Entrapment/Escorting and Patrolling Missions in Multi-Robot Cluster Space Control

Ignacio Mas, Steven Li, Jose Acain and Christopher Kitts

***Abstract*— The tasks of entrapping/escorting and patrolling around an autonomous target are presented making use of the multi-robot cluster space control approach. The cluster space control technique promotes simplified specification and monitoring of the motion of mobile multi-robot systems of limited size. Previous work has established the conceptual foundation of this approach and has experimentally verified and validated its use for 2-robot, 3-robot and 4-robot systems, with varying implementations ranging from automated trajectory control to human-in-the-loop piloting. In this publication, we show that the problem of entrapping/escorting/patrolling is trivial to define and manage from a cluster space perspective. Using a 3-robot experimental testbed, results are shown for the given tasks. We also revise the definition of the cluster space framework for a three-robot formation and incorporate a robot-level obstacle avoidance functionality.**

***Index Terms*— cluster space, multi-robot systems, formation control, robot teams, escorting.**

I. Introduction

Robotic systems offer many advantages to accomplishing a wide variety of tasks given their strength, speed, precision, repeatability, and ability to withstand extreme environments [1]. While most robots perform these tasks in an isolated manner, interest is growing in the use of tightly interacting multi-robot systems to improve performance in current applications and to enable new capabilities. Potential advantages of multi-robot systems include redundancy, increased coverage and throughput, flexible reconfigurability and spatially diverse functionality. For mobile systems, one of the key technical considerations is the technique used to coordinate the motions of the individual vehicles. A wide variety of techniques have been and continue to be explored. Behavioral methods [2], [3], [4] are based on designing a set of actions or behaviors for each element in the group such that desirable group behavior emerges as a result. Another approach makes use of potential fields to represent the desired formation pattern and trajectory [5]. An alternative method uses leader-follower patterns [6], [7]. A variant of this is leader-follower chains, in which follower robots control their position relative to one or more local leaders which, in turn, are following other local leaders in a network that ultimately are led by a designated leader [8].

This work has been sponsored through a variety of funding sources to include Santa Clara University Technology Steering Committee grant TSC131 and National Science Foundation Grant No. CNS-0619940. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

The authors are with the Robotic Systems Laboratory, Santa Clara University, 500 El Camino Real, Santa Clara CA 95053, United States. {iamas,jacain,ckitts}@scu.edu

Despite the wide range of multi-robot coordination methods, few papers explicitly address the escort/entrapment task [9]. Our work explores this issue using the cluster space control method, a specific centralized approach for robot clusters of limited size (on the order of ones to tens) and locale (such that global communication is available).

The motivation of the cluster space [1] approach is to promote the simple specification and monitoring of the motion of a mobile multi-robot system. This strategy conceptualizes the n-robot system as a single entity, a cluster, and desired motions are specified as a function of cluster attributes, such as position, orientation, and geometry. These attributes guide the selection of a set of independent system state variables suitable for specification, control, and monitoring. These state variables form the system's cluster space. Cluster space state variables may be related to robot-specific state variables, actuator state variables, etc. through a formal set of kinematic transforms. These transforms allow cluster commands to be converted to robot-specific commands, and for sensed robot-specific state data to be converted to cluster space state data. As a result, a supervisory operator or real-time pilot can specify and monitor system motion from the cluster perspective. Our hypothesis is that such interaction enhances usability by offering a level of control abstraction above the robot- and actuator-specific implementation details.

Previous work presented a generalized framework for developing the cluster space approach for a system of n robots, each with m degrees of freedom (DOF)[1]. This framework was successfully demonstrated for both holonomic and non-holonomic two-robot systems, including several cluster-space-based versions of regulated motion [10], automated trajectory control [11], [12], human-in-the-loop piloting [13], [14], and potential field-based obstacle avoidance [15], [16], [17]. The method was also implemented for three-robot [18] and four-robot [19] non-holonomic systems.

II. Cluster Space Representation of a Three-Robot System

To further develop the application of the cluster space framework, we have applied it to the specification and control of three differential drive robots operating in a plane [18]. This section reviews the selection of cluster space variables, and presents the derivation of an exact set of kinematic transforms; an approximate, simplified set of transforms has been presented and demonstrated in previous work [18], [20].

Preprint submitted to 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. Received February 13, 2009.

A. Cluster Space State Variable Selection

Fig. 1 depicts the relevant reference frames for the planar 3-robot problem. We have chosen to locate the cluster frame {C} at the cluster's centroid, oriented with Y_c pointing toward Robot 1. Based on this, the nine robot space state variables (three robots with three DOF per robot) are mapped into nine cluster space variables for a nine DOF cluster.

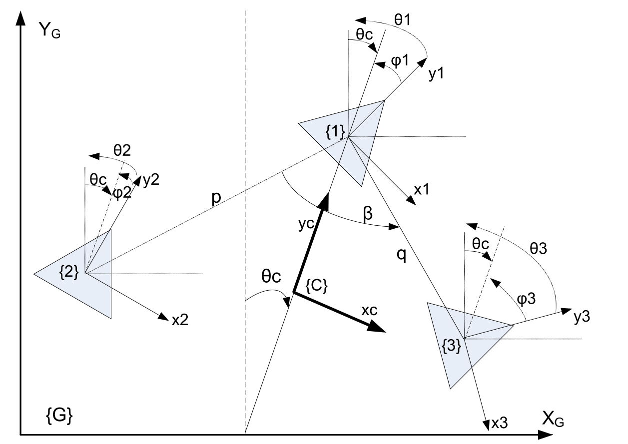

Fig. 1. Reference Frame Definition Placing Cluster Center at Triangle Centroid

Given the parameters defined by Fig. 1, the robot space pose vector is defined as:

$$\overrightarrow{R} = (x_1, y_1, \theta_1, x_2, y_2, \theta_2, x_3, y_3, \theta_3)^T, \quad (1)$$

where $(x_i, y_i, \theta_i)^T$ defines the position and orientation of robot i. The cluster space pose vector definition is given by:

$$\overrightarrow{C} = (x_c, y_c, \theta_c, \phi_1, \phi_2, \phi_3, p, q, \beta)^T, \quad (2)$$

where $(x_c, y_c, \theta_c)^T$ is the cluster position and orientation, ϕ_i is the yaw orientation of rover i relative to the cluster, p and q are the distances from rover 1 to rover 2 and 3, respectively, and β is the skew angle with vertex on rover 1.

B. Kinematic Transformations

Given the aforementioned selection of cluster space state variables, it is possible to express the forward and inverse position kinematics of the three-robot system. The forward position kinematics are given by:

$$x_c = \frac{x_1 + x_2 + x_3}{3}, \quad (3)$$

$$y_c = \frac{y_1 + y_2 + y_3}{3}, \quad (4)$$

$$\theta_c = atan2\left(\frac{2x_1 - x_2 - x_3}{2y_1 - y_2 - y_3}\right), \quad (5)$$

$$\phi_1 = \theta_1 + \theta_c, \quad (6)$$

$$\phi_2 = \theta_2 + \theta_c, \quad (7)$$

$$\phi_3 = \theta_3 + \theta_c, \quad (8)$$

$$p = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}, \quad (9)$$

$$q = \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2}, \quad (10)$$

$$\beta = acos\frac{p^2 + q^2 - (x_3 - x_2)^2 - (y_3 - y_2)^2}{2pq}, \quad (11)$$

and the inverse position kinematics are therefore defined by:

$$x_1 = x_c + \frac{1}{3}\sqrt{\kappa}\sin(\theta_c), \quad (12)$$

$$y_1 = y_c + \frac{1}{3}\sqrt{\kappa}\cos(\theta_c), \quad (13)$$

$$\theta_1 = \phi_1 - \theta_c, \quad (14)$$

$$x_2 = x_c + \frac{1}{3}\sqrt{\kappa}\sin(\theta_c) + p\cos(\gamma), \quad (15)$$

$$y_2 = y_c + \frac{1}{3}\sqrt{\kappa}\cos(\theta_c) + p\sin(\gamma), \quad (16)$$

$$\theta_2 = \phi_2 - \theta_c, \quad (17)$$

$$x_3 = x_c + \frac{1}{3}\sqrt{\kappa}\sin(\theta_c) + q\cos(\beta + \gamma), \quad (18)$$

$$y_3 = y_c + \frac{1}{3}\sqrt{\kappa}\cos(\theta_c) + q\sin(\beta + \gamma), \quad (19)$$

$$\theta_3 = \phi_3 - \theta_c, \quad (20)$$

where

$$\kappa = p^2 + q^2 + 2pq\cos(\beta), \quad (21)$$

and

$$\gamma = atan2\left(\frac{q\sin(\beta)}{p + q\cos(\beta)}\right) - atan2\left(\frac{-\cos(\theta_c)}{-\sin(\theta_c)}\right). \quad (22)$$

By differentiating the forward and inverse position kinematics, the forward and inverse velocity kinematics can easily be derived, obtaining the Jacobian and Inverse Jacobian matrices. Symbolically:

$$\dot{\overrightarrow{C}} = J(\overrightarrow{R}) * \dot{\overrightarrow{R}}, \quad (23)$$

where

$$J(\overrightarrow{R}) = \begin{pmatrix} \frac{\partial c_1}{\partial r_1} & \frac{\partial c_1}{\partial r_2} & \cdots & \frac{\partial c_1}{\partial r_9} \\ \frac{\partial c_2}{\partial r_1} & \frac{\partial c_2}{\partial r_2} & \cdots & \frac{\partial c_2}{\partial r_9} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial c_9}{\partial r_1} & \frac{\partial c_9}{\partial r_2} & \cdots & \frac{\partial c_9}{\partial r_9} \end{pmatrix}, \quad (24)$$

and conversely:

$$\dot{\overrightarrow{R}} = J^{-1}(\overrightarrow{C}) * \dot{\overrightarrow{C}}, \quad (25)$$

where

$$J^{-1}(\overrightarrow{C}) = \begin{pmatrix} \frac{\partial r_1}{\partial c_1} & \frac{\partial r_1}{\partial c_2} & \cdots & \frac{\partial r_1}{\partial c_9} \\ \frac{\partial r_2}{\partial c_1} & \frac{\partial r_2}{\partial c_2} & \cdots & \frac{\partial r_2}{\partial c_9} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial r_9}{\partial c_1} & \frac{\partial r_9}{\partial c_2} & \cdots & \frac{\partial r_9}{\partial c_9} \end{pmatrix}. \quad (26)$$

Due to limited space, the full algebraic expressions for $J(\overrightarrow{R})$ and $J^{-1}(\overrightarrow{C})$ are not included. It can be verified that $J(\overrightarrow{R}) * J^{-1}(\overrightarrow{C}) = I_{(9x9)}$.

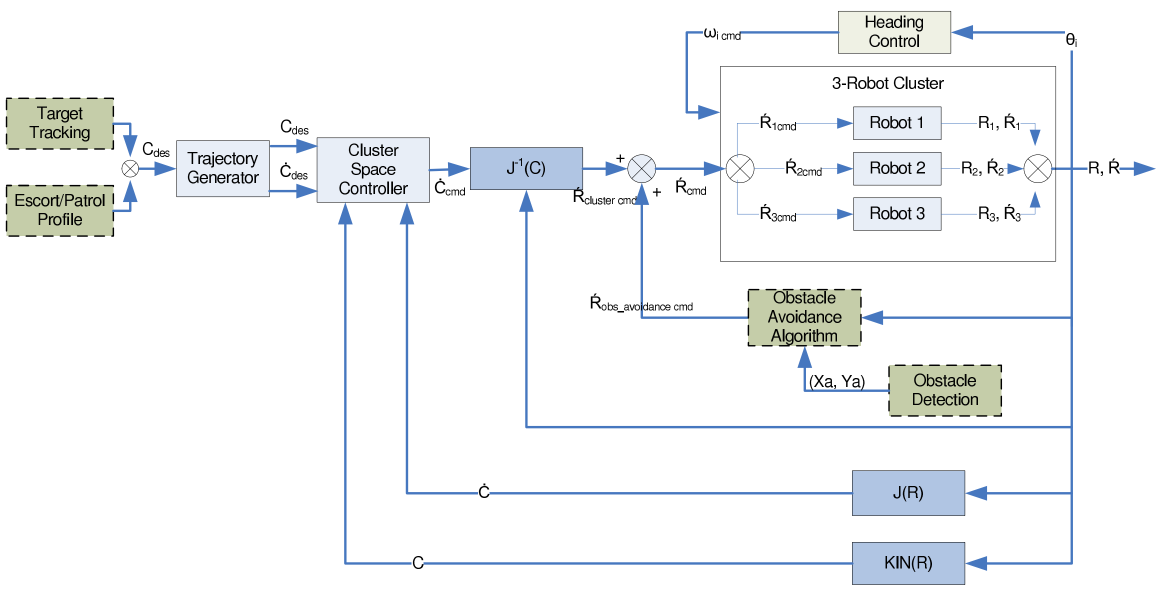

Fig. 2. Cluster Space Control Architecture for a Mobile Three-Robot System. In this cluster space control architecture, desired motions and control actions are computed in the cluster space; control actions are converted to the robot space through the use of the inverse Jacobian relationship and combined with control actions from the Obstacle Avoidance algorithm.

C. Control Framework

Fig. 2 presents the control architecture for trajectory-based cluster space control of the experimental three-robot system. A cluster level PID controller compares cluster position and velocity with desired trajectory values and outputs cluster commanded velocities, which are translated into individual robot velocities through the inverse Jacobian. State data from the robots are converted to cluster space information through the forward kinematics and Jacobian and fed back into the controller to close the loop.

The non-holonomic constraints given by the differential-drive motion of the robots effectively reduce the number of independently specified cluster pose variables to six. As a consequence, inner-loop robot-level heading control is implemented on each robot and the cluster space controller does not regulate the three cluster parameters corresponding to yaw orientation of the robots relative to the cluster, specifically ϕ_i.

The architecture of Fig. 2 differs from that presented in previous work [18] in two aspects, each of which is indicated in the diagram with dotted lines. First, the trajectory generator is fed by two new functions that are used to specify the desired cluster space variable trajectory. The first is a target tracking function that measures the target's position; this information is typically used to specify the centroid location for the cluster. The second is an escort/patrol profile manager that specifies the desired values for the remaining cluster variables based on the manner in which the escort/patrol function should be implemented. This is explained in Section III-A. The second change to the control architecture is the inclusion of obstacle avoidance functions. The position of the obstacle is tracked, and an algorithm produces avoidance velocities that are combined with commanded velocities from the inverse jacobian function. The avoidance algorithm is described in Section III-B.

TABLE I

ESCORTING MISSION DEFINITION IN TERMS OF CLUSTER PARAMETERS

Cluster Parameters	Mission		
	Escorting	Patrolling	Oriented Escorting
Centroid (x_c, y_c)	at Target	at Target	at Target
Orientation (θ_c)	Constant	$\omega \cdot t$	$Atan2(V_y, V_x)$
Shape (p, q, β)	Constant	Constant	Constant

III. THE ESCORTING MISSION

The mission of escorting can be seen as the task of surrounding and maintaining a formation around a target whose movement is not known a priori but can be measured in real-time [9]. As the target moves, the formation moves to keep the target at its centroid, maintains the distance from the formation vehicles to the target, and evenly distributes the formation vehicles around the target. An oriented escorting task can be considered to be escorting when the orientation of the formation is aligned with the direction of motion of the target. A patrolling task can be defined as a further extension where the formation rotates around the target for surveillance purposes.

Entrapment may be specified similarly. In fact, from a motion control point of view, it may be implemented in precisely the same way although the purpose would be to reduce target escape windows rather than preventing intrusion by external agents.

A. Escorting/Patrolling in Cluster Space Control

The cluster control approach is a natural way of specifying the escorting/patrolling problem: the cluster centroid tracks the target as the shape of the cluster specifies the platoon

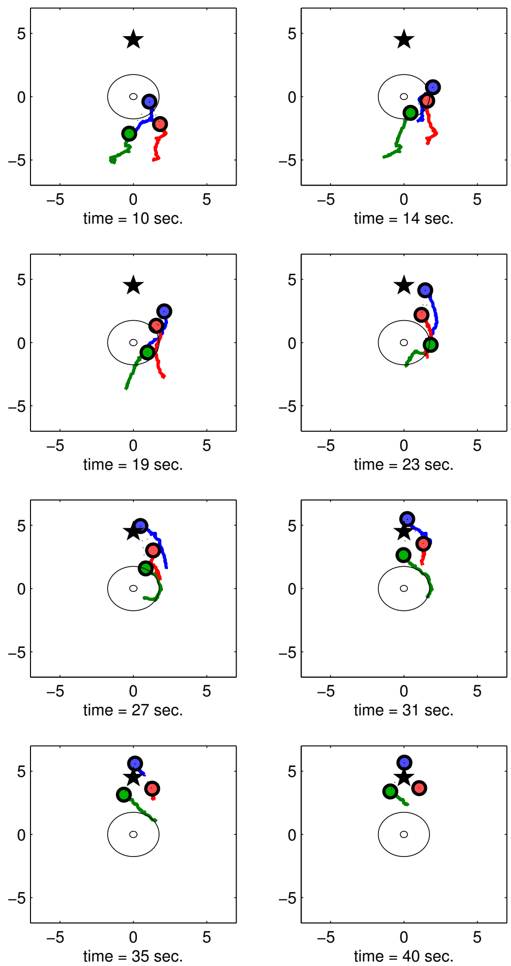

Fig. 3. Test results showing a 3-robot cluster (colored circles) negotiating an obstacle -at (0,0)- while getting to its desired final position (star). The detection and avoidance regions are shown as concentric circles. As the formation moves and the robots get inside the obstacle detection region, the potential field comes into play. The axes represent global x and y coordinates in meters.

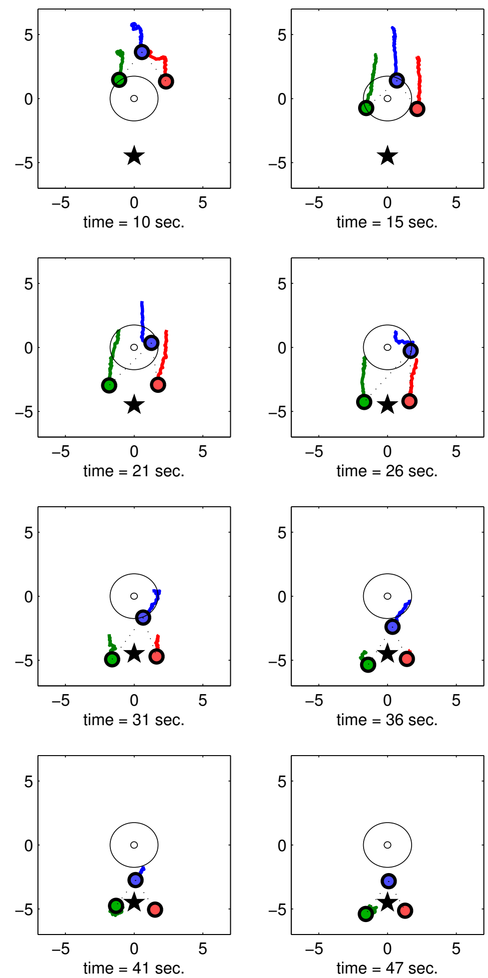

Fig. 4. Test results showing a 3-robot cluster (colored circles) negotiating an obstacle -at (0,0)- while getting to its desired final position (star). The detection and avoidance regions are shown as concentric circles. This time, robot 2 (green) goes past the obstacle by the left while robots 1 and 3 go past it by the right. The axes represent global x and y coordinates in meters.

distribution around it. This approach also allows for simple specification of the patrolling task. This can be achieved by simply varying the cluster orientation parameter θ_c which results in the rotating motion of the robots around the cluster centroid. In the same way, specifying the desired cluster orientation as a function of the target velocity defines the required behavior of the oriented escorting task. Table I shows the cluster parameter specifications that meet the requirements for the different missions.

B. Obstacle Avoidance

An additional feature required for this task is obstacle avoidance. The robots must not collide with the target at any time. To prevent this from happening, a robot-level avoidance function is implemented. Given the robot i position (x_i, y_i), the obstacle position (x_a, y_a) and the distance function

$$d_{ai} = \sqrt{\left(\frac{x_i - x_a}{\nu}\right)^2 + \left(\frac{y_i - y_a}{\xi}\right)^2}, \tag{27}$$

where $\nu, \xi > 0$, the avoidance potential function implemented [21], [22], [23] is:

$$V_{ai} = \left(min\left\{0, \frac{d_{ai}^2 - R^2}{d_{ai}^2 - r^2}\right\}\right)^2, \tag{28}$$

where $R > r > 0$. R and r are the radii of detection and avoidance regions, respectively. This function is infinite at the boundary of the avoidance region and zero outside the detection region. To break the symmetry when it occurs, different

Preprint submitted to 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems. Received February 13, 2009.

shapes of the potential function, for example ellipsoids, can be obtained by choosing different values for the coefficients ν, ξ. The partial derivatives of V_a with respect to x and y are used to command the robots away from the obstacle. The combined robot-level velocity commands for robot i in global coordinates are then

$$\dot{x}_{i\ cmd} = \dot{x}_{i\ cluster} + k_{ax}\frac{\partial V_{ai}}{\partial x}, \qquad (29)$$

and

$$\dot{y}_{i\ cmd} = \dot{y}_{i\ cluster} + k_{ay}\frac{\partial V_{ai}}{\partial y}, \qquad (30)$$

where $(\dot{x}_{i\ cluster}, \dot{y}_{i\ cluster})$ are the commanded velocities for robot i given by the inverse jacobian output and (k_{ax}, k_{ay}) are the obstacle avoidance control gains.

To demonstrate the functionality of the obstacle avoidance algorithm, experiments where an obstacle is in the path of the formation are conducted. Fig. 3 and Fig. 4 show test results using the testbed described in section IV-A. The formation shape is momentarily altered while the robots negotiate the obstacle but it is recovered after that. For these tests, robots are not considered to be obstacles to the other robots and it is assumed that the formation controller keeps the robots from colliding with each other.

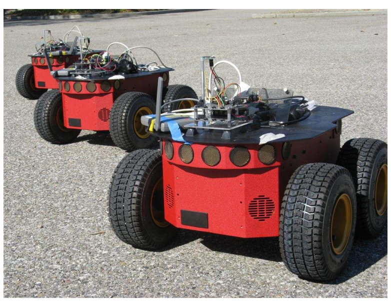

Fig. 5. Pioneer Robots from MobileRobots Inc. with custom sensor and communication suites.

IV. Experiments

In this section, a brief description of the testbed utilized followed by results of the execution of an escorting mission and a patrolling mission are presented.

A. Testbed

Experimental tests were conducted using four differential drive robots with custom sensor suites based on Ultra-Wide Band (UWB) tracking technology [18]. Onboard radio-modems relay robot state information to a central computer that executes the control algorithms and transmits back velocity commands. One of the robots is in this case the target unit, which follows a predefined trajectory that is unknown to the cluster. The position of the target is then used as the input to the 3-robot cluster x_c and y_c parameters. Fig. 5 shows the robots used in these experiments with their sensor and communication suites.

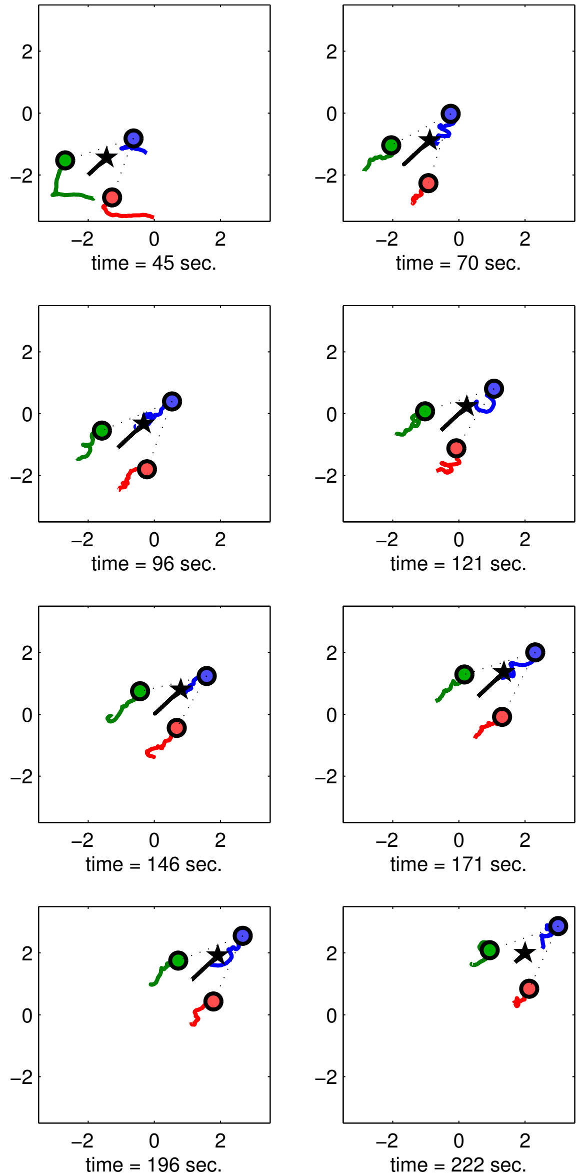

Fig. 6. Escorting test results using the multi-robot testbed showing a 3-robot cluster (circles) following the target (star) while maintaining the triangular formation. The axes represent global x and y coordinates in meters.

B. Results

The first result is from an escorting mission where the target follows a straight path. The formation centroid tracks the target and the robots in the formation keep a relative distance of 2 meters. Fig. 6 shows the position of the target and escorting robots at different points in time. In the first frame the cluster robots get in position around the target avoiding any collisions with it, then the target and the

Preprint submitted to 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. Received February 13, 2009.

formation start moving. After 200 seconds the target stops and the formation stays in position around it. The tracking mean square errors (difference between target and cluster centroid positions) in x and y, after the initial transients are over, are 0.027 meters and 0.063 meters, respectively.

The second mission is the patrolling task. This time, the target again follows a rectilinear trajectory but the formation rotates around the target keeping the relative distances. This is achieved simply by defining a linear trajectory for the cluster heading parameter θ_c. Fig. 7 shows the motion of the individual robots resulting from this specification. Again, the robots get to their position and start tracking the target while patrolling around it. The tracking mean square errors (difference between target and cluster centroid positions) in x and y, after the initial transients are over, are 0.023 meters and 0.013 meters, respectively.

V. Future Works and Conclusions

Ongoing works include the study of alternative cluster definitions under the assumption that they may result more convenient for specifying and monitoring requirements for different missions. Upgrading the obstacle avoidance algorithm to deal with multiple obstacles is also under study. A new vision-based multi-robot testbed is being developed which will allow for more accurate position measurements as well as faster loop rates. A formal stability analysis of the cluster control architecture is currently being developed.

The tasks of escorting and patrolling around a moving target were addressed in the context of cluster space control. The problem was specified in terms of simple cluster parameter trajectories resulting in complex robot motions that satisfied the required formation behavior. An obstacle avoidance algorithm based in potential field techniques was implemented at the robot level to prevent collisions between the formation robots and the target. Experiments using an UWB-based multi-robot testbed demonstrated the functionality of the proposed obstacle avoidance approach. The testbed was then used to execute successfully an escorting mission where the robots stay in formation around a moving target and a patrolling mission where the robot formation rotates around the moving target. A simple specification of the tasks in cluster space produced complex robot-level behaviors that achieved the proposed missions.

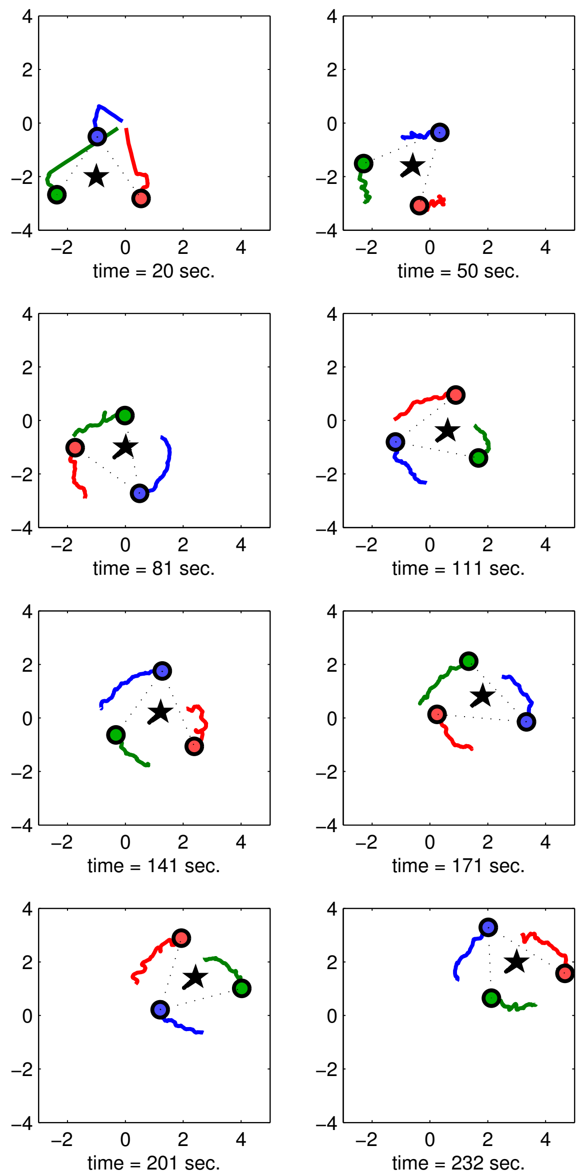

Fig. 7. Patrolling mission results using the multi-robot testbed showing a 3-robot cluster (circles) following the target (star) while rotating around it (patrolling) and maintaining the triangular formation. The robots get in formation around the target (time=20s) and then start tracking it. The trails show the resulting robot motions of the patrolling while escorting task. The axes represent global x and y coordinates in meters.

VI. Acknowledgments

The authors gratefully thank Dr. Edward Schaefer for his help deriving the cluster inverse kinematic expressions. We also want to thank Michael Neumann and Jocelyn Chang for their cooperation running the experiments.

References

[1] C. Kitts and I. Mas, “Cluster space specification and control of mobile multi-robot systems,” *accepted for publication in IEEE/ASME Transactions on Mechatronics*, 2009.

[2] T. Balch and R. Arkin, “Behavior-based formation control for multi-robot teams,” *Robotics and Automation, IEEE Transactions on*, vol. 14, no. 6, pp. 926–939, Dec 1998.

[3] V. Gazi and K. Passino, “Stability analysis of swarms,” *Automatic Control, IEEE Transactions on*, vol. 48, no. 4, pp. 692–697, April 2003.

[4] R. Saber and R. Murray, “Flocking with obstacle avoidance: cooperation with limited communication in mobile networks,” *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 2, pp. 2022–2028 Vol.2, Dec. 2003.

[5] S. S. Ge, C.-H. Fua, and K. W. Lim, “Multi-robot formations: queues and artificial potential trenches,” *Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004 IEEE International Conference on*, vol. 4, pp. 3345–3350 Vol.4, 26-May 1, 2004.

[6] J. Desai, J. Ostrowski, and V. Kumar, “Controlling formations of

Preprint submitted to 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. Received February 13, 2009.

multiple mobile robots," *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 4, pp. 2864–2869 vol.4, May 1998.
[7] H. Takahashi, H. Nishi, and K. Ohnishi, "Autonomous decentralized control for formation of multiple mobile robots considering ability of robot," *Industrial Electronics, IEEE Transactions on*, vol. 51, no. 6, pp. 1272–1279, Dec. 2004.
[8] A. Das, R. Fierro, V. Kumar, J. Ostrowski, J. Spletzer, and C. Taylor, "A vision-based formation control framework," *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 5, pp. 813–825, Oct 2002.
[9] G. Annelli, F. Arrichiello, and S. Chiaverini, "The entrapment/ escorting mission for a multi-robot system: Theory and experiments," *Advanced intelligent mechatronics, 2007 ieee/asme international conference on*, pp. 1–6, Sept. 2007.
[10] R. Ishizu, "The design, simulation and implementation of multi-robot collaborative control from cluster perspective," Master's thesis, Santa Clara University, Dec. 2005.
[11] P. Connolley, "Design and implementation of a cluster space trajectory controller for multiple holonomic," Master's thesis, Santa Clara University, Jun. 2006.
[12] T. To, "Automated cluster space trajectory control of two nonholonomic robots," Master's thesis, Santa Clara University, Jun. 2006.
[13] M. Kalkbrenner, "Design and implementation of a cluster space human interface controller, including applications to multi-robot piloting and multi-robot object manipulation," Master's thesis, Santa Clara University, Jun. 2006.
[14] B. Tully, "Cluster space piloting of a non-holonomic multi-robot system," Master's thesis, Santa Clara University, Jun. 2006.
[15] P. Chindaphorn, "Cluster space obstacle avoidance for two nonholonomic robots," Master's thesis, Santa Clara University, Jun. 2006.
[16] K. Stanhouse, "Cluster space obstacle avoidance for mobile multi-robot systems," Master's thesis, Santa Clara University, Jun. 2006.
[17] C. Kitts, K. Stanhouse, and P. Chindaphorn, "Cluster space collision avoidance for mobile multi-robot systems," in draft, available at http://rsl.engr.scu.edu /publications.html.
[18] I. Mas, O. Petrovic, and C. Kitts, "Cluster space specification and control of a 3-robot mobile system," *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 3763–3768, May 2008.
[19] E. Girod, "Design and implementation of four robot cluster space control," Master's thesis, Santa Clara University, Jun. 2008.
[20] I. Mas, J. Acain, O. Petrovic, and C. Kitts, "Error characterization in the vicinity of singularities in multi-robot cluster space control," *2008 IEEE Robotics and Biomimetics, Bangkok, Thailand. IEEE International Conference on*, feb 2009.
[21] S. Mastellone, D. Stipanovic, C. Graunke, K. Intlekofer, and M. Spong, "Formation control and collision avoidance for multi-agent nonholonomic systems: Theory and experiments," *The International Journal of Robotics Research*, vol. 27, no. 1, pp. 107–126, jan 2008.
[22] D. Stipanovic, P. Hokayem, M. W. Spong, and D. Siljak, "Cooperative avoidance control for multiagent systems," *Journal of Dynamic Systems, Measurement and Control*, vol. 129, no. 5, pp. 699–707, 2007.
[23] G. Leitmann and J. Skowronski, "Avoidance control," *Journal of Optimization Theory and Applications*, vol. 23, pp. 581–591, 1977.