Aggregation of Composition States for Markov Estimation in Level 2 Fusion

Stephen C. STUBBERUD
Rockwell Collins
Poway, CA 92064, USA

and

Kathleen A. KRAMER
University of San Diego
San Diego, CA 92110, USA

ABSTRACT

In sensor fusion, the use of composition information can help define and understand relationships between targets. This process, part of the Situational Assessment problem, also referred to as Level 2 fusion, can be quite complex when using standard classification approaches such as the Bayesian taxonomy. Determination of the number and type of elements that comprise a group can vary from report to report based on the type of sensors, the environment, and the behavior of the group. Estimation of group composition that can take these factors into account has been developed using a Markov chain approach. If the number of potential target classes is significant and the various standard group compositions are numerous, the computational complexity becomes unmanageable. This effort investigates a useful and computationally attainable Level 2 composition state estimate based upon the use of state aggregation.

Keywords: Situational Assessment, Level 2 Fusion, Markov Chain Approach, Group Composition, Sensor Data Fusion

1. INTRODUCTION

Situational Assessment, also referred to as Level 2 fusion, has numerous definitions to the fusion community. This work uses the Joint Directors Lab (JDL) definition [1] that it is the development and interpretation of relationships between objects. Knowledge of relationships between objects can determine which objects are working in coordination as a single unit, which objects are supporting the efforts of other objects either directly or indirectly, and which objects are having minimal or no relationships with other objects. To develop Level 2 objects and determine these relationships, the individual Level 1 objects or targets must be compared, combined, and interpreted.

State vector representation is a straightforward technique to develop a Level 2 object description. The states are the individual elements of the state vector. They can take on a variety of representations that are both quantitative and qualitative. In this effort, a battlespace problem is examined. First presented in [2], the proposed state representation of the overall Level 2 object is

$$\mathbf{x} = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} group\_kinematics \\ composition \\ formation \\ group\_extent \end{bmatrix} \tag{1}$$

where

$$\mathbf{a} = \begin{bmatrix} x_{group} \\ \dot{x}_{group} \\ y_{group} \\ \dot{y}_{group} \end{bmatrix}, \mathbf{b} = \begin{bmatrix} \#-class_1 \\ \#-class_2 \\ \vdots \\ \#-class_n \end{bmatrix},$$

$$\mathbf{c} = \begin{bmatrix} formation \\ formation\_change \end{bmatrix},$$

$$\text{and } \mathbf{d} = \begin{bmatrix} extent\_\text{footprint} \\ extent\_\text{region\_of\_influence} \end{bmatrix}.$$

Each of the sub-vectors that comprise the overall state vector may utilize different representation schemes. The implementation scheme of Figure 1, often referred to as the Bowman model, has been proposed for higher levels of fusion in both [2] and [3] and uses a different prediction and update approach for each representation. The prediction and update of the composition component of the Level 2 state vector is considered in this work for the case where the measurements or Level 1 entities use a Bayesian taxonomy [4] to represent uncertainty.

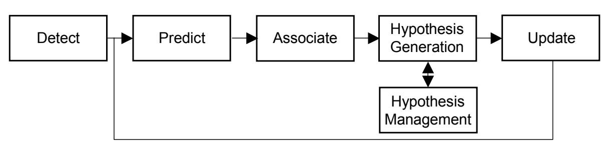

Figure 1: Level 2 Architecture based on the Bowman model.

In [5], [6], and [7], the composition sub-vector was designed to work in conjunction with a Markov chain. This implementation requires that each state element represents a possible composition. The value of each state element is then a probability of that defined composition. The sum total of the values of the entire sub-vector is then constrained to equal one, and all elements must be nonnegative. In [8], estimation of the composition was considered where the number of targets was known but the actual class contained uncertainty. Using the Bayesian taxonomy to provide the measurements for each target class, this implies that the class for each target could take n values where n indicates the number of possible classes. This implies that there are $n^{num\_targets}$ different permutations and a related number of combinations. Evaluation of the probabilities clearly can cause the computational complexity to grow quickly. The concept of restricting the combinations by the limiting the various compositions was considered in [8]. While this reduced the computations, it still was an exceedingly complex problem to implement.

In this work, the use of aggregation of states is proposed as a means of estimation of composition. Using such an approach, similar classification states, such as tanks and armored personnel carriers (APCs), would be mapped into aggregate states, such as armored vehicles. The algorithm also includes the reduction of the possible combinations using the aforementioned composition restrictions. In Figure 2, a battle group of three APCs and three tanks is shown. The aggregated group is simply an armored battle group of six elements. The aggregate does not concern itself with misidentified tanks that are assumed to be APCs. Thus, there is just one group-permutation instead of having six different ones (six tanks, five tanks/1APC, 4 tanks/2 APCs, etc.). Oftentimes, the information of interest in assessing the situation is the aggregate group, rather than information on individual targets within the group.

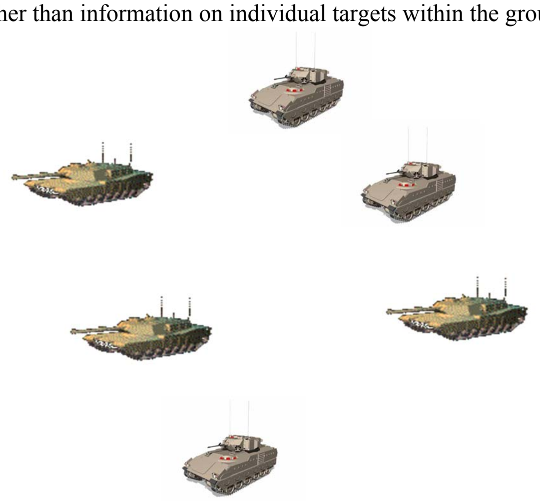

Figure 2: Armored battle group with six elements

In Section 2, the development of the general state representation and the aggregated state representation is presented. This is followed in Section 3 with an overview of the Markov chain as applied to the Level 2 battlespace problem. Section 4 contains the results of an initial test case.

2. STATE DEFINITIONS FOR COMPOSITION

This effort discusses a Markov chain approach to estimation for the Level 2 fusion problem when composition of a group of targets is considered. The current assumption is that the number of targets is known accurately while the classification of the elements of the group is unknown. One of the standard individual classification techniques is that of a Bayesian taxonomy. With this approach, a target has a number of possible classes that can exist in the battlespace as seen in Figure 3. The top node is referred to as the root node and is always assigned a probability of 1.0. At each level, a more granular description of the target is developed until at the lowest level of nodes, referred to as the leaf nodes, contains the most individualized description of the target that is desired. Here, the leaf nodes are defined as the individual target types. A set of nodes that branch off from a higher node are referred to as the children of the higher node. Similarly, the higher node is considered the parent node of the children nodes. Thus, *Truck* and *Recon* are the children nodes of the node *Light*. The root node, *Target,* is the parent node of *Armor* and *Light.* In this application, the probability of a parent node is equal to the sum of the probabilities of their children.

Based on *a priori* information, the nodes at a specific level are set. Usually, for ease, it is the leaf nodes that have their probabilities defined first with the scores then propagated up through the tree. New information is provided by reports from sensor systems. In a Bayesian taxonomy, each new set of data is "injected" into the classification as a set of probabilities for each potential target at a specific node as in Figure 3.

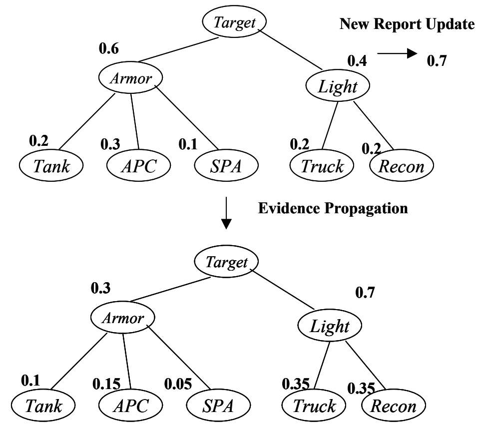

Figure 3: A Bayesian taxonomy can represent the probable classification of the target

For a given sensor, each potential class has an associated probability which is derived from the information gathered about features of the target. Many of the potential classes have a probability of zero, but others have nonzero probabilities. In Table 1, a set of probabilities is defined for each target class for a given sensor in a sensor field. These conditional probabilities are defined as the probabilities of what class is reported by the sensor given the target is of a specific class. For example, row one of Table 1 is for the class *Tank*. The probabilities of that row define, given that a tank exists, there is a 0.45 probability that the sensor system will classify the target as a tank while there is a 0.05 probability that the sensor system will define it as a truck.

Table 1. Example probabilities based on a sensor's abilities to identify target class given a specific target type is present.

Actual \ Detected Class	Tank	APC	Truck	Recon	TEL
Tank	.45	.45	.05	.02	.03
APC	.45	.45	.05	.02	.03
Truck	0	0	.90	.10	0
Recon	0	.05	.05	.85	.05
TEL	.10	0	0	.10	.80

An example of how a set of reports from different sensors would work in a Level 1 fusion classification system is as follows. The initial leaf nodes for a set of targets as defined in Table 1 are set as probabilities 0.2 for each target since there are five target classes. If the sensor of Table 1 reports a tank, then by Bayes theorem

$$\Pr(A \mid B) = \frac{\Pr(B \mid A)\Pr(A)}{\Pr(B)},$$

the probability becomes 0.45 for a tank, 0.45 for and APC, and 0.10 for a TEL. As this senor continuously reports a tank, the probability trends towards 0.5 for a tank and 0.5 for an APC while the TEL slowly trends towards a probability of zero. If another sensor with better discrimination between a tank and APC is used, the probability of a tank will vary towards the tank as the class with the highest probability repeatedly reported becomes the dominant class. Classes with probabilities of zero, unless the algorithm is modified, will go to zero completely and be unable to vary from this attraction point.

Depending on the number of target classes, the classification vector can become quite large. This can even occur with a relatively small number of target classes. To overcome this computational bottleneck, aggregation of the classes based on the group compositions is proposed. While aggregation loses some of the finer understanding of the individual elements, the overall benefit is the quick dissemination of information to those in the decision process. The battlefield groups for five element are defined in Table 2. In Table 3, the aggregation of the groups and the individual elements are presented. The concept is similar to using the middle set of nodes, *armor* and *light,* in Figure 3 as opposed to the five leaf nodes of individual target types.

Table 2: The composition of individual groups are based on standard military doctrine and are not an exhaustive list of combinations

Groups	Num Tanks	Num APCs	Num Trucks	Num TELs	Num Recon
Armored	6	0	0	0	0
Combined Arms 1	3	3	0	0	0
Combined Arms 2	2	4	0	0	0
Mech. Infantry	1	4	0	0	1
Heavy Infantry	0	5	0	0	1
Light Infantry	0	1	4	0	1
Supply	0	0	5	0	1
TEL 1	0	0	3	1	2
TEL 2	0	1	3	1	1

Table 3: The aggregated list delineates different groups based on their primary missions and capabilities.

Groups	Armored	Support	TEL
Heavy Armored	6	0	0
Heavy Infantry	5	1	0
Light Infantry/Supply	1/0	5/6	0
TEL	1/0	4/5	1

The aggregated individual elements clearly reduce the combinatorics of the problem. The aggregated groups still have interactions as seen in the first two and last two aggregated groups, but it is less so that the with those defined in Table 2. These aggregated groups define the state that is the output for the update component of the estimation process. It is also the input to the prediction component.

The predicted state is not comprised of group compositions but a number of individual elements of a specific aggregated class that will be provided by the classification sensor system with the next report.

3. THE MARKOV CHAIN

The Markov chain [9,10] provides the estimation that is core to this development. Although the concept and the implementation of the Markov chain are straightforward, the complexity of the model and power of its application lie in the development of the transition probabilities.

Basic Implementation

The Level 2 implementation from [2] has two major components: update and prediction. In Level 1 implementations, these components are often combined to be part of a state estimation routine, typically the Kalman filter. Such techniques combine both of these model components into a single algorithm. For Level 2 problems, the estimation routine is not easily coupled to a single algorithm, and estimation routines, such as a Kalman filter, require very complex mathematical models to handle the system dynamics and the measurement reports.

The Markov chain can be used in the two steps of estimation for the Level 2 problem with greater ease than commonly used estimation approaches. One Markov chain model can be develop to predict measurement reports for association; another can be developed to incorporate the new measurements into the state vector. Also, as opposed to the dynamic model

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k \qquad (2)$$

a time-step probability model [9] is developed

$$\mathbf{x}_{k+1} = \mathbf{T}_{k+1|k}\mathbf{x}_k = \left(\prod_{l=i}^{k+1} \mathbf{T}_{l|l-1}\right)\mathbf{x}_i = \begin{bmatrix} pr_{state_1\_to\_state_1} & pr_{state_2\_to\_state_1} & \cdots & pr_{state_n\_to\_state_1} \\ pr_{state_1\_to\_state_2} & pr_{state_2\_to\_state_2} & \cdots & pr_{state_n\_to\_state_2} \\ \vdots & \vdots & \ddots & \vdots \\ pr_{state_1\_to\_state_n} & pr_{state_2\_to\_state_n} & \cdots & pr_{state_n\_to\_state_n} \end{bmatrix}\mathbf{x}_i \qquad (3)$$

The state in Eq. (3) represents the complete set of possible compositions for the groups. Two important factors in the implementation are the definition of the state elements and the generation of transition probabilities.

While, often, the state vector is the same dimension on both sides of the equation, this need not always be the case. The estimated vector on the left-hand-side of the equation can be a composite or expansion of the state vector on the right-hand-side of the equation. The state vector on the right-hand-side of the equation is usually generated using the most recent measurement/report information.

State Aggregation

The concept of the composition component of the Level 2 state in Eq. (1) can take numerous forms. The state can contain all possible numbers of each class in the group. For the Markov chain state vector, this requires that the state include a state element for 0 to n, where n indicates the number of targets determined to be in the group. Another state representation that can be used by the Markov chain would have each individual target class with a possible number of each class being available as in Eq. (4):

$$\begin{bmatrix} 0-Class_1 \\ 1-Class_1 \\ \vdots \\ n-Class_1 \\ 0-Class_2 \\ \vdots \\ n-Class_2 \\ \vdots \\ 0-Class_m \\ \vdots \\ n-Class_m \end{bmatrix} \tag{4}$$

Another possible representation would be similar to that of the column labels in Table 2. The state vector could be the various possible composition groups. An additional group known as *other* could be incorporated to identify unknown or misidentified groups.

In [8], the application of the state representation as discussed with Eq.(4) was used for the prediction estimate while the group representation for the state vector was used by the update estimate. Even with an average number of target classes (*e.g.,* six), a handful of groups (*e.g.,* ten), and just six targets, the problem became computationally unwieldy.

A change in the aggregation scheme was considered. The first step in this effort was to aggregate the classes and the groups. As seen in Table 3, the aggregated groups were taken from nine to four. The individual classes were also reduced from five to three. This aggregation in states first reduced the number of states and the number of transition probabilities. More importantly, many of the outlying or other classes due to target misclassifications instantly became part of existing groups. For example, the group *Heavy Armored* now contains many more permutations of Tanks and APCs . This can be seen with many of the groups.

While these two new aggregation approaches provided an appreciable reduction in computations, the complexity was still significant, particularly in the prediction where the input state vector is the aggregated group state representation and the output is an aggregated composition vector as seen in Eq. (4).

The next component of the aggregation operated on the number of targets of a specific class. Instead of defining an individual number of targets for each state, the state elements represented a range of targets of a given class. Thus, the state representation in the prediction was aggregated as

$$\begin{bmatrix} 0-Class_1 \\ 1-4-Class_1 \\ \vdots \\ n-(n-r)-Class_1 \\ 0-Class_2 \\ \vdots \\ n-(n-k)-Class_2 \\ \vdots \\ 0-Class_m \\ \vdots \\ n-(n-r)-Class_m \end{bmatrix} \tag{5}$$

This second level of aggregation provided the greatest reduction in computational complexity. Also, the development of the transition probabilities could become less rigorous as grosser estimates can be used without loss of the general accuracy.

4. EXAMPLE OF AGGREGATION IN COMPOSITION

In Figure 2, a combined arms force of three tanks and three APCs is shown. In Figure 4, a report of two tanks, two APCs, and two trucks is given for that group. In fact, a tank and APC have been misidentified as each other. With the aggregation of the groups and the targets, the process of estimating becomes significantly simpler than with the full state representation as done in [8]. If the tank and APC were considered different classes, then the process would need to consider the error in the individual reports of these similar vehicles.

To demonstrate the aggregated composition approach, a six target example is provided. The actual group is the non-aggregated group (Table 3) *Heavy Infantry*. Without aggregation, the most probable result would be that of *Other* as an APC would most likely be misrepresented as a tank and no unit with a tank has a Recon unit. The groups named *Combined Arms* would follow in probability.

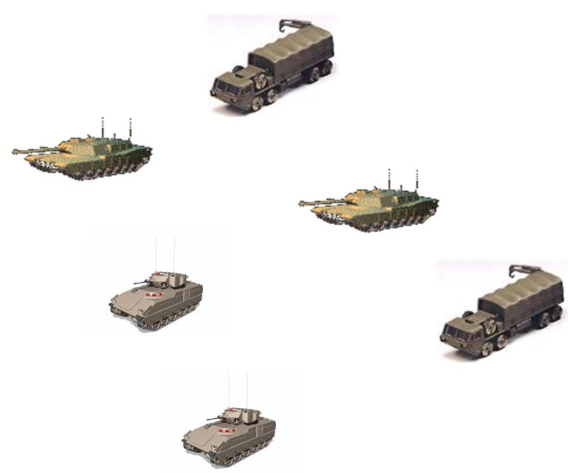

Figure 4: The reported group has two misidentifications as trucks. Also, a tank and APC have been switched.

The probabilities associated with the classification of each target are given in Table 1. These probabilities indicate the probability that given a specific target (the row) the system will assume that the target is a given class (the column). The aggregation of these classes results in the probabilities listed in Table 4.

Table 4. State transition probabilities for aggregate states.

Actual\ Detected Class	Armored	Support	TEL
Tank	.90	.07	.03
APC	.90	.07	.03
Truck	0	1.00	0
Recon	.05	.90	.05
TEL	.10	.10	.80

An aggregated score set of probabilities can be generated using the estimated number of units (*i.e.,* APCs and Tanks) in a specific aggregated class (*i.e., Armored*) and weight the different elements' probabilities appropriately using Bayes Theorem.

Using the probabilities in Table 4, a set of transition probabilities for the prediction state that estimates the next report on the individual targets classifications is generated. In Table 5, the actual probabilities for the *Heavy Armored* aggregated group are generated. From these values, one can see that the transition probabilities can be approximated using bin values such as *low, medium, high, etc.* In Table 6, the applied transition probabilities are defined. A similar set of probabilities for the update problem are defined in Table 7. With the update transition, the resulting vector requires normalization after being computed.

Table 5. Development of the actual transition probabilities for *Heavy Armored*

Groups	Armored tanks/ APCs	Support Truck/ recon	TEL
Heavy Armored Pright=0.53 P1err = 0.354 P2err=0.098 P3+err = 0.016	6	0	0
Heavy Infantry $P2{+}err_{armor}$=.114 $P3{+}err_{armor}$ = 0.016	5	1	0
Light Infantry/Supply $Prob2{+}err_{recon}$=0.033 $Prob1errto_{TEL}$=.23	1/0	5/6	0
TEL	1/0	4/5	1

Three reported classifications for each target in the group are shown in Table 8. With the first report, the classes are mapped to their aggregates as shown in the table, and the update is applied. Then a prediction is generated. The process is repeated for each new report. The update progression is shown in Table 9. A similar set of predictive estimates for each report is presented in Table 10.

In implementation of an actual Level 2 system, the prediction would be used to associate the Level 2 track with the new report. Such information can be used to look for targets that have split off from the group or to detect targets that are traveling too close to each other to be separated.

Table 6. Heuristically developed prediction transition matrix.

Group\ Reported - Predicted	Armored			Support			TEL		
	4-6	1-3	0	4-6	1-3	0	2+	1	0
Heavy Armored	.95	.05	0	0	.5	.5	0	.2	.8
Heavy Infantry	.8	.2	0	0	.5	.5	0	.2	.8
Light Infantry/ Supply	0	.5	.5	.9	.1	0	0	.2	.8
TEL	0	.5	.5	.9	.1	0	.1	.8	.1

Table 7: Heuristically derived update transition matrix

Group\ Reported - Predicted	Armored			Support			TEL		
	4-6	1-3	0	4-6	1-3	0	2+	1	0
Heavy Armored	.9	.1	0	0	.1	.9	0	.05	.9
Heavy Infantry	.9	.5	0	0	.9	.1	0	.05	.9
Light Infantry/ Supply	.05	.8	.8	.9	.5	.05	0	.05	.9
TEL	0	.5	.8	.8	.5	0	.9	.8	.1

Table 8: Three Bayesian reports on the classification of the six targets in the group

Report	Tgt. Class	Tgt. 1	Tgt. 2	Tgt. 3	Tgt. 4	Tgt. 5	Tgt. 6
#1	Armored	0.5	0.5	0.75	0.8	0.3	0.45
	Support	0.4	0.4	0.25	0.2	0.7	0.55
	TEL	0.1	0.1	0	0	0	0
#2	Armored	0.63	0.68	0.82	0.8	0.19	0.38
	Support	0.35	0.31	0.18	0.2	0.81	0.62
	TEL	0.02	0.01	0	0	0	0
#3	Armored	0.62	0.78	0.86	0.8	0.11	0.42
	Support	0.37	0.22	0.14	0.2	0.89	0.58
	TEL	0.01	0	0	0	0	0

Table 9: The resulting update probabilities for compositions.

Aggregated Group	With Report 1	With Report 2	With Report 3
Heavy Armored	0.2066	0.2266	0.2322
Heavy Infantry	0.3394	0.3535	0.3604
Light Infantry/ Supply	0.2842	0.2819	0.2764
TEL	0.1698	0.1380	0.1309

Table 10: The resulting prediction probabilities for measurements

Aggregated Group	Predicted Report 2	Predicted Report 3	Predicted Report 4
4-6 Armored	0.4678	0.4981	0.5089
1-3 Armored	0.3052	0.2920	0.2874
0 Armored	0.2270	0.2099	0.2037
4-6 Support	0.4086	0.3779	0.3666
1-3 Support	0.3184	0.3321	0.3371
0 Support	0.2730	0.2901	0.2963
2+ TEL	0.0170	0.0138	0.0131
1 TEL	0.3019	0.2828	0.2785
0 TEL	0.6812	0.7034	0.7084

6. CONCLUSIONS

The aggregation of the state vector used by a Markov chain estimation technique for the composition state of a Level 2 fusion object was developed and implemented for an example application. The technique leveraged database information on the potential classes in the given scenario. Unlike the initial work on this problem [8], the transition probabilities are more easily computed and the probabilities associated with the state elements are not dispersed to the point of uselessness over a large number of states. The reduction in processing requirements resulted in a noticeably faster computational process than previously.

The use of aggregation allows for the use of more computationally efficient approaches for generating the transition probabilities and, in the future, the development of the reports used by the updates. In this latter case, the use of fuzzy logic or the use of binning or quantization of values can be investigated.

The technique developed in this effort can be incorporated easily with a Level 2 implementation that considers the number of targets being tracked, as in [6]. Aggregated groups provide the ability to determine if a detected target could be included in the group. Also, these aggregated groups can provide a probability that a specific target type or types is missing from the detections.

7. REFERENCES

[1] A. Steinberg, C. Bowman, F. White, "Revisions to the JDL Data Fusion Model", P**roceedings of the SPIE Sensor Fusion: Architectures, Algorithms, and Applications III**, pp 430-441, 1999.

[2] S. C. Stubberud, P.J. Shea, and D. Klamer, "Data Fusion: A Conceptual Approach to Level 2 Fusion (Situational Awareness)," **Proceedings of SPIE, Aerosense03**, Orlando, FL., April, 2003.

[3] J. Llinas, C. Bowman. G.L. Rogova, A. Steinberg, E. Waltz, and F. White, "Revisions and extensions to the JDL data fusion model II," **Proceedings of Fusion 2004**, Stockholm, Sweden, pp.1218-1230, July, 2004.

[4] J. Pearl, **Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference**, Morgan Kaufmann Publishers, San Mateo, CA, 1988.

[5] S. C. Stubberud and A. M. Azevedo, Jr., "Using Probabilistic State Estimation For Situational Assessment," **Proceedings of the 7th Multi-conference on Systemics, Cybernetics, and Informatics (SCI 2003)**, Orlando, Florida, July, 2003.

[6] Stubberud, S.C. and K.A. Kramer, "Situational Assessment: Coupled Probabilistic State Estimation For Group Composition," **Proceedings of the 8th World Multiconference on Systemics, Cybernetics, and Informatics (SCI 2004)**, Orlando, Florida, Vol 8, pp.262 – 267, July, 2004.

[7] A. Boutet, P. Rouyer, S. Stubberud, and K. Kramer, "Level 2 Fusion Estimation: Group Composition With Multiple Classes Using A Markov Chain Approach" **Proceedings of the 9th Multi-conference on Systems, Cybernetics and Informatics**, July 2005.

[8] S. C. Stubberud and K.A. Kramer, "Level 2 Fusion: Situational Assessment Composition Fusion With Uncertainty Classification," **Proceedings of the 18th International Conference on Systems Engineering**, pp. 348 – 354, August, 2005.

[9] Norris, J.R., **Markov Chains**, Cambridge Press, New York, 1997.

[10] A.S. Poznyak, A.S., K. Najim, an E. Gomez-Ramirez, **Self-Learning Control of Finite Markov Chains**, Marcel-Dekker, New York, 2000.