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

AIAA 2007-6589

Probabilistic Operator-Multiple Robot Modeling Using Bayesian Network Representation

Frédéric Bourgault, Nisar Ahmed, Danelle Shah and Mark Campbell
Sibley School of Mechanical and Aerospace Engineering
Cornell University, Ithaca, NY 14853, USA
{fb42,nra6,dcs45,mc288}@cornell.edu

Abstract

A framework using Bayesian Networks (BNs) to model interactions between humans and multiple semi-autonomous vehicles is introduced. Discrete operator decisions are modeled as probabilistic BN blocks with conditional dependencies on individual system states. Human decision data was collected on the RoboFlag simulation testbed, which served as an experimental platform to observe human control of simulated semi-autonomous teams of robot vehicles. The data is used to generate non-parametric decision models through a method based on Parzen density estimation. The identified probabilistic graph can be used in estimation form (update distributions based on new data) or prediction form (predict the probability of a decision based on new data). Actual decision modeling results are shown using data from the RoboFlag experiments.

I. Introduction

This paper develops an innovative Bayesian approach for human interactions with semi-autonomous robotic systems performing missions in unstructured and dynamic environments. 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.

The aim of this work 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 it will lead to the automatic 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.

A seminal reference on decision modeling which summarizes the results of operator-machine systems research in the Cognitive Sciences community is given by Sheridan.[1] Examples include quickness of physical reaction, levels of short term memory, and effectiveness of different user interfaces. This work has led to integrated databases for modeling/prediction of perception and motor skills[2–5] and provides valuable insight into how users make decisions as a function of parameters such as stress, interface type, and time. In these studies however, many of the environmental parameters are typically constrained in order to reduce the complexity of the system and adequately study a single parameter. Similarly, research in the field of human-robot interaction in multi-robot systems is often limited to only a few humans and a small number of robots. Increased levels of autonomy are required by the robotic platform to improve the scalability of such systems since the communication bandwidth and the ability for the human operator to provide simultaneous assistance to several robots are limited.

Advances in statistical estimation theory and computer power have allowed for more sophisticated modeling techniques where the simplifying constraints are relaxed; operator decision modeling which use Markov Decision Processes (MDP's)[6] is a good example. The long term objectives of the approach proposed here

are twofold. Firstly, decisions and environmental variables are identified from data and used to develop a probabilistic model. The model can then be used for prediction, even in the presence of uncertainties in the environment and across multiple users. Secondly, the causes of decreased performance or mission failures in human-robotic systems will be studied at a later time. As systems become more complex, adding automation to reduce the occurrence of failures or increase overall performance becomes more desirable. However, despite system complexity, human operators are often hesitant to give too much control over to automation.[7] This paper focuses on the first objective of obtaining probabilistic decision models from data, while the development and verification of metrics and estimation methods of human-robotic systems performance to complement these models will be the subject of further work.

Details about an operator decision modeling methodology using a Bayesian Network graph framework, softmax and discrete random variables were provided in Refs. 8, 9. Such a network enabled a formal probabilistic model of human decisions to be realized. This paper extends this theory in two important ways with: i) Non-parametric density modeling using Parzen kernels (also see Ref. 10), and ii) Pre-processing decision data into initial strategies using prioritized conditional dependencies (also see Ref. 11).

Experimental human decision data was collected at the Air Force Research Laboratory (AFRL) using Cornell's RoboFlag testbed[12, 13] to explore basic operator-vehicle interactions in the context of an adversarial game of "search-and-identify." Some of the human decision models obtained from applying this theoretical framework are presented.

This paper is structured as follows. First, Section II describes the proposed model for coupled operator-multiple robot systems based on probabilistic graphs, also known as Bayesian Networks (BNs). This includes a brief overview of BN theory and how they can be used to probabilistically model operator decisions. Section III introduces some background and motivation for model reduction, as well as a proposed method to be used towards developing a Bayesian Network model. Section IV describes the RoboFlag game simulator and the experimental setup. Section V presents some of the experimental data and corresponding decision modeling results. Finally, conclusions and ongoing research directions are outlined in Section VI.

II. Probabilistic Models for Coupled Operator-Multiple Robot Systems

In this paper, BN models[14, 15] are used to model human decisions, task allocation, interactions, and sensory inputs. These models capture in a formal unified probabilistic framework the important elements of the problem, such as: decisions' probabilistic dependencies on state parameters,[8] variations across human operators and environmental conditions, vehicles dynamics (e.g. using an Extended Kalman Filter for vehicle navigation estimation or an Information Filter for tracking) and opponents' intentions. These models scale well, as only dependent states are used to define the coupled probabilities, thus making the framework ideal to model collaborative human-robot systems[9] in adversarial situations. It is also an ideal framework for modeling human-human collaboration as well as hierarchical decision systems.

A BN model itself is represented by a directed graph, with each arrow indicating conditional probabilistic dependency. By maintaining probabilistic connections between nodes, the model can be used for formal probabilistic analysis, such as estimation and prediction. To illustrate the ability of BN models to capture joint human-robot dynamics, Fig. 1 shows a model of a single Unmanned Air Vehicle (UAV) tracking ground targets with a camera. This model can be decomposed into three distinct parts: a vehicle pose model, a target tracking model, and an operator decision model. The vehicle/UAV model includes the mode of operation (U), position/attitude state (X_{uav}), and sensor measurements (Y_{uav}). Estimation of the vehicle state can be accomplished using the well known Kalman Filter,[16] which can be written in terms of the graph as $P(X_{uav}|U, Y_{uav})$ (probability of the UAV state, given the mode of operation and measurements). The Target Tracking model includes the target type (V), position state (X_{tar}), and camera measurement (Y_{cam}), which depends on the location of the target and location/attitude of the UAV. Joint estimation of the target type and state is a classic target tracking problem, which can be solved with a multiple model Kalman Filter,[16] $P(X_{tar}, V|X_{uav}, Y_{cam})$.

In this example, the operator decisions (D) are the discrete tasking and re-tasking of the UAV planner (mode of operation (U)) to orbit a target, travel to new target, or loiter until the next time step. The conditional probability $P(D|X_{uav}, V, X_{tar}, U)$ naturally captures the probability of a certain decision given the UAV mode and its location, and the target type and its location. The focus of this paper is on the operator decision model and coupling with the other two models. 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.[15]

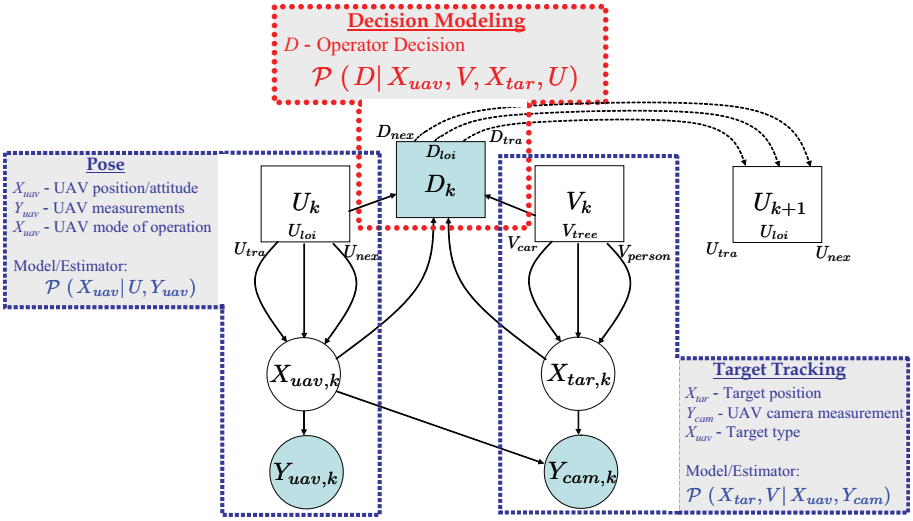


Figure 1. Probabilistic graph model of a single operator tasking a single UAV to track (and estimate states) of a target. Circles denote continuous vector valued random variables, squares discrete random variables, and shaded blocks indicate that direct measurements are available.

A. BN Representation for Probabilistic Decision Modeling

Operator decisions are modeled using a BN defining conditional probabilistic dependencies on environmental variables. Formally, the conditional probability density function (PDF) of the operator's decisions D based the vector of "parent" variables X, is is given by $P(D|X)$. This function is also called the likelihood function because it denotes how likely a given decision instance is as a function of the parent variables. The work here assumes that the operator decisions are discrete, while the parent variables can either be discrete or continuous. Technically, for discrete decisions, $P(D|X)$ should be referred to as a Probability Mass Distribution (PMD). In this paper, we will keep referring to it as a PDF since, in general, the decisions may either be discrete or continuous. Figure 2(a) illustrates the corresponding BN. Also, as discussed in SectionV-A and illustrated in Fig. 7(a), it is possible to simplify the learning process of the decision block for a complex multi-vehicle system by subdividing the block into multiple hierarchical levels of decision making, i.e. coordination, strategic, tactical and low-level vehicle control, leading to the following notation:

$$P(D|X) = P(U_{coord}, U_{strat}, U_{tact}, U_{veh}|X). \tag{1}$$

Since decisions at one level often depend only on the state of the world X and the decision made at the level above, Equation 1 generally simplifies nicely as follows by applying the probability chain rule

$$P(U_{coord}, U_{strat}, U_{tact}, U_{veh}|X) = P(U_{coord}|X) \cdot P(U_{strat}|U_{coord}, X) \cdot P(U_{tact}|U_{strat}, X) \cdot P(U_{veh}|U_{tact}, X). \tag{2}$$

B. Non-parametric Probabilistic Decision Models Using Parzen Density Estimation

Consider the case of n_D discrete decision classes in D, and a set of system state variables X. The operator decision likelihoods $\Lambda_i(X) = P(D = i|X)$, $\forall i \in [1, ..., n_D]$ may simply be obtained from the following Bayes' rule

$$P(D = i|X) = \frac{P(X|D = i) \cdot P(D = i)}{P(X)} = \frac{P(X|D = i) \cdot P(D = i)}{\sum_{j=1}^{n_D} P(X|D = j) \cdot P(D = j)} \tag{3}$$

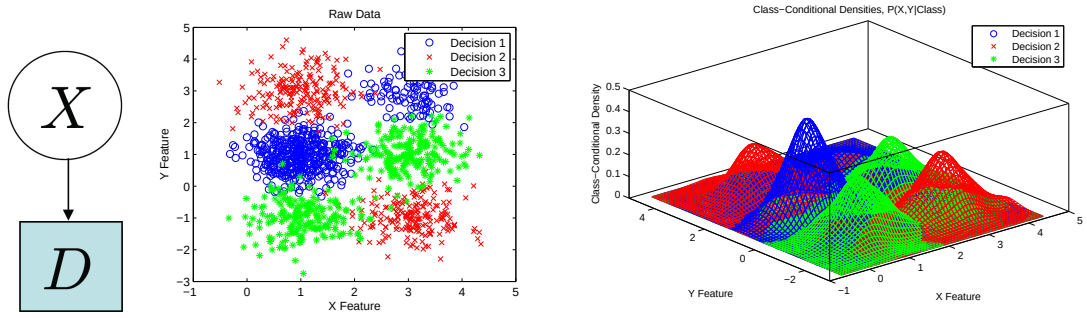
where the decision class priors, $P(D = i)$, $\forall i$, may be estimated from the training data using frequency counts, while the class-conditional PDFs for the state-variables, $P(X|D = i)$, $\forall i$, can be estimated via parametric, semi-parametric, or non-parametric methods. If $P(X|D)$ may not easily be represented by any known parametric PDF, then semi-parametric, or nonparametric density estimation techniques are used.[17]

Parzen kernel methods are widely used for non-parametric density estimation in statistical classification because the estimates are easy to train and can offer robust performance without requiring any assumptions about the underlying class-conditional densities.[18] For a given decision class i with training set x^i with n_p points, the Parzen density estimate is given by a sum of n_i kernel PDFs centered about each sample point, or $P(X|D=i) \approx \sum_{p=1}^{n_i} f(X, x_p^i)$. Gaussian kernels are common and lead to the following sum of Gaussians mixture model (for a q-dimensional state-space):

$$\hat{P}(X=x|D=i) = \frac{1}{n_p(2\pi)^{q/2}|H|^{q/2}} \sum_{p=1}^{n_p} \exp\Big\{ -\frac{1}{2}(x-x_p^i)^T H^{-1}(x-x_p^i)\Big\}, \quad H = h\Sigma, \tag{4}$$

where h is the window, or bandwidth parameter which scales the kernel covariance Σ and acts as a smoothing parameter for the estimated density. The kernel covariance Σ usually is either assumed to be diagonal, e.g. identity, or may be the unbiased estimate of the class-conditional covariance computed from the data. It may also be computed adaptively based on the local data. The parameter h may also be fixed or selected adaptively based on popular criteria, such as: likelihood maximization for the individual class-conditional densities, and minimization of the estimated probability of misclassification.[19,20] Although not used in this paper, the cost of computation and storage of all training data for Parzen can be reduced significantly via data condensation techniques.[21,22]

The nonparametric approach leads to the simple BN representation shown in Fig. 2(a). Since Parzen density estimates asymptotically approximate any probability density function, they are ideal for generating multi-modal and/or non-linearly correlated decision surfaces. As an example, Fig. 2(c) shows the bi-modal Parzen density estimates $P(X|D=i)$ obtained for the corresponding $n_D = 3$ classified training data sets x^i shown in Fig. 2(b). Likelihood maximization is used to select the kernel window parameter h for each class.



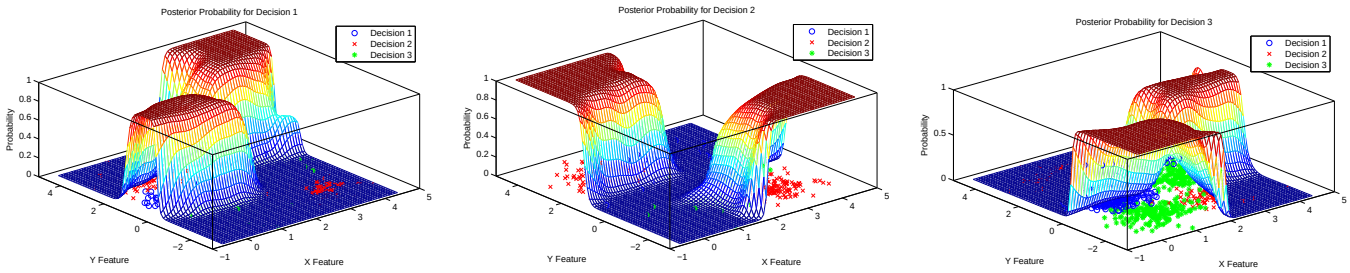
(a) BN (b) x^i (c) $P(X|D=i)$

Figure 2. Parzen regression for n_D=3 classified decision data sets: (a) Simple Bayesian Network (BN) model showing conditional probabilistic dependency of the decisions D on the parent state variables X only; (b) 2D sample training sets, $\{x^1, x^2, x^3\}$; and (c) Parzen conditional probability density estimates $P(X|D=i)$ for decisions i=1,2 and 3.

It is interesting to notice that deterministic decisions, $d(X)$, function of the state, may be obtained directly from the Parzen PDF estimates $P(X|D=i)$'s by applying the following rule

$$d(X) = \arg\max_i P(X|D=i) \cdot P(D=i). \tag{5}$$

However, probabilistic decisions may be generated from the likelihood functions $\Lambda_i(X) = P(D=i|X)$ obtained by applying Equation 3 to the Parzen estimates. Both equations here assume that the cost of making a wrong decision is uniform across the state X. They may also easily be modified to include a non-uniform cost or risk function parameter which would skew the distributions and slightly move the decision boundaries. Figures 3(a) to 3(c) show the post-processed likelihood functions for decisions i=1,2 and 3, respectively, for the sample data and PDF estimates of Figs. 2(b) and 2(c). Notice that the likelihood gradients are fairly steep which indicative mostly separable decision data and correspondingly sharp decision boundaries.



(a) $P(D = 1|X)$ (b) $P(D = 2|X)$ (c) $P(D = 3|X)$

Figure 3. Parzen regression for n_D=3 classified decision data sets: (a)–(c) Non-parametric decision likelihoods $\Lambda_i(X) = P(D = i|X)$ for decisions i=1, 2 and 3, respectively, obtained for the 2D training data sets shown in Fig. 2.

III. Model Reduction

A. Background and Motivation

Collaborative human-robot systems evolving in complex environments generate very large numbers of parent variables. Typically, the associated level of complexity forbid the generation of BNs. Practical BNs require that some form dimension or variable ranking and reduction is performed before it can be learnt. Figure 4 shows three possible ways this can be achieved. The reduction method used here assumes pre-clustered data, as shown by the shaded areas in Fig. 4.

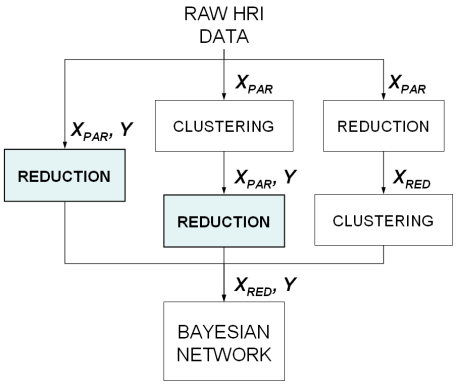


Figure 4. Three scenarios for data reduction prior to learning a Bayesian Network: (*leftmost path*) raw data comes already classified; (*middle path*) data must be classified (manually, or by using a common algorithm such K-means to find natural clusters in the high-dimensional space) prior to reduction; and (*rightmost path*) the number of parent variables is reduced prior to clustering and then uses X_{RED} to cluster into separate events.

Developing a low dimensional representation of data embedded in high dimensional space is a key problem in many fields. Several reduction techniques have been developed for applications ranging from semi-supervised learning and pattern recognition to separating overlapping noisy signals.[23,24] While many of these approaches are statistically sound and commonly used, they do not necessarily allow for a probabilistic graph model to take advantage of the resulting lower-dimensional space. Additionally, the reduced dimensions often have little to no physical meaning and may not provide intuitive insight to the problem.

Common reduction methods include for example, Principal Components Analysis (PCA) which linearly transforms the data onto new orthogonal coordinates such that redundancy and noise can be reduced out,[25] but it is not optimized for class separability and is sensitive to scaling issues.[20] Laplacian Eigenmaps and Diffusion Maps are able to obtain a lower-dimensional embedding of the data onto a nonlinear manifold[23,24,26] and retain natural clusters of the data, but they are also sensitive to scaling and can not be efficiently implemented with sparse data. Here, a variance-ratio reduction method is used to simplify the data used in the Bayesian Network.

B. Variance-Ratio Reduction Method

The aim of the method first introduced in Ref. 11 is to significantly reduce the data dimensionality in order to simplify learning the Bayesian Network by considering only the most influential parent variables for each nodes or events. Here, the 'most influential variables' for a particular event are considered to be those that exhibit highest dependencies from that event compared to other events and maintain natural clusters in the data.

For a given set of data separated into pre-clustered decision modes, the variance ratio is defined as

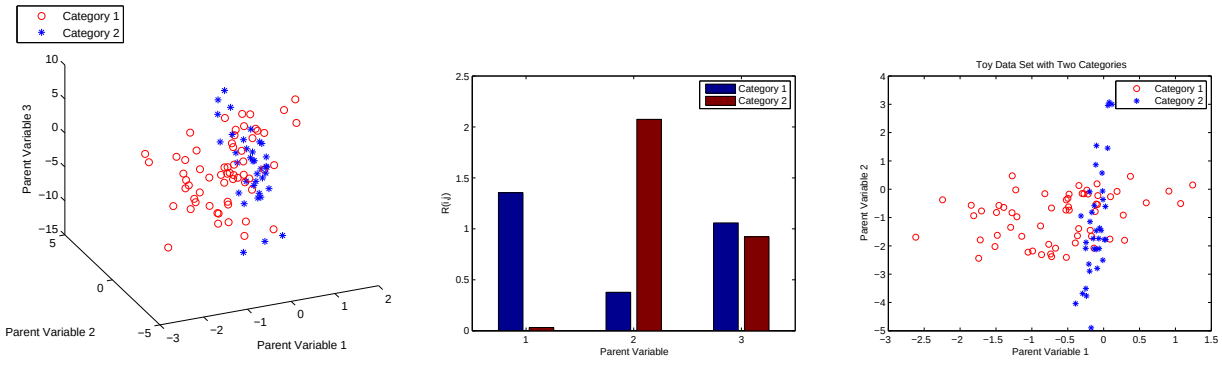
$$R_{i,j} = \frac{\sigma_{i,j}}{\sigma_{all,j}}, \qquad \forall i \in [1, ..., n_D],\ j \in [1, ..., n_X] \tag{6}$$

where n_D and n_X are the numbers of operator decision modes and parent variables, respectively, $\sigma_{all,j}$ represents the variance of all user decisions over the jth variable, and $\sigma_{i,j}$ represents the variance of the ith decision mode over the jth variable.

The ratio $R_{i,j}$ gives an intuitive metric as to how 'dependent' the modes are to the parent variables; if $R_{i,j}$ is small, the dependency of the ith mode on the jth variable is high compared to that of other modes on the same variable; if $R_{i,j}$ is large, the jth variable does not distinguish well the ith mode from the rest of the data. Parent variables associated with lowest values of $R_{i,j}$ can then be kept for use in a Bayesian Network, while those associated with higher values over all modes are ignored. The result is a reduced system model in lower-dimensional space which preserves probabilistic clusters in the data. The advantages to this reduction method over other commonly-used methods are:[11]

- while a large amount of data will improve the accuracy of the algorithm, it can be implemented using sparse data;
- the technique does not transform the original parent variables such that the data lose their physical meaning and intuitive patterns;
- the data does not need to be pre-scaled, i.e. $R_{i,j}$ is unitless and is unaffected by scaling; and
- the reduced system model contains parent variables that can be easily used in a probabilistic graph framework.

Figure 5 illustrates the usefulness of the variance-ratio reduction method on a sample problem. Data from two events (defined as *Category 1* and *2*) are recorded in three-dimensional space as plotted in Fig. 5(a). For each event and parent variable combination, the variance ratio $R_{i,j}$ was calculated and represented by a bar shown in Fig. 5(b) where it is easily observed that *Category 1* data has the highest relative dependency on *Parent Variable 2*, and *Category 2* data has the highest relative dependency on *Parent Variable 1*. Consequently, the third parent variable, since it does not contribute much additional differentiation information, can be eliminated and a Bayesian Network model can be created based on a reduced two-dimensional space. One can observe in Fig. 5(c) that enough data is retained by these two parent variables such that the events are probabilistically distinguishable. This method not only eliminates the need for normalization, but it also reduces out the variables that do not help distinguishing between different categories.



(a) (b) (c)

Figure 5. Data reduction example: (a) Sample data set with three parent variables; (b) Values of $R_{i,j}$ for each category i and parent variable j; and (c) Reduced sample data set with two parent variables.

IV. RoboFlag Experiment

A. Background

The Cornell RoboFlag simulation and experimental testbed[12,27,28] is used to experimentally explore the development and evaluation of realistic solutions for semi-autonomous control of coupled operator-multiple vehicles systems. Many key aspects of future systems can be set up and studied with this testbed, such as: inclusion of a single or multiple human operators, different levels of tasking and autonomy, cooperative planning, and uncertainties associated with incomplete information, latency, and an intelligent adversary.

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 Blue vehicles: two fast moving search vehicles ($SV1$, $SV2$), each with the ability to locate entities (reducing location uncertainty); and one slow moving identification or ID vehicle (IDV), with the ability to both locate entities and identify their type (increase ID probability). During the mission, three Red entities could be encountered: two stationary (flag or robot) targets ($TRGT1$, $TRGT2$) with the ability to tag any Blue vehicles that come in contact with it; and a Chaser vehicle ($CHSR$) designed to chase and tag any Blue vehicle within it's sensor field of view. Upon getting tagged, a Blue team vehicle must automatically return all the way to home base at a very slow pace before being able to resume playing.

When a target is first detected, its location is highly uncertain and is contained inside a large probability circle. The uncertainty radius decreases as more localization measurements are collected by maintaining the target within the sensor field-of-view (as depicted in Fig. 6) of one or more Blue team vehicles. The rate of decrease is approximately exponential and mimics traditional estimation/tracking software. Search and ID vehicles can cooperate and fuse sensory information in order to improve information collection. The ID probability is given by a bar on the right side of the GUI. Only the ID vehicle can collect identity information and move the probability from its initial, a priori value. The ID probability improves more rapidly if the location uncertainty is small. When users are confident of the final target type, the user chooses (flag or robot) formally using a GUI input, as shown in Fig. 6. Once both targets have been localized and identified by the user, the user may then terminate the game by pressing the "Finish" button and the final time gets recorded.

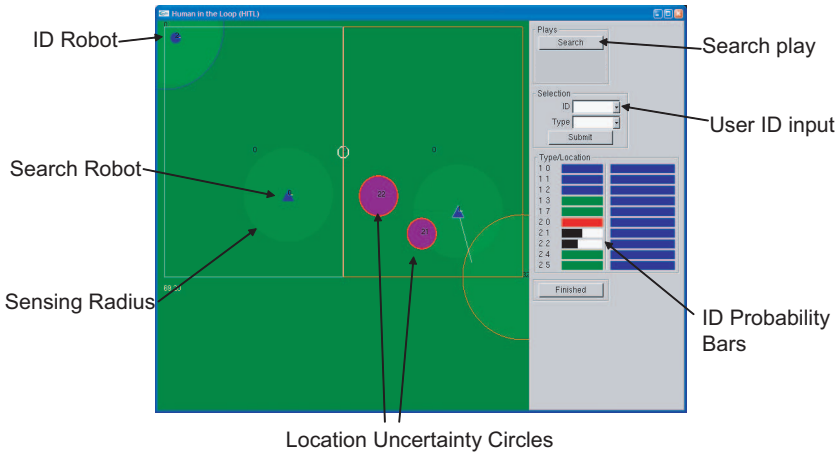


Figure 6. RoboFlag GUI for the first set of games implemented at AFRL/HECP.

C. Experiment

Experiments were conducted at the Air Force Research Laboratory (AFRL). Sixteen subjects were selected from the AFRL subject pool. 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. Target combination (i.e., 2 robots/0 flag, 1 robot/1 flag, and 0 robot/2 flags)

Data recorded included all robot telemetry, users clicks (saved as "events"), and digital videos of the game display and subject for each trial. Each event was assumed to be a new decision or the continuation of a current decision by the user. Four additional trials were added after the original 16, where users were asked to "describe" their decisions, actions, and strategies during the games. These four trials had the same initial conditions as for the first four trials.

In this first set of Roboflag experiments, users controlled the robots by first selecting a vehicle, i.e. left-clicking on it, followed by a single way point allocation, i.e. right-clicking at the desired destination[a]. As soon as the way point is attributed, the robot immediately proceeds towards it until it reaches destination, at which point it switches into a loitering mode. It is possible for the user to re-allocate a different destination to the same robot multiple times while it is selected. It is the responsibility of the user to avoid collision with the enemy entities and evade the Chaser. There is no form of automation, apart from the automatic "Go Home" move when a vehicle does get tagged, and the "Loiter" mode when a vehicle has reached its destination before being re-tasked. A typical cooperative behavior elicited by the users would involve sending two search vehicles 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. Another behavior which proved very useful to avoid the ID vehicle getting tagged was to use one the search vehicles to seek out the enemy Chaser and decoy it away so the slower ID vehicle could then safely be sent in to identify a target.

V. Experimental Results

A. Coupled Operator-Vehicle Decision Modeling

For the search-and-identify RoboFlag scenario described in the previous section, the state of the world may be represented by the following primary state vector

$$X = [X_{SV1}, X_{SV2}, X_{IDV}, X_{TRGT1}, X_{TRGT2}, X_{CHSR}] \tag{7}$$

which is composed of the partial state vectors for the two search vehicles, X_{SV1} and X_{SV2}, the ID vehicle, X_{IDV}, the two targets, X_{TRGT1} and X_{TRGT2}, and the Chaser, X_{CHSR}. These partial state vectors are in turn decomposed as in the following:

$$\begin{aligned}
X_{SV1} &= [x, y, selected, tagged, mode] \\
X_{SV2} &= [x, y, selected, tagged, mode] \\
X_{IDV} &= [x, y, selected, tagged, mode] \\
X_{TRGT1} &= [x, y, detected, uncLoc, ID, uncID] \\
X_{TRGT2} &= [x, y, detected, uncLoc, ID, uncID] \\
X_{CHSR} &= [x, y, detected, uncLoc]
\end{aligned}$$

where x and y are continuous coordinate variables; *selected* and *tagged* are boolean indicating whether the vehicle is currently "tagged" or "selected" by the user, respectively; *mode* is a discrete multi-modal variable representing the vehicle strategic mode of behaviors; *detected* is a boolean to specify whether a red team entity has been detected and is currently visible; *uncLoc* is its location uncertainty radius; ID is a target entity type, i.e. flag or robot, and *uncID* is its ID probability. There are a total 31 primary variables affecting the operator's decisions, all of which were recorded in the telemetry data with the exception of the hidden *mode* variables. However, the operator decisions were often made based on secondary variables made of linear and non-linear combinations of these primary variables. For instance, the decision to evade the Chaser is often made when the Chaser comes within a certain range from the vehicle, namely $r2CHSR_{SelVeh}$ in Table 2), irrespective of the actual x and y location of the vehicles within the enemy zone.

While a very large number of possible secondary variables exist, one may be able to use prior knowledge of the system to determine the most probable subset of secondary variables. One can also use several linear or non-linear reduction techniques to extract underlying embedding of data (briefly discussed in Sec. III). For the Roboflag experiment, a total of 106 primary and secondary variables were considered, clearly illustrating the need for model reduction. Using the variance-ratio reduction method, the 106-state vector has been

[a]Note: It was also possible to first select a "search play" button and then provide a series of way points in a single allocation, but that option was very seldom used.

reduced to one which contains only 14 state variables, listed in Table 2, which are better correlated with individual decision modes than with the entire data set.

Given the world state X, we assume four hierarchical levels of control/decision making, namely:

- a team coordination decision layer denoted by U_{coord}, representing the discrete coordination decisions made by the operator when tasking his team of robots, e.g. tasking a search vehicle to decoy the Chaser in order to allow the ID vehicle to identify a target;
- a strategic control layer U_{strat} representing the vehicles' strategic modes given U_{coord} and X;
- a tactical control layer U_{tact} representing the vehicles' control input given U_{strat} and X, e.g. each vehicle's waypoint allocation to fulfill its strategic assignment; and
- a low-level vehicle control layer U_{veh} representing the low-level actuation necessary for each vehicle to fulfill its tactical control input, e.g. described as a deterministic function of the tactical waypoint assignment.

Depending on a semi-autonomous system's level of sophistication, the decisions made within each control layer may either be made by a human operator, or relegated to the autonomy in either a fixed, or an adaptive manner.

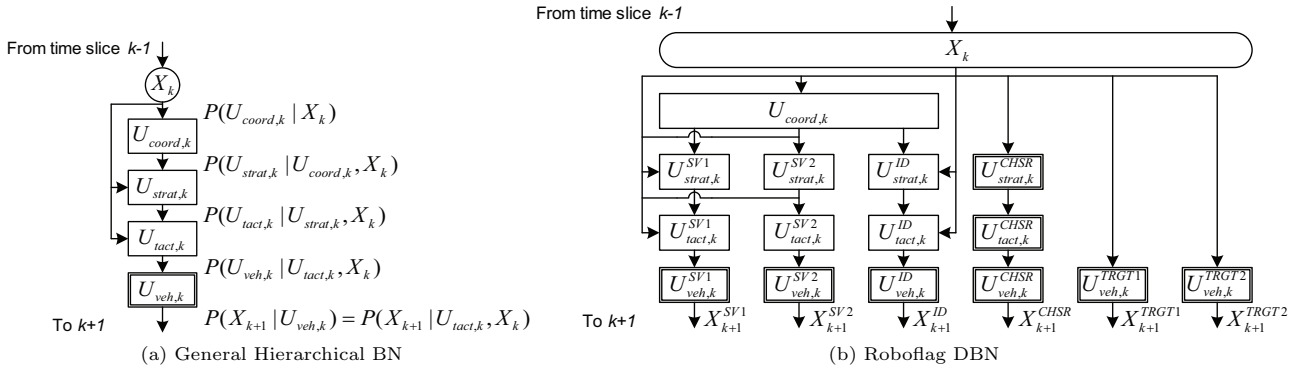
Figure 7(a) shows one time slice of the a general hierarchical DBN model for collaborative human-robot systems. Figure 7(b) shows the actual expanded network specific for the Roboflag experiment. Rectangular nodes indicate random variables with a discrete number of outputs, while circular nodes denote continuous random variables. A rectangular node with double borders represents a deterministic function on a random variable with an output probability defined as 1 given an input. Following the conventional Bayesian Network notation, an arrow from parent nodes l and m to child node n denotes the conditional probability $P(n|l,m)$. Hence, if two nodes are not directly connected in the DBN by an arrow, they are assumed to be conditionally independent given each node's parents. The top node represents the combined state vector at time step k,

$$X_k = [X_{SV1,k}, X_{SV2,k}, X_{IDV,k}, X_{TRGT1,k}, X_{TRGT2,k}, X_{CHSR,k}]$$

while the nodes in the coordination, strategic, tactical and low-level control layers at time step k are denoted

$$\begin{aligned} U_{coord,k} &= [U^{SV1}_{coord,k}, U^{SV2}_{coord,k}, U^{ID}_{coord,k}], \\ U_{strat,k} &= [U^{SV1}_{strat,k}, U^{SV2}_{strat,k}, U^{ID}_{strat,k}], \\ U_{tact,k} &= [U^{SV1}_{tact,k}, U^{SV2}_{tact,k}, U^{ID}_{tact,k}, U^{CHSR}_{tact,k}], \text{ and} \\ U_{veh,k} &= [U_{SV1,k}, U_{SV2,k}, U_{ID,k}, U_{CHSR,k}, U_{TRGT1,k}, U_{TRGT2,k}], \end{aligned}$$

respectively.



(a) General Hierarchical BN (b) Roboflag DBN

Figure 7. Hierarchical DBN model at time step k: (a) General DBN of coupled operator-vehicles team; and (b) detailed expanded DBN of the Roboflag experiement.

Inference from the DBN can be accomplished at each time slice k using marginal probabilities obtained from the joint distribution of the network. In order to compute the joint distribution, we only need to consider the conditional probabilities associated with each node. An interesting point which complicates the human decision modeling from the first Roboflag experiment data is that the strategic control layer, U_{strat}, is hidden in the operator's mind and is therefore not directly observable. The strategic decision had to be inferred by an external observer from the world state and the tactical move by the operator.

B. Strategic Level Human Decisions

Table 1 describes the list of the strategic level decisions made by the human operators when tasking the robots.

Table 1. Definition of the human operator strategic level decisions/actions when allocating tasks to vehicle.

Decisions:	Definitions:
StrategicPos (SP)	Strategic positioning: Positioning of an asset so it is ready to use later, e.g. usually the first moves before starting target search.
SaftyZone (SZ)	Move to Safety Zone.
ID/Localiz (IDL)	Positioning to ID and/or Localize one of the Targets.
SearchTarget (ST)	Searching for either or both Targets.
SearchChaser (SC)	Searching for Chaser.
Escort (ES)	Escortint 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 in anticipation of decoying the Chaser. Note: This move is different than "SearchChaser" and "Decoy" which are used to actively find and decoy the Chaser.
Evade (EV)	Evading, i.e. getting away from Chaser to prevent being tagged.
Decoy (D)	Attracting the Chaser away from the ID Vehicle or keeping it distracted by intentionally getting chased.
Avoid (A)	Immediate reaction to Avoid collision with either a Target or Chaser upon detection or when it suddenly enters enters the vehicle's field-of-view.
SelectionError (SE)	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.

Notice that in this set of experiments, the operator's strategic decisions defined in Table 1 are actually hidden. Unless a subject specifically states what his intention was when tasking a robot, e.g. during the video recording, there are no direct means of determining the true decision with certainty.

A solution to this problem, for application in which the strategic layer node description is not required, is to hide the strategic level decisions into the coordination layer and refer to it by $U'_{strat} = \{U_{coord}, U_{strat}\}$ as illustrated in Fig. 8a. A drawback from using this approach is that the new combined layer may become more complicated and much more difficult to learn than the alternative two separate decision layers. It may also be more difficult to interpret.

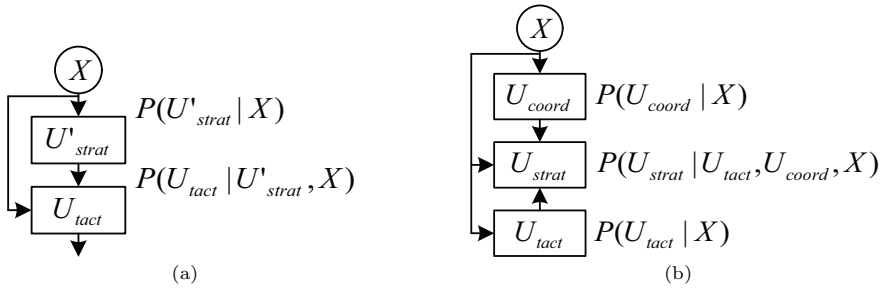


Figure 8. Roboflag coupled operator-vehicles team high-level BN for strategic level inference: (a) Strategic layer hidden in the coordination layer, $U'_{strat} = \{U_{coord,U_{strat}}\}$; (b) Arc reversal inference.

Another alternative is to probabilistically infer the strategic decisions based on the parent nodes, i.e. the current state of the world X at the time of the event and the coordination input U_{coord}, and the child node, i.e. the actual tactical control input U_{tact} following the a strategic decision, such as the waypoint assignment when the mouse event was recorded. To do so, as illustrated in Fig. 8(b), the natural arrow direction from child node U_{tact} to parent node U_{strat} must be reversed by applying Bayes' rule in reverse. This is often

referred to as arc reversal.[29–31]

The equivalent of an arc reversal operation was performed manually on the Roboflag data by replaying each trial data and prompting a human observer to interpret the player's decision at every recorded event. This provided an initial data classification but also introduced new uncertainty associated with different observers' interpretations of the data , which may or may not accurately reflect the players' true intentions. Fortunately, the BN framework allows for the probabilistic model of the human decision-interpreter, obtained from multiple users' interpretations of the same decision data, to be integrated as an extra sensor node in the network. However, the generation of such models is out of the scope of this paper.

Figure 9 illustrates the total number of times each of the actions described in Table 1 were used based on the interpretation made by a single human observer. Unsurprisingly, the operators spent a significant amount of time searching for targets and positioning their robots either strategically for later use, or for IDing/Localizing the targets. Another play that consumed 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 identify 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 close enough to the Chaser to stay in its sensor field-of-view and entice it to give chase, but far enough to avoid being tagged. In fact the total combined number of events recorded for these two plays is much larger than any of the other events.

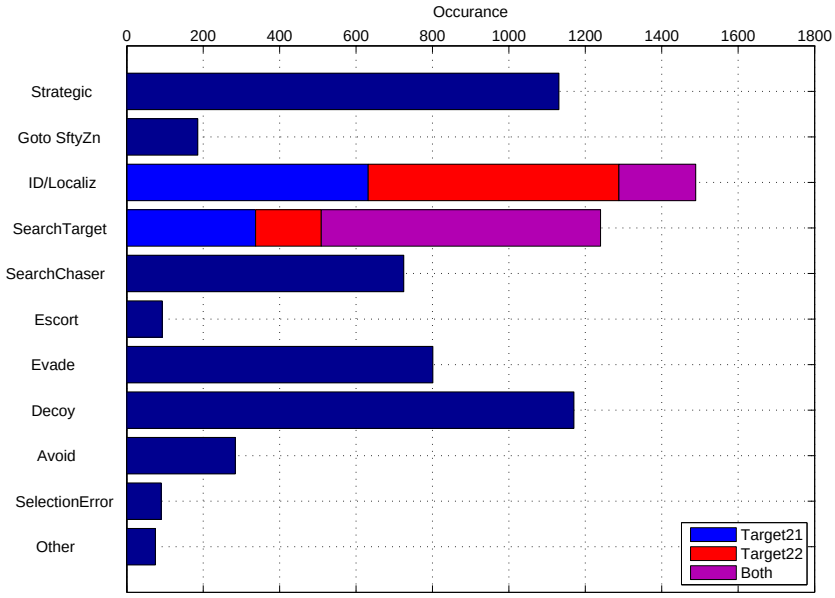


Figure 9. Frequency of human strategic decisions defined in Table 1. Note that these were compiled by human observers whose interpretations are subject to uncertainty.

Table 2 lists the three most 'significant' reduced state variables on which each of the strategic level operator decisions defined in Table 1 are conditioned. The numbers in the table, for each decision considered, indicate the rank of the top three significant variables and correspond to the increasing order of the lowest three variance ratios $R_{i,j}$'s (Equation 6) obtained from performing the data reduction. As with all model identification procedures, the number of parent variables used for model fitting is a user/application dependent choice. It is obtained here by selecting only those variables associated with the lowest three values of $R_{i,j}$ for each decision.

For demonstration purposes, some strategic decision modeling results illustrating how the 16 human subjects tasked their Search and ID vehicles are shown in the following three subsections. These decisions are based on the state dependency matrix of Table 2. In subsection 1, a single parent variable is sufficient to distinguish between two different strategic modes. On the other hand, subsections 2 and 3 show that more than one variable are often required. For simplicity, it was assumed here that the operator was using a single vehicle. The corresponding BN representation for these decisions is shown in Fig. 10.

Table 2. Top three most significant state from the reduced state vector (*vertical*) for conditioning each operator strategic level decision (*horizontal*) as defined in Table 1.

Reduced State Variables, X:	Strategic Level Decisions, U_{Strat}:									
	SP	SZ	IDL	ST	SC	ES	EV	D	A	SE
Elapsed game time [s]				3						
y_{CHSR}	3									2
$uncLoc_{CHSR}$	1			1	1		3			
$uncLoc_{TRGT1}$					3	1				
$uncID_{TRGT1}$				2						
IDV r2 closest $TRGT$										3
$SelVeh$ r2 closest $TRGT$		2					2	2	1	
$SelVeh$ r2 furthest $TRGT$								3		
$SelVeh$ r2 $CHSR$		1	3		2		1	1	2	
IDV's WPr2 furthest $TRGT$										1
IDV's WPr2 $CHSR$			1						3	
$uncLoc$ of $TRGT$ closest 2 $SelVeh$		3	2			2				
$uncLoc$ of $TRGT$ furthest 2 $SelVeh$						3				
$uncID$ of $TRGT$ furthest 2 $SelVeh$	2									

Abbreviation Keys:			
CHSR	Chaser	TRGT	Target
IDV	ID Vehicle	SelVeh	Selected Vehicle
uncLoc	Location uncertain radius	uncID	ID probability
2	to	r2	range/distance to
WP	Waypoint allocated	WPr2	range from WP to

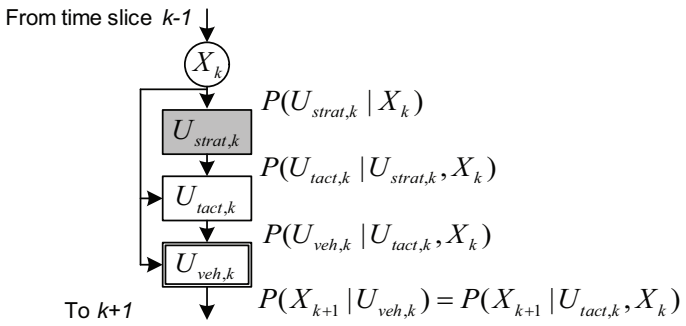
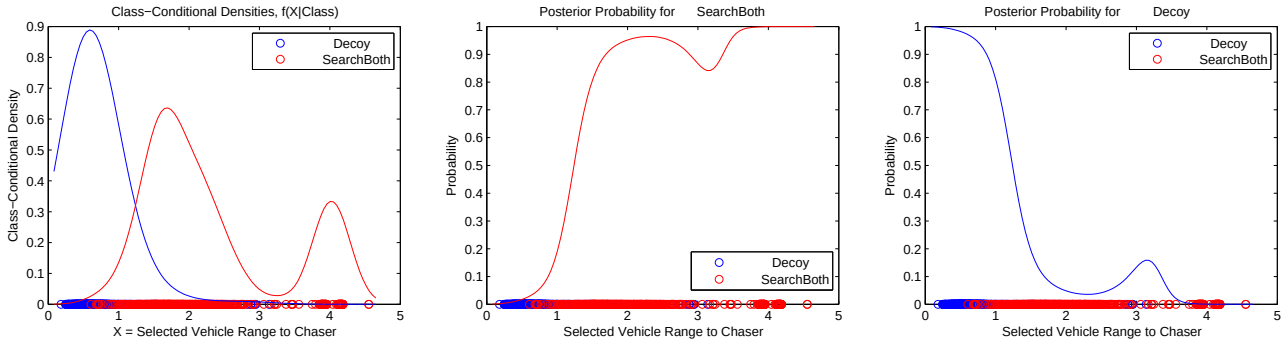


Figure 10. Strategic decision node (*shaded*) in a hierarchical graph model for a single vehicle).

1. Case 1: Search vs. Decoy in 1D

Figure 11(a) shows decision data and Parzen class-conditional density estimates for Decoy and Search for Target plotted against the selected vehicle's Range to the Chaser. Figs. 11(b) and 11(c) show the corresponding non-parametric decision probability likelihoods, $P(U_{strat} = Search|X)$ and $P(U_{strat} = Decoy|X)$. Several interesting facts are illustrated by these plots. Firstly, users very rarely chose to Search for Target if the Chaser vehicle came too close. This decision was most often made at the beginning of the game, when the Blue vehicles were furthest from the enemy zone, or in the middle of the game so long as the Chaser was at a safe distance. In contrast, a Decoy move requires that the selected vehicle stay near enough to the Chaser to keep it within sensor range, but far enough to avoid being tagged. This relative range of motion is very limited, causing the variance of Decoy in the variable Range to Chaser very small compared to that of Search for Target.



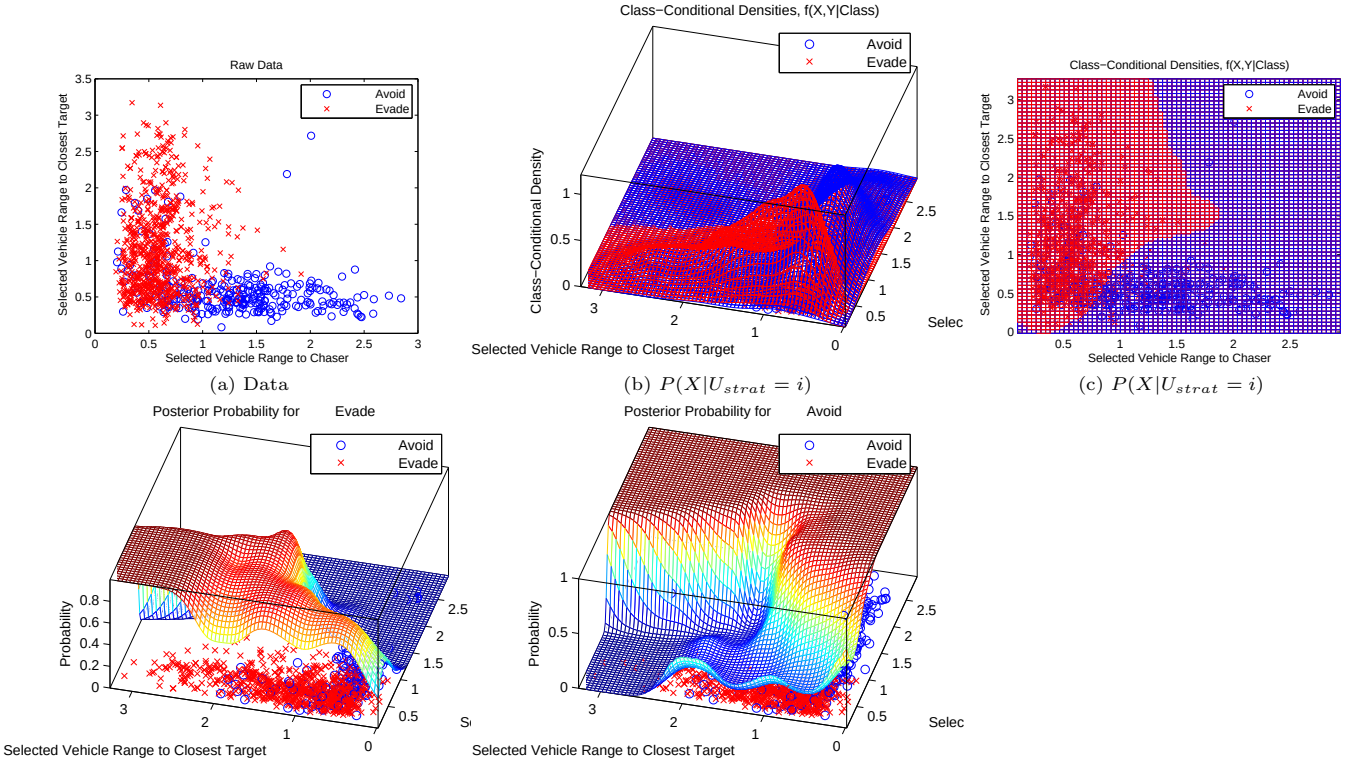
(a) $P(X|U_{strat} = i)$ (b) $P(U_{strat} = Search|X)$ (c) $P(U_{strat} = Decoy|X)$

Figure 11. Strategic decision data for $U_{strat} \in$ {Search, Decoy} and reduced model in one dimension X={SelVeh range to CHSR ($r2CHSR_{SelVeh}$)}: (a) decision data in 1D with Parzen conditional PDF estimates $P(X|U_{strat} = i)$ for $i \in \{Search, Decoy\}$; (b) and (c) non-parametric decision likelihoods $P(U_{strat} = i|X)$ for decisions $i = Search$ and $Decoy$, respectively.

Notice that in this example the decision likelihoods are sharp even considering the low dimensional representation. This indicates that the two decisions are easily distinguishable as they are well separated from each other. However, other decision data may overlap with these two decision in this one-dimensional space creating the need for additional dimensions. *Case 3* illustrates this point with the inclusion of a third decision data set.

2. Case 2: Evade vs. Avoid in 2D

Figure 12(a) shows decision data for Evade and Avoid plotted against the Range to Closest Target and the Range to Chaser for the selected vehicle. The Parzen class-conditional density estimates for these



(a) Data (b) $P(X|U_{strat} = i)$ (c) $P(X|U_{strat} = i)$

(d) $P(U_{strat} = Evade|X)$ (e) $P(U_{strat} = Avoid|X)$

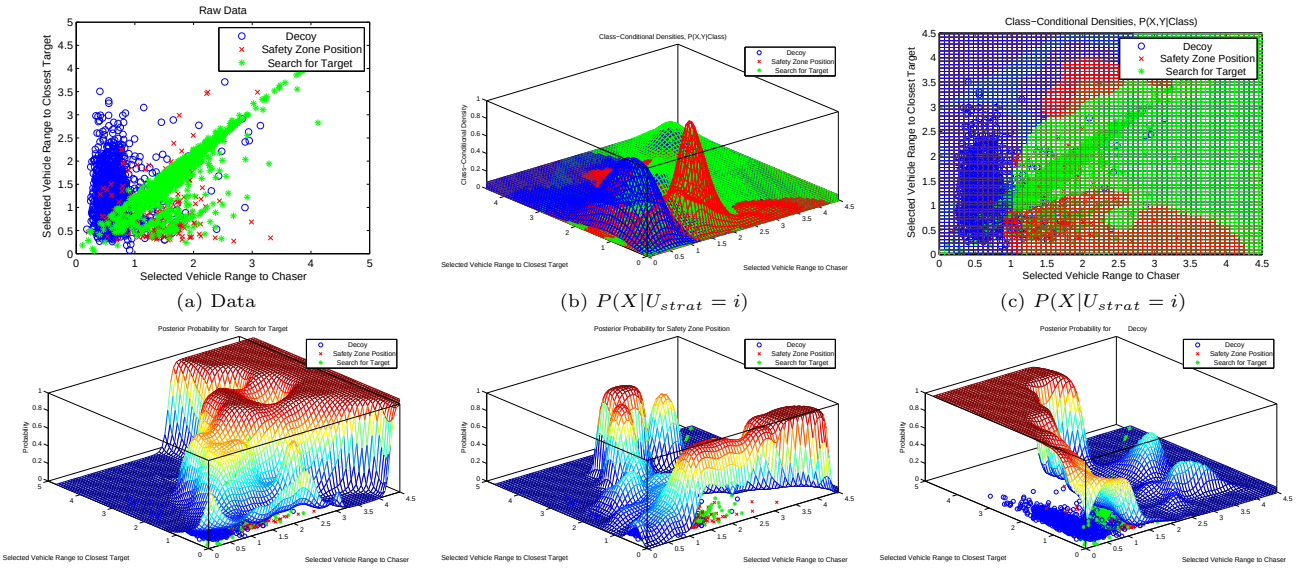
Figure 12. Strategic decision data for $U_{strat} \in$ {Evade, Avoid} and reduced model in dimensions X={SelVeh range to CHSR ($r2CHSR_{SelVeh}$), SelVeh range to closest TRGT ($r2closestTRGT_{SelVeh}$)}: (a) scatter plot of decision data on reduced dimensions; (b) Parzen conditional PDF estimates $P(X|U_{strat} = i)$ for $i \in \{Evade, Avoid\}$; (c) top view of the Parzen PDFs showing the deterministic decision boundaries; (d) and (e) non-parametric decision likelihoods $P(U_{strat} = i|X)$ for decisions $i = Evade$ and $Decoy$, respectively.

decisions are shown in in both Figs. 12(b) and 12(c). The resulting deterministic decision boundaries resulting from these estimates (see Equation 5) can clearly be seen on Fig. 12(c). Figs 12(d) and 12(e) show the corresponding decision likelihood surfaces (see Equation 3). For the Evade decision, users tried to flee the Chaser vehicle when it came within their sensor range, regardless of their range to the nearest target. Similarly, users chose to Avoid a target when it became suddenly visible along a vehicle's path, regardless of the vehicle's range to Chaser.

In this example, considering two parent variables instead of a single one is necessary to better separate the two decisions data. As shown in Fig. 12(a) the choice of Range to Closest Target and Range to Chaser is ideal to separate this data as each decision seems to be align along one of the two dimensions. As a result, Figs. 12(d) and 12(e) show sharp decision boundaries between the Evade and Avoid moves. Although in this case two parent variables seem to be sufficient, considering a third variable may help to distinguish the part of the data near the origin where there is significant overlap. i.e. where both the Rage to Closest Target and the Range to Chaser are small.

3. Case 3: Search vs. Go to Safety Zone vs. Decoy in 2D

Figure 13(a) shows decision data for Decoy, Go to Safety Zone, and Search for Target plotted against the Range to the Closest Target and the Range to the Chaser for the selected vehicle. The Parzen class-conditional density estimates in Figs. 13(b) and 13(c) show the principal clusters for the three decisions, while Figs. 13(d)-(f) show the corresponding decision probability surfaces.



(a) Data (b) $P(X|U_{strat} = i)$ (c) $P(X|U_{strat} = i)$

(d) $P(U_{strat} = Search|X)$ (e) $P(U_{strat} = SftyZn|X)$ (f) $P(U_{strat} = Decoy|X)$

Figure 13. Strategic decision data for $U_{strat} \in$ {Search, SafetyZone, Decoy} and reduced model in dimensions X={SelVeh range to CHSR ($r2CHSR_{SelVeh}$), SelVeh range to closest TRGT ($r2closestTRGT_{SelVeh}$)}: (a) scatter plot of decision data on reduced dimensions; (b) Parzen conditional PDF estimates $P(X|U_{strat} = i)$ for $i \in \{Search, SftyZn, Decoy\}$; (c) top view of the Parzen PDFs showing the deterministic decision boundaries; (d)–(f) non-parametric decision likelihoods $P(U_{strat} = i|X)$ for decisions $i = Search$, $SftyZn$ and $Decoy$, respectively.

It is clear from these plots that vehicles are more likely to Decoy when they are close to the Chaser, which agrees with intuition. It is also not surprising that vehicles are more likely to be sent back to the Safety Zone once the distance to the closest target becomes small, since this indicates that either a target's location or ID was ascertained successfully by moving as close to it as possible. Since the Search for target data set exhibits high variance in both the Target and Chaser Range dimensions, it overlaps significantly with the data for the other two decisions, which makes probabilistic decision discrimination quite difficult. However, it is still clear from Fig. 13 that when the distance to both Chaser and Target are large (i.e. when the Chaser and Targets are not within vehicle sensing range), the user is very likely to task the vehicle to search for targets. In this case a third parent state variable would improve separation among the three decisions and therefore lead to sharper decision probabilities.

C. Tactical Level Human Decisions

Tactical moves, i.e. waypoint assignments, vary greatly depending on the strategic intentions of the operator. However, as depicted in Fig. 14, given a strategic task assignment U_{strat} to a particular vehicle and given the world state X, it is possible to obtain from the data a BN node to model the tactical decision layer U_{tact}. The shaded node on Fig. 14 represents the tactical moves for a single vehicle given the strategic decision of searching for targets. Figure 15 shows an example of how to obtain the probabilistic model for this node using non-parametric Parzen density estimation. $P(U_{tact}U_{strat}, X)$.

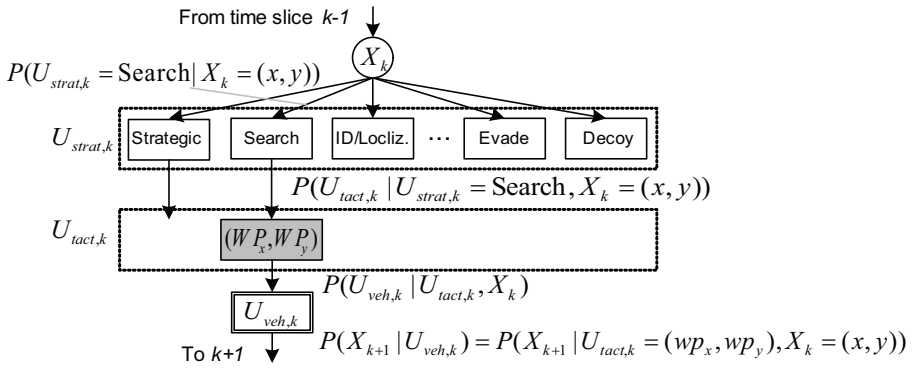


Figure 14. Tactical Search decision node (*shaded*) in a hierarchical graph model for a single vehicle.

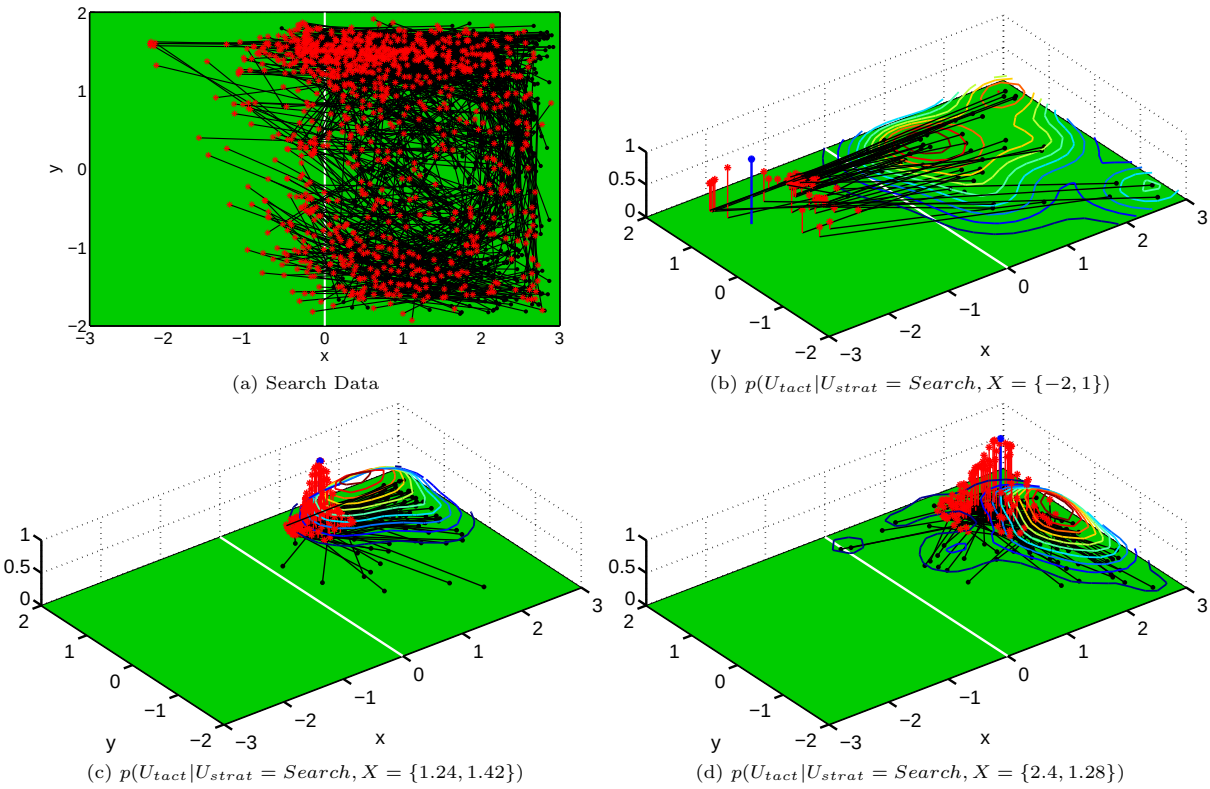
Figure 15(a) illustrates the entire set of tactical search data obtained for the first set of Roboflag experiments, i.e. vehicle locations at time of the event (*red stars*), waypoint assignments (*black dots*), and corresponding trajectories (*black lines*). As a modeling example, Figs. 15(b) to 15(d) show a series of three contours of the PDF representing the tactical waypoint assignments, $p(U_{tact}|U_{strat} = Search, X = \{x_{veh}, y_{veh}\})$, given the vehicle is in search mode and given the state of the world, $X = \{x_{veh}, y_{veh}\}$ at three consecutive locations, i.e. {-2,1}, {1.24,1.42} and {2.4,1.28} denoted by (*blue stars*). In this case, the reduced set of state variables chosen x_{veh} and y_{veh} represent the vehicle's location on the field. As can be seen from Fig. 15 the choice of these variables makes sense for an illustrative example as they are intuitive to a human observer. On the other hand, applying the model reduction technique could result in a better choice of variables for both the reduced state of the world X and the waypoint destination U_{tact}, e.g. relative range and bearing.

In this example, the PDF $p(U_{tact}|U_{strat} = search, X = \{x_{veh}, y_{veh}\})$, where the coordinates $\{x_{veh}, y_{veh}\}$ are represented by a *blue star* in Figs. 15(b) to 15(d), is obtained by performing a weighted sum of Gaussians over each tactical data p

$$p(U_{tact} = \{x_{tact}, y_{tact}\}|U_{strat} = search, X = \{x_{veh}, y_{veh}\}) = C \cdot \sum_{p=1}^{n_p} \gamma_p \mathcal{N}_p(\{x_{tact_p}, y_{tact_p}\}, \sigma), \qquad (8)$$

where the weigths $\gamma_p = \frac{1}{2\pi\sigma_\gamma^2} \exp^{-\frac{(x_p - x_{veh})^2 + (y_p - y_{veh})^2}{2\sigma_\gamma^2}}$ decrease exponentially with the range to the vehicle location, σ_γ determines the radius of influence, each Gaussian $\mathcal{N}_p = \frac{1}{2\pi\sigma^2} \exp^{-\frac{(x_{tact} - x_{tact_p})^2 + (y_{tact} - y_{tact_p})^2}{2\sigma^2}}$ is centered around the corresponding assigned waypoint for a data point p, σ is specifies the standard deviation, and C is a normalizing constant such that the integration of the probability density function sums to 1. Both $sigma$ and $sigma_\gamma$ may be determined adaptively based on the local density of data.

Using the above technique, it is possible to compute on-line, from the stored data, the corresponding tactical waypoint assignment PDF for any given state, given the strategic level decision. To reduce memory usage and increase the speed of the online computations the large data set stored may be drastically reduced using Girolami's data condensation technique.[21] It should be noticed that the PDFs may also be precomputed and tabulated on a discretized grid for quick reference. These PDFs may then be used in prediction mode to generate decisions automatically using sampling techniques. In fact, this is illustrated in the sequence from Figs. 15(b) to 15(d). The initial location for the vehicle in Fig. 15(b) was selected arbitrarily, but the subsequent initial vehicle location (given state) in Fig. 15(c) was a random sample from the PDF represented by the contours in Fig. 15(b). This is also true for the initial location in Fig. 15(d) which was sampled from the PDF in Fig. 15(c). Work is underway to combine the decision models from each hierarchical layer to



(a) Search Data

(b) $p(U_{tact}|U_{strat} = Search, X = \{-2, 1\})$

(c) $p(U_{tact}|U_{strat} = Search, X = \{1.24, 1.42\})$

(d) $p(U_{tact}|U_{strat} = Search, X = \{2.4, 1.28\})$

Figure 15. Search tactical data and modeling: example $p(U_{tact}|U_{strat}, X)$: (a) Vehicle location at the time of the event data (*red stars*) and corresponding tactical waypoint assignments (*black dots*) given the strategic decision of searching for targets; (b)–(d) Contours of the probabilistic tactical decision models $p(U_{tact}|U_{strat}, X)$ obtained for a vehicle in search mode $U_{strat} = Search$ and located (*blue star*) at three consecutive places $X = \{x_{veh} = -2, y_{veh} = 1\}$, $X = \{x_{veh} = 1.24, y_{veh} = 1.42\}$ and $X = \{x_{veh} = 2.4, y_{veh} = 1.28\}$, respectively.

create a complete probabilistic human operator emulator based on the Roboflag data collected. This model combined with system performance metrics will then be used to to optimize the human-robots interactions in coupled systems with adaptive autonomy.

VI. 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. Discrete operator decisions are modeled as nodes in a hierarchical BN, with conditional dependencies on the state of the world and the output from the parent node in the previous decision layer. Non-parametric decision models based on data are obtained using a method based Parzen density estimation. Data was collected in a series of tests using a multiple robot adversarial game simulator. The theory is applied to operator decision data and probabilistic decision models are generated for both the strategic and the tactical hierarchical layers. Work is underway to combine the decision models from the different hierarchical layers to achieve a complete human-robot system BN model based on the data. Such models will be used to study adaptive autonomy, human-robot interface optimization, and the implementation of automatic control of human level capability. They may also be generated from observations to model adversaries' strategies, or the behaviors of other types of object of interest. Hence, the general probabilistic modeling framework promises to impact various complex and/or adversarial applications requiring adaptive planning.

The ongoing research effort includes:

- *Higher order decision models*, i.e. when decisions are clustered in two distinct subspaces, will be explored. Approaches include clustering of decision data and mixtures of Gaussians for automatic identification.

- *Recursive models* will be developed to include time dependency while maintaining statistical rigor of the framework. 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.
- *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 a likelihood ranking.
- *Performance Metrics* and estimation methods to gauge performance level and identify causes for decreased productivity or mission failures in human-robotic systems will be studied. These metrics will be useful as reliable utility measures for optimized adaptive autonomy and for the evaluation of user interfaces, situation awareness, fatigue, and other factors.

Finally, two new RoboFlag tests are currently being implemented. The first aims at studying the use of cooperative vehicle decision aids, and the second focuses on adaptive tasking,[32] where the game will switch between manual and decision aids with varying levels of autonomy in order to maximize performance.

Acknowledgements

Thank you to Dr. Scott Galster from AFRL/Human Effectiveness Directorate who led the data collection experiment using the Roboflag system. 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]T. B. Sheridan, *Telerobotics, Automation, and Supervisory Control*. Cambridge, MA: MIT Press, 1992.

[2]D. E. Kieras and D. Meyer, "Epic: A cognitive architecture for computational modeling of human performance," http://www.eecs.umich.edu/ kieras/epic.html.

[3]D. E. Kieras, D. E. Meyer, J. A. Ballas, and E. J. Lauber, "Modern computational perspectives on executive mental processes and cognitive control. where to from here?" University of Michigan, Tech. Rep. TR-98/ONR-EPIC-12, 1999.

[4]M. Byrne, "The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications," in *Cognitive architecture*, J. Jacko and A. Sears, Eds. Mahwah, NJ: Lawrence Erlbaum, 2003.

[5]——, "Act-r/pm: Act-r perceptual-motor," http://chil.rice.edu/projects/RPM/index.html.

[6]A. Kirlik, "Modeling strategic behavior in human-automation interaction: Why an "aid" can (and should) go unused," *Human Factors*, vol. 35, no. 2, pp. 221–42, 1993.

[7]C. Miller, M. Pelican, and R. Goldman, "Tasking interfaces for flexible interaction with automation: Keeping the operator in control," in *Conference on Human Interaction with Complex Systems*, Urbana-Champaign, IL, 2000.

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

[9]M. Campbell, F. Bourgault, S. Galster, and D. Schneider, "Coupled, probabistic operator-multiple robot models using bayesian networks," in *Proceedings of the 2007 IEEE International Conference on Robotics and Automation (ICRA'07)*, 2007.

[10]N. Ahmed, M. Campbell, and F. Bourgault, "Nonparametric, probabilistic operator decision models," in *submitted to the 2008 IEEE International Conference on Robotics and Automation (ICRA'08)*, 2008.

[11]D. Shah and M. Campbell, "State-dependent probabilistic model reduction for evaluation of human-robotic autonomous systems," in *Proceedings of the 2008 IEEE International Conference on Systems, Man and Cybernetics (SMC'07)*, Montreal, Canada, October 2007.

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

[13]M. Campbell, R. D'Andrea, D. Schneider, A. Chaudhry, S. Waydo, J. Sullivan, J. Veverka, and A. Klochko, "Roboflag games using systems based, hierarchical control," in *American Control Conference*, Denver, CO, 2003.

[14]D. Heckerman, "A tutorial on learning with bayesian networks," Microsoft Corporation, Tech. Rep., 1995, mSR-TR-95-06.

[15]K. Murphy, "Dynamic bayesian networks: Representation, inference and learning," PhD thesis, UC Berkeley, 2002.

[16]Y. Bar-Shalom, X. Rong Li, and K. T., *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*.

[17]G. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*, ser. Wiley Series in Probability and Mathematical Statistics. New York, NY: Wiley-Interscience, 1992.

[18]A. Gray and A. Moore, "Nonparametric density estimation: Toward computational tractability," in *SIAM International Conference on Data Mining*, 2003.

[19]B. Silverman, *Density Estimation for Statistics and Data Analysis*, ser. Monographs on statistics and applied probability; 26. London; New York: Chapman and Hall, 1986.

[20]K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed., ser. Computer Science and Scientific Computing Series. San Diego, CA: Academic Press, 1990.

[21]M. Girolami and C. He, "Probability density estimation from optimally condensed data samples," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 25, no. 10, pp. 1253–1264, 2003.

[22]G. Babich and O. Camps, "Weighted parzen windows for pattern classification," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 18, no. 5, pp. 567–570, 1996.

[23]Y. Bengio, J.-F. Paiement, and P. Vincent, "Out-of-sample extensions for LLE, isomap, MDS, Eigenmaps, and spectral clustering," Département d'Informatique et Rechereche Opérationnelle, Université de Montréal, Tech. Rep. 1238, July 2003.

[24]M. Belkin and P. Niyogi, "Laplacian Eigenmaps for dimensionality reduction and data representation," Dept. Comp. Sci. and Statistics, University of Chicago, Tech. Rep. TR 2002-01, January 2002.

[25]J. Shlens, "A tutorial on Principal Components Analysis," Salk Institute for Biological Studies, University of California, San Diego, Tech. Rep., December 2005.

[26]S. Lafon and A. Lee, "Diffusion maps and coarse-graining: A unifified framework for dimensionality reduction, graph partitioning, and data set parameterization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1393–1403, 2006.

[27]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.

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

[29]R. Howard and J. e. Matheson, *Readings on the Principles and Applications of Decision Analysis.* Menlo Park, CA: Strategic Decisions Group, 1981.

[30]S. Olmsted, "On representing and solving decision problems," PhD thesis, Department of Engineering-Economic Systems, Stanford University, 1983.

[31]R. Shachter, "Probabilistic inference and influence diagrams," *Operations Research*, vol. 36, no. 4, pp. 589–604, 1988.

[32]R. Parasuraman, "Effects of adaptive function allocation on human performance," in *Human Factors and Advanced Aviation Technologies*, D. Garland and J. Wise, Eds. Daytona Beach, FL: ERAU, 1993.