Target Tracking Prediction Through Group Correlation Analysis

Brandon J. Bovey
Department of Systems Engineering
University of Virginia
Charlottesville, VA, U.S.A.
brandon_bovey@hotmail.com

Donald E. Brown
Department of Systems Engineering
University of Virginia
Charlottesville, VA, U.S.A.
brown@virginia.edu

Abstract - This paper presents a method for predicting target locations based upon the correlation of movements for targets traveling in groups. The algorithm is designed for incorporation into a modern tracking system as a supplemental module, providing updated estimates of target velocities based upon the results of a correlation analysis. The approach is presented for application in the realm of ground-based tracking systems, although the approach is general and can be applied in any domain where target movements are correlated.

Keywords: Multiple-target tracking, tracking prediction, group tracking, correlation analysis.

1 Introduction

Tracking systems are employed in a wide variety of applications. Advances have brought about effective tracking systems in domains ranging from air-traffic control to particle physics. The focus of our research is in the domain of military tracking systems, in particular ground-based military vehicles.

The objective of tracking systems is to maintain the progression of targets over time by receiving intermittent sensor reports on target states. More formally, the basic idea is to partition the sensor reports such that each partition corresponds to an individual target. Conventional tracking systems employ three major components to this end: the correlation module, the track maintenance and filtering module, and the prediction module [1][2]. The module of interest in our approach is the prediction module, the method by which the system is able to estimate future target states and thus properly associate future target detections with target tracks.

Tracking prediction research has been primarily focused on the tracking problem at the level of individual targets without group contextual information. Further, tracking systems have traditionally relied upon non-contextual motion models, such as the Kalman filter [3][4], for the prediction of future target states. This approach has been quite successful for applications in domains such as projectile tracking, where target movements are dictated almost entirely by Newtonian mechanics. However, ground-based vehicles react intelligently to non-homogenous surroundings (roads, water, obstacles, et cetera), so motion models are inadequate for this application. To account for the intelligent behavior of ground-based targets, research at the University of Virginia has yielded contextual prediction methods such as the Terrain-Based Tracker (TBT) [6]. The TBT prediction algorithm accounts for the influence of terrain in order to model the intelligence of vehicle operators. Tracking system performance can be further improved by utilizing additional sources of contextual information.

Group tracking has not been extensively researched as an aid for individual target prediction. Group tracking has been examined as a method for conserving computational resources and as a method for tracking closely spaced targets through use of group centroid techniques [1]. Further, the logistics of utilizing these group tracks in full tracking systems has been examined. For example, Yang et al researched methods of initializing group tracks [5]. Unfortunately, these methods do not provide a strong contextual framework to facilitate individual target prediction, although ground-based military vehicles typically move in groups and in accordance with general military tactics [7].

2 Problem Statement

We describe an original method for utilizing correlated movements among targets in groups to accurately predict the future states of each target. For example, a convoy traveling in a single file line exhibits a simple case of correlated movements. The movements of a vehicle at the tail of a convoy will be highly correlated to the movements of the vehicle leading the convoy. Other less obvious situations are also captured and utilized by the algorithm.

Our method is a supplemental module (which is in turn comprised of three components) in a modern tracking system. Please refer to Figure 1. Thus, the group context modules (Group Identification, Correlation of Movements, and Collective Influence of Group) can be used in conjunction with systems utilizing prediction algorithms ranging from the conventional Kalman Filter Tracker to
the context-based Terrain-Based Tracker. The versatility of the method is a direct result of its output. When applicable, the module updates the last observed velocity of a target to represent where the target is believed to be going based upon a correlation analysis. Velocity is a critical parameter in many prediction systems, so the group context modules work very effectively in conjunction with modern tracking systems.

3 Presented Method for Prediction

This section presents our method for utilizing the correlated movements of targets to increase prediction accuracy. The method can be divided into three distinct modules: group identification, correlation identification, and group correlation prediction. A tracking system utilizing this technique is similar to conventional systems, as the group correlation analysis modules are supplemental modules that do not modify the overall tracking framework.

3.1 Group Identification

If the movements of targets are correlated, the targets function as a coordinated group or team. Identifying the potential presence of groups is thus the first logical step in employing information of correlated movements. This is also important from a computational standpoint, as the clustering acts as a means of gating targets that will be considered to have correlated movements. Although the precise tactical nature of the group will vary depending upon the application and situation, correlations can still be effectively used without precise knowledge of a team’s function.

The application we discuss in this paper is for that of ground-based vehicles. In the context of military divisions of vehicles, one could define a group as a battalion or platoon of vehicles. Further, within each group the interactions could be defined based upon the both the types of vehicles in the group as well as their function or objective. For example, a platoon of tanks might be in a search and destroy mode. These pieces of information provide a strong description of how the vehicles will interact with one another as well as how they will react to their surroundings. Unfortunately, this type of information will typically not be known of enemy vehicles in combat situations. Groups need to be defined in a more convenient and general fashion.

Clustering techniques provide a method of identifying the potential presence of groups. Individual vehicles in groups will exhibit similarities among one another, both in location and velocity. Since clustering methods partition data into subsets of similar data, clustering is a natural fit. Potential clustering methods include hierarchical methods or partitional approach. In particular, single-link hierarchical clustering and partitional nearest neighbor clustering are the good, efficient methods for the identification of groups, as neither shape nor a defined center of the clusters are of relevance; only similarity in location and velocity of the targets are of interest. Nonetheless, while location is the clustering attribute, velocity is exploited in the subsequent correlation analysis.

At least two clustering methods are suitable for the identifications of groups, although the appropriateness of the techniques is dependent upon the dimensionality of the distance function. Single-link hierarchical clustering is in general a more robust method than nearest neighbor clustering, since the clusters formed by the algorithm are not dependent upon the ordering of the data. The robustness does come at a cost. Hierarchical techniques are \(O(n^3)\) algorithms, whereas partitional nearest neighbor clustering is an \(O(n^2)\) algorithm. The clustering technique must be selected according to the complexity of the problem. For example, if the clustering distance function incorporates information of only Euclidian distance, a nearest neighbor approach is most suitable. If the distance function accounts for several different attributes, on the other hand, the increased complexity and computational overhead of hierarchical techniques is justified. Finally, for situations in which more precise clustering is needed, we use density estimation techniques, such as kernel density estimation.

3.2 Identification of Correlated Movements

With potential groups identified, the next step in our approach is to determine which pairs of target tracks within the groups have in fact demonstrated correlated movements. A statistical method capable of objectively scoring the correlation is needed. The presented method utilizes the product moment correlation coefficient, \(r\). This statistical function scores the correlation between two variables. The sample correlation of \(n\) pairs of values, \((x_1, y_1), \ldots, (x_n, y_n)\), is:

\[
r = \frac{n \sum x_i y_i - (\sum x_i)(\sum y_i)}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}}
\]  

where \(x_i\) in our application is the observation at time \(i\) and \(r \in [-1, 1]\), with a value near 0 representing no correlation between the variables, near 1 representing a strong positive correlation, and near -1 representing a strong negative correlation [9].

There are some key assumptions associated with using the product moment correlation coefficient. Firstly, the product moment correlation coefficient measures the degree of linear relationship between the variables. Our correlation analysis, therefore, only considers linear relationships between the movements in target tracks. Secondly, the two variables under consideration should be measured on a ratio scale. Locations can be considered interval scale variables because they have an arbitrary zero
point. However, we can easily convert between location scales (e.g. Universal Transverse Mercator and Sate Plane). Thus we can perform the correlations on a common scale. Finally, the two variables must have a bivariate normal distribution. While this assumption is not always true for tracking data, the violation of this assumption does not dramatically affect the results. This is because we are using the correlation as a filter on further group tracking processes and not as a final measure of location accuracy. In particular, the increased prediction accuracy realized through the use of the proposed method shows how well we can do even when the assumption on our filter may be violated [10].

The value of $n$ in the calculation of $r$ requires further discussion for several reasons. The two tracks under consideration may have different track history lengths, and thus contain different numbers of observations describing the targets' past states. The tracks may also be missing observations or the observations in the tracks may be received in sensor scans with non-constant intervals. However, only consecutive observations received in a constant interval, $T$, can be used in the calculation of $r$. We define a function $\ell(track_i)$ to be used in the calculation of $n$. This function returns the number of sequential observations separated in time by interval $T$ in track$_i$, beginning at the current time. $t$. Thus, the value of $n$ for each correlation coefficient between track$_i$ and track$_j$ is calculated using a simple heuristic:

$$n = \min\{k, \ell(track_i), (\ell(track_j)\} \tag{2}$$

where the parameter $k$ is the maximum number of observations to compare.

In summary, if two target tracks are described by two variables, $x$ and $y$, at time $t$ the cross-correlation coefficient would be calculated for pairs $(x_t, y_t), \ldots, (x_{t+n}, y_{t+n})$.

Time lags must also be addressed when considering the correlated movements of vehicles. Our discussion up to this point has assumed a lag of 0, although it will not necessarily be the case that target tracks that are related are correlated at a time lag of 0. This will certainly be the case for vehicles traveling very closely and with relatively infrequent sensor scans. However, vehicles that have coordinated movements and are separated by some distance will be correlated at a time lag of with some positive integer greater than 0 with the leading vehicle. Figure 2 depicts a correlation at a lag greater than 0. In this figure, track 2 is correlated to track 1 at a lag of 1. This information allows for an accurate prediction of the state of target track 2 at $t=5$.

The computation of the correlation coefficient for track$_i$ (with descriptive variable $x$) and track$_k$ (with descriptive variable $y$) at a lag of $g$ involves the comparison of points $(x_t, y_t), (x_{t+g}, y_{t+g}), \ldots, (x_{t+ng}, y_{t+ng})$. Further, we must generalize the heuristic for the calculation of $n$ to account for lagged correlation computations:

$$n = \min\{k, \ell(track_i), (\ell(track_j) - g, 0\} \tag{3}$$

We have not specifically addressed which observation attributes are used in the correlation computations between two tracks. Each observation will not necessarily consist of only one value of interest. This state description can include many possible pieces of relevant information:

- Location: $x$ and $y$
- Velocity: $dx/dt$ and $dy/dt$
- Acceleration: $d^2x/dt^2$ and $d^2y/dt^2$

The analysis of correlated movements in our method requires only a comparison of velocity. In order to identify the correlation between two tracks in terms of velocity, the product moment correlation coefficient must be calculated twice, once for both $dx/dt$ and $dy/dt$. Nonetheless, it is desirable to obtain a single value that represents the correlation in movements between the tracks.

Our method to obtain a single value is to take the mean of the $r$ values for $vx$ and $vy$. This approach is useful, as a mean with an absolute value near unity signifies that both $vx$ and $vy$ are strongly correlated, whereas values near 0 indicate that both $vx$ and $vy$ are not both strongly correlated with the same sign.

We have presented the basic algorithm for the calculation of the mean correlation coefficient between track$_i$ and track$_k$. Since the lag of the strongest correlation between track$_i$ and track$_k$ is not initially known, the mean correlation coefficients for track$_i$ and track$_k$ ($\forall i,j, i \neq j, i,j \in [\text{group}_i]$) are computed and stored for lags between 0 and the maximum lag, $G$. For each group$_i$, the result is a conceptual three-dimensional matrix with dimensions, $s \times s \times (G+1)$, where $s$ is the number of targets in group$_i$.

Once the correlations have been computed, they can be utilized to better predict future locations of targets. The last observed or estimated velocity of the target is a critical parameter for many algorithms in the prediction of the next target location. Therefore, the correlation information is used to update the last observed or estimated target velocity.

### 3.3 Treatment of Correlations with Lag Greater than 0

Strong correlations with $g>0$ are the most predictive correlations that can be obtained from the correlation analysis. This type of correlation allows for the use of
past observations of one target to better predict the future state of another target.

For each track, we search the mean correlation coefficients of all tracks (\(i \neq j\)) and all \(g\) (0 < \(g\) < \(G\)) to find the greatest mean correlation coefficient, \(r_{ij,g}\). The correlation associated with track \(i\) and track \(j\) at a lag of \(g\) is deemed significant and utilized for prediction if \(r_{ij,g} > \tau_0\), the significance threshold for correlations with \(g=0\).

Let \(t\) be the time of the last sensor scan once a satisfactory correlation is found. The correlation suggests that the movement of \(track_j\), between time \(t\) and \(t+1\) will mirror that of \(track_i\), between \(t-g\) and \(t-g+1\). First, the observed straight-line velocity, with components \(v_x'\) and \(v_y'\), of \(track_j\), between time \(t-g\) and \(t-g+1\) is calculated:

\[
v_{x\_corr} = \alpha v_x + (1-\alpha)v_{x\_last}\]

\[
v_{y\_corr} = \alpha v_y + (1-\alpha)v_{y\_last}\]

where \(\alpha\) is the correlation weighting parameter. It should be noted that \(\alpha\) can be considered a function of the product moment correlation coefficient, \(\alpha(r)\), as the level of correlation dictates the confidence in the expected velocity. However, the threshold for determining significant correlations is typically set very high, so \(\alpha(r)\) is considered a constant, \(\alpha\), in this discussion.

### 3.4 Treatment of Correlations with Lag of 0

Targets often travel very closely to one another and have strongly correlated movements with \(g=0\). Utilizing this information in prediction is more difficult than the case for \(g>0\), as there is no historical basis for prediction. Nonetheless, the correlation provides us with useful information for prediction. The approach for \(g=0\) rests on the basis that the correlated group members collectively provide a stronger estimate of their true straight-line velocity than does each vehicle’s last observed velocity.

The targets in the group which have correlated movements with \(g=0\) must first be identified. The desired subsets of targets in the group are identified through partitional nearest neighbor clustering using a specialized distance function. The distance between track \(i\) and track \(j\) for the nearest neighbor clustering is:

\[d_{ij} = \begin{cases} \text{if } r_{ij,0} > \tau_0, 0 \text{, else } \infty \end{cases}\]

where \(\tau_0\) is the significance threshold for correlations with \(g=0\).

For each cluster of targets that is formed, a group velocity is calculated using an average:

\[
v_{x\_group} = \frac{\sum v_{x\_corr}}{N}\]

where \(N\) is the number of targets in the cluster.

Similar to the case for \(g>0\), the updated estimate of the velocity for \(track_i\) at time \(t\) using knowledge of the correlation is then:

\[
v_{x\_corr} = (\alpha)v_{x\_group} + (1-\alpha)v_{x\_last}\]

\[
v_{y\_corr} = (\alpha)v_{y\_group} + (1-\alpha)v_{y\_last}\]

In general, this updated velocity will be more accurate than the individual observed velocities because the group velocity offsets the noise among the correlated targets’ individual observations. This method differs from other group approaches that use the group velocity to update the velocities of individual targets due to the method by which the group is formed. Other approaches have used velocity and distance similarity criteria to form groups; our approach utilizes similarity in distance then a similarity in previous movements through the use of a correlation analysis.

### 4 Performance Evaluation

The approach we have presented lies within the logic of the prediction module. However, in order to identify the performance change associated with the modification of any module, the module must be placed within a full tracking system. The systems used for the performance evaluation can be described as simple multi-target tracking systems utilizing sequential nearest neighbor data association. More advanced multiple hypothesis methods are not considered.

Two systems have been implemented in C on the Solaris platform for the purpose of evaluating tracking system performance: a Terrain-Based Tracker and a simple Kalman Filter Tracker. Figure 3 is a screen shot of the software. Both systems utilize nearest neighbor techniques for correlation. Track maintenance and filtering is simply accomplished through creating new tracks for uncorrelated observations and deleting old tracks that have not been updated for a period of several scans. These straightforward techniques are justified by the accuracy of the test data; there are no errant
observations in the data nor are false detections introduced in testing.

4.1 Measures of Performance

Two metrics are used in evaluating the success of the tracking systems for the data sets: percentile score and percent correct correlation.

The percentile score metric is used for evaluating prediction accuracy. The metric can be applied for both Gaussian and non-Gaussian estimations since the probabilities for a target residing at locations on the terrain are discretized on a grid. Let $s_i^f$ represent the probability or likelihood of the target residing in the $i^{th}$ grid cell. Denote the true target location (or the center of the elliptic error probable) as $x_{true}^f$ on the $N$ cell grid. The percentile score, $p$, is then:

$$ p = \frac{\sum_{i=1}^{N} s_i^f \{ \{ x_i^f \leq x_{true}^f \} \}}{\sum_{i=1}^{N} s_i^f} \quad (12) $$

where $\{ x_i^f \leq x_{true}^f \}$ is a Boolean expression [11]. Consistent with standard percentile uses, scores are in the range [0,1] (or [0%, 100%]), with increasing scores representing increasing levels of success. This score provides a good measure of prediction accuracy if the grid used for the discretization is sufficiently fine and covers the entire region of interest.

Percent correct correlation provides a metric for overall tracking system accuracy. The score is calculated by examining the percent of observation-to-track fusions for each track that correspond to the target of the immediately preceding observation-to-track fusion. More formally, denote $R_{ij}$ as the observation fused to track $j$ at time $t$, and denote $O_{ti}$ as the observation produced by target $i$ at time $t$. If target $i$ produces a series of detections over time, $\{O_{t1}, O_{t2}, \ldots, O_{tn}\}$, $n-1$ correlation decisions will be scored for target $i$. The percent correct correlation score for just the observations produced by target $i$ is then:

$$ PCC = \frac{\sum_{t=2}^{n} R_{ij} == R_{j,i-1}}{n-1} \quad (13) $$

where $\{R_{ij} == R_{j,i-1}\}$ is a Boolean expression. The correlation process is viewed as a binary process in this context, as the system either correlates correctly or does not each time correlation is performed.

It should be noted that we tuned the parameters for the Kalman filter and the correlation analysis modules using simulated annealing.

4.2 Experimental Design

Tracking systems are complex algorithms that operate under a variety of circumstances. Thus, various tracking scenarios must be investigated when identifying the overall performance of a system. Non-robust solutions may perform extremely under very specific situations. However, the experimentation in our research evaluates the accuracy and robustness of the proposed supplemental prediction method. Several variables representing “scenario parameters” are varied to this end:

- Use of the correlation analysis modules, {True, False}
- Dataset, {1,2,3,4}
- Prediction method, {TBT, Kalman filtering}
- Sensor update interval, {5 minutes, 10 minutes}
- Circular error probable measurement error standard deviation, {25 m, 50 m, 100 m}

The datasets used in the evaluations describe ground-based vehicles during training exercises at the National Training Center. These datasets represent different situations:

- A large collection of targets which do not seem to act in a coordinated manner (Dataset 1)
- A small collection of targets which appear to function in some coordinated manner for some period of time (Dataset 2)
- A large, dense collection of targets which appear to function in a coordinated manner for some period of time (Dataset 3)
- A large collection of targets, many of which may appear to be functioning in a correlated manner, some of which are singletons (Dataset 4)

The results of the tests are not deterministic due to the randomness induced by the measurement error. Fortunately, the variance for a given scenario between runs is not extremely large, so each scenario is sampled only three times. The results presented later in the paper, in particular the p-values, validate that this number of samples is adequate for the process under examination.

The scenario parameters result in a number of runs: all permutations of the datasets (4), use of the correlation analysis modules (2), prediction intervals (2), sensor update intervals (2), and measurement sigmas (3) are run multiple times (3) for a total of 288 samples of each metric, percentile score and percent correct correlation.

5 Results

Linear regression provides a means of accounting for all of the scenario parameters and runs in an effort to determine the significance of the correlation analysis modules. The scenario parameters are the predictor variables in the model. Additionally, there are two different predictor variables and thus models, as there are two distinct measures of performance.

The models can be used to evaluate the significance of the correlation analysis modules by examining the coefficient of the associated categorical variable with regards to (1) the value, both sign and magnitude, of the coefficient and (2) the p-value for the t-test of the coefficient. So two linear regression models are developed using least squares estimation:

$$ y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \beta_7 x_7 + \varepsilon $$

where:

- $y$ = mean percentile score or percent correct correlation
- $x_1$ = prediction type; categorical variable: 0 = Kalman filter, 1 = TBT
- $x_2$ = use of correlation analysis modules; categorical variable: 0 = correlation analysis modules not used, 1 = correlation analysis modules used
\( x_3 \) = scan interval; continuous variable, in seconds
\( x_4 \) = measurement error sigma; continuous variable, in meters
\( x_5 \) = file type; categorical variable: 1 = Dataset 2, 0 = Otherwise
\( x_6 \) = file type; categorical variable: 1 = Dataset 3, 0 = Otherwise
\( x_7 \) = file type; categorical variable: 1 = Dataset 4, 0 = Otherwise
\( \varepsilon \) = normally distributed zero mean error term

where all of the regular assumptions of linear regression hold true.

It should be noted that percentile score, one of the response variables, is a proportion. It is often the case that response variables of this nature are best modeled logistically. However, conventional linear regression for this application is justified, as the model is not constructed for the purpose of predicting outside the range of the training data; the model is built as means of identifying the relationship between the scenario parameters and the measures of performance.

### 5.1 Percentile Score Model

The linear regression model with a response variable of percentile score yields the parameters found in Table 1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>p-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta_0 )</td>
<td>0.814</td>
<td>0.000</td>
</tr>
<tr>
<td>( \beta_1 )</td>
<td>-0.359</td>
<td>0.000</td>
</tr>
<tr>
<td>( \beta_2 )</td>
<td>0.0326</td>
<td>0.001</td>
</tr>
<tr>
<td>( \beta_3 )</td>
<td>-0.000261</td>
<td>0.000</td>
</tr>
<tr>
<td>( \beta_4 )</td>
<td>0.000393</td>
<td>0.010</td>
</tr>
<tr>
<td>( \beta_5 )</td>
<td>0.108</td>
<td>0.000</td>
</tr>
<tr>
<td>( \beta_6 )</td>
<td>0.0953</td>
<td>0.000</td>
</tr>
<tr>
<td>( \beta_7 )</td>
<td>0.156</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Table 1: Percentile Score Model Parameters

All of the parameters, \( \beta_i \)'s, have extremely low p-values for the t-tests, indicating that there is strong evidence against the null hypothesis, \( H_0: \beta_i=0 \). In particular, the coefficient for use of the correlation analysis modules, \( \beta_2 \), has a p-value of 0.001. For this parameter there is very strong evidence in favor of the alternative hypothesis, \( H_a: \beta_2>0 \).

Additionally, the model scores exceedingly well for several different metrics. Please refer to Table 2 for these results.

<table>
<thead>
<tr>
<th>Model Metric</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-squared</td>
<td>85.5%</td>
</tr>
<tr>
<td>( R\text{-squared}_{adj} )</td>
<td>85.2%</td>
</tr>
<tr>
<td>( F )</td>
<td>235.90</td>
</tr>
<tr>
<td>F p-value</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Table 2: Percentile Score Model Metrics

The strong model scores indicate that the parameter values associated with the scenario parameters can be used for conclusions about tracking system performance.

### 5.2 Percent Correct Correlation Model

The linear regression model with a response variable of percent correct correlation yields the parameters found in Table 3.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>p-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta_0 )</td>
<td>0.922</td>
<td>0.000</td>
</tr>
<tr>
<td>( \beta_1 )</td>
<td>0.0363</td>
<td>0.000</td>
</tr>
<tr>
<td>( \beta_2 )</td>
<td>0.321</td>
<td>0.000</td>
</tr>
<tr>
<td>( \beta_3 )</td>
<td>-0.000444</td>
<td>0.000</td>
</tr>
<tr>
<td>( \beta_4 )</td>
<td>-0.000269</td>
<td>0.000</td>
</tr>
<tr>
<td>( \beta_5 )</td>
<td>-0.316</td>
<td>0.000</td>
</tr>
<tr>
<td>( \beta_6 )</td>
<td>-0.344</td>
<td>0.000</td>
</tr>
<tr>
<td>( \beta_7 )</td>
<td>-0.386</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Table 3: Percent Correct Correlation Model Parameters

Again, all of the \( \beta_i \)'s have extremely low p-values for the t-tests. In particular, the coefficient for use of the correlation analysis modules, \( \beta_2 \), has a p-value of zero to three significant digits. For this parameter there is again very strong evidence in favor of the alternative hypothesis, \( H_a: \beta_2>0 \).

As Table 4 shows, the model with percent correct correlation scores slightly better than did the percentile score model as regards several metrics.

<table>
<thead>
<tr>
<th>Model Metric</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-squared</td>
<td>91.0%</td>
</tr>
<tr>
<td>( R\text{-squared}_{adj} )</td>
<td>90.8%</td>
</tr>
<tr>
<td>( F )</td>
<td>404.22</td>
</tr>
<tr>
<td>F p-value</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Table 4: Percent Correct Correlation Model Metrics

Once more, the sound model scores show that the parameter values associated with the scenario parameters can be used for conclusions about system performance.

The correlation analysis modules present additional computational expense. The thesis examines the overhead both theoretically and experimentally.

### 5.3 Results Interpretation

The models for both metrics show that the parameter associated with the use of the correlation analysis modules is extremely significant. Additionally, the parameter in both models is positive, denoting that the correlation modules positively affect overall system performance. In particular, percentile score increases an additive 3.26% and the score for percent correct correlation increases an additive 3.63% when using the group correlation analysis modules. Upon first inspection these values are rather humble. It must be reemphasized that the group correlation analysis modules only activate for prediction when the level of correlation between targets exceeds the significance threshold. This thresholding resulted in only 1 – 20% of the predictions in any dataset utilizing the group contextual information. Given this, the modest gain as averaged over all predictions becomes rather large when considering individual cases.
5.4 Implementation Issues

The theoretical algorithmic expense of the modules is examined in terms of an asymptotic analysis:

\[ g(n) \text{ is said to be } O(f(n)) \]

if \( \exists \text{ constants } C_0 \text{ and } N_0 \rightarrow g(n) < C_0 f(n) \forall n > N_0 \)

where \( n \) is the number of targets in the data in this application.

The overall efficiency of the correlation modules is dictated by the contribution of each module: the group identification module, \( g_1(n) \), is \( O(n^2) \), the correlation identification module, \( g_2(n) \), is worst case \( O(n^2) \), and the correlation prediction module, \( g_3(n) \), is \( O(n) \). The overall effect of the correlation modules, \( g(n) \), is then:

\[ g(n) = C_1 f_1(n) + C_2 f_2(n) + C_3 f_3(n) < C_0 f(n) \]

Substituting in the values yields:

\[ g(n) = C_1 n^2 + C_2 n^2 + C_3 n < C_0 n^2 \]

So we say that for large \( n \), \( n^2 \) is the upper limit on the correlation analysis modules.

Further, the effect of the correlation modules in terms of run time is examined in testing. The tests revealed that the correlation modules increase CPU processing time between 1% and 9%. This increase represents a nominal increase in expense relative to the performance gain realized through use of the modules. The tests were run on a dual CPU Sun Ultra II.

6 Summary and Conclusions

Our research has defined a method for identifying and utilizing correlated target movements. The generic nature of the result of the correlation analysis modules, an updated velocity reflecting correlation information, allows the modules to be utilized in various tracking systems. The method has been shown to be effective in both Kalman filter systems and TBT systems. Further, the correlation analysis modules have been shown to increase tracking system performance in terms of both mean percentile score and percent correct correlation. This conclusion is drawn from interpreting the linear regression models derived from the results of experimentation. The parameters of the models show that the use of the correlation analysis modules affects tracking performance in a positive manner at extremely high levels of significance. Additionally, the performance increase brought about through use of the correlation analysis modules comes with relatively light computational overhead.

Future work resides in improving upon the basic approach outlined in this paper. First, an output from the correlation analysis modules other than velocity should be considered for prediction systems that do not rely exclusively on motion models. Second, non-linear correlated movements is an area that warrants further research. Finally, correlated movements between targets of opposing allegiance should be examined.

7 References