A Hybrid Multirobot Control Architecture for Object Transport

Michael A. Neumann and Christopher A. Kitts, *Senior Member, IEEE*

***Abstract*—A hybrid force/position control architecture for object transportation by a mobile robot formation is introduced. The architecture consists of an object level controller that computes actuation forces/torques to push an object along a desired path. These desired forces/torques are provided to a multirobot actuation subsystem that uses a closed-loop hybrid force/position controller to generate commands for individual robots within the subsystem. Accordingly, these robots apply the required net force and torque to the object while also maintaining relative positions with respect to themselves and to the object, thereby ensuring stable motion and efficient torque generation. This novel use of a hybrid controller for the mobile multirobot cluster prevents large environmental forces on the object. Furthermore, abstraction of the cluster as an actuator enhances flexibility, making design of the object controller independent of the number, location, and maneuverability of individual robots. The architecture is validated experimentally.**

***Index Terms*—Cluster space control, mobile robots, multirobot systems, object transportation.**

I. Introduction

Multirobot cooperation is a promising field with potential benefits ranging from providing redundancy to enabling reconfigurable, distributed sensing at scales that would be impractical or prohibitively expensive with a single unit [1]. One area of research within this field is object transportation, where a group of robots moves an object that is too large or heavy for a single robot.

Most multirobot object transportation research has focused on terrestrial applications, although examples exist as well for aerial [2], [3] and marine environments [4]–[6]. There are three primary strategies for multirobot object transportation: form (or force) closure, where robots prevent object motion in all directions; conditional closure, where external forces (like gravity or friction) prevent movement in directions that are not constrained by the robots; and object closure, where the robots surround the object with inter-robot distances that guarantee that the object remains inside the mobile group of robots [7]. Manipulators, static and mobile, typically move objects using form closure [8]–[10]. Small robot groups, usually pairs, often utilize conditional closure [11]–[13]. Object closure is used mainly by decentralized multirobot control techniques [14], [15], though centralized techniques have used it as well [16].

Force feedback has been used in manipulator systems when transporting objects [17]. In contrast, formations of mobile robots without grasping mechanisms typically have controllers that solely control the positions and/or velocities of the robots and object, though force feedback has been utilized in some instances [18]. The use of force control and force feedback has several benefits such as preventing large forces that could damage the object or the robots themselves.

Our previous work in multirobot object transportation divided the objective of pushing a box along a desired path into two tasks: 1) applying forces to the box to move it along the path; and 2) maintaining the robot formation's desired size and location relative to the box [19]. In this paper, we unify these tasks by controlling the mobile multirobot cluster with a hybrid force/position control architecture, a strategy that integrates several simple controllers and which has previously been employed for controlling a single static manipulator [20]. The benefits of this hybrid multirobot controller compared with other works are the explicit use of force control, which ensures system safety and enables dynamic object control; the integration of robot force and position control, promoting features such as the simple reassignment of either to different dimensions of the multirobot system (a feature that will be particularly attractive for pushing jointed or flexible objects); and the abstraction of the multirobot system as a single closed loop actuator subsystem within an overall object control system that focuses on the desired motion of the transported object. This abstraction decouples decisions regarding desired object control forces and torques from actuator details such as the number of robots being used to transport the object, the relative locations of the robots, etc.

In this paper, we present our abstracted and unified control architecture. This begins with a simple object controller focused on path keeping, an actuation subsystem implemented through the use of multiple robots described by a "cluster space" state representation, and a hybrid position/force inner loop actuation control law that computes velocity commands for individual robots. We then describe a simple experimental testbed used to evaluate this technique. Finally, we present experimental results for two robots using this strategy in order to move an object along a prescribed path.

II. Object Controller

Fig. 1 depicts the scenario of interest. We wish to move a large object along a desired path. The position of the object is denoted by frame $\{\mathrm{O}\}$, and the path is defined by the function $p = f(x, y)$. For ideal path control, the object moves along the desired path such that the object frame's origin coincides with an instantaneous path frame $\{\mathrm{P}\}$ that has its $\hat{x}_P$ unit vector aligned tangentially to the path in the direction of motion; the object is allowed to move along the path at an arbitrary open loop rate

Manuscript received September 3, 2015; revised February 1, 2016; accepted April 21, 2016. Date of publication June 14, 2016; date of current version December 13, 2016. Recommended by Technical Editor J. Wang. This work was supported in part by the Santa Clara University Robotic Systems Laboratory and the School of Engineering, and in part by the National Science Foundation under Grant CNS-0619940.

The authors are with the Department of Mechanical Engineering, Santa Clara University, Santa Clara, CA 95053 USA (e-mail: maneumann@scu.edu; ckitts@scu.edu).

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

Digital Object Identifier 10.1109/TMECH.2016.2580539

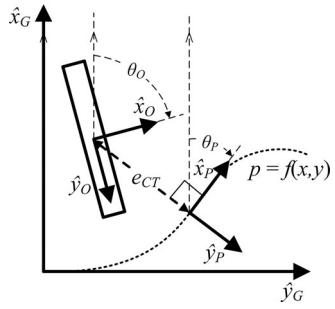

Fig. 1. Object's location is defined by a frame {O} located at its center. The instantaneous frame {P} has its origin at the point of the path closest to {O} with the $\hat{x}_P$ unit vector tangent to the path in the direction of desired motion. The distance between the origins of {O} and {P} is defined as the cross-track error, (e_{CT}).

with no explicit timing requirements such as is typical in full trajectory control. In practice, the object may be offset from the path by a cross-track error (e_{CT})

The object path controller is the outermost loop in the proposed architecture, shown in Fig. 2. The controller calculates the torque and force to apply to the object using the path control algorithm described in [21]. For this initial paper, the desired object control force (F_{obj}) along $\hat{x}_O$ is set to a constant at a value of the average static friction force between the object and the ground. An object control torque (T_{obj}) is used to turn the object towards the desired path if a cross-track error exists. We calculate this torque using a proportional law based on the error between the object's heading (θ_O) and the desired object heading (θ_D), as shown in

$$T_{\mathrm{obj}} = k_H(\theta_D - \theta_O). \tag{1}$$

The desired object heading has a default value of the local path heading (θ_P) with an offset angle that steers the object back to the path, as shown in (2). This offset angle is proportional to the perpendicular distance from the path (e_{CT}) although it is limited such that it can never be more than $\pm 90°$, which is equivalent to moving directly toward the path. The gain (k_{CT}) dictates how aggressively the object is turned in order to steer it back onto the path

$$\theta_D = \theta_P - \max(\min(k_{\mathrm{CT}} e_{\mathrm{CT}}, \pi/2), -\pi/2). \tag{2}$$

The desired object control force F_{obj} and torque T_{obj} are combined into a desired actuation vector τ consisting of the force in the $\hat{x}_O$ direction, F_x; the force in the $\hat{y}_O$ direction, F_y; and the torque around $\hat{z}_O$, T_z

$$\tau = (F_x, F_y, T_z)^T = (F_{\mathrm{obj}}, 0, T_{\mathrm{obj}})^T. \tag{3}$$

III. Cluster Space MultiRobot Control

The desired actuation vector in (3) is provided to the actuator system for implementation. In this paper, implementation is performed by a multirobot formation, and the control of that formation leverages our previous work in cluster space control.

Mobile robots are typically controlled using position state variables based on their individual location and orientation, which we refer to as robot space pose variables. An example of a position vector containing the robot space variables for two planar vehicles is

$$\mathbf{r} = (x_1, y_1, \theta_1, x_2, y_2, \theta_2)^T \tag{4}$$

where (x_i, y_i) is the position of robot i in the plane, and (θ_i) is the orientation of robot i.

Alternatively, the position of a mobile multirobot system can be described using state variables representing the location and geometry of the robot group. This is the approach adopted by the cluster space control technique, an operational space inverse Jacobian/Jacobian transpose control architecture described in detail in [22] and which has been used in a wide variety of applications [23], [24]. Cluster space control uses a full degree-of-freedom state representation and treats the mobile multirobot group as a virtual articulating mechanism. The geometric nature of this approach is particularly relevant for object transport given the alignment and relative position constraints inherent between the robots and the object.

For the simple two-robot cluster used in this research, we assign a cluster frame at the center of the two robots and align the cluster frame's $\hat{y}_C$ unit vector such that it points toward robot 1 as shown in Fig. 3. The cluster position (x_c, y_c, θ_c) is defined as the location and orientation of the cluster frame with respect to the object frame, which is signified by the superscript O preceding the variables in the figure. The size of the cluster d is the distance from the cluster frame origin to each robot. Finally, the relative orientations of each robot with respect to the cluster frame are given by (φ_1, φ_2). The complete cluster pose vector is provided in

$$\mathbf{c} = (x_c, y_c, \theta_c, d, \varphi_1, \varphi_2). \tag{5}$$

As described in [22], the robot space and cluster space pose vectors are related to each other through kinematic functions, as shown in (6) and (7). The elements of a Jacobian $\mathbf{J}$ are defined by the partial derivatives of the kinematic equations with respect to the robot variables, as shown in (8). Furthermore, the robot space and cluster space velocities are related through Jacobian transforms as shown in (9) and (10)

$$\mathbf{c} = \mathrm{invkin}(\mathbf{r}) = [g_1(r_1, \ldots, r_6) \ \ldots \ g_6(r_1, \ldots, r_6)]^T \tag{6}$$

$$\mathbf{r} = \mathrm{invkin}(\mathbf{c}) = [h_1(c_1, \ldots, c_6) \ \ldots \ h_6(c_1, \ldots, c_6)]^T \tag{7}$$

$$J_{i,j} = \partial g_i(\mathbf{r})/\partial r_j \tag{8}$$

$$\dot{\mathbf{c}} = \mathbf{J}\dot{\mathbf{r}} \tag{9}$$

$$\dot{\mathbf{r}} = \mathbf{J}^{-1}\dot{\mathbf{c}}. \tag{10}$$

These relationships allow robot cluster motion to be specified in and control commands to be computed in the cluster space, promoting ease of use, abstraction, and well-behaved cluster space behavior [22]. The kinematic transforms are then used to convert cluster control commands into individual robot-specific commands for implementation.

Similar to robot and cluster space pose vectors, we can also define robot and cluster space force vectors as described in [25]. The robot space force vector $\mathbf{f}$ consists of forces and torques

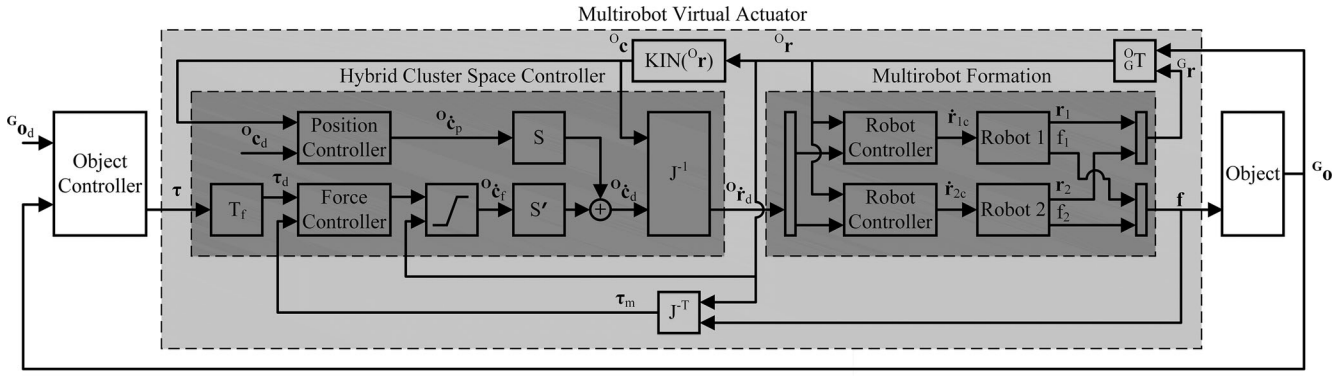

Fig. 2. Control architecture consists of a series of controllers. The outermost controller is the object controller, which calculates the desired force and torque to apply based on the location of the box relative to the desired path. The hybrid controller generates robot space commands based on the desired cluster space variables, application force, and torque. Robot level controllers are used to transform the robot space commands that assume holonomic vehicles into commands that can be accepted by the mobile robots.

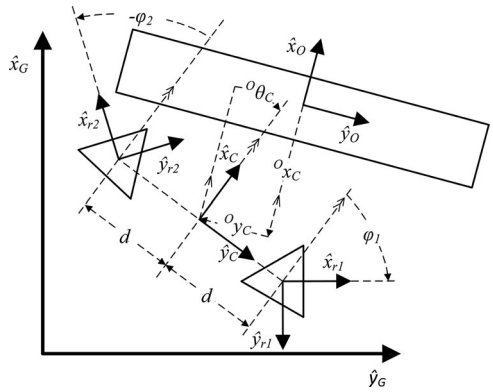

Fig. 3. Cluster space variables, x_c, y_c, θ_c, d, φ_1, and φ_2, define the shape of the formation and its position and orientation relative to the reference frame fixed in the box.

corresponding to the corresponding elements of the robot space position vector

$$\mathbf{f} = (f_{x1}, f_{y1}, T_1, f_{x2}, f_{y2}, T_2)^T. \tag{11}$$

The cluster space force vector for our two robot formation is

$$\tau = (f_x, f_y, T_c, f_d, \Gamma_1, \Gamma_2)^T \tag{12}$$

where the variables are related to the corresponding elements of the cluster space position vector: (f_x, f_y) are the forces applied by the cluster in the $\hat{x}_C$ and $\hat{y}_C$ directions; T_c is the torque about the cluster origin, f_d is the inline force between robots, and (Γ_1, Γ_2) are the torques from each individual robot. As with the robot space and cluster space velocities, the robot space and cluster space force vectors are related through the Jacobian transforms shown in (13) and (14) [25]

$$\mathbf{f} = \mathbf{J}^T \tau \tag{13}$$

$$\tau = \left(\mathbf{J}^T\right)^{-1} \mathbf{f}. \tag{14}$$

IV. Hybrid Position/Force Cluster Control

For multirobot object transport, we want to control contact forces between the robots and the object while simultaneously controlling the positions of the robots relative to each other and the object in order to provide stable actuation and efficient torque generation. To achieve this, we adopt a hybrid controller that implements force control in a direction perpendicular to the face of the object and position control within the plane of the object's surface. Each of these controllers is implemented in the cluster state space given its level of abstraction and convenient geometric representation. Each controller also uses a resolved rate control strategy given that the robots are commanded through velocity set points.

Accordingly, the object controller's actuation commands, provided in (3), are used to specify the f_x and T_c elements of the desired cluster space force vector, provided in (12), given that these elements result in forces on the object

$$\tau_d = (F_{\text{obj}}, 0, T_{\text{obj}}, 0, 0, 0)^T. \tag{15}$$

The force controller is also provided the actual cluster torque vector, which is generated by applying the forward kinematic transforms, shown in (14), to the measured robot-space forces, which are provided by force sensors. For the experimental results presented in this paper, the force controller consisted of the following resolved rate control law:

$$\dot{\mathbf{c}}_f = (\tau_d - \tau_m)\, \mathbf{k}_p + (\tau_d)\, \mathbf{k}_{ff} \tag{16}$$

where $\dot{\mathbf{c}}_f$ is the force controller's cluster velocity command, $\mathbf{k}_p$ is a vector of proportional gains, $\mathbf{k}_{ff}$ is a vector of feed-forward gains, τ_d is the desired cluster space force vector defined in (15), and τ_m is the measured cluster space force vector. The rotational command in $\dot{\mathbf{c}}_f$, $\dot{\theta}_c$, generated by the controller is limited by a command conditioner to prevent the robots from disengaging from the object.

In parallel to the force controller, a simple proportional controller was used for cluster space position control

$$\dot{\mathbf{c}}_p = (\mathbf{c}_d - \mathbf{c})\, \mathbf{k}_p \tag{17}$$

where $\dot{\mathbf{c}}_p$ is the position controller's cluster velocity command, $\mathbf{c}_d$ is the desired cluster space pose, $\mathbf{c}$ is the measured cluster space pose, and $\mathbf{k}_p$ is a vector of proportional gains.

A departure from the hybrid force/position architecture used with manipulators is the use of velocity-based compensation commands, $\dot{\mathbf{c}}_f$ and $\dot{\mathbf{c}}_p$, instead of forces and torques. This was

motivated by the fact that the mobile robots available for testing accept forward and rotational velocity commands. Ideally, a dynamic controller would have been used; this will be explored in future work.

The outputs of the two controllers are combined through a pair of selection matrices, $\mathbf{S}$ and $\mathbf{S}'$. $\mathbf{S}$, a diagonal matrix with all elements set to 0 or 1, is used to select the outputs from the position controller to pass into the command vector. $\mathbf{S}'$ passes outputs from the force controller to the command vector and is a related to $\mathbf{S}$, with complementary diagonal elements such that

$$\mathbf{S}' = \mathbf{I} - \mathbf{S}. \tag{18}$$

The sum of the selection matrices' outputs is a vector of desired velocities in cluster space, ${}^{o}\dot{\mathbf{c}}_d$. This desired cluster space velocity vector is then multiplied by an inverse Jacobian matrix, which transforms the cluster space velocity vector into desired robot space velocities, as described in (10).

The desired robot space command vector is split into commands for each individual robot. However, these commands contain lateral velocities that are not achievable by the nonholonomic testbed vehicles. To compensate, the commanded velocity bearing is converted to a set point for a robot-level heading control loop such that the robot turns toward the desired direction of motion. In parallel, the commanded forward velocity is limited to its projection in the direction of the velocity bearing; this leads to full speed when the robot is headed in the right direction but a proportionally reduced speed when a turn is required. Equations for the individual robot level controllers are available in [26].

V. Testbed

To validate the control architecture, a series of experiments were undertaken to push a box along a specified path. Two Pioneer AT^{TM} robots were outfitted with a hinged force sensor, allowing them to apply forces perpendicular to the face of the box. The robots were also equipped with wireless modems for sending force measurements to an off-board controller and for receiving velocity commands. An UltraWideBand tracking system was used to track position of the robots and the box; this system provides position measurements that approximate a normal distribution with a variance of 0.04 m [27]. The mobile robot with modifications can be seen in Fig. 4. A component block diagram of the experimental system is available in [19].

The testbed provided several challenges for verifying the proposed control architecture. First, the forces associated with moving a box across a surface are dominated by Coulomb friction, so the interaction forces varied less than desired. As an option for future indoor tests, we plan to place the box on casters in order to vary this characteristic. Second, the robots receive velocity commands implemented by their own on-board controllers to achieve desired wheel speeds. This use of a resolved rate control strategy makes it challenging to try to control interaction forces; in the future, we plan to use a model-based dynamic controller in order to improve performance [25]. Third, latency associated with the off-board computing of cluster commands via a wireless link required us to operate at slow speeds; in the future, a more robust communication system and the migration of some control functions to on-board the robots will allow operation at higher velocities. Finally, the experimental workspace was limited due to the coverage area of the UltraWideBand tracking system. In future work, we will apply this technique to a larger scale and more dynamic marine testbed using multiple boats.

Fig. 4. Experiment utilized Pioneer robots with custom communication and force sensing equipment. One of the two robots used is shown above.

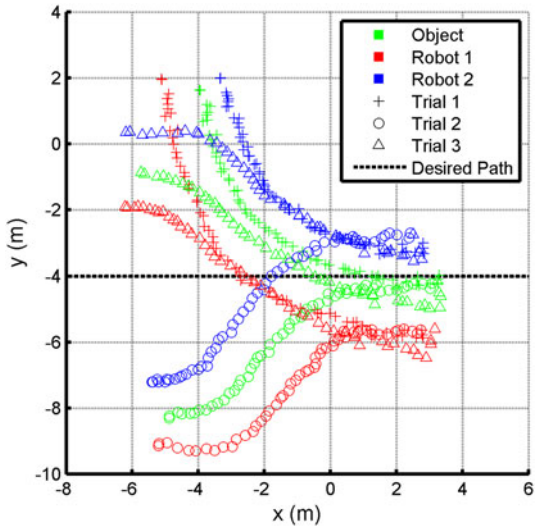

Fig. 5. Three trial runs performed with the hybrid position/force architecture. All runs start on the left side of the plot and converge to the desired path shown by the dashed horizontal line.

VI. Experiments

Experimentation consisted of a series of trials, where the box was given a random offset from a desired straight line path. The robots were placed so that their force plates were nearly in contact with the side of the box prior to the start of each trial. In all three cases, the robots repositioned the box to the desired path and then tracked it with a relatively small offset, as can be seen in Fig. 5. Ideally, the robots would have pushed the box further along the line, but the workspace was limited.

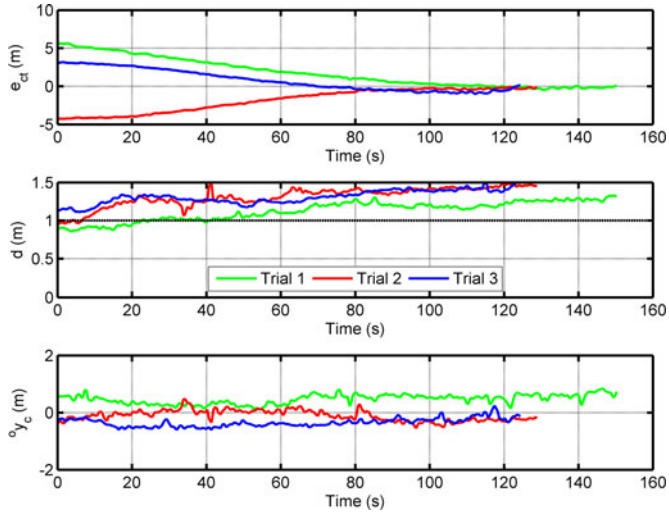

Fig. 6. Time histories of the cross-track error (e_{CT}), inter-robot distance (d), and lateral offset between the cluster center and the middle of the box $(^{O}y_C)$ from the three trials show that the formation is able to push the object to the appropriate path while maintaining the desired cluster pose.

In the first trial, which started perpendicular to the desired path, the average cross-track error (e_{CT}) after the box reached the line was -0.17 m with an root mean square (RMS) error of 0.21 m. The formation also maintained a 1 m inter-robot distance (d) set point with a steady-state offset of 0.13 m and an root mean square (RMS) error of 0.17 m. The lateral offset between the cluster center and the center of the box (y_c) stayed fairly constant with a steady-state offset of 0.47 m. The time histories for e_{CT}, d, and $^{O}y_c$ for all three trials are shown in Fig. 6.

The box converged to the path in the last two trials as well, though there was some slight overshoot during the third trial. While it appears that d is steadily increasing in the first trial, trials 2 and 3 show that the variable remains bounded within 0.5 m of the desired value. Offsets in the cluster space parameters, which were larger for d than $^{o}y_c$, are not surprising given the use of a proportional controller and were deemed acceptable. We note that an integral control term could be used to improve d performance in the future. Furthermore, we note that the object-level controller computes its torque commands based on the real-time value of d thereby preventing such deviations from directly impacting object-level control.

Due to the nature of the testbed and force sensors, the cluster force variables were not controlled smoothly to the desired values. The force measurements were noisy, and even when filtered there were steady-state offsets due to calibration issues. However, the control architecture overcame these challenges and passed appropriate commands to the robots in order to achieve the desired goal of path following. Fig. 7 shows the forward velocity commands to the two robots from the first trial. As expected, robot 1 (the right robot) is initially given a larger velocity command, effectively pushing harder in order to turn the object to the left toward the direction of the desired path. As time passes and the object nears the desired path, the difference between the two commands decreases. Finally, when the object is being pushed along the path, the two commands are effectively the same.

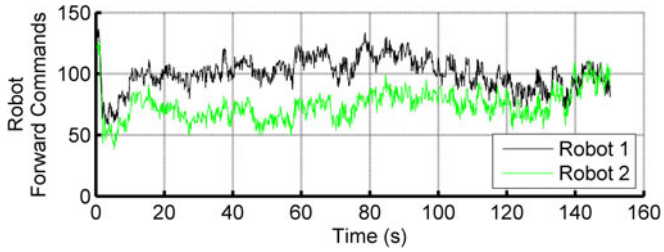

Fig. 7. Robot commands over time in trial 1 show that the force controller is effective for the object transportation application. Initially robot 1 is commanded at a higher velocity, in order to turn the object towards the desired path. As the object aligns with the path the difference between the commands for the two robots decreases.

VII. Conclusion

In this paper, we have integrated several conventional control approaches within a hybrid force/position control architecture in a novel way, using it control a group of mobile robots in order to transport a large object along a desired path. Use of the force control element of this architecture prevents the application of potentially damaging interaction or environmental forces and improves actuation effectiveness. Use of the position control function ensures that the robots are positioned in a configuration for stable pushing and efficient torque generation with respect to the object, preventing the robots from slipping off the end of the box and positioning them so that adequate torques can be applied in either direction. The hybrid architecture is implemented as the inner loop of the object control system, promoting flexibility by making the design of the object controller independent of the number, location or type of robots used. An initial set of experimental demonstrations show that the architecture is effective at safely transporting a large object along a desired path.

Moving forward, we plan to extend control architecture functionality in several ways. First, we are working on an optimization function to determine robot position set points to ensure stability and efficient torque generation for a different numbers of robots and for more complex objects (e.g., angled objects, objects with hinged joints, etc.). We are also integrating a state machine to manage the process of having the robot formation approach, transport, and disengage from an object. We also plan to apply the control architecture to more challenging testbeds, such as an existing multiboat system operating in a more dynamic marine environment.

References

[1] M.-Y. Chow, S. Chiaverini, and C. Kitts, "Guest editorial introduction to the focused section on mechatronics in multirobot systems," *IEEE/ASME Trans. Mechatronics*, vol. 14, no. 2, pp. 133–140, Apr. 2009.

[2] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," *Auton. Robots* vol. 30, no. 1, pp. 73–86. Sep. 2010. [Online]. Available: http://link.springer.com/article/10.1007/s10514–010–9205–0/fulltext.html

[3] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, "Cooperative grasping and transport using multiple quadrotors," in *Proc. 10th Int. Symp. Distrib. Auton. Robot. Syst.*, Lusanne, Switzerland, 2010, pp. 545–558.
[4] J. M. Esposito, M. G. Feemster, and E. Smith, "Cooperative manipulation on the water using a swarm of autonomous tugboats," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 1501–1506.
[5] M. G. Feemster and J. M. Esposito, "Comprehensive framework for tracking control and thrust allocation for a highly overactuated autonomous surface vessel," *J. Field Robot.*, vol. 28, no. 1, pp. 80–100, Jan./Feb. 2011.
[6] Y. Hu, L. Wang, J. Liang, and T. Wang, "Cooperative box-pushing with multiple autonomous robotic fish in underwater environment," *IET Control Theory Appl.*, vol. 5, no. 17, pp. 2015–2022, Nov. 2011.
[7] G. A. S. Pereira, M. F. M Campos, and V. Kumar, "Decentralized algorithms for multi-robot manipulation via caging," *The Int. J. Robotics Res.*, vol. 23, no. 7–8, pp. 783–795, Aug. 2004.
[8] O. Khatib *et al.*, "Vehicle/arm coordination and multiple mobile manipulator decentralized cooperation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 1996, pp. 546–553.
[9] C. P. Tang, R. M. Bhatt, M. Abou-Samah, and V. Krovi, "A screw-theoretic analysis framework for payload transport by mobile manipulator collectives," *IEEE/ASME Trans. Mechatronics*, vol. 11, no. 2, pp. 169–178, Apr. 2006.
[10] D. Sun and J. K. Mills, "Manipulating rigid payloads with multiple robots using compliant grippers," *IEEE/ASME Trans. Mechatronics*, vol. 7, no. 1, pp. 23–34, Mar. 2002.
[11] M. J. Mataric, M. Nilsson, and K. T. Simsarin, "Cooperative multi-robot box-pushing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.: 'Human Robot Interaction Coop. Robots'*, 1995, vol. 3, pp. 556–561.
[12] D. Rus, B. Donald, and J. Jennings, "Moving furniture with teams of autonomous robots," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 'Human Robot Interaction Coop. Robot'*, 1995, vol. 1, pp. 235–242.
[13] R. G. Brown and J. S. Jennings, "A pusher/steerer model for strongly cooperative mobile robot manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 'Human Robot Interaction Coop. Robot'*, 1995, vol. 3, pp. 562–568.
[14] M. Rubenstein *et al.*, "Collective transport of complex objects by simple robots: Theory and experiments," in *Proc. Int. Conf. Auton. Agents Multi-Agent Syst.*, 2013, pp. 47–54.
[15] P. Song and V. Kumar, "A potential field based approach to multi-robot manipulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, vol. 2, pp. 1217–1222.
[16] I. Mas and C. A. Kitts, "Object manipulation using cooperative mobile multi-robot systems," in *Proc. World Congr. Eng. Comput. Sci.*, 2012, vol. 1, pp. 324–329.
[17] A. Stroupe, T. Huntsberger, A. Okon, H. Aghazarian, and M. Robinson, "Behavior-based multi-robot collaboration for autonomous construction tasks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2005, pp. 1495–1500.
[18] Z. Wang, Y. Takano, Y. Hirata, and K. Kosuge, "Decentralized cooperative object transportation by multiple mobile robots with a pushing leader," in *Distributed Autonomous Robotic Systems 6*, New York, NY, USA: Springer, Jan. 2007, pp. 453–462.
[19] M. A. Neumann, M. H. Chin, and C. A. Kitts, "Object manipulation through explicit force control using cooperative mobile multi-robot systems," in *Proc. World Congr. Eng. Comput. Sci.*—, 2014, vol. 1, pp. 364–369.
[20] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. Robot. Autom.*, vol. RA-3, no. 1, pp. 43–53, Jan. 1987.
[21] C. A. Kitts *et al.*, "Field operation of a robotic SWATH boat for shallow water bathymetric characterization," *J. Field Robotics*, vol. 29, no. 6, pp. 924–938, Nov./Dec. 2012.
[22] 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.
[23] P. Mahacek, C. A. 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.
[24] T. Adamek, C. A. Kitts, and I. Mas, "Gradient-based cluster space navigation for autonomous surface vessels," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 2, pp. 506–518, Jan. 2014.
[25] I. Mas and C. A. Kitts, "Dynamic control of mobile multi-robot systems: the cluster space formulation," *IEEE Access* vol. 2, pp. 558–570, May 2014. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber = 6818372
[26] I. Mas and C. A. Kitts, "Obstacle avoidance policies for cluster space control of nonholonomic multirobot systems," *IEEE/ASME Trans. Mechatronics*, vol. 17, no. 6, pp. 1068–1079, Dec. 2012.
[27] I. Mas, J. Acain, O. Petrovic, and C. A. Kitts, "Error characterization in the vicinity of singularities in multi-robot cluster space control," in *Proc. IEEE Int. Conf. Robot. Biometrics*, 2008, pp. 1911–1917.