We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

BOOK CITATION INDEX
CLARIVATE ANALYTICS
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Model-Based Nonlinear Cluster Space Control of Mobile Robot Formations

Ignacio Mas, Christopher Kitts and Robert Lee
Santa Clara University
USA

1. Introduction

Multi-robot systems have the potential to improve application-specific performance by offering redundancy, increased coverage and throughput, flexible reconfiguration, and/or spatially diverse functionality (Kitts & Egerstedt, 2008). For mobile systems, a driving consideration is the method by which the motions of the individual vehicles are coordinated. Centralized approaches have been successfully demonstrated (Yamaguchi & Arai, 1994; Tan & Lewis, 1996) and have been found to be useful for material transport, regional synoptic sampling, and sensing techniques where active stimulus and/or signal reception are spatially distributed (Hashimoto et al., 1993; Rus et al., 1995; Tang et al., 2006). Such approaches, however, typically suffer from limited scalability and the need for global information. As an alternative, decentralized approaches have been shown to hold great promise in addressing scalability and limited information exchange (Siljak, 1991; Ikeda, 1989; Yang et al., 2005); such approaches often employ control strategies that are behavioral (Balch & Hybinette, 2000; Flinn, 2005; Khatib, 1985), biologically-inspired (Murray, 2007), optimization-based (Dunbar & Murray, 2006), or potential field-based (Leonard & Fiorelli, 2001; Ogren et al., 2004; Justh & Krishnaprasad, 2004; Stipanovic et al., 2004). In this chapter, we present our work relating to the cluster space control technique for multi-robot systems, specifically its implementation using a nonlinear, model-based controller in both kinematic and dynamic forms. The cluster space state representation provides a simple means of specifying and monitoring the geometry and motion characteristics of a cluster of mobile robots without sacrificing flexibility in specifying formation constraints or limiting the ability to fully articulate the formation (Kitts & Mas, 2009). The cluster space control 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 are related to robot-specific state variables 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. With the formal kinematics defined, the controller is composed such that desired motions are specified and control compensations are computed in the cluster space. For a kinematic controller, suitable for robots with negligible dynamics such as many low-speed wheeled robots, compensation commands are transformed to robot space through the inverse Jacobian relationship. For a dynamic controller, appropriate for clusters of marine and aerial

robots, compensation commands are transformed to robot space through a Jacobian transpose relationship. In either case, the resulting robot-level commands are transformed to actuator commands through a vehicle-level inverse Jacobian. If robot space variables are sensed, they are transformed to the cluster space through the use of forward kinematic relationships in order to support control computations. Desired cluster space motions may be provided as regulation inputs, by a trajectory generator, by a realtime pilot, or by a higher-level application-specific controller. The Jacobian and inverse Jacobian matrices are functions of the cluster's pose and therefore must be updated at an appropriate rate. We have successfully used this control approach to demonstrate cluster-space-based versions of regulated motion (Ishizu, 2005), automated trajectory control (Connolley, 2006; To, 2006), human-in-the-loop piloting (Kalkbrenner, 2006; Tully, 2006), and both centralized and decentralized formation control (Mas & Kitts, 2010a). This work has included experiments with 2-, 3- and 4-robot planar land rover clusters (Mas et al., 2008; Mas, Acain, Petrovic & Kitts, 2009; Girod, 2008), with 2- and 3- surface vessel systems (Mahacek et al., 2009) and aerial blimps (Agnew, 2009), for robots that are both holonomic and non-holonomic, for robots negotiating obstacle fields (Kitts et al., 2009), and for target applications such as escorting and patrolling (Mahacek et al., 2009; Mas, Li, Acain & Kitts, 2009). In the following sections, we review the cluster space control strategy to include its formulation, the development of the appropriate kinematic relationships, and the composition of its control architecture. We also present the development of a kinematic and a dynamic nonlinear, model-based partitioned controller. For each case, we present experimental results that verify these techniques and demonstrate the capabilities of the cluster space control approach.

2. Cluster space framework

The cluster space approach to controlling formations of multiple robots was first introduced in (Kitts & Mas, 2009). 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. Consider the general case of a system of n mobile robots where each robot has m DOF, with $m \leq 6$, and an attached body frame, as depicted in Fig. 1. Typical robot-oriented representations of pose use mn variables to represent the position and orientation of each of the robot body frames, $\{1\}, \{2\}, \cdots, \{n\}$, with respect to a global frame $\{G\}$. In contrast, consideration of the cluster space representation starts with the definition of a cluster frame $\{C\}$, and its pose. The pose of each robot is then expressed relative to the cluster frame. 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. An additional set of variables defining the shape of the formation complete the representation.

2.1 Selection of cluster space variables

We select as our state variables a set of position variables (and their derivatives) that capture the cluster's pose and geometry. For the general case of m-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
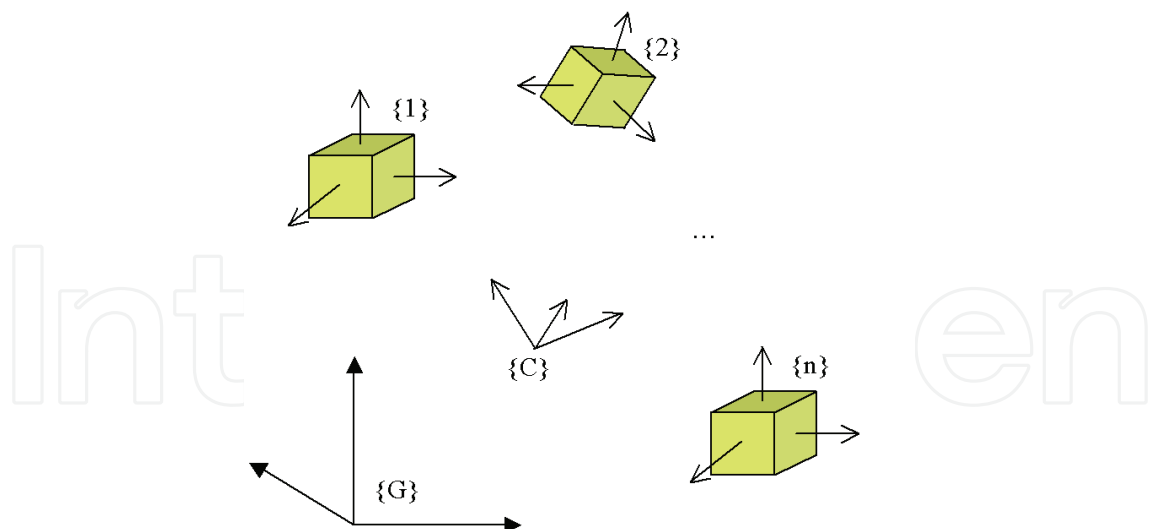
Fig. 1. Cluster and Robot Frame Descriptions with Respect to a Global Frame

for robot i with respect to $\{C\}$ are $(x_i, y_i, z_i, \alpha_i, \beta_i, \gamma_i)$ for $i = 1, 2, \cdots, n$:

$$
\begin{aligned}
c_1 &= f_1(x_c, y_c, z_c, \alpha_c, \beta_c, \gamma_c, x_1, y_1, z_1, \alpha_1, \beta_1, \gamma_1, \cdots, x_n, y_n, z_n, \alpha_n, \beta_n, \gamma_n) \\
c_2 &= f_2(x_c, y_c, z_c, \alpha_c, \beta_c, \gamma_c, x_1, y_1, z_1, \alpha_1, \beta_1, \gamma_1, \cdots, x_n, y_n, z_n, \alpha_n, \beta_n, \gamma_n) \\
&\vdots \\
c_{mn} &= f_{mn}(x_c, y_c, z_c, \alpha_c, \beta_c, \gamma_c, x_1, y_1, z_1, \alpha_1, \beta_1, \gamma_1, \cdots, x_n, y_n, z_n, \alpha_n, \beta_n, \gamma_n).
\end{aligned}
\tag{1}
$$

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 kinematic mechanism 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. A general methodology for selecting the number of variables corresponding to each category given the number of robots and their DOF is described in (Kitts & Mas, 2009). Furthermore, an appropriate selection of cluster variables allows for centralized or distributed control architectures (Mas & Kitts, 2010a). As an example of cluster variables selection, 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. This method can be thought as analogous to the cartesian or operational space control used for serial manipulator chains, where motions can be specified and monitor with respect to the

end effector position, and kinematic transforms relate variables in operational space and joint space.

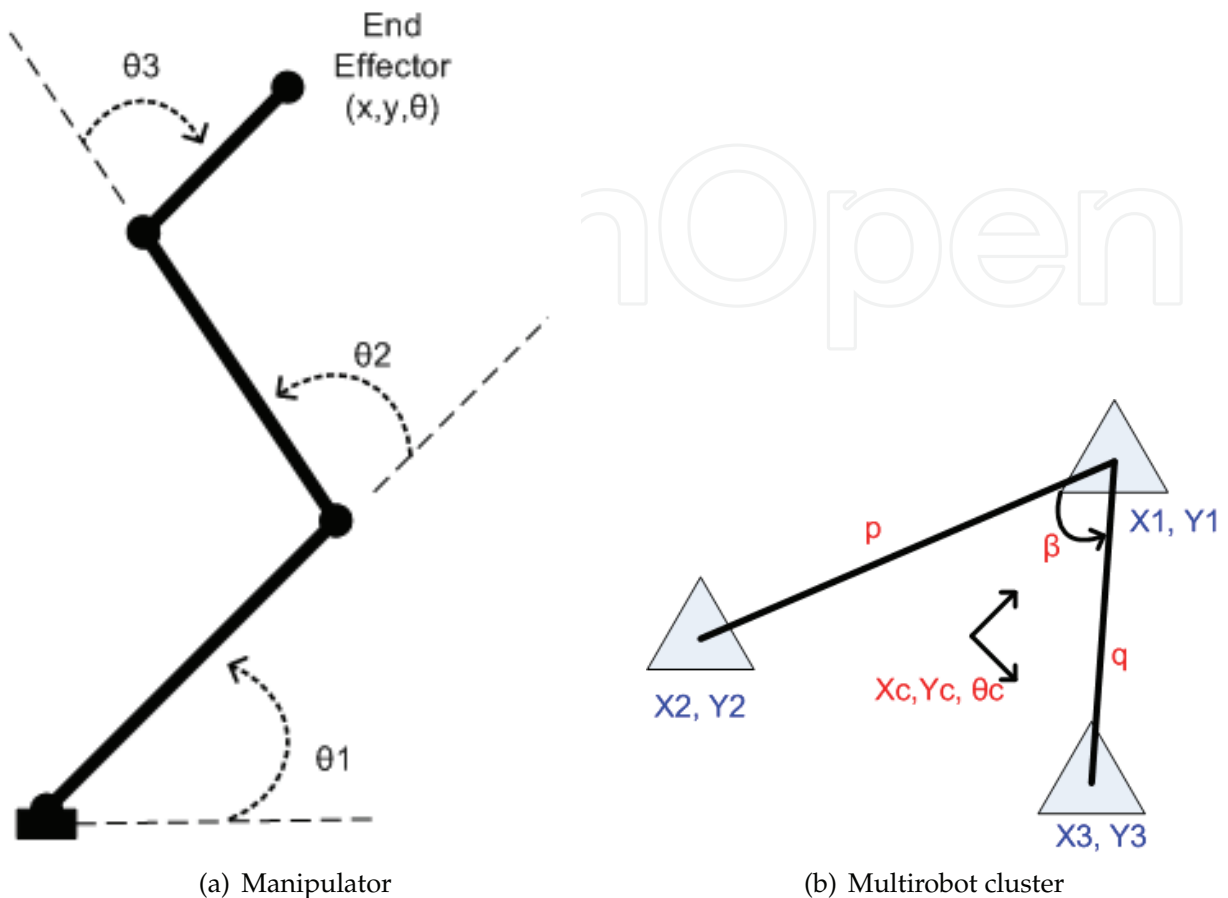(a) Manipulator   (b) Multirobot cluster

Fig. 2. Analogy between definition of variables for a serial chain manipulator and a mobile multirobot system. For manipulator chains, joint space variables are related to operational space variables through formal kinematics. In the cluster framework robot space variables–robot positions–are related to cluster space variables–cluster position, orientation and shape.

2.2 Cluster kinematic relationships

We wish to specify multi-robot 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. We can define $mn \times 1$ robot and cluster pose vectors, r and c, respectively. These state vectors are related through a set of forward and inverse position kinematic relationships:

$$
c = KIN(r) = \begin{pmatrix} g_1(r_1, r_2, \cdots, r_{mn}) \\ g_2(r_1, r_2, \cdots, r_{mn}) \\ \vdots \\ g_{mn}(r_1, r_2, \cdots, r_{mn}) \end{pmatrix} \tag{2}
$$

$$r = INVKIN(c) = \begin{pmatrix} h_1(c_1, c_2, \cdots, c_{mn}) \\ h_2(c_1, c_2, \cdots, c_{mn}) \\ \vdots \\ h_{mn}(c_1, c_2, \cdots, c_{mn}) \end{pmatrix}. \tag{3}$$

We may also consider the formal relationship between the robot and cluster space velocities, $\dot{r}$ and $\dot{c}$. From (2), we may compute the differentials of the cluster space state variables, c_i, and develop a Jacobian matrix, $J(r)$, that maps robot velocities to cluster velocities in the form of a time-varying linear function:

$$\dot{c} = J(r)\dot{r} \tag{4}$$

where

$$J(r) = \begin{pmatrix} \frac{\partial c_1}{\partial r_1} & \frac{\partial c_1}{\partial r_2} & \cdots & \frac{\partial c_1}{\partial r_{mn}} \\ \frac{\partial c_2}{\partial r_1} & \frac{\partial c_2}{\partial r_2} & \cdots & \frac{\partial c_2}{\partial r_{mn}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial c_{mn}}{\partial r_1} & \frac{\partial c_{mn}}{\partial r_2} & \cdots & \frac{\partial c_{mn}}{\partial r_{mn}} \end{pmatrix}. \tag{5}$$

In a similar manner, we may develop the inverse Jacobian, $J^{-1}(c)$, which maps cluster velocities to robot velocities. Computing the robot space state variable differentials from (3) yields:

$$\dot{r} = J^{-1}(c)\dot{c} \tag{6}$$

where

$$J^{-1}(c) = \begin{pmatrix} \frac{\partial r_1}{\partial c_1} & \frac{\partial r_1}{\partial c_2} & \cdots & \frac{\partial r_1}{\partial c_{mn}} \\ \frac{\partial r_2}{\partial c_1} & \frac{\partial r_2}{\partial c_2} & \cdots & \frac{\partial r_2}{\partial c_{mn}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial r_{mn}}{\partial c_1} & \frac{\partial r_{mn}}{\partial c_2} & \cdots & \frac{\partial r_{mn}}{\partial c_{mn}} \end{pmatrix}. \tag{7}$$

3. Cluster space kinematic control

Our initial work in nonlinear cluster space control used a kinematic controller in the cluster space. This controller specifies desired cluster space velocities for the multi-robot formation as a function of the errors in the cluster's position, orientation and shape. These velocities are transformed to robot-specific velocities, which serve as instantaneous command set-points for dynamic speed controllers that execute on each individual robot. This architecture is particularly appropriate for the formation control of robots that have such speed control functionality, which is the case for many commercially available wheeled robots (Schwager et al., 2009). The initial implementation of our nonlinear kinematic controller was developed in (Lee, 2007). Because each robot has an on-board velocity control system, a model of the closed loop dynamics is used rather than a model specific to the robot's physical parameters. This model is combined with the relevant kinematic and frame transforms in order to establish a relationship between the commanded cluster space velocity and the actual cluster space pose:

$$c = \int \dot{c}\, dt = \int J(r)\dot{r}\, dt = \int J(r)\, DYN\, \dot{r}_{cmd}\, dt = \int J(r)\, DYN\, J^{-1}(c)\, \dot{c}_{cmd}\, dt, \tag{8}$$

where DYN is the model of the closed loop dynamics. Given this mapping, a partitioned controller may be developed by inverting this relationship and substituting commanded

cluster velocities with a proportional error-driven control term. With this accomplished, the control equation may be expressed in the form:

$$\dot{c}_{cmd} = \gamma\Big(M(c)K(c_{des} - c) + \beta(c)c\Big). \tag{9}$$

Our work in using this technique has been accomplished using both a dynamic model for each individual robot's translation and rotation as well as for dynamic models of each robot's individual actuators.

3.1 Model-based kinematic cluster control architecture

Kinematic control of the formation is performed by having the controller compute a cluster space velocity command, which is then transformed to a robot space velocity vector using (6). This computation exploits a partitioning strategy, which decomposes the control into a model-based portion and an idealized servo portion. The model-based term exploits knowledge of the formation's dynamics to cancel out nonlinearities and decouple the cluster parameters. Figure 3 shows the kinematic control architecture of the non-linear partitioned controller. The cluster's inverse Jacobian transform converts the controller's cluster space velocity output to the individual velocity commands for each robot in the formation.
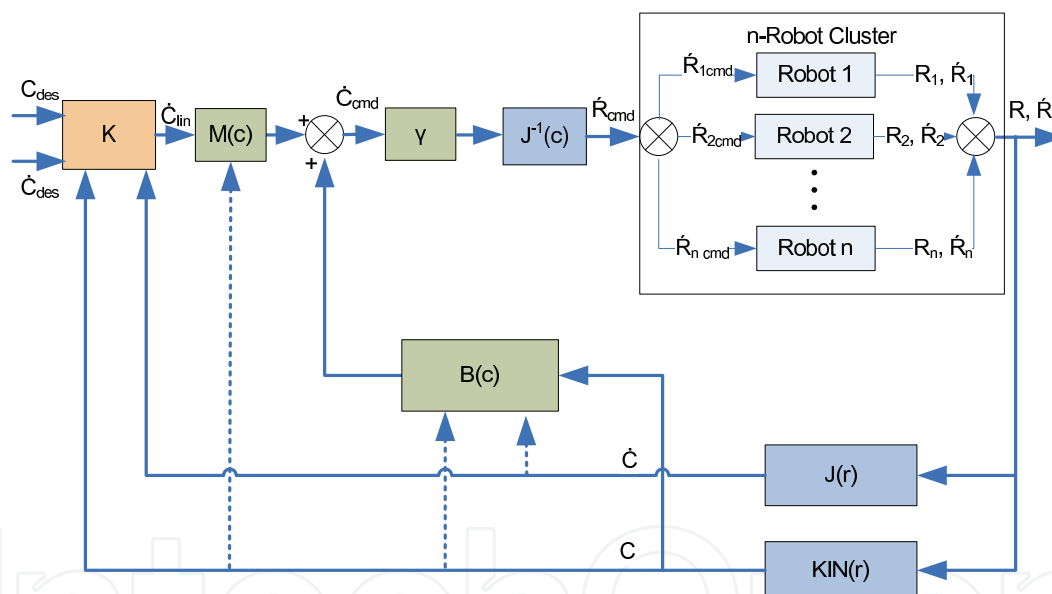

Fig. 3. Model-based kinematic cluster space control architecture for a mobile *n*-robot system. Desired control velocities are computed in cluster space and a partitioned control architecture decouples the system. The inverse Jacobian matrix converts the resulting cluster space velocities to robot space velocities that are then applied to the system. Robot sensor information is converted to cluster space through the Jacobian and kinematic relationships.

3.2 Experiments with a formation of two land rovers

Extensive verification of the nonlinear cluster space controller has been performed through both simulation and hardware experimentation. Here, we summarize a few of the experimental results conducted on a two-robot formation of custom-built omni-wheeled robots. As a two-robot system, the robot space pose is defined as:

$$r = (x_1, y_1, \theta_1, x_2, y_2, \theta_2)^T, \tag{10}$$

where $(x_i, y_i, \theta_i)^T$ defines the position and orientation of robot i. For this formation's cluster space formulation, the cluster frame $\{C\}$ was located at the midpoint of the cluster, as shown in Figure 4. The resulting cluster space pose is represented as:

$$c = (x_c, y_c, \theta_c, \phi_1, \phi_2, d)^T, \tag{11}$$

where $(x_c, y_c, \theta_c)^T$ is the position and orientation of the cluster, ϕ_i is the yaw orientation of robot i relative to the cluster, and d is the single required shape variable defined as half the separation between robots. The cluster space and robot space state variables may be related through a set of forward and inverse kinematic transforms. The derivative of these expressions leads to the forward and inverse velocity kinematic transforms, which are characterized by a Jacobian matrix. The derivation of these transforms are provided in (Lee, 2007). Experimental evaluation of this formulation was conducted with two holonomic,
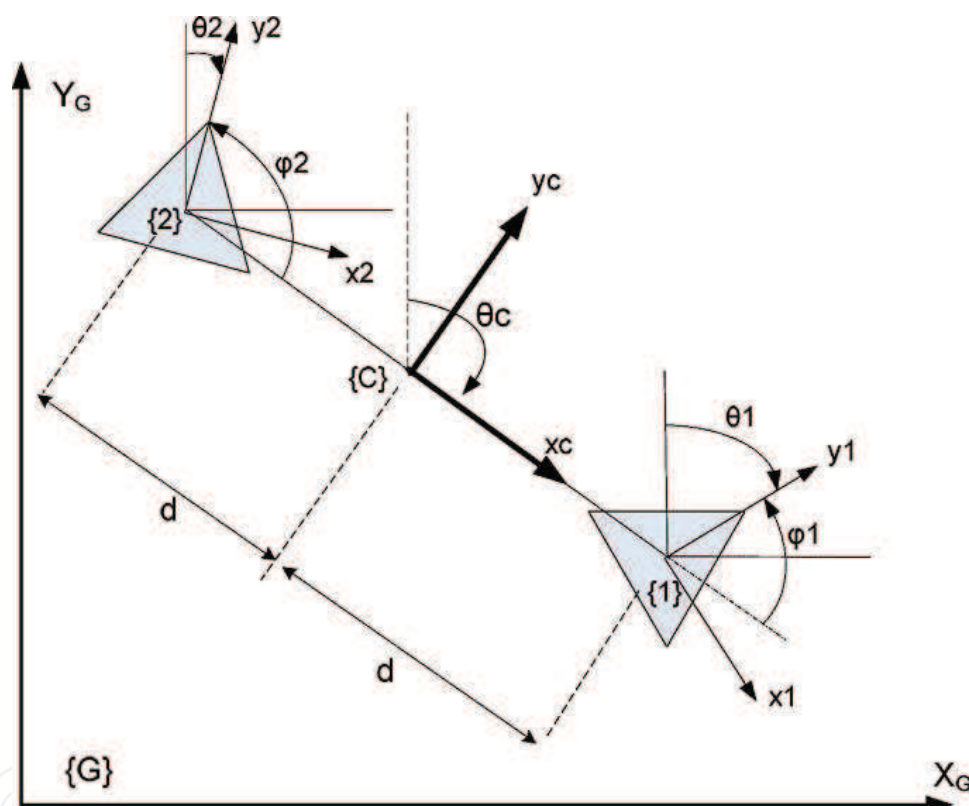

Fig. 4. Reference frame definition placing the cluster center at the center of the two robots

omni-wheeled robots operating in a plane. These student-developed robots consisted of a chassis with three omni-wheels, power components, a sonar suite, a 900 MHz serial radio-modem, and an Atmel AGMEGA128-based microcontroller for on-board control. The robots are capable of closed loop velocity control through the use of wheel encoders and industrial PID controller, with the vehicle-to-wheel inverse kinematic computation performed by the robot's microcontroller. The robots communicate with an off-board control computer for human interfacing and cluster space control computation, and an overhead camera system tracks robot position and orientation. Figure 5 shows the omni-wheeled robots used for experimentation.

Fig. 5. Omni-wheeled robots utilized for experimentation

3.3 Results

As the simplest of the two experimental systems discussed in this article, one test result from (Lee, 2007) is shown here to demonstrate the behavior of the controller. In this test, the two-robot cluster was given concurrent pose step inputs that required a both translation (a relocation of the cluster centroid) and a rotation through a 90 degree angle, while maintaining cluster size. For this experiment, cluster space sensing and control executed at a rate of approximately 2 Hz, and the accuracy of the overhead vision system was approximately +/- 5 cm over the 15 ft x 15 ft workspace. Results of the maneuver are shown in Figure 6. As can be seen, the cluster space variables of interest are each controlled (within the stated accuracy of the position tracking system) as if they were uncoupled, critically damped (with some initial saturation), 2nd order systems, which is the objective of our controller.

4. Cluster space dynamic control

For some robotic platforms, the kinematic model approximation described in the previous section may not hold true and a dynamic approach to modeling and control may be required. Examples of such robots are land rovers with non-negligible dynamics, aerial robots or marine robotic vehicles. For the development of a cluster space dynamic model, it is assumed that the robots composing the system are holonomic and that the formation stays away from singularities. Cluster space singular configurations are described in (Mas, Acain, Petrovic & Kitts, 2009). Next, we will show the relationship between cluster space generalized forces, composed of forces and torques in cluster space, and robot space generalized forces, composed of robot space forces and torques. This derivation is based on the work developed for operational space control of serial chain manipulators presented in (Khatib, 1987) and (Khatib, 1980) and the details are shown in (Mas & Kitts, 2010b). The dynamics of the system in cluster space can be represented by the Lagrangian $\mathcal{L}(c,\dot{c})$:

$$\mathcal{L}(c,\dot{c}) = T(c,\dot{c}) - U(c). \tag{12}$$
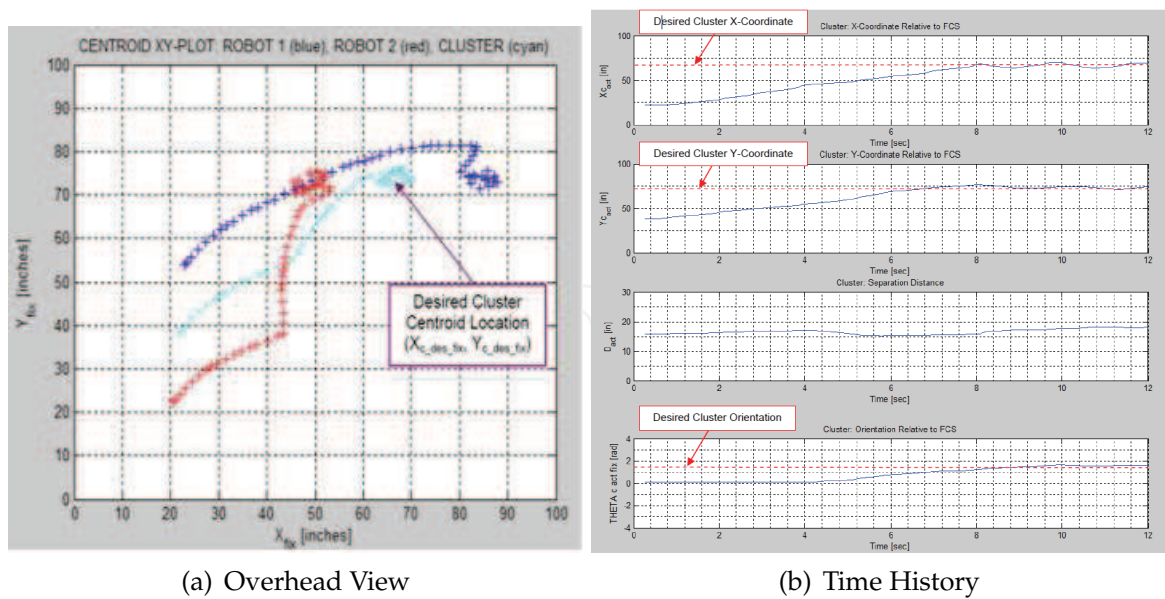
(a) Overhead View   (b) Time History

Fig. 6. Kinematic cluster control experiment results with a formation of two holonomic robots. Position and orientation trajectories.

The kinetic energy of the system can be represented as a quadratic form of the cluster space velocities

$$T(c,\dot{c}) = \frac{1}{2}\dot{c}^T \Lambda(c)\dot{c}, \tag{13}$$

where $\Lambda(c)$ is the $mn \times mn$ symmetric matrix of the quadratic form, i.e., the kinetic energy matrix, and $U(c) = U(KIN(r))$ represents the potential energy due to gravity. For rovers on a plane, the gravity force is canceled out by the force normal to the surface and the gravitational potential energy term can be neglected. For other systems, including aerial unmanned vehicles (AUVs), underwater autonomous vehicles (UAVs) or planar rovers operating on an inclined plane, the gravity term must be included. Let $p(c)$ be the vector of gravity forces in cluster space

$$p(c) = \nabla U(c). \tag{14}$$

Using Lagrangian mechanics, the equations of motion in cluster space are given by

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{c}}\right) - \frac{\partial \mathcal{L}}{\partial c} = F. \tag{15}$$

The equations of motion in cluster space can then be derived from (15) and written in the form

$$\Lambda(c)\,\ddot{c} + \mu(c,\dot{c}) + p(c) = F \tag{16}$$

where $\mu(c,\dot{c})$ is the vector of cluster space centrifugal and Coriolis forces and F is the generalized force vector in cluster space. The equations of motion (16) describe the relationships between positions, velocities, and accelerations of the formation location, orientation, and shape variables and the forces defined in cluster space acting on the formation. The dynamic parameters in these equations are related to the parameters of the robot dynamic models. The dynamics in robot space can by described by

$$A(r)\,\ddot{r} + b(r,\dot{r}) + g(r) = \Gamma \tag{17}$$

where $b(r,\dot{r})$, $g(r)$ and Γ represent, respectively, velocity dependent forces, gravity and generalized forces in robot space. $A(r)$ is the $mn \times mn$ robot space kinetic energy matrix. The relationship between the kinetic energy matrices $A(r)$ and $\Lambda(c)$ corresponding, respectively, to the robot space and cluster space dynamic models can be established (Khatib, 1980; Mas & Kitts, 2010b) by exploiting the identity between the expressions of the quadratic forms of the system kinetic energy with respect to the generalized robot and cluster space velocities,

$$\Lambda(c) = J^{-T}(r)\, A(r)\, J^{-1}(r). \tag{18}$$

The relationship between $b(r,\dot{r})$ and $\mu(c,\dot{c})$ can be established by the expansion of the expression of $\mu(c,\dot{c})$ that results from (15),

$$\mu(c,\dot{c}) = J^{-T}(r)\, b(r,\dot{r}) - \Lambda(r)\, \dot{J}(r,\dot{r})\, \dot{r}. \tag{19}$$

The relationship between the expressions of gravity forces can be obtained using the identity between the functions expressing the gravity potential energy in the two spaces and the relationships between the partial derivatives with respect to the variables in these spaces. Using the definition of the Jacobian matrix (6) yields

$$p(c) = J^{-T}(r)\, g(r). \tag{20}$$

Finally, we can establish the relationship between generalized forces in cluster space and robot space, F and Γ. Using (18), (19), and (20), the cluster space equations of motion (16) can be rewritten as

$$J^{-T}(r)\left[A(r)\,\ddot{r} + b(r,\dot{r}) + g(r)\right] = F. \tag{21}$$

Substituting (17) yields

$$\Gamma = J^{T}(r)\, F \tag{22}$$

which represents the fundamental relationship between cluster space forces and robots space forces. This relationship is the basis for the dynamic control of the robot formation from the cluster space perspective.

4.1 Model-based dynamic cluster control architecture

The dynamic control of the formation is performed by generating a cluster space generalized force vector F that is then transformed to a robot space force vector Γ using (22). In order to obtain such a control vector, we use a nonlinear dynamic decoupling approach (Craig, 2005). In this approach, we partition the controller into a model-based portion and a servo portion. The model-based portion uses the dynamic model of the cluster to cancel out nonlinearities and decouple the cluster parameters. The resulting control law then has the form

$$F = \Lambda(c)\, F_m + \mu(c,\dot{c}) + p(c). \tag{23}$$

where $\Lambda(c)$, $\mu(c,\dot{c})$ and $p(c)$ are the cluster space dynamic model parameters. F_m is the command force vector acting on an equivalent cluster space unit mass decoupled system, which we define as

$$F_m = \ddot{c}_{des} + K_p e_c + K_v \dot{e}_c, \tag{24}$$

where $e_c = c_{des} - c$ and $\dot{e}_c = \dot{c}_{des} - \dot{c}$ are, respectively, the cluster space position and velocity errors, and K_p and K_v are positive definite matrices. Figure 7 shows the dynamic control architecture of the non-linear partitioned controller.
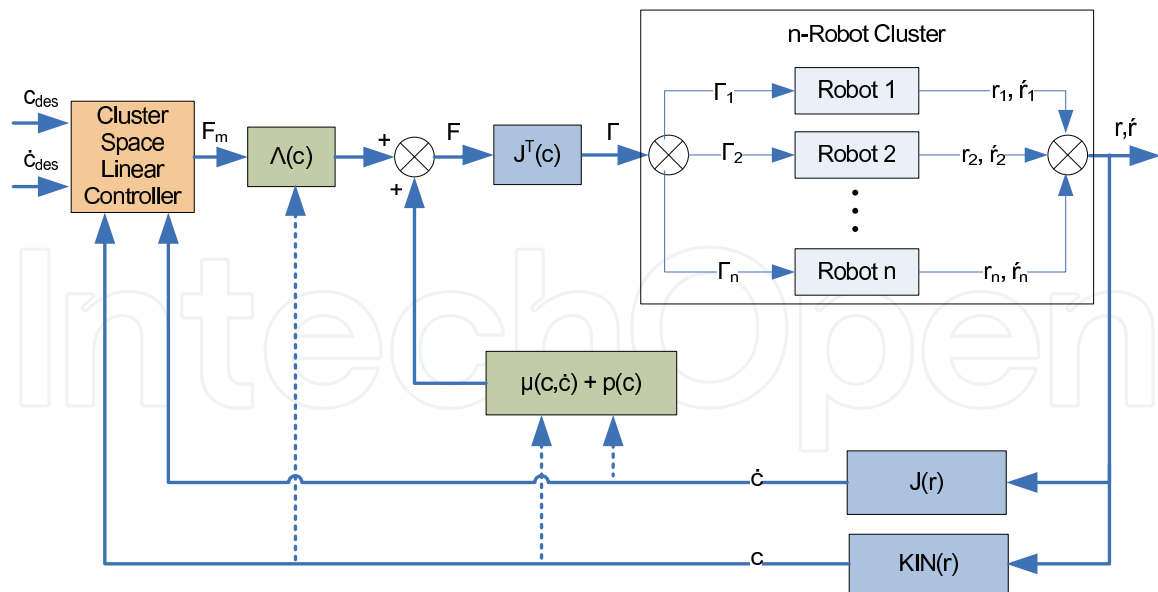
Fig. 7. Model-based dynamic cluster space control architecture for a mobile n-robot system.

4.2 Experiments with a formation of three surface vessel marine robots

To illustrate the functionality of the proposed formation control approach applied to systems with non-negligible dynamics, we conducted experimental tests with a group of three autonomous surface vessels (ASV). In order to apply the method to a planar three-robot system, the cluster space variables must be defined and the kinematic transforms must be generated. Figure 8 depicts the relevant reference frames for the planar three-robot problem. We have chosen to locate the cluster frame {C} at the cluster's centroid, oriented with Y_c pointing toward robot 1. Based on this, the nine robot space state variables (three robots with three DOF per robot) are mapped into nine cluster space variables for a nine DOF cluster.

Given the parameters defined by Figure 8, the robot space pose vector is defined as:

$$\overrightarrow{r} = (x_1, y_1, \theta_1, x_2, y_2, \theta_2, x_3, y_3, \theta_3)^T, \tag{25}$$

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

$$\overrightarrow{c} = (x_c, y_c, \theta_c, \phi_1, \phi_2, \phi_3, p, q, \beta)^T, \tag{26}$$

where $(x_c, y_c, \theta_c)^T$ is the cluster position and orientation, ϕ_i is the yaw orientation of robot i relative to the cluster, p and q are the distances from robot 1 to robots 2 and 3, respectively, and β is the skew angle with vertex on robot 1. Given this selection of cluster space state variables, we can express the forward and inverse position kinematics of the three-robot system. These complete expressions can be found in (Mas, Li, Acain & Kitts, 2009). By differentiating the forward and inverse position kinematic equations, the forward and inverse velocity kinematics can easily be derived, obtaining the Jacobian and inverse Jacobian matrices. It should be noted that this particular selection of cluster space variables is not unique, and different sets of variables may be chosen following the same framework when more convenient for a given task. To validate the approach with experimental results, a testbed of three autonomous surface vessel (ASV) marine robots is used. Each robot is an off-the-shelf kayak retrofitted with two thrusters producing a differential drive behavior and an electronics box that includes motor controllers, GPS, a compass, and a wireless communication system. A remote central computer receives sensor information from the ASVs, executes the cluster
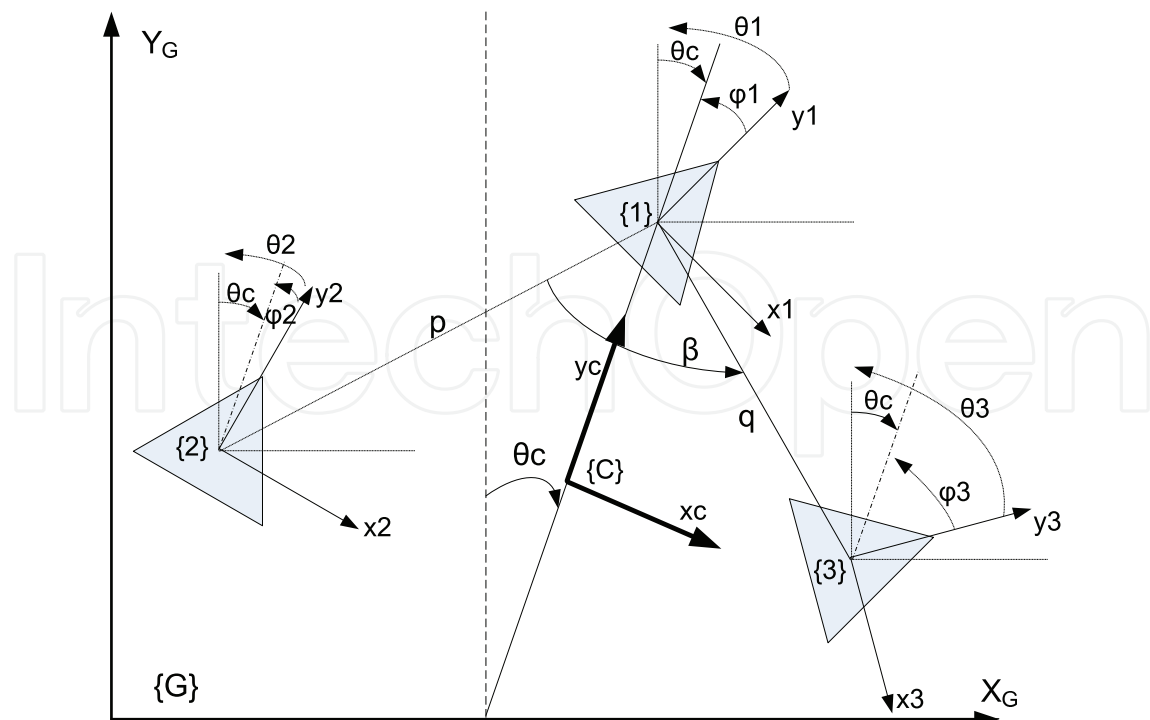
Fig. 8. Reference frame definition placing the cluster center at the triangle centroid

controller algorithms, and sends the appropriate compensation signals. A detailed description of this testbed can be found in (Mahacek et al., 2009). Figure 9 shows the ASVs used for the experiments. To accommodate for the non-holonomic constraints, a robot-level heading control inner-loop is implemented on each robot to achieve required bearings. The model-based partitioned controller makes use of a dynamic model of the cluster to compute the appropriate compensation. The cluster dynamic equation is obtained through the model parameters of the ASVs. Using (17), the parameters for the ith ASV are (Mahacek, 2009):

$$A_i(r) = \begin{pmatrix} 150kg & 0 & 0 \\ 0 & 150kg & 0 \\ 0 & 0 & 41kgm^2 \end{pmatrix}, \tag{27}$$

$$b_i(r, \dot{r}) = \begin{pmatrix} 100\frac{kg}{s}\dot{r}_x \\ 400\frac{kg}{s}\dot{r}_y \\ 25\frac{kgm^2}{s\,rad}\dot{\theta} \end{pmatrix}, \tag{28}$$

$$g_i(r) = 0. \tag{29}$$

Using (18), (19), and the Jacobian matrices given by the cluster definition, the cluster dynamic parameters can be computed in execution time to produce dynamic compensation in the controller.

4.3 Results

Two experimental tests implementing the cluster dynamic controller are shown in this article. On the first one, the formation of ASVs follows a rectangular position trajectory, composed with a cluster rotation on the second half of it. The cluster shape parameters are held constant throughout the test. An overhead view of the resulting motions, and desired and measured

Fig. 9. Autonomous surface vehicles with propulsion systems and custom sensor and communication suites.

values for the cluster parameters over time are shown in Figure 10. In the second test, the position and orientation of the cluster are held constant and the shape parameters follow specified trajectories. An overhead view, and desired and measured values for the cluster parameters over time are shown in Figure 11. In both runs, the cluster parameters follow their desired values over time. Table 1 shows the mean squared errors for the cluster space parameters in both tests. Position sensing errors due to GPS receivers as well as errors in the estimation of the ASVs dynamic model parameters result in tracking errors during the experiments. In the second test, additional environmental disturbances introduced by wind and currents decreased the system performance. Overall, the experiments illustrate the basic functionality of the non-linear partition controlled model-based approach to cluster control of formations of mobile robots with non-negligible dynamics.

	MSE	
Cluster Parameter	Test 1 - Position Traj.	Test 2 - Shape Traj.
x_c (m^2)	1.26065	9.65675
y_c (m^2)	1.36074	4.58394
θ_c (rad^2)	0.00446	0.00003
p (m^2)	0.53422	4.01754
q (m^2)	0.17589	20.37102
β (rad^2)	0.07355	0.00016

Table 1. Experimental Results. Mean Square Errors for Test 1–Position Trajectories–and Test 2–Shape Trajectories–.

5. Conclusions

The cluster space control approach for planar robots was briefly reviewed and two alternative model-based non-linear control architectures were presented. A kinematic approach to controlling formations of robots that have on-board closed-loop velocity control capabilities was presented. This controller exploits a partitioning strategy, which decomposes the control
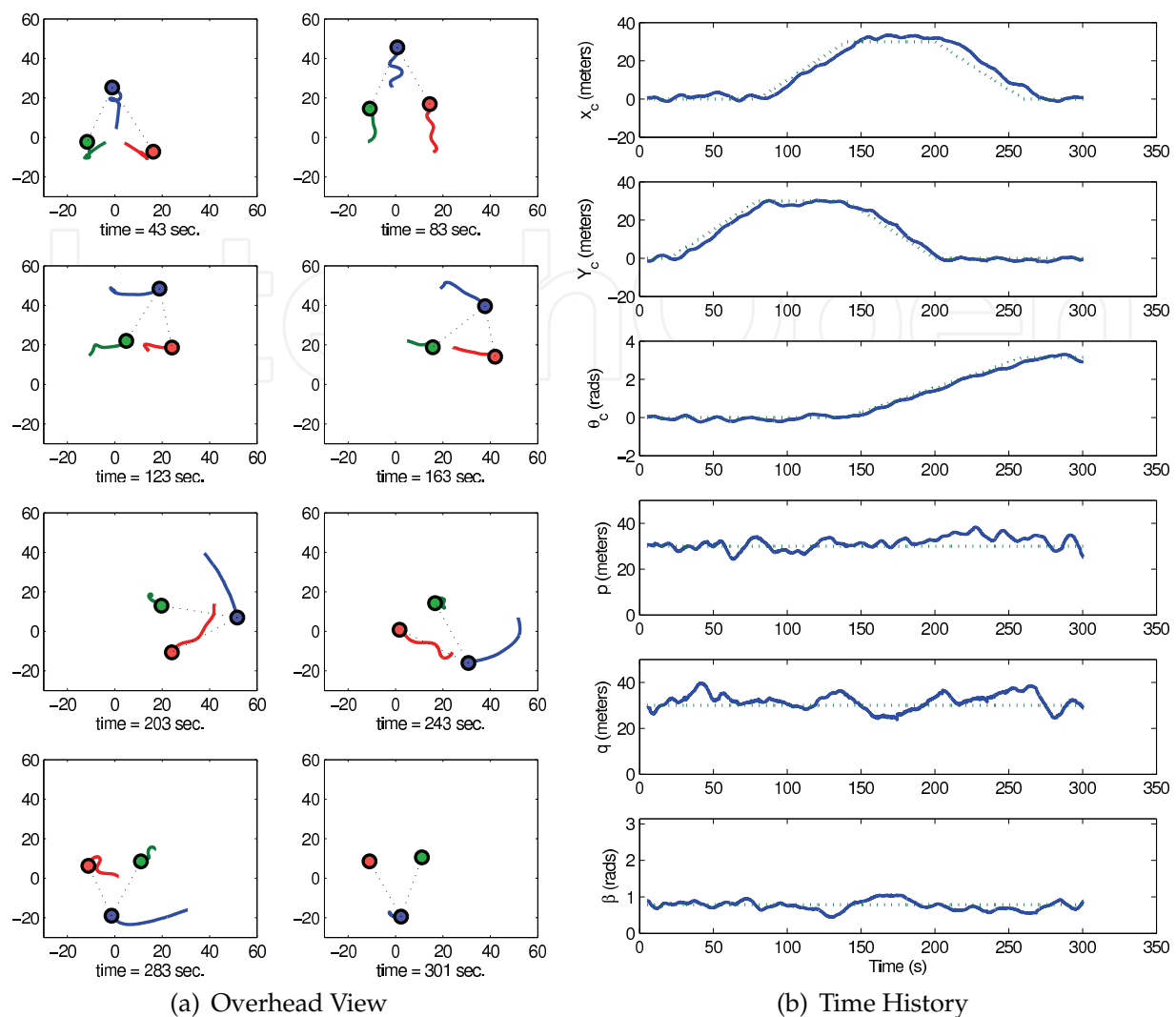
Fig. 10. Dynamic cluster control experiment 1 results. Position and orientation trajectories.

into a model-based portion and an idealized servo portion. The model-based term exploits knowledge of the formation's dynamics to cancel out nonlinearities and decouple the cluster parameters. Experimental results using two omni-wheeled robots illustrate the effectiveness of the architecture. A dynamic approach to be used when the dynamics of the robots are not negligible was proposed and the equations of motion for the cluster space variables were derived. The parameters of the cluster space dynamics were then defined as a function of the dynamic parameters of the robots in the formation. It was shown that generalized forces in cluster space can be related to forces in robot space through the Jacobian transpose matrix. A non-linear cluster level dynamic partitioned controller was proposed. The model-based portion of such a controller cancels out the cluster space non-linear dynamics and allows for the cluster variables to be decoupled. The servo portion of the controller then effectively sees a set of decoupled unit mass plants. The proposed model-based dynamic controller was then applied to an experimental testbed composed of three ASVs. Results were presented in order to demonstrate the functionality of the system. The experiments showed the ability of the formation to navigate following position, orientation, and shape trajectories. Ongoing work includes the integration of obstacle avoidance methods and addressing in detail the
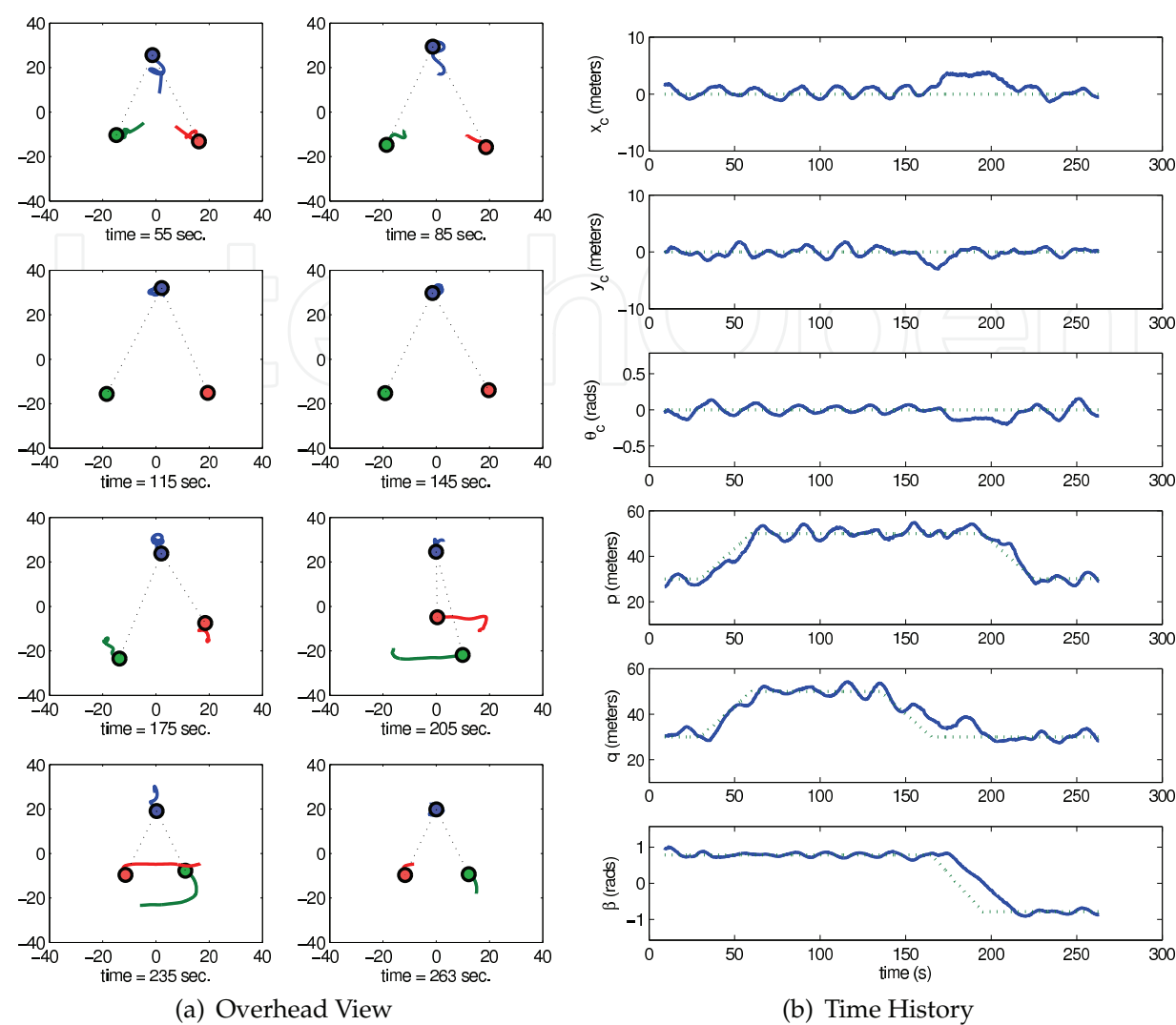
(a) Overhead View   (b) Time History

Fig. 11. Dynamic cluster control experiment 2 results. Formation shape trajectories.

impact of errors in the estimations of the dynamic parameters of the robots. The study of alternative cluster definitions is being conducted under the assumption that they may be more convenient for specifying and monitoring requirements for different missions, they can be used to avoid singularities, and can be selected to reduce computational requirements. Future applications using the cluster space approach include marine environment survey via vehicle differential measurements and dynamic beamforming using cluster controlled smart antennae arrays (Okamoto et al., 2010).

6. Acknowledgments

7. References

Agnew, S. (2009). *Cluster space control of a 2-robot aerial system*, Master's thesis, Santa Clara University.

Balch, T. & Hybinette, M. (2000). Behavior-based coordination of large-scale robot formations, *MultiAgent Systems, 2000. Proceedings. Fourth International Conference on* pp. 363–364.

Connolley, P. (2006). *Design and implementation of a cluster space trajectory controller for multiple holonomic*, Master's thesis, Santa Clara University.

Craig, J. (2005). *Introduction to Robotics, Mechanics and Control*, Pearson Prentice Hall, Third Edition.

Dunbar, W. & Murray, R. M. (2006). Distributed receding horizon control for multi-vehicle formation stabilization, *Automatica* 42(4): 549–558.

Flinn, E. (2005). Testing for the 'boids', *Aerospace America* 43(6): 28–29.

Girod, E. (2008). *Design and implementation of four robot cluster space control*, Master's thesis, Santa Clara University.

Hashimoto, M., Oba, F. & Eguchi, T. (1993). Dynamic control approach for motion coordination of multiple wheeled mobile robots transporting a single object, *Intelligent Robots and Systems '93, IROS '93. Proceedings of the 1993 IEEE/RSJ International Conference on* 3: 1944–1951 vol.3.

Ikeda, M. (1989). Decentralized control of large scale systems, *Three decades of mathematical system theory: A collection of surveys at the occasion of the 50th birthday of Jan C. Willems* pp. 219–242.

Ishizu, R. (2005). *The design, simulation and implementation of multi-robot collaborative control from cluster perspective*, Master's thesis, Santa Clara University.

Justh, E. W. & Krishnaprasad, P. S. (2004). Equilibria and steering laws for planar formations, *Systems and Control Letters* 52: 25 – 38.

Kalkbrenner, M. (2006). *Design and implementation of a cluster space human interface controller, including applications to multi-robot piloting and multi-robot object manipulation*, Master's thesis, Santa Clara University.

Khatib, O. (1980). *Commande Dynamique dans l'espace opérationnel des robots manipulateurs en présence d'obstacles*, Thése de docteur-ingénieur, Ecole Nationale Supérieure de l'aéronautique et de l'espace, Toulouse, France.

Khatib, O. (1985). The potential field approach and operational space formulation in robot control, *Proc. Fourth Yale Workshop on Applications of Adaptive Systems Theory, Yale University, New Haven, Connecticut* pp. 208–214.

Khatib, O. (1987). A unified approach for motion and force control of robot manipulators: The operational space formulation, *Robotics and Automation, IEEE Journal of* 3(1): 43 –53.

Kitts, C. A. & Mas, I. (2009). Cluster space specification and control of mobile multirobot systems, *Mechatronics, IEEE/ASME Transactions on* 14(2): 207–218.

Kitts, C. & Egerstedt, M. (2008). Design, control, and applications of real-world multirobot systems [from the guest editors], *Robotics & Automation Magazine, IEEE* 15(1): 8–8.

Kitts, C., Stanhouse, K. & Chindaphorn, P. (2009). Cluster space collision avoidance for mobile two-robot systems, *Intelligent Robots and Systems, IEEE/RSJ International Conference on* pp. 1941–1948.

Lee, R. (2007). *Model-based cluster-space control of multiple robot systems*, Master's thesis, Santa Clara University.

Leonard, N. & Fiorelli, E. (2001). Virtual leaders, artificial potentials and coordinated control of groups, *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*

3: 2968–2973 vol.3.

Mahacek, P. (2009). *Design and cluster space control of two autonomous surface vessels*, Master's thesis, Santa Clara University.

Mahacek, P., Mas, I., Petrovic, O., Acain, J. & Kitts, C. (2009). Cluster space control of autonomous surface vessels, *Marine Technology Society Journal* 43(1): 13–20.

Mas, I., Acain, J., Petrovic, O. & Kitts, C. (2009). Error characterization in the vicinity of singularities in multi-robot cluster space control, *2008 IEEE Robotics and Biomimetics, Bangkok, Thailand. IEEE International Conference on* .

Mas, I. & Kitts, C. (2010a). Centralized and decentralized multi-robot control methods using the cluster space control framework, *Advanced Intelligent Mechatronics, IEEE/ASME International Conference on - in press* .

Mas, I. & Kitts, C. (2010b). Dynamic control of mobile multi-robot systems: The cluster space formulation, *Mechatronics, IEEE Transactions on* p. Submitted.

Mas, I., Li, S., Acain, J. & Kitts, C. (2009). Entrapment/escorting and patrolling missions in multi-robot cluster space control, *Intelligent Robots and Systems. IEEE/RSJ International Conference on* pp. 5855–5861.

Mas, I., Petrovic, O. & Kitts, C. (2008). Cluster space specification and control of a 3-robot mobile system, *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on* pp. 3763–3768.

Murray, R. M. (2007). Recent research in cooperative control of multi-vehicle systems, *Journal of Dynamics Systems, Measurement and Control* 129(5): 571–583.

Ogren, P., Fiorelli, E. & Leonard, N. (2004). Cooperative control of mobile sensor networks:adaptive gradient climbing in a distributed environment, *Automatic Control, IEEE Transactions on* 49(8): 1292 – 1302.

Okamoto, G., Chen, C. & Kitts, C. (2010). Beamforming performance for a reconfigurable sparse array smart antenna system via multi-robot control, *Proceedings of the SPIE Defense, Security, and Sensing Conference*.

Rus, D., Donald, B. & Jennings, J. (1995). Moving furniture with teams of autonomous robots, *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on* 1: 235–242 vol.1.

Schwager, M., Rus, D. & Slotine, J. (2009). Decentralized, adaptive coverage control for networked robots, *The International Journal of Robotics Research* 28(3): 357–375.

Siljak, D. (1991). *Decentralized Control of Complex Systems*, New York: Academic.

Stipanovic, D. M., Inalhan, G., Teo, R. & Tomlin, C. J. (2004). Decentralized overlapping control of a formation of unmanned aerial vehicles.

Tan, K.-H. & Lewis, M. (1996). Virtual structures for high-precision cooperative mobile robotic control, *Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on* 1: 132–139 vol.1.

Tang, C. P., Bhatt, R., Abou-Samah, M. & Krovi, V. (2006). Screw-theoretic analysis framework for cooperative payload transport by mobile manipulator collectives, *Mechatronics, IEEE/ASME Transactions on* 11(2): 169 –178.

To, T. (2006). *Automated cluster space trajectory control of two non-holonomic robots*, Master's thesis, Santa Clara University.

Tully, B. (2006). *Cluster space piloting of a non-holonomic multi-robot system*, Master's thesis, Santa Clara University.

Yamaguchi, H. & Arai, T. (1994). Distributed and autonomous control method for generating shape of multiple mobile robot group, *Intelligent Robots and Systems '94.*

'Advanced Robotic Systems and the Real World', IROS '94. Proceedings of the IEEE/RSJ/GI International Conference on 2: 800–807 vol.2.

Yang, T., Yu, H., Fei, M. & Li, L. (2005). Networked control systems: a historical review and current research topics, *Measurement and Control* 38(1): 12–16.

Multi-Robot Systems, Trends and Development

Edited by Dr Toshiyuki Yasuda

This book is a collection of 29 excellent works and comprised of three sections: task oriented approach, bio inspired approach, and modeling/design. In the first section, applications on formation, localization/mapping, and planning are introduced. The second section is on behavior-based approach by means of artificial intelligence techniques. The last section includes research articles on development of architectures and control systems.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ignacio Mas, Christopher Kitts and Robert Lee (2011). Model-Based Nonlinear Cluster Space Control of Mobile Robot Formations, Multi-Robot Systems, Trends and Development, Dr Toshiyuki Yasuda (Ed.), ISBN: 978-953-307-425-2, InTech, Available from: http://www.intechopen.com/books/multi-robot-systems-trends-and-development/model-based-nonlinear-cluster-space-control-of-mobile-robot-formations

INTECH
open science | open minds