Definition and Demonstration of a Methodology for Validating Aircraft Trajectory Predictors

Robert A. Vivona*
Engility Corporation, Billerica, MA 01821

Mike M. Paglione†
FAA William J. Hughes Technical Center, Atlantic City, NJ 08405

Karen T. Cate‡
NASA Ames Research Center, Moffett Field, CA 94035

Gabriele Enea§
Engility Corporation, Billerica, MA 01821

This paper presents a new methodology for validating an aircraft trajectory predictor, inspired by the lessons learned from a number of field trials, flight tests and simulation experiments for the development of trajectory-predictor-based automation. The methodology introduces new techniques and a new multi-staged approach to reduce the effort in identifying and resolving validation failures, avoiding the potentially large costs associated with failures during a single-stage, pass/fail approach. As a case study, the validation effort performed by the Federal Aviation Administration for its En Route Automation Modernization (ERAM) system is analyzed to illustrate the real-world applicability of this methodology. During this validation effort, ERAM initially failed to achieve six of its eight requirements associated with trajectory prediction and conflict probe. The ERAM validation issues have since been addressed, but to illustrate how the methodology could have benefited the FAA effort, additional techniques are presented that could have been used to resolve some of these issues. Using data from the ERAM validation effort, it is demonstrated that these new techniques could have identified trajectory prediction error sources that contributed to several of the unmet ERAM requirements.

I. Introduction

THE prediction of four-dimensional (4D) aircraft trajectories is a foundational requirement for many advanced air traffic control (ATC) and air traffic management (ATM) concepts. In particular, it is a trajectory predictor (TP), generating 4D (position, altitude, time) trajectories for all aircraft of interest that is the engine that drives airborne and ground-based decision support tool (DST) technology. The effectiveness of DST capabilities, including conflict detection, conflict resolution, and clearance or maneuver advisory generation, is directly dependent on the capabilities and performance of the DST's TP.

With an ever increasing number of research and development efforts using DST technology to meet the demands of the U.S.'s NextGen[1] and Eurocontrol's Single European Sky ATM Research (SESAR)[2] concepts, validation of DST technology is a significant concern. Since most of these DSTs rely on aircraft trajectory prediction, validation of TP technology is similarly a significant issue. In the past, a DST's TP has typically not been validated independent from the validation of the DST itself. In the future, it is desired to develop common TP capabilities so that new DSTs do not have to provide their own customized TP. The use of common TP capabilities would require the ability to validate existing TP capabilities against new requirements imposed by each new DST application, most

[*] Chief Research Engineer, Associate Fellow AIAA.
[†] FAA Project Lead and Engineer, Senior Member AIAA.
[‡] Research Scientist, Senior Member AIAA.
[§] ATC Research Engineer, Member AIAA.

likely requiring or being greatly simplified by a separate TP validation methodology. Even when validating a DST with its own dedicated TP, the process of independently validating its TP may provide valuable insight into why the DST was unable to meet a specific TP-dependent requirement. In the extreme, TP validation may enable the validation of the DST's TP even if the DST fails to meet all of its requirements, reducing the amount of additional effort required to achieve full DST validation.

This paper presents a new methodology (Section II and III) for performing TP validation. This methodology significantly expands the concepts for TP validation originally developed by the Federal Aviation Administration (FAA)/Eurocontrol Action Plan 16 (AP16) Committee on Common Trajectory Prediction.[3] Recognizing the inherent complexities in validating a TP, the new methodology defines a set of techniques and a multi-staged procedural framework designed to reduce the effort in identifying and resolving validation failures. The techniques, most new to TP validation, were inspired by the lessons learned from a number of field trials, flight tests and simulation experiments for the development of trajectory-predictor-based automation. When applied within the new procedural framework that adds structure to the TP validation process, these techniques are designed to support TP validation in avoiding the potentially large costs associated with failures during a single-stage, pass/fail approach.

To illustrate the real-world applicability of the methodology, the validation effort performed by the FAA for its En Route Automation Modernization (ERAM) system is analyzed as a case study (Section IV). This initial validation effort identified that the ERAM system failed to achieve six of its eight requirements associated with trajectory prediction and conflict probe.[4] All requirements have since been achieved, but to illustrate how the methodology could have benefitted the FAA effort, the validation techniques used for ERAM are analyzed. Next, additional techniques from the new methodology that could have been performed to help identify and resolve validation issues prior to the completion of the initial FAA validation effort are proposed (Section V). Finally, a demonstration of some of the proposed techniques using data from the ERAM validation effort is described (Section VI). This demonstration illustrates how some of the TP issues that ultimately caused failures in the ERAM validation could potentially have been identified and rectified earlier, using less costly validation techniques, an objective of the new methodology.

II. Validation Methodology Overview

A methodology is the collection of techniques, practices, and procedures used for some discipline. This section starts by defining validation, verification and requirements in the context of TP validation. It ends by describing the core techniques of the new methodology. In Section III, these techniques are put into a new, multi-staged procedural framework that describes the use of different data sources to achieve full validation.

A. Verification & Validation

Accurate and efficient trajectory prediction is crucial for many ATM automation systems to achieve their potential for increasing the safety, security, and capacity of air transportation operations. In particular, the underlying TP produces the predicted 4D trajectories to enable air traffic service providers and aircraft operators to assess the impacts of predicted trajectories and provide advisories for meeting ATM objectives in a timely manner. In order to achieve the accuracy and performance requirements of such automation, a thorough process to validate the automation's TP is required. TP validation primarily focuses on two exercises:

1. Functional verification
2. Performance validation

Functional verification is the process that confirms the TP performs its required functions under the required conditions. These required functions are defined within the TP's functional requirements. Performance validation is the process where the TP's performance is confirmed to be within acceptable limits. Those performance limits are defined within the TP's performance requirements. The two primary forms of performance requirements are computational speed and prediction accuracy.

Validation of a TP's prediction accuracy requires additional clarification. This type of validation is the process of determining the degree to which a model is an accurate representation of the real system being modeled from the perspective of the model's intended uses. In the case of TP validation, the model is the TP's generation of a predicted trajectory and the real system being modeled is an aircraft's actual flown trajectory in the operational environment. The validation process is performed through a comparison of predicted and actual system behavior. The validity of the model is determined by the acceptability of the differences between these predicted and actual system behaviors (i.e., are prediction errors within acceptable limits). Therefore, selection of comparison metrics and the sources for the predicted and actual system data becomes a crucial step for performing the validation process. The metrics selected should explicitly characterize those aspects of the TP which have defined limits on prediction accuracy. Selected data sources, including simulations, field tests and operational data, should include or reflect appropriate error sources that would be experienced in actual operations to effectively excite model inaccuracies.

B. Direct versus Indirect TP Requirements

One of the greatest challenges for a TP validation methodology is dealing with the variety of TP requirements. There are two classes of TP requirements against which validation is performed:[**]

- Direct TP requirements
- Indirect TP requirements

Direct TP requirements are those that define explicit requirements for TP functionality or performance. Examples include required accuracy for predicted altitude profiles and trajectory response time (i.e., time from receipt of TP inputs to return of a trajectory). It is logical to expect that most TP requirements would be of this type, but unfortunately that is typically not the case.[5]

Indirect TP requirements are those that define required overall system performance for functions (e.g., DST functions) that are dependent on TP performance. Examples include required false alarm/missed alert rate for a conflict probe and, in the extreme, user acceptability requirements for a conflict probe that rely on a series of empirical tests to validate. Indirect TP requirements require sufficient TP performance to support the dependent functionality in achieving its performance requirement. Because indirect TP requirements never explicitly specify a required level of TP performance, they rely on a different validation approach than direct TP requirements. Unfortunately, it is currently common for a TP to have more (sometimes far more) indirect TP requirements than direct TP requirements, though it is not uncommon for a TP to have a mixture of requirements from both classes. As such, the validation methodology must handle both types of requirements.

C. Core Validation Techniques

The methodology uses a set of core validation techniques at different stages of the validation process. These core techniques are described below. The overall process as applied to the validation of a TP, referred to as the validation framework, is presented in Section III.

1. Use of Validation Stages

TP validation can be an expensive exercise. The collection of validation data, including high-fidelity simulation, field-test, and operational data (especially if collecting operational validation data requires deployment of a system in the field) can be, in the extreme, cost prohibitive. This reality inspired the AP16 committee to explore the development of a validation database, where validation data can be pooled to share the cost over the international community. Unfortunately, issues with the applicability of using validation data collected by a different organization create difficulties in sharing such data even under the best of circumstances. In addition, it is probable that indirect TP requirements will continue to make up the majority of requirements for many TPs, at least in the near future, and that this will increase the chances that a TP might not meet all of its requirements the first time through the validation process. This could, theoretically, require the TP to undergo the same costs multiple times to ultimately achieve validation.

To deal with these issues, the methodology was developed as a series of potentially iterative validation stages, starting with simpler, cheaper efforts that could identify significant impediments to validation before investing in more expensive efforts. If the costs for all tests were minimal, then the methodology could just recommend a single pass/fail stage (most likely using data from the actual operational environment or a high-fidelity simulation environment like at the FAA Technical Center) and the TP would iteratively attempt to pass the validation tests in this environment.[††] Because the costs of testing in these environments are significant, this could easily be a cost-prohibitive approach. Alternatively, if an inexpensive effort could either invalidate or at least indicate a high risk that the TP would not meet one or more of its requirements, then the TP could be improved before attempting to

[**] Each class of TP requirements includes all types of performance requirements, including computational and prediction accuracy performance. TP functional requirements are always direct requirements since they describe TP functionality.

[††] There are other reasons why a single pass/fail approach might not be desired even if cost were not an issue, but the cost issue is sufficient to motivate a different approach.

validate via an expensive effort. By building the confidence in the TP's ability to achieve its requirements in stages, iteration can hopefully be performed early in the cycle before the most significant costs have been incurred.

2. White Box Testing

In the context of TP validation, white box tests are those that take advantage of knowledge regarding the internal processing of the TP being validated. These tests are chosen to excite specific internal processing in the TP to achieve some objective of the validation process. Black box tests, on the other hand, do not require such knowledge and are designed to be independent of any specific TP processing. In support of a validation process, the concept is to use a series of white box tests, focusing on different levels of processing within the TP, to "build up" the validation until the entire TP is validated. This approach is primarily envisioned to support validation of prediction accuracy performance requirements.

Though the creation of specific white box tests depends on the specific TP being validated, a useful way to identify TP processing targets within white box tests is to analyze the common processing levels found in all TPs. The Common TP structure[6] as developed by the AP16 Committee is useful for this purpose. This Common TP structure, simplified and divided into three modeling levels, is presented in Figure 1. The lowest level of TP modeling is the Math Modeling level. This is where modeled aircraft behaviors are turned into mathematical equations for integration or geometric approximation. Error sources at the Math Modeling level include equation of motion approximations, aircraft performance model limitations, atmospheric modeling limitations, and selection of an inaccurate math model for the behavior (e.g., using a constant track math model when the behavior is following a great circle path). The next higher level of TP modeling is the Behavior Modeling level. This is the identification of required aircraft maneuvers (behaviors) to meet identified constraints. Error sources at the Behavior Modeling level include behavior model approximations when actual maneuver details are not known (e.g., simplified modeling of complex guidance behavior) and selection of an inaccurate behavior model for the constraint when the actual behavior is not known (e.g., using a constant vertical rate maneuver for a cruise altitude change when the pilot chooses a max climb thrust maneuver). The next higher level of TP modeling is the Constraint & Initial Condition Modeling level. This is the identification of the trajectory's initial condition and required constraints that must be achieved by processing the input state and intent data. Error sources at this level include incorrect identification of constraints from the input intent and inaccurate modeling of constraints when input intent is missing (e.g., no information on if/how to reconnect an aircraft that is currently off its input flight plan route).

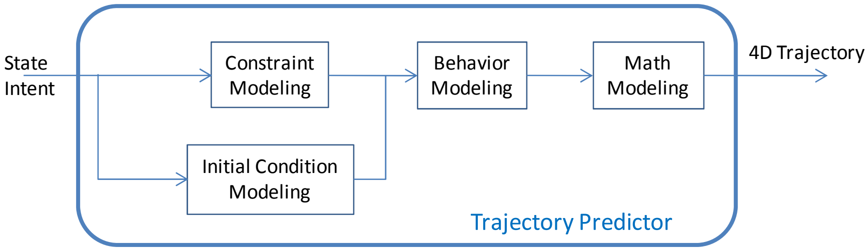

Figure 1: Common TP structure - modeling layers.

The general white-box testing approach is to isolate and test different sources of errors during the validation process so that if iteration to meet a requirement is necessary, the validation process itself supports the identification of where (e.g., at what modeling level) the TP needs to be improved. Since higher level modeling tends to add errors to those from lower level modeling, isolating errors typically focuses on separating out higher level modeling errors so that the lower level modeling of the TP can be validated first. If the lower level modeling is insufficient to meet a requirement, then no improvement to the higher level modeling will enable the requirement to be met. For example, lateral intent errors (constraint modeling) due to inaccurate route modeling (e.g., strategies for reconnecting the aircraft position to a flight plan route when the aircraft is not currently on its flight plan route) can cause significant along-path errors. These errors are in addition to along-path errors caused by other error sources that exist with or without the lateral intent error, such as errors in ground speed due to wind errors from an inaccurate atmospheric model (math modeling). If an along-path prediction accuracy requirement cannot be met solely because of wind modeling errors, then no improvements to TP's route modeling will enable the TP to meet this requirement. In this case, it is beneficial to isolate, analyze and resolve the wind modeling errors prior to dealing with the lateral intent errors. The ultimate benefit of this approach is that the relative contribution of errors at different modeling levels can be determined. This is valuable if the TP fails to meet its performance requirements, because it provides insight into where the most significant improvements can be made to the overall performance of the TP.

One theoretical approach to isolating the different modeling errors would be to literally extract the TP functions for different modeling levels (or combinations of modeling levels) from the TP, create input data for these extracted functions, and then analyze their outputs. In practice, it would be very difficult (often impossible) to extract functions that map directly to the conceptual models of TP processing in Figure 1 since actual TP implementations rarely segregate their modeling levels into separate functions. Therefore, a more useful approach is to test the entire TP while controlling which modeling level errors are excited in the input data (see Figure 2). For example, the Math Modeling level can be tested using cases that are absent of any significant errors in initial condition, constraint and behavior modeling. This would require the initial condition and constraints, as well as the resultant aircraft maneuvers, to be well defined and known from the TP inputs. For example, when the input aircraft state is on its flight plan route and the aircraft is flying that route at a known altitude and speed, the behavior of following the great circle paths and turns defined by the input route is fully known. In this situation the math modeling would be the source of trajectory prediction errors. To test the Behavior Modeling level, test cases without initial condition and constraint errors would be used. An example would be a change in cruise altitude where the final altitude (constraint) is known but the maneuver to achieve this altitude is not known. Behavior model testing includes the effects of errors in the math models associated with those behavior models.

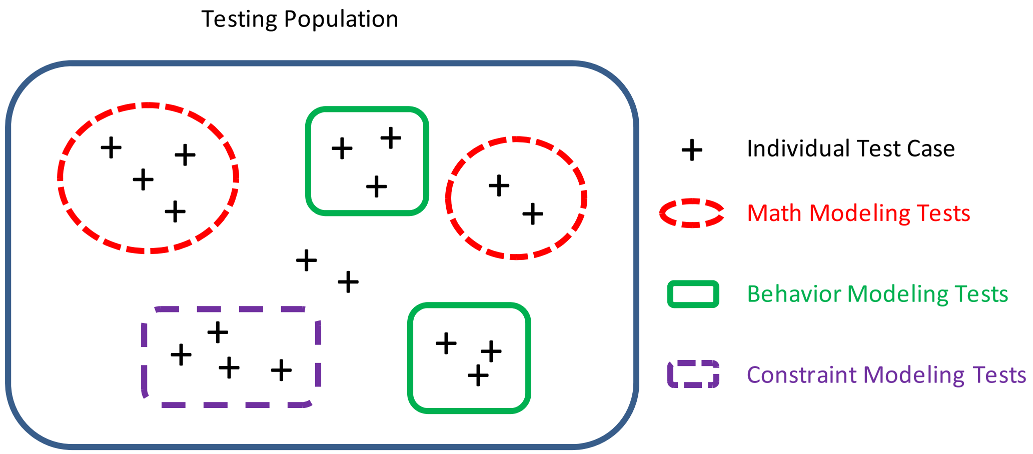

Figure 2: Test case set divided into modeling-level subsets.

A rigorous process for applying this approach would be to start testing at the lowest modeling level (math modeling) and work up each modeling level until either:

1. The given modeling level fails to meet the requirements, or
2. The given modeling level is too close to failing the requirement to believe higher levels will succeed, or
3. All levels of modeling have been tested and have met the requirements.

In practice, the use of white box testing doesn’t need to rigorously follow the above approach. In other words, all lower level modeling does not need to be validated before moving to the next modeling level. It is expected that white box tests that cover a few major modeling-level issues will be designed to isolate specific error sources of concern. For example, the prediction accuracy of the TP without lateral intent errors (as described above) may be validated prior to validating the TP with these errors to isolate the impact of this particular error source on the TP’s prediction accuracy.

3. Test Bench Testing

Test bench testing is the process of testing a TP independent of its client system by feeding input data directly into the TP’s interface and evaluating the TP’s output. The major benefit of this approach is that specific, controlled inputs can be sent to the TP and the resultant outputs evaluated. Test bench testing can be used for both black box and white box testing, though only direct TP requirements can be validated since the entire automation system is not run in the test bench environment. Test benching testing is expected to provide different levels of benefit to the three main validation efforts:

Functional Verification: Since functional verification is the process of verifying that the TP performs its required functions under specified input conditions, test bench testing is ideal for this effort. TP inputs for each specified condition can be created directly without the need to develop a client system-level (operational) scenario. Since no actual trajectory data[‡‡] (as needed for prediction accuracy analyses) is required, the TP output can be evaluated directly for each specified scenario to verify proper TP functionality.

Prediction Accuracy Validation: Prediction accuracy validation requires both TP predicted and actual trajectories for a given aircraft, so it is unlikely that initial prediction accuracy efforts will use test bench testing as an approach. Where test bench testing is expected to support prediction accuracy validation is during iteration after a TP has failed to meet a prediction accuracy requirement. High-fidelity validation tests, including high-fidelity simulation, field-test, and operational data tests are typically expensive in terms of resources and cost. If a TP fails to meet its performance requirements during one of these tests, it would be expensive to rerun the test with an updated version of the TP to validate the new TP performance. With proper planning, it is possible to use test bench testing to replace the need to rerun the high-fidelity test. For example, if during the first running of the high-fidelity test, sufficient TP input data to rerun the TP in a test bench environment and the resultant actual trajectory data is stored, then this data can be used to run a test bench test with the same fidelity as the high-fidelity test for validation purposes. The new TP would be given the same input data as the previous TP was exposed to during the high-fidelity test and the resultant predicted trajectory would be compared to the actual trajectory recorded during the high-fidelity test. In the majority of cases, the results should be equivalent to rerunning the high-fidelity test with the new TP.[§§]

Computational Performance Validation: Though computational performance validation requires full client-system level testing to ultimately validate (since TP computational performance is not typically independent of client system impacts), test bench testing can provide some benefit. Test bench testing can be used to measure the TP's computational speed under controlled conditions (i.e., for a specific set of inputs) without the need for expensive validation data since actual trajectory data is not required. Hence, these tests are often less expensive to run than full client system level tests. A TP whose performance under test bench conditions is either not acceptable or only marginally so has a high risk of not being acceptable under full client system level testing. Test bench testing can be used to run quick, cheap initial computational performance tests or to roughly check computational performance when rerunning high-fidelity validation results during TP capability iteration to meet a requirement.

Though not all current TP's can be extracted from their client systems to be tested in a test bench environment, it is expected that if test bench testing proves to be a highly effective validation technique, future TPs will be designed with interfaces that enable this testing approach.

III. Validation Procedural Framework

The procedural framework for applying the new validation methodology is illustrated in Figure 3. The process begins with functional verification and then proceeds through a series of performance validation stages of increasing fidelity and, typically, cost. Not all stages are required to be performed. Depending on the operational concept of the TP's client system, some levels of testing may not even be feasible. For example, current-day operational data will often not provide suitable validation data for a TP that will require input data from a future operational environment (e.g., clearance information that exists only after the TP's client system has been deployed).

At any verification/validation stage, if the TP either fails or is too close to failing (e.g., has a high risk that at a later stage will fail) to meet a performance requirement, then the TP needs to be revised before proceeding to later stages of testing. If the TP revisions invalidate any previously performed validation efforts, then these tests should be rerun and additional TP capability revisions performed, as required, until the current TP capabilities achieve acceptable results.

The methodology assumes that, by default, the testing will either treat the TP or the entire client system (including the TP) as a black box. White box testing is performed only if:

1. It is believed that the TP has a high risk of failing the current validation stage
2. The TP has failed a validation stage (i.e., during TP revision efforts)

In the former case, the black box test cases can be performed in a sequence based on "white-box" knowledge, isolating specific modeling layers in order to isolate potential error sources. The only difference between white box and black box testing in this case may just be the order in which the test cases are performed. In the later case, more

[‡‡] An "actual" trajectory consists of a series of sensed aircraft states as the aircraft flies consistently (or inconsistently) with the clearances associated with the intent inputs for a predicted trajectory.

[§§] In cases where the inputs to the TP are changed to improve the accuracy of the TP, these new inputs would need to have been available and recorded during the high-fidelity test so the new TP inputs can be created for the test bench test.

detailed white box testing may be used to help identify the appropriate modeling layer changes to enable the revised TP to pass the failed validation stage.

Details of the specific verification/validation stages are described next.

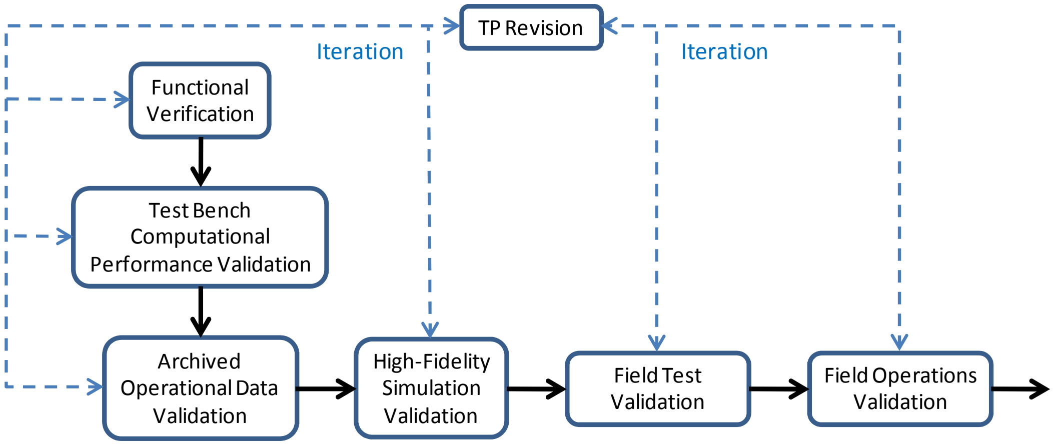
TP Revision
Iteration
Iteration
Functional Verification
Test Bench Computational Performance Validation
Archived Operational Data Validation
High-Fidelity Simulation Validation
Field Test Validation
Field Operations Validation

Figure 3: Framework for applying validation methodology.

A. Functional Verification

Functional verification testing is where the functional response of the TP to specified inputs is compared to the functional requirements to verify that the TP is responding in a required manner. Functional verification should be performed prior to any validation tests because TP capability changes required to achieve functional verification have a high potential to invalidate the results of previously performed validation tests. Test bench testing is the preferred approach for functional verification because in this testing approach, the functional response of the TP[***] is a direct result of the test (TP) inputs. During test bench testing, inputs can be created that, based on the functional requirements, which may be directly defined for specific TP inputs, require a specific TP response. The TP can then be given those inputs and the output of the TP analyzed to determine whether the TP's functional response was acceptable.

Verification testing does not require a specific operational fidelity of input data. On the contrary, verification testing may require inputs that are rare in actual operations (e.g., only possible in significantly degraded situations) to excite the desired functional response. Verification testing also does not require actual trajectory data for comparisons. Only the TP outputs are required, since they can be evaluated directly against the functional requirements.

B. Test Bench Computational Performance Validation

Test bench testing can be used to measure the TP's computational speed under controlled conditions, i.e., for a specific set of inputs. These tests are often less expensive to run than full client system tests and can be valuable in identifying potential computational speed issues. Since these tests do not need high-fidelity validation data (just representative input data is required), they can be run before the more costly prediction accuracy validation tests.

There are two main issues in using test bench testing for computational performance validation. The first is that only direct TP requirements can be tested, since the TP is extracted from the rest of the client system during these tests. The second is that these tests are typically not conservative in their assessment of TP computational speed since other client system processing that would impact the TP computational performance in the operational environment (e.g., through shared processors) are not included. Even with these limitations, test bench testing can be an effective tool in estimating whether the TP will meet its computational performance requirements prior to performing higher fidelity, more costly testing methods. Certainly, if the TP performance during test bench testing does not meet the computational requirements, it is safe to conclude that the TP will fail to meet these requirements

[***] The "functional response of the TP" can be measured in many ways, and the specific response measurement used is dependent on the type of testing being performed (e.g., white box or black box) and the particular functional requirement being verified. For example, a black box test of a TP given a new cruise altitude input may verify that the trajectory contains a cruise segment at this new altitude in the output trajectory.

in the operational environment. Also, if the impact of system processes on the TP performance can be estimated (e.g., will decrease the TP performance by approximately 10%), then the test bench testing can provide an estimate of whether the TP will meet its computational requirements.

One additional benefit of test bench testing is the ability to perform extreme condition testing, where the TP is run through scenarios where computational loads are predicted to be at their highest. These conditions may be difficult to create during client system testing. Test bench testing, given the caveats described above, may provide the only approach to test these conditions.

C. Archived Operational Data Validation

Data from several current operational systems (e.g., ARTCC Host systems, ETMS) are regularly recorded by the FAA to keep an historical record of the National Airspace System (NAS). This data is used by the FAA and other organizations to perform offline analyses, often to baseline current-day NAS performance. The benefit to TP validation is that this recorded data typically includes both the TP's client system inputs (e.g., flight plans, altitude clearances) as well as the actual trajectory data (e.g., radar tracks) for aircraft whose trajectories need to be predicted by the TP. Often the operational data that is recorded are also the input data sources for new automation technologies added to the NAS. If this data contains enough client-system-level inputs to run the system in standalone mode then it is possible that computational load performance validation can be performed for both direct and indirect TP requirements. If the data also includes actual aircraft trajectory data, then it may be possible to perform prediction accuracy performance validation for both direct and indirect TP requirements. Since the data is regularly recorded by the FAA, use of this data is an inexpensive alternative to collecting validation specific operational data from field tests or operational facilities.

A possible limitation for performing prediction accuracy validation from archived operational data is the impact of controller actions on the actual trajectories. Since the data was recorded from real operations, this means that the actual aircraft were impacted by clearances issued by the controllers at that time. Many of these clearances are not directly reflected as changes to today's operational data, though they impact the aircraft's actual trajectory. For example, a verbal clearance for an aircraft to intercept its flight plan route at a downstream waypoint may not be reflected as a change to that aircraft's flight plan. If the TP being validated would have access to intent sources beyond those available or archived today, then using archived operational data would add an additional error source that would not exist in the TP's operational environment, making this data unacceptable for its validation.

The existence of these controller-issued clearances also impacts the acceptability of using archived operational data for the validation of indirect TP requirements related to a conflict probe. For example, to effectively determine the false alarm rate of a conflict probe, the aircraft must be allowed to fly unimpeded through predicted conflicts to see if a loss of separation actually occurs. Any clearances issued to the aircraft prior to reaching the predicted loss of separation will eliminate the ability to determine whether predictions before the clearance accurately or inaccurately predicted a conflict. Since new conflict probe technology should not be less accurate than the existing technology, it is expected to be rare that a controller would not issue a clearance to resolve a conflict detected by the new conflict probe. Hence, it is unlikely that the archived operational data would be useful for validating an indirect TP requirement based on conflict probe false alarm rate. Some TP validation efforts (see Ref. 7) have gotten around these limitations by using time-shifted operational data to test indirect TP requirements for conflict probes. This approach takes raw operational data and then adjusts specific aircraft by shifting all of their data (e.g., actual trajectory, clearances) in time to artificially create conflicts that did not exist in the real operational environment. This is actually closer to a high-fidelity simulation approach than an archived operational data approach since the raw operational data is manipulated to create TP validation events (e.g., conflicts). This time-shifting approach, if applicable, could be used as a lower cost alternative to high-fidelity simulation.

Archived FAA data is not the only source for archived data that could be used at this stage. If a validation database, such as the one proposed by the AP16 group, were available, then this data would be suitable for prediction accuracy validation at this stage. It is assumed that validation database data would only be useful for direct TP requirements since its purpose would be to support TP validation independent of any particular automation system.

D. High-Fidelity Simulation Validation

The Test Bench Computational Performance and Archived Operational Data Validation efforts previously described represent opportunities to perform inexpensive validation efforts to limit the chance of validation failure during more expensive validation efforts. As discussed above, they will not be appropriate for all validation efforts, but can be effective under the right conditions. The use of simulation, though significantly more expensive than these previous two validation efforts, should be an appropriate approach for most, if not all TP validation efforts and can significantly reduce the risk of validation failure during field-test or field-operations validation. Simulation used for validation purposes requires a sufficiently high level of fidelity in modeling aircraft motion and/or the operational environment to be used as a surrogate for actual operational data. The required simulation fidelity for validating different performance requirements is discussed below.

The major benefit to simulation is the added control over error sources. In a simulation environment, the number of error types and magnitudes can be controlled, enabling a wide range of error situations to be covered by a much smaller number of validation runs. In the actual operational environment, either some (during field-test validation) or all (during field-operations validation) of the error sources are uncontrolled. In operational environments, it may take a large number of runs to capture rare events (e.g., combinations of errors), which may make it impractical to try to validate the TP in this environment for those situations. This makes high-fidelity simulation a preferred environment for testing less-common and extreme conditions when those conditions occur infrequently in the actual operational environment. The TP may even be validated in high-fidelity simulation under greater than maximum error cases to provide a margin of error for unexpected, extreme situations in the operational environment. Aside from the costs, the major disadvantage to using high-fidelity simulation is that not all of the error sources in the operational environment may be well understood. This makes modeling these errors challenging. This is especially true for modeling highly-correlated error sources that tend to occur in tandem.

The type of high-fidelity simulation used for validation is dependent on the class and type of performance requirement being validated. The requirements for high-fidelity simulation are presented in Table 1 for each combination of direct/indirect and computational/prediction accuracy requirement.

Table 1: Type of simulation environment by requirement class and type.

	Computational Performance Requirements	**Prediction Accuracy Performance Requirements**
Direct TP Requirement	High-Fidelity Operational Environment Simulation	High-Fidelity Aircraft Simulation
Indirect TP Requirement	High-Fidelity Operational Environment Simulation	High-Fidelity Operational & Aircraft Environment Simulation

1. Computational Performance Requirements

Computational performance, whether for direct or indirect requirements, requires a simulation environment that closely emulates the actual operational environment, including a high-fidelity simulation of:

- Client system data load – creates proper impact of client system processing on TP processing via shared processors, etc.
- Client system TP triggers – different uses of the TP by the client system may have different speed requirements
- Computationally intensive TP scenarios – e.g., exciting constraint relaxation or conditional constraint handling
- Hardware configurations

The simulation must accurately model the impact of interacting TP and client system processing on shared hardware resources (processors, memory, etc.). Since the aircraft actual trajectory data is not relevant for computational performance validation, the aircraft simulation can be of any fidelity level that suits the above simulation requirements. Typically, a high-fidelity aircraft simulation should not be required.

High-fidelity simulation facilities that simulate ATC/ATM operations, such as those at the FAA Technical Center, are appropriate for this stage of TP validation. Lower-fidelity environments could also be used, if they can meet the above requirements. For example, a standalone version of the client system, running on deployment-level hardware and fed by proper data to emulate the conditions above, could meet these requirements. For indirect TP requirements, the simulation environment must be able to simulate enough of the operational environment so client system metrics of performance can be properly measured.

2. Prediction Accuracy Requirements

For validation of direct TP prediction accuracy requirements, a high-fidelity aircraft simulator should be used to generate simulated actual trajectory data for comparison against the TP predicted trajectories. It is assumed that a high-fidelity operational environment simulation (see previous) would not be required, but some mechanism for creating the proper TP inputs would need to exist. Potentially, test-bench testing techniques could be used to validate the TP for these requirements. If high-fidelity TP inputs cannot be created in any other way, then a high-fidelity operational environment simulation which includes a high-fidelity aircraft simulation is required.

The goal of prediction accuracy validation is to validate that the TP's prediction error (PE) when predicting an aircraft's actual trajectory in the real operational environment ($PE_{TP\rightarrow Actual\frac{A}{C}}$) is within required limits. If simulation data is being used for validation, then the best estimate of the TP's PE that can be measured is with respect to the simulated aircraft's actual trajectory ($PE_{TP\rightarrow Simulated\frac{A}{C}}$). The difference between $PE_{TP\rightarrow Actual\frac{A}{C}}$ and $PE_{TP\rightarrow Simulated\frac{A}{C}}$ is the trajectory error (TE) between the simulated aircraft's actual trajectory and the actual trajectory of the aircraft in the real operational environment:

$$PE_{TP\rightarrow Actual\frac{A}{C}} = PE_{TP\rightarrow Simulated\frac{A}{C}} + TE_{Simulated\frac{A}{C}\rightarrow Actual\frac{A}{C}} \qquad (1)$$

With respect to TP validation, a high-fidelity aircraft simulator is one in which the simulator's $TE_{Simulated\frac{A}{C}\rightarrow Actual\frac{A}{C}}$ is low enough such that the ability of the TP to meet its requirements using simulated data ($PE_{TP\rightarrow Simulated\frac{A}{C}}$) is a good indicator that the TP will meet its requirements when using operational data ($PE_{TP\rightarrow Actual\frac{A}{C}}$). If $TE_{Simulated\frac{A}{C}\rightarrow Actual\frac{A}{C}}$ is too large, then TP validation testing using simulated data may not accurately reflect the performance of the TP in the actual operational environment.

For validation of indirect prediction accuracy requirements using high-fidelity simulation, all of the requirements described for computational performance requirements also apply. An additional requirement is that the simulation environment must also provide a high-fidelity aircraft simulation to provide accurate actual trajectory data. This can be difficult to achieve, in practice, if the high-fidelity operational environment requires the simulation of many aircraft simultaneously. It is typically not practical to hook up a large number of high-fidelity aircraft simulators to such an environment. In this case, the fidelity of the available aircraft simulator will need to be assessed (i.e., assessing the magnitude of the TE term in equation 1) to determine whether prediction accuracy validation can be performed. If this validation stage is used primarily to reduce the risk of failing during later field validation efforts, then the use of lower fidelity aircraft simulators may be acceptable for prediction accuracy validation at this stage.

E. Field-Test Validation

The goal of field-test validation is to test the TP's capabilities under near-operational, but highly observed and controlled conditions. Though validation efforts are performed within the actual operational facility in which the TP's client system will operate, the environment is characterized as only near-operational because the procedures used are specific to the test and do not completely represent current operational procedures. Therefore, field-test TP validation typically includes:

- temporary implementation of the TP (or its client system) within an operational facility
- collecting operational data (e.g., radar tracks, Host flight plan data) directly from the facility's operational systems
- using actual commercial (with line pilots) or test (with test pilots) aircraft
- using controlled operational test procedures that may be outside the scope of current-day operations

Though not always necessary, this field-test validation can be a critical component to an overall validation approach. The advantage of this type of validation is that it enables testing to be performed under somewhat controlled operational conditions. While some variables such as wind magnitude and direction can only be observed, other critical conditions such as pilot procedures can be controlled. One disadvantage of field testing, due to the nature of controlling the operational environment, is the limited number of test scenarios that can be studied. Finally, though not necessarily as expensive as full field-operations validation, field-test validation is typically resource intensive, often including large test teams and significant facility coordination efforts. Therefore, field-test validation is primarily used when the desired operational scenarios can be properly defined and controlled in a field test and either the:

- cost of performing the field tests is significantly less than the cost of potentially failing the operational tests, or
- the operational scenario is difficult or impossible to create under current-day operations

One main use of field-test validation is to isolate high risk scenarios in the validation process before committing to full operational validation. These can include:

- extreme condition testing – scenarios which put maximum stress on TP capabilities, creating a high risk of unacceptable TP performance
- safety critical testing – scenarios for which extra safety procedures may be required to remove potential risks to aircraft safety

Extreme condition scenarios may occur rarely in actual operations. If a critical scenario can be created in a field-test environment, then field-test validation enables this case to be examined without needing to wait for the situation to occur in normal operations (a potential cost savings).

Another use of field-test validation is to validate the TP under the sort of operational conditions that will exist only after its client system is deployed. If the TP's client system supports a new ATC/ATM operational concept that does not exist today, then the only way to validate the TP's performance in this new environment with operational-level fidelity is to create, under controlled conditions, a limited version of this environment within an operational facility. This is exactly the environment created during field-test validation.

F. Field-Operations Validation

Field-operations validation is performed in the actual operational environment in which the TP's client system will operate. Similar to field-test validation, these efforts are performed within an operational facility and data is collected directly from the facility's operational systems. The distinction between field-operations and field-test validation is that field-operations validation is based only on the use of actual aircraft (e.g., commercial traffic) and actual operational procedures (no test aircraft or test procedures). This may limit field-operations validation to only those TP client systems which do not alter or significantly alter current-day operations.

The goal of field-operations validation is to validate the TP's performance in its actual deployed environment, the ultimate level of validation fidelity. The major limitation of field-operations validation, other than its reliance on current-day procedures, is the inability to control the environment variables. Since all of the operational data is based on actual operations, the test scenarios are limited to whatever conditions occur the day of the testing. To capture all of the desired test conditions, particularly extreme conditions, is unrealistic due to time and cost constraints. It can also be difficult to capture large amounts of data for statistically significant results if the validation efforts impact the facility operations in a significant way. For these reasons, it is just not practical to perform all validation at this level. It is possible that in the extreme, this level of validation will only be used to "confirm" the results of the previous validation tests, i.e., validate the validation process. This would occur if the numbers of validation scenarios far exceed what can be captured during field-operations validation. Therefore, the explicit role of this validation level is dependent on the TP's client system and its operational environment.

IV. Case Study: ERAM TP Validation

To illustrate the real-world applicability of the methodology, a recent validation effort performed by the FAA is analyzed as a case study. The FAA is deploying a new ATC system to replace both the existing Host Computer System (HCS) and its decision support tool in the en route domain, called the User Request Evaluation Tool (URET). The replacement system is called ERAM (En Route Automation Modernization). The formal Factory Acceptance Test (FAT) to validate ERAM performance was conducted in September 2007.[4] The focus of the FAT was on two key functions of ERAM: trajectory prediction and the prediction of future losses of separation between two aircraft or between an aircraft and a Special Use Airspace (SUA). The validation tests used metrics defined to measure the ability of ERAM to perform these functions and an extensive set of computer analysis tools, developed over several years, to quantify these metrics. These metrics and tools, documented in Ref. 4, had also been effectively used in past FAA validation efforts, particularly for URET.

The performance requirements used to validate ERAM during the FAT were derived from the requirement that the ERAM Flight Data Processing (FDP) and Conflict Probe Tool (CPT) subsystems must perform at least as well as the legacy HCS and URET. As summarized in Table 2, the principal requirements were altitude prediction accuracy (requirements FDP9389 and FDP9390), strategic missed conflict alert rate and strategic false conflict alert rate (requirements ERD1879-C3 through C6), and warning time metrics (requirements ERD1879-C1 and C2). Unfortunately, ERAM passed only two of the eight requirements during the formal test. Detailed results of the FAT are documented in Ref. 4. Because it failed a significant number of its validation tests, the FAA initiated an effort to correct the ERAM deficiencies through iterative modifications to the ERAM software. Over the following year, ERAM was modified and re-evaluated until finally in November of 2008, ERAM passed all of its requirements.

Table 2: ERAM testing results from the original FAT[4].

Requirement Number	Description	Requirement	ERAM Result	Met Requirement	Requirement Type
FDP9389	Vertical trajectory accuracy – altitude for level flight	0.0016	0.0088	No	Direct
FDP9390	Vertical trajectory accuracy – altitude for non-level flight	0.1431	0.1785	No	Direct
ERD1879-C1	aircraft-to-aircraft immediate conflict prediction warning time > 10 min.	854 seconds	740 seconds	Yes	Indirect
ERD1879-C2	aircraft-to-aircraft immediate conflict prediction warning time < 10 min.	104 seconds	128 Seconds	Yes	Indirect
ERD1879-C3	aircraft-to-aircraft missed conflict alert rate	0.025	0.067	No	Indirect
ERD1879-C4	aircraft-to-aircraft false conflict alert rates	0.16, 0.28, 0.007, 0.005	0.23, 0.063, 0.016, 0.008	No	Indirect
ERD1879-C5	aircraft-to-airspace missed conflict alert rate	0.02	0.062	No	Indirect
ERD1879-C6	aircraft-to-airspace false conflict alert rates	0.14, 0.007, 0.01, 0.003, 0.003	0.08, 0.01, 0.002, 0.001, 0.03	No	Indirect

Lessons learned from the ERAM validation effort indicated that the approaches taken to achieve ERAM validation, though ultimately successful, could certainly be improved. This paper proposes specific techniques from the new validation methodology that had the potential to improve the ERAM validation effort. These techniques are presented in the next section. To provide some context, it is useful to first identify how the approaches actually used during the FAT relate to the techniques and framework of the new methodology.

The first two requirements in Table 2 relate to the altitude accuracy of the ERAM TP, making them direct requirements. For these two requirements, the FAA used data recorded from the HCS and the primary radars to run ERAM and compare the ERAM TP predictions against recorded actual aircraft data. This is an example of Archived Operational Data Validation from Figure 3. For the indirect conflict probe requirements, the operational data were altered, by adjusting the message times (i.e. time shifting) for each individual flight by a constant, to create known conflict situations.[7] Since the recorded operational data had been impacted by air traffic control precisely to maintain aircraft separation, using the unaltered operational data to validate the performance of the CPT would have been difficult. Using the time shifting process to generate conflicts enabled the validation team to evaluate the performance of the CPT under controlled conditions. This is an example of a High-Fidelity Simulation Validation (Figure 3) using operational data. Even though operational data is used, because the conflict events are "created" through the alteration of this data, the validation is actually using simulated conflict events. This simulation approach creates a high-fidelity simulation because the time shifted archived operational data retains many of the idiosyncrasies that occur in the field (e.g. surveillance radar inconsistencies, message delays, etc.). All testing performed for the FAT followed black box testing procedures. However, the iterations and follow-on testing that took place after the FAT did include some additional white box techniques to identify error sources.

During ERAM development for the FAT and through the iterations leading to ultimate validation, the process of identifying and resolving trajectory and conflict prediction errors was a costly effort, taking years of iteration to reach acceptable levels. With limited existing methods for approaching TP validation, the methods employed by the ERAM team were developed ad hoc and were not always systematic in their approach to identifying errors. In hindsight, the new validation methodology offers several techniques for structuring validation tests to support systematic identification and resolution of prediction errors, which have the potential to improve the ERAM iteration process to meet its requirements. Section V describes several techniques from the methodology that could have been applied to the ERAM FAT and Section VI presents quantitative examples of their applicability, retrospectively, using actual data from the FAT.

V. Proposed Techniques for ERAM TP Validation

At the time of the formal test in 2007, six out of the eight ERAM performance requirements were not achieved. As an outcome of these results, the test team, composed of the development contractor and FAA participants, continued to correct issues and verify the automation against the requirements until the system passed all of its requirements approximately one year later. This was a costly exercise, but necessary to ensure ERAM performed correctly before deployment. The new validation methodology uses techniques designed to identify and resolve issues that could potentially result in failed requirements as part of the validation process. The application of additional techniques from the new methodology may have, at minimum, enabled the identification and resolution of problems in ERAM sooner, with less cost, and probably would have resulted in a more positive FAT outcome.

Evaluating the approaches used by the test team, it was determined that the following additional techniques from the methodology had the potential to improve the effectiveness of the ERAM validation effort:

1. Perform the validation in stages, starting with validation of requirements whose success/failure reduces risk in validating later requirements
2. Apply "white box" testing techniques during validation testing, not just during iterative development
3. Use test bench testing techniques during iterative development to successfully meet the requirements

Due to the generalized nature of the methodology, it should not be assumed that this is the only set of additional techniques that could have been applied. The selection of specific techniques is left to the judgment of the group responsible for validation; these additional techniques should be viewed as one possible application of the methodology. It should also not be assumed that the lack of these (or similar) techniques implies the original validation was performed incorrectly. The goal of the methodology is to apply techniques that reduce the overall effort in achieving successful TP validation. The end result of a successful validation, as was the case for ERAM, should be the same in all cases.

Three detailed examples of applying these techniques are discussed next. The first example describes the benefits of ordering the requirements for testing. The next two examples discuss specific applications of white-box and test-bench testing techniques for validating the two direct requirements.

A. Order the Requirements for Validation

This technique focuses on performing the validation of the requirements in a specific order, as opposed to performing the validation for all requirements simultaneously as was done in the original ERAM validation effort. Specifically, the validation effort should:

- Validate the direct requirements (FDP9389 and FDP9390) before the indirect requirements
- Validate the direct requirement for level flight (FDP9389) before the direct requirement for non-level flight (FDP9390)

The approach is to validate the requirements in sequence, iterating on the TP capabilities at each stage (if necessary) until the current requirement is met before moving on to validate the next requirement. The main advantage of this approach is that by ordering the requirements based on potential error sources, problems identified and resolved in achieving earlier requirements can remove potential risks in achieving later requirements. For ERAM, since TP altitude prediction accuracy directly impacts acceptable conflict probe performance, the TP altitude prediction accuracy requirements (FDP9389 and FDP9390) should be validated before the conflict probe performance requirements.[†††] It is not guaranteed that failure to meet the altitude prediction accuracy requirements will cause the conflict probe performance requirements to fail, but it certainly increases the risk of this failure. Similarly, any improvement to the altitude prediction accuracy of the TP should only improve the conflict probe performance. Since both direct requirements failed to be met and four of the six indirect requirements also failed to be met (see Table 2), iteration to meet the direct requirements has the potential to either achieve one or more of the failed indirect requirements or at least reduce one error source's (vertical prediction inaccuracy) contribution to these requirements not being met. An additional benefit of focusing on the direct requirements first is that it takes less effort to perform this validation, since no additional processing on the operational data (time-shifting) is necessary for testing or retesting the direct requirements.

[†††] In general, it is expected that direct requirements should be validated before indirect requirements since direct requirements are only impacted by TP performance and indirect requirements are dependent on TP performance and other factors.

Furthermore, the requirement for level flight TP prediction accuracy (FDP9389) should be validated prior to the requirement for non-level flight (FDP9390). If the TP doesn't achieve the vertical accuracy requirement for level flight, then one of the following occurs more often than is acceptable:

- The actual trajectory is level when the predicted trajectory is in transition (climb or descent), or
- Both the actual and predicted trajectories are level, but at different altitudes

In both cases, the error is most likely caused by incorrect intent modeling. In the first case, it is possible that level errors could be caused by climb or descent rate errors (e.g., inaccurate aircraft performance or wind models causing predicted top-of-climb or top-of-descent errors), but intent errors should dominate these cases. Identifying and resolving the level flight errors have the potential to also positively impact the TP accuracy for non-level flight (FDP9390). On the other hand, TP vertical prediction accuracy when non-level is significantly impacted both by intent errors (e.g., predicted level flight when the aircraft is actually climbing or descending) as well as other modeling errors (e.g. inaccurate modeling of climb or descent rate). Therefore, it would be preferable to validate the non-level flight requirement after the level flight requirement has identified and resolved as many intent modeling errors as possible. This isolation of the (primarily intent) errors in the level flight cases is a form of white box testing.

B. White Box and Test Bench Testing Techniques: Phase of Flight Analysis

The application of white box techniques to isolate error sources and test bench techniques to efficiently iterate on TP capabilities to achieve required ERAM performance can be illustrated by the application of these techniques to achieving the first direct requirement: FDP9389. Requirement FDP9389 is focused on the altitude accuracy of ERAM's TP for flights in level segments. Since the metric focuses on those segments where the aircraft's actual trajectory is level, these segments can belong to either the aircraft's cruise, descent or climb phase of flight. In the climb and descent phases of flight, level segments are typically caused by interim altitude clearances that level the aircraft off before reaching its final altitude (cruise altitude for climbs, waypoint defined constraint altitudes for descents). Errors in handling these interim altitudes, including inaccurate models of when the aircraft will be released from an interim altitude constraint or missing/incorrect prediction of procedurally defined interim altitudes, can impact the TP's ability to meet the level flight accuracy requirement. In the cruise phase of flight, the errors in predicting actual level flight segments is expected to be less likely, since the altitude constraint is typically the flight-plan-defined cruise altitude. Since the error sources vary based on the different phases of flight, it is beneficial to identify the phase of flight to which each actual level flight segment belongs. By adding this additional piece of information to the validation analysis, this allows the ERAM level flight accuracy (LFA) performance metric to be divided into three sub sets of data, one for each phase of flight:

$$LFA_{Total} = LFA_{Climb} + LFA_{Cruise} + LFA_{Descent} \tag{2}$$

Then, if ERAM performance fails to meet the total level flight accuracy requirement, the data can immediately identify which phase of flight contributed the largest accuracy errors.[‡‡‡] The benefit of this white box technique is that it helps separate the sources of error in the analysis, which helps in identifying which error sources should be focused on in the TP capability iteration process.

Moreover, inputs to the TP should also be recorded when running the initial full operational data test[§§§] in order to enable the TP to be run in a test bench fashion during any required TP capability iterations to meet the requirement. From the white box testing results, the phase of fight that contributed most to the requirement failure is identified; a small representative set of these flights can then be down-selected and run through the test bench TP for each TP capability modification until the altitude accuracy of these cases reaches a target level that improves the chances that the requirement will be met.[****] Due to the smaller effort to run the limited number of test bench cases (as opposed to the full system test), this iteration process can be quickly run multiple times, as necessary. After the

[‡‡‡] The FAA metric is actually (see Ref. 4) the ratio of failed level flight segments (i.e., those which exceed an acceptable altitude error limit) to total level flight segments, so "contributed the largest accuracy errors" in this context means contributed the most failed level flight segments.

[§§§] It is desired to initially run the full set of (not time-shifted) operational data through ERAM to test whether FDP9389 is met, since no additional effort would need to be performed if ERAM met this requirement.

[****] Choosing an appropriate target level requires engineering judgment based on the specific situation. The target level, when achieved, represents when iteration on the representative set should end and re-evaluation of the TP performance for all flights is required.

test bench tests are completed, the full scenario with all the flights must then be run to determine whether the iteration has successfully met the level flight requirement. If not, the process is performed again using the latest validation results as a starting point. Using test bench iteration, the effort of iterating to meet the requirement should be reduced over running the full system test each time the TP is modified.

C. White Box and Test Bench Testing Techniques: Lateral Intent Error Analysis

After requirement FDP9389 is met the validation moves to the next direct requirement, FDP9390. This requirement is focused on the altitude accuracy of ERAM's TP for aircraft in non-level flight segments. The metric focuses on all the flights whose actual trajectory is transitioning, either climbing or descending. Transitioning flight can exist in any of the three phases of flight: climb, cruise (during a cruise altitude change) or descent. Errors affecting the accuracy of the TP in non-level flight include error sources in addition to those in level flight. Aircraft performance modeling errors of climb and descent rates heavily affect the vertical accuracy of the TP in predicting transition segments. Moreover, lateral intent modeling errors can cause significant altitude errors for the prediction of transitioning phases of flight. For example, an aircraft given a direct route to its arrival fix without a flight plan change (causing a lateral intent error if the predicted trajectory follows the flight plan route) would initiate its top-of-descent earlier than predicted (in time and downrange along-path distance due to the shorter along-path distance to the arrival fix). This would cause an altitude error after the aircraft enters the descent phase of flight (Figure 4).

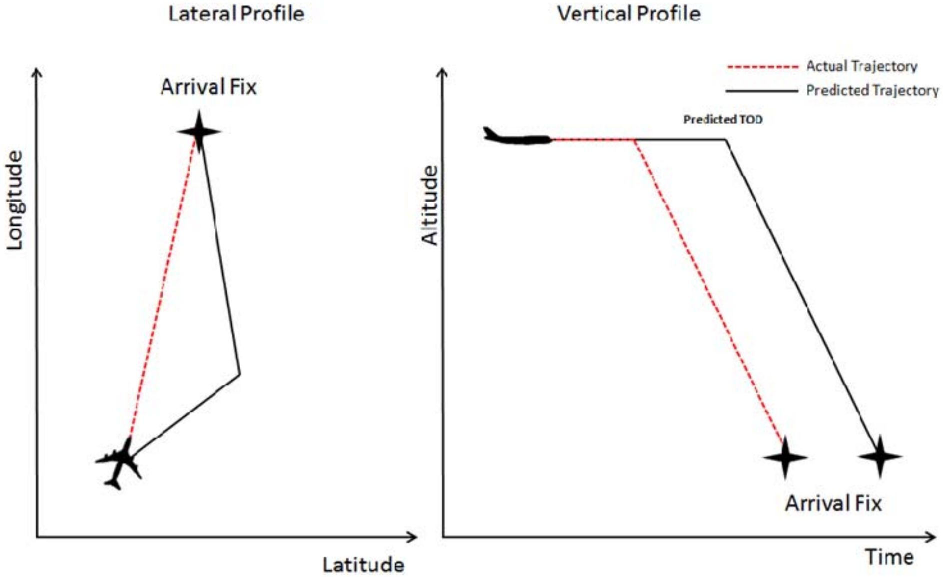

Figure 4: Impact of lateral intent modeling error on vertical accuracy.

Therefore, analyzing non-level flight accuracy (NLFA) in each phase of flight, but now separating this analysis into those flights with and without lateral intent error would isolate this additional source of error:

$$NLFA_{Total} = NLFA_{Lateral\ error} + NLFA_{No\ lateral\ error} \tag{3}$$

$$NLFA_{Lateral\ error} = NLFA_{Climb,lat\ err} + NLFA_{Cruise,lat\ err} + NLFA_{Descent,lat\ err} \tag{4}$$

$$NLFA_{No\ lateral\ error} = NLFA_{Climb,no\ lat\ err} + NLFA_{Cruise,no\ lat\ err} + NLFA_{Descent,no\ lat\ err} \tag{5}$$

It is advantageous to analyze flights without lateral intent errors first, since the errors that exist in these cases would also exist in cases with lateral intent errors (but the reverse is not always true). If the accuracy for flights without lateral intent errors ($NLFA_{No\ lateral\ error}$) is unacceptable, then a test bench analysis and iterative resolution approach should be used to improve the TP's capability in predicting non-level flight segments without lateral intent errors. Once the cases without lateral intent errors have reached a target improvement value, the full operational data test of ERAM needs to be run again to see if the new TP capabilities meet the full requirement ($NLFA_{Total}$). If not, and assuming that the results for flights without lateral intent errors has met its target value,[††††] then representative cases with lateral intent errors may be selected, analyzed and iteration performed to improve the TP's capabilities in dealing with these errors. After the iteration is completed, the full operational data test with all the flights may then be run to validate that the iterations have indeed successfully met the full non-level flight accuracy requirement. During this full operational data test, the level flight requirement (FDP9389) must also be rechecked to ensure that any changes made to meet the non-level requirement have not inadvertently led to a new failure to meet the level flight requirement. Further iteration may need to be performed until both requirements have been met.

The processing required to run the indirect requirement validation tests (i.e., time-shifting to create conflicts) and the validation of the indirect requirements is performed after the two direct requirements are met. The expectation is that any TP capability modifications made to meet the direct requirements will either enable the indirect requirements to be met without modification or will reduce the degree to which the indirect requirements are not met.

VI. Demonstration of New ERAM TP Validation Techniques

To demonstrate the potential benefits of applying the new methodology to the ERAM validation, two examples are presented. The first example demonstrates the benefits of validating the requirements in a specific order, namely, validating the level flight vertical prediction accuracy requirement FDP9389 first. It also demonstrates the benefits of white box testing based on phase of flight. The second example demonstrates the benefit of using white box testing techniques to separate the lateral intent errors associated with aircraft non-adherence to their flight plan routes. The data used for both examples are the legacy operational data sets from the original ERAM FAT in 2007.

A. White Box Testing of Level Flight Altitude Prediction Accuracy (FDP9389) Before Other Requirements

The basic element of the ERAM TP Level Flight Accuracy (LFA) metric is an actual trajectory segment window where the actual trajectory is level (see Ref. 4). Each flight in the operational data typically has many level and non-level windows along its actual trajectory. For each level window, if the maximum error between the actual and predicted trajectories exceeds 500 ft, then the entire window is considered to have unacceptable error. The LFA metric is defined as the ratio of unacceptable windows to total windows for all flights. For the test results in 2007, evaluation of FDP9389 indicated that about 0.9 percent of the level flight segments have unacceptable error, while the requirement was to be less than approximately 0.2 percent (0.0088 ERAM result and 0.0016 requirement from Table 2). Thus, about 5 times as many level flight segments were in error, indicating one or more systematic errors causing altitude prediction inaccuracy.

Post-FAT analyses performed by the FAA determined that errors in ERAM's handling of procedural level-off constraints during descents were a significant factor contributing to ERAM's unmet requirements during the FAT. This result could have been identified quicker and resolved earlier as part of the FAT through the use of white box testing. By breaking the LFA metric into three components, one for windows in each of the climb, cruise and descent phases of flight, the test team should have been able to identify that a significant number of windows with unacceptable error were in the descent phase of flight. For illustrative purposes, a flight sample was selected from the original archived data set that exhibited significant trajectory prediction errors during level segments in the descent phase of flight. The sample flight is a Boeing 737-700 that departed from Owen Roberts International Airport, Cayman Islands with a destination of Newark Liberty International Airport in New Jersey. The traffic sample begins within Washington Air Route Traffic Control Center (ARTCC) in level cruise at an altitude of 39,000 feet. Figure 5 is a vertical view that plots the aircraft's reported transponder altitude (right-most trajectory above 21,000 ft, in red) versus its ERAM predicted trajectories, recalculated at various times along its cruise portion of flight. The sample flight travels in a north easterly path until it transitions control (well after top of descent) to the adjacent New York ARTCC and arrives at its final destination, Newark Airport. The predicted trajectories begin their descent quite early with respect to the aircraft's actual top of descent (TOD). This is the main reason for the large vertical errors during the level (and non-level) window segments in the descent. The cause of these early TOD predictions can be seen from the figure to result from the improperly modeled restriction at 21,000 ft. This improper restriction is not specific to just this example flight, but negatively impacted the descent predictions of many similar arrivals in Washington ARTCC airspace. The use of the phase of flight white box testing technique would have enabled the testing team to identify that there was a significant issue with aircraft in their descent phase of flight, leading to the identification and a resolution of this source of intent error.

[††††] If the target value for flights without lateral intent errors has not been completely met, then iteration on a new representative set of these cases should be performed until an acceptable result is achieved.

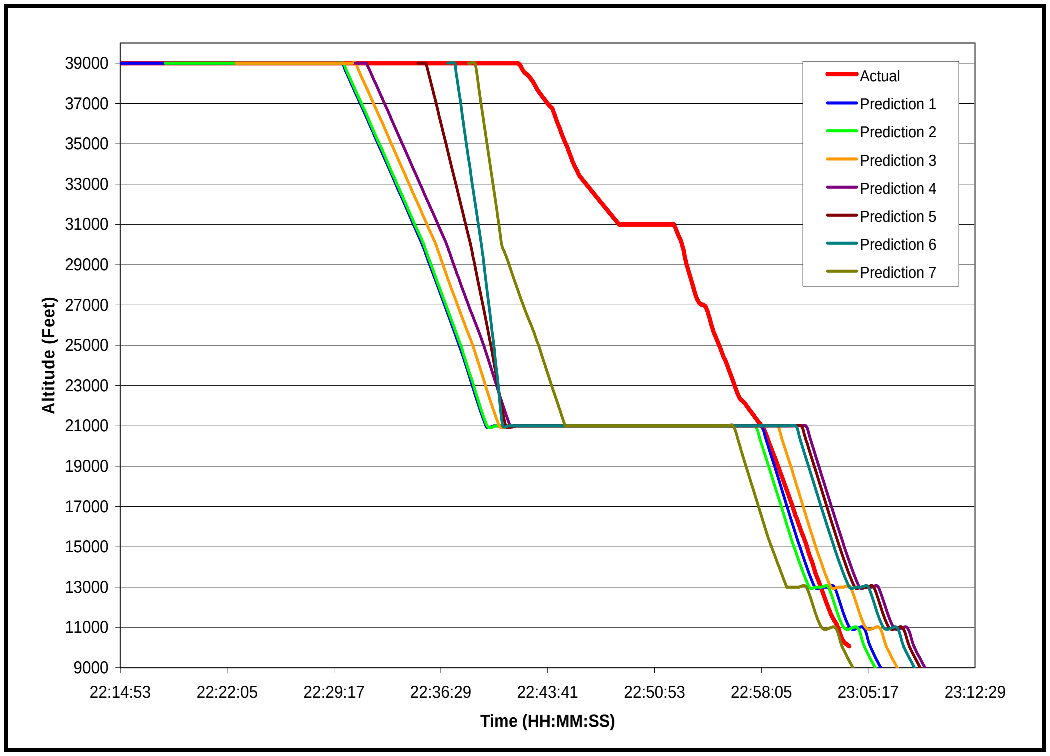

Figure 5: Vertical View - Altitude Versus Time

It should also be noted that this intent error, which is relatively easy to detect when analyzing level flight window segments, is also a source of error for some of the non-level flight windows in descent. From Figure 5, the majority of the aircraft's actual trajectory where it is descending and a predicted trajectory is level would result in an unacceptable non-level window when calculating the non-level flight accuracy (NLFA) metric used for requirement FDP9390. If the incorrectly modeled altitude restriction were identified and removed in the process of achieving the level flight requirement FDP9389, it is reasonable to expect a noticeable improvement in achieving the non-level flight requirement FDP9390 before any analysis for this requirement has even begun. This is one of the expected benefits of performing the requirement validation in stages.

Another of the expected benefits of validating direct requirements before indirect requirements is that modifications to meet a direct requirement are likely to improve the system's ability to meet the indirect requirements. This beneficial impact can be illustrated for the ERAM validation effort by analyzing the 36 flights that generated unacceptable level flight errors, contributing to ERAM's failure to meet the direct requirement FDP9389 (one being the sample flight above). For these 36 flights, the number of level flight segments per flight with unacceptable errors ranged from one to 42 with an average of about 13. Twenty of these 36 flights also generated 36 conflict predictions that were identified as false alerts, contributing to ERAM's failure to meet the indirect false alarm rate requirement ERD1879-C4.

Figure 6 illustrates a histogram of the reduction in the number of unacceptable level flight segments achieved after corrective actions were applied (post the FAT) to meet all the requirements in Table 2. Though one of the 36 flights actually had more unacceptable level flight segments (the bar with a negative value in Figure 6), the corrective actions significantly improved the remaining 35 flights by removing most of their unacceptable level flight segments. Of the 36 corrected flights, only 12 contributed to 22 false alert conflict predictions, compared to 20 contributing 36 false alert predictions before the corrections, a reduction in both the number of flights with, and the total number of, false alerts. Therefore, iterating to meet the direct level requirement FDP9389 likely would have resulted in an improvement in ERAM's false alert rate before attempting to achieve requirement ERD1879-C4.

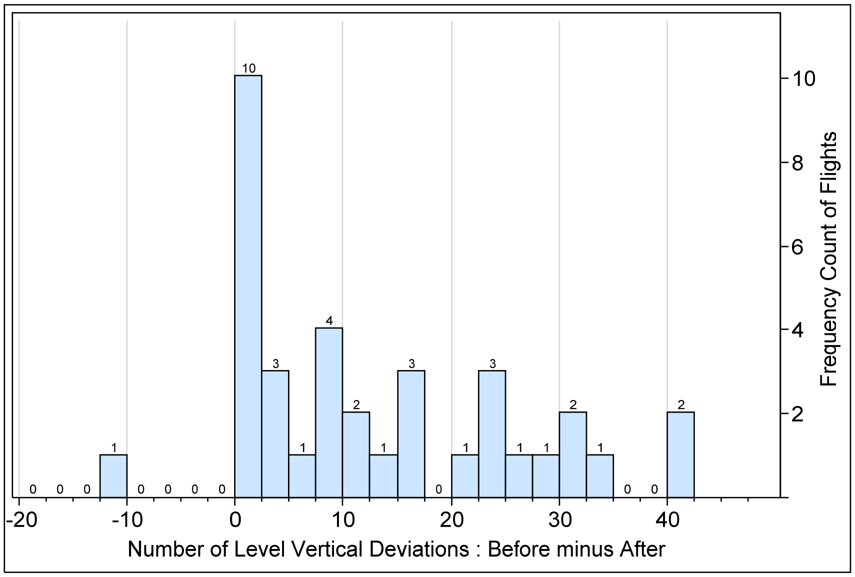

Figure 6: Reduction of Vertical Deviations from Corrective Action

B. White Box Testing to Isolate Lateral Intent Errors

To illustrate another example of beneficial white box testing, the ERAM validation data was analyzed to determine the benefit of isolating cases with lateral routing errors from those without these errors. In theory, identifying and fixing prediction errors for flights without lateral routing errors should also improve flights with lateral routing errors, while providing a simpler environment to identify and resolve common error sources (since lateral routing errors can obscure the impacts of other error sources). To separate out the flights with lateral routing errors, the lateral adherence of a flight to its flight plan route was used. An effective method for defining lateral adherence was defined in Ref. 8. The lateral-adherence algorithm used for this analysis calculates the lateral adherence state for each aircraft's surveillance position report. If the position is within 1.0 nautical mile and heading within 30 degrees of intercepting the known route of flight, the aircraft is considered in lateral adherence, otherwise out. [‡‡‡‡]

First, the lateral-adherence algorithm was evaluated for its effectiveness in isolating lateral routing errors. For the 2007 ERAM validation data set, the trajectory prediction accuracy was measured for each flight in the sample. Three metrics were analyzed including the mean horizontal error (straight line, time coincident distance between the predicted and actual positions), the mean unsigned cross-track error (lateral side-to-side, spatially coincident distance between the predicted and actual position), and mean unsigned along-track error (longitudinal along-path distance between predicted and actual position).[§§§§] These metrics were correlated and partitioned by in- and out-of-adherence state. The results are presented in the Scatter Plot Matrices illustrated Figure 7 and Figure 8, respectively.[10]

For Figure 7 and Figure 8, the three trajectory metrics are correlated by producing a three by three matrix with a total of nine cells. The frequency histograms are provided on the diagonal cells for mean horizontal, cross-track, and along-track errors, respectively. The histogram's y-axis represents the frequency count of mean errors (x-axis), scaled equally for each error metric. Each non-diagonal cell plots each flight's metric as a function of the other metric (e.g. mean horizontal error versus cross-track error). Figure 7 shows that for the in-adherence state, a flight's mean horizontal error is highly correlated to the mean along-track error, while this is not the case for mean cross-track error. The correlation metric[*****] for horizontal to cross-track prediction errors is only 0.17, while the horizontal to along-track error is near one at 0.99. The linear relationship between the along- and cross-track errors

[‡‡‡‡] This algorithm uses tighter bounds than used by ERAM for lateral adherence. ERAM uses conformance boxes centered at the predicted trajectory extending 2.5 nautical miles side-to-side.

[§§§§] These metrics are defined in detail in Ref. 9.

[*****] This term is defined in numerous texts as the *Pearson R* linear correlation coefficient. It measures how well pair wise data fits a straight line (see Refs. 11 and 12). If near +1.0 or -1.0, data fits a line well and near zero does not.

for the in-adherence state is very weak with a correlation coefficient of 0.03. Thus, for measurements within the lateral adherence, trajectory errors manifest themselves mainly in the along-track dimension, indicating an error in predicted ground speed. Figure 8 presents the scatter plot results for the out of adherence state. For out of adherence data, the horizontal error is highly correlated to both cross- and along-track errors, with correlation values of 0.72 and 0.95, respectively. The correlation between the metrics of mean cross- and along-track errors is 0.5, indicating they have a modest linear relationship. Therefore, Figure 7 and Figure 8 show that filtering on lateral adherence state can effectively filter out the lateral routing error. Thus, by focusing on the in-adherence flights first, errors in ground speed prediction manifested as along-track errors, many of which are common to both in- and out-of-adherence cases, can be investigated without the complicating impacts of lateral routing errors.

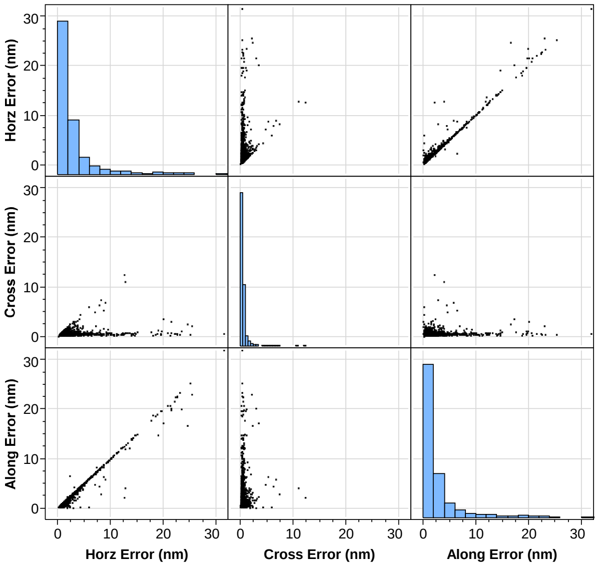

Figure 7: Scatter Plot Matrix of Correlations – Samples “In” Adherence State

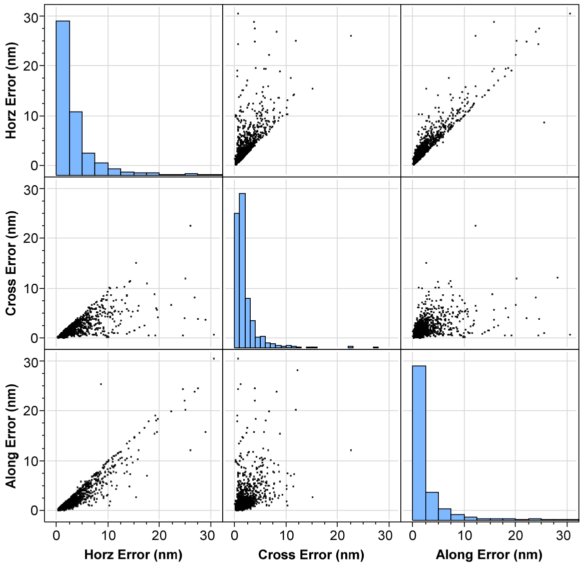

Figure 8: Scatter Plot Matrix of Correlations – Samples “Out” of Adherence State

Now that it is clear that the lateral adherence algorithm would have isolated cases with primarily along-track errors, the question is whether this set of cases was worth analyzing in isolation. Figure 9 is a histogram of the reduction in the flight’s mean along-track error after the post-FAT corrective actions took place. For illustrative purposes, it includes only the top ten percent of the largest along-track errors of flights that were in lateral adherence. This amounted to 158 flights from the complete set of approximately 2200 flights from the original test.

Of these 158 flights, eight had larger errors after corrective action (bars with negative values in Figure 9) and the remaining 150 flights were indeed improved. One of the eight flights with larger errors indicated a difference of 26 nautical miles, while the others are degraded less than five nautical miles. The 150 improved flights exhibited a reduction in mean along-track error ranging from slightly above zero to 32 nautical miles. Since the vast majority of these flights were eventually improved to pass the requirements, the results illustrate, albeit only retrospectively, that segregating out the in-adherence flights that exhibited large trajectory errors provides a good initial focus group to expend resources on corrective actions. Though not analyzed due to lack of data, it is expected that the TP modifications to achieve the benefits in Figure 9 would have improved the TP performance for out-of-adherence flights as well.

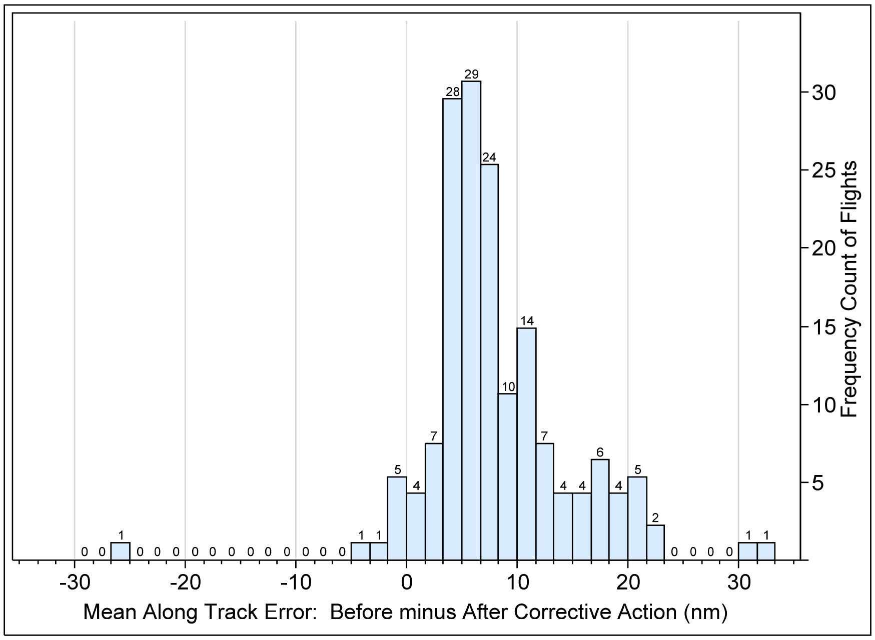

Figure 9: Histogram on Impact of Corrective Actions on Along-Track Error

To further investigate the impacts of separating out the in-adherence cases for analysis, it was desired to determine whether these flights could be shown to have had a significant effect on failing the false alert requirement, ERD1879-C4. Like the previous example for level altitude errors, more than half of the 158 laterally adhering flights with significant along-track errors generated false alerts. These conflict prediction errors occurred both before and after the corrective actions were implemented. The hypothesis being tested assumes that the number of selected flights with and without false alerts does not change between the FAT ERAM run and the corrective action run. If this hypothesis can be rejected and the number of flights with false alert events is indeed lower for the corrective action run, it indicates a significant reduction in the false alarms after the corrective actions were taken. To effectively test the hypothesis, a formal categorical statistical method, called a contingency table, is employed and a chi-squared statistical method applied.[13,14] The results are presented in Table 3. From the formal test, of the 158 laterally adhering flights, 84 flights exhibited at least one false alert event and 74 didn't exhibit any false alerts. After the corrective actions, only 58 flights exhibited false alert events and 100 didn't exhibit any.

Table 3: Contingency Table for False Alerts of Sample Flights

Flights After Corrective Action	Flights from FAT		
	With False Alert	Without False Alert	Total
With False Alert	47	11	58
Without False Alert	37	63	100
Total	84	74	158
	χ^2=14.083, *df*=1; p-value=0.000		

Formally, the test determines if the number of flights with and without false alerts between the two runs can be considered statistically equivalent as expressed in (6) by calculating the squared difference between the two cells from Table 3 where the two runs disagree, divided by their sum.[†††††] The test statistic, χ^2, is defined as follows:

$$\chi^2 = \frac{(n_{21}-n_{12})^2}{(n_{21}+n_{12})} \tag{6}$$

where,

n_{21} is the quantity of flights in the second row, first column of the table
n_{12} is the quantity of flights in the first row, second column of the table

If the hypothesis is true, this ratio will follow a chi-squared distribution with one degree of freedom (see Ref. 13 and 14 for further details). The application of (6) produces a very significant effect in the form of a very small p-value of 0.000.[‡‡‡‡‡] As a result, the hypothesis that the number of flights with false alert events is equivalent before and after the corrective actions can be rejected at a significance level of at least 0.05. Therefore, it can be stated that there is a statistical correlation between the corrective actions on the in-adherence flights and a reduction in ERAM false alarm rate. This illustrates that separately analyzing the performance of these flights during validation testing of requirement ERD1879-C4 could have effectively supported the identification of errors and necessary ERAM capability iterations to ultimately meet this requirement.

VII. Concluding Remarks

A new methodology has been proposed to support the validation of aircraft trajectory predictors for ATC/ATM applications. This methodology includes a collection of techniques and a multi-stage procedural framework for TP validation designed to reduce the effort in identifying and resolving validation failures, avoiding the potentially large costs associated with failures during a single-stage, pass/fail approach. As a case study, the FAA's validation of the ERAM TP, which initially failed to achieve six of its eight TP requirements, was analyzed and specific techniques from the methodology that could have improved the ERAM validation were applied. Two examples evaluated the ERAM TP direct and indirect performance requirements in stages, using white box techniques to isolate error sources. Using actual data from the ERAM validation effort, the results illustrate that the application of additional techniques from the new validation methodology could have, at minimum, identified the problems in ERAM sooner, potentially reducing iteration costs and quite possibly improving the validation outcome of the formal test.

Though the development of the methodology was focused on supporting TP validation, the same techniques and processes could be useful in the research and development of a new trajectory predictor. To control validation costs, the methodology was designed to be done in stages, progressively building up confidence in the TP's ability to meet its requirements. Due to the significant risk of failing to meet a requirement at any stage, the methodology was designed to be iterative, supporting the identification of why a requirement was not met (through white box testing) and low effort retesting (through test bench testing) during iterative TP development to meet the requirement. Since many TP requirements are defined in terms of their client automation's functionality (indirect requirements), these techniques are designed to support validating such requirements. These techniques, originally designed to resolve validation issues, are equally valuable to TP developers during the development of a new trajectory predictor, especially during the research and development phase. In the early development of a TP, since TP performance requirements are rarely available, issues with TP performance are typically first identified as issues in the performance of its client automation's functionality (e.g., conflict probe). Identifying how to modify the TP to enable proper performance of a client application function is equivalent to iterating on the TP capabilities to meet a failed indirect performance requirement. The methodology's iteration and white box testing techniques should be directly applicable to identifying and resolving such TP modeling issues.

[†††††] In Ref. 14, the test is referred to as the *McNemar's* test and is specifically designed for testing two data sets that are not independent. This is clearly the case in this study where the same flights are examined between two runs of ERAM.

[‡‡‡‡‡] Ref. 11 defines the p-value as the smallest level of significance at which the null hypothesis would be rejected. If the p-value is small and less than the required value, often set at 0.05 in common practice, the null hypothesis should be rejected.

VIII. Acknowledgements

The authors would like to express their appreciation to the FAA/Eurocontrol Action Plan 16 (AP16) Committee for all of its work in the development of Common Trajectory Prediction capabilities. This group's accomplishments in finding common ground across TPs supporting a disparate set of ATM and airborne automation applications in the US, Europe and Australia have directly supported the development of the approaches described in this paper.

IX. References

[1] Joint Planning and Development Office, "Concept of Operations for the Next Generation Air Transportation System," Version 2.0, 2007.

[2] SESAR Consortium, "The ATM Target of Operations," Technical Report No. DLT0612-001-02-00, Toulouse, France, 2007.

[3] Mondoloni, S., Paglione, M., et. al., "A Structured Approach for Validation and Verification of Aircraft Trajectory Predictors", *23rd Digital Avionics System Conference*, Salt Lake City, Utah, 2004.

[4] Ryan, H., Chandler, G., et. al., "Evaluation of En Route Automation's Trajectory Generation and Strategic Alert Processing: Analysis of ERAM Performance", FAA Technical Note, DOT/FAA/TC-TN08/10, November, 2008.

[5] Rentas, T., Green, S., Cate, K., "Characterization Method for Determination of Trajectory Prediction Requirements", *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, Hilton Head, South Carolina, 2009.

[6] Eurocontrol/FAA Action Plan 16, "Common TP Structure and Terminology in support of SESAR & NextGen", Version 1.0, January 29, 2010.

[7] Paglione, M., R. D. Oaks, K. D. Bilimoria, "Methodology for Generating Conflict Scenarios by Time Shifting Recorded Traffic Data," *Proceedings of the American Institute of Aeronautics and Astronautics (AIAA) Technology, Integration, and Operations (ATIO) Technical Forum,* November 2003.

[8] Paglione, M., I. Bayraktutar, G. McDonald, J. Bronsvoort , "Lateral Intent Error's Impact on Aircraft Prediction," *Air Traffic Control Quarterly*, Vol. 18 (1), 29-62, 2010.

[9] Paglione, M., R. D. Oaks, "Implementation and Metrics for a Trajectory Prediction Validation Methodology," *Proceedings of the American Institute of Aeronautics and Astronautics (AIAA) Guidance, Navigation, and Control Conference*, August 20-23, 2007.

[10] SAS Institute, 2007, JMP® *Software: ANOVA and Regression Course Notes*, SAS Institute Inc. Cary, NC.

[11] Devore, Jay L. 2000, *Probability and Statistics for Engineering and the Sciences*, 5th Edition, ISBN 0-534-37281-3, Duxbury Press, Belmont, CA.

[12] Box, George E. P., Hunter, J. Stuart, Hunter, William G., *Statistics for Experimenters, Design, Innovation, and Discovery*, Second Edition, Hoboken, NJ, John Wiley & Sons, 2005.

[13] Kachigan, *Statistical Analysis, An Interdisciplinary Introduction to Univariate and Multivariate Methods*, Radius Press, 1986.

[14] Agresti, A., *Categorical Data Analysis*, 2nd ed., John Wiley and Sons, 2002.