

Experiments in Multirobot Coordination

Joshua A. Marshall^{a,*} Terence Fung^a Mireille E. Broucke^a
Gabriele M. T. D'Eleuterio^b Bruce A. Francis^a

^a*Department of Electrical & Computer Engineering, University of Toronto,
10 King's College Rd, Toronto, ON, Canada M5S 3G4*

^b*Institute for Aerospace Studies, University of Toronto, 4925 Dufferin St,
Toronto, ON, Canada M3H 5T6*

Abstract

Consequent to previously published theoretical work by Marshall, Broucke, and Francis, this paper summarizes the apparatus and results of multirobot coordination experiments conducted at the University of Toronto Institute for Aerospace Studies. These experiments successfully demonstrated the practicality of cyclic pursuit as a distributed control strategy for multiple wheeled-robot systems. Moreover, the pursuit-based coordination algorithm was found to be surprisingly robust in the presence of unmodelled dynamics and delays due to sensing and information processing. The findings of this research not only bode well for continuing research on pursuit-based coordination strategies, but also for other cooperative multirobot control techniques employing similar local interactions.

Key words: Autonomous robots, Multiagent coordination, Cooperative control

1 Introduction

In [1,2], Marshall et al. introduced the notion of *pursuit* as a technique for coordinating the motions of multiple wheeled-robots in the plane. More specifically, *cyclic pursuit* was studied in a purely theoretical way as a means for achieving certain regular geometric formations for a system of identical kinematic unicycles. This approach is particularly simple to implement in that the n robots are ordered such that robot i pursues robot $i + 1$, modulo n .

* Corresponding author. Tel.: +1 416 978 6289; fax: +1 416 978 0804.
Email address: marshall@control.toronto.edu (Joshua A. Marshall).

The current research on pursuit strategies for multirobot systems is partially motivated by the prevalence of similar distributed control algorithms found in nature [3–5]. Moreover, this type of formation strategy might have potential application in the deployment of distributed sensor arrays, enabling scientists to collect simultaneous seismological, meteorological, or other pertinent environmental data on planetary exploration missions [6]. More generally, multi-robot systems might find application in terrestrial, space, and oceanic exploration, military surveillance and rescue missions, or even automated highway systems [7]. Hence, the question of how to prescribe desired global behaviors through the design of only simple and local interactions is of significant and practical interest. The relevance of this problem is punctuated by the recent and growing amount of research in this field; see, for example, [8–10,2,11] and references therein.

This paper is organized as follows. Section 1.1 provides a brief review of the relevant background theory on which the experiments described in this paper are based. Section 1.2 introduces the purpose of our experiments and indicates their significance. An overview of the experimental setup and hardware is provided in Sec. 2, while Sec. 3 describes the engineering design and implementation issues. Finally, detailed descriptions of the experiments and ensuing observations are located in Sec. 4.

1.1 Theoretical Background

Consider a system of n wheeled robots in the plane, each modelled as a kinematic unicycle with nonlinear state model

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos \theta_i & 0 \\ \sin \theta_i & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}, \quad i = 1, 2, \dots, n, \quad (1)$$

where $(x_i, y_i) \in \mathbb{R}^2$ is the i -th robot's position, $\theta_i \in (-\pi, \pi]$ is its orientation, and $(v_i, \omega_i) \in \mathbb{R}^2$ are control inputs. Let α_i denote the difference between the i -th robot's heading and the heading that would take it directly towards its target, $i + 1$ (as in Fig. 1). An intuitive pursuit law for Eq. (1) is to assign robot i 's angular speed ω_i in proportion to the heading error, α_i . Following [2], we consider the case when the robots move with the same fixed forward speed, v_R . In particular, we are interested in the distributed control law

$$v_i = v_R \text{ and } \omega_i = k_\alpha \alpha_i, \quad (2)$$

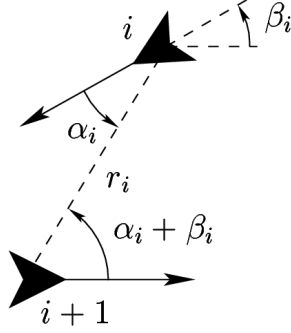


Fig. 1. Relative coordinates, with robot i in pursuit of $i + 1$

where $k_\alpha > 0$ is constant. Simulation and analysis results have shown that asymptotically stable circular pursuit trajectories are achievable, as illustrated for $n = 3$ in the simulation of Fig. 3. The relative coordinates $[r_i, \alpha_i, \beta_i]^\top$ of Fig. 1 prove useful, since in steady state these variables become constant.

Let $z_i \in \mathbb{R}^2$, $i = 1, 2, \dots, n$, and consider the following definition.

Definition 1 (after [12], p. 93) *Let n and $d < n$ be positive integers so that $p := n/d > 1$ is a rational number. Let R be the positive rotation in the plane, about the origin, through angle $2\pi/p$ and let $z_1 \neq 0$ be a point in the plane. Then, the points $z_{i+1} = Rz_i$, $i = 1, 2, \dots, n - 1$ and edges $e_i = z_{i+1} - z_i$, $i = 1, 2, \dots, n$, define a generalized regular polygon, which is denoted $\{p\}$.*

By this definition, $\{p\}$ can be interpreted as a directed graph with vertices z_i (not necessarily distinct) connected by edges e_i as determined by the ordering of points. Since p is rational, the period of R is finite and, when n and d are coprime, this definition is equivalent to the well-known definition of a regular polygon as a polygon that is both *equilateral* and *equiangular*. Moreover, when $d = 1$, $\{p = n\}$ is an *ordinary* regular polygon (i.e., its edges do not cross one another). However, when $d > 1$ is coprime to n , $\{p\}$ is a *star* polygon since its sides intersect at certain extraneous points, which are not included among the vertices [12, pp. 93–94]. If n and d have a common factor $m > 1$, then $\{p\}$ has $\tilde{n} = n/m$ distinct vertices and \tilde{n} edges traversed m times. Note that the trivial case when $d = n$ has not been included since this corresponds to the geometrically uninteresting situation where the vertices are all coincident.

Fig. 2 illustrates some example possibilities for $\{p\}$ when $n = 9$. In the first instance, $\{9/1\}$ is an ordinary polygon. In the second instance, $\{9/2\}$ is a star polygon since 9 and 2 are coprime. In the third, the edges of $\{9/3\}$ traverse a $\{3/1\}$ polygon 3 times, because $m = 3$ is a common factor of both 9 and 3.

Consider the $n = 3$ robots of Fig. 3. Under the pursuit law Eq. (2), they converge to a $\{3/1\}$ generalized regular polygon formation. Indeed, in [2, Sec. IV] it was revealed that, subject to the inputs Eq. (2), certain generalized

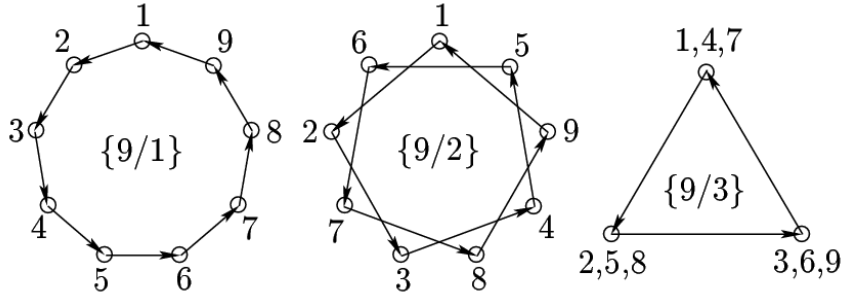


Fig. 2. Example generalized regular polygons $\{9/d\}$, $d \in \{1, 2, 3\}$

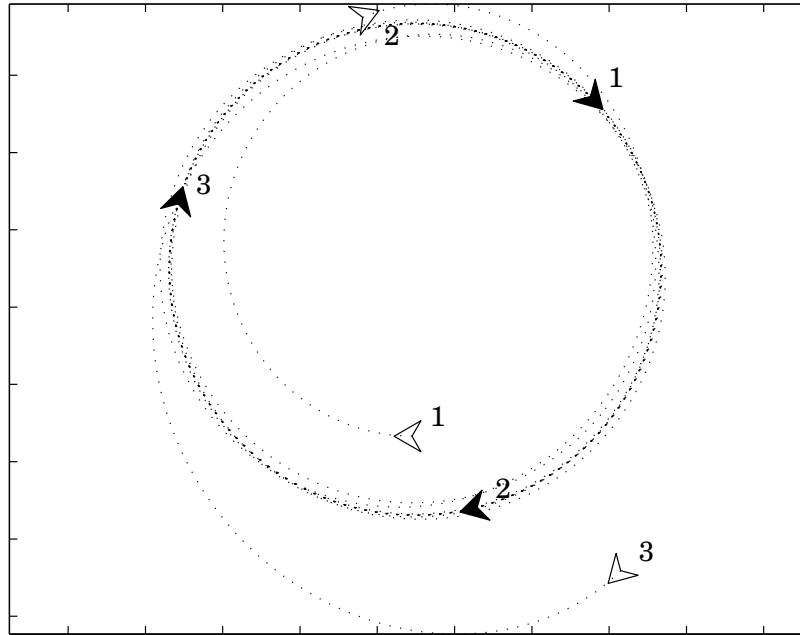


Fig. 3. Three unicycles subject to control law Eq. (2), with $k_\alpha = 0.6$

regular polygon formations are locally asymptotically stable, while others are not. Table 1 lists all possible equilibrium formations and gives their stability.

1.2 Experimental Purpose

During recent years, several university-based research groups have developed testbeds for experimentation in multirobot control. In most cases, these multivehicle testbeds are designed without a specific set of control experiments in mind, thus not for the sole purpose of validating a particular theory. Some examples are the *MIT Multivehicle Testbed* [13], Caltech's *MVWT-II Multivehicle Wireless Testbed* [14], the *Brigham Young Unmanned Air Vehicle Testbed* [15], and the University of Illinois's *HoTDeC, or Hovercraft Testbed for Decentralized Control* [16], to name only a few.

Table 1
Equilibrium polygons with stable polygons shaded [2]

$d = 1$	2	3	4	5	6
{2/1}	{3/2}	{4/3}	{5/4}	{6/5}	{7/6}
{3/1}	{4/2}	{5/3}	{6/4}	{7/5}	{8/6}
⋮	⋮	⋮	⋮	⋮	⋮
{7/1}	{8/2}	{9/3}	{10/4}	{11/5}	{12/6}
{8/1}	{9/2}	{10/3}	{11/4}	{12/5}	{13/6}
⋮	⋮	⋮	⋮	⋮	⋮
{17/1}	{18/2}	{19/3}	{20/4}	{21/5}	{22/6}
{18/1}	{19/2}	{20/3}	{21/4}	{22/5}	{23/6}
⋮	⋮	⋮	⋮	⋮	⋮
{49/1}	{50/2}	{51/3}	{52/4}	{53/5}	{54/6}
{50/1}	{51/2}	{52/3}	{53/4}	{54/5}	{55/6}
⋮	⋮	⋮	⋮	⋮	⋮

Likewise, a fleet of so-called *Argo Rovers* (the allusion being to the Greek myth of Jason, the Argonauts and the Golden Fleece, since names belonging to the Argonauts have been bestowed on rovers of the fleet) have been constructed at the Space Robotics Laboratory of the University of Toronto Institute for Aerospace Studies (UTIAS). The robots were designed to be capable of lengthy autonomous operation and have each been equipped with a host of sensing, communication, and actuation devices. For more specific details, see [17].

Since the theoretical results summarized in Sec. 1.1 were based on ideal kinematic unicycles, one might naturally question whether the intuitive control law Eq. (2) has more general applicability (e.g., to real robots, possessing non-trivial dynamics, such as the *Argo Rovers*). Also, despite the growing amount of theoretical research on cooperative control strategies employing local interaction-based techniques, there are relatively few instances of experimental research validating their worth and reporting on the challenges of implementation. Therefore, the purpose of our experiments is twofold:

- i. Determine if the theoretical results of [2], obtained for kinematic unicycles, can be observed in practise using the four-wheeled *Argo Rovers*;
- ii. Investigate the practicality of Eq. (2) as a coordination strategy given real hardware restrictions (e.g., processing delays, sensor limitations).

A brief description of our experimental procedure, a detailed summary of the



Fig. 4. An Argo Rover in the MarsDome at UTIAS

results, and a discussion of our observations and their significance follow.

2 Overview of the Rovers

One of the Argo Rovers is shown in Figure 4, posing in UTIAS's indoor-outdoor testing facility called MarsDome. Built using the Tamiya TXT-1 4×4 Pick-up chassis, the rovers were designed to be fully autonomous mobile robots suitable for outdoor use in reasonable environmental conditions.

2.1 *Microelectronics and Software*

Each rover possesses a 700 MHz Pentium[®] III processor-based computer (Cell Computing[®] Plug-N-Run) with a 1 GB microdrive, 256 MB of RAM, 2 PCMCIA slots, 2 USB ports, and runs the Debian-Linux operating system. All Linux-based software is capable of accessing the onboard sensors and actuators by way of a Siemens C164 20 MHz 16-bit microcontroller (for processing low-level hardware routines). All custom software for the rovers is developed using the C/C++ languages in Linux. Furthermore, the rovers are each fitted with a wireless Ethernet PCMCIA card used for remote software development, operation, and potentially for direct communication between the rovers.

2.2 Power Delivery System

In order to conserve payload space and to lower the rover's center of gravity (i.e., for improved stability), each rover is powered by 1.2 V Saft Nickel-Metal Hydride (NiMH) VH F battery cells, ten of which are located (in series) inside each rubber tire. Current is subsequently delivered to the individual rover systems by way of a custom designed circular slip-ring within each wheel hub.

2.3 Motion Actuators and Encoders

Front and rear wheel steering axis angles are adjustable independently via servomotor driven mechanisms (Hitec model HS-300) at each wheel axis. Thus, each wheel axis angle is directly specifiable (in software) through a servomotor input command $u_\phi \in [-1, 1]$. For example, if the rear axis is fixed with a zero steering angle then a servomotor input of $u_\phi \approx 0$ at the front wheel axis would result in straight-line driving, while $u_\phi > 0$ and $u_\phi < 0$ would correspond to right and left car-like steering (in the forward direction), respectively.

Moreover, each rover is propelled at all four wheels by a geared throttle motor (Alan's Models part number 1105/7; gear ratio 6 : 1), allowing the rover to easily move forward or backward. The rovers are capable of traveling at speeds of not much more than 0.5 m/s with a minimum turning radius of approximately 0.65 m. The throttle is specifiable (in software) through a motor input command $u_f \in [-1, 1]$. The robot is stopped when $u_f \approx 0$ and moves forward and backward for $u_f > 0$ and $u_f < 0$, respectively.

The rovers are all mounted with US Digital Corporation rotary optical encoders (generating 512 cycles per shaft turn) in the hub of each wheel.

2.4 Camera-based Vision Systems

Each rover is equipped with two CCD array cameras (Logitech[®] QuickCam[®] Pro 3000) capable of acquiring up to 640×480 pixel resolution images at a frequency of 30 Hz. Furthermore, each camera is fixed to a stereovision head using custom supports allowing for individual pan and tilt by way of servomotor mechanisms.

3 Design and Implementation

This section describes, in detail, the hardware and software engineering designs used to fulfil the experimental purpose described in Sec. 1.2.

3.1 Rover Dynamics

As a design tool and, perhaps more importantly, to illustrate how significantly different the Argo Rovers are from ideal kinematic unicycles, a simple model of the rover dynamics is first developed. In doing so, it is assumed that each wheel rolls without laterally slipping, thus having similar nonholonomic characteristics to the already studied kinematic unicycles.

Owing to limited workspace in the laboratory environment, the rovers were operated with their front and rear wheel axes *locked* for tightest turning. Let ϕ_f and ϕ_r denote the front and rear wheel axis angles. In practise, locking of the wheel axes was accomplished by assigning the appropriate servomotor inputs u_{ϕ_f} and u_{ϕ_r} so that $\phi_f = -\phi_r =: \phi$. Therefore, in this case, each rover's configuration can be described by the vector of coordinates $q = [x, y, \theta, \phi]^\top$. If the front and rear wheel-axis pairs are each modelled as just a single wheel (see Fig. 5), then one can use the nonholonomic constraints, which act at each wheel to prevent it from slipping laterally, to develop the kinematic model

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \phi \cos \theta \\ \cos \phi \sin \theta \\ \frac{1}{l} \sin \phi \\ 0 \end{bmatrix} v_f + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \omega_\phi, \quad (3)$$

where the rover is driven by a forward velocity input v_f acting at the front wheel-axis point p_f (or at the axis-point p_r — since the wheels are locked it does not matter) in the direction of the wheel and is steered by an angular steering velocity input ω_ϕ . The length l is approximately 0.3 m.

The kinematic model Eq. (3) constitutes a first step in describing the rover as a mechanical system. In reality, the rovers have mass, and thus dynamics. One can extend the above kinematic model to include dynamic effects due to translation and rotation of the rover's body mass. Let m denote the rover's mass (approximately 15 kg), I_p its body moment of inertia about the point (x, y) in Fig. 5, and I_s the effective inertia that needs to be overcome by the steering actuator (assumed constant). Suppose one ignores friction and the (minor) inertial effects due to rotation of the wheels. Then, the dynamical

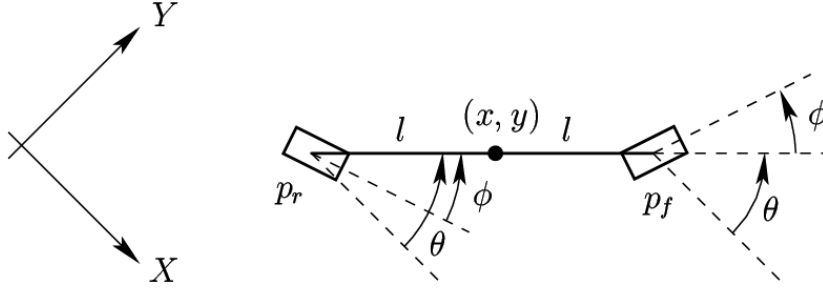


Fig. 5. Body centred axis model with wheel-axes locked

equations of motion are

$$\dot{x} = v_f \cos \phi \cos \theta \quad (4a)$$

$$\dot{y} = v_f \cos \phi \sin \theta \quad (4b)$$

$$\dot{\theta} = v_f \frac{1}{l} \sin \phi \quad (4c)$$

$$\dot{\phi} = \omega_\phi \quad (4d)$$

$$\dot{v}_f = \left(m \cos^2 \phi + \frac{1}{l^2} I_p \sin^2 \phi \right)^{-1} \left[v_f \omega_\phi \left(m - \frac{1}{l^2} I_p \right) \cos \phi \sin \phi + f \right] \quad (4e)$$

$$\dot{\omega}_\phi = \tau / I_s, \quad (4f)$$

where f is the throttle force input, divided evenly between the front and rear wheels, acting in the direction of the wheels, and τ is the representative steering torque input. Since, on the real rovers, the steering angle ϕ is directly specifiable by way of the steering servomotor inputs, the steering torque τ of Eq. (4f) is not actually an available input. Instead, the real steering mechanism dynamics are a function of the unmodelled servomotor characteristics, making f and ϕ the assignable inputs (through u_f and u_ϕ , respectfully).

3.2 Speed Regulation

To approximate the control law Eq. (2), it was necessary to equalize the rover speeds. Therefore, a basic speed regulator was designed for each rover using feedback from its four wheel encoders. The actual rolling speed v_f was estimated by numerically differentiating position data acquired from the encoders every $T = 0.1$ s (or 10 Hz). The estimated wheel-speeds were averaged to generate $\hat{v}_f(kT)$, $k = 0, 1, 2, \dots$, at each time step. Because of significant noise in the speed estimates, the differentiated encoder data was low-pass filtered (-3 dB at 1.35 Hz). Based on our knowledge of the model in Sec. 3.1, a PI compensator was implemented digitally using finite-difference approximations and the estimated speed error $e_v(kT) := v_R - \hat{v}_f(kT)$. Thus, for the servomotor input command $u_v \in [-1, 1]$, the employed discrete-time PI controller was

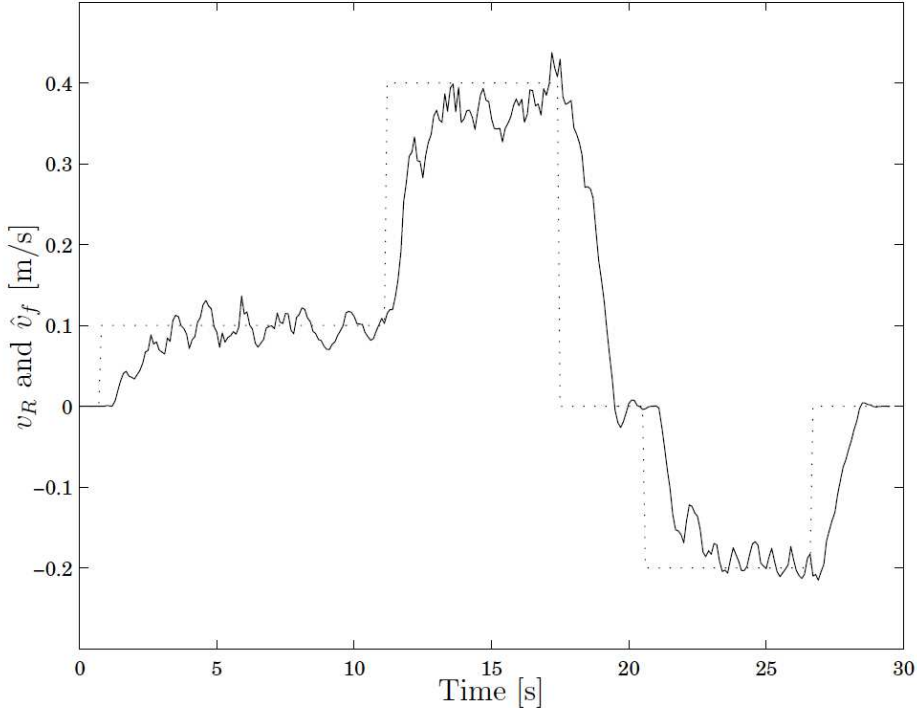


Fig. 6. Example speed regulator response with changing reference speeds

$u_v(kT) = k_P e_v(kT) + u_I(kT)$, where

$$u_I(kT) = u_I(kT - T) + k_I \frac{T}{2} [e_v(kT) + e_v(kT - T)],$$

with $u_I(0) = 0$. Through online tuning experiments, gains of $k_P = 1.5$ and $k_I = 2.5$ were found to work well. An example of the speed regulator response is provided in Fig. 6, as estimated using encoder data. Due to particularities of the transmission (significant play), of the chassis (significant sway), and of the wheels (placement of the batteries), small fluctuations in speed were always present and could be audibly discerned while the rovers were running, even during open-loop driving.

3.3 Multirobot Pursuit using Vision

As described in Sec. 2.3, each rover's steering angle ϕ is *directly* specifiable by way of its steering servomotors, thus the torque input τ of Eq. (4f) is not actually an available input. Therefore, the actual steering mechanism dynamics were ignored and ϕ was itself computed so as to emulate $\dot{\theta} = k_\alpha \alpha$. In other words, using Eq. (4c), one obtains

$$v_{f_i}(kT) = v_R \text{ and } \phi_i(kT) = \arcsin(lk_\alpha \alpha_i(kT)/v_R), \quad (5)$$

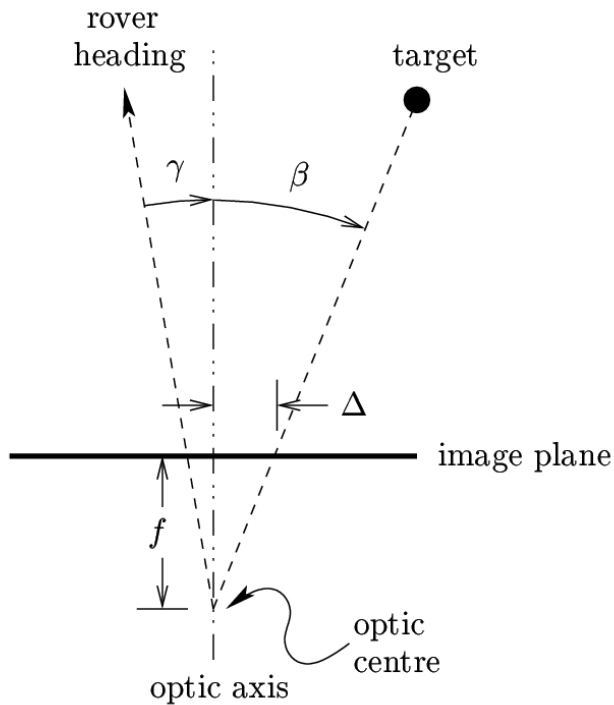


Fig. 7. Camera image plane geometry

for $i = 1, 2, \dots, n$, as an intuitive pursuit law for rovers, which mimics Eq. (2).

The right camera on each rover (although, it matters not which one) was used to acquire 160×120 pixel (low resolution) images, which were then used to estimate the heading error $\alpha(kT)$. The rovers were each suited with a cylinder of different colored cardboard so that ordering of the robots could be accomplished by simply ordering the colors. Target localization was done by scanning the pixels in an acquired image and comparing each pixel's hue value with a preset nominal target value. Those pixels within a specified tolerance of the nominal were recorded and their horizontal positions averaged to compute an estimate of the target's location in the image.

By using knowledge of the camera's focal length and orientation, the heading error $\alpha(kT)$ was estimated as follows. The location of pixels within the specified tolerance of hue value units were recorded and their horizontal positions averaged to compute the horizontal centroid of the cylinder of desired colour in the image (see the distance Δ in Fig. 7, measured in pixels). Let β denote the angle from the camera's optic axis to the point that is Δ pixels from the optic axis (note that Δ changes sign if the target switches sides of the optic axis) such that $\beta = \arctan(\Delta/f)$, where $f \approx 200$ pixels is the focal length of the camera. Let γ denote the angle between the camera's optic axis and the rover's heading θ , from Eq. (4). Therefore, the overall heading error was approximated at each time step $t = kT$ using the formula $\alpha(kT) = \gamma(kT) + \beta(kT)$.

Because the horizontal field-of-view (FOV) of the onboard cameras is relatively low (approximately 34 degrees), in our initial experiments the rovers lost track of their targets very easily. To augment the FOV, its panning servomotor was employed, increasing the FOV to approximately 150 degrees. This was done by adjusting the angle $\gamma(kT)$ in Fig. 7 so as to actively centre the target cylinder in the image plane. Similar to what was done for speed regulation in Sec. 3.2, a PI compensator was used to track the angle $\beta = 0$.

Notice how computation of the control law Eq. (5) for each rover was based on sensing and data processing carried out *locally*, thus in a completely decentralized fashion (i.e., no global positioning techniques were used). This differs, for example, from the overhead camera global positioning system used in [14].

4 Experiments and Observations

A variety of experiments were conducted using teams of two, three, and four rovers. Despite the significant physical differences between ideal kinematic unicycles and the Argo Rover systems, which is a natural conclusion of Sec. 3, the outcome was surprisingly positive. Preliminary experiments were done using two rovers. In this case, the only theoretically possible formation is the $\{2/1\}$ polygon (i.e., two robots diametrically opposite each other on a circular path). To ensure the two rovers were within each other's FOV, it was necessary to rotate their stereovision heads by 90 degrees (e.g., see the rovers in Fig. 8). This rotation was accounted for in software by adding (for rotation left) or subtracting (for rotation right) $\pi/2$ to the angle γ in Fig. 7. The inside body-centred camera was always used to compute α . It was found that, so long as no rover lost the other from its view, the robots always converged to a $\{2/1\}$ -polygon formation.

4.1 Stability of the $\{3/1\}$ -polygon Formation

In theory, the possible equilibrium formations for three robots are the $\{3/1\}$ and $\{3/2\}$ polygons. Although both resemble equilateral triangles, it is the robots' ordering on the circle circumscribed by each polygon that is different. As per Definition 1, a $\{3/1\}$ formation corresponds to the case when the i -th robot's target, $i + 1$, lies at a heading error of $\alpha_i = \pm\pi/3$. Conversely, a $\{3/2\}$ formation corresponds to $\alpha_i = \pm2\pi/3$. Following Table 1, of the two possible equilibria only the $\{3/1\}$ -polygon formation is locally asymptotically stable.

As previously noted, Fig. 3 shows three unit-speed unicycles, subject to Eq. (2), converging to a $\{3/1\}$ formation (the simulation ends after 45 s). On

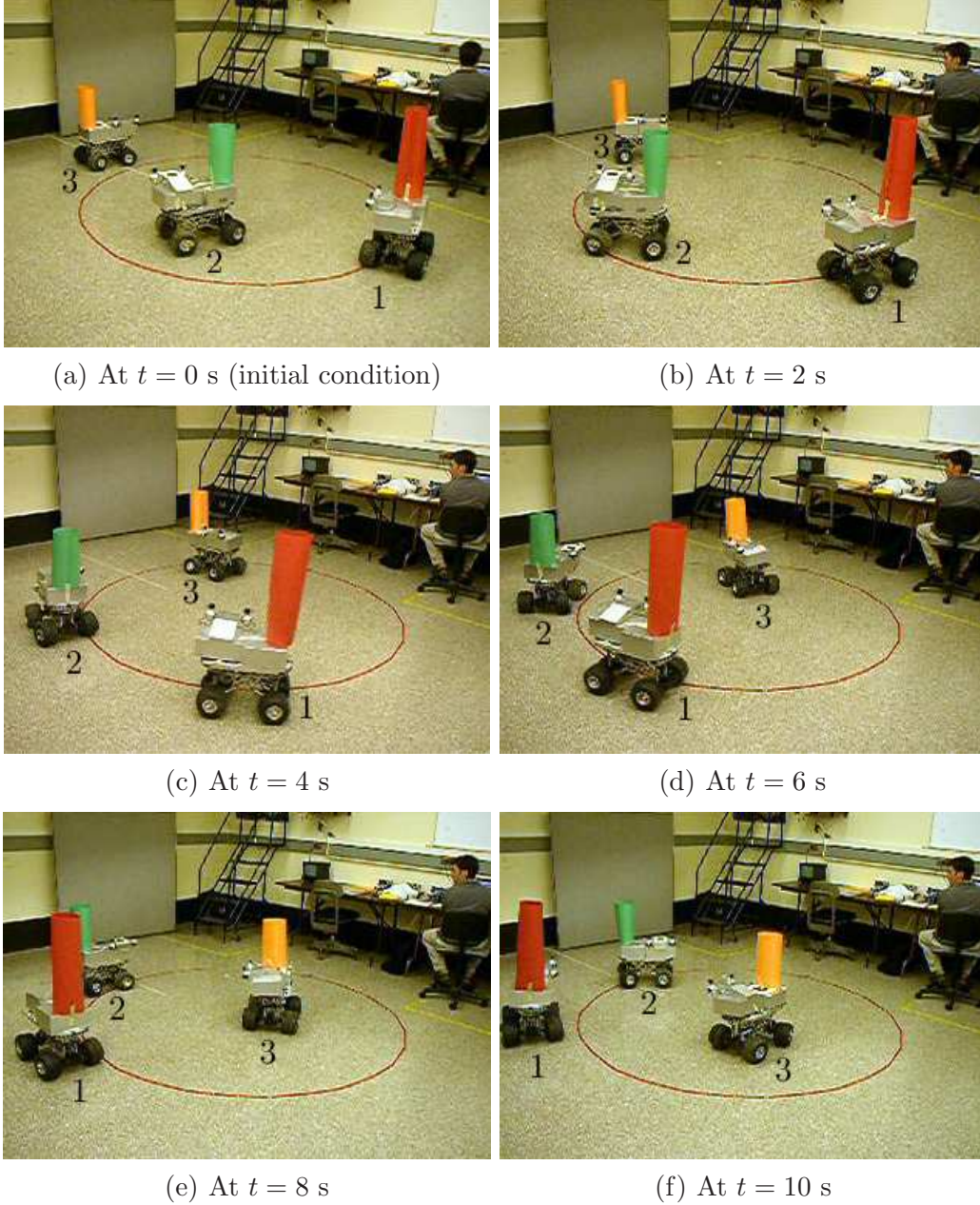


Fig. 8. Generating a $\{3/1\}$ formation with $k_\alpha = 0.2$

the other hand, Fig. 8 presents a sequence of captured images of three Argo Rovers in cyclic pursuit, subject to Eq. (5). Fig. 9 shows the heading errors, α_i , $i = 1, 2, 3$, as a function of time for each of the simulated unicycles in Fig. 3 (dotted lines) and for the actual rovers of Fig. 8 (solid lines). The time axis in Fig. 9 corresponds almost exactly to the times noted for each image frame in Fig. 8. The actual rover heading errors were recorded only every second, although they were computed every 0.1 s. Clearly, owing to their physical differences, the unicycle and rover trajectories should not be expected to match in the transient. However, Fig. 9 shows that their steady-

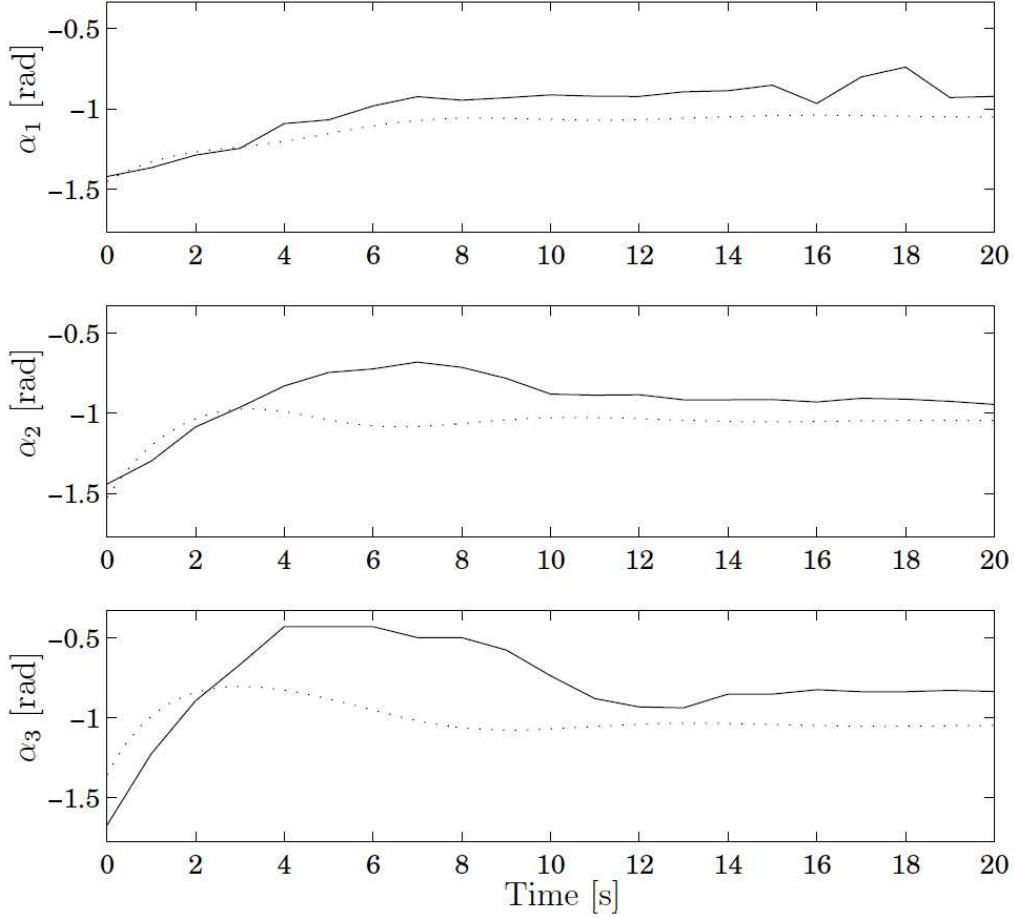


Fig. 9. Kinematic unicycle (dotted lines, cf. Fig. 3) together with actual rover (solid lines, cf. Fig. 8) target heading errors; in both cases, the errors converge to approximately $\alpha_i = -\pi/3$

state behaviors both tend to equally spaced motion around a stationary circle of fixed radius. Convergence of the real rovers to a stable $\{3/1\}$ formation, with $\alpha_i = -\pi/3$, is clear from Fig. 8.

Additional experiments were performed where the rovers were first allowed to achieve a steady-state $\{3/1\}$ formation. Subsequently, one of the rovers was deliberately perturbed from this equilibrium by either altering its heading, halting it temporarily, or slightly changing its location. So long as the rovers were able to maintain their targets within view, the group always returned to a $\{3/1\}$ -polygon configuration, demonstrating its stability as a formation.

4.2 Formation Radius and the $\{4/1\}$ -polygon

Results equivalent to those described above for the $\{3/1\}$ formation were also found using four robots. Fig. 10 shows four rovers maintaining a $\{4/1\}$ -polygon

formation. Furthermore, in [2, Corollary 2] it was proved that the kinematic unicycles traverse a circle of radius $\rho = v_R n / k_\alpha \pi d$ at equilibrium, where $\{n/d\}$ is the formation. Therefore, by increasing (resp. decreasing) the gain k_α we should have expected to observe a proportional decrease (resp. increase) in the radius traversed by the rovers, which was indeed the case. Fig. 10 shows four rovers in cyclic pursuit, each with gain $k_\alpha = 0.3$, after having stabilized to a $\{4/1\}$ -polygon configuration. At approximately $t = 7$ s, the gain k_α was decreased from 0.3 to 0.1 on all the rovers. The sequence of images shows how the rovers continued to maintain a $\{4/1\}$ formation while, at the same time, the polygon’s radius effectively tripled in size. Identical results were also observed for groups of two and three rovers.

4.3 The $\{3/2\}$ -polygon Formation

In theory, the $\{3/2\}$ formation for unicycles is unstable (see Table 1). However, computer simulations suggest that, while maintaining the ordering of robots, *almost*-circular trajectories are achievable for lengthy time periods. Fig. 11a shows a simulation of three unit-speed unicycles that start roughly in the $\{3/2\}$ configuration (the simulation ends after 45 s). Despite the fact that they do not converge to a $\{3/2\}$ polygon, their motion appears to almost maintain a $\{3/2\}$ formation. Interestingly, among the six important eigenvalues associated with the system’s linearization about the $\{3/2\}$ polygon (in relative coordinates; see [2] for details), there is only one complex-conjugate pair of unstable eigenvalues and these eigenvalues lie particularly close to the imaginary axis ($\lambda \approx 0.0419 \pm j1.5303$). If the simulation of Fig. 11a is continued for more than 250 s, the unicycles eventually break their pattern of motion and rearrange themselves into a stable $\{3/1\}$ formation, as illustrated by Fig. 11b.

Fig. 12 presents a sequence of captured images of three rovers in cyclic pursuit (using the same pursuit order as the rovers in Fig. 8). Both the unicycles of Fig. 11 and the rovers of Fig. 12 started close to a $\{3/2\}$ formation. Consequently, their resulting trajectories appear *qualitatively* similar, maintaining the ordering of a $\{3/2\}$ polygon yet never actually converging to a stable formation. If allowed to run for long enough, the rover formation was also seen to “wobble,” as in Fig. 11a. However, even after several minutes, evolution of the rovers into a $\{3/1\}$ -polygon formation was never observed¹, unlike what happens in simulation for unicycles (cf. Fig. 11b). Nevertheless, it is clear from Fig. 12 that the $\{3/2\}$ polygon is not asymptotically stable for rovers, as predicted by the theory for unicycles.

¹ On the other hand, this type of maneuver was likely not even possible given the limited field-of-view of the cameras and the fact that a protocol for collision avoidance was not (yet) implemented.

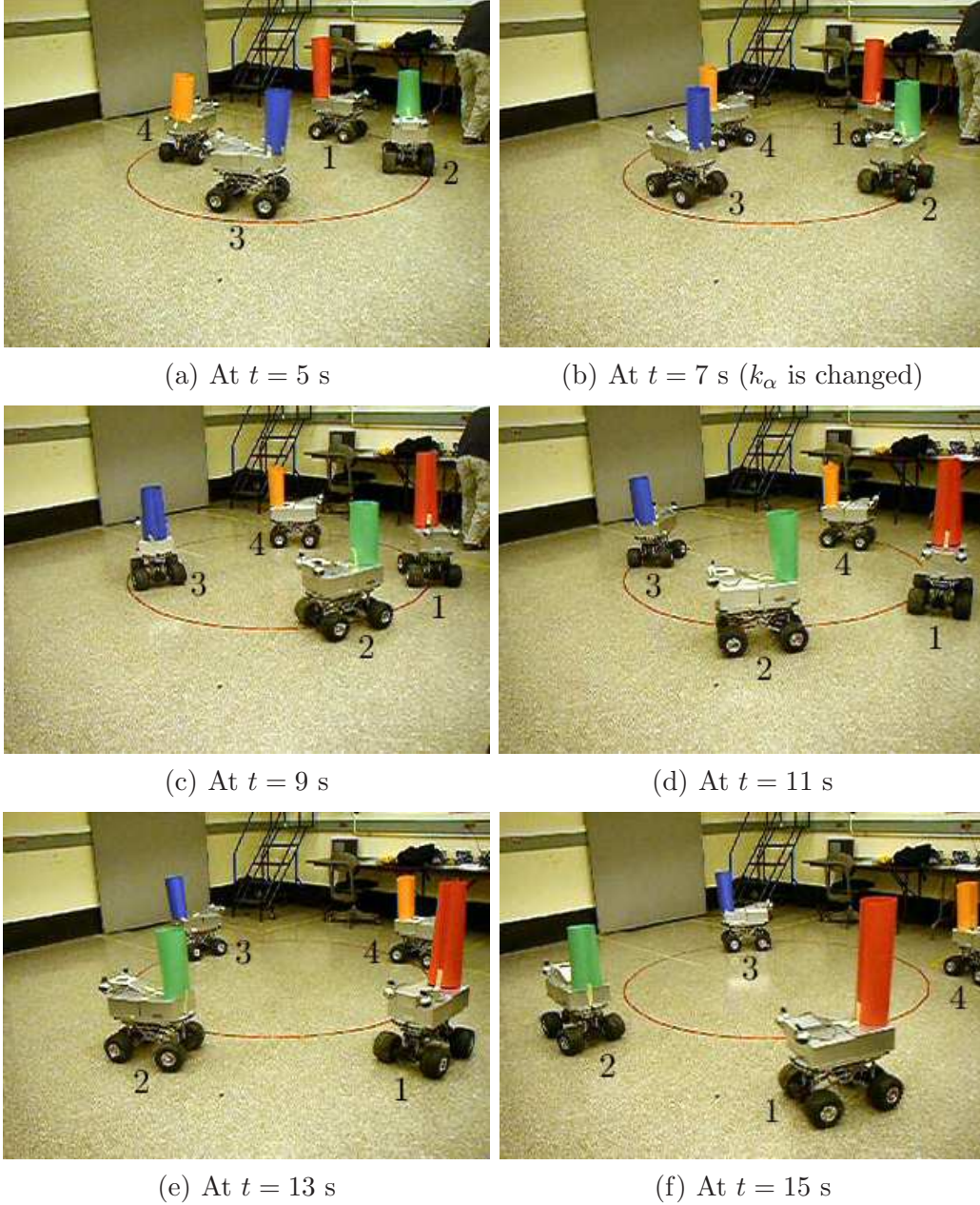
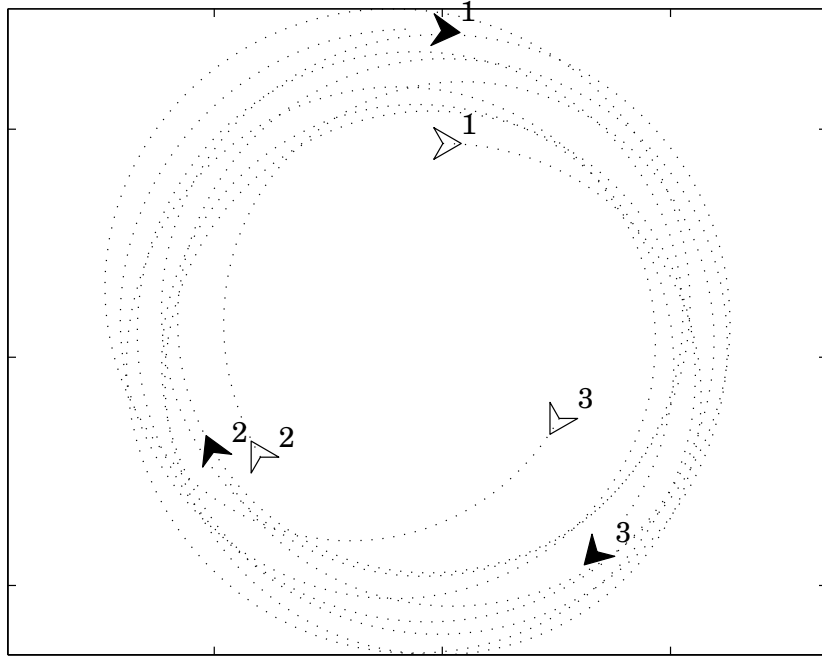
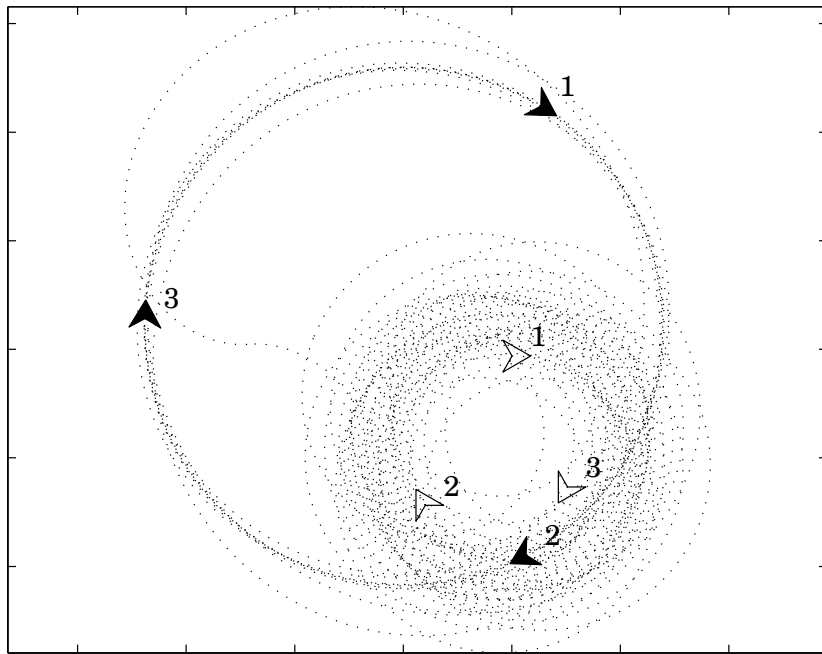


Fig. 10. A $\{4/1\}$ formation after k_α is changed from 0.3 to 0.1

Similar to Fig. 9, in Fig. 13 the heading errors α_i , $i = 1, 2, 3$, have been plotted as a function of time for each of the simulated unicycles in Fig. 11 (dotted lines) and for the actual rovers of Fig. 12. The time axis in Fig. 13 corresponds almost exactly to the times noted for each image frame in Fig. 12. Again, owing to their physical differences, the unicycle and rover trajectories should not be expected to match. However, Fig. 13 shows how their behaviors are qualitatively the same, with oscillations appearing in the heading errors of both the unicycles and the rovers.



(a) After 45 s



(b) After 300 s

Fig. 11. Unicycles demonstrating a $\{3/2\}$ almost-stable formation with $k_\alpha = 0.2$; the initial conditions are the same for both (a) and (b)

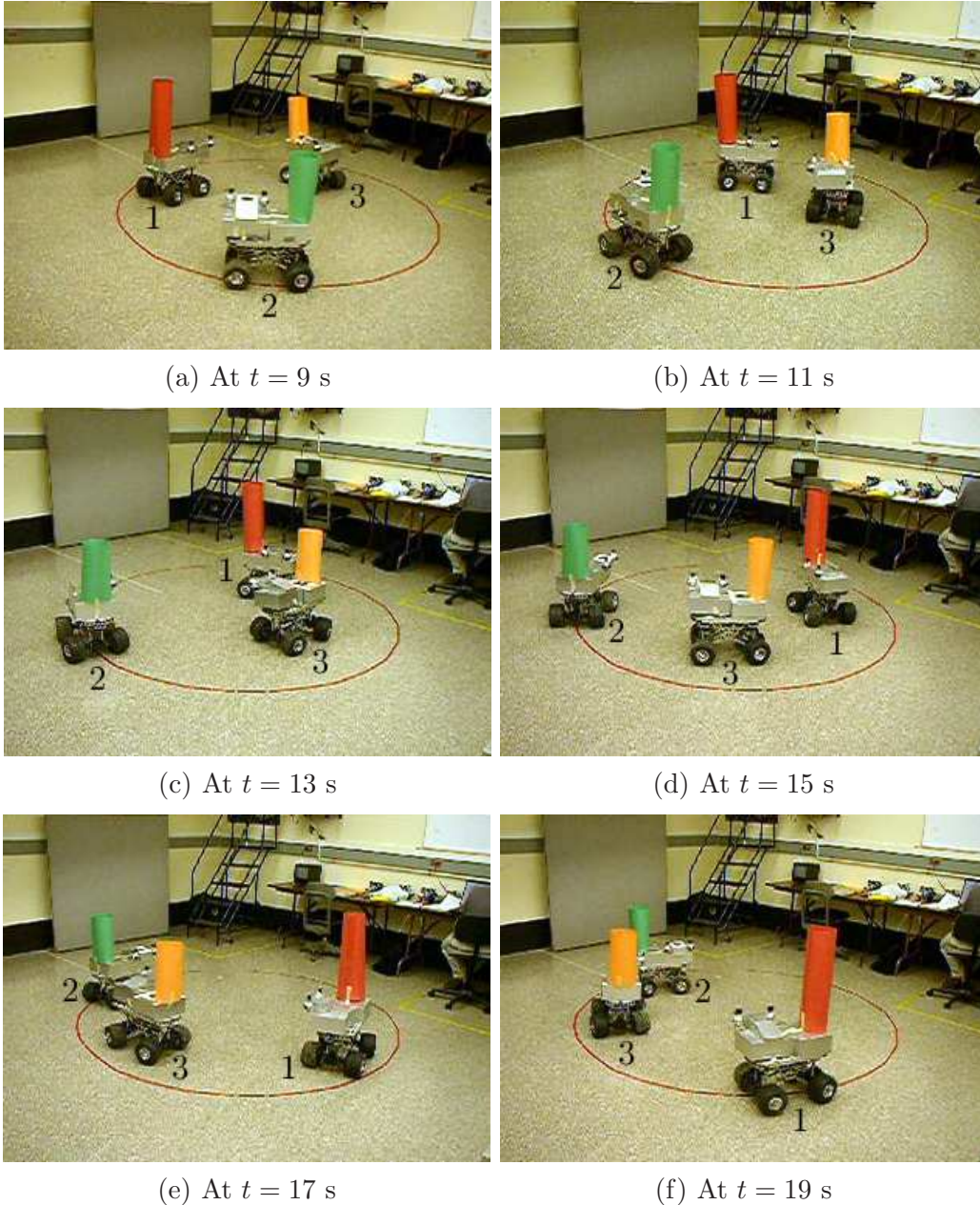


Fig. 12. A $\{3/2\}$ semi-stable formation with $k_\alpha = 0.1$

5 Conclusions

In this paper, details concerning the apparatus and results of multirobot pursuit experiments have been presented. By adapting the hardware and developing software for the existing Argo Rovers, experiments were conducted using groups of two, three, and four rovers, the purpose of which was to determine whether the theoretical results obtained in [2] could be applied in practise to real systems distinct from ideal kinematic unicycles. Given the physical differences between unicycles and the Argo Rovers, and that there were delays

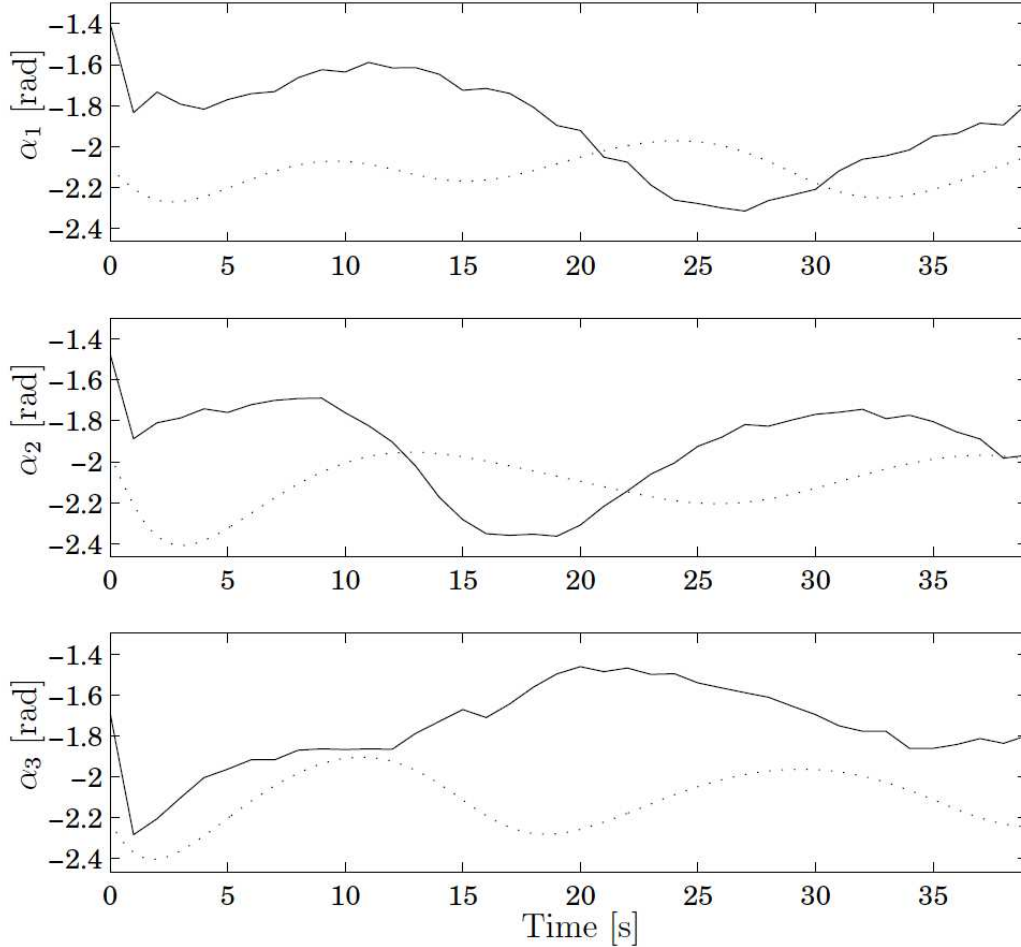


Fig. 13. Kinematic unicycle (cf. Fig. 11a, dotted lines) together with actual rover (cf. Fig. 12, solid lines) heading errors; note how their behaviors are qualitatively consistent in that similar oscillations appear in both cases

in the system due to sensing and information processing not accounted for in [2], the presented results are very encouraging.

However, success is not to say there were not limitations. Firstly, owing to the difficulties in bringing multiple rovers into working order (i.e., free of hardware difficulties), experiments were limited to $n = 4$ rovers. Although it is likely that the reported results extend to $n > 4$ rovers, no experiments were conducted to confirm this. Secondly, the rovers were severely limited by the FOV of their cameras. Even with the inclusion of camera servoing, for certain initial conditions the rovers inevitably lost their target, thus limiting the range of experiments that could be tried. On the other hand, computation of the control law was based solely on sensing and data processing carried out *locally* (i.e., without any explicit communication, nor the use of an overhead camera system or other GPS). Finally, experiments were further restricted by the fact that no method of collision avoidance was employed, a practical issue not considered in this research.

In conclusion, the cyclic pursuit strategy developed for unicycles in [2] was found to be practical from the point-of-view of robustness to unmodelled dynamics, disturbances in the robot velocities, and delays in the system due to sensing and information processing. These experimental findings not only bode well for continuing research on cooperative control strategies based on the notion of pursuit, but also for other cooperative control techniques employing similar local interactions among autonomous agents.

Acknowledgements

The authors gratefully acknowledge financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Centre for Earth and Space Technology (CRESTech).

The authors would like to thank M. A. Mirza and A. M. Rotenstein for helpful discussions regarding the vision systems, and R. Yachoua for logistical help and for taking video. Most of the utilized Linux I/O code was written by L. Ng and the C164 microcontroller was originally programmed by E. J. P. Earon.

References

- [1] J. A. Marshall, M. E. Broucke, B. A. Francis, A pursuit strategy for wheeled-vehicle formations, in: Proceedings of the 42nd IEEE Conference on Decision and Control, Maui, Hawaii, 2003, pp. 2555–2560.
- [2] J. A. Marshall, M. E. Broucke, B. A. Francis, Formations of vehicles in cyclic pursuit, *IEEE Transactions on Automatic Control* 49 (11) (2004) 1963–1974.
- [3] C. W. Reynolds, Flocks, herds, and schools: A distributed behavioural model, *Computer Graphics* 21 (4) (1987) 25–34.
- [4] J. K. Parrish, S. V. Viscido, D. Grünbaum, Self-organized fish schools: An examination of emergent properties, *Biological Bulletin* 202 (2002) 296–305.
- [5] D. Grünbaum, S. Viscido, J. K. Parrish, Extracting interactive control algorithms from group dynamics of schooling fish, in: V. J. Kumar, N. E. Leonard, A. S. Morse (Eds.), *Cooperative Control: A Post-workshop Volume: 2003 Block Island Workshop on Cooperative Control*, Vol. 309 of *Lecture Notes in Control and Information Sciences*, Springer-Verlag, Inc., 2004, pp. 103–117.
- [6] T. D. Barfoot, E. J. P. Earon, G. M. T. D’Eleuterio, Controlling the masses: Control concepts for multi-agent mobile robotics, in: *Second Canadian Space Exploration Workshop*, Calgary, Canada, 1999.

- [7] A. R. Girard, J. B. de Sousa, J. K. Hedrick, An overview of emerging results in networked multi-vehicle systems, in: Proceedings of the 40th IEEE Conference on Decision and Control, Orlando, Florida, 2001, pp. 1485–1490.
- [8] A. Jadbabaie, J. Lin, A. S. Morse, Coordination of groups of mobile autonomous agents using nearest neighbor rules, *IEEE Transactions on Automatic Control* (2003) 988–1001.
- [9] E. W. Justh, P. S. Krishnaprasad, Equilibria and steering laws for planar formations, *Systems & Control Letters* 52 (2004) 25–38.
- [10] Z. Lin, M. E. Broucke, B. A. Francis, Local control strategies for groups of mobile autonomous agents, *IEEE Transactions on Automatic Control* 49 (4) (2004) 622–629.
- [11] W. Ren, R. W. Beard, T. W. McLain, Coordination variables and consensus building in multiple vehicle systems, in: V. J. Kumar, N. E. Leonard, A. S. Morse (Eds.), *Cooperative Control: A Post-workshop Volume: 2003 Block Island Workshop on Cooperative Control*, Vol. 309 of *Lecture Notes in Control and Information Sciences*, Springer-Verlag, Inc., 2004, pp. 171–188.
- [12] H. S. M. Coxeter, *Regular Polytopes*, Methuen & Co. Ltd., London, 1948.
- [13] E. King, Y. Kuwata, M. Alighanbari, L. Bertucelli, J. P. How, Coordination and control experiments on a multi-vehicle testbed, in: Proceedings of the 2004 American Control Conference, Boston, Massachusetts, 2004, pp. 5315–5320.
- [14] Z. Jin, S. Waydo, E. Wildanger, M. Lammers, H. Scholze, P. Foley, D. Held, R. M. Murray, MVWT-II: The second generation Caltech multi-vehicle wireless testbed, in: Proceedings of the 2004 American Control Conference, Boston, Massachusetts, 2004, pp. 5321–5326.
- [15] T. W. McLain, R. W. Beard, Unmanned air vehicle testbed for cooperative control experiments, in: Proceedings of the 2004 American Control Conference, Boston, Massachusetts, 2004, pp. 5327–5331.
- [16] V. Vladimerou, A. Stubbs, J. Rubel, A. Fulford, G. E. Dullerud, A hovercraft testbed for decentrallized and cooperative control, in: Proceedings of the 2004 American Control Conference, Boston, Massachusetts, 2004, pp. 5332–5337.
- [17] M. A. Mirza, D. M. Beach, E. J. P. Earon, G. M. T. D’Eleuterio, Development of a next-generation autonomous robotic network and experimental testbed, in: Proceedings of the Planetary and Terrestrial Mining Sciences Symposium, Sudbury, Canada, 2004.