Addis Ababa University
Addis Ababa Institute of Technology
School of Electrical and Computer Engineering

MULTI-OBJECTIVE OPTIMIZATION OF TRAIN SPEED PROFILES: THE CASE OF AYAT TO MEGENAGNA LINE OF ADDIS ABABA LIGHT RAIL TRANSIT

A thesis Submitted to the Addis Ababa Institute of Technology, School of Graduate Studies, Addis Ababa University
In Partial Fulfillment of the Requirements for the Degree of MASTER OF SCIENCE IN ELECTRICAL ENGINEERING (RAILWAY ELECTRICAL ENGINEERING)

By Tamirat Gebremariam

Advisor: Dr. Mengesha Mamo

July, 2014

Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

MULTI-OBJECTIVE OPTIMIZATION OF TRAIN SPEED PROFILES: THE CASE OF AYAT TO MEGENAGNA LINE OF ADDIS ABABA LIGHT RAIL TRANSIT

By Tamirat Gebremariam

APPROVED BY BOARD OF EXAMINERS

__________________________________ ______________

Chairman, Graduate Committee Signature

Dr. Mengesha Mamo ________________

Advisor Signature

_______________________ _________________

Internal Examiner Signature

_______________________ _________________

External Examiner Signature

DECLARATION

I, the undersigned, declare that this thesis is my original work, has not been presented for a degree in this or other universities, all sources of materials used for this thesis work have been fully acknowledged.

Name: Tamirat Geberemariam Signature: ______________

Place: Addis Ababa Institute of Technology, Addis Ababa University, Addis Ababa

Date of Submission: _July, 2014_____.

This thesis work has been submitted for examination with my approval as a university advisor.

Dr. Mengesha Mamo Signature: _______________

Advisor's Name

ABSTRACT

The plan for the operation of trains on the Addis Ababa Light Rail Transit (AALRT) is based on a fixed interval of riding between stations. Fixed riding time between stations will have the effect of inefficient energy consumption by the trains. Furthermore, the transportation capacity of the network cannot be optimal. By properly managing the reference trajectories the trains use, it is possible to have optimum train operation with respect to energy consumption as well as network capacity.

In this thesis, an optimization of train speed profiles is done. A multi-objective optimization problem has been formulated by making energy and time as the components of the two element objective vector function. A point mass model of the operation of trains has been developed by considering all the important force components acting on the train. The distance to travel between stations is discretized into 20 equal length elements where a two stage solution procedure has been applied to get to the final results. The first stage of the solution procedure is the application of a multi-objective genetic algorithm based optimization technique taking vector of riding modes as the decision variable. Using the developed algorithms for the calculation of cost functions for every type of riding mode, the MATLAB optimization toolbox determines a Pareto-optimal set of riding modes. The second stage of the solution process smoothes out the results found in the previous stage of the solution process without bringing about considerable change in the values of the cost functions.

Different solutions for every section from Ayat station to Megenagna station are generated and they are essentially tradeoff solutions. It has been observed that the fastest ride between stations can be completed within a time of less than 3 minutes. This is equivalent to a 50% reduction in riding time over the plan. By shifting from the fastest to the slowest trajectories, it is possible to save up to 38.18% of energy, while 23.98% reduction in riding time can be achieved by preferring the fastest profiles over the slowest ones.

Keywords: Speed profile, Energy consumption, Running time, Multi-objective, Optimization

ACKNOWLEDGEMENTS

First and foremost, I would like to thank God the almighty for helping me in my life. My special thanks go to my advisor Dr. Mengesha Mamo for encouraging, inspiring and supporting me in every stage of the research work. If it were not for him, nothing could have come out as a result. I do appreciate Ethiopian Railways Corporation (ERC) for giving me the chance to pursue my graduate study in Electrical Engineering for Railway Systems. Finally, I would like to use this chance to thank friends and family who were there for me in the process of the research work.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND SYMBOLS

AALRT	Addis Ababa Light Rail Transit
ATC	Automatic Train Control
ATO	Automatic Train Operation
ATP	Automatic Train Protection
ATS	Automatic Train Supervision
EW	East – West line
GA	Genetic Algorithm
HRT	Heavy Rail Transit
LRT	Light Rail Transit
LRV	Light Rail Vehicle
MO	Multi-objective
MOO	Multi-objective Optimization
PTC	Positive Train Control
TE	Tractive Effort
TKM	Train Kinematic Model
a	Acceleration
$\bar{a}$	Average acceleration
A	Adhesion
b	Deceleration
$\bar{b}$	Average deceleration
B	Brake force
e	Energy consumption in a discrete section
E	Total energy consumption
J	Jerk
M	Effective mass of train
m	Riding mode
n	Number of axles per train
N	Number of discrete distance elements
P	Power
r	Train resistance (lb/ton)

Rc, Rg,Rr	Curvature resistance, Gradient resistance, Rolling resistance
R	Total Train resistance
s	Distance
S_{final}	Stopping point of train
t	Time
v	Speed
w	Axle weight
W	Weight of train
α	Vertical gradient angle
β	Brake force coefficient
μ	Adhesion force coefficient
η	Mechanical Efficiency
ρ	Mass correction factor

CHAPTER ONE

INTRODUCTION

1.1 Background

The demand for a high capacity and high speed transportation system in the city of Addis Ababa has become a serious issue in recent times particularly associated with the increasing population size and the increasing economy of the city. It has also been observed that the existing transportation services which are delivered by taxis and buses have highly increased the amount of environmental pollution in recent times. It was because of these reasons that the government now planned the construction of a two route Light Rail Transit (LRT) system. Since rail transport is environmentally friendly, and since it can provide services at higher speed and capacity, it is expected to improve future transportation of the city [1, 2].

Although a considerable amount of transportation capacity improvement is expected after the project is completed, it is difficult to say that optimum capacity utilization will be achieved. This is because of the fact that the plan for the operation of the trains is based on a fixed riding time [3, 4]. Indeed, capacity of the network is related to the riding time between stations and the number of passengers onboard. Therefore, to improve transportation capacity, it is necessary to reduce the riding time between stations. One of the most common methods to achieve this is by modifying the control strategies that the train driver uses.

The driving strategies are also associated with the energy consumption of the train. There is usually a tradeoff between capacity and energy consumption [5]. It usually is difficult to increase capacity while at the same time reducing energy consumption. It is then important to come up with driving strategies that can be used to improve both energy consumption and transport capacity.

The speed profile of a train is the speed versus time or speed versus distance curve that the train undergoes while it travels between two stations. It has been shown that the optimal speed profile of a train should consist of only four types of driving modes: powering, speed holding (cruising), coasting (power off) and braking [6].

The running train has associated with it various operating costs including energy consumption, riding time, pollution level, passenger comfort, and transportation capacity. Energy consumption by the train corresponds to the dissipation made while the train runs by motoring to compensate for the resistive forces. Riding time is the time it takes for the train to travel between consecutive stations. Pollution level is usually measured in terms of the amount of CO_2 emissions made while the train operates. Passenger comfort corresponds to the upper bound on the level of jerk experienced. Transportation capacity is the total amount of traffic that can be serviced on the network within an hour of operating time [5, 7].

The speed profile optimization problem considers a lot of constraints. These include maximum acceleration rate, maximum braking rate, track alignment, speed restrictions, loading, train resistance, inter-station distance, headway, passenger comfort and signaling type.

Train operation takes on various types of control mechanisms. In recent times, various operating companies have deployed automatic systems that can efficiently act for system variations and that can ensure optimum operation. These systems include Automatic Train Protection (ATP), Automatic Train Supervision (ATS), Automatic Train Operation (ATO) and Positive Train Control (PTC). Such systems are dynamic systems that can instantaneously gather train status information and process such information within a very short time interval to output train control information. The output from these systems is used to drive control circuits that are deployed on the railway network. Due to the highly robust and reliable subsystems that such systems incorporate, it is usually not cheap to deploy them [7].

Addis Ababa Light Rail Transit (AALRT) is an urban railway transit project that is under construction. It includes a North-South line of 16.674 km and an East-West line of 16.998 km. Both lines have a common path of length 2.61 km. Nine LRT stations are placed in the Ayat-Megenagna line. The longest interval is 1250 meters and the shortest interval is 725 meters [1, 3].

The operation of trains on the AALRT network is planned to be based on a manual drive without the deployment of any of the above automatic control mechanisms [3]. A driver based operation of the trains shall be usually dependent on the actions of the driver when it comes to setting the combinations of riding modes and the speed switching points.

1.2 Statement of the Problem

A number of previous researches have been done on the development of an optimal train control strategy to minimize energy consumption. Only few researches considered travel time as an objective [5, 8, 9]. Furthermore, most of the papers [11, 12, 13, 17] did consider only a few set of constraints on the system.

Since there will not be an automatic supervision system on AALRT to guide the operation of the trains, the driver will take full responsibility as to what kind of reference speed profiles to use. This means that, the operation of the trains cannot be fully optimal with respect to minimizing both energy and time.

It is desirable to develop optimal speed profiles for the operation of the trains on the AALRT network by considering both energy and time as the objectives and by taking into account the effect of every kind of system constraint. It can then be assured that maximum possible energy savings can be made while at the same time improving network capacity.

Since the formulation of a single objective optimization problem makes the problem incomplete, by resulting in a single solution which will be best for one objective and worst for the other, it is better to shift to a multi-objective optimization technique that is driven by the notion of Pareto-optimality. Multi-objective optimization can determine equally important solutions in a single run. In fact, it is possible to get multiple solutions that obey the tradeoff between the two antagonistic objectives, while at the same time improving both objectives as much as possible [14].

1.3 Objectives of the Study

1.3.1 General Objective

This research aims to determine Pareto-optimal set of driving strategies for the operation of the trains on the railway line from Ayat station to Megenagna station of AALRT. Multi-objective optimization (MOO), which is preferred when there are multiple objective functions to be minimized, will be used. MOO can generate multiple tradeoff solutions in a single run. The effects of track alignment, speed restrictions, acceleration and braking limits, jerk limit, stopping distance, adhesive force and maximum Tractive Effort (TE) will be integrated in the solution process.

1.3.2 Specific Objectives

The specific objectives of this thesis are:

- To evaluate different speed profile optimization techniques and choose the best among them
- To model the operation of a train
- To formulate a multi-objective optimization problem by considering time and energy as the objectives
- To choose from various decision variables to be used in the optimization process
- To determine optimal speed profiles as tradeoff solutions for every section from Ayat to Megenagna
- To post-process train speed profiles

1.4 Methodology

After reviewing various related materials, a Train Kinematics Model (TKM) will be developed that is used to calculate various parameters associated with the operation of the train. These parameters include energy consumption, travel time, acceleration and braking rate, tractive effort, train resistance, and stopping distance.

Since Energy consumption and riding time are considered as the objectives to be minimized, a two objective optimization problem will be formulated and a multi-objective optimization technique will be used to search for Pareto-optimal solutions. Discrete space based modeling shall be adopted in this research by discretizing the distance to travel between stations. The set of vectors of riding modes that are finally found as solution represents well distributed tradeoff solutions that can be taken by a decision maker as to choose from one another.

Since there are only four types of riding modes in an optimal speed profile [6], the principle of determination of optimal set of riding modes makes it even more easier and faster as compared to other methods which are based on a vector of speed or time values as the decision variable. Multi-objective optimization using the optimization toolbox of MATLAB will be used in the solution process. The optimizer essentially uses the TKM to get values for cost functions. A post-processing will be finally applied to the results of the optimization process to result in speed profiles that can be applied in the actual scenario of AALRT. Figure 1.1 summarizes the methodology used in this thesis.

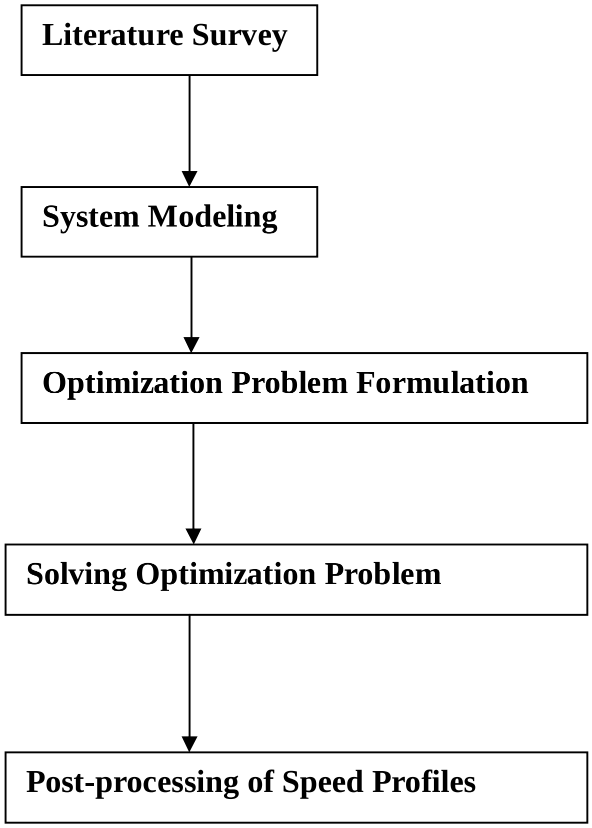


Figure 1.1 Summary of the methodology

1.5 Thesis Organization

This thesis is organized into six chapters. Chapter 1 introduced about the situation of transportation in the city of Addis Ababa and discussed the requirements for the determination of optimal speed profiles to minimize costs of operation. Chapter 2 summarizes the efforts of previous studies related to the optimization of train speed profiles. Chapter 3 presents the development of a typical Train Kinematics Model that can be used to compute various train parameters. Chapter 4 introduces a speed profile optimization approach that is based on the determination of an optimal set of riding modes. Chapter 5 presents the results of a MATLAB based multi-objective optimization and discusses in detail various solutions. Chapter 6 concludes the study and suggests for future work.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The motion of a train between two consecutive stations is constrained by various parameters such as the alignment of the track, the status of the signals, the separation of the stations, the nature of the train and various requirements of the passengers. There are usually specified travelling time requirements by the passengers. At the same time, the operating company needs to conserve the operating energy. These two requirements of the operating company and of the passengers are usually antagonistic to one another that it is usually difficult to satisfy both of them at the same time. Various researches have been conducted in the past to come up with optimal riding modes that are meant to bring about the minimization of some kind of cost function [5, 6, 7, 11, 12, 13, 15, 16, 17, 18]. This chapter reviews some of the most important previous works related to the optimization of train speed profiles.

2.2 Train Kinematics Model

The motion of a train can be modeled and represented by the various force components and the motion quantities that act on it at a particular time and location [19]. The force components that act on the train include weight of the train, Tractive Effort (TE), rolling resistance, air resistance, gradient resistance, curvature resistance, brake effort and adhesion. Figure 2.1 shows a simplified diagram of the forces acting on a three car train. The resistance component is the sum of all kinds of resistances acting on the train including track resistance, rolling resistance and air resistance.

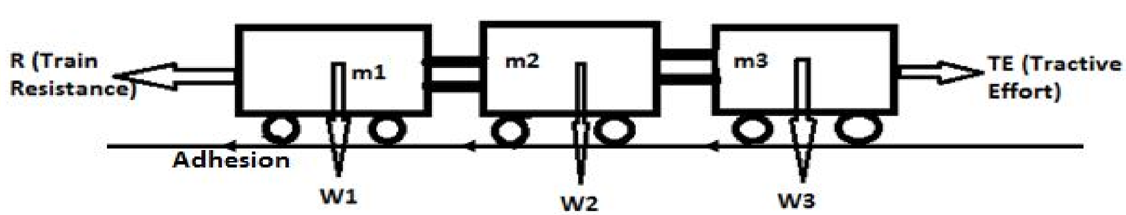


Figure 2.1: Simplified force diagram of a three car train

2.2.1 Tractive Effort and Adhesion

Moving a train along a route involves many force components, including TE, resistance, braking force and train weight. While the TE provides a necessary force to move a train, resistance is the force that opposes the movement and speed of the train. To accelerate or decelerate a train, the TE must be transferred between wheels and the running surface of the rail through a friction force, called adhesion [7]. TE is the mechanical force available at the rim of the driving wheels of the train. It is usually related to the speed of the train as [19]:

$$TE_{max} = P * \eta / v \tag{2.1}$$

Equation 2.1 shows that the relationship between the maximum TE and the speed of the train is an inverse relation. P is the maximum power developed by the traction motors and is usually a constant value. The transmission efficiency (η) corresponds to the nature of the mechanical force transmission system. Generally the total traction power (P) is the sum of the individual maximum power values developed by each of the traction motors [20].

$$P = \sum_{i=1}^{n} Pi \tag{2.2}$$

The TE available on the driving wheels of the train is bounded by the maximum amount of adhesion available between train wheel and the rail. Adhesion accounts for the maximum amount of TE that can be available to drive the vehicle. It is the product of the weight of the train by an adhesion coefficient. The coefficient of adhesion is a function of speed and takes on various forms [20].

$$A = \mu * M * 9.81 \tag{2.3}$$

$$TE = min\{TE_{max}, A\} \tag{2.4}$$

In equation 2.3, M is the effective mass of the train which is actual mass of the train multiplied by a mass correction factor ρ (usually taken as 1.04) [8].

The relationship between train speed and adhesion coefficient is usually a nonlinear one and it is difficult to derive mathematical expressions directly. It is usually through field data followed by mathematical interpolation that some relations are formulated in the past. The relationship between speed and adhesion coefficient is usually formulated as [21]:

$$\mu = 7.5/(v + 44) + 0.16 \tag{2.5}$$

where v is in kph and μ is unit less. This formulation was made after running a German class 19 electric locomotive up to 160 kph in 1943 and is employed in estimating adhesion coefficients at any given speed in various countries including Germany, Austria and Switzerland [21]. A formulation made after running an electric locomotive in France gave the following relationship [7]:

$$\mu = 0.24 * (8 + 0.1v)/(8 + 0.2v) \quad (2.6)$$

again here v is in kph while μ is unit less.

2.2.2 Train Resistances

To determine whether the propulsion system of a train is able to operate with speed (v), the total resistance, denoted as R, must be known. On a level track alignment without wind effect, it was found that the total resistance can be expressed by a quadratic equation formulated as [22]

$$R = C_1 + C_2V + C_3 V^2 \quad (2.7)$$

where the coefficients C_1, C_2 and C_3 are dependent on the axle load, number of axles, cross section of the train, and shape of the train. It was found that C_1, varies with the number of axles, axle load, and track type, and increases linearly with the number of axles, while C_2 and C_3 vary with train length and the front or rear area of the train, respectively [22].

Davis equation, which was developed and validated by the data from the Pennsylvania and Burlington railroads, is expressed as [23]:

$$r = 1.3 + 29/w + bV + CAV^2/wn \quad (2.8)$$

where r is unit resistance in pounds per ton; w is weight per axle in tons; b is an experimental coefficient based on flange friction, shock, sway, and concussion. C is the drag coefficient based on the shape of the front end of the car or locomotive; V is the speed of the train in mph; and A is the cross-sectional area in square feet of the car or locomotive. Later, the modified Davis equation was developed in 1970 by Committee 16 of the American Railway Engineering Association (AREA). Its intent was to recognize changes in resistance factors, increased train operating speed, and improved track conditions over the earlier days. The modified Davis equation is thus developed and formulated as [24]:

$$r = 0.6 + 20/w + 0.01V + KV^2/wn \quad (2.9)$$

where K, the air resistance coefficient, is 0.07 for cars, 0.0935 for containers, and 0.16 for trailers on flatcars. Both the Davis and the modified Davis equations were derived for calculating unit resistance of a train, which considered weight per axle, number of axles per car, and the degree of aerodynamic and drag effects. Yet another formulation for train resistance was made by the Canadian National Railway and is expressed as [10]:

$$R=1.5+18n/w+0.03v+CaV^2/(1000w) \qquad (2.10)$$

Where r=the rolling resistance of the vehicle in lb/ton

n=number of axles

W=total weight in tons of train

V=velocity in mph

C=Canadian National streamlining coefficient

A=cross sectional area of the train

The value of C for modern lightweight Passenger Equipment =2.0. Because of its applicability to light weight train cars in modern days, the formula presented in equation 2.10 is preferred to be used in this research. In addition to the rolling resistance and wind resistance, additional resistance terms were proved to exist in the operation of the train. Some of the most important types of these are track gradient resistance and track curvature resistance. Track grade resistance is proportional to the angle (in degree) of the inclined track and can be directly derived from the relationship between train weight and the track grade. It was found that the grade resistance was 20 lb/ton per track grade (in percentage). On the other hand, the resistance associated with a horizontal track curvature was determined by field tests and experiments. It was found that the resistance due to horizontal curvature was 0.8 lb/ton per track curvature (in degrees). The total resistance is the sum of all resistive forces acting on the train, which are measured in pounds per ton [20].

2.3 Previous Work on Speed Profile Optimization

Although energy consumption has been the objective of many researches in the past, travel time is also being considered on some of the most recent studies. Although there are a lot of different approaches for the minimization of train energy consumption, the most commonly used one is through proper management of the operation of trains [7]. This led to the development of optimal riding trajectories that could be used as references. It has been shown that the optimal trajectory consists of only four types of riding modes [6]. Therefore, the determination of optimal trajectory consists in the determination of the combination of riding modes together with the set of switching points.

The problem of speed profile optimization has been analyzed by a number of authors. The type of the optimal strategy consisting of four successive control levels (full power, speed holding, coasting and full braking) was introduced by Howlett et al. [6]. The optimal strategy consisting of constraint set including variable track profile has been studied by Howlett and Cheng [11]. They determined the optimal driving strategy for a typical diesel-electric train. Various assumptions were made on their work. These included the consideration of constant gradient in a discrete distance element, assumption of a resistance parameter linear to the train speed, consideration of a constant braking acceleration and a discrete set of control inputs. The solution procedure was purely analytical and the results were put in terms of predefined system parameters.

Pavel Pokorny [12] has done an extensive work on the analytical procedure for an electric train. He used nonlinear parametric programming to come up with the optimal driving strategy which defined the values of the set of driving modes, and the corresponding switching points. The paper defined the objective to be a weighted sum of both the energy consumption and ride time. Constraints included track profile, global speed limit, a resistance term which is linearly related to the speed, schedule ride time and station separation. The results of this paper put the optimal switching points in terms of system parameters and a computational method was required to calculate them.

Wang et al. [13] used mixed linear integer programming to formulate the problem. They discretized the travelling distance and assumed that for a discrete distance element, system parameters such as gradient, speed limit, and resistance are constant. They approximated the initial nonlinear problem with a set of linear problems. The result of the solution procedure was a set of optimal switching points and speed values. An existing solver was used to solve final solution values and it was shown that computational time reached 10 minutes to generate optimal trajectories.

Analytical methods of solving the optimization problems were not accurate and fast particularly when the problem considers a lot of constraints. Dynamic programming was considered as a powerful method for quite complex problem formulation, yet it was shown to be very slow.

Ko et al. [15] used dynamic programming in the optimization of train speed profiles. They defined the optimization problem by using energy consumption as the objective to be minimized. A set of state equations that can act as constraints are then defines. Boundary conditions are transformed into a penalty function that can be included in the objective. State equations are time uniform discretized and linearized using first order Taylor series expansion and Trapezoidal rule for the approximation of integration. Finally, the overall problem equation is reduced into discrete sets such that dynamic programming (DP) can be applied. The final model contains an N-stage decision process, which is executed using digital computer. The paper then showed optimal speed profiles first for a plain track without speed limits, then for an actual track with varying gradient and speed restrictions. The results show that boundary conditions are satisfied within 0.6m and 0.1m/s. Computational error is also found very small to be applied for a practical scenario.

Jong and Chang [16] developed a Train Performance Simulation model. The paper models the operation of a train using two different types of optimization problems. These are the shortest operation time and the proper operation time problems. The first case considers ride time between stations as the objective to be minimized while the later defines a term which is a function of both the ride time and energy consumption. In both cases it considers equations of motion and speed restrictions as constraints. Additional resistance parameters such as tunnel resistance and starting resistance are also taken into account. In addition to this, speed restrictions due to various elements such as switches, curvatures, blocking signals, etc, are considered in the problem definition. The paper shows that the first type of problem, the short operation time problem, which results in speed profiles with a sharp variation in speed from maximum acceleration to full braking, cannot be applicable in practice due to higher discomfort level and wear on the system. The second problem, however, can be optimized to get a proper setting in both the ride time and the energy consumption. Direct method is used by the discretization of one of the independent variables, time, distance or velocity, to solve the problem. Results show that the optimal trajectories that are achieved using this method are accurate when compared to that of commercial software.

Search techniques using heuristic algorithms seemed to deal with problem complexity and resulted in a more accurate solution although they were sometimes too slow when compared with gradient methods.

Kang [17] worked on the determination of the optimal trajectory by calculating a single coasting point in an inter station run. It was assumed that for a short distance ride such as in a metro system, the optimal speed profile consists of only three types of riding modes, by neglecting the cruising phase. It aimed to determine the coasting point that will satisfy distance and time requirement, as well as minimizing energy consumption. Train Performance Simulation (TPS) and GA blocks are designed using simulink. The objective in the paper was the minimization of energy consumption. Two trajectories are first calculated. These are the forward direction trajectory consisting of power driving and coasting, and a backward trajectory consisting of the braking curve. The braking curve is constructed such that the train can stop exactly at the point of the next station. The braking point is defined as the point where the two trajectories meet.

The coasting point is determined using genetic algorithm (GA). The algorithm defines the difference between actual driving time and the target driving time as the fitness function to be minimized. Several speed profiles, which could satisfy the distance and time requirements, are first generated. They are finally compared by their associated energy consumption value. This is achieved by repeatedly executing GA. The fitness function is defined as [17]:

$$F = K * |(t - t_d)/t_d| \tag{2.11}$$

Where:

t is the actual drive time.

t_d is the target drive time.

Coasting points are encoded into chromosomes to form an initial population of solutions. After calculation of the value of F, two parent chromosomes are selected based on their fitness value and crossover is applied on them to result in two offspring. Mutation operation is applied on the offspring at a rate of 2%. Speedy convergence is achieved by preserving the parent chromosomes. The GA process is done until the determined search time is reached. Eight chromosomes make up the initial population, each with a size of 9 bits. This means that we can encode the coasting point value to be within 1m and 512m. This selection is made for a particular case the paper studied. The two chromosomes with the highest order of fitness are chosen and one bit is chosen from the 9 bits in the chromosome to determine the point of crossover. Three different trajectories are generated each with different coasting points and associated energy consumption. The trajectory with the least energy consumption is finally chosen as the optimal one.

Wong and Ho [18] worked on the optimization of train running trajectory by the determination of multiple coasting points on an inter-station run. With the increase in the number of coasting points, an efficient speed profile can be achieved that can handle variations on the track profile, local speed limitations, etc. A weighted sum of both the energy consumption and the ride time formed the objective. Because of the need to have faster convergence, MARK (Minimum Allele Reserve Checker) is introduced as an improvement over genetic algorithm. The paper determined optimal trajectories for single point, two point and multi point coasting. Simulation results are generated for both the single point and multi point coasting scenarios.

All the above methods are single objective optimization methods, particularly focusing on the minimization of energy consumption, and they cannot be directly applied to the scenario where multiple objectives are defined. For instance, the ride time in AALRT is required to be reduced by the passengers while the railway company wants to reduce energy consumption. This is a particular scenario which triggers the need for the definition of a multi-objective optimization problem.

R´emy Chevrier [5] used evolutionary algorithm to optimize the problem formulated using two objectives, energy and time. A special type of multi-objective optimization algorithm, Indicator Based Evolutionary Algorithm (IBEA) is used in this paper. The whole distance to ride between consecutive stations is partitioned into sub-sections. For each subsection are defined entrance speed (V_0), exit speed (Vx) and intermediate speeds (V_1 and V_2). The evolutionary algorithm is responsible to calculate the three speeds (V_1, V_2 and Vx). Algorithms are developed for all types of riding modes, acceleration, cruising, coasting and braking. The particular mode to choose depends on the corresponding speed values. For instance, the entrance phase is between V_0 and V_1. If $V_0<V_1$, then an acceleration mode is executed. Similarly, the exit phase is between V_2 and Vx. The intermediate phase is the profile section between V_1 and V_2. Any of the 4 types of riding modes could exist in the intermediate phase of the profile. The aim of this paper was to

determine the optimal set of speed values within a section. These values are searched for by the evolutionary algorithm. A case study has been made on an existing railway network and results were analyzed. It was shown that the results were diversified and accurate.

2.4 Train Control Regimes

In general, train control for most transit operations represents a cycle of different motion regimes, including acceleration, cruising, coasting, and braking. Figure 2.2 shows the most commonly used train control types [7]. Control I is used when the distance to travel is shorter. Control II operation drives shorter travel time but consumes more energy, compared to those in controls III and IV. Control III operation is commonly used for reducing energy consumption. By using control IV operation, the consumed energy can be further reduced, despite the longest travel time.

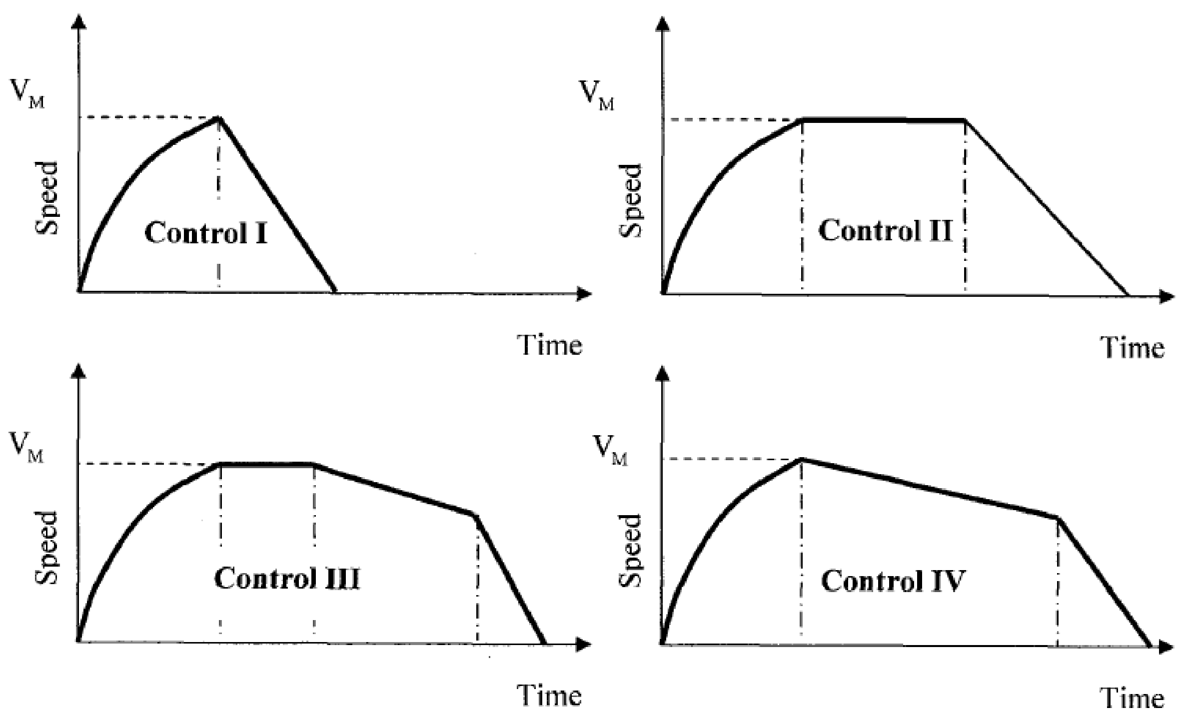


Figure 2.2: Four cases of inter-station train control regimes [7].

CHAPTER THREE

TRAIN KINEMATICS MODEL

3.1 Introduction

In this chapter special focus is given to the kinematics of the train that will be used on the AALRT. Some of the most important information about the Light Rail Vehicles (LRV) was collected and is presented in table 3.1. Using this set of data, together with the alignment of the track, it can be possible to model the actual performance of the trains on AALRT.

Number of cars per train	3
Number of power bogies	2
Number of unpowered bogies	1
Number of axles per bogie	2
Number of electric motors per bogie	2
Power per electric motor	130 KW
Total loaded mass of train	63.02 ton
Mass per axle	10.05 ton
Maximum speed of train on a level track	70 kph
Maximum speed on a level crossing	50 kph
Maximum jerk	1 m/s^3
Minimum average acceleration	0.5 m/s^2 for 0<v<40 kph, 1 m/s2 for 0<=v<=70 m/s^2
Minimum average deceleration from maximum speed	1m/s^2
Minimum average deceleration while at emergency brake	2 m/s^2
Average travelling speed	>=20 kph
Average dwelling time	30 sec
Number of passengers per train	317 (8 persons/m^2) ; 254(6 persons/m^2)

Table 3.1 AALRT LRV data [25, 26].

Section	From (m)	To (m)	Level Crossing Range
EW1-EW2	21050	22300	21940-21960
EW2-EW3	19940	21050	20067-20216
EW3-EW4	19080	19940	-
EW4-EW5	18220	19080	18990-19010
EW5-EW6	17500	18220	-
EW6-EW7	16520	17500	17142-17287
EW7-EW8	15440	16520	15790-15810
EW8-EW9	14600	15440	-

Table 3.2: Railway track information from Ayat station to Megenagna station [1, 26]

3.2 Force Equations

TE can be computed by equating the work done at the rim of the driving wheel with that performed by the torque or turning effort of the engine or motor. In general, the engine power consumed for the TE is limited not to exceed the adhesion between wheel and track; otherwise wheel slip will occur and the locomotive will lose traction. Adhesion is a function of the friction at the point of wheel-rail contact, the adhesion coefficient is often taken as 0.25, which represents the percentage of locomotive weight that is available as effective TE. Since the adhesion coefficient, denoted as μ, of a train has non-linear characteristics to its corresponding speed, denoted as v, it is difficult to derive mathematically, but it can be obtained mainly through field tests [7].

$$TE = \min\{\frac{P*\eta}{v}, A\} \quad (3.1)$$

The value of η is usually taken as η=0.7 [7]. Given that there are four motors on the train, the total power delivered by the motors equals: $P=4*P_{motor}=4*130$ KW=520 KW per train.

$$Adhesion\ (A) = \mu * M * 9.81 \quad [N] \quad (3.2)$$

$$\mu = (\frac{7.5}{3.6*v+44} + 0.16) \quad (3.3)$$

$$A(v) = 98.966 + 4636.7/(44 + 3.6 * v) \quad [N] \quad (3.4)$$

The maximum tractive effort is given as:

$$TE_{max} = P * \eta/v \quad [N] \quad (3.5)$$

The effective tractive effort is the minimum of the maximum TE and adhesion [7].

$$TE = \min\{TE_{max}, A\} \quad [N] \quad (3.6)$$

$$TE = \min\{.7 * \frac{520}{v}, A\} \quad [KN] \tag{3.7}$$

$$TE_{max} = 364/v \ [N] \tag{3.8}$$

$$TE = \min\{\frac{364}{v}, A\} \quad [KN] \tag{3.9}$$

$$v_{max}=70kph=19.4mps \tag{3.10}$$

The brake force is usually given as a constant multiple of the train weight [27]:

$$B = 9.81 * M * \beta \ [N] \tag{3.11}$$

The coefficient of brake force is usually taken as β=0.09 [27]. (3.12)

Therefore, $B = 55640.56\ [N]$ (3.13)

Using the Canadian National version of train resistance formula, which can be used for various train types, we have the following formula for the rolling resistance of the train:

$$r = 1.5 + 18n/w + 0.03v + cav^2/1000 * w \ [kg/ton] \tag{3.14}$$

$$R_r = 9.81 * M * r \ [N] \tag{3.15}$$

$$R_r(v) = 1.9868 + 0.0415 * v + 0.0011 * v^2 \quad [KN] \tag{3.16}$$

There are two components of resistance associated with the track alignment, gradient resistance (Rg) and curvature resistance (Rc).

$$Rg = 9.81 * M * tan\alpha \ [N] \tag{3.17}$$

Where α is the vertical gradient inclination of the track.

$$R_c = M * 9.81 * 700/radius \ [N] \tag{3.18}$$

$$R = Rc + Rg + Rr \ [N] \tag{3.19}$$

The plots for the TE, Adhesion, Resistance and Brake Effort (BE) are shown in the figures 4.1 to figure 4.3. We can see that the effect of Adhesive force on the available TE is considerable at smaller speed values. We can also see that BE is a constant force to be applied at the brake discs on the wheel axle set of the train cars. The fact that it is constant accounts for the uniform brake deceleration rate of the train.

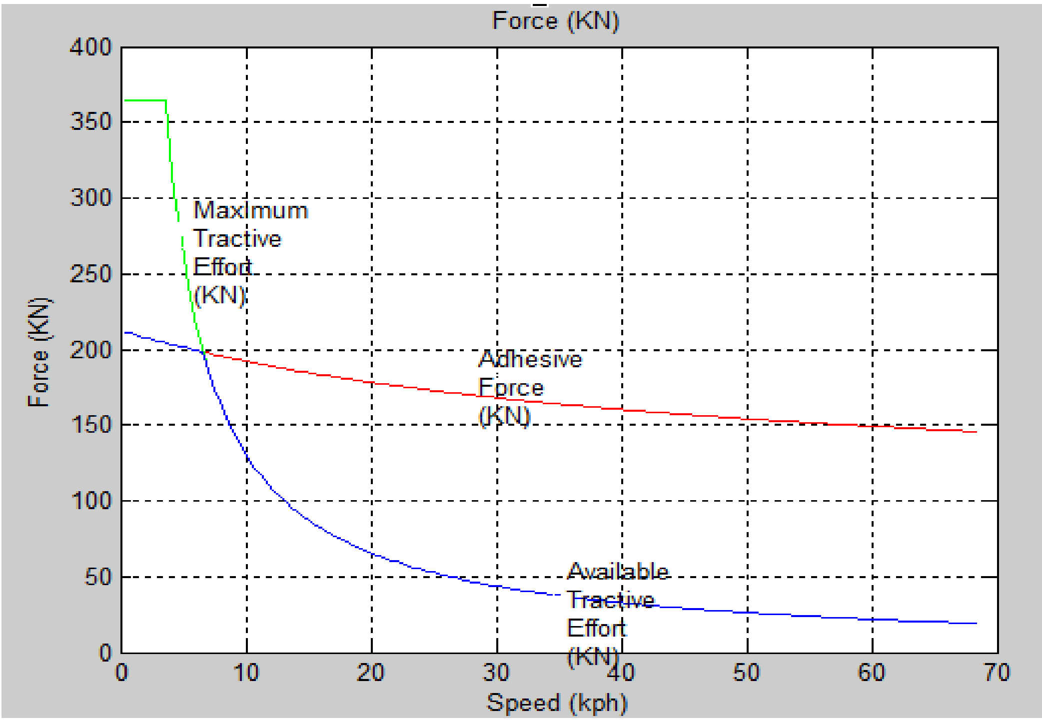


Figure 3.1: Tractive Effort and Adhesive force

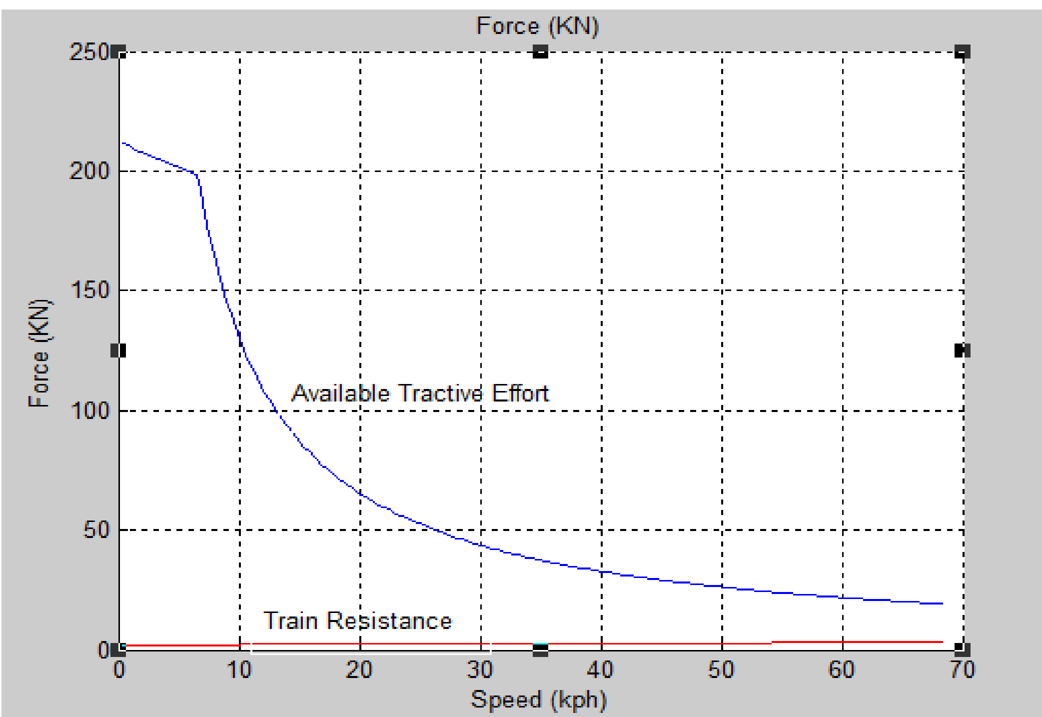


Figure 3.2: Available Tractive Effort and Train Resistance

Note from figure 4.3 that train resistance values are very small as compared to the other forces, tractive effort and braking effort. This is due to the rather small weight of the train cars constituting the LRT train.

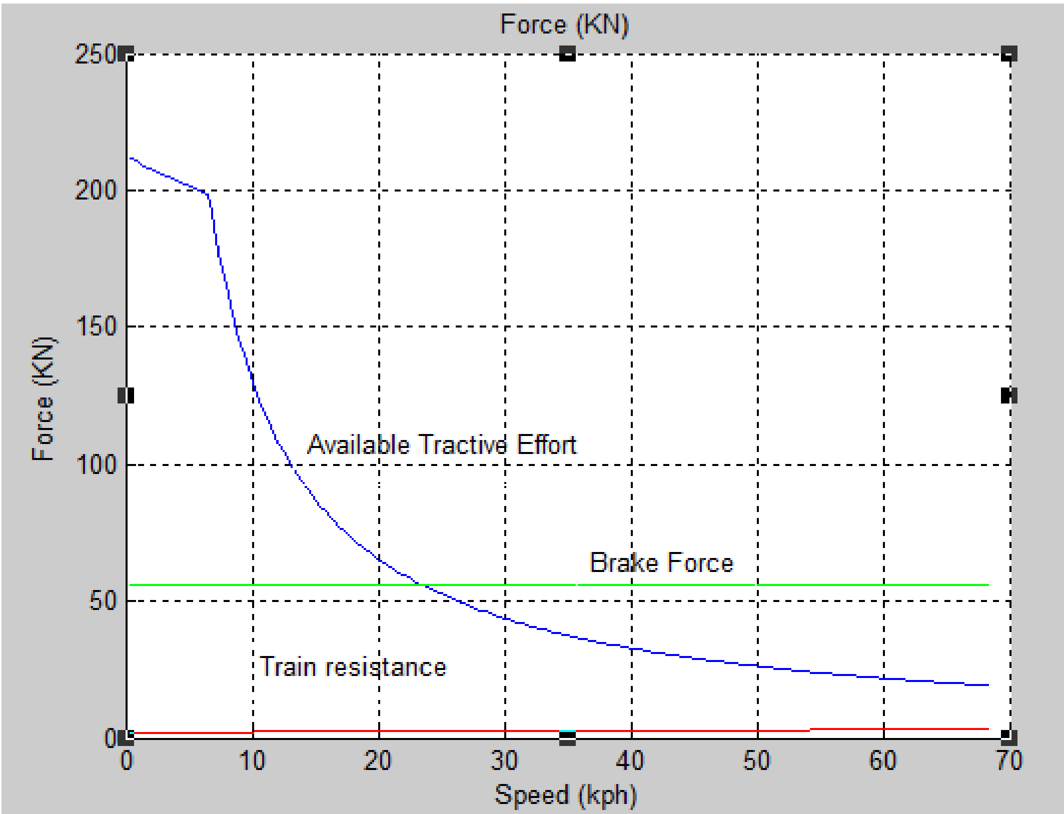


Figure 3.3: Tractive Effort, Brake Effort and Train Resistance

3.3 Discrete Space Model of Train Kinematics

In this thesis, discretization of the distance to travel between stations is made to model the system. It is then followed by the determination of cost values for every discrete element for any type of riding mode. The calculation is made by using speed as the independent variable. In that case, the values of energy consumption and riding time are represented in terms of speed. Figure 3.4 shows the discretization of distance into N points. Every point can be expressed by the corresponding values for parameters like speed, time and distance. The following variables are used in the model formulation.

v represents the speed of the train

s represents the location of the train e represents the energy consumed by the train

TE is tractive effort

t represents time

a is train acceleration

Δt represents change in time

Δe is change in energy

Δv is change in speed

Δs is change in distance

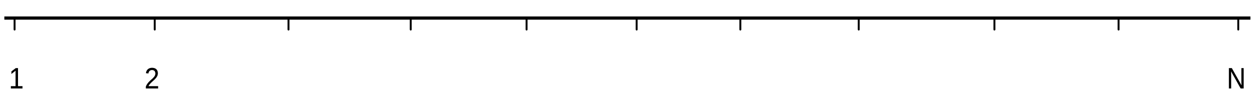


Figure 3.4: Discretization of the distance to travel into N elements.

Now using motion equations [16],

$$\Delta s = v\Delta t \quad (3.20)$$

$$\Delta v = a\Delta t \quad (3.21)$$

It follows that,

$$\Delta t = \Delta v / a \quad (3.22)$$

Assuming that there are N discrete speed values:

$$v_i \in \{v_1, v_2, v_3, \dots v_N\}. \quad (3.23)$$

Assuming that v is constant for a small discrete speed element [16],

$$\Delta s_i = v_i * (t_{i+1} - t_i) \quad (3.24)$$

$$\Delta s_i = v_i * (v_{i+1} - v_i) / a_i \quad (3.25)$$

$$s_{i+1} = s_i + v_i(v_{i+1} - v_i) \quad (3.26)$$

$$S = \sum_1^N \Delta s_i \quad (3.27)$$

$$\Delta e_i = TE_i * v_i * \Delta t_i \quad (3.28)$$

$$E = \sum_{i=1}^{N} \Delta e_i \quad (3.29)$$

$$\Delta t_i = (v_{i+1} - v_i)/a_i \quad (3.30)$$

$$t = \sum_{i=1}^{N} \Delta t_i \quad (3.31)$$

$$E = \sum_{1}^{N} (TE_i * v_i * (v_{i+1} - v_i)/a_i) \quad (3.32)$$

$$a_i = (TE_i - R_i - B_i)/M \quad (3.33)$$

$$t = \sum_{1}^{N} (v_{i+1} - v_i)/a_i \quad (3.34)$$

$$average\ acceleration\ (\bar{a}) = \frac{v_j - v_1}{t_j - t_1} \quad (3.35)$$

$$average\ deceleration\ (\text{Б}) = |\frac{v_j - v_N}{t_j - t_N}| \quad (3.36)$$

Jerk is the rate of change of acceleration. It has the units of m/s^3. It is formulated as [7]:

$$Jerk = |\frac{a_{j+1} - a_j}{t_{j+1} - t_j}| \quad (3.37)$$

CHAPTER FOUR

A TRAIN SPEED PROFILE OPTIMIZATION PROBLEM

4.1 Introduction

In this thesis, a new approach for speed profile optimization, discrete space based modeling followed by the determination of an optimal set of riding modes is used. Previous researches particularly those focusing on the use of evolutionary computation have set the decision vector in the problem formulation to be a vector of terminal speed values within a discrete distance element [5, 8, 9].

Multi-objective optimization, unlike single objective optimization, focuses on the determination of an optimal set of decision vectors. The mappings of those vectors in the objective space are essentially tradeoff solutions that none of them is better than any other one. The notion is based on the principle of pareto optimality. A multi-objective optimization problem can be formulated as [28]:

$$\text{Min } \mathbf{F}(\mathbf{P}), \mathbf{P} \in \mathbf{R}^{n}, \tag{4.1}$$

where $\mathbf{P}=\{p_1,p_2,\ldots,p_n\}$ is an n-dimensional vector having n decision variables or parameters and defines a feasible set of **P**. $\mathbf{F}=\{f_1,f_2,\ldots,f_m\}$ is an objective vector with m objective components to be minimized, which may be competing or non commensurable to each other [28].

Usually the components of the objective vector are antagonistic to one another so that it is impossible to minimize all of them at the same time. Therefore, the main concept in multi-objective optimization focuses on the determination of a solution vector $x \in \boldsymbol{P}$ that results in fairly distributed tradeoff points [28, 29].

4.2 Pareto Dominance

An objective vector F_a in a minimization problem is said to dominate another objective vector F_b, denoted by $F_a \prec F_b$, iff $f_{a,i} \leq f_{b,i}\ \forall\ i \in \{1,2,\ldots,m\}$ and $f_{a,j} < f_{b,j}\ \exists\ j \in \{1,2,\ldots,m\}$ [29]. Figure 4.1 shows Pareto-optimal solutions in a two dimensional objective space.

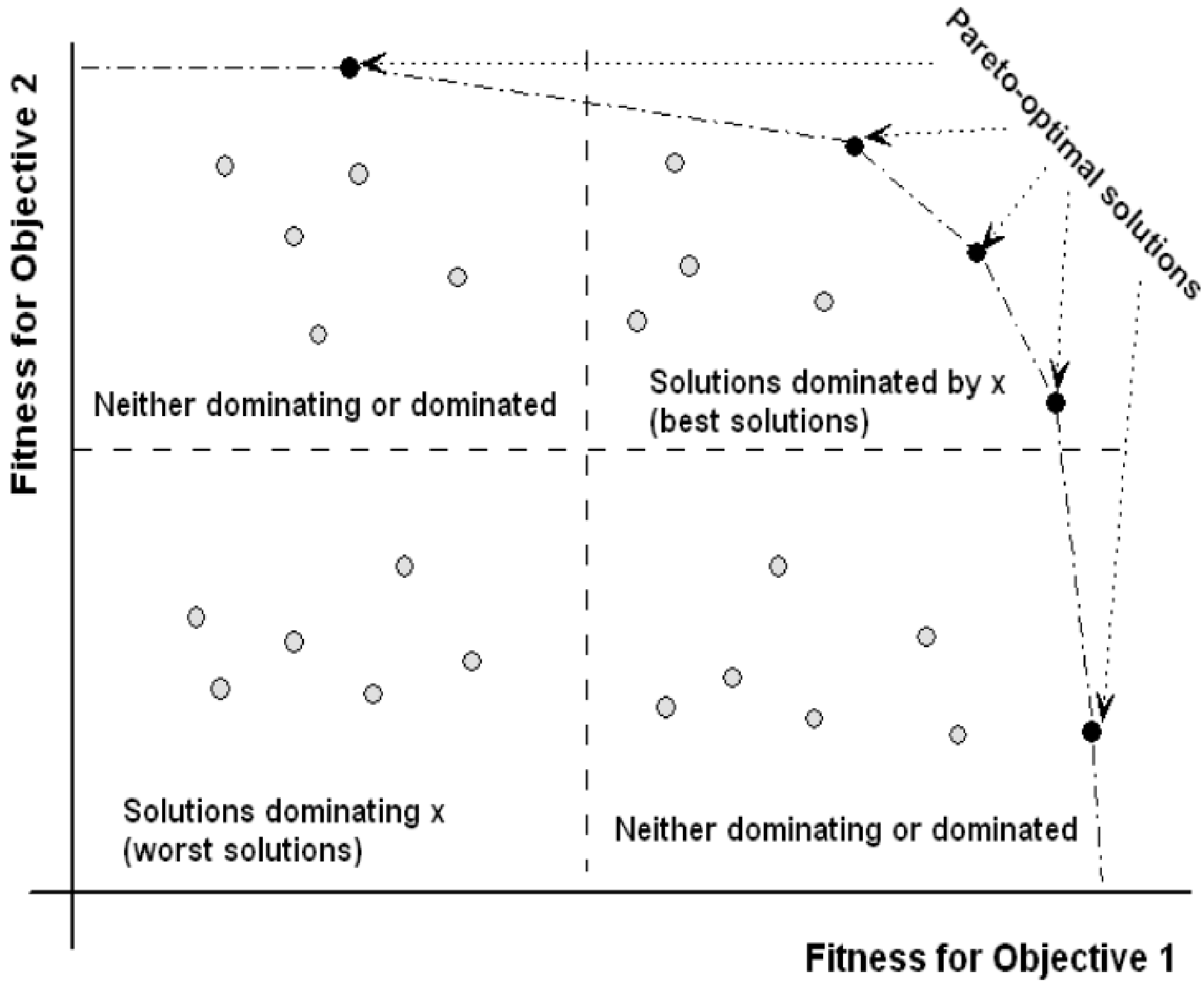


Figure 4.1: Concept of Pareto optimality [29].

The Pareto optimal solutions make a tradeoff curve in the objective space. The points on this tradeoff curve are usually the ones that are searched for by a multi-objective optimization.

4.3 Problem Formulation

The speed profile optimization problem can be formulated by considering the discrete space model for the operation of trains presented in the previous chapter. Every discrete point has associated with it a particular speed value. Hence, a vector of terminal speed values can be taken as the decision variable in the problem formulation. Let v_j represents the speed value at a discrete point j where $j \in \{1,2,3,\ldots,N\}$. The constraint set includes speed restriction at a particular point, boundary values for speed and distance, intermediate speed values, limits for average acceleration and deceleration rates, and maximum jerk. Equations 3.20-3.37 are used in the formulation of the optimization problem as:

$$\text{Min}\ (E, t) \tag{4.3}$$

Subject to:

1. Speed restrictions

$$v_j \leq v_{j,max} \tag{4.4}$$

2. Boundary conditions

$$v_1 = v_N = 0 \tag{4.5}$$

$$S_N = S_{final} \tag{4.6}$$

3. Non zero intermediate speed

$$v_j > 0\ \forall\ j \in \{2,3, \dots N-1\} \tag{4.7}$$

4. Minimum average acceleration

$$\frac{v_j - v_1}{t_j - t_1} \geq \{0.5 \text{ for speed up to 40 kph and 1 for speed up to 70 kph}\} \tag{4.8}$$

5. Minimum average deceleration

$$\left| \frac{v_j - v_N}{t_j - t_N} \right| \geq 1 \tag{4.9}$$

6. Jerk limit

$$\left| \frac{a_{j+1} - a_j}{t_{j+1} - t_j} \right| \leq 1 \tag{4.10}$$

Where

$$E = \sum_1^N (TE_j * v_j * (v_{j+1} - v_j)/a_j) \tag{4.11}$$

$$a_j = (TE_j - R_j - B_j)/M \tag{4.12}$$

$$t = \sum_1^N (v_{j+1} - v_j)/a_j \tag{4.13}$$

4.4 Vector of Riding Modes as the Decision Variable

The basic principle that this study uses to optimize train speed profile is that optimum profile consists of only four types of riding modes. Whatever the type of procedure used, the end result is always a set of these four types of modes. It is therefore important to use this fact to choose from different types of decision variables. The decision variable can be speed vector or time vector as has been applied by many researches in the past [5, 8, 9]. In those papers, the distance to travel is descretized into equal length sections and the values of speed or time at the ends of these discrete sections makes up a decision vector. As can be shown in figure 4.2 the distance from s0 to s4 is discretized into 4 sections of equal length. The solution procedure then tries to determine a combination of decision variable values at the terminals of each section. That is the decision vector could be a vector of terminal time values where it is represented as t={t0,t1,t2,t3,t4}. A vector of terminal speed values is represented by v={v0,v1,v2,v3,v4}. Formulation of the decision variable to be a vector of speed or time will have the direct impact of widening the search space. This can be shown by the following paragraphs.

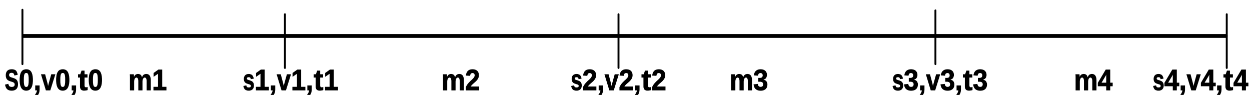


Figure 4.2: Decision variables to be used in a discrete space representation.

Here, {s1,s2,s3,s4} represents discrete distance values. An optimization problem can aim to determine an optimum vector of time, speed or riding mode. We assume that we have the following relation for the bound constraints of the speed vector: 0<vj<20, where vj is an element of the speed vector or equivalently the train speed at one of the terminal of section j. Again assuming that motoring is the riding mode in the first section and that braking is the riding mode in the last section, m1=1 and m4=4. The problem now gets reduced to determining only v2 and v3. In order to arrive at the optimal combination of these speed values, at least considering only integer speed values, we need to search within a region of $2^{(20)}$ possible points.

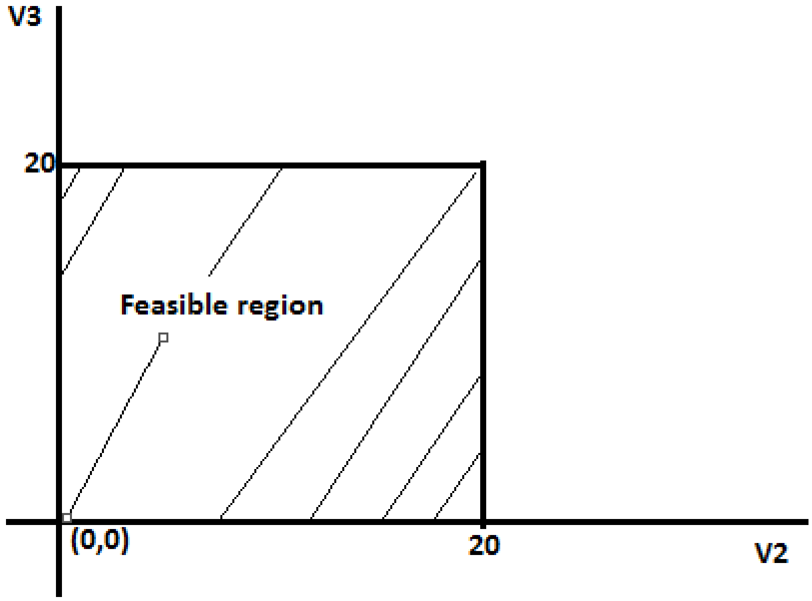


Figure 4.3: Feasible region by considering terminal speed as the decision variable

Now consider a riding mode based decision space. For the same type of problem shown above, it is required to determine the optimal combination of modes m2 and m3, because it is already assumed that m1 is always motoring and m4 is always braking. Each mode {m2 or m3} takes on four different integer values corresponding to the four types of riding modes. Let motoring=1, cruising=2, coasting=3 and braking=4 be the values assigned to the four types of riding modes. The following is the maximum possible combination of modes.

m2	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4
m3	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4

Table 4.1: Possible combination of riding modes for a 2 section discrete space

That is, there are only 16 different combinations of riding modes to be searched for to finally get to the optimum combination of riding modes. From these points, it can be seen that terminal speed based optimization of train speed profile will have a continuous decision space, and it widens the search space as compared to the case where a vector of riding modes is taken as the decision variable. In fact, if there are N discrete sections in which it is required to determine the optimal set of riding modes, a total of $4^{(N)}$ different combinations of modes make up the decision space. Indeed, this is a much smaller value if compared to the size of the decision space made by the vector of continuous speed values. It follows that, it is better to choose riding mode vector as the decision variable.

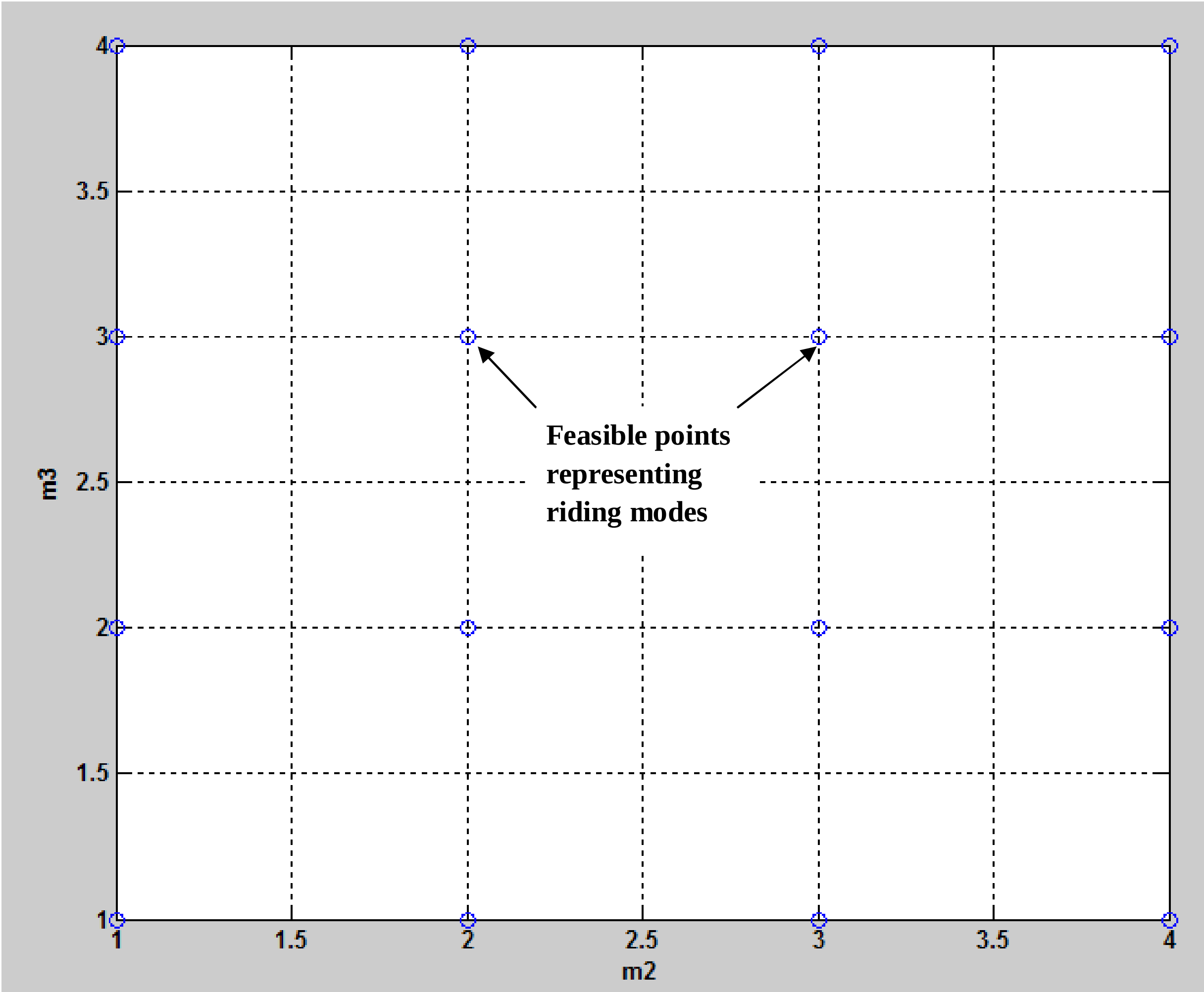


Figure 4.4: Feasible region with riding modes as the decision variable.

A two step optimization process will be done in this thesis: the first being determination of optimum combination of Pareto-optimal riding modes. The seconds step is actually a post-processing step which is aimed to result in smooth speed profile and make sure that unwanted combinations of riding modes cannot appear in the final speed profile. Unwanted combination of modes could be, for instance, a braking followed by acceleration, which is not necessary because of the associated higher jerk. We do have the following assumptions in the solution process:

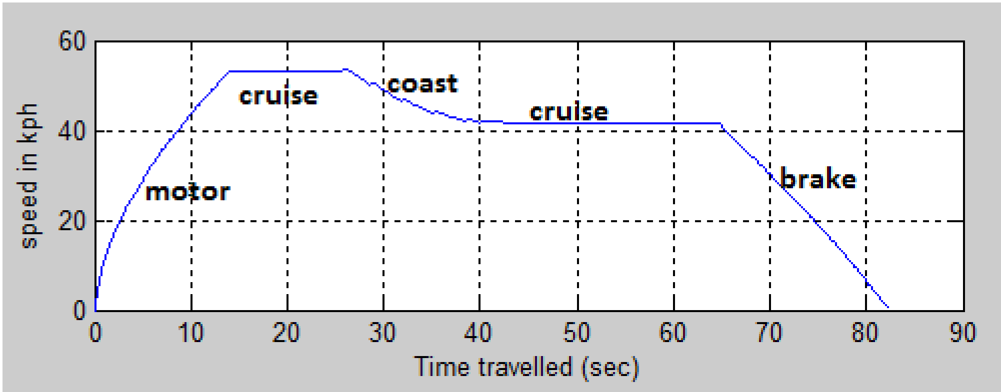
1. It is assumed that the starting mode of every journey should be motoring, while the final mode should be braking, to ensure that the train stops at the next station.
2. Continuously varying speed profile usually consumes more energy than the one which has a few switching points. Therefore, the post processing phase shall focus on the minimization of the number of switching points.
3. The train should not be driven at very low speeds so that there is no probability of the train to come to stop before arriving at the next station. In order to ensure this requirement, braking and coasting operations should be triggered only at some elevated initial speed values.
4. The inter-stations distance is discretized into 20 sections of equal length.

After applying the above assumptions, only the intermediate 18 sections are left to be solved because the riding modes in the first and the last sections are already known. It follows that there are a total of $4^{(18)}$ combinations of riding modes to search from to arrive at the optimum. The following are two examples of combinations of riding modes for 20 sections path. Their corresponding speed profiles are shown in figure 4.5.

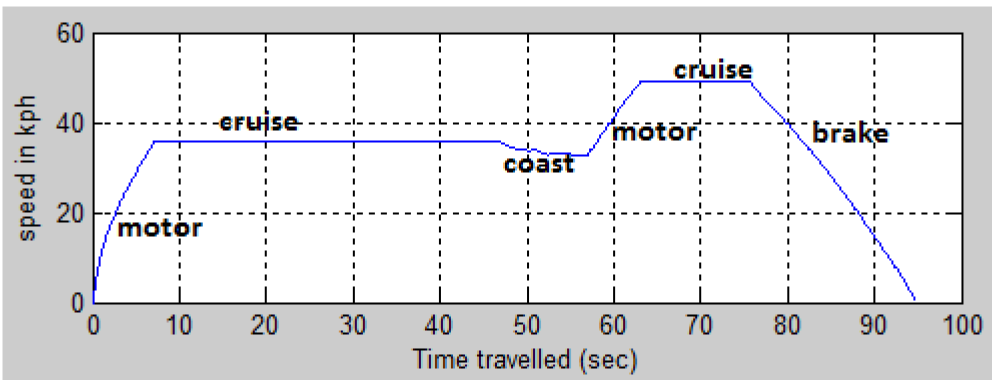
m1=[1 1 1 2 2 2 2 3 3 3 3 3 3 2 2 2 2 2 4 4]

m2=[1 2 2 2 2 2 2 2 2 2 3 3 3 1 2 2 2 2 4 4].

Here m1 corresponds to a speed profile consisting of the combination of riding modes motoring, cruising, coasting, cruising and braking. On the contrary, m2 is a vector of riding modes equivalent to motoring, cruising, coasting, acceleration and finally braking. For these two different combinations of riding modes, there are two different associated costs (energy and travel time). Generally, a smaller value of energy dissipation corresponds to a longer interval of travel time.



(a) Speed profile corresponding to vector m1.



(b) Speed profile corresponding to vector m2.

Figure 4.5: Representation of riding mode vectors with speed profiles

Every discrete element is represented by a riding mode (motoring, cruising coasting or braking). The first and the last elements are assumed to have motoring and braking modes. By considering speed as the independent variable, the following formulations are made. Note from figure 4.6 that the distance element from point 5 to 6 consists of multiple iteration points with incremental speed values in the computation of the riding modes. In fact, every discrete element consists of multiple iteration points that correspond to incremental speed values. Let j represent the index of a discrete distance element and let i represent the index of the iteration point within a discrete distance element.

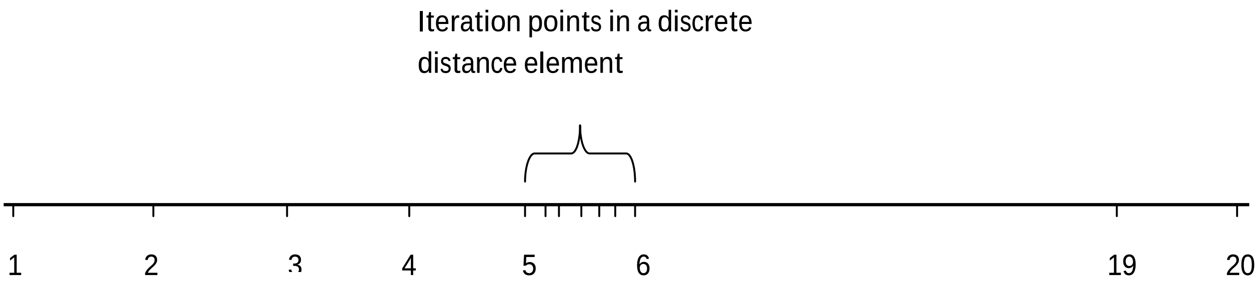


Figure 4.6: Distance discretization with 20 elements

It follows that:

$j \in \{1,2,3,\ldots,20\}$ (4.14)

$i \in \{1,2,3,\ldots,N_j\}$ (4.15)

here N_j is the maximum number of iterations within a discrete distance element to calculate riding regime by considering speed as the independent variable. Parameters are then represented by using these two index values. For instance, v_{ji} is the speed value in section j after i iterations. Assuming that the total number of iterations in the last discrete element N_{20}=k, the optimization problem formulation of equation 4.3 can be modified as in the following:

$$\text{Min}\ (E, t) \quad (4.16)$$

Subject to:

1. Speed restrictions

$$v_{ji} \leq v_{ji,max} \quad (4.17)$$

2. Boundary conditions

$$v_{11} = v_{20\,k} = 0 \quad (4.18)$$

$$S_{20\,k} = S_{final} \quad (4.19)$$

3. Non zero intermediate speed

$$v_{ji} > 0\ \forall\ j \in \{2,3,\ldots 19\}\ and\ \forall\ i \in \{1,2,3,\ldots Nj\} \quad (4.20)$$

4. Minimum average acceleration

$$\frac{v_{ji}-v_{11}}{t_{ji}-t_{11}} \geq \{0.5 \text{ for speed up to 40 kph and 1 for speed up to 70 kph}\} \quad (4.21)$$

5. Minimum average deceleration

$$\left|\frac{v_{ji}-v_{20\,k}}{t_{ji}-t_{20\,k}}\right| \geq 1 \tag{4.22}$$

6. Jerk limit

$$\left|\frac{a_{j+1\,i}-a_{ji}}{t_{j+1\,i}-t_{ji}}\right| \leq 1 \forall\ j \in \{1,2,3,\ldots 20\}\ and\ \forall\ i \epsilon \{1,2,3,\ldots Nj\} \tag{4.23}$$

Where

$$E = \sum_{j=1}^{20}\sum_{i=1}^{N_j}(TE_{ji} * v_{ji} * (v_{j\,i+1} - v_{j\,i})/a_{ji}) \tag{4.24}$$

$$a_{ji} = (TE_{ji} - R_{ji} - B_{ji})/M \tag{4.25}$$

$$t = \sum_{j=1}^{20}\sum_{i=1}^{N_j}(v_{j\,i+1} - v_{ji})/a_{ji} \tag{4.26}$$

The motoring phase requires the application of maximum TE [5]:

$$TE_{ji} = \min\left\{\frac{364}{v_{ji}}, A\right\}[KN] \tag{4.27}$$

Coasting and Braking phases have no need for the application of TE [5]:

$$TE_{ji} = 0 \tag{4.28}$$

Cruising phase requires the application of some amount of TE to make the train move at constant speed. If the train resistance is positive, the TE should be equal to the resistance. Otherwise, if the resistance is negative, the applied TE must be zero and braking force should be applied to compensate for the resistance value [5]:

$$TE_{ji} = \{R_{ji,}0\} \tag{4.29}$$

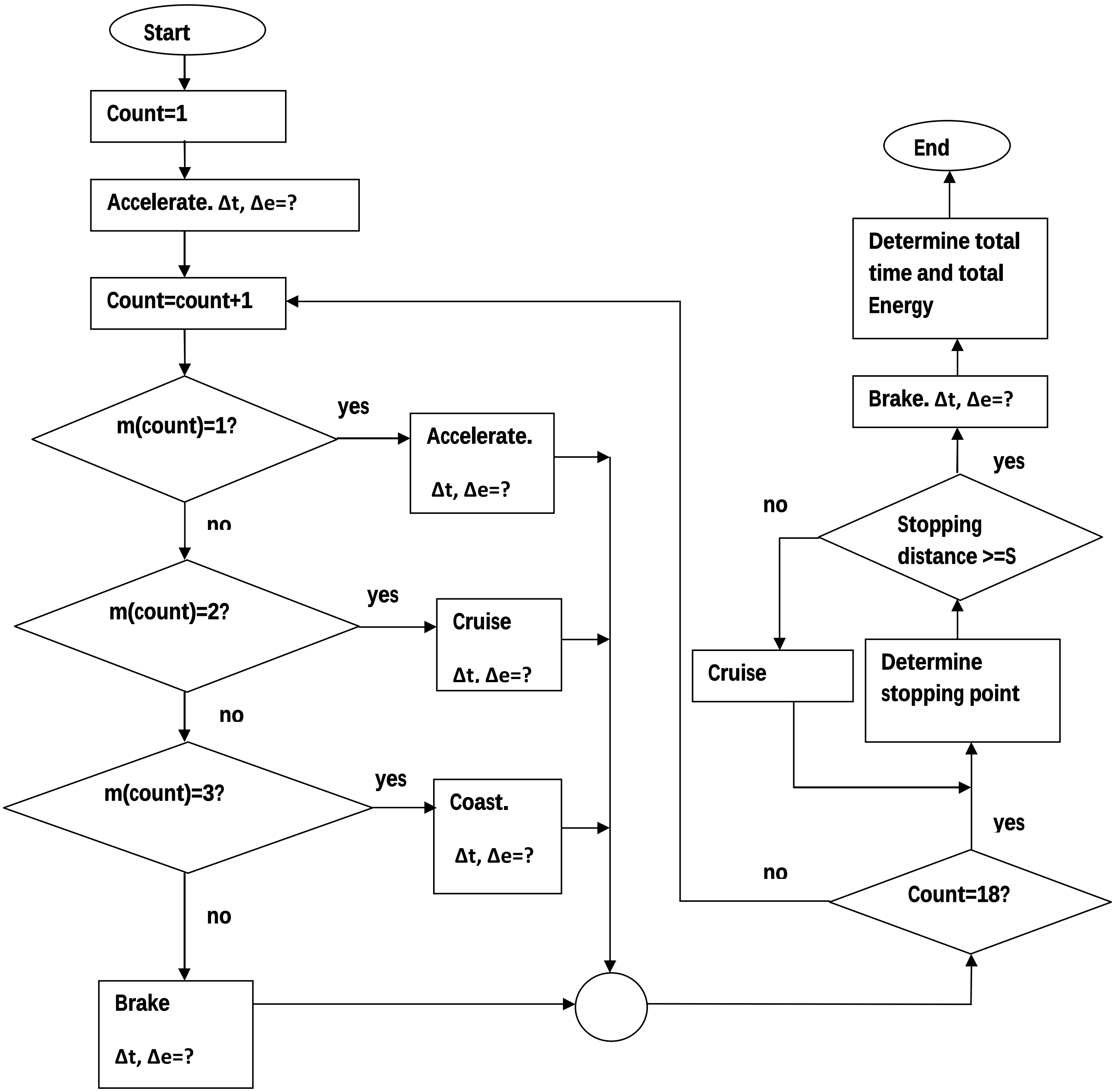


Figure 4.7: Flowchart of Speed profile construction and cost calculation

Figure 4.7 shows the process of constructing speed profiles for an input vector of riding modes. Note that the first discrete distance element is provided with acceleration phase. Note also that in order to stop at the exact location, the braking point is calculated starting from the eighteenth section. If the stopping point is calculated to be before the actual stopping point, then the train has to cruise until the correct braking point is achieved.

4.5 Algorithms

In this section, various algorithms that make up the TKM are explained. Algorithms 1 through 4 are used to calculate driving regimes. That is they are used to calculate energy consumption and riding time associated with a riding mode. For instance, algorithm 1 calculates the amount of energy consumption by the acceleration phase for a given initial speed, location, error value of previous mode, gradient and location vectors. Algorithm 5 calculates the total energy consumption and the total riding time by an input set of riding modes. Algorithm 6 is used to determine the baking distance. The inputs for the algorithms are the values of tractive effort, total resistance, brake effort that are calculated using equations 3.9, 3.19, and 3.13, respectively.

Algorithm 1 : function accelerate(v(1),s(1),e,T,R)

Data: v(1): the initial speed of the train at the beginning of the distance segment, t(1): the initial time value, s(1): the initial location of the train, e: previous trajectory error, tractive effort (T), Resistance (R), len: length of discrete distance element, M: mass of the train.

Result: (Δt Distance Δe err): a vector containing the time spent, the distance travelled, the energy dissipated, and the error from the end of the subsection

1. Energy=0; count=1;
2. While s(count)<=(len+s(1)+e) && v(count)<=v_{max}
3. if T> R

    v(count+1)=v(count)+1;

    t(count+1)=t(count)+M/(T(count)-R(count));

    s(count+1)=s(count)+v(count)*(t(count+1)-t(count));

    Energy=Energy+T(count)*v(count)*(t(count+1)-t(count));
4. elseif T<R

    v(count+1)=v(count)-1;

    t(count+1)=t(count)-M/(T(count)-R(count));

    s(count+1)=s(count)+v(count)*(t(count+1)-t(count));

   end
5. count=count+1;
6. end

7. Δt=t(count-1)
8. Δe=Energy;
9. Distance=s(count-1)
10. err=s(1)+len+e-Distance;

Algorithm 2 : function brake(v(1),s(1),e,R,B)

Data: v(1): the initial speed of the train at the beginning of the distance segment, t(1): the initial time value, s(1): the initial location of the train, e: previous trajectory error, tractive effort, Resistance, brake effort, len: length of discrete distance element, M: mass of the train.

Result: (Δt, Δe, Distance, err): a vector containing the time spent, the distance travelled, the energy dissipated, and the error from the end of the subsection

```
1. count=1;
2. while s(count)<=(len+s(1)+e) && v(count)>=(2)
3. if B(count)+R(count)>0
       v(count+1)=v(count)-1;
       t(count+1)=t(count)+M/(B(count)+R(count));
       s(count+1)=s(count)+v(count)*(t(count+1)-t(count));
4. elseif B(count)+R(count)<0
       v(count+1)=v(count)+1;
       t(count+1)=t(count)-M/(B(count)+R(count));
       s(count+1)=s(count)+v(count)*(t(count+1)-t(count));
   end
5. count=count+1;
6. end
7. Δt=t(count-1)
8. Δe=0;
9. Distance=s(count-1)
10. err=s(1)+len+e-Distance;
```

Algorithm 3 : function cruise(v(1),s(1),e,R)

Data: v(1): the initial speed of the train at the beginning of the distance segment, t(1): the initial time value, s(1): the initial location of the train, e: previous trajectory error, Resistance vector, len: length of discrete distance element.

Result: (Δt Δe Distance err): a vector containing the time spent, the distance travelled, the energy dissipated, and the error from the end of the subsection

1. Energy=0; count=1;
2. while (s(count)<=(len+s(1)+e) && v(count)>=2 && v(count)<=v_{max})
3. v(count+1)=v(count);
4. t(count+1)=t(count)+1;
5. s(count+1)=s(count)+v(count);
6. if R(count)>0

    Energy=Energy+R(count)*v(count);

   end
7. count=count+1;

   end
8. Δt=t(count-1)
9. Δe=Energy;
10. Distance=s(count-1)
11. err=s(1)+len+e-Distance;

Algorithm 4 : function coast(v(1),s(1),e,R)

Data: v(1): the initial speed of the train at the beginning of the distance segment, t(1): the initial time value, s(1): the initial location of the train, e: previous trajectory error, Resistance vector, len: length of discrete distance element, M: mass of the train.

Result: (Δt Δe Distance err): a vector containing the time spent, the distance travelled, the energy dissipated, and the error from the end of the subsection

1. Energy=0; count=1;
2. while s(count)<=(len+s(1)+e) && (v(count)>=2 && v(count)<=v_{max})
3. if R(count)>0

```
        v(count+1)=v(count)-1;
        t(count+1)=t(count)+M/(R(count));
        s(count+1)=s(count)+v(count)*(t(count+1)-t(count));
4.  elseif R(count)<0
        v(count+1)=v(count)+1;
        t(count+1)=t(count)-M/(R(count));
        s(count+1)=s(count)+v(count)*(t(count+1)-t(count));
    end
5.  count=count+1;
    end
6.  v1=v(count-1);
7.  Δt=t(count-1)
8.  Δe=0;
9.  Distance=s(count-1)
10. err=s(1)+len+e-Distance;
```

Algorithm 5: function total_cost(m)

Data: m: a vector consisting of riding modes as the components

Result: (E,t) the amount of energy dissipation, and total riding time as a result of using the input vector of riding modes

```
1.  err=0;
2.  (t(1),e(1))=accelerate(v(1),s(1),err,T,R);
3.  for i=2:19
        if m(i)==1
              (t(i),e(i))=accelerate(v(i),s(i),err,T,R);
        elseif m(i)==2
              (t(i),e(i))=cruise(v(i),s(i),err,T,R);
        elseif  m(i)==3
              (t(i),e(i))=coast(v(i),s(i),err,T,R);
        elseif m(i)==4 && v(i)>=8
              (t(i),e(i))= brake(v(i),s(i),err,T,R,B);
```

```
        elseif m(i)==4 && v(i)<8
                (t(i),e(i))=accelerate(v(i),s(i),err,T,R);
        end
    end
4.  (t(20),e(20))=brake(v(20),s(20),err,T,R);
5.  E=sum(e); t=sum(t)
```

Algorithm 6: function brake_distance(v(1),s(1),e,R,B)

Data: v(1): the initial speed of the train at the beginning of the distance segment, t(1): the initial time value, s(1): the initial location of the train, e: previous trajectory error, resistance, brake force, M: mass of the train

Result: (brake_distance): the stopping location.

```
1.  count=1;
2.  while v(count)>=0
            v(count+1)=v(count)-1;
            t(count+1)=t(count)+M/(B(count)+R(count));
            s(count+1)=s(count)+v(count)*(t(count+1)-t(count));
            count=count+1;
    end
3.  brake_distance=s(count-1);
```

The MATLAB built-in multi-objective optimization algorithm essentially evolves population of riding mode vectors to arrive at solutions that are very close to the pareto front and that are fairly distributed. Shown in figure 4.9 is the MATLAB script to generate optimal speed profiles. It makes use of the MATLAB built in functions gaoptimset and gamultiobj. The functions f12 and fc12 are the cost calculation functions for the section EW1-EW2. f12 calculates the total energy consumption while fc12 calculates riding time. The function gaoptimset sets different parameters the optimizer uses. These parameters include the population size, crossover rate and maximum number of generations. The handles of the cost calculating functions are passed to the function gamultiobj. The result of the optimizer is shown in figure 4.9. It can be shown that the optimization process converged. Each row of the x matrix represents a vector of riding modes. In fact, by rounding off the elements of the output matrix, as can be shown in figure 4.10, it is possible to get elements ranging from 1 to 4. The values of the elements represent a riding mode, and a row vector with 20 elements represents the combination of riding modes that make up a speed profile. Different rows correspond to different solutions that are essentially tradeoff to one another. The variable y represents the energy consumption for the corresponding speed profile represented by a row from the matrix x.

Constraints are handled implicitly within the calculation of costs and by using the smoothing procedure. For instance, speed limits, track resistance, adhesive limit, and maximum TE are defined in the algorithms for the calculation of costs for every kind of mode. The stopping distance requirement is handled within the speed profile construction process where the breaking distance is calculated and the stopping point error is minimized. The smoothing process minimizes the jerk value available in the speed profiles. The other constrains, average acceleration rate and average deceleration rates are analyzed after the solutions are generated and those solutions which do not satisfy the requirements are dropped.

```
lb=ones(1,20);
ub=4*lb;
f1=@(m)f21(m);
f2=@(m)fc21(m);
h=@(m)[f1(m) f2(m)];
options=gaoptimset('generations',500,'populationsize',16,...
'CrossoverFraction',0.9,'EliteCount',6);
[x y]=gamultiobj(h,20,[],[],[],[],lb,ub,options);
m=smz(round(x));
[a b]=size(m);
for i=1:a
    pfun(m(i,:),g,1);
end
```

Figure 4.8: MATLAB script used to generate optimal speed profiles.

```
Optimization terminated: average change in the spread of Pareto solutions less than options.TolFun.

x =

  Columns 1 through 10

    3.3355    3.1435    2.6543    2.7225    3.1947    2.6103    2.5473    2.6436    2.7883    2.7757
    1.6173    1.3090    1.1465    2.7772    1.4416    1.3475    2.1809    1.9919    2.3768    1.2490
    3.3355    3.1435    2.6543    2.7226    3.1947    2.6103    2.5473    2.6436    2.7883    2.7757
    2.8554    2.6831    2.3837    2.7281    3.1355    2.5707    2.5624    2.6098    2.7676    2.7338
    1.6173    1.3090    1.1465    2.7772    1.4417    1.3475    2.1809    1.9919    2.3768    1.2490

  Columns 11 through 20

    3.5221    2.4213    2.5694    2.9678    2.8035    1.8690    3.1475    2.1160    2.7707    2.9598
    1.3301    1.7153    1.2369    1.7234    1.1807    1.1116    2.2195    1.1065    1.7001    2.4238
    3.5221    2.4213    2.5694    2.9678    2.8036    1.8690    3.1476    2.1160    2.7707    2.9598
    3.1815    2.4007    2.5459    2.0776    2.6903    2.0787    3.0689    2.0095    2.5894    2.8206
    1.3301    1.7153    1.2369    1.7234    1.1807    1.1116    2.2195    1.1064    1.7001    2.4238


y =

  1.0e+007 *

    0.4266    0.0000
    4.0845    0.0000
    0.4266    0.0000
    0.7881    0.0000
    4.0845    0.0000
```

Figure 4.9: output of the optimizer

```
>> m=round(x)

m =

  Columns 1 through 16

     3     3     3     3     3     3     3     3     3     3     4     2     3     3     3     2
     2     1     1     3     1     1     2     2     2     1     1     2     1     2     1     1
     3     3     3     3     3     3     3     3     3     3     4     2     3     3     3     2
     3     3     2     3     3     3     3     3     3     3     3     2     3     2     3     2
     2     1     1     3     1     1     2     2     2     1     1     2     1     2     1     1

  Columns 17 through 20

     3     2     3     3
     2     1     2     2
     3     2     3     3
     3     2     3     3
     2     1     2     2
```

Figure 4.10: Vector of riding modes

4.6 Smoothing of Speed Profiles

The results that are achieved as a result of the multi-objective optimization by using vector of riding modes as the decision variable are usually not feasible for application in the actual operation of the trains because of the following reasons:

1. The Pareto-optimal solutions usually have many switching points and that it is difficult for the driver to apply them at the correct instant.
2. The resulting speed profiles have multiple ups and downs that result in unwanted jerk on the motion of the train.

It is therefore important to process the results of the optimizer to come up with suitable speed trajectories that do have lower jerk and smaller switching points [5]. But it is difficult to smooth out the initial speed profile without affecting the values of associated cost functions. In order to transform the initial speed profile into one that is suitable for application, we need to consider some important assumptions.

I. Because of the small resistance values of the LRT trains, the costing phase is usually a very small positive or negative acceleration. For very small values of acceleration or deceleration, the coasting regime can be approximated by a cruising regime.

Figure 4.11: A small acceleration or deceleration rate of coasting regime can be approximated by a cruising regime.

II. A cruising regime is a constant speed regime. A cruise controller of the train is responsible to maintain the speed of the train at a constant value by changing the input tractive effort to the traction motors. The cruising regime is usually approximated by continuous acceleration and coasting regimes. We can make use of this fact to approximately replace consecutive sequence of acceleration and coasting with cruising regimes [5].

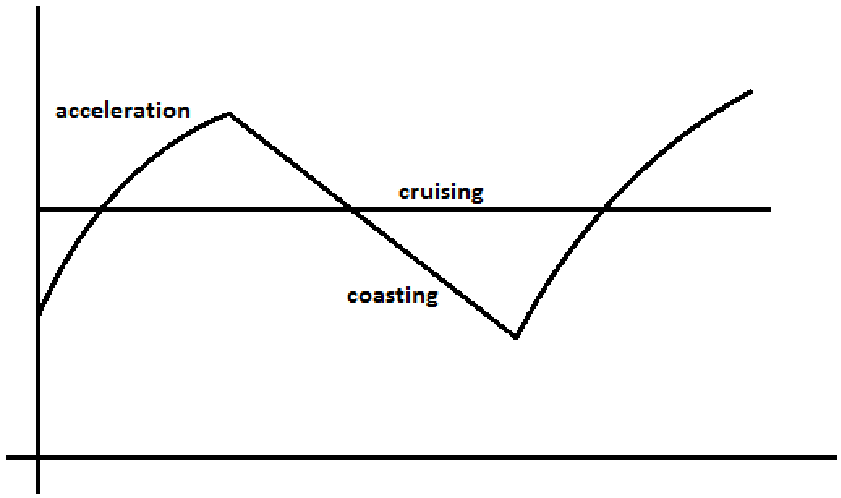


Figure 4.12: an acceleration followed by a coasting regime can be replaced by cruising regimes.

III. Shifting of riding modes: refers to the replacement of acceleration or a coasting phase with a cruising phase. As can be seen in figure 4.13, the original trajectory consisting of the motion regimes a1, c1 and a2 is transformed in to a new trajectory consisting of a1, a2' and c2'. We see that there is an associated error as a result of the transformation represented by the shaded region in the figure. The area of the shaded region is usually equivalent to the cost function. For instance if the curve is speed versus time, then the area is a distance element. It follows that as a result of the transformation, an error in the distance to travel occurs. This error value can be minimized by collecting acceleration phases that are separated by a short cruising phase. That is, we can collect same riding modes that are separated by a short cruising phase without bringing about too much error on the cost function while at the same time reducing the number of switching points and smoothing out the initial speed profile.

Figure 4.14 shows the result of the smoother applied on the optimizer output shown in figure 4.10. It can be seen that there are smaller number of switching points after the smoothing operation.

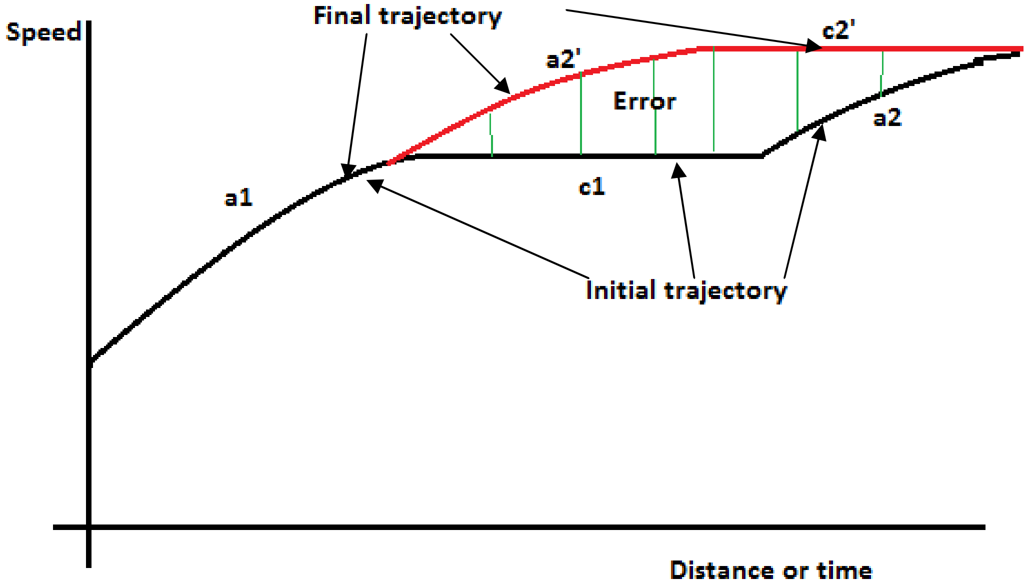


Figure 4.13: Replacement of riding modes.

```
>> m=smz(m)

m =

  Columns 1 through 16

     3     3     3     3     3     3     3     3     3     3     3     3     3     2     2     2
     2     1     1     1     1     3     2     2     2     2     1     1     1     1     1     2
     3     3     3     3     3     3     3     3     3     3     3     3     3     2     2     2
     3     3     3     3     3     3     3     2     2     2     3     3     3     3     3     3
     2     1     1     1     1     3     2     2     2     2     1     1     1     1     1     2

  Columns 17 through 20

     3     4     3     3
     2     1     2     2
     3     4     3     3
     2     2     3     3
     2     1     2     2
```

Figure 4.14: Smoothed out vector of riding modes

CHAPTER FIVE

RESULTS AND DISCUSSION

In this chapter, optimal speed trajectories for the railway sections from Ayat station to Megenagna station are presented. The solution for every section is made in both the upward and downward directions of ride. In addition to the trajectories, values of cost functions and constraints are also presented.

5.1 Results

In this section, a thorough analysis of the results found is made. The implication of the resulting trajectories in terms of their applicability in the actual scenario and the validity of constraints are also analyzed.

5.1.1 Trajectories for the Sections from Ayat Station to Megenagna Station

1. **Section EW1-EW2**

The trajectories for this section are shown in fig. 5.1 to fig 5.5. It can obviously be observed that the results are trade off solutions; as one of them is the best in minimizing the energy consumption of the train while the other improves the time to travel between stations. For the upward direction of motion, the fastest trajectory needs only 107 seconds of drive to complete the journey at the expense of 5.23 MJ of energy. The energy conserving trajectory for the same direction of motion consumes only 3.26 MJ of energy to complete the journey within 125 seconds. The downward direction of motion can be accomplished within just 118 seconds at the expense of the largest amount of energy for the section equal to 16.42 MJ. An energy conserving strategy for the same direction of motion needs an expenditure of 12.71 MJ of energy within 153 seconds of drive. The trajectory shown in figure 5.1 constitutes of cruising, coasting and cruising regimes between the initial motoring and the final braking modes. We can see that the coasting phase is a positive acceleration regime. This accounts for the negative track gradient on the corresponding sections. The values for the constraint functions are evaluated and are presented in table 5.1. We can see that the constraints for the speed limits, average acceleration and braking rates ($\bar{a}$ and $\bar{b}$), and the maximum allowable jerk (J) values are satisfied.

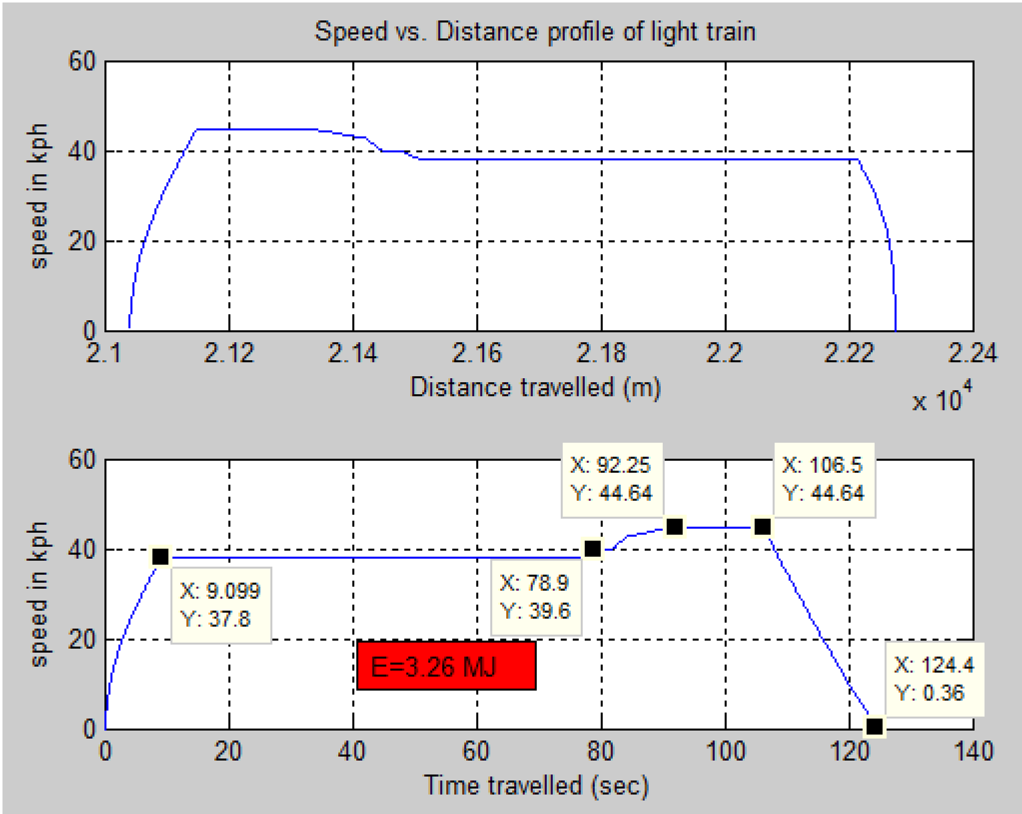


Figure 5.1: Train speed profile 1 for EW1-EW2 movement.

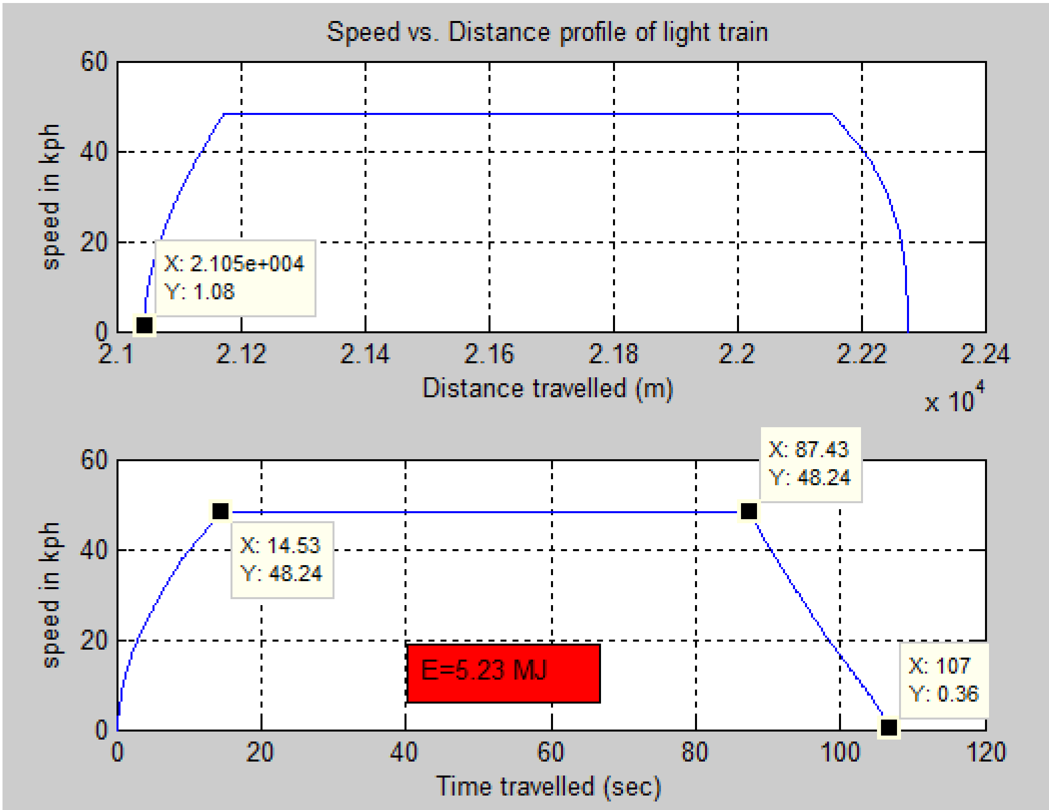


Figure 5.2: Train speed profile 2 for EW1-EW2 movement.

Section	Trajectory	ā	Б	$v_{crossing}$	J_{max}
EW1-EW2	1	3.98	2.46	38	0.52
	2	3.32	2.47	49	0.58
EW2-EW1	1	3.39	3.65	34	0.6
	2	3.29	3.64	34	0.58
	3	2.64	3.45	30	0.68

Table 5.1: Constraint evaluation for section EW1-EW2

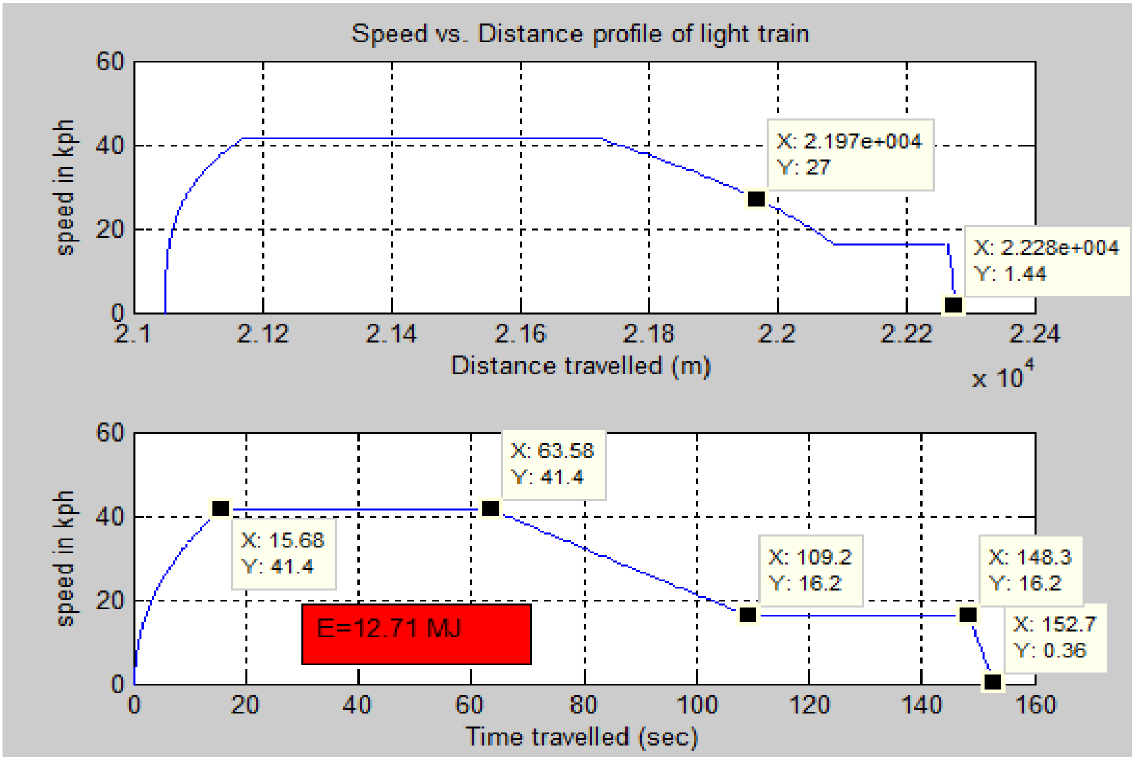


Figure 5.3: Train speed profile 1 for EW2-EW1 movement.

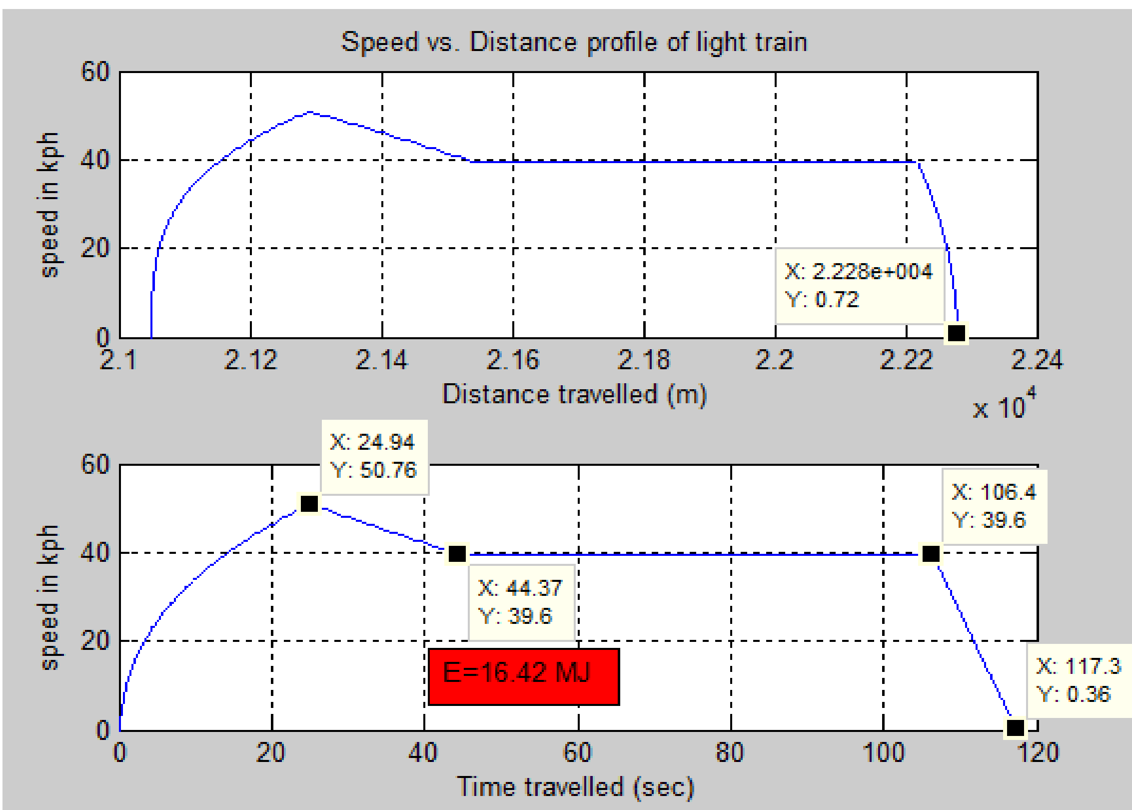


Figure 5.4: Train speed profile 2 for EW2-EW1 movement.

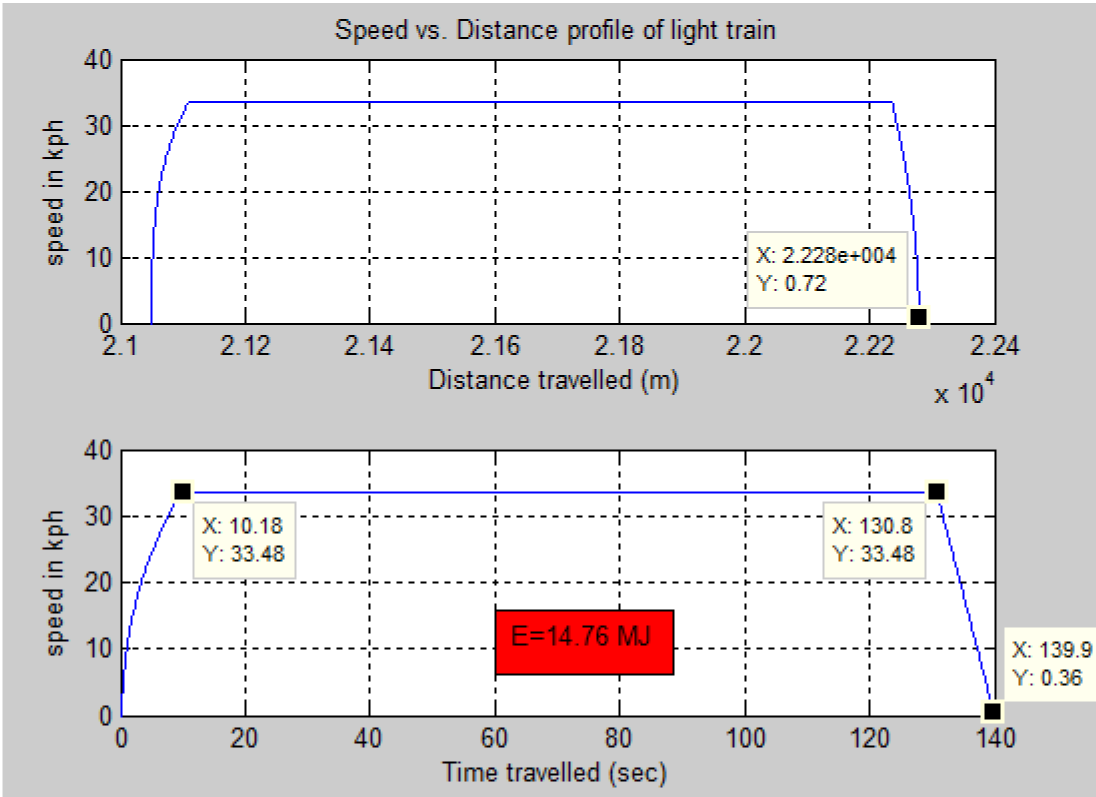


Figure 5.5: Train speed profile 3 for EW2-EW1 movement.

2. Section EW2-EW3

For the upward direction of motion in this section, the fastest trajectory needs only 100 seconds of drive to complete the journey, while the most energy conserving one needs 3.05 MJ of energy. The downward direction of motion can be accomplished within just 95 seconds with the fastest trajectory, while the minimum amount of energy needed to complete the journey is 3.23 MJ.

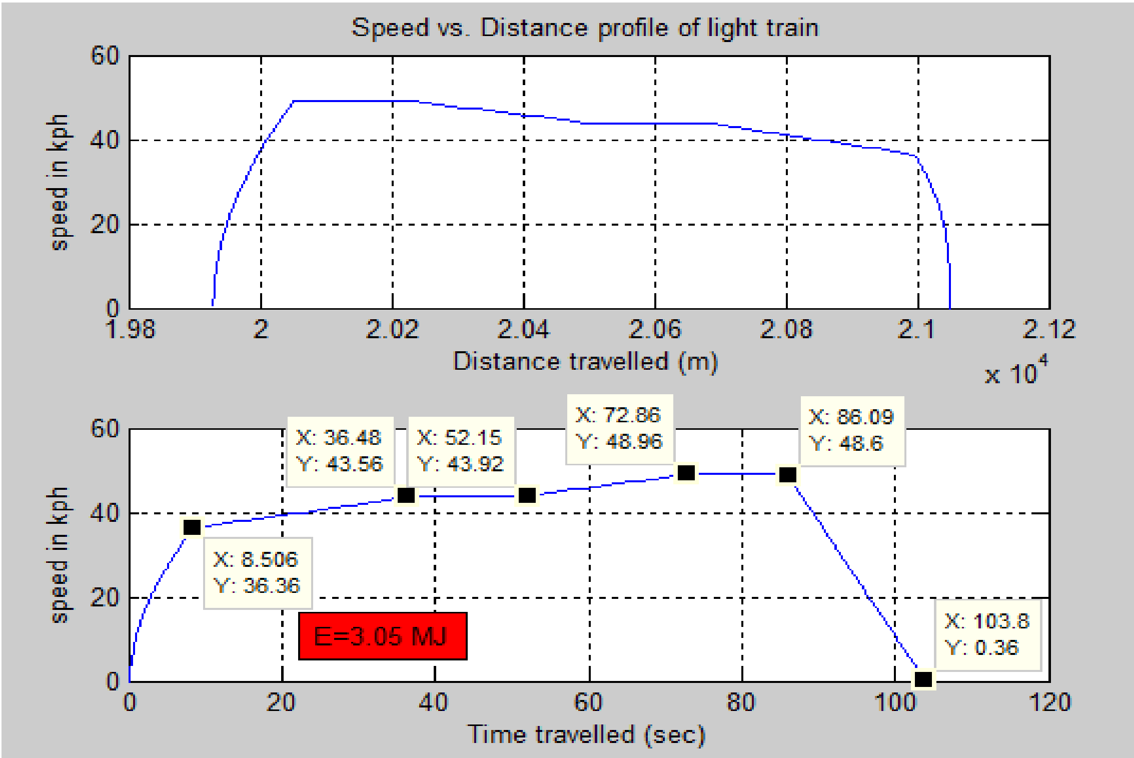


Figure 5.6: Train speed profile 1 for EW2-EW3 movement.

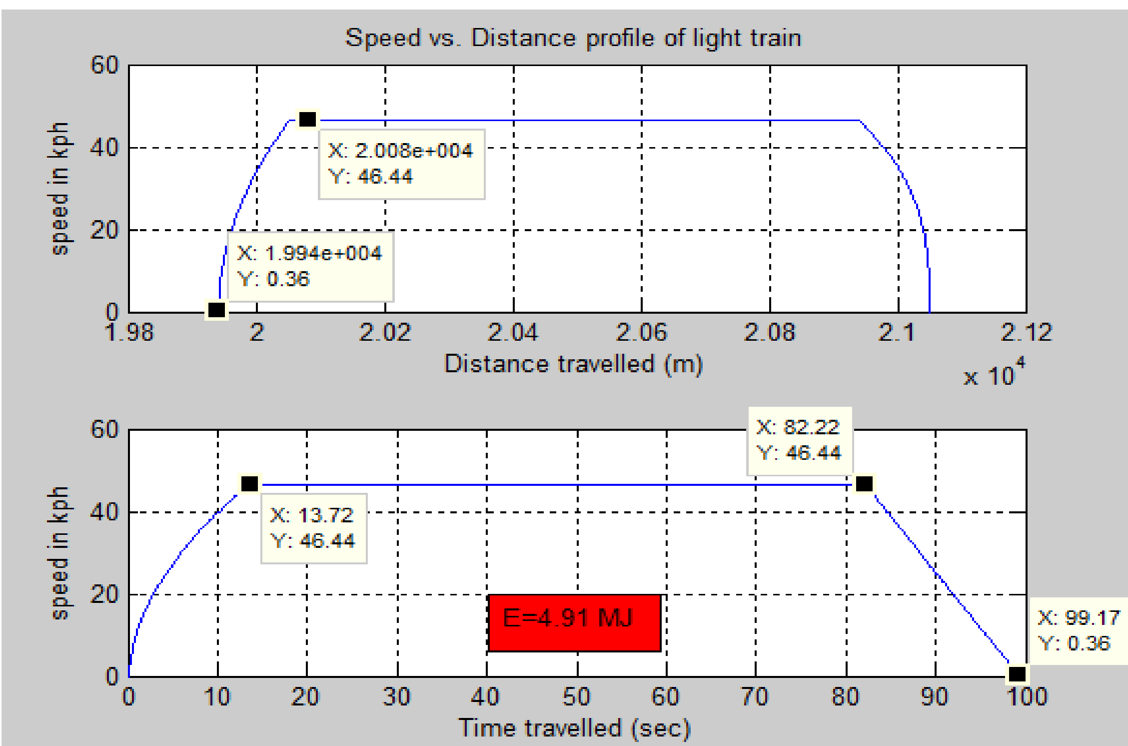


Figure 5.7: Train speed profile 2 for EW2-EW3 movement.

Section	Trajectory	ā	Б	$v_{crossing}$	J_{max}
EW2-EW3	1	4.27	2.7	49	0.48
	2	3.38	2.73	47	0.51
EW3-EW2	1	3.9	2.94	36	0.38
	2	3.21	3.37	45	0.44
	3	3.26	2.97	45	0.56

Table 5.2 Constraint evaluation for section EW2-EW3

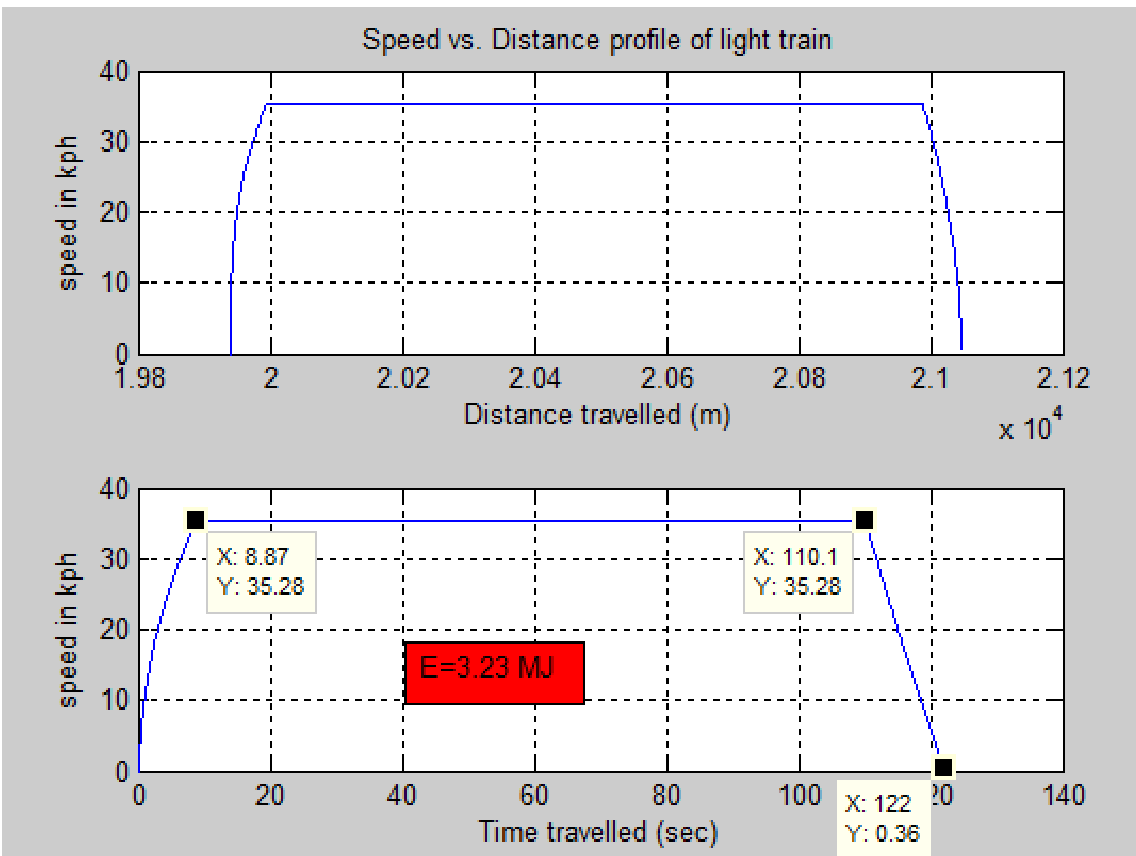


Figure 5.8: Train speed profile 1 for EW3-EW2 movement.

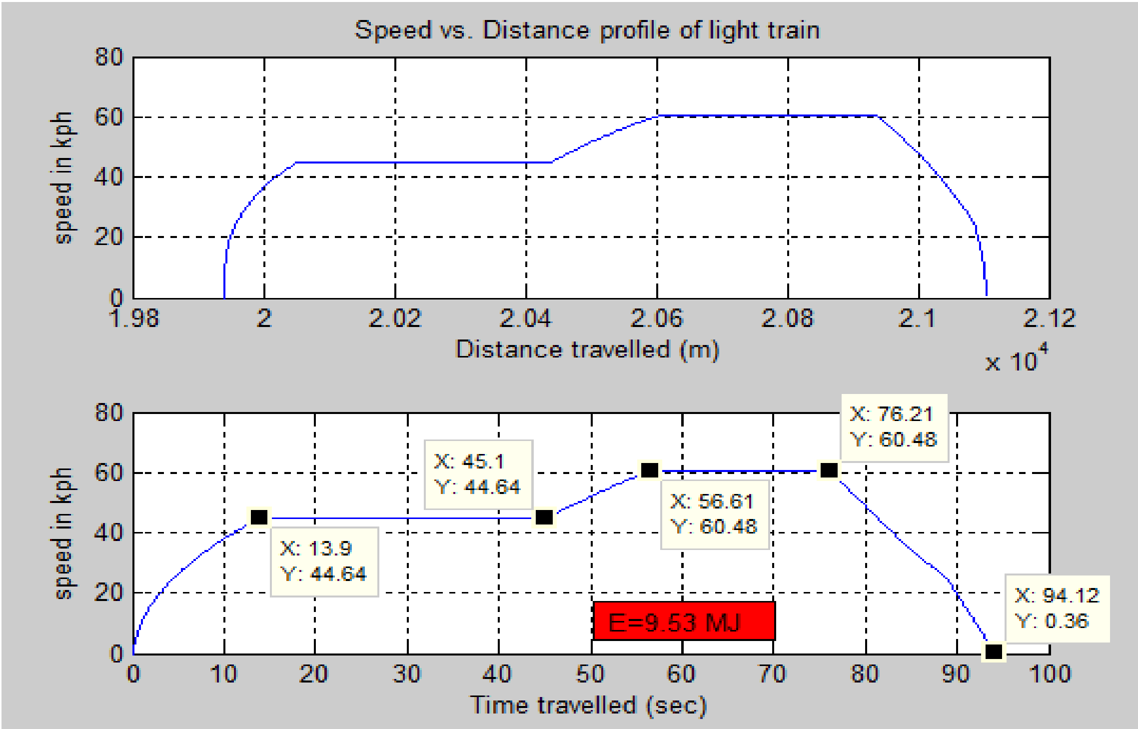


Figure 5.9: Train speed profile 2 for EW3-EW2 movement.

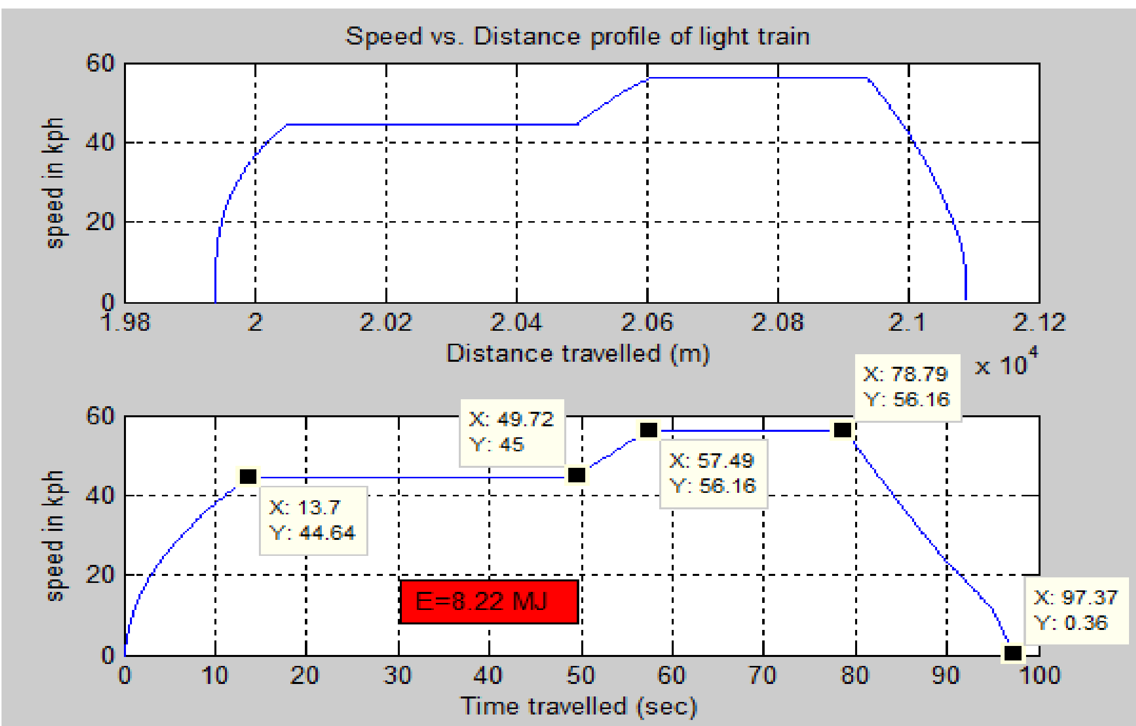


Figure 5.10: Train speed profile 3 for EW3-EW2 movement.

3. Section EW3-EW4

The upward direction of motion in this section can be completed within just 79 seconds by utilizing the fastest trajectory, while the slowest trajectory needs an 8 MJ of energy. The downward direction of motion can be completed within just 70 seconds at the expense of the largest amount of energy for the section equal to 8.31 MJ. An energy conserving strategy for the same direction of motion needs an expenditure of 2.5 MJ of energy within 90 seconds of drive.

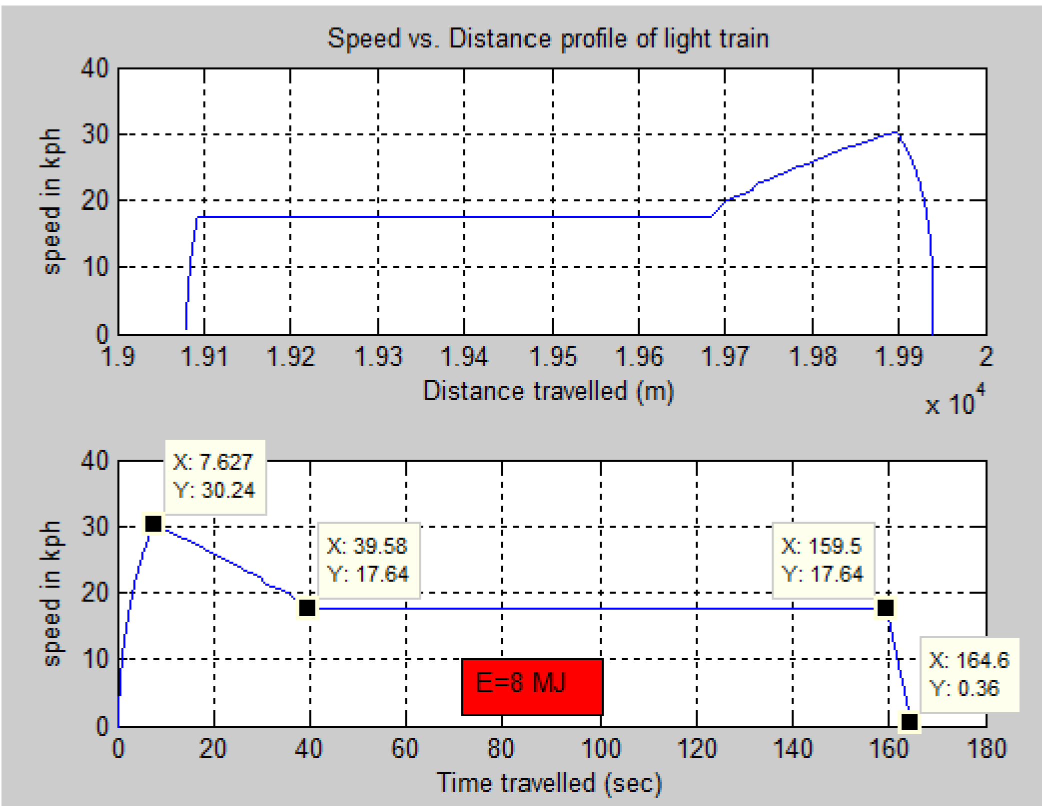


Figure 5.11: Train speed profile 1 for EW3-EW4 movement.

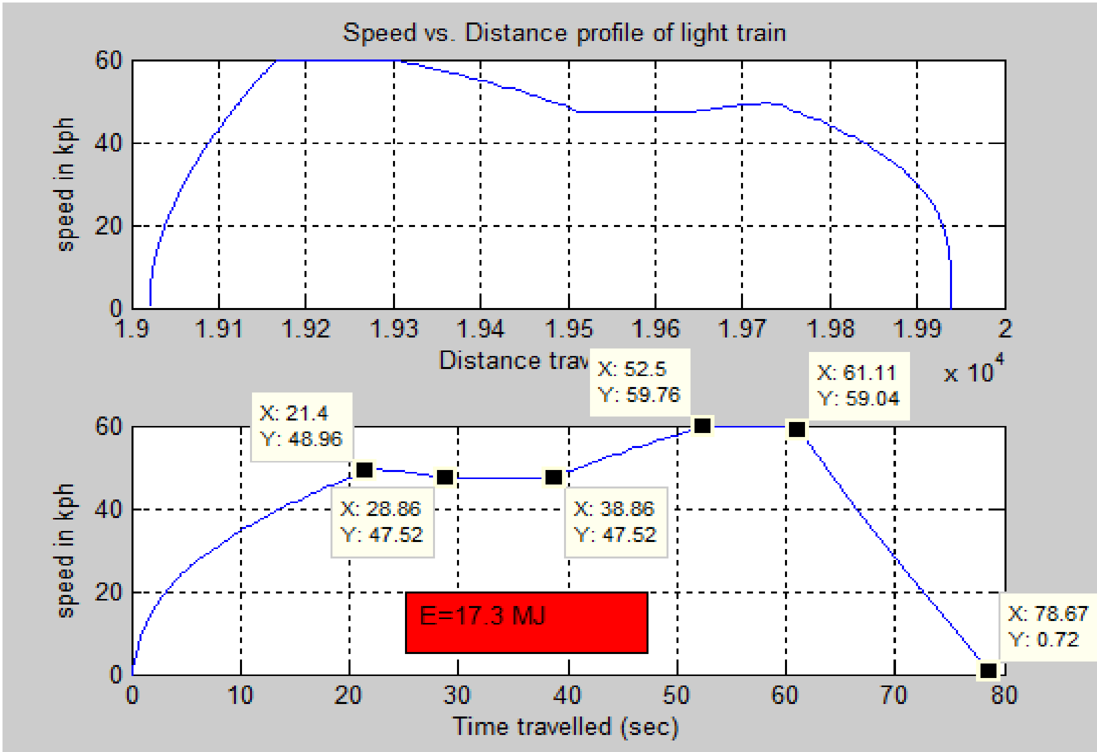


Figure 5.12: Train speed profile 2 for EW3-EW4 movement.

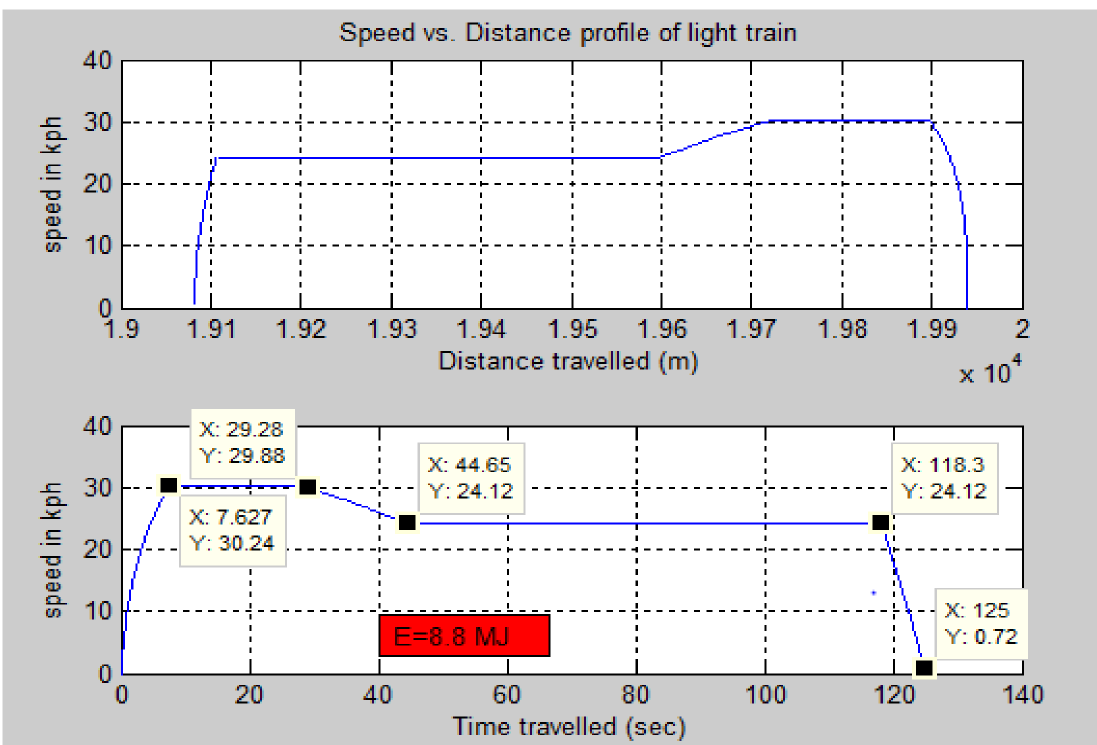


Figure 5.13: Train speed profile 3 for EW3-EW4 movement.

Section	Trajectory	ā	Б	$v_{crossing}$	J_{max}
EW3-EW4	1	3.95	3.46	-	0.65
	2	2.29	3.35	-	0.5
	3	3.96	3.6	-	0.45
EW4-EW3	1	3.6	3.5	-	0.94
	2	4.9	3.24	-	0.64
	3	5	3.57	-	0.84

Table 5.3: Constraint evaluation for section EW3-EW4

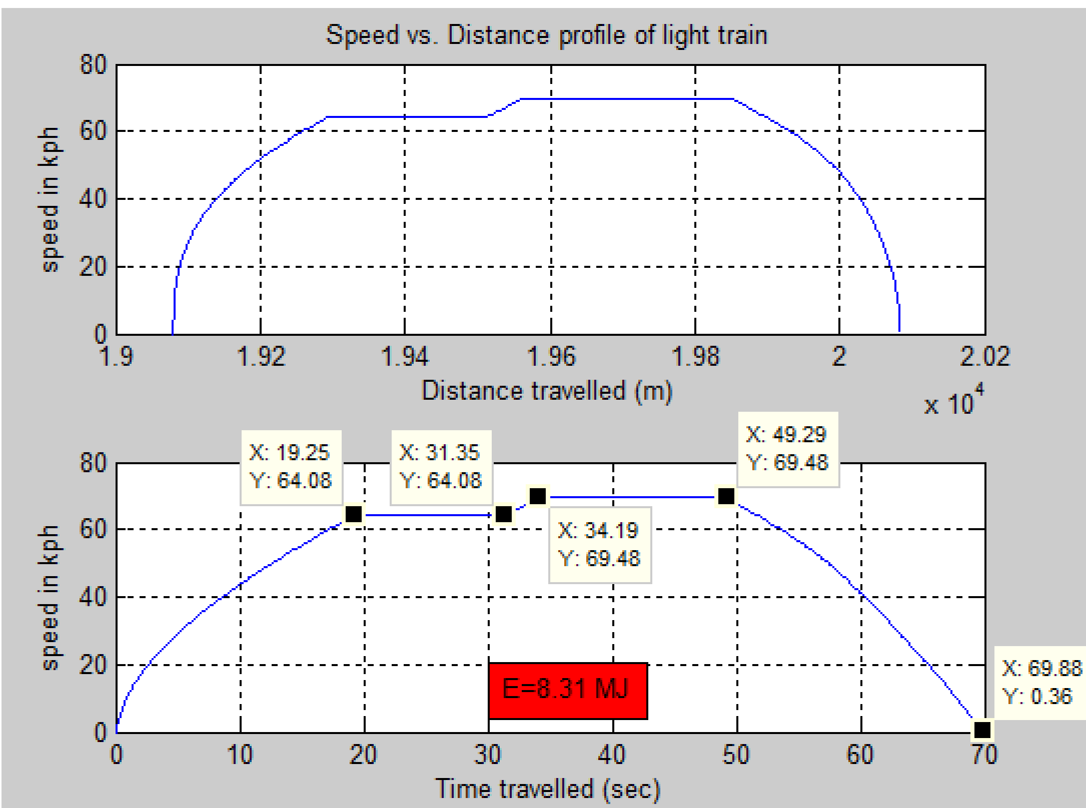


Figure 5.14: Train speed profile 1 for EW4-EW3 movement.

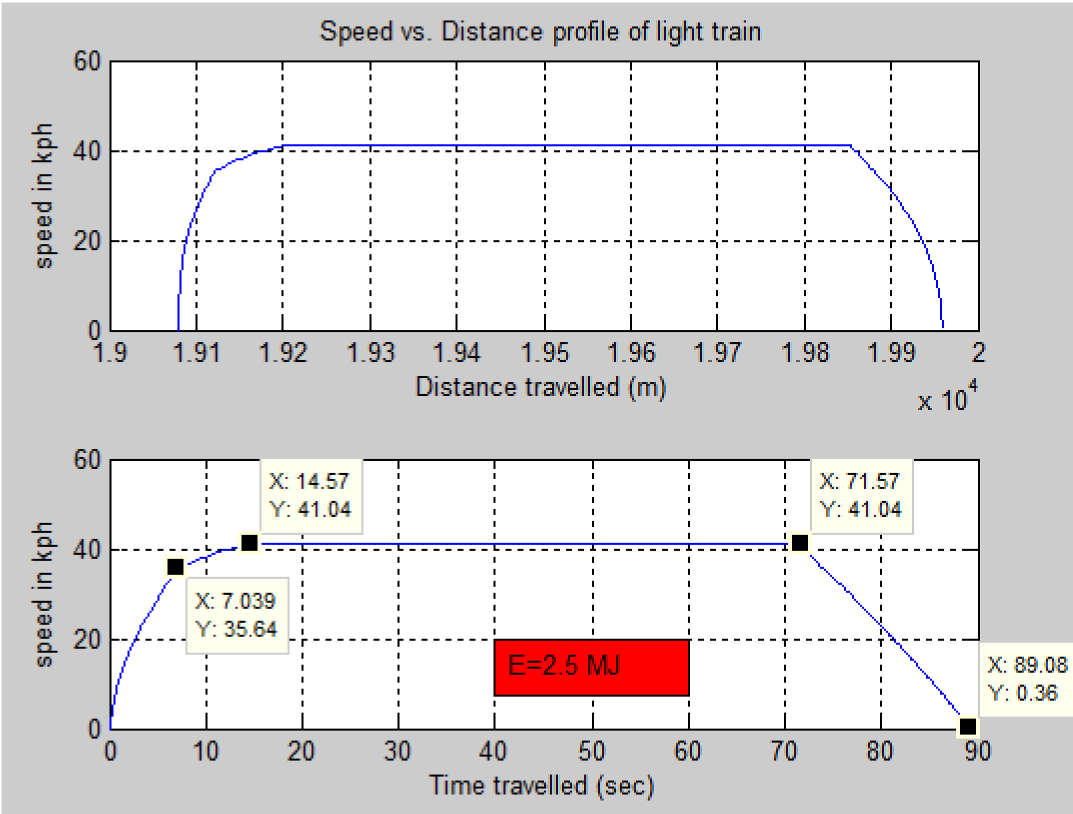


Figure 5.15: Train speed profile 2 for EW4-EW3 movement.

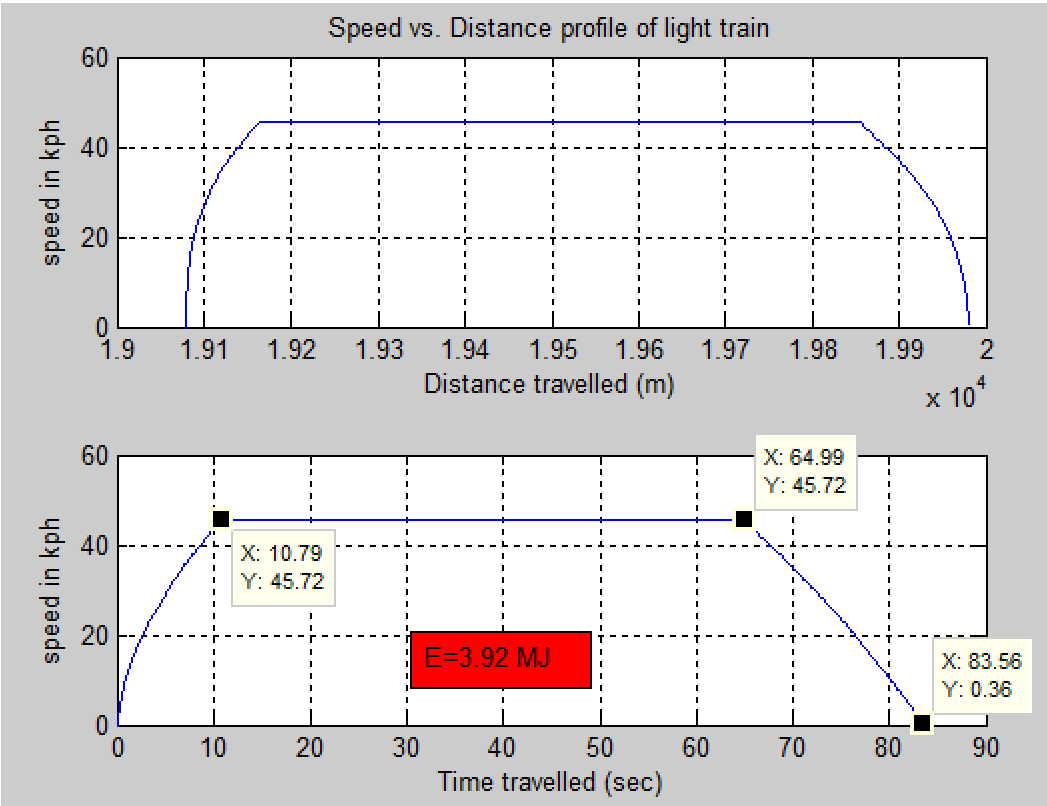


Figure 5.16: Train speed profile 3 for EW4-EW3 movement.

4. Section EW4-EW5

For the upward direction of motion in this section, the fastest trajectory needs only 84 seconds to complete the journey, while an energy conserving strategy needs an expense of 19 MJ. The downward direction of motion can be completed within just 84 seconds at the expense of 4.16 MJ of energy. An energy conserving strategy for the same direction of motion needs an expenditure of just 2.59 MJ of energy within 94 seconds of drive.

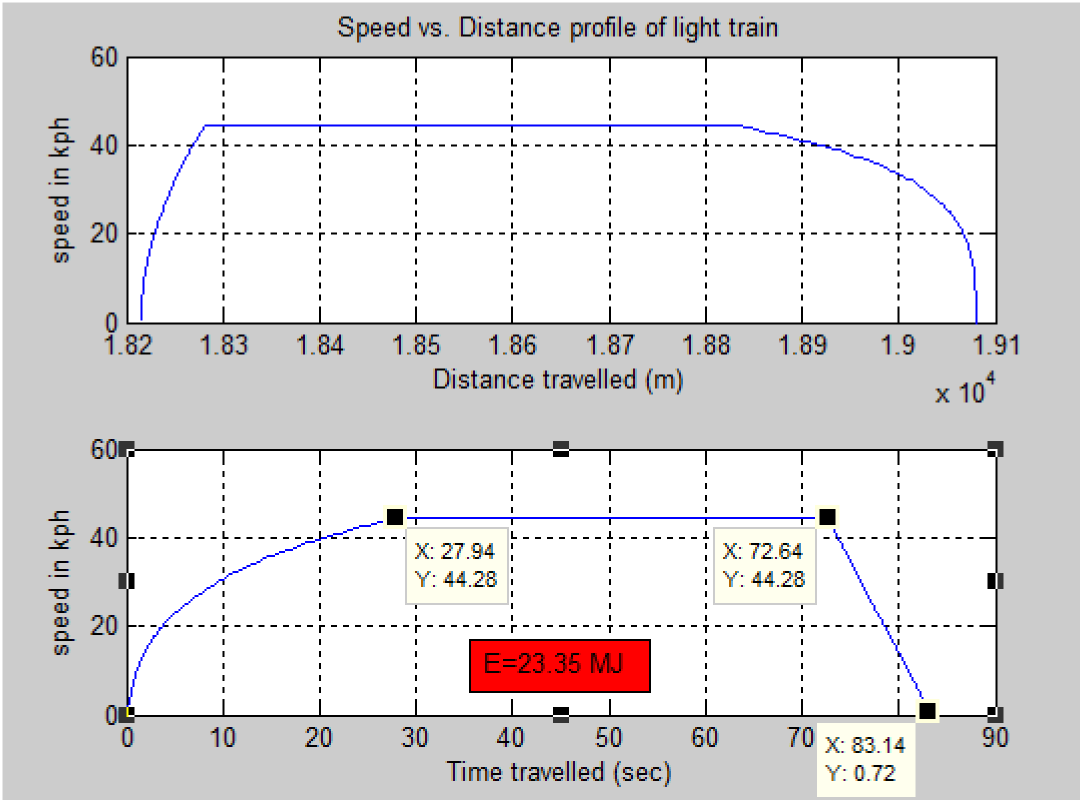


Figure 5.17: Train speed profile 1 for EW4-EW5 movement.

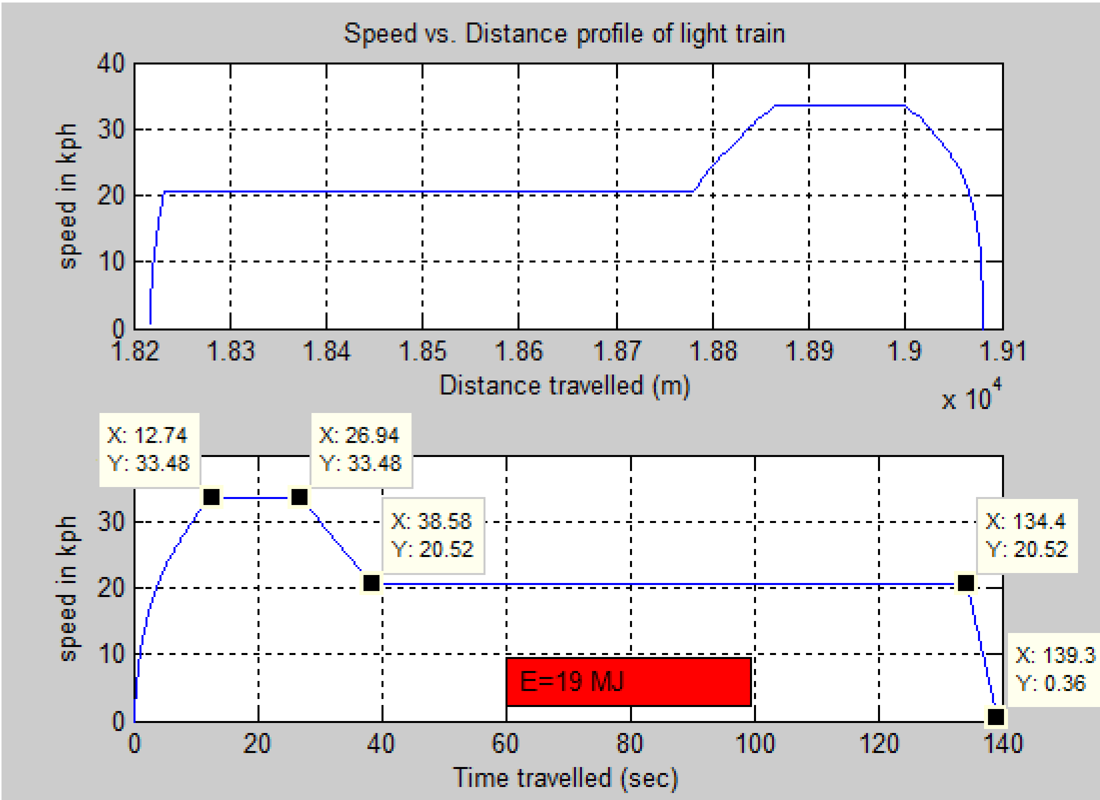


Figure 5.18: Train speed profile 2 for EW4-EW5 movement.

Section	Trajectory	ā	Б	$V_{crossing}$	J_{max}
EW4-EW5	1	1.58	1.87	38	0.82
	2	2.63	3.66	34	0.72
EW5-EW4	1	4.32	2.8	42	0.97
	2	3.66	2.78	43	0.59

Table 5.4: Constraint evaluation for section EW4-EW5

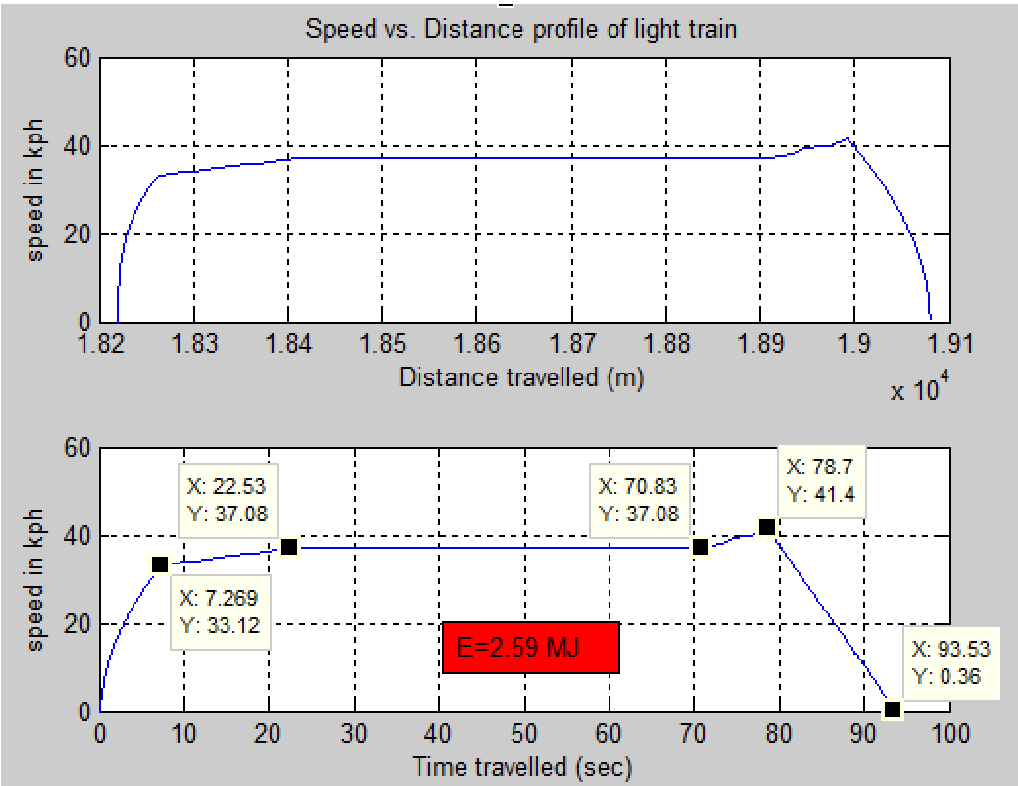


Figure 5.19: Train speed profile 1 for EW5-EW4 movement.

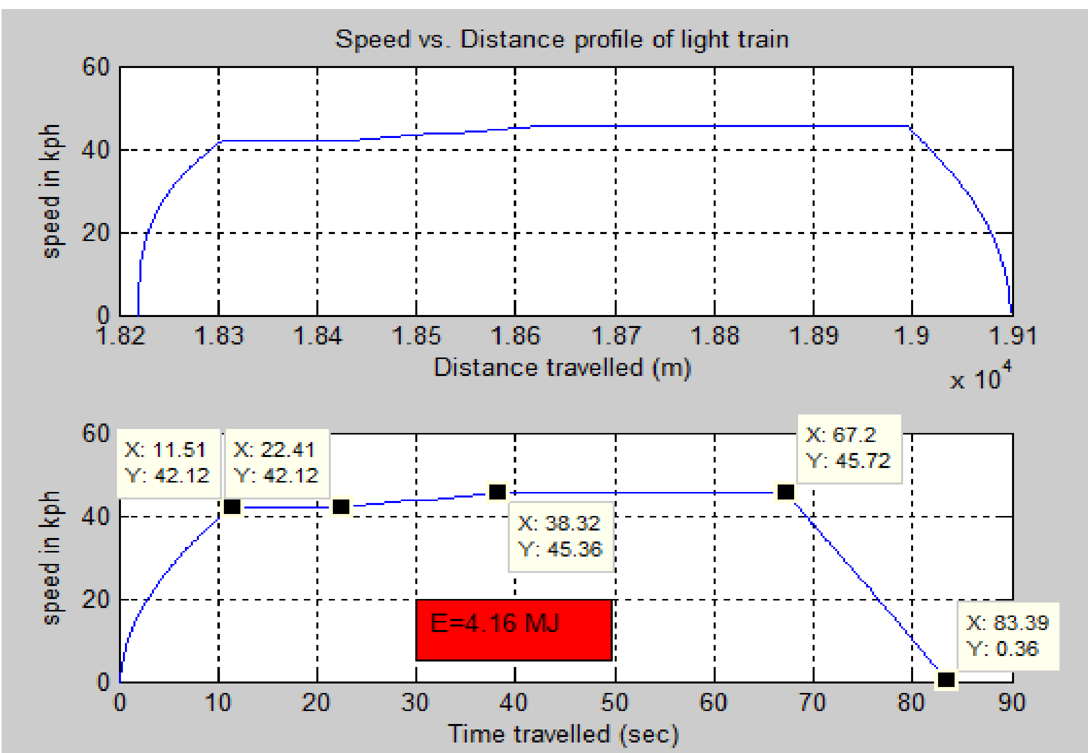


Figure 5.20: Train speed profile 2 for EW5-EW4 movement.

5. Section EW5-EW6

For the upward direction of motion in this section, the fastest trajectory needs only 81 seconds of drive to complete the journey at the expense of 11.2 MJ of energy. The energy conserving trajectory for the same direction of motion consumes only 8 MJ of energy to complete the journey within 117 seconds. The downward direction of motion can be accomplished within just 114 seconds while an energy conserving strategy needs 5 MJ of energy within 121 seconds of drive.

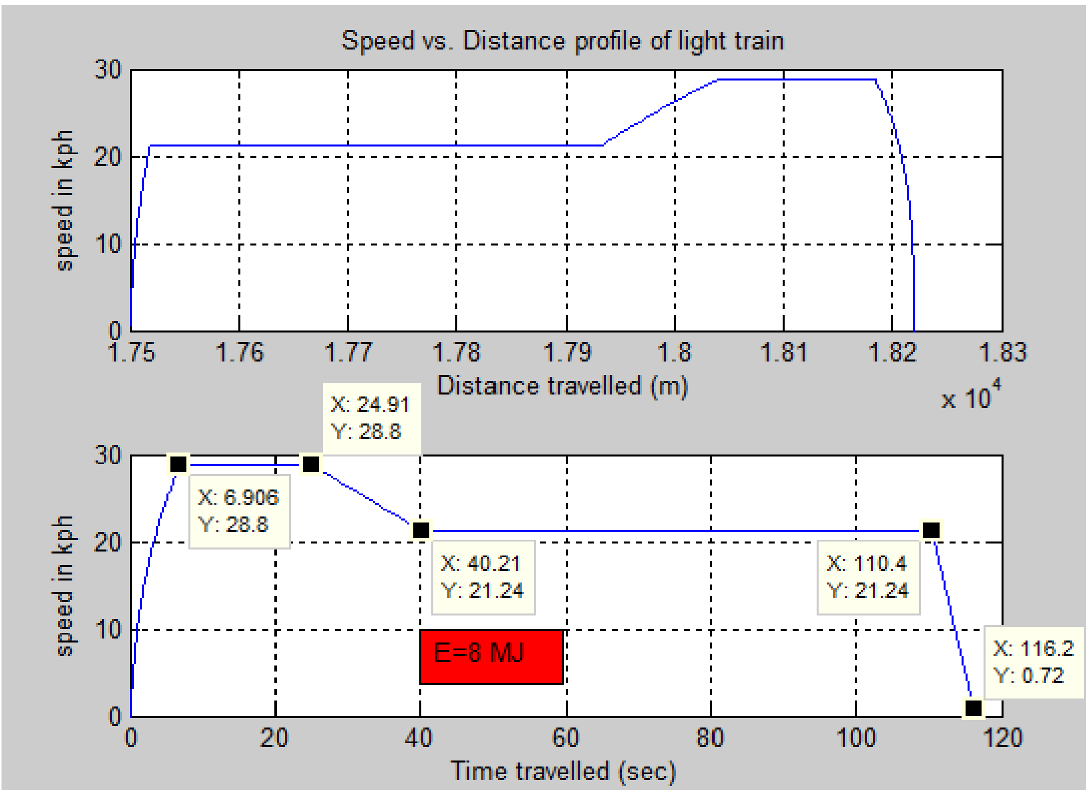


Figure 5.21: Train speed profile 1 for EW5-EW6 movement.

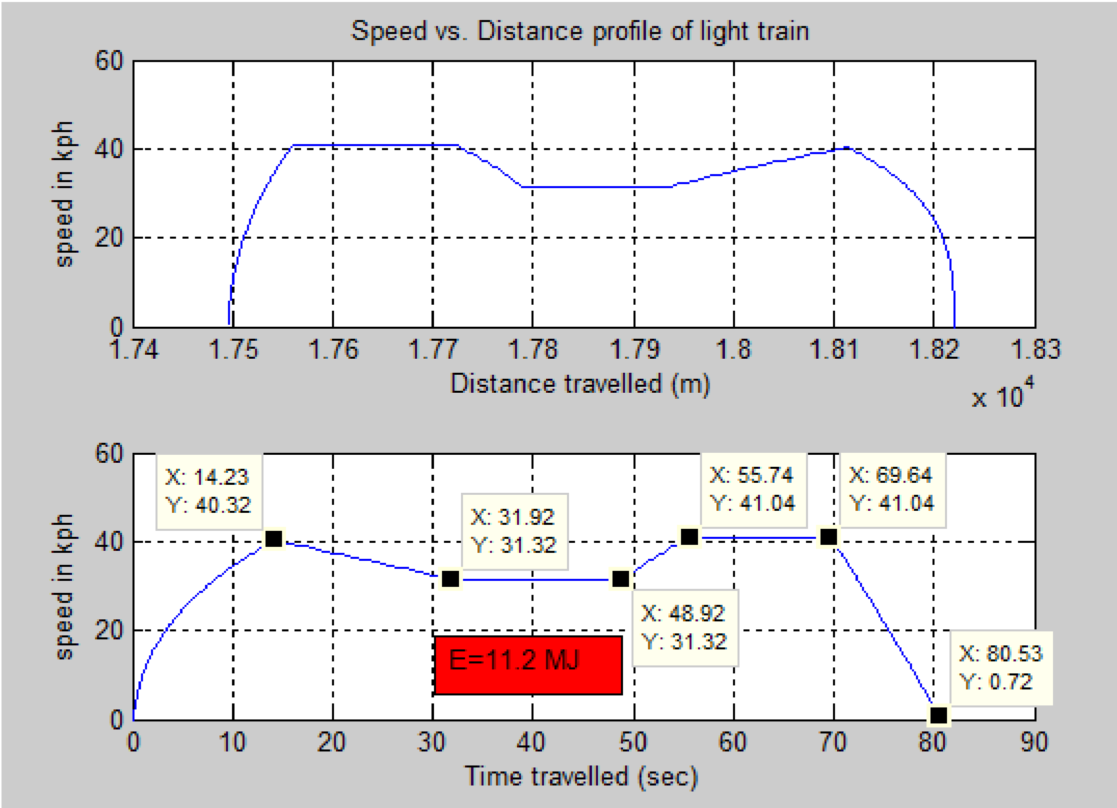


Figure 5.22: Train speed profile 2 for EW5-EW6 movement.

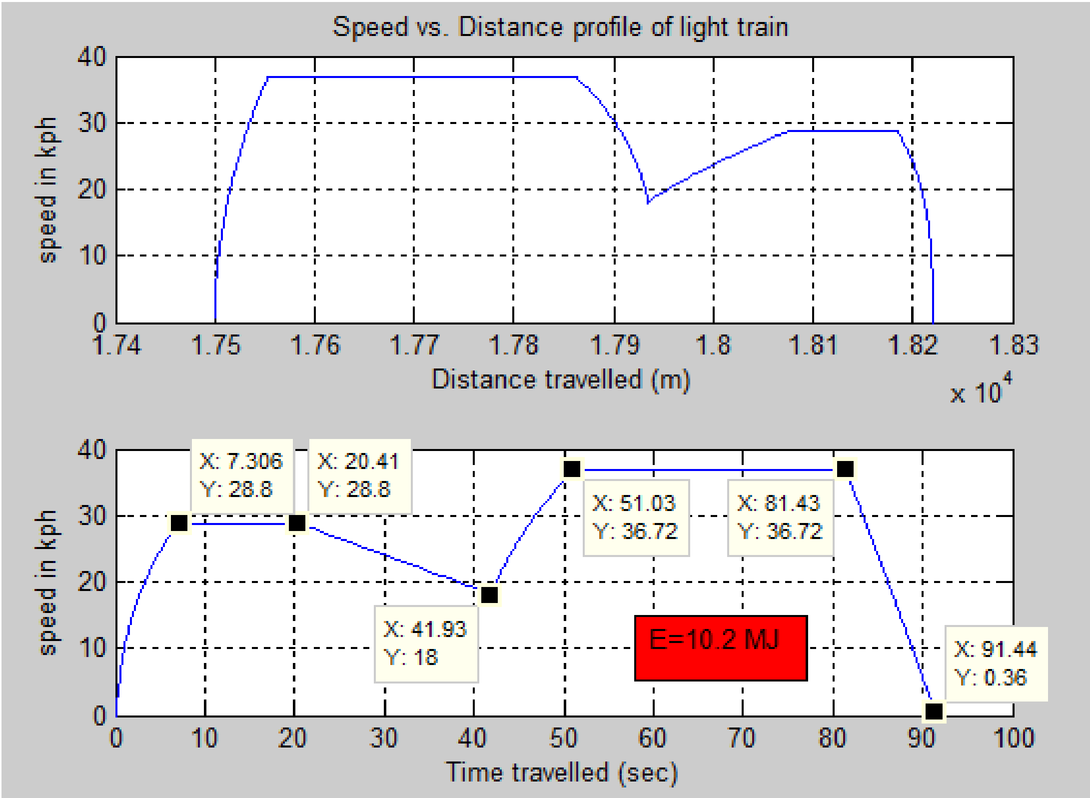


Figure 5.23: Train speed profile 3 for EW5-EW6 movement.

Section	Trajectory	ā	Б	$v_{crossing}$	J_{max}
EW5-EW6	1	4.17	3.66	-	0.78
	2	2.83	3.76	-	0.77
	3	3.9	3.66	-	0.9
EW6-EW5	1	4.37	3.36	-	0.65
	2	4.12	3.32	-	0.65

Table 5.5: Constraint evaluation for section EW5-EW6

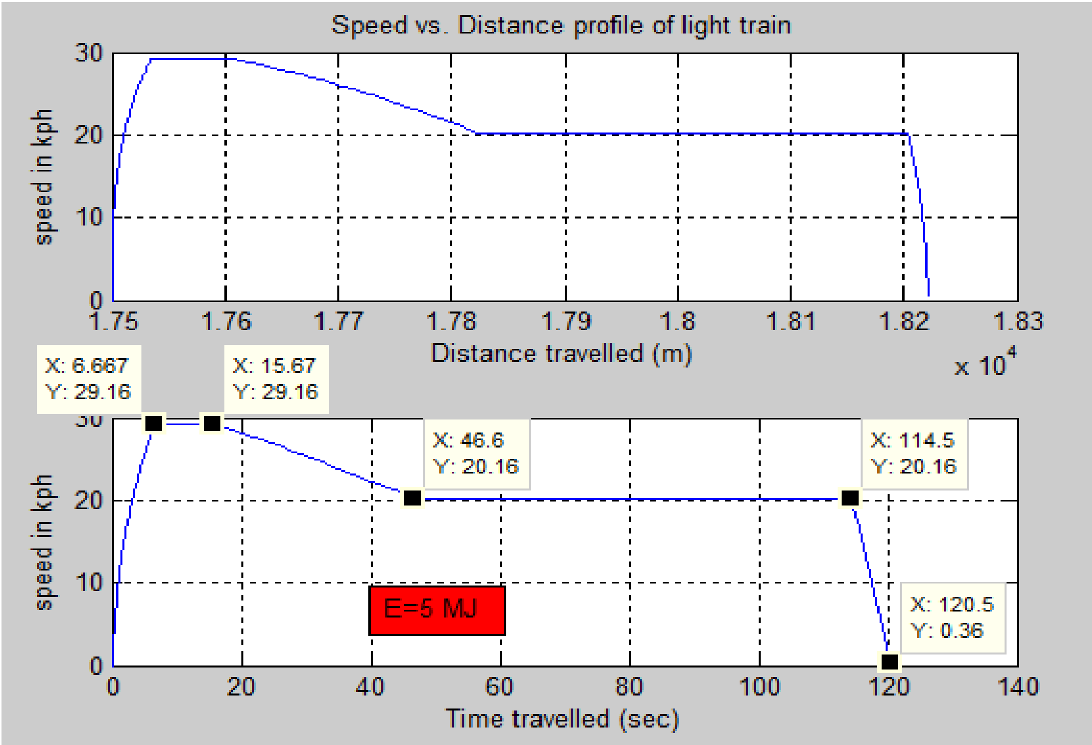


Figure 5.24: Train speed profile 1 for EW6-EW5 movement.

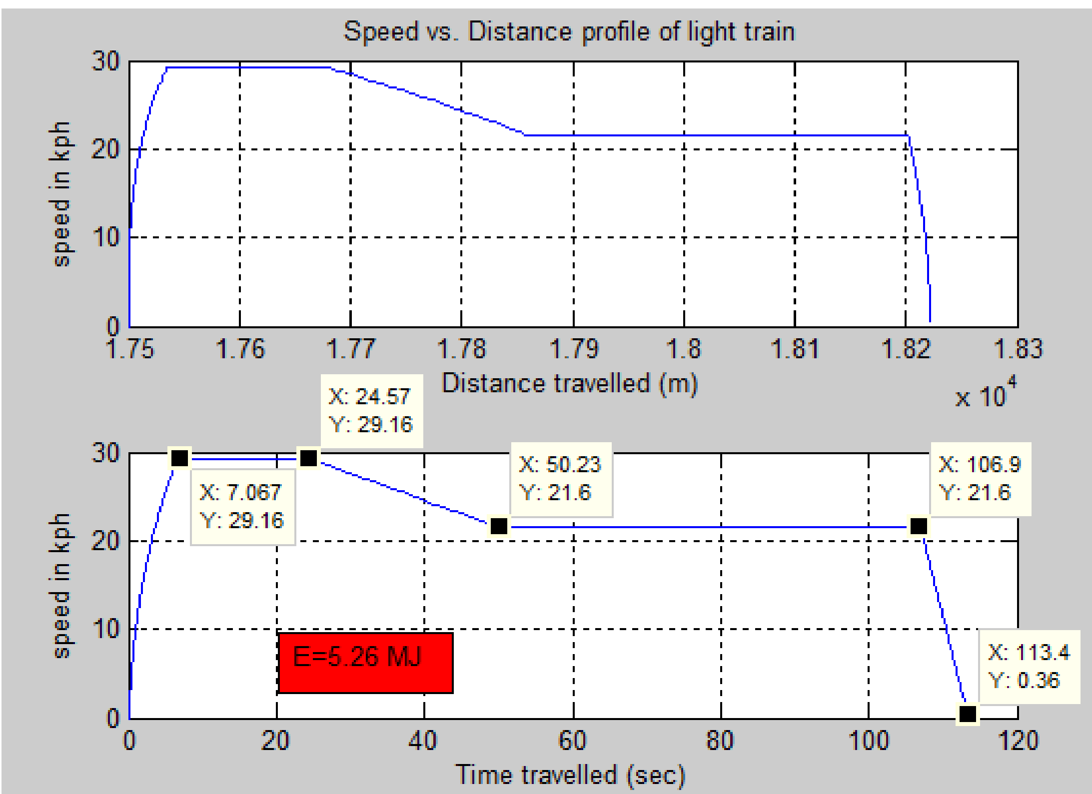


Figure 5.25: Train speed profile 2 for EW6-EW5 movement.

6. Section EW6-EW7

The fastest trajectory in the upward direction needs only 96 seconds of drive to complete the journey at the expense of 6.74 MJ of energy. An energy conserving strategy for the same direction consumes only 2.93 MJ of energy to complete the journey within 119 seconds. The downward direction of motion can be completed within just 92 seconds at the expense of the largest amount of energy for the section equal to 13.74 MJ. An energy conserving strategy for the same direction of motion needs an expense of 11.2 MJ of energy within 117 seconds of drive.

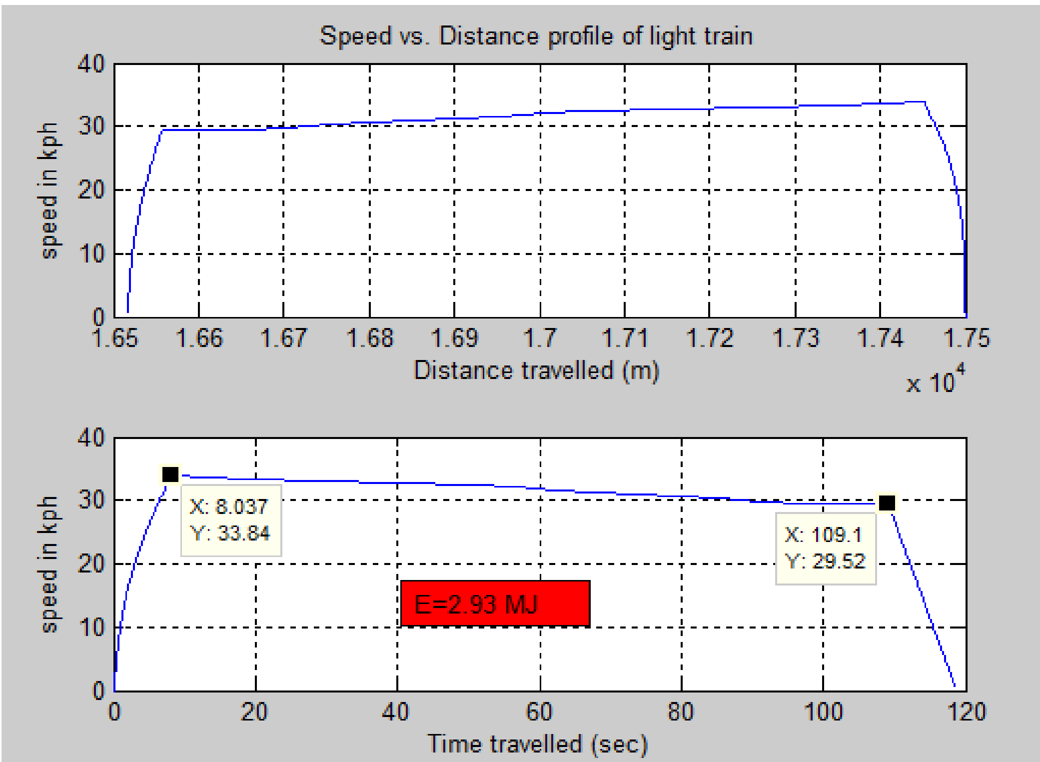


Figure 5.26: Train speed profile 1 for EW6-EW7 movement.

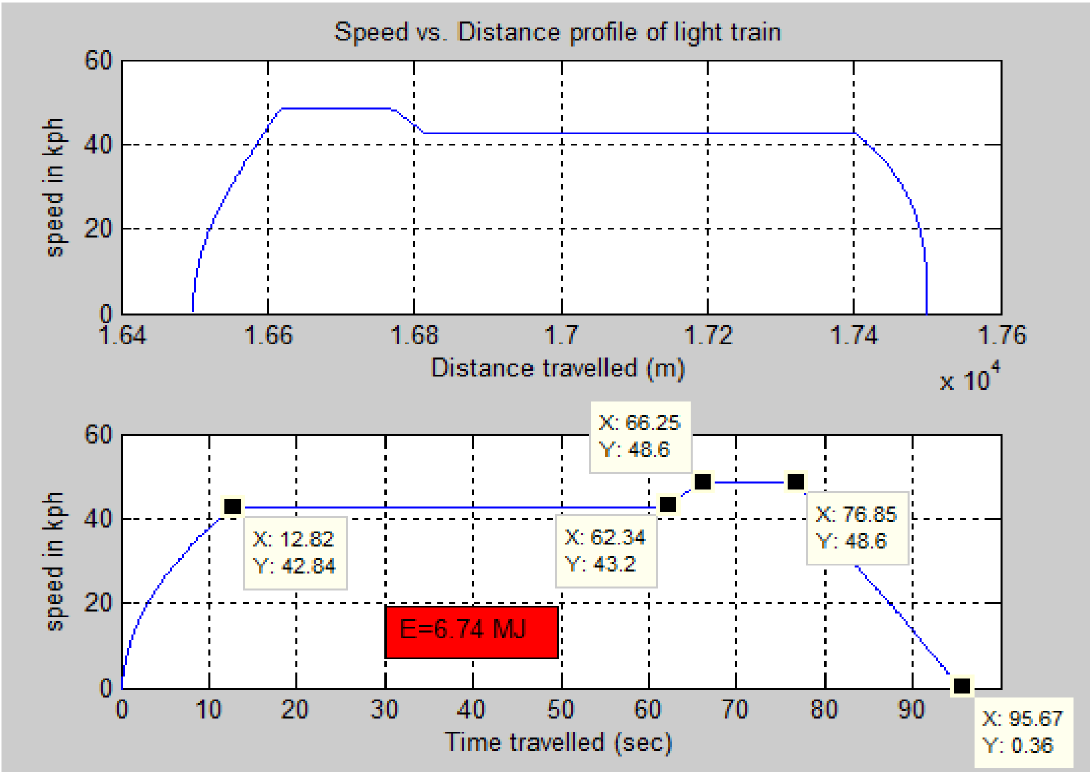


Figure 5.27: Train speed profile 2 for EW6-EW7 movement.

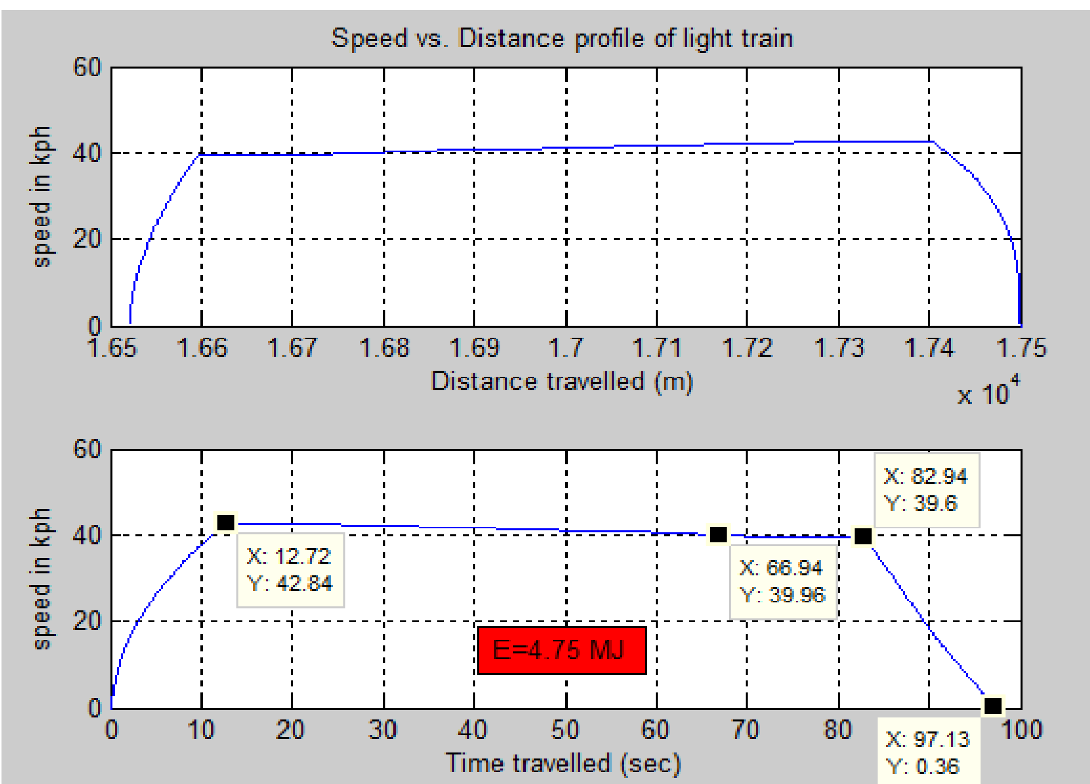


Figure 5.28: Train speed profile 3 for EW6-EW7 movement.

Section	Trajectory	ā	Б	$V_{crossing}$ (kph)	J_{max}
EW6-EW7	1	4.12	3.31	34	0.71
	2	3.34	2.58	43	0.8
	3	3.37	2.79	43	0.89
EW7-EW6	1	4.16	3.76	26	0.64
	2	2.47	3.76	26	0.64

Table 5.6: Constraint evaluation for section EW6-EW7

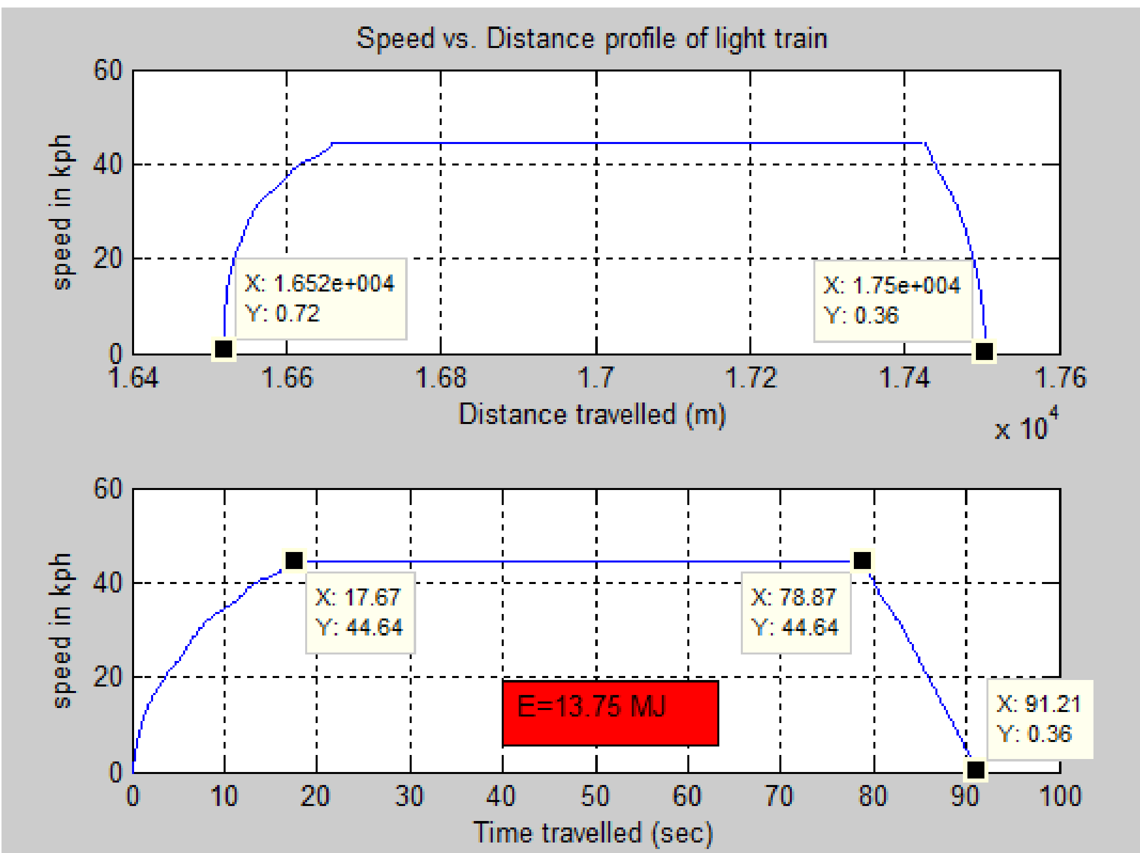


Figure 5.29: Train speed profile 1 for EW7-EW6 movement.

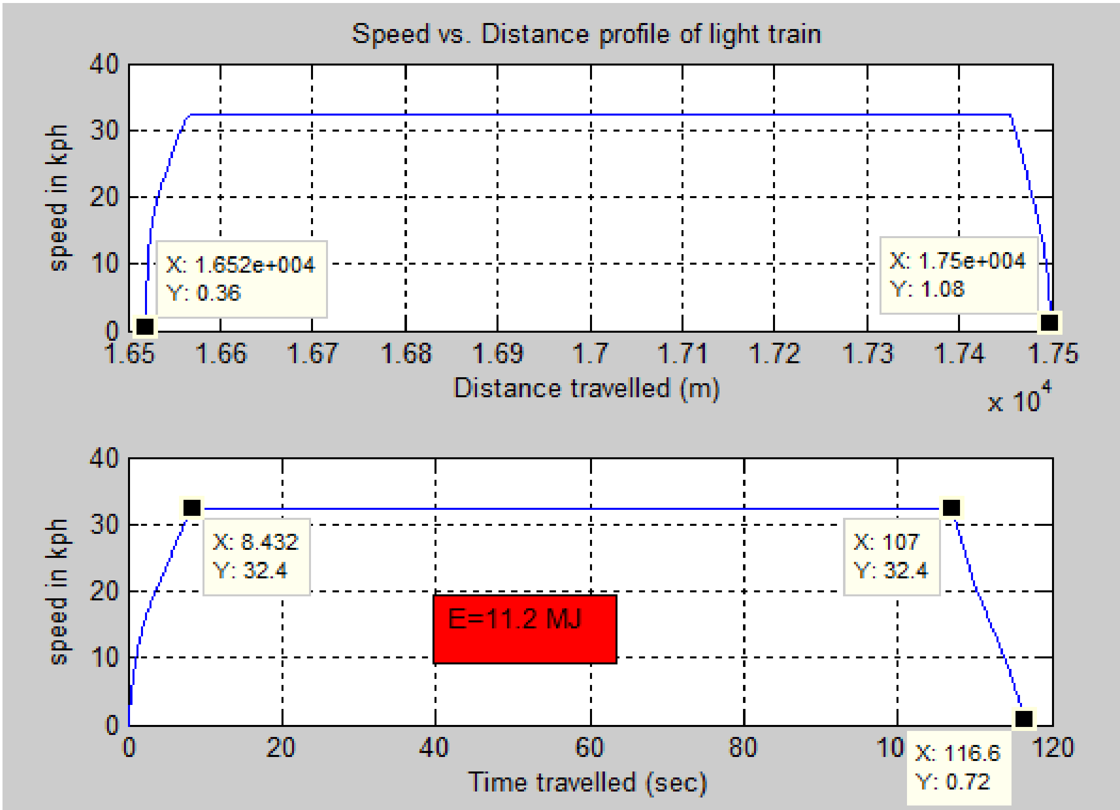


Figure 5.30: Train speed profile 2 for EW7-EW6 movement.

7. Section EW7-EW8

For the upward direction of motion in this section, the fastest trajectory needs only 105 seconds of drive to complete the journey at the expense of 11.45 MJ of energy. The energy conserving strategy for the same direction of motion consumes only 9.8 MJ of energy to complete the journey within 125 seconds. The downward direction of motion can be completed within just 99 seconds at the expense of the largest amount of energy for the section equal to 10.34 MJ. An energy conserving strategy for the same direction of motion needs an expenditure of 6.87 MJ of energy within 141 seconds of drive.

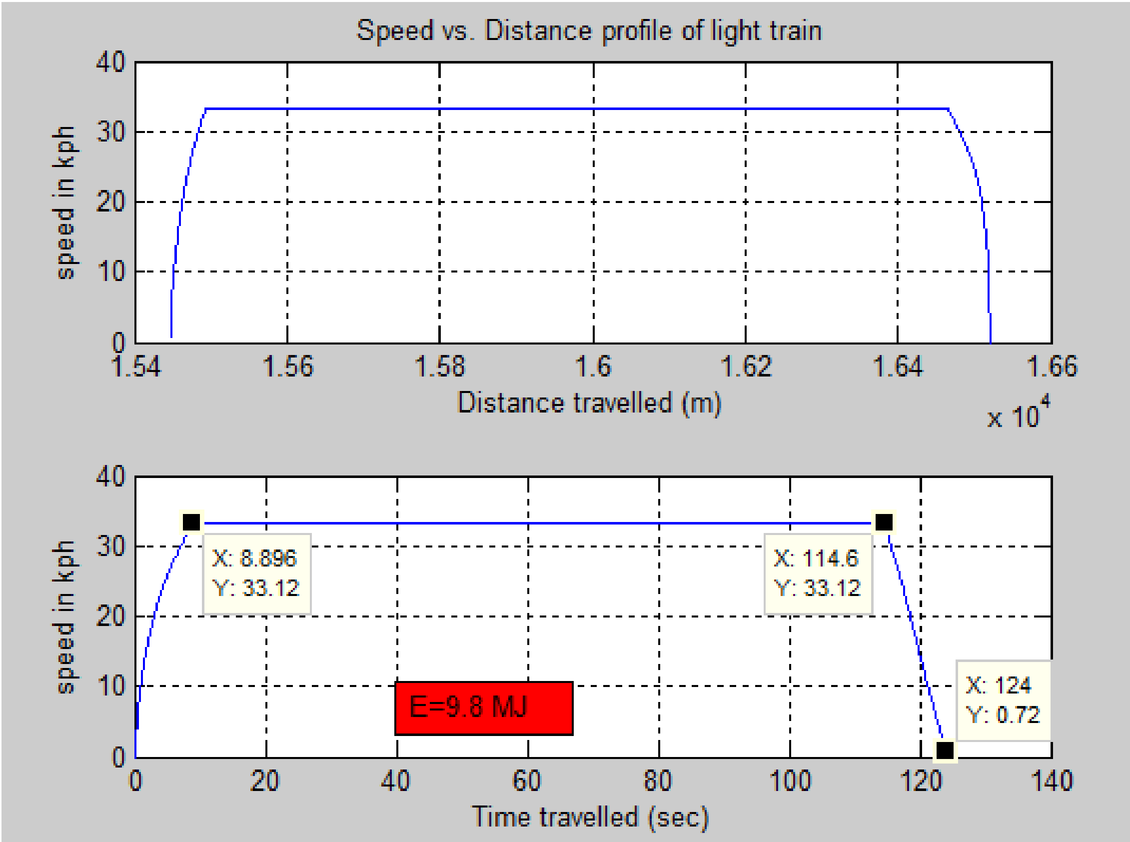


Figure 5.31: Train speed profile 1 for EW7-EW8 movement.

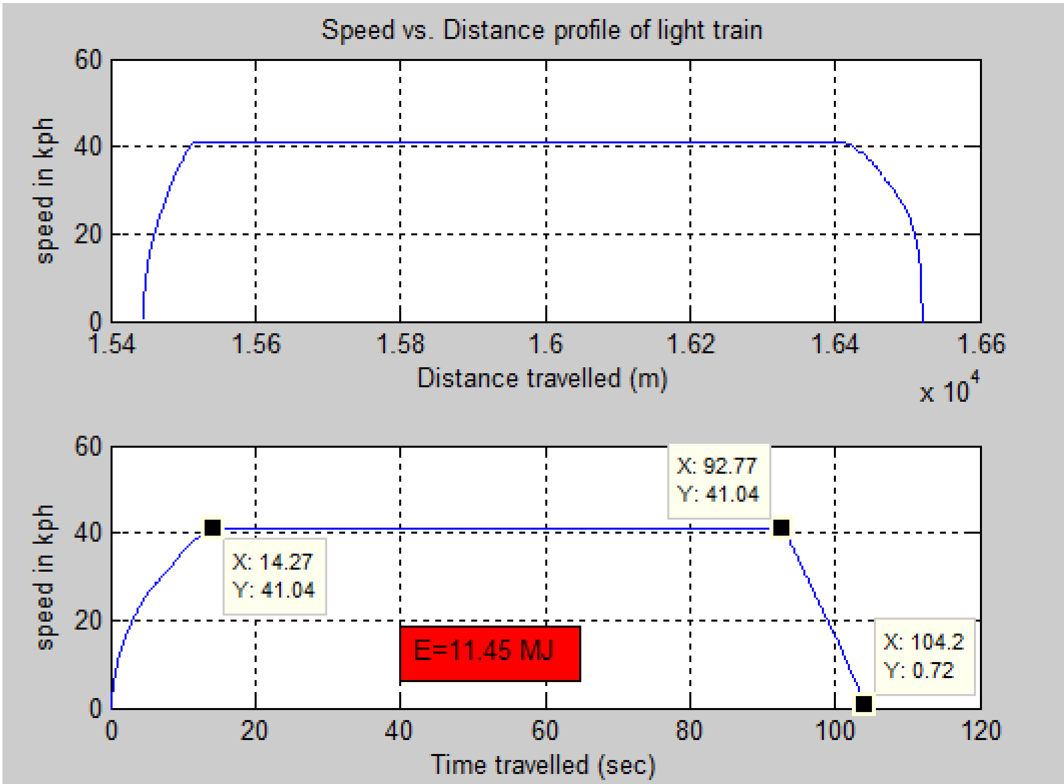


Figure 5.32: Train speed profile 2 for EW7-EW8 movement.

Section	Trajectory	ā	Б	$v_{crossing}$	J_{max}
EW7-EW8	1	3.72	3.52	34	0.41
	2	2.88	3.59	42	0.48
EW8-EW7	1	3.68	3.5	34	0.74
	2	2.62	3.6	48	0.83

Table 5.7: Constraint evaluation for section EW7-EW8

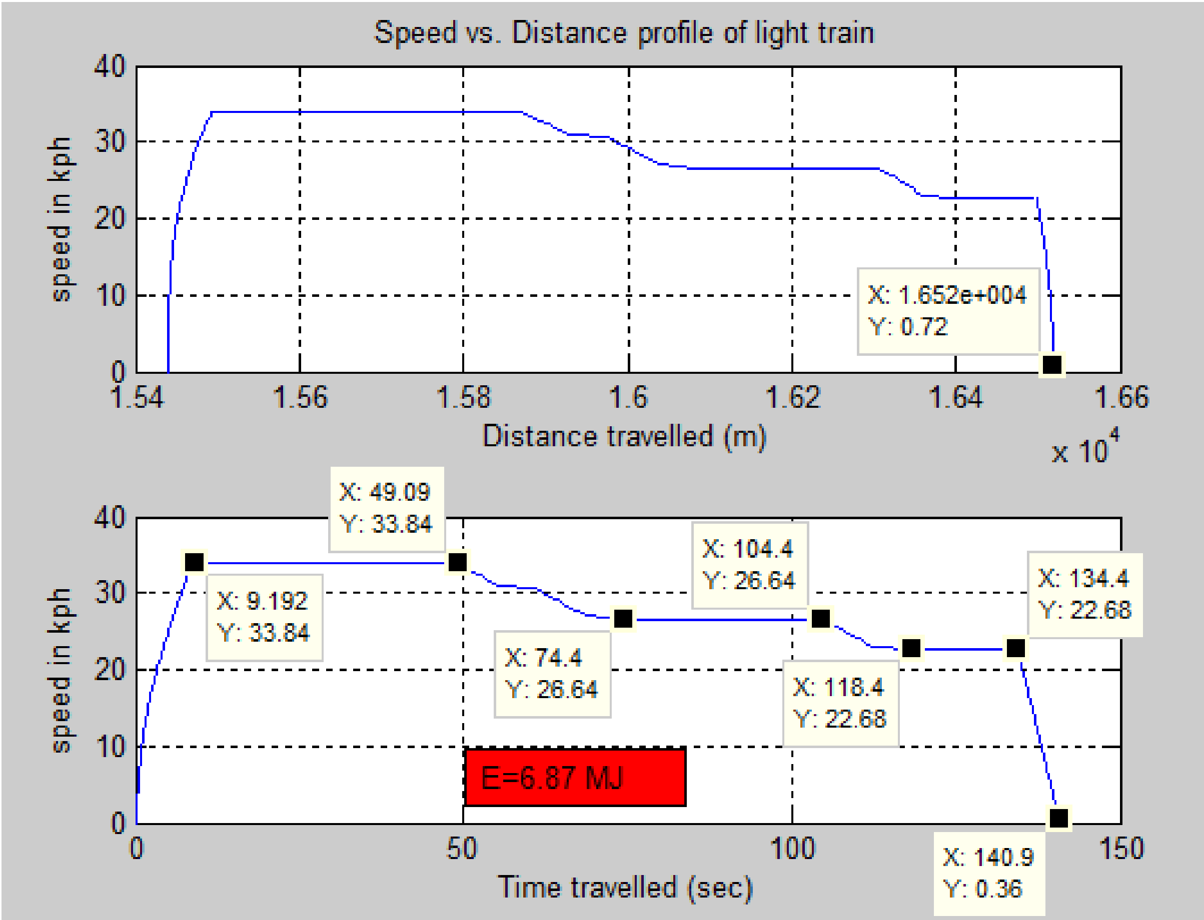


Figure 5.33: Train speed profile 1 for EW8-EW7 movement.

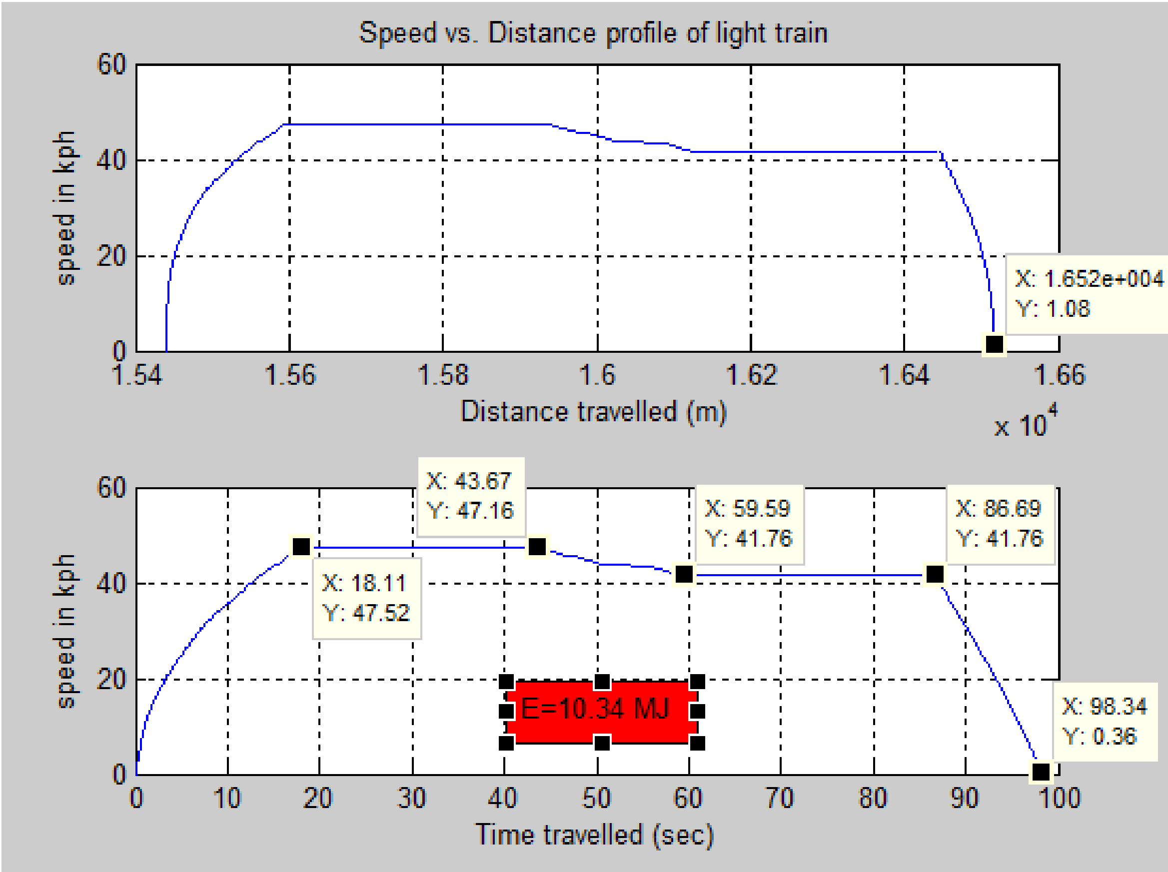


Figure 5.34: Train speed profile 2 for EW8-EW7 movement.

8. Section EW8-EW9

For the upward direction of motion, the fastest trajectory needs only 75 seconds of drive to complete the journey at the expense of 9.88 MJ of energy. The energy conserving trajectory for the same direction of motion consumes only 3.8 MJ of energy to complete the journey within 101 seconds. The downward direction of motion can be completed within just 71 seconds at the expense of the largest amount of energy for the section equal to 11.23 MJ. An energy conserving strategy for the same direction of motion needs an expenditure of 2.57 MJ of energy within 100 seconds of drive.

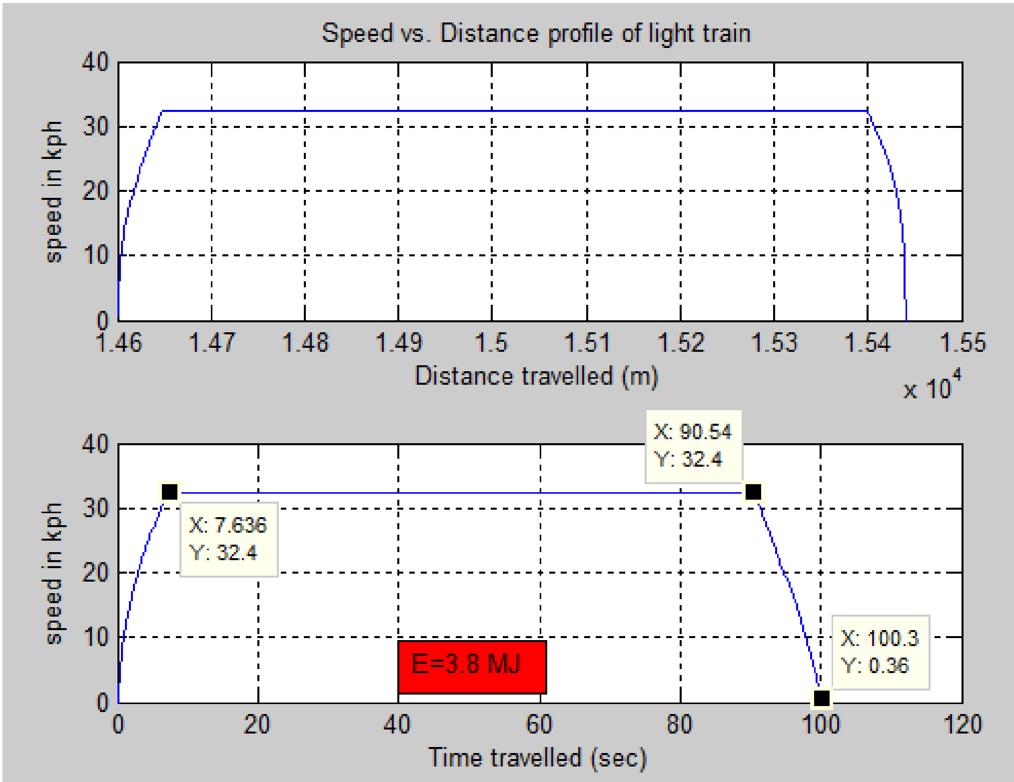


Figure 5.35: Train speed profile 1 for EW8-EW9 movement.

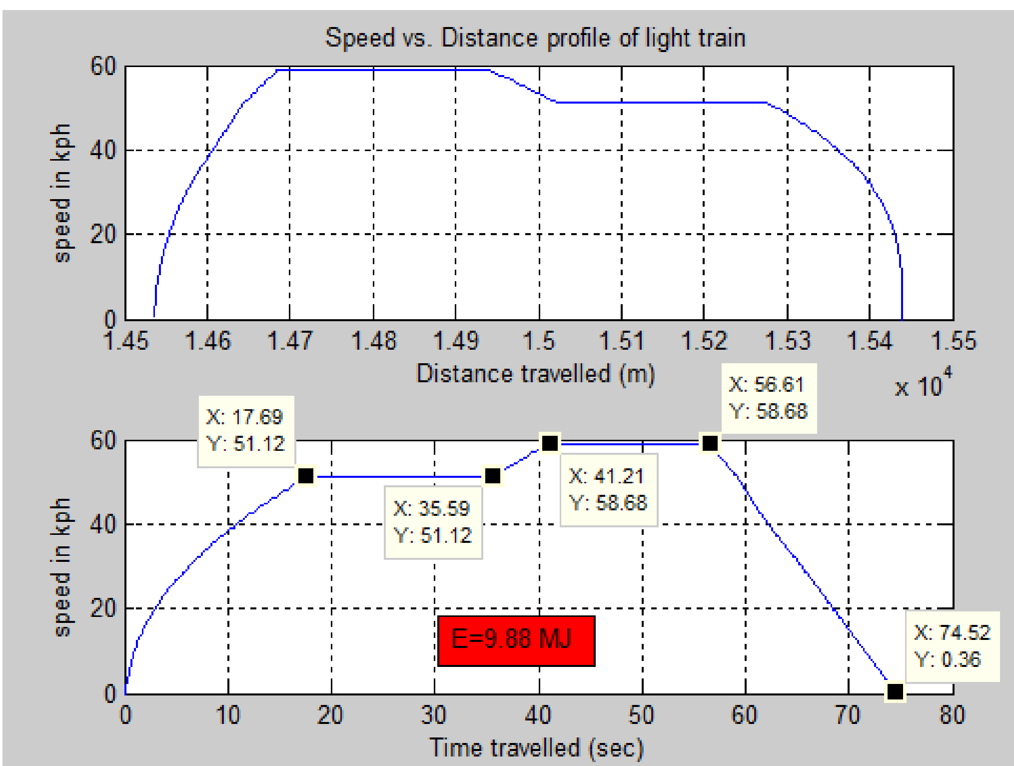


Figure 5.36: Train speed profile 2 for EW8-EW9 movement.

Section	Trajectory	ā	Б	$v_{crossing}$	J_{max}
EW8-EW9	1	4.24	3.42	-	0.55
	2	2.89	3.27	-	0.73
EW9-EW8	1	4.59	2.85	-	0.66
	2	2.38	3.25	-	0.72
	3	3	3.17	-	0.67

Table 5.8 Constraint evaluation for section EW8-EW9

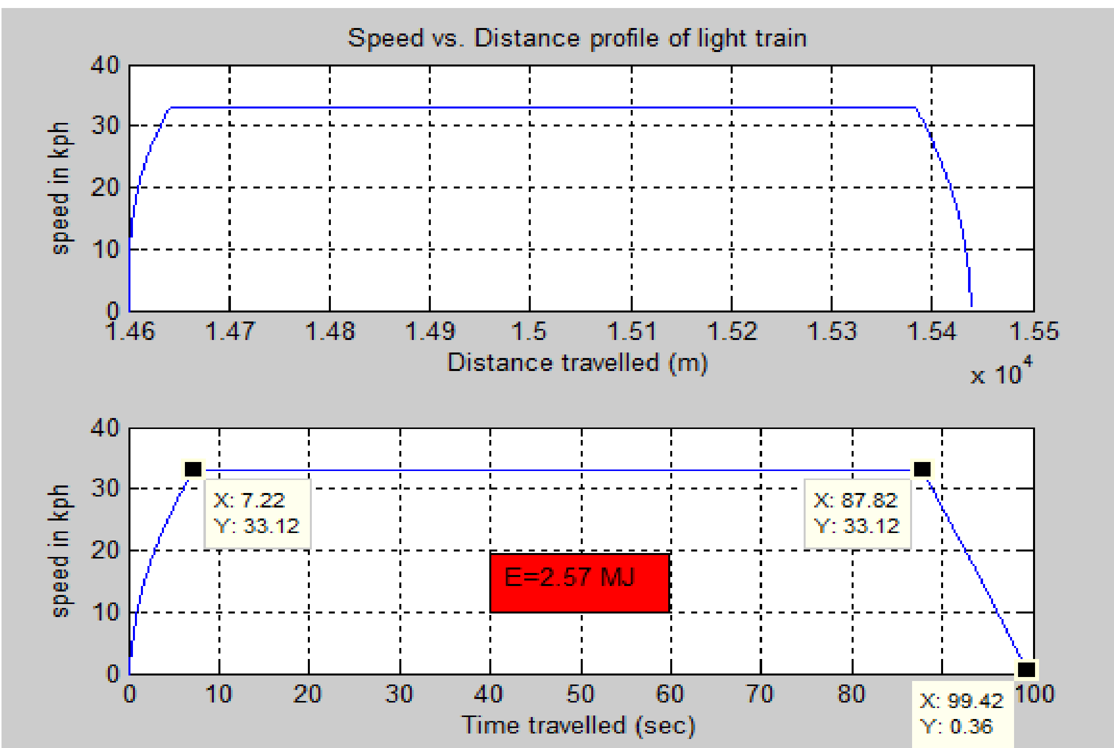


Figure 5.37: Train speed profile 1 for EW9-EW8 movement.

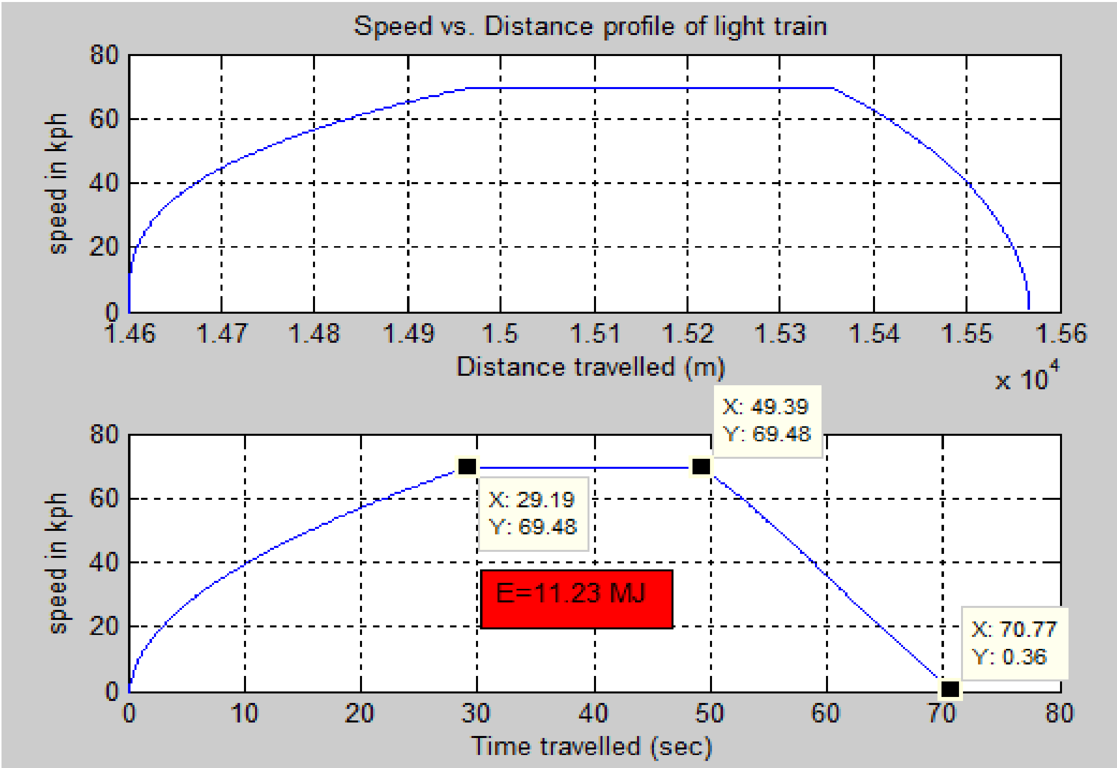


Figure 5.38: Train speed profile 2 for EW9-EW8 movement.

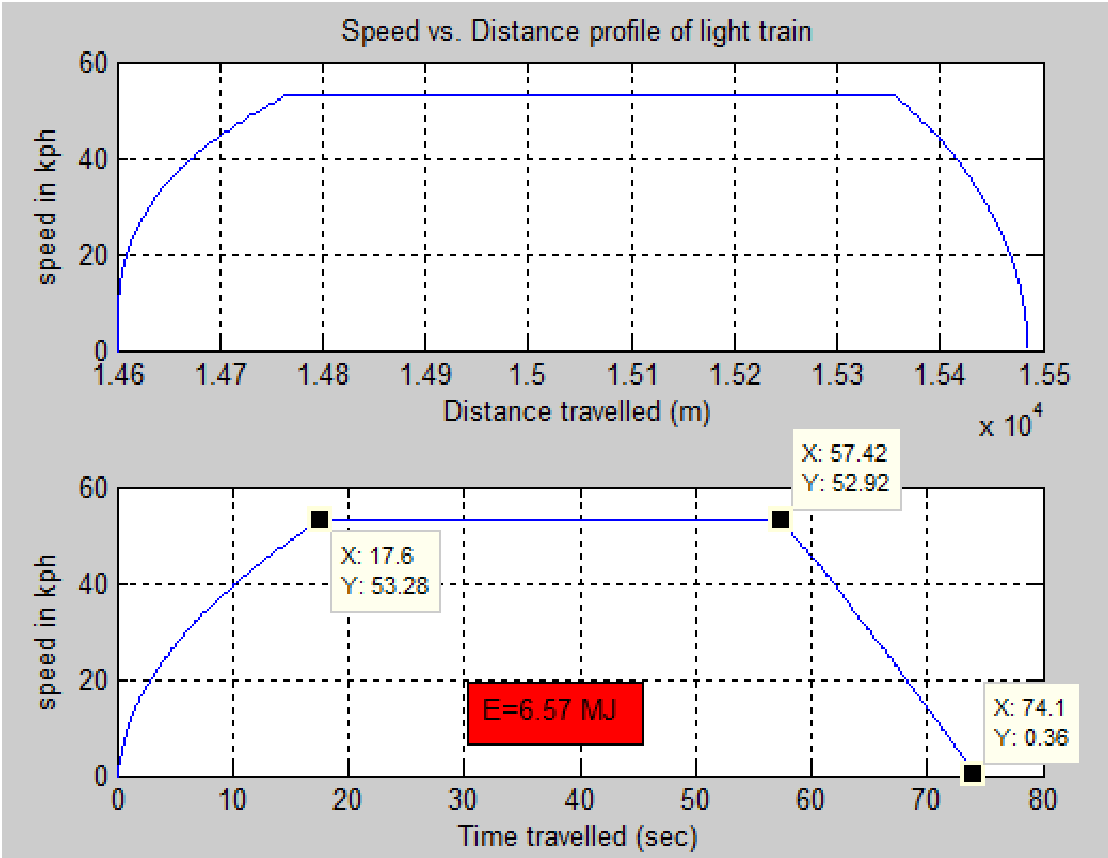


Figure 5.39: Train speed profile 3 for EW9-EW8 movement.

5.1.2 Power plot

By using the speed profiles that are generated as optimal for the railway sections, it can be possible to plot a power versus distance curve. The power curve can be used in an analysis of the power flow in the operation of multiple trains. It can be seen that the power plot consists of constant values for the motoring phase where full powering is applied. Coasting and braking phases have no power consumption, while the cruising phase is associated with variable power demand according to the train resistance values at various locations.

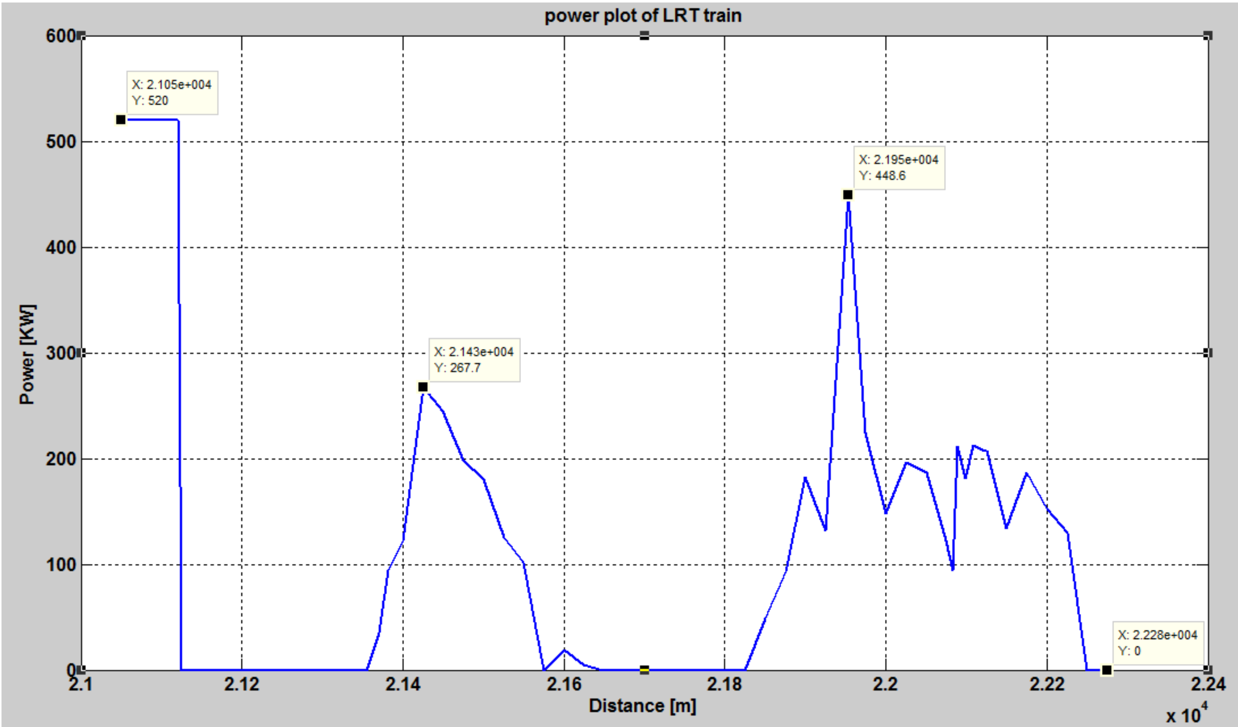


Figure 5.40: Power plot for section EW2-EW1 travel

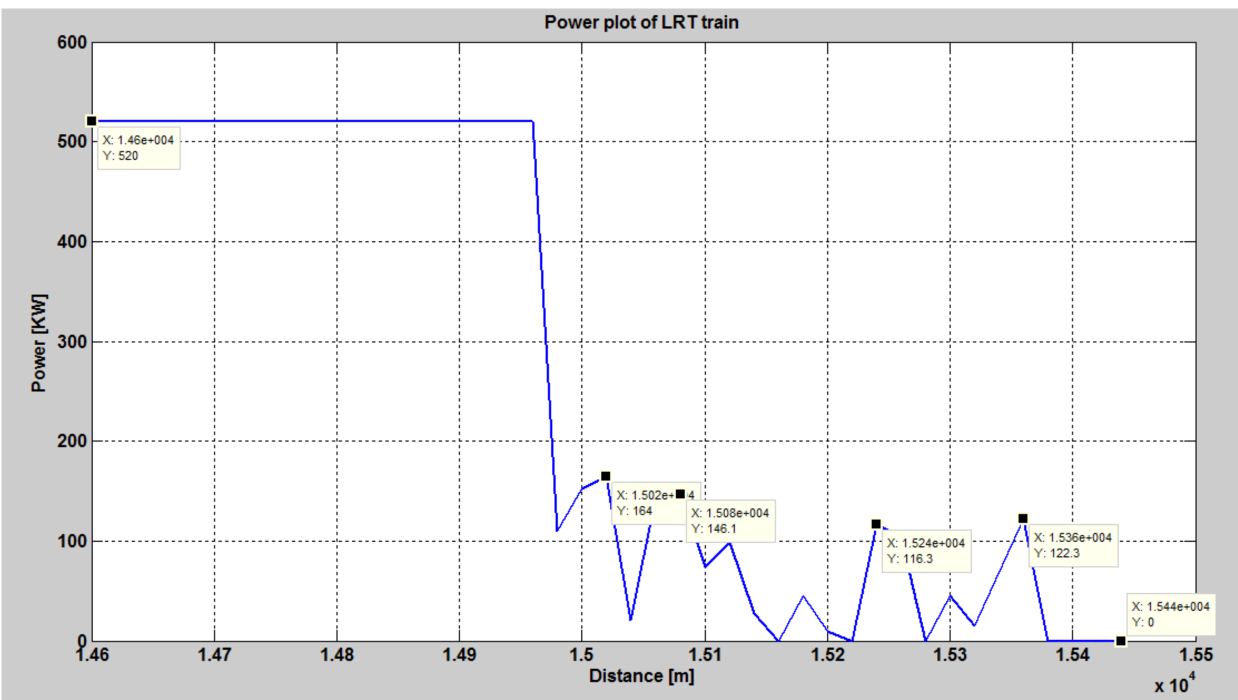


Figure 5.41: Power plot for section EW9-EW8 travel

5.2 Discussion

In this research a two stage multi-objective optimization approach has been used to result in tradeoff solutions as reference speed profiles for the operation of trains on the AALRT network. Various results are achieved for every section from Ayat station to Megenagna station. Constraint handling is made by using different approaches. Track alignment parameters, speed limitations and stopping point requirements are included in the algorithms for the calculation of cost functions. Requirements for the acceleration rate, braking rate and jerk values are handled by individually analyzing resulting trajectories and by eliminating unwanted solutions.

The fastest of all the trajectories is found for the section EW4 to EW3 at the expense of 2.31 kwh of energy where it took it only 70 seconds of ride. The most energy conserving ride is also determined to be in the same section where it needs 0.695 kwh of energy to ride within 90 seconds of time. It is shown that longer sections with prevailing positive track gradient consume too much energy as compared to the sections which are shorter. It is also seen that speed profiles with wider interval of coasting phase are energy conserving as compared to the ones with smaller or no coasting regimes.

It can be seen that riding time between stations can be completed within a time of less than 180 seconds for every section by including station dwelling time. Indeed this is equivalent to 50% reduction from the plan for riding modes.

$$\%\text{ Reduction in riding time} = (\text{planned riding time} - \text{Possible riding time}) / \text{planned riding time} \quad (5.1)$$

$$\%\text{ Reduction in riding time} = (6\text{ min} - 3\text{ min}) / (6\text{ min})$$

$$\%\text{ Reduction in riding time} = 50\%$$

$$\%\, Energy\ saving = \left(\sum E_{fastest} - \sum E_{slowest}\right) / \sum E_{fastest} \quad (5.2)$$

$$\%\, Energy\ saving = (169.06\ MJ - 104.51 MJ) / 169.06 MJ$$

$$\%\, Energy\ saving = 38.18\ \%$$

$$\%\, Time\ saving = \left(\sum t_{slowest} - \sum t_{fastest}\right) / \sum t_{slowest} \quad (5.3)$$

$$\%\, Time\ saving = \frac{1935\ sec - 1471\ sec}{1935\ sec}$$

$$\%\, Time\ saving\ = 23.98\ \%$$

The above figures show that the speed profiles that are generated as the fastest can bring about up to 50% reduction in riding time over the plan. Furthermore, by choosing the fastest trajectories over the slowest ones, it is possible to save up to 38.18% of energy, while 23.98% of reduction in riding time can be achieved by preferring the fastest profiles over slowest ones.

The effect of rail profile is clearly shown in the resulting profiles in terms of the energy consumption a trajectory needs. That is, sections with dominating negative grade resulted in trajectories of smaller energy demand than the opposite ones. For instance, the section from EW 4 to EW5 constitutes of 66.7% positive grade. In that section, the trajectories consume energy from 5.282 kwh to 6.413 kwh. But the reverse section EW5-EW4 needs energy from 0.695 kwh to 1.426 kwh because of the associated negative grade. This shows us that negative gradients have the effect of conserving energy by incorporating coasting phases in the trajectories.

CHAPTER SIX

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

This research aimed to determine optimal speed profiles for the motion of trains on the AALRT network. The dynamics of train operation are represented by a set of kinematic equations by including all the forces that act on the train. Train resistances such as rolling and wind resistances are also considered in the equations of motion. Since the LRT trains are composed of a few number of train cars, a point mass approximation is considered. Important track parameters that are considered in the model include vertical track gradient and horizontal track curvature. A discrete space model has been used in the model. In that case, the distance between stations is disctetized into equal length sections.

Algorithms are developed to calculate cost functions for every type of riding mode. Two cost functions are considered in the optimization problem, energy and time. Algorithms are also developed for the calculation of total energy consumption and total time spent. A two objective optimization problem has been formulated. These objectives are energy and time. Constraints are also defined for speed limitation, stopping distance, acceleration and braking rates, jerk values, adhesion, etc. The decision variable is made to be a vector of riding modes. Each riding mode is represented by an integer value ranging from 1 to 4, representing motoring, cruising, coasting and braking, respectively. Since a 20 section discrete space is used, the decision vector is a 20 dimensional vector, each dimension representing the riding mode for a discrete distance element. The objective space is a two dimensional space with time and energy representing the dimensions of the space.

A MATLAB based multi-objective optimization toolbox is used to result in an optimal set of solutions that are well distributed and are equally optimal. Various settings are made on the built in MATLAB function to ensure speedy convergence and accuracy. The results of the optimizer are then made subject to a post-processing phase which does smooth out trajectories by reducing the number of switching points and minimizing the off-set on the original values of the cost functions. The smoothing algorithm considered various assumptions.

After the two stage optimization, various solutions are achieved as optimal speed profiles for the operation of trains on every section from Ayat station to Megenegna station of AALRT. The results so found are essentially trade-off solutions that, when one is best in minimizing the energy consumption of the ride, the other is best in minimizing the time to travel. Constraint values are also computed for every section and are found to satisfy requirements. Energy consumption by the trajectories ranges from 0.695 kwh to 6.49 kwh. It has been observed that larger amount of energy consumption was associated with wider sections with considerable positive gradient. Ride time between stations ranges from 71 seconds being the fastest of all the trajectories to 165 seconds being the slowest. The fastest of all the trajectories is found for the section EW9 to EW8 at the expense of 3.122 kwh of energy where it took it only 71 seconds of ride. The most energy conserving ride is for the section EW4 to EW3 at 0.695 kwh of energy to ride within 80 seconds of time.

The fastest ride trajectories can be used for peak hour ride. By assuming dynamic dwelling time to ensure that no two trains can enter a section, fastest inter-station running can be completed within just 180 seconds of ride at peak hours. This is equivalent to 50% decrease from the planned 6 minutes riding time between stations. By shifting from the fastest trajectories to the slowest ones for every section from Ayat to Megenagna, it is possible to save up to 38.18% of energy, while 23.98% of reduction in riding time can be achieved by preferring the fastest profiles over slowest ones.

6.2 Future Work

Future research areas related to the speed profile optimization are listed below:

- The resistance equations developed in this paper for the LRT system no longer work for other systems such as subway or Heavy Rail Transit (HRT) where the effect of wind and complex track geometry prevails. Therefore, the equations developed in this paper need to be improved to be applied for those other railway systems.
- The optimization techniques used in this paper can be extended and used for the optimization of train speed profiles for HRT systems.
- Automatic train control systems use various algorithms to determine optimal train speed profiles by collecting instantaneous train status information. The algorithms developed in this paper can be further developed for the application to real time train control systems.

REFERENCES

[1] China Railway Group Limited, “Addis Ababa LRT Project East West Line: Project Study Report”, Ethiopian Railways Corporation, Addis Ababa, 2009

[2] “Bankable Feasibility Study for Addis Ababa Light Rail Transit (AALRT) Project”, Vol. 1, Ethiopian Railways Corporation, Addis Ababa, July 2009

[3] China Railway Group Limited, “Preliminary Design: Chapter XIV Signaling System Volume in One”, Ethiopian Railways Corporation, Addis Ababa, June 2013.

[4] “Bankable Feasibility Study for Addis Ababa Light Rail Transit (AALRT) Project”, Vol. 4, Ethiopian Railways Corporation, Addis Ababa, July 2009

[5] R. Chevier, “An Evolutionary Multi-objective Approach for Speed Tuning Optimization with Energy Saving in Railway Management”, hal-00465116, Version 1, Mar. 19, 2012

[6] P. Howlett, “An optimal strategy for the control of a train”, J. Austral. Math. Soc., Ser. B, Vol. 31, ISSN 1442-4436, pp. 454–471., 1990.

[7] Kitae Kim, “Optimal Train Control on Various Track Alignments Considering Speed and Schedule Adherence”, PHD dissertation, New Jersey Institute of Technology, Jan. 2010

[8] Remy Chevrier, Paola Pellegrini and Joaquin Rodriguez, “Energy saving in railway timetabling: A bi-objective evolutionary approach for computing alternative running times”, hal-00742039, version 1 – 15, Universite Lille Nord de France, Oct. 2012

[9] Remy Chevrier, Paola Pellegrini and Joaquin Rodriguez, “Computing multiple running times for railway timetabling: a speed-level based model for constructing alternative speed profiles”, hal-00909478, version 1, Universite Lille Nord de France, 2012

[10] American Railway Engineering Association, “American Railway Engineering Manual”, Washington, DC, 1999

[11] P. G. Howlett and J. Cheng, “Optimal Driving Strategies for a Train on a track with continuously Varying Gradient”, J. Austral. Math. Soc. Ser. B, Vol. 38, pp. 388-410, 1997

[12] P. Pokorny, “A remark on energy optimal strategies for a train movement”, ANIZAM J. 50 (E), pp. E1-E23, 2008

[13] Y. Wang et al. “Optimal trajectory planning for trains using mixed integer linear programming”, Proceedings of the 14th international IEEE Conference on Intelligent Transportation Systems (ITSC 2011), pp. 1598–1603, Washington, DC, Oct. 2011

[14] Yoshikazu Sawaragi, Hirotaka Nakayama and Tetsuzo Tanino, “Theory of Multi-objective Optimization”, Academic Press, Inc., Tokyo, 1995

[15] H. Ko, T. Koseki, and M. Miyatake, "Application of dynamic programming to the optimization of the running profile of a train", Computers In Railways IX, vol. 15, pp. 103-112, 2004

[16] J. C. Jong and S. Chang, “Algorithms for Generating Train Speed Profiles”, Journal of the Eastern Asia Society for Transportation Studies, Vol. 6, pp. 356 - 371, 2005

[17] Moon-Ho Kang, “A GA-based Algorithm for Creating an Energy-Optimum Train Speed Trajectory”, Journal of International Council on Electrical Engineering, Vol. 1, No. 2, pp. 123-128, 2011

[18] K. K. Wong and T. K. Ho, “Dynamic coast control of train movement with genetic algorithm”, International Journal of Systems Science, 35(13-14). pp. 835-846, 2004

[19] A. Lipetz, “Diesel Engine in rail Transportation”, Engineering Bulletin Research Series 49, Purdue University, Lafayette, India, 1935

[20] W. Hay, “Railroad Engineering”, 2nd Ed., John Wiley & Sons, Inc., New York, 1982

[21] Z. Filipovic, “Electrische Bahnen”, Springer-Verlag, pp. 23-27, Berlin-Haidelberg-New York, 1995

[22] E. Schmidt, “Freight Train Resistance and Its relation to Average Car Weight”, University of Illinois Engineering Experiments Station Bulletin, 31(48), Urbana, Illinois, 1935

[23] W. Davis Jr., “Tractive Resistance of Electric Locomotives and Cars”, General Electric Review 29, pp. 685-708, 1926

[24] American Railway Engineering Association (AREA), “Manual for Railway Engineering”, Washington, DC, 1984

[25] China Railway Group (CRECG) Project Manager Office for Light Rail Project of Ethiopia, “Technical Specifications of Vehicles”, Ethiopian Railways Corporation, Addis Ababa, July 2013

[26] China Railway Group (CRECG) Project Manager Office for Light Rail Project of Ethiopia, “Signaling System Detailed Design”, Ethiopian Railways Corporation, Addis Ababa, Jan. 2014

[27] Railtrack PLC, “Calculation of Brake Force Data for Rolling Stock Library”, London, 1995

[28] K.C. Tan, E.F. Khor and T.H. Lee, “Multiobjective Evolutionary Algorithms and Applications”, Springer-Verlag London Limited, London, 2005

[29] A. Abraham, L. Jain and R. Goldberg, "Evolutionary Multiobjective Optimization Theoretical Advances and Applications", Springer-Verlag London Limited, London, 2005

[30] M. Mitayake, and H. Ko, "Optimization of Train Speed Profiles for Minimum energy Consumption", IEEJ Trans 5: pp. 263-269, 2010

[31] P. Howlett, "The Optimal Control of a Train", Annals of Operations Research 98, pp. 65–87, 2000

[32] X. Li et al., "Energy Constraint Operation Strategy for High Speed Railway", International Journal of Innovative Computing, Information and Control, Volume 8, Number 10(A), October 2012

[33] M. Larranga "Optimization Techniques Applied to Railway Systems", Master thesis, Universitad del Pais, Sep. 21, 2012

APPENDICES

Appendix I: Summary of Driving Strategies

Section	Traj. No.	t1	Next riding mode	t2	Next riding mode	t3	Next Riding mode	t4	Next Riding mode	t5	Next riding mode	t6	Next riding mode	E (MJ)	Time (sec)
EW1-EW2	1	0	Motor	9	Cruise	78.9	Coast	92.2	Cruise	106.5	Brake			**3.26**	**125**
	2	0	Motor	14.5	Cruise	87.4	Brake							**5.23**	**108**
EW2-EW3	1	0	Motor	8.5	Coast	36.5	Cruise	52.2	Coast	72.9	Cruise	86.1	Brake	**3.05**	**104**
	2	0	Motor	13.9	Cruise	82	Brake							**4.91**	**100**
EW3-EW4	1	0	Motor	7.63	Coast	39.6	Cruise	159.5	Brake					**8**	**165**
	2	0	Motor	21.4	Coast	28.9	Cruise	38.9	Motor	52.5	Cruise	61.1	Brake	**17.3**	**79**
	3	0	Motor	7.63	Cruise	29.2	Coast	44.7	Cruise	118.3	Brake			**8.8**	**126**
EW4-EW5	1	0	Motor	27.9	Cruise	72.6	Brake							**23.35**	**84**
	2	0	Motor	12.7	Cruise	26.9	Coast	38.6	Cruise	134.4	Brake			**19**	**140**
EW5-EW6	1	0	Motor	6.9	Cruise	24.9	Coast	40.2	Cruise	110.4	Brake			**8**	**117**
	2	0	Motor	14.2	Coast	31.9	Cruise	48.9	Motor	55.7	Cruise	69.6	Brake	**11.2**	**81**
	3	0	Motor	7.3	Cruise	20.4	Coast	41.9	Motor	51	Cruise	81.4	Brake	**10.2**	**92**
EW6-EW7	1	0	Motor	12.8	Cruise	62.3	Motor	66.3	Cruise	76.9	Brake			**6.74**	**96**
	2	0	Motor	8	Coast	109	Brake							**2.93**	**119**
	3	0	Motor	12.7	Coast	66.9	Cruise	82.9	Brake					**4.75**	**98**
EW7-EW8	1	0	Motor	8.9	Cruise	114.6	Brake							**9.8**	**125**
	2	0	Motor	14.3	Cruise	92.8	Brake							**11.45**	**105**
EW8-EW9	1	0	Motor	7.6	Cruise	90.5	Brake							**3.8**	**101**
	2	0	Motor	17.7	Cruise	35.6	Motor	41.2	Cruise	56.6	Brake			**9.88**	**75**
EW2-EW1	1	0	Motor	15.7	Cruise	63.6	Coast	109.2	Cruise	148.3	Brake			**12.71**	**153**
	2	0	Motor	24.9	Coast	44.4	Cruise	106.4	Brake					**16.42**	**118**
	3	0	Motor	10.2	Cruise	130.8	Brake							**14.76**	**140**
EW3-EW2	1	0	Motor	8.9	Cruise	110.1	Brake							**3.23**	**123**
	2	0	Motor	13.9	Cruise	45.1	Motor	57	Cruise	76.2	Brake			**9.53**	**95**
	3	0	Motor	13.7	Cruise	49.7	Motor	57.5	Cruise	78.8	Brake			**8.22**	**98**
EW4-EW3	1	0	Motor	19.2	Cruise	37.3	Coast	34.2	Cruise	49.3	Brake			**8.31**	**70**
	2	0	Motor	7	Coast	14.6	Cruise	71.6	Brake					**2.5**	**90**
	3	0	Motor	10.8	Cruise	65	Brake							**3.92**	**84**
EW5-EW4	1	0	Motor	7.3	Coast	22.5	Cruise	70.2	Coast	78.7	Brake			**2.59**	**94**
	2	0	Motor	11.5	Cruise	22.4	Coast	58.3	Cruise	67.2	Brake			**4.16**	**84**
EW6-EW5	1	0	Motor	6.7	Cruise	15.7	Coast	46.6	Cruise	114.5	Brake			**5**	**121**
	2	0	Motor	7	Cruise	24.6	Coast	50.2	Cruise	106.9	Brake			**5.26**	**114**

EW7-EW6	1	0	Motor	17.7	Cruise	78.9	Brake							**13.75**	**92**
	2	0	Motor	8.43	Cruise	107	Brake							**11.2**	**117**
EW8-EW7	1	0	Motor	9.2	Cruise	49	Coast	74.4	Cruise	104.4	Coast	118	Cruise	**6.87**	**141**
	2	0	Motor	18.1	Cruise	43.7	Coast	59.6	Cruise	86.7	Brake			**10.34**	**99**
EW9-EW8	1	0	Motor	7.2	Cruise	87.8	Brake							**2.57**	**100**
	2	0	Motor	29.2	Cruise	49.4	Brake							**11.23**	**71**
	3	0	Motor	17.6	Cruise	57.4	Brake							**6.57**	**75**

Appendix II MATLAB Code for Acceleration Regime

```
function [ Time Distance Energy v1 err] = ac( x,g,l,h,dist,e )
clear t v s;
v=zeros(1,100);
u=zeros(1,100);
F=zeros(1,100);
R=zeros(1,100);
A=zeros(1,100);
T=zeros(1,100);
rp=zeros(1,100);
len=round(l(length(l))-l(1))/20;
s=zeros(1,100);
t=zeros(1,100);
count=1;
index(1)=1;
t(1)=h;
v(1)=x;
s(1)=dist;
E=0;
if (v(1)==0)
    F(1)=364000/(x+1);
else
    F(1)=364000/x;
end
vm=19.4;
while s(count)<(len+dist+e) && v(count)<=vm
        if((s(count)>=21940 && s(count)<=21960)||(s(count)>=20067 &&
s(count)<=20216) ||...
        (s(count)>=18990 && s(count)<=19010) || (s(count)>=17142 &&
s(count)<=17287)|| (s(count)>=15790 && s(count)<=15810))
```

```
    vm=13.8;
    end
u(count)=.16+(7.5/(3.6*v(count)+44));
R(count)=1000*1.04*(1.9836+.0415*v(count)+.0009*v(count)^2);
         if (s(count)>=20825 && s(count)<=21081)
            rp(count)=rp(count)+227.3;
         elseif (s(count)>=16563 && s(count)<=17098)
             rp(count)=rp(count)+554.3;
         elseif (s(count)>=17658 && s(count)<=17947)
             rp(count)=rp(count)+990;
         elseif (s(count)>=15097 && s(count)<=15380)
             rp(count)=rp(count)+781.2;
         end
rp(count)=rp(count)+63.02*1000*9.81*1.04*g(index(length(index)));
R(count)=R(count)+rp(count);
m=63020*1.04;
w=9.81*m;
A(count)=w*u(count);
if count~=1
    F(count)=364000/v(count);
    T(count)=min(A(count),F(count));
end
T(1)=min(A(1),F(1));
if (T(count)>R(count))
v(count+1)=v(count)+1;
t(count+1)=t(count)+63020*1.04/(T(count)-R(count));
s(count+1)=s(count)+v(count)*(t(count+1)-t(count));
E=E+T(count)*v(count)*(t(count+1)-t(count));
elseif (T(count)<R(count))
v(count+1)=v(count)-1;
t(count+1)=t(count)-63020*1.04/(T(count)-R(count));
s(count+1)=s(count)+v(count)*(t(count+1)-t(count));
end
if any(find(s(count)>l+(dist-l(1)),1,'last'))
index(count+1)=find(s(count)>l+(dist-l(1)),1,'last');
end
count=count+1;
end
```

```
if count>1
v1=v(count-1);
Time=t(count-1);
Distance=s(count-1);
else
    v1=v(count);
Time=t(count);
Distance=s(count);
end
Energy=E;
err=dist+len+e-Distance;
end
```

Appendix III MATLAB Code for Braking Regime

```
function [ Time Distance Energy v1 err] = br( x,g,l,h,dist,e )
clear t v s;
v=zeros(1,500);
u=zeros(1,500);
F=zeros(1,500);
R=zeros(1,500);
A=zeros(1,500);
T=zeros(1,500);
rp=0;
len=round(l(length(l))-l(1))/20;
s=zeros(1,500);
t=zeros(1,500);
count=1;
index(1)=1;
t(1)=h;
v(1)=x;
s(1)=dist;
E=0;
while s(count)<=(len+dist+e) && floor(v(count))>=(2)
u(count)=.16+(7.5/(3.6*v(count)+44));
R(count)=1000*1.04*(1.9836+.0415*v(count)+.0009*v(count)^2);
         if (s(count)>=20825 && s(count)<=21081)
            rp=rp+227.3;
         elseif (s(count)>=16563 && s(count)<=17098)
```

```
            rp=rp+554.3;
        elseif (s(count)>=17658 && s(count)<=17947)
            rp=rp+990;
        elseif (s(count)>=15097 && s(count)<=15380)
            rp=rp+781.2;
        end
rp=rp+63.02*1000*9.81*1.04*g(index(length(index)));
R(count)=R(count)+rp;
m=63020*1.04;
w=9.81*m;
A(count)=w*u(count);
B1=55640.4;
B(count)=min(B1,A(count));
if B(count)+R(count)>0
v(count+1)=v(count)-1;
t(count+1)=t(count)+63020*1.04/(B(count)+R(count));
s(count+1)=s(count)+v(count)*(t(count+1)-t(count));
elseif B(count)+R(count)<0
v(count+1)=v(count)+1;
t(count+1)=t(count)-63020*1.04/(B(count)+R(count));
s(count+1)=s(count)+v(count)*(t(count+1)-t(count));
end
if any(find(s(count)>l+(dist-l(1)),1,'last'))
index(count+1)=find(s(count)>l+(dist-l(1)),1,'last');
end
count=count+1;
rp=0;
end
v1=v(count-1);
Time=t(count-1);
Distance=s(count-1);
Energy=E;
err=dist+len+e-Distance;
end
```

Appendix IV MATLAB Code for Coasting Regime

```
function [ Time Distance Energy v1 err] = co( x,g,l,h,dist,e)
clear t v s;
```

```
v=zeros(1,1000);
u=zeros(1,1000);
F=zeros(1,1000);
R=zeros(1,1000);
A=zeros(1,1000);
T=zeros(1,1000);
rp=0;
m=63020*1.04;
w=9.81*m;
len=round(l(length(l))-l(1))/20;
s=zeros(1,1000);
t=zeros(1,1000);
count=1;
index(1)=1;
t(1)=h;
v(1)=x;
s(1)=dist;
E=0;
if (v(1)==0)
    F(1)=364000/(x+1);
else
    F(1)=364000/x;
end
vm=19.4;
while s(count)<=(len+dist+e) && ((v(count))>=2 && v(count)<=vm)
    if((s(count)>=21940 && s(count)<=21960)||(s(count)>=20067 &&
s(count)<=20216) ||...
        (s(count)>=18990 && s(count)<=19010) || (s(count)>=17142 &&
s(count)<=17287)|| (s(count)>=15790 && s(count)<=15810))
    vm=13.8;
    end
u(count)=.16+(7.5/(3.6*v(count)+44));
R(count)=1000*1.04*(1.9836+.0415*v(count)+.0009*v(count)^2);
         if (s(count)>=20825 && s(count)<=21081)
            rp=rp+227.3;
         elseif (s(count)>=16563 && s(count)<=17098)
             rp=rp+554.3;
         elseif (s(count)>=17658 && s(count)<=17947)
```

```
            rp=rp+990;
        elseif (s(count)>=15097 && s(count)<=15380)
            rp=rp+781.2;
        end
rp=rp+63.02*1000*9.81*1.04*g(index(length(index)));
R(count)=R(count)+rp;
m=63020*1.04;
w=9.81*m;
A(count)=w*u(count);
if count~=1
    F(count)=364000/v(count);
    T(count)=min(A(count),F(count));
end
T(1)=min(A(1),F(1));
if R(count)>0
v(count+1)=v(count)-1;
t(count+1)=t(count)+63020*1.04/(R(count));
s(count+1)=s(count)+v(count)*(t(count+1)-t(count));
elseif R(count)<0
v(count+1)=v(count)+1;
t(count+1)=t(count)-63020*1.04/(R(count));
s(count+1)=s(count)+v(count)*(t(count+1)-t(count));
end
if any(find(s(count)>l+(dist-l(1)),1,'last'))
index(count+1)=find(s(count)>l+(dist-l(1)),1,'last');
end
count=count+1;
rp=0;
end
v1=v(count-1);
Time=t(count-1);
Distance=s(count-1);
Energy=E;
err=dist+len+e-Distance;
end
```

Appendix V MATLAB Code for Cruising Regime

```
function [ Time Distance Energy v1 err] = cr( x,g,l,h,dist,e)
```

```
clear t v s;
v=zeros(1,500);
u=zeros(1,500);
F=zeros(1,500);
R=zeros(1,500);
A=zeros(1,500);
T=zeros(1,500);
rp=zeros(1,500);
len=round(l(length(l))-l(1))/20;
s=zeros(1,500);
t=zeros(1,500);
count=1;
index(1)=1;
t(1)=h;
v(1)=x;
s(1)=dist;
E=0;
if (v(1)==0)
    F(1)=364000/(x+1);
else
    F(1)=364000/x;
end
vm=19.4;
while s(count)<=(len+dist+e) && ((v(count))>=3 && v(count)<=vm)
        if((s(count)>=21940 && s(count)<=21960)||(s(count)>=20067 &&
s(count)<=20216) ||...
        (s(count)>=18990 && s(count)<=19010) || (s(count)>=17142 &&
      s(count)<=17287)|| (s(count)>=15790 && s(count)<=15810))
    vm=13.8;
    end
u(count)=.16+(7.5/(3.6*v(count)+44));
R(count)=1000*1.04*(1.9836+.0415*v(count)+.0009*v(count)^2);
         if (s(count)>=20825 && s(count)<=21081)
            rp(count)=rp(count)+227.3;
         elseif (s(count)>=16563 && s(count)<=17098)
             rp(count)=rp(count)+554.3;
         elseif (s(count)>=17658 && s(count)<=17947)
             rp(count)=rp(count)+990;
```

```
         elseif (s(count)>=15097 && s(count)<=15380)
             rp(count)=rp(count)+781.2;
         end
rp(count)=rp(count)+63.02*1000*9.81*1.04*g(index(length(index)));
R(count)=R(count)+rp(count);
m=63020*1.04;
w=9.81*m;
A(count)=w*u(count);
if count~=1
    F(count)=364000/v(count);
    T(count)=min(A(count),F(count));
end
T(1)=min(A(1),F(1));
v(count+1)=v(count);
t(count+1)=t(count)+1;
s(count+1)=s(count)+v(count);
if R(count)>0
E=E+R(count)*v(count);
end
if any(find(s(count)>l+(dist-l(1)),1,'last'))
index(count+1)=find(s(count)>l+(dist-l(1)),1,'last');
end
count=count+1;
end
if count>1
v1=v(count-1);
Time=t(count-1);
Distance=s(count-1);
else
v1=v(count);
Time=t(count);
Distance=s(count);
end
Energy=E;
err=dist+len+e-Distance;
end
```

Appendix VI MATLAB Code for Total Energy Calculation

```
function [ y ] = f12( m )
m=round(m);
clc;
clear tt D E l g;
l=[21050 21075 21100 21120.6 21124.4 21150 21175 21200 21225 21236.7 21240.5
21265 21290 21315 21340 21350 21355.6 21360.4 21370.6 21375.8 21381.3 21400
21425 21450 21475 21500 21525 21550 21575 21600 21625 21649 21675 21700
21725 21750 21769.6 21775.2 21800 21825 21850 21875 21900 21925 21953.7
21975 22000 22025 22050 22075 22083.8 22089.5 22098.7 22108.9 22120.5
22125.8 22150 22175 22200 22225 22250 22275];

g=[0.011200703 0.01320115 0.009223693 -0.007894983 -0.02461683 -0.039631086
-0.056891848 -0.072993684 -0.056500223 -0.052704628 -0.063392297
-0.064936479 -0.046851336 -0.030013509 -0.018002917 -0.014287172
-0.014584884 -0.001960788 0.00192308 0.00727292 0.012300396 0.017202545
0.042438164 0.038428343 0.030414057 0.027210067 0.017602727 0.013601258
-0.028959504 -0.0008 -0.003333352 -0.01153923 -0.019603766 -0.012801049
-0.02841146 -0.021433493 -0.012500977 -0.033032361 -0.005600088 -0.00520007
0.004400043 0.012000864 0.027610518 0.018803323 0.07390997 0.034821091
0.021605041 0.030013509 0.02841146 0.017047931 0.012281628 0.032626046
0.027461329 0.032776212 0.032091985 0.0318343 0.01920354 0.02841146
0.022405622 0.018403116 0.018002917 0.007600219];
l=wrev(l);
g=-wrev(g);
err=0;
[tt(1) D(1) E(1) v1 err]=ac1(0,g,l,0,l(1),err);
for i=2:18
    if m(i)==1
        [tt(i) D(i) ,E(i), v1 err]=ac1(v1,g,l,tt(i-1),D(i-1),err);
    elseif m(i)==2
            [tt(i) D(i) ,E(i), v1 err]=cr1(v1,g,l,tt(i-1),D(i-1),err);
    elseif  m(i)==3
                [tt(i) D(i) ,E(i), v1 err]=co1(v1,g,l,tt(i-1),D(i-1),err);
    elseif m(i)==4 && v1>=8
        [tt(i) D(i) ,E(i), v1 err]=br1(v1,g,l,tt(i-1),D(i-1),err);
    elseif m(i)==4 && v1<8
        [tt(i) D(i) ,E(i), v1 err]=ac1(v1,g,l,tt(i-1),D(i-1),err);
```

```
    end
end
D(19)=D(18);
tt(19)=tt(18);
[~,~,e]=bdist1(v1,D(19),g,l);
rp1=0;
R1=0;
E(19)=0;
count=1;
R1(count)=0;
while e<0
tt(19)=tt(19)+1;
D(19)=D(19)-v1;
[~,~,e]=bdist1(v1,D(19),g,l);
if any(D(19)<=(l+(D(18)-l(1))))
index=find(D(19)<=(l+(D(18)-l(1))),1,'last');
end
R1(count)=1000*1.04*(1.9836+.0415*v1+.0009*v1^2);
rp1=rp1+63.02*1000*9.81*1.04*g(index(length(index)));
R1(count)=R1(count)+rp1;
if R1(count)>0
E(19)=E(19)+R1(count)*v1;
end
rp1=0;
count=count+1;
end
[tt(20) D(20) ,E(20), v1 err]=br1(v1,g,l,tt(19),D(19),err);
y=sum(E);
end
```

Appendix VII MATLAB Code for Riding Time Calculation

```
function [ c] = fc12( m )
clear tt D E;
clc;
m=round(m);
l=[21050 21075 21100 21120.6 21124.4 21150 21175 21200 21225 21236.7 21240.5
21265 21290 21315 21340 21350 21355.6 21360.4 21370.6 21375.8 21381.3 21400
21425 21450 21475 21500 21525 21550 21575 21600 21625 21649 21675 21700
```

```
21725  21750  21769.6  21775.2  21800  21825  21850  21875  21900  21925  21953.7
21975  22000  22025  22050  22075  22083.8  22089.5  22098.7  22108.9  22120.5
22125.8 22150 22175 22200 22225 22250 22275];

g=[0.011200703 0.01320115 0.009223693 -0.007894983 -0.02461683 -0.039631086
-0.056891848 -0.072993684 -0.056500223 -0.052704628 -0.063392297
-0.064936479 -0.046851336 -0.030013509 -0.018002917 -0.014287172
-0.014584884 -0.001960788 0.00192308 0.00727292 0.012300396 0.017202545
0.042438164 0.038428343 0.030414057 0.027210067 0.017602727 0.013601258
-0.028959504 -0.0008 -0.003333352 -0.01153923 -0.019603766 -0.012801049
-0.02841146  -0.021433493  -0.012500977  -0.033032361  -0.005600088  -0.00520007
0.004400043 0.012000864 0.027610518 0.018803323 0.07390997 0.034821091
0.021605041 0.030013509 0.02841146 0.017047931 0.012281628 0.032626046
0.027461329   0.032776212   0.032091985   0.0318343   0.01920354   0.02841146
0.022405622 0.018403116 0.018002917 0.007600219];
l=wrev(l);
g=-wrev(g);
err=0;
[tt(1) D(1) E(1) v1 err]=ac1(0,g,l,0,l(1),err);
for i=2:18
    if m(i)==1
        [tt(i) D(i) ,E(i), v1 err]=ac1(v1,g,l,tt(i-1),D(i-1),err);
    elseif m(i)==2
            [tt(i) D(i) ,E(i), v1 err]=cr1(v1,g,l,tt(i-1),D(i-1),err);
    elseif  m(i)==3
                [tt(i) D(i) ,E(i), v1 err]=co1(v1,g,l,tt(i-1),D(i-1),err);
    elseif m(i)==4 && v1>=8
        [tt(i) D(i) ,E(i), v1 err]=br1(v1,g,l,tt(i-1),D(i-1),err);
    elseif m(i)==4 && v1<8
        [tt(i) D(i) ,E(i), v1 err]=ac1(v1,g,l,tt(i-1),D(i-1),err);
    end
end
D(19)=D(18);
tt(19)=tt(18);
[~,~,e]=bdist1(v1,D(19),g,l);
while e<0
tt(19)=tt(19)+1;
D(19)=D(19)-v1;
```

```
[~,~,e]=bdist1(v1,D(19),g,l);
end
[tt(20) D(20) ,E(20), v1 err]=br1(v1,g,l,tt(19),D(19),err);
c=tt(20);
end
```

Appendix VIII MATLAB Code for Brake Distance Calculation

```
function [ v1,bd,err ] = bdist( x,ds,g,l )
m=63020;
w=9.81*m;
u=zeros(1,100);
R=zeros(1,100);
A=zeros(1,100);
rp=zeros(1,100);
s=zeros(1,100);
t=zeros(1,100);
count=1;
index(1)=1;
t(1)=0;
v(1)=x;
s(1)=ds;
while v(count)>=0
%calculate values for the vectorized parameters
u(count)=.16+(7.5/(3.6*v(count)+44));
R(count)=1000*1.04*(1.9836+.0415*v(count)+.0009*v(count)^2);
         if (s(count)>=21040 && s(count)<=21100)
            rp(count)=rp(count)+1516;
         end
rp(count)=rp(count)+63.02*1000*9.81*g(index(length(index)));
R(count)=R(count)+rp(count);
A(count)=w*u(count);
B1=55640.4;
B(count)=min(B1,A(count));
v(count+1)=v(count)-1;
t(count+1)=t(count)+63020*1.04/(B(count)+R(count));
s(count+1)=s(count)+v(count)*(t(count+1)-t(count));
if any(find(s(count)>l+(ds-l(1)),1,'last'))
index(count+1)=find(s(count)>l+(ds-l(1)),1,'last');
```

```
end
count=count+1;
end
v1=v(count-1);
bd=s(count-1);
err=l(length(l))-bd;
end
```

Appendix IX MATLAB Code for Brake Curve Calculation

```
function [t v s ] = bp( x,g,l,h,dist)
clear t v s;
v=zeros(1,100);
u=zeros(1,100);
R=zeros(1,100);
A=zeros(1,100);
rp=0;
w=9.81*63020;
len=round(l(length(l))-l(1))/20;
s=zeros(1,100);
t=zeros(1,100);
count=1;
index(1)=1;
t(1)=h;
v(1)=x;
s(1)=dist;
E=0;
while floor(v(count))>=0
u(count)=.16+(7.5/(3.6*v(count)+44));
R(count)=1000*1.04*(1.9836+.0415*v(count)+.0009*v(count)^2);
         if (s(count)>=20825 && s(count)<=21081)
            rp=rp+227.3;
         elseif (s(count)>=16563 && s(count)<=17098)
             rp=rp+554.3;
         elseif (s(count)>=17658 && s(count)<=17947)
             rp=rp+990;
         elseif (s(count)>=15097 && s(count)<=15380)
             rp=rp+781.2;
         end
```

```
rp=rp+63.02*1000*9.81*1.04*g(index(length(index)));
R(count)=R(count)+rp;
A(count)=w*u(count);
B1=55640.4;
B(count)=min(B1,A(count));
v(count+1)=v(count)-.1;
t(count+1)=t(count)+6302*1.04/(B(count)+R(count));
s(count+1)=s(count)+v(count)*(t(count+1)-t(count));
if any(find(s(count)>l+(dist-l(1)),1,'last'))
index(count+1)=find(s(count)>l+(dist-l(1)),1,'last');
end
count=count+1;
rp=0;
end
t=t(1:count-1);
v=v(1:count-1);
s=s(1:count-1);
v1=v(count-1);
end
```

Appendix X MATLAB Code to Smooth out Speed Profiles

```
function [ c ] = smz( a )
[x ~]=size(a);
for k=1:x
    j=1;
    b=a(k,[2:j+6]);
        ind=find(b==1);
        len=length(ind);
        i=1;
        while i<=len
            temp=a(k,j+i);
            a(k,j+i)=a(k,ind(i)+j);
            a(k,ind(i)+j)=temp;
            i=i+1;
        end
        j=j+len+1;
    while j<=18
        if j <12
```

```
            b=a(k,[j+1:j+6]);
        else
            b=a(k,[j+1:18]);
        end
        ind=find(b==a(k,j));
        len=length(ind);
        i=1;
        while i<=len
            temp=a(k,j+i);
            a(k,j+i)=a(k,ind(i)+j);
            a(k,ind(i)+j)=temp;
            i=i+1;
        end
        j=j+len+1;
        if(18-j)<4
            break;
        end
    end
end
c=a;
end
```