Control of Platoons of Nonholonomic Vehicles Using Redundant Manipulator Analogs[1]

Bradley E. Bishop
Systems Engineering Department (Stop 14A),
United States Naval Academy,
Annapolis, MD 21402
e-mail: bishop@usna.edu

In this work, we consider the control of platoons of cooperating nonholonomic vehicles. Using techniques based on redundant manipulator control, the platoon is treated as a single entity with a set of platoon-level objectives. The class of tricyclelike robots, with limits on steering and speed, is chosen because it represents a vast class of real, nonholonomic vehicles beyond the basic differential drive. The method presented uses platoon redundancy to limit the impact of vehicle constraints on the platoon-level objectives. A simulation study is presented to show the efficacy of the method. [DOI: 10.1115/1.2168159]

Keywords: multirobot control, nonholonomic systems

1 Introduction

Control of platoons of cooperating vehicles is a challenging problem. When those vehicles are nonholonomic, the difficulties are compounded. In this work, we present a nonholonomic vehicle platoon controller based on the algorithm developed in [1]. The approach proposed in this paper has the benefit that it does not require unit-level path planning or any specification of the final formation of the platoon. This method shares some of the desirable characteristics of behavior-based systems (such as that of [2]) in that the platoon formation will be determined by the environment to some extent, although the systems-theoretic framework guarantees system stability and performance. The proposed method allows for limits on achievable speed, steering angle, and steering rate. Related work on cooperating nonholonomic mobile robots can be found in [3] for manipulators and [4] for strict platoon formation control based on graph theory.

The remainder of this paper is organized as follows: Section 2 contains details of the kinematics model used, and Sec. 3 covers the basic controller. Section 4 contains simulation results for a platoon of eight vehicles following a moving target under kinematic constraints. Section 5 offers conclusions and future work.

2 Nonholonomic Vehicle Model

The basic vehicle model that will be considered in this work is a tricycle drive, as shown in Fig. 1, where the steering angle is α, the body heading is θ, vehicle length is d, and the location of the vehicle is (x,y), defined at the tip.

Vehicle kinematics are given by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = r\omega \begin{bmatrix} \cos(\theta+\alpha) \\ \sin(\theta+\alpha) \end{bmatrix}$$

$$\dot{\theta} = \frac{r\omega}{d\left[1+\tan^2\left(\frac{\pi}{2}-\alpha\right)\right]^{1/2}} = \frac{r\omega}{d}\sin(\alpha) \tag{1}$$

where r is the radius of the front wheel and ω is the angular velocity of the front wheel (which can be positive or negative, allowing backing).

In order for this model to be representative of a large class of nonholonomic vehicles, the speed of the units $(r\omega)$ was assumed to be bounded in absolute value to 1.0 m/s and the steering angle α was limited to a range of $\pm\pi/4$, with a wheel radius r of 2.5 cm and a body length d of 25 cm. Assuming a desired velocity for the vehicle point, $(\dot{x}_d,\dot{y}_d)$, the vehicle controller is given by

$$\alpha^d = \tan^{-1}\left(\frac{\dot{y}_d}{\dot{x}_d}\right) - \theta$$

$$\dot{\alpha} = K_\alpha(\alpha^d - \alpha) \tag{2}$$

$$\omega = \frac{1}{r}\sqrt{\dot{x}_d^2+\dot{y}_d^2}$$

where K_α is a gain for the steering control and $|\dot{\alpha}| \leq 40$ rad/s.

This unit-level controller is not intended to achieve set-point regulation, only to attempt to match an underlying velocity vector field. It can be shown that the convergence will be achieved if the derivative of the desired heading angle along a minimum-radius path of the vehicle is less than the body angular velocity over that arc (dictated by the maximum speed and steering angle). Details are excluded for brevity. The platoon-level controller that follows is designed to utilize the redundancy of the platoon to overcome the difficulties associated with the nonholonomicity.

3 Platoon Controller

The basic platoon controller used was originally developed for holonomic vehicles. Details can be found in [1], but a brief overview is warranted.

3.1 Base (Holonomic) Controller. The holonomic-unit swarm controller regulates platoon-level functions, such as mean and variance, while still allowing the individual units some degree of autonomy. This controllerf, based on redundant manipulator methods such as those discussed in [5] for manipulators and in [6] for mobile systems, has proven to be an extremely effective technique for platoon control [1].

Given a platoon of n holonomic vehicles, the $2n$-dimensional state is given by $q=[x_1,x_2,\ldots,x_n,y_1,y_2,\ldots,y_n]^T$. Given an m-dimensional platoon-level function of the state $f(q)$, the unit velocities are related to task space velocities by $\dot{f}(q)=J(q)\dot{q}$, where $J(q)$ is the Jacobian of the platoon function

$$J(q) = \begin{bmatrix} \frac{\partial f_1(q)}{\partial q_1} & \cdot & \cdot & \frac{\partial f_1(q)}{\partial q_{2n}} \\ & & \cdot & \\ & & \cdot & \\ \frac{\partial f_m(q)}{\partial q_1} & \cdot & \cdot & \frac{\partial f_m(q)}{\partial q_{2n}} \end{bmatrix} \tag{3}$$

In Eq. (4), we see a simple and useful task function $f(q)$. The mean determines the platoon position while variance dictates the spread of the elements without explicitly defining each unit's position

[1]This work was partially supported by the Office of Naval Research National Naval Responsibility in Naval Engineering (NNR-NE) program, Grant No. N0014-03-1-0160.

Contributed by the Dynamic Systems, Measurement, and Control Division of ASME for publication in the Journal of Dynamic Systems, Measurement, and Control. Manuscript received April 1, 2005; final manuscript received November 30, 2005. Assoc. Editor: Sunil K. Agrawal. Paper presented at the IEEE International Conference on Robotics and Automation 2005.

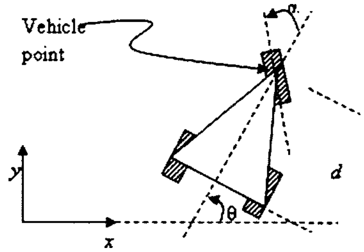

Fig. 1 Basic nonholonomic vehicle configuration

$$f(q)=\begin{bmatrix}\mu_x\\ \sigma_x^2\\ \mu_y\\ \sigma_y^2\end{bmatrix}=\begin{bmatrix}\frac{1}{n}\sum_{i=1}^{n} q_i\\ \frac{1}{n-1}\sum_{i=1}^{n}(q_i-\mu_x)^2\\ \frac{1}{n}\sum_{i=n+1}^{2n} q_i\\ \frac{1}{n-1}\sum_{i=n+1}^{2n}(q_i-\mu_y)^2\end{bmatrix} \quad (4)$$

For a large platoon, the number of state variables will be greater than the number of task variables in $f(q)$. Although each unit is constrained by the platoon-level function, there are an infinite number of possible configurations for the platoon that still achieve the desired task profile. A basic gradient projection controller [5] can be used to control the platoon in a centralized manner [1]

$$\dot{q}=J^{+}\{K[f_d(q)-f(q)]+\dot{f}_d(q)\}+(I-J^{+}J)v \quad (5)$$

where J^{+} is the pseudoinverse of J given by $J^T(JJ^T)^{-1}$, K is a controller gain matrix, f_d is a desired task function trajectory, and v is an encoded secondary task projected onto the null space of the primary task using $(I-J^{+}J)$. The secondary task v can be any velocity-based objective [7,8]. The controller uses a centralized architecture with feedback of the platoon functions f as well as environmental feedback for obstacle avoidance and target tracking [1]. In the sequel, a nonholonomic extension of Eq. (5) is developed.

3.2 Nonholonomic Extension of Base Controller. An initial nonholonomic controller was developed using $f(q)$ as defined in Eq. (4) along with the vehicle kinematics of Eqs. (1) and (2). Each unit takes, as inputs, the desired velocities developed in Eq. (5) and uses the control from Eq. (2). It can be shown that the platoon-based nonholonomic controller ((1), (2), and (5)) guarantees bounded velocities and positions for bounded desired task variables, although estimates on performance and convergence rates are difficult to derive. Related work on stability and performance analysis for nonholonimic platoons can be found in [9,10].

The secondary objective v is defined by an obstacle-avoidance vector, projected onto the null space of the Jacobian. The obstacle-avoidance routine is given by

$$\begin{bmatrix}v_{\mathrm{rep}o}(i)\\ v_{\mathrm{rep}o}(i+n)\end{bmatrix}=\begin{cases}\cos(\theta_{io})\left(\dfrac{1}{d_{mio}}-\dfrac{1}{d_{\min o}}\right)\begin{bmatrix}\cos(\theta_{mio})\\ \sin(\theta_{mio})\end{bmatrix} & \text{when } |\theta_o|<\dfrac{\pi}{2}, \quad d_{mio}<d_{\min o}\\ \begin{bmatrix}0\\0\end{bmatrix} & \text{when } |\theta_o|\geq\dfrac{\pi}{2}, \quad d_{mio}\geq d_{\min o}\end{cases} \quad (6)$$

where d_{mio} is the distance from unit i to the nearest obstacle point, $d_{\min o}$ is the cutoff distance for activation of the avoidance vector, θ_{mio} is the angle of the vector pointing away from the nearest obstacle point and θ_{io} is angle between the current heading θ and the point on the obstacle closest to the projected unit path. This obstacle-avoidance routine is designed to limit the influence of obstacles on unit motions when the unit has already passed the obstacle or will pass very far away from it on its current heading.

A similar term is used for interrobot repulsion. This obstacle avoidance term has a unique $d_{\min r}$ and no path-projection component:

$$\begin{bmatrix}v_{\mathrm{rep}r}(i)\\ v_{\mathrm{rep}r}(i+n)\end{bmatrix}=\begin{cases}\left(\dfrac{1}{d_{mir}}-\dfrac{1}{d_{\min r}}\right)\begin{bmatrix}\cos(\theta_{mir})\\ \sin(\theta_{mir})\end{bmatrix} & \text{when } d_{mir}<d_{\min r}\\ \begin{bmatrix}0\\0\end{bmatrix} & \text{when } d_{mir}\geq d_{\min r}\end{cases} \quad (7)$$

The total repulsive vector for each unit is the sum of Eqs. (6) and (7) and all of the individual units' repulsive vectors (if they exist) are concatenated into v used in Eq. (5).

Note that obstacle avoidance could be made primary, as in [11], an alternative parametrization of the controller developed in [1]. In this investigation, we choose to select a primary objective with a more readily definable, analog metric, as opposed to the pass-fail of obstacle avoidance, in order to better evaluate the efficacy of the proposed scheme. This suggests that collisions may occur, primarily when obstacles are very large or have significant convexities [12]. All the work herein is amenable to obstacle avoidance as the primary objective, and the controller can be refined with methods from motion planning or even behavior-based approaches [13].

Because of the constraints on the unit motion, desired individual unit velocities will not always be achievable, resulting in errors in tracking a desired platoon-level function $f(q)$. The redundancy of the system can be used to alleviate at least part of this difficulty. This is accomplished by comparing the achievable unit velocities in a given configuration to the desired unit velocities and projecting the difference error back onto the null space of the primary Jacobian. Effectively, this method attempts to find a set of unit velocities that are both in accordance with $f(q)$ and the closest to achievable velocities defined by the steering and speed limitations.

Because the null space of the Jacobian is a nonlinear function of the state and the projection is a local optimization method, the error correction will not be perfect. Additionally, each vehicle admits a range of achievable velocities. The correction is carried out iteratively for each unit whose desired velocity is not achievable, until either all units have achievable velocities or a preset number of iterations has been reached. To prevent collisions, obstacle avoidance takes precedence over velocity matching when the vehicles are close to obstacles ($d_{mio}<d_{\min o}/4$) or other units

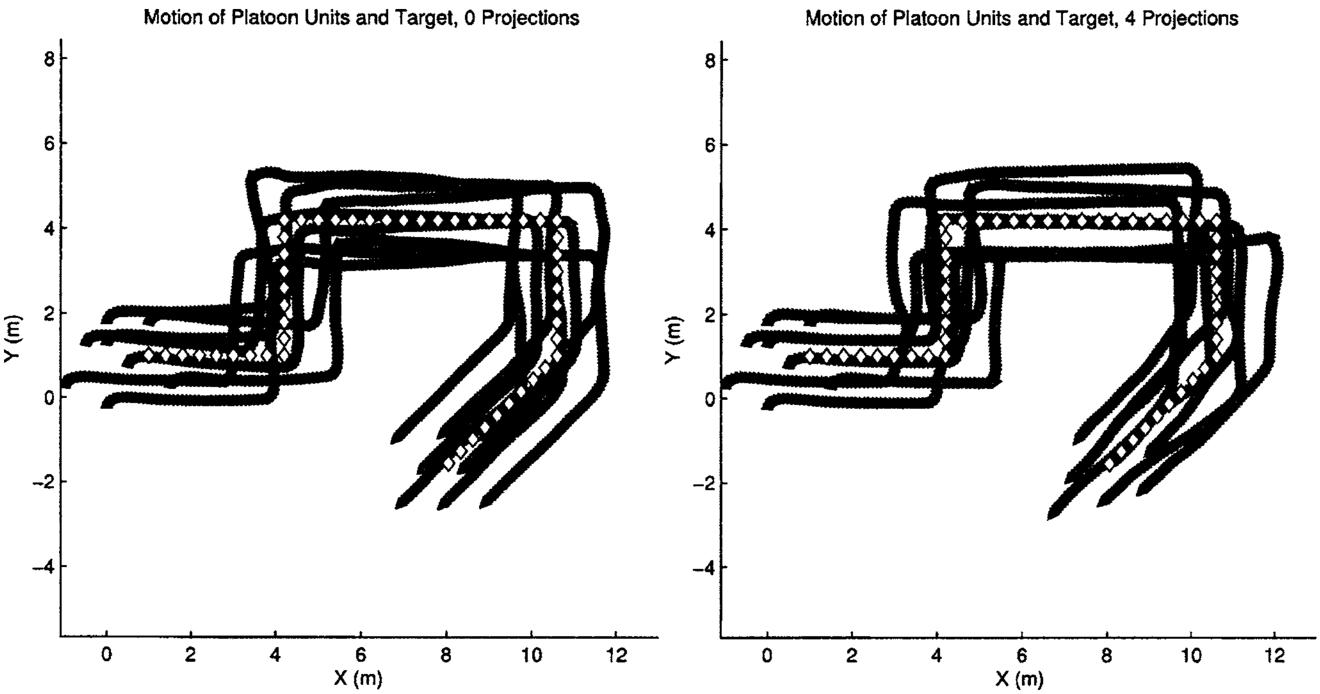

Fig. 2 platoon trajectories with zero and four error projection. The diamonds represent the moving target and the triangles are the units.

$(d_{mir} < d_{\min}/8)$.

The algorithm is given by the following pseudocode for an n unit platoon:

- Compute $\dot{q}^d$ from Eq. (5) using $v = v_{\text{repr}} + v_{\text{repo}}$ as in Eqs. (6) and (7)
- Set $v_{\text{corr}}(i) = 0$, $i \in [1, \ldots, 2n]$
- For $j = 1$:max iterations (or all velocities achievable)
  - For $i = 1 : n$
  - If $((d_{mio} > d_{\text{mino}}/4)$ and $(d_{mir} > d_{\text{minr}}/8))$
    Update v_{corr}:

$$\begin{bmatrix} v_{\text{corr}}(i) \\ v_{\text{corr}}(i+n) \end{bmatrix} += \begin{bmatrix} q^{\text{best}}(i) \\ q^{\text{best}}(i+n) \end{bmatrix} - \begin{bmatrix} \dot{q}^d(i) \\ \dot{q}^d(i+n) \end{bmatrix}$$

  where $\dot{q}^{\text{best}}$ is the achievable velocity for the system closest to $\dot{q}^d$, defined by the possible nonholonomic limits on steering and speed,
  - Compute new desired velocities

$$\dot{q}^d = J^{+}\{K[f_d(q) - f(q)] + \dot{f}_d(q)\} + (I - J^{+}J)(v_{\text{rep}} + v_{\text{corr}})$$

- Implement unit-level control using $\dot{q}^d$ and (2)

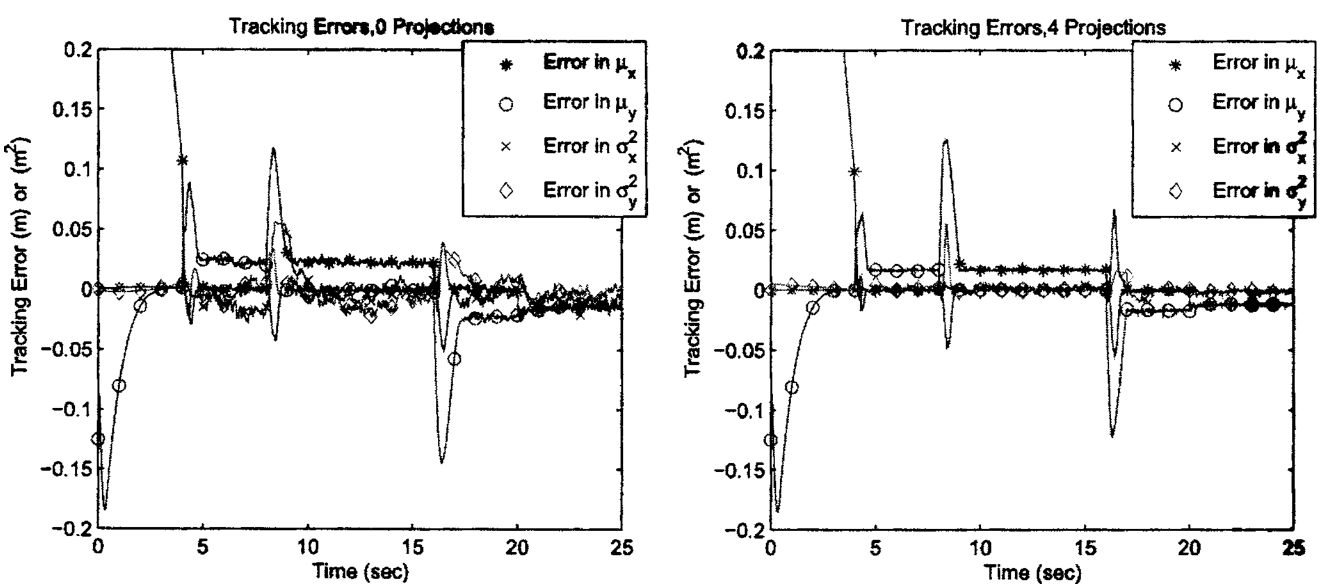

Fig. 3 Close-up of errors in platoon-level functions with zero and four error projections (common transient truncated)

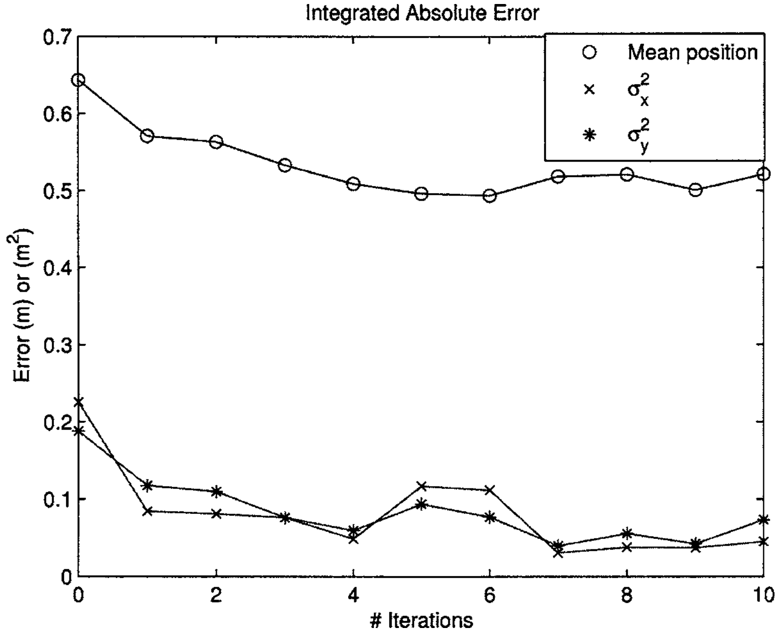

Fig. 4 Integrated error for various iterations, excluding transient mean position-tracking error

4 Nonholonomic Controller Results

A simulation was developed to test the efficacy of the proposed scheme. A holonomic moving target executed a sequence of maneuvers at speed 0.8 m/s while a platoon of eight nonholonomic units pursued. The goal was to match the platoon mean position to the target location while avoiding obstacles and maintaining the variance of the initial deployment ($\sigma_x^2=0.6384$ m^2 and $\sigma_y^2=0.5536$ m^2) under the limits on steering and speed as defined in Sec. 2. The interrobot $d_{\min r}$ was set at 1.0 m, while the robot-obstacle $d_{\min o}$ was 3.0 m. The gain matrix used in Eq. (5) is $K=\text{diag}(50,10,50,10)$, with $K_\alpha=100$.

Results of the simulations with no error projection (the base controller) and four error projections are shown in Fig. 2 and Fig. 3. The units all begin with $\theta=\pi/2$ and zero velocity on the left-hand side of the operational area. While the various unit trajectories for the two cases appear very similar, the overall error is significantly decreased in the projected case, as shown in Fig. 3 (with common transient portion truncated).

As a performance metric, the integrated absolute value of each error signal was computed. Performance was clearly enhanced by the controller, with integrated absolute tracking error decreased by 5.3% for mean position, 78.5% for σ_x^2, and 68.4% for σ_y^2. In fact, if we remove the transient portion of the tracking evolution (before ~4 s, while the initial error decays), the absolute integrated error in tracking position is reduced by 21%. Fewer iterations resulted in slightly degraded performance, and four iterations were found to be optimal for this example (see Fig. 4).

Experience across many trials has shown that the control method is very effective and reduces integrated absolute error significantly. Experimentation with obstacle-laden environments shows that the method performs even better under high maneu-

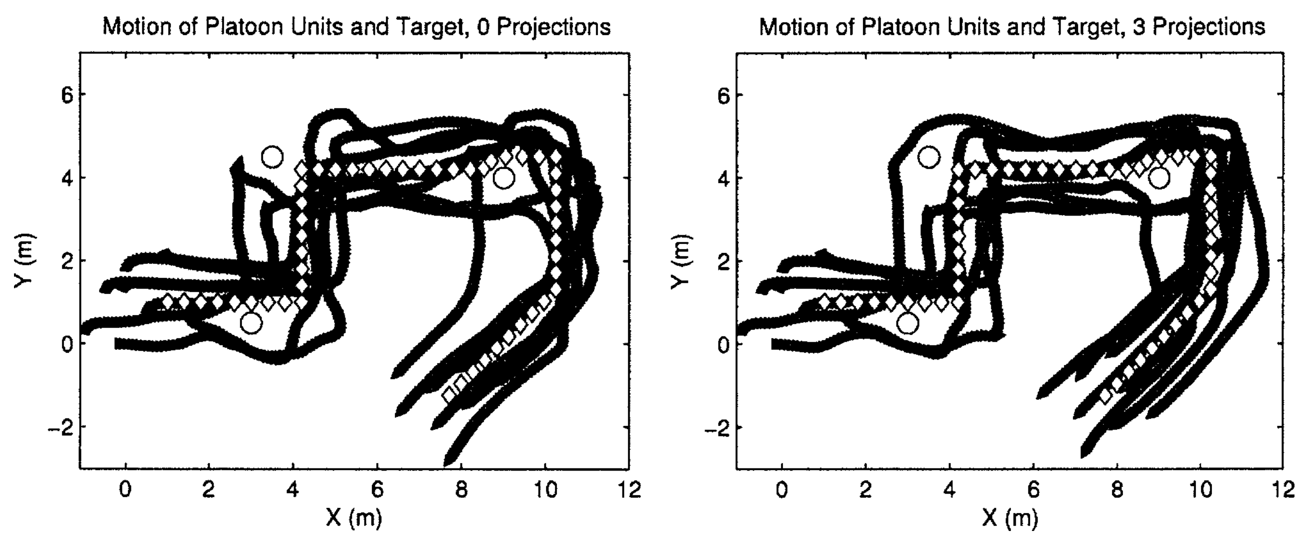

Fig. 5 Platoon trajectories with zero and three error projections. The diamonds are the moving target, the triangles are the units, and circles represent obstacles.

vering requirements, as shown in Fig. 5, which includes obstacles and random initial heading for the platoon units. For this example, the optimal number of iterations is three, with integrated absolute error reduced by 53.6% for mean position, 71.7% for σ_x^2 and 46% for σ_y^2.

5 Conclusions

In this paper, we have presented a novel method for control of platoons of cooperating nonholonomic vehicles representative of a broad class of real systems. An error vector is computed between the nominal desired unit velocities and the achievable velocities for the system based on a steering angle limit of $-\pi/4 \leq \alpha \leq \pi/4$ and an absolute speed limit of 1.0 m/s. The velocity error is iteratively projected onto the null space of the Jacobian in order to find a suitable new desired velocity that achieves regulation of the platoon-level function $f(q)$ and matches, as well as possible, the achievable velocity. Errors are recomputed and reprojected until the desired velocity is within achievable limits or a preset number of iterations has been reached. The approach presented herein is applicable to autonomous surface vessels [14], underwater vehicles [15], and even unmanned aerial vehicles.

References

[1] Bishop, B. E., 2003, "On the Use of Redundant Manipulator Techniques for Control of Platoons of Cooperating Robotic Vehicles," IEEE Trans. Syst. Man Cybern., **33**(5), pp. 608–615.

[2] Balch, T., and Arkin, R., 1994, "Communication in Reactive Multiagent Robotic Systems," Auton. Rob., **1**(1), pp. 27–52.

[3] Tanner, H. G., Loizou, S. G., and Kyriakopoulos, K. J., 2003, "Nonholonomic Navigation and Control of Cooperating Mobile Manipulators," IEEE Trans. Rob. Autom., **19**(1), pp. 53–64.

[4] Desai, J. P., Ostrowski, J. P., and Kumar, V., 2001, "Modeling and Control of Formations of Nonholonomic Mobile Robots," IEEE Trans. Rob. Autom., **17**(6), pp. 905–908.

[5] Siciliano, B., 1990, "Kinematic Control of Redundant Robot Manipulators: A Tutorial," J. Intell. Robotic Syst., **3**, pp. 201–212.

[6] Samson, C., Le Borgne, M., and Espiau, B., 1991, *Robot Control: The Task Function Approach*, The Oxford Engineering Science Series, Clarendon Press, Oxford.

[7] Tan, Y.-C., and Bishop, B. E., 2004, "Evaluation of Robot Swarm Control Methods for Underwater Mine Countermeasures," *Proc. 36th Annual Southeastern Symposium on Systems Theory*, IEEE Press, Piscataway, March, pp. 294–298.

[8] Stilwell, D. J., and Bishop, B. E., 2002, "Redundant Manipulator Techniques for Path Planning and Control of a Platoon of Autonomous Vehicles," *Proc. of the IEEE Conference on Decision and Control*, IEEE Press, Piscataway, December, pp. 2093–2098.

[9] Pant, A., Seiler, P., and Hedrick, K., 2002, "Mesh Stability of Look-Ahead Interconnected Systems," IEEE Trans. Autom. Control, **47**(2), pp. 403–407.

[10] Stipanovic, D. M., Inalhan, G., Teo, R., and Tomlin, C. J., 2004, "Decentralized Overlapping Control of a Formation of Unmanned Aerial Vehicles," Automatica, **40**(8), pp. 1285–1296.

[11] Antonelli, G., and Chiaverini, S., 2003, "Obstacle Avoidance for a Platoon of Autonomous Underwater Vehicles," *6th IFAC Conf. on Manoeuvring and Control of Marine Craft*, pp. 143–148.

[12] Latombe, J.-C., 1990, *Robot Motion Planning*, Springer-Verlag, Berlin.

[13] Arkin, R., 1998, *Behavior-Based Robotics*, The MIT Press, Cambridge, MA.

[14] Bishop, B. E., 2004, "Design and Control of Platoons of Cooperating Autonomous Surface Vessels," *Proc. 7th Annual Maritime Transportation System Research and Technology Coordination Conference*, Transportation Research Board of the National Academies, Washington, D.C., November, pp. 1–11, http://www.trb.org/Conferences/MTS/Program.asp, http://www.trb.org/Conferences/MTS/lA%20Bishop%20Paper.pdf

[15] Stilwell, D. J., and Bishop, B. E., 2000, "Platoons of Underwater Vehicles: Communication, Feedback and Decentralized Control," IEEE Control Syst. Mag., **20**(6), pp. 45–52.