Edward W. Large

GRASP Laboratory
University of Pennsylvania
3401 Walnut Street, Suite 301C
Philadelphia, Pennsylvania 19104, USA

Henrik I. Christensen

Centre for Autonomous Systems
Kungliga Tekniska Hoegskolan
NADA/CVAP
S-100 44 Stockholm, Sweden

Ruzena Bajcsy

GRASP Laboratory
University of Pennsylvania
3401 Walnut Street, Suite 301C
Philadelphia, Pennsylvania 19104, USA

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

Abstract

The dynamic-systems approach to robot path planning 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. Competition among task constraints is able to deal with problems that arise when combining constraint contributions, making it possible to specify 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 make decisions regarding which behavior to execute in a given situation, resulting in the execution of sequences of behaviors that satisfy task requirements. We discuss the scalability of the competitive-dynamics approach to the design of more complex autonomous systems.

1. Introduction

Over the past 20 years or so, 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 a robot's actions during a navigation task. In spite of this effort, however, the question, What is the best space in which to represent robot behavior? remains open. Geometric representations (e.g., Schwartz and Sharir 1983; Latombe 1991) model the geometry of the agent and the external environment. The problem with this approach is that it is too static. Configuration-space representations (Lozano-Pérez and Wesley 1979; Murray, Li, and Sastry 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; Rimon and Koditschek 1993) 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 such as that described by Lumelsky and Stepanov 1987 and/or a representation whose dimensions correspond to robot behavior as in the work of 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, Dose, and Engels 1995). 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 International Journal of Robotics Research
Vol. 18, No. 1, January 1999, pp. 37-58,

- The environment is also modeled at a behavioral level. The environment provides *task constraints*, which 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 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 (Desai, Wang, Zefran, and Kumar 1996), it is appropriate for planning and control in dynamically changing environments. In addition, 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 important, 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, nonindependent contributions to the vector field can combine in such a way that they give rise to attractors corresponding to undesired behaviors. Spurious attractors may cause 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 nonindependent 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 contributions of different task constraints at any given time. The resultant weighted combination of constraints is similar in some respects to certain connectionist approaches (Jacobs, Jordan, and Barto 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.

The 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 to reflect the nature of the task. Given appropriately chosen functions that tie these parameters to the environment, this type of competition solves spurious attractor problems for the case of two constraints, (*target* and *obstacles*), and scales to the design of more complex systems. In this paper, a three-constraint system is used to simulate decision making for a pair of cooperating robots (cf. the work of Adams et al. 1995). Competition among task constraints allows each agent to make simple decisions about which behavior to execute in a given situation, resulting in sequences of behavior that are generated opportunistically, in response to specific environmental situations. Finally, we propose a set of general design principles intended to serve as guidelines 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. We then discuss potential problems regarding the representation of multiple behavioral requirements, including the development of spurious attractors. 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 by making decisions that generate 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 the 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, ϕ $(-\pi \leq \phi \leq \pi)$, and velocity, v (Schöner and Dose 1992). In this paper, we focus on a single behavioral dimension: the heading direction. We assume that velocity is controlled by a dynamics similar to that described by Neven and Schöner (1996).

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

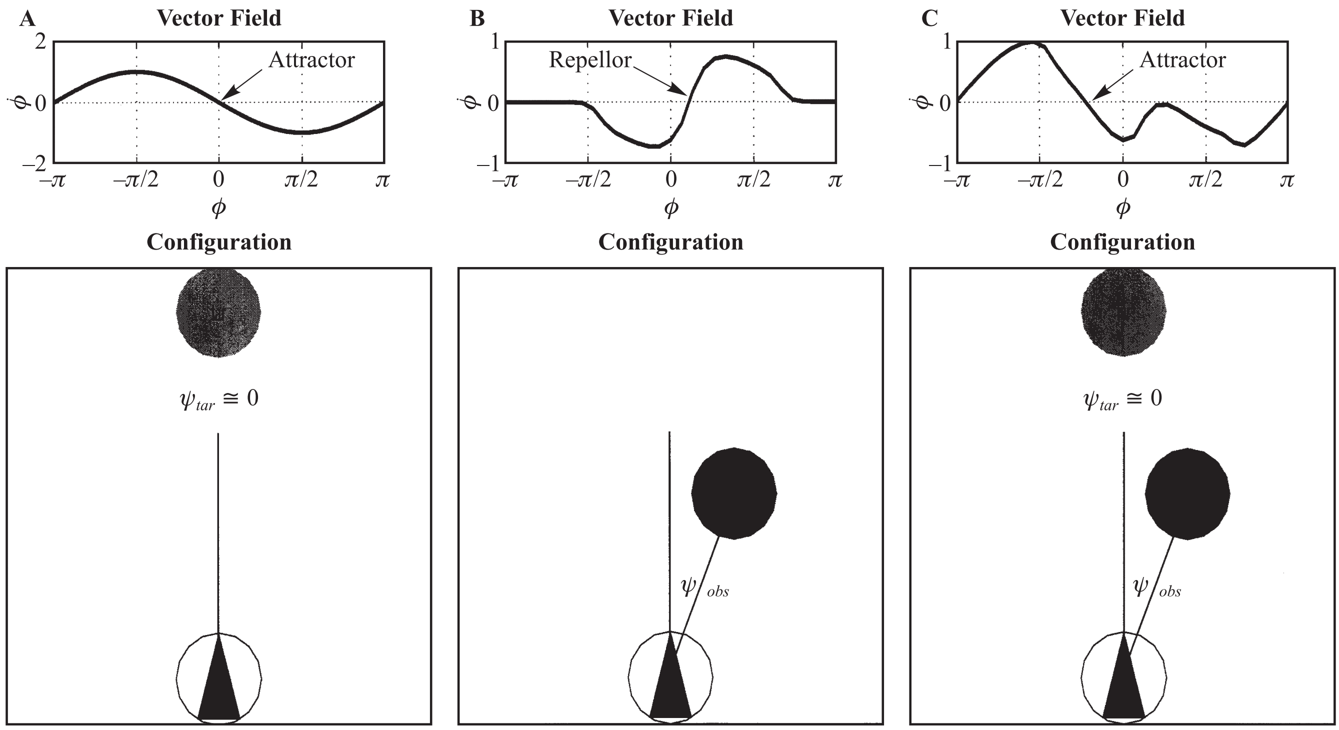

Fig. 1. Task constraints and behavioral dynamics. An agent, a target, and the corresponding vector field are illustrated in (a). The target constraint is expressed as a heading direction (zero corresponds to the current heading of the agent). The desired behavior of heading toward the target is expressed as an attractor (negative slope) in the vector field that governs agent-heading direction. An agent, an obstacle, and the corresponding vector field are shown in (b). The obstacle constraint is also expressed as a heading direction; however, the undesired behavior of the heading toward the obstacle is expressed as a repellor (positive slope) in the vector field. A more complex configuration is shown in (c). The target and obstacle constraints are additively combined into a single vector field. The attractor corresponds to steering around the obstacle *en route* to the target location.

For example, in the navigation task, the heading direction ψ_{tar} represents the direction to the target location, while the direction ψ_{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, Dose, and Engels 1995).

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:

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

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

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

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

where ϕ is the agent's heading direction in world coordinates, and ψ_{tar} is the direction toward the target location.

A to-be-avoided behavior is specified as a repellor (shown in Fig. 1b):

$$F_{obs_i} = R_{obs_i} \times W_{obs_i} \times D_{obs_i}. \tag{3}$$

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

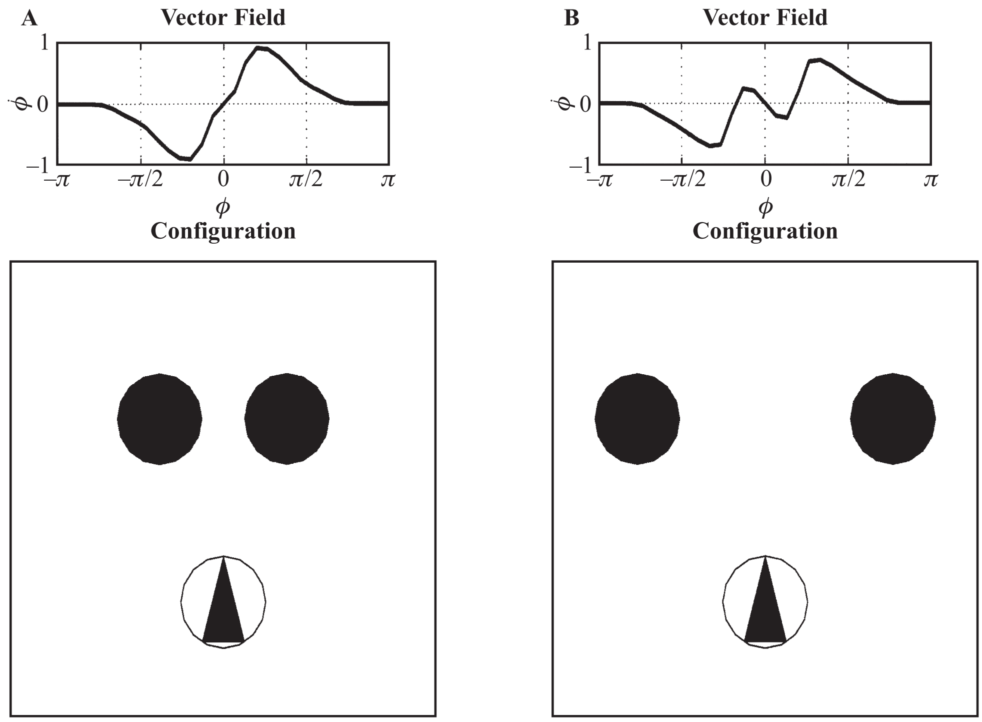

Fig. 2. Dependent and independent constraints: (a) two obstacles are configured so that there is not enough space between them for the agent to pass through. These constraints are dependent, and superposition of their contributions to the vector field creates a repellor at their average heading direction, effectively modeling a single obstacle; and (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.

$$R_{obs_i} = \frac{(\phi - \psi_i)}{\Delta\psi_i} e^{1-\left|\frac{\phi-\psi_i}{\Delta\psi_i}\right|}, \tag{4}$$

a second limits the angular range of the contribution,

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

and a third scales the strength of the contribution according to the obstacle's distance from the agent:

$$D_{obs_i} = e^{-\frac{r_i - R_i - R_{agent}}{d_0}}. \tag{6}$$

Here: ϕ is the heading direction of the agent; ψ_i is the direction to obstacle i; $\Delta\psi_i$ is the angular range subtended by the obstacle; R_i is the radius of the obstacle; R_{agent} is the radius of the agent; and σ is 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 this approach can be found in Schöner and Dose's work (1992). Multiple obstacles are handled by summing the contributions of individual obstacles:

$$F_{obs} = \sum_{i=1}^{n} F_{obs_i}. \tag{7}$$

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

$$\dot{\phi} = F_{tar} + F_{obs} + \sqrt{Q_b}\,\xi(t). \tag{8}$$

Here, $\sqrt{Q_b}\,\xi(t)$ represents a Gaussian noise term with zero mean and variance Q_b. Because certain constraints are modeled as repellors, the planning dynamics is augmented by this stochastic term ensuring escape from unstable fixed points (repellors).

An important feature of this approach is the concept of asymptotic stability of behavior, brought about by generating behavior from a dynamics, rather than directly from the task constraints. Qualitative change in behavior arises through change in the number, nature, or stability of attractors and repellors. 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 path-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 repellor to the vector field, while avoidance of multiple obstacles is modeled by summing multiple range-limited repellor 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 Figure 2a 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 repellor in the vector field at their average location: behaviorally, a single obstacle. In Figure 2b, the agent again faces two obstacles, but this time they are positioned far enough apart for the agent to pass between them. These two constraints are independently valid, and an attractor is formed in the vector field, corresponding behaviorally to steering between the two obstacles.

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. Schöner and Dose (1992) deal with this problem using a competitive interaction among obstacles. Sensed obstacles that overlap are forced to compete in such a way that 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 Schöner 1995; Schöner, Dose, and Engels 1995). 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 is 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 repellor 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 repellor, an attractor is created between the two obstacles. This attractor will cause the agent to get stuck at this location.

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 repellor exists that would cause the agent to turn around and leave the hallway. However, the repellor 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. To deal with situations such as these, we further modify the strength of each contribution with a specific weight that is assigned to each type of task constraint (target and obstacles):

$$\dot{\phi} = |w_{tar}|F_{tar} + |w_{obs}|F_{obs} + \sqrt{Q_b}\xi(t) . \qquad (9)$$

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

$$\dot{w_i} = \alpha_i w_i(1 - w_i^2) - \sum_{j \neq i} \gamma_{j,i} w_j^2 w_i + \sqrt{Q_t}\xi(t) . \qquad (10)$$

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 term $\sqrt{Q_t}\xi(t)$ represents Gaussian noise, with zero mean and variance Q_t. The parameters α_i and $\gamma_{j,i}$ are referred to as the *competitive advantage* and the *competitive interaction*, respectively.

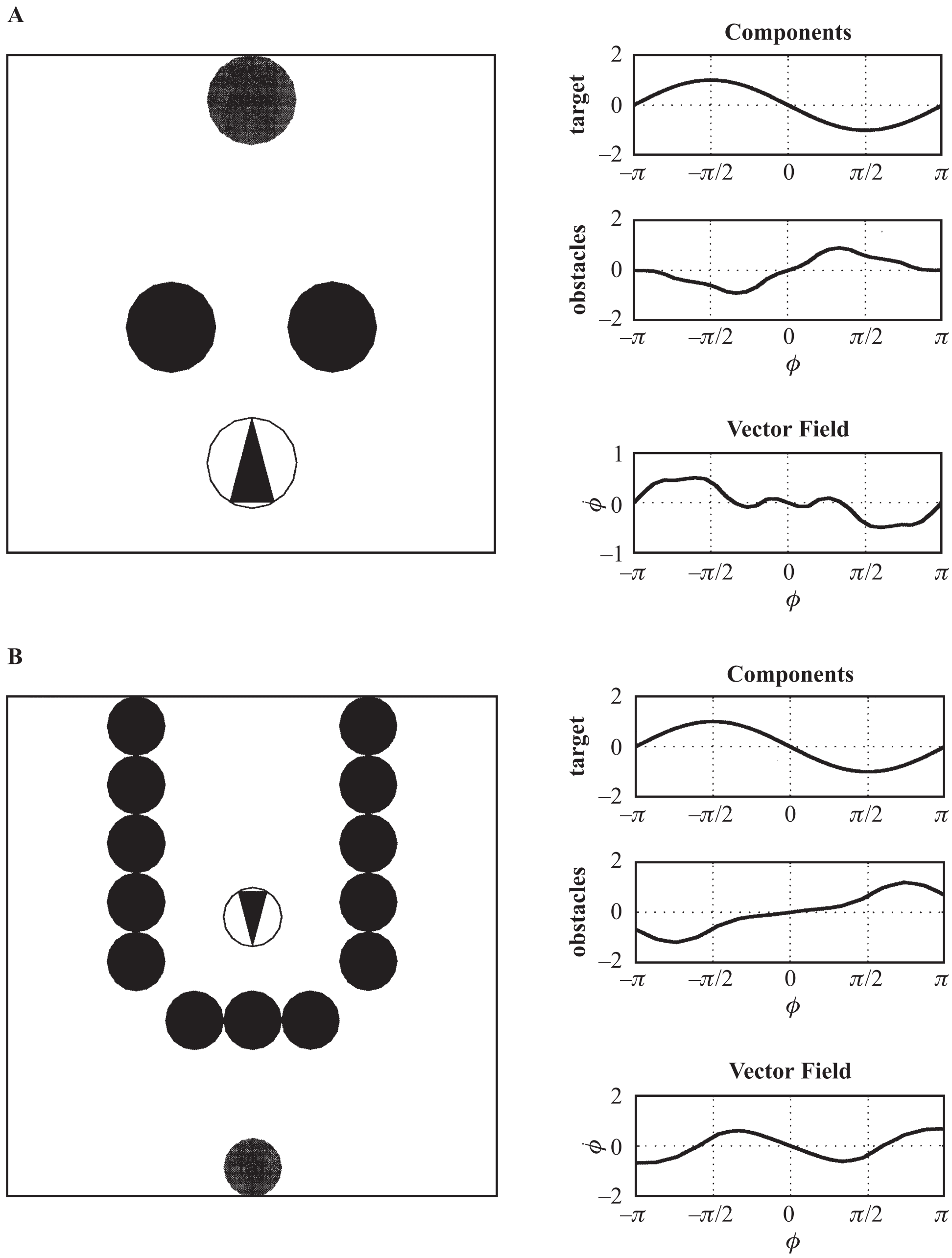

Fig. 3. Spurious attractors. Two obstacles dead ahead provide almost, but not quite, enough space for the agent to pass between them (a). The obstacles contribution to the dynamics reveals a shallow attractor. Yet when the target also lies straight ahead, its attractor contribution, combined additively with the repellor, creates a spurious attractor in the composite vector field. This will cause the agent to get stuck at this location. A hallway trap is shown in (b). 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.

Competitive advantage, α_i, describes the degree current to which constraint i is appropriate to the agent's 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.

If we consider just the first term of the dynamic equation, then α_i is the only parameter, and this system resembles the normal form for a pitchfork bifurcation (Guckenheimer and Holmes 1983). 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. The second term specifies competitive interaction from other constraints. The dynamics in the case of multiple competing constraints is more complex, and is investigated below for the cases of two and three interacting constraints.

Equation (10) describes a competitive dynamics similar to that proposed by 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, which is a computationally intensive proposition. Our use of competitive dynamics (i.e., eq. (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 to the specification of more complex systems.

In the next section, we use competition to address the issue of spurious attractors for the case of two task constraints. In this process, we outline a set of design principles that will be applicable to the specification of larger numbers of behavioral requirements. We claim that this strategy of competitive interaction among task constraints is general enough to support systems in which the agent possesses a rich set of potential behaviors. In the following section, we demonstrate scalability, using this methodology to design agents that perform a more complex task.

3. Competition for the Case of Two Task Constraints

We first use competitive dynamics to address the spurious attractor problem for the case of two task constraints. Our development proceeds in three stages. First, we perform a stability analysis that tells us how relative values of the parameters α_i and $\gamma_{j,i}$ determine the resultant weighting of task constraints. In the second stage, we identify situations where the two constraints, target and obstacles, are incompatible. This leads to the design of functional forms that tie competitive interactions, that is, the $\gamma_{j,i}$, to specific situations. In a final 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, α_i, of each constraint.

3.1. Stability Analysis

A linear stability analysis (Perko 1991) was performed on the system described by eq. (10) for $i \in 1, 2$; that is, the case of two behavioral constraints, assuming $\gamma_{j,i} > 0$. 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; that is, it determines whether the fixed point is an attractor or repellor of the competitive dynamics. Because the stability of each equilibrium point changes depending upon the values of the parameters α_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.

Table 1. Fixed Points and Stability Conditions for Two-Constraint Competition

ω_{tar}	ω_{obs}	Stability	
0	0	Stable	$\alpha_{tar}, \alpha_{obs} < 0$
0	± 1	Stable	$\gamma_{obs,tar} > \alpha_{tar}$
± 1	0	Stable	$\gamma_{tar,obs} > \alpha_{obs}$
$\pm A_{tar,obs}$	$\pm A_{obs,tar}$	Stable	$\alpha_{obs} > \gamma_{tar,obs}$
		and	$\alpha_{tar} > \gamma_{obs,tar}$

The results of our analysis are shown in Table 1. There are nine equilibrium points, because each nonzero point has both a positive and a 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 points reduce to four unique equilibrium points.

Each equilibrium point corresponds to a different behavior, because each unique equilibrium point yields a qualitatively different composition of task constraints, a different vector field governing behavior. The first point, (0, 0), corresponds to both constraints, target and obstacle being effectively turned off. This point is stable (an attractor 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 inhibition from obstacles is greater than the competitive advantage of target. The resultant behavioral composition is appropriate in situations such as those depicted in Figure 2a, in which the superposition 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 of 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 the competitive interaction from target is greater than the competitive advantage of obstacles. This behavior is appropriate in situations where there are no obstacles near the agent.

Note that the above stability conditions are not mutually exclusive. When both conditions are satisfied, we have *bi-*

stability, and hysteresis will determine the outcome of the competition: 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 so-called averaging solution (Schöner and Dose 1992) 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 superposition in the vector field.

The averaging solution is given by

$$A_{i,j} = \sqrt{\frac{\alpha_i \alpha_j - \alpha_j \gamma_{j,i}}{\alpha_i \alpha_j - \gamma_{i,j} \gamma_{j,i}}}. \tag{11}$$

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 is equivalent to that described by 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 (namely, target and obstacles), four behaviors are possible: doing nothing, seeking a target, avoiding obstacles, and navigation (target seeking plus obstacle avoidance, arising from the averaging solution). We 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 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. Competitive Interaction

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 repellor collide (see Fig. 3), unwanted consequences may result, because the two contributions are (1) nonindependent, and (2) contradictory. We design "fixed-point detectors" that capture the location and stability of the fixed points for each

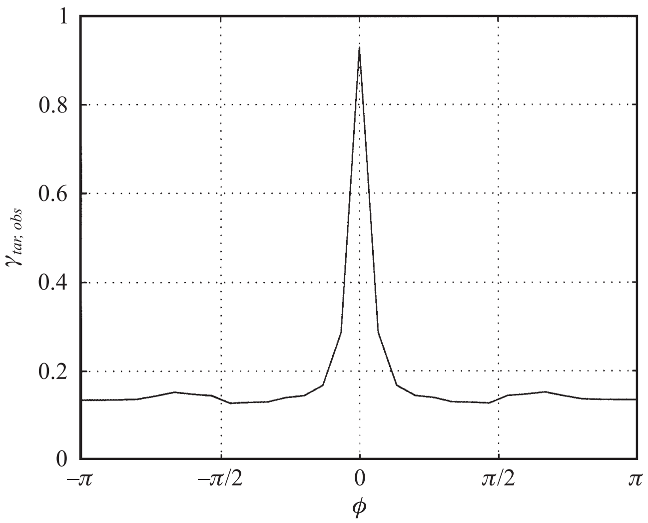

Fig. 4. Competitive interaction between obstacles and target for the spurious attractor example shown in Figure 3a.

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 repellors for the individual contributions to the behavioral dynamics. For the *target* contribution, we use

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

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 falls to zero as the magnitude of the contribution grows. The constant c_1 determines the rate of falloff, allowing the specification of a safety margin around the attractors and repellors if necessary. At a repellor, P_{tar} has a value of one; at an attractor, minus one; and elsewhere, values approach zero. Thus, it describes the location and stability of the fixed points of the target contribution to the behavioral dynamics.

We use a similar equation for obstacles. However, because individual obstacle contributions are range limited, that is, have values near zero outside an obstacle's range, eq. (12) will identify these areas as fixed points. Thus, we sum the range-limiting functions for the obstacles given in eq. (5) (i.e., $W_{obs} = \sum_{i=1}^{n} W_{obs_i}$), and use this as a multiplicative factor:

$$P_{obs} = W_{obs}\, sgn(\frac{dF_{obs}}{d\phi})\, e^{c_1 |F_{obs}|}. \tag{13}$$

As above, this function has a value of one at a repellor, 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 that describes the competitive interaction between obstacles and target as

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

The graph of eq. (14) is shown in Figure 4, corresponding to the situation depicted in Figure 3a. It is strongly peaked at the point of attractor-repellor collision, and constant c_2 determines the rate of drop-off around the collision. Note that it also provides a certain level of background competition that we will later use to help determine the appropriate level of competitive advantage, α_{tar}, for target.

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

3.3. Competitive Advantage

In the previous section, we designed functions that capture situations in which target and obstacles should compete; that is, when attractor and repellor contributions would "collide." From the stability analysis, we know how to pick relative values of α_i and $\gamma_{j,i}$ in such a way that we can specify the type of behavior that we would like in any specific situation. In this section, we complete the design, choosing 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 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 compete 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 eq.(14) (shown in Fig. 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 a 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} (eq. (6)), which is a component of the function F_{obs_i}. To count the number of obstacles around the agent, we sum the D_{obs_i}. We then limit the maximum value of the α_{obs}, resulting in the following function for competitive advantage:

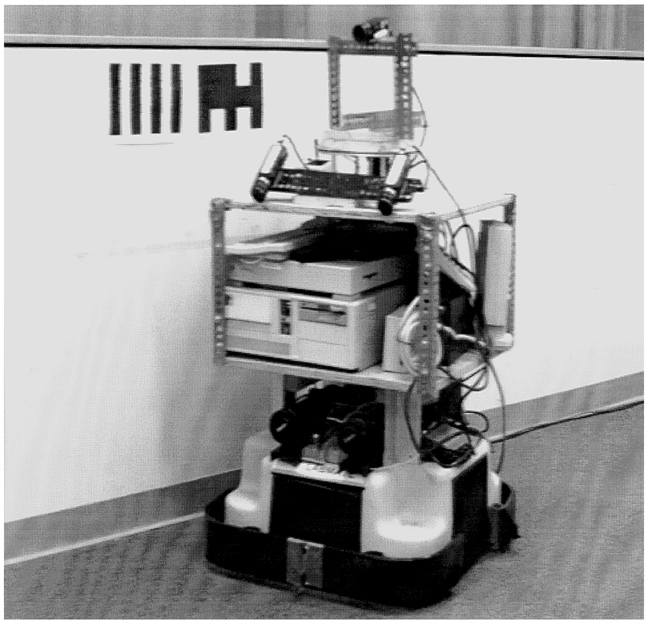

Fig. 5. An autonomous robot used to test the dynamical systems controller.

$$\alpha_{obs} = \tanh \sum_{i=1}^{n} D_{obs_i} . \tag{15}$$

This completes our design.

3.4. Examples

In Section 2.3, we saw two situations in which spurious attractors were created by superposition of nonindependent, contradictory contributions in the vector field. In this section, we demonstrate how competition deals with these situations.

The dynamical decision-making and path-planning system has been implemented as a controller for the mobile platforms in our laboratory. One of these robots is shown in Figure 5. Two separate CCD camera rigs provide the dynamical controller with the necessary information about targets and obstacles in its environment. The robots identify and track targets using a single turntable-mounted camera and a landmark-detection algorithm (Venetianer, Large, and Bajcsy 1997). They identify obstacles using a separate stereo pair and an inverse-perspective algorithm (Mallot, Bülthoff, Little, and Bohrer 1991). Each of the examples in this and the following sections has been tested and verified on the mobile platforms. For presentation clarity, the examples presented in this paper have been generated as the dynamical system controller was run in simulation mode.

In Figure 3a, a spurious attractor arose when a repellor, created by two obstacles, was combined with an attractor from the target contribution. Figure 6 shows four snapshots

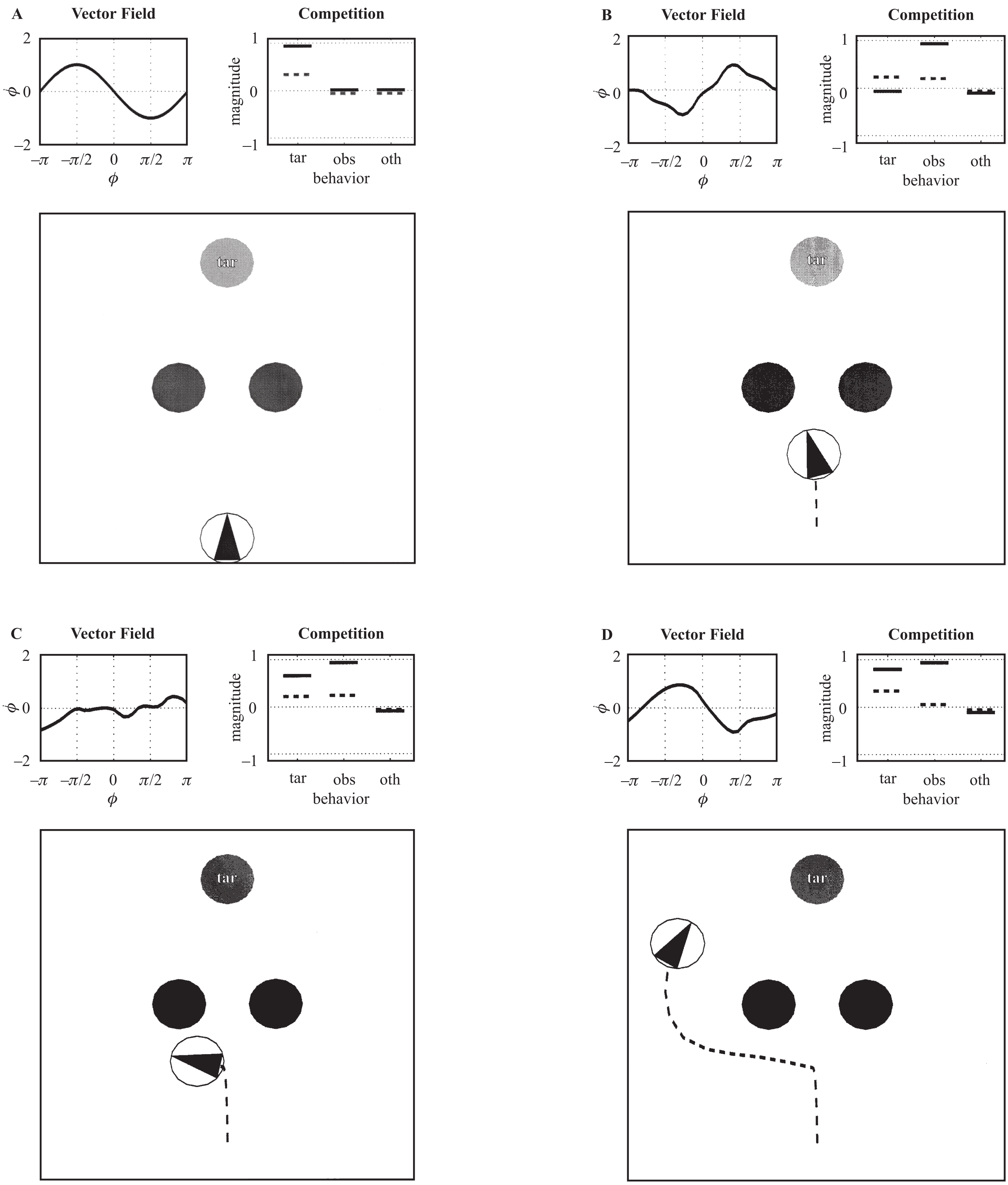

Fig. 6. Competition avoids the creation of a spurious attractor. Each panel shows a current configuration (bottom), and the corresponding vector field (top left). Heading direction is plotted so that $\phi = 0$ corresponds to the agent's current heading direction. The current competitive situation is also given (top right): competitive advantage, α_i (dashed lines), and the current weighting, ω_i (solid lines), are shown for each constraint. Competitive interaction is not shown. The agent approaches two obstacles that it cannot pass between (a). This situation could create a spurious attractor (b) (compare with Figure 3a), but it does not because competition deactivates target; competitive interaction for this situation is shown in Figure 4. Once the agent has turned away from the obstacles, target is reactivated (c). The agent then rounds the leftmost obstacle, steering toward the target location (d).

from an episode in which the agent, using competition among behavioral constraints, successfully navigates this situation. Figure 6a 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 6b 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 6b 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 6c 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 active as well. This is the "averaging" solution, resulting in a composite behavior that combines the two constraints. Finally, Figure 6d 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 nontrivial behavior that arises from the competitive dynamics in the case of two task constraints.

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* is deactivated by competition (see Fig. 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 ($\gamma_{obs,tar}$) drops, and the influence of *target* once again causes the agent to turn toward the target. The problem here is not simply that *target* and *obstacles* are contradictory. Rather, in this context, *target* is not a useful constraint. The agent is trapped in an enclosure from which it must escape before the *target*-constraint becomes useful. In other words, the agent must establish the intermediate goal of escaping from the enclosure.

In this case, 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 (1) is 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

$$\alpha_{tar} = a_{tar} - (1 - V_{tar})a_{tar}\, \alpha_{obs}. \tag{16}$$

Here, a_{tar} is the competitive advantage for target, as described above. The competitive advantage of obstacles is α_{obs}, 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:

$$V_{tar} = \begin{cases} 0 & \text{if obstacle between agent and goal,} \\ 1 & \text{otherwise.} \end{cases} \tag{17}$$

Thus, the second term of eq. (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 7 shows an example of the agent successfully negotiating the hallway trap using the competitive advantage described by eq. (16). In Figure 7a, 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 7b, the agent has encountered the dead end and begins to turn away. If the collision of an attractor and repellor were the only factor in deactivating target, the agent would quickly turn back toward the target. However, because α_{tar} falls below the background level of competition, target is turned off. Next, Figure 7c shows an interesting, and less obvious, case in which an attractor and a repellor collide: when the attractor is supplied by obstacles, and the repellor is contributed by target. Here the agent must move directly away from the target 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 repellor. Finally, Figure 7d shows the agent successfully making its way out of the enclosure and toward the target. This behavioral sequence arises due to competition among behavioral constraints, which allows the agent to decide which behavior is appropriate, depending upon the situation. This example also demonstrates a case in which the behavioral situation itself, rather than the contradictory nature of task constraints, temporarily rules out a particular contribution.

In summary, we have described a method of weighting behaviors that precludes the creation of spurious attractors in the vector field through competition among individual contributions. The basic idea is that the competition equations detect situations in which nonindependent 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. Additionally, in the process of constructing the above system, we outlined a design methodology: stability analysis, design of competitive interaction, and design of competitive advantage. In the next

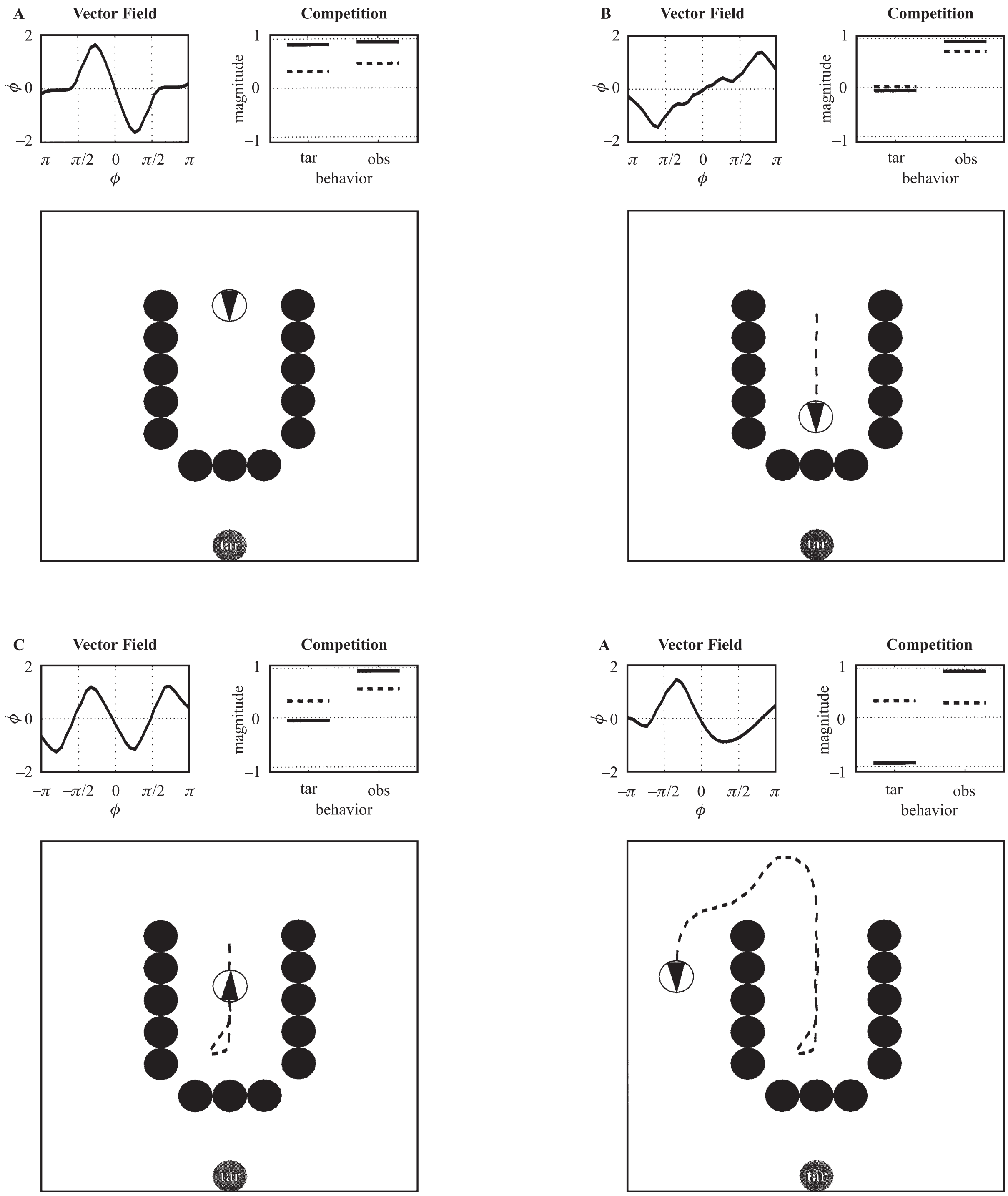

Fig. 7. The agent successfully negotiating the hallway trap. Each panel shows the current configuration (bottom), and the corresponding vector field (top left). Heading direction is plotted so that $\phi = 0$ corresponds to the agent's current heading direction. The current competitive situation is also shown (top right): competitive advantage, α_i (dashed lines), and the current weighting, ω_i (solid lines), are shown for each constraint. Competitive interaction is not shown. The agent moves down the hallway toward the target (a). The agent faces the spurious attractor situation shown in Figure 3b. Competitive advantage, α_{tar} , drops below the background level of competition created by eq. (14), allowing the agent to turn away from the dead end (b). The agent moves out of the hallway, due to active competition between *obstacles* and *target* (c), and successfully leaves the trap (d).

section, we construct a system of three task constraints to demonstrate how this approach scales when task requirements are more complex.

4. Cooperation through Competition

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; that is, they must perform the navigation task. Second, we impose the constraint that the two agents must remain near one another as they make their way toward the target location. Thus, each agent must respect a third behavioral requirement, which we call *other*.

We begin by making some simplifying assumptions. First, we wish to build upon our previous design. Target seeking and obstacle avoidance are to operate as in a single-agent case, described above. Second, agents are to avoid collision with one another in the same way as they would avoid stationary obstacles. Thus another agent is, among other things, an obstacle to be avoided. Finally, there is no centralized control. The behavior of each individual agent is governed by the same dynamic equations, but generated independently, each by the time course of its own behavioral variables. The two agents do not communicate with one another about their plans; however, we assume that they know one another's position.

Let us consider the additional constraint of staying near the companion agent. Similarly to target seeking, we can model this constraint as an attractor:

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

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

$$\dot{\phi} = |w_{tar}|F_{tar} + |w_{obs}|F_{obs} + |w_{oth}|F_{oth} + \sqrt{Q_b}\xi. \tag{19}$$

The potential problem with combining these three constraints lies in the composition of *target* and *other*, as depicted in Figure 8. In Figures 8a and 8b, we see a situation in which two agents are headed toward the target, yet one is considerably ahead of the other. In Figure 8a, we look at the situation from the point of view of Agent 1 (bottom). The composition of the *target* and *other* contributions sum in such a way that the agent is to move straight ahead. This is acceptable for Agent 1, since both the target and the other agent lie in the same direction. In Figure 8b, we see the situation from the point of view of Agent 2 (middle). The target and the other agent lie in opposite directions, and the composition of these two contributions cancel one another entirely. This is clearly not acceptable. A different situation is depicted in Figure 8c, shown from the point of view of Agent 1 (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 nonindependent contributions averages the corresponding constraints. In some cases (Fig. 8a), this yields appropriate behavior; in other cases (Figs. 8b and 8c), is does not.

Our task is 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 eq. (10) for three behaviors. As above, we assumed $\gamma_{j,i} > 0$. The results of the analysis are shown in Table 2, revealing eight unique equilibrium points. The stability analysis also reveals distinct *classes* of solutions. Thus, rather than describe each behavior individually, we describe each class of stable fixed points. This makes the job of understanding the competitive dynamics easier. It also illustrates important features regarding the scalability of this approach.

The first class of solutions corresponds to deactivating all constraints. This solution is stable whenever all competitive advantages are less than zero. The second 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 third 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; that is, it is given by eq. (11).

Note that, similarly to the two-dimensional case, the above stability conditions are not all mutually exclusive. In cases of bistability, equilibrium points are determined by hysteresis; thus the resultant behavior of the agent is determined by its

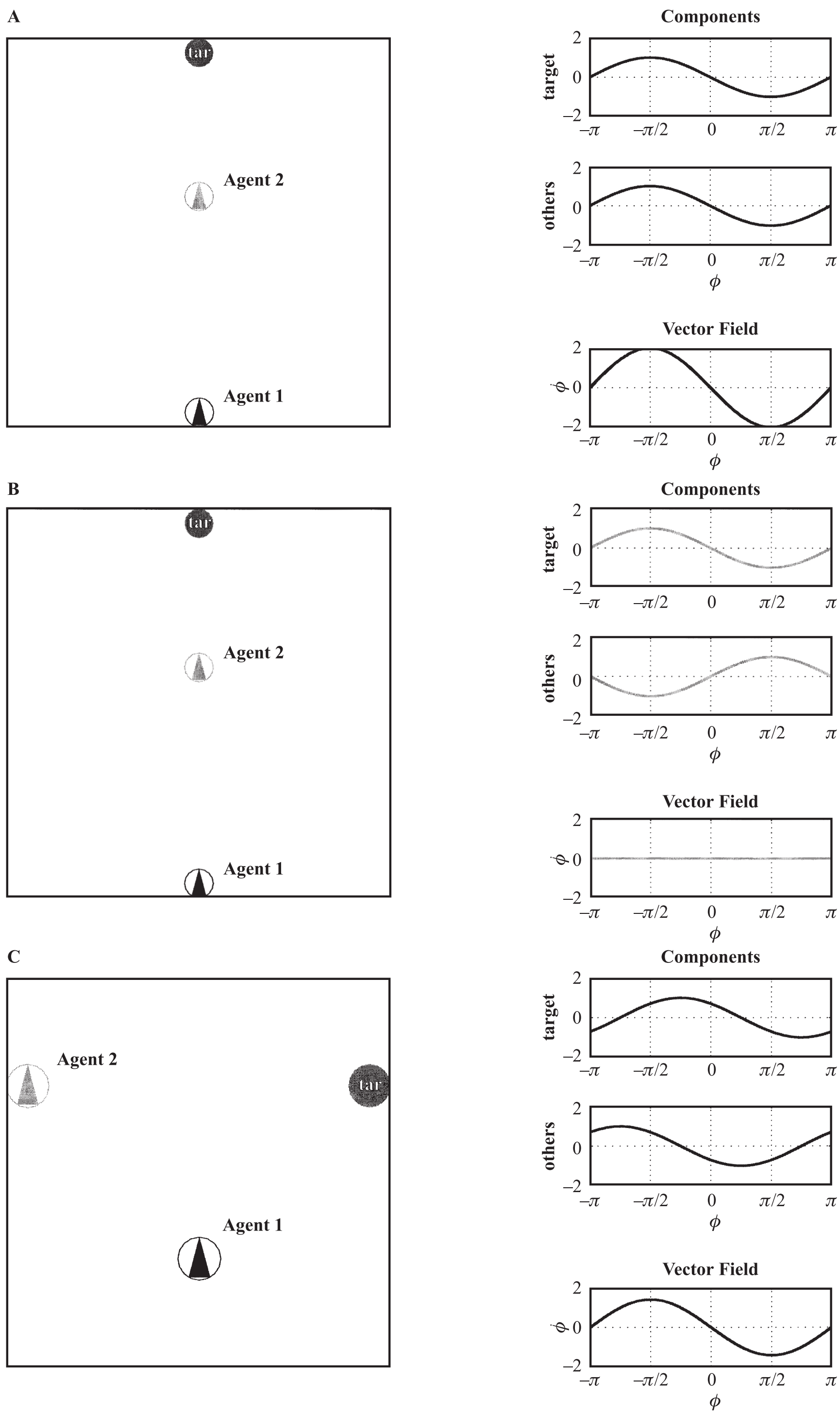

Fig. 8. Nonindependent constraints: when constraints are dependent, additive composition results in constraint averaging. From the point of view of Agent 1 (bottom), the target and the other agent are in the same direction, so constraint averaging is acceptable (a). From the point of view of Agent 2 (middle), the target and the other agent are in opposite directions, so constraint averaging is clearly unacceptable (b). From the point of view of Agent 1 (bottom), the target and the other agent lie in different directions. Averaging may be unacceptable, although this judgment depends somewhat on the task specification (c).

Table 2. Fixed Points and Stability Conditions for Three-Constraint Competition

ω_{tar}	ω_{obs}	ω_{oth}	Stability	
0	0	0	Stable	$\alpha_{tar}, \alpha_{obs}, \alpha_{oth} < 0$
± 1	0	0	Stable	$\gamma_{tar,j} > \alpha_j,\ \forall j \neq tar$
0	± 1	0	Stable	$\gamma_{obs,j} > \alpha_j,\ \forall j \neq obs$
0	0	± 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	$\alpha_{oth} > \gamma_{i,oth},\ \forall j \neq oth$

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 "three-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 eq. (10), but it is revealed by the stability analysis.

Second, the above observations regarding the bifurcation structure of the competition dynamics generalize to systems with any number of constraints. Therefore, the stability analysis need not be performed explicitly for larger systems; fixed points and stability conditions can be written down directly. Thus, we can count the number of unique behaviors that arise in a system of n constraints. It is simply the number of ways to choose zero active constraints, plus the number of ways to choose one active constraint, and so on. In other words, the number of behaviors generated in such a system is

$$N = \binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n} = \sum_{i=1}^{n} \binom{n}{i} = 2^n. \quad (20)$$

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.

Third, given an existing $(n-1)$-constraint system, one adds the n^{th} constraint simply by considering interactions with each of the existing $n-1$ constraints. The existing system functions 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*.

In summary, the stability analysis reveals important facts about the scalability of this approach. In an n-constraint system, competition provides 2^n unique behaviors. However, designing the system requires at most n^2 design decisions: $n^2 - n$ competitive interactions, plus n competitive advantages. Furthermore, the analysis revealed that such systems implicitly obey a modularity principle. A new task constraint can be added to the existing system without disturbing the previous design in any way. We will use these facts in the design of our three-constraint system.

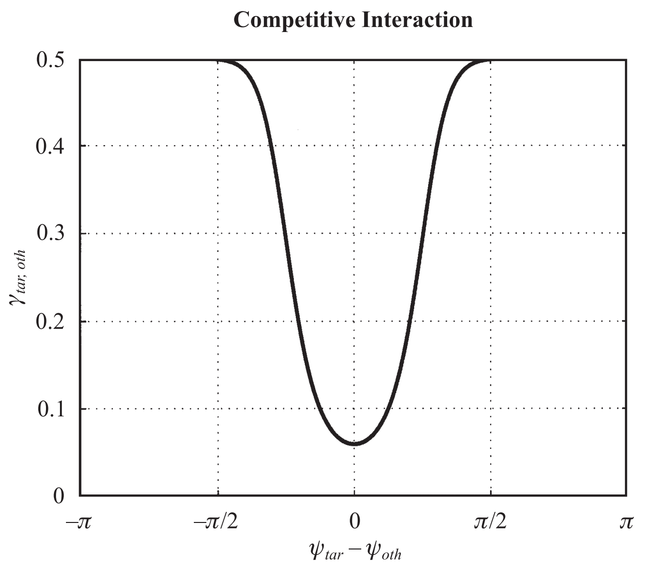

Fig. 9. Competitive interaction between *target* and *other,* $\gamma_{tar,oth} = \gamma_{oth,tar}$, for $b_1 = 0.25$, $b_2 = 4.0$, and $b_3 = 3.0$. Competition is maximum except when the target and the other agent lie in approximately the same direction, that is, when $\psi_{tar} - \psi_{oth}$ is near zero.

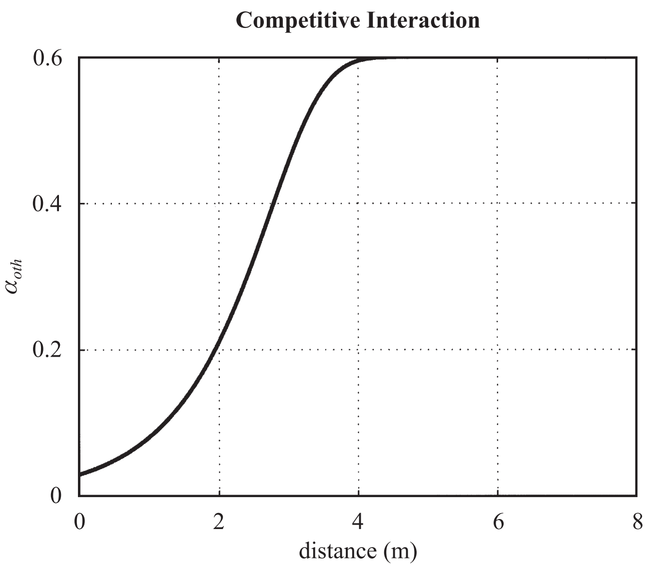

Fig. 10. The competitive advantage of other, for $a_{oth} = 0.6$, $d_1 = 3$, and $V_{oth} = 1$. As distance to the other agent increases beyond d_1, competitive advantage, α_{oth}, increases beyond competitive interaction from target (see Fig. 8).

4.2. Competitive Interaction

First, 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, in this section, we design the competitive interaction between our new constraint and the two existing constraints.

The new constraint is that of staying near the other agent. As described above, this behavior is modeled as a global attractor centered in the direction of the other agent. The potential difficulty lies in the composition of target and other as shown in Figure 8. Because both constraints are modeled as global attractors, their respective contributions will always be nonindependent, 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:

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

Equation (21) is graphed in Figure 9. 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 determines the maximum level of competitive interaction.

4.3. Competitive Advantage

Next we consider the competitive advantage of *other*. The constraint other is similar to that of target, but because we simply want the agents to remain near one another, we deactivate *other* when the agents are close enough. Thus we choose

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

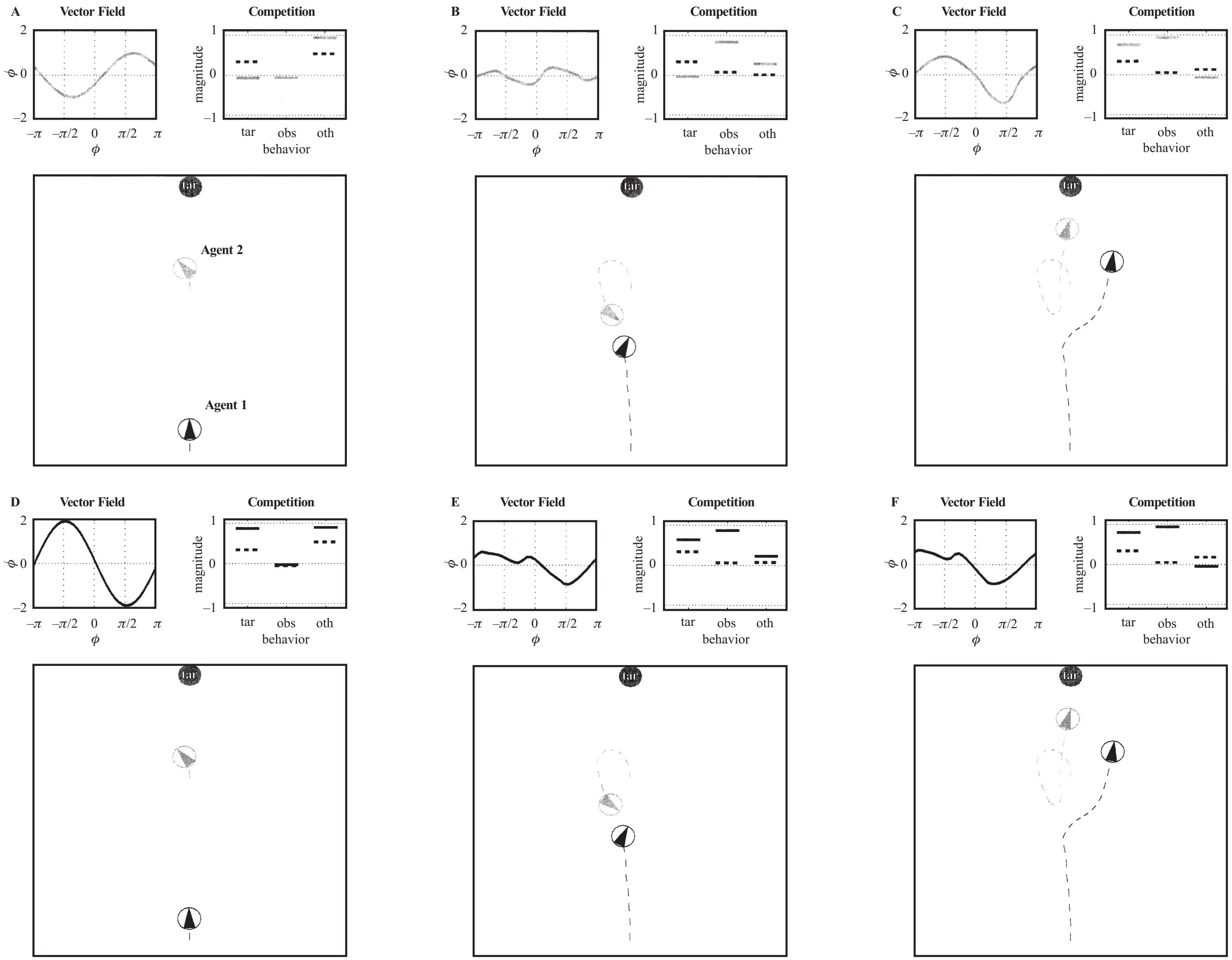

Fig. 11. An example of cooperative navigation. Each panel shows the current configuration (bottom), and the corresponding vector field (top left). Heading direction is plotted so that $\phi = 0$ corresponds to the agent's current heading direction. The current competitive situation is also shown (top right): competitive advantage, α_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 ((a) – (c)) shows vector field and competition for Agent 2 (above in configuration); the bottom row of panels ((d) – (f)) shows vector field and competition for Agent 1 (below in configuration). Agent 2 reverses course to join Agent 1 ((a) and (d)); the agents meet one another, and Agent 2 reverses course again ((b) and (e)); and the agents move toward the target together ((c) and (f)).

Here, r_{oth} is the distance to the other agent, and the constant d_1 determines how close we wish the agents to be (Fig. 10.). For simplicity, we choose $d_1 = d_0 + 1$, where d_0 is the distance at which the agents begin to consider one another as obstacles to be avoided (see eq. (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 implements an enclosure detector similar to that defined for target in eq. (16).

4.4. Examples

In this section, we examine two examples of the cooperative behavior. First, we look at the situation depicted in Figures 8a and 8b, 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 2 (above in the configuration; Figs.8a–8c) and Agent 1 (below in the configuration; Figs. 8d–8f). Notice that initially, Agent 2 must decide whether to move toward the target or toward Agent 1. In Figure 11a, Agent 2 begins to come about because *other* wins the competition with *target*. Below, Agent 1's *other* and *target* constraints are both activated, because these two constraints are consistent with one another, according to eq. (21), and thus they do not compete. In Figure 11b, we see that the agents have moved close to one another. Agent 2 is deactivating its other constraint, while obstacles has become activated, and the agent begins to veer away from collision with Agent 1. Agent 1 has activated all constraints, because there is some competition among all three, yet all competitive advantages are stronger than all competitive interactions. Finally, in Figure 11c, Agent 2 has reversed course yet again, this time accompanying Agent 1 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 two agents navigating cooperatively without centralized control and with no communication, save that they know one another's position. Each agent generates a different behavioral sequence using identical dynamical equations, because each is able to make independent decisions. Different decisions 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 nontrivial class of behavior that was identified in the stability analysis of Section 4.1.

Next, we look at a new situation. In Figure 12a, 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. The behavioral dynamics and competition are displayed for Agent 1 only, due to the symmetry of the situation. In Figure 12a, as they move forward both *target* and *obstacles* are active; *other* is not active, because the agents are near one another. In Figure 12b, the agents separate. Notice in Figure 12b that the agents are quite far apart, and the competitive advantage of *other* is 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. In Figure 12c, the two agents round the wedge, and 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 12d shows the situation after the agents have come together and resumed their original course.

This sequence displays some interesting properties dynamical decision making. 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 Figure 12b, the two agents are 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 is due to hysteresis, a simple kind of memory that determines system performance according to its past history.

In summary, these two examples 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.

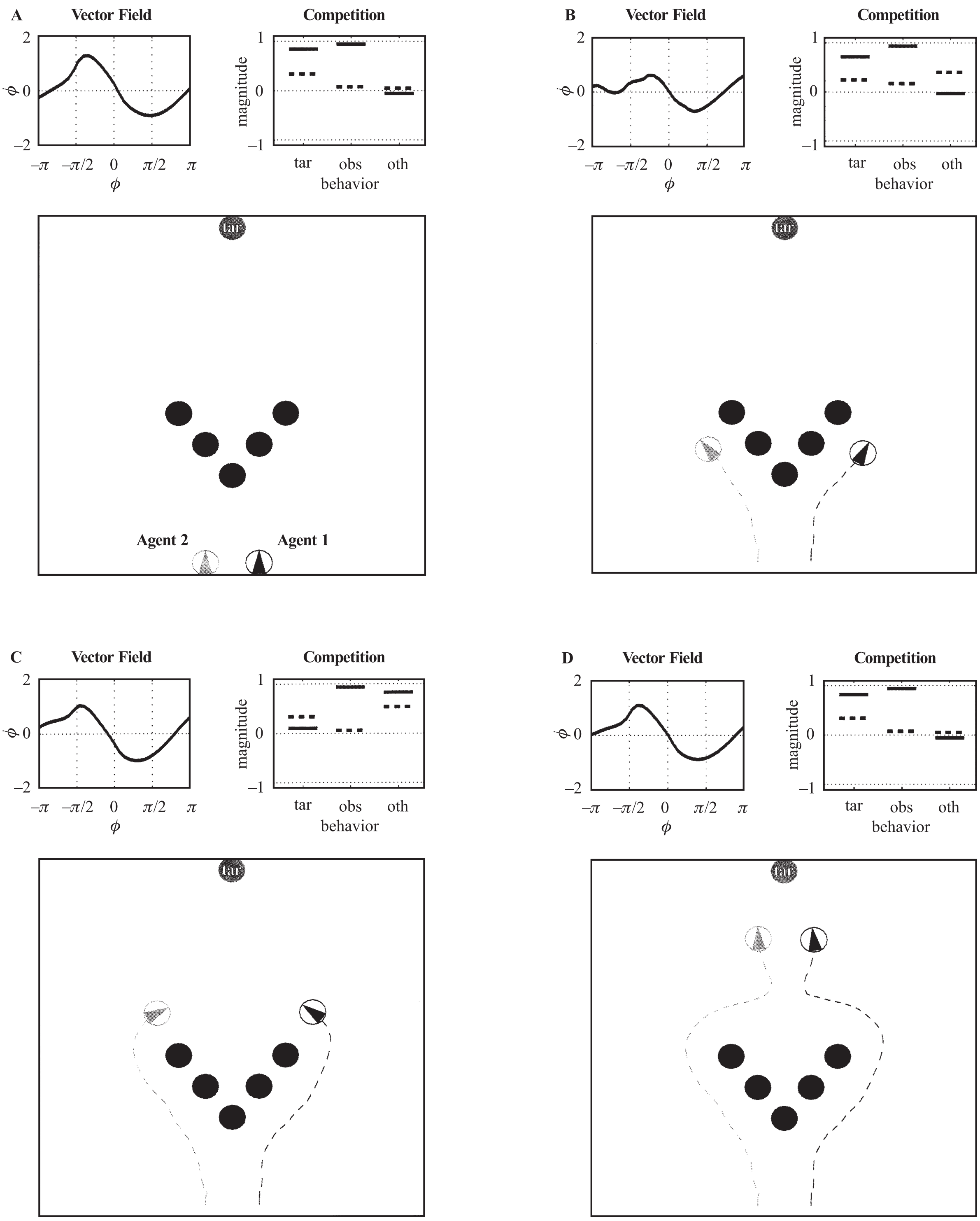

Fig. 12. A second example of cooperative navigation. Each panel shows the current configuration (bottom), and the corresponding vector field (top left). Heading direction is plotted so that ϕ =0 corresponds to the agent's current heading direction. The current competitive situation is also shown (top right): competitive advantage, α_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). The two agents navigate toward the target together (a); they are driven apart by obstacles, but continue toward the goal (b). The agents pass the obstacles, and move toward one another (c), and continue toward the target together (d).

5. Discussion

We set out to examine the issues of representation for the interaction of an agent with its environment in navigation-like tasks. We have taken care to separate the physical and geometrical models of the agent, its environment, and its overt behavior, from the task-oriented and context-dependent constraints that determine which action is appropriate in any specific situation. Yet we have adopted a physics-based model, that is, the so-called dynamical systems approach (Schöner and Dose 1992; Schöner, Dose, and Engels 1995), at both levels of description. At the level of overt action, a dynamics of behavior is defined over a space of behavioral variables. Task constraints contribute to shape a vector field that governs robot actions. At the level of decision making, a dynamics is defined over a space of task constraints. Task constraints compete for representation at the behavioral level, modeling decisions about which actions to perform, based on the current context. Our contribution has been to show how complex combinations of task constraints can be dealt with by adding a dynamic layer that is capable of managing task complexity at the behavioral level.

Our main concern has been to investigate scaling of the dynamical systems approach when behavioral requirements extend beyond simple navigation. However, it is also important to understand the relationship of this approach to other approaches for navigation-like tasks. Most relevant is the so-called potential field approach to robotic navigation. The mathematical concepts underlying the potential field approach link it closely with the dynamical systems approach. The main difference between the two is the state space over which the system dynamics is defined. According to the potential field approach, the state space is taken to be the configuration space, whereas in the dynamical systems approach, the state space is the space of robot behaviors. The important implication of this difference is that the dynamical systems approach models desired *behaviors* as stable fixed points of a dynamical system, whereas the potential field approach models the target *location* as a fixed point, and robot behavior corresponds to a transient of the associated dynamical system. This has important consequences for many aspects of system behavior.

Nevertheless, both the potential field and the dynamical systems approach intelligently blend multiple sources of information about the environment into a nonlinear dynamical system that generates smooth trajectories for an autonomous agent. As such, the dynamical approach to decision making investigated in the paper is potentially applicable to scaling of either method as the number of task constraints grows. The basic idea is to allow the behavior-generating system to intelligently blend information from multiple sources into task-appropriate behavior when possible, and to arbitrate between and sequence behaviors when blending is not possible. The details of the design will change, depending upon the nature of the system that generates overt behavior, because different dynamic approaches to behavior generation have different limitations and constraints. However, the underlying methodology of system design that we have proposed should remain unchanged.

The current approach has three major implications for the dynamic systems approach to path 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 nonindependent contributions to the vector-field dynamics are combined by superposition. 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 is possible to specify tasks that are more complex than simple navigation. This ability arises from the ability to determine which constraints should contribute to the behavioral dynamics, that is, to decide which behavior is appropriate, in any given situation. Each behavior arises as an asymptotically stable fixed point of the competitive dynamics, specifying a qualitatively unique combination of task constraints. This provides a number of interesting properties. First, 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. Second, 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. Third, 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.

As we have seen, in attempting to satisfy a complex set of behavioral requirements, each agent executes a sequence of behaviors. Here the sequences are not programmed explicitly; rather, they arise as the competitive dynamics arbitrates between different behaviors. The decision to execute a new behavior is modeled as a bifurcation in the competitive dynamics, which arises as the competition parameters adapt to the surroundings; thus sequences are generated opportunistically. However, it is also possible to program behavioral sequences explicitly. In a closely related approach, Steinhage and Schöner (1998) use a similar type of dynamic competition to program complex sequences similar to those that can be expressed as finite-state machines. It seems likely that it will prove useful to include both types of sequence generation in autonomous systems, as task and environmental complexity grows.

Finally, the competitive-dynamics solution scales nicely to the design of complex systems. We have shown that to design such a system, one must make n^2 design decisions, designing competitive- advantage functions for each individual behavior, and designing competitive-interaction functions between each pair of behaviors. However, the same analysis reveals that an n-constraint system gives rise to 2^n unique behaviors. The analysis also implies a unique type of modularity. One does not design the system hierarchically, yet one can add task constraints without disturbing the operation of the previously designed system. To do this, the designer considers how a new task constraint will interact with each of the existing constraints, but need not reconsider the structure of the previous system. In this paper, three task constraints sufficed to construct 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.

While the theoretical relationship between discrete automata (traditionally used to model decision making) and dynamical systems (traditionally used for control) has been studied by others (such as Brockett 1994), we have investigated the plausibility of developing a design methodology for robotic planning systems using dynamical systems in a way that scales to the modeling of complex systems of behavioral requirements. We have shown that, for a single robot or a pair of cooperative robots, decision making, path planning, and control can be modeled entirely using continuous nonlinear differential equations. We hope to combine the advantages of the dynamical system approach (stability, flexibility, robustness, etc.) with the ability to make decisions and carry out complex sequences of behavior to achieve well-defined goals. We have seen that the dynamical decision-making approach to managing task complexity offers a number of advantages in this regard, including scalability. The primary input from the designer is to set the priorities between competing behaviors. This is not too surprising, considering that these priorities depend on the task, the situation, and the context. Thus, both the physical aspects of control, and the ability to make discrete decisions about switching control strategies, can be successfully captured within the framework of continuous differential equations.

Acknowledgments

This work was in part supported by NSF grants IRI93-07126 and SBR89-20230, as well as U.S. Army grant DAAH04-96-1-0007 and ARPA grant N00014-92-J-1647. It was carried out during a six-month visit to the GRASP Laboratory by the second author. The authors wish to thank Tjeerd Dijkstra and Gregor Schöner for many insightful discussions, as well as Andrew Hicks, Gregor Schöner, and two anonymous reviewers for thoughtful comments on an earlier draft of this paper.

References

Adams, J., Bajcsy, R., Ko□eck'a, J., Kumar, V., Mandelbam, R., Mintz, M., Paul, R., Wang, C.-C., Yamamoto, Y., and Yun, X. 1995. Cooperative material handling by human and robotic agents: Module development and system synthesis. *Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Sys. IEEE* pp. 195–197.

Amari, A. 1977. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biol. Cybernet.* 27:77–87.

Brockett, R. W. 1994. Dynamical systems and their associated automata. In *Systems and Networks: Mathematical Theory and Applications,* vol. 77, Berlin: Akademi-Verlag, pp. 49–69.

Brooks, R. 1989. A robot that walks: Emergent behaviours from a carefully evolted network. *Neural Computation* 1(2):253–262.

Desai, J., Wang, C., Zefran, M., and Kumar, V. 1996. Motion planning for multiple mobile manipulators. *Proc. of the Int. Conf. on Robot. and Automat.* Madison, WI, Omnipress, pp. 477-484.

Engels, C., and Schöner, G. 1995. Dynamic field endow behavior-based robots with representations. *Robot. Autonomous Sys.* 14:55–77.

Guckenheimer, J., and Holmes, P. 1983. *Nonlinear Oscillations, Dynamical Systems and Bifurcations of Vector Fields,* Springer-Verlag, vol. 42 of *Applied Mathematical Sciences,* New York: Springer-Verlag.

Jacobs, R. A., Jordan, M. I., and Barto, A. 1990. Task decomposition through competition in a modular architecture: The what and where in vision tasks. *Cognitive Science* 15:219–250.

Jordan, M. I., and Jacobs, R. A. 1994. Hierarchical mixtures of experts and the em algorithm. *Neural Computation* 6:181–214.

Khatib, O. 1986. Real time avoidance for manipulators and mobile robots. *Int. J. Robot. Res.* 5(1):90–98.

Latombe, J. C. 1991. Robot Motion Planning. Boston, Kluwer Academic.

Lozano-Perez, T., and Wesley, W. A. 1979. An algorithm for planning collision-free paths among polyhedral obstacles. *Comm. ACM* 22(10).

Lumelsky, V. J., and Stepanov, A. A. 1987. Path-planning strategies for a point mobile automaton moving amidst unknow obstacles of arbitrary shape. *Algorithmica* 2:403–430.

Mallot, H., Bülthoff, H., Little, L., and Bohrer, S. 1991. Inverse-perspective mapping simplifies optical flow computation and obstacle detection. *Biol. Cybernet.* 64: 177–185.

Murray, R. M., Li, Z., and Sastry, S. S. 1991. *Robotic Manipulation.* New York: CRC Press.

Neven, H., and Schöner, G. 1996. Neural dynamics parametrically controlled by image correlations organize robot navigation. *Biol. Cybernet.* 75:293–301.

Perko, I. 1991. *Differential Equations and Dynamical Systems:* Texts in Applied Mathematics. Berlin: Springer-Verlag.

Rimon, E., and Koditschek, D. E. 1993. Exact robot navigation using artificial potential functions. *IEEE Trans. Robot. Automat.* 8(5):501–519.

Schöner, G., and Dose, M. 1992. A dynamic systems approach to task-level systems integration used to plan and control autonomous vehicle motion. *Robot. Autonomous Sys.* 10:253–267.

Schöner, G., Dose, M., and Engels, C. 1995. Dynamics of behaviour: Theory and applications for autonomous robot architectures. *Robot. Autonomous Sys.* 16(2):213–246.

Schöner, G., and Kelso, J. 1988. Dynamic pattern generation in behavioral and neural systems. *Science* 239:1513–1520.

Schwarz, J. T., and Sharir, M. 1983. 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.

Steinhage, A., and Schöner, G. 1998. *Dynamical systems for the behavioral organization of autonomous robot navigation.* In P. S. Schenker and G. T. Mc Kee (Eds.), Sensor Fusion and Decentralized Control in Robotic Systems: Proceeding of SPIE. pp. 160–180. Bellingham, WA: SPIE Publishing.

Venetianer, P., Large, E. W., and Bajcsy, R. 1997. Methodology for evaluation of task performance in robotic systems: A case study in vision-based localization. *Machine Vision Appl.* 9:304–320.