2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Cluster Space Specification and Control of a 3-Robot Mobile System

Ignacio Mas, Ognjen Petrovic, Christopher Kitts
Robotic Systems Laboratory, Santa Clara University
500 El Camino Real, Santa Clara CA 95053
iamas@scu.edu, opetrovic@scu.edu

Abstract— **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 two diverse 2-robot systems and with varying implementations ranging from automated trajectory control to human-in-the-loop piloting. In this paper, we present the cluster space control of a 3-robot system. In doing so, we develop the fundamental kinematic relationships, illustrate the closed-loop control framework, describe the simulation and hardware testbed environments used for verification, and present initial experimental results of the successfully implemented system.**

I. INTRODUCTION [7]

Robotic systems offer many advantages to accomplishing a wide variety of tasks given their strength, speed, precision, repeatability, and ability to withstand extreme environments. 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. Because of the physical distribution of components and the potential for limited information exchange, decentralized control approaches hold great promise [1]-[3],[6], and these techniques have been explored for a variety of systems [4],[5]. Centralized approaches exploiting global information are often not preferred due to limited scalability and the challenges of maintaining the necessary communication links for many of the applications explored. Our work, however, explores a specific centralized approach for potential application to robot clusters of limited size (on the order of ones to tens) and locale (such that global communication is available) with the understanding that other control modes may be required for augmentation in order to achieve robust performance.

The motivation of the cluster space [7] 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. Our current work focuses on systems of limited number of robots in which each robot is capable of closed-loop velocity control, a level of functionality typically built into a variety of commercially available robotic platforms.

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)[7]. This framework was successfully demonstrated for both holonomic and non-holonomic two-robot systems, including several cluster-space-based versions of regulated motion [8], automated trajectory control [9]-[10], human-in-the-loop piloting [11]-[12], and potential field-based obstacle avoidance [13]-[15].

II. CLUSTER SPACE REPRESENTATION OF A THREE-ROBOT ROVER 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. This section reviews the selection of cluster space variables, the derivation of the relevant kinematic transforms, a brief description of the cluster configuration limitations given by the cluster singularities, and the formulation of an appropriate control architecture.

A. Cluster Space State Variable Selection

Figure 1 depicts the relevant reference frames for the planar 3-robot problem. Because of the sensor data used in experimentation, the global frame conventions were selected as follows: Y_g points to the North, X_g points to the East, and θ_g is the compass-measured heading. For our work, 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.

978-1-4244-1647-9/08/$25.00 ©2008 IEEE.

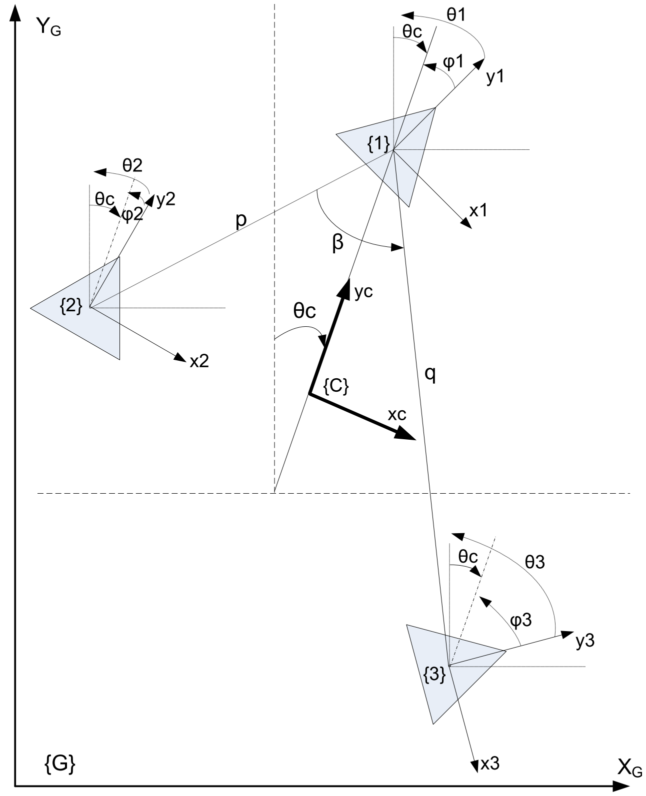

Fig. 1. Reference Frame Definition Placing Cluster Center at Triangle Centroid

Given the parameters defined by Figure 1, the cluster space state variable definition is given by:

$$\overrightarrow{C} = (x_c, y_c, \theta_c, \phi_1, \phi_2, \phi_3, p, q, \beta)^T \tag{1}$$

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.

The robot space state variable is defined as:

$$\overrightarrow{R} = (x_1, y_1, \theta_1, x_2, y_2, \theta_2, x_3, y_3, \theta_3)^T \tag{2}$$

where $(x_i, y_i, \theta_i)^T$ defines the position and orientation of robot i.

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:

$$x_c = \frac{x_1 + x_2 + x_3}{3} \tag{3}$$

$$y_c = \frac{y_1 + y_2 + y_3}{3} \tag{4}$$

$$\theta_c = atan2 \frac{2/3(x_1) - 1/3(x_2 + x_3)}{2/3(y_1) - 1/3(y_2 + y_3)} \tag{5}$$

$$\phi_1 = \theta_1 + \theta_c \tag{6}$$

$$\phi_2 = \theta_2 + \theta_c \tag{7}$$

$$\phi_3 = \theta_3 + \theta_c \tag{8}$$

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

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

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

and the inverse position kinematics are therefore defined by:

$$x_1 = x_c + (1/3) r \sin\theta_c \tag{12}$$

$$y_1 = y_c + (1/3) r \cos\theta_c \tag{13}$$

$$\theta_1 = \phi_1 - \theta_c \tag{14}$$

$$x_2 = x_c + (1/3) r \sin\theta_c - p \sin(\beta/2 + \theta_c) \tag{15}$$

$$y_2 = y_c + (1/3) r \cos\theta_c - p \cos(\beta/2 + \theta_c) \tag{16}$$

$$\theta_2 = \phi_2 - \theta_c \tag{17}$$

$$x_3 = x_c + (1/3) r \sin\theta_c + q \sin(\beta/2 - \theta_c) \tag{18}$$

$$y_3 = y_c + (1/3) r \cos\theta_c - q \cos(\beta/2 - \theta_c) \tag{19}$$

$$\theta_3 = \phi_3 - \theta_c \tag{20}$$

where $r = \sqrt{(q + p\cos\beta)^2 + (p \sin\beta)^2}$.

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:

$$\dot{\overrightarrow{C}} = J(\overrightarrow{R}) * \dot{\overrightarrow{R}} \tag{21}$$

where

$$J(\overrightarrow{R}) = \begin{pmatrix} \frac{\partial c_1}{\partial r_1} & \frac{\partial c_1}{\partial r_2} & \cdots & \frac{\partial c_1}{\partial r_9} \\ \frac{\partial c_2}{\partial r_1} & \frac{\partial c_2}{\partial r_2} & \cdots & \frac{\partial c_2}{\partial r_9} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial c_9}{\partial r_1} & \frac{\partial c_9}{\partial r_2} & \cdots & \frac{\partial c_9}{\partial r_9} \end{pmatrix} \tag{22}$$

and conversely:

$$\dot{\overrightarrow{R}} = J^{-1}(\overrightarrow{C}) * \dot{\overrightarrow{C}} \tag{23}$$

where

$$J^{-1}(\overrightarrow{C}) = \begin{pmatrix} \frac{\partial r_1}{\partial c_1} & \frac{\partial r_1}{\partial c_2} & \cdots & \frac{\partial r_1}{\partial c_9} \\ \frac{\partial r_2}{\partial c_1} & \frac{\partial r_2}{\partial c_2} & \cdots & \frac{\partial r_2}{\partial c_9} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial r_9}{\partial c_1} & \frac{\partial r_9}{\partial c_2} & \cdots & \frac{\partial r_9}{\partial c_9} \end{pmatrix} \tag{24}$$

Due to limited space, the full algebraic expressions for $J(\overrightarrow{R})$ and $J^{-1}(\overrightarrow{C})$ are not included here. They are presented in [16].

C. Cluster Singularities

In robotic manipulator chains, singularities occur in configurations where the Jacobian and inverse Jacobian matrices become singular[17]. Expanding this notion for cluster space configurations, singularities will occur where either $J(\overrightarrow{R})$ or $J^{-1}(\overrightarrow{C})$ become singular. For the three-robot mobile system presented in this paper, singularities occur where the geometry of the cluster becomes degenerated. In this

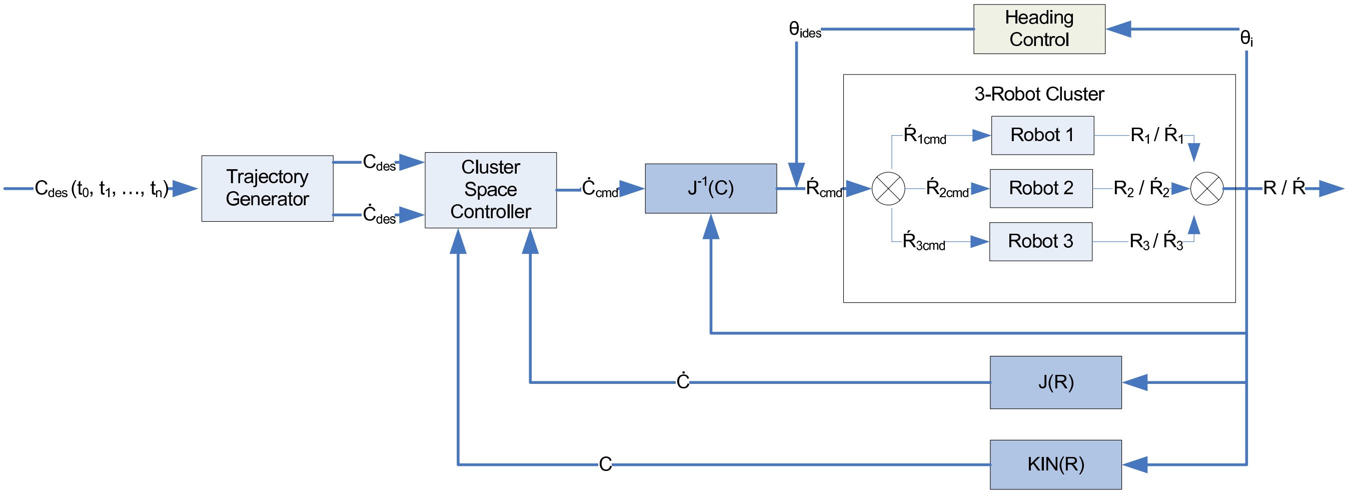

Fig. 2. Cluster Space Control Architecture for a Mobile Three-Robot System. In this cluster space control architecture, desired motions and control actions are computed in the cluster space; control actions are converted to the robot space through the use of the inverse Jacobian relationship.

particular cluster specification, singular configurations are found when $p = 0$, $q = 0$ or $\beta = 0, \pi$. The first two are defeated by collision avoidance algorithms, but singularities in the skew angle β may be of concern. Ongoing work is addressing this issue. So far, we have developed alternative configuration definitions where the same cluster pose can be attained without a singularity occurrence. For the tests presented in following sections, the cluster is maintained away from the mentioned singular configurations.

D. Control Framework

Figure 2 presents the control architecture for trajectory-based cluster space control of the experimental three-robot system described later in this paper. 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. Data from the robots are converted to cluster space information through the forward kinematics and Jacobian and fed back into the controller.

The non-holonomic constrain given by the differential-drive motion of the robots effectively reduces the cluster to a six DOF system. As a consequence, an 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.

III. SYSTEM SIMULATION

To facilitate development and evaluation of the cluster space concept, we developed a simulator using the Matlab/Simulink environment. This simulator includes a simple three-dimensional world representation of robot motion using the Virtual Reality Markup Language (VRML) Toolbox. The simulator supports the use of robot kinematic and dynamic models of several holonomic and non-holonomic multi-robot systems available for experimentation[11]-[14]. The simulator supports evaluation of both automatic controllers as well as interactive, human-in the-loop controllers through the use of joystick inputs[18]. Figure 3 shows the 3-D virtual-reality model of the three-robot cluster.

IV. HARDWARE TESTBED

A cluster control testbed provides experimental capabilities for multi-robot command-and-control and collaboration experiments. Santa Clara University students developed and over the years upgraded the testbed to support variable number of robots controllable over the Internet. The system is designed to scale up to 10 to 20 robots. In order to conduct the experiments described in this paper, Amigobot rovers from ActivMedia Robotics[19] were utilized. This testbed has been successfully used in the past to demonstrate a variety of 2 robot cluster experiments including trajectory following and obstacle avoidance[8][10][12][13].

A. Testbed System Overview

The testbed employs a distributed architecture where independent processes manage different types of inputs and outputs. The main components are a) operator side autonomous controller or human in the loop piloting interface; b) network data server gluing inputs and outputs together; c) robot side hardware drivers; d) robots; e) auxiliary sensors, utilities or analysis tools. Figure 4 summarizes the high level architecture.

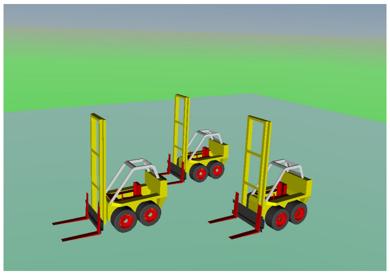

Fig. 3. Simulated 3-D Virtual-reality model of a 3-robot cluster.

B. Testbed Software Description

A cluster controller running under the Simulink/Matlab environment drives the control process from the operator side. The cluster follows joystick commands, a trajectory, or an independent cluster center device such as another robot. The operator can visualize the state of the system via a virtual reality display. The controller interfaces with RBNB Data Turbine to receive sensor readings and to send driving commands. Simulink/Matlab controls the cluster in real time, and therefore trades off closed loop control performance in favor of rapid prototyping and ease of software integration.

RBNB Data Turbine, a flexible, open-source, data server supports a variety of communication modes over the TCP/IP network[20]. This project employs the channelized publish/subscribe communication method. Each robot has two associated channels, one for control commands and one for sensor telemetry. The Data Turbine adds a time varying packet delay and connection latency affecting the overall control performance. A previous study on the Data Turbine performance shows the control of slow moving robots through the Data Turbine Internet connection is possible[21].

The Amigobot driver toolkit translates controller commands to signals that Amigobot robots understand and feeds sensor telemetry packets back to the controller. The Amigobot drivers talk to the robots via radio modems and to the controller via Data Turbine. The drivers are built using Aria, an object oriented, multi threaded, open-source toolkit for Amigobot control provided by the Amigobot manufacturer. We extended the toolkit to understand communication via the Data Turbine and added a new driver for radio modem hardware. The power of Amigobot toolkit resides in its ability to efficiently control a wide variety of robotic vehicles, not just Amigobots.

MobileSim is an open-source Amigobot simulator, also provided by the Amigobot manufacturer. It supports multiple robots and it has a map editor to position robots in the workspace. The purpose of this simulator is to reduce on site tuning and debugging time. For example, the effects of network delay, controller performance gains or switching to different class of robots can fairly accurately be debugged using this simulator. It also provides an intuitive customization interface for users to design and simulate their own robots.

C. Testbed Hardware Description

Amigobots are differential drive robots capable of translation speeds up to 0.75m/s and rotational speeds $300°/s$. Amigobots receive commands and send out telemetry over a 900MHz radio link. The communication link preserves data integrity, but it does not guaranty packet delivery. SCU students designed custom sensor and communication subsystems consisting of a Garmin 18 differential GPS unit, digital Devantech compass and a Ricochet 128Kbits/s radio modem. These subsystems are controlled by BasicX microcontrollers linked through RS-232 interfaces. The package is capable of outputting telemetry at a 5Hz rate.

The current design goal of the testbed is flexibility. The Robotic Systems Lab conducts different types of cluster experiments on different types of robotic platforms. For example, cluster space behavior is being examined applied to robotic kayaks and remotely controlled airplanes. A flexible testbed means fast and simple transitioning from development to deployment phase, integration of new local or distant sensors, remote teleoperation using video feedback and ease of integration with a variety of software tools and programming languages.

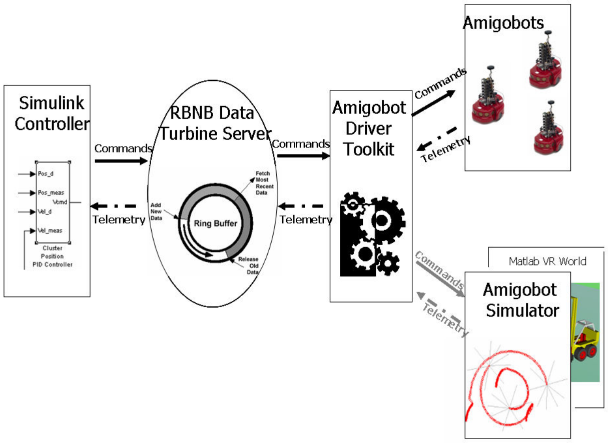

Fig. 4. Experimental Testbed block diagram

Fig. 5. Amigobot rover with custom subsystems

V. EXPERIMENTAL RESULTS

We have successfully implemented our cluster space controller within the described hardware testbed in order to experimentally demonstrate closed loop cluster space specification and control. This section presents the results from trajectory-controlled cluster parameter variations and composed motions test cases. In performing these experiments, performance was limited by the quality of our sensors; the GPS and digital compass components were specified to a general accuracy of +/-3m and +/-5 degrees, respectively. PID controllers were implemented for cluster as well as heading control. The main objective of these tests was to validate the cluster space definition and architecture, therefore basic functionality was pursued and no considerable efforts were made towards performance optimization.

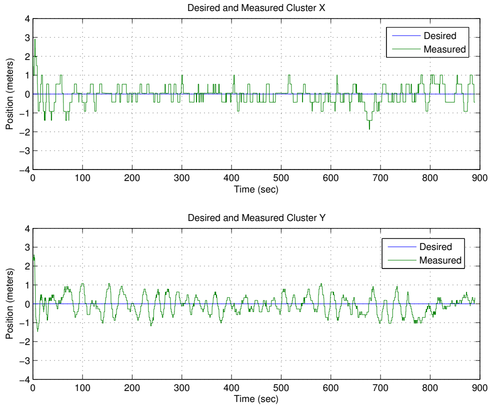

Fig. 6. Cluster geometry variation test results - Cluster position

A. *Cluster parameter variation tests*

Our initial experiments were performed specifying simple geometry parameter variations while keeping constant desired values of cluster position. For one such demonstration, the cluster was defined with initial parameters $p = q = 10m$, a cluster orientation of $\theta_c = 0°$ and a skew angle $\beta = 90°$. The trajectory generator provided successive variations on θ_c, p, q and β. Figures 6 and 7 show the results of one such experiment. As can be seen, the cluster kept its position to within 2 meters while properly following the cluster geometry parameter trajectories. The same trajectory was executed by the Matlab/Simulink simulator resulting in minimal deviations from desired positions due to the absence of sensor errors. It is worth noting in Figure 7 that errors in some parameters depend on values of others -and their proximity to singular regions-, indicated here by an increment in θ_c error as β gets close to π.

B. *Cluster trajectory with rotation tests*

Another experiment was performed specifying a sinusoidal translation motion composed with a cluster rotation motion while keeping constant desired values of the additional cluster parameters. In this case, the cluster was defined with parameters $p = q = 10m$ and a skew angle $\beta = 90°$. The trajectory generator provided a circular path with a radius of 5 m, adding then a cluster rotation of $1.5°/s$. Figures 8 and 9 show the output of one this tests. After the transient, the cluster follows the trajectory within 2 meters and the θ_c desired is accurately tracked. The remaining parameters are regulated to stay close to the desired values. Detailed statistical measures of performance are presented in [16]. Individual robot position errors due to GPS time-varying inaccuracies -mostly multipath errors- produce momentarily different position offsets for each robot, either canceling each other or magnifying the total cluster parameter errors as shown, for instance, for β in Figure 9 at around 350 seconds.

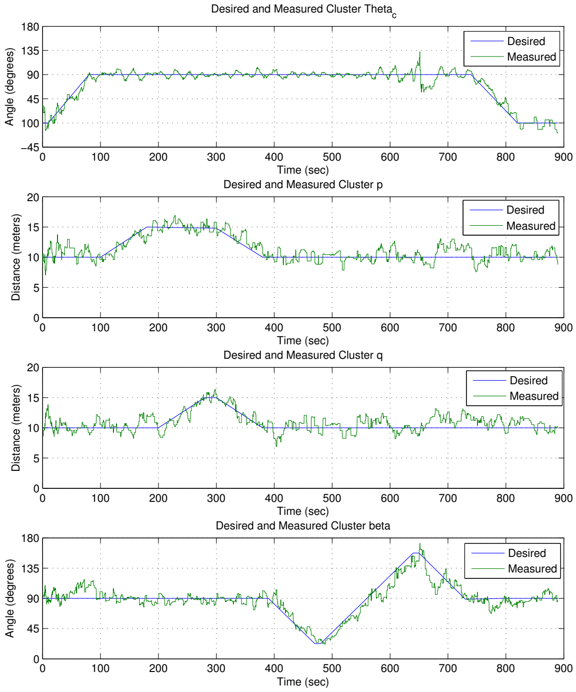

Fig. 7. Cluster geometry variation test results - Cluster parameters

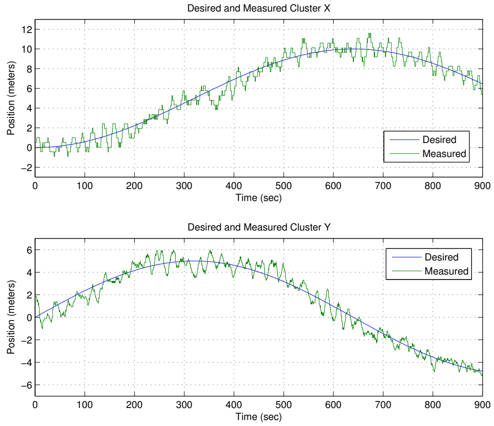

Fig. 8. Cluster translation and rotation test results - Cluster position

VI. FUTURE WORKS AND CONCLUSIONS

A. *Future Works*

Ongoing works include definitions of alternative geometries for three-robot clusters intended to address task-specific needs, including the study of cluster singularities and their dependency on the chosen cluster geometry. In the future, we also plan to explore the scalability of this approach to systems with more robots and additional degrees of freedom; this will include experimental demonstrations that exploit the array of our robotic devices that operate in land, sea, air and space [22]. To address tractability issues, we plan to

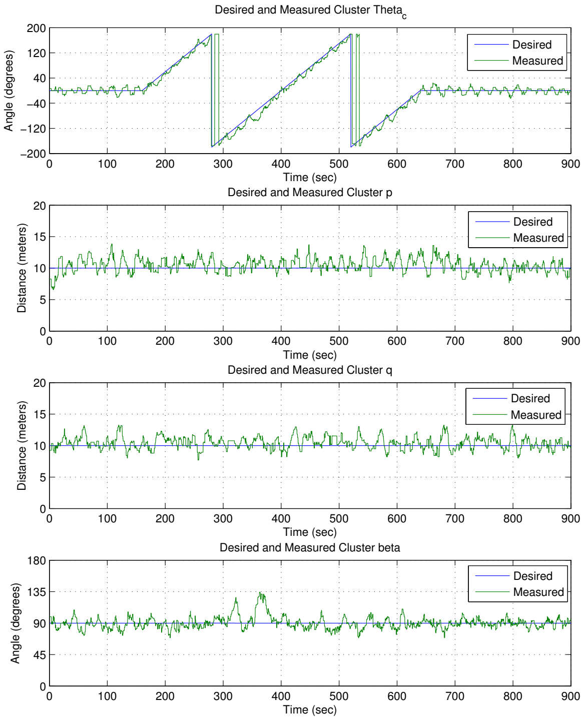

Fig. 9. Cluster translation and rotation test results - Cluster parameters

investigate the use of dual rate control approaches in which the inverse Jacobian is updated at a rate slower than the primary servo rate. In addition, we will examine methods of linking application-oriented task specifications to cluster space primitives in order to support goal directed behaviors of the multi-robot system.

B. Conclusions

The cluster space state representation of mobile multi-robot systems was conceptually presented and experimentally evaluated for a three-robot system as a means of specifying and controlling the desired mobility characteristics of mobile multi-robot systems. Formal kinematic equations relating cluster space state variables with those required for system actuation were developed for a three-robot mobile system and a cluster-level control architecture was laid out. The resulting system was simulated thoroughly in the Matlab/Simulink environment. Then, a multi-platform Internet-based hardware testbed was utilized to run various experiments. As a result, the three-robot cluster space definition and control architecture was validated and basic functionality was proven. Our results indicate that this control approach allows cluster space motions to be specified and monitored by a single operator or pilot, even when the equivalent robot-specific motions are quite complex. Our ongoing and future work in this field is focused on translating this capability into enhanced performance and cost-effective improvements in the operator/robot ratio for controlling multi-robot systems.

VII. ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of Ryan Ishizu, Ray Chindaphorn, Tran To, Brian Tully and Mike Rasay. 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.

REFERENCES

[1] D. Siljak, Decentralized Control of Complex Systems, *Academic* 1991.
[2] M. Ikeda, Decentralized control of large-scale systems, Three Decades of Mathematical Systems Theory, *H. Nijmeijer and J. Schumacher, Eds., Springer-Verlag*, Berline, 135, 1989, pp. 219-242.
[3] T. Yang, Networked control systems: a historical review and current research topics, *Measurement and Control*, v 38, Feb., 2005, p 12-16.
[4] J. Feddema, C. Lewis, and P. Klarer, Control of multiple robotic sentry vehicles, *Proceedings of SPIE - The International Society for Optical Engineering*, v 3693, 1999, p 212-223
[5] R. Carpenter, Decentralized control of satellite formations, *International Journal of Robust and Nonlinear Control*, v 12, n 2-3, February/March, 2002, Control of Satellite Formations, p 141-161.
[6] S. Stankovic, M. Stanojevic, and D. Siljak. Decentralized overlapping control of a platoon of vehicles. IEEE Transactions on Control Systems Technology, v 8, n 5, Sep, 2000, p 816-831.
[7] C. Kitts, Cluster Space Specification and Control of Mobile Multi-Robot systems: Part 1 Conceptual Framework, *Submitted to IEEE/ASME Transactions on Mechatronics*.
[8] R. Ishizu, The Design, Simulation and Implementation of Multi-Robot Collaborative Control from Cluster Perspective. Advisor: C. Kitts, *Santa Clara Univ. Master of Science in Elec. Eng. Thesis*, Dec 2005.
[9] P. Connolley, Design and Implementation of a Cluster Space Trajectory Controller for Multiple Holonomic Robots. Advisor: C. Kitts, *Santa Clara Univ. Master of Science in Mech. Eng. Thesis*, June 2006.
[10] T. To, Automated Cluster Space Trajectory Control of Two Non-Holonomic Robots. Advisor: C. Kitts, *Santa Clara Univ. Master of Science in Computer Eng. Thesis*, June 2006.
[11] M. Kalkbrenner, Design and Implementation of a Cluster Space Human Interface Controller,. . . Advisor: C. Kitts. *Santa Clara Univ. Master of Science in Mech. Eng. Thesis*, June 2006.
[12] B. Tully, Cluster Space Piloting of a Non-holonomic Multi-Robot System. Advisor: C. Kitts. *Santa Clara Univ. Master of Science in Computer Eng. Thesis*, June 2006.
[13] P. Chindaphorn, Cluster Space Obstacle Avoidance for Two Non-Holonomic Robots. Advisor: C. Kitts. *Santa Clara University Master of Science in Computer Engineering Thesis*, June 2006.
[14] K. Stanhouse, Cluster Space Obstacle Avoidance for Mobile Multi-Robot Systems. Advisor: C. Kitts. *Santa Clara University Master of Science in Electrical Engineering Thesis*, June 2006.
[15] 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*.
[16] I. Mas, Three-Robot Cluster Specification Document, *http://rsl.engr.scu.edu/publications.html*, 2007
[17] C. Gosselin and J. Angeles, Singularity Analysis of Closed-Loop Kinematic Chains, *IEEE Transactions on Robotics and Automation*, Vol. 6 No. 3, June, 1990.
[18] M. Neumann and C. Kitts, Control of Two Holonomic Robots Using a Camera-Referenced Specification of Motion,*Collection of Technical Papers AIAA 3rd Unmanned-Unlimited Technical Conference, Workshop, and Exhibit, Chicago, IL*, September 20-23, 2004.
[19] ActivMedia Robotics. *http://www.activrobots.com/*.
[20] RBNB Data Turbine, Creare Inc. *http://rbnb.creare.com/*.
[21] O. Petrovic, C. Kitts, R. Rasay, and M. MacKinnon, NETROL: An Internet-Based Control Architecture for Robotic Teleoperation, *Submitted to 2008 IEEE Aerospace Conference*.
[22] C. Kitts, Surf, Turf, and Above the Earth: An Aggressive Robotics Development Program for Integrative Undergraduate Education. *IEEE Robotics and Automation Magazine*, v 10, n 3, Sept 2003, pp 30-36.