Cluster Space Specification and Control of Mobile Multirobot Systems

Christopher A. Kitts, *Senior Member, IEEE*, and Ignacio Mas, *Student Member, IEEE*

***Abstract*—The *cluster space* state representation of mobile multirobot systems is introduced as a means of enabling enhanced control of mobile multirobot systems. A conceptual framework is proposed for the selection of appropriate cluster space state variables for an n-robot system, the development of formal kinematics that associate the cluster space state variables with robot-specific variables, and the implementation of a cluster space control system architecture. The cluster space approach is then demonstrated for examples of two- and three-robot clusters consisting of differential drive robots operating in a plane. In these examples, we demonstrate cluster space variable selection, review the critical kinematic relationships, and present experimental results that demonstrate the ability of the systems to meet control specifications while allowing a single operator to easily specify and supervise the motion of the clusters.**

***Index Terms*—Cluster space, collaborative control, formation control, multirobot systems, robot teams.**

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. While most robots perform these tasks in an isolated manner, interest is growing in the use of tightly interacting multirobot systems to improve performance in current applications and to enable new capabilities. Potential advantages of multirobot systems include redundancy, increased coverage and throughput, flexible reconfigurability, spatially diverse functionality, and the fusing of physically distributed sensors and actuators. Applications capable of exploiting such features range from remote and *in situ* sensing to the physical manipulation of objects, and the domains for such applications include land, sea, air, and space.

This vision, however, faces numerous technical challenges that must be overcome in order to field cost-effective multirobot systems; these challenges include interrobot communication, relative position sensing and actuation, control paradigms appropriate to real-time multisystem control, the fusion of distributed sensors/actuators, man–machine interfaces allowing efficient human direction/supervision of these systems, and design approaches supporting the economical production of such systems. These topics are being explored for multirobot systems operating in a wide range of domains to include satellite formations [1]–[5], robotic flocks of aircraft [6]–[8], land rover clusters [9]–[11], underwater robot fleets [12]–[14], and manipulator workcells [15]–[17].

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 [18]–[20], and these techniques have been explored for a variety of systems [11], [16], [21], [22]. Behavioral, biologically inspired, and potential field techniques have been demonstrated with great success [23], [24] although they often lack the mathematical formality of other control approaches. Centralized approaches exploiting global information are also of interest [25], [26], but they are often not preferred due to limited scalability and the challenges of maintaining the necessary communication links for many of the applications explored. However, they may be reasonable when interaction among the robots is required by applications such as the real-time fusing of sensors or actuators [27]–[29].

The work presented in this paper explores a specific centralized control approach for 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.

II. Cluster Space Framework

The motivation of this research is to promote the simple specification and monitoring of the motion of a mobile multirobot system. Central to our strategy are the concepts of considering the n-robot system as a single entity, a *cluster*, and specifying motions with respect to cluster attributes, such as position, orientation, and geometry. Our approach is to use these attributes to guide the selection of a set of independent system state variables suitable for specification, control, and monitoring. This collection of state variables constitute the system's *cluster space* and can be related to robot-specific state variables, actuator state variables, etc., through a formal set of kinematic transforms. A supervisory operator or real-time pilot specifies and monitors system motion, and centralized control computations are executed with respect to the cluster space variables. Kinematic transforms allow compensation commands to be derived for each individual robot (and ultimately for each actuator on each

Manuscript received August 28, 2008; revised October 23, 2008. First published February 24, 2009; current version published April 15, 2009. Recommended by Guest Editor P. X. Liu. This work was supported in part by the National Science Foundation under Grants EIA0079815, E1A0082041, and CNS0619940, in part by the National Aeronautics and Space Administration (NASA) Ames Research Center under Grant NNA04CK64A, and in part by Santa Clara University under Grant TSC131.

The authors are with the Robotic Systems Laboratory, Santa Clara University, Santa Clara, CA 95053 USA (e-mail: ckitts@scu.edu; iamas@scu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMECH.2009.2013943

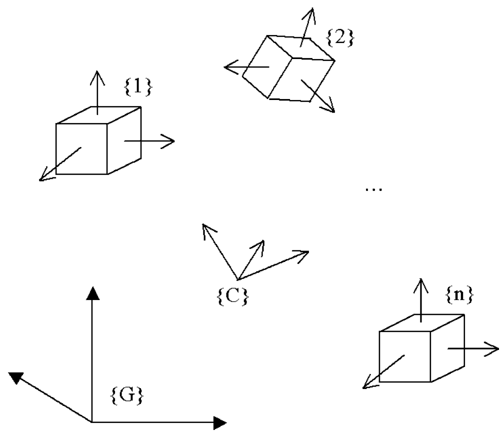

Fig. 1. Robot pose using conventional versus cluster space representations. A conventional description of a cluster of robots provides individual robot frame descriptions with respect to a global frame, typically in the form of a homogeneous transform T. The cluster space description establishes a cluster reference frame; references to individual robots in the cluster are made with respect to this cluster frame.

robot), and they also allow data from a variety of sensor packages to be converted to cluster space state estimates. Our current paper focuses on systems 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.

As an example of this, consider a group of two robots that may be driven in a plane. The cluster space view of this simple multirobot system could be represented as a line segment at a certain location, oriented in a specific direction, and with a particular size. A pilot could "drive" the cluster along an arbitrary path while varying the orientation and size of the line segment. Similarly, a three-robot planar system could be represented as a triangle at a certain location, oriented in a certain direction, and with a specific shape. The pilot could "drive" this cluster along an arbitrary path while varying the shape and size of the triangle. The triangle could be "flattened" into a straight line while driving through a narrow passage. Overall, the cluster space approach allows the pilot to specify and monitor motions from the cluster space perspective, with automated kinematic transformations converting this point of view to and from robot-specific drive commands and sensor data.

The first step in the development of the cluster space control architecture is the selection of an appropriate set of cluster space state variables. To do this, we introduce a cluster reference frame and select a set of state variables that capture key pose and geometry elements of the cluster.

A. Introduction of a Cluster Reference Frame

Consider the simplified, general case of an n-robot system where each robot has the same m DOF and an attached body frame, as depicted in Fig. 1. Let $m = p + r$, where p is the number of translational DOFs and r is the number of rotational DOFs for each robot.

Typical robot-oriented representations of pose use (*nm*) variables to represent the position and orientation of each of the robot body frames, $\{1\}$, $\{2\}$, ..., $\{n\}$, with respect to a global frame $\{G\}$. These may be formalized as n robot-specific homogeneous transforms, ${}^G_1T, {}^G_2T, \ldots, {}^G_nT$, where for robot i

$${}^G_iT = \begin{pmatrix} {}^G_iR & {}^GP_{i_{\text{org}}} \\ [0 \quad 0 \quad 0] & 1 \end{pmatrix} \tag{1}$$

where ${}^G_iR$ is a rotation matrix denoting the orientation of the $\{i\}$ frame with respect to $\{G\}$, and ${}^GP_{i_{\text{org}}}$ is a vector specifying the translation of the origin of frame $\{i\}$ with respect to $\{G\}$.

In contrast, consideration of the cluster space representation starts with the definition of a cluster frame $\{C\}$, with a pose denoted by ${}^G_CT$. The pose of each robot is expressed relative to the cluster frame: ${}^C_1T$, ${}^C_2T, \ldots, {}^C_nT$, such that for robot i

$${}^G_iT = {}^G_CT * {}^C_iT. \tag{2}$$

With this formulation, the n-robot system's (*nm*) DOFs are now represented by $[(n + [1])m]$ cluster space variables in $(n + 1)$ homogeneous transforms with m cluster space constraints.

We note that the positioning of the $\{C\}$ frame with respect to the n robots is often critical in achieving a cluster space framework that benefits the operator/pilot. In practice, $\{C\}$ is often positioned and oriented in a manner with geometric significance, such as at the cluster's centroid and oriented toward the "lead" vehicle or alternatively, coincident with a lead vehicle's body frame. The selected placement policy dictates the nature of the m cluster space constraints that are present among the $[(n + [1])m]$ cluster space variables.

B. Selection of Cluster Space State Variables

We select as our state variables a set of position variables (and their derivatives) that capture the cluster's pose and geometry. These variables $c_1, c_2, \ldots, c_{mn}$ are generally related to (and in many cases are equivalent to) the cluster-oriented pose variables used in ${}^G_CT$ and ${}^C_1T$, ${}^C_2T, \ldots, {}^C_nT$. For the general case of n 6-DOF robots, where the pose variables of $\{C\}$ with respect to $\{G\}$ are $(x_c, y_c, z_c, \alpha_c, \beta_c, \gamma_c)$ and where the pose variables for robot i with respect to $\{C\}$ are $(x_i, y_i, z_i, \alpha_i, \beta_i, \gamma_i)$ for $i = 1,2, \ldots, n$

$$\begin{aligned} c_1 &= f_1(x_c, y_c, z_c, \alpha_c, \beta_c, \gamma_c, x_1, y_1, z_1, \alpha_1, \beta_1, \gamma_1, \ldots, x_n, \\ &\quad y_n, z_n, \alpha_n, \beta_n, \gamma_n) \\ c_2 &= f_2(x_c, y_c, z_c, \alpha_c, \beta_c, \gamma_c, x_1, y_1, z_1, \alpha_1, \beta_1, \gamma_1, \ldots, x_n, \\ &\quad y_n, z_n, \alpha_n, \beta_n, \gamma_n) \\ &\vdots \\ c_{6n} &= f_{6n}(x_c, y_c, z_c, \alpha_c, \beta_c, \gamma_c, x_1, y_1, z_1, \alpha_1, \beta_1, \gamma_1, \ldots, x_n, \\ &\quad y_n, z_n, \alpha_n, \beta_n, \gamma_n). \end{aligned} \tag{3}$$

The appropriate selection of cluster state variables may be a function of the application, the system's design, and subjective criteria such as operator preference. In practice, however, we have found great value in selecting state variables based on the metaphor of a virtual rigid body that can move through space while being arbitrarily scaled and articulated. This leads to the use of several general categories of cluster pose variables (and their derivatives) that specify cluster position, cluster orientation, relative robot-to-cluster orientation, and cluster shape.

Cluster position variables express the location of the cluster with respect to the global frame as denoted by the components of ${}^G P_{C_{\mathrm{org}}}$; p variables are required to specify this given the assumption that all robots have identical DOFs. Cluster orientation variables specify the orientation of the cluster frame with respect to the global frame as denoted by the fundamental rotations represented by ${}^G_C R$; given the rigid body metaphor, this requires $o = 3$ variables for systems where $n \geq p = 3$ or $o = p - 1$ variables for other cases, where o is the number of cluster orientation variables. The use of cluster position and orientation variables captures the fundamental value of referring directly to cluster pose while specifying the desired motion of the aggregate multirobot system.

Relative robot-to-cluster orientation variables express the relative orientation of each individual robot with respect to the cluster frame as denoted by the fundamental rotations represented by each ${}^G_i R$ for $i = 1$–n, requiring *nr* variables. These variables are often unavailable for independent specification due to nonholonomic constraints; furthermore, for holonomic systems they often are derived collectively given a cluster payload orientation requirement.

Finally, cluster shape variables may be used to express the geometry of the multirobot cluster; given that a total of *nm* cluster space variables are sought, the number of cluster shape variables s is given by

$$s = (nm) - p - o - (nr) = p(n-1) - o \tag{4}$$

$$s = \begin{Bmatrix} p(n-1) - 3, & \text{for } n \geq p = 3 \\ p(n-2) + 1, & \text{otherwise} \end{Bmatrix}. \tag{5}$$

These shape variables allow direct specification of geometrical features of interest, such as cluster size, which are generally functions of, but not identical to, the components of ${}^G P_{i_{\mathrm{org}}}$ for $i = 1$–n. As the number of robots and DOFs grow, we have found the use of such geometric metaphors, such as specific shapes or formation patterns, to be particularly valuable in contributing to an operator's ease of use, with the selection of particular shapes driving the choice of specific variables and their associated specification terminology.

C. Examples of State Variable Selection

Using the methodology introduced in the previous section, Table I specifies the allocation of each different type of cluster pose variable for a variety of two- to four-robot systems with a range of DOFs; for each, an example of how these pose variables might be used is provided. Section V details the simple cases of two and three planar land rovers.

III. Cluster Space Kinematics

We wish to specify multirobot system motion and compute required control actions in the cluster space using cluster state variables selected as described in the previous section. Given that these control actions will be implemented by each individual robot (and ultimately by the actuators within each robot), we develop formal kinematic relationships relating the cluster space variables and robot space variables.

A. Position Kinematics

To do this, we define (*mn*) × 1 robot and cluster pose vectors, ${}^G\vec{R}$ and $\vec{C}$, respectively. These are related through position kinematic relationships

$$\vec{C} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{mn} \end{pmatrix} = \mathrm{KIN}({}^G\vec{R}) = \begin{pmatrix} g_1(r_1, r_2, \ldots, r_{mn}) \\ g_2(r_1, r_2, \ldots, r_{mn}) \\ \vdots \\ g_{mn}(r_1, r_2, \ldots, r_{mn}) \end{pmatrix} \tag{6}$$

$${}^G\vec{R} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_{mn} \end{pmatrix} = \mathrm{INVKIN}(\vec{C}) = \begin{pmatrix} h_1(c_1, c_2, \ldots, c_{mn}) \\ h_2(c_1, c_2, \ldots, c_{mn}) \\ \vdots \\ h_{6n}(c_1, c_2, \ldots, c_{mn}) \end{pmatrix}. \tag{7}$$

For example, for the planar two-robot system from Table I, the robot and cluster pose vectors are

$${}^G\vec{R} = (x_1, y_1, \theta_1, x_2, y_2, \theta_2)^T \tag{8}$$

where the variables (x_i, y_i, θ_i) specify the pose for rover i with respect to the global frame $\{G\}$ and

$$\vec{C} = (x_c, y_c, \theta_c, \theta_{r1}, \theta_{r2}, d)^T \tag{9}$$

where (x_c, y_c, θ_c) specifies the pose of frame $\{C\}$ with respect to frame $\{G\}$, the θ_{ri} angles specify the orientations of each rover with respect to $\{C\}$, and d specifies half of the separation distance between the two rovers.

B. Velocity Kinematics

We also consider the formal relationship between the robot and cluster space velocities, ${}^G\dot{\vec{R}}$ and $\dot{\vec{C}}$. From (6), we may compute the partial derivatives of the cluster space pose variables c_i and develop a Jacobian matrix J which maps robot velocities to cluster velocities in the form of a time-varying linear function

$$\begin{aligned}\dot{\vec{C}} = \begin{pmatrix} \dot{c}_1 \\ \dot{c}_2 \\ \vdots \\ \dot{c}_{mn} \end{pmatrix} &= {}^G J({}^G\vec{R})\,{}^G\dot{\vec{R}} \\ &= \begin{pmatrix} \frac{\partial g_1}{\partial r_1} & \frac{\partial g_1}{\partial r_2} & \cdots & \frac{\partial g_1}{\partial r_{mn}} \\ \frac{\partial g_2}{\partial r_1} & \frac{\partial g_2}{\partial r_2} & \cdots & \frac{\partial g_2}{\partial r_{mn}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_{mn}}{\partial r_1} & \frac{\partial g_{mn}}{\partial r_2} & \cdots & \frac{\partial g_{mn}}{\partial r_{mn}} \end{pmatrix} \begin{pmatrix} \dot{r}_1 \\ \dot{r}_2 \\ \vdots \\ \dot{r}_{mn} \end{pmatrix}.\end{aligned} \tag{10}$$

As the kinematic functions in (6) are generally nonlinear, the Jacobian is a function of ${}^G\vec{R}$ and can be written ${}^G J({}^G\vec{R})$.

TABLE I
EXAMPLE OF VARIABLE SELECTION FOR PROTOTYPICAL MULTIROBOT SYSTEMS

Multi-Robot Cluster Type	# Robots	Robot DOFs (total, translation, rotation)	System DOFs	Cluster Position Variables	Cluster Orientation Variables	Relative Orientation Variables	Shape Variables
general	*n*	*m, p, r*	*nm*	*p*	*If* $n \geq p=3$, $o=3$ *Else*, $o=p-1$	*nr*	*If* $n \geq p=3$, $s=p(n-1)-3$ *Else*, $s=p(n-2)+1$
2 Planar Surface Rovers/Vessels [30]	2	3,2,1	6	2: (x_c, y_c)	1: θ_c	2: θ_{r1}, θ_{r2}	1: Line: d=size
2 Aero/hydro-statically stable Robots	2	4,3,1	8	3: (x_c, y_c, z_c)	2: θ_c, ψ_c (pitch, yaw)	2: φ_{r1}, φ_{r2} (relative yaw)	1: Line: d=size
2 Aircraft or Spacecraft	2	6,3,3	12	3: (x_c, y_c, z_c)	2: θ_c, ψ_c (pitch, yaw)	6: θ_{rn}, ϕ_{rn}, ψ_{rn} for n=1,2 (relative pitch, roll, yaw)	1: Line: d=size
3 Planar Surface Rovers/Vessels [39]	3	3,2,1	9	2: (x_c, y_c)	1: θ_c	3: θ_{r1}, θ_{r2}, θ_{r3}	3: Triangle: d, β, ξ (size, lead angle, skew angle)
3 Aero/hydro-statically stable Robots	3	4,3,1	12	3: (x_c, y_c, z_c)	3: θ_c, ϕ_c, ψ_c (pitch, roll, yaw)	3: θ_{r1}, θ_{r2}, θ_{r3}	3: Triangle: d, β, ξ (size, lead angle, skew angle)
3 Aircraft or Spacecraft	3	6,3,3	18	3: (x_c, y_c, z_c)	3: θ_c, ϕ_c, ψ_c (pitch, roll, yaw)	9: θ_{rn}, ϕ_{rn}, ψ_{rn} for n=1,2,3 (relative pitch, roll, yaw)	3: Triangle: d, β, ξ (size, lead angle, skew angle)
4 Planar Surface Rovers/Vessels [41]	4	3,2,1	12	2: (x_c, y_c)	1: θ_c	4: θ_{r1}, θ_{r2}, θ_{r3}, θ_{r4}	5: Cross: o, p, q, r, β (height, width, height offset, width offset, skew)

In a similar manner, we may develop the inverse Jacobian ${}^G J^{-1}({}^G\vec{R})$, which maps cluster velocities to robot velocities. Computing the partial derivatives of the robot space pose variables from (7) yields

$$
{}^G\dot{\vec{R}} = \begin{pmatrix} \dot{r}_1 \\ \dot{r}_2 \\ \vdots \\ \dot{r}_{mn} \end{pmatrix} = {}^G J^{-1}({}^G\vec{R})\dot{\vec{C}}
$$

$$
= \begin{pmatrix} \dfrac{\partial h_1}{\partial c_1} & \dfrac{\partial h_1}{\partial c_2} & \cdots & \dfrac{\partial h_1}{\partial c_{mn}} \\ \dfrac{\partial h_2}{\partial c_1} & \dfrac{\partial h_2}{\partial c_2} & \cdots & \dfrac{\partial h_2}{\partial c_{mn}} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial h_{mn}}{\partial c_1} & \dfrac{\partial h_{mn}}{\partial c_2} & \cdots & \dfrac{\partial h_{mn}}{\partial c_{mn}} \end{pmatrix} \begin{pmatrix} \dot{c}_1 \\ \dot{c}_2 \\ \vdots \\ \dot{c}_{mn} \end{pmatrix}. \quad (11)
$$

IV. CLUSTER SPACE CONTROL FRAMEWORK

With the formal kinematics defined, we may compose a system controller in which desired motions are specified and control compensations are computed in the cluster space. Fig. 2 depicts a general control architecture of this type. Desired cluster space motions may be specified by a human-in-the-loop pilot interface, by an automated trajectory controller, or as regulation constants. The controller computes the control law in cluster space and issues compensation commands that are transformed to robot space through the use of the inverse Jacobian relationship, as specified by (11). Individual velocity commands are then provided to each of the robots, which, in turn, use their onboard closed-loop velocity control functionality to achieve this command. Cluster space variables are required to compute the required control; therefore, any state data sensed in robot space must be transformed through the use of the forward kinematic relationships, as specified in (6) and (10). We note again that the Jacobian and inverse Jacobian transforms vary as a function of ${}^G\vec{R}$ and must be updated as the cluster changes its pose.

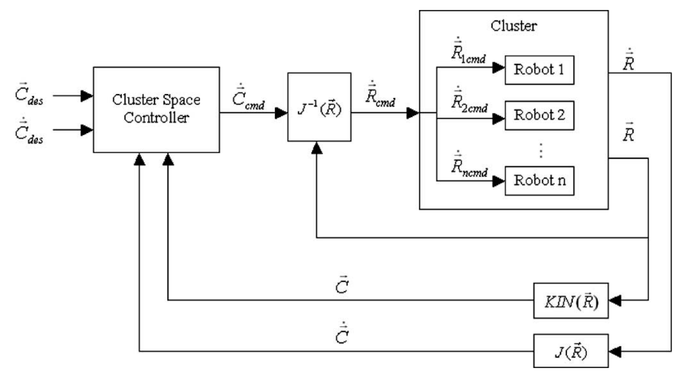

Fig. 2. Cluster space control architecture for a mobile multirobot system. In this cluster space control architecture, desired motions and control actions are computed in the cluster space; control actions are converted to the robot space through the use of the inverse Jacobian relationship.

We have successfully used this control approach to demonstrate cluster-space-based versions of regulated motion [30], automated trajectory control [31], [32], human-in-the-loop piloting [33], [34], partitioned model-based control [35], and potential-field-based obstacle avoidance [36]–[38]. This work has included experiments with two-, three-, and four-robot planar land rover clusters [39]–[41], with two-boat surface vessel systems [42], and for robots that are both holonomic and nonholonomic.

V. EXAMPLE FORMULATIONS FOR TWO- AND THREE-ROBOT CLUSTERS

To demonstrate the simple application of this framework, we have applied it to the specification and control of two-robot and three-robot clusters of wheeled differential drive rovers operating in a plane. This section reviews the selection of cluster space variables for each of these examples; the resulting kinematic transforms are provided in the Appendix.

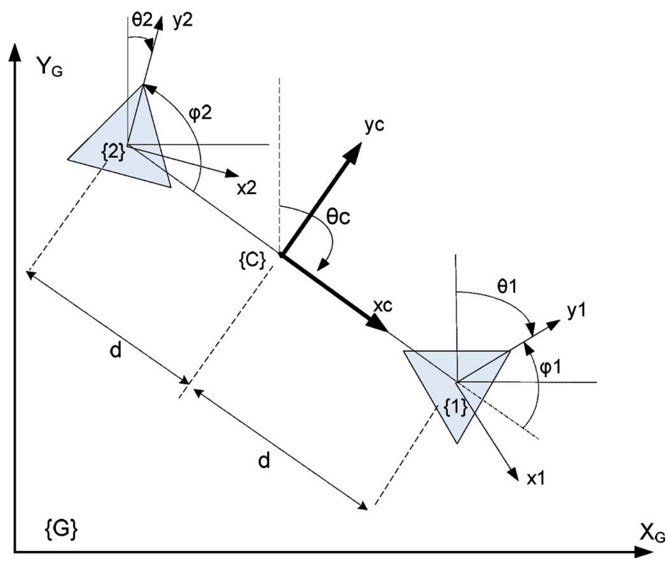

Fig. 3. Pose reference frames for the planar two-robot system.

A. Two-Robot Cluster

Fig. 3 depicts the relevant reference frames for the planar two-robot problem. Because of the sensor data used in experimentation, the global frame conventions were selected as follows: $\vec{y}_G$ points to the north, $\vec{x}_G$ points to the east, and $\vec{\theta}_G$ is the compass-measured heading. For our paper, we have chosen to locate the cluster frame $\{C\}$ at the cluster's centroid, oriented with $\vec{x}_c$ pointing toward robot 1. Based on this, Table II summarizes the variables of interest, the applicable DOFs, and the acting constraints, following the protocol presented in Section II.

B. Three-Robot Cluster

Fig. 4 depicts the frame assignments for the planar three-robot cluster with $\{C\}$ once again positioned at the centroid and oriented toward robot 1. Table II summarizes the variables of interest, the applicable DOF, and the acting constraints, following the protocol from Section II.

VI. Experimental Testbed

Several testbeds have been developed in order to evaluate our work in cluster space control. A MATLAB/Simulink simulation environment is routinely used to support iterative development prior to committing to hardware experiments. To enhance visualization of cluster motion, this simulator includes a simple 3-D world representation of robot motion using the Virtual Reality Markup Language (VRML) Toolbox [30]. The simulator includes models of several holonomic and nonholonomic multirobot hardware systems that are available for experimentation, and it supports evaluation of both automated trajectory controllers as well as interactive, human-in-the-loop controllers through the use of joystick inputs [33], [43].

For the experiments presented in Section VII, the testbed consisted of two commercially available differential drive chasses with a custom suite of sensing, communication, and control equipment, as depicted in Fig. 5. Developed specifically as a low-cost testbed to support rapid integration and test of new control strategies, the design of the system emphasizes these features at the expense of performance in terms of robot speed, sensor accuracy, and communication/servo rates.

Each chassis is an Amigobot, available from Mobile Robots, Inc., and is capable of closed-loop velocity control through the use of wheel encoders and a Renesas SH7144-based microcontroller. The chassis is commanded via a serial RS-232 connection through which desired robot-level translational and rotational velocities are specified. On top of each robot is mounted a student-integrated suite of sensors and microcontrollers. Sensors include a WAAS-enabled Garmin 18 GPS receiver operating at 5 Hz and a Devantech CMPS03 digital compass that updates at 10 Hz. A modular network of six BasicX-24 microcontrollers are used for component interfacing and data handling; these microcontrollers have a simple learning curve appropriate for a student-centered testbed, limited but sufficient computational speed and memory, and multitasking and floating-point math capabilities.

A Maxstream $9X$stream 900 MHz serial radiomodem is used to receive remotely generated robot-level velocity commands and return robot sensor data to an offboard control station consisting of a Pentium IV-class personal computer. The computer runs a GUI that allows a human operator to specify robot or cluster-level directives and to monitor activity. The computer also executes a MATLAB-based control program that ingests sensor data and directives from the GUI, performs the desired control computation, and returns new velocity set points to be sent to each robot. The MATLAB-based controller is particularly valuable for rapid integration and test given that our investigators typically have a strong desire to use MATLAB, while our testbed developers typically consist of students with embedded systems programming knowledge. We note that we currently use only $\sim$20% of the wireless system's bandwidth.

VII. Experimental Demonstration

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 selected regulated and trajectory-controlled test cases for two- and three-robot clusters.

In performing these experiments, performance was limited by the quality of our sensors; the GPS components provided position data with a precision of $\pm$0.95 m (CEP) over the duration of the experiments, and the digital compass was accurate to within $\pm 5^{\circ}$. Performance was also constrained by the use of only proportional control with manual tuning. In addition, control commands were often bounded in order to maintain robot velocities at a speed of less than 1 m/s.

A. Two-Robot Regulated Motion

This experiment was performed without a trajectory generator by simply specifying constant desired values of specific cluster speeds and using the controller to regulate their values. The

TABLE II
SUMMARY OF CLUSTER SPACE STATE VARIABLE SELECTION PROCESS

	Two-Robot System	Three-Robot System
Robot Space State Variables	${}^{G}\vec{R} = (x_1, y_1, \theta_1, x_2, y_2, \theta_2)^T$ ${}^{G}\dot{\vec{R}} = (\dot{x}_1, \dot{y}_1, \dot{\theta}_1, \dot{x}_2, \dot{y}_2, \dot{\theta}_2)^T$	${}^{G}\vec{R} = (x_1, y_1, \theta_1, x_2, y_2, \theta_2, x_3, y_3, \theta_3)^T$ ${}^{G}\dot{\vec{R}} = (\dot{x}_1, \dot{y}_1, \dot{\theta}_1, \dot{x}_2, \dot{y}_2, \dot{\theta}_2, \dot{x}_3, \dot{y}_3, \dot{\theta}_3)^T$
Robot Degree of Freedom Assessment		
Number of robots (n)	$n = 2$	$n = 3$
Number of robot DOFs (m)	$m = 3$	$m = 3$
Number of robot translational DOFs (p)	$p=2$	$p=2$
Number of robot rotational DOFs (r)	$r = 1$	$r = 1$
Number of system DOFs ($n{\cdot}m$)	$n{\cdot}m = 2{\cdot}3 = 6$	$n{\cdot}m = 3{\cdot}3 = 9$
Selection of Cluster Space State Variables		
Number of cluster position variables (p)	$p = 2$	$p = 2$
Number of cluster orientation variables (o)	Given $n \geq p \neq 3$, $o = p\text{-}1 = 1$	Given $n \geq p \neq 3$, $o = p\text{-}1 = 1$
Selection of {C} pose variables	x_c, y_c, θ_c	x_c, y_c, θ_c
Cluster space constraints for {C}	${}^{C}y_{1org} = {}^{C}y_{2org} = 0,\ {}^{C}x_{1org} = -{}^{C}x_{2org}$	${}^{C}x_{1org} = 0,\ {}^{C}x_{2or} + {}^{C}x_{3org} = 0,\ {}^{C}y_{1org} + {}^{C}y_{2org} + {}^{C}y_{3org} = 0$
Number of relative robot orientation variables	$n{\cdot}r = 2{\cdot}1 = 2$	$n{\cdot}r = 3{\cdot}1 = 3$
Selection of relative robot orientation variables	ϕ_1, ϕ_2	ϕ_1, ϕ_2, ϕ_3
Number of cluster shape variables (s)	Given $n \geq p \neq 3$, $s = p(n\text{-}2)+1 = 1$	Given $n \geq p \neq 3$, $s = p(n\text{-}2)+1 = 3$
Selection of cluster shape variables	Size = $d = {}^{C}x_{1org} = -{}^{C}x_{2org}$	Side 1-2 length: p, Side 1-3 length: q, Lead angle: β
Summary of Cluster Space State Variables	$\vec{C} = (x_c, y_c, \theta_c, \phi_1, \phi_2, d)^T$ $\dot{\vec{C}} = (\dot{x}_c, \dot{y}_c, \dot{\theta}_c, \dot{\phi}_1, \dot{\phi}_2, \dot{d})^T$	$\vec{C} = (x_c, y_c, \theta_c, \phi_1, \phi_2, \phi_3, p, q, \beta)^T$ $\dot{\vec{C}} = (\dot{x}_c, \dot{y}_c, \dot{\theta}_c, \dot{\phi}_1, \dot{\phi}_2, \dot{\phi}_3, \dot{p}, \dot{q}, \dot{\beta})^T$

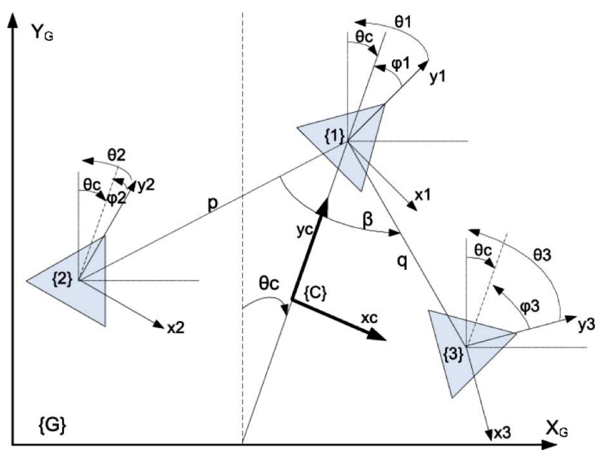

Fig. 4. Pose reference frames for the planar three-robot system.

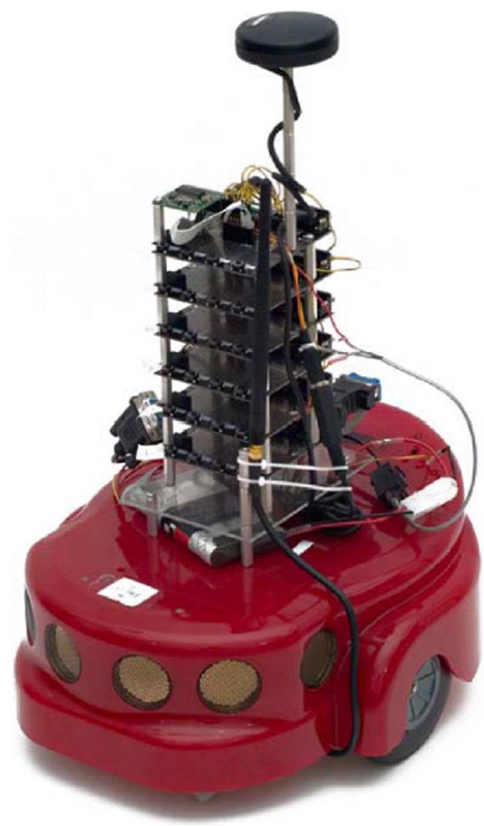

Fig. 5. Experimental testbed. Clusters of these vehicles were controlled using the cluster space technique. Onboard equipment includes GPS, a compass, and an array of microcontrollers. A GUI allows an operator to directly specify and monitor the motion of the two-robot cluster in real time.

control loop servo rate was run at ~3 Hz, the fastest possible rate at that time, in order to approximate continuous time control; sensor acquisition and robot velocity set point commands were executed at the same rate. Fig. 6 shows the results of a demonstration in which the robots are initially positioned with a size of $d = 4$ m and a cluster heading of $\theta_C = 90°$. The controller was given desired values for these parameters of $d_{\rm des} = 10$ m and $\theta_{\rm Cdes} = 0°$; the remaining cluster parameters were uncontrolled. As can be seen, once the transient died out, cluster size was controlled to within 1 m and cluster heading was properly regulated.

B. Three-Robot Trajectory-Based Motion

This experiment used a trajectory generator to specify that the cluster simultaneously translate along a 10-m-diameter circle, rotate at 1.5°/s, and maintain its shape of a right triangle with sides $p = q = 10$ m. Fig. 7 shows the resulting motion [39]; the mean square errors were 0.504 m for X_C, 0.464 m for Y_C, 0.038° for θ_C, 1.589 m for p, 1.224 m for q, and 0.031° for β. These experiments were run at a control servo rate of 10 Hz.

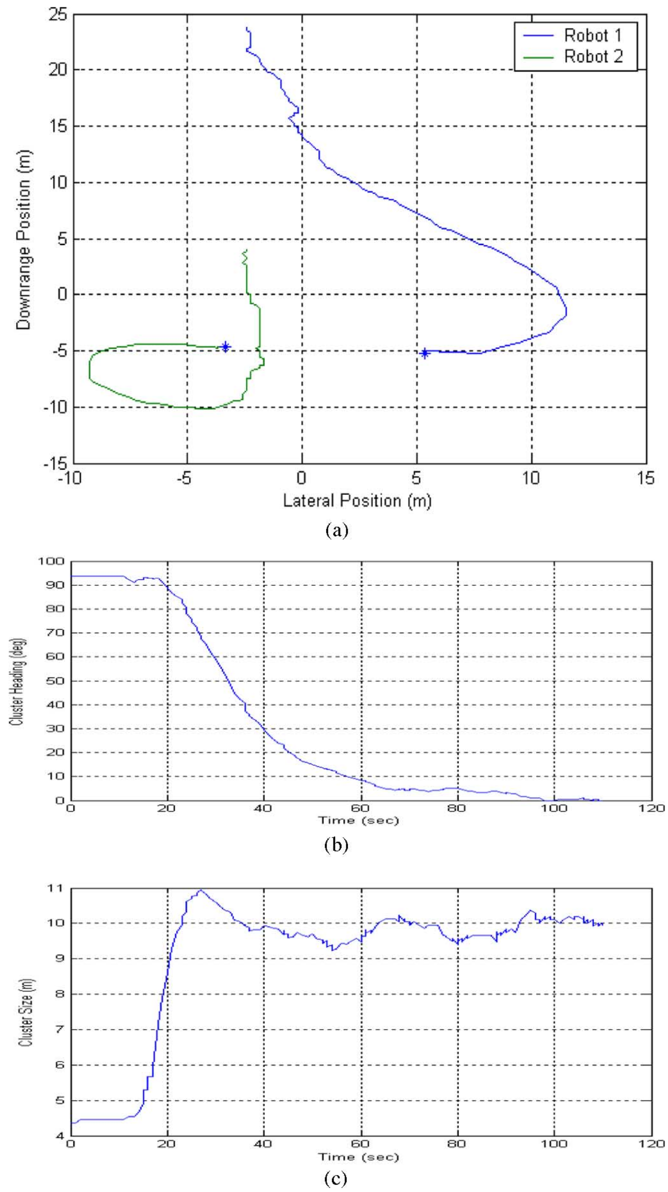

Fig. 6. Regulated motion for the two-robot cluster: experimental results. (a) Overhead view of the actual path. (b) Time history of cluster heading given a desired value of $\theta_C = 0°$ at $t = 10$ s. (c) Time history of cluster size given a desired value of $d = 10$ m at $t = 10$ s.

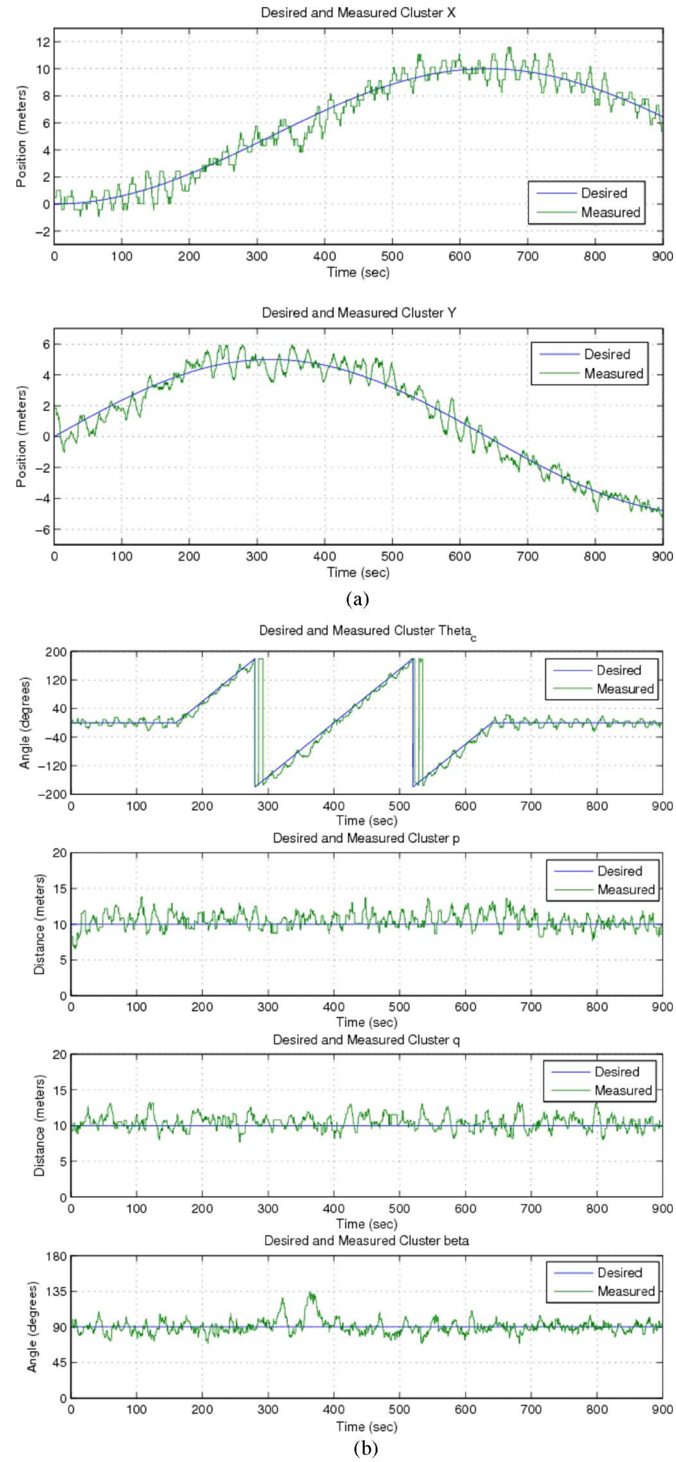

Fig. 7. Trajectory following motion for the three-robot cluster: experimental results.

VIII. Performance Considerations

In considering the use of the cluster space control technique, several issues relating to performance are worthy of discussion.

A. *Shape Variable Selection*

Many interesting implementation issues arise due to the selection and role of the cluster space shape variables. For example, with increasing numbers of robots and DOFs, the choice of appropriate shape variables is often ambiguous even for a specific geometric template (e.g., there are many ways to geometrically specify a shape as simple as a triangle). In addition, the selection of the appropriate geometric template is often a function of the task, application, and operator. Furthermore, given a particular geometric template, singular configurations of that geometry are possible, a condition that can provide challenges to effective control; this specific problem is discussed further in the following subsection.

One strategy for effective shape variable selection is to select variables representing geometric parameters of little interest. The judicious use of default values for such parameters can reduce complexity for the operator when directing the system without preventing the option for complete specification if desired; a good example of this is the skew angle of a triangular three-robot formation. We are also working on techniques to support geometric template switching in order to provide the most effective real-time shape metaphor for the operator as configurations and tasks change during operation. This would allow an operator using a triangle metaphor for a three-robot cluster to seamlessly direct the cluster to "line up" with certain parameters specific to a linear template in order to achieve the next task, such as driving through a narrow passage.

B. Singular Configurations

Configurations may exist for which the Jacobian or its inverse becomes singular. For example, for the three-robot cluster defined in Section V-B, singularities exist for the following cluster space values: $p = 0, q = 0, p = \infty, q = \infty$, and $\beta = \pi$. The first two may be avoided with a collision-avoidance policy between the robots, and the second two are not valid for the practical applications under consideration. The final configuration, however, is reached in a straight-line configuration in which robot 1 is between the other robots. This may be avoided by policy; however, a straight-line configuration is likely to be desired for arbitrary applications.

When using a controller architecture such as that depicted in Fig. 2, the practical effect of operating at or even near such singularities is an undesirable computational amplification. Exceptionally high commanded robot velocities can result in the forward portion of the control loop; these may be addressed by incorporating thresholds on such commands. In the feedback path, errors in sensor measurements taken in the robot space and which must be transformed to the cluster space via the kinematic relationships may be significantly amplified as well, thereby preventing effective control; this effect is analytically and experimentally characterized in [40].

We are currently exploring alternatives to simply avoiding singular configurations. One successfully demonstrated strategy for the three-robot straight-line singularity is to identify equivalent geometries for different cluster space values. For example, a nonsingular three-robot straight-line configuration is achieved for $\beta = 0$, although robot 1 is positioned at the end of the line in this case. This can be exploited by identifying nonsingular configuration options for the operator or through abstraction with different robot assignments used by the operator than those used for real-time cluster control. An additional strategy currently under development is to use nonminimal cluster space variable representations with the objective of avoiding kinematic singularities at the expense of added computational complexity.

C. Computational Complexity

While the motivation of the cluster space approach is to improve specification and monitoring of mobile multirobot motion, the inclusion of the kinematic transformations in the real-time control loop adds computational overhead. The practical effect of this overhead is the need for more powerful processors or the reduction of the control loop servo rate. These demands may make the cluster space approach intractable for suitably large clusters. Although we have not yet quantitatively characterized how computational complexity increases with the number of robots and increasing DOFs, we are able to offer several observations and strategies for accommodating it in a practical manner.

First, we note that the centralized approach and our previously stated focus on applications with local operation and global communication makes the use of an offboard computer for kinematic and control computations practical, which is precisely why we have implemented our own testbed in this configuration. Use of an offboard nonmobile computer allows more powerful processors to be used to address the computational demand.

Second, computational complexity is a function of the placement of $\{C\}$ and the choice of cluster space variables. As a simple illustration for both the two- and three-robot clusters, placing $\{C\}$ coincident with robot 1's body frame dramatically simplifies the kinematic relationships thereby reducing computational overhead. Third, direct sensing of cluster space variables can reduce the number of kinematic computations executed in the feedback loop.

A fourth option is to implement a dual-rate controller that updates the inverse Jacobian and, if used, the Jacobian matrices, given that they are a function of the cluster's pose, at a rate slower than the primary control servo loop. The Jacobian update rate would be determined based on the rates of change of the cluster's pose, trading accuracy for computational complexity. Dual-rate controllers such as this are used for robotic manipulators [44]. A fifth strategy, one we have yet to fully explore, is to reduce the effective size of the cluster state space in cases where certain cluster space variables are held constant; this may occur for certain applications and may also result when operators use default values for obscure shape parameters in large clusters, as discussed in Section VIII-A. Of course, a sixth option is to execute the real-time control loop in conventional robot space and to use the kinematic transforms simply to support direct interaction with the operator.

While these techniques allow the cluster space technique to be incrementally applied to clusters of larger size, the technique does not scale to arbitrary cluster sizes such as those targeted by many of the techniques discussed in Section IX. We note, however, that such a limitation does not prevent the cluster space control approach from being a powerful and unique strategy for applications making use of a limited number of robots operating in a local area with global communication.

IX. Related Work

Noteworthy comparisons and contrasts can be made by relating the work presented in this paper to control system approaches developed for both the formation control of mobile robots as well as the Cartesian control of robotic manipulators.

A. Formation Control of Multiple Robots

Control of nonholonomic vehicles is a well-studied topic in the field of robotics. Work on stable and adaptive tracking for such vehicles has addressed both kinematic and dynamic vehicle controllers and has developed the conditions for stability [45], [46]. Within a single vehicle, compelling work has also been done on the coordinated control of independently steered wheels that have strict kinematically derived pose requirements in order to effectively implement vehicle-level maneuvers [47].

For multirobot systems, significant previous work in cooperative and formation keeping control has been developed, drawing on work in control theory, robotics, and biology [48] and applicable for robotic applications throughout land, sea, air, and space. Notable work in this area includes the use of leader–follower techniques, in which follower robots control their position relative to a designated leader [49], [50]. 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 is led by a designated leader. For example, a follower robot might position itself with respect to the local leader by sensing and controlling its relative distance and bearing to that leader, or by maintaining its relative distance between two local leaders [51].

Several approaches employ artificial fields as a construct to establish formation keeping forces for individual robots within a formation. For example, potential fields may be used to implement repulsive forces among neighboring robots and between robots and objects in the field in order to symmetrically surround an object to be transported [52]. Potential fields and behavioral motion primitives have also been used to compute reactive robot drive commands that balance the need to arrive at the final destination, to maintain relative locations within the formation, and to avoid obstacles [23], [53], [54]. As another example, the virtual bodies and artificial potentials (VBAPs) approach uses potential fields to maintain the relative distances both between neighboring robots as well as between robots and reference points, or “virtual leaders,” that define the “virtual body” of the formation [55], [56]; this approach has been successfully demonstrated in field tests of underwater robots performing a distributed sampling mission [12].

Several alternate approaches have used decentralized control theory [18], [57] as the conceptual basis for the modeling and control of robot clusters; a benefit of this approach is its ability to allow detailed analysis of system stability, controllability, and observability. One example of an implemented system using this approach is the RATLER network of robotic sentry vehicles, which has been successfully demonstrated in applications ranging from perimeter security to minefield maintenance [11].

Compared to these methods, the most distinctive feature of the cluster space approach is the abstracted level at which cluster pose and motions are both specified and controlled. The control system designer can select a control policy specific to the motion quantities of interest rather than seeking an applicable configuration of virtual body reference points or tuning the strengths of competing potential or behavioral fields. In addition, because cluster motion is specified as an aggregate body, there is no need for explicit leaders, real or virtual. Furthermore, in contrast to many “virtual rigid body” approaches, the cluster space represents all pose DOFs, allowing the cluster to be fully articulated if this is demanded by the application.

In contrast, the cluster space approach is centralized, requiring global state information in order to compute compensation commands. The alternative approaches are decentralized, relying on local information. The result is the potential for better performance albeit at the cost of significantly higher demands on computation and communication resources. Enhanced performance for some applications, such as those requiring precise control of cluster variables such as interrobot spacing, is also possible given that these states are directly controlled in our framework.

As discussed in Section VIII-C, these demands make the cluster space approach intractable for suitably large clusters and impractical for clusters operating with limited or periodically unavailable communication channels. Accordingly, the current focus for our cluster space work is for relatively small groups of mobile robots that operate within a limited workspace and with ample communication bandwidth such that offboard computational support is available as necessary; in addition, applications of interest are those in which reliable high bandwidth communications are required for the task (e.g., real-time distributed sensor fusion) or in which critical cluster space quantities can be implicitly sensed (e.g., multirobot object transport [33]).

B. *Cartesian Control of Manipulators*

A significant comparison can be made between the cluster space approach and the Cartesian (or operational) space control approach that has been developed for serial chain manipulators [44], [58]. In both, kinematic transforms allow motion commands to be specified in an alternate space that can improve the quality of operator interaction and motion characteristics. Just as it would be painstaking for an operator to directly specify the joint trajectories required to move the endpoint of a 6-DOF articulated manipulator in a straight line, it would be overwhelming to have a mobile cluster pilot independently drive several robots to implement a cluster-level directive with any level of complexity. A second comparison is that in both approaches, the kinematics relate a vector of variables referenced in the global frame to a vector of variables that are referenced in separate frames (e.g., separate link frames for a manipulator, and a combination of global and cluster frames for our approach).

In contrast, one significant difference lies in the scope of the Cartesian space and cluster space controllers. To date, the cluster space approach has only been used to update velocity set points that are implemented by each robot’s closed-loop velocity control system. In contrast, Cartesian controllers have been proven useful not only in computing velocity set points for joints but also for full dynamic control in which joint forces and torques are computed and directly applied to the manipulator in order to achieve the desired result.

X. Ongoing and Future Work

To date, we have successfully used the cluster space control approach to demonstrate cluster-space-based versions of regulated motion control [30], automated trajectory control [31], [32], human-in-the-loop piloting [33], [34], partitioned-model-based control [35], and potential-field-based obstacle avoidance [36]–[38]. This work has included experiments with two-, three-, and four-robot planar land rover clusters [39]–[41], with two-boat surface vessel systems [42], and for robots that are both holonomic and nonholonomic. Current initiatives include implementing cluster space control on small clusters of aerostatically stable 4-DOF blimps, using redundant shape variables to overcome the computational challenges associated with singularities, and integrating a spoken dialog interface to allow verbal command of the system.

In the future, we plan to explore the scalability of this approach to systems with more robots and additional DOFs; this will include experimental demonstrations that exploit the array of our robotic devices that operate in land, sea, air, and space [59]. To address tractability issues, we plan to 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 multirobot system. Finally, to improve our ability to experimentally verify and validate our work, we are continuously improving the capabilities and performance of our hardware-in-the-loop testbed by adding diverse pose sensing suites, additional wireless communication links, and more powerful onboard computers.

XI. Summary and Conclusion

We have introduced the cluster space state representation as a means of specifying the control of mobile multirobot systems. Our conceptual framework includes a methodology for selecting cluster space state variables, the use of formal kinematics relating these cluster space state variables to robot-specific variables, and an architecture for incorporating these kinematics into a cluster space control system.

To demonstrate this technique, we have applied it to the control of two- and three-robot clusters consisting of wheeled differential drive rovers operating in a plane. This included a particular choice of clusters space variables for each as well as the presentation of the resulting kinematic transformations. It also included the use of these transforms in the real-time control of real robots. These experiments utilized a low-cost testbed consisting of commercially available robot chasses with custom sensing suites and an offboard computer that wirelessly communicated with the robots and computed the real-time control law. Experimental results show that simple cluster space motions can be specified and monitored by a single operator or pilot, even when the equivalent robot-specific motions are quite complex. We believe that this will lead to enhanced performance, cost-effective improvements in the operator/robot ratio for controlling multirobot systems, and a reduced learning curve for new operators/pilots.

Our ongoing work includes applying cluster space control to systems of increasing DOFs and in different domains, addressing implementation issues such as control near singular configurations and computational complexity, and integrating this motion control strategy with application layer controllers that support collaborative multirobot behavior.

Acknowledgment

The authors thank student researchers who assisted with performing experiments and who have contributed to the development of the cluster space approach, to include authors listed in [30]–[42]. This work has benefited greatly through the feedback of numerous colleagues; the authors are particularly indebted to Dr. J. Melton and the Robotic Systems Laboratory (RSL) graduate research team for their invaluable comments and critical review regarding this paper. 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, NASA Ames Research Center, or Santa Clara University.

Appendix

Given the selection of cluster space state variables presented in Section IV-A, the forward position kinematic relationships for the two-robot system are [30]

$$\begin{aligned}
x_c &= \frac{x_1 + x_2}{2} \qquad y_c = \frac{y_1 + y_2}{2} \\
d &= \frac{1}{2}\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \\
\theta_c &= \mathrm{ATAN2}\,(x_1 - x_2, y_1 - y_2) \\
\phi_1 &= \theta_1 - \mathrm{ATAN2}\,(x_1 - x_2, y_1 - y_2) \\
\phi_2 &= \theta_2 - \mathrm{ATAN2}\,(x_1 - x_2, y_1 - y_2)
\end{aligned}$$

where ATAN2 is a two-argument computational function that calculates a four-quadrant arc tangent with a range of $[\pi, -\pi]$ [60]. The inverse position kinematic relationships are

$$\begin{aligned}
x_1 &= x_c + d\sin(\theta_c) \qquad & y_1 &= y_c + d\cos(\theta_c) \\
x_2 &= x_c - d\sin(\theta_c) \qquad & y_2 &= y_c - d\cos(\theta_c) \\
\theta_1 &= \theta_c + \phi_1 \qquad & \theta_2 &= \theta_c + \phi_2.
\end{aligned}$$

By differentiating the position kinematic expressions, the velocity kinematics may be expressed as

$$\dot{\vec{C}} = \begin{bmatrix}
\frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\
\frac{x_1 - x_2}{A} & \frac{-x_1 + x_2}{A} & \frac{y_1 - y_2}{A} & \frac{-y_1 + y_2}{A} & 0 & 0 \\
\frac{y_1 - y_2}{B} & \frac{-y_1 + y_2}{B} & \frac{-x_1 + x_2}{B} & \frac{x_1 - x_2}{B} & 0 & 0 \\
\frac{-y_1 + y_2}{B} & \frac{y_1 - y_2}{B} & \frac{x_1 - x_2}{B} & \frac{-x_1 + x_2}{B} & 1 & 0 \\
\frac{-y_1 + y_2}{B} & \frac{y_1 - y_2}{B} & \frac{x_1 - x_2}{B} & \frac{-x_1 + x_2}{B} & 0 & 1
\end{bmatrix} \dot{\vec{R}}$$

where $\vec{C} = [\,x_c\ y_c\ d\ \theta_c\ \phi_1\ \phi_2\,]^T$, $\vec{R} = [\,x_1\ x_2\ y_1\ y_2\ \theta_1\ \theta_2\,]^T$, $A = 2\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$, $B = (x_1 - x_2)^2 + (y_1 -$

$y_2)^2$, and

$$\dot{\vec{R}} = \begin{bmatrix} 1 & 0 & \sin(\theta_c) & d\cos(\theta_c) & 0 & 0 \\ 1 & 0 & -\sin(\theta_c) & -d\cos(\theta_c) & 0 & 0 \\ 0 & 1 & \cos(\theta_c) & -d\sin(\theta_c) & 0 & 0 \\ 0 & 1 & -\cos(\theta_c) & d\sin(\theta_c) & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \dot{\vec{C}}.$$

Given the selection of cluster space state variables presented in Section IV-B, a simplified approximation of the forward position kinematic relationships for the three-robot system are [39]

$$\begin{aligned} x_c &= \frac{x_1 + x_2 + x_3}{3} \\ y_c &= \frac{y_1 + y_2 + y_3}{3} \\ \theta_c &= \text{ATAN2}(\{2/3(x_1) - 1/3(x_2 + x_3)\}, \\ &\quad \{2/3(y_1) - 1/3(y_2 + y_3)\}) \\ \phi_1 &= \theta_1 + \theta_c \\ \phi_2 &= \theta_2 + \theta_c \\ \phi_3 &= \theta_3 + \theta_c \\ p &= \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \\ q &= \sqrt{(x_3 - x_1)^2 + (y_1 - y_3)^2} \\ \beta &= \cos^{-1}\left(\frac{p^2 + q^2 - (x_3 - x_2)^2 - (y_3 - y_2)^2}{2pq}\right). \end{aligned}$$

The resulting inverse position kinematic relationships for the two-robot system are

$$\begin{aligned} x_1 &= x_c + (1/3)r\sin\theta_c \\ y_1 &= y_c + (1/3)r\cos\theta_c \\ \theta_1 &= \phi_1 - \theta_c \\ x_2 &= x_c + (1/3)r\sin\theta_c - p\sin(\beta/2 + \theta_c) \\ y_2 &= y_c + (1/3)r\cos\theta_c - p\cos(\beta/2 + \theta_c) \\ \theta_2 &= \phi_2 - \theta_c \\ x_3 &= x_c + (1/3)r\sin\theta_c + q\sin(\beta/2 - \theta_c) \\ y_3 &= y_c + (1/3)r\cos\theta_c - q\sin(\beta/2 - \theta_c) \\ \theta_3 &= \phi_3 - \theta_c \end{aligned}$$

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

Due to limited space, the Jacobian and inverse Jacobian for the three-robot system are not included here. The three-robot kinematics provide adequate performance when $p \approx q$. An exact kinematic form is presented in [61].

References

[1] D. Folta, L. Newman, and D. Quinn, "Design and implementation of satellite formations and constellations," *Adv. Astronaut. Sci.*, vol. 100, no. 1, pp. 57–70, 1998.

[2] R. DeFazzio, S. Owens, and S. Good, "Follow that satellite: EO-1 maneuvers into close formation with landsat-7," *Adv. Astronaut. Sci.*, vol. 109, no. 3, pp. 2095–2113, 2002.

[3] J. Goutoule and F. de Boer, "Large interferometer antennas synthesized by satellites in formation for earth remote sensing," in *Proc. 2000 Int. Geosci. Remote Sens. Symp. (IGARSS)*, vol. 2, pp. 869–870.

[4] D. Miller, A. Saenz-Otero, J. Wertz, A. Chen, G. Berkowski, C. Brodel, S. Carlson, D. Carpenter, S. Chen, S. Cheng, D. Feller, S. Jackson, B. Pitts, F. Perez, and J. Szuminski, "Spheres: A testbed for long duration satellite formation flying in micro-gravity conditions," *Adv. Astronaut. Sci.*, vol. 105, no. 1, pp. 167–179, 2000.

[5] B. Palmintier, C. Kitts, P. Stang, and M. Swartwout, "A distributed control architecture for small satellite and multi-spacecraft missions," presented at the 2002 AIAA/USU Conf. Small Satellites, Logan, UT, Paper SSC02-IV-6.

[6] T. Speer, E. Mills, and J. Tate, "Formation flight technology," *Aircr. Eng.*, vol. 43, no. 7, pp. 4–8, Jul. 1971.

[7] I. Kaminer, O. Yakimenko, V. Dobrokhodov, M. Lizarraga, and A. Pascoal, "Cooperative control of small UAVs for naval applications," in *Proc. 2004 IEEE Conf. Decis. Control*, vol. 1, pp. 626–631.

[8] R. Becker, C. Borowski, O. Petrovic, C. Kitts, and N. Quinn, "SCU aerial robotics team—Experimentation with an autonomous UAV observation platform," in *Proc. 2004 AUVSI Unmanned Syst. North Amer. Symp.*, pp. 1661–1673.

[9] C. Ortiz, K. Konolige, R. Vincent, B. Morisset, A. Agno, M. Eriksen, D. Fox, B. Limketkai, J. Ko, B. Steward, and D. Schulz, "Centibots: Very large scale distributed robotic teams," in *Proc. 2004 16th Innovative Appl. Artif. Intell. Conf. (IAAI)*, pp. 1022–1023.

[10] R. Grabowski, L. Navarro, C. Paredis, and P. Khosla, "Heterogeneous teams of modular robots for mapping and exploration," *Auton. Robots—Spec. Issue Heterogeneous Multirobot Syst.*, vol. 8, no. 3, pp. 293–308, Jun. 2000.

[11] J. Feddema, C. Lewis, and P. Klarer, "Control of multiple robotic sentry vehicles," *Proc. SPIE*, vol. 3693, pp. 212–223, 1999.

[12] E. Fiorelli, N. E. Leonard, P. Bhatta, D. Paley, R. Bachmayer, and D. M. Fratantoni, "Multi-AUV control and adaptive sampling in Monterey Bay," *IEEE J. Ocean. Eng.*, vol. 31, no. 4, pp. 935–948, Oct. 2006.

[13] Y. Tan and B. Bishop, "Evaluation of robot swarm control methods for underwater mine countermeasures," in *Proc. 2004 Annu. Southeastern Symp. Syst. Theory*, vol. 36, pp. 294–298.

[14] E. Yang and D. Gu, "Nonlinear formation-keeping and mooring control of multiple autonomous underwater vehicles," *IEEE/ASME Trans. Mechatronics*, vol. 12, no. 2, pp. 164–178, Apr. 2007.

[15] H. Geismar, C. Sriskandarajah, and N. Ramanan, "Increasing throughput for robotic cells with parallel machines and multiple robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 1, no. 1, pp. 84–89, Jul. 2004.

[16] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, and A. Casal, "Coordination and decentralized cooperation of multiple mobile manipulators," *J. Robot. Syst.*, vol. 13, no. 11, pp. 755–764, Nov. 1996.

[17] D. Sun and J.K. Mills, "Manipulating rigid payloads with multiple robots using compliant grippers," *IEEE/ASME Trans. Mechatronics*, vol. 7, no. 1, pp. 23–34, Mar. 2002.

[18] D. Siljak, *Decentralized Control of Complex Systems*. New York: Academic, 1991.

[19] M. Ikeda, "Decentralized control of large-scale systems," in *Three Decades of Mathematical Systems Theory*, H. Nijmeijer and J. Schumacher, Eds. Berlin, Germany: Springer-Verlag, 1989, pp. 219–242.

[20] T. Yang, H. Yu, M. Fei, and L. Li, "Networked control systems: A historical review and current research topics," *Meas. Control*, vol. 38, no. 1, pp. 12–16, Feb. 2005.

[21] R. Carpenter, "Decentralized control of satellite formations," *Int. J. Robust Nonlinear Control*, vol. 12, no. 2/3, pp. 141–161, Feb./Mar. 2002.

[22] S. Stankovic, M. Stanojevic, and D. Siljak, "Decentralized overlapping control of a platoon of vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 8, no. 5, pp. 816–831, Sep. 2000.

[23] T. Balch and R. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Trans. Robot. Autom.*, vol. 14, no. 6, pp. 926–939, Dec. 1998.

[24] E. Flinn, "Testing for the 'boids'," *Aerosp. Amer.*, vol. 43, no. 6, pp. 28–29, Jun. 2005.

[25] H. Yamaguchi and T. Arai, "Distributed and autonomous control method for generating shape of multiple mobile robot group," in *Proc. 1994 EEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 800–807.

[26] K. Tan and M. Lewis, "Virtual structures for high-precision cooperative mobile robotic control," in *Proc. 1996 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 132–139.
[27] M. Hashimoto, F. Oba, and T. Eguchi, "Dynamic control approach for motion coordination of multiple wheeled mobile robots transporting a single object," in *Proc. 1995 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 1944–1951.
[28] D. Rus, B. Donald, and J. Jennings, "Moving furniture with teams of autonomous robots," in *Proc. 1995 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 235–248.
[29] C. P. Tang, R. M. Bhatt, M. Abou-Samah, and V. Krovi, "Screw-theoretic analysis framework for cooperative payload transport by mobile manipulator collectives," *IEEE/ASME Trans. Mechatronics*, vol. 11, no. 2, pp. 169–178, Apr. 2006.
[30] R. Ishizu, "The design, simulation and implementation of multi-robot collaborative control from the cluster perspective," M.S. thesis, Dept. Electr. Eng., Santa Clara Univ., Santa Clara, CA, Dec. 2005.
[31] P. Connolley, "Design and implementation of a cluster space trajectory controller for multiple holonomic robots," M.S. thesis, Dept. Mech. Eng., Santa Clara Univ., Santa Clara, CA, Jun. 2006.
[32] T. To, "Automated cluster space trajectory control of two non-holonomic robots," M.S. thesis, Dept. Comput. Eng., Santa Clara Univ., Santa Clara, CA, Jun. 2006.
[33] M. Kalkbrenner, "Design and implementation of a cluster space human interface controller, including applications to multi-robot piloting and multi-robot object manipulation," M.S. thesis, Dept. Mech. Eng., Santa Clara Univ., Santa Clara, CA, Jun. 2006.
[34] B. Tully, "Cluster space piloting of a nonholonomic multi-robot system," M.S. thesis, Dept. Comput. Eng., Santa Clara Univ., Santa Clara, CA, Jun. 2006.
[35] R. Lee, "Model-based cluster-space control of multiple robot systems," M.S. thesis, Dept. Mech. Eng., Santa Clara Univ., Santa Clara, CA, Jun. 2007.
[36] P. Chindaphorn, "Cluster space obstacle avoidance for two non-holonomic robots," M.S. thesis, Dept. Comput. Eng., Santa Clara University, Santa Clara, CA, Jun. 2006.
[37] K. Stanhouse, "Cluster space obstacle avoidance for mobile multi-robot systems," M.S. thesis, Dept. Electr. Eng., Santa Clara Univ., Santa Clara, CA, Dec. 2006.
[38] C. Kitts, K. Stanhouse, and P. Chindaphorn, "Cluster space collision avoidance for two-robot systems," submitted to *IEEE/RSJ Int. Conf. Intell. Robots and Systems*, St. Louis MO, Oct. 2009.
[39] I. Mas, O. Petrovic, and C. Kitts, "Cluster space specification and control of a 3-robot mobile system," in *Proc. 2008 IEEE Int. Conf. Robot. Autom.*, Pasadena, CA, pp. 3763–3768.
[40] I. Mas, J. Acain, O. Petrovic, and C. Kitts, "Error characterization in the vicinity of singularities in multi-robot cluster space control," accepted for publication in *Proc. 2008 IEEE Int. Conf. Robot. Biomimetics*, Bangkok, Thailand.
[41] E. Girod, "Design and implementation of four robot cluster space control," M.S. thesis, Dept. Mech. Eng., Santa Clara Univ., Santa Clara, CA, Jun. 2008.
[42] P. Mahacek, I. Mas, O. Petrovic, J. Acain, and C. Kitts, "Cluster space control of a 2-robot autonomous surface vessels system," *Marine Technol. Soc. J.*, accepted for publication.
[43] M. Neumann and C. Kitts, "Control of two holonomic robots using a camera-referenced specification of motion," presented at the AIAA 3rd "Unmanned-Unlimited" Tech. Conf., Workshop, Exhib., Chicago, IL, Sep. 20–23, 2004.
[44] J. Craig, *Introduction to Robotics: Mechanics and Control*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 2004.
[45] Y. Kanayama, "A stable tracking control method for a non-holonomic mobile robot," in *Proc. 1991 IEEE/RSJ Int. Workshop Intell. Robots Syst.*, pp. 1236–1241.
[46] T. Fukao, H. Nakagawa, and N. Adachi, "Adaptive tracking control of a nonholonomic mobile robot," *IEEE Trans. Robot. Autom.*, vol. 16, no. 5, pp. 609–615, Oct. 2000.
[47] Y. Li, L. Yang, and G. Yang, "Network-based coordinated motion control of large-scale transportation vehicles," *IEEE/ASME Trans. Mechatronics*, vol. 12, no. 2, pp. 208–215, Apr. 2007.
[48] V. Kumar, N. E. Leonard, and A. S. Morse, Eds., *Cooperative Control: A Post-Workshop Volume, 2003 Block Island Workshop on Cooperative Control*. New York: Springer-Verlag, 2005.
[49] Z. Wang and D. Gu, "A local sensor based leader-follower flocking system," in *Proc. 2008 IEEE Int. Conf. Robot. Autom.*, Pasadena, CA, pp. 3790–3795.
[50] Y. Hur, R. Fierro, and I. Lee, "Modeling distributed autonomous robots using CHARON: Formation control case study," in *Proc. 2003 IEEE Int. Symp. Object-Oriented Real-Time Distrib. Comput.*, Hokkaido, Japan, pp. 93–96.
[51] A. Das, R. Fierro, V. Kumar, J. Ostrowski, J. Spletzer, and C. Taylor, "A vision-based formation control framework," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 813–825, Oct. 2002.
[52] P. Song and V. Kumar, "A potential field based approach to multi-robot manipulation," in *Proc. 2002 IEEE Int. Conf. Robot. Autom.*, vol. 2, pp. 1217–1222.
[53] R. Dougherty, V. Ochoa, Z. Randles, and C. Kitts, "A behavioral control approach to formation-keeping through an obstacle field," in *Proc. 2004 IEEE Aerosp. Conf.*, vol. 1, pp. 168–175.
[54] F. E. Schneider and D. Wildermuth, "A potential field based approach to multi robot formation navigation," in *Proc. 2003 IEEE Int. Conf. Robot., Intell. Syst. Signal Process.*, pp. 680–686.
[55] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proc. 2001 IEEE Conf. Decis. Control*, pp. 2968–2973.
[56] P. Ögren, E. Fiorelli, and N. E. Leonard, "Formations with a mission: Stable coordination of vehicle group maneuvers," in *Proc 2002 Symp. Math. Theory Netw. Syst.*, pp. 1–15.
[57] M. E. Sezer and D. D. Siljak, "Robust stability of discrete systems," *Int. J. Control*, vol. 48, no. 5, pp. 2055–2063, 1988.
[58] O. Khatib, "The operational space formulation in the analysis, design, and control of manipulators," in *Proc. 1986 Int. Symp. Robot. Res.*, pp. 261–270.
[59] C. Kitts, "Surf, turf, and above the Earth," *IEEE Robot. Autom. Mag.*, vol. 10, no. 3, pp. 30–36, Sep. 2003.
[60] *MATLAB Function Reference*, The Mathworks, Inc., Natick, MA, 2008.
[61] I. Mas, S. Li, J. Acain, and C. Kitts, "Entrapment/escorting and patrolling missions in multi-robot cluster space control," submitted to *IEEE/RSJ Int. Conf. Intell. Robots and Systems*, St. Louis MO, Oct. 2009.

Christopher A. Kitts (S'98–A'00–M'03–SM'05) received the B.S.E. degree from Princeton University, Princeton, NJ, the M.P.A. degree from the University of Colorado, Boulder, and the M.S. and Ph.D. degrees from Stanford University, Stanford, CA.

He was a Research Engineer and an Operational Satellite Constellation Mission Controller. He was an Officer in the U.S. Air Force Space Command, a National Aeronautics and Space Administration (NASA) Contractor with Caelum Research Corporation, a Department of Defense (DoD) Research Fellow at the U.S. Phillips Laboratory, and the Graduate Student Director of the Space Systems Development Laboratory, Stanford University. He is currently an Associate Professor at Santa Clara University, Santa Clara, CA, where he is also the Director of the Robotic Systems Laboratory and is engaged in an aggressive field robotics program specializing in the design, control, and teleoperation of highly capable field robotic systems for scientific discovery, technology validation, and engineering education.

Ignacio Mas (S'06) received the Engineering degree in electrical engineering from the Universidad de Buenos Aires, Buenos Aires, Argentina. He is currently working toward the Ph.D. degree in multirobot navigation and formation control in the Mechanical Engineering Department, Santa Clara University, Santa Clara, CA.

He was a Satellite Systems Engineer. He is currently a Research Assistant in the Robotic Systems Laboratory, Santa Clara University, where he is a primary member of the satellite mission operations team.