Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Human Teams for Large Scale Multirobot control

Huadong Wang, Shih Yi Chien, Michael Lewis
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA 15260 USA
ml@sis.pitt.edu

Prasanna Velagapudi, Paul Scerri, & Katia Sycara
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213 USA
pkv@cs.cmu.edu

***Abstract*—We are developing an architecture for controlling robot teams based on considering how control difficulty for different tasks grows with increases in team size. Our analysis suggests that assignments of persons to *commander* (single commands to entire robot team), *operator* (commands to individual robots), and *coordinator* (control of interdependent robots) roles can lead to the most efficient organization. The ability to assign tasks within or between *operators* makes scheduling these interactions an important factor in team performance. Two possible ways to organize *operators* are through Individual Assignments of robots or as a Call Center in which *operators* service robots from the population as needed. In recent experiments we have found that participants performing an Urban Search And Rescue (USAR) foraging task using waypoint control were at or over their limits when controlling 12 robots. The present study uses the same robots, environment, and level of autonomy but with teams of two *operators* assigned to control 24 robots. These *operators* controlled teams of 12 robots in the Individual Assignment condition. In the Call Center condition operators shared control of the 24 robots. For this task and level of robot autonomy Individual Assignment participants performed marginally better searching larger regions but without finding more victims.**

***Keywords*—human robot interaction, multirobot systems, human factors**

I. INTRODUCTION

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 decision making demands of a MRS are likely to exceed the cognitive capabilities of a human operator. Autonomous cooperation among robots, in particular, will likely be needed because it is these activities [1] that theoretically impose the greatest decision making load.

Controlling multiple robots substantially increases the complexity of the operator's task because attention must constantly be shifted among robots in order to maintain situation awareness (SA) and exert control. In the simplest case an operator controls multiple independent robots interacting with each as needed. A search task in which each robot searches its own region would be of this category although minimal coordination might be required to avoid overlaps and prevent gaps in coverage especially if robots are in close proximity. Control performance at such tasks can be characterized by the average demand of each robot on human attention [2]. Under these conditions increasing robot autonomy should allow robots to be neglected for longer periods of time making it possible for a single operator to control more robots.

Established methods of estimating MRS control difficulty, neglect tolerance and fan-out [2] are predicated on such independence of robots and tasks. In neglect tolerance the period following the end of human intervention but preceding a decline in performance below a threshold is considered time during which the operator is free to perform other tasks. If the operator services other robots over this period the measure provides an estimate of fan-out, the number of robots that might be controlled. The approach presumes that operating an additional robot imposes an additive demand on cognitive resources. These measures are particularly attractive because they are based on readily observable aspects of behavior: the time an operator is engaged controlling the robot, interaction time (IT), and the time an operator is not engaged in controlling the robot, neglect time (NT). For more strongly cooperative tasks and larger teams individual autonomy alone is unlikely to suffice. The round-robin control strategy used for controlling individual robots would force an operator to plan and predict actions needed for multiple joint activities and be highly susceptible to errors in prediction, synchronization or execution. For the most highly dependent tasks such as teleoperating two robots to push a box, coordination demand completely occupies an operator's attention excluding any other task [3].

While independent robot control leads to more tractable attention sharing and prediction for single operators, dependent control poses even greater difficulties for teams of operators. If robots are not rigidly assigned to small teams under the control of a single operator then each event requiring cooperation will either occupy more of an operator's attention than corresponding independent tasks or require the operator to find the controller of another robot and assume the communication and coordination overhead needed to coordinate with him. These interaction times are likely to be highly variable making it difficult to schedule interactions without introducing excessive idle times. Since even moderate variability in NT has been shown [4] capable of leaving an operator spending 90% of the time waiting, avoiding such bottlenecks is crucial.

We are developing an architecture for controlling robot teams based on these observations. We begin by considering how control difficulty for different control tasks grows with increases in team size. Borrowing concepts and notation from computational complexity we can identify tasks such as identifying a victim through a robot's camera as O(n) because demand should increase linearly with the number of robots to be monitored. Another form of control such as designation of a search region by drawing a box on a GUI (Graphical User Interface), being independent of the number of robots, would be O(1). Practical applications are likely to require some mixture of control regimes. In our prior work with wide area search munitions [5], 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. Dependent tasks such as box pushing [3] or even choosing a subteam to perform a particular task [1], by contrast, appear to be arbitrarily difficult and therefore prime candidates for automation. We begin by observing that O(n) tasks although requiring substantial autonomy on the part of the robots impose only minor demands on the human. Because O(1) commands have global effects, however, it is architecturally important that they originate with a single operator or group to avoid contention. We term this the *commander*'s role. O(n) tasks such as display monitoring, approving targets, or marking victims are the sort of robot centric tasks described by the neglect tolerance model. Because these tasks are presumed independent they can be shared between operators as well as within neglect intervals. We term these *operator* roles. Finally, there are the tasks requiring explicit human coordination we will refer to as the *coordinator*'s role. We expect these tasks to be longer, more difficult, and more varied in length. Recent findings in scheduling [6] show that under high variability, such as when there are multiple types of jobs (independent vs. dependent), lower mean response times are obtained by partitioning hosts according to processing time ranges. Fortunately in our domain the difficulty/length of jobs can be expected to closely follow our O(1)/O(N) vs. O(NM) distinction. On this basis the partitioned jobs strategy amounts to mutually exclusive assignments of people to *operator* or *coordinator* roles. Considering these roles leads to an architecture in which efficiency is improved by automating as much of the *coordinators*' tasks as possible. The ability to assign tasks within or between *operators* makes scheduling these interactions the next important adjustment.

Two possible ways to organize *operators* are through Individual Assignments of robots or as a Call Center [5] in which *operators* service robots from the population as needed. Individual Assignment has the advantage of reducing the number of robots the *operator* must monitor. Call center offers the scheduling advantage of load balancing in that a pool of operators are available as robots need servicing. For monitoring, Call Center offers the redundant observer advantage in that a second observer with partially overlapping perceptual judgments may detect things missed by the first. We expect the effects of these advantages to interact with the types of autonomy possessed by the controlled robots. If robots lacked simple obstacle avoidance, for example, we would expect *operators* in either condition to be overloaded by even small teams and would expect little difference between conditions. If navigation and path planning were fully autonomous, by contrast, we would expect benefits to accrue to Call Center *operators* due to both scheduling and redundant observer advantages. If robots were able to self-diagnose, however, we might expect to see a strong scheduling advantage for Call Center but lose the redundant observer advantage since *operators* would no longer need to monitor closely. We would additionally expect to see substantial differences between types of autonomy in the numbers of robots that could be adequately controlled.

In recent experiments [7] we have found that participants performing an Urban Search And Rescue (USAR) foraging task using waypoint control were at or over their limits when controlling 12 robots. Participants asked merely to explore showed very similar performance in area covered and reported similar levels of workload on the NASA-tlx. Participants in a perceptual search condition in which the foraging task was performed without the requirement to navigate found twice the victims when monitoring 12 robots and reported substantially lower workload.

The present study uses the same robots, environment, and level of autonomy but with teams of two *operators* assigned to control 24 robots. These *operators* controlled teams of 12 robots in the Individual Assignment condition. In the Call Center condition operators shared control of the 24 robots. These data were collected to provide a control for comparisons between Individual Assignment and Call Center regimes under different forms of robot autonomy.

II. METHODS

A. USARSim and MrCS

The reported experiment was conducted using the USARSim robotic simulation with 24 simulated UGVs performing Urban Search and Rescue (USAR) foraging tasks. USARSim is a high-fidelity simulation of urban search and rescue (USAR) robots and environments developed as a research tool for the study of human-robot interaction (HRI) and multi-robot coordination. USARSim supports HRI by accurately rendering user interface elements (particularly camera video), accurately representing robot automation and behavior, and accurately representing the remote environment that links the operator's awareness with the robot's behaviors. USARSim can be downloaded from www.sourceforge.net /projects/usarsim and serves as the basis for the Virtual Robots Competition of the RoboCup Rescue League. USARSim uses Epic Games' UnrealEngine2 [8] to provide a high fidelity simulator at low cost. Validation studies showing agreement for a variety of feature extraction techniques between

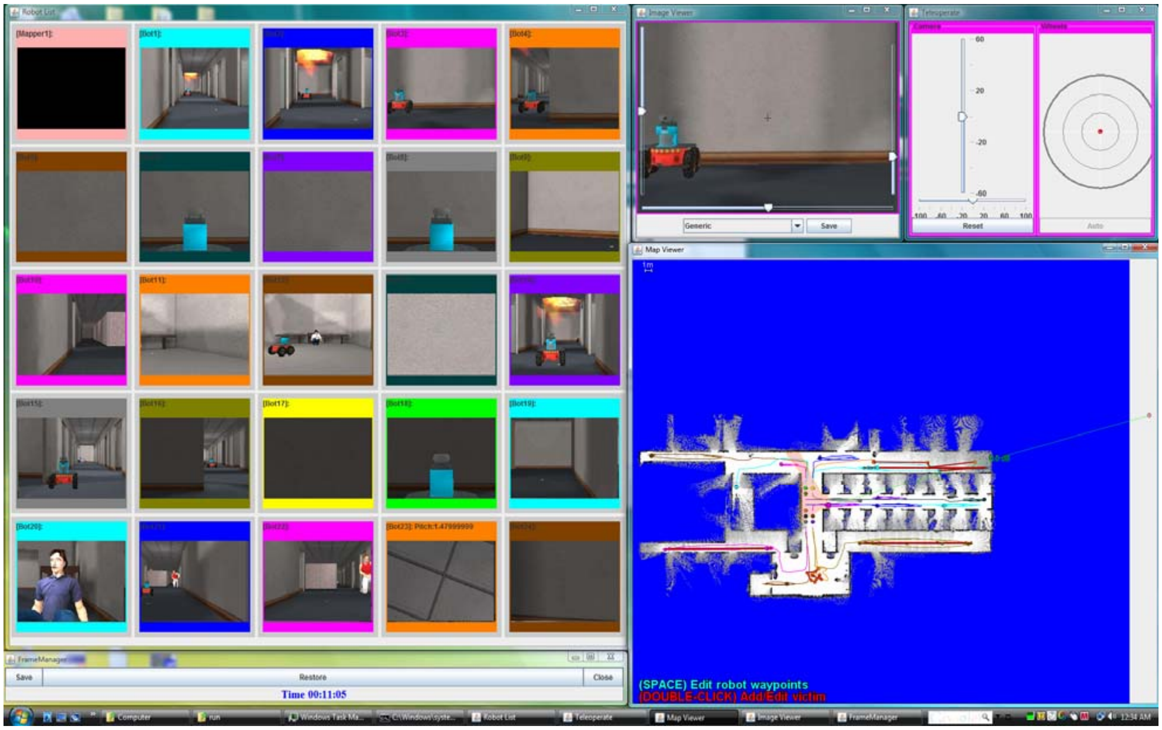

Figure 1. GUI for Multirobot control

USARSim images and camera video are reported in Carpin et al. [9]. Other sensors including sonar and audio are also accurately modeled. Validation data showing close agreement in detection of walls and associated Hough transforms for a simulated Hokuyo laser range finder are describedin [10]. The current UnrealEngine2 integrates MathEngine's Karma physics engine [11] to support high fidelity rigid body simulation. Validation studies showing close agreement in behavior between USARSim models and real robots being modeled are reported in [12,13,14,15,16].

MrCS (Multi-robot Control System), a multirobot communications and control infrastructure with accompanying user interface developed for experiments in multirobot control and RoboCup competition [17] was used in these experiments. MrCS provides facilities for starting and controlling robots in the simulation, displaying camera and laser output, and supporting inter-robot communication through Machinetta a distributed mutiagent system. Figure 1 shows the elements of the MrCS. The operator selects the robot to be controlled from the colored thumbnails at the top of the screen. To view more of the selected scene shown in the large video window the operator uses pan/tilt sliders to control the camera. Robots are tasked by assigning waypoints on a heading-up map on the Map Viewer (bottom right) or through a teleoperation widget (upper right). The current locations and paths of the robots are shown on the Map Data Viewer (bottom right).

B. *Experimental Conditions*

A large USAR environment previously used in the 2006 RoboCup Rescue Virtual Robots competition [17] was selected for use in the experiment. The environment was a maze like hall with many rooms and obstacles, such as chairs, desks, cabinets, and bricks. Victims were evenly distributed within the environment. A second simpler environment was used for training. The experiment followed a between groups design. Twenty four robots were controlled by teams of 2 participants. In the *individual control* condition participants were assigned 12 UVs each. Only the map information and victims marked were shared. In the *call-center* condition participants shared control of the 24 UVs having equal access in assigning waypoints and viewing video from all 24 robots. Maps with marked victims were shared the same as in the *individual control* condition.

C. *Participants*

60 paid participants (30 teams) were recruited from the University of Pittsburgh community. None had prior experience with robot control although most were frequent computer users.

D. *Procedure*

After collecting demographic data the participant read standard instructions on how to control robots via MrCS. In the following 30 minute training session, participants in all co

Table 1 Behavior Coding Result

Variables	Individual Control (N = 15)		Call-Center Control (N = 15)		T-value	P
	$\bar{x}$	SD	$\bar{x}$	SD		
1. Information Sharing	4.13333	2.94877	4.66667	3.86683	-0.42477	0.674253
2. Self-Organization	2.46667	1.80739	2.40000	1.54919	0.10847	0.914400
3. Problem Solving	2.86667	1.99523	2.66667	1.98806	0.27501	0.785328
4. Meta-Cognition	4.20000	3.78342	4.93333	3.51460	-0.55000	0.586680
5. Team Coordination	2.53333	2.50333	6.80000	6.36059	-2.41749	0.022388
6. Non-task Related	3.42857	3.41297	2.93333	3.05817	0.41210	0.683519
7. Interpersonal Affect	1.13333	1.99523	0.40000	0.73679	1.33535	0.192517
8. Closed Loop Communications	0.80000	1.32017	6.30769	5.75014	-3.61132	0.001277
Sub-total:	21.33333	15.62355	30.26667	19.62676	-1.37920	0.178749

conditions practiced control operations. Participants were encouraged to find and mark at least one victim in the training environment under the guidance of the experimenter. After. the training session, participants then began the real testing sessions (25 minute) in which they performed the search task controlling 24 robots in teams. Their conversation was recorded for the behavior coding analysis. After the task, the participants were asked to complete the NASA-TLX workload survey.

III. Results

Overall participants were successful in searching the environment in both conditions finding as many as 21 victims per team on a trial. The average number of victims found was 13.60 in the *individual control* condition but slightly lower at 12.33 for the *call-center control* condition. A paired t-test, however, found no significant difference, t (15) = 0.876, p = 0.38. The region explored, however, showed a significant advantage t (15) = 2.716, p = 0.011 for *individual control*.

Table1 shows the observed information sharing behavior under the two conditions. There were significant differences in the number of team coordination conversations between the *individual control* condition and the *call-center* (t= -2.417, P<0.022). Similar results were found for numbers of closed loop communications between the two condition (t= -3.61, P<0.001).

Table 2 shows process measures of team performance There were significant differences in number of missions (sets of waypoints) issued, t(28)= 2.312, p<0.028(as shown in Figure 4); however, the number of switch between robots and the waypoints issued per mission were very similar across the conditions.

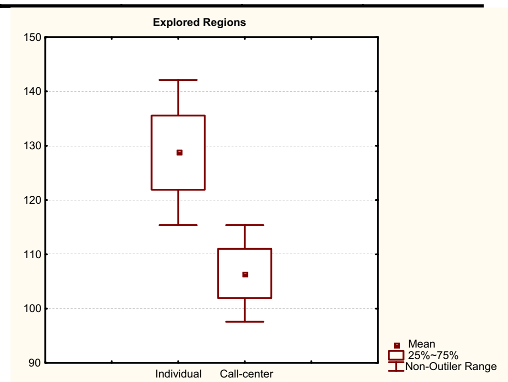

Figure 2. Explored Regions

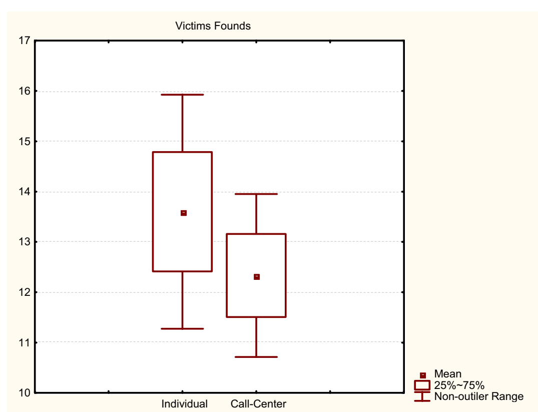

Figure 3. Victims Found

Table 2 Control Mode

Variables	Individual (N = 15)		Call-Center (N = 15)		T-value	P
	$\bar{x}$	SD	$\bar{x}$	SD		
Switch No.	236.2	71.22	234.7	104.7	0.046	0.963
Mission issued	188.9	61.18	144.0	43.66	2.312	0.028
Waypoints issued/Mission	1.763	0.403	1.784	0.456	-0.132	0.895

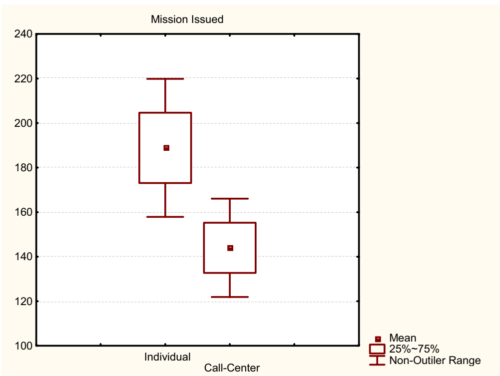

Figure 4. Mission Numbers

More robots were completely neglected in the *call-center* condition t (28) = -2.512, p = 0.018, and still more robots were operated only once, t (28) = -3.148, p = 0.004.

Table 3. Neglected Robots

Number of Robots	Individual	Call-Center
Totally	2.00	4.26
After the Initial Move	4.73	7.66

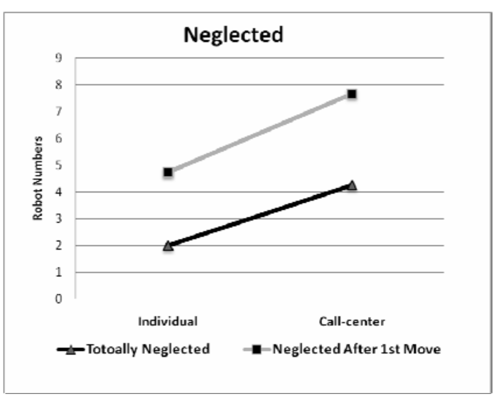

Figure 5. Neglected Robots

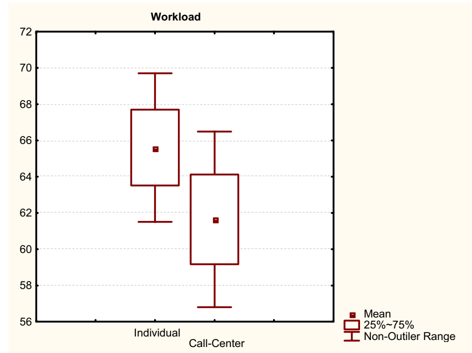

Figure 4. NASA-TLX Measurement of Mental workload

No significant difference (p=0.225) in mental workload was found between the conditions although the graph suggests a small advantage for the *call-center.*

IV. Discussion

These data provide a baseline control for future investigations of the effects of different forms of robot autonomy on these control regimes. In the present experiment participants in the *call-center* condition were confronted with a bank of videos (Figure 1) much like a security guard monitoring too many surveillance cameras. Under these conditions coordination demands were ambiguous. Participants were frequently observed to reach some form of agreement for dividing robots to be controlled at the outset (like a self-organized individual condition). These discussions are reflected in the large numbers of "team coordination" and "closed loop" communications found for this group. Later as they performed the task some robots with apparently ambiguous assignments were never moved out from the center of the building. The effects of this diffusion of responsibility can be seen in the greater numbers of neglected robots in the *call-center* condition and the smaller areas they searched with fewer robots. It is notable that despite comparable levels of switching between robots and complexity of tasking, participants in the *individual* condition tasked their robots more frequently and felt marginally more overworked doing so. We speculate that under the experimental conditions participants in the *individual* condition felt responsible for operating the full team of robots assigned to them, a number we know to be at the limit of their capabilities. The demands on call-center participants, by contrast, were more ambiguous leading to less frequent re-tasking and feelings of load. Anecdotally, in this experiment *call-center* participants were never observed to assign or take control of particular robots. Instead participants universally took or ceded responsibility for searching regions of the building. This task-centered approach to robot assignment is precisely the sort of behavior the *call-center* regime was designed to foster raising hopes that we may yet find the regime's advantages under more favorable conditions of robot autonomy.

ACKNOWLEDGMENT

This research was supported by AFOSR Grant FA9550-07-1-0039 and 2008 AFOSR FA9550-08-1-0356.

REFERENCES

[1] 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.

[2] J. Crandall, M. Goodrich, D. Olsen, and C. Nielsen. Validating human-robot interaction schemes in multitasking environments. IEEE Transactions on Systems, Man, and Cybernetics, Part A, 35(4):438–449, 2005.

[3] J. Wang and M. Lewis, Assessing coordination overhead in control of robot teams, Proceedings of 2007 IEEE International Conference on Systems, Man, and Cybernetics, 2645-2649, 2007.

[4] M. Goodrich, B. Morse, C. Engh, J. Cooper, and J. Adams. Towards Using UAVs in Wilderness Search and Rescue: Lessons from Field Trials. To appear in Interaction Studies Journal, special issue on "Robots in the Wild: Exploring Human-Robot Interaction in Naturalistic Environments"

[5] M. Lewis, J. Polvichai, K. Sycara, and P. Scerri, Scaling-up Human Control for Large UAV Teams, In N. Cooke (Ed.), The Human Factors of Remotely Piloted Vehicles, New York: Elsevier, pp. 237-250, 2006.

[6] M. Harchol-Balter, M. Crovella, and C. Murta. On Choosing a Task Assignment Policy for a Distributed Server System. TOOLS '98: Proceedings of the 10th International Conference on Computer Performance Evaluation: Modelling Techniques and Tools, pp. 231 -- 242, 1998

[7] H. Wang, M. Lewis, P. Velagapudi, P. Scerri, and K. Sycara, How search and its subtasks scale in N robots 2009 Human Robot Interaction Conference, ACM.

[8] (UE 2) UnrealEngine2, http://udn.epicgames.com/Two/rsrc/Two/KarmaReference/KarmaUserGuide.pdf, accessed February 5, 2008.

[9] S. Carpin, T. Stoyanov, Y. Nevatia, M. Lewis and J. Wang. Quantitative assessments of USARSim accuracy". Proceedings of PerMIS 2006

[10] S. Carpin, J. Wang, M. Lewis, A. Birk and A. Jacoff. High fidelity tools for rescue robotics: Results and perspectives, Robocup 2005 Symposium, 2005.

[11] Mathengine, MathEngine Karma User Guide, http://udn.epicgames.com/Two/KarmaReference/KarmaUserGuide.pdf, accessed May 3, 2005.

[12] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, C. Scrapper. (2006b). Bridging the gap between simulation and reality in urban search and rescue. Robocup 2006: Robot Soccer World Cup X, Springer, Lecture Notes in Artificial Intelligence, 2006.

[13] M. Lewis, S. Hughes, J. Wang, M. Koes, and S. Carpin. Validating USARsim for use in HRI research, Proceedings of the 49th Annual Meeting of the Human Factors and Ergonomics Society, Orlando, FL, 2005, 457-461

[14] C. Pepper, S. Balakirsky, and C. Scrapper. Robot Simulation Physics Validation, Proceedings of PerMIS'07, 2007.

[15] B. Taylor, S. Balakirsky, E. Messina and R. Quinn. Design and Validation of a Whegs Robot in USARSim, Proceedings of PerMIS'07, 2007.

[16] M. Zaratti, M. Fratarcangeli and L. Iocchi. A 3D Simulator of Multiple Legged Robots based on USARSim. Robocup 2006: Robot Soccer World Cup X, Springer, LNAI, 2006.

[17] S. Balakirsky, S. Carpin, A. Kleiner, M. Lewis, A. Visser, J. Wang and V. Zipara. Toward hetereogeneous robot teams for disaster mitigation: Results and performance metrics from RoboCup Rescue, Journal of Field Robotics, 2007