ENERGY OPTIMIZATION OF AN IN-WHEEL-MOTOR ELECTRIC GROUND VEHICLE OVER A GIVEN TERRAIN WITH CONSIDERATIONS OF VARIOUS TRAFFIC ELEMENTS

THESIS

Presented in Partial Fulfillment of the Requirements for the Degree Master of Science in the Graduate School of The Ohio State University

By

Christopher James Wiet

Graduate Program in Mechanical Engineering

The Ohio State University

2014

Master's Examination Committee:

Dr. Junmin Wang, Advisor and Dr. Manoj Srinivasan
ABSTRACT

Finding an effective alternative energy source to fossil fuels is a significant concern in today’s primarily ground-transportation-heavy society. While hybrid-electric vehicles offer a good compromise of power and efficiency, pure electric vehicles are showing increasing promise as a transportation option that can utilize a truly renewable energy source. The problem up until now, however, has been that vehicles powered only by batteries have severely limited travel ranges. This thesis, therefore, investigates methods for optimizing the travel distance of a pure electric ground vehicle. Previous research has shown a two-stage optimization of both vehicle velocity as well as in-wheel motor torque distribution can offer up to a 25% increase in efficiency [1]. The next step, investigated here, is the effect that more realistic traffic conditions such as other vehicles and traffic lights have on the optimal solution. While these constraints obviously result in less energy savings, they still allow for a substantial increase over a naively actuated system.
DEDICATION

This document is dedicated to my family. I love them with all my heart!
ACKNOWLEDGMENTS

I would like to thank, first and foremost, my advisor Dr. Junmin Wang. He has been very helpful in not only providing a strong direction for my thesis, but also in providing excellent discussion and insight into any problems I encountered along the way. I also believe he does a great job of making his students feel at home, as he and his family were always been very hospitable during graduation parties and holidays. I have greatly enjoyed my time under his guidance, and again want to thank him for all he has done for me.

I would also like to thank my friends and family that were so supportive during my entire time as a graduate student. Without their understanding and motivation, I’m sure I would not be where I am today.
VITA

2004-2007 ...................................................... Worthington Kilbourne High School

2007-2012 ...................................................... B.S. Mechanical Engineering, The Ohio State University

2013-present................................................... Graduate Teaching Assistant, Department of Mechanical and Aerospace Engineering, The Ohio State University

PUBLICATIONS


FIELDS OF STUDY

Major Field: Mechanical Engineering

Optimal Control, Modeling and Control Electric Vehicle Systems
# TABLE OF CONTENTS

ABSTRACT ........................................................................................................... ii

DEDICATION ....................................................................................................... iii

ACKNOWLEDGMENTS ....................................................................................... iv

VITA .................................................................................................................... v

PUBLICATIONS .................................................................................................. v

FIELDS OF STUDY ........................................................................................... v

TABLE OF CONTENTS ...................................................................................... vi

LIST OF TABLES ............................................................................................... viii

LIST OF FIGURES ............................................................................................. ix

Chapter 1: Introduction ...................................................................................... 1

1.1 Background and Motivation ....................................................................... 1

1.2 Objective of this Research ....................................................................... 2

1.3 Organization of this Thesis ..................................................................... 3

Chapter 2: Background and Literature Review ............................................. 5

2.1 Introduction ............................................................................................. 5

2.2 Electric Motors ......................................................................................... 5

2.3 Alternative Energy Sources for Transportation .................................... 8

2.3.1 Present-Day Energy Source Distribution ....................................... 8

2.3.2 Hybrid Electric Vehicles .................................................................... 9

2.3.3 Pure Electric Vehicles ...................................................................... 10

2.4 Optimal Control ...................................................................................... 13

2.4.1 History of optimal control ................................................................. 13

2.4.2 Continuous framework ....................................................................... 15

2.4.3 Discrete framework .......................................................................... 19
Chapter 3:  Model of the EGV System

3.1 Introduction ........................................................................... 26
3.2 Physical Parameters of the EGV ............................................. 26
  3.2.1 Electrical Components ....................................................... 27
  3.2.2 Characterization of Motor Efficiency ............................... 28
  3.2.3 Mechanical Components .................................................... 30
3.3 Derivation of the Equations of Motion .................................... 30
3.4 Discretization of the Dynamic Model for Digital Implementation .... 33

Chapter 4:  Optimal Control Algorithm

4.1 Overall Structure ................................................................. 36
4.2 Dynamic Programming Loop .................................................. 38
  4.2.1 Physical System Constraints ............................................. 38
  4.2.2 General Structure ............................................................. 39
  4.2.3 Calculate Minimal Energy Consumption from State Change ... 41
4.3 Traffic Constraints ............................................................... 42
  4.3.1 Introduction .................................................................. 42
  4.3.2 Preceding Vehicle ............................................................ 43
  4.3.3 Traffic Lights ................................................................. 44

Chapter 5:  Simulation Results ....................................................... 46

5.1 Baseline Cases ...................................................................... 46
  5.1.1 Constant Speed ............................................................... 46
  5.1.2 Equal Actuation ............................................................... 48
5.2 Optimal Trajectory with an Open Road ................................... 50
5.3 Effect of Traffic Constraints on the Optimal Solution ............... 51
  5.3.1 Preceding Vehicle ............................................................ 51
  5.3.2 Traffic Lights ................................................................. 54

Chapter 6:  Conclusions and Future Work ..................................... 58

6.1 Summary ............................................................................. 58
6.2 Contributions ....................................................................... 58
6.3 Recommendations for Future Work ....................................... 59

References 61
LIST OF TABLES

Table 3.1: Parameters of the EGV’s Electrical Components ........................................ 28
Table 3.2: Four Cases for Characterizing Overall Motor Efficiency .......................... 29
Table 3.3: Physical Characteristics of the EGV .......................................................... 30
Table 4.1: Vehicle Constraints .................................................................................. 38
Table 5.1: Comparison of Open Road DP Solution to Baseline SOCs ....................... 51
LIST OF FIGURES

Figure 2.1: Free-Body Diagram of a DC Motor ................................................................. 6

Figure 2.2: Diagram of a Brushless DC Motor ................................................................. 7

Figure 2.3: Percent of US Transportation Sector Consumption, April 2013 [2] ............... 8

Figure 2.4: Illustration of Various HEV Powertrain Configurations: (a) Series, (b) Parallel, (c) Series-Parallel ................................................................................................. 9

Figure 2.5: Diagram of Tesla Model X Undercarriage [8] .................................................. 11

Figure 2.6: Example of Vehicle with 4 In-Wheel Electric Motors ..................................... 12

Figure 2.7: The Brachistochrone Problem [15] .................................................................. 13

Figure 2.8: Bernoulli’s Cycloid—the Solution to the Brachistochrone Problem .............. 14

Figure 2.9: (a) Optimal Path from $a$ to $e$, (b) Two Possible Optimal Paths from $b$ to $e$ [16] ......................................................................................................................... 19

Figure 2.10: Shortest Path Example ..................................................................................... 21

Figure 2.11: Flow Chart of the Computational Procedure [16] ........................................ 24

Figure 3.1: Prototyping EGV at the VSCL ....................................................................... 26

Figure 3.2: In-wheel motors before and after installation ..................................................... 27

Figure 3.3: Li-ion Battery Pack installed in the EGV ........................................................ 27

Figure 3.4: Efficiency Maps for Driving and RegenerativeBraking of an In-Wheel Motor ................................................................................................................................. 29

Figure 3.5: FBD of the EGV on a Slope .............................................................................. 31
Figure 3.6: Aerial view of the Chosen Path ................................................................. 34
Figure 3.7: Possible Speed Profiles between Discrete Nodes .......................................... 34
Figure 4.1: Top Level of Optimal Control Algorithm ....................................................... 36
Figure 4.2: Terrain Profile Used in Simulations .............................................................. 37
Figure 4.3: Find Optimal Speed Block Diagram ............................................................. 40
Figure 5.1: Ending SOC for Constant Speed Case (1-50 km/h) ........................................ 46
Figure 5.2: Ending SOC for Constant Speed Case (25-40 km/h) ..................................... 47
Figure 5.3: Comparison of Torques between CV and EA Cases ....................................... 49
Figure 5.4: Comparison of Ending SOC between CV and EA Cases .............................. 49
Figure 5.5: Open Road Optimal Solution ........................................................................ 50
Figure 5.6: Effect of Preceding Vehicle Speed on EGV Speed Profile ............................ 52
Figure 5.7: Optimal Solution with Preceding Vehicle Energy Gains ............................... 53
Figure 5.8: DP results with Single Traffic Light at 1500m ............................................. 55
Figure 5.9: DP results with Traffic Light at Various Positions: ........................................ 56
CHAPTER 1: INTRODUCTION

1.1 Background and Motivation

Optimizing and conserving energy is a large focus of society today. This is largely due, in part, to a predicted energy shortage if global consumption continues on its current trajectory [2]. Many ways of saving energy, therefore, have been researched and implemented to this point, such as super-efficient windows for large buildings, lights bulbs that last years and use minimal electricity, as well as smarter manufacturing technologies such as optimizing blast furnace operation [3]. Some of the most significant efforts in energy saving, however, has been put into the area of industrial and personal transportation. The large majority of vehicles still run on fossil fuels, but an increasing amount of research has been put into developing and optimizing alternative energy sources such as electric, biofuels, natural gas, and even solar power for transportation. One of the more promising solutions for ground vehicles seems to be electric power, as many car manufacturers offer at least hybrid electric (HEVs), if not a variety of pure electric vehicles (PEVs). Hybrid vehicles attempt to mate the utility of a gasoline/diesel engine with the environmentally friendly characteristics of an electric drive. In order to push the envelope of sustainability even further, however, an increasing effort has been put into developing vehicles that run on pure electric power [4]. The research presented in this thesis attempts to add to this effort by creating a simulation framework that
predicts and estimates the effects of various optimization techniques on an electric ground vehicle (EGV), with the ultimate goal of augmenting the real-world utility of such vehicles.

1.2 Objective of this Research

There are two primary objectives of this work. The first is to develop a model of a driving scenario that takes into account the dynamics of a pure electric ground vehicle in as realistic a setting as possible. This includes modeling the efficiency of the four in-wheel motors, the effect of various terrain profiles on the vehicle’s behavior, as well as offering as realistic as possible models of various traffic elements. Specifically, two different elements will be considered: a preceding vehicle that cannot be passed, as well as the simulation of a traffic light. All these elements are then gathered and utilized in the next objective.

The second primary objective is to develop a dynamic programming algorithm in MATLAB that will solve for the optimal combination of vehicle speed and in-wheel motor torques across a given terrain profile. As stated above, this program will seek to utilize all the models developed in order to simulate as realistic a driving scenario as possible. Therefore, the program should take various initial condition inputs as well as various traffic states, and it should be able to effectively simulate the system response and resolve the best possible solution over the entire domain. The results of this work, and the evaluation of these objectives are discussed in the subsequent chapters.
1.3 Organization of this Thesis

The remainder of this thesis is organized as follows. Chapter 2 presents a technical background of the topics important in the derivation of a model and the control of a pure electric ground vehicle. First a brief introduction to hybrid-electric and pure electric vehicles is given. Next the various types of pure electric vehicles and a description of their motors is discussed. Then the specific vehicle used in this research is introduced. Finally, background on various optimal control methods are presented and discussed.

Chapter 3, in turn, presents a detailed discussion of the model for the physical system. First the physical components of the in-wheel motor pure electric vehicle at the Ohio State University Vehicle Systems and Control Laboratory are presented. Then the simplified model and resulting equations of motion are derived, followed by the methods for its implementation into the computer program.

In Chapter 4, the theory and methods of the optimal control algorithm used to solve for the optimal motor torques are given. First the overall structure of the dynamic programming algorithm is presented, followed by a detailed description for each of its subsystems. Also explained here are the methods for implementing various traffic models and constraints into the system.

Chapter 5 gives the results of the optimal control algorithm applied to various initial conditions and driving scenarios. First, various baseline cases with naïve vehicle control are presented and discussed. Then the resulting optimal trajectories over an open
road are discussed, followed by more complicated cases involving the addition of a preceding vehicle and traffic lights.

Finally, conclusions of the research as well as recommendations for possible future work are given in Chapter 6.
CHAPTER 2: BACKGROUND AND LITERATURE REVIEW

2.1 Introduction

When attempting to solve for the optimal trajectory of a given system, it is important to not only be able to accurately model the system, but also utilize the various methods for finding optimal solutions effectively. Therefore, an overview of electric motors and how they are typically used in ground propulsion is given here, along with a description of various methods of optimal control.

2.2 Electric Motors

Converting electro-magnetic energy to mechanical energy has been a topic of interest pretty much since the discovery of electricity and magnetism. The first electric motor is typically attributed to Michael Faraday in 1821, and it consisted of metal wire suspended in cups of mercury which contained permanent magnets at their base [5]. Further improvements were made by people such as Nikola Tesla [6], and eventually both direct and alternating current motors and generators became extremely important contributions to the modernization of society. Since then, many iterations and significant improvements have been made, but the fact that a changing electric field results in a changing magnetic field remains the same basic concept on which all electric motors are built.

A simplified free-body diagram of a DC motor is given in Figure 2.1.
where $L$, $R$, and $i$ are armature inductance, resistance and current, respectively, $e$ is the back emf generated in the circuit from the movement of the rotor, $J$ is the moment of inertia of the rotor, $b$ is the motor viscous friction constant, $\theta$ is the rotation angle of the rotor, and $T$ is the motor load torque. Also included in this model are constant factors, $K_t$ and $K_e$, that represent the motor torque constant and electromotive force constant, respectively. They relate torque and back emf to current and angular velocity as follows:

$$T = K_t i$$
$$e = K_e \dot{\theta}$$

For simplicity, as well as the fact that $K_t = K_e$ in general, $K$ will be used to hereafter denote both parameters. From Newton’s 2nd Law and Kirchhoff’s voltage law, the governing equations can thus be written as follows:

$$J\ddot{\theta} + b\dot{\theta} = Ki$$
$$L \frac{di}{dt} + Ri = V - K\dot{\theta}$$
The topic discussed at length in this thesis is finding the optimal motor torque and speed such that consumption of electrical energy is minimized. Understanding these basic equations that relate the mechanical and electrical characteristics of a DC motor, therefore, is key.

The specific type of motor used in this research is called a brushless direct current (BLDC) motor. There are many possible configurations of such a motor, but the basic concept is that a permanent magnet is surrounded by numerous coiled wires, each with a conducting core. The current supplied to these surrounding wires is then alternated, resulting in a cyclical magnetic force which, in turn, generates a rotary motion of the central magnet. A diagram of a BLDC motor is given in Figure 2.2.
2.3 Alternative Energy Sources for Transportation

2.3.1 Present-Day Energy Source Distribution

Due to the finite availability of fossil fuel resources, as discussed in Section 1.1, the search for alternative energy sources for ground transportation is a large focus of society today. While fossil fuels still claim over 95% of the energy consumption from the transportation sector in the US (Figure 2.3), alternative energy sources such as natural gas, hydrogen, biodiesel, ethanol, electric, and even solar power are claiming a more significant percentage each year.

Figure 2.3: Percent of US Transportation Sector Consumption, April 2013 [2]

While ethanol currently represents the largest source of alternative transportation energy [7], electric power is showing more and more promise as another viable option. One of the most significant limiting factors for battery-powered vehicles up to this point, however, has been their range.
2.3.2 Hybrid Electric Vehicles

In an attempt to alleviate this concern, the concept of the hybrid-electric vehicle (HEV) was developed to marry the utility and range of a gasoline or diesel engine with the environmentally friendly characteristics of an electric drive. Such vehicles can have a variety of configurations, but in general there are three types of powertrain arrangements: series, parallel, or series-parallel. An illustration of these three types is given in Figure 2.4.

![Illustration of Various HEV Powertrain Configurations](image)

(a) series  
(b) parallel  
(c) series-parallel

Figure 2.4: Illustration of Various HEV Powertrain Configurations: (a) Series, (b) Parallel, (c) Series-Parallel
Stated simply, series HEVs deliver most of the power for acceleration from the battery, which is then supplemented by the engine via a generator. Parallel HEVs conversely gain most of their power from the engine, which, in turn, is supplemented by the battery. In both cases, however, all power sources are eventually combined into a single output from an electric motor. The series-parallel model is somewhat more complicated than the other two more basic configurations. Instead of a specific energy source set to act as the primary, this arrangement is designed to operate at whichever ratio of battery to gasoline engine power is optimal for any given scenario. This not only takes into account the power required for a given torque or speed command, but also attempts to optimize the use of the battery through regenerative braking. This obviously involves a significant amount of complexity in terms of control decisions, but the result is a much more efficient use of energy. Still, in all these cases, the fact remains that the vehicle always must rely on fossil fuels to some extent. The goal, however, is to eliminate such a dependence altogether, so, even though limited range is still somewhat of an issue, pure electric vehicles are gaining more and more popularity.

2.3.3 Pure Electric Vehicles

In general, pure electric ground vehicles (EGVs) can be divided into two distinct families: 1) indirectly-driven type and 2) directly-driven type [4]. Indirectly-driven EGVs usually contain a single large electric motor that transfers power to the wheels via mechanical transmissions and differential gears. Commercially available EGVs are usually of this type, and are called plug-in electric vehicles (PEVs) because their batteries are charged by plugging into a wall socket or charging station. An example of a
commercially available PEV is the Tesla Model X, and Figure 2.5 shows a diagram of its undercarriage assembly.

![Figure 2.5: Diagram of Tesla Model X Undercarriage [8]](image)

It can be clearly seen that, although this model contains a front and a rear motor, each motor drives two wheels via a differential gear set and axle configuration.

Directly-driven EGVs, on the other hand, instead of relying on various mechanical power transfer devices, have an electric motor in each wheel that are all controlled independently. Vehicles of this type are typically meant for lower speed, highly controlled situations and NASA’s new Lunar Electric Rover (LER) [9] and the lithium-ion EGV at The Ohio State University Vehicle Systems and Control Laboratory (VSCL) [10] are examples of such vehicles (a more detailed description of VSCL EGV is given in Chapter 3). A diagram of a vehicle with 4 in-wheel motors is presented in Figure 2.6.
Note that the motors apply torque directly to each wheel, and they are only linked via electrical connections that converge to a central controller hub. Such a configuration results in over-actuation, and thus results in more optimization potential than an indirectly-driven design. With this enhanced capability, however, a significant amount of complexity and complicated control problems are also inevitably introduced. If a suitable control strategy can be developed, though, the capacity for savings is vast, as there are no losses due to mechanical energy transfer, as well as the capability of each wheel to be controlled independently for maximum efficiency.

Much of the control effort up to this point has been put toward the area of safety of such vehicles. Again due to the feature of over-actuation, it is not trivial to accurately drive around corners or even in a straight line. Therefore, various control strategies such as road-friction estimation [11], a system of electric differentials [12], as well as motion tracking control schemes [13] have been investigated in order to provide a comprehensive mechanism for accurately driving a directly-driven EGV. Thus the next step is to
investigate ways to optimize the energy use over a given path. Before this can be done, however, an algorithm that guarantees a globally optimal solution must be developed in order to explore various control scenarios.

2.4 Optimal Control

2.4.1 History of optimal control

The task of finding the best possible solution to certain problems has been a subject of interest since the time of ancient Greeks. It started with things such as the shortest distance between two points, a line, and “the plane curve of a given length that encloses the largest possible area”, which the Greeks knew to be a circle [14]. In 1696, however, Johann Bernoulli proposed a problem that can be said to be the birth of the study of true optimal control. The problem is as follows: “If in a vertical plane two points A and B are given, then it is required to specify the orbit AMB of the movable point M, along which it, starting from A, and under the influence of its own weight, arrives at B in the shortest possible time.” [15] A diagram of this problem is given in Figure 2.7.

![Figure 2.7: The Brachistochrone Problem [15]](image)
Bernoulli, Newton, Leibniz and L’Hospital, among others, all offered solutions to this problem, and in doing so, laid the framework for an area of mathematics called calculus of variations. The solution to this specific question, a cycloid (Figure 2.8), happens to be somewhat interesting in itself. The real contribution of this study, however, was the formalization of methods that began to consider the transfer of the state of a dynamic system with minimum cost from one point to another, known as calculus of variations [14].

![Figure 2.8: Bernoulli’s Cycloid—The Solution to the Brachistochrone Problem](image)

This basis for calculus of variations was then improved by mathematicians such as Euler, Lagrange and Legendre in the 1700s and later expanded upon by Hamilton and Weierstrass in the 1800s. Only recently, however, has optimal control been considered as an independent field of study. In the 1950s, the two fundamental ideas of true optimal control were developed somewhat independently by Pontryagin in the Soviet Union and Bellman in the United States. The results of their work, the *maximum principle*
(sometimes inversely called the \textit{minimum principle}), and dynamic programming, respectively, are outlined in the following sections.

\textbf{2.4.2 Continuous framework}

\textbf{2.4.2.1 Calculus of Variations}

In optimal control problems, the objective is to determine a function that minimizes a specified \textit{functional}—the performance measure \cite{16}. The analogous problem in calculus is to determine a point that yields the minimum value of a function. Taking a function \( f \) to be a relation that uniquely associates members of one set \( \mathcal{D} \), called the domain, with members of another set \( \mathcal{R} \), called the range, a \textit{functional} \( J \) is a rule of correspondence that assigns each function in a certain class \( \Omega \) a unique real number. In other words, a \textit{functional} is a function of a function.

Just as in calculus, to find the minimum of a function one must first find the points where the derivative is equal to zero, so too must the minimum value of a functional, called the \textit{extremal}, lie on a point where the variation of that functional is zero. Due to the linearity of functionals, the variation of a functional of \( n \) variables can be described through the definition of its increment \( \Delta J \), as follows:

\[
\Delta J \triangleq J(x + \delta x) - J(x)
\]  

where \( x \) is a function with variation \( \delta x = x - x_0 \). From here, the increment of a functional can be written as:

\[
\Delta J(x, \delta x) = \delta J(x, \delta x) + g(x, \delta x) \cdot \| \delta x \|
\]  

where \( \delta J \) is linear in \( \delta x \), and \( \| \cdot \| \) is the \textit{norm}. If the following relation is true:
\[
\lim_{\|\delta x\| \to 0} \{g(x, \delta x)\} = 0
\]

(2.7)

then \( J \) is said to be differentiable on \( x \), and \( \delta J \) is called the variation of \( J \) evaluated for the function \( x \). Using these definitions, the fundamental theorem of the calculus of variations can be formalized as follows [16]:

*If \( x^* \) is an extremal, the variation of \( J \) must vanish on \( x^* \). That is,*

\[
\delta J(x^*, \delta x) = 0 \text{ for all admissible } \delta x
\]

(2.8)

To demonstrate how this theorem is applied, consider an example where \( x \) is a scalar function in the class of functions with continuous first derivatives. It is desired to find the function \( x^* \) for which the functional

\[
J(x) = \int_{t_0}^{t_f} g(x(t), \dot{x}(t), t) \, dt
\]

(2.9)

has a relative extremum, where \( J \) is a functional of the function \( x \), \( g \) is a function that maps a number to the point \((x(t), \dot{x}(t), t)\), and \( t_0 \) and \( t_f \) are the beginning and ending time, respectively, corresponding to the end points of the curve \( x_0 \) and \( x_f \). The increment \( \Delta J \) can first be used to solve for the variation \( \delta J \), resulting in the following equation:

\[
\delta J(x, \delta x) = \left[ \frac{\partial g}{\partial x}(x, \dot{x}, t) \right]_0^f \delta x
\]

(2.10)

\[
+ \int_{t_0}^{t_f} \left\{ \frac{\partial g}{\partial \dot{x}}(x, \dot{x}, t) - \frac{d}{dt} \left[ \frac{\partial g}{\partial \dot{x}}(x, \dot{x}, t) \right] \right\} \delta x(t) \, dt
\]

Substituting \( x \) with an extremal curve \( x^* \) and applying the fundamental theorem yields:

\[
\delta J(x^*, \delta x) = 0 = \int_{t_0}^{t_f} \left[ \frac{\partial g}{\partial \dot{x}}(x^*, \dot{x}^*, t) - \frac{d}{dt} \left[ \frac{\partial g}{\partial \dot{x}}(x^*, \dot{x}^*, t) \right] \right] \delta x(t) \, dt
\]

(2.11)
Just taking the expression inside the integral, therefore, provides a necessary condition for $x^*$ to be an extremal, called the Euler equation \([16]\), given as follows:

\[
\frac{\partial g}{\partial x}(x^*(t), \dot{x}^*(t), t) - \frac{d}{dt} \left[ \frac{\partial g}{\partial \dot{x}}(x^*(t), \dot{x}^*(t), t) \right] = 0
\]  
\(2.12\)

for all $t \in [t_0, t_f]$. This can be adapted for application in any number of dimensions and also with the final boundary conditions either specified or left free by augmenting the system with Lagrange multipliers, $p_1(t), ..., p_n(t)$.

### 2.4.2.2 Necessary Conditions for Optimal Control

These techniques of calculus of variation can then be used to determine necessary conditions for optimal control. Stated explicitly, the problem is to find an admissible control $u^*$ that causes the system

\[
\dot{x}(t) = a(x(t), u(t), t)
\]  
\(2.13\)

to follow an admissible trajectory $x^*$ that minimizes the performance measure

\[
J(u) = h(x(t_f), t_f) + \int_{t_0}^{t_f} g(x(t), u(t), t)dt
\]  
\(2.14\)

also known as the cost function. When doing this, it is convenient to use the function $\mathcal{H}$, called the Hamiltonian \([16]\), defined as:

\[
\mathcal{H}(x(t), u(t), p(t), t) \triangleq g(x(t), u(t), t) + p^T(t)[a(x(t), u(t), t)]
\]  
\(2.15\)

where $p^T(t)$ is the transpose of the vector of Lagrange multipliers. Using this notation, the necessary conditions for optimality are therefore as follows:
resulting in the following relation:

\[
\dot{x}^*(t) = \frac{\partial \mathcal{H}}{\partial p}(x^*(t), u^*(t), p^*(t), t) \tag{2.16}
\]

\[
\dot{p}^*(t) = -\frac{\partial \mathcal{H}}{\partial x}(x^*(t), u^*(t), p^*(t), t) \tag{2.17}
\]

\[
0 = \frac{\partial \mathcal{H}}{\partial u}(x^*(t), u^*(t), p^*(t), t) \tag{2.18}
\]

resulting in the following relation:

\[
\left[\frac{\partial h}{\partial x}(t_f) - p^*(t_f)\right]^T \delta x_f + \left[\mathcal{H}(t_f) + \frac{\partial h}{\partial t}(t_f)\right] \delta t_f = 0 \tag{2.19}
\]

where, for simplicity, \(t_f\) is used to denote all arguments of a given function evaluated at the final time.

2.4.2.3 Variational Approach with Constraints

These necessary conditions, however, assume no bounds on the control or the states. In order to represent realistically actuated systems where the control inputs are bounded, however, these necessary conditions must be modified. This can be done simply by replacing Equation (2.18) with the following:

\[
\mathcal{H}(x^*(t), u^*(t), p^*(t), t) \leq \mathcal{H}(x^*(t), u(t), p^*(t), t) \tag{2.20}
\]

for all admissible \(u(t)\), and for all \(t \in [t_0, t_f]\). Of note is that \(u^*(t)\) is a control that causes \(\mathcal{H}(x^*(t), u(t), p^*(t), t)\) to assume it global, or absolute, minimum. This generalization of the fundamental theorem when considering bounded control and states, therefore, is known as Pontryagin’s minimum principle.
2.4.3  **Discrete framework**

The variational approach taken by Pontryagin, as discussed above, provides an elegant, robust method for formulating optimal control problems as nonlinear two-point boundary-value problems. This approach, however, does not lend itself gracefully to application with digital computers. Instead, a method developed by Richard Bellman [17], termed dynamic programming, can be used for both formulation and solution of optimal control problems in a discrete architecture.

2.4.3.1  **The Principle of Optimality**

Just as before, the goal is to determine the optimal control of the form

\[ u^*(t) = f(x(t), t) \]  \hspace{1cm} (2.21)

where \( f \) is a functional relationship called the optimal control law, or the optimal policy. In dynamic programming, this optimal policy is found by employing a concept called the principle of optimality. Figure 2.9 illustrates the choice of an optimal path from starting at \( a \) and ending at \( e \).

![Figure 2.9: (a) Optimal Path from \( a \) to \( e \), (b) Two Possible Optimal Paths from \( b \) to \( e \) [16]](image)

\( J_{ab} \) and \( J_{be} \) represent the cost associated with segments \( a-b \) and \( b-e \), respectively, such that the minimum cost \( J_{ae}^* \) from \( a \) to \( e \) is therefore:
The real consequence of the principle of optimality, however comes when considering another possible path, \( b-c-e \), with an associated cost of \( J_{bce} \). What can be shown through contradiction is the fact that, if \( a-b-e \) is, in fact, the optimal path from \( a \) to \( e \), then \( b-e \) is the optimal path from \( b \) to \( e \), not \( b-c-e \). Bellman summarized this result as follows [18]:

An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

2.4.3.2 Dynamic Programming

Dynamic programming is a computational technique which extends the above decision-making concept to sequences of decisions, which together define an optimal policy and trajectory [16]. Mathematically, the general dynamic programming algorithm can be represented by the two following equations [16]:

\[
J_{ae}^* = J_{ab} + J_{be}
\]  
(2.22)

\[
C_{ax_ih}^* = J_{ax_i} + J_{x_ih}
\]  
(2.23)

\[
J_{ah}^* = \min \{ C_{ax_1h}^*, C_{ax_2h}^*, \ldots, C_{ax_ih}^* \}
\]  
(2.24)

where,

- \( \alpha \) is the current state
- \( u_i \) is an allowable decision (control) elected at the state \( \alpha \)
- \( x_i \) is the state adjacent to \( \alpha \) which is replaced by application of \( u_i \) at \( \alpha \)
- \( h \) is the final state
- \( J_{ax_i} \) is the cost to move from \( \alpha \) to \( x_i \)
- \( J_{x_ih}^* \) is the minimum cost to reach the final state \( h \) from \( x_i \)
- \( C_{ax_ih}^* \) is the minimum cost to go from \( \alpha \) to \( h \) via \( x_i \)
- \( J_{ah}^* \) is the minimum cost to go from \( \alpha \) to \( h \) (by any allowable path)
- \( u^*(\alpha) \) is the optimal decision (control) at \( \alpha \)
From this, it can be said that the optimal decision at \( \alpha, u^*(\alpha) \), is the decision that leads to Equation (2.24). Note that, in order to calculate the minimum cost at the current state \( C_{\alpha x t h}^* \), the minimum cost to reach the final state from the current state, \( J_{x t h}^* \), must be known. In order to successfully compute the optimal trajectories, therefore, the cost must be calculated first from the final state \( h \), moving backwards toward the initial state. Also of note is the fact that this method results in a list (or table) of the optimal trajectories starting from all points. Thus, in order to get the optimal trajectory starting from the initial state and ending at the final state, one must simply start at the initial state and track the series of optimal trajectories until the final state is reached.

An illustration of utilizing this algorithm to calculate an optimal path through a graph is shown in Figure 2.10.

![Figure 2.10: Shortest Path Example](image)

The path of minimum total cost is shown in blue as the path A-C-E-D-F, with cost 22. Note that if the optimal trajectory was attempted to be reached by starting at node A and choosing the path of least cost at each intersection, the result would be the path A-B-C-E-
D-F, with a cost of 25. This type of process is called a greedy algorithm, and it very rarely results in the truly optimal solution.

2.4.3.3 Computational Procedure

As should be evident by the structure of the problems discussed thus far, dynamic programming natively assumes a system with discretized states. Therefore, in order to apply the numerical procedure presented above to an inherently continuous system (e.g. one that varies continuously with time), the system must first be discretized in terms of finite differences. This means system differential equations become difference equations and integrals in a performance measure are approximated by summations. Most typically, this is done in time-varying systems by dividing the time interval \( t_0 \leq t \leq t_f \) in \( N \) equal increments of size \( \Delta t \). To determine the optimal control applied at \( t = k\Delta t \) in an \( N \)-stage process, therefore, Equations (5.1) and (2.24) become:

\[
C_{k,N}^*(x(k), u(k)) = J_{k,k+1}(x(k), u(k)) + J_{k+1,N}^*(x(k+1)) \quad (2.25)
\]

\[
J_{k,N}^*(x(k)) = \min_{u(k)}[C_{k,N}^*(x(k), u(k))] \quad (2.26)
\]

which, taken together, form the \textit{functional equation of dynamic programming} [16].

Starting with the definition of an optimal control problem, given in Equations (2.13) and (2.14), the state equation can now be denoted as the following:

\[
x(k + 1) = a_D(x(k), u(k)) \quad (2.27)
\]

where \( a_D(k) = x(k) + \Delta t \ a(k) \). In a similar fashion, the performance measure can be denoted as follows:
\begin{equation}
J = h(x(N)) + \sum_{k=0}^{N-1} g_D(x(k), u(k))
\end{equation}

where \( g_D(k) = \Delta t \, g(k) \). For a \( K \)-stage process, therefore, one can obtain the recurrence relation that generates the optimal cost, given here:

\[
J^*_K(x) = \min_{u(N-K)} \{g_D(x, u) + J^*_{K-1}(a_{d}(x, u))\}
\]

for \( K = 1, 2, \ldots, N \). The solution to this recurrence equation is an optimal control law or optimal policy, \( u^* \), which is obtained by trying all admissible control values at each admissible state value. A sample computational procedure that achieves this solution is given by the flow chart in Figure 2.11.
While this method always results in the global optimal solution, it also requires a significant amount of storage and computing power for large, multi-dimensional systems. It does outperform a naïve, exhaustive algorithm due to the fact that dynamic programming only considers those controls that satisfy the principle of optimality instead of all admissible controls that cause admissible trajectories. This results in the number of calculations merely increasing linearly with the number of stages instead of
exponentially. Once the number or states and orders of the system get somewhat large, however, the number of storage locations and computations, while not outrageous like during direct enumeration, still results in large computation times. Bellman calls this the “curse of dimensionality”, and finding methods to speed up this process is a topic of ongoing research [19].
CHAPTER 3: MODEL OF THE EGV SYSTEM

3.1 Introduction

In order implement a digital optimal solution algorithm such as the one discussed in Section 2.4.3, first the physical system has to be represented as a theoretical model that accurately recreates the dynamics. To do this, first a free body diagram is drawn to represent the forces present in the vehicular system, and from that, the equations of motion can be derived. The following sections outline this procedure as well as introduces how the system was discretized for implementation in a digital simulation.

3.2 Physical Parameters of the EGV

The vehicle considered in this research, seen in Figure 3.1, is a pure electric ground vehicle, which is equipped with four independently actuated in-wheel motors.

![Figure 3.1: Prototyping EGV at the VSCL](image-url)
The following sections outline the electrical and mechanical components along with a discussion on how efficiency of the electric motors is represented.

3.2.1 Electrical Components

Each wheel is independently actuated by a customized 7.5 kW in-wheel motor which is mounted in the wheel, as shown in Figure 3.2. As discussed in Section 2.2, the motors are permanent-magnet brushless direct-current (BLDC) electric motors.

![Figure 3.2: In-wheel motors before and after installation](image)

The single source of power is a 72 V lithium-ion power battery pack, comprised of 22 battery cells connected in series, as shown in Figure 3.3.

![Figure 3.3: Li-ion Battery Pack installed in the EGV](image)
The battery pack has a nominal capacity of 200 Ah and is capable of delivering up to 40 kW peak power. Each battery cell is also equipped with a battery management system (BMS) chip which can not only monitor various battery states such as voltage, state of charge (SOC), and state of health (SOH), but also automatically balance the battery voltages. Table 3.1 presents a summary of all the electrical components contained within the EGV [10].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In-Wheel Motors</strong></td>
<td></td>
</tr>
<tr>
<td>Motor mass</td>
<td>25 kg</td>
</tr>
<tr>
<td>Combined inertia of motor, wheel, and tire</td>
<td>2.31 kg·m²</td>
</tr>
<tr>
<td>Maximal motor power</td>
<td>7.5 kW</td>
</tr>
<tr>
<td>Maximal motor torque</td>
<td>150 Nm</td>
</tr>
<tr>
<td>Maximal motor rotational speed</td>
<td>900 RPM</td>
</tr>
<tr>
<td><strong>Battery Pack</strong></td>
<td></td>
</tr>
<tr>
<td>Number of battery cells in series</td>
<td>22</td>
</tr>
<tr>
<td>Nominal voltage of one battery cell</td>
<td>3.3 V</td>
</tr>
<tr>
<td>Total battery pack voltage</td>
<td>72 V</td>
</tr>
<tr>
<td>Nominal capacity</td>
<td>200 Ah</td>
</tr>
<tr>
<td>Maximal current</td>
<td>800 A</td>
</tr>
<tr>
<td>Battery cell mass</td>
<td>3.6 kg</td>
</tr>
<tr>
<td>Battery cell dimensions</td>
<td>0.36m × 0.065m × 0.28m</td>
</tr>
</tbody>
</table>

Table 3.1: Parameters of the EGV’s Electrical Components

### 3.2.2 Characterization of Motor Efficiency

In addition to these nominal characteristics of the EGV’s components, previous work has also been performed to quantify the efficiency maps of the in-wheel motors. Efficiency of a BLDC motor is a function of motor speed and torque, and this relationship changes depending on whether the motor is generating power (driving mode), or recuperating power (regenerative braking mode). In order to account for this change when solving for the optimal control trajectory, the effect of both torque and
wheel speed on efficiency was experimentally documented [10], and the results are shown in Figure 3.4.

![Efficiency Maps for Driving and Regenerative Braking of an In-Wheel Motor](image)

Figure 3.4: Efficiency Maps for Driving and Regenerative Braking of an In-Wheel Motor

These maps are used as inputs to the optimal control algorithm discussed in Chapter 4, where the goal is to keep the vehicle operating in the most efficient regions as much as possible, when given a certain terrain profile. In addition to these efficiency maps, an extra factor of 0.8 is applied if the motor is located in the rear of the EGV instead of at the front. This is done simply to simulate the effect of possible variation between the front and rear configurations. The resulting consequence is that four representations for overall efficiency are possible, as shown in Table 3.2.

<table>
<thead>
<tr>
<th></th>
<th>Front</th>
<th>Rear</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driving</td>
<td>$\eta_{\text{drive}}(T, \omega)$</td>
<td>$0.8\eta_{\text{drive}}(T, \omega)$</td>
</tr>
<tr>
<td>Regenerative Braking</td>
<td>$\eta_{\text{brake}}(T, \omega)$</td>
<td>$0.8\eta_{\text{brake}}(T, \omega)$</td>
</tr>
</tbody>
</table>

Table 3.2: Four Cases for Characterizing Overall Motor Efficiency
where $\eta(T, \omega)$ represents the efficiency of the motor bi-linearly interpolated at the specified torque and angular speed values.

### 3.2.3 Mechanical Components

Furthermore, in addition to the parameters of the electrical systems of the EGV, the mechanical characteristics are needed for analysis of the external forces acting on the vehicle. The important physical characteristics of the EGV, therefore, are given in Table 3.3.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M$</td>
<td>Total EGV mass</td>
<td>800 kg</td>
</tr>
<tr>
<td>$A$</td>
<td>Frontal area</td>
<td>1.66 m$^2$</td>
</tr>
<tr>
<td>$L$</td>
<td>Wheelbase</td>
<td>1.89 m</td>
</tr>
<tr>
<td>$b$</td>
<td>Height of center of gravity</td>
<td>0.6 m</td>
</tr>
<tr>
<td>$R_{eff}$</td>
<td>Tire effective radius</td>
<td>0.33 m</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Tire-road coefficient of static friction</td>
<td>0.8</td>
</tr>
<tr>
<td>$C_a$</td>
<td>Air drag coefficient</td>
<td>0.37</td>
</tr>
</tbody>
</table>

Table 3.3: Physical Characteristics of the EGV

### 3.3 Derivation of the Equations of Motion

With the various parameters of the EGV known, the system can now be modeled with a free-body diagram (FBD). The FBD of the motors themselves can be found in Section 2.2. From this, the torque on the wheels can be understood as a function of the input voltage and current, which can then be applied to the EGV as a whole. Since the goal is to maximize efficiency over a given terrain, the effect of terrain can be simulated by placing the EGV on a slope and analyzing the forces present. Also of note is that only longitudinal motion is considered, meaning the two motors at both the front and the rear
can be grouped into single points, denoted 1 (front) and 2 (rear). Figure 3.5 shows this simplified model of an EGV.

![Figure 3.5: FBD of the EGV on a Slope](image)

From the figure, $\theta$ is the slope angle in the driving direction, $v$ is the vehicle longitudinal velocity, $b$ is the center of gravity height, $L$ is the distance of the wheelbase, $M$ is the total mass, $g$ is the acceleration due to gravity, $F_{x1}$ and $F_{x2}$ are the forces on each tire due to the driving torque, and $F_{n1}$ and $F_{n2}$ are the normal forces on the wheels. The driving force, $F_t$, can thus be represented by the following equation:

$$F_t = 2(F_{x1} + F_{x2}) = F_a + F_G + F_r + F_w$$  \( (3.1) \)

The various forces listed in Equation (3.1) that make up the necessary overall driving force are defined as follows:

$$F_a = M\dot{v}$$  \( (3.2) \)

$$F_G = Mg \sin \theta$$  \( (3.3) \)

$$F_r = C_r M g \cos \theta$$  \( (3.4) \)

$$F_w = \frac{1}{2} \rho A C_a v^2$$  \( (3.5) \)
where \( F_a \) is the accelerating resistance, \( F_g \) is the force due to gravity component along the slope, \( F_r \) represents the rolling resistance, \( F_w \) represents the aerodynamic resistance, \( C_r \) and \( C_a \) are the rolling and aerodynamic resistance coefficients, respectively, \( \rho \) represents the air density, and \( A \) is the frontal area of the EGV. From here, the power consumption of the EGV can be described as follows:

\[
P_{EGV} = F_t v = \sum_{j=1}^{2} 2\eta(j)P_{motor}(j)
\]

where \( P_{motor} = T\omega \) represents the power of a single in-wheel electric motor, and \( \eta \) represents the efficiency calculated as shown in Table 3.2.

To find the maximum allowable torque to avoid wheel slip, the normal forces on the front and rear wheels also need to be defined, as given in Equations (3.7) and (3.8).

\[
F_{n1} = \frac{Mg}{L} \left( \frac{L}{2} \cos \theta - b \sin \theta \right) - M \frac{b}{L} \dot{v}
\]

\[
F_{n2} = \frac{Mg}{L} \left( \frac{L}{2} \cos \theta + b \sin \theta \right) + M \frac{b}{L} \dot{v}
\]

From these relations, the torque constraints due to friction are given as follows:

\[
T_{1,max} = F_{x1,max}R_{eff} = \mu F_{n1}R_{eff}
\]

\[
T_{2,max} = F_{x2,max}R_{eff} = \mu F_{n2}R_{eff}
\]

where \( R_{eff} = v/\omega \) is the effective radius of the tires, and \( \mu \) is the coefficient of static friction between the tires and the road.
3.4 Discretization of the Dynamic Model for Digital Implementation

Before the model derived above can be implemented into a dynamic programming algorithm like the one outlined in the next section, it first needs to be represented in terms of change of state instead of just as static relationships. Also, since dynamic programming is the method used to find optimality, this needs to be done in a discrete format rather than a continuous one.

Since one of the most common applications for utilizing optimal control on a dynamic system is for the minimization of time, a system is typically discretized with respect to time. In this particular case, however, discretizing with respect to time will result in a different number of nodes for each trial. This is discussed in more detail in Chapter 4, but essentially this is due to the fact that varying the vehicle velocity is the chief method of minimizing the overall energy consumption. Since velocity is an independent variable of the system, different cases will result in different velocity profiles, and therefore different total trip times. Thus, in an effort to keep the number of discrete states consistent for a given terrain profile, the system was discretized, instead, with respect to position. Note that the position referenced here is defined with respect to a coordinate system attached to the ground. Therefore, since the slope of the terrain can vary infinitely between 0-90° uphill or downhill, the total distance traveled, as seen by the EGV, may be greater than the recorded distance.

The specific terrain profile used in this research is given in Figure 3.6 (a graph of its altitude profile is given in Figure 4.2).
This path is a loop that connects Cambria Way, Turtle Station Way, and Blue Jacket Road and is located near downtown Columbus. The total distance is about 3000m, and it was chosen because it seemed to provide enough altitude variation, while staying within the limits of typical city road characteristics.

Furthermore, for simplicity’s sake, constant acceleration was assumed between each node. This results is three possible velocity profiles between nodes, as shown in Figure 3.7.

![Figure 3.7: Possible Speed Profiles between Discrete Nodes](image)
From this, since change in position is chosen as an input, the relationship between distance, velocity, and time between each node can be derived as follows:

\[ \Delta s = \frac{[v(k + 1) + v(k)]}{2} \Delta t(k) \] (3.11)

\[ v(k + 1) = v(k) + \dot{v}(k) \Delta t \] (3.12)

where \( \Delta s \) is the distance between each node, \( k \) is the index of the current node, and \( \Delta t \) is the change in time between \( k \) and \( k + 1 \).

With this discrete structure, the amount of battery drain between consecutive nodes can be derived. After applying the no-slip constraint given in Equations (3.7) and (3.8) to the definition of power consumption found in Equation (3.6), the change in state of charge of the battery pack in a discrete format can thus be described as follows:

\[ SOC(k + 1) = SOC(k) - \frac{P_{EGV}(k)}{E_{bat,max}} \Delta t \] (3.13)

where \( SOC(k) \) is the current state of charge, \( P_{EGV}(k) \) is the total power consumption, and \( E_{bat,max} \) is the maximum charge of the battery. Note that battery power is positive during discharging and negative during charging, and also that the thermal effects and transient dynamics of the batteries are disregarded in this model. In the next chapter, the dynamic programming algorithm that minimizes the loss in battery charge is outlined in detail and discussed.
CHAPTER 4: OPTIMAL CONTROL ALGORITHM

4.1 Overall Structure

The algorithm employed to find the optimal solution to the energy minimization problem utilizes many of the techniques outlined in Section 2.4.3, and the resulting structure is very similar to the one given in Figure 2.11. Some alterations and adaptations, however, had to be made, and this chapter presents the actual algorithm used in full detail. To start, Figure 4.1 gives the top level of the program, showing four distinct parts.

Figure 4.1: Top Level of Optimal Control Algorithm
The overall goal of the algorithm is to minimize the energy consumption by optimizing two different variables within the EGV. The first is velocity, with the general concept that it is easier to move faster on a downward slope, and, conversely, more difficult to move uphill due to the effect of gravity. The second is distribution of torque to the front and rear wheels. The idea here is that, again depending on the slope, it may be more efficient to send the power to one side or the other. The DP algorithm takes into account the effect of the entire terrain profile as well as all the characteristics of the EGV and determines the best combination of speed and torque distribution.

To do this, looking at Figure 4.1, the user first must input the desired boundary conditions and parameters of the traffic constraints. These inputs are then used to automatically generate the various physical parameters of the system in the initialization phase. This includes extracting the terrain information from the Geographic Information System (GIS) and Global Positioning System (GPS) data, as seen in Figure 4.2.

![Figure 4.2: Terrain Profile Used in Simulations](image)

This step also includes importing the parameters of the EGV given in Table 3.1 and Table 3.3, and defining the states of the traffic models for the entire domain (more information about the traffic models is given in Section 4.3). Next, this initialized framework is input into the dynamic programming subprogram, where the optimal trajectories at each node
are calculated and stored in a table. Once this is done for all nodes, starting with the last and moving to the first, the result is output to a final stage for post-processing. Actions performed during this final stage include compiling results, plotting, and comparison with other cases. Since the most important part of this algorithm is the dynamic programming loop, it is discussed in detail in the next section.

**4.2 Dynamic Programming Loop**

**4.2.1 Physical System Constraints**

Before the optimal trajectories can be calculated, the constraints on the system need to be enforced by the algorithm. This not only greatly reduces the computation time due to a much smaller domain that must be checked, but also increases the realism of the model. For instance, the calculated input torques cannot exceed the capabilities of the physical motors, or the speed range of the EGV should not be very close to zero, as real people do not drive that slowly. A complete list of the various physical constraints enforced on the system are given in Table 4.1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motor Torque</td>
<td>-80 Nm</td>
<td>100 Nm</td>
</tr>
<tr>
<td>Battery SOC</td>
<td>0.5</td>
<td>0.8</td>
</tr>
<tr>
<td>Speed of the EGV</td>
<td>25 km/h</td>
<td>40 km/h</td>
</tr>
<tr>
<td>Acceleration of the EGV</td>
<td>-</td>
<td>0.981 m/s²</td>
</tr>
</tbody>
</table>

Table 4.1: Vehicle Constraints

The limits on the battery SOC were chosen as such due to the fact that charging or discharging outside that range results in a greatly reduced life of the battery. The actual maximum speed of the EGV is around 60 km/h, but this range (~15-25 mph) was chosen to more accurately simulate speed ranges within a congested city area. The next sections
look at how these constraints are applied to the calculation of the optimal speed and torque profiles.

4.2.2 General Structure

The location of the dynamic programming portion within the overall structure of the program can easily be seen in Figure 4.1. For all cases, the total distance was divided into 100 equal sections, meaning the total number of nodes, \( N = 101 \). Since, as discussed in Section 3.4, the system was discretized with respect to distance, this resulted in a distance of 30m between each node. Then, to start the DP, the current state \( k \) is set to 100 because all calculations use the notation \( k + 1 \) to represent the next step. With this information, the first step of this subprogram is to calculate the slope between the current node, \( k \), and the next node, \( k + 1 \), as follows:

\[
\theta = \tan^{-1}\left(\frac{\Delta h}{\Delta s}\right)
\]  

(4.1)

where \( \Delta h = h(k + 1) - h(k) \), \( \Delta s = 30m \), and \( \theta \) is the slope of the terrain, as shown in the FBD in Figure 3.5.

Next, the program tries all admissible current and next speeds and stores the results of the combination that results in the least energy drain. A block diagram of how this is accomplished is shown in Figure 4.3.
First, both the current and next speeds are set the minimum allowable value. From here, the state parameters resulting from that velocity profile are calculated as follows:

\[
    v_{avg} = \frac{v_k + v_{k+1}}{2} \quad (4.2)
\]

\[
    \Delta t = \frac{\Delta s}{v_{avg}} \quad (4.3)
\]

\[
    a = \frac{v_{k+1} - v_k}{\Delta t} \quad (4.4)
\]

where \(v_{avg}\) is the average speed of the EGV, \(\Delta t\) is the time elapsed, and \(a\) is the acceleration of the EGV, all between node \(k\) and \(k + 1\). Next, if specified, the preceding
vehicle constraint is applied, and finally the energy consumption based on the current calculated state is determined and stored. Once all possible combinations of \( v(k) \) and \( v(k+1) \) are tried, the combination resulting in the minimum energy consumption is selected and the algorithm moves on to the next node.

The two most important blocks in the diagram in Figure 4.3 are the preceding vehicle constraint, as well as the block where the states change with minimum energy is calculated. The traffic constraint is detailed in Section 4.3.2, while the minimum energy consumption calculation is shown in the following section.

4.2.3 Calculate Minimal Energy Consumption from State Change

This is the step where all the equations of motion derived in Section 3.3 are evaluated at the current node, and the constraints from Section 4.2.1 are enforced. If at any point, any one of the calculated states exceeds the domain of admissible values, the result for that iteration is set to ‘NaN’, and the program moves to the next step. As such, first the acceleration calculated in Equation (4.4) is checked. Next the angular speed of the motor is calculated as follows:

\[
\omega = \frac{v_{avg}}{R_{eff}}
\]  

(4.5)

From here, all the forces are calculated using Equations (3.1)-(3.5). With this information, the minimum required motor torque and power are calculated for both the front and rear wheels. Next, all the torque values that are greater than the minimum required torque and lie within the constraints shown in Table 4.1 are stored. Then the power associated with each of these possible torques is calculated using the efficiency maps from Figure 3.4 and
the efficiency relationships given in Table 3.2. Once the power due to each torque is known, the resulting change in SOC is calculated using the following equation:

\[
SOC(k + 1) = SOC(k) - \frac{P(k)}{E_{max, bat}}
\]  

(4.6)

where \(P(k)\) is the power consumption at the current step, \(E_{max, bat}\) is the total energy capacity of the battery, and \(SOC(k + 1) - SOC(k) = \Delta SOC\). The final step, therefore, is to simply choose the torque value that corresponds to the minimum \(\Delta SOC\). Once this is done for all possible choices of \(v(k)\) and \(v(k + 1)\), the program moves to the next step, \(k = k - 1\), until it reaches the start of the path, where \(k = 1\).

4.3 Traffic Constraints

4.3.1 Introduction

The algorithm outlined above can be run with or without any traffic constraints. The first simulations were run assuming an open road, and the results are given in Section 5.2. Since traffic plays a significant role in ground transportation, however, adding some way to simulate its effect, even in a primitive manner, can provide significant insight into any energy optimization strategy. In order to simulate various traffic conditions, however, some interesting hurdles need to be overcome. First of all, traffic in general is very complicated, as it involves highly dynamic interactions of many different elements. While significant advancements in vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications have made tracking these interactions much more feasible [20], another problem exists for application in this research. That is, specifically for a DP algorithm, all constraints need to be known at the beginning of the routine, meaning only
simulated or simplified models can be used. With this in mind, the following sections show how two simplified traffic conditions, a preceding vehicle and a traffic light, can be implemented as constraints on the dynamic programming algorithm.

4.3.2 Preceding Vehicle

One of the most obvious instances of traffic is the fact that other vehicles are around that must be avoided. Since the driving model developed for this simulation only takes into account longitudinal motion, the logical step is to simulate the effect of other vehicles by placing one in front of the EGV that cannot be passed (a vehicle behind will not affect the EGV’s trajectory and any number of vehicles in front if the EGV can always be condensed to the effect of just one). Mathematically, this condition can be expressed, in discrete format, as follows:

\[ x_{pre}(k - 1) + \sum_{t(k-1)}^{t(k)} v_{pre} \Delta t \geq x_{EGV}(k - 1) + \sum_{t(k-1)}^{t(k)} v_{EGV} \Delta t \] (4.7)

where \( x_{pre} \) and \( x_{EGV} \) are the longitudinal locations of the preceding vehicle and EGV, respectively, \( t(k) \) is the time to reach the node \( k \) and \( v_{pre} \) and \( v_{EGV} \) are the speeds of the preceding vehicle and EGV, respectively. In essence, this inequality says that the EGV cannot reach a given node before the preceding vehicle. For the purposes of this research, the entire velocity profile of the preceding vehicle is known. Therefore, this inequality can be satisfied at all nodes \( k = 1, ..., N \).

The problem with Equation (4.7), however, is that it relies on comparing the time at which the EGV and the preceding vehicle reach each node. As stated above, this is considered known for the preceding vehicle, but the EGV is different. The change in time
can be calculated at each step, but since the EGV velocity is one of the variables of the optimization algorithm, the total time from the beginning of the trip cannot be known until the first node is reached in the optimization process. Since the constraint described in Equation (4.7) must be applied moving “forward”, the following method can be used to enforce the constraint while maintaining optimality.

First, the program is run assuming no preceding vehicle constraint, thus generating a complete velocity profile for the EGV. Then the total time from the first state to the next state is checked for each node using the following equations:

\[
t_{pre}(k + 1) = \sum_{i=1}^{k+1} \frac{2\Delta s}{v_{pre}(i) + v_{pre}(i + 1)}
\]

\[
t_{EGV}(k + 1) = \frac{2\Delta s}{v_{EGV}(k) + v_{EGV}(k + 1)} + \sum_{i=1}^{k} t_{EGV}(i)
\]

If the EGV reaches the next node before the preceding vehicle, the current speed choice is rejected and replaced by the next lowest possible value. From here, the time checked again and the process is repeated until a speed that is low enough to avoid a collision is found. Once this is performed for each node in the system, the result is an optimal trajectory for the EGV that stays behind the preceding vehicle.

4.3.3 Traffic Lights

Traffic lights, in principle, are simple devices. In their most basic form, they can be easily represented by a pattern of red and green that simple repeats. With advances in V2I technology, however, they are becoming more and more complex by adapting to the current state of traffic [20], [21]. Since, again, implementation into a DP algorithm
requires complete \textit{a priori} knowledge of the signal timings, however, such complexity will be disregarded. For the purposes of this research, a traffic signal can be represented by the following matrix:

\[
\begin{bmatrix}
  g_{i,j} & r_{i,j} & g_{i+1,j} & r_{i+1,j} & \cdots \\
g_{i,j+1} & r_{i,j+1} & g_{i+1,j+1} & r_{i+1,j+1} & \cdots \\
  \vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix}
\] (4.10)

where each row represents the sequence of when the light will turn green and red for all time (yellow lights can be lumped with either the green or red portion of the sequence). The total number of rows represents the total number of lights within the chosen path. A typical light sequence lasts 100s, where the green light lasts 40s and the red lasts 60s \cite{21}. The sequence can therefore be represented as follows:

\[
[g_i, r_i, g_{i+1}, r_{i+1}, \cdots] = [0, 40, 100, 140, \cdots]
\] (4.11)

Similar to the problem with incorporating the preceding vehicle constraint, a traffic light constraint also relies on comparing the time at which the EGV reaches the node that contains the light. Therefore, in order to enforce this constraint, a similar algorithm is followed. First, a DP solution is found assuming no traffic constraints. Then the time at which the EGV reaches the node with the traffic light is calculated using Equation (4.9). If the light is green, the EGV is allowed to keep its current speed trajectory and move forward. If the light is red, however, the program is run again, ranging from the beginning of the path to the current node. This time, however, minimum and maximum velocity constraints are enforced that ensure the EGV will pass through the light when it is green. Results of various placements and timings of the traffic lights are presented in Section 5.3.2.
CHAPTER 5: SIMULATION RESULTS

5.1 Baseline Cases

In an effort to provide a reference with which the various optimal solutions could be compared against, various naïve actuation cases were simulated. Since the DP algorithm optimizes the EGV’s actuation with respect to both speed and torque distribution, observing the effect of holding each constant can be beneficial. The methods for this are outlined and their results tabulated below.

5.1.1 Constant Speed

The first baseline case considered is if the EGV speed is held constant for the entire path. Within this constraint, however, the motor torque distribution is still optimized. This is achieved in the program described in Chapter 4 by setting the minimum and maximum velocity constraints both equal to the desired speed. Figure 5.1 shows the resulting ending battery SOC for velocities ranging from 1-50 km/h.

Figure 5.1: Ending SOC for Constant Speed Case (1-50 km/h)
For velocities near 1 km/h, the performance of the EGV in terms of energy use is extremely bad. Once the speed is set to more reasonable values of 10 km/h and higher, however, the ending SOC ranges from about 67-69%, with the maximum value occurring at 26 km/h. Since, as discussed in Section 4.2.1, the velocity range chosen for this research is 25-40 km/h, a plot showing just that range is given in Figure 5.2.

![Figure 5.2: Ending SOC for Constant Speed Case (25-40 km/h)](image)

From this, it is clear that, for this particular terrain profile, the further the constant speed gets away from the maximum at 26 km/h, the lower the ending SOC. This data will be used to compute the percent increase in energy efficiency for the optimal cases using Equation (5.1), as follows:

\[
% \text{Increase in Efficiency} = \frac{SOC_{opt}(N) - SOC_{BL}(N)}{\Delta SOC_{BL}} \times 100\% \quad (5.1)
\]

where, if \( N \) represents the last node in the system, \( \Delta SOC = SOC(1) - SOC(N) \) and the subscripts \( opt \) and \( BL \) correspond the optimal solution and the chosen baseline case, respectively.
5.1.2 Equal Actuation

As noted above, the constant velocity cases shown in Figure 5.2 still involve optimization of the distribution of motor torques between the front and rear. The next baseline case, therefore, is to examine the effect of assuming equal actuation in both the front and rear motors. This is achieved by first running the simulation with torque optimization, and then simply setting the front and rear torques to the average of the two at each node. After this, the new torque profile can be input into a forward simulation of the EGV over the given terrain, resulting in a new velocity profile as well as SOC at each node. An example case, where the EGV speed was held to 30 km/h, is shown in Figure 5.3 to demonstrate the difference in torque distribution. Note that both cases represent a condition where the EGV’s speed was held constant over the entire terrain. In order to differentiate between the two, the case with optimal torque distribution is called “Constant Velocity” (CV), while the constant velocity case with averaged motor torques is called “Equal Actuation” (EA).
Figure 5.3: Comparison of Torques between CV and EA Cases

Note that the EA torques, first of all, lie on top of each other at all points, and also that they lie exactly halfway between the CV front and rear torques. The plot of altitude vs. distance is also shown to observe how the torques change with terrain.

This was then done for all constant velocity cases between 25 and 40 km/h, the results which are given below in Figure 5.4.

Figure 5.4: Comparison of Ending SOC between CV and EA Cases
From the top plot, it is clear that the EA case is well below the case with optimized torque distribution. The percent increase in efficiency is shown in the lower plot, where it seems the CV case results in around 17-18% energy gain over the EA by the end of the path.

5.2 Optimal Trajectory with an Open Road

Before observing the effect of traffic, first the result without any external constraints needs to be documented. This will be considered a best case scenario when compared against the various traffic scenarios presented in Section 5.3. As stated previously, the speed range of the EGV for all subsequent cases is 25-40 km/h, and the open road results are shown in Figure 5.5.
The first axis shows SOC of four different scenarios: the dynamic programming optimal solution (DP), the constant velocity case that contains optimized torques (CV), as well as equal actuation cases for both the DP and CV. For the constant speed cases, 31 km/h was chosen because that value corresponds with the average velocity during the DP case, as can be seen in the second axis. The third axis shows the torque distribution of both the CV and DP cases, while the bottom axis simply shows the terrain profile for reference.

In order to quantify the performance of the best case scenario, a comparison of the various ending SOCs for each of the cases given in Figure 5.5 is shown in Table 5.1.

<table>
<thead>
<tr>
<th></th>
<th>EA (opt.)</th>
<th>CV</th>
<th>DP</th>
</tr>
</thead>
<tbody>
<tr>
<td>EA (const.)</td>
<td>18.18%</td>
<td>24.84%</td>
<td>40.61%</td>
</tr>
<tr>
<td>EA (opt.)</td>
<td></td>
<td>8.15%</td>
<td>27.41%</td>
</tr>
<tr>
<td>CV</td>
<td></td>
<td></td>
<td>20.97%</td>
</tr>
</tbody>
</table>

Table 5.1: Comparison of Open Road DP Solution to Baseline SOCs

Since the constant velocity equal actuation case had the lowest ending SOC, it was chosen as the base reference for calculating the percent energy gain. Looking at the DP column, it is evident that the true optimal solution offers significant savings over any of the other scenarios, with over 40% improvement over the worst case scenario. These results will be the basis for evaluating the effect of the various traffic models, as presented in the next section.

5.3 Effect of Traffic Constraints on the Optimal Solution

5.3.1 Preceding Vehicle

The first traffic element examined is the preceding vehicle. The method for its implementation is given in Section 4.3.2, and the first cases all involve a constant
preceding vehicle speed. Figure 5.6 shows a compilation of what happens to the EGV velocity profile as the preceding vehicle speed is increased.

Figure 5.6: Effect of Preceding Vehicle Speed on EGV Speed Profile
From the figure, it is clear that the EGV always successfully stays behinds the preceding vehicle for the entire trip. Of note is that the EGV momentarily travels faster than the preceding vehicle at certain points, but since it had always travelled slower for most of the time to that point it was in no danger of a collision. In addition, the percent gain in ending SOC is tabulated for each case and presented in Figure 5.7. The energy improvement is calculated against three baseline conditions: 1) CV cases where the EGV speed is set to the preceding vehicle speed (torque is optimized), 2) the EA cases where the EGV speed is set to the preceding vehicle speed, and 3) the EA cases corresponding to each optimal EGV trajectory with a preceding vehicle constraint.

![Figure 5.7: Optimal Solution with Preceding Vehicle Energy Gains](image)

As expected, the DP optimal solution has the most increase in ending SOC over the constant speed EA case because this baseline does not optimize speed or torque distribution. The percent increase, in this case, starts equal to the EA case assuming optimal velocity because the preceding vehicle constraint was so restrictive that it only allowed for a single EGV velocity. As the preceding vehicle speed increases, however, the percent gain of the optimal solution over the worst case scenario increases until, at 40
km/h, it equals the same amount of gain as the optimal solution that assumes as open road (see Table 5.1). This also makes sense because the preceding vehicle constraint was no harsher than the velocity constraint enforced on the system to ensure typical city-driving speeds. Looking at the other cases, for preceding vehicle speeds between 25-30 km/h, the DP solution has more increase in efficiency over the optimal speed EA case, which stays consistent at around 20%. For speeds greater than 30 km/h, however, the CV case becomes worse, suggesting that 30 km/h is the limit of where the EGV has enough freedom that optimizing the speed has more effect than optimizing the motor torques.

Next, the effect of the other traffic element, a traffic light, is considered.

5.3.2 Traffic Lights

The method for representing a traffic light signal is outlined in Section 4.3.3. The first case presented is a single traffic light set in the middle of the overall path, at 1500m. For all scenarios, three cases are considered for comparison. First is the optimal trajectory assuming an open road, as presented in Section 5.2, denoted “opt” in the figures. Next is an equal actuation case that assumes a constant EGV velocity of 31 km/h, denoted “EA (v=31)”. Finally, the actual output from the DP algorithm while enforcing the traffic light constraint is show, and is denoted “light”.
Figure 5.8: DP results with Single Traffic Light at 1500m

The first axis shows time on the y-axis and distance on the x-axis for comparison with the other states with respect to distance. The traffic signals are represented by red lines to denote when the EGV cannot pass through, and it is clear that the EGV successfully navigates the specified path while always passing through a green light. Looking at the optimal solution with no traffic constraint, it is clear that the EGV would have hit a red light at around t=150s. The velocity plot shows how the solution was increased from the optimal case in order to allow the EGV to just barely pass before the red light begins.
Once the EGV passes the red light at 1500m, the velocity profile converges back to the optimal trajectory and continues until the end of the trip. Analyzing the SOC graph, it is evident that the light constraint decreased the efficiency, but only by a very small margin, as energy savings of 37% over the EA case are still present.

Figure 5.9 shows the results of the DP algorithm with two additional traffic light arrangements, one with two light locations and the other with three.

Here, it is again clear that, for the light constrained case, the EGV always passes through the traffic light when it is green. In plot Figure 5.9 (a) the velocity is lessened from the unconstrained case so as to make it through the light at 700m just when it turns green. Figure 5.9 (b) shows a similar trend, but velocity adjustment continues until the third
light is reached, where after the velocity profile converges back to the unconstrained case. Looking at the ending SOC of both cases, it is clear that the case with three lights resulted in much less energy savings than the case with two lights. This is to be expected because the case with three lights enforced a speed constraint that lasted much longer than the two light case. Still, even with such significant restriction, savings of about 23% are seen over the naïve case. This suggests that, even with traffic that severely restricts the movement of the EGV, substantial energy savings can still be achieved with an optimization of the speed and in-wheel motor torque.
CHAPTER 6: CONCLUSIONS AND FUTURE WORK

6.1 Summary

This thesis focuses on the modeling and optimal control of a pure electric vehicle across a given terrain. Specifically, it investigates the extent to which simplified traffic constraints affect the optimal speed trajectory of the EGV. By utilizing a discretized model of an EGV traveling over various slopes, a dynamic programming algorithm was able to solve for the optimum motor torques to minimize the energy consumption over any given profile. Initial results assuming an open road show about 40% energy saving gain in the change of battery SOC over a case which assumes constant speed and equal actuation over the entire path. In order to increase the realism of the model, two simplified traffic elements were implemented as constraints on the system. It is shown that a preceding vehicle restricts the velocity distribution of the EGV, but only extremely restrictive cases result in negligible energy saving gain over the baseline case. Similarly, the implementation of various traffic lights results in somewhat less energy saving gain compared to a truly optimal case, but still offers significant benefit over a naïve speed trajectory.

6.2 Contributions

The contributions of this thesis are summarized as follows:
I. Compiling a detailed, although simplified model of an electric ground vehicle. Various elements of the EGV had previously been tested and quantified, but this research combines various elements such as force analysis on a slope, battery characteristics and performance, as well as in-wheel motor efficiencies into a single, cohesive model to be used in a dynamic programming algorithm.

II. Developing methods to model simplified traffic scenarios in such a way that they can be used to enforce constraints on an optimal control problem. This is done for the case of a preceding vehicle as well as a traffic light.

III. Developing a dynamic programming algorithm that varies the speed and in-wheel motor torques to find the optimal operation of an EGV. First, the existing code has been made more useable and robust. Next, the code was adapted to include traffic models in such a way that is not typically native to a dynamic programming procedure.

IV. Quantifying the results of the aforementioned simulations for analysis and suggestions for efficacy in real-world implementation. The general result is that, even with fairly strict traffic constraints, optimizing the speed and torque distribution of an EGV can significantly increase its energy efficiency, and therefore range.

6.3 Recommendations for Future Work

One of the first recommendations for possible future projects is taking into account lateral motion in addition the longitudinal motion already discussed. This can be useful when talking about optimizing motor torques around turns, lane changes maneuvers, etc.
Other than this, however, barring any adjustments to the physical characteristics of the EGV, there is little room for expansion of this project within the confines of a dynamic programming framework. The reason for this is twofold: first, any traffic models must be significantly simplified due to the fact that all states must be known *a priori*. Second, the algorithm requires a significant amount of computation time due to the fact that it is an exhaustive search. This limits its usefulness to offline simulations that can merely suggest the effect of real-world conditions. Another suggestion, therefore, is to begin development of an algorithm that can approximate the optimal solution in real-time. This could be useful for online tests, where the effect of actual traffic and inefficiencies in the EGV can be directly monitored, with the eventual goal of implementation in commercially available vehicles.
REFERENCES


[26] M. Barth, S. Mandava, K. Boriboonsomsin and H. Xia, "Dynamic ECO-Driving for 
Arterial Corridors," in IEEE Forum on Integrated and Sustainable Transportation 
Systems, Vienna, Austria, 2011.

[27] B. Asadi and A. Vahidi, "Predictive Cruise Control: Utilizing Upcoming Traffic 
Signal Information for Improving Fuel Economy and Reducing Trip Time," IEEE 