Dynamics of Convergence to a Symbol Anchoring Consensus in Distributed Agents

Sam McKennoch and Linda G. Bushnell
{skennoch}@u.washington.edu

Dept of EE, University of Washington
Seattle WA, 98195-2500

UWEE Technical Report
Number UWEETR-2005-0006
3/17/05

Department of Electrical Engineering
University of Washington
Box 352500
Seattle, Washington 98195-2500
PHN: (206) 543-2150
FAX: (206) 543-3842
URL: http://www.ee.washington.edu

Dynamics of Convergence to a Symbol Anchoring Consensus in Distributed Agents

Sam McKennoch, *Student Member,* and Linda G. Bushnell, *Senior Member, IEEE*

Abstract **– This work seeks to present the initial steps towards developing an autonomously learned communication language among distributed agents. Symbols are anchored directly in the environment by having agents assign symbols to any distinct specimen in the environment that they are able to sense. Agents start with no previous knowledge of the environment. Agents then perform a classification and consensus routine in order to come to approximate agreement on specimen-symbol relationships. Convergence of the population to the same lexicon is examined and compared to a traditional control system where the consensus method used is analogous to a feedback controller.**

I. Introduction

MULTI-ROBOT systems, although complex in terms of coordination and communication, are extremely powerful in terms of tasks that they can perform. For example, having a multi-robot system allows for increased robustness over single robot systems, as the loss of a single robot is not necessarily catastrophic. However, in order for robots to successfully reason about and interact with their environment, they must possess accurate internal representations, typically symbolic, of objects in their environment. Furthermore, in order for robots to interact with each other there must a method by which their individual internal representations are brought into sufficient agreement to enable the robots to perform cooperative tasks. This problem, as depicted in Fig. 1, is that of symbol anchoring in a multi-robot system.

Symbol grounding, one important component of symbol anchoring, involves levels of representation known as iconic, indexical, and symbolic. Icons are direct representations of objects, such as sensor observations, indices associate icons together, and symbols group together other symbols as well as indices. If there exists a path from all symbols to icons then the system is grounded. Thus there is a relationship between agents' internal representations of objects and the environment. Without grounding, symbolic reasoning between agents can suffer for many reasons, including the "frame problem" [4]. The environment gives multiple agents a common source of reference.

Symbol grounding assumes that agents see their environment in terms of objects. However, situated, embodied robots instead see their environment in terms of sensor readings, observations. Symbol anchoring is a linguistic term borrowed originally from the study of situational semantics. It originally referred the attempt to find meaning in sentences in the environment in which they are spoken [3]. In robotics, we add a high degree of pragmatism to symbol anchoring as we are interested in using it to help create functional multi-robot systems. Thus symbol anchoring involves taking techniques from pattern recognition, to find objects in observations, and then using symbol grounding to symbolically represent these objects.

Symbol anchoring is an active area of research. However, many of these experiments only use two robots or less, thus limiting applicability and not adequately addressing the consensus problem. For example, in [11], two robots are put in an environment with multiple light sources. Their task is to symbolically anchor the light sources with four light sensors on each robot. Also, many of these experiments have in some way pre-optimized sensors or symbols based on very specific domain knowledge. Feature extraction from sensors also is pre-determined, often using techniques from computer vision to find features such as colored shapes or orientation of objects [9], [10]. In [4], two cleaning robots develop a lexicon of symbols describing their locations and locations of litter piles. Feature detectors interpret sensor

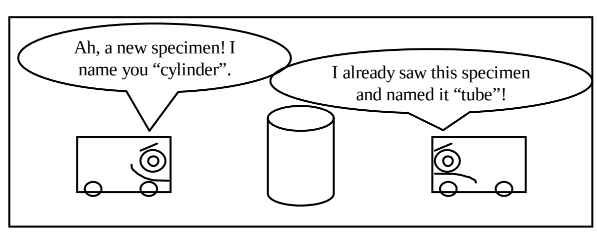


Fig. 1. Illustration of the problems in reaching a symbol anchor consensus in a distributed population operating in an uncharacterized environment.

Manuscript received September 17, 2004. This work is supported in part by the US National Science Foundation Grant No. ECS-0322618 and Grant No. ECS-0137422.

Sam McKennoch* and Linda G. Bushnell are with the Electrical Engineering Department, University of Washington, Seattle, WA 98195-2500 USA (e-mail: skennoch@u.washington.edu).

data and send these features to competence modules. Once again the feature detector has been designed in such a way that it looks to detect specific features, thus incorporating domain knowledge. Still other research puts a human in the loop, and/or establishes a command hierarchy among agents [6].

There are a number of challenges to symbol anchoring. For example, in a dynamic environment, internal representations of symbols must change in order for the system to remain grounded. However communicated symbols cannot change so quickly as to prohibit understanding between agents. Thus there must be a mechanism to update internal representations based on information communicated from other agents as well as new observations [4]. In other words, anchoring involves establishment and maintenance of these connections between symbols and sensor data. Besides anchor maintenance, other challenges include the uncertainty and ambiguity in real sensor measurements, the possible indefiniteness of an object (is an object an instance of some broader class of objects), the point of view an object is viewed from by different agents, and the reaching of consensus on symbol anchorings between robots [3]. Regarding consensus, the problem is compounded by polysemy (where one symbol maps to multiple specimen) and synonymy (where one specimen maps to multiple symbols). Polysemy and synonymy create incoherence in inter-agent comprehension, thus it is important that polysemy and synonymy be damped over time [10].

The long-term goals of this project are as follows. Agents should be able to perform the distributed exploration of environments (extra-planetary surfaces, the ocean floor, etc.) where the risk of failure due to communication occlusions and general system failure is great enough to justify a distributed approach, i.e., loss of a single agent is negligible. Agents should investigate and gather data on interesting discovered specimen based on prior conditions generally defining what is interesting. Finally, it is desired that scalability issues of communicating between large numbers of agents be overcome by communicating in a distributed manner through a learned communication system.

Towards those goals, this specific research effort examines symbol anchoring consensus among populations of distributed homogeneous agents. Other perspectives used to study symbol anchoring and its related issues include linguistics [1], artificial intelligence [5], intelligent control [8] and centralized control [2]. The approach presented here is a pragmatic one which could be applied to the desired applications, borrowing from these other areas as needed. As an example, a centralized control approach leads to problems when the population size is increased to a large number (100+) similar to what might be used in a real-world application. With large populations, it becomes more efficient to allow information to be processed in a distributed, parallel manner. This work takes a distributed approach, but acknowledges that a form of central control could eventually be necessary in gathering the information discovered by the distributed robot population. Also, this research assumes that no domain knowledge is available. Thus segmentation of sensor data into potential objects, hereafter called specimen, must be completely unsupervised. Perfect communication back to human operator may not be available due to time delays or occlusions. Also, the lethality of the environment to the agents presents a potential problem to using a well-defined hierarchy. These problems are avoided in this research by making all agents complete autonomous and of equal rank.

More specifically, a combination of principal component analysis (PCA) for dimensional reduction, hierarchal clustering for specimen identification and discriminant analysis for specimen classification is used in order to learn a specimen lexicon directly from the environment. Two simple methods for symbol anchoring consensus are then explored. A model of the consensus process is used to investigate scaling the number of agents upward in relation to the consensus method used.

This paper is organized as follows. Section II presents the simulation environment in which agents are placed. Section III details the symbol anchor consensus algorithms, which are capable of dealing with polysemy and synonymy. Section IV develops simple models of the consensus process and relates them to the more complex consensus problem presented by the simulation environment. Section V gives simulation results for lexicon (the set of all symbols used by each agent) convergence with different agent population sizes. Issues considered include steady-state error elimination, symbol diversity, robustness to sensor noise and scalability.

II. Lexicon Learning Environment

This section describes the simulation environment into which simulated embodied agents are placed. Parameters for the simulation environment are also discussed.

All agents are homogenous in that they are assumed to have the same sensors and actuators. Heterogeneity can offer many advantages to a multi-robot system, [7], but also increases the complexity of systems. In symbolically anchored systems with heterogeneous robots care must taken in designing specific mechanisms for interpreting observations made by different sensor suites into icons. The problem becomes one of sensor fusion. However, since this paper focuses on consensus, homogeneous robots are used to simplify the comprehension of observation data meanings between agents. Agents make these observations within their surrounding sensing radius into an environment tiled with specimen (Fig. 2).

Parameters for this model include the number of independent and dependent sensors. Independent sensors are each assigned a region in which to be active (value of one) and are inactive elsewhere (value of zero). Thus, for example, with 3 independent sensors, there are 2^3, or 8, possible specimen types. Dependent sensor values are used to artificially increase the dimensionality of observations, and are linear combinations of the independent sensors values measured in a given observation. Readings from both types of sensors are then perturbed by Gaussian noise to a level determined by a noise perturbation parameter in order to model the noise present in a real-world environment. For simplicity, the specimen environment is static. Agent movement in the environment takes places randomly. This type of movement is highly inefficient, but ensures that all specimens will eventually be viewed.

III. Lexicon Consensus Algorithm

In this section, the symbol anchor consensus algorithms, which can deal with polysemy and synonymy, are detailed.

Agents seek to reach consensus on the lexicon. Consensus is composed of agreement, validity and termination. For this application, agreement means that there is a one to one mapping between every specimen-symbol pair, and that this mapping is the same for all agents. Validity means that if all agents start with an assigned symbol for a specific specimen, then all agents will keep this symbol assignment. Finally, termination means that all agents eventually reach a consensus with each other. More precisely, let I be a set of unique indices on the agents; $i,j \in I$; S be the set of all Specimen, Λ is the set of all symbol anchors, Σ is the set of assigned symbols, i.e., the lexicon, O is the set of all observations. Using this notation, the following functions are defined: G: $S \times I \rightarrow O$, H: $O \rightarrow \Sigma$, $\Lambda(S,I)=H \circ G$, $\square\Diamond\exists\, \Lambda:(S \times I \rightarrow \Sigma \wedge \Lambda$ *is bijective).* In words, this last statement says that for all agents there always eventually exists a common mapping from specimen to symbols that is unique for every specimen and symbol. The algorithm to be used in subsequent simulations, assumes that in the worst case there is infinite time for convergence, and that all specimens are correctly identified by at least one agent. The algorithm proceeds according to the following steps, where times $t < t' < t'' < t'''$ are used as superscripts to indicate the set value at a specific time.

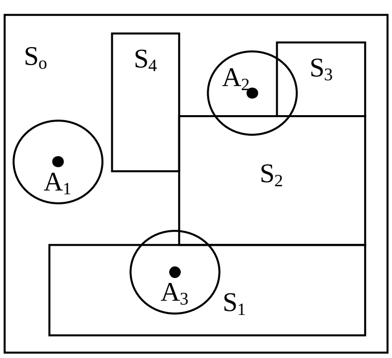


Fig. 2. An example agent environment showing three agents and their sensing ranges, as well four specimens. Specimen 2 will most likely be assign two different names by Agents 2 and 3, thus they will subsequently have to reach consensus in both the name and meaning of this specimen.

***Observe**:* Agent$_i$, observes specimen within its sensing radius over time.

$$Observe(S_i, i) \rightarrow O_i^{t'}$$

Segment: If no symbols exist, then normalize data, perform PCA and hierarchal clustering to assign a symbol name to similar observations. The best number of clusters to use, η_o, is determined as in (1) by minimizing the variance within a cluster and maximizing the variance between clusters.

$$\eta_o : \min_{\eta}\left(\frac{\overline{IntraClusterVariance(\eta)}}{\overline{InterClusterVariance(\eta)}}\right) \qquad \eta = \{1,2,\ldots,19,20\} \tag{1}$$

where η is the number of clusters that the data was split into. After η_o is chosen, non-viable clusters containing too few observations are removed.

$$Segment(O_i^{t'}) \rightarrow \Lambda_i^{t'}$$

***Classify**:* Otherwise perform discriminant analysis (minimize the Mahalanobis distance between cluster centers and observations) on the new observations in order to classify them according to pre-existing symbols. If there exists observations for which no acceptable symbol assignment can be found, add new symbols.

$$Classify(\Lambda_i^t, O_i^{t'}) \rightarrow \Lambda_i^{t'}$$

***Generate**:* Generate pseudo-test-observations from clustered data.

$$Generate\left(\Lambda_i^{t'}\right) \rightarrow \hat{O}_i^{t'}$$

***Transmit**:* Send generated observations to Agent$_j$, chosen randomly.

$$Transmit\left(\hat{O}_i^{t'}, j\right)$$

***Classify**:* Agent$_j$, performs discriminant analysis on the new observations, classifying them according to pre-existing symbols. If there is data for which no acceptable symbol assignment exists, then add new symbols. New observations may cause symbols to merge or split. If new observations which are classified to a single specimen by Agent$_i$ are classified to multiple specimen by Agent$_j$, or if new observations which are classified to multiple specimen by Agent$_i$ are classified to a single specimen by Agent$_j$ then all observations involved from both Agent$_i$ and Agent$_j$ are resegmented to determine the best number of clusters that they should be divided into based on (1). More observations can be requested to resolve ambiguity.

$$Classify\,(\Lambda_j^{t'}, \hat{O}_i^{t'}) \rightarrow \Lambda_j^{t''}$$

Assign: Without consensus feedback symbol names are directly accepted by Agent$_j$ from Agent$_i$, on face value. With consensus feedback, as in [5], symbol names from Agent$_i$ cause an increase for the symbol name in Agent$_j$'s Preference Vector, P^j. The symbol with the most votes is the symbol that is used, for each specimen. Thus consensus feedback allows agents to have a memory. In [5], the authors showed that using memory allows convergence in their system in *O(Nlog(N))* as opposed to $O(N^2)$ without consensus feedback, where N is the number of agents in the population. A similar speed-up is expected in the system presented here.

$$Assign\,(\Lambda_j^{t''}, \Lambda_i^{t'}, [P^j]) \rightarrow \Lambda_j^{t'''}$$

The above steps repeat with a listening and speaking agent being chosen randomly at each cycle through the algorithm to simulate the way a physical distributed system would proceed. The simulation ends when a maximum number of steps have been exceeded, or when the lexicon converges and remains at the number of specimen in the environment for a fixed number of steps.

IV. Simple Consensus Models

In this section simple models of the consensus process are developed and related to the more complex consensus problem presented by the simulation environment. For the general case, this consensus is extremely difficult to prove, thus consensus is shown to occur for simple cases mathematically, with more general cases left to simulation.

Case 1: Three agents all uniquely initialized to different symbol names for one specimen

For this case, perfect segmentation is assumed. State A is defined as the state where each agent has a different name for the specimen (1-1-1); state B occurs when two agents agree on the name, but the other does not (2-1); and state C is the consensus state (3). A listening and a speaking agent are chosen at random. The listening agent automatically accepts the speaking agent's name for the specimen. The state flow can be described as a Markov Chain.

$$T_3 = \begin{bmatrix} 2/3 & 1/3 \\ 0 & 1 \end{bmatrix} \qquad (2)$$

T_3 is the transition table for the Markov Chain in Fig. 3. State A is ignored because state B necessarily follows from state A. Raising T_3 to the nth power, gives probabilities that starting from state r (the row index), state c (the column index) is reached after n transitions. The consensus state is an absorbing state in that once achieved, the agents will never fall out of consensus (according to consensus validity). For three agents, the number of steps it takes to achieve a 95% certainty that the agents have achieved consensus is 7.4. As the number of steps approaches infinity, this probability approaches 1, thus the lexicon asymptotically converges with certainty of one (consensus agreement and termination).

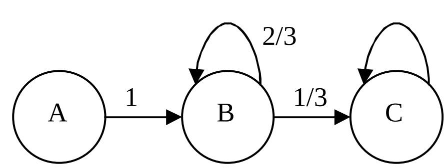


Fig. 3. Markov chain for the consensus process with three agents.

Case 2: Four and five agents all uniquely initialized to different symbol names for one specimen

These cases are found in an identical manner to case for three agents, but are more complex. The state definitions for these cases are as follows, with the format of (Number of Agents, Lexicon Configuration): (4,2-1-1), (4,3-1), (4,2-2), (4,1), (5,2-1-1-1), (5,3-1-1), (5,2-2-1), (5,3-2), (5,4-1), and (5,5).

$$T_4 = \begin{bmatrix} 1/2 & 1/3 & 1/6 & 0 \\ 0 & 1/2 & 1/4 & 1/4 \\ 0 & 2/3 & 1/3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$T_5 = \begin{bmatrix} 0.4 & 0.3 & 0.3 & 0 & 0 & 0 \\ 0 & 0.3 & 0.3 & 0.1 & 0.3 & 0 \\ 0 & 0.4 & 0.4 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0.7 & 0.3 & 0 \\ 0 & 0 & 0 & 0.2 & 0.6 & 0.2 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

For four agents, the number of steps it takes to achieve a 95% certainty that the agents have achieved consensus is 18.7, while for five agents it is 34.1 steps. Although, it is dangerous to generalize based on only three cases, the given data converges with 95% certainty in $O(N^{2.1})$. This result makes sense considering the slightly different case where half an agent population is initialized to one symbol, and the other half to a different symbol. In this case, consensus is reached in $O(N^2)$ [5], slightly faster as would be expected. Once again, as the number of steps approaches infinity, the probability of convergence approaches 1, thus the lexicon asymptotically converges with certainty of one.

Case 3: Three agents and two specimen with no initialization

In this case, the initial lexicon size of the population is zero. When an agent is chosen to speak, if it has not been chosen before, it first observes its environment. It segments the data into clusters, and assigns a symbol name to each cluster. Next, the agent attempts to communicate with other agents that have already segmented their environment. Agents are assumed to be randomly placed on each turn. There is equal chance of observing any specimen, thus the probability of observing any specimen is ½. The notation used to represent states in this case includes the symbol used by each agent for each specimen. For example, consensus is represented by {[α,β], [α,β], [α,β]}, where $\{\alpha,\beta,\gamma,\delta\}\in\Sigma$. All of the states in Table I should be viewed as including those states symmetric to themselves as well. For example, {[α,β], [α,0], [β,0]} and {[α,β], [β,0], [α,0]} are symmetric. Zero designates that this specimen has not yet been discovered by that agent.

Once again a state transition matrix can be generated using probabilities found by examining the probability that a specific agent is chosen to speak, the probability that a specific agent is chosen to listen, and the probability that a specific specimen with be observed. For this case, the number of steps it takes to achieve a 95% certainty that the agents have achieved consensus is 14, approximately twice that of the simpler case with three agents. As before, as the number of steps approaches infinity, the probability of convergence approaches 1, thus the lexicon asymptotically converges with certainty of one.

TABLE I
STATE DEFINITIONS FOR LEXICON AGREEMENT WITH TWO SPECIMENS, THREE AGENTS AND NO INITIALIZATION

Lexicon Configuration
{[0,0],[0,0],[0,0]}
{[α,0],[0,0],[0,0]}
{[α,β],[0,0],[0,0]}
{[α,β],[0,β],[0,0]}
{[α,β],[α,β],[0,0]}
{[α,0],[α,0],[0,0]}
{[α,β],[α,0],[0,0]}
{[α,β],[α,β],[γ,δ]}
{[α,β],[γ,β],[α,0]}
{[α,β],[α,β],[0,γ]}
{[α,β],[α,0],[0,β]}
{[α,β],[0,β],[0,β]}
{[α,β],[0,β],[0,γ]}
{[α,β],[0,γ],[0,γ]}
{[α,β],[α,β],[α,0]}
{[α,0],[γ,0],[γ,0]}
{[α,0],[α,0],[α,0]}
{[α,β],[α,β],[α,β]}

Having demonstrated convergence in the above cases, we now conjecture, but do not prove, the following.

Conjecture 1: If agents do not always perfectly segment their environment initially consensus will still occur according to a very similar mechanism as that presented in Cases 1, 2 and 3 as long as at least one agent eventually correctly and uniquely segments each specimen (the specimen are then labeled as being distinct), and when presented with such observations, other agents split and merge symbols as appropriate.

With imperfect classification, polysemy and synonymy can appear in the specimen-symbol mappings. Thus symbols must subsequently be merged and split as appropriate. However, the imperfect classification will create more states in the Markov Chain, and thus further increase convergence time. In all the cases discussed, there is only one absorbing state in the chain, thus as long as there exists a path from the beginning state to the absorbing state, the system will eventually converge to the absorbing consensus state. It follows from Conjecture 1 that such a state always exists when the conditions in the conjecture are met.

V. SIMULATION RESULTS

In this section simulation results are given for lexicon (the set of all symbols used by each agent) convergence with different agent population sizes. Issues considered include steady-state error elimination, symbol diversity, robustness to sensor noise and scalability. In describing the simulations to be presented, the following format will be used: (*A*, *N*, *W*, *C*), where *A* is the number of agents; *N* is the number of specimen; *W* is the number of grids on a side of the environment; and *C* is the number of simulation iterations over which to average results.

A. Eliminating Steady-State Error in Lexicon Convergence

There are two issues that prevent convergence in the presented simulations.

- More time is needed because contact between agents is probabilistic. In these cases, the final lexicon size is too high.
- Not all specimens have been initially correctly classified. In these cases, lexicon size is too low. In order for agents who have misclassified specimen to be able to correctly merge or split them, they need a correct model.

Thus, improvements were made to the simulation.

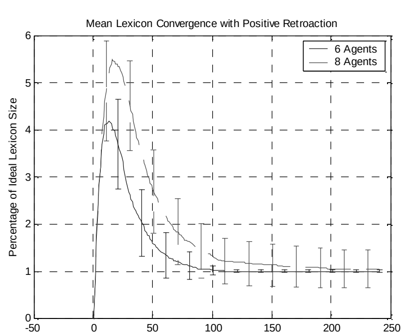


Fig. 4. (6/8,4,4/5,100) with Consensus Feedback.

Symbols that are diminutive, in that they have too few observations due to the rarity of the specimen, are augmented by adding more observations. The minimum percentage of the world that a specimen covers is increased in order to prevent specimen rarity. Also a longer time is allowed for convergence to be reached. These improvements drop the number of non-converging trials over a run of 100 simulations from 55 to 0. Converging results are shown in Fig. 4 using consensus feedback.

B. Symbol Diversity

An alternative method of visualizing lexicon convergence is by looking at the diversity in names for a single specimen as a simulation proceeds. Specimen that are more commonly sensed by agents as in Fig 5a tend to have a longer more complicated convergence process as there are more initial naming conventions compared to the more rarely observed specimen in Fig 5b.

Regarding the effects of using of consensus feedback, without consensus feedback (Fig 5c), the percentage corresponding to each symbol varies greatly on every iteration. It is also more difficult for new symbols to gain a foothold. With consensus feedback (Fig 5d) there is more stability as symbols are eliminated more gradually.

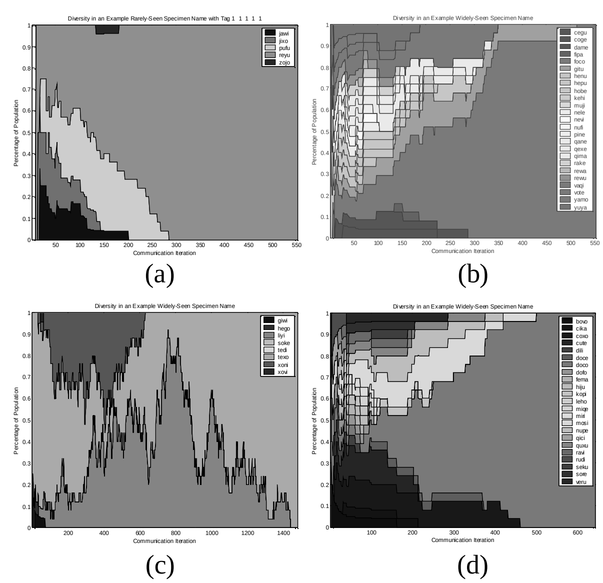


Fig. 5. (a) Diversity in naming for a rare specimen; (b) Diversity in naming for a common specimen; (c) Diversity in naming with no feedback; (d) Diversity in naming with consensus feedback. The vertical axes in this figure indicate the percentage of the population using a specific symbol. Different shadings indicate different symbols, which have been arbitrarily named using an alternating consonant-vowel pattern. The horizontal axes correspond to communication iterations.

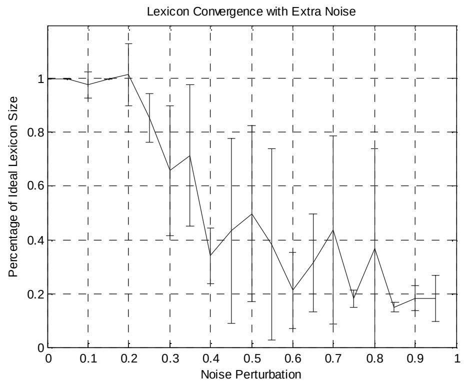


Fig. 6. Lexicon Convergence vs. Noise Perturbation. Error Bars indicate one standard deviation.

C. Robustness to Noise

Small amounts of noise in systems can be beneficial in helping to generalize parameters or mappings, such as symbol anchors. However, too much noise prevents convergence to the correct lexicon size as clusters are incorrectly merged. In Fig. 6, as noise perturbations are increased above 20%, lexicon convergence is greatly diminished.

D. Scalability

In order to be applicable to the applications previously mentioned the algorithm must be able to scale to larger amounts of agents. As the number of agents is scaled upward the benefits of using consensus feedback become more apparent.

In general, consensus feedback causes a higher overshoot, but then allows the final value to be approached at a faster rate than is the case without consensus feedback. The use of consensus feedback in this distributed system can be roughly compared to the use of proportional-derivative feedback in a traditional control system. Without using consensus

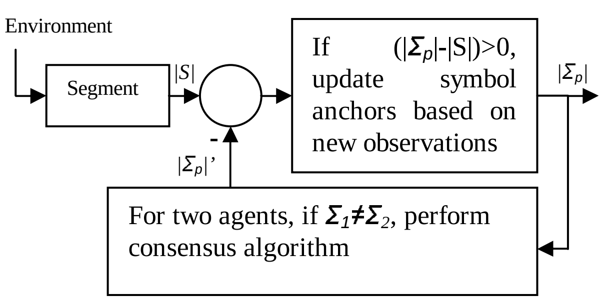
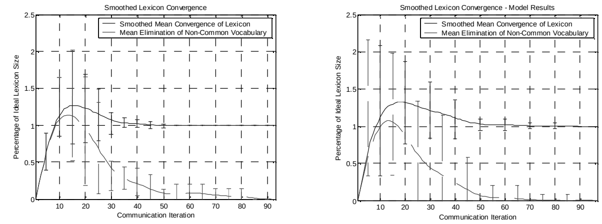
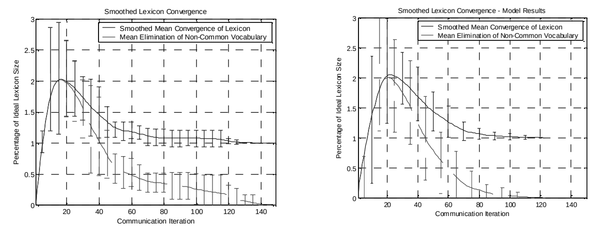


Fig. 7. Symbol anchoring consensus system drawn to resemble a traditional control system, with the consensus algorithm acting like a feedback controller. $|S|$ is the ideal number of specimen in the environment, Σ represents the lexicon for specific agents or for the entire population, depending on the subscript.



(a) (b)

Fig. 8. (6,5,11,30) without consensus feedback, actual results (a) and modeled results (b).



(a) (b)

Fig. 9. (6,5,11,30) with consensus feedback, actual results (a) and modeled results (b).

feedback, the system resembles more of a control system with proportional control. Fig. 7 demonstrates this concept by laying out the system in terms of a traditional control system.

In the case of no consensus feedback, as the number of extra symbols above the ideal value decreases, the consensus algorithm will be able to decrease the number of symbols less because so many agents already agree. It is proportionally activated. In the case of consensus feedback, the proportional nature of the consensus algorithm described remains in take, but also the number of extra symbols changes in proportion to the change in the number of extra symbols. If symbols are being eliminated with consensus feedback, it means that there has been enough communication throughout the population and that the symbols have been around long enough to dominate. However, for the test cases looked at in this paper, stability does not appear to be affected as is the case with traditional

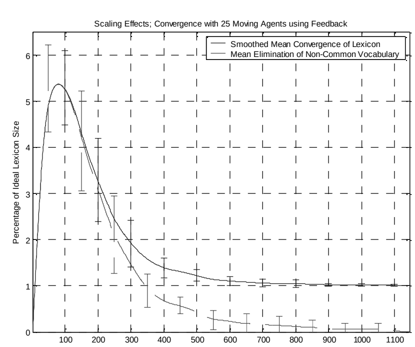


Fig. 10. (25,4,10,10) With consensus feedback, the average convergence time was 732.6 iterations, with 232.3 standard deviation.

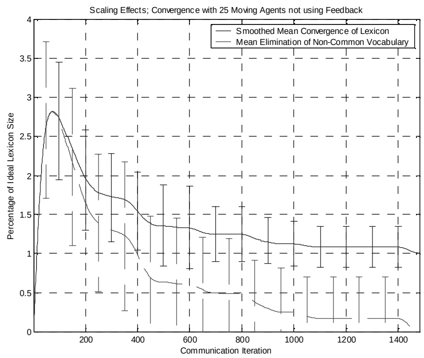


Fig. 11. (25,4,10,10) Without consensus feedback, the average convergence time was 676.6 iterations, with 376.8 standard deviation.

control systems.

Due to the intensive nature of full simulations, a simplified model was developed in which segmentation and classification were modeled probabilistically, while the consensus algorithm was performed as normal. This model was tested and compared to simulations using six agents. The model results are similar is respect to maximum overshoot value, time of maximum overshoot, convergence times, and general shape of curves. Results are shown in Fig. 8 and Fig. 9.

Model results indicate that for populations larger than 29 agents, consensus feedback will converge faster. This prediction is consistent with the simulation results in Fig. 10 and Fig. 11 in which convergence times are very close.

If the number of agents is increased further to 100 agents, then the use of consensus feedback allows convergence to be reached nearly 50 times as fast (approximately 4,000 iterations with consensus feedback vs. 200,000 iterations without consensus feedback). With consensus feedback, the maximum lexicon size is large, but the rate of convergence is much more rapid.

VI. SUMMARY AND FUTURE WORK

In this research effort an algorithm that allows agents to reach consensus on a specimen lexicon in which the specimen were discovered by the agents was designed, simulated and modeled. Agents anchored symbols to specimens without any prior domain knowledge. Lexical convergence and the speed thereof, were derived mathematically for simple cases, and conjectured to occur for more complex cases. Problems in convergence in subsequent simulations were investigated and addressed. The use or non-use of consensus feedback was examined, with the results being compared to typical control system responses. Consensus feedback was shown to be analogous to a proportional-derivative feedback controller in lexicon convergence. For agent populations of size greater than 29, consensus feedback was shown to be a faster method. Polysemy and synonymy were dealt with by appropriately merging and splitting symbols. The effects of scaling the number of agents and adding noise were also investigated in-line with the pragmatic approach described in Section I.

Much work remains to be done in this research effort. In the future this research effort will investigate issues such as language syntax, the use of non-binary sensors, standard non-linear data sets such as the Iris data, heterogeneous sensor suites, and spatial effects where agents are restricted to communicate only with their neighbors. The above issues will be examined in regards to standard issues of convergence, stability, and robustness in the face of disturbances (such as the loss of an agent) as language lexicon and syntax are modulated with environmental complexity.

REFERENCES

[1] Allexandre, C. and Popescu-Belis, A., "Emergence of grammatical conventions in an agent population using a simplified tree adjoining grammar," in *Proceedings of the International Conference on Multi Agent Systems*, 1998, pp. 383 –384.

[2] Billard, A., Epars, Y., Schaal, S. and Cheng, G. "Discovering Imitation Strategies through Categorization of Multi-Dimensional Data," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.

[3] Coradeschi, S. and Saffioti, A., "An introduction to the anchoring problem," *Robotics and Autonomous Systems*, Vol. 43, 2003, pp. 85-96.

[4] Jung, D. and Zelinsky A., "Grounded Symbol Communication between Heterogenous Cooperating Robots," Autonomous Robots, Vol. 8, No. 3, 2000, pp. 270-292.

[5] Kaplan, F., *La naissance d'une langue chez les robots*. Paris: HERMES Publications, 2001.

[6] Khoo, A. and Horswill I., "Grounding inference in distributed multi-robot environments," *Robotics and Autonomous Systems*, Vol. 43, 2003, pp. 121-132.

[7] Murphy, R., Ausmus, M., Bugajska, M., Ellis, T., Johnson, T., Kelley, N., Kiefer, J., and Pollock, L., "Marsupial-Like Mobile Robot Societies," in *Proceedings of the Third International Conference on Autonomous Agents*, 1999, pp. 364-365.

[8] Noble, J., "Talk is cheap: Evolved strategies for communication and action in asymmetrical animal contests," in *Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior*, 2000.

[9] Steels, L. and Baillie, J., "Shared grounding of event descriptions by autonomous robots," *Robotics and Autonomous Systems*, Vol. 43, 2003, pp. 163-173.

[10] Steels, L. and Kaplan, F., "Situated Grounded Word Semantics," in *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence IJCAI'99*, 1999, pp. 862-867.

[11] Vogt, P., "Anchoring of semiotic symbols," *Robotics and Autonomous Systems*, Vol. 43, 2003, pp. 109-120.