Flocking, Formation Control and Path Following for a Group of Mobile Robots

Luis Valbuena R and Herbert G. Tanner

Abstract

The paper presents a multi-functional control law for a group of mobile robots. The controller concurrently integrates flocking, formation control and path following, as basic components in a navigation framework for mobile sensor networks tasked with searching and harvesting information. The convergence of the closed loop system is ensured using Ryan's invariance principle; the particular tool being necessary due to the use of a nonsmooth artificial potential field for formation control. Simulations and experiments corroborate the theoretical convergence results.

Index Terms

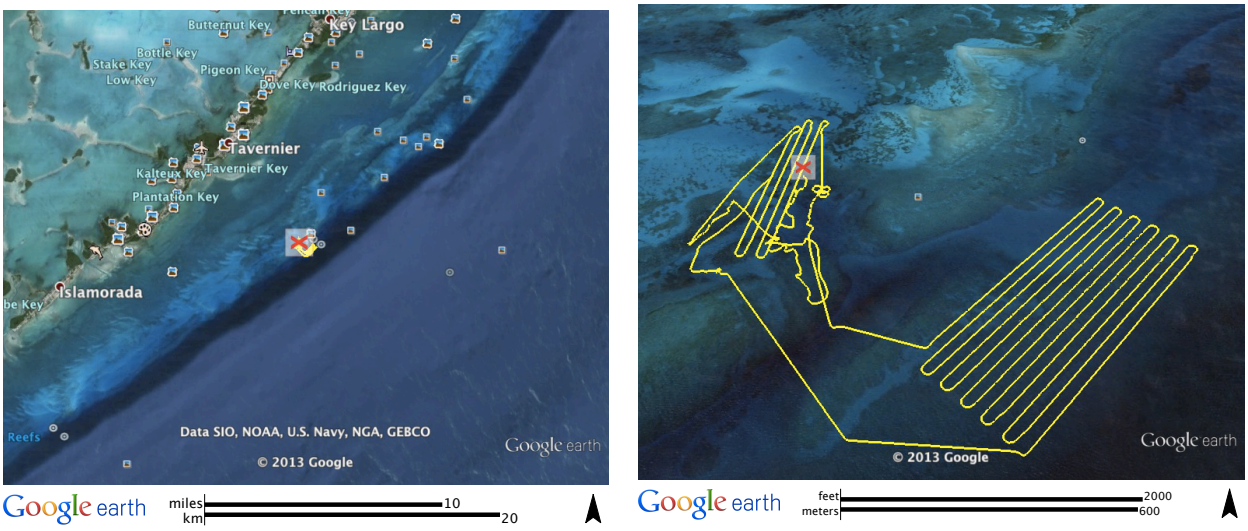
Flocking; formation control; path following.

I. Introduction

In the spring of 2011 an Autonomous Underwater Vehicle (AUV) from the CSHEL lab of the University of Delaware conducted sea bed surveys for habitat mapping in the Conch Reef, Florida. The vehicle used an on-board camera to create a mosaic of the ocean floor, for the purpose of—among other science objectives—characterizing coral growth. Figure 1 shows a Google Earth view of the location of that dive, and a more detailed aspect of the path that the underwater vehicle followed during this 4-hour survey. From the perspective of this paper, what is interesting in this figure is (a) the scale of the overall area coverage achieved with a single AUV, and (b) the particular pattern that oceanographers use to map the ocean floor using robotic vehicles.

Area coverage depends on many parameters—for example, field of sensor view and vehicle speed—but with operational limitations on sensors and speed, battery life is the main constraint. The number of vehicles deployed therefore matters. As this number increases, either the size of the surveyed area increases or the time to mission completion decreases. For the particular application, mission time is also of essence, because the the surface vessel and associated personnel necessary to transport and deploy the robotic vehicle itself come with significant financial overhead. For a 50 feet surface support vessel, it makes little difference if one or ten robots are being deployed. For certain applications, therefore, currently served by single

L. Valbuena is with the Department of Mechanical Engineering, University of New Mexico, Albuquerque, NM, 87110 USA e-mail: (lavalbuenar@unm.edu).
Dr. Tanner is with the Department of Mechanical Engineering, University of Delaware, Newark, DE, 19717 USA e-mail: (btanner@udel.edu).
Manuscript received DATE; revised DATE.



(a) Area covered (b) Pattern followed

Fig. 1. Views of coverage of a marine survey mission using an autonomous underwater vehicle in Conch Reef, Florida. X marks the location of the initial robot deployment. (Courtesy of Dr. Art Trembanis)

autonomous robotic vehicles—like searching for plane wreckage and black boxes—scaling up to small groups makes sense in terms of satisfying the objectives of the mission and managing its cost.

The choice of the coverage pattern, on the other hand, is mainly empirical. From Fig. 1(b) it appears that AUV practitioners prefer a heuristic "lawn-mower's problem" solution [1]—which is in general NP-hard [2]— for their mission planning. In the particular case of the AUV participating in the mission of Fig. 1, path planning is done by defining way-points. Irrespectively of how the pattern to be followed is generated, if multiple vehicles are to be used in a mission, planning is simpler if the collection of vehicles moves as a coherent stable formation, compared to having each vehicle assigned its own individual lot (fewer waypoints). Thus, formation control algorithms that enable groupsof autonomous vehicles to follow designated paths can be useful in mission planning and execution, both in the particular motivating application, as well as in a variety of other problem domains where robots are used for search and mapping: space exploration, search and rescue, environmental monitoring, surveillance, etc.

This paper aims at addressing this need. It presents a control architecture that coordinates a group of vehicles to fall into formation of specified shape, flock, and follow a predefined path, all at the same time through coexisting controllers that are implemented by concurrently evolving software processes. While centralized in principle, the architecture can be implemented in a decentralized way, depending on the nature of the graph that captures the communication topology between the vehicles.

The controllers reported in this paper take into account the nonholonomic constraints that vehicles can be subject to, based on standard input-output feedback linearization techniques [3], [4]. The theoretical convergence analysis of the formation controller builds on results from non-smooth analysis, since the potential function used for formation control is nondifferentiable. In this context, one of the paper's technical contributions is the first formal proof of stability for a combined flocking and a provably convergent formation controller for second-order discontinuous dynamical systems. The path following controller sits at a higher level, and drives the whole group, without switching, along paths composed of straight line segments and Bézier curves, taking

advantage of the stability properties provided by the flocking and formation controllers. The three controllers are integrated seamlessly and work concurrently in a way that no leaders (actual or artificial) need to be used.

Individually, the problems of formation control, path following and flocking have been studied extensively. We can also find work addressing instances that combine two of the three. In formation control there is a multitude of different working assumptions about the capabilities of the agents, the range of the sensors, and the information structure used to embed the formation specification. For instance, in some work [4]–[8], it is not entirely clear if particular cases such as non-degenerate equilibrium formations can be addressed. Other approaches [4], [5], [9]–[11] avoid switching or deliberatively smoothen it. In a significant amount of work on formation control [4], [5], [8], [9], [11]–[20] vehicles are assumed holonomic or their final orientation is ignored. The implication of this is that if the group attempts to move as a rigid body from the final configuration, the formation will temporarilly dissolve.

Not all nonholonomic formation controllers explicitly guarantee collision avoidance, but some do [21]–[23]. At steady state, control designs in the aforementioned work may in principle allow the group to move as a cohesive unit along a given direction, although the time-varying nature of the formation's destination points can introduce steady-state errors. When formation controllers for nonholonomic vehicles are explicitly designed so that the formation is steered along a desired path or trajectory, two general approaches are followed. Either a specific point on the formation—this could be an actual leading vehicle, a virtual vehicle, or an arbitrary reference point such as the centroid [24]—is made to track a designated trajectory [7], [25]–[30], or the path specification is broken down to individual reference trajectories in the spirit of virtual structures [1], [18], [31], [32]. In some sense, these two alternatives represent the two ends of the spectrum, the former giving some vehicles authority over the rest imposing a hierarchy, and the latter implementing a flat organization where everyone has their own motion specification. These motion specifications involve velocity directives.Yet, there may not necessarily be a need to match a particular speed reference, like in the motivating underwater survey application mentioned earlier.

Although some links between provably convergent formation control and flocking have been identified [33], the two behaviors have not been integrated into a single control design. Among the different flocking control approaches that exist, there are some [34], [35] that do not offer a formal proof that formation stabilization will indeed occur. Out of many others that do (e.g., [36]–[39]) few consider switching [40] or nonholonomic constraints [41]–[43]. And although switching and kinematic constraints may not constitute insurmountable obstacles in implementing flocking controllers, these controllers in general cannot guarantee that any desired and oriented shape will be achieved (cf. [18], [24]).

In terms of cooperative path following, part of the existing work involves control specifications that include velocity references [32], [44]–[46], and thus falls into the same category as those formation controllers that are based on virtual structures and virtual leaders. Other approaches do not [47], [48], but may not always be accompanied with convergence proofs [49] (cf. [50]). What is interesting in path following approaches based on vector fields in particular [48], [51], [52] is that the method may not be limited to paths of a certain geometry, (cf. [46], [47]). This vector-field approach resorts to a switching mechanism when it comes to selecting a field specific to the closest path segment, in order to avoid sinks, dead zones and singularities. Switching also appears in solutions that involve triangular decompositions of a polygonal "corridor" around the desired path [53], [54], (cf. [55], [56]) although this switching is limited to the corridor's boundaries. Related approaches [51] alleviate the

switching problem but they do not appear to be applicable to paths that are not closed curves.

At this time, there exist enough control technology know-how to offer a multitude of solutions to the problem of driving a formation of nonholonomic vehicles along given geometric paths. Strictly on an application front, therefore, this paper's contribution is in the context of specialized methodological improvements. However, the true value of the reported analysis is on the theoretical side, because it shows that provably convergent flocking, formation control, and path following can be integrated in a single control architecture, and that all the component behaviors can coexist, run at the same time, and be active concurrently without adversely interfering with each other. In fact, the component behaviors exhibit a synergy which has not been observed before: flocking, formation, and path following controllers work together by reducing near-collision instances, destabilizing the singularities of the formation controller, and enhancing the cohesion of the group. What is more, the combination increases robustness even if one of the three controller components goes off-line at steady state, the other two can be sufficient in keeping the system within its control specification.

This paper is organized as follows: Section II describes formally the problem addressed in this paper. Section III describes the development of the first component of the control architecture, which is the formation controller, while Section IV is devoted to path following. Then, Section V shows how formation control and flocking can be integrated. In Section VI simulation and experimental results from the implementation of this architecture on a group of wheeled mobile robots are reported. Section VII summarizes the paper. To make the paper more accessible, some of the more technical discussions have been collected in an Appendix.

II. Problem Statement

A group of robots, initialized at random positions and orientations, is required to fall into a specified formation without colliding, and continue moving together as a rigid body following a prescribed path while maintaining the formation they have fallen into. We assume that the robots move in an obstacle-free environment.

Denoting x_i and y_i the position coordinates of robot i, and φ_i its orientation, we assume that its kinematics is captured by the equations of the unicycle:[1]

$$\dot{\mathrm{x}}_i = v_i \cos\varphi_i \tag{1a}$$

$$\dot{\mathrm{y}}_i = v_i \sin\varphi_i \tag{1b}$$

$$\dot{\varphi}_i = \omega_i \ . \tag{1c}$$

Given a small parameter $L > 0$, this kinematics can be input-output feedback linearized using as output the vector

$$x_{2i} \triangleq \begin{bmatrix} \mathrm{x}_i + L\cos\theta_i \\ \mathrm{y}_i + L\sin\theta_i \end{bmatrix} ,$$

[1] Slightly different models for ground mobile robots have appeared in the literature [57]–[59].

leading to a feedback linearized dynamical system of a double integrator with a state formed by stacking x_{2i} on $x_{1i} = (v_i^\intercal, \omega_i^\intercal)^\intercal$,

$$\dot{x}_{2i} = x_{1i} \ , \qquad \dot{x}_{1i} = \begin{bmatrix} \Gamma_i \\ \Omega_i \end{bmatrix} \ , \tag{2}$$

and with x_{1i} driven by Γ_i, Ω_i, which are the transformed inputs of the feedback-linearized system. Let d_0 be the distance between two robot centers at which the robots come in contact with each other. Then to avoid collisions between the robots, one must ensure that $\|x_{2i} - x_{2j}\| \geq d_0$ for all $i \neq j$ as i and j range in $\{1, \ldots, N\}$. Stack now all robot outputs in the form of a combined vector

$$x \triangleq [x_1^\intercal \quad x_2^\intercal]^\intercal; \quad x_1 \triangleq [x_{11} \ \cdots \ x_{1i} \ \cdots \ x_{1n}]^\intercal \ , x_2 \triangleq [x_{21} \ \cdots \ x_{2i} \ \cdots \ x_{2n}]^\intercal \ ,$$

and let the desired formation for this group of robots be described in terms of their feedback linearized coordinates x by means of a labeled directed graph called the *formation graph* $G = \{V, E, C\}$:

Definition 1 ([60]). *$G = \{V, E, C\}$ is a labeled directed graph consisted of:*

- *A set of vertices $V = \{v_1 \ldots v_n\}$, with each vertex associated to one robot in the group.*
- *a set of edges $E = \{(i, j) : \{1 \ldots n\} \times \{1 \ldots n\}\}$ containing ordered pairs of nodes that denote the existence of pairwise position specifications between robots, and*
- *a set of labels $C = [c_{ij}]$ for $(i, j) \in E$, that explicitly express quantitatively the relative position specification for each pair of robots.*

The formation graph $G = \{V, E, C\}$ expresses the pairwise relative position requirements between the robots in the set E, and codifies these requirements in an array C of desired relative position *vectors* q_d.

Assume the formation graph G is connected, and consider the incidence matrix[2] B of the complete graph associated with the nodes of G, As heads and tails of the edges are marked with 1 and -1 respectively, we can multiply the concatenated vector of robots position x_2 to get relative positioning (cf. [40])

$$q = Bx_2 \ . \tag{3}$$

Thus q is the stack vector of *all* relative position vectors between the robots in the group. Since the formation graph G is connected, not all the elements of q need to be actively controlled; it suffices to concentrate on a subset of the elements of q that correspond to edges in a spanning tree of G. For once the relative position vectors associated with edges in a spanning tree of G are fixed, all the remaining position vectors are uniquely determined.

Let us thus assume that G is a spanning tree, and that the stack vector c formed by the elements of array C (basically converting the array into a vector), after possibly some renumbering of nodes, is

$$c = [q_{d1}^\intercal \quad q_{d2}^\intercal \ \cdots \ q_{d(n-1)} \quad q_{d(n-1)}^\intercal]^\intercal \ .$$

[2]The incidence matrix B of a directed graph is a $m \times n$ matrix with $m = |V|$ and $m = |E|$ such that $b_{ij} = 1$ if edge j arrive to node i, $b_{ij} = -1$ if edge j leaves node i, and $b_{ij} = 0$ if node i is not associated with edge j.

Then the problem addressed in this paper can be formulated as follows:

Problem Statement. *For a group of n points in $\mathbb{R}^2$ with double integrator dynamics (2) the states of which are stacked in a vector x, we seek a global control law that makes all these moving points fall into a formation codified in vector c, while ensuring that for all $i \neq j$ and $t > 0$, $\|x_{1i}(t) - x_{1j}(t)\| \geq d_0$, that the formation shape is maintained, and that the group moves along a prescribed path composed by straight line segments and Bézier curves.*

III. Formation Controller Design

The solution to formation control given in this paper follows the lines of a particular multi-agent navigation function approach [60], that applies to coordinating a group of N disc-shaped robots into an arbitrary geometric formation. If we define $\gamma(q)$ to be a positive definite function that encodes the proximity of the system to its desired configuration, and let $\beta(q)$ be be another scalar positive semidefinite function which vanishes only when any robots come in contact with each other. Then the multi-agent navigation function takes the form

$$\varphi(q) = \frac{\gamma(q)}{\beta(q)} \ . \tag{4}$$

Taking κ to be a positive tuning parameter, and recalling that c is the desired formation specification encoded in the definition of the formation graph (Definition 1), we can express function $\gamma(q)$ in the form

$$\gamma(q) = \gamma_d^{\kappa}(q) \ , \text{ with } \gamma_d(q) \triangleq \sum_{(i,j)\in E} \|q - c\|^2 \ . \tag{5}$$

Similarly, once we introduce tuning parameters r, μ and a that relate to the center distance between the disc-shaped robots at which collision occurs, and determine the derivative of function β at that location, we can express β analytically. In doing so we use q_i to denote the ith element of the stack vector q:

$$\beta(d) = \log\left(\mu - ae^{-(-r+d+d^2)^2}\right) \ , \text{ for } d(q) = \min(\|q_i\|) \ . \tag{6}$$

The term $\min(\|q_i\|)$ identifies the distance between those two robots that are closer to each other, and allows function β to encode proximity to collision without multiplying pair-wise proximity metrics. Avoiding the introduction of such a product enables a complete proof of the properties of the critical points of function φ [60]; the price for this is that now the potential function that coordinates the robots into formation is non-differentiable. Tools from nonsmooth analysis that can be brought to bear for this case are reviewed briefly in Appendix A.

If the dynamics of q, the latter defined in (3), become of the form

$$\dot{q} = -\nabla\varphi(q) \ ,$$

in the regions where $\varphi(q)$ is differentiable—the general treatment for both differentiable and non-smooth cases is in Section V—then the desired formation is guaranteed to be reached asymptotically [60]. Dynamics (2) is, however, of second order. The nontrivial extension of this control strategy from first to second order systems, and its integration with flocking controllers is postponed until Section V.

IV. Formation Path Following

We make use of windowing functions [61, Chapter 7] to avoid kinematic singularities and additional switching. The latter could be otherwise be introduced by path following approaches rooted in artificial potential fields constructed based on geometric primitives such as straight lines and curved segments. For the case of straight line reference paths—for extensions to Bézier curves see Appendix C—and with an appropriate selection of tuning parameters a_0 and r_0, we can define the windowing functions as follows (see also Fig. 2):

$$
\begin{aligned}
w_a(s) &= \frac{1}{2}\big[\tanh(a_0 s) - \tanh\big(a_0(s - \|\vec{v}\|)\big)\big] \\
w_R(q_s) &= \frac{1}{2}\big[\tanh\big(a_0(q_s + r_0)\big) - \tanh\big(a_0(q_s - r_0)\big)\big] \\
\theta(q_s) &= \frac{\pi}{2} e^{-a_0 q_s^2} \ .
\end{aligned}
$$

Function $w_a(s)$ is a windowing function acting along a path parameterized by $s \in \mathbb{R}$. The function is essentially the length of the projection on the path, of the vector $\vec{P}$ from the initial path point P_0 to the spatial position of interest p (see Fig. 2). Vector $\vec{v}$ extends from the initial point P_0 to the final point P_1 on the reference path, and $\hat{v}$ is the unit vector along the direction of $\vec{v}$. All these vectors are on the plane of motion, however they are considered embedded in $\mathbb{R}^3$ because of our need to take cross products. Windowing function $w_R(q_s)$ acts in a direction perpendicular to the path at distance q_s from it, and $\theta(q_s)$ is a function that will regulate the constructed vector field's orientation depending on the distance from the reference path.

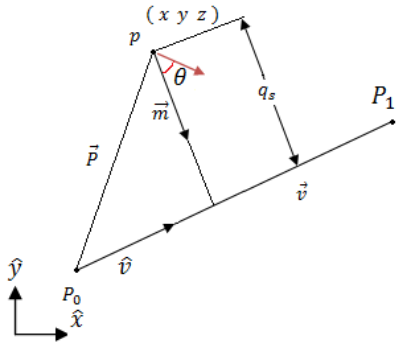
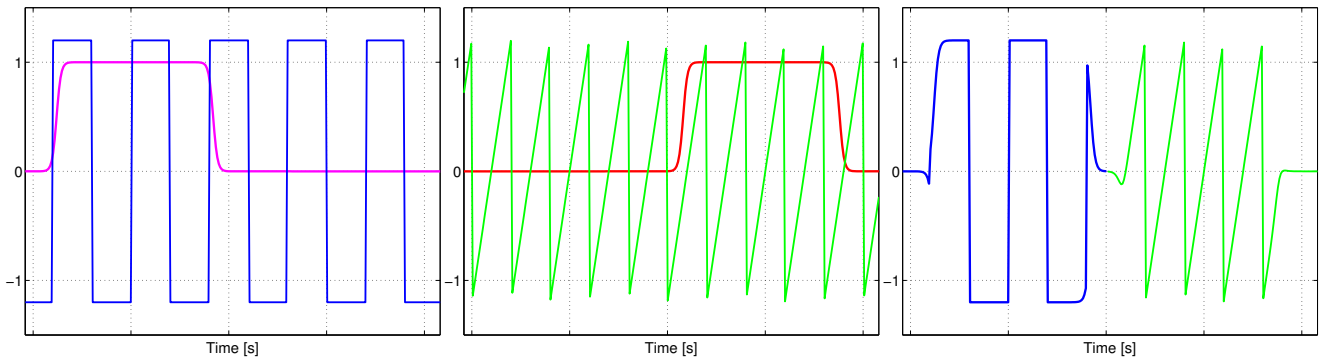


Fig. 2. Definition of windowing function parameters in the case of straight line segment reference paths

Vector $\vec{m}$ that appears in Fig. 2 is defined as $\vec{m} = \hat{v} \times (\hat{v} \times \vec{P})$, and a rotation matrix M is constructed as follows:

$$
M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \ . \tag{7}
$$

The windowing functions allow a smooth blending of vector fields designed for different segments of the desired path (Fig. 3). Figures 3(a) – 3(c) show a pulse and tooth-shaped signal and their corresponding windowing functions. The windowing functions essentially select a region of interest out of the whole domain of the signal, and attenuate it smoothly at the boundaries. Figure 3(c) shows the results of filtering these signals through their windowing functions and superimposing the outputs.



(a) (b) (c)

Fig. 3. Use of windowing functions to merge signals without introducing discontinuities. The purpose of windowing functions in signal processing is to mask the discontinuity in frequency both at the beginning and end of the recorded signal; on this work, windowing functions are used to merge straight line segments with Bézier curves so a smooth path is constructed without using switching approaches. In (a) and (b) the windowing function displayed in magenta and red respectively multiplies the original signals in blue and green to obtain the processed signal by adding up the latter products, as displayed in (c). Note that the windowing function tends to zero at the end points of the signal attenuating the effect of the discontinuity.

Through the use of appropriate windowing functions, we produce a single, smooth vector field u_p capable of steering the system (2) along the entire reference path. The input given to the system for path following consists of two components, which are superimposed. If we assume that the desired path is composed of n straight line segments, and m Bézier curve segments, we can think of the control input component u_p as formed by superposing the straight line path segment signal u_s and a Bézier curve segment signal u_b:

$$u_p = u_s + u_b = \sum_{i=1}^{n} u_{s,i} + \sum_{j=1}^{m} u_{b,j} \quad . \tag{8}$$

The following theorem establishes the convergence along straight line segments (see Fig. 2), of the a reference point in the (assumed rigid) formation. The proof can be found in Appendix B and centers around an application of the invariance principle.

Theorem 1. *Assume that the geometric center of the formation, $p_{\rm cm}$ has dynamics of the form*

$$\begin{aligned} \dot{p}_{\rm cm} &= \sum_{i=1}^{n} u_{s,i} \quad , \\ u_{s,i} &= w_{ai}\big(\vec{P}\cdot\hat{v}\big)\; w_{Ri}(q_s)\frac{M_i\,\vec{m}}{\|M_i\,\vec{m}\|} \quad . \end{aligned} \tag{9}$$

Then, in set $D = \{p_{\rm cm} \in \mathbb{R}^3 : 0 \le \vec{v}\cdot p_{\rm cm} \le 1\}$, the system (2) converges to set $A = \{k\vec{v} : 0 \le k \le 1\}$ for some fixed $\vec{v}$.

We can then show (see Appendix C) that Bézier curves can be diffeomorphically mapped to straight line segments, and thus the vector field that is developed for convergence to line segments can be adapted through the push back map to allow convergence to curved reference paths.

Assuming that robots fall into formation and maintain it, one would only need to steer the geometric center of this formation, denoted $p_{\rm cm}$ along the designated path. This is done by applying in (2) an input component u_p, uniformly to all vehicles.

V. Flocking with Formation Control

This section presents the extension of the formation control approach outlined in Section III to the case of the second order dynamics (2), also ensuring that the formation flocks, i.e., synchronizes its individual vehicle velocity vectors, as it achieves the desired shape. Flocking provides damping and facilitates asymptotic stability for formation control, while the formation stabilization pushes the limit sets of the dynamics to regions of the state space where velocities are synchronized. How this synergy is achieved is shown in the proof of the following theorem, found in Appendix D.

Recall the vehicle dynamics as expressed in (2), with x_1 and x_2 being the relative velocity and position stack vectors, respectively, living in a set

$$\mathcal{Q} = \mathbb{R}^{nN(N-1)/2} \times \mathbb{R}^{nN(N-1)/2} \setminus \{(x_1, x_2) : \|x_{2,i}\| < d_0, \forall (i,j) \in N \times N\} \ . \tag{10}$$

Letting Ln denote the generalized gradient vector field [62], we make the following statement.

Theorem 2. *Consider a group of N robots with dynamics as (2). If the control input in (2) is chosen as*

$$\begin{bmatrix} \Gamma_1^\intercal & \Omega_1^\intercal & \cdots & \Gamma_N^\intercal & \Omega_N^\intercal \end{bmatrix} = u_f^\intercal \tag{11}$$

where

$$u_f = -\mathrm{Ln}(\nabla(\varphi(x_2))) - x_1 \ , \qquad \varphi(x_2) = \frac{\gamma(x_2)}{\beta(d(x_2))} \ , \qquad d(x_2) = \min\{\|x_{2,i}\|\} \ , \tag{12}$$

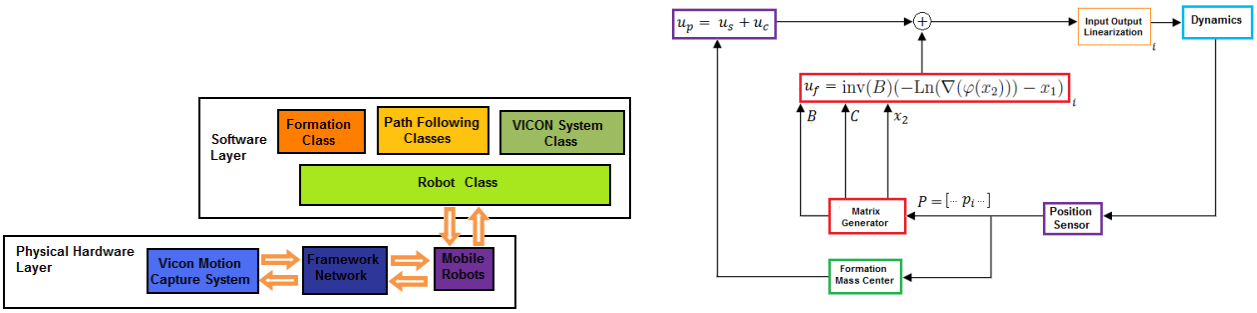
then collisions between robots are avoided, all robot velocity vectors become asymptotically the same, and the robots' relative position vectors $x_{2,i}$ converge to desired constants defined by the formation graph specifications.

Now since (2), augmented with the orientation dynamics (1c), is diffeomorphic to (1), any convergent trajectory of the former is mapped to a convergent trajectory of the latter. To account for the introduction of parameter L involved in the mapping of the state of (1) to that of (2), one can conservatively set d_0 to twice the sum of the robot's radius plus L.

VI. Integration, Testing and Performance Evaluation

All the components needed to implement the entire methodology are interfaced as shown in Fig. 4. We identify two major layers: a physical hardware and a cyber / software layer. The connection between the blocks of the physical layer in Fig. 4 is enabled by a wireless communication network, having links that do not depend on robot distances.

In our experimental assessment of the reported controllers, we use a type of mobile robot which is a skid-steering differentially driven platform, localized by a motion capture system (Fig. 5). Each robot has a Mini-ITX Motherboard with VIA C7 1.5GHz Processor, 1 GB of RAM memory, a wireless communication card and a hard disc of 60 GB that allow each robot to run Ubuntu Maverick Meerkat. Our client programs make the robots move by communicating with PLAYER 3.02. Although (1) does not strictly describes kinematics of skid-steering vehicles [58], it may still be considered a good approximation, especially when the origin of the body frame lies very close to the axis of the instantaneous radius of curvature.



(a) (b)

Fig. 4. (a) Layer hierarchy of the framework. (b) Implementation schematics.

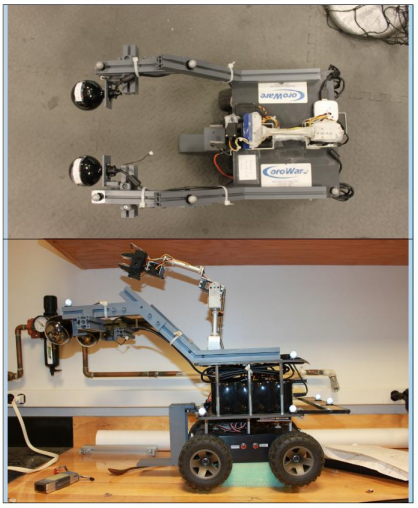


Fig. 5. Corobot. Differential robot manufactured by CoroWare. The metallic frame is custom made by us for the purpose of data collection.

A. Integration

Denoting $\text{inv}(B)$ denote the Moore-Penrose generalized inverse of matrix B, and letting u_p be the path following control signal, u_f the formation control signal, and $-x_1$ the flocking signal from the consensus controller, we can piece together the the complete control law that integrates path following, formation control, and flocking as

$$u_{\text{global}} = u_p + \text{inv}(B)(-\text{Ln}(\nabla(\varphi(x_2))) - x_1) \ . \tag{13}$$

With reference to Fig. 4(b), we can read the complete control signal on the right of the summing junction of the block diagram. This signal is mapped to linear and angular acceleration inputs for each differential mobile robot using the input-output linearization transformation. All controllers work concurrently: their input signals are superimposed, and key to being able to mesh seemlessly with each other is that the path following signal, being uniform over all vehicles, has no effect on flocking and formation controllers which work on relative positions (and velocities). Transformation (3) works like a noise cancellation circuit based on operational amplifiers:[3] it rejects the common component.

[3]See [63].

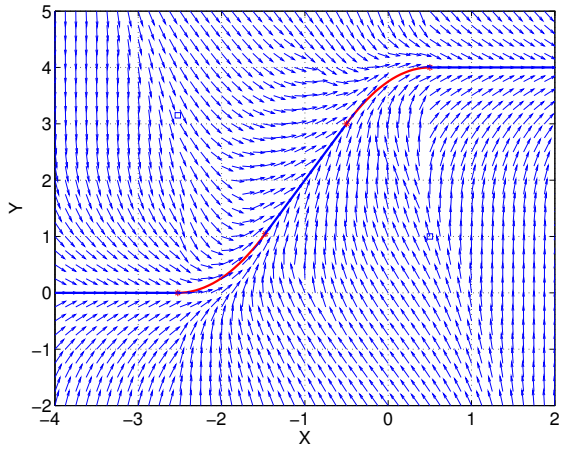


Fig. 6. Vector field to drive the formation along the desired path used in the simulation tests. The blue solid line is the path composed by straight line segments and the red solid line are the Bézier line segments.

B. Simulation

To test the closed loop system in simulation, we designed the path of in Fig. 6. It is composed of three straight lines and two Bézier curves. It can be verified that the vector field resulting from (8) for these five path segments is continuous everywhere and converges to the desired path.

The results of this simulation are presented in Fig. 7. Initially the five robots are scattered around the workspace. The formation specification calls for the robots to be in a diamond formation, with one of the robots enclosed in the middle. To reach this desired formation, the robots on the right have to switch positions. As seen in the snapshot of Fig. 7(h), the robot paths cross without them colliding with each other. After the configuration indicated by the snapshot in Fig. 7(g), we can verify that robots have achieved the formation specification and move as a rigid body along the desired path.

The time history of the key terms in the different control signals that are superimposed to produce the final control law is displayed in Fig. 8. The first part of this figure, Fig. 8(a) shows the time history of the numerator γ of $\varphi(x_2)$, which is a metric of the difference between the current configuration of the group of robots and the desired formation specification. The square markers in the trajectory of γ signify the time instances where γ passes over a nonsmooth cusp. Figure 8(b) displays how the distance between the center of mass of the group of robots and the reference path decreases with time. Here, color-coding in the segments of this trajectory refers to transitions between Bézier path segments (red/lighter) and straight lines (blue/darker). After the robots have traversed the second Bézier curve and attained the desired formation (at about $t = 675$ simulation seconds), this distance is practically zero. Figure 8(c) shows the time evolution of the velocity differences between the members of the flock. For this particular simulation, the desired formation was attained at around $t = 675$ simulation seconds, the flocking behavior manifested itself at around $t = 800$ simulation seconds, and accurate path following was practically exhibited just after the 350th simulation second.

C. Experimental results

The objective of the experimental tests we conducted was to map a designated area in a fashion similar to that indicated in Fig. 1(b): steer robots along a meandering path, but instead of having one, as in the case of the mission in Fig. 1(b), have a line

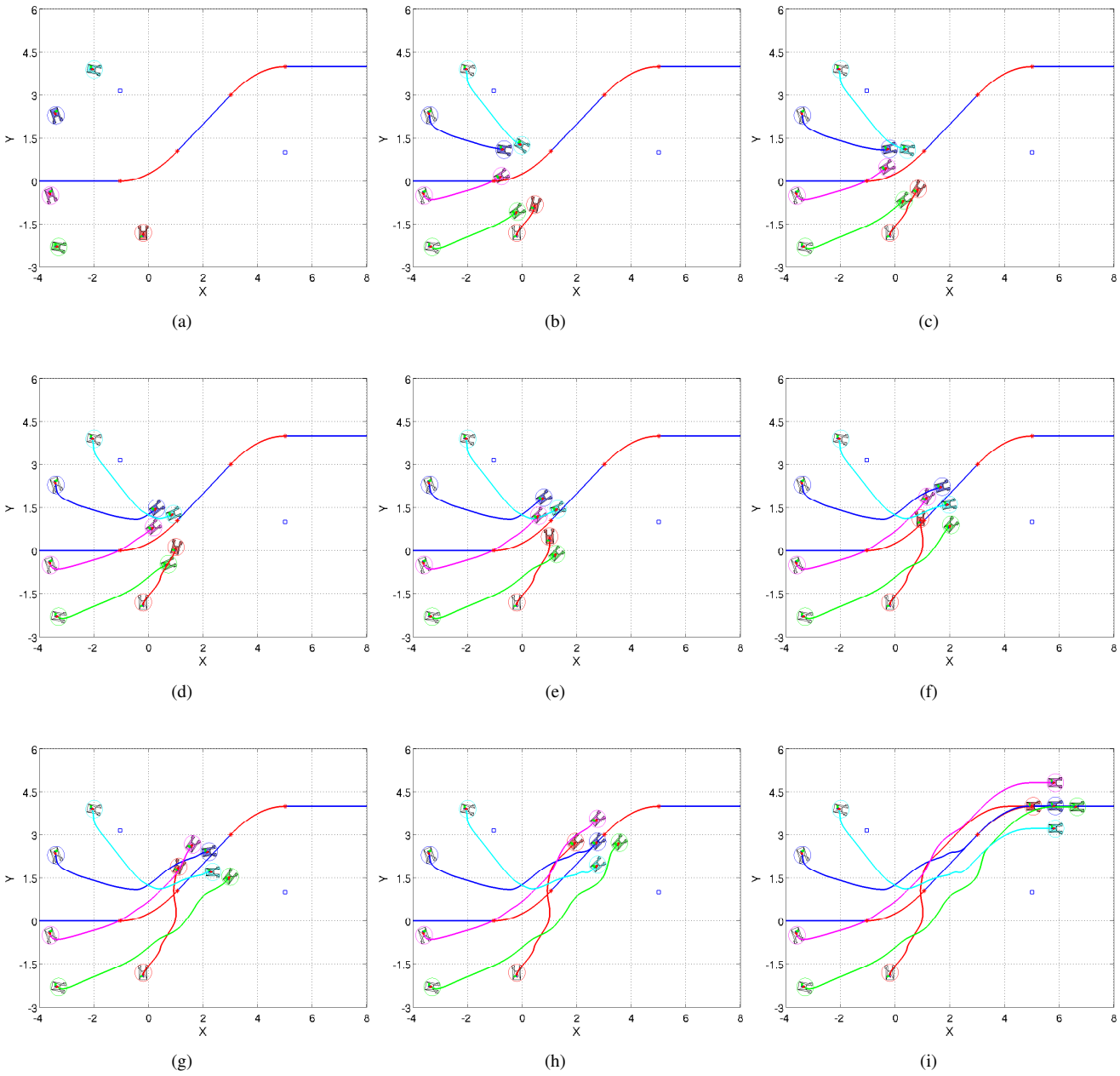
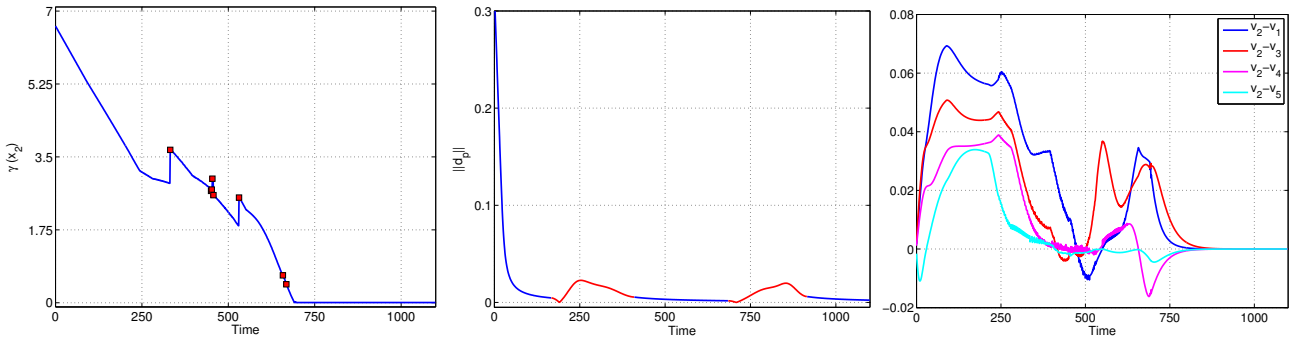


Fig. 7. Simulation results of a group of fie robots falling into formation a following the desired path. (a) $t = 0$. (b) $t = 250$. (c) $t = 300$. (d) $t = 350$. (e) $t = 480$. (f) $t = 550$. (g) $t = 650$. (h) $t = 720$. (i) $t = 1000$.

formation of robots—in this case, there are just two robots (Fig. 9(a)), due to space and motion capture envelope limitations. The robots are using two webcams each, and take successive pictures of the floor, which they subsequently combine into a mosaic like the one shown in Fig. 9(b).

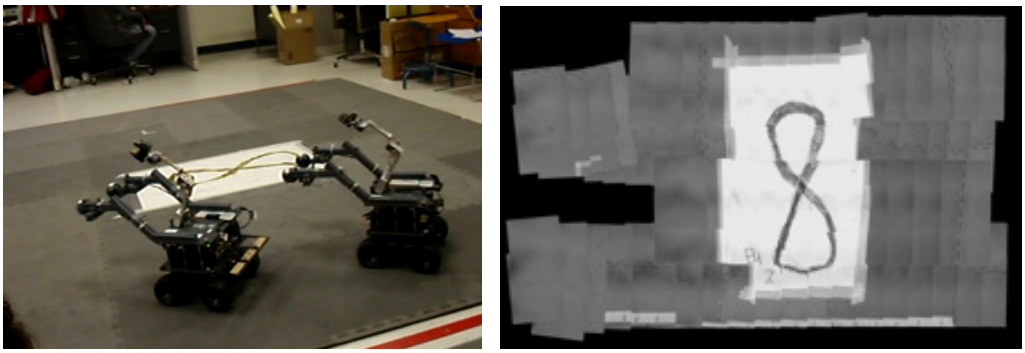
The desired shape for the group of robots is a line formation, motivated by the need to cover as much ground as possible in each straight path segment of the meander. Thus, the line formation follows the path while keeping the robots aligned perpendicular to the desired path. It is important that the formation is kept compact with fairly tight tolerances, otherwise there can be gaps left between the images that neighboring robots take, compromising coverage and making mosaicking difficult.

Our control laws require position feedback which is provided by our motion capture system at a rate of 100 Hz. Figure 10



(a) (b) (c)

Fig. 8. (a) Evolution of formation control among the group of robots. (b) Evolution of path following control of the group of robot's center of mass. (c) Evolution of flocking control.



(a) (b)

Fig. 9. (a) The two robots in the experiment; (b) A mosaic created by a robot taking pictures while executing a meandering maneuver—courtesy of Prasanna Kannappan

shows the paths of the two robots while executing the required maneuver. The desired (oriented) path is shown in Fig. 10(i). Each robot is color-coded in the figure, and is surrounded by a matching (in color) circle that represents its "personal space" (of radius d_0). The actual robot shape is outlined by a polygon that shows where the main body part of the robot is, and where the booms holding its down-facing cameras are pointing.

In Fig. 10(a) the red (lighter) robot starts needs to move to the left to fall into formation with the blue (darker) robot. Figure 10(a) shows the paths of the two robots as they maneuver into formation. After the the initial maneuvering of Fig. 10(a), we see in Fig. 10(b) that the desired formation shape has been achieved, although the final orientation is not condusive to the robots moving together as a rigid body. It is there that the path following and flocking controller signals take the initiative to steer the pair south and along the first segment of the desired path as shown in Fig. 10(c). At the end of that segment, the formation changes orientation and turns right as it follows a Beziér curve that links that first line segment to the one on the right of the leftmost vertical segment. In Fig. 10(d) we see the group climbing north that straight line segment, before turning sharply left to follow the leftmost straight line segment south. The formation completes this maneuver in Fig. 10(e) and Fig. 10(f), before turning left again to follow the last portion of the desired path heading north, as shown in Figs. 10(g) and 10(h).

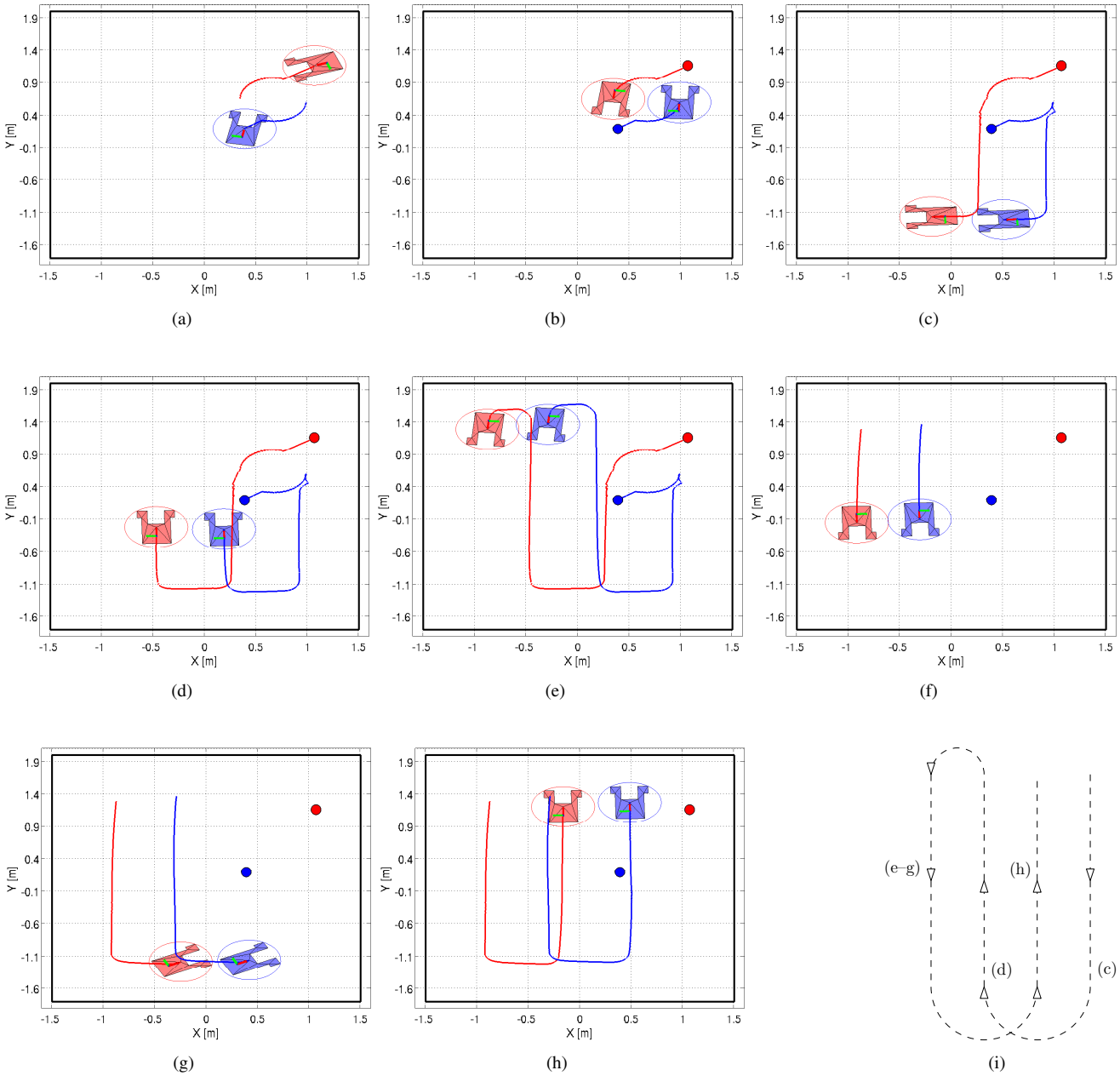


Fig. 10. Progression of the experimental trial with the two robots following a meandering curve. Thick dots mark the vehicles' initial positions, and the short trail behind each robot show the recent portion of the path it has traversed so far.

VII. Conclusion

A homogeneous group of mobile robotic sensor platforms can be effectively coordinated to perform maneuvers required for search, coverage, and mapping missions, as if they were a single, much larger, rigid structure. Composing such a structure out of identical modular components, has advantages related to cost and robustness, since reconfiguration through relatively small software adjustments allows the deployment of such a group even when a few of its members become inoperable. The particular motion control methodology outlined in this paper brings together three distinct controlled mobility behaviors, namely, formation control, path following and flocking, into a single framework where all of them act concurrently but in

concert with each other. The mobility behaviors integrated here draw from existing work, but in order for the confluence of control action described above to occur, several specialized innovations had to be implemented in the constituent technologies. The overall architecture has proven effective both in numerical simulations as well as in experimental implementation on a small group of wheeled mobile robots.

APPENDIX A
NONSMOOTH ANALYSIS

A. Generalized directional derivatives and gradients

Take a function $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$, and consider the vector differential equation

$$\dot{x} = f(x,t) \ . \tag{14}$$

Take co denotes the convex hull[4] and let N and N_f be any sets of zero i.e., $\mu N = 0$. The operator μ is the Lebesgue measure[5] and N_f is a set in which $f(x,t)$ is not defined; see [64].

When the right hand side of (14) is discontinuous, the solution can be expressed in the form of a differential inclusion:

$$\begin{aligned} \dot{x}(t) &\in F(t,x(t)) \\ F(t,x(t)) &= \mathrm{co}\{\lim f(x_i) | x_i \to x, x_i \notin N_f \cup N\} \ . \end{aligned} \tag{15}$$

A Filippov solution of (14) on $[0,t_1] \subset \mathbb{R}$ is an absolutely continuous map $x : [0,t_1] \to \mathbb{R}^n$ that satisfies (15) for almost all $t \in [0,t_1]$ [62].

Definition 2 ([65])**.** *Let f be Lipschitz near a given point x in $X \subset \mathbb{R}^n$, y be any other point near x, v an arbitrary vector and t a positive scalar. The* generalized directional derivative *of f at x in the direction of v, denoted $f^\circ(x,v)$, is defined as*

$$f^\circ(x,v) = \limsup_{\substack{y \to x \\ t \downarrow 0}} \frac{f(y+tv) - f(y)}{t} \ . \tag{16}$$

The generalized directional derivative expresses the rate of change of an non-smooth function at a point of non-differentiability. The generalized gradient is the analog of the regular gradient for the case where functions are not differentiable. Let X^* be the dual space of X, where the open unit ball is denoted B_* and for the norm $\|\zeta\|_*$ we write $\|\zeta\|_* \triangleq \sup\{\langle \zeta, v\rangle : v \in X, \|v\| \leq 1\}$. The generalized gradient of f at x, denoted $\partial f(x)$, is the subset of X^* given by

$$\left\{\zeta \in X^* : f^\circ(x,v) \geq \langle \zeta, \xi \rangle \text{ for all } v \text{ in } X\right\} \ .$$

If we allow set Ω_f to contain all point in $x + \varepsilon B$ for which f is not differentiable, and S be any set of measure zero, then in

[4]The convex hull co of a set C is the set of all convex combinations of points in C:

$$\mathrm{conv}(C) = \left\{\theta_1 c_1 + \ldots + \theta_k c_k \mid c_i \in C, \theta_i \geq 0, i = 1, \ldots, k, \sum_{i=1}^{k} \theta_i = 1\right\}.$$

[5]It is a standard way to assign a measure to a subset contained in a n-dimensional Euclidean space; if $n = i,$ with $i = 1, 2, 3$, the Lebesgue measure correspond to length, area and volume respectively.

finite-dimensional spaces, the generalized gradient, defined over a ball εB centered at x, can take the form

$$\partial f(x) = \mathrm{co}\Big\{ \lim_{i\to\infty} \nabla f(x_i) : x_i \notin S \cup \Omega_f \Big\} \ .$$

The generalized gradient $\partial f(x)$ is the convex hull of all points of the form $\lim \nabla f_i$, where $\{x_i\}$ is any sequence which converges to x while avoiding $S \cup \Omega_f$.

While the gradient is a vector, the generalized gradient is a convex set. A set C is convex if the line segment between any two points in C lies in C, it means that for any c_1 and $c_2 \in C$ and any $0 \leq \theta \leq 1$, we have $\theta c_1 + (1-\theta)c_2 \in C$ [66].

For the distance function, in particular, there are interesting properties that its generalized gradient enjoys. One of them is the following [65, Theorem 2.5.6]:

Theorem 3 ([65]). *Let x belong to $\mathrm{cl}(\Upsilon)$. Then $\partial d_\Upsilon(x)$ equals the convex hull of the origin and the set $\big\{w = \lim \frac{w_i}{\|w_i\|}$: $w_i \perp \Upsilon \quad at \quad x_i \to x, w_i \to 0\big\}$.*

B. Ryan's invariance principle

Ryan's invariance theorem allows us to conclude the weak convergence of the solutions of a differential inclusion to a set [67]. To this end, consider the non-autonomous initial-value problem

$$\dot{x}(t) \in F(t, x(t)), \quad x(t) \in G, \quad x(t_0) = x^0, \tag{17}$$

where $G \subset \mathbb{R}^N$. We need the following definitions:

Definition 3 ([67]). *A solution x of the differential inclusion* (17) *is said to be maximal, if it does not have a proper right extension which is also a solution of* (17)*.*

Definition 4 ([67]). *Every solution of* (17) *can be extended to a maximal solution.*

Definition 5 ([67]). *A solution $x \in AC([t_0, \omega; G])$ of* (17) *is precompact if it is maximal and the closure $cl(x([t_0, \omega)))$ of its trajectory is a compact subset of G.*

Definition 6 ([68]). *Let $x(t)$ be a bounded solution of* (17)*. If x_t defined by $x_t(0) = x(t+0)$ has no positive limit points on the boundary of X, then $x(t)$ is precompact.*

Theorem 4 ([67]). *Let $V : G \to \mathbb{R}$ be locallly Lipschitz. Define*

$$u : G \to \mathbb{R}, \quad z \to u(z) \triangleq \max\{V^\circ(z, \phi) \mid \phi \in X(z)\}$$

Suppose that $U \subset G$ is non-empty and that $u(z) \leq 0$ for all $z \in U$. If x is a precompact solution of (17) *with trajectory in U, then, for some constant $\mathsf{c} \in V(\mathrm{cl}(U) \cap G)$, x approaches the largest weakly-invariant set in $\Sigma \cap V^{-1}(\mathsf{c})$, where $\Sigma = \{z \in \mathrm{cl}(U) \cap G \mid u(z) \geq 0\}$.*

APPENDIX B

PROOF OF THEOREM 1: CONVERGENCE TO STRAIGHT-LINE SEGMENTS

Without loss of generality take the $\hat{x}$ axis of the reference frame along the reference path A (see Fig. 2)—this is a translation-rotation of the original problem where we put the origin at P_0 and $\vec{v} = P_1 - P_0$.

Letting I be the (2×2) identity matrix, write

$$\vec{m} = \hat{v} \times (\hat{v} \times \vec{P}) = (\hat{v}\,\hat{v}^\intercal - I)\vec{P} \ .$$

Consider a weak-Lyapunov candidate function (for planar motion)

$$V(\vec{P}) = \frac{1}{2}\|\vec{P}\|^2 = \frac{1}{2}\big(p_x^2 + p_y^2\big) \ .$$

Taking the derivative of this function with respect to time along (9) yields

$$\dot{V}(\vec{P}) = \frac{w_a(\vec{P}\cdot\hat{v})\, w_R(q_s)}{\|M(\hat{v}\,\hat{v}^\intercal - I)\vec{P}\,\|}\big[M(\hat{v}\,\hat{v}^\intercal - I)\vec{P}\,\big]^\intercal\vec{P} \tag{18}$$

The numerator of the scalar factor is composed of windowing functions that take values between zero and one, and the denominator merely scales the vector on the right to unitary length. Therefore the sign of the right-hand side of (18) is determined by

$$\big[M(\hat{v}\,\hat{v}^\intercal - I)\vec{P}\,\big]^\intercal\vec{P} = \big[M\big((\hat{v}\,\hat{v}^\intercal)\vec{P} - \vec{P}\,\big)\big]^\intercal\vec{P} \ ,$$

(see [69]) and since $\hat{v} = [1\ 0\ 0]^\intercal$ and $\vec{P} = [p_x\ p_y\ 0]^\intercal$, we have $(\hat{v}\,\hat{v}^\intercal)\vec{P} - \vec{P} = [0\ \ -p_y\ 0]^\intercal$. Recalling the expression (7) of the rotation matrix M, we have

$$M\big((\hat{v}\,\hat{v}^\intercal)\vec{P} - \vec{P}\big) = \begin{bmatrix} 0 \\ -p_y\cos\theta \\ -p_y\sin\theta \end{bmatrix} \ ,$$

and computing its inner product with $\vec{P}$,

$$\begin{bmatrix} 0 & -p_y\cos\theta & -p_y\sin\theta \end{bmatrix} \cdot \begin{bmatrix} p_x \\ p_y \\ 0 \end{bmatrix} = -p_y^2\cos\theta \ ,$$

from which it follows that $\dot{V}(\vec{P}) \leq 0$, since $|\theta| < \frac{\pi}{2}$.[6] Note now that as $V(\vec{P})$ is radially unbounded and continuous, the set

$$\Omega_c = \{(p_x, p_y) : V(\vec{P}) \leq c\}, \quad \Omega_c \subset D$$

is a compact, positively invariant set. If we define $E \triangleq \{(p_x, p_y) \in \Omega_c : \dot{V}(\vec{P}) = 0\}$, and evaluate vector field (9) there, we see that E is an invariant set. The set E is essentially the line that passes from points P_0 and P_1 (Fig. 2). From [70, Theorem 4.4] it follows that every trajectory starting in Ω_c approaches E as $t \to \infty$.

[6]This follows from the construction of the vector field: points enabled by the windowing functions are assigned a vector that is initially normal to vector $\vec{P}$, before rotated based on $\theta(q_s)$ as illustrated in Fig. 2.

APPENDIX C
CONVERGENCE TO BÉZIER SEGMENTS

For the convergence analysis, we show there is a diffeomorphism that can map the curved Bézier line into a straight line. In principle, the pull-back map of this diffeomorphism can translate the straight line controller so that it is used for convergence to Bézier curves. A diffeomorphism is merely a coordinate transformation; it does not affect the convergence properties of a vector field. The structure of the two controllers is the same, it is merely the scaling that changes; vector $\vec{m}$ always points toward the reference curve along the normal vector to the curve.

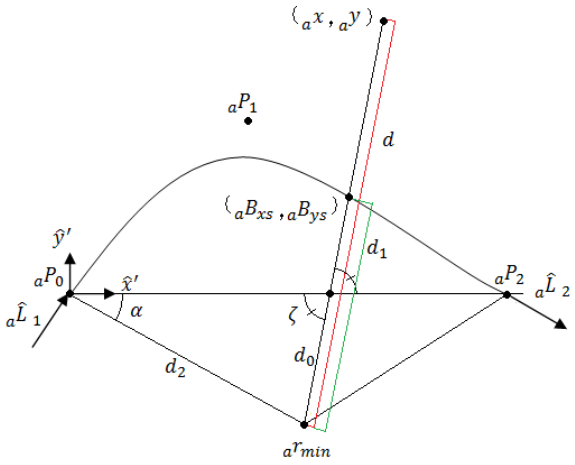


Fig. 11. Mapping to simplify the analysis for Bézier line segments

First, we start by translating and rotating the locus of the Bézier curve defined by points ${}_aP_0$, ${}_aP_1$, and ${}_aP_2$, in order to simplify like the expressions of the Bézier curve and the roots of some polynomials involved in the required derivations. We denote the initial frame as I_r and the frame of analysis as I_a, indicated in Fig. 11.

Let $\mathrm{atan2_d}(\cdot,\cdot)$ be a customized arc tangent function defined as

$$\mathrm{atan2_d}(y,x) = \begin{cases} \dfrac{\arccos\left(\frac{x}{\sqrt{x^2+y^2}}\right)\arcsin\left(\frac{y}{\sqrt{x^2+y^2}}\right)}{\sqrt{\arctan\left(\frac{y}{x}\right)^2}}\,, & x\neq 0, y\neq 0 \\ \pi/2\,, & x=0, y>0 \\ -\pi/2\,, & x=0, y<0 \\ \pi\,, & y=0, x<0 \\ 0\,, & y=0, x>0 \\ 0\,, & x=0, y=0 \end{cases} .$$

and define $\vartheta = \mathrm{atan2_d}({}_rP_{2y} - {}_rP_{0y}, {}_rP_{2x} - {}_rP_{0x})$. Then a point from frame I_r translated to frame I_a as follows

$${}^r_a p = \begin{bmatrix} \cos\vartheta & \sin\vartheta \\ -\sin\vartheta & \cos\vartheta \end{bmatrix} \left({}_r p - {}_r P_0\right) .$$

The diffeomorphism sending $({}_a x, {}_a y)$ to (z_1, z_2) according to

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \frac{d_0 d_1}{d^2} \begin{bmatrix} {}_a x - {}_a r_{\min x} \\ {}_a y - {}_a r_{\min y} \end{bmatrix} \ , \tag{19}$$

transforms the Bézier curve of Fig. 11 into a straight line interval P_0, P_2. The point ${}_a r_{\min}$ is located at the intersection of the normals to the Beziér at points ${}_a P_0$ and ${}_a P_2$ on the plane of motion (see Fig. 11). The other parameters involved in this (ray) scaling are expressed (with reference again to Fig. 11) as

$$\begin{aligned} d_2 &= \sqrt{{r_{\min x}}^2 + {r_{\min x}}^2} \\ \alpha &= |\mathrm{atan2_d}({}_a r_{\min y}, {}_a r_{\min x})| \\ \zeta &= |\mathrm{atan2_d}({}_a y - {}_a r_{\min y}, {}_a x - {}_a r_{\min x})| \\ d_0 &= \frac{\sin\alpha}{\sin\zeta} d_2 \\ d &= \sqrt{({}_a x - {}_a r_{\min x})^2 + ({}_a y - {}_a r_{\min y})^2} \ . \end{aligned}$$

If we denote $t_s \in [0,1]$ the admissible solution of the polynomial in parameter t

$$\begin{aligned} -[2({}_a x - {}_a r_{minx})\,{}_a P_{1y} - 2({}_a y - {}_a r_{miny})\,{}_a P_{1x} + {}_a P_{2x}({}_a y - {}_a r_{miny})]t^2 \\ + [2({}_a x - {}_a r_{minx})\,{}_a P_{1y} - 2({}_a y - {}_a r_{miny})\,{}_a P_{1x}]t \\ + {}_a r_{minx}({}_a y - {}_a r_{miny}) - {}_a r_{miny}({}_a x - {}_a r_{minx}) = 0 \ , \end{aligned}$$

and define ${}_a B_{xs} \triangleq 2\,t_s(1-t_s)\,{}_a P_{1x} + {}_a P_{2x} t_s^2$, ${}_a B_{ys} \triangleq 2\,t_s(1-t_s)\,{}_a P_{1y}$. then we define the remaining parameter d_1 as

$$d_1 = \sqrt{({}_a B_{xs} - {}_a r_{\min x})^2 + ({}_a B_{ys} - {}_a r_{\min y})^2}$$

APPENDIX D
PROOF OF THEOREM 2

The equilibria of the system appear at configurations where

$$x_1 = 0 \ , \qquad\qquad \partial\,\varphi(x_2) = 0 \ .$$

(∂ denotes generalized gradient; see Appendix A.) Thus the system is at equilibrium when all robots are either at their desired relative configurations, or at a saddle point of φ. The set of the latter can be shown to consist of isolated configurations which are unstable [60], and the attraction region of these unstable equilibria is of measure zero. For this reason, the only practically attainable equilibrium is $x_1 = 0, x_2 = c$. Our weak-Lyapunov function candidate is defined as

$$V(x_1, x_2) = \frac{1}{2}\|x_1\|_2^2 + \varphi(x_2)$$

which is positive definite because $\varphi(x_2)$ is positive definite with respect to x_2.[7] At configurations where the minimum relative distance function $d(x_2)$ is nondifferentiable, $\dot{x}$ belongs to a Filippov set [64]

$$\dot{x} \in F(x) \ . \tag{20}$$

Specifically, first compute

$$\partial\varphi(x_2) = \frac{1}{\beta^2}\left(\kappa\beta(\gamma_d)^{\kappa-1}\nabla(\gamma_d) - \gamma_d^{\kappa}\Big(\frac{\partial\beta}{\partial d}\Big)\lambda\right) \ , \quad \lambda \in \partial d \ ,$$

and then write the differential inclusion as

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} \in \begin{bmatrix}-\partial\varphi(x_2) - x_1\\ x_1\end{bmatrix} = F(x) \ .$$

Once you express the generalized gradient of V in the form

$$\partial V(x) = \begin{bmatrix}x_1\\ \partial\varphi(x_2)\end{bmatrix}$$

then you can find the generalized derivative of V along a vector in $F(x)$ as [65, Proposition 2.1.2]

$$V^{\circ}(x,\xi) = \max\{\, p^{\intercal}\cdot\xi,\ p \in \partial V(x),\ \xi \in F(x)\} \ .$$

Expanding the generalized directional derivative gives a long expression

$$V^{\circ}(x,\zeta) = \max\Bigg\{ -\frac{x_1^T}{\beta^2}\left(\kappa\beta(\gamma_d)^{\kappa-1}\nabla(\gamma_d)^T - \gamma_d^{\kappa}\Big(\frac{\partial\beta}{\partial d}\Big)\zeta\right) - \|x_1\|_2^2 \\ + \frac{x_1^T}{\beta^2}\left(\kappa\beta(\gamma_d)^{\kappa-1}\nabla(\gamma_d)^T - \gamma_d^{\kappa}\Big(\frac{\partial\beta}{\partial d}\Big)\xi\right)\Big|\ \zeta,\xi \in \partial d\Bigg\}$$

which eventually simplifies to

$$V^{\circ}(x,\zeta) = \max\Bigg\{ -\|x_1\|_2^2 + \left(\frac{\gamma_d^{\kappa}}{\beta^2}\Big(\frac{\partial\beta}{\partial d}\Big)\|x_1\|\,\|\zeta-\xi\|\right)\Big|\ \zeta,\xi \in \partial d\Bigg\} \ .$$

The maximum value for the norm $\|\zeta - \xi\|$ is achieved when the unit vectors ζ and ξ, which are members of ∂d, are along different, orthogonal coordinate directions. If K_ζ is an upper bound on $\frac{\partial\beta}{\partial d}$ [60], it then follows that

$$V^{\circ}(x,\zeta) \leq -\|x_1\|_2^2 + \sqrt{2}\frac{K_\zeta}{\beta}\|x_1\|\varphi \ . \tag{21}$$

Now depending on whether φ is differentiable at the given configuration x_2, we consider two cases.

a) Case I: suppose φ is differentiable at x_2; then $\zeta = \xi$, $V^{\circ}(x,\zeta) = \dot{V}(x)$, and consequently

$$\dot{V}(x_1,x_2) = -\|x_1\|^2 \leq 0 \ .$$

[7] φ might depend on a subset of the components of x_2 but because of the $N-1$ connectivity (graph) constrains between the agents, fixing the specific componets, determines all the elements in x_2.

Since $V(x_1, x_2)$ is radially unbounded, the set

$$\Omega = \{(x_1, x_2) \in \mathbb{R}^{nN(N-1)} | V(x_1, x_2) \leq V\big(x_1(0), x_2(0)\big)\}$$

is a positive invariant set that is closed by continuity and bounded due to radial unboundedness of V. Therefore, solutions that start in Ω cannot escape Ω, which implies that robots cannot collide since collision configurations send $V \to \infty$.

Let c be a constant vector and define $E \subset \Omega$ as follows:

$$E = \{(x_1, x_2) \in \mathbb{R}^{nN(N-1)} | \quad \dot{V}(x_1, x_2) = 0\} \iff E = \{(x_1, x_2) \in \mathbb{R}^{nN(N-1)} | \quad x_1 = 0, x_2 = \mathsf{c}\} \ ,$$

The dynamics on E is

$$0 = \dot{x}_1 = -\nabla(\varphi(\mathsf{c})) - x_1$$

and the biggest invariant set $M \subset E$ is thus $(x_1, x_2) = (0, \mathsf{c})$. The trivial solution is the only one which can stay in E.

Invoking LaSalle's principle [70] we conclude that every solution starting in Ω will approach M as $t \to \infty$, and by [70, Corollary 4.1] we see that trajectories asymptotically converge to $(0, \mathsf{c})$.

b) Case II: suppose now that φ is nondifferentiable at x_2, i.e., $\zeta \neq \xi$. Then, there are regions of the state space where $V^\circ(x, \zeta)$ is positive (Fig. 12). However, the solutions for (11) still have to be bounded: if they were not, then, for every constant $\mathsf{C} \geq 0$, there would be some $t \geq 0$ such that $V(x(t)) > \mathsf{C}$. There would be initial conditions $x(0)$ where $V^\circ\big(x(0), \zeta\big) \geq 0$, implying

$$\|x_1\| < \frac{\varphi}{\beta} \|\xi - \zeta\| K_\zeta \ . \tag{22}$$

This happens in the region on the upper left hand side in both parts of Fig. 12. If x were to grow without bound, then its trajectory $x(t)$ would eventually hit the boundary of the workspace, the latter marked by those points where φ blows up. It is ensured [60] that $-\nabla\varphi$ is unbounded, there, pointing in the direction which x_2 decreases. Thus x_2 cannot grow unbounded. If x_2 is to remain bounded, then the only option left for $V(x)$ to increase is if x_1 increases; but that contradicts (22). Therefore neither x_1 nor x_2 can grow without bound. Since $x(t)$ is a bounded solution, and it has no positive limit point on the boundary of the workspace [60], then $x(t)$ must be precompact [68, Lemma 4.8].

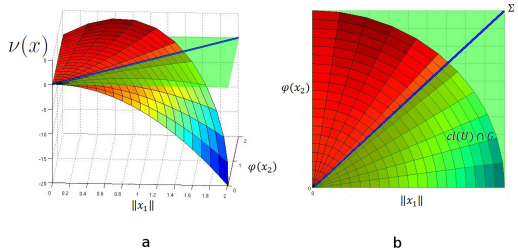


Fig. 12. Sketch for $V^\circ(x, \zeta)$. $V^\circ(x, \zeta)$ has three important regions, $V^\circ(x, \zeta) \geq 0$, $V^\circ(x, \zeta) \leq 0$ and $V^\circ(x, \zeta) = 0$.

In this case, define (see [67, Theorem 2.11])

$$\nu : G \to \mathbb{R} \ , \quad x \to \nu(x) \triangleq \max\{V^\circ(x, \zeta) \mid \zeta \in F(x)\} \leq -\|x_1\|^2 + \sqrt{2}\frac{K_\zeta}{\beta}\|x_1\|\varphi \ .$$

Letting μ be the navigation function constant parameter, it follows that is a nonempty set $U \subset \mathcal{Q}$ where $\nu(x) \leq 0$. Such a set can be defined as

$$U = \left\{ \|x_1\| \geq \frac{\varphi}{\log(\mu)} \|\xi - \zeta\| K_\zeta \right\} \ . \tag{23}$$

To complete this proof, let cl denote closure and define

$$\Sigma \triangleq \{x \in \mathrm{cl}(U) \cap \mathcal{Q} \mid \nu(x) \geq 0\} \ .$$

Then invoke Ryan's integral invariance principle—Theorem 4, quoted in Appendix A—to conclude that $x(t)$ converges to largest weakly-invariant set within

$$\Sigma \cap V^{-1}(\mathsf{c}) \ .$$

It can actually be shown that once on $\Sigma \cap V^{-1}(\mathsf{c})$, the trajectories of (11) remain tangent to Σ, and the vector field is non vanishing anywhere except for $(x_1,\ x_2) = (0, \mathsf{c})$, that is, $\{x | x_1 = 0, \varphi(c) = 0\}$. What this implies is that the largest weakly invariant set in $\Sigma \cap V^{-1}(\mathsf{c})$ is $(x_1,\ x_2) = (0, \mathsf{c})$.

The Filippov set (20) along $\Sigma \setminus \{x | x_1 = 0, \varphi(c) = 0\}$, see Fig. 13, is given by

$$F(x) = \begin{bmatrix} -\frac{1}{\beta^2}\left(\kappa\beta(\gamma_d)^{\kappa-1}\nabla(\gamma_d) - \gamma_d^\kappa\left(\frac{\partial\beta}{\partial d}\right)\lambda\right) - x_1 \\ \\ x_1 \end{bmatrix} \ ,$$

and we can verify that $0 \notin F(x)$, so there can be no equilibrium anywhere but at $(x_1,\ x_2) = (0, \mathsf{c})$. To see that, note that since $\lambda \in \partial d(\|x_2\|)$, and the latter is the convex hull of some unit vectors e_i, the former can be written as $\lambda = \Sigma_i a_i e_i$ where $\Sigma_i a_i = 1$; thus λ cannot be zero. On the other hand, the only place where both x_1 and γ_d are zero, and therefore $F(x) = 0$, is only at $x_1 = \varphi(\mathsf{c}) = 0$, by construction of $\varphi(x_2)$. For $\varphi(x_2)$ to vanish one needs either $x_2 = 0$, or $\frac{\partial\beta}{\partial d} = 0$, the latter implying that agents are so far away from each other that there is no interaction between them; this happens when $d \to \infty$, as (6) suggests.

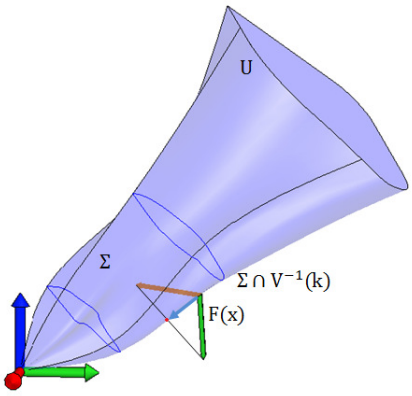


Fig. 13. Filippov set along $\Sigma \setminus \{x | x_1 = 0, \varphi(c) = 0\}$

Since $F(x)$ contains a tangent direction to the set $\Sigma \setminus \{x | x_1 = 0, \varphi(\mathsf{c}) = 0\}$, the set $\Sigma \setminus \{x | x_1 = 0, \varphi(\mathsf{c}) = 0\}$ cannot be weakly-invariant; rather, there is a solution that escapes any bounded proper subset of $\Sigma \setminus \{x | x_1 = 0, \varphi(\mathsf{c}) = 0\}$. Let

$d_\Sigma(x) = \inf\{\|x-\sigma\| \mid \sigma \in \Sigma\}$. The tangent cone T_Σ to $\mathrm{cl}(\Sigma)$ at a point x is then

$$T_\Sigma(x) = \{\zeta : d^\circ(x,\zeta) = 0\} \ .$$

Algebraically, the fact that some $v \in F(x)$ belongs to this tangent cone can be expressed as

$$\exists\, v \in F(x) : \max\left\{\langle \zeta, v\rangle \mid \zeta \in d_\Sigma(x)\right\} = 0 \iff \exists\, v \in F(x) \mid \langle \zeta, v\rangle \leq 0 \ . \tag{24}$$

Note that computing $d_\Sigma(x)$ requires knowledge of the point σ in the closure of Σ that is closest to x, which is in general unknown. Theorem 3 provides an alternative expression for the normal of Σ, which we will be using presently.

Note however, that the direction of inequality (24) is determined by direction of ζ and v, and not their magnitude. It will prove convenient to scale both factors in $\langle \zeta, v\rangle$ by $\|x_1\|$. Then for an arbitrary element $v \in F(x)$, the first factor would appear as

$$\frac{v}{\|x_1\|} = \begin{bmatrix} -\frac{1}{\beta\|x_1\|}\left(2\kappa(\gamma_d)^{\kappa-1}(x_2-\mathsf{c}) - \frac{\beta}{\sqrt{2}}\|x_1\|\left(\sum_i a_i\xi_i\right)\right) - \frac{x_1}{\|x_1\|} \\ \\ \frac{x_1}{\|x_1\|} \end{bmatrix} \ .$$

On the other hand, since

$$\Sigma = \left\{x = (x_1,x_2) : \Psi(x) \triangleq \|x_1\|^2 - \frac{2}{\beta^2}K_\zeta^2\varphi(x_2)^2 = 0\right\} \ ,$$

then for $x \in \Sigma$, by Theorem 3 the scaled second factor would be of the form

$$\frac{\zeta}{\|x_1\|} = \frac{\partial\Psi(x)}{\|x_1\|} = \begin{bmatrix} 2\frac{x_1}{\|x_1\|} \\ -2\frac{\|x_1\|}{\beta K_\zeta}\left(\sum_i a_i\xi_i\right) \ . \end{bmatrix}$$

Therefore, vector ζ in (24), being in $\partial d_\Sigma(x)$ will be parallel to $\frac{\partial\Psi(x)}{\|x_1\|}$. The scaled inner product of (24) is now expanded as

$$\left\langle \frac{\zeta}{\|x_1\|}, \frac{v}{\|x_1\|}\right\rangle = -\frac{4\kappa(\gamma_d^{\kappa-1})}{\beta\|x_1\|^2}x_1^\intercal\cdot(x_2-\mathsf{c}) - \frac{2}{K_\xi\beta}x_1^\intercal\cdot\Big(\sum_i a_i\xi_i\Big) + \frac{\sqrt{2}}{\|x_1\|}x_1^\intercal\cdot\Big(\sum_i a_i\xi_i\Big) - 2 \tag{25}$$

Note now that we just need a single element v in $F(x)$ to guarantee (24). recall that $F(x)$ is a subset of $T_\Sigma(x)$ and write

When $(x_1,x_2) \to (0,\mathsf{c}) \implies \gamma_d \to 0$ and with an appropriately large exponent κ ($\gamma^{\kappa-1}$ still dominates over κ) we can make the first two terms of (25) arbitrarily small. Thus

$$\left\langle \frac{\zeta}{\|x_1\|}, \frac{v}{\|x_1\|}\right\rangle \to \frac{\sqrt{2}}{\|x_1\|}x_1^\intercal\cdot\Big(\sum_i a_i\xi_i\Big) - 2$$

from below as $(x_1,x_2) \to (0,\mathsf{c})$. For that limit we can write

$$\frac{\sqrt{2}}{\|x_1\|}x_1^\intercal\cdot\Big(\sum_i a_i\xi_i\Big) - 2 \leq \sqrt{2} - 2 < 0$$

because ξ_i is a unit vector and $\sum_i a_i = 1$. Thus, not only do trajectories exists that once they hit $\Sigma \cap V^{-1}(\mathsf{c})$ stay in that set, but these trajectories will have to evolve until they reach $(x_1,x_2) \to (0,\mathsf{c})$. By continuity of the right hand side of (25) and Ψ, the largest invariant set in Σ, where $\Psi = 0$ can only be where $x_1 = 0$, $\varphi(\mathsf{c}) = 0$.

REFERENCES

[1] D. Edwards, T. Bean, D. Odell, and M. Anderson, "A leader-follower algorithm for multiple AUV formations," in *Proceedings of the IEEE/OES Autonomous Underwater Vehicles Conference*, (Sebasco ME, USA), pp. 40–46, 2004.

[2] E. M. Arkin, S. P. Fekete, and J. S. Mitchell, "Approximation algorithms for lawn mowing and milling," *Computational Geometry*, vol. 17, no. 1–2, pp. 25–50, 2000.

[3] Y. Yamamoto and X. Yun, "Coordinating locomotion and manipulation of a mobile manipulator," *IEEE Transaction on Automatic Control*, vol. 39, no. 6, pp. 1326–1332, 1994.

[4] J. R. Lawton, R. W. Beard, and B. J. Young, "A decentralized approach to formation manneuvers," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 6, pp. 933–941, 2003.

[5] R. Beard, J. Lawton, and F. Hadaegh, "A feedback architecture for formation control," in *Proceedings of the American Control Conference*, vol. 6, pp. 4087 –4091 vol.6, 2000.

[6] P. Ögren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1292–1302, 2004.

[7] S. Mastellone, D. M. Stipanović, C. Graunke, K. Intlekofer, and M. W. Spong, "Formation control and collision avoidance for multi-agent nonholonomic systems: Theory and experiments," *International Journal of Robotics Research*, vol. 13, pp. 107–126, 2008.

[8] D. M. Stipanović, P. F. Hokayem, M. W. Spong, and D. D. Sïljak, "Avoidance control for multi-agent systems," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 129, pp. 699–707, 2007.

[9] R. Olfati-Saber and R. M. Murray, "Distributed cooperative control of multiple vehicle formations using structural potential functions," in *IFAC World Congress*, 2002.

[10] P. Ögren and N. Leonard, "Obstacle avoidance in formation," in *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 2492 – 2497 vol.2, 2003.

[11] H. Tanner and A. Kumar, "Towards decentralization of multi-robot navigation functions," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4132 – 4137, 2005.

[12] M. M. Zavlanos and K. J. Kyriakopoulos, "Decentralized motion control of multiple mobile agents," in *11th Mediterranean Conference on Control and Automation*, 2003.

[13] D. V. Dimarogonas and K. J. Kyriakopoulos, "A feedback stabilization and collision avoidance scheme for multiple independent nonholonomic non-point agents," in *Proceedings of IEEE International Symposium on Intelligent Control*, 2005.

[14] D. Dimarogonas, K. Kyriakopoulos, and D. Theodorakatos, "Totally distributed motion control of sphere world multi-agent systems using decentralized navigation functions," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2430 –2435, 2006.

[15] M. De Gennaro and A. Jadbabaie, "Formation control for a cooperative multi-agent system using decentralized navigation functions," in *American Control Conference*, pp. 1346 – 1351, 2006.

[16] T. Summers, C. Yu, B. Anderson, and S. Dasgupta, "Formation shape control: Global asymptotic stability of a four-agent formation," in *Proceedings of the 48th IEEE Conference on Decision and Control*, pp. 3002 –3007, 2009.

[17] K.-K. Oh and H.-S. Ahn, "Distance-based sequential formation control of mobile agents by using motion primitives," in *IEEE International Symposium on Intelligent Control*, pp. 1464 –1469, 2010.

[18] Y. Cao and W. Ren, "Distributed coordinated tracking with reduced interaction via a variable structure approach," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 33 –48, 2012.

[19] S. Coogan and M. Arcak, "Formation control with size scaling using relative displacement feedback," *American Control Conference*, 2012.

[20] B. Anderson, C. Yu, S. Dashgupta, and A. Morse, "Control of a three-coleader formation in the plane," *Systems and Control Letters*, vol. 56, pp. 573–578, 2007.

[21] S. Loizou and K. Kyriakopoulos, "Navigation of multiple kinematically constrained robots," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 221 –231, 2008.

[22] H. Tanner, S. Loizou, and K. Kyriakopoulos, "Nonholonomic navigation and control of cooperating mobile manipulators," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 53 – 64, 2003.

[23] S. Loizou and K. Kyriakopoulos, "Closed loop navigation for multiple non-holonomic vehicles," in *IEEE International Conference on Robotics and Automation*, pp. 4240–4245, 2003.

[24] H. Su, X. Wang, and Z. Lin, "Flocking of multi-agents with a virtual leader," *IEEE Transaction on Automatic Control*, vol. 54, no. 2, pp. 293–307, 2009.

[25] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 905–908, 2001.

[26] R. Vidal, O. Shakernia, and S. Sastry, "Formation control of nonholonomic mobile robots with omnidirectional visual servoing and motion segmentation," in *IEEE International Conference on Robotics and Automation*, (Taipai, Taiwan), pp. 584–589, 2003.

[27] W. Dong, "Flocking of multiple mobile robots flocking of multiple mobile robots based on backstepping," *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, vol. 41, no. 2, pp. 414–424, 2011.

[28] F. Xie and R. Fierro, "Stabilization of nonholonomic robot formations: A first-state contractive model predictive control approach," *Journal of Computing and Information Technology*, vol. 1, pp. 37—50, 2009.

[29] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, "Leader–follower formation control of nonholonomic mobile robots leader–follower formation control of nonholonomic mobile robots with input constraints," *Automatica*, vol. 44, pp. 1343–1349, 2008.

[30] G. Mariottini, F. Morbidi, D. Prattichizzo, G. Pappas, and K. Daniilidis, "Leader-follower formations: Uncalibrated vision-based localization and control," in *IEEE International Conference on Robotics and Automation*, (Rome, Italy), pp. 2403–2408, 2007.

[31] T. van der Broek, N. van de Wouw, and H. Nijmeijer, "Formation control of unicycle mobile robots: a virtual structure approach," in *IEEE Conference on Decision and Control*, (Sanghai, China), pp. 8328–8333, 2009.

[32] J. Ghommam, H. Mehrjerdi, M. Saad, and F. Mnif, "Formation path following control of unicycle-type mobile robots," in *Robotics and Autonomous Systems*, vol. 58, pp. 727—736, 2010.

[33] D. Dimarogonas and K. Kyriakopoulos, "A connection between formation control and flocking behavior in nonholonomic multiagent systems," in *IEEE International Conference on Robotics and Automation*, (Orlando, FL), pp. 940–945, 2006.

[34] T. Balch and R. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926 –939, 1998.

[35] D. D. Paola, R. D. Asmundis, A. Gasparri, and A. Rizzo, "Decentralized topology control for robotic networks with limited field of view sensor," *American Control Conference*, pp. 3167 – 3172, 2012.

[36] F. Cucker and S. Smale, "On the mathematics of emergence," *Japanesse Journal of Mathematics*, vol. 2, pp. 197–227, 2007.

[37] C. Yoshioka and T. Namerikawa, "Formation control of nonholonomic multi-vehicle systems based on virtual structure," in *Proceedings of the 17th World Congress of The International Federation of Automatic Control*, pp. 5149– 5154, 2008.

[38] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[39] M. Zavlanos, H. Tanner, A. Jadbabaie, and G. Pappas, "Hybrid control for connectivity preserving flocking," *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2869 –2875, 2009.

[40] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in fixed and switching networks," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 863–867, 2007.

[41] S. A. P. Quintero, G. E. Collins, and J. ao P. Hespanha, "Flocking with fixed-wing uavs for distributed sensing: A stochastic optimal control approach," in *American Control Conference*, (in print), 2013.

[42] W. Dong, Y. Guo, and J. Farrell, "Formation control of nonholonomic mobile robots," in *American Control Conference*, (Minneapolis, MN), pp. 5602–5607, 2006.

[43] N. Moshtagh, A. Jadbabaie, and K. Daniilidis, "Distributed geodesic control laws for flocking of nonholonomic agents," in *44th IEEE Conference on Decision and Control*, (Seville, Spain), pp. 2835–2840, 2005.

[44] Y. Lan, G. Yan, and Z. Lin, "Synthesis of distributed control of coordinated path following based on hybrid approach," *IEEE Transactions on Automatic Control*, vol. 56, no. 5, pp. 1170–1175, 2011.

[45] R. Chabcheloo, A. Aguiar, A. Pascoal, C. Silvestre, I. Caminer, and J. Hespanha, "Coordinated path following in the presence of communication losses and time delays," *SIAM Journal in Control Optimization*, vol. 48, no. 1, pp. 234–265, 2009.

[46] E. Børhaug, A. Pavlov, E. Panteley, and K. Y. Pettersen, "Straight line path following for formations of straight line path following for formations of underactuated marine surface vessels," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 493–506, 2011.

[47] O. Sørdalen and C. Canudas de Wit, "Exponential control law for a mobile robot: extension to path following," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 6, pp. 837 –842, 1993.

[48] D. Nelson, D. Barber, T. McLain, and R. Beard, "Vector field path following for miniature air vehicles," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 519 –529, 2007.

[49] E. Frew, D. Lawrence, C. Dixon, J. Elston, and W. Pisano, "Lyapunov guidance vector fields for unmanned aircraft applications," in *American Control Conference*, pp. 371–376, 2007.

[50] H. Tanner and D. Christodoulakis, "Decentralized cooperative control of heterogeneous vehicle groups," *Robotics and Autonomous Systems*, vol. 55, no. 11, pp. 811–823, 2007.

[51] V. Concalves, L. Pimenta, C. Maia, B. Dutra, and G. Pereira, "Vector fields for robot navigation along time-varying curves in n-dimensions," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 647–659, 2010.

[52] M.-Y. Cheng and Y.-H. Wang, "Velocity field construction of contour following tasks represented in NURBS form," *IEEE Transaction on Automatic Control*, vol. 54, no. 10, pp. 2405–2410, 2009.

[53] P. Iscold, G. Pereira, and L. Tôrres, "Development of a hand-launched small UAV for ground reconnaissance," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 1, pp. 335–348, 2010.

[54] L. Pimenta, G. Pereira, and R. Mesquita, "Fully continuous vector fields for mobile robot nativation on sequences of discrete triangular regions," in *IEEE International Conference on Robotics and Automation*, pp. 1992–1997, 2007.

[55] B. Yordanov, G. Batt, and C. Belta, "Model checking discrete-time piecewise affine systems: application ot gene networks," in *European Control Conference*, 2007.

[56] V. Kallem, A. Komoroski, and V. Kumar, "Sequential composition for navigating a nonholonomic cart in the presence of obstacles," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1152–1159, 2011.

[57] L. Carcciolo, A. De Luca, and S. Iannitti, "Trajectory tracking control of a four-wheel differentially driven mobile robot," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2632–2638, 1999.

[58] K. Kozlowski and D. Pazderski, "Modeling and control of a 4-wheel skid-steering mobile robot," *International Journal of Applied Mathematics and Computer Science*, vol. 14, no. 4, pp. 477–496, 2004.

[59] G. Campion, G. Bastin, and B. D. Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 1, pp. 47–62, 1996.

[60] H. Tanner and A. Boddu, "Multi-agent navigation functions revisited," *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1346–1359, 2012.

[61] A. V. Oppenheim, R. W. Schafer, and J. R. Buch, *Discrete Time Signal Processing*. Prentice Hall, 1998.

[62] J. Cortes, "Discontinuous dynamical systems: A tutorial on solutions, nonsmooth analysis, and stability.," *IEEE Control Systems Magazine*, vol. 28, no. 3, pp. 36–73, 2008.

[63] J. Fraden, *HandBook of Modern Sensors*. Springer, 2004.

[64] D. Shevitz and B. Paden, "Lyapunov stability theory of nonsmooth systems," *IEEE Transactions on Automatic Control*, pp. 1910 – 1914, 1994.

[65] F. H. Clarke, *Optimization and Nonsmooth Analysis*. Society for Industrial and Applied Mathematics, 1990.

[66] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[67] E. P. Ryan, "An integral invariance principle for differential inclusions with applications in adaptive control," *SIAM Journal in Control Optimization*, vol. 36, pp. 960–980, 1998.

[68] J. P. LaSalle, *The Stability of Dynamical Systems*. Society for Industrial and Applied Mathemtics, 1976.

[69] L.-W. Tsai, *Robot Analysis*. John Wiley & Sons,Inc, 1998.

[70] H. K. Khalil, *Nonlinear Systems*. Prentice Hall, 2002.