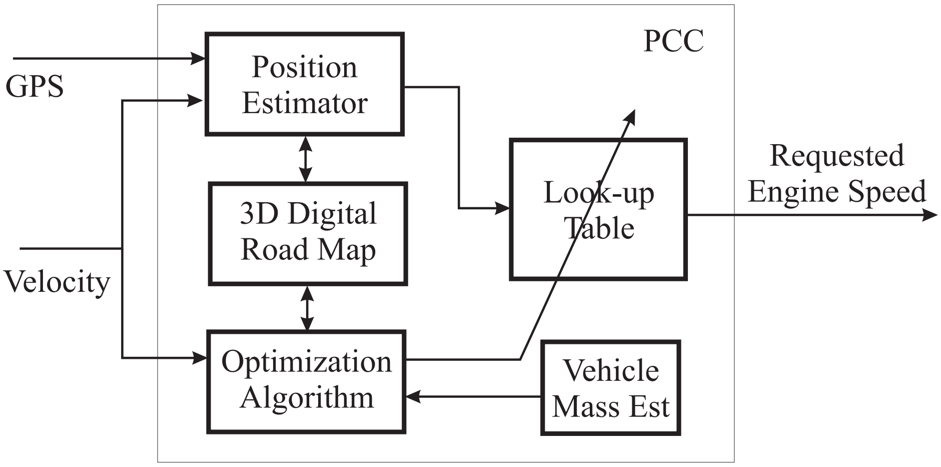


Figure 1.1: The basic structure of the PCC system [5]

Scania Inc., as stated in [6], developed an Expert Cruise Controller (ECC) using look-ahead road information. The ECC is a sort of rule-based control strategy which relies on logical reasoning of how a driver should control the vehicle in order to solve the optimization problem. The main idea in the ECC is dividing the road into several sections, describing uphill and downhill slopes along the road. Different control strategies are implemented for different section types. Up to 3.4% lower fuel consumption was obtained in simulations. However, as a rule-based controller, the ECC sometimes totally lacks feedback, and therefore relies heavily on the accuracy of the road map.

In [7], the author designed the Model Predictive Cruise Control (MPCC) in heavy trucks by using road topography. In this work, a model predictive control (MPC) scheme was used to control the longitudinal behavior of the vehicle, which entailed determining acceleration and brake levels and also which gear to engage. The optimization was accomplished through discrete dynamic programming. A cost function was used to define the optimization criterion to minimize fuel consumption. Computer simulations with a load of 40 metric tons showed that the fuel consumption could be reduced by 2.5% with a negligible change in travel time.

As the continuation of work [7], authors in [8] further devised a look-ahead control system which uses a dynamic programming algorithm in a predictive control scheme by constantly feeding the conventional CC with new set points, as shown in Figure 1.2. The algorithm was evaluated in a real truck on a highway, and the fuel consumption was shown to be reduced by up to 3.5%. Generally speaking, prior work focused mainly on the design of road map based optimal cruise controllers to minimize fuel consumption on designated level terrains. In these cases the road information was generated from GPS measurements.

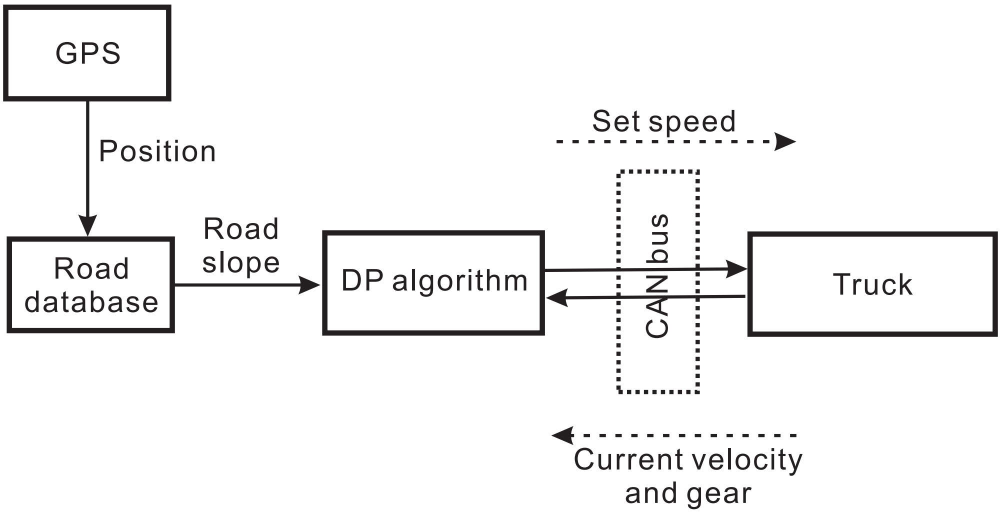


Figure 1.2: The basic structure of the MPCC system [8]

1.2.2 Solving of hybrid optimal control problems

In this dissertation, a 3D geographic information system (GIS) road geometry based optimal powertrain control system is designed to reduce the heavy truck fuel consumption and travel time. The optimizer of the real-time optimal control system (OC) attempts to solve an optimal control problem. Optimal control deals with the problem of finding a control law for a given system such that a certain optimality criterion is achieved. A control problem includes a cost function that is a function of state and control variables. An optimal control is a set of solutions describing the paths of the control variables that minimize the cost function subject to constraints, which can be expressed using the relatively general formulation:

$$
\begin{aligned}
\text{minimize} \quad F[x(t_f), u(t_f), t_f] & & \qquad (1.1)\\
\text{subject to} \quad \frac{dx}{dt} &= f(x, u, t) \qquad (1.2)\\
h(x, u, t) &= 0 \qquad (1.3)\\
g(x, u, t) &\leq 0 \qquad (1.4)
\end{aligned}
$$

where t_f is the final time, and $x(t_f)$ and $u(t_f)$ are the final states and controls, respectively. Additionally, Equation (1.1) is the objective function to be minimized, Equations (1.2) are the differential equations of system dynamics, and Equations (1.3) and (1.4) are a set of system equality and inequality constraints.

Optimal control problem solution

Due to the curse of dimensionality, many optimal control problem cannot be solved analytically. The curse of dimensionality, a term coined by Richard Bellman, is the problem caused by the exponential increase in volume associated with adding extra dimensions to a mathematical space. There are mainly three classes of approaches on finding the numerical solution of the optimal control problem as depicted in Equations (1.1) to (1.4): dynamic programming (DP) based approaches, indirect methods, and direct methods [9].

The DP approach was first described in [10] and was extended to include constraints on the state and control variables in [11]. In [1] and [4], DP was used to find the optimal speed trajectory that minimizes vehicle fuel consumption. In [12] and [8], authors developed a DP based real-time predictive powertrain controller to reduce fuel consumption based on road information. Generally, the use of DP is limited to small problems since the dimensionality increases rapidly with problem size. An advantage of DP is the high probability of finding the global optimum.

Indirect methods [13] focus on obtaining a numerical solution to the classical necessary conditions for optimality according to the Pontryagins maximum principle. The result is a two-point boundary value problem (TPBVP) which can be solved in various ways [9]. Indirect methods have been applied in [2] and [3] to solve the fuel minimization problem with road grade variation, and in [5] to develop a real-time PCC system to reduce fuel consumption of heavy trucks. Some disadvantages of these methods are that they are very sensitive to the initial guess and inequality constraints are difficult to handle.

Direct methods transform the infinite dimensional optimal control problem into a finite dimensional nonlinear programming (NLP) problem and then solve the resulting NLP problem of the following form by using of the optimizer:

$$\begin{aligned} \text{minimize} \;\; f(x) & & \\ \text{subject to} \;\; c_j(x) &= 0, \quad j = 1, \ldots, k \\ c_j(x) &\leq 0, \quad j = k+1, \ldots, m \\ x_{iL} &\leq x_i \leq x_{iU}, \quad i = 1, \ldots, n \end{aligned} \tag{1.5}$$

where f and c_j are the objective and constraint functions respectively, x_{iL} and x_{iU} are lower and upper bounds for the variable x_i, and k, m, and n are the number of equality constraints, total constraints, and design variables, respectively. The functions f and c_j are usually assumed twice continuously differentiable. Reversely, the optimal control problem may be interpreted as an extension of the NLP problem to an infinite number of variables [14]. A structure of methods for solving the optimal control problem is depicted in Figure 1.3.

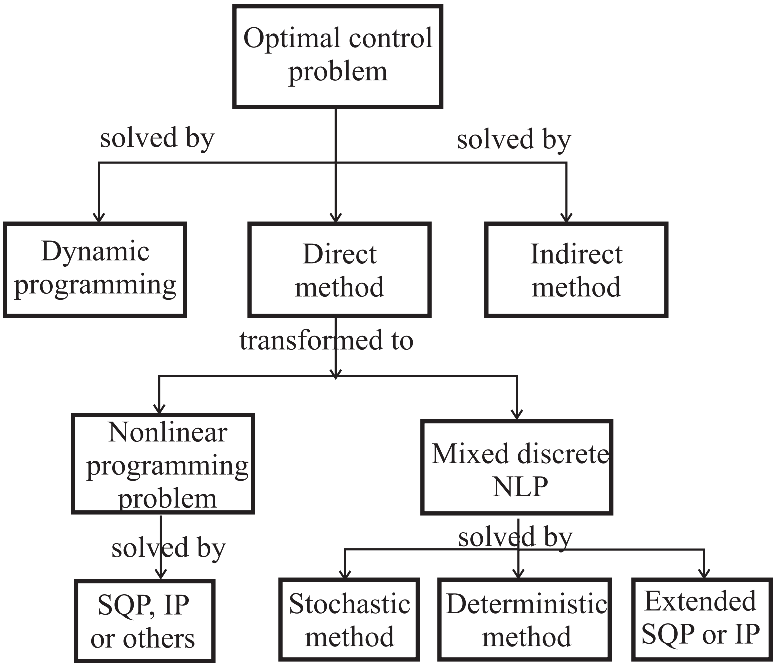


Figure 1.3: Method structure of solving dynamic optimization problem

There are two general strategies of the direct method: the control parametrization method [15] where only the controls u are discretized, and the collocation method [16] in which both the controls u and state trajectories x are discretized using polynomials on finite elements, and the coefficients of these polynomials and element sizes become decision variables in a large-scale NLP. The collocation approach can be quickly used to solve a number of practical trajectory optimization problems.

In the collocation method, the discretization is achieved by first dividing the time interval into a prescribed number of subintervals whose endpoints are called nodes [17]. The unknowns are the values of the controls and the states at these nodes and the state and control parameters. The cost function and the state equations can be expressed in terms of these parameters and can be solved by a standard gradient based nonlinear programming method, or a heuristic method such as simulated annealing. The time trajectories of both the control and the state variables can be obtained by using an interpolation scheme. In most collocation schemes, linear or cubic splines are used as the interpolating polynomials [18].

In [14], [19] and [20], instead of using piecewise continuous polynomials as the interpolation between prescribed subintervals, orthogonal polynomials such as Legendre and Chebyshev polynomials were used as pseudospectral methods for approximating the control and state variables.

The sequential quadratic programming (SQP) algorithm has been one of the most successful gradient based methods for solving NLP problems especially where a significant degree of nonlinearity is present. The SQP method is a procedure that generates iterative converging to a solution of the problem by solving the quadratic approximations to NLP [21]. The SQP method has been widely applied to solve NLP problems in the area of ground and space vehicle path planning, obstacle avoidance, and trajectory optimization [22], [23]. Here the direct collocation method transforms the optimal control problem into a NLP problem by discretizing the vehicle trajectory into a number of segments and approximating the equations of motion along those segments with cubic polynomials.

The SQP algorithm consists of four main steps and its flow chart is illustrated in Figure 1.4 [24]:

a. Compute the gradient of the Lagrange function L: Lagrange function, with the Lagrange multipliers λ, can be written as:

$$L(x, \lambda) \;=\; f(x) + \lambda c(x), \quad x = \{u, v\} \tag{1.6}$$

where x is the parameter vector that needs to be optimized, which may include the control u and state v. The second order approximation of Equation (1.6) is:

$$\begin{aligned} L(x, \lambda) \;=\; & L(x_i, \lambda) + \nabla L(x_i, \lambda) \cdot (x - x_i) + \\ & \frac{1}{2}(x - x_i)^T \cdot H_L(x_i, \lambda) \cdot (x - x_i) \end{aligned} \tag{1.7}$$

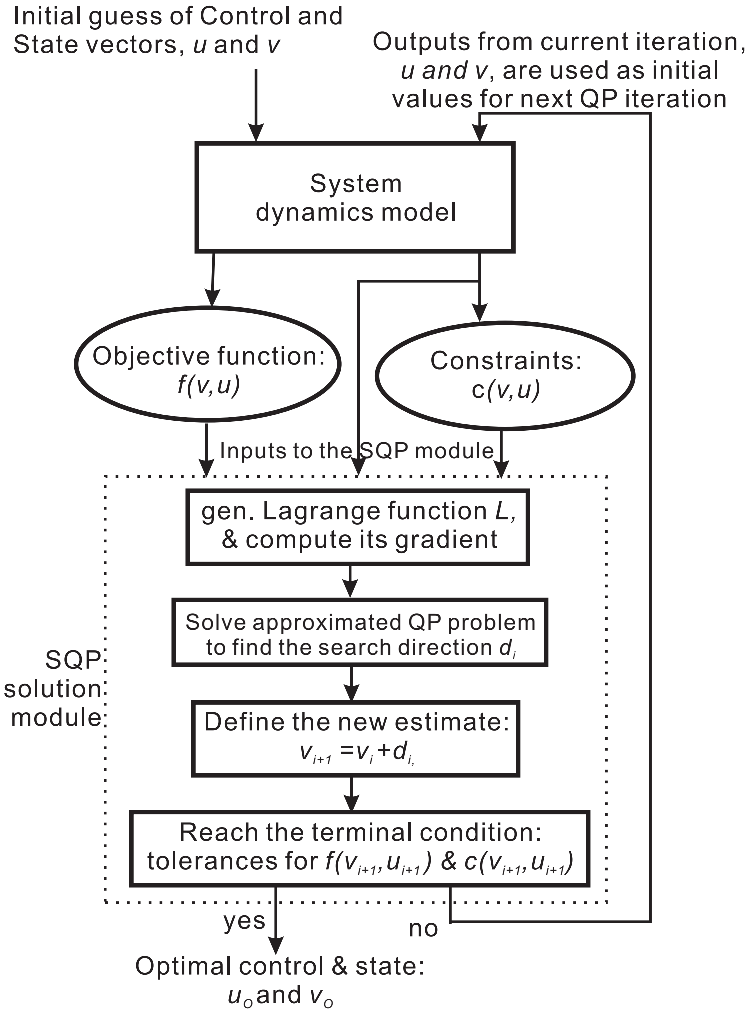

Initial guess of Control and State vectors, u and v
Outputs from current iteration, u and v, are used as initial values for next QP iteration
System dynamics model
Objective function: f(v,u)
Constraints: c(v,u)
Inputs to the SQP module
gen. Lagrange function L, & compute its gradient
Solve approximated QP problem to find the search direction d_i
SQP solution module
Define the new estimate: v_{i+1} =v_i+d_i,
Reach the terminal condition: tolerances for f(v_{i+1},u_{i+1}) & c(v_{i+1},u_{i+1})
yes
no
Optimal control & state: u_o and v_o

Figure 1.4: Complete SQP algorithm flow chart

where H is the approximated Hessian matrix of the Lagrange function, and the linear approximation of the equality constraint by its gradient $\nabla c(x)$ is:

$$c(x) \quad = \quad c(x_i) + \nabla c(x_i) \cdot (x - x_i) \tag{1.8}$$

b. Find the optimum search direction $d = x - x_i$, by solving the quadratic programming (QP) problem, and formulate a QP subproblem by Equation (1.7):

$$\min_d \frac{1}{2} d^T \cdot H_L \cdot d + \nabla f(x_i) \cdot d$$

$$\text{subject to}: c(x_i) + \nabla c(x_i) \cdot d \;=\; 0$$

c. Define the new estimate: the calculated optimum d_i can be used as a new search direction to perform a line search, and find a new estimate x_{i+1}.

d. Evaluate whether $f(x_{i+1})$ and $c(x_{i+1})$ have reached terminal conditions.

On the other hand, the interior point (IP) method provides an alternative method to solve the nonlinear NLP problem. The algorithm is essentially a barrier method in which the subproblems are solved by using a SQP iteration with trust regions. The SQP idea is used to efficiently handle nonlinearities in the constraints. The trust region method allows the algorithm to treat convex and non-convex problems uniformly, permitting the direct use of second derivative information [25], [26]. This method has been successfully applied to solve the trajectory optimization NLP problems in space and manned aircraft [27], but has only seen limited use in ground vehicle applications.

Mixed-discrete NLP solution

The throttle and gear shifting in this work are continuous and discrete control inputs, respectively. When some variables are discrete and others are continuous, the NLP problem is called a mixed discrete-continuous nonlinear programming (MDNLP), which has the following form [28]:

$$\begin{aligned}
\text{minimize}\;\; & f(x) && && (1.9)\\
\text{subject to}\;\; c_j(x) &= 0 \quad j = 1, \ldots, k\\
c_j(x) &\leq 0, \quad j = k+1, \ldots, m\\
x_i &\in D_i, \; D_i = (d_{i1}, d_{i2}, \ldots, d_{iq_i}), \quad i = 1, \ldots, n_d\\
x_{iL} &\leq x_i \leq x_{iU}, \quad i = n_d + 1, \ldots, n
\end{aligned}$$

where n_d is the number of discrete variables, D_i is the set of discrete values for the ith variable, q_i is the number of the discrete values for the ith variable, and d_{ik} is the kth discrete value for the ith variable.

In general, a MDNLP problem can be solved by optimization methods which can be classified into two groups: stochastic and deterministic methods. Some of the stochastic methods for global optimizations include the sequential random search and various evolutionary programming methods such as simulated annealing and genetic algorithms. Most of these stochastic methods do not need a priori knowledge about the objective function. These methods, however, usually require a large number of function evaluations, which makes these methods impractical to the real-time optimal control problem involving computationally intensive processes in this research.

When the objective and constraint functions are explicitly expressed, deterministic methods can be applied. These methods include the branch and bound, sequential linear programming, cutting plane techniques, outer approximation, Lagrange relaxation approaches, and so on. Another group of methods are called the rounding methods, which use simple dynamic rounding-off techniques. Most of these methods such as branch and bound either use the closely related NLP to reach to the main MDNLP, or rely on the successive solutions of the related mixed integer programming such as outer approximation [29].

The branch and bound is basically an implicit enumeration method in which one systematically tries to reduce the number of trials to reach the minimum point [28]. The penalty function approach is to treat discrete requirements as explicit constraints and construct an objective function penalizing deviations from discrete values. The Lagrangian relaxation method replaces the original MDNLP problem with a sequence of convex and separable approximate subproblems that are solved using Lagrangian relaxation and minimizing the dual function with a subgradient method [30]. For optimization problems with nonlinear constraints, linearly constrained Lagrangian methods solve a sequence of subproblems that minimize an augmented Lagrangian function subject to the linearization of constraints.

These deterministic approaches are based on explicitly expressed objective and constraint functions.

In [31], authors generated time-optimal velocity profiles for a group of path-constrained vehicles with fixed and known initial and goal locations. Each vehicle robot must follow a fixed path, arrive at its goal as quickly as possible, and stay in communication with other robots in the arena throughout its journey. Authors sought to solve this multi-objective optimization problem by generating optimal velocities along the paths. The problem was formulated as a NLP with constraints on the kinematics, dynamics, collision avoidance, and communication, where the discrete requirements were approximated as continuous value constraints by using sequential linear approximation.

Additionally, a lot of research has been conducted to find the extension of the well-known continuous nonlinear NLP solvers such as SQP and IP to solve the MDNLP problem. Authors in [32] presented a new trust region SQP method for the MDNLP problem, where convexity and relaxation are not required. However, the authors assumed that the model functions are smooth in the sense that an increment of an integer variable by one leads to a small change of function values. The author in [33] introduced an extension of the SQP method to solve the mixed-integer programming problem (MINLP). The general idea is to combine a SQP step with a direct search cycle in the integer space. Hessian information is updated based on difference formulae at neighbored grid points.

A simple approach for the MDNLP problems is to first obtain an optimum solution using a continuous approach, e.g., SQP or IP. Then using heuristics, the variables are rounded-up to the nearest available discrete values to obtain a discrete solution. However, it is not necessary to round-up all variables to their nearest discrete neighbours. Some of them could be decreased while others could be increased. The main concern of a rounding-off approach is the selection of variables to be increased and the variables to be decreased [28]. This is a simple idea, but it often results in an infeasible design for problems having a large number of variables.

For the MDNLP problem at hand, this research couples a direct collocation method with a deterministic discrete variable search strategy with the following steps:

1. Apply the direction collocation method to transform the infinite dimensional dynamic optimization problem into a finite dimension (MDNLP) problem, where all design variables are treated as continuous ones;
2. Apply any standard nonlinear programming code to find an optimum in the design space of continuous design variables;
3. Find the discrete design variables by using the deterministic discrete search methods, e.g., penalty function, rounding-off or Lagrangian relaxation, at the final step.

1.2.3 Real-time optimal control system design

In this work, a real-time optimal control system is developed to command the fuel optimal truck behavior online by using the optimizer as described in Section 1.2.2. Optimal control policies are classified as either open-loop or closed-loop optimal controls. An open-loop policy determines the optimal time program for the inputs to a dynamic system corresponding to a particular initial condition. On the other hand, a closed-loop policy is a stronger form in which the optimal feedback of a dynamic system's states for any initial condition is determined [9]. This work will focus on closed-loop policies because

- A closed-loop solution is more stable than the open-loop solution;
- Current NLP solvers are capable of getting fast convergence and generating reliable numerical solutions for large-scale NLP problem.

The dominant open-loop design approach in the literature is some form of rule-based design, relying on engineering intuition and logic. The heuristically derived rule-based optimal fuel controller was developed in [6] by Scania Inc, which is called Expert Cruise Controller

(ECC) and uses look ahead road information. The ECC implements different control strategies for different road section types, i.e., crest or sag slope. But the ECC switches different control strategies completely based on the road-map and therefore is highly sensitive to road-map inaccuracy. On the other hand, optimal control strategies, e.g., DP, has been used to generate the optimal sequence of control inputs to improve hybrid vehicle fuel economy, which are then studied to derive implementable control rules in [34], [35].

A typical optimal closed-loop feedback control is the model predictive control (MPC) [36] and the rolling horizon concept. Its main idea is to solve on-line a finite-horizon open-loop optimal control problem considering the current state as the initial state for the problem as indicated in Figure 1.5. The problem is formulated and solved at each discrete-time instance with the following steps:

1. Obtain measurements of the states of the system;

2. Compute an optimal input signal by minimizing a given cost function over a certain prediction horizon in the future using a model of the system;

3. Implement the first part of the optimal input signal until new measurements/estimates of the state are available. Then, repeat with step 1.

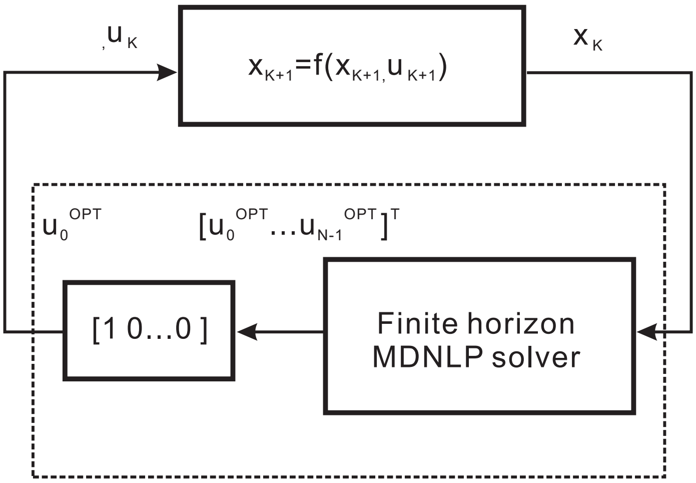


Figure 1.5: Basic model predictive control loop [36]

Although nonlinear MPC has become a well-established control approach, its application to time-critical systems requiring fast feedback is still a major computational challenge. In [7], the MPC scheme was applied as a Model Predictive Cruise Control system (MPCC) in heavy trucks to reduce the truck fuel consumption and travel time by using road topography. In [37], authors investigated a new multi-level iteration scheme, and extended the idea to real-time iterations. This novel approach took into account the natural hierarchy of different time scales inherent in the dynamic model. The authors applied the investigated multi-level iteration scheme to various vehicle optimal control applications and discussed the computational performance of the scheme.

Normally, MPC is possible if the plant has sufficiently slow dynamics. If a plant has fast dynamics, then the usual notions of explicit feedback theory are necessary for successful implementation. In the system in this research, the slow and fast dynamics are separated by the optimal speed trajectory generation and truck speed control, respectively. Trajectory generation is the slow outer-loop while control is the fast inner-loop. The concepts of inner and outer loops are formalized by a two degree-of-freedom (DOF) control system architecture where the inner loop is used for stabilizing a nominal reference trajectory generated by the outer loop [38]. Figure 1.6 demonstrates an inner-outer loop structure for achieving optimal feedback control. Under this structure traditional linear or nonlinear control theory can be used to design the inner loop while optimal control techniques are used for the outer loop [39].

By applying this two DOF optimal control design concept, a two-level adaptive cruise control (ACC) synthesis method was presented in [40]. In the outer loop, an indirect optimization method is used to solve a TPBVP problem and then find the desired vehicle acceleration with the cost function consisting of a penalty for vehicle range and range rate errors. In the inner loop, an adaptive control algorithm is designed to ensure the vehicle follows the outer loop acceleration command accurately.

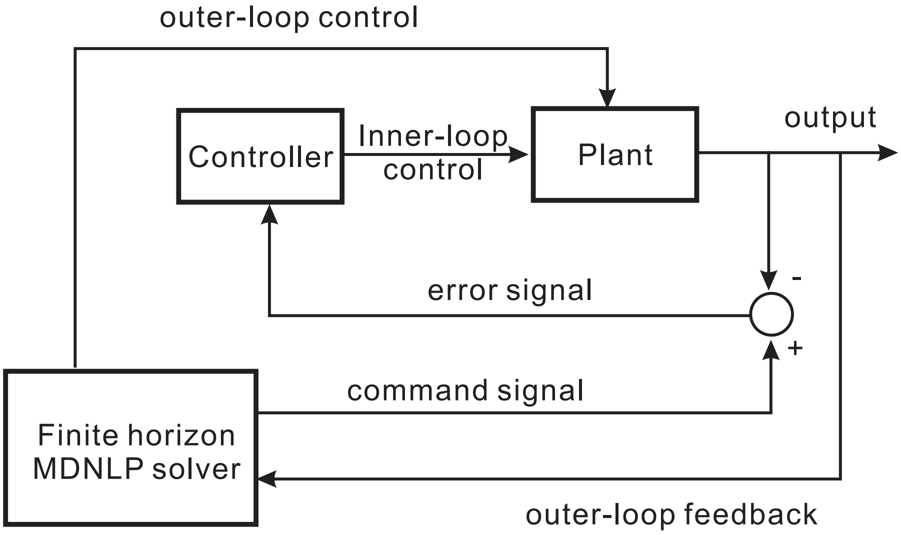


Figure 1.6: Real-time optimal feedback control through an inner-outer loop structure [39]

Additional two DOF optimal control approaches applied to various real-time vehicle maneuvers, e.g., minimum time double lange change at high speed, were studied in [41]. In their work, the authors developed a two-level driver model. On the anticipation level, optimal control problems for a reduced vehicle dynamics model are solved repeatedly on a moving prediction horizon to yield near optimal setpoint trajectories for the full model. On the stabilization level, a nonlinear position controller is developed to accurately track the setpoint trajectories with a full motor vehicle dynamics model in real-time. In this work, the optimal control problem is transformed to a NLP problem by a sparse direct collocation method and then solved by a gradient based SQP solver.

In order to build a stable optimal feedback control system, a two DOF optimal control approach is applied in the fuel-optimal control system in this research with the scheme as depicted in Figure 1.6:

1. In the outer loop, the optimal control problem is solved using the MDNLP solver repeatedly on a moving prediction horizon to yield near optimal setpoint trajectories;

2. In the inner loop, a feedback controller is developed to accurately track the optimal trajectories over some period;

3. The procedure is repeated over the next prediction horizon.

1.3 Contributions

The contributions of this work are mainly in the following three aspects:

- A mixed discrete-continuous nonlinear programming (MDNLP) solver was designed and experimentally validated to solve the fuel-optimal throttle, brake torque engine retarder (generated from the engine retarder), velocity trajectory, and especially the gear shifting. The optimal gear-shifting control has been widely studied to improve the performance of the automatic transmission with respect to passengers comfort, gear shift duration [42], and fuel efficiency [35]. However, the road map based fuel optimal gear-shifting control has only been considered in [12], but has not been implemented and validated in any previous road tests.

- A complete system scheme was built up to integrate the real-time optimal control and map-matching system, and utilize commercial GIS road geometries that are accurate nationwide.

- A sensitivity analysis was conducted by applying real road geometries to investigate how the change of the terrain and the errors in the terrain data effect the gain in fuel economy and the system performance.

1.4 Outline

In Chapter 2, road geometry, simulated road geometry, drive cycle, and fuel consumption baseline are described. Chapter 3 details the heavy truck system modeling and the relation between terrain variation and truck fuel consumption. The real-time optimal control system design and implementation are described in Chapter 4. Chapter 5 shows simulation and real road test results as well as their analyses. Work for sensitivity analysis is introduced in Chapter 6. Finally, a summary of the work and the recommendation for future work are listed in Chapter 7.

Chapter 2
Road Geometry and Baseline

Since road geometry is analyzed to obtain the best truck operations and maximal fuel performance, the basic definitions for highway design are introduced in this chapter. Based on these definitions, a simulated highway is generated to assist the analysis of truck fuel consumption on different terrains before real road geometries are considered. The fuel consumption baseline and drive cycle are defined to evaluate the designed real-time optimal control (OC) system performance.

2.1 Road Geometry

In this section, the basic road definitions are provided and then a set of road profiles are simulated. The simulated road profiles should be close to real road situations based on some important road design parameters. Additionally, the calculation of real road grade and the analysis of terrain type are given.

Based on [43] and suggestions from Intermap, the road profiles are classified into three types: level, rolling, and mountainous using the following parameters:

- Degree of grade: the grade of the road slope;
- Number of transitions: number of road slopes;
- Frequency components: the frequency of the change of slopes.

2.1.1 Basic road definitions

To simplify the design, some road and vehicle conditions are fixed in this work, based on [43]:

- Road type: rural freeways (arterial highways), which are designed for large volumes of traffic at high speeds;
- Truck weight/power ratio: 120 kg/kW, which represents the size and type of vehicle normally used for design control for main highways;
- Units: Metric, which can be changed to US customary upon request.

By further referencing [43], some important definitions for the road design are:

1. Design speed: a selected speed used to determine the various geometric design features of the highway, which is set to 90, 110 or 120 km/h, in this project. The OC system has a 90 km/h reference speed;
2. Degree of road grade: vertical elevation over longitudinal length of a road slope, G = D/L, in percentage, as shown in Figure 2.1.

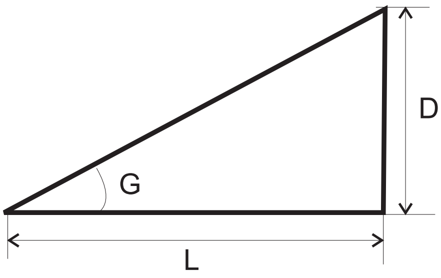


Figure 2.1: Road grade

3. Vertical alignments: a vertical curve is used to provide a smooth transition between vertical tangents of different slope rates. Vertical (parabolic) curves consist of symmetric crest and sag, shown in Figure 2.2, and are constrained by:
    - Minimum length of a vertical tangent (L_t): the minimum length of a short vertical tangent between two consecutive vertical curves is $L_t = 0.3v$ in meters, where v is design speed in km/h.

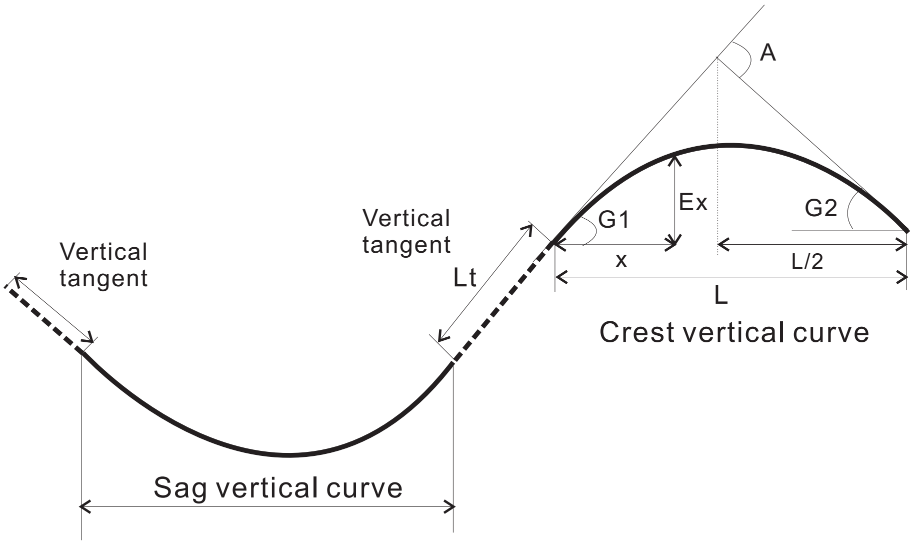


Figure 2.2: Vertical curves and tangents

- Minimum length of a vertical curve (L): this is a major control parameter and related to safety sight distance. It can be calculated by:

$$
\begin{aligned}
L &= KA, \quad A = |G_1 - G_2|, \quad K = \begin{cases} 39, & v = 90 \\ 74, & v = 110 \\ 95, & v = 120 \end{cases} \\
L &= 0.6v, \quad \text{if A} < 1 \\
E_x &= \frac{(G_2 - G_1)x^2}{2L} + G_1 x, \quad E_x : \text{elevation of curve}
\end{aligned}
$$

4. Terrain classifications:

- Level: highway sight distances are generally long and not blocked by vertical restrictions;
- Rolling: highway slopes gently rise and fall, and there are occasional steep slopes;
- Mountainous: longitudinal changes in the elevation of the ground with respect to the road are abrupt.

In general, rolling terrain generates steeper grades than level terrain, causing trucks to reduce speeds; and mountainous terrain has even greater effects, causing some trucks to operate at crawl speeds. For a rural highway, the maximum road grade is related to the design speed and terrain type, as shown in Table 2.1. The minimum road grade is around 0.5%, for the surface drainage design.

Table 2.1: Maximum road grade relative to design speed and terrain type

Terrain type	Design speed (km/h)		
	90	110	120
Level (grade %)	4	3	3
Rolling	5	4	4
Mountainous	6	5	N/A

5. Frequency related parameters which are in a certain highway section include:
   - Frequency component: the frequency of the change of the slope. The frequency can be low, medium or high;
   - Number of transitions: the approximate number of slope transitions can be calculated, if the frequency component is given and the minimum length of each vertical curve is known;
   - Coupling between the slope transitions: how far two slopes are apart. This distance can be calculated, if both the frequency component and the number of transitions are known.
6. Elevation of the entrance and exit: comparing with the entrance, the elevation of the exit is higher or lower.

2.1.2 Parameter based highway simulation

The parameters defined in Section 2.1.1 are used to generate a simulated highway section. First step, the user should choose some values or conditions of parameters, in order to describe the special features of a highway profile, which is shown in Table 2.2. For the purpose of simplicity, the highway length and design speed are only given three fixed values, respectively.

Table 2.2: Fixed parameters for the parameter based highway simulation

Input Parameter	Pre-defined condition		
Highway length(km)	10	50	100
Design speed (km/h)	90	100	120
Terrain type	Level	Rolling	Mountainous
Exit vs. Entrance	Lower	-	Higher
Frequency component	Low	Medium	High

In this research, six highway profiles with a 50 km length were generated. The associated parameters are shown in Table 2.3. For example, Simulation 1 has a design speed of 110 km/h; terrain is rolling; and exit is lower than entrance and the frequency component is low.

Table 2.3: Parameter based highway simulation examples

Simulation	Length	Speed	Terrain	Elevation	Frequency	Figure
1	50	110	R	L	L	Figure 2.3
2	50	110	R	L	M	Figure 2.3
3	50	110	R	L	H	Figure 2.3
4	50	90	L	H	M	Figure 2.4
5	50	90	R	H	M	Figure 2.4
6	50	90	M	H	M	Figure 2.4

The diagrammatic comparison of simulations 1, 2 and 3 is shown in Figure 2.3. In can be seen from Table 2.3, the only difference among them is the frequency component, which is low, medium, and high for three cases, respectively. As the diagram shows, the high frequency road (gray curve) has faster slope change than the medium frequency road (dashed), which in turn has faster change than the low frequency road (black).

Simulation 4, 5, and 6 are compared together in Figure 2.4 to show the difference in terrain type. It is clear that the level road (dark curve) has the smallest road grades compared

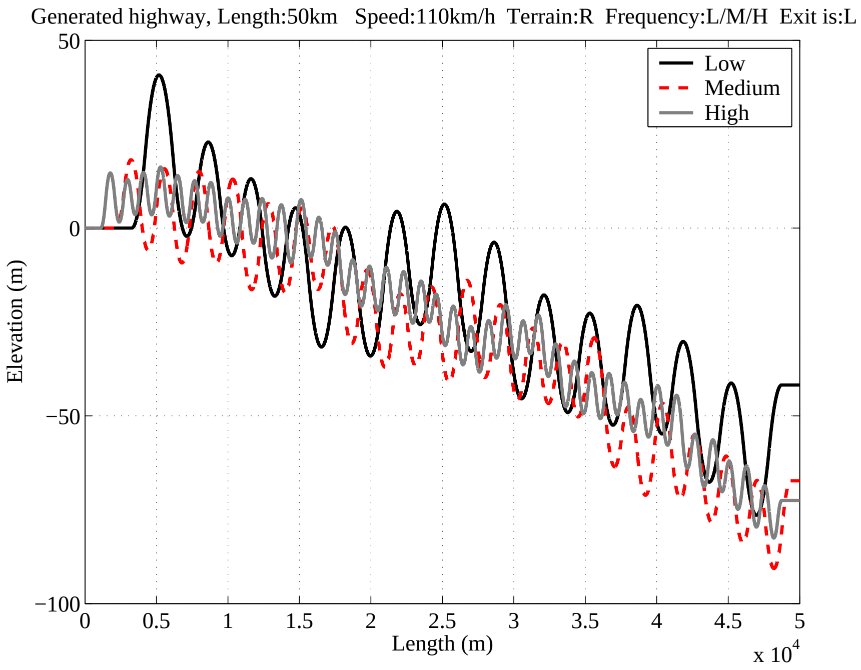


Figure 2.3: Generated highway, comparison of simulations 1/2/3

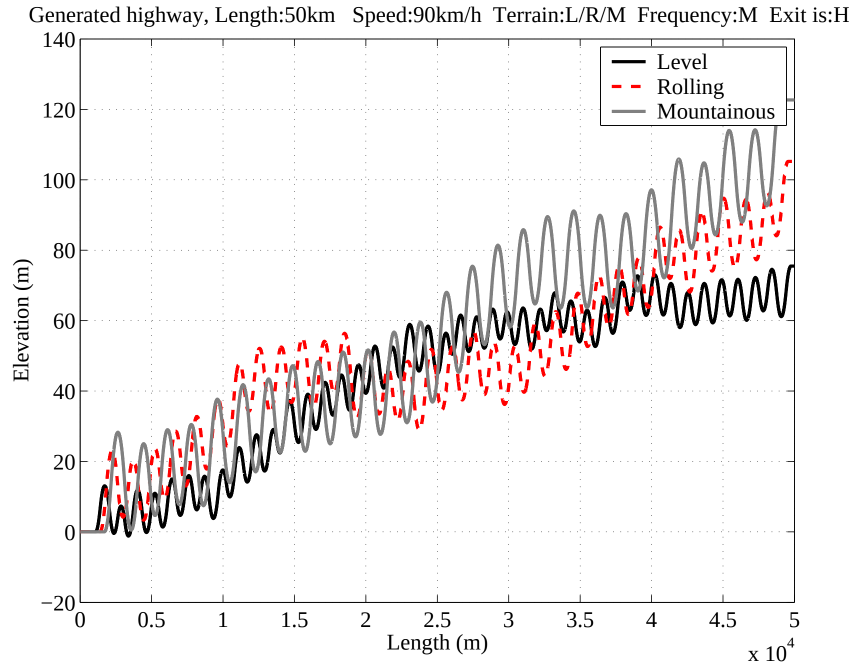


Figure 2.4: Generated highway, comparison of simulations 4/5/6

with the rolling road (dashed) and mountainous road (gray), which has the steepest grades. In Table 2.2, the maximum grade for these three simulations is 4%, 5%, and 6%, respectively.

2.1.3 Calculation of real road grade

The simulated road geometries are generated to quantify the fuel consumption change on different terrain types. While in the OC system implementation, the real road geometry is required. The real road geometry applied is a 3D road vector, with accurate longitudinal (x), lateral (y) positions, and elevation (z). The road slop can be obtained from the road vector by calculating the point-to-point slope (both in x and y directions) at each point and then passing a second order polynomial filter with a L meter window over the result. The filter helps to remove the high frequency component from the instantaneous slope, leaving a smooth profile. For a simple representation, all road profiles in this work are drawn only in the x and z frame. The slope is multiplied by 100 to obtain a percentage slope value ϕ at sampling point k as:

$$\phi(k+1) = 100\frac{z(k+1)-z(k)}{\sqrt{(x(k+1)-x(k))^2+(y(k+1)-y(k))^2}} \tag{2.1}$$

The original slope value is passed to a Savitzky-Golay smoothing filter [44]. The coefficients of the filter are obtained by applying a least squares adjustment. This adjustment is used to identify a second order polynomial function which minimizes the sum of the square of the residuals between the function and the original slope over a L meter window. The value of the function at the point of interest is then selected as the filter output at that point. The next point is then moved and the process is repeated.

2.1.4 Real road geometries and terrain type analysis

Intermap's 3D road geometries acquired in California and West Virginia are applied in this work. These consist of four test routes in California as shown in Figure 2.5, and one

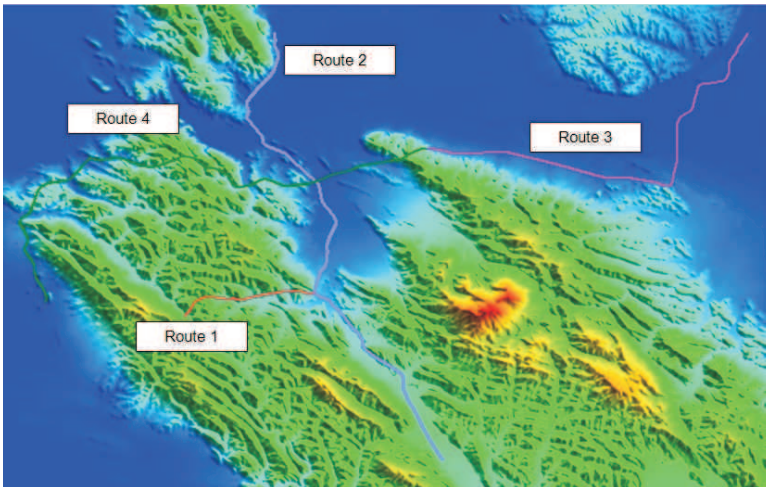


Figure 2.5: Overview of Intermap's road geometries in California

in West Virginia. In order to test the OC system on different terrain types, routes number 2 and 3 in CA, and the route in WV are used in this research. They are level, rolling, and mountainous terrains, and named as routes R1, R3, and R5, respectively. It is also interested in the reverse direction of routes R1, R3, and R5, which are named as R2, R4, and R6.

The details of these six road profiles are provided in Table 2.4, where the mean, maximum, and minimum road slopes are listed as a percentage. The σ in this table represents the standard deviation of the road slope value.

Table 2.4: Intermap road profiles analysis

		slope(%)				
route	length (m)	mean	max	min	σ	terrain type
R1	37000	-0.21	2.65	-4.33	1.05	level
R2	37000	0.21	4.33	-2.65	1.05	level
R3	47000	0.27	4.88	-3.87	1.34	rolling
R4	47000	-0.27	3.87	-4.88	1.34	rolling
R5	54000	0.21	5.57	-5.43	3.05	mountainous
R6	54000	-0.21	5.43	-5.57	3.05	mountainous

The road maps for R1 and R2 are provided in Figure 2.6, where the road elevation and grade maps are given for R1 and R2. It can be seen from the figure that road grades for R1 and R2 are mostly within the grade range of [-4% 4%], and the reference design speed is 90 km/h. Thus, by comparing to the terrain classification as described in Table 2.1, it is clear that the R1 and R2 are both level terrains.

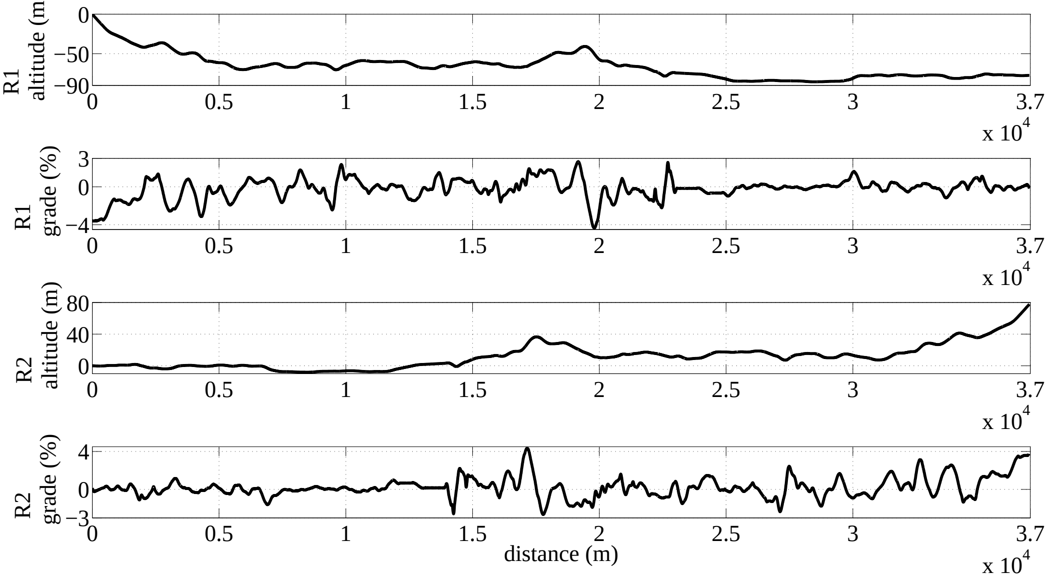


Figure 2.6: Road elevation and grade maps for R1 and R2

The road elevation and grade maps for R3 and R4 are depicted in Figure 2.7. It is seen that road grades for R3 and R4 are mostly within the range of [-5% 5%]. Therefore, R3 and R4 can be classified as the rolling terrain with reference to Table 2.1. Additionally, the maps for R5 and R6 are shown in Figure 2.8. R5 and R6, which are within [-6% 6%] grade range, have far larger road grades than R1 to R4. Thus, R5 and R6 are mountainous terrains. In this section, the simulated roads R1 - R6 are not generated and then compared with the corresponding real road geometries, since this is out of the scope of this research.

2.2 Driver Cycle and Baseline

In this section, the fuel consumption baseline for the specific road section and drive cycle are defined to evaluate the performance of the designed OC system. A drive cycle

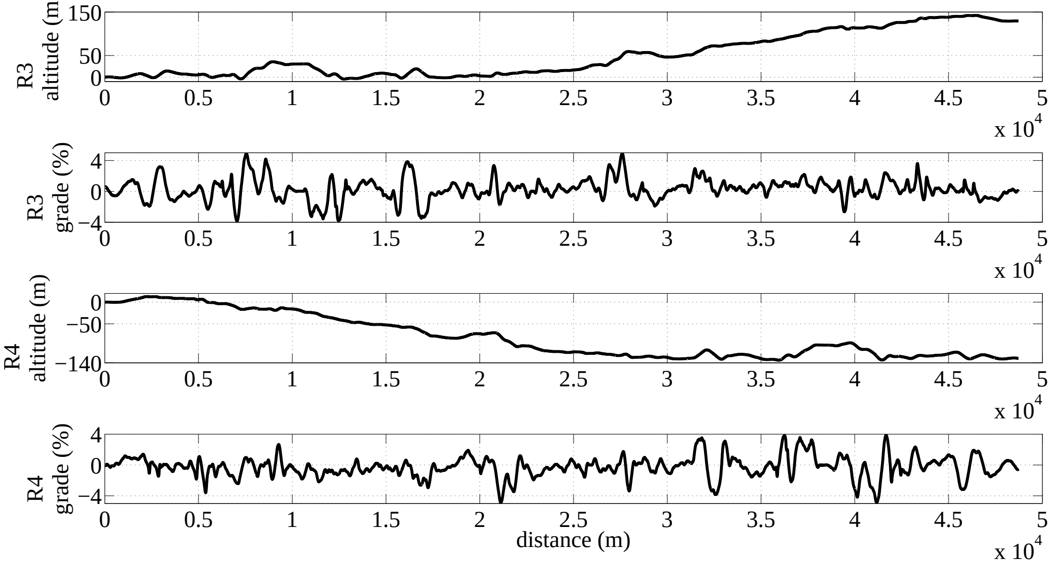


Figure 2.7: Road elevation and grade maps for R3 and R4

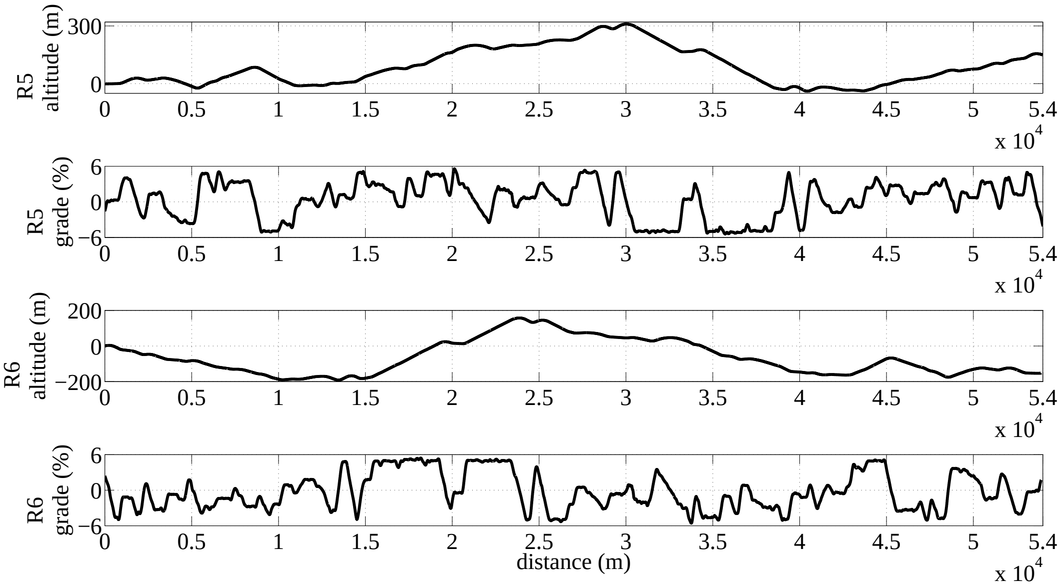


Figure 2.8: Road elevation and grade maps for R5 and R6

constitutes a series of vehicle speeds as a function of time on a specific road section. For fuel economy research, the definition and selection of the baseline, which is fuel consumption for a normal drive cycle, is critically important. The main current practices in drive cycle development are segment-splicing, Monte Carlo simulation, and “engineering” approaches. The first approach is based on real-driving data, and the second approach is simulated from a realistic driving behavior model. The third approach, on the other hand, is defined by a designer to implement some truck feasibility testing [45].

If the drive cycle developed by the first method is applied, the function of the OC system could then be compared to the real truck driving condition, or even better, experienced drivers’ behavior, and therefore the gain in the fuel economy is more meaningful. However, this driver cycle can hardly be developed in this work since it requires a large amount of driving data collection on the test routes R1 to R6, which is out of the scope of the database of this research.

Instead, the drive cycle is defined by the third method as a constant speed to implement some truck feasibility testing. The fuel consumption and travel time baseline is then calculated from a standard cruise controller (CC) from the experimental truck to perform this drive cycle. The CC is applied to maintain the desired vehicle velocity by changing throttle position. Additionally, a function of brake pedal control is engaged only if the truck reaches an upper speed limit. The gear selection is determined based on the engine speed and current gear position.

This baseline is generic because the CC is highly engaged in the real-time truck driving. Moreover, by integrating the automatic braking control in the CC, the baseline has the feature close to the real drivers’ behavior, that is keeping the cruise speed and braking only if the speed limit is reached. Thus, with the application of this baseline controller, the function of the OC system can be compared to some near-real truck driving conditions and consequently, resulting in a realistic and meaningful gain in the fuel economy.

2.3 Conclusion

This chapter described real and simulated road geometries and the comparison baseline that will be used to evaluate the optimal control system (OC) performance. First, some important road design parameters, e.g., road grade, terrain type, and frequency component, were defined, from [43]. The special values or conditions of these parameters were chosen to describe main features of a road profile. Subsequently, simulations were conducted in Matlab to generate various road profiles. Simulation results were finally provided and analyzed to show that the applied road generation method is correct. More importantly, by using the simulated road profiles, the relation between the fuel consumption and road geometry could be analyzed. The features of real road geometries were also introduced in this chapter, which include the calculation of road grade from 3D elevation data and the analysis of terrain type. Additionally, the fuel consumption baseline and drive cycle were defined to evaluate the designed OC performance.

Chapter 3

Heavy Truck Model and Terrain Based Fuel Consumption

A heavy truck model needed for the design of a model-based OC system is developed and evaluated in this chapter. In addition, simulations are run by using the model to quantify how a change in terrain type effects the truck fuel efficiency.

3.1 Heavy Truck Model

A Class 8 truck longitudinal model is described and the important truck parameters are listed in Appendix A. The model includes the engine, driveline, wheel, and truck dynamics. A tire model is not used, and therefore a no-slip condition is assumed, which is discussed in Section 3.1.5. This model is a hybrid system, a system with both continuous and discrete parts. The position and velocity are continuous states. Throttle position and brake are continuous inputs, and gear shifting is a discrete control input. In this work, the hybrid system is approximated by a continuous system.

3.1.1 Longitudinal dynamics

Longitudinal vehicle dynamics typically include many losses such as rolling resistance, air drag, and road grade or slope. The developed model has one degree of freedom and is derived using the equation of motion for the free body diagram shown in Figure 3.1. To describe the longitudinal motion of a vehicle, the dynamics are derived from the loads on the vehicle, and has one degree of freedom:

$$m\frac{dv}{dt} = F_w - F_s - F_{rr} - F_a \tag{3.1}$$

where F_w is wheel drive force, F_s is longitudinal force due to road grade, F_{rr} is rolling resistance force, and F_a is air drag force:

$$\begin{aligned} F_s &= mg\sin\phi \\ F_{rr} &= C_{rr}mg \\ F_a &= \frac{1}{2}\rho_{air}C_dA_{fr}v^2 \end{aligned}$$

where ϕ is road grade, C_{rr} is rolling resistance coefficient, ρ_{air} is air density, C_d is aerodynamic drag coefficient, A_{fr} is truck frontal area, and v is truck velocity.

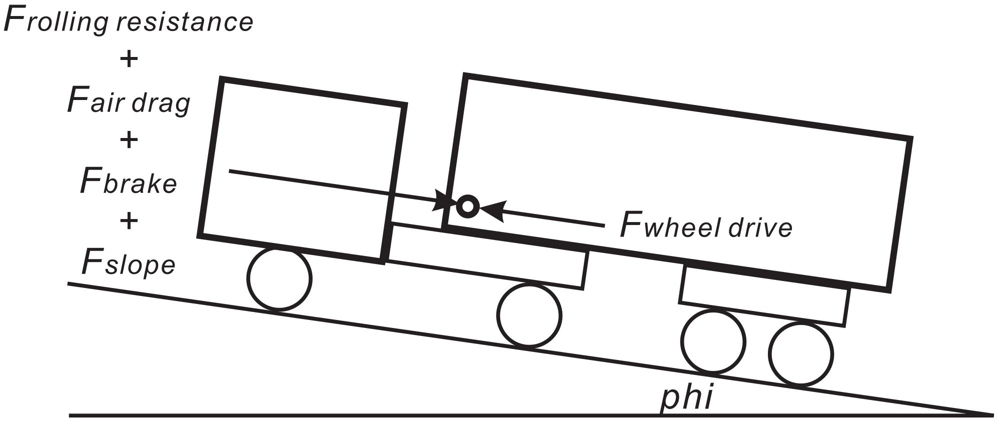


Figure 3.1: Longitudinal Free Body Diagram (FBD)

3.1.2 Engine map

The engine model is designed based on a rectangular engine map and shows a steady-state relation between the current engine speed and the maximum engine torque. The engine is normally operating in the range $\omega_e \subset [1100,\ 1800]$ rpm, and the maximum engine torque can be approximated as a function of the engine speed, ω_e. The comparison of real and approximated engine map is shown in Figure 3.2. With a normalized throttle position u, the desired engine torque can be calculated from:

$$\begin{aligned} T_m &= c_1\omega_e^3 + c_2\omega_e^2 + c_3\omega_e + c_4 \\ T_e &= T_m u \end{aligned} \tag{3.2}$$

where T_m and T_e are maximum and desired torque, and c_1 to c_4 are coefficients calculated from an engine map curve fitting.

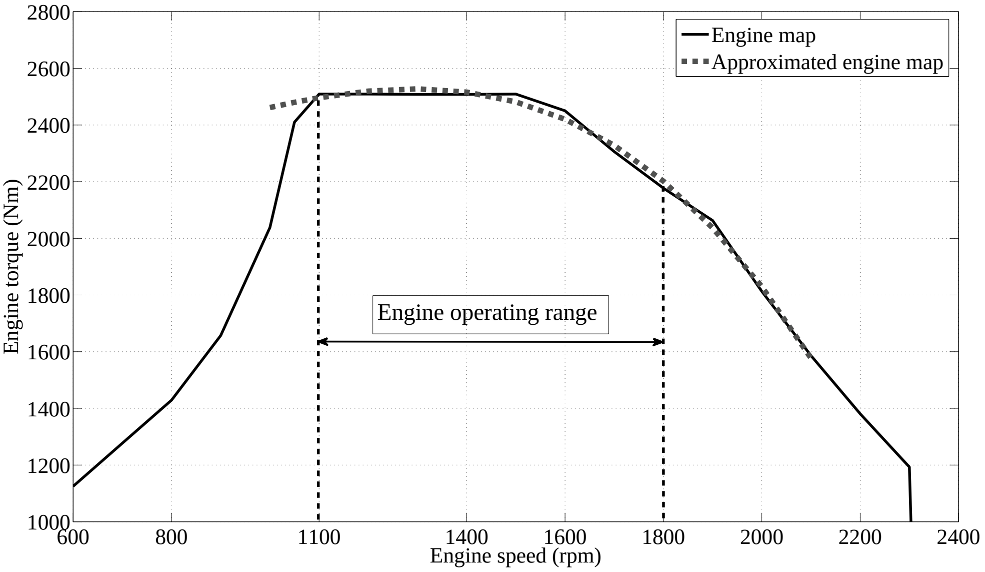


Figure 3.2: Comparison of real and approximated engine map

3.1.3 Engine efficiency and fuel consumption map

A Brake Specific Fuel Consumption (BSFC) map is a measure of fuel efficiency within an internal combustion engine. The BSFC number, B_n, is the rate of fuel consumption divided by the produced power and has the unit of g/kW-h. To calculate the engine efficiency, the lower heating value LHV of the fuel is used, which is defined as the amount of heat released by combusting a specified quantity of the fuel and returning the temperature of the combustion products to 150 °C. The LHV value for diesel fuel is 0.0119531 kW-h/g provided by Eaton. Therefore, the relationship between BSFC number and the engine efficiency of the applied diesel engine can be represented as follows:

$$\eta_e = \frac{1}{0.0119531B_n} \tag{3.3}$$

However, in order to make the calculation more straightforward, the truck fuel consumption in this work is calculated depending on a modified BSFC map as shown in Figure 3.3, which directly gives the fuel consumption time rate $\frac{dm_f}{dt}$ (g/sec). This map is derived from the original BSFC map by using the relation as in Equation (3.4).

$$\frac{dm_f}{dt} = \frac{B_nP}{3600} \tag{3.4}$$

where B_n is the original BSFC number and P is the engine power. Additionally, by substituting B_n from Equation (3.4) into Equation (3.3), the engine efficiency is rewritten as:

$$\eta_e = \frac{dt}{dm_f} \cdot \frac{P}{3600} \cdot \frac{1}{0.0119531} \tag{3.5}$$

This equation gives a direct relation between the engine efficiency and the fuel consumption time rate $\frac{dm_f}{dt}$ (g/sec). Actually, the task of the truck fuel minimization is equal to a work of the engine efficiency maximization. However, in the rest of this work the former instead of the latter is targeted and presented.

By using a 3D polynomial least-square fitting [46], the fuel consumption time rate $\frac{dm_f}{dt}$ (g/sec) can be approximated as a continuous function of engine speed and power P, shown in Equation (3.6).

$$\frac{dm_f}{dt} = b_1\omega_e^5P + b_2\omega_e^4P + b_3\omega_e^3P + b_4\omega_e^2P + b_5\omega_eP + b_6P \tag{3.6}$$

where b_1 to b_6 are coefficients. Engine power P in kilowatt (kW) can be calculated from T_e and ω_e by using the relation:

$$P = \frac{2\pi T_e \omega_e}{60000}$$

The real and approximated BSFC maps are compared in Figure 3.3, and the comparison errors are shown in Figure 3.4, which are close to zero in the engine operating range. When the engine is idling, an 800 rpm idle speed and 3 kW engine output are used. If the truck is coasting at a different engine speed, a 3 kW load is used as well.

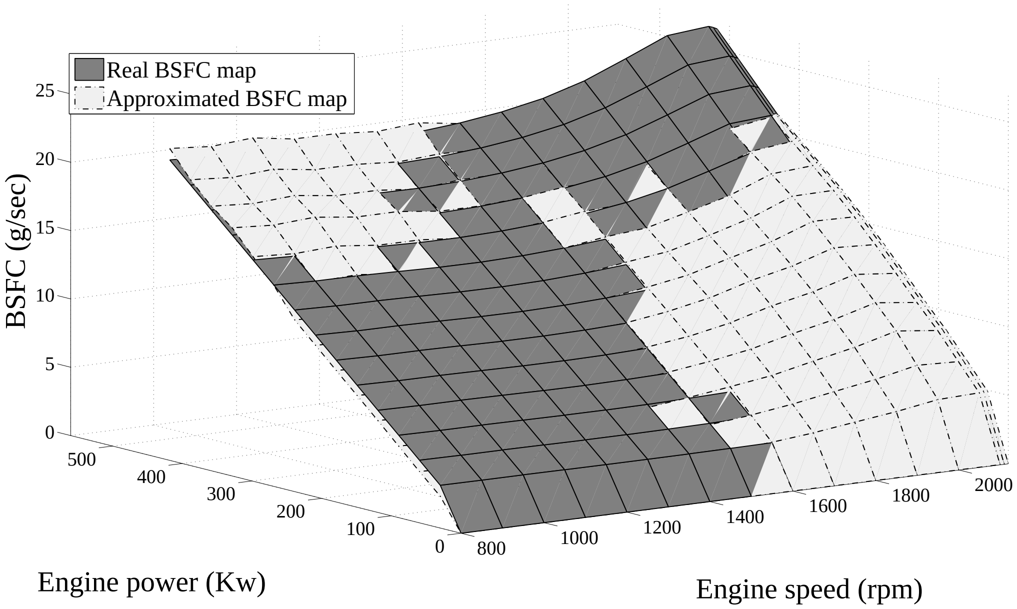


Figure 3.3: Comparison of real and approximated BSFC time rate map

3.1.4 Driveline

A complete block diagram of the truck model is shown in Figure 3.5, which consists of the engine, transmission, wheel, and external forces. The powertrain modeling assumes:

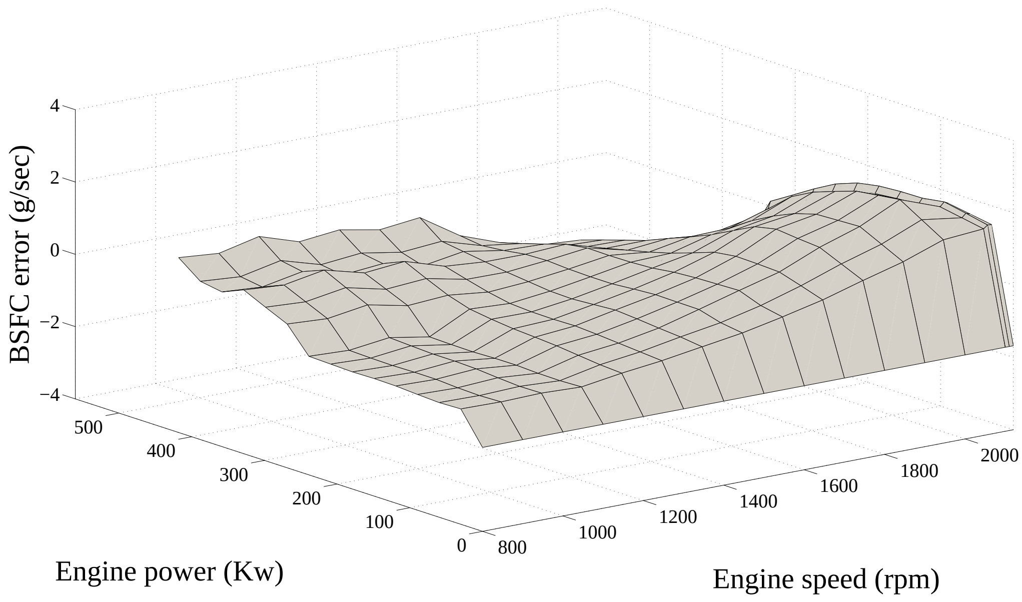


Figure 3.4: BSFC map approximation error: real minus approximated map

- Maximum engine torque is approximated from a rectangular engine map in Section 3.1.2;
- Driveline includes: stiff clutch, transmission, final drive, and wheel with no slip;
- Time delay on the gear shifting: one second without engine torque input.

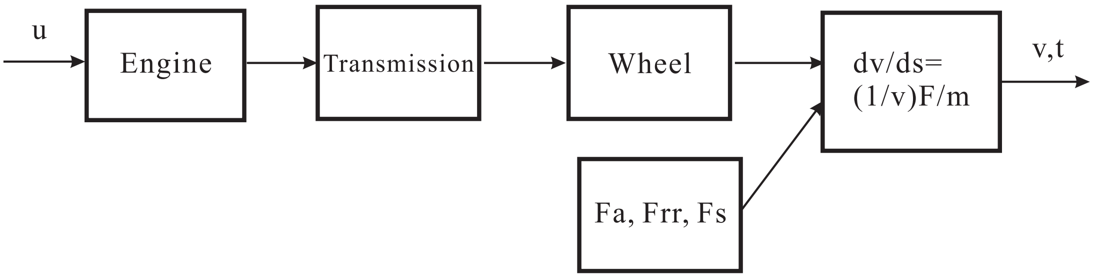


Figure 3.5: Longitudinal FBD and powertrain model

The driveline is assumed stiff and considered to transmit the power generated from the engine to clutch, transmission, propeller shaft, final drive, and finally to the wheel. The propeller shaft is assumed massless. The relation between the engine and clutch is:

$$9.55 J_e \dot{\omega}_e = T_e - T_c \tag{3.7}$$

where T_e is the desired engine torque as calculated in Equation (3.2), T_c is the load from clutch, T_w is the wheel torque, ω_e and ω_w are the engine and wheel rotation speed, and J_e and J_w are the engine and wheel inertia (kg-m^2). The factor of 9.55 in the equation is used to change the unit of ω_e from rpm to rad/s.

The clutch in this work is presumed stiff, and therefore: $T_c = T_t$ and $\omega_e = \omega_c$, where T_t and ω_c are the transmission torque and clutch rotation speed, respectively. Subsequently, the gear shifting point g in the transmission is modeled using two parameters, the transmission efficiency and ratio, η_t and n_t, with the relation:

$$T_t \eta_t n_t = T_p \tag{3.8}$$

$$\omega_c = n_t \omega_t \tag{3.9}$$

where T_p is the propeller shaft torque and ω_t is the transmission rotation speed.

Similar to the clutch, the propeller shaft is assumed stiff as well. This gives $T_p = T_f$ and $\omega_t = \omega_p$, where T_f and ω_p are the load from the final drive and the propeller shaft rotation speed, separately. The torque and rotation speed of the final drive are calculated from the propeller shaft with the relation:

$$T_f \eta_d n_d = T_w \tag{3.10}$$

$$\omega_p = n_d \omega_w \tag{3.11}$$

where η_t and n_t are the efficiency and ratio of final drive, and T_w is the wheel torque.

Finally, the torque and rotation speed on the wheel can be written as:

$$J_w \dot{\omega}_w = T_w - rF_w - T_b \tag{3.12}$$

where J_w is the wheel inertia (kg-m^2), F_w is the wheel drive force, and T_b is the desired brake torque generated from the engine retarder, which is determined by a normalized brake input $b \subset [0,\ 1]$ multiplied by the maximum engine retarder brake torque T_{bm} ($T_b = bT_{bm}$). In addition, the relationship between the engine speed and truck velocity is defined as:

$$\omega_e = \frac{30}{\pi} \frac{n_t n_d}{r} v \tag{3.13}$$

If the engine, transmission, final drive, and the wheel are considered together with the longitudinal forces, a complete truck longitudinal model is:

$$\frac{dv}{dt} = \frac{r}{J_w + mr^2 + \eta_d n_d^2 \eta_t n_t^2 J_e} \cdot (\eta_d n_d \eta_t n_t T_e - F_s r - F_{rr} r - F_a r - T_b) \tag{3.14}$$

As mentioned previously, gear shifting is a discrete control input. The discrete gear ratio, n_t, is approximated as continuous values, while in the OC operation the continuous ratios will be rounded off to find the nearest discrete gear ratio, which is the so called rounding-off method [47].

The road map is position dependent rather than time dependent, [33, 14]. Equations (3.6) and (3.14) are differentiated with respect to position rather than time by substituting: $dt = \frac{1}{v} dp$. This results in the fuel consumption position rate function and the complete truck longitudinal model:

$$\frac{dm_f}{dp} = \frac{1}{v} \cdot (b_1 \omega_e^5 P + b_2 \omega_e^4 P + b_3 \omega_e^3 P + b_4 \omega_e^2 P + b_5 \omega_e P + b_6 P) \tag{3.15}$$

$$\frac{dv}{dp} = \frac{1}{v} \cdot \frac{r}{J_w + mr^2 + \eta_d n_d^2 \eta_t n_t^2 J_e} \cdot (\eta_d n_d \eta_t n_t T_e - F_s r - F_{rr} r - F_a r - T_b) \tag{3.16}$$

where p represents the truck position, and P is the engine power.

3.1.5 Tire slip and gradeability

Although at the beginning of this section, it was stated that a tire model is not used and a no-slip condition (between the tire and the road) is assumed, it is still interested to investigate how this assumption effects the vehicle model and control system performance.

Since only the truck longitudinal dynamics is modeled, the form of slip which can have effect is the roll slip. Roll slip is defined as the relative motion between a tire and the surface on which it is moving. In general, the roll slip has a large effect on truck's gradeability, which represents the maximum slope a truck can climb while maintaining a particular speed before the roll slip occurs. This gradeability is affected by many factors including wheel radius, gross weight, and surface condition.

In this research, the gradeability data for the experimental truck to maintain a set speed of 25 m/s are provided by Eaton. Therefore, if the maximum slope doesn't appear frequently in the test route which is true for R1 - R6, the no-slip assumption will not heavily impact the accuracy of the truck model and the performance of the control system.

3.2 Model Validation

In this section, the developed truck dynamics and powertrain model are validated by comparing online test data from routes R5 and R6 and offline truck model simulation data. Before the OC system tests are conducted, a series of truck baseline runs were performed by using the CC to cruise on the reference speed of 25 m/s, and then test data were recorded. A feedback CC block diagram is given in Figure 3.6, where a PID control is used to calculate the desired engine torque to maintain the reference speed of 25 m/s.

The throttle input and gear shifting together with the calculated road slope collected from the baseline runs are used as inputs to the truck simulation model. The simulation outputs such as velocity, fuel rate, and total fuel consumption are then compared to the

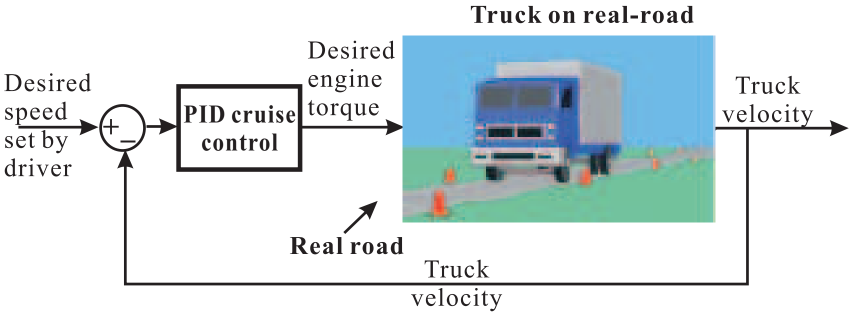


Figure 3.6: Block diagram for the cruise controlled truck (CC)

truck velocity and fuel consumption measured from baseline runs on R5 and R6. The basic validation system setup is shown in Figure 3.7, where the notations CC_torque, CC_gear, CC_speed, CC_fuel_rate, and CC_fuel_total are truck torque, gear shifting, speed, fuel rate, and total fuel consumption measured from the baseline runs on R5 and R6.

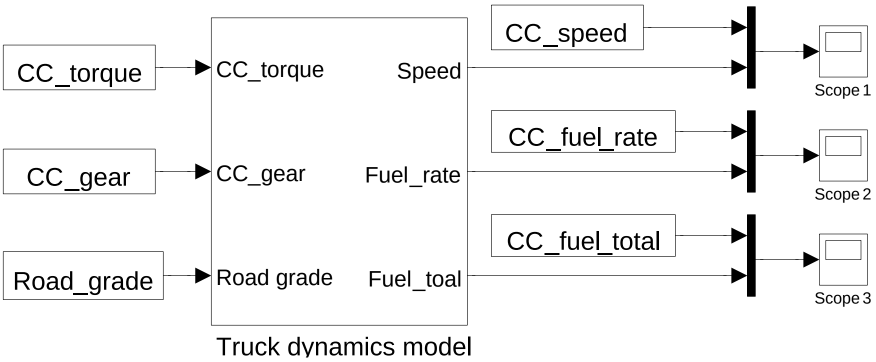


Figure 3.7: Block diagram for model validation

3.2.1 Using test data from R5

The measured data from the baseline test run (road grade, the CC torque, and truck gear) are shown in Figure 3.8. The comparison of the CC speed (dark) and simulation model output speed (dotted gray) is given in Figure 3.9. Additionally, the comparison of fuel rate (liter/s) and total fuel consumption (liter), for the CC and the simulation model is shown in Figure 3.10. It can be seen in Figure 3.9 that a speed error occurs at 100 - 150 s. It is because there is a small difference between the brake torque values calculated in the simulation and

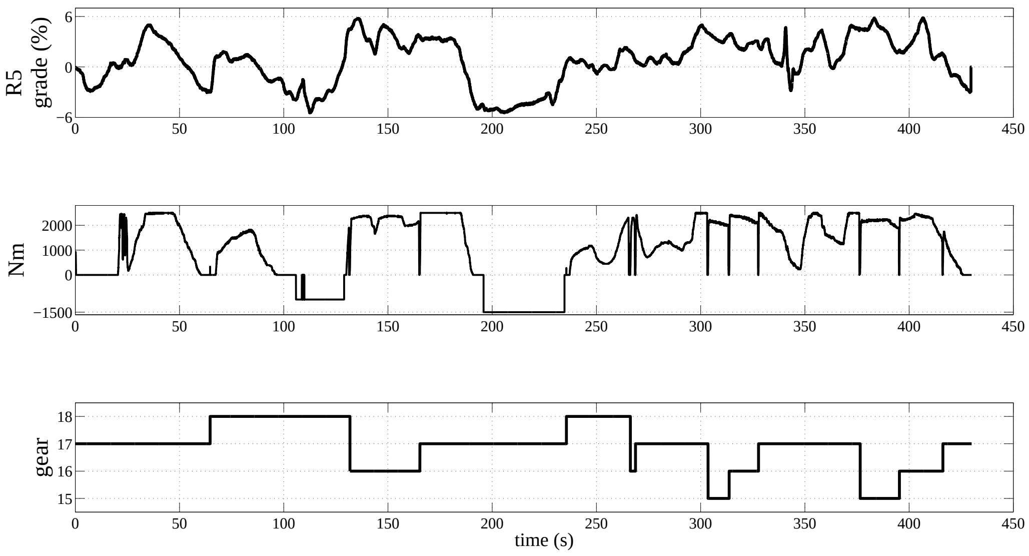


Figure 3.8: Measured data: road grade, CC torque, and gear, R5

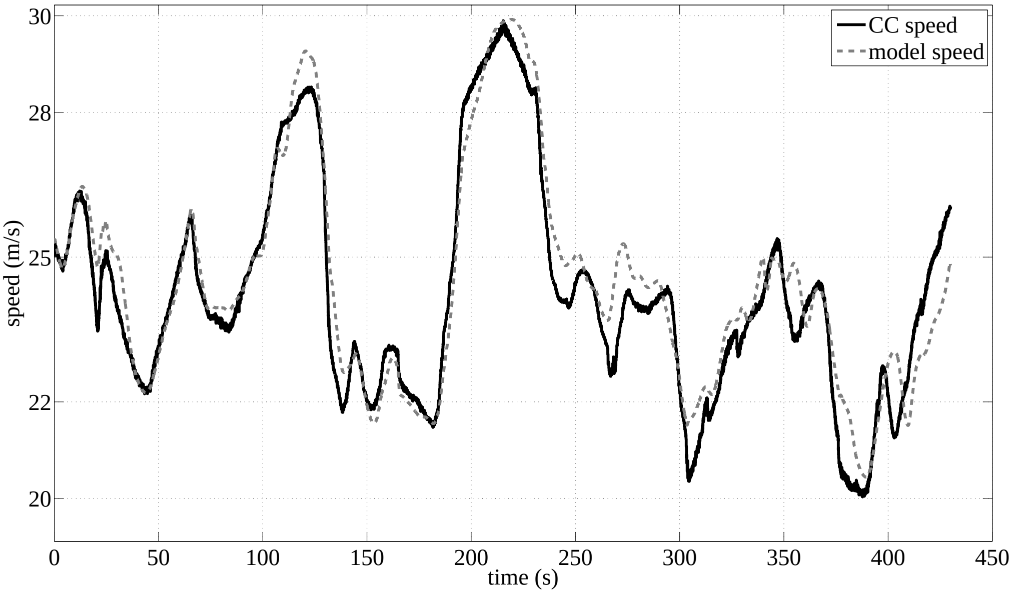


Figure 3.9: Comparison of CC and simulation model speeds, R5

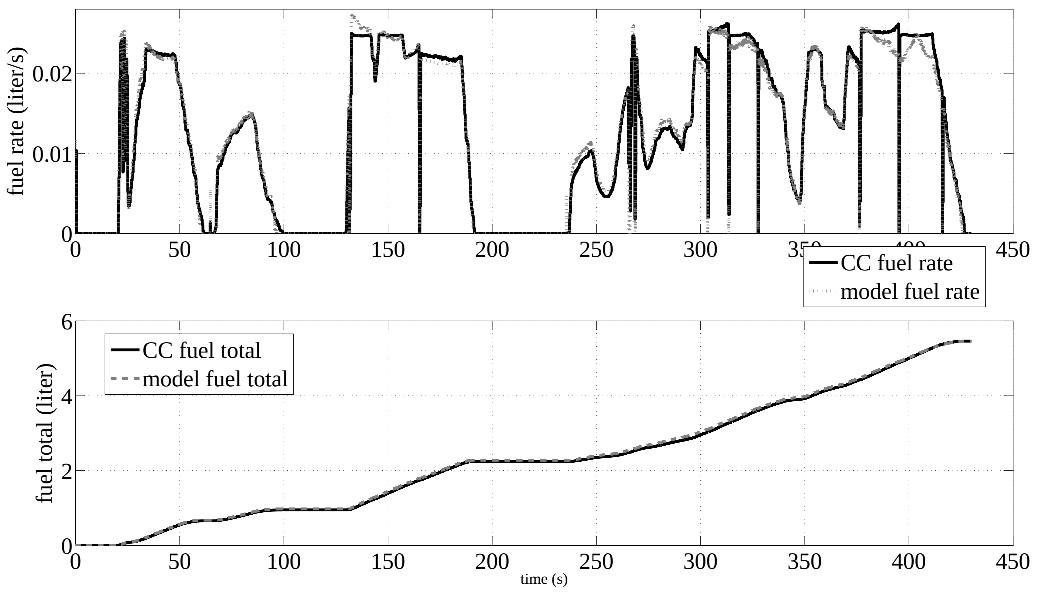


Figure 3.10: Comparison of CC and simulation model fuel use, R5

generated from the engine retarder in the experimental truck. Additionally, there are some small speed errors at 250 - 450 s, which are due to the modeling error of the gear shifting. As stated in Section 3.1.4, a one second time delay on the gear shifting is modeled, when there is no engine torque input. However, this is just an approximated value, and the real gear shifting time delay may vary based on different operation conditions. In this work, these modeling errors are acceptable and will not largely impact the system performance, since on the normal level and rolling highways braking and gear shifting are not frequently demanded. Meanwhile, it can be seen in Figure 3.10 that the simulation and CC fuel rate and total fuel consumption match each other quite well. Therefore, the developed truck dynamics and powertrain model is accurate enough to predict the real truck performance

3.2.2 Using test data from R6

The measured CC test run data on R6 are shown in Figure 3.11, i.e., road grade, the CC torque, and truck gear. The comparison of the CC speed and simulation model speed

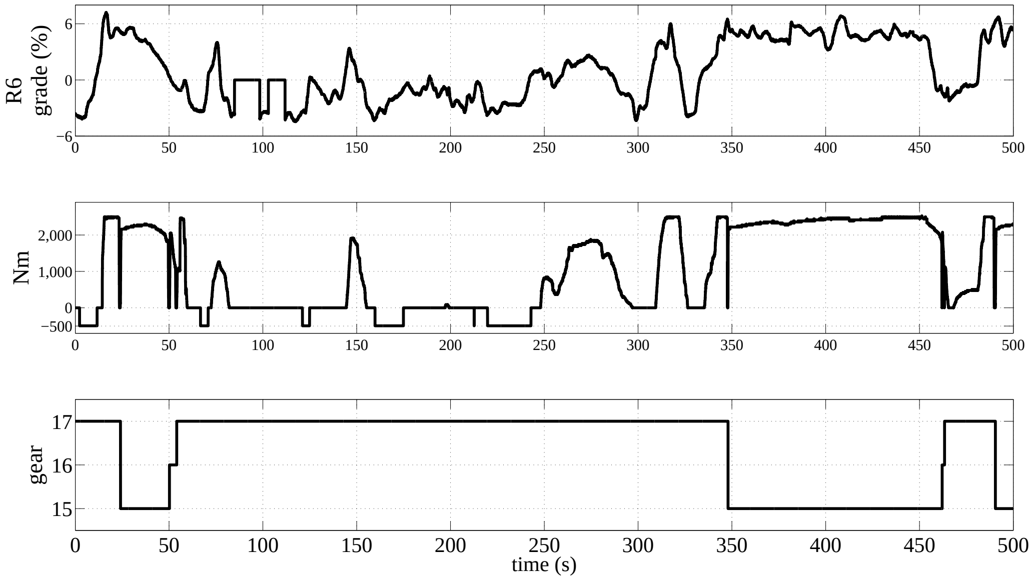


Figure 3.11: Measured data: road grade, CC torque, and gear, R6

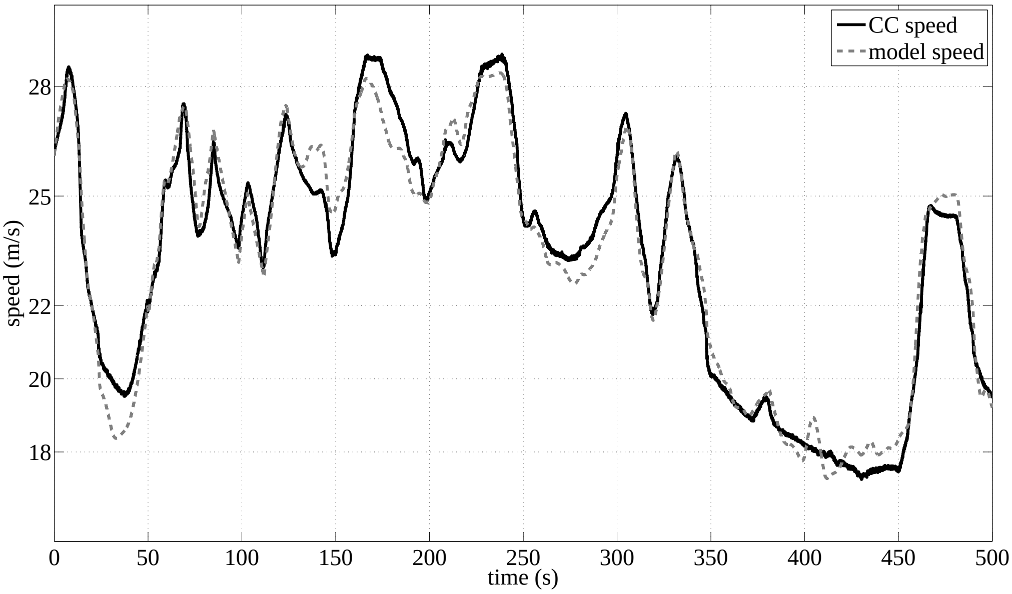


Figure 3.12: Comparison of CC and simulation model speed, R6

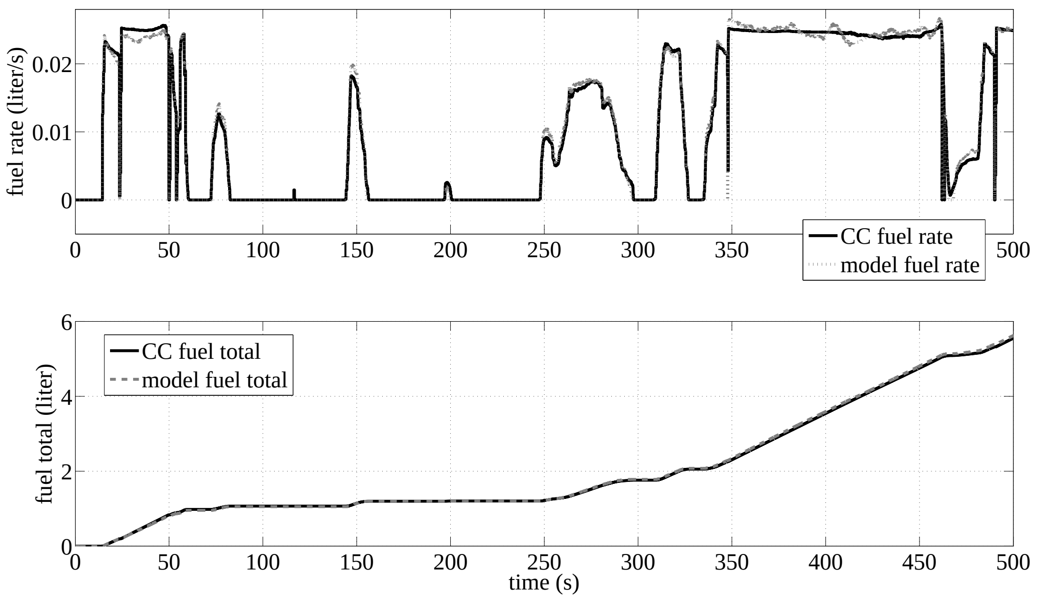


Figure 3.13: Comparison of CC and simulation model fuel use, R6

is shown in Figure 3.12. It can be observed that the simulation model speed matches the measured CC speed closely. There are some speed errors at 0 - 50 s and 350 - 500 s in Figure 3.12, which are due to the modeling error of the gear shifting time delay as discussed previously. Additionally, the errors occurred at 150 - 250 s are mainly resulted from the modeling error in the brake torque. The comparison of fuel rate (liter/s) and total fuel use (liter) for the CC and simulation model is presented in Figure 3.13. It can be seen in that the simulation and CC fuel rate and total fuel consumption are close to each other.

Based on the comparison results in this section, it can be seen that the truck model is accurate enough to predict a real truck's dynamic movement and fuel consumption. By using this model, the OC system can be designed and the high fidelity simulation tests can be conducted to evaluate the system performance before the real road tests are conducted.

3.3 Terrain Based Fuel Consumption

Finally in this chapter, simulations using the developed truck model are performed to generally quantify how the change in terrain type effects the truck fuel efficiency. The designed truck model is simulated to run on different highway profiles, generated previously in Section 2.1.2. A standard CC is used to track the constant speed. Through this simulation, the relation between fuel consumption and terrain conditions is analyzed. The road parameters fixed for the simulation are shown in Table 2.4.

Table 3.1: Simulated highways for fuel consumption analysis

Length	Design speed (km/h)	Terrain type	Frequency	Elevation
24 km	90/110	L/R/M	Low/Medium/High	High

In the following simulations, the fuel consumptions (liter) are calculated and compared for different highway profiles, which have the same driving distance (24 km), and different driving speeds (90 km/h and 110 km/h). The results are illustrated in Figures 3.14 and 3.15 as 3D plots, where the x, y, and z axes represent the change in terrain type, frequency component, and the corresponding fuel consumption change, respectively. Several points can be observed from these figures:

- At the same driving speed, the truck has a larger fuel consumption on the terrain with mountainous type, high frequency component, or both compared to the other terrains, as shown in Figure 3.14;
- Fuel consumption is heavily impacted by the change in terrain type, driving speed, and road frequency component;
- Larger road grade, driving speed, and road frequency component result in larger fuel consumption.

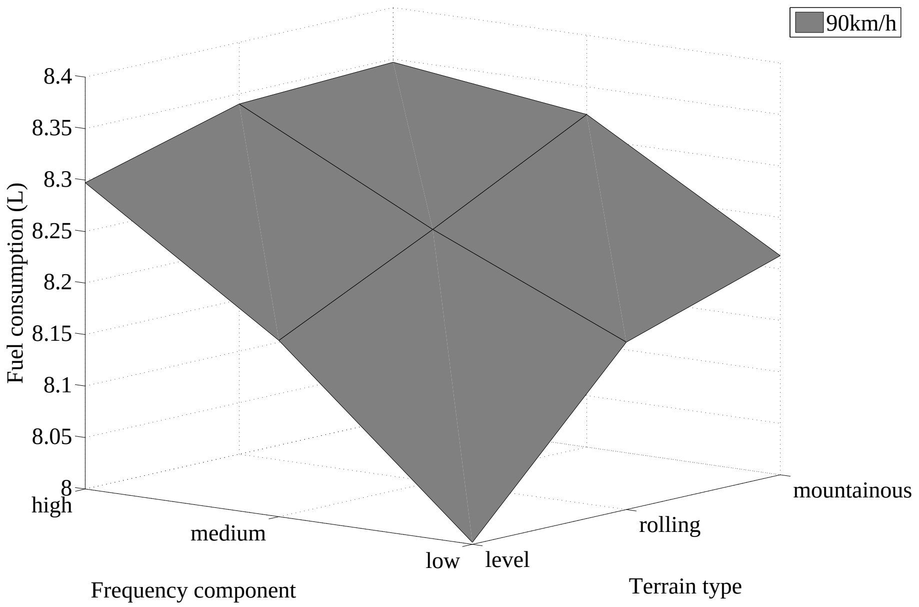


Figure 3.14: Fuel consumption and terrain conditions, 90 km/h

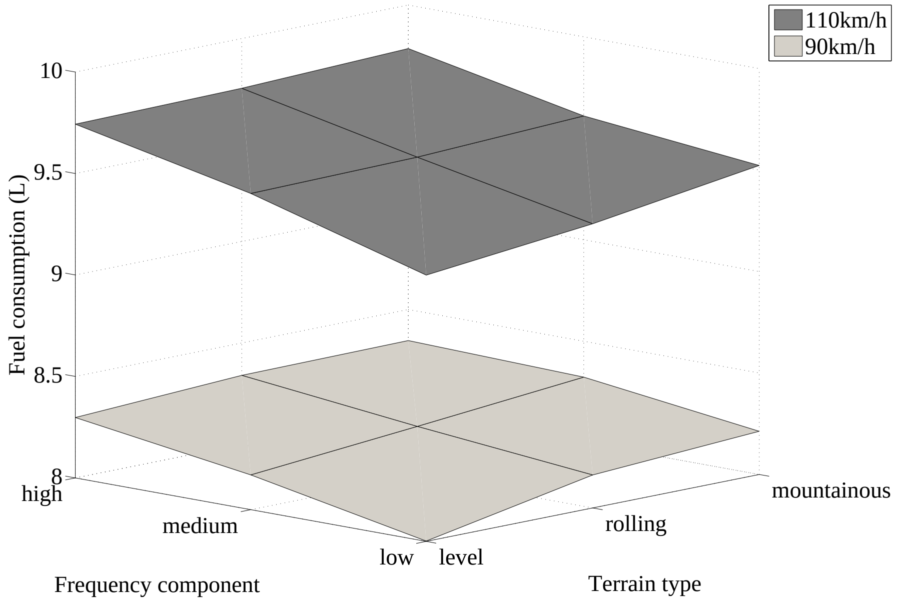


Figure 3.15: Fuel consumption and terrain condition, comparing 90 and 110 km/h

3.4 Conclusion

In order to design the model-based OC algorithm and test the system performance in a simulation environment, a heavy truck's dynamics and powertrain model were developed and validated in this chapter. Only longitudinal truck dynamics were considered and the applied engine map, fuel consumption map, and truck parameters were provided by Eaton. The model was validated by comparing the real road test data with the simulation data. In addition, simulations were run by using the developed model to quantify how the change in terrain type effects the truck fuel consumption. The simulation results showed that truck fuel consumption is impacted by the change in terrain type, driving speed, and road frequency component. A more in depth analysis of how the change in terrain type effects the truck fuel economy will be provided in Chapter 5.

Chapter 4
Optimal Control System Design

In this chapter, a 3D GIS road geometry based optimal powertrain control system (OC) is designed to reduce the heavy truck fuel consumption and travel time. The OC optimizer attempts to find a constrained minimum of a nonlinear scalar function of several variables, e.g., truck throttle u, gear g, brake b, and velocity v, based on the road geometry.

4.1 NLP solver operation

The direct collocation method is applied to transform the stated optimal control problem into a general nonlinear programming (NLP) problem formulation described in Equation (4.1). The resulting NLP problem is then solved by using the optimizer.

$$\begin{aligned} \text{minimize} \quad & f(x) && \\ \text{subject to} \quad c_i(x) &= 0, \quad i = 1, \ldots, q \\ g_j(x) &\leq 0, \quad j = 1, \ldots, m \\ x_{iL} &\leq x_i \leq x_{iU}, \quad i = 1, \ldots, n \end{aligned} \tag{4.1}$$

where f, c, and g are the objective, equality constraint, and inequality constraint functions respectively, x_{iL} and x_{iU} are lower and upper bounds for the variable x_i, and q, m, and n are the number of equality constraints, inequality constraints, and design variables, respectively. The functions f and c are twice continuously differentiable in this work.

4.1.1 Direction collocation method

The main reason to use the direct collocation method is that it does not need to consider explicitly deriving the necessary conditions such as maximum principle. This advantage makes the method especially attractive for complicated optimization problems. In this work, this method provides an approximated numerical solution to the optimization problem by discretizing the states and control inputs in each prediction horizon ($L = 3000$ m) into a set of step points ($s = L/h = 120$), with the step distance ($h = 25$ m). Thus, each prediction horizon is broken into smaller position intervals as: $p = \{p_1, p_2, \ldots, p_s\}$. Subsequently, the NLP problem is searching for the parameter vector including the number of $4s$ state and control inputs at the grid points such as $x = \{u_1, g_1, b_1, v_1, u_2, g_2, b_2, v_2, \ldots, u_s, g_s, b_s, v_s\}$ that minimize the objective function subject to specified constraints. The key point of the collocation method is to replace the original set of system dynamic constraints with a set of defect constraints $\zeta_k = 0$, which are imposed on each interval in the discretization. The equality constraint is then represented as $c(x) = [\zeta_1, \zeta_2, \ldots, \zeta_{s-1}]^T$ [14].

The collocation point is selected as the center of the segment between each set point. The state and control trajectories between the step points could be interpolated either by linear or cubic spline polynomial functions. If it is assumed that states vary linearly and the control inputs are constant across the step interval, the defect constraints for a linear collocation method can be written as in Equation (4.2).

$$\begin{aligned} \zeta_k &= f(x_{kc}, u_{kc}) - \dot{x}_{kc}, \qquad k = 1, \ldots, s-1 \\ x_{kc} &= \frac{x_{k+1} + x_k}{2}, \quad u_{kc} = u_k \\ \dot{x}_{kc} &= \frac{x_{k+1} - x_k}{h} \end{aligned} \tag{4.2}$$

Additionally, for a cubic collocation method, states are assumed to vary cubically and the control inputs vary linearly across the position step interval. The defect constraints are described in Equation (4.3).

$$
\begin{aligned}
\zeta_k &= f(x_{kc}, u_{kc}) - \dot{x}_{kc}, \qquad k = 1, \ldots, s-1 \\
x_{kc} &= \frac{x_i + x_{k+1}}{2} + \frac{h}{8}(\dot{x}_k - \dot{x}_{k+1}) \\
u_{kc} &= \frac{u_k + u_{k+1}}{2} \\
\dot{x}_{kc} &= -\frac{3}{2h}(x_k - x_{k+1}) - \frac{(\dot{x}_k + \dot{x}_{k+1})}{4}
\end{aligned}
\tag{4.3}
$$

In this work, the linear collocation method is applied, since the step length of 25 m is small. For such a short interval, a high-order cubic trajectory interpolation method, which makes the NLP problem really complex, would not significantly improve the system performance.

4.1.2 Evaluation of objective and constraint functions and their gradients

After the number of $4s$ (s is the number of step points) state and control inputs are initialized, the objective function $f(x)$, which is the sum of fuel consumption and travel time, is evaluated using Euler's numerical integration method with the step length h along the state and control vectors of Equations (3.15) and (3.16):

$$f(x) = J_{\text{fuel}} + J_{\text{time}} + J_{\text{gear}} \tag{4.4}$$

$$J_{\text{fuel}} = Qh \sum_{k=1}^{s-1} \frac{1}{v_k} \frac{dm_f}{dp}, \tag{4.5}$$

$$J_{\text{time}} = Rh \sum_{k=1}^{s-1} \frac{1}{v_k}, \tag{4.6}$$

$$J_{\text{gear}} = O \sum_{k=1}^{s-1} |g_k - g_{k-1}| \tag{4.7}$$

where J_{gear} is used to eliminate the frequent gear shifting between neighboring grid points. Q, R, and O are weighting factors which currently are determined by steady state model analysis to weight fuel consumption more than travel time and gear shifting penalty.

In Equation (4.5), the fuel consumption position rate $\frac{dm_f}{dp}$ has been given in Equation (3.15) and is described again in the following.

$$\frac{dm_f}{dp} = \frac{1}{v} \cdot (b_1\omega_e^5 P + b_2\omega_e^4 P + b_3\omega_e^3 P + b_4\omega_e^2 P + b_5\omega_e P + b_6 P)$$

where b_1 to b_6 are coefficients, P is the engine power, and ω_e is the engine speed. The engine power P in kilowatt (kW) can be calculated from T_e and ω_e by using the relation:

$$P = \frac{2\pi T_e \omega_e}{60000}$$

where T_e is the desired engine torque and ω_e is the engine speed. With a normalized throttle position u, T_e can be calculated from: $T_e = T_m u$, where T_m is the maximum engine torque approximated from the engine map. In addition, the relationship between the engine speed, ω_e, and truck velocity, v, is defined as:

$$\omega_e = \frac{30}{\pi}\frac{n_t n_d}{r} v$$

The equations above describe how the objective function in Equation (4.5) can be related to the variables need to be optimized, e.g., truck throttle u, gear g, brake b, and velocity v.

The analytical gradient and Hessian of the objective function, with respect to u, g, b, and v, are $\nabla f(x)$ and $\nabla^2 f(x)$ shown below.

$$\nabla J_{\text{total}} = [\frac{\partial J_{\text{total}}}{\partial u}, \frac{\partial J_{\text{total}}}{\partial g}, \frac{\partial J_{\text{total}}}{\partial b}, \frac{\partial J_{\text{total}}}{\partial v}] \tag{4.8}$$

$$\nabla^2 J_{\text{total}} = \begin{bmatrix} \frac{\partial^2 J_{\text{total}}}{\partial^2 u} & \frac{\partial^2 J_{\text{total}}}{\partial u \partial g} & \frac{\partial^2 J_{\text{total}}}{\partial u \partial b} & \frac{\partial^2 J_{\text{total}}}{\partial u \partial v} \\ \frac{\partial^2 J_{\text{total}}}{\partial g \partial u} & \frac{\partial^2 J_{\text{total}}}{\partial^2 g} & \frac{\partial^2 J_{\text{total}}}{\partial g \partial b} & \frac{\partial^2 J_{\text{total}}}{\partial g \partial v} \\ \frac{\partial^2 J_{\text{total}}}{\partial b \partial u} & \frac{\partial^2 J_{\text{total}}}{\partial b \partial g} & \frac{\partial^2 J_{\text{total}}}{\partial^2 b} & \frac{\partial^2 J_{\text{total}}}{\partial b \partial v} \\ \frac{\partial^2 J_{\text{total}}}{\partial v \partial u} & \frac{\partial^2 J_{\text{total}}}{\partial v \partial g} & \frac{\partial^2 J_{\text{total}}}{\partial v \partial b} & \frac{\partial^2 J_{\text{total}}}{\partial^2 v} \end{bmatrix} \tag{4.9}$$

The nonlinear equality constraint (4.10) is transformed to the defect constraints as described in Equation (4.2):

$$\begin{aligned} c_k(x) &= \zeta_k = v_{k+1} - v_k - h\frac{dv_{kc}}{dp} = 0 && (4.10) \\ \frac{dv_{kc}}{dp} &= \frac{1}{v_{kc}} \cdot \frac{r}{J_w + mr^2 + \eta_d n_d^2 \eta_t n_t^2 J_e} \cdot (\eta_d n_d \eta_t n_t T_e - F_s r - F_{rr} r - F_a r - T_b) && (4.11) \end{aligned}$$

and the gradient and Hessian functions are ∇c_k, and $\nabla^2 c_k$.

$$\begin{aligned} \nabla c_k &= [\frac{\partial c_k}{\partial u}, \frac{\partial c_k}{\partial g}, \frac{\partial c_k}{\partial b}, \frac{\partial c_k}{\partial v}] && (4.12) \\ \nabla^2 c_k &= \begin{bmatrix} \frac{\partial^2 c_k}{\partial^2 u} & \frac{\partial^2 c_k}{\partial u \partial g} & \frac{\partial^2 c_k}{\partial u \partial b} & \frac{\partial^2 c_k}{\partial u \partial v} \\ \frac{\partial^2 c_k}{\partial g \partial u} & \frac{\partial^2 c_k}{\partial^2 g} & \frac{\partial^2 c_k}{\partial g \partial b} & \frac{\partial^2 c_k}{\partial g \partial v} \\ \frac{\partial^2 c_k}{\partial b \partial u} & \frac{\partial^2 c_k}{\partial b \partial g} & \frac{\partial^2 c_k}{\partial^2 b} & \frac{\partial^2 c_k}{\partial b \partial v} \\ \frac{\partial^2 c_k}{\partial v \partial u} & \frac{\partial^2 c_k}{\partial v \partial g} & \frac{\partial^2 c_k}{\partial v \partial b} & \frac{\partial^2 c_k}{\partial^2 v} \end{bmatrix} && (4.13) \end{aligned}$$

Additionally, the linear inequality constraints $g(x)$ is defined by: $u \subset [0,\ 1]$, $g \subset [1,\ 18]$, $b \subset [0,\ 1]$, and $v \subset [v_l,\ v_u]$, where v_l and v_u are the lower and upper bounds of velocity.

4.1.3 Discrete gear ratio calculation

As stated in Section 1.2.2, the throttle and gear shifting are continuous and discrete inputs respectively, which makes the NLP problem a mixed-discrete NLP (MDNLP) problem. In general, the MDNLP problem can be solved by two groups of optimization methods: stochastic and deterministic methods. Some of the stochastic methods for global optimization include the sequential random search and various evolutionary programming methods such as simulated annealing and genetic algorithms. When the objective and constraint functions are explicitly expressed, deterministic methods can be applied. These methods include branch and bound, sequential linear programming, cutting plane techniques, outer approximation, Lagrange relaxation approaches, dynamic rounding-off techniques, and so on [29].

Additionally, as surveyed and determined previously in Section 1.2.2, in this research an extension interior-point (IP) algorithm with a rounding-off method is applied to solve the MDNLP problem. First, all discrete gear ratios, $n_t \subset [14.40, 0.73]$, are handled as continuous variables and optimized by using the IP method. Secondly, the optimal discrete gear ratios are obtained by rounding up each continuous gear ratio to the nearest discrete value above it, corresponding to a downshift. This method works well especially for those problems whose available discrete sizes are small [47]. In the problem each continuous gear ratio only has two nearest discrete gear ratios to round to, one above and one below, and the truck normally runs in a high gear on highways, which reduces the discrete sizes even further.

The two disadvantages of using a rounding-off method could be: the best discrete design point is not always selected; and the obtained discrete optimum may not be able to satisfy all feasibilities [48]. In Section 5.3.1, experimental results will be given to show how these disadvantages are overcame.

4.1.4 MDNLP solution module

For the MDNLP problem at hand, a solution suggested is to couple a direct collocation method with a deterministic discrete variable search strategy with the following steps:

1. Apply the direction collocation method to transform the infinite optimal control problem into a finite dimension (MDNLP) problem, where all design variables are treated as continuous;

2. The IP method, a standard NLP solver, is employed to find the optimum in the design space of continuous design variables;

3. The deterministic discrete search method (i.e., dynamic rounding-off) is introduced at the final step to find the discrete design variables.

The applied IP solver is from the MATLAB function 'fmincon'. The algorithm is essentially a barrier method in which the subproblems are solved by a sequential quadratic programming (SQP) iteration with trust regions [9, 10]. The complete IP optimizer function is presented in Figure 4.1.

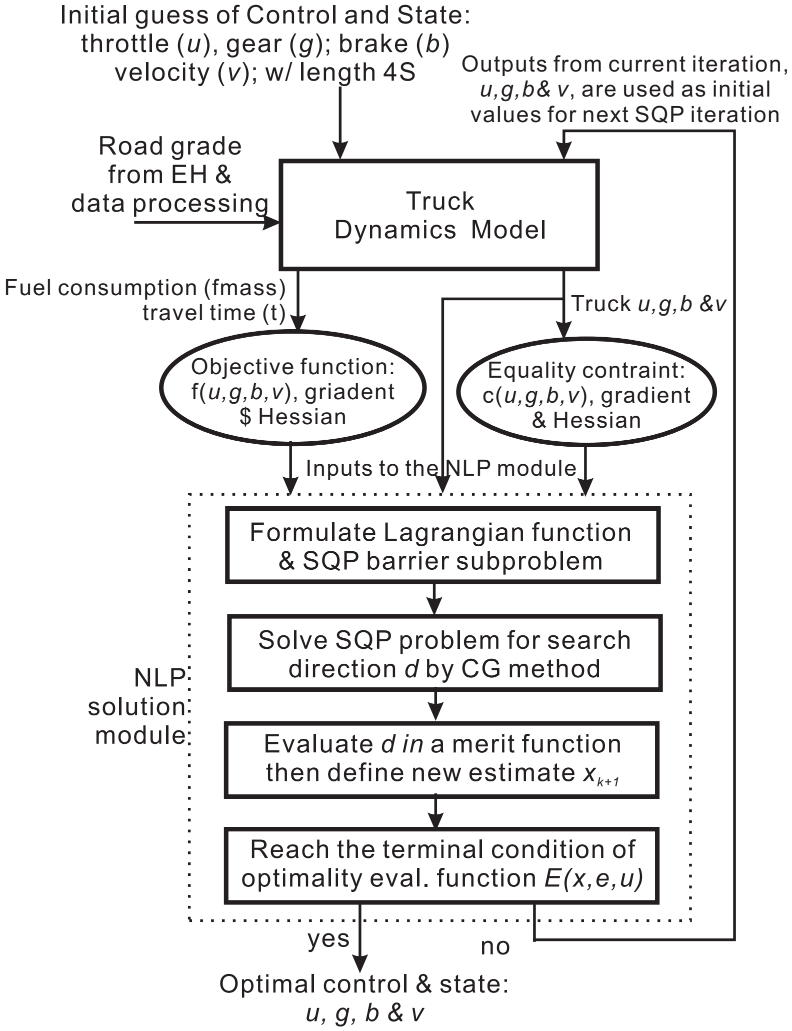

Initial guess of Control and State:
throttle (u), gear (g); brake (b)
velocity (v); w/ length 4S
Outputs from current iteration,
u,g,b& v, are used as initial
values for next SQP iteration
Road grade
from EH &
data processing
Truck
Dynamics Model
Fuel consumption (fmass)
travel time (t)
Truck u,g,b &v
Objective function:
f(u,g,b,v), griadent
$ Hessian
Equality contraint:
c(u,g,b,v), gradient
& Hessian
Inputs to the NLP module
Formulate Lagrangian function
& SQP barrier subproblem
Solve SQP problem for search
direction d by CG method
NLP
solution
module
Evaluate d in a merit function
then define new estimate x_{k+1}
Reach the terminal condition of
optimality eval. function E(x,e,u)
yes
no
Optimal control & state:
u, g, b & v

Figure 4.1: Complete NLP solver algorithm flow chart

By using the IP solver, each barrier subproblem is of the form:

$$\min_{(x,b)} \quad f(x) - \mu\Sigma_{i=1}^{4s}\text{ln}e_i \tag{4.14}$$

$$\text{subject to} \quad c(x) = 0, \quad g(x) + e = 0 \tag{4.15}$$

where $\mu > 0$ is a barrier parameter and the slack variable e is assumed to be positive. The trust region strategies are used to globalize the SQP iteration because they facilitate the use of the second derivative information when the problem is non-convex. The algorithm consists of five main stages:

- Formulate the Lagrangian function, L, of the barrier problem based on Equations (4.14) and (4.15), which is defined as:

$$\begin{aligned} L(x, e, \lambda_c, \lambda_g) = f(x) - \mu \Sigma_{i=1}^{4s} \text{ln} e_i + \\ \lambda_c^T c(x) + \lambda_g^T (g(x) + e) \end{aligned} \tag{4.16}$$

  where λ_g and λ_c are Lagrange multipliers with equality and inequality constraints.

- Define the search direction $d_x = x - x_i$ and $d_e = e - e_i$ and formulate the quadratic barrier subproblem based on Equations (4.14) - (4.16):

$$\begin{aligned} \min_{(x,e)} \quad & \nabla f(x_k)^T d_x + \frac{1}{2} d_x^T \nabla_{xx}^2 L_k d_x - \\ & \mu S_k^{-1} d_e + \frac{1}{2} d_e^T \nabla_{ee}^2 L_k d_e \qquad (4.17) \\ \text{subject to} \quad & \nabla c(x_k)^T d_x + c(x_k) = r_c \\ & \nabla g(x_k)^T d_x + d_e + g(x_k) + e_k = r_g \\ & (d_x, d_e) \subset T_k \end{aligned}$$

  This is a so called SQP trust region approach, where $\nabla_{xx}^2 L$ and $\nabla_{ee}^2 L$ are the Hessian of the Lagrangian with respect to x and e, respectively, the vector $r = (r_c,\ r_g)$ is a residual vector, and S=diag$(e_1, ..., e_{4s})$. The closed and bounded set T_k defines the region around x, where the approximated model and constraints in Equations (4.17) can be sufficiently trusted. The variables λ_g and λ_c are computed by a least squares approach.

- Solve the SQP problem and find an approximated solution of the optimum search direction $d = (d_x,\ d_e)$, by using the conjugate gradient (CG) method.

- If the direction d provides sufficient decrease in the merit function $\phi(x, e; \nu)$, then $x_{k+1} = x_k + d_x$, $e_{k+1} = e_k + d_e$, and compute new λ_g and λ_c; else set $x_{k+1} = x_k$, $e_{k+1} = e_k$, and shrink the trust region.

- Stop criterion is: $E(x, e; \mu) \leq \varepsilon_\mu$, where $E(x, e; \mu)$ is a function defined to measure the optimality conditions of the barrier problem, and ε_μ is a small positive value.

Finally, the optimal discrete gear ratio is obtained by rounding up each continuous gear ratio to the nearest discrete value above it, corresponding to a downshift.

4.1.5 Global and local optimization analysis

In real world optimization problems, functions of many variables have a large number of local minima. Local optimization methods are those methods used to straightforwardly find an arbitrary local minima. On the other hand, global optimization methods are developed to find the global minimum of a function, which is much more challenging and has been difficult to solve for many problems so far.

Stochastic methods such as simulated annealing and genetic algorithms are global optimization methods. Stochastic methods usually start from one point or multiple points drawn randomly from the search region. Trial points are generated according to some distribution (usually a uniform distribution) over the search region, and a better point is usually accepted or ranked higher than other points. Most deterministic methods, e.g., SQP, IP, usually use gradient information and are local optimization methods. Sometime the standard SQP and IP algorithms can avoid a local minimum and find a global minimum for some special problems, e.g., a convex optimization problem.

A set $S \subseteq R^n$ is said to be convex if whenever x, $y \in S$ and $0 \leq t \leq 1$, there is $(1-t)x + ty \in S$, which implies that a set is convex if the line segment joining any two points in the set lies within the set as given in Figure 4.2.

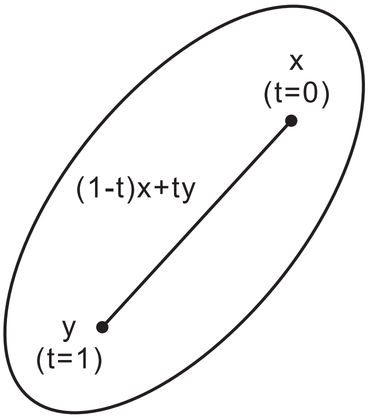


Figure 4.2: Convex set

A function f is said to be convex if D_f is a convex set and if whenever x, $y \in D_f$, and $0 \leq t \leq 1$, there is:

$$f((1-t)x + ty) \;\; \leq \;\; (1-t)f(x) + tf(y) \tag{4.18}$$

This definition means that a function is convex if the line segment joining any two points of the function curve lies entirely above the function curve as depicted in Figure 4.3. If the objective function is convex, inequality constraints are convex, and the equality constraint functions are affine, the problem is called a convex optimization NLP problem [49].

For a convex problem, any starting point can be picked up to start a local search, and it is guaranteed to obtain the global minimum. Unfortunately, most optimization problems are nonconvex, and there are many local optima. Thus, different starting points may result in quite different final solutions for local optimization methods. Most researchers have recognized the problem of finding a good starting point, and some have even recommended

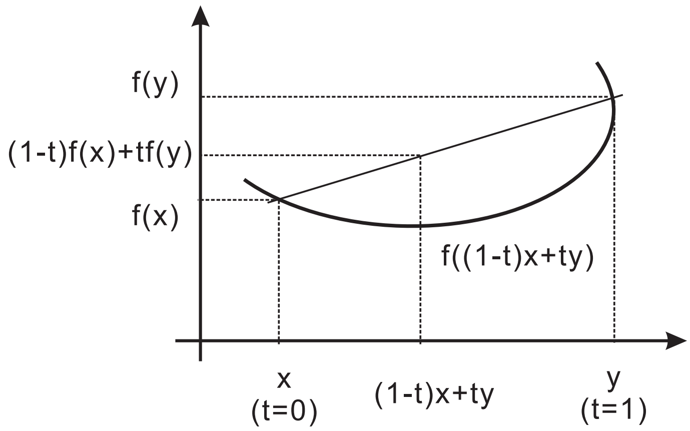


Figure 4.3: Convex function

making several attempts (trying several times, for example) at finding good starting points [50].

In a traditional truck diesel engine, without any electric control inputs, the Brake Specific Fuel Consumption (BSFC) map typically has a convex shape [3]. However, determined by various control inputs such as throttle and gearshift, the BSFC time rate map applied in this work does not have a convex shape even in the operating range $\subset$ [1100 1800]rpm, which is depicted in Figure (4.4). Therefore, the objective function as shown in Equation (4.4), as well as the NLP problem in this work, are not convex. This makes it non-trivial to find the global minimum by using the IP method.

However, it can be seen from Figure 4.4, the BSFC map is relatively flat under the nominal highway truck operation condition, i.e., relatively smooth and slow changes of the engine power and speed, which corresponds to a small fuel consumption variation range on the BSFC map. Thus, the nonconvex problem could have a small number of local minima, when the upper and lower bounds are tightly specified as in Section 4.1.2. For the problem at hand, by defining the initial guess as shown later in Section 4.4.1, the IP can exhibit a rapid second-order convergence toward a local minimum, which could be identical or close to the global minimum. A simple test further shows that by using any initial guess satisfying the inequality constraints in Section 4.1.2 on any road conditions the optimal solutions

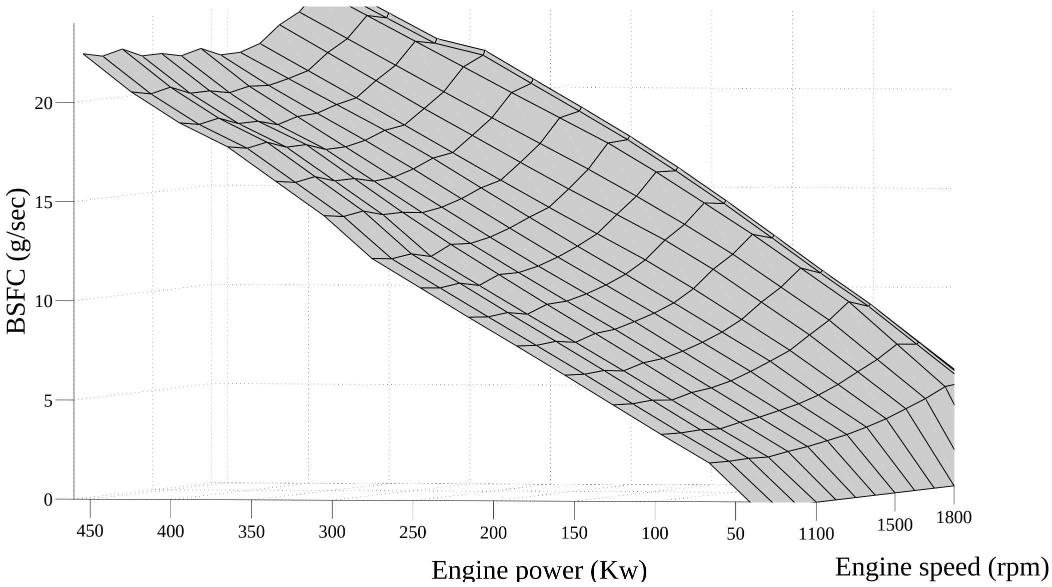


Figure 4.4: Zoomed-in BSFC time rate map

converge to the lower or higher speed bounds if only the minimal fuel or time is demanded in the objective function, respectively. These results are in accordance with the global minimum which can be expected in the common driving behavior that is: driving with low speed consumes the minimal fuel; and driving with minimal travel time demands large fuel consumption.

Similarly, if the complete objective function as described in Equation (4.4) is taken into account, simulation results display that for any feasible initial guess, which is the arbitrary choice of throttle within [0, 1], zero brake, the top gear of 18, and the constant set speed, the solver is able to quickly converge to the unique optimal solution. This solution could be identical or close to the global minimum as well, since the applied objective function is just a weighted sum of the minimal fuel, time, and the gear shifting.

Additionally, when the objective function is not convex, the fuel optimal speed trajectory may very heavily depend on the road grade. Furthermore, if a problem covers a range of speeds and accelerations that calls for gear shifts, the gear shifting is highly nonlinear and generates oscillation between gears, which could be eliminated by using the penalty function as in Equation (4.7).

4.1.6 Series NLP solver operation

The NLP solver designed previously is modified in this section to sovle the optimal brake torque in a better way. An example of running the optimizer on a positive 4% single crest is shown in Figure 4.5, where the elevation map, optimal speed, engine torque, and truck gear are given from the top to the bottom plots. Such an optimal solution can be obtained with any feasible initial guess stated above.

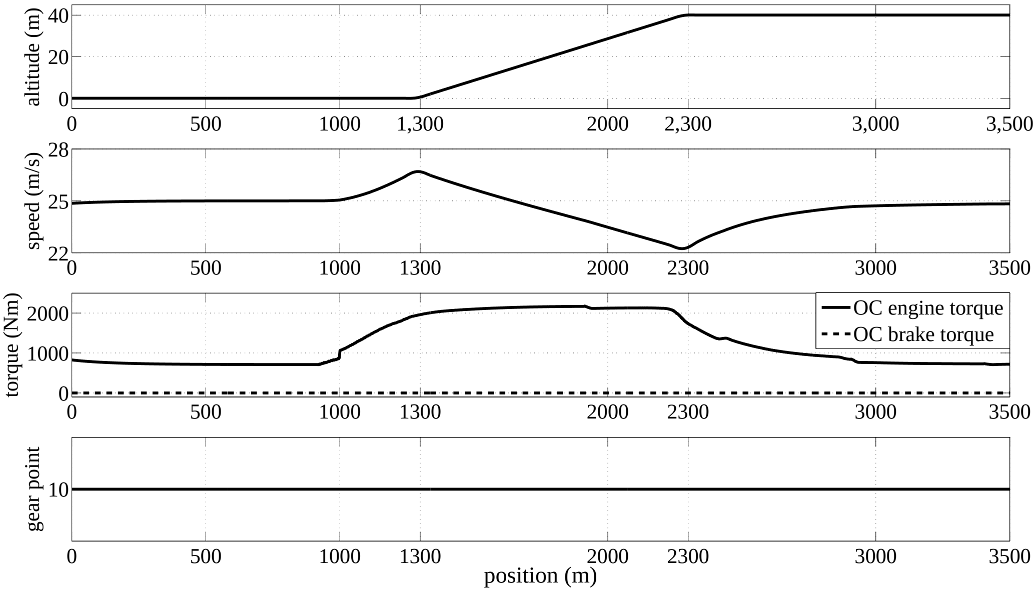


Figure 4.5: Optimal velocity, engine/brake torque, and gear shifting, on a 4% crest

However, on a large downhill terrain, if the constant set speed is taken as the initial guess, it could be possible that large brake toque T_b is calculated from the optimizer to maintain the constant speed. The brake torque T_b is determined by a brake input $b \subset [0,\ 1]$ multiplied by a maximum torque ($T_b = bT_{bm}$). However, this optimal solution wastes the potential energy which could be otherwise gained on the downhill. Therefore, the single

NLP optimizer in Figure 4.1 is modified to demand the least possible braking, which is both beneficial for fuel efficiency and vehicle operation.

The basic flow chart of using NLP optimizers in a series operation is illustrated in Figure 4.6, which includes the following steps:

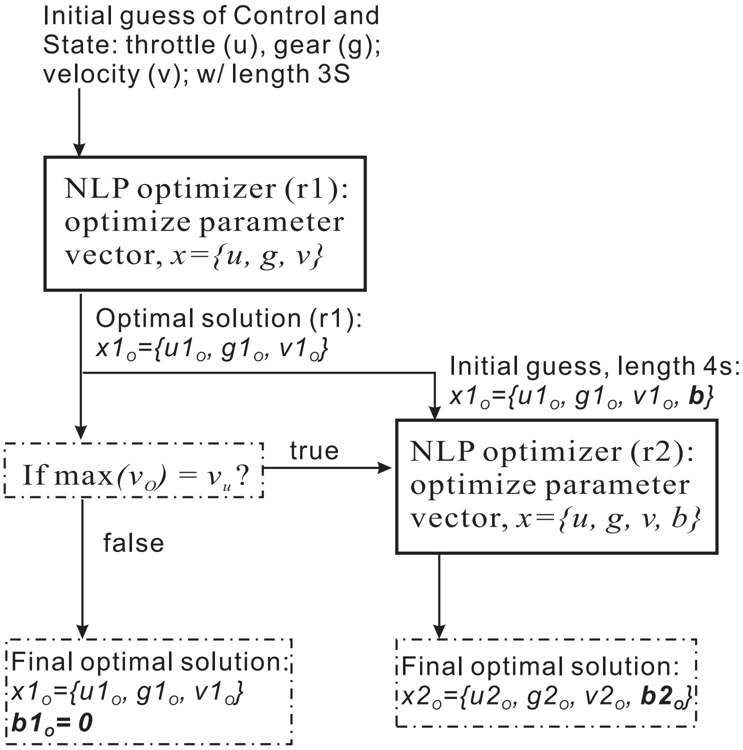
1. NLP first run ($\mathrm{NLP}_{\mathrm{r1}}$): brake input T_b is omitted from Equation (4.11), as well as from the state and control inputs that need to be optimized. The only system control inputs are truck throttle and gear, and the optimal solution includes optimal throttle, truck gear, and velocity trajectory;

2. Speed limit check: detect if the optimal speed from $\mathrm{NLP}_{\mathrm{r1}}$ has reached v_u, the upper speed bound. If it is true, a NLP second run is triggered. Otherwise, the solution from $\mathrm{NLP}_{\mathrm{r1}}$ is deemed as the final optimal solution and T_b is set as a zero vector;

3. NLP second run ($\mathrm{NLP}_{\mathrm{r2}}$): T_b is taken into account in Equation (4.11) and the state and control inputs that need to be optimized. The control inputs are throttle, brake, and truck gear, and the initial guess is the optimal solution from $\mathrm{NLP}_{\mathrm{r1}}$ plus brake as a zero vector. Afterwards, the solution from $\mathrm{NLP}_{\mathrm{r2}}$ is used as the final optimal solution.

Another benefit of using this method is that it improves the computation efficiency. Since on the normal level and rolling highways without a heavy traffic the truck brake is not frequently commanded, it is reasonable to run the $\mathrm{NLP}_{\mathrm{r1}}$ first on most roads. This way the optimizer minimizes a reduced size parameter vector which consists of only the truck throttle, gear, and velocity. Therefore, it is faster and more efficient than calculating the full size parameter vector all the time which includes also the brake torque.

By using this method, the maximum fuel saving, maximum truck speed, and minimum brake command are gained on a large single downhill. The optimal solution by using the NLP optimizer series operation on a -4% single sag slope is demonstrated in Figure 4.7. It is seen that the truck reduces speed and engine torque before the sag and then gains speed from coasting down the sag slope. It can be observed from the bottom plot that only if the



Initial guess of Control and State: throttle (u), gear (g); velocity (v); w/ length 3S
NLP optimizer (r1): optimize parameter vector, x={u, g, v}
Optimal solution (r1): x1o={u1o, g1o, v1o}
Initial guess, length 4s: x1o={u1o, g1o, v1o, b}
If max(vO) = vu?
true
false
NLP optimizer (r2): optimize parameter vector, x={u, g, v, b}
Final optimal solution: x1o={u1o, g1o, v1o} b1o= 0
Final optimal solution: x2o={u2o, g2o, v2o, b2o}

Figure 4.6: Series NLP optimizer operation for brake calculation

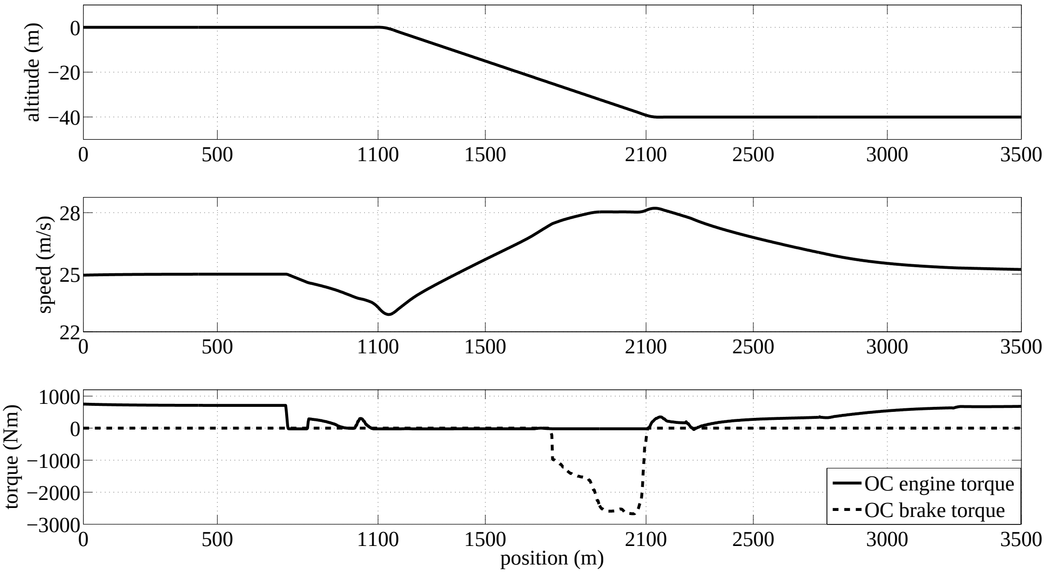


Figure 4.7: Optimal velocity, engine/brake torque, and gear shifting, on a -4% sag

truck reaches the velocity upper bound v_u, brake is commanded. This is of a great benefit to real road driving.

4.2 Real-time optimal control system

The optimal velocity calculated by using the NLP solver needs to be performed in real time. This is accomplished by a feedback controller. In the system in this research, the slow and fast dynamics are separated by the NLP solver and truck speed control, respectively. Trajectory generated in the NLP solver is the slow outer-loop while the control is the fast inner-loop. The concepts of inner and outer loops are formalized by a two degree-of-freedom (DOF) control system architecture where the inner loop is used for stabilizing a nominal reference velocity trajectory generated by the outer loop [38]. Figure 4.8 demonstrates an inner-outer loop structure for achieving optimal feedback control. Under this structure, traditional PID control is used to design the feedback regulation on the inner loop while the outer loop uses the NLP solver.

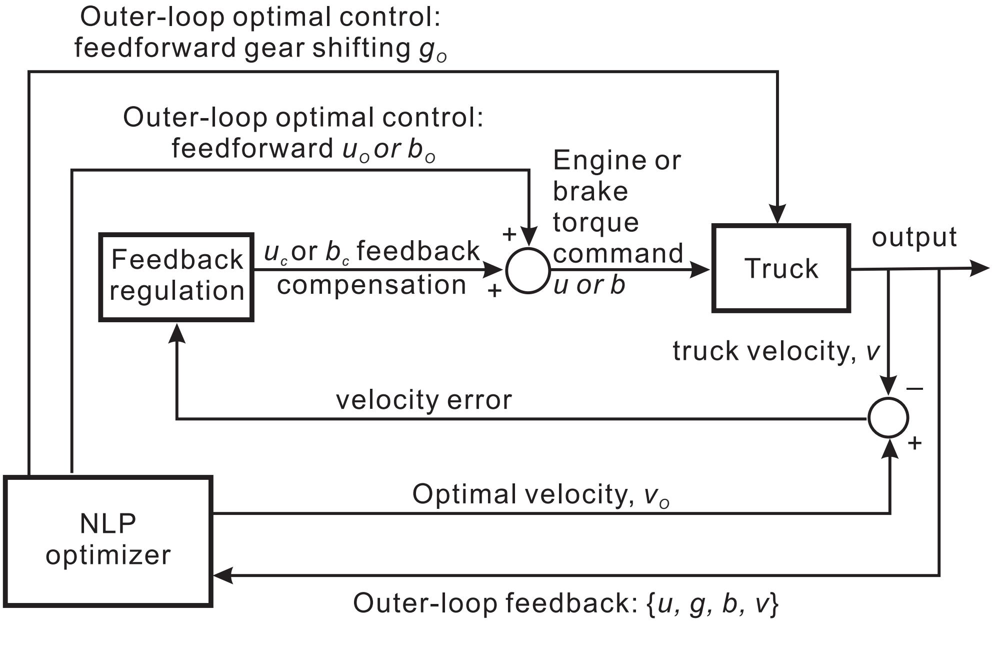

Outer-loop optimal control: feedforward gear shifting g_o
Outer-loop optimal control: feedforward u_o or b_o
Engine or brake torque command u or b
Feedback regulation
u_c or b_c feedback compensation
+
+
Truck
output
truck velocity, v
velocity error
–
+
Optimal velocity, v_o
NLP optimizer
Outer-loop feedback: {u, g, b, v}

Figure 4.8: Real-time fuel optimal feedback control through two DOF structure

The function of the two DOF optimal feedback control system was detailed in Section 1.2.3, which consists of the following steps for the fuel-optimal control system:

1. In the outer loop, the optimal control problem is solved using the NLP solver repeatedly on a moving prediction horizon to yield near optimal velocity trajectories. Meanwhile, the optimal controls such as u_o, g_o, b_o are fed forward to the inner loop. In the outer loop feedback, u and g are the measured engine and engine brake torque;

2. In the inner loop, a feedback (PID) regulation controller is developed to calculate the control compensation u_c or b_c, based on the velocity error between the optimal and real truck velocities;

3. Finally, the control signals, $u = u_o + u_c$, $b = b_o + b_c$, and g_o, are used to accurately track the optimal velocity trajectory over some period;

4. The procedure is repeated over the next prediction horizon.

4.3 Complete real time system structure

The real time test system consists of various modules, including the map-matching system and an optimal powertrain control system (OC) with the NLP solver, as presented in Figure 4.9.

4.3.1 CAN interface and Electronic Horizon

The controller area network (CAN) interface gets sensor inputs from CAN receivers and sends them to the main software module. It also receives control commands from software modules and then sends to a CAN transmitter. Electronic horizon (EH) is a map-matching system to utilize the truck's current position and the road information ahead, based on measurements from GPS, inertial navigation system (INS) sensors, and the road geometry. EH outputs both road geometry information and positioning information. The road geometry

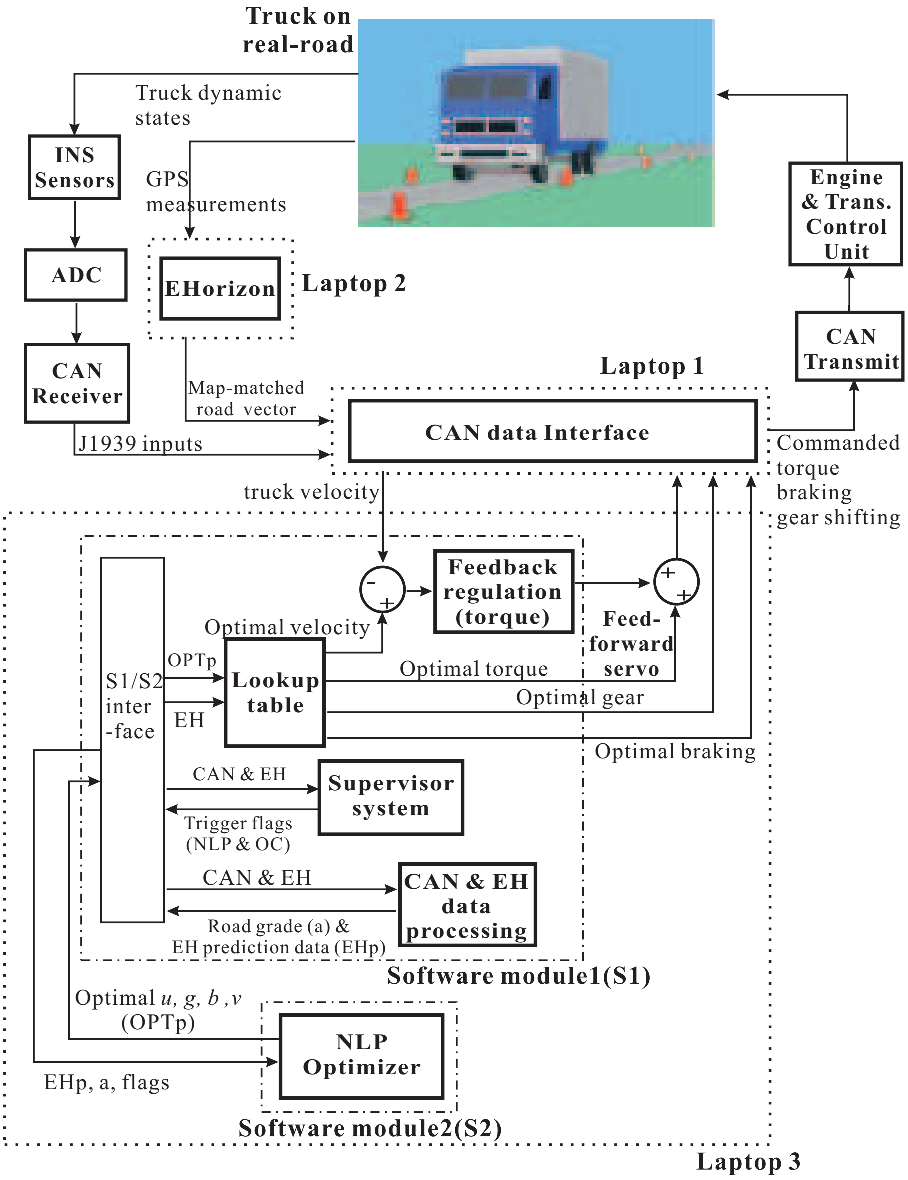


Figure 4.9: Real time experimental system setup

ahead of the vehicle's current location - called the "horizon" - is sent as an ordered series of Shape Point (SP) messages. The vehicle position is sent as a map-matched Position Message (PM) with coordinates relative to the horizon information that have previously been sent. This allows the application to know where the vehicle is with respect to the road geometry that it already received from the SP messages.

The data structure in the SP and PM are listed in the following:

- shpIndex: Index for each shape point;
- *dE*: Distance between every two shape points in the east direction;
- *dN*: Distance between every two shape points in the north direction;
- *dZ*: Elevation difference between every two shape points;
- branchBit: Flag for the branch road.

Normally, the distance between every two shape points is around 3.5 m, and therefore there are around 900 shape points in a prediction horizon which is 3 km long, as determined in Section 4.1.1. The EH update is depicted in Figure 4.10.

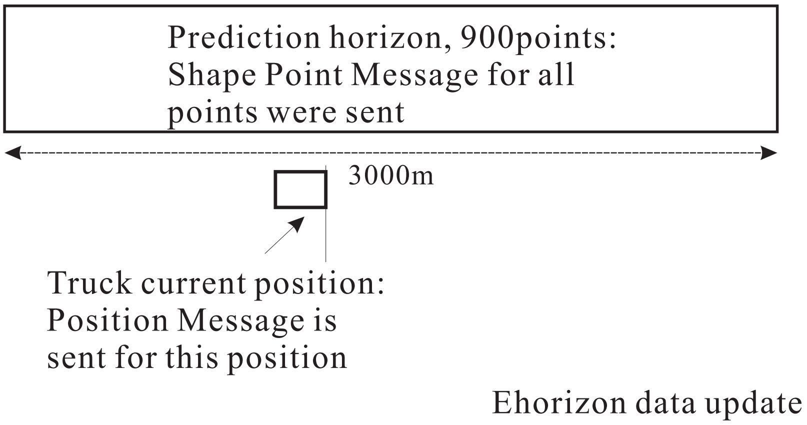


Figure 4.10: Electronic Horizon system information update

4.3.2 Software Modules 1 and 2

Software module 1 (S1) mainly consists of:

- CAN and EH data processing: calculate distance and road grade for a prediction horizon, and save them to an EH buffer;
- Supervisor system: read EH and CAN inputs and handle different OC operation conditions, e.g., start, stop, traffic in/out, fork road, by sending different 'flags';

- Lookup table: output map-matched optimal states and control inputs, which are within the truck operation limitations, from an OC solutions buffer;
- Optimal velocity tracking: apply the two DOF feedback control structure as described in Section 4.2, where a torque control mode is used as a feedback regulation and the optimal torque command is applied as a feedforward servo control. This local tracking controller can regulate the system to closely track the desired trajectory and guarantee the closed-loop system is stable.

Software module 2 (S2) runs the NLP optimizer algorithm.

S1 and S2 are implemented in MATLAB and SIMULINK in Laptop 3 which communicate to a CAN data interface in Laptop 1 as shown in in Figure 4.9. In Laptop 3 the NLP optimizer and other subsystems such as data processing and supervisor system are run on two different Matlab platforms S1 and S2, which communicate to each other by exchanging the corresponding data from a unique workspace. Such a parallel system structure is crucial for the correct functioning of the multi-rate system in the real time application where the update rate of the subsystems in S1 is 0.01s and the required NLP optimization time in S2 is approximately 1s. If all subsystem functions were run in a series structure in S1, the fast rate data subsystems would be stalled by the slow rate NLP optimizer and consequently the map data from EH would be delayed significantly. A clear description of the complete system flow chart is given in Figure B.6 in Appendix B.

4.4 Real time system operation

The OC is a look-ahead controller which solves the optimization problem for a certain prediction horizon ahead, and performs the optimal velocity trajectory by applying the desired torque and optimal gear.

4.4.1 OC prediction update and initialization

In this work, the prediction horizon L and step distance h are chosen as 3000 m and 25 m, respectively, and there are 120 step points, $s = L/h = 120$. Thus, the NLP problem is searching for the optimal parameter vector $x = \{u, g, b, v\}$, which consists of the number of $4s$ state and control inputs.

The selection of L and h is based on the road profile, truck length, and computation time. The slope length of a single highway crest or sag is normally shorter than 3000 m [43]. Therefore, the 3000 m horizon is sufficiently long for the NLP solver to “see” the upcoming hills. The step distance h is selected close to the length of the experimental truck. It has a length of 22.86 m which is made up of a conventional tractor and a single 16.15 m trailer. Additionally, it is evident that the shorter the horizon, the less costly the solution of the on-line optimization problem. However, when a really short prediction horizon is used, the actual truck velocity trajectory will differ from the predicted optimal trajectory, even if no model plant mismatch and no disturbances are present. It is mainly because the solution for the repeated minimization over a finite horizon and the solution for the infinite horizon problem differ significantly if a short horizon is chosen [51]. Thus, a much longer prediction horizon, 3000 m, is used.

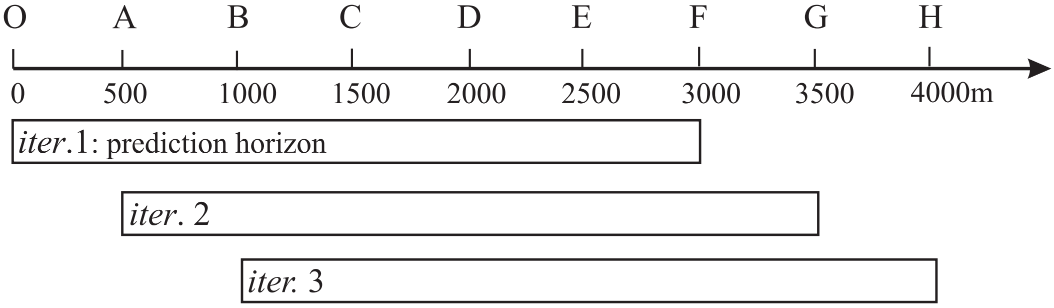


Figure 4.11: Recursive implementation of the optimal control system

In the real time application, the OC works repeatedly with the prediction horizon moving forward, as in Figure 4.11. While the truck runs on the starting section OA, the solver calculates the optimal u, b, g, and v, for the prediction horizon OF, and then saves them to a buffer, which is the first iteration (*iter*.1). It is assumed that the OC starts while the truck is on a level road with constant speed close to the reference speed of 25 m/s. Therefore, the initial guess of u, b, g, and v can be simply chosen as level road steady-state values, i.e., $u = 0.4$, v=25 m/s, $g = 18$, for NLP_{r1} in the series NLP solver operation.

When the truck runs on AB, it "reads" out the map-matched optimal solutions obtained on *iter*.1 and performs the optimal velocity trajectory by applying the calculated torque and optimal gear shifting. Meanwhile, the NLP solver calculates the optimal solutions for the next horizon AG, which is the second iteration (*iter*.2). The initial guess here is the optimal solution for AF from *iter*.1 plus the level road steady-state values for the small section FG. It is evident the optimal solution for AB from *iter*.1 is used as the initial guess for *iter*.2 and is performed by the truck in real-time. Thus, on AB the deviation between the truck's actual velocity trajectory and the optimal trajectory calculated on *iter*.2 is relatively small, which assures the continuation of the optimal trajectory in the buffer as well as the actual truck speed.

4.4.2 Optimization time

It can be seen from Section 4.4.1 that the optimization time t_{OP} is a key issue for the OC real-time operation. As shown in Figure 4.11, the NLP solver needs to solve the optimal solution for the next prediction horizon $L = 3000$ m when the truck runs on the window moving section Lw=500 m, e.g., on OA to calculate the optimal solution for OF. If the truck reference velocity is 25 m/s, then t_{OP} must be less than 20 s. This NLP solver is run on a Windows XP Laptop (Intel Core2 6300 1.86 Hz, 2 Gb of RAM).

Here, t_{OP} can be reduced by providing the NLP solver with gradients of objective and constraint functions, as in Equations (4.8), (4.9), (4.12), and (4.13). This way the number of function evaluations at each 'fmincon' iteration is largely reduced. Consequently, the computation time for each prediction horizon is reduced to around 1 s, when the truck moves 25 m at a 25 m/s reference speed. The computation time is quick enough for the real-time implementation on the PC hardware mentioned above. By applying this method, the MDNLP solver can quickly solve more complex problems, e.g., smaller h and longer L. Additionally, by comparison, it is found that with a longer L and shorter h, the gain in fuel economy could be improved, which could be further quantified in future work.

4.5 Conclusion

A 3D GIS road geometry based optimal powertrain control system was designed to reduce the heavy truck fuel consumption and travel time. A direct collocation method was applied to transform the optimal control problem into a mixed-discrete nonlinear programming (MDNLP) problem to find the constrained minimum of a nonlinear scalar function of several state and control inputs, e.g., truck throttle u, gear g, brake b, and velocity v, based on the road geometry. An OC optimizer which uses an interior-point algorithm plus a rounding-off method was designed to solve this MDNLP problem.

The real time test system consists of various modules, including the map-matching system and the OC with the MDNLP solver. The OC is a look-ahead controller, which solves the optimization problem for a 3000 m prediction horizon ahead of the truck and performs the optimal velocity trajectory by applying the desired torque and optimal gear. For each prediction horizon, the OC computation time is around 1 s, which is quick enough for the real-time implementation loaded on the selected PC hardware.

Chapter 5

Road Tests and Result Analysis

The developed OC system is implemented in MATLAB and SIMULINK using a model-based design method. Both simulation and real road tests are conducted to evaluate the performance of the OC. In this chapter, the basic OC function is tested and analyzed using the simple simulated terrain. Subsequently, real road geometries R1 to R4 are used in a high fidelity simulation environment to further validate the OC system performance. Finally, the real road tests on routes R5 and R6 are conducted and the tests results are analyzed.

In this work, the reference speeds are set ranging from 25 m/s to 31.1 m/s and the maximum speed variation is 2 m/s. As stated in [43], trucks generally increase speed by up to about 7 percent on downgrades and decrease speed by 8 percent or more on upgrades as compared to their operation on level ground. Therefore, the 2 m/s variation would be acceptable for real world driving, and comfortable for truck drivers.

5.1 Experimental system setup

In this section, both the simulation system and real-time experimental system setups are described.

5.1.1 Simulation system setup

In order to precisely simulate the real experimental environment and evaluate the designed OC system before the real road test, a high fidelity simulation setup is developed. The control system is then evaluated off-line by using the simulation model to examine and tune the algorithms and parameters. The complete simulation system as shown in Figure 5.1 is built in MATLAB and SIMULINK by the efforts both from Auburn and Eaton. The simulation setup includes a simulated map-matching system, a real OC system, and a simulated heavy truck model. The first two modules are designed by Auburn, and the last one is provided from Eaton. The system update rate is 0.01 s, which is the same as the real road test system.

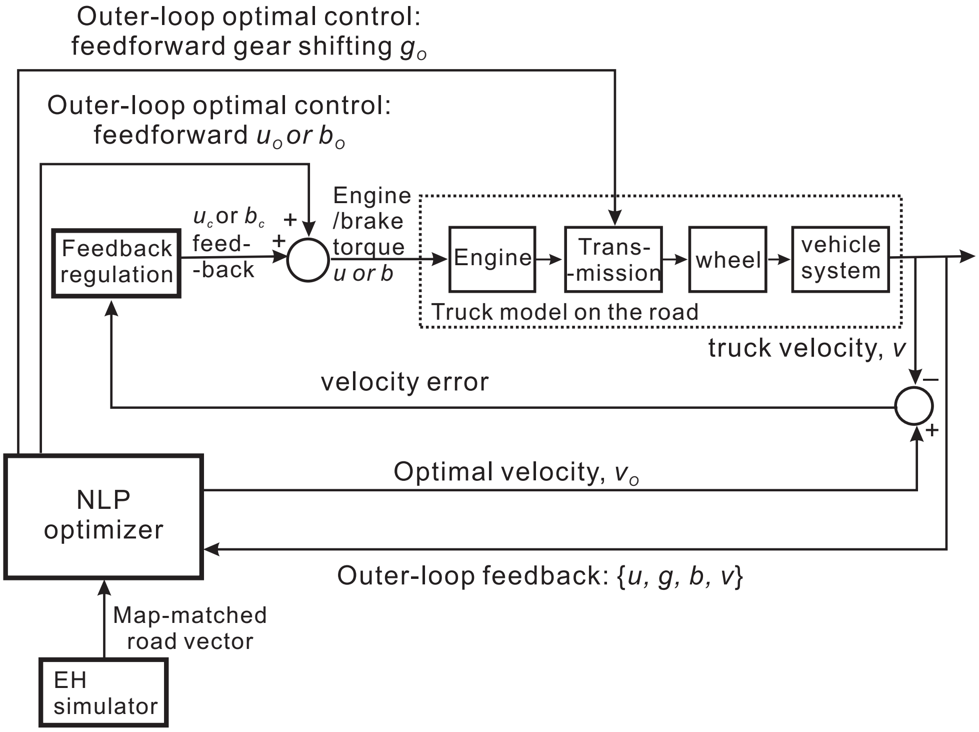


Figure 5.1: Simulation system setup

In the simulation test, a heavy truck model is replaced the real truck and interfaced with the other systems. The truck model includes three main modules: environment and driver cycle, operator, and plan and controls. The detailed functions of each module are listed as follows:

- Environment and drive cycle: the drive cycle is defined as a constant speed of 25 m/s, and environment parameters such as the road conditions are generated from Intermap's 3D road geometry.
- Operator: it consists of a modified PID cruise controller to generate both acceleration and brake pedal commands and then send to the driver outputs.

- Plant and controls: the plant side includes the engine, torque converter, transmission, final drive, wheel, and vehicle longitudinal dynamics. The controls contain an engine control and shift control.

By using this model, the baseline CC control run can be conducted and the fuel consumption and travel time baseline is deemed reliable.

In the simulation test, the optimal velocity profile calculated from the OC is used to replaced the constant speed of 25m/s as the drive cycle in the environment and drive cycle module. To perform this drive cycle, the cruise controller in the operator module is used to generate the torque compensation, which together with the calculated optimal engine and brake torque are sent as the engine and engine brake commands to the engine system blocks in the plant and controls module. Additionally, the optimal gear shifting from the OC is sent as the transmission command directly to the transmission system block.

5.1.2 Real-time experimental system setup

The real time experimental test system which consists of various modules, including the map-matching system and the OC with the NLP solver, has been discussed in Section 4.2 and is shown in Figure 5.2 again.

In the real-time experiment, the heavy truck is a Class 8 Caterpillar truck. It is equipped with an internal combustion diesel engine: CAT C13, 12.5 liter diesel engine, 6 cylinders, max horsepower: 410 kW at 1900 rpm, max torque: 2509 Nm at 1200 rpm. An Eaton Fuller AutoShift 18 speeds transmission is installed in the truck, which is a shift-by wire automated manual transmission (AMT) system. This allows gear shifting operation without driver involvement, similar to an automatic transmission, while possessing the high efficiency of a manual transmission.

The communication between the ECU and the actuators and sensors are based on a J1939 protocol on top of a CAN 2.0B at a baud rate of 1 Mb per second. A laptop-CAN interface card is used to interface the CAN receive/transmit via a RS232 port on the

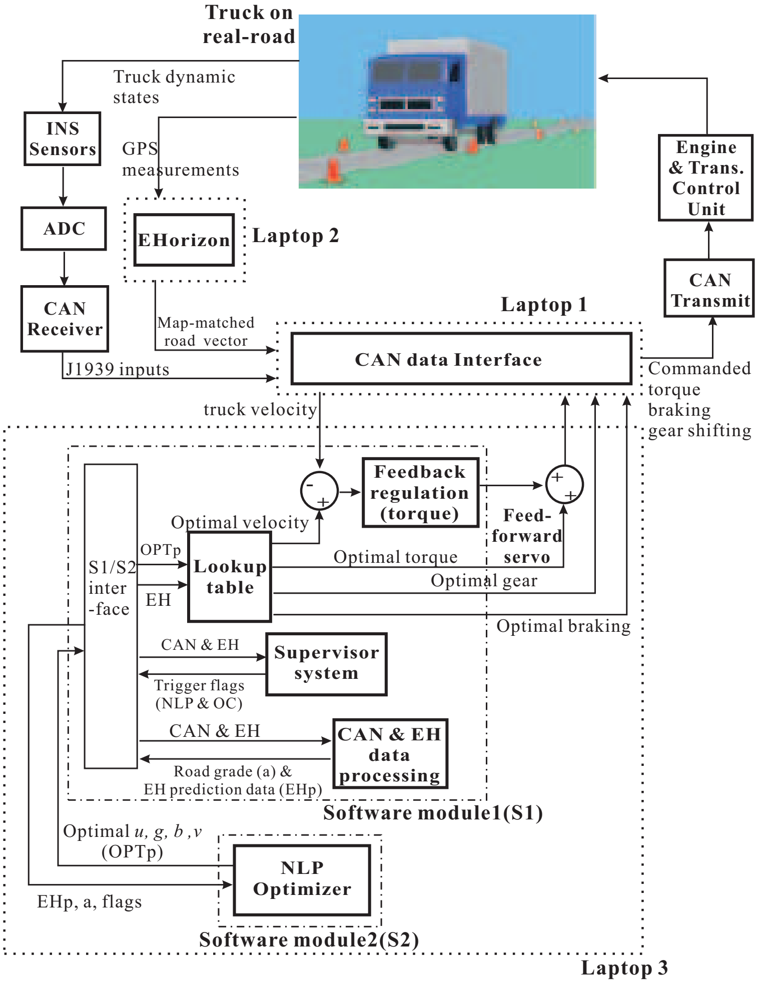


Figure 5.2: Real time experimental system setup

Laptop 1 as shown in Figure 5.2. From the Vector CANape software installed on the laptop, important sensors measurements such as engine torque, engine speed, fuel rate, engine brake torque, transmission speed, gear position, wheel speed, gyroscope data, etc., can be obtained. Additionally, an EH control unit integrated on the Laptop 2 calculates the truck position using a low cost GPS, wheel speed, and gyroscope data. The map data are then continuously transmitted in a standardized format to the Laptop 1 through the CAN bus. The complete OC system is run on the Laptop 3 (Windows XP Laptop, Intel Core2 6300 1.86 Hz, 2 Gb of RAM), where the update rate of the subsystems in S1 is 0.01s and the required NLP optimization time in S2 is approximately 1s. The communication between Laptop 1 and 3 is via a USB connection.

This rapid prototyping system enables the engineers to easily test the vehicle on the road in a real driving phase. From the real-time measurement, the engineers could quickly analyze the performance of the OC, modify the NLP solver, and build the modified code to prototype vehicle computer in a fast and cost-effective manner [34].

5.2 Simulation tests with simulated roads

This section describes how the change of terrain effects system performance. Numerical solutions are obtained to show and support the analyses. First, the comparison of the OC and CC fuel consumption and travel time on a single crest and sag is discussed. Second, the change of fuel savings and travel time on different road grades and slope lengths are simulated and analyzed.

5.2.1 Single sag and crest

The comparison of the OC and CC performances, with a 25 m/s set speed, on simulated single hills is show in this section. As stated in Section 2.2, the CC is applied to maintain the desired vehicle velocity and a function of brake pedal control is engaged if the truck reaches an upper speed limit of 27 m/s. On the other hand, the OC has a speed variation of ±2 m/s. Simulations are conducted on a typical single sag and crest in highway design, which has a slope length of 1000 m, elevation of ±40 m, and a resulting road grade of ±4%, respectively. The elevation maps are given in the top plot of Figures 5.3 and 5.5.

On the -4% sag, the OC is able to reduce 6.86% fuel consumption with 0.1% travel time increase, compared with the CC. The comparison of system performance and fuel consumption is shown in Figures 5.3 and 5.4. It is seen that the OC reduces truck speed, engine torque, and fuel rate before the sag and then gains speed from coasting down the sag slope. In this way, the OC is able to reduce the truck fuel consumption as well as to

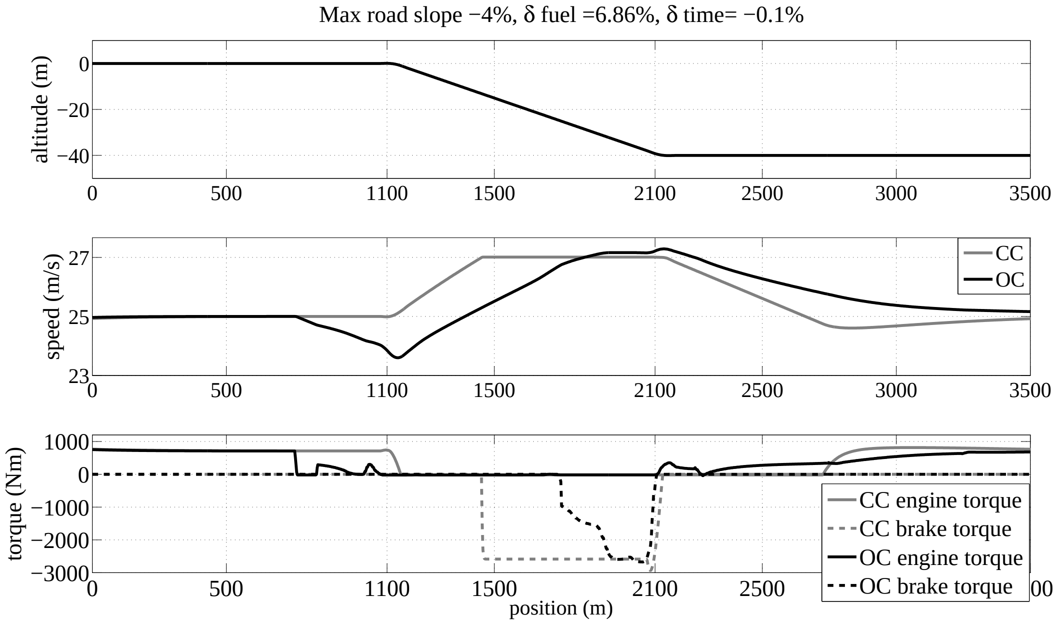


Figure 5.3: Comparison of velocity, engine/brake torque, and gear: CC & OC, on a -4% sag

maintain nearly the same travel time. Additionally, it can be seen from the bottom plot of Figure 5.3 that the OC commands less braking than the CC does, by reducing speed before the sag slope. This is of a great benefit to real road driving as it reduces brake wear.

The comparison of system performance on a 4% single crest is shown in Figures 5.5 and 5.6, where the OC is able to reduce the fuel consumption by 0.68%, while increasing the travel time by 0.59%. It is seen that the OC accelerates the truck by requiring large engine torque and fuel rate through 1000 - 1300 m prior to climbing the crest. While on the crest, the OC reduces torque but the CC demands large torque. An additional benefit is gained from the gear shifting command. It is clear from the bottom plot of Figure 5.5 that by increasing truck speed before the crest slope, there is less speed loss and subsequently less downshifting as required by the OC's operation.

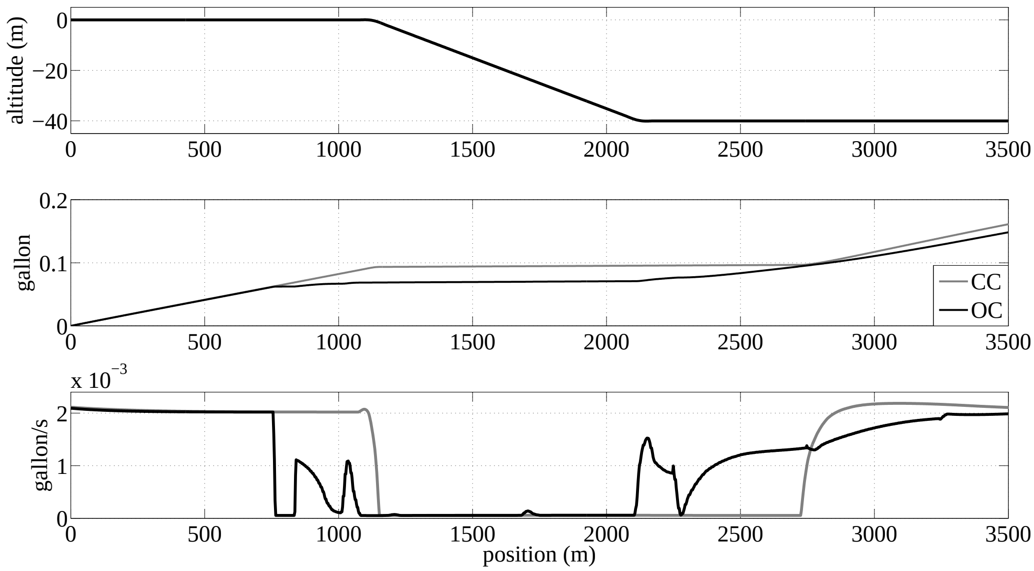


Figure 5.4: Comparison of fuel consumption and rate: CC & OC, on a -4% sag

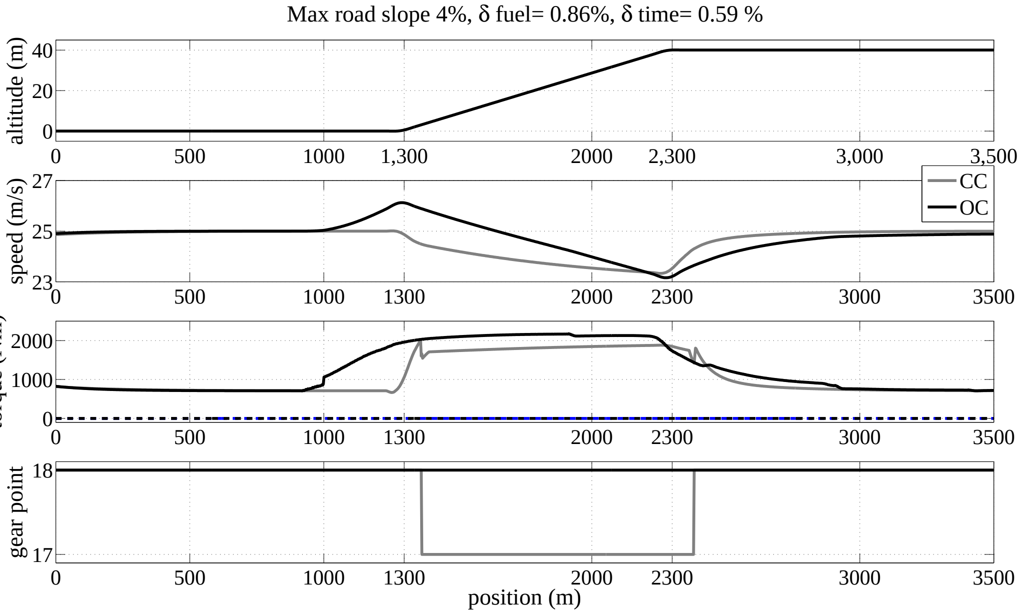


Figure 5.5: Comparison of velocity, engine/brake torque, and gear: CC & OC, on a 4% crest

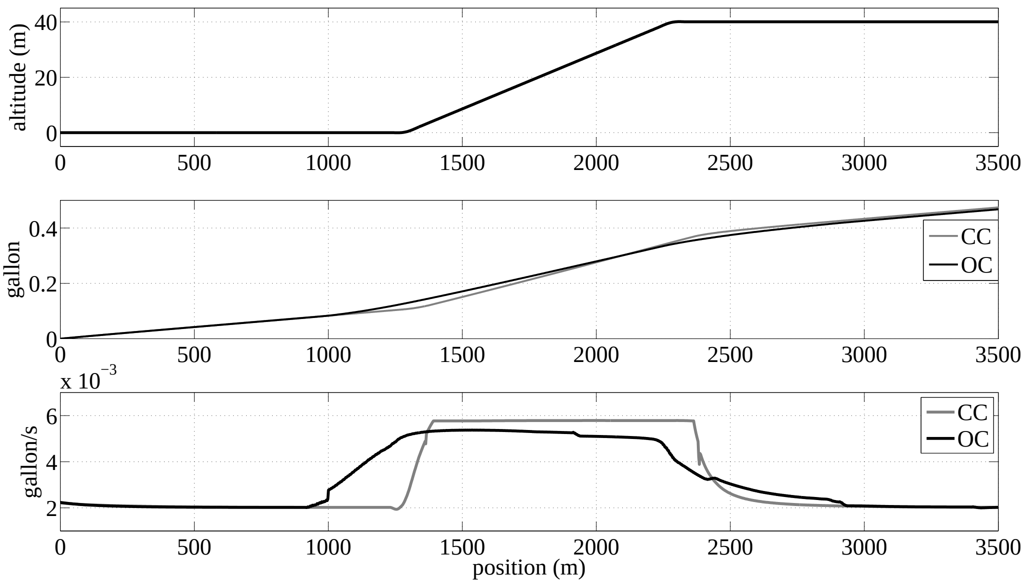


Figure 5.6: Comparison of fuel consumption and rate: CC & OC, on a 4% crest

5.2.2 Different road grade and slope length

The comparison of the OC and CC performances, with a 25 m/s set speed, on single hills with different road grades and slope lengths is discussed in this section. The road grades and slope lengths vary from -5% to 5% and 0 to 1500 m, respectively. For each combination of road grade and slope length, a simulation is conducted to calculate the fuel saving and travel time change of the OC compared to the CC. The changes in fuel saving and increase travel time are shown in Figures 5.7 and 5.8, where the positive percentage value represents savings.

It can be seen that the maximum fuel saving, 12.2%, is obtained on the -5% road grade and 1500 m slope length. On the contrary, the OC consumes 1.83% more fuel than the CC on the maximum road grade and slope length 5% and 1500 m, respectively. Additionally, for most road grades smaller than 4%, the fuel saving is positive which means the OC is able to gain fuel saving over the CC. Figure 5.8 shows that the OC incurs very small travel time increases, less than -1%, compared to the CC. The OC can gain the largest travel time

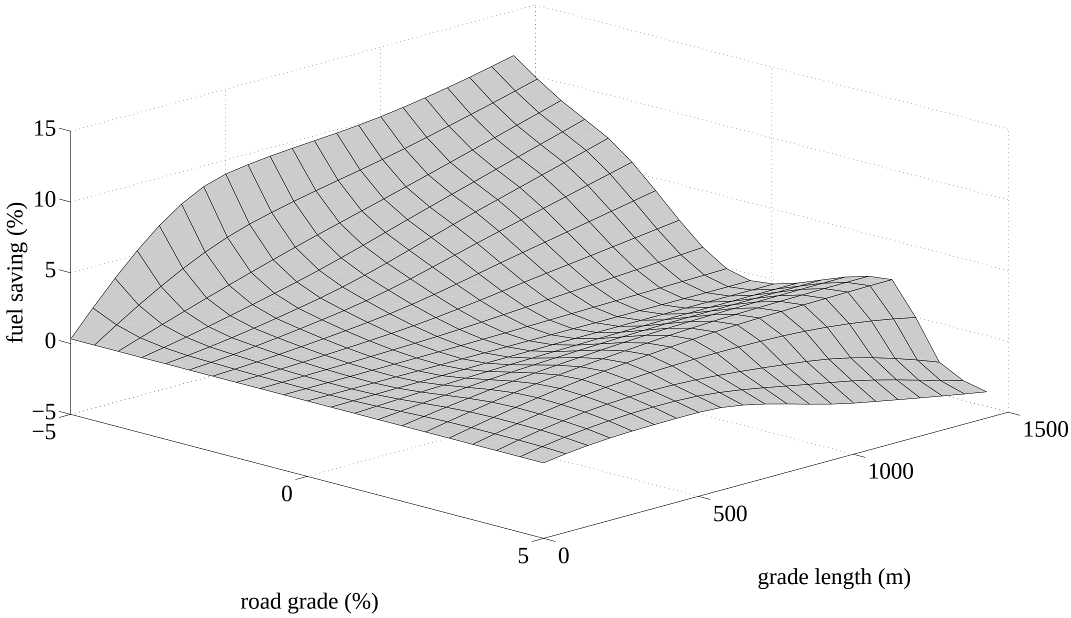


Figure 5.7: Fuel savings on different road grades and slope lengths, OC vs. CC

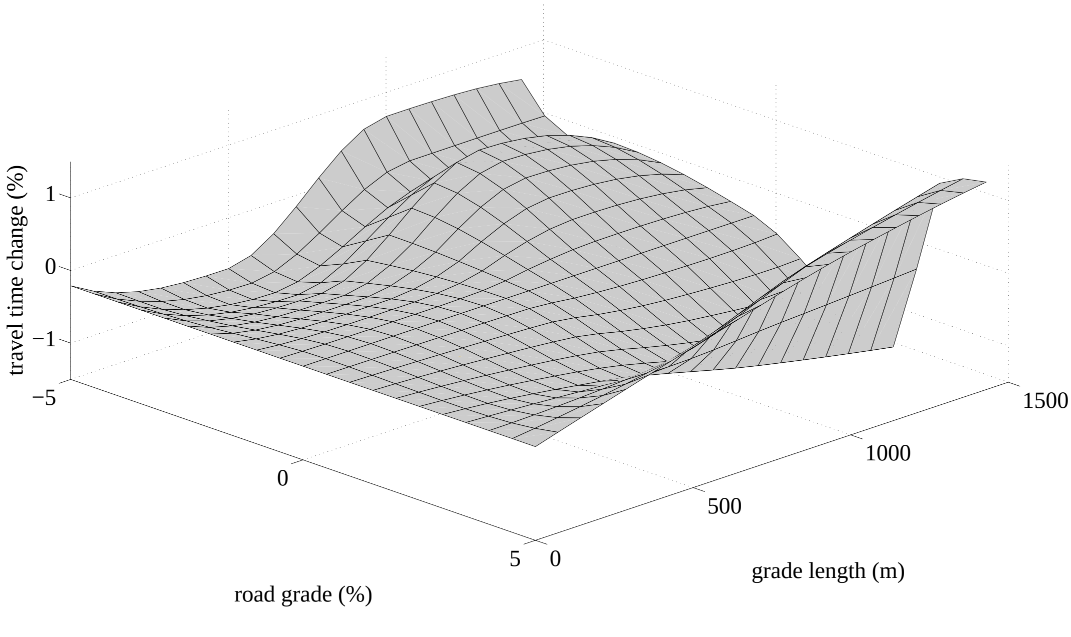


Figure 5.8: Travel time increases on different road grades and lengths, OC vs. CC

reduction, 2.1%, on the 5% road grade and 1500 m slope length corresponding to the road condition where the maximum fuel increase is obtained.

In general, the gain in fuel economy and the change of travel time are directly related to the change of road grade and slope length. Therefore, on large single sag sections large fuel savings could be gained, while on large crest sections the OC may consume more fuel but reduce travel time, etc.

5.3 Simulation tests with real road geometries

Subsequently, simulation tests are conducted to evaluate the OC system performance on different real road geometries, R1 to R4. The details of all six road profiles were provided previously in Table 2.4 and repeated in Table 5.1 in this chapter, where the mean, maximum, and minimum road slopes are listed as percentages. The comparison of the OC and CC system fuel consumption and travel time are shown and analyzed. Finally, how the change of set speed effects the fuel saving is investigated.

Table 5.1: Intermap road profiles analysis

		slope(%)				
route	length (m)	mean	max	min	σ	terrain type
R1	37000	-0.21	2.65	-4.33	1.05	level
R2	37000	0.21	4.33	-2.65	1.05	level
R3	47000	0.27	4.88	-3.87	1.34	rolling
R4	47000	-0.27	3.87	-4.88	1.34	rolling
R5	54000	0.21	5.57	-5.43	3.05	mountainous
R6	54000	-0.21	5.43	-5.57	3.05	mountainous

5.3.1 Real road geometries tests and results analyses

The OC and CC performances are compared on real road geometries R1 to R4. The set speed is 25 m/s, the speed variation for the OC is ±2 m/s, and the upper speed limit for the CC is 27 m/s. The comparison of fuel efficiency (FE) in miles/gallon, fuel consumption in gallons (F), and travel time (T) from the OC and CC is shown in Table 5.2, where positive percentage values represent savings. It is clear that the OC gains an average fuel saving 2.86% and 2.66% with 1.0% travel time increases on R1/2 and R3/4, respectively, when compared to the CC.

Table 5.2: Comparison of fuel consumption and travel time, R1 to R4

Route	$\mathrm{FE_{CC}}$	$\mathrm{F_{CC}}$	$\mathrm{F_{OC}}$	Diff(%)	$\mathrm{T_{CC}}$(s)	$\mathrm{T_{OC}}$	Diff(%)
R1	12.41	2.64	2.57	2.78	1462.6	1478.2	-1.1
R2	8.87	3.15	3.07	2.40	1484.5	1497.4	-0.8
Mean	7.99	2.90	2.82	**2.66**	1473.6	1492.4	**-1.0**
R3	6.53	4.50	4.38	2.60	1869.1	1883.8	-0.8
R4	9.17	3.20	3.10	3.10	1861.4	1883.7	-1.1
Mean	7.62	3.85	3.74	**2.86**	1865.2	1883.8	**-1.0**

Furthermore, Table 5.2 shows that the average fuel saving from R3/4 (2.86%) is larger than that from R1/2 (2.66%). The reason is that the gain in fuel saving on rolling terrains R3/4 is larger than that on level terrains R1/2. Additionally, when looking at the same road section but different directions such as R1 versus R2 or R3 versus R4, the gains of fuel saving on R1 (2.78%) is larger than that on R2 (2.40%). Similarly, fuel savings on R4 is larger than on R3. This difference is in that R1 and R4 are downhill terrains which have negative mean road grade and R2 and R3 are uphill roads with positive mean road grade as can be seen in Table 5.1. In general, the observation here on real road geometries is in accordance with what was found in Section 5.2.2. Therefore, the gain in fuel saving could be more significant on the rolling terrain with negative mean road grade.

Additionally, it is also necessary to investigate how the travel time increase effects the fuel saving. Therefore, another group of baseline runs in simulation are conducted on R1 to R4 and the CC reference speed is slowed down (to the different values on R1 to R4) to make the CC and OC has almost the same travel time. Then, the fuel consumption from the OC is compared with the new CC runs and the result are listed in Table 5.3. It is seen the OC is still able to gain an average fuel saving 2.40% and 2.01% without travel time increases on R1/2 and R3/4, respectively. These fuel savings are still acceptable for this research.

Table 5.3: Comparison of fuel consumption with same travel time, R1 to R4

Route	F_{CC}	F_{OC}	Diff(%)	T_{CC}(s)	T_{OC}	Diff(%)
R1	2.61	2.57	2.00	1478.0	1478.2	-0.02
R2	3.13	3.07	2.02	1497.0	1497.4	-0.03
Mean	2.87	2.82	**2.01**	1487.5	1487.8	**-0.02**
R3	4.47	4.38	2.20	1883.4	1883.8	-0.03
R4	3.18	3.10	2.58	1883.2	1883.7	-0.03
Mean	3.83	3.74	**2.40**	1883.3	1883.8	**-0.03**

Finally, the maximum CC fuel efficiency obtained on R1 is 12.41miles/gallon, because it is a level downhill terrain and therefore has better fuel efficiency than its reverse direction R2, which is an uphill road. R3 generally has the worst fuel efficiency among all roads for the reason that it is an uphill rolling terrain. The condition is similar to what was observed in Section 3.3.

In order to analyze the OC and CC system performance, the simulation results on R3 are shown since R3 is a typical rolling terrain. The comparison of engine and brake torque, truck gear, speed, fuel rate, and total fuel consumption from the CC and OC is shown in Figures 5.9 and 5.10. It can be seen that both systems perform the drive cycle, and have a maximum speed of 27 m/s.

Additionally, it can be observed from the bottom plots of Figures 5.9 that the OC and CC have similar gear shifting points. Thus, the OC gear shifting calculated by a rounding-off method as described in Section 4.1.3 is feasible for truck driving conditions which have been defined as various constraints in Section 4.1.2. Moreover, the OC requires less downshifting than the CC, which is an optimal gear selection to reduce fuel consumption. Thus, by using a rounding-off method for the problem at hand, the disadvantages as discussed in Section 4.1.3 could be overcome. Additionally, by considering the term J_{gear} in the cost function (4.4), the rapidly repeated gear shifts are prevented.

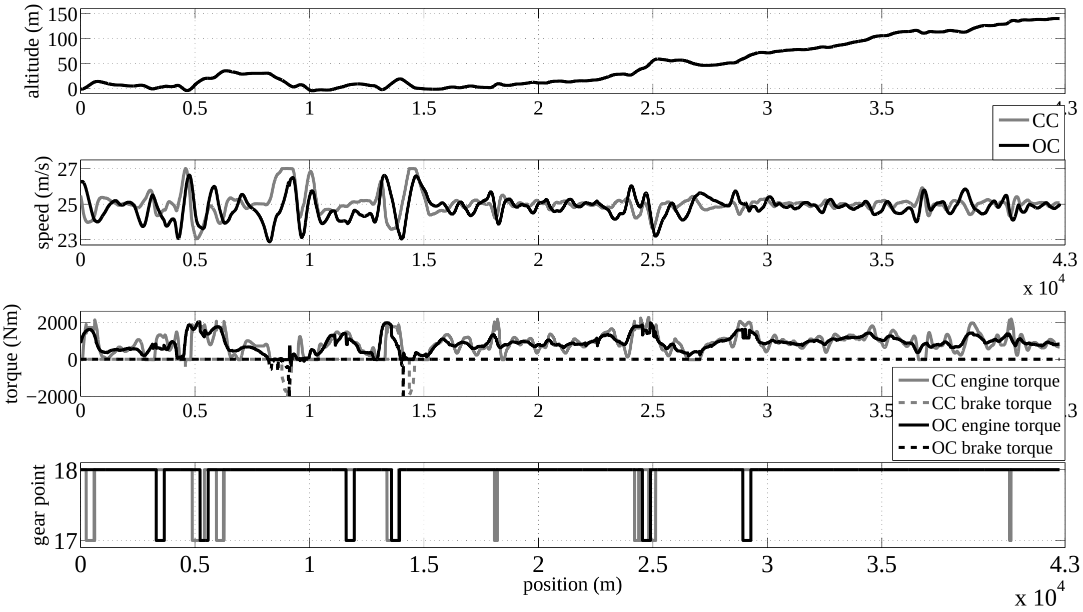


Figure 5.9: Comparison of velocity, torque, and gear on R3 with 25 m/s set speed, OC vs. CC

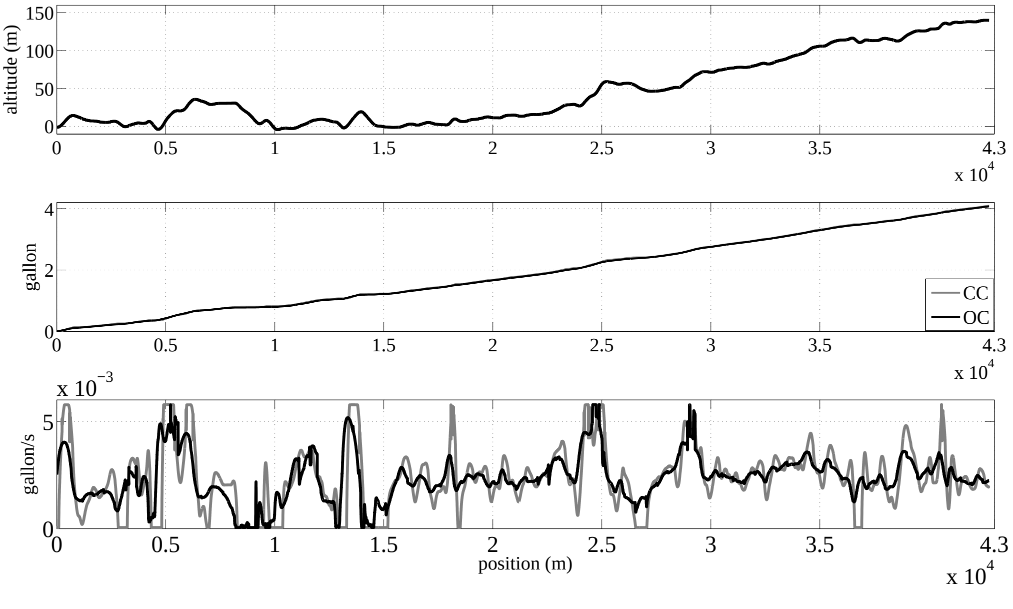


Figure 5.10: Comparison of fuel consumption and rate on R3 with 25 m/s set speed, OC vs. CC

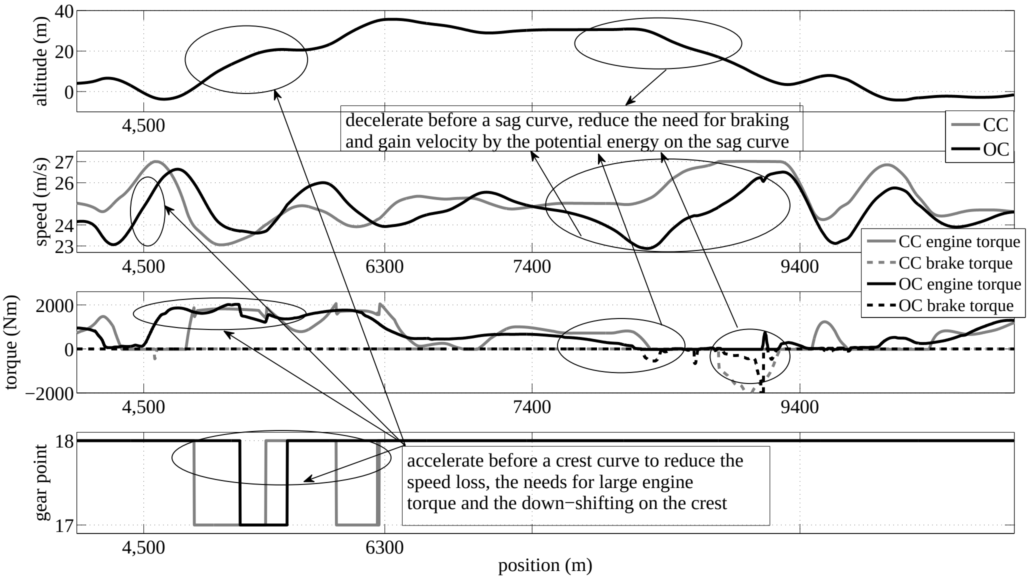


Figure 5.11: Zoomed-in comparison of velocity, torque, and gear on R3 with 25 m/s set speed, OC vs.CC

Figure 5.11 shows a section of the R3 tests, which illustrates the following basic functions of the OC:

- Accelerate the truck before a crest hill to reduce the speed loss and the need for large engine torque and frequent down-shifting on it, which is shown at 4.5-6.3 km. The engine torque is kept large before but not throughout the crest, where the CC keeps large torque.

- Require less downshifting than the CC to reduce fuel consumption and speed loss compared to the CC on the large crest hill, as shown at 4.5-6.3 km.

- Decelerate the truck before a sag hill in order to gain the velocity from potential energy and reduce the need for braking on the sag slope, which can be seen between 7.4 to 9.4 km (where the engine torque is reduced in front of a sag slope).

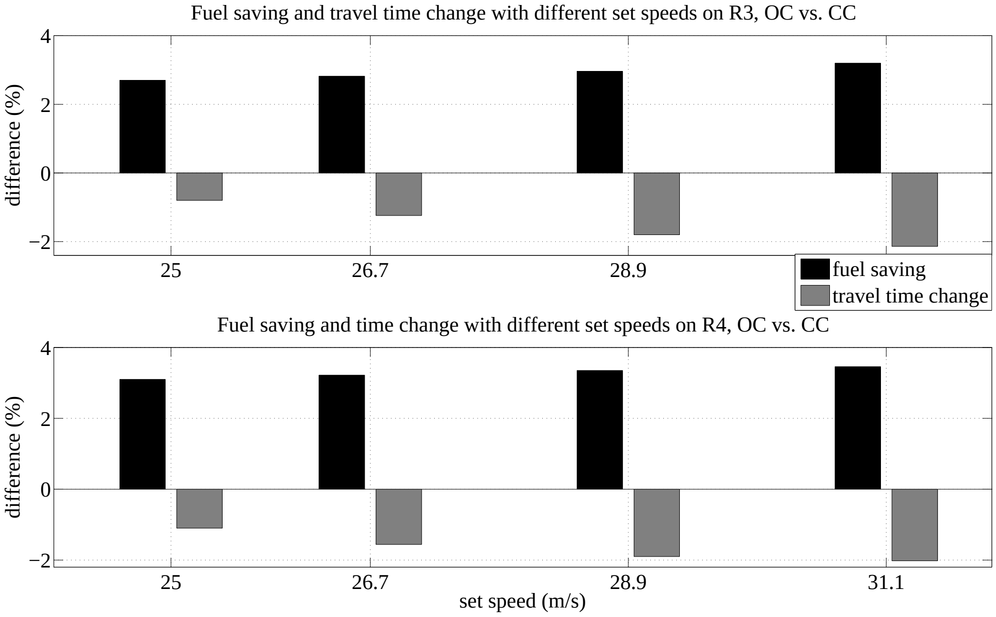


Figure 5.12: Fuel saving and travel time change with different set speeds, OC vs CC

5.3.2 Analyses of different set speeds for fuel savings

The reference speed thus far has been 25 m/s. However, it is also interested about how the change of the set speed effects the gain in fuel saving and the change of travel time from using the OC. Hence, more simulation tests are conducted on real roads R3 and R4, with different reference speeds, 25 m/s, 26.7 m/s, 28.9 m/s, and 31.1 m/s which correspond to 56, 60, 65 and 70 mph and are typical US highway speed limits.

The simulation results are depicted in Figure 5.12, where the gain in fuel saving and the change of travel time in percentages are plotted as dark and gray bars, respectively. The same situations can be found both on R3 and R4: the faster the set speed, the larger the gain in fuel saving (at the cost of increasing travel time). Hence, it could be beneficial to use the OC system with high driving speed if the travel time increase is acceptable.

In addition, the OC and CC system performance on R4 with the maximum set speed of 31.1 m/s is analyzed. The comparison of engine and brake torque, truck gear, speed, fuel rate, and total fuel consumption from the CC and OC is shown in Figures 5.13 and 5.14.

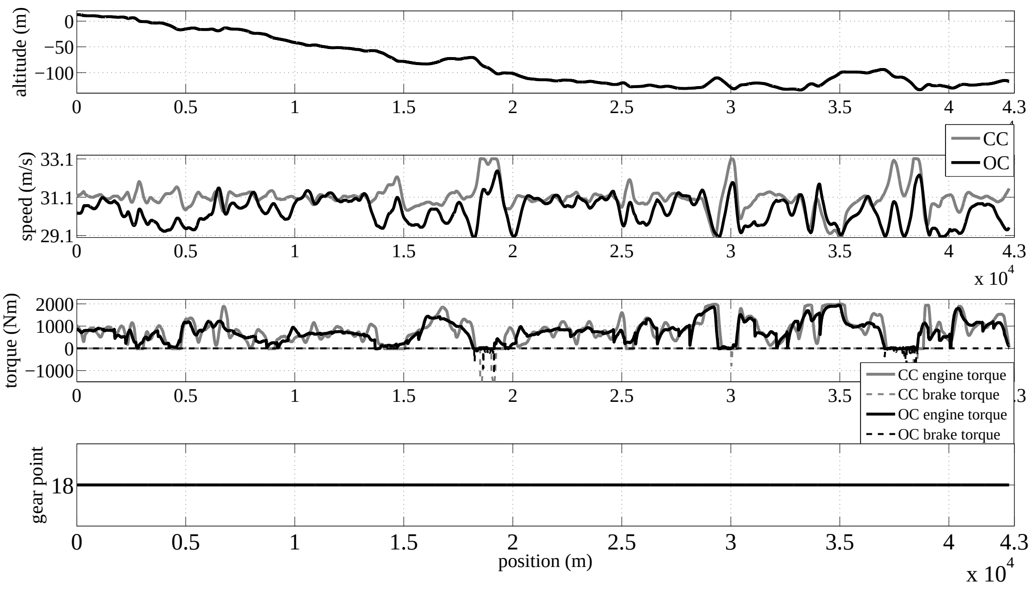


Figure 5.13: Comparison of velocity, torque, and gear on R4 with 31.1 m/s set speed, OC vs. CC

R4 is the reverse direction of R3, and a downhill terrain with negative mean road grade as indicated in Table 5.1. Therefore, although both systems perform the drive cycle and have a maximum speed of 33.1 m/s, the OC has larger speed loss than the CC in that it functions to reduce speed if there are sag slopes coming in, which appear more often in R4. Furthermore, it can be seen from the third plot in Figure 5.13, that on such a downhill terrain, the OC demands far less braking than the CC does and neither controller demands downshifting commands. Finally, as presented in the second plot of Figure 5.14, there is significant fuel savings by using the OC.

5.4 Real road test

Different from [5] and [8], the road test in this work is conducted to evaluate the OC performance on mountainous terrains R5 and R6. The road maps were shown previously in Figure 2.8. By testing the designed OC on R5, similar performance is obtained, as shown in Figure 5.15. It is seen that the OC requires large engine torque to climb a crest at 850

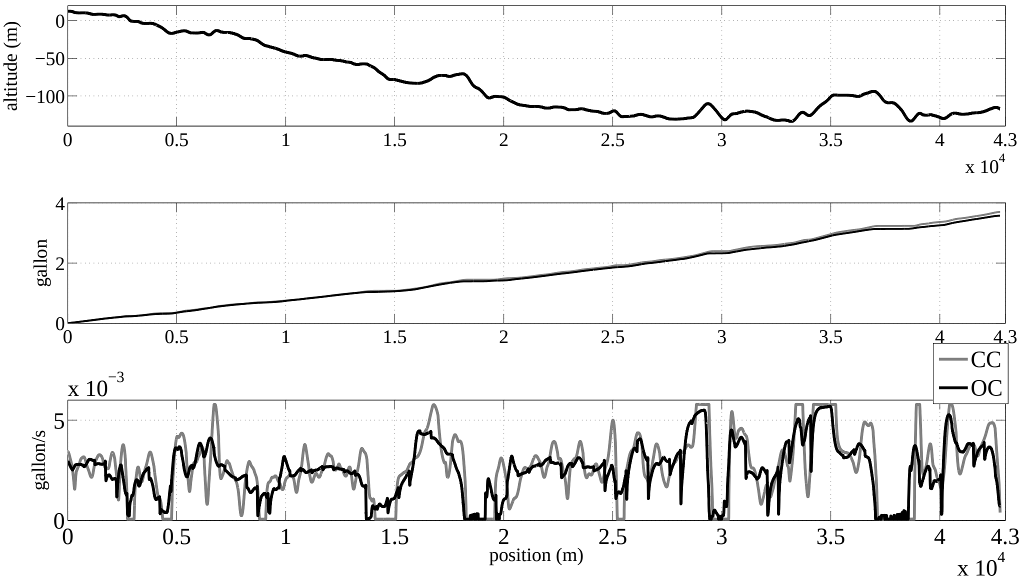


Figure 5.14: Comparison of fuel consumption and rate on R4 with 31.1 m/s set speed, OC vs. CC

m, which is earlier than the CC. Then the OC reduces the torque quicker than the CC while a sag slope is predicted at 3900 m. Additionally, the OC maintains a higher gear for this section than the CC, where the gray and black curves are for the CC and OC results, respectively.

The comparison of fuel consumption (F) and travel time (T) is shown in Table 5.4, which includes R5, and its two sections, S1 [0, 38000] m and S2 [38001, 54243] m. It is seen that for R5, the OC consumes a little more fuel (0.46%) than the CC does, while the OC is able to save 1% fuel on section S1.

Table 5.4: Comparison of fuel consumption and travel time for R5, and its sections

Route	mean grade	F_{CC}	F_{OC}	Diff(%)	T_{CC}(s)	T_{OC}	Diff(%)
R5	0.88 %	26.20	26.32	**-0.46**	2263.7	2257.2	**0.27**
R5(S1)	0.58 %	17.35	17.18	**1.00**	1364.0	1366.6	**-0.19**
R5(S2)	1.49 %	8.85	9.14	**-3.28**	899.7	890.6	**0.92**

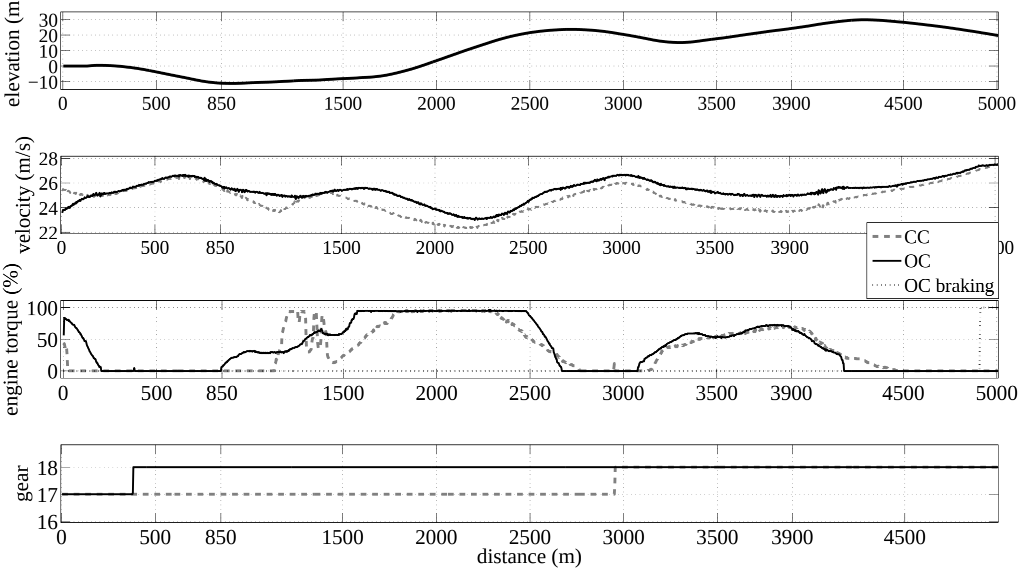


Figure 5.15: Comparison of velocity, engine torque, and gear position on a section of R5

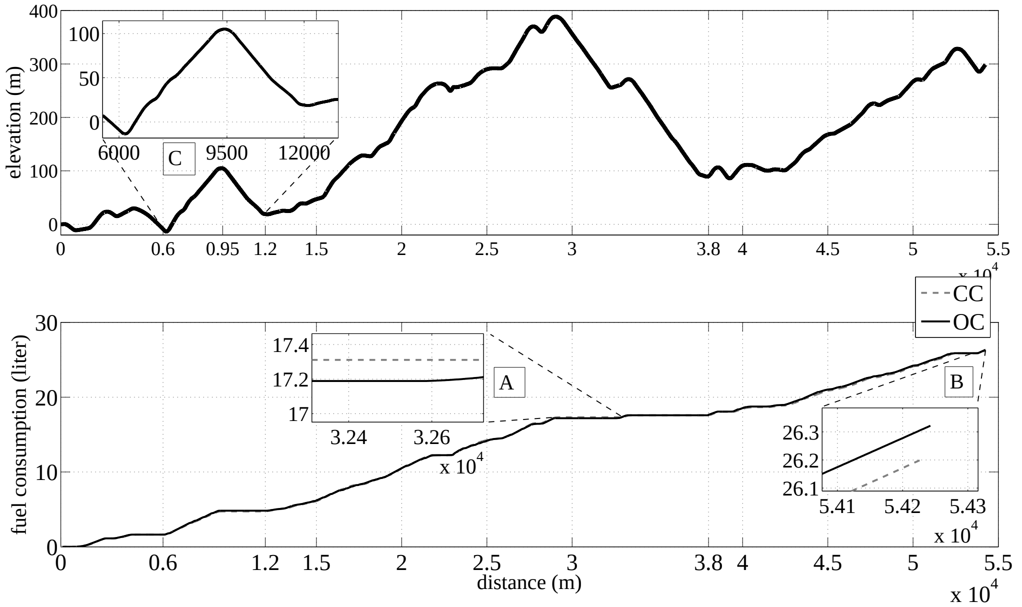


Figure 5.16: Comparison of fuel consumption w.r.t.road geometry on R5, CC vs.OC

Additionally the comparison of fuel consumption related to road geometry is shown in Figure 5.16. Fuel savings are not significant on S1 mainly because this section is highly mountainous, with a steep road grade and long slope length. An example is shown in the zoomed-in plot C in Figure 5.16, where the slope has maximum road grade of 5% and slope length longer than 3 km. To climb such a large crest at a “crawl” speed, the OC has to require full engine torque at a low gear, the same as the CC. Similarly, on a long sag slope, both the OC and CC control the truck to coast down, using an engine idling output of 3 kW and a corresponding engine idling fuel consumption. On section S2, the OC consumes more fuel than the CC, as shown in plot B in Figure 5.16. This is because S2 is nearly a constant uphill route, with a mean road grade of 1.49%, as shown in Table 5.4. For a long crest slope, the OC functions to accelerate the truck beforehand to reduce the speed loss, which consumes more fuel than the CC.

Similar to its functions on R5, the OC consumes a little more fuel (0.36%) than the CC on R6 where the OC is only able to save fuel consumption on some sections but consumes more on the others, which is shown in Table 5.5. However, it can be observed from the elevation map in Figure 5.17, there is an altitude difference of 300 m over R6 and the mean road grade is positive, so there is less room for reducing fuel consumption by decelerating before downhills.

Table 5.5: Comparison of fuel consumption and travel time for R6

Route	F_{CC}	F_{OC}	Diff(%)	T_{CC}(s)	T_{OC}	Diff(%)
R6	22.57	22.65	**-0.36**	2230.5	2220.7	**0.45**

By testing the designed OC system on rolling and mountainous terrains, the effect of the terrain change on the gain in fuel economy is investigated. It is seen that the OC is not really effective to save fuel consumption on a highly mountainous terrain, especially when the crest road is steep and slope length is long.

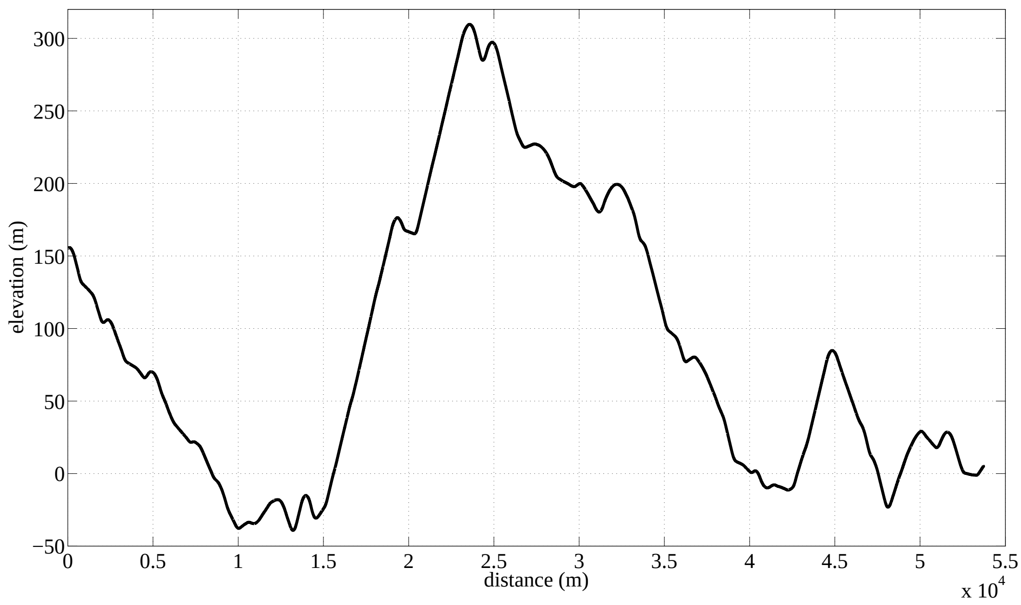


Figure 5.17: Elevation map of R6

5.5 Conclusion

In this chapter, simulation and real road tests were conducted to evaluate the performance of the designed OC system, compared to the CC. First, the simulated single hills were used to show that the basic functions of the OC fuel saving are from: accelerating the truck before a crest slope to reduce the speed loss, large engine torque, and downshifting on the crest; or decelerating before a sag slope and then gaining velocity from potential energy on the slope. Subsequently, real road geometries R1 to R4 were used in a high fidelity simulation environment to further validate the OC system performance and show the designed system is able to reduce fuel consumption up to 3.0% on level or rolling terrains with small increases in travel time, when compared to the CC. Finally, real road tests on routes R5 and R6 were conducted and the tests results were analyzed. It was seen that the OC could only save small amounts of fuel on highly mountainous terrain, which have a steep crest slope and long slope length. Thus, it was found that the gain in fuel economy is directly effected by the change in terrain type.

Chapter 6
Sensitivity Analysis of Road Map Errors

Thus far, the road maps applied in the OC system are considered without any errors, which means in Figure 4.9 the road map in the OC system perfectly matches the real road. However, this is not true for a real road test since road map errors could be generated from many resources such as elevation data acquisition, the map-matching system, road grade calculation, etc.

The road maps applied in this work are provided by Intermap which are commercial GIS road geometries and are part of a commercial data set that will be accurate nationwide. Additionally, the map matching system from Intermap is a commercial product and is of high accuracy. But it is still necessary to investigate error sensitivity to determine how the errors in the terrain data effect the gain in fuel economy and the system behavior. The results can be used to quantify the accuracy requirement for the road map to gain fuel economy.

6.1 Terrain Error Sensitivity

The terrain error sensitivity is investigated to find how errors in the terrain data affect the control system performance. In general, the GIS road map error is represented as a root mean square error (RMSE), which is the square root of the mean square error (MSE). MSE is obtained by calculating the square of the deviations of road map points in x_m, y_m, and z_m from their true position as in x, y, z, summing up the measurements and then dividing by the total number of points. For example in x direction:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(x_{mi} - x_i)^2}{n}} \tag{6.1}$$

The road map applied in this work is generated from the digital surface model (DSM) dataset by using Intermap's airborne Interferometric Synthetic Aperture Radar (IFSAR) technology. The map is a surface view of the earth containing both location and elevation information. The DSM dataset as well as the road map have the pixel size, horizontal accuracy RMSE, and vertical accuracy RMSE as 5 m, 2 m, and 1 m, respectively.

Intermap Technologies produces digital elevation models, digital surface model (DSM) products from our airborne Interferometric Synthetic Aperture Radar (IFSAR) technology. The DSM dataset as well as the road map have the step spacing, horizontal accuracy RMSE, and vertical accuracy RMSE as 5 m, 2 m, and 1 m, respectively.

In order to test the OC system performance with low accuracy maps, some error is injected into the road maps R1 - R6 which have been used in the previous chapter. Among those maps, R3 is taken as an example for the analysis and the results are deemed general for the reason that R3 is a typical rolling terrain. Some road profiles are produced by adding certain amounts of error to the original R3 map based on different error injection schemes. The errors considered in this work and their definitions are as follows:

- Absolute error (AE): a horizontal shift in the position of the slope
- Relative error (RE): a vertical difference of the slope value

Both AE and RE are general errors in real-road maps, where AE could be results of the accuracy limitations of the map-matching system and GPS receiver, and RE is due to the GIS elevation data acquisition or road grade calculation errors.

6.1.1 Different road maps

Several road profiles are produced by adding certain amounts of error to the original road geometry R3. In the simulation in this chapter, Intermap high accuracy road geometry is deemed and applied as the real road. The description of road and various road maps are given as follows:

- Real road R3r: real road of route 3
- Road map R3: Intermap road geometry of route 3, which is of high accuracy
- Road map R3.1: R3 with an AE and RE of 80 and 30 cm
- Road map R3.2: R3 with an AE and RE of 100 and 20 cm
- Road map R3.3: R3 with an AE and RE of 150 and 30 cm
- Road map R3.4: R3 with an AE and RE of 300 and 30 cm
- Road map R3.5: R3 with an AE and RE of 300 and 100 cm

In order to show the RMSE errors in a more straightforward way, R3.1 to R3.5 are subtracted by R3 to obtain an approximation of the vertical error in meters as descried in Table 6.1, where the mean, maximum, and minimum errors are listed in meters. The σ in this table represents the standard deviation of the slope error value. The comparison of vertical error distribution density is presented in Figure 6.1.

Table 6.1: Analysis of vertical errors in maps R3.1 to R3.5

map	mean(m)	max	min	σ
R3.1	-0.024	8.898	-5.716	0.776
R3.2	0.016	3.234	-9.793	1.053
R3.3	1.016	4.234	-8.793	1.203
R3.4	0.033	6.467	-19.586	2.106
R3.5	0.049	9.701	-29.380	3.159

The comparison of R3 (dark), R3.2 (gray), and R3.5 (dashed gray) is shown in Figure 6.2, where the top plot is the elevation, the middle plot is the road grade, and the bottom plot is the zoomed-in road grade comparison. From the figure, it can be seen that the AE results in the hills in R3 shifting in position either to the left or right. On the other hand, by adding RE the elevation data changes and thereby road grades from R3.2 and R3.5 are different from R3, among which R3.5 has the most significant errors.

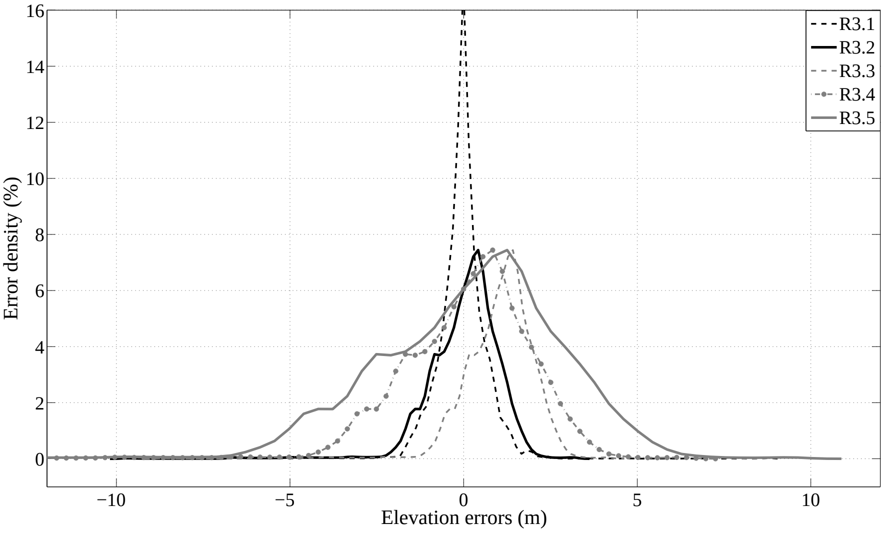


Figure 6.1: Comparison of vertical error distribution density, R3.1 to R3.5

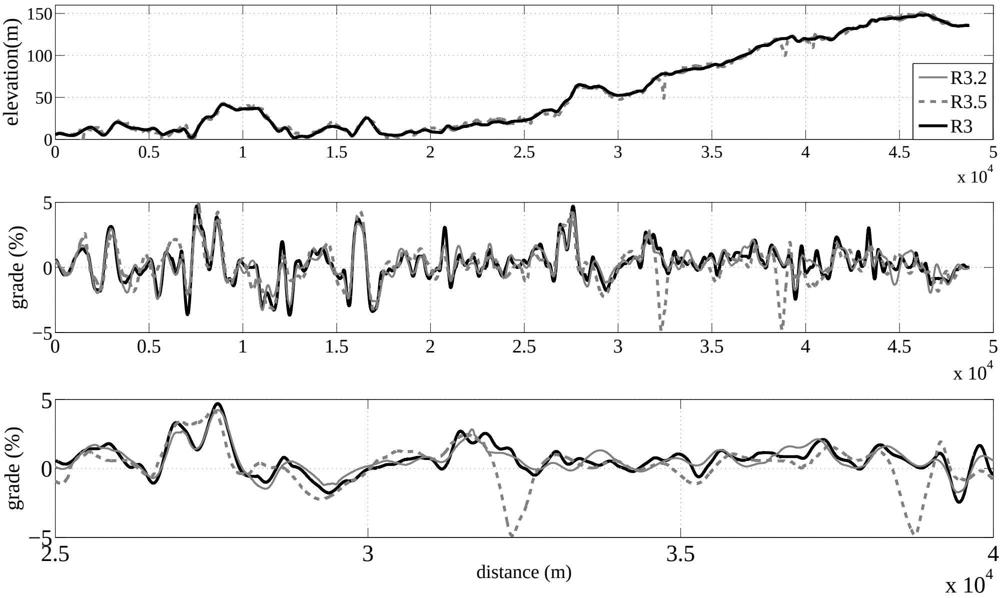


Figure 6.2: Comparison of road maps with different accuracies, R3 vs. R3.2 and R3.5

In addition, road maps R3.4 and R3.5 have similar AE of 3 m, which is close to the horizontal position accuracy RMSE (4 m) of the commercial GPS receivers applied in the real road test. Hence, it is reasonable to see the OC simulation test using R3.4 as a real road test, since in this simulation only the road map errors are taken into account which has the RMSE accuracy similar to that of the GPS receiver. The errors in different tests can be compared in the following:

- Horizontal error of simulation test using R3.4: AE (3 m) on the top of R3;
- Horizontal error of real road test using R3: horizonal GPS RMSE (4 m) on top of R3.

6.1.2 Simulation setup

For the purpose of numerically quantifying the impact from the road map errors on the fuel savings from using road maps R3 and R3.1 to R3.5, the following steps are conducted for the simulation:

1. Run the CC on real road R3r, where the total fuel usage is set as the fuel consumption baseline (BL);
2. Run the OC based on R3: fuel usage is compared to the BL and then the gain in fuel saving using the high accuracy map is obtained, which is the fuel saving baseline (FSBL);
3. Run the OC based on R3.1 to R3.5: each fuel usage is compared to the BL and the fuel savings by using low accuracy maps are saved;
4. Each fuel saving from step 3 is compared to the fuel saving baseline in step 2, and the differences in fuel saving by using maps R3 and R3.1 to R3.5 are found.

This simulation procedure can be seen in Fig. 6.3.

6.1.3 Simulation results

In this section, the simulation tests for the terrain error sensitivity are conducted and results are analyzed. The truck speed is set at 25 m/s, and all the OC simulations obtain nearly the same travel time. Different fuel usages from the OC using different road maps,

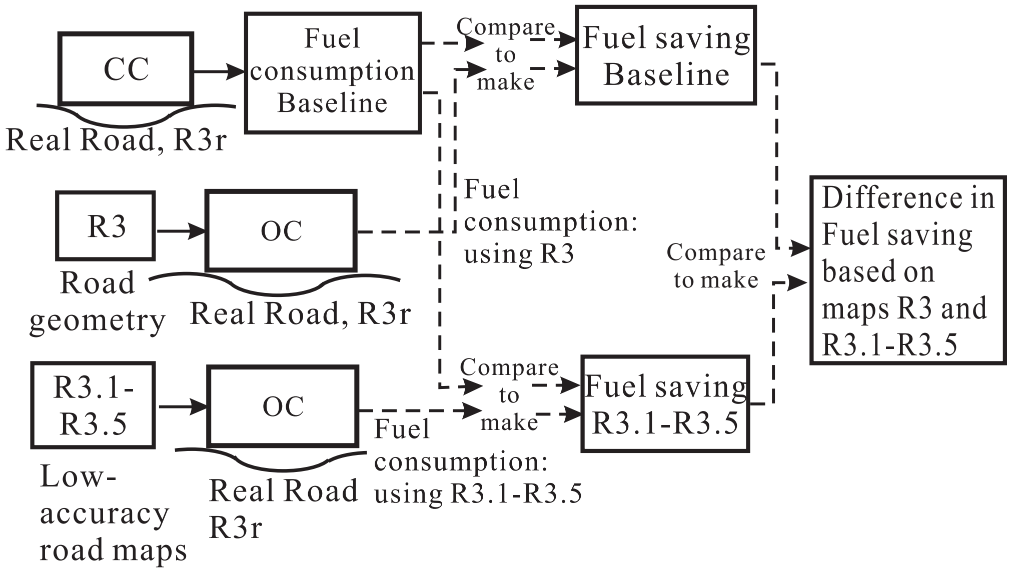


Figure 6.3: Block diagram of simulation setup for terrain error sensitivity analysis

R3 and R3.1 - R3.5, are compared with the BL. The gains of fuel savings with respect to different road maps are then obtained, and illustrated in Figure 6.4, where the dark bars represent the fuel saving. The gray bars represent the relative difference in fuel saving, which are calculated by subtracting the FSBL by fuel saving using R3.1 to R3.5. The first dark bar in the figure corresponds to the FSBL. In general, the gain in fuel savings is directly related to the road map accuracy (i.e., the more accurate the road map, the larger the gain in fuel savings).

Additionally, as indicated in Figure 6.4, the fuel savings from R3.1 to R3.4 are all larger than 2.0%, while for R3.5 the gain in fuel saving is less than 1% although it has the same AE as R3.4. This can be explained by the fact that R3.5 has the largest RE (1 m) among all road profiles which can be seen looking back to Section 6.1.1. Therefore, it can be concluded that RE has a larger impact than AE on the OC fuel saving.

The reason the fuel saving is not largely impacted by AE is that a couple of meters shifting in the slope position is insignificant when compared to the general kilometer slope length as well as the 25 meters sampling distance. However, if there is a large positive RE presented in the map which increases the road grade, the OC could command larger

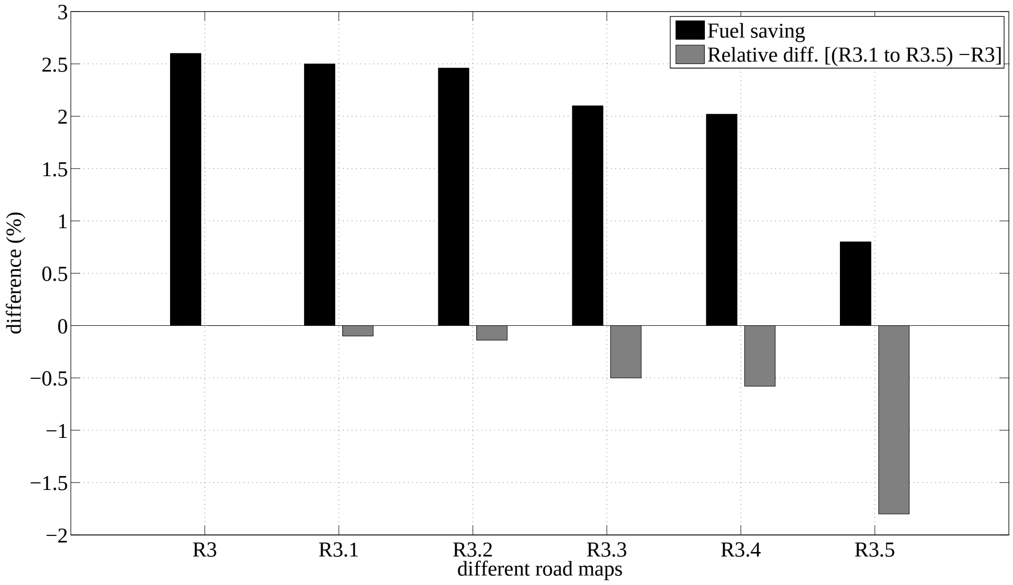


Figure 6.4: Fuel saving using different road maps: R3 and R3.1-R3.5

engine torque and unnecessary downshifting to climb the presumed steeper crest. On the other hand, if a large negative RE appears in the map which functions to decrease the road grade, the OC will command larger speed reduction or braking. It would require large torque compensation from the feedback velocity tracking subsystem in real driving to maintain the optimal speed trajectory. Both would result in large fuel usage and consequently reduce the gain in fuel savings.

Furthermore, as discussed in Section 6.1.1, the OC simulation test using R3.4 has similar horizontal accuracy to the real road test where the applied GPS receiver has RMSE of approximately 4 m. As demonstrated in Figure 6.4, the OC using R3.4 is able to gain 2.0% fuel saving, which additionally verifies the OC is still able to gain fuel saving on the real road test even if there are horizontal errors in the GPS receiver and road map.

Other than effecting the gain in fuel economy, road map errors also impact the OC system performance and make it function incorrectly. The comparison of the OC performances by using different road maps, in terms of v, u, and g in a section from 35 to 45 km, is given in Figure 6.5. By using R3.2 and R3.5, the OC system could not function fully correctly if the following errors occurred:

- Negative RE (road grade decreased): Results in the deceleration of the truck and reduction in engine torque before a false sag hill in the map. Since on a real road trucks can not gain the speed from potential energy on the sag, large engine torque and downshifting are commanded to accelerate the truck back to the set speed, which is displayed between 36 to 38 km;
- Positive RE (road grade increased): Results in the acceleration of the truck before a fake crest hill by demanding large engine torque, which increases the truck speed in real road, as can be seen between 43 to 44 km.

Those system functions produce unpredicted large fuel consumption in the OC system and therefore reduce the fuel saving.

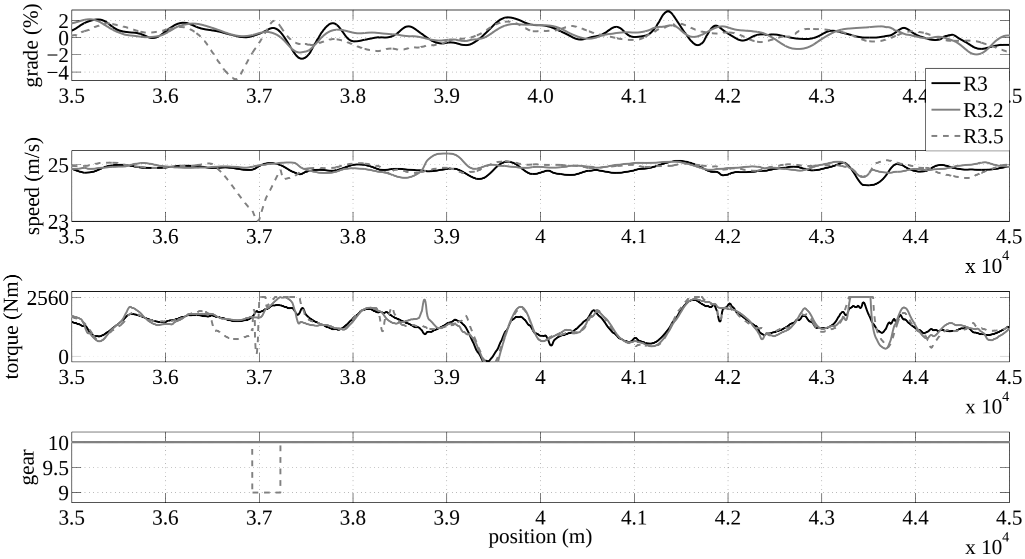


Figure 6.5: Zoomed-in comparison of the OC performances by using maps, R3, R3.2, and R3.5

From the simulation results above, it is clear that road map errors largely effect both the gain in fuel saving and the system performance. Because correct implementation of the OC highly relies on the prediction road maps in real time, the road maps are required to have high accuracy.

6.2 Conclusion

This chapter described results of the terrain map error sensitivity analyses. The changes of fuel economy and system performance with respect to road maps with different errors were discussed. It was found that the road map error effects both the fuel economy and the system performance. The impact from the absolute error (a shift in the slope position) on the fuel saving is not as evident as that from the relative error (a different slope value). Even if the impact from absolute error on the gain in fuel saving is not evident, this impact on system performance is significant. Therefore, in real-time implementation, road maps are required to have small absolute and relative errors.

Additionally, this chapter has shown that by using road maps with the absolute error close to the GPS receiver error accuracy, the OC simulation emulates a real road test in terms of horizontal error accuracy. The test results show that the OC is still able to gain acceptable fuel saving, which verifies the OC could function in a reasonable manner even if horizontal errors in the GPS receiver and road map are fixed to present accuracies in real road scenarios.

Chapter 7

Conclusion

7.1 Concluding Remarks

This work described the research to investigate the benefit of using a 3D road geometry based optimal powertrain control (OC) system in reducing the heavy truck fuel consumption. The OC was designed using a nonlinear programming solver to predict truck engine or brake torque, gear shifting, and velocity trajectory based on road geometry to minimize fuel consumption and travel time. Computer simulations and real road tests of a Class 8 truck were conducted with Intermap's real 3D road geometry to evaluate the OC system performance.

Chapter 1 presented the motivation and objectives of conducting this work. Additionally, the background and literature were studied to cover not only the closely related topics such as fuel optimal vehicle trajectory optimization and real-time fuel optimal control system design, but some more general theoretical aspects as well, i.e., optimal control problems, mixed-discrete nonlinear programming (NLP) solution, real-time optimal control system design. After analyzing and comparing the prior work, the contributions of this work were given at the end of Chapter 1.

Real and simulated road geometries and the comparison baseline to evaluate the OC performance were given in Chapter 2. First, some important road design parameters, e.g., road grade, terrain type, frequency component, were defined to describe the main features of a road profile and to generate the simulated road section. Second, by using the simulated road profiles, the relation between the fuel consumption and road geometry was be analyzed. Furthermore, the features of real road geometries R1 to R6 were introduced, which include the calculation of road grade from 3D elevation data, the illumination of real road elevation/grade maps, and the analysis of terrain type. As a final step, a cruise controller (CC) with the automatic braking function was applied to perform the defined drive cycle, generate the fuel consumption baseline, and evaluate the designed OC system performance.

Chapter 3 described the work to develop and validate heavy truck dynamics and powertrain models, for the purpose of designing the model-based OC system and testing the system performance in a simulation environment. Only longitudinal truck dynamics were considered and the applied engine map, Brake Specific Fuel Consumption (BSFC) map, and truck parameters were provided by Eaton Corp. The model was validated by comparing the real road test data with the simulation data. In addition, simulations were run by using the developed model to quantify how the change in terrain type effects the truck fuel consumption and show that fuel consumption is impacted by the change in terrain type, driving speed, and road frequency component.

In Chapter 4, a 3D GIS road geometry based OC system was designed to reduce the heavy truck fuel consumption and travel time. A direct collocation method was applied to transform the optimal control problem into a mixed-discrete nonlinear programming (MDNLP) problem to find the constrained minimum of a nonlinear scalar function of several variables such as the normalized truck throttle command u, brake command b, gear point g, and velocity v. An optimizer of the OC which uses an interior-point algorithm plus a rounding-off method was designed to solve this MDNLP problem. The real time operation system consists of various modules, including the map-matching system and the OC with the MDNLP solver. The OC is a look-ahead controller, which solves the optimization problem for a 3000 m prediction horizon ahead, and performs the optimal velocity trajectory by applying the calculated torque and optimal gear.

Simulation and real road tests were presented in Chapter 5 to evaluate the performance of the designed OC system, compared to the CC. First, the simulated single hills were used to show that the basic functions of the OC fuel saving are to accelerate the truck before a crest slope or decelerate it before a sag slope. Subsequently, real road geometries R1 to

R4 were used in a high fidelity simulation environment to further validate the OC system performance and show the designed system is able to reduce fuel consumption up to 3.0% on level terrains (R1 and R2) or rolling terrains (R3 and R4) with small travel time increases. Finally, the real road tests on routes R5 and R6 were conducted and test results denoted that the OC could only save a small amount of fuel on highly mountainous terrain, which has the steep crest slope and long slope length. Thus, it was found that the gain in fuel economy is directly effected by the change in terrain type.

Finally, in Chapter 6 the terrain error sensitivity analysis, i.e., how the errors in the road map impact the OC fuel saving and system performance, was investigated. It was found that the road error effects both the fuel economy and the system performance. Some conclusion could be: the impact from the absolute error (a shift in the slope position) on the fuel saving is not as evident as that from the relative error (a different slope value); even if the impact from absolute error on the gain in fuel saving is not evident, this impact on system performance is significant such as commanding incorrect engine torque, gear shifting, velocity trajectory, etc. Therefore, in real-time implementation, road maps are required to have small absolute and relative errors.

7.2 Future Work

7.2.1 Modification of the optimal control system

Future work would consider modifying the OC for a better real-time implementation, and testing the system on a wide range of terrain types such as conducting road tests on typical level and rolling terrains. Additionally, it is proposed to set up a Monte Carlo simulation where the road elevation varies to predict a mean saving on three terrain types, as well as to simulate and predict the fuel saving on any specific routes. For instance, by setting up a corresponding Monte Carlo simulation for a route which has 45% level, 35% rolling, and 20% mountainous terrains, it would be possible to forecast the fuel saving on this route by using the designed road geometry based optimal control system.

7.2.2 Design of an optimal power management system for a hybrid electric truck

More crucially, the design of a road map based hybrid electric truck fuel optimization system would be considered as an meaningful extension of this work, since knowledge of the road information ahead could be helpful to improve the hybrid electric truck's fuel efficiency.

In general, the hybrid electric truck produced by Eaton employs a parallel-type diesel-electric architecture which integrates an advanced diesel engine, an electric traction motor, Lithium-Ion batteries, an automatic clutch, and an automated manual transmission system. The motor is directly linked between the output of the clutch and the input to the transmission. This architecture provides the regenerative braking during deceleration and allows efficient motor assist and recharge operations by the engine [34].

Hence, in future work the optimal power management system (OPM) of the hybrid electric truck's powertrain could be developed to command the engine torque, motor torque, and gear selection to minimize the hybrid electric truck's fuel consumption using 3D road geometries. The gain in fuel economy using the OPM could be quantified when compared to the baseline control which could be a classic load leveling type rule-based algorithm implemented on the same hybrid electric truck. The idea of this approach is based on the concept of "load-leveling", which attempts to operate the irreversible energy conversion device such as an internal combustion engine in an efficient region and uses the reversible energy storage device as a load leveling device, to compensate the rest of the power demand. Finally, the proposed OPM would have the following basic functions:

- Determine the suitable operation modes: motor only, engine only, and power assist.
- Command the proper torque level and its split between the motor and engine, and maintain adequate energy in the battery.
- Decelerate the truck before a downhill by reducing engine/motor torque in order to gain the velocity from potential energy. The OPM recharges the battery when the

truck is coasting down from the downhill. If necessary, the regenerative braking is commanded by applying a negative torque to the electric motor.

- Accelerate the truck and require downshifting before a uphill in order to reduce the speed loss and the need for large engine torque on the hill. The engine torque is kept large before but not throughout the hill.

Bibliography

[1] A. B. Schwarzkopf and R. B. Leipnik, “Control of highway vehicles for minimum fuel consumption over varying terrain,” *Transportation Research*, vol. 11, no. 4, pp. 279 –286, 1977.

[2] D. Chang and E. Morlok, “Vehicle speed profiles to minimize work and fuel consumption,” *Journal of transportation engineering*, vol. 131, no. 3, pp. 173 –181, 2005.

[3] M. Ivarsson, J. Åslund, and L. Nielsen, “Optimal speed on small gradients - consequences of a non-linear fuel map,” The International Federation of Automatic Control, Seoul, Korea, 2008.

[4] J. N. Hooker, “Optimal driving for single-vehicle fuel economy,” *Transportation Research*, vol. 22A, no. 3, pp. 183 –201, 1988.

[5] F. Lattemann and K. Neiss, “The predictive cruise control - a system to reduce fuel consumption of heavy duty trucks,” *SAE Paper* , no. 2004-01-2616, 2004.

[6] E. Kozica, “Look ahead cruise control: road slope estimation and control sensitivity,” M.Sc. Thesis, KTH - Royal Institute of Technology, Stockholm, Sweden, 2005, iR-RT-EX-0524.

[7] E. Hellström, “Explicit use of road topography for model predictive cruise control in heavy trucks,” M.Sc. Thesis, Linköping University, 2005.

[8] E. Hellström, M. Ivarsson, J. Åslund, and L. Nielsen, “Look-ahead control for heavy trucks to minimize trip time and fuel consumption,” *Control Engineering Practice*, vol. 17, no. 2, pp. 245 –254, 2009.

[9] P. I. Barton, R. J. Allgor, W. F. Feehery, and S. Galán, “Dynamic optimization in a discontinuous world,” *Ind. Eng. Chem. Res.*, vol. 37, pp. 966 –981, 1998.

[10] R. Bellman, *Dynamic Programming.* Princeton, NJ: Princeton University Press, 1957.

[11] R. Luus, “Application of dynamic programming to high-dimensional nonlinear optimal control problems,” *Int. J. Control*, vol. 52, pp. 239 –250, 1990.

[12] E. Hellström, M. Ivarsson, J. Åslund, and L. Nielsen, “Look-ahead control for heavy trucks to minimize trip time and fuel consumption,” Fifth IFAC Symposium on Advances in Automotive Control, Monterey, CA, USA, 2007.

[13] A. E. Bryson and Y. Ho, *Applied Optimal Control.* New York: Hemisphere, 1975.

[14] J. T. Betts, “Survey of numerical methods for trajectory optimization,” *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193 –207, 1998.

[15] D. Kraft, “On converting optimal control problems into nonlinear programming problems,” *Comput. Math. Prog.*, vol. 15, pp. 261 –280, 1985.

[16] T. H. Tsang, D. M. Himmelblau, and T. F. Edgar, “Optimal control via collocation and non-linear programming,” *Int. J. Control*, vol. 21, pp. 763 –768, 1975.

[17] D. G. Hull, “Conversion of optimal control problems into parameter optimization problems,” *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 1, pp. 57 –60, 1997.

[18] C. R. Hargraves and S. W. Paris, “Direct trajectory optimization using nonlinear programming and collocation,” *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 4, pp. 338 –342, 1987.

[19] F. Fahroo and I. M. Ross, “Direct trajectory optimization by a chebyshev pseudospectral method,” American Control Conference, Chicago, IL, USA, 2000.

[20] Q. Gong, I. M. Ross, W. Kang, and F. Fahroo, “Connections between the covector mapping theorem and convergence of pseudospectral methods for optimal control,” *Computational Optimization and Applications*, vol. 41, pp. 307 –335, 2008.

[21] P. T. Boggs and J. W. Tolle, “Sequential quadratic programming for large-scale nonlinear optimization,” *Journal of Computational and Applied Mathematics*, vol. 124, no. 1-2, pp. 123–137, 2000.

[22] B. R. Geiger, J. F. Horn, A. M. DeLullo, L. N. Long, and A. F. Niessner, “Optimal path planning of uavs using direct collocation with nonlinear programming,” AIAA Guidance, Navigation, and Control Conference, Keystone, Colorado, USA, 2006.

[23] J. D. Schwartz and M. Milam, “On-line path planning for an autonomous vehicle in an obstacle filled environment,” IEEE Conference on Decision and Control, Cancun, Mexico, 2008.

[24] M. J. D. Powell, “A fast algorithm for nonlinearly constrained optimization calculations,” *Numerical Analysis*, pp. 144–157, 1978.

[25] R. H. Byrd, M. E. Hribar, and J. Nocedal, “An interior point method for large scale nonlinear programming,” *SIAM Journal on Optimization*, vol. 9, pp. 877 –900, 1999.

[26] R. H. Byrd, J. C. Gilbert, and J. Nocedal, “A trust region method based on interior point techniques for nonlinear programming,” *Mathematical Programming*, vol. 89, no. 1, pp. 149 –185, 2000.

[27] B. Mettler and Z. Kong, “Receding horizon trajectory optimization with a finite-state value function approximation,” American Control Conference, Seattle, Washington, USA, 2008.

[28] M. W. Huang and J. S. Arora, "Optimal design with discrete variables: Some numerical experiments," *International Journal for Numerical Methods in Engineering*, vol. 40, pp. 165 –188, 1997.

[29] B. Sharif, G. Wang, and T. ElMekkawy, "Mode pursuing sampling method for discrete variable optimization on expensive black-box functions," *Transactions of ASME, Journal of Mechanical Design*, vol. 130, no. 2, pp. 1 –11, 2008.

[30] M. Bremicker, P. Y. Papalambros, and H. T. Loh, "Solution of miced-discrete structural optimization problems with a new sequential linearizaiotn algorithm," *Computers and Structures*, vol. 37, no. 4, pp. 451 –461, 1990.

[31] H. B. P. Abichandani and M. Kam, "Multi-vehicle path coordination under communication constraints," American Control Conference, Seattle, Washington, USA, 2008.

[32] O. Exler and K. Schittkowksi, "A trust region sqp algorithm for mixed-integer nonlinear programming," *Optimization Letters*, vol. 3, no. 1, pp. 269 –280, 2007.

[33] G. van de Braak, M. J. Bünner, and K. Schittkowski, "Optimal design of electronic components by mixed-integer nonlinear programming," *Optimization and Engineering*, vol. 5, pp. 271 –294, 2004.

[34] C. C. Lin, H. Peng, J. W. Grizzle, J. Liu, and M. Busdiecker, "Control system development for an advanced-technology medium-duty hybrid electric truck," SAE, Fort Worth, TX, USA, 2003.

[35] B. Wu, C. Lin, Z. Filipi, H. Peng, and A. D., "Optimal power management for a hydraulic hybrid delivery truck," *Vehicle System Dynamics*, vol. 42, no. 1-2, pp. 23 –40, 2004.

[36] E. F. Camacho and C. Bordons, *Model predictive control.* London: Springer, 1999.

[37] J. Albersmeyer and D. Beigel and C. Kirches and L. Wirsching and H. G. Bock and J. P. Schlöder, *Fast Nonlinear Model Predictive Control with an Application in Automotive Engineering.* Berlin/Heidelberg: Springer, 2009.

[38] J. Strizzi, I. M. Ross, and F. Fahroo, "Towards real-time computation of optimal controls for nonlinear systems," AIAA Guidance, Navigation, and Control Conference, Monterey, CA, USA, 2002.

[39] Q. Gong, W. Kang, N. S. Bedrossian, F. Fahroo, P. Sekhavat, and K. Bollino, "Pseudospectral optimal control for military and industrial applications," IEEE Conference on Decision and Control, New Orleans, LA, USA, 2007.

[40] C. C. Lin and H. Peng, "Optimal adaptive cruise control with guaranteed string stability," *Vehicle System Dynsmics*, vol. 31, pp. 313 –330, 1999.

[41] M. Vögel, O. von Stryk, R. Bulirsch, T. M. Wolter, and C. Chucholowski, "An optimal control approach to real-time vehicle guidance," *In: W. Jäger, H.-J. Krebs (eds.): Mathematics, Key Technology for the Future, Springer-Verlag*, pp. 84 –102, 2003.

[42] A. Haj-Fraj and F. Pfeiffer, “Optimal control of gear shift operations in automatic transmissions,” *Journal of the Franklin Institute*, vol. 338, no. 2-3, pp. 371 –390, 2001.

[43] American association of state highway and transportation officials, *A policy on geometric design of highways and streets, fourth edition*, 2001.

[44] A. Savitzky and M. J. E. Golay, “Smoothing and differentiation of data by simplified least squares procedures,” *Analytical Chemistry*, vol. 36, no. 1, pp. 1627 –1639, 1964.

[45] Certification Division, “Federal test procedure review project,” Office of Mobile Sources, Office of Air & Radiation, U.S. Environmental Protection Agency, Preliminary Technical Report EPA 420-R-93-007, May 1993.

[46] A. Bjrck, *Numerical Methods for Least Square Problem.* Philadelphia: SIAM, 1996.

[47] W. Gutkowski, *Discrete structural optimization: design problems and exact solution methods.* New York: Springer-Verlag, 1997.

[48] S. Kitayama, M. Arakawa, and K. Yamazaki, “Penalty function approach for the mixed discrete nonlinear problems by particle swarm optimization,” *Structural and Multidisciplinary Optimization*, vol. 32, no. 3, pp. 191 –202, 2006.

[49] J. Renegar, *A Mathematical View of Interior-Point Methods in Convex Optimization.* Philadelphia: MPS-SIAM, 2001.

[50] G. Zhou, “A modified sqp method and its global convergence,” *Journal of Global Optimization*, vol. 2, pp. 193–205, 1997.

[51] F. Allgöwer, T. A. Badgwell, and J. S. Qin, “Nonlinear predictive control and moving horizon estimation - an introductory overview,” *In P. M. Frank, editor, Advances in Control, Highlights of ECC’99*, pp. 391 –449, 1999.

Appendix A

Abbreviations and frequently used symbols

A.1 List of Abbreviations

Abbreviation	Description
ACC	Adaptive cruise control
AE	Absolute error: a horizontal shift in the curve position
BL	Baseline
BSFC	Brake specific fuel consumption
CAN	Controller area network
CC	Cruise control system
DP	Dynamic programming
DSM	Digital surface model
ECU	Electronic control uint
EH	Electric horizon, name of a map matching system
GIS	Geographic information system
GPS	Global positioning system
INS	Initial navigation system
IP	Interior point method
PCC	Predictive cruise control
MDNLP	Mixed-discrete nonlinear programming
MINLP	Mixed-integer nonlinear programming
MPC	Model predictive control
NLP	Nonlinear programming
OC	Optimal control system
RE	Relative error: a vertical different slope value for the curve
rpm	Revolutions per minute
SQP	Sequential quadratic programming
TPBVP	Two-point boundary value problem

A.2 List of frequently used symbols

The experimental truck is a Caterpillar Class 8 tractor and trailer. Frequently used symbols as well as those truck parameters applied in this work are listed in the following table.

Symbol	Description	Value	Unit
A_{fr}	Truck frontal area	7.43	m^2
b	Braking pedal position in percentage		%
C_d	Aerodynamic drag coefficient	0.65	-
C_{rr}	Rolling resistance coefficient	0.0072	-
F_w	Wheel drive force		N
F_s	Longitudinal force due to road grade		N
F_{rr}	Rolling resistance force		N
F_a	Air drag force		N
g	Truck gear	18	-
h	Step length	25	m
L	Prediction horizon	3000	m
m	Truck mass	31.7	tons
m_f	Fuel consumption		liter
n_t	Transmission ratio		-
n_d	Final drive ratio	3.25	-
p	Position		meter
P	Engine power		kW
J_e	Engine rotation inertia	3.0	kg-m^2
J_w	Wheel rotation inertia	3.21	kg-m^2
r	Wheel radius	0.51	m
s	Number of set points	120	-
T_m	Maximum engine torque	2509	Nm
T_e	Desired engine torque		Nm
T_w	Wheel torque		Nm
T_{bm}	Maximum engine retarder brake torque	4500	Nm
T_b	Desired braking torque		Nm
u	Throttle percentage position		
v	Truck velocity		m/s
ϕ	Road grade		%
ρ_{air}	Air density	1.23	kg/m^3
ω_e	Engine rotation speed		rpm
ω_w	Wheel rotation speed		rad/s
η_t	Transmission efficiency	0.985	-
η_d	Final drive efficiency	0.97	-
t	Time		s

Appendix B

Implementation of optimal control system for real road test

The real time test system which consists of various modules, including the map-matching system and the OC with the NLP solver, has been discussed in Section 4.2 and is shown in Figure B.1 again. In this chapter, the detailed system implementation for each module either in Software module 1 (S1) or 2 (S2) is given.

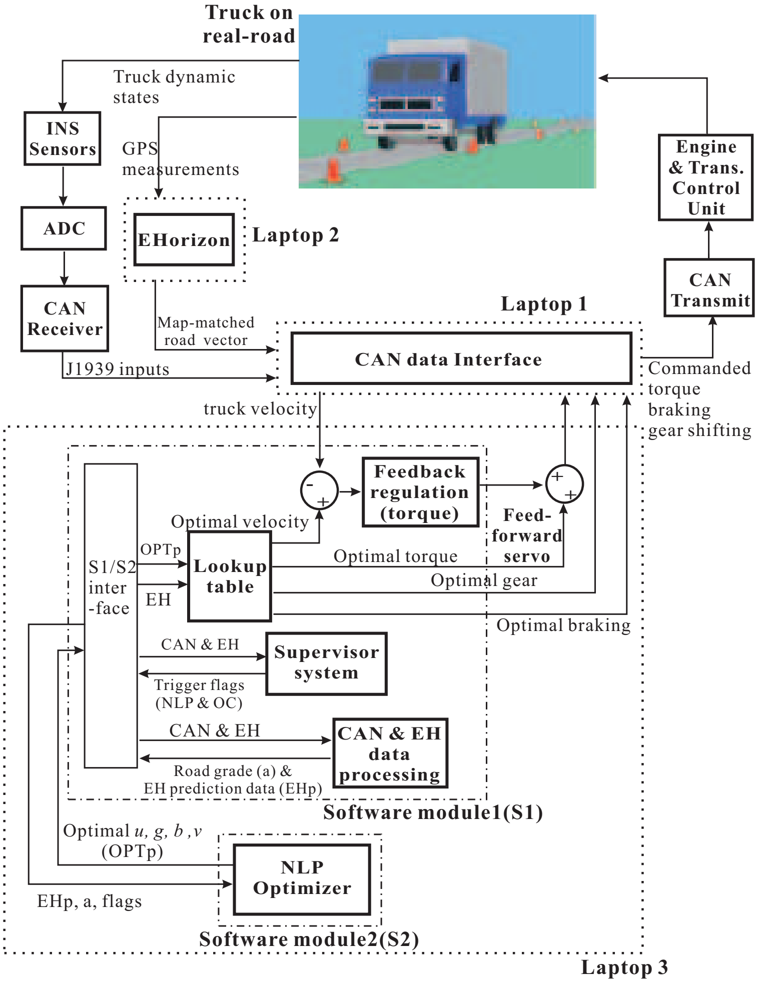


Figure B.1: Real time experimental system setup

The complete system scheme of S1 in Laptop 3, which communicates with Laptop 1 by CAN data interface, is presented in Figure B.2. It is seen that S1 acquires EH road geometry and J1939 vehicle states data from a CAN receive block, and sends general commands and engine control commands to a CAN transmitter. Meanwhile, S1 exchanges data with S2, where the NLP optimizer runs to calculate the optimal solution. In the following, the scheme and functions of S1 and S2 are given in detail.

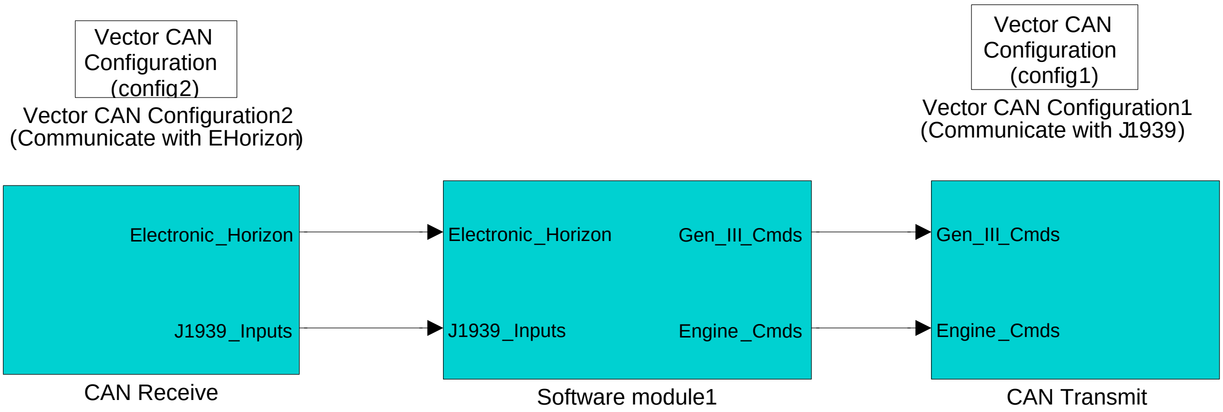

Vector CAN Configuration (config2)
Vector CAN Configuration2 (Communicate with EHorizon)
Vector CAN Configuration (config1)
Vector CAN Configuration1 (Communicate with J1939)
Electronic_Horizon
J1939_Inputs
CAN Receive
Electronic_Horizon
J1939_Inputs
Gen_III_Cmds
Engine_Cmds
Software module1
Gen_III_Cmds
Engine_Cmds
CAN Transmit

Figure B.2: Complete system scheme of S1, and CAN receive/transmit blocks

B.1 Software module 1

When looking into the S1 block, the system structure is displayed in Figure B.3, which is made up of the vehicle speed control enabler, real-time optimal control (OC) system, feedback regulation, and engine torque management blocks. Vehicle speed control enabler block reads truck speed from J1939 data and generates the output 'Mas_Allowed', which has a value of 1 to indicate the OC is turned on if truck speed is larger than 20 m/s. The OC block acquires both vehicle and EH data, and generates and outputs both the optimal solution and override commands, where the former includes 'Setspeed' (optimal speed trajectory), optimal torque, and truck gear and the latter are generated to override the ECU with the optimal solution.

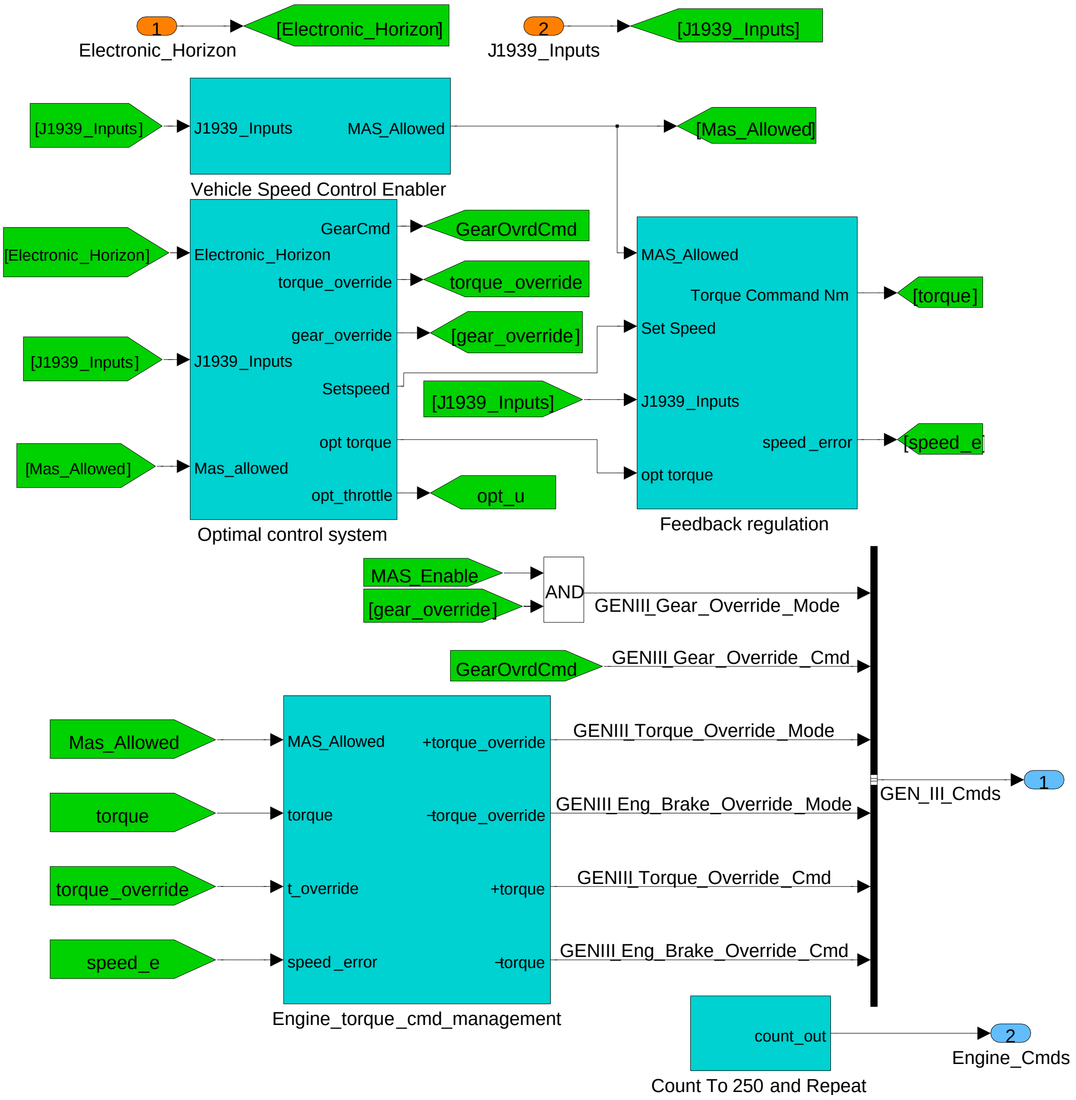


Figure B.3: System scheme of S1

Feedback regulation block is designed to track the optimal speed by applying a torque control mode as a feedback regulation and the optimal torque command as a feedforward servo control. The generated engine torque command is not real time applicable until it has been checked in engine torque command management block for truck operation limitations. The outputs from this block are engine and braking override and torque commands, respectively, which together with the gear override and shifting commands from the OC block are sent as ‘Gen_III_Cmds’ to the CAN transmit block.

B.1.1 Optimal control system scheme

The OC system block is the essential part of the complete system, which includes three main system blocks: 'CAN and EH data processing', 'Supervisor system', and 'Map-matching lookup table', as demonstrated in Figure B.4. The J1939 data applied are acceleration pedal on/off, braking on/off, vehicle speed, selected gear, and current engine speed.

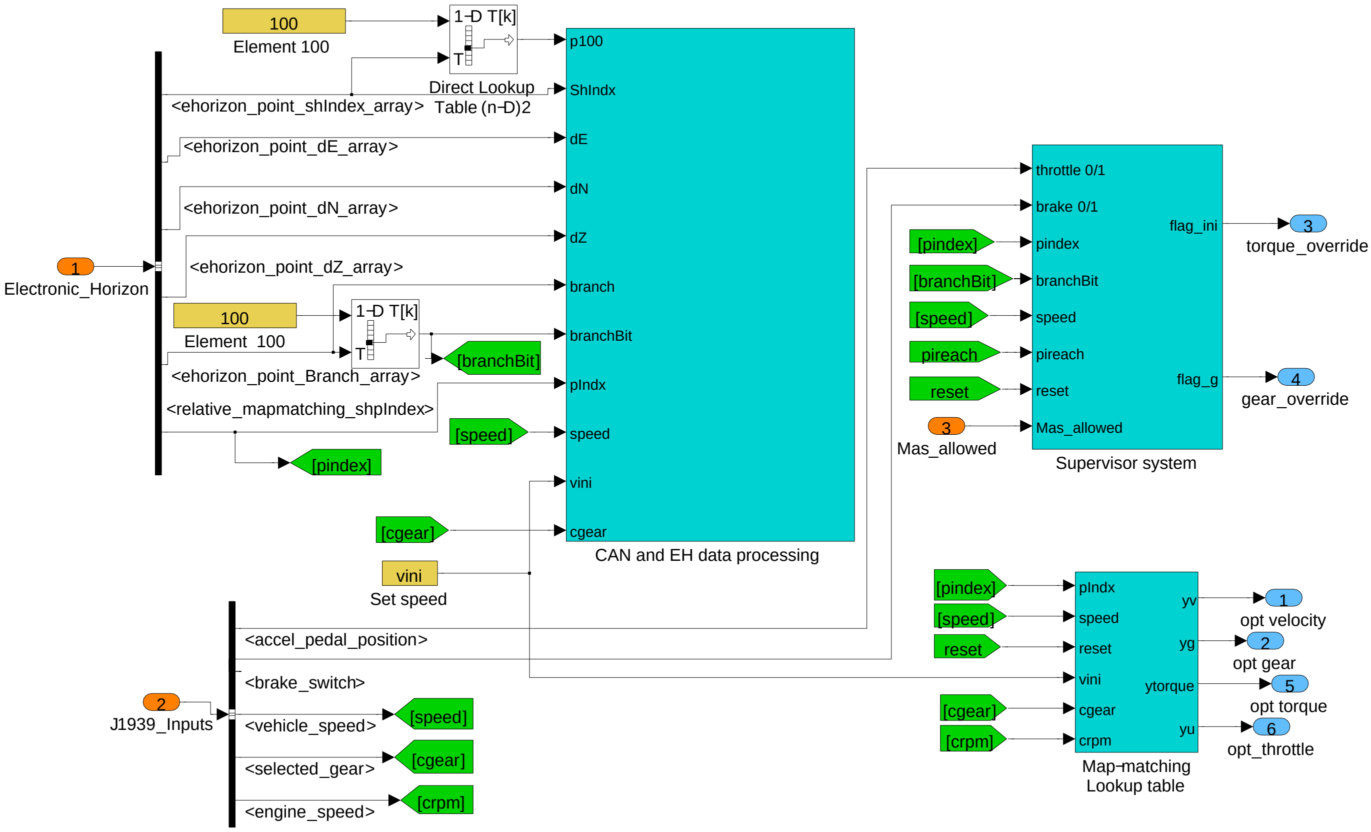


Figure B.4: Optimal control system scheme

From EH, the road geometry ahead of the vehicle's current location - called the 'horizon' - is sent as an array of Shape Point (SP) messages. The key EH data acquired by 'CAN and EH data processing' block are listed in the following:

- shpIndex array: Index array for all shape points in the prediction horizon;

- p100: The 100^{th} point in shpIndenx array. The actual prediction horizon is starting from this point to the end point.

- dE, dN and dZ array: Distance array between every two shape points in east and north direction and elevation difference array between every two shape points;

- relative map-matching shpIndex (pindex): Index for the current vehicle position which can be matching to the shpIndex array;

- branchBit: Flag for the branch road.

Normally, the distance between every two shape points is around 3.5 m, and therefore there are around 900 shape points in a prediction horizon. The EH update is depicted in Figure B.5.

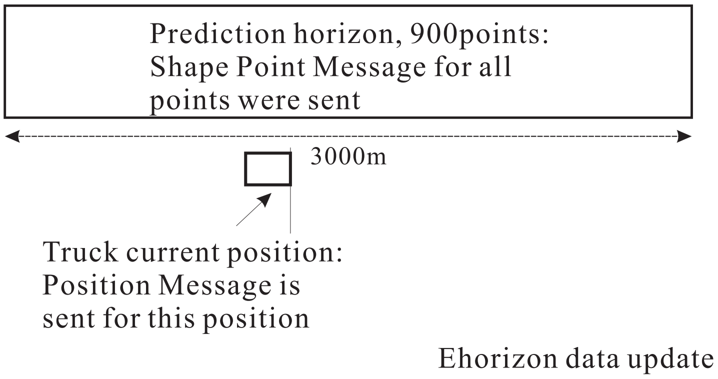


Figure B.5: Electronic Horizon system information update

A simplified OC system flow chart and how S1 and S2 interface to each other in parallel are depicted in Figure B.6. The supervisor system is implemented in Stateflow and consists of two main state variables, 'flag_ini' and 'flag_b' which represent the operation conditions of the OC system and the NLP optimizer, respectively. For example, if 'MAS_allowed' is larger than zero, then the OC is set on and 'CAN and EH processing' block is enabled. This data processing block is written by embedded Matlab files with different M functions. Function 'checkEH.m' checks if there are sufficient nonzero points in the 'EH_shpIndex_array'. If true,

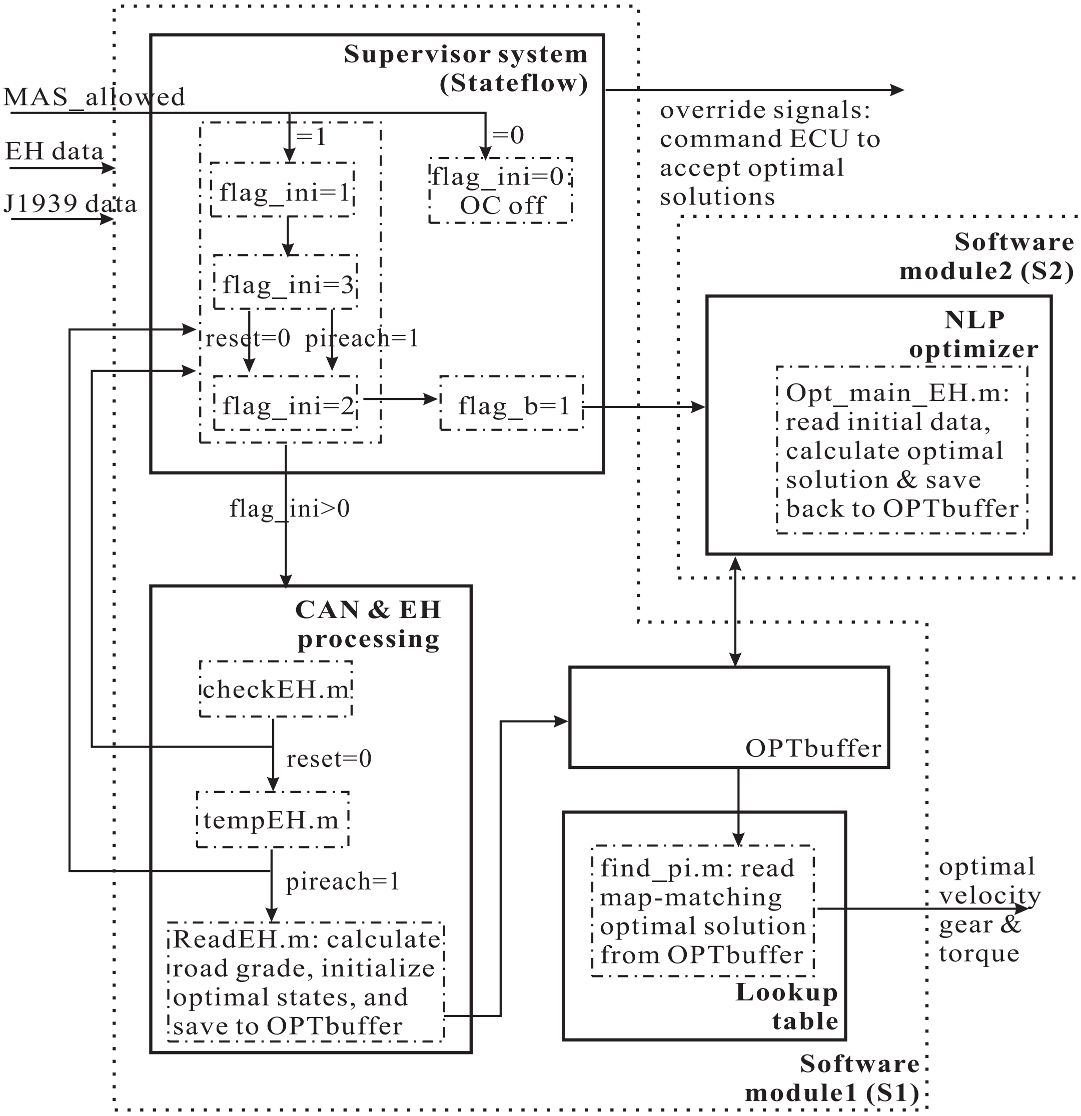

Supervisor system (Stateflow)
MAS_allowed
EH data
J1939 data
=1
=0
flag_ini=1
flag_ini=0; OC off
flag_ini=3
reset=0
pireach=1
flag_ini=2
flag_b=1
override signals: command ECU to accept optimal solutions
Software module2 (S2)
NLP optimizer
Opt_main_EH.m: read initial data, calculate optimal solution & save back to OPTbuffer
flag_ini>0
CAN & EH processing
checkEH.m
reset=0
tempEH.m
pireach=1
ReadEH.m: calculate road grade, initialize optimal states, and save to OPTbuffer
OPTbuffer
find_pi.m: read map-matching optimal solution from OPTbuffer
Lookup table
optimal velocity gear & torque
Software module1 (S1)

Figure B.6: Optimal control system flow chart

it sends a flag, 'reset=0'. If 'reset=0', function 'tempEH.m' is operated to detect if the point p100 has reached a prediction position point. If true, 'pireach=1' is set. Subsequently, in function 'readEH.m', road grade/distance array is calculated based on EH data and optimal solution is initialized for the prediction horizon which together with the 'EH_shpIndex_array' are all put into a buffer, called 'OPTbuffer'.

At the same time, 'reset=0' and 'pireach=1' are sent back to the supervisor system and subsequently it sets 'flag_ini=2' and 'flag_b=1', which trigger the NLP solver in S2 to function in parallel with S1 by: acquire the road grade and initial data array from 'OPTbuffer'; calculate the optimal velocity, torque, and gear shifting; and save the optimal solution array back to the 'OPTbuffer' with the position in accordance with 'EH_shpIndex_array'.

Finally, if 'flag_ini' is larger than zero, a function 'find_pi.m' in 'Lookup table' block is trigged to read the map-matched optimal states and control from 'OPTbuffer' and output to an upper level block. Additionally, the entry 'flag_ini' from the supervisor system is outputted as the engine torque override command. In the following sections, each block in S1 and S2 is given in detail.

B.1.2 Supervisor system

Supervisor system is implemented in Sateflow to handle different the OC operation conditions, e.g., start, stop, traffic in/out, fork road, by sending different 'flags', as shown in Figures B.7 and B.8.

Four states, i.e., OC, NLP, Gear, t_fork, can be identified in Figure B.8, which represent the operation conditions of the OC system, the NLP solver, gear shifting and fork road detection as follows:

- OC state:

    1. OC_off, flag_ini=0: the OC system is set off;
    2. OC_ini, flag_ini=1: the OC system is initialized; While shpIndex array has zero elements in next 900 points, or previous 150 pindex are all equal, the OC functions as a cruise control system, which outputs a constant set speed;
    3. OC_read, flag_ini=2: a predicted shpIndex point is reached, and the NLP solver is triggered/run in S2;
    4. OC_on, flag_ini=3: the OC system is on normal running, and waits for another NLP trigger, flag_ini=2, for the next prediction horizon;

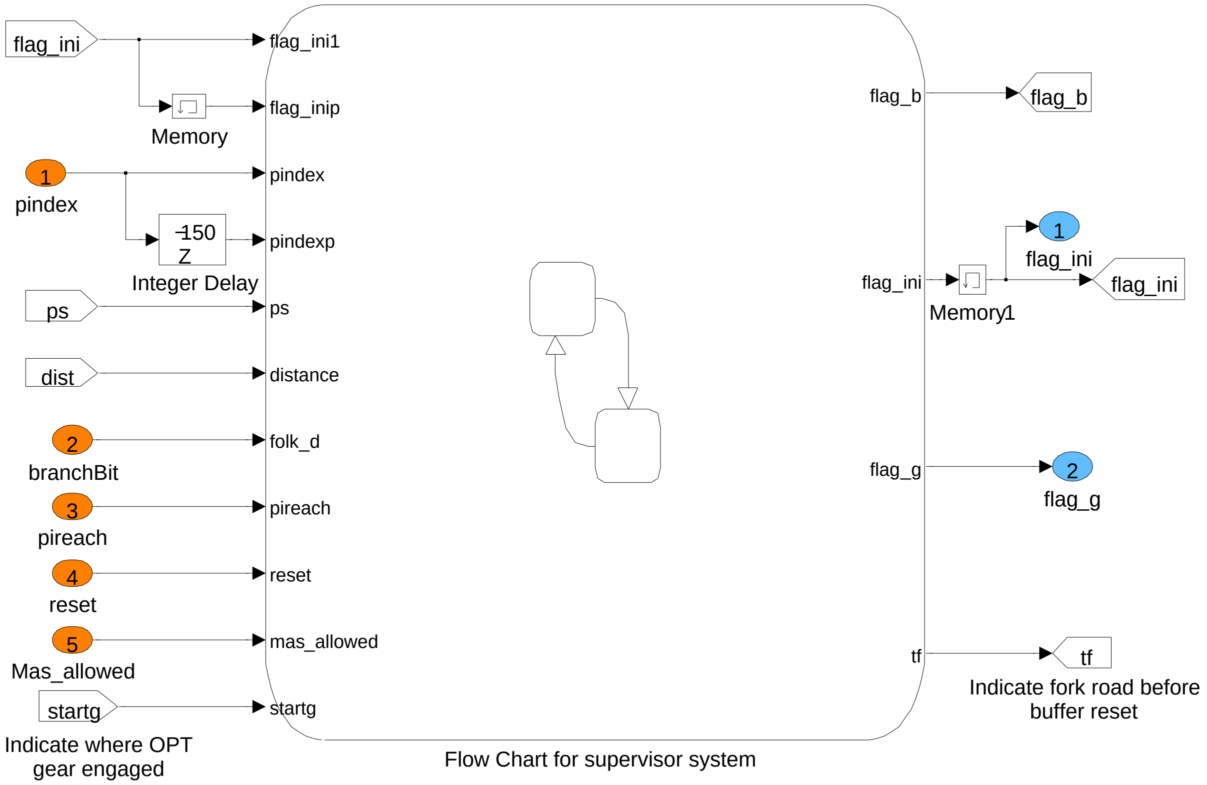


Figure B.7: Supervisor system scheme

5. State flow of OC: OC_off ⟶ (if MAS_allowed==1) ⟶ OC_ini ⟶ OC_read ⟷ OC_on ⟶ (if MAS_allowed==0) ⟶ OPT_off.

- NLP state:

1. NLP_off, flag_b=0: the NLP solver is set off;

2. NLP_on, flag_b=1: the NLP solver is triggered (by flag_ini==2) for a fuel minimization calculation. It is applied at most situations;

3. NLP_on2, flag_b=2: the NLP solver is triggered (by flag_ini==2) for a constant speed tracking calculation. It is applied after the OPT system is first initialized, flag_ini=1 ⟶ flag_ini=2; or the road grade in the prediction horizon is highly nonlinear, e.g., large grade/grade change;

4. State flow of NLP: NLP_off ⟶ NLP_on or NLP_on2 ⟶ NLP_off.

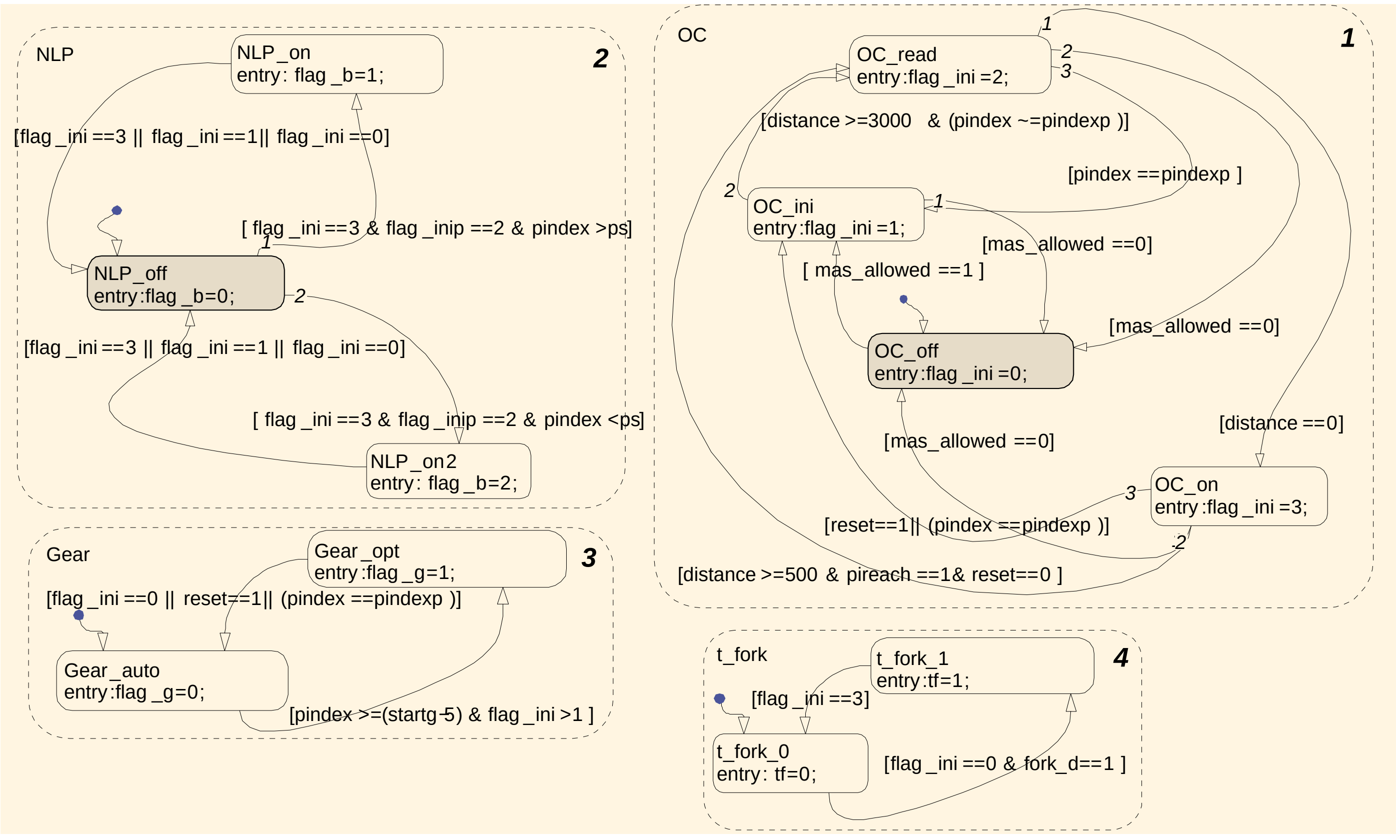

NLP
NLP_on
entry: flag_b=1;
2
[flag_ini ==3 || flag_ini ==1|| flag_ini ==0]
[flag_ini ==3 & flag_inip ==2 & pindex >ps]
1
NLP_off
entry:flag_b=0;
2
[flag_ini ==3 || flag_ini ==1 || flag_ini ==0]
[flag_ini ==3 & flag_inip ==2 & pindex <ps]
NLP_on2
entry: flag_b=2;
Gear
Gear_opt
entry:flag_g=1;
3
[flag_ini ==0 || reset==1|| (pindex ==pindexp)]
Gear_auto
entry:flag_g=0;
[pindex >=(startg-5) & flag_ini >1]
OC
1
OC_read
entry:flag_ini =2;
2
3
1
[distance >=3000 & (pindex ~=pindexp)]
[pindex ==pindexp]
2
OC_ini
entry:flag_ini =1;
1
[mas_allowed ==0]
[mas_allowed ==1]
[mas_allowed ==0]
OC_off
entry:flag_ini =0;
[distance ==0]
[mas_allowed ==0]
3
OC_on
entry:flag_ini =3;
[reset==1|| (pindex ==pindexp)]
2
[distance >=500 & pireach ==1& reset==0]
t_fork
t_fork_1
entry:tf=1;
4
[flag_ini ==3]
t_fork_0
entry: tf=0;
[flag_ini ==0 & fork_d==1]

Figure B.8: Supervisor system Stateflow chart

- Gear state:

  1. Gear_auto, flag_g=0: command the system to run automatic gear shifting. It is applied when the OC system is turned on and truck is passing the first 500 m after the OC is initialized; or when shpIndex array has zero elements in next 900 points, or previous 150 pindex are all equal;

  2. Gear_opt, flag_g=1: command the system to run optimal gear shifting. It is applied when the OC system is on and truck has passed 500 m after the OC is initialized

  3. State flow of gear state: Gear_auto ⟷ Gear_opt.

- t_fork state:

1. t_fork_0, tf=0: command the OC to function as normal since there is no fork road indicated in the prediction horizon;
2. t_fork_1, tf=1: command the OC to restart and re-initialize since there is a fork road in the prediction horizon. For example in Figure B.9, if there is a folk road starting from point F, the NLP optimizer calculates the optimal solution for FG, while truck heads on FH. EH can detect this folk road and send a flag 'fork_d=1', which changes the state to t_fork_1;
3. State flow of t_fork state: t_fork_0 ⟷ t_fork_1.

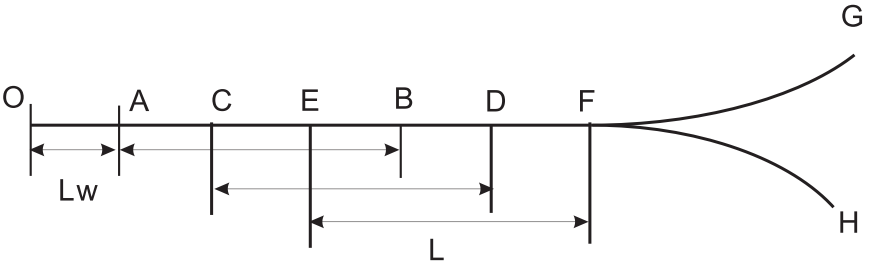


Figure B.9: Fork road condition

B.1.3 CAN and EH data processing

This block is designed to calculate distance and road grade for a prediction horizon, initialize the optimal solution, and save them to 'OPTbuffer'. It is mainly written by an embedded Matlab file block which includes M functions as follows:

- checkEH.m: check if there are 900 nonzero points in the EH_shpindex_array. If true, send a flag, 'reset=0';
- out_flago.m: detect if the previous NLP calculation finishes in a 500 m interval. If true, send a flag, 'flag_o=0', otherwise, 'flag_o>0' ;

- tempEH.m: if ‘reset=0’, detect if the 100th point in index array has reached a prediction point. If true, save current 900 points EH data into a temporary buffer ‘EHbuffer’, and send a flag, ‘pireach=1’;
- processEH.m: if ‘pireach==1’, read EHbuffer, from which select 3 km data, and save to a new buffer, ‘EHbuffer900’; Find indices corresponding to position 0.5 k/1 k/1.5 k/2 km, and save to a vector, ‘startpic’, to trigger next iteration;
- readEH.m: read buffer ‘EHbuffer900’, calculate road grade, and distance and initialize optimal solution for each iteration. Save all data into a new buffer, ‘OPTbuffer’;
- process_flagb.m: detect which kind of optimal calculation is required such as for fuel minimization, or speed tracking.

B.1.4 Map-matching lookup table

This block is designed to read map-matched optimal states and control inputs corresponding to the current shpIndex, ‘pindex’ from ‘OPTbuffer’, and send them to the block in the upper system level. It is mainly written by an embedded Matlab file block which includes M functions as follows:

- find_pi.m: find and output optimal velocity, engine, brake torque, and gear shifting from ‘OPTbuffer’, corresponding to the current shpIndex;
- find_g.m: detect if the commanded gear shifting can run the engine over or under speed. If true, change the commanded gear to the nearest available one.

B.2 Software module 2

At the final step, the system scheme of the NLP solver in S2 is illuminated in Figure B.10, where the M file embedded block, ‘r_flagb’ reads the variables ‘flag_ini’ and ‘flag_b’ saved from the ‘Supervisor system’ block in S1, and sends them to the NLP solver block.

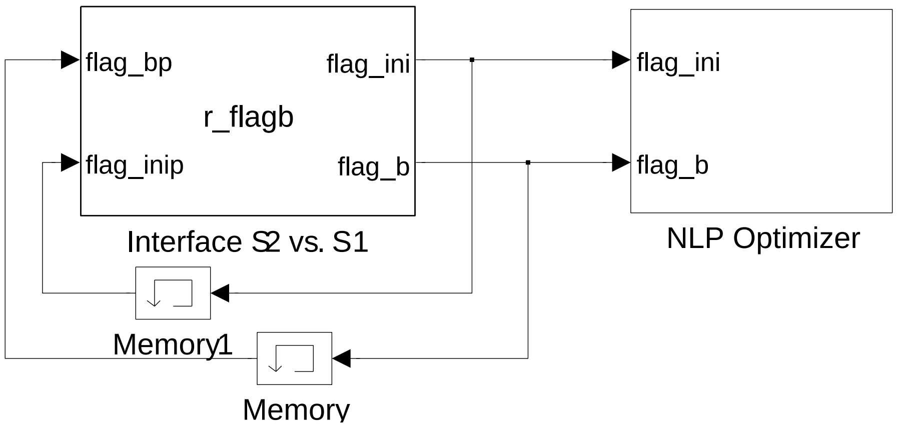


Figure B.10: Nonlinear programming solver system scheme in S2

The NLP solver is generated by a M file s-function ‘opt_main_EHr.m’, which runs whenever ‘flag_b’ is larger than zero, to acquire the road grade and initial data array from ‘OPTbuffer’; calculate the optimal velocity, torque, and gear shifting; and save the map-matched optimal solution array back to ‘OPTbuffer’. While working, it calls M files as follows:

- opt_con_gear_uvgs.m: call optimization function and return the optimal solution;
- fmincon.m: nonlinear constrained optimizer calling from optimization toolbox;
- obj_truck_gear_uvgs.m: the objective function;
- con_truck_gear_ugvs.m: constraint functions;
- hessinterior.m: calculate and provide Hessian to the NLP solver.