Memorandum

U.S. Department
of Transportation

Federal Railroad
Administration

Date: APR - 1 2014

Subject: Positive Train Control Technical Bulletin PTC-14-03
Substitution of Railroad Braking Model Algorithm Test Results in lieu of Live
Field Brake Testing of Positive Train Control Systems

From: Robert C. Lauby Robert C Lauby
Associate Administrator for Railroad Safety
Chief Safety Officer

To: All Federal Railroad Administration Staff Directors, Field Employees, and
Participating State Employees

The purpose of this technical bulletin is to provide the Federal Railroad Administration (FRA) Office of Railroad Safety personnel with guidance regarding the conditions under which it is allowable to substitute laboratory brake test model results in lieu of conducting of field testing.

FRA will allow the substitution of laboratory modeling of braking algorithm performance in lieu of live field braking testing, provided the accuracy of the model has been demonstrated and the model has been verified and validated to the satisfaction of FRA as outlined in the attached report, Braking Test Model Extrapolation Positive Train Control Systems, Version 1.0.

FRA will require full field testing of the braking algorithm to FRA satisfaction under all operating conditions in the event of:

- Significant deviations (greater than 15 percent) between the model results and observed field results.
- Any field braking event that occurs where the train passes its stop target at any speed.

Questions regarding this guidance should be directed to Dr. Mark Hartong, Senior Scientific/Technical Advisor, at (202) 493-1332 or Mark.Hartong@dot.gov.

Attachment

Version 1.0

U.S. Department of Transportation
Federal Railroad Administration

BRAKING TEST MODEL EXTRAPOLATION
POSITIVE TRAIN CONTROL SYSTEMS

Office of Railroad Safety
Washington, DC 20590

FOREWORD

This document defines a set of necessary actions for the use of railroad braking model algorithm test results in lieu of live field brake testing of Positive Train Control (PTC) systems in high risk operating scenarios. The Office of Railroad Safety guidelines in this document ensure an appropriate level of verification and validation of braking models in support of PTC. System Certification is required by the Rail Safety Improvement Act of 2008 (RSIA). Since it is impractical to cover all situations or conditions that may arise, the auditor or monitor must supplement them with good judgment as required.

Please forward any deficiencies, clarifications, or suggested improvements regarding the content of this document to the Signal and Train Control Division Staff Director, Federal Railroad Administration, 1200 New Jersey Avenue SE, Washington, DC 20590.

Version 1.0

Table of Revisions

Version	Date	Notes
1.0	3/19/2014	Technical bulletin routed for issuance

TABLE OF CONTENTS

Introduction

Background

Certification of PTC systems is required by 49 U.S.C. § 20157(h). This statute states that "The Secretary shall not permit the installation of any positive train control system or component in revenue service unless the Secretary has certified that any such system or component has been approved through the approval process set forth in part 236 of title 49, Code of Federal Regulations, and complies with the requirements of that part."

A critical function of PTC is when braking is initiated by the PTC system in an attempt to stop a train at, or short of, a stop target. To ensure safety, trains must have sufficient distance in which to stop. Enforcing a stop to early (extreme distance from target) unnecessarily reduces the capacity of the line and adversely affects the throughput of the railroads network. Enforcing a stop too late (short distance from target) could result in collisions because the train would not be able to stop within the available distance and may occupy a section of track that is potentially already allocated to another train. The wide variability in train characteristics (i.e. types, length and weight, number of brakes, initial speed, and effective braking force.) and environmental conditions (i.e. track grade, ambient temperature, and coefficient of friction between train wheels and the rail head) further complicates implementation of this functionality. It is impossible to calculate the precise stopping distance consistently. This distance can vary significantly due to the condition of the train and the environmental conditions at the time of braking.

When evaluating the braking behavior of PTC systems, the Federal Railroad Administration (FRA) has previously relied on field testing various train configurations. This field testing method requires the PTC system to initiate braking with a set of predefined train characteristics and environmental conditions (i.e. heaviest train, least amount of effective braking force available, stopping from highest velocity on steepest gradient). The performance of the train is observed, and the scenario is repeated under the same conditions. If the PTC system that initiated the braking is able to stop the train before the predetermined stopping point in all cases, it is assumed that the PTC system will successfully stop any train operating within the defined conditions.

There are some limitations to this approach. First, repetition of field braking tests is limited due to the availability of resources (e.g., financial, schedule, workforce, and equipment). Second, the values of the test attributes are not consistent. There are minor variations, for example, in speed, effective braking force, coefficient of friction, and ambient temperature that occur during each test execution. Third, some operational scenarios present an unacceptable level of risk in attempting due to the consequences if the PTC system does or does not perform as expected.

Using validated and verified models accurately allows for the extrapolation of results to define PTC system operating parameters. Using these models also allows for the ability to confidently carry out a large numbers of tests with consistent parameters and results. Verification ensures a model is built correctly, while validation ensures the use of the correct model. The verification and validation process also addresses whether the model is accurate for the purposes it is used.

Scope and Purpose

A fundamental limitation of any engineering assessment is that there are very few absolutes. No model is ever completely accurate. In terms of braking, this means there will always be a probability (albeit very small) that it will not be able to stop a train short of the intended stop point. Similarly, there are no guarantees that, even though verified and validated, a model will be "correct" and fully emulate a situation. The best that can be hoped for is that these probabilities can be reduced to acceptable levels.

The remainder of this document addresses two issues: what are acceptable means for verification and validation, and under what circumstances can results of a verified and validated braking model be used as a substitute for actual field testing to establish PTC system operating restrictions. This document is not a substitute for good judgment, experience, and common sense.

Entities who propose to substitute braking models results for actual field testing results may only do so after demonstrating to FRA that they satisfied the verification, validation, and accuracy requirements outlined in the remainder of the document. FRA will make the final determination if the use of a model is appropriate.

Simulation Modeling

Simulation modeling is used in conjunction with testing and analysis to gain confidence that the design implementation is performing as expected. The development of a simulation model that has undergone formal verification and validation is not only desirable, but is essential to determine its consistency with reality. Without verification and validation of the model, there is no basis to place confidence in a study's results.
Verification is the process of determining that the implementation of a model and its associated data accurately represent the developer's conceptual description and specifications. Validation is the process of determining the degree to which a simulation model and its associated data are an accurate representation of the real world from the perspective of the intended uses of the model. It is important to remember that validation does not imply verification, nor does verification imply validation. However, validation is often blended with verification, especially when measurement data is available for the system being modeled. If comparing system measurements and model results suggests that the results produced by the model are close to those obtained from the system, then the implemented model is assumed to be both a verified implementation of the assumptions and a valid representation of the system.

Validation

Before use, any model proposed for braking studies must complete the following four types of validation:

- *Conceptual Model Validation*: This is the process of determining if the scope and level of detail of the proposed model is sufficient for the purpose at hand, and that any assumptions are correct. This answers whether the conceptual model contains all the necessary details to meet the objectives of the simulation study.

  There are no formal methods for validating a conceptual model. Review of project specifications that outline the objectives of the project and the approach (whether formally or informally expressed) can be useful. Circulating the specifications among subject matter experts who have a detailed knowledge of the system and obtaining feedback on the adequacy is the best approach to validation. The model validation is completed when there is consensus by the subject matter experts.

- *Data Validation*: This is the process of determining if the data required for model building, validation, and experimentation is sufficiently accurate. The source of any data is studied to determine its reliability. The data is analyzed for inconsistencies, with all inconsistencies investigated and resolved. Procedures need to be in place for collecting or estimating inaccurate or unavailable data and performing sensitivity analyses to ascertain the effects of any inaccuracies.

- *White-box Validation*: This is the process of determining if the constituent parts of the computer model represent the corresponding real world elements with sufficient accuracy. This is a detailed, or micro, check of the model that determines whether each part of the model represents the real world with sufficient accuracy. White-box validation ensures the content of the model is true to the real world (in this way it is an indirect form of conceptual model validation).

  The model code will have received an independent second check to ensure that it contains the right data and logic. Elements of the model are observed during model operation, with the behavior compared to the real world. Multiple approaches are used. These include, but are not limited to:
  - Stepping through the model event by event.
  - Stopping the mode at predetermined checkpoints, predicting what will happen next, running the model on and checking what happens.
  - Interactively setting up conditions to force certain events.
  - Tracing the progress of an item through the model.

  Finally the actual and expected results of the program are compared for consistency, accuracy, and plausibility.

- *Black-box Validation*: This is the process of determining if the overall model represents the real world with sufficient accuracy. This is an overall, or macro, check of the model's operation that determines whether the overall model provides a sufficiently accurate representation of the real world system.

  The difficulty with this form of validation is there may not be any accurate real world data with which to perform such a comparison. If this is the case, the comparison can

> be made against the expectations and intuition of those who have a detailed knowledge of the real system, or against other models. Comparison against approximate real world data may not give absolute confidence in the model, but it should help to increase confidence.

Each of these four modes of validation is data driven. The model is used to predict the system's behavior and then compare the system's behavior and the model's forecast to determine if they are the same. The system's data may come from an operational system or be obtained by conducting experiments on the system, e.g., field tests.

The use of data-driven predictive validation generally requires a significant amount of effort to acquire and analyze data to support the model validation. Various results (e.g., outputs) of the simulation model being validated are compared to results of other (valid) models. For example: (1) simple cases of a simulation model are compared to known results of analytic models and (2) the simulation model is compared to other simulation models that have been validated. If historical data exists (e.g., data collected on a system specifically for building and testing a model), part of the data is used to build the model with the remaining data used to determine (test) whether the model behaves as the system does.

Predictive validation also necessitates a sensitivity analysis, which consists of changing the values of the input and internal parameters of a model to determine the effects on the model's output behavior. The same relations should occur in the model as in the real system. This technique can be used qualitatively (directions of outputs only) and quantitatively (both directions and (precise) magnitudes of outputs). Those parameters that are sensitive (i.e., cause significant changes in the model's behavior or output) should be made sufficiently accurate prior to using the model.

The final step in the validation process is to compare values of the metrics chosen to measure the agreement between model outputs with the experimental data and to make an assessment of model accuracy. The determination if the validated system-level model is adequate for its intended use is a programmatic decision and involves both technical and nontechnical requirements such as schedule, availability, financial resources, public perception, etc. Stakeholders who are not part of the validation team will typically determine these nontechnical requirements. Therefore, the interpretation of adequacy is limited to include only the acceptable agreement between experimental and simulation outcomes.

Not every model element requires validation. Model elements associated with model functional requirements usually do not require validation. These elements are only dealt with in the verification phase. Trivial examples of this are model elements that allow the user to select various model options (i.e., switches or knobs). However, a model element that has not been deemed critical may, in fact, be fully functional when the simulation model is deployed. In this case, the model element may still be exercised, but the model documentation should note that this particular element has been "verified, but not validated."

Some observed behaviors of the actual system will be difficult to model or validate given the scope and resources of the model's development and validation efforts. In such cases, using simplifications (or approximations) in the model may provide an acceptable way forward. For example, if a model element requires a stochastic data-generating mechanism, a probability density function with a limited number of parameters (e.g., a Gaussian distribution) may be used in place of what appears to be, based on analysis of data from the actual system, a more complex data-generating mechanism. In doing this, a conservative approach should be used. That is, in this example, employing a simplified data-generating mechanism in the model should not result in overly optimistic behavior with respect to the actual system.

Verification

The verification phase focuses on comparing the elements of a simulation model with the description of the requirements and capabilities of the model.

Verification is an iterative process aimed at determining whether the product of each step in the development of the simulation model fulfills all the requirements levied on it by the previous step and is internally complete, consistent, and correct. If agreement is not obtained, the model is adjusted to bring it in closer agreement with the observed behavior of the actual system (or errors, in observation/experimentation or reference models/analyses, are identified and rectified). Verification addresses two main elements: the code and the calculations.

The purpose of code verification is to confirm that the software is working as intended. The main focus of this activity is to identify and eliminate programming and implementation errors within the software (software quality assurance) and to verify the correctness of the numerical algorithms that are implemented in the code (numerical algorithm verification). Code verification is the responsibility of both the code developer and the model developer.

Code verification is partially accomplished using software quality assurance (SQA) procedures. SQA performed by the code developer is used to ensure that the code is reliable (implemented correctly) and produces repeatable results on specified computer hardware, operating systems, and compilers. SQA is typically accomplished using configuration management, and static and dynamic software quality testing. SQA procedures are needed during the software development process and production computing.

Since it cannot be proven that a code is error free, the accumulation of well thought out test cases provides evidence that the code is sufficiently error free and accurate. These test problems must be documented, accessible, repeatable, and capable of being referenced. Documentation must also record the computer hardware used, the operating system, compiler versions, etc.

The purpose of calculation verification is to quantify the error of a numerical simulation by demonstration of convergence for the particular model under consideration (or a closely related one) and, if possible, to provide an estimation of the numerical errors induced by the use of the model. The types of errors being identified and removed by calculation verification include insufficient spatial or temporal discretization, insufficient convergence tolerance, incorrect input options, and finite precision arithmetic. Barring input errors, insufficient grid refinement is typically the largest contributor to error in calculation verification assessment.

In general, uncertainty quantification requires a numerical solution; therefore, calculation verification is required to quantify the numerical accuracy of the uncertainty analysis method being used. Approximate uncertainty analysis methods, which are typically required for computationally intensive models, can introduce errors into the numerical solution and must be verified. All errors associated with the uncertainty analysis method must be quantified during the calculation verification activity. Since numerical errors cannot be completely removed, the calculation verification test problems aim to quantify the numerical accuracy of the model. These test problems must be documented, accessible, repeatable, and capable of being referenced. Also required in the documentation is a record of the computer hardware used, the operating system, compiler versions, etc.

Often performance requirements are neglected while developing the domain model. Complex systems can have interactions that produce unexpected results in seemingly benign situations. Prototypes developed with small problem sets may not scale to large problems that will deal with production systems. More model detail does not necessarily generate a better answer and may make a simulation intractable. In discrete event simulation, the appropriate event queue implementation, random number generators, and sorting or searching algorithms can make a large difference in performance.

Accuracy

The evaluation of performance of the model development process has received relatively little attention. There is no clear-cut, quantitative, theoretically based standard that unambiguously defines “acceptable accuracy.” Consequently, differing explicit (or implicit) standards of accuracy exist.

The problem is magnified by a lack of agreement among scientists regarding the most suitable measures and procedures for determining (1) model accuracy (i.e. the extent to which model-predicted events approach a corresponding set of independently obtained, reliable observations (usually measured)) and precision (i.e., the degree to which model-predicted values approach a linear function of the reliable observations), and (2) the extent to which the model’s behavior is consistent with prevailing scientific theory.
The performance of a predictive model is determined by looking at its accuracy, or equivalently at its errors. Ideally, a predictive model has zero error on future data. This is almost never actually achieved, partly because models are imperfect. Even the ideal model will not have zero errors all the time. Instead, model development attempts to minimize the expected error (or risk) on future data.

Evaluation of precision and accuracy often provides the most tangible means of establishing model credibility. For this reason, the development, examination, and recommendation of methods that may be used to determine and compare the accuracy and precision of models are of primary concern and must be fully documented.