Quantitative Modeling of Multi-Agent Systems

Jason Held and Salah Sukkarieh
ARC Centre of Excellence in Autonomous Systems, University of Sydney, NSW 2006, Australia
j.held@acfr.usyd.edu.au

Abstract

This paper presents a network centric method of modeling multi-agent systems in a decentralized data fusion, feature localization scenario. The robotic multi-agent system is abstracted using a matrix of metric interactions stored within a Dynamic Bayesian Network. This model, called a system map, provides a tool which may be used in automated design of multiple robot systems. Results from Monte Carlo simulations are presented which demonstrate how system maps can be an accurate model for robotic systems.

1 Introduction

Most approaches to modeling multiple robotic systems are based on complexity theory solutions [Daniel Bernstein, 2002][Holland, 1992][Levin, 1998] which consider the formation of a complex dynamic system to be based on a set of simple rules. These are emergent behavior solutions which are valuable in reasoning about non-linear actions and path plans but are not conducive to *a priori* system design. Design methods are fail due to the difficulties in complexity and computation time. What is needed is a multi-agent systems model which quantifies the system without explicitly attempting to model its complex behaviors.

This paper presents a Network Centric Operations (NCO) [Stein, 2000][Evidence Based Research, 2003] approach to modeling multi-agent systems. Assuming that each platform can communicate its status to the rest of the system, it is possible to record system level measurements during a mission. Monte Carlo simulations of a mission are executed for every possible combination of robots in the designer's inventory. Each simulation records measurements (here on called *metrics*) of that system's state over a discrete set of time. Since there is no *a priori* information on metric interactions, a Dynamic Bayesian Network (DBN) [Murphy, 2002][Murphy, 2004][John N. Gowdy, 2004][Fabio Ramos, 2004] is then used to learn this probabilistic relationship over each metric in the system, with the network's parameters representing metric interactions. It is these interactions, recorded in a matrix, which form the mathematical abstraction of system attributes called a *system map*. Mission requirements, when also presented as a system map, can then be compared with the set of possible system maps to determine a feasible configuration.

This paper is organized as follows. Section 2 describes how to determine a system map using DBNs. Section 3 describes the experiments, simulator, and metrics used, and shows the current Monte Carlo approach to learning the potential attributes of the system within a set of system maps. Section 4 shows an analysis of the resulting sets of system maps. Section 5 concludes and discusses the direction of future research.

2 System Maps

A system map is a probabilistic model of a system. In it, the metric interactions are used to quantify system attributes. It is important to note that system attributes may vary widely for a single set of assets and organization. To solve this problem, the system map attempts to capture the potential interaction of system metrics, rather than determination of a system's current state.

2.1 Dynamic Bayesian Networks (DBNs)

DBNs are directed acyclic graphs used here to determine the structure of metric interactions over time in a Markovian fashion. Each metric is a node in the graph storing a (Gaussian) probability distribution function (pdf), with influences between metrics represented as edges. This is estimated for each metric M based on the posterior probability information from the simulation by assuming that the parameters, $P(\theta|M)$, has *Maximum Likelihood* estimates θ_{ML} calculated with the maximization of evidence Z_L as

$$Pr(M|Z_L) \approx \frac{Pr(M)}{Pr(Z_L)} Pr(Z_L|\theta_{ML}, M) Pr(\theta_{ML}|M) \quad (1)$$

and

$$\theta_{ML} = \arg\max[\theta \log(Z_L|\theta)]. \quad (2)$$

This method of learning has been used widely in pattern recognition applications such as speech recognition [John N. Gowdy, 2004], learning genetic interactions [Linus Gransson, 2001], and human figure motion tracking [Vladimir Pavlovic, 1999][Fabio Ramos, 2004]. As such, DBN is a proven method of discovering hidden patterns in structures and can determine system maps even in absence of *a priori* information.

The relationship between each metric M_i to its children is calculated given its parents $Pa(M_i)$, where i is the current of r metrics monitored in the system. The joint probability of this relationship is defined as

$$P(M_1, \ldots, M_r) = \prod_i P(M_i|Pa(M_i)). \quad (3)$$

Computing posterior relationship probabilities is accomplished using an adaptation of work on feature extraction using structure learning algorithms [Fabio Ramos, 2004]. In [Fabio Ramos, 2004], the challenge of modeling systems with non-linear dependencies is tackled through approximating non-linear to linear metric relations, learning a new structure for each discrete time slice. Over the set of time steps, a pattern appears which may be identified and described using a Bayesian network. In [Fabio Ramos, 2004], this approach was demonstrated to successfully identify motion pattern between parts of the human body in motion from a video recording.

2.2 DBN Implementation

The algorithm works as follows. First the structure set of M is learned in the first time step. This is done using a search over all possible combinations of acyclic graphs and monitored by a scoring function. The scoring function chosen by [Fabio Ramos, 2004] was the Bayesian Information Criterion (BIC) [Heckerman, 1996] (also known as the Minimum Description Length, or MDL approach [Suzuki, 1998]). This is

$$BIC(G) = \sum_i \sum_r logP(M_i|Pa(M_i), \theta_i, D^r) - \frac{np_i}{2} logN, \quad (4)$$

where np_i is the number of parameters in the distribution of metrics M and N is the number of samples. This scoring function is then maximized using a greedy search until reaching a local maximum for this first time step.

This procedure is repeated for the remaining time steps, where correlations between systems in consecutive time steps are learned. In [Fabio Ramos, 2004], the procedure to do this is the same as above, with added constraints protecting independence of variables over time. The algorithm continues to iterate over the time samples, learning the structure for each, until new representations converge with the previous time's sample. Convergence is considered effective once the difference between representations falls within Kullback-Leibler divergence [Thomas M. Cover, 1991], also known as *relative entropy*. Using θ to represent the conditional probability distribution of the influence structure $P(Pa(M_i)|M_i)$ for a time step, this divergence is measured as

$$KL(\theta||\widetilde{\theta}) = \sum_M P_\theta(M) ln \frac{P_\theta(M)}{P_{\tilde{\theta}}(M)}. \quad (5)$$

The DBN parameters form the resultant system map for the tested system of robots. This configuration will be recorded with the system map for later use during mission analysis. Results from the DBN provide a function for calculating each metric based on the influences of other metrics. This can be done as a weighted sum over the values for each metric, using the weights w as the influences of other metrics as

$$M_i = \sum_r M_r w_r + \mu_i, \quad (6)$$

with μ_i as the mean for metric M_i.

3 Experimentation

Simulations of two, three, and four UAV systems were conducted in matlab on a 2.4GHz computer. In order to determine system level metrics a software library was created to run in the background, measuring metric values at each time step and recording it in a history. A DBN was learned from that history to determine the resultant system map for each single simulation run.

3.1 The Simulator

The platforms are placed first and fly in a static circular pattern (constant bank angle) until receiving an initial set of point feature location estimations. The features placed have a static mean with high variance. The mission begins with a provided path planner. Sensors localize and reduce the feature uncertainties with noise included in the sensor model. Information is passed every 3 seconds in a Decentralized Data Fusion (DDF) [Hugh Durrant-Whyte, 2001] fashion to other connected platforms in accordance with a predetermined communications link plan. A DDF system consists of a network of sensors which fuse information locally based on individual observations and information communicated from its neighbors in the system. No central processing or fusion occurs. A memory loss function is then applied, causing feature uncertainty to increase at a rate of several meters per second.

The goal of the path planner is to exercise as many metric interactions as possible, in order to capture the full range or system performance and potential. A system taking a static path, for example, may not produce the full range of information accuracy. This was shown to produce interactions which may not fully represent the system's capabilities. One approach to this problem is to apply a dynamic path planner. The dynamic path planner chose paths for each platform individually based on mutual information gain. Platforms flew within the limits of their vehicle model to observe each feature, maximizing information accuracy. With memory loss and the corresponding increase in each feature's uncertainty ellipse, platforms revisit features in an emergent pattern maximizing this metric. Sensor model uncertainties produce different system flight paths with each simulation. Observations of features occur in a periodic and semi-ordered fashion but not all features become localized, as the simulation contains more features than platforms. Sample output for a single simulation is shown in Fig 2.

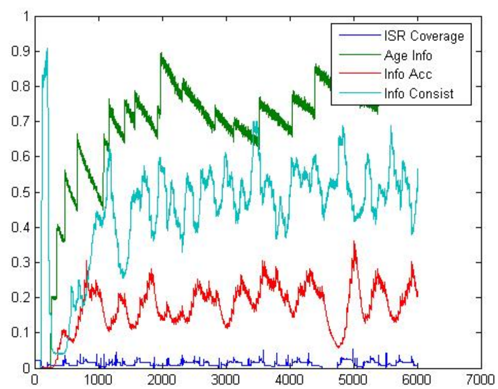

Figure 1: Example output (raw metrics) from one simulation run.

3.2 Metrics

Metrics chosen must be accurate measurements used to define the system. In this case, metrics were chosen to measure system factors which may affect or be affected by changes in DDF. A random noise was introduced to each metric to simulate uncertainty in metric observations. The following metrics were used:

- M_1**: Sensor Coverage.** An individual sensor platform's footprint is calculated geometrically based on the UAV orientation with respect to the ground. Sensing characteristics of the Brumby Mk III's side looking cameras are estimated with a bearings only sensor model. Individual platform sensing results in a irregular quadrilateral polygon whose size varies depending on bank angle. Surface area A for a platform n approaches infinity at zero bank angle, so the sensor is deactivated during straight and level flight. This area is taken from Bretschneider's formula [Weisstein, 1999] using sides a, b, c, d, and intersects p and q as follows:

$$A_n = (1/4)\sqrt{4p^2q^2 - (b^2 + d^2 - a^2 - c^2)^2}. \quad (7)$$

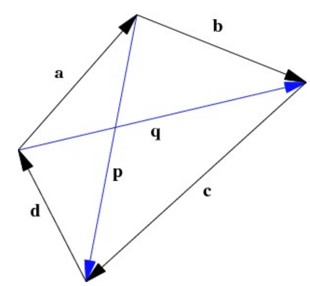

Figure 2: Area footprint approximation.

System coverage is then estimated as the total area of each observation A_n divided by the total possible surface area, a.k.a., the area of operations, A_o, as follows,

$$M_1 = \frac{\sum_n A_n}{nA_o} \quad (8)$$

- M_2**: Information Age.** Features are identified with the current time stamp T at each observation. While a feature is unobserved its age remains at the value of the previously known time step. The mean of all feature times $\hbar$ is used for system level age of information. In this way, the system scores highest values in this metric with the most new information. With n platforms and h features, this is calculated as,

$$M_2 = \sum_n \frac{\hbar}{n * T} \quad (9)$$

- M_3**: Information Accuracy.** Location estimates of features are used to judge the system's information. Accuracy is taken as the determinant of the inverse covariance matrix, Y of each feature, f. Several trial simulations were conducted to experimentally determine a maximum value which can normalize this metric. Each trial used three UAVs flying around three features, with each feature continually within close sensor range (approximately 200 meters).

$$M_3 = \frac{\sum_f ||Y_f^{-1}|| * 1000}{f} / 0.6372 \quad (10)$$

- M_4**: Information Consistency.** A ratio of localized to non-localized features provides information consistency. The system's measure of consistency is the mean of each individual platform's consistency, calculated as,

$$M_4 = \frac{1}{n} \sum_n (\frac{1}{f} \sum_f \frac{a_f}{\bar{a}}), \quad (11)$$

where a is an individual feature accuracy within platform n's database, realized when determining M_3.

3.3 Monte Carlo Simulations

Uncertainties in the sensor model and feature locations cause different paths determined in the dynamic path planner, resulting in a variety of possible metric interactions learnable in the system map. A sample of 50 simulations are conducted for each system map learned, to determine a distribution of possible metric interactions. A mean and variance for each interaction metric is used to determine the quantity and accuracy of the system map fields over n simulations. Understanding the distribution of possible behaviors provides a very clear picture of system performance. Distributions themselves can then be used in comparing different systems.

3.4 Criteria of Evaluation

In order to demonstrate that a system map can be a model for multi-agent systems, results should display the following characteristics.

1. **Consistency.** Assuming that metrics are recorded accurately a system should show consistent interactions for the same system under the same situation. This shows resiliency of the model to noise in the system, regardless of uncertainty in the features, noise in the sensor model, or the method of recording the metrics themselves. Executing two separate Monte Carlo simulation sets for the same system with the same feature locations should result in system map distributions falling within their expected variance.
2. **Stability.** A system is considered stable if one or more solutions exists which allow it to solve a problem [Maria Chli, 2003]. Metric interactions which are unstable do not invalidate the system map, but are considered unusable for the purpose of understanding system ability. Using this definition, unstable systems or conditions should coincide with high variances in the system map. Stable systems, likewise should show lower variances in the system map.
3. **Resilience to Perturbation.** Minor perturbations to the system should result in a likewise minor response in the system map, rather than displaying a radically different system. Changing the sensor model on one of the UAVs in a 3 UAV system, for example, should not drastically change the learned metric interactions but may still show minor identifiable changes. Significant changes, however, such as the inclusion of an additional UAV, should be have a correspondingly significant change to the system map, noticeable in both the mean and variance of metric interaction distributions.

4 Results

4.1 Consistency

System maps were conducted for two identical systems of 3 UAVs executing a localization mission of identical feature locations. Comparing metric distributions between system maps 1 and 2 (see Table I and Table II) show high consistency in M_2, M_3, and M_4 interactions. Mean values are all within two standard deviations of each other and variances are nearly identical, showing that these metric interactions as very accurate representations of the system. M_1 however, shows high variances with its interactions to the rest of the system. Further examining the simulation, Brumby Mk III's side looking cameras only actively sense during banking maneuvers, where the information is most useful (i.e., most likely to observe a feature). This results in far fewer samples than M_2, M_3, and M_4, which are always "active" to learn the relationships.

Figure 3: Brumby Mk III was the model for simulations. Side looking cameras collected data only during banking, causing higher variance in learning sensor coverage metric interactions.

A metric with large variances across all of its interactions shows one scenario of how individual metrics within the system map may be validated in a network centric system engineering scenario. In this case, the variances between M_1 to M_3 and M_4 are consistent (although high) and mean interactions between these three metrics fall within two standard deviations of the corresponding variances. They can be used then to represent system behavior, albeit with larger uncertainty. The M_1

Table 1: System Map 1 Means and Variances

Means	M1	M2	M3	M4
M1	0	1.4902	0.7356	-0.1162
M2	0	0	0.1530	0.3737
M3	0	0	0	-0.2718
M4	0	0	0	0
Variances				
M1	0	1.9719	0.3041	2.2861
M2	0	0	0.0021	0.0104
M3	0	0	0	0.0335
M4	0	0	0	0

Table 2: System Map 2 Means and Variances

Means	M1	M2	M3	M4
M1	0	2.1818	0.5391	0.3756
M2	0	0	0.1537	0.3904
M3	0	0	0	-0.3347
M4	0	0	0	0
Variances				
M1	0	3.6139	0.2664	2.2373
M2	0	0	0.0018	0.0104
M3	0	0	0	0.0407
M4	0	0	0	0

to M_2, connection, however, has variances which are not only large but also inconsistent. Therefore, the M_1 to M_2 specific interaction cannot be used to represent this system.

4.2 Stability.

A known instability in the path planner occurs when the number of platforms increases to four UAVs. A four UAV system showed the system map distribution (Table III) which shows vast increases in M_1 interactions. Interactions between M_2, M_3, and M_4 however, are only slightly increased, showing their resiliency of the representation to specific instabilities. Raw output is shown in Fig. 6 with the resulting system map in Table III. M_1 interactions are no longer valid for the model, although the model itself remains intact.

4.3 Resilience to Perturbation

Comparing mean and variance values for the system maps below show how a relatively minor change in the system asset capability can produce a correspondingly minor adjustment to the representation. Three types of system changes were compared. First Monte Carlo simulations of a 3 UAV system was perturbed by modifying individual sensor capabilities, which were increased one platform at a time from 1000 meters to 3000 meters effective range. Finally, a change in asset is introduced with removal and additions of UAVs.

The results of these changes are shown in Table 4 and 5, below, listed in order of decreasing number of assets and sensor capability.

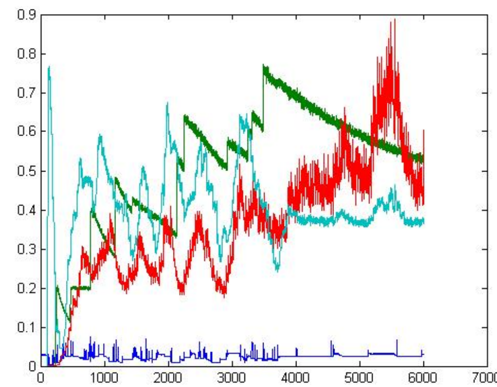

Figure 4: The 4 UAV system did not show the stability of the 3 UAV system. The corresponding system map (Table III) showed how this resulted in the model.

Table 3: 4 UAV System Means and Variances

Means	M1	M2	M3	M4
M1	0	1.3626	3.6062	-2.1089
M2	0	0	0.1641	0.1493
M3	0	0	0	0.2062
M4	0	0	0	0
Variances				
M1	0	17.4208	38.3602	3.1513
M2	0	0	0.0400	0.0835
M3	0	0	0	0.0993
M4	0	0	0	0

5 Conclusion

This paper introduces the system map and demonstrates its use in modeling multiple vehicle systems in dynamic and uncertain environments. It has advantages in that the system model can be learned in a data driven fashion, and its probabilistic representation allows it to fit

Table 4: Mean Metric Interactions Comparison

	M_{12}	M_{13}	M_{14}	M_{23}	M_{24}	M_{34}
4UAV	1.36	3.61	-2.11	0.16	0.15	0.21
3UAV	2.31	0.37	0.63	0.17	0.36	-0.29
3,3,1k	2.48	0.51	0.24	0.16	0.35	-0.26
3,1,1k	2.18	0.54	0.38	0.15	0.39	-0.34
1,1,1k	2.18	0.52	0.16	0.15	0.38	-0.27
2UAV	2.75	-0.09	2.26	0.20	0.37	-0.65
1UAV	3.25	-4.55	2.42	0.16	0.14	-0.45

Table 5: Variance Metric Interactions Comparison

	M_{12}	M_{13}	M_{14}	M_{23}	M_{24}	M_{34}
4uav	17.42	38.36	3.151	0.040	0.084	0.099
3@3k	2.432	0.201	1.563	0.001	0.007	0.045
3,3,1	2.131	0.223	2.674	0.002	0.005	0.028
3,1,1	3.614	0.266	2.237	0.002	0.012	0.041
3@1k	1.868	0.250	2.306	0.001	0.005	0.031
2uav	0.981	0.573	2.306	0.002	0.003	0.007
1uav	7.231	3.800	0.379	0.002	0.001	0.002

well within the realm of currently common tools, such as DBNs. This is an improvement over other multivariate regression techniques, which require additional work to cover cyclic patterns evident in stable dynamic system metrics. Likewise, this method can determine a model for systems even in less stable conditions. The system itself is not defined as unstable, only the portions of the representation (specific metric interactions) which do not solve the mission problem are labeled unstable.

Work here also demonstrates the usefulness of system maps in the analysis of complex systems performance. If a multiple objective mission statement is also presented as a desired system map, an optimal system can be defined as the system map closest to the mission statement. Learning, for example, that the information accuracy (M_3) and information consistency (M_4) interaction becomes positive only when the ratio of UAVs to features becomes close to 1 can be very useful to mission planners tasked at allocating assets.

The only apparent downside to system maps is the requirement for Monte Carlo simulations to determine a distribution of metric interaction. Each system configuration requires a day of simulations to complete, denying the possibility of real time execution on robotic systems in its current form. Real time execution remains a possibility however, when considering that it is the lack of a priori knowledge which requires the use of a Dynamic Bayesian Network in the first place. Determining the network structure is time intensive but once the network structure is determined, realization of parameters can be done in a real time implementation. This is the next logical direction of future research.

References

[Alex Brooks, 2004] Tobias Kaupp-Stefan Williams Hugh DurrantWhyte Alex Brooks, Alexei Makarenko. Implementation of an indoor active sensor network, 2004.

[Daniel Bernstein, 2002] Neil Immerman-Shlomo Zilberstein Daniel Bernstein, Robert Givany. The complexity of decentralized control of markov decision processes. 2002.

[Evidence Based Research, 2003] Inc Evidence Based Research. Network centric operations conceptual framework version 1.0. Technical report, Office of Force Transformation, November, 2003 2003.

[Fabio Ramos, 2004] Hugh Durrant-Whyte Fabio Ramos. Learning complex motion structures, 2004.

[Georgios Chalkiadakis, 2003] Craig Boutilier Georgios Chalkiadakis. Bayesian reinforcement learning for coalition formation under uncertainty, 2003. bayesianCoalForm.pdf.

[GPO, 1997] US Army GPO. *US Army Field Manual (FM) 101-5, Staff Organization and Operations.* Command and General Staff College, Washington, DC, 1997.

[Group, 2004] PA Consulting Group. A network centric operations case study: Us/uk coalition combat operations during operation iraqi freedom. Case Study PAC-0203K, June 23, 2004 2004.

[Heckerman, 1996] David Heckerman. A tutorial on learning with bayesian networks. Technical Report MSR-TR-95-06, The Microsoft Corporation, 1996.

[Holland, 1992] J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.* MIT Press, Cambridge, MA, 1992.

[Hugh Durrant-Whyte, 2001] Mike Stevens Hugh Durrant-Whyte. Data fusion in decentralised sensing networks. Technical report, Australian Centre for Field Robotics, The University of Sydney NSW 2006, 2001.

[John N. Gowdy, 2004] Chris Bartels Jeff Bilmes John N. Gowdy, Amarnag Subramanya. Dbn based multi-stream models for audio-visual speech recognition. Technical report, Clemson University, Clemson, SC, 29634, University of Washington, Seattle, WA, 98195., 2004.

[Levin, 1998] S. A. Levin. Ecosystems and the biosphere as complex adaptive systems. *Ecosystems*, 1(5):431–436, 1998.

[Linus Gransson, 2001] Timo Koski Linus Gransson. Using a dynamic bayesian network to learn genetic interactions. Technical report, Graduate School of Biomedical Research and Department of Mathemathics, Linkping University, 2001.

[Maria Chli, 2003] Philippe De Wilde Maria Chli. Stability of multi-agent systems, 2003.

[Murphy, 2002] Kevin Patrick Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. Phd, University of California, Berkeley, 2002.

[Murphy, 2004] Kevin Murphy. Bayes net toolbox for matlab, 2004.

[Stein, 2000] David S. Alberts John J. Garstka Frederick P. Stein. *NETWORK CENTRIC WARFARE: Developing and Leveraging Information Superiority*. DoD C4ISR Cooperative Research Program. CCRP Publications, 2 edition, 2000.

[Suzuki, 1998] Joe Suzuki. Learning bayesian belief networks based on the mdl principle: An efficient algorithm using the branch and bound technique. *IEICE Transactions on Information and Systems*, E-81D(12), 1998.

[Thomas M. Cover, 1991] Joy A. Thomas Thomas M. Cover. *Elements of Information Theory*. Wiley, New York City, New York, USA, 1991.

[Vladimir Pavlovic, 1999] Tat-Jen Cham Kevin P. Murphy Vladimir Pavlovic, James M. Rehg. A dynamic bayesian network approach to figure tracking using learned dynamic models. In *International Conference on Computer Vision (ICCV 99)*, pages 94–101, Corfu, Greece, 1999.

[Weisstein, 1999] Eric W. Weisstein. Bretschneider's formula, 1999.

[Yau, 2002] Pee Eng Yau. *An Exploratory Analysis On The Effects Of Information Superiority On Battle Outcomes*. Master of operations research, US Naval Postgraduate School, 2002.