Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

Attractor dynamics generates robot formations: from theory to implementation

Sergio Monteiro†, Miguel Vaz‡ and Estela Bicho†
†Dept of Industrial Electronics and ‡Dept of Mathematics for Science and Technology
University of Minho
Campus de Azurem
4800-058 Guimaraes (PORTUGAL)
sergio.monteiro@dei.uminho.pt, miguel@mct.uminho.pt and estela.bicho@dei.uminho.pt

***Abstract*— We show how non-linear attractor dynamics can be used to implement robot formations in unknown environments. The desired formation geometry is given through a matrix where the parameters in each line (its leader, desired distance and relative orientation to the leader) define the desired pose of a robot in the formation. The parameter values are then used to shape the vector fields of the dynamical systems that generate values for the control variables (i.e. heading direction and path velocity). Then these dynamical systems are tuned such that the control variables are always very close to one of the resultant attractors. The advantage is that the systems are more robust against perturbations because the behavior is generated as a time series of asymptotically stable states.**

Experimental results (with three Khepera robots) demonstrate the ability of the team to create and stabilize the formation, as well as avoiding obstacles. Flexibility is achieved in that as the senses world changes, the systems may change their planning solutions continuously but also discontinuously (tunning the formation versus split to avoid obstacle). [1].

I. Introduction

The problem of controlling a team of autonomous mobile robots that should navigate in a prescribed geometric formation is of growing interest in the robotics research community. Some of the many tasks that would benefit from the solution to this problem are, for instance: payload transportation [1], capturing/enclosing invaders [2], satellite cluster formation [3], spacecraft formation [4] and environment exploration/reconnaissance [5].

There are many, and diverse, approaches to solve these problems. Some of the most relevant reported results include the use of virtual structures [6] [7], vision based approaches [8] [9], leader-follower methods [10] and graph theory [11].

In this paper we continue previous work reported on [12]. There we presented a possible solution to the problem of multi robot formation control, by proposing a decentralized and distributed control architecture completely formalized as a non linear dynamical system that allows each robot to maintain a desired pose within the formation, and also enables the robots to change the shape of the formation in order to avoid obstacles. In that paper, only simulation results were presented, thus lacking the necessary confirmation on real world applications. Here we discuss the first implementations on real robot platforms and present and discuss some results from experiments with a team of three Khepera robots.

Perhaps the most closely related work to ours is the one reported in [13]. In their formulation, a team of robots has one designated *lead robot*, which all other robots follow directly or indirectly (following another team mate). We also use this type of team organization. They also develop two types of controllers (feedback controllers), that control either the position and orientation of the robot to a leader, or the position relative to two robots. We, instead use three types of specialized controllers (using the *attractor dynamics approach*) [12], which can be seen as a form of position and orientation control. In terms of team structure they use the concepts of transition matrix and control graphs, explicitly switching formations in the presence of obstacles. Our formations are flexible in the presence of obstacles, i.e., the formation will adapt itself and maneuver between the obstacles without explicitly switch formations. We also take the advantages of the *attractor dynamics approach* in terms of suitability for use with platforms with low-level sensors and low computational resources [14].

Particular to our work, we use non-linear dynamical systems theory to design and implement these controllers. Specifically, the time course of the control variables are obtained from (constant) solutions of dynamical systems. The attractor solutions (asymptotically stable states) dominate these solutions by design. The benefit is that overt behavior of each robot is generated as a time course of asymptotically stable states, that, therefore, contribute to the overall stability of the complete control system and makes it robust against perturbations.

The rest of the paper is structured as follows: in section II we present our framework for teams of two robots navigating in column, oblique and line formations; after we generalize to any team with N robots maintaining a geometric configuration; in section III we discuss the implementation on real robots and present results of some

[1]Project financed by the Portuguese Foundation for Science and Technology(POSI/SRI/38051/2001)

experiments conducted with Kheperas; we end the paper with conclusions and an outlook on future work.

II. Building Formations

As said before we use the *Dynamical systems approach to behavior-based robotics* [15] [16] [17] [14] [18] to build our robot formations. Here we will briefly describe how this approach can be used for such purpose. For a more detailed explanation please see [12].

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, $v_{\mathrm{i}}(t)$,($\mathrm{i} = \mathrm{leader}, \mathrm{follower}$) are differential equations

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

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

Task constraints define contributions to the vector fields, $f_{\mathrm{i}}(\phi_{\mathrm{i}}, \mathrm{parameters})$ and $g_{\mathrm{i}}(v_{\mathrm{i}}, \mathrm{parameters})$. Each constraint may be modeled 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. A repulsive force-let is used to avoid the values of the behavioral variable that must be avoided.

Now, consider two robots that navigate in a world, keeping constant the distance between them. Then, we state that they are either in a *column* formation, if one is exactly behind the other (see figure 1.a)), or in a *line* formation, if they navigate side-by-side (see figure 1.c)), or in an *oblique* formation, otherwise (see figure 1.b)).

From this set of basic two robot formations, more complex ones can be derived, as we will see later in section II-D. Next, in sections II-A to II-C we present the control architecture for each of these two robot formations.

A. Two robots in column

A dynamical system that causes a follower robot to navigate in column formation, maintaining a constant distance, with its leader is:

$$\dot{\phi}_{\mathrm{i}} = f_{\mathrm{col,i}} = -\lambda_{\mathrm{col}} \sin\left(\phi_{\mathrm{i}} - \psi_{\mathrm{i}}\right) \tag{3}$$

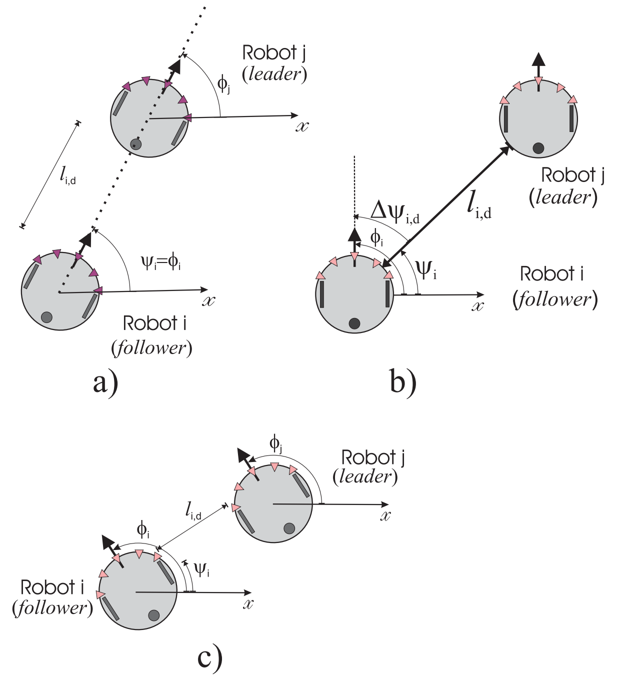


Fig. 1. Possible formation for teams with only two robots. They can be either in a) column formation; b) oblique formation; c) line formation. The heading direction of the leader and the follower are, respectively, ϕ_i and ϕ_j. ψ_i is the direction at which the follower sees the leader. $l_{i,d}$ is the desired distance between both robots. $\Delta\psi_{i,d}$ is the desired difference between the followers heading and the direction at which sees the leader.

This dynamical system ensures that the robot steers to the desired heading direction, ψ_i (the direction at which the follower sees its leader), by making it an asymptotically stable state of the system. Parameter λ_{col} (> 0) is the strength of attraction to the attractor and corresponds to the relaxation rate.

Path velocity is controlled to ensure that the follower adequate its velocity to the leader's one, while trying to maintain the desired distance to it. This is accomplished by making the value of the desired velocity equal to

$$v_{\mathrm{i,d}} = \begin{cases} v_{\mathrm{j}} - (l_{\mathrm{i,d}} - l_{\mathrm{i}})/T_{2\mathrm{c}} & if l_{\mathrm{i}} \geq l_{\mathrm{i,d}} \\ -v_{\mathrm{j}} - (l_{\mathrm{i,d}} - l_{\mathrm{i}})/T_{2\mathrm{c}} & else \end{cases} \tag{4}$$

T_{2C} is a parameter that smooths the robot movement, by controlling its accelerations and decelerations.

B. Two robots in oblique

A dynamical system that causes a follower robot to navigate in an oblique formation, maintaining a constant distance and relative orientation, with its leader is:

$$\begin{aligned} \dot{\phi}_{\mathrm{i}} &= f_{\mathrm{oblique}}(\phi_{\mathrm{i}}) \\ &= f_{\mathrm{attract}}(\phi_{\mathrm{i}}) + f_{\mathrm{repel}}(\phi_{\mathrm{i}}) \end{aligned} \tag{5}$$

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

$$f_{\mathrm{k}}(\phi_{\mathrm{i}}) = -\lambda_{\mathrm{oblique}} \lambda_{\mathrm{k}}(l_{\mathrm{i}}) sin(\phi_{\mathrm{i}} - \psi_{\mathrm{k}}) \tag{6}$$

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

$$\psi_{\mathrm{attract}} = \psi_{\mathrm{i}} + \Delta\psi_{\mathrm{i,d}} - \pi/4 \tag{7}$$

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

$$\lambda_{\mathrm{attract}}(l_{\mathrm{i}}) = 1/(1 + \exp(-(l_{\mathrm{i}} - l_{\mathrm{i,d}})/\mu)). \tag{8}$$

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

$$\psi_{\mathrm{repel}} = \psi_{\mathrm{i}} + \Delta\psi_{\mathrm{i,d}} + \pi/4 \tag{9}$$

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

$$\lambda_{\mathrm{repel}}(l_{\mathrm{i}}) = 1 - \lambda_{\mathrm{attract}}(l_{\mathrm{i}}). \tag{10}$$

The attractor location of the resultant vector field, is thus dependent on the distance between the two robots. 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. 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 the leader's direction of movement. 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 $\psi_{\mathrm{i,d}} = \psi_{\mathrm{i}} + \Delta\psi_{\mathrm{i,d}}$.

Path velocity is controlled exactly in the same way as for column formation.

C. Two robots in Line

A dynamical system that causes a follower robot to navigate in a line formation, maintaining a constant distance, with its leader is similar to the one of oblique formation. The only difference lies in $\Delta\psi_{i,d}$, which is fixed and equal $\pm\pi/2$ depending on the follower driving on the right or left of the leader.

In line formation, the path velocity does not depend only on the distance and velocity of the leader, but we also have to take into account the heading direction of the leader and the direction at which it is seen by the follower. A set of heuristic rules have been written that make the follower accelerate or decelerate depending on the leader's pose relative to the follower:

$$\begin{aligned} v_{\mathrm{i,d,line}} &= DE_1 \cdot v_{\mathrm{j}}(1 - |\sin(\psi_i)|) + \\ &+ DE_2 \cdot v_{\mathrm{j}}(1 - |\cos(\psi_i)|) + \\ &+ AC_1 \cdot v_{\mathrm{j}}(1 + K_v\,|\sin(\psi_i)|) + \\ &+ AC_2 \cdot v_{\mathrm{j}}(1 + K_v\,|\cos(\psi_i)|) \end{aligned} \tag{11}$$

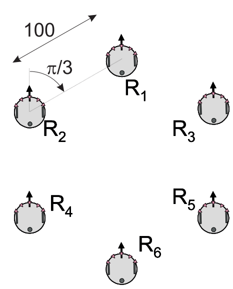
where DE_1, DE_2, AC_1 and AC_2 are mutually exclusive activation variables that embed the relative attitude of the leader regarding the follower. They are set and reset by testing the direction at which the leader is seen by the follower and the heading direction of the leader (see [12] for details).

D. N-Robot formations

Teams of robots with more than two robots are built by specifying pairs of leader-follower teams and stating the particular configuration to achieve. A complete team specification is accomplished by means of a *formation matrix* as shown in equation 12.

$$\mathbf{S} = \begin{pmatrix} L_1 & \Delta\psi_{1,\mathrm{d}} & l_{1,\mathrm{d}} \\ L_2 & \Delta\psi_{2,\mathrm{d}} & l_{2,\mathrm{d}} \\ \ldots & \ldots & \ldots \\ L_{\mathrm{N}} & \Delta\psi_{\mathrm{N,d}} & l_{\mathrm{N,d}} \end{pmatrix} \tag{12}$$

For a team of N robots, where each robot is identified by a specific identification number, the formation matrix has N rows and three columns. Row *i* relates to the robot with identification number *i*. The contents of the columns specify the values that characterize a formation, $\Delta\psi i, d$ and $l_{i,d}$ in columns two and three, respectively, and the identification number of this robots leader, in column one. The team leader is identified by having its row with $l_{i,d} = 0$ and $\Delta\psi i, d = 0$, while the third column is the distance it should stop from the target.



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

Fig. 2. 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.

III. Implementation Results

This control architecture has been implemented in a team of three Khepera robots. These are small sized robots (about 6cm diameter) equipped with six infrared distance sensors (from 2 to 5.5 cm range) and have as processing unit a Motorola 68000. In these experiments one external PC was used to centralize the information regarding the formation. Its purpose was to allow a user to input the desired geometric formation, construct the corresponding formation matrix, and then communicate to each robot its desired pose within the formation. When the starting order is given, the team leader starts, then, to broadcast to its followers its actual position, heading and velocity. Due to radio communication problems, in these experiments we were restricted to have the same leader to all the robots, which is the team leader.

Since the robots only have distance to obstacle sensors, they can't detect their team-mates and have to rely on communicated information, thus using a global coordinate system. Cartesian coordinates are updated, every computation cycle, by a dead-reckoning rule ($\dot{x}_i = v_i cos(\phi_i)$, $\dot{y}_i = v_i sin(\phi_i)$) 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 time step equal to the actual computation time. Sensory information and leader's position are updated once per each cycle. The target information is defined by a goal position in space (i.e. $(x_{\text{tar}}, y_{\text{tar}})$) using the global coordinate system.

Computation time per each cycle is greatly dependent on the desired formation that the robot is performing. Thus, if one robot is performing a column formation, in these robots, its computation time is typically between 40ms and 50ms per cycle. This time increases to values between 65ms to 85ms in the cases of either oblique or line formation. This means that, in principle, when doing column formation the observed results, in terms of dead reckoning, should be more precise than for the other two formation behaviors. The parameters are tuned such that the relaxation rates are adapted automatically as a function of the computation cycle.

In the next figures we show the results of two conducted experiments. More specifically figures 3,4 and 5 report one experiment where the Kheperas start in a column formation and then switch to a triangle formation, in an obstacle free environment. In figures 6, 7 and 8 the robots should maintain the line formation, but one of them has to avoid an obstacle.

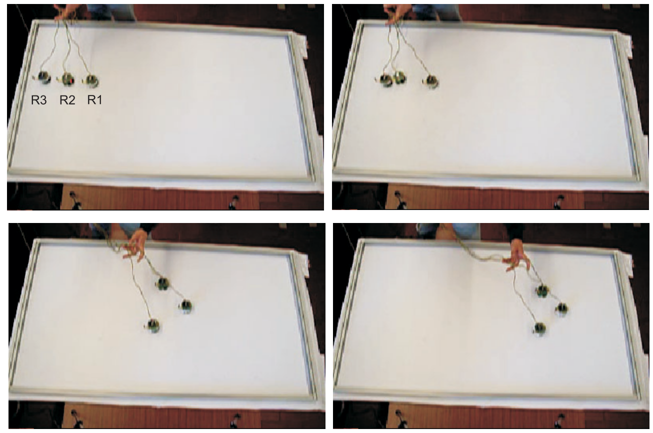


Fig. 3. Video snapshots of three Kheperas switching from a column to a triangle formation. Up left: shows the robots starting position, which is in column with 150mm separation from each other. Up right: at $t = 2s$ the leader, robot $R - 1$ is moving towards the goal and the followers try to position themselves. Because the robot R_3 is moving faster, almost hits R_2. Down left: at $t = 16s$ the team is almost in formation, only the distances are slightly larger than desired. Down right: at $t = 19s$ the team is now in formation

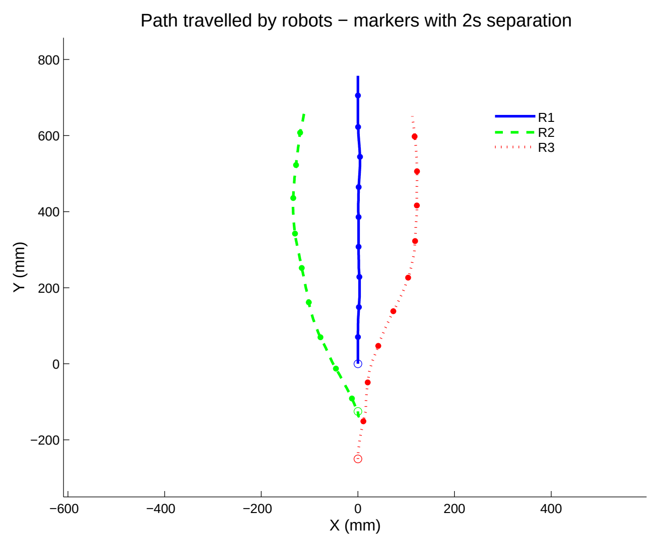


Fig. 4. Plot of the path traveled by three Kheperas in the situation depicted by figure 3. The initial positions are depicted by the the large white circles. Large colored circles appear with 2s interval.

IV. Conclusion and Future Work

In this paper, we have shown how non-linear attractor dynamics can be used as a framework to build controllers that implement teams of robots that navigate according to a prescribed geometric formation while doing obstacle avoidance. The environment is not known a-priori and it can change over time. We have presented real results for teams of Khepera robots performing a line formation and switching from column to triangle formation. Although we have presented our results with only three robots, this framework scales naturally to teams with more robots without extra computation costs. Flexibility in terms of stabilizing the formation versus split to avoid obstacles is inherent to the framework and does not need explicit orders. Further work includes the study of the suitability of this framework to deal with rigid formations, because object transportation is one of our current research interests. Probably this will mean adding some extra work at the coordination level. Another topic will be to supply the individual agents with some cognitive capacities, in terms of memory, anticipation, forgetting, etc., as this will allows us to perform more efficiently some higher level tasks.

Acknowledgments

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). The contribution from Rui Soares are gratefully acknowledged. We also thank the whole *Dynamic Group* and in particular Nuno Fernandes and Paulo Cesar for their help with *DynView*.

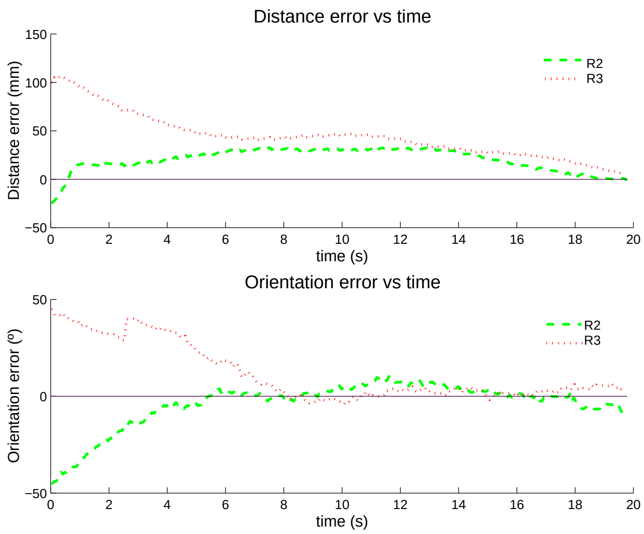


Fig. 5. Plots of the difference between the actual and the desired distance (top plot), $l_i - l_{i,d}$, and of the difference between the actual and desired difference between the heading direction of the follower and the direction at which it sees his leader (bottom plot), $\Delta\psi_i - \Delta\psi_{i,d}$. These plots are shown for the two followers. As expected, as time evolves these values tend to zero, meaning that the robots are closer to formation (when exactly in formation these values are zero) and that the heading direction of the follower converges and follows the moving attractor.

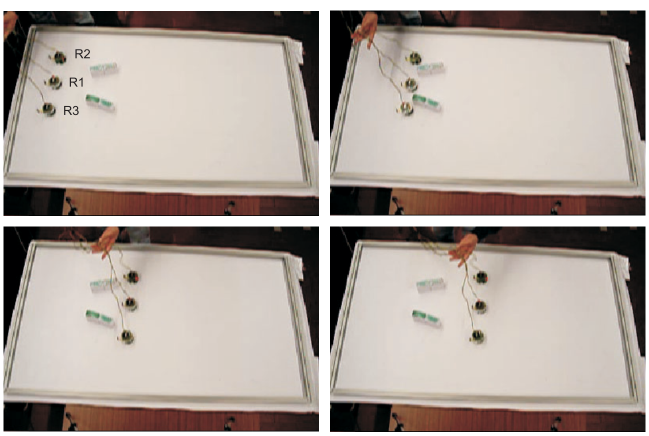


Fig. 6. Video snapshots of three Kheperas moving in line formation. Up left: shows the robots starting position. They have a separation of 150mm from each other. The leader is robot R_1 . Up right: at $t = 4s$ the robots approach the obstacles. Robot R_3 does not have space to pass without leaving formation, thus will have to avoid the obstacle. Down left: at $t = 10s$, after overtaking the obstacle the robot R_3 starts to rejoin the formation. Down right: at $t = 18s$ the robots are again almost in formation.

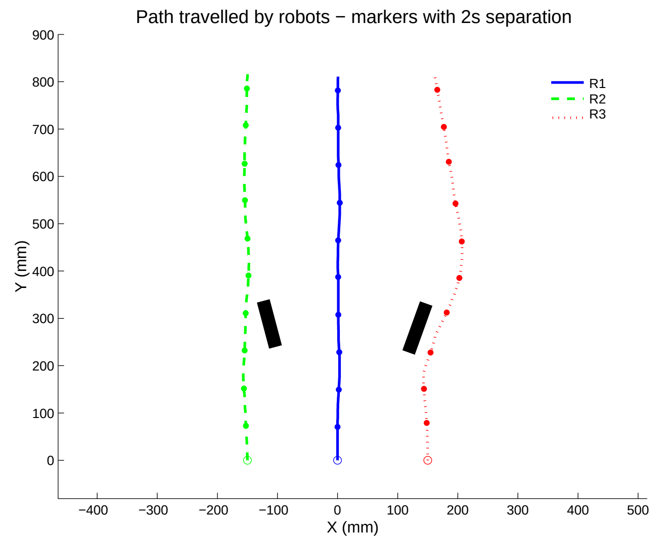


Fig. 7. The path traveled by the tree Kheperas navigating in a line formation, as depicted in figure 6. Obstacles (the two black boxes) are located, roughly, at $(x, y) = (-100, 300)$ and $(x, y) = (150, 300)$, such that robot R_3 has to maneuver around in order to avoid it. As soon as the obstacle is overtaken, it tries again to stabilize the formation.

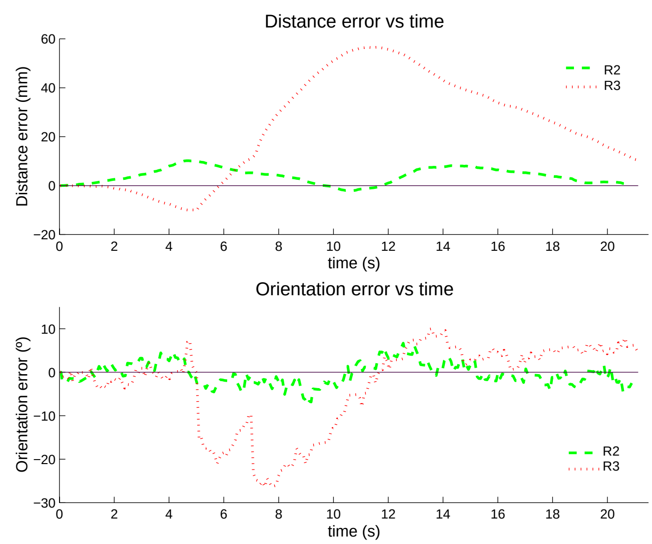


Fig. 8. Plots for $l_i - l_{i,d}$ (top plot) and $\Delta\psi_i - \Delta\psi_{i,d}$ (bottom plot), for the formation in figure 7 are shown here, for the two followers. Robot R_3 notices the presence of the obstacle around $t \approx 5.5s$, as can be seen by the sudden increase in the values for both plots, meaning that a bifurcation in the dynamics has just happened. At $t \approx 11s$ the obstacle is completely overtaken and the robot tries again to stabilize the formation.

REFERENCES

[1] P. Johnson and J. Bay, "Distributed control of simulated autonomous mobile robot collectives in payload transportation," *Autonomous Robots*, vol. 2, no. 1, pp. 43–64, 1995.

[2] H. Yamaguchi, "A cooperative hunting behavior by mobile-robot troops," *The International Journal of Robotics Research*, vol. 18, no. 8, pp. 931–940, September 1999.

[3] F. Bauer, J. Bristow, K. Hartman, D. Quinn, and J. How, "Sattelite formation flying using an innovative autonomous control system (autocon) environment," in *AIAA Guidance, Navigation and Control Conference*, August 1997.

[4] W. Ren and R. W. Beard, "Virtual structure based spacecraft formation control with formation feedback," in *AIAA Guidance and Control Conference*, Monterey, CA, August 2002, aIAA paper n.o 2002-4963.

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

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

[7] B. Young, R. Beard, and J. Kelsey, "A control scheme for improving multi-vehicle formation maneuvers," in *Proc. of the American Control Conference*, Arlington, VA, June 25-27 2001, pp. 704–709.

[8] A. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, "A vision-based formation control framework," *IEEE*

Transactions on Robotics and Automation, vol. 18, no. 5, pp. 813–825, October 2002.
[9] R. Vidal, O. Shakernia, and S. Sastry, “Formation control of nonholonomic mobile robots with omnidirectional visual servoing and motion segmentation,” in *IEEE Conference on Robotics and Automation*, 2003.
[10] J. Fredlund and M. Mataric, “A general local algorithm for robot formations,” *IEEE Transactions on Robotics and Automation, special issue on Multirobot systems*, vol. 18, no. 5, pp. 837–846, October 2002.
[11] R. Olfati-Saber and R. Murray, “Graph rigidity and distributed formation stabilization of multi-vehicle systems,” in *Proc. of the 41st Conference on Decision and Control*, Las Vegas, NV, December 2002.
[12] E. Bicho and S. Monteiro, “Formation control for multiple mobile robots: a non-linear attractor dynamics approach,” in *2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Las Vegas, NV, October 27-31 2003, pp. 2016–2022.
[13] J. Desai, J. 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, December 2001.
[14] E. Bicho, *Dynamic Approach to Behavior-Based Robotics: design, specification, analysis, simulation and implementation*. Aachen: Shaker Verlag, 2000, iSBN 3-8265-7462-1.
[15] 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*, vol. 10, pp. 253–267, 1992.
[16] G. Schöner, M. Dose, and C. Engels, “Dynamics of behavior: Theory and applications for autonomous robot architectures,” *Robotics and Autonomous Systems*, vol. 16, pp. 213–245, 1995.
[17] 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, pp. 424–447, May 2000.
[18] E. W. Large, H. I. Christensen, and R. Bajcy, “Scaling the dynamic approach to path planning and control: Competition amoung behavioral constraints,” *The International Journal of Robotics Research*, vol. 18, no. 1, pp. 37–58, 1999.