U.S. Department of Transportation

Federal Railroad Administration

Positive Train Control Braking Algorithm Evaluation Methodology Enhancement

Office of Research, Development, and Technology
Washington, DC 20590

DOT/FRA/ORD-20/27

Final Report
July 2020

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
July 2020

3. REPORT TYPE AND DATES COVERED
Technical Report
September 11, 2017–March 10, 2019

4. TITLE AND SUBTITLE
Positive Train Control Braking Algorithm Evaluation Methodology Enhancement

5. FUNDING NUMBERS
DTFR5311D00008L
Task Order F00782

6. AUTHOR(S)
Shad Pate, Rachel Anaya, Matt Holcomb

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

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
Washington, DC 20590

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
DOT/FRA/ORD-20/27

11. SUPPLEMENTARY NOTES
COR: Jared Withers

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

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This report describes the investigation and implementation of changes performed by Transportation Technology Center, Inc. (TTCI) to improve the efficiency and capabilities of the current Positive Train Control (PTC) braking algorithm evaluation methodology for freight operations.

Improvements included:

- Simulated cases with an addition of PTC enforcement warnings and enabling analysis of the results with a railroad network model
- A determination of whether additional granularity is needed in scenarios that represent common industry operations
- Modified the simulation process to increase simulation efficiency, which will decrease time and labor to complete PTC enforcement braking algorithm evaluations

14. SUBJECT TERMS
Positive Train Control, PTC, head end, HE, braking algorithm, simulation, enforcement, freight

15. NUMBER OF PAGES
63

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI
Std. 239-18
298-102

METRIC/ENGLISH CONVERSION FACTORS

ENGLISH TO METRIC

LENGTH (APPROXIMATE)

1 inch (in)	=	2.5 centimeters (cm)
1 foot (ft)	=	30 centimeters (cm)
1 yard (yd)	=	0.9 meter (m)
1 mile (mi)	=	1.6 kilometers (km)

AREA (APPROXIMATE)

1 square inch (sq in, in^2)	=	6.5 square centimeters (cm^2)
1 square foot (sq ft, ft^2)	=	0.09 square meter (m^2)
1 square yard (sq yd, yd^2)	=	0.8 square meter (m^2)
1 square mile (sq mi, mi^2)	=	2.6 square kilometers (km^2)
1 acre = 0.4 hectare (he)	=	4,000 square meters (m^2)

MASS - WEIGHT (APPROXIMATE)

1 ounce (oz)	=	28 grams (gm)
1 pound (lb)	=	0.45 kilogram (kg)
1 short ton = 2,000 pounds (lb)	=	0.9 tonne (t)

VOLUME (APPROXIMATE)

1 teaspoon (tsp)	=	5 milliliters (ml)
1 tablespoon (tbsp)	=	15 milliliters (ml)
1 fluid ounce (fl oz)	=	30 milliliters (ml)
1 cup (c)	=	0.24 liter (l)
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 yard (cu yd, yd^3)	=	0.76 cubic meter (m^3)

TEMPERATURE (EXACT)

[(x-32)(5/9)] °F = y °C

METRIC TO ENGLISH

LENGTH (APPROXIMATE)

1 millimeter (mm)	=	0.04 inch (in)
1 centimeter (cm)	=	0.4 inch (in)
1 meter (m)	=	3.3 feet (ft)
1 meter (m)	=	1.1 yards (yd)
1 kilometer (km)	=	0.6 mile (mi)

AREA (APPROXIMATE)

1 square centimeter (cm^2)	=	0.16 square inch (sq in, in^2)
1 square meter (m^2)	=	1.2 square yards (sq yd, yd^2)
1 square kilometer (km^2)	=	0.4 square mile (sq mi, mi^2)
10,000 square meters (m^2)	=	1 hectare (ha) = 2.5 acres

MASS - WEIGHT (APPROXIMATE)

1 gram (gm)	=	0.036 ounce (oz)
1 kilogram (kg)	=	2.2 pounds (lb)
1 tonne (t)	=	1,000 kilograms (kg)
	=	1.1 short tons

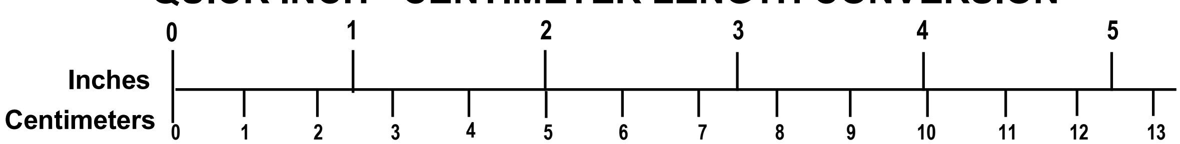
VOLUME (APPROXIMATE)

1 milliliter (ml)	=	0.03 fluid ounce (fl oz)
1 liter (l)	=	2.1 pints (pt)
1 liter (l)	=	1.06 quarts (qt)
1 liter (l)	=	0.26 gallon (gal)
1 cubic meter (m^3)	=	36 cubic feet (cu ft, ft^3)
1 cubic meter (m^3)	=	1.3 cubic yards (cu yd, yd^3)

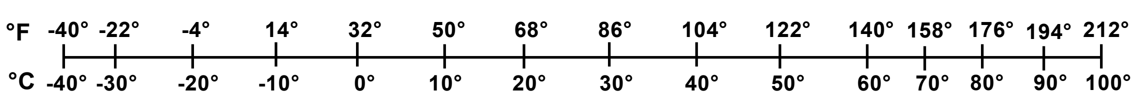
TEMPERATURE (EXACT)

[(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

Contents

Illustrations

Tables

Executive Summary

This report describes the tasks and actions taken to improve the efficiency and capabilities of the Positive Train Control (PTC) braking algorithm evaluation methodology for freight operations. The processes for executing the simulations and conducting the analysis of the results were enhanced. Results from the analysis were used to develop PTC deceleration rates that can be used within a railroad network model to enable the evaluation of the impact of PTC braking algorithms that is operating from a network perspective—as opposed to the perspective of a single train. Additionally, the simulation scenarios were reviewed and updated by Transportation Technology Center Inc. (TTCI), while working with an advisory group (AG) of freight railroad representatives. The Federal Railroad Administration (FRA) funded this project from September 11, 2017, to March 10, 2019, during which TTCI and the AG reviewed current scenarios used in the simulation methodology and determined if additional scenarios were needed, and/or if additional granularity was desired for scenarios that are more common within industry operations. In addition, simulation cases where the train crew adjusts their handling of the train based on the PTC enforcement Warning Time Input (WTI) were created and simulations ran to analyze deceleration rates with train crew inputs. Finally, TTCI worked to identify bottlenecks in the current PTC braking algorithm simulation process.

Areas of improvement in the PTC braking algorithm simulation process that were identified as having the greatest negative impact in the process were: software delays, processing issues, labor-intensive server setup, and manual data analysis. The following improvements and additional capabilities were included:

- Adding granularity to scenarios that represent common industry operations
- Adding simulation cases where the train crew adjusts their operations based on the PTC enforcement WTI and enabling analysis of the results with a railroad network model
- Modifying the simulation process to increase simulation efficiency, which decreases time and labor requirements to complete PTC enforcement braking algorithm evaluations, including:
  - Combining all the Train Operations and Energy Simulator (TOES™) simulation command files in a central data repository to allow access to multiple simulation servers
  - Improving the PTC braking algorithm simulation Test Controller/Logger (TCL) application to run automatic queries of a SQL database for TCL setup parameters
  - Configuring all simulation servers to login and restart simulations automatically in the event of server restart

1. Introduction

From September 11, 2017, to March 10, 2019, the Federal Railroad Administration (FRA) sponsored a project conducted by Transportation Technology Center, Inc. (TTCI) to enhance the current methodology used for Positive Train Control (PTC) braking algorithm evaluation for freight operations. The project focused on three areas:

- Reviewing the simulation scenarios to determine if additional scenarios were needed, or if additional granularity was desired for scenarios that are more common within industry operations
- Expanding the simulation capabilities so the results can be used within a railroad network model to evaluate the impact of PTC braking algorithms on the operation from a network perspective—as opposed to the perspective of a single train
- Identifying inefficiencies in the PTC freight braking simulation process and implementing efficiency improvements for running simulations, as well as analyzing simulation data

1.1 Background

FRA previously funded a research program in which TTCI, together with the Class I freight railroads, developed a methodology for analyzing PTC braking algorithm safety and performance characteristics. This analysis methodology is focused primarily on evaluating the algorithm against safety and operational objectives from the perspective of a train experiencing a PTC enforcement. To achieve this, for each configuration of the algorithm, hundreds of thousands of PTC braking enforcement simulations were run, each simulating a train operating at a constant speed on a pre-determined track grade. As the train approached the stop target, a warning was issued by the PTC system, but the simulated train continued to travel at its defined speed until the braking algorithm initiated a penalty brake enforcement application. The simulated stopping location was recorded and compared to the stop target, and an analysis was performed using this data for many simulations to determine the safety and operational performance of the PTC braking algorithm.

This methodology was accepted by the industry and FRA as the preferred approach for evaluating and demonstrating the effectiveness of the PTC braking algorithm and is utilized to evaluate each new braking algorithm software build currently used by the Class I railroads. However, as the characteristics of rail operations change and experience with the PTC braking algorithm simulation methodology is gained, it is useful to periodically review the simulation scenarios included in the simulation matrix to identify possible updates.

Additionally, as the methodology is now regularly used, areas for further refinement have been identified. Currently, the methodology simulates the response of the braking algorithm if the train crew takes no action. Operational performance of the algorithm can be evaluated, but only in terms of how far short of the target the train stops when a PTC penalty brake enforcement is applied. However, with visual and audible warnings from the system, the crew will generally adjust their train handling before the warning time reaches zero to avoid a PTC enforcement. An important aspect of the project that was considered was how the actions of the train crew are affected by the PTC system warnings, and the resulting impact on the overall operation, from a network perspective—as opposed to the perspective of a single train—is an important aspect to consider.

Previously, the implementation of the simulation methodology took 1 to 2 weeks to simulate and analyze a single configuration of the PTC braking algorithm, depending on the architecture and infrastructure used. Currently, there are currently four different configurations of the PTC enforcement braking algorithm that are generally simulated: 1) emergency brake backup (EBB) disabled with train brake force estimated by the PTC braking algorithm; 2) EBB enabled with train brake force estimated by the PTC braking algorithm; 3) EBB disabled with estimated train brake force provided by the PTC back office; and 4) EBB enabled with estimated train brake force provided by the PTC back office. It can take upwards of 2 months to complete simulations on a new build with all four configurations. With the continued need to reevaluate new releases of the PTC enforcement braking algorithm and the proposed additional scenarios, a more efficient simulation process is desired to provide results to the industry in a timely manner and reduce labor and time needed to complete simulations.

1.2 Objectives

The project objective was to investigate and implement changes to improve the efficiency and capabilities of the current PTC braking algorithm evaluation methodology for freight operations. Specifically, the following three aspects were considered:

- Updates to the simulation scenarios in the simulation matrix
- Additional capabilities to support the use of the results within a network model to support an analysis of the impact of the PTC braking algorithm from a network perspective
- Efficiency improvements to the implementation of the methodology to reduce the time and labor required to execute the process

1.3 Overall Approach

TTCI worked with an advisory group (AG) of freight railroads representatives to review the current simulation matrix and determine if any modification/refinement was needed. Additional simulation scenarios to address changes in railroad operating characteristics, as well as additional granularity for scenarios that are more common in the industry, were considered. TTCI documented all changes made to the simulation matrix.

To support the use of results from the braking algorithm simulation process in a railroad network model for a performance evaluation of the PTC braking algorithm from a network perspective, TTCI first worked with the AG to determine the scenarios and train handling rules to be added to the simulation process. TTCI then determined the data output and format needed to be used in a railroad network model. Finally, TTCI documented and implemented changes to support this performance evaluation.

To improve its efficiency, TTCI software and engineering personnel familiar with the simulation process coordinated to identify, document, and implement improvements. Improvements were made to the software, as well as to the implementation of the methodology used to run simulations.

1.4 Scope

The project scope included tasks to improve the efficiency and capabilities of the current PTC braking evaluation methodology for freight operations. The improvements were limited to:

- Updating the simulation matrix based on a review with the AG
- Modifying the process to produce results that can be used in a railroad network to evaluate the impact of a PTC braking algorithm from the perspective of the railroad network
- Making improvements to the implementation of the methodology to improve the efficiency with which the simulations are run and results are processed

Modification of the process to support railroad network braking algorithm impact evaluation was limited to the changes to the current process to support this type of analysis. Note that the results of the railroad network model and performing a railroad network braking algorithm impact analysis were not included in the project scope.

Improvements to the implementation of the methodology included an execution of simulations to evaluate the improvements made, but the project scope did not include the execution of the full simulation matrix or evaluation of a specific PTC braking algorithm.

The project scope included evaluating bottlenecks in the simulation process, as well as implementing software and system enhancements to improve the implementation of the methodology.

1.5 Organization of this Report

The sections in this report and its appendices describe the work performed to investigate and implement changes to improve the efficiency and capabilities of the current PTC braking algorithm evaluation methodology for freight operations, including:

- Section 2 provides a review of the simulation matrix to determine if scenarios are needed, or if additional granularity is needed in scenarios that represent common industry operations.
- Section 3 adds and runs Warning Time Input (WTI) simulation cases, where the train handling is adjusted based on the PTC enforcement warning to support an analysis of the operational impact of the braking algorithm from a railroad network perspective by using the results with a railroad network model.
- Section 4 depicts the modification of the simulation process and supporting software and systems to increase simulation efficiency, which will decrease time and labor to complete PTC enforcement braking algorithm evaluations.
- Section 5 provides a review of the work performed by TTCI with the AG that describes the results received and how those results can be beneficial for future work.
- Appendix A illustrates the braking PTC test, while Appendix B shows the WTI PTC test.
- Appendix C provides the enforcement evaluation process overview and communications interface specification.

2. Braking Algorithm Evaluation Methodology and Updates

The initial task completed within this project was a review of the current simulation matrix to determine if additional scenarios or additional granularity within the most common operational scenarios were necessary.

Section 2.1 provides an overview of the simulation process, consisting of background information that originally appeared in the final report from previous FRA-funded braking algorithm research conducted by TTCI [1].

Section 2.2 gives a brief update to modifications made to the Monte Carlo simulation test matrix.

2.1 Simulation Testing

The simulation testing component of the enforcement algorithm evaluation methodology makes use of a set of computer software tools to employ a Monte Carlo simulation process, which results in a set of output data that can be analyzed to identify the statistical probability and confidence that the algorithm will meet the specified safety and performance criteria. The Monte Carlo method involves running large numbers of simulations with inputs to the simulations randomly assigned based on the practical and physical distributions and limits that define the system. A deterministic evaluation is not feasible because of the wide range of parameters that affect the stopping distance of a freight train and the interdependence of these parameters. Thus, the Monte Carlo simulation process is the preferred method of evaluating the enforcement algorithm.

2.1.1 Overview of Simulation Testing Process

The simulation testing process is intended to evaluate the enforcement algorithm over the full range of operating scenarios that the system is expected to encounter while also considering the practical variability of the parameters that can significantly affect train stopping distance. The simulations are organized into test scenarios, each of which represents a potential operating scenario for the system to encounter. The test scenario is defined by the nominal train consist, the nominal track profile, the initial speed and location of the train, and the target stopping position.

Multiple braking enforcement simulations are run for each test scenario. The values of the parameters that can have a significant effect on train stopping distance are randomly selected for each simulation from distributions that represent the practical range of values for the given parameter. In some cases, the distribution of values for a parameter is affected by the value randomly selected for a different, but related parameter.

The test scenarios that make up the complete simulation test matrix are intended to include the boundary operating conditions and represent the full range of conditions that can occur. To make the simulation process more efficient, the test scenarios are organized into batches that are executed together. A batch could contain any number of test scenarios, each representing a different nominal operating scenario, and each test scenario could contain any number of individual simulations, each representing a potential specific instance of the test scenario. Figure 1 illustrates the relationship between batches, test scenarios, and simulations.

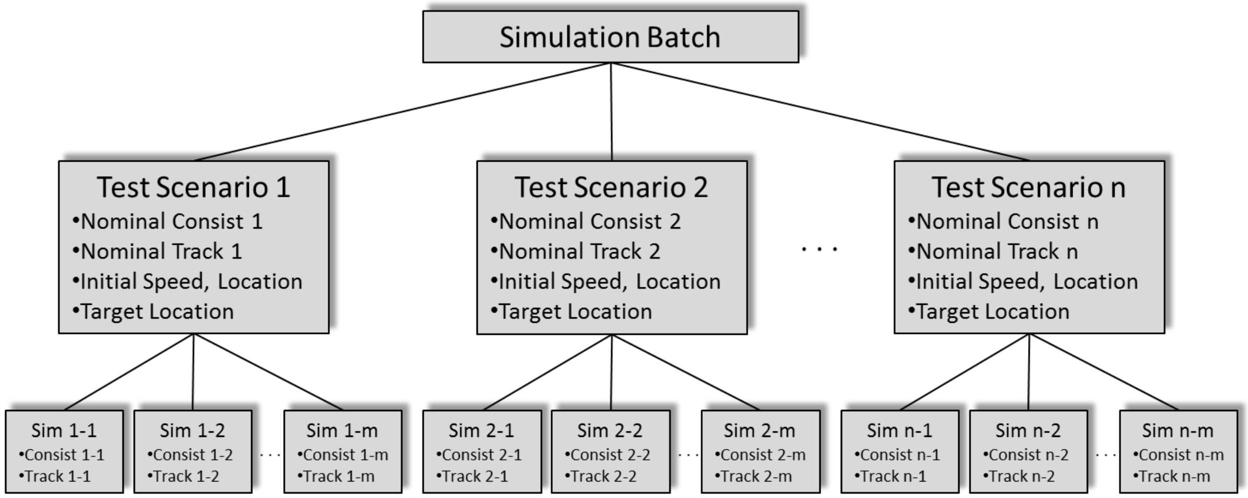


Figure 1. Organization of Simulations

For each simulation, the train is modeled as approaching the target at the defined initial speed, the enforcement algorithm triggers a brake application to prevent a violation of the stop target, and the response of the train is modeled. The result of the individual simulation represents a single possible stopping location for the given test scenario with the given enforcement algorithm. The aggregate result of the simulations for the entire test scenario then defines the distribution of possible outcomes. This data is analyzed to determine the safety and performance characteristics of the enforcement algorithm for the given test scenario. These characteristics can then be analyzed together to quantify the overall safety and performance characteristics of the enforcement algorithm.

2.1.2 Simulation Testing Tools

The simulation testing portion of the enforcement algorithm evaluation methodology requires the following three components, as Figure 2 illustrates:

- 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 test controller/logger (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 enforcement algorithm 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

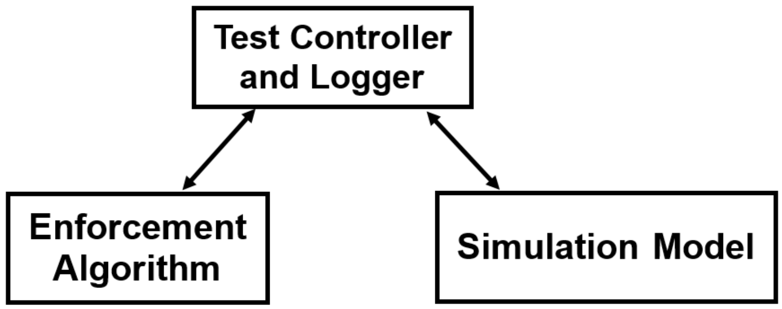


Figure 2. Simulation Testing Components

Simulation Model

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 car within the train and the specific characteristics of the track, and be capable of reporting train status data at regular, frequent intervals. Therefore, Train Operations and Energy Simulator (TOES™) was the simulation model selected for enforcement algorithm evaluation. TOES™ is a longitudinal train dynamics model developed by the American Association of Railroads (AAR) that models the status of every car in a given train at every time step of the simulation. Car status data includes location, velocity, acceleration, forces acting on the car, and brake system component status.

The model allows the user to enter specific characteristics for each car in the train, including car 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 essentially any currently used freight railcar and arrange them into any desired train consist. The model also 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 include some of the most accurate and proven models currently available to the railroad industry. These include a variety of draft gear models, multi-platform cars, an aerodynamic drag routine, and a variety of user-customizable car components. TOES™ also includes a theoretical fluid dynamics model of the air brake system. This model has been shown 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.

Test Controller/Logger Software

A custom software application was necessary to manage the vast number of simulations required to generate the necessary statistical significance for the safety and performance of the enforcement algorithm over the entire range of potential operating scenarios. To support the industry in the development and testing of a safe and operationally efficient braking enforcement algorithm, TTCI developed (i.e., using internal research and development funds) a TCL software application capable of generating and executing thousands of braking enforcement Monte Carlo simulations based on 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

The user can generate simulation input data by setting up a batch of test scenarios to be evaluated. The user selects a train consist and track profile and enters initial train speed and location, as well as target stopping location for each test scenario in the batch.

The train consists are defined by the user by selecting the desired cars and arranging them in the desired order. Each car is defined by the nominal components and characteristics of the car and the potential variation of these components and characteristics, also defined by the user. The variation of the car components and characteristics can be 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 also defines the potential variation of environmental characteristics and the variation because of errors in reported data, such as track characteristics, train speed, and location.

The user selects how many 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, 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 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 is then passed along to the enforcement algorithm. When the enforcement algorithm 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. The TCL continues to advance the simulation until the train is stopped. The enforcement algorithm can also 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.

Interface to Enforcement Algorithm

The enforcement algorithm evaluation methodology can be applied to evaluate any enforcement algorithm for any North American freight PTC implementation. As such, the methodology treats the software implementation of the enforcement algorithm 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 enforcement algorithm 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 enforcement algorithm to be implemented as an executable software application running on the same machine as the TCL software, as a virtual machine with a separate IP address, but operating on the same hardware as the TCL software, or as software running on separate hardware that communicates over TCP/IP.

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

To ease the integration of an untested enforcement algorithm 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 developer of the enforcement algorithm software 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 also available to support the development of the interface on the enforcement algorithm side without releasing the proprietary TCL software source code [1].

2.2 Test Matrix Review and Refinement

TTCI worked with the AG to review the current simulation matrix to: 1) determine if modifications/refinement were needed for any of the simulation scenarios, 2) to identify if additional scenarios were needed due to changes in the operational environment since the initial simulation matrix was created, and 3) to determine if additional granularity was desired for the more common operational scenarios. Changes to the simulation matrix were documented by TTCI and the Monte Carlo simulation methodology was updated to reflect the new test matrix. Based on input from the AG, it was determined that no additional granularity was required in the simulation matrix. The only change needed was the addition of 200-car unit trains, both loaded and empty. The updated simulation matrix is attached in Appendix A.

3. Enhancements to Support PTC Braking Algorithm Railroad Network Operational Impact Analysis

The objective of the PTC braking enforcement function is to only initiate a penalty brake enforcement when not doing so would result in a violation of an authority or speed limit, and not when the train crew is operating the train consistent with operating rules and within established authority and speed limits. PTC braking algorithms are designed to stop a train short of the target stopping location with some level of conservatism to account for the variability in the stopping distance of the train and provide a warning to the crew to indicate that the train is approaching a point where the system is predicting a penalty brake enforcement will be required to prevent an authority or speed limit violation. To prevent PTC from initiating a penalty brake application as a train approaches a speed restriction or stop target, the engineer will typically begin to slow the train when the warning is provided by the system, which may be earlier as they would begin to slow the train in the absence of the PTC warning. Slowing the train earlier can have an effect on following trains and other trains on the network. The current PTC braking algorithm evaluation methodology only considers the impact on the train being simulated. However, combining results from individual train scenario simulations with the capabilities of railroad network models, an analysis of the overall network impact of a conservative braking algorithm can be evaluated.

Due to the complexity of simulating all the trains on a network over a reasonable amount of simulation time, railroad network models generally use a simplified train performance calculator to determine when a given train will begin to slow for an upcoming speed or authority limit. By modeling both a normal deceleration rate (i.e., without PTC) and a PTC deceleration rate (i.e., the deceleration rate resulting from earlier initiation of train braking with PTC), the impact of the braking algorithm can be analyzed within the network model.

The established PTC braking algorithm simulation methodology can be used to estimate the PTC deceleration curve, based on simulating the engineer's reaction to the PTC warning for various scenarios. However, modeling this reaction to the warning curve and calculating PTC deceleration rates required several changes to how simulations are executed as well as identifying the scenarios to be used.

3.1 Identification of Operational Scenarios and Development of Train Handling Methods for WTI Simulations

TTCI identified which operational scenarios would be added to the current train-level simulation process. Input from the AG was used to determine that a subset of the current Monte Carlo simulation scenarios could be used for the WTI simulations. The WTI simulation matrix that was used during this project can be found in Appendix B. Results from these simulations were used to develop PTC deceleration rates that can be used to support a performance evaluation of the PTC enforcement algorithm in a railroad network-level model. These simulation scenarios can be used to evaluate new releases of the onboard PTC software, determine if the PTC deceleration rates have changed due to modifications of the PTC braking algorithm, and then used in the railroad network model to determine any changes to the operational impact.

For the WTI simulations, TTCI developed the capability within the TCL software to modify the train handling during the simulation to slow the train down based on the warning time feedback from the PTC braking algorithm, a user-specified warning time threshold number of seconds, and a user-specified distance short of the target to reach the target speed. Figure 3 shows the system parameters for TCL with the addition of warning time feedback parameters. The warning time feedback parameters included the ability to enable or disable the functionality as well as two warning time timers ("Timer 1," "Timer 2") and a target offset ("Target Offset"). Timer 1 and Target Offset were used for the WTI simulations and analysis. Timer 2 is used by TCL during the simulation to modify train handling as the train is being slowed, but was not needed for the calculation of the PTC deceleration rates.

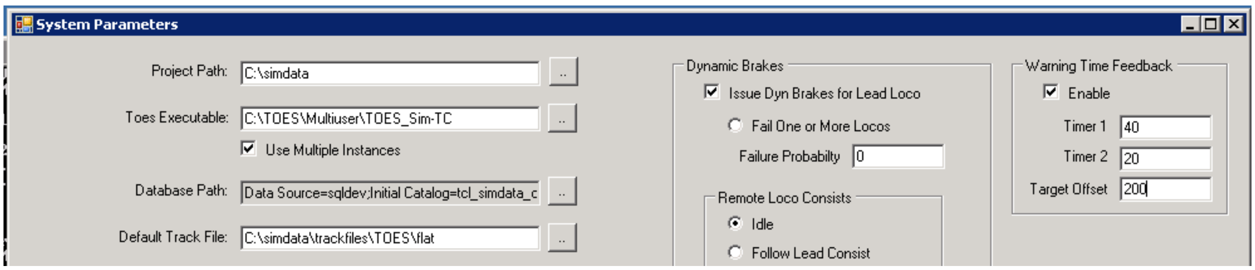


Figure 3. Warning Time Feedback Variables in TCL System Parameters

The TCL logic was modified so that, when the warning time drops below the user input value for Timer 1, the speed of the train and the location of the train is recorded in the results table. This, along with the target location and target speed, can be used during the analysis of the WTI simulations to determine the average deceleration rate of the train from the point the engineer begins slowing the train to the stop target location. This is calculated by first subtracting the Target Offset value set by the user from the distance the train was from the target location at the point warning time dropped below Timer 1. Next, the drop in speed over that distance is calculated by subtracting the target speed from the train speed at the point the warning time dropped below Timer 1. Using the speed drop over the known distance, the deceleration rate can be calculated using the equation in Figure 4.

$$Decelerations\left(\frac{ft}{sec^2}\right) = \frac{{Speed_{Drop}}^2}{(2 * Distance)}$$

Figure 4. Train Accelerations Rate Equation

Which:

$Speed_{Drop}$ is equal to the speed at the point the warning time dropped below Timer 1 minus the target speed (ft./sec.).

Distance is equal to the distance between the train location at the point the warning time dropped below Timer 1 and the target location minus the user input Target Offset (feet).

Simulations were run using the WTI simulation test matrix (Appendix B) along with a modified version of the I-ETMS™ PTC braking algorithm (Version 6.3.17.3), that included the capability to supply WTI during the simulation process. The calculated deceleration rates from these simulations were used for the analysis shown in Section 3.3.

When warning time feedback is enabled, the TCL logic is also modified to use this deceleration rate. This adjusts the train handling during the simulation to match a speed profile that would satisfy the calculated deceleration rate. The logic implemented within TCL starts with throttle notch adjustments and dynamic brake applications to satisfy the calculated deceleration rate, with gradual air brake applications as needed. This capability may be beneficial for future testing to evaluate PTC braking algorithm performance as the train slows when approaching a speed restriction or stop target. For example, evaluations could include:

- Train crew ability to pull within a specified distance of the target with PTC
- Warning time behavior as crew slows the train
- Scenarios that result in a braking enforcement even though the crew is reacting to the PTC warning

TTCI reviewed an existing railroad network model to determine what capabilities can be used to input PTC deceleration rates for PTC braking algorithm impact analysis. Currently, the model allows the user to specify a PTC deceleration rate on a train type basis, with unit and manifest trains grouped as a general freight train type, and intermodal trains as an expedited train type. Based on this, TTCI used results from WTI simulations to develop PTC deceleration rates for the general freight train type and the expedited train type. The deceleration rates were calculated using two different methods and can be implemented within railroad network models to best represent the model's behavior. In the first method, the deceleration rates were calculated using equation in Figure 4. In the second method, the deceleration rates were calculated initially using the same equation, but the effect of track grade was removed from the deceleration rate by calculating the deceleration due to the track grade and offsetting the PTC deceleration rate by this value. Using both methods, the deceleration rates were calculated using two different values for the engineer response time, referred to as Timer 1. The first set of simulations set Timer 1 to 40 seconds to determine deceleration rates if engineers react to the PTC warning at approximately 40 seconds to enforcement, and the second set of simulations used 20 seconds. Table 1 in Section 3.3 shows the results of this analysis. Values in Table 1 can be used in railroad network models that allow the user to input PTC deceleration data based on a train type.

Further analysis was completed on the simulation results to generate equations for deceleration rates that use more specific information on the train type, such as load condition and locomotive power arrangement, as well as variables for current train speed, average grade in front of train, train weight, and train length. Section 3.2 shows the results of this analysis, and they may be used within networking models to more accurately model the PTC deceleration rate of a specific train in a specific scenario.

3.2 Documentation and Implementation of Simulation Methodology Changes for Railroad Network Model Performance Evaluation

Several changes were needed to implement the ability to run simulations using WTI. The first was a change to the interface specification between TCL and the enforcement algorithm that would allow the warning time to be provided to TCL during the simulations. The previous interface specification between the enforcement algorithm and TCL did not include information about the PTC warning time, so the interface specification and integration test tool were updated to support this capability. The updated interface specification is attached in Appendix C. TTCI worked with Wabtec to review the updates and test an enforcement algorithm build that supports this functionality.

Modifications to include the warning time feedback parameters were also needed within TCL, shown in Figure 3, as well as updates to record warning time values in the result table. The result table was updated to include fields for:

- "WARNING_TIME" – Represents the simulation time step when the warning time dropped below "Timer 1"
- "WARNING_POSITION" – Represents the train position at WARNING_TIME
- "WARNING_SPEED" – Represents the train speed at WARNING_TIME

The logic for TCL was updated so that when warning time feedback is enabled, TCL monitors the warning time values sent from the enforcement algorithm and uses the user input value for Timer 1 to record data as soon as the warning time drops below Timer 1. When warning time is not enabled, the warning time records are set to NULL in the result table.

Additional train handling logic was implemented within TCL (i.e., for simulations where warning time feedback is enabled) that drives the train to a stop once the warning time from the enforcement algorithm drops below the value for Timer 1. TCL still records data for penalty and emergency brake enforcements, if there are any during the simulation, as well as the stopping information once the train came to a stop. However, this data is not needed for the purposes of the analysis conducted in this project.

3.3 Warning Time Input Simulation Analysis and Results

WTI Simulations were run using a Timer 1 value set to 40 seconds and repeated with a Timer 1 value set to 20 seconds. In both cases, the Target Offset value was set to 200 feet. Table 1 shows the resulting deceleration rates for all train simulations in the simulation matrix, separated by the criteria currently used by the railroad network model. To calculate the deceleration rate, the equation in Figure 6 was used.

The simulations were run using the WTI simulation matrix included in Appendix B.

Table 1 shows the deceleration rates calculated using the WTI simulations, using the method described in Section 3.1.

Table 1. PTC Deceleration Rate

Train Type	**WTI Deceleration Rate (ft./sec.^{2}) Timer 1 = 20 sec.**	**WTI Deceleration Rate with Deceleration due to Grade Removed (ft./sec.^{2}) Timer 1 = 20s**	**WTI Deceleration Rate (ft./sec.^{2}) Timer 1 = 40 sec.**	**WTI Deceleration Rate with Deceleration due to Grade Removed (ft./sec.^{2}) Timer 1 = 40 sec.**
Freight	-0.31	-0.45	-0.24	-0.38
Expedited Trains	-0.32	-0.49	-0.24	-0.41

Figure 5 shows deceleration rates from all WTI simulations, grouped by track grade. The analysis of the data showed that the values for deceleration can vary widely. Forces on the train vary by speed, weight of train, track grade, and train length. The plot in Figure 5 includes manifest, intermodal and unit train types, loaded and empty consists, and various locomotive power configurations. A complete list of train types simulated with WTI can be found in Appendix B.

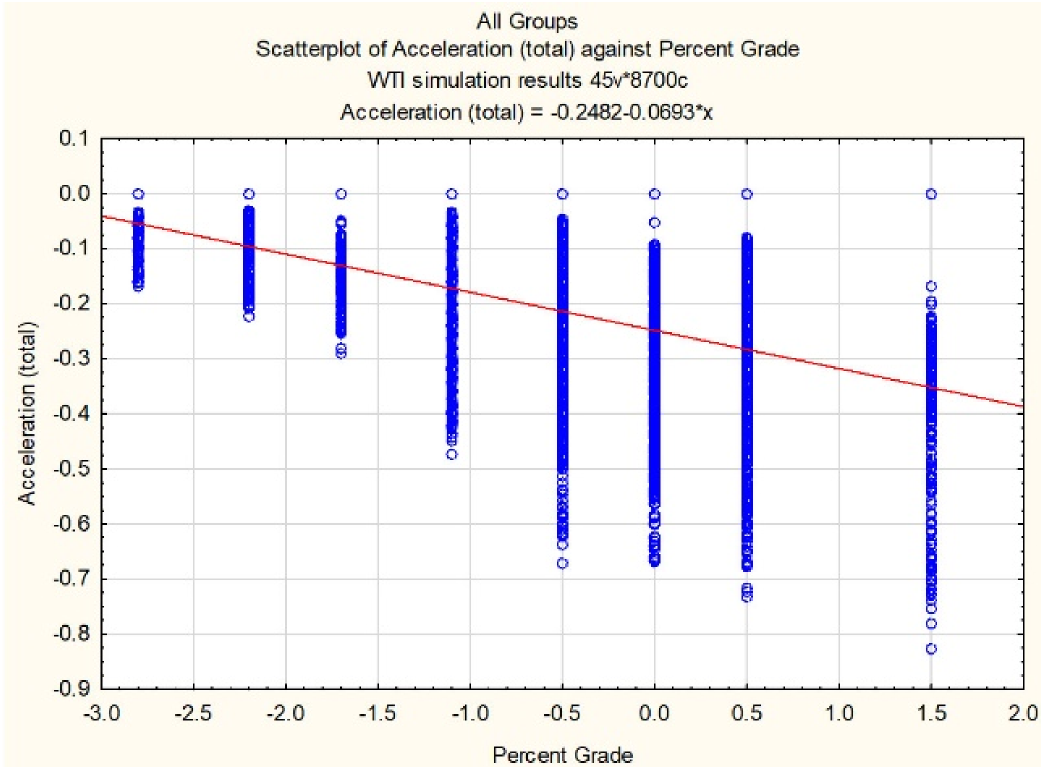


Figure 5. Scatterplot of Deceleration Rates Against Grade from WTI Simulations

Further analysis broke the WTI simulation data down by train type, power configuration, and load status. A regression analysis of the broken-down data sets was completed to develop deceleration rate equations using speed, track grade, train weight, and train length as input variables. Appendix C-C provides detailed results of the regression analysis using the analytics software package Statistica®. The equation in Figure 6 was derived from the regression analysis, with the coefficients for each specific consist type listed in Table 2.

$$Deceleration\ rate = A + B * Initial\ train\ velocity + C * Percent\ grade + D * Total\ train\ weight + E * Train\ Lenth$$

Figure 6. Resulting Deceleration Rate Equation from Regression Analysis

Initial train velocity is in miles per hour, percent grade is a value, total train weight is in pounds, and train length is in feet. The detailed Statistica® analysis is attached in Appendix C-C.

Table 2. Coefficients for Each Train Type Simulated

Coefficients for Determining Acceleration						
Consist Type	**A**	**B**	**C**	**D**	**E**	**Adjusted R^2**
INT HE Empty	-0.258	-0.006	-0.071	-9.884E-09	2.365E-05	0.938
INT HE Loaded	-0.226	-0.004	-0.076	3.757E-09	3.983E-06	0.902
INT DP Empty	-0.246	-0.007	-0.102	-1.762E-08	2.199E-05	0.951
INT DP Loaded	-0.243	-0.005	-0.102	-1.302E-08	2.136E-05	0.929
Unit HE Empty	-0.293	-0.006	-0.088	6.047E-09	1.187E-05	0.947
Unit HE Loaded	-0.328	-0.001	-0.185	-1.322E-09	1.053E-05	0.555
Unit DP Empty	-0.297	-0.007	-0.111	4.678E-09	4.402E-06	0.921
Unit DP Loaded	-0.336	-0.003	-0.182	-6.189E-10	8.129E-06	0.627
Manifest HE	-0.352	-0.006	-0.098	4.811E-09	2.614E-05	0.752
Manifest DP	-0.219	-0.005	-0.100	9.375E-10	4.082E-06	0.919

The adjusted R^2 values in Table 2 show a high confidence in the equations for most of the train consist types. For unit head end (HE) power loaded, unit distributed power (DP) loaded, and manifest HE, the adjusted R^2 values were lower indicating that improvements could be made for these train types. The plots shown for these train types in Appendix C-C do show a bi-modal trend within these data groups. A further breakdown of these consist types may lend to equations with better correlation, but for this project the consists were only broken down to the types shown in Table 2.

4. Braking Algorithm Evaluation Methodology Refinement

In this task, TTCI worked to improve the efficiency of the simulation process and reduce simulation time and labor costs associated with running the simulations for each new release of the PTC braking algorithm. Specifically, TTCI did the following:

- Identified and addressed bottlenecks in the simulation process
- Implemented enhanced automation in batching, executing, and storing results
- Implemented methods for automating analysis of results

TTCI identified bottlenecks in the three pieces of software (e.g., TCL, TOES™, and the PTC enforcement algorithm) used in the simulation process, and also identified bottlenecks in the interfaces between them.

Specifically, TTCI improved bottlenecks during the setup of the simulations, during simulation execution, between the end of one simulation and the beginning of the next, and while writing results to the database. As part of this process, TTCI identified fundamental limitations due to the design and implementation of the individual software packages and differentiated them from bottlenecks relating to the use of these software packages in the PTC Monte Carlo braking algorithm simulation process.

TTCI evaluated the bottlenecks to identify methods for improving the efficiency of the process. Each potential method was evaluated in terms of potential benefit (enhanced efficiency) and difficulty of implementation. Based on this analysis, TTCI developed a list of recommended modifications to implement to enhance the overall Monte Carlo simulation and evaluation process.

Originally, simulations required several hours of setup, monitoring, and troubleshooting whenever there were unexpected glitches in communication between TOES™, TCL and/or the database. Without the physical monitoring of each virtual machine, they would sit idle waiting for the next command, or for an error to be acknowledged.

Data analysis also required extensive hands-on pre-processing to prepare results for reporting. The following is a list of the identified bottlenecks and summaries of the improvements implemented to improve the braking algorithm simulation process.

- **TCL**
  - Inefficiencies in simulation setup and monitoring
    - Developed the capability to setup and start batches of simulations from a single point (multi-batch); wherein, TCL software instances on each server read batch simulation values from a SQL database
    - Initiated automatic e-mail notifications for simulation batches that have failed or simulation batches that are complete
    - Modified TCL to automatically start and resume simulations in cases where the simulation server restarted
  - Limitations in error handling

        - Developed error handling that attempts to automatically rerun simulations that have failed to complete and, if simulations continue to fail, sends e-mail notifications to users
    - Issues when simulations were executed by a user other than the one that entered the TCL settings
        - Modified all TCL simulation settings to write to local the user instead of the current user so the settings would remain the same
    - Efficiency of generating TOES™ consist files
        - Modified architecture so that TOES™ files are generated in common network storage shared by all the simulation servers
- **Logging**
    - The available logged data was insufficient for troubleshooting simulations of interest.
        - Logging data now sent to SQL database for ease of reference
- **Simulation servers**
    - The simulations had to be run on the server where TOES™ files were generated locally.
        - Moved TOES™ files to central network location where each simulation server has access to them
    - It was a time-consuming process for loading and setting up new versions of the enforcement algorithm.
        - Changed from manual configuration on each server to single setup and clone
        - Added the capability to specify emergency brake backup setting in simulation initialization message instead of manually modifying configuration files on each simulation machine
    - The servers went idle waiting for login information after server reboot.
        - Servers set to auto login with common user after restart
- **Analysis**
    - Inefficiency due to manual analysis process
        - Developed standardized data result templates

4.1 TCL Enhancements

TCL was improved through several stages, with each step leading to a more efficient process requiring less human intervention.

Table 3 lists the improvements that were implemented.

Table 3. TCL Enhancements

TCL Improvement	Details
Added error reporting (message box notification)	
Added error handling	Added ability to handle run-time errors
	Added ability to re-run simulation on error
	Added ability to re-run project on error
	Added ability to re-generate TOES™ and consist files on error
Simulation error recovery	Re-run failed sim. If sim fails after re-run, mark sim as failed and move to the next sim
	If multiple sims fail, then fail project and move to next project.
	If multiple project fail stop batch (or run the next batch if in the multi batch processing mode) Send email notification of error to simulation users
Added application level logging	
Added email notification for critical errors and batch completion	Send email notifications of error to simulation users
Added code description and comments	
Added exception recovery from TCL-enforcement algorithm communication failure	
Added communication timeout error between 'TCL and TOES™' and 'TCL and the enforcement algorithm.'	
Added validation for generated TOES™ and Consist files	Notified user if files for simulations are not generated
Automatically closed any active TCL, TOES™ Simulation Engine and SimInit instances on application startup (except instances created by multi batch processing)	
Added fix to close/dispose all open SQL connections after use	
Added message box to notify user when TCL process is already running	
Added ability to save registry files to HKEY_CURRENT_USER and HKEY_USERS for multi batch processing purpose. Registry data is copied on application startup. This is necessary for multi-batch processing.	
Added modification to command file construction	Saved simulation command parameters to database
	Simulation uses Monte Carlo speed and location errors saved in table, if defined, otherwise it generates them
Performed code refactoring	

Error checking was added to allow TCL to restart simulations that stop due to any number of reasons, such as communications lag, invalid initial conditions, or network interruptions. Upon the restart, the error is logged and, after attempting to restart a set number of times, TCL will bypass that simulation and continue to the next simulation. If this error continues with the next simulation and TCL is unable to successfully complete any simulations, it will stop and send out a batch error notification via email to the simulation users. In the past, after one failed attempt, the entire program would stop and wait for user interaction to restart. Since it can take several hours or days to complete an entire batch, this enables simulations to run without frequent monitoring.

Aside from the error handling improvements, TTCI implemented some additional capabilities and improvements within TCL for setup and simulation execution. Figure 7 shows the system parameters for TCL with the additional parameters added to provide the following improvements:

- Emergency Brake Backup – Gives user the capability to indicate whether emergency brake backup is enabled for TCL. TCL will use this to populate the EBB field in the initialization message sent to the enforcement algorithm, and the enforcement algorithm will run the simulation with the appropriate EBB setting. This previously had to be completed by modifying configuration files on the enforcement algorithm every time the user needed to change the setting.
- Emergency Brake Option – Allows the user to determine what type of emergency application TCL will send to TOES™ after the enforcement algorithm requests an emergency. Options include two-way emergency, HE only emergency, and no emergency.
- Log Level – Allows the user to indicate the level of logging for TCL; mainly used for troubleshooting.
- Warning Time Feedback – Parameters added to support warning time feedback simulations, as described in Section 3).

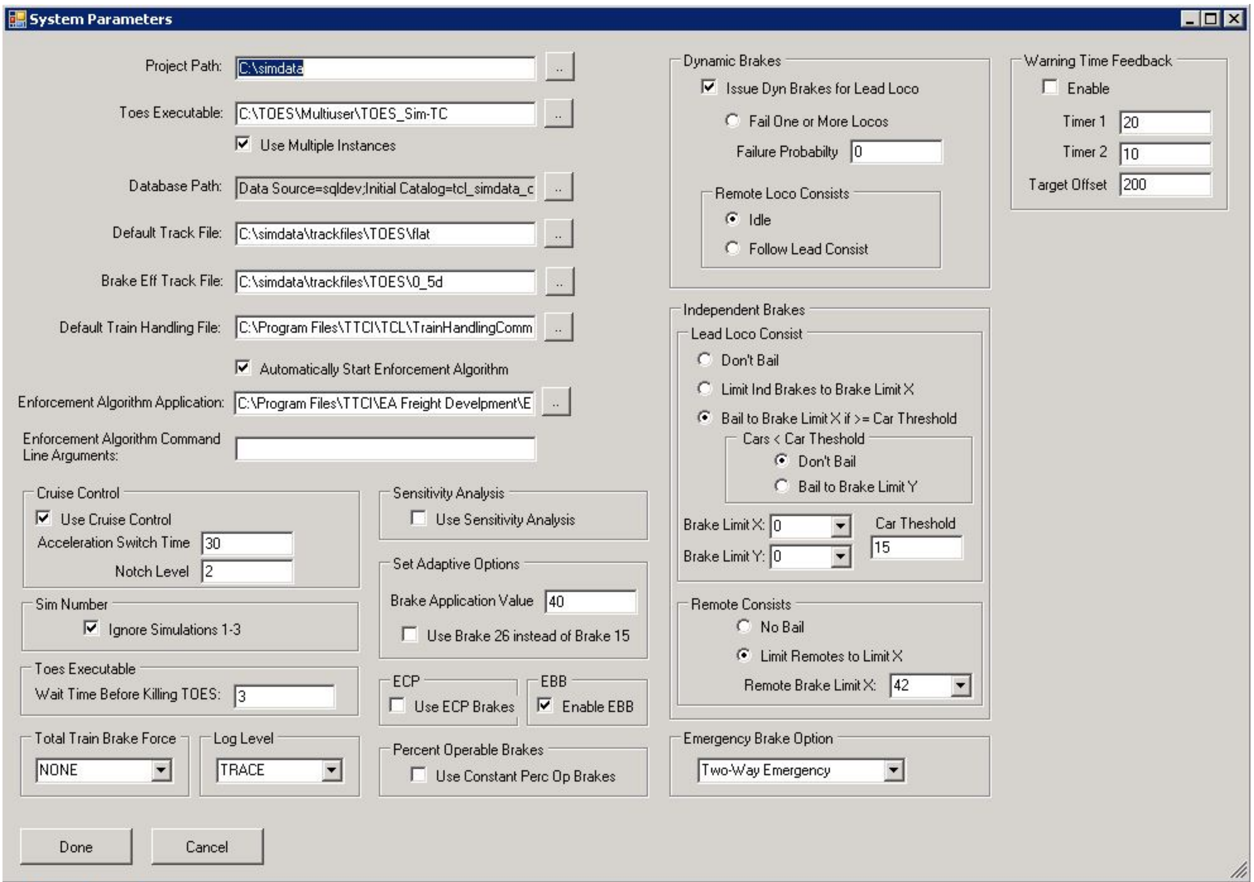


Figure 7. Example of TCL Setup Parameters

TCL was also modified to enable multiple batches to run in a more efficient manner. TCL was previously designed to execute a single batch with a single instance of the software, with user interaction required to start the next batch when the first batch completed. The multi-batch capability introduces a SQL table used to setup and enable batches that are executed by individual instances of the TCL software, which query the table to determine the next batch to start after a batch is completed, when in multi-batch mode.

The new version of TCL, shown in Figure 8, has an option to run in manual mode, with normal user setup of simulations; or auto mode, which uses the multi-batch table in SQL to automatically setup and start simulations. If automatic is selected, upon the virtual machine powering on, TCL will query the SQL multi-batch table for prepared batches of simulation sets. If a batch is enabled within the multi-batch table, all the TCL parameter settings, shown in Figure 7, for that batch will be configured within the multi-batch table and TCL will automatically configure itself to those settings and start the simulations. After completing each batch of simulations, TCL will again query the SQL database, and if another batch is enabled it will setup accordingly and run simulations. This will continue until all enabled batches are complete. This allows for simulations to continue after a power outage or virtual machine reboot, as well as running without user intervention. The multi-batch TCL is configured in the SQL table using a SQL script prior to starting a batch of simulations. Future improvements to multi-batch may include a graphical user interface (GUI) to simplify batch configurations.

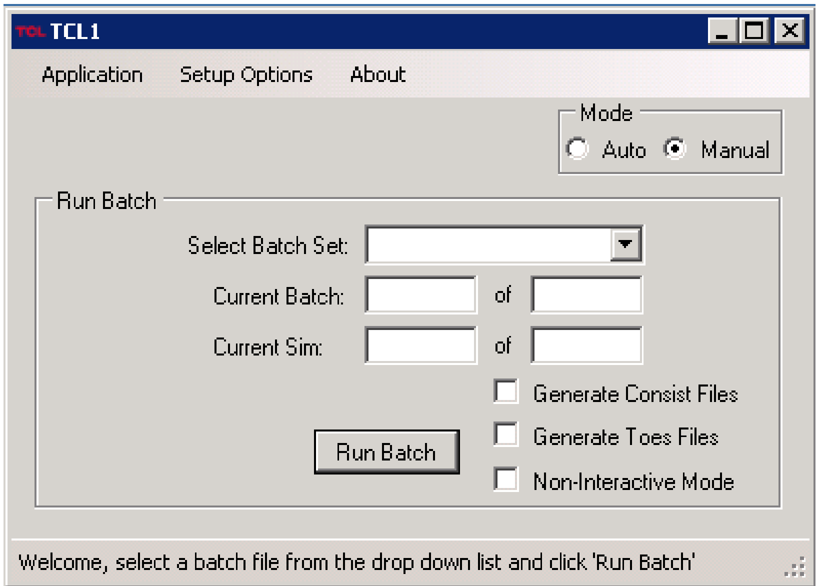


Figure 8. TCL with Auto/Manual Modes for Multi-Batch Processing

For ease of setting up and monitoring the simulations, the multi-batch table has database flags showing which simulations are currently queued, running, or complete and TCL is configured to use a general login to allow any TCL user to see the program running. This assists with eliminating confusion and human error by allowing each user to clearly identify which servers are in use as well as monitor progress of a simulation batch.

In addition to the common login, the simulation matrix was also redistributed to take advantage of the ability to automatically start, with larger batches being broken down to run concurrently on more virtual machines. This improvement maximizes efficiency and improves on the required time to run large batches.

4.2 Logging Enhancements

Capturing logs of TOES™ consist file and simulation file parameters in a database allows for increased efficiency when specific simulation input values need to be recovered. For example, this may be of value when simulation analysis shows unexpected results that require investigation and the analyst is interested in details of the consist and simulation parameters used in the simulation. Below is an overview of the information logged for each consist file generated, and each simulation created.

- Consist file logging:
  - Consist name
  - Locomotive weight
  - Tare weight
  - Total car weight (tare and load)
  - Locomotive brake force
  - Car brake force

- Train length

- Simulation file logging:
  - Simulation name
  - Brake pipe pressure
  - Brake pipe pressure leakage
  - Cars with brakes cut-out
  - Ambient pressure and temperature
  - Coefficient of friction
  - Speed error
  - Location error
  - Head of train brake pipe pressure error
  - Rear of train brake pipe pressure error
  - Track grade error

4.3 Server Enhancements

Historically, the braking algorithm simulation servers were configured as stand-alone systems. Each system was capable of running braking algorithm simulations for a subset of freight braking train types and grades that had files for those simulations loaded on the local machine. To enhance the capabilities and speed in which an entire Monte Carlo simulation set can be executed, the servers were reconfigured to maximize speed and efficiency of the freight braking simulations. The following server enhancements were made:

- Centralized network data storage
- Increased storage capacity
- Configured servers to automatically login in the event of system restart

In the current configuration braking algorithm simulations, there are 25 virtual simulations servers. These virtual servers share a common hardware platform and use virtual server software to divide the physical server hardware into many virtual servers that can run independent processes from each other. This configuration allows for up to 25 instances of TCL to run at a time. To allow all 25 virtual servers access to the TOES™ data files, as well as a common storage, all the freight braking simulation servers were configured to use a central networked data storage area. This data storage area is used to store the TOES™ configuration files needed for simulations as well as the data output from the simulations. By combining all the TOES™ train configuration files in one common storage area, any of the simulation servers can run any assigned simulation at any time. Prior to this enhancement, servers were only able to run simulations loaded on the local hard drive of the server, which slowed processing time and efficiency.

The centralized storage area is also used to store the output TOES™ data files for each simulation. Although the simulation output data values are stored on a SQL server, these TOES™ files are sometimes required during post-simulation data analysis. By combining all the output data in a single location, the efficiency in accessing the TOES™ files, if they are needed during post-simulation analysis, is increased.

To minimize the time required to run simulations, as well as allow for multiple simulation types to be run, the virtual servers can be cloned in the future to allow for more capacity.

The network storage is built on a Network Addressable Storage (NAS) solution. NAS allows for multiple computers and servers to access the same storage simultaneously via a TCP/IP network. This allows for all the virtual servers to be setup with TCP/IP addresses in the same network subnet and access the common NAS storage at any time. The NAS is an expandable solution that allows storage growth as more virtual simulation servers are deployed. Figure 9 shows the NAS and simulation network configuration.

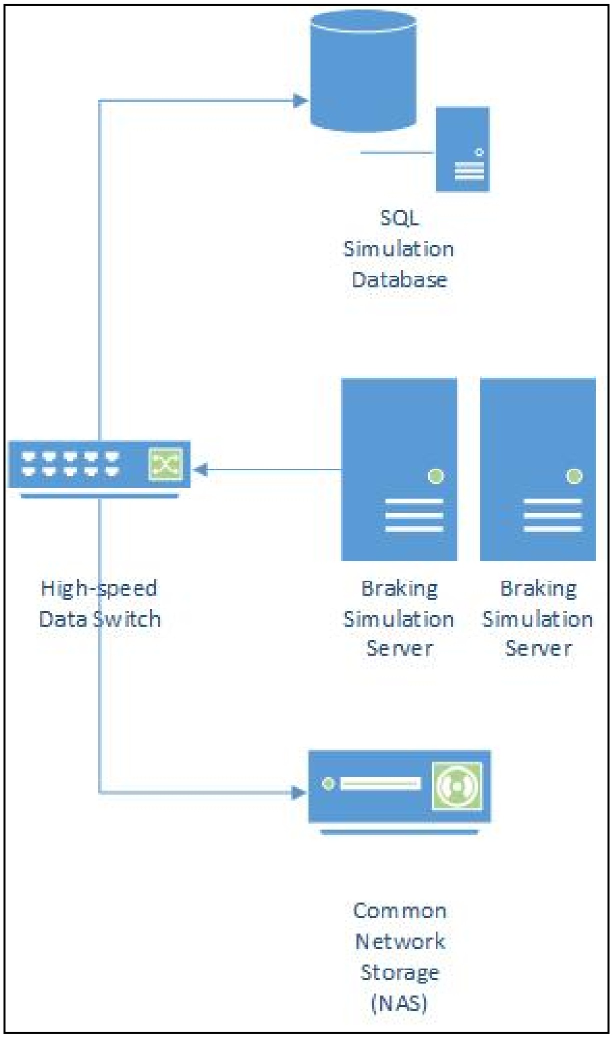


Figure 9. Freight Braking Simulation Server Configuration

Automatic login is another enhancement to the simulation servers. Previously, when servers were restarted, the simulation processing ceased and the server went idle until a user was able to login and resume simulations. The freight braking simulation servers are now setup to login automatically and continue TCL queries from the multi-batch SQL database for the next simulation to start. The addition of automatic login increases efficiency of running a braking algorithm simulation set by assuring servers can login and continue working in the event of a system failure and/or restart. The changes made to the freight braking server configurations in conjunction with TCL and multi-batch improvements minimizes simulation server down time and maximizes efficiency while running simulations.

4.4 Automation of Analysis

When the braking algorithm simulation evaluation process was established, a procedure was developed that used Statistica® to process the resulting simulation data and present the safety and performance metrics. The analysis has changed to include more parameters and other information, which has led to the process becoming tedious and time-consuming. Manually processing the data also led to increased opportunities for user errors. To improve this process, SQL queries were created that filter out extraneous data such as incomplete simulations, duplicate simulations, and errors caused by the programs and simulations that do not meet the correct operating parameters. Additionally, templates for displaying the data were developed to create consistent results, and the data analysis process was reviewed for efficiency. Upon review, TTCI made minor changes to improve the data processing required as part of analysis and s determined that completely automating data processing was not necessary. With the improvements made, analysis that took several days can now be completed in a few hours.

The templates cover both the overall results for each train type, as well as individual results. Overall results are most commonly used to show that safety and performance metrics are met, while the individual results allow for more in-depth analysis by train consist, track profile, and speed, and allows the user to identify possible areas of interest. Combined with the logging enhancements described in Section 4.2, exploratory data analysis can be more easily performed and issues can be quickly resolved. Figure 10 shows an example analysis template.

1/2. Overall Probability of Stopping Short of the Target & 99.5-Percentile Distance:

a. For All Slopes (Combined):

OverRun Value (ft.)	-52	-51	-50	-2	-1	0	1	2	3
Count (No. of Cases)	46	28	48	1	1	1	2	1	1
Cumulative Count	86494	86522	86570	86948	86949	86950	86952	86953	86954
Percentage	0.052874	0.032184	0.055173	0.001149	0.001149	0.001149	0.002299	0.001149	0.001149
Cumulative Percentage	99.4193	99.4515	99.5067	99.9412	99.9423	99.9435	99.9458	99.9469	99.9481
100% - Cum. Percentage	0.6336	0.5807	0.5485	0.0599	0.0588	0.0576	0.0565	0.0542	0.053

Approx. 99.94% of all cases stop at or before the target

Equivalent to the 99.5%, approximately 51 feet before target

Approx. 0.06% fail in stopping before the target position (i.e., prob. of overrun)

b. For Maj. Slopes (1.1d-0.5i):

OverRun Value (ft.)	-68	-67	-66	-2	-1	0	1	2	3
Count (No. of Cases)	39	24	15	1	1	1	2	1	1
Cumulative Count	68615	68639	68654	68956	68957	68958	68960	68961	68962
Percentage	0.056522	0.034783	0.021739	0.001449	0.001449	0.001449	0.002899	0.001449	0.001449
Cumulative Percentage	99.442	99.4768	99.4985	99.9362	99.9377	99.9391	99.942	99.9435	99.9449
100% - Cum. Percentage	0.6145	0.558	0.5232	0.0652	0.0637	0.0623	0.0609	0.0579	0.0565

Figure 10. Sample of Data Analysis Template

5. Conclusion

TTCI worked closely with the AG to review and update the Monte Carlo simulation matrix. TTCI also identified bottlenecks in the current freight braking algorithm simulation process. Once bottlenecks were identified, TTCI engineers developed methods to reduce or mitigate them. Strategies to increase the efficiency of the simulation process were developed by TTCI engineers and software programmers. These improvements included upgrades to TCL software, multi-batch, simulation log management, and server and network enhancements.

In addition to the improvements made to the simulation process, TTCI also incorporated warning time input into the simulation process that allows for braking algorithm simulations to be run to establish train deceleration rates with PTC operations. This data can be used with railroad network models to evaluate the operational impact of the PTC braking algorithm from the perspective of the entire rail network, as opposed to a single train.

6. References

1. Brosseau, J., Moore Ede, B., Pate, S., Wiley, R. B., and Drapa, J., (October 2009), Development of an Operationally Efficient PTC Braking Enforcement Algorithm for Freight Trains, DOT-FRA-ORD-13/34, Washington, DC: U.S. Department of Transportation, Federal Railroad Administration.

Appendix A.
PTC Braking Test Matrix

Consist Information						
Consist Name	Type	Description	Length (ft)	Makeup	# of Loaded	# of Empty
IVEADM	Intermodal	Very Long	15868	A	0	74
IVEBDM	Intermodal	Very Long	15842	B	0	83
IVECDM	Intermodal	Very Long	15777	C	0	74
IVEDDM	Intermodal	Very Long	15910	D	0	76
IVEEDM	Intermodal	Very Long	15747	E	0	76
IVLADM	Intermodal	Very Long	16372	A	79	0
IVLBDM	Intermodal	Very Long	15772	B	68	23
IVLCDM	Intermodal	Very Long	15874	C	74	15
IVLDDM	Intermodal	Very Long	15838	D	66	10
IVLEDM	Intermodal	Very Long	15758	E	62	19
ILEADE	Intermodal	Long	10666	A	0	46
ILEADM	Intermodal	Long	10666	A	0	46
ILEBDE	Intermodal	Long	10536	B	0	50
ILEBDM	Intermodal	Long	10536	B	0	50
ILECDE	Intermodal	Long	10759	C	0	49
ILECDM	Intermodal	Long	10759	C	0	49
ILEDDE	Intermodal	Long	10645	D	0	42
ILEDDM	Intermodal	Long	10645	D	0	42
ILEEDE	Intermodal	Long	10722	E	0	57
ILEEDM	Intermodal	Long	10722	E	0	57
ILLADE	Intermodal	Long	10934	A	50	0
ILLADM	Intermodal	Long	10934	A	50	0
ILLBDE	Intermodal	Long	10578	B	46	9
ILLBDM	Intermodal	Long	10578	B	46	9
ILLCDE	Intermodal	Long	10668	C	43	7
ILLCDM	Intermodal	Long	10668	C	43	7
ILLDDE	Intermodal	Long	10667	D	38	10
ILLDDM	Intermodal	Long	10667	D	38	10
ILLEDE	Intermodal	Long	10622	E	49	13
ILLEDM	Intermodal	Long	10622	E	49	13
IMEADE	Intermodal	Medium	8064	A	0	43
IMEAHE	Intermodal	Medium	8064	A	0	43
IMEBDE	Intermodal	Medium	7963	B	0	46
IMEBHE	Intermodal	Medium	7963	B	0	46
IMECDE	Intermodal	Medium	7909	C	0	38
IMECHE	Intermodal	Medium	7909	C	0	38
IMEDDE	Intermodal	Medium	7899	D	0	39
IMEDHE	Intermodal	Medium	7899	D	0	39
IMEEDE	Intermodal	Medium	7909	E	0	38
IMEEHE	Intermodal	Medium	7909	E	0	38
IMLADE	Intermodal	Medium	8290	A	48	0
IMLAHE	Intermodal	Medium	8290	A	48	0
IMLBDE	Intermodal	Medium	7972	B	35	9
IMLBHE	Intermodal	Medium	7972	B	35	9

Figure A1. PTC Braking Test Matrix

Appendix B.
Warning Time Input Test Matrix

Train Type	Track Grade	Brake Set PSI	Target Stop Position Feet
Int_LongDE_WTI	flat.trk		30680
Int_LongDE_WTI	flat.trk		28040
Int_LongDE_WTI	flat.trk		26280
Int_LongDE_WTI	0_5i.trk		30680
Int_LongDE_WTI	0_5i.trk		28040
Int_LongDE_WTI	0_5i.trk		26280
Int_LongDE_WTI	1_5i.trk		27600
Int_LongDE_WTI	0_5d.trk		30680
Int_LongDE_WTI	0_5d.trk		28040
Int_LongDE_WTI	0_5d.trk		26280
Int_LongDE_WTI	1_1d.trk		70900
Int_LongDE_WTI	1_1d.trk		63500
Int_LongDE_WTI	1_1d.trk		61400
Int_LongDE_WTI	1_7d.trk		63200
Int_LongDE_WTI	2_2d.trk		63300
Int_LongDE_WTI	2_8d.trk		66400
Int_LongDE_WTI	flat.trk		30680
Int_LongDE_WTI	flat.trk		28040
Int_LongDE_WTI	flat.trk		26280
Int_LongDE_WTI	0_5i.trk		30680
Int_LongDE_WTI	0_5i.trk		28040
Int_LongDE_WTI	0_5i.trk		26280
Int_LongDE_WTI	1_5i.trk		27600
Int_LongDE_WTI	0_5d.trk		30680
Int_LongDE_WTI	0_5d.trk		28040
Int_LongDE_WTI	0_5d.trk		26280
Int_LongDE_WTI	1_1d.trk		70900
Int_LongDE_WTI	1_1d.trk		63500
Int_LongDE_WTI	1_1d.trk		61400
Int_LongDE_WTI	1_7d.trk		63200
Int_LongDE_WTI	2_2d.trk		63300
Int_LongDE_WTI	2_8d.trk		66400
Int_LongDE_WTI	flat.trk		30680
Int_LongDE_WTI	flat.trk		28040
Int_LongDE_WTI	flat.trk		26280
Int_LongDE_WTI	0_5i.trk		30680
Int_LongDE_WTI	0_5i.trk		28040
Int_LongDE_WTI	0_5i.trk		26280
Int_LongDE_WTI	1_5i.trk		27600
Int_LongDE_WTI	0_5d.trk		30680
Int_LongDE_WTI	0_5d.trk		28040
Int_LongDE_WTI	0_5d.trk		26280
Int_LongDE_WTI	1_1d.trk		70900
Int_LongDE_WTI	1_1d.trk		63500
Int_LongDE_WTI	1_1d.trk		61400

Figure B1. Warning Time Input Test Matrix

Enforcement Algorithm Evaluation Process Overview

and

Communications Interface Specification

Revision 7

November 1, 2018

Transportation Technology Center, Inc.
A subsidiary of the Association of American Railroads

Modification Log

Description	Date
Revision 2 – First draft	June 2010
Revision 3 – Changes to termination logic	August 24, 2010
Revision 4 – Formatting and restructuring; added data message specification and field testing overview	September 13, 2010
Revision 5 – Added target speed to Init message	September 15, 2010
Revision 6 – Added description of installation test procedures in Appendix C-B	January 25, 2011
Revision 7 – Changed CRC bytes in Init message and enforcement algorithm (EA) status message to add needed fields	November 1, 2018

Document Description

This document describes the concept of operations for the evaluation of PTC braking EA software in both a simulation and field test environment. The document also includes interface protocol specifications for the integration of supplier provided EA software into the TTCI testing environment.

Concept of Operations

This section describes the concept of operations for enforcement algorithm evaluation in both a simulation and field test environment.

Simulation Testing

This section describes the simulation test process and required interfaces. Figure C1 illustrates the simulation testing process flow. To start the process, TCL is configured to execute a batch of simulations. The EA application is started and configured to communicate with TCL and EA-Init using a specified IP address and two distinct ports. The simulation testing then proceeds as follows:

1. TCL starts EA-Init and TOES™ at the beginning of each simulation
2. EA-Init sends an initialization message to EA over TCP/IP using the admin port
3. EA sends a status message to TCL over TCP/IP using the data port
4. TCL propagates the TOES™ simulation by 1 second, receives train status data and sends this data to EA over TCP/IP using the data port
5. Steps 3 and 4 are repeated until EA determines a penalty brake application is necessary. At that time EA updates the status code in the status message sent in Step 3 to instruct TCL to apply the penalty brake. TCL then initiates the penalty application in TOES™ and steps 3 and 4 continue until the train speed is less than 0.5 mph.
6. EA sends a terminate message to both TCL—using the data port—and EA-Init—using the admin port
7. EA-Init shuts down and TCL proceeds with the next test until the end of the test batch

The TCL software can run multiple simulations on a single test machine. For this reason, the supplier EA software should have the ability to set both the admin port and data port using configuration files.

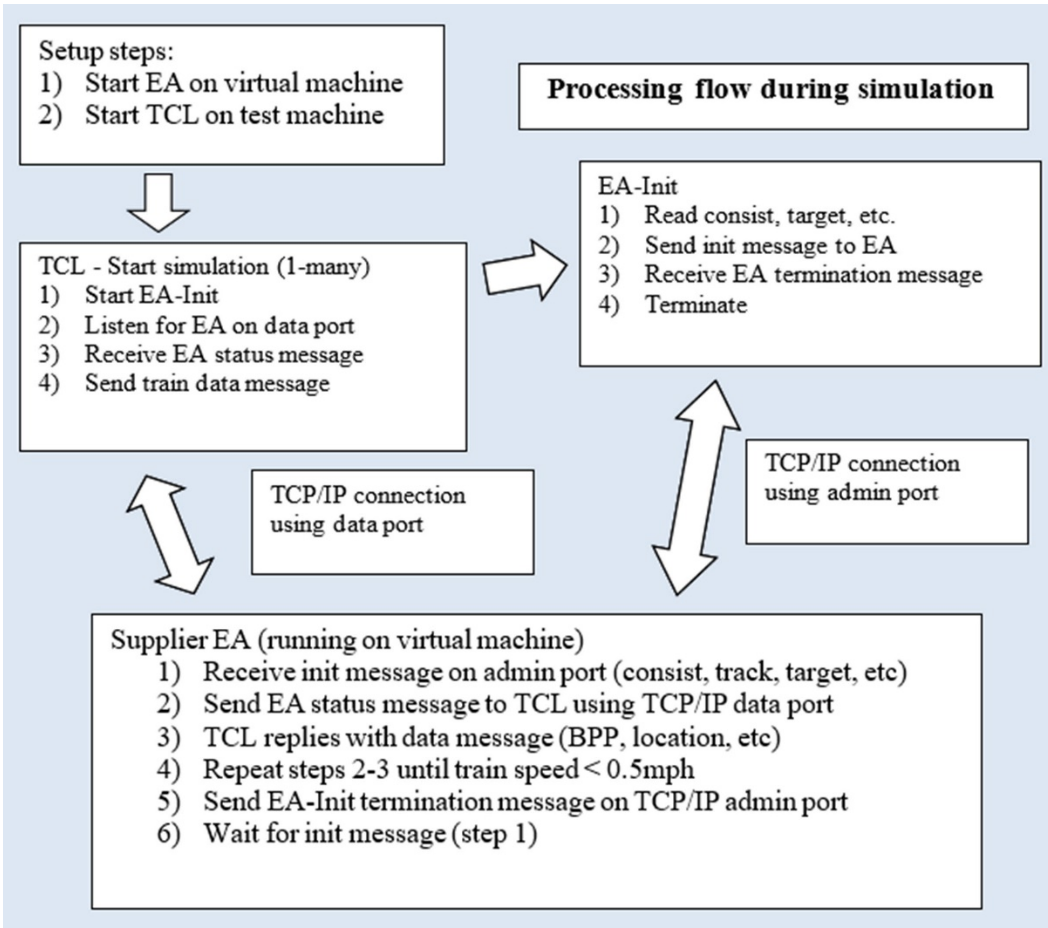


Figure C1. Simulation Test Process Flow

Field Testing

This section describes the field test process and required interfaces. The general process flow for field testing is designed to be very similar to simulation testing and the interfaces are identical. Figure C2 illustrates the process flow for field testing. The primary difference between field and simulation testing is that, during field testing, the EA software and the EA-Init application reside on a test computer that is connected through an Ethernet cable to the locomotive onboard computer (OBC). As in simulation testing, the EA is started and configured to interface the EA-Init application and the locomotive OBC through a specified IP address and two distinct ports.

The EA-Init application is then started and used to send an initialization message to the EA software over TCP/IP using the admin port. Once initialized, EA sends a status message to the locomotive OBC application over TCP/IP using the data port. The test is then run, with the locomotive OBC application sending data to the EA software at 1 Hz frequency and the EA software responding with a status message using the data port. When the EA software determines a penalty application is necessary, it sends the appropriate status message to the locomotive OBC, which then initiates the penalty application on the train. When the train comes to a stop, the EA software sends a terminate message to the locomotive OBC—using the data port—and to the EA-Init application—using the admin port.

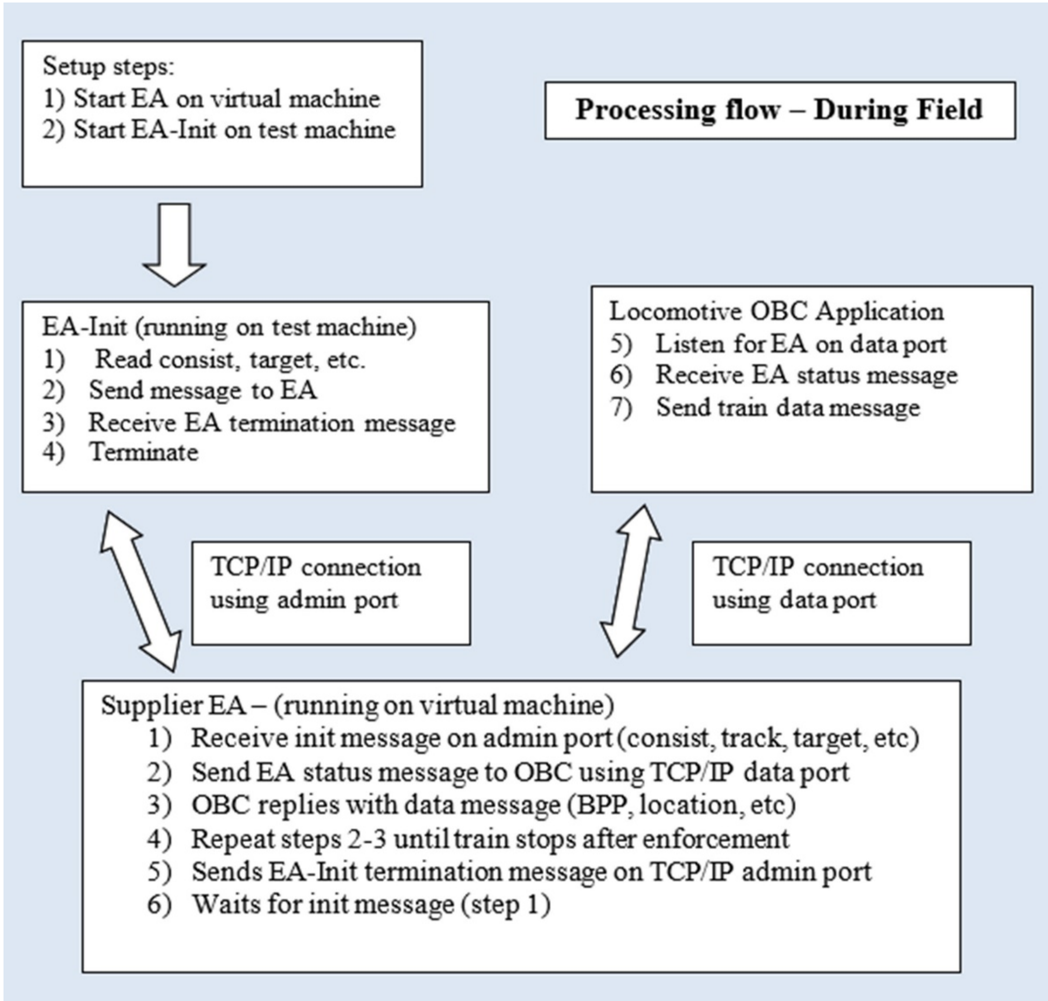


Figure C2. Field Test Process Flow

Track Data

TTCI and the EA supplier will coordinate the development of track data that will be used by the supplier-provided EA software. TTCI will provide track profile data for each track section that will be utilized in testing. The supplier will use this track profile data to generate the track data store to be used by their EA software. Specific track sections for each individual test will be identified in the initialization message using an agreed upon identifier.

Machine Configuration

Supplier-provided EA software must be delivered in one of three forms:

- As a virtual machine image that can be run on the test machines
- As a software executable that can run on the test machines
- As hardware that can be installed in the TTCI test environment (note that for simulation testing, multiple simulations are planned to be run concurrently.)

The current test machines run the Microsoft® Windows XP operating system with 4 GB of RAM. TTCI and the EA supplier shall create a mutually agreeable machine configuration for running the provided EA software.

Protocol Test Application

TTCI will provide a protocol test application for the EA supplier to use in development of software that can communicate using protocols developed by TTCI (see Appendix C-A).

Interface Specifications

This section specifies the format for the various messages used in the enforcement algorithm evaluation processes described in the previous section.

Initialization Message Specification

Table C1 specifies the format for the initialization message to be sent from the EA-Init application to the supplier's EA application at the beginning of each simulation and field test.

Table C1. Initialization Message

Field Name	Description	Data Length	Data Type	Notes
START_BYTES	Bytes for framing	2 bytes	21,930 (0x55aa)	Static
MESSAGE_ID	Message identifier	1 byte	3 (0x03)	Static
TRACK_FILE_ID	The track file number	2 bytes	unsigned short	None
TARGET_LOCATION	The target stopping location (footage)	4 bytes	unsigned Integer	None
TARGET_SPEED	The target speed (mph)	1 byte	Unsigned integer	None
START_LOCATION	The initial starting track location (in feet)	4 bytes	Unsigned Integer	None
TRAIN_TYPE	Train type 0 – Unknown 1 – General freight 2 – Unit freight 3 – Intermodal 4 – Passenger 5 – High speed passenger 6 – Tilt train	1 byte	UNSIGNED INTEGER	0–6
ORIENTATION	Lead loco orientation 0 – Unknown 1 – Front 2 – Back	1 byte	UNSIGNED INTEGER	0–2
TRAILING_TONS	Trailing tonnage (cars only)	2 bytes	unsigned short	0–30,000
CARS_NO_BRAKES	Number of cars with inoperative brakes	2 bytes	unsigned short	0–999
AXLES	Number of axles (cars and locomotives)	2 bytes	unsigned short	0-3,996
TOTAL_LENGTH	Train length (feet) – including locomotives	2 bytes	unsigned short	60–15,000
LOADS	Loaded car count	2 bytes	unsigned short	0–999
EMPTIES	Empty car count	2 bytes	unsigned short	0–999
CAR_BRAKE_FORCE	Car braking force (lbs.) (optional)—not including locomotives	4 bytes	unsigned integer	0–2,000,000

Field Name	Description	Data Length	Data Type	Notes
LOCOMOTIVES	The number of locomotives	1 byte	UNSIGNED INTEGER	0–24
For each Loco				
POSITION	The locomotive position in the train	2 bytes	unsigned short	0–999
TONNAGE	The tonnage of the locomotive	2 bytes	unsigned short	20–300
STATUS	Locomotive Status 0 – Unknown 1 – Run 2 – Isolated	1 byte	UNSIGNED INTEGER	0–2
LENGTH	The length of the locomotive (feet)	1 byte	UNSIGNED INTEGER	60–90
HORSEPOWER	Locomotive horsepower	2 bytes	unsigned short	0–10,000
End For				
Spare		3 byte		
Emergency Brake Backup	0 – False 1 - True	1 bytes	UNSIGNED INTEGER	0–1
END_BYTES	Bytes for Framing	2 bytes	30,875 (0x789b)	Static

Replace the CRC 4 bytes with a byte used for determining if emergency brake backup is set to true or false and 3 spare bytes.

The TRACK_FILE_ID field identifies the section of track according to an agreed upon identifier.

The TARGET_LOCATION field specifies the target stop position in feet from the beginning of the track section for the simulation. The track section for the simulation is defined in the track file indicated by the TRACK_FILE_ID field, as discussed above.

The CAR_BRAKE_FORCE field is an optional input designed for cases when the railroad customer plans to supply the enforcement algorithm with a total train braking force that is calculated offline by a preprocessor. In these cases, the railroad or EA supplier can provide the algorithm for calculating the total train braking force and this field can be populated. Otherwise, this field can be ignored.

Train Data Message Specification

Table C2 specifies the format for the train data message that is sent to the EA software. This message is sent from the TCL application during simulation testing and from the locomotive OBC application during field testing. In simulation testing, this will occur at 1 Hz frequency simulation time (i.e., faster than real time) and in field testing, this will occur at 1 Hz frequency real time.

Table C2. Train Data Message

Field Name	Description	Data Length	Data Type	Notes
START_BYTES	Bytes for framing	2 bytes	21,930 (0x55aa)	Static
TRN_LOC	Current train location (footage)	8 bytes	Double	Sent as feet, must be within limits defined in track data file
TRN_SPD	Current train speed (mph)	8 bytes	Double	0 to 999.99 mph
BPP_HEAD	Current brake pipe pressure at head of train (psi)	8 bytes	Double	Range from 0 to 999.99
BPP_END	Current brake pipe pressure at End-of-Train (psi)	8 bytes	Double	Range from 0 to 999.99
NOTCH	Current locomotive throttle position	8 bytes	Double	0–8
DYN_BRAKE_V	Dynamic braking voltage	8 bytes	Double	0 to 80V
HW_DISC1	Hardware discrete byte 1 • Bit A: TL01 - Slow Speed • Bit B: TL03 - Throttle D • Bit C: TL06 - Generator Field • Bit D: TL07 - Throttle C • Bit E: TL08 - Fwd Ctl • Bit F: TL09 - Rev Ctl • Bit G: TL10 - Wheel Slip • Bit H: TL12 - Throttle B	1 byte	Byte	HGFEDCBA (LSB) 1 = High 0 = Low
HW_DISC2	Hardware discrete byte 2 • Bit A: TL15 - Throttle A • Bit B: TL16 - Engine Run • Bit C: TL17 - Dyn Brake Setup • Bit D: TL21 - Dyn Brake Circuit Active • Bit E: TL05 - Emg Sand • Bit F: Alternator (Engine Running) • Bit G: TL23 Sand • Bit H: ISOLATE	1 byte	Byte	HGFEDCBA (LSB) 1 = High 0 = Low

Field Name	Description	Data Length	Data Type	Notes
HW_DISC3	Hardware Discrete Byte 3 - (spare) • Bit A: (NOT SUPPLIED) • Bit B: (NOT SUPPLIED) • Bit C: (NOT SUPPLIED) • Bit D: (NOT SUPPLIED) • Bit E: (NOT SUPPLIED) • Bit F: (NOT SUPPLIED) • Bit G: (NOT SUPPLIED) • Bit H: Brakes Cut Out	1 byte	Byte	HGFEDCBA (LSB) 1 = High 0 = Low
SPARE	(not used)	1 byte	Byte	Not used
CRC 32	CRC32 over data (not required in V3.4)	4 bytes	UNSIGNED INTEGER32	Not used
END_BYTES	Bytes for framing	2 bytes	30875 (0x789b)	Static

EA Status Message Specification

Table C3 specifies the format for the EA status message. This message is sent by the EA software to the TCL application (e.g.., simulation testing) or the locomotive OBC application (e.g., field testing) once at the beginning of the test and then again after each time a train data message is received.

Table C3. EA Status Message

Field Name	Description	Data Length	Data Type	Notes
START_BYTES	Bytes for framing	2 bytes	Byte (0x55aa)	Static
STATUS	Health status 00 – OK 01 – Error 02 – Completed	2 bytes	short	Values 0 thru 2
APPLY_BR	Apply service brake	1 byte	Boolean	0 – false 1 – true
APPLY_EB	Apply emergency brake	1 byte	Boolean	0 – false 1 – true
Spare		3 bytes		
Warning Time	Enforcement warning time	1 byte	UNSIGNED INTEGER8	0–255

Field Name	Description	Data Length	Data Type	Notes
END_BYTES	Bytes for framing	2 bytes	30,875 (0x789b)	Static

Replace the CRC 4 bytes with a byte used for enforcement warning time and 3 spare bytes.

Appendix C-A:
Protocol Test Application

The protocol test application is provided to EA developers to assist in the development of interfaces to the TCL and locomotive OBC software. The protocol test application has the following features:

- Simulates TCL/locomotive OBC inputs
- Uses current TTCI EA protocol specifications
- Allows the user to test input values
- Sends sample initialization message to EA software

The Microsoft® Visual C# 2008 source code for this application will be provided to the EA supplier to assist in development and testing.

The following two figures illustrate the operation of the test application. The first shows the train data message screen and the second shows the initialization message screen.

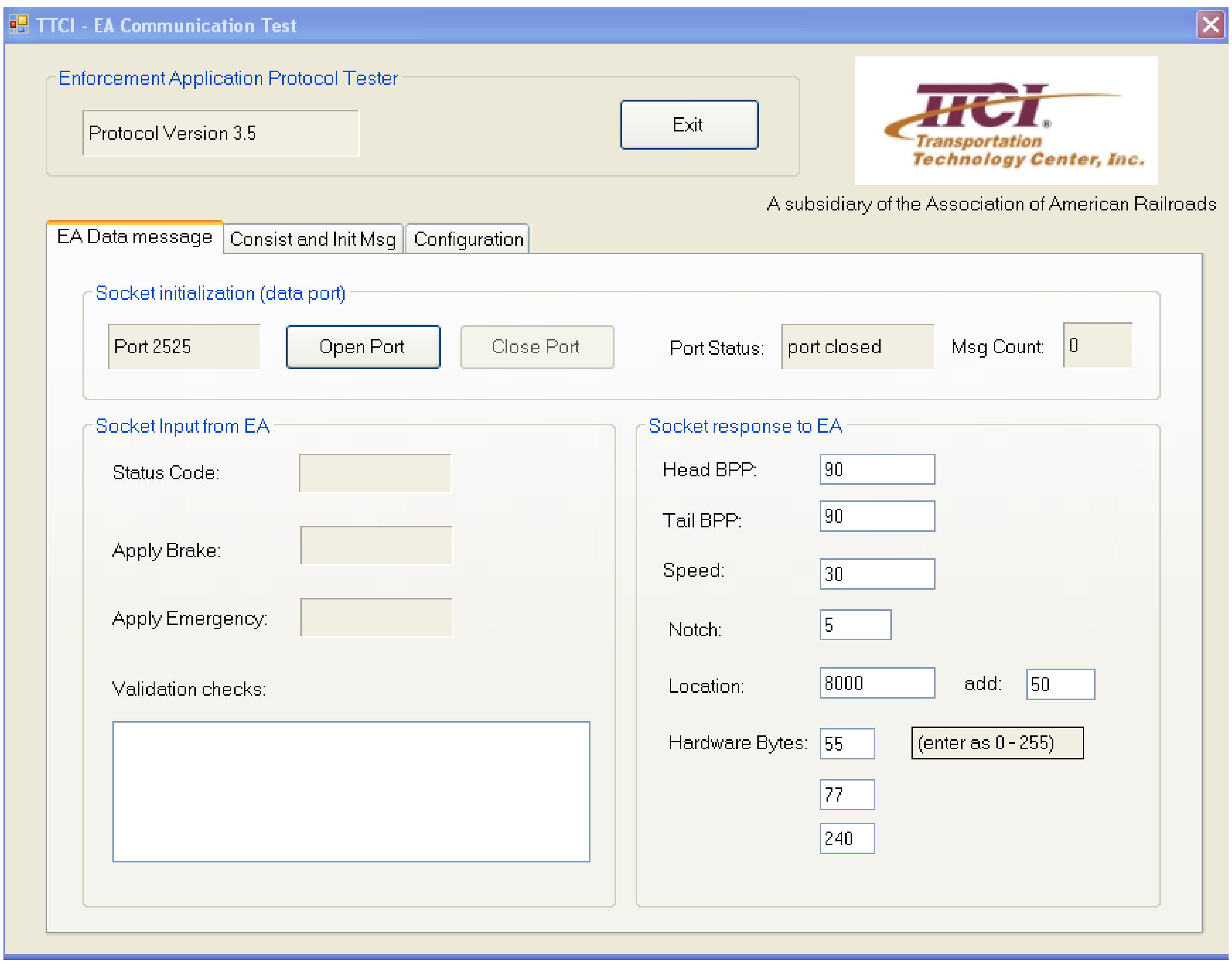


Figure C3. EA Protocol Test Application—Data Message Tab

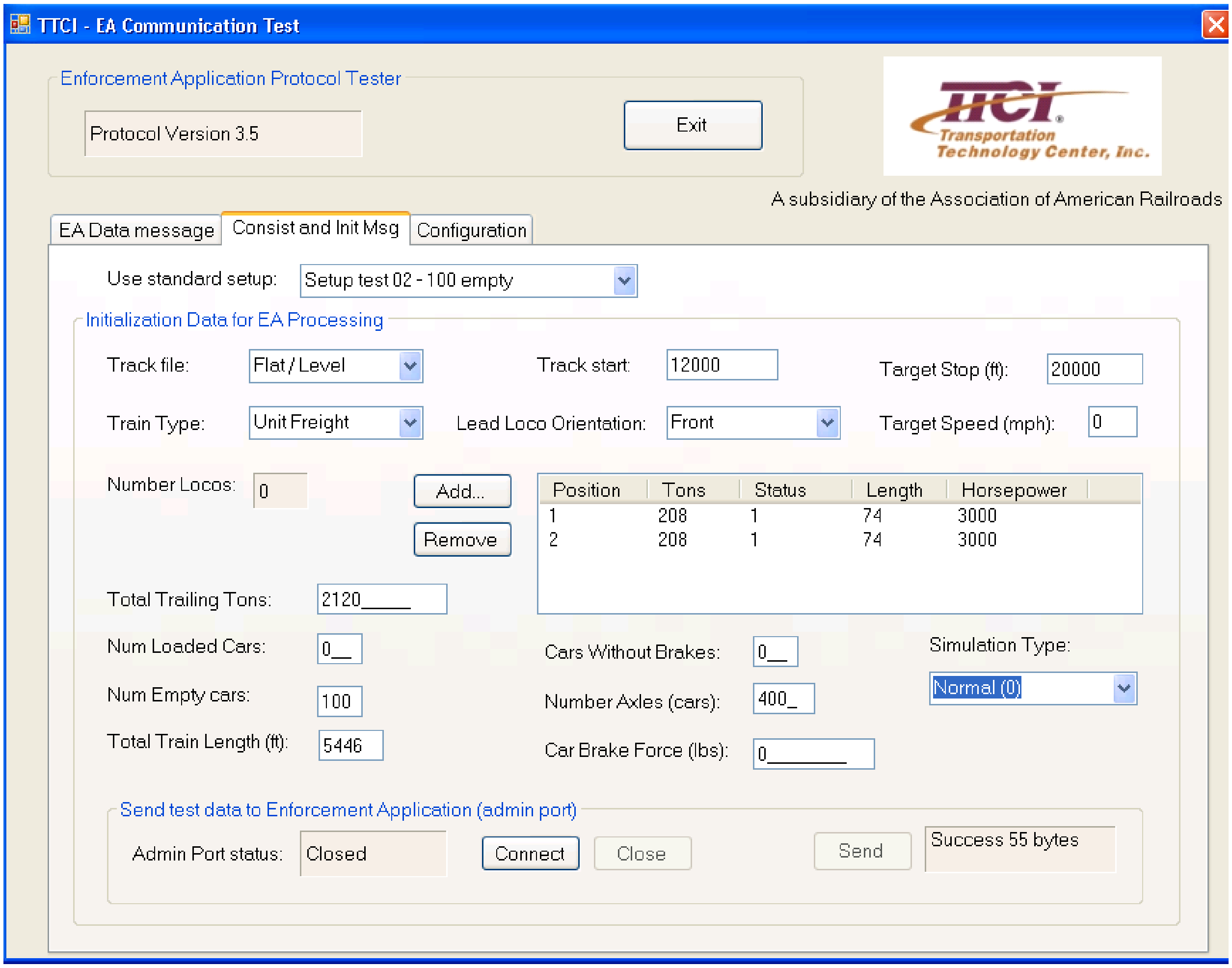


Figure C4. EA Protocol Test Application—Initialization Message Tab

Appendix C-B:
Installation and Setup Testing

This section describes how the protocol test application is used to validate the machine setup and to ensure that the EA software is installed and configured properly. The process is described as follows:

1. There are several test scenarios described in this section. These scenarios match test scenarios in the TTCI simulation environment.
2. Using the protocol test app, the input parameters are entered by selecting a setup test using the EA Communications Test application. This causes the loading of parameters to the screen fields:

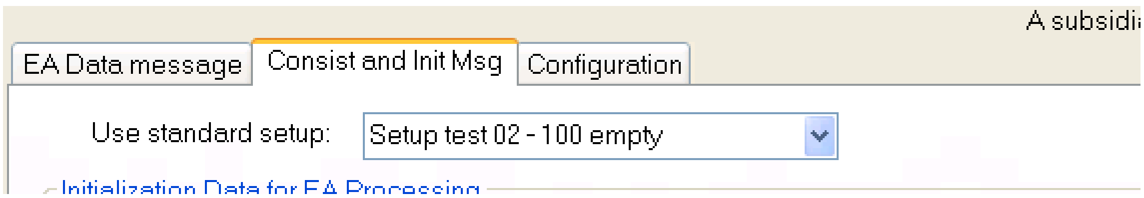


Figure C5. Select Input Parameters

3. After starting the simulation test, the application sends test data to the EA software, and the EA software should trigger a brake application. This is displayed on the EA data message tab:

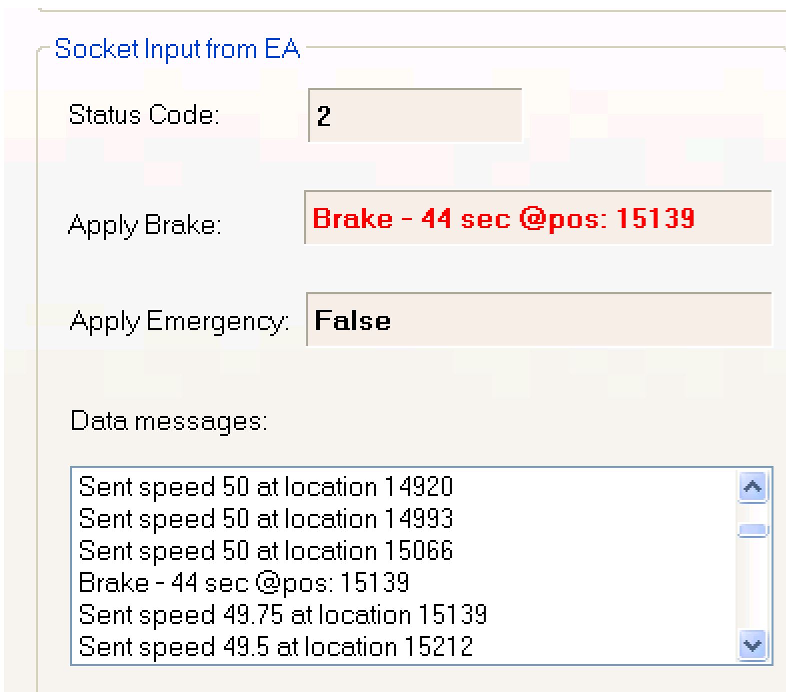


Figure C6. EA Data Message

4. The brake position should be recorded for each of the test scenarios in the test matrix.

5. After installation of the virtual machine image or EA software at the TTCI test lab, the test matrix is executed to validate the installation process.
6. As a final step, a TCL test batch matching the test matrix is executed and the results are compared to those supplied in step 4. The test results should be similar to those in step 4, but will vary slightly due to TOES™ variations and TCL's use of the cruise control feature to maintain train speed.

Table C4. Setup Test Matrix

Test 1	Unit coal – 100 cars, 2 locomotives, 30 mph, flat track
Test 2	Unit coal – 100 cars (empty), 2 locomotives, 50 mph, flat track
Test 3	General freight – 20 loads, 20 empty, 2 locomotives, 40 mph, 1.5 percent decline (TrackId = 8,034)
Test 4	General freight – 20 loads, 2 locomotives, 20 mph, 1.5 percent incline (TrackId= 8,036)

This test must match a test batch in the TTCI test environment.

Appendix C-C.
Predicted Acceleration Plots

Brake force was calculated for a 20-second warning time using the process for general regression in Statistica®. The results and statistical significance assigned to each variable are shown in the illustrations below. Normal probability plots are used to demonstrate the goodness of fit.

Intermodal, HE power, empty:

```
Prediction equation for:  Acceleration (total) = -0.2581327173-
0.005662713061*"INIT_TRAIN_VEL"-0.07105956446*"Percent Grade"-9.884469084e-009*"Total
train weight"+2.365124179e-005*"Train Length"
```

Univariate Tests of Significance, Effect Sizes, and Powers for Acceleration (total) (Int HE Empty in WTI simulation results for 20s warning time)
Sigma-restricted parameterization
Effective hypothesis decomposition

Effect	SS	Degr. of Freedom	MS	F	p	Partial eta-squared	Non-centrality	Observed power (alpha=0.05)
Intercept	0.628612	1	0.628612	485.651	0.000000	0.561673	485.651	1.000000
INIT_TRAIN_VEL	4.103747	1	4.103747	3170.460	0.000000	0.893223	3170.460	1.000000
Percent Grade	2.188300	1	2.188300	1690.630	0.000000	0.816875	1690.630	1.000000
Total train weight	0.001422	1	0.001422	1.099	0.295236	0.002890	1.099	0.181562
Train Length	0.008277	1	0.008277	6.395	0.011851	0.016593	6.395	0.713081
Error	0.490566	379	0.001294					

Test of SS Whole Model vs. SS Residual (Int HE Empty in WTI simulation results for 20s warning time)

Dependent Variable	Multiple R	Multiple R^2	Adjusted R^2	SS Model	df Model	MS Model	SS Residual	df Residual	MS Residual	F	p
Acceleration (total)	0.968505	0.938002	0.937348	7.422075	4	1.855519	0.490566	379	0.001294	1433.531	0.00

Figure C7. Intermodal HE Power Empty Statistical Significance

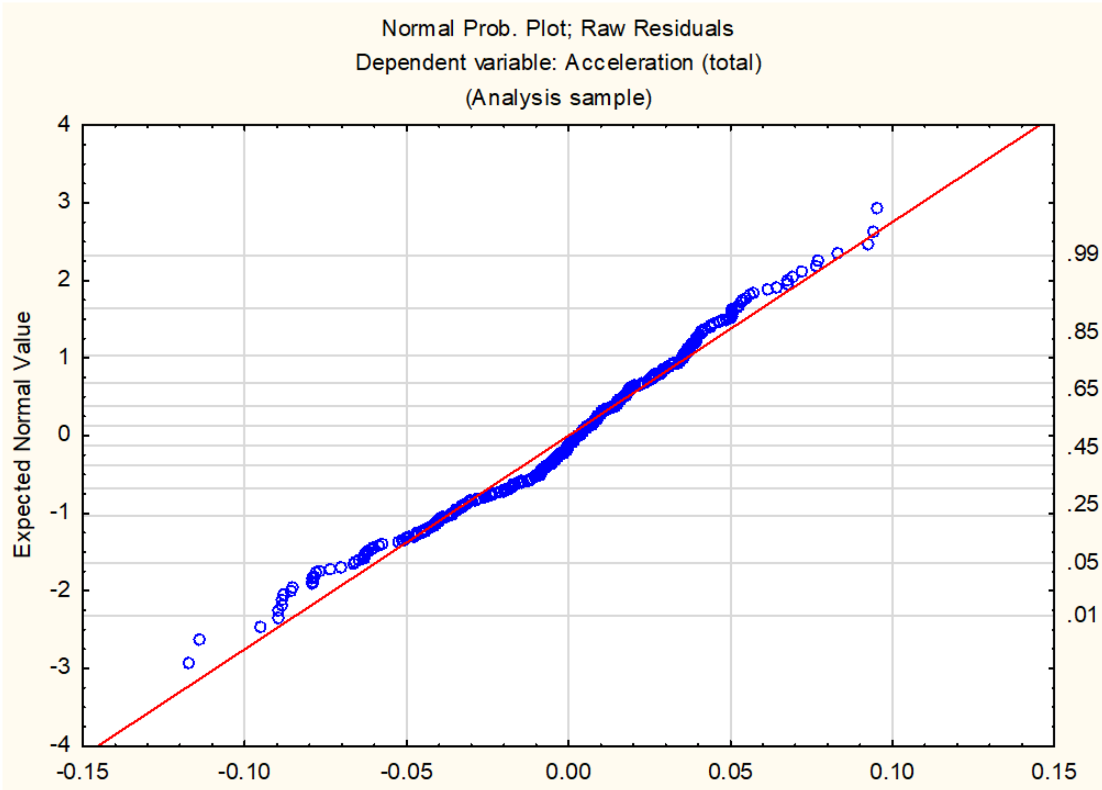


Figure C8. Intermodal HE Power Empty Probability Plot

Intermodal, HE power, loaded:

```
Prediction equation for:  Acceleration (total) = -0.2255443124-
0.004425853473*"INIT_TRAIN_VEL"-0.07590480791*"Percent Grade"+3.757110141e-009*"Total
train weight"+3.983331614e-006*"Train Length"
```

Univariate Tests of Significance, Effect Sizes, and Powers for Acceleration (total) (Int HE Loaded in WTI simulation results for 20s warning time)
Sigma-restricted parameterization
Effective hypothesis decomposition

Effect	SS	Degr. of Freedom	MS	F	p	Partial eta-squared	Non-centrality	Observed power (alpha=0.05)
Intercept	0.617398	1	0.617398	319.237	0.000000	0.457204	319.237	1.000000
INIT_TRAIN_VEL	3.342689	1	3.342689	1728.397	0.000000	0.820157	1728.397	1.000000
Percent Grade	2.503454	1	2.503454	1294.456	0.000000	0.773523	1294.456	1.000000
Total train weight	0.000398	1	0.000398	0.206	0.650430	0.000542	0.206	0.073755
Train Length	0.000275	1	0.000275	0.142	0.706332	0.000375	0.142	0.066358
Error	0.732979	379	0.001934					

Test of SS Whole Model vs. SS Residual (Int HE Loaded in WTI simulation results for 20s warning time)

Dependent Variable	Multiple R	Multiple R²	Adjusted R²	SS Model	df Model	MS Model	SS Residual	df Residual	MS Residual	F	p
Acceleration (total)	0.950068	0.902628	0.901601	6.794675	4	1.698669	0.732979	379	0.001934	878.3272	0.00

Figure C9. Intermodal HE Power Loaded Statistical Significance

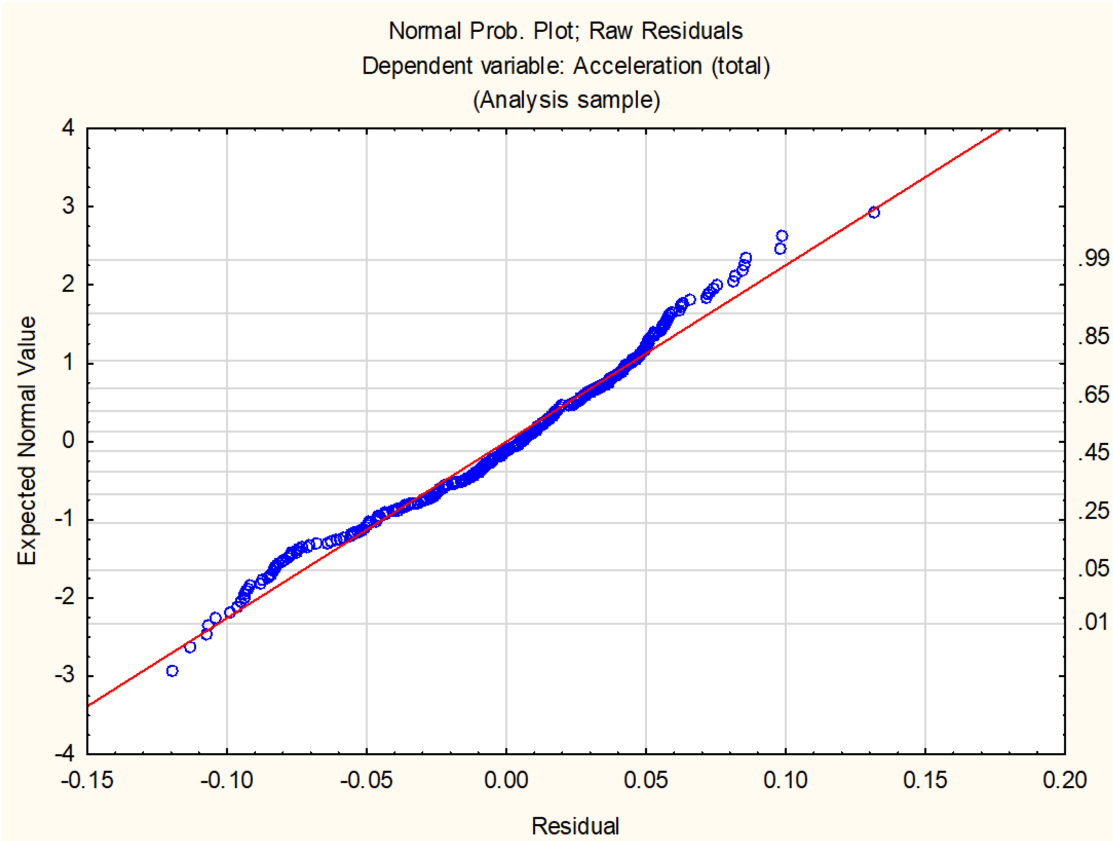


Figure C10. Intermodal HE Power Loaded Probability Plot

Intermodal, DP, empty:

```
Prediction equation for:  Acceleration (total) = -0.2457155281-
0.006759925573*"INIT_TRAIN_VEL"-0.102090469*"Percent Grade"-1.761681413e-008*"Total
train weight"+2.198616475e-005*"Train Length"
```

Univariate Tests of Significance, Effect Sizes, and Powers for Acceleration (total) (Int DP Empty in WTI simulation results for 20s warning time)
Sigma-restricted parameterization
Effective hypothesis decomposition

Effect	SS	Degr. of Freedom	MS	F	p	Partial eta-squared	Non-centrality	Observed power (alpha=0.05)
Intercept	2.06411	1	2.06411	1272.989	0.000000	0.625243	1272.989	1.000000
INIT_TRAIN_VEL	11.69621	1	11.69621	7213.362	0.000000	0.904342	7213.362	1.000000
Percent Grade	9.03363	1	9.03363	5571.275	0.000000	0.879544	5571.275	1.000000
Total train weight	0.02403	1	0.02403	14.819	0.000128	0.019052	14.819	0.970268
Train Length	0.03689	1	0.03689	22.753	0.000002	0.028957	22.753	0.997477
Error	1.23718	763	0.00162					

Test of SS Whole Model vs. SS Residual (Int DP Empty in WTI simulation results for 20s warning time)

Dependent Variable	Multiple R	Multiple R²	Adjusted R²	SS Model	df Model	MS Model	SS Residual	df Residual	MS Residual	F	p
Acceleration (total)	0.975439	0.951481	0.951227	24.26189	4	6.065474	1.237178	763	0.001621	3740.737	0.00

Figure C11. Intermodal DP Empty Statistical Significance

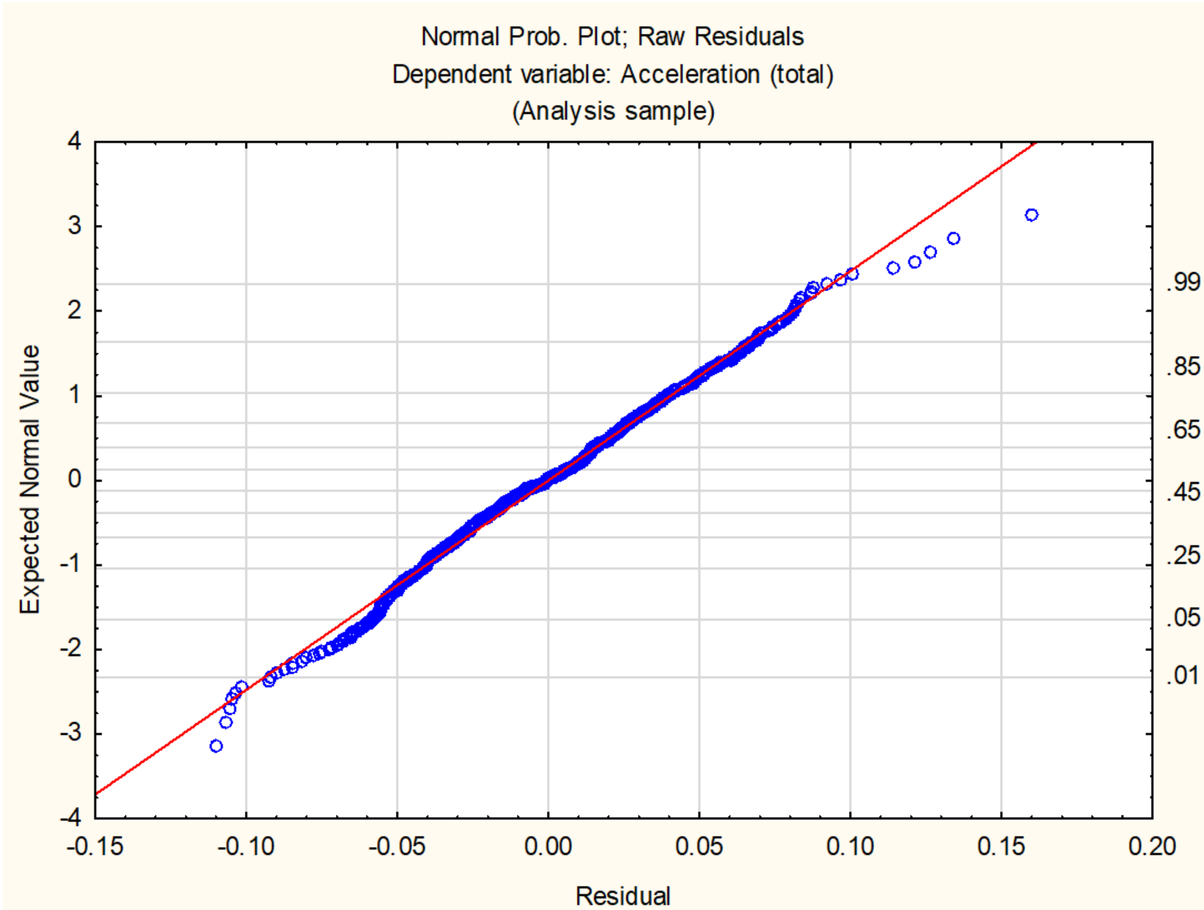


Figure C12. Intermodal DP Empty Probability Plot

Intermodal, DP, loaded:

```
Prediction equation for:  Acceleration (total) = -0.2431909774-
0.005178826432*"INIT_TRAIN_VEL"-0.1019671362*"Percent Grade"-1.301773715e-008*"Total
train weight"+2.136420592e-005*"Train Length"
```

	Univariate Tests of Significance, Effect Sizes, and Powers for Acceleration (total) (Int DP Loaded in WTI simulation results for 20s warning time) Sigma-restricted parameterization Effective hypothesis decomposition							
Effect	SS	Degr. of Freedom	MS	F	p	Partial eta-squared	Non-centrality	Observed power (alpha=0.05)
Intercept	2.225395	1	2.225395	1146.276	0.000000	0.600372	1146.276	1.000000
INIT_TRAIN_VEL	7.803421	1	7.803421	4019.454	0.000000	0.840458	4019.454	1.000000
Percent Grade	9.060452	1	9.060452	4666.937	0.000000	0.859483	4666.937	1.000000
Total train weight	0.011737	1	0.011737	6.046	0.014162	0.007861	6.046	0.689958
Train Length	0.018648	1	0.018648	9.606	0.002011	0.012433	9.606	0.871901
Error	1.481298	763	0.001941					

	Test of SS Whole Model vs. SS Residual (Int DP Loaded in WTI simulation results for 20s warning time)										
Dependent Variable	Multiple R	Multiple R^2	Adjusted R^2	SS Model	df Model	MS Model	SS Residual	df Residual	MS Residual	F	p
Acceleration (total)	0.963930	0.929160	0.928789	19.42929	4	4.857324	1.481298	763	0.001941	2501.953	0.00

Figure C13. Intermodal DP Loaded Statistical Significance

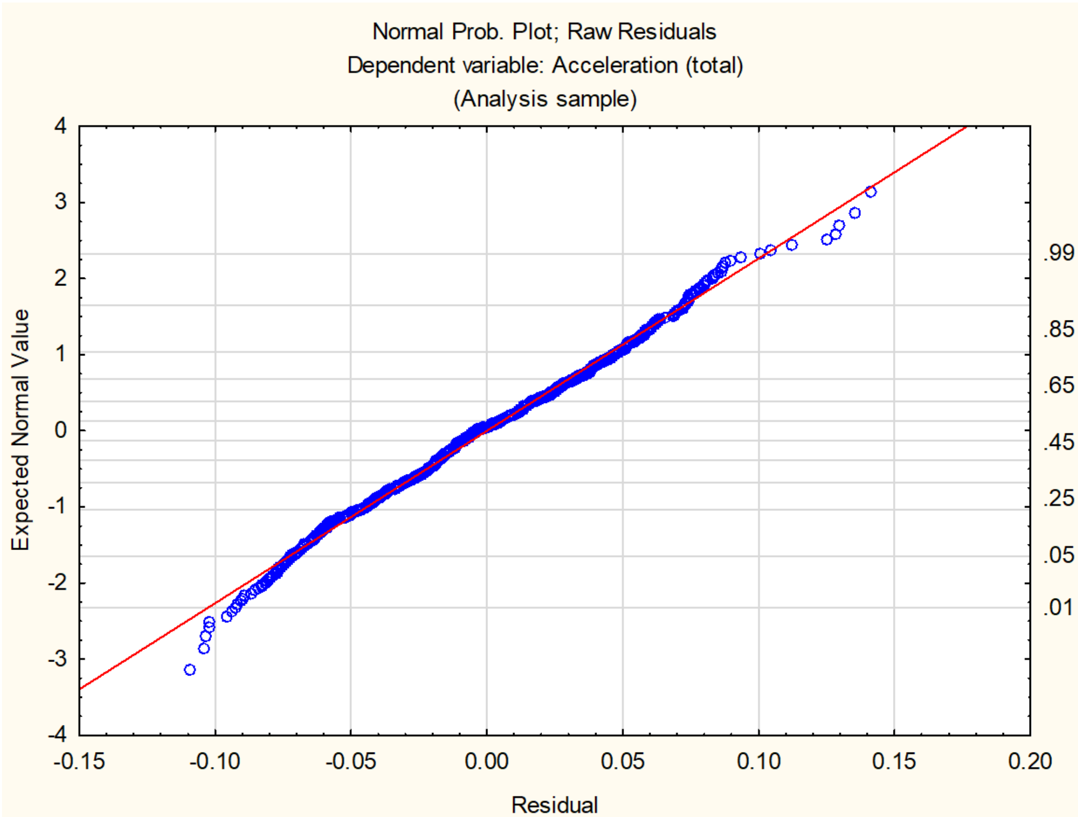


Figure C14. Intermodal DP Loaded Probability Plot

Unit train, HE power, empty:

```
Prediction equation for:  Acceleration (total) = -0.2930718435-
0.005567671979*"INIT_TRAIN_VEL"-0.0879631033*"Percent Grade"+6.047437169e-009*"Total
train weight"+1.187227208e-005*"Train Length"
```

Effect	Univariate Tests of Significance, Effect Sizes, and Powers for Acceleration (total) (Unit HE Empty in WTI simulation results for 20s warning time) Sigma-restricted parameterization Effective hypothesis decomposition							
	SS	Degr. of Freedom	MS	F	p	Partial eta-squared	Non-centrality	Observed power (alpha=0.05)
Intercept	1.129521	1	1.129521	1033.961	0.000000	0.795991	1033.961	1.000000
INIT_TRAIN_VEL	2.715200	1	2.715200	2485.488	0.000000	0.903653	2485.488	1.000000
Percent Grade	1.791098	1	1.791098	1639.567	0.000000	0.860861	1639.567	1.000000
Total train weight	0.011629	1	0.011629	10.645	0.001249	0.038618	10.645	0.901619
Train Length	0.030825	1	0.030825	28.217	0.000000	0.096233	28.217	0.999570
Error	0.289492	265	0.001092					

Dependent Variable	Test of SS Whole Model vs. SS Residual (Unit HE Empty in WTI simulation results for 20s warning time)										
	Multiple R	Multiple R²	Adjusted R²	SS Model	df Model	MS Model	SS Residual	df Residual	MS Residual	F	p
Acceleration (total)	0.973529	0.947759	0.946971	5.252001	4	1.313000	0.289492	265	0.001092	1201.917	0.00

Figure C15. Unit Train HE Power Empty Statistical Significance

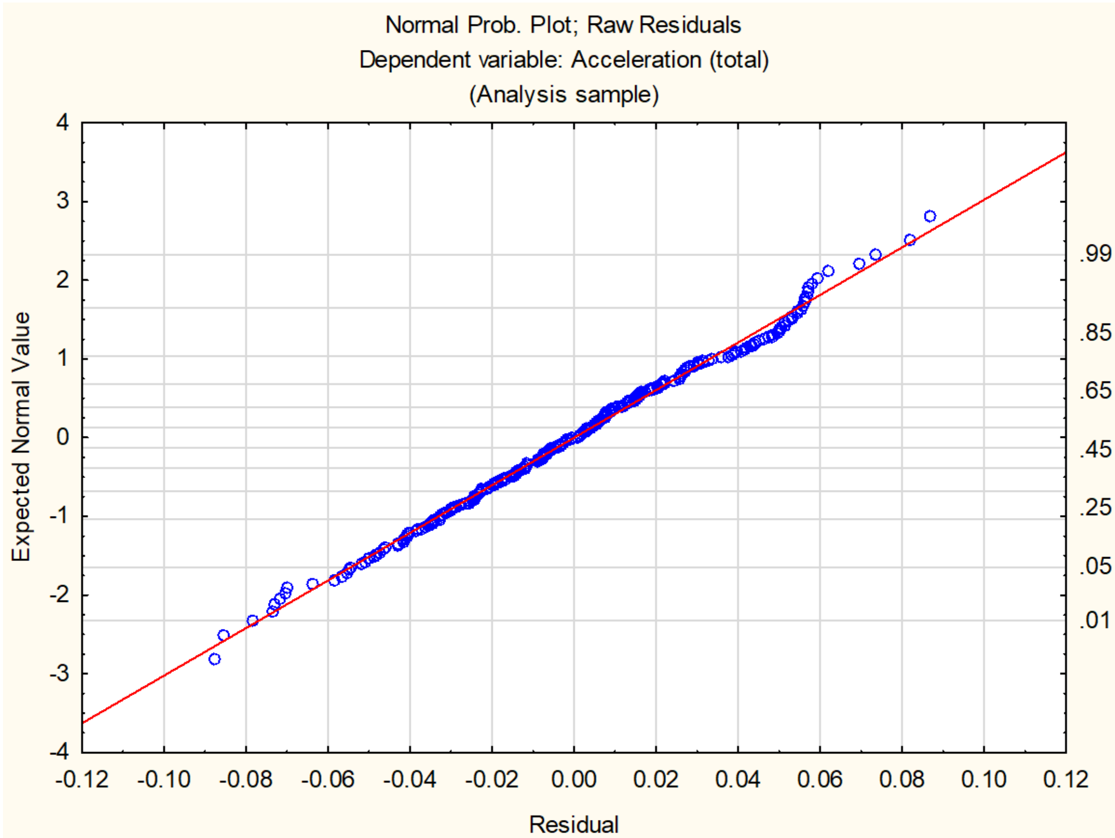


Figure C16. Unit Train HE Power Empty Probability Plot

Unit, HE power, loaded:

```
Prediction equation for:  Acceleration (total) = -0.3284914512-
0.001359807051*"INIT_TRAIN_VEL"-0.1849572822*"Percent Grade"-1.322000796e-009*"Total
train weight"+1.053002802e-005*"Train Length"
```

Univariate Tests of Significance, Effect Sizes, and Powers for Acceleration (total) (Unit HE Loaded in WTI simulation results for 20s warning time)
Sigma-restricted parameterization
Effective hypothesis decomposition

Effect	SS	Degr. of Freedom	MS	F	p	Partial eta-squared	Non-centrality	Observed power (alpha=0.05)
Intercept	0.065559	1	0.065559	2.7462	0.098755	0.010996	2.7462	0.378723
INIT_TRAIN_VEL	0.100142	1	0.100142	4.1949	0.041604	0.016700	4.1949	0.531995
Percent Grade	7.429411	1	7.429411	311.2123	0.000000	0.557516	311.2123	1.000000
Total train weight	0.001329	1	0.001329	0.0557	0.813689	0.000225	0.0557	0.056350
Train Length	0.049339	1	0.049339	2.0668	0.151805	0.008298	2.0668	0.299123
Error	5.896503	247	0.023872					

Test of SS Whole Model vs. SS Residual (Unit HE Loaded in WTI simulation results for 20s warning time)

Dependent Variable	Multiple R	Multiple R²	Adjusted R²	SS Model	df Model	MS Model	SS Residual	df Residual	MS Residual	F	p
Acceleration (total)	0.749694	0.562041	0.554949	7.567098	4	1.891775	5.896503	247	0.023872	79.24499	0.00

Figure C17. Unit Train HE Power Loaded Statistical Significance

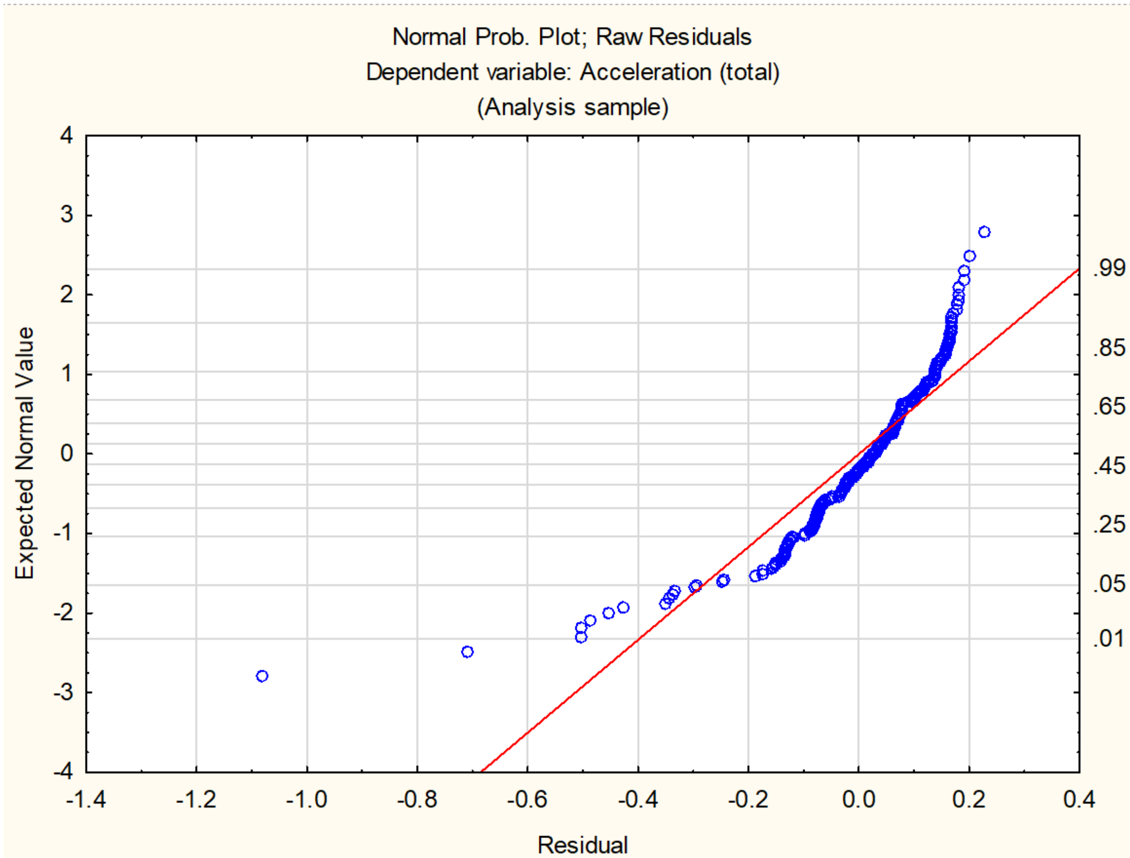


Figure C18. Unit Train HE Power Loaded Probability Plot

Unit, DP, empty:

```
Prediction equation for:  Acceleration (total) = -0.2966699088-
0.006671565187*"INIT_TRAIN_VEL"-0.1113740445*"Percent Grade"+4.678493176e-009*"Total
train weight"+4.402221962e-006*"Train Length"
```

Effect	Univariate Tests of Significance, Effect Sizes, and Powers for Acceleration (total) (Unit DP Empty in WTI simulation results for 20s warning time) Sigma-restricted parameterization Effective hypothesis decomposition							
	SS	Degr. of Freedom	MS	F	p	Partial eta-squared	Non-centrality	Observed power (alpha=0.05)
Intercept	4.80884	1	4.80884	1984.068	0.000000	0.668922	1984.068	1.000000
INIT_TRAIN_VEL	14.18826	1	14.18826	5853.897	0.000000	0.856347	5853.897	1.000000
Percent Grade	10.45319	1	10.45319	4312.856	0.000000	0.814537	4312.856	1.000000
Total train weight	0.06286	1	0.06286	25.934	0.000000	0.025730	25.934	0.999119
Train Length	0.03421	1	0.03421	14.117	0.000182	0.014172	14.117	0.963559
Error	2.38010	982	0.00242					

Dependent Variable	Test of SS Whole Model vs. SS Residual (Unit DP Empty in WTI simulation results for 20s warning time)										
	Multiple R	Multiple R²	Adjusted R²	SS Model	df Model	MS Model	SS Residual	df Residual	MS Residual	F	p
Acceleration (total)	0.959661	0.920949	0.920627	27.72850	4	6.932125	2.380102	982	0.002424	2860.108	0.00

Figure C19. Unit Train DP Empty Statistical Significance

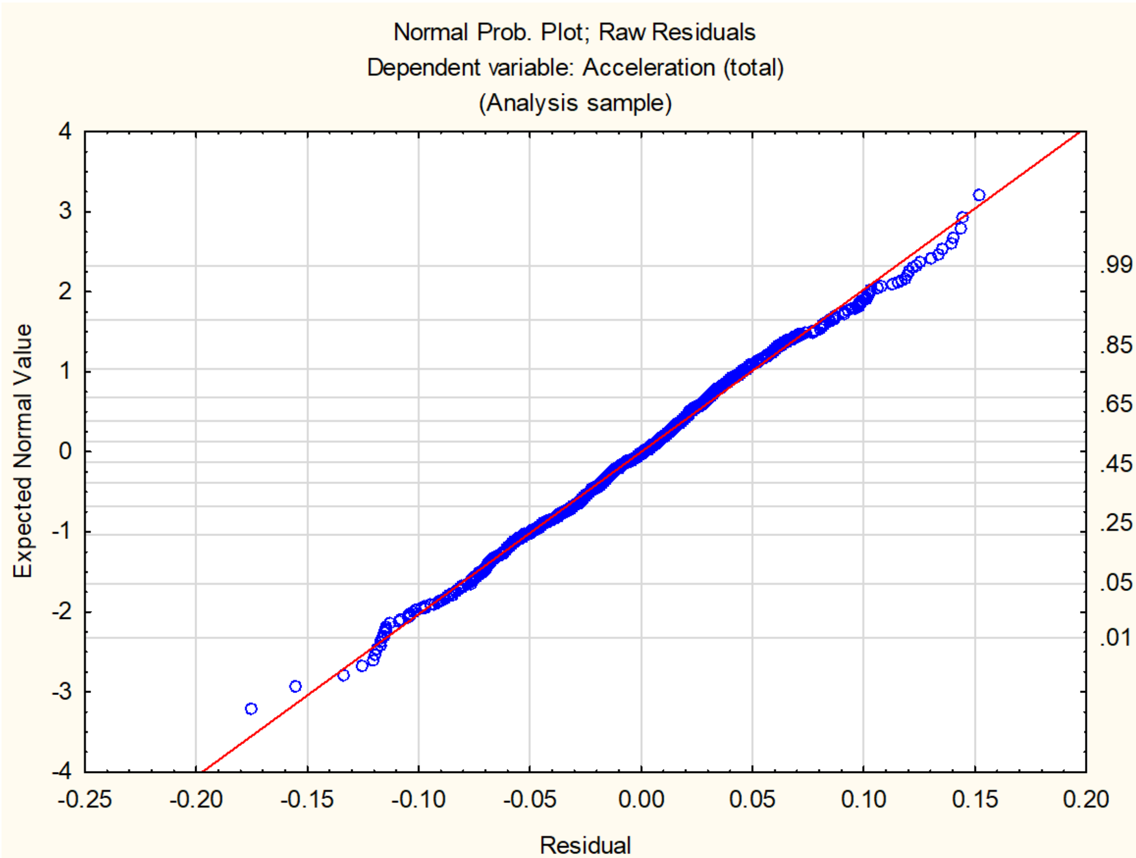


Figure C20. Unit Train DP Empty Probability Plot

Unit, DP, loaded:

```
Prediction equation for:  Acceleration (total) = -0.3357501876-
0.0031702844*"INIT_TRAIN_VEL"-0.1816791282*"Percent Grade"-6.188591879e-010*"Total
train weight"+8.129343346e-006*"Train Length"
```

Effect	SS	Degr. of Freedom	MS	F	p	Partial eta-squared	Non-centrality	Observed power (alpha=0.05)
	Univariate Tests of Significance, Effect Sizes, and Powers for Acceleration (total) (Unit DP Loaded in WTI simulation results for 20s warning time) Sigma-restricted parameterization Effective hypothesis decomposition							
Intercept	4.65481	1	4.65481	263.499	0.000000	0.222832	263.499	1.000000
INIT_TRAIN_VEL	1.95785	1	1.95785	110.830	0.000000	0.107620	110.830	1.000000
Percent Grade	26.20538	1	26.20538	1483.431	0.000000	0.617471	1483.431	1.000000
Total train weight	0.03910	1	0.03910	2.214	0.137141	0.002403	2.214	0.318142
Train Length	0.34989	1	0.34989	19.807	0.000010	0.021098	19.807	0.993538
Error	16.23448	919	0.01767					

Dependent Variable	Multiple R	Multiple R²	Adjusted R²	SS Model	df Model	MS Model	SS Residual	df Residual	MS Residual	F	p
	Test of SS Whole Model vs. SS Residual (Unit DP Loaded in WTI simulation results for 20s warning time)										
Acceleration (total)	0.792814	0.628553	0.626937	27.47162	4	6.867906	16.23448	919	0.017665	388.7777	0.00

Figure C21. Unit Train DP Loaded Statistical Significance

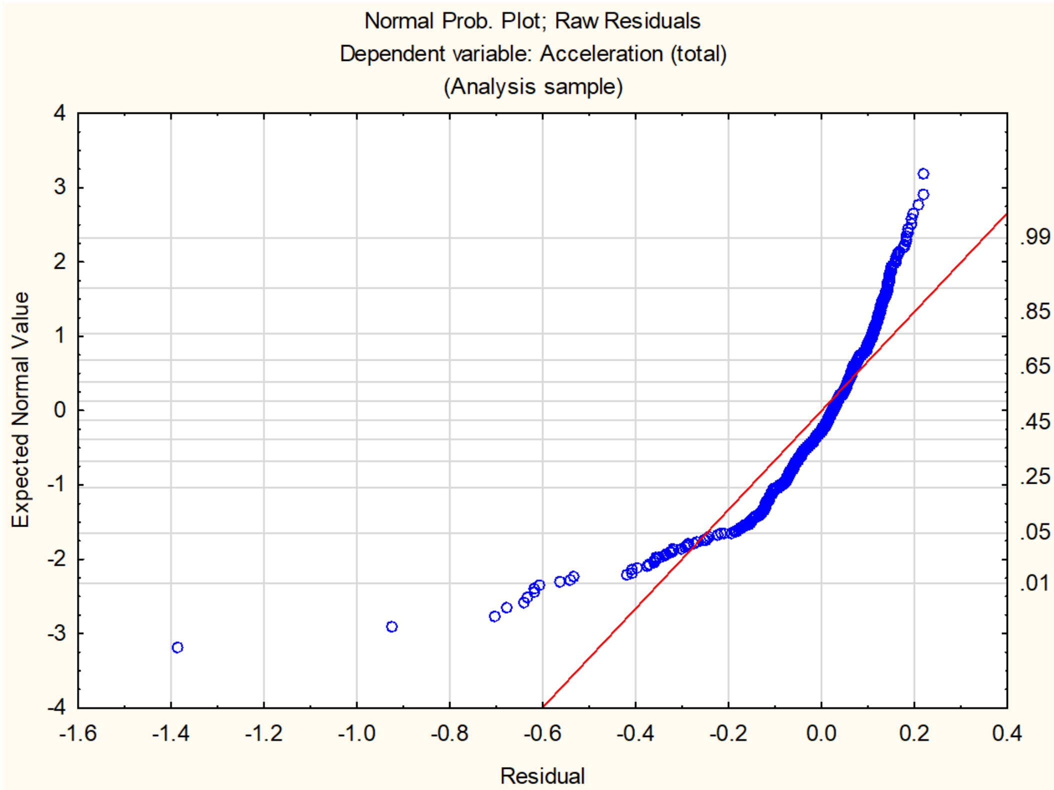


Figure C22. Unit Train DP Loaded Probability Plot

Manifest, HE, mixed load:

```
Prediction equation for:  Acceleration (total) = -0.3523227771-
0.005813667431*"INIT_TRAIN_VEL"-0.09839584946*"Percent Grade"+4.811065099e-009*"Total
train weight"+2.6136144e-005*"Train Length"
```

	Univariate Tests of Significance, Effect Sizes, and Powers for Acceleration (total) (Manifest HE in WTI simulation results for 20s warning time) Sigma-restricted parameterization Effective hypothesis decomposition							
Effect	SS	Degr. of Freedom	MS	F	p	Partial eta-squared	Non-centrality	Observed power (alpha=0.05)
Intercept	33.27831	1	33.27831	2864.385	0.000000	0.651680	2864.385	1.000000
INIT_TRAIN_VEL	14.41762	1	14.41762	1240.977	0.000000	0.447687	1240.977	1.000000
Percent Grade	16.59403	1	16.59403	1428.309	0.000000	0.482649	1428.309	1.000000
Total train weight	0.13479	1	0.13479	11.601	0.000676	0.007521	11.601	0.925629
Train Length	0.38673	1	0.38673	33.287	0.000000	0.021279	33.287	0.999929
Error	17.78709	1531	0.01162					

	Test of SS Whole Model vs. SS Residual (Manifest HE in WTI simulation results for 20s warning time)										
Dependent Variable	Multiple R	Multiple R²	Adjusted R²	SS Model	df Model	MS Model	SS Residual	df Residual	MS Residual	F	p
Acceleration (total)	0.867339	0.752277	0.751630	54.01530	4	13.50383	17.78709	1531	0.011618	1162.323	0.00

Figure C23. Manifest Train HE Power Mixed Load Statistical Significance

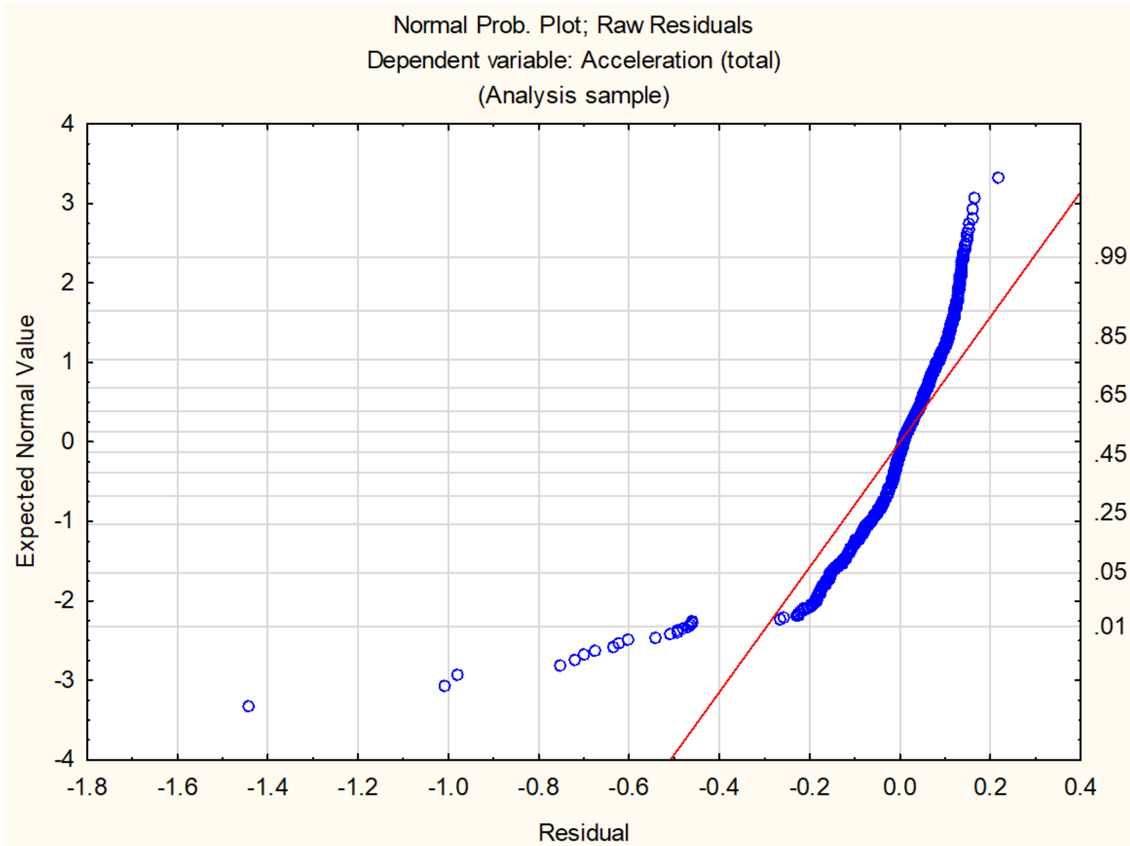


Figure C24. Manifest Train HE Power Mixed Load Probability Plot

Manifest, DP, mixed load:

```
Prediction equation for:  Acceleration (total) = -0.2187308977-
0.005368261801*"INIT_TRAIN_VEL"-0.09986834916*"Percent Grade"+9.375050189e-010*"Total
train weight"+4.082148574e-006*"Train Length"
```

Effect	Univariate Tests of Significance, Effect Sizes, and Powers for Acceleration (total) (Manifest DP in WTI simulation results for 20s warning time) Sigma-restricted parameterization Effective hypothesis decomposition							
	SS	Degr. of Freedom	MS	F	p	Partial eta-squared	Non-centrality	Observed power (alpha=0.05)
Intercept	3.57743	1	3.57743	2354.25	0.000000	0.564128	2354.25	1.000000
INIT_TRAIN_VEL	14.09763	1	14.09763	9277.42	0.000000	0.836073	9277.42	1.000000
Percent Grade	16.82881	1	16.82881	11074.76	0.000000	0.858924	11074.76	1.000000
Total train weight	0.04321	1	0.04321	28.43	0.000000	0.015391	28.43	0.999624
Train Length	0.05304	1	0.05304	34.90	0.000000	0.018826	34.90	0.999960
Error	2.76409	1819	0.00152					

Dependent Variable	Test of SS Whole Model vs. SS Residual (Manifest DP in WTI simulation results for 20s warning time)										
	Multiple R	Multiple R²	Adjusted R²	SS Model	df Model	MS Model	SS Residual	df Residual	MS Residual	F	p
Acceleration (total)	0.958557	0.918831	0.918653	31.28949	4	7.822372	2.764088	1819	0.001520	5147.772	0.00

Figure C25. Manifest Train DP Mixed Load Statistical Significance

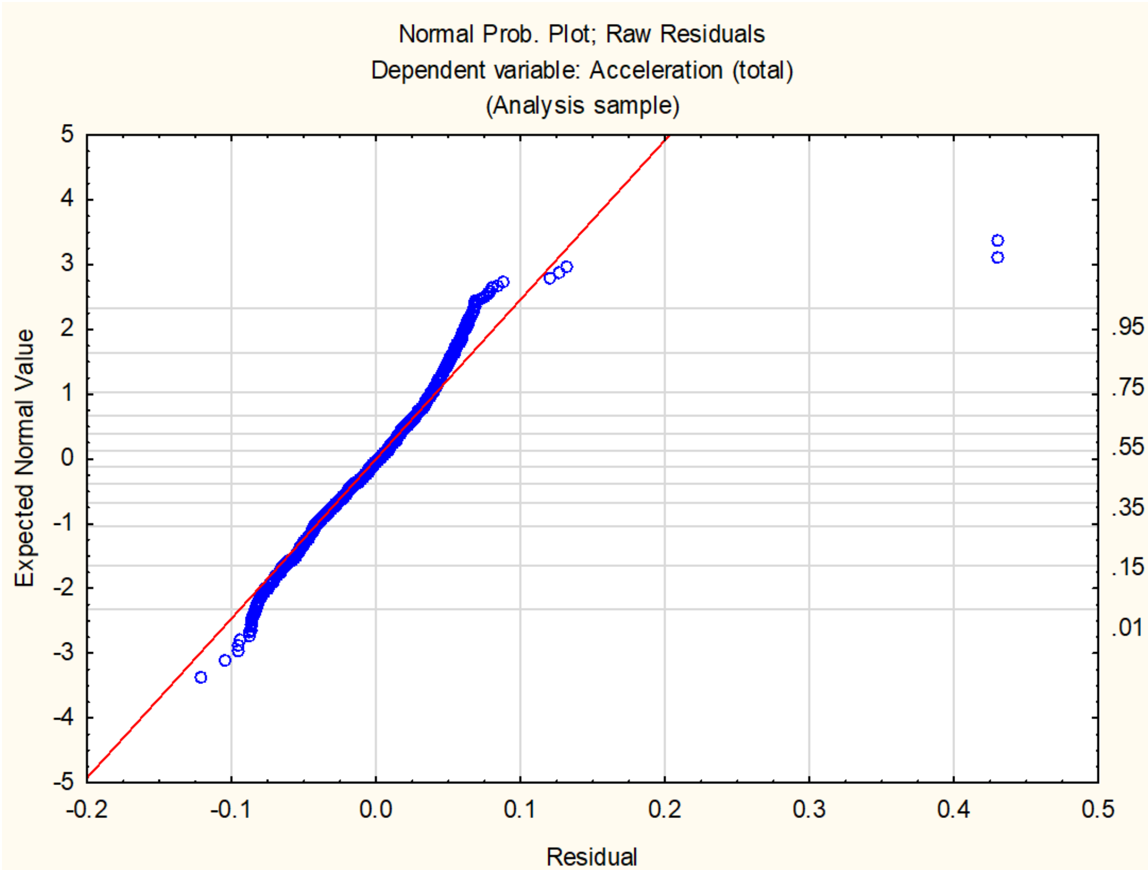


Figure C26. Manifest Train DP Mixed Load Probability Plot

Abbreviations and Acronyms

ACRONYMS	EXPLANATION
AAR	Association of American Railroads
AG	Advisory Group
DP	Distributed Power
EBB	Emergency Brake Backup
EA	Enforcement Algorithm
FRA	Federal Railroad Administration
GUI	Graphical User Interface
HE	Head End
NAS	Network Addressable Storage
OBC	Onboard Computer
PTC	Positive Train Control
SQL	Standardized Query Language
RTC	Rail Traffic Controller
TCL	Test Controller and Logger
TCP/IP	Transmission Control Protocol/Internet Protocol
TOES™	Train Operation Energy Simulator
TTCI	Transportation Technology Center Inc.
WTI	Warning Time Input