GMD Report 135

GMD –
Forschungszentrum
Informationstechnik
GmbH

Christian Verbeek

InFo: The Control of an Autonomous Mobile Robot by Merging Behavioral Intentions

June 2001

GMD –
Forschungszentrum Informationstechnik GmbH
Schloß Birlinghoven
D-53754 Sankt Augustin
Germany
Telefon +49 -2241 -14 -0
Telefax +49 -2241 -14 -2618
http://www.gmd.de

Anschrift des Verfasser/Address of the author:
Christian Verbeek
Institut für Autonome intelligente Systeme
GMD – Forschungszentrum Informationstechnik GmbH
D-53754 Sankt Augustin
E-Mail: christian.verbeek@gmd.de

ISSN 1435-2702

Abstract: This paper discusses the problem of behavioral decomposition of the control program for an autonomous mobile robot. We argue that the modularity of the control program of behavior based robots requires the behaviors' outputs to express the behaviors' long term goals or intentions. We present a novel cooperative behavior assembling mechanism which provides different interfaces to the behavior system and gives control signals in form of a linear and angular velocity. This mechanism combines potential fields on different levels and produces fast and robust activity in a tight interaction loop with the environment.

Keywords: autonomous mobile robot, behavior based control, cooperative behavior assembling mechanism

Zusammenfassung: In der vorliegenden Arbeit wird die Problematik der Modularisierung eines verhaltensbasierten Steuerungsprogrammes für einen autonomen mobilen Roboter behandelt. Es stellt sich heraus, dass die Modularität Ausgaben der einzelnen Verhalten erfordert, welche langfristige Ziele oder Intentionen wiederspiegeln. Dazu wird ein Mechanismus zur Verhaltensfusionierung vorgestellt, welcher den einzelnen Verhalten innerhalb des Verhaltenssystems unterschiedliche Schnittstellen anbietet und Ausgaben in Form einer Geschwindigkeit und einer Winkelgeschwindigkeit erzeugt. Dies wird über die Kombination von Potentialfeldern auf unterschiedlichen Ebenen erreicht, wodurch reaktives und robustes Verhalten erzeugt wird.

Schlüsselwörter: autonome mobile Roboter, verhaltensbasierte Steuerung, kooperative Verhaltensfusionierung

Contents

Chapter 1

Introduction

One of the main concerns of Behavior Based Robotics is the tight coupling of sensation and action. In general control programs following the behavior based paradigm which was first introduced by Arkin [1] and Brooks [8] divide into several modules. These modules are called behaviors [9], schemas [2] or action maps [23] and are the basic building blocks of the overall control program which enables the autonomous robot to fulfill its assign task. Each behavior has access to sensory information and produces output to control the robot's motion. Behavior based control programs can be divided into two mayor groups according to the method for combining and coordinating multiple behavioral activity streams.

Competitive methods use some kind of arbiter to choose the behavior from the behavior system whose output suits the current situation. The arbitration mechanism can often be viewed as a winner-take-all network in which the single response of the winning behavior is directed to the robots actuators. Examples for competitive approaches are Brooks Subsumption architecture [10],[7] in which the lower levels in the architecture have no awareness of the higher levels. The mechanisms of coordination are inhibition and suppression so that the output of a single behavior only is directed to the robots effectors. Maes Action selection Network [20],[21],[22],[27],[14] makes use of a winner-take-all algorithm to choose the controlling behavior. Kosecka and Bajcsy have developed a similar approach based on discrete event systems [18]. This method uses supervisory control to enable and disable the connection between behaviors and actuators.

The advantage of competitive methods is the concentration of control within a single unit during a short period of time. In case of malfunction of the robot the engineer only has to debug the behavior which had control while the error occurred. Disadvantages arise in dynamic environments or with fast moving robots where sensory input changes quickly. The arbiter has to switch between behaviors oftenly so that discontinuities in the stream of control signals send to the robots actuators arise. Another point of criticism is the necessity to encode all information for managing the current situation in the behavior which will be chosen. Redundance of the behaviors are the result since many skills like obstacle avoidance are necessary in a large variety of situations.

Cooperative methods compute the control signals for the robot's actuators by merging the outputs of the individual behaviors. Arbitration is replaced by a continues superposition so that control signals smoothly adapt to even fast changing situations. Examples for cooperative control architectures are Arkin's Motor Schemas [3],[4],[5], Myer's Saphira [16],[17], Steinhage's and Schöner's Dynamical Systems [25],[26] and last but not least Rosenblatt's Distributed Architecture for Mobile robot Navigation (DAMN) [24]. Each of these architectures makes use of a different method for merging the output of the individual behaviors within the behavior system. The Motor Schema approach is based on potential fields while the underlying mathematical method in Saphira is Fuzzy Logic. The dynamical Systems approach makes use of ordinary differential equations and DAMN superimposes functions rating the quality of possible control signals. Despite these formal differences all architectures suffer from the restriction that the dimension of the control signals send to the robot's actuators determines the interface between the individual behaviors and the behavior assembling mechanism. Every of the mentioned architectures has to find a compromise between controllability of the robot's motion via the generated control signals and need of finding a representation of behavioral response amenable to fusion. As it is shown in [28] today's cooperative approaches choose the linear velocity u and the robot's orientation ϕ for controlling a unicycle like mobile robot. This implies that individual behaviors have to generate the pair (u, ϕ).

Chapter 2

The InFo approach

The main disadvantage of today's behavior based cooperative architectures is the restriction of the output of the individual behaviors to an uniform interface to the behavior assembling mechanism. Therefor a behavior that is responsible for driving the robot to a certain position has to generate the same output as a behavior that tries to turn the robot with an exact angular velocity. The above mentioned approaches find a compromise by the vector (u, ϕ) to which the behaviors' real goals have to be converted.

Aside of this problem there is a need to regulate behavioral activity not only on the reactive layer by the use of behavioral output, but to introduce some kind of tactical layer that is able to resolve conflicts of behavioral goals [11]. In tradition of the Dual Dynamics Design Scheme [12],[13] the InFo approach defines the behavior activity α to introduce a minimal interface between individual and therefor intrinsically separated behaviors.

2.1 Behavioral activity

The activity of an individual behavior is computed via a differential equation of the form:

$$\dot{\alpha} = \sum_{i}^{L} [\kappa_i(t) \cdot (1 - \alpha)] - \sum_{i}^{M} (\lambda_i(t) \cdot \alpha) + \sum_{i}^{N} [\nu_i(t) \cdot (\mu_i(t) - \alpha)] \qquad (2.1)$$

The parameter κ is called a stimulus, λ is called a inhibition and μ is called a reference with strength ν. The activity of a behavior is influenced by L stimuli, M inhibitions and N references. The initial activity at time t_0 (when the robot is turned on the first time) is $\alpha(t_0) = 0$, i.e. all behaviors are inactive. The dynamical systems of the individual behaviors can be coupled by introducing functional correlation between κ, λ, μ or ν and the activity of a different behavior. By this means a network of dynamical systems arises that is able to produce sequences of active behaviors or to suppress behaviors while others are active.

To simplify the arrangement of behaviors in groups the real activity $\hat{\alpha}$ is introduced. As long as a behavior does not belong to a group the real activity is equal to the activity α otherwise

$$\hat{\alpha} = \hat{\boldsymbol{\alpha}} \cdot \alpha \tag{2.2}$$

with $\hat{\boldsymbol{\alpha}}$ being the real activity of the leading behavior.

2.2 Behavioral goals

To overcome the restriction of an uniform interface the behavior assembling mechanism within the InFo architecture makes four different interfaces available. These interfaces are:

- position (x, y)
- orientation (φ)
- linear velocity υ
- angular velocity ϖ

Thus a behavior responsible for reaching the robot's charging station can simply generate a position as output while a turn-behavior would generate an angular velocity. Thereby the development of behaviors is simplified and the behavior assembling mechanism collects behavioral output that reflects the behaviors' intentions.

Every position, orientation, linear or angular velocity generated by the behaviors' is a value $\hat{\gamma}$ assigned, called the real relevance. Therefor behaviors produce vectors of the form $\mathsf{PO} = (x, y, \rho_0, \hat{\gamma})^T$, $\mathsf{OO} = (\varphi, \hat{\gamma})^T$, $\mathsf{VO} = (\upsilon, \hat{\gamma})^T$ and $\mathsf{AO} = (\varpi, \hat{\gamma})^T$ called position-object, orientation-object, velocity-object and angular-velocity-object respectively. The parameter ρ_0 defines the distance to the aspired position (x, y) at which the robot slows down.

2.3 The real relevance

During every time step a behavior calculates proposals for the robot's motion in the form of k positions $(x_1, y_1) \dots (x_k, y_k)$, l orientations $\phi_1 \dots \phi_l$, m angular velocities $\varpi_1 \dots \varpi_m$ and n linear velocities $\upsilon_1 \dots \upsilon_n$. Every proposal is a relevance γ assigned. $\gamma = 0$ means the goal is irrelevant while $\gamma = 1$ shows maximum relevance. The relevances belonging to each type of proposal are summed giving Γ_{P}, which is the sum of the relevance of all produced PO. Further on the sum of relevances is characterized by Γ_{O} for the OO, Γ_{A} for the AO and Γ_{V} for the VO. In a second step four scaling factors are calculated each of which belonging to one type of proposal. If x stands for the type of proposal, the scaling factor is

$$\xi_{\mathrm{x}} = \begin{cases} 1 & \text{if } \Gamma_{\mathrm{x}} \leq 1 \\ \frac{1}{\Gamma_{\mathrm{x}}} & \text{else} \end{cases} \tag{2.3}$$

The real relevance used within the PO, OO, AO and VO is calculated by

$$\hat{\gamma} = \xi_{\mathrm{x}} \cdot \gamma \cdot \hat{\alpha} \tag{2.4}$$

when the relevance belongs to a goal of type x. $\hat{\alpha}$ is the behavior's real activity.

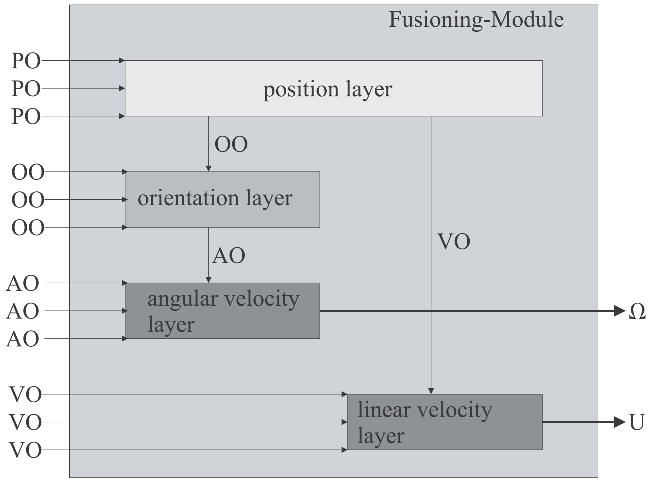

Figure 2.1: The Fusioning-Module (*FM*). On the left the interface to the behavior system is shown. PO are collected by the position layer. The positions within the PO contribute to a OO and VO generated by the position layer according to their real relevance. The orientation layer gives a AO from the orientations hold by the OO weighted by the real relevances. The angular velocity layer evaluates the weighted sum of angular velocities producing the control signal Ω. The velocity layer combines the VO in a similar way producing the second control signal U.

2.4 The behavior assembling mechanism

The InFo architecture provides a behavior assembling mechanism in form of the Fusioning-Module (*FM*). The *FM* collects all the PO, OO, AO and VO generated by the individual behaviors and calculates control signals which are send to the robot's actuators in form of a linear velocity U and an angular velocity Ω. The inner structure of the *FM* is shown in figure 2.1.

2.4.1 The position layer

Within the position layer the PO act as attractive potentials of the form

$$P(x, y, \mathsf{PO}) = \hat{\gamma} \cdot \begin{cases} \frac{1}{2\cdot\rho_0} \cdot \rho(x, y, x_z, y_z)^2 & \text{if } \rho < \rho_0 \\ \left(\rho(x, y, x_z, y_z) - \frac{\rho_0}{2}\right) & \text{else} \end{cases} \tag{2.5}$$

with

$$\rho(x, y, x_z, y_z) = \sqrt{(x - x_z)^2 + (y - y_z)^2}. \tag{2.6}$$

This potential combines the advantage of a parabolic well for $\rho < \rho_0$ and a conic well for $\rho \geq \rho_0$ [19]. The sum of these potentials define the artificial potential field

$$P_G(x, y, \mathsf{PO}_1, \mathsf{PO}_2, \dots, \mathsf{PO}_N) = \sum_i^K P_i(x, y, \mathsf{PO}_i) \tag{2.7}$$

whose local variations reflect the structure of the free space surrounding the robot. Following the idea of Khatib [15] the artificial force

$$-\text{grad}\, P_{\text{G}} = -\begin{pmatrix} f_x \\ f_y \end{pmatrix} \tag{2.8}$$

is used to plan the robot's motion during the next control loop cycle. To reach the aimed location an orientation object

$$\mathsf{OO} = \begin{pmatrix} \beta \\ \hat{\gamma} \end{pmatrix} \tag{2.9}$$

with

$$\beta = \text{atan2}(f_y, f_x) \tag{2.10}$$

and a velocity object

$$\mathsf{VO} = \begin{pmatrix} u \\ \hat{\gamma} \end{pmatrix} \tag{2.11}$$

with

$$u \sim \sqrt{f_x^2 + f_y^2} \cdot \cos\frac{\beta}{2} \tag{2.12}$$

is generated and send to the lower layers. In both cases the relevance of these goals is given by the sum of the real relevances from the PO

$$\hat{\gamma} = \sum_{i=1}^{K} \hat{\gamma}_i. \tag{2.13}$$

By this means the robot rotates towards the minimum of the artificial potential field P_{G} and moves with a positive linear velocity proportional to the strength of $\vec{\nabla} P_{\text{G}}$ and the cosine of half the polar angle β.

2.4.2 The orientation layer

This layer collects the OO produced by the behavior system and the OO produced by the position layer, so that the number of OO is $L+1$. Each OO is represented as a 1-dimensional potential

$$O(\Phi, \mathsf{OO}) = -\hat{\gamma} \cdot \cos(\Phi - \varphi). \tag{2.14}$$

The sum of these potential defines a potential field

$$\begin{aligned} O_{\text{G}}(\Phi, \mathsf{OO}_1, \mathsf{OO}_2, \ldots, \mathsf{OO}_L, \mathsf{OO}_{L+1}) &= \sum_{i=1}^{L+1} O_i \\ &= -\overbrace{\sum_{i=1}^{N} \hat{\gamma}_i \cos\varphi_i}^{a} \cdot \cos\Phi - \overbrace{\sum_{i=1}^{N} \hat{\gamma}_i \sin\varphi_i}^{b} \cdot \sin\Phi \\ &= -\sqrt{a^2 + b^2} \cdot \cos\left(\Phi - \text{atan2}(b, a)\right) \end{aligned} \tag{2.15}$$

with its only minimum in the interval $]-\pi,\pi]$ at

$$\varphi_{\mathrm{G}} = \mathrm{atan2}\left(\sum_i \hat{\gamma}_i \sin\varphi_i, \sum_i \hat{\gamma}_i \cos\varphi_i\right). \tag{2.16}$$

To go for this minimum a AO is generated

$$\mathsf{AO} = \begin{pmatrix} \omega \\ \hat{\gamma} \end{pmatrix} \tag{2.17}$$

with ω being proportional to $\varphi_{\mathrm{G}} \in]-\pi,\pi]$

$$\omega \sim \varphi_{\mathrm{G}} \tag{2.18}$$

and the relevance

$$\hat{\gamma} = \sum_{i=1}^{L+1} \hat{\gamma}_i. \tag{2.19}$$

2.4.3 The angular velocity layer

The behavior system produces M AO and the orientation layer one AO, so that the number of AO processed within the angular velocity layer is $M+1$. A single AO is represented by a 1-dimensional potential

$$A(\omega, \mathsf{AO}) = \hat{\gamma} \cdot (\omega - \varpi)^2. \tag{2.20}$$

The superposition of these potential forms a potential field

$$A_{\mathrm{G}}(\omega, \mathsf{AO}_1, \mathsf{AO}_2, \ldots, \mathsf{AO}_M, \mathsf{AO}_{M+1}) = \sum_{i=1}^{M+1} A_i(\omega, \mathsf{AO}_i) \tag{2.21}$$

with its only minimum at

$$\Omega = \frac{\sum_{i=1}^{M+1} \hat{\gamma}_i \varpi_i}{\sum_{i=1}^{M+1} \hat{\gamma}_i}. \tag{2.22}$$

The minimum Ω defines the angular velocity which is send to the robot's actuators.

2.4.4 The linear velocity layer

Similar to the angular velocity layer a single VO is represented by a 1-dimensional potential

$$V(u, \mathsf{VO}) = \hat{\gamma} \cdot (u - \upsilon)^2. \tag{2.23}$$

The minimum of the potential field

$$V_{\mathrm{G}}(u, \mathsf{VO}_1, \mathsf{VO}_2, \ldots, \mathsf{VO}_N, \mathsf{VO}_{N+1}) = \sum_{i=1}^{N+1} V_i(u, \mathsf{VO}_i). \tag{2.24}$$

can be found at

$$U = \frac{\sum_{i=1}^{N+1} \hat{\gamma}_i \upsilon_i}{\sum_{i=1}^{N+1} \hat{\gamma}_i}. \tag{2.25}$$

This is the linear velocity which is use as steering value for the robot's effectors.

Chapter 3

Experiments

The framework provided by the InFo architecture is used to build a control program for a virtual mobile robot. The use of a simulation instead of experiments with real robots is justified by utilization of sensory information which is also available on real robots. The virtual robot has 8 tactile sensors represented by the variables $t_1 \ldots t_8$ with $t_n \in [0;1]$. Furthermore 16 infrared sensors are simulated. These sensors give the distance in cm along a ray to the nearest object with an accuracy of ± 5cm. The maximum measurable distance is 200cm. If there is no object the infrared sensors, represented by $r_1 \ldots r_{16}$ give -1. The tactile and infrared sensors are mounted equally distributed on the robot's chassis. The numbering starts with the sensor mounted in driving direction $\phi = 0°$ and continues counterclockwise ending with t_8 at $\phi = 315°$ and r_{16} at $\phi = 337.5°$ respectively. According to a vision sensor used on the GMD-Soccer-Robots [6] the virtual robot can detect the distance and the relative orientation to a marked position in its workspace. The distance d is measured in cm with an relative error of $\pm 5\%$ while the orientation ϕ is measured in degrees with an absolute error of $\pm 5°$. A microphone enables the robot to detect music. This sensor is represented by the variable m. If no music is played $m = 0$ otherwise $m \in [1;2;3]$ representing three different styles named waltz, rumba and samba respectively.

Figure 3.1 shows a behavior system consisting of eight behaviors.

3.1 Obstacle avoidance

The behavior called *Avoid* will be discussed in detail since it shows how the InFo architecture is used efficiently. *Avoid* utilizes the 16 infrared sensors represented by $r_1 \ldots r_{16}$ to prevent the robot from crashing into the environment's borders or obstacles within the robot's workspace. The ulterior motive of this behavior is to turn the robot to a sensor's opposite direction if this sensor measures a distance not equal -1, i.e. an obstacle is detected within the sensor's reach. This most simple strategy is applied to any of the 16 sensors independently. To realize the turning orientation objects (OO) are generated referring to a single sensor. The sensor giving r_1 pointing at $\phi = 0°$ yields a OO with $(180°, \hat{\gamma}_1)$ while the sensor giving r_{12} pointing at $\phi = 247.5°$ yields $(67.5°, \hat{\gamma}_{12})$. To get the obstacle avoidance behavior started the real relevance $\hat{\gamma}$ of the 16 orientation objects has

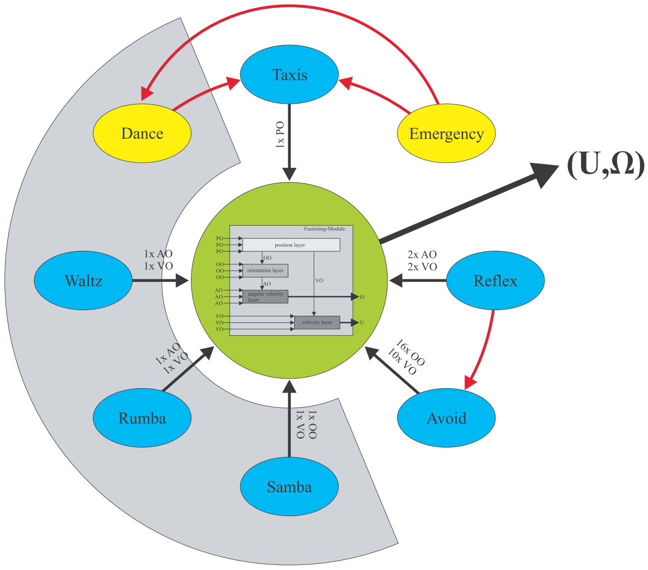

Figure 3.1: Eight individual behaviors are arrange around the Fusioning-Module. The behavioral output in the form of PO, OO, AO and VO is collected by the *FM* which calculates the steering command (U, Ω).

to be modulate with the distance measured by the individual sensors. If the measured distance is short the corresponding OO has to obtain high relevance while a long distance leads to a low relevance. If the sensor measures -1 the relevance should be zero. This correlation between sensory input and behavioral output can be modeled by

$$g(x) = \begin{cases} e^{-\left(\frac{x}{40\text{cm}}\right)^2} & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases} \tag{3.1}$$

where x is the distance measured by a infrared sensor. Table 3.1 shows the calculation of the orientation objects' relevance.

One can see from figure 3.1 that *Avoid* not only produces 16 OO but also generates 10 velocity objects (VO). This is mandatory to make *Avoid* an active behavior that not only turns the robot away from obstacles but also tries to keep a safety margin to obstacles. The number of VO is 10 because sensors pointing sidewards are not accounted. Table 3.2 shows the values for the velocity u and the relevance γ of the VO produced by *Avoid*.

3.2 Motion patterns

The eight individual behaviors enable the robot to drive to the position sensed by the virtual camera, to avoid obstacles and to perform motion patterns characterized as waltz, samba and rumba. Figure 3.2 shows the robot's performance in three situations. The situations differ from each other by the value sensed by the microphone. The figure 3.2a) refers to $m = 1$, figure 3.2b) to $m = 2$ and figure 3.2c) to $m = 3$ so that *Waltz*,*Rumba* and *Samba* are active respectively.

Sensor	Orientation	Relevanz
1	$\varphi_1 = 180$	$\gamma_1 = g(r_1)$
2	$\varphi_2 = 202.5$	$\gamma_2 = 0.95g(r_2)$
3	$\varphi_3 = 225$	$\gamma_3 = 0.9g(r_3)$
4	$\varphi_4 = 247.5$	$\gamma_4 = 0.5g(r_4)$
5	$\varphi_5 = 270$	$\gamma_5 = 0.5g(r_5)$
6	$\varphi_6 = 292.5$	$\gamma_6 = 0.5g(r_6)$
7	$\varphi_7 = 315$	$\gamma_7 = 0.9g(r_7)$
8	$\varphi_8 = 337.5$	$\gamma_8 = 0.95g(r_8)$
9	$\varphi_9 = 0$	$\gamma_9 = g(r_9)$
10	$\varphi_{10} = 22.5$	$\gamma_{10} = 0.95g(r_{10})$
11	$\varphi_{11} = 45$	$\gamma_{11} = 0.9g(r_{11})$
12	$\varphi_{12} = 67.5$	$\gamma_{12} = 0.5g(r_{12})$
13	$\varphi_{13} = 90$	$\gamma_{13} = 0.5g(r_{13})$
14	$\varphi_{14} = 112.5$	$\gamma_{14} = 0.5g(r_{14})$
15	$\varphi_{15} = 135$	$\gamma_{15} = 0.9g(r_{15})$
16	$\varphi_{16} = 157.5$	$\gamma_{16} = 0.95g(r_{16})$

Table 3.1: The sixteen OO generated by *Avoid*.

Sensor	Velocity	Relevanz
1	$u_1 = -100$	$\gamma_1 = g(r_1)$
2	$u_2 = -60$	$\gamma_2 = 0.95g(r_2)$
3	$u_3 = -20$	$\gamma_3 = 0.9g(r_3)$
7	$u_7 = 100$	$\gamma_7 = 0.9g(r_7)$
8	$u_8 = 100$	$\gamma_8 = 0.95g(r_8)$
9	$u_9 = 100$	$\gamma_9 = g(r_9)$
10	$u_{10} = 100$	$\gamma_{10} = 0.95g(r_{10})$
11	$u_{11} = 100$	$\gamma_{11} = 0.9g(r_{11})$
15	$u_{15} = -20$	$\gamma_{15} = 0.9g(r_{15})$
16	$u_{16} = -60$	$\gamma_{16} = 0.95g(r_{16})$

Table 3.2: *Avoid* produces 10 VO.

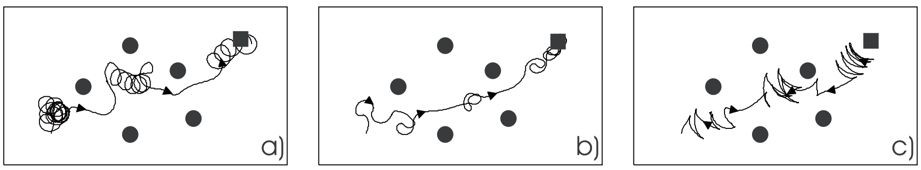

Figure 3.2:

The positions of the obstacles, the starting position and the goal position are the same in all three simulation runs. In figure 3.2a) one can see the robot spinning counterclockwise in the beginning clockwise in the middle of the way from starting to goal position and counterclockwise near the goal position. This spinning is initiated by the behavior *Waltz* which produces a AO to set the robot's angular velocity. in conjunction with a VO the robot drives circles with a radius of approximately 80cm. Together with *Waltz* the *Taxis*-behavior is active producing a PO to drive the robot to the goal position illustrated by the filled square in the upper right corner. The *Avoid*-behavior is also active and prevents the robot from crashing into obstacles since *Waltz* and *Taxis* have no awareness of this danger. The behaviors are designed individually and do not depend on each other so that *Waltz* can easily be substituted by *Rumba* in figure 3.2b). Like *Waltz* the *Rumba*-behavior produces a AO and a VO. The sign of the angular velocity changes on a much faster time-scale so that the robot's spinning direction changes every second. Again the three behaviors *Taxis*, *Avoid* and *Rumba* cooperate well so that the robot reaches the goal position without contact to obstacles. Figure 3.2c) shows the robot's path when *Rumba* is substituted by *Samba*. The *Samba*-behavior produces a OO and a VO to make the robot drive back and forth on a line orthogonal to the vector pointing from the robot to the goal position. Even with this handicap *Taxis* and *Avoid* are able to steer the robot safely to the goal position.

Chapter 4

Discussion

A behavior based architecture for controlling a mobile robot is presented that overcomes the restriction of a uniform interface between the behavior assembling mechanism and the individual behaviors. Thereby behaviors are able to produce control signals corresponding to the behavior's aspired goal and therefore control signals reflect the individual behaviors' intentions. Behavior assembling is distributed into four layers within the fusioning-module. Lower layers include the output of higher layers into the assembling process which is based on the superposition of artificial potential fields. The behavior assembling mechanism's output is a control signals in the form of a linear velocity U and an angular velocity Ω which are send to the robot's actuators. By this means accurate motion control is possible while the individual behaviors work with items like positions or orientations.

A behavior system consisting of eight individual behaviors is presented that shows the potentiality of the InFo approach. The robot is able to drive towards a position within its workspace while avoiding obstacles and performing complex motion patterns.

Bibliography

[1] R. C. Arkin. Path planning for a vision-based autonomous robot. In *Proceedings of the SPIE Conference on Mobile Robots*, pages 240–249, Cambridge, MA, 1986.

[2] R. C. Arkin. Motor schema based navigation for a mobile robot. *Proceedings of the IEEE International Conference on Robotics and Automation*, 1987.

[3] R. C. Arkin. Motor schema-based mobile robot navigation. *International Journal of Robotics Research*, 8(4):92–112, 1989.

[4] R. C. Arkin. Integrating behavioral, perceptual, and world knowledge in reactive navigation. *Robotics and Autonomous Systems*, 6:105–122, 1990.

[5] R. C. Arkin. Behavior-based robot navigation for extended domains. *Adaptive Behavior*, 1(2):201–225, 1992.

[6] A. Bredenfeld, T. Christaller, H. Guenther, J. Hermes, G. Indiveri, H. Jaeger, H.-U. Kobialka, P.-G. Plöger, P. Schoell, and A. Siegberg. Robocup-2000: Robot soccer world cup iv. Springer (to appear).

[7] R. Brooks. A hardware retargetable distributed layered architecture for mobile robot control. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 106–110, Raleigh, NC., May 1987.

[8] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, 1986.

[9] Rodney A. Brooks. A robot that walks; emergent behaviors from carefully evolved network. Technical report, Massachusetts Institute of Technology Artificial Intelligence Laboratory, 1989.

[10] Rodney A. Brooks. The whole iguana. In M. Brady, editor, *Robotics Science*, pages 432–456. MIT Press, Cambridge, MA., 1989.

[11] H. Hexmoor and D. Kortenkamp. Issues on building software for hardware agents. *The Knowledge Engineering Review*, 10(3):301,304, 1995.

[12] Herbert Jaeger. The dual dynamics design scheme for behavior-based robots: A tutorial. Technical Report 966, GMD - Forschungszentrum Informationstechnik GmbH, 1996.

[13] Herbert Jaeger and Thomas Christaller. Dual dynamics: Designing behavior systems for autonomous robots. In S. Fujimura and M. Sugisaka, editors, *Proceedings of the International Symposium on Artificial Life and Robotics (AROB '97)*, Beppu, Japan, February 1997.

[14] L. Kaelbling and S. Rosenheim. Action and planning in embedded agents. In P. Maes, editor, *Designing Autonomous Agents*, pages 35–48. MIT Press, Cambridge, MA., 1991.

[15] Omar Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal for Robotics Research*, 5(1):90–98, 1986.

[16] Kurt Konolige and Karen Myers. The saphira architecture for autonomous mobile robots. Technical report, Artificial Intelligence Center SRI International, Menlo Park, CA, 1996.

[17] Kurt Konolige, Karen Myers, and Enrique Ruspini. The saphira architecture: a design for autonomy. *Journal of experimental theoretical artificial Intelligence*, 9:215–235, 1997.

[18] Jana Kosecka and Ruzena Bajcsy. Discrete event systems for autonomous mobile agents. In *Proceedings Intelligent Robotics Systems 1993*, pages 21–31, Zakopane, July 1993.

[19] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[20] P. Maes. Situated agents can have goals. *Robotics and Autonomous Systems*, 6:49–70, 1990.

[21] Pattie Maes. How to do the right thing. *Connection Science Journal*, 1(3):291–323, 1989.

[22] Pattie Maes. Adaptive action selection. In *Proceedings of the Cognitive Science Conference '91*, Chicago, 1991.

[23] Jukka Riekki and Juha Röning. Reactive task execution by combining action maps. In *Proceedings IROS 97*, 1997.

[24] Julio K. Rosenblatt. DAMN: A distributed architecture for mobile navigation. *Journal of experimental Artificial Intelligence*, 9:339–360, 1997.

[25] A. Steinhage and G. Schöner. Dynamical systems for the behavioral organization of autonomous robot navigation. In *Sensor Fusion and Decentralized Control in Robotic Systems: Proceedings of SPIE*, volume 3523, pages 169–180. Schenker P S, McKee G T (eds.), 1998.

[26] Axel Steinhage. *Dynamical Systems for the Generation of Navigation Behavior*. PhD thesis, Universität Bochum, 1997.

[27] Toby Tyrrell. An evaluation of Maes's bottom-up mechanism for behavior selection. *Adaptive Behavior*, 2(4):307–348, 1994.

[28] Christian Verbeek. *Reaktive Steuerung autonomer mobiler Roboter*. PhD thesis, Universität Bielefeld, 2001.