Proceedings of the 1997 IEEE
International Conference on Robotics and Automation
Albuquerque, New Mexico - April 1997

Dynamic Robot Planning: Cooperation through Competition*

Edward W. Large*, Henrik I. Christensen† & Ruzena Bajcsy*

* GRASP Laboratory
University of Pennsylvania
3401 Walnut Street, Room 301C
Philadelphia, PA 19104-6228

† Laboratory of Image Analysis
Aalborg University
Fr. bajers Vej 7D
DK-9220 Aalborg Ø

Abstract

We address scaling of the "dynamic systems" approach for robot planning to multi-agent cooperation. To accommodate this extension it is necessary to carefully consider how individual behaviors contribute to the vector field. To avoid spurious minima and related problems a competition dynamics is introduced and its stability is analyzed. A system of two cooperating agents is designed, and examples are presented to illustrate the utility of this approach.

1 Introduction

Over the past twenty or so years there has been a great deal of research in the field of robot path planning and control. Much of this work has focused on finding the best or most appropriate space in which to represent the robot actions during the navigation task. In spite of this effort, however, the issue of what is the best space in which to represent robot behavior remains an open question.

Geometric representations (e.g. [1, 2]) model the geometry of the agent and the external environment. The difficulty with the geometric approach, however, is that it is too static. Configuration space representations [3, 4, 2] include geometry and kinematics. The difficulty here is that these spaces are complex and only simple configurations are computationally feasible. Potential field representations [5, 6] build upon configuration space representations, defining a state space over which a potential field can be defined.

The above approaches rely upon global representations of the world in which the robot operates. Another possibility is to define a representation in which the dimensions correspond to robot behavior (e.g. [7, 8]). Using such a representation Schöner and colleagues have developed a "dynamical systems" approach for robot path planning and control. In this approach a set of behavioral variables defines a state space in which a dynamics of robot behavior is described [9, 10].

This approach has several advantages. The level of modeling is at the level of behaviors. The dimensions of the state space correspond to *behavioral variables*, such as heading direction and velocity. The environment is also modeled at a behavioral level. The environment provides *task constraints*, that provide the system with behavioral information. Task constraints are modeled as component forces that define attractors and repellors of a dynamical system. The contributions are combined into a single vector field by additive composition. Planning and control governed by a dynamical system that generates a time course of the behavioral variables. The dynamics are specified as a vector field that governs system behavior.

In this paper, we investigate the scalability of the dynamic approach. In particular, we investigate its applicability to the control of a pair of cooperating robots. The problem we will encounter is that non-independent contributions to the vector field can create spurious attractors and cause related problems. We propose a solution that involves competition among task constraints. We deal with multiple constraints by dynamically computing weighting coefficients that determine the relative contribution of different task constraints at any given time. Competition among behavioral constraints leads to the generation of behavioral sequences.

2 Dynamic Planning and Control

The behavior of an agent is modeled as a time course of behavioral variables generated by a dynamics that incorporates both planning and control knowledge. We focus on the dynamics of heading direction, assuming that velocity can be appropriately controlled (see, for example, [11]). For our one-dimensional system, the dynamics take the following form.

$$\dot{\phi} = f(\phi). \qquad (1)$$

*Work supported in part by: NSF Grants IRI93-07126 and SBR89-20230, as well as ARMY Grant DAAH04-96-1-0007 and ARPA Grant N00014-92-J-1647. It was carried out during a 6 month visit to the GRASP Laboratory by the second author.

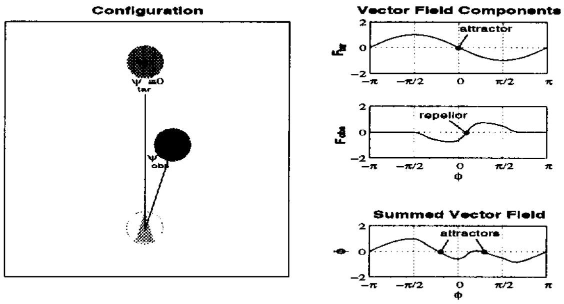

Figure 1: Task constraints and their contributions to the vector field. A target (attractor, F_{tar}); an obstacle (repellor, F_{obs}); their composition, $\dot{\phi} = F_{tar} + F_{obs}$.

Task constraints define contributions to the vector field, $f(\phi)$, by modeling desired behaviors (e.g. heading toward the target) as attractors,

$$F_{tar} = -a * \sin(\phi - \psi_{tar}), \quad (2)$$

and to-be-avoided behaviors (e.g. heading toward an obstacle) as repellors (see Figure 1).

$$F_{obs_i} = R_{obs_i} \times W_{obs_i} \times D_{obs_i} \quad (3)$$

The repellor corresponding to an individual obstacle is the product of three functions. R_{obs_i} sets up a generic repellor in the direction of the obstacle, W_{obs_i} limits the angular range, and D_{obs_i} scales the strength according to the obstacle's distance from the agent. Thus, the contribution of each obstacle is range limited. Details regarding the specific functional forms can be found in [9]. Multiple obstacles are handled by summing the contributions of individual obstacles.

$$F_{obs} = \sum_{i=1}^{n} F_{obs_i} \quad (4)$$

Finally, the contributions of individual task constraints are combined additively into a single vector field as illustrated in Figure 1.

$$\dot{\phi} = F_{tar} + F_{obs} + noise \quad (5)$$

Because certain constraints are modeled as repellors, the planning dynamics is augmented by a stochastic term that guarantees escape from unstable fixed points.

2.1 Scaling the Dynamic Systems Approach

In a simple navigation system, with two behavioral constraints, *target* and *obstacles*, the above approach works well [9]. But can the same approach be used to describe more complex tasks? Consider a system in which we have two agents, and both agents must obey the same constraints as in the above system, i.e. they must perform a navigation task. Further, we add the constraint that they must stay near one another as they make their way toward the target location. We will call this additional constraint *other*. Thus, each agent must respect three behavioral constraints. Similarly to target seeking, we model this new constraint as a global attractor.

$$F_{oth} = -a \sin(\phi - \psi_{oth}) \quad (6)$$

Then the contribution of *other* is additively combined to the composite vector field (Figure 2):

$$\dot{\phi} = F_{tar} + F_{obs} + F_{oth} + noise \quad (7)$$

In addition, we assume that if the agents come too close to one another, they are to avoid collision in the same way as they would avoid stationary obstacles.

2.2 Spurious Attractors and Constraint Averaging

Unless care is taken, as the number of constraints grows, non-independent contributions to the vector field can combine in such a way that they give rise to attractors corresponding to undesired behaviors, such as running into obstacles or getting stuck in an area and never reaching a target location. Figure 2 shows a spurious attractor. Two obstacles are situated in front of the agent in such a way that there is almost, but not quite enough space for the agent to pass between them. If only the contribution of *obstacles* to the vector field is considered, a repellor with a shallow slope is created at their average location. If the target is placed behind the obstacles, so that its attractor contribution to the vector field collides with this repellor, an attractor is created between the two obstacles. The agent will get stuck at this location.

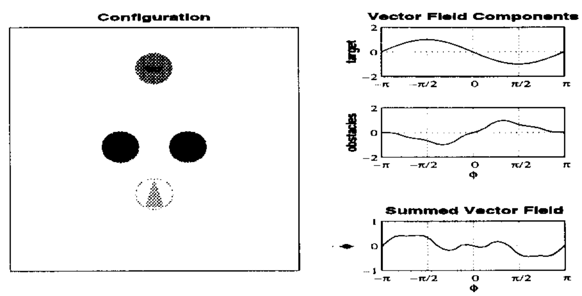

Figure 2: A spurious attractor.

A related problem is constraint averaging. This crops up in the composition of *target* and *other*, as depicted in Figure 3, where two agents are headed toward the target, yet one is considerably ahead of the other. We see the situation from the point of view of Agent 2 (middle). The target and the other agent

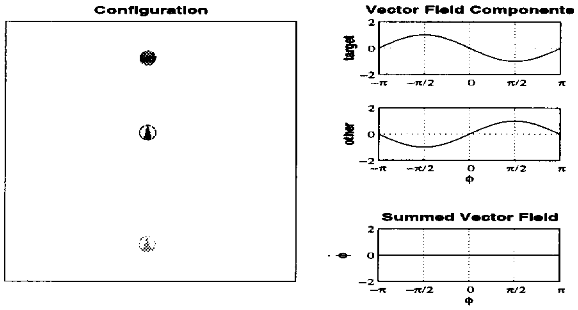

Figure 3: Constraint averaging.

lie in opposite directions, so in composition these two contributions cancel. The problem here is that non-independent contributions to the vector field combine in such a way that a single attractor lies in their average direction. In some cases this yields appropriate behavior, but in most cases (e.g. Figure 3) is does not.

3 Competition Among Task Constraints

To deal with such situations, we modify the strength of each contribution with a weight that is assigned to each type of task constraint.

$$\dot{\phi} = |w_{tar}|F_{tar} + |w_{obs}|F_{obs} + |w_{oth}|F_{oth} + noise \quad (8)$$

Weights are assigned through a competitive dynamics that operates at a faster time scale than the behavioral dynamics. This determines the strength of each contribution depending upon the current situation.

$$\dot{w}_i = \alpha_i w_i(1 - w_i^2) - \sum_{j \neq i} \gamma_{j,i} w_j^2 w_i \quad (9)$$

In Equation 9, $i, j \in \{tar, obs, oth\}$ index the constraints. The parameters to the competition dynamics are the α_i and the γ_{ji} referred to as *competitive advantage* and *competitive interaction*, respectively. The competitive advantage of each constraint is determined by its applicability in the current situation, while competitive interaction summarizes the degree to which the constraint is consistent or inconsistent with other active constraints. Given appropriately chosen functions that tie the competition parameters to situations in the environment, we show that this type of competition allows us to scale the dynamic approach to the design of a system of three task constraints. Competition also produces simple and complex sequences of behavior that are generated opportunistically, in response to specific environmental situations.

4 Cooperation through Competition

Our development will proceed in three stages. First, we perform a stability analysis that will tell us how relative values of the parameters α_i and γ_{ji} determine the weights of the component behaviors. Second, we identify situations where constraints are incompatible, leading to the design of functional forms that tie competitive interactions, i.e. the γ_{ji}, to specific situations. Finally, we determine which environmental situations call for the activation of which behaviors. This leads to the design of functional forms for the competitive advantage, α_i, of the constraints.

4.1 Stability Analysis

A linear stability analysis (e.g. [12]) was performed on the system described by Equation 9 assuming $\alpha_i, \gamma_{j,i} > 0$, for the case of three behavioral constraints. The analysis reveals the qualitative behavior of the competitive dynamics, by enumerating the set of equilibrium points for the three-dimensional system and classifying each equilibrium point according to its stability. Because the stability of each equilibrium point changes depending upon the values of the parameters α_i and γ_{ji}, we also computed a set of stability conditions. The analysis (Table 1) revealed eight equilibrium points, corresponding to eight unique behaviors that can be generated by the agent.

The stability analysis reveals three classes of stable solutions. Rather than describe each behavior individually, we describe only each class of stable solutions. This will make the job of understanding the competitive dynamics easier. It will also illustrate important features regarding the scalability of this approach.

The first class of solutions corresponds to one constraint being activated, and the others deactivated. Let us refer to the active behavior as behavior i. This solution is stable so long as $\gamma_{i,j} > \alpha_j$, $\forall\, j \neq i$. In other words, behavior i is the sole winner of the competition whenever it actively inhibits every other behavior.

The second class of solutions corresponds to two constraints being activated and the third deactivated. Let i and j be the activated constraints, and k be the deactivated constraint. Then this solution is stable whenever $\alpha_i > \gamma_{j,i}$ and $\alpha_j > \gamma_{i,j}$. Additionally, it must be the case that $\gamma_{i,k} > \alpha_k$ or $\gamma_{j,k} > \alpha_k$. The latter condition says at least one of the active constraints must be inhibiting behavior k. This so-called "averaging solution" is given by:

$$A_{i,j} = \sqrt{\frac{\alpha_i\alpha_j - \alpha_j\gamma_{j,i}}{\alpha_i\alpha_j - \gamma_{i,j}\gamma_{j,i}}} \quad (10)$$

If there is no competition between constraints, $\gamma_{i,j} = 0$, $\forall\, i, j$, both constraints are activated at full strength.

w_{tar}	w_{obs}	w_{oth}	Stability
0	0	0	Unstable
± 1	0	0	$\gamma_{tar,j} > \alpha_j$, $\forall j \neq tar$
0	± 1	0	$\gamma_{obs,j} > \alpha_j$, $\forall j \neq obs$
0	0	± 1	$\gamma_{oth,j} > \alpha_j$, $\forall j \neq oth$
$\pm A_{tar,obs}$	$\pm A_{obs,tar}$	0	$\alpha_{tar} > \gamma_{obs,tar}$ $\alpha_{obs} > \gamma_{tar,obs}$ $\gamma_{i,oth} > \alpha_{oth}$, $i \in \{tar, obs\}$
$\pm A_{tar,oth}$	0	$\pm A_{oth,tar}$	$\alpha_{tar} > \gamma_{oth,tar}$ $\alpha_{oth} > \gamma_{tar,oth}$ $\gamma_{i,obs} > \alpha_{obs}$, $i \in \{tar, oth\}$
0	$\pm A_{obs,oth}$	$\pm A_{oth,obs}$	$\alpha_{obs} > \gamma_{oth,obs}$ $\alpha_{oth} > \gamma_{obs,oth}$ $\gamma_{i,tar} > \alpha_{tar}$, $i \in \{obs, oth\}$
$\pm A_{tar,obs,oth}$	$\pm A_{obs,tar,oth}$	$\pm A_{oth,tar,obs}$	$\alpha_{tar} > \gamma_{i,tar}$ $\forall j \neq tar$ $\alpha_{obs} > \gamma_{i,obs}$ $\forall j \neq obs$ $\alpha_{oth} > \gamma_{i,oth}$ $\forall j \neq oth$

Table 1: Fixed points and stability conditions for competitive dynamics.

The resulting behavioral dynamics reduces to that described by [9]. If there is some competition, both are still active, but at reduced levels.

The final class of solutions consists of a "3-constraint averaging" solution, where all three contributions are active. This point is stable as long as $\alpha_i > \gamma_{j,i}$ for all $j \neq i$. While it is possible to write down a closed form for this solution, the equation is somewhat messy and uninformative, so it is not included here.

This analysis points to some interesting scaling properties of the competitive dynamics. First, a complex conspiracy of competitive interactions is not required to activate or deactivate a constraint. The important implication of this observation is that we can design the competitive dynamics simply by considering pairs of behaviors. Second, we can count the number of unique behaviors that arise in a n-constraint system. It is simply the number of ways to chose one active behavior, plus the number of ways to chose two active behaviors, and so on. In other words, the number of behaviors generated in such a system is:

$$N = \binom{n}{1} + \cdots + \binom{n}{n} = \sum_{i=1}^{n} \binom{n}{i} = 2^n - 1 \quad (11)$$

The stability analysis reveals important facts about the scalability of this approach. In an n-constraint system, competition provides $2^n - 1$ unique behaviors. Designing the system requires at most n^2 design decisions: $n^2 - n$ competitive interactions, plus n competitive advantages. Furthermore, the analysis describes how the relative values of the $\gamma_{j,i}$ and α_i determine which behavior arises from the competitive dynamics. We use this information to design functions that tie competitive interaction and competitive advantage to situations in the environment.

4.2 Competitive Interaction

In this section, we determine competitive interaction among the constraints. We begin by detecting the situations in which *target* is incompatible with *obstacles*, with the goal of preventing the creation of spurious attractors. Our strategy is based on the observation that whenever an attractor and a repellor collide (e.g. Figure 2), unwanted consequences may result, because the two contributions are non-independent and contradictory.

We design "fixed point detectors" that describe the location and stability of the fixed points for each contribution to the behavioral dynamics. We then use these functions to define competitive interaction between the two task constraints. For the *target* contribution, we use:

$$P_{tar} = sgn(\frac{dF_{tar}}{d\phi})\ e^{-c_1|F_{tar}|} \quad (12)$$

This function has two factors. The first calculates the sign of the slope of the vector field contribution. This determines whether a fixed point is an attractor (negative slope) or a repellor (positive slope). The second finds fixed points, using a function that has a value of one when the vector field contribution is equal to zero, and quickly falls to zero as the magnitude of the contribution grows. The parameter $c_1 > 0$ determines the rate of fall off around the fixed points. Thus, P_{tar} has a value of one at a repellor, minus one at an attractor, and values approaching zero elsewhere.

The situation is slightly more complicated for *obstacles*. Because obstacle contributions are range limited, i.e. have values near zero outside an obstacle's range, Equation 12 will identify these areas as fixed points. Thus, for *obstacles*, we sum the range-limiting functions for the obstacles, saturating to insure boundedness, and use the result as a multiplicative factor.

$$P_{obs} = \tanh(W_{obs})\ sgn(\frac{dF_{obs}}{d\phi})\ e^{-c_1|F_{obs}|} \quad (13)$$

where $W_{obs} = \sum W_{obs_i}$ (see [9]). As above, this function has a value of one at a repellor, minus one at an attractor, and values approaching zero elsewhere.

Next we use P_{tar} and P_{obs} to construct a function that describes the competitive interaction between *obstacles* and *target*:

$$\gamma_{obs,tar} = \frac{e^{-c_2 P_{tar} P_{obs}}}{e^{c_2}} \quad (14)$$

This function is strongly peaked at the point of attractor-repellor collision. The constant $c_2 > 0$ determines the rate of drop off around the collision.

Next we choose the competitive interaction between *target* and *obstacles*. For the current navigation task, it is never appropriate for the agent to turn off the *obstacles* constraint. Thus, we choose a small constant value for $\gamma_{tar,obs}$, allowing this constraint to be activated whenever the agent approaches an obstacle.

Finally, we consider the *other* constraint. In most situations the we will want to enforce competition between the *target* and *other*, although there will also be in some situations in which moving in the average direction is the appropriate behavior. When the target and the other agent are in opposite directions we wish to force a decision, but when they lie in the same direction, both constraints can be satisfied simultaneously. We can accomplish this type of competitive interaction using the following function.

$$\gamma_{oth,tar} = b_1(\tanh(-b_2 \cos(\psi_{tar} - \psi_{oth}) + b_3) + 1) \quad (15)$$

Competition is high except for a certain region around an angular difference of zero. The size of this region can be adjusted using the constant b_3, while the slope of the boundary is adjusted using b_2. b_1 sets the maximum level of competitive interaction.

4.3 Competitive Advantage

In this section, we choose values for the competitive advantages so that, in situations where the two behaviors compete, we can determine the outcome of the competition. First, we note that *target* should be turned on whenever possible. Using knowledge of the background level of competition created by Equation 14, we choose a constant value for α_{tar}. Whenever *obstacles* actively competes with *target*, $\alpha_{tar} < \gamma_{obs,tar}$, and target will be deactivated.

Next, we must decide how to set competitive advantage for the obstacle contribution. Intuitively, we observe that obstacles should have high competitive advantage when they are nearby and/or when there are many of them around the agent. The function D_{obs_i} (a component of F_{obs_i}, see [9]) grows exponentially fast as the agent approaches an obstacle. To count the number of obstacles around the agent, weighted exponentially according to distance, we sum the D_{obs_i}, saturating to ensure boundedness. This yields the following function for competitive advantage.

$$\alpha_{obs} = \tanh(\sum_{i=1}^{n} D_{obs_i}) \quad (16)$$

Finally we determine the competitive advantage of *other*. In terms of competitive advantage, *other* is somewhat different from *target*. Because we simply want the agents to remain "near" one another, we want to deactivate *other* when the agents are close enough. Thus we choose:

$$\alpha_{oth} = \tanh(\frac{e^{r_{oth}}}{e^{d_1}}) \quad (17)$$

Here, r_{oth} is the distance to the other agent, and the constant d_1 determines how close we wish the agents to be. Thus, the agents will try to maintain a maximum distance of d_1 between one another. If they get farther away than d_1, they will activate the *other* constraint, if they get too close they will activate *obstacles*.

5 Examples

First, we look at the situation depicted in Figure 2, in which two agents approach a target, with one far in front of the other. We will look at the competition parameters for Agent 1 (bottom). In Figure 4A, Agent 2 (top) begins to turn around due to competition between *target* and *other*. Agent 1 continues directly ahead. Both *target* and *other* are active, because both lie in the same direction. Figure 4B shows that Agent 2 has come around to meet Agent 1, and Agent 1's *obstacle* constraint becomes active. It is close to Agent 2, and its *other* constraint begins to give way, but all three constraints are active. In Figure 4C, the agents are near one another, thus Agent 1's *other* constraint is deactivated. Finally, Figure 4D, shows the two agents successfully approaching the target. In this example, the competitive dynamics gives rise to a sequence of behaviors, implementing cooperative navigation.

Next, we look at a new situation. In Figure 5A, the two agents are moving together (Agent 1, right Agent 2, left) as they come upon a wedge-shaped configuration of obstacles, designed to drive them apart. Initially, both *target* and *obstacle* are active. In Figure 5A, they move forward, avoiding obstacles. In Figure 5B, the agents are driven apart. Notice that *target* and *obstacles* are active, while *other* is not yet active; the competitive advantage of *other* is not strong enough for it to become active and suppress *target*. As the two agents round the wedge (Figure 5C), *other* overcomes *target*, and the agents move toward one another. In Figure 5D the agents resume their course toward the target. Again, a behavioral sequence is generated.

6 Concluding remarks

The current investigation has three implications for the dynamic systems approach. First, through competitive dynamics, it is possible to design systems composed of multiple behavioral constraints that give rise to sequences of behavior. Second, competitive interaction among constraints is able to deal with problems that are caused by multiple non-independent contributions to the behavioral dynamics. Finally, the competitive dynamics solution scales nicely to the design of more complex systems. In this paper we used the example of two cooperating robots, but more complex systems are possible.

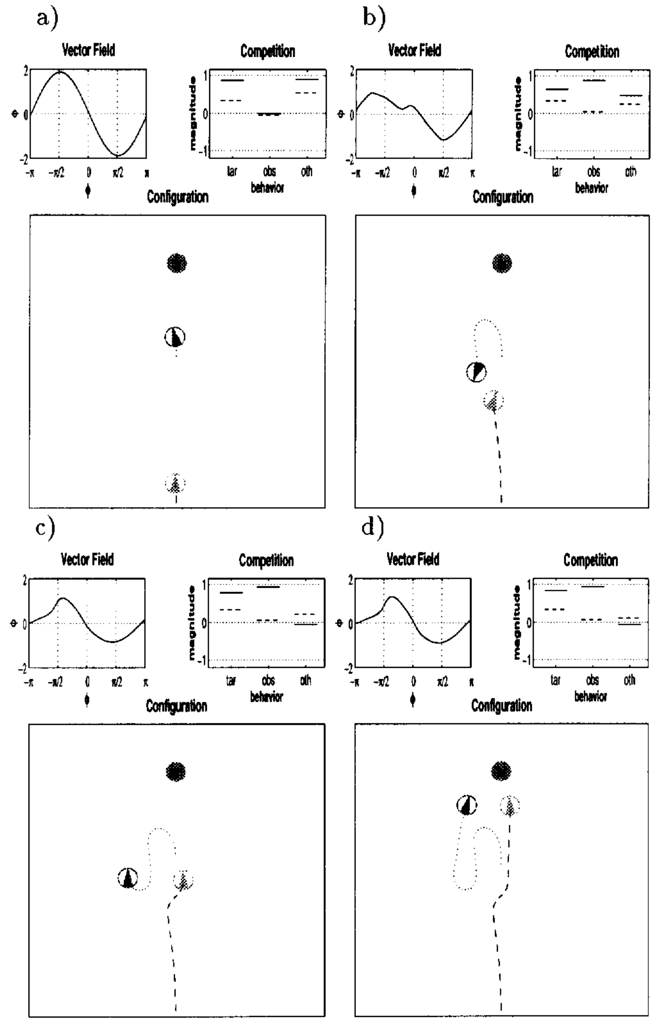

Figure 4: Cooperative navigation (I). Constraint weights (solid) and competitive advantages (broken) are shown in upper left panels for Agent 1 (bottom/right).

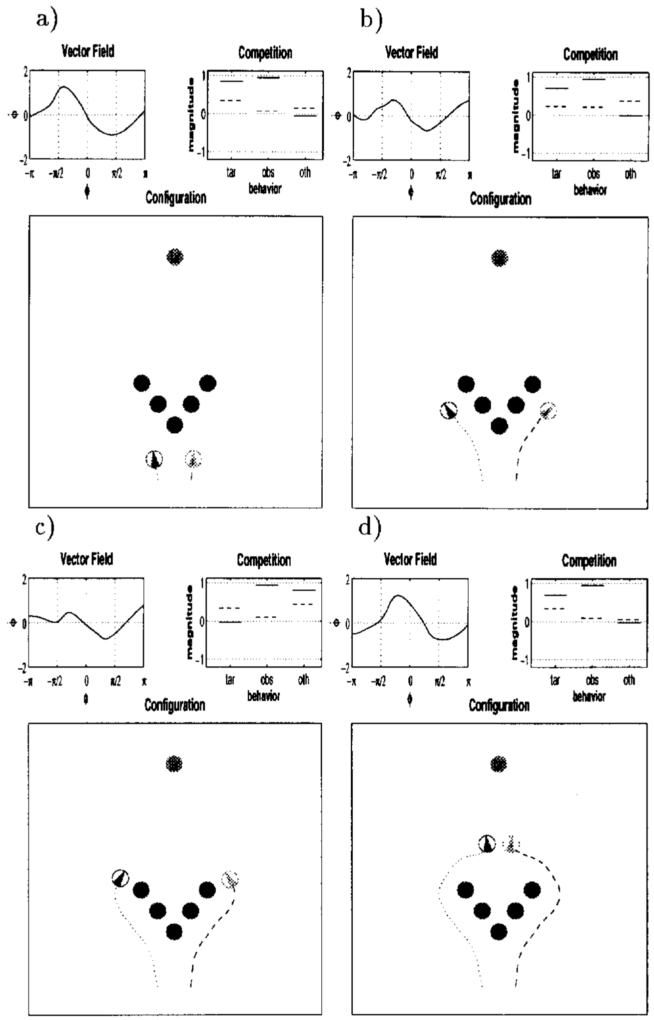

Figure 5: Cooperative navigation (II). Constraint weights (solid) and competitive advantages (broken) are shown in upper left panels for Agent 1 (right).

References

[1] J. T. Schwartz and M. Sharir, "On the "piano movers" problem. i. the case of two-dimensional rigid polygonal body moving amidst polygonal barriers," *Comm. Pure Appl. Math.*, vol. 36, pp. 345 – 398, 1983.

[2] J. C. Latombe, *Robot Motion Planning*. Norwell, MA: Kluwer Academic Publishers, 1991.

[3] T. Lozano-Perez and W. A. Wesley, "An algorithm for planning collision free paths among polyedral obstacles," *Communication of the ACM*, vol. 22, no. 10, 1979.

[4] V. J. Lumelsky and A. A. Stepanov, "Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape," *Algorithmica*, vol. 2, pp. 403–430, 1987.

[5] O. Khatib, "Real-time avoidance for maniopulators and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986. Considered the initial paper on potential fields for robot planning.

[6] D. Koditschek, "Robot planning and control via potential functions," *Robotics Review*, 1992.

[7] V. Braitenberg, *Vehicles. Experiments in Synthetic Psychology.* MIT Press, 1989.

[8] R. Brooks, "Intelligence without reason," Tech. Rep. 1293, Massachussetts Institute of Technology, April 1991.

[9] G. Schöner and M. Dose, "A dynamics systems approach to task level systems integration used to plan and control autonomous vehicle motion," *Robotics and Autonomous Systems*, vol. 10, pp. 253–267, October 1992.

[10] G. Schöner, M. Dose, and C. Engels, "Dynamics of behaviour: theory and applications for autonomous robot architectures," *Robotics and Autonomous Systems*, vol. 16, no. 2–4, pp. 213 – 246, 1996.

[11] H. Neven and G. Schöner, "Dynamics parametrically controlled by image correlations organize robot navigation," *Biological Cybernetics*, in press.

[12] L. Perko, *Differential Equations and Dynamical Systems.* No. ISBN-0387974431 in Texts in Applied Mathematics, Berlin: Springer Verlag, February 1991.