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

Swarm-Based Object Manipulation using Redundant Manipulator Analogs

Bradley E. Bishop, *Member, IEEE*

Abstract- **In this paper, we present a method utilizing manipulability concepts and redundant manipulator analogs for swarm manipulation of a rigid object in the plane using autonomous surface vessels. The key concepts involve maximizing some metric defined in relation to a manipulability ellipsoid, which represents the mapping between applied thrust and configuration space accelerations for the object. By casting this problem in the framework of redundant manipulation, the proposed method can be carried out in real time, in a possibly changing environment, and can accommodate nonholonomic vehicles and thrust limits on the swarm. The controller includes attachment pose optimization, engagement control as the ASVs approach and attach to the object, and the manipulation control. Simulation studies demonstrate the efficacy of the proposed methods.**

I. Introduction

Controller development for effective cooperation among units in swarms of robotic systems requires significant care. A great deal of effort has been directed at developing centralized and decentralized control strategies for a wide variety of swarming applications [1,2,3,4,5,6,7]. A concern that is not addressed by these controller formulations is the ability to quantify and allocate swarm resources appropriately for various missions and objectives. A framework that encodes both cooperation and appropriate capability arbitration was presented in [8]. In this paper, we extend that conceptual framework to the problem of cooperative manipulation of water-borne rigid objects.

Cooperative manipulation using mobile robots has a long history. Methods using units as virtual castors are common for ground-based systems [9]. Other methods include behavior-based schemes [10] and a variety of additional techniques and domains (see, e.g., [11, 12, 13]).

The controller used in this work is most closely related to the system in [9], but the current method was designed for swarms of autonomous surface vessels (ASVs) that rigidly attach to an object to be manipulated on the surface of the water instead of ground-based robots manipulating a carried object. The proposed controller has the following characteristics:

i) Generation of locally optimal attachment points and thrust angles for the ASVs
ii) Nonholonomic motion control to achieve contact and lock-on of the ASV on the object
iii) Local thrust redistribution to optimally utilize available capability

This work was supported in part by the Office of Naval Research under the National Naval Responsibility in Naval Engineering (NNR-NE) Program (Grant N0014-03-1-0160). B. E. Bishop is with the United States Naval Academy, Annapolis MD 21402 (phone 410-293-6117; fax: 410-293-2215; email: bishop@usna.edu).

The remainder of the paper is organized as follows. In Section II, we discuss the problem statement and the assumptions on swarm composition. In Section III, we define the control method that is used to achieve swarm control. Section IV includes several simulations demonstrating the efficacy of the system. Finally, we offer conclusions and some ideas for future work in Section V.

II. Problem Statement

A. Vehicle Kinematics

In this paper, we will consider control of a swarm of n cooperating vehicles with kinematics that approximate autonomous surface vessels (ASVs). The model that we will use, while not reflecting the full richness of a surface vessel, contains the crucial kinematics needed for development of a controller that can achieve attachment between the vessel and a specified point on an object to be manipulated.

The free-motion model of the system (when the ASV is not in contact with the body to be manipulated) bears a strong similarity to the kinematics of a tricycle-drive robot, but also loosely represents the basic velocity kinematics of a surface vessel that has a fixed thruster and a single control surface (rudder). The dynamics are given in (1) and illustrated in Figure 1, where the vehicle speed is u and the rudder angle is δ. Note that there is a factor of ½ in the dynamics of this system that is not present in standard tricycle-drive systems, to represent in a limited way the effects of viscous friction and drag for ASVs.

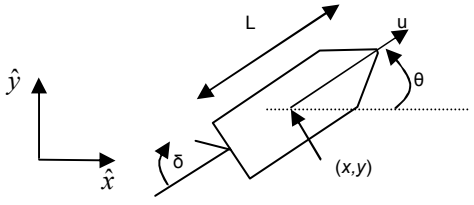

Figure 1: Autonomous Surface Vessel Kinematic Model

$$\begin{aligned} \dot{x} &= u\cos(\theta) \\ \dot{y} &= u\sin(\theta) \\ \dot{\theta} &= \frac{u}{2L}\sin(\delta) \end{aligned} \tag{1}$$

This is an underactuated system, where the two control inputs (speed u and rudder angle δ) control three state variables (x, y, θ) through a nonlinear relationship (we use a canonical set of coordinates in this example, as the actual control of the vessel will be in terms of the pose with respect to the object to be manipulated). The result of this

underactuation is a nonholonomic constraint, limiting the achievable velocities of the system.

To accommodate this type of actuation, we write a single-unit open-water controller that tracks a desired velocity $\left(\dot{x}^d, \dot{y}^d, \dot{\theta}^d\right)$ for a given unit using the two available actuations (u, δ). The controller is given in (2), where K_u is a control gain and any numerical difficulties with the arctangent are handled using the four quadrant *atan2*.

$$u = \sqrt{(\dot{x}^d)^2 + (\dot{y}^d)^2} \tag{2}$$
$$\dot{\theta}^d = K_u\left(\tan^{-1}\left(\frac{\dot{y}^d}{\dot{x}^d}\right) - \theta\right)$$
$$\delta = \begin{cases} \operatorname{sgn}\left(\frac{2L\dot{\theta}^d}{u}\right)\frac{\pi}{4} & when \quad \left|\frac{2L\dot{\theta}^d}{u}\right| > \sin\left(\frac{\pi}{4}\right) \\ \sin^{-1}\left(\frac{2L\dot{\theta}^d}{u}\right) & when \quad \left|\frac{2L\dot{\theta}^d}{u}\right| \le \sin\left(\frac{\pi}{4}\right) \end{cases}$$

We will assume that each ASV has limits on the rudder angle (as shown in (2)), the achievable speed (u) and the available thrust (which is not a component of the kinematic model, but will be important for manipulation). We allow that individual units may have differing limitations (as we may be using various types and sizes of ASV). The desired motion of the ASV will be determined by the swarm-level controller defined in Section III.

B. Task Model

The task to be accomplished by the swarm of ASVs is cooperative manipulation of a rigid body through pushing, as seen in Figure 2.

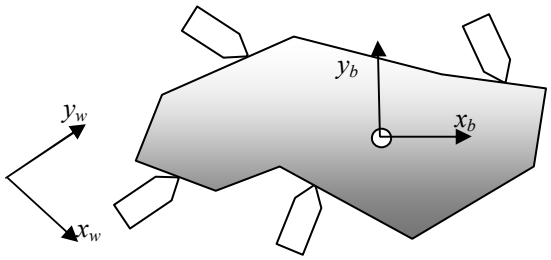

Figure 2: Rigid object being manipulated by a swarm of cooperating vehicles

Given a set of n units making contact with a vessel at points defined in the body-fixed reference frame (located at the center of mass of the object, for simplicity), the relationship between the body-frame accelerations A of the rigid body and the unit thrust vector q is given by:

$$A = \begin{bmatrix} a_x \\ a_y \\ \alpha \end{bmatrix} = J\begin{bmatrix} \tau_1 \\ \vdots \\ \tau_n \end{bmatrix} = Jq \tag{3}$$
$$J = \begin{bmatrix} m^{-1}\cos(\theta_1) & \cdots & m^{-1}\cos(\theta_n) \\ m^{-1}\sin(\theta_1) & \cdots & m^{-1}\sin(\theta_n) \\ I_z^{-1}(x_1\sin(\theta_1) - y_1\cos(\theta_1)) & \cdots & I_z^{-1}(x_n\sin(\theta_n) - y_n\cos(\theta_n)) \end{bmatrix}$$

where θ_i is the angle and (x_i, y_i) the position of the i[th] vessel with respect to the coordinate frame attached to the rigid body, τ_i is the thrust applied by that vessel (replacing speed u as a control variable when in contact with the target object), m is the overall system mass and I_z is the moment of inertia of the system about the z_o axis. Note that we have chosen for this discussion to ignore the nonlinear hydrodynamic forces (added mass, etc. [14]) that would affect the actual motion. Additionally, we have lumped the mass of the ASVs together with the mass of the rigid body, and we assume that moment of inertia of the ASVs about the z axis is small relative to the vessel, no matter the configuration.

A crucial consideration for this formulation is the possibility that the Jacobian may become singular. It is noted that singularities only occur kinematically when all units have the same body-frame angle or algorithmically on sets of measure zero.

The world-coordinate accelerations of the body are defined by:

$$R = \begin{bmatrix} \cos(\theta_w) & -\sin(\theta_w) & 0 \\ \sin(\theta_w) & \cos(\theta_w) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$
$$A_w = \begin{bmatrix} \ddot{x}_w \\ \ddot{y}_w \\ \ddot{\theta}_w \end{bmatrix} = R \cdot A - K_f \begin{bmatrix} K_f \dot{x}_w \\ K_f \dot{y}_w \\ K_{f\theta}\dot{\theta}_w \end{bmatrix}$$

where (x_w, y_w, θ_w) is the pose of the object in the world coordinate frame and K_f, $K_{f\theta}$ represent values for viscous friction.

C. Capability Function

It is the goal of the system to determine a set of locations and angles for the ASVs such that any desired body-frame accelerations can be accomplished with minimal thrust from the ASVs. Using the framework of [8], we must develop a functional representation of the capabilities of the system to achieve desired objectives.

The capability function that is considered for this application involves the use of the manipulability ellipsoid and is a coordinated concept (as opposed to the individual unit concept of the capability functions discussed in [8]). Given the Jacobian-like matrix J from (3), relating unit thrust to body-frame accelerations, we can define an analog to the manipulability ellipsoid, following [15]. The developed manipulability ellipsoid analog encodes the ease of transmission of thrust to body acceleration, which is a slight deviation from the standard velocity-based method.

The modified manipulability ellipsoid that we will use in this work (which we will refer to as simply the *capability ellipsoid*) is given by the set of all A_w such that:

$$A_w{}^T\left(JJ^T\right)A_w \le 1$$

Other formulations of the dynamic manipulability for robot manipulators are given in [15, 16]. The selected formulation represents the ease of converting unit thrust into body-frame accelerations.

The capability functions that we will consider are the

volume of the capability ellipsoid as well as direction specific capability, based on the *task compatibility* concept of [17]:

Volume: $V = \sqrt{\det(JJ^T)}$

Task compatibility: $\alpha = \left[u^T (JJ^T) u\right]^{-\frac{1}{2}}$

where α is the measure of the distance between the origin and the capability ellipsoid in the chosen direction u, corresponding to the relative ease of transmission of thrust to acceleration in that specified direction.

III. CONTROL SYSTEM

The complete controller for the task discussed in Section II includes multiple phases, each handling a distinct aspect of the problem. Each segment of the control will be addressed separately in the sections that follow. The phases include:

i) A numerical *attachment pose generator* which computes desired attachment points and thrust angles

ii) An *engagement* controller which achieves gross motion of the ASVs to the attachment points

iii) Manipulation of the object using redundant manipulator analogs, involving appropriate accommodation for reconfiguration and limitations on available thrust

A. Attachment pose generator

The primary objective of the swarm of ASVs will be to efficiently utilize the available thrust. As such, we want to select attachment points and thrust angles that allow not just manipulation of the object, but *efficient* manipulation.

The technique by which we select attachment poses involves a gradient controller defined on the capability function (volume or task compatibility) of choice, along the available directions defined by the exterior of the object to be manipulated.

We begin with a set of preliminary, phantom locations on the boundary of the object to be manipulated, defined by projection of the current locations toward the center of mass of the object. We then generate a dynamic system that modifies the phantom locations by following the gradient of the capability measure. That is, each phantom location moves along the boundary of the object toward a configuration that provides the greatest capability. If we choose V as the capability measure, and we define the body-frame pose of the phantom attachment pose for unit i as $[x_i, y_i, \theta_i]$, we have the following gradient-based controller to determine the desired attachment poses:

$$
\begin{aligned}
&D = \max\left(\nabla_{x,y} V \circ s^+, \nabla_{x,y} V \circ s^-\right) \\
&\begin{bmatrix} \dot{x}_i^{\,d} \\ \dot{y}_i^{\,d} \end{bmatrix} = \begin{cases} Ds^+ \; if & \nabla_{x,y} V \circ s^+ \geq \nabla_{x,y} V \circ s^- \\ Ds^- \; if & \nabla_{x,y} V \circ s^+ < \nabla_{x,y} V \circ s^- \end{cases} \\
&\dot{\theta}_i = \frac{\partial V}{\partial \theta_i}
\end{aligned}
\tag{5}
$$

where s^+ and s^- are the unit vectors representing the two local tangent directions (left and right) for the object at the currently defined contact point, $\nabla_{x,y} V$ is the spatial gradient of the capability function at the current contact point and ∘ is the vector dot product, acting as a projection operator of the gradient onto the available directions of motion. Thus, the motion of the phantom location along the boundary can follow the edge even at corners.

There are two limitations that must be imposed on the motion of the phantom attachment points using the derivation above. Specifically: θ_i must be achievable, based on the geometry of the object and the ASV; and no two vehicles can occupy the same attachment point, nor be within a specified distance σ of each other when the system terminates.

The method works well for convex objects, but significant concavities may result in local minima in the gradient search which must be dealt with carefully. Accommodation for such significant concavities is beyond the scope of this work. It is important to understand that the system that is generating the desired attachment points does so in the absence of the actual vessels, so that 'motion' of the phantom point does not need to obey the kinematic constraints on the actual vessels, which will move to the specified points using a different method, the *engagement* controller.

B. Engagement Controller

In order for a set of units to manipulate an object, they must be brought into contact with that unit. As such, we need to develop a controller that can bring the ASVs (with kinematics given in (1)) into contact with the body to be manipulated, based on the desired attachment poses.

To do this, we assume that a dual-layer controller exists. The inner layer generates the desired attachment poses for the units based on the capability that is desired, as seen in Section III.A. The outer layer of control brings the ASV into position using a boundary-layer-like approach.

The full controller, which must include obstacle avoidance, speed accommodation, etc., is excluded for brevity, but a generic overview is appropriate. The basic engagement controller is given by (6) (a pictorial overview can be found in Figure 3) where K_e is a controller gain and the remainder of the variables are defined in Figure 3. Note that the sign in $\bar{t}$ is defined to swing the ASV toward the target heading along the shortest path. The results of this engagement system (desired unit-level velocities, $\left(\dot{x}_i^{\,d}, \dot{y}_i^{\,d}\right)$) are used as inputs for the nonholonomic controller from (2) above.

$$
\begin{aligned}
&\vec{e} = \begin{bmatrix} x_i^d \\ y_i^d \end{bmatrix} - \begin{bmatrix} x_i \\ y_i \end{bmatrix} \qquad \lambda = \tan^{-1}\left(\frac{e(2)}{e(1)}\right) \\
&\bar{t} = \begin{bmatrix} \cos\left(\lambda \pm \frac{\pi}{2}\right) \\ \sin\left(\lambda \pm \frac{\pi}{2}\right) \end{bmatrix} \qquad \gamma = (\pi + \lambda) - \theta_i^d \\
&\begin{bmatrix} \dot{x}_i^d \\ \dot{y}_i^d \end{bmatrix} = K_e e + (\| e \| \gamma)^2 \bar{t}
\end{aligned}
\tag{6}
$$

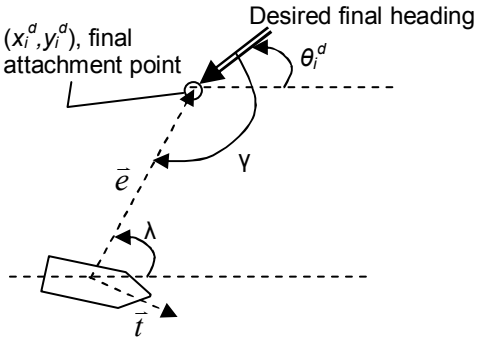

Figure 3: The engagement problem

Fundamental to this control concept is the line projected from the desired final contact point along the desired thrust angle, which is the final heading in Figure 3. The engagement controller has two distinct components: a radial control component that closes distance between the ASV and the target point (given by $K_e e$), and an angular control component that brings the ASV onto the projected line (given by $(\| e \| \gamma)^2 \vec{t}$). If the ASV is far from the projection line, the angular component of the control is dominant. As the ASV nears the projected line, the radial component of the control begins to dominate and the vessel steers toward the target point. Careful design of the gains used for this control requires knowledge of the unit kinematics, but is quite tractable. The system functions well if the distance between the initial position of the ASVs and the object is much greater than the turn radius of the ASVs. If this is not the case, other nonholonomic motion planning methods must be used.

C. Manipulation Controller

Once the units are in contact with the object to be manipulated in the desired attachment poses, the object can be manipulated. The base controller for manipulation of the object utilizes the Jacobian J along with a redundant manipulator formulation, which is very similar to the method in [9]. We base our complete control scheme on the methodology from [1], using redundant manipulator analogs (later referred to as *kinematic control* in [2]). This control method was originally designed to regulate swarm-level functions, such as mean and variance, while still allowing the individual units some degree of autonomy. The controller, based on redundant manipulator methods such as those discussed in [18], has proven to be an extremely effective technique for swarm control. The basic controller must be modified in this case to meet the requirements of the task at hand, but follows the fundamental development well.

The standard formulation of the controller is as follows:

$$A_w^{\ d} = \begin{bmatrix} K_p\left(x_w^{\ d} - x_w\right) + K_v\left(\dot{x}_w^{\ d} - \dot{x}_w\right) + \ddot{x}_w^{\ d} \\ K_p\left(y_w^{\ d} - y_w\right) + K_v\left(\dot{y}_w^{\ d} - \dot{y}_w\right) + \ddot{y}_w^{\ d} \\ K_{p\theta}\left(\theta_w^{\ d} - \theta_w\right) + K_{v\theta}\left(\dot{\theta}_w^{\ d} - \dot{\theta}_w\right) + \ddot{\theta}_w^{d} \end{bmatrix} \qquad (7)$$

$$q^d = J^+\left(R^{-1} A_w^{\ d}\right) + K_N (I - J^+ J) v$$

where q^d is the vector of desired thrust values τ_i, J^+ is the pseudoinverse of J given by $J^T(JJ^T)^{-1}$, K_p, K_v $K_{p\theta}$ and $K_{v\theta}$ are controller gains, K_N is the null-space gain value, $[x_w^{\ d}(t), y_w^{\ d}(t), \theta_w^{\ d}(t)]^T$, is the desired object trajectory in the world coordinate frame, $(I - J^+J)$ is the null-space projection operator that guarantees coordination, and v is an encoded secondary task. One of the primary differences between this work and that of [9] (and related work) is the null-space control aspect. A secondary task v can be carried out on the null space of the primary task, which allows systems with more degrees of freedom than task variables to coordinate thrust in a way that maintains the swarm-level function (body-frame acceleration) while using any additional degrees of freedom to carry out secondary objectives.

In the case of object manipulation, the secondary objective of the system involves adherence to the limitations on available thrust. In [19], a methodology for controlling nonholonomic swarms was developed, and was extended later to dynamic swarms [20]. The fundamental technique applied in those works accommodates any deviation between a desired and achievable unit actuation (there, it involved velocities and torque, here we apply it for thrust computation). Details can be found in [19]. The key concept is that this formulation redistributes available thrust to try to achieve the best possible approximation of the desired body-frame accelerations. When the thrusts are all achievable, or the maximum number of iterations has been reached, we check the computed q again. If the desired thrust is too great for one or more units, every thrust value is scaled by the same factor S, defined as:

$$S = \max\left(\begin{cases} \dfrac{\tau_i^-}{\tau_i} & if \quad \tau_i < 0 \\ \dfrac{\tau_i^+}{\tau_i} & if \quad \tau_i > 0 \end{cases}\right) \forall i \quad s.t. \quad \tau_i > \tau_i^{\operatorname{sgn}\tau_i} \qquad (8)$$

This guarantees that the coordination of the thrust is achieved so that the body-frame acceleration vector is aimed appropriately in the task space. Note that the definitions accommodate different available thrust in the forward and reverse directions, which allows us to utilize this method for heterogeneous swarms.

IV. Simulations

In this section, we demonstrate a complete instantiation of the control scheme. We use four ASVs (redundant in terms of the 3DOF task which we wish to accomplish) and the following parameters:

Controller:

$K_e = K_u = K_N = 1$, $K_p = K_v = K_{p\theta} = K_{v\theta} = 10$

Capability: volume V of the capability ellipsoid

Object to be manipulated:

Rectangular solid, 3m X 2m

m = 1000 kg, I_z = 1083 kg-m^2

$K_f = K_{f\theta} = 0.25$
Initial pose: $[x_w, y_w, \theta_w] = [0, 0, 0]$
Target pose: $[x_w, y_w, \theta_w] = [-5, -10, \pi]$

ASV parameters:

L = 0.5m, limit on rudder angle of $\pi/4$
Max thrust = 500N in positive direction, 300N in reverse for units 1 and 2, 250N in the positive and 300N in reverse for units 3 and 4
Initial positions: $[x_1\ x_2\ x_3\ x_4] = [10\ 15\ 13\ 7]$
$[y_1\ y_2\ y_3\ y_4] = [-0.1\ 0\ 0.1\ 0.5]$
$[\theta_1\ \theta_2\ \theta_3\ \theta_4] = [\pi\ \pi\ \pi\ \pi{+}0.1]$

We note that the Jacobian J will be singular if all of the units have an identical heading that is a multiple of $\pi/2$, which is a degenerate case and not of general interest.
We start the controller by performing attachment point generation. We define the phantom positions of the units around the perimeter of the object to be moved, by simply projecting the starting poses toward the center of mass of the object. We then apply the gradient descent attachment pose generation scheme. The result is shown in Figure 4 - Figure 6, below. The profile of capability (volume of the ellipsoid), shown in Figure 5, is iteration-based, as opposed to time-based. The attachment pose generator was run as a sampled-data simulation, so the timeframe of the evolution was not relevant. The capability ellipsoids shown in Figure 6 show the efficacy of the attachment pose generation scheme. Note that the phantom attachment poses passed closer to the origin of the coordinate frame as they moved along the two horizontal edges from their initial poses, but the overall capability was monotonically increasing.

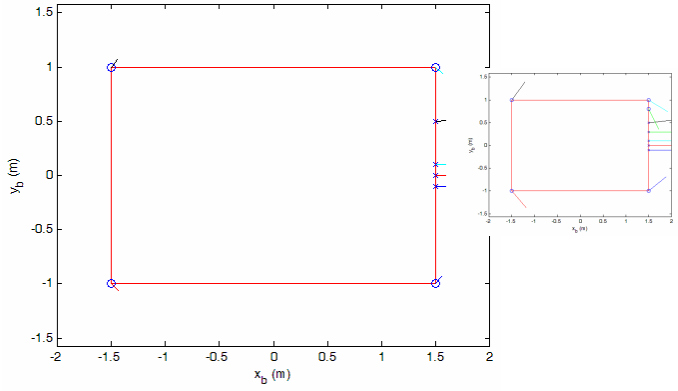

Figure 4: Attachment pose generation results. Initial poses shown with an 'x', desired poses with an 'o.' The inset figure shows a five unit example, where the two units in the upper right are prevented from converging.

After the desired attachment poses were generated, the engagement controller was applied for the four vessels, as seen in Figure 7. Note that the path taken by unit #4 (dotted line) was selected purposefully to show the effect of the varying sign in (6), as the correct direction would have been counterclockwise. All of the other units followed the appropriate path.

Finally, the manipulation controller was applied to move the object to [-5, -10] with an orientation of π. The trajectory of the object is shown in Figure 8. As can be seen from Figure 9, the thrust is redistributed to make better use of the capabilities of the units. The fact that the modified thrust vector is of greater magnitude indicates, in this case, a better use of resources, as large desired thrust that would be beyond the available thrust threshold is redistributed among other units and is available to be applied to the system with no degradation of the primary task.

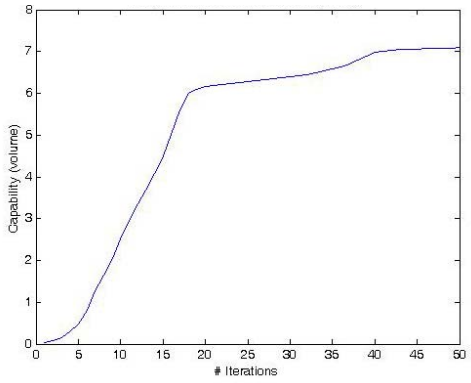

Figure 5: Capability trajectory during attachment pose generation

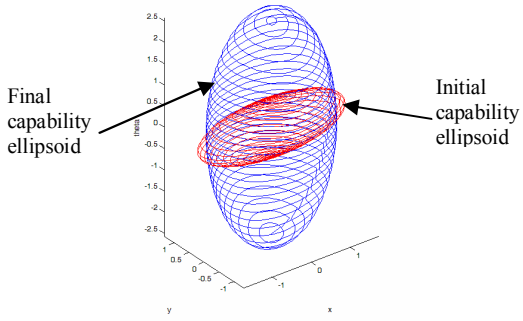

Figure 6: Capability ellipsoid at the start and end of attachment pose generation.

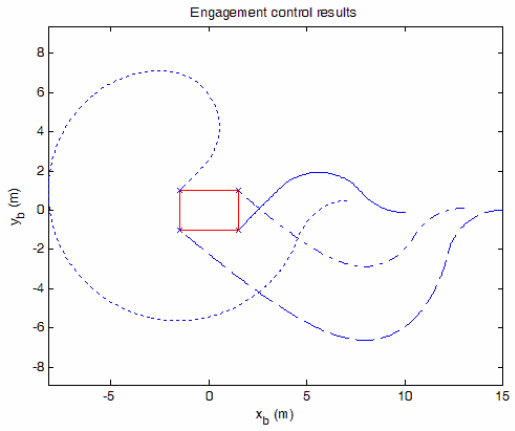

Figure 7: Results of the engagement control for unit #1 (solid), #2 (dashed) #3 (dash-dot) and #4 (dotted).

V. Conclusion

In this work, we have presented a complete cooperative manipulation controller for groups of autonomous surface

vessels. The controller includes attachment pose generation, engagement control and cooperative manipulation. The primary contributions of the work are the optimization of the attachment poses and the null-space thrust re-distribution. The engagement controller is also new, but is *ad hoc* in design and may fail if the gains are poorly selected or the ASVs are too close to the desired poses.

The framework presented in this work is flexible, allowing a wide variety of additional secondary objectives, depending on the nature of the ASVs and the attachments. Using null-space methods to adjust the poses of the units during operation is an exciting and promising area of research, as is the accommodation of push-only, frictional contact. Preliminary investigations into these formulations suggest that existing framework could be used with little modification to allow these new constraints and abilities, although such are beyond the scope of this work.

The work as it stands is centralized, but the underlying controller has been shown to be amenable to partial decentralization [21], which is an area of continued research. The addition of, and accommodation for, more realistic hydrodynamic forces will also enhance the results, especially when coupled with the dynamics-based techniques from [20] and a more realistic set of objects and ASVs.

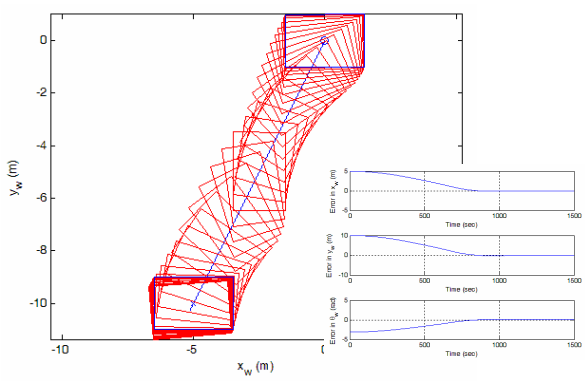

Figure 8: Object motion (main) and tracking errors (inset)

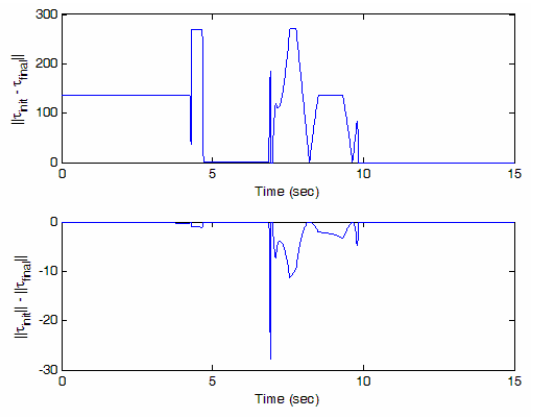

Figure 9: The difference between the raw desired thrust and the thrust after redistribution, shown two ways.

REFERENCES

[1] B.E. Bishop, "On the Use of Redundant Manipulator Techniques for Control of Platoons of Cooperating Robotic Vehicles," *IEEE Transactions on Systems, Man and Cybernetics,* Vol. 33, No. 5, Sept. 2003, pp. 608 – 615.

[2] G. Antonelli and S. Chiaverini, "Fault tolerant kinematic control of platoons of autonomous vehicles," in *Proc. IEEE Int. Conf. on Robotics and Automation*, New Orleans, LA, 2004, pp. 3313–3318.

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

[4] E. Roszkowska, "Provably Correct Closed-Loop Control for Multiple Mobile Robot Systems," *Proc. IEEE Int. Conf. on Robotics and Automation*, Barcelona, Spain 2005, pp. 2821 - 2827.

[5] V. Gazi and K. M. Passino, "Stability analysis of swarms," *IEEE Trans. Automat. Contr.*, vol. 48, no. 4, pp. 692–697, 2003.

[6] N. E. Leonard and E. Fiorelli, "Virual leaders, artificial potentials, and coordinated control of groups," in *Proc. IEEE Conf. Decision and Control*, Orlando, FL, 2001, pp. 2968 – 2973.

[7] F. Zhang, M. Goldgeier, and P. S. Krishnaprasad, "Control of small formations using shape coordinates," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Taipai, Taiwan, September 2003, pp. 2510–2515.

[8] B. E. Bishop, "On the Use of Capability Functions for Cooperative Objective Coverage in Robot Swarms," *IEEE Int. Conference on Robotics and Automation,* April 2007, pp. 2306 – 2311.

[9] D.J. Stilwell and J.S. Bay, "Optimal control for cooperating mobile robots bearing a common load", *IEEE Int. Conference on Robotics and Automation*, 8-13 May 1994 Page(s):58 – 63

[10] W. ZhiDong , E. Nakano, and T. Takahashi, "Solving function distribution and behavior design problem for cooperative object handling by multiple mobile robots," *IEEE Transactions on Systems, Man and Cybernetics, Part A,* Volume 33, Issue 5, Sept. 2003 Page(s):537 - 549

[11] L. Qingguo and S. Payandeh, "Modeling and analysis of dynamic multi-agent planar manipulation," Proceedings 2001 IEEE *International Symposium on Computational Intelligence in Robotics and Automation*, 29 July-1 Aug. 2001 Page(s):200 - 205

[12] J. Shao, J. Yu, L. Wang and W. Zhang, "Motion planning of cooperative disk-pushing for multiple biomimetic robotic fish," 2005 *IEEE International Conference on Systems, Man and Cybernetics,* 10-12 Oct. 2005 Page(s):3143 – 3148

[13] E. Smith, M. Feemster, and J. Esposito, "Swarn Manipulation Of An Unactuated Surface Vessel," *Thirty-Ninth Southeastern Symposium on System Theory*, March 2007 Page(s):16 – 20

[14] T. Fossen. *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*. Trondheim, Norway: Marine Cybernetics, 2002.

[15] T. Yoshikawa, *Foundations of Robotics*, Cambridge, MA, The MIT Press, 1990.

[16] K. Kazerounian and Z. Wang, "Global versus local optimization in redundancy resolution of robotic manipulators," *International Journal of Robotics Research*, vol. 7, no. 5, pp. 3 – 12, 1988.

[17] S. L. Chiu, "Task compatibility of manipulator postures," *International Journal of Robotics Research,* vol. 7., no. 5., pp. 13 – 21, 1988.

[18] B. Siciliano, "Kinematic Control of Redundant Robot Manipulators: A Tutorial," *Journal of Intelligent Robotic Systems*, Vol. 3, pp. 201-212, 1990.

[19] B. E. Bishop, "Control of Platoons of Nonholonomic Vehicles Using Redundant Manipulator Analogs," *ASME Journal of Dynamic Systems, Measurements and Control, Special Issue on Novel Robots and Control,* Vol. 128, No. 1, pp 171 – 175, 2006.

[20] B.E. Bishop, "Dynamics-based control of robotic swarms" *Proc. 2006 IEEE Int. Conference on Robotics and Automation*, May 15-19, 2006 Page(s):2763 - 2768

[21] D.J. Stilwell, B.E. Bishop and C.A. Sylvester, "Redundant Manipulator Techniques for Partially Decentralized Path Planning and Control of a Platoon of Autonomous Vehicles," *IEEE Transactions on Systems, Man and Cybernetics Part B,* Volume 35, Issue 4, Aug. 2005 Page(s):842 - 848.