Algorithm Steering for Mixed-Initiative Robot Teams

Dhruba Baishya and Michael Lewis
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA 15208
dhrubajbaishya@gmail.com, ml@sis.pitt.edu

Abstract. While much research has been devoted to human-robot interaction (HRI) with individually controlled robots or round-robin control for independently operating ones, the problems of controlling autonomously coordinating robot teams remain largely unexplored. Although MAS researchers have devoted significant effort to understand human interaction with teamwork algorithms other significant classes of coordination algorithms have not received comparable attention. In particular, human interaction with biologically inspired and optimizing control algorithms has been long neglected. These algorithms which are ideal for tightly coordinated tasks such as formation flying or simultaneous rendezvous require highly coordinated mutual adjustments yet have goals that can be simply specified. The problem arises when the operator wants the system to do anything else. Because there is little or no connection between the parameters available to the operator and the behavior that results we call such algorithms *opaque*. We conjecture that in many cases this problem may be relatively easy to solve and propose a taxonomy-in-progress to help identify classes of algorithms to be considered.

Keywords: multirobot systems, human-robot interaction, teamwork algorithms.

Applications for multirobot systems (MrS) such as interplanetary construction or cooperating uninhabited aerial vehicles will require close coordination and control between human operator(s) and teams of robots in uncertain environments. Human supervision will be needed because humans must supply the perhaps changing, goals that direct MrS activity. Robot autonomy will be needed because the aggregate demands of decision making and control of a MrS are likely to exceed the cognitive capabilities of a human operator. Controlling robots that must act cooperatively, in particular, will likely be difficult because it is these activities [7] that theoretically impose the greatest decision making load. Because some functions of a MrS such as identifying victims among rubble depend on human input, evaluating the operator's span of control as the number of controlled entities scale is critical for designing feasible human-automation control systems.

Current estimates of human span of control limitations are severe. Miller [11], for example, showed that under expected target densities, a controller who is required to authorize weapon release for a target identified by a UCAV could control no more than 13 UAVs even in the absence of other tasks. A similar breakpoint of 12 was found by [4] for retargeting Tomahawk missiles. Smaller numbers (3-9) [3] have typically been found for ground robots which usually require more frequent attention.

To extend operator span of control to larger teams we must consider how control difficulty for different control tasks grows with increases in team size. Computational complexity theory [13] offers one possible approach. Borrowing concepts and notation from computational complexity, authorization for weapon release after operator verification of each UAV-detected target, can be considered O(n) because demand increases linearly with the number of UAVs to be serviced. Anther form of control such as designation of an attack region by drawing a box on a GUI (Graphical User Interface), being independent of the number of UAVs, would be O(1). Practical applications are likely to require some mixture of control regimes. In our prior work with wide area search munitions [9], for example, the operator specified search and jettison areas and ingress and egress routes , O(1), but was also required to authorize attacks and allowed to command UAVs directly, both tasks of O(n) complexity. Examined from this perspective the most complex tasks faced in controlling large teams are likely to be those that involve choosing and coordinating subgroups of UVs. Simply choosing a subteam to perform a particular task (the iterated role assignment problem), for example, has been shown to be O(mn) [7].

The flip side of reducing command complexity for the human is to increase the reliance on automation to accomplish the task. The human's O(1) search command, for example, requires path planning, coordination, resource management, etc. from the robot team. The complexity of tightly coordinated tasks leads to a much shorter span of control than the O(n) tasks cited earlier. For a tightly coupled task such as box pushing (2 UGVs moving a box), for example, the operator is completely occupied commanding 2 robots [19] because every movement by one robot displaces the box requiring a corresponding movement by the other to catch up. For this reason for all but the simplest tasks coordination will need to be automated for robots to remain amenable to human control.

Relative to human control, multirobot coordination approaches can be divided into two general classes: teamwork algorithms that involve explicit assignment of roles and execution of plans and coordination schemes whose mechanisms are less cognitively accessible. In teamwork approaches such as Playbook [12], MissionLab [1], or Machinetta [16] the coordinated activities are planned out in advance often with the aid of graphical tools. Later during the mission these plans are executed by the robot team with human input limited to such things as calling plays (canned plans) or filling a role within a plan such as approving targets. While the moment to moment behaviors of the robots may involve complex interactions needed to coordinate movement the existence of a cognitively accessible plan governing their overall pattern of behavior provides the needed context for an operator to monitor and intervene as needed.

Many tightly coordinated tasks such as formation flying or simultaneous rendezvous require highly coordinated mutual adjustments yet have goals that can be simply specified. These tasks are probably best performed by algorithms based on optimization or biologically inspired control laws that can handle the extensive computation required. [10, 2], for example, have developed models for a human to assign tasks to a UAV cluster where the tasks are heavily constrained by the physical platform, such as the large turning radius of LOCAAS munitions, the sequence of activities that must occur for LOCAAS munitions, the timing constraints imposed by coordinated strike missions, and the physical relationships that must be enforced for certain types of search. The UAVs autonomously generate a plan that represents their constraints, and then present this plan to an operator who may influence the plan by changing costs, priorities, and constraints.

Because optimizing or biologically inspired control algorithms are very different from the cognitive processes a human might employ for the same problem, they can be difficult for human operators to comprehend or control. [15], for example, reported difficulties operators experienced in controlling a system making optimal weapons to target assignments because they could not designate targets directly but instead needed to adjust weights through trial and error to find a plan containing a desired target. This is a general problem of a class of algorithms we call opaque because their inner workings are not cognitively accessible to an operator. The difficulty of interacting through such algorithms is that while the primary goal for which they were designed, such as the region to be searched or rendezvous point, is easily expressed most other aspects of their behavior cannot be controlled directly. In both our examples control of other aspects of behavior required adjusting algorithm parameters in a way that could not be directly linked to the consequent behaviors.

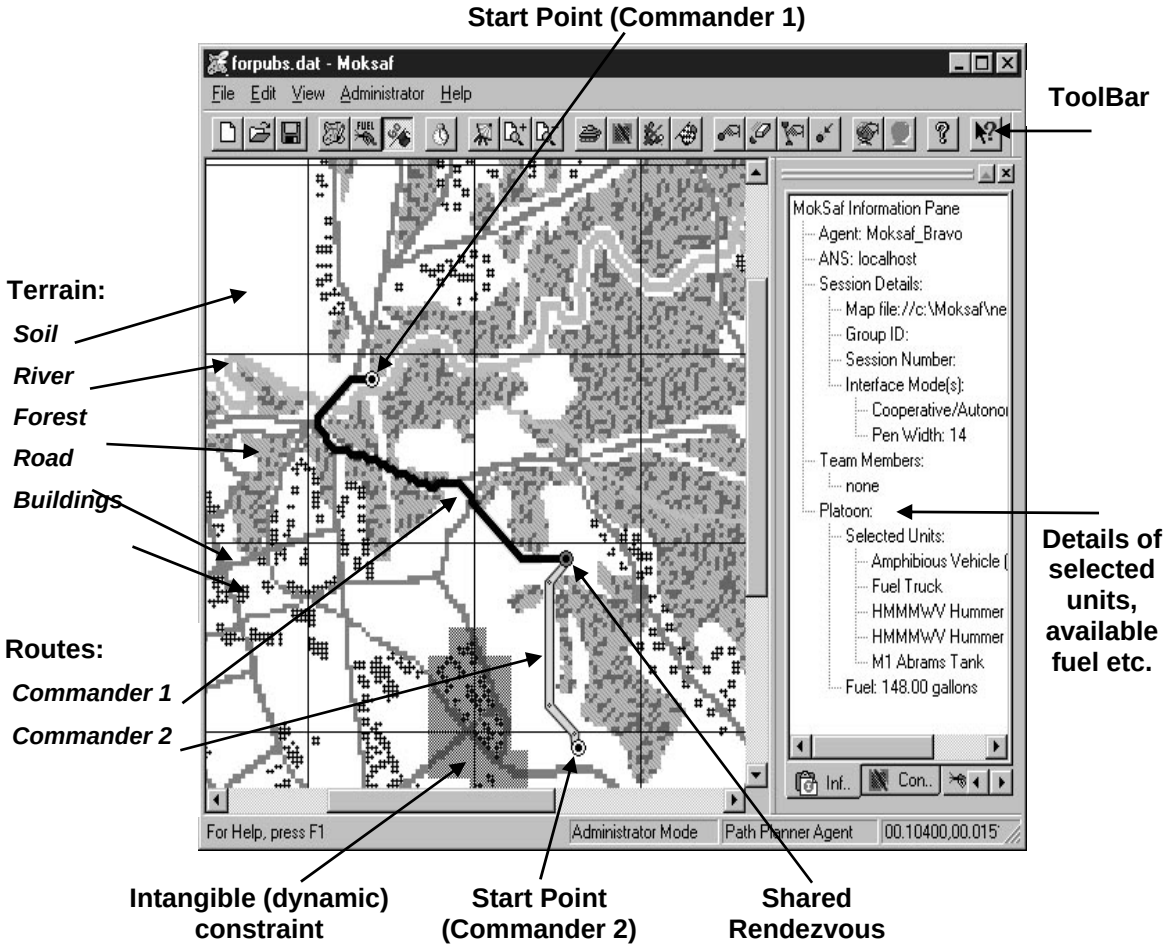
Robotics researchers studying multirobot control have frequently attempted to provide the operator with additional avenues of control usually through adjustments to parameters where a direct cognitive link could be found or imagined. Ron Arkin [1], for example, allowed operators to change the value of wanderlust, the magnitude of random deviations from a planned path the robot was allowed. Lynn Parker allowed the operator to adjust the twin tropisms impatience and acquiescence [14] to affect a robot's willingness to persist or abandon a role. One could imagine similar schemes involving biologically inspired local control laws such as broadcasting a change to a separation parameter in order to alter the dispersion of UAVs flying in formation. In each of these cases, however, the change in behavioral parameters does not directly impact the task but rather the way the robots perform the task. This opens the door for unanticipated consequences such as a wandering robot prevented by obstacles from returning to its path or a UAV formation that breaks apart due to sensing errors that grow at greater separations. What is needed for more effective human control is something akin to inverse kinematics that allows the operator to communicate the desired effect directly.

Because there are relatively few types of multirobot tasks that are both operationally relevant and admit optimizing or biologically inspired solutions (target assignment, parallel search, formation following, rendezvous, etc.) we believe these problems might be solved in a divide and conquer fashion. Our conjecture is that for any one of these task types there will be a finite and hopefully small number of *task relevant commands* an operator might desire to employ. In the weapon to target task, for example, task

relevant commands would include those referencing particular targets as well as global target types. To accommodate such commands the algorithm would need provisions to allow the operator to raise/lower the weight associated with a particular target rather than only for a target class. This adjustment would not require any significant change to the algorithm and could be presented transparently to the operator in terms of the domain as a command to either force inclusion or exclusion of targets. This simple change could have eliminated the difficulties experienced by Roth's subjects. We hypothesize that simple solutions of this sort may be frequent rather than rare and that commonly reported difficulties controlling opaque algorithms arise only because they have been written with a single goal without considering other actions an operator might desire.

MokSAF

The MokSAF path planning system based on Dykstra's algorithm, illustrates how an opaque algorithm can be effectively steered with minimal changes by paying attention to the task and task relevant commands. Human decision-makers, such as military commanders, face time pressures and an environment where changes may occur in the task, division of labor, and allocation of resources. Information such as



terrain characteristics, location and capabilities of enemy forces, direct objectives and doctrinal constraints

Figure 1. MokSAF Interface

are all part of the commander's "infosphere." Special purpose algorithms with access to this information can plan, criticize, and predict the consequences of actions at a greater accuracy and finer granularity than

the human commanders can. There is also, however, information that is not directly accessible to software. Such information includes intangible or multiple objectives involving morale, the political impact of actions (or inaction), intangible constraints, and the symbolic importance of different actions or objectives. When participating in a planning task, commanders must translate these intangible constraints into tangible ones to interact with planning algorithms.

We developed a computer-based simulation called MokSAF to evaluate how humans can interact with planning algorithms within a team environment. MokSAF is a simplified version of a virtual battlefield simulation called ModSAF (modular semi-automated forces). MokSAF allows two or more commanders to interact with one another to plan routes in a particular terrain. Each commander is tasked with planning a route from a starting point to a rendezvous point by a certain time. The individual commanders must then evaluate their plans from a team perspective and iteratively modify these plans until an acceptable team solution is developed.

The interface used within the enhanced MokSAF Environment is illustrated in Figure 1. The interface presents a terrain map, a toolbar, and details of the team plan. The terrains displayed on the map include soil (plain areas), roads (solid lines), freeways (thicker lines), buildings (black dots), rivers and forests. The rendezvous point is represented as a red circle and the start point as a yellow circle on the terrain map. As participants create routes with the help of a planner using Dykstra's algorithm, the routes are shown in bright green. The second route shown is from another MokSAF commander who has agreed to share a route. The algorithm is steered by the partially transparent rectangles representing intangible constraints that the user has drawn on the terrain map. These indicate which areas should be avoided when determining a route. As reported in [18, 8] this mixed initiative approach led to superior performance.

For this problem Dykstra's algorithm works efficiently by treating soil types as weights (asphalt=1 and water=infinity) so paths follow roads if they can, avoid open water and forests, etc. In terms of the single goal of shortest, lowest cost path it succeeds, however, there is no way to incorporate the intangible constraints known only to the commander. Upon considering the commander's task it is apparent that *task relevant* commands need to include excluding some areas from paths. This is easily achieved within the algorithm by artificially changing the soil weight of designated areas and easily used by the operator who can express his intent directly in the problem domain by marking exclusion areas on the map. In a close parallel to the weapons to targets example a simple but task informed modification to the algorithm overcame the problems of indirect control.

While three examples do not prove a principle we hope that by making our approach to re-engineering opaque algorithms explicit we can extend it to a larger class and come to understand for which types of algorithms it is appropriate.

As presently conceived the process consists of four steps:

1) identify objects in the domain subject to control; weapons, UAVs, and paths in our examples

2) examine the user's task and identify potential goals other than the focal goal of the algorithm

3) identify direct manipulation or other forms of command that might let the operator express these goals in a direct way in terms of domain objects

4) identify ways these commands might be realized within the algorithm. If none are available examine potential for transferring control over some assets to user.

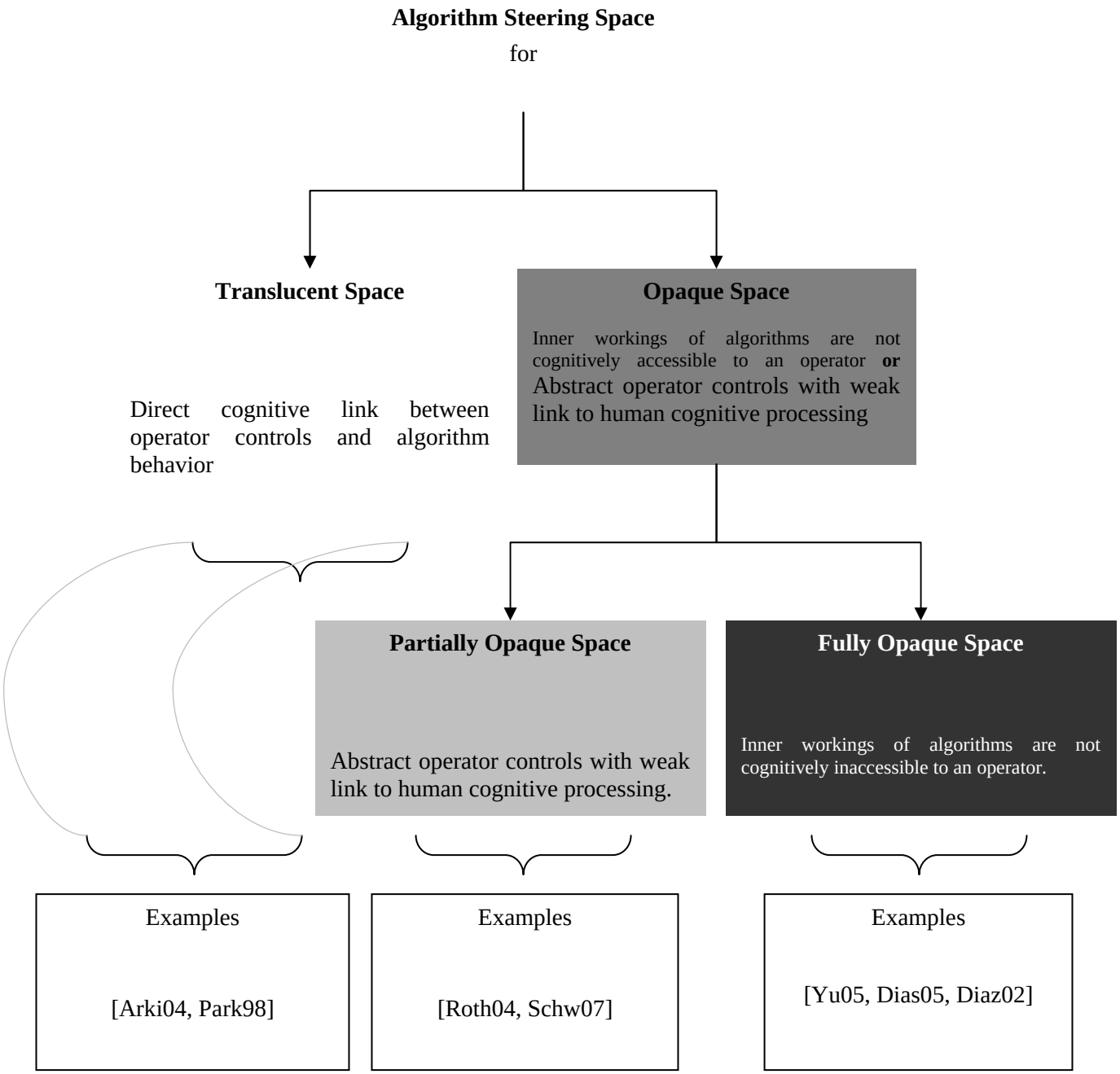
To begin our effort we are proposing a "strawman" taxonomy to help find representative cases to test.

A Possible Taxonomy for Algorithm Steering

We are proposing a user-based taxonomy for algorithm steering. This MrS Space is broadly classified into three categories –First, transparent space, where the operator can express goals explicitly in terms of domain objects. Second, translucent space, where some cognitive links exist between operator controls and algorithm behavior. Third, opaque spaces, where inner workings of algorithms are not cognitively accessible to an operator, or operator controls are "abstract" with weak links to human cognitive processing. Within opaque spaces, MrS could attain different levels of opaqueness. So this space has been further classified into two categories – partially and fully opaque spaces.

As discussed earlier, Ron Arkin [1], allowed operators to change the value of wanderlust, the magnitude of random deviations from a planned path the robot was allowed. Using wanderlust in place of "magnitude of random deviations…," allows MrS controllers to associate behavior with a cognitively well-defined

operator. Similar cognitively well-defined operators are used by Lynn Parker [14]. She allowed the operator to adjust the twin tropisms impatience and acquiescence to affect a robot's willingness to persist or abandon a role. [8], by contrast, provided a steering mechanism that allowed the operator to directly manipulate the planned path.



In our taxonomy, [8] would be considered transparent while [1, 14] are categorized as systems with translucent algorithm steering, or simply translucent space.

E. Roth [15], asked her operator controllers to adjust weights through trial and error to find a plan containing a desired target. In this case, the operator controls changes in algorithm behavior but they are cognitively ill-defined. “Adjust weights through trial and error…,” is an abstract control with weak link to human cognitive processing. [15, 17] are therefore categorized into partially opaque space.

Considering the difficulties involved in achieving an autonomous MrS and the need for deployed systems to accept human input, a mixed-initiative approach is necessary. However, there exist large pools of MrS, where inner workings of algorithms are not cognitively accessible at all to an operator. Such cognitive inaccessibility results in high opacity and reduced performance. Chih-Han Yu [20] et al, for example, reported in their experiments using multirobot POMDPs that with increasing complexity of an office environment, controllers (i.e. humans) found it very difficult to observe or follow the optimal policy being executed by their robots. Similar full opacity can be seen in market-based optimization MrS. [6], for example, allotted leaders to subgroups to enhance market-based multirobot coordination. This introduction of leaders among subgroups resulted in improved performance but made it even more difficult for observers to follow. [20, 5, 6] are qualified members of fully opaque space.

Discussion

Classifying algorithms by degree of opacity is one potentially fruitful way to proceed because it orders coordination algorithms in terms of human difficulty. It is easy to imagine other classification schemes, however, that might be equally fruitful. The distinction between centralized and distributed control is lost in our current scheme but almost certainly plays an important role in determining effective forms of re-engineering. The more general problems of controlling biologically inspired systems with emergent behavior or affecting behavior through complex adjustments to teamwork algorithm parameters as attempted in [9] are not yet addressed within our evolving framework.

References

1. Endo, Y., MacKenzie, D., and Arkin, R. (2004). Usability evaluation of high-level user assistance for robot mission specification, IEEE Transactions on Systems, Man, and Cybernetics- Part C, 34(2), 168-180.
2. Beard, R., McLain, T., Nelson, D., Kingston, D., "Decentralized cooperative aerial surveillance using fixed-wing miniature UAVs," IEEE Proceedings: Special Issue on Multi-Robot Systems, 2006.
3. J. W. Crandall, M. A. Goodrich, D. R. Olsen, and C. W. Nielsen. Validating human-robot interaction schemes in multitasking environments. IEEE Transactions on Systems, Man, and Cybernetics, Part A, 35(4):438–449, 2005.
4. Cummings, M. and Guerlain, S. An interactive decision support tool for real-time in-flight replanning of autonomous vehicles, AIAA Unmanned Unlimited Systems, Technologies, and Operations, 2004.
5. Dias, M. B., Zlot, R. M., Karla, N., Stenz, A. T. Market-based multi-robot coordination: a survey and analysis. Robotics Institute, Carnegie Mellon University, Tech. Rep. CMU-RI-TR-05-13, 2005.
6. Dias, M.B., Stentz, A. Opportunistic optimization for market-based multirobot control. Proceedings of IEEE/RSJ International Conference on Intelligent Robots and System, 2002, 2714- 2720
7. B. Gerkey and M. Mataric. A formal framework for the study of task allocation in multi-robot systems. International Journal of Robotics Research, 23(9):939–954, 2004.
8. Lenox, T., Hahn, S., Lewis, M., Payne, T. and Sycara, K. (2000) Task characteristics and intelligent aiding. Proceedings of the 2000 IEEE International Conference on Systems, Man, and Cybernetics, Oct 8-11, Nashville, TN, 1123-1127.
9. Lewis, M., Polvichai, J., Sycara, K., and Scerri, P. Scaling-up human control for large scale systems, In Cooke, N., Pringle, H., Pedersen, H., and Connor, O. (Eds.) Human Factors of Remotely Operated Vehicles, New York: Elsevier, 237-250, 2006.
10. McLain, T. and Beard, R. Coordination Variables, Coordination Functions, and Cooperative Timing Missions, AIAA Journal of Guidance, Control, and Dynamics, vol. 28, no. 1, pp. 150-161, January-February 2005.
11. Miller, C. Modeling human workload limitations on multiple UAV control, Proceedings of the Human Factors and Ergonomics Society 47th Annual Meeting, 526-527, 2004.
12. Miller, C., & Parasuraman, R. Designing for flexible interaction between humans and automation: Delegation interfaces for supervisory control. Human Factors, 49.57-75, 2007.
13. Papadimitriou, C. (1994). Computational Complexity. Addison Wesley, 1994.

14. Parker, L.E., ALLIANCE: An Architecture for Fault Tolerant Multi-Robot Cooperation, IEEE Transactions on Robotics and Automation , 1998
15. Roth, E., Hanson, M., Hopkins, C., Mancuso, V., and Zacharias, G. “Human in the loop evaluation of a mixed-initiative system for planning and control of multiple UAV teams,” Proceedings of the Human Factors and Ergonomics Society 48th Annual Meeting, 280-284, 2004.
16. Scerri, P., Vincent, R., and Mailler, R. Coordination of Large-Scale Multiagent Systems, Birkhauser 2006.
17. Schwager, M., Slotine, J. J., Rus, D. Decentralized, Adaptive Control for Coverage with Networked Robots. Proceedings of IEEE International Conference on Robotics and Automation, 2007, 3289-3294
18. Sycara, K. and Lewis, M. (2004). Integrating intelligent agents into human teams. In E. Salas and S. Fiore (Eds..) Team Cognition: Understanding the Factors that Drive Process and Performance, Washington, DC: American Psychological Association, 203-232.
19. Wang, J. and Lewis, M. Assessing coordination overhead in control of robot teams, Proceedings of 2007 IEEE International Conference on Systems, Man, and Cybernetics, 2645-2649, 2007.
20. Yu, C., Chuang, J., Gerkey, B., Gordon, G. J., Ng, A. Open-loop plans in multi-robot POMDPs. Technical Report, Stanford University, 2005