Dynamic Neural Fields for Robot Control

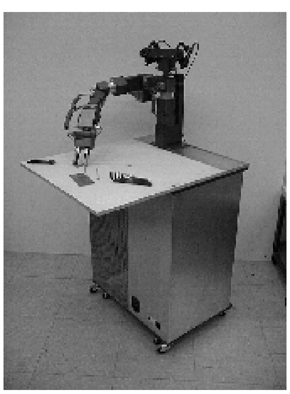
Axel Steinhage*

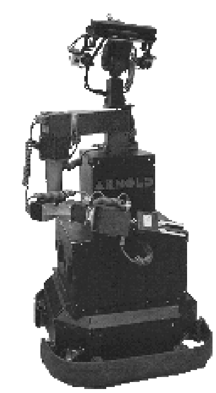
December 14, 2000

1 Introduction

The mathematical theory of nonlinear dynamical systems has proven to be an elegant and easy to use framework to generate robot behavior (for an overview see e.g. [6]). The so-called *dynamic approach* invented by Schöner in 1995 [4] provides a set of design principles which facilitate the creation of differential equations for so-called *behavioral variables* the solution of which generate the robot's behavior over time. This approach has been further developed and extended by concepts for the generation of more complex behavior and behavioral sequences [5], behavioral organization and learning [7]. We applied the dynamic approach to the demanding task of man-machine interaction [2]. For this purpose three types of dynamics turned out to be particularly useful: 1) we use so-called *attractor dynamics* to generate parameterizable elementary behaviors. The key concept is to express behavioral goals by stable fixed points of the differential equation which describes the development of the behavioral variable over time [8] 2) we use so-called *activation dynamics* for behavioral organization. Here, we assign to every elementary behavior a dynamic state variable n_i the dynamics of which has stable states at zero or one. By multiplication with the corresponding attractors in the behavior generating dynamics this state variable is used to switch on or off the elementary behaviors. Whether the activation dynamics is in the stable state one or zero depends on the presence of the so-called *sensor context* of the corresponding behavior and on the activity pattern of all elementary behaviors and their logical interrelations. 3) the third type of dynamics that we use are so-called *dynamic neural fields* which have been invented by Amari [1] as a model for neural information processing. In the following we will describe briefly the nature of these fields and how we applied them to the problem of behavior generation for our anthropomorphic service robots Arnold and Cora.[1]

2 Short description of the robots Arnold and Cora





We built the robots Cora (left) and Arnold (right) to interact with humans in fetching and delivering tasks (Arnold) and in industrial assembly tasks (Cora). Therefore, both robots have an anthropomorphic design with vision (stereo camera system), audition (microphones) and touch (bumpers for Arnold and *artificial skin* for Cora) being the only sensor modalities. They are both equipped with a simple two-finger gripper and a seven degree of freedom manipulator arm which has a similar geometry as the human arm. Cora has an additional degree of freedom in its trunk joint and will be equipped with a force-torque sensor in its wrist. Both robots are controlled by several on-board Pentium PC's running the real time operating system QNX.

An important requirement for man-machine interaction is a smooth stable behavior. We have achieved this goal by letting all of the robots' behaviors be generated entirely by dynamical systems. Especially for the navigation behavior of Arnold and for the manipulator control *neural fields* have turned out to be the tool of choice.

*Institut für Neuroinformatik, Ruhr-Universität Bochum, 44780 Bochum, Germany, Axel.Steinhage@neuroinformatik.ruhr-uni-bochum.de

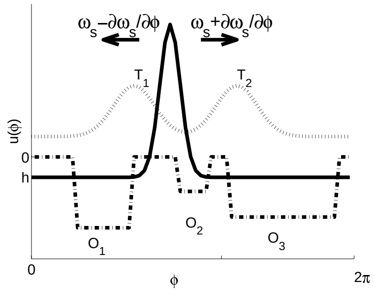
[1] Our work is funded by the german government (BMBF) in the project MORPHA (FKZ 01 IL 902 H1)

3 Dynamic Neural Fields

A simple neural field of the AMARI-type has the following form:

$$\tau\dot{u}(\phi,t) = -u(\phi,t) + s(\phi,t) + \int_{-\infty}^{\infty} \omega(\phi-\phi')f(u(\phi',t))d\phi' \tag{1}$$

Here, $u(\phi,t)$ is an *activity distribution* over the behavioral variable ϕ which is the parameterization of a behavior such as the robot's heading direction in navigation, for instance.



The *input*, $s(\phi,t)$, defines the external conditions imposed on the behavior. For navigation, this can be a sum $s(\phi,t) = s_O(\phi,t)+s_T(\phi,t)$ consisting of a negative contribution, s_O, specified by the angular positions of obstacles O_i and a positive contribution, s_T, given by directions towards targets T_i (see figure on the left). Through the integration in (1) the *interaction kernel* $\omega(\phi-\phi')$ collects activity from a neighborhood around the reference point ϕ. By means of the sigmoidal *nonlinearity* $f(u) = \frac{1}{e^{-cu}+1}$ only sufficiently high activity $u(\phi',t)$ contributes to this convolution. Given that all parameters are set appropriately, the dynamics (1) converges on the *time scale* τ to a stable *peak solution* which makes a compromise between the obstacle- and the target information (see the heavy line in the figure). The position of this peak can then be used to control the forward movement of the robot in a navigation task. Usually, the interaction kernel $\omega(\phi)$ has the form of a *Mexican Hat* $\omega_s(\phi) = ke^{-\frac{\phi^2}{2\sigma^2}} - H_0$ with an excitative part of with σ and an infinitely wide inhibitory part with asymptotical negative threshold $-H_0$. In addition to the influences of obstacles and targets, the position of the peak can be further controlled externally by means of a mathematical concept described in [3]. The basic idea is to make the interaction kernel in (1) asymmetric by building the sum of the symmetrical kernel and its spatial derivative: $\omega(\phi) = \omega_s(\phi) - n_l\frac{\partial\omega_s(\phi)}{\partial\phi} + n_r\frac{\partial\omega_s(\phi)}{\partial\phi}$

In the static case, the *activation variables* (see introduction) n_l, n_r vanish such that the kernel is symmetric. Activating the elementary behavior *move left* by driving the dynamics of behavioral organization into the fixed point $n_l = 1$ makes the peak (and hence the robot) move to the left until the kernel's asymmetry is in equilibrium with the obstacle information. Activating n_r shifts the peak to the right (see figure). This example shows how multiple behaviors (target acquisition, obstacle avoidance, external guidance) can be integrated in a single dynamics resulting in particularly smooth movement. We apply neural fields also to the problem of manipulator control [2] and scene interpretation [6].

4 Conclusion

Designing robot control systems entirely as continuous nonlinear dynamical systems turns out to be a remarkably powerful approach. It combines the property of nonlinear dynamical systems to provide a rich set of possible stable states (which corresponds to multiple behavioral states) with the smoothness and stability of continuous control. Neural fields, as specific nonlinear dynamics, adds the possibility to combine multiple behavioral constraints and continuous input distributions in a single elegant equation.

References

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

[2] T Bergener, C Bruckhoff, P Dahm, H Janßen, F Joublin, R Menzner, A Steinhage, and W von Seelen. Complex behavior by means of dynamical systems for an anthropomorphic robot. *Neural Networks 12*, pages 1087–1099, 1999.

[3] R Menzner, A Steinhage, and W Erlhagen. Generating interactive robot behavior: A mathematical approach. In *From Animals to Animats 5: Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior*, 2000.

[4] G Schöner, M Dose, and C Engels. Dynamics of behavior: Theory and applications for autonomous robot architectures. *Robotics and Autonomous Systems*, 16:213–245, 1995.

[5] A Steinhage. *Dynamical Systems for the Generation of Navigation Behavior (Ph.D. thesis)*. Number ISBN 3-8265-3508-1 in Berichte aus der Physik. SHAKER-Verlag, Aachen, Germany, 1998.

[6] A Steinhage. The dynamic approach to anthropomorphic robotics. In *Proceedings of the Fourth Portuguese Conference on Automatic Control, Controlo 2000*, 2000.

[7] A Steinhage and T Bergener. Learning by doing: A dynamic architecture for generating adaptive behavioral sequences. In *Proceedings of the Second International ICSC Symposium on Neural Computation, NC'2000*, pages 813–820, 2000.

[8] A Steinhage and G Schöner. Self-calibration based on invariant view recognition: Dynamic approach to navigation. *Robotics and Autonomous Systems*, 20:133–156, 1997.