Scaling the Dynamic Approach to Autonomous Path Planning: Planning Horizon Dynamics*

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

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

† Centre for Autonomous Systems
Dept. Numerical Analysis and Computing Science
Kungliga Tekniska H gskolan
S-100 44 Stockholm
Sweden

Abstract

In the *dynamical systems* approach to robot path planning both sensed and remembered information contribute to shape a nonlinear vector field that governs the behavior of an autonomous agent. Such systems perform well with partial knowledge of the environment and in dynamically changing environments. Nevertheless, it is a local heuristic approach to path planning, and it is not guaranteed to find existing paths. We describe a method of adjusting the spatial resolution of the planner using a dynamical system that operates at a faster time scale than the planning dynamics. This improves the system's ability to utilize both sensed and remembered information, and to solve a larger range of problems without resorting to global path planning.

1 Introduction

Many approaches to path planning assume global knowledge of the environment (e.g. [Latombe, 1991; Khatib, 1986; Connolly and Grupen, 1993]). Global algorithms can guarantee solutions to path planning problems, i.e. they can guarantee that the agent will find a path to the target if one exists. They can also find paths that meet optimality conditions such as shortest path length, minimum energy consumption, and so forth. In many interesting situations, however, environment knowledge may be unavailable or incomplete [Lumelsky and Stepanov, 1987; Rimon and Koditschek, 1993]. For example, knowledge of the environment may need to be updated by the agent as task execution proceeds. Dynamic environments also violate the assumptions of global path planning. When targets and obstacles move freely about the world, global representations can quickly become obsolete.

In contrast to global approaches, behavior-based systems react to sensory input with minimal internal representation of the environment (e.g. [Brooks, 1989]). Systems are designed in terms of elementary actions, or behaviors, and sensory information feeds into movement control at a low level of processing in a way that is specific to each behavior. Purely reactive systems are not, however, able to guarantee the existence of path planning solutions. Thus, for navigation in cluttered environments, there remains a need for representation, where actions are determined not only based on immediate sensor readings, but also based on an internal model of the environment.

The *dynamical systems* approach strikes an interesting balance between the extremes of global path planning and purely reactive systems. The dynamic approach shares with behavior-based approaches the ability to control actions based upon sensory input. In addition, this formalization includes the ability to build internal representations of the environment [Engels and Schöner, 1995]. A memory representation makes information available so that the system can act on knowledge about the environment that is not currently registered by the system's sensors. Behavioral information, as derived from both sensed and remembered information, shapes a vector field that controls the behavior of an autonomous agent. As a result, such systems perform well without prior global knowledge of the environment and in dynamic environments.

It is important to point out, however, that dynamic planning represents a local approach. Although the existence of a partial environment map will allow such a system to outperform a purely reactive system, in complex environments the dynamic approach will fail to find existing solutions to path planning problems. It is these

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

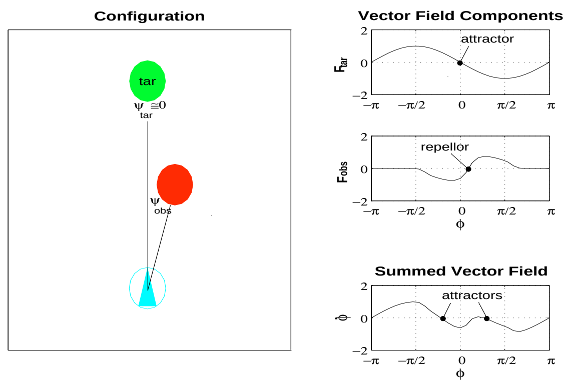

Figure 1 The dynamical systems approach to path planning.

situations that we investigate in this paper. In many of these situations the system can fail because it does not make use of its environment model in an appropriate way. We propose a method for improving the system's ability to utilize its environment model, thus solving a larger range of problems without resorting to global path planning.

2 The Dynamical Systems Approach to Path Planning

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]. 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 illustrated in Figure 1.

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. In this paper, we focus on a single behavioral dimension, heading direction. We assume that velocity is controlled by a dynamics similar to that described by [Neven and Schöner, in press]. For our one-dimensional system, the dynamics take the following form.

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

Task constraints define contributions to the vector field, $f(\phi)$, by modeling desired behaviors as attractors and to-be-avoided behaviors as repellors of the behavioral dynamics. The contributions of individual task constraints are combined additively into a single vector field, specifying the planning dynamics as shown in Figure 1.

$$\dot{\phi} = |w_{tar}|F_{tar} + |w_{obs}|F_{obs} + \cdots + noise \quad (2)$$

Because certain constraints are modeled as repellors, the planning dynamics is augmented by a stochastic term that guarantees escape from unstable fixed points (repellors). For the specific functional forms corresponding to task constraint contributions, see [Schöner and Dose, 1992].

To deal with spurious attractors as well as to incorporate additional task constraints, [Large *et al.*, 1997] further modified the strength of each contribution with a specific weight, w_i assigned to each type of task constraint. Weights are assigned through a competitive dynamics that determines the strength of each contribution depending upon the current situation.

$$\dot{w}_i = \alpha_i w_i(1 - w_i^2) - \sum_{j \neq i} \gamma_{j,i} w_j^2 w_i + noise \quad (3)$$

The state space of this dynamical system corresponds to the set of task constraints. The parameters, α_i and $\gamma_{j,i}$, are referred to as competitive advantage and competitive interaction, respectively. The task dynamics operates at a faster time scale than the planning dynamics. It models decision-making regarding which task constraints are applicable at a given time, and allows the system to generate sequences of behaviors.

2.1 Multiple Obstacles

According to this approach, avoidance of a single obstacle is modeled by adding a range-limited repellor to the vector field (see Figure 1), while avoidance of multiple obstacles is modeled by summing multiple range-limited repellor contributions.

$$F_{obs} = \sum_{i=1}^{n} R_{obs_i} e^{-\frac{r_i - R_i - R_{agent}}{d}} \quad (4)$$

Here, R_{obs_i} is a function that sets up a range limited repellor in the direction of obstacle i, and the exponential term scales the strength of the obstacle's contribution to the vector field. The parameters of the scaling term are the distance from the agent to the obstacle, r_i (center to center), the radius of the obstacle, R_i, and the radius of the agent R_{agent}. The parameter d determines the distance at which the agent begins to take an obstacle into account, and is set to a constant value, $d = d_0$. Obstacles that are very far from the agent do not affect the behavioral dynamics, whereas nearby obstacles affect the planning dynamics strongly. Thus, d

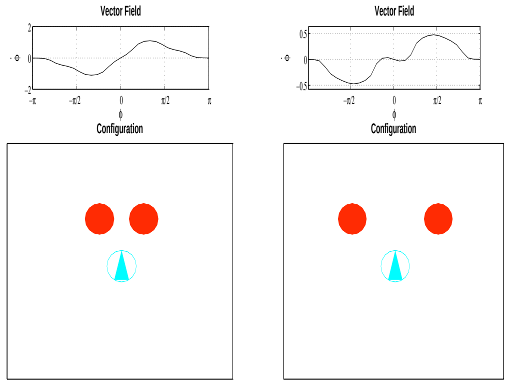

Figure 2 Integration of multiple obstacles in the vector field.

determines the *spatial resolution* of the planning dynamics.

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. In the left panel, 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 the right panel, 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, corresponding to the behavior of steering between the two obstacles.

2.2 Environment Maps

Memory can be implemented using an Amari-type 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 desired properties [Engels and Schöner, 1995; Schöner *et al.*, 1996]. A second function of the neural field is that it enables the system to store information about its environment in the form of a "cognitive map". As the system explores its environment, it is able to add to its knowledge. 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. For the purposes of this study, it is not necessary to consider the dynamics of the the Amari field, but simply to assume that information about the environment is represented as a non-overlapping grid of spatial locations.

3 When Local Planning Fails: Problem and Approach

Many path planning situations are well handled by the dynamic approach as described above. Consider the case depicted in Figure 3 (top), for example. The agent begins to move toward the target, but encounters a wall. It veers to the left, turns around, finds a doorway, and continues toward the target. A more difficult situation arises, however, when the doorway is removed, as shown in Figure 3 (middle). In this case the agent veers to the left, encounters the left wall and turns around. If the doorway were there, the agent would find it and successfully make its way toward the target location. In this case, however, the simple local strategy embodied in the planning dynamics fails, and the agent loops indefinitely. The reason is that d_0, the parameter that specifies the spatial scale at which planning occurs, is a small constant.[1] Thus the agent only takes into account a few obstacles at any given time. By the time it encounters the wall to the right, the agent has forgotten the wall to the left, causing the cycling behavior.

One possible approach to this problem is to allow the planner to take all three walls into account simultaneously by increasing the spatial scale, d. Figure 3 (bottom) shows the behavior in this situation. As expected, the agent avoids the configuration and successfully reaches the target. This example also shows the difficulty with this approach. The agent was unable to find the more efficient path through the door. This is because increasing the spatial scale decreases spatial resolution.

Our approach will be to dynamically adapt spatial resolution in a way that is appropriate to the agent's current environment. To do this, we must solve two problems. The first involves detecting when the agent should plan at a larger spatial scale. The second involves changing scales in a stable fashion, so that efficient planning is achieved. The following sections address these issues.

3.1 The Homeward Component

We detect cycling behavior by computing an instantaneous estimate of homing behavior,

$$h = (v/v_{max})\cos(\phi - \psi_{tar}) \tag{5}$$

[1] In these examples the spatial scale, d_0, was set to two meters, roughly corresponding to the visual capabilities of the actual robots in our laboratory. For comparison, the radius of individual obstacles in Figure 3 is one meter.

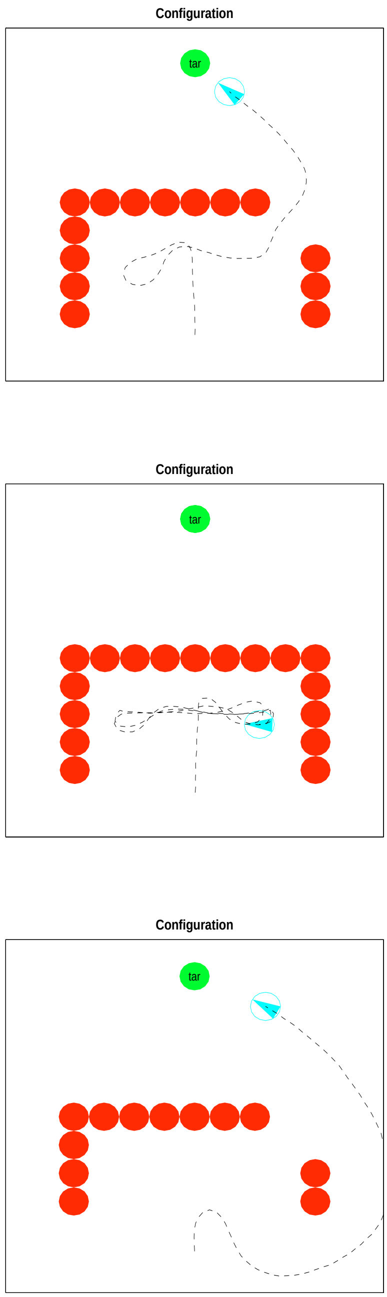

Figure 3 The effect of spatial resolution on planning dynamics. Top: Planning at a relatively small spatial scale, the agent arrives at its target. Middle: The agent gets stuck in a behavioral loop when the doorway is removed. Bottom: Planning on a larger spatial scale, the agent avoids the entire configuration of obstacles, missing the more efficient path through the doorway.

where v in the current velocity of the agent, v_{max} represents the agent's maximum velocity, ϕ is the agent's current heading, and ψ_{tar} is the heading direction of the current target. When the agent is heading toward the target at maximum velocity, $h = 1$, when the agent is heading away from the target at maximum velocity, $h = -1$, When the agent has zero velocity, $h = 0$, and so forth. We call this measure the *homeward component*, because of its close relationship with a circular statistic of the same name [Batschelet, 1981].

Next we smooth h using a simple linear dynamics to obtain a time-averaged estimate of homing behavior.

$$\dot{x_i} = \eta_i(h - x_i) \quad \eta_i > 0 \tag{6}$$

Here the x_i represent estimates of homing behavior at several different time scales. The η_i determine the time scales over which the point estimates are smoothed. When the agent enters a looping behavior the smoothed homeward components, x_i, will approach values of zero with time constants determined by the η_i. Figures 4 (top) and (bottom) show instantaneous and smoothed homeward components for the situations of Figures 3 (top) and (middle), respectively. In each figure, we show smoothing at three different time scales, x_1, x_2, and x_3. Note that in the top panel the smoothed homeward components retain relatively large values, indicating good homing behavior, while in the bottom panel each approaches zero, indicating that the agent is not making good progress toward the target.

3.2 Planning on Multiple Spatial Scales

Next, we address the question of how to adaptively change spatial resolution so that the planning dynamics will take into account information that is appropriate to its current environment. For reasons discussed below, we assume that several discrete levels of spatial resolution are required, and that at any given time the agent is to plan at a single level.

We achieve this end by utilizing the competitive dynamics introduced above in conjunction with constraint competition. In this case however, we reinterpret the meaning of the variables,

$$\dot{r_i} = \alpha_i r_i(1 - r_i^2) - \sum_{j \neq i} \gamma_{j,i} r_j^2 r_i + noise \tag{7}$$

where r_i indicates activation of planning level i. For a local stability analysis of the competitive dynamics, see [Large *et al.*, 1997]. In order instantiate the proper switching behavior, we define the competitive advantage, α_i, and competitive interaction, $\gamma_{j,i}$, parameters in such a way

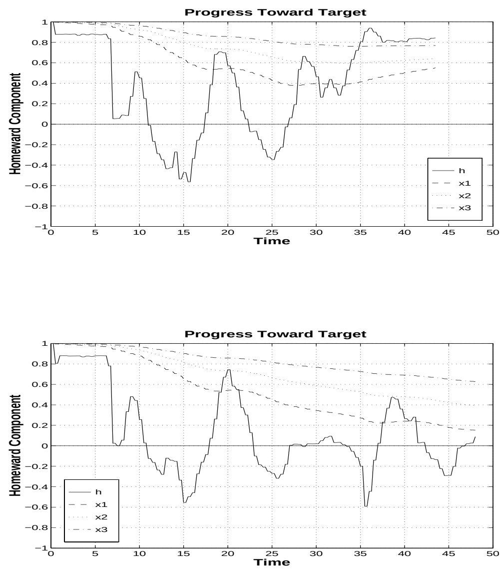

Figure 4 Top: The homeward component, h, and smoothed homeward components, x_i, for the situation shown in Figure 3 (top). Bottom: The homeward components for the situation shown in Figure 3 (middle). The smoothed values approach zero, reflecting the agent's cycling behavior.

that either exactly one planning level is active, or no planning levels are active (i.e., planning is at the base level). We do this by defining the parameters as follows.

$$\alpha_i = \alpha_{thresh} - x_i \tag{8}$$

$$\gamma_{j,i} = \begin{cases} \max_{i<j}(\alpha_i) + .2 & \text{if } j > i \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Here, α_{thresh} is a threshold level for the activation of a planning level. With these parameter definitions, the spatial scale dynamics has exactly $n+1$ fixed points, where n in the number of planning levels. This system is stable when exactly one planning level is active ($r_i = 1; r_j = 0, \forall j \neq i$) or when no planning levels are active ($r_i = 0, \forall i$). Intuitively, the competition between planning levels works as follows. When $x_i < \alpha_{thresh}$, planning level i attempts to activate itself. The $\gamma_{j,i}$ enforce competition among planning levels such that if two levels request activation, the level representing the larger spatial resolution will win.

We can then define spatial scale, used in Equation 4, as:

$$d = d_0 + d_1 r_1 + \cdots + d_n r_n. \tag{10}$$

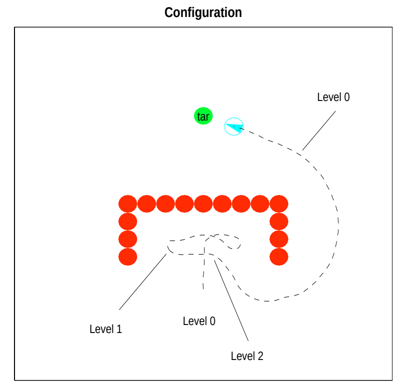

Figure 5 An agent escaping from an enclosure using multiple planning levels. Initially planning at level zero, the agent explores the enclosure. After making poor progress toward the target, the planning level jumps to level one, then level two before the agent escapes.

4 Examples

Next, we show two examples of a simulated agent using multiple levels of spatial resolution to escape from behavioral cycles. We used three planning levels, with parameter values $d_0 = 2$, $d_1 = 2$, $d_2 = 4$, $d_3 = 6$, $\alpha_{thresh} = 0.4$, $\eta_1 = .1$, $\eta_1 = .05$, and $\eta_1 = .025$.

First, we examine the example of Figure 3, in which the agent is enclosed by walls on three sides (Figure 5). Initially the spatial resolution of the planning dynamics is set to the base level, $d = d_0$. Upon encountering the long wall, the agent turns left. At the left wall the planning dynamics causes the agent to turn around. At about this point, x_1 falls below threshold, and planning level one is activated (marked in the figure), so $d = d_0 + d_1$. About half way to the right wall, planning level two is activated, disabling planning level 1. Once this occurs, the entire enclosure is represented in the planning dynamics simultaneously, and the agent escapes from the enclosure. As the smoothed homeward components increase above threshold, planning returns to the base level.

The second example shows a more difficult situation, a circular room with the only exit opposite the target. At first the agent moves toward the the target, and as it explores the space, planning levels one, and two are activated (Figure 6). Once level two is activated a single attractor is formed in the planning dynamics, corresponding to the exit. The agent makes its way through the door, and eventually to the target.

5 Discussion

We have described how to incorporate the notion of planning at multiple spatial scales into a system that plans paths through a two dimen-

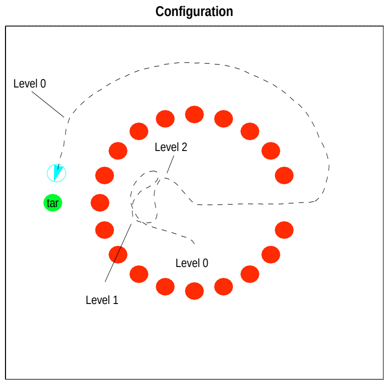

Figure 6 An agent escaping from a more difficult trap.

sional workspace using nonlinear dynamics. For our simulations, we assumed that the agent possessed a partial representation of the environment, and we addressed the issue of how best to use the information represented in its internal map.

This basic strategy is applicable in a variety of different settings. First of all, it seems natural to employ the planning horizon as a mechanism for selection of sensors. Many robots are equipped with a range of different sensors such as sonar, vision, laser, tactile, and so forth. Each of these sensors has a different range and resolution. Thus, dynamic selection of a discrete planning level may be used for selection of sensors in response to the need for spatial detail and/or range. For close range navigation either infrared sensors or vision modules such as the inverse perceptive mapping may be used. On our particular platform we employ ultra-sonic sonars and inverse-perspective mapping. Thus, inverse-perspective mapping may be used for close range navigation, while sonar may be used for longer range sensing.

There are several possible applications of the planning horizon dynamics beyond those presented in the paper. Map building presents one interesting possibility. Another possibility that we have not yet explored is planning at multiple levels simultaneously. This may be useful in situations where it is necessary to take both global and local constraints into account. We are currently experimenting with several applications of this approach using the mobile platforms in our laboratory.

References

[Amari, 1977] A. Amari. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27:77–87, 1977.

[Batschelet, 1981] E. Batschelet. *Circular Statistics in Biology*. Academic Press, Inc, 1981.

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

[Connolly and Grupen, 1993] C. I. Connolly and R. A. Grupen. The applications of harmonic functions to robotics. *Journal of Robotic Systems*, 10(7):931–946, October 1993.

[Engels and Schöner, 1995] C. Engels and G. Schöner. Dynamic fields endow behavior-based robos with representations. *Robotics and Autonomous Systems*, 14:55–77, 1995.

[Khatib, 1986] O. Khatib. Real-time avoidance for maniopulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98, 1986.

[Large *et al.*, 1997] E. W. Large, H. I. Christensen, and R. Bajcsy. Scaling the dynamic approach to path planning and control: Competition among behavioral constraints. Technical Report 409, GRASP Laboratory, University of Pennsylvania, 1997.

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

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

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

[Rimon and Koditschek, 1993] E. Rimon and D. E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–519, 1993.

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

[Schöner *et al.*, 1996] G. Schöner, M. Dose, and C. Engels. Dynamics of behaviour: theory and applications for autonomous robot architectures. *Robotics and Autonomous Systems*, 16(2–4):213 – 246, 1996.