2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Towards Probabilistic Operator-Multiple Robot Decision Models

Mark Campbell†, Frédéric Bourgault†, Scott Galster†† and David Schneider†
† Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853, USA
Email: {mc288, fb42, drs44}@cornell.edu
††Human Effectiveness Directorate, Wright-Patterson Air Force Base Dayton, OH, USA
Email: scott.galster@wpafb.af.mil

***Abstract*— Coupled operator-multiple vehicle systems are modelled in a unified framework using probabilistic graphs to yield a methodology for analyzing semi-autonomous systems. The framework uses conditional probabilistic dependencies between all elements, leading to a Bayesian network (BN) with probabilistic evaluation capability. Vehicle attitude/navigation states and target/classification states can be evaluated using nonlinear estimators such as the EKF, Multiple Model filter, information filter, or other approaches. Discrete operator decisions are being modeled as Bayesian network blocks, with conditional dependencies on the vehicle and tracking estimators. Initial decision models use combinations of softmax and discrete probability distributions.**

I. Introduction

Interfacing human operators with multiple autonomous robots engaged in tasks such as search and identify, search and rescue, and cooperative monitoring is a challenging problem. It is critical to the success of such missions to develop an integrated operator-multiple robot system that is robust and efficient in the presence of evolving uncertain environmental parameters.

Important factors for the operator-robots interface are the amount of information displayed, workload level, automation of tasks which enhance overall system efficiency by taking advantage computer execution speed and human ability for complex tactics and strategies [9], [14]. Previous studies on human-automation coordination have revealed both benefits and costs of particular interfaces and designs [11], [12]. As stated in [10]:

> Overreliance, reduced situation awareness, mistrust, mode errors, loss of operator skill, and unbalanced mental workload are among the costs that have been found to be associated with particular styles of interaction between human and automation.

The objective of this paper is to develop a modeling methodology that encapsulates both autonomous, or semi-autonomous platforms and operator decisions in a unifying probabilistic framework. This framework will be useful for analyzing how operators control/task/make decisions with multiple robots, specifically enabling the identification of operator decisions from data (average, confidence, dependencies), and the evaluation of user interfaces, situation awareness, fatigue, or other factors. Such a modeling approach could also be used for predicting operator decisions and evaluate and design important concepts such interactive decision aids and adaptive autonomy levels.

Research in decision modeling has been on-going for many years in the Cognitive Sciences community [14]. Examples include the average time for a person to physically select a choice with their hands, the level of short term memory, and comparison of interfaces. These studies provide valuable insight into how users make decisions as a function of parameters such as stress, interface type, and time. In these studies, however, the researchers typically constrain many of the environmental parameters in order to reduce the complexity of the system and adequately study a single parameter.

Advancements in statistical estimation theory and computer power has allowed for more advanced modeling techniques where these constraints are relaxed; operator decision modeling which use Markov Decision Processes (MDP's) [7] is a good example. The approach proposed here is to identify decisions and environmental variables from data, and develop a model that can be used for prediction, even in the presence of uncertainties in the environment and across multiple users.

The RoboFlag testbed [2], [3] was designed to explore basic operator-vehicle interactions in the context of a game of "capture the flag." Initial work performed a series of human in the loop (HitL) studies [16] utilizing either the simulator or experimental robots, and a specifically designed Playbook with a hierarchy of cooperative control/autonomy tools. This work showed particular trends, such as 1) multiple operators on one terminal cooperate to improve situation awareness and improve command selection; 2) multiple operators scored higher in the game total than single operators; and 3) automations, such as autonomously sending vehicles home to refuel, were used much more as the vehicle speeds and numbers increased. The work also performed tests on the simulator versus experimental robot, showing good correlation. The AFRL/HE group has also implemented a series of HitL studies using RoboFlag [5], [10], [15], examining human performance differences and subjective ratings of mental workload and situation awareness, and how they relate to RoboFlag game outcomes.

This paper is structured as follows. First, the proposed hybrid model for coupled operator-multiple robot systems is described in Section II. This is followed by a brief overview of Bayesian network theory and how they can be used to probabilistically model operator decisions. The background on the RoboFlag game simulator and experimental conditions is covered in Section III. Some initial decision modeling results and experimental data are presented and discussed in Section IV. Finally, conclusions and ongoing research directions are outlined in Section V.

1-4244-0602-1/07/$20.00 ©2007 IEEE.

II. Coupled Operator-Platform Models

In order to model the operator decisions, a probabilistic model using a BN [6], [8] is used because it captures the important elements of the problem, including: i) decisions are modelled with probabilistic dependency on environmental parameters, thus intuitively showing what system parameters drive the decision making process, ii) vehicle system models can easily be defined in this framework, such as using an Extended Kalman Filter for vehicle navigation estimation or an Information Filter for tracking, and iii) the model scales well, as only dependent states are used to define the coupled probabilities.

A BN model itself is represented by a directed graph, with each arrow indicating conditional probabilistic dependency. An example of a single robot tracking a target using a sensor is shown in Fig. 1. This model can be decomposed into three distinct parts: 1) a vehicle model, which is typically evaluated using an on-line recursive estimator for vehicle navigation and attitude, 2) a target tracking model, which is typically evaluated using an on-line recursive estimator for tracking, and 3) an operator decision model. In this case, the operator decisions are to re-task the vehicle planner to orbit a target, travel to new target, or loiter until another command is given. Each of these three models can be cast as a conditional probability, as is shown in Fig. 1. For example, the vehicle sensor output block is give as, $\mathcal{P}(Y|X) = \mathcal{N}(Cx, \Sigma_v)$. Note that the model shown in Fig. 1 is a BN with no time dependence; a Dynamic Belief Network (DBN) model can be developed by repeating the BN model over a series of time slices, with an appropriate time based model [8].

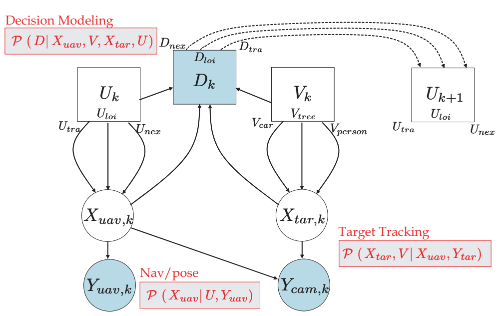

Fig. 1. Probabilistic graph model of a single operator tasking a single UAV to track (and estimate states) of a model. Circles denote continuous vector valued random variables, squares discrete random variables, and shaded blocks indicated that direct measurements are available.

A. Bayesian Network Decision Model Blocks

Operator decision models are defined here using a BN with conditional, probabilistic dependencies on environmental variables. This is written formally as a conditional probability $\mathcal{P}(D|\,X)$

where D is the operators decision and X is a vector of "parent" variables. This function is also called the likelihood function because it denotes how likely the decision data is, dependent on the parent variables. The work here assumes that the operator decisions are discrete, while the parent variables can either be discrete or continuous. Fig. 2 shows the three versions of this decision model, where the decision has a direct conditional probabilistic dependency on the parent variables. The other two cases in Fig. 2 are more complex, graphical models of the operator decisions.

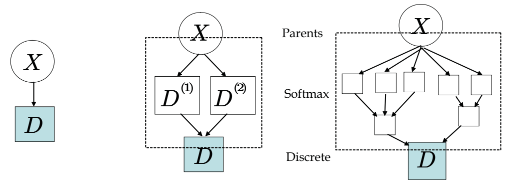

Fig. 2. Bayesian Network based decision models. Left: simple model showing only conditional probabilistic dependency on the parent state variables. Middle and Right: more complex decision models, with hidden random variable blocks between the discrete decision block and the parent variable block.

While the softmax distribution has very nice properties for decision modeling, a big disadvantage is that it requires the decisions (and decision data) to nicely separate using hyperplanes. Obviously this is very limiting. A solution to this problem is to create a Bayesian network with combinations of discrete and softmax variables. This is shown in Fig. 2 (middle and right). In this case, the BN contains the parents (continuous or discrete random variables), logit/softmax random variables (intermediate and hidden), and a discrete random variable which acts to select which logit/softmax variables contribute to the decision distribution.

While similar to neural network theory, the BN decision block is developed and interpreted using only statistical properties. Consider first using s softmax variables, defined for the lth variable as

$$\mathcal{P}_{lk} = \mathcal{P}\left(D^{(l)} \middle| X\right) = \frac{e^{\tilde{w}_l^T \tilde{x}_k}}{1 + \sum_{j=2}^{m} e^{\tilde{w}_j^T \tilde{x}_k}} \tag{1}$$

where $\tilde{w}_j = \frac{1}{\sigma_j}[w_j b_j]^T$ and the state vector to be $\tilde{x} = [x1]^T$ where w is a relative weighting of the parents ($||w|| = 1$), σ is the steepness of the threshold, and b is a "bias" from zero. It is noted that given m decision variables, only $m-1$ weight sets (w, σ, b) are required (one is redundant can be arbitrarily selected).

The joint density of all variables is written as $\mathcal{P}(D, D^1, \cdots, D^s, X)$. Using Bayes Theorem, the joint density conditioned on the parent variables is written as

$$\mathcal{P}\left(D, D^1, \cdots, D^s \middle| X\right) = \frac{\mathcal{P}\left(D, D^1, \cdots, D^s, X\right)}{\mathcal{P}(X)} \tag{2}$$

The softmax distribution has several important properties.

- optimizing the likelihood function to estimate the distribution parameters is a convex problem, which guarantees convergence to a global maximum
- the likelihood ratio, or how likely one decision is compared to another, is a single hyperplane defining a decision boundary
- the final estimates have Gaussian statistical confidence regions, given enough data.

Ref. [1] outlines a solution procedure for this joint density in order to find a common, conditional density. The steps include:

1) Using Bayes Theorem on the right side to separate out all s the internal softmax variables random variables;
2) applying the rules of conditional independence (for the directed graph) to the right side;
3) marginalizing out all internal softmax variable, D^l on the left side.

These simplifications then yield:

$$\mathcal{P}(D|X) = \sum_{D^1:D^s} \mathcal{P}\left(D|\, D^1, \cdots, D^s\right) \cdot \prod_{l=1}^{s} \mathcal{P}\left(D^l \middle| X\right) \quad (3)$$

Equation (3) shows that the likelihood $\mathcal{P}(D|X)$ can be written as a function of the internal softmax densities, and a discrete random variable density which selects which softmax random variables contribute to the decision density. Notice that this likelihood is identical to that of the original softmax conditional probability, albeit with a more complex functional representation. This implies that the BN decision model block can be directly compared to other decision model blocks of varying structure and complexity simply by comparing their likelihood functions. Considering Fig. 2, this implies that one can compare the likelihood functions for each of the three examples shown in this figure. This is a powerful result because both the structure and internal density parameters can be optimized and compared using the *same* likelihood function.

As described in [1], the BN decision block has the following properties:

- For $m - 1$ convex decision clusters, the likelihood function never has any minima;
- The asymptotic likelihood function has $s!$ maxima, all with identical likelihoods; and
- Parameter estimates, given sufficient data, have Gaussian confidence regions as the softmax decision variables also had. Under a mild set of regularity conditions, the MLE approaches a normal distribution Therefore, one can develop confidence bounds about the weight estimates (and decisions) using Gaussian theory or a Chi-square distribution with n_w degrees of freedom.

A final point is a result of the asymptotic MLE theory: Even with multiple maxima, the regularity conditions still hold and therefore, the Gaussian error properties hold as well.

III. RoboFlag Game

A. Background

RoboFlag, is a simulation and experimental testbed [3], [13] with autonomous, fast-moving teams of vehicles, and is therefore an excellent system to aid in the development and evaluation of realistic solutions for semi-autonomous control. The objective of the RoboFlag competition [4] is to venture into opponent territory, locate and capture the "flag," and return to the "home base." Many key aspects future systems are included in this game, including a human operator, team dynamics, different levels of tasking, cooperative planning, and uncertainties such as incomplete information, latency, and an intelligent adversary.

RoboFlag exists in both hardware (real robots), and software (a real time simulator). A software arbiter read a text

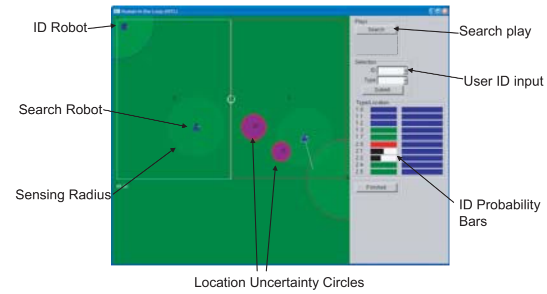

Fig. 3. RoboFlag GUI for the first set of games implemented by AFRL/HECP.

file where game conditions, such as vehicle and sensor types, communication cones, weather patterns, etc., can be changed and updated. The arbiter also relays commands to the robots.

B. Description

A set of RoboFlag games was conceived to study operator decision making within a search-and-identify type of mission. Operators controlled three vehicles: two fast moving search vehicles ($SV1$, $SV2$), each with the ability to locate entities; and one slow moving ID vehicle (IDV), with the ability to both locate entities and identify their type. During the mission, three entities could be encountered: a stationary flag; a stationary red robot that can tag blue vehicles which come in too close proximity; and a red Chaser vehicle ($CHSR$), designed to chase any blue vehicle within a particular range. Upon getting tagged, a blue team vehicle must automatically return all the way to home base at a very slow pace before it can return to play.

At first, when an entity is detected by the search vehicles, its location is highly uncertain and is contained inside a probability circle. Slowly the uncertainty radius decreases as more localization data is collected, i.e. while target is in sensor field-of-view. Search vehicles can cooperate and fuse sensory information in order to improve information collection. The ID vehicle collects identity information more quickly if the location uncertainty is small. When users are confident of the final entity type, the user chooses (flag or red robot) formally using a GUI input, as shown in Fig. 3. Once both targets have been localized and identified by the user, he may then terminate the game by pressing the "Finish" button and the final time gets recorded.

As shown in Fig. 3, uncertainty information is displayed as an uncertainty circle which decreases in size as more information is collected by the search vehicle. This decrease is approximately exponential, and mimics traditional estimation/tracking software. The ID probability is given by a bar on the right side of the GUI. Only the ID vehicle could move the probability from its initial, a priori value.

Users had two approaches to controlling the robots: 1) single way points by selecting a vehicle, i.e. left-clicking on it, followed by a way point allocation, i.e. right-clicking at the desired destination , and 2) selecting the search play button, followed by the entry of a series of way points. In addition, cooperation could be enabled by the user commanding two search vehicles to go to a single target (with the effect of reducing the uncertainty faster), or by reducing the uncertainty before or while the ID vehicle was near the target.

C. Experiment

Experiments were led by Dr. Scott Galster at AFRL/HE. Sixteen subjects were selected from the AFRL subject pool. Data recorded included all robot telemetry, and all users clicks (saved as "events."). Each event was assumed to be a user decision (even a continuation of the current user decision).

Each subject was initially trained, and then completed a 4x4 matrix of 16 trials, where two parameters were varied:

1) Location of targets within field
2) Type of targets (i.e., 0/2,1/1 and 2/0 of tagging robot/flag)

In addition, to facilitate decision evaluation, digital videos were recorded for all trials. Finally, four additional trials were added after the original 16, where users were asked to "describe" their decisions, actions, and strategies during the games.

IV. Experimental Results

Figure 4 recapitulates the performance for the 16 subjects over their 20 trials. It is hard to compare the subjects performance between trials and amongst each other due to the randomization of each trial's initial conditions. However, there appears to be a slight decrease in mean final game time from trial #1 to trial #16 as the subjects are learning and becoming more efficient.

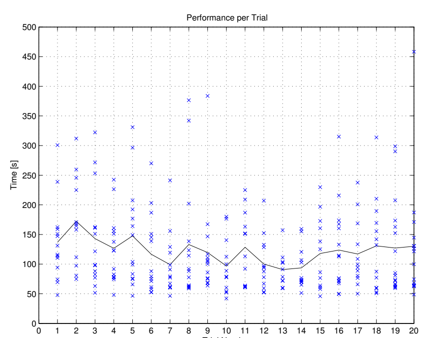

Fig. 4. Performance per trial: Final game time for all subjects at each trial with their mean time (*solid line*).

As mentioned in the previous section, the workload for the subject is increased during trials #17 to 20 as they have to describe on the fly what they are doing while being recorded. This translates into a slight increase in final mean time for these trials compared to the ones just before. However, one can still see how the subjects have learned since these trials are repeats of trials #1 to 4.

A. Strategic Level Human Decisions

Table I describes the list of the strategic level decisions made by the human operators when tasking the robots. Unless a subject specifically states what his intention was when tasking a robot, there is no direct mean to determine the actual true decision. Instead, the strategic decisions must be probabilistically inferred from the current state of the world the actual tactical control input, i.e. way point assignment when the mouse event was recorded.

TABLE I
DEFINITION OF THE HUMAN OPERATOR STRATEGIC LEVEL DECISIONS/ACTIONS WHEN ALLOCATING TASKS TO VEHICLE.

Decisions:	Definitions:
StrategicPos	Strategic positioning: Positioning of an asset so it is ready to use later, e.g. usually the first moves before starting target search.
SaftyZone	Move to Safety Zone.
ID/Localiz	Positioning to ID and/or Localize one of the Targets.
ID/LocalizBoth	Positioning to ID and/or Localize Both Targets.
SearchTarget	Searching for either or both Targets.
SearchChaser	Searching for Chaser.
Escort	Escort ID vehicle: Tasking a Search vehicle to either i) move along with the ID vehicle on its way to a Target to protect it from the Chaser, or ii) guard a position somewhere along the ID vehicle's path to the target to be ready to decoy the Chaser. Note: This move is different than "SearchChaser" and "Decoy" which are used to actively find and decoy the Chaser.
Evade	Evading, i.e. getting away from Chaser to prevent being tagged.
Evade+ID/Loc	Evading and IDing/Localizing Target Combination: Getting away from Chaser while intently IDing and/or Localizing one, or both, of the Targets.
Evade+Search	Evading and Searching for Targets Combination: Getting away from Chaser while intently searching for remaining Target(s).
Decoy	Attracting the Chaser away from the ID Vehicle or keeping it distracted by intently getting chased.
Decoy+Localiz	Decoying and Localizing a Target Combination: Decoying Chaser while Localizing one, or both, of the Targets.
Decoy+Search	Decoying and Searching for Target Combination: Getting away from Chaser while intently searching for the remaining Target.
Avoid	Immediate reaction to Avoid collision with either a Target or Chaser when suddenly detected. Next move usually consists in positioning to ID/Localize the newly detected Target or Evade the Chaser.
SelectionError	Operator Selection Mistake: Usually when the operator tries to select an agent but clicks next to it resulting in a wrong destination (waypoint) assignment.
Other	Other unlisted decisions.

Figure 5 illustrates the actual total number of times each of the actions described in Table I were used. Unsurprisingly, the operators spent a lot of time searching targets and positioning their robots either strategically for later use, or for IDing/Localizing the targets. Another play that ended consuming a lot of the subjects attention was the combination of Searching for and Decoying the Chaser in order to leave the path free for the slow ID vehicle to make its way to IDing the targets. This required a high level of attention and coordination on the part of the users as they had to keep a search vehicle within the Chaser's sensor field-of-view while making sure not to get tagged by it. In fact the total number of events recorded for these two plays together is larger than any of the other events.

Tables II and III respectively show how the ID and search vehicles' strategic decision mode probabilities are assumed to be conditioned on the state vectors. Each"x" in the tables indicates that the mode in the corresponding row is conditionally dependent on one or more of the scalar state

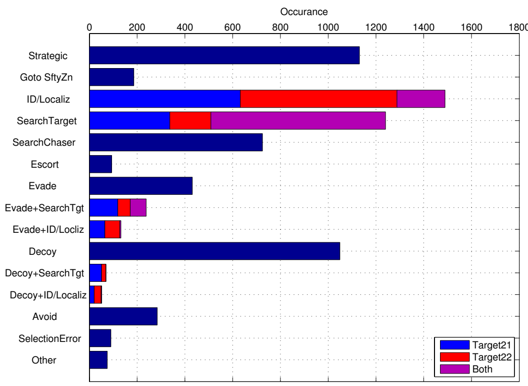

Fig. 5. Frequency of human strategic decisions defined in Table I. Note that these were compiled by human observers whose interpretations are subject to uncertainty.

TABLE II

SEARCH VEHICLE i STRATEGIC LEVEL DECISION MATRIX: GENERAL CONDITIONAL DEPENDENCIES ON STATE VECTORS MARKED BY "X"'S.

Decisions: $U_{SVi}^{Strategic}$	X_{SVi}	X_{SVj}	X_{IDV}	X_{TRGT1}	X_{TRGT2}	X_{CHSR}
StrategicPos	x	x	x	x	x	
SaftyZone	x			x	x	x
Localize	x			x	x	
LocalizeBoth	x			x	x	
SearchTarget	x	x		x	x	
SearchChaser	x	x		x	x	x
Escort	x	x	x	x	x	
Evade	x					x
Evade+Loc	x			x	x	x
Evade+Search	x			x	x	x
Decoy	x		x			x
Decoy+Loc	x		x	x	x	x
Decoy+Search	x		x	x	x	x
Avoid	x			x	x	x
SelectionError		x	x	x	x	x

TABLE III

ID VEHICLE STRATEGIC LEVEL DECISION MATRIX: GENERAL CONDITIONAL DEPENDENCIES ON STATE VECTORS MARKED BY "X"'S.

Decisions: $U_{IDV}^{Strategic}$	X_{IDV}	X_{SVi}	X_{SVj}	X_{TRGT1}	X_{TRGT2}	X_{CHSR}
StrategicPos	x	x	x	x	x	
SaftyZone	x			x	x	x
ID/Localiz	x			x	x	
ID/LocalizBoth	x			x	x	
SearchTarget	x	x	x	x	x	
Evade	x					x
Evade+Loc	x			x	x	x
Evade+Search	x			x	x	x
Avoid	x			x	x	x
SelectionError		x	x	x	x	x

variables of the state vector in the corresponding column. For instance, the probability of the ID vehicle mode "Evade" is conditioned on knowledge of the ID vehicle's location and the location of the chase vehicle, which are contained in the state vectors X_{IDV} and X_{CHSR}, respectively. As with all DBNs, these conditional dependence assignments are based on designer experience. However, the definition of these state variables is also convenient since all non-decision states are observable and serve as the basis for all other observable information about the state of the game that are important in operator decision making.

However, not all vehicle decision modes are conditioned easily on the given state variables, since some decisions may depend on more subtle relationships between variables in the vehicle and target states. For instance, the mode "Localize" for each search vehicle can be assumed to be conditionally dependent on the relative distance of each search vehicle to the nearest target.

Initial results of decision modeling for how the 16 users tasked the Search and ID vehicles are shown in the following subsections based on the intuitive state dependency matrices of Tables II and III. For instance in subsections IV-B and IV-C the models assumes three tasks for the ID vehicle: 1) move vehicle to a loiter pattern, 2) move vehicle near target to identify its type, and 3) move vehicle to avoid the Chaser. Decisions of which modes to task the ID vehicle were primarily a function of the uncertainty circle size, probability of ID, and range to the Chaser.

B. Case 1: ID Vehicle Evade

Fig. 6 shows the simplest case of avoiding the chaser. Users always tasked the ID vehicle to avoid the Chaser when the range to the Chaser was small; this decision was not a function of the other system variables.

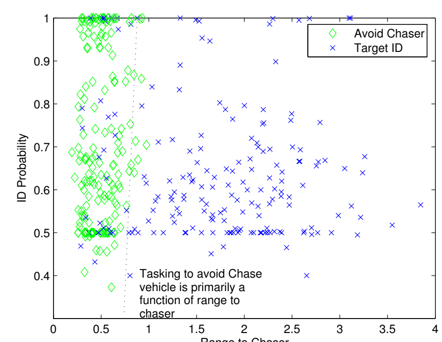

Fig. 6. Decision data and decision boundaries from the RoboFlag games: Operator tasking the ID robot to Evade Chaser.

C. Case 2: ID Vehicle ID/Localize

Fig. 7 shows that users typically tasked the ID vehicle to move and identify the target when the ID probability was small; the uncertainty was also usually small, but occasionally users would coordinate the motion of two vehicles and task the ID vehicle as the search vehicles were moving. The loiter task was commonly used at the start of the game and in mid-game as a waiting point.

D. Case 3: Multiple decisions

Figs. 8(a) to (c) are three different representations of the same decision data for i) Strategic Positioning, ii) Search Target, iii) Move to ID/Localize, and iv) Move to Safety Zone. Fig. 8(a) corresponds to the actual vehicle state location at the time of the decision while Fig. 8(b) represents the tactical control input corresponding given that strategic decision. Fig. 8(c) is just another way of representing the data

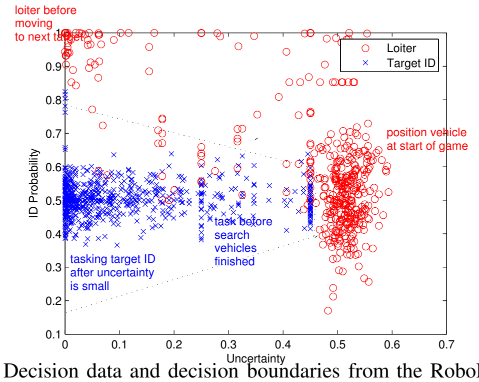

Fig. 7. Decision data and decision boundaries from the RoboFlag games: Operator tasking the ID robot to move and identify an entity, or to loiter before being re-tasked.

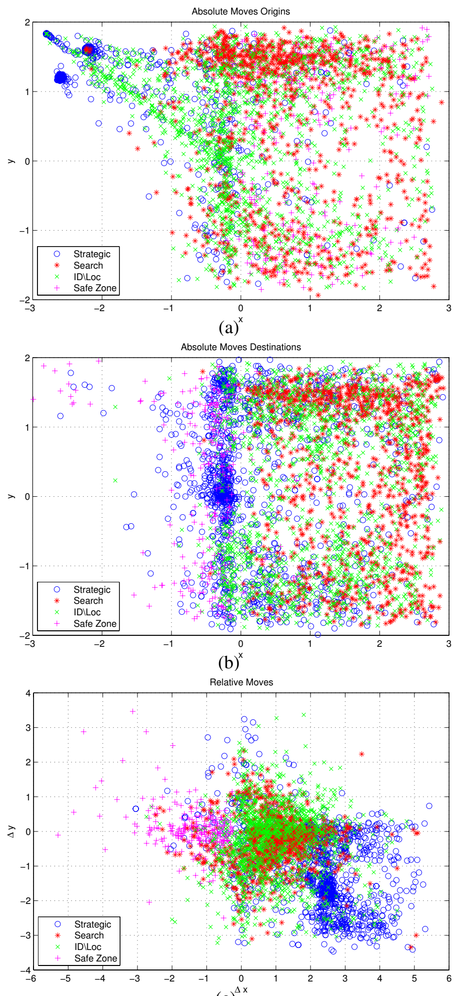

Fig. 8. Decision data from the RoboFlag games: (a) tasked vehicle sate location at mouse event; (b) way point location assigned to vehicle, and (c) relative destination assigned to vehicle.

of Fig. 8(b). This shows that it is often possible to distinguish more easily certain decisions depending on which states their are projected.

E. Case 4: Evade vs. Decoy decisions

Figs. 9(a) to (c) represent the same projections as described in Subsection IV-D for the Evade and Decoy decisions. Although both decision occurs for similar state loca-

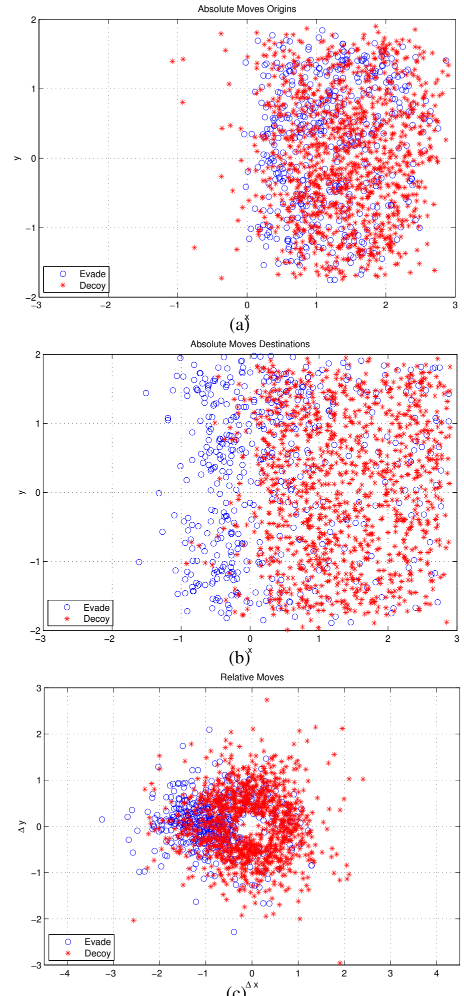

Fig. 9. Decision data from the RoboFlag games: (a) tasked vehicle sate location at mouse event; (b) way point location assigned to vehicle, and (c) relative destination assigned to vehicle.

tion (Fig.9(a)) their tactical outputs are very distinguishable when compared in Figs. 9(b) and 9(c). Evade tend to send the vehicle back into the safe zone while Decoy tend to move around in the the enemy zone "making circles" around the Chaser clearly illustrated by the radial relative moves in Fig. 9(c).

F. Case 5: Target identification decisions

Fig. 10 illustrates the operators target identification decisions. Basically when a subject felt he had acquired enough information about a target type corresponding to a high ID probability he would submit his identification input. Most subjects waited until they had reached at least 80% of probability of the target being either a robot or a flag. However, in some cases, the subject became frustrated by getting his ID vehicle tagged multiple times and decided to ID the target even without information rather than having to wait for the vehicle to become available again. Note that no data point are in the upper right corner of Fig 10 above the dash dotted sensing boundary. This boundary correspond

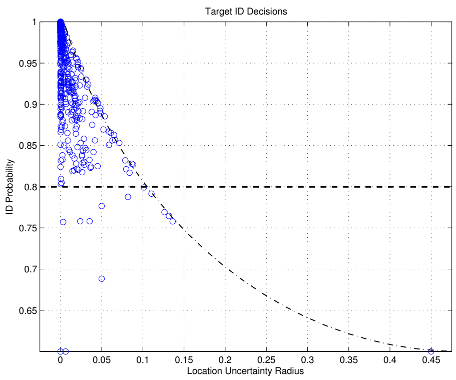

Fig. 10. Decision data and decision boundaries from the RoboFlag games: The Data points represents the information available to the users when they ID'ed the Targets with sensing boundary (*dash dotted line*) and type uncertainty threshold (*horizontal dashed line*).

to the combined rate of reduction of location and ID type uncertainties when the ID vehicle is positioned in close proximity to the target. Therefore the only parameter directly affecting the decision in this case is the ID probability.

V. Conclusions and Ongoing Work

A modeling methodology for coupled operator-multiple vehicle systems is proposed using a unified framework in a probabilistic graph setting. The framework uses conditional probabilistic dependencies between all elements, leading to a Bayesian network with probabilistic evaluation capability. Vehicle attitude/navigation states and target/classification states can be evaluated using nonlinear estimators such as the EKF, Multiple Model filter, information filter, or other approaches. Discrete operator decisions are being modeled as Bayesian network blocks, with conditional dependencies on the vehicle and tracking estimators. Initial decision models use combinations of softmax and discrete probability distributions. Maximum likelihood estimation is used to estimate the structure and internal distribution parameters of the model. Maintaining statistical formalism with the graph and estimation tools will enables a probabilistic model with confidence bounds to be developed.

The ongoing research effort aims at:

- *Higher order decision models* such as when decisions are clustered in two distinct subspaces will be explored. Approaches include clustering of decision data and mixtures of Gaussians.
- *Recursive models* which add time dependency and maintain statistical rigor will be developed. Approaches include a log-likelihood receding horizon, underbounding the likelihood for potential inference methods, and neural network tools with a statistical evaluation at the output.
- *Models for prediction* are being developed because of the statistical rigor of the BN and optimization of the likelihood. Confidence estimates using Gaussian tails or Chi-square hypothesis testing allow the model to be used predictively, with the weight error models as dependent parameters.
- *Structural Learning* will attempt to find the best fit with minimal conditional dependencies in the BN model. Approaches include using the Bayesian Information Criterion and likelihood ranking.

Finally, two RoboFlag tests will be implemented. The first will study the use of cooperative vehicle decision aids, and the second will study adaptive tasking [9], where the game will switch between manual and decision aids with varying levels of autonomy in order to maximize performance.

Acknowledgements

This work is partly supported by the Air Force Office of Scientific Research (AFOSR), USAF, under contract number FA9550-05-1-0118. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFOSR or the U.S. Government.

References

[1] M. Campbell, S. Sukkarieh, and A.H. Goktogan. Operator decision modeling in cooperative uav systems. In *Proceedings of The AIAA Guidance, Navigation, and Control Conference (GNC'06)*, Keystone, CO, August 2006.

[2] Mark Campbell, Raffaello D'Andrea, David Schneider, Atif Chaudhry, Steve Waydo, Jeff Sullivan, Jesse Veverka, and Atif Klochko. Roboflag games using systems based, hierarchical control. In *American Control Conference*, Denver, CO, 2003.

[3] R. D'Andrea and M. Babish. The roboflag testbed. In *American Control Conference*, Denver, CO, 2003.

[4] R. D'Andrea and R. Murray. The roboflag competition. In *American Control Conference*, Denver, CO, 2003.

[5] Scott M. Galster and R. S. Bolia. Decision quality and mission effectiveness in a simulated command and control environment. In *Second Human Performance, Situation Awareness, and Automation Conference*, 2004.

[6] D. Heckerman. A tutorial on learning with bayesian networks. Technical report, Microsoft Corporation, 1995. MSR-TR-95-06.

[7] Alex Kirlik. Modeling strategic behavior in human-automation interaction: Why an "aid" can (and should) go unsused. *Human Factors*, 35(2):221–42, 1993.

[8] K. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, UC Berkeley, 2002.

[9] R. Parasuraman. Effects of adaptive function allocation on human performance. In D.J. Garland and J.A. Wise, editors, *Human Factors and Advanced Aviation Technologies*. ERAU, Daytona Beach, FL, 1993.

[10] R. Parasuraman, S. Galster, and C. Miller. Human control of multiple robots in the roboflag simulation environment. In *IEEE International Conference on Systems, Man, and Cybernetics*, Washington DC, 2003.

[11] R. Parasuraman and V.A. Riley. Humans and automation: Use, misuse, disuse, abuse. *Human Factors*, 39:230–253, 1997.

[12] N.B. Sarter, D. Woods, and C. Billings. Automation surprises. In G. Salvendy, editor, *Handbook of Human Factors and Ergonomics*. Wiley, New York, NY, 1997.

[13] D. Schneider, M. Campbell, S. Galster, A. Klochko, and J. Veverka. The roboflag test system for decentralized autonomous and semi-autonomous cooperative multi-agent systems research. *submitted IEEE Transactions on Control Systems Technology*, 2006.

[14] Thomas B. Sheridan. *Telerobotics, Automation, and Supervisory Control*. MIT Press, Cambridge, MA, 1992.

[15] P.N. Squire, S.M. Galster, and R. Parasuraman. The effects of levels of automation in the human control of multiple robots in the roboflag simulation environment. In *Second Human Performance, Situation Awareness, and Automation Conference*, 2004.

[16] J. Veverka and M. Campbell. Experimental study of information load on operators in semi-autonomous systems. In *AIAA Guidance, Navigation and Control Conference*, Austin TX, 2003.