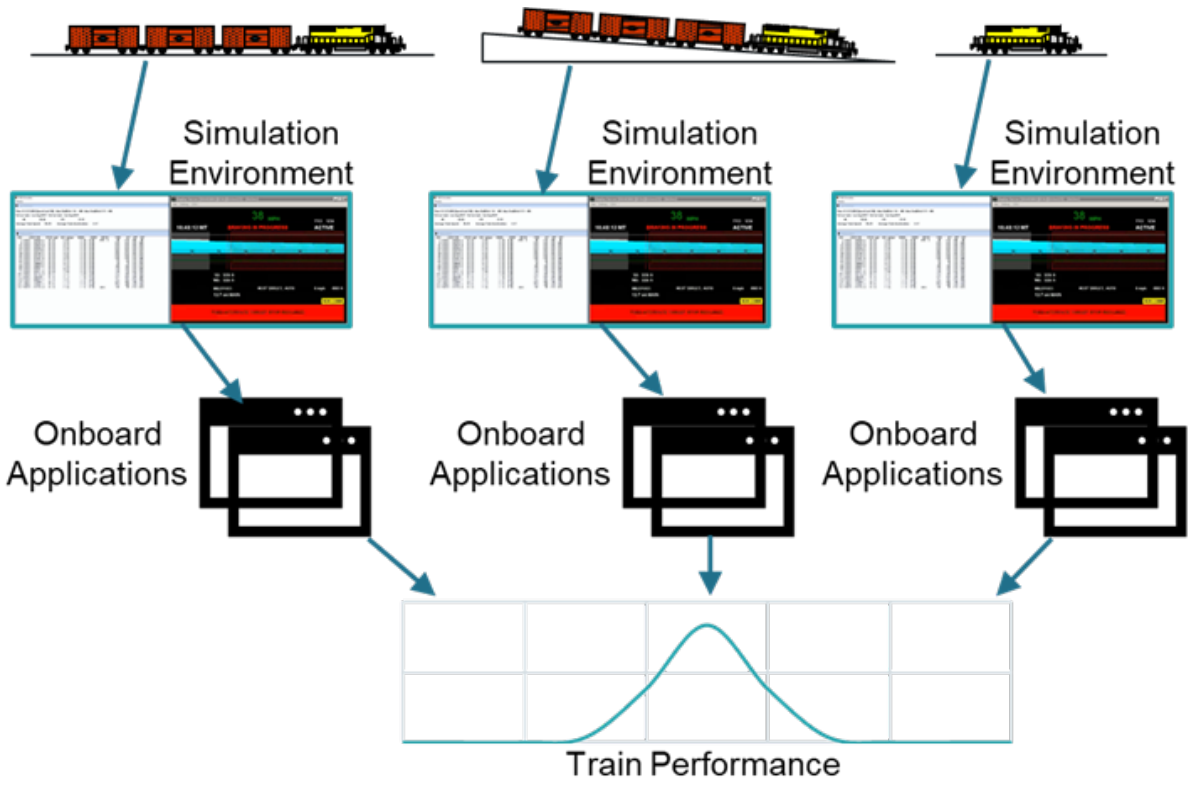
U.S. Department
of Transportation
Federal Railroad
Administration

Office of Research,
Development and Technology
Washington, DC 20590

Fully Scalable Train Braking Simulation Environment



REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
26-07-2022	Technical Report	Oct. 1, 2021 – Sept. 30, 2022

4. TITLE AND SUBTITLE
Fully Scalable Train Braking Simulation Environment

5a. CONTRACT NUMBER
DTFR5311D00008L

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Rachel Anaya, ORCid: 0000-0002-2925-9054
Shad Pate, ORCid: 0000-0001-7191-5919

5d. PROJECT NUMBER

5e. TASK NUMBER
693JJ621F000005

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Transportation Technology Center, Inc.
55500 DOT Road
PO BOX 11130
Pueblo, CO 81001-0130

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Department of Transportation
Federal Railroad Administration
Office of Railroad Policy and Development
Office of Research, Development, and Technology (RD&T)
Washington, DC 20590

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
DOT/FRA/ORD-23/19

12. DISTRIBUTION/AVAILABILITY STATEMENT
This document is available to the public through the FRA website

13. SUPPLEMENTARY NOTES
COR: Francesco Bedini Jacobini

14. ABSTRACT
Transportation Technology Center, Inc. (MxV Rail), through a research project funded by Federal Railroad Administration (FRA), developed a Concept of Operations (ConOps), documented the infrastructure design, and developed software to support a virtualized fully scalable train braking simulation environment (FSTBSE) that expands upon the capabilities of an existing simulation environment originally developed for evaluation of braking algorithms for Positive Train Control (PTC) applications. The new environment will provide the ability to complete simulations more efficiently and support a concept for on-demand simulations to support new software functions and processes with the potential to improve the safety and operational efficiency of train control and other applications, such as Interoperable Train Control (ITC) PTC and Energy Management Systems (EMS).

15. SUBJECT TERMS
Fully scalable train braking simulation environment, brake enforcement algorithm, PTC

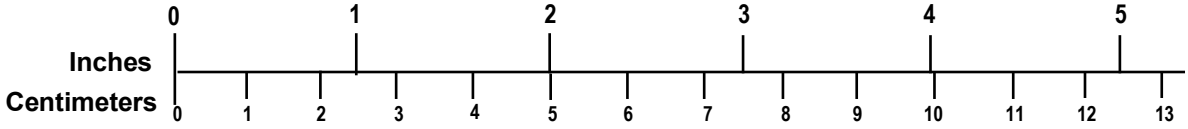
16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Rachel Anaya, Sr. Engineer I
Unclassified	Unclassified	Unclassified		90	**19b. TELEPHONE NUMBER** *(Include area code)* 719-584-0607

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

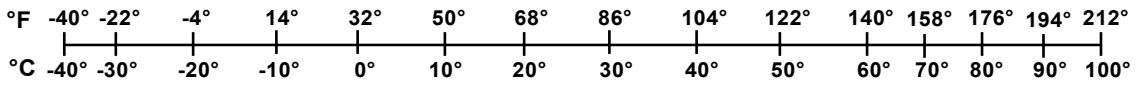
METRIC/ENGLISH CONVERSION FACTORS

ENGLISH TO METRIC	METRIC TO ENGLISH
LENGTH (APPROXIMATE)	**LENGTH (APPROXIMATE)**
1 inch (in) = 2.5 centimeters (cm)	1 millimeter (mm) = 0.04 inch (in)
1 foot (ft) = 30 centimeters (cm)	1 centimeter (cm) = 0.4 inch (in)
1 yard (yd) = 0.9 meter (m)	1 meter (m) = 3.3 feet (ft)
1 mile (mi) = 1.6 kilometers (km)	1 meter (m) = 1.1 yards (yd)
	1 kilometer (km) = 0.6 mile (mi)
AREA (APPROXIMATE)	**AREA (APPROXIMATE)**
1 square inch (sq in, in^2) = 6.5 square centimeters (cm^2)	1 square centimeter (cm^2) = 0.16 square inch (sq in, in^2)
1 square foot (sq ft, ft^2) = 0.09 square meter (m^2)	1 square meter (m^2) = 1.2 square yards (sq yd, yd^2)
1 square yard (sq yd, yd^2) = 0.8 square meter (m^2)	1 square kilometer (km^2) = 0.4 square mile (sq mi, mi^2)
1 square mile (sq mi, mi^2) = 2.6 square kilometers (km^2)	10,000 square meters (m^2) = 1 hectare (ha) = 2.5 acres
1 acre = 0.4 hectare (he) = 4,000 square meters (m^2)	
MASS - WEIGHT (APPROXIMATE)	**MASS - WEIGHT (APPROXIMATE)**
1 ounce (oz) = 28 grams (gm)	1 gram (gm) = 0.036 ounce (oz)
1 pound (lb) = 0.45 kilogram (kg)	1 kilogram (kg) = 2.2 pounds (lb)
1 short ton = 2,000 pounds (lb) = 0.9 tonne (t)	1 tonne (t) = 1,000 kilograms (kg) = 1.1 short tons
VOLUME (APPROXIMATE)	**VOLUME (APPROXIMATE)**
1 teaspoon (tsp) = 5 milliliters (ml)	1 milliliter (ml) = 0.03 fluid ounce (fl oz)
1 tablespoon (tbsp) = 15 milliliters (ml)	1 liter (l) = 2.1 pints (pt)
1 fluid ounce (fl oz) = 30 milliliters (ml)	1 liter (l) = 1.06 quarts (qt)
1 cup (c) = 0.24 liter (l)	1 liter (l) = 0.26 gallon (gal)
1 pint (pt) = 0.47 liter (l)	
1 quart (qt) = 0.96 liter (l)	
1 gallon (gal) = 3.8 liters (l)	
1 cubic foot (cu ft, ft^3) = 0.03 cubic meter (m^3)	1 cubic meter (m^3) = 36 cubic feet (cu ft, ft^3)
1 cubic yard (cu yd, yd^3) = 0.76 cubic meter (m^3)	1 cubic meter (m^3) = 1.3 cubic yards (cu yd, yd^3)
TEMPERATURE (EXACT)	**TEMPERATURE (EXACT)**
[(x-32)(5/9)] °F = y °C	[(9/5) y + 32] °C = x °F

QUICK INCH - CENTIMETER LENGTH CONVERSION



QUICK FAHRENHEIT - CELSIUS TEMPERATURE CONVERSIO



For more exact and or other conversion factors, see NIST Miscellaneous Publication 286, Units of Weights and Measures. Price $2.50 SD Catalog No. C13 10286

Updated 6/17/98

Acknowledgements

MxV Rail appreciates the assistance of the advisory group (AG) in providing information and feedback during the development of the Concept of Operations (ConOps) for the Fully Scalable Train Braking Simulation Environment (FSTBSE) and Railinc for their assistance in the development of the FSTBSE.

Contents

Illustrations

Tables

Executive Summary

Transportation Technology Center, Inc. (herein known as MxV Rail), through a research project funded by the Federal Railroad Administration (FRA), developed a Concept of Operations (ConOps), documented the infrastructure design, and developed software to support a virtualized Fully Scalable Train Braking Simulation Environment (FSTBSE). The FSTBSE expands upon the capabilities of an existing simulation environment originally developed for evaluating braking algorithms for Positive Train Control (PTC) applications, will allow more efficient simulations, and will support a concept for on-demand simulations in support of new software functions and processes that may improve the safety and operational efficiency of train control and other applications, such as Interoperable Train Control (ITC), PTC, and Energy Management Systems (EMS).

The existing simulation environment is used to model train braking and enforcement algorithms (EAs) for PTC systems. This technology was designed to improve the safety of railway operations by enforcing a train stop, when needed, through application of the train air brakes. When EAs are updated, evaluation of their performance through field testing in real world conditions is cost and time prohibitive. Consequently, evaluation of PTC EAs has been a key use of the simulation environment to date. While the process to evaluate an EA in the simulation environment is more time- and cost-efficient than real world testing, it requires several weeks for each new version of an EA to be evaluated. Additionally, the railroad industry has identified a need for a simulation environment capable of supporting on-demand simulations, i.e., the ability for railroads to create and run simulations using specific railroad operational conditions in near real-time. Examining the limitations of the current simulation environment and the additional needs of the railroad industry led to the development of the FSTBSE concept.

During this project, MxV Rail developed a ConOps for the FSTBSE which documents the FSTBSE infrastructure design to support the capabilities of the current simulation environment as well as the expanded capabilities of the FSTBSE. They developed a program to convert PTC track data files compliant with the Association of American Railroads (AAR) standard for the PTC track data model into Train Operations and Energy Simulator (TOES™) track files, and identified the future work necessary to support FSTBSE implementation and deployment. This project was performed in collaboration with Railinc and an advisory group (AG) consisting of Class I and short line railroad personnel. Railinc supported the effort through implementation efforts and will host the new simulation environment.

During identification and documentation of the infrastructure design, MxV Rail and Railinc identified the elements of the infrastructure and software functions required to support the concept. During this process, it was determined that a cloud-based solution, running in a Linux environment, is the desired approach for the simulation environment. The cloud-based structure allows Railinc to develop the infrastructure necessary to provide a scalable environment.

The software application developed during this project supports the desired on-demand capabilities of the simulation environment by allowing users to upload PTC Extensible Markup Language (XML) files containing PTC track data compliant with the PTC track data model standard and convert them to a format that can be used by TOES for simulations. This allows users to model train operations over real world tracks without extensive manual input.

Initial work on this project was completed between October 2021 and September 2022.The ConOps created in this project will be used to guide the implementation of the FSTBSE as the project continues through future phases. Ongoing collaboration with Railinc will be required for these future phases, during which time the FSTBSE will be implemented and deployed. Recommendations for future phases of the project include a second phase supporting Railinc efforts in development and evaluation of the functions and performance of the FSTBSE, and a third phase to expand FSTBSE support of direct interaction with railroad systems.

1. Introduction

The Federal Railroad Administration (FRA) funded a research project conducted by Transportation Technology Center, Inc. (MxV Rail) to develop the concept, document the infrastructure design, and support the initial development tasks for a virtualized Fully Scalable Train Braking Simulation Environment (FSTBSE) that expands upon the capabilities of an existing train braking simulation environment to complete simulations more efficiently and support a concept for on-demand simulations.

On-demand simulations are conceived to support new software functions and processes with the potential to improve the safety and operational efficiency of train control and other applications (i.e., Interoperable Train Control (ITC), Positive Train Control (PTC), and Energy Management Systems (EMS)), allow railroad users to model train operations for specified train consists, tracks, and train handling, and to evaluate brake applications for specific conditions.

1.1 Background

PTC is a technology designed to improve the safety of railway operations by enforcing a train stop, when needed, through application of the train air brakes. PTC applications typically utilize a predictive braking enforcement algorithm (EA) to estimate the braking distance of the train and to determine when a penalty brake application is required.

Verifying the performance of a PTC EA through field testing can be costly and time-consuming, and it is difficult to fully test and validate the process without significant impact to railroad operations. With FRA funding, MxV Rail developed a process to use the Train Operations and Energy Simulator (TOES™) program and customized software to simulate the performance of a PTC EA for freight operations. This process is described in several reports (Brosseau, Moore Ede, Pate, & Wiley, 2013; Pate, Anaya, & Holcomb, 2019) and has been demonstrated as an efficient and effective method for verification of PTC EAs.

To facilitate this process, a Monte Carlo simulation methodology was developed to perform a comprehensive evaluation of PTC EAs, including approximately 3,800 simulation scenarios, with each scenario comprised of a train consist, simulation speed, track grade, and a target speed and location. Train consists used in the simulation scenarios included a mix of unit trains (e.g., coal, tank, refrigerated box, grain, and auto-racks), mixed general freight, and intermodal freight. Each train type has multiple configurations for length, locomotive power (i.e., head-end only or distributed power), and loading conditions. Trains were operated on grades ranging from a 1.5 percent incline to a 2.8 percent decline and run at maximum authorized speeds as well as intermediate and slow speeds on certain grades. For each scenario, 100 Monte Carlo simulations were generated. For each simulation, train consist and simulation variables were selected from defined ranges and distributions, representing the real world differences of these variables. The simulations were then executed in the simulation environment. Results from the simulations (i.e., the scenario details, enforcement location and speed, stopping location, and target speed and location) were recorded and analyzed.

1.2 Objectives

The primary purpose of this project was to develop the FSTBSE concept, infrastructure design, and initial development by accomplishing the following tasks:

- Identify and develop use cases for running simulations in a scalable virtualized simulation environment
- Develop a Concept of Operations (ConOps) for the FSTBSE from the use cases.
- Document an infrastructure design that allows the FSTBSE to mimic the functionality of the current simulation environment and support the expanded capabilities of the FSTBSE identified in the ConOps
- Support Railinc during the initial development and implementation of tasks for the FSTBSE
- Develop a program to create TOES track files from existing PTC track files to support on-demand simulations
- Document the possible changes needed to the Test Controller and Logger (TCL)-EA interface specification document to support capabilities identified in the use cases and ConOps

1.3 Overall Approach

The following describes the overall approach to the project. MxV Rail:

- Collaborated with a railroad advisory group (AG) to define the use cases for the FSTBSE to support the capabilities of the current simulation environment and additional identified capabilities
- Collaborated with Railinc through bi-weekly virtual meetings and in-person meetings to review and document the infrastructure design needed to support the FSTBSE
- Collaborated with Railinc to identify the elements of the FSTSBE infrastructure design that would be supported by Railinc
    - Identified and documented potential changes to the TCL-EA interface needed to support the FSTBSE concept
    - Developed the ConOps, which was reviewed by the AG and Railinc
- Developed a program to convert PTC track files to TOES track files, i.e., TOES Track File Generator (TOES TFG), using sample PTC track files provided by the AG (this task was concurrent with the above tasks)

1.4 Scope

The scope of this project identified use cases for the FSTBSE, including those for on-demand simulations, and developing a ConOps for the FSTBSE.

Within the scope of the project, researchers planned to collaborate with Railinc to develop the infrastructure design to best fulfill the objectives of the FSTBSE. Development of software to support the implementation of the infrastructure design was outside the scope of this project, although providing support to Railinc in initial development tasks was within the scope.

The scope of this project also included developing the capability to create simulation track files from PTC track data that can be used in TOES.

Documentation of modifications to TCL or modifications to the interface between TCL and EA were within the scope, but implementing these changes was not within the scope of this project.

1.5 Organization of the Report

- Section 1 introduces the project and defines the objectives, approach, and scope of the research.
- Section 2 provides an overview of the existing simulation environment used as the basis for the FSTBSE concept.
- Section 3 presents the development of the ConOps for the FSTBSE.
- Section 4 describes the initial development support for the FSTBSE.
- Section 5 provides an overview of the program developed to create TOES track files from PTC Extensible Markup Language (XML) track files.
- Section 6 summarizes potential changes to the TCL-EA interface that may be needed to support implementation of the FSTBSE.
- Section 7 summarizes the findings of the project.
- Appendix A contains the ConOps.

2. Current Braking Algorithm Simulation Environment

A simulation environment currently exists to support the PTC EA evaluation methodology described in Section 1. Section 2.1 is an overview of the software components used for this simulation environment. A Structured Query Language (SQL) database stores the information from these software components to create the files needed to run simulations and generate results. The simulation environment also requires access to a centralized file storage containing input files used by TOES during the simulations.

2.1 Simulation Testing Tools

The software tools used within the current simulation environment include TOES TCL and the EA being evaluated. The following subsections are a brief overview of these software tools. More detailed descriptions are available in previous FRA research reports (Brosseau, Moore Ede, Pate, & Wiley, 2013; Pate, Anaya, & Holcomb, 2019).

2.2 Simulation Model – TOES

TOES is a longitudinal train dynamics model developed by the Association of American Railroads (AAR) that models the location, velocity, acceleration, forces acting on the railcar, and brake system component status of every railcar in a specified train consist at every time step of the simulation.

The model allows the user to enter:

- Specific characteristics for each railcar in the train consist
- Track characteristics that affect the longitudinal motion of the train (i.e., track grade and curvature) allowing any section of track to be modeled
- Environmental conditions that can affect the longitudinal motion of the train, such as ambient temperature and the coefficient of friction between the wheels and brake shoes
- Train handling commands, such as throttle and brake settings, at any time step in the simulation

2.2.1 TCL Software

TCL has the capability to generate and execute thousands of braking enforcement simulations using a Monte Carlo probability method. TCL uses operating scenarios and parameter variation distributions entered by the user.

The TCL application performs the following three major functions:

- Generation of random simulation inputs
- Execution of individual simulations
- Logging of output data

To generate simulation input data, the user sets up a batch of test scenarios for evaluation. The user selects a train consist and track profile and enters the initial train speed and location, as well

as the target stopping location for each test scenario in the batch. Below is a summarized list of TCL functions:

- Train consist modeling – Creates train consists using existing locomotive and car models, specifying the location and load of each vehicle within the train consist.
- Track profile – Supports user selection of a track profile from current profiles modeled in the simulation environment.
- Train handling – Supports manual creation of a train handling text file using a specified format to set up desired train handling commands. TCL can select existing train handling files to use in simulations.
- Initial conditions – Supports setup of initial simulation conditions for each simulation scenario. These include starting location, simulation test speed, starting throttle or dynamic settings, target location, and speed.
- Configuration of EA to support emergency brake backup – Supports configuration of the EA for emergency brake backup to be enabled or disabled from TCL. When enabled, emergency brake applications can be set to be applied from the head-end only, or as a two-way application.
- Dynamic brake behavior – The dynamic brake can be enabled or disabled, which will determine whether the dynamic brake in use prior to a brake application triggered by the EA will remain in use after it is triggered. Behavior can be defined for both head-end locomotives and remote distributed power locomotives.
- Locomotive air brake behavior – This can be configured within TCL and can be set up using the number of cars in the train consist. Behavior can be configured for both head-end locomotives and remote distributed power locomotives.
- Back office brake force – Supports user selection of the desired method for calculating and providing brake force to the EA to simulate back office processes that provide this information.

Train consists are defined by the user by selecting the preferred railcars and arranging them in the desired order. Each railcar is defined by the nominal components and characteristics of the railcar and their potential variation, which can be specified by the user. The variation of the railcar components and characteristics can be represented by several different distribution methods, allowing the user to define the variability of a given parameter to match its actual, real world variation. The user also characterizes the potential variation of environmental attributes as well as variations due to errors in reported data, such as track characteristics, train speed, and train location.

The user selects the number of simulations the TCL software will run for each scenario in the Monte Carlo batch. The TCL software then generates the input data for each simulation scenario by randomly selecting values for the variable parameters from the input distributions defined by the user.

Once the simulation input data is generated, the user can run the batch through the TCL software. The TCL application runs every simulation scenario individually in the simulation model by advancing the train toward the target at a given speed. At each second of simulation time, the

simulation model reports train status data to TCL and then passes it to the EA. When the EA predicts an impending target overrun, it sends a command to initiate a penalty brake enforcement to the TCL application, which executes the penalty in the simulation model. TCL continues to advance the simulation until the train is stopped. The EA can also send a command to initiate an emergency brake enforcement (if so configured), which TCL then executes in the simulation model.

Once the train has stopped the simulation is complete, and the TCL software logs the output data in a database for post-process analysis.

2.2.2 EAs

The intent of the braking EA evaluation methodology is that it can be applied to evaluate any freight EA for any PTC implementation. Therefore, the simulation environment treats the software implementation of the EA as a black box that communicates with the simulation components over a standard communications interface. A document detailing the communications process and protocols was prepared for EA software developers.

To allow for the most flexibility in the simulation setup, the interface was designed to communicate over transmission control protocol/internet protocol (TCP/IP). This allows the EA to be implemented as an executable software application running on the same machine as the TCL software, as a virtual machine (VM) with a separate IP address operating on the same hardware as the TCL software, or as software running on separate hardware that communicates over TCP/IP.

To allow for more efficient execution of the simulations, the interface was also designed with flexibility for initializing the simulation test process. The TCL software can execute the EA software directly if it is run on the same machine. Alternatively, an EA initialization module was developed that sends an initialization message to the EA software, indicating that the previous simulation is complete and the new simulation is beginning. This allows the EA software to re-initialize internal parameters and other data for the new simulation.

2.3 Generalized Simulation Process Using the Existing Simulation Environment

Simulation setup in the current simulation environment is done using TCL along with available data within a SQL database. The SQL database includes data for modeled TOES vehicles and TOES track files, previously created train consist files, and previously created batches. TCL is used to create a batch file containing simulations with the desired train consist, track, and operational settings, then it generates the files needed by TOES to run the simulations and stores them in a centralized location that is accessible by all simulation instances. TCL can also be used to create new TOES train consist files using modeled TOES vehicles stored in the SQL database. The new train consist file is saved to the database and can be used when setting up new simulations. If the train consist being created needs a new vehicle modeled, data for that vehicle is manually loaded into various SQL tables so it will be selectable when creating the train consist.

A simulation scenario is created within a batch, which can be comprised of a single simulation scenario or multiple scenarios. A simulation scenario is made up of four key components: a train consist, a track profile, initial conditions, and target information. Currently, batches and

simulation scenarios are created using TCL. When adding a simulation scenario to a batch, the user can select a train consist and track profile from available train consists and tracks stored in the SQL database as well as enter the desired initial conditions and target information. If the simulation scenario will use a new train consist or track profile, this needs to be created and stored in the SQL database before creating the simulation scenario. As described above, the train consist can be created using TCL; currently, however, adding a new track profile requires the file to be created manually outside of TCL, and the SQL database is then updated so the new track file is selectable when creating a simulation scenario.

Once a batch is created, that batch can be selected for execution through TCL. The user can set simulation-level parameters prior to the start of simulations, such as the generation of TOES train consist and simulation files, emergency brake backup configuration and behavior, dynamic brake behavior, locomotive brake behavior, etc. TCL will configure the machine using the simulation preferences, and it will begin executing the simulations by stepping through each simulation scenario defined within the batch. Figure 1 illustrates the organization of simulations within a batch. The scenarios and simulations can be expanded to show Simulation Batch 1 to Simulation Batch *n*, where each simulation batch has different test scenarios and simulations set up.

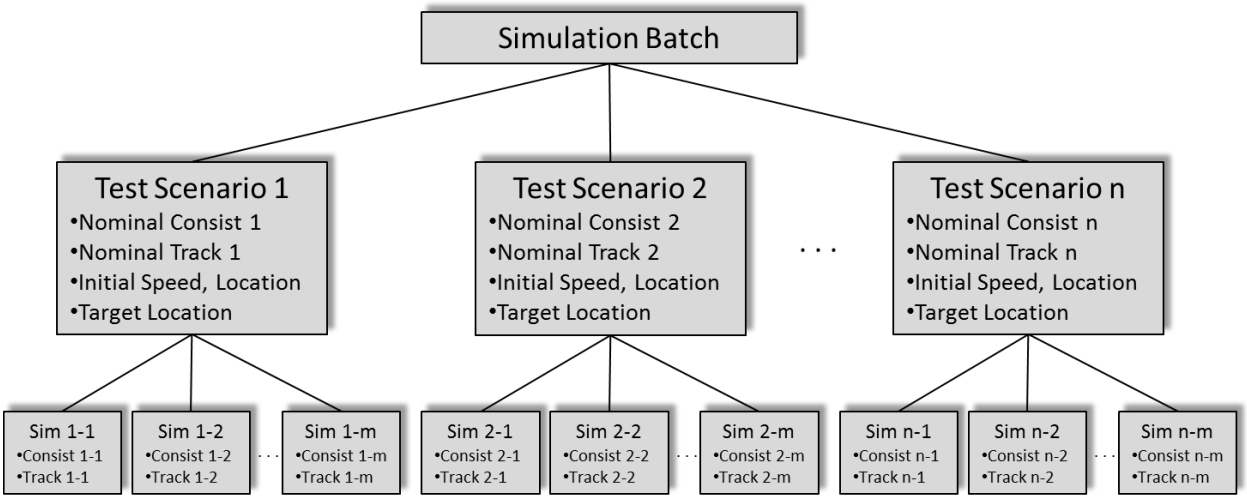


Figure 1. Organization of simulations

TCL will run each simulation to completion and will write a simulation results record to the SQL database. The user will then retrieve the simulation results data, through queries executed on the SQL database, for processing and analysis.

3. ConOps and Infrastructure Design

3.1 Development of the ConOps

In this phase of the project, MxV Rail, in collaboration with the AG, produced a FSTBSE ConOps. The FSTBSE objectives were to retain the core functionality of the current simulation environment while improving efficiency and expanding the functionality to allow users or user software applications to create on-demand simulations that allow railroad users or software applications to create and run simulations using their own railroad operations. Initially, the FSTBSE will allow railroad users to use the on-demand functionality by interacting with it manually to create and run the desired simulations, but the intent is that the final FSTBSE will allow users or user software applications to directly create and run simulations through an application program interface (API). The development of the ConOps included a review of the current simulation capabilities with the AG, as well as discussions of the desired improvements and additional functionality needed to support the objectives of the new simulation environment.

3.1.1 Key Limitations

During the development of the ConOps, the team identified the key limitations with the current simulation environment with respect to developing a system that meets the objectives of the FSTBSE. These key limitations fell within three categories: user access, scalability, and manual interaction.

3.1.1.1 User Access

User access for the current simulation environment is limited to users that are knowledgeable with the existing simulation tools.

3.1.1.2 Scalability

The current simulation environment uses physical servers to set up and run multiple instances of the simulation tools. There is an upper limit for the number of instances that can be run simultaneously, based on the size of the servers and the resources available to the machines running the simulation tools. Currently, most of the simulation software tools run on machines using a Windows operating system, which also introduces some scalability limitations due to the need for licenses and associated licensing costs.

3.1.1.3 Manual Interaction

The current simulation environment involves some level of manual user input during various steps of the overall simulation process. Initially, a user needs to configure the simulation tools, (i.e., TCL, TOES, or the EA) for every instance that is created. This includes setting up configuration files in the EA and TCL and setting up IP addresses on any machines running simulation tools for that instance. Every simulation instance also needs to be recreated or updated when one or more of the simulation tools is updated to a different version.

The current simulation environment also requires a user to be involved in the setup and starting of simulations awaiting execution of one or more instance. Initially, the user needs to verify the simulation software tools are running on each simulation instance being used and then the user either needs to access TCL on each instance or configure entries in a SQL database table to set

up simulation behavior and start the simulations. The user also needs to monitor simulations running across the instances and respond to any errors that occur.

Upon completion of the desired simulations, the user needs to manually retrieve the results from the SQL database and analyze the data to produce the data outputs desired.

3.1.2 Key Improvements

For the ConOps development, the team collaborated with the AG to identify the key improvements needed in the development of a system that meets the objectives of the FSTBSE. These improvements were identified from the two major objectives of the system: improved efficiency and supporting on-demand simulations.

3.1.2.1 Improved Efficiency

Based on the limitations identified within the current simulation environment, improved efficiency of the system can be realized by eliminating the hardware limitations and reducing the need for manual configuration, execution, and analysis of the simulations. To support this, the ConOps outlines a cloud-based solution to host the software components to allow for dynamic scalability of the environment using simulation demand, as well as the development and implementation of a Simulation Controller/Manager to reduce manual intervention by:

- Dynamically starting and configuring simulation instances as needed
- Allocating simulations awaiting execution to simulation instances
- Retrieving simulation results data and producing required output analysis

3.1.2.2 Supporting On-Demand Simulations

The capability to support on-demand simulations, which will provide users or user software applications the capability to set up user-defined simulations and run them in near real-time, is not available in the current simulation environment. Through discussions with the AG, it was determined that the initial implementation of this functionality will support simulations setup and run manually through a web interface, with future capabilities to support simulations setup and run automatically via software applications interfacing with the environment through API connections.

The team determined the functionality needed for on-demand simulations by reviewing the current simulation environment and the different types of simulation efforts currently used. This included reviewing and documenting the different simulation efforts to date as a set of use cases that outline the data required to set up and execute the simulations and the data output they provided. The use cases were shared and reviewed with the AG during regularly scheduled AG meetings and documented as supplemental information in the ConOps.

The identified functionality for on-demand simulations was used during the development of the ConOps and the major capabilities identified included the ability to:

- Create user-defined train consists
- Create user-specified track files
- Define desired simulation train handling

- Set up and run user-defined simulation scenarios
- Access data output from simulations
- Support API connections for automated simulations

3.2 Infrastructure Design

MxV Rail collaborated with Railinc, the organization that will develop and host the environment on behalf of the industry, and the AG to support the design of the simulation environment that will implement the functionality described in the ConOps. The team worked closely with Railinc to review the current simulation environment design to identify the key aspects to be considered in the infrastructure design of the new simulation environment. An overview of the current simulation environment design is described in Section 3.2.1, and the infrastructure design for the new simulation environment is described in Section 3.2.6.

3.2.1 Current Simulation Environment Design

The design of the current simulation environment contains a number of VMs housing the simulation tools (TCL, TOES, and EAs), a SQL database, and a central storage location. The VMs are configured to set up simulation instances, with each simulation instance having a single instance of TOES, TCL, and an EA. The simulation instances are set up so they are segregated from each other, allowing simulations to run in parallel. Figure 2 is an overview of the current simulation design. Each simulation instance includes a Windows VM with either a separate Linux VM housing the EA (for EAs provided in Linux) or the EA installed on the Windows VM (for EAs provided as a Windows executable).

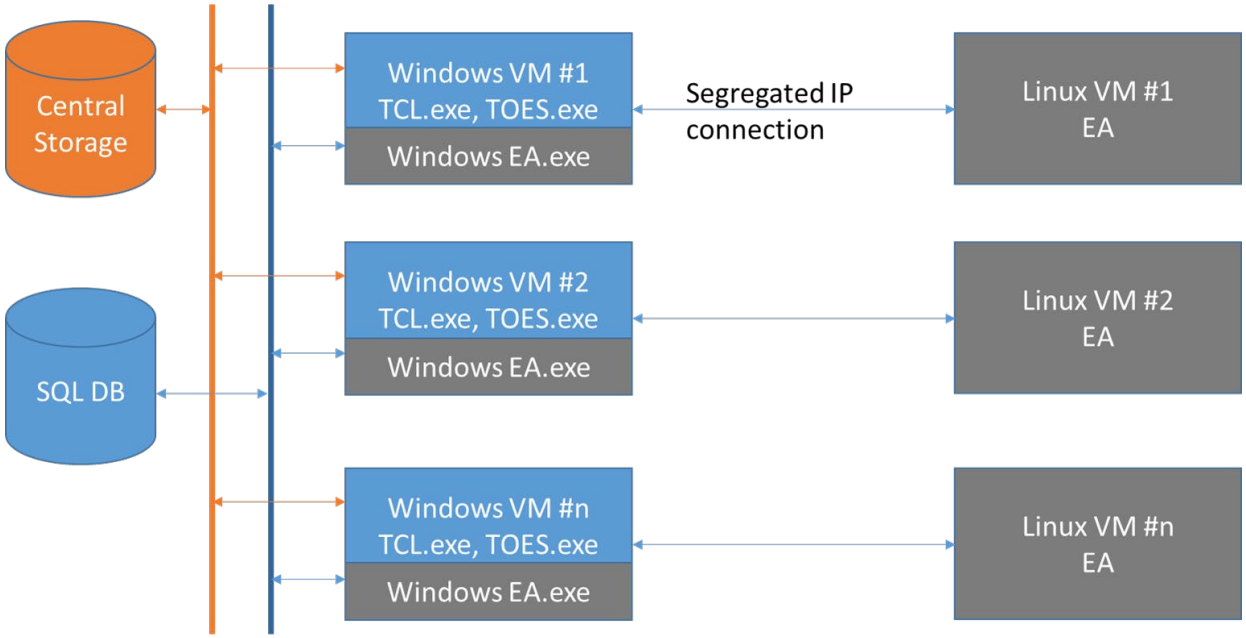


Figure 2. Current simulation design

The current simulation design is limited by the physical host, which supports up to 75 simulation instances. Each instance must be configured individually but all 75 simulation instances can be operated concurrently.

3.2.1.1 Configuration and Setup of Windows VMs

Initially, a Master Windows VM is created and configured with two IP addresses. The first IP address is used to set up communication with the SQL database and connect the VM to the central storage location. The second IP address is used to configure a segregated network over which the simulation tools communicate. The Master VM also has TCL and TOES installed and configured on its local storage. The other VMs are created as clones of the Master, with each clone requiring modification to configure the simulation tools to communicate on a specific segregated network. The Master VM is updated and saved whenever changes are required for the TCL or TOES simulation tools. This update requires new clones to be created and configured to propagate the changes throughout all the VMs.

3.2.1.2 Configuration and Setup of EA

EAs are provided by vendors for evaluation in the simulation environment. The EA software is treated as a black box, with summary simulation results recorded in a results table within the SQL database and log files saved in the central storage during the execution of the simulations. The summary results are used to analyze the simulations. The log data recorded for each simulated time step contains the messages that are sent between the EA and TCL and the messages sent between TCL and TOES. The log data can be used to review the messages between the simulation tools, mainly for troubleshooting purposes.

For EAs provided for use in a Linux VM, a Master EA Linux VM is created that is configured to run in a Linux environment. For each additional simulation instance, VM clones are created from the Master VM. Each clone requires additional configuration to set up a segregated network for communications with the associated Windows VM running TOES and TCL.

For EAs provided as a Windows executable, the executable is loaded on the desired Windows VM running TCL and TOES with the simulation tools requiring some additional configuration to communicate on each Windows VM's local network.

3.2.1.3 SQL Database

The SQL database hosts a series of SQL tables that are used by TCL to create simulation files and store results and information used for data analysis. These tables are used to store TOES vehicle and track information, user-created train consist data, user-created batch information, ranges and distribution types for varied parameters, summary train consist and summary simulation information for every simulation created, and simulation results for each simulation executed.

3.2.1.4 Central Storage Location

A central storage location is used in the current simulation environment to allow any simulation instance access to the files needed for simulations set up by the user. Without the central storage location, these files would have to be created or copied locally to a simulation instance for TOES to run the simulations. TCL is configured to access the central storage location and write TOES train consist and simulation files to this location whenever the user directs TCL to create the files. TCL uses information from the various SQL database tables as well as settings from the TCL preferences input to create the files requested and to save them in the central storage location when complete.

When running simulations, TCL points TOES to the location of the necessary TOES train consist, track, and command files so TOES can initialize the desired simulation. TCL shows TOES where any output log files are written. TCL also uses the central storage location to record the messages passed between TOES and TCL and those passed between the EA and TCL in separate log files.

3.2.2 New Simulation Environment Design

The new simulation environment was designed to ensure the capabilities of the current environment remain while addressing key limitations and supporting the expanded functionality described in the ConOps. The core simulation functionality and methodology will remain the same with the simulation tools (i.e., TCL, TOES, and an EA) interacting to execute the simulations.

The design for the new simulation environment has three key, logical components:

1. User Interface
2. Simulation Controller/Manager
3. Scalable Simulation Environment

An overview of each of these components is provided in the subsequent subsections with additional detail of the envisioned capabilities and functionality provided in Section 5 of the ConOps.

3.2.2.1 User Interface

From the typical user's perspective (a typical user refers to any user with access to set up and request simulations), the user interface will be the only mechanism for interacting with the new simulation environment. Figure 3 is an overview of the user interface. The blue shading in the diagram indicates the elements of the user interface accessible by the typical user.

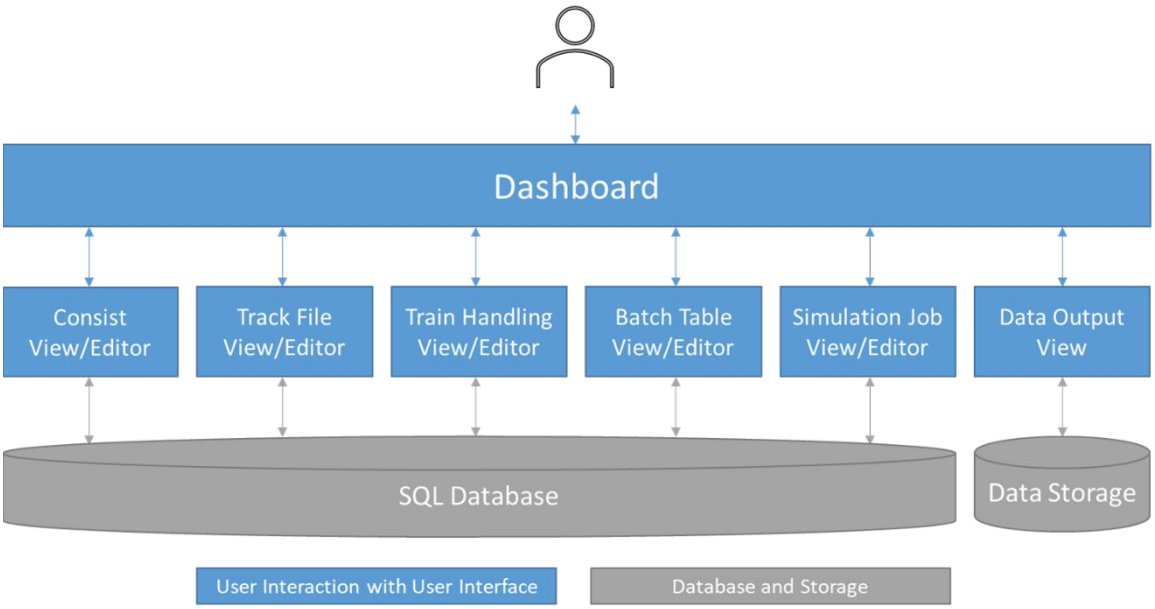


Figure 3. Overview of user interface

The following list provides an overview of the capabilities available to the user for each blue box shown in Figure 3:

- Dashboard – Users will have an overview of past and current simulation jobs as well as notifications for available simulation results and analyses not yet viewed. Users will have access to all the other View/Editors from the Dashboard.
- Train Consist View/Editor – Users will be able to view previously modeled train consists and to create new train consists.
- Track File View/Editor – Users will be able to view previously modeled track files and to create new track files.
- Train Handling View/Editor – Users will be able to view previously developed train handling files as well as create new user-defined train handling behavior.
- Batch Table View/Editor – Users will be able to view simulation scenarios modeled within existing batch tables, as well as create and save new simulation scenarios as part of a new batch table.
- Simulation Job View/Editor – Users will be able to request a simulation job, including the user selected simulation scenario(s), user-defined simulation behavior, and data outputs.
- Data Output View – Users will have access to view and download simulation results.

From the user interface, users will be able to create user-defined train consists, track files, and train handling files and use these files to set up batches of simulations containing one or more simulation scenarios. Users can request a job to run simulation scenarios with user-defined behavior and user-defined output data. Upon completion of the simulation job, users can access the user interface to view the data output and analysis through the Data Output View. Initially, users will be able to access the user interface through a web interface developed by Railinc. The team intends to later implement API connections with railroad back office systems to automate tasks within the user interface.

3.2.2.2 Simulation Controller/Manager

The Simulation Controller/Manager will automate tasks that currently require user involvement within the current simulation environment, such as the creation and setup of simulation instances, management of simulations requiring execution, and gathering required output data for simulation analysis.

The Simulation Controller/Manager will monitor simulation requests created in the user interface to dynamically start simulation instances within the scalable environment. This includes managing the setup, creation, and execution of the simulations. Upon completion, the Simulation Controller/Manager will pull the required data outputs and perform the data analysis selected by the user when setting up the job in the Simulation Job View/Editor. Figure 4 is an overview of how the Simulation Control/Manager interfaces with the SQL database to monitor available simulation job(s), as well as how the Simulation Control/Manager interacts with the scalable simulation environment to start instances for the simulation job(s).

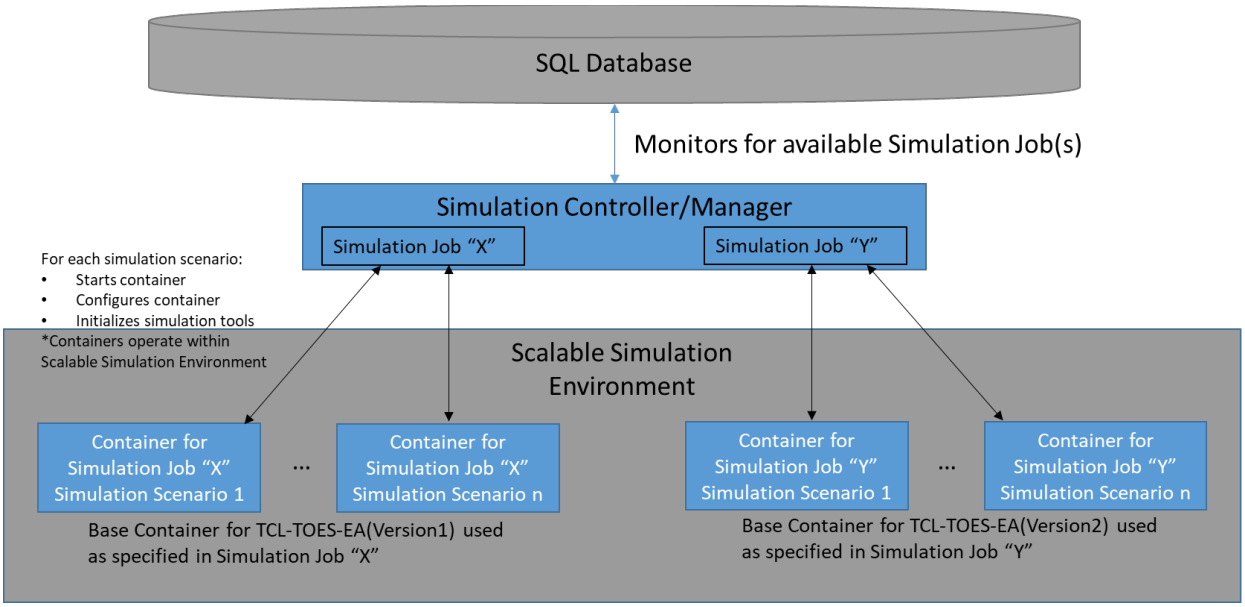


Figure 4. Overview of simulation controller/manager running simulation scenarios in scalable simulation environment

3.2.2.3 Scalable Simulation Environment

The scalable simulation environment will have the capability to store available combinations of simulation tools where the combinations differ because of the EA included, i.e., TCL-TOES-EA (Vendor A Version 1), TCL-TOES-EA (Vendor A Version 2), TCL-TOES-EA (Vendor B Version 1), etc. These tool combinations will be stored in containers within the scalable simulation environment. A container is a preconfigured set of simulation tools that can be saved as a "base container" and accessed by the Simulation Controller/Manager. The use of containers will allow the Simulation Controller/Manager to initiate multiple instances of the tools without the limitations present in the current simulation environment, mainly without the need to manually set up and configure the simulation tools. The number of parallel instances will be limited only by the capacity of the infrastructure, primarily the amount of cloud-based computing available for the scalable simulation environment.

Through evaluation of the simulation environment design and the desired capabilities, a Linux environment was chosen for the scalable simulation environment. (See Section 4.3 for additional information.) This will also require all simulation tools to operate within the Linux environment.

4. Development Support for the FSTBSE

The FSTBSE is being developed and housed by Railinc with the support of MxV Rail. Development includes the creation of the user interface, the Simulation Controller/Manager, and the scalable simulation environment to support the capabilities outlined in the ConOps, which includes supporting the capabilities available with the current simulation environment.

In collaboration with Railinc, the team conducted detailed reviews of the current simulation environment, the functionality needed for the FSTBSE, how that functionality can be achieved, and what changes are needed to the current simulation environment to support the desired functionality. The following subsections consider the three key components of the FSTBSE architecture (i.e., the user interface, the Simulation Controller/Manager, and the scalable simulation environment) and outline the findings and the planned development path. Further development, testing, and deployment of this environment will require work beyond the scope of this phase of the project.

4.1 FSTBSE User Interface

The functionality identified for the FSTBSE user interface is performed in the current environment by manually accessing the SQL database tables or through multiple interactions with the TCL user interface. By analyzing how TCL and users interact with the SQL database in the current environment, the team was able to identify the activities and functionality that will shift from TCL to the FSTBSE user interface. Through implementation review and discussion, it was decided that Railinc will develop a web-based user interface that will provide users the necessary functionality currently provided by TCL, along with additional capabilities not currently available within TCL, without compromising the security of the Railinc environment. To facilitate the new interface, an open-source, object-relational SQL database was found to be more useful than the existing SQL database, since moving to a new database will support a wider range of operating systems and allow easier integration with Railinc's current infrastructure. Therefore, the functionality that remains with TCL requiring access to the SQL database will need to be updated to access the new database correctly. The following subsections provide an overview of these necessary updates. Because some of the existing simulation software tools will remain in the FSTBSE, this section indicates the changes that will be made by Railinc as part of the new development, and the changes that will be made to the existing software components.

4.1.1 Consist View/Editor

4.1.1.1 FSTBSE

The FSTBSE user interface will have read/write access to all SQL tables that pertain to the viewing, modification, or creation of train consists. Railinc will implement functionality outlined within the ConOps for the Consist View/Editor and for saving the necessary data to the SQL tables to allow TCL to continue to generate TOES train consist files.

4.1.1.2 Existing System

User functionality pertaining to viewing, modifying, or creating train consists will be removed within the SQL database through the consist editing screen in TCL. TCL will retain read access to the SQL tables to generate TOES train consist files needed for execution of TOES

simulations, drawing from the information provided in the Consist View/Editor. TCL will also have write access to the SQL database for recording the summary train consist information as train consist files are generated. SQL database connections that remain within TCL will be updated to interface with the new open-source, object-oriented SQL database.

4.1.2 Track File View/Editor

4.1.2.1 FSTBSE

The FSTBSE user interface will have the ability to create user-generated track files and store them as TOES track files either in a central storage location or in a new SQL database table that can be used to generate the TOES track file. The best option for storage will be determined by calculating which option provides the best performance. The FSTBSE user interface will also have read/write access to the SQL table that stores the available track names for track files within the FSTBSE. Railinc will implement functionality outlined within the ConOps for the Track File View/Editor and will save the necessary data to the central storage and/or SQL tables so TCL will continue to be able to generate TOES text track files.

4.1.2.2 Existing System

TCL will be updated as needed to generate TOES track files when a request to generate TOES simulation files is received by the Simulation Controller/Manager. SQL connections within TCL that are relevant to this functionality will be updated to interface with the new open-source, object-oriented SQL database.

4.1.3 Train Handling File View/Editor

4.1.3.1 FSTBSE

The FSTBSE user interface will have the ability to create user-generated train handling files as a text file and either store them in a central storage location or in a new SQL database that can be used by TCL to issue train handling commands during a simulation. The best option for storage will be determined by calculating which option provides the best performance. The FSTBSE user interface will have read/write access to the SQL tables related to the train handling files. Railinc will implement functionality outlined within the ConOps for the Train Handling File View/Editor and will save the necessary data to the files and/or SQL tables so TCL will be able to use the train handling files during TOES simulations.

4.1.3.2 Existing System

TCL will be updated as needed to read the train handling files created by the Train Handling File View/Editor and to issue train handling commands in TOES during a simulation. TCL can currently run simulations with train handling files provided in a specific format, and it is envisioned that a similar process will be used so TCL will have access to the same information during a simulation. SQL connections within TCL that are relevant to this functionality will be updated to interface with the new open-source, object-oriented SQL database.

4.1.4 Batch Table View/Editor

4.1.4.1 FSTBSE

The FSTBSE User Interface will have read/write access to all the SQL tables that pertain to viewing, modifying, and creating a simulation scenario within a batch table. Railinc will implement functionality outlined within the ConOps for the Batch Table View/Editor to save the necessary data to the SQL tables so TCL will continue to be able to generate TOES simulation files.

4.1.4.2 Existing System

Functionality pertaining to viewing, modifying, or creating batch tables through the TCL batch editor will be removed. TCL will retain read access to the SQL tables for the purpose of generating TOES simulation files and will also have write access to the SQL database for recording the summary simulation information as files are generated. SQL connections that remain within TCL will be updated to interface with the new open-source, object-oriented SQL database.

4.1.5 Simulation Job View/Editor

4.1.5.1 FSTBSE

The FSTBSE user interface will have read/write access to the SQL table that currently stores all the simulation behavior setup data for simulations enabled to run. Additional SQL tables may be created to support the creation of simulation jobs. Railinc will implement functionality outlined within the ConOps for the Simulation Job View/Editor and to save the necessary data to the SQL tables. A process for providing data to TCL through the Simulation Controller/Manager will be developed so TCL will be able to run the simulations specified by the Simulation Controller/Manager with the desired simulation behaviors.

4.1.5.2 Existing System

Functionality pertaining to the setup of simulation behavior will be removed from TCL. The capability of starting simulations directly through TCL will also be removed. Railinc will determine how TCL will be controlled through the Simulation Controller/Manager. TCL will need write access to the simulation results table within the SQL database to record data as simulations are being run. TCL will be modified to support this process and will update the SQL connections that remain within TCL to interface with the new open-source, object-oriented SQL database.

4.1.6 Data Output View

4.1.6.1 FSTBSE

The FSTBSE user interface will have read/write access to SQL tables set up for the data output view. Railinc will implement functionality outlined within the ConOps for the Data Output View and save the necessary data to the SQL tables and/or a user accessible storage location.

4.1.6.2 Existing System

TCL will not have any interaction with the Data Output View, but may need read access to the SQL tables storing the result data to support the data analysis required for the Data Output View.

4.2 FSTBSE Simulation Controller/Manager

The Simulation Controller/Manager is a new simulation environment component to be developed by Railinc. The desired functionality of the Simulation Controller/Manager is outlined within the ConOps. During development of the ConOps, it was determined that the Simulation Controller/Manager will monitor the SQL database to identify if a simulation job has been requested, and then the Simulation Manager/Controller will interact with the scalable simulation environment to start the desired number of simulation tool containers for the job. For each instance of simulation tools initiated, the Simulation Controller/Manager needs to pass information to TCL so it can correctly set up and run simulations. The information identified to date that must be provided to TCL during this process includes:

- Request to generate TOES train consist files
- Request to generate TOES simulation files
- Request to execute simulations, which will also require the following information:
  - Simulation scenario information
  - Access to TOES files
  - Desired simulation behavior
  - Desired output data

Once TCL initiates an instance of the simulation tools, it will use the information provided by the Simulation Controller/Manager for the requested job. This information could include requests to create TOES simulation files, simulation behavior setup, and simulation scenario information, which TCL will use to generate and run jobs in the same manner that it does in the current simulation environment. TCL will have access to the SQL database as required to create requested TOES files and will execute simulations by interacting directly with TOES and an EA as needed. TCL will be updated to operate as desired; however, further development, implementation, and testing of the interaction between the Simulation Controller/Manager and TCL will be required and will fall outside the scope of this phase of the project.

Railinc will also develop functionality outlined in the ConOps to monitor instances of the simulation tools as they are running and to gather required output data needed upon completion of a simulation instance.

4.3 FSTBSE Scalable Simulation Environment

FSTBSE's simulation environment will need to be scalable so it will have the capability to run many parallel instances of the same, or different, simulation tools with minimal, or no, manual setup of simulation instances. It was determined that a cloud-based solution, running in a Linux environment, was the desired approach for the simulation environment. The cloud-based structure will allow Railinc to develop the infrastructure so it can have access to simulation instances that can be run in parallel. Railinc also developed an approach that will use containers

in the Linux environment. The use of containers will allow Railinc to start multiple instances of the simulation tools without the limitations present in the current simulation environment, mainly without the need to manually set up and configure the simulation tools.

Using a Linux environment for the scalable simulation environment requires that the simulation tools operate in that Linux environment. To date, TOES and TCL have been configured to run only within a Windows environment. Outside of the scope of this project, an effort to compile and test the TOES simulation engine in Linux has been initiated. Further development, implementation, and testing of TOES and TCL working within the scalable simulation environment will be necessary, but these tasks fall outside the scope of this phase of the project. TCL and TOES will be made available to Railinc to be used within the scalable simulation environment.

5. TOES TFG

TOES allows users to create track files using a track builder program. This program allows a user to create a model for any section of track but requires manual entry of track characteristics such as curvature, elevation, and track length. As part of this project, MxV Rail documented, developed, and implemented a stand-alone program to create TOES track files from PTC track files that can be used by TOES and TCL. The program, called TOES TFG, automates the manual process of modeling surveyed PTC track files. It does this by allowing a user to import an existing PTC track data file and create a TOES track file for a single track within the PTC track data file to be used by TCL or TOES. TOES TFG will support on-demand simulations capability in the FSTBSE by allowing the user to easily create a TOES track file from a PTC track data file to simulate real-world conditions.

5.1 Development of the TOES TFG

The available data formats for PTC track data files were reviewed with the AG, and the XML format was chosen as the input file format type for the FSTBSE. XML files were selected due to their ease in human and machine reading without the need for additional security and encryption data required for other available file types, as well as their widespread use by the railroad industry. Three key areas of development were identified during the conception of the TOES TFG:

- Identifying the significant PTC track file data that would be required to create a TOES track file
- Importing and converting the PTC XML file data into TOES compatible files
- Identifying the key capabilities of the user interface

5.1.1 Identifying Significant PTC Track File Data

To begin the data evaluation process, several railroad members of the AG provided sample XML PTC track data files, which were then compared to the AAR *Manual of Standards and Recommended Practices* (MSRP) Standard S-9503 (Association of American Railroads, 2015) which specifies the format of PTC track data files. The PTC track data files and S-9503 were reviewed to identify the key data within subdivision files needed to generate a TOES track file. Additionally, during the review of the files, several issues were identified with respect to the railroad-provided XML PTC track data files and the S-9503 specifications. The following list includes the issues identified and the method selected for resolving the issues:

- PTC XML track files provided by the AG contained placeholder values or extra data that was not identified in S-9503.
    - Missing/extra data that is not required for track file creation will be disregarded.
    - Missing data that is required for track file creation will cause TOES TFG to display an error.
- Track data points are not consistently spaced.
    - The spacing between track data points in the provided files ranged from 6 feet to 5,000 feet. The track files will be created with the existing data. In the future, this

may need to be addressed if it causes issues due to excessive file size or difficulties in calculating grade or curvature.

- Single data points were identified as possible outliers.
  - These data points may be erroneous, but they will be included in the track files. Users will have the opportunity to review the files after they are created to determine the validity of all data.

5.1.2 Importing and Converting PTC Track Data Files

Once the XML PTC track files were evaluated, work began to automatically extract the necessary information from each PTC track data file and develop a process for creating a TOES track file from the PTC file. The main steps identified for the process are:

- Read in and populate a list of PTC tracks into TOES TFG user interface
- Read in and populate the data for track footage
- Read in and convert PTC elevation data into TOES elevation inputs
- Read in and convert PTC heading and location data into TOES curve inputs

The TOES TFG program requires, at minimum, the track length, elevation, and curvature to create a valid track file. Once the data was identified and a process established to convert the information into a useable format for TOES, work began to develop an intuitive user interface.

5.1.3 Key User Interactions

The TOES TFG automates the creation of TOES track files from XML PTC track data files while retaining the key user capabilities provided by the existing TOES track file builder. These key capabilities enable the user to review, modify, and save TOES track files and are described in further detail in Table 1.

Table 1. Key User Interactions

Feature	Description
Provided ability to add/edit/remove inputs and update a TOES track	Allows the user to review and modify a generated track file. Also allows a track to be created from scratch (existing TOES functionality), if necessary.
Implemented dialog for initial values input to obtain initial position, speed, and elevation	The only initial value requiring manual entry will be speed. The initial position and elevation will both be read in from the PTC XML file. All values are editable.
Implemented change tracking to notify when updating/saving is needed	Changes to the input values in TOES TFG require a new TOES track file to be generated for the values to be included in the output track file. This will remind users that a TOES track file needs to be saved before generating a new track file, updating a track file, and/or exiting the application.

5.2 TOES TFG

TOES TFG creates a track file using XML PTC track data from the processes and user interactions described in Section 5.1. Figure 5 details the process flow of the program. Figure 6

through Figure 10 are screenshots from the TOES TFG program that illustrate the application of the processes described in Figure 5.

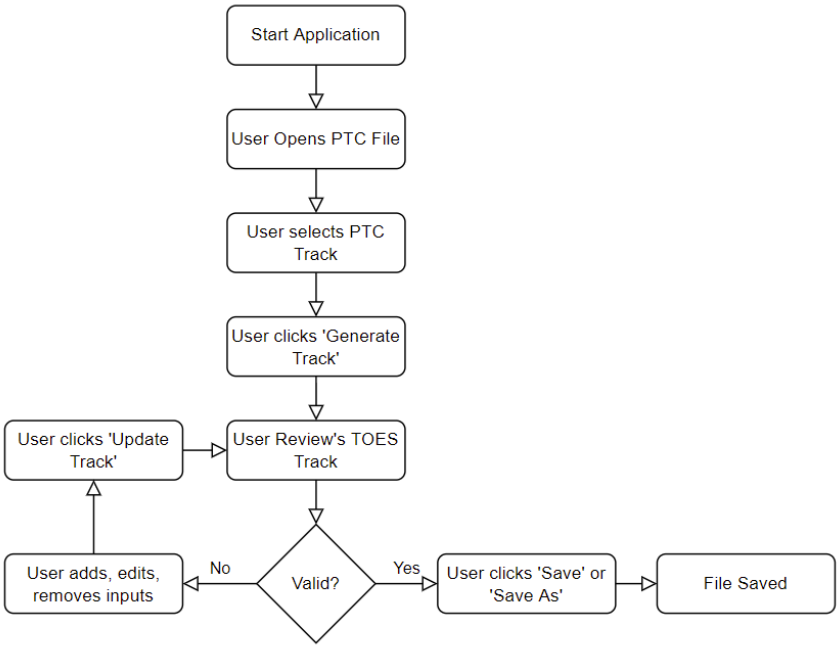


Figure 5. Track file creation process flow

Users initiate the program and then select a PTC track data file to open. Figure 6 shows the dashboard the users see when they have loaded the PTC track data file. Selecting a PTC track segment from the left side of the screen will then prompt the user to enter an initial position, elevation, and speed, as shown in Figure 7. The default values are set to zero for location and elevation, but a new value may be entered according to user preference.

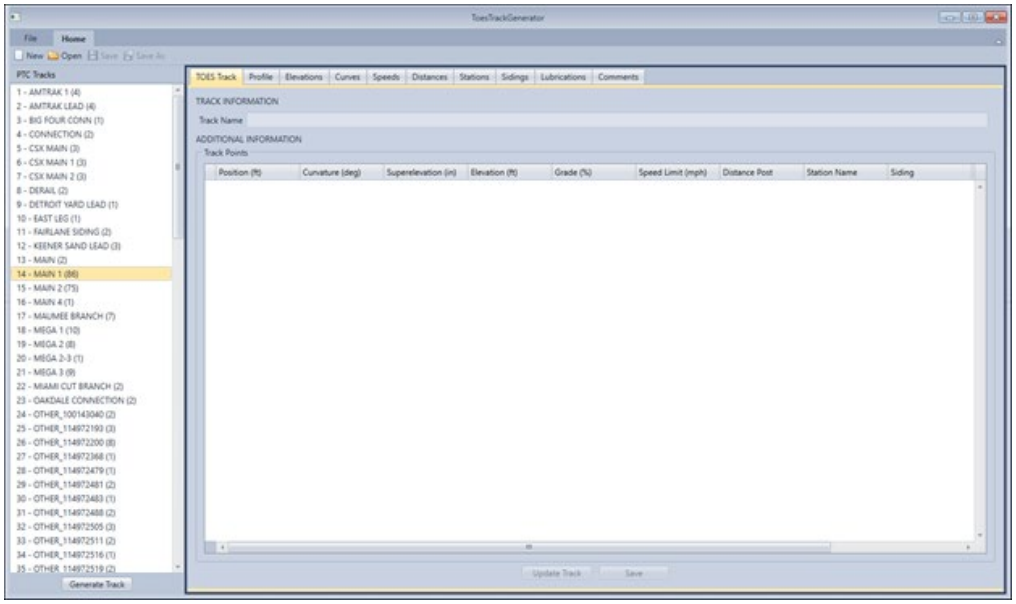


Figure 6. Selection screen

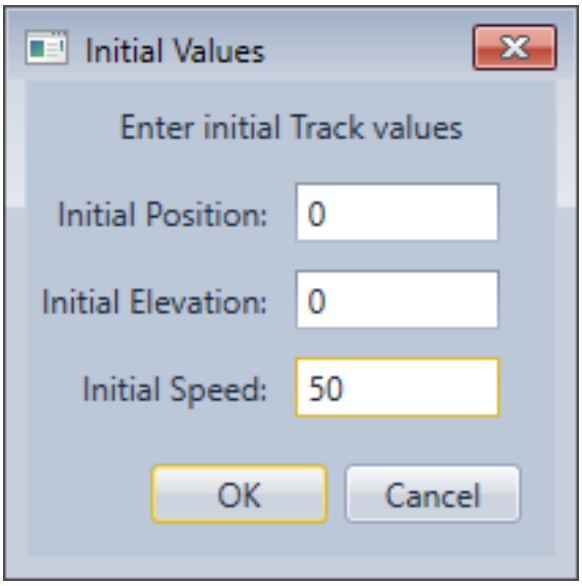


Figure 7. Initial track values screen

After the PTC track is selected, the program populates the following tabs with data from that track: TOES Track, Profile, Elevation, and Curves. The XML files contain additional data, but these are the only values that TOES requires to create a track. Speeds, distances, stations, sidings, lubrication conditions, and comments are user-entered information that can be stored in the track file for reference.

In Figure 8, the track profile is displayed in graphical format to allow users to visually inspect the track elevation and curvature imported from the PTC track data file. The numerical values corresponding to the elevation display are shown on the Elevation tab, as shown in Figure 9.

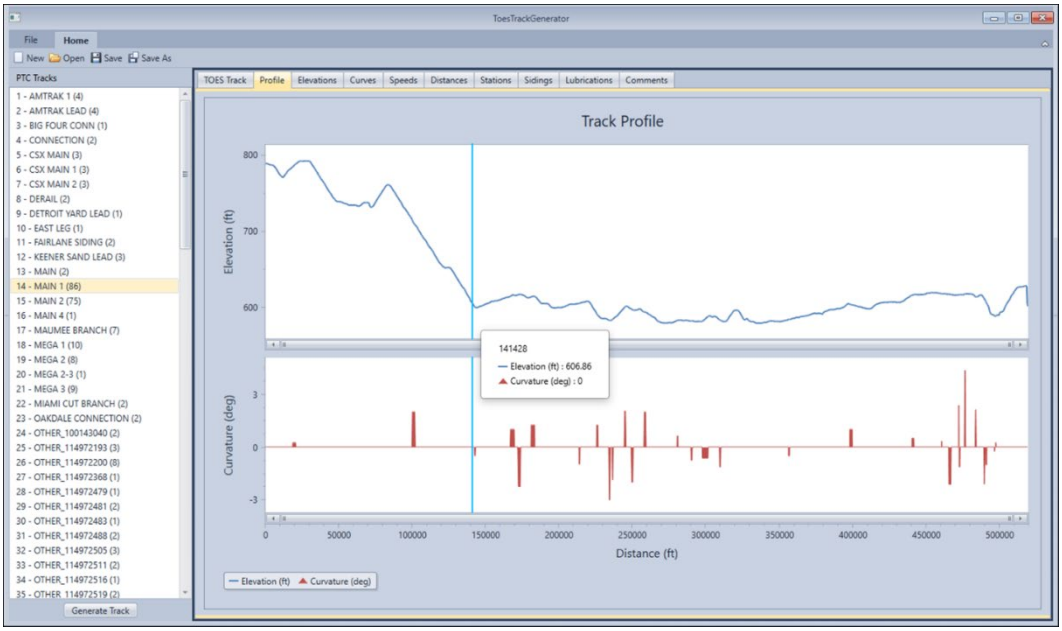


Figure 8. Track profile screen

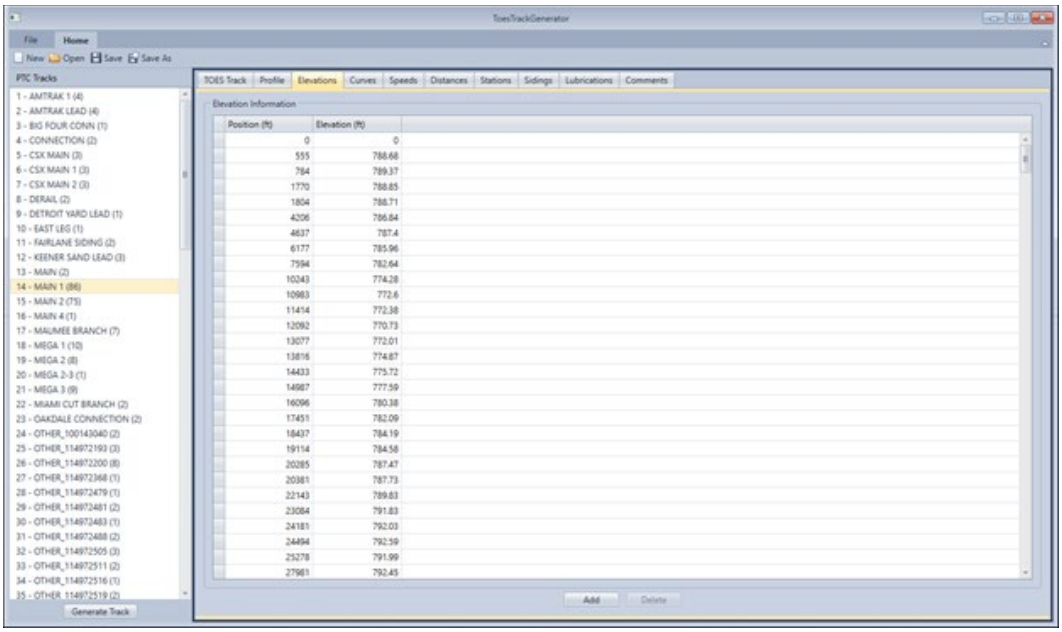


Figure 9. Elevation display screen

The user manually enters speed values to indicate the track speed and any speed restrictions. In this initial version, speed from the track data file is not directly imported. Speeds included in the track data files may contain multiple values using train type; however, speed entries are not currently used by TOES during simulations and are included in the TOES track file for information only. Figure 10 shows a sample data point.

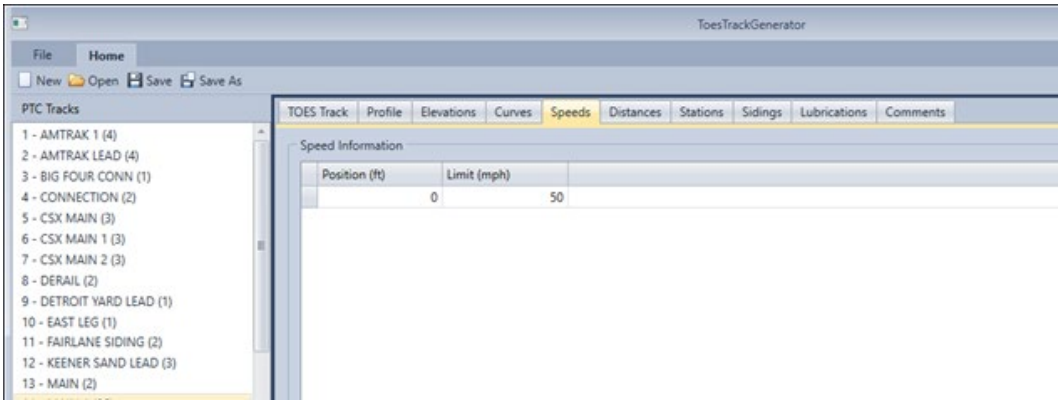


Figure 10. Speed display screen

5.3 Future Capabilities

The current version of the TOES TFG is limited to supporting creation of a TOES track file for a single track within a single PTC track data file. Potential future enhancements could include the following:

- Enable users to create a route that includes several tracks within the track data file, stitched together to create a seamless TOES track file.

- Enable users to create a single TOES track file for a route involving several tracks in any given PTC track data file. Users would be able to simulate an entire trip using these extended track files.
- Update the program to include more information from the PTC track data file, such as the locations of stations, grade crossings, sidings, lubrication, and track speeds. These values are not required to run a track file in TOES and as such, were assigned a lower priority for implementation in the TOES TFG.

6. TCL to EA Documentation

To support the methodology and the initial environment developed for PTC EA evaluations, an interface specification document was created by MxV Rail as part of a prior effort to define the communications between TCL and the EA (FRA, 2021). The TCL-EA interface specification document allows testing of any EA in the simulation environment if it adheres to the interface specification.

There are three messages defined in the interface specification document: an initialization message, an EA status message, and a train data message. The initialization message allows TCL to set up a simulation with the EA by providing the following information:

- Track file
- Train starting location and speed
- Target location and speed
- PTC summary train consist information

In the current simulation environment, the track file provided in the initialization message is the name of the TOES track file that aligns with a version of track data pre-loaded within the EA software.

The PTC summary train consist information provided to the EA in the initialization message includes the information necessary for the simulation received by the PTC onboard system during the initialization process; however, it does not use the same format as the industry standard PTC back office to onboard interface.

The EA status message is created by the EA and provides TCL information for the next time step to do one of the following: apply a penalty or emergency brake and continue to the next time step; continue to the next time step without a penalty or emergency brake application; or terminate the current simulation.

The train status message is created by TCL and sent to the EA every time step. Message data contained in the train status message is similar to the data that the PTC onboard system receives from the locomotive when operating in revenue service.

6.1 Potential Future Changes

Depending on the implementation of the new simulation environment, changes to the TCL-EA interface specification may be needed to allow the EA to communicate correctly within the new simulation environment. Initially, the current interface specification document is planned to be used in TCL-EA communications, but changes may be required in the future. The following sections describe these changes.

6.1.1 Track Data Files

Currently, all track files used for simulations with EAs are preloaded on the EA so it can use the track file specified in the initialization message. When running simulations with these files, the location of the train sent from TCL to the EA is a TOES footage location, which the EA can process to determine the train location. The new on-demand interface will allow users to pick

their own track files and the EA they wish to evaluate. If the track file selected or created by the user does not currently exist in the EA software, a process will need to be developed so TCL and the EA can initialize a simulation on the track and ensure the location information provided by TCL in the train status message aligns with the EA track data. Two potential implementation approaches for this process include:

- Develop a process to allow the user to select PTC track file data to be loaded onto the EA and create a corresponding TOES track file. This would require an update of the initialization and train status messages within the existing interface specification so TCL can provide the needed information for initialization and train location in a format compatible with a PTC track data file.
- Develop a process to dynamically load TOES track files onto the EA in a format readable by the EA and use the existing interface specification between TCL and the EA.

Any changes or updates to support new functionality to the existing message specification document will be defined and documented in conjunction with the EA vendors.

6.1.2 TCL-EA Communications

The EA and TCL currently communicate across the Windows VMs using TCP/IP. The TCP/IP communication protocol is planned to continue to support communications between TCL and the EA within the Linux environment of the FSTBSE; however, if any modifications are needed to the TCL-EA interface to facilitate communications using TCP/IP, these changes will be documented and provided to EA suppliers in an updated version of the TCL-EA interface specification document.

7. Conclusion

The objective of this project was to develop the FSTBSE concept, infrastructure design, and initial development by accomplishing the following tasks:

- Identify and develop use cases for running simulations in a scalable virtualized simulation environment.
- Develop a ConOps for the FSTBSE from the use cases.
- Document an infrastructure design that allows the FSTBSE to mimic the functionality of the current simulation environment and support the expanded capabilities of the FSTBSE identified in the ConOps.
- Support Railinc during the initial development and implementation of tasks for the FSTBSE.
- Develop a program to create TOES track files from existing PTC track files to support on-demand simulations.
- Document the possible changes needed to the TCL-EA interface specification document to support capabilities identified in the use cases and ConOps.

The FSTBSE will retain the core functionality of the current simulation environment while improving efficiency and allowing users to create on-demand simulations. On-demand simulations will allow railroads to create and run simulations using specific operational conditions. Initially, the FSTBSE will allow users on-demand functionality by accessing the FSTBSE through a web interface to create the desired simulation, but future phases will allow users to directly create and execute simulations through an API.

During identification and documentation of the infrastructure design, MxV Rail and Railinc identified the elements of the infrastructure and software functions required to support the concept. During this process, it was determined that the desired approach for the simulation environment is a cloud-based solution running in a Linux environment. This cloud-based design will allow Railinc to develop the infrastructure necessary to provide a scalable environment.

A program was developed to create TOES track files from existing PTC track files to support on-demand simulations. This program was developed using the existing TOES track file creation program and expanded upon its functionality to support automated conversion of PTC track data files to a format useable in the TOES simulation environment.

The ConOps created in this project will be used to guide the implementation of the FSTBSE as the project continues through the next phases. Ongoing development will be required for future phases of this work to implement and deploy the FSTBSE. Phase II of the project is planned to provide continued support of the initial implementation and deployment of the FSTBSE, including testing of its functionality. Phase III of the project is planned to support implementation of an API that will allow direct setup and execution of simulations from external systems, e.g., railroad back office systems.

8. References

Association of American Railroads. (2015). *AAR Manual of Standards and Recommended Practices Section K-VI S-9503 "Interoperable Elecronic Train Management Systems (I-ETMS®) Subdivision File."* AAR.

Brosseau, J., Moore Ede, B., Pate, S., Wiley, R. & Drapa, J. (2013). *Development of an Operationally Efficient PTC Braking Enforcement Algorithm for Freight* (Report No. DOT/FRA/ORD-13/34). FRA.

FRA (2021). *Positive Train Control Passenger Braking Algorithm Enhancement – Phase II* Fully Scalable Train Braking Simulation Environment (Report No. RR 22-12). FRA.

Pate, S., Anaya, R., & Holcomb, M. (2019). *PTC Braking Algorithm Evaluation Methodology Enhancement* (Report No. DOT/FRA/ORD-20/27). FRA.

Appendix A.
Fully Scalable Train Braking Simulation Environment Concept of Operations

Revision History

Date	Revision	Description	Author
03/16/2022	0.1	Initial Draft	Shad Pate, Rachel Anaya
4/18/2022	0.2	Updated Draft	Shad Pate, Rachel Anaya
7/1/2022	1.0	Version 1	Shad Pate, Rachel Anaya

A.1 Scope

This document provides a conceptual description for a Fully Scalable Train Braking Simulation Environment (FSTBSE) that provides the capability to dynamically allocate simulation resources, as needed, to meet the railroad industry's freight train braking simulation needs and to support increased safety and efficiency of train control applications. This scalable simulation environment is intended to be used in a variety of capacities, ranging from supporting near real-time optimization of train braking predictions in train control applications to the more traditional use of evaluating enforcement algorithms (EA) of the Interoperable Train Control Positive Train Control (ITC PTC) system through Monte Carlo simulations.

This Concept of Operations (ConOps) documents industry consensus on the uses, interactions with other systems/components, and capabilities of the scalable virtualized simulation environment. The ConOps identifies needed changes to the current setup, execution, and analysis of simulations used to support verification of PTC braking algorithms and to model train operations.

A.1.1 Document Overview

- Section A.1 introduces and describes the scope of the FSTBSE.
- Section A.2 provides background information on the current system.
- Section A.3 provides justification for the development of the FSTBSE.
- Section A.4 provides details of the proposed FSTBSE.
- Section A.5 describes the FSTBSE capabilities and operating scenarios.
- Section A.6 presents the expected impacts from developing the FSTBSE.
- Section A.7 summarizes the expected improvements gained from the proposed system as well as its limitations. It also describes alternatives and trade-offs considered.
- Section A.8 provides use cases for the current simulation environment.

A.1.2 System Overview

The FSTBSE is a web-based tool used to simulate the behavior of freight trains in real-world scenarios in an efficient and cost-effective method. It allows the user to examine results over a multitude of operational and environmental variables without requiring field testing of every possible variation. The simulations can be run with an EA to provide safety and operational performance data for the EA that is used by the industry to support validation of the algorithms prior to deploying the algorithms in operation.

A.2 Current Simulation Environment

A.2.1 Background

Through previous Federal Railroad Administration (FRA) funded research projects (e.g., *Development of an Operationally Efficient PTC Braking Enforcement Algorithm for Freight Trains [1]* and *PTC Braking Algorithm Evaluation Methodology Enhancement [2]*) and other algorithm enhancement projects, Transportation Technology Center, Inc. (TTCI) and the railroad industry developed a methodology for evaluating PTC braking EAs that depends on the use of simulations. This evaluation methodology produces statistical results over a broad range of different train consist, speed, and grade combinations. It is used to produce safety and performance metrics for the PTC braking EA that the industry can use to support the use of the braking EA in PTC [1].

During these research projects, the team developed a simulation environment to support the PTC braking EA evaluation methodology. These efforts included the development of the Test Controller and Logger (TCL) software, which facilitates the setup and creation of simulations and the communication between PTC braking EAs and the simulation model during the execution of simulations. Figure A-1 shows the main simulation components in the current simulation environment.

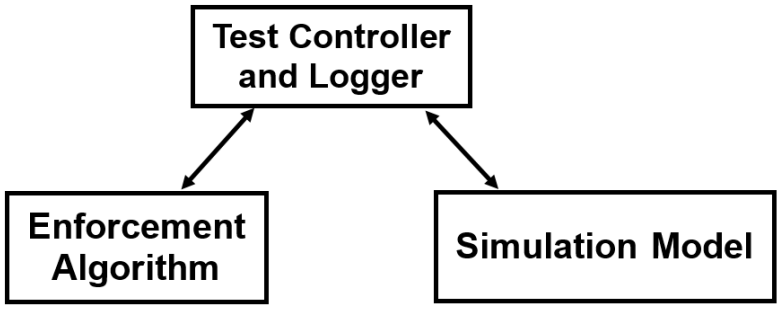


Figure A-1. Simulation Testing Components

To reduce the amount of time to complete a set of simulations, the simulation environment is scalable, meaning parallel instances of the model in Figure A-1 can be set up and configured to run simulations simultaneously.

In addition to the evaluation of PTC braking EAs, this simulation environment can and has supported the industry with other braking EA-related simulations. These simulations have included, but are not limited to, the following:

- Simulation modeling of field test data
- Targeted simulations to support detailed evaluation of the specific functionality of a PTC braking EA
- Simulations performed before and after a change to PTC braking EAs to evaluate the impact of the change on braking EA performance
- Specialty train types or trains not currently part of the PTC braking EA evaluation simulation matrix

The EA evaluation process and specifications for the communication between the EA and TCL is detailed in this document. Use cases for the current simulation environment are provided in Section A.8.

A.2.2 Operational Policies and Constraints

The scalability of the current simulation environment is subject to the physical limitations of available hardware, as well as the manual setup and configuration of each instance that must be repeated each time a testing component is updated.

PTC braking EA vendors need to provide the team with a version of their EA that supports the TCL-EA interface specification document [3] so simulations can be run using the EA in this simulation environment.

A.2.3 Description of the Current System

Section A.2.3.1 provides a description of the simulation testing tools as documented in previous FRA research reports [1,2].

A.2.3.1 Simulation Testing Tools

The simulation testing portion of the EA evaluation methodology requires the following three components, as illustrated in Figure A-1:

- A proven, validated train action simulation model that accurately models the response of a given train under given conditions, with the ability to modify train, track, and environmental characteristics that can affect the stopping distance of the train
- A TCL software application that can generate the simulation inputs to the model from input provided by the user, run large batches of simulations using Monte Carlo simulation techniques, and log the required output
- The EA under evaluation, implemented as a standalone software application incorporating a common interface to the simulation test components to receive train status and command brake enforcement applications

A.2.3.1.1 Simulation Model – Train Operations and Energy Simulator (TOES™)

To model any given braking enforcement scenario, the chosen simulation model must accurately depict the response of the train to given inputs, be capable of modeling the specific characteristics of each component of each railcar within the train and the specific characteristics of the track, and be capable of reporting train status data at regular, frequent intervals. TOES is a longitudinal train dynamics model developed by the Association of American Railroads (AAR) that models the status of every railcar in a given train at every time step of the simulation. Railcar status data includes location, velocity, acceleration, forces acting on the railcar, and brake system component status.

The TOES model allows the user to enter specific characteristics for each railcar in the train, including railcar weights and dimensions, aerodynamic properties, truck characteristics, coupler and draft gear characteristics, and brake system components and characteristics. This flexibility allows the user to model any currently used freight railcars and arrange them into any train consist desired. The model allows the user to enter track characteristics that affect the

longitudinal motion of the train (i.e., track grade and curve) allowing any section of track to be modeled. Finally, the model allows the user to enter environmental conditions that can affect the longitudinal motion of the train, such as ambient temperature and the coefficient of friction between the wheels and brake shoes. The TOES model allows the user to enter train handling commands, such as throttle and brake settings, at any time step in the simulation to model how the train reacts to these commands.

The components that make up the TOES model are some of the most accurate and proven models currently available to the railroad industry. These include a variety of draft gear models, multiplatform railcars, an aerodynamic drag routine, and a variety of user-customizable railcar components. TOES includes a theoretical fluid dynamics model of the air brake system, which has proven to be a significant improvement over similar models empirically derived from test data. The air brake model within TOES can simulate the automatic and independent air brakes, a range of brake valve and brake shoe types, any length of brake pipe, brake cylinder dimensions, and reservoir volumes.

A.2.3.1.2 Test Controller and Logger (TCL)

A custom software application was necessary to manage the large number of simulations required to generate the necessary statistical significance for the safety and performance of the EA over the entire range of potential operating scenarios. To support the industry in the development and testing of a safe and operationally efficient braking EA, the team developed a TCL software application capable of generating and executing thousands of braking enforcement Monte Carlo simulations that uses operating scenarios and parameter variation distributions entered by the user.

The TCL application performs the following three major functions:

- Generation of random simulation inputs
- Execution of individual simulations
- Logging of output data

Simulation input data can be generated by setting up a batch of test scenarios for evaluation. The user selects a train consist and track profile and enters initial train speed and location and target stopping location for each test scenario in the batch.

The user defines train consists by selecting the desired railcars and arranging them in the preferred order. Each railcar is defined by the nominal components and characteristics of the railcar and the potential variation of these components and characteristics (also defined by the user). The variation of the railcar components and characteristics are represented by a variety of distributions, allowing the user to define the variability of a given parameter to match its actual, real-world variation. The user defines the potential variation of environmental characteristics and variation caused by errors in reported data, such as track characteristics, train speed, and location.

The user selects the number of simulations the TCL software will run for each test scenario in the Monte Carlo process. The TCL software then generates the simulation input data for each simulation within each test scenario, by randomly selecting values for the variable parameters using the input distributions defined by the user.

Once the simulation input data is generated, the user runs the batch through the TCL software. The TCL application runs each simulation for each test scenario individually in the simulation model by advancing the train toward the target at the given speed. At each second of simulation time, the simulation model reports train status data to the TCL, which transmits it to the EA. When the EA predicts an impending target overrun, it sends a command to initiate a penalty brake enforcement to the TCL application, which then executes the penalty in the simulation model. The TCL continues to advance the simulation until the train is stopped. The EA can send a command to initiate an emergency brake enforcement, which TCL then executes in the simulation model.

Once the train has stopped, the simulation is complete, and the TCL software logs the output data in a database for post-process analysis.

A.2.3.1.3 EAs

The EA evaluation methodology can be applied to evaluate any EA for any freight PTC implementation. The methodology treats the software implementation of the EA as a black box that communicates with the simulation testing components over an open communications interface. A document detailing the communications process and protocols was prepared for use by developers of EA software to be evaluated using the methodology.

To allow for the most flexibility in the test setup, the interface was designed with communications over transmission control protocol/internet protocol (TCP/IP). This allows for the EA to be implemented as an executable software application running on the same machine as the TCL software as a virtual machine (VM) with a separate IP address, but it operates on the same hardware as the TCL software, or as software running on separate hardware that communicates over TCP/IP.

The interface was designed with flexibility for initializing the simulation test process to allow for more efficient execution of the simulations. The TCL software can execute the EA software directly if it is run on the same machine. Alternatively, an EA initialization module was developed that sends an initialization message to the EA software indicating that the previous simulation is complete and the new simulation is beginning. This allows the EA software to re-initialize internal parameters and other functions for the new simulation.

To ease the integration of an untested EA with the TCL software setup, a protocol test application was developed. The protocol test application replicates the communications to and from the TCL software with the current protocols, but without the additional functionality of the TCL software. This allows the EA software developer to test its communications interface and debug any issues locally, resulting in reduced time and cost associated with the integration process. The source code for the protocol test application is available to support the development of the interface on the EA side.

A.2.3.2 Simulation Setup and Execution

Simulation setup is done using TCL and available data within a Structured Query Language (SQL) database. The SQL database includes data for modeled TOES vehicles and TOES track files, previously created train consist files, and previously created batches. TCL can be used to create new train consist files using modeled vehicles stored in the SQL database. The new train consist file is saved to the database and can be used when setting up new simulations. If the train

consist being created needs a new vehicle modeled, then data for that vehicle is manually loaded into various SQL tables so it will be selectable when creating the train consist.

A simulation scenario is created within a batch, which can be comprised of a single simulation scenario or multiple different scenarios. A simulation scenario is made up of four key components: a train consist, a track profile, initial conditions, and target information. Currently, batches and simulation scenarios are created using TCL. When adding a simulation scenario to a batch, the user can select a train consist and track profile from available train consists and tracks stored in the SQL database as well as enter the desired initial conditions and target information. If the simulation scenario will use a new train consist or track profile, these must be created and stored in the SQL database before creating the simulation scenario. As described, the train consist can be created using TCL; however, adding a new track profile currently requires manual creation outside of TCL, and the SQL database must be updated so the new track file is selectable when creating a simulation scenario.

Once a batch is created, it can be selected for execution through TCL. The user can set some simulation level parameters prior to the start of simulations, such as generation of TOES train consist and simulation files, emergency brake backup configuration and behavior, dynamic brake (DB) behavior, locomotive brake behavior, etc. TCL will configure the machine using the simulation preferences and will begin executing the simulations by stepping through each scenario defined within the batch. Figure A-2 illustrates the organization of simulations within a batch.

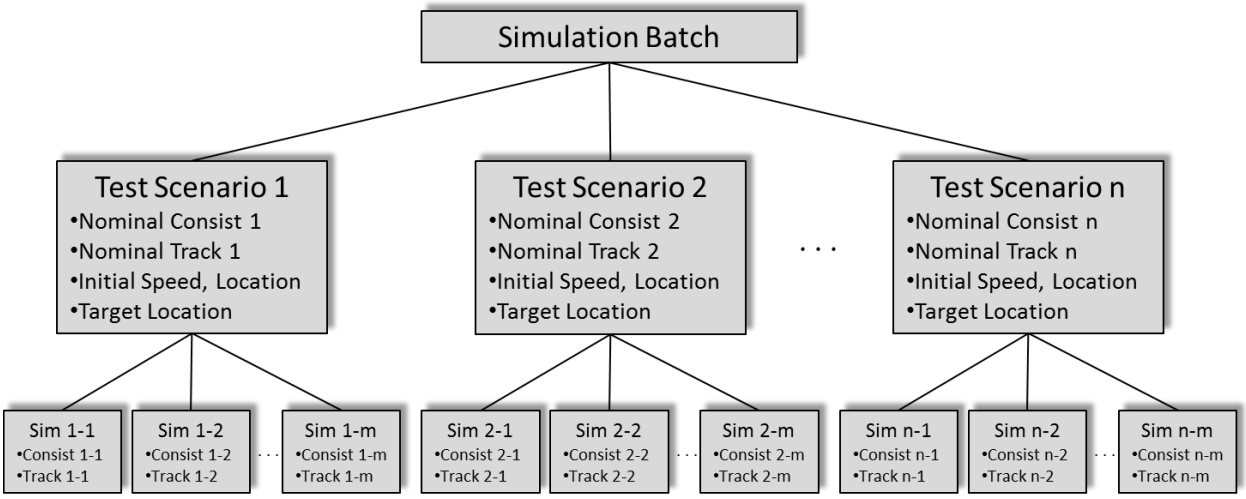


Figure A-2. Organization of Simulations

TCL will run each simulation to completion and write a simulation results record to the SQL database, which is then used to analyze the simulation results. The scenarios and simulations shown in Figure A-2 can be expanded to show Simulation Batch 1 to Simulation Batch n, where each simulation batch has different test scenarios and simulations setups. Multiple instances of the setup shown in Figure A-1 can be configured, with each instance capable of setting up and running a different simulation batch.

Typically, the setup of these simulations is a manual process and can require additional data from the railroads, additional modeling within the simulation environment, and/or modifications to the EA undergoing testing.

A.2.3.3 Data Analysis

Upon completion of the simulation batch, raw data is available for analysis. The data is processed to provide a "big picture" look that is broken down by train type. For industry-driven simulations, freight trains are grouped into Unit, Intermodal, and Manifest train types. Unit trains are either fully loaded or fully empty train consists made up of a single vehicle type, with multiple locomotives. Intermodal trains contain intermodal vehicles with locomotives, and may include a mix of single platform, three-platform, or five-platform vehicles. Manifest trains are comprised of a variety of freight cars seen in revenue operations. These trains can contain as few as one locomotive and up to several locomotives and 200 railcars. To provide a sampling of real-world operations, each freight train type contains different locomotive power configurations, depending on the length and weight of the train consist being modeled.

The overall results include the percentage of simulations that stopped at or before the target, the percentage that overran the target, and the probability for not meeting the performance target. The safety objective, as stipulated by FRA, states that 99.5 percent of the train consists stop at or before the target. Updates to the EA can affect the calculations used to determine the braking curves for a setup; therefore, the simulations are rerun any time a significant change is introduced to the EAs. While any simulation that stops past the target is considered a failure, the distance beyond the target may indicate extenuating circumstances that should be examined more closely to determine why the simulation overran the target. The performance objective is often referred to as "Undershoot" and is defined as stopping more than 500 feet before the target when the initial train speed is under 30 mph or stopping more than 1,200 feet before the target when the initial train speed is at or above 30 mph. Undershoot distances and probability are not necessary for safety analysis, but they are of great interest to railroads since they may impact the productivity and operations on track.

Additional analysis can be performed by grouping the simulation results by train consist, track profile, and/or initial speed. These individual results demonstrate a more comprehensive picture of the factors leading to overruns or undershoots and allow for troubleshooting on any train consists that may have provided unexpected values. The results include the percentage of simulations in which the EA requested an emergency brake application, and the location where the enforcement was applied.

Results from these simulations can be used and shared by the railroads to support their cases for using the PTC braking EA.

A.2.4 Modes of Operation for the Current System

TCL can run either EA evaluation simulations or TOES stopping distance simulations. EA evaluation simulations require TCL to interface with the EA and TOES to convey commands and status updates that drive the simulation. Commands, simulation status, and output data are recorded by TCL and used during the evaluation of the EA. A TOES stopping distance simulation runs with TCL initializing a TOES simulation and issuing a brake command, then stopping distance output data is recorded and the simulation is ended.

TCL can run both EA evaluation of TOES stopping distance simulations in either automated mode or manual mode. For automated mode, TCL periodically queries the SQL database to see if any batches are set up and configured to execute. If a batch is available to run, an instance of TCL, running in automated mode, will select that batch and configure the simulation parameters

to the settings defined for that batch in the SQL database. TCL will execute the scenarios and simulations within the batch and mark it complete when finished. TCL will then begin querying the database to see if another batch is available and repeat the process as necessary.

For manual mode, the user accesses an instance of TCL, sets the simulation parameters as desired, selects the desired batch, and starts the simulations. TCL executes the scenarios and simulations within the batch and then sits idle in manual mode until the user starts another batch.

Whether in automated mode or manual mode, TCL has built-in error handling that will attempt to restart simulations that have encountered an issue. TCL will document issues that cannot be addressed in the error handling log and attempt to move on to the next simulation or scenario. For issues that persist across multiple simulation scenarios, TCL will error out of the simulations and notify users on an email distribution list.

A.2.5 User Classes and Other Involved Personnel

A.2.5.1 Users

TTCI personnel are the only users of the current simulation environment and the activities they perform can be categorized into the following user classes:

- System Administrator – users that perform administrative functions such as modifying the database, updating the simulation environment, creating new simulation instances, and running simulations
- Standard User – users that interact with the simulation environment to build and run simulations

A.2.5.2 Involved Personnel

Railroads, both individually and in joint industry efforts, are other involved parties that currently drive the use of the current simulation environment.

EA vendors are involved in the simulation environment by providing a version of their software to operate within the environment based on the TCL-EA specification document [3].

A.2.6 Support Environment

The support environment was set up to house multiple instances of the simulation tools, access to a SQL database, and storage for the simulation files and logs. The support environment has evolved over the life of the simulation environment.

The current simulation environment uses a VM setup to include up to 75 instances of the simulation tools, allowing for 75 different simulation batches to run concurrently. Each instance of the simulation tools is run on a one-to-one-to-one setup for TOES, TCL, and the EA. This configuration was deemed most efficient after initial simulation environments identified issues in sharing resources, mainly a TOES Fortran error due to sharing violations on the machine or a broadcast message error within EA(s) (this error signals that one or more of the EAs would be affected by a broadcast message sent by one EA and received by all EAs operating on the same machine).

The simulation environment contains a central data storage, where each simulation instance has access to the TOES simulation file storage. Use of a central data storage allows a previously created simulation to run from any simulation instance set up and connected to the central storage.

The current simulation environment includes an automated mode for TCL, which allows the user to set up simulations in the SQL database without having to access each VM running TCL individually. The current environment also requires the user to initially configure the simulation tools across each instance as well as update every instance when there is a change to any of the simulation tools.

A.3 Justification for Fully Scalable Simulation Environment

A.3.1 Justification of Changes

The current methodology for PTC braking EA evaluation simulates the braking EA under a static set of scenarios to verify that it performs acceptably over a broad set of operating conditions, using the established performance criteria, as described in Section A.2. This is illustrated on a conceptual level in Figure A-3.

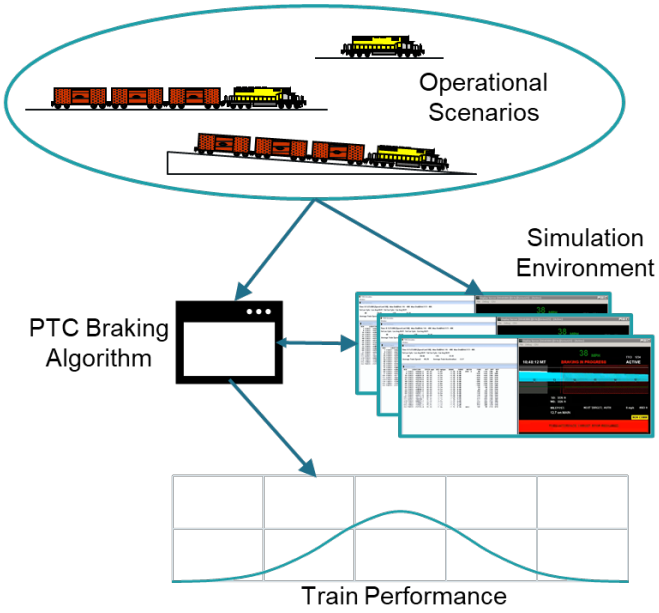


Figure A-3. Current Simulation Evaluation

This methodology is effective at demonstrating the performance of an existing PTC braking EA. However, it fails to take advantage of the opportunity to use the simulation tools and Monte Carlo simulation concept to provide input to the PTC braking EA (and other train control applications) to further optimize the overall operational performance. Figure A-4 shows a different concept for use of the simulation environment in which simulation scenarios are defined dynamically in near real-time using actual train consist and route information, and the simulated train braking performance is used in train control applications to optimize the safety and performance for each train.

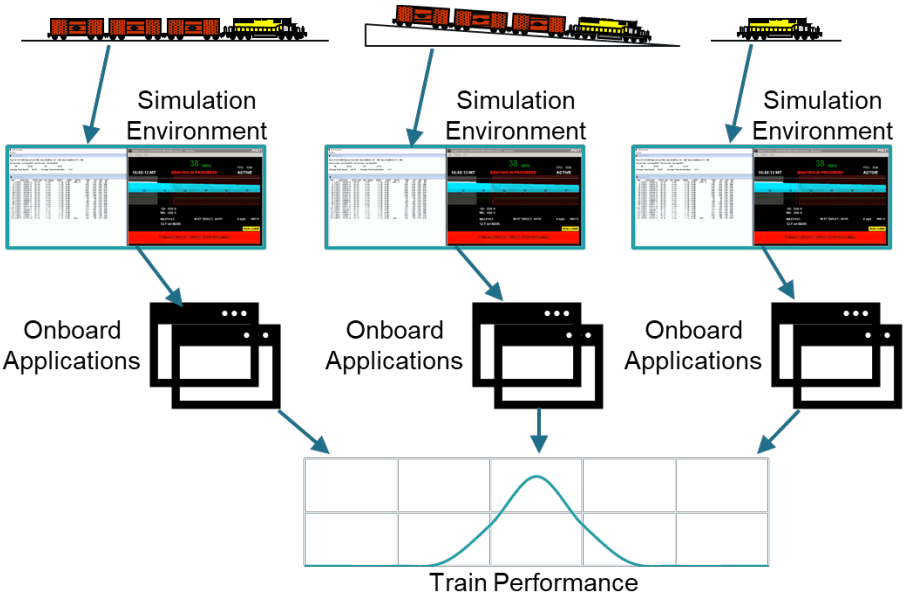


Figure A-4. Dynamic Simulation Evaluation

Generally, the existing PTC braking EA simulation environment has proven to be a valuable tool for the railroad industry in providing the capability to evaluate PTC braking EAs in an efficient and cost-effective manner. However, there are limitations to the scalability of the environment which prevent its use in a broader capacity, such as the real-time concept described. Additionally, there are opportunities to use the environment to support other freight train braking simulation use cases and to complete the current PTC braking EA simulation evaluation more efficiently with a dynamically scalable environment. These opportunities provide the justification for the changes to the environment described within this ConOps.

The objectives for the on-demand and fully scalable environment are to create a more efficient simulation environment, allow users to create on-demand simulations for modeling their own operations, and support a broader set of use cases. A fully scalable simulation environment will allow more simulations to be run in a shorter timeframe.

A.3.1.1 User Support Limitations of Current Environment

The development of the legacy simulation environment began in 2009, and it has evolved to meet the changes and advancements of PTC operations and PTC braking EAs. Features and capabilities have been added to the current environment as needed, and while these were implemented in the most effective method possible at the time, practical limits are now being reached.

A.3.1.1.1 Simulation Environment Limitations

The simulation tools shown in Figure A-1 are designed to set up a simulation instance. Each simulation instance uses one or two VMs including a Windows VM running a single instance of TOES and TCL. EAs are integrated with the simulation instance by either running the EA in the same Windows VM or by creating a separate VM to house the EA, and configuring the two VMs to work together.

Multiple simulation instances are set up by creating new VMs, with each simulation instance segregated from the others. Theoretically, the number of simulation instances is not limited, but the resources required to run each are limited by physical hardware and available personnel to set up the environments.

A.3.1.1.2 Setup Limitations

Initialization and setup of the simulation instances have been automated to some extent but still require each machine to be manually configured in the final steps. The manual tasks are small but can require significant support time because they must be repeated for each of up to 75 machines.

A.3.1.1.3 Operating Limitations

The legacy simulation environment can automatically start simulations and record the results after users manually configure a table in the SQL database. This table is broken down into batches of simulations, but with an increased number of simulation instances, the table could be further decimated to reduce the time required to run the entire set of simulations. Errors due to interrupted communications, hardware issues, simulation issues, or programming can be logged and skipped unless the errors persist. When errors persist, the user is required to interact with the VM to understand and correct the error.

A.3.1.2 Hardware Limitations

The legacy simulation environment is run on two physical servers that support both the freight and passenger PTC braking EA simulation efforts, and as such, are often running at maximum processing capacity.

Currently, the servers can support up to 75 simulation instances of the simulation tools for freight braking EA testing. Adding additional simulation instances or running freight simulation instances concurrently with those for passenger simulations degrades the computing resources across each simulation instance.

A.3.1.3 Scalability Limitations

The scalability of the legacy simulation environment is limited by the hardware and user support limitations described above. Additionally, the current environment cannot scale to support on-demand simulations or to manage a larger volume of simulations running simultaneously. As a result, the current environment is limited in its ability to support the broader freight train braking simulation use cases envisioned.

A.3.2 Description of Desired Changes

The new simulation environment should support the vision of a broader set of freight train braking simulation use cases, as described in Section A.3.1, including the capability to interact with other systems and components to set up and execute simulations, be fully scalable to meet a dynamic simulation demand, allow for more efficient execution, reduce the requirements for user interaction, and have an intuitive user interface.

A.3.2.1 Changes to the Current Environment

A.3.2.1.1 Simulation Environment Changes

The simulation environment will automatically manage and create simulation instances as needed to efficiently meet user demands and requests. The simulation environment will be scalable to address the hardware limitations of the current environment.

A.3.2.1.2 Setup Changes

A web-based interface will support the same functionality as the current simulation environment, with additional functionality for building simulations, creating user-defined train consists and track files, and updating inputs for vehicles.

A.3.2.1.3 Operating Changes

TCL will no longer be used for setting up and initiating simulations. Instead, an intuitive user interface will be needed to support the setup and creation of simulations with a separate process managing their execution in a fully scalable simulation environment. This environment must be able to interact with and operate TOES, TCL, and EA to execute simulations.

The user interface will replace some of the current functionality that TCL provides to give the user the ability to easily create user-defined, “on-demand” simulations. An on-demand simulation is one where the user can create or select from existing data items such as the train consist, track profile, train handling, initial conditions, and enforcement target information, as well as select the desired number of simulations and output data requested. The core functionality of TCL, i.e., creation of TOES files, communication with TOES and EA, and recording summary simulation results, will remain in TCL.

The desire for the system to be easily scalable and to store and call simulation instances on-demand has driven the desired architecture of the simulation environment to operate in a Linux environment, and this will require TOES, TCL, and the EA to operate in the same Linux environment.

Initially, users will manually access the system through a user interface directly in the new simulation environment. This environment will require the capacity to evolve to support integration with railroad back-office systems for automation of some or all of the tasks completed through the user interface.

A.3.2.2 Capabilities of a Fully Scalable Simulation Environment

The fully scalable simulation environment must be able to support increased efficiency, whether through speed of computations, increased instances of the simulation setup, or other undefined operations. The baseline operating speed used for comparison will be the current time required to run a full Monte Carlo simulation set in the current simulation environment across 75 simulation instances.

Simulations should run with minimal user interaction beyond entering the initial settings. This may require error checking and automated processes to resume or skip simulations. In the current environment these simulations could have stopped due to a communications error between

TOES, TCL, and the EA, powering off the machine incorrectly or unexpectedly, bad configurations of one or more of the inputs, or other unforeseen incidents.

The simulation environment will be able to interact with EAs. Vendors will have the specifications for and be provided support for issues related to operating in the Linux environment. EAs will continue to interface with TCL using the interface specifications [3], which may be updated to support new functionality within the scalable simulation environment.

Results will need to be available through either direct database interaction or a downloadable report. The reports must be configurable to provide the same results currently accessible upon completion of simulations. Raw data may be requested for further analysis and provided in a .csv or database file.

A.3.2.3 On-Demand Capabilities

Users will need to be able to create user-specified simulation scenarios like those described in Section A.8. The user will be able to create simulation scenarios by selecting desired train consists, track profiles, and train handling as well as providing initial conditions and target information. The system will need to store the user-defined data so TCL can access it to create and run simulations, as it does in the current simulation environment.

Notifications will be created between the system and users to inform users of pending actions or the completion of requested simulations.

A.3.3 Priorities Among Changes

The highest priority change is to create a simulation environment that will be fully scalable, while retaining all the current features and functionality of the legacy environment. A second priority is to improve the simulation process by decreasing the time required to run larger sets of simulations, while a third priority is to improve the user experience by creating an intuitive user interface that supports creation of on-demand simulations. A future priority is to expand the environment to support on-demand simulations created and executed by external applications.

A.3.4 Assumptions and Constraints

A fully scalable environment is primarily constrained by the following characteristics:

- TOES, TCL, and the EA must operate on a one-to-one-to-one relationship; each parallel instance created must be segregated from each other.
- TOES requires access to a storage location with the track file, train consist file, and initial conditions for each simulation. TCL requires access to the same storage location and requires access to a database.
- It is assumed that the scalable train braking simulation environment will be hosted on a Linux platform, based on initial design considerations for the concept.

A.3.4.1 TOES Constraints

TOES was built in Fortran for use in Windows. Due to the age of the language, compilers for Fortran are not widely available. Therefore, TOES needs to be compiled into Linux to work in the proposed simulation environment. Upon completion of the Linux compilation, testing will be

completed on both the Windows and Linux versions of TOES to show that the program has not been compromised or lost functionality.

A.3.4.2 TCL Constraints

TCL was developed for use in Windows and therefore will need to be compiled into Linux. TCL also uses Windows messaging to communicate with TOES and will require modifications to support messaging in Linux. Testing to verify required messages are being passed correctly after modifications are made will be required. Additionally, TCL currently uses a Microsoft SQL database to query for simulations ready to be run, to create simulations for TOES, and to store the results. It is assumed that a different SQL database will be used within the Linux environment, which will require updates to TCL. Subsequent testing of TCL will be needed to verify it functions as expected after the updates.

A.4 Fully Scalable Train Braking Simulation Environment Concept

A.4.1 Objective

A new train braking simulation environment is needed to support the railroad industry's ongoing and evolving needs associated with PTC braking EA simulation capabilities and to allow the railroad industry to independently set up and run train braking and operation simulations on an as-needed basis, either manually or using other applications. The FSTBSE will expand the capabilities and address the following limitations of the current simulation environment:

- Improve scalability of simulation instances by removing limitations imposed by physical host machines
- Decrease setup time required to configure a new instance of the simulation tools or update software across existing instances
- Reduce user interaction required by controlling and managing the creation, execution, and analysis of simulations automatically
- Allow intuitive user interfaces for creating and setting up simulations that will support manual creation of on-demand simulations as well future capabilities for integrating with railroad systems for automation of on-demand simulation requests

A.4.2 Scope

The FSTBSE will provide the same capabilities as the current simulation environment so it can supplement or replace the current simulation environment as needed, in addition to supporting on-demand simulation capabilities either manually through a user interface or using other applications through an application interface.

The scope of the simulation environment is limited to train braking simulation capabilities (including Monte Carlo simulation) available with TCL and TOES as they exist in the existing simulation environment. Therefore, no changes to simulation capabilities are currently in the scope of the project.

While the FSTBSE will be able to complete simulation jobs more efficiently across virtual resources, it is not intended to increase the speed of a single simulation, as doing so is bounded by the speed of communication and processing time between TCL, TOES, and the EA. Changes to these aspects of the environment are outside the scope of the FSTBSE.

A.4.3 Operational Policies and Constraints

The following policies and constraints influence the operation of the FSTBSE:

- The FSTBSE will only be available for use by authorized users, which may include personnel from or designated representatives of the research team, Railinc, and AAR member railroads.
- PTC brake EA vendors will need to provide a version of their software that conforms to the TCL-EA specification document [3], which is subject to updates for the new simulation environment. The vendor EA will need to operate in a Linux environment.

A.4.4 Description of the Proposed System

The FSTBSE has three key components:

1. User Interface – Users will access the FSTBSE through the user interface, which provides the following capabilities:
    a. Dashboard
    b. Consist View/Editor
    c. Track File View/Editor
    d. Train Handling File View/Editor
    e. Batch Table View/Editor
    f. Simulation Job View/Editor
    g. Data Output View
2. Simulation Controller/Manager – Controls the operations and functions of the FSTBSE. Users will not access the Simulation Controller/Manager directly, as it runs in the background to provide the following capabilities:
    a. Simulation Job Processing
    b. Simulation Job Monitoring
    c. Data Output Processing
    d. Error Handling
3. Scalable Simulation Environment – Provides the necessary infrastructure for the FSTBSE, including the following:
    a. Stored Containers
    b. Container Operations

A.4.4.1 User Access

Users will request access to the FSTBSE through a defined method and then access the system through a web-based user interface. System and user organization administrators will authorize users before they are granted access to the system. Users will have a secure process to login to the FSTBSE.

A.4.4.2 User Interface

FSTBSE users will manually access the system through a user interface. A dashboard will serve as the main page of the interface, and users will be able to access the different view/editor interfaces from this dashboard.

Figure A-5 is an overview of the FSTBSE user interface. The blue shading in the figure indicates the areas the user will be able to directly interface.

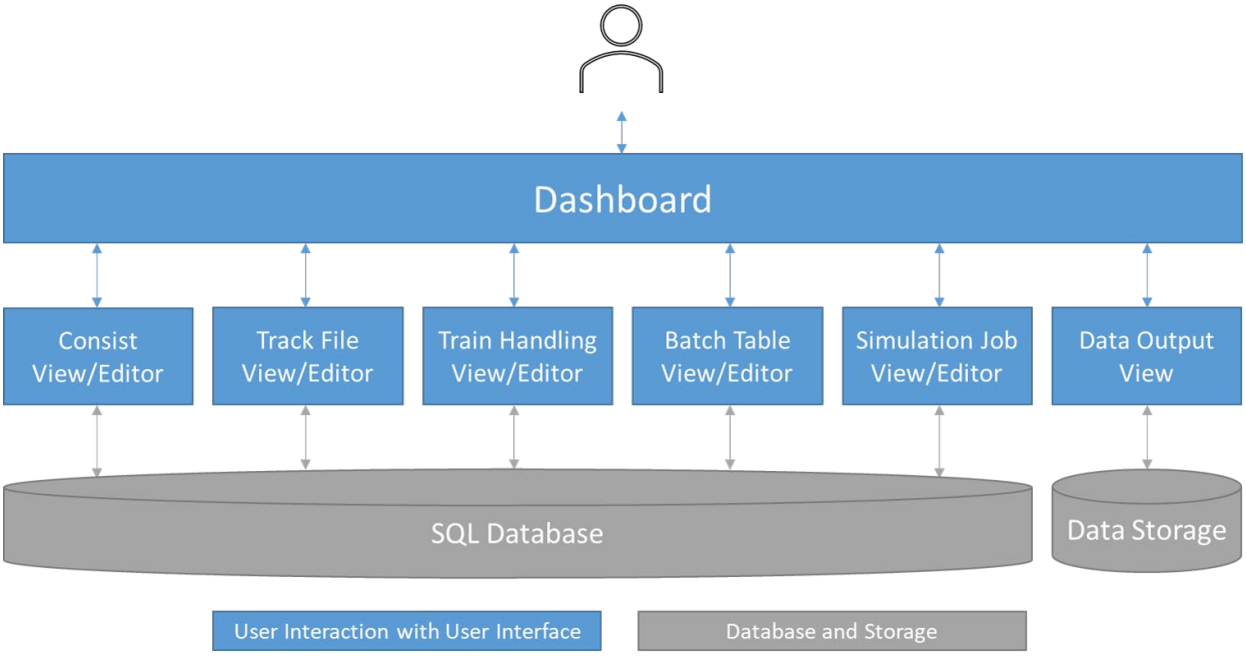


Figure A-5. Overview of User Interface

A.4.4.2.1 Dashboard

Outstanding and completed tasks associated with the user will be available from the dashboard and the user will be able to access the different view/editors from this page. If there is data output that the user has not accessed, the dashboard will indicate to the user that new data output is available.

A.4.4.2.2 Consist View/Editor

The FSTBSE will provide the capability to create and store new train consist models. The existing train consist models will be imported and stored in the new simulation environment to allow users access to prebuilt train consists. New train consist models can be created by adding each rail vehicle of the train consist, and any required information specific to that vehicle, or by copying an existing train consist model and modifying it as needed. Data provided by the user includes general data about the train consist and data for specific vehicles within it. The user-provided data must include adequate detail to allow the system to add vehicles for the train consist model, either from existing vehicle models or by creating a new vehicle based on the data provided. It is assumed that new vehicle models will be built using a previously modeled "base" vehicle that matches the type of vehicle being requested, along with the user-provided data and/or vehicle-specific data from Umler®, to create a representative model.

The Consist View/Editor includes a table view of the train consist that shows each vehicle within it and the pertinent information associated with each vehicle. The Consist View/Editor will include an option to display summary train consist information for the model, like the summary consist data contained in a PTC 01030 Train Consist message [4]. The Consist View/Editor will allow the user to export a train consist model in a readable format such as an Excel file.

User-provided data for a train consist and the vehicles therein includes:

- General train consist information
  - Consist name
  - Consist type (unit freight, general freight, intermodal, etc.)
  - Industry usage flag
    - Set to True – allows other railroads to view and select the consist model for use in simulations
    - Set to False – allows only the railroad associated with the user to access and use the train consist model
  - Consist comments (optional)
- Locomotive information
  - Reporting marks or Umler vehicle code
  - Horsepower
  - DB status
  - Run/isolate status
  - Weight
  - Length
  - Position in train consist
- Vehicle information
  - Reporting marks or Umler vehicle code
  - Tare weight
  - Length
  - Load
  - Brake cut-out status
  - Position in train consist

The system will initially allow the user to manually create the train consist model through an application or web interface.

A.4.4.2.3 Track File View/Editor

The FSTBSE will provide the capability to create new track files. The existing track models will be imported and stored in the new simulation environment to allow users access to prebuilt track files. New track files can be created from user-provided inputs for the elevation and curvature of the track file being modeled or by uploading an XML version of a PTC track data file(s) compliant with the ITC PTC track data file standard, as well as from user-provided inputs for the track(s) to be modeled from the PTC track data file.

The Track File View/Editor includes a graphical view showing the track gradient and curvature versus distance. The Track File View/Editor allows the user to export a track file in a readable format such as an Excel file.

User-provided data will be needed when creating a new track file or when creating a track file from a PTC track data file(s).

- User input data to create a new track file:
  - Track file name
  - Track file comments (optional) – provides more detailed information about the track for user reference
  - Elevation data point entries
  - Curvature data point entries
  - Milepost or text point entries (optional)
  - Industry usage flag (default to False, may only be included as Admin functionality)
    - Set to True – allows other railroads to view and select the track file for use in simulations
    - Set to False – allows only the railroad associated with the user to access and use the track file
- User actions and input data to create a track file(s) from an XML PTC track data file(s):
  - Create track file for single track name within subdivision
    - Select or load PTC track data file
    - Select track name from populated list
    - Track file name
    - Track file comments (optional) – provides more detailed information about the track for user reference
    - Industry usage flag (default to False, may only be included as Admin functionality)
      - Set to True – allows other railroads to view and select the track file for use in simulations
      - Set to False – allows only the railroad associated with the user to access and use the track file
  - Create a track file for a route within a single XML PTC track data file (planned for future)
    - Select or load a PTC track data file
    - Select track names and transition locations for the route
    - Track file name

        - Track file comments (optional) - provides more detailed information about track for user reference
        - Industry usage flag (default to False, may be included only as Admin functionality)
            - Set to True – allows other railroads to view and select track file for use in simulations
            - Set to False – allows only the railroad associated with the user to access and use the track file
    - Create track file for route spanning across multiple XML PTC track data files (planned for future)
        - Select or load PTC track data files
        - Select track names and transition locations for route
        - Track file name
        - Track file comments (optional) - provides more detailed information about track for user reference
        - Industry usage flag (default to False, may be included only as Admin functionality)
            - Set to True – allows other railroads to view and select track file for use in simulations
            - Set to False – allows only the railroad associated with the user to access and use the track file

The system will initially allow the user to manually create the track file or upload the PTC track data file(s) through an application or web interface.

A.4.4.2.4 Train Handling View/Editor

The FSTBSE will provide the capability to create new train handling files. The existing train handling files will be imported and stored in the new simulation environment to allow users access to prebuilt train handling files. New train handling files can be created from manual user input values or by copying and modifying an existing file.

The Train Handling View/Editor will include a table view of the file that shows each train handling command included within the file. The Train Handling View/Editor will allow the user to export a train handling file in a readable format such as an Excel file.

User-provided data for a train handling file includes:

- Train handling file name
    - Train handling file comments (optional) – provides more detailed information about the track for user reference
    - For each train handling entry:
        - Select train handling command from list of available commands

- Brake set, brake release, brake emergency, run/idle/dynamic, locomotive bail, locomotive independent, sand, and additional commands can be added (if they exist in TOES)
- Select simulation time/location to issue train handling command

A.4.4.2.5 Batch Table View/Editor

The FSTBSE will provide the capability to create new batch tables from manual user input or by copying and modifying existing batch tables. The existing batch tables will be imported and stored in the new simulation environment to allow users access to prebuilt batches. A batch can contain one or more simulation scenarios created by the user.

The Batch Table View/Editor will include a table view of the batch table that shows each simulation scenario defined within the batch table and the pertinent information for each. The Batch Table View/Editor will allow the user to export a batch table in a readable format such as an Excel file.

The FSTBSE will allow the user to set up initial simulation conditions for each scenario being created within a batch and save them in user-defined batches.

User input data for a batch includes:

- Batch table name
- Batch table comments (optional) – provides more detailed information about batch being created
- Industry usage flag
  - Set to True – allows other railroads to view and select batch when creating an operational scenario
  - Set to False – allows only railroad associated with user to access and use batch when creating operational scenario
- Simulation scenario name
  - For each simulation scenario name added to the batch, user input data includes:
    - Simulation scenario comments (optional)
    - Consist file name
      - Selectable from previously created train consist models available to the user
    - Track file name
      - Selectable from previously created track files available to the user
    - Train handling file name
      - Selectable from previously created train handling files available to the user
      - "NULL" is acceptable and simulation will run without user-defined train handling commands
    - Initial simulation conditions

        - Train start speed
        - Train start location
        - Locomotive throttle/dynamic initial simulation settings
        - Movement direction
        - Number of simulations
    - Target information (only applicable for simulations with EA interaction)
        - Target speed(s)
        - Target location(s)

A.4.4.2.6 Simulation Job View/Editor

The FSTBSE will allow users to create, save, and request a simulation job through the Simulation Job View/Editor. A simulation job will be created from user input, which will allow the user to customize the following:

- Batch tables to include in the simulation job, with the ability to select all simulation scenarios within batch tables or subsets of simulation scenarios
- Desired simulation mode and simulation type
- Setup simulation behavior
- Ability to modify the values and behavior of simulation parameter values and variation
- Select desired data output and analysis

The Simulation Job View/Editor will allow the user to select previously created simulation jobs to execute, and it will allow the user to modify a previously created simulation job and save it as a new job. The FSTBSE will have specific industry wide simulation jobs (i.e., Monte Carlo PTC braking EA evaluation simulation jobs) available that will either not allow customization by the user or only allow limited modifications.

User input data to create a simulation job includes the following steps:

1. Enter simulation job name
    - Simulation job comments (optional) provide information about the simulation job being created
2. Select batch table(s)
    - Selectable from previously created batch tables available to the user. For each batch table the user has the following options:
        - Default setting will include all simulation scenarios within the batch
        - User can select a subset manually or by filtering
3. Select simulation mode
    - TOES-only simulation
    - Simulation interacting with EA

- Select EA from stored versions available to the user

4. Select simulation type
   - Stop distance simulation – runs specified simulations with a penalty and/or emergency brake application and records stopping distance (available in TOES-only simulation mode)
   - Train handling simulation – runs specified simulations issuing user-defined train handling commands to the simulation; train handling commands will be issued to the simulation until completion or until the EA issues a brake enforcement (available in both simulation modes)
   - Enforcement simulation – runs specified simulations against an EA with a target speed and location defined (available in simulations interacting with EA)
5. Setup simulation behaviors – default setting will be defined; user only needs to update if desired simulation behavior is different than defaults
   - Select Emergency Brake Backup (EBB) settings and behavior (only applicable for enforcement simulations)
     - EBB
       - Enabled
       - Disabled
     - EBB application behavior
       - Head end train application – emergency brake applied from head end locomotive consist only
       - Head and rear end train application – emergency brake applied from head end and remote locomotive consists
       - No emergency application - the request for emergency will be logged, but emergency brake will not be applied
   - Select DB settings and behavior
     - Use DB after enforcement (only applicable for enforcement simulations)
       - Enabled
       - Disabled
     - DB behavior
       - Probability of DB failure
       - Remote locomotive consist(s) DB behavior at enforcement (only applicable for enforcement simulations)
         - ✓ Go to idle
         - ✓ Follow DB settings from lead locomotive consist
   - Select Locomotive Air Brake behavior

        - Locomotive bailing
            - Enabled
            - Disabled
        - Locomotive bail settings for long and short consists
        - Remote locomotive consist(s) locomotive bail settings at enforcement (only applicable for enforcement simulations)
    - Set Cruise Control logic
        - Enabled – TCL will attempt to drive the simulation at specified test speed if cruise control is enabled
        - Disabled
    - Select Back-Office Brake Force Calculation method
        - None
        - Industry
        - Specific provided value
6. Set simulation parameter values and variation – Default settings will be defined; user only needs to update if simulation behavior is different than defaults
    - Specific parameters that can be changed have not been defined yet, but could include turning on and off variance of simulation parameter values and variation, setting a specific value for a varied parameter, and updating the desired parameter range and distribution type
7. Select desired data output – at least of one of the following will need to be selected:
    - Raw summary simulation results from SQL database. Each simulation has records for the following:
        - Initial conditions
        - Target information
        - Penalty application time
            - Speed and location at penalty application
        - Emergency application time
            - Speed and location at emergency application
        - Stopping time and location
    - Detailed analysis of simulations on a scenario-by-scenario basis
    - Detailed analysis of the simulations grouped per user selected groups: train type, track grade, speed, etc.
    - Maximum in-train force per time step of the simulation
    - TOES logged data

- Summary train consist data – summary data stored for each train consist as it was generated for the simulation
- Summary simulation parameter data – summary data stored for each simulation as it was generated

Once a simulation job is created the user can request it be completed. Requested simulation jobs will be processed by the Simulation Controller/Manager (see Section A.4.4.4) with output data available to the user upon completion.

A.4.4.2.7 Data Output View

Upon completion of the simulation job, the user will receive notification (email and dashboard) and data output options selected in the simulation job will be available to the user in the Data Output View. These results will be displayed on the dashboard and available for download as either a .pdf or .csv (or other) file type. Figure A-6 is an overview of the storing, notification, and accessing of the data outputs.

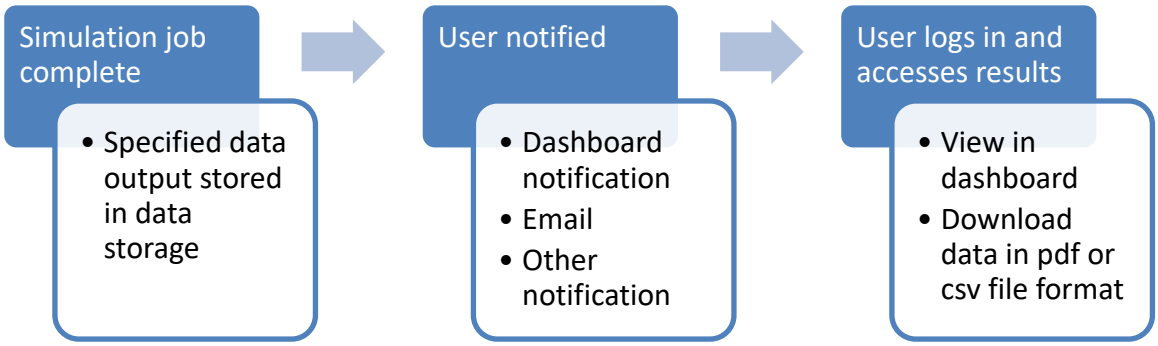


Figure A-6. Data Output Process

The capability to purge stored data will be required, but specific rules on data retention and purging will be determined later.

A.4.4.3 Application Interface (Future Functionality)

It is envisioned that the FSTBSE will be expanded in the future so that many of the user functions will be supported through an automated process using external applications that communicate with the FSTBSE through a standard application programming interface (API). These functions include the following:

- Consist selection/editing – It is envisioned that this capability will be expanded in the future to support creation of train consist models through an automated process. Using train consist data sent from an external railroad system to the FSTBSE via the API, the FSTBSE system will select an existing train consist model or create a new train consist model.
- Track file creation – It is envisioned that this capability will be expanded in the future to support creation of track files through an automated process. Using track data and route

information sent from an external railroad system to the FSTBSE via the API, the FSTBSE system will create a track file for the desired route.

- Train handling file creation – It is envisioned that this capability will be expanded in the future to support creation of train handling files through an automated process. Using train handling information sent from an external railroad system to the FSTBSE via the API, the FSTBSE system will create and store the train handling file in the FSTBSE.
- Batch creation – It is envisioned that this capability will be expanded in the future to support creation of batch tables through an automated process. Using simulation scenario information sent from an external railroad system to the FSTBSE via the API, the FSTBSE system will create a batch table and store the simulation scenarios within the batch table in the FSTBSE
- Simulation Job creation – It is envisioned that this capability will be expanded in the future to support creation of simulation jobs through an automated process. Using simulation job information sent from an external railroad system to the FSTBSE via the API, the FSTBSE system will create, save, and/or request a simulation job in the FSTBSE.
- Data output – It is envisioned that this capability will be expanded in the future to support viewing and downloading data outputs through an automated process. Available data outputs will be sent from the FSTBSE to external railroad systems via the API.

A.4.4.4 Simulation Controller/Manager

The Simulation Controller/Manager will be responsible for processing requested simulation jobs, monitoring their status, gathering output data needed for the jobs, and logging and reporting errors during processing. The Simulation Controller/Manager will have the ability to start a to-be-determined (TBD) number of simulation container. A container is a self-contained collection of the simulation tools (i.e., combinations of TCL, TOES, and EA when applicable – as described in Section A.2.3.1) necessary to set up and execute a simulation scenario. A base container for the desired combination of simulation tools (i.e., different EA versions, TCL only, TOES-TCL only), will be set up, stored, and available for the Simulation Controller/Manager's use. The Simulation Controller/Manager will be able to call and run multiple containers in parallel until all available TBD containers are in use, at which point the Simulation Controller/Manager will maintain a queue of simulation scenarios to be executed for a simulation job and will process them as current containers finish and close (allowing the creation of new containers). Figure A-7 is an overview of how the Simulation Controller/Manager processes simulation job(s).

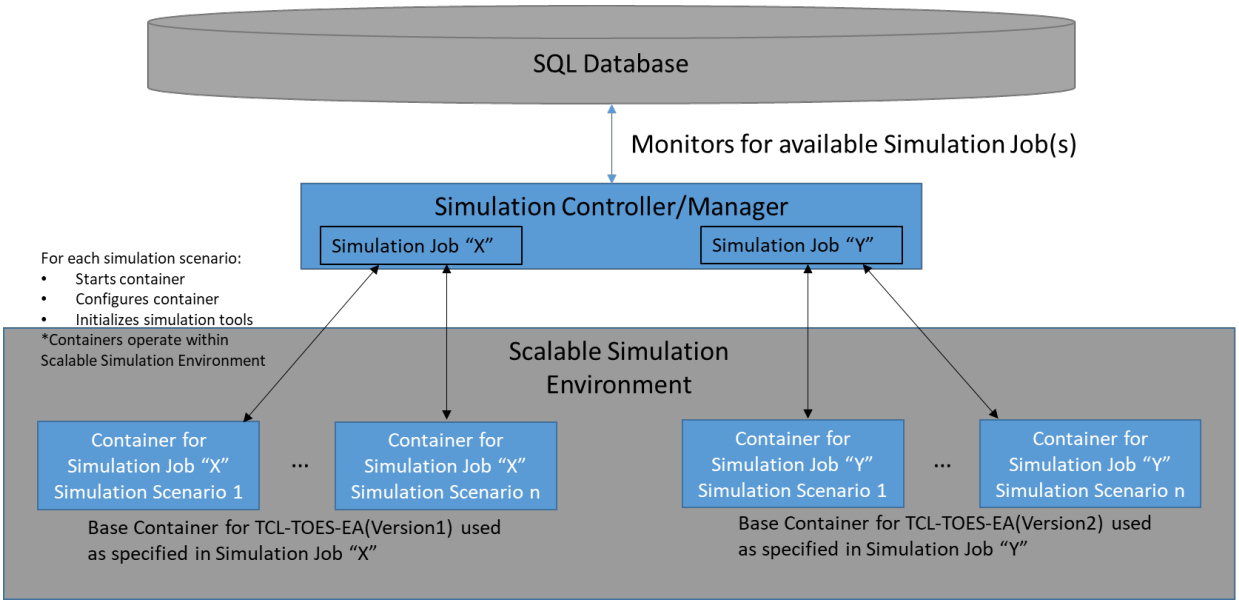


Figure A-7. Overview of Simulation Controller/Manager Processing Simulation Jobs

A.4.4.4.1 Simulation Job Processing

The Simulation Controller/Manager will monitor the database that stores simulation job requests that are pending processing. When a simulation job request is created in the database, the Simulation Controller/Manager will methodically process each simulation scenario within the job from the user data input during job creation. If TOES simulation files for train consist, track, or commands need to be created for the simulation scenario, then the Simulation Controller/Manager will manage the creation of a container to write the TOES files before running the simulation scenario. The following steps are completed for each simulation scenario within a simulation job:

1. Start a container with the simulation tools necessary for the simulation scenario:
    a. TCL only – if the Simulation Controller/Manager is creating TOES simulation files for a simulation scenario; no simulations are included in this setup
    b. TCL-TOES only – if the Simulation Controller/Manger is running the simulation scenario in the TOES-only simulation mode
    c. Specified TCL-TOES-EA – if the Simulation Controller/Manager is running the simulation scenario with EA interaction
2. Set container environmental parameters based on how the simulation job was configured when created
3. Designate access to simulation files needed for the simulation scenario
4. Initiate simulation tools in container

The Simulation Controller/Manager will repeat this process for all simulation scenarios in a simulation job.

A.4.4.4.2 Simulation Monitoring

The Simulation Controller/Manager will monitor containers that it has started as well as the queue of remaining simulation scenarios in a simulation job. Once all the simulation scenarios within a job are completed, the Simulation Controller/Manager will start the process for gathering needed data outputs and analysis. The Simulation Controller/Manager will monitor error codes received from a container and will enter the associated simulation scenario back into the queue a TBD number of times or until the simulation scenario is completed successfully.

A.4.4.4.3 Data Output Processing

Upon completion of a simulation job, the Simulation Controller/Manager will gather the required output data that was specified by the user during the creation of the job. The Simulation Controller/Manager will complete the necessary analysis on the output data. Output data and analysis will be stored in the appropriate format and saved in a data storage location. The Simulation Controller/Manager will set a notification in the database that data is available for the user in the Data Output View of the user interface.

A.4.4.4.4 Error Handling

The Simulation Controller/Manager will log errors it receives during the processing of simulation scenarios. Persistent errors of the same simulation scenario will trigger notifications (by email) to system administrators and to the user who created the associated job. Error information will be used for troubleshooting purposes to determine what caused the error and what changes can be made to address the situation in the future.

A.4.4.5 Scalable Simulation Environment

The Scalable Simulation Environment will be the infrastructure in which the Simulation Controller/Manager can start containers. The size of the infrastructure is TBD, but it will have the capability for many containers to run at the same time. The Scalable Simulation Environment will house the database and file storage needs for the FSTBSE.

A.4.4.5.1 Stored Containers

A repository of the different simulation tool setups, saved as containers, will be stored in the Scalable Simulation Environment. These containers will be configured by system administrators and the availability of the container to be used for a simulation job will be managed through a database table. The simulation type options available during the creation of a simulation job will depend on which containers are enabled for that user.

A.4.4.5.2 Container Operations

Once a container is called in the Scalable Simulation Environment, its sole purpose is to complete the simulations associated with that container, as set up by the Simulation Controller/Manager. Upon completion of this job, the container will communicate its status with the Simulation Controller/Manager and terminate.

A.4.5 User Classes and other Involved Personnel

The expected users of the FSTBSE are Railinc, TTCI, and AAR member railroads. Users will be assigned to one or more of the following user classes:

- System Administrator – This user class will require full access to the simulation environment. They will manage the addition or removal of users and assign user classes.
- Railroad Administrator User – This user class will have full access to the system used by their organization. They will manage the authorized users within their organization and the user class to which they are assigned.
- Railroad User – This user class will need to be able to access the user interface and the capabilities of the user interface for their organization.
- Environment Administrator – This user class will manage the simulation environment.
- EA Administrator User – This user class will have the ability to provide versions of its EA to the System Administrator to be integrated into the simulation environment. They will indicate which organizations are authorized to interact with its EA.
- Read-Only Organizational User – This user class will be able to access only the results of previously completed simulations.

A.4.6 Support Environment

The FSTBSE will be hosted and managed by Railinc and they will provide support for FSTBSE users. Railinc will support integration testing and troubleshooting for the simulation tools that will operate within the FSTBSE.

TTCI will provide support for the integration and troubleshooting of the TCL and TOES software packages in the FSTBSE and provide support for integration and troubleshooting of vendor EA software within the FSTBSE.

EA vendors will provide Railinc with software packages for versions of their EA to be integrated in the FSTBSE. EA vendors will be contacted for troubleshooting support as needed.

A.5 Operational Scenarios

The FSTBSE will support setting up and running three main scenarios: a single simulation, multiple simulations, or Monte Carlo simulations. The following sections describe each scenario in more detail.

A.5.1 Single Simulation Scenario

A.5.1.1 Background

The system allows the user to set up a single simulation scenario containing a single train consist, a single track, and a single train handling file. The user will have the ability to run a single simulation of the scenario or multiple Monte Carlo simulations of the scenario with the variations tailored by the user. Initial setup of this information will be stored in the batch table. For additional simulations using the initial batch file, the user will have the option to select the original scenario and then modify settings before running the simulation.

A.5.1.2 Initial Conditions

- User is securely logged in.
- User has knowledge of the desired train consist, track, train handling, and simulation settings.
- User has knowledge of the desired output data from the simulation.

A.5.1.3 Desired Outcome

The user creates and requests a simulation job for a single simulation scenario and accesses the simulation results after the FSTBSE has completed the simulation job.

A.5.1.4 Steps

1. User views existing train consists: is the desired train consist available?
    a. Yes – Move to step 2
    b. No – User creates a train consist using Consist View/Editor
2. User views existing tracks: is the desired track available?
    a. Yes – Move to step 3
    b. No – User creates a track using Track File Editor
3. User views train handling files: does the desired train handling file exist?
    a. Yes – Move to step 4
    b. No – User can create the file using Train Handling View/Editor commands
    c. None needed – Move to step 4
4. User selects train consist, track, and train handling information to create and save a batch through the Batch Table View/Editor.

5. User creates a simulation job by selecting preferences in Simulation Job View/Editor.
6. User submits the simulation job request.
7. Scalable Simulation Job View/Editor queues the simulation job to run in FSTBSE.
8. Simulation Controller/Manager pulls the simulation job request from the SQL Database, starts a container, and initializes the simulation.
9. Simulation Controller/Manager creates and stores the simulation results.
10. Simulation Controller/Manager notifies the user that simulation results are available.
11. User views or downloads simulation results in the Data Output View.

A.5.1.5 Variations

1. If the train consist, track, and train handling files were previously created and saved in a batch, then the user will start at Step 5.
2. The train consist and track information were previously created and saved in the Batch table, and the user needs to add or update train consist and track information.
    a. User selects the prebuilt train consist and makes any desired changes in the Consist View/Editor.
    b. User selects a prebuilt track and makes any desired changes in the Track File View/Editor.
    c. User continues to Step 3.

A.5.2 Multiple Simulation Scenario

A.5.2.1 Background

The system allows the user to create multiple simulation scenarios (combinations of one or more track, train consist, and train handling commands). The user will have the ability to run a single simulation for each scenario, tailored to the parameters and values the user inputs, run multiple simulations for each scenario using a combination of user provided parameters and values (including default Monte Carlo variations for information not provided by the user), or run multiple simulations for each scenario using a combination of user provided parameters and values that include user-modified Monte Carlo variations. Initial setup of this information will be stored in the batch table. For additional simulations using the initial batch files, the user will have the option to select the original scenarios and then modify settings before running the simulations.

A.5.2.2 Initial Conditions

- User is securely logged in.
- User has knowledge of the desired train consists, tracks, train handling, and simulation settings.
- User has knowledge of the desired output data required from the simulations.

A.5.2.3 Desired Outcome

The user creates and requests a simulation job for multiple simulation scenarios and accesses the simulation results after the FSTBSE has completed the job.

A.5.2.4 Steps

1. Are all needed train consists available?
    a. Yes – Move to Step 2
    b. No – User creates needed train consists using Consist View/Editor
2. Are all needed tracks available?
    a. Yes – Move to Step 3
    b. No – User creates needed tracks using track editor
3. Do the train handling files exist?
    a. Yes – Move to Step 4
    b. No – User can create train handling commands
    c. None needed – Move to Step 4
4. User creates simulation scenarios in the Batch Table View/Editor. Each simulation scenario has a designated train consist, track, and train handling file.
5. User creates simulation job preferences in Simulation Job View/Editor. Simulation job preferences can be set individually for each simulation scenario or as a global setting for all simulations contained in the job.
6. User submits simulation job request.
7. Scalable Simulation Job View/Editor queues the simulation job to run in FSTBSE.
8. Simulation Controller/Manager pulls the simulation job request from the SQL Database, starts a container, and initializes simulations for each simulation scenario defined within the job.
9. Simulation Controller/Manager creates and stores the simulation results.
10. Simulation Controller/Manager notifies the user that simulation results are available.
11. User views or downloads the simulation results in the Data Output View.

A.5.2.5 Variations

1. If previously created simulation scenarios are available in batch table, the user will start at Step 5.

A.5.3 PTC Monte Carlo Simulations Scenario

A.5.3.1 Background

For PTC braking EA evaluation simulations, the user will be able to select from a predetermined simulation matrix. The user may elect to run the full simulation matrix or use filters to execute a subset of the simulation matrix. Filtering options may include a combination of train type, track grade, simulation speed, or other options as needed.

The user will not be able to modify which simulation parameter values and variation are used within the simulations and will not be able to change the range and distribution types for these parameters. The default values will be used based on the values agreed upon by the industry, which will be saved in the database and updated based on industry-recommended changes.

The user will be able to select one or more configurations for the evaluation. Currently, the four selectable configurations are:

- EBB enabled without providing back-office brake force
- EBB enabled with providing back-office brake force
- EBB disabled without providing back-office brake force
- EBB disabled with providing back-office brake force

Data outputs and analysis will be available to the industry.

A.5.3.2 Initial Conditions

- User is securely logged in.
- User defines simulation settings or uses default PTC Monte Carlo Simulation settings.
- User defines the EA for analysis.

A.5.3.3 Desired Outcome

The user requests a simulation job for the Monte Carlo simulation and accesses the results after the FSTBSE has completed the job. The user accesses the PTC Monte Carlo standardized results and raw data if desired.

A.5.3.4 Steps

1. User selects PTC Monte Carlo simulation job from Simulation Job View/Editor.
2. User selects available EA.
3. User selects full simulation matrix.
4. User selects one (or more) desired configurations.
5. User selects desired output.
6. Scalable Simulation Job View/Editor queues the simulation job to run in FSTBSE.
7. Simulation Controller/Manager pulls the simulation job request from the SQL Database, starts a container, and initializes simulations for each scenario.

8. Simulation Controller/Manager creates and stores the simulation results.
9. Simulation Controller/Manager notifies the user that simulation results are available.
10. User views or downloads the simulation results in the Data Output View.

A.5.3.5 Variations

1. During Step 3, the user selects a subset of the simulation matrix using the filter options. The user selects one (or more) of the following:
    a. Train types
    b. Speed
    c. Track grades
2. After the selection, the user continues to Step 4.

A.6 Summary of Impacts

A.6.1 Operational Impacts

The current environment will continue operating during creation of the FSTBSE, which will allow simulations to be run without impacting timelines. When the new simulation environment has been tested and proven to operate as expected, the speed of running simulations is expected to increase due to increased availability of simulation instances. After testing of the new environment is completed, simulation tasks may be fully shifted to the new environment or continue to operate in both the FSTBSE and the legacy system, as needed.

A.6.2 Organizational Impacts

TTCI will no longer have complete ownership of the simulation environment but will retain the rights and access to the system like those provided by the current simulation environment. Railinc will be responsible for the simulation environment and its interface, with the TTCI providing TOES, TCL, and simulation expertise. TTCI will maintain the current simulation environment until such time as it is deemed unnecessary or no longer cost-effective to maintain.

A.6.3 Impacts during Development

During the development of the FSTBSE, simulations run in the new simulation environment will be compared to the current environment results and evaluated for accuracy. In order to verify that TCL and TOES are running correctly, personnel will need to create and run tests and evaluate the results. These simulation results will be evaluated for statistically similar stopping distances, stopping locations, and operations.

The proposed simulation environment will create a container that holds a virtual TOES and TCL instance and then duplicate that container as many times as needed. Testing will need to be conducted to determine that there is no cross talk or messaging issues between the containers, as well as to ensure that data read/write access is available to all containers without causing any errors.

Vendors may need to provide different builds for both simulation environments (testing and current). Depending on the level of development required, this may be prohibitive to new vendors.

A.7 Analysis of the Proposed System

A.7.1 Summary of Improvements

Creating a fully scalable simulation environment will allow for increased instances of simulation sets, decreased overall simulation time, and may allow for improvements to creating rail vehicles, tracks, and train consists. The FSTBSE will support on-demand simulations for additional users.

A.7.2 Limitations

The FSTBSE will operate in a Linux environment, which requires modification to TOES and TCL and the communications between these programs. Additionally, the database used by TCL is currently a Microsoft SQL database and will need to be converted to a database that is compatible with a Linux environment.

A.7.3 Alternatives and Trade-offs Considered

Initially, it was assumed that the fully scalable environment would continue in a virtual Windows environment, but discussions led to the decision to use a Linux environment. Remaining in a Windows environment would have required purchasing extra licensing for virtual Windows-based machines. Additionally, a Windows-based environment would not allow for efficient scalability. Finally, the team does not currently have the developers or experience necessary to support Windows-based programs.

A.8 Use Cases for Current Simulation Environment

FRA has facilitated a wide variety of freight train braking simulations to support the industry with PTC braking algorithm related testing and evaluation. The efforts that were key in developing the current simulation environment or were key uses of the simulation environment were identified and are documented in the subsequent subsections as Use Cases. The documentation for each Use Case includes a brief description and a summary of the setup, execution, and output data or results obtained. These Use Cases guided the development of the ConOps.

A.8.1 PTC EA Evaluation through Monte Carlo Simulation

A.8.1.1 Description

The Monte Carlo simulation set was created to perform a comprehensive evaluation of PTC braking EAs. It contains close to 3,800 simulation scenarios, with a scenario being comprised of a train consist, simulation speed, track grade, and target speed and location. Train consists include a mix of unit trains (coal, tank, refrigerated box, grain, autorack cars), mixed general freight, and intermodal freight. Each train type has multiple configurations for length, locomotive power (head end only or distributed power), and loading. Trains are operated on grades between a 1.5 percent incline to a 2.8 percent decline and run at maximum authorized speed as well as intermediate and slow speeds on certain grades. Figure A-8 gives a brief outline of the speed and grade combinations simulated.

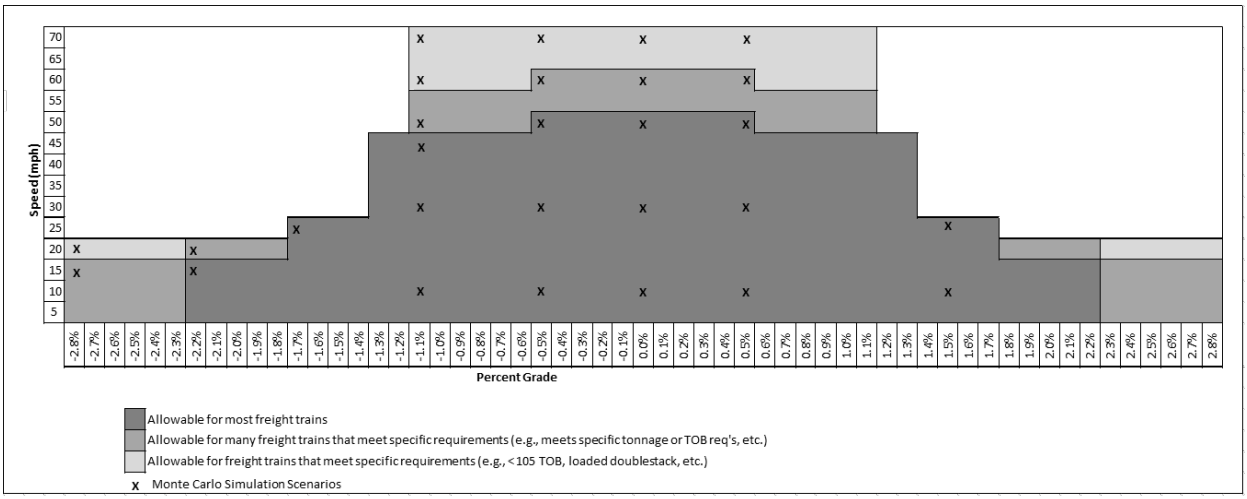


Figure A-8. Outline of Simulated Speed and Grade Combinations

For each scenario, 100 Monte Carlo simulations are generated, with train consist and simulation variables selected from defined ranges and distributions for each simulation. The simulations are then executed with TCL, the EA being tested, and TOES. TCL records simulation results, such as general scenario information, enforcement location and speed, stopping location, and target speed and location.

The Monte Carlo simulations are executed in four configurations:

- EBB enabled without back-office brake force

- EBB enabled with back-office brake force
- EBB disabled without back-office brake force
- EBB disabled with back-office brake force

Based on industry requests, TTCI executes Monte Carlo simulations on EA builds provided by the vendors. Simulation results are then shared with the vendor and the industry.

A.8.1.2 Setup

The setup requires up to 75 sets of simulation tools, referred to individually as a simulation instance (i.e., TCL, TOES, and the EA) configured to run in parallel. The train consists, track profiles, train handling, and initial conditions are already configured within 75 predefined batches of simulation scenarios.

A.8.1.3 Execution

The 75 predefined batches are assigned across the available simulation instances, one batch per simulation instance, and the user selects an EBB and back-office brake force behavior configuration to use for all the batches. A simulation instance will run the simulation scenarios defined in the batch sequentially until completion. If fewer than 75 simulation instances were configured, a subset of the batches (equal to the number of available simulation instances) will run and, upon completion of a batch, the user can assign another batch within the Monte Carlo simulation matrix. If no more batches are available, then the user can set up the simulation instance to start another EBB and back-office brake force configuration, if desired. Upon completion of all the Monte Carlo simulations, the user can pull the result data for analysis.

A.8.1.4 Output

For each of the EBB and back-office brake force behavior configurations, the Monte Carlo simulation results are analyzed to determine the probability of a train stopping short of the target location, the probability of a train overrunning the target, and the location where 99.5 percent of the trains stop relative to the target location. These results are provided for each train type (unit, mixed freight, and intermodal) and are used to support the safety case for the PTC EA. The results provide a performance metric for the probability of stopping short of the performance target, which is stopping the train more than 500 feet short of the target when the train enforcement speed was less than 30 mph or stopping the train more than 1,200 feet short of the target when the train enforcement speed was 30 mph or greater.

If desired, additional data analyses can be provided that characterize the Monte Carlo simulations on a scenario-by-scenario basis.

A.8.2 Field Test Modeling

A.8.2.1 Railroad Field Tests

A.8.2.1.1 Description

To support verification of the TOES models and the Monte Carlo simulation methodology, TTCI used data gathered from field testing of specific PTC EA versions under a wide range of operating conditions. TTCI coordinated with the industry to set up and conduct different field test scenarios across multiple railroads as well as at the Transportation Technology Center (TTC). The field tests included loaded unit trains, empty intermodal trains, and mixed freight trains with various speed and grade combinations. The testing included trains with and without distributed power.

TTCI worked with each railroad to gather data from the field tests to model and simulate the tests in TOES. The data gathered included:

- Detailed train consist information
- Track charts or PTC database
- PTC logs and locomotive event recorders

A.8.2.1.2 Setup

Field test data was used to complete the following tasks:

- Model the train consist(s) used in the field tests in TOES
- Model the track profile(s)
- Set up a simulation file for each field test completed so the simulated train speed, train location, and train handling conditions (throttle notch, DB, and/or brake sets) at the point of enforcement braking were the same as when the enforcement was initiated during the field test

A.8.2.1.3 Execution

TOES was used to simulate train behavior for each field test that included a penalty brake application.

Depending on the availability of the track profile within the EA, some field tests were modeled by setting up the scenario in TCL and generating 100 Monte Carlo simulations of that scenario to obtain a distribution of stopping distances.

A.8.2.1.4 Output

After the field tests were simulated in TOES and/or TCL, the simulation's stopping distance or stopping distance distribution was compared to the stopping distances from the field tests.

The comparison of these stopping distances was used to support the use of the TOES model and the Monte Carlo simulation methodology to support the evaluation of EAs through simulations.

A.8.2.2 DB Field Tests

A.8.2.2.1 Description

TTCI conducted a series of DB field testing at the TTC and on railroad track and then modeled and simulated the field test in the simulation environment using data gathered during the field tests. Data gathered included:

- Detailed train consist information
- Track charts or PTC database
- PTC logs and locomotive event recorders

A.8.2.2.2 Setup

Field test data was used to complete the following tasks:

- Model the train consist(s) used in the field tests in TOES
- Model the track profile(s)
- Set up a simulation file(s) for each field test completed so the simulated train speed, train location, and train handling conditions (throttle notch, DB, and/or brake sets) at the point of enforcement braking were the same as when the enforcement was initiated during the field test
- Set up desired train braking (dynamic and locomotive air brake) behavior based on the behavior used in the field test

A.8.2.2.3 Execution

TOES was used to simulate train behavior for each field test that included a penalty application and dynamic train braking.

Depending on the availability of the track profile within the EA, some field tests were modeled by setting up the scenario in TCL and generating 100 Monte Carlo simulations of that scenario to obtain a distribution of the stopping distances. The desired train braking behavior is configurable within TCL and was set up to match the field test being modeled.

A.8.2.2.4 Output

The results of the stopping distances from the TOES simulations were compared to the field test stopping distances and were used to help verify how TOES models DBs.

Results from simulations run in TCL would produce a stopping distance distribution for each simulation scenario created in TCL. This stopping distance distribution was used to compare to the actual stopping distance from the field test and was used to help support the Monte Carlo simulation methodology for evaluating PTC braking EAs.

A.8.2.3 Emergency Brake Field Tests

A.8.2.3.1 Description

TTCI conducted a series of emergency brake field tests at the TTC and on industry track and then modeled and simulated the field test in the simulation environment. Data gathered from the field tests was used to model the tests in TOES. Data gathered included:

- Detailed train consist information
- Track charts or PTC database
- PTC logs and locomotive event recorders

A.8.2.3.2 Setup

Field test data was used to complete the following tasks:

- Model the train consist(s) used in the field tests in TOES
- Model the track profile(s)
- Set up a simulation file(s) for each field test completed so the simulated train speed, train location, and train handling conditions (throttle notch, DB, and/or brake sets) at the point of enforcement braking were the same as when the enforcement was initiated during the field test
- Set up the desired emergency braking behavior based on the behavior used in the field test

A.8.2.3.3 Execution

TOES was used to simulate train behavior for each field test that included a penalty application followed by a train emergency braking.

Depending on the availability of the track profile within the EA, some field tests were modeled by setting up the scenario in TCL and generating 100 Monte Carlo simulations of that scenario to obtain a distribution of the stopping distances. The desired train emergency braking behavior is configurable within TCL and was set up to match the field test being modeled.

A.8.2.3.4 Output

The results of the stopping distances from the TOES simulations were compared to the field test stopping distances and were used to help validate how TOES models the emergency brake application.

Results from simulations run in TCL produced a stopping distance distribution for each simulation scenario created. The stopping distance distribution was used to compare to the actual stopping distance from the field test and was used to help support the Monte Carlo simulation methodology for evaluating PTC brake EAs.

A.8.2.4 Distributed Power Field Tests

A.8.2.4.1 Description

TTCI conducted a series of distributed power field tests at the TTC and on industry track and then modeled and simulated the field test in the simulation environment. Data gathered from the field tests was used to model the tests in TOES. Data gathered included:

- Detailed train consist information
- Track charts or PTC database
- PTC logs and locomotive event recorders

A.8.2.4.2 Setup

Field test data was used to complete the following tasks:

- Model the train consist(s) used in the field tests in TOES
- Model the track profile(s)
- Set up a simulation file(s) for each field test completed so the simulated train speed, train location, and train handling conditions (throttle notch, DB, and/or brake sets) at the point of enforcement braking were the same as when the enforcement was initiated during the field test
- Set up the desired distributed power train braking behavior (head-end locomotive dynamic and air brake behavior and remote locomotive dynamic and air brake behavior) based on the behavior used in the field test

A.8.2.4.3 Execution

TOES was used to simulate train behavior for each field test that included a penalty application followed by distributed power train braking.

Depending on availability of the track profile within the EA, some field tests were modeled by setting up the scenario in TCL and generating 100 Monte Carlo simulations of that scenario to obtain a distribution of the stopping distances. The desired distributed power train braking behavior is configurable within TCL and was set up to match the field test being modeled.

A.8.2.4.4 Output

The results of the stopping distances from the TOES simulations were compared to the field test stopping distances and were used to help validate how TOES models brake applications for trains with distributed power.

Results from simulations run in TCL produced a stopping distance distribution for each simulation scenario created. The stopping distance distribution was used to compare to the actual stopping distance from the field test and was used to help support the Monte Carlo simulation methodology for evaluating PTC brake EAs.

A.8.2.5 Independent Brake Field Tests

A.8.2.5.1 Description

TTCI conducted a series of locomotive independent braking field tests at the TTC and then modeled and simulated the field test in the simulation environment. Data gathered from the field tests was used to model the tests in TOES. Data gathered included:

- Detailed train consist information
- Track charts or PTC database
- PTC logs and locomotive event recorders

A.8.2.5.2 Setup

Field test data was used to complete the following tasks:

- Model the train consist(s) used in the field tests in TOES
- Model the track profile(s)
- Set up a simulation file(s) for each field test completed so the simulated train speed, train location, and train handling conditions (throttle notch, DB, and/or brake sets) at the point of enforcement braking were the same as when the enforcement was initiated during the field test
- Set up desired the locomotive independent braking behavior based on the behavior used in the field test

A.8.2.5.3 Execution

TOES was used to simulate train behavior for each field test that included a penalty application and locomotive independent braking.

Depending on the availability of the track profile within the EA, some field tests were modeled by setting up the scenario in TCL and generating 100 Monte Carlo simulations of that scenario to obtain a distribution of the stopping distances. The desired locomotive independent braking behavior is configurable within TCL and was set up to match the field test being modeled.

A.8.2.5.4 Output

The results of the stopping distances from the TOES simulations were compared to the field test stopping distances and were used to help validate how TOES models locomotive independent brakes.

Results from simulations run in TCL produced a stopping distance distribution for each simulation scenario created. The stopping distance distribution was used to compare to the actual stopping distance from the field test and was used to help support the Monte Carlo simulation methodology for evaluating PTC brake EAs.

A.8.3 Specialty Train Type Simulations

A.8.3.1 Description

TTCI built models for a few different specialty train types and simulated them in TOES with and without an EA. Data gathered for the specialty train types included:

- Detailed vehicle and train consist information
- Desired simulation scenarios

Some of the specialty type trains simulated were:

- Roadrailers
- Snowplows
- Auto-Train (Passenger cars coupled with autorack cars)
- Work trains

A.8.3.2 Setup

Model test data was used to complete the following tasks:

- Model the vehicle(s) and train consist(s) for the specialty train type in TOES
- Set up desired simulation scenarios using modeled train consists

A.8.3.3 Execution

Simulation scenarios were executed in TCL to generate 100 Monte Carlo simulations of each specialty train type.

A.8.3.4 Output

The results from the simulations were used to help validate the train consists modeled in the TOES simulation environment. With confidence in the modeled train consists, simulation scenarios can be set up for these trains to be run with an EA. The results were analyzed to determine where the train stops in reference to the target location and initial speed, and the stopping distance distributions were compared to expected stopping distances provided by the railroads.

A.8.4 Other Simulation Testing

A variety of other simulation tests (not described in the previous sections) throughout the life of the freight braking projects have added additional TCL setup and simulation functionality, including the following:

- Back-office brake force calculations – TCL has the capability to provide back-office brake force to the EA depending on the method selected
  - Industry back-office brake force calculation
  - Actual back-office brake force as the train consist was built by TCL
  - User provided back-office brake force
  - No back-office brake force provided
- Record summary train consist information for a train consist created using TCL
- Record simulation parameters selected for each simulation created by TCL
- TOES output logs

A.9 References

1. Brosseau, J., Moore Ede, B., Pate, S., Wiley, R. & Drapa, J. (2013). Development of an Operationally Efficient PTC Braking Enforcement Algorithm for Freight (Report No. DOT/FRA/ORD-13/34). FRA.
2. Pate, S., Anaya, R., & Holcomb, M. (2019). PTC Braking Algorithm Evaluation Methodology Enhancement (Report No. DOT/FRA/ORD-20/27). FRA.
3. *Enforcement Algorithm Evaluation Process Overview and Communications Interface Specification, Rev 8.* (2021).
4. *Positive Train Control Office-Locomotive Segment – Interface Control Document.* AAR MSRP, Section K-IV, Standard No. S-9361.V3.1. (2021).

A.10 Terms

Term	Definition
Batches	Groups of simulation scenarios, usually based on similar initial settings or train consist makeup.
Container	A simulation instance packaged together and saved in a Linux environment so it can be accessed multiple times to set up and run simulations.
Monte Carlo Method	The statistical process used to randomize variables from expected ranges and the probability that the values will be seen in real world conditions.
Enforcement Algorithm (EA)	A program used to evaluate train operations against the safety target and provide braking commands to improve the safety of train operations.
Simulation	A single train consist, track, and command file grouping used to model a single scenario.
Simulation Instance	A combination of simulation tools configured to run simulations.
Simulation Scenario	A train consist, track, and command file grouping with a user-defined number of simulations. Each simulation within a simulation scenario is created using the Monte Carlo process.
Test Controller/ Logger (TCL)	Program that interacts with TOES and EA to set up and execute simulations, and to log results.
Train Operations and Energy Simulator (TOES™)	A longitudinal train dynamics model developed by the AAR that models the status of every railcar in each train at every time step of the simulation.
The Umler® System | Railinc	Umler® is the source of critical data for more than two million pieces of North American rail, steamship, and highway equipment.

Abbreviations and Acronyms

ACRONYM	EXPLANATION
AAR	Association of American Railroads
AG	Advisory Group
API	Application Programming Interface
ConOps	Concept of Operations
CONOPS	Concept of Operations
DB	Dynamic Brake
EA	Enforcement Algorithm
EBB	Emergency Brake Backup
EMS	Energy Management Systems
FRA	Federal Railroad Administration
FSTBSE	Fully Scalable Train Braking Simulation Environment
IP	Internet Protocol
ITC	Interoperable Train Control
MSRP	Manual of Standards and Recommended Practices
MxV Rail	Transportation Technology Center, Inc.
PTC	Positive Train Control
SQL	Structured Query Language
TBD	To-be-determined
TCL	Test Controller and Logger
TCP/IP	Transmission Control Protocol/Internet Protocol
TOES TFG	TOES Track File Generator
TOES™	Train Operations and Energy Simulator
TTC	Transportation Technology Center
VM	Virtual machine
XML	Extensible Markup Language