AIAA Guidance, Navigation and Control Conference and Exhibit
20 - 23 August 2007, Hilton Head, South Carolina

AIAA 2007-6517

Implementation and Metrics for a Trajectory Prediction Validation Methodology

Mike M. Paglione* and Robert D. Oaks†
William J. Hughes Technical Center, Atlantic City Int'l Airport, NJ

At the heart of every air traffic decision support tool's functionality is its trajectory prediction, where a trajectory is defined as the 4-dimensional path of an aircraft. This paper presents a comprehensive implementation for measuring the accuracy of a trajectory prediction in support of a validation methodology. The process includes four main processing areas: (1) parsing and checking the actual positional data of an aircraft (i.e., the aircraft's actual trajectory), (2) parsing the trajectory predictions, (3) comparing the actual and predicted aircraft trajectory by sampling and measuring, and (4) analyzing the results. This paper presents detailed descriptions of the sampling process and metrics used to measure the accuracy of a predicted trajectory. Several aspects of the analysis and implementation are provided as well, such as inferential statistical approaches and graphical user interfaces to examine individual flights.

Nomenclature

ACID	Aircraft Identification
API	Application Program Interface
ARTCC	Air Route Traffic Control Center
ATC	Air Traffic Control
CID	Computer Identification
COTS	Commercial Off-the-Shelf
CPAT	Conflict Probe Assessment Team
DST	Decision Support Tool
ERAM	En Route Automation Modernization
FAA	Federal Aviation Administration
FL	Flight Level
GUI	Graphical User Interface
IBST	Interval Based Sampling Technique
JDBC	Java Database Connectivity
JFK	John F. Kennedy International Airport
JOGL	Java bindings for OpenGL
JPDO	Joint Planning and Development Office
NAS	National Airspace System
SEGV	Software Engineering, Graphics, and Visualization Research Group at Rowan University
SQL	Structured Query Language
TP	Trajectory Predictor
URET	User Request Evaluation Tool
UTC	Coordinated Universal Time
ZDC	Washington. ARTCC
ZME	Memphis ARTCC

* FAA Project Lead of Conflict Probe Assessment Team, Simulation and Analysis Group, AIAA Member.
† Principal Staff Member, General Dynamics Information Technology, AIAA Member.

I. Introduction

Most air traffic service providers in the United States and Europe anticipate significant growth in air traffic that is expected to outpace the capacity limits of their aviation systems, resulting in greater congestion and inefficiency. Broad advances in ground-based and airborne automation, such as decision support tools (DSTs), are envisioned to mitigate the problem. These tools have many purposes and typically serve to lower the complexity of airspace problems faced by the current human decision makers operating the system. They include tools that serve to predict future conflicts between aircraft, both for ground based controllers or airborne pilots, allowing more strategic separation management of aircraft. Air traffic management DSTs include capabilities that forecast where and when traffic workload would stress the system, allowing air traffic supervisors to make more efficient adjustments to either avoid the condition or alter staff and/or airspace accordingly. Such tools also include air traffic metering tools to efficiently sequence aircraft into en route and arrival flows, maximizing the capacity of the system. A common thread in all these DSTs is the accurate and timely modeling of the aircraft's current state and anticipated future path. This function is referred to as the trajectory predictor (TP) process. The trajectory is the actual or future 4-dimensional path of the aircraft. TP accuracy can be measured by post flight comparisons of predicted and observed aircraft trajectories. Since the predicted trajectory is the fundamental input that sustains the DST's capabilities and functions, the accuracy of the TP has a direct impact on the DST's overall performance and usability.

In order to attain the specified accuracy requirements of a DST, it is necessary to validate the DST's TP. Ref. 1 presents a TP validation methodology that can drive the performance of a TP toward a targeted level. To achieve this, the validation methodology must first measure the trajectory performance of the TP, and then identify the specific sources of any resulting errors, thus facilitating an improvement. Again, the TP operates in context of a DST, therefore improvement of a TP's predictions correlates to an improvement in its companion DST. Ref. 2 defines some metrics used within this methodology and shows how these metrics can assess a TP's impact on a DST. Figure 1 summarizes this graphically.

This paper presents the implementation details in measuring a TP's accuracy, which is a critical element of validating a TP. It provides concise definitions of the metrics calculated, discusses the methods required to begin analysis of a TP, and demonstrates, with examples, the tools and steps required to perform this analysis.

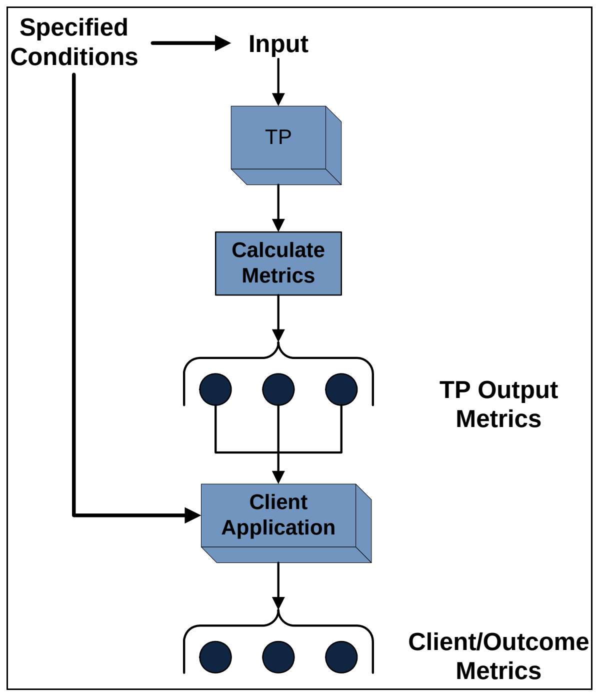

Figure 1. Illustration of Application of TP Metrics to Higher-Level Applications[‡]

II. Measurement of Trajectory Accuracy

Measurement and analysis of a TP's performance focuses on TP accuracy. Accuracy may be defined as the degree of conformity of a measured or calculated quantity to its actual (true) value. For TP measurement, accuracy is the difference between the TP's predicted path of the aircraft to the actual path the aircraft flew. As shown in Fig. 1, the TP is driven by input data and as a result produces aircraft trajectories that drive the DST to produce other client application products (e.g. conflict probe predictions). Ultimately, decision makers will focus measurements on the client's outcomes, where metrics on the TP's predictions are a source.

There are many considerations in measuring the TP's accuracy. First, the true path of the aircraft needs to be determined. This is usually done by examining radar surveillance reports and other supplemental air traffic control (ATC) data, such as ATC clearances in the form of flight plan amendments and vertical altitude clearances. Since the data may have errors, some level of reasonableness checking is needed. Second, the TP's trajectory predictions are captured and parsed. Third, the actual aircraft paths are compared to the predicted trajectories. Since there is

[‡] Adapted from Figure 8 in Ref. 2.

usually a large amount of data, a statistical sample can be taken to estimate the errors. Next, the analysis begins by calculating descriptive and inferential statistics of the errors. Graphical tools can also be used to aid in this analysis phase.

Several implementation schemes are possible. The approach presented in this paper uses Java- based applications that parse the input files, check for reasonableness, sample and calculate the metrics, and upload the data into a set of relational database tables. The relational database acts as a catalyst for the analysis phase. Structured Query Language (SQL) calls are designed to calculate the various statistics and explore the results. Finally, an individual flight's predicted trajectories can be extracted for graphical display against the truth data, providing insight into the operational source and impact of the errors.

III. Interval Based Sampling Technique

The Interval Based Sampling Technique (IBST) is the trajectory accuracy sampling method developed by the Federal Aviation Administration's (FAA) Conflict Probe Assessment Team (CPAT). It has been previously documented in Ref. 3 and has been used in a number of FAA studies and test programs. As early as 1999, the TPs within the operational DSTs of the User Request Evaluation Tool (URET) and the Center TRACON Automation System were evaluated using this technique. More recently, it is being employed to evaluate the TP in the En Route Automation Modernization (ERAM) system, which will replace the en route operational systems such as URET and the Host Computer System.

IBST is a two-step process that pairs the track and trajectory points to measure the prediction errors for an entire flight. This sampling technique takes the perspective of the DST user, the air traffic controller. The active trajectory at the time the controller is looking at the display may be several minutes old and in error. Consequently, in the IBST the trajectories are sampled at the current time for a look-ahead time of 0 seconds and at a number of parameter times in the future (e.g., 300, 900, and 1200 seconds). This is contrasted with a sampling technique that uses the internal build time of the trajectory to start the sampling .[4, 5]

The age of the trajectory, which is internal to the DST, is irrelevant to the controller; only the accuracy of the prediction is important. The controller uses track data to safely separate aircraft and a DST to resolve future aircraft conflicts. The CPAT designed the interval based sampling technique from the perspective of the air traffic controller to answer two fundamental questions: ***How accurately is the DST's trajectory currently predicting the present position of the aircraft?*** and ***How accurately is the DST's trajectory currently predicting the future position of the aircraft?***

The two primary steps in the IBST are:

1) An aircraft is selected for measurement and the track points are sampled in succession a parameter number of seconds (e.g., 120 seconds) until the end of the track is reached. Each track point selected as a sample has a specific time associated with it, which is referred to as the sample time. The aircraft's trajectories are then searched to find the most recent trajectory for the given sample time. This operation is repeated for every track point that is sampled. This first sampling step obtains position prediction error data for a look-ahead time of 0 seconds. This data answers the first of the air traffic controller's questions on accuracy, namely the accuracy of the DST's prediction for the present position of the aircraft. A second sampling operation is necessary to obtain error data for other look-ahead times into the future.
2) Once a track point and its current trajectory are selected for sampling, a second sampling step is executed. The second step samples future points on the trajectory relative to the current sample time. As discussed previously, the first sampling step selects a point on the trajectory that has the same time value as the current track point, corresponding to a look-ahead time of 0 seconds. The second step selects points on the trajectory that are defined as a parameter set of times into the future (e.g., 300, 600, 900, and 1800 seconds). It then finds the future track reports that have the same times as the selected trajectory points. For each look-ahead time, the spatial errors are calculated between the selected trajectory points and their corresponding track points. This second step answers the second of the air traffic controller's questions on accuracy, namely the accuracy of the DST's prediction of the future position of the aircraft.

A graphic depiction of the IBST is shown in Fig. 2. The line labeled **Track** represents the time line for an aircraft track. The time point labeled T_S represents the initial interpolated track point.

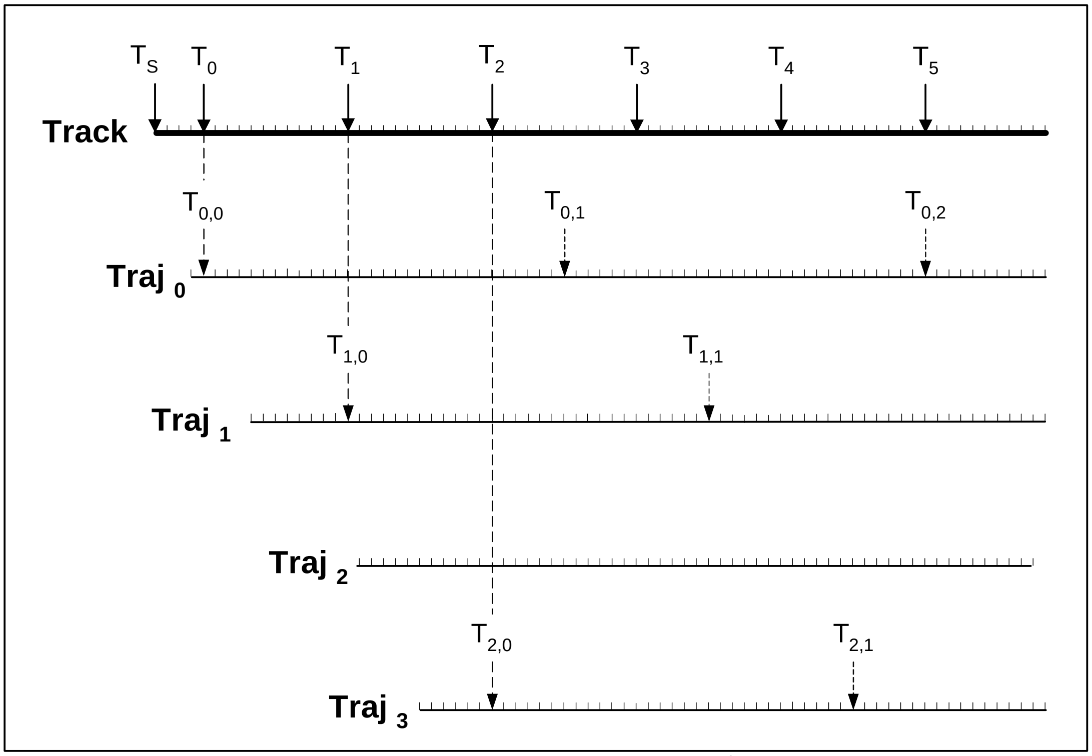

Figure 2. Time-line for the IBST[§]

The sampling time to start computing metrics for this track is represented by T_0, which is defined as

$$T_0 = T_S + \Delta T_S \tag{1}$$

where ΔT_S is a parametric value (a multiple of the interpolation interval) that establishes the starting time at a point where the track is more stable.

The trajectories for this aircraft are presented in Fig. 2 by the time lines labeled **Traj$_0$**, **Traj$_1$**, **Traj$_2$**, and **Traj$_3$**. The trajectory to be sampled for a particular track sampling time is the trajectory with the latest trajectory build time not exceeding the track sampling time. The selected trajectories, which have been interpolated using the same interval used for the track data. In Fig. 2, the trajectory labeled **Traj$_0$** would be sampled for the sampling time T_0. This point is labeled $T_{0,0}$ and represents the look-ahead time of 0 seconds for the trajectory sampling time T_0.

Metrics are computed at the time point labeled T_0 and at the incremented time points $T_{0,1}$ and $T_{0,2}$, defined as

$$T_{i,j+1} = T_{i,j} + \Delta T_i \tag{2}$$

where ΔT_i is the parametric sampling interval for a specific sampling time.

The trajectory sampling process continues until either the end of the track is reached, the end of the trajectory is reached, or the time exceeds $T_0 + \Delta T_{win}$, where ΔT_{win} is a parametric input. Then the next track sampling time T_{i+1} is computed as

$$T_{i+1} = T_i + \Delta T \tag{3}$$

where ΔT , is the parametric sampling interval for sampling a specific track and trajectory.

[§] Adapted from Figure 4 in Ref. 3.

IV. Trajectory Prediction Accuracy Metrics

In this section, a set of basic TP metrics are defined.** An aircraft's track may be approximated by a series of 4-dimensional position points. These four dimensions are:

1) time
2) x-coordinate
3) y-coordinate
4) altitude

The x- and y-coordinates are in a stereographic coordinate system with the x-axis representing the west-to-east direction and the y-axis representing the south-to-north direction. Both the x- and y-coordinates are measured in nautical miles. The altitude represents the aircraft's pressure altitude measured in feet. An aircraft's trajectory is usually considered to be an aircraft's predicted track, also consisting of a series of 4-dimensional position points.

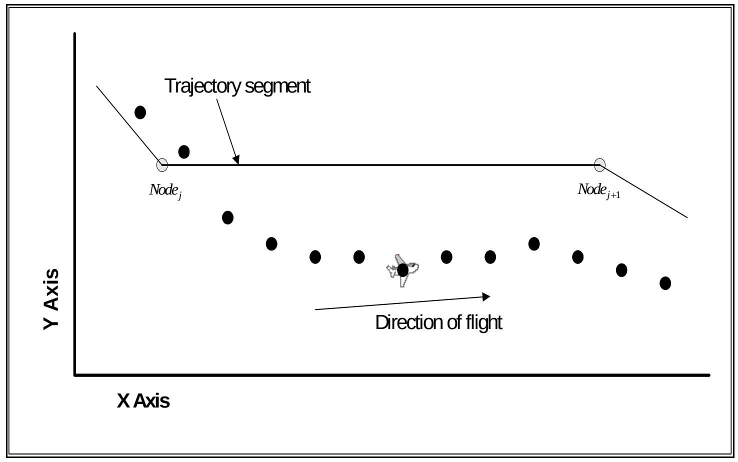

Figure 3. Aircraft Track and Trajectory

An example of a track and a trajectory is illustrated in Fig. 3, which shows an xy-plane containing points that represent the position of an aircraft flying in an easterly direction. Two trajectory nodes, represented as $Node_j$ and $Node_{j+1}$, along with corresponding line segments connecting the trajectory nodes are also shown in Fig. 3. Each of these track position points and trajectory nodes would have an associated time, x- and y-coordinate, and altitude.

The track position point with the overlaid graphic of an airplane represents a specific point of interest, which will be discussed in the following subsections on the various trajectory accuracy metrics.

A. Basic Metrics for Trajectory Prediction

1. Horizontal Metrics

Figure 4 shows the three basic metrics for trajectory prediction accuracy that lie in the horizontal plane: the horizontal error (e_{horiz}), along-track error (e_{along}), and cross-track error (e_{cross}). These metrics are based on the coordinates of the aircraft, which is denoted as AC in Fig. 4, and a trajectory segment containing the points TJ_1 and TJ_2. These coordinates are defined as vectors in Eq. (4), Eq. (5), and Eq. (6).

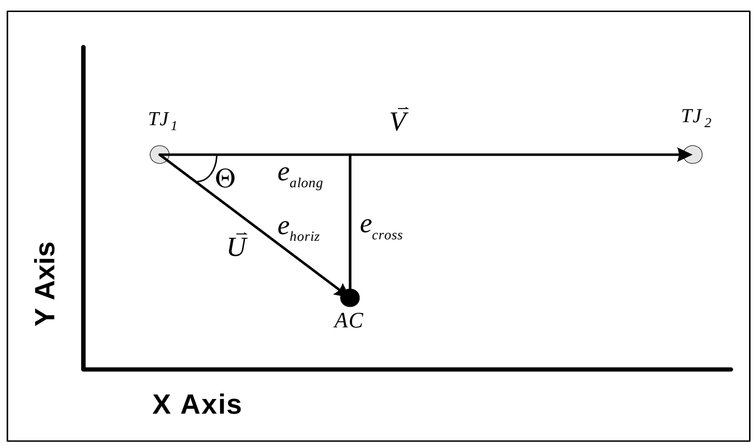

Figure 4. Three Basic Trajectory Prediction Metrics

$$AC = \begin{pmatrix} x_{AC} \\ y_{AC} \end{pmatrix} \tag{4}$$

** These metrics were first presented in Ref. 6 and later in Ref. 7

$$TJ_1 = \begin{pmatrix} x_{TJ_1} \\ y_{TJ_1} \end{pmatrix} \tag{5}$$

$$TJ_2 = \begin{pmatrix} x_{TJ_2} \\ y_{TJ_2} \end{pmatrix} \tag{6}$$

Note also that Fig. 4 shows positive trajectory prediction accuracy errors. Of course, by its definition as a distance between two coordinate points e_{horiz} is always positive. The along-track error e_{along} is positive if the aircraft is ahead of the corresponding route segment node. The cross-track error e_{cross} is positive if the aircraft is to the right of the route segment.

The vectors $\vec{U}$ and $\vec{V}$ shown in Fig. 4 are defined as:

$$\vec{U} = \begin{pmatrix} u_x \\ u_y \end{pmatrix} = \begin{pmatrix} x_{AC} - x_{TJ_1} \\ y_{AC} - y_{TJ_1} \end{pmatrix} \tag{7}$$

$$\vec{V} = \begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} x_{TJ_2} - x_{TJ_1} \\ y_{TJ_2} - y_{TJ_1} \end{pmatrix} \tag{8}$$

First of all, it is obvious that the horizontal error is merely the magnitude of the vector $\vec{U}$, which can be calculated as:

$$e_{horiz} = \left|\vec{U}\right| = \sqrt{u_x^2 + u_y^2} \tag{9}$$

For the along-track error, consider the angle between the vectors $\vec{U}$ and $\vec{V}$, which is shown as Θ in Fig. 4. This angle can be found using a geometric interpretation of the vector dot product, which can be calculated as:

$$\vec{U} \bullet \vec{V} = \left|\vec{U}\right|\left|\vec{V}\right| \cos\Theta \tag{10}$$

$$\cos\Theta = \frac{\vec{U} \bullet \vec{V}}{\left|\vec{U}\right|\left|\vec{V}\right|} \tag{11}$$

The along-track error (e_{along}) is the projection of the vector $\vec{U}$ onto the vector $\vec{V}$, which is defined as:

$$e_{along} = \left|\vec{U}\right| \cos\Theta = \frac{\left|\vec{U}\right|\left(\vec{U} \bullet \vec{V}\right)}{\left|\vec{U}\right|\left|\vec{V}\right|} = \frac{\left(\vec{U} \bullet \vec{V}\right)}{\left|\vec{V}\right|} \tag{12}$$

Numerically this can be calculated as:

$$e_{along} = \frac{\left(\vec{U} \bullet \vec{V}\right)}{\left|\vec{V}\right|} = \frac{u_x v_x + u_y v_y}{\sqrt{v_x^2 + v_y^2}} \tag{13}$$

The cross-track error can be derived using the vector cross product of the vectors $\vec{U}$ and $\vec{V}$, which is defined as:

$$\vec{U} \times \vec{V} = \hat{n}\left|\vec{U}\right|\left|\vec{V}\right| \sin\Theta \qquad (14)$$

where $\hat{n}$ is a unit pseudo-vector perpendicular to both $\vec{U}$ and $\vec{V}$. Since $\hat{n}$ is a unit vector, Eq. (14) may be rewritten as:

$$\left|\vec{U} \times \vec{V}\right| = \left|\vec{U}\right|\left|\vec{V}\right| \sin\Theta \qquad (15)$$

$$\sin\Theta = \frac{\left|\vec{U} \times \vec{V}\right|}{\left|\vec{U}\right|\left|\vec{V}\right|} \qquad (16)$$

The cross-track error (e_{cross}) can then be derived as:

$$e_{cross} = \left|\vec{U}\right| \sin\Theta = \frac{\left|\vec{U}\right|\left(\left|\vec{U} \times \vec{V}\right|\right)}{\left|\vec{U}\right|\left|\vec{V}\right|} = \frac{\left(\left|\vec{U} \times \vec{V}\right|\right)}{\left|\vec{V}\right|} \qquad (17)$$

Using traditional vector operations, this can be calculated as:

$$e_{cross} = \frac{u_x v_y - u_y v_x}{\sqrt{v_x^2 + v_y^2}} \qquad (18)$$

These trajectory prediction accuracy metrics are generic in the sense that they are defined for any given track position point and two trajectory points along a trajectory segment. Sections IV.B and IV.C define how these metrics can be interpreted specifically as time-coincident or spatially-coincident trajectory prediction accuracy errors, depending on how the two trajectory points are selected.

2. Vertical Metrics

The vertical error represents the difference between the track altitude and the predicted altitude. This error, depicted in Fig. 5, lies perpendicular to the horizontal plane. A positive vertical error indicates that at a corresponding point in time the aircraft is above where the trajectory predicted it would be.

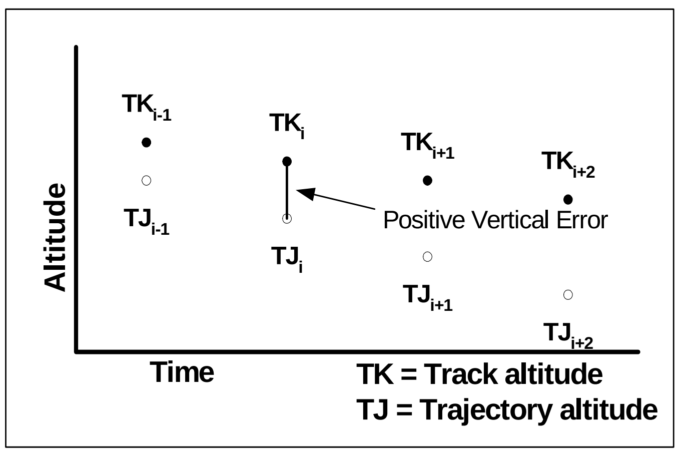

Figure 5. Vertical Trajectory Error[††]

B. Time-coincident Trajectory Prediction Errors

To calculate time-coincident trajectory prediction errors, the point TJ_1 and TJ_2 are selected such that TJ_1 is the point along the trajectory segment with a corresponding time as the point of interest, which is usually found using linear interpolation, and TJ_2 is the following node along the trajectory. Thus, the metrics are formed by projecting the actual track position of the aircraft onto a line segment of the trajectory. Again, this particular line segment is formed by the time-coincident point, TJ_1 and the next time ordered node, TJ_2 along the trajectory.

[††] Adapted from Figure 4 in Ref. 3.

Figure 6 shows an example in which the point TJ_1, a point on the trajectory line segment between $Node_j$ and $Node_{j+1}$, has the same time value as the track point of interest. The point TJ_2 is the end of the trajectory line segment. The situation shown in Fig. 6 is equivalent to the example shown in the Section IV.A to derive the trajectory prediction accuracy metrics.

Figure 7 shows an example in which the point TJ_1, which again has the same time value as the track point of interest, lies on an earlier trajectory line segment. Again, the point TJ_2 is the endpoint of the trajectory line segment. In this example the along-track error is positive, much more so than in the previous example, and due to the geometry of the situation, the cross-track error has switched signs and is now negative.

The horizontal error, e_{horz}, is always unsigned, but the along- and cross-track errors are signed indicating the orientation of the aircraft's actual position relative to the predicted position. In Fig. 6, both along and cross-track are positive values with the actual aircraft ahead and to the right of the time-coincident trajectory prediction.

Figure 8 shows an example in which the point TJ_1, again having the same time value as the track point of interest, lies on a later trajectory line segment. In this case the point TJ_2 is the end point or node of the next trajectory line segment. In this example the along-track error is negative, which reflects the fact that the aircraft is behind where it would be expected to be in the trajectory. Due to the geometry of this example, the cross-track error is now positive.

It can be detected whether or not the projection of the aircraft's position point lies on the trajectory segment by defining the scalar k value as the along-track error divided by the magnitude of the vector $\vec{V}$, as presented in Eq. (19).

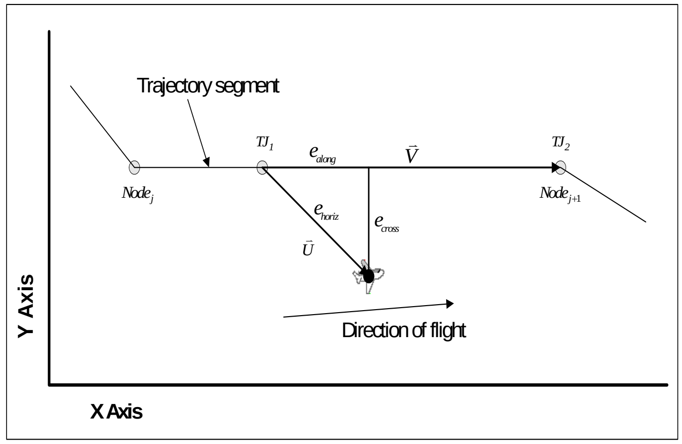

Figure 6. Example of Time-coincident Trajectory Prediction Errors (1 of 3)

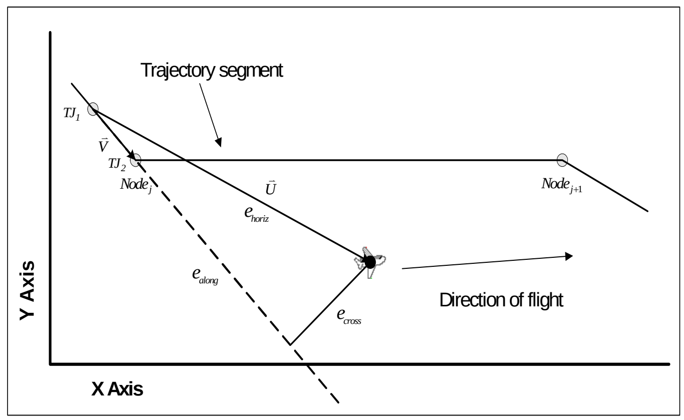

Figure 7. Example of Time-coincident Trajectory Prediction Errors (2 of 3)

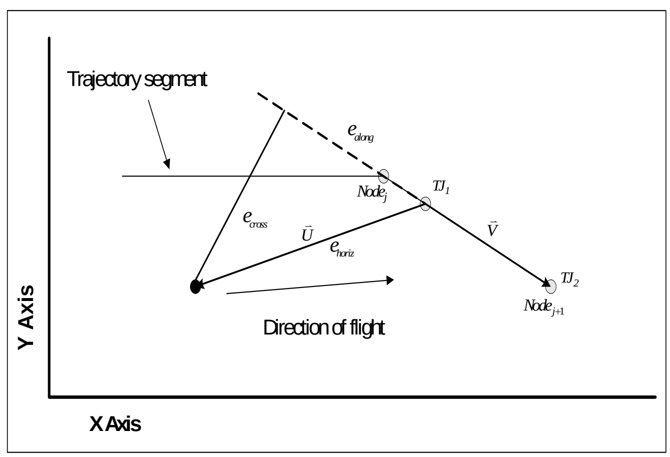

Figure 8. Example of Time-coincident Trajectory Prediction Errors (3 of 3)

$$k = \frac{e_{along}}{\left|\vec{V}\right|} = \frac{\left(\vec{U} \bullet \vec{V}\right)}{\left|\vec{V}\right|^2} = \frac{u_x v_x + u_y v_y}{v_x^2 + v_y^2} \tag{19}$$

Three conditions can exist: (1.) If $0.0 \le k \le 1.0$, then the projection of the aircraft's position point lies on the route segment defined by the points TJ_i and TJ_{i+1}. (2.) If $k < 0$ (which also means that that the along-track error e_{along} is negative), then the aircraft's position can be interpreted to be before the route segment. (3.) If $k > 1.0$, then the aircraft's position can be interpreted to be after the route segment.

C. Spatially-coincident Trajectory Prediction Errors

To calculate the spatially-coincident trajectory prediction errors, the point TJ_1 is selected to be the point on the nearest trajectory segment that has the shortest perpendicular distance. In general, the closest segment is the trajectory segment with the shortest perpendicular from the track point to the segment. This situation is depicted in Fig. 6. In this case the spatially-coincident trajectory prediction errors are the same as the time-coincident trajectory prediction errors. This occurs when the scalar k value defined in Eq. (19) is between zero and one. Depending on the geometry, the definition of the spatially-coincident position may be different. If the perpendicular intersects the extension of the segment ($k < 0$ or $k > 1.0$) the distance to the segment is not the length of the perpendicular; instead it is the distance from the track point to the nearer end of the segment. The segment with the minimum adjusted distance is considered to be the closest segment. This situation is depicted in Fig. 9, in which the closest perpendicular is on the extension of a trajectory segment. In this case, the point TJ_1 is defined to be the end point of the closest segment and TJ_2 is the point on the extended trajectory segment. If all the trajectory segments were evaluated and the minimum distance remained between the point TJ_1 and the aircraft, then this distance would be the spatially-coincident cross-track error.

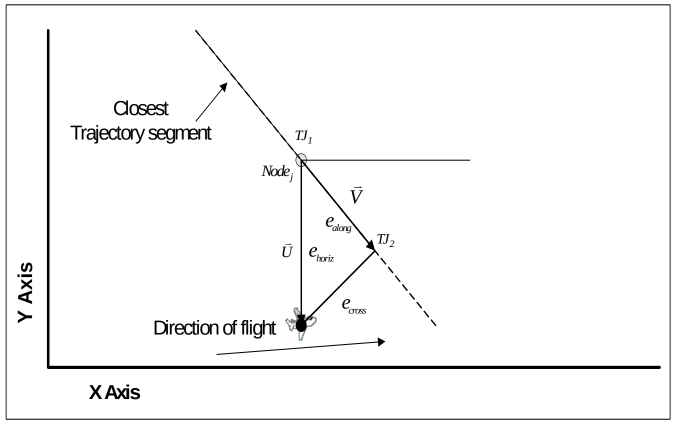

Figure 9. Example of Spatial-coincident Trajectory Prediction Errors

The spatially-coincident along-track error is the actual traveled distance along the trajectory between the time-coincident trajectory position to the spatially projected point on the trajectory. For example, for the simple case in Fig. 6, the spatially-coincident along-track error is the same as the time-coincident version. However, if the spatially-coincident cross-track error is formed from the distance to an end point of a segment as in Fig. 9, the along-track distance is calculated by again taking the traveled distance from the time-coincident trajectory position along the trajectory segments to this end point or node.

For spatially-coincident errors, the projected position on the trajectory also represents the closest point on the trajectory relative to the current position of the aircraft. It is often necessary to determine the coordinates of this projected position. For example, it can be used to calculate the time difference, referred to as predicted time error, between the projected point and the time of the time-coincident position. If on the segment (unlike an end node like Fig. 9), Eq. (20) provides the coordinates for this position, referred to as the projected position *r*.

$$x_r = kv_x + x_{TJ1} \text{ and } y_r = kv_y + y_{TJ1} \tag{20}$$

where x_r is the x-dimension stereographic coordinate, similarly y_r is the y-coordinate of this position, k value is defined in Eq. (19), v_x and v_y are defined in Eq. (8), and x_{TJ1} and y_{TJ1} represent the coordinates for the first node of the trajectory segment that the projected point *r* resides. The point TJ_1 in Eq. (20) is not the same point in Fig. (9), or in Fig. (6), but the actual trajectory segment's first node that the closest projected point *r* resides on.

V. Implementation

The implementation of these metrics to perform the trajectory accuracy measurement consists of a set of tools developed mostly internally by CPAT. The processing tools include a series of object oriented Java applications, linked together with a set of Linux shell scripts. The analysis uses both commercial off-the-shelf (COTS) analytical tools and graphical user interfaces developed under academic and government partnerships. As illustrated in Fig. 10, the processing these tools perform includes four main areas: (1) parsing of air traffic control data, (2) parsing of trajectory predictions, (3) sampling and measurement, and (4) analysis of results. At the heart of all the processing is a relational database that acts as both the catalyst and storage area for the input and output data.

For the first main processing area, the various sources of air traffic data are read and uploaded into a set of relational database tables. As described earlier, the air traffic data consists of air traffic controller clearances, such as flight plans and interim altitude messages, and position reports. This data provides the actual aircraft position for comparison to the predicted position, so errors here will cause the incorrect measurement of the trajectory prediction. Thus, the data is first interpolated to a parameter interval (nominally 10 seconds) and checked for reasonableness. This reasonableness checking includes detecting and correcting bad data points (e.g., a reported altitude of 0) or positions that surpass the typical flight envelope of civilian jet aircraft (e.g., ground speeds either too fast or slow). A set of algorithms are run resulting in approximately 2% of the flight data being dropped and another 1% being modified. This process is documented in detail in Ref. 11. The result is a database table containing the verified actual aircraft positions. These positions are now able to be used to confidently compare against the trajectory predictions.

The second main processing area is the parsing of the trajectory predictions. Since the source of the trajectory predictions may be from various DSTs, and therefore in various formats, the data is parsed and loaded into a standard database table. The prediction data is not checked for reasonableness like the actual air traffic data but is verified for completeness.

The third main processing area compares the actual corrected air traffic control data to the parsed trajectory predictions. The difference between actual and prediction constitutes the error in the trajectory prediction. The process is comprised of the sampling and measurement tasks combined. For sampling, the IBST, as defined in Section III, is applied to determine which of the actual aircraft positions and trajectory predictions should be measured. Once the positions are selected by the IBST, the measurements produce a trajectory metrics database table. As defined in Section IV, the metrics, such as horizontal and vertical errors, are captured as fields within the trajectory metrics database table. This table provides a means for the analyst to query for specific flights, summarize the results from all or groups of flights, and export the data to COTS or other external tools for analysis.

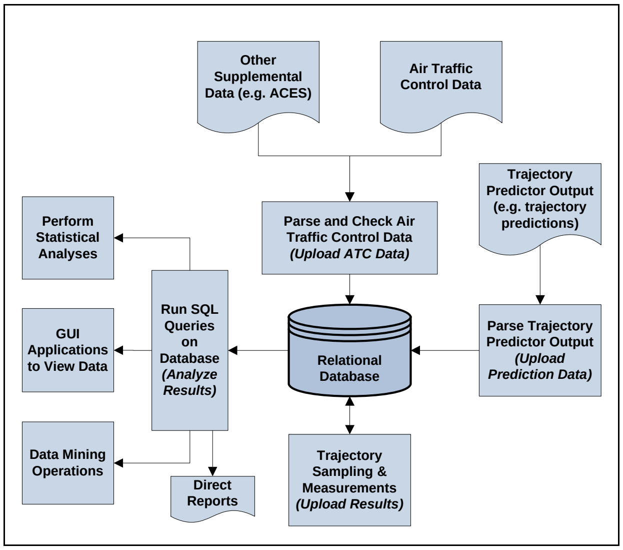

Figure 10. Implementation Processes

A key design feature of the trajectory metrics table is the "measure-and-flag-all" approach. All positions selected by the IBST have measurements calculated. However, many will be filtered out in the later analysis depending on flag settings and the interest of the analyst. For example, an aircraft position and trajectory is sampled by the IBST with a look-ahead time of zero. Consider that the next position selected for measurement with the same trajectory is at a look-ahead time of ten minutes and that during the time interval of zero to ten minutes an ATC clearance is issued at seven minutes. In this example, the clearance flag is set for the ten-minute look-ahead time measurement indicating that the ATC personnel cleared the aircraft either horizontally or vertically and the measurement is likely to be in error. It is left to the analyst whether or not to exclude this measurement. One point of view would consider all measurements valid and include the measurement in later analyses. Another point of view would exclude this flagged measurement, since the clearance was unavailable at the time the trajectory prediction was generated. This "measure-and-flag-all" measurement approach provides the analyst with the ability to decide which data to include or exclude based on particular analytical objectives.

The fourth and final processing area comprises the analysis step. All the results, as well as input files, are accessible within the comprehensive relational database. In particular, the trajectory metrics table contains all the errors calculated based on the IBST. SQL scripts are developed to directly calculate statistics on these measurements. The statistics are spooled to direct reports or exported into a COTS statistical package. Descriptive and/or inferential statistical analyses are performed on the results. Graphical user interfaces (GUIs) may also be used to plot the actual aircraft paths versus the trajectory predictions providing insight into the sources and geometries of the errors. Section VI provides detailed descriptions of these methods of analysis.

VI. Analysis of Trajectory Prediction Errors

The analysis step of the trajectory validation methodology utilizes the implementation in Section V to bring meaning to the error measurements defined in Section IV. There are many aspects to the analysis. Descriptive statistics summarize the data under study. For trajectory accuracy measurements, it provides basic summary information about the population of error measurements collected. It includes statistics such as the sample mean and sample standard deviation on the metrics defined in Section IV. Inferential statistics are applied to the data sets to help draw some conclusion about the results. Data mining techniques are effective methods to discover other unexpected information about the trajectory predictions. Individual flights are examined in detail via specially designed GUI applications that plot the aircraft positions and trajectory predictions overlaid on the airspace in which the aircraft flew. The following sub-sections will provide more detailed descriptions of the inferential statistical analysis performed on many flights and the complimentary individual flight-by-flight analyses.

A. Inferential Statistics

After defining and implementing TP metrics using a sampling technique as discussed in Section III, a common task is to use the results to test a DST's TP. Although there are many types of tests that utilize these methods, this section discusses regression testing as a representative example. After system deployment, regression testing performs selective testing of the system or a component of the system to verify that modifications (e.g., a new release) have not introduced new problems and that the system still complies with its requirements. Regression testing requires that a baseline version of the trajectory modeler software be run with a given traffic sample. This same traffic sample is then run through the upgraded software, which is referred to as the new release version. The analysts compare the trajectory accuracy of both runs using the interval-based sampling method as defined in Ref. 6. Analysts can examine several trajectory accuracy metrics simultaneously using this process, but for simplicity, this section will focus only on the cross-track error. As defined previously, the cross-track error is the perpendicular distance between the sampled aircraft surveillance position and the time-coincident trajectory predicted position expressed in units of nautical miles and has a positive sign if the prediction is to the right of the aircraft. To compare the runs, the difference between the baseline and new release sample mean is calculated. Since the sample mean is a statistic and thus a random variable of the true population mean, a statistical hypothesis test is used that considers the variation in both samples. If the true population means were known, the difference between the two means could be calculated exactly. If the difference were zero, it would be concluded that the runs were equivalent. As described by Devore in Ref. 8, the Two-Sample t-Test provides a statistical hypothesis test that provides a criterion to reject the hypothesis that the sample means are not equal. This null hypothesis is expressed as:

$$H_o : \mu_b - \mu_n = 0 \tag{21}$$

where μ_b is the population mean of the baseline run and μ_n is the population mean of the new release run.

The test assumes the trajectory measurements from each run are normally distributed random variables, and the runs are independent from one another. The following subsections explore both these assumptions further.

1. *Assumption of Independent Sample Runs*

Since the same air traffic sample is input into both runs of the trajectory model, the other variables that influence trajectory accuracy are expressed in the variability of flights in the two runs. These flights are the same for each run, therefore their influence has a proportional effect on both runs. If a specific flight exhibits higher than normal error in the baseline run, it would be expected that the same flight would have similar high error in the new release run. Of course, if the upgrade was to reduce these errors, some flights may exhibit better performance in the new release; but on average, if the flights perform in the same manner between runs, the runs are not independent. In Ref. 3, a trajectory accuracy example illustrated this lack of independence between runs, resulting in erroneous conclusions. An alternative technique was recommended and is presented again in the following section.

2. *Application of a Paired t-Test*

Instead of taking the difference between the sample means, the sample measurements are paired for the same flight and position. The large variability between flights and linear dependence between runs is effectively blocked out of the experiment. A new statistic, called the sample difference, is produced by calculating the difference between paired trajectory measurements of same flight and position from the two runs. This is expressed as:

$$D_i = x_i - y_i \tag{22}$$

where i is the particular measurement from the two runs, x_i is the trajectory measurement for the baseline run and y_i is the same for the new release run.

Therefore, the hypothesis now is that the sample mean of D_i's is equal to zero. For large sample sizes, the mean of the differences between two numbers is equal to the difference between the means of the same set of numbers. Therefore, while the hypothesis in Eq. (21) is the same, the test statistic compared to a Two-Sample t-Test is not (see Ref. 3 for details). The following equation expresses the Paired t-Test's test statistic:

$$t = \frac{\overline{D}}{s_D \big/ \sqrt{n}} \tag{23}$$

where $\overline{D}$ and s_D are the mean and sample standard deviation of the differences (i.e., the D_i's) and n is the sample size of these differences.

The rejection region of the Paired t-Test is expressed such that the null hypothesis is rejected if $t \geq t_{\alpha/2,n-1} \; or \; t \leq -t_{\alpha/2,n-1}$, where $t_{\alpha/2,n-1}$ or $-t_{\alpha/2,n-1}$ are parameters taken from the Student's t-distribution, α is the significance level of the test, and $n-1$ is the degrees of freedom for this test (number of samples minus one).

3. Example Application of the Paired t-Test

To test the hypothesis in Eq. (21) for the measurements of trajectory cross-track error, two runs were performed on an available trajectory modeler and the cross-track error was measured at a look-ahead time of 900 seconds. The sample scenario was based on two hours of recorded traffic data from the Indianapolis Air Route Traffic Control Center (ARTCC) in May 1999. The trajectory modeler produced over 5000 trajectories for each of the runs. The baseline run produced a sample mean of 0.60 nautical miles of cross-track error and a sample standard deviation of 5.58 nautical miles (square root of the sample variance). The new release run produced a sample mean of 0.56 nautical miles and sample standard deviation of 5.62 nautical miles. The sample mean of the differences is 0.038 nautical miles and sample standard deviation of the differences is 0.559 nautical miles. Since the same traffic sample was run through the trajectory modeler, both runs are balanced with the same quantity of 832 measurements of cross-track error.

As shown in Ref. 3 and discussed previously in this paper, the Paired t-Test offers significance precision due to the heterogeneity in the runs. By applying Eq. (23) on the above values, the test statistic t equals 1.99. The rejection region defined in the above Section VI.2 equals ± 1.96, using a significance of 0.05 and 831 degrees of freedom. This value is found in Table A.5 of Ref. 6 as the critical value taken from a Student's t-distribution. Therefore, the hypothesis that the mean horizontal error of the two runs is equivalent can be rejected (i.e., $t \; is \; \geq t_{0.025,831} \; or \; \leq -t_{0.025,831}$). Therefore, the upgrade or new release trajectory model is considered statistically different to the previous baseline version. In this case, it has slightly less error.

As discussed in Ref. 8 and shown explicitly in Ref. 3, the Paired t-Test has a property of improving the precision of the test statistic when there is a correlation between runs and significant heterogeneity between samples (in this example the difference between flights).

4. Assumption of Normality of Samples

Even though the data was paired correctly, the result in the previous example is surprising, since the difference in sample means was only 0.038 nautical miles. Further inspection of the data showed that six measurements of the 832 total were more than six standard deviations larger than the sample mean of the differences. Removal of these six outliers produced very different results with a test statistic of only 0.116, which is well below the 1.96 rejection criterion.

In Ref. 8, Devore offers insight into why the Paired t-Test was sensitive to the outliers in the example. The underlying Student's t-distribution used in the test statistic is approximately normally distributed with large sample sizes, which is often the case with trajectory accuracy measurements. Normally distributed parametric tests can perform poorly when the underlying distribution has heavy tails. These tests depend on sample mean that can be very unstable in the presence of heavy tails caused by outliers. Alternative non-parametric approaches relax the assumption of normality and rely on a more robust metric, the sample median of the observed values. This approach is presented in detail in Ref. 9.

In summary, the parametric Paired t-Test is a useful inferential statistical tool to use for regression testing a trajectory predictor. It is an example of the type of inferential techniques that can be employed to determine with some statistical confidence that your trajectory predictions have not degraded over time. In addition, more robust alternate techniques have been introduced that build on this Paired t-Test using non-parametric approaches.

B. Graphical User Interface Tools

Analysts can also use graphical tools to visualize the TP metrics. In 2001, CPAT supported the development of a prototype tool named TrajectoryGUI that was written in Java and interacted with an Oracle relational database using the Java Database Connectivity (JDBC) Application Program Interface (API)[§§] to execute the database queries for collecting flight data and flight trajectories. In 2005, the Software Engineering, Graphics, and Visualization (SEGV) Research Group at Rowan University developed an upgraded version of the TrajectoryGUI for CPAT; initially as part of a senior class project and then through a funded internship by the FAA and a selected Rowan student. While still written in Java using the JDBC, for this upgrade, the SEGV completely redesigned the application to use the JOGL[***] library, which implements the OpenGL graphics standard[†††], to provide the plots. This upgraded version is documented in Ref. 10.

When an analyst launches TrajectoryGUI, a selection window is generated. An example is presented in Fig. 11. The analyst uses this window to select the specific data to be used for plots and tabular display during the session.

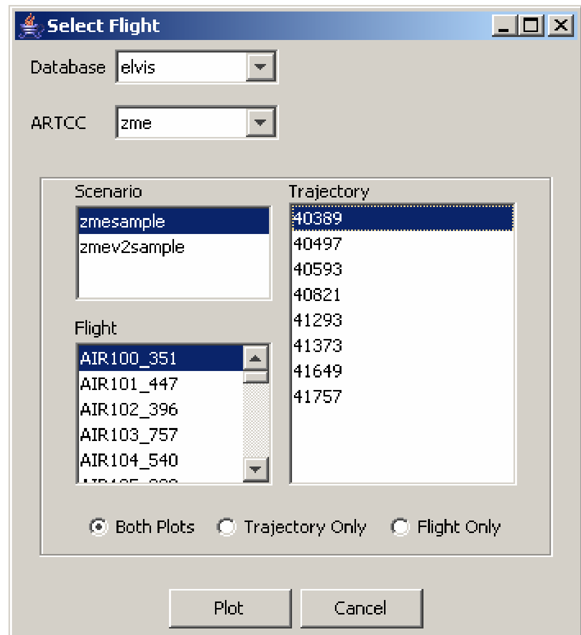

Figure 11. TrajectoryGUI: Selection Window[‡‡]

The analyst first identifies the database that contains the tables providing the desired data for this session. These databases contain the trajectory accuracy measurements and supporting traffic data[3]. The analyst then selects the appropriate ARTCC from another drop-down list. In the example shown in Fig. 11, the analyst has identified a local database named “elvis” and has selected ZME, which is the identifier for the Memphis ARTCC.

TrajectoryGUI uses the database and ARTCC information to query the database and fill the scenario list area, which will identify the available scenario cases. In the example, two scenarios were identified and the analyst has selected the scenario case identified as ZMESAMPLE. This selection causes the flight list text area to be filled, identifying the available flights. In the example, the analyst has selected flight AIR100_351, which represents a flight with the aircraft identification (ACID) of AIR100 and the computer identification (CID) of 351. This selection causes the trajectory text area to be populated with the build times of the trajectories that were generated by the DST’s TP. The analyst now selects the desired trajectory to be plotted. In the example, the analyst has selected the trajectory with a build time of 40,389 seconds.[‡‡‡] At this point the analyst can use the radio buttons to select the plotting option: trajectory and flight path, trajectory only, or flight only. After the desired option has been chosen, the analyst then clicks the plot button.

---

‡‡ Adapted from Figure 1 in Ref. 10.

§§ The Java Database Connectivity (JDBC) is a Java API that enables Java applications to execute Structured Query Language (SQL) statements providing database connectivity with a wide variety of SQL-compliant databases. For further information see http://java.sun.com/products/jdbc/.

*** JOGL is short for Java bindings for OpenGL. The JOGL Project hosts a reference implementation for OpenGL API, and is designed to provide hardware-supported 3D graphics written in Java. It is part of a suite of open-source technologies initiated by the Game Technology Group at Sun Microsystems. For further information about the JOGL API Project see https://jogl.dev.java.net.

††† OpenGL is an open standard for developing portable, interactive 2D and 3D graphics applications that is guided by the OpenGL Architecture Review Board. For further information see http://www.opengl.org.

‡‡‡ Time measurement is in seconds of the day based on 86,400 seconds in a day.

TrajectoryGUI first queries the database and presents the application's main window. The analyst can choose from a number of two-dimensional plots. As illustrated in Fig. 12, this example presents an X-Y plot and a T-Z plot with the metrics table for the selected flight. This window also provides the main interface with the analyst. The X-Y plot, located on the left side of the window, presents the positional data for the flight in a horizontal plane. It is a uniform scaled graph with units in nautical miles. The positive X-axis represents east and the positive Y-axis represents north. The T-Z plot, located on the right side, presents the altitude data for the flight. The vertical units are feet and represent altitude with 0 being sea level. The horizontal units are seconds with the leftmost position of the graph denoting the beginning of the flight. The metrics data is presented in a table located below the two plots. The table is filled with the accuracy measurement data selected from the database. At the bottom of the main window is a text area that is used to report information such as operational results, program mode, status, and errors. At the top of this window, an array of functions is available to the analyst that can be used to probe the plots.

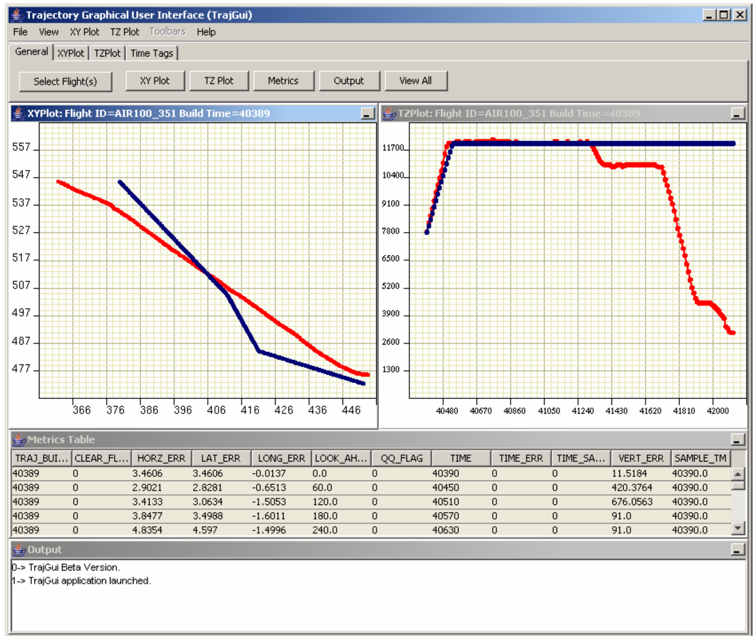

Figure 12. TrajectoryGUI: Main Window[§§§]

Figure 13 shows the result of re-centering to the area around the beginning of the flight data and zooming in to a plot view area of 100 square nautical miles. The offset functionality is useful if the trajectory lies near or directly on top of the flight's actual path. If this is the case, the analyst can offset the trajectory creating separation between the data. The analyst uses the *Move Legend* function when the legend is located in an area where data is displayed. Resetting a plot allows the analyst to return to the initial viewing area of the plot.

The T-Z plot window displays the T (time) and Z (altitude) coordinates of the flight's track data and the flight's trajectory data in an interactive coordinate system. Many of the same features exist for the T-Z plot, but they are used independently within the plots.

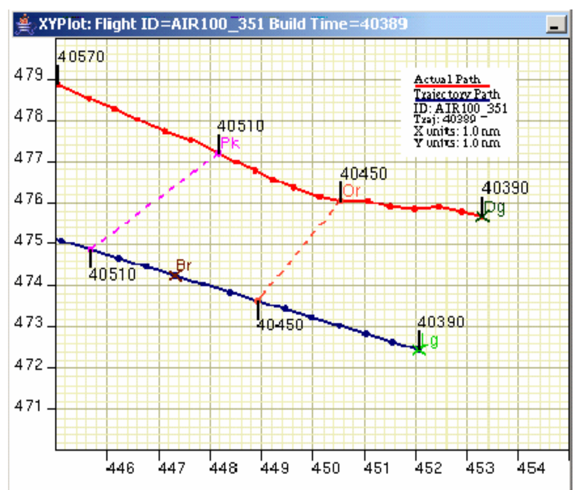

Figure 13. TrajectoryGUI: Manipulated X-Y Plot[****]

Additional capabilities of TrajectoryGUI include: an image export of the X-Y and T-Z plots for use in presentations and reports, a file export of the metrics data in a comma delimited file for import into other application programs, an interface that provides the ability to select and deselect the metrics that are displayed in the metrics table, a configuration file for maintaining lists of accessible databases along with their connection parameters, the default metric fields to be included in the metrics table, and the desired colors for display of the actual flight and trajectory paths.

[§§§] Adapted from Figure 2 in Ref. 10.
[****] Adapted from Figure 3 in Ref. 10.

VII. Case Study

In this section, examples of a trajectory accuracy analysis implementing this methodology are provided first on a single sample flight and next a scenario of many flights.

A. Sample Flight's Analysis

The sample flight documents the trajectory prediction of a civilian airliner traveling through Washington ARTCC (ZDC) originating from Dallas Fort Worth, Texas with the destination of John F. Kennedy International Airport (JFK) in New York. Figure 14 illustrates the top down stereographic view of the aircraft's horizontal path overlaid on the ZDC high-altitude sectors it through which it travels. On its journey to JFK, the sample flight is traveling in a northeasterly direction where ZDC receives its air traffic control for the flight at 20:14 UTC (Coordinated Universal Time) from Indianapolis ARTCC. Figure 15 illustrates the time versus altitude profile of the aircraft. It enters ZDC at Flight Level 390 (FL 390) and at approximately 20:29 UTC is cleared to descend to FL 380. It begins its descent to FL 380 about two minutes later. It receives a series of descent clearances and is handed-off to New York ARTCC at 20:56 UTC during a brief cruising segment at FL 240.

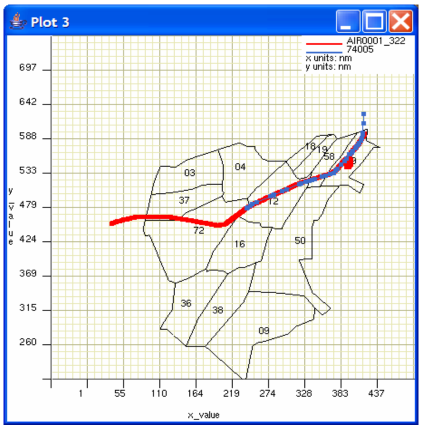

Figure 14. TrajectoryGUI: Overall Track Versus Trajectory X-Y Plot of Sample Flight

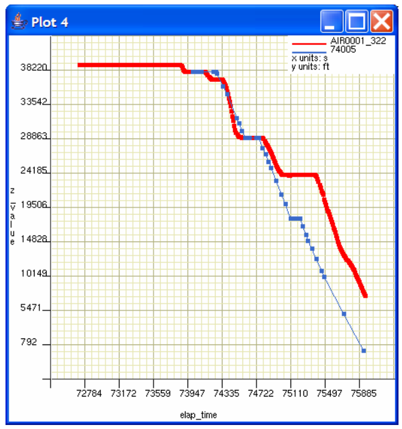

Figure 15. TrajectoryGUI: Overall Track Versus Trajectory T-Z Plot of Sample Flight

The focus of this example is the trajectory built at 74005 seconds (20:33:25 UTC). This trajectory is illustrated in both Fig. 14 and Fig. 15 (blue segmented line) and overlaid with the surveillance track positions (red thicker line). Of particular interest is the complete turn performed later in the flight beginning roughly at 20:50 UTC. Clearly, the trajectory does not reflect this turn, which probably was a result of a verbal air traffic control clearance not entered into the automation system. An extraction of the trajectory metrics calculated for the 74005 second trajectory is listed in Table 1 below. A sample was taken at 74040 seconds (20:34:00) with a look-ahead time every five minutes up to 20 minutes in the future. At the first measurement time at look-ahead time of zero, the horizontal error was almost a half a mile with a zero vertical error. However, as the look-ahead time progressed and approached the turn depicted in close-up view in Fig. 16, the horizontal errors increased significantly. Due to the un-modeled turn, the error reaches up to 32 nautical miles horizontally. The clearly visible cross-track error is approximately 12 nautical miles, but the bulk of the error is found in the along-track error. The additional travel time caused by the turn manifests in as much as a -32 nautical mile along-track error, which translates to as much as 4.4 minutes lag in the trajectory prediction.

The vertical error for the first measurement is zero but increases as the flight descends through a series of clearances not available at the trajectory build time. Thus, the column labeled "Clear Flag" in Table 1 represents when a clearance was initiated after the trajectory was sampled, explaining the vertical errors shown.

The analysis discussed so far is for just one of the trajectories generated for this flight. The TP generated 34 predicted trajectories for this flight. Using the 20 minute look-ahead time window with a step of five minutes and a sampling time of every two minutes, the IBST sampled 18 of these trajectories, producing over 109 measurements with an average of about six measurements per trajectory. The median error for all the sampled trajectories for horizontal error and the unsigned cross- and along-track errors are plotted in Fig. 17. As expected the errors in general increase as the look-ahead time increases.

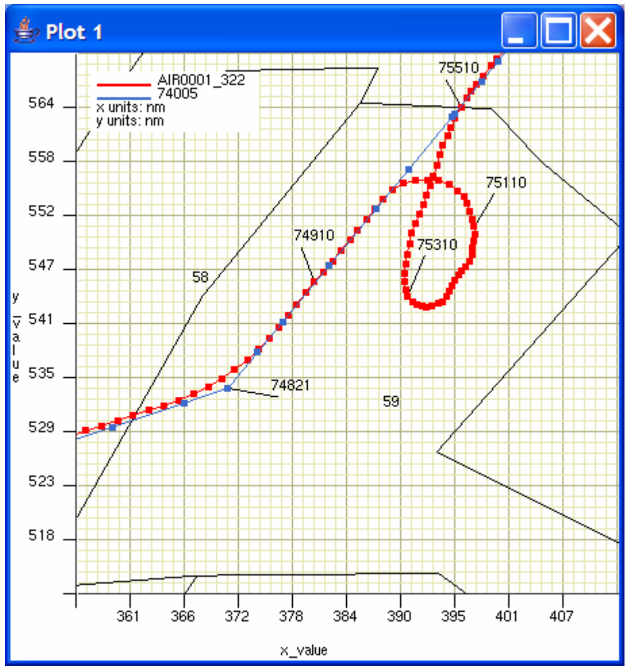

Figure 16. Close View of Track Versus Trajectory X-Y Plot of Sample Flight

Table 1: Sample Flight's Trajectory Metrics Data

Sample Time	Measurement Time		Look-Ahead Time	Horizontal Error	Cross-track Error	Along-track Error	Vertical Error	Clear Flag
Seconds	Seconds	HH:MM:SS	Seconds	Nautical Miles	Nautical Miles	Nautical Miles	Feet	
74040	74040	20:34:00	0	0.4	0.3	-0.3	0	0
74040	74340	20:39:00	300	0.1	-0.1	0.0	793	1
74040	74640	20:44:00	600	1.2	-0.5	-1.0	0	1
74040	74940	20:49:00	900	2.1	-0.1	2.1	2096	1
74040	75240	20:54:00	1200	34.6	11.9	-32.5	6952	1

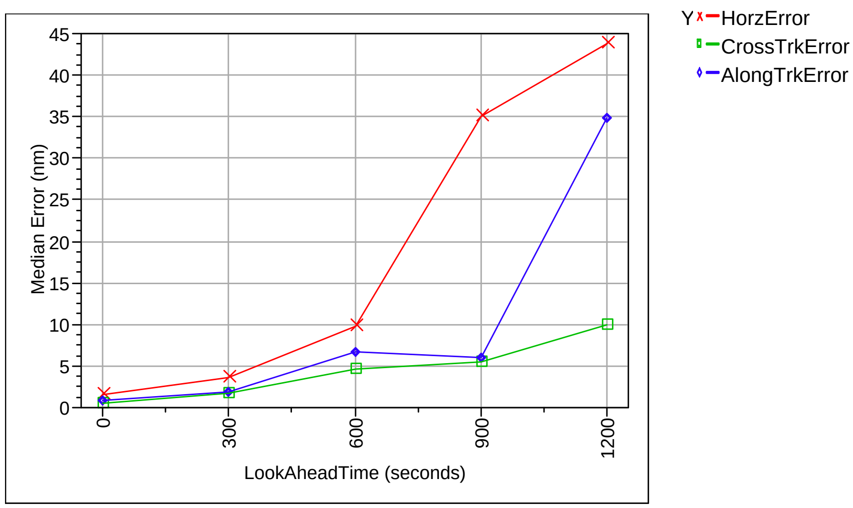

Figure 17. Sample Flight's Median Error per Look-ahead Time per Error Type

B. Scenario of Flight's Analysis

The full traffic scenario, from which the sample flight was extracted, was recorded on March 17, 2005 and contained approximately 460,000 surveillance track reports and 14,000 air traffic control clearances. The recording began about 18:00 UTC and ended at 23:30 UTC, capturing the typical afternoon peak of traffic for the day. Following the reasonableness checks and requiring a flight plan clearance to precede the track reports, there were 2406 flights recorded and available for analysis. For various reasons, only 2024 flights were modeled and available for analysis by the TP. This produced approximately 140,000 trajectory accuracy measurements under the same IBST parameters as in the sample flight (e.g. sample time two minutes and look-ahead window up to 20 minutes). Figure 18 illustrates the errors in the horizontal dimension. The horizontal error is an unsigned error and the sign of the cross- and along-track generates a symmetric distribution about zero. The area of the histograms provides an indication of the overall performance. From Fig. 18, the along-track error exhibits a larger area than cross-track error. Thus, more flights and measurements have larger errors in the along-track dimension, providing a bigger impact on the horizontal error.

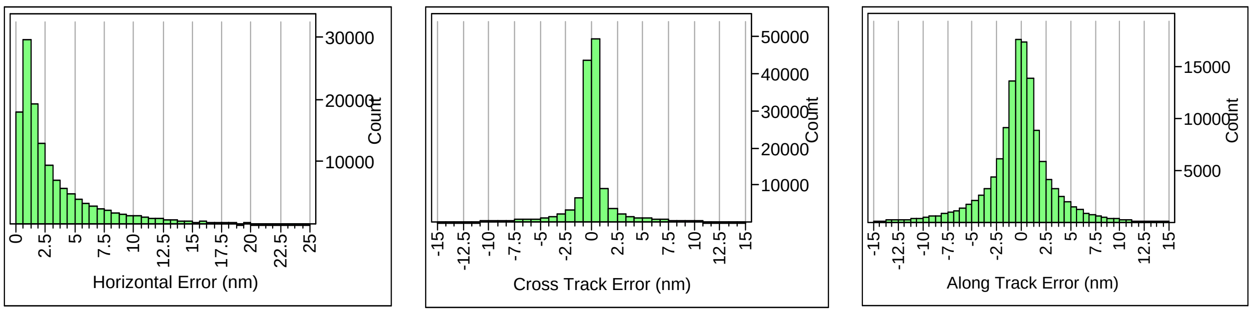

Figure 18. Histograms for Horizontal Dimension Trajectory Errors for All Flights

As discussed in Section VI.A, inferential statistics can be applied to determine the impact of many factors that may influence the performance of the TP or test whether the TP has degraded from one version to the next. One example of such an analysis is to determine if the TP has a larger error when the trajectory was built before the first track position report was captured versus later in the flight. The expectation is that the TP utilizes the initial surveillance track reports to synchronize with the intent information in the flight plan. For the same reasons the difference between flights was presented in Section VI.A's hypothesis test, in this example the difference between the median horizontal errors per flight was calculated. The hypothesis test then serves to evaluate whether the trajectories built without track data is equivalent statistically to the trajectories built with this information. After applying the Paired *t*-Test as described earlier in Section VI.A, the results of 1823 differences produced a mean difference of -3 nautical miles and a test statistic of -21.89, providing overwhelming evidence to reject the null hypothesis. This conclusion is strengthened by the histogram presented in Fig. 19. The distribution of the differences is clearly skewed negative indicating that the evidence overwhelming shows that the trajectory's horizontal error is statistically larger before track begins. This result is not surprising, but clearly shows how statistical tests, particularly the Paired *t*-Test, provide a powerful method for investigating a TP's performance.

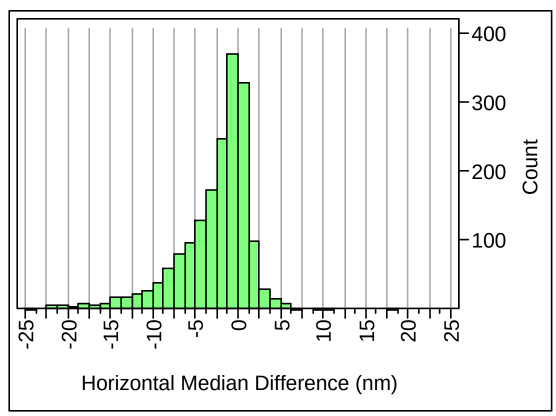

Figure 19. Difference in Horizontal Median Error per Flight between Early Trajectories and Not

VIII. Conclusion

A trajectory validation methodology must first measure the performance errors of the TP, and then identify the sources of these errors, thus facilitating an improvement to the TP. This paper addressed a specific methodology and implementation to measure the trajectory performance errors of a TP. The most challenging part of this process is its sampling, measuring, and analyzing tasks, which were described in this paper. The key contribution of this implementation is the use of a relational database of metrics to perform two complimentary analyses. First, individual flights are examined in a micro-view using both graphical and numerical results; then the findings from the individual flights are supplemented by descriptive and inferential statistical approaches in a macro-view to draw overall conclusions of many flights. In tandem, these techniques provide the foundations to performing a full TP validation.

It is noteworthy that this approach has been used for many years to measure the trajectory accuracy of DST's such as URET and it continues to be applied to systems such as ERAM, which relies even more on its underlying TP's predictions within its flight data processing functions. In the future National Airspace System (NAS) of 2025 as described in Ref. 12, the FAA through the Joint Planning Development Office (JPDO) envisions an aviation system that relies on trajectory-based operations to manage the forecasted ever increasing high-density and high-complexity air traffic demand. This future TP driven NAS will require the techniques presented in this paper even more.

References

[1]Paglione, Mike, Carlos Garcia-Avello, Sip Swierstra, Robert Vivona, Steven Green, October 2005, *A Collaborative Approach to Trajectory Modeling Validation*, 24th Digital Avionics Systems Conference, Washington, DC.

[2]Mondoloni, Stéphane, Sipka Swierstra, and Mike Paglione, October 2005, *Assessing Trajectory Prediction Performance – Metrics Definition*, 24th Digital Avionics Systems Conference, Washington, DC.

[3]Cale, Mary Lee, Shurong Liu, Robert D. Oaks, Mike Paglione, Dr. Hollis F. Ryan, and Scott Summerill, December 2001, *A Generic Sampling Technique for Measuring Aircraft Trajectory Prediction Accuracy*, 4th USA/EUROPE Air Traffic Management R&D Seminar, Santa Fe, NM.

[4]Brudniki, D. J., W. C. Arthur, and K. L. Lindsay, April 1998 *URET Scenario-based Functional Performance Requirements Document*, MTR98W0000044, The MITRE Corporation, McLean, VA.

[5]Paielli, R. A., and H., Erzberger, October 1999, *Conflict Probability Estimation Generalized to Non-Level Flight*, Air Traffic Control Quarterly, vol. 7, no. 3.

[6]Paglione, Mike M., Dr. Hollis F. Ryan, Robert D. Oaks, J. Scott Summerill, and Mary Lee Cale, May 1999, *Trajectory Prediction Accuracy Report User Request Evaluation Tool (URET) / Center TRACON Automation System (CTAS)*, DOT/FAA/CT-TN99/10, U.S. Department of Transportation, Federal Aviation Administration, William J. Hughes Technical Center, Atlantic City International Airport, NJ.

[7]Ryan, Hollis F., Mike M. Paglione, Steven M. Green, August 2004, *Review of Trajectory Accuracy Methodology and Comparison of Error Measurement Metrics*, Proceedings of the American Institute of Aeronautics and Astronautics (AIAA) Guidance, Navigation, and Control Conference, Providence, RI.

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

[9]Paglione, M. M., July 2004, *Non-Parametric Approach to a Paired Data Hypothesis Test in Trajectory Accuracy Analysis*, International Conference on Computational & Experimental Engineering Sciences (ICCES), Madeira, Portugal.

[10]Santiago, C. III, M. M. Paglione, A. S. Rusu, S. Liu, R. D. Oaks, and S. Putney, October 2005, *Using Graphical Software for Evaluating Aircraft Trajectory Predictions*, 50th Annual Air Traffic Control Association Annual Conference Proceedings, Grapevine, TX.

[11]Ryan, Hollis F., Mike M. Paglione, August 2003, *Heuristic Methods to Post Process Aircraft Radar Track Data*, Proceedings of the American Institute of Aeronautics and Astronautics (AIAA) Guidance, Navigation, and Control Conference, Austin, TX.

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