Using Competition between Task Constraints to Scale the Dynamical Systems Approach to Planning and Control

E. Large*, H.I. Christensen† and R. Bajcsy*

* GRASP Laboratory
University of Pennsylvania
3401 Walnut Street
Philadelphia, PA 19104
USA

†Lab of Image Analysis[1]
Aalborg University
Fr. Bajers Vej 7
DK-9220 Aalborg Ø
Denmark

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. Situations arise, however, in which superposition of contributions can lead to the formation of spurious attractors and cause related problems. To rectify such problems we define a task space dynamics that provides competition between a set of constraints. We find that competition among task constraints is able to deal with problems that arise when combining constraint contributions. We show how competition among constraints enables agents to execute sequences of behaviors that satisfy task requirements.

Contents

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 robot actions for the navigation task. In spite of this effort, however, the issue of what is the best space in which to represent robot behaviors remains an open question. Geometric representations (e.g. [?, ?]) model the geometry of the agent and the external environment. The difficulty with this approach is that it is too static. Configuration space representations [?, ?] 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 [?, ?] 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. Arkin[?] has described an approach where the potential field is evaluated locally, which reduces the computational complexity, but the problem of local (spurious) minima implies that the method may not be successful. To handle spurious minima Connolly and Grupen [?] have described how harmonic functions may be used. The method requires full propagation over the configuration space, however, which makes it computationally expensive and ill-suited for application in dynamically changing environments. A review of related aproaches can be found in [?].

An alternative is to define a local representation (e.g. [?]) and/or a representation whose dimensions correspond to robot behavior (e.g. [?]). The so-called *dynamical systems* approach for robot path planning and control uses such a local behavior-based representation [?, ?]. 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 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.

The mathematical concepts forming basis of this approach are closely related to those of the potential field approach. It is important to note, however, that the 'dynamical systems' approach to navigation models desired *behaviors* as stable fixed points of a dynamical system. By contrast, the 'potential field' approach to navigation 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. For an in-depth analysis of the difference between the 'potential field' and 'dynamical systems' approaches to navigation, see [?].

Our work, presented in this paper, has been motived by the dynamical systems approach because it is suitable for modeling the dynamics of the robot's interaction with its environment while carrying out navigation tasks. 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. [?]), 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 [?]. 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. Such behaviors may include 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. [**?**, **?**]), 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 and obstacles) 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*). Competition among task constraints produces simple 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. Finally the implications of using behavioral dynamics is discussed.

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 [?]. 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 [?].

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 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 [?].

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 (Figure 1A) and to-be-avoided behaviors as repellors (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),

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

where ϕ is agent 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.

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

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,

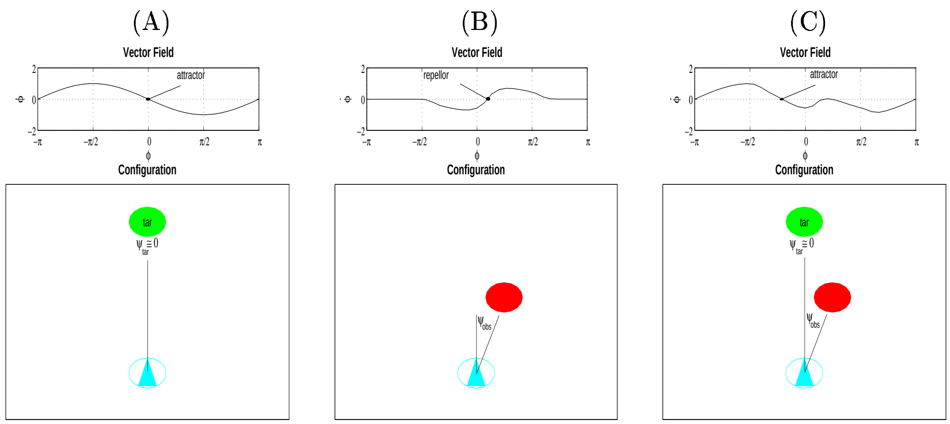

Figure 1: Task Constraints and Behavioral Dynamics. (A) An agent, a target, and the corresponding vector field. The *target* constraint is expressed as a heading direction (zero corresponds to the current heading of the agent). The desired behavior of heading toward the target is expressed as an attractor (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.

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

a second limits the angular range of the contribution,

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

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

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

The parameters to these function are, ϕ, the heading direction of the agent, ψ_i, the direction to obstacle i, $\Delta\psi_i$, the angular range subtended by the obstacle, R_i, the radius of the obstacle, R_{agent}, the size (radius) of the agent, and σ, 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 [?]. 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 planning dynamics, as illustrated in Figure 1C.

$$\dot{\phi} = F_{tar} + F_{obs} + noise \tag{8}$$

Because certain constraints are modeled as repellors, the planning dynamics is augmented by a stochastic term that guarantees escape from unstable fixed points (repellors). 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 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 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. Consider the two situations depicted in Figure 2, for example. 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 repellor 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.

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 [?] 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 [?], 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 [?, ?]. 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.

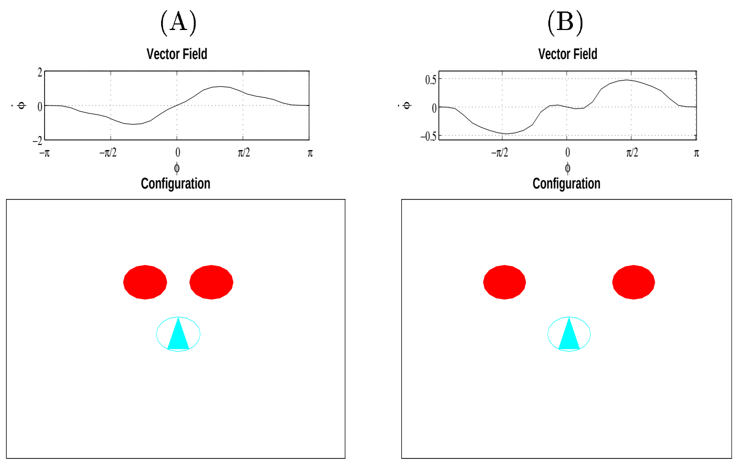

Figure 2: Dependent and Independent Constraints. (A) Two obstacles configured so that there is not enough space between them for the agent to pass through. These constraints are dependent, and superposition of their contributions to the vector field creates a repellor at their average heading direction, 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 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.

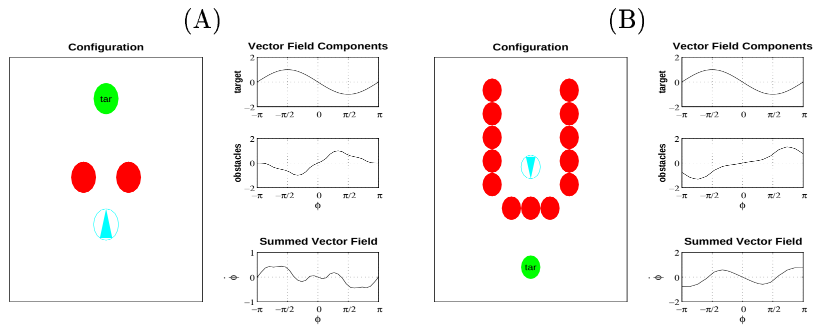

Figure 3: Spurious Attractors. (A) Two obstacles dead ahead provide almost, but not quite enough space for the agent to pass between them. The *obstacles* contribution to the dynamics reveals a shallow attractor. Yet when the target also lies straight ahead, its attractor contribution, combined additively 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 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 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. 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.

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

The state space of this dynamical system corresponds to the set of task constraints. In thois paper, we will consider a two dimensional system, with state vector $[w_{tar}, w_{obs}]$. The parameters, α_i and $\gamma_{j,i}$, are referred to as *competitive advantage* and *competitive interaction*, respectively. In our application, competitive advantage, α_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. A stability analysis for the competitive dynamics is described below.

Equation (10) describes a competitive dynamics similar to that proposed in [**?**] 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 [**?**]. Implementing competition in an Amari field solved this problem, but at the expense of simulating a two dimensional integrodifferential equation. Our use of competitive dynamics (i.e. Equation (10)) will not be vulnerable to the correspondence problem because we use competition to determine the weighting of a fixed set of behavioral constraints. Thus, it is not necessary to resort to more computationally expensive means. 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 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 α_i and $\gamma_{j,i}$ determine the resultant weighting of component behaviors. In the second stage we identify situations in which the two constraints, *target* and *obstacles* are incompatible. This leads to the design of functional forms that tie competitive interactions, i.e. the $\gamma_{j,i}$, to specific situations. Third, 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 (e.g. [?]) 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 repellor 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 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 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.

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 repellor 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

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	± 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	$\alpha_{tar} > \gamma_{obs,tar}$

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 that 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 simple two constraint system, this type of solution is appropriate, in general, when the environmental situation is ambiguous.

Finally, the point $(w_{tar}, w_{obs}) = (A_{tar,obs}, A_{obs,tar})$ 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 [**?**]. 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

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

If there is no competition between constraints, $\gamma_{i,j} = 0 \ \forall \ i, j$, and both constraints are activated at full strength. In this case the resulting behavioral dynamics reduces to that described in [**?**]. 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 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 Figure 3), unwanted consequences may result, because the two contributions are 1) non-independent and 2) contradictory. We design "fixed point detectors" that describe the location and stability of the fixed points for each contribution to the behavioral dynamics. We then use these functions to define competitive interaction between the two task constraints.

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 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 $c_1 > 0$ determines the rate of fall off, allowing the specification of a safety margin around the attractors and repellors if necessary. P_{tar} 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 *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) will 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.

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

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 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 Equation (14), for the situation depicted in Figure 3A, is shown in Figure 4. Note that it is strongly peaked at the point of attractor-repellor 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, α_{tar}, for *target*.

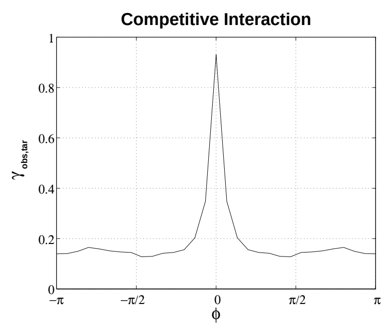

Figure 4: Competitive Interaction between *obstacles* and *target* for the spurious attractor example of Figure 3A.

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 small constant value, allowing this constraint 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 *obstacle* should compete, i.e. when attractor and repellor contribution would "collide". From the stability analysis we know how to pick relative values of α_i and $\gamma_{j,i}$ such 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, the we sum the D_{obs_i}. We then limit the maximum value of the α_{obs}, resulting in the following function for competitive advantage.

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

This completes our design.

3.4 Examples

In section 2.3 we saw two situations in which spurious attractors were created by 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 (shown, for this situation, 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.

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 (of the type shown in 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:

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

Here, a_{tar} is the competitive advantage for *target*, as described above. α_{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:

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

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

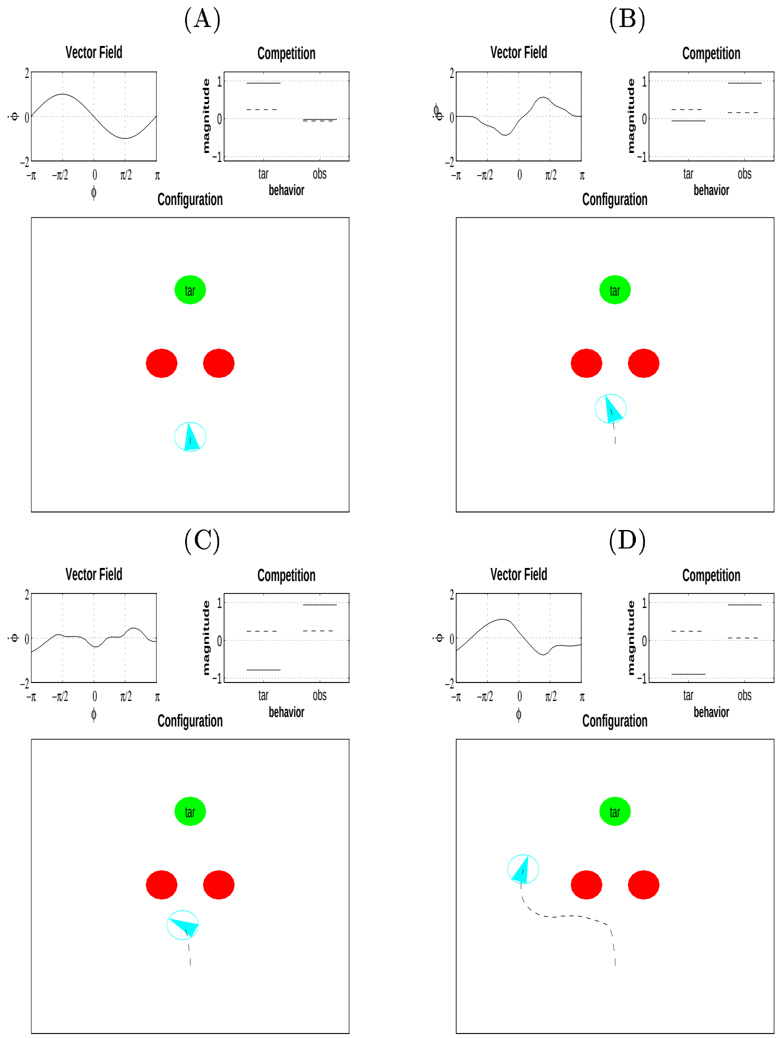

Figure 5: Competition avoids the creation of a spurious attractor. Each panel shows the current configuration (bottom), and the corresponding vector field (top left). Heading direction is plotted so that $\phi = 0$ corresponds to the agent's current heading direction. The current competitive situation is also shown (top right): Competitive advantage, α_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.

nature of behavioral constraints, temporarily rules out a particular behavioral contribution. For the examples below we choose $a_{tar} = 0.4$.

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 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 regardless of active competition from *obstacles*. Next, Figure 6C 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 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 repellor. 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.

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, design of competitive interaction functions, design of competitive advantage functions. This same methodology can be used to design systems obeying a larger number of task constraints (e.g. [**?**]).

4 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 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-

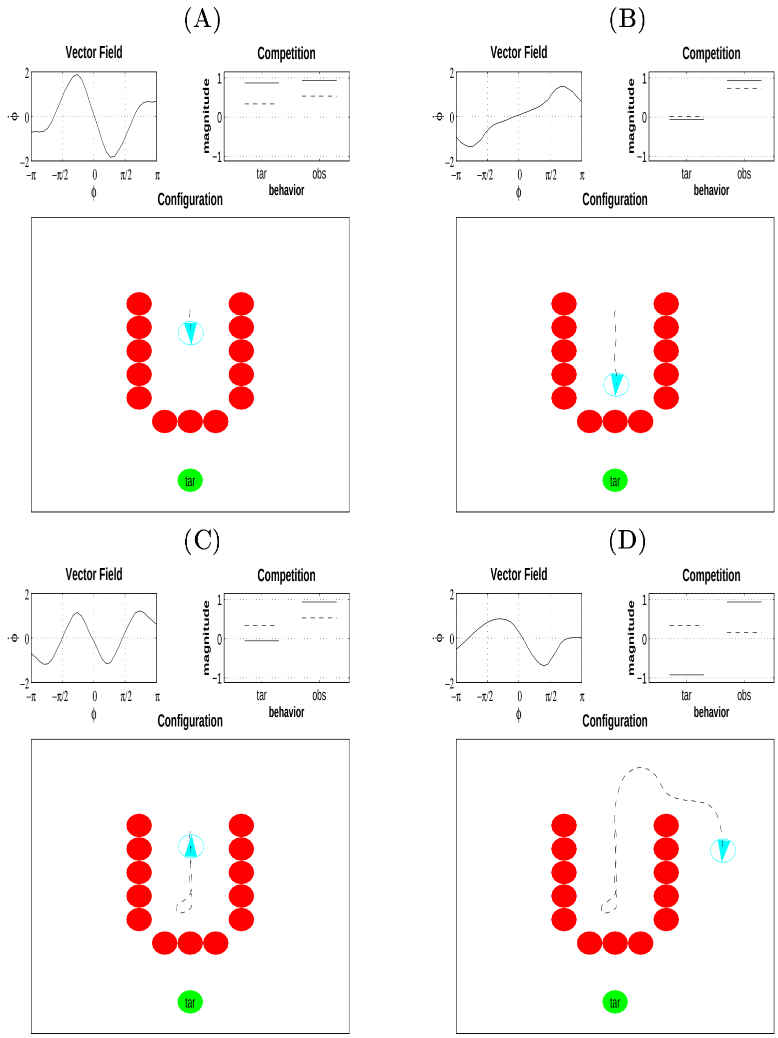

Figure 6: The agent successfully negotiating the hallway trap. Each panel shows the current configuration (bottom), and the corresponding vector field (top left). Heading direction is plotted so that $\phi = 0$ corresponds to the agent's current heading direction. The current competitive situation is also shown (top right): Competitive advantage, α_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, α_{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.

called *dynamical systems* approach [**?**, **?**], 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 repellors of the behavioral dynamics. Attractors and repellors are combined additively into a vector field that governs the agent's behavior. The problem we encountered was that in certain situations superposition gives rise to spurious attractors. Our contribution has been to show how such problems 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 two 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 the formation of spurious attractors, 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* in tied to the environment through functions that implement heuristic judgments about when particular constraints are more or less critical.

Second, competitive dynamics determines which constraint(s) 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. [?]).

We have shown that, for a single robot 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. [?]). 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. These priorities depend on the task, situation, context, 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 task oriented, are captured within the framework of continuous differential equations.

5 Future work

The presented methodology has been implemented in MATLAB and only been tested on a rather limited set of examples. A simple interface to a LABMATE robot has also been provided, but space has not permitted presentation of any real-world examples. Future work will emphasize evaluation of the methodology in the context of a real platform, with extensive tests related to the complexity of the environment.

In this paper it has been described how combinations of two behaviors can be designed to eliminate spurious minima. Future work will also address an extension of the methodology to handle combinations of an arbitrary number of task constraints. The approach presented here can be extended to handle a large number of interacting behaviors and the theoretical methods for doing this are straight-forward. Initial work on such an extension has been described in [?].

6 Bibliography