JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 18, No. 5, September–October 1995

Expert System for Processing Uncorrelated Satellite Tracks

Michael Hecker
U.S. Naval Postgraduate School, Monterey, California 93943

Through an array of ground-based radar sites and optical cameras, the United States military tracks objects in near and far Earth orbit. The sensors provide epoch and ephemeris information that is used to update a database of known objects. Although the majority of the sensor observations are matched to their corresponding satellites, a small percentage are beyond the capabilities of current association software and can not be correlated. These uncorrelated targets must be manually fitted by orbital analysts in a labor intensive process. As an alternative to this human intervention, the use of artificial intelligence techniques to augment the present computer code was explored. Specifically, an expert system, a potential replacement for an orbital analyst, was developed for processing uncorrelated targets. An evaluation function was used in conjunction with a rule-based system to capture the orbital analysts' expertise. The initial results are very good with the tested portions of the system matching the performance of the experts with an accuracy of 99%. Although not yet complete, the code developed in this research illustrates the potential of using artificial intelligence to process uncorrelated targets.

Nomenclature

f_k	= kth feature of a candidate
h	= altitude of an observation
ID	= boolean variable; true if the current object under consideration is the same as the object originally associated with a UCT, false otherwise
i	= orbital inclination
P	= orbital period
$\dot{P}$	= orbital decay rate
p	= number of observations in a track
w_k	= kth feature of a prototypical vector used to define a class of objects
$\Delta U, \Delta V, \Delta W$	= errors along three orbital axes between an observation and a propagated orbit
$\Delta\bar{U}, \Delta\bar{V}, \Delta\bar{W}$	= average errors along three orbital axes between the observations that constitute a track and a propagated orbit
$\Delta W > 10$	= number of observations in a track with a ΔW magnitude greater than 10
μ_A	= fuzzy membership in the set of acceptable tracks, tracks with desirable attributes
μ_{GOOD}	= fuzzy membership in the set of good tracks, tracks the expert will consider fitting to the database
μ_R	= fuzzy membership in the set of tracks that should be rejected, tracks with undesirable attributes
Ω	= right ascension angle

I. Introduction

IN the United States, the military is responsible for maintaining a database of objects in low Earth orbit (LEO) and geosynchronous Earth orbit (GEO). Known as the orbital catalog, it provides a current picture of the near space environment that is used by both the military and civilian communities. To update the database, space pointing radar sites and optical cameras provide epoch and ephemeris information on objects as they pass overhead. These sensor observations are then used to update and maintain the currency of the orbital catalog. Correlating the sensor data to known objects in the database is a nettlesome problem of multiple tracks with multiple targets and is primarily handled in software.

Each entry in the orbital catalog identifies an object and the elements used to define its orbit. Additionally, the catalog maintains a record of the last 75 observations attributed to the object. Figure 1 outlines the primary method used to update the catalog. As an orbiting object passes through a sensors field of view, one or more observations are produced. The observations from a single pass are grouped into a track. The raw information from the intercept is converted to geocentric coordinates and presented to software as right ascension, declination, altitude, and time of observation. Based on a priori knowledge, sensors also make an initial correlation, called a tag, between a sighted object and known objects in the orbital catalog.

As observations are received, they are stored in a queue. Every 15 min the queue is emptied, and an attempt is made to fit the data to the orbital catalog. Using the sensor assigned tag as an index into the database, the orbital elements of an object are retrieved. Software then uses a modification[1] of the Brouwer–Lyddane model

CDR Hecker received a bachelor's degree in Aerospace Engineering from the U.S. Naval Academy in 1979 and master's degrees in Computer Science and Astronautical Engineering from the Naval Postgraduate School, Monterey, California in 1992. Primary area of interest is practical applications of artificial intelligence. He is currently assigned to the Naval Space Technology Program in Washington, D.C., where he manages the systems engineering division. He is a member of the AIAA.

Received Jan. 25, 1994; revision received Nov. 1, 1994; accepted for publication Nov. 15, 1994.

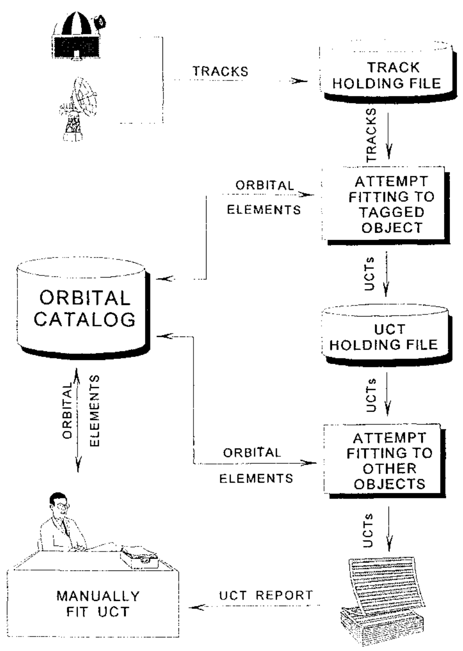

Fig. 1 Data flow for UCT processing.

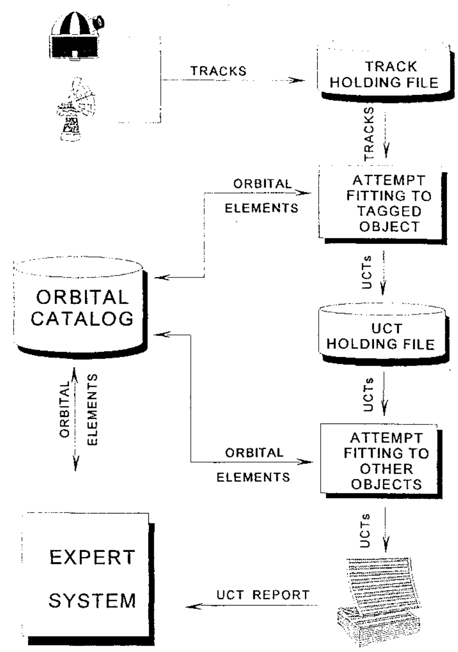

Fig. 2 Role of the expert system.

to propagate the object's orbit out to the epoch of the sensor observations. The observed position of the satellite as reported by the sensor is then compared with the predicted position given by the algorithm and, if the differences are within tolerance, the observations are assigned to an object. Using least-mean-square for curve fitting, these new observations are then combined with previously recorded observations to develop an updated orbit for the satellite. This new orbit is defined by a new set of orbital elements which are then written back to the database.

Tracks that fail to fit using the described process are designated uncorrelated targets (UCT) and are written to a holding file for off-line processing. Once a day, software makes a more extensive attempt to fit uncorrelated tracks by broadening the search through the orbital catalog for a more suitable object to associate with the observations. Unlike the realtime processing, this search will consider candidate objects other than the one originally given by the sensor tag.

Should the methods fail, then automated processing of observations is abandoned. For the remaining UCTs, a report is generated that lists the tracks and the residual errors that result when the uncorrelated tracks are compared to various orbits in the database. This listing is presented to the human experts, orbital analysts, who manually fit the observations to the database. Using heuristics and mental pattern matching, they scan the printout, select good observations, and use an iterative process in an attempt to fit the selected UCTs to the database. If they are successful, the updated orbit is again used to define a new set of orbital elements that is written back to the database.[2]

II. Problem Statement

Automated processing of UCTs is based on a truncated series solution to the Brouwer–Lyddane model. This solution is simple enough to be implemented as a realtime computer algorithm capable of processing the 230,000 observations received each month. However, accuracy is limited to several hundred meters, making the use of this model a compromise between computational speed and numerical accuracy. Consequently, whereas the overall performances of software is very good, approximately 1.5% of the incoming observations—3000–4000 per month—cannot be correlated to objects in the database and result in UCTs that must be manually fitted by human experts. Although the orbital analysts are quite adept at handling the uncorrelated tracks, the effort is tedious and requires two to four workhours each day. To reduce this burden, an expert system is proposed. As shown in Fig. 2, the goal is to replace the person in the loop.

For this application, the primary advantage of an expert system over traditional software solutions is that no model is used. Generally, improving the accuracy of a solution method requires a more complete mathematical model. In the particular case of Brouwer–Lyddane, uncertainties in accounting for atmospheric drag are the principal source of misassociations. For orbital prediction, several special theory models are available that address this problem.[3] Whereas it may be possible to reduce the number of UCTs through the increased accuracy afforded by these models, such improvements come at the cost of several orders of magnitude in computational complexity. Given current resources, a realtime implementation of a special theory model is not feasible.

By contrast, an expert system focuses on emulating the decision-making processes of a human specialist and is only indirectly concerned with the problem's underlying mathematics and physics. The expert system has an efficiency governed by the complexity of the rule set used by the expert and, with effective rules, has the potential for quicker runtime performance. Because the orbital analysts rely on pattern matching when processing UCTs, the ultimate success of an artificial intelligence approach depends on efficiently capturing in software this pattern-matching capability.

III. Methodology

A. Expert System Approach

An expert system has been defined as "a computer program that represents and reasons with knowledge of some specialized subject with a view to solving problems or giving advice."[4] The objective of an expert system is to capture in a computer program the knowledge and experience of an human authority. Once this knowledge has been encoded, the computer can be used to solve problems that previously required the skills of the human operator. As shown in Fig. 3, the key elements of an expert system are the problem domain, the knowledge base, and an inference engine. The problem domain is the input data provided to the expert system. It includes any facts deemed pertinent to the situation the system was designed to analyze. The knowledge base is a set of heuristics that can be used to find a solution to the problem. These heuristics take the form of rules that are used to guide the search. The inference engine does the actual work of finding a solution. Using one of several techniques, it applies the knowledge base to the problem domain until a conclusion is reached or the available facts are exhausted. In effect, the problem domain represents facts about the problem, the knowledge base is the expertise that can help solve the problem, and the inference engine controls the search for a solution.

Table 1 Satellite information, first line of data

Parameter	Value
Satellite ID	19163
I, deg	64.95
P, min	675.73
$\dot{P}$, min/day	−0.0003
International designation	86 043 A
Ω, deg	139.05
Element set	
epoch (date), yr, month, day	920211

Table 2 Observation data, second and following lines of data

Parameter	Value
Track epoch (date), yr, month, day	920210
Track epoch (time), h, min, s.s	041454.07
h, nmile	10317
ΔU, nmile	−13
ΔV, nmile	−75
ΔW, nmile	39
Tag	21854
Sensor	404

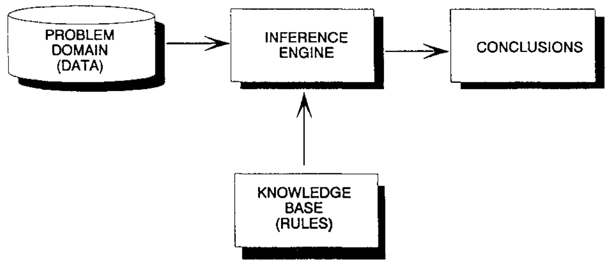

Fig. 3 Generic expert system.

As this problem is defined, the list of UCTs provided by software in conjunction with the orbital catalog represent the problem domain. Rules for the knowledge base were elicited from the orbital analysts through a knowledge engineering process. The knowledge engineering was accomplished through interviews between the system developer and the human experts, example cases of UCTs that had been correctly correlated, and as explained subsequently an application of machine learning. For an inference engine, NASA's CLIPS was selected.[5]

B. Internal Representation of Facts

On a conceptual level, the experts and the expert system deal with tracks with a UCT being a track that has yet to be matched to any object in the orbital catalog. Tracks are composed of one or more observations from a single sensor of an object in orbit. Observations, in turn, are composed of several fields of data. The printout of UCTs reviewed by the orbital analysts contains one to two thousand entries in the following format:

```
19163 64.95    675.73 −0.0003 86043   A 139.05  920211
  920210  041454.07  10317  −13   −75 39 21854   404
  920210  041535.20  10316  −15   −79 44 21854   404
  920210  041616.34  10315  −15   −96 27 21854   404
  920210  041657.47  10316  −15   −86 35 21854   404
```

The first line lists the satellite that the program is currently matching to the observations. This information comes from the orbital catalog and is known data about an object in orbit. Following the satellite data is one or more lines of observation data. Meanings of the various fields, along with example values taken from the first two lines of the entry, are explained in Tables 1 and 2. Taken in its entirety, the entry provides selected information about an object, a track, and the distance of the track from the object's projected orbit. In this example, sensor 404 assigned a track of four observations to the object numbered 21854 in the orbital catalog. The first observation took place at dawn on February 10,1992 with an altitude of 10,317 nmiles. The processing algorithms determined that this track, in fact, did not belong to object 21854, thus generating a UCT. Attempts are made to fit the UCT to the orbits of several other objects, in this case 19163. The resulting ΔU, ΔV, and ΔW errors for the first observation in the track are −13, −75, and 39 nmile, respectively.

Table 3 UCT feature set

Parameter	Value
Sensor	404
P, min	675.73
$\dot{P}$, min/deg	−0.0003
$\Delta\bar{U}$, nmile	−15
$\Delta\bar{V}$, nmile	−84
$\Delta\bar{W}$, nmile	36
$\Delta W > 10$, nmile	4
p	4
Tag	21854
ID	0
Age, days	−1

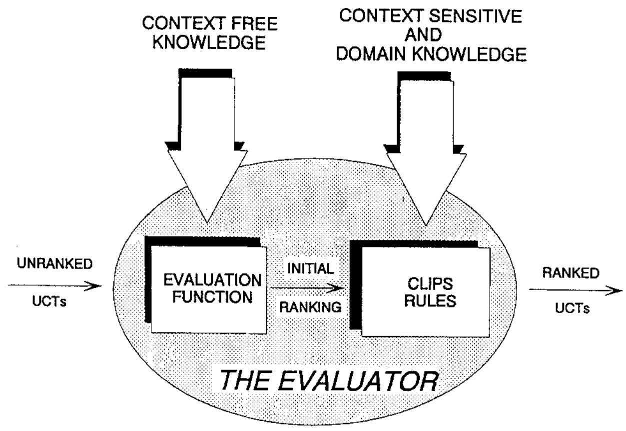

Fig. 4 Expert system architecture.

When analyzing UCTs, the experts are not so much interested in the individual observations as the aggregate features of tracks as a whole and how well a track correlates with a particular orbit. This implies that facts about a track should be based on statistical values derived from the observations that make up that track. Statistics can provide a variety of parameters for analysis; pertinent ones were selected based on the information gleaned during the knowledge engineering process and refined during the development of the expert system. The goal is to present to the expert system only that information that supports the expert's decision-making process and remove any extraneous data. Data available to the orbital analysts but not used included inclination, altitude, and ascension angle.

UCTs took the form shown in Table 3. This is the internal representation of facts, the representation of data that the expert system is aware of and utilizes.

C. Internal Representation of Knowledge

The work performed by the orbital analysts can be broken down into two functional parts. The first is an evaluation phase where the list of UCTs is canvassed and good candidates are selected. The good candidates are then manually fit to the database using an iterative technique similar to the synthesis process used in engineering design. Consequently, the expert system has two modules: the evaluator and the synthesizer. It is shown conceptually in Fig. 4. The evaluator examines how well a given UCT correlates to a selected orbit and ranks the results. The rankings are then passed to the synthesizer that performs the task of actually fitting the good data to the database. An existing support routine that performs the curve fitting is called during the iterative attempts to fit the data.

D. Evaluator

Evaluation is the process of deciding between a set of options known as candidates. The purpose is to determine the worthiness of candidates relative to each other with the goal of selecting the best one. As defined by Mitri,[6] it is "described in terms of establishing numeric scores or qualitative ratings for a candidate" where the candidate is "represented in terms of attributes (criteria) that are relevant to the evaluation." In the case of processing UCTs, features come from the association of a track with an object in the catalog. Orbital decay rate, errors along the three axes, and the number of observations in the track are a few of the criteria deemed important by the experts when examining UCTs. Some of these attributes come from the observations that make up the track whereas others are based on the association of the track with a particular satellite.

Evaluation functions are algorithms that determine the goodness of a candidate. They first appear in artificial intelligence research into gaming theory, notably in the works of Samuel[7] and Berliner.[8] In its most common form, the evaluation function is based on a weighted summation where the individual criteria are multiplied by a scaling factor and the results added to produce a composite score. The scaling factors, or weights, relay the relative importance of each criteria. Determining the form of an evaluation function for a particular application can be difficult since no set methodology exists for developing one. Although knowledge provided by human experts provides valuable guidance, it is rarely detailed enough for formulating an analytical solution. Humans tend to reason in relative rather than absolute terms. Consequently, whereas experts can generally relate which features are more or less important than others, the information is usually insufficient for determining the weights to be used in an evaluation function.

To overcome this problem, inductive learning was used. As a form of machine learning, it presents the computer with a large number of positive and negative examples that are used for training. The examples serve as input/output pairs from which the machine induces a function—in this case an evaluation function—relating the input to the output. The mechanism used was a neural network whose architecture was first proposed by Kohonen et al.[9] The learning algorithm of Ref. 9 differs significantly from the back propagation approach[10] more commonly seen in neural networks and depends on vector quantization to map an m-dimensional feature space into an n-dimensional classification space. Once properly trained, the weights used by the neural network to perform this mapping can be extracted for use in the evaluation function. The neural network can then be disposed of and does not appear in the final implementation of the expert system.

The general form of the evaluation function used by the expert system is

$$\mu_{\mathrm{GOOD}} = (1 + \mu_A - \mu_R)/2 \tag{1}$$

where μ_A and μ_R take the form

$$\mu_{A,R} = \left[\sum_{k=1}^{n}(f_k - w_k)^2\right]^{\frac{1}{2}} \tag{2}$$

The algorithm requires that values for the input features f_k reside in the range (0,1). The quantity μ_{GOOD} measures the goodness of each track, also taking on a value from (0, 1). It is then scaled from 0 to 100 to produce the ranking of each track; zero represents a track with a poor fit to the orbit under consideration whereas 100 represents a good fit. The functions μ_A and μ_R measure the Euclidean distance between a given feature vector and two prototypical vectors that define a perfectly acceptable UCT (μ_A) and a perfectly rejectable UCT (μ_R). As the value of μ_A approaches one and the value of μ_R approaches zero, the goodness of track improves and μ_{GOOD} reaches a limiting value of one. With the values of μ_A and μ_R moving toward the opposite extremes, μ_{GOOD} is minimized.

The form of Eq. (2) is dictated by the neural network. Whereas μ_A and μ_R have a mathematical basis in the theory of vector quan-

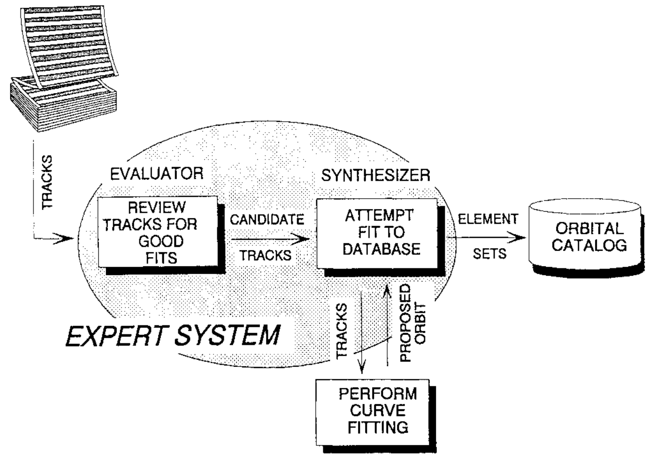

Fig. 5 Evaluator.

tization, μ_{GOOD} does not. Instead, Eq. (1) is better understood using fuzzy logic theory which views μ_{GOOD} as a membership function.[11] Membership functions measure the degree to which an instance is a member of a particular set and, unlike standard set operators which are strictly binary, vary continuously from non-membership through partial membership to full membership. In this sense, μ_{GOOD} computes the degree to which a UCT is a member of the set of good UCTs. Whereas fuzzy logic provides a framework for using membership functions, no mathematical method exists for determining the exact form a membership function should take. Consequently, μ_{GOOD} was found experimentally.

For the UCT problem, 6 of the 11 features identified in Table 3 are used by the evaluation function: p, $\Delta\bar{U}$, $\Delta\bar{V}$, $\Delta\bar{W}$, $\Delta W > 10$, and ID. With the exception of ID, all must be scaled to dimensionless quantities in the range (0,1) prior to use with the function. These six features are then mapped to a single classification given by μ_{GOOD}. Despite its simplicity, the evaluation function embodies a number of rules used by the orbital analysts to determine the worthiness of a UCT. When assessing data, the experts simultaneously examine several features and the contribution they make to a good candidate. The analysts favor tracks with a large number of observations and small residual errors. Of the residuals, ΔW represents cross-plane errors and receives particular attention. Because inclinations of most orbits are very stable, large values for ΔW are suspect and indicate that a UCT has not been associated with the correct object from the database. It is for this reason that the parameter $\Delta W > 10$ is also included in the UCT feature set since analysts will scan a listing specifically looking for double digit cross-track errors. Finally, a great deal of credence is placed in the ability of sensors to correctly tag a track. Consequently, a favorable attribute is for the satellite from the database that is currently being considered to be the same as the one originally given by the sensor in the tag field (i.e., ID is true). Through the proper selection of input features and weights, the knowledge of these rules and the compensatory nature between them is captured by the evaluation function.

Evaluation functions are limited in that they can only perform correlation and lack the capability of applying context sensitive information or performing any sort of logical reasoning. For example, a UCT may have all of the required attributes to receive a high ranking from the evaluation function. The expert may have a particular piece of information, however, that would eliminate an otherwise good UCT from consideration. To account for this information, the evaluation function used in this application is augmented by a set of rules that apply expert knowledge not available to the function. The layout is shown in Fig. 5. The evaluation function produces an initial ranking based on Eq. (1) which is then refined through the application of rules that make adjustments for specific situations. It is through these additional rules that allowances are made for drag. Orbital decay contributes to change in the orbital elements which leads to errors in predicting satellite motion. The analysts compensate for this by relaxing the requirements for low residual errors. The relationship, however, is neither monotonic nor linear (i.e., doubling

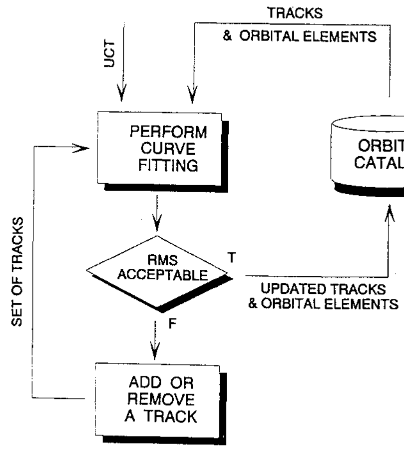

Fig. 6 Synthesizer algorithm.

the value of P does not always lead to a doubling in allowable values for $\Delta\bar{U}$, $\Delta\bar{V}$, and $\Delta\bar{W}$) and, therefore can not be accounted for in an evaluation function. Another factor considered is age, the epoch of a UCT relative to the epoch of an object's element set. If an element set has not been updated in several days, then greater leniency is permitted when judging a recent track as acceptable. In the opposite case, where the element set is current but a UCT is a few days old, different rules apply that specifically look at the features $\dot{P}$, p, and $\Delta\bar{U}$, $\Delta\bar{V}$, and $\Delta\bar{W}$. Additional rules consider incidental factors such as sensors that are temporarily producing poor data and thrustings of active payloads.

E. Synthesizer

Once a track is deemed good, the task of fitting it to the database remains. Using a least-mean-square algorithm, the effort centers on finding the orbit that most closely fits the original set of tracks plus the added UCT. Initially, the original element set for the object is used to propagate a curve. Based on the fit error between each observation and this curve, the expert may decide to eliminate some of the original tracks. This results in a new set of tracks that is used to determine a new element set and, by propagating another curve from this new element set, another set of fit errors is computed. The expert continues to refine the selection of observations by recalling discarded tracks or eliminating more tracks until a suitable orbit is found. The entire process is dynamic with the element set affecting the fit of the tracks and the selection of tracks affecting the element set. The solution is found by synthesis with the expert proposing alternate sets of tracks in an attempt to converge on an answer.

Several rules are used by the experts to reach a final combination of orbital elements and tracks. The overall goal is to minimize the rms error for an entire set of tracks while keeping changes in orbital parameters "reasonable." Although the experts use a variety of techniques, the primary method for arriving at an acceptable fit is to add and remove tracks from the data set until the tracks' root-mean-square error is minimized. Initially, the analyst will add the UCT, propagate the orbit, and observe the residual errors. If the root-mean-square error for the entire orbit is low—roughly defined as an rms in nautical miles less than 10% of the orbital period in minutes—and consistent errors exist across all of the tracks and all of the observations that make up the tracks, then a fit has been achieved. Lacking this, effort centers on minimizing the residual errors for the most recent tracks while allowing errors to increase for tracks with older epochs. This acknowledges that the orbit is dynamic and that the elements used to describe the orbit today will poorly fit the observations of a few days or weeks past. Tolerances for errors on older tracks increase with an increase in orbital decay rate and age of the tracks' epochs. The number and size of the tracks involved is also a factor in judging the quality of a fit. If a sufficient number of recent tracks are available, older tracks with excessive errors may simply be discarded and no longer used for orbit determination. However, there is a limit to the number of tracks an analyst will use to define an orbit. Only under unusual circumstances is an orbit built on three or fewer tracks.

Other rules are used when dealing with an outlier, a single track or observation with persistently high $\Delta\bar{U}$, $\Delta\bar{V}$, and $\Delta\bar{W}$ errors. A single observation is rarely removed from consideration unless it represents a one observation track. A single track may be removed if doing so leads to small, consistent errors across all remaining tracks. At this point the errant track would be considered a mis-association and, time permitting, attempts will be made to fit it to a different object in the catalog. Still other rules deal with unanticipated thrustings on payloads (noted by low errors across several older tracks followed by a sudden increase for more recent tracks) and orbits that produce positive rates of orbital decay (indicative of a poorly defined or rapidly changing orbit). While applying all these rules, changes in the orbital parameters are monitored to guide the analyst towards a credible solution: periods should decrease, not increase; inclinations should remain nearly constant; the rate of orbital decay should not change by more than an order of magnitude unless the orbit's period is extremely short (indicating the object may be approaching re-entry) or the orbit is poorly defined (indicated by high residual errors for all tracks attributed to an object).

When completed, the synthesizer routine will perform these functions for the expert system using the algorithm outlined in Fig. 6. After adding the new track, repeated calls are made to a curve fitting routine which generates a set of orbital elements. The calling loop is exited when the fit is satisfactory or no progress is being made. Until the exit criterion is met, adjustments are made to the set of tracks used to define the orbit. Although some judgement is required when deciding to terminate the loop, the majority of the synthesizer's knowledge lies in the rules that determine which tracks to include in the definition of an orbit.

IV. Results

A set of 1445 UCTs was gathered over a 2-day period. Of these, 230 were used to develop the evaluation function and rules used by the expert system while the remainder were used as a test set. The specific evaluation function produced by the neural network is Eq. (1) where

$$\mu_A = [(P' - 0.6411)^2 + ((\Delta W > 10)' - 1.0214)^2 + (\Delta\bar{U}' - 0.8364)^2 + (\Delta\bar{V}' - 0.7368)^2 + (\Delta\bar{W}' - 0.8866)^2 + (ID - 0.4603)^2]^{\frac{1}{2}} \tag{3}$$

$$\mu_R = [(P' - 0.3118)^2 + ((\Delta W > 10)' - 0.6962)^2 + (\Delta\bar{U}' - 0.6942)^2 + (\Delta\bar{V}' - 0.4871)^2 + (\Delta\bar{W}' - 0.4636)^2 + (ID - 0.1754)^2]^{\frac{1}{2}} \tag{4}$$

The prime notation indicates the feature set has been scaled to the range (0,1). The variable ID remains discrete. A complete listing of the test results and rules is available from Hecker.[12]

Generally, those tracks that were classified as acceptable by the experts received an evaluator ranking of 50% or greater. Therefore, when assessing the evaluator's performance, the following assumptions were made: the orbital analysts correctly classified all UCTs, an evaluator ranking of 50% or greater indicates the evaluator found the UCT acceptable, and an evaluator ranking of less than 50% indicates the evaluator found the UCT unacceptable. Using the experts' classification of UCTs as truth, the accuracy of the evaluator module was 91.4% using the evaluation function alone and 99.1% using the evaluation function in conjunction with the expert system rules. The experts and the expert system disagreed on nine tracks, all of which received evaluator rankings of approximately 50%. Although the code used was neither compiled nor optimized, near realtime speeds were achieved. Keeping the rules in their native CLIPS format and using the CLIPS inference engine as an interpreter, the 1445 test cases were processed in less than 15 min on a personal computer-based system.

Rules for the synthesizer have been formulated and the controlling algorithm has been defined. The task of integrating calls to the curve fitting routine, however, remains to be implemented. As such, test results for the synthesizer are still forthcoming.

V. Conclusions

As indicated by this research, a practical expert system for correlating UCTs is achievable. Further research is required to complete the synthesizer module, and a number of software engineering issues must be considered before a real-world system could be deployed. The initial success shown here, however, validates the application of artificial intelligence techniques to the problem of UCTs.

Acknowledgments

The guidance of Y. Kanayama, Department of Computer Science at the Naval Postgraduate School, and I. M. Ross, Department of Aeronautics and Astronautics at the Naval Postgraduate School, is greatly appreciated. In addition, Edna Jenkins, Senior Analyst of the Naval Space Surveillance Command, must be thanked for the knowledge she imparted to this project.

References

[1]Solomon, D., "The NAVSPASUR Satellite Motion Model," Naval Research Lab. Rept. N00014-87-C-2547, Washington, DC, Aug. 1991, pp. 1–22.

[2]Knowles, S., Melson, C., Jenkins, E., and Perini, D., "Uncorrelated Target (UCT)/Breakup Processing at NAVSPASUR," Naval Space Surveillance Center, Malchren, VA, Feb. 1990.

[3]Bate, R., Mueller, D., and White, J., *Fundamentals of Astrodynamics*, Dover, New York, 1971, Ch. 9.

[4]Jackson, P., *Introduction to Expert Systems*, 2nd ed., Addison–Wesley, New York, 1990, p. 3.

[5]Anon., "CLIPS Reference Manual," NASA Rept. JSC-22948, Jan. 1991.

[6]Mitri, M., "A Task-Specific Problem-Solving Architecture for Candidate Evaluation," *AI Magazine*, Vol. 12, Issue 3, p. 95.

[7]Samuel, A., "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal of Research and Development*, Issue 3, 1959, pp. 211–229.

[8]Berliner, H., "Experiences in Evaluation with BKG—A Program That Plays Backgammon," *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, MIT Press, Cambridge, MA, 1977, pp. 428–433.

[9]Kohonen, T., Barna, G., and Chrisley, R., "Statistical Pattern Recognition with Neural Networks: Benchmarking Studies," *Proceedings from the Inst. of Electrical and Electronic Engineers International Conference on Neural Networks*, Elsevier Scientific, NY, 1988, pp. 64–65.

[10]McClelland, J., and Rumelhart, D., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. I, MIT Press, Cambridge, MA, 1986, Ch. 10.

[11]Kosko, B., *Neural Networks and Fuzzy Systems*, Prentice–Hall, Englewood Cliffs, NJ, 1992, pp. 145–149.

[12]Hecker, M., "An Expert System for Processing Uncorrelated Satellite Tracks," Naval Postgraduate School, Monterey, CA, Dec. 1992.