1

Invited Article NNK98076SP

Complex Behavior by means of Dynamical Systems for an Anthropomorphic Robot

Thomas Bergener, Carsten Bruckhoff, Percy Dahm
Herbert Janßen, Frank Joublin, Rainer Menzner
Axel Steinhage[1], Werner von Seelen

Institut für Neuroinformatik, Lehrstuhl für Theoretische Biologie
Ruhr-Universität Bochum, 44780 Bochum, Germany

Running title: Dynamics for Robot Behavior

[1]Send correspondence, proofs and reprints to:
Dr. A. Steinhage, Institut für Neuroinformatik
Ruhr-Universität Bochum, 44780 Bochum, Germany
Phone: +49 234 700 7969, Fax: +49 234 709 4209,
Email: Steinhage@neuroinformatik.ruhr-uni-bochum.de

This work was supported by the german ministry for education and research (bmbf) within the NEUROS project through grant number FKZ01IN504F

Complex Behavior by means of Dynamical Systems for an Anthropomorphic Robot

Keywords:

Anthropomorphic Robot; Robot Human Interaction; Dynamic Approach; Behavior Organization;

Abstract:

We present an architecture to generate behavior for an anthropomorphic robot. The goal is to equip the robot with the capacity to interact with a human. Motivated by the research on biological systems, our basic assumption is that the behavior to perform determines the external and internal structure of the behaving system. We describe the anthropomorphic design of our robot and present a distributed control system that generates human-like navigation and manipulation behavior. As the mathematical framework for this purpose we have developed a control system which is entirely based on dynamical systems in the form of instantiated dynamics and neural fields. We also present a dynamic scheme for behavioral organization based on competitive dynamics.

1 Introduction

The behavior of living organisms is based on their own sensory information. As the acquisition of sensor information is an active process which changes the state of the system within the environment, the behaving system directly receives feedback from the external world. This feedback is the basis for the high flexibility of living systems which allows them to autonomously react to changing conditions in their surrounding. Within their environment, biological organisms always have to interact with other living systems. In particular among human beings this interaction can have the form of a partnership which allows them to solve a given task more efficiently. As every biological organism is the result of a long-term evolution, their internal behavioral control mechanism as well as their bodily shape are optimized with respect to the tasks they have to solve.

Optimizing the solution of certain tasks has always been the major goal of robotics research too. Within the field of industrial robotics this has led to machines which are highly specialized for their specific manufacturing purpose. The "interaction" between robots and humans is restricted to supervision or supply by the human. Industrial robots are not built to flexibly behave in dynamically changing environments. In most cases, their environment, i.e. the factory cells, is specifically designed to allow for a reliable operation of the preprogrammed robot.

Another line of robotics research has concentrated on artificial autonomous systems. It turned out that closing the feedback loop between the robot and the external world can be achieved for low-level behaviors based on low-level sensor information (Brooks, 1991). Along with this goes the idea to omit central high-level representations of the robot's behavioral state and its environment in favor of decentralized sensor-near representations that are sufficient just for the behavior to generate. This view has been motivated by findings of the sciences investigating nervous systems. The analysis of Braitenberg (Braitenberg, 1984), for instance, has offered a perspective on how the structure of nervous systems embodies behaviors in a rather distributed and low-level fashion.

Along the way of trying to translate the basic principles of biological behavior generation into an architecture applicable to artificial robot systems, surprisingly complex behavior has already been achieved (Steinhage, 1998). However, in these applications autonomy is still restricted to a reaction of the robot to changes in the lifeless world.

Hence, our goal is to design a system, which has the capacity to interact with human beings as partners in solving common tasks. Based on the biologically motivated conviction that the behavior to perform determines the external and internal structure of the behaving system, we use a robot which has an anthropomorphic design. As formal framework for translating basic principles of biological information processing for behavior generation we choose the so-called *dynamic approach to robotics* (Schöner et al., 1995) which is based on the mathematical theory of dynamical systems.

This paper demonstrates our approach for a number of behaviors implemented on the anthropomorphic robot ARNOLD (Fig. 1). The selection of these behaviors is directed towards an interaction between the robot and the human being.

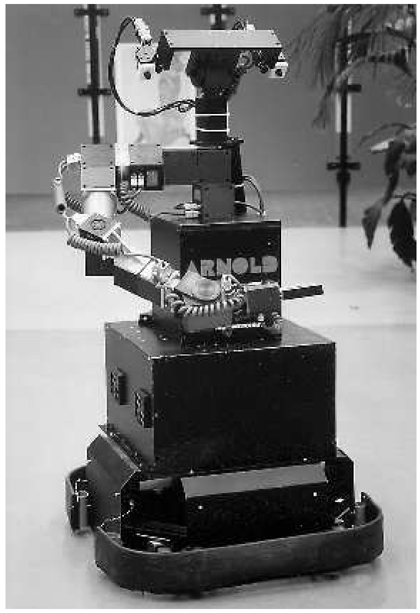

Figure 1: Robot ARNOLD: The active vision head and the anthropomorphic arm are mounted on top of a slim body. The rotation axis of the shoulder joint, the vehicle and the head's pan joint are identical, simplifying the coordination of the coupled control systems.

2 Concepts

2.1 Why anthropomorphism?

Every robot needs to be adapted to its environment and task - just like evolution adapts creatures to their environment and ecological niche. In the case of our project objective which is to develop a robot for service tasks like bringing and fetching objects in an indoor environment, this means it has to move on flat surfaces in sufficently lit rooms to use passive photo detectors as its main sensors. Doors will have a certain minimum size and the robot will have to avoid both static and dynamic obstacles. It also will have to manipulate objects that are positioned on tables or other horizontal surfaces in locations that are comfortable for a standing or sitting human to manipulate objects within.

In fact, the case of an indoor environment makes designing an appropriate robot much easier since this specific environment is artificial in that it is highly adapted to the specifics of our human anatomy, physiology and skills.

In our project NEUROS (Neural Robot Skills) we designed and built the mobile robot ARNOLD as a development and demonstration platform. ARNOLD's main components are the camera head, the arm and the mobile base (Fig. 1). The camera head is designed to perform well for ARNOLD's typical tasks which include orientation and locomotion in an indoor environment, recognizing, locating and grasping objects. While for navigation tasks a wide field of view is necessary to avoid obstacles in cluttered spaces, recognition and visually controlled manipulation of small objects require a sufficiently high sensor resolution. We adopted the basic principles of the human visual system within the limits of currently available hardware by using two pairs of cameras, one with a small field of view and one with wide angle optics. This mimics the varying resolution of the human eye which results in foveal and peripheral vision (DeValois and DeValois, 1988). The use of stereo vision - i.e. depth perception based on triangulation between a sensor pair - for manipulation tasks is another analogy to the biological model: due to the accuracy limitations of our eye distance and retina resolution, humans use stereo based depth mainly up to a distance of about one meter which also defines the space in which we can do manipulation tasks (Collewijn and Erkelens, 1990).

The general shape of the body has to enable movement in cluttered environments and especially through tight passages like door frames. To navigate in the close vicinity of walls and obstacles leads to the requirement to see these places without self-occlusion while wandering around. Since we did not use a biped for technical reasons, we chose to employ a pyramidal shape for the robots body to allow for visually controlled movement in tight spaces.

With a height of 1.35 meters ARNOLD is about as tall as a sitting adult, an ideal size to manipulate objects on tables. The arm with its seven degrees of freedom is a good approximation of the human arm within the limits of currently available hardware. The maximum grasping radius is about one meter with a maximum load of 1.5 kilogram. Since only six degrees of freedom are necessary to grasp an object at any location and in any orientation, the seventh degree of freedom results in redundancy: The angle of the elbow relative to the shoulder-end-effector axis can be chosen independently of the end-effector position and orientation. Similar to the human elbow this can be used to either satisfy additional contraints like avoiding obstacles or to generate simpler and smoother trajectories. The anthropomorphic assembly with the arm sideways and below the head ensures that the arm does not occlude the object to grasp.

ARNOLD's mobile platform can be controlled by setting the forward velocity and the rotation speed around the center axis meaning that the robot is able to rotate in place. The platform's square footprint is small enough to pass doors and to navigate in common office environments.

2.2 Behavior based robotics

Even if a robot is to act in an a priori unknown environment one can find some basic assumptions about its "world". For instance for the case of our service robot we assume sufficient illumination and flat floors of the working area. Using these assumptions it is possible to develop fast and reliable techniques to perceive those aspects of the world that are relevant to control the robot's behavior.

In the mid-eighties Braitenberg and Brooks defined a new approach to robotic systems (Braitenberg, 1984, Brooks, 1986). The basic principle in what Brooks first called *behavior-based robotics* is to directly link a continuous stream of sub-symbolic sensor input with a mechanism that controls the effectors such that the robot behaves in the intended way.

Assuming that the effect of the robot's actions is not completely predictable, sensing and acting must be done closed loop in a self-corrective way. Aiming at a behavioral goal thus means to consecutively improve the system's state concerning some behavioral regime. A robot that locomotes in an unknown environment and controls an active sensor determines its own limited view of the world meaning that controlling the actuators and evaluating the sensor data can not be discussed separately. We understand this feedback as a dynamic interaction with the environment. The crucial question of autonomous robotics must then be to find concepts that enable us to perform a given task with the minimum amount of resources - i.e. information gathering and processing - necessary.

At this point we have to describe the impact of the sensor data on the robot's actions. A representation, if needed at all, must be strongly adapted to a specific behavioral ability of the robot. The lack of knowledge about the environment thus leads to a shift from describing the world to describing the robot's abilities. Such a representation must cover all qualitative and quantitative aspects of a single behavior and, at the same time, it has to be restricted to them as far as possible. The chosen representation must provide two mappings: First, a projection of the sensor input onto the parameters of the interaction scheme and second, a projection of this state space onto the robot's effectors.

We claim two important qualities for the interaction scheme itself: *stability*, i.e. it must aim at a behavioral goal even under strong perturbations of the sensor input and *flexibility*, i.e. it must offer different strategies in qualitatively different situations.

For this kind of problem dynamical systems are the tool of choice: They can model time-continuous systems in abstract state spaces that develop in time depending on the system's state and on a continuous input. Stabilization through feedback of the system's state is a natural principle in dynamical systems.

Non-linearities provide to construct different modes of a system's behavior: Bifurcations of the underlying differential equations produce discrete changes of the position and quality of the system's fixed points depending on the input and thus achieve the required flexibility. Furthermore, standard methods for dynamical systems let us analyze and predict the system's behavior for different sensory situations.

Section 3 will introduce a systematic way to design dynamic robot behaviors.

2.3 Hierarchy of Time Scales

A network of loosely coupled dynamic systems can be quite difficult to analyze. To keep the overall system stable and usable we define a *hierarchy of time scales* for the incorporated behaviors, i.e. coupled dymamics are separated by different time scales. By adjusting the relaxation speed of the single dynamical system, the slower modules will dominate or "enslave" the faster ones, since the faster dynamics will always relax to stable states with the state of the slower systems acting as quasi constant parameters. Without changing the interconnection scheme we can thus induce a hierarchical order into the network of loosely coupled behaviors and keep the analysis of the robot's overall behavior - especially its stability - manageable. In addition, this hierarchical order can be flexibly changed by inverting the relation of the time scales within the system depending on the sensor input (Steinhage and Schöner, 1997).

2.4 Complex Behavior

The same principles that guide our design of the robot's basic behaviors hold for the organization of complex behaviors. An action selection scheme must be realized by a system that continually evaluates the situation the robot is in, its current goals and the last actions. Since we abandon to equip the robot with the capability of symbolic reasoning we neither need any symbolic representation but describe the deciding parameters as continuous variables in an abstract state space. An appropriate behavioral pattern will then be a fixed point in this state space, a stable solution of a dynamical system parameterized by the sensor situation. A changing environment will drive the action selection scheme to bifurcate and a new appropriate behavioral pattern will arise (see section 4 for the implemented framework).

2.5 A New Interpretation of Behavior-Based Robotics

Summarizing we claim a new interpretation of the approach of behavior based robotics: Dynamic control and continuous feedback must be fundamental properties of autonomous systems. In our opinion dynamical systems are the ideal tool to develop such systems in a flexible but stable way. Stability can be proofed mathematically while the system's flexibility is achieved by bifurcations of the underlying non-linear differential equations. These characteristics go beyond the previous behavior based architectures (Braitenberg, 1984, Brooks, 1986) in that they provide a well defined generative framework for designing autonomous systems. Both, the behavioral modules and their coupling scheme are designed using the same dynamic mechanisms. This allows for flexible hierarchies within the system as the time scales of the behaviors may be altered depending on the sensor input (Steinhage and Schöner, 1997).

The projected behaviors determine the architecture of the control system and the hardware design with a strong interrelation between these two. An inadequate design or adaptation of either of these will limit the robot's performance.

3 Implementation

The following section describes concrete implementations of these general principles. For the example of approaching a door, we will explain how the corresponding dynamical system, described by a differential equation, can be derived. Further we will show how this architecture can be generalized to achieve a smooth integration of sensor information using the concept of neural fields. The section is closed by presenting a collection of elementary behaviors implemented on ARNOLD which serve as a basis for the behavioral organization architecture described in the next section.

3.1 Instantiated Dynamics

The first step towards a mathematical formulation of a behavior generating dynamics is the specification of the corresponding *behavioral variables*. These variables can be scalar or vector valued. In some simple cases a behavioral variable directly parameterizes the elementary behavior to be generated. As an example for such an *instantiated dynamics* we describe the target acquisition behavior of our robot when approaching a door (see Steinhage and Schöner, 1997) for other examples). This behavior can be parameterized by the robot's current *heading direction* $\varphi(t)$ measured relative to an arbitrary but fixed global reference direction $\varphi_0 = 0$. Varying the heading direction while keeping the forward velocity v constant generates a trajectory. The variable $\varphi(t)$ therefore describes the *behavioral state* of the system with respect to the behavior of locomotion. The basic idea of the dynamic approach is to map the behavioral state of the system onto the mathematical state of a dynamics described by an appropriate differential equation. For the example of locomotion, a trajectory can be generated by integrating the dynamical system

$$\tau_\varphi \dot{\varphi}(t) = F(\varphi, t, \vec{p}) \tag{1}$$

in time. Here, $\dot{\varphi} = \frac{d\varphi}{dt}$ is the temporal derivative of the heading direction. The parameter τ_φ is the *time scale* of the dynamics which controls how fast the behavioral variable can change. The function F describes how the heading direction changes in time depending on the current value of φ, the time t and on a vector of parameters $\vec{p}$. Fig. 2 shows a sketch of the door passing behavior implemented on ARNOLD: The goal is to cross the threshold between two door posts perpendicularly. The positions of the door posts relative to the robot are determined by a model based stereo algorithm using camera images acquired during locomotion. The detailed description of this algorithm can be found in (Steinhage and Bergener, 1998). Using the position information of the door posts, the distance and orientation of the door relative to the robot can be calculated. A smooth trajectory can be generated by approaching a location D in front of the door from far distances r_D and turning into a perpendicular orientation φ_T towards the threshold of the door at near distances. These two parts of the overall goal to pass the door can be expressed by two terms on the right hand side of the behavior generating dynamics:

$$\tau_\varphi \dot{\varphi} = (1 - e^{-r_D^2/\delta^2}) \sin(\varphi_D - \varphi) \; + \; e^{-r_D^2/\delta^2} \sin(2\varphi_D - \varphi_T - \varphi) \tag{2}$$

The second term can be neglected for large distances $r_D \gg \delta = \text{const}$ where $e^{-r_D^2/\delta^2} \simeq 0$, while the first term can be neglected for small distances $r_D \ll \delta$

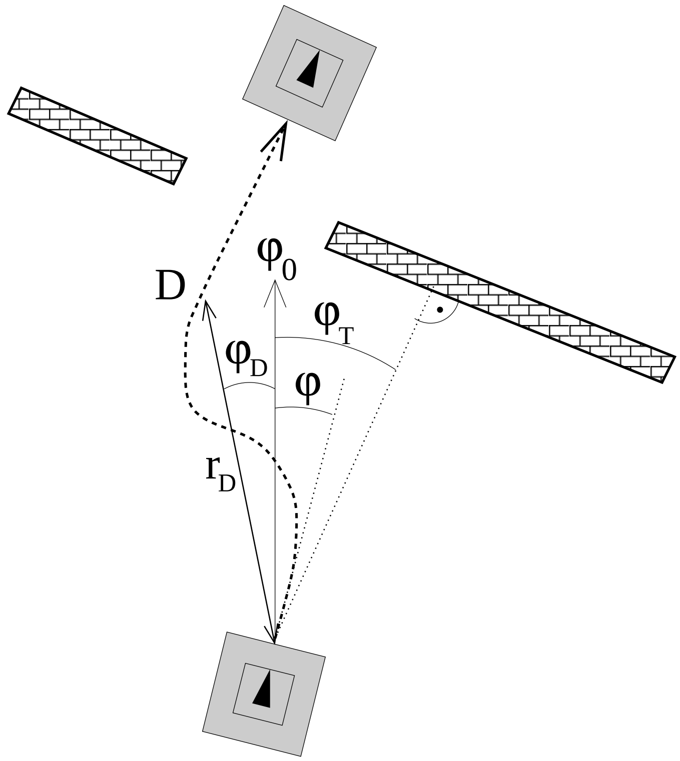

Figure 2: The door passing behavior. From the initial position the robot first approaches a location D in front of the door by stabilizing a heading direction $\varphi = \varphi_D$. Near position D, the dynamics stabilizes the direction $\varphi = \varphi_T$ which is perpendicular to the threshold of the door.

for which the exponential function is $\simeq 1$.

Within the dynamic approach behavioral goals are mapped onto stable states of the underlying dynamical system. A stable state φ_s is given when the dynamic state variable at $\varphi = \varphi_s$ does not change in time, i.e. $\dot{\varphi}|_{\varphi=\varphi_s} = 0$ and, in addition, $\frac{d\dot{\varphi}}{d\varphi}|_{\varphi=\varphi_s} < 0$. The latter means that small deviations $\Delta\varphi = \varphi - \varphi_s$ from this state lead to a relaxation of the system back into the state φ_s. For large distances, for which the first term in equation (2) dominates, the dynamics has a stable state at $\varphi = \varphi_D$. This specifies the goal to approach the location D by turning the heading towards the direction of D. For small distances the second term dominates and a stable state exists at $\varphi = 2\varphi_D - \varphi_T$ specifying the goal to turn smoothly such that finally $\varphi = \varphi_D = \varphi_T$ and the robot can pass through the door.

This example shows that the mathematical equation for generating complex behavior can be derived in a straightforward way by expressing behavioral goals through stable states of the underlying dynamics. Further it is possible to "blend" multiple parts of an overall behavior into a single dynamics. Sensor information acts on the dynamical system by means of the parameters $\vec{p}$ in (1). E.g. for the door passing behavior the positions of the door posts determined by the stereo vision system act on the control dynamics (2) through the position of the stable states φ_D, φ_T and through varying their strength of attraction specified by $\exp(-r_D^2/\delta^2)$. As the robot moves, the stable states move on the time scale of the movement in real time. Therefore, the generated behavior directly depends on the sensor information providing the possibility to flexibly react to changes in the robot's environment. Conversely, the acquired sensor information depends on the behavior of the robot itself as the door posts' positions move relative to the robot's location. The time derivative of the behavioral variable φ is directly used by the motor controller of the robot platform to drive the wheels. Hence, there is a dynamic interaction between the robot and the environment expressed by the behavior generating dynamical system.

3.2 Neural Fields

In the application we discussed, the value of the behavioral variable directly represents an aspect of the physical behavior of the system. Although the number and positions of the stable states that represent the behavioral goals may change discontinuously (e.g. through a bifurcation of the dynamics), the behavioral variable itself changes continuously. There are cases, however, for which a more general form of behavioral variable has to be chosen. Sometimes it may be necessary to represent the presence or absence of certain instances in the environment, like multiple targets for example. In those cases, the behavioral variable φ does not always have a unique value but can have multiple values or even no value at all depending on the environment. Generally spoken, there exist cases in which it is necessary to express the behavioral state by a continuous function of the behavioral variable. E.g. a peak of that function extended over a range of values of the behavioral variable would represent, to which extent these values influence the actual behavior of the system. In the dynamic approach such "non instantiated" behavioral variables can be represented by a so-called *neural field*. These fields have initially been invented by Amari (Amari, 1977) to model information processing of cortical layers. They are equivalent to continuous neural networks embedded in a low dimensional space in which the "units", i.e. locations φ within the field are laterally coupled through an interaction kernel $\omega(\varphi, \varphi')$ of defined metric in that space and receive external input $S(\varphi, t)$. In the Amari-equation for the field activation $u(\varphi, t)$

$$\tau \dot{u}(\varphi, t) = -u(\varphi, t) + S(\varphi, t) + h + \int_{-\pi}^{\pi} \omega(\varphi, \varphi')\sigma(u(\varphi', t))d\varphi' \qquad (3)$$

the constant h defines the global mean activation level within the field and the transfer function $\sigma(u)$ controls the local threshold of activation. Depending on the parameter h and the form of S, σ and ω, the activation dynamics (3) can have different types of solutions (Amari, 1977). From the point of view of behavior generation the *homogeneous solution* $u(\varphi, t) = h$ and the *localized solutions* are of particular interest. The first represents the state in which no value of φ is specified, i.e. where no behavioral goal is given. The second represents the state in which one or more values φ_i are specified through the maxima of localized peaks of activation at the positions φ_i in the field representing one or more behavioral goals.

We applied the concept of neural fields to the problem of local path planning. Whenever ARNOLD is moving obstacles have to be avoided. Simultaneously one or more targets may be given to the robot. We specify these multiple behavioral goals by implementing a neural field over the behavioral variable $\varphi \in [-\pi, \pi[$ which represents all possible movement directions. We use the value φ with the maximal field activation $u_{\max}(\varphi, t_0)$ at a given time t_0 as heading direction for the platform control dynamics. Consequently, we select σ and ω such that unimodal solutions of (3) are stabilized. This is done by assigning a sigmoid transfer function $\sigma(u) = 1/(e^{-cu} + 1)$ with $c = \text{const} \gg 1$ and an interaction kernel with a short-range excitation term and a constant long-range inhibition term H_0:

$$\omega(\varphi, \varphi') = k e^{-(\varphi - \varphi')^2/\delta_\omega^2} - H_0 \qquad (4)$$

While the interaction kernel ω guarantees that a unique peak is stabilized, the actual behavior of the robot is controlled by the time dependent input $S(\varphi, t)$ of the field. This input reflects the multiple behavioral goals and therefore consists of a contribution $T(\varphi) > |h|$ for target acquisition and a contribution $O(\varphi) < 0$ for obstacle avoidance. The formulation of the dynamics as a neural field allows to specify a continuous set of potential targets through a superposition of Gaussians centered around every target direction (see Fig. 3, dotted line). In the absence of obstacles and given this function as input $S(\varphi, t) = T(\varphi, t)$, the lateral interaction within the field "selects" one of the potential target directions coded in $T(\varphi, t)$ and suppresses the others. This implements *decision making* as the robot autonomously selects one of many potential targets to approach.

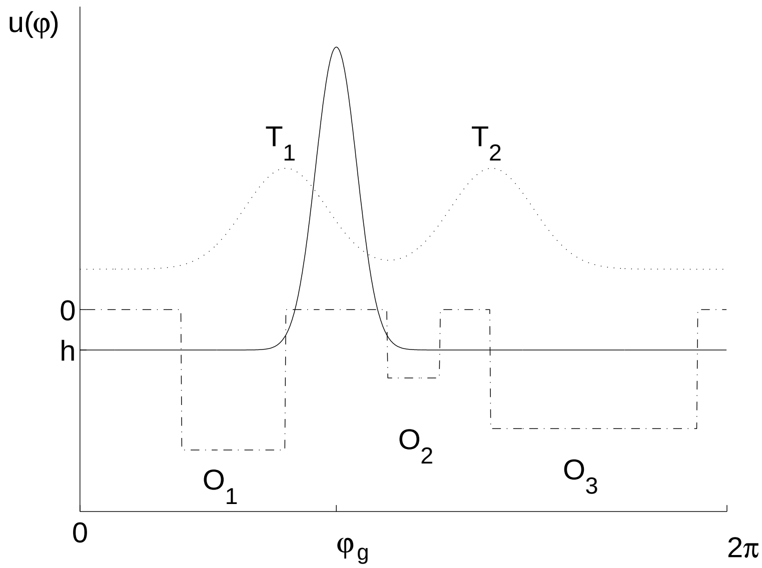

Figure 3: The neural field for path planning (schematically). The obstacle contribution $O(\varphi) < 0$ (dashed line) is strong for directions φ in which obstacles have been detected. The target contribution $T(\varphi) > |h|$ (dotted line) is strong for directions towards given targets. Given these two contributions as input, the lateral interaction within the neural field stabilizes a peak of the activation $u(\varphi, t)$ (solid line) the maximum of which specifies the next gaze direction φ_g. The robot will head towards this direction, pass between the two obstacles O_2 and O_3 and finally approach target T_1. The decision which target to approach, i.e. the position of the peak, is dependent on both, the field input $O(\varphi), T(\varphi)$ and the current state of the field $u(\varphi, t)$.

The information about the function $d(\varphi, t) \in]0, \infty[$ which represents the distance of obstacles from the robot is provided by a stereo algorithm based on the so-called *inverse perspective mapping* a detailed description of which can be found in (Bohrer et al., 1991). The obstacle contribution of the field input is then:

$$O(\varphi, t) = -Ae^{-d^2(\varphi,t)/\delta_O^2} \tag{5}$$

with constants $A > 0$ and δ_O. The dashed line in Fig. 3 shows an example of an obstacle contribution $O(\varphi, t_0)$ for a cluttered environment.

Corresponding to the anthropomorphic design described in section 2, Arnold has a limited field of view $\varphi_g \pm \beta$ around the current gaze direction φ_g, i.e. the orientation of the stereo camera head. Analyzing the locomotion behavior of humans, it can be observed that usually the gaze is shifted towards the intended walking direction as the focus of attention before the movement is performed. In dynamic terms this means that the head rotation dynamics acts on a faster time scale than the locomotion dynamics. This behavior is implemented by controlling the orientation φ_g of the stereo camera head using the same neural field (3) as for the locomotion. This means that ARNOLD's gaze is shifted towards the maximum of the field representing the area with the less obstacles and the most attractive targets before the platform moves in this direction. This results in a very smooth human-like behavior even in cluttered environments.

Although the distance function $d(\varphi, t)$ is only given within the limited field of view, previously seen obstacles are usually still present after the gaze dynamics moved the head away from their direction. Again analyzing human locomotion behavior it can be observed that humans seem to have an idea about the geometry of their surrounding outside their current field of view: E.g. once an area has been identified as a deadlock, this direction is avoided for a certain time and different paths to a given target are tried out. This behavior can be copied by implementing a dynamic short-term memory for the field input S:

$$\begin{aligned} \dot{S}(\varphi, t) &= \frac{1 - R(\varphi)}{\tau_1} \cdot [T(\varphi, t) - S(\varphi, t)] \\ &+ \frac{R(\varphi)}{\tau_2} \cdot [T(\varphi, t) + O(\varphi, t) - S(\varphi, t)] \qquad (6) \\ R(\varphi) &= \frac{1 + \tanh[c \cdot (\beta - |\varphi - \varphi_g|)]}{2}, \quad c = \text{const} \gg 1 \qquad (7) \end{aligned}$$

The right side of the dynamics (6) again has two contributions reflecting two behavioral goals: the first term specifies a stable state at $S = T$, the second term specifies a stable state at $S = T + O$. By means of the sigmoid range function $R(\varphi)$, the first term dominates the dynamics for angles φ outside the visual field $\varphi_g \pm \beta$ and the input S relaxes to the target distribution $T(\varphi, t)$ on the slow time scale $\tau_1 \gg 1$. Inside the visual field, the dynamics relaxes to the sum of the obstacle- and the target-contribution $T(\varphi, t) + O(\varphi, t)$ on the fast time scale $\tau_2 \ll \tau_1$ (Fig.3, solid line). This mechanism implements a short-term memory as previously gathered obstacle information from outside the current visual field has an influence on the neural field input for a certain memory time τ_1. Equipped with this memory, ARNOLD avoids deadlocks and displays smooth flexible path planning.

The height $u_{\max}$ of the stabilized peak within the neural field is a measure for the quality of the decision to move in that direction: If no obstacles are present and a strong target is given, the peak is strong. Hence we use the height of the peak to control the rotation and translation velocity of ARNOLD to ensure a decelaration in cluttered, unreliable or quickly changing obstacle configurations. As the limited space does not allow a detailed description of the neural field's behavior here, we would like to direct the readers attention to more elaborated analyses contained in (Dahm et al., 1998) and (Bruckhoff and Dahm, 1998).

3.3 Further Extension of the Concepts

The concepts described so far have been extended to implement a sophisticated manipulator control for ARNOLD's seven degree of freedom (7dof) robot arm. This architecture is the basis for the two behaviors *generic grasping* and *interactive control*.

Manipulator Control

For ARNOLD's manipulator arm, we have developed a closed form solution for the inverse kinematics (Dahm and Joublin, 1997) which directly maps the position and orientation of the end-effector (6 dof) and the angle of the elbow (*arm angle*) on a circle around the shoulder-wrist-axis (1 dof) onto the joint angles (7 dof). Corresponding to the anthropomorphic design, described in section 2, the additional degree of freedom represented by the arm angle is used for obstacle avoidance during the grasping process. In analogy to the one dimensional neural field for the heading direction which controls the 2-d platform position for target acquisition and obstacle avoidance, the end-effector's 3-d position is controlled by a two dimensional neural field for the two angles of the grasping direction γ, θ. This field is implemented over a two dimensional sphere surface centered around the tip of ARNOLD's end-effector. The maximum of the localized activation peak within this field is read out to control the movement of the end-effector. The distance and direction of targets are provided by a stereo algorithm and are fed into the neural field in a similar way as already described for the platform movement. However, we transfer the distances into so-called *time to contact* measures (e.g. Schöner, 1994) which encode the estimated time until a collision with a target or obstacle would happen if the end-effector was moved further into a particular direction with the current velocity. As the capacities of the visual processing do not yet allow the recognition of three dimensional obstacle positions and orientations relative to the manipulator arm, the obstacle contributions fed into the neural field stem from computer simulations with predefined obstacle configurations. The generated trajectory is locally optimized with respect to a corridor of maximal safety. The same method we described here for the end-effector's position is applied to its orientation and the arm angle. Only the topologies of the corresponding neural fields are different.

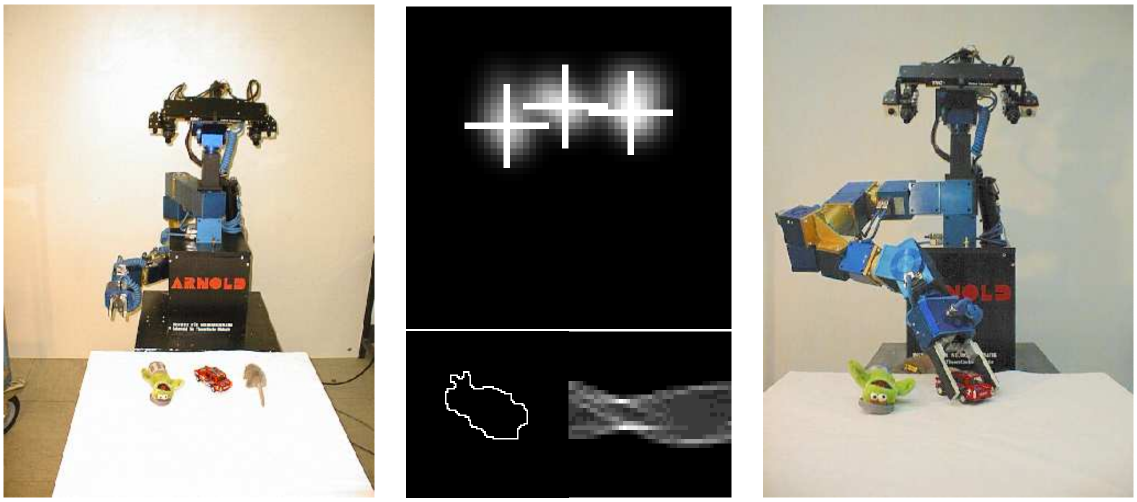

Figure 4: ARNOLD with a table scene in front (left). Coarse bird's eye view of the three objects, eroded bird's eye view of the toy car to grasp and its Hough transformation (middle). ARNOLD grasps the toy car after having deposed the first object on the platform base already (right).

Generic Grasping

The manipulator control is used for grasping objects. The object's position and orientation define the target contributions for the two dimensional neural fields which control the end-effector's position and orientation. Typical grasping approaches need precise models of the objects a robot should grasp. By contrast, we have endowed our robot with the ability to grasp simple objects in a general manner. The necessary parameters, i.e. the object's position and orientation, are extracted by a sequence of image processing steps. First, a coarse disparity image is calculated from a stereo image pair (see Fig. 4, middle). The disparity image is transformed into a bird's eye view of the entire scene in front of the robot. This allows to segment the scene into single objects. Second, the head control dynamics drives the attention to each object to acquire more precise disparity information. Based on this finer disparity image, the object's cartesian position relative to the robot can be estimated. To grasp an object safely, an appropriate target orientation for the gripper must be found. To obtain the orientation of the object, an eroded shape of the object is extracted from the image. Afterwards the remaining image edges, which correspond to the outer border of the object, are processed in a Hough transformation (Fig. 4, middle) providing all principle orientations of the object. The gripper is then oriented in a suitable grasping orientation (here perpendicular to the narrowest extension of the object) and the object is grasped (Fig. 4, right). ARNOLD can also grasp objects from the user's hands. The manipulator control dynamics acts on an intermediate time scale faster than the platform control dynamics but slower than the head control dynamics. When ARNOLD tracks an object held in front of the cameras, the head follows the movements of the user's hand immediately before the end-effector tries to approach the object. The target acquisition dynamics of the platform follows on a much slower time scale reducing the overall distance between the user and the robot. This again results in a very smooth and human-like behavior.

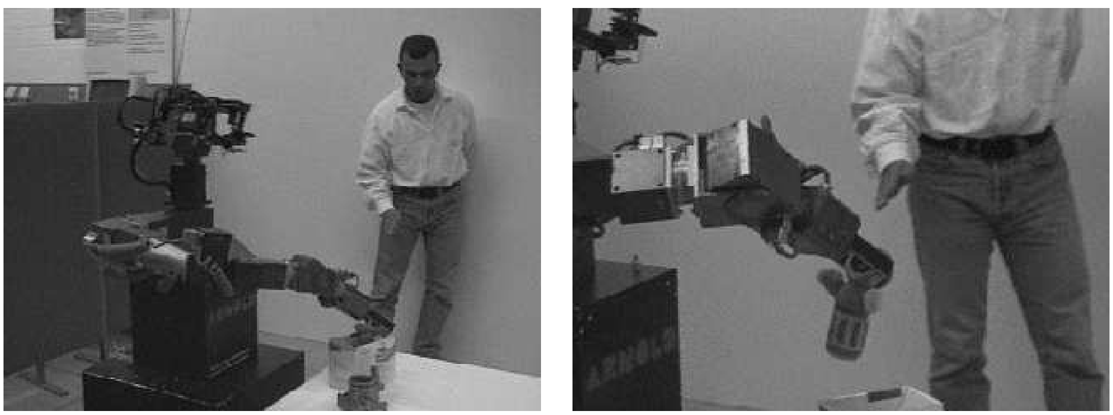

Figure 5: Interactive manipulator control: ARNOLD imitates the user's arm movement and grasps an object.

Interactive Manipulator Control

A different way of providing target positions and orientations for the manipulator is to let ARNOLD mimic a human's grasping behavior. We implemented an interactive manipulator control: The robot observes the user's hand while the user observers ARNOLD's end-effector (see Fig. 5). From the stereo image pairs the position and orientation of the hand is extracted by solving the correspondence problem of matching skin colored image patches in the left and right image. The cartesian positions of the human's wrist and hand tip are used to control the position and the orientation of the arm's end-effector in a very comfortable and intuitive manner (see Fig. 5). To close and open the gripper we use the distance between wrist and hand tip such that opening and closing of the user's hand results in opening and closing the gripper. An interesting characteristic of this approach is that the signal space can be scaled arbitrarily, e.g. large movements of the user's hand may lead to fine movements of the end-effector.

Further Abilities

Beside the presented abilities of the robot ARNOLD a global navigation system was developed and implemented. A graph-based topological map is constructed from scratch at run time and is used to re-localize and to solve complex navigation tasks.

To demonstrate the strength of dynamically controlled anthropomorphic robots in the field of man-machine interaction an application that couples head, arm and platform dynamics was developed. Fast stereo and color vision algorithms provide to control all degrees of freedom of the robot depending on a human user (Bergener and Dahm, 1997).

4 Behavior Organization

4.1 Solving Complex Tasks

To solve complex tasks it is necessary to bind the robot's basic capabilities within an action selection mechanism. In contrast to existing approaches that base on e.g. discrete event systems (Ramadge and Wonham, 1989, Košecká and Bogoni, 1994), hybrid systems (Lemmon et al., 1993) or command fusion systems (Rosenblatt, 1997) the action selection scheme presented here is inherently time dependent. It follows the principle of a stable and flexible interaction with the environment described in section 2. According to the basic behaviors this scheme is realized as a dynamical system that develops a stable pattern of active behaviors in time depending on the sensory situation and the internal state. Hence a change of the currently active behavior is not an instantaneous switching (like in the approaches mentioned above) but a variation of continuous parameters on a specified time scale (Steinhage and Bergener, 1998).

To allow the system to flexibly develop behavioral sequences under changing conditions of the environment parameterize the scheme by defining a so-called sensor context for each behavior and logical rules concerning pairs of behaviors. These constraints replace the programming of complete behavioral sequences and allow for concurrently running behaviors. Beyond that we define fallback behaviors that become active whenever the robot is strongly disturbed. Such a fallback behavior will normally just acquire information that is needed to start up a goal directed behavior. As soon as a goal directed action arises this will suppress the fallback behavior again.

All internal states as well as the abstract sensor context are expressed as continuous variables. Logical rules are mapped to parameters of the dynamical system that realizes the behavioral organization.

After introducing dynamical systems in instantiated variables in section 3.1 and neural fields in section 3.2 we use a third form of behavioral dynamics here: To model the activation state of each behavior we establish a competitive scheme of abstract state variables. In contrast to the neural field these variables do not have any topological relation. They differ from the variables that were used in section 3.1 in that their state is not related to a physical state of the robot or its effectors. This new kind of dynamics organizes a set of interacting states which have an only abstract meaning, in this case a behavior's activity (Steinhage, 1998).

4.2 The Arbitration Scheme

As behavioral variable we select a state vector $\vec{n}$, the elements n_i of which describe the activity of each behavior i. If, for instance, the behavior *approach a target* is denoted by the number $i = 1$, a value of $n_1 \simeq 0$ means that this behavior is currently not active while $|n_1| \simeq 1$ means that this behavior is active. The subset of currently active behaviors is therefore represented by all elements n_i of the state vector $\vec{n}$ for which $|n_i| \simeq 1$.

While for the generation of a single behavior the current sensor input may be sufficient, this is certainly not true for a complex system consisting of many elementary behaviors: logical and temporal requirements must also be taken into account when activating or deactivating behaviors. If we want the robot to approach a specific one of a set of many visible doors for instance, it is not sufficient that the sensors detect the shape of a door but also that the right door is selected by a recognition behavior which must have been active before. Furthermore, some behaviors can be active simultaneously while others may not. It does not make sense to activate a searching behavior simultaneously with the behavior target acquisition for instance, whereas the two behaviors target acquisition and obstacle avoidance *must* be active together if we do not want to risk a collision. In our approach we take these logical and temporal requirements into account by defining two matrices $\gamma_{i,j} \in \{0,1\}$ and $A_{i,j} \in \{0,1\}$ which serve as constant parameters for the dynamical system that controls the state vector $\vec{n}$. The matrix $\gamma_{i,j}$ defines the mutual exclusions between the behaviors: if the element j should not be active simultaneously with the element i, we set $\gamma_{i,j} = 1$. If i can be active simultaneously with j, we set $\gamma_{i,j} = 0$.

The matrix $A_{i,j}$ defines the logical presuppositions. We set $A_{i,j} = 1$ if the previous activity of j is necessary to activate i and $A_{i,j} = 0$ otherwise.

For controlling the behavioral state vector $\vec{n}$, we select a competitive nonlinear dynamical system:

$$\tau \dot{n}_i = \alpha_i n_i - |\alpha_i| n_i^3 - \sum_j \gamma_{i,j} n_j^2 n_i + \xi_t \tag{8}$$

$$\text{with} \quad \alpha_i = 2\rho_i - 1 \tag{9}$$

$$\text{and} \quad \dot{\rho}_i = \frac{(1-\rho_i) I_i}{\tau_{\rho,2}} + \frac{\rho_i (I_i - 1)}{\tau_{\rho,1}} \tag{10}$$

Here, τ is the time scale of the competitive dynamics and α_i is the so called *competitive advantage*. While $\alpha_i < 0$, the dynamics (8) is in the stable fixed point $n_i = 0$ and the behavior i is deactivated. Otherwise, if $\alpha_i > 0$, the activation of n_i depends on the activity of all the behaviors j that compete with i and for which therefore $\gamma_{i,j} = 1$: if all the n_j are deactivated, the activation of behavior i relaxes to one of the stable fixed points $n_i = \pm 1$ which specify the activity of the behavior number i. However, if at least one of the competing n_j is active, n_i can not be activated and remains in the stable fixed point $n_i = 0$. This mechanism prevents the simultaneous activation of two competing behaviors characterized by $\gamma_{i,j} = 1$. The small stochastic noise term ξ_t in (8) kicks the system out of unstable states $n_i = 0$. The competitive advantage α_i in (9) depends indirectly on the input function $I_i \in [0,1]$: For $\rho_i \simeq 1$, an active input $I_i \simeq 1$ makes $\alpha_i \simeq 1$ and thus activates behavior number i. Conversely, $I_i \simeq 0$ deactivates behavior i through an $\alpha_i \simeq -1$. The so called *refractory term* ρ_i given by (10) acts as a filter for the input I_i: if an input $I_i \simeq 0$ switches off behavior i, the refractory term follows to $\rho_i \to 0$ on a fast time scale $\tau_{\rho,1} \simeq 1$. When an input becomes active again ($I_i \to 1$), the dynamics (10) follows to $\rho_i \to 1$ on the slow time scale $\tau_{\rho,2} \gg 1$ inhibiting the activation of behavior i for that time $\tau_{\rho,2}$. Through this mechanism, a behavior that has just been deactivated cannot instantaneously be activated again, preventing the system from undesired behavioral oscillations. The input function I_i is a combination of the so called *sensor* *context* $C_i \in [0,1]$ and a part which contains the logical presuppositions:

$$I_i = C_i \prod_j \left(1 - A_{i,j}(1-m_j)\right), \tag{11}$$

$$\dot{m}_i = \frac{(1-m_i)n_i^2}{\tau_n} + \frac{m_i(n_i^2-1)}{\tau_i} \tag{12}$$

In the sensor context, we subsume all sensor conditions required to activate behavior i: if $C_i \simeq 0$, the required conditions are not fulfilled and the input in (11) is set to $I_i \simeq 0$. $C_i \simeq 1$ means, that all sensor conditions are fulfilled. Under these circumstances the input depends on the product term in (11) which implements a logical AND condition: if at least one of the factors in the product term is $\simeq 0$, the input is switched off too. This happens if one of the presuppositions with $A_{i,j} = 1$ is not fulfilled, indicated by $m_j = 0$.

In the dynamics (12) m_i acts as a *short term memory* for the activation of a behavior: a deactivation $n_i \to 0$ is followed by $m_i \to 0$ on the slow time scale $\tau_i \gg 1$, while an activation $n_i^2 \simeq 1$ is instantaneously followed by $m_i \to 1$ on the fast time scale $\tau_n \simeq 1$. Through this mechanism, a presupposition j for the behavior i can activate behavior i for a time τ_i even though the required behavior j may already be switched off. By this, the switching process between mutually presupposing behaviors is made more stable and even short activations of presuppositions can be used for slow switchings. The control flow for the complete arbitration scheme is shown in Fig. 6.

Since it is error-prone to directly edit the matrices A and γ we developed a simple language to parameterize the arbitration scheme that maps linguistic expressions to the matrix elements.

4.3 Passing a Door

To demonstrate the presented arbitration scheme we realized the task of passing a door. The task was divided into seven simple behaviors: visually search the door (behavior `search_door`), track it with the active vision head (`track_door`), approach it from a certain distance (`approach_door`), move to a position right in front of the door in the appropriate orientation to pass it (`near_approach`), pass the door frame (`pass_door`), stop in case of a collision (`collision`) and move back from the door after a collision (`free_coll`).

We will describe the behavior `approach_door` exemplarily: The navigation dynamics for this behavior was already described in section 3.1. Its sensor context

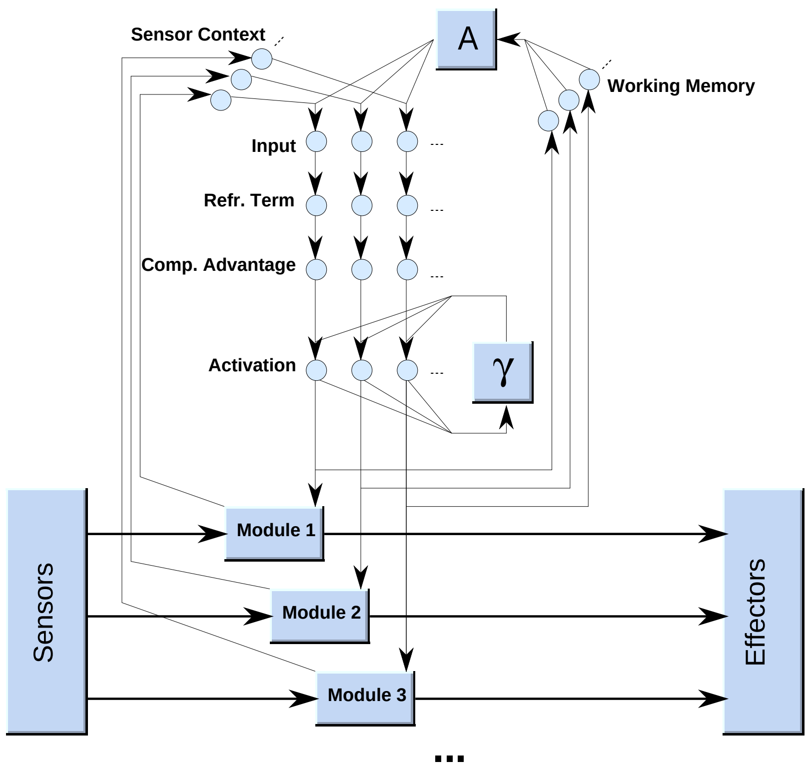

Figure 6: The arbitration scheme: The basic behavioral modules, each defining a data transformation between sensors and effectors, are controlled by a dynamical system that switches the modules' activities. A module's activation or deactivation depends on its sensor context and on logical rules concerning mutual exclusions (coded in the matrix γ) and presuppositions (coded in matrix A).

$C_{approach\_door}$ is set to one whenever a door is detected and its distance is bigger than a threshold and zero else. The rules for `approach_door` were defined as follows:

```
behavior approach_door(
  requires track_door;
  competes near_approach;
  competes search_door;
)
```

A `requires`-line produces an entry of one in the corresponding element in the matrix A and a `competes`-line an entry of one in γ respectively. The behavior `approach_door` thus can only be active when the door is stably tracked. When it runs it suppresses the behaviors `near_approach` and `search_door`. As soon as the robot comes close to the door the behavior `approach_door` is deactivated (by a changing sensor context) and the behavior `near_approach` starts.

A sample run of this framework demonstrates its suitability (Fig. 7 and 8). The lines in figure 7 show the activity ($|n_i| \simeq 1$) of each behavior. At time step 300 the robot touches a person that suddenly stepped into the door frame and the `collision`-behavior is activated, triggered by the bumper contact. The robot moves back and passes the door in the second try. Figure 8 shows the robot's path through the door.

This experiment was performed under various conditions and perturbations: We provoked collisions on different points of the sequence and occluded the scene while the robot approached the door. The presented system emerged to be very robust. In case of strong perturbations it deactivates the currently active navigation behavior and the `search_door` behavior comes up trying to newly detect the door. Even adding new behaviors appeared fairly easy since we could leave the existing framework unchanged and only had to define the rules concerning the new behavior (adding one row and one column in the matrices A and γ respectively).

5 Conclusion and Outlook

5.1 Conclusion

This paper proposes a methodology to understand a robot as an active part in a dynamic environment. The resulting interactions are described as dynamical systems

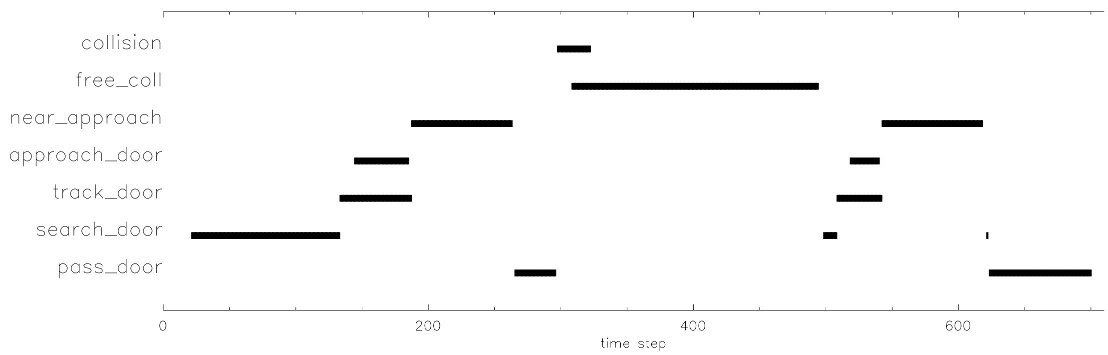

Figure 7: This plot shows the activity of the different behaviors during the run (a time step equals 200 ms): After the visual search the tracking the door and approaching it run in parallel. When the distance to the target point is less than 1.3 meters these behaviors are switched off and the robot starts to position in front of the door. At time step 300 the robot has a collision in the door frame. It moves back and passes the door in the second try.

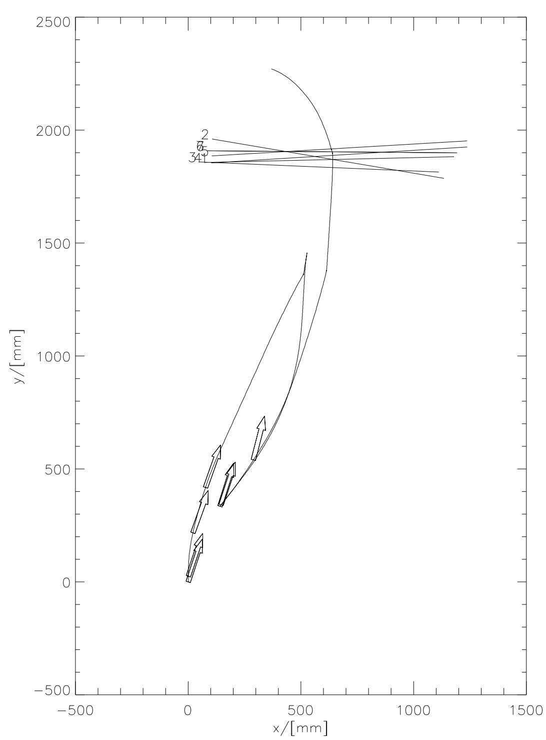

Figure 8: The robot's path in the door experiment according to the odometric measurement: The numbered lines show the estimated positions of the door. The arrows mark the positions and view directions in which the robot detected the door.

on different levels: On the level of a single behavior the concepts of instantiated dynamics and neural fields are used to design flexible control strategies generating different stable behaviors in changing environments. On the level of behavioral organization a competitive dynamics selects one or more behaviors from a repertoire for the current situation.

The use of non-linear differential equations introduces a new quality in behavior based robotics. It is demonstrated how bifurcations of these systems can provide different modes of control in a single compact equation. The separation of time scales is demonstrated as a tool to introduce order into loosely coupled control systems and to keep those systems manageable and stable.

Compared to more symbolical or planning-based approaches, dynamical systems are able to generate behaviors with low latency using little computing power while showing smooth movement. Since behavioral control in our case is mainly sensor based as opposed to representation based, they perform especially well in environments with low temporal stability and consistency (compare Arkin's discussion of reactive systems in (Arkin, 1998)). Tasks that require complex representations and a lot of world knowledge have not yet been addressed for dynamical systems. As demonstrated in this paper, dynamical systems can be used on the level of behavioral architecture for decision making, implementing emergent behavior sequences and parallel behavior execution and thus significantly go beyond the possibilities of reactive systems.

Anthropomorphic robot design is proposed to optimally adapt the robot to human environments and to facilitate the coupling of a visual sensor and dynamically controlled effectors.

5.2 Future Work

Future work will concentrate on extending our work on autonomous robots to *interactive autonomous robots*. Based on the results of this project the user must be represented as a very special part of the robot's environment. Hence the development of an intelligent assistant will not decrease the robot's autonomy but use it in the sense of a more efficient interaction of man and machine.

In addition to the visual sensor haptic and acoustic input will be used. A speech interface that provides to bias the robot's decision making concerning modes of a single behavior (e.g. "left" or "right") or, on the level of behavioral organization, to propose a certain behavior is already implemented on the robot ARNOLD.

References

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

Arkin, R. C. (1998). *Behavior-Based Robotics*. MIT Press.

Bergener, T. and Dahm, P. (1997). A framework for dynamic man-machine interaction implemented on an autonomous mobile robot. In *Proceedings of the IEEE International Symposium on Industrial Electronics, ISIE'97*.

Bohrer, S., Lütgendorf, A., and Mempel, M. (1991). Using Inverse Perspective Mapping as a Basis for two Concurrent Obstacle Avoidance Schemes. In *ICANN–91*, Helsinki. Elsevier, North Holland.

Braitenberg, V. (1984). *Vehicles. Experiments in Synthetic Psychology*. MIT Press, Cambridge, Mass.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation, RA-2.(1)*.

Brooks, R. A. (1991). New approches to robotics. *Science*, 253:1227–1232.

Bruckhoff, C. and Dahm, P. (1998). Neural fields for local path planning. In *Proceedings of the International Conference on Intelligent RObotic Systems (IROS 98)*.

Collewijn, H. and Erkelens, C. (1990). Binocular eye movements and the perception of depth. In Kowler, E., editor, *Eye Movements and Their Role in Visual and Cognitive Processes*, pages 213 – 261. Elsevier Science Publishers B.V., Amsterdam.

Dahm, P., Bruckhoff, C., and Joublin, F. (1998). A neural field approach to robot motion control. In *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics (SMC'98)*.

Dahm, P. and Joublin, F. (1997). Closed form solution for the inverse kinematics of a redundant robot arm. Technical report, Institut für Neuroinformatik, Lehrstuhl für Theoretische Biologie, Ruhr-Universität Bochum.

DeValois, R. and DeValois, K. (1988). *Spatial Vision*. Oxford Psychology Series No. 14. Oxford University Press, Oxford.

Košecká, J. and Bogoni, L. (1994). Application of discrete event systems for modeling and controlling robotic agents. *Int. Conference on Robotics and Automa-*

tion, San Diego.

Lemmon, M., Stiver, J. A., and Antsaklis, P. J. (1993). Event identification and intelligent hybrid control. In Grossman, R. L., editor, *Hybrid Systems*, pages 269–296. Springer, Berlin, viii edition.

Ramadge, J. P. G. and Wonham, W. M. (1989). The control of discrete event systems. *Proceedings of the IEEE*, pages 81–97.

Rosenblatt, J. K. (1997). Damn: A distributed architecture for mobile nvigation. Technical Report CMU-RI-TR-97-01, Carnegie Mellon University, Robotics Institute.

Schöner, G. (1994). Dynamic theory of action-perception patterns: The time-before-contact paradigm. *Human Movement Science*, 3:415–439.

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

Steinhage, A. (1998). *Dynamical Systems Generate Navigation Behavior (Ph.D. thesis)*. Number ISBN 3-8265-3508-1 in Berichte aus der Physik. SHAKER-Verlag, Aachen, Germany.

Steinhage, A. and Bergener, T. (1998). Dynamical systems for the behavioral organization of an anthropomorphic mobile robot. In Pfeifer, R., Blumberg, B., Meyer, J., and Wilson, S., editors, *From Animals to Animats 5: Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior*, pages 147–152. The MIT Press/Bradford Books.

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