

# Scaling the Dynamic Approach to Path Planning and Control: Competition among Behavioral Constraints

Edward W. Large  
GRASP Laboratory  
University of Pennsylvania

Henrik I. Christensen  
Laboratory of Image Analysis  
Aalborg University

Ruzena Bajcsy  
GRASP Laboratory  
University of Pennsylvania

GRASP Laboratory Technical Report 409

Submitted to *International Journal of Robotics Research*

**Draft:** Please do not quote or distribute

Address correspondence to:

Edward W. Large

GRASP Laboratory

University of Pennsylvania

3401 Walnut Street, Suite 301C

Philadelphia, PA 19104

voice: (215) 898 0343

fax: (215) 573 2048

email: [large@grip.cis.upenn.edu](mailto:large@grip.cis.upenn.edu)

## Abstract

The *dynamic systems* approach to robot planning and control defines a “dynamics” of robot behavior in which task constraints contribute independently to a nonlinear vector field that governs robot actions. We address problems that arise in scaling this approach to handle complex behavioral requirements. We propose a dynamics that operates in the space of task constraints, determining the relative contribution of each constraint to the behavioral dynamics. We find that competition among task constraints is able to deal with problems that arise when combining constraint contributions. Competitive dynamics makes it possible to specify behavioral requirements (i.e. tasks) that are more complex than simple navigation. To demonstrate the utility of this approach, we design a system of two agents to perform a cooperative navigation task. We show how competition among constraints enables agents to execute sequences of behaviors that satisfy task requirements. In addition, we demonstrate the scalability of our competitive dynamics approach to the design of complex robotic planning systems.

# 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. [Schwartz and Sharir, 1983; Latombe, 1991]) model the geometry of the agent and the external environment. The difficulty with this approach is that it is too static. Configuration space representations [Lozano-Perez and Wesley, 1979; Murray *et al.*, 1994] include geometry and kinematics. The difficulty here is that these spaces are extremely complex and so only simple configurations are computationally feasible. Potential field representations [Khatib, 1986; Koditschek, 1992] build upon configuration space representations, defining a state space over which a potential field can be defined. Yet these representations too can be extremely complex.

The above approaches rely upon global representations of the world in which the robot operates. Another possibility is to define a local representation (e.g. [Lumelsky and Stepanov, 1987]) and/or a representation whose dimensions correspond to robot behavior (e.g. [Brooks, 1989]). The so-called *dynamical systems* approach for robot path planning and control uses such a local behavior-based representation [Schöner and Dose, 1992; Schöner *et al.*, 1996]. In this approach a set of behavioral variables defines a state space in which a “dynamics” of robot behavior is described. This approach has the following features.

- 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 repellers of a dynamical system. The contributions are combined into a single vector field by additive composition.
- Planning and control are governed by a dynamical system that generates a time course of the behavioral variables. The dynamics is specified by erecting a vector field that governs the

behavior of the system.

Our work, presented in this paper, has been motivated by this approach because it is suitable for modeling the dynamics of the robot's interaction with its environment in the navigation task. In our view, this approach has several advantages. First, it does not make unreasonable assumptions, or place unreasonable constraints on the environment in which the robot navigates. Although it is a local approach, and therefore is not applicable to optimal path planning (c.f. [Desai *et al.*, 1996]), it is appropriate for planning and control in dynamically changing environments. In addition, the fact that a behavior is generated by a nonlinear dynamical system means that we can make use of properties, such as stability, bifurcation, and hysteresis, that enable planning decisions to be made and carried out in a flexible, yet stable way. Similar modeling principles have been successfully applied to develop theories of biological motion [Schöner and Kelso, 1988]. Most importantly, as we will show, the dynamical systems approach is applicable to the production of behaviors that are more complex than simple navigation, as long as one can express the requisite behavior in terms of constraints in the space of behavioral variables.

In spite of its potential advantages, the generation of complex behaviors by nonlinear dynamical systems poses certain problems. One fundamental difficulty with the simultaneous representation of multiple constraints in a nonlinear vector field concerns the creation of spurious attractors. 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. Spurious attractors may give rise to undesired behaviors such as running into obstacles, or getting stuck in an area and never reaching a target location. In this paper, we investigate situations in which non-independent contributions to the vector field can create spurious attractors and cause related problems. We propose a solution that deals with multiple behavioral requirements using weighting coefficients that determine the relative contribution of different task constraints at any given time. The resultant weighted combination of constraints is similar in some respects to certain connectionist approaches (e.g. [Jacobs *et al.*, 1990; Jordan and Jacobs, 1994]), but it is not learned, rather it is computed dynamically in response to the current environmental situation through a competitive dynamics.

Our competitive dynamics enforces competition among task constraints (e.g. targets, obstacles, other agents, etc.) based upon two factors: the applicability of a particular constraint in the current

situation, which determines its *competitive advantage*, and the degree to which the constraint is consistent or inconsistent with other active contributions to the vector field, captured as *competitive interaction*. These two parameters are bound to the agent’s current situation through functions that are engineered by a designer and reflect the nature of the task. Given appropriately chosen functions that tie these parameters to the environment, we show that this type of competition solves spurious attractor problems for the case of two constraints, (*target* and *obstacles*), and we demonstrate the scalability of this approach by designing a system of three task constraints. The three-constraint system is used to control of a pair of cooperating robots (c.f. [Adams *et al.*, 1995]). Competition among task constraints produces simple and complex sequences of behavior that are generated opportunistically, in response to specific environmental situations. We propose a set of general design principles intended to serve as a guideline for the synthesis of systems with more extensive behavioral repertoires.

This paper is organized as follows. In Section 2 we briefly review the most important concepts of the dynamical systems approach to path planning and control, discussing potential problems related to spurious attractors, and scaling to multiple behavioral requirements. In Section 3 we develop a competitive dynamics solution to the problem of spurious attractors for the case of two task constraints. We propose a general design methodology for engineering such competitive dynamics. We show examples of the resultant system solving situations it could not solve before, and generating simple sequences of behaviors. In Section 4 we apply our design methodology to a system of two cooperative agents, each operating under three task constraints. We show more complex behavioral sequences generated by this system. In a final section, we discuss the implications for scaling of dynamical systems approach to the design of even more complex systems.

## 2 The Dynamical Systems Approach to Planning and Control

In the dynamic approach behavior is described in terms of a set of variables that define behavioral dimensions. For the task of autonomous robot navigation one may represent the behavior of the agent using heading direction,  $\phi$  ( $-\pi \leq \phi \leq \pi$ ), and velocity,  $v$  [Schöner and Dose, 1992]. In this paper, we focus on a single behavioral dimension, heading direction. We assume that velocity is controlled by a dynamics similar to that described by [Neven and Schöner, in press].

Task constraints are expressed as points or parameterized sets of points in the space spanned by

the behavioral variables. For example, in the navigation task, the heading direction  $\psi_{tar}$  represents the direction to the target location, while the direction  $\psi_{obs}$  represents the direction to an obstacle, as shown in Figure 1. Thus, desired behavioral states (such as moving toward a target) and undesired behavioral states (such as moving toward an obstacle) are represented in a way that is invariant to changes in the frame of reference [Schöner *et al.*, 1996].

Figure 1 goes about here

## 2.1 Behavioral Dynamics

The behavior of the agent is modeled as a time course of the behavioral variables generated by a behavioral dynamics that incorporates both planning and control knowledge. For our one-dimensional system, the dynamics take the following form.

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

Task constraints define contributions to the vector field,  $f(\phi)$ , by modeling desired behaviors as attractors (Figure 1A) and to-be-avoided behaviors as repellers (Figure 1B) of the behavioral dynamics. Thus, task constraints affect the behavioral dynamics, they do not specify behavioral patterns directly. Behavioral patterns are generated by the behavioral dynamics.

A desired behavior is modeled as an attractor of the behavioral dynamics (shown in Figure 1A),

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

where  $\phi$  is agent heading direction in world coordinates, and  $\psi_{tar}$  is the direction toward the target location.

A to-be-avoided behavior is specified as a repellor (shown in Figure 1B).

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

The repellor corresponding to an individual obstacle (Figure 1B) is the product of three functions. One function sets up a generic repellor in the direction of the obstacle,

$$R_{obs_i} = \frac{(\phi - \psi_i)}{\Delta\psi_i} e^{1 - \frac{\phi - \psi_i}{\Delta\psi_i}}, \quad (4)$$

a second limits the angular range of the contribution,

$$W_{obs_i} = \frac{1}{2}[\tanh(h_1(\cos(\phi - \psi_i) - \cos(2\Delta\psi_i + \sigma))) + 1], \quad (5)$$

and a third scales the strength of the contribution according to the obstacle’s distance from the agent.

$$D_{obs_i} = e^{-\frac{r_i - R_i - R_{agent}}{d_0}} \quad (6)$$

The parameters to these function are,  $\phi$ , the heading direction of the agent,  $\psi_i$ , the direction to obstacle  $i$ ,  $\Delta\psi_i$ , the angular range subtended by the obstacle,  $R_i$ , the radius of the obstacle, and  $R_{agent}$ , the radius of the agent, and  $\sigma$ , a safety margin. The constant  $d_0$  represents the distance at which the agent begins to take obstacles into account. Obstacles that are very far from the agent do not affect the behavioral dynamics, whereas nearby obstacles affect the dynamics quite strongly. Further details regarding these specific functions can be found in [Schöner and Dose, 1992]. Multiple obstacles are handled by summing the contributions of individual obstacles.

$$F_{obs} = \sum_{i=1}^n F_{obs_i} \quad (7)$$

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

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

Because certain constraints are modeled as repellers, the planning dynamics is augmented by a stochastic term that guarantees escape from unstable fixed points (repellers). This term can also be thought of as modeling presence of noise, for example in determining obstacle position, or in controlling effectors. An important feature of this approach is the concept of asymptotic stability of behavior, brought about by generating behaviors from a dynamics, rather than directly from the task constraints. Not only is the system robust to noise, it requires the presence of noise.

Qualitative change in behavior arises through change in the number, nature, or stability of attractors and repellers. Such changes correspond to bifurcations in the vector field, which are

brought about by movement of the agent through the environment. Note, for example, the parameters to  $F_{obst}$ . As the agent moves, the distance to the obstacle, and the angular range subtended by the obstacle vary. Changes to these parameters cause bifurcations in the vector field that bring about qualitative changes in the agent’s behavior, modeling planning decisions.

## 2.2 Superposition of Task Constraints and Spurious Attractors

In the dynamic approach, avoidance of a single obstacle is modeled by adding a range-limited repeller to the vector field, while avoidance of multiple obstacles is modeled by summing multiple range-limited repeller contributions. This strategy works because linearly dependent contributions lead, through superposition, to averaging among corresponding constraints, while linearly independent contributions allow for the expression of constraints that are incompatible, contradictory, or independently valid. To understand what this means, consider the two situations depicted in Figure 2. In Panel A, the agent faces a pair of obstacles that are positioned too closely together for the agent to pass between them. The constraints represented by the two obstacles lead to a single repeller in the vector field at their average location: behaviorally a single obstacle. In Panel B, the agent again faces two obstacles, but this time they are positioned far enough apart for the agent to pass between. These two constraints are independently valid, and an attractor is formed in the vector field, behaviorally corresponding to steering between the two obstacles.

Figure 2 goes about here

An important restriction on this approach to combining obstacle constraints is that sensed obstacles with a high degree of overlap cannot be allowed to contribute separately to the vector field, because averaging of their contributions can create spurious attractors. Schönner and Dose [Schönner and Dose, 1992] deal with this problem through a competitive interaction among obstacles. Sensed obstacles that overlap are forced to compete in such a way the only one “representative” obstacle is allowed to contribute to the vector field. More recent work has implemented competition among sensed obstacles using a neural field architecture [Amari, 1977], with the general purpose of cleaning up noisy perceptual information so that separate contributions to the behavioral dynamics are guaranteed to have the desired properties [Engels and Schönner, 1995; Schönner *et al.*, 1996]. A second function of the neural field is that it enables the system to store information about its



environment in the form of a cognitive map. As the system explores its environment, it is able to add to its knowledge. Through neural field dynamics, sensed and remembered information are integrated into the vector field so that the system can make use of environmental information even when it is not being directly sensed.

### 2.3 Competition Among Task Constraints

Our implementation of the dynamic approach has revealed that competition among obstacle constraints does not completely solve the spurious attractor problem. Situations can be created in which the combination of the target contribution with multiple obstacle contributions creates spurious attractors. Figure 3 shows two such situations. In Figure 3A, 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 repeller with a shallow slope is created at their average location. If the target is placed behind the obstacles, however, so that its attractor contribution to the vector field “collides” with this repeller, an attractor is created between the two obstacles. This attractor will cause the agent to get stuck at this location.

Figure 3 goes about here

Figure 3B shows another situation in which the agent has moved down a hallway toward a target location and has reached a dead-end; it is thus prevented from making further progress toward the target. Once again, if only the *obstacles* contribution is considered, a shallow repeller exists that would cause the agent to turn around and leave the hallway. However, the repeller is contradicted by the target contribution and the agent is stuck at the dead end.

The reason that spurious attractors are created in these situations is that the relative strength of each contribution ( $F_{tar}$  and  $F_{obs}$ ) to the vector field is determined solely by the fixed time scale of the individual contributions to the planning dynamics. In order to deal with situations such as we have described above, we further modify the strength of each contribution with a specific weight that is assigned to each type of task constraint, *target* and *obstacles*.

$$\dot{\phi} = |w_{tar}|F_{tar} + |w_{obs}|F_{obs} + noise \tag{9}$$

Weights are assigned through a competitive dynamics that determines the strength of each contribution depending upon the current situation.

$$\dot{w}_i = \alpha_i w_i (1 - w_i^2) - \sum_{j \neq i} \gamma_{j,i} w_j^2 w_i + noise \quad (10)$$

The state space of this dynamical system corresponds to the set of task constraints; the first system we will consider is two dimensional, with state vector  $[w_{tar}, w_{obs}]$ . The parameters,  $\alpha_i$  and  $\gamma_{j,i}$ , are referred to as *competitive advantage* and *competitive interaction*, respectively. In our application, competitive advantage,  $\alpha_i$ , describes the degree to which constraint  $i$  is appropriate to the agent's current situation. Competitive interaction,  $\gamma_{j,i}$ , is used to describe the extent to which constraint  $j$  is consistent or inconsistent with constraint  $i$  given the current situation.

The first term of the dynamic equation is a normal form for the Hopf bifurcation (e.g. [Guckenheimer and Holmes, 1983]). If we consider this term alone then  $\alpha_i$  is the only parameter. When  $\alpha_i < 0$ ,  $w_i$  has a stable fixed point at 0; when  $\alpha_i > 0$ ,  $w_i$  has stable fixed points at 1 and -1. The second term specifies competitive interaction from other constraints. The dynamics in the case of multiple competing constraints is more complex, and will be investigated for the cases of two and three interacting constraints by stability analyses, described below.

Equation (10) describes a competitive dynamics similar to that proposed in [Schöner and Dose, 1992] for implementing competition among sensed obstacles. For the case of competition among obstacles, however, the difficulty in applying this competitive scheme was that it meant equating each obstacle with a dimension of the state space. This required determining, in each simulation cycle, a correspondence between currently sensed obstacles and previously sensed obstacles, a computationally difficult task [Schöner and Dose, 1992]. Implementing competition in an Amari field solved this problem, but at the expense of simulating a two dimensional integrodifferential equation, a computationally intensive proposition. Our use of competitive dynamics (i.e. Equation (10)) will not be vulnerable to the correspondence problem, however, because we use competition to determine the weighting of a fixed set of behavioral constraints. Therefore, it is not necessary to resort to more computationally expensive means. In fact, we will see that this approach scales nicely, both in terms of computational and design complexity, to the specification of more complex systems.

In the next section, we use competition to address the issue of spurious attractors in the two constraint case. In this process, we outline a set of design principles that will be applicable to the specification with larger numbers of behavioral requirements. We claim that this strategy of competitive interaction among task constraints, combined with our design methodology, is general enough to support systems in which the the agent has a rich set of behaviors. In the following section, we demonstrate scalability by using this methodology to design an agent with three task constraints.

### 3 Solving the Spurious Attractor Problem for the Two Constraint Case

As introduced above, we will first use competitive dynamics to address the spurious attractor problem for the two behavior case. Our development will proceed in three stages. First, we perform a stability analysis that will tell us how relative values of the parameters  $\alpha_i$  and  $\gamma_{j,i}$  determine the resultant weighting of component behaviors. The second stage consists of a geometrical analysis in which we identify situations where the two constraints, *target* and *obstacles* are incompatible. By “geometric” we refer here to the geometry of the state space of the behavioral dynamics. This leads to the design of functional forms that tie competitive interactions, i.e. the  $\gamma_{j,i}$ , to specific situations. Finally, we perform a behavioral analysis. In this stage we determine which environmental situations call for the activation of which behaviors. This leads to the design of functional forms for the competitive advantage,  $\alpha_i$ , of each constraint.

#### 3.1 Stability Analysis

A linear stability analysis (e.g. [Perko, 1991]) was performed on the system described by Equation (10) for  $i \in 1, 2$ , i.e. the case of two behavioral constraints. The analysis reveals the qualitative behavior of the competitive dynamics by enumerating the set of equilibrium points for the two-dimensional system and classifying each equilibrium point according to its stability, i.e. it determines whether the fixed point is an attractor or repeller of the competitive dynamics. We assumed  $\alpha_i, \gamma_{j,i} > 0$ . Because the stability of each equilibrium point changes depending upon the values of the parameters  $\alpha_i$  and  $\gamma_{j,i}$ , we also computed a set of stability conditions, relative values

of the parameters that determine the conditions under which each fixed point is stable or unstable.

The results of our analysis are shown in Table 1. There are nine equilibrium points, because each non-zero point has both a positive and negative value. The positive and negative values have the same stability conditions, and in addition, the absolute magnitude of each weight is used to determine the contribution of the corresponding behavioral constraint. Thus, due to symmetry, these nine reduce to four unique equilibrium points.

Table 1 goes about here

Each of the four unique equilibrium points corresponds to a different composition of constraints in the vector field that governs the behavioral dynamics. Thus, each leads to qualitatively different behavior for the agent. The first point,  $(0, 0)$  corresponds to both constraints, (*target* and *obstacle*) being effectively turned off. This point is unstable (a repeller of the competitive dynamics) as long as  $\alpha_{tar}, \alpha_{obs} > 0$ .

The point  $(w_{tar}, w_{obs}) = (0, 1)$  corresponds to the activation of *obstacles*, and the deactivation *target*. It is stable as long as  $\gamma_{obs,tar} > \alpha_{tar}$ . In other words, this point is an attractor of the competitive dynamics whenever competitive interaction from *obstacles* to *target* is greater than the competitive advantage of *target*. The resultant behavioral composition is appropriate in situations such as depicted in Figure 2A, in which the additive composition of these two constraints would lead to the creation of a spurious attractor in the vector field.

The point  $(w_{tar}, w_{obs}) = (1, 0)$  corresponds to the activation of *target*, and deactivation *obstacles*. It is stable as long as  $\gamma_{tar,obs} > \alpha_{obs}$ . In other words, this point is an attractor of the competitive dynamics whenever competitive interaction between *target* and *obstacles* is greater than the competitive advantage of *obstacles*. This behavior is appropriate in situations in which there are no obstacles near the agent.

Note that the above stability conditions are not mutually exclusive. When both conditions are satisfied, we have *bistability*, and hysteresis will determine the outcome of the competition. In other words the behavior that is selected by the competition will depend upon the previous history of the system. Although we will not see an example of hysteresis in our two constraint system, this type of solution is appropriate, in general, when the environmental situation is ambiguous.

Finally, the point  $(w_{tar}, w_{obs}) = (A_{tar}, A_{obs})$  corresponds to the activation of both constraints. It is stable whenever  $\alpha_{tar} > \gamma_{obs,tar}$  and  $\alpha_{obs} > \gamma_{tar,obs}$ . This is the so-called “averaging” solution

[Schöner and Dose, 1992]. It is an attractor of the competitive dynamics whenever the competitive advantages of both constraints outweigh the competitive interactions between them. This solution yields a behavior in which both constraints are combined by additive superposition in the vector field.

The “averaging” solution, is given by

$$A_{i,j} = \sqrt{\frac{\alpha_i \alpha_j - \alpha_j \gamma_{j,i}}{\alpha_i \alpha_j - \gamma_{i,j} \gamma_{j,i}}} \quad (11)$$

If there is no competition between constraints,  $\gamma_{i,j} = 0 \forall i, j$ , both constraints are activated at full strength. In this case the resulting behavioral dynamics reduces to that described in [Schöner and Dose, 1992]. If there is some competition, both are still active, but at reduced levels. This behavior is appropriate when the two constraints are both in play and are not in conflict with one another.

In summary, the stability analysis reveals two important facts about the competitive dynamics. First, it tells us that in a system of two behavioral constraints, *target* and *obstacles*, three behaviors are possible: target seeking alone, obstacle avoidance alone, and the combination, target seeking plus obstacle avoidance (arising from the the so-called “averaging” solution). We will design our system so that as the environmental situation changes, parameters to the competitive dynamics will also change, causing bifurcations in the competitive dynamics. These bifurcations allow the system to “decide” which of these three behaviors is appropriate in any given situation. Second, this analysis describes how different values of the competition parameters select categories of behavior. In the next two sections, we complete our design by choosing functions that bind the values of these parameters to specific situations in the environment.

### 3.2 Geometrical Analysis

In this section, we determine 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 repeller collide (see Figure 3), unwanted consequences may result, because the two contributions are 1) non-independent and 2) contradictory. Thus, our analysis depends upon the geometry of the behavioral state space. Specifically, 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.

Our first task is to design functions that identify attractors and repellers for the individual contributions to the behavioral dynamics. For the *target* contribution, we use:

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

This function has two factors. The first calculates the sign of slope of the vector field contribution. This determines whether a fixed point is an attractor (negative slope) or a repeller (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 falls to zero as the magnitude of the contribution grows. The constant  $c_1$  determines the rate of fall off, allowing the specification of a safety margin around the attractors and repellers if necessary.  $P_{tar}$  has a value of one at a repeller, minus one at an attractor, and values approaching zero elsewhere. Thus, it describes the location and stability of the fixed points of the *target* contribution to the behavioral dynamics.

The situation is slightly more complicated for *obstacles*, however. Because individual obstacle contributions are range limited, i.e. have values near zero outside an obstacle's range, Equation (12) may identify these areas as fixed points. Thus, for *obstacles*, we sum the range-limiting functions for the obstacles given in Equation (5) (i.e.  $W_{obs} = \sum_{i=1}^n W_{obs_i}$ ), and use this as a multiplicative factor.

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

As above, this function has a value of one at a repeller, minus one at an attractor, and values approaching zero elsewhere. Thus, it describes the location and stability of the fixed points of the *obstacles* contribution to the behavioral dynamics.

Next we design the competitive interaction function itself. We use  $P_{tar}$  and  $P_{obs}$  to construct a function describes the competitive interaction between *obstacles* and *target* as:

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

The graph of Equation (14) is shown in Figure 4, for the situation depicted in Figure 3A. Note that it is strongly peaked at the point of attractor-repeller collision. The constant  $c_2$  determines the

rate of drop off around the collision. Note also that it also provides a certain level of background competition that we will later use to help determine the appropriate level of competitive advantage,  $\alpha_{tar}$ , for *target*.

Figure 4 goes about here

Finally, we choose the competitive interaction between *target* and *obstacles*. For the current navigation task, it is never appropriate for *target* to deactivate the *obstacles* constraint. Thus, we simply choose a low constant value, such as  $\gamma_{tar,obs} = 0.05$ , allowing this constraint to be activated whenever the agent approaches an obstacle.

### 3.3 Behavioral Analysis

In the previous section, we designed functions that capture situations in which *target* and *obstacle* should compete, i.e. when attractor and repeller contribution would “collide”. From the stability analysis we know how to pick relative values of  $\alpha_i$  and  $\gamma_{j,i}$  in such a way that we can choose the type of constraint composition that we would like. In this section, we complete the design, choosing values for the competitive advantages so that, in situations where the two behaviors compete (near the peak of Equation (14)), we can determine the outcome of the competition.

First, we note that the *target* constraint should be turned on whenever possible. For example, we can choose a constant value of  $\alpha_{tar} = a_{tar}$  such that whenever *obstacles* actively competes with *target*,  $\gamma_{obs,tar} > a_{tar}$  and *target* will lose the competition. On the other hand, as long as  $a_{tar}$  exceeds the background level of competition created by Equation (14) (see Figure 4), *target* will be activated.

Next, we must decide how to set the 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. We have already encountered a function that grows exponentially fast as we approach an obstacle,  $D_{obs_i}$  (Equation (6)), which is a component of the function  $F_{obs_i}$ . In order to count the number of obstacles around the agent, we simply sum the  $D_{obs_i}$ . We then limit the maximum value of the  $\alpha_{obs}$ , resulting in the following function for competitive advantage.

$$\alpha_{obs} = \tanh \sum_{i=1}^n D_{obs_i} \quad (15)$$

This completes our design.

### 3.4 Examples

In section 2.3 we saw two situations in which spurious attractors were created by additively combining non-independent, contradictory contributions to the vector field. In this section we demonstrate how competition deals with these situations using output from a graphic simulator. The same simulator serves as a dynamic planner/controller for a pair of mobile robot platforms in our laboratory.

First, in Figure 3A, a spurious attractor arose when a shallow repellor, created by two obstacles, was combined with an attractor from the *target* contribution. Figure 5 shows four snapshots from an episode in which the agent, using competition among behavioral constraints, successfully navigates this situation. Figure 5A shows the agent *en route* toward the target. It is far enough from the obstacles that it has not yet seen them, thus  $w_{tar} = 1$ ,  $w_{obs} = 0$ , and the vector field consists only of the attractor contribution. Figure 5B shows the situation shortly after the agent has detected both obstacles. The reader should compare this situation with that of Figure 3A. Unlike in Figure 3A, however, in Figure 5B the vector field consists solely of the *obstacles* contribution. This is because competitive interaction increased, as shown in Figure 4,  $\gamma_{obs,tar} > \alpha_{tar}$ , and *target* is deactivated, while  $\alpha_{obs} > \gamma_{tar,obs}$ , and *obstacles* is activated. Figure 5C shows the situation a few time steps later, when the agent has turned away from the target. Competitive interaction has dropped, so that  $\alpha_{tar} > \gamma_{obs,tar}$  and *target* is turned on, while it is still the case that  $\alpha_{obs} > \gamma_{tar,obs}$  so *obstacles* is turned on as well. This is the “averaging” solution, resulting in a composite behavior that combines the two constraints. Finally, Figure 5D shows the agent as it rounds the leftmost obstacle, successfully approaching the target. It is the combination of task constraints that causes the agent to round the obstacle, rather than to simply steer away from the obstacle. Note also that the agent has produced a sequence of behaviors: a seek behavior, followed by an avoid behavior, followed by a composite behavior. This simple sequence demonstrates each unique behavior that arises from the competitive dynamics for the case of two constraints.

Figure 5 goes about here



Next, we turn to a more complex situation in which the agent is trapped in an enclosure that is preventing it from reaching the target location. In this situation, depicted in Figure 3B, *target* will be deactivated by competition (see Figure 4) only so long as the agent is pointed more-or-less directly toward the target location. When the agent turns away, the competitive interaction ( $\gamma_{obs,tar}$ ) will drop, and the influence of target will once again cause the agent to turn toward the target. The problem here is not simply that *target* and *obstacles* are contradictory. Rather, in this situation, *target* is not a useful constraint. The agent is trapped in an enclosure from which it must escape before target seeking behavior becomes useful. In other words, the agent must establish the intermediate goal of escaping from the enclosure.

This is the type of situation in which it is appropriate to temporarily disable *target*, until the agent has escaped from the enclosure. We can characterize this general type of situation heuristically by observing that the agent is 1) surrounded by obstacles, and 2) has no line-of-sight path to the goal. Thus, we rewrite the expression for competitive advantage of the target as:

$$\alpha_{tar} = a_{tar} - (1 - V_{tar})a_{tar}\alpha_{obs} \quad (16)$$

Here,  $a_{tar}$  is the competitive advantage for *target*, as described above.  $\alpha_{obs}$  is the competitive advantage of *obstacles*, which increases as obstacles get close and/or increase in number, and  $V_{tar}$  takes on a value of 0 when there is no line-of-sight path to the target location:

$$V_{tar} = \begin{cases} 0 & \text{if obstacle between agent and goal} \\ 1 & \text{otherwise} \end{cases} \quad (17)$$

Thus, the second term of Equation (16) implements a heuristic “enclosure detector”, and  $\alpha_{tar} \cong 0$  when the agent is trapped. This is an example of a situation in which the behavioral situation itself, rather than the contradictory nature of behavioral constraints, temporarily rules out a particular behavioral contribution. For the examples below we choose  $a_{tar} = 0.4$ .

Figure 6 goes about here

Figure 6 shows an example of the agent successfully negotiating the hallway trap using the competitive advantage described by Equation (16). In Figure 6A it travels to the end of the hallway. Both the constraints are active, because the agent is avoiding the walls, thus the contribution of

*obstacles* forms an attractor dead ahead, consistent with the direction to the target. In Figure 6B the agent has encountered the dead-end and has begun to turn away. If the collision of an attractor and repeller were the only factor in deactivating *target* the agent would quickly turn back toward the target. However, because  $\alpha_{tar}$  falls below the background level of competition *target* is turned off regardless of active competition from *obstacles*. Next, Figure 6C shows an interesting, and less obvious, case in which an attractor and a repeller collide: when the attractor is supplied by *obstacles* and the repeller is contributed by *target*. Here the agent must move directly away from the target in order to escape from an enclosure, and make further progress toward the target. Note that  $\alpha_{tar} > 0$  (the agent no longer senses the obstacles between itself and the target) yet  $w_{tar} = 0$ ; this is again due to increased competition from the collision of an attractor with a repeller. Finally, Figure 6D shows the agent successfully making its way out of the enclosure and toward the target. Once again, we have seen a behavioral sequence arise due to competition among behavioral constraints. We have also seen a situation in which the behavioral situation itself, rather than the contradictory nature of task constraints, temporarily rules out a particular contribution. We will encounter more complex sequences in the next section.

In summary, we have described a method of weighting behaviors that precludes the creation of spurious attractors in the vector field through competition amongst individual contributions. The basic idea is that the competition equations detect situations in which non-independent contributions to the vector field are contradictory, and sets the parameters of the competition in such a way that one of the contributions is turned off. We have shown that this method works well in the case of the simple two-constraint system. In addition, in the process of constructing the above system, we outlined a design methodology: stability analysis, geometrical analysis, behavioral analysis. In the next section, we design a system of three task constraints to demonstrate how this approach scales when system requirements are more complex.

## 4 A Case of Three Constraints

In this section, we consider the case of a more elaborate system. We have two agents, and we define an extension of the navigation task as follows. First, both agents must obey the same constraints as in the above system, *target*, and *obstacles*, i.e. they must perform the navigation task. Second, we impose the additional constraint that the two agents must remain 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 requirements.

We begin by making some simplifying assumptions. First, both agents' behavior is governed by the same behavioral dynamics, with the same parameter values. Thus, the behavior of each individual agent is described by identical equations, but generated independently, by the time course of its own behavioral variables. Second, target seeking and obstacle avoidance are to operate as in single agent case, described above. Third, if the agents come too close to one another, they are to avoid collision in the same way as they would avoid stationary obstacles.

Next, we consider the additional constraint of staying near the other agent. Similarly to target seeking, we model this constraint as a global attractor.

$$F_{oth} = -a \sin(\psi_{oth}) \quad (18)$$

The contribution of *other* is weighted and added to the composite vector field:

$$\dot{\phi} = |w_{tar}|F_{tar} + |w_{obs}|F_{obs} + |w_{oth}|F_{oth} + noise \quad (19)$$

The potential problem with combining these three constraints lies in the composition of *target* and *other*, as depicted in Figure 7. In Figures 7A and B, we see a situation in which two agents are headed toward the target, yet one is considerably ahead of the other. In Figure 7A we look at the situation from the point of view of Agent 2 (bottom). The composition of the *target* and *other* contributions sum in such a way the the agent is to move straight ahead. This is acceptable for Agent 2, since both the target and the other agent are lie in the same direction. In Figure 7B we see the situation from the point of view of Agent 1 (middle). The target and the other agent lie in opposite directions, and the composition of these two contribution cancel one another entirely. This is clearly not acceptable. A different situation is depicted in 7C, shown from the point of view of Agent 2 (bottom). The target is to the right, and the other agent is to the left. The two contributions sum such that a single attractor lies in their average direction. This situation may also be unacceptable. Thus, the problem with the composition of *target* and *other*, is that summing these non-independent contributions averages the corresponding constraints. In some cases (Figure 7A) this yields appropriate behavior, in other cases (Figures 7B and C) is does not.

Figure 7 goes about here

Our task will be to design the competitive dynamics so that both agents will behave in a sensible manner when these three constraints are combined. We proceed according to the design methodology outlined in Section 3. First, we perform a stability analysis to determine the equilibrium points of the competitive dynamics and their associated stability conditions. Second we decide when the behaviors are consistent/inconsistent to determine appropriate competitive interaction functions. Finally, a behavioral analysis determines which constraints are appropriate in which situations, yielding the competitive advantages.

#### 4.1 Stability Analysis

As in the case of two constraints, we performed a stability analysis to determine the unique equilibrium points of Equation (10) for three behaviors. As above, we assumed  $\alpha_i, \gamma_{j,i} > 0$ . The results of the analysis are shown in Table 2, revealing eight unique equilibrium points, seven of which are stable for some parameter regime. The stability analysis also reveals distinct *classes* of solutions. Thus, rather than describe each behavior individually, we describe only each class of stable fixed points. This will make the job of understanding the competitive dynamics easier. It will also illustrate important features regarding the scalability of this approach.

Table 2 goes about here

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 is active, and simultaneously inhibits all other behaviors.

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  $k$ . The former condition is equivalent to the condition of the averaging solution for the case of two behaviors. Furthermore, the averaging solution itself is the same as it would be in the two-constraint case, i.e. it is given by Equation (11).

Note that, similarly to the two-dimensional case, the above stability conditions are not all mutually exclusive. In cases of bistability, equilibrium points will be determined by hysteresis, thus

the resultant behavior of the agent will be determined by its past history. As we shall see below, this provides a type of behavioral stability that is useful in ambiguous situations.

The final class of solutions consists of a “3-constraint averaging” solution, where all three contributions are active. This point is stable so 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, it is not particularly informative, so it is not included here.

This analysis points up some interesting properties of the competitive dynamics that have implications for scaling this competitive strategy to systems composed of larger numbers of constraints. First, note the stability conditions for each class of behaviors. Summarized, these conditions tell us that a constraint is deactivated when it is inconsistent with any single active constraint; conversely, it is activated only when it is consistent with all other active constraints. Thus, 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, it is not necessary to consider more complex interactions. This is not obvious simply by inspection of Equation (10), but it is revealed by the stability analysis.

Second, we can count the number of unique behaviors that arise in a  $n$ -constraint system. It is simply the number of ways to choose one active behavior, plus the number of ways to choose two active behaviors, and so on. In other words, the number of behaviors generated in such a system is:

$$N = \binom{n}{1} + \dots + \binom{n}{n} = \sum_{i=1}^n \binom{n}{i} = 2^n - 1. \quad (20)$$

This count,  $N$ , includes only qualitatively different compositions, it does not consider the continuously many graded compositions that arise from the “averaging” solutions as unique. Finally, consider that the straightforward implementation of the dynamics, using Euler integration of Equation (10), has time complexity  $O(n^2)$  for each time-step. Thus, although implementation is not cheap, it is certainly tractable, and it is also vectorizable.

In summary, the stability analysis has revealed important facts about the scalability of this approach. In an  $n$ -constraint system, competition provides  $2^n - 1$  unique behaviors, while simulation of the system has time complexity  $O(n^2)$  per Euler time step. In addition, designing the system requires at most  $n^2$  design decisions:  $n^2 - n$  competitive interactions, plus  $n$  competitive

advantages. Furthermore, given an existing  $(n - 1)$ -constraint system, one adds the  $n^{\text{th}}$  constraint simply by considering interactions with each of the existing  $n - 1$  constraints. The existing system will function as previously designed without unwanted interactions caused by the introduction of the new constraint. Thus, the stability analysis has also revealed how to design the competition parameters for our three-constraint system: we need only specify the competitive advantage for the new constraint, *other*, and the competitive interactions between the new constraint and the existing constraints, *target* and *obstacles*.

## 4.2 Geometrical Analysis

Next we use the results of the stability analysis to design competitive interactions. Our first goal is to build upon the previous design. We wish to have each agent behave as in the previous system with respect to *target* and *obstacles*. As discovered above, we can accomplish this goal by simply leaving the competitive advantages and competitive interactions for these two constraints the same as in the previous design. Our new constraint will not interfere with the existing system of constraints. Therefore, we simply design the competitive interaction between our new constraint and the two existing constraints.

Next we add the new constraint, that of staying near the other agent. As described above, this behavior is modeled as a global attractor centered in the direction of other agent. The primary difficulty lies in the composition of *target* and *other* as shown in Figure 7. Because both constraints are modeled as global attractors, their respective contributions will always be non-independent, thus simply summing the contributions to the vector field will cause averaging between the corresponding constraints. In most situations the resulting behavior will not be appropriate, and we will want to enforce strong competition between the *target* and *other* so that the agent must decide to go in one direction or the other. However, there will also be some situations when moving in the average direction does represent 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. In general, we can say that when the two goals lie in approximately the same direction, we allow the averaging solution. When they are in very different directions, we must force a decision. We can accomplish this type of competitive interaction using the following function.

$$\gamma_{tar,oth} = \gamma_{oth,tar} = b_1(\tanh(-b_2 \cos(\psi_{tar} - \psi_{oth}) + b_3) + 1) \quad (21)$$

Equation (21) is graphed in Figure 8. Note that 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  $b_2$ . The parameter  $b_1$  sets the overall level of competitive interaction.

Figure 8 goes about here

### 4.3 Behavioral Analysis

Next we need to decide upon 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:

$$\alpha_{oth} = a_{oth} \tanh \frac{e^{r_{oth}}}{e^{d_1}} - (1 - V_{oth})a_{oth}\alpha_{obs} \quad (22)$$

Here,  $r_{oth}$  is the distance to the other agent, and the constant  $d_1$  determines how close we wish the agents to be. For simplicity, we choose  $d_1 = d_0 + 1$ , where  $d_0$  is the distance at which the agent begins to consider one another as obstacles to be avoided (see Equation (6)). 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 closer than  $d_0$ , they will activate *obstacles*. The constant,  $a_{oth}$ , determines the maximum level of advantage for *other*. The second term of the equation enables implements an “enclosure detector” similar to that defined for *target* in Equation (16).

Figure 9 goes about here

### 4.4 Examples

In this section, we examine two examples of the cooperative behavior. First, we look at the situation depicted in Figure 7A and B, in which two agents approach a target, but with one far in front of the other. We look at the situation both from the perspective of Agent 1 (above in the configuration;

top 3 panels) and Agent 2 (below in the configuration; bottom 3 panels). Notice that initially, Agent 1 must decide whether to move toward the target or toward Agent 2. In Figure 10A Agent 1 begins to come about because *other* wins the competition with *target*. Below, Agent 2's *other* and *target* constraints are both activated, because these two constraints are consistent with one another, according to Equation (21), and thus they do not compete. In Figure 10B we see that the agents have moved close to one another. Agent 1 (top) has deactivated its *other* constraint, while *obstacles* has just become activated, and the agent begins to veer away from collision with Agent 2. Agent 2 (bottom) has activated all constraints, because there is some competition among all three, yet all competitive advantages are stronger than all competitive interactions. Finally, in Frame C, Agent 1 has reversed course yet again, this time accompanying Agent 2 to the target. The two agents are close, so *other* is deactivated for both, whereas *obstacles* is still active, as they avoid collision with one another *en route* to the target.

This episode illustrates a different behavioral sequence for each agent, but generated from identical dynamical equations. Different behaviors arise because the individual agents face different situations from moment to moment, and these situations are reflected in the parameters of the competition dynamics for each agent. The agents navigate cooperatively because the functions that tie the competition parameters to specific situations select behaviors that satisfy an appropriate set of behavioral requirements. Note that these two cooperative sequences displayed an example of each class of behavior that was identified in the stability analysis of Section 4.1.

Figure 10 goes about here

Next, we look at a new situation. In Figure 11A, 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. We display the behavioral dynamics and competition for Agent 1 only, due to the symmetry of the situation. In Figure 11A, as they move forward both *target* and *obstacles* are active because they take account of one another as obstacles; *other* is not active, because the agents are near one another. In Figure 11B, the agents are driven apart. Notice that the agents are quite far apart and the competitive advantage of *other* is quite strong, stronger, in fact, than that of *target*. Yet *target* is active, while *other* is not. This is a hysteresis effect. Both competitive advantages are less than the competitive interaction between these two constraints, and *target* wins the competition



because it was previously active. In this situation, the previous history of the system determines its behavior. As the two agents round the wedge (Figure 11C), the competitive advantage of *other* increases above the level of competitive interaction from target (i.e.  $\alpha_{oth} > \gamma_{tar,oth}$ ), thus it acquires enough strength to deactivate *target*, and the agents move toward one another. Figure 11D, shows the situation after the agents have come together resumed their original course.

Figure 11 goes about here

This sequence is not as complex as the previous sequence, displaying only alternation of two-constraint behaviors. Nevertheless, it displays some interesting properties of our competitive dynamics solution. First, it displays the flexibility of this approach to cooperative navigation. Initially, the agents navigate toward the target location together, but the unknown environment forces them apart. The agents are able to make decisions such that they flexibly respond to the demands of a new and unforeseen situation, coming together once again when the environment allows. Second, in this situation hysteresis allows for a special kind of behavioral stability. In Panel B the two agents were far from one another, but there is was no line-of-sight path toward one another. This is an ambiguous situation: it is not clear whether they should continue toward the target, or give up that goal and try to find one another. In this situation, the agents continue to do what they had been doing previously, moving toward the target. This behavior was due to hysteresis, a simple kind of memory that determines system performance according to its past history.

In summary, these two example have demonstrated how flexibility, arising from bifurcations in the competitive dynamics, allows a system to generate simple and complex sequences of behaviors that enable a pair of autonomous agents to satisfy the behavioral requirements of a cooperative navigation task. Behavioral complexity arises from two sources, the number of individual behaviors available for each agent to satisfy requirements, and the existence of two agents working together, generating different sequences, to satisfy the constraints. Thus, these examples serve to demonstrate that the addition of a competitive dynamics, operating in the space of task constraints, allows us to scale the dynamic systems approach to planning and control beyond simple navigation to cooperative navigation. Furthermore, our analysis of the competitive dynamics indicates that even more elaborate systems are possible.

## 5 Discussion

We set out to examine the issues of representation for agent/environment interaction in navigation-like tasks. We have taken care to separate non-physical, context dependent, socio-cultural constraints that determine appropriate planning and control of robot actions from physical and geometric models of the agent and its environment. Nevertheless, we have adopted a physics-based model, i.e. the so-called *dynamical systems* approach [Schöner and Dose, 1992; Schöner *et al.*, 1996], that has been successfully employed in modeling behavior in biological systems. 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 and repellers of the behavioral dynamics. Attractors and repellers are combined additively into a vector field that governs the agent’s behavior. Our main concern has been to investigate how this approach scales to the design of systems when behavioral requirements extend beyond simple navigation. The problem we encountered was that simple additive composition does not scale nicely to handle complex behavioral requirements. Our contribution has been to show how combinations of constraints can be dealt with by adding a second “layer” of dynamics that is capable of managing task complexity at the behavioral level. The state space for this dynamic layer is the space of task constraints.

The current investigation has three major implications for the dynamic systems approach to planning and control. First, competitive interaction among task constraints is able to deal with problems, such as spurious attractors and constraint averaging, that arise when non-independent contributions to the vector field dynamics are combined by additive composition. Our competitive dynamics enforces competition among task constraints (e.g. targets, obstacles, other agents, etc.) when their respective vector field contributions are inconsistent with one another. The winners of the competition are determined based upon which constraints are most applicable in the current situation. Thus, *competitive interaction* is determined by functions designed to detect when individual contributions are inconsistent, while *competitive advantage* is tied to the environment through functions that implement heuristic judgments about when particular constraints are more or less critical.

Second, the competitive dynamics makes it possible to specify behavioral requirements (i.e. tasks) that are more complex than simple navigation. This ability arises from the ability to de-

termine which constraints should contribute to the behavioral dynamics, i.e. which behavior is appropriate, in any given situation. We defined a “behavior” as a qualitatively unique combination of task constraints that defines a particular set of contributions to the behavioral dynamics. Each combination of task constraints arises as an asymptotically stable fixed point of the competitive dynamics, providing a number of interesting properties. First, each behavior is stable in the sense that it is robust to the presence of noise in the system. This property arises from the stability of the fixed points that generate the behaviors. Second, each behavior is stable in the sense that it is robust to ambiguity in the environment. This property arises due to hysteresis – when more than one fixed point is stable, the past history of the system determines performance. Third, the agent is able to flexibly determine which behavior is appropriate at any given time. This property arises due to bifurcations in the competitive dynamics: As new situations arise, parameters change, old fixed points disappear, and new fixed points appear.

The latter point implies that in attempting to satisfy a complex set of behavioral requirements, the agent will execute a sequence of behaviors. Here the sequences are not explicitly programmed responses to stereotypical situations, rather they arise as the competitive dynamics chooses among different behaviors. The decision to execute a new behavior is modeled as a bifurcation in the competitive dynamics, that arises as the competition parameters adapt to the surroundings; thus sequences are generated opportunistically. It is also possible to explicitly program behavioral sequences using dynamics similar to that which we have described here (c.f. [Steinhage and Schöner, 1996]).

Finally, the competitive dynamics solution scales nicely to the design of complex systems. We have shown that an  $n$  constraint system gives rise to  $2^n - 1$  unique behaviors. Also, the stability analysis revealed that in order to design such a system, the designer needs make  $n^2$  design decisions, designing competitive advantage functions for each individual behavior, and designing competitive interaction functions between each pair of behaviors. This also implies a very nice type of scalability: given an existing system, one can add behavioral requirements without disturbing the operation of the previously designed system. The designer must consider how each new constraint will interact with each of the existing constraints, but need not reconsider the design of the entire system. In this paper, a relatively simple system of three behavioral requirements allowed us to design a cooperative navigation system, one in which two robots navigate independently, yet cooperatively,

through an environment. In ongoing research we are designing more elaborate systems with larger numbers of task constraints for each agent.

We have shown that, for a single robot or a pair of cooperative robots, decision making and planning can be modeled almost entirely using continuous nonlinear differential equations. The theoretical relationship between discrete automata (traditionally used for modeling planning and decision making) and dynamical systems (traditionally used for control) has been studied by others (e.g. [Brockett, 1994]). Here we have investigated the possibility of developing a design methodology for building robotic planning systems using dynamical systems in a way that scales to the modeling of complex systems of behavioral requirements. In this way we hope to combine the advantages of the dynamical system approach (stability, flexibility, robustness, etc.), with the ability to plan and carry out complex sequences of behavior to achieve well-defined goals. We have seen that the competitive dynamics approach to managing task complexity offers a number of advantages in this regard, including scalability. The primary input from the designer is in setting the priorities between competing behaviors. This should not be too surprising considering, that these priorities depend on the situation, context, social order, and so forth. In other words, not only those aspects of control that are physical/geometric are represented in a continuous fashion, but even the switching between different control strategies, which comes from the difference in the prioritization schemas that are less physical and more social and/or cultural, can be captured within the framework of continuous differential equations.

## References

- [Adams *et al.*, 1995] J. Adams, R. Bajcsy, J. Košecká, V. Kumar, R. Mandelbam, M. Mintz, R. Paul, C.-C. Wang, Y. Yamamoto, and X. Yun. Cooperative material handling by human and robotic agents: Module development and system synthesis. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Aug. 1995.
- [Amari, 1977] A. Amari. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27:77–87, 1977.
- [Brockett, 1994] R. W. Brockett. Dynamical systems and their associated automata. In *Systems and Networks: Mathematical Theory and Applications*, volume 77, pages 49–69. Akademi-Verlag,

Berlin, 1994.

- [Brooks, 1989] R. Brooks. A robot that walks: Emergent behaviours from a carefully evolved network. *Neural Computation*, 1(2):253–262, Summer 1989.
- [Desai *et al.*, 1996] J. Desai, C. Wang, M. Zefran, and V. Kumar. Motion planning for multiple mobile manipulators. In *Proceedings of the International Conference on Robotics and Automation*, 1996.
- [Engels and Schöner, 1995] C. Engels and G. Schöner. Dynamic fields endow behavior-based robots with representations. *Robotics and Autonomous Systems*, 14:55–77, 1995.
- [Guckenheimer and Holmes, 1983] J. Guckenheimer and P. Holmes. *Nonlinear Oscillations, Dynamical Systems and Bifurcations of Vector Fields*, volume 42 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1983.
- [Jacobs *et al.*, 1990] R. A. Jacobs, M. I. Jordan, and A. Barto. Task decomposition through competition in a modular architecture. *find reference*, 1990.
- [Jordan and Jacobs, 1994] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6:181–214, 1994.
- [Khatib, 1986] O. Khatib. Real-time avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98, 1986. Considered the initial paper on potential fields for robot planning.
- [Koditschek, 1992] D. Koditschek. Robot planning and control via potential functions. *Robotics Review*, 1992.
- [Latombe, 1991] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Norwell, MA, 1991.
- [Lozano-Perez and Wesley, 1979] T. Lozano-Perez and W. A. Wesley. An algorithm for planning collision free paths among polyedral obstacles. *Communication of the ACM*, 22(10), 1979.

- [Lumelsky and Stepanov, 1987] V. J. Lumelsky and A. A. Stepanov. Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica*, 2:403–430, 1987.
- [Murray *et al.*, 1994] R. M. Murray, Zexiang Li, and S. S. Sastry. *Robotic Manipulation*. CRC Press, 1994.
- [Neven and Schöner, in press] H. Neven and G. Schöner. Dynamics parametrically controlled by image correlations organize robot navigation. *Biological Cybernetics*, in press.
- [Perko, 1991] L. Perko. *Differential Equations and Dynamical Systems*. Number ISBN-0387974431 in Texts in Applied Mathematics. Springer Verlag, Berlin, February 1991.
- [Schöner and Dose, 1992] 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*, 10:253–267, October 1992.
- [Schöner and Kelso, 1988] G. Schöner and J.A.S Kelso. Dynamic pattern generation in behavioral and neural systems. *Science*, (239):1513–1520, 1988.
- [Schöner *et al.*, 1996] G. Schöner, M. Dose, and C. Engels. Dynamics of behaviour: theory and applications for autonomous robot architectures. *Robotics and Autonomous Systems*, 16(2–4):213 – 246, 1996.
- [Schwartz and Sharir, 1983] 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.*, 36:345 – 398, 1983.
- [Steinhage and Schöner, 1996] A. Steinhage and G. Schöner. Dynamic approach to action selection and sequence generation: An example from navigation. *Manuscript in preparation*, 1996.

## Figure Captions

**Figure 1.** Task Constraints and Behavioral Dynamics. (A) An agent, a target, and the corresponding vector field. The *target* constraint is expressed as a heading direction (zero corresponds to the current heading of the agent). The desired behavior of heading toward the target is expressed as an attractor (negative slope) in the vector field that governs agent heading direction. (B) An agent, an obstacle, and the corresponding vector field. The *obstacle* constraint is also expressed as a heading direction. However, the undesired behavior of heading toward the obstacle is expressed as an repellor (positive slope) in the vector field. (C) A more complex configuration. The *target* and *obstacle* constraints are combined into a single vector field additively. The attractor corresponds to steering around the obstacle *en route* to the target location.

**Figure 2.** Dependent and Independent Constraints. (A) Two obstacles configured so that there is not enough space between them for the agent to pass through. These constraints are dependent, and superposition of their contributions to the vector field creates a repellor at their average heading direction, effectively modeling a single obstacle. (B) Two obstacles configured so that there *is* enough space for the agent to pass between them. These constraints are independent, and an attractor is formed in their average direction, allowing the agent to steer between them.

**Figure 3.** Spurious Attractors. (A) Two obstacles dead ahead provide almost, but not quite enough space for the agent to pass between them. The *obstacles* contribution to the dynamics reveals a shallow attractor. Yet when the target also lies straight ahead, its attractor contribution, combined additively with the repellor, creates a spurious attractor in the composite vector field. This will cause the agent to get stuck at this location. (B) A hallway trap. Once again, the *obstacles* constraint creates a shallow repellor. However, because the target lies directly beyond, adding its contribution creates a spurious attractor in the composite vector field. Once again the agent is stuck.

**Figure 4.** Competitive Interaction between *obstacles* and *target* for the spurious attractor example of Figure 3A.

**Figure 5.** Competition avoids the creation of a spurious attractor. Each panel shows the current configuration (bottom), and the corresponding vector field (top left). Heading direction is plotted so

that  $\phi = 0$  corresponds to the agent’s current heading direction. The current competitive situation is also shown (top right): Competitive advantage,  $\alpha_i$  (dashed lines), and the current weighting,  $w_i$  (solid lines), are shown for each constraint. Competitive interaction is not shown. (A) The agent approaches two obstacles that it cannot pass between. (B) This situation could create a spurious attractor (compare with Figure 3A), but it does not because competition deactivates *target*; competitive interaction for this situation is shown in Figure 4. (C) Once the agent has turned away from the obstacles, *target* is reactivated. (D) The agent rounds the leftmost obstacle, steering toward the target location.

**Figure 6.** The agent successfully negotiating the hallway trap. Each panel shows the current configuration (bottom), and the corresponding vector field (top left). Heading direction is plotted so that  $\phi = 0$  corresponds to the agent’s current heading direction. The current competitive situation is also shown (top right): Competitive advantage,  $\alpha_i$  (dashed lines), and the current weighting,  $w_i$  (solid lines), are shown for each constraint. Competitive interaction is not shown. (A) The agent moves down the hallway toward the target. (B) The agent faces the spurious attractor situation shown on Figure 3B. Competitive advantage,  $\alpha_{tar}$ , drops below the background level of competition created by Equation (14), allowing the agent to turn away from the dead-end. (C) The agent moves out of the hallway, due to active competition between *obstacles* and *target*. (D) The agent successfully leaves the trap.

**Figure 7.** Non-Independent Constraints. When constraints are dependent, additive composition results in constraint averaging. (A) From the point of view of Agent 2 (bottom), the target and the other agent are in the same direction, so constraint averaging is acceptable. (B) From the point of view of Agent 1 (middle), the target and the other agent are in opposite directions, so constraint averaging is clearly unacceptable. (C) From the point of view of Agent 2 (bottom), the target and the other agent lie in different directions. Averaging may be unacceptable, although this judgment depends somewhat on the task specification.

**Figure 8.** Competitive Interaction between *Target* and *Other*,  $\gamma_{tar,oth} = \gamma_{oth,tar}$ , for  $b_1 = 0.25$ ,  $b_2 = 4.0$ ,  $b_3 = 3.0$ . Competition is maximum except when the target and the other agent lie in approximately the same direction, i.e. when  $\psi_{tar} - \psi_{oth}$  is near zero.

**Figure 9.** Competitive Advantage of *Other*, for  $a_{oth} = 0.6$ ,  $d_1 = 3$ , and  $V_{oth} = 1$ . As distance



to the other agent increases beyond  $d_1$ , competitive advantage,  $\alpha_{oth}$ , increases beyond competitive interaction from target (see Figure 8).

**Figure 10.** An example of cooperative navigation. Each panel shows the current configuration (bottom), and the corresponding vector field (top left). Heading direction is plotted so that  $\phi = 0$  corresponds to the agent’s current heading direction. The current competitive situation is also shown (top right): Competitive advantage,  $\alpha_i$  (dashed lines), and the current weighting,  $w_i$  (solid lines), are shown for each constraint. Competitive interaction is not shown. The top row of Panels shows vector field and competition for Agent 1 (above in configuration); The bottom row of Panels shows vector field and competition for Agent 2 (below in configuration). (A) Agent 1 reverses course to join Agent 2. (B) The agents meet one another and Agent 1 reverses course again. (C) The agents move toward the target together.

**Figure 11.** A second example of cooperative navigation. Each panel shows the current configuration (bottom), and the corresponding vector field (top left). Heading direction is plotted so that  $\phi = 0$  corresponds to the agent’s current heading direction. The current competitive situation is also shown (top right): Competitive advantage,  $\alpha_i$  (dashed lines), and the current weighting,  $w_i$  (solid lines), are shown for each constraint. Competitive interaction is not shown. Vector field and competition are shown for Agent 1 (right). (A) The two agents navigate toward the target together. (B) They are driven apart by obstacles, but continue toward the goal. (C) The agents pass the obstacles, and move toward one another. (D) They continue toward the target together.

## **Acknowledgments**

This work was in part supported 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. The authors wish to thank Tjeerd Dijkstra for many insightful discussions and Andy Hicks for helpful comments on an earlier draft of this paper.

**Table 1** Fixed points and stability conditions for two-constraint competition

$w_{tar}$	$w_{obs}$	Stability
0	0	Unstable $\alpha_{tar}, \alpha_{obs} > 0$
0	$\pm 1$	Stable $\gamma_{obs,tar} > \alpha_{tar}$
$\pm 1$	0	Stable $\gamma_{tar,obs} > \alpha_{obs}$
$\pm A_{tar,obs}$	$\pm A_{obs,tar}$	Stable $\alpha_{obs} > \gamma_{tar,obs}$ and $\alpha_{tar} > \gamma_{obs,tar}$

**Table 2** Fixed points and stability conditions for three-constraint competition

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

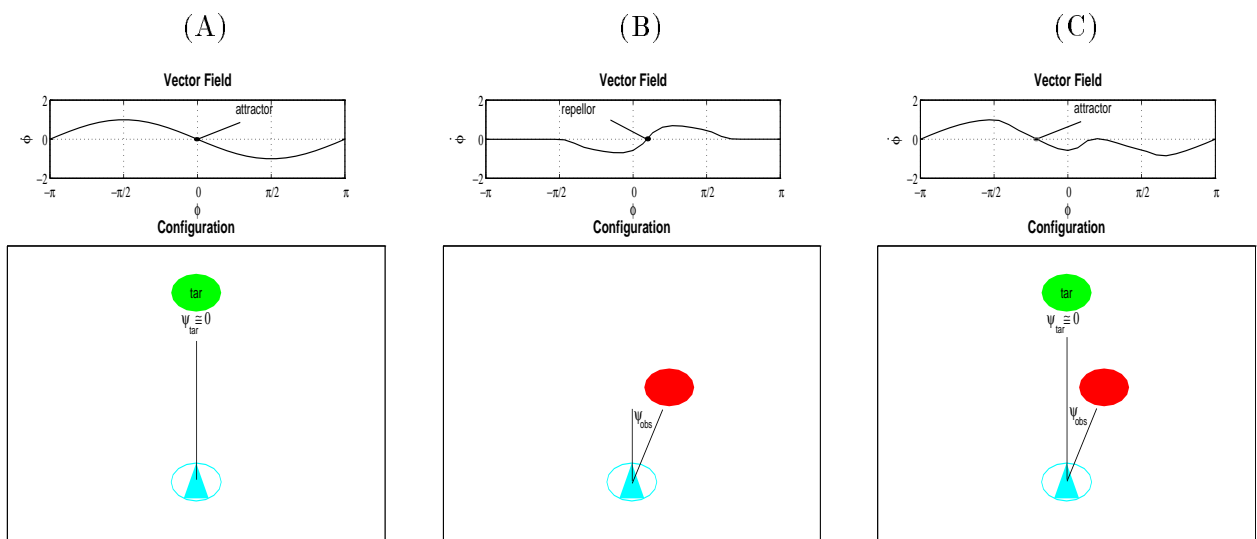


Figure 1

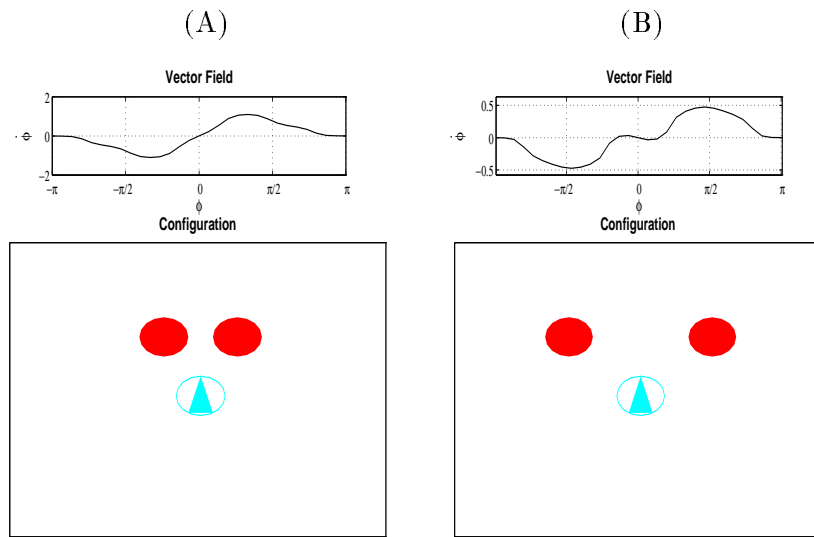


Figure 2

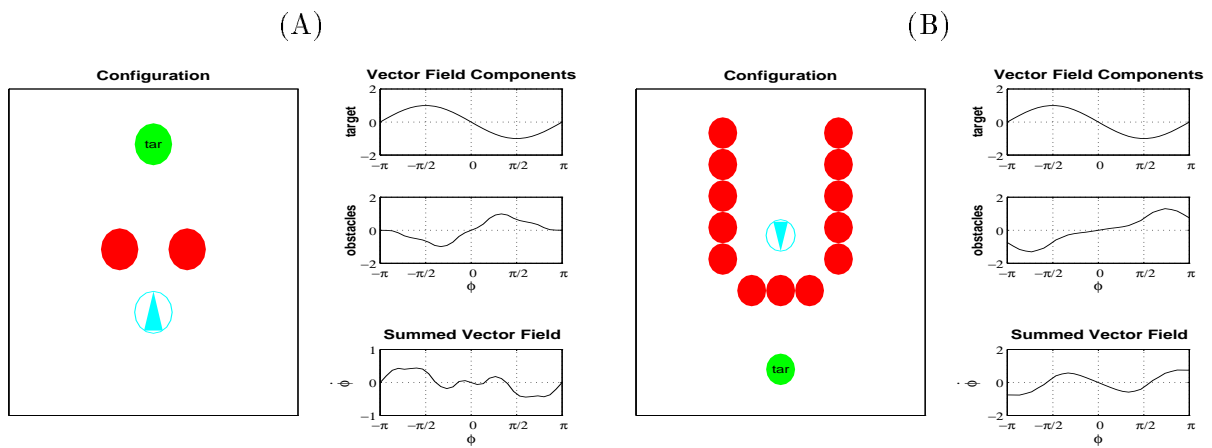


Figure 3

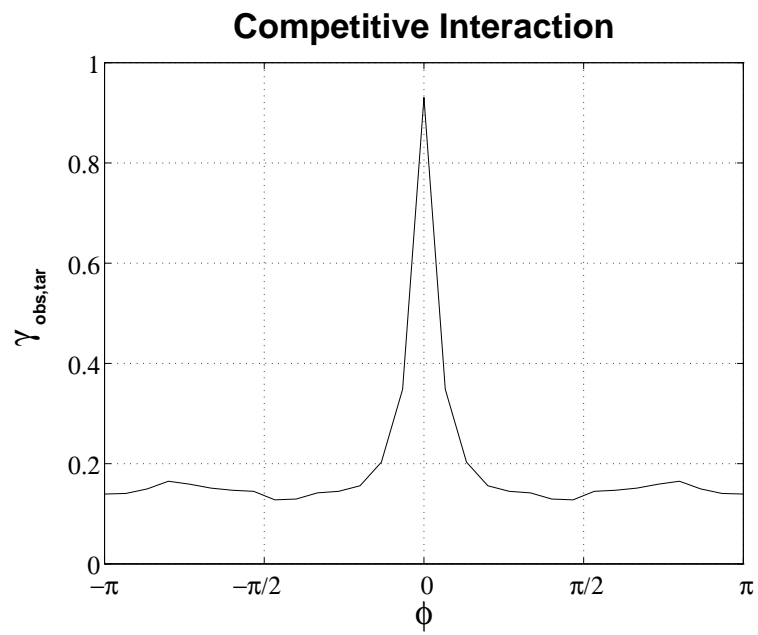


Figure 4



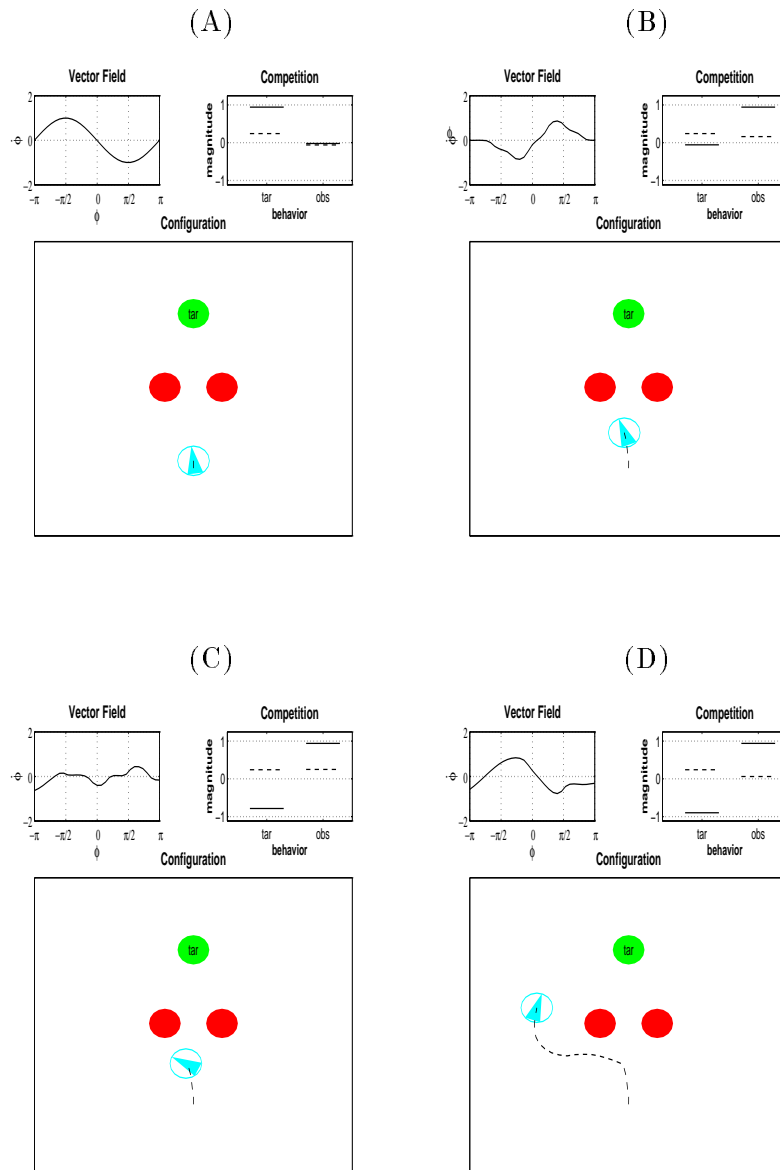


Figure 5

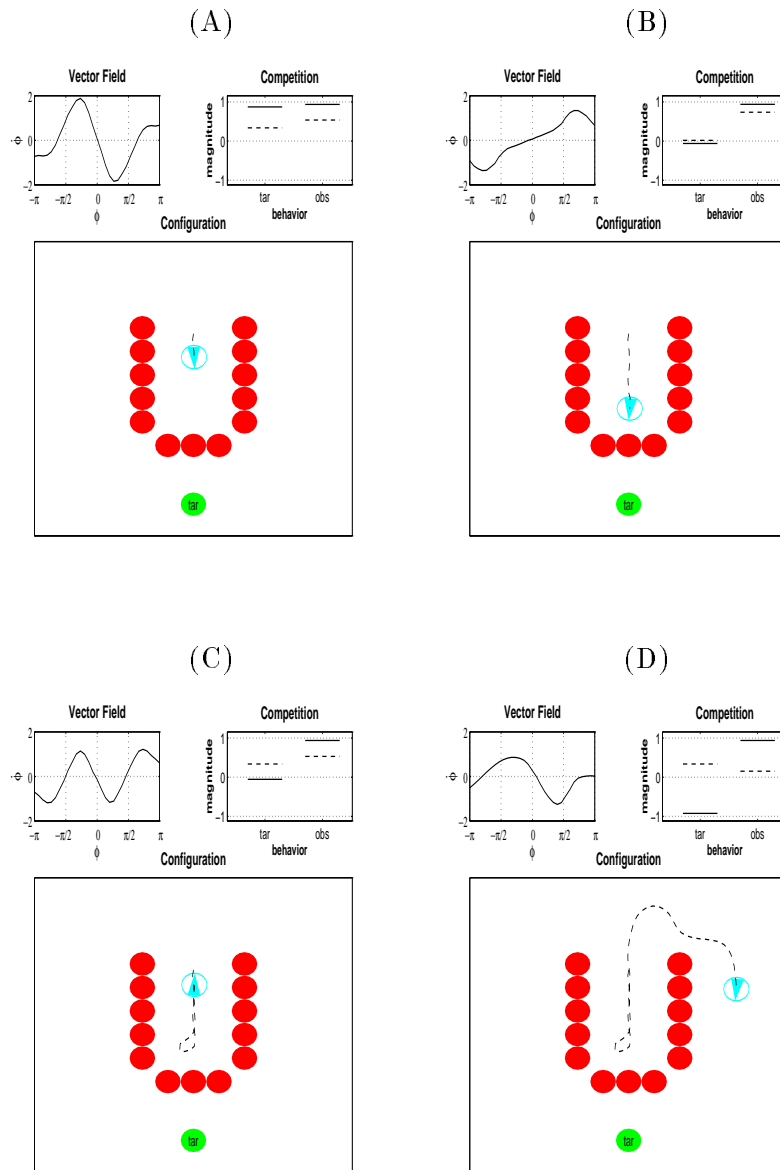


Figure 6

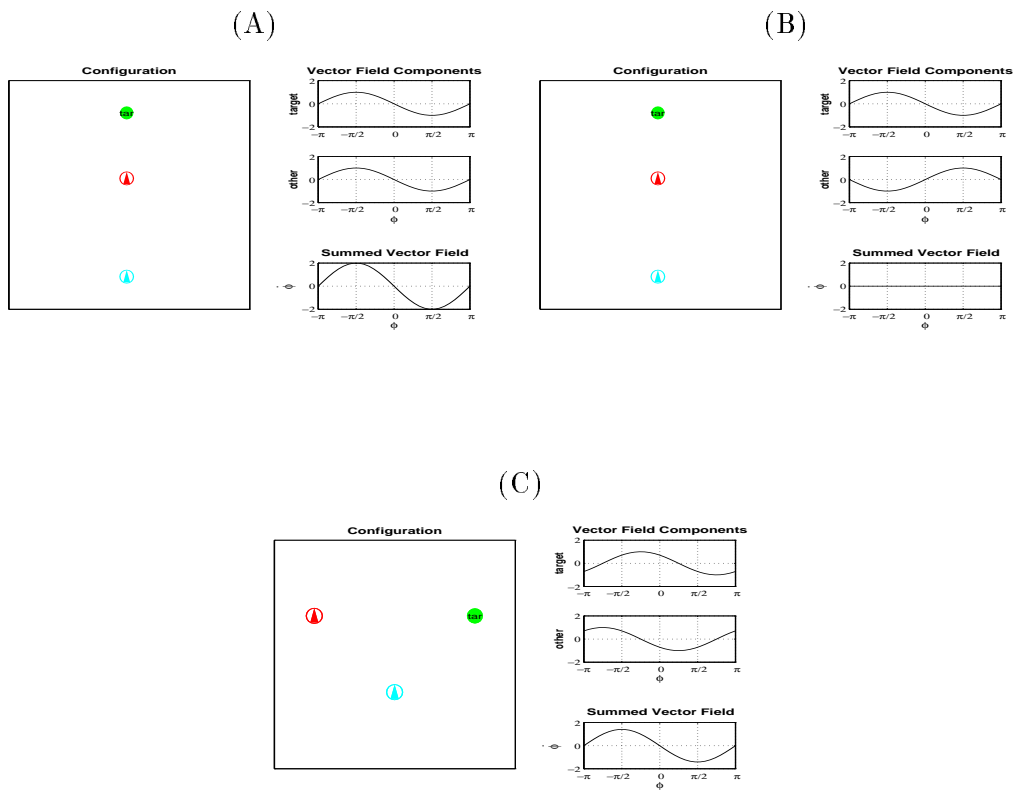


Figure 7

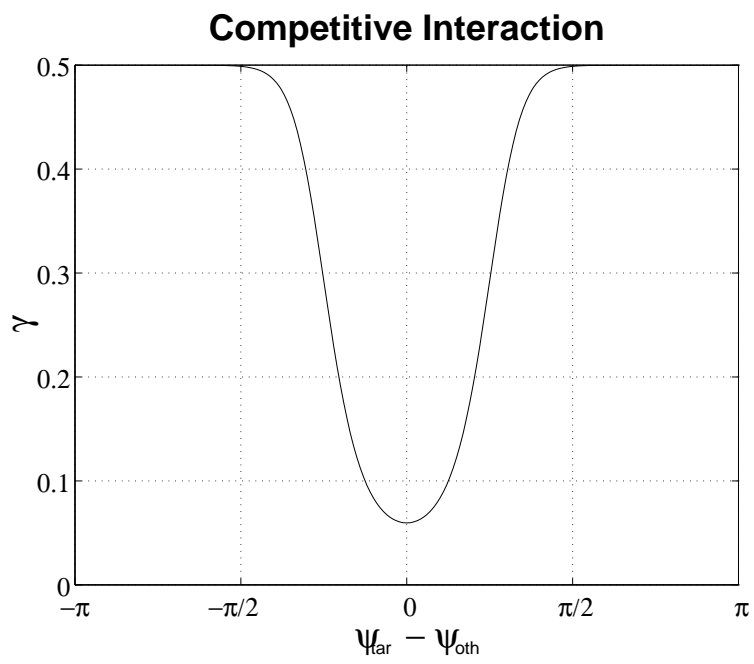


Figure 8

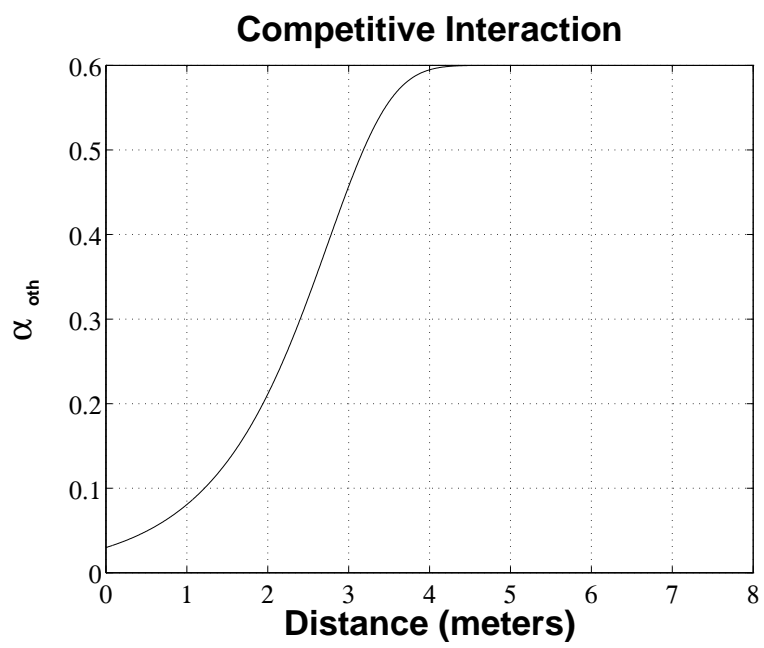


Figure 9

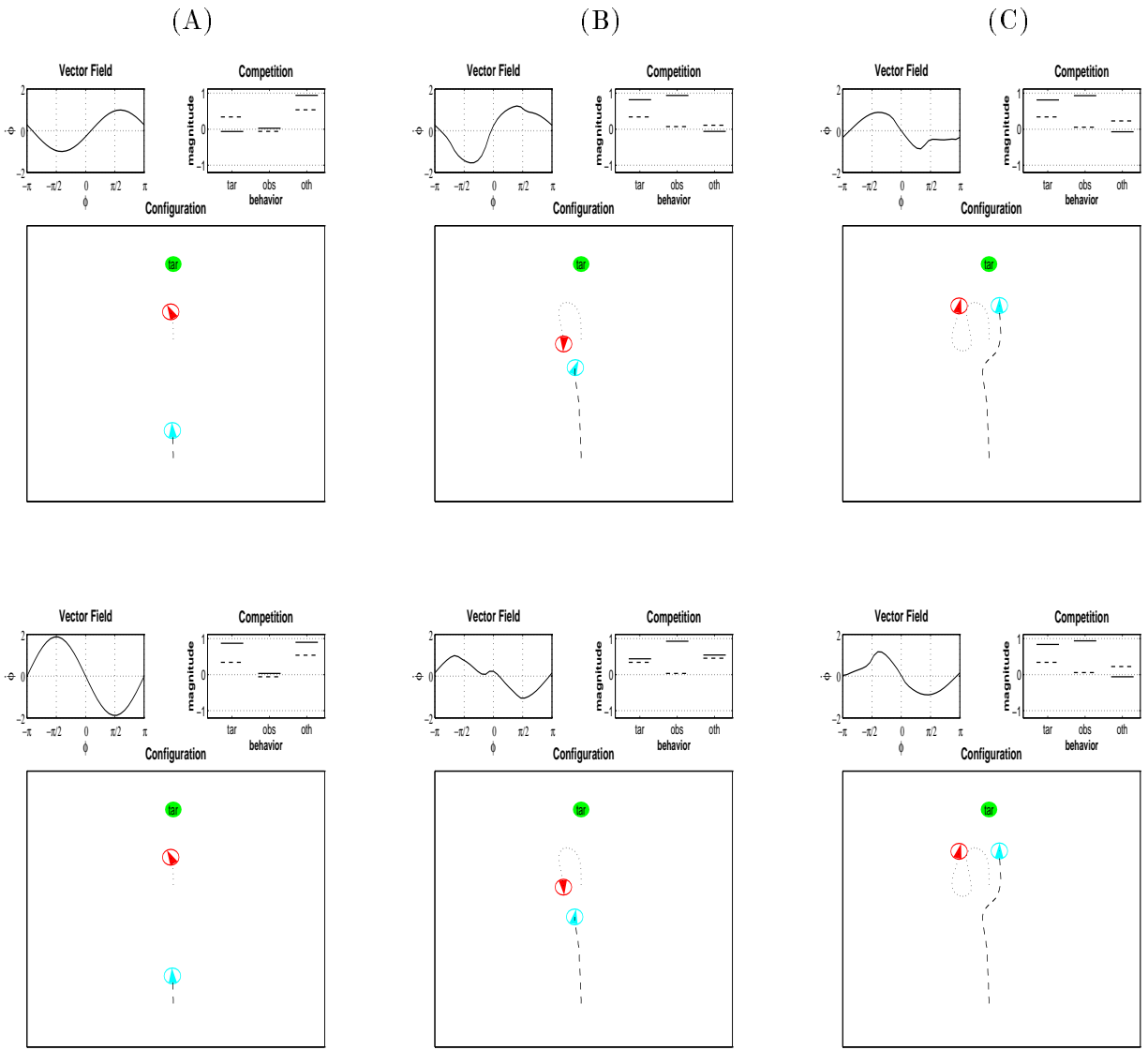


Figure 10

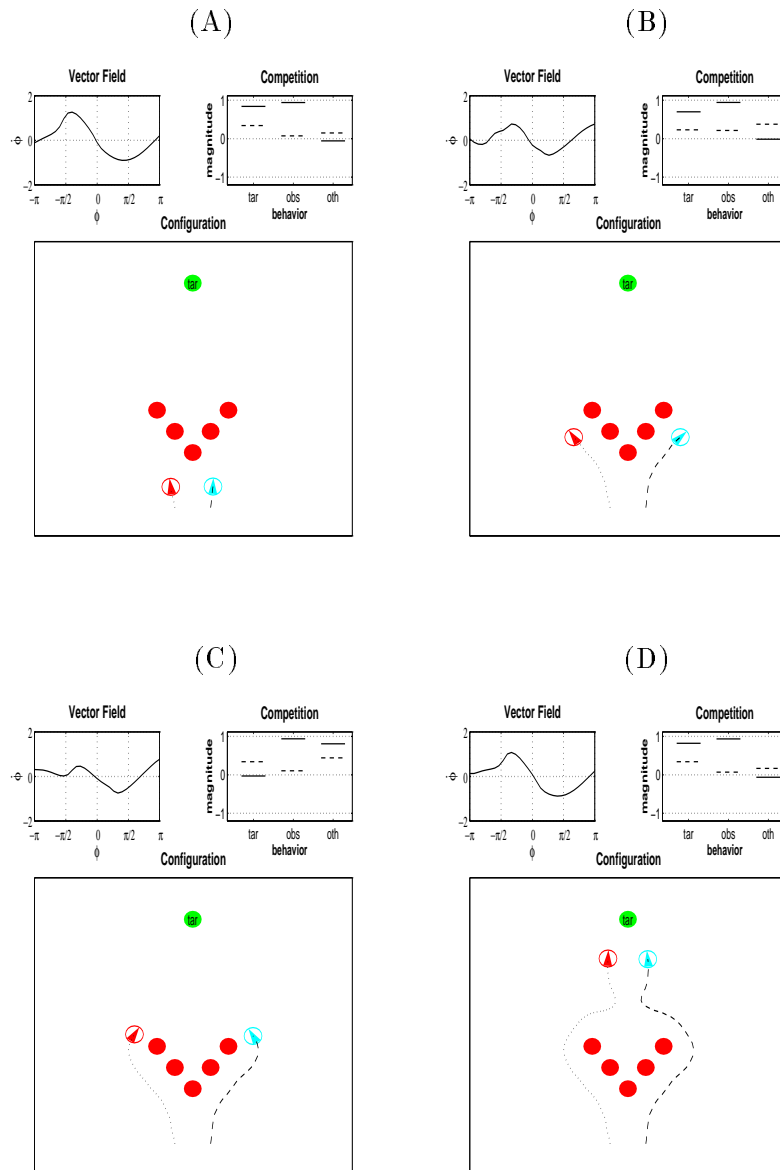


Figure 11