Proceedings of the 2003 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Las Vegas, Nevada · October 2003

Formation control for multiple mobile robots: a non-linear attractor dynamics approach

Estela Bicho and Sergio Monteiro
Department of Industrial Electronics, University of Minho
4800-058 Guimaraes(Portugal)
estela.bicho@dei.uminho.pt, sergio.monteiro@dei.uminho.pt

***Abstract*— In this paper we focus on modelling formations of non-holonomic mobile robots using non-linear attractor dynamics (see video). The benefit is that the behavior of each robot is generated by time series of asymptotically stable states which therefore contribute to the robustness against environmental perturbations. This study extends our previous work [10]. Here we develop a set of decentralized and distributed basic control architectures that allows each robot to maintain a desired pose within a formation and to enable changes in the shape of the formation which are necessary to avoid obstacles. Simulation results, for teams of four and six mobile robots driving in cluttered and unknown environments, while simultaneously trying to drive in line, column, square, diamond and hexagon are presented. We explain how this approach naturally extends to larger teams of robots.**

I. INTRODUCTION

In this paper we address the fundamental problem underlying the control and coordination of multiple autonomous mobile robots that must drive maintaining a desired geometric formation and simultaneously avoid collisions with obstacles, in an unknown environment. This problem has been receiving much attention from researchers working on cooperative robotics (see e.g [1], [2], [6], [9], [10], [11], [16] and [17] for some interesting works). The motivation is that there are many interesting applications (e.g. box pushing [9], payload transportation [7], capturing/enclosing an invader[16]) that require the robots to coordinate their movements more closely.

In [10] we have proposed a non-linear dynamical systems approach to behavior-based formation control. As a case study we have presented only the example of navigation in triangle formation, integrated with obstacle avoidance, for a team of three autonomous robots. In this study the distance was not controlled and velocity control, which is also important to maintain the configuration, was not explained formally. The flexibility and reconfigurability of our *dynamical systems approach to formation control* remained therefore an open question. Here we extend that seminal work by generalizing the approach to other geometric formations and for larger teams of robots. Concretely, here we develop a set of decentralized and distributed basic control architectures for line, column and "oblique" formation for a team of two robots. These dynamic control architectures can then be easily combined and generate more complex geometric formations for larger teams of robots. As examples we show teams of four and six mobile robots driving in line, column, square, diamond and hexagon. We demonstrate the flexibility of our dynamic control architectures by presenting the ability to avoid sensed obstacles integrated with movement in formation. Although we present examples for formations for teams of four and six robots, more complex general configurations (larger number of robots) can be solved by our approach.

We assume that the robots have no prior knowledge of the environment and we follow a master-referenced strategy for each robot in the team. The control architecture of each robot is structured in terms of elementary behaviors (i.e. obstacle avoidance and "keep formation" behavior). The individual behaviors and their integration are modelled by non-linear dynamical system and bifurcations are used to make design decisions around points at which a system must switch from one type of solution to another. The advantage is that the mathematical properties associated with the concepts (c.f. section III) enable system integration including stability of the overall behavior of the autonomous systems. The dynamical systems that govern the behavior of each robot are tuned so that the movement of each robot in time is generated as a time series of attractor (i.e. asymptotically stable) states. The benefit is that asymptotical stability can be maintained and thus the systems are robust against environmental perturbations.

The rest of the paper is structured as follows: next, in section II we outline the basic assumptions behind this work. In section III we show: i) how control architectures for formation control (line, column and "oblique") for teams of two robots can be modelled by attractor dynamics formulated at the level of heading direction and path velocity; ii) how these control architectures can be combined to generate more complex geometric formations for larger teams of robots (e.g. square, diamond). Simulation results are presented in section IV and demonstrate the ability of the robots to drive in formation and simultaneously avoid obstacles. The paper ends in section V with conclusions and an outlook for future work.

II. The team of robots and their tasks

We briefly discuss the organization of the team of mobile robots and we outline the basic assumptions behind this work.

A team of N robots has one designated *Lead Robot* labelled R_1 (the notion of a *Lead Robot* is in analogy with the work of Desai, Ostrowsky and Kumar[6]). This robot drives from an initial position to a final goal destination which might be defined by a cartesian position in space or signalled by a source emitting a specific signal (e.g. a tower emitting and IR signal or identified by a specific color). Within the formation, each robot (except the *Lead Robot*) depends on one of the others. Thus there are many *leaders* and many *followers* but a unique *Lead Robot*. We decompose the team of N robots into $N-1$ sub-teams of 2-robots each. The control of each sub-team follows a *leader-follower* decentralized motion control strategy (see example in Figure 1). Each *follower* robot

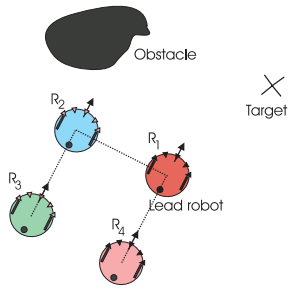


Fig. 1. Example of a team with $N(=4)$ robots in formation. R_1 is the *Lead Robot*, R_2 follows robot R_1. R_2 is the leader for robot R_3. Robot R_4 also follows R_1.

takes its *leader* as a reference point and its motion must be controlled in order to fulfill the following task requirements (see Figure 2): ***i)*** To maintain a desired relative angle between the *leader* and the *follower* (i.e. $\Delta\psi_{\mathrm{i,d}}$); ***ii)*** to maintain a desired distance to the *leader* (i.e. $l_{\mathrm{i,d}}$); and ***iii)*** simultaneously avoid collisions with obstacles that may appear. The simulated robots are based

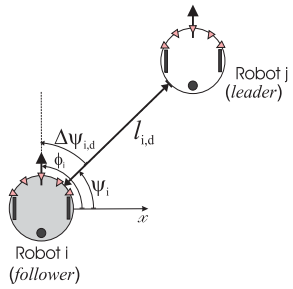


Fig. 2. $\Delta\psi_{\mathrm{i,d}}$ is the desired relative angle between the follower (*Robot*$_{\mathrm{i}}$) and its leader (*Robot*$_{\mathrm{j}}$). $l_{\mathrm{i,d}}$ is the desired distance of the follower to the leader.

on a physical mobile robot where the motion control architecture for the *Lead Robot* has been previously implemented and evaluated [4][5]. (Videos can be found on http://www.dei.uminho.pt/pessoas/estela)

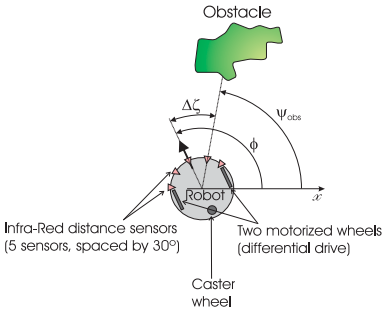


Fig. 3. Each robot has a number of distance sensors mounted on a ring which is centered on the robot's rotation axis. These are used to measure the distance to obstructions at the directions in which they are pointing in space. The simulated sensors mimic the IR sensors mounted on the physical robot, i.e. they have a distance range of 60 cm and an angular range of $\Delta\zeta = 30$ deg.

III. Attractor dynamics for robot formation

In this section we first present how basic and simple control architectures for teams of two robots that generate navigation in formation (e.g. line, column and "oblique") can be built based on the so called *A dynamical systems approach to behavior-based robotics*[13][14][5][4]. With these basic control architectures, more complicated formations (e.g. square, polygon, star, hexagon) can be achieved for larger teams of robots. The basic ideas of the approach are the following: (1) The *Behavioral variables heading direction*, ϕ ($0 \leq \phi \leq 2\pi$ rad), with respect to an arbitrary but fixed world axis, and *path velocity*, v, are used to describe, quantify and internally represent the state of the robot system with respect to elementary behaviors. (2) Behavior is generated by continuously providing values to these variables, which control the robot's wheels. The time course of each of these variables is obtained from (constant) solutions of dynamical systems. The attractor solutions (asymptotically stable states) dominate these solutions by design. In the present systems, the *behavioral dynamics* of heading direction, $\phi_{\mathrm{i}}(t)$, and velocity, $v_{\mathrm{i}}(t)$,(i = leader, follower) are differential equations

$$\dot{\phi}_{\mathrm{i}} = f_{\mathrm{i}}(\phi_{\mathrm{i}}, \text{parameters}) \quad (1)$$

$$\dot{v}_{\mathrm{i}} = g_{\mathrm{i}}(v_{\mathrm{i}}, \text{parameters}). \quad (2)$$

Task constraints define contributions to the vector fields, $f_{\mathrm{i}}(\phi_{\mathrm{i}}, \text{parameters})$ and $g_{\mathrm{i}}(v_{\mathrm{i}}, \text{parameters})$. Each constraint may be modelled either as a repulsive or as an attractive force-let, which are both characterized by three parameters: (a) which value of the behavioral variable is specified? (b) how strongly attractive or repulsive the specified value is?; and (c) over which range of values of the behavioral variable a force-let acts? Thus, in isolation,

each force-let creates an attractor (asymptotically stable state) or a repeller (unstable state) of the dynamics of the behavioral variables. An attractive force-let serves to attract the system to a desired value of the behavioral variable (e.g. the direction in which the *leader* is seen might be a desired value for *follower*'s heading direction, see Figure 4). A repulsive force-let is used to avoid the values of the behavioral variable that must be avoided (for example, the directions in which obstacles lie are values that the heading direction must avoid, see Figure 3).

Next, we built the behavioral dynamics, i.e. we derive the vector fields of Eq. 1 and Eq. 2.

A. *2-robot formations*

The heading direction and path velocity dynamics for the *follower* robot depends on the type of desired formation (line, column or oblique).

1) ***Column formation dynamics****: $Robot_i$* is said to drive in column formation with $Robot_j$ if it drives behind it at a desired distance (see Figure 4). To be in column

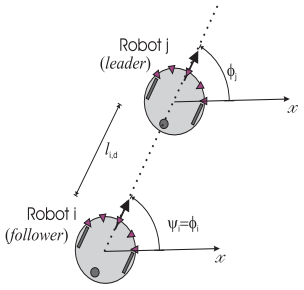


Fig. 4. Two robots in column formation. ψ_i is the direction at which $Robot_i$ sees $Robot_j$ from its current position and with respect to the external reference frame. For column formation ψ_i is the desired value for ϕ_i.

formation, the *follower* ($Robot_i$) must drive behind its *leader* ($Robot_j$), i.e. it must steer to the direction where it sees the *leader* with respect to the external reference frame. This means that the desired relative angle between the leader and the follower is zero ($\Delta\psi_{i,d} = 0$) and thus the desired value for the heading direction, ϕ_i, of $Robot_i$ is ψ_i. A simplest dynamical system for $Robot_i$'s heading direction that generates navigation in column formation taking its *leader* as a reference point is

$$\dot{\phi}_i = f_{col,i} = -\lambda_{col} \sin(\phi_i - \psi_i) \tag{3}$$

which erects an attractor for ϕ_i directly at the direction at which the *leader* lies as seen from the current position of the *follower* (i.e. ψ_i). λ_{col} (> 0) is the strength of attraction to the attractor and corresponds to the relaxation rate (i.e. inverse of the local time scale).

2) ***"Oblique" formation dynamics****:* We say that $Robot_i$ drives in oblique formation with respect to $Robot_j$ when during motion it maintains fixed (equal to a pre-defined angle $\Delta\psi_i$) the direction at which it sees $Robot_j$ (see Figure 2).

A dynamical system for the heading direction of $Robot_i$ that generates "oblique" formation taking $Robot_j$ as a reference point is

$$\begin{aligned} \dot{\phi}_i &= f_{oblique}(\phi_i) \\ &= f_{attract}(\phi_i) + f_{repel}(\phi_i) \end{aligned} \tag{4}$$

where each term defines an attractive force (k = attract, repel)

$$f_k(\phi_i) = -\lambda_{oblique}\lambda_k(l_i) sin(\phi_i - \psi_k) \tag{5}$$

where the first contribution, $f_{attract}$, erects an attractor at a direction

$$\psi_{attract} = \psi_i + \Delta\psi_{i,d} - \pi/4 \tag{6}$$

The strength of this attractor ($\lambda_{oblique}\lambda_{attract}(l_i)$ with $\lambda_{oblique}$ fixed), increases with distance, l_i, between the two robots:

$$\lambda_{attract}(l_i) = 1/(1 + \exp(-(l_i - l_{i,d})/\mu)). \tag{7}$$

The second contribution, f_{repel}, sets an attractor at a direction pointing away from the *leader*,

$$\psi_{repel} = \psi_i + \Delta\psi_{i,d} + \pi/4 \tag{8}$$

with a strength ($\lambda_{oblique}\lambda_{repel}(l_i)$) that decreases with distance, l_i, between the robots,

$$\lambda_{repel}(l_i) = 1 - \lambda_{attract}(l_i). \tag{9}$$

Because these two attractive forces are overlapping only one attractor results from their superposition. The direction at which the resulting attractor emerges depends on the distance between the two robots. This is illustrated in Figure 5.

3) ***Line formation dynamics****:* Two robots are said to be in line formation if they drive side-by-side at a desired distance (see Figure 6).

A behavioral dynamics for the heading direction of $Robot_i$ that generates line formation taking $Robot_j$ as a reference point is exactly given by the same dynamical systems as for "oblique" formation. The difference is that $\Delta\psi_{i,d} = \pm\pi/2$ in Eqs. 6 and 8.

4) ***Integration with obstacle avoidance****:* An obstacle avoidance dynamics formulated at the level of heading direction has been previously elaborated and implemented on a vehicle platform on which the simulated robots here are inspired[3][4][5]:

$$\dot{\phi}_i = \sum_s f_{obs,s}(\phi_i) \tag{10}$$

where $f_{obs,s}$ are repulsive "force-lets", defined around each direction in which obstructions (either due to static obstacles or due to the other robots) are sensed. These are characterized by (a) the direction, ψ_s, to be avoided, (b) the strength, $\lambda_{obs,s}$, of repulsion, and (c) the range, σ_s

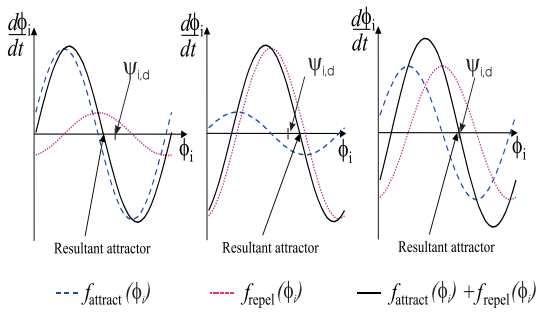


Fig. 5. This figure shows the two contributions to the "oblique" formation dynamics and their superposition for the three different physical situations. Left: When the distance between the two robots is larger than the desired distance the attractive force erected at direction ψ_{attract} is stronger than the attractive set at direction ψ_{repel}. Their superposition leads to an attractor at a direction still pointing towards the movement direction of the leader robot. Middle: Conversely, when the distance between the two robots is smaller than the desired distance, the reverse holds, i.e. the attractive force set at direction ψ_{attract} is now stronger than the attractive force at direction ψ_{repel}. The resulting oblique formation dynamics exhibits an attractor at a direction pointing away from leader's direction of movement. Right: When the robots are at the desired distance the two attractive forces have the same strength which leads to a resultant attractor at the direction $\psi_{i,d} = \psi_i + \Delta\psi_{i,d}$.

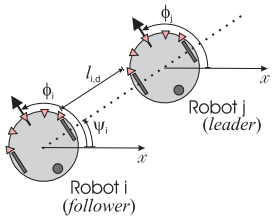


Fig. 6. Two robots in a line formation. $Robot_j$ is the *leader* for $Robot_i$ which must drive such that it sees its *leader* perpendicularly (i.e. $\psi_{i,d} = \psi_i + \pi/2$ if $Robot_i$ is to the left or $\psi_{i,d} = \psi_i - \pi/2$ if $Robot_i$ is to the right) and simultaneously keep a desired distance, $l_{i,d}$, between them.

over which repulsion acts. These repulsive force-lets can be straightforwardly erected by the distance sensors:

$$f_{\text{obs,s}}(\phi_i) = \lambda_{\text{obs,s}}(\phi_i - \psi_s)\exp[-\frac{(\phi_i - \psi_s)^2}{2\sigma_s^2}] \qquad (11)$$

where $\psi_s = \zeta_s + \phi_i$ is the direction in space into which an IR sensor, mounted at angle ζ_s from the frontal direction, is pointing. The strength of repulsion, $\lambda_{\text{obs,s}}$, is a decreasing function of sensed distance, d_s, to the obstruction, as estimated from the IR output with crude calibration. The functional form

$$\lambda_{\text{obs,s}} = \beta_1 \exp[-d_s/\beta_2] \qquad (12)$$

depends on two parameters controlling overall strength (β_1) and spatial rate of decay (β_2).

The range

$$\sigma_s = \arctan\left[\tan(\frac{\Delta\zeta}{2}) + \frac{R_{\text{robot}}}{R_{\text{robot}} + d_s}\right] \qquad (13)$$

is adjusted taking both sensor sector, $\Delta\zeta$, and the minimal passing distance of the robot (at size R_{robot} of the platform) into account.

Note that the right hand side of Eq. 11 really only depends on the distance measures, d_s, obtained from the sensors, not actually on ϕ_i (to see this, replace $\phi_i - \psi_s$ by ζ_s, which is fixed).

Finally, because we have formulated all the behavioral dynamics at the level of heading direction the contributions that generate the basic formations and the contributions arising from the detected obstacles can be integrated, adding the corresponding contributions to the vector field:

$$\dot{\phi}_i = \sum_s f_{\text{obs,s}}(\phi_i) + \gamma_{\text{line,i}} f_{\text{line,i}}(\phi_i) + \qquad (14)$$
$$\gamma_{\text{col,i}} f_{\text{col,i}}(\phi_i) + \gamma_{\text{oblique,i}} f_{\text{oblique,i}}(\phi_i) + f_{\text{stoch}}$$

where $\gamma_{\text{line,i}}$, $\gamma_{\text{col,i}}$ and $\gamma_{\text{oblique,i}}$ are mutually exclusive boolean variables that determines which configuration is desired for the formation. The vector field is augmented with a stochastic contribution, f_{stoch} that guarantees escape from repellers (which might happen when a bifurcation in the vector field takes place) and simultaneously models perturbations.

*5) **Path velocity control**:* In the previous subsection we modelled the changes in the heading direction of a *follower* robot. Here we focus on the path velocity. In any case, the *follower*'s path velocity must be controlled so that this robot can maintain the desired distance to its *leader*. Additionally, velocity control must be constrained by sensed obstructions, either due to static obstacles or due to the other robots that might come to close and thus collisions with them must be avoided. This can be accomplished by means of a dynamic system for the path velocity:

$$\dot{v}_i = \gamma_{\text{obs,i,i}} g_{\text{obs,i}}(v_i) + \gamma_{\text{line,i}} g_{\text{line,i}}(v_i) + \qquad (15)$$
$$\gamma_{\text{col,i}} g_{\text{col,i}}(v_i) + \gamma_{\text{oblique,i}} g_{\text{oblique,i}}(v_i)$$

where each contribution defines simply a linear dynamical system for the path velocity ($f = \{\text{obs}, \text{line}, \text{col}, \text{oblique}\}$):

$$\dot{v}_i = g_f(v_i) = -\alpha_f(v_i - v_{i,d,f}) \qquad (16)$$

that sets an attractor at the desired path velocity, $v_{i,d,f}$, with a relaxation rate controlled by α_f (> 0).

The desired path velocity, $v_{i,d,f}$, is a function of whether or not obstacles are sensed and by the requirement to keep a desired distance to the *follower* robot.

When the *followers*'s heading direction is inside the repulsion range created by sensed obstructions then the obstacle avoidance term dominates (i.e. $\gamma_{\text{obs,i}} = 1$, $\gamma_{\text{line,i}} = 0$, $\gamma_{\text{col,i}} = 0$ and $\gamma_{\text{oblique,i}} = 0$) and in this case the desired path value for the path velocity is:

$$v_{i,d,\text{obs}} = d_{\min}/T_{2c,\text{obs}} \qquad (17)$$

which tries to stabilize a particular time to contact, $T_{2c,\text{obs}}$, with the obstacle. $d_{\min}$ is the minimum distance given by the distance sensors. Reversely, when no obstructions

are sensed or the robot's heading direction is outside the repulsive effect of obstacle contributions then the particular desired value for the velocity depends on the desired configuration for the formation. For column and oblique formation the desired value for the path velocity is

$$v_{\mathrm{i,d,col/oblique}} = \begin{cases} v_{\mathrm{j}} - (l_{\mathrm{i,d}} - l_{\mathrm{i}})/T_{2\mathrm{c}} & if\, l_{\mathrm{i}} \geq l_{\mathrm{i,d}} \\ -v_{\mathrm{j}} - (l_{\mathrm{i,d}} - l_{\mathrm{i}})/T_{2\mathrm{c}} & else \end{cases} \quad (18)$$

Which makes the *follower* to accelerate or decelerated depending on the *leader*'s path velocity, v_{j}, and on the requirement to maintain the distance $l_{\mathrm{i,d}}$ to the *leader*. The parameter $T_{2\mathrm{c}}$ permits also to control accelerations and decelerations such that the robot's movement is smooth.

While in line formation, to ensure that the *follower* robot is able to reach and maintain the desired distance to its *leader* one needs to take into account besides the *leader*'s path velocity, v_{j}, also the *leader*'s current heading direction (ϕ_{j}) and the direction at which it sees the *leader* from its current position, ψ_{i}. A set of rules can be defined that constrains the *follower*'s desired path velocity during line formation:

$$\begin{aligned} v_{\mathrm{i,d,line}} &= DE_1 \cdot v_{\mathrm{j}}(1 - |\sin(\psi_i)|) + \\ &+ DE_2 \cdot v_{\mathrm{j}}(1 - |\cos(\psi_i)|) + \\ &+ AC_1 \cdot v_{\mathrm{j}}(1 + K_v|\sin(\psi_i)|) + \\ &+ AC_2 \cdot v_{\mathrm{j}}(1 + K_v|\cos(\psi_i)|) \end{aligned} \quad (19)$$

These rules depend on the activation, of mutually exclusive boolean variables DE_1, DE_2, AC_1 and AC_2, dictated by the direction at which the *follower* sees the *leader* and the heading direction of the *leader*. DE_1 and DE_2 are logical variables that indicate that the *follower* robot must decelerate. Reversely, AC_1 and AC_2 are logical variables that indicate that the *follower* robot must accelerate. In Figure 7 is shown these boolean variables and how their activation is done. The activation of these variables are

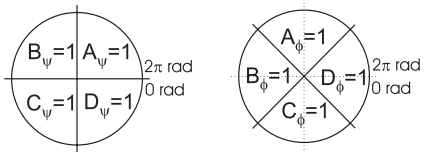


Fig. 7. Two trigonometric circles representing the activation of logical variables. Left: results from testing the direction at which the follower sees the *leader* ψ_{i}. If $0 \leq \psi_{\mathrm{i}} < \pi/2$ then $A_\psi = 1$ and $B_\psi = C_\psi = D_\psi = 0$. If $\pi/2 \leq \psi_{\mathrm{i}} < \pi$ then $B_\psi = 1$ and $A_\psi = C_\psi = D_\psi = 0$, and so on. Right: results from testing the heading direction of the leader (ϕ_{j})

determined by the following boolean functions:

$$DE_1 = A_\psi B_\phi + B_\psi C_\phi + A_\phi\left(C_\psi + D_\psi\right) \quad (20)$$

$$DE_2 = A_\psi C_\phi + D_\psi B_\phi + D_\phi\left(B_\psi + C_\psi\right) \quad (21)$$

$$AC_1 = A_\psi\left(A_\phi + D_\phi\right) + C_\psi\left(B_\phi + C_\phi\right) \quad (22)$$

$$AC_2 = B_\psi\left(A_\phi + B_\phi\right) + D_\psi\left(C_\phi + D_\phi\right) \quad (23)$$

Specifically, DE_1 and DE_2 are activated when the *follower* is ahead of its *leader*. In this case the *follower*'s desired velocity, $v_{\mathrm{i,d,line}}$, must decrease until the *leader*'s velocity, v_{j}, is reached. The other two variables AC_1 and AC_2 are activated when the *follower*'s desired velocity must increase, i.e. when it drives behind the leader.

B. *N-robot formations*

In the previous subsections we have addressed solutions to the problem of generating two-robot formations. Here we show how those solutions easily extend to teams of N-robots and present examples of maintaining particular geometric configurations.

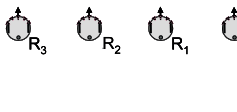
We assume that a higher level controller defines the shape of the formation trough a matrix **S**:

$$\mathbf{S} = \begin{pmatrix} L_1 & \Delta\psi_{1,\mathrm{d}} & l_{1,\mathrm{d}} \\ L_2 & \Delta\psi_{2,\mathrm{d}} & l_{2,\mathrm{d}} \\ \ldots & \ldots & \ldots \\ L_{\mathrm{N}} & \Delta\psi_{\mathrm{N,d}} & l_{\mathrm{N,d}} \end{pmatrix} \quad (24)$$

This matrix codes the shape of the formation in the following way: each row defines the pose of a particular robot in the formation. Row i ($= 1,2,3,...,N$) in **S** is a vector $\mathbf{S}_{\mathrm{i}} = \begin{pmatrix} L_{\mathrm{i}} & \Delta\psi_{\mathrm{i,d}} & l_{\mathrm{i,d}} \end{pmatrix}$, where L_{i} ($L_{\mathrm{i}} \neq R_{\mathrm{i}}$) identifies the *leader* robot for Robot i, $\Delta\psi_{\mathrm{i,d}}$ (rad) is the desired relative angle between Robot i and its *leader* and $l_{\mathrm{i,d}}$ (cm) the desired distance to its *leader*.

When Robot i is the *Lead Robot* the parameters for its dynamics are $L_{\mathrm{i}} = 0$, $\Delta\psi_{\mathrm{i,d}} = 0$ and $l_{\mathrm{i,d}}$ defines the distance at which it must stop from the target position.

In Figures 8 and 9 we define the matrixes S that determine the shapes of two diferent formations. In both cases we assume that Robot R_1 is the *lead robot* (i.e. moves toward the target location), and that the desired distance between the robots is 100 cm.

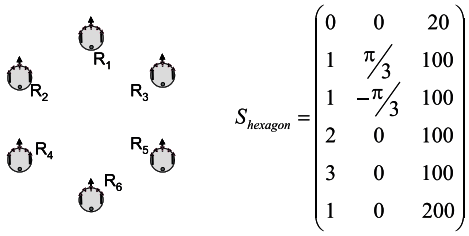


$$S_{line} = \begin{pmatrix} 0 & 0 & 20 \\ 1 & \pi/2 & 100 \\ 2 & \pi/2 & 100 \\ 3 & -\pi/2 & 100 \end{pmatrix}$$

Fig. 8. Example of a line formation. Robot R_1 is the *lead robot*, Robot R_2 follows R_1 keeping it on its right side, Robot R_3 follows R_2 keeping it on its right side and Robot R_4 follows Robot R_1 keeping it on its left side

IV. Simulation results

The complete dynamic architectures were evaluated in computer simulations. These were generated by a software simulator written in MATLAB. We modelled the robotic platforms, based on the physical prototype in which the dynamic control architecture for the *Lead Robot* has been previously implemented. In the simulation the robots are represented as triplets (x_i, y_i, ϕ_i) (i= 1,2...,N), consisting



$$S_{hexagon} = \begin{pmatrix} 0 & 0 & 20 \\ 1 & \pi/3 & 100 \\ 1 & -\pi/3 & 100 \\ 2 & 0 & 100 \\ 3 & 0 & 100 \\ 1 & 0 & 200 \end{pmatrix}$$

Fig. 9. Example of hexagon formation. Robot R_1 is the *Lead Robot*, Robot R_2 follows R_1 on the left side and maintaining an oblique formation, Robot R_3 follows R_1 on the right side and maintaining an oblique formation. Robot R_4 follow Robot R_2 in a column formation. Robots R_5 follow Robot R_3 maintaining a column formation. Robot R_6 follows R_1 in column formation.

of the corresponding two Cartesian coordinates and the heading direction. Cartesian coordinates are updated by a dead-reckoning rule ($\dot{x}_i = v_i cos(\phi_i)$, $\dot{y}_i = v_j sin(\phi_i)$) while heading direction, ϕ_i, and path velocity, v_i, are obtained from the corresponding behavioral dynamics. All dynamical equations are integrated with a forward Euler method with fixed time step, and sensory information is computed once per each cycle. Distance sensors are simulated through an algorithm reminiscent of ray-tracing. The target information is defined by a goal position in space. It is assumed here that the *leader* robots broadcasts their current velocity and heading direction to their *followers*.

Figures 10 to 15 show simulation runs of the complete system which demonstrates the smooth behavior consistent with all imposed constraints. Figure 10 depicts the trajectories generated by the complete behavioral dynamics when the robots start at random initial positions and must then create and maintain a desired geometric formation: line, column, square and diamond. The environment is free of static obstacles, however the robots must avoid collisions among themselves. Figure 11 shows snapshots of another simulation run where the robots must drive keeping a line formation and simultaneously avoid collisions with obstacles. The heading direction dynamics for each robot, when they are at the positions depicted in snapshots B, C and D, are presented in Figure 12. Figure 13 presents the robots driving in a diamond formation while avoiding obstacles and driving through narrow passages. Figure 14 shows the initial and final snapshots of a team of six robots reaching a hexagon formation. A final simulation run is presented in Figure 15. Here the robots navigate in a very cluttered environment and try to drive always in square formation.

More results, either simulation snapshots or videos, can be found in httt://www.dei.uminho.pt/pessoas/estela. They are not presented here due to space limitations.

V. Conclusion and Future Work

We have demonstrated how non-linear attractor dynamics can be used to design a behavior-based dynamic

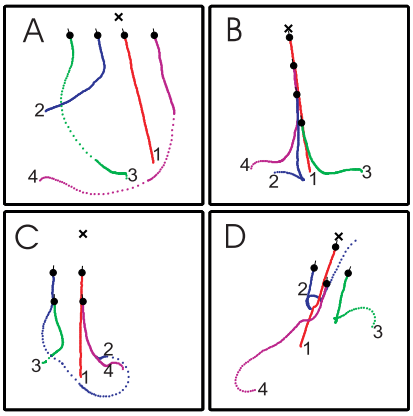


Fig. 10. Examples of simulation runs in the absence of static obstacles. Panel A: Line formation. Panel B: Column formation. Panel C: Square formation. Panel D: Diamond formation.

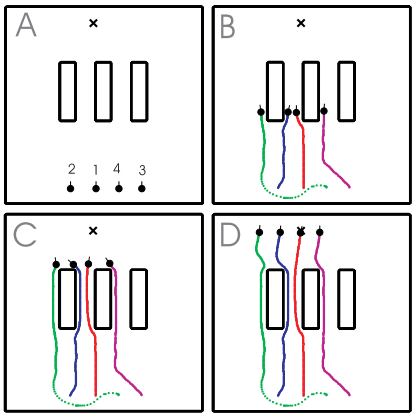


Fig. 11. Snapshots of a simulation run where the robots must drive keeping a line formation (as defined by the matriz on Figure 8) and simultaneously avoid collisions with the obstacles.

control architectures that enables a team of N-robots to drive in a geometric formation and simultaneously avoid obstacles. In particular, we have shown teams of four and six mobile robots driving in line, column, square and diamond and hexagon. The robots have no prior-knowledge of their environment. Simulation studies have shown that the generated trajectories are smooth and collisions, with static obstacles and among the robots, are avoided. Flexibility is achieved in that as the sensed world and/or the information shared among the robots change, the systems may change their planning solutions continuously but also discontinuously (tuning the formations versus split to avoid obstacles).

The work described here imposes of course further research. First, the complete architectures must be implemented and their performance evaluated in a team of physical mobile robots. Currently, we are initiating to implement these on a team of physical robots. The implementation on different robot platforms will also permit to infer how easy it is to transfer our control

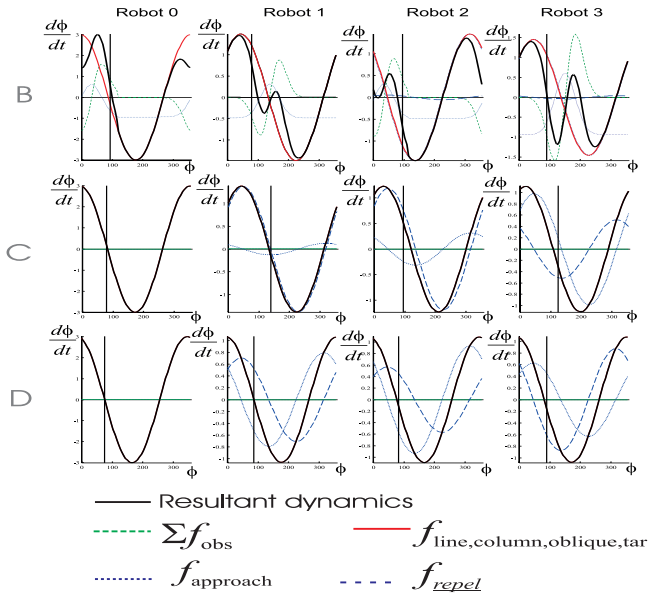


Fig. 12. Heading direction dynamics for the four robots when they are at points depicted in snapshots B, C and D in Figure 11. In each plot the intersection of the vertical line with the horizontal axis gives the current value of the heading direction of the correspondent robot. As we can see the heading direction of each robot is always in or very near an attractor (i.e. a zero with a negative slope) of the resultant dynamics as it moves. As each robot moves, the heading direction dynamics change and thus the resultant attractor shifts pulling the heading direction along.

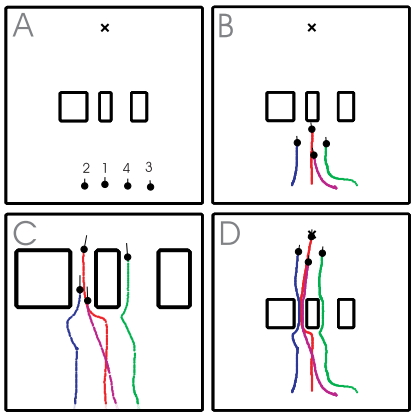


Fig. 13. A simulation run where the robots try to drive always in a diamond formation. In Panel C the region of the obstacles and the narrow passages was amplified to show in more detail the robots avoiding colisions, with the static obstacles and among themselves, while crossing the narrow passage.

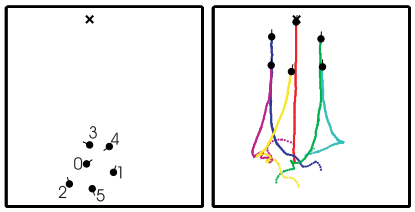


Fig. 14. Initial and final snapshots of a simulation run with six robots acquiring a hexagon formation. $Robot_0$ is the team leader..

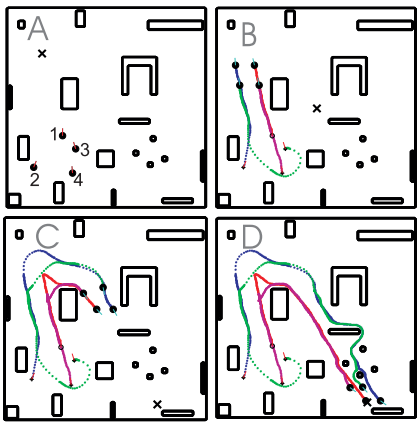


Fig. 15. Snapshots of a simulation run where the robots drive always in square formation. The robots avoid collisions with the obstacles while simultaneously trying to maintain the square formation.

architecture from one type of robots to another.

ACKNOWLEDGEMENTS

This project was supported, in part, through grants SFRH/BD/3257/2000 and POSI/SRI/38051/2001 to E.B. from the portuguese Foundation for Science and Technology (FCT). We are grateful to Rui Soares for his help.

VI. REFERENCES

[1] T Balch and R C Arkin. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation, Vol. 14, No.6, 926–939, December, 1998.*

[2] T Balch and M Hybinette. Social Potentials for scalable Multi-robot Formations. *in Proc. IEEE Int. Conf. Robotics and Automation, San Francisco, 2000.*

[3] E Bicho and G Schöner. The dynamic approach to autonomous robotics demonstrated on a low-level vehicle platform. *Robotics and Autonomous Systems*, 21:23–35, 1997.

[4] E Bicho. Dynamic Approach to Behavior-Based Robotics: design, specification, analysis, simulation and implementation. Shaker Verlag, ISBN 3-8265-7462-1, Aachen, 2000.

[5] E Bicho, P Mallet, and G Schöner. Target representation on an autonomous vehicle with low-level sensors. *The International Journal of Robotics Research, Vol. 19, No.5, May 2000, pp.424-447.*

[6] J Desai, J Ostrowski and V Kumar. Modeling and Control of Formations of Nonholonomic Mobile Robots. *in IEEE Transactions on Robotics and Automation, Vol.17, No. 6, pp.905–908, December 2001.*

[7] P Johnson and J Bay . Distributed control of simulated autonomous mobile robot collectives in payload transportation. *Autonomous Robots*, 2(1):43–64, 1995.

[8] E W Large, H I Christensen, and R Bajcy. Scaling the dynamic approach to path planning and control: Competition amoung behavioral constraints. *International Journal of Robotics Research*, 18(1):37–58, 1999.

[9] M A Lewis and K Tan. High precision formation control of mobile robots using virtual structures. *Autonomous Robots*, 4:387–403, 1997.

[10] S Monteiro and E Bicho. A Dynamical Systems Approach to Behavior-based Formation Control. *in Proc. IEEE Int. Conf. Robotics and Automation, 2606–2611, 2002.*

[11] A Paulino and H Araújo. Control Aspects of Maintaining Non-Holonomic Robots in Geometric Formation. *SIRS2001–Proceedings of the 9th International Symposium on Intelligent Robotic Systems*, pp. 18-20 July, 2001, Toulouse, France.

[12] R Soares and E Bicho. Using attractor dynamics to generate decentralized motion control of two mobile robots transporting a long object in coordination. *in Proc. of the workshop on Cooperative Robotics, in IROS 2002: 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, September 30-October 4, 2002, EPFL Lausanne, Switzerland.*

[13] G Schöner and M Dose. A dynamical systems approach to task-level system integration used to plan and control autonomous vehicle motion. *Robotics and Autonomous Systems*, 10:253–267, 1992.

[14] G Schöner, M Dose, and C Engels. Dynamics of behavior: Theory and applications for autonomous robot architectures. *Robotics and Autonomous Systems*, 16:213–245, 1995.

[15] J Sweeney, T J Brunette, Y Yang and R Grupten. coordinated Teams of Reactive Mobile Platforms. *in Proc. IEEE Int. Conf. Robotics and Automation, 299–304, 2002.*

[16] H Yamaguchi. A cooperative hunting behavior by mobile-robot troops. *The International Journal of Robotics Research*, 18(8):931–940, September 1999.

[17] P K C Wang. Navigation Strategies for multiple autonomous robots moving in formation. *Robotics and Autonomous Systems*, 16:213–245, 1995.