Addis Ababa University
Addis Ababa Institute of technology
Department of Electrical and Computer Engineering

Addis Ababa City Light Railway Transit Speed Profile for Optimum Train Drive: Case of Line from Menelik Square to Kality

A Thesis Submitted to Addis Ababa University
Addis Ababa Institute of technology
In Partial Fulfillment of the Requirement for the Degree of Master of
Science in Electrical Engineering
(Railway Engineering)

By
Abebe Teklu

Advisor
Dr. Mengesha Mamo

August, 2014
Addis Ababa, Ethiopia

Addis Ababa University

Addis Ababa Institute of technology

Department of Electrical and Computer Engineering

Addis Ababa City Light Railway Transit Speed Profile for Optimum Train Drive: Case of Line from Menelik Square to Kality

By AbebeT eklu

APPROVED BY BOARD OF EXAMINERS

Dean, School of Electrical and Computer Engineering	Signature
Dr. Mengesha Mamo Advisor	Signature
Internal Examiner	Signature
External Examiner	Signature

Declaration

I, Abebe Teklu, 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: Abebe Teklu Signature: _______________

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

Date of Submission: ____

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

Dr. Mengesha Mamo Signature: ______________

Advisor's Name

Abstract

Optimum train driving is the key aspect of railway transportation development and depends on energy consumed, passenger comfort and punctuality. In this thesis three mathematical models have been developed considering nearly all traction mechanics and track information to generate appropriate speed profile. The first problem solved is minimization of energy consumed whereby the given journey must be completed on the allocated time. The second problem is minimization of journey time to complete the given journey; here the maximum energy consumed (the drive rating not to be exceeded) for fastest train drive is taken as one of the constraints. The third problem is designed to have different speed profile for different traffic condition ranging from high to low exempting the two extremes. In order to generate appropriate speed of the train at each discrete point, we used genetic algorithm with Deb's constraint handling technique. Finally, we implemented the problems for North to South line of Addis Ababa light railway transit. The result obtained from simulation helps the driver to attain the required driving performance by driving the train with the respective speed profile.

Key terms: speed profile, genetic algorithm, optimization, journey (travel) time, energy saving, traction system, train movement resistance, optimum train driving.

Acknowledgment

I thank the Almighty God for making me the guy whom I used to be and my advisor Dr. Mengesha Mamo for his continuous support and encouragement. Thank you my mother Yekite Mekonn and my father Teklu Woldegiorgis for raising me as wise as you are. I also thank my brothers, sisters and friends for their continuous motivation to finish this program.

Table of content

List of figures

List of tables

List of symbols and abbreviation

List abbreviation

EEPCO	Ethiopian Electric Power Corporation
ERC	Ethiopia railway corporation
GW	Giga Watt
ICS	Interconnected System
J	Joule
Km/hr	Kilometer per hour
lb	lower bound
LRT	Light railway transit
mps	Meter per second
MW	Mega Watt
SCS	Self-Contained System
sec	Second
ub	upper bound

List symbols

a	Acceleration in meter-per-second-square
A_{tr}	Cross-sectional area of the train
A_{tu}	Cross-sectional area of the tunnel
$\mathcal{B}$	Blocking factor
B_t	Braking Effort (newton)
C_{ar}	Air drag coefficient
$C_{rolling}$	Rolling resistance coefficient

D	Degree of main curve (degree)
E	Energy consumed (joule)
F_{ar}	Aero dynamic resistance (newton)
F_{cr}	Curve resistance (newton)
F_{gr}	Gradient resistance (newton)
$F_{rolling}$	Rolling resistance (newton)
F_{rr}	Running resistance (newton)
F_t	Tractive effort (newton)
g	Gravitational acceleration
L	Length of main curve (m)
P	Power in Watt
R	Radius of curvature
S	Length of sub-curve (m)
T_{run}	Time of run
V	Speed in meter-per-second
V_{rel}	Relative speed of the wind (meter per second)
Θ	Angle of sub-curve (degree)
μ	Coefficient of friction
ρ	Air density in kg/ cubic-meter
τ	Torque

1. Introduction

1.1. Background

Energy consumption is an essential element of economic development and this development is resulted from expansion of industrialization with huge amount of energy consumption for productivity. Industrialization is not only the advantage for development but has negative effect on environment and cost of energy consumed. With these and other core reasons, scientists tried to develop new technologies or enhance the performance of already existed technologies for better energy consumption with noticeable economic development in a given sector. For example, railway transportation system evolved from steam engine drive through diesel engine to purely electrical train. Due to this continuous improvement; noise and environmental pollution from train operation greatly reduces. Practically, greater energy consumption efficiency has an advantage in reduction of environmental pollution and operating cost with increase of profit **[1]**.

Energy efficiency is clearly defined as: *Energy efficiency is understood to mean the utilization of energy in the most cost effective manner to carry out a manufacturing process or provide any service, whereby energy waste is minimized and the overall consumption of primary energy resources is reduced* **[1]**. In general, energy consumption scheme of any energy consuming sector must be efficient enough so that wastage of power is greatly reduced.

Transportation system is one of the services on which most of the world populations are forced to use. Huge amount of electrical power is consumed and has ultimate effect on the environment. These effects can be classified as: **[2], [3]**

- Air pollutant gases like Carbon Monoxide (CO), Nitrogen Oxides (NO_x), Hydrocarbons and Volatile Organic Compounds (HC/VOC), Carbon Dioxide (CO_2), Sulfur Dioxide (SO_2) and other greenhouse gases , and
- Water pollutants which are resulted from air pollutant fallout, Construction and Maintenance of Infrastructure waste.

- Noise pollution: road accounts for approximately 70% of total noise emissions by transportation, rail accounts for 10% of total noise emissions by transportation, and air transportation accounts for 20% of total noise emissions by transportation for Belgium, France, Germany, Sweden, United Kingdom, Unites States and Japan till 1990.

Due to the above reasons, it is mandatory to design a better means of transportation system with minimum energy consumption and environmental friendly behavior.

Development strategy of railway industry is in the direction of minimizing energy consumed and total travel time to complete the given journey. Energy consumed by train operation comprises of mechanical energy used for overcoming driving resistance and losses caused by transmission and transformation of energy in traction and power supply chain [**4**].

Energy consumed by train operation can be reduced using mass reduction, aerodynamic and friction reduction by appropriate design of train and space utilization **[4].** It is also possible to use optimal construction parameters of railway lines, modern design of railway stations, optimal traffic management and energy efficient dynamic train movement in order to achieve optimum train driving **[5]**. The other method is using regenerative braking where braking power is feedback to the catenary and other train running on the same track consumes it. We can also use efficient convertors, to mean, use semiconductor device for voltage conversion which results in quality power and low power loss **[6].**

Controlling speed trajectory of the train is an efficient energy saving strategy and this can be achieved by generating efficient speed profile based on wayside signals, capacity of the train and track information (curve, gradient, speed restriction of the track)**[7].** Train running between two stations has four running steps. The initial step of train driving is acceleration. In this step, the startup resistors are cutout one by one. When all the starting resistors are cutout the train cruises (zero acceleration). When the train approaches the next station, it is possible to disconnect power from motors which results in what is called Coasting. During coasting the train decelerates. Further, the driver can apply break to stop the train faster. Depending on the topology of the track between stations, controlling stage change can result energy efficient train operation **[8].**

Explaining each step [**8**, **9**]:-

- **Acceleration**

  During this stage speed of the train is increasing and traction force required is usually significantly larger than the aerodynamic, rolling and possibly gradient resistance. Acceleration can also be obtained from highly negative gradient track without the requirement of tractive effort.

- **Constant speed operation (cruising)**

  Since the speed is considered constant, the required traction force is equal to the sum of the driving resistances (aerodynamic, rolling and gradient). Energy may or may not be consumed depending on the sign of total resistance.

- **Rolling(coasting)**

  Both rolling and braking are portions of the operation where the speed decreases. In the case of rolling, the braking force and the traction force are both equal to zero. During such driving, speed of the train is obtained from driving resistance and momentum.

- **Braking**

  Traction force is zero and speed of the train is obtained from the combination of braking force and driving resistance.

Travel time is also the core section of railway transportation system to attract considerable number of passengers **[10].** Station to station speed trajectory determines not only energy consumed but also the total travel time. So, generating appropriate speed trajectory of the train can result optimum journey time driving and/or energy consumption driving based on the interest of optimality. Train running along the track is based on the timetable which defines journey time to complete the given journey. Timetable always adds slack time to counter balance unexpected time loss **[11]**.

In general, having better journey time is in the expense of energy to overcome all driving resistance along the track. Practically, speed of the train can't increase without bound because of so many constraints considered such as passenger comfort, speed limit of the track, mechanical design of the vehicle, maximum energy consumption level and speed limit of the vehicle itself. So, energy consumption and journey time should be studied together in order to achieve optimum solution on both terms.

Generation of speed profile for optimum train driving is a complex problem subjected to linear and nonlinear constraints. To solve such kind of problems, we need to have good knowledge of solving techniques **[12, 13]**.

1.2. Statement of the problem

Ethiopian Electric Power Corporation (EEPCO) has two power supply system namely interconnected system (ICS) and self-contained system (SCS). The ICS consists of 11 hydro plants, one geothermal plant and 15 diesel power plants with a total capacity of 2022.2 MW, of which 91% is generated from hydropower plants. The SCS comprises of small hydropower and diesel plant, which can produce a total power of 36.21MW **[14]**. EEPCO has new hydropower project as part of government's growth and transformation plan. According to the plan, Ethiopia will generate a total of over 10GW in 2014/15 **[15].**

According to **[16],** Ethiopia is one of the seven fastest growing countries in African and development comes in the cost of energy. From the total power generated and country's development point of view we can argue that energy consumption scheme of each sector must be efficient. To mean, unnecessary energy wastage must be eliminated by ensuring energy efficient operation for productivity.

Ethiopian Railway Corporation (ERC) is in position of controlling Ethiopian railway transport sector and energy utilization of train operation must be efficient. Not only energy consumption of train operation but also travel time must be in optimality condition in order to attract considerable number of passengers.

Driving performance which is amount of energy consumed and time required to complete the given journey is dependent on speed trajectory used by the train. Speed profile signifies time spent to complete the given journey and energy consumed for that period **[8]**. Generally, train driving requires shortest possible time and minimum energy consumption. For this, best means of generating appropriate speed profile must be developed. Train run between two stations has four stages: Acceleration (below maximum allowable acceleration), Constant speed (below maximum allowable speed), Coasting and Braking (below maximum deceleration). By controlling the speed transition dynamically we can achieve optimum train driving performance.

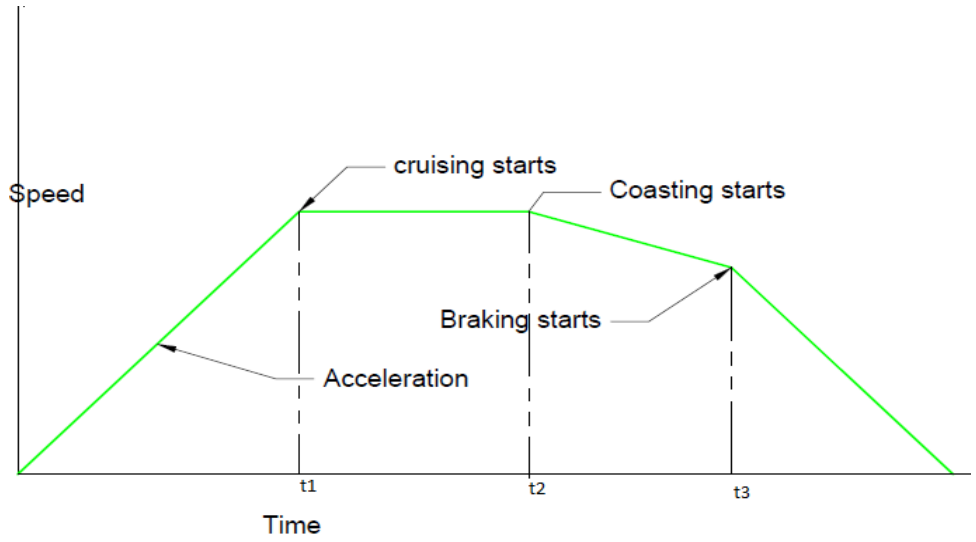


Figure 1-1 Train speed profile

1.3. Objective of the thesis

1.3.1. General objective

- To develop model which considers mechanics of the train and track profile information for generating appropriate speed profile which results optimum train drive. Optimum train drive is driving of train with minimum energy consumption and/or travel time depending on different operational scenario.

1.3.2. Specific objective

- Understand and analyze the designed speed profile
- Study and select desired operation scenarios depending on traffic
- Develop MATLAB program for the analysis of the model
- Derive optimal speed profiles for different running scenarios

1.4. Research method

Initially, literature review is conducted. It is the study of different research works on related topics, traction system and optimization methods used for solving optimization problems. Having brief understanding of the problem, collection of necessary information which is essential to achieve the objective of this thesis is done. The primary information required for the development of mathematical model is track and train information. Even though information collection is not completed due to unavailability of the required documents; assumptions and information from international railway manufacturing companies is used.

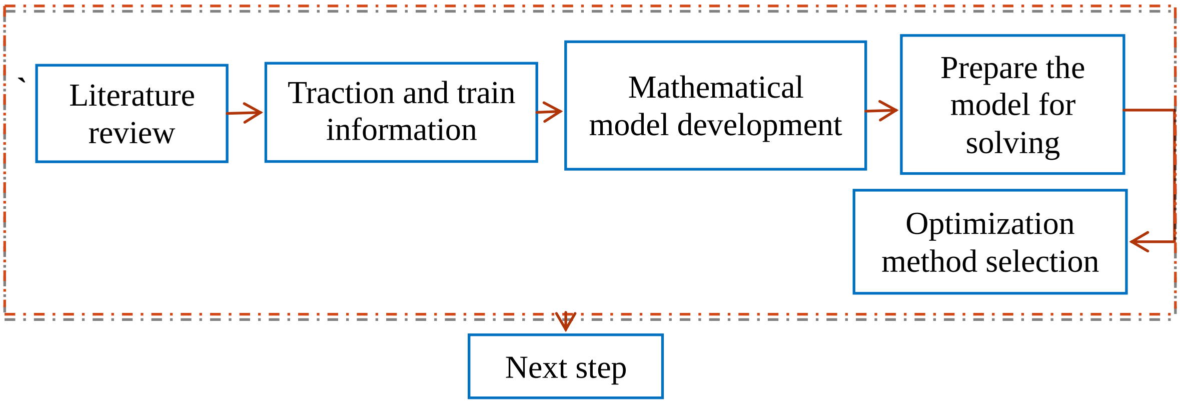


Figure 1-2 Research method

After selection of optimization technique, which is genetic algorithm, how to apply the method for the problem at hand follows. Practically, the very problem of genetic algorithm for constrained optimization problem is the way how to handle constraints. For this matter, after intensive review of constraint handling technique, Deb's constraint handling technique is selected. Having the method of handling the constraints, the principle used in genetic algorithms is applied. Selection, crossover, mutation and replacement strategies are selected from the available technique of real coded genetic algorithm. After the development of pseudo algorithm for each operation, MATLAB code is developed which can return the actual running time, fitness value, energy consumed in Joule and the MATLAB plot which shows the stage between each distance point, speed transition, fitness of the generation and number of feasible population in each generation.

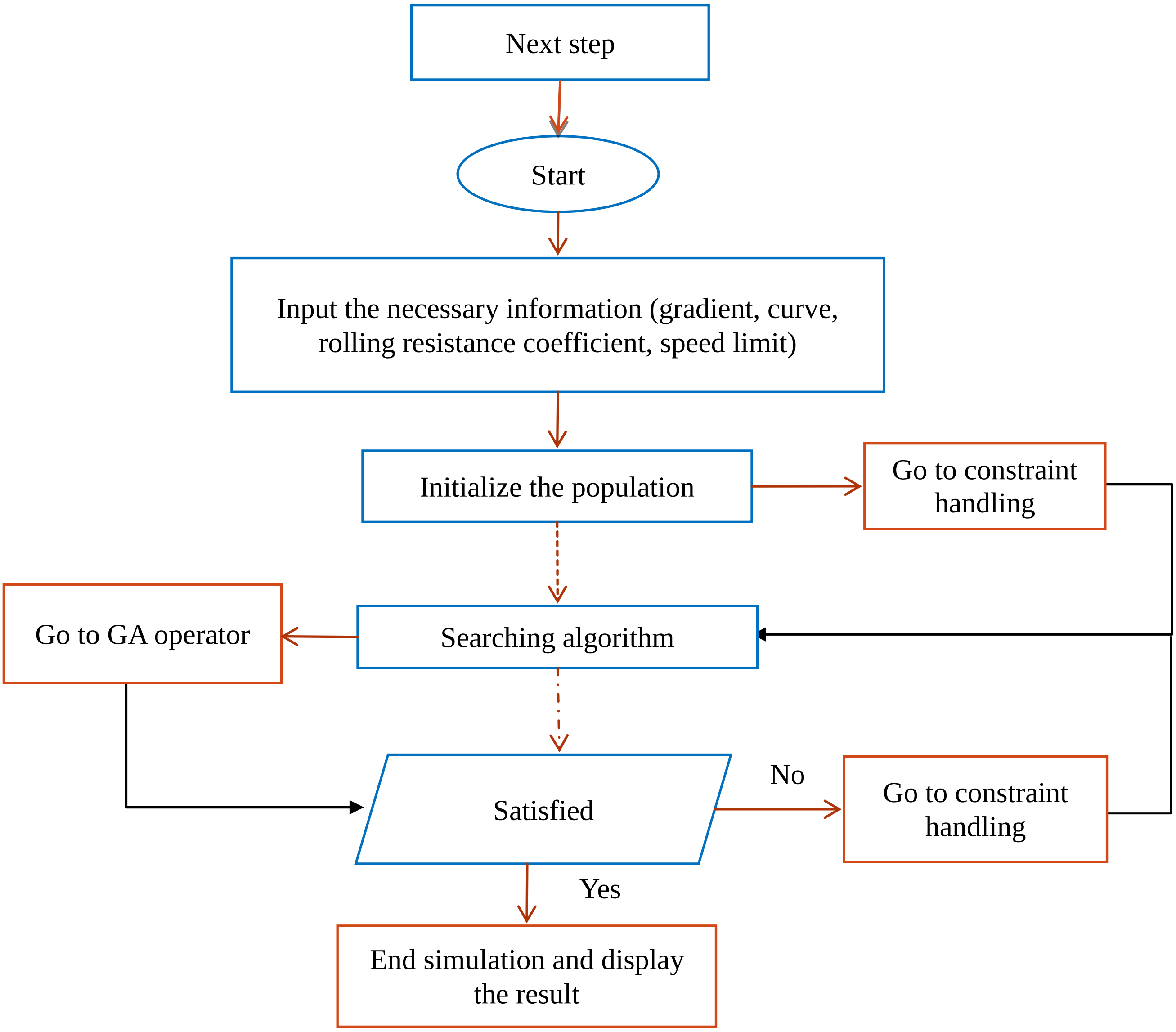


Figure 1-3 Genetic algorithm initial step

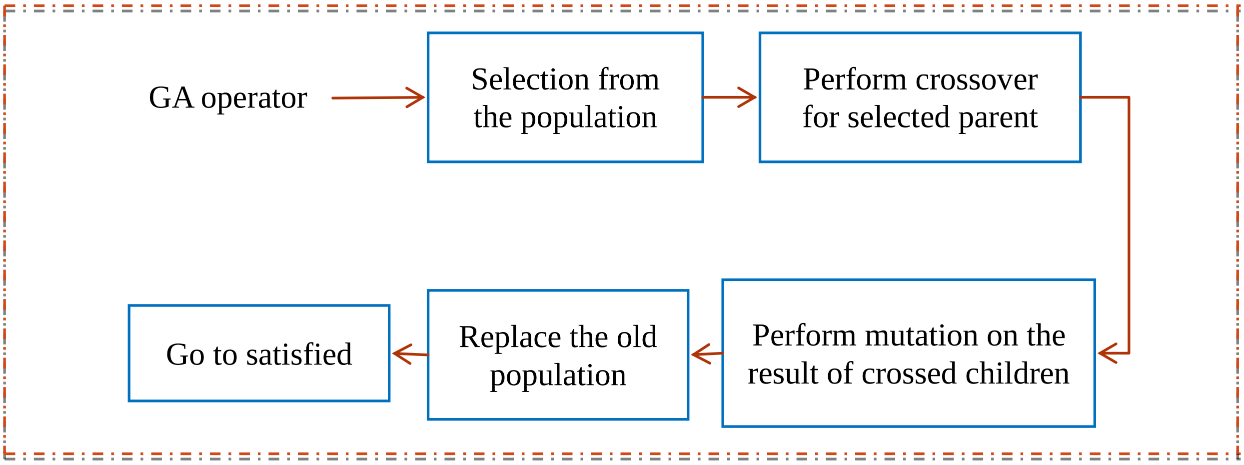


Figure 1-4 Genetic algorithm genetic operator

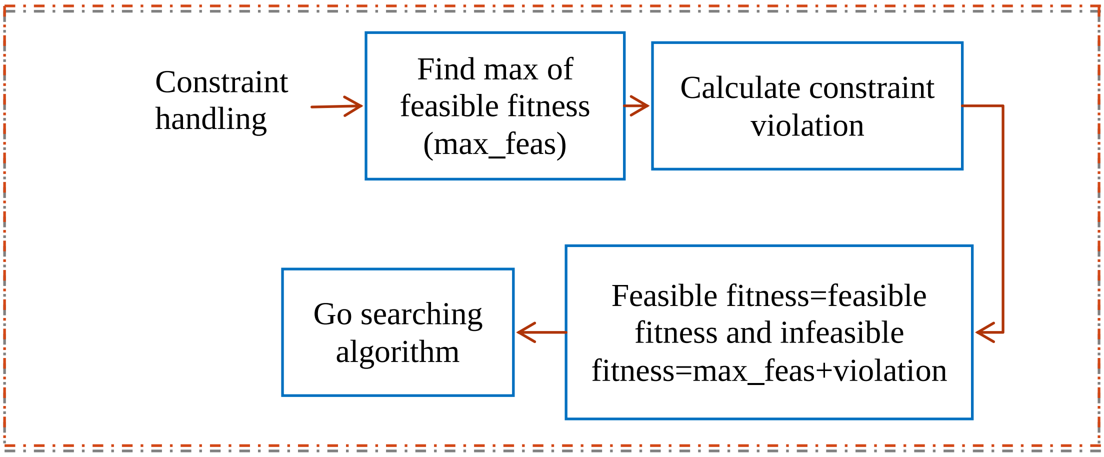


Figure 1-5 Genetic algorithm constraint handling

1.5. Scope of the thesis

This thesis develops means of generating speed profile for different traffic condition considering nearly all traction mechanics and track information for north to south line of Addis Ababa LRT. However, wind resistance and tunnel resistance is not considered. Tunnel resistance has an immense effect on train running resistance and wind resistance prevails for high speed train. Signaling and curve speed limit are not also considered. The maximum time required to complete each sections of the track is taken to be the maximum headway between trains. This is because of information deficiency from ERC.

1.6. Thesis organization

This thesis is organized in five chapters. The first chapter includes introduction which provides clear information about the background of the thesis work, statement of the problem, research method and limitation of the thesis.

Chapter two is about theoretical background and literature review. This section provides clear understanding of traction system which is about types of traction system, tractive effort, tractive resistance and energy consumption. Some information about railway development in Ethiopia and also detail information about the newly constructing Addis Ababa light railway project is discussed. Finally, other researchers work on the area of optimum train driving is presented.

Chapter three is about the development of mathematical modelling, how to solve the mathematical model, solving algorithm and selection of optimization technique for solving the model.

Chapter four is about MATLAB simulation. After the development of MATLAB code depending on the solving algorithms developed in chapter three, speed profile is generated for different traffic conditions. After having the simulation result, we discussed what the simulation result is to mean.

The final chapter is recommendation and conclusion. Here, we conclude our work based on the result obtained and discussed in chapter four. Further recommendation for the development of new model or improvement of the result in this thesis is suggested.

2. Theoretical background and literature review

2.1. Theoretical background

2.1.1. Traction system

2.1.1.1. Types of traction system

Traction means pulling or gripping of power. Traction system in general can be a system which doesn't use any electrical energy at any stage such as steam engine drive, internal combustion engine drive and a system which can use electrical energy at some stage such as diesel electric drive, battery electric drive and straight electric drive. Each of the drive system has its own advantage and disadvantage.

The three kind of traction system used in railway transportation system are steam power, diesel power and electric power. These traction systems can be compared by looking at their efficiency and relationship between weight and output power. This time, electric traction system is the one widely applied in the railway industry. This is because of environmental friendliness, high starting torque, easy speed controllability and many other reasons **[9]** and **[17]**.

2.1.1.2. Tractive effort

To accelerate or decelerate the train, tractive effort must be transferred between wheels and running surface of the rail through a friction force, called adhesion. Adhesion which determines tractive effort that can be applied before the wheels begin to slip is dependent on the weight per wheel. At startup, tractive effort is nearly constant but as speed increases further current in traction motor falls and hence so does the tractive effort. In general**, [18**] defines tractive effort as follows:

> *It is the tractive force at the locomotive driving wheels (drivers) at the rail that starts and moves tonnage up various grades. The maximum tractive force that can be developed at the rail is equal to the weight on drivers multiplied by the adhesion (coefficient of friction) of the wheels on the rail.*

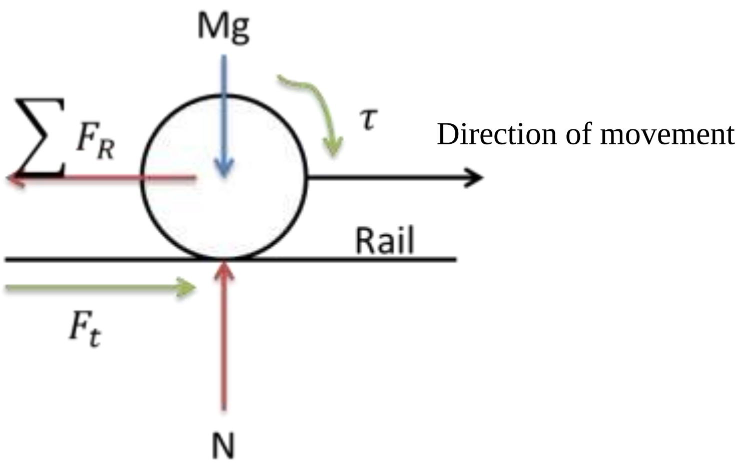


Figure 2-1 force interaction between wheel and rail

At constant velocity, $\tau = r * F_\mu$ (τ is torque and r is radius of the wheel) and tractive force is equivalent to frictional force $F_t = F_\mu$. The maximum tractive effort produced by traction motor is limited by the friction (adhesion).

$$F_{tmax} = \mu * M_{tot} * g \text{ --------------------------- } 2.1$$

Where:

μ adhesive coefficient

M_{tot} mass of the vehicle

g gravitational acceleration

Low value of μ results undesired wheel slip and wastage of energy. This situation also prevents increase of vehicle speed. Wheel rail adhesion analysis was conducted by some researchers and the following result were noticed: [**19**]

1. For a dry and clean wheel surface, the adhesion coefficient μ does not vary with vehicle speed;
2. For a track surface covered by oil, the adhesion coefficient μ can drop down to a very low level and this level is approximately maintained with increase in vehicle speed;
3. For water covering the wheel/rail contact surface, the adhesion coefficient μ reduces with increase in vehicle speed.

The desired acceleration can be obtained by providing sufficient tractive effort. On the other hand, the braking effort in addition to running resistance is the source of desired deceleration. Most of the time tractive effort is dependent on the available power of propelling installation, the desired speed on the section and the available adhesive force of the driven wheel **[20]**.

In the assumption of constant power tractive effort is simply **[9]:**

$$F_t = \frac{P}{V}, \text{where V is velocity and P is power} ------------2.2$$

In general, tractive effort versus speed curve of the given locomotive will take the following format:

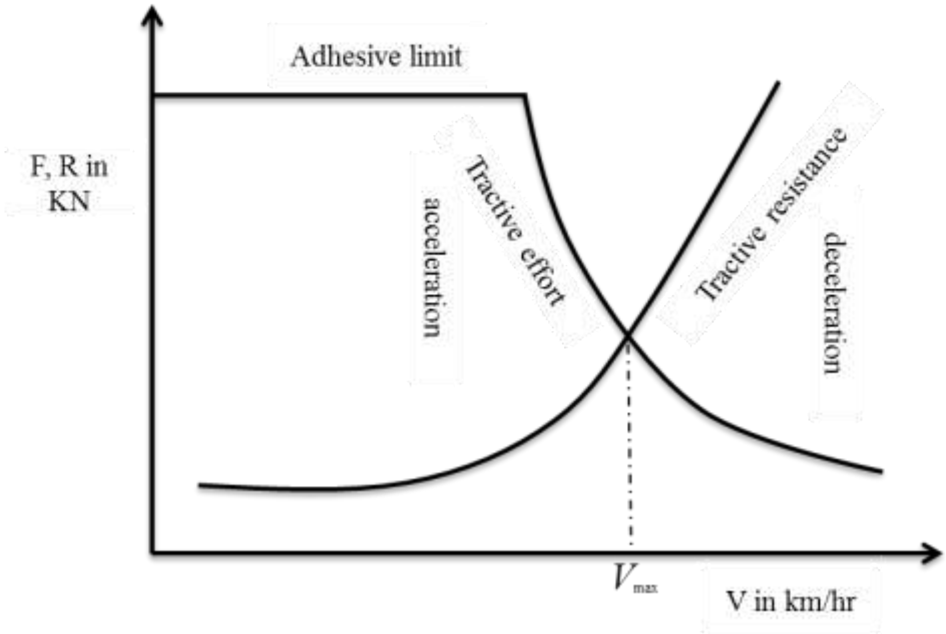


Figure 2-2 tractive effort vs speed curve

Magnitude of acceleration for maximum pull up force is around 02-1.2 m/s^2 and maximum deceleration can be substantially higher than starting acceleration because all axels are brake; but to protect passenger and loadings the maximum value is in the range of 0.5-1.3 m/s^2 [**20**]. For calculation of acceleration, deceleration and energy consumption; it is better to approximate tractive effort curve using piecewise polynomial function.

2.1.1.3. Train movement (tractive) resistance

Train movement along the rail suffers from various resistance forces which opposes longitudinal traverse of the train. Those resistances are complex in nature and can be grouped in to four: [**18], [19], [20], [21], [22], [23]** and **[24]**

a. Running (driving) resistance
b. Curve resistance
c. Gradient resistance
d. Tunnel resistance

Running resistance: is due to friction between wheel and rail, wave action which is rotation of axels, shafts, mechanical transmission and aerodynamics. This resistance can be broadly classified as rolling and aerodynamic resistance **[25]** and **[26]**

$$F_{ar} = \frac{1}{2} * \rho {V_{rel}}^2 A_{tr}\, C_{ar} - 2.3$$

F_{ar} is aerodynamic resistance in newton, V_{rel} relative velocity to wind in m/s, ρ air density in kg/cubic-meter, A_{tr} is train frontal area usually 10 meter-square, C_{ar} is air drag coefficient.

$$F_{rolling} = C_{rolling} * M_{tot} * g -2.4$$

$F_{rolling}$ is rolling resistance in newton, M_{tot} total mass of the train in kg, g is gravitational acceleration and $C_{rolling}$ rolling resistance coefficient. The rolling resistance coefficient is dependent on type of train, as well as the number of axles, speed and weight of the train.

$$C_{rolling} = C_0 + C_1 * \left(\frac{V}{V_0}\right) + C_2 \left(\frac{V}{V_0}\right)^2 -2.5$$

V_0 is the reference speed usually 100km/hr, which is 27.78 m/s.

Then, combining the rolling and aerodynamic resistance we can have the running resistance

$$F_{rr} = M_{tot} * g\left(C_0 + C_1 * \left(\frac{V}{V_0}\right) + C_2 \left(\frac{V}{V_0}\right)^2\right) + \frac{1}{2} * \rho {V_{rel}}^2 A_{tr} * C_{ar} - - - - - - - - 2.6$$

$$F_{rr} = (A + B * V + C * (V)^2) M_{tot} * g - 2.7$$

At higher speed values (over 100 Km/hr) aerodynamic resistance has a great effect on the running resistance and is dependent on frontal area of the train, its shape and speed. Predominant contributors for running resistance can be explained as follows**:-[25]**

A= from journal resistance, track resistance and rolling resistance

B= from flange friction, flange impact, rolling resistance of wheel/rail and wave action of the rail

C= from Head-end wind pressure, turbulence between cars, yaw angle of wind tunnels and rear drag

For passenger trains: [**27**]

$$\mathrm{F_{rr}} = 6.4\mathrm{M_{tone}} + 129 * \mathrm{AXEl} + 0.033 * \mathrm{M_{tone}}\mathrm{V} + 0.91\mathrm{V}^2 ------------ 2.8$$

Where: $\mathrm{M_{tone}}$ mass of the locomotive in tone, AXEl number of axel and V is speed of the train in mps.

Mass of the fully loaded ERC locomotive is 59.24 tones with six-axel and substituting this in the above equation;

$$\mathrm{F_{rr}} = 1153.136 + 1.95492\mathrm{V} + 0.91\mathrm{V}^2$$

$\mathrm{F_{rr}}$ is resistance in N, V is speed in m/s

Curve resistance: Vehicle running at a speed V with a radius of R experience centrifugal acceleration of $a = \frac{V^2}{R}$ and this acceleration is the reason for undesired effect like passenger discomfort, displacement of wagon load, risk of overturning and high lateral force on the track which increases curve resistance, risk of rail tilting and risk of lateral displacement of whole track **[19]**.

Empirical formula for curve resistance can be expressed by Rockl’s formula **[28]:**

$$\mathrm{Fcr} = \frac{6.5}{\mathrm{R} - 55} * \mathrm{g} * \mathrm{M_{tot}}(\mathrm{N}) \;\; \mathrm{R} > 300m -------------------2.9$$

$$\mathrm{Fcr} = \frac{5}{\mathrm{R} - 30} * \mathrm{g} * \mathrm{M_{tot}}(\mathrm{N}) \;\; \mathrm{R} < 300m ------------------- 2.10$$

It can also be expressed as the percentage of gradient resistance and depends on the type of gauge used, where D is degree of curvature **[29]**.

- Broad Gauge track 0.04% per degree: $0.0004M_{tot}Dg$
- Medium gauge track 0.03% per degree: $0.0003M_{tot}Dg$
- Narrow Gauge track 0.02 % per degree: $0.0002M_{tot}Dg$

This method allows to measure curve resistance at each point of the given curve. If L is length of the given curve and we want to know the degree of curvature at the track length S with in the long curve **[30]:**

$$\frac{S}{L} = \frac{\theta}{D} \quad ----------------------------- 2.11$$

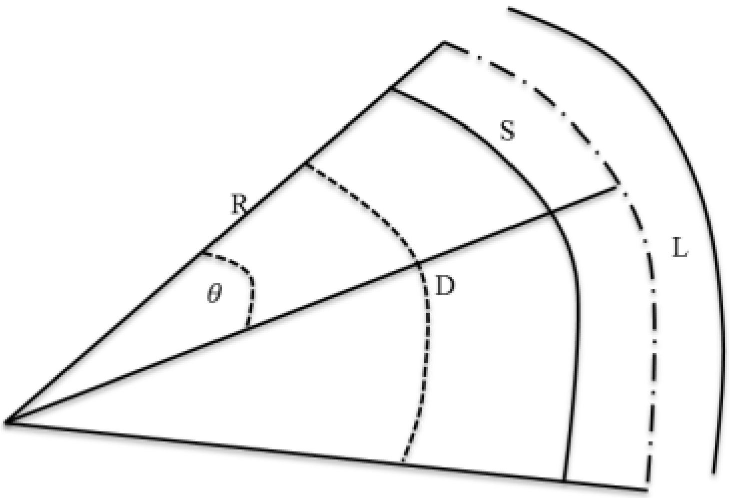


Figure 2-3 Curve resistance

<u>Gradient resistance</u>: is not dependent on speed of the train but weight. Location of gradient has important effect on overloading that can be permitted on the traction motor. If the ruling grade happens to be after a down gradient, it doesn't create any problem as momentum of the train takes it down the steep gradient. Other wises, temperature of the traction motor will rise. Location of the stop signal in relation to the gradient may also cause overloading of the traction motor **[20]**. Mathematically it can be represented as:

$$Fgr = M_{tota} * g * \sin(\theta) \quad --------------------- 2.12$$

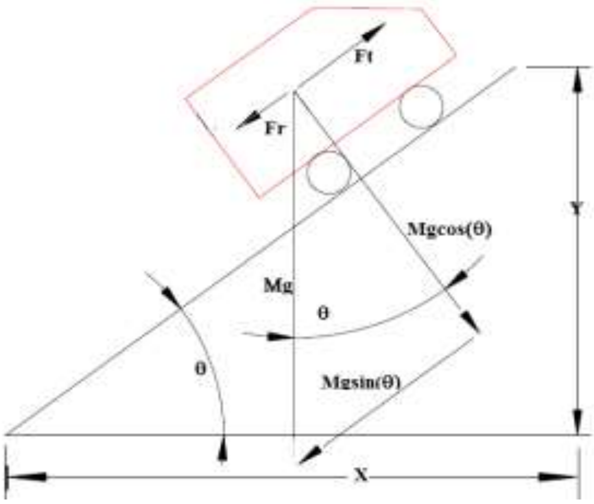


Figure 2-4 Gradient resistance

Tunnel resistance: is due to the presence of tunnels and sometimes it is called throat of railway transportation system. Especially with steam driven railway system, long tunnels cause a serious damage as the driver may fail to read the way side signal due to smoke and gases. Eventually such problems are solved by the existence of electric traction system [**30**]. Train length, tunnel length, ratio of cross-sectional area of the train to cross-sectional area of the tunnel and tunnel roughness are the major factors that affect tunnel resistance. Running resistance formula used for open air can be used for this situation with the introduction of a factor called tunnel factor T_f **[24]** and **[35].**

$$F_{tr} = A + B * V + T_f * C * V^2 \text{ --------------------- } 2.13$$

Where $T_f \geq 1$ is the ratio of tunnel drag to open air drag and highly dependent of the blocking factor which $\mathcal{B} = \frac{A_{tr}}{A_{tu}}$ where A_{tr} is cross-sectional area of the train and A_{tu} is cross-sectional area of the tunnel.

Tunnel length(feet)	Train type	$\mathcal{B}$=0.4	$\mathcal{B}$=0.65
2000	Passenger	4	6
5000	Passenger	6.3	12

Table 2-1 Canadian AREMA, C constant for different blocking factor

2.1.1.4. Energy consumption in traction system

There are two kind of energy consumption calculation, one is based on the total driving resistance and distance traveled by the train, and the other is based on total power consumed and time of run **[10**], **[23]** and **[33].**

- **Based on resistance and distance**

  Energy is calculated by integrating the total resistance (which is tractive effort) over the total distance that the train traveled.

$$E = \int_{x1}^{x2} F(v_x)_t * dx ------------------ 2.14$$

  Where: F_t is the tractive effort which balances the total resistance for a distance between x1 and x2, and is dependent on speed of the train.

- **Based on power and time**

  Total power required for the given journey at a given speed can be calculated as

$$p = V * F_t ------------------- 2.15$$

  Energy consumption of the train to complete the given journey is obtained by integrating the power required over the total time elapsed.

$$E = \int_{0}^{t} P * dt ----------------------- 2.16$$

  Where: V is speed of the train, F_t is tractive effort and P is the power.

The distance based approach is used for this thesis; this is due to the consideration of continuously varying gradient and curve.

2.1.2. Train mechanics

Mechanical equation which governs movement of the train is the foundation for energy and journey time calculation. Railway traction is one dimensional problem located on the longitudinal direction of the track and is governed by Newton second law of motion [**34].**

$$\sum F = M * a ------------------------- 2.17$$

$$\begin{cases} M * \dfrac{dV}{dt} = F_t - B_t - \sum F_R \\ V = \dfrac{dX}{dt} \\ \sum F_R = F_{rr} + F_{cr} + F_{gr} \end{cases} \qquad ---------------------- 2.18$$

2.1.3. Over view of Ethiopian railway (thesis area)

Railway transportation system had been used as a major freight and passenger transport to the eastern part of Ethiopia from 1917 to 2010. The system comes to existence during the reign of Emperor Menelik II and covers a total of 781km powered by diesel engine and jointly owned by Ethiopia and Djibouti. Great improvement of road network is prevailing in Ethiopia, but with limited connectivity, high cost of transportation and poor quality of service. The mobility need of the country population and the development of transportation system are far from compatibility. Therefore; the country is in need of modern, economic, time saving and long lasting transportation which will ease import-export system and result fast development. To this end, the government of Ethiopia has embarked on railway system. The main reasons for the renewed interest in the railway are environmental, economical, and safety related issues **[35]**.

This time, new railway line is under construction and Addis Ababa light railway is one of the newly constructing railway transport sector. It has two routes, which are the East-West and North-South. The main line of N-S line is 16.689km in full length, with the starting point mileage of YCK1+745 and terminal point mileage of YCK18+418. It has the subgrade section about 10.057km in length, the elevated section about 5.977km in length (including a common rail section 2.662km in length), and the underground section about 0.655km in length [**36**].There are 22 LRT stations and five of them are shared with East-West corridor. The minimum separation distance between stations is 435 meter; the maximum separation is 1972 meter and the average interval is 775 meter [**37**].

2.1.4. Tramcar used in Addis Ababa Light Railway Transit

6-axel double-articulated 70% low floor light tramcar is intended to be applied for Addis Ababa light railway transit (LRT) project. It allows bidirectional drive and it has the capacity of 286 passengers (64 seats and 6 passengers per m^2) with the life time of about 30 years. The design speed of the tramcar is 70km/hr [**36**]. The dead weight (which is the crucial component for generating speed profile) of the tramcar is 59.24 tone with length of 28.4 meter and width of 2.65 meter. At start up, tramcar has the chance to accelerate at maximum acceleration which is greater than or equal to $1.0m/s^2$, on the other hand, average braking deceleration is dependent on the type of braking applied. It has to be greater than or equal to $1.1m/s^2$ for normal braking and greater than or equal to $2.0m/s^2$ for emergency braking. The other characteristic of the tramcar, which can't be left unsaid, is jerk characteristic. Jerk is nothing but the rate of change of acceleration which has greater effect on passenger comfort. Generally, rate of change of acceleration or deceleration must not exceed $1.0m/s^3$ **[38]**.

2.2. Literature review

The best strategy of optimizing energy and journey time for train operation is generation of appropriate speed profile and forcing the system to follow it. Dynamic programming, mixed integer programming, genetic programming, and analytical methods can be applied to generate appropriate speed profile depending on different operational scenarios. When we say operational scenario, it is to mean, operation of the train in either saving energy, minimizing delay, minimizing energy or jointly minimizing journey time and energy or others modes. So many researches were conducted to achieve optimum train driving, but as far as my knowledge is concerned none of them can provide appropriate speed profile for different operational scenarios considering nearly all operational constraints. Here, six paper works on the issue of generating appropriate speed profile for optimum train operation are presented.

[39] The first paper work is minimization of fuel consumption on continuously varying gradient track. The journey must be completed within the allocated time. Therefore, the problem to be solved is minimization of fuel consumption which has direct relationship with the power developed by the locomotive subjected to time and distance constraints. Single level of braking acceleration is applied and speed decrease is always obtained from braking stage. If the locomotive consumes fuel, then the locomotive is in acceleration stage not in deceleration stage. One can argue that to consume less energy, it is possible to decrease speed of the train for positive resistance by applying relatively low tractive effort or braking effort and completing the journey on the allocated time. Generally, negative acceleration can be obtained from positive tractive effort, braking or coasting stage; and not mandatory only from braking. For the given discrete points, the paper tried to find appropriate speed with different acceleration. The model doesn't consider tractive effort limit and jerk constraint. Only energy minimization is performed and can work for one operational scenario. The developed mathematical model is solved by analytical method**.**

Jyh-Chering Jong et al conducted a research to develop train simulator for generating speed profile. Object oriented programming in C^{++} is the programming language used to write a code for train simulator. In the train simulator there are two kinds of models, namely, Shortest operational time and proper operational time. Shortest operational time is to mean minimizing the journey time among all speed profile. For this to happen, full traction is used to power acceleration, keep speed as close as speed limit and used full braking deceleration. And the other model which is proper operational time model gives a weight for energy and journey time. Depending on the operational scenario and interest of the company, appropriate weight value is given for time and energy. Speed, distance and tractive effort are the constraints considered for solving. The problem is solved for flat gradient not for continuously varying gradient, but can work for two operational scenarios **[40]**.

[**41]** The other research work aims at minimizing trip time under a certain energy constraint; the problem defined in this paper can be named as energy constraint operational problem. Since the paper is concerned about high speed train, only aero dynamic resistance is applied to calculate appropriate speed profile. Trip time objective function is not subjected to jerk (comfort) and tractive effort constraints which can influence passenger attraction and capacity of traction motor. Solving algorithm applied by this paper is analytical method. Energy, distance and speed are the constraints for this paper and works for one operational scenario.

[**42**] Propose a mechanism for optimization of speed profile in railway integrating energy saving. The paper developed an optimization model with three objective functions:

reduction of travel time, reduction of delay and minimization of energy consumption. It tried to compare energy consumption, travel time and delay by solving the developed model considering one objective, two objectives and three objective functions. Evolutionary algorithm is the method used to solve the developed model. It is applied for continuously varying gradient. Comfort and jerk constraint, tractive effort constraint are not considered. It works for three operational scenarios.

A Master thesis entitled *"Optimization Techniques Applied to Railway Systems"* was done *by Maialen Larranaga.* This thesis work focuses on energy minimization of railway system but not travel time. The paper used allocated travel time as one of the constraints for calculating the speed profile to obtain energy saving strategy. The mathematical model used by this paper is excellent, but for simulation reason only rolling resistance is considered. The final outcome of the simulation result allows saving about 24 % of energy compared to the applied speed profile during the time and works for one operational scenario [**43**].

The other paper work on the same issue is "*Study on optimal train movement for minimum energy consumption*" by *Panagiotis Gkortzas*. The thesis is primarily designed for energy consumption calculation and optimal train driving strategies for minimum energy consumption. Dynamic programming and Hamilton-Jacobi-Bellman equation (Bellman's backward approach) was used as a means of solving the optimization problem. As it is clearly defined on the paper, the thesis work has many drawbacks. Trains are expected to run at constant acceleration and from energy point of view such assumption will result high amount of energy consumption. The only resistance considered for the model is rolling resistance and other resistances are left for other researchers to consider. Comfort constraint is not also considered **[23].**

paper	Solving method	Operational scenario	Objective function	Applied Resistance (for solving)	Constraints	Acceleration and deceleration
[39]	Analytical method	One operational	Minimizing fuel consumption	Gradient resistance Rolling resistance	Time of run and speed	Constant deceleration Variable acceleration
[40]	Object oriented programing concept	Two operational	Minimize energy and Proper operational time	Rolling resistance	Speed, time and tractive effort	Constant acceleration and deceleration
[41]	Analytical method	One operational	Minimizing trip time	Aero dynamic resistance	Energy consumption Speed	Variable deceleration and acceleration
[42]	Evolutionary genetic algorithm	Three operational	Minimize energy, Time and delay	Gradient resistance	Time and energy depending on objective, Speed	>>
[43]	Dynamic programming	One operational	Minimizing energy	Rolling resistance	Comfort, jerk, acceleration, speed, energy, time, tractive effort	>>
[23]	Dynamic programming	One operational	Minimize energy	Rolling resistance	Speed, tractive effort and time	>>
Ours	Genetic algorithm with Debs' constraint handling technique	Three operational	Minimize energy, Minimize time and Proper operational	Rolling resistance, Gradient resistance, Curve resistance	Comfort, jerk, acceleration, speed, energy, time, tractive effort	>>

Proper operational is when the train operates in proper portion of time and energy optimization than in extreme cases.

Table 2-2 Comparison of thesis work

3. Mathematical modeling and solving algorithm

3.1. Mathematical modeling

The problem to be solved is minimization of energy consumption and/or journey time, and these problems can be solved either in a separate way, multi-objective or proper operational mechanisms. It is better to develop the model in distance approach rather than time approach as it allows adding continuously varying gradient resistance, tunnel resistance and curve resistance which are clearly distance dependent.

Constraints which affect train operation are the key aspect of mathematical modelling and each of those constraints is explained briefly as follows:-

- **Train dynamic equation:**

This is the longitudinal equation of motion for the train and can't be violated in any operational time **[33]**.

$$\begin{cases} M_{tot}\dfrac{dV}{dt} = F_t(x, V) - F_{rr} - F_{cr} - F_{gr} \\ V = \dfrac{dx}{dt} \end{cases} ------------ 3.1$$

- **Speed constraint**

It is due to train formation, curvature, downgrade, and switch, track strength, blocking signal, weather and temporary construction. A train travels along a rail line must obey all the speed limits at every location of the line. Consequently, the maximal allowable speed at a particular location is the minimum of all speed limits: [**40**]

$$V \leq \min(V_r, V_g, V_o, V_w, V_s, V_t) ---------------- 3.2$$

Where:-

V_t=speed limit due to train makeup/formation (km/hr)

V_r=curvature speed limit (km/hr)

V_g=downgrade speed limit (km/hr)

V_o=switch/turnout speed limit (km/hr)

V_w=speed limit due to track strength (km/hr)

V_s= speed limit due to blocking signal (km/hr)

- **Initial and final condition**

Initial and final speed of the locomotive must be zero and can't be violated. At initial time the locomotive must reach to the initial running point and at final time it must reach to the destination point.

$$V(0,0)=V(S,T)=0; \quad X(T)=S, X(0)=0$$

- **Jerk and acceleration (comfort) constraint**

Vehicle acceleration, either positive or negative (braking), and jerk can have a significant impact on the safety and comfort of passengers. The rate of change of acceleration is a jerk. This is also the third derivative of displacement and thus has units of m/s^3. The components that create discomfort and instability for vehicle passengers are excessive acceleration, which results in excessive external forces, and excessive jerk which, together with reaction time, results in excessive or deficient human reaction forces **[44].** For example, when designing a train and elevators, engineers will typically be required to keep the jerk less than 2 m/s^3 for passenger comfort. Then, an accepted criterion is that bounded longitudinal accelerations and jerks can guarantee a certain degree of comfort in longitudinal control, especially in Stop-and-go scenarios **[45].**

$$\mathrm{Jerk} = \frac{da}{dt} = \frac{d^3x}{dt^3} - - - - - - - - - - - - - - 3.3$$

According to **[38]**, to satisfy passenger comfort jerk of the train used in Ethiopia should obey the following rule.

$$|\mathrm{Jerk}| = \leq 1$$

- **Allowable tractive and braking effort**

This is related to vehicle characteristics and is mostly obtained from manufacturer data sheet. For example Ingetem Bilbao tramway, 70% low floor with maximum speed of 80km/hr with rated power of 98KW and maximum power of 167KW has the following allowable tractive and braking effort curve **[46].**

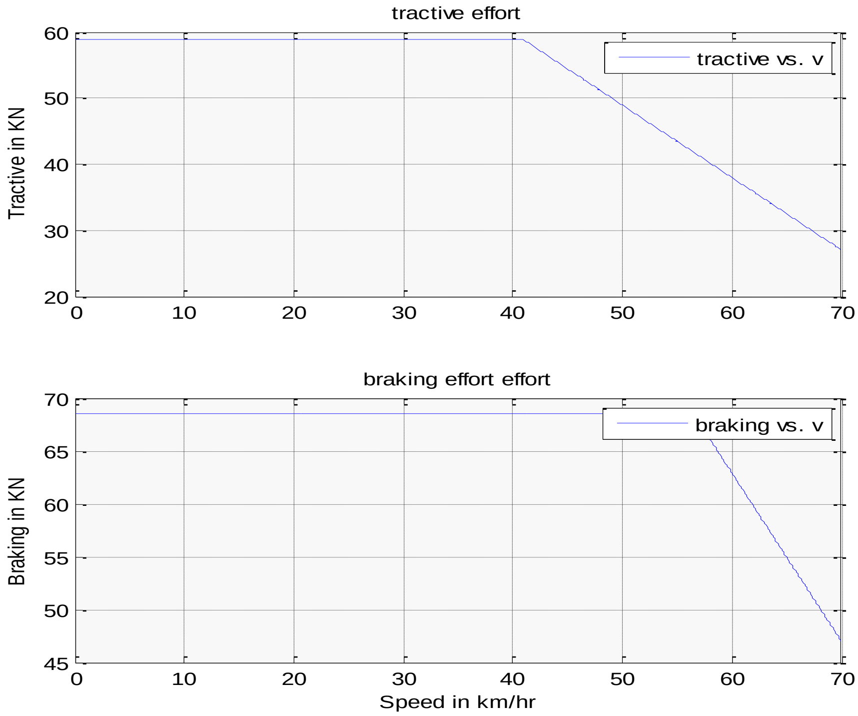


Figure 3-1 tractive and braking effort vs speed curve for Ingetem Bilbao tramway

Our locomotive has rated power of 130KW with rated speed of 1800rpm and maximum speed of 4377rpm [**38**]. To find maximum power, first we have to find the maximum torque assuming that maximum torque is obtained during maximum power. Speed of the motor for maximum torque must be lower than synchronous speed **[47].**

$$T_{rated} = \frac{P_{rated}}{\omega_{rated}}, T_{max} = \frac{P_{max}}{\omega} - - - - - - - - - 3.4$$

The margin of stability is expressed as the ratio of maximum torque to rated torque. This ratio is in the range of 2 to 3 **[47]**, but hard to obtain the ratio greater than 2.3 and it is good to have ratio of greater than or equal to 1.6 [**48].**

$$\frac{T_{max}}{T_{rated}} \geq 1.6 \underset{gives}{\Longrightarrow} T_{max} = 1.6T_{rated} - - - - - - - - - - - - 3.5$$

Substituting the expression

$$1.6\frac{P_{rated}}{\omega_{rated}} = \frac{P_{max}}{\omega} \underset{gives}{\Longrightarrow} 116 = \frac{P_{max}}{\omega} - - - - - - - - - - - - 3.6$$

Both values are unknown and from [**47**] the maximum torque happens relatively at the range of 20-30% slip.

The motor used by ERC train is six poles with rated operation frequency of 71Hz **[38]** and calculating operating speed of the train for maximum torque;

$$\omega = (1 - s) * \omega_{sy} \underset{\text{implies}}{\Longrightarrow} \omega_{sy} = 120 * \frac{71}{6} \underset{\text{imlies}}{\Longrightarrow} \omega_{sy} = 1420\text{rpm} - - - 3.7$$

Synchronous speed is lower than rated speed, to mean the motor always works in generator mode and which is not right, so the rolling stock spec given at [**38** has a problem and the pole is converted to 4 ($\omega_{sy} = 2130$). Taking the maximum torque happened at 25%:

$$\omega = .75 * 2130 = 1597.5 \text{ rpm} - - - - - 3.8$$

Substituting this to the original equation:

$$P_{max} = 116 * 1597.5 = 184.60\text{KW}$$

Now, based on the manufacturer data sheet and the calculation performed above, we formulated the tractive effort limit of the motor used by ERC. At base speed, tractive force times by base speed must be greater than or equal to maximum power delivered by the four motors.

$$F_t * V_{base} = 4 * 184.6\text{KW}$$

Knowing the base speed which is 41 km/hr:

$$F_t = 64.83\text{KN}$$

Therefore:

$$\begin{cases} F_t(V) \le 64.83 \text{ KN} & V \le 41\text{km/hr} \\ F_t(V) \le -1.097V + 109.777 \text{ KN} & V \ge 41\text{km/hr} \quad - - - -3.9 \end{cases}$$

By the same fashion, braking curve must be updated to the specification given in our locomotive. Braking resistance of the system is about 1.55 ohm at $20^{o}$and can range between +7/-5%. The data sheet that we have, braking resistance is 1.25 at low temperature and 1.5 at high temperature with rated power of 67KW. Since the expression is nearly the same, braking curve is used without any change.

$$B_t(V) \leq 68.47KN \qquad V \leq 56.51km/hr$$
$$B_t(V) \leq -1.592V + 158.408\ KN \qquad V \geq 56.51km/hr \quad ------3.10$$

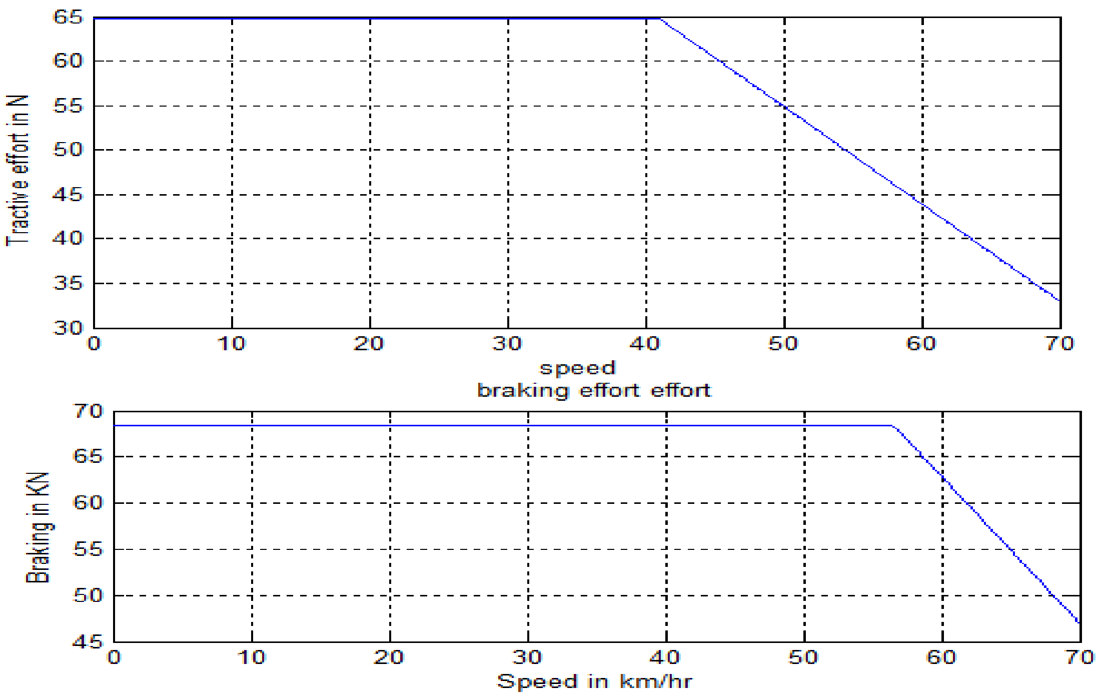


Figure 3-2 tractive and braking effort vs speed curve updated to Ethiopian tramcar

There are four train driving steps which are acceleration (high tractive effort and zero breaking effort), cruising (certain tractive power with zero acceleration and zero braking effort), coasting (energy saved due to this mode, both tractive and braking effort is zero) and finally braking (tractive effort is zero and braking effort probably with maximum braking effort). From this we can say that, tractive and braking effort product is equated to zero **[8]**.

$$F_t(V) * B_t(V) = 0 ----------------------3.11$$

- **Journey time**

The total time required to complete the given journey can't be violated because it results delay or timetable and scheduling disruption. It is composed of stopping time due to signal indication, station stoppage time and some extra time for correction.

- **Minimum distance between two trains**

The distance between two consecutive trains in the given section must be in the range specified.

After having the limiting factor, the general model for energy and journey time minimization (Based on **[43]**, we formulated our model to meet our requirement which is, the original model is developed for energy optimization but we also formulated for journey time and proper operational model. As a result of this, based on the objective function applied, the number of constraints considered varies.)

$$\min \begin{array}{l} E = \int_{x_{st}}^{x_{en}} F_t * dx \\ T_{run} = \int_{x_{st}}^{x_{en}} \frac{1}{V} * dx \end{array} \text{------------------------}3.12$$

Subjected to:

$$\frac{dx}{dt} = V \text{------------------------------}3.13$$

$$M_{tot}\frac{dV}{dt} = F_t - B_t - M_{tot}g(A + BV + CT_fV^2) - 0.0004M_{tot}Dg - M_{tot}g * \sin(\theta) - 3.14$$

$$F_t * B_t = 0 \text{------------------------------}3.15$$

$$\begin{cases} F_t(V) \le 64.83KN & V \le 41\ km/hr \\ F_t(V) \le -1.097V + 109777\ \ KN & V \ge 41km/hr \\ B_t(V) \le 68.47KN & V \le 56.51km/hr \\ B_t(V) \le -1.592V + 158.408\ KN & V \ge 56.51km/hr \end{cases} \text{----------}\ 3.16$$

$$\left|\frac{d^2V}{dt^2}\right| \le 1.0 \text{------------------------------}3.17$$

$$\begin{cases} \frac{dV}{dt} \le 1.0, & \text{if acceleration} \\ \left|\frac{dV}{dt}\right| \le 1.1, & \text{if normal braking} \\ \left|\frac{dV}{dt}\right| \le 2.0, & \text{if emergency braking} \end{cases} \text{---------------------}\ 3.18$$

$$V \le \min(V_r, V_g, V_o, V_w, V_s, V_t) \text{------------------------}3.19$$

$$T_{run} \leq T \text{ --------------------------------- } 3.20$$

$$E \leq E_{max} \text{ --------------------------------- } 3.21$$

3.1.1. How to solve the model

Solving the mathematical model require a means of incorporating continuously varying track information. For this to happen, the total section must be divided in to number of discrete distance point based on the track profile information distribution. There is no need of dividing the station distance equally; we can divide in such a way that all the track profile information can be incorporated for modelling, analysis and simulation. Generally, the total distance between two stations is divided in to N steps with different step size.

Figure 3-3 discretized distance point

- **Differential equation approximation**

  Based on the information presented on **[49]**, **[50]**, **[51]**, **[52]** and **[53]** we can approximate differential equations accordingly.

  - **Acceleration approximation**

Using forward Euler formula:

$$\frac{dV}{dt} = \frac{\Delta V}{\Delta t} = \frac{(V_{i+1} - V_i)}{t_i} \text{ ----------------- } 3.22$$

Using classical derivative formula:

$$V = \frac{dx}{dt} = \frac{\Delta x}{\Delta t} \rightarrow V_{av} = \frac{x_{i+1} - x_i}{t_i} \rightarrow t_i = 2 * \frac{x_{i+1} - x_i}{V_{i+1} + V_i} \text{ -- } 3.23$$

Substituting Δt in equation 3.22, we have:

$$\frac{dV}{dt} = \frac{(V_{i+1}^2 - V_i^2)}{2(x_{i+1} - x_i)} \text{ ------------------- } 3.24$$

- ### Rate of change of acceleration approximation

The rate of change of acceleration can be approximated by different approximation formula like Tylor series expansion, Euler formula and other approximation method.

Tylor series approximation method:

$$\begin{cases} s(x) = s(x_i) + \dot{s}(x_i) * (x - x_i) + \ddot{s}(x_i) * \dfrac{(x - x_i)^2}{2} + \text{higher order} \\ \ddot{s}(x_i) = 2 * \dfrac{s(x) - \left(s(x_i) + \dot{s}(x_i) * (x - x_i)\right)}{(x - x_i)^2} \\ \ddot{s}(x_i) = 2 * \dfrac{s(x) - s(x_i)}{(x - x_i)^2} - \dfrac{2 * \dot{s}(x_i)}{x - x_i} \end{cases} - 3.25$$

Now substituting the necessary information in the above equation:

$$\begin{cases} \dfrac{d^2V}{dt^2} = \ddot{V}_i = 2 * \dfrac{V_{i+1} - V_i}{(t_{i+1} - t_i)^2} - \dfrac{2 * \dot{V}_i}{(t_{i+1} - t_i)} \\ \ddot{V}_i = 2 * \dfrac{V_{i+1} - V_i}{(t_{i+1} - t_i)^2} - \dfrac{(V_{i+1}^2 - V_i^2)}{(x_{i+1} - x_i)} * \dfrac{1}{(t_{i+1} - t_i)} \\ t_{i+1} = \dfrac{2 * (x_{i+2} - x_{i+1})}{V_{i+2} - V_{i+1}} \end{cases} --- 3.26$$

Euler approximation method:

$$\frac{d^2V}{dt^2} = \frac{\Delta(\Delta V)}{(\Delta t)^2} = \frac{(V_{i+1} - 2 * V_i + V_{i-1})}{(x_{i+1} - x_i)^2} * V_i^2 --------- 3.27$$

Though Taylor approximation gives more accurate result, due to simplicity and easy applicability for solving, Euler approximation method is used.

- ## Objective function approximation
  - ### Energy consumption approximation

Now coming to energy consumption approximation, previously we said that energy consumption of the train is dependent on the distance traveled and tractive effort. The integration can be evaluated numerically using different methods like rectangular rule, trapezoidal rule, Simpson rule and others [**52**].

Taking trapezoidal rule we can approximate energy consumed by the train over the given distance range:

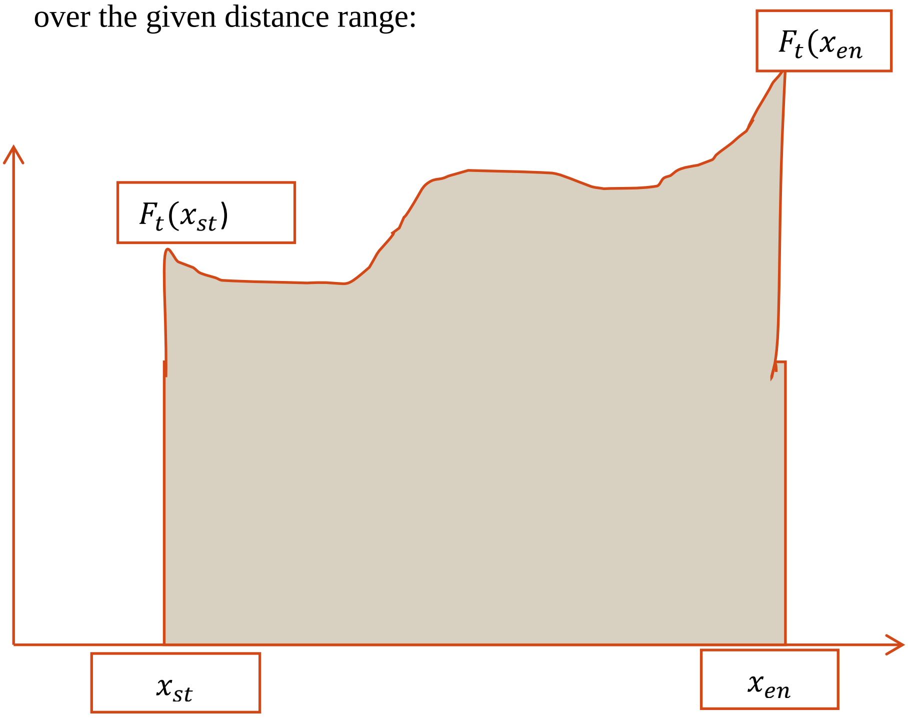


Figure 3-4 Trapezoidal approximation of integral

$$E = \int_{x_{st}}^{x_{en}} F_t * dx$$

$$= \frac{(x_{en} - x_{st}) * \left(F_t(x_{en}) + F_t(x_{st})\right)}{2} ----------- 3.29$$

For N discrete level between the start and the end point, total energy consumed is the sum of energy consumed at each interval and tractive effort is dependent on speed of the train at the given interval.

$$\left\{ E = \sum_{i=2}^{i=N} \frac{\left(F_t(x_i, V_i) + F_t(x_{i-1}, V_{i-1})\right) * \Delta x_{i-1}}{2} \right. ------- 3.30$$

- **Journey time approximation:**

Suppose Δt is the time required to complete a distance between x_i and x_{i+1}; V is the average velocity of speed at i and i + 1, then

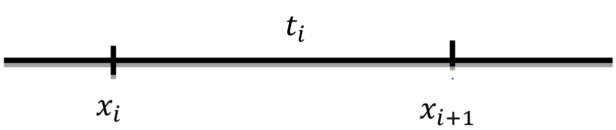


Figure 3-5 Time calculation between discrete points

$$\Delta t = t_i = 2 * \frac{x_{i+1} - x_i}{V_{i+1} + V_i} ------------------------3.31$$

The total time taken to complete the entire journey is the summation of t_i's.

$$T_{run} = \sum_{i=1}^{N-1} t_i = \sum_{i=1}^{N-1} 2 * \frac{x_{i+1} - x_i}{V_{i+1} + V_i} -----------------3.32$$

Having the above approximation, we can formulate the optimization problems. This thesis interested in developing and solving three models namely minimum energy model, minimum journey time model and proper operational model subjected to complex constraints.

1. Minimizing energy for a fixed journey time T

This is for the best energy consumption without violating the specified journey time.

$$\min \quad \sum_{i=2}^{i=N} \frac{\left(F_t(x_i, V_i) + F_t\left(x_{i-1,}V_{i-1}\right)\right) * (x_i - x_{i-1})}{2}$$

Subjected to: =

$$M_{tot} \frac{(V_{i+1}^2 - V_i^2)}{2(x_{i+1} - x_i)}$$
$$= F_t(V_i) - B_t(V_i) - M_{tot}g\left(A + BV_i + C * T_f * V_i^2\right) - (0.0004DgM_{tot}) - (M_{tot}g \sin(\theta))$$

$$F_t(x_i, V_i) * B_t(x_i, V_i) = 0$$

$$\begin{cases} F_t\left((x_i, V_i)\right) \le 64.83\ KN & V_i \le 41 km/hr \\ F_t\left((x_i, V_i)\right) \le -1.097V_i + 109.777\ KN & V_i \ge 41 km/hr \\ B_t\left((x_i, V_i)\right) \le 68.47KN & V_i \le 56.51 km/hr \\ B_t\left((x_i, V_i)\right) \le -1.592V_i + 158.408\ KN & V_i \ge 56.51 km/hr \end{cases}$$

$$\left| \frac{(V_{i+1} - 2 * V_i + V_{i-1})}{(x_{i+1} - x_i)^2} * V_i^2 \right| \le 1.0$$

$$\begin{cases} \dfrac{(V_{i+1}^{2} - V_i^2)}{2 * (x_{i+1} - x_i)} \leq 1.0, & \text{if acceleration} \\ \left|\dfrac{(V_{i+1}^{2} - V_i^2)}{2 * (x_{i+1} - x_i)}\right| \leq 1.1, & \text{if normal braking} \\ \left|\dfrac{(V_{i+1}^{2} - V_i^2)}{2 * (x_{i+1} - x_i)}\right| \leq 2.0, & \text{if emergency braking} \end{cases}$$

$$V_i \leq \min\left(V_r, V_g, V_o, V_w, V_s, V_t\right)$$

$$\sum_{i=1}^{N-1} 2 * \frac{x_{i+1} - x_i}{V_{i+1} + V_i} \leq T$$

2. Minimizing journey time for the given energy consumption

It is better to modify the objective function for total journey time because we don't need to create any disruption on the schedule. For this matter, we converted the objective function which is total time minimization to percentage time minimization.

$$\min \frac{\sum_{i=1}^{N-1} 2 * \frac{x_{i+1} - x_i}{V_{i+1} + V_i} - T}{T}$$

Subjected to: =

$$M_{tot} \frac{(V_{i+1}^{2} - V_i^2)}{2(x_{i+1} - x_i)} = F_t(V_i) - B_t(V_i) - M_{tot} g(A + BV_i + C * T_f * V_i^2) - (0.0004DgM_{tot}) - (M_{tot} g \sin(\theta))$$

$$F_t(x_i, V_i) * B_t(x_i, V_i) = 0$$

$$F_t\left((x_i, V_i)\right) \leq 64.83 \text{ KN} \qquad V_i \leq 41 \text{km/hr}$$
$$F_t\left((x_i, V_i)\right) \leq -1.097V_i + 109.777 \text{ KN} \qquad V_i \geq 41 \text{km/hr}$$
$$B_t\left((x_i, V_i)\right) \leq 68.47 \text{KN} \qquad V_i \leq 56.51 \text{km/hr}$$
$$B_t\left((x_i, V_i)\right) \leq -1.592V_i + 158.408 \text{ KN} \qquad V_i \geq 56.51 \text{km/hr}$$

$$\left|\frac{(V_{i+1} - 2 * V_i + V_{i-1})}{(x_{i+1} - x_i)^2} * V_i^{2}\right| \leq 1.0$$

$$\begin{cases} \dfrac{(V_{i+1}^2 - V_i^2)}{2*(x_{i+1}-x_i)} \le 1.0, & \text{if acceleration} \\ \left|\dfrac{(V_{i+1}^2 - V_i^2)}{2*(x_{i+1}-x_i)}\right| \le 1.1, & \text{if normal braking} \\ \left|\dfrac{(V_{i+1}^2 - V_i^2)}{2*(x_{i+1}-x_i)}\right| \le 2.0, & \text{if emergency braking} \end{cases}$$

$$V_i \le \min\left(V_r, V_g, V_o, V_w, V_s, V_t\right)$$

$$\sum_{i=2}^{i=N} \frac{\left(F_t(x_i, V_i) + F_t\left(x_{i-1}, V_{i-1}\right)\right) * (x_i - x_{i-1})}{2} \le E_{max}$$

3. Proper operational model

In such kind of model, all of the components to be optimized need to have a percentage by how much to be optimized. Normally, the objective function takes the following form **[40]**:

$$\min K_1 f(x) + K_2 g(x) ---------------------------3.33$$

Where K_1 and K_2 are percentage of the respective function to be optimized and summed to one. One fundamental concept in such kind of system is that each of the functions to be optimized must be in the same range.

For our system, energy and journey time must be in the same range and we already converted the journey time to unit less percentage value and energy will be converted to the same range by:

$$\min \left\{ \frac{\sum_{i=2}^{i=N} \frac{\left(F_t(x_i,V_i)+F_t\left(x_{i-1},V_{i-1}\right)\right)*(x_i - x_{i1})}{2} - E_{max}}{E_{max}} \right. ----- 3.34$$

The problem to be solved is:

$$\min \left\{ K_1 \frac{\sum_{i=2}^{i=N} \frac{\left(F_t(x_i,V_i)+F_t\left(x_{i-1},V_{i-1}\right)\right)*(x_i - x_{i1})}{2} - E_{max}}{E_{max}} + K_2 \frac{\sum_{i=1}^{N-1} 2 * \frac{x_{i+1}-x_i}{V_{i+1}+V_i} - T}{T} \right.$$

Subjected to:

$$M_{tot} * \frac{(V_{i+1}^{2} - V_i^2)}{2 * (x_{i+1} - x_i)}$$
$$= F_t(V_i) - B_t(V_i) - M_{tot}g(A + B * V_i + C * T_f * V_i^2) - (0.0004DgM) - (M_{tot}g\sin(\theta))$$

$$F_t(x_i, V_i) * B_t(x_i, V_i) = 0$$

$$F_t\big((x_i, V_i)\big) \le 64.83 \text{ KN} \qquad V_i \le 41\text{km/hr}$$
$$F_t\big((x_i, V_i)\big) \le -1.097V_i + 109.777 \text{ KN} \qquad V_i \ge 41\text{km/hr}$$
$$B_t\big((x_i, V_i)\big) \le 68.47\text{KN} \qquad V_i \le 56.51\text{km/hr}$$
$$B_t\big((x_i, V_i)\big) \le -1.592V_i + 158.408 \text{ KN} \qquad V_i \ge 56.51\text{km/hr}$$

$$\left|\frac{(V_{i+1} - 2 * V_i + V_{i-1})}{(x_{i+1} - x_i)^2} * V_i^{\,2}\right| \le 1.0$$

$$\begin{cases} \dfrac{(V_{i+1}^{2} - V_i^2)}{2 * (x_{i+1} - x_i)} \le 1.0, & \text{if acceleration} \\ \left|\dfrac{(V_{i+1}^{2} - V_i^2)}{2 * (x_{i+1} - x_i)}\right| \le 1.1, & \text{if normal braking} \\ \left|\dfrac{(V_{i+1}^{2} - V_i^2)}{2 * (x_{i+1} - x_i)}\right| \le 2.0, & \text{if emergency braking} \end{cases}$$

$$V_i \le \min\big(V_r, V_g, V_o, V_w, V_s, V_t\big)$$

3.2. Selecting optimization technique

Optimization techniques applied in railway system ranges from analytical solution through numerical method to dynamic programming and evolutionary algorithm. For this thesis, optimization technique used is one of evolutionary algorithm which is genetic algorithm (GA). Even though genetic algorithm is slow in finding optimum solution for constrained optimization problem; the search space used by this method allow to search efficiently large space and genetic operators adds additional value in protecting premature convergence in addition to ending up at the local optimum value . Generally genetic algorithm is an optimization and search technique using the principle of genetics and natural selection [**54**]. The number of initial population and how the population can be generated is the big challenge of this method. Too large initial population size will result unnecessary computational effort and too small initial population will narrow the search space. In general, initial population size of 30 is known to be quite good for different kind of optimization problems [**55**].

Advantages of a GA includes: [**56**]

- Work with continuous or discrete variables,
- Doesn't require derivative information,
- Simultaneously searches from a wide sampling of the cost surface,
- Deals with a large number of variables,
- Provides a list of optimum variables, not just a single solution,
- May encode the variables so that the optimization is done with the encoded variables,
- Works with numerically generated data, experimental data, or analytical functions
- GA is conceptually simple and relatively easy to implement.

GA is applicable for any kind of optimization problems, but the difficulty comes when we apply it for constrained optimization problems. There are many parameters and settings that can be implemented differently in various problems:

- How to create chromosomes and what type of encoding chooses?
- How to select parents for crossover in the hope that the better parents will produce better offspring?
- How to define crossover and mutation?, the two basic operators of GA
- How to convert constrained optimization problem in to unconstrained optimization problem?

To answer the above questions, different researches were conducted and different kinds of constraint handling techniques were developed. One can have a look at [**57]**, [**58], [59], [60]** references to have detail knowhow about constraint handling techniques. For this thesis, Debs' constraint handling technique with tournament selection is used. The method uses information from the current population to tune the penalty parameter. In reality the system doesn't use any penalty parameter rather it takes violation of the current population and maximum of feasible fitness in current population. If the population is feasible, then the fitness function can be the original objective functional value and for infeasible population the fitness will be the sum worst feasible fitness and violation of the given population [**61], [62], [63]**.

$$\widehat{F(\vec{x})} = \begin{cases} f(\vec{x}) & \text{feasible} \\ fmax(\vec{x}) + \sum_{j=1}^{m} c_j(\vec{x}) & \text{else} \end{cases} \quad ------------------- 3.35$$

There are for principal operations in a genetic algorithm; selection, crossover, mutation and replacement operation **[64]**.

- **Selection operator: -** is the means of selecting best chromosome from the current population to generate new offspring. There are different methods in selecting the best chromosomes:**[65]** and **[66]**
    - Roulette wheel selection,
    - Rank selection
    - Tournament selection

  Elitism plays an important role in not losing the best chromosome from the already existed population due to crossover and mutation, on which, there is a great probability of losing the best chromosome. The one with best fitness value will evolve to the next generation without any change.

- **Crossover operator: -** generates offspring from two chosen individuals in the population by exchanging some bits from parents. Offspring thus inherit some characteristics from each parent. Arithmetic crossover is used for this thesis **[67].**
- **Mutation operator: -** generates offspring by randomly changing one or several bits in an individual. Mutation prevents local searches of the search space and increases the probability of finding global optima. Non uniform mutation is used for this thesis. This mutation is dependent on each generation and maximum generation which allows creating diversity on the population. Mathematically it is represented as:**[68]**

$$x_{ij} = \begin{cases} x_{ij} + \left(ub_j - x_{ij}\right) * \gamma(t) & \alpha < 0.5 \\ x_{ij} - \left(x_{ij} + lb_j\right) * \gamma(t) & \alpha \geq 0.5 \end{cases} \quad ------------------3.36$$

With

$$\gamma(t) = \alpha 1 * \left(1 - \frac{t}{tmax}\right)^{b} \quad ------------------------ 3.37$$

$\alpha 1, \alpha$ is random at $[0\ 1]$

b is user defined constant parameter

t, tmax is current generation and maximum geneation

Or: **[69]**

$$x_{ij} = \begin{cases} x_{ij} + \Delta\left(t, ub_j - x_{ij}\right) & \alpha = 0 \\ x_{ij} - \Delta\left(t, x_{ij} - lb_j\right) & \alpha = 1 \end{cases} ----------------------3.38$$

With

$$\Delta(t, y) = y * (1 - r^{\left(1 - \frac{t}{tmax}\right)^{b}}) ------------------------3.39$$

r is random number [0 1]

This functional value approaches to zero as generation advances and takes the value between 0 and y.

The other operator of genetic algorithm is replacement which is creation of the next generation based on crossover children, elite children and old population.

3.3. Solving algorithms

In the presence of continuous gradient, there is a possibility of having negative or positive driving resistance. Depending on the sign of total resistance and sign of acceleration in addition to sign of tractive effort we can declare the following operations.

i. **Positive gradient**

For positive gradient, total resistance of the train is positive and we have the following driving scheme.

I. **Positive acceleration**: - this is definitely acceleration stage and consumes energy.

II. **Zero acceleration**: since resistance is positive, train consumes energy to preserve initial speed and it is called cruising stage.

III. **Negative acceleration**: for this case, we can have different situations

A. **Positive tractive effort**: train consumes energy to decrease speed when target speed is lower than initial speed.

B. **Zero tractive effort**: it is clearly coasting stage.

C. **Tractive is less than zero**: braking stage

ii. **Negative gradient**

For negative gradient, train movement resistance can be either negative or positive. The positive resistance case is similar to positive gradient resistance case discussed above. So, let’s deal with negative total resistance, due to the negativity of total resistance normally train will accelerate without the application of tractive effort.

I. **Negative acceleration:** it is clearly brake stage

II. **Zero acceleration**: it is clearly cruising stage and tractive effort is negative, so no energy is consumed.

III. **Positive acceleration**: Here, we have the following driving scheme:

A. **Positive tractive effort**: this is definitely acceleration stage and consumes power.

B. **Negative tractive effort**: velocity increase due to the total resistance but braking applied to control speed increase of the train.

C. **Zero tractive effort**: it is coasting stage

Steps of driving are classified based on the sign of acceleration and tractive force. Generally, our step determination is as follows:

Steps	Acceleration	Tractive effort	Braking effort
1	Positive	Exist	Zero
1.5	Negative	Exist	Zero
2	Zero	Depends on total resistance	Depends on total resistance
3	Negative or positive	Zero	Zero
3.5	Positive	Zero	Exist
4	Negative	Zero	Exist

Table 3-1 Stage of driving

After having the above steps handling technique, based on the principle we defined in the previous section which is genetic algorithm with Debs' penalty constraint handling method pseudo code for MATLAB application is developed. It has the following core parts:

- Initial population generation
- Creating new children for next generation
- Fitness evaluation

- **Initialization population generation**

This is the first step of the algorithm and all initial population must not violate initial, final and boundary condition constraint. At zero distance, speed of the train must be zero and at the final destination too. Speed of the train must be in the range of 0 -70km/hr for main line, for level crossing speed range must be in the range of 0-50km/hr and 0-25km/hr for train passing through the lateral switch position **[70].** In the middle of the track section, speed of the train can't be zero. Since we are using real coded genetic algorithm, each of the chromosomes in the population is converted to the real value by the formula:

$$populations = initial + (final - initial) * \sum_{i=1}^{bit} chromosom(i) * \frac{2^{bit-i}}{2^{bit}-1} \qquad 3.40$$

```
function [speed_mps, speed_binary]=correct_generate(nbit, numpopulation, numparameter)
Speed_binary=round(rand(numpopulation, nbit));
Speed_mps=zeros(numpopulation,numparameter);
For i=1:numpopuation
      For j=2:numparameter-1
            Speed_mps(i,j)=convert binary to real value;
            while speed_mps(i,j)==0
                  force the speed to be none zero;
             end while
       end For
       If speed_mps(i,1)~=0
          force to zero
        end If
        If speed_mps(i,numparameter)~=0
            force to zero;
         end If
end For
```

Speed_mps= $\text{speed\_mps} \Big/ 2^{\frac{\text{nbit}}{\text{numparameter}}-1}$ *700/36;

Figure 3-6Pseudo code for generating initial population with speed limit of 70km/hr

➢ **Next population generation**

This part includes selection, crossover, mutation and replacement strategy. Appropriate algorithm must be developed to perform these all activities.

a. Selection operator

Selection return elite children evolved to the next generation without any change and the two parrents prepared for mating. Elite childrens are allowed to be selected to feel the mating pool and two populations selected from the old population are compared to fill the mating pool. Selection principle followes the following rule as depicted in Debs' method **[61]**:

- Two chromosoms are feasible, one with lower fitness is selected
- One feasible and the other infeasible, feasible chromosome selected
- Both chromosomes are infeasible, select the one wth lower violation

Here, 'feasible_population' is member of feasible population,'numelite' is number of elite children and 'num_feasible' is number of feasible population from the current population.

```
function [elite, parent1, parent2]=selection(population, numelite)
Call fitness function ; %%penality, feasible_population, fitness
If feasible_population
    If numelite<=num_feasible
       [feasible_fit, index]=sort fitness of feasible population;
       For i=1:numelite
           elite(i,:)=feasible_population(index(i),:);
       end For
  Else
      elite=feasible_population;
      [fit, index]=sort fitness;
      For i=num_feasible+1:numelite
           elite(i,:)=population(index(i),:);  % not member of feasible population
       end For
   end If
For i=1:numchild
       select1=ceil(rand*numpop);
       select2=ceil(rand*numpop);
       feasi1=searching(population(select1,:),  feasible_population);
       feasi2=searching(population(select2,:),  feasible_population);
       If feasi1==true&&feasi2==true
            If fitness(select1)>=fitness(select2)
                parent1(i,:)=population(select2,:);
           Else
                parent1(i,:)=population(select2,:);
            end If
     Elseif feasi1==false&&feasi2=true
             parent1(i,:)=population(select2,:);
     Elseif feasi2==false&&feasi1=true
             parent1(i,:)=population(select2,:);
      Else
           If penalty(select1)>=penalty(select2)
               parent1(i,:)=population(select2,:);
           Else
               parent1(i,:)=population(select1,:);
           end If
     end IF
% PARENT TWO SELECTION
Do what is done for parent1
End For
Else
 parent1 and parent2 is null
End If
```

Figure 3-7 Pseudo code for selection operator

b. Crossover operator:

This returns already selected elite children and child generated by crossover. Crossover for particular location is performed if the randomly generated number is less than or equal to the predefined crossover probability. Here, ‘pc’ is crossover probability,’ numelite’ is number of elite children, ‘numpop’ is number of current population and ‘numparent’ is number of parent prepared for crossover.

```
function  [elite, child]=crossover(population, pc, numelite)
Call selection function;
For i=1:numparent
       If rand<=pc
          loop: mask1=rand(1,numpara);
           mask2=ones(1,numpara)-mask1;
           child1(i, :)=mask1*parent1(i,:  )+mask2*parent2(i, :);
           child1(i, :)=mask2*parent1(i,:  )+mask1*parent2(i, :);
            If any out of bound
                goto loop;
             end If
      Else
           child1(i,:  )=parent1(i,:  );
           child2(i, : )=parent2(i,: );
     end If
end For
child=[child1;  child2];
```

Figure 3-8 Pseudo code for crossover operator

c. Mutation operator

Non uniform mutation, which is dependent on each generation, is used. Here, 'gen' is to mean each generation number, 'pm' is mutation probability and 'maxgeneration', is maximum generation. If randomly generated number is lower than the predefined mutation probability, for particular point mutation is performed.

```
function  [mutate]=mutation(population,pm,gen,maxgeneration)
For i=1:numpop
  For j=2:numpara-1
    If rand<=pm
      multiply=rand * (1 - gen/maxgeneration^b);
      If rand<05
        population(i,j)=population(i,j)+(19.444-  population(i,j))*multiply;
     Else
        population(i,j)=population(i,j)  -  population(i,j)*multiply;
    end If
  end For
end For
```

Figure 3-9 pseudo code for mutation operator

d. Replacement strategy

Replacement strategy used for this thesis is the simple replacement method which is replacing old population other than the elated ones by the newly generated children. So, children generated by crossover followed by mutation will replace old populations which are not the member of children evolved to the next generation without modification.

- **Fitness evaluation**

It is evaluated based on the system explained at the means of constraint handling technique. We decided to return the feasible fitness value and feasible population from the current population. This information has a great effect on selection of parent to fill the mating pool.

```
function [feasible_fitness, fitness,penalty,feasible_population]=new_objective(population)
Call constraint evaluation function;
Call original objective function, fitness(x);
Find feasible population, update feasible-fitness;
Find maximum of feasible fitness, Fmax_feasible;
Sum violation for each population V(x), penality;
If (feasible)
   fitness(x)=fitness(x);
Else
   fitness(x)=Fmax_feasible+V(x);
end If
```

Figure 3-10 Pseudo code for fitness evaluation

4. MATLAB simulation and discussion of simulation result

4.1 MATLAB simulation result

Each of the optimization problems developed is equipped with MATLAB '.m' functions. The fastest time with maximum energy consumption driving scheme is obtained by accelerating the train at the maximum acceleration from startup to the maximum allowed speed and if the distance can be covered by braking, then braking follows; otherwise speed keep (cruising) follows till braking point **[71]** and **[72].** The fastest driving scheme is helpful to bound energy consumption of the train for time related optimization problem.

All the models developed are applied for 21 sections of the track in north-south line of Addis Ababa LRT and simulation result obtained is analyzed.

Gradient variation with in small range is approximated by average of the consecutive gradients. The simulation is conducted for 100 population generated randomly with in the boundary limit of speed 25km/hr for track that uses the lateral switch, 50km/hr for level crossing and 70km/hr for mainline.

The MATLAB code has 11 functions. Two of them are searching functions. One of the searching functions performs searching of zero speed in the middle of the track (which is not allowed) and the other searching function is used for controlling the boundary condition (which is protection of negative and greater than upper boundary speed). The other two functions are responsible for handling objective function. The first objective function handles the original fitness and the other handles the new fitness developed by the principle of Debs' constraint handling technique. Three of the MATLAB functions are genetic algorithm operator functions which are selection, crossover and mutation. Constraint handling function is used for handling the constraints and tractive effort function is the function used for determining tractive effort, braking effort and stage at each discrete point with in the given section. Maximum driving function allows determining the fastest driving scheme. The final function is the function which calls all the other functions for the number of generation. Stopping criteria used for this thesis is maximum number of generation.

In order to obtain a better solution we apply the principle of using the final population obtained from the respective model for next round simulation. This way one can have better solution than starting from randomly generated population every time.

1. **Between station 27-26**

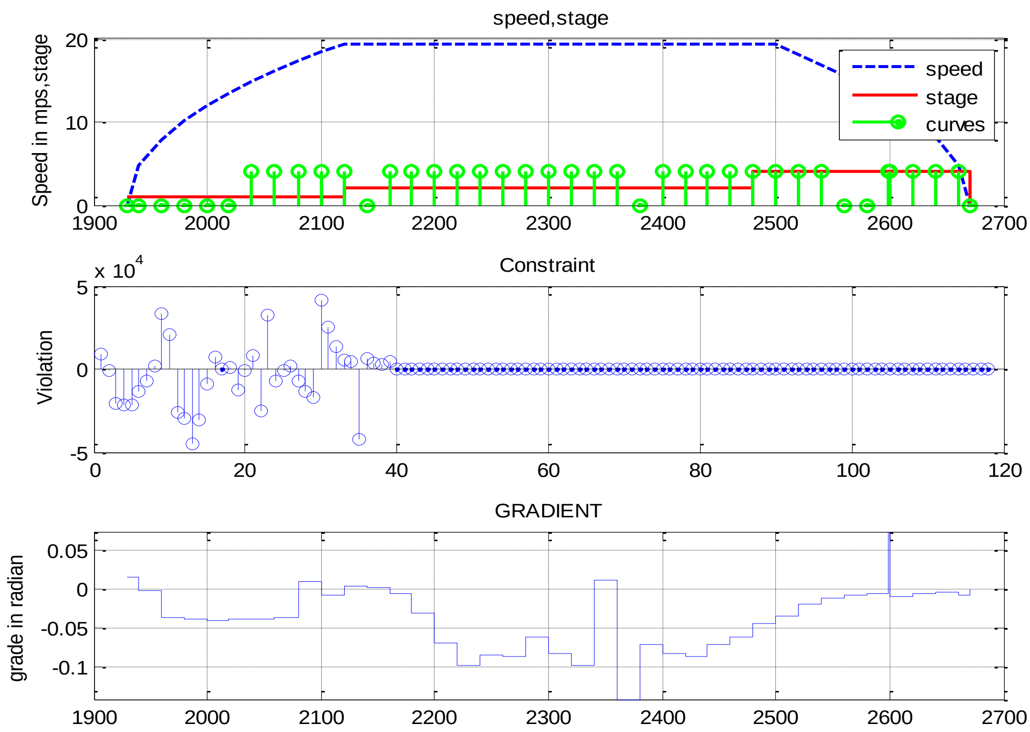


Figure 4-1 Fast drive, gradient and constraint violation for st26-27

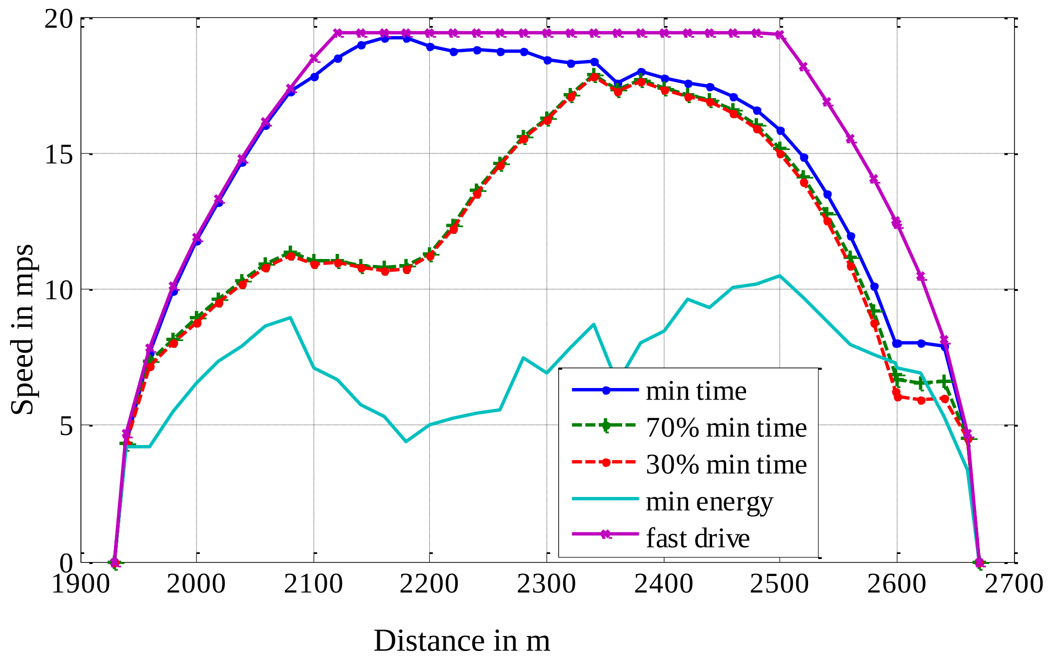


Figure 4-2 Five speed profile for st26-27

2. Between station 26-25

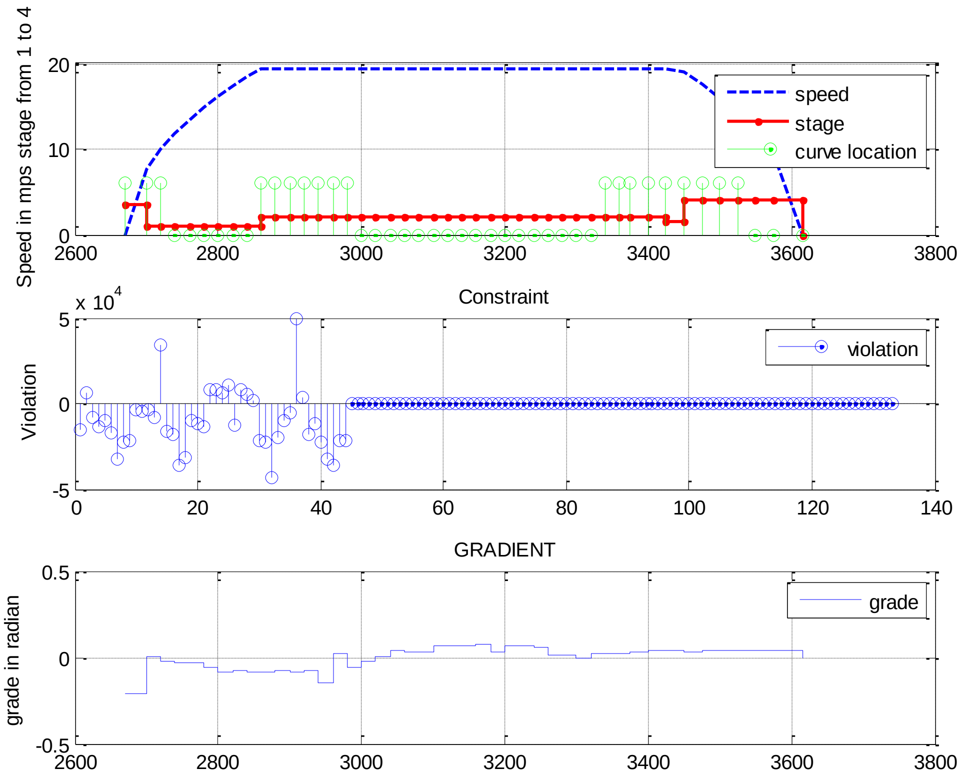


Figure 4-3 Fast drive, gradient and constraint violation for st25-26

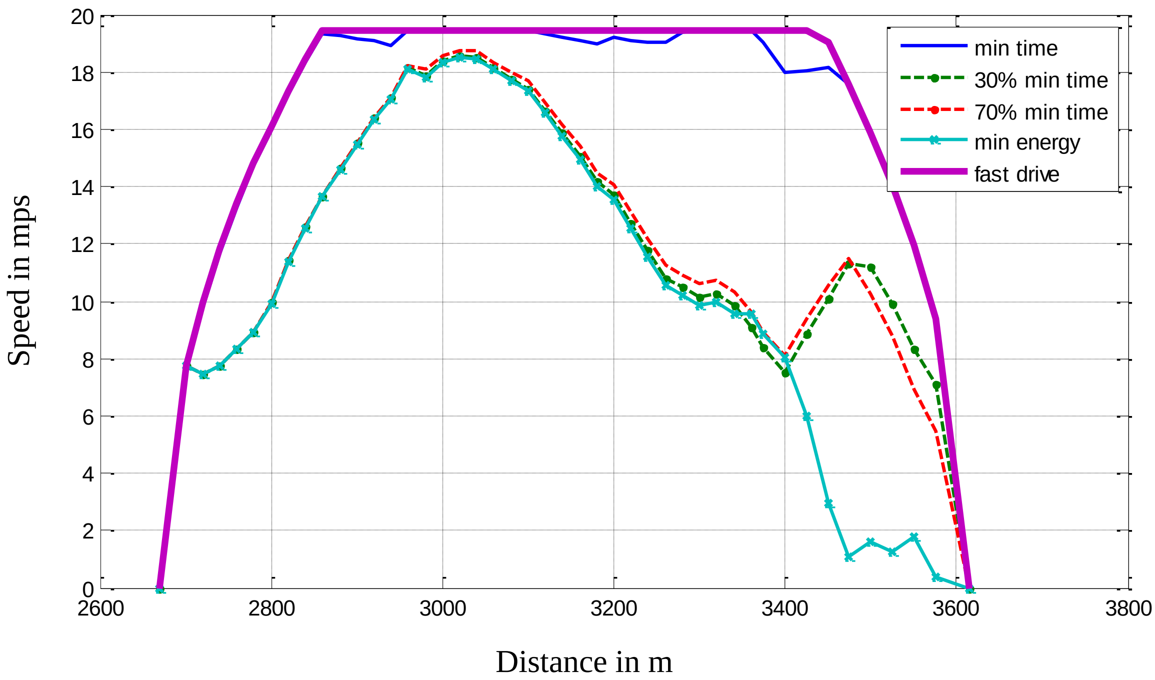


Figure 4-4 Five speed profile for st25-26

3. Between station 25-24

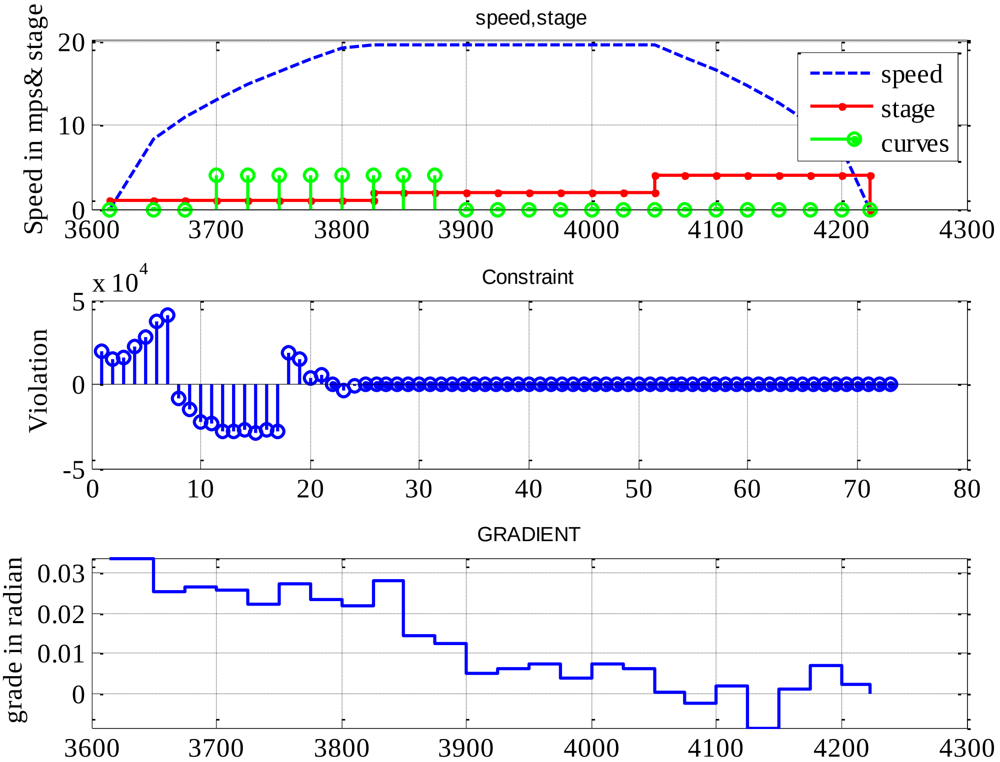


Figure 4-5Fast drive, gradient and constraint violation for st24-25

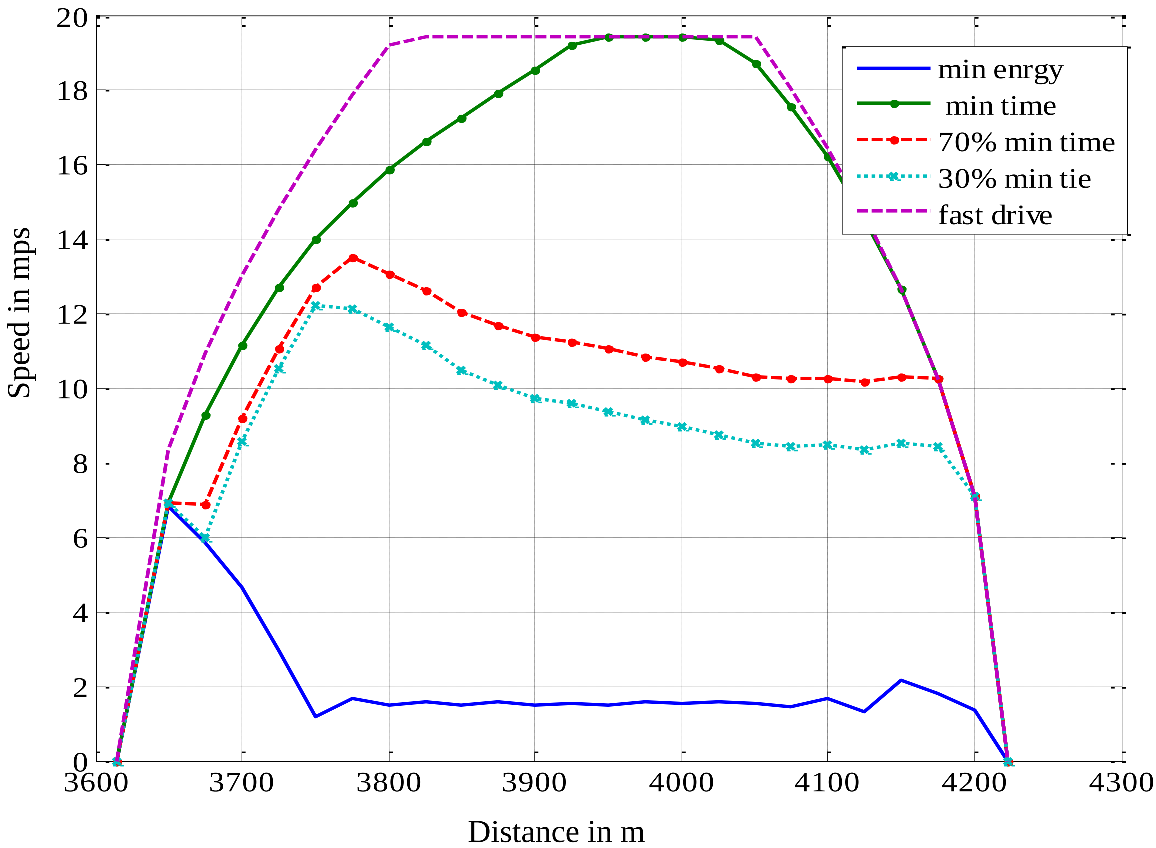


Figure 4-6 Five speed profile for st24-25

4. Between station 24-23

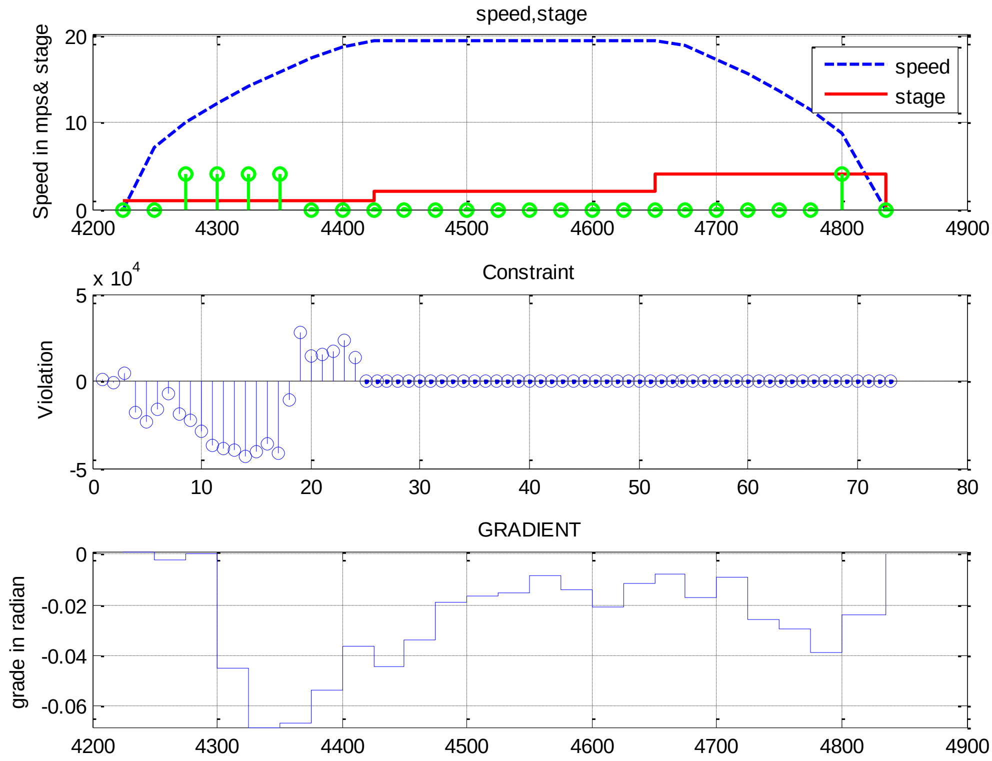


Figure 4-7 Fast drive, gradient and constraint violation for st23-24

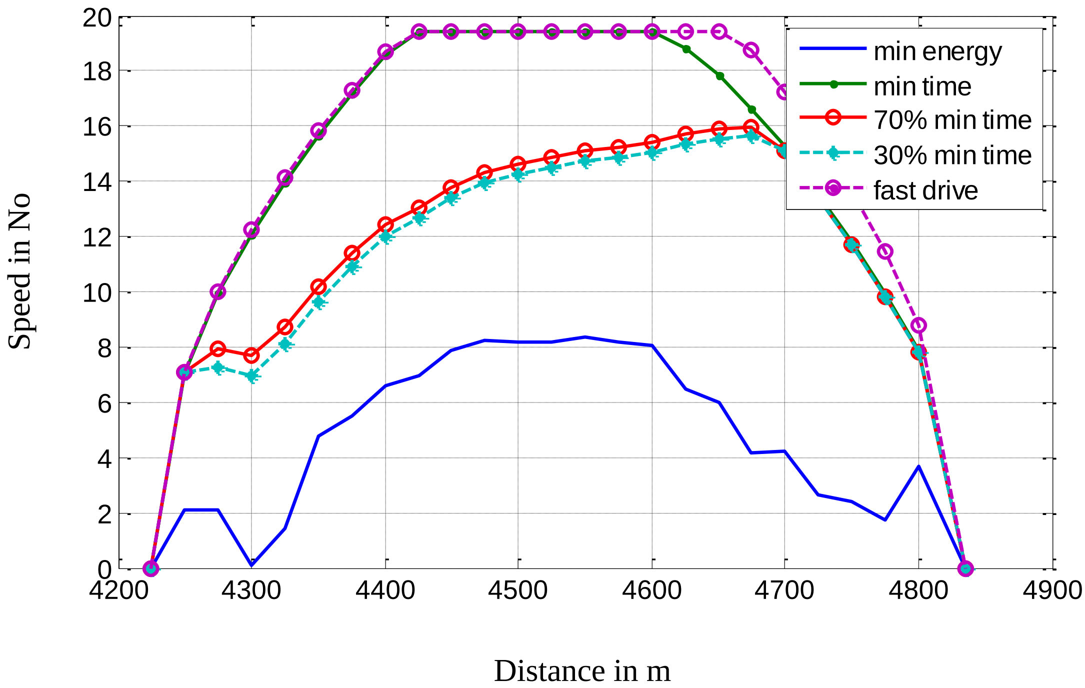


Figure 4-8 Five speed profile for st24-23

5. Between station 23-22

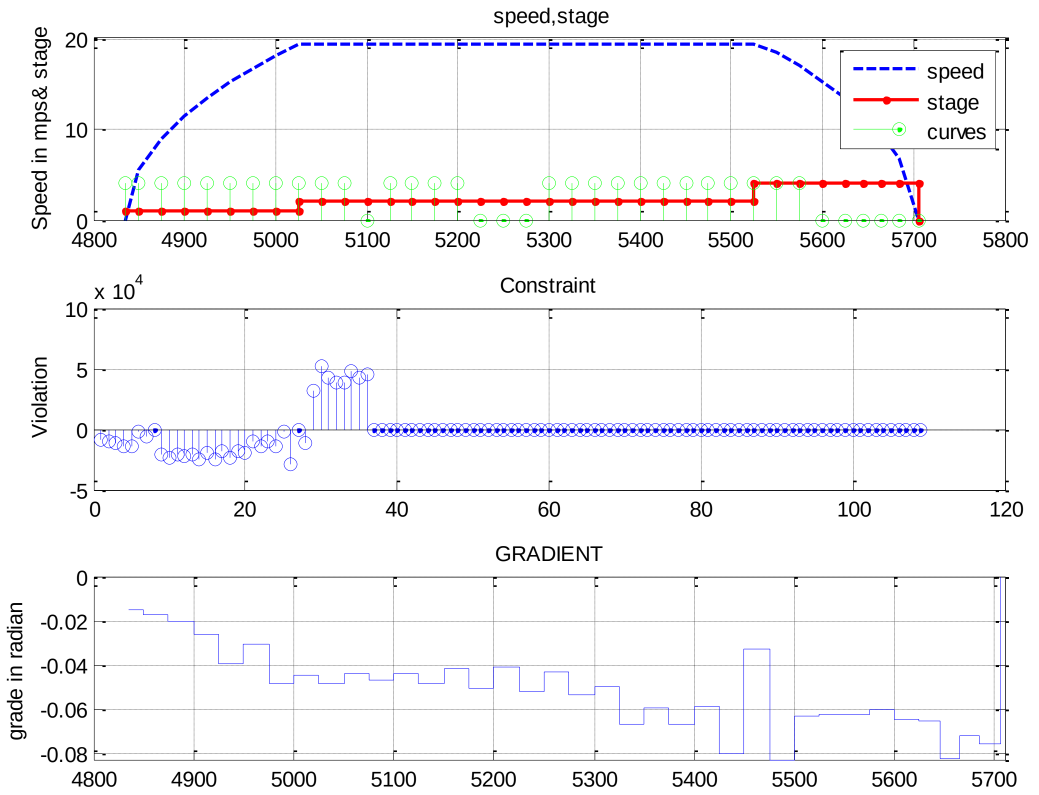


Figure 4-9 Fast drive, gradient and constraint violation for st22-23

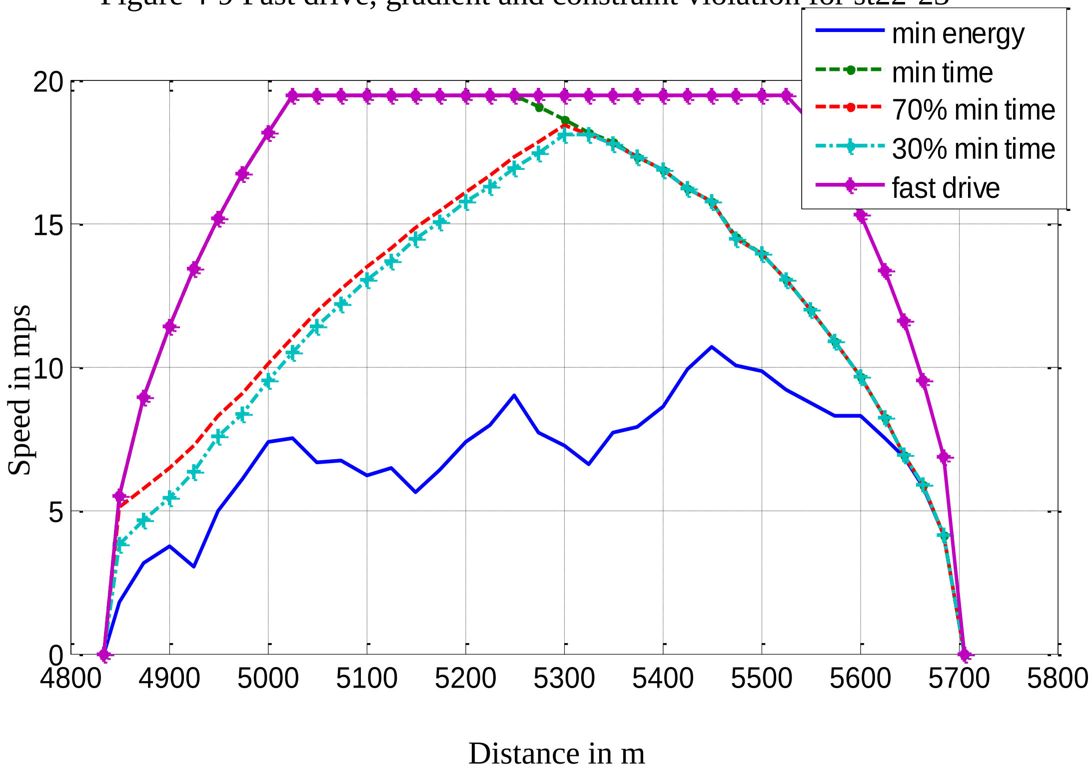


Figure 4-10 Five speed profile for st22-23

6. Between station 21 and 22

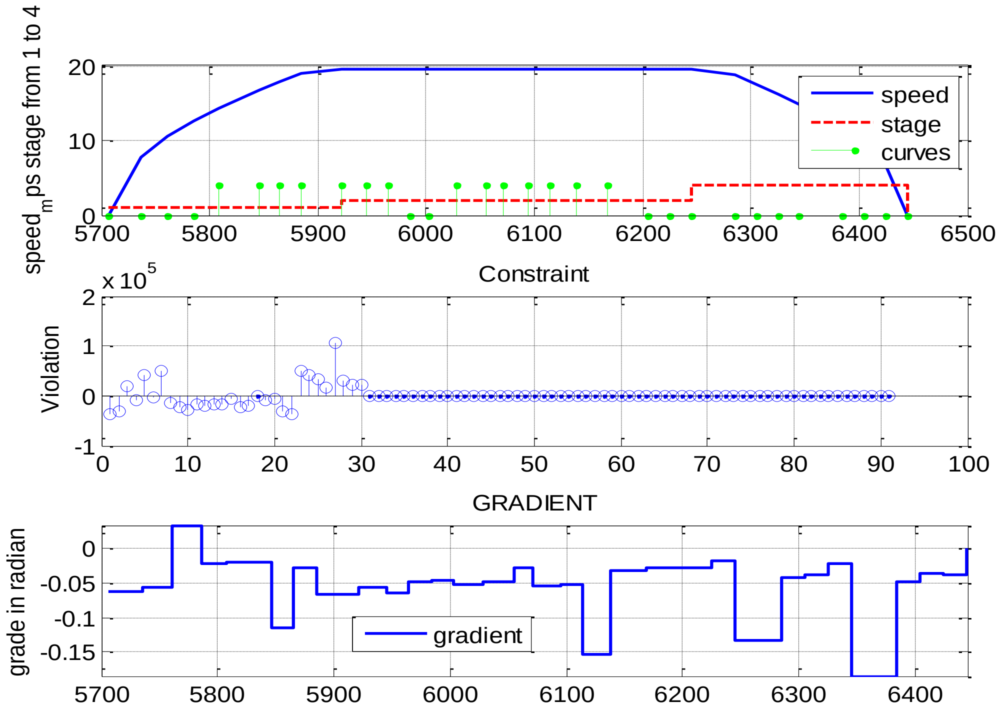


Figure 4-11 Fast drive, gradient and constraint violation for st21-22

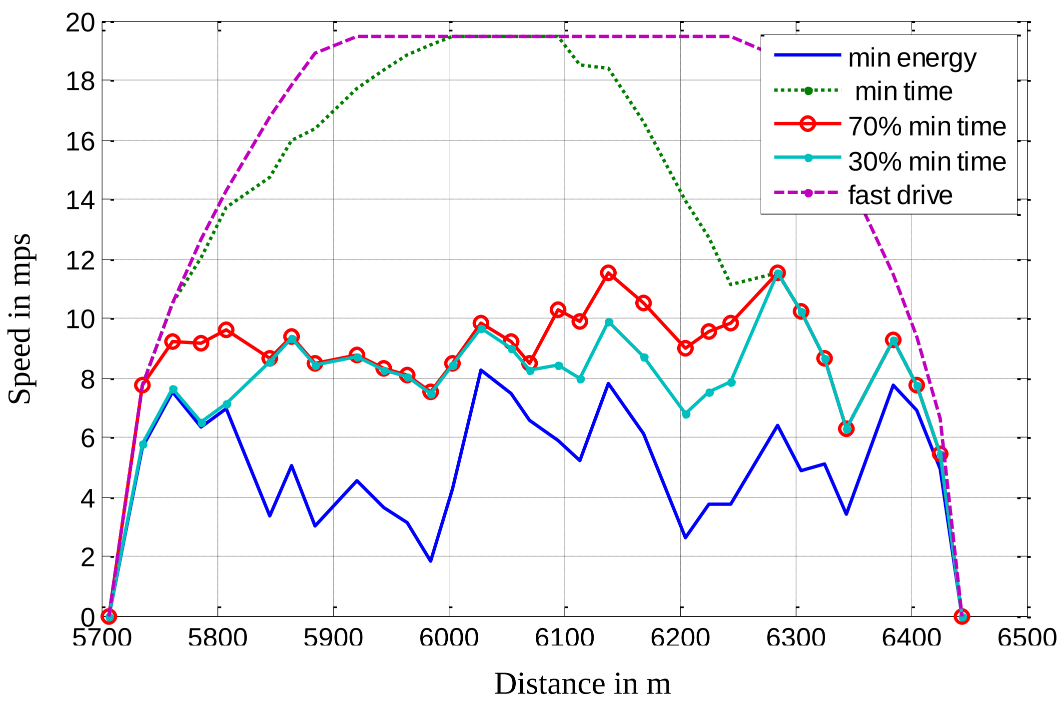


Figure 4-12 Five speed profile for st22-21

7. Between station 20 and 21

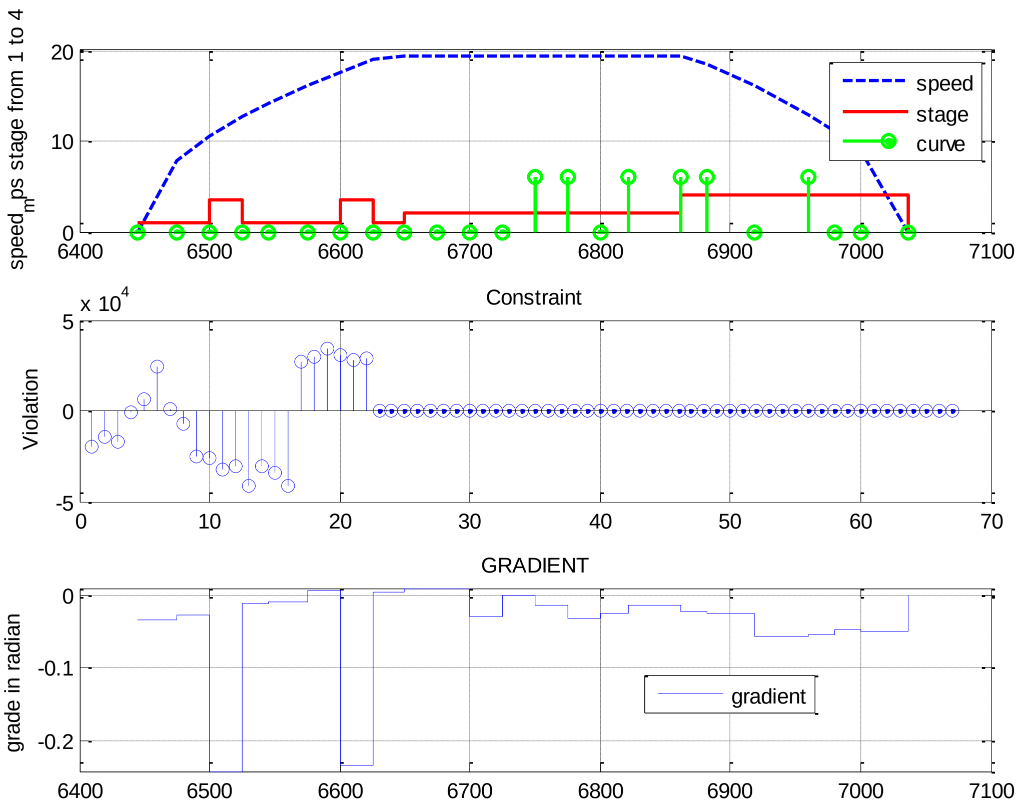


Figure 4-13 Fast drive, gradient and constraint violation for st20-21

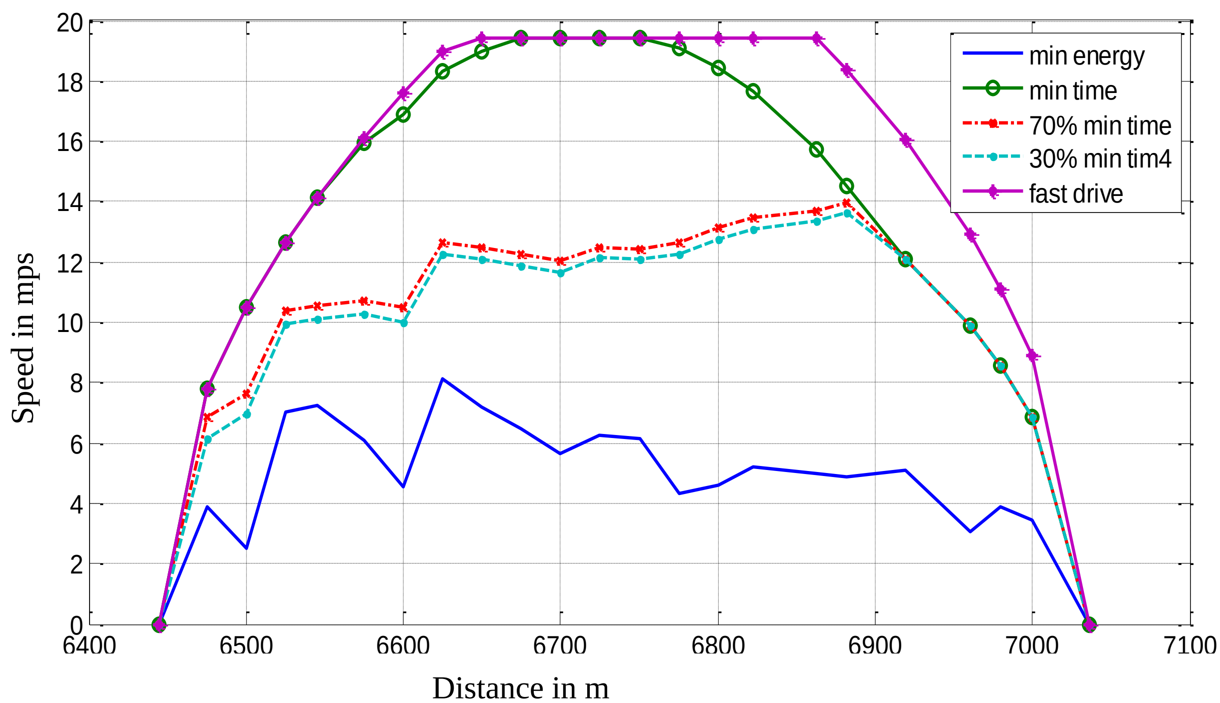


Figure 4-14 Five speed profile for st21-20

8. Between station 19 and 20

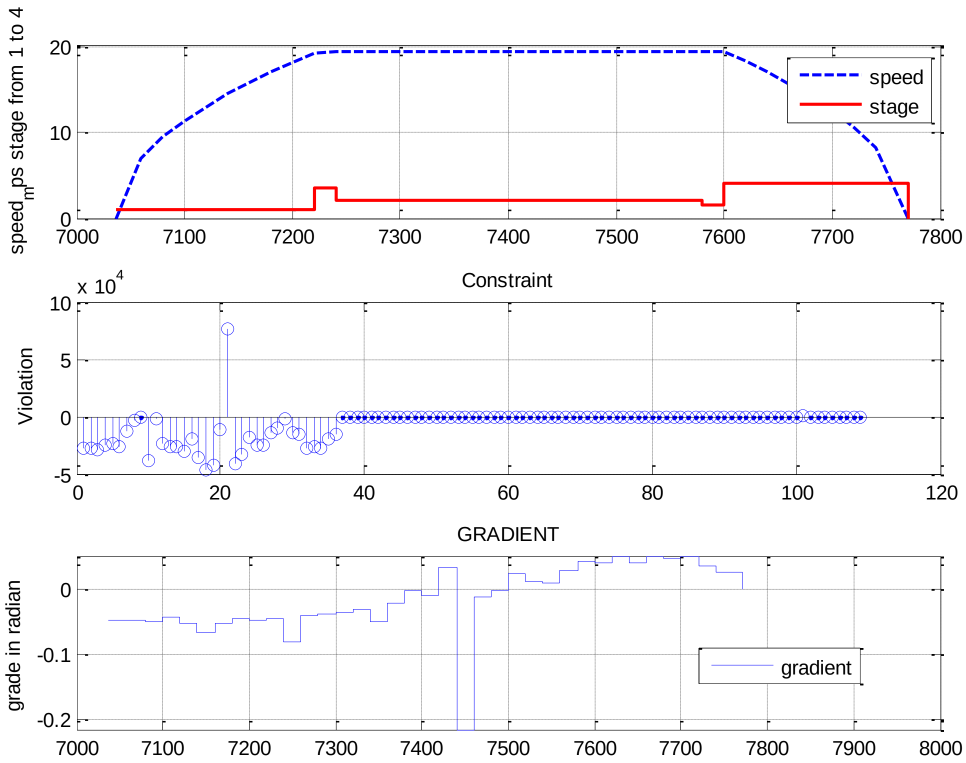


Figure 4-15 Fast drive, gradient and constraint violation for st20-19

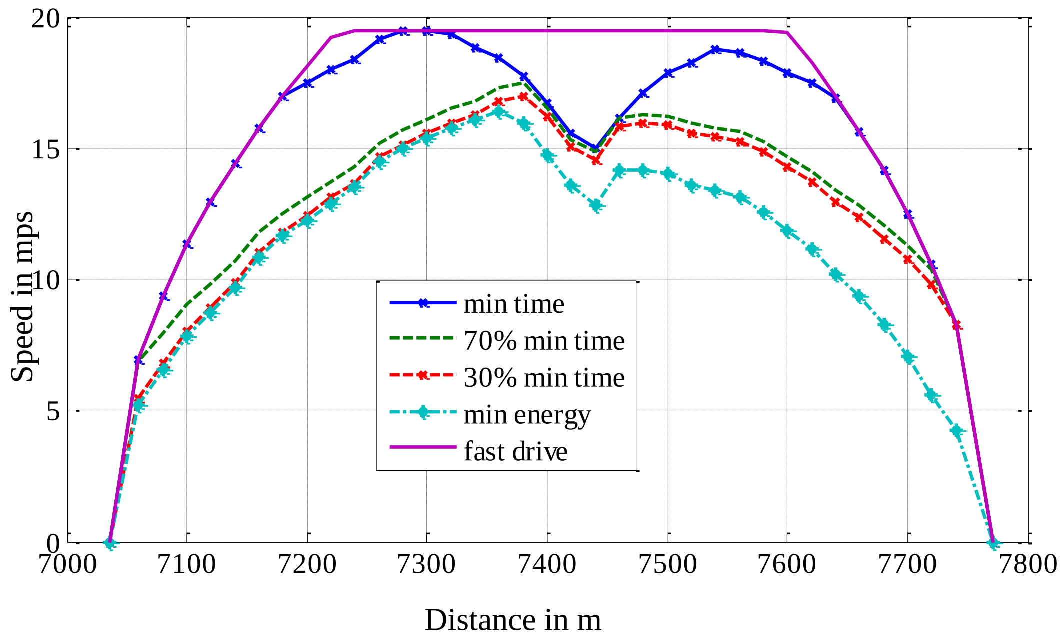


Figure 4-16 Five speed profile for st20-19

9. Between station 18 and 19

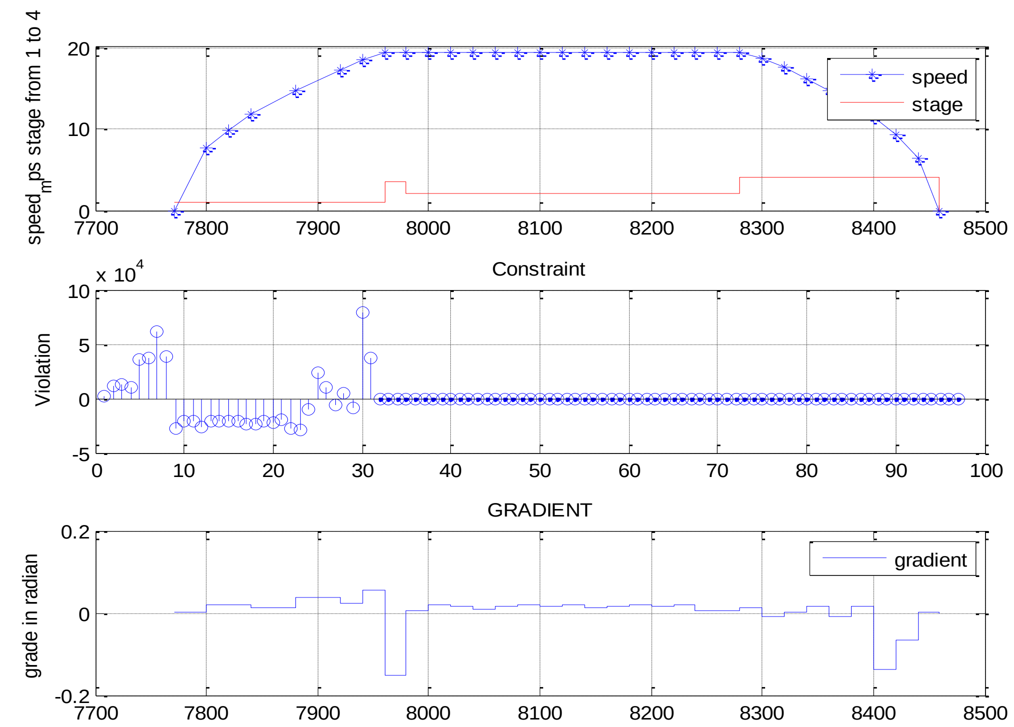


Figure 4-17 Fast drive, gradient and constraint violation for st18-19

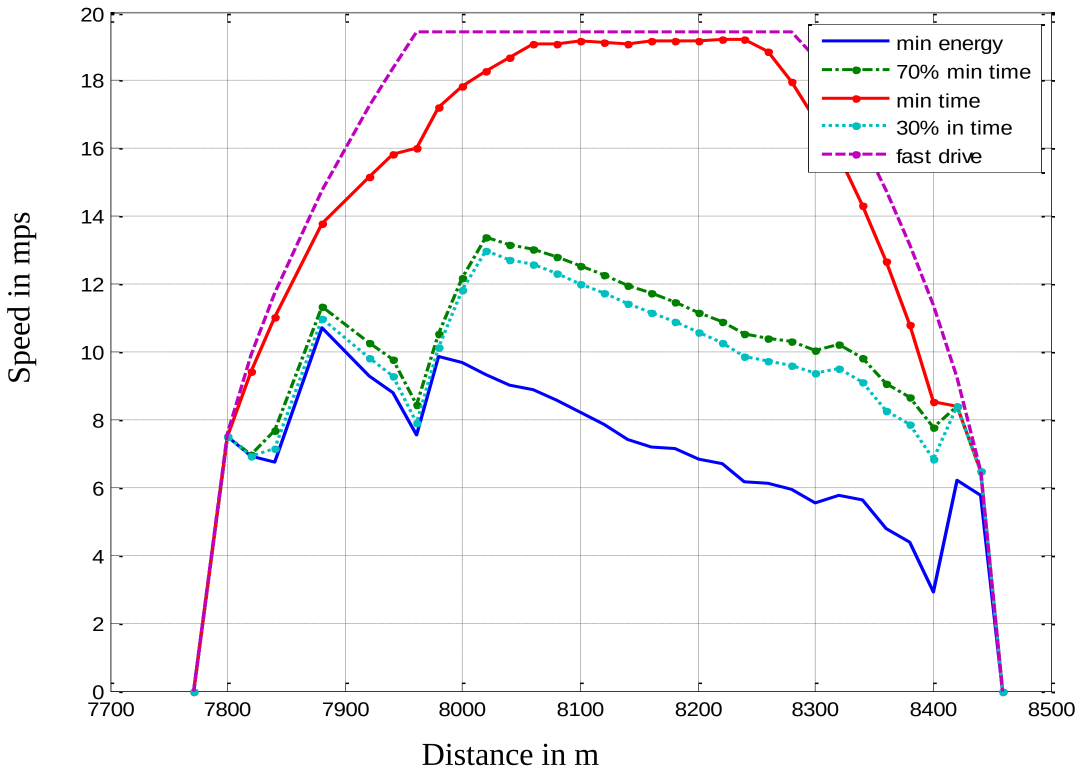


Figure 4-18 Five speed profile for st19-18

10. Between station 17 and 18

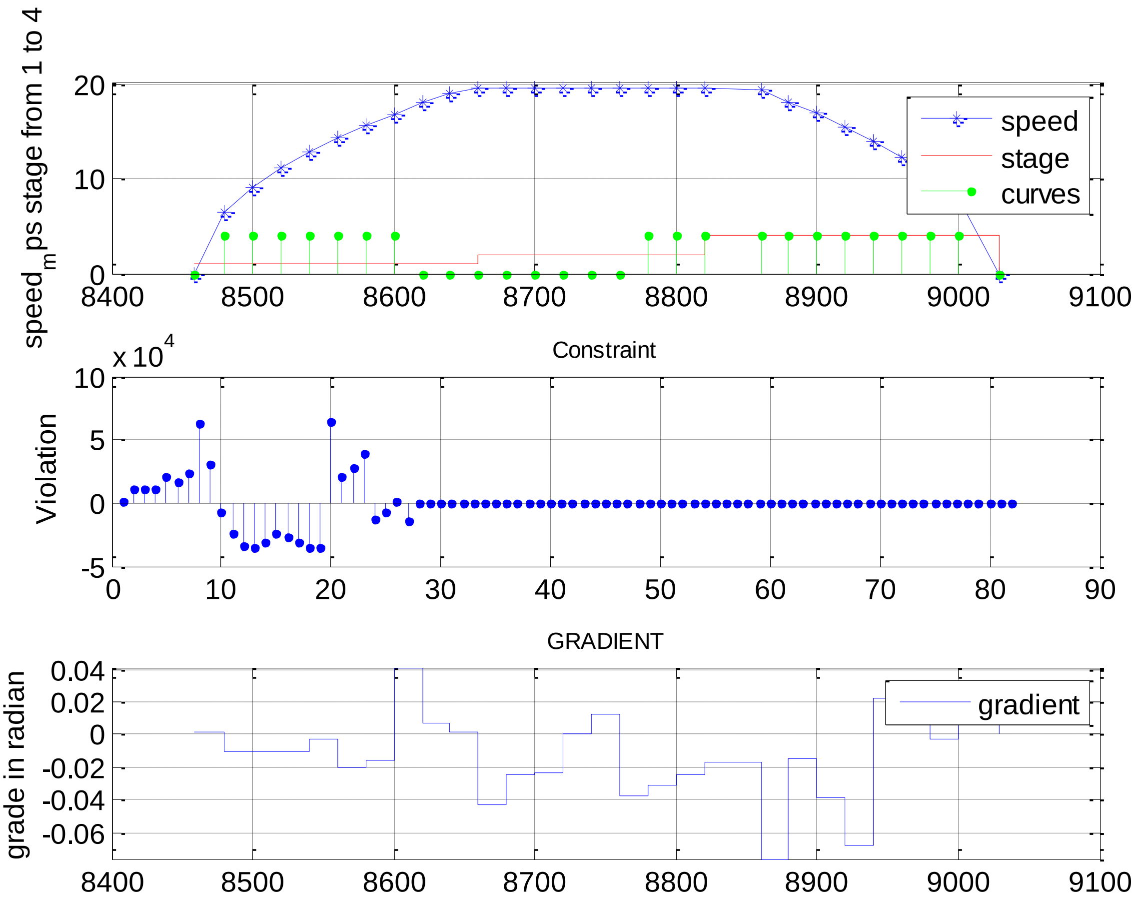


Figure 4-19 Fast drive, gradient and constraint violation for st17-18

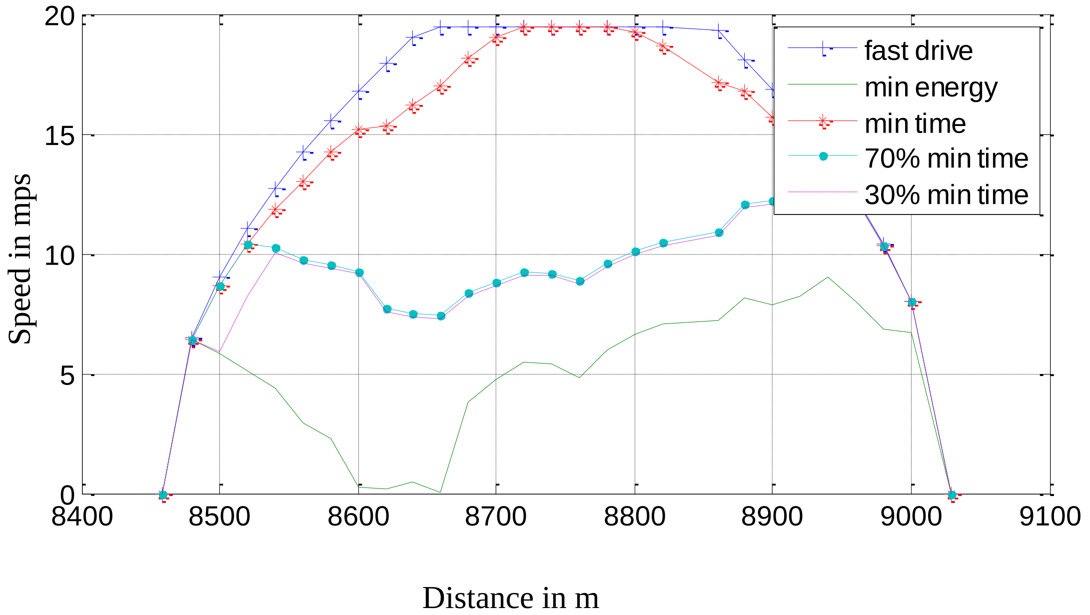


Figure 4-20 Five speed profile for st17-18

11. Between station 16 and 17

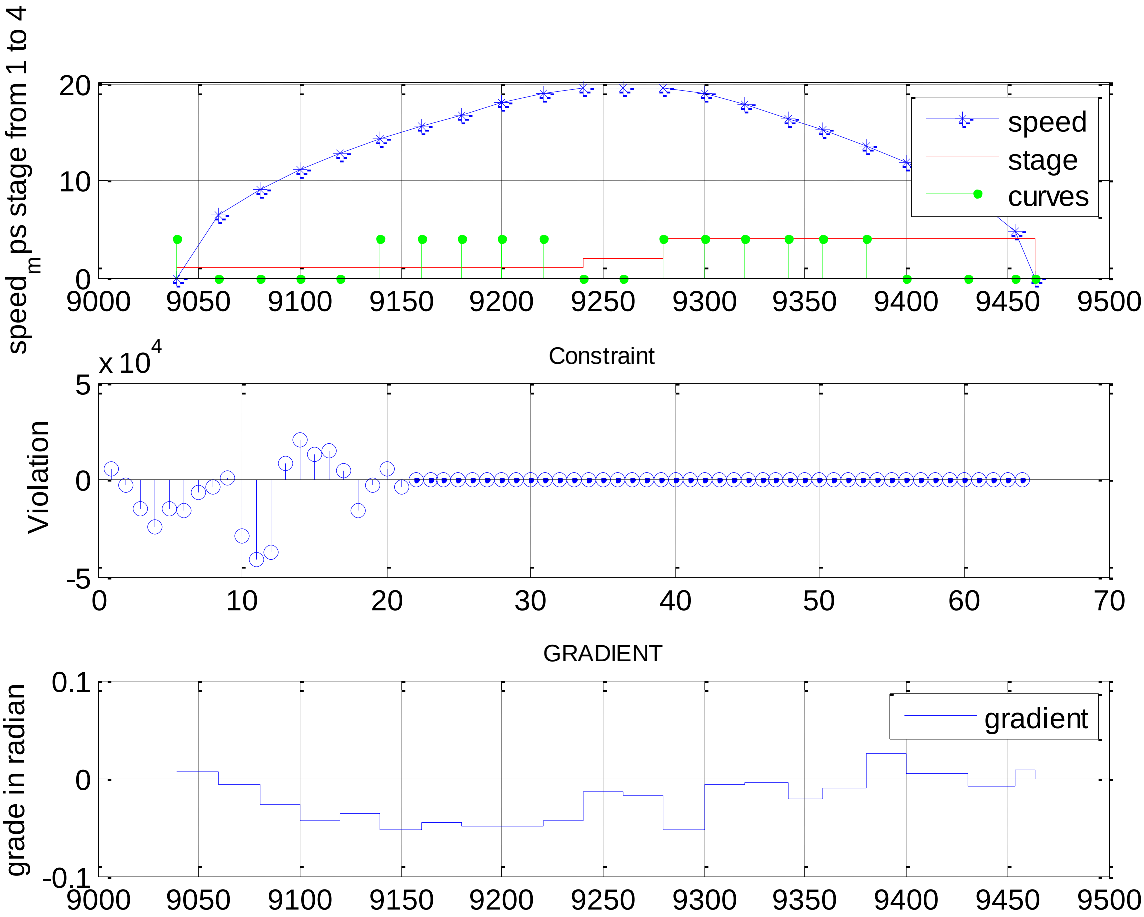


Figure 4-21 Fast drive, gradient and constraint violation for st16-17

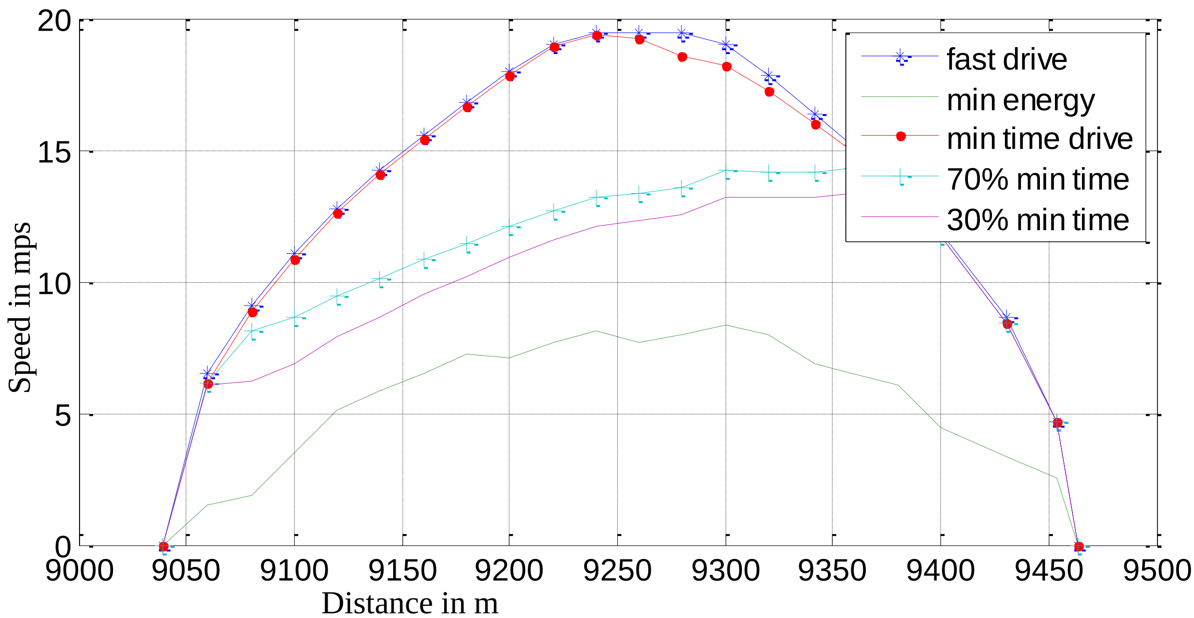


Figure 4-22 Five speed profile for st16-17

12. Between station 15 and 16

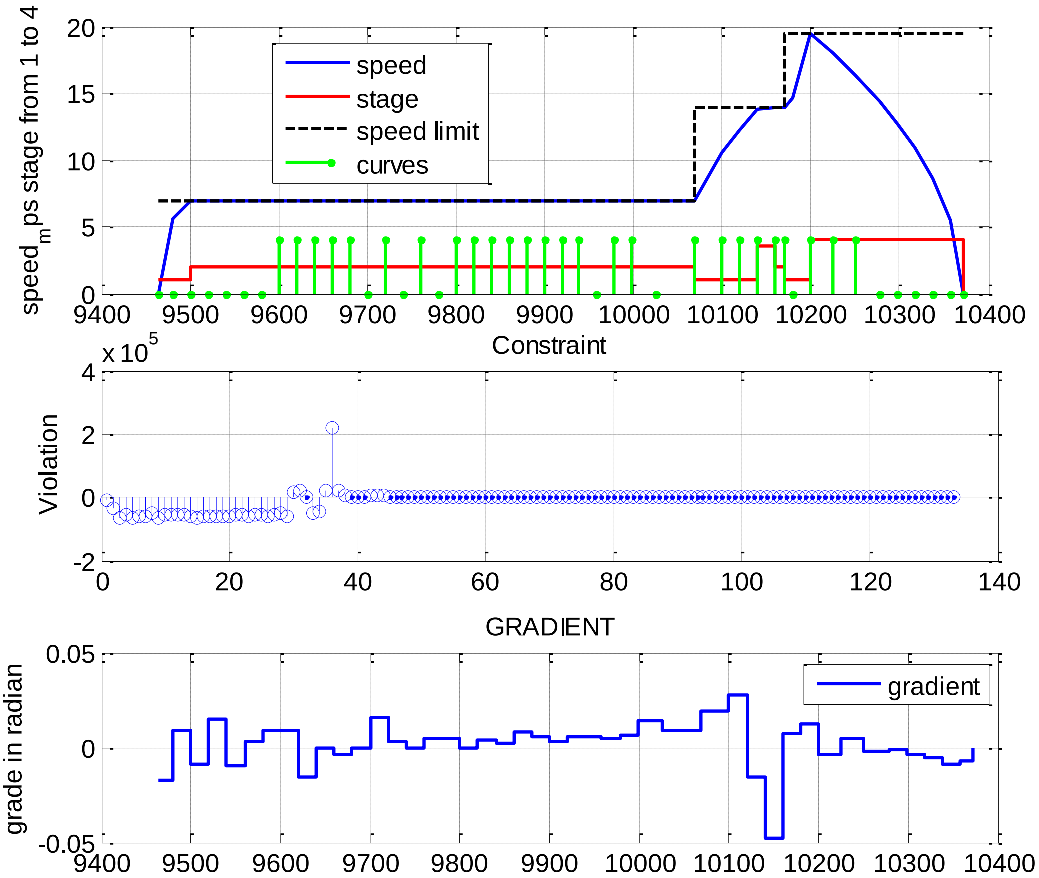


Figure 4-23 Fast drive, gradient and constraint violation for st15-16

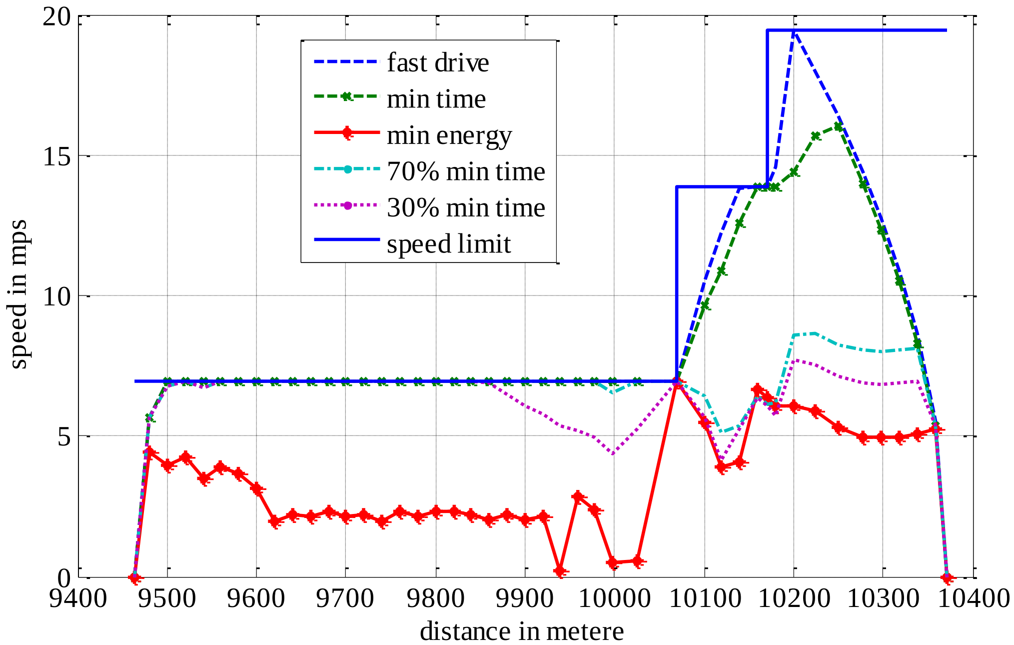


Figure 4-24 Five speed profile for st16-15

13. Between station 14 and 15

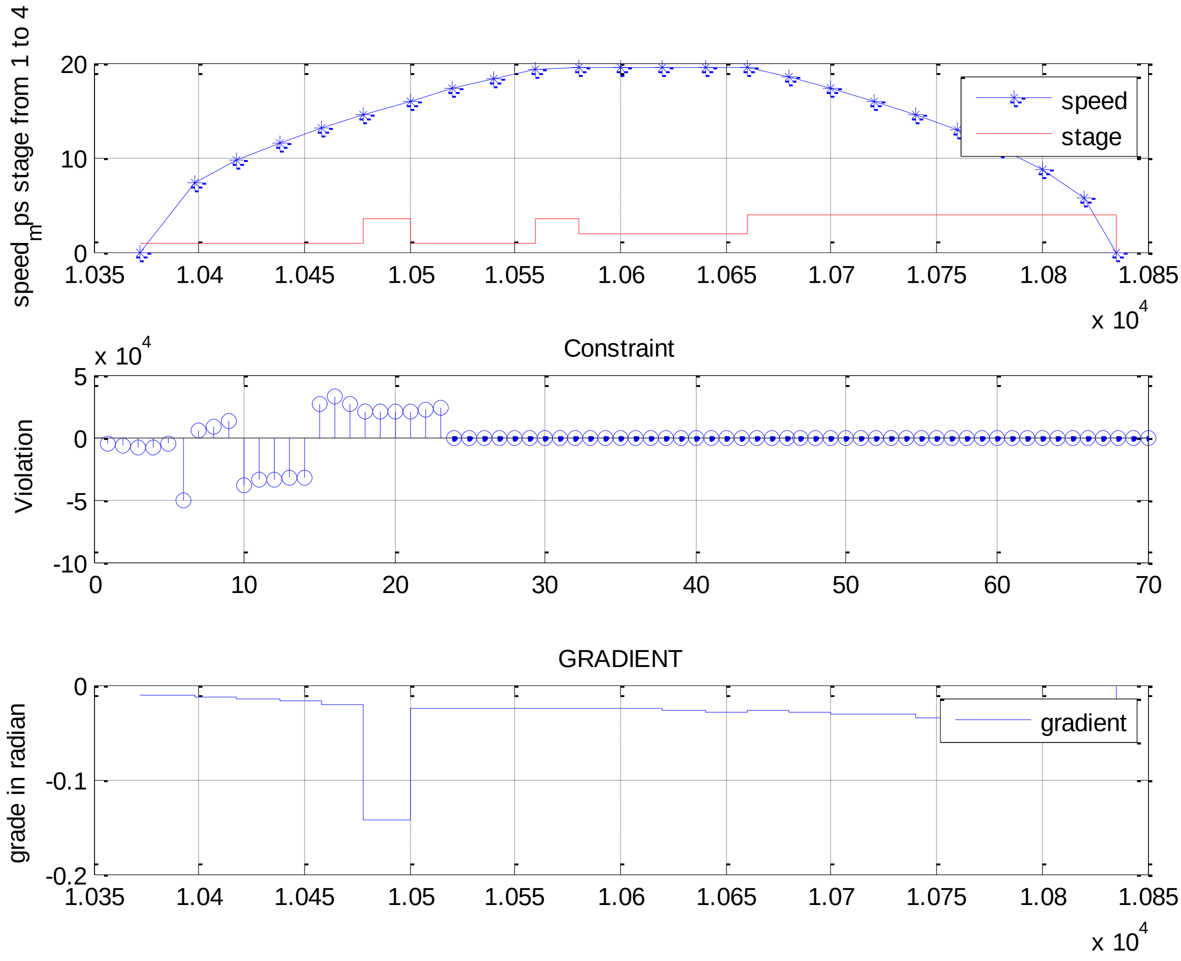


Figure 4-25 Fast drive, gradient and constraint violation for st14-15

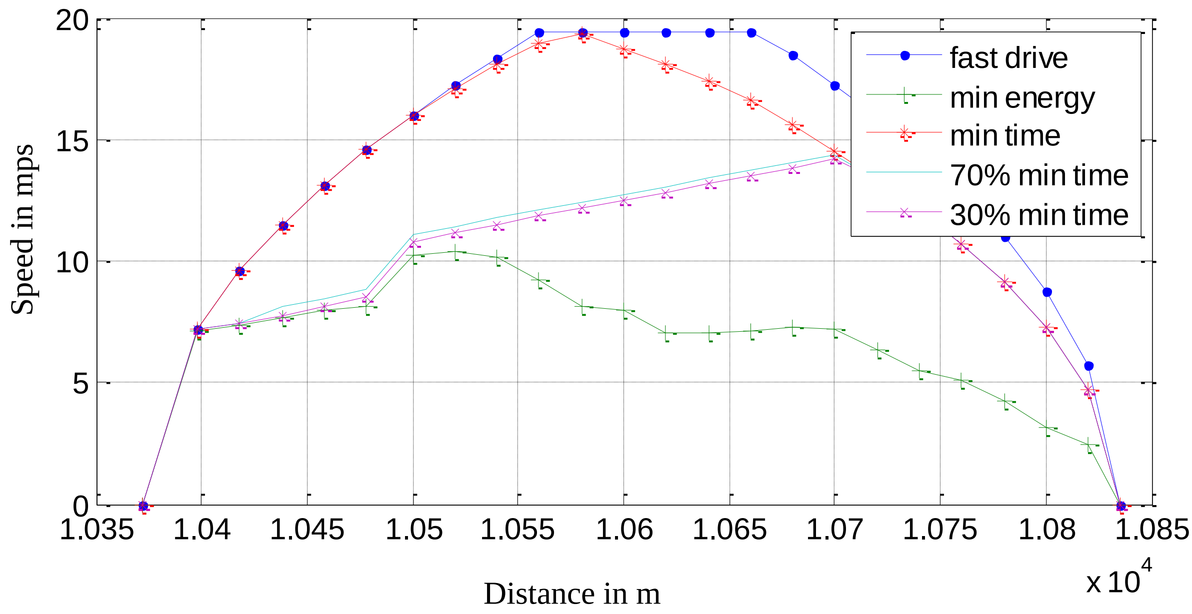


Figure 4-26 Five speed profile for st15-14

14. Between station 13 and 14

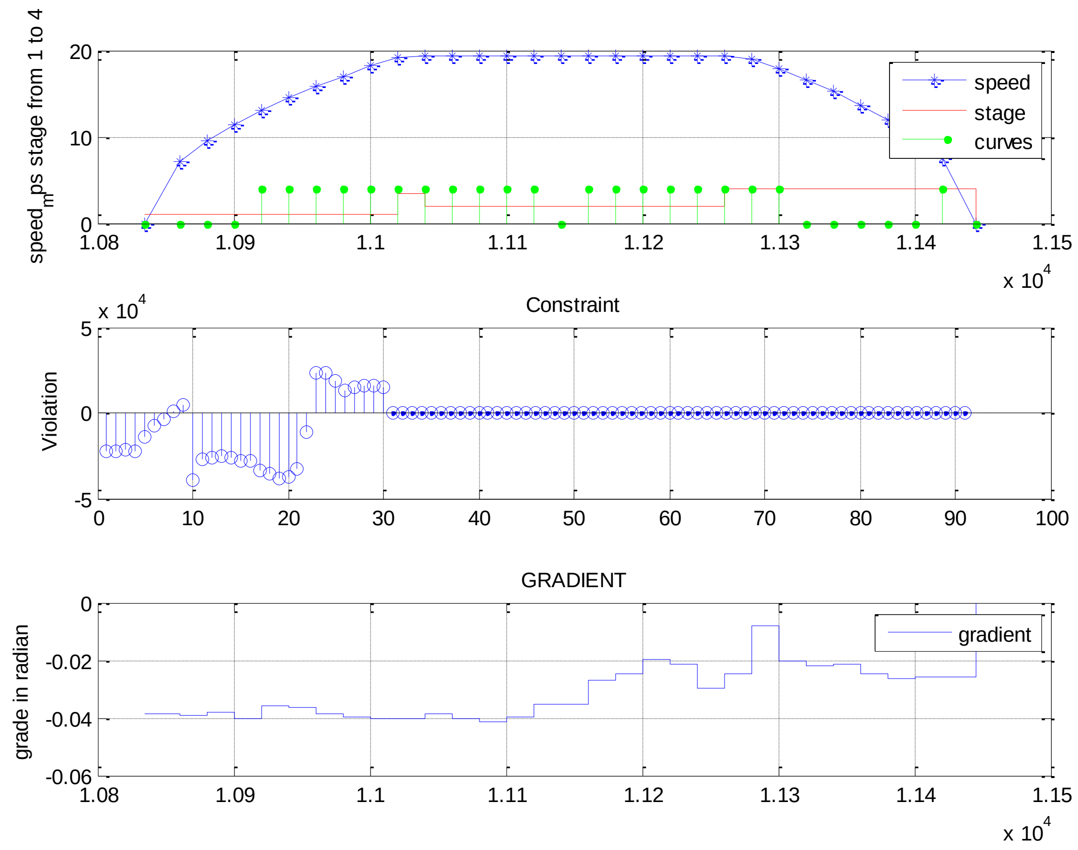


Figure 4-27 Fast drive, gradient and constraint violation for st13-14

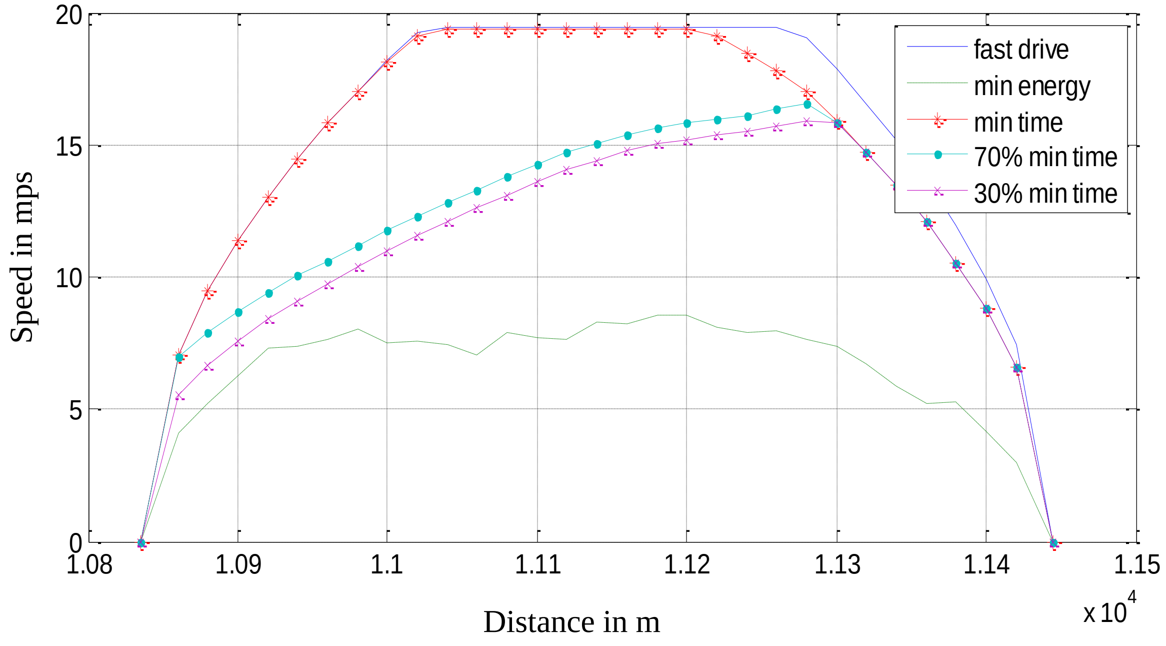


Figure 4-28 Five speed profile for st14-13

15. Between station 12 and 13

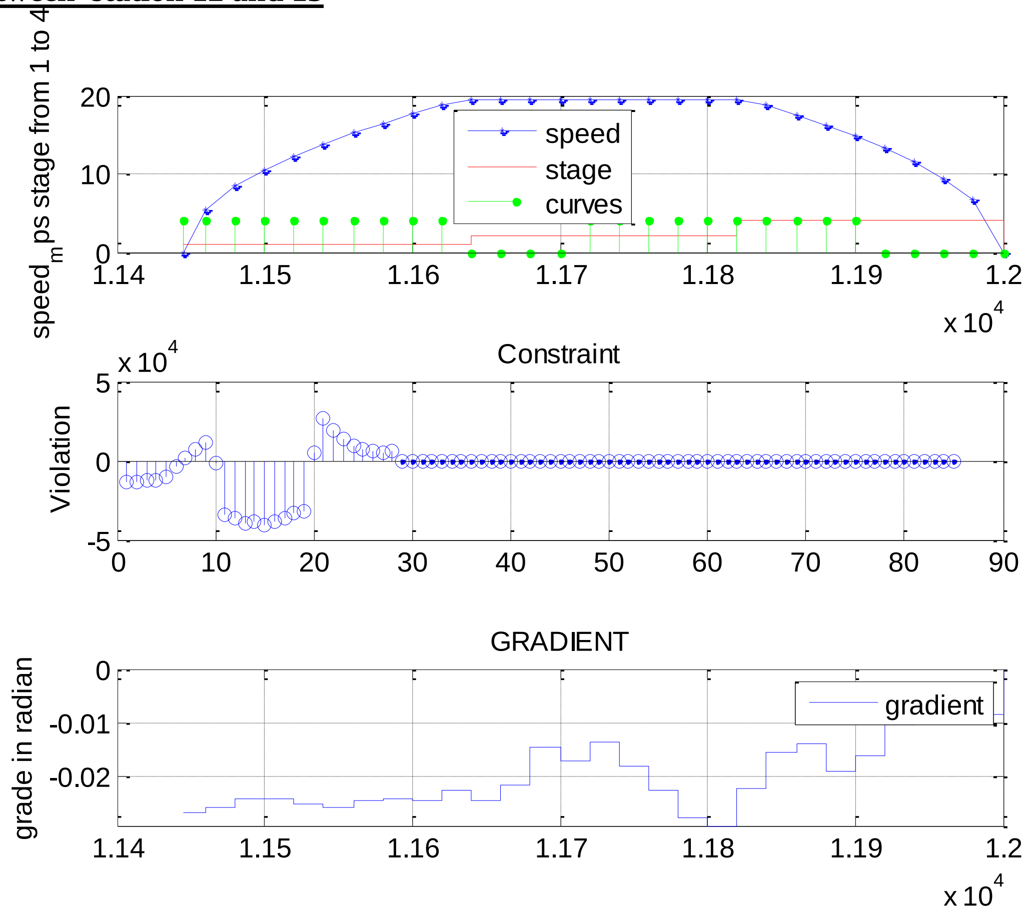


Figure 4-29 Fast drive, gradient and constraint violation for st12-13

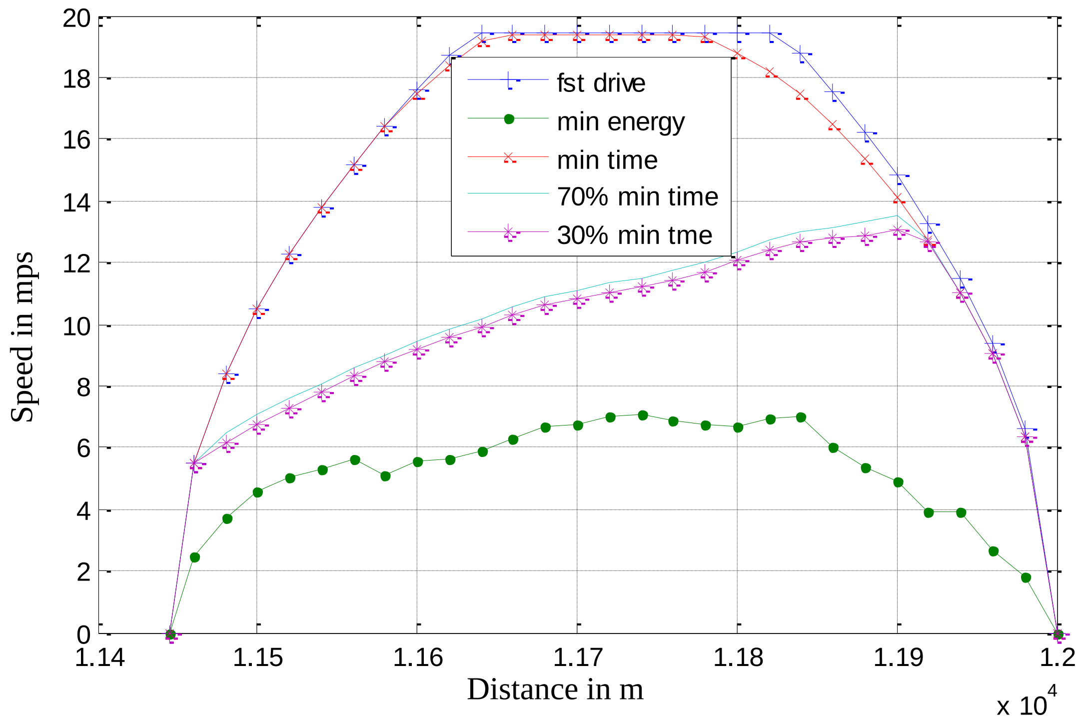


Figure 4-30 Five speed profile for st12-13

16. Between station 11 and 12

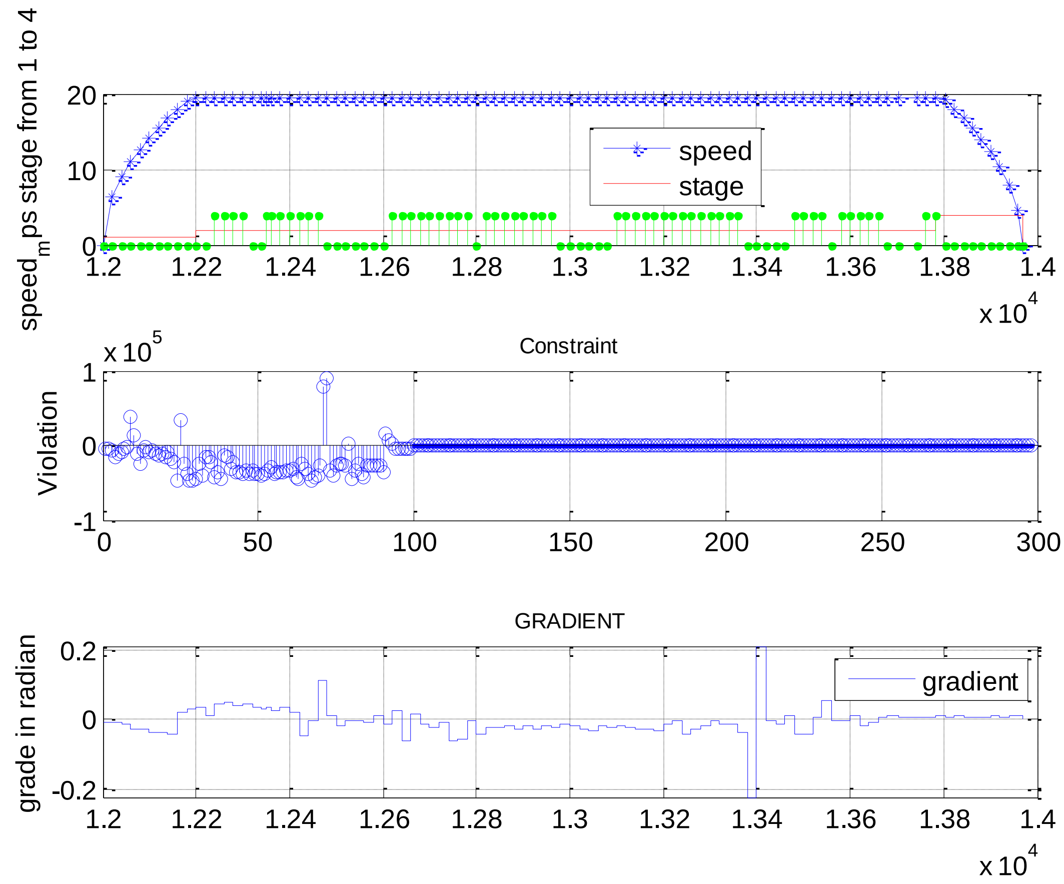
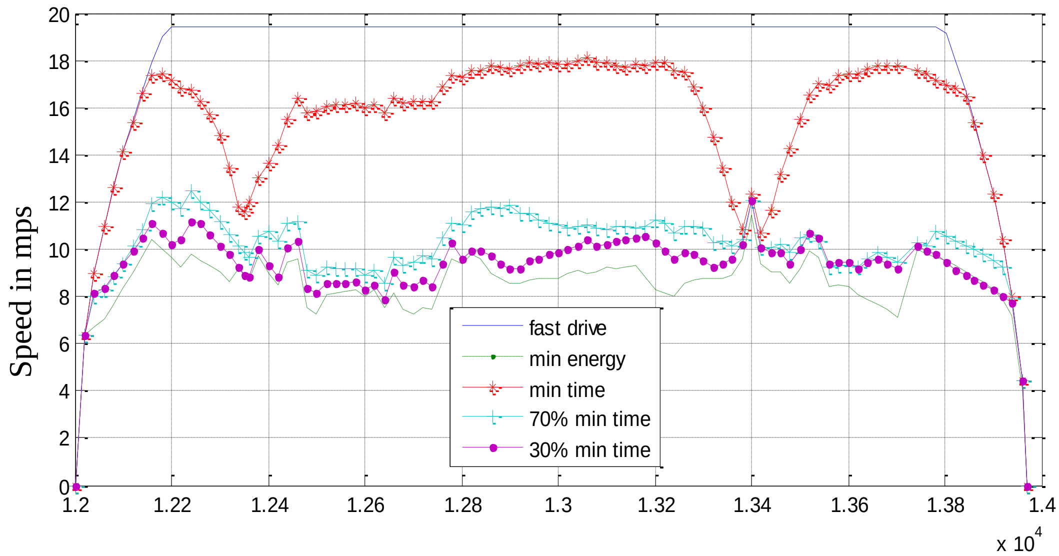


Figure 4-31 Fast drive, gradient and constraint violation for st11-12



Distance in m

Figure 4-32 Five speed profile for st11-12

17. Between station 10 and 11

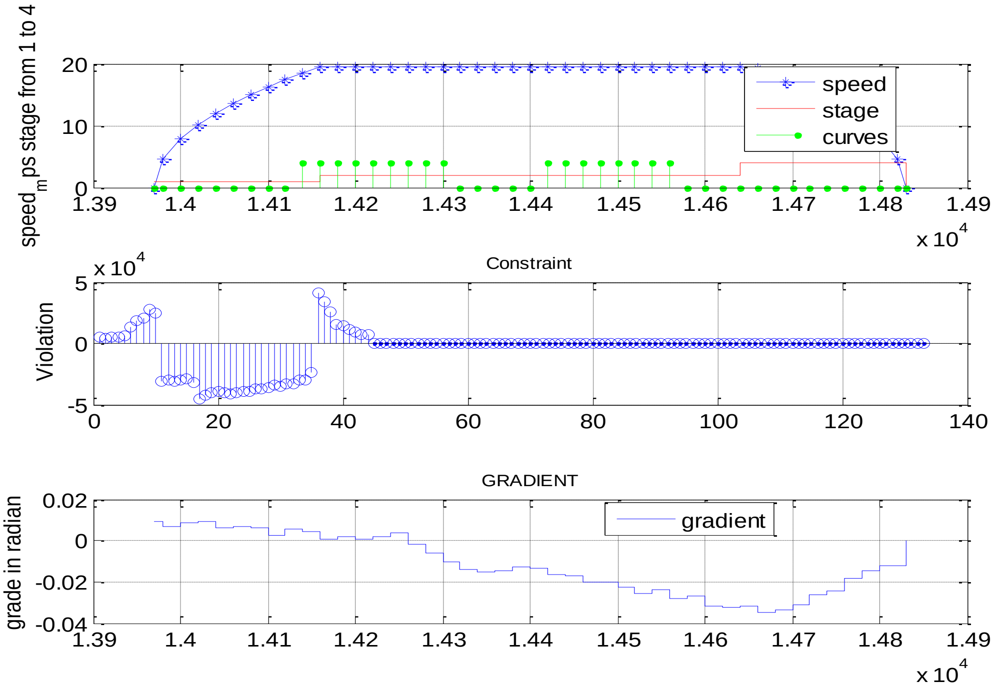


Figure 4-33 Fast drive, gradient and constraint violation for st10.11

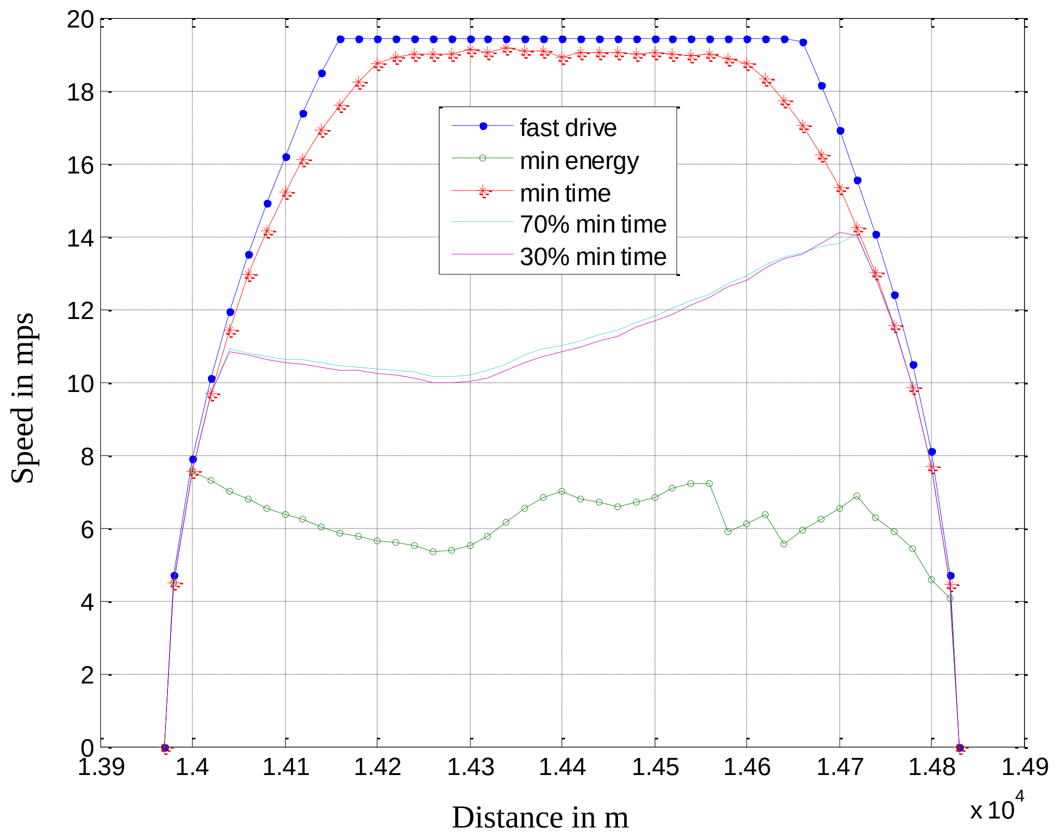


Figure 4-34 Five speed profile for st10-11

18. Between station 9 and 10

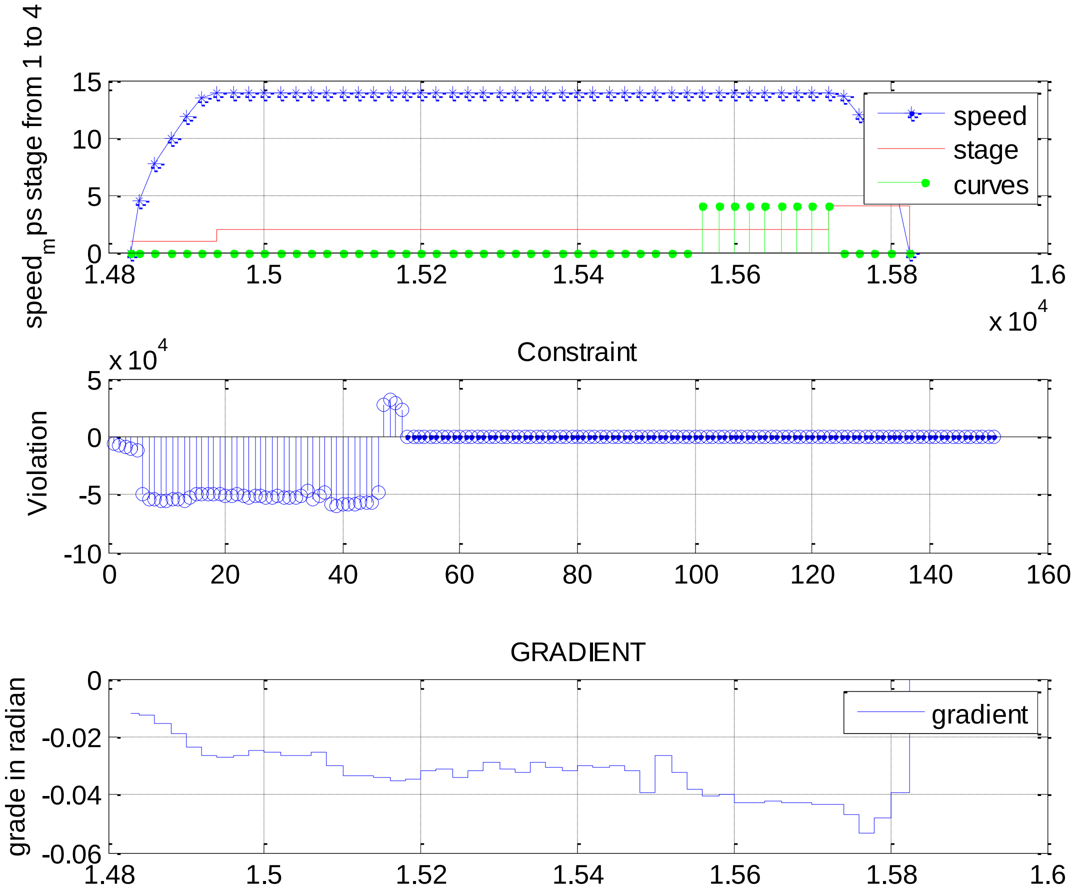


Figure 4-35 Fast drive, gradient and constraint violation for st9-10

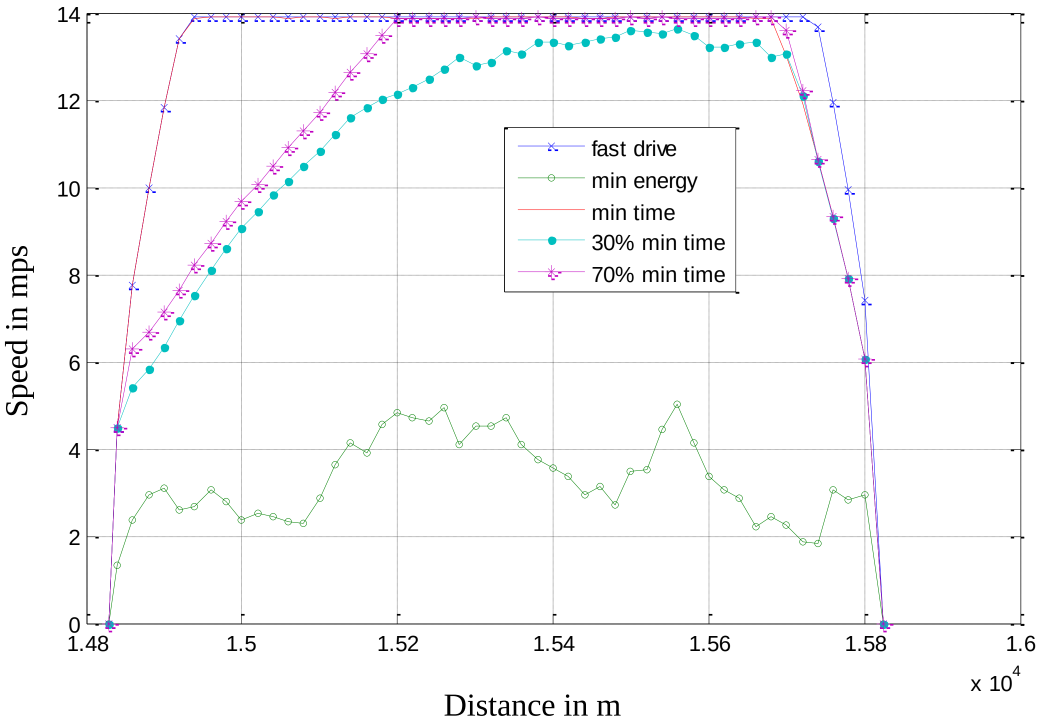


Figure 4-36 Five speed profile for st9-10

19. Between station 8 and 9

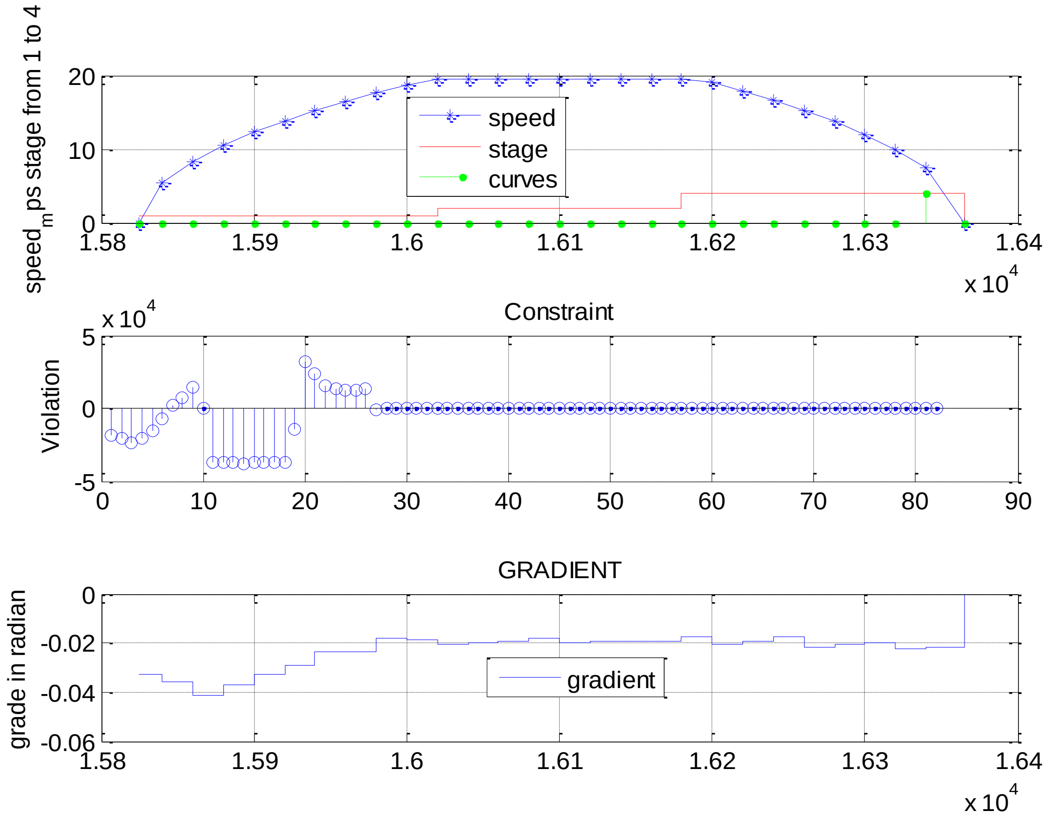


Figure 4-37 Fast drive, gradient and constraint violation for st8-9

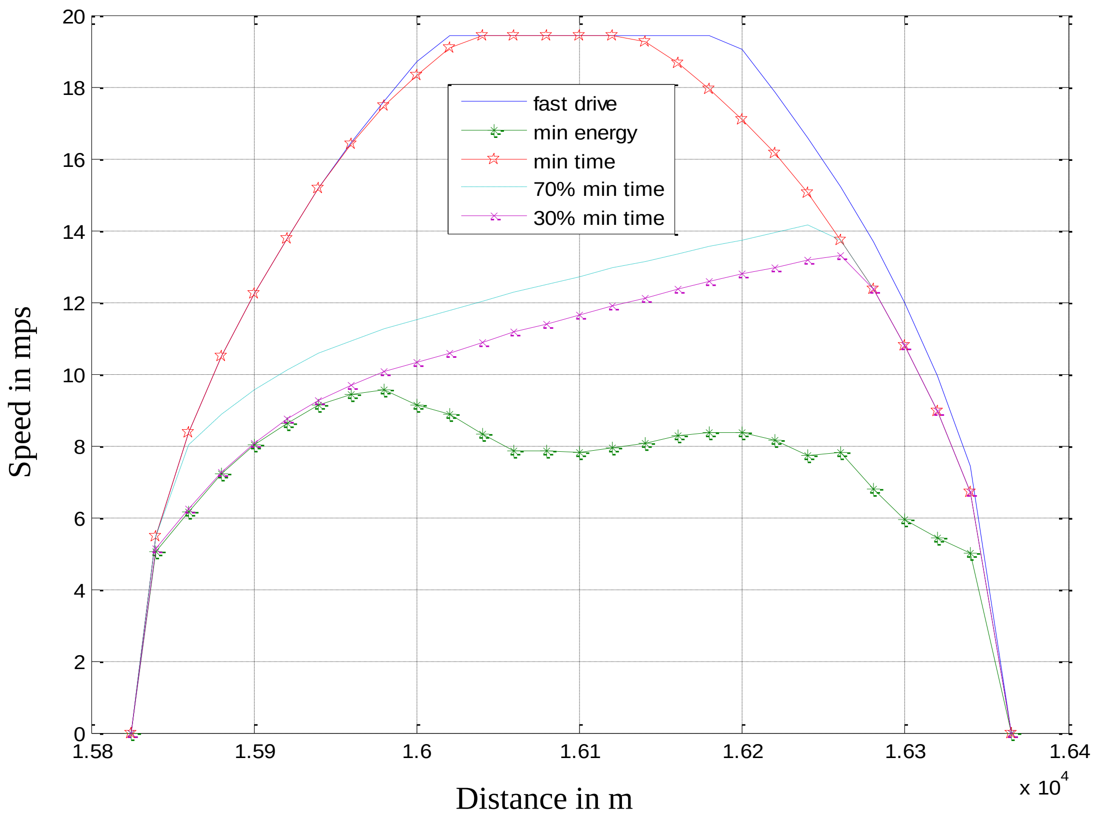


Figure 4-38 Five speed profile for st8-9

20. Between station 7 and 8

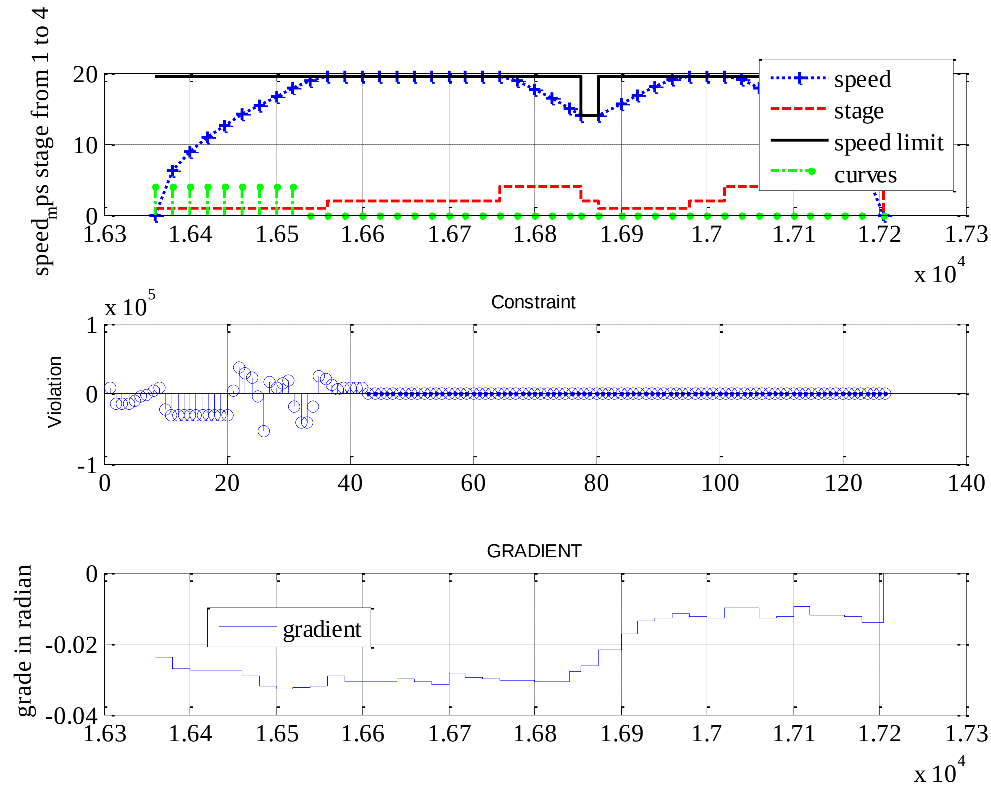


Figure 4-39 Fast drive, gradient and constraint violation for st7-8

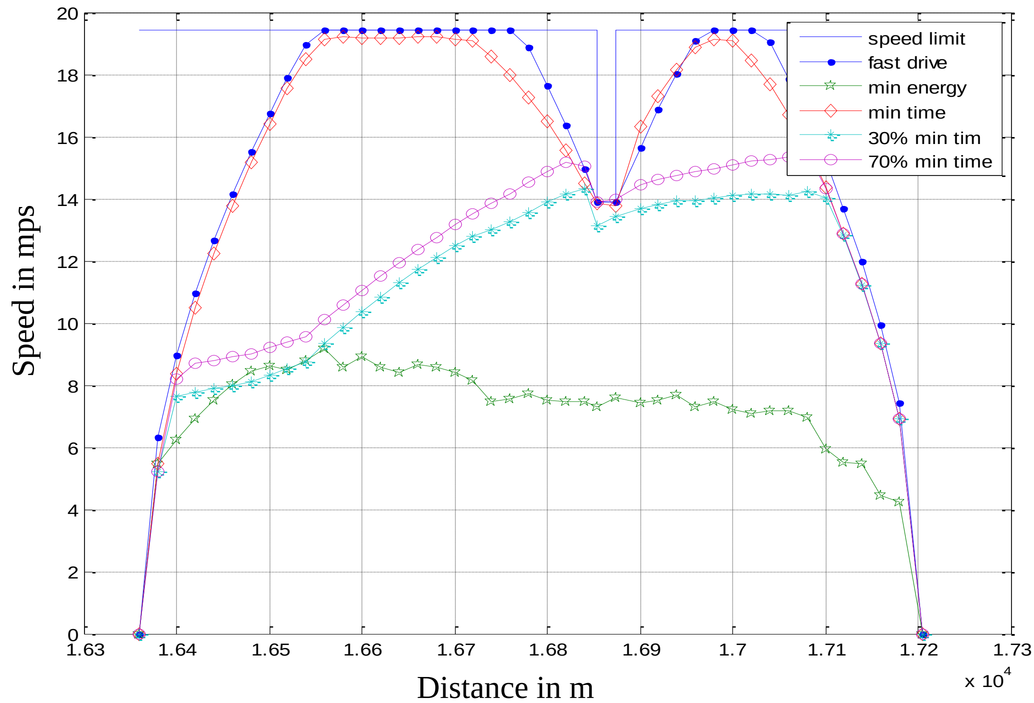


Figure 4-40 Five speed profiles for st7-8

21. Between station 6 and 7

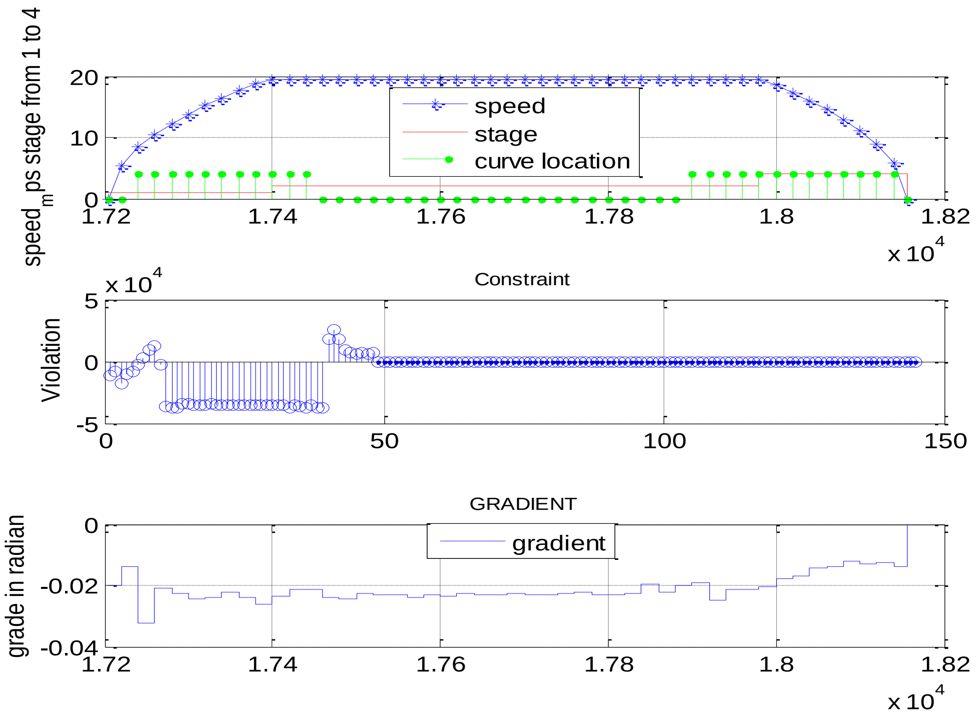


Figure 4-41 Fast drive, gradient and constraint violation for st6-7

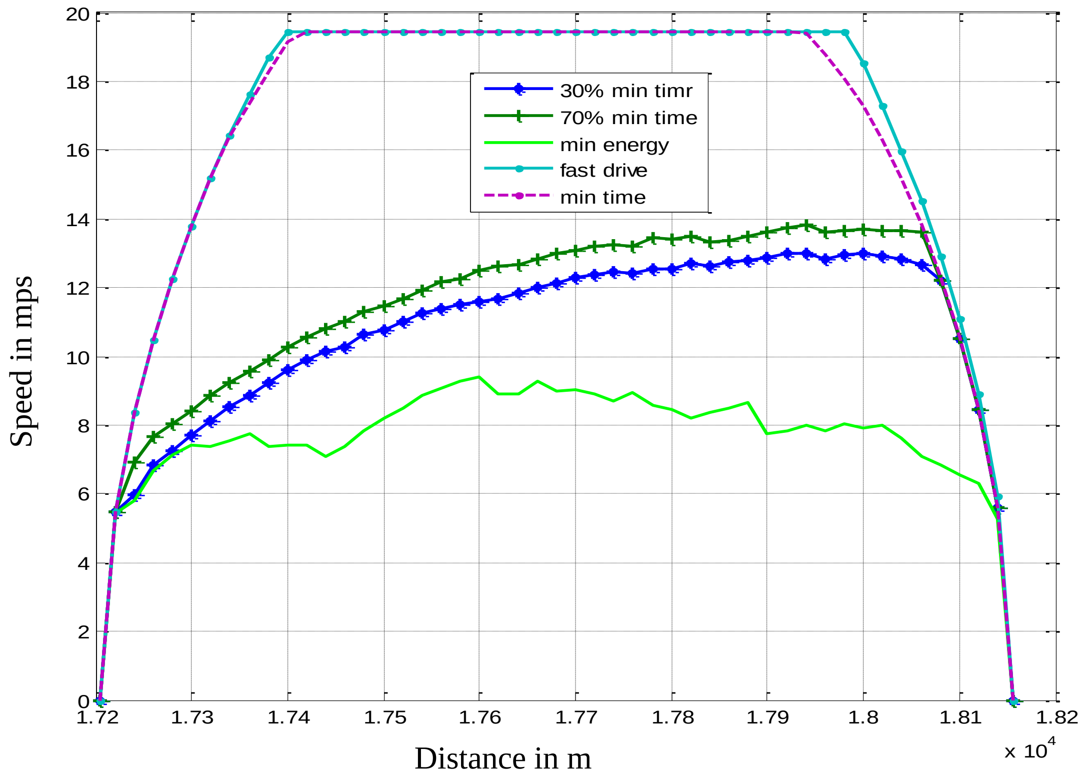


Figure 4-42 Five speed profiles for st6-7

BETWEEN STATIONS	Distance (m)	Discrete pt	FASTEST DRIVE				MINIMUM ENERGY			MINIMUM TIME		
			Time(sec)	power(W)	Energy (J)	Violation	Time(sec)	power(W)	Energy (J)	Time(sec)	power(W)	Energy (J)
7-06	951	49	67.48	1.39E+05	9.40E+06	1.29E+05	129.25	3.05E+03	3.94E+05	68.63	1.34E+05	9.18E+06
7-08	845	43	62.40	2.30E+05	1.43E+07	2.68E+05	126.86	3.14E+03	3.98E+05	65.40	1.96E+05	1.28E+07
9-08	540	28	46.34	1.86E+05	8.61E+06	1.46E+05	81.39	3.34E+03	2.72E+05	48.45	1.73E+05	8.36E+06
10-09	995	51	84.92	5.34E+04	4.53E+06	1.11E+05	342.76	0.00E+00	0.00E+00	88.51	5.12E+04	4.53E+06
10-11	861	45	62.85	2.02E+05	1.27E+07	2.99E+05	146.27	1.02E+04	1.50E+06	65.71	1.91E+05	1.26E+07
12-11	1969	100	119.84	1.79E+05	2.14E+07	2.82E+05	236.08	2.97E+04	7.00E+06	138.18	1.54E+05	2.13E+07
13-12	555	29	47.12	1.95E+05	9.20E+06	1.18E+05	106.59	0.00E+00	0.00E+00	48.13	1.85E+05	8.90E+06
14-13	610	31	49.94	1.52E+05	7.57E+06	1.45E+05	94.60	0.00E+00	0.00E+00	52.41	1.42E+05	7.44E+06
15-14	463	24	42.38	2.01E+05	8.53E+06	2.40E+05	71.12	8.48E+03	6.03E+05	46.40	1.76E+05	8.17E+06
16-15	908	45	119.34	1.36E+05	1.62E+07	3.30E+05	359.96	1.37E+04	4.93E+06	121.16	1.08E+05	1.31E+07
17-16	425	22	40.43	1.98E+05	8.02E+06	7.36E+04	116.30	9.11E+02	1.06E+05	41.32	1.94E+05	8.02E+06
18-17	561	28	47.89	2.93E+05	1.40E+07	3.42E+05	360.00	4.59E+03	1.65E+06	49.82	2.62E+05	1.31E+07
19-18	688	33	53.95	3.25E+05	1.75E+07	3.29E+05	106.77	3.79E+04	4.04E+06	57.43	2.75E+05	1.58E+07
20-19	735	37	56.37	1.44E+05	8.11E+06	7.59E+04	86.29	1.19E+03	1.02E+05	58.13	1.39E+05	8.08E+06
21-20	591	23	48.98	1.62E+05	7.96E+06	2.11E+05	100.13	0.00E+00	0.00E+00	54.67	1.40E+05	7.63E+06
21-22	739	29	56.58	2.60E+05	1.47E+07	4.35E+05	160.07	7.63E+03	1.22E+06	68.39	1.90E+05	1.30E+07
22-23	871	37	63.37	1.30E+05	8.26E+06	3.40E+05	157.21	0.00E+00	0.00E+00	77.12	1.07E+05	8.26E+06
24-23	610	25	49.96	1.61E+05	8.06E+06	1.18E+05	208.47	4.49E+02	9.36E+04	52.65	1.53E+05	8.06E+06
25-24	608	25	49.84	3.27E+05	1.63E+07	2.21E+05	360.00	1.19E+04	4.28E+06	54.34	3.00E+05	1.63E+07
26-25	945	46	67.17	2.79E+05	1.88E+07	1.42E+05	359.96	1.73E+04	6.22E+06	67.81	2.64E+05	1.79E+07
27-26	741	40	56.67	1.71E+05	9.68E+06	2.21E+05	117.00	2.89E+03	3.38E+05	62.11	1.56E+05	9.68E+06
Total sum	16211	790	1293.82	4.12E+06	2.44E+08	4.58E+06	3827.08	1.56E+05	3.31E+07	1386.77	3.69E+06	2.32E+08
Total time in minute		21.56					63.78			23.11		

Table 4-1 MATLAB simulation result for the entire section (fast drive, minimum energy drive, minimum time drive)

BETWEEN STATIONS	Distance (m)	Discrete pt	70% MINIMUM TIME			30 % MINIMUM TIME		
			Time(sec)	power(W)	Energy (J)	Time(sec)	power(W)	Energy (J)
7-06	951.00	49.00	93.10	8.32E+03	7.75E+05	98.55	4467.47	440260.00
7-08	845.00	43.00	80.42	2.15E+04	1.73E+06	85.04	15044.06	1279300.00
9-08	540.00	28.00	57.76	1.68E+04	9.71E+05	62.84	4589.54	288400.00
10-09	995.00	51.00	98.11	6.84E+03	6.71E+05	103.90	4050.11	420810.00
10-11	861.00	45.00	87.66	4.41E+04	3.87E+06	88.37	42492.93	3755300.00
12-11	1969.00	100.00	200.79	4.17E+04	8.37E+06	216.38	33879.93	7330800.00
13-12	555.00	29.00	63.96	7.84E+03	5.01E+05	65.35	5859.76	382910.00
14-13	610.00	31.00	60.89	8.38E+03	5.10E+05	65.65	3431.29	225260.00
15-14	463.00	24.00	53.95	1.89E+04	1.02E+06	54.82	14082.93	772050.00
16-15	908.00	45.00	137.31	4.41E+04	6.05E+06	149.31	35910.28	5361700.00
17-16	425.00	22.00	47.40	3.26E+04	1.55E+06	51.19	12992.93	665160.00
18-17	561.00	28.00	67.69	5.11E+04	3.46E+06	69.91	47889.13	3347800.00
19-18	688.00	33.00	76.65	8.42E+04	6.45E+06	80.02	75043.36	6005300.00
20-19	735.00	37.00	65.80	6.54E+03	4.30E+05	70.43	2181.04	153620.00
21-20	591.00	23.00	66.24	7.11E+03	4.71E+05	68.81	4685.62	322430.00
21-22	739.00	29.00	91.53	3.07E+04	2.81E+06	101.83	20064.99	2043200.00
22-23	871.00	37.00	91.49	3.98E+03	3.64E+05	97.88	1794.80	175670.00
24-23	610.00	25.00	61.94	1.94E+04	1.20E+06	63.56	13608.57	864920.00
25-24	608.00	25.00	67.73	1.15E+05	7.79E+06	77.25	79479.32	6140000.00
26-25	945.00	46.00	95.02	7.35E+04	6.99E+06	98.52	63126.92	6314696.68
27-26	741.00	40.00	74.22	1.66E+04	1.23E+06	75.84	15406.52	1168400.00
Total sum	16211.00	790.00	1739.66	6.59E+05	5.72E+07	1845.44	500081.52	47457986.68
Total time in minute			28.99			30.76		

Table 4-2MATLAB simulation result for the entire section (70% minimum time drive and 30% minimum time drive)

4.2. Discussion of simulation result

For discussion purpose we applied percentage energy and time sacrifices which can be calculated based on the following principle:

$$\%\text{time sacrifice}(\text{time save}) = \frac{\text{Max}_{\text{time}} - \text{Min}_{\text{time}}}{\text{Max}_{\text{time}}} * 100 - - - - - - 4.1$$

$$\%\text{energy sacrifice}(\text{energy save}) = \frac{\text{Max}_{\text{enegry}} - \text{Min}_{\text{energy}}}{\text{Max}_{\text{energy}}} * 100 - - - - - - 4.2$$

%time sacrifice signifies the percentage of extra time needed to complete the given journey by one driving mode compared to the other while saving energy consumed. It can also be defined as time saved by one driving mode to the other driving mode while scarifying energy consumed. %energy sacrifice on the other hand signifies the percentage of extra energy consumed by one driving mode compared to the other while saving journey time.

4.2.1. Discussion of result for selected section

We have 21 sections in North to South line of Addis Ababa light railway project and discussion principle used for one section can be applied for other sections of the track. As a result of this, we selected three sections from the entire sections depending on special result obtained and track situation; for example existence of level crossing.

I. Between station 22 and 23 (about 871 meter):

This section is discretized in to 37 points with continuously varying gradient and curve between [4776.372 4881.827], [4898.839 5079.745], [5100.624 5208.707] and [5275.891 5583.072]. Minimum journey time is obtained from the fastest train driving mode (63.3679sec) and maximum travel time is obtained whenever train is driven to save energy (157.2088 sec). The maximum running time allowed to complete each section of the track is taken to be 360 sec which is the maximum headway between two trains. Due to this, the time difference between minimum energy and minimum time driving mode seems very high. In real phenomena, maximum time required to complete the given journey may not be equal to the maximum headway and we don’t find such amount of time variation.

Energy consumed during minimum energy drive is null, such kind of driving is also noticed for other sections of the line (**Table 4-1, Table 4-2**). The fastest driving mode is equipped with constraint violation and is not applicable in real traffic condition. It provides the best travel time with the worst energy consumption and this energy level is applied to limit the maximum energy consumption in time related minimization model.

Driving type	Between station 22-23			%Energy sacrifice per min energy	Extra energy in Joule	Extra time in sec	%Time sacrifice per min time
Fastest time	63.3679	1.30E+05	8.26E+06	1.00E+02	8.26E+06		
Min time	77.1208	1.07E+05	8.26E+06	1.00E+02	8.26E+06		
70% time	91.4857	3.98E+03	3.64E+05	1.00E+02	3.64E+05	14.36	15.70
30% time	97.8774	1.79E+03	1.76E+05	1.00E+02	1.76E+05	20.76	21.21
Min energy	157.2088	0.00E+00	0.00E+00			80.09	50.94
	Time(sec)	Power (watt)	Energy (joule)				

Table 4-3 Energy and time comparison of train driving modes between station 22 and 23

Time sacrifice of minimum energy driving mode compared to minimum time driving mode is 50.94% (80.1 sec) with energy saving of 100% (8.26E+06J). Why the energy scarification becomes extremely high? From the gradient, we can clearly notice that all portion of the track is negative and this gradient force can act as a tractive effort while driving. During energy minimization, gradient allows the train to accelerate and reach destination without the requirement of high tractive effort. Speed profile obtained from energy minimum driving mode signifies that speed of the train is below 10 mps on the entire track and due to the presence of negative gradient tractive effort required for such speed level is null. Even at the curved section, gradient resistance can counter balance curve resistance or speed of the train reduces not to consume power. For example; between 5400 meter and 5500 meter we have curve with increase of gradient. Speed of the train can't increase or keep because gradient resistance can't counter balance other movement resistance. That is why speed of the train is forced to decrease by the application of braking force. On the other hand, train speed during time minimization driving mode reaches to 19.444 mps and extremely high amount of energy is consumed.

The third model is applied with 70% and 30% time minimization. These two modes imply optimization of 70% of time and 70% of energy respectively. 70% time minimization driving mode sacrifices 100% (3.64e+05J) energy to save 65.7231 sec and

30% time minimization driving mode on the other hand sacrifices 100% (1.76e+05J) in saving of 59.334 sec compared to minimum energy driving mode. Based on the interest of driving strategy one of the three driving model can be applied to obtain optimum train operation. **(Figure 4-9, Figure 4-10,** Table **4-1, Table 4-2)**

II. Between station 26 and 25 (945 meter)

The section is discretized into 46 discrete points with curves at [2598.425 2722.256], [2851.263 2984.33] and [3320.091 3544.619]. The first half section of the track is negative gradient and the next half section is positive gradient. Such kind of gradient distribution is also noticed between station 19 and 20 (**Figure 4-15**). MATLAB simulation is conducted for more than 15000 generation for each type of driving modes.The number of generation required for this section in minimum energy driving mode is considerably higher than the other sections. In energy minimizing driving mode, the primary target is to minimize energy consumed and speed combination which results low energy consumption is selected.To reduce number of generation required we can consider the braking power minimization too. This is applied when we assume regenerative braking. Unnecessary braking can be replaced by coasting mode or braking with small braking effort which can save more time and braking power. For example in our case, if the first section of the track has a chance to increase speed of the train as noticed in 70% or 30% minimum time driving mode (which reduces braking power and journey time, actually not due to the consideration of regenerative braking!), higher speed for the start of positive section is provided. To reduce energy consumed for positive gradient section; either we apply a brake or speed decrease with small tractive effort application which results highly negative accelerative resistance to counter balance gradient force, such situation is noticed after more than 15000 generations. Generally initial population plays an important role on the number of generation required for genetic algorithm.

Between station	25-26			%Energy sacrifice per min energy	Extra energy in Joule	Extra time in sec	%Time sacrifice per min time
Fastest time	49.8372	3.27E+05	1.63E+07	7.38E+01	1.20E+07		
Min time	54.339	3.00E+05	1.63E+07	7.38E+01	1.20E+07		
70% time	67.7284	1.15E+05	7.79E+06	4.51E+01	3.51E+06	13.39	19.77
30% time	77.2528	7.95E+04	6.14E+06	3.04E+01	1.86E+06	22.91	29.66
Min energy	359.999	1.19E+04	4.28E+06			305.66	84.91
	Time(sec)	Power (watt)	Energy (joule)				

Table 4-4 Energy and time comparison of train driving modes between station 25 and 26

The final simulation result shows that 84.91% (305.66 sec) time can be saved in minimum time driving mode compared to minimum energy driving mode, but this is in the expense of huge amount of energy (around 73.8% (1.2E+07J)). 70% minimum time driving mode sacrifice 45.1% (3.51e+06J) energy in saving 292.2706 sec and 30% minimum time driving mode sacrificed 30.4% (1.86e+06J) energy in saving 282.7462 sec compared to minimum energy driving mode.**(Figure 4-3, Figure 4-4, Table 4-1, Table 4-2)**

III. Between station 15 and 16 (about 908 meter)

This section has level crossing between [10069 10170] and train is forced to use lateral switch while reaching the common section of north-south and east-west line ([9464 10069]). There is also curve at [9599.367 9696.561], [9714.106 9735.384], [9751.706 9772.983], [9783.886 9950.583], [9966.824 10015.627], [10061.165 10170.218] and [10185.729 10260.469] with continuously varying gradient.

The section is discretized in to 45 points and speed limit for level crossing is 50km/hr, for lateral switch position is 25 km/hr and for main line is 70km/hr. Train driven in such speed restriction track must obey the restriction and simulation result shows that this rule is obeyed.

Driving type	Between station 15-16			%Energy sacrifice per min energy	Extra energy in Joule	Extra time in sec	%Time sacrifice per min time
Fastest time	119.342	1.36E+05	1.62E+07	6.95E+01	1.13E+07		
Min time	121.1602	1.08E+05	1.31E+07	6.24E+01	8.19E+06		
70% time	137.3104	4.41E+04	6.05E+06	1.85E+01	1.12E+06	16.15	11.76
30% time	149.3082	3.59E+04	5.36E+06	8.02E+00	4.30E+05	28.15	18.85
Min energy	359.9608	1.37E+04	4.93E+06			238.80	66.34
	Time(sec)	Power (watt)	Energy (joule)				

Table 4-5 Energy and time comparison of train driving modes between station 15 and 16

Though fastest train drive can provide the best driving strategy in time, minimum time driving strategy is the one with the best train operation in time (number of passenger transportation). This is because of constraint violation in fastest train driving mode. Time saving by minimum time driving mode compared to minimum energy driving mode is around 66.34% (238.80 sec) with energy sacrifice of 62.4% (8.19E+06J).

The other model is applied to minimize percentage of time and energy both together. For example, in our case, the model is used to minimize 70% of time (30% of energy) on one occasion and 30% of time (70% energy) on another occasion. Simulation result shows that 18.5% (1.12e+06J) energy sacrificed in saving 222.6504 sec for 70% minimum time driving mode and 30% minimum time driving mode sacrificed 8.02% (4.30e+05J) energy in saving 210.6526 sec compared to minimum energy driving mode.**(Figure 4-23, Figure 4-24, Table 4-1, Table 4-2)**

4.2.2. Discussion of the total result

Total time taken to complete the journey from Menelik square to Kality and Energy consumed for each of the driving scheme is compared. Based on time of run; the best driving mode is the fastest driving mode, but this mode is an ideal mode and is not practically applicable in real traffic condition. The other driving modes are obtained from optimization principle and are applicable in real traffic condition. The worst driving mode is obtained from energy minimization model. Here, time of driving is not the primary target and time required to complete the given journey is more. Time sacrifices of minimum energy driving mode against fast driving mode is 66.193% (2533.268 sec), against minimum time driving mode is 63.764% (2440.32 sec), against 70% minimum time driving mode is about 54.54% (2087.42 sec) and 51.78% (1981.64 sec) against 30% minimum time driving mode.

Generally, total time taken to complete the journey from Menelik square to Kality under minimum energy driving mode is 63.78 minute, 21.564 minute under fast driving mode, 23.11 minute under minimum time driving mode, 28.99 minute under 70% minimum time driving mode and 30.76 minute under 30% minimum time driving mode. Total time required in minimum energy driving mode is more than two fold of the other driving modes and it is advisable to use minimum time or proper operational driving modes for more number of passenger transportation.

Generally, maximum time of run between stations is too much (360 sec) for all section of the track and by applying appropriate maximum time of run for each section of the track based on the maximum speed allowed, number of trains and distance to be covered; we can reduce time variation between minimum energy driving mode and other driving modes.

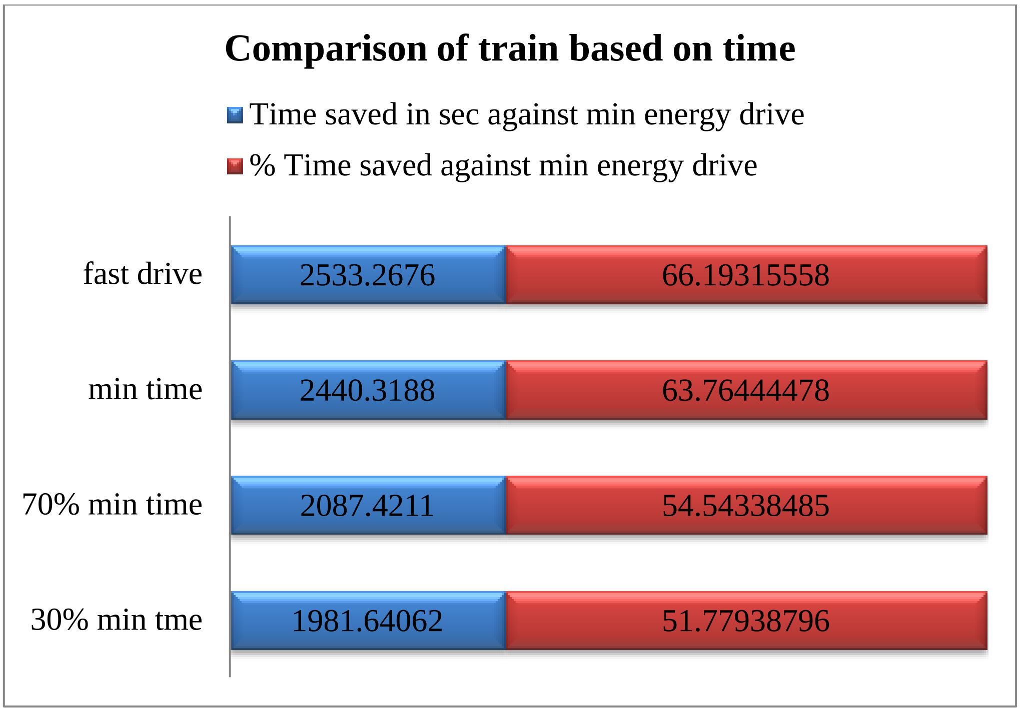


Figure 4-43 Journey time saved against minimum energy drive (journey time sacrifice of minimum energy drive against the other drive)

Based on energy, the best train driving in saving energy is minimum energy driving mode and the worst driving strategy is from the ideal driving (fast train driving). Train driven in fast train driving mode shows constraint violation and is not applicable in real traffic condition. Thus, we can say that the worst driving strategy is obtained from minimum time driving mode. Comparison of energy scarification between different driving modes explains more about each driving strategy related to energy saving. Energy sacrifice of fast train driving against 30% minimum time driving is 80.544% (1.96E+08J), 76.54% (1.87E+08J) against 70% minimum time, 4.788% (1.17E+07J) against minimum time and 86.407% (2.11E+08J) against minimum energy driving mode.

Generally, total energy required for 30% minimum time driving is 4.75E+07J, 5.72E+07J for 70% minimum time driving, 2.32E+08J for minimum time driving, 3.31E+07J for minimum energy driving and 2.44E+08J for fast train driving mode. If the primary target of driving is to save energy, then minimum time and fast train driving strategy is out of the ring.

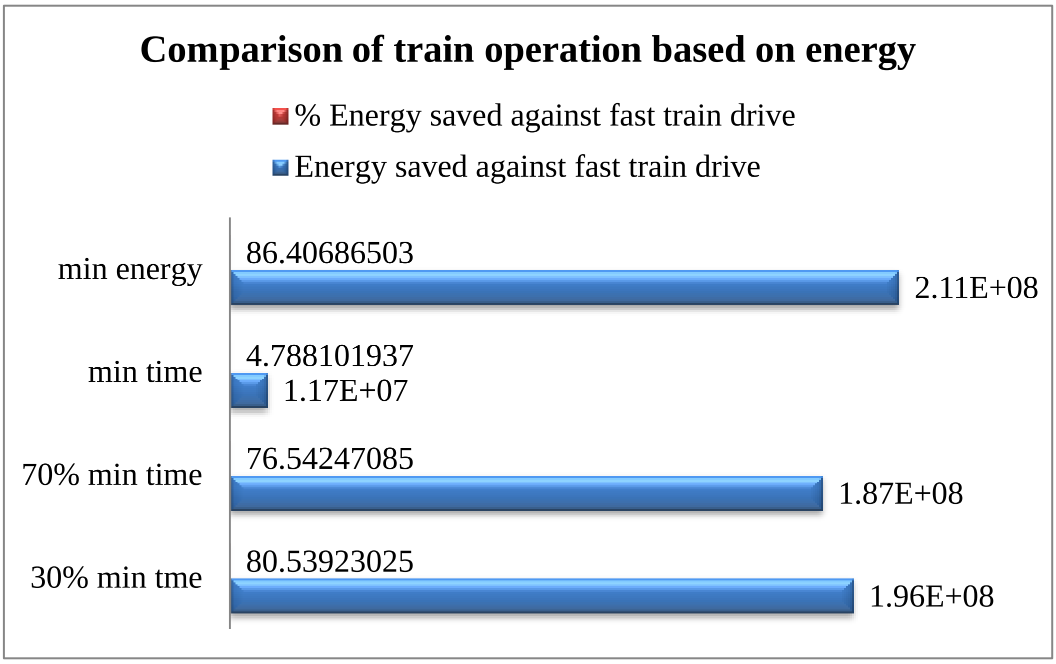


Figure 4-44 Energy saving against fast drive (energy sacrifice of fast drive against the other drive)

Optimum operation of the train can be achieved based on the interest at which optimality is required. As we can see from the total result, if we want to optimize energy by 100%, then we have to sacrifice journey time (extended time is required) to complete the given journey. This in return results low number of passenger transportation. If we want to optimize journey time by 100%, then we have to sacrifice huge amount of energy which results high bill. This driving scheme allows transporting more number of passengers in short period of time.

The other optimization problem applied is to optimize certain percentage in energy and certain percentage in time. The objective function for this mode of driving is formulated to optimize energy and time at the same time which results optimum operation of the train in energy and time rather than time only or energy only (**see 34**). This kind of optimization provides a chance of win-win opportunity. We can select percentage of energy and time to optimize both of them in such a way that the sum of percentage of time and energy must be equal to 100%. This optimization problem didn't allow selecting percentage value of 100, because the system doesn't provide appropriate constraints for this percent value. For 100% either in time or energy the first two extreme optimization problems discussed in the above paragraph must be applied. Generally, this optimization problem helps to transport the required number of passengers with acceptable energy consumption (acceptable bill).

5. Conclusion and recommendation

5.1. Conclusion

Even though we don't find the actual driving speed profile used by ERC, appropriate models with three objective functions subjected to complex constraints have been developed. The problems are designed and solved to find optimum speed profile in terms of energy and/or journey time to complete the given journey and compared among each other. We can call the models developed are speed tuning models in which at each discrete points speed value satisfying all constraints with minimum of the respective objective function is obtained using genetic algorithm.

MATLAB program is developed and implemented for Addis Ababa LRT from Menelik Square to Kality. From simulation result, we noticed that huge amount of energy can be saved in minimum energy driving mode and minimum journey time is obtained from minimum time mode with no constraint violations compared to the other modes. Comparison of the two modes shows that energy saving in minimum energy driving mode is sufficiently large (1.99E+08J) in losing 2440.3188 sec (40.67 minute). We loss 2.41E+07J of energy in saving of 2087.421 sec (34.79 minute) for 70% minimum time driving mode and 1.43E+07J of energy in saving 1981.64 sec (33.027 minute) for 30% minimum time driving mode compared to minimum energy driving mode.

In minimum energy driving mode, we find some of the sections can be driven with zero energy, which seems very ideal, is resulted from the rolling resistance considered and extended travel time constraint.

After all, different speed profile for different operational scenario is suggested. By increasing the number of generation we can have more accurate result. Generally, the models can be applicable for any traffic condition ranging from low to high traffics.

1. During high traffic condition, where the number of passengers to be transported has higher priority than energy consumed, we can apply minimum time driving model. This model allows completing a give journey with in short period of time and more number of passengers can be transported, but energy consumed is somewhat as high as fast train driving mode (ideal driving).

2. During low traffic condition (energy consumption has higher priority than journey time), we can apply minimum energy driving model. This model is extremely slower than any of the models developed and number of passengers to be transported is low.
3. The other model which is very flexible and can be applied for traffic condition other than the two extreme traffics is proper operational model. This model allows railway companies to generate different speed profile depending on the percentage of energy or time minimization required. For example, we apply the model developed for 70% minimum time and 30% minimum time operational scenarios; the result obtained shows the performance expected from the model and can be applicable in real traffic condition.

5.2. Recommendation

- Due to lack of real information, the models developed takes approximation and some datasheet from international railway standards and railway manufacturing companies. Thus, by considering the actual information we can improve the result obtained in this thesis.
- Energy consumption can be modeled based on the catenary voltage and current.
- The number of train and regenerative braking can be considered. Regenerated energy is consumed by the other train in the same section; no damping resistor required.
- Energy saving using Uninterrupted Power Supply system is the other section of energy saving strategy by storing considerable amount of energy.
- MATLAB code developed to solve the model can be modified to overcome computational time.

Reference

[1] "Energy Efficiency Technologies and Benefit", Sustainable Energy Regulation and Policymaking for Africa, Module 12.

[2] *http://www.aceee.org/sector/transportation*, November 26, 2013

[3] *http://www.people.hofstra.edu/geotrans/eng/ch8en/conc8en/ch8c2en.html*, August 19, 2013

[4] "EVENT (Evaluation of Energy Efficiency Technologies for Rolling Stock and Train Operation of Railways)", Final Report Submitted to the Sub-commission Energy Efficiency, Berlin, March 2003.

[5] T. J. Mlinarić and K. Ponikvar," Energy Efficiency of Railway Lines", Transportation Economy Preliminary Communication, Apr. 26, 2010.

[6] Masafumi, Miyatake and Hideyoshi K., "Optimization of Train Speed Profilefor Minimum Energy Consumption", IEEJ Transactions on Electrical and Electronic Engineering, Vol.5, 2010, pp.263–269.

[7] Pengling Wang et al, "Optimization Analysis on the Energy Saving Control for Trains with Adaptive Genetic Algorithm", International Conference on Systems and Informatics, 2012, pp.439-444.

[8] Erik Lindgreen and Spencer C.Sorenson, "Simulation of Energy Consumption and Emission from Rail Traffic", Technical University of Denmark, 2005.

[9] H. Partab, "Modern Electric Traction", Natraj Offset Printer (Delhi), Gagan Kapur for Dhanpat Rai& Co.(P) LTD Educational and Technical publisher,2010.

[10] Piotr Lukaszewicz and Evert Andersson,"Green Train energy consumption Estimations on high-speed rail operations", KTH Railway Group, Stockholm, 2009.

[11] S. Açıkbaş and M. T. Söylemez," Energy Wise Driving of a Mass Transit Train", *http://www.emo.org.tr/ekler/384c4cf5d30e460_ek.pdf*, March 7, 2014.

[12] David G. Luenberger and Yinyu Ye, "Linear and Nonlinear Programming", Third Edition, New York, Springer Science and Business Media, 2008.

[13] Edwin K. P. Chong and Stanislaw H. Zak, "An Introduction to Optimization", Second Edition, USA, A Wiley-Inter-science Publication, 2001.

[14] "Introduction to the Ethiopian Electric Power Corporation (EEPCO) ", *http://www.eepco.gov.et*, August 19, 2013.

[15] The Federal Democratic Republic of Ethiopia Growth and Transformation Plan (GTP) 2010/11-2014/15, Draft Ministry of Finance and Economic Development (MoFED) September 2010 Addis Ababa.

[16] *http://www.ventures-africa.com*, August19, 2013

[17] B. L. Theraja and A. K. Theraja, "A Text Book of Electrical Technology, volume two, AC & DC Machines", 23Rev Ed Edition, S. Chand Publishing, India, July 2008, Chapter 43.

[18] *http://www.engsoc.org/~josh/AREMA/chapter2%20-%20Railway%20industry%20Overview.pdf*, August 19, 2013.

[19] Shaofeng Lu, "Optimizing Power Management Strategies for Railway Traction Systems", Doctoral Thesis, University of Birmingham, October 2011.

[20] Coenraad Esveld, "Modern Railway Track", 2nd Edition, Delft University of Technology, 2001.

[21] Dr.-ing.Thomas Albrecht et al, "Railway Timetable and Traffic Analysis, Modeling and Simulation", *www.eurailpress.de/leseprobe/214/download/magazine.pdf*, September 21, 2013

[22] Ph. Roumeguere, "Railway Management Engineering", Third Edition, England, Ashgate publishing Company, 2006.

[23] Panagiotis Gkortzas,"Study on optimal train movement for minimum energy consumption", Master Thesis, Malardalen University of Sweden, School of Innovation and Design Engineering.

[24] Haichao Nan, "Modelling Driving behaviour in Open Track", Master Thesis, TU Delft University, 2011.

[25] *http://5at.co.uk/uploads/Articles%20and%20papers/CANADIAN%20RESISTANCE%20PAPER.PDF*, December 11, 2013.

[26] Erik Lindgreen &Spencer C. Sorenson, "Driving Resistance from Railroad Train", Technical University of Denmark, Feb 28, 2005.

[27] *http://www.trainsimitalia.net/atdownload/guide/din_traz/train_resistance_calculations.html*, August 19, 2013

[28] T. J. Mlinarić, K. Ponikvar, "Energy Efficiency of Railway Lines", Transportation Economy Preliminary Communication, Vol.23, 2011, No 3, pp.187-193.

[29] *http://www.most.gov.mm/techuni/media/CE_04017_p2chap8, teacher.buet.ac.bd/cfc/lec10_curve_web.pdf*, August 19, 2013

[30] *http://www.gisceu.net/PDF/U74.PDF*, August 19, 2013

[31] Arthur J. Manson, “Railroad Electrification and Electric Locomotive”, Simmons-Boardman, New York, 1923.

[32] Jakub Novák, “Single Train Passing Through a Tunnel “, European Conference on Computational Fluid Dynamics, ECCOMAS CFD 2006.

[33] José A. Lozano, Jesús Félez, Juan de Dios Sanz and José M. Mera, “Railway Traction”, Universidad Politécnica de Madrid, Spain, *http://www.intchopen.com*, March 23, 2014.

[34] Roman Bosquet et al, “Model of High Speed Train Energy Consumption”, World Academy of Science, Engineering and Technology, Vol.78, 2013, pp.1912-1916.

[35] Ethiopian Railway Corporation, Project Brief Outline, 2007.

[36] Addis Ababa E-W & N-S (Phase I) Light Rail Transit Project, Preliminary Design, Chapter XIII Communication System, Vol.1.

[37] Addis Ababa E-W & N-S (Phase I) Light Rail Transit Project, Preliminary Design, Chapter XIV signaling System, Vol.1.

[38] South-north Line and East-west Line (Phase I) Light Rail EPC Project of Addis Ababa, Ethiopia, Technical Specifications of Vehicles, July 2013.

[39] P. G. Howeltt and J. Cheng, “Optimal Driving Strategies for a Train on a Track with Continuously Varying Gradient”, J.Austral.Math.Soc.Ser.B38 (1997), June 1994, pp.388-410.

[40] Jyh-Cherng JONG and Sloan CHONG, “Algorithm for Generating Train Speed Profiles”, .Journal of the Eastern Asia Society for Transportation Studies, Vol.6, 2005, pp.356 - 371.

[41] Xian Li et al, “Energy Constraint Operation Strategy for High Speed Railway”, International Journal of Innovative Computing, Information and Control, Vol.*8*, No 10(A),October 2012, pp. 6869-6883.

[42] Rem Chevrier et al, ” Multi-objective Evolutionary Algorithm for Speed Tuning Optimization with Energy Saving in Railway”, Rail Rome, 2011.

[43] Maialen Larranaga, “Optimization Techniques Applied to Railway Systems”, Master Thesis, Universidad Del Pais Vasco/ Euskal Herriko Unibertsitatea, September 21, 2012.

[44] Dale Litwhiler and Delton Martin, “Ac 2008-1330: An Investigation of Acceleration and Jerk Profiles of Public Transportation Vehicles “, Pennsylvania State University, American Society for Engineering Education, 2008.

[45] John-Jairo Martinez and Carlos Canudas-de-Wit, “A Safe Longitudinal Control for Adaptive Cruise Control and Stop-and-Go Scenarios”, Control Systems Technology, IEEE Transactions, Vol.15, March 200, pp.264-258.

[46] http://www.ingeteam.it/userfiles/allegati/tramway_1235587123.pdf, May 23, 2014.

[47] Stephen J.Chapman, ”Electrical Machinery Fundamentals”, 4th Edition, New York, McGraw-Hill, 2005.

[48] Ion Boldea and Syed A. Nasser, “The Induction Machine Handbook”, United States of America, CRC Press, 2002.

[49] Erwin krayszing, “Advanced Engineering Mathematics”, 10th Edition, United States of America, Laurie Rosa tone, 2011.

[50] S. D. Conte and Carl de Boor, “Elementary Numerical Analysis an Algorithm Approach”, 3rd Edition, United States of America, McGraw-Hill Book Company, 1980.

[51] L. F. Shampine, R. C. Allen, Jr. and S. Pruess, “Fundamental of Numerical Computing”, John Wiley & Sons, Inc., Canada, 1987.

[52] Alan Jeffrey, “Advanced Engineering Mathematics”, United States of America, HARCOURT/Academic Press, 2002.

[53] George W. Collins, “Fundamental of Numerical Method and Data Analysis”, 2nd Edition, 2003.

[54] David A. Coley,” An Introduction to Genetic Algorithm”, Singapore, World Scientific Publishing Co. Pte. Ltd, 1999.

[55] Colien R. Reeves and Jonathan E. Row, “GA Algorithm Principle and Perspective, A Guide to GA Theory”, United States of America, Kluwer Academic Publishers, 2003.

[56] Randy L. Houpt and Sue Ellen Houpt, “Practical Generic Algorithm”, 2ndEdition, United States of America, John Wiley & Sons, Inc, 2004 .

[57] Carlos A. Coello and Efren Mezura Montes, ”Constraint-Handling in Genetic Algorithms Through the Use of Dominance-based Tournament Selection”, Advanced Engineering Informatics, Vol.16, Issue 3, July 2002, Pages 193–203.

[58] http://www.researchgate.net/profile/Khaled_Rasheed/publication/2356412_An_Adaptive_Penalty_Approach_for_Constrained_Genetic-Algorithm_Optimization/links/0fcfd50a0743e3ef5b000000, March 18, 2014.

[59] Kalyanmoy Deb and Soumil Srivastava, “A Genetic Algorithm Based Augmented Lagrangian Method for Accurate, Fast and Reliable Constrained Optimization”,

Journal Computational Optimization and Applications, Vol.53, Issue 3, December 2012, pp. 869-902.

[60] Tetsuyuki Takahama and Setsuko Sakai," Efficient Constrained Optimization by the ε Constrained Rank-Based Differential Evolution", Evolutionary Computation (CEC), IEEE Congress, June 2012, pp.1-8.

[61] Biruk Girma Tessema,"A self-adaptive Genetic Algorithm for Constrained Optimization", Master's Thesis, Oklahoma State University, December 2006.

[62] Kalyanmoy Deb, "An Efficient Constraint Handling Method for Genetic Algorithms", Comput. Methods Appl. Mech. Engrg. 186 (2000), pp.311-338.

[63] Helio J. C. Barbosa and Afonso, "An Adaptive Penalty Method for Genetic Algorithm in Constrained Optimization Problems", *http://cdn.intechopen.com/pdfs-wm/844.pdf*, April 25, 2014.

[64] Hong-Shung Wang et al, "Evolution of Appropriate Crossover and Mutation Operators in a Genetic Process", Applied Intelligence, Vol.16, 2002, pp.7-18.

[65] Omar Al Jadaan, "Improved Selection Operator for GA", Journal of Theoretical and Applied Information Technology, 2005-2008, pp.269-277.

[66] R. Sivaraj, "A Review of Selection Methods in Genetic Algorithm", International Journal of Engineering Science and Technology, Vol.3, No.5, May 2011, pp.3972-3978.

[67] Stjepan Picek et al, "On the Recombination Operator in the Real Coded Genetic Algorithms", Evolutionary Computation (CEC), IEEE Congress, June 2013, pp.3103 - 3110.

[68] Cristian Munteanu and Vasile Lazarescu, "Improving Mutation Capability in Real Coded Genetic Algorithm", *http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.41.6687&rep=rep1&type=pdf*, May 7, 2014.

[69] F. Herrera, "Tackling and Tools for Behavioral Analysis ", Artificial Intelligence Review, Vol.12, 1998, pp. 265-319.

[70] Addis Ababa E-W & N-S (Phase I) Light Rail Transit Project, Signaling System Detailed Design, Part I , Technical, Specification, Chapter 2,Version: ET/AA/LRT/CREC/EPC/DD/XH-01/A.

[71] Piort Lukaszewiczzz, “Energy Consumption and Running Time for Trains, Modelling of Running Resistance and Driver Behavior Based on Full Scale Testing”, Doctoral Thesis, Royal Institute of Technology, Stockholm, 2001.

[72] Kitea Kim, “Optimal Train Control on Various Track Alignment Considering Speed and Scheduling Adherence Constraints”, Doctoral Thesis, New Jersey’s Science and Institute of Technology, January 2010.

Appendix

MATLAB CODE (between station 7 and 8).

This is for section having level crossing and minimum energy drive; all the functions for other train driving are the same with slight modification.

A. Searching section – searches zero speed in the middle of the track and boundary condition violation

```
function [ logic]= searching(data,info)
  logic=false;
 if ~isempty(info)
     [row, ~]=size(info);
     for i=1:row
         if data==info(i,:)
               logic=true;
               break;
         end
     end
 end
end

function logic=searchings(info)
logic=false;
[row, column]=size(info);
distance=[16360,16380,16400,16420,16440,16460,16480,16500,16520,16540,
                16560, 16580,16600,16620,16640,16660,16680,16700,16720,16740, ...
         16760, 16780,16800,16820,16840,16854,16874,16900,16920,16940, ...
         16960, 16980,17000,17020,17040,17060,17080,17100,17120,17140,...
         17160,17180,17205];
    for i=1:row
        for j=1:column
           if distance(i)<16854
               if info(i,j)<0 ||info(i,j)>700/36
                   logic=true;
                   break;
               end
           elseif distance(i)<16874&&distance(i)>=16854
                if info(i,j)<0 ||info(i,j)>500/36
                   logic=true;
                   break;
                end
           else
                if info(i,j)<0 ||info(i,j)>700/36
                   logic=true;
                   break;
                end
           end
        end
    end
```

```
end
```

B. Speed generation function

```
function [generation_mps, generation_binary]= correct_generate(nbit, numpopulation,
numparameter)
generation_binary= round(rand(numpopulation, nbit));
generation_mps=zeros(numpopulation,numparameter);
distance=[16360,16380,16400,16420,16440,16460,16480,16500,16520,16540, ...
              16560 6580,16600,16620,16640,16660,16680,16700,16720,16740, ...
16760,16780,16800,16820,16840,16860,16880,16900,16920,16940, ,...
        16960, 6980,17000,17020,17040,17060,17080,17100,17120,17140,...
        17160,17180,17205];
for i=1:numpopulation
   for j=0:numparameter-1
      for k=1:nbit/numparameter
generation_mps(i,j+1)=generation_mps(i,j+1)+2^(nbit/numparameter-
k)*generation_binary(i,j*nbit/numparameter+k);
      end
      if j==0||j==numparameter-1
         if generation_mps(i,j+1)~=0
            generation_binary(i,
j*nbit/numparameter+1:j*nbit/numparameter+nbit/numparameter)=
zeros(1,nbit/numparameter);
            generation_mps(i,j+1)=0;
         end
      else
         while generation_mps(i,j+1)==0
            generation_binary(i,:)= round(rand(1, nbit));
            for k=1:nbit/numparameter
               generation_mps(i,j+1)=generation_mps(i,j+1)+2^(nbit/numparameter-
k)*generation_binary(i,j*nbit/numparameter+k);
            end
         end
      end
      if distance(j+1)>=16854&&distance(j)<=16874
         generation_mps(i,j+1)=generation_mps(i,j+1)/(2^(nbit/numparameter)-1)*500/36;
      else
         generation_mps(i,j+1)=generation_mps(i,j+1)/(2^(nbit/numparameter)-1)*700/36;
      end
   end
end
```

C. Tractive effort handling function

```
function[tract,tractive,braking,distance,stage]=correct_tractives(velocity)
distance=[16365,16380,16400,16420,16440,16460,16480,16500,16520,16540,...
        16560,16580,16600,16620,16640,16660,16680,16700,16720,16740,...
        16760,16780,16800,16820,16840,16860,16880,16900,16920,16940,...
        16960,16980,17000,17020,17040,17060,17080,17100,17120,17140,...
        17160,17180, 17205];
```

```
gradient_radian=[-0.024009222,-0.027013132,-0.027513876,-0.027513876,...
                 -0.027513876,-0.029016274,-0.032021869,-0.033023985,...
                 -0.032522911,-0.032021869,-0.029016274,-0.031019881,...
                 -0.031019881,-0.031019881,-0.030018017,-0.031019881,...
                 -0.031520859,-0.028515446,-0.029517131,-0.030018017,...
                 -0.030518934,-0.030518934,-0.031019881,-0.031019881,...
                 -0.028014647,-0.026512416,-0.022007102,-0.017503574,...
                 -0.013501641,-0.013001465,-0.011501014,-0.012501302,...
                 -0.013001465,-0.010000667,-0.010000667,-0.013001465,...
                 -0.012501302,-0.009500572,-0.012001152,-0.012001152,...
                 -0.012501302,-0.01400183];
% size is 43
% curve 16333.318   16533.748
curve=30.00038889;
Mass=59240;
a=1153.136;
b=1.95492;
c=0.91;
[numpop, numpara]=size(velocity);
tract=zeros(numpop,numpara);
gradient_resistance=zeros(1,numpara-1);
other_resistance=zeros(numpop,numpara-1);
accelarative_resistance=zeros(1,numpara-1);
curve_resistance=zeros(1,numpara);
stage=zeros(numpop,numpara);
tractive=zeros(numpop,numpara);
braking=zeros(numpop,numpara);
 for j=1:numpop
   for i=1:numpara-1
       if distance(i)>=16333.318&& distance(i)<=16533.748
           if i==1
            curve_resistance(i)=0.0004*Mass*9.8*(distance(i)-16333.318)...
                   *curve(1)/200.429;
           else
            curve_resistance(i)=0.0004*Mass*9.8*(distance(i)-...
                   distance(i-1))*curve(1)/200.429;
           end
           other_resistance(j,i)=(a+b*(velocity(j,i))+...
               c*(velocity(j,i))^2)+curve_resistance(i);
       else
          other_resistance(j,i)=(a+b*(velocity(j,i))+c*(velocity(j,i))^2);
       end         gradient_resistance(i)=Mass*9.8*sin(gradient_radian(i));
accelarative_resistance(i)=1.08*Mass*(velocity(j,i+1)^2- velocity(j,i)^2)/...
     (2*(distance(i+1)-distance(i)));
 tract(j,i)= accelarative_resistance(i)+  gradient_resistance(i)+...
     other_resistance(j,i);

% STAGE DETERMINATION FOR POSITIVE GRADIENT

      if gradient_radian(i)>=0
```

```
    if abs(velocity(j,i+1)-velocity(j,i))<=1e-4
        stage(j,i)=2;
        tractive(j,i)= tract(j,i);
    elseif velocity(j,i+1)>velocity(j,i)
        stage(j,i)=1;
        tractive(j,i)= tract(j,i);
    else
        if  tract(j,i)<-1e-4
             braking(j,i)=tract(j,i);
             stage(j,i)=4;
        elseif abs(tract(j,i))<=1e-4
             stage(j,i)=3;
        else
            stage(j,i)=1.5;
            tractive(j,i)= tract(j,i);
       end
    end
else
    if abs(gradient_resistance(i))>other_resistance(j,i)
         if abs(velocity(j,i+1)-velocity(j,i))<=1e-4
                stage(j,i)=2;
                braking(j,i)= tract(j,i);
         elseif velocity(j,i+1)<velocity(j,i)
                stage(j,i)=4;
                braking(j,i)=tract(j,i);
          else
                if tract(j,i)<-1e-4
                     braking(j,i)=tract(j,i);
                     stage(j,i)=3.5;
                elseif abs(tract(j,i))<=1e-4
                     stage(j,i)=3;
                else
                    stage(j,i)=1;
                    tractive(j,i)= tract(j,i);
                end
         end
    elseif abs(gradient_resistance(i))<other_resistance(j,i)
        if velocity(j,i+1)>velocity(j,i)
            stage(j,i)=1;
            tractive(j,i)= tract(j,i);
        elseif velocity(j,i+1)==velocity(j,i)
                stage(j,i)=2;
                tractive(j,i)= tract(j,i);
        else
                if tract(j,i)<-1e-4
                     braking(j,i)=tract(j,i);
                     stage(j,i)=4;
                elseif abs(tract(j,i))<=1e-4
                    stage(j,i)=3;
                else
```

```
                    stage(j,i)=1.5;
                    tractive(j,i)= tract(j,i);
                end
            end
        else
            if velocity(j,i+1)>velocity(j,i)
                    tractive(j,i)=tract(j,i);
                    stage(j,i)=1;
            elseif velocity(j,i+1)==velocity(j,i)
                stage(j,i)=2;
                tractive(j,i)= tract(j,i);
            else
                stage(j,i)=4;
                braking(j,i)=tract(j,i);
            end
        end
    end
  end
end
```

D. Constraint handling function

```
function [constraint, cequal]=correct_constraints(velocity)
[ tract, ~, ~,distance, ~]= correct_tractives(velocity);
[numpop, numpara]=size(velocity);
const1=zeros(numpara,numpop);% tractive effort constraint
const2=zeros(numpara-1,numpop);%acceleration and braking constraint
const3=zeros(numpara-2,numpop);%jerk constraint
const4=zeros(1,numpop);
for i=1: numpop
  for j=1:numpara-1
    %Tractive constraint
      if tract(i,j)>=0
        if velocity(i,j)<=11.39
            const1(j,i)=tract(i,j)-64830;
        else
            const1(j,i)=tract(i,j)-109777+3949.2*velocity(i,j);
        end
      else
        if velocity(i,j)<=15.697
            const1(j,i)=abs(tract(i,j))-68470;
        else
            const1(j,i)=abs(tract(i,j))-158408+5731.2*velocity(i,j);
        end
      end
     %Acceleration constraint
        if (velocity(i,j+1)>=velocity(i,j))
            const2(j,i)=(velocity(i,j+1)^2-velocity(i,j)^2)...
                /(2*(distance(j+1)-distance(j)))-1;
        else
```

```
            const2(j,i)=abs((velocity(i,j+1)^2-velocity(i,j)^2)...
               /(2*(distance(j+1)-distance(j))))-1.1;
          end
          %Time constraint
          const4(i)=const4(i)+2*(distance(j+1)-distance(j))/...
             (velocity(i,j+1)+velocity(i,j));
          % Jerk constraint
          if j>1&&j<numpara-1
             const3(j-1,i)=abs((velocity(i,j+1)-2*velocity(i,j)+velocity(i,j-1))*velocity(i,j-
1)^2/(distance(j)-distance(j-1))^2)-1;
          end
  end
  const4(i)=const4(i)-360;
end
      constraint=[const1;const2;const3; const4];
      cequal=[];
```

E. Actual objective function

```
function [fitness]=correct_objective(velocity)
[numpop, numpara]=size(velocity);
tractive_power=zeros(numpop,1);
 [~,tractive, ~,distance]= correct_tractives(velocity);
for i=1:numpop
   for j=1:numpara-1
    tractive_power(i)=tractive_power(i)+((tractive(i,j)+tractive(i,j+1))/2)...
        *(distance(j+1)-distance(j));
   end
end
fitness=tractive_power;
```

F. Updated objective function by Debs' method

```
function [feasible_fitness,fitness_new,penalty,feasible_population]=
new_objective(population)
constraint=correct_constraints(population);
fitness=correct_objective(population);
[numpop,~]=size(population);
[cons,~]=size(constraint);
penalty=zeros(numpop,1);
feasible_pop=0;
 for i=1:numpop
    for j=1:cons
       if constraint(j,i)>1e-4
          penalty(i)=penalty(i)+(constraint(j,i));
       end
    end
   if penalty(i)==0
     feasible_pop=feasible_pop+1;
     feasible_fitness(feasible_pop)=fitness(i);
```

```
      feasible_population(feasible_pop,:)=population(i,:);
    end
 end
if feasible_pop==0
    feasible_fitness=[];
    feasible_population=[];
    fitness_new=penalty;
else
     max_feasible_fitness=max(feasible_fitness);
    for i=1:numpop
        if penalty(i)==0
            fitness_new(i)=fitness(i);
        else
            fitness_new(i)=max_feasible_fitness+penalty(i);
        end
    end
end
```

G. Selection function

```
 function [elite,parent1,parent2,
fitness_new,feasible_fitness,feasible_population]=correct_select(population, numelite)
[feasible_fitness,fitness_new,penalty,feasible_population]= new_objective(population);
[numpop,numpara]=size(population);
elite=zeros(numelite,numpara);
if ~isempty(feasible_population)
    [~, index_feasible]=sort(feasible_fitness);
    num_feasible=length(feasible_fitness);
    if numelite<=num_feasible
        for i=1:numelite
            elite(i,:)=feasible_population(index_feasible(i),:);
        end
    else
        for i=1:num_feasible
            elite(i,:)=feasible_population(i,:);
        end
        [~,index]=sort(fitness_new);
        for i=num_feasible+1:numelite
            for k=1:numpop
                if ~searching(population(index(k),:),elite)
                    break;
                end
            end
            elite(i,:)=population(index(k),:);
        end
    end
else
    [~,index]=sort(fitness_new);
    for i=1:numelite
        elite(i,:)=population(index(i),:);
```

```
    end
end
numchild=ceil((numpop-numelite)/2);
for i=1:numchild
    select1=ceil(rand*numpop);
    select2=ceil(rand*numpop);
    feasi1=searching(population(select1,:),feasible_population);
    feasi2=searching(population(select2,:),feasible_population);
    if feasi1==true&& feasi2==true
        if fitness_new(select1)>=fitness_new(select2)
            parent1(i,:)=population(select2,:);
        else
            parent1(i,:)=population(select1,:);
        end
    elseif feasi1==false&& feasi2==true
        parent1(i,:)=population(select2,:);
    elseif feasi1==true&& feasi2==false
        parent1(i,:)=population(select1,:);
    else
        if penalty(select1,:)>=penalty(select2,:)
            parent1(i,:)=population(select2,:);
        else
            parent1(i,:)=population(select1,:);
        end
    end
% PARENT TWO SELECTION
    select1=ceil(rand*numpop);
    select2=ceil(rand*numpop);
    feasi1=searching(population(select1,:),feasible_population);
    feasi2=searching(population(select2,:),feasible_population);
    if feasi1==true&& feasi2==true
        if fitness_new(select1)>=fitness_new(select2)
            parent2(i,:)=population(select2,:);
        else
            parent2(i,:)=population(select1,:);
        end
    elseif feasi1==false&& feasi2==true
        parent2(i,:)=population(select2,:);
    elseif feasi1==true&& feasi2==false
        parent2(i,:)=population(select1,:);
    else
        if penalty(select1,:)>=penalty(select2,:)
            parent2(i,:)=population(select2,:);
        else
            parent2(i,:)=population(select1,:);
        end
    end
end
```

H. Crossover function

```
function [elite, child,fitness_new,feasible_fitness,feasible_population ]=
correct_crossover(population,pc,numelite)
[elite,parent1,parent2,
fitness_new,feasible_fitness,feasible_population]=correct_select(population, numelite);
[numparent, ~]=size(parent1);
[numpop,  numpara]=size(population);
child1=zeros(numparent,numpara);
child2=zeros(numparent,numpara);
if ~isempty(parent1)
   for i=1:numparent
      if rand<pc
        mask1=rand(1,numpara);
         mask2=ones(1,numpara)-mask1;
         child1(i,:)=mask1.*parent1(i,:)+mask2.*parent2(i,:);
         child2(i,:)=mask2.*parent1(i,:)+mask1.*parent2(i,:);
         while searching(0,child1(i,2:numpara-1))||searching(0,child2(i,2:numpara-
1))||searchings(child1(i,:))||searchings(child2(i,:))
             mask1=rand(1,numpara);
             mask2=ones(1,numpara)-mask1;
             child1(i,:)=mask1.*parent1(i,:)+mask2.*parent2(i,:);
             child2(i,:)=mask2.*parent1(i,:)+mask1.*parent2(i,:);
        end
      else
          child1(i,:)=parent1(i,:);
          child2(i,:)=parent2(i,:);
      end
   end
      child=[child1; child2];
      [numchild,~]=size(child);
      if numchild+numelite>numpop
        [ fit]= correct_objective(child);
         [~, index]=max(fit);
         child(index,:)=[];
      end
else
   child=[];
end
```

I. Mutation function

```
function mutate= correct_mutation(population, mp,gen, maxgeneration)
%generation dependant mutation
% mp must be very small which is less that 1/numpara
% we use non uniform mutation schem
% b is choosen to be 4, to make the dependency of mutation on generation
% high
distance=[16360,16380,16400,16420,16440,16460,16480,16500,16520,16540,16560,...
        16580,16600,16620,16640,16660,16680,16700,16720,16740,16760,...
```

```
        16780,16800,16820,16840,16860,16880,16900,16920,16940,16960,...
        16980,17000,17020,17040,17060,17080,17100,17120,17140,17160,...
        17180,17205];
[numpop, numpara]=size(population);
logic=false;
for i=1:numpop
   for j=2:numpara-1
       if rand<mp
          multiplier=rand*(1-gen/maxgeneration)^4;
            if rand<0.5
               if distance(j)<16854
                  population(i,j)=population(i,j)+ (700/36-population(i,j))*multiplier;
                  if population(i,j)<0||population(i,j)>700/36
                     logic=true;
                  end
               elseif distance(j)>=16854&&distance(j)<=16874
                   population(i,j)=population(i,j)+ (500/36-population(i,j))*multiplier;
                   if population(i,j)<0||population(i,j)>500/36
                     logic=true;
                  end
               else
                   population(i,j)=population(i,j)+ (700/36-population(i,j))*multiplier;
                   if population(i,j)<0||population(i,j)>700/36
                     logic=true;
                  end
               end
            else
               population(i,j)=population(i,j)- ( population(i,j))*multiplier;
               if distance(j)<16854
                  if population(i,j)<0||population(i,j)>700/36
                     logic=true;
                  end
               elseif distance(j)>=16854&&distance(j)<=16874
                  if population(i,j)<0||population(i,j)>500/36
                     logic=true;
                  end
               else
                  if population(i,j)<0||population(i,j)>700/36
                     logic=true;
                  end
               end
            end
            while population(i,j)==0||logic
                  if rand<mp
                   multiplier=rand*(1-gen/maxgeneration)^4;
                   if rand<0.5
                       if distance(j)<16854
                          population(i,j)=population(i,j)+ (700/36-population(i,j))*multiplier;
                          if population(i,j)<0||population(i,j)>700/36
                             logic=true;
```

```
                end
            elseif distance(j)>=16854&&distance(j)<=16874
                population(i,j)=population(i,j)+ (500/36-population(i,j))*multiplier;
                if population(i,j)<0||population(i,j)>500/36
                    logic=true;
                end
            else
                population(i,j)=population(i,j)+ (700/36-population(i,j))*multiplier;
                if population(i,j)<0||population(i,j)>700/36
                    logic=true;
                end
            end
        else
            population(i,j)=population(i,j)- ( population(i,j))*multiplier;
            if distance(j)<16854
                if population(i,j)<0||population(i,j)>700/36
                    logic=true;
                end
            elseif distance(j)>=16854&&distance(j)<=16874
                if population(i,j)<0||population(i,j)>500/36
                    logic=true;
                end
            else
                if population(i,j)<0||population(i,j)>700/36
                    logic=true;
                end
            end
        end
       end
      end
     end
    end
end
mutate=population;
```

J. Genetics function

```
function  [speed, fitnesss,actual_energy, populations,feasible_generation,total_time]=
correct_geneticss( generation, populations)
[~, numpara]=size(populations);
speed=zeros(generation,numpara);
fitnesss=zeros(generation,1);
feasible_generation=zeros(generation,1);
for i=1:generation
   [elite, child,fitness,feasible_fitness,feasible_populations]=
correct_crossover(populations,0.9,10) ;
   if ~isempty(feasible_fitness)
       feasible_generation(i)=length(feasible_fitness);
       [fitnesss(i), index]=min(feasible_fitness);
       speed(i,:)=feasible_populations(index,:);
```

```
    else
        [fitnesss(i), index]=min(fitness);
        speed(i,:)=populations(index,:);
    end

    mutate=correct_mutation(child,0.02,i,generation);
    populations=[elite;mutate];
end
[~, tractive,~,distance, stage]= correct_tractives(speed(generation,:));
constraint=correct_constraints(speed(generation,:));
total_time=360+constraint(length(constraint));
actual_energy=0;
for i=1:length(distance)-1
actual_energy=actual_energy+(distance(i+1)-distance(i))*(tractive(i)+tractive(i+1))/2;
end
subplot(3,1,1)
hold off
plot(distance,speed(generation,:), 'r');
title('speed')
ylabel('speed in mps&& stage')
grid on
hold on
stairs(distance,stage)
grid on
legend('speed', 'stage')
subplot(3,1,2)
hold off
plot(fitnesss)
ylabel('fitness')
legend('fitness')
grid on
subplot(3,1,3)
hold off
stem(feasible_generation)
legend('feasible pop')
grid on
ylabel('feasible generation')
```

K. Maximum drive function

```
function [total_time,fitness, energy, constraint,total_violation,velocity, speed_limit]=
maximum()
distance=[16365,16380,16400,16420,16440,16460,16480,16500,16520,16540,...
          16560,16580,16600,16620,16640,16660,16680,16700,16720,16740,...
          16760,16780,16800,16820,16840,16860,16880,16900,16920,16940,...
          16960,16980,17000,17020,17040,17060,17080,17100,17120,17140,...
          17160,17180,17205];
gradient_radian=[-0.024009222,-0.027013132,-0.027513876,-0.027513876,...
               -0.027513876,-0.029016274,-0.032021869,-0.033023985,...
               -0.032522911,-0.032021869,-0.029016274,-0.031019881,...
```

```
        -0.031019881,-0.031019881,-0.030018017,-0.031019881,...
        -0.031520859,-0.028515446,-0.029517131,-0.030018017,...
        -0.030518934,-0.030518934,-0.031019881,-0.031019881,...
        -0.028014647,-0.026512416,-0.022007102,-0.017503574,...
        -0.013501641,-0.013001465,-0.011501014,-0.012501302,...
        -0.013001465,-0.010000667,-0.010000667,-0.013001465,...
        -0.012501302,-0.009500572,-0.012001152,-0.012001152,...
        -0.012501302,-0.01400183];
velocity=zeros(1,length(distance));
 for j=length(distance):-1:2
    speed=sqrt(velocity(j)^2+2.2*(distance(j)-distance(j-1)));
    if distance(j)<=16874
      break;
    else
       if speed>700/36
        if velocity(j-2)<700/36
          velocity(j-1)=700/36;
          speed_limit(j-1)=700/36;
        end
        break;
       else
        velocity(j-1)=speed;
       end
    end
 end
for i=2:length(distance)
   speed=sqrt(velocity(i-1)^2+2*(distance(i)-distance(i-1)));
   if distance(i)<16854
      if speed>700/36
       if velocity(i-1)<=700/36
         velocity(i)=700/36;
       end
      else
       velocity(i)=speed;
      end
   elseif distance(i)==16854
      velocity(i)=500/36;
      for j=i:-1:2
        speed=sqrt(velocity(j)^2+2.2*(distance(j)-distance(j-1)));
       if speed>700/36
          if velocity(j-2)<700/36
             velocity(j-1)=700/36;
          end
          break;
       else
          velocity(j-1)=speed;
       end
      end
   else
      if distance(i)<=16874&&distance(i)>16854
```

```
            velocity(i)=500/36;
        else
            speed=sqrt(velocity(i-1)^2+2*(distance(i)-distance(i-1)));
            if velocity(i+1)~=0
                velocity(i)=velocity(i+1);
                break;
            else
                if speed>700/36
                  if velocity(i-1)<=700/36
                     velocity(i)=700/36;
                  end
                else
                  velocity(i)=speed;
                end
            end
        end
    end
end
% SPEED LIMIT
for i=1:length(distance)
    if distance(i)<16854
        speed_limit(i)=700/36;
    elseif distance(i)>=16854&&distance(i)<16874
        speed_limit(i)=500/36;
    else
        speed_limit(i)=700/36;
    end
 end
[~, tractive,~,~, stage]=correct_tractives(velocity);
constraint=correct_constraints(velocity);
total_violation=0;
for i=1:length(constraint)
    if constraint(i)>1e-4
        total_violation=total_violation+constraint(i);
    end
end
% normalized fitness
total_time=360+constraint(length(constraint));
[~,fitness]= new_objective(velocity);
energy=0;
for i=1:length(velocity)-1
    energy=energy+(tractive(i+1)+tractive(i))*(distance(i+1)-distance(i))/2;
end
subplot(3,1,1)
hold off
plot(distance,velocity)
grid on
hold on
stairs(distance,stage,'r')
stairs(distance,speed_limit,'g')
```

```
legend('speed','stage', 'speed limit')
ylabel(' speed_mps stage from 1 to 4')
grid on
hold on
subplot(3,1,2)
stem(constraint)
title( 'Constraint')
ylabel(' Violation')
grid on
subplot(3,1,3)
gradient_radian(length(distance))=0;
stairs(distance, gradient_radian)
title('GRADIENT')
legend('gradient')
ylabel('grade in radian')
grid on
```

L. Function call

% for maximum drive

```
[total_time,fitness, energy, constraint,total_violation,velocity, speed_limit]= maximum();
```

% For genetics

```
[Populations]= correct_generate(430,100,43);
[speed, fitnesss,actual_energy,
disp(' input generation number')
generation=input(' ');
populations,feasible_generation,total_time]= correct_geneticss( generation, populations);
```