Scaling the Dynamical Systems Approach to Path Planning*

Edward W. Large
GRASP Laboratory
University of Pennsylvania
3401 Walnut Street, Room 301C
Philadelphia, PA 19104-6228

Abstract

As robotic agents are called upon to perform in increasingly complex and naturalistic physical environments, it will be necessary to develop design strategies that enable the specification of complex tasks that can be carried out in flexible ways. This requires the both ability to react flexibly to dynamically changing environments as well as the ability to perform decision-making and sequencing of behavior. We describe an approach to decision-making and sequencing that utilizes the qualitative theory of dynamical systems to model stable decision-making in changing environments. The task dynamics methodology is considered as an approach to scaling the 'dynamical systems' approach to autonomous path planning.

1 Introduction

Understanding the integration of task directed decision-making with the generation of individual behaviors is fundamental to understanding autonomous agents. If the design of autonomous systems is to scale to the performance of complex tasks, agents must be able to reason in abstract terms about the *task* to be performed. This endeavor represents a nontrivial problem, and an adequate solution must display several properties. First, there is a need for a structured design methodology that supports analysis. Second, to make design manageable and flexible, it is essential that the methodology allow for decomposition of the problem into manageable subproblems. Finally, the approach should provide a powerful method of task description so that complex systems can specified in meaningful ways.

The "dynamical systems" approach for path planning and control [1, 2], is chosen as a point of departure for this study because it is especially appropriate for describing autonomous behavior in dynamic environments. In this approach a set of behavioral variables defines a state space in which a dynamics of robot behavior is described. The dimensions of the state space correspond to *behavioral variables*, such as heading direction and velocity. Planning and control governed by a dynamical system that generates a time course of the behavioral variables. The dynamical system is specified as a vector field that governs system behavior. The environment is also modeled at a behavioral level. The environment provides behavioral information that parameterizes a set of *task constraints*. Task constraints themselves are modeled as component forces that define attractors and repellors of a dynamical system. The contributions are combined into a single vector field by superposition. The resulting behavioral dynamics models the generation of behavior in a stable, yet flexible way.

This paper describes an extension to the dynamical systems approach, a task dynamics [3] that mediates complex autonomous performance. Specifically, a competitive dynamical system is used to model decision-making at the task level. The dimensions of the state space correspond to the set of task constraints. The state variables determine the relative weighting of component forces in the behavioral dynamics. Decision-making and sequencing are governed by a dynamical system operating at a faster time scale than the behavioral dynamics. The task dynamics enforces a competition among task constraints for representation at the behavioral level. The task is described by a set of parameters to the task dynamics. These parameters are tied to the environment and change as the robot's situation changes. Qualitatively different behaviors arise a fixed points of the task dynamics. Behavioral vector fields are synthesized as task execution proceeds. Bifurcations in the task dynamics give rise to behavior switching, modeling decision-making and sequencing.

After briefly reviewing the behavioral and task dynamics, an example of system performance is described, and the potential for scalability of such systems is considered.

*Work supported in part by: NSF Grants IRI93-07126 and SBR89-20230, as well as ARMY Grant DAAH04-96-1-0007 and ARPA Grant N00014-92-J-1647.

2 A Dynamic Systems Approach to Path Planning

2.1 Behavioral Dynamics

According to the dynamic approach, the behavior of an agent is modeled as a time course of behavioral variables generated by a dynamics that incorporates both planning and control knowledge. We focus on the dynamics of heading direction, assuming that velocity can be appropriately controlled (see, for example, [4]). For this one-dimensional system, the dynamics take the following form.

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

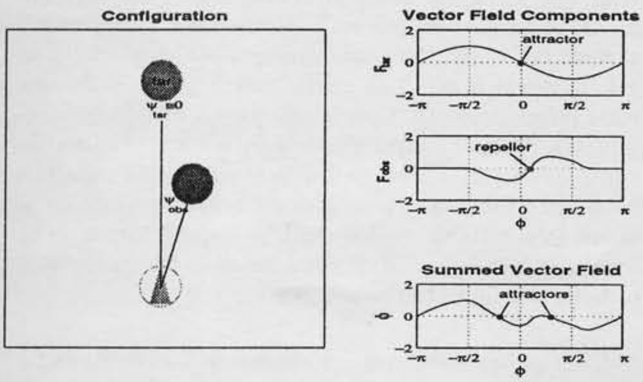

Figure 1: Task constraints and their contributions to the vector field. A target (attractor, F_{tar}); an obstacle (repellor, F_{obs}); their composition, $\dot{\phi} = F_{tar} + F_{obs}$.

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

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

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

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

The repellor corresponding to an individual obstacle is the product of three functions. R_{obs_i} sets up a generic repellor in the direction of the obstacle, W_{obs_i} limits the angular range, and D_{obs_i} scales the strength according to the obstacle's distance from the agent. Thus, the contribution of each obstacle is range limited. Details regarding the specific functional forms may be found in [1]. Multiple obstacles are handled by summing the contributions of individual obstacles.

$$F_{obs} = \sum_{i=1}^{n} F_{obs_i} \tag{4}$$

Finally, the contributions of individual task constraints (*target* and *other*) are combined additively into a single vector field as illustrated in Figure 1.

$$\dot{\phi} = F_{tar} + F_{obs} + \xi_t \tag{5}$$

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

2.2 Task Dynamics

To incorporate additional task constraints, the strength of each contribution with may be modified by a specific weight w_i assigned to each type of task constraint [3].

$$\dot{\phi} = |w_1|F_1 + |w_2|F_2 + \cdots + |w_n|F_n + \xi_t \tag{6}$$

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

$$\dot{w}_i = \alpha_i w_i(1 - w_i^2) - \sum_{j \neq i} \gamma_{j,i} w_j^2 w_i + \xi_t, \tag{7}$$

where i and j index the task constraints. The parameters to the competition dynamics are the α_i and the γ_{ji} referred to as *competitive advantage* and *competitive interaction*, respectively. The competitive advantage, α_i, of constraint i is determined by its applicability in the current situation, while the competitive interaction, $\gamma_{j,i}$, summarizes the degree to which constraint i is consistent or inconsistent with constraint j. Competition produces sequences of behavior that are generated reactively, in response to specific environmental situations. For a closely related system, capable of generating complex preprogrammed sequences, see [5].

2.3 An Example of Cooperative Navigation

As an example of the operation of such a system, consider a task that is slightly more complicated than autonomous navigation. Suppose there are two agents, and both agents must navigate autonomously toward a target location. In addition assume that the the task requires two agents to stay near one another as they make their way toward the target. Thus, the task is described using three constraints, and we will call the additional constraint *other*. Similarly to target seeking, we model the third constraint as a global attractor.

$$F_{oth} = -a \sin(\phi - \psi_{oth}) \tag{8}$$

Then the contribution of *other* is added to the composite vector.

$$\dot{\phi} = |w_{tar}|F_{tar} + |w_{obs}|F_{obs} + |w_{oth}|F_{oth} + noise \quad (9)$$

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

As an example of how this system works, consider the simulated situation illustrated in Figure 2. The agents A and B begin in the lower left, moving together and steering more or less directly toward the target. At this point the only active constraint is *target*. The agents encounter a wall, activating the *obstacle* constraint, so that one steers to the left and the other to the right. Had there been only a single small obstacle, this would have been a good decision, however, in this case agent B enters an enclosure. After a short time, agent B's *other* constraint is activated, deactivating *target*. Similarly, agent A begins to look for agent B. After first trying a bad direction, agent B loops around, and the two agents move along a wall until they meet once again. As they near one another, their *obstacle* constraints remains active (they are avoiding collision with one another), and they activate their *target* constraints. Finally, agent B follows A to the target. In this configuration, only A's *target* constraint is active. For B, however, all three constraints are active. This is because B is headed simultaneously toward its target and toward agent A, while it must also avoid collision with agent A.

This example shows several features of the competitive dynamics approach. First, the task dynamics is able to mediate between three interacting constraints that together describe the task of cooperative navigation. Second, depending upon each agent's situation, the agent synthesizes a specific behavior, through a weighting of task constraints in the behavioral dynamics, that is appropriate to the agent's current situation. Finally, this adaptive weighting of task constraints results in the generation of sequences of behaviors, where each behavior corresponds to a fixed point of the task dynamics. Activation and deactivation of specific constraints results from bifurcations in the task dynamics that arise as the agent's situation changes.

3 Scaling the Dynamic Systems Approach

The above model is formulated entirely within the qualitative theory of dynamical systems (for an introduction of dynamical systems theory, see [6]). Task-level decision-making is modeled as a competitive dynamical system operating at a faster time scale than the behavioral dynamics. The task dynamics determines the relative contribution of various constraints

Configuration

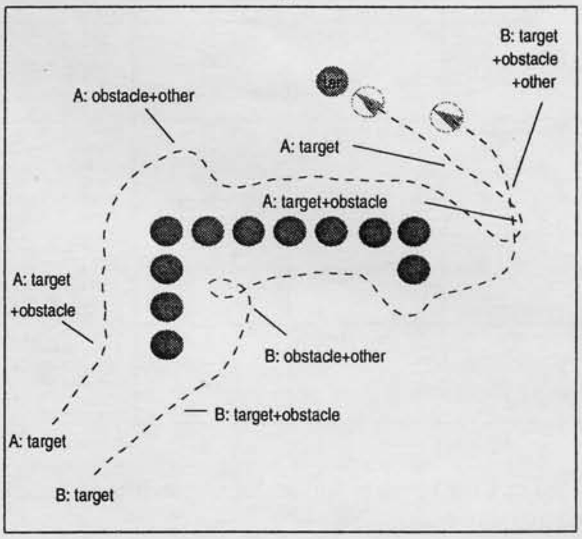

Figure 2: Cooperative navigation (I). Constraint weights (solid) and competitive advantages (broken) are shown in upper left panels for Agent 1 (bottom/right).

to the behavioral vector field. Fixed points of the task dynamics thus correspond to specific behaviors that are executed through the behavioral dynamics. Several questions remain to be addressed, however. How is such a system designed and analyzed? How well does the methodology scale to more complex systems? Is such a system modular? How does one describe the task to be performed? These issues are taken up in this section.

3.1 Stability Analysis

Because the task-level reasoning is described as a dynamical system, it supports an form of analysis that is directly relevant to the agent's ability to perform a complex task. In particular, a stability analysis (e.g. [6]) can be performed on the system described by Equation 7 assuming $\alpha_i, \gamma_{j,i} > 0$, for the case of three task constraints (for further description of this analysis, see [3]). Such an analysis reveals the qualitative behavior of the task dynamics, by enumerating the set of equilibrium points for the three-dimensional system and classifying each equilibrium point according to its stability. Because the stability of each equilibrium point changes depending upon the values of the parameters α_i and γ_{ji}, we also computed a set of stability conditions. The analysis (Table 1) reveals eight equilibrium points, corresponding to eight unique behaviors that can be generated by the agent, seven of which are stable.

One can group the attracting fixed points into three classes of stable solutions. The first class of solutions

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

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

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

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

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

If there is no competition between constraints, $\gamma_{i,j} = 0,\ \forall\, i, j$, both constraints are activated at full strength. The resulting behavioral dynamics reduces to that described by [1]. If there is some competition, both are still active, but at reduced levels. The final class of solutions consists of a "3-constraint averaging" solution, where all three contributions are active. This point is stable as long as $\alpha_i > \gamma_{j,i}$ for all $j \neq i$.

This methodology supports such an analysis due to the fact that the task-level decision-making is governed by a dynamical system. The analysis of the three constraint system revealed that any possible combination of constraints is available to the agent to synthesize a corresponding behavior depending upon the situation. In addition, the stability analysis tells us which relative parameter values correspond to which behaviors. This information that is valuable when designing a system to perform a specific task, as described below.

3.2 Modularity and Scalability

The above analysis points to some interesting properties of the task dynamics regarding modularity and scalability. The first thing to notice is that a complex conspiracy of competitive interactions is not required to activate or deactivate a constraint. A constraint is deactivated whenever interaction from any single competing constraint is great enough, conversely a constraint can be active only when it is consistent with all other active constraints. Further, this observation scales to systems of arbitrary numbers of constraints. This implies that one can add a constraint to a previous design without disturbing the previous system, simply by considering the interaction of a new constraint with each individual existing constraint. This feature provides the kind of modularity that is necessary for the incremental design (or evolution) of complex autonomous systems.

A somewhat more general observation about the form of the above analysis is that one designs the task dynamics by considering only pairs of constraints. This has important implications for the scalability of the approach. First, note that designing the system requires at most n^2 design decisions: $n^2 - n$ competitive interactions, plus n competitive advantages. Next, note that one can count the number of unique behaviors that arise in a n-constraint system. It is simply the number of ways to chose one active behavior, plus the number of ways to chose two active behaviors, and so on. In other words, the number of behaviors generated in such a system is:

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

Thus, in an n-constraint system, the task dynamics provides $2^n - 1$ unique behaviors, yet the system is polynomial in both design complexity and time complexity.

3.3 Task Description

Finally, the task is described by providing a set of functions that link situations in the environment to the parameters of the task dynamics. Relative parameter values determine how behaviors are composed, and the evolution of parameter values determine when bifurcations occur, thus how behaviors are sequenced. The above stability analysis describes how the relative values of the $\gamma_{j,i}$ and α_i determine which behavior arises from the task dynamics. This information can be used to design functions that tie competitive interaction and competitive advantage to situations in the environment.

First, consider the design of competitive interaction among task constraints. Competitive interaction must be able to capture situations in which different task constraints (modeled as attractors and repellors of the behavioral dynamics) are incompatible with one another. In general, when combining an attractor and a repellor one must guard against the formation of spurious attractors (or repellors). When combining like types of constraints (e.g. attractor and attractor) one must guard against unwanted constraint averaging. For details regarding such considerations see [3].

Consider the design of competitive interaction between *target* and *obstacle*. Whenever an attractor and a repellor collide, unwanted consequences may result, because the two contributions are non-independent and contradictory. Thus, one may construct a function that describes the competitive interaction between *obstacle* and *target* as:

$$\gamma_{obs,tar} = \frac{e^{-c_2 P_{tar} P_{obs}}}{e^{c_2}}, \qquad (12)$$

where P_{tar} and P_{obs} are "fixed point detectors" that capture the location and stability of the fixed points for each contribution to the behavioral dynamics (see [3]). This function is strongly peaked at the point of attractor-repellor collision, as shown in Figure 3). The attractor and repellor do not compete unless they are close together in the behavioral state space.

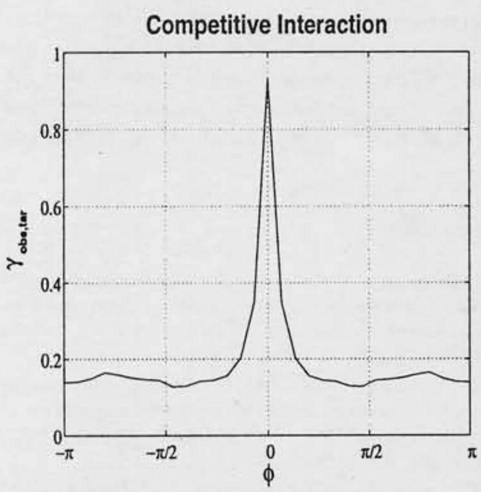

Figure 3: Capturing incompatibility between an attractor and a repellor: Competitive interaction between *obstacle* and *target*.

As a second example, consider competition between two attractors such as *target* and *other*. If both constraints are active, the agent will move in an average direction. Thus, in most situations it is necessary to enforce competition between the two attractors, although there will also be in some situations in which moving in their average direction is the appropriate behavior. When the target and the other agent are in opposite directions it is necessary to force a decision, but when they lie in the same direction, both constraints can be satisfied simultaneously. We can accomplish this type of competitive interaction using the following function.

$$\gamma_{oth,tar} = b_1(\tanh(-b_2 \cos(\psi_{tar} - \psi_{oth}) + b_3) + 1) \quad (13)$$

Competition is high except for a certain region around an angular difference of zero, as illustrated in Figure 4.

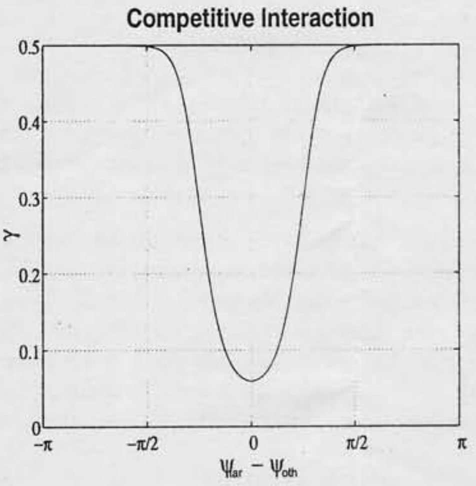

Figure 4: Capturing incompatibility between two attractors: Competitive interaction between *target* and *other*, $\gamma_{oth,tar} = \gamma_{tar,oth}$. Competition is maximum unless the target and the other agent lie in approximately the same direction.

Finally, competitive advantages must be chosen so that, when two behaviors compete the proper outcome of the competition can be determined. As an example, consider the competitive advantage of *other*. To enforce the constraint that the agents remain close together, one may choose a function that will deactivate *other* when the agents are close enough to one another.

$$\alpha_{oth} = \tanh\left(\frac{e^{r_{oth}}}{e^{d_1}}\right) \qquad (14)$$

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

Thus, the according to this approach the task is described in terms of a set of functions that tie the parameters of the task dynamics to situations in the

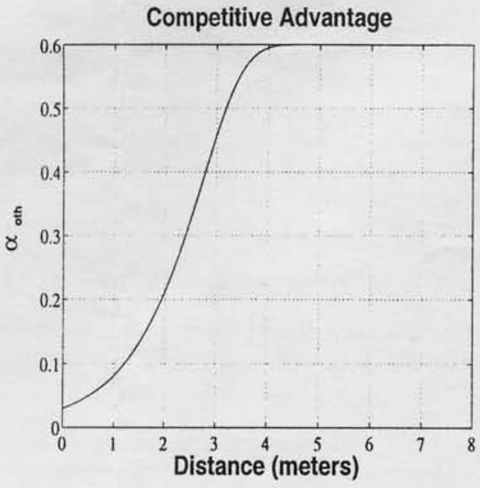

Figure 5: Competitive advantage of *other*. As distance to the other agent increases beyond a threshold value, competitive advantage, α_{oth}, increases beyond competitive interaction from target.

environment. As the agent moves about the world, the relative values of these parameters change, leading to bifurcations in the task dynamics. Bifurcations model decision-making and sequencing of behavior. Thus, for this methodology to be applicable, it is necessary that the task be describable in terms if such competitive advantage and competitive interaction functions.

4 Concluding Remarks

A task dynamics has been proposed as a method for describing complex behaviors within a dynamical systems framework. A competitive dynamical system makes it is possible to specify requirements that are more complex than simple navigation. This ability arises from the ability to determine which constraints should contribute to the behavioral dynamics, i.e. which behavior is appropriate, in any given situation. We defined a "behavior" as a qualitatively unique combination of task constraints that defines a particular set of contributions to the behavioral dynamics. Each combination of task constraints arises as an asymptotically stable fixed point of the competitive dynamics, providing a number of interesting properties. First, each behavior is stable in the sense that it is robust to the presence of noise in the system. This property arises from the stability of the fixed points that generate the behaviors. Second, each behavior is stable in the sense that it is robust to ambiguity in the environment. This property arises due to hysteresis – when more than one fixed point is stable, the past history of the system determines performance. Third, the agent is able to flexibly determine which behavior is appropriate at any given time. This property arises due to bifurcations in the competitive dynamics: As new situations arise, parameters change, old fixed points disappear, and new fixed points appear.

An advantage of the task dynamics approach is that it is based upon the qualitative theory of dynamical systems, and thus a formal analysis of the system is possible. Further, due to the choice of state space this analysis is directly relevant to understanding and designing the decision-making and sequencing behavior of the autonomous system. A weakness of this method lies in the task description. The task is described by mathematical functions for which the design principles have yet to be articulated. This does not mean that such principles cannot be formalized, simply that this job has yet to be done. If it turns out to be possible to formalize a set of design principles, then it may be feasible to create a task description language in which a designer could describe the task, and a 'compiler' could produce competitive advantage and interaction functions that describe the task within this framework. Another possibility is that such functions could be learned using an appropriate function approximation method.

Perhaps the primary advantage of this approach is that the task dynamics solution scales nicely to the design complex systems. In this paper we used the example of two cooperating robots, but more complex systems are possible. Essentially, the design decomposes the problem into interactions between pairs of behaviors. This allows not only for a compact system description, but also permits incremental design. Thus more complex systems can evolve from simpler systems. Future research will exploit this property of the task dynamics approach to the design of more complex autonomous agents and systems of cooperating agents.

References

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

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

[3] E. W. Large, H. I. Christensen, and R. Bajcsy, "Scaling the dynamic approach to path planning and control: Competition among behavioral constraints," Tech. Rep. 409, GRASP Laboratory, University of Pennsylvania, 1997.

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

[5] A. Steinhage and G. Schöner, "The dynamic approach to autonomous robot navigation," in *Proceedings IEEE International Symposium on Industrial Electronics*, 1997.

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