A Dynamical Systems Approach to Behavior-Based Formation Control

Sergio Monteiro and Estela Bicho

Department of Industrial Electronics, University of Minho

4800-058 Guimaraes(Portugal)

`sergio.monteiro@dei.uminho.pt, estela.bicho@dei.uminho.pt`

Abstract

Dynamical systems theory is used here as a theoretical language and tool to design a distributed control architecture that generates navigation in formation, integrated with obstacle avoidance, for a team of three autonomous robots. In this approach the level of modeling is at the level of behaviors. A "dynamics" of behavior is defined over a state space of behavioral variables. The environment is also modeled in these terms by representing task constraints as attractors (i.e. asymptotically stable states) or reppelers (i.e. unstable states) of behavioral dynamics. For each robot attractors and reppelers are combined into a vector field that governs the behavior. The resulting dynamical systems that generate the behavior of the robots are non-linear. Computer simulations support the validity of our dynamic model architectures.

1 Introduction

One important and fundamental problem in the control of multiple mobile robots is the maintenance of a geometric configuration during movement. The reason is that there are many interesting tasks (e.g. box pushing [1], payload transportation [2], capturing/enclosing an invader [3]) that require the robots to coordinate their movements more closely. In the literature there are several recent works on formation control (e.g. [1,4,5]). Some of the works reported do not include strategies for obstacle avoidance and others do not deal with autonomy.

To control and coordinate the movement of multiple mobile robots driving in formation there are mainly to types of control: model-based control (e.g. [6]) and behavior-based control (for a review see [7]). In the first, exact models that describe the kinematics/dynamics of the robots and the tasks to be achieved are built. In some cases it is possible to optimize their performance. Despite the high level of theoretical formalization of this approach its success has been limited, particularly for operation in dynamic and unknown environments. By contrast, in behavior-based control exact models are not necessary. The system is structured in terms of elementary behav-

iors, each with a particular goal in "mind". Balch and Arkin [5] have proposed a behavior-based approach to formation control. In their work the individual behaviors are implemented using schema theory.

In this paper we investigate how behavior-based formation control can be modeled by non-linear dynamical systems (or more specifically non-linear attractor dynamics). The work is based on the so called Dynamic Approach to Behavior Generation [8,9] which provide a number of useful concepts. As a case study we choose the example of navigation in triangle formation for a team of three autonomous robots. We assume that the robots have no prior knowledge of the environment and we follow a master-referenced strategy. A *master* robot drives from an initial position to a final target destination. The two other robots (the *slaves*) take the *master* robot as a reference point, and stear so as two keep a triangle formation (see Figure 1). An additional task constraint for the team is that the robots must avoid any obstacles that may appear and, when possible, they must return to the formation.
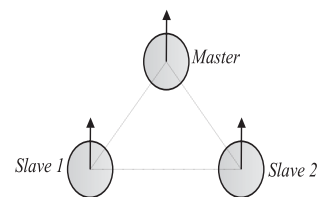

Figure 1: Three robots driving in master-referenced triangle formation.

The control architecture of each robot is structured in terms of elementary behaviors. The individual behaviors and their integration are generated/modeled by non-linear dynamical system and we use bifurcation theory to make design decisions around points at which a system must switch from one type of solution to another. The benefit is that the mathematical properties associated with the concepts (c.f. Section 2) enable system integration including stability of the overall behavior of the autonomous systems.

The rest of the paper is structured as follows: In Sec-

tion 2 we define and describe the behavioral dynamics for each robot in the team. Results obtained from simulation studies are presented in Section 3. The paper ends in Section 4 with discussion, conclusions and an outlook for future work.

2 Attractor dynamics for robot formation

In the following subsection we describe the method used to implement obstacle avoidance and target acquisition for the *master* robot and provide, on that occasion, a brief review of the basic principles of the dynamic approach to behavior generation (see [9,10])for more details). Then in Subsection 2.2 we consider the behavioral dynamics for the two slave robots.

2.1 Behavioral dynamics of the *master*

(1) *Behavioral variables* are used to describe, quantify and internally represent the state of the robot system with respect to elementary behaviors. For target acquisition and obstacle avoidance of the *master* vehicle, the *heading direction*, ϕ_{master} ($0 \leq \phi_{master} \leq 2\pi$), with respect to an arbitrary but fixed world axis, is an appropriate behavioral variable. As is illustrated in Figure 2, the direction, ψ_{tar}, in which a target position lies as seen from the current position of the *master* robot specifies a desired value for the heading direction. Directions, ψ_{obs}, in which obstacles lie specify values of heading direction that must be avoided. The path velocity is also a behavioral variable. (2) The *master* behavior is generated by contin-
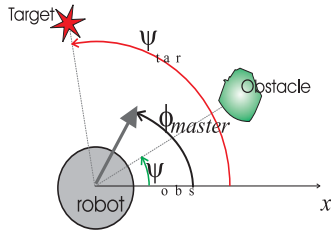

Figure 2: Constraints for the dynamics of ϕ_{master} are the directions at which obstacles and target lie from the current position of the robot, i.e. directions ψ_{obs} and ψ_{tar}.

uously providing values to its behavioral variables, which control the robot's effectors. The time course of these values is obtained from solutions of a dynamical system. The attractor solutions (asymptotically stable states) dominate these solutions by design. In the present system, the *behavioral dynamics* of heading direction, $\phi_{master}(t)$, is a differential equation

$$\dot{\phi}_{master} = f_{master}(\phi_{master}) \qquad (1)$$
$$= f_{obs}(\phi_{master}) + f_{tar}(\phi_{master})$$

built from a number of additive contributions that express task constraints. Each constraint is cast either as a repulsive (Figure 3) or as an attractive force-let (Figure 4),

which are both characterized by three parameters: (a) which value of the behavioral variable is specified (e.g., ψ_{obs} or ψ_{tar}); (b) how strongly the repulsion or attraction effect is; and (c) over which range of values of the behavioral variable this force-let acts. Thus, in isolation, each force-let erects a repeller (unstable state) or an attractor (asymptotically stable state) of the dynamics of the behavioral variable. An attractive force-let serves to attract the system to a desired value of the behavioral variable (here the direction in which a target lies). A repulsive force-let is used to avoid the values of the behavioral variable that violate a task constraint (here the directions in which obstacles lie). Note that a repulsive force-let erects two semi-attractors at the left and right boundaries of the repulsive zone (corresponding to passing on the left or right of the obstacle). This method of constructing a
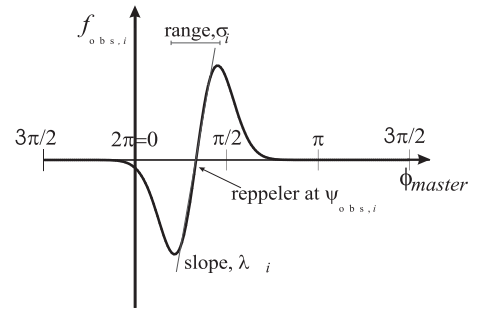

Figure 3: A contribution to the dynamics of heading direction expressing the task constraint "avoid moving toward obstacles" is a force-let with a zero-crossing at the direction, $\psi_{obs,i}$ at which an obstruction has been detected. Every distance sensor ($i = 1, 2, \ldots, 7$) contributes such a force-let centered on the direction in which the sensor points. The positive slope of force at the zero-crossing makes that direction a repeller. By decreasing this slope with increasing measured distance, only nearby surfaces repel strongly. The range of the force-let is limited based on sensor range and on the constraint of passing without contact.

behavioral dynamics can be used on systems with low-level sensors by defining a force-let for each sensor (see [11]). The three parameters defining each force-let are obtained from sensory input. For example the *master* robot used in this project has seven distance sensors mounted on a ring which is centered on the robot's rotation axis. These are used to measure the distance to surfaces at the height of the robot, which are thus obstacles. Each sensor ($i = 1, 2, \ldots, 7$) contributes with a repulsive force-let

$$f_{obs,i} = \lambda_i(\phi_{master} - \psi_{obs,i}) \exp\left[-\frac{(\phi_{master} - \psi_{obs,i})^2}{2\sigma_i^2}\right] \tag{2}$$

illustrated in Figure 3. Here $\psi_{obs,i}$ is the direction in the world in which sensor i is pointing. As the heading direction, ϕ_{master}, is defined relative to the same reference frame, the relevant difference, $\phi - \psi_{obs,i} = -\theta_i$ is actually a constant, the inverse of the angle, θ_i, at which the distance sensor i is mounted relative to the forward direction

of the robot.

Orientation toward the target which lies at direction ψ_{tar} is brought about by erecting an attractor at this direction with strength λ_{tar}. Because target acquisition is desired from any starting orientation of the *master* robot the range over which this contribution exhibits its attractive effect is the entire full circle, i.e. from 0 to 2π rad. As a consequence, there is a repeller at the back, in the direction opposite to that toward the target (see Figure 4). A simple mathematical form can be:

$$f_{\text{tar}}(\phi_{master}) = -\lambda_{\text{tar}} sin(\phi_{master} - \psi_{\text{tar}}) \qquad (3)$$
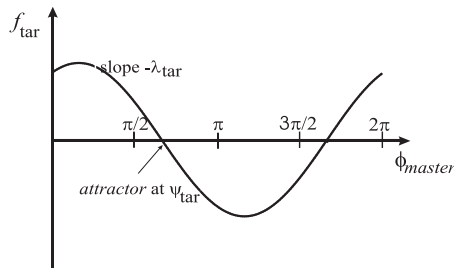
Figure 4: A contribution to the dynamics of heading direction expressing the task constraint "move toward the target" is a force with a zero-crossing at the specified direction toward the target, ψ_{tar}. The negative slope at the zero-crossing makes this an attractor of the dynamics. The target contribution is sinusoidal and extends over the entire range of heading direction. This leads to a repeller in the direction $\pi + \psi_{\text{tar}}$ opposite to ψ_{tar}.

The target contribution and the contributions arising from the detected obstacles all act at the same time. The *master* heading direction dynamics is thus simply the sum over these:

$$\dot{\phi}_{master} = \sum_{i=1}^{7} f_{\text{obs},i}(\phi_{master}) + f_{\text{tar}}(\phi_{master}) \qquad (4)$$

More sophisticated control over activation and deactivation of such contributions can be obtained using activation networks [12,13] but is not necessary here. Since some of the force-lets have limited range, this superposition is a non-linear dynamical system, which may have multiple attractors and repellers (typically few). By design the system is tunned so that the heading direction is in a resulting attractor of this dynamics (i.e. Eq. 4) most of the time (see Figure 5).

Up to this point we have only addressed the control of the master's heading direction. For this robot to move it must have some path velocity, of course. As it moves, sensory information changes and thus attractors (and repellers)
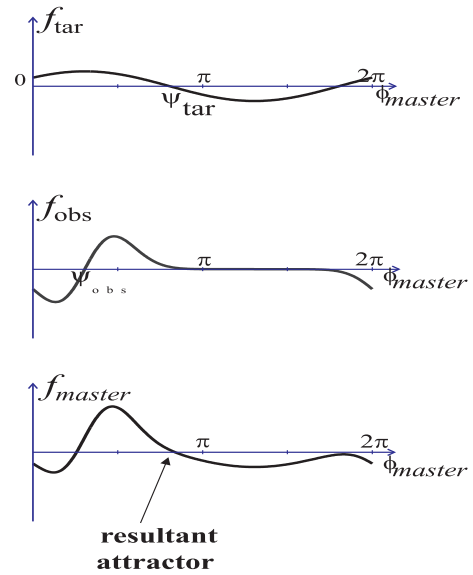
Figure 5: Resultant attractor (bottom Panel) from the superposition of the repulsive force-let (middle Panel) from obstacle constraints and attractive force (top Panel) due to the target constraint. Parameters must be tuned so that the system is relaxed in the attractor.

shift. The same happens if obstacles or the target move (c.f. Subsection 2.2) in the world. Since the heading direction must be in or near an attractor at all times, for the design principle to work, we must limit the rate of such shifts to permit the robot's heading direction to track the attractor as it moves and thus stay close to a stable state. One way this can be accomplished is by controlling the path velocity, v_{master} of the vehicle as explained in [14].

The complete behavioral dynamics for the *master* robot has been implemented and evaluated in detail on a physical mobile robot (see e.g Chapter 4 in [10]).

2.2 Behavioral dynamics of the slaves
We have two *slave* robots and the task requirements of each one is to stear at a particular orientation with respect to the *master* robot and to avoid collisions with obstacles.

We begin by making some simplifying assumptions: First, the *slave* robots are "identical" to the *master* robot. Second, the *slaves* behavior is governed by the same behavioral dynamics as for the *master* robot. The behavior of each *slave* is therefore described by identical dynamical systems but generated independently by the time course of their own behavioral variables (ϕ_j, v_j, $j= slave_1$, $slave_2$). Third, if the robots come too close to each other, they must avoid collisions among them as they avoid collisions with obstacles (stationary or not).

Slave 1 must steer so as to maintain the *master* to its right side. Its behavioral dynamics is governed by:

$$\dot{\phi}_{slave_1} = f_{slave_1}(\phi_{slave_1}) \qquad (5)$$
$$= f_{\text{obs}}(\phi_{slave_1}) + f_{master,slave_1}(\phi_{slave_1})$$

where $f_{\text{obs}}(\phi_{slave_1})$ erects repellers at the directions at which this robot detects obstructions, and has the same functional form of Eq.(2). $f_{master,slave_1}$, has the functional form of Eq.(3), and erects an attractor at direction $\psi_{desired,slave_1} = \psi_{master,slave_1} + \Delta\psi$ pointing as depicted in the top panel of Figure 6.

Conversely, *Slave* 2 must drive so as to maintain the *master* to its left side. Its behavioral dynamics is governed by:

$$\dot{\phi}_{slave_2} = f_{slave_2}(\phi_{slave_2}) \qquad (6)$$
$$= f_{\text{obs}}(\phi_{slave_2}) + f_{master,slave_2}(\phi_{slave_2})$$

but where $f_{\text{obs}}(\phi_{slave_2})$ erects reppelers at the directions at which *slave* 2 senses obstructions and $f_{master,slave_2}$ constructs an atractive force at direction $\psi_{desired,slave_2} = \psi_{master,slave_2} - \Delta\psi$ which attracts *slave* 2 to drive as illustrated in the bottom panel of Figure 6.
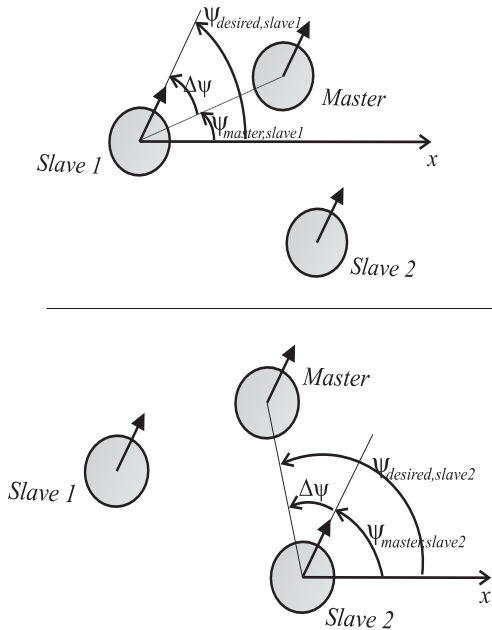

Figure 6: Top Panel: the task constraint for *slave* 1 is to drive so as to keep the *master* robot at an angle $\Delta\psi$ to the right of its heading direction. The desired heading direction (i.e. attractor) for this slave is therefore $\psi_{desired,slave_1}$ with respect to an external (but arbitrary) axis. Bottom Panel: analogously, *slave* 2 must stear at a direction $-\Delta\psi$ with respect to the direction at which it "sees" the *master*. This means that the desired heading direction for this slave is $\psi_{desired,slave_2}$. $\Delta\psi$ permits to control the exact shape of the triangle formation.

The path velocity of each *slave* (v_i, $i = slave_1$, $slave_2$) is controlled so that its "time to contact" with the mas-

ter is kept constant. The "time to contact" is chosen much larger than the relaxation time of their heading direction dynamics. This way we can guarantee that the slaves heading direction can track and follow one of the resultant attractors in the corresponding dynamics. Thus, their *behavioral variables* (or state variables) are in an asymptotically stable state (although this moves).

3 Results

The complete dynamic architectures were evaluated in computer simulations. These were generated by a software simulator written in MATLAB. The simulated environments are defined as 2D occupancy matrixes. The robots have no information about the environment. We modeled the robotic platforms, based on the physical prototype in which the dynamic control architecture described in Subsection 2.1 has been previously implemented (see [3] for more details than those presented here). The seven infra-red detectores (which are used on the physical prototype to measure distances to detected obstructions) are modeled as distance sensors. These are simulated through an algorithm reminiscent of ray tracing. The target information is defined by a goal position in space which is only known by the *master* robot. It is assumed here that the *master* robot broadcasts its current position to the *slaves*.

In the simulation the robots are represented as triplets (x_j, y_j, ϕ_j) ($j=$ *master*, $slave_1$, $slave_2$), consisting of the corresponding two Cartesian coordinates and the heading direction. Cartesian coordinates are updated by a dead-reckoning rule ($\dot{x}_j = v_j cos(\phi_j)$, $\dot{y}_j = v_j sin(\phi_j)$) while heading direction, ϕ_j, and path velocity, v_j, 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.

Figure 7 shows a simulation run of the complete system which demonstrates the smooth behavior consistent with all imposed constraints.

Figure 8 demonstrates that scenarios with convex obstacles are not an absolute limitation to our approach.

Figure 9 shows snapshots of a third simulation run of the complete system. Figures 10 and 11 depict the heading direction dynamics for each robot, at the points showed in snapshots A and C respectively.

4 Conclusion

We have demonstrated how non-linear attractor dynamics can be used to design a dynamic control architectures
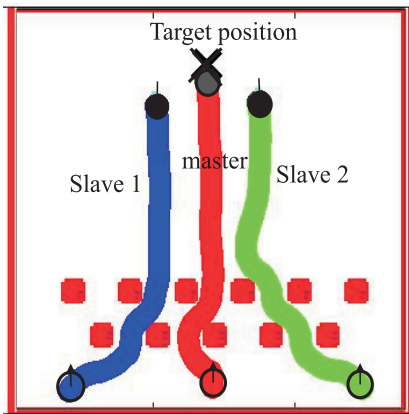
Figure 7: The trajectories for the three robots generated by the complete behavioral dynamics implemented in simulation. The target is indicated by a cross on the top. The robots are indicated by circles with a hair indicating their heading direction. Initially, the robots are positioned in line in the bottom of the arena. The robots avoid the obstacles and as soon as it is possible they drive in triangular formation.
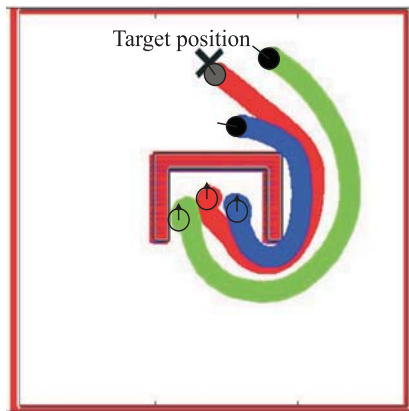

Figure 8: The trajectories for the three robots generated by the complete behavioral dynamics when the robot team is faced with a convex obstacle. The robots' initial positions are indicated by the open circles with a hair indicating their heading direction. The robots move smoothly around the convex obstacle toward the target while simultaneously trying to maintain the triangle formation

Figure 9: Snapshots of a simulation run of the complete system. The robots are positioned initially as illustrated in Panel A. The *master* robot is positioned in the middle and pointing at a direction 270 deg. *Slave* 1 is placed to the right side of the *master* pointing at direction 0 deg. *Slave* 2 is placed on the left side of the *master* and pointing at direction 180 deg. As one can see the two slaves avoid each other (Panel B) and then these stear so that *slave* 1 tries to put itself to the left of *master* while *slave* 2 tries to put itself to the right (Panels B-C). Then they drive in triangle formation. In tight spaces the triangle is more narrow (Panels C-D). When possible the triangle formation assumes the specific shape determied by parameter $\Delta\psi(= 45deg)$.

that enables a team of three robots to drive in triangle formation and simultaneously avoid obstacles. The robots have no prior-kowledge of their environment. Simulation studies have shown that the generated trajectories are smooth. Flexibility is achived in that as the sensed world changes, the systems may change their planning solutions continuously but also descontinuously (tunning the triangle formation versus split to avoid obstacles).

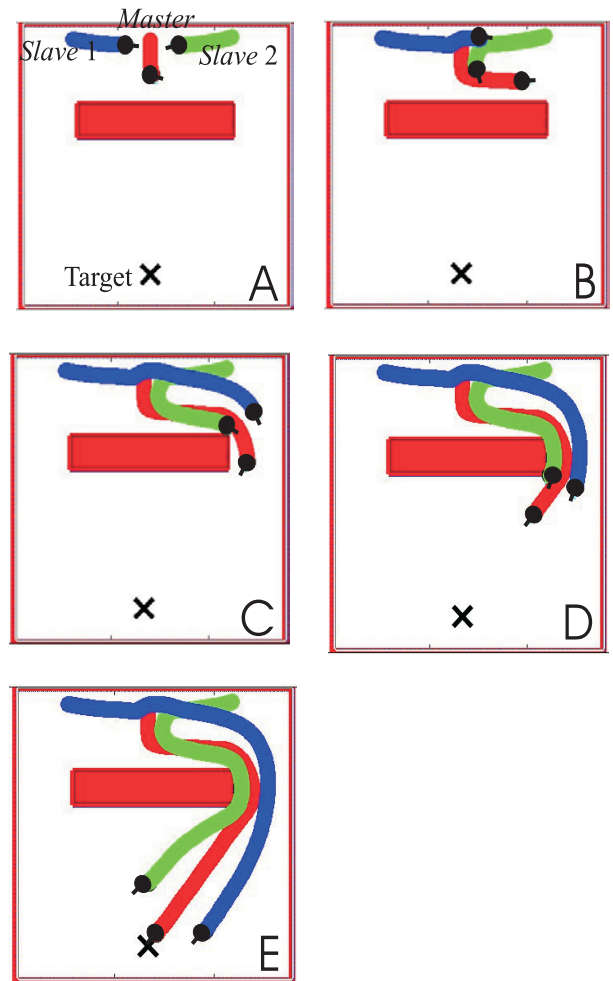The work described here imposes of course further research. The complete distributed dynamical architecture must be implemented (and evaluated) in a robot team composed of three physical autonomous robots. We will first assume that the *master* robot communicates the slaves its position in space. Then, no explicit communication will be assumed and the *slaves* will be responsable to estimate the direction at which the *master* lies with respect to themselves by making use of low-level sensors. To behave inteligently the robots will have to be endowed with cognitive behaviors like antecipition, memory, fogetting and robustness against noisy and contradictory sensory information. To generate these more abstracts forms of behavior we will use representations of information based on dynamic neural fields [14,15].
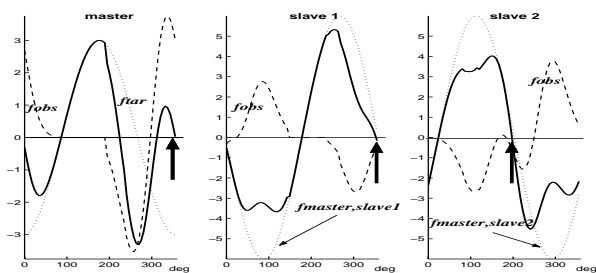
Figure 10: Heading direction dynamics for the three robots when they are at points depicted in snapshot A in Figure 9. These three plots illustrate the heading direction dynamics of the *master*, *slave* 1 and *slave* 2 respectively. Left plot: Contribution of the sensed obstructions is the dashed line (f_{obs}) and the doted line represents the contribution of the target (f_{tar}). The resultant dynamics (i.e. sum over all contributions) is the black continuous line. Middle Plot: Obstacle contributions (f_{obs}) to the dynamics of *slave*1 is the dashed line and the contribution that makes this robot to follow the *master* ($f_{master,slave_1}$) keeping it to its right side is the doted line. The resultant dynamics is the black continuous line. Right Plot: The individual contributions (f_{obs} and $f_{master,slave_2}$) and their superposition for the heading direction of *slave* 2. The black arrow in each plot indicates the current state (i.e. heading direction in the world) of the corresponding robot. As one can see the heading direction of each robot is very close to an attractor (asymptotically stable state) of the resultant dynamics.
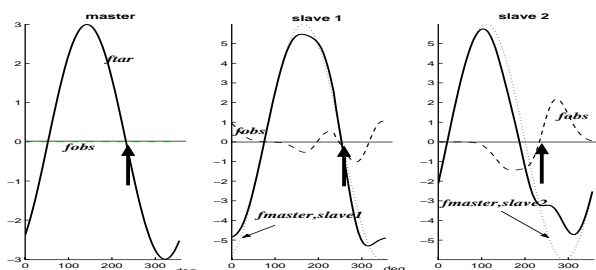

Figure 11: Heading direction dynamics for the three robots when they are at points depicted in snapshot C in Figure 9. Again, the headind direction of the robots is very close to a fixed point attractor, except for *slave* 2 because at the time of this snapshot a bifurcation had just occured in the heading direction dynamics of this robot.

Acknowledgements

References

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

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

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

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

[5]   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.*

[6]   M Uchiyama and P Dauchez. A symmetric hybrid position/force control scheme for the coordination of two robots. *in Proc. IEEE Int. Conf. Robotics and Automation, 350–355, 1988.*

[7]   R C Arkin. *Behavior-Based Robotics*. MIT Press, Cambridge, 1998.

[8]   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.

[9]   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.

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

[11]   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.

[12]   A Steinhage. *Dynamical Systems for the Generation of Navigation Behavior*. Shaker Verlag, Aachen, 1998.

[13]   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.

[14]   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.*

[15]   W Erlhagen. *Lokalisierte, stationäre Verteilung in neuronalen Feldern (Localized stationary distribuitions in neural fields)*. Harri Deutsch, Frankfurt, 1997.