THE OSU MULTI-VEHICLE COORDINATION TESTBED

Rafael Fierro and EzzAldeen Edwan

MARHES Laboratory, School of Electrical and Computer Engineering
Oklahoma State University, Stillwater, OK 74078, USA
{rfierro, ezzalde}@okstate.edu

ABSTRACT

In this paper we present the OSU Multi-Vehicle Coordination Testbed, a group of nonholonomic robots for testing and implementing hybrid control approaches for formation control, multi-agent coordination, multi-robot learning, and real-time wireless networked control system design. The vehicles are equipped with a suite of sensors, a vision system and wireless Ethernet for communications. The testbed will allow us to carry out experimental research to verify theoretical results that have been validated using only simulation. In addition, we describe a framework for adaptive leader-following formation control. The novel aspect of this research is that we explicitly consider the dynamics of the vehicles and develop a *modular* adaptive formation control scheme.

1. INTRODUCTION

In recent years we have witnessed a revolution in information, wireless communication, and sensor technologies. These advances present unprecedented challenges and new opportunities for the years to come. We now have the resources necessary to develop truly intelligent systems. Multiple, autonomous agents that interact cooperatively in unknown, dynamic environments can realize such systems. These networked autonomous agents (*i.e.*, vehicles) will be able to navigate in space, build coherent maps of an unstructured environment, identify and retrieve targets, and gather information with limited human intervention. To realize this vision, multi-agent systems require mathematical tools that extend beyond classical hybrid systems design, and experimental platforms where new control algorithms can be implemented on.

We are interested in tasks that include exploration [1], surveillance [2], search and rescue operations [3], and mapping of unknown or partially known environments [4]. In all these applications, there is a need to have the robots estimate their relative positions and orientations with respect to their neighbors and maintain a desired formation. The experimental testbed presented here will allow to develop, implement and evaluate novel coordination control algorithms.

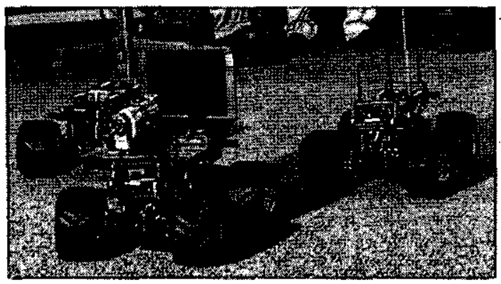

Figure 1: The mobile robot experimental platform.

This paper is organized as follows. First, in section 2, the mobile robot hardware is described in some detail. The software architecture is outlined in section 3. Next, section 4 presents an adaptive formation control using two robots in a leader-following configuration. Finally, some concluding remarks and directions for future work are provided in section 5.

2. THE MOBILE ROBOT

The MARHES Laboratory X-treme robots (see Figure 1) are based upon a commercially available radio control truck from Tamiya Inc., with significant modifications.

The robot has a servo controller for steering and a PWM speed controller for forward/backward motion. An on-board notebook computer provides the computational power for signal/image processing, motion control and IEEE 802.11b wireless networking. Also, a PCMCIA Multi-function I/O card from National Instruments is used for interfacing the computer with a suite of analog and digital sensors. The suite of sensors includes IR distance sensors, accelerometers, odometer, and an omnidirectional camera described next.

2.1. Omnicam Sensor

The platform uses an omnidirectional camera from Remote Reality^{TM} as its vision sensor. One of the primary advan-

0-7803-7523-8/02/$17.00 ©2002 IEEE

tages of these catadioptric sensors is that they afford a single effective point of projection. This means that after an appropriate calibration, every point in the omnidirectional image can be associated with a unique ray through the focal point of the camera. This allows azimuth and elevation angles to every teammate (and target) visible in the $360°$ *field-of-view* image to be estimated, making the camera an ideal choice for cooperative sensing, exploration and mapping tasks. A target or other robots can be identified in the image using a YUV based color feature extractor which provides robustness to variations in illumination [5]. See Figure 2.

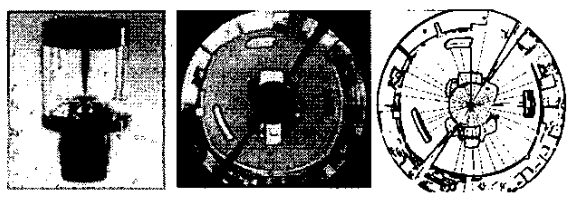

Figure 2: Omnicam (left), typical image (center) and the ensuing range map (right).

3. SOFTWARE ARCHITECTURE

The experimental testbed described above would be useless without a well defined software architecture. In this section, we briefly outline our architecture for coordination and control of multiple mobile robots. We are motivated by theoretical ideas and modern tools in software engineering, and the emerging theory of hybrid systems.[1]

The software for each robot consists of components called *agents*. Agents operate concurrently and lend themselves to *parallel composition*. Each agent consists of a main *mode*, that may in turn have *submodes*. A mode is a discrete state characterized by a set of differential equations and constraints, which generally involves a feedback control law. The concept of modes allows us to formally define the notion of sequential composition. Definition of submodes implicitly define the notion of *hierarchical composition.*

The basis for the software framework and formal definitions of composition are provided in [7]. Here, all robots run the same software. It includes many *estimator* agents operating concurrently and *controller* agents operating sequentially. Well-defined input and output ports describe the exchange of information between the agents. Our multi-threaded software architecture encapsulates algorithms and data in the usual object-oriented manner together with control of a thread within which the algorithms will execute, and a number of events that allow communication with other objects. At the top of the hierarchy, *coordinator* objects are able to control the execution of the lower level objects to service high-level goals. To offer platform independence, only the lowest level objects should be specific to any hardware, and these should have a consistent interface for communication with the coordinator objects that control their execution.

Control algorithms are currently being incorporated into the multi-threaded software architecture for such basic functionality as obstacle avoidance, wall-following, mapping and localization, and group behavior. Examples of group behavior may include formation keeping, collaborative mapping and exploration, and cooperative manipulation to mention just a few.

[1] A hybrid system here refers to a collection of digital programs that interact with each other in a physical world that is analog in nature [6].

4. A FRAMEWORK FOR ADAPTIVE FORMATION CONTROL

In this section, we describe an adaptive leader-following control algorithm. In our previous work [8], we developed stable control algorithms only considering the kinematic model of the robot; that is, we did not consider the dynamics of the platform. We assumed an *ideal velocity controlled* vehicle which is an unrealistic assumption. In this paper, we would like to improve the performance of the formation control algorithms by considering the robot dynamics. In practice, parameters such as mass, inertia, and friction are partially known; therefore adaptive techniques [9] can be used to learn the unknown dynamics of the vehicle *on the fly.*

The kinematic formation control laws were motivated by ideas from the well established area of input-output feedback linearization [10]. The kinematics of the i^{th} robot can abstracted by a unicycle model:

$$\dot{x}_i = v_i \cos\theta_i, \quad \dot{y}_i = v_i \sin\theta_i, \quad \dot{\theta}_i = \omega_i, \qquad (1)$$

where $q_i \equiv (x_i,\ y_i,\ \theta_i) \in SE(2)$, and v_i and ω_i are the linear and angular velocities, respectively.

We consider a subgroup of two robots shown in Figure 3. First, we describe a *kinematic* controller for leader–following, adopted from [11]. Then, we present an adaptive controller that takes into account the unknown dynamics of the follower vehicle. The lead robot is assumed to follow a trajectory $g(t)$ provided by a higher-level planner [12].

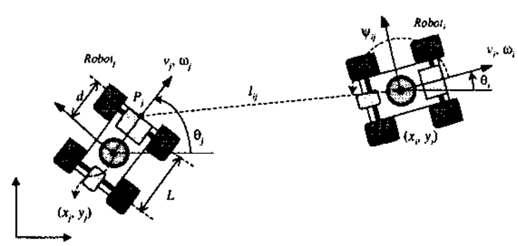

Figure 3: Two robots in a leader–following configuration.

By using this basic leader-following controller (denoted $SB_{ij}C$ here), robot R_j follows R_i with a desired *S*eparation l_{ij}^d and desired relative *B*earing ψ_{ij}^d. Note that this relative bearing describes the heading direction of the follower with respect to the leader. The two-robot system is transformed into a new set of coordinates where the state of the leader is treated as an exogenous input. Thus the kinematic equations are given by

$$\dot{\boldsymbol{z}}_{ij} = \boldsymbol{G}_1(\boldsymbol{z}_{ij}, \beta_{ij})\boldsymbol{u}_j + \boldsymbol{F}_1(\boldsymbol{z}_{ij})\boldsymbol{u}_i, \quad \dot{\beta}_{ij} = \omega_i - \omega_j, \tag{2}$$

where $\boldsymbol{z}_{ij} = [l_{ij} \quad \psi_{ij}]^T$ is the system output, $\beta_{ij} = \theta_i - \theta_j$ is the relative orientation, $\boldsymbol{u}_j = [v_j \quad \omega_j]^T$ is the input for R_j, $\boldsymbol{u}_i = [v_i \quad \omega_i]^T$ is R_i's input, and

$$\begin{aligned} \boldsymbol{G}_1 &= \begin{pmatrix} \cos\gamma_{ij} & d\sin\gamma_{ij} \\ \frac{-\sin\gamma_{ij}}{l_{ij}} & \frac{d\cos\gamma_{ij}}{l_{ij}} \end{pmatrix} \\ \boldsymbol{F}_1 &= \begin{pmatrix} -\cos\psi_{ij} & 0 \\ \frac{\sin\psi_{ij}}{l_{ij}} & -1 \end{pmatrix} \\ \gamma_{ij} &= \beta_{ij} + \psi_{ij}. \end{aligned}$$

By applying input-output feedback linearization, the control velocities for the *follower* are given by

$$v_j = k_1\tilde{l_{ij}}\cos\gamma_{ij} - l_{ij}\sin\gamma_{ij}(k_2\tilde{\psi_{ij}} + \omega_i) + v_i\cos\beta_{ij} \tag{3}$$

$$\omega_j = \frac{1}{d}[k_1\tilde{l_{ij}}\sin\gamma_{ij} + l_{ij}\cos\gamma_{ij}(k_2\tilde{\psi_{ij}} + \omega_i) + v_i\sin\beta_{ij}] \tag{4}$$

where

$$\tilde{l_{ij}} = l_{ij}^d - l_{ij}, \quad \tilde{\psi_{ij}} = \psi_{ij}^d - \psi_{ij},$$

d is the offset to an off-axis *reference* point P_j on the robot, and $k_1,\ k_2 > 0$ are the user-selected controller gains.

The closed-loop linearized system becomes

$$\dot{l}_{ij} = k_1\tilde{l_{ij}}, \quad \dot{\psi}_{ij} = k_2\tilde{\psi_{ij}}, \quad \dot{\beta}_j = \omega_i - \omega_j. \tag{5}$$

In [13], we proved that under suitable assumptions on the motion of the lead robot, the closed-loop system is stable; that is, the output vector $\boldsymbol{z}_{ij}$ converges to the desired value z_{ij}^d arbitrarily fast and the internal *dynamics* of the robot, *i.e.*, the relative orientation β_{ij}, are bounded.

4.1. Controller for the Dynamic Model

It is well-known in the literature that a mobile robot (*i.e.*, the follower R_j) system having an n-dimensional configuration space $\mathcal{C}$ with generalized coordinates $\boldsymbol{q}_j \in \mathbb{R}^n$, and subject to m constraints can be described by

$$\begin{aligned} \dot{\boldsymbol{q}}_j &= \boldsymbol{S}(\boldsymbol{q}_j)\boldsymbol{u}_j \\ \bar{\boldsymbol{M}}(\boldsymbol{q}_j)\dot{\boldsymbol{u}}_j + \bar{\boldsymbol{V}}_m(\boldsymbol{q}_j, \dot{\boldsymbol{q}}_j)\boldsymbol{u}_j + \bar{\boldsymbol{F}}(\boldsymbol{u}_j) &= \bar{\boldsymbol{B}}(\boldsymbol{q}_j)\boldsymbol{\tau}_j \end{aligned} \tag{6}$$

where, in our case $n = 3$, $m = 1$, $\boldsymbol{u}_j(t) \in \mathbb{R}^2$ is a velocity vector, $\bar{\boldsymbol{M}} \in \mathbb{R}^{2\times 2}$ is a symmetric, positive definite inertia matrix, $\bar{\boldsymbol{V}}_m \in \mathbb{R}^{2\times 2}$ is the centripetal and coriolis matrix, $\bar{\boldsymbol{F}} \in \mathbb{R}^{2\times 1}$ is the surface friction, $\boldsymbol{\tau} \in \mathbb{R}^{2\times 1}$ is the input vector, and $\bar{\boldsymbol{B}} \in \mathbb{R}^{2\times 2}$ is a nonsingular matrix that depends on the steering system of the mobile platform. Finally, $\boldsymbol{S} \in \mathbb{R}^{3\times 2}$ is a Jacobian matrix that transforms velocities in mobile base coordinates to velocities in Cartesian coordinates.

By using the standard regressor formulation, the robot dynamics given in (6) can be expressed as follows

$$\bar{\boldsymbol{M}}\dot{\boldsymbol{u}}_j + \bar{\boldsymbol{V}}_m\boldsymbol{u}_j + \bar{\boldsymbol{F}} = \bar{\boldsymbol{B}}\boldsymbol{\tau}_j = \boldsymbol{Y}(\dot{\boldsymbol{u}}_j, \boldsymbol{u}_j)\boldsymbol{p}_j \tag{7}$$

where $\boldsymbol{p}_j \in \mathbb{R}^r$ is a vector consisting of the unknown *constant* robot parameters (*i.e.*, mass, inertia, friction and so forth), and $\boldsymbol{Y} \in \mathbb{R}^{2\times r}$ is the regression matrix.

To incorporate the dynamic model, we will employ the backstepping technique as in [14]. This formulation allows us to design the adaptive formation controller in a modular fashion. Let us define the *velocity tracking error* as

$$\tilde{\boldsymbol{u}}_j = \boldsymbol{u}_j - \boldsymbol{u}_j^a \tag{8}$$

where $\boldsymbol{u}_j$ is the *desired* velocity vector given in (3) and (4) that produces a stable leader-following closed-loop system. Moreover, $\boldsymbol{u}_j$ is differentiable provided the leader's velocity vector $\boldsymbol{u}_i$ is differentiable. $\boldsymbol{u}_j^a$ is the *actual* velocity vector of the vehicle. Thus, the dynamic control objective is to design $\boldsymbol{\tau}_j$ in (6) such that $\|\tilde{\boldsymbol{u}}_j\| \to 0$ as $t \to \infty$.

It is not difficult to show that

$$\bar{\boldsymbol{M}}\dot{\tilde{\boldsymbol{u}}}_j = \bar{\boldsymbol{B}}\boldsymbol{\tau}_j - \bar{\boldsymbol{V}}_m\tilde{\boldsymbol{u}}_j - \bar{\boldsymbol{F}} - \boldsymbol{Y}_c\boldsymbol{p}_j \tag{9}$$

where $\boldsymbol{Y}_c(\dot{\boldsymbol{u}}_j, \boldsymbol{u}_j, \boldsymbol{u}_j^a)$ is the regression matrix defined in (7) and $\dot{\boldsymbol{u}}_j \in \mathbb{R}^2$ represents the time derivative of the desired velocity $\boldsymbol{u}_j$. By using Lyapunov theory and Barbalat's lemma, it can be shown that the following controller stabilizes the closed-loop system

$$\begin{aligned} \boldsymbol{\tau}_j &= \bar{\boldsymbol{B}}^{-1}(-\boldsymbol{K}_d\dot{\tilde{\boldsymbol{u}}}_j + \boldsymbol{Y}_c\hat{\boldsymbol{p}}_j) & (10) \\ \dot{\hat{\boldsymbol{p}}}_j &= -\boldsymbol{\Gamma}\boldsymbol{Y}_c^T\tilde{\boldsymbol{u}}_j & (11) \end{aligned}$$

where $\boldsymbol{K}_d, \boldsymbol{\Gamma}$ are symmetric and positive definite matrices and $\hat{\boldsymbol{p}}_j$ is the estimation of the unknown parameter vector $\boldsymbol{p}_j$.

Simulation results are depicted in Figures 4 - 6. As it can bee seen, the adaptive controller successfully accomplishes the formation control task. Velocity and formation errors asymptotically converge to zero, and the parameter estimation vector remains bounded.

5. CONCLUSIONS

In this paper, we have presented an experimental platform to support research for multi-vehicle coordination, formation

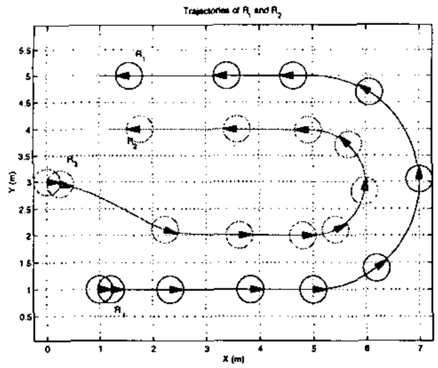

Figure 4: Leader-following adaptive formation control.

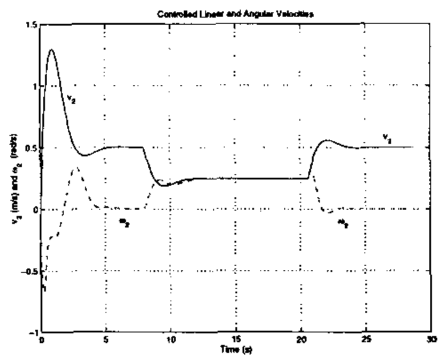

Figure 5: Controlled velocities for the follower robot.

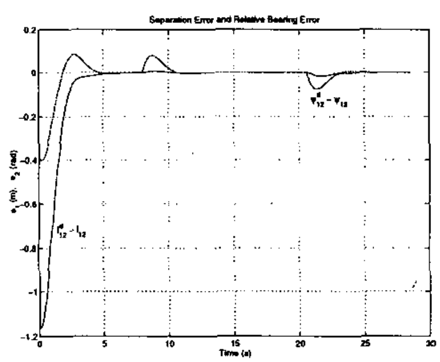

Figure 6: Separation and bearing errors.

control and wireless interconnected nonlinear systems. We have also described a framework for modular adaptive formation control. Experimental results only considering the kinematic controller have been reported elsewhere. Currently, we are implementing the adaptive dynamic formation controller presented herein, and extending previous experiments to more complex scenarios where robots need to exhibit a variety of behaviors such as localization, target acquisition, collaborative mapping and formation keeping.

6. REFERENCES

[1] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative multi-robot exploration," in *Proc. IEEE Int. Conf. Robot. Automat.*, San Francisco, CA, April 2000, pp. 476–481.

[2] J. Feddema and D. Schoenwald, "Decentralized control of cooperative robotic vehicles," in *Proc. SPIE Vol. 4364, Aerosense*, Orlando, FL, April 2001.

[3] J. S. Jennings, G. Whelan, and W. F. Evans, "Cooperative search and rescue with a team of mobile robots," in *Proc. IEEE Int. Conf. on Advanced Robotics (ICAR)*, Monterey, CA, July 1997, pp. 193–200.

[4] John W. Fenwick, Paul M. Newman, and John J. Leonard, "Cooperative concurrent mapping and localization," in *Proc. IEEE Int. Conf. Robot. Automat.*, Washington, D.C., May 2002, pp. 1810–1817.

[5] J. Spletzer, A. Das, R. Fierro, C. J. Taylor, V. Kumar, and J. P. Ostrowski, "Cooperative localization and control for multi-robot manipulation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Maui, Hawaii, Oct. 2001, pp. 631–636.

[6] A. van der Schaft and H. Schumacher, *An Introduction to Hybrid Dynamical Systems*, vol. 251 of *Lecture Notes in Control and Information Sciences*, Springer, London, 2000.

[7] R. Alur, A. Das, J. Esposito, R. Fierro, Y. Hur, G. Grudic, V. Kumar, I. Lee, J. P. Ostrowski, G. Pappas, J. Southall, J. Spletzer, and C. J. Taylor, "A framework and architecture for multirobot coordination," in *Experimental Robotics VII*, D. Rus and S. Singh, Eds., pp. 303–312. Springer Verlag, 2001.

[8] A. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, "A framework for vision based formation control," *Multi-Robot Systems: A Special Issue of IEEE Trans. Robot. Automat.*, October 2002, Accepted for publication.

[9] T. Fukao, H. Nakagawa, and N. Adachi, "Adaptive tracking control of a nonholonomic mobile robot," *IEEE Trans. on Robotics and Automation*, vol. 16, no. 5, pp. 609–615, October 2000.

[10] A. Isidori, *Nonlinear Control Systems*, Springer-Verlag, London, 3rd edition, 1995.

[11] J. Desai, J. P. Ostrowski, and V. Kumar, "Controlling formations of multiple mobile robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, Leuven, Belgium, May 1998, pp. 2864–2869.

[12] R. Fierro, P. Song, A. K. Das, and V. Kumar, "Cooperative control of robot formations," in *Cooperative Control and Optimization*, R. Murphey and P. Pardalos, Eds., vol. 66 of *Applied Optimization*, chapter 5, pp. 73–93. Kluwer Academic Press, Dordrecht, 2002.

[13] R. Fierro, A. Das, V. Kumar, and J. P. Ostrowski, "Hybrid control of formations of robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, Seoul, Korea, May 2001, pp. 157–162.

[14] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot: Backstepping kinematics into dynamics," *J. Robotic Systems*, vol. 14, no. 3, pp. 149–163, 1997.