Journal of the Eastern Asia Society for Transportation Studies, Vol. 6, pp. 356 - 371, 2005

ALGORITHMS FOR GENERATING TRAIN SPEED PROFILES

Jyh-Cherng JONG
Ph. D., P.E., Research Scientist
Civil, Hydraulics and Informatics Research Center
Sinotech Engineering Consultants, Inc.
171 Nanking E. RD. Sec. 5, Taipei, 105, Taiwan(R.O.C.)
Fax:+886-2-2765-5010
E-mail:jcjong@sinotech.org.tw

Sloan CHANG
Research Scientist
Civil, Hydraulics, and Informatics Research Center
Sinotech Engineering Consultants, Inc.
171 Nanking E. RD. Sec. 5, Taipei, 105, Taiwan(R.O.C.)
Fax: +886-2-2765-5010
E-mail: slchang@sinotech.org.tw

Abstract: In this research, a train simulator (TrainSim) is developed using Object-Oriented Programming (OOP) concepts with C++ language. Two algorithms for generating train speed profiles are embedded in the system. Both generate speed profiles obeying equation of motion and physical constraints of train and railway geometries.

The shortest time speed profile generated from TrainSim is compared with a commercial Train Operation Model (TOM). The result shows that the difference is only 0.16%. When compared with manual calculation by the expert from Taiwan Railway Administration (TRA), it is found that the difference in proper operation time between TrainSim and the TRA expert is -0.12% on average. These results demonstrate that the proposed algorithms are robust and efficient, and are very useful and flexible for train operation research and applications.

Key Words: Train Performance Simulator, Speed Profiles, Object-Oriented Modeling

1. INTRODUCTION

Train speed profiles are extensively used in railway operation and research applications, such as train performance calculation, journey time estimation, energy consumption evaluation, capacity analysis, train scheduling, new route planning, old route upgrading, etc. They are 2 dimensional descriptions of velocity-distance and time-distance to record the train dynamics along its journey. During the trip from its start station to end station, by alternating powering, coasting, and braking modes, the train has four operation regimes, including acceleration, deceleration, constant speed, and stop. The movement of a train is very complex and governed by many factors. As a result, a precise and reasonable speed profile cannot be easily obtained from analytical approaches (Jong 2003). In early stages, this task is done by hand calculation with worksheets or graphic method with transparencies (AREMA, 1999; Hay, 1982; Jong, 2001). Since this is a labor intensive and time-consuming job, a modern approach is to build up a train performance simulator to perform such computations (Howard et al, 1983).

After more than a decade’s development, many computer models have been developed. For example, Howard et al. (1983) review 27 existing Train Performance Simulators (TPS) and develop a generic TPS model in their research. Capillas (1987) introduces a train running simulation program to compute train dynamics and electric demand. Goodman (1987) develops a program which includes single-train simulation and multiple-train simulation. The single-train simulation module is capable of assessing traction options to achieve a specified performance and basic energy costs. Uher (1987) presents a Train Operation Model (TOM), which consists of a train performance simulator and a train movement model. Kikuchi (1991) develops a model to simulate train movements and compute the relationships among speed, time, and distance. Unlike other models that use detailed tractive effort and resistance to compute train dynamics, Kikuchi's model employs acceleration – speed relation for the calculation. Recently, Chen and Ku (2000) present a framework of train speed planner and TPS using Matlab/Simulink. Since the system is developed in the Matlab environment, it is easily programmed but less efficient.

Although many computer models have been developed and some of them become commercial software (e.g., TOM), their computation algorithms are not clearly revealed in the public domain due to business secrets. Especially, they may not take practical rules into considerations and thus, are difficult to apply to specific railway systems. In addition, as mentioned in Howard et al. (1983), most of the programs are written in FORTRAN language, which is designed for procedural programming and thus, is not flexible for handling complicated system with many objects. For these reasons, a train simulator (TrainSim) is developed in this study using Object-Oriented Programming (OOP) concepts with C++ language. TrainSim is evolved from the model introduced in Jong et al. (2003). Currently two algorithms for generating train speed profiles are embedded in the program. One is to create a speed curve with the shortest operation time, whereas the other is to produce a speed profile with a proper operation time that may have possible energy saving. The latter is designed to mimic the practices of Taiwan Railway Administration (TRA) experts in preparing speed profiles that are actually put into real operation. Both algorithms generate speed profiles obeying equation of motion and physical constraints of train and railway geometries. The main features that differentiate the proposed models and algorithms from previous researches include (1) considering neutral section effect, (2) taking starting and tunnel resistances into considerations, (3) automatically determining speed limits based on rolling stock, curvature and downgrade speed rules, (4) providing shortest time and proper time operation strategies, and (5) providing options to change train states by time, space, and velocity increments.

The remainder of this paper is organized as fallows: In section 2, the mathematical model for train operation is presented. Next, the algorithms for generating speed profiles are developed. From that, the framework of TrainSim is outlined. In section 4, the important components, such as the data structure and class relationships, are introduced using Unified Modeling Language (UML), a new standard for system analysis and design in software engineering (Rumbaugh, 1999). In section 5, a real railway link from TRA is selected to produce a speed profile with a proper operation time, followed by model verification and validation. Finally, concluding remarks are presented.

2. THE MATHEMATICAL MODEL

The movement of a train along a route is governed by the equation of motion and constrained by speed limits. The former is also called vehicle dynamic model, which is resulted from Newton's Second Law of Motion. The latter is to regulate the maximal speed of the train along the route for safety reasons. In addition, a particular operation objective may be set to guide the movement of the train. The following subsections present the mathematical models for generating speed profiles.

2.1 Equation of Motion

The movement of a train along a route is influenced by many forces, including tractive effort, train resistance, braking forces, and equivalent mass, as shown in Figure 1. Tractive effort provides the propulsion to overcome resistances and to accelerate the train. Train resistance is resulted from train characteristics and alignment geometries. Braking force is used to decelerate the train and bring it to full stop. During the movement of the train, the wheels, shafts, axles may store kinetic energy. As a result, The net force available for accelerating/decelerating the train includes not only its static mass, but also the rotating mass (Vuchic, 1982). Their sum is called equivalent mass of the train (Andrew, 1986).

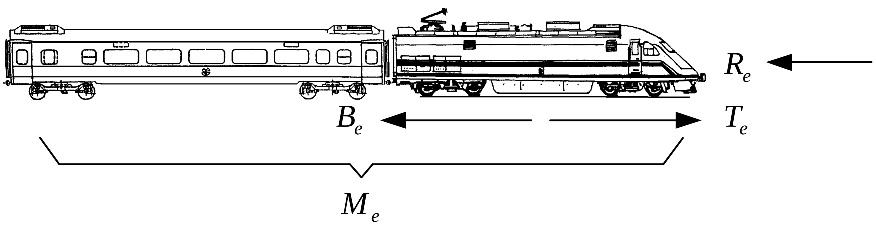

Figure 1. Force - Mass diagram for Train Movements

Let T_e, R_e, B_e, and M_e denote tractive effort, train resistance, braking force, and equivalent mass, respectively. Following the Newton's Second Law of Motion, their relation is derived below:

$$T_e(v) - R_e(v,i,r) - B_e(v) = M_e \frac{dv}{dt} \tag{1}$$

where

$R_e(v,i,\gamma) = R_s(v) + R_r(v) + R_g(i) + R_c(\gamma) + R_t$ = train resistance (N)

$T_e(v)$	= tractive effort	(N)	$R_s(v)$	= starting resistance	(N)
$R_r(v)$	= running resistance	(N)	$R_g(i)$	= grade resistance	(N)
$R_c(\gamma)$	= curvature resistance	(N)	R_t	= tunnel resistance	(N)
$B_e(v)$	= braking force	(N)	M_e	= train equivalent mass	(kg)
v	= train velocity	(m/s)	i	= grade	(‰)
γ	= curve radius	(m)	t	= time	(s)

In equation (1), the tractive effort and braking force can be controlled by train driver or ATC (automatic train control) computer. Therefore, there are three operating modes in driving a train, as listed in Table 1.

Table 1. Train Driving Modes

Train operating mode		Tractive effort condition $T_e = 0$	Tractive effort condition $T_e > 0$
Braking force condition	$B_e = 0$	Coasting mode	Powering mode
	$B_e > 0$	Braking mode	—

2.2 Speed Constraints

In addition to the equation of motion, train dynamics are also regulated by speed limits due to train formation, curvature, downgrade, 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:

$$V \le V_{\max} = \min\{V_t^-, V_r^-, V_g^-, V_o^-, V_w^-, V_s^-\} \tag{2}$$

where V =train speed (km/h)
$V_{\max}$ =maximal allowable speed (km/h)
V_t^- =speed limit due to train makeup/formation (km/h)
V_r^- =curvature speed limit (km/h)
V_g^- =downgrade speed limit (km/h)
V_o^- =switch/turnout speed limit (km/h)
V_w^- =speed limit due to track strength (km/h)
V_s^- = speed limit due to blocking signal (km/h)

Speed limit due to train makeup/formation must be obeyed at any time and any places, whereas those caused by curvature, downgrade and track strength may vary at different locations. As to switch speed limit and signal speed regulation, they are often omitted in single train simulation.

2.3 Planning Objective

As discussed in the foregoing sections, the operation of a train along a line must obey the equation of motion and the speed constraints. As a result, station – to – station movement may consist of four operation regimes, i.e., stop, acceleration, constant speed, and deceleration. The conditions of the net force acting on the train and its velocity for each regime are summarized in Table 2.

Table 2. Force and Velocity Conditions for Four Operation Regimes

Operation Regimes	Net force	Velocity
Stop	$T_e(v) - R_e(v,i,\gamma) - B_e(v) = 0$	$v = 0$
Acceleration	$T_e(v) - R_e(v,i,\gamma) - B_e(v) > 0$	$0 \le v \le v_{\max}$
Constant speed	$T_e(v) - R_e(v,i,\gamma) - B_e(v) = 0$	$v > 0$
Deceleration	$T_e(v) - R_e(v,i,\gamma) - B_e(v) < 0$	$0 \le v \le v_{\max}$

Note that there does not exist definite relations between train driving modes and operation regimes. For example, a train running on steep upgrade with powering mode may lead to deceleration due to grade resistance. Similarly, a train on steep downgrade with coasting mode may also result in acceleration. By alternating different driving modes and operation regimes, there are many ways to finish a train run and each has different travel time and energy consumption. Therefore, the speed profiles from start station to end station are not unique. In fact, the selection of speed profiles is determined by the planning goal. But each must follow the equation of motion and speed constraints. Two planning goals for generating speed profiles are introduced below:

(1) Shortest Operation Time

The speed profile generated under the goal of "shortest operation time" represents the minimal journey time among all speed profiles. The framework of the mathematical model is formulated as:

$$\text{Min} \int dt \tag{3}$$

subject to
$$T_e(v) - R_s(v) - R_r(v) - R_g(i) - R_c(\gamma) - R_t - B_e(v) = M_e \frac{dv}{dt} \tag{4}$$

$$V \le V_{\max} = \min\{V_t^-, V_r^-, V_g^-, V_o^-, V_w^-, V_s^-\} \tag{5}$$

For shortest time operation, the train must adopt full traction for powering acceleration, keep as close as to speed limit, and use full braking force for deceleration. Due to these reasons listed below, shortest operation time is only studied in theory, but cannot work in practice.

- It will induce overstrain of the train driver, and easily cause train delay.
- It wastes energy and is inefficient.
- It will wear the wheels, and deteriorate train performance.
- The frequent exchange of powering and braking mode operation results in the discomfort of passengers.

(2) Proper Operation Time

Due to the defects of "shortest operation time", railway operator tends to slightly increase journey time. To reserve operation margins for recovering train schedule when delay occurs, and to reduce energy consumption. In addition, it can alleviate the deteriorations of track and train. The planning goal is called "proper operation time". The framework of the model is formulated below:

$$\text{Min} \quad K_1 \int dt + K_2 \int E dt \tag{6}$$

subject to
$$T_e(v) - R_s(v) - R_r(v) - R_g(i) - R_c(\gamma) - R_t - B_e(v) = M_e \frac{dv}{dt} \tag{7}$$

$$V \le V_{\max} = \min\{V_t^-, V_r^-, V_g^-, V_o^-, V_w^-, V_s^-\} \tag{8}$$

where K_1 and K_2 are the weights of travel time and energy consumption
E =energy consumption rate (kW)

The weights between travel time and energy consumption depend on route, train and operation conditions. For busy line where train flow is closed to line capacity, the weight on trip time may be higher. On the contrary, if the travel demand is not too high and operating cost is a major concern, the weight on energy consumption may be higher. In Taiwan, TRA adopts the following rules to obtain the speed profile for proper time operation (Jong, 2001):

- Slightly decreasing speed limit to increase operation margins
- Ignoring speed limit increase in short section (usually less than 1 km)
- Sometimes using 85% of tractive effort for powering mode to account for voltage drop and performance deterioration

- Slightly increasing grade resistance coefficient when train starts to allow smooth acceleration
- Employing coasting acceleration in steep downgrade section when possible
- Using coasting deceleration from high speed section to low speed section when possible
- Applying level deceleration rate instead of full deceleration rate for braking mode in steep upgrade section
- If possible, using coasting deceleration before full braking in arriving a stop station
- Applying coasting operation for passing neutral section

3. SOLUTION ALGORITHM

The mathematical models introduced in the previous section are only prototypes. They are hardly solved by existing optimization software and analytical approach. In fact, special algorithms must be developed to solve the problem numerically. The proposed algorithms are explained below:

3.1 Train Representation

The mass of the train could be represented as single point, multiple points, or a line. Single point representation is computationally efficient, but less accurate. Multiple points and single line representation are complex, but more accurate. However, if the speed constraints could be carefully settled, single point representation is still an adequate approach in most applications (Howard et al., 1983).

The proposed algorithm employs single point representation of train in calculating vehicle dynamics, but uses a single line representation in determining speed limit to account for the train length. The relations among speed profiles, speed limits, and the length of train are depicted in Figure 2. The key point is that the entire train must obey the speed limits.

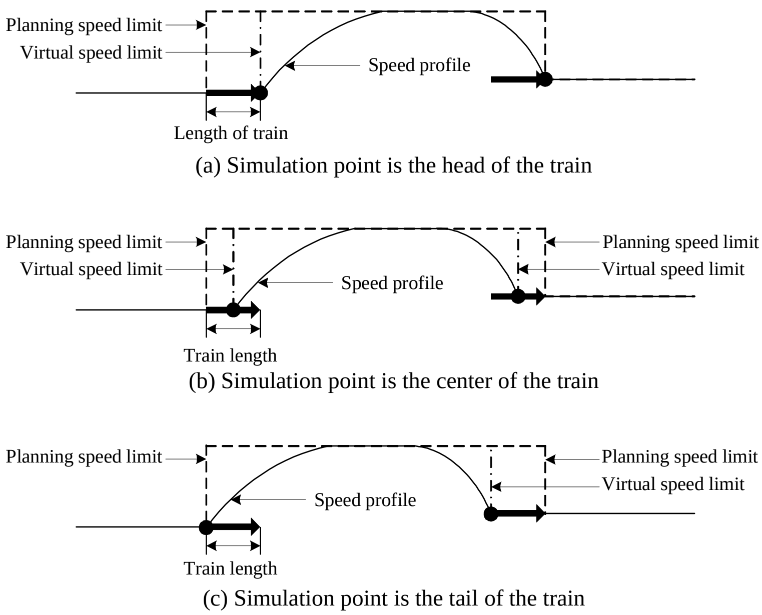

Figure 2. The Relation among Speed Limits and the Length of Train

3.2 Train Dynamic Calculations

The kernel of train movement simulation is the calculation of train dynamics. At the first glance, it seems that the train dynamic model in equation (1) can be solved by basic kinematic equations with constant acceleration/deceleration rate. However, due to the fact that tractive effort and train resistances vary with speed, the net force acting on the train is not fixed. As a result, the acceleration/deceleration rate of a moving train is not constant either. Alternatively, one may think that equation (1) could be solved by direct integration over time. Unfortunately, forces that influence train movement are not explicit functions of time. On the contrary, most of them are velocity dependent. Another approach is to integrate equation (1) over speed. But this only works in certain situations, where the tractive effort is an explicit function of speed or the train is in coasting operation mode (Jong, 2003). Consequently, most computer programs employ numerical methods to approximate the solutions. The algorithms all assume that the net force acting on the train is constant over a short section and then use iterative computational cycles based on time, distance, or velocity increments to calculate train dynamics (Howard et al., 1983). For example, Kikuchi (1991) uses distance increment method to simulate train travel on a rail transit line. Capillas (1987), Goodman (1987), and Uher (1987) adopt a time increment approach.

The proposed algorithm provides option for selecting time, space, or velocity as the independent variable in calculating train dynamics. The detailed computation formulas are discussed below:

(1) Time Increment

Let a (m/s^2), v (m/s), s (m), t (s) denote acceleration, velocity, space, and time, respectively. By definitions, we have

$$v = \int a dt = \lim_{\Delta t_i \to 0} \sum_i \bar{a}_i \Delta t_i \tag{9}$$

$$s = \int v dt = \lim_{\Delta t_i \to 0} \sum_i \bar{v}_i \Delta t_i \tag{10}$$

Assume that the net force at t_i is $F_i = F(v_i)$ (N). Then after a small time increment Δt_i, the dynamics of the train can be determined by the following equations:

$$\bar{a}_i \approx a_i = \frac{F_i}{M_e} \tag{11}$$

$$v_{i+1} = v_i + a_i (t_{i+1} - t_i) \tag{12}$$

$$s_{i+1} = s_i + \bar{v}_i (t_{i+1} - t_i) \tag{13}$$

(2) Distance Increment

If the independent variable for integration is s, then t and v can be expressed as:

$$t = \int \frac{ds}{v} = \lim_{\Delta s_i \to 0} \sum_i \frac{\Delta s_i}{\bar{v}_i} \tag{14}$$

$$v = \int \frac{ads}{v} = \lim_{\Delta s_i \to 0} \sum_i \frac{\bar{a}_i \Delta s_i}{\bar{v}_i} \tag{15}$$

If space increment Δs_i is small enough, then $\bar{a}_i$ can be assumed as a constant. Assume that the average acceleration during Δs_i can be computed by the force at s_i. Then the dynamics of the train can be described below:

$$\bar{a}_i \approx a_i = \frac{F_i}{M_e} \tag{16}$$

$$t_{i+1} = t_i + \frac{s_{i+1} - s_i}{\bar{v}_i} \tag{17}$$

$$v_{i+1} = v_i + \frac{a_i (s_{i+1} - s_i)}{\bar{v}_i} = v_i + \frac{2a_i (s_{i+1} - s_i)}{v_{i+1} + v_i} \Rightarrow v_{i+1} = \sqrt{v_i^2 + 2a_i (s_{i+1} - s_i)} \tag{18}$$

(3) Velocity Increment

If the integral variable is v, by physic and mathematic definitions we get

$$t = \int \frac{dv}{a} = \lim_{\Delta v_i \to 0} \sum_i \frac{\Delta v_i}{\bar{a}_i} \tag{19}$$

$$s = \int \frac{vdv}{a} = \lim_{\Delta v_i \to 0} \sum_i \frac{\bar{v}_i \Delta v_i}{\bar{a}_i} \tag{20}$$

Similarly, if Δv_i is small enough, then $\bar{a}_i$ can be considered as a constant. Let $\bar{F}_i$ be the average force during the velocity change in Δv_i. The dynamics of the train can be calculated from the following equations:

$$\bar{a}_i = \frac{\bar{F}_i}{M_e} \tag{21}$$

$$t_{i+1} = t_i + \frac{v_{i+1} - v_i}{\bar{a}_i} \tag{22}$$

$$s_{i+1} = s_i + \frac{\bar{v}_i (v_{i+1} - v_i)}{\bar{a}_i} = s_i + \bar{v}_i (t_{i+1} - t_i) \tag{23}$$

3.3 Determination of the Location and Initial Speed for Deceleration

It is very difficult to determine the location and initial speed for train braking or coasting in a deceleration regime. A simple approach is the one introduced in Uher (1987), which calculates both the forward and backward speed trajectories and then finds the intersection of the two trajectories to identify the location and initial speed for deceleration. The forward calculation starts from the origin of the train and moves toward the destination, while the backward calculation starts from the destination and moves the train toward the origin. Note that backward calculation can be carried out only when the train arrives at a low speed section or at a stop station.

Let the subscript f and b denote the forward and backward profiles, respectively. If the conditions of $v_{f,k} \le v_{b,k}$ and $v_{f,(k+1)} \ge v_{b,(k+1)}$ hold during backward calculation process, then both trajectories intersect each other, as shown in Figure 3. The intersection point can be calculated by the following linear system:

$$\begin{bmatrix} s_c \\ v_c \end{bmatrix} = \begin{bmatrix} m_f & -1 \\ m_b & -1 \end{bmatrix}^{-1} \begin{bmatrix} m_f s_k - v_{f,k} \\ m_b s_k - v_{b,k} \end{bmatrix} \tag{24}$$

where $m_f = \frac{v_{f,(k+1)} - v_{f,k}}{s_{k+1} - s_k}$ is the slope of forward line and $m_b = \frac{v_{b,(k+1)} - v_{b,k}}{s_{k+1} - s_k}$ is the slope of backward line.

The solution can be found if and only if the inverse matrix exists. Since $m_f \ge 0$ and $m_b < 0$, the coefficient matrix is nonsingular and thus, the solution exists.

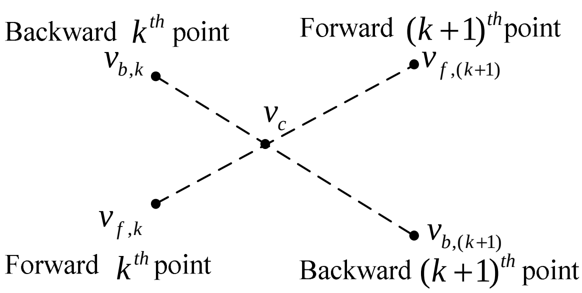

Figure 3. The Intersection of Forward and Backward Trajectories

3.4 Simulation Procedure

The overall procedure for generating train speed profiles is outlined in Figure 4. The track data in the proposed algorithm is described in a linked list format, similar to the one introduced in Giger (1987). Each node in the linked list represents a geometry change point in ascending order of mileage. When a train travels in reverse direction, the tract data must be reversed too. Thus, at the beginning of the simulation, the moving direction of the train must be determined. Next, a linked list template derived from the track data is prepared for the simulation. In the list, the speed limit for each section is determined according to train formation, curvature, and grade. Since alignment resistance (the sum of curvature resistance, grade resistance, and tunnel resistance) is independent of train speed, the alignment resistance is also calculated for each node in the list. At the next step, the computation procedure is divided into two parts. One is for shortest time operation, while the other is for proper time operation. As mentioned before, in proper time operation, the speed limit will be reduced to increase operation margins. Thus, the speed limit for proper time operation must be further adjusted. From now on, the computation procedures for both planning objectives are almost the same. First, the speed increase of the section whose length is less than train length plus a specified distance (for proper time operation) must be eliminated. Then a virtual speed limit based on the simulation point (head, middle, or tail of the train) is created. Finally, the computation is run into different sub-procedure for shortest time operation and proper time operation, as shown in Figure 5 and Figure 6, respectively.

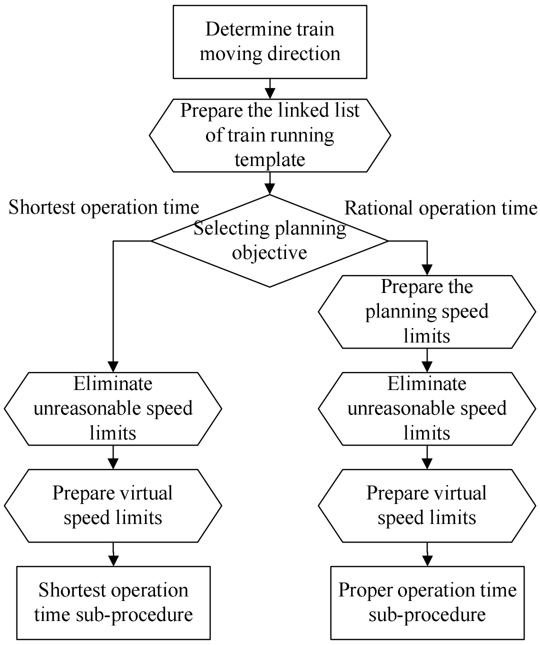

Figure 4. The Main Procedure for Generating Train Speed Profiles

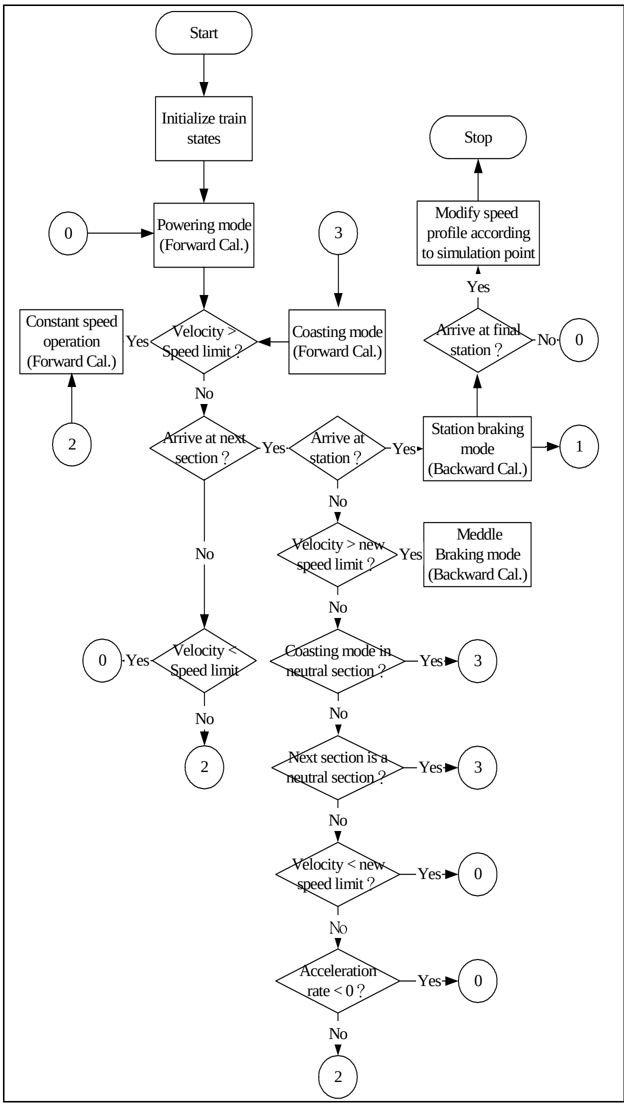

Figure 5. Sub-procedure for Shortest Time Operation

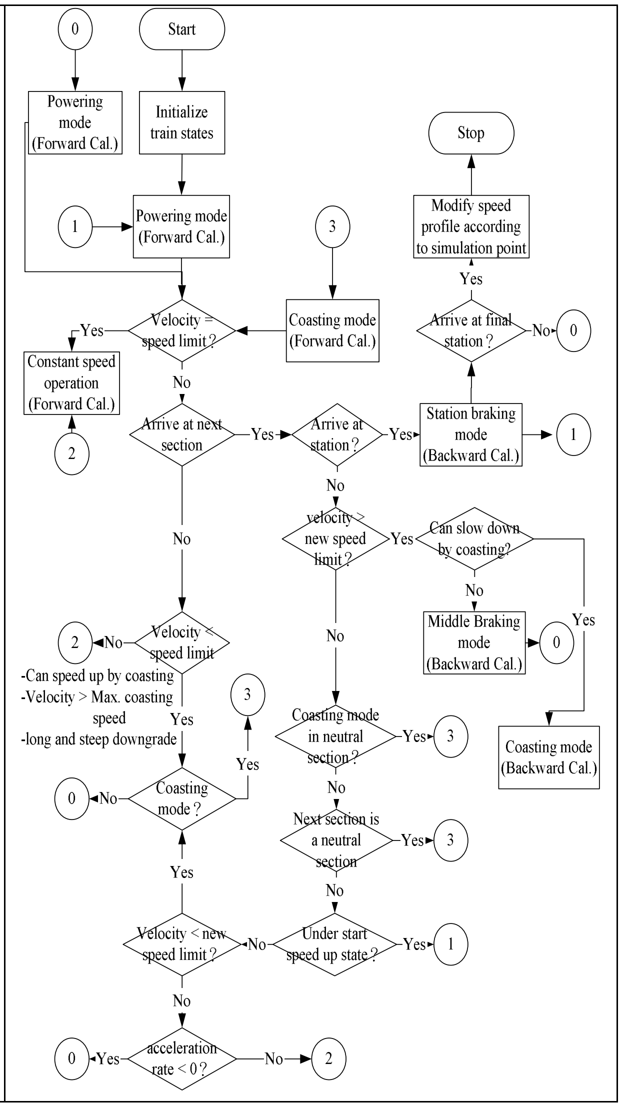

Figure 6. Sub-procedure for Proper Time Operation

There are many differences between the two sub-procedures for shortest time and proper time operations. First, the train adopts full tractive effort in powering mode for shortest time operation, but may use smaller traction for proper time operation. Second, in shortest time simulation the train does not take any coasting operation, while in proper time operation, the train seeks any possible chances to adopt coasting modes for potential energy savings.

4. SYSTEM ANALYSIS AND DESIGN

The proposed algorithm interacts between different components of a rail system, including power car, non-power car, train, railway, speed regulation rules, and stopping patterns of the train. For easy representation of the system, the Object-Oriented Programming (OOP) concept is employed to develop TrainSim. In system analysis and design phases, the Unified Modeling Language (UML) is used to construct use case and class diagrams. In system development phase, C++ language is selected for its computation efficiency.

TrainSim is a window-based GUI software, which is composed of three different modules, including "Database Editor", “Simulator”, and “Output Viewer”. Figure 7 depicts the use case diagrams of this program, which is useful to analyze the requirements of the software from user’s point of view.

(1) “Edit Database” use case: edit and modify the rolling stock database, rail line database, train database, and speed rules database.

(2) “Simulate Train Operations” use case: edit, modify, and execute train run curve simulation schemes.

(3) “View Simulation Results” use case: view the simulation result.

The input data for TrainSim are classified into six categories: power car, non-power car, train, rail line, curvature speed rule, and grade speed rule. “Edit Database” use case provides all required functions for database maintenance operations, such as create, edit, save, delete, rename, export, and import. “Simulate Train Operations” use case is the kernel of “TrainSim”. It is used to compile run case data from database and to calculate train dynamics or perform other additional functions. “View Simulation Results” use case demonstrates the train speed profiles produced by the “Simulate Train Operations” use case, and provides some diagram operations such as scroll, print, save, zoom, etc. The details of these use cases are displayed in Figure 8~10.

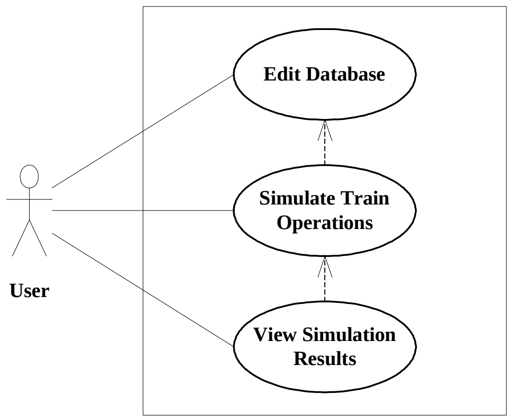

Figure 7. Use Case Diagram of “TraimSim”

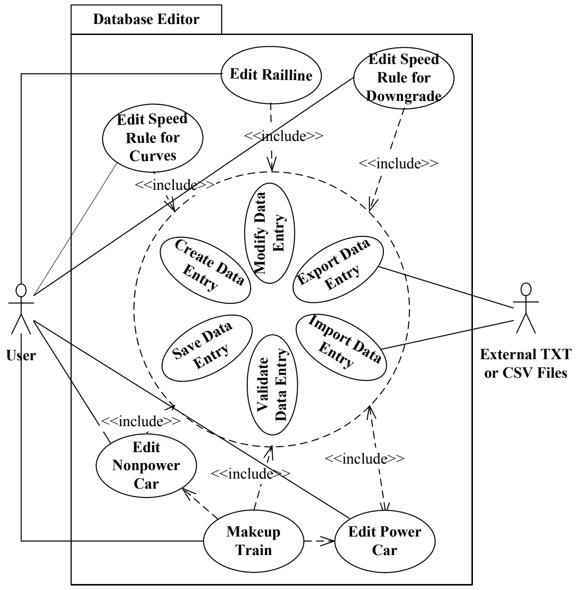

Figure 8. Use Case Diagram of “Edit Database”

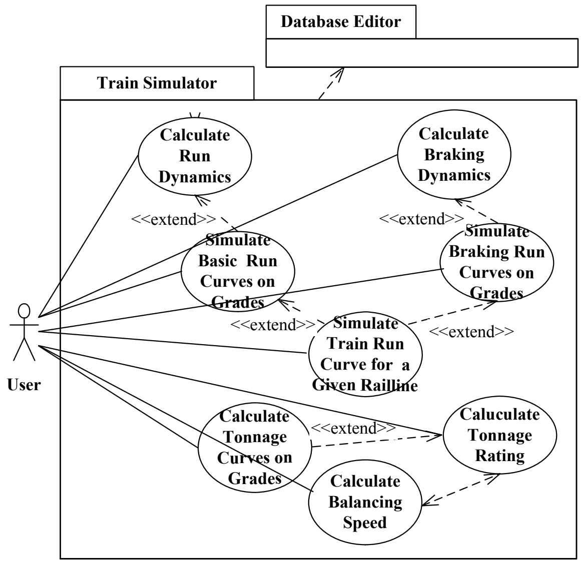

Figure 9. Use Case Diagram of “Simulate Train Operations”

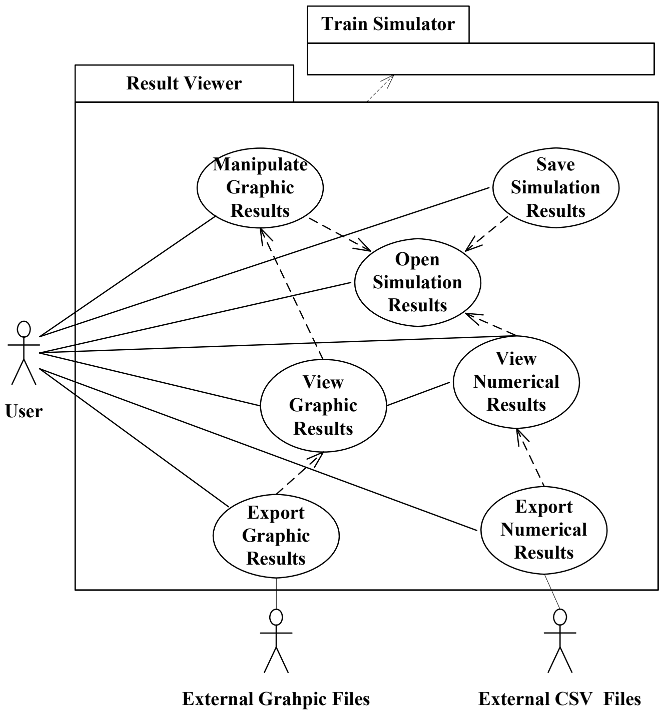

Figure 10. Use Case Diagram of “View Simulation Results”

5. CASE STUDY

To demonstrate the proposed model and algorithms for generating speed profiles, a real railway link (Tai-Chung line) from TRA is selected for case study. The Tai-Chung line is about 85.54 km long with maximal grade of 20‰ and minimal radius of 268 m. The test train is TRA E200 electric locomotive hauling 15 passenger cars (35SP) of 525 tons. The start resistance of E200 and 35SP are 5‰ and 3‰, respectively, while the running resistance of the train is listed in equation (25).

$$R_r = 8829.235 + 41.846V + 2.005V^2 \quad (\mathrm{N}) \tag{25}$$

The tractive effort and running resistance of the test train is illustrated in Figure 11. The dwell times are 1 minute at each station except Tai-Chung station whose dwell time is 2 minutes due to heavy passenger flow.

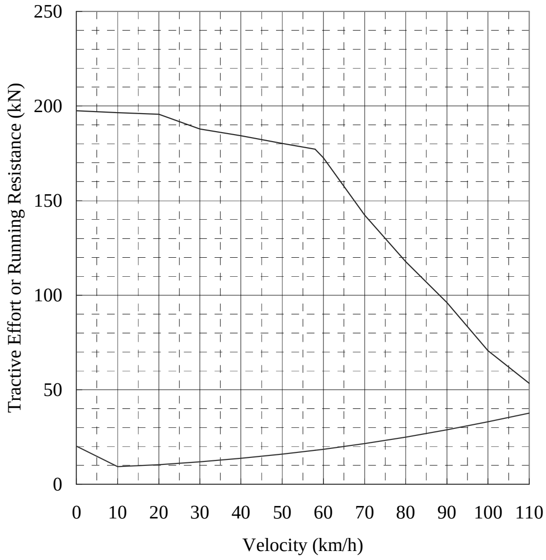

Figure 11. The Tractive Effort and Running Resistance of the Test Train

Figure 12 shows the speed profiles generated from TrainSim for both shortest time and proper time operations in downward direction of Tai-Chung line. The line segment on the top of the figure is the speed limit determined from train formation/makeup, curvature and grade speed limits. The speed profile for shortest time operation is drawn in a solid line, whereas that for proper time operation is sketched in a dot line. At the bottom of the figure are the vertical profile, horizontal curves, the stations along Tai-Chung line, and the mileage.

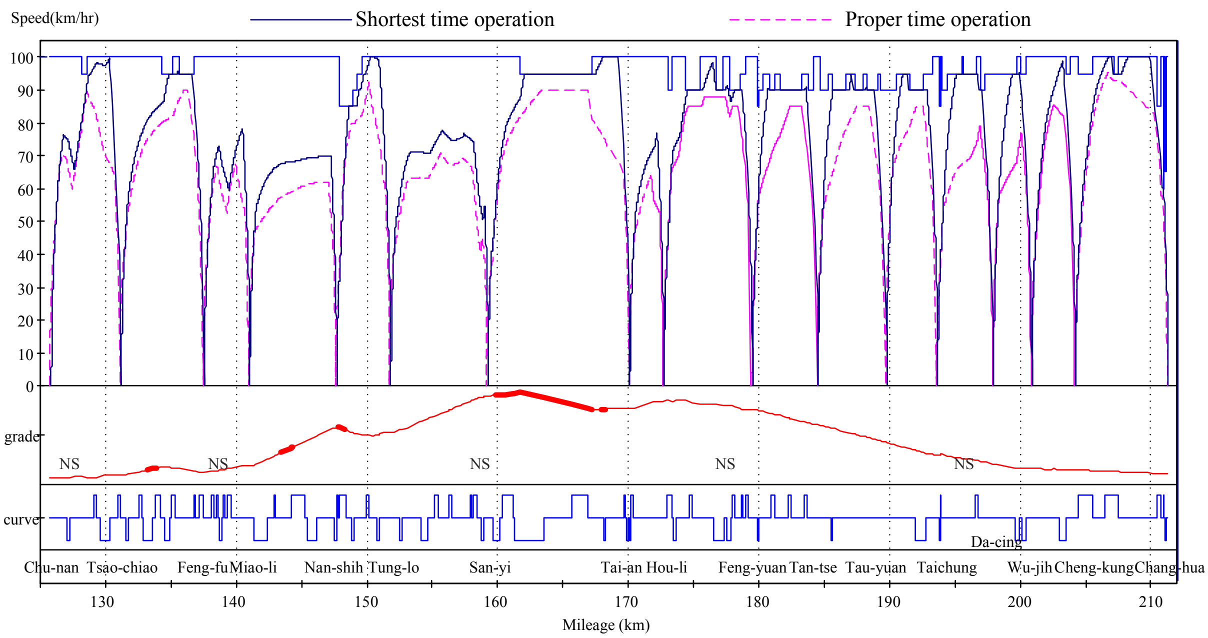

Figure 12. Train Speed Profiles for Downward Operation along TRA Tai-Chung Line

From Figure 12, we have the following observations:

- The area below shortest time profile is larger than the proper time profile. It indicates that the average speed for shortest time profile is higher and its operation time is shorter.
- Due to the decrease in tractive effort and the use of coasting mode, the speed profile of proper time operation can not reach speed limits.
- The speed profile of shortest time operation cannot reach speed limits in steep and short sections, including Chu-nan ~ Tsao-chiao, Feng-fu ~ Miao-li ~ Nan-shih, Tung-lo ~ San-yi, Tai-an ~ Hou-li, and Wu-jih ~ Cheng-kung.
- The ratio of proper operation time to shortest time for each section ranges from 107.72% ~ 120.91%. The smallest ratio (107.72%) happens in Hou-li ~ Feng-yuan section, while the largest one (120.91%) occurs in Taichung ~ Da-cing section.

6. MODEL VERIFICATION AND VALIDATION

In this section, the results from TrainSim are compared with a commercial software and manual calculation by a TRA expert to verify and validate the proposed model and algorithm. The rail link and the test train for the verification and validation are the same as those described in case study.

For shortest time operation, TrainSim is compared with the commercial software TOM (Uher, 1987). However, due to the limitations of the input format in TOM, the input data to TrainSim must be adjusted so that the results can be compared on the same basis. Since TOM dose not consider starting resistance, tunnel resistance, and the effect of neutral section, these parameters must be removed from TrainSim. The curvature resistance in TOM is $700/\gamma$ by default and cannot be adjusted, which is different from that ($600/\gamma$) used in TRA. Thus, TrainSim must use the same curvature resistance formula. Fortunately, TrainSim provides friendly interfaces for setting running resistance and curvature resistance. So the input data can be adjusted as close as possible for both models. The running times between adjacent stations are listed in Table 3. It is found that the difference between TrainSim and TOM ranges from -0.2% ~ 0.59%. The total running time results from TrainSim is 95.62 minutes, while TOM ends up with 95.47 minutes. The average difference is only 0.16%. This demonstrates the proposed algorithm is accurate.

Table 3. Running Time Comparison of TOM and TrainSim (Shortest Operation Time)

Section	Running time(min.)			Section	Running time(min.)		
	TOM (1)	TrainSim (2)	△(%)		TOM (1)	TrainSim (2)	△(%)
Chu-nan~Tsao-chiao	5.77	5.77	0.00	Hou-li~Feng-yuan	6.82	6.82	0.00
Tsao-chiao~Feng-fu	6.66	6.68	0.30	Feng-yuan~Tan-tse	5.61	5.64	0.53
Feng-fu~Miao-li	4.77	4.77	0.00	Tan-tse~Tau-yuan	5.78	5.81	0.52
Miao-li~Nan-shih	7.90	7.92	0.25	Tau-yuan~Taichung	5.66	5.67	0.18
Nan-shih~Tung-lo	5.10	5.09	-0.20	Taichung~ Da-cing	5.11	5.14	0.59
Tung-lo~San-yi	8.18	8.20	0.24	Da-cing ~Wu-jih	4.05	4.06	0.25
San-yi~Tai-an	9.41	9.41	0.00	Wu-jih~Cheng-kung	4.51	4.52	0.22
Tai-an~Hou-li	4.30	4.30	0.00	Cheng-kung~Chang-hua	5.83	5.82	-0.17
				Total	95.47	95.62	0.16

Note: $\Delta = [(2)-(1)]/(1)\times 100\%$

Since most commercial software does not provide option for proper time operation, the result of TrainSim is compared with manual calculation by the TRA expert. Table 4 shows that the difference of proper operation times computed from TrainSim and the TRA expert ranges from -7.11% ~ 5.53%. The average difference is -0.12%. For the sections with higher variations, we have further check the speed profile and discussed with the TRA expert. The difference is due to the reason that the combination of driving modes in TrainSim and the one used in manual calculation are not completely the same. However, the TRA expert thinks that a train can be driven in difference ways. Both reserve operation margins and are applicable to real operation.

Table 4. Running Time Comparison of TRA Expert and TrainSim (Proper Operation Time)

Section	Running time(min.)			Section	Running time(min.)		
	TRA expert (1)	TrainSim (2)	△(%)		TRA expert (1)	TrainSim (2)	△(%)
Chu-nan~Tsao-chiao	5.39	5.10	5.53	Tai-an~Hou-li	3.49	3.40	2.61
Tsao-chiao~Feng-fu	5.91	5.97	-1.01	Hou-li~Feng-yuan	6.10	6.55	-7.11
Feng-fu~Miao-li	4.25	4.10	3.59	Feng-yuan~Tan-tse	5.17	5.17	0.00
Miao-li~Nan-shih	7.09	7.28	-2.64	Tan-tse~ Taichung	7.84	8.12	-3.51
Nan-shih~Tung-lo	4.40	4.23	3.94	Taichung~Wu-jih	6.74	6.42	4.86
Tung-lo~San-yi	7.49	7.43	0.80	Wu-jih~Cheng-kung	3.70	3.70	0.00
San-yi~Tai-an	9.18	9.05	1.43	Cheng-kung~Chang-hua	6.24	6.26	-0.32
				Total	83.00	83.10	-0.12

Note: $\Delta = [(1)-(2)]/[[(1)+(2)]/2] \times 100\%$

7. CONCLUDING REMARKS

This paper introduces two mathematical models and algorithms that are used in TrainSim for generating train speed profiles. The proposed models consider starting resistance, tunnel resistance and the effect of neutral section that are often ignored in vehicle dynamic models. Three different approaches for calculating train dynamics are also presented in the paper. The algorithms presented in this research do not need explicit input of speed limit at each rail section. Rather, the speed constraints are automatically determined in the simulation. The speed profiles generated from TrainSim are compared with a commercial software and hand calculation by the TRA expert. The results demonstrate that the proposed algorithms are robust and efficient, and are very useful and flexible for train operation research and applications.

REFERENCES

American Railway Engineering and Maintenance-of-Way Association (AREMA) (1999) **Manual for Railway Engineering, Vol. 4. System Managemen**t.

Andrews, H. I. (1986) **Railway Traction: The Principles of Mechanical and Electrical Railway Traction.** Elsevier, New York.

Capillas, E. H. and Vadillo, J. V. (1987) Computer Simulation of the Basic Parameters for Designing an Underground Railway Line In T. K. S. Murthy et al. (eds.), **Computers in Railway Management.** Computational Mechanics Publications, Southampton, U. K.
Chen, S. Y., Ku, B. Y. (2000) A Structure of Computer Aided Train Speed Planners and Performance Simulator for the New Century. **Proceedings of International Conference on Rail Transportation**, Taiwan, 229-239.

Giger, P. (1987) A Data Concept for Simulation of Railway Networks. In T. K. S. Murthy et al. (eds.), **Computer in Railway Management.** Computational Mechanics Publications, Southampton, U.K.

Goodman, C. J., Mellitt, B. and Rambukwella, N. B. (1987) CAE for the Electrical Design of Urban Rail Transit Systems, In T. K. S. Murthy et al. (eds.), **Computers in Railway Operations**. Computational Mechanics Publications, Southampton, U. K.

Hay, W. W. (1982) **Railroad Engineering, 2nd edition**. John Wiley & Sons, Inc., Canada.

Howard, S. M., Gill, L. C., and Wong, P. J. (1983) Review and Assessment of Train Performance Simulation Models, **Transportation Research Record 917**, 1-6.

Jong, J. C. (2001) **A Status Review of Planning and Operational Models for Rail Transportation System**. (in Chinese) Sinotech Engineering Consultants, Inc., Taiwan.

Jong, J. C. (2003) Analytical Solutions for Predicting Train Coasting Dynamics. **Proceedings of the Eastern Asia Society for Transportation Studies (EAST)**, Japan, Fukuoka.

Jong, J. C., Chang, S., and Chang, E. (2003) Train Run Curve Simulator. **The 6th Asia-Pacific ITS Forum**, Taiwan, Taipei.

Kikuchi, S. (1991) A Simulation Model of Train Travel on a Rail Transit Line, **Journal of Advanced Transportation, Vol. 25, No. 2**, 211-224.

Rumbaugh, J., Jacobson, I., and Booch, G. (1999) **The Unified Modeling Language Reference Manual**. Addison-Wesley Longman, Inc., Massachusetts.

Uher, R. A. and Disk, D. R. (1987) A Train Operations Computer Model, In T. K. S. Murthy et al. (eds.), **Computers in Railway Operations**. Computational Mechanics Publications, Southampton, U. K.

Vuchic, V. R. (1981) **Urban Public Transportation — Systems and Technology.** Prentice-Hall, Inc, Englewood Cliffs, New Jersey.