Definition and Demonstration of a Methodology for Validating Aircraft Trajectory Predictors

R. A. Vivona, M. M. Paglione, K. T. Cate, G. Enea

Highlights from the Federal Aviation Administration's Support of the National Airspace System

AIAA Guidance, Navigation, and Control Conference

Toronto, Canada August 4, 2010

The World's Forum for Aerospace Leadership

Outline

- Motivation
- TP Validation Overview
- TP Validation Methodology
  - Core Techniques
  - Procedural Framework
- Case Study: Application to ERAM validation
  - FAA Validation Results
  - Example benefits of applying new techniques
- Concluding Remarks

Motivation

- Prediction of 4D aircraft trajectories is cornerstone of future international ATC/ATM concepts
  - U.S. NextGen: Trajectory Based Operations
  - European SESAR: Business Trajectory
- Decision Support Tools (DSTs) using an aircraft trajectory predictor (TP) provide these trajectories

  => DST Validation required. Why independent TP validation?
- Past: Independent TP Validation not performed
  - TP indirectly validated by validating TP-dependent DST functionality
- Benefits of independent TP validation methodology
  - Supports development of Common TP capabilities
  - Improves DST validation by validating major component

TP Validation Overview

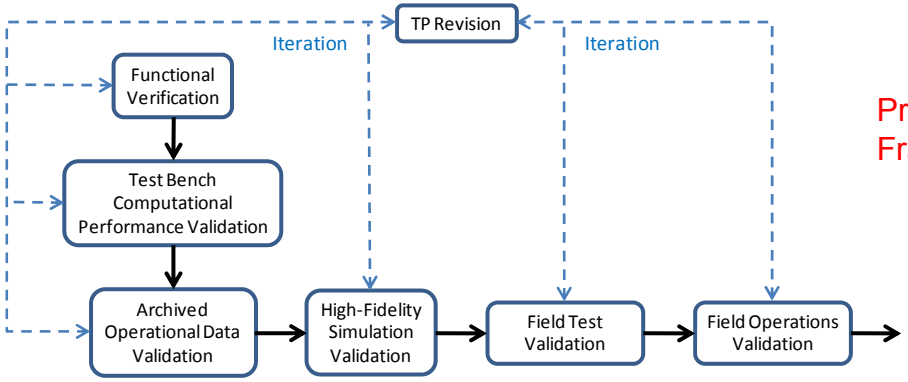
- Focuses on two exercises
  - Functional Verification
    - Confirms TP performs required functions under required conditions (*functional requirements*)
  - Performance Validation
    - Confirms TP performance is within acceptable limits (*performance requirements*)
    - Primary performance requirements:
      - Computational speed
      - Prediction accuracy (most challenging)

- Two classes of TP requirements
  - Direct requirements
    - Define explicit requirements for TP functionality or performance
    - Example: Maximum predicted altitude error or response time
  - Indirect requirements
    - Define required overall system performance for TP-dependent functions
    - Example: Conflict probe false alarm/missed alert rate
  - Not uncommon to have a mixture of both classes
    - Very common today to have more indirect requirements
  - Validation for the different classes requires different approaches

TP Validation Methodology

- Goal: Avoid costs from failures during single-stage, pass/fail validation
- Recognizes:
  - Iteration likely required to achieve validation success
  - Collection of validation data (simulation, field test, operational) can be expensive
    - Iteration can incur costs multiple times
- Core techniques
  - Main approaches used within and to define various stages of validation process
- Procedural framework
  - Overall process as applied to validation of TP
  - Not required steps, but validation options and guidance to create a complete validation effort

Core Techniques
- Use validation stages
- White box testing
- Test bench testing



Procedural Framework

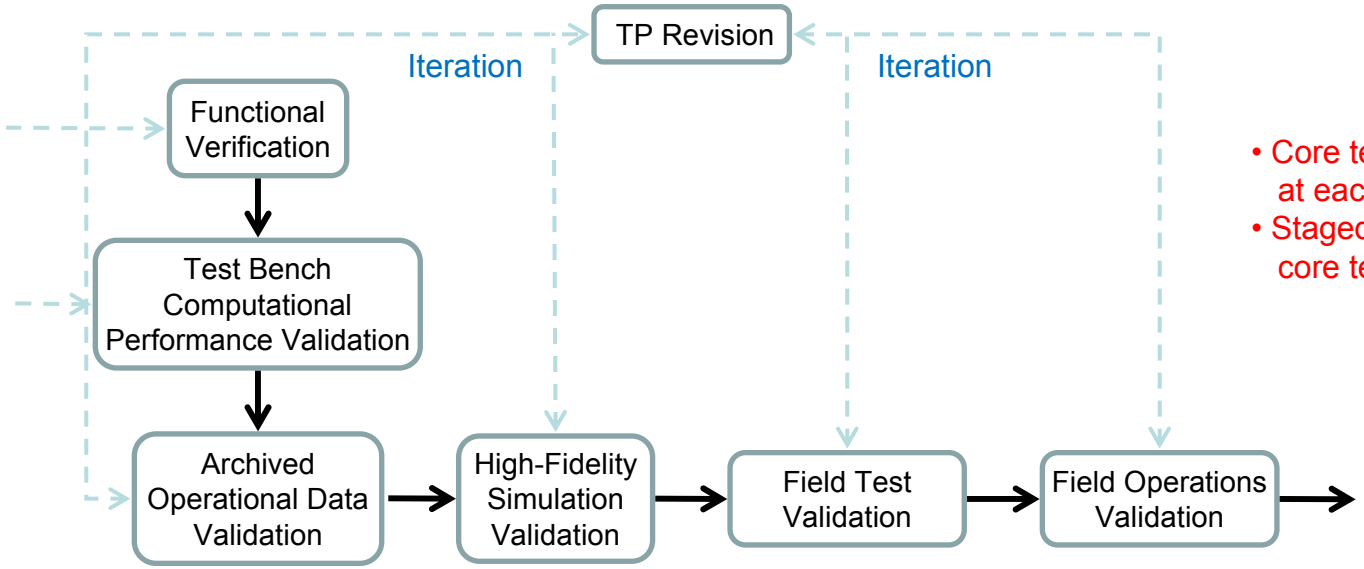
TP Validation Methodology:
Core Techniques

- Use of validation stages
  - Perform validation as a series of (potentially iterative) validation efforts
    - Minimize costs if one or more requirements not met
    - Reduces overall effort in meeting all requirements
  - Applications:
    - Framework: Start with simpler, cheaper efforts that are necessary, but not sufficient
      - Iterate at earlier stages before most significant costs incurred
    - Validate requirements in sequence, not all at once
      - Start with requirements whose success improves chances of achieving other requirements
      - Example: direct requirements before indirect requirements

- White box testing:
  - Use knowledge of internal TP processing to isolate error sources during testing
    - Identify potential critical error sources when performing validation tests
    - If test fails, test results support identification of impact of the various error sources
      - Improves identification of required TP modifications during iteration
  - May only require collecting some additional data during planned tests
    - Identifiable from input data
    - No extra test cases added
  - Example: identifying which cases have or don't have lateral routing errors

TP Validation Methodology:
Procedural Framework

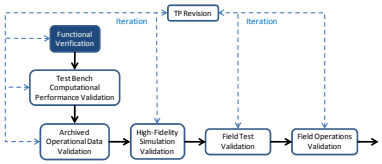
- Provides a series of validation stages
  - Increasing fidelity and, typically, cost
- Represents options, not requirements, for overall validation effort
  - Not all stages need to be performed
  - Some may not be feasible
- Iteration on TP capabilities at any stage
  - If fail or close to failing to meet requirements



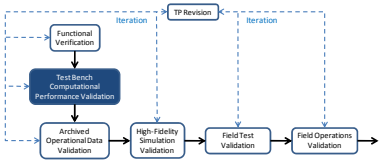
- Core techniques added at each stage
- Staged approach is a core technique

TP Validation Methodology:
Procedural Framework

- Functional Verification
  - Validation of functional requirements
    - Confirm TP functional response to specified inputs
  - Required if functional requirements exist
  - Performed prior to any performance validation tests

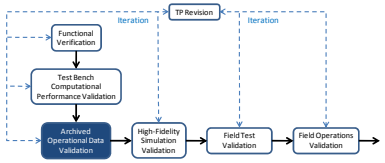


- Test Bench Computational Performance Validation
  - Validation of <u>direct</u> computational speed performance requirements
    - TP response time under specified conditions
  - Tests TP outside of client system (test bench)
  - Pro:
    - May be less expensive to run than full system tests
    - Requires just representative TP input data
    - Can be used for extreme condition testing
  - Con:
    - Necessary, but not sufficient for validation
      - TP performance typically impacted by system (e.g., shared processors)

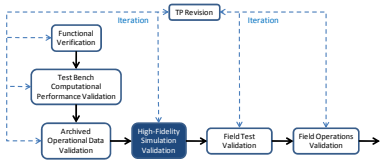


TP Validation Methodology:
Procedural Framework

- Archived Operational Data Validation
  - Validation of all performance requirements
  - Uses pre-recorded operational data (e.g., ETMS)
    - Input data for TP's client system
    - Actual aircraft trajectory data (for prediction accuracy requirements)
  - Pro:
    - Inexpensive way to validate feasible performance requirements
  - Con:
    - Impact of controller actions may limit feasibility
      - Example: Not useful for conflict false alert rate requirement
    - Not feasible if TP client system changes current operations

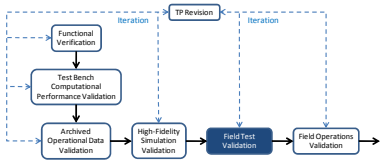


- High-Fidelity Simulation Validation
  - Validation of all performance requirements
  - Uses simulation of aircraft and/or operational environment
    - Too low fidelity impacts feasibility of validation tests
  - Pro:
    - Should be a feasible for most validation efforts
    - Enables statistically significant numbers of test cases to be run
  - Con:
    - Can be expensive
    - Knowledge and modeling of error sources required

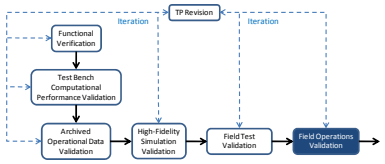


TP Validation Methodology:
Procedural Framework

- Field-Test Validation
  - Validation of all performance requirements
  - Near-operational conditions
    - In actual facility
    - Uses test procedures and/or test aircraft
  - Pro:
    - Can create future operational conditions with operational fidelity
    - Can test extreme or safety critical conditions
  - Con:
    - Often expensive
    - Limited number of test scenarios



- Field-Operations Validation
  - Validation of all performance requirements
  - Actual operational conditions
    - In actual facility
    - Uses actual procedures and aircraft
  - Pro:
    - Actual deployment environment, ultimate fidelity
  - Con:
    - Expensive
    - No control of environment, rare cases may not occur during testing
    - Difficult to capture large amounts of data for statistical significance
    - May limit to TP client systems that don't significantly alter current operations



ERAM Case Study:

FAA ERAM Validation Results

- FAA En Route Automation Modernization (ERAM) Factory Acceptance Test (FAT) in September 2007
  - Used pre-recorded data from Host Computer System and primary radars
    - Archived Operational Data and High-Fidelity Simulation (time-shifted archived data) Validation
  - 6 of 8 TP-related requirements not met
    - Two (level and non-level flight) altitude prediction accuracy requirements (direct) – not met
    - Two conflict probe warning time requirements (indirect) – met
    - Four conflict probe false alarm/missed alert rate requirements (indirect) – not met
- FAA initiated effort to correct ERAM deficiencies
  - Iterative modifications and re-evaluation performed
  - November 2008, ERAM passed all requirements
- Lesson Learned
  - Years of costly iteration (starting before FAT) required to ultimately meet requirements
    - Ad Hoc iterative procedures used
  - Approaches need improvement
- New methodology techniques could have helped!

ERAM Case Study:
Examples

- Paper presents several proposed techniques for improving ERAM validation approach
  - Analyses performed to determine, retrospectively, whether techniques would have helped
  - Used actual data from FAA FAT

- Two examples illustrate benefits of:
  - Iterating to achieve the level-flight altitude (LFA) accuracy requirement before validating other requirements
    - Technique: using validation stages
  - Identifying flights without lateral intent errors during false alarm rate testing
    - Technique: white box testing

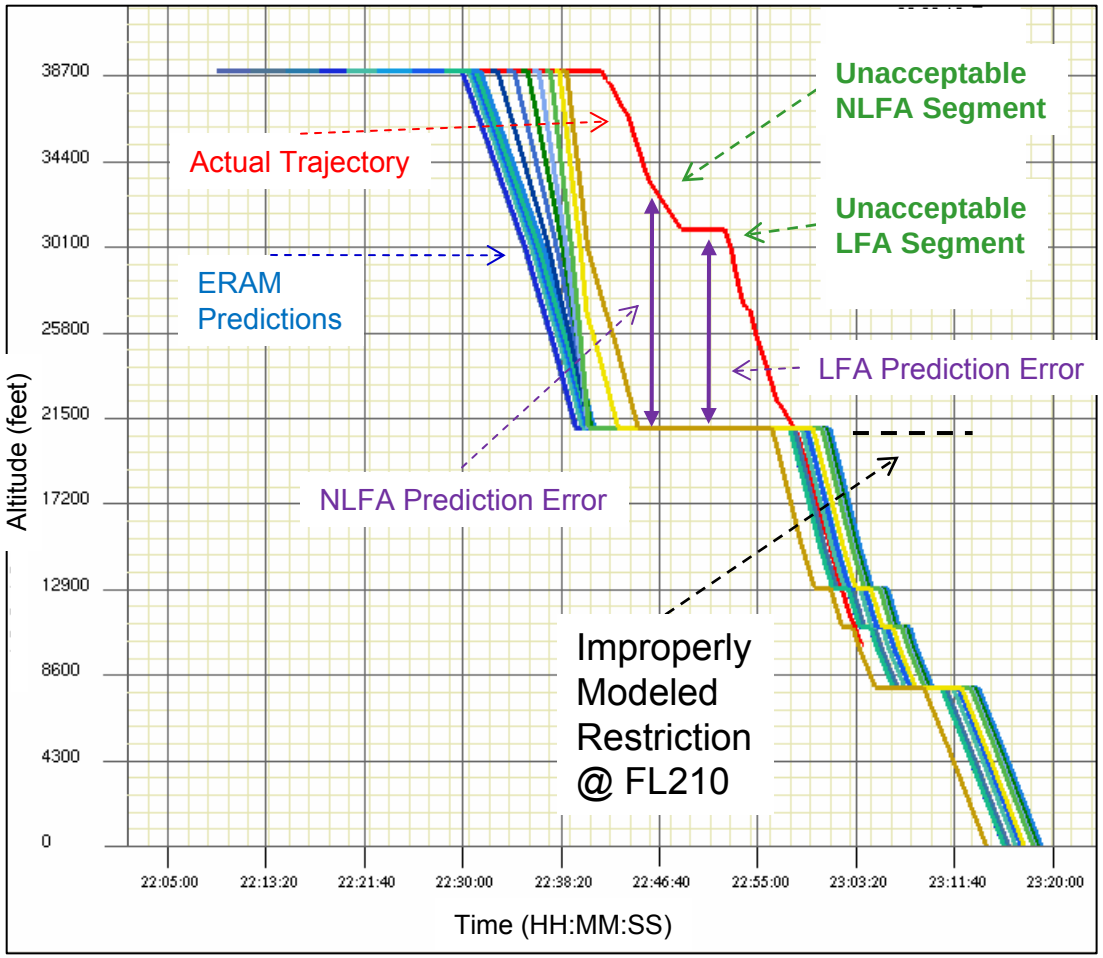
ERAM Case Study:
LFA Accuracy Requirement Example

- Validation Metric
  - If maximum prediction error >500 ft, level segment is unacceptable
  - Level flight accuracy metric: $LFM = \frac{\text{\# unacceptable LF segments}}{\text{Total \# LF segments}}$
- FAT results
  - Requirement: LFM ≤ 0.0016
  - Initial ERAM result: LFM = 0.0088 (unmet)
- Proposed Technique: Iterate to achieve LFA accuracy requirement before validating against other requirements
  - Analyze if would have improved meeting other requirements
    - Non-level flight (direct) accuracy (NLFA) requirement (unmet)
    - False Alarm Conflict probe (indirect) requirement (unmet)

ERAM Case Study:

LFA Accuracy Iteration Impact on NLFA Accuracy

- FAA post-FAT analysis identified:
  - Improperly modeled level-off restrictions
  - Impacted many flights in descent
- Impacts both LFA and NLFA accuracy
- Easier to identify in LFA accuracy analysis
- Iterating to meet LFA accuracy requirement would likely have improved NLFA accuracy performance

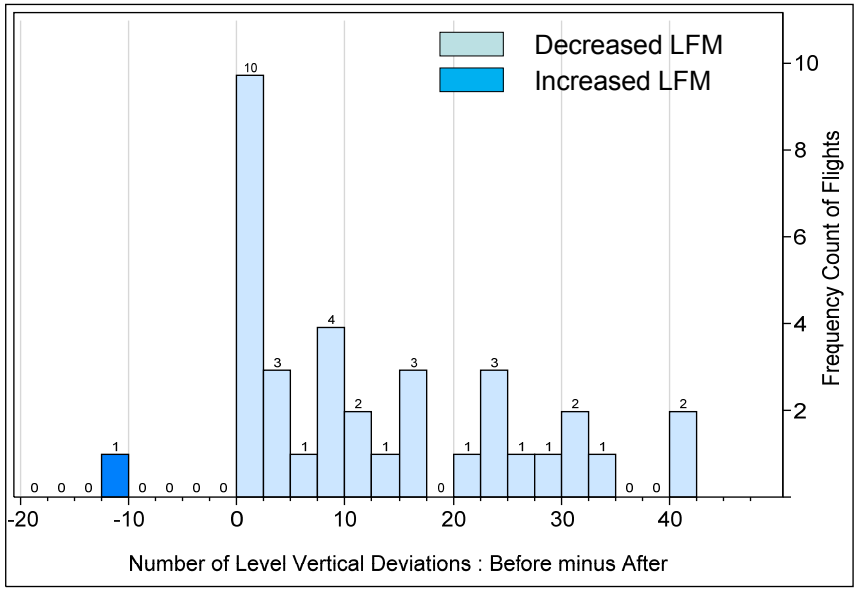


Example Flight: Vertical Profile (B737-700, cruising FL390 in ZDC)

ERAM Case Study:

LFA Accuracy Iteration Impact on False Alarms

- 36 flights had unacceptable LFA segments during FAT
  - Majority needed improvement to achieve post-FAT validation
- FAT results for these 36 flights
  - Flights with false alarms = 20
  - Number of false alarms = 36
- Post-FAT results for these 36 flights
  - Flights with false alarms = 12
  - Number of false alarms = 22
- Iterating to meet the LFA requirement would likely have improved false alarm performance



Post-FAT impact on 36 flights with unacceptable level flight altitude segments during FAT

ERAM Case Study:

Identifying Flights without Lateral Intent Errors Example

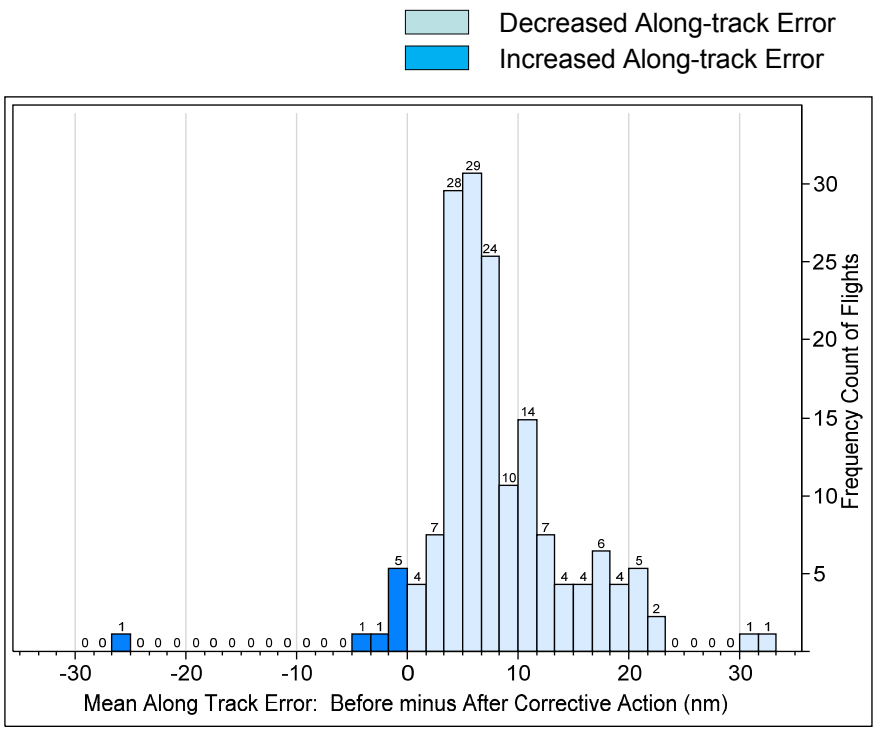
- Proposed Technique: Identify flights with and without lateral intent (routing) errors during false alarm testing
  - Analyze flights <u>without</u> lateral intent errors
    - Sufficient along-track lateral errors to require TP modification post-FAT?
    - Modifying TP for these flights had a positive impact on false alarm performance?
  - If so, approach would have
    - Identified along-track error impact on false alarms
    - Could have eliminated common error sources with lateral route error flights
      - Easier to resolve without lateral routing error impact

- Used lateral route adherence to identify flights without lateral intent errors
  - For each actual state (surveillance report), in adherence if:
    - Position within 1 nmi and heading within 30 degrees of flight plan route
  - Only required analyzing actual state and route data
    - No additional testing required
  - Effectively identified flights with primarily along-track lateral errors

ERAM Case Study:

Identifying Flights without Lateral Intent Errors Example

- Analyzed 158 in-adherence flights
  - Top 10% largest along-track errors
    - Majority of 158 flights were improved post-FAT to achieve validation
    - Along track error reduced
- Flights without lateral intent errors needed ERAM TP modifications to achieve validation
- Analyzed impact of these flights on false alarm performance
  - False alarms during FAT = 84
  - False alarms post-FAT = 58
- Iteration on these flights likely would have improved false alarm performance



Post-FAT reduction in mean along-track error (top 10% – 158 flights)

Concluding Remarks

- New TP validation methodology proposed
  - Core techniques and multi-staged approach
  - Designed to:
    - Reduce effort in identifying and resolving validation failures
    - Avoid costs associated with single-stage, pass/fail approaches
- ERAM case study analyzed
  - Illustrated new techniques could have benefitted ERAM validation by:
    - Ordering requirements to reduce overall effort
    - Identifying error sources during testing to provide TP iteration support
- Validation techniques could also support new TP R&D
  - New TP development
    - Rarely have direct TP requirements
    - Iteration typically required
  - New methodology adds structure to approaching these development issues