The Pennsylvania State University
The Graduate School
College of Engineering

# FORMULATION AND COMPARISON OF TWO REAL-TIME PREDICTIVE GEAR SHIFT ALGORITHMS FOR CONNECTED/AUTOMATED HEAVY-DUTY VEHICLES

A Thesis in
Mechanical Engineering
by
Chu Xu



Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2019

The thesis of Chu Xu was reviewed and approved* by the following:

Hosam K. Fathy
Professor of Mechanical Engineering
Thesis Advisor

Sean N Brennan
Professor of Mechanical Engineering

Daniel Haworth
Department Head of Mechanical Engineering

*Signatures are on file in the Graduate School.

# Abstract

This thesis examines the problem of predictive gear scheduling for fuel consumption minimization in connected/automated heavy trucks. The literature highlights the fuel economy benefits of such predictive scheduling, but there is a need to optimize such scheduling online, in real time. To address this need, we begin by using dynamic programming (DP) to schedule gear shifting offline, in a manner that achieves a globally-optimal Pareto tradeoff between the conflicting objectives of minimizing (i) fuel consumption and (ii) shift frequency. The computational cost of DP is unfavourable for online implementation, but we present two algorithms addressing this challenge. Both algorithms rely on the fact that in the *Pareto limit* where fuel consumption minimization is the sole objective, DP furnishes a simple *static shift map*. Our first algorithm trains a recurrent neural network to prune the shift schedule generated by this map. The second algorithm performs this pruning in a direct manner tailored to reduce the schedule's rain flow count. We simulate these algorithms for different drive cycles. Both algorithms achieve a reasonable tradeoff between fuel consumption and gear shift frequency. However, the rain flow count algorithm is both more effective in approaching the DP-based Pareto front and more computationally efficient.

# Table of Contents

# List of Figures

# List of Tables

# Acknowledgments

This work is funded by the ARPA-E NEXTCAR program. The author gratefully acknowledges this support. The author thanks Evan Pelletier and Mohamed Wahba from Intelligent Vehicles and Systems Group (IVSG) at Penn State University for assistance with the Simulink Real-time validation. Furthermore, assistance, support, and extensive collaboration from Volvo Group North America in the simulation-based analyses included in this thesis is gratefully acknowledged by the author.

# Chapter 1 Introduction

## 1.1 Literature Review

This thesis examines the problem of optimizing the upcoming sequence of gear shifts in a heavy-duty truck for both fuel economy and operator comfort. The thesis is motivated by the rapid growth in connected and automated vehicle (CAV) technology [1]. Connected and automated vehicles have the ability to predict and perhaps optimize their upcoming speeds (and therefore road loads), at least to some degree. This creates an exciting opportunity for optimizing these vehicles' powertrain and chassis control for objectives such as fuel consumption minimization. This is particularly valuable for heavy-duty trucks. Existing work by Hellstr*ö*m *et al.*, for instance, demonstrates fuel consumption reductions of 2.5% to 4.77% through a combination of optimal predictive gear scheduling and cruise control [2].

The literature already examines the problem of optimizing vehicle speed and gear shift trajectories for objectives including fuel economy, emissions, and drivability. Some of the algorithms used for this optimization include deterministic dynamic programming (DDP) [2–8], stochastic dynamic programming (SDP) [9–11], Pontryagin methods [12], economic model-predictive control (MPC) [13–15], fuzzy logic [16–19], neural networks [20, 21], machine learning [22], and genetic optimization [23]. Throughout this literature, the potential benefits of predictive optimization are clear, but one particular question lingers, namely: *can optimal (or sub-optimal) trajectories be found online, in real time, in a computationally efficient manner?*

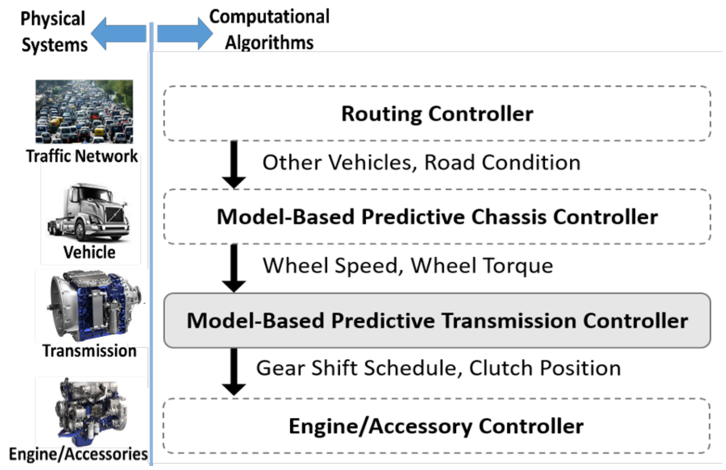


**Figure 1.1.** Conceptual Framework for Hierarchical Connected/Automated Vehicle Control.

## 1.2 Thesis Contributions

This thesis examines the challenge of optimizing the sequence of gear shifts in a heavy-duty truck for the conflicting objectives of: (i) minimizing fuel consumption; and (ii) improving drivability by minimizing the frequency of gear shifting. The thesis's goal is to furnish and evaluate algorithms for solving this multi-objective optimization problem online, in real time. In doing so, we make the key assumption that the truck is capable of predicting its upcoming duty cycle (i.e., its wheel force and longitudinal velocity trajectories) forward in time. This assumption makes predictive optimal gear scheduling possible, and makes the thesis's algorithms suitable for implementation in vehicles equipped with predictive speed trajectory optimization. Of particular relevance, in this context, is the advent of connected and automated vehicles (CAVs) capable of coordinating their speed trajectories in a predictive manner, through hierarchical control schemes such as the one sketched in Figure 1.1. Such vehicles have the potential to optimize their choices of routes, speed trajectories, transmission control, and engine/accessory control in a manner that accounts for surrounding traffic and upcoming road profiles [24]. The intent of this work is to furnish the optimal predictive gear shift scheduling algorithm required for such vehicle's overall hierarchical control schemes. This algorithm is presented as the third layer (from the top) in the hierarchical control scheme shown in Figure 1.1.

Figure 1.2 summarizes both the contributions of this thesis and the connections between them. We begin by using dynamic programming (DP) to obtain globally-optimal gear shift trajectories for different vehicle duty cycles, where a "duty cycle" consists of a known upcoming vehicle speed trajectory and a known upcoming terrain. Optimization is performed for a Pareto combination of two objectives, namely, fuel economy and gear shift frequency. In the Pareto limit where fuel economy is the sole optimization objective, the DP solution collapses to a simple static shift map. This makes it possible for us to construct two *online* gear shift trajectory optimization algorithms, both of them exploiting the above static shift map.

The first proposed online strategy appears in a prior publication by the authors [25], and serves as an important benchmark for the new work presented here. This online strategy combines ideas from dynamic programming and machine learning. Specifically, suppose that the above static shift map is used for obtaining a tentative gear schedule for some upcoming time horizon. Intuitively, one can argue that some of the shifts in this tentative schedule are essential for fuel consumption minimization, while others represent unnecessary gear hunting. This raises an interesting possibility: perhaps a neural network can be trained to predict, at some time instant $k$, the truly Pareto-optimal gear shift decisions at the following time steps, given the tentative shift schedule computed using the static shift map. In other words, perhaps a neural network can be trained to "filter" the tentative gear shift schedule generated by the static shift map, bringing it closer to the Pareto-optimal shift schedule one would obtain through offline DP. We construct such a neural network and train it using the gear shift schedules generated through offline DP optimizaiton for a mix of urban and suburban drive cycles. The end product is an online model-based predictive gear shift optimization algorithm that combines the static shift map with the neural network. Our previous research explored the possibility of creating a single version of this algorithm for online gear shift schedule optimization in both urban and highway settings. In contrast, this thesis explores the possibility of training different versions of this algorithm for different driving scenarios.

The primary contribution of this thesis is to develop a second online gear shift scheduling algorithm inspired by the concept of "rain flow counting". Specifically, the algorithm enumerates the number of times that a given gear shift schedule

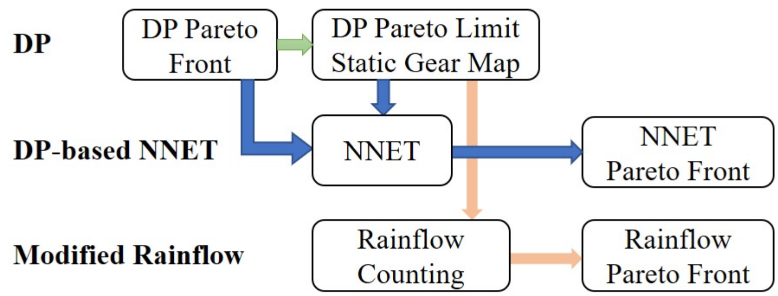


**Figure 1.2.** Conceptual layout of the structure of this thesis.

switches from up-shifting to down-shifting, and vice versa. This furnishes a gear shift "cycle count" over a predictive gear shift scheduling horizon. It is possible to reduce this cycle count by "flattening" one or more gear shift cycles. Consider, for instance, a gear shift schedule such that, over the upcoming 5 seconds, the vehicle is in the following gears: 2,2,4,3,3. This schedule has a cycle count of 1, in the sense that there is one sequence of up-shifting followed by down-shifting. This cycle count can be reduced by adopting the following shift schedule instead: 2,2,3,3,3. In performing this flattening, we reduce the unnecessarily large magnitude of the up-shift while eliminating the down-shift altogether, thereby reducing the cycle count. In a shift schedule with many gear shifts, there are potentially many different cycles and therefore many different potential pathways for flattening the schedule. Our algorithm relies on the above static shift map to generate an initial tentative gear shift schedule, then recursively generates a full "family tree" of possible flattened versions of this schedule. We choose the flattened schedule that minimizes fuel consumption, subject to engine torque feasibility constraints and a predefined cap on the number of cycles over an optimization horizon.

Both of the above proposed algorithms are computationally tractable and feasible for online implementation, at least for reasonable prediction horizons (e.g., 20 seconds or less). Moreover, both of them attempt to replicate the Pareto-optimal results of offline DP online, in real time, starting with gear schedules generated by the static shift map. This raises two questions. First, how do these two algorithms compare to each other in terms of their ability to replicate the Pareto front generated by offline dynamic programming? Second, how do these algorithms compare in terms of computational tractability? The remainder of this thesis addresses these questions, and is organized as follows. Chapter 2 provides important background details regarding the underlying optimization challenge. This includes parameters of the target heavy-duty vehicle, the vehicle's model, and information about the drive cycles used for comparing the algorithms. Chapter 3 presents the DP optimization problem formulation and simulation results. Moreover, it analyzes the Pareto tradeoff between fuel consumption and gear shift frequency, and generates the static shift map as a realization of the DP Pareto limit. Chapter 4 introduces the neural network-based online policy, and examines the Pareto front generated by this policy. Chapter 5 describes the proposed rain flow count algorithm, and compares the two proposed online algorithms in terms of Pareto optimality and computational efficiency. Chapter 6 finally summarizes this thesis's conclusions.

# Chapter 2 | Problem Setup

To set up an optimal gear shift scheduling problem, one needs a detailed description of the underlying vehicle's chassis/powertrain, as well as a description of the drive cycles to be used for optimization and assessment. Both sets of information are summarized below.

## 2.1 Target Vehicle Description

The focus of this thesis is on gear shift schedule optimization for heavy-duty trucks. This is an interesting problem for at least three reasons. First, heavy-duty truck transmissions typically offer a much larger selection of gears compared to other vehicles, e.g., passenger sedans. This increases the computational complexity of the problem. Second, market pressure makes it critical for truck gear scheduling to eke out the best fuel efficiency levels possible, subject to constraints on operator comfort. This highlights the importance of gear shift schedule optimization. Third, heavy-duty trucks are typically quite power-limited, with very small ratios of propulsion power capacity to gross vehicle weight compared to other road vehicles. This highlights the practical constraints on gear shift schedule optimization.

The work presented in this thesis is conceptually applicable to a broad range of heavy-duty vehicles. However, we apply this work specifically to the Volvo VNL300 with a D13 500HP diesel engine as a target vehicle. Table 2.1 gives the key parameters of this target vehicle. This vehicle has a conventional powertrain, where power flows from the engine to the wheels through a clutch, a 12-ratio gear box, plus a final drive. For simplicity, we assume a constant accessory torque and neglect both the reflected driveline inertia and energy losses in the vehicle's

Table 2.1. Target Vehicle Parameters

| | |
|---|---|
| **Vehicle Volvo** | VNL300 |
| **Diesel Engine** | D13 500HP |
| **Automated Manual Transmission** | 12-speed GR:11.73/9.21/7.09/5.57/4.34/3.41/2.69/ 2.12/1.63/ 1.28/1.00/ 0.785 |
| **Final Drive** | 2.47 |
| **Vehicle mass** | 35000 kg (full load) |

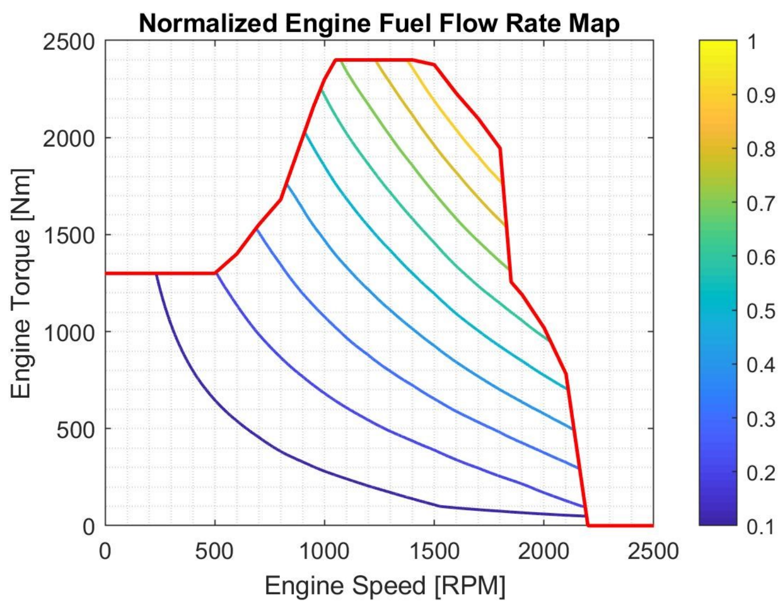


**Figure 2.1.** Normalized engine fuel flow rate with respect to engine speed and engine torque.

transmission. These assumptions place some limitations on the fidelity of this thesis's optimization studies, but the thesis's approach can be easily extended to incorporate the above neglected effects.

We model the diesel engine using a fuel flow rate map provided by Volvo, as in Figure 2.1. The thesis constrains the engine torque within the redline region in order to get better engine performance; however, the feasible operating points extend slightly beyond this region. The fuel consumption map is normalized, but shows the trend of the fuel flow rate with respect to engine speed and engine torque.

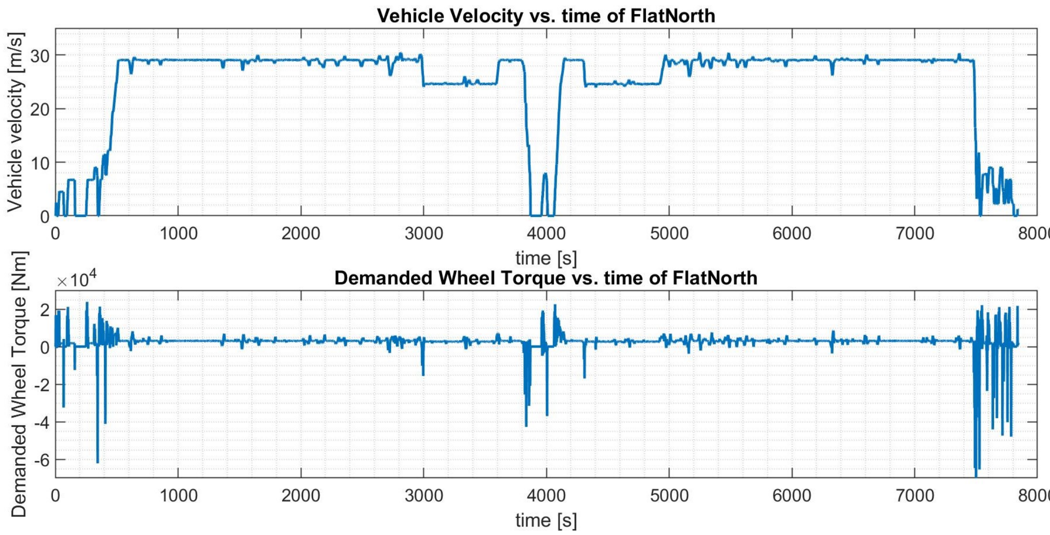


**Figure 2.2.** Vehicle velocity and demanded wheel torque of the FlatNorth cycle.

## 2.2 Drive Cycle Information

To simulate the proposed gear shift algorithms, we assume that an upper-level chassis controller provides our proposed algorithms with reference driveline output speed and torque trajectories to track, at least over a reasonable upcoming prediction horizon. This assumption is consistent with the paper's focus on predictive gear shift schedule optimization for a connected and automated vehicle. Our work in this paper focuses on 6 test drive cycles including a mix of urban, suburban, and highway driving. These drive cycles are representative driving use cases experienced by Volvo truck operators. The total lengths, total durations, and ranges of road grades for these cycles are listed in Table 2.2. This paper examines and compares simulation results for all of these drive cycles. Moreover, the paper presents in-depth results and comparisons for two representative drive cycles, namely, FlatNorth cycle and MixedDrvC cycle, corresponding to highway and city scenarios. The wheel speed and demanded wheel torque trajectories for these two drive cycles are shown in Figure 2.2 and Figure 2.3, respectively.

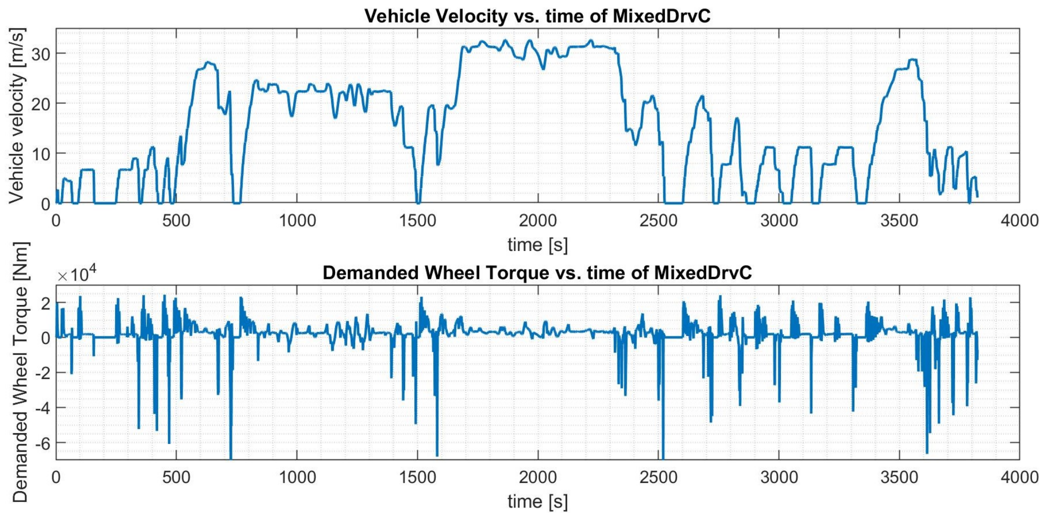


**Figure 2.3.** Vehicle velocity and demanded wheel torque of the MixedDrvC cycle.

**Table 2.2.** Drive Cycle Information

| **Name** | **Distance (km)** | **Duration (min)** | **Road Slope (percentage)** |
|---|---|---|---|
| 81Marion | 41.85 | 35.8 | -4.50∼5.20 |
| FlatNorth | 194.98 | 130.1 | -3.70∼3.80 |
| ILwest | 371.33 | 227.7 | -3.80∼5.81 |
| INwest | 258.86 | 163.7 | -3.50 ∼4.10 |
| MixedDrvC | 60.25 | 63.7 | -5.30∼6.71 |
| OHwest | 259.72 | 170.8 | -5.00∼4.81 |

# Chapter 3
# DP Optimization and Pareto front

## 3.1 DP Problem Formulation

This paper uses DP to search for the globally optimal solution because of the non-convex and discrete nature of the gear shift scheduling problem [26]. We formulate this DP problem as follows. The state variable is the current gear position, $x(k)$, which can take any integer value from 1 to 12. The input variable is the gear shift change command, $u(k)$, and the gear shift dynamics are governed by Equation 3.1. We allow the gear shift command $u(k)$ to take any integer value in the interval [-4, 4], provided the resulting gear is feasible (e.g., capable of satisfying wheel torque requests, etc.). This constraint on the input range is intended to reflect: (i) the physical limitations of the underlying drivetrain; (ii) the low likelihood of very large gear shifts in practice; and (iii) the diminishing benefits (in terms of both drivability and fuel economy) associated with larger input ranges.

$$x(k+1) = \begin{cases} 12, & \text{if } x(k)+u(k) > 12. \\ 1, & \text{if } x(k)+u(k) < 1. \\ x(k)+u(k), & \text{otherwise.} \end{cases} \tag{3.1}$$

The optimization objective is to minimize the cost function $J$, which is a linear combination of two objectives, namely, fuel consumption and gear shift frequency. The intent of the latter objective is to reduce operator's discomfort. The objective of the problem is shown as below:

$$\min J = \sum_{k=1}^{N} [\dot{m}_f(k) + \lambda N_{sc}(k)] \Delta T \tag{3.2}$$

where $N$ is the number of time steps in the vehicle duty cycle, and $\Delta T$ is the discrete time step used for dynamic programming. We set the controller time step to $\Delta T = 1s$ throughout the remainder of this thesis. The quantity $\dot{m}_f$ is the mass flow rate of fuel input to the engine, which is a known function of the engine angular velocity $\omega_{eng}$ and the engine's output torque $\tau_{eng}$, shown in Figure 2.1. The weight $\lambda$ is a penalty coefficient on gear shift frequency. The counter $N_{sc}$ indicates the number of gear shifts over the vehicle duty cycle. The indicator $N_{sc}(k)$ equals 1 if the transmission gear ratio changes at time step $k$, and equals 0 otherwise. Furthermore, we apply the following constraints to our optimization problem:

$$\omega_{e-min} \leq \omega_{eng}(k) \leq \omega_{e-max} \tag{3.3}$$

$$\tau_{e-min} \leq \tau_{eng}(k) \leq \tau_{e-max}(\omega_{eng}) \tag{3.4}$$

$$\omega_{eng}(k) = \omega_{wheel}(k) \times R_{fd} \times R_{trans}(x(k)) \tag{3.5}$$

$$\tau_{predl}(k) = \frac{\tau_{wheel}(k)}{R_{fd} \times R_{trans}(x(k))} \tag{3.6}$$

$$m\dot{v}(k) = F_{prop}(k) - \frac{1}{2}\rho C_d A_f v(k)^2 - \mu m g \cos\theta(k) - m g \sin\theta(k) \tag{3.7}$$

where $\omega_{eng}$ is the engine's angular velocity within a lower bound $\omega_{e-min}$ and a higher bound $\omega_{e-max}$, $\tau_{eng}$ is constrained within the engine redline, the maximum engine torque $\tau_{e-max}$ is a function of the engine speed and the minimum engine torque $\tau_{e-min}$ is a scalar. $\omega_{wheel}$ is the (known) wheel angular velocity of the vehicle as a function of time, $\tau_{wheel}$ is the (known) torque demanded at the wheel as a function of time, $\tau_{predl}$ is the pre-driveline torque demand (which equals the engine torque demand minus the accessory torque demand), $R_{fd}$ is the final drive gear ratio, and $R_{trans}$ is the transmission gear ratio. The longitudinal vehicle dynamics are considered as redundant constraints. However, we list it here to keep the problem description complete. The vehicle dynamics are presented using standard

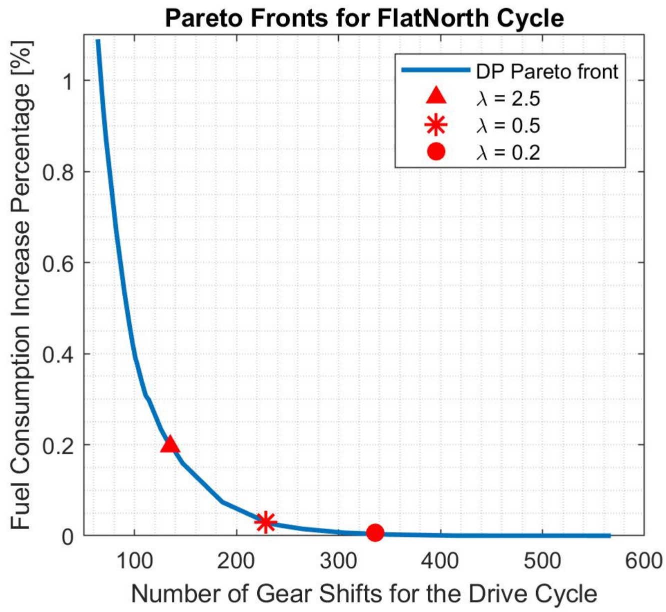


**Figure 3.1.** DP Pareto front of the FlatNorth drive cycle.

formula in Equation 3.7, where $F_{prop}$ is the proportion force, $\rho$ is the air density, $C_d$ is the drag coefficient, $A_f$ is the frontal area of the truck, $v$ is the vehicle speed, $\mu$ is the road resistance coefficient, $\theta$ is the road grade. The authors use SI units for all the variables in the above Equations.

## 3.2 DP Simulation Results

Solving the above DP problem for a given vehicle duty cycle furnishes a Pareto-optimal gear shift schedule, i.e., a schedule that achieves an optimal tradeoff between fuel consumption and gear shift frequency. When the scalarization weight $\lambda$ varies in the above DP problem, a conflict emerges between the fuel consumption minimization and shift frequency minimization objectives. One can navigate this trade-off by solving the DP problem for different values of $\lambda$. One interesting fact is that the Pareto front of optimal fuel consumption levels versus shift frequency is cycle-dependent. Figure 3.1 and Figure 3.2 plot this Pareto fronts for two representative drive cycles. The fuel consumption levels in the Pareto fronts are normalized with respect to the corresponding Pareto-limit fuel consumption levels (where $\lambda = 0$).

In the Pareto limit where the gear shift penalty weight $\lambda$ reaches zero, the DP

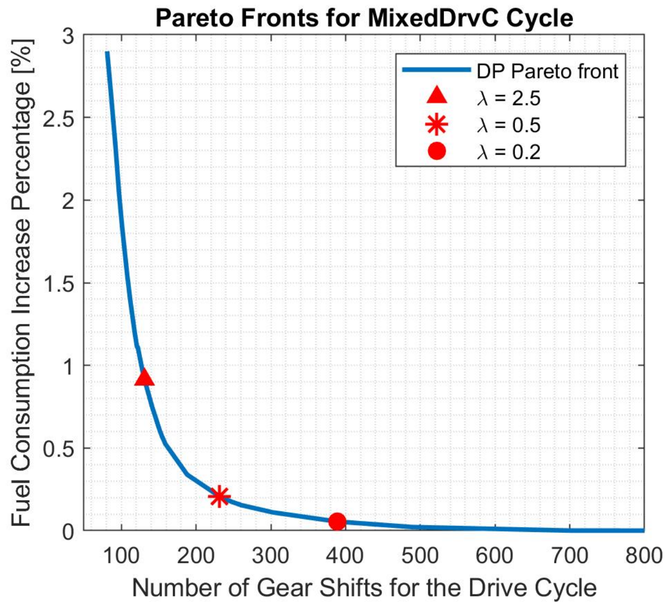


**Figure 3.2.** DP Pareto front of the MixedDrvC drive cycle.

algorithm approaches a solution that minimizes fuel consumption independently at every instant in time. Gear shift frequency is inconsequential in that scenario, and as a result the DP optimizer shifts as often as it needs to in order to minimize fuel consumption independently at each instant in time. Thus, compared to the dynamic, cycle-dependent optimal gear shift schedule from the DP Pareto front, the optimal gear choice becomes a static decision made solely based on the predicted/demanded wheel torque and speed at every moment in time, as in Figure 3.3. For every combination of wheel speed and torque, this static shift map shows the gear ratio that minimizes instantaneous fuel consumption. Given the drive cycle information, one can use this shift map to obtain gear shift schedules with globally-minimal fuel consumption, in simulation, because they represent the Pareto limit where instantaneous fuel consumption is the only objective. Penalties on frequent shifting, regardless of whether they represent drivability or the fuel losses associated with shift transients, are not accounted for in this Pareto limit. Thereby, the Pareto-limit control policies can serve as optimistic benchmarks for the fuel savings one may be able to achieve through predictive gear schedule optimization. We use these benchmarks when analyzing the fuel savings potential of Pareto-optimal shift schedules corresponding to nonzero values of $\lambda$.

Every point on a Pareto front is, by definition, a "Pareto-optimal" point, in

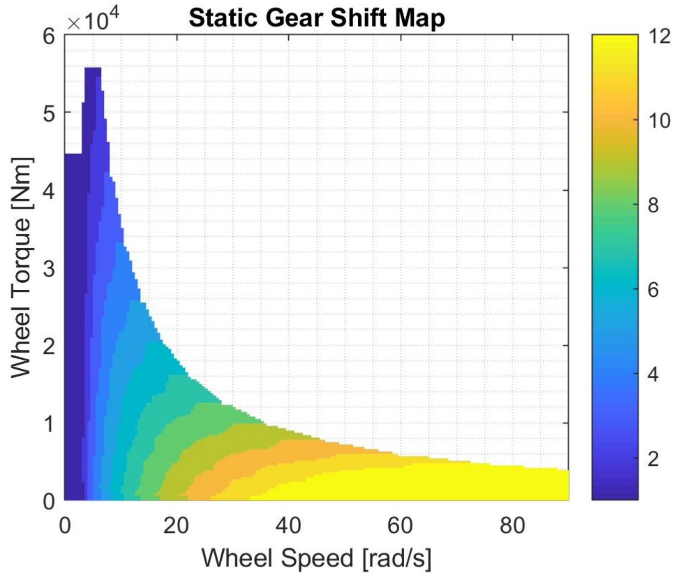


**Figure 3.3.** Static gear shift map with minimal fuel consumption with respect to wheel speed torque.

the sense that it is not possible to reduce fuel consumption beyond that point without incurring the penalty of more frequent gear shifting. Different Pareto-optimal solutions correspond to different levels of preference for fuel consumption minimization versus drivability. Figure 3.1 and Figure 3.2 show the Pareto fronts for the FlatNorth and MixedDrvC cycles, respectively. Three different Pareto-optimal points are highlighted on these Pareto fronts, corresponding to the scalarization weights $\lambda = 2.5$, 0.5 and 0.2. Increasing this scalarization weight leads to a reduction in the total number of gear shifts for a given cycle, at the expense of an increase in fuel consumption. We select the penalty weight $\lambda = 0.5$ for the analyses in the remainder of this paper because it provides a reasonable tradeoff between these two objectives, but the analyses can be repeated for other values of $\lambda$. Compared to the Pareto-limit gear shift strategy, this weight decreases gear shift frequency substantially to one shift per 35 seconds on the highway and 16 seconds in the city. The corresponding increases in fuel consumption are quite modest, namely, 0.030% on the highway and 0.179% in the city. Figure 3.4 and Figure 3.5 show the gear shift schedules from the Pareto limit and Pareto optimum ($\lambda = 0.5$) for the FlatNorth and MixedDrvC cycles. The zoom-in plots demonstrate significant decreases in gear shift frequency.

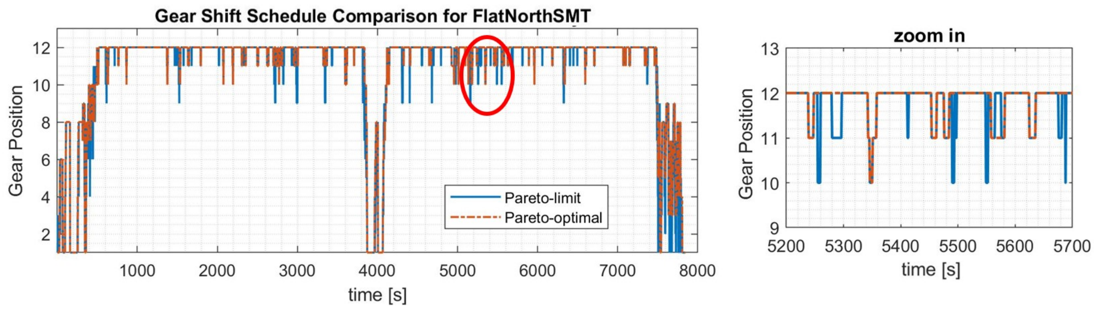


**Figure 3.4.** Gear shift schedule comparison of the FlatNorth cycle.

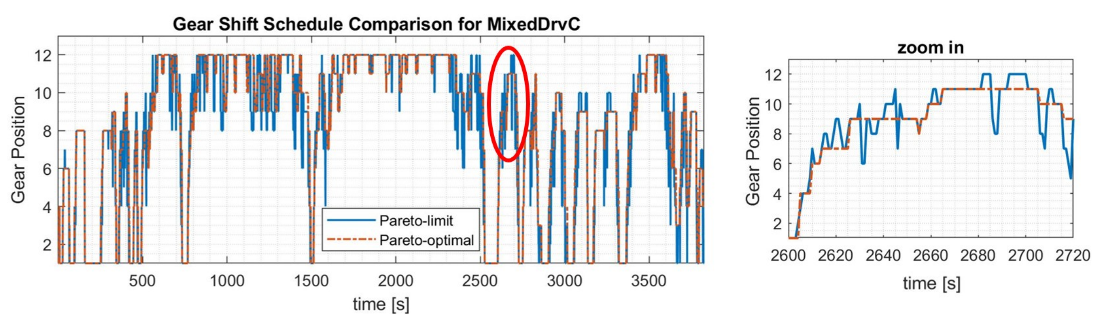


**Figure 3.5.** Gear shift schedule comparison of the MixedDrvC cycle.

# Chapter 4
# DP-based Neural Network

This chapter answers the following question: is it possible to train a neural network to schedule gear shifts online in a truck, in a computationally tractable manner, such that this schedule achieves fuel savings comparable to the offline solution of the corresponding DP problem? On the one hand, we seek the attractive computational tractability and learning capabilities traditionally associated with neural networks. On the other hand, we seek to remain close to the globally Pareto-optimal fuel consumption savings achievable through DP.

Our approach to this problem is twofold. First, we use the offline DP optimization results, for a mix of urban and highway duty cycles, to train a neural network-based online shift scheduler. Second, we train the neural network to predict the Pareto-optimal gear shift decisions for the following 5 time steps assuming that the neural network has access to a longer future history of Pareto-limit shift decisions. In other words, we train the neural network to predict the optimal dynamic gear shifting decisions, 5 seconds into the future, given a sequence of future gear shifts obtained using the static shift map. Intuitively, we are training the neural network to "prune" the gear shift sequence obtained using the static shift map by keeping the gear shifts critical for fuel consumption minimization and rejecting unnecessary gear hunting. From a more fundamental perspective, the use of the static shift map to pre-process the inputs to the neural network reduces the number of inputs from a set of wheel torque/speed predictions to a set of Pareto-limit gear shift predictions. This leads to a neural network that is smaller, and therefore potentially easier to train.

## 4.1 Neural Network Framework

The neural network's structure is shown in Figure 4.1. The recurrent nature of the neural network is appealing because it provides the neural network with its own internal dynamics, in the form of an internal "memory" of past gear shift events. This can potentially aid in preventing gear hunting. The neural network uses a sigmoid transfer function for the hidden layers and a pure linear transfer function for the output layer.

The input of the neural network is a sequence of gear shifts including the current transmission gear and the Pareto-limit gears selected by the static shift map over a future time horizon, $N_{in}$. The output of the neural network is a sequence or predicted Pareto-optimal gear choices (i.e., from time step $k+1$ to $k+N_{out}$), for a chosen scalarization weight $\lambda$, where $N_{out}$ is the output horizon. We construct a neural network with 2 hidden layers in this thesis, but this can be easily adjusted. The goal is to essentially solve a classification problem of 12 classes, namely, the 12 gear positions. The output is a $N_{out}$-element vector with each element in range [0,1]. We equally divide the range [0,1] in 12 segments, and the location of the output elements shows the gear position.

In this study, we set $N_{in}$ to be 7, meaning that the neural network's inputs include a 7-second sequence of gear selections generated by the static shift map. Moreover, we set $N_{out}$ to be 5, meaning that the neural network's output is a 5-second sequence of gear selections. Together, these choices ensure that the neural network attempts to optimize the gear shift schedule over a portion of full prediction horizon, consistent with the model predictive control literature. The number of neurons in each hidden layer has a significant influence on the performance and complexity of the neural network, as shown in Figure 4.2. The computational cost of training the neural network increases considerably with the size of its hidden layers. Moreover, neural network sizes substantially larger than 10 neurons per hidden layers do not produce substantial improvements in mean square validation error. Given this tradeoff, we construct the network with 10 neurons per hidden layer in order to balance the computational cost of fitting the network versus the accuracy with which it learns.

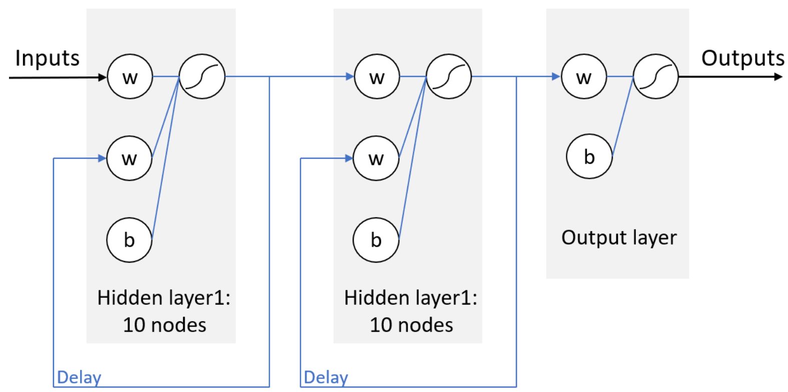


**Figure 4.1.** Neural network framework.

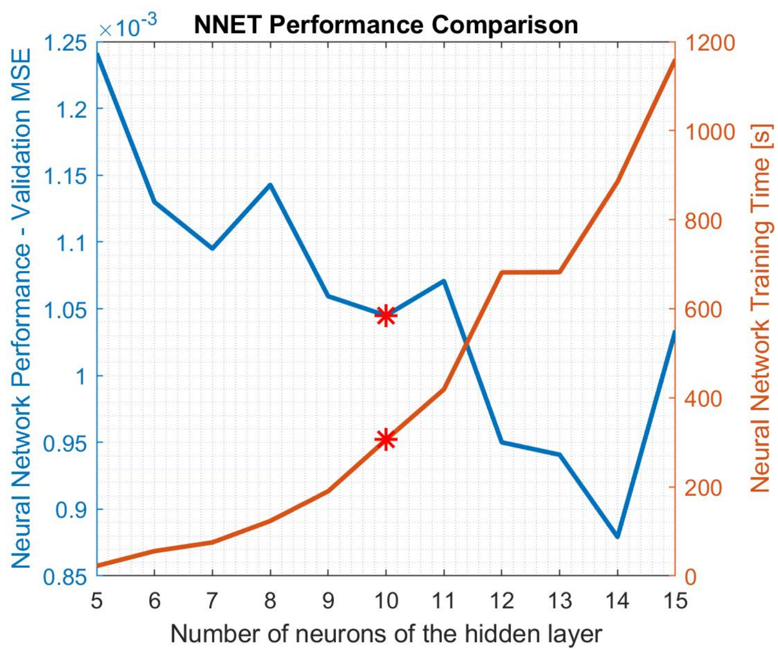


**Figure 4.2.** NNET performance comparison with respect to the number of neurons of the hidden layer.

## 4.2 Pareto Fronts and Simulation Results

We train the neural network offline using a mix of DP-optimized gear shift schedules for a mix of known urban and highway drive cycles, including the cycles listed in Table 2.2. Drive cycle used but not listed in Table 2.2 include EPA drive cycles such as HWFET and US06, modified to comply with the wheel torque constraints associated with heavy-duty truck propulsion. The intent of adding more drive cycles is to enrich the training data set. The chosen scalarization weight $\lambda$ is essential to the performance of the neural network because of the fact that different sets of training data lead to distinct neural networks. If we march through the DP Pareto fronts with different $\lambda$ values, we are able to get the data sets to train several neural networks. We choose the DP solutions with 26 different penalty weights (from 0.005 to 21) to train the neural networks.

The above training process furnishes different neural networks (i.e., different values of the neural network weights) for different values of $\lambda$. One can assess these neural networks in simulation for different drive cycles, in terms of this thesis's two key performance metrics (namely, fuel consumption and gear shift frequency). This assessment shows that the neural networks corresponding to different values of the weight $\lambda$ do not necessarily constitute a Pareto front, even when the DP data used for training them is Pareto-optimal. To remedy this, we select a subset of these neural network that constitutes a Pareto front, in the sense that every member of this subset is non-dominated. We then compare the resulting neural network-based Pareto fronts to the DP-based "true" Pareto fronts for different representative drive cycles, as shown in Figures 4.3 and 4.4. This comparison shows that there is a loss of Pareto optimality involved in using the proposed neural network-based shift strategy as opposed to dynamic programming. However, this loss tends to diminish in the limit as fuel consumption minimization becomes the sole optimization objective. The red stars and triangles in these figures both correspond to the weight $\lambda = 0.5$. In particular, the red star represents the DP-optimal shift schedule generated for that weight, and the red triangle represents the neural network trained using that specific shift schedule. Interestingly, the neural network implicitly learns to compromise gear shift frequency somewhat in order to maintain a comparable fuel consumption level. In other words, for a given value of $\lambda$, the neural network matches the DP solution much better in terms of fuel consumption minimization versus gear shift frequency.

Figures 4.5 and 4.6 compare the DP and neural network results in more depth by showing the gear shift schedules they generate for a highway cycle ("FlatNorth") and an urban/suburban cycle ("MixedDrvC"). We can see that the neural network is able to predict the correct DP-optimal gear position for a large fraction of the time. The neural network tends to shift gears slightly more frequently than the Pareto-optimal policy for a comparable level of fuel consumption. One can see

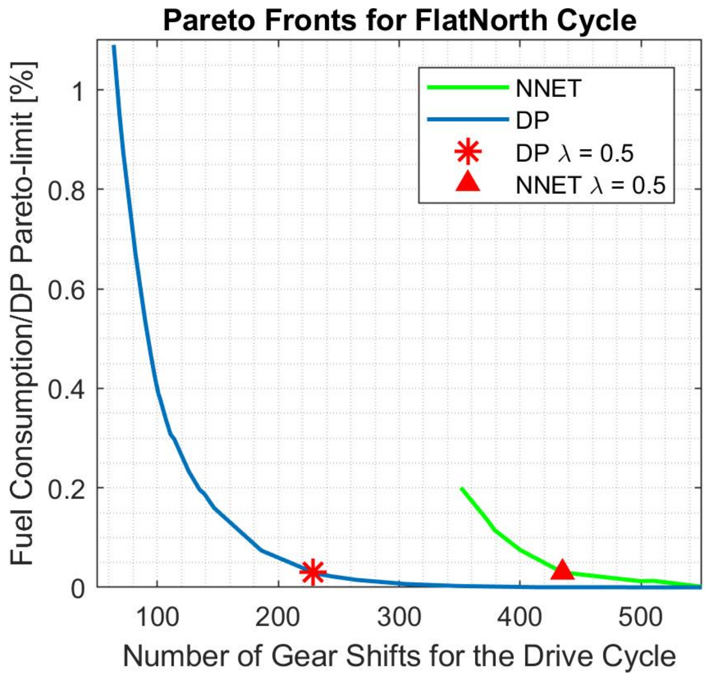


**Figure 4.3.** Pareto fronts comparison of DP and NNET of FlatNorth drive cycle.

clearly the difference between the Pareto-limit gear schedule in Figure 3.4 and the neural network gear schedule in Figure 4.5. The neural network gear shifts less frequently than the Pareto-limit. The fact that the neural network is able to predict the correct gear ratio for a larger fraction of the time in highway scenarios is consistent with the smaller variations in vehicle speed and wheel torque during highway driving.

Table 5.3 extends the above comparison between the DP-based and neural network-based shift strategies ($\lambda = 0.5$) in two ways. First, instead of comparing the gear schedules generated by the two strategies, Table 5.3 compares the corresponding trip fuel consumptions and gear shift frequency. Second, Table 5.3 performs this comparison for all six drive cycles used for one offline-trained the neural network, as opposed to just two representative examples. For every duty cycle, Table 5.3 lists the percent increase in fuel consumption compared to the static shift map when using DP with penalty 0.5 or the neural network for shift scheduling. The trip fuel consumption corresponding to the neural network remains within approximately 0.1% of the Pareto limit trip fuel consumption for all these 6 duty cycles. In city or mixed drive scenarios, the total trip fuel consumption with the neural network lies

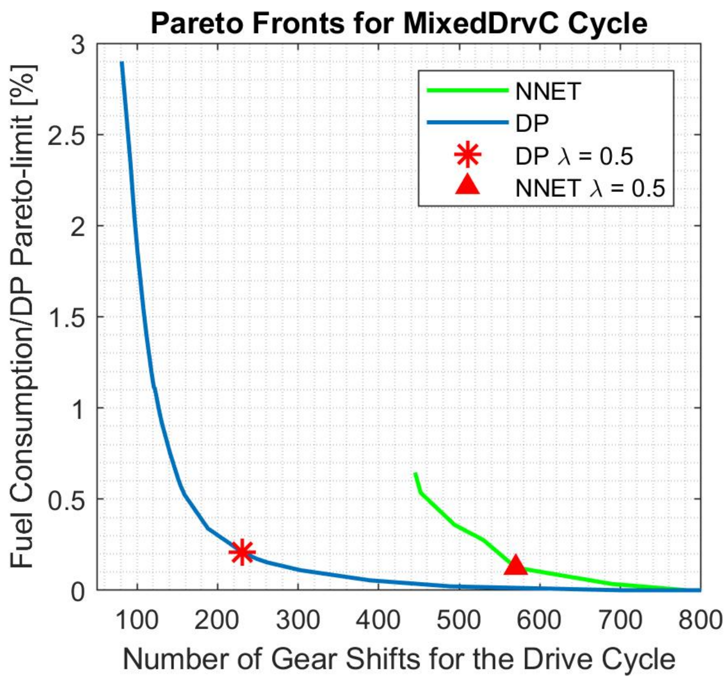


**Figure 4.4.** Pareto fronts comparison of DP and NNET of MixedDrvc drive cycle.

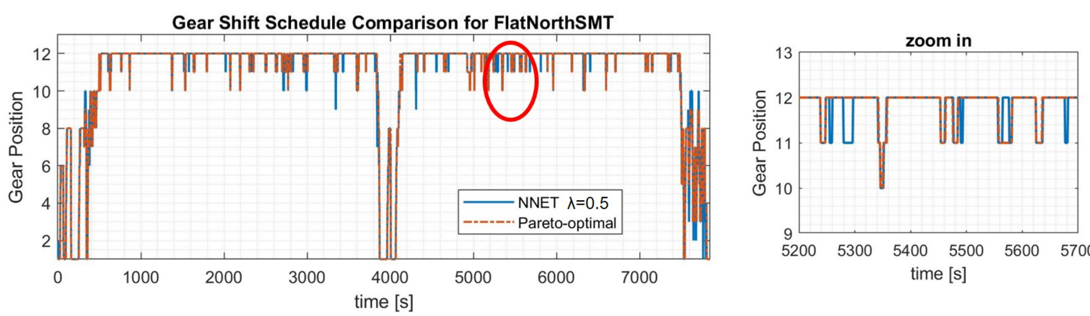


**Figure 4.5.** Gear shift schedule comparison between neural network and DP Pareto optimum of FlatNorth cycle.

between the Pareto-limit and Pareto-optimal levels, while in highway, the neural network outcome can successfully keep almost same fuel consumption with DP Pareto optimum. This is a very important result, because it shows our neural network's success in achieving fuel consumption levels comparable to DP, at a much lower computational cost. However, with regards to gear shift frequency, the neural network shifts gears roughly twice as the DP training data for $\lambda = 0.5$. Gear shifts occur roughly every 6-8 seconds in the city and every 16 seconds on highway. These results reveal one disadvantage of the neural network, namely, the significant

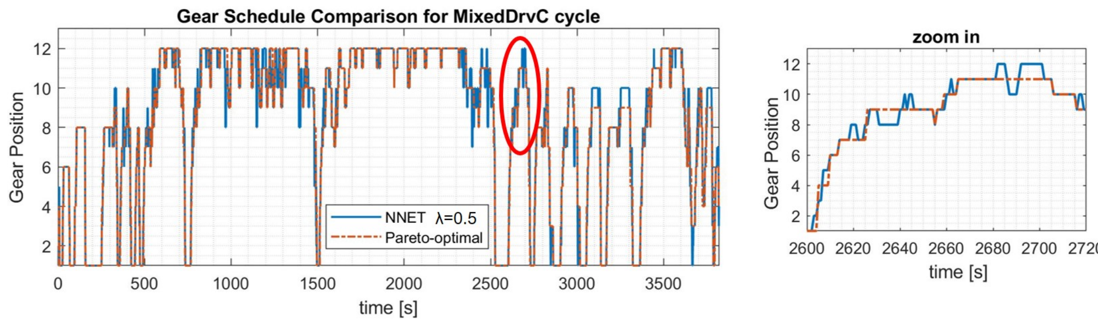


**Figure 4.6.** Gear shift schedule comparison between neural network and DP Pareto optimum o MixedDrvC cycle.

corresponding deterioration in the second objective (i.e., the gear shift frequency).

# Chapter 5 MRFC Algorithm

## 5.1 Introduction of the MRFC Algorithm

Rain flow counting is a well-developed method for analyzing the fatigue failures of materials under time-varying stress cycles [27, 28]. The ideas behind rain flow counting have recently been applied in other domains, one example being the analysis of electric vehicle battery degradation under time-varying loads [29]. One can use this method to count the number of closed-loop cycles inherently present in complex load cycles. The rain flow counting algorithm is able to recognize the up-down patterns in a data set and memorize the initial strain position and the magnitude of the varying load.

As mentioned in Chapter 3, the static gear shift map coincides with, and can therefore easily substitute for, the globally optimal shift schedule obtained from dynamic programming when gear frequency is not an optimization objective. However, one can never implement this gear shift schedule in practical vehicles because of the numerous up-down shift actions. A need arises for "filtering" the Pareto-limit shift schedules generated by the static shift map prior to using them in practice, in order to prevent unnecessary gear hunting. In this Chapter, we develop a modified rain flow counting algorithm that performs this filtering action by "flattening" the shift schedules generated by the static map. Our proposed rain flow counting algorithm is conceptually quite simple, and can be understood using the arbitrary gear shift schedule in Fig. 5.1. Suppose that this is the Pareto-limit gear shift schedule generated by the static shift map using the wheel speed and torque trajectories demanded by a higher-level powertrain controller. First, we

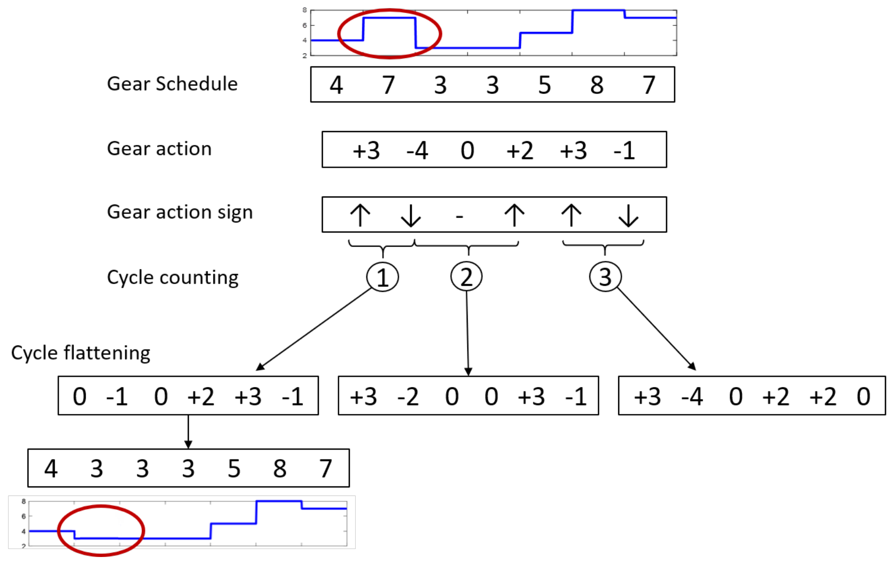


**Figure 5.1.** One example of the modified rain flow counting process.

extract the gear change actions at different moments in time by subtracting every pair of consecutive gear positions. Second, we record the signs of each gear change action. Third, we count every consecutive pair of up/down or down/up actions as one cycle. This furnishes a "modified" rain flow counting algorithm that is subtly different from classical rain flow counting in terms of how it accounts for closed versus open cycles. Intuitively, a closed cycle is one where the gear shifts back to the original position or shifts past this original position. The total cycle count in classical rain flow counting is the sum of the total number of closed cycles. We, however, neglect the actual values of the gear position when performing our cycle count. This means that we count all gear shifting cycles regardless of whether they are open or closed. The actual values of the vehicle gear positions are considered later in the filtering/flattening process. Given the above cycle counting process, one can "flatten" any given shift cycle by subtracting the same number of gear shifts from a consecutive pair of up-shift and down-shift events until one of those events no longer occurs. For example, instead of shifting upwards by 3 gear positions then shifting downwards by 4 gear positions, a "flattened" schedule would not shift upwards at all, and would then only shift down by 1 gear. In a gear shift schedule with multiple cycles, flattening any given cycle generates a "child" schedule, with the original schedule serving as the "parent". By generating all the potential children of the Pareto-limit shift schedule, and then generating additional generations of

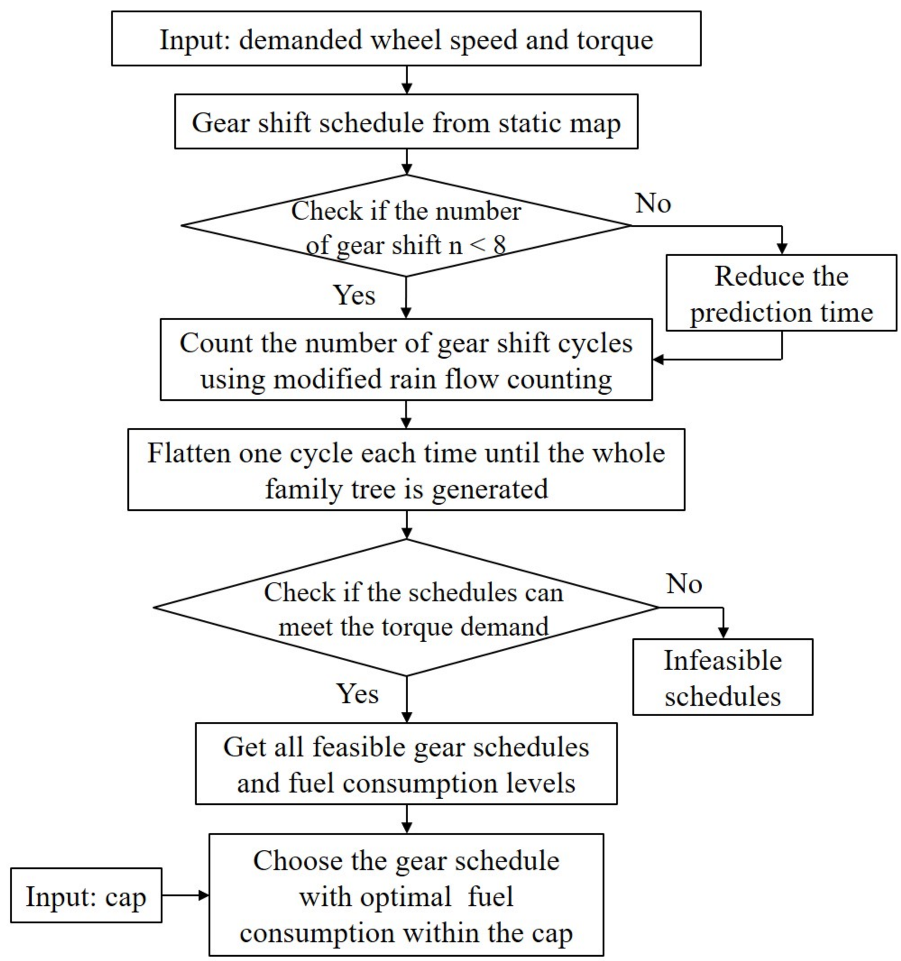


**Figure 5.2.** Flowchart of the modified rain flow counting algorithm.

children recursively, one eventually obtains a full "family tree" of flattened shift schedules.

Two final modifications are necessary for the practical implementation of the above rain flow algorithm. First, the algorithm's computational complexity grows rapidly with the number of gear shifts. In fact, if the static shift map dictates $n$ gear shifts for a given prediction horizon, the algorithm may require as many as $2^n \times n!$ computations for that horizon. We address this tractability challenge by adaptively reducing the length of the prediction horizon if $n$ exceeds 8. Second, there is no guarantee that the shift schedule furnished by the above rain flow algorithm corresponds to feasible engine torque and speed trajectories versus time. We address this problem by removing all gear schedules that violate engine torque and speed limits *a priori*, before implementing the rain flow method.

Before implementing the above rain flow algorithm, this thesis defines three key parameters that influence its results but can be easily adjusted for different practical implementations, namely, the prediction time $t_p$, implementation time $t_i$ and gear shift cap $c$. The prediction time $t_p$ is how long we predict into the future when performing cycle flattening. It determines the length of the static gear shift schedule required as input to the rain flow counting algorithm. This time horizon should not be larger than the prediction horizon used in upper-level predictive powertrain controllers. The implementation time $t_i$ is the duration of time over which the gear schedule generated by the rain flow algorithm is implemented before the algorithm needs to be run again. Finally, the cap $c$ constrains the number of gear shifts allowable over the prediction horizon (into the future) plus some look-back time (into the past). The idea, here, is that more frequent gear shifting may be feasible over the prediction horizon if the vehicle has not shifted gears frequently over the recent past. The rain flow counting algorithm is constrained to only select those members of the flattened gear schedule family tree that meet this cap. Among those gear shift schedules, the algorithm chooses the schedule with the lowest predicted fuel consumption. The only exception to this rule occurs when all such flattened gear schedules violate engine torque/speed constraints, in which case the gear schedule with the lowest number of gear shifts is chosen instead. The flow chart in Figure 5.2 shows the process for obtaining the optimal gear shift schedule using the modified rain flow counting strategy.

## 5.2 Pareto Fronts and Simulation Results

The results of modified rain flow counting strategy with different cap values manifest the trade-off between fuel economy and gear shift frequency. Lower cap value leads to less frequent gear shifts and higher fuel consumption. Prediction time and implement time only slightly influence the results. Table 5.1 shows the key parameters in this simulation study. We analyze the Pareto fronts of the modified rain flow counting algorithm using cap values of 2 to 30. Because the DP-based neural network has a 5-second implementation time horizon, we choose the same value to make sure the results are comparable. Figure 5.3 and Figure 5.4 show the comparison of three Pareto fronts of DP, NNET and modified rain flow counting for the FlatNorth and MixedDrvC cycles. For both highway and city scenarios, the

**Table 5.1.** Modified Rain Flow Counting Parameters

| **Parameter Name** | **Value** |
|---|---|
| **Predict time** $t_p$ | 20s |
| **Implement time** $t_i$ | 5s |
| **Cap** $c$ | $2 \sim 30$ |

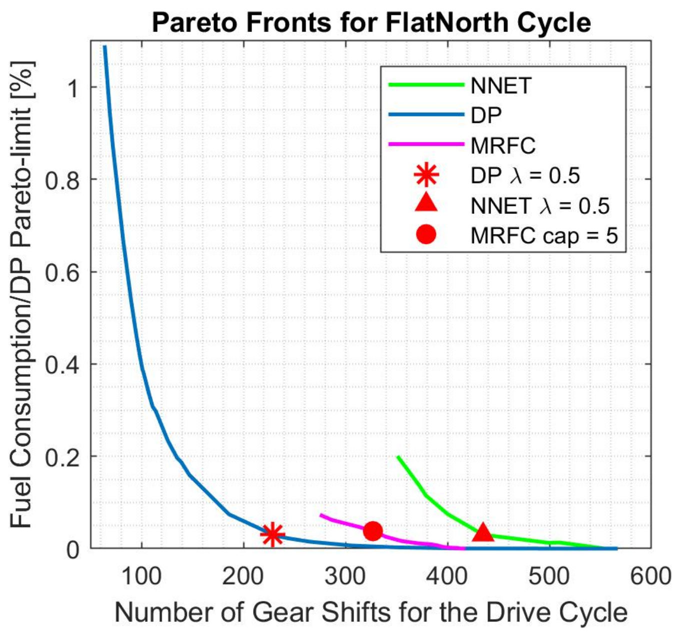


**Figure 5.3.** Pareto fronts comparison of DP, NNET and modified rain flow counting of FlatNorth cycle.

modified rain flow counting strategy furnishes a closer fit to the DP Pareto Front. The limitation of this method is that the cap value cannot be lower than 2 within the counting time window. In contrast, the neural network can be trained with a higher penalty coefficient and lead to a bigger compromise of the fuel consumption.

Here we choose $c = 5$ as an implementation case to show the detailed simulation results comparison of the gear shift schedules. Red dots in Figure 5.3 and Figure 5.4 present the resulting Pareto front locations of the modified rain flow counting strategy for this specific case. The corresponding 20-second prediction time horizon means we constrain the gear shift number to be less or equal than 5 in a 40-second time window. The choice of a average of 8 seconds per gear shift ensures a fair comparison between this algorithm and DP Pareto-optimum. If a reduced prediction time window is necessary, we also adaptively reduce $t_p$ by 8 seconds and adjust the

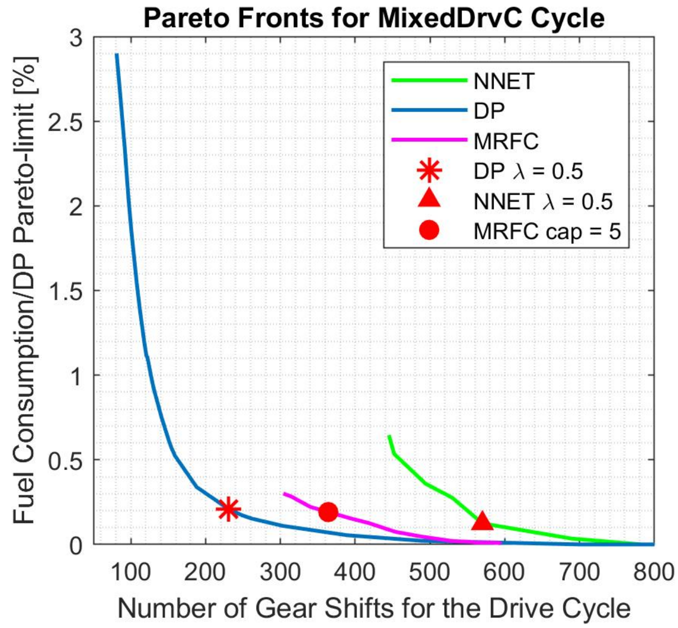


**Figure 5.4.** Pareto fronts comparison of DP, NNET and modified rain flow counting of MixedDrvC cycle.

corresponding cap value (e.g., the predict time can be 12 seconds with a cap value 4 in the whole 32-second window, etc.).

One can observe the gear shift schedule difference between the DP Pareto limits and the modified rain flow counting schedules in Figure 5.5 and Figure 5.6. This innovative counting and flattening strategy is capable of reducing the gear shift frequency in a manner that also satisfies the requirements of vehicle speed control. In the same zoom-in time horizon (2600s to 2720s) of MixedDrvC cycle, the neural network produces 26 gear shifts, while the modified rain flow counting algorithm only gives 16 gear shifts. This is not sufficient to prove that at each interval the rain flow algorithm achieves fewer shift actions, however in the whole trip, the gear shift frequency from the modified rain flow counting algorithm is indeed lower than the gear schedule provided by the DP-based neural network. The fuel consumption level stays quite comparable between these two methods. Table 5.3 lists both objectives for all the drive cycles using the above three algorithms. One can see the degree to which the modified rain flow counting algorithm outperforms the neural network in terms of reducing gear shift frequency. The total trip fuel consumption corresponding to modified rain flow counting remains within approximately 0.2% of the Pareto limit trip fuel consumption for all these 6 duty cycles. This shows that

Table 5.2. Number of gear shifts comparison of different strategies

| Total Number of Gear Shifts for the Drive Cycle | | | | |
|---|---|---|---|---|
| **Drive Cycle** | **DP** $\lambda = 0.5$ | **NNET** $\lambda = 0.5$ | **MRFC** $cap = 5$ | **Volvo VNL300 Strategy** |
| FlatNorth | 229 | 435 | 327 | 204 |
| MixedDrvC | 231 | 570 | 364 | 393 |

the modified rain flow counting algorithm succeeds in delivering fuel consumption levels comparable to DP.

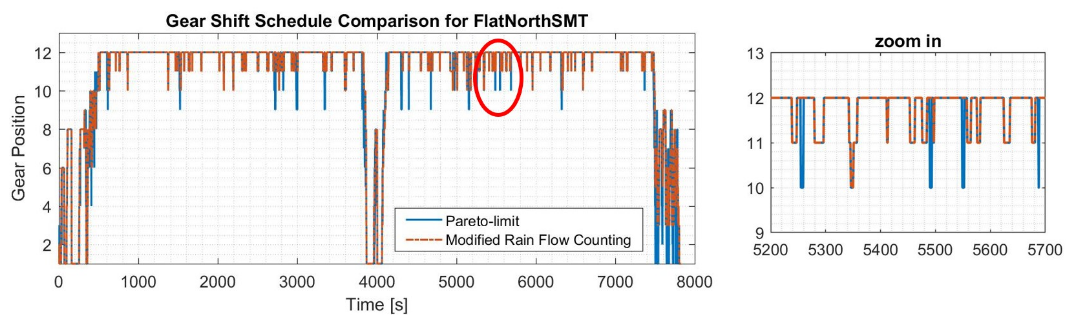


**Figure 5.5.** Gear shift schedule comparison between modified rain flow counting algorithm and DP Pareto-limit of FlatNorth cycle.

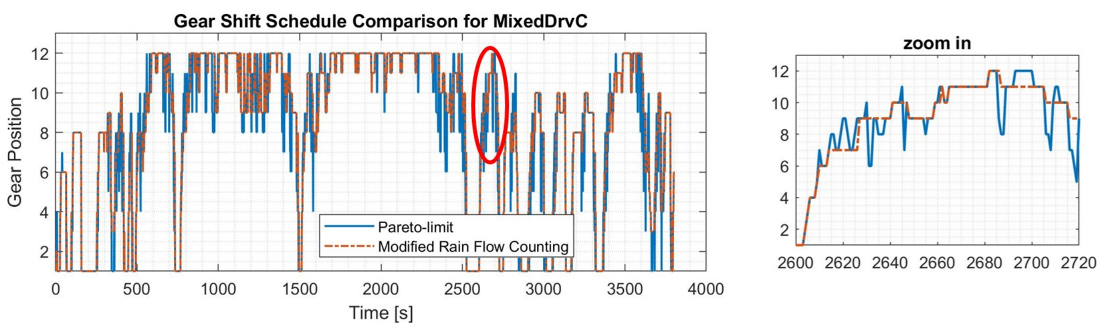


**Figure 5.6.** Gear shift schedule comparison between modified rain flow counting algorithm and DP Pareto-limit of MixedDrvC cycle.

Table 5.2 compares this thesis's gear shift frequency results to the outputs of a benchmark gear shift strategy used by Volvo for heavy-duty truck chassis and driveline simulation. Compared to the neural network, the modified rain flow algorithm is much closer to both the DP-optimal solution and the benchmark algorithm in terms of gear shift frequency, for both urban and highway driving.

Table 5.3. FUEL CONSUMPTION and gear frequency COMPARISON OF DIFFERENT DRIVE CYCLES

| | **DP** $\lambda = 0.5$ | | **NNET** $\lambda = 0.5$ | | **MRFC** $cap = 5$ | |
|---|---|---|---|---|---|---|
| **Drive Cycle** | **Obj. 1** | **Obj. 2** | **Obj. 1** | **Obj. 2** | **Obj. 1** | **Obj. 2** |
| 81Marion | 0.130% | 17.55 | 0.100% | 7.62 | 0.109% | 10.36 |
| FlatNorth | 0.030% | 34.27 | 0.037% | 16.91 | 0.038% | 23.94 |
| ILwest | 0.022% | 70.28 | 0.021% | 38.25 | 0.035% | 47.53 |
| INwest | 0.012% | 76.74 | 0.015% | 42.52 | 0.023% | 59.77 |
| MixedDrvC | 0.179% | 15.68 | 0.101% | 6.07 | 0.172% | 10.45 |
| OHwest | 0.028% | 46.17 | 0.018% | 25.11 | 0.021% | 32.37 |

Obj.1: Fuel consumption increase percentage compared to DP Pareto limit
Obj.2: Gear shift frequency of the drive cycle, average time [s] per shift.

The algorithm shifts gears more frequently than the benchmark on the highway, and less frequently in the urban scenario. These differences in gear shift frequency represent both the specific tuning of the algorithm and, more importantly, the desire to retain as much as possible of the fuel consumption benefit achievable through optimization algorithms such as DP.

We compare the total computation time for each cycle in Matlab for these three strategies. The results are shown in Table 5.4. Three primary observations are listed below. First, both the neural network and the modified rain flow counting strategy can significantly reduce the elapsed time compared to DP. Second, the modified rain flow counting has much better performance in terms of computation time. Third, the total computation time for the neural network is proportional to the trip duration time, while the modified rain flow counting algorithm's computation time depends on the gear shift frequency from the static map within the prediction horizon. This is the reason why 81Marion and MixedDrvC cycles with shorter trip time have slightly longer computation time using the modified rain flow counting strategy.

Table 5.4. Total Computation time comparison of different Drive Cycles

| Total Computation Time for the Drive Cycle [s] | | | | |
|---|---|---|---|---|
| **Drive Cycle** | **Duration** | **DP** | **NNET** | **MRFC** |
| 81Marion | 2175 | 139.5 | 4.60 | 0.109 |
| FlatNorth | 7847 | 301.8 | 13.94 | 0.038 |
| ILwest | 13660 | 430.3 | 18.44 | 0.030 |
| INwest | 9822 | 367.9 | 14.72 | 0.023 |
| MixedDrvC | 3824 | 144.1 | 5.60 | 0.171 |
| OHwest | 10248 | 379.6 | 15.00 | 0.020 |

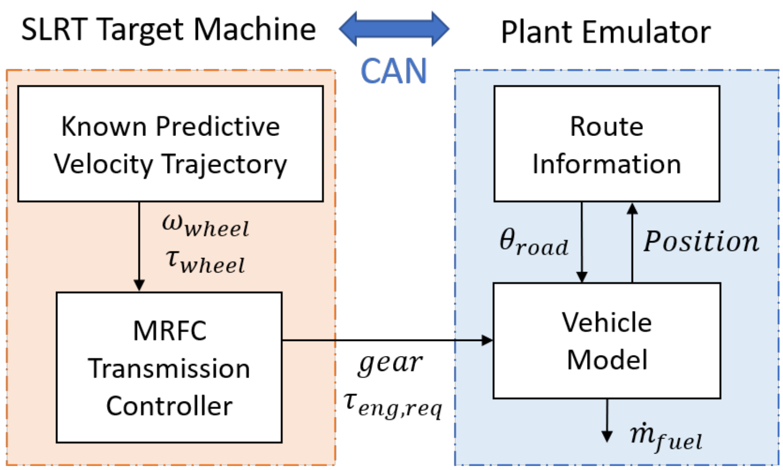


**Figure 5.7.** Simulink Real-time Validation Structure.

## 5.3 Validation on SLRT Target Machine

To validate the ability of the modified rain flow strategy to run online (in real time), we set up a Simulink Real-time setup conceptually sketched in Figure 5.7. The MRFC gear shift controller runs in a Simulink Real-time (SLRT) R2016b environment on a target machine. A control-oriented model of the VNL300 vehicle system [30] (including engine, transmission and chassis) runs in real time on a separate dedicated machine as the plant model emulator, so as to isolate the control algorithms and provide a quantifiable measure of the real-time computational performance of the rain flow algorithm. Communication between the two real-time machines is accomplished via the Controller Area Network (CAN) protocol.

The above SLRT setup produces identical gear shift schedules and fuel consumption levels as offline simulations, while running successfully in hard real time. More specifically, the above hardware-in-the-loop study shows that the controller needs a maximum of 0.428ms to run one cycle of the MRFC-based gear shift scheduling algorithm. Given the fact that the algorithm is set up to run one such optimization cycle every second, significant computational capacity remains available for other, higher-level optimizers (e.g., speed trajectory optimizers, etc.).

# Chapter 6 Conclusion

This thesis investigates the application of DP, a DP-based neural network and a modified rain flow counting strategy, to furnish and compare computationally tractable yet effective real-time control policies for heavy-duty vehicle gear shift scheduling. A DP problem is formulated using the Volvo VNL300 truck's parameters, and the Pareto front of DP depends on the foreknown drive cycle information. To implement the strategy online, DP results for a set of cycles are used to train a neural network. The neural network shows success in achieving gear shift schedules comparable to the DP Pareto optimum. Furthermore, we develop a modified rain flow counting strategy to compare with the DP-based neural network. For the chosen DP penalty coefficient and rain flow cap, both these two online policies compromise the total fuel consumption within only 0.2%. Moreover, we compare the Pareto fronts of DP, DP-based neural networks and the modified rain flow counting algorithm of representative drive cycles. On the one hand, the latter fits closer to the DP Pareto front in both representative cycles with the limitation of further reduction of the gear shift frequency. On the other hand, the Pareto fronts of DP-based neural networks have slightly bigger distance to DP, but they can be extended by choosing larger scalarization weight $\lambda$. These two algorithms are particularly attractive given the computational benefits associated with using the neural network or modified rain flow counting strategy online instead of DP. The overall conclusion of this study is that the DP-based neural network and modified rain flow shift strategies are both viable, and can indeed form foundations for online predictive gear scheduling.

# Bibliography

[1] Kamal, M. A. S., T. Hayakawa, and J. i. Imura (2018) "Road-Speed Profile for Enhanced Perception of Traffic Conditions in a Partially Connected Vehicle Environment," *IEEE Transactions on Vehicular Technology*, pp. 1–1.

[2] Hellström, E., M. Ivarsson, J. Åslund, and L. Nielsen (2009) "Look-ahead control for heavy trucks to minimize trip time and fuel consumption," *Control Engineering Practice*, **17**(2), pp. 245–254.

[3] Hellström, E., A. Fröberg, and L. Nielsen (2006) *A real-time fuel-optimal cruise controller for heavy trucks using road topography information*, *Tech. rep.*, SAE Technical Paper.

[4] Ivarsson, M., J. Åslund, and L. Nielsen (2010) *Impacts of AMT gear-shifting on fuel optimal look ahead control*, *Tech. rep.*, SAE Technical Paper.

[5] Kim, D., H. Peng, S. Bai, and J. M. Maguire (2007) "Control of integrated powertrain with electronic throttle and automatic transmission," *IEEE Transactions on Control Systems Technology*, **15**(3), pp. 474–482.

[6] Lei, Y., Y. Fu, K. Liu, L. Xingzhong, Z. Liu, Y. Zhang, and X. Fu (2017) *Research on Optimal Gearshift Strategy for Stepped Automatic Transmission Based on Vehicle Power Demand*, *Tech. rep.*, SAE Technical Paper.

[7] Ngo, D., T. Hofman, M. Steinbuch, A. Serrarens, and L. Merkx (2010) "Improvement of fuel economy in Power-Shift Automated Manual Transmission through shift strategy optimization-an experimental study," in *Vehicle Power and Propulsion Conference (VPPC), 2010 IEEE*, IEEE, pp. 1–5.

[8] Škugor, B., J. Deur, and V. Ivanović (2016) *Dynamic programming-based design of shift scheduling map taking into account clutch energy losses during shift transients*, *Tech. rep.*, SAE Technical Paper.

[9] Kolmanovsky, I., I. Siverguina, and B. Lygoe (2002) "Optimization of powertrain operating policy for feasibility assessment and calibration: Stochastic dynamic programming approach," in *American Control Conference, 2002. Proceedings of the 2002*, vol. 2, IEEE, pp. 1425–1430.

[10] Ngo, V. D., J. A. Colin Navarrete, T. Hofman, M. Steinbuch, and A. Serrarens (2013) "Optimal gear shift strategies for fuel economy and driveability," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, **227**(10), pp. 1398–1413.

[11] Liu, B., L. Li, X. Wang, and S. Cheng (2018) "Hybrid Electric Vehicle Downshifting Strategy based on Stochastic Dynamic Programming during Regenerative Braking Process," *IEEE Transactions on Vehicular Technology*, pp. 1–1.

[12] Fröberg, A. and L. Nielsen (2007) "Optimal fuel and gear ratio control for heavy trucks with piece wise affine engine characteristics," *IFAC Proceedings Volumes*, **40**(10), pp. 335–342.

[13] Lu, J., S. Hong, J. Sullivan, G. Hu, E. Dai, D. Reed, and R. Baker (2017) "Predictive transmission shift schedule for improving fuel economy and drivability using electronic horizon," *SAE International Journal of Engines*, **10**(2017-01-1092), pp. 680–688.

[14] Guo, L., B. Gao, Q. Liu, J. Tang, and H. Chen (2017) "On-line Optimal Control of the Gearshift Command for Multispeed Electric Vehicles," *IEEE/ASME Transactions on Mechatronics*, **22**(4), pp. 1519–1530.

[15] Guo, L., B. Gao, Y. Gao, and H. Chen (2017) "Optimal Energy Management for HEVs in Eco-Driving Applications Using Bi-Level MPC," *IEEE Transactions on Intelligent Transportation Systems*, **18**(8), pp. 2153–2162.

[16] Liu, Y., D. Qin, H. Jiang, C. Liu, Y. Zhang, and Z. Lei (2009) "Shift schedule optimization for dual clutch transmissions," in *Vehicle Power and Propulsion Conference, 2009. VPPC'09. IEEE*, IEEE, pp. 1071–1078.

[17] Mashadi, B., A. Kazemkhani, and R. B. Lakeh (2007) "An automatic gear-shifting strategy for manual transmissions," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, **221**(5), pp. 757–768.

[18] Yamaguchi, H., Y. Narita, H. Takahashi, and Y. Katou (1993) *Automatic transmission shift schedule control using fuzzy logic*, *Tech. rep.*, SAE Technical Paper.

[19] Hayashi, K., Y. Shimizu, Y. Dote, A. Takayama, and A. Hirako (1995) "Neuro fuzzy transmission control for automobile with variable loads," *IEEE Transactions on Control Systems Technology*, **3**(1), pp. 49–53.

[20] Wu, S. and H. Lu (2006) "Fuzzy neural network control in automatic transmission of construction vehicle," in *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, vol. 1, IEEE, pp. 2512–2516.

[21] Tan, J., X. Yin, Y. Lei, and A. Ge (2005) *Research on a neural network model based automatic shift schedule with dynamic 3-parameters*, *Tech. rep.*, SAE Technical Paper.

[22] Qin, G., A. Ge, and J.-J. Lee (2004) "Knowledge-based gear-position decision," *IEEE transactions on intelligent transportation systems*, **5**(2), pp. 121–125.

[23] Yin, X., W. Wang, X. Chen, and H. Lu (2013) *Multi-Performance Optimization of the Shift Schedule for Stepped Automatic Transmissions*, *Tech. rep.*, SAE Technical Paper.

[24] Groelke, B., J. Borek, C. Earnhardt, J. Li, S. Geyer, and C. Vermillion (2018) "A Comparative Assessment of Economic Model Predictive Control Strategies for Fuel Economy Optimization of Heavy-Duty Trucks," in *2018 Annual American Control Conference (ACC)*, IEEE, pp. 834–839.

[25] Xu, C., A. Al-Mamun, S. Geyer, and H. K. Fathy (2018) "A Dynamic Programming-Based Real-Time Predictive Optimal Gear Shift Strategy for Conventional Heavy-Duty Vehicles," in *American Control Conference (ACC), 2018*, IEEE, pp. 5528–5535.

[26] Bellman, R. (1954) *The theory of dynamic programming*, *Tech. rep.*, RAND Corp Santa Monica CA.

[27] Matsuishi, M. and T. Endo (1968) "Fatigue of metals subjected to varying stress," *Japan Society of Mechanical Engineers, Fukuoka, Japan*, **68**(2), pp. 37–40.

[28] Downing, S. D. and D. Socie (1982) "Simple rainflow counting algorithms," *International journal of fatigue*, **4**(1), pp. 31–40.

[29] Schaltz, E., A. Khaligh, and P. O. Rasmussen (2008) "Investigation of battery/ultracapacitor energy storage rating for a fuel cell hybrid electric vehicle," in *Vehicle Power and Propulsion Conference, 2008. VPPC'08. IEEE*, IEEE, pp. 1–6.

[30] Chiara, F., J. Wang, C. B. Patil, M.-F. Hsieh, and F. Yan (2011) "Development and experimental validation of a control-oriented Diesel engine model for fuel consumption and brake torque predictions," *Mathematical and Computer Modelling of Dynamical Systems*, **17**(3), pp. 261–277.