Received April 2, 2014, accepted May 10, 2014, date of publication May 19, 2014, date of current version May 30, 2014.

Digital Object Identifier 10.1109/ACCESS.2014.2325742

Dynamic Control of Mobile Multirobot Systems: The Cluster Space Formulation

IGNACIO MAS[1], (Member, IEEE), AND CHRISTOPHER A. KITTS[2], (Senior Member, IEEE)
[1]Instituto Tecnologico de Buenos Aires, Consejo Nacional de Investigaciones Cientificas y Tecnicas, Ciudad de Buenos Aires C1106ACD, Argentina
[2]Department of Mechanical Engineering, Santa Clara University, Santa Clara, CA 95053 USA

Corresponding author: I. Mas (imas@itba.edu.ar)

This work was supported in part by the Santa Clara University (SCU) Robotic Systems Laboratory, in part by the SCU School of Engineering, in part by the the SCU Sustainability Program, in part by the SCU Technology Steering Committee, in part by NOAA under Grant NA03OAR4300104 through the University of Alaska Fairbanks under Subaward UAF 05-0148, and in part by the National Science Foundation under Grant CNS-0619940.

ABSTRACT The formation control technique called cluster space control promotes simplified specification and monitoring of the motion of mobile multirobot systems of limited size. Previous paper has established the conceptual foundation of this approach and has experimentally verified and validated its use for various systems implementing kinematic controllers. In this paper, we briefly review the definition of the cluster space framework and introduce a new cluster space dynamic model. This model represents the dynamics of the formation as a whole as a function of the dynamics of the member robots. Given this model, generalized cluster space forces can be applied to the formation, and a Jacobian transpose controller can be implemented to transform cluster space compensation forces into robot-level forces to be applied to the robots in the formation. Then, a nonlinear model-based partition controller is proposed. This controller cancels out the formation dynamics and effectively decouples the cluster space variables. Computer simulations and experimental results using three autonomous surface vessels and four land rovers show the effectiveness of the approach. Finally, sensitivity to errors in the estimation of cluster model parameters is analyzed.

INDEX TERMS Cluster space, multirobot systems, dynamic control, formation control, marine robotics.

I. INTRODUCTION

Autonomous or tele-operated robotic systems offer many advantages to accomplishing a wide variety of tasks given their strength, speed, precision, repeatability, and ability to withstand extreme environments. Whereas 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, and spatially diverse functionality [1]. 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, drawing on work in control theory, robotics, and biology [2] 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 [3], [4]. 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 [5]. 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 [6]. 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 [7], [8]. 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 [9], [10].

Many of these formation control techniques have been applied to marine surface and land rover vehicles. A biologically inspired behavioral-based method has been used to control multiple underwater vehicles [11]. A distributed elastic behavior for a deformable chain-like formation of small autonomous underwater vehicles has also been reported in the literature [12]. Formation control of multiple vehicles that form a floating runway at sea has been also proposed [13], and the virtual structure approach has been used to control a fleet of vessels [14]. The leader-follower method was also applied to control groups of autonomous surface vessels [15] and wheeled robots [16].

A. CLUSTER SPACE APPROACH

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

In contrast to swarm-like formation control methods [17], [18], where the benefits are obtained by abstracting to a low dimensional representation, our approach maintains the same dimensionality in the cluster level. This allows for a fully controllable system–where the position of each member can be specified and controlled– as well as for formation state monitoring, where instantaneous values of cluster parameters can be obtained from robot state measurements. The simplification is given by the selection of cluster space parameters in such a way that a subset of them is relevant to the task and another subset has secondary importance (e.g. can be kept constant throughout the task). This restricts the scalability of the method to formations of ones to tens of robots. We believe that a wide variety of multirobot applications still fit within this category, and alternative methods can be adopted when larger formations are required. Complex applications that take advantage of the full degree of freedom control capability provided by the cluster space framework, such as marine dynamic asset guarding [19] or object manipulation [20] were demonstrated and others continue to be explored.

Previous work presented a generalized framework for developing the cluster space approach for a system of n robots, each with m degrees of freedom (DOF) [21]. This framework has been successfully demonstrated implementing kinematic controllers—where the dynamics of the system are considered negligible—for both holonomic and non-holonomic systems, with augmentations for potential field-based obstacle avoidance [22], and utilizing different robot platforms ranging from planar rovers [23] to marine autonomous surface vessels [24] and aerial blimps [25]. Centralized and distributed implementations of cluster space control were also proposed [26].

B. ROBOT MANIPULATOR CONTROL ANALOGY

A significant comparison can be made between the cluster space method and the Cartesian (or operational) space control approach that has been developed for serial chain manipulators [27], [28] and has also been applied to humanoid robots [29]. 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. This similarity leads to many cases in which manipulator-oriented analyses and control approaches can be applied to the cluster space control of mobile robot clusters. For example, the Jacobian transpose transform can be used to relate the dynamic forces in each of these spaces, as is confirmed in Section IV.

In contrast, there are several differences between the cluster space method and Cartesian control of serial chain manipulators. These differences often lead to significant departures in the implementation of analyses and control approaches, and on occasion they yield new opportunities and/or challenges. One obvious example is that mobile robot clusters are not physically connected as are manipulators; therefore, we consider them to be virtual articulating mechanisms that lack interconnecting structural forces or torques. As another example, serial manipulator control methods employ strict kinematic conventions to prescribe the location of link frames, define geometric link parameters, and specify joint space position variables. Our cluster space method provides significant flexibility in this regard, allowing a range of options in assigning the cluster frame, in numbering the robots, and in defining cluster shape parameters. Indeed, it is this flexibility that allows the approach to be adapted for a wide range of purposes such as tailoring the shape description based on operator preference, adapting the level of control (de)centralization [26], and dynamically switching representations to avoid computational singularities [30]. Unfortunately, this flexibility leads to a far more challenging effort to perform tasks such as deriving the cluster Jacobian transforms, particularly since some cluster shape descriptions may yield parallel robot chains.

C. CONTRIBUTIONS

In this paper, after introducing the cluster space approach in Section II, we propose in Section III a model for cluster space formation dynamics and define its parameters as functions of the dynamics of the robots forming the group. The cluster dynamic equations, which were briefly presented in [31], are here for the first time fully derived from the robots' dynamics using Lagrangian mechanics. We prove that generalized forces in cluster space are related to generalized forces in robot space through the cluster Jacobian transpose matrix. Then we propose a Jacobian transpose type of controller, similar to that used for Cartesian control of serial chain manipulators, that can be used to control the robot formation from the cluster perspective. The resulting cluster-level control actions are transformed into robot-level forces applicable to the members of the formation. In Section IV a nonlinear model-based partition controller is presented and its stability is addressed in the Lyapunov sense.

In order to demonstrate the functionality of the proposed methodology, in Section V we apply it to different robotic platforms. First, simulation results show the advantages of the controller. Then, experimental results utilizing three marine autonomous surface vessels and four wheeled land rovers illustrate the applicability of the method to real-world systems.

Previous work by the authors on this line of research focused on kinematic modeling and control of robot formations for which the dynamics could be neglected. Such method was valid for a limited universe of robots in a limited universe of environments. The novel approach presented here introduces the capability of modeling, from the formation perspective, the dynamics of the robots in the group as well as the effects of environment perturbations, expanding the theory to a more comprehensive space of applications.

II. CLUSTER SPACE FRAMEWORK OVERVIEW

The cluster space approach to controlling formations of multiple robots was first introduced in [21]. 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 Figure 1.

Typical robot-oriented representations of pose use mn variables to represent the position and orientation of each of the robot body frames, $\{1\}, \{2\}, \ldots, \{n\}$, 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 geometrically meaningful way, such as at the cluster's centroid and oriented toward a particular vehicle or alternatively, coincident with one of the vehicle's body frame. An additional set of variables defining the shape of the formation complete the representation.

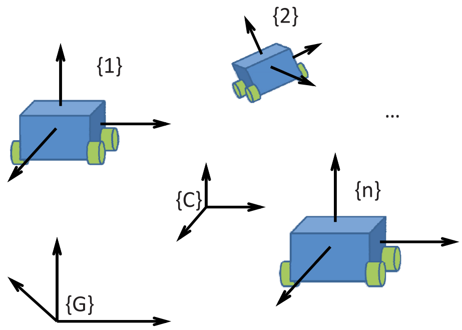

FIGURE 1. Cluster and robot frame descriptions with respect to a global frame.

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

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

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, as well as typical selections for given systems, are described in [21].

B. CLUSTER KINEMATIC RELATIONSHIPS

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.

1) POSITION KINEMATICS

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:

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

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

2) VELOCITY KINEMATICS

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 an $mn \times mn$ Jacobian matrix, $J(r)$, that maps robot velocities to cluster velocities in the form of a time-varying linear function:

$$\dot{c} = J(r)\,\dot{r}. \quad (4)$$

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

$$\dot{r} = J^{-1}(c)\,\dot{c}. \quad (5)$$

C. EXAMPLE KINEMATIC DEFINITIONS

As an example of alternate kinematic definitions for mobile robot clusters, consider the different ways of representing a three-robot planar cluster as shown in Figure 2. In each of these cases, the conventional robot space pose would be represented by $r = (x_1, y_1, \theta_1, x_2, y_2, \theta_2, x_3, y_3, \theta_3)$ where each (x_i, y_i, θ_i) represents the linear and angular location for robot i. The cluster space pose description is represented by $c = (x_c, y_c, \theta_c, s_1, s_2, s_3, \phi_1, \phi_2, \phi_3)$ where (x_c, y_c, θ_c) specifies the cluster frame position and orientation, the s_i variables specify the cluster shape, and the ϕ_i variables represent the relative orientation of each robot with respect to the cluster frame. However, since each of the three clusters have been defined differently, they will have different cluster frame position values, and their shape parameters will be completely distinct.

These varying representations lead to a range of kinematic dependencies that drive critical characteristics of the formation control architecture. For example -as shown in Figure 2- with the leader-follower representation, the sparse nature of the resulting Jacobian transforms clearly indicate the decentralized nature of the resulting controller; in contrast, for the third example with the cluster frame at the centroid of the triangle, the Jacobian transforms are nearly fully populated with nonzero terms, indicating the tight interdependence of robot positions with respect to achieving a specific cluster space pose. As a side note, it is interesting to observe that the leader-follower configuration, which is perhaps the most used formation control approach, becomes a simple implementation that is subsumed within the range of options provided by the cluster space control methodology.

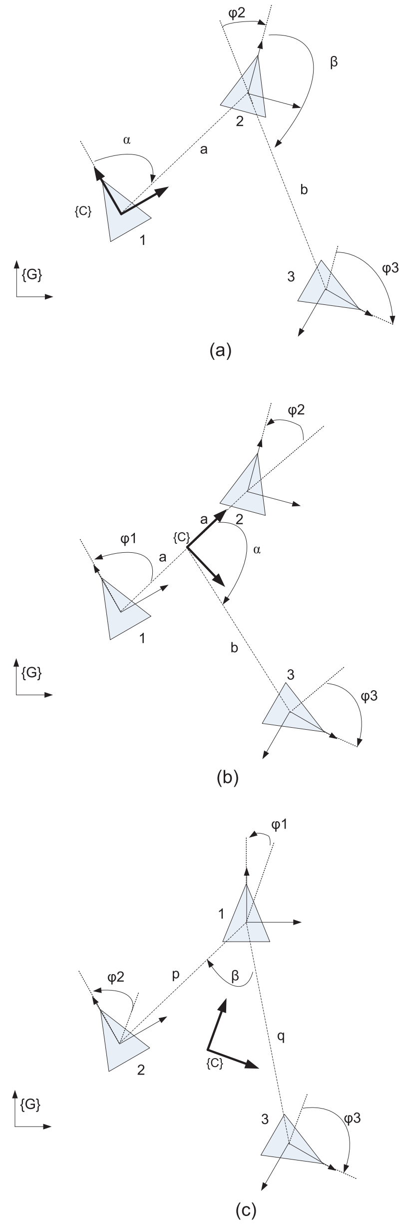

FIGURE 2. Three-robot planar cluster representation options. These three representations show variation in assigning the location of the cluster frame, numbering the robots, and defining the shape of the cluster: (a) A leader-follower chain approach in which the cluster frame is placed on the lead robot and the location of the subsequent robot is provided by distance and angle shape variables; (b) The cluster frame is located between robots 1 and 2, oriented towards robot 2, while robot 3's location is described with a distance-angle description with respect to this frame; (c) A highly integrated geometric description in which the cluster frame is placed at the cluster centroid and oriented towards robot 1 and the triangle's shape is provided as a side-angle-side description.

III. CLUSTER SPACE DYNAMICS

In previous work, a kinematic model of the robots was considered under the assumption that many commercially available robotic platforms provide closed-loop velocity control. Based on this model, inverse Jacobian cluster space controllers were implemented for groups of mobile rovers [23]. In some cases, however, the kinematic model approximation may not hold true and a dynamic approach to modeling and control may be required. Examples of such situations are land rovers with non-negligible dynamics, aerial robots or marine robotic vehicles.

For the development of the cluster space equations of motion, it is assumed that the robots composing the system are holonomic and that the formation shape stays away from singularities. Cluster space singular configurations occur when the geometry of the cluster becomes degenerate and the Jacobian matrix becomes singular, as described in [32].

Next, we derive 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 [27] and [33]; we note that our presentation explicitly provides a number of critical steps in this derivation that were not provided in [27] nor, to our knowledge, in any subsequent publications. Ultimately, we verify that, even with the kinematic variations allowed by the cluster space methodology, the Jacobian transpose transforms virtual cluster space forces to physical robot space forces.

The dynamics of the system in cluster space can be represented by the Lagrangian $\mathcal{L}(c, \dot{c})$:

$$\mathcal{L}(c, \dot{c}) = T(c, \dot{c}) - U(c). \tag{6}$$

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

$$T(c, \dot{c}) = \frac{1}{2}\dot{c}^T \Lambda(c)\, \dot{c}, \tag{7}$$

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

$$p(c) = \nabla U(c). \tag{8}$$

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

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{c}}\right) - \frac{\partial \mathcal{L}}{\partial c} = F. \tag{9}$$

The equations of motion in cluster space can then be derived from (9) and written in the form

$$\Lambda(c)\, \ddot{c} + \mu(c, \dot{c}) + p(c) = F \tag{10}$$

where $\mu(c, \dot{c})$ is the vector of cluster space friction, centripetal and Coriolis forces and F is the generalized force vector in cluster space.

The equations of motion (10) 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

$$A(r)\, \ddot{r} + b(r, \dot{r}) + g(r) = \Gamma \tag{11}$$

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 [33] 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,

$$\Lambda(c) = J^{-T}(r)\, A(r)\, J^{-1}(r). \tag{12}$$

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 (9),

$$\mu(c, \dot{c}) = \dot{\Lambda}(c)\, \dot{c} - \nabla T(c, \dot{c}). \tag{13}$$

Using (12) and (4), we can get expressions for the terms of $\mu(c, \dot{c})$ in (13):

$$\begin{aligned}\dot{\Lambda}(c)\dot{c} &= \dot{J}^{-T}(r, \dot{r})A(r)J^{-1}(r)\dot{c} + J^{-T}(r)A(r)\dot{J}^{-1}(r, \dot{r})\dot{c} \\ &\quad + J^{-T}(r)\dot{A}(r)J^{-1}(r)\dot{c} \qquad (14)\\ &= J^{-T}(r)\dot{A}(r)\dot{r} + J^{-T}(r)A(r)\dot{J}^{-1}(r, \dot{r})\dot{c} \\ &\quad + \dot{J}^{-T}(r, \dot{r})A(r)\dot{r} \qquad (15)\end{aligned}$$

replacing $A(r)$ with its equivalent expression in (12), we get

$$\begin{aligned}\dot{\Lambda}(c)\dot{c} &= J^{-T}(r)\dot{A}(r)\dot{r} + \Lambda(r)J(r)\dot{J}^{-1}(r, \dot{r})\dot{c} \\ &\quad + \dot{J}^{-T}(r, \dot{r})A(r)\dot{r}\end{aligned} \tag{16}$$

using the identity $J(r)\dot{J}^{-1}(r, \dot{r}) = -\dot{J}(r, \dot{r})J^{-1}(r)$, we obtain

$$\begin{aligned}\dot{\Lambda}(c)\dot{c} &= J^{-T}(r)\dot{A}(r)\dot{r} - \Lambda(r)\dot{J}(r, \dot{r})\dot{r} \\ &\quad + \dot{J}^{-T}(r, \dot{r})A(r)\dot{r}.\end{aligned} \tag{17}$$

Regarding the second term of (13), taking the gradient of the kinetic energy with respect to the cluster space coordinates, we have

$$\nabla T(c, \dot{c}) = \frac{1}{2}\nabla\left[\, \dot{c}^T\, \Lambda(c)\, \dot{c}\,\right] \tag{18}$$

using (12), we get

$$\nabla T(c,\dot{c}) = \frac{1}{2}\nabla\big[\,\dot{r}^T A(r)\,\dot{r}\,\big] = \frac{1}{2}\nabla\big[\,\sum_i\sum_j a_{ij}(r)\dot{r}_i\dot{r}_j\,\big] \tag{19}$$

Separating the expression in two terms and given that $\delta c = J(r)\delta r$, we can express the gradient in the first term as partial derivatives with respect to the robot coordinates

$$\nabla T(c,\dot{c}) = \frac{1}{2}J^{-T}(r)\begin{pmatrix} \sum_i\sum_j \dot{r}_i\dot{r}_j\frac{\partial a_{ij}(r)}{\partial r_1} \\ \vdots \\ \sum_i\sum_j \dot{r}_i\dot{r}_j\frac{\partial a_{ij}(r)}{\partial r_{mn}} \end{pmatrix} + \sum_i\sum_j a_{ij}(r)\dot{r}_j\nabla\dot{r}_i \tag{20}$$

which can be rewritten as

$$\nabla T(c,\dot{c}) = \frac{1}{2}J^{-T}(r)\begin{pmatrix} \dot{r}^T A_{r_1}(r)\,\dot{r} \\ \vdots \\ \dot{r}^T A_{r_{mn}}(r)\,\dot{r} \end{pmatrix} + \sum_i \nabla\dot{r}_i\sum_j a_{ij}(r)\dot{r}_j \tag{21}$$

where $A_{r_i}(r)$ indicates the partial derivatives of the matrix $A(r)$ with respect to the i-th robot space variable. We can then define the resulting column vector as $l(r,\dot{r})$, obtaining

$$l_i(r,\dot{r}) = \frac{1}{2}\,\dot{r}^T A_{r_i}(r)\,\dot{r}, \qquad i = 1,2,\ldots,mn. \tag{22}$$

Therefore, we get

$$\begin{aligned}\nabla T(c,\dot{c}) &= J^{-T}(r)\,l(r,\dot{r}) + \sum_i \nabla\dot{r}_i\sum_j a_{ij}(r)\dot{r}_j \\ &= J^{-T}(r)\,l(r,\dot{r}) + \sum_i \nabla\dot{r}_i\,a_i(r)\dot{r} \\ &= J^{-T}(r)\,l(r,\dot{r}) + \nabla\dot{r}\,A(r)\dot{r}\end{aligned} \tag{23}$$

The expression $\nabla\dot{r}$ can be written as

$$\nabla\dot{r} = \begin{pmatrix} \frac{\partial^2 r_1}{\partial c_1\partial t} & \cdots & \frac{\partial^2 r_{mn}}{\partial c_1\partial t} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 r_1}{\partial c_{mn}\partial t} & \cdots & \frac{\partial^2 r_{mn}}{\partial c_{mn}\partial t} \end{pmatrix} \tag{24}$$

and assuming the second partial derivatives exist and are continuous, then by the Clairaut's theorem and using the inverse Jacobian matrix introduced in (5), we have

$$\nabla\dot{r} = \frac{\partial}{\partial t}\begin{pmatrix} \frac{\partial r_1}{\partial c_1} & \cdots & \frac{\partial r_{mn}}{\partial c_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial r_1}{\partial c_{mn}} & \cdots & \frac{\partial r_{mn}}{\partial c_{mn}} \end{pmatrix} = \dot{J}^{-T}(r,\dot{r}). \tag{25}$$

Therefore, we can express (23) as

$$\nabla T(c,\dot{c}) = J^{-T}(r)l(r,\dot{r}) + \dot{J}^{-T}(r,\dot{r})A(r)\dot{r}, \tag{26}$$

and finally from (17) and (26), we can express the terms of (13) as:

$$\begin{aligned}\dot{\Lambda}(c)\dot{c} &= J^{-T}(r)\dot{A}(r)\dot{r} - \Lambda(r)\dot{J}(r,\dot{r})\dot{r} + \dot{J}^{-T}(r,\dot{r})A(r)\dot{r} \\ \nabla T(c,\dot{c}) &= J^{-T}(r)l(r,\dot{r}) + \dot{J}^{-T}(r,\dot{r})A(r)\dot{r},\end{aligned} \tag{27}$$

where the i-th component of the vector l is

$$l_i(r,\dot{r}) = \frac{1}{2}\,\dot{r}^T A_{r_i}(r)\,\dot{r}, \qquad i = 1,2,\ldots,mn. \tag{28}$$

The subscript r_i indicates the partial derivative with respect to the i-th robot space variable. Noting from the definition of $b(r,\dot{r})$ that

$$b(r,\dot{r}) = \dot{A}(r)\,\dot{r} - l(r,\dot{r}), \tag{29}$$

then (13) can be written

$$\mu(c,\dot{c}) = J^{-T}(r)\,b(r,\dot{r}) - \Lambda(r)\,\dot{J}(r,\dot{r})\,\dot{r}. \tag{30}$$

Furthermore, in the particular case where the velocity dependent robot space forces $b(r,\dot{r})$ have the form $b(r,\dot{r}) = B(r)\dot{r}$ as in the case of viscous friction, then

$$\mu(c,\dot{c}) = \Upsilon(c)\,\dot{c} - \Lambda(c)\,\dot{J}(r,\dot{r})J^{-1}(c)\,\dot{c}, \tag{31}$$

where

$$\Upsilon(c) = J^{-T}(r)\,B(r)\,J^{-1}(r) \tag{32}$$

is the cluster space friction coefficients matrix and the second term in (31) corresponds to the cluster space Coriolis and centripetal forces.

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 (5) yields

$$p(c) = J^{-T}(r)\,g(r). \tag{33}$$

Finally, we can establish the relationship between generalized forces in cluster space and robot space, F and Γ. Using (12), (30), and (33), the cluster space equations of motion (10) can be rewritten as

$$J^{-T}(r)\big[A(r)\,\ddot{r} + b(r,\dot{r}) + g(r)\big] = F. \tag{34}$$

Substituting (11) yields

$$\Gamma = J^T(r)\,F \tag{35}$$

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.

IV. CONTROL ARCHITECTURE

The 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 (35). In order to obtain such a control vector, we use a nonlinear dynamic decoupling approach [28]. 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

$$F = \Lambda(c)\, F_m + \mu(c, \dot{c}) + p(c) \tag{36}$$

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

$$F_m = \ddot{c}_{des} + K_p\, e_c + K_v\, \dot{e}_c, \tag{37}$$

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 3 shows the control architecture of the nonlinear partition controller.

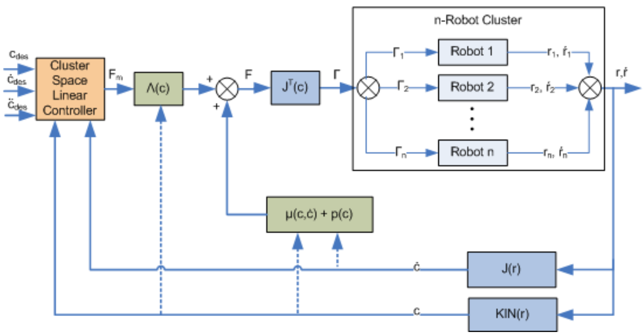

FIGURE 3. **Cluster space control architecture for a mobile *n*-robot system. Desired control forces are computed in cluster space and a partition control architecture decouples the system. The Jacobian transpose matrix converts the resulting cluster space forces to robot space forces that are then applied to the system. Robot sensor information is converted to cluster space through the Jacobian and kinematic relationships. Solid lines indicate signals and dotted lines indicate parameter passing.**

The stability of the formation utilizing the controller described by (36) and (37) can be addressed from a Lyapunov perspective. Considering the positive definite candidate Lyapunov function $V = \frac{1}{2}\dot{e}_c^T\, \dot{e}_c + \frac{1}{2}e_c^T\, K_p\, e_c$, and taking the derivative with respect to time, we obtain $\dot{V} = \dot{e}_c^T \left(\ddot{e}_c + K_p\, e_c\right)$. Using the control law (36) and the cluster dynamics (10), and replacing in the expression of $\dot{V}$, we obtain $\dot{V} = -\dot{e}_c^T\, K_v\, \dot{e}_c$, which shows that $\dot{V} \leq 0$ and the system is Lyapunov stable. □

V. EXPERIMENTS

To illustrate the functionality of the proposed formation control approach applied to systems with non-negligible dynamics, we generated computer simulations of a formation of three holonomic robots and conducted experimental tests with two different robotic testbeds: a group of three autonomous surface vessels (ASV) and a formation of four land rovers.

In order to exemplify the application of the method, let us take the planar three-robot system case. The cluster space variables must be defined and the kinematic transforms must be generated.

A. CLUSTER SPACE STATE VARIABLE DEFINITION

Figure 4 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.

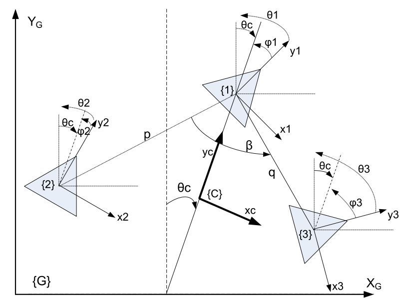

FIGURE 4. **Reference frame definition for a three-robot system placing the cluster center at the triangle centroid**

Given the parameters defined by Figure 4, the robot space pose vector is defined as:

$$r = (x_1, y_1, \theta_1, x_2, y_2, \theta_2, x_3, y_3, \theta_3)^T, \tag{38}$$

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:

$$c = (x_c, y_c, \theta_c, \phi_1, \phi_2, \phi_3, p, q, \beta)^T, \tag{39}$$

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 expressions are included in Appendix A. 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.

B. COMPUTER SIMULATIONS

A MATLAB/Simulink™ model of the control architecture was generated to validate the approach. The model allows for defining different dynamics for each robot that in turn produce complex dynamics in the cluster space. Such dynamics are intended to be canceled out by the model-based cluster controller.

The robots are modeled following (11), where for robot i:

$$A_i(r) = \begin{pmatrix} a_i & 0 & 0 \\ 0 & a_i & 0 \\ 0 & 0 & Izz_i \end{pmatrix}, \tag{40}$$

$$b_i(r,\dot{r}) = \begin{pmatrix} b_{ti}\dot{r}_{xi} \\ b_{ti}\dot{r}_{yi} \\ b_{ri}\dot{\theta}_i \end{pmatrix}, \quad g_i(r) = 0, \tag{41}$$

where a_i is the mass and Izz_i is the moment of inertia of the rotational DOF (yaw), b_{ti} and b_{ri} are the linear and rotational friction coefficients and $\dot{r}_{xi}, \dot{r}_{yi}, \dot{\theta}_i$ are the linear and angular velocities. The matrix A_i has size $m \times m$ and form the block diagonal robot space kinetic energy matrix $A(r) = diag(A_1(r), A_2(r), \ldots, A_n(r))$.

TABLE 1. Robot dynamics for simulation Cases 1 and 2.

	Robot 1	Robot 2	Robot 3
a_i (mass, kg)	10	5	1
b_{ti} (Friction coeff., kg/s)	1	5	10

Three different simulation test cases are presented in this article. In all of them, the same cluster space trajectory (position, orientation, and shape trajectories) is the input to the system and different robot dynamics or different controllers are used. In the first test case, the parameters of the robot dynamic models are defined[1] as shown in Table 1, and a linear (PID) controller with no model-based dynamic compensation generates the cluster space forces that are then translated to robot space forces to be applied to the robots.

In the second test case, the robot dynamic model parameters and initial conditions are the same as in the first test case (Table 1), but now the model-based partition controller is implemented.

TABLE 2. Robot dynamics for simulation Case 3.

	Robot 1	Robot 2	Robot 3
a_i (mass, kg)	1000	10	0.01
b_{ti} (Friction coeff., kg/s)	0.1	1000	100

For test case 3, the same model-based partition controller is used, but now the dynamic model parameters of the robots are those shown in Table 2. Different initial conditions are used in order to distinguish the output from that of test case 2.

Figure 5 shows the output of the simulation of test cases 1, 2, and 3. The test case 1 output shows how control actions applied to certain cluster variables also affect other

[1]The inertia Izz and rotational friction coefficient b_r are held constant at a unit value for all the simulation runs given that the robots are holonomic and no trajectories for the ϕ_i cluster variables are implemented.

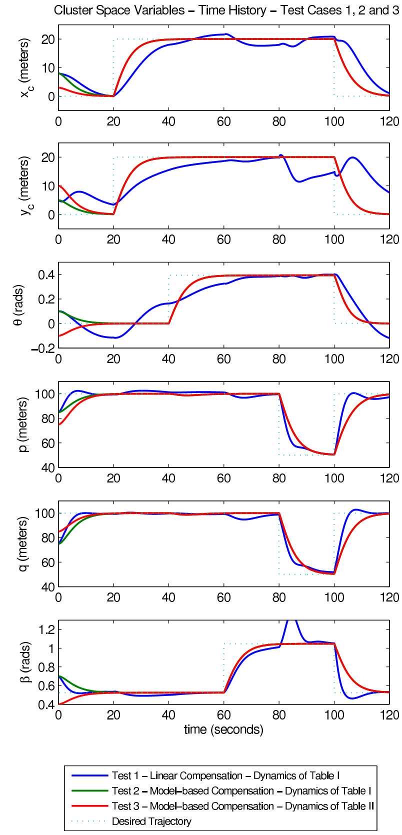

FIGURE 5. Simulation results for test cases 1, 2, and 3. Tests 1 and 2 with the system dynamics of Table 1 and test 3 with system dynamics of Table 2. Test 1 implements a PID controller with no model-based dynamic compensation. Test 2 and 3 implement a nonlinear model-based partition controller.

variables in the system, indicating coupling among them. For the test case 2, the controller cancels out the nonlinearities resulting from the cluster dynamics and the outputs behave as smooth critically damped second order decoupled systems.

Although the robot dynamics of test case 3 are considerably different from those of test case 2, the response of the system after the initial transient–due to different initial conditions–is almost identical, showing the effectiveness of the controller in canceling out the different formation dynamics resulting from the different characteristics of the particular robots in the formation.

C. FORMATION OF THREE SURFACE VESSEL MARINE ROBOTS

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 controller algorithms, and sends the appropriate compensation signals. A detailed description of this testbed can be found in [24]. Figure 6 shows two of the three 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.

FIGURE 6. Autonomous surface vehicles with propulsion systems and custom sensor and communication suites.

1) ASV DYNAMIC MODEL

The model-based partition 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 (11), the parameters for the i-th ASV are [34]:

$$A_i(r) = \begin{pmatrix} 150kg & 0 & 0 \\ 0 & 150kg & 0 \\ 0 & 0 & 41kgm^2 \end{pmatrix}, \tag{42}$$

$$b_i(r,\dot{r}) = \begin{pmatrix} 100\frac{kg}{s}\dot{r}_x \\ 400\frac{kg}{s}\dot{r}_y \\ 25\frac{kgm^2}{s\,rad}\dot{\theta} \end{pmatrix}, \quad g_i(r) = 0. \tag{43}$$

Using (12), (30), 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.

2) ASV EXPERIMENTAL RESULTS

One of the experimental tests performed is presented. The formation of ASVs moves north (y axis) keeping a triangular shape. Then one of the sides of the triangle (p) and the skew angle (β) decrease until the formation reaches a straight line configuration, later the original triangular pose is attained. An overhead view of the resulting motions is shown in Figure 7, and desired and measured values for the cluster parameters over time are shown in Figure 8.

D. FORMATION OF FOUR WHEELED LAND ROVERS

To apply the control framework to a testbed of four wheeled land rovers, a different cluster definition is used. In this case, the system has 12 DOF resulting in a cluster definition with 3 cluster parameters describing the cluster frame

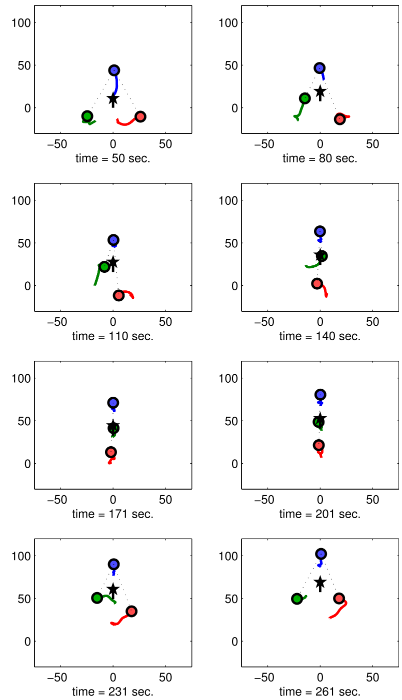

FIGURE 7. Autonomous surface vehicles experimental results. Overhead view. The circles represent the robots and the star represents the desired position of the cluster centroid. The x and y axes represent meters.

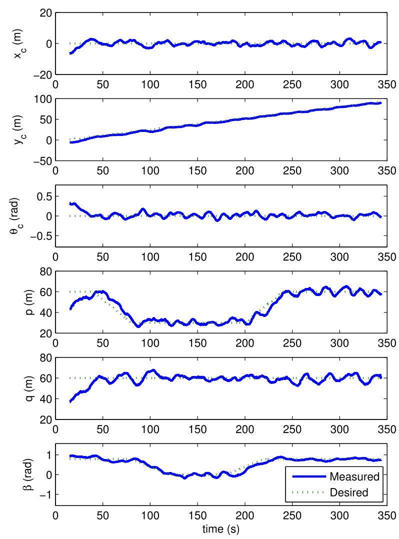

FIGURE 8. Autonomous surface vehicles experimental results. Time history of the cluster space parameters.

position and orientation, 5 cluster parameters describing the formation shape and 4 parameters for robot orientations with respect to the cluster. Figure 9 shows the selection of cluster space variables. A detailed description of the cluster definition and the kinematic transforms are reported in [20]. The rovers used for the experiments are Pioneer AT™ robots retrofitted with GPS, compass, and wireless communication systems. A remote central computer receives sensor information, executes the cluster controller algorithms, and sends the appropriate compensation signals. To accommodate for the non-holonomic constraints, a robot-level heading control

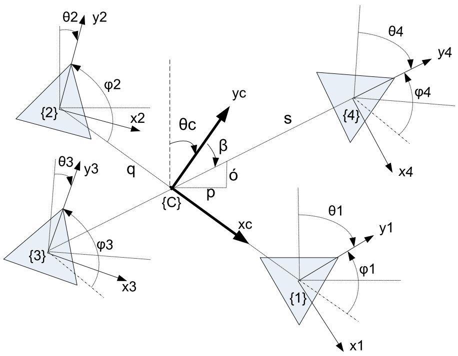

FIGURE 9. Reference frame and cluster space variables definition for a four-robot planar system.

inner-loop is implemented on each robot to achieve required bearings.

1) LAND ROVER DYNAMIC MODEL

The cluster dynamic equation is obtained through the model parameters of the land rovers. Using (11), the parameters for the i-th robot are:

$$A_i(r) = \begin{pmatrix} 20kg & 0 & 0 \\ 0 & 20kg & 0 \\ 0 & 0 & 50kgm^2 \end{pmatrix}, \tag{44}$$

$$b_i(r, \dot{r}) = \begin{pmatrix} 10\frac{kg}{s}\dot{r}_x \\ 40\frac{kg}{s}\dot{r}_y \\ 10\frac{kgm^2}{s\,rad}\dot{\theta} \end{pmatrix}, \quad g_i(r) = 0. \tag{45}$$

Again, using (12), (30), and the Jacobian matrices given by the cluster definition, the cluster dynamic parameters can be computed in execution time to be used by the controller.

2) LAND ROVER EXPERIMENTAL RESULTS

An experimental test is presented where the formation of rovers moves first in the southwest direction maintaining a square shape. Then the formation moves in the northwest direction and lastly the size of the square is augmented (parameters q and s increase). An overhead view of the resulting motions is shown in Figure 10.

In the results of both testbeds, the cluster parameters follow their desired values over time. Table 3 shows the mean squared errors for each cluster space parameter. Position sensing errors as well as errors in the estimation of the robot dynamic model parameters introduce additional tracking errors compared to those seen in the simulations. Overall, the experiments illustrate the functionality on different platforms of the nonlinear partition controlled model-based approach to cluster control of formations of mobile robots with non-negligible dynamics.

VI. MODEL PARAMETER SENSITIVITY

The model-based partition controller makes use of the dynamics of the robots composing the formation in order to generate

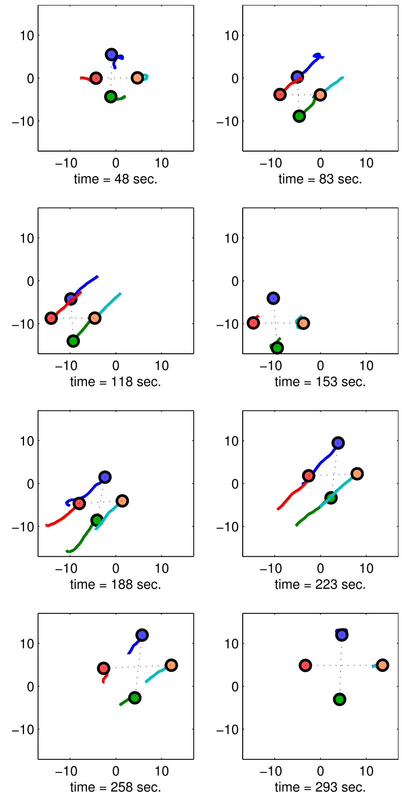

FIGURE 10. Land rovers experimental results. Overhead view. The circles represent the robots and the x and y axes represent meters.

TABLE 3. Experimental results - mean square errors of the different cluster space variables for Tests 1 and 2.

Three-ASV Test		Four-Rover Test	
Cluster Var.	MSE	Cluster Var.	MSE
x_c (m^2)	1.67434	x_c (m^2)	1.81538
y_c (m^2)	3.69479	y_c (m^2)	2.81963
θ_c (rad^2)	0.00313	θ_c (rad^2)	0.01212
p (m^2)	12.17775	$ó$ (m^2)	0.22938
q (m^2)	9.47157	p (m^2)	0.21060
β (rad^2)	0.01039	q (m^2)	0.55369
		s (m^2)	1.22900
		β (rad^2)	0.02267

commanded actions. This approach assumes that the dynamic models of the robots are known. Although the parameters of such models can be, for the most part, provided by the manufacturer or measured for each individual robot, it is interesting to analyze the response of the system to errors in the estimation of these parameters.

A set of simulations were generated where errors in the values of the dynamic model parameters were introduced. These errors are defined as percentage variations of the true parameter value. Figure 11 depicts the response of the system with errors ranging from 0% to 140% of the true model values in 20% steps. As it can be seen, a 20% variation results in a minimal deviation from the perfect model case. Depending on the requirements of the task, overshoots resulting from errors of 20% to 40% may be acceptable.

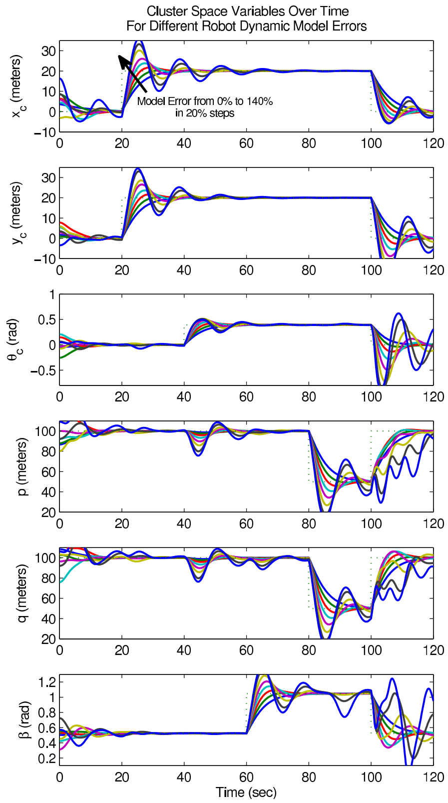

FIGURE 11. Desired trajectories and system outputs of the cluster space variables for inaccurate models of the robots in the formation.

When the errors reach 140% the formation starts showing signs of instability.

VII. CONCLUSION

The cluster space control approach for mobile robots was briefly reviewed and 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 nonlinear cluster level partition controller was proposed. The model-based portion of such a controller cancels out the cluster space nonlinear 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. Stability of the closed-loop control architecture was proven indicating that system is Lyapunov stable.

The approach was validated with computer simulations. First comparing a simple linear controller with a model-based nonlinear controller, where an improvement in performance can be readily seen in terms of errors and cluster variable coupling. Then, another comparison is made utilizing the same controller with two different sets of robots, each with substantially different dynamics. The respective responses are almost identical, illustrating the robustness of the controller to variations in the dynamics of the plant.

The proposed implementation was then applied to two experimental testbeds, one composed of three ASVs and the other with four land rovers. Results with both platforms were presented in order to demonstrate the functionality of the system. The experiments showed the ability of the formation to navigate following cluster position and shape trajectories.

Sensitivity to model parameter variations was analyzed by introducing errors in the robot dynamic parameters of the model-based controller.

Ongoing work includes the addition of obstacle avoidance methods and addressing in detail the 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 [35].

APPENDIX

A. THREE-ROBOT CLUSTER DEFINITION

Given the selection of cluster space state variables presented in Section V, the forward position kinematic relationships for the three-robot system are:

$$x_c = \frac{x_1 + x_2 + x_3}{3} \tag{46}$$

$$y_c = \frac{y_1 + y_2 + y_3}{3} \tag{47}$$

$$\theta_c = atan2\,(2x_1 - x_2 - x_3, 2y_1 - y_2 - y_3) \tag{48}$$

$$\phi_1 = \theta_1 - \theta_c \tag{49}$$

$$\phi_2 = \theta_2 - \theta_c \tag{50}$$

$$\phi_3 = \theta_3 - \theta_c \tag{51}$$

$$p = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{52}$$

$$q = \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2} \tag{53}$$

$$\begin{aligned}\beta = atan2\big(&(x_3 - x_1)sin(\alpha) + (y_3 - y_1)cos(\alpha), \\ &(x_3 - x_1)cos(\alpha) - (y_3 - y_1)sin(\alpha)\big),\end{aligned} \tag{54}$$

where

$$\alpha = atan2\big(y_1 - y_2, x_2 - x_1\big), \tag{55}$$

and $atan2(y, x)$ is the 4-quadrant arctangent [28]. The inverse position kinematics are therefore defined by:

$$x_1 = x_c + \frac{1}{3}\sqrt{\kappa}\,\sin(\theta_c) \tag{56}$$

$$y_1 = y_c + \frac{1}{3}\sqrt{\kappa}\,\cos(\theta_c) \tag{57}$$

$$\theta_1 = \phi_1 + \theta_c \tag{58}$$

$$x_2 = x_c + \frac{1}{3}\sqrt{\kappa}\sin(\theta_c) + p\cos(\gamma) \quad (59)$$

$$y_2 = y_c + \frac{1}{3}\sqrt{\kappa}\cos(\theta_c) + p\sin(\gamma) \quad (60)$$

$$\theta_2 = \phi_2 + \theta_c \quad (61)$$

$$x_3 = x_c + \frac{1}{3}\sqrt{\kappa}\sin(\theta_c) + q\cos(\beta+\gamma) \quad (62)$$

$$y_3 = y_c + \frac{1}{3}\sqrt{\kappa}\cos(\theta_c) + q\sin(\beta+\gamma) \quad (63)$$

$$\theta_3 = \phi_3 + \theta_c, \quad (64)$$

where

$$\kappa = p^2 + q^2 + 2pq\cos(\beta), \quad (65)$$

and

$$\gamma = atan2\big(q\sin(\beta), p + q\cos(\beta)\big) + atan2\big(\cos(\theta_c), -\sin(\theta_c)\big). \quad (66)$$

ACKNOWLEDGMENTS

The authors gratefully thank Thomas Adamek, Ketan Rasal, Jasmine Cashbaugh and Paul Mahacek, for developing and managing the ASV and land rover experimental testbeds, and Michael Neumann for his thoughtful comments. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation, NOAA, or Santa Clara University.

REFERENCES

[1] C. Kitts and M. Egerstedt, "Design, control, and applications of real-world multirobot systems [from the guest editors]," *IEEE Robot. Autom. Mag.*, vol. 15, no. 1, p. 8, Mar. 2008.

[2] V. Kumar, N. E. Leonard, and A. S. Morse, *Cooperative Control: A Post-Workshop Volume, 2003 Block Island Workshop on Cooperative Control*. New York, NY, USA: Springer-Verlag, 2005.

[3] G. Mariottini *et al.*, "Vision-based localization for leader-follower formation control," *IEEE Trans. Robot.*, vol. 25, no. 6, pp. 1431–1438, Dec. 2009.

[4] B. Smith, A. Howard, J.-M. McNew, J. Wang, and M. Egerstedt, "Multi-robot deployment and coordination with embedded graph grammars," *Auton. Robots*, vol. 26, no. 1, pp. 79–98, Jan. 2009.

[5] A. K. Das, R. Fierro, V. Kumar, J. P. 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.

[6] P. Song and V. Kumar, "A potential field based approach to multi-robot manipulation," in *Proc. IEEE ICRA*, vol. 2. May 2002, pp. 1217–1222.

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

[8] F. Schneider and D. Wildermuth, "A potential field based approach to multi robot formation navigation," in *Proc. IEEE Int. Conf. Robot., Intell. Syst. Signal Process.*, vol. 1. Oct. 2003, pp. 680–685.

[9] K.-H. Tan and M. A. Lewis, "Virtual structures for high-precision cooperative mobile robotic control," in *Proc. IEEE/RSJ IROS*, vol. 1. Nov. 1996, pp. 132–139.

[10] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proc. 40th IEEE Conf. Decision Control*, vol. 3. Dec. 2001, pp. 2968–2973.

[11] P. McDowell, J. Chen, and B. Bourgeois, "UUV teams, control from a biological perspective," in *Proc. MTS/IEEE OCEANS*, vol. 1. Oct. 2002, pp. 331–337.

[12] S. Kalantar and U. Zimmer, "Control of open contour formations of autonomous underwater vehicles," *Int. J. Adv. Robot. Syst.*, vol. 2, pp. 309–316, Dec. 2005.

[13] A. Girard and J. Hedrick, "Formation control of multiple vehicles using dynamic surface control and hybrid systems," *Int. J. Control*, vol. 76, no. 9, pp. 913–923, Jun. 2003.

[14] I.-A. F. Ihle, R. Skjetne, and T. Fossen, "Nonlinear formation control of marine craft with experimental results," in *Proc. 43rd IEEE CDC*, vol. 1. Dec. 2004, pp. 680–685.

[15] F. Fahimi, "Non-linear model predictive formation control for groups of autonomous surface vessels," *Int. J. Control*, vol. 80, pp. 1248–1259, Aug. 2007.

[16] R. Fierro *et al.*, "A framework and architecture for multirobot coordination," *Int. J. Robot. Res.*, vol. 21, nos. 10–11, pp. 977–995, Oct./Nov. 2002.

[17] C. Belta and V. Kumar, "Abstraction and control for groups of robots," *IEEE Trans. Robot.*, vol. 20, no. 5, pp. 865–875, Oct. 2004.

[18] R. A. Freeman, P. Yang, and K. M. Lynch, "Distributed estimation and control of swarm formation statistics," in *Proc. Amer. Control Conf.*, Jun. 2006, pp. 749–755.

[19] P. Mahacek, C. Kitts, and I. Mas, "Dynamic guarding of marine assets through cluster control of automated surface vessel fleets," *IEEE/ASME Trans. Mechatronics*, vol. 17, no. 1, pp. 65–75, Feb. 2012.

[20] I. Mas and C. Kitts, "Object manipulation using cooperative mobile multi-robot systems," in *Proc. World Congr. Eng. Comput. Sci.*, Oct. 2012, pp. 324–329.

[21] C. A. Kitts and I. Mas, "Cluster space specification and control of mobile multirobot systems," *IEEE/ASME Trans. Mechatronics*, vol. 14, no. 2, pp. 207–218, Apr. 2009.

[22] C. Kitts, K. Stanhouse, and P. Chindaphorn, "Cluster space collision avoidance for mobile two-robot systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 1941–1948.

[23] I. Mas, O. Petrovic, and C. Kitts, "Cluster space specification and control of a 3-robot mobile system," in *Proc. IEEE ICRA*, May 2008, pp. 3763–3768.

[24] P. Mahacek, I. Mas, O. Petrovic, J. Acain, and C. Kitts, "Cluster space control of autonomous surface vessels," *Marine Technol. Soc. J.*, vol. 43, no. 1, pp. 13–20, 2009.

[25] M. S. Agnew, P. Dal Canto, C. A. Kitts, and S. Li, "Cluster space control of aerial robots," in *Proc. IEEE/ASME Int. Conf. AIM*, Jul. 2010, pp. 1305–1310.

[26] I. Mas and C. Kitts, "Centralized and decentralized multi-robot control methods using the cluster space control framework," in *Proc. IEEE/ASME Int. Conf. AIM*, Jul. 2010, pp. 115–122.

[27] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. Robot. Autom.*, vol. 3, no. 1, pp. 43–53, Feb. 1987.

[28] J. Craig, *Introduction to Robotics, Mechanics and Control*, 3rd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2005.

[29] L. Sentis, J. Park, and O. Khatib, "Compliant control of multicontact and center-of-mass behaviors in humanoid robots," *IEEE Trans. Robot.*, vol. 26, no. 3, pp. 483–501, Jun. 2010.

[30] F. Huizenga, "Singularity avoidance for the cluster space control of mobile multi-robot systems," M.S. thesis, Dept. Mech. Eng., Santa Clara Univ., Santa Clara, CA, USA, May 2012.

[31] I. Mas, C. Kitts, and R. Lee, "Model based nonlinear cluster space control of mobile robot formations," in *Multi Robot Systems, Trends and Development*, T. Yasuda, Ed. Hampshire, U.K.: InTech, 2011, pp. 53–70.

[32] I. Mas, J. Acain, O. Petrovic, and C. Kitts, "Error characterization in the vicinity of singularities in multi-robot cluster space control," in *Proc. IEEE Int. Conf. Robot. Biomimet.*, Bangkok, Thailand, Feb. 2009, pp. 1911–1917.

[33] O. Khatib, "Commande dynamique dans l'espace opérationnel des robots manipulateurs en présence d'obstacles," Ph.D. dissertation, Ecole Nationale Supérieure de l'aéronautique et de l'espace, Toulouse, France, 1980.

[34] P. Mahacek, "Design and cluster space control of two autonomous surface vessels," M.S. thesis, Dept. Mech. Eng., Santa Clara Univ., Santa Clara, CA, USA, Dec. 2009.

[35] G. Okamoto, C. Chen, and C. Kitts, "Beamforming performance for a reconfigurable sparse array smart antenna system via multiple mobile robotic systems," *Proc. SPIE*, vol. 7706, pp. 77060Z-1–77060Z-11, Apr. 2010.

IGNACIO MAS (S'06–M'12) received the Engineering degree in electrical engineering from the Universidad de Buenos Aires, Buenos Aires, Argentina, and the Ph.D. degree in mechanical engineering from Santa Clara University, Santa Clara, CA, USA.

He was a Satellite Systems Engineer and Spacecraft Systems Designer at the NASA Ames Research Center, Mountain View, CA, USA. He is currently a CONICET Assistant Researcher with the Instituto Tecnologico de Buenos Aires, Buenos Aires. His research interests include multirobot systems, coordinated navigation, and formation control.

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

He was a Research Engineer and an Operational Satellite Constellation Mission Controller. He was an Officer with the U.S. Air Force Space Command, Colorado Springs, CO, a NASA Contractor with the Department of Defense Research Fellow, Caelum Research Corporation, Rockville, MD, USA, and the Graduate Student Director of the Space Systems Development Laboratory at Stanford University. He is currently an Associate Professor at Santa Clara University, Santa Clara, CA, USA, where he is also the Director of the Robotic Systems Laboratory.