Proceedings of the 2011 Winter Simulation Conference
S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu, eds.

CALIBRATION OF URBAN RAIL SIMULATION MODELS: A METHODOLOGY USING SPSA ALGORITHM

Zhigao Wang

China Sustainable Transportation Center
CITIC Building, Room 1903
No. 19 Jianguomenwai Dajie
Beijing 100004, CHINA

Haris N. Koutsopoulos

KTH-Royal Institute of Technology
Stockholm, SWEDEN

ABSTRACT

Rail simulation model calibration is a process of adjusting model parameters while comparing model output with observations from the real rail system. There is a lack of systematic methodology for calibrating urban rail simulation models. Based on a simulator developed for urban rail operations and control, the paper demonstrates a methodology of calibrating model parameters, and specifically, fine-tuning some of the simulation inputs. The calibration process is modeled as a multi-variate optimization problem and solved by the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm. A case study of the Massachusetts Bay Transportation Authority (MBTA) Red Line shows that the methodology improves the simulation model dramatically in terms of replicating the track block runtimes. At the same time, it upgrades the station specific dwell time parameters and enhances a-priori boarding rates at stations fairly effectively.

1 INTRODUCTION

An urban rail operations and control simulator consists of several individual models, which require a number of inputs as well as parameters (Koutsopoulos and Wang 2007). Models can be deterministic or stochastic. For example, passenger arrival is generally modeled as a stochastic process. Some models have analytical formulations, such as dwell time models, while others do not, like signal control logic, which is modeled as a discrete control depending on the occupancy of related track blocks. The simulation model works as an integration of these interactive models, with their inputs and parameters, as shown in Figure 1 (Wang 2006).

Parameters and inputs that are included by the simulation model fall into either the “gray” category (information is partially known) or “white” category (information is known and reliable). In order to make the simulator replicate the real behavior of the system, a number of model parameters and inputs which belong to the “gray” category require calibration. For example, the dwell time model for the simulator could be developed based on small dwell time samples from a limited number of stations. Parameter values from such models provide valuable information; however, generalizing them as being applicable to all stations is not realistic. Station specific parameter values can then be determined through the calibration process. Another case is that the observations from the real system are collected under configurations (time, space or other specific conditions) that are not readily replicable by simulation. For example, for many urban rail systems, there are abundant observations of OCS (Operations Control System) runtimes generated every day, which are good measurements of the current system behavior. However, to get their equivalents from simulation for the purpose of comparing the simulated and observed behavior, the demand when the observations were collected must be available as simulation input. This is under many circumstances impossible, or expensive to obtain. Using the known information to infer the demand through the calibration process becomes an approach worth exploring, and in some cases, one of the most cost-effective options.

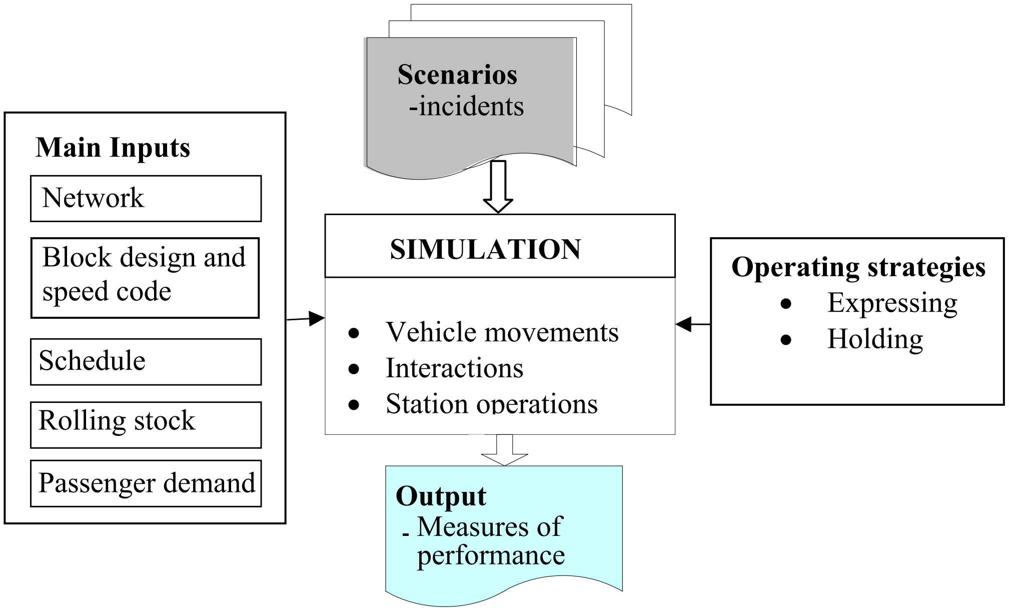

Figure 1: Components of a rail simulation model

Calibrating the simulation model generally involves comparing the simulated system behavior with the equivalents of actual system observations. In order to do this, a set of measurements has to be defined based on the available observations from the real system and simulation output. Data that can be used for this purpose include manually collected data and/or data recorded by OCS, which is available in many rail transit systems.

2 MODEL FORMULATION

The output Y^S of a simulation model *S* is a function of true parameters *P* and input data *I*:

$$Y^S = S(P, I) \tag{1}$$

Y^S is an estimate of the true system measurement *Y*. The goal of the calibration is to make Y^S as close to *Y* as possible by adjusting parameters *P*, and in some cases, input *I*.

The calibration of simulation model *S* is formulated as an optimization problem. The objective is to minimize the difference between *Y* and Y^S while letting simulation parameters and input deviate from a-priori values. The general form of the model can be formulated as

$$\begin{aligned}
&MinZ = \sum_{i=1}^{m} w_1 (Y - Y^S)^2 + \sum_{j=1}^{n} w_2 (P^a - P)^2 + \sum_{k=1}^{l} w_3 (I^a - I)^2 \\
&st. \\
&\qquad l_P \le P \le u_P \\
&\qquad l_I \le I \le u_I \\
&\qquad Y^S = S(P, I)
\end{aligned} \tag{2}$$

where S is simulation model; Y and Y^S are measurements collected from real system and measurements collected from simulation; P^a and P are a-priori values for parameter and their estimated values during calibration; I^a and I are simulation initial input (a-priori values) and estimated input during calibration; l_* and u_* are lower and upper bounds allowed for related variable category *; w_i are weights; and *m, n, l* are numbers of items included.

3 ALGORITHM

The above model is essentially a multivariate stochastic optimization problem. The general representation of this type of model is the minimization of the objective function (usually called a loss function) $Z(\theta)$ with respect to a set of parameters θ. The general approach to solving such problems is based on an iterative process of searching for optimal parameter values. The search path is guided by the gradient measurement associated with the loss function.

The model is difficult to solve since it is simulation-based and hence, lacking a closed form solution. Thus direct gradient measurements are unavailable. However, there are algorithms that can solve such problems based on approximations to the gradient formed from measurements of the loss function. The general procedure of seeking the optimal solution θ^* is based on the following updating function

$$\theta_{k+1} = \theta_k - a_k g_k(\theta_k) \tag{3}$$

where $g_k(\theta_k)$ is the estimate of the gradient $g(\theta) \equiv \nabla Z(\theta)$ at iteration *k*, and a_k is the step size at iteration *k*.

There are a number of Stochastic Approximation (SA) algorithms that can solve problems where only the approximation of the gradients is available. Examples of such algorithms include Finite-Difference SA (FDSA) and Simultaneous Perturbation SA (SPSA) algorithm.

FDSA requires that each parameter θ_k be perturbed one at a time and corresponding measurements being evaluated. The gradient estimate is calculated by differencing the measurements, then dividing by the difference interval, used to form the perturbation. For example, for a two-sided differential FDSA, for each parameter θ_i ($i = 1, 2 \ldots p$), the gradient estimate is calculated by

$$g_{ki}(\theta_k) = \frac{Z(\theta_k + c_k e_i) - Z(\theta_k - c_k e_i)}{2c_k} \tag{4}$$

where e_i denotes a vector with one in i^{th} place and zero elsewhere, and c_k is a small positive number that becomes smaller as *k*, the number of iterations, increases. The calculation of the gradient vector requires $2p$ function evaluations.

Compared to FDSA, SPSA approximates the gradient $g_k(\theta_k)$ by randomly perturbing all the elements of θ_k simultaneously to get two measurements of *Z*, while each individual gradient $g_k(\theta_k)$ is formed from a ratio involving that element and the difference between the two corresponding measurements. For two-sided simultaneous perturbation, the gradient is approximated by

$$g_{ki}(\theta_k) = \frac{Z(\theta_k + c_k \Delta_k) - Z(\theta_k - c_k \Delta_k)}{2c_k \Delta_{ki}} \tag{5}$$

where $\Delta_k = (\Delta_{k1}, \Delta_{k2}, \cdots, \Delta_{kp})$ is the random perturbation vector of dimension *p*, following a user specified distribution.

Spall (1998) reports that: "Under reasonably general conditions, SPSA and FDSA achieve the same level of statistical accuracy for a given number of iterations, even though SPSA uses *p* times fewer function evaluations than FDSA (because each gradient approximation uses only 1/*p* the number of evaluations)." The mathematical rationale and experimental results can be seen in Spall (1998). As a result, the SPSA makes the calibration much more efficient than FDSA since it requires far less number of simulation evaluations. The SPSA proceeds according to the following steps:

1. Initialization. Assign initial values for SPSA parameters *a*, *c*, A, α and γ in order to calculate the gain sequences $a_k = (A + k)^{\alpha}$ and $c_k = c / k^{\gamma}$. Choose a-priori values of simulation parameters and input that are also included as subject of calibration, set number of iterations *k* = 0; set maximum number of iterations desired to perform $K_{\max}$;
2. Evaluation of loss function.
3. Generation of the perturbation vector. A Bernoulli ±1 distribution with probability of 0.5 is used to generate a seed matrix for perturbation. To accommodate the different magnitudes of θ_{ki}, the actual perturbation Δ_{ki} is calculated as

$$\Delta_{ki} = \delta_{ki} \bullet \rho\theta_{ki} \tag{6}$$

   where δ_{ki} is the Bernoulli random number and is a constant (0 ≤ ρ ≤ 1);
4. Gradient approximation. The simulation is run with parameters $\theta_k \pm c_k \Delta_k$, the gradient is calculated by

$$g_k(\theta_k) = \frac{Z(\theta_k + c_k\Delta_k) - Z(\theta_k - c_k\Delta_k)}{2c_k} \begin{bmatrix} \Delta_{k1}^{-1} \\ \Delta_{k2}^{-1} \\ \vdots \\ \Delta_{kp}^{-1} \end{bmatrix} \tag{7}$$

5. Updating *θ*. $\theta_{k+1} = \theta_k - a_k g_k(\theta_k)$
6. Convergence. Two convergence criteria are used to terminate the optimization: $Z(\theta_{k+l}) - Z(\theta_k) \le \varepsilon$ or $k = K_{\max}$, where *l* is the number of iterations after iteration *k* during which the loss function changes less than a very small number *ε*. If neither of the 2 criteria is satisfied, go to 3, otherwise stop.

4 CASE STUDY

4.1 Line Characteristics

The above methodology is applied in a simulation study on the Massachusetts Bay Transportation Authority (MBTA)'s Red Line, which is simulated with a generic simulator - SimMETRO (Koutsopoulos and Wang 2007).

The Red Line consists of a trunk part and 2 branches. The trunk part (Alewife to JFK) is 8.4 miles long and has 12 stations. After JFK, the Ashmont line is 3.4 miles long and Braintree line is 9.1miles long.

The network under study consists of the one-way section of the Red Line from Alewife to South Station. This section has 10 stations and runs two routes destined to Ashmont and Braintree, respectively (Figure 2).

The simulator is calibrated to 4-6pm period. During which, train dispatching headways at Alewife are scheduled at 4 minutes, except for 4 pairs of trains having 3 minutes headways that start at 4:40pm, 4:50pm, 5:00pm and 5:10pm respectively. In total, 15 Ashmont and 17 Braintree trains are scheduled;

with 2 pairs of consecutive Braintree trains scheduled starting at 4:43pm and 4:53pm. However, the actual dispatching headways during the afternoon peak have very high variability, ranging from less than 2 minutes to more than 10 minutes, as shown in Figure 3. Given this, the dispatching headways are represented by an empirical distribution in the model calibration.

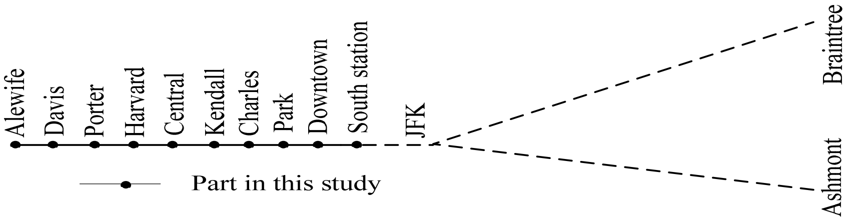

Figure 2: The MBTA Red Line Network

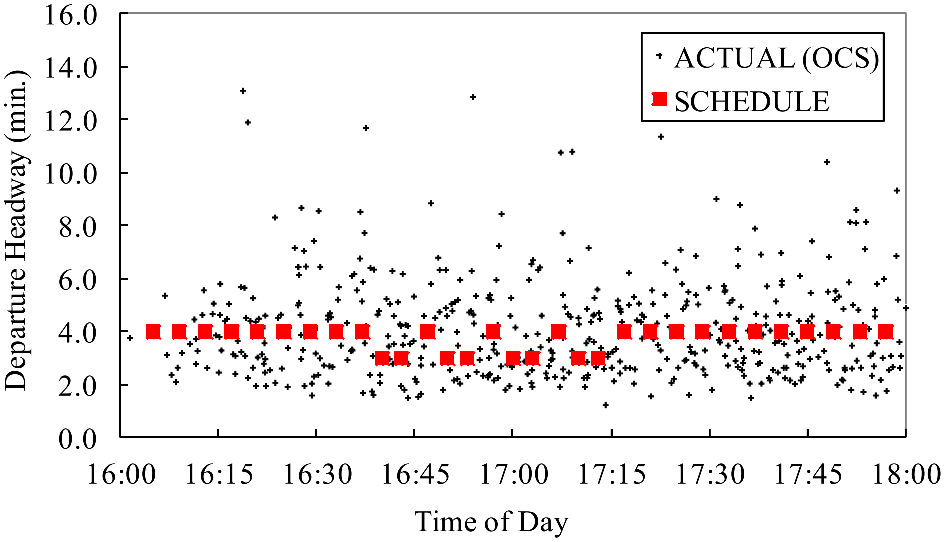

Figure 3: Scheduled vs. actual dispatching headway at Alewife

The Red Line has many sources of randomness both from the demand and supply sides, such as passenger arrival, boarding and alighting behavior, vehicle variability, track conditions and OCS block time uncertainty, level of power supply variability, vehicle-track adhesion variability, operators' behavior, incidents, etc. These uncertainties are represented in the simulation model with respective level of stochasticity (Wang 2006).

4.2 Parameters and Input for Calibration

4.2.1 Dwell Time Model Parameters

The currently implemented dwell time model is Puong's model(Puong 2000). It is developed based on small dwell time samples collected at Kendall and South Station. It is understandable that different stations may have different platform configurations and dwelling environments, thus the model parameters may not be universally applicable. Allowing model parameters to have some station specific freedom in calibration is necessary if different dwell models for different stations are not considered.

The model captures the dwell time by the following parameters: the constant C_0, boarding time per passenger α, alighting time per passenger β, and crowding factor γ.

The suggested values for dwell model parameters are given later in Table 1. These values are used as initial values and will be tuned during the calibration.

4.2.2 Demand

The 1997 CTPS (Central Transportation Planning Staff) count were the most reliable passenger demand data at the study.

While the demand data is old; it contains valuable information, such as the split between routes, the relative magnitude of station demand, the fraction of alightings to the train load, etc. So, 1997 demand are good a-priori values to start with during the calibration process.

4.3 Measurement

OCS track occupancy times record the moments when trains enter and leave the block. Based on these moments, the duration for the train to cross the entry and exit points is called block runtimes. More generally, these times can be used to calculate the overall trip times for trains. The simulator mimics the entering and exiting events and generates data that are comparable to OCS data.

Since block runtime measures train movement on the block level, it is the most fundamental and detailed measure to capturing system behavior. In addition, block runtimes for station blocks are good estimates of dwell times which reflect demand levels; they are easy to extract from OCS records and simulation output; and large enough samples are easy to obtain, thus facilitating the comparison between simulation and real system measurements.

The OCS observations also show periodical differences in terms of block run times during 4-6pm period, especially for blocks where station platforms are located. This is mainly due to the time-dependent characteristics of demand level and related train dispatching pattern. So, periodical block runtimes are used to capture the pattern. This is also consistent with the time-specific demand level and multi-period dispatching used in the calibration, as discussed before.

4.4 Formulation for Red Line

With the above selected input data and system measurements for calibration, the model is formulated for the Red Line as follows.

$$
\begin{aligned}
MinZ = {} & w_t \sum_{i=0}^{p}\sum_{j=0}^{m}\left(t_{ij}^{sim} - t_{ij}^{OCS}\right)^2 + w_a \sum_{i=0}^{p}\sum_{k=1}^{n}\left(a_{ik}^{sim} - a_{ik}^{CTPS}\right)^2 + w_f \sum_{i=0}^{p}\sum_{k=1}^{n}\left(f_{ik}^{sim} - f_{ik}^{CTPS}\right)^2 \\
& + w_c \sum_{k=0}^{n}\left(c_k^{sim} - c_k^{mit}\right)^2 + w_\alpha \sum_{k=0}^{n}\left(\alpha_k^{sim} - \alpha_k^{mit}\right)^2 + w_\beta \sum_{k=0}^{n}\left(\beta_k^{sim} - \beta_k^{mit}\right)^2 \\
st. & \qquad\qquad (8) \\
& l_k^a \le \frac{a_{ik}^{sim}}{a_{ik}^{CTPS}} \le u_k^a ; l_k^f \le \frac{f_{ik}^{sim}}{f_{ik}^{CTPS}} \le u_k^f ; l_k^c \le \frac{c_k^{sim}}{c_k^{mit}} \le u_k^c ; \\
& l_k^\alpha \le \frac{\alpha_k^{sim}}{\alpha_k^{mit}} \le u_k^\alpha ; l_k^\beta \le \frac{\beta_k^{sim}}{\beta_k^{mit}} \le u_k^\beta ; t = S(a, f, c, \alpha, \beta, \cdots)
\end{aligned}
$$

Where, t_{ij}^{sim} and t_{ij}^{OCS} are simulated and OCS observed average block runtime within period i for block j; a_{ik}^{sim} and a_{ik}^{CTPS} are simulation estimated and CTPS counted passenger arrival rate during period i at station

k; f_{ik}^{sim} and f_{ik}^{CTPS} are simulation estimated and CTPS counted passenger alighting fraction during period i at station k; c_k^{sim}, α_k^{sim}, and β_k^{sim} are simulation estimated dwell time model constant, boarding time per passenger, and alighting time per passenger at station k; c_k^{mit}, α_k^{mit} and β_k^{mit} are the dwell time model suggested constant, boarding time per passenger, and alighting time per passenger at station k; l_k^* and u_k^* are lower bound and upper bound for parameter * at station k; w_* is the weight for item *; and $S(*)$ is the simulation.

4.5 Model solution

4.5.1 Selection of Parameter Values

A number of optimization model parameters need carefully selected values. These parameters include the SPSA algorithm parameters, the weights in objective function, and the lower and upper bounds for parameters to be calibrated. Finding the proper values for these parameters often needs quite a few test runs. Without elaborating on these parameters, either the optimization will not be able to reach the optimal, or the speed of convergence will be slow.

Seeking the appropriate values for these model parameters is actually a process to calibrate the optimization model itself, which is used to calibrate the simulator. There are efforts reported in the literature that provide general guidelines on how to choose SPSA model parameters, such as Spall (1998).

The weights for objective functions are also very important for the performance of the model. Weights for the parameters not only reflect the relative confidence that each to-be-calibrated parameter deserves to be within the same category of parameters, but also play the role on trading off the contributions to objective functions among parameters of different categories. These weights drive calibrated parameters to move towards desired direction.

Since there has not been a systematic approach to know what are the best values for these parameters and weights, trial and error were used to find relatively good set of values in terms of improving model performance. The suggested values by literature are used as starting points (Spall 1998, Wang 2006).

The bounds provide a solution space for the problem. They set constraints to parameters in terms of how far they can deviate from the a-priori values. They confine the optimization from reaching out to spaces that are not possible to contain the optimal solution based on the given information, thus reduce the computational effort. The actual scope of the bounds in many cases relies on judgment and confidence on the known information, and the performance of test runs. For example, if the test runs show that the bound for some parameter is constantly reached for many iterations, it means the bound might be revalued to allow larger freedom.

4.5.2 Objective Values

The model was solved by SPSA algorithm. Figure 4 illustrates the change of the loss function with respect to iterations. The objective values reduce quickly in the first few iterations then start bouncing with a general tendency of decreasing. As the number of iterations increases, the gain sequences of SPSA algorithm become smaller thus the adjustments to the variables per step are smaller, and the oscillation of the objective function is less intensive, until it settles down at a certain level without changing too much for some consecutive iterations. The solution near this realm is taken as the optimal solution.

4.5.3 Dwell Model Parameters

Table 1 shows the calibrated dwell time model parameters. It can be seen that the constants for all the stations are larger than the initial values. This explains the possibility that Puong's model could have missed some source of dead time. The dead time captures all the time that is non-boarding and alighting related, including the time when trains stay still with door closed after boarding (which happens in a few cases). Note that Puong's model used dwell time samples measured from the door open to door close time, which does not count the standing time after doors close. As expected, the dead time at Park station almost doubled, since the operator has to open and close the door on both sides of the train, and move back and forth twice, and this takes some extra time. It indicates that the calibration picks-up these details and improves the model.

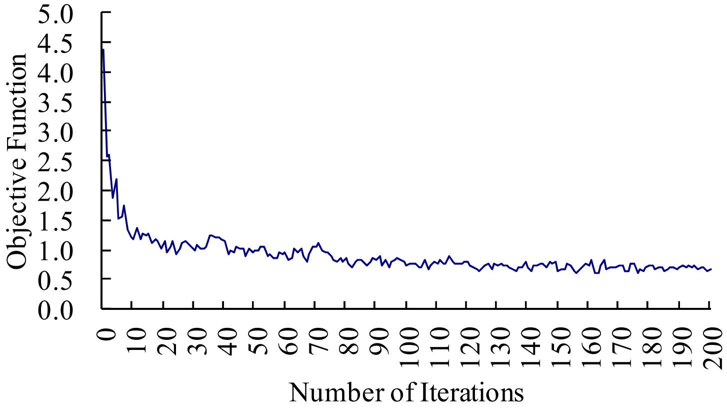

Figure 4: Convergence of objective function

Table 1: Calibrated values for dwell model parameters

Parameter	C_0	α	β
Puong's model	12.52	2.27	1.82
Davis	15.23	2.4	1.89
Porter	13.61	2.09	1.84
Harvard	18.12	2.04	1.83
Central	16.71	2.29	1.77
Kendal	13.91	2.34	1.83
Charles	20.04	2.41	1.88
Park	23.83	2.52	1.89
Downtown	14.14	2.37	1.87
South Station	17.4	2.34	1.83

4.5.4 Passenger Demand

In order to evaluate the effectiveness of the calibration, the estimated passenger arrival rates obtained from the calibration were compared to the actual arrival rates in 2004 at select stations for which such data was available. Note that only the 1997 demand data rates were used for calibration (as a-priori values). The 2004 data was used exclusively for validation purposes after calibration.

Figure 5 compares the calibrated arrival rates at the 4 stations for which data from 2004 is available. For each station arrival rates reported correspond to 30 min intervals for the 4–6 pm peak period. The graph illustrates the a-priori values (1997), the actual values (2004), and the calibrated values for the period from 4–6 pm, in 30 min intervals. The results indicate that the calibration estimates the actual arrival rates fairly well, especially for high demand stations.

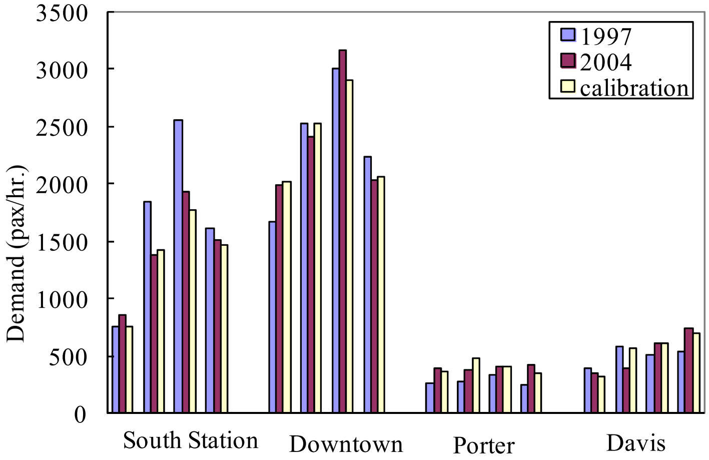

Figure 5: Demand calibration

4.5.5 Measurements

In order to compare the effectiveness of calibration, before and after calibration system behavior measurement comparisons are done. Aimed at capturing the train movement and train-passenger interaction, train movement at block level, or more specifically, the run time distributions for all the blocks were compared. Most of the blocks gain significant improvement after the calibration, especially those that are sensitive to congestion and demand levels, like the station blocks where platforms are located. These blocks were divided into 4 half-hour period and compared directly between simulation and OCS values during the calibration, the final solution shows the averages for each period have improved significantly, as shown by Table 2, the differences between simulation and OCS were reduced substantially after calibration (the values in brackets are after-calibration differences).

Table 2: Station block runtime difference: OCS vs. simulation (sec.)

Location	4:00-4:30	4:30-5:00	5:00-5:30	5:30-6:00
Davis	-8.1(-2.7)	-4.3(0.0)	-7.8(-2.0)	-7.5(-2.8)
Porter	-2.8(-0.2)	+0.8(-0.3)	-2.2(-0.5)	-5.9(-1.4)
Harvard	-9.7(+0.8)	-5.5(-0.6)	-3.2(-0.7)	-4.4(-1.6)
Central	-3.2(+1.3)	-5.2(+0.3)	-6.7(+3.0)	-5.0(0.0)
Kendall	-7.6(-0.4)	-4.8(-1.4)	-7.8(+1.6)	-1.9(-1.6)
Charles	-14.3(0.0)	-15.0(+0.4)	-2.2(+0.7)	-19.0(-2.5)
Park	-32.8(-2.9)	-11.0(-0.3)	-3.2(+3.4)	-1.9(-1.3)
Downtown	-9.3(+0.9)	-3.6(+1.1)	0.0(+2.1)	+2.0(+1.5)
South Station	-11.7(-3.8)	-6.7(-0.6)	+9.7(+3.4)	+13.7(-0.2)

Two hour aggregation is also compared for all blocks before and after calibration. The comparison shows that the improvement is considerable. It can be seen clearly from Figure 6 that before calibration, most of the blocks with high runtime values, which are all station blocks or blocks where congestion occurs, are underestimated, while the blocks with small runtime values, which are in most cases less frequently impacted by congestion, are estimated very well. This infers that free-run train movement is better captured than that with dwell time and congestion.

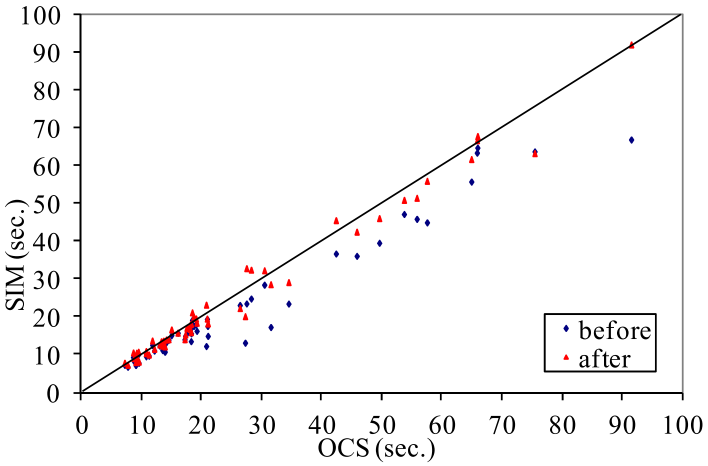

Figure 6: Block runtime before and after calibration

5 CONCLUSION

Urban rail operations and control simulators are valuable tools for system evaluation. Calibration is one of the key steps for a simulator to be useful. An operations and control simulator is usually integrated with several stochastic models, and involves a number of inputs and parameters, which are difficult to calibrate without a systematic methodology. Based on a simulation done with SimMETRO, a calibration process is developed and solved by SPSA method. It provides not only an approach to systematically adjust the model parameters, but also fine-tuning the simulation input that are not strictly reliable, such as a-priori passenger demand. The study shows that the calibration process improves the parameters and refines the input, in some cases, significantly. The measurements comparisons done before and after calibration show that the simulator behaves more closely to the actual system observations with relatively small computational effort. This proves that the method is sound and the algorithm is efficient.

ACKNOWLEDGMENTS

The research was partially supported by U.S. National Science Foundation, project CMS-0339108, and a grant from MBTA. The authors would like to thank MBTA for the data used in this study.

REFERENCES

Koutsopoulos, H. N., and Z. Wang. 2007. "Simulation of Urban Rail Operations: Application Framework." In *Transportation Research Record: Journal of the Transportation Research Board, No. 2006*, 84-91. Washington, DC: Transportation Research Board, National Research Council.

Puong, A. 2000. "Dwell Time Model and Analysis for MBTA Red Line." Massachusetts Institute of Technology Research Memo.

Spall, J. C. 1998. "Implementation of the Simultaneous Perturbation Algorithm for Stochastic Optimization." *IEEE Transactions on Aerospace and Electronic Systems* 34(3):817-823.

Wang, Z. 2006. "Simulation of Urban Rail Operations and Control." PhD thesis, Northeastern University.

AUTHOR BIOGRAPHIES

ZHIGAO WANG is a Senior Program Specialist and Project Manager in the China Sustainable Transportation Center (CSTC). Before joined CSTC, he received his Ph.D. in Transportation from Northeastern

University in Boston, MA and subsequently worked as an Analyst for TranSystems in the Boston area. His email address is zhigaowang@chinastc.org.

HARIS N. KOUTSOPOULOS is Professor and Head of the Traffic and Logistics Division in the School of Architecture and the Build Environment at the Royal Institute of Technology (KTH) in Sweden. He received his M.Sc. and Ph.D. degrees in Transportation Systems from MIT. His research focused on ITS, simulation-based dynamic traffic behavior, and calibration of simulation models, He has published extensively in these areas and he is a member of Traffic Flow Theory Committee at TRB, and in the editorial board of two international journals. His email address is hnk@infra.kth.se.