GMD Report 106

GMD –
Forschungszentrum
Informationstechnik
GmbH

Christian Verbeek

Prevention of Local Equilibria in Behavior Based Navigation

July 2000

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: A technique for merging the outputs of many behaviors in a cooperative way is presented. The study of the correlation between different behavioral response vectors and the behavior system's overall properties shows that the representation of behavioral response has to be chosen carefully. Based upon this insight ideas from fuzzy control theory are used to develop a general model for behavior cooperation. By an appropriate choice of the parts involved in this model an algorithm for behavior fusion can be derived analytically. Simulations show that this approach yields a highly reactive and robust robot controller.

Keywords: behavior system, behavioral fusion, behavioral response, mobile robot, laserscanner, fuzzy set theory

Zusammenfassung: Es wird eine Methode zur kooperativen Fusionierung der Antworten vieler Verhalten vorgestellt. Die Untersuchung des Zusammenhanges zwischen unterschiedlichen Antwortvektoren und den globalen Eigenschaften des Verhaltenssystems zeigt, dass die Repräsentation der Antwortvektoren mit Bedacht gewählt werden muss. Auf dieser Grundlage wird die Theorie der unscharfen Mengen benutzt, um ein allgemeines Modell zur Verhaltensmischung zu erstellen. Durch geschickte Wahl der Komponenten dieses Modells kann ein Algorithmus zur Verhaltensüberlagerung analytisch hergeleitet werden. Simulationen belegen, dass die vorgestellte Art der Verhaltensüberlagerung einen hochreaktiven und gleichzeitig robusten Regler hervorbringt.

Schlüsselwörter: Verhaltenssystem, Verhaltensfusionierung, Verhaltensantwort, mobiler Roboter, Laserscanner, Theorie der unscharfen Mengen

1 Introduction

Complex control can be achieved by decomposing the control task into small units, each providing the robot with the ability to reach a certain goal. These units are called behaviors [4] or (motor) schemas [1]. To achieve complex goals, behaviors and their respective goals have to be assembled in some way. On the one hand there are competitive methods for behavioral response coordination ([5], [11], [9]). These methods avoid conflicts between active behaviors by arbitration, i.e. the selection of one behavior output ignoring the others. Even though arbitration theoretically ensures that the robot remains in a stable state all the time (provided that the single behaviors are designed correctly) the switching between behaviors is problematic. On the other hand there are cooperative behavior assembling methods ([13], [12], [8], [1]). These methods use the output of more than one behavior concurrently at a time. This overcomes the problem of switching between behaviors, but can result in an unwanted overall response, whenever conflicting behavioral goals have to be assembled.

In this paper a general approach of merging behavior response vectors based on the theory of fuzzy sets is described. Algorithms for merging linear velocities, angular velocities and reference directions are derived from these general results. We demonstrate, using a paradigmatic behavior system of a soccer playing robot, that the fusion of angular velocities causes locally stable equilibria, in which the robot gets trapped. This unwanted behavior results from conflicting goals of the competing behaviors within the behavior system. It is shown, that by replacing angular velocities by reference directions these equilibria vanish, because conflicts can be resolved using the more far-reaching temporal information carried by directions.

The first Section describes a novel approach for merging behavior response. In Section 3 a paradigmatic behavior system is presented in which the behaviors' response vectors consist of linear velocities and angular velocities. The fusion of these response vectors leads to locally stable equilibria. This problem is solved in Section 4 by replacing angular velocities by reference directions.

2 Fusion of behavioral response vectors

Most types of autonomous mobile robots move on a flat floor and thus have two degrees of freedom. The robots used in our work-group have two independently driven wheels. The wheels' speed is controlled by a computer program running on a Siemens C167 micro-controller. This is our low level motor controller that takes as reference a linear velocity and an angular velocity. A behavior system that is running on a PC with a 100MHz Intel-Pentium Processor mounted on the robot is sending these commands to the low level motor controller. Within the behavior system the designer is free in choosing any representation of control signals for the translational and rotational degree of freedom, as long as all behaviors are using the same representation. In general a behavior generates a control signal t_i for the translational degree of freedom and r_i for the rotational degree of freedom. The index i is to distinguish control signals generated by different behaviors. The t_i could be a desired acceleration or velocity of the robot. The r_i could be chosen as a torque, angular velocity or a direction. The behavior system has to merge all these values coming from the different behaviors, compute an overall response and perform a mapping of this overall response to the motor controller interface. In the following a general approach of merging the control signals of the single behaviors is presented.

The architecture for designing behavior systems used in our work-group is Dual-Dynamics [12]. Within this design scheme single behaviors produce a response vector consisting of the mentioned t_i and r_i and an activation value α_i. According to perceived stimuli behaviors compute their activation reflecting the necessity of taking control over the robot. An activation of zero is generated if the behavior should not influence the robots motion. On the other hand an activation of one assigns highest priority to the behavior. The idea of behavioral activation is described in detail in [7]. The behavioral response vector of the single behavior with index i then has the form:

$$\mathbf{r}_i = \begin{pmatrix} t_i \\ r_i \\ \alpha_i \end{pmatrix} \tag{1}$$

t_i : translational control signal computed by behavior i

r_i : rotational control signal computed by behavior i

α_i : activation of behavior i

To find an algorithm that merges the behavior response vectors appropriately a look on fuzzy set theory can help. Fuzzy sets have the nice property that operations like intersection (AND) and union (OR) of fuzzy sets are well defined. Cooperative behavior assembling can be interpreted as an intersection of behavioral response, i.e. the output of the behavior system is given by the output of behavior 1 AND behavior 2 AND ... AND behavior N, if the overall number of behaviors is N. The problem is, that this AND operation is not defined on response vectors. But by interpreting response vectors as a parameterization of fuzzy membership functions, the well defined fuzzy intersection operation can be performed. We propose that the response vector of behavior i defines two fuzzy membership functions, A_i referring to the translational and B_i referring to the rotational degree of freedom in robot motion. These functions describe the membership of the robot's current configuration to the configuration desired by a single behavior. In this context a configuration $\vec{q} = \begin{pmatrix} \tau \\ \rho \end{pmatrix}$ is a specification of the robot's acceleration/linear velocity and torque/angular velocity/orientation, depending on the chosen representation for the translational and rotational degree of freedom. The desired configuation of behavior i is given by $\vec{q_i} = \begin{pmatrix} t_i \\ r_i \end{pmatrix}$. This yields A_i and B_i to be functions of τ and ρ respectively, parameterized by α_i, t_i and r_i. Furthermore $A_i(\tau, \alpha_i, t_i)$ and $B_i(\rho, \alpha_i, r_i)$ have to respect the following constraints.

- Both functions are continuously defined on R^3 and assign to each element $x \in R^3$ a real number in the interval $[0, 1]$.
- A_i has its maximum at t_i, while B_i has its maximum at r_i. This means that a control state equivalent to the desired control state has maximum membership.
- The behaviors activation α_i has to influence the membership of control states unequal to the desired control state. If the behavior's activation is zero all possible control states must have maximum membership, since the behavior does not try to influence the robot's motion and therefore has no preferred control state.
- In the case of a maximum activation of the behavior, control states unequal to t_i and r_i should have rapidly decreasing membership with increasing distance to the desired control state.

$A_i(\tau, \alpha_i, t_i)$	$B_i(\rho, \alpha_i, r_i)$
$R^3 \rightarrow [0,1]$	$R^3 \rightarrow [0,1]$
$arg_{max}(A_i) = t_i \ \forall \alpha_i, t_i$	$arg_{max}(B_i) = r_i \ \forall \alpha_i, r_i$
$\forall \tau \ A_i(\tau, 0, t_i) = 1$	$\forall \rho \ B_i(\rho, 0, r_i) = 1$
$\forall \alpha_1 < \alpha_2, \ \tau \neq t_i :$ $A_i(\tau, \alpha_1, t_i) > A_i(\tau, \alpha_2, t_i)$	$\forall \alpha_1 < \alpha_2, \ \rho \neq r_i :$ $B_i(\rho, \alpha_1, r_i) > B_i(\rho, \alpha_2, r_i)$
	if $r_i = r_i + 2n\pi \ : \ B_i(\rho) = B_i(\rho + 2n\pi)$ $n = 0, \pm 1, \pm 2, \ldots$

Table 1: Properties of the membership functions A_i and B_i.

- If r_i is periodic, B_i has to be periodic too.

A short form of these properties is listed in Table 1. After this transformation of the response vectors of the single behaviors into two membership functions, behavioral fusion can now be performed by a fuzzy intersection process. In the end the behavior system must produce control signals that reflect the desires of all behaviors simultaneously. Therefore the intersection must be a fuzzy AND operation $\otimes$. With N being the number of single behaviors within the behavior system the intersection produces two new membership function given by:

$$C = A_1 \otimes A_2 \otimes \ldots \otimes A_N \quad (2)$$

$$D = B_1 \otimes B_2 \otimes \ldots \otimes B_N \quad (3)$$

Since we need a crisp output the information contained in C and D has to be compressed. This is done by a defuzzification process Δ. The control state produced by the whole behavior system, reflecting the desired control states of all behaviors simultaneously is then given by:

$$\tau_{final} = \Delta(C) \quad (4)$$

$$\rho_{final} = \Delta(D) \quad (5)$$

$$(6)$$

Depending on the chosen unit of the t_i and r_i, τ_{final} and ρ_{final} have to be mapped on the interface of the low level motor controller. It it assumed, that the low level motor controller takes a desired linear velocity U and a desired

angular velocity Ω as reference. For the case that the t_i are linear velocities and the r_i are angular velocities this mapping is trivially given by:

$$U = \tau_{final} \quad (7)$$
$$\Omega = \rho_{final} \quad (8)$$

If the r_i are directions measured in the robot's frame the mapping could be done by:

$$\begin{aligned} \Omega &= -\kappa\rho_{final} \quad (9) \\ \kappa &\in R^+ \\ \rho_{final} &\in]-\pi;\pi] \end{aligned}$$

In the following chapters a behavior system will be investigated making use of this mechanism for behavior fusion.

3 Linear and angular velocity as behavioral response

A behavior system is designed to demonstrate the ideas presented in the previous section in detail. The desired configuration of the single behaviors is given in terms of a linear velocity and an angular velocity. So we may replace the general variables t_i and r_i by u_i and ω_i which is the desired linear velocity and desired angular velocity of behavior with index i respectively. The behavior response vector of behavior i then has the form:

$$\mathbf{r}_i = \begin{pmatrix} u_i \\ \omega_i \\ \alpha_i \end{pmatrix} \tag{10}$$

u_i : desired linear velocity computed by behavior i

ω_i : desired angular velocity computed by behavior i

α_i : activation of behavior i

This vector parameterizes two functions A_i and B_i referring to the translational and the rotational degree of freedom in robot motion respectively. A suitable choice of the functions A_i and B_i, respecting the constraints given in Table 1 would be:

$$A_i(\tau, \alpha_i, u_i) = \exp(-\alpha_i(\tau - u_i)^2) \tag{11}$$

$$B_i(\rho, \alpha_i, \omega_i) = \exp(-\alpha_i(\rho - \omega_i)^2) \tag{12}$$

The shape of these functions for varying α_i is outlined in Figure 1. Both functions assign a real number in [0,1] to each $\tau \in R$ and $\rho \in R$ respectively. For all $\alpha_i \neq 0$ the maximum of these functions is at $\tau = u_i$ and $\rho = \omega_i$. The behavior's activation influences the form of these functions by sharpen the maximum for increasing α_i. In the case that the behavior's activation is zero both functions are constantly 1 for all τ and ρ. Beside the properties, which are also outlined in Table 1, A_i and B_i are continuous functions in all their arguments. This is necessary, since neighboring configurations must have neighboring membership to the behaviors desires. Otherwise discontinuities would occur, which lead to a discontinuous flow of control signals produced by the behavior system. The intersection of the A_i and B_i can be performed using the t-norm $t_{ap} = x \times y$, which is a fuzzy AND operator. The choice of this operator is guided by the necessity that the formed functions C and

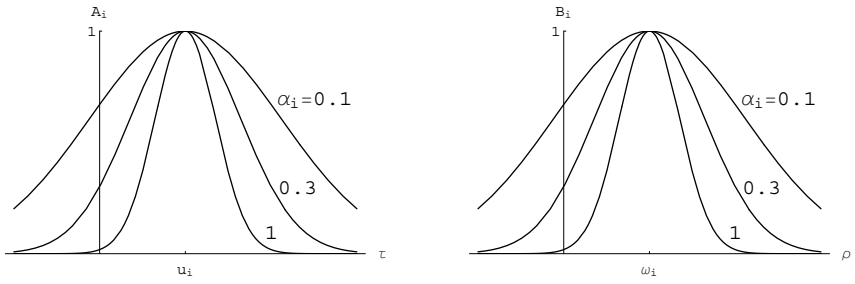

Figure 1: The behavior response vector r_i of behavior i parameterizes functions A_i and B_i which assign a real number in the interval $[0, 1]$ to all possible configurations of the robot. These functions are fuzzy membership functions that can be merged using a common fuzzy intersection operation.

D must be continuous functions in all arguments like the A_i and B_i. The functions formed by intersection are then given by:

$$C = \prod_{i=1}^{N} A_i(\tau, \alpha_i, u_i) \tag{13}$$

$$D = \prod_{i=1}^{N} B_i(\rho, \alpha_i, \omega_i) \tag{14}$$

Since the behavior system must generate a crisp output that can be fed into the low level motor controller, the information held by the functions C and D has to be compressed. This can be done by a defuzzification process Δ. The two most common defuzzification algorithms are the mean of maxima and the center of gravity algorithm [6]. It is shown in §1 that both algorithms produce the same output in this special case, so that we can set $\Delta(x) = arg_{max}(x)$. The argument that maximizes C is the linear velocity that reflects the desired linear velocities of all behaviors simultaneously. The same holds for the argument that maximizes D, which is the merged angular velocity. As shown in §2 there is no need to compute these values numerically. The merged linear velocity and angular velocity are simply given by:

$$\tau_{final} = arg_{max}(C) = \frac{\sum_{i=1}^{N} \alpha_i u_i}{\sum_{i=1}^{N} \alpha_i} \tag{15}$$

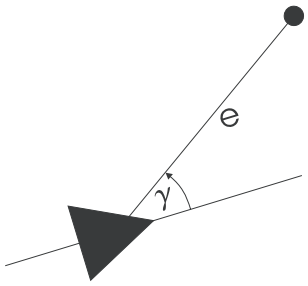

Figure 2: The length of the vector pointing from the robot (triangle) to the target (circle) is given by e. The angle between the x-axis of the robots frame and the vector pointing towards the target is given by γ.

$$\rho_{final} = arg_{max}(D) = \frac{\sum_{i=1}^{N} \alpha_i \omega_i}{\sum_{i=1}^{N} \alpha_i} \tag{16}$$

3.1 Experimental results

The behavior system that is used to demonstrate the properties of this method for behavior fusion contains three behaviors. The first behavior, which is called "Taxis", drives the robot to a presented target. This can be done by a control law like:

$$u_{Taxis} = 100 \cos\left(\frac{\gamma}{2}\right) \tanh(e) \tag{17}$$

$$\omega_{Taxis} = \gamma \tag{18}$$

$$\gamma \in [-\pi, \pi]$$

For the meaning of γ and e see Figure 2. A simulation run with the "Taxis" behavior being the only behavior within the behavior system is shown is Figure 3. This behavior works well as long as there are no obstacles, like other robots, on the playground. To enable the robot to handle situations in which it must avoid obstacles two behaviors called "AvoidLeft" and "AvoidRight" are put into the behavior system. Both behaviors make use of sensors measuring the distance from the robot to the next obstacle. The "AvoidLeft" behavior uses a sensor measuring distances on a line including an angle of 20° with the robot's frame x-axis. The sensor attached to "AvoidRight" measures distances on a line including an angle of −20° with the robot's

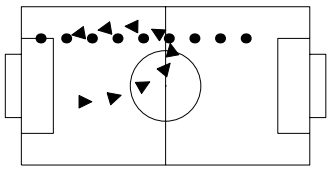

Figure 3: The robot (triangle) is initially placed in the lower left quarter of the playground. The target (circle) is placed in the upper right quarter of the playground. Nine snapshots of the simulation run are taken. The first snapshot is taken when the simulation starts. The target moves with constant velocity from the right to the left and the robot is driven by the "Taxis" behavior towards the target. Because of the constant movement of the target the robot cannot reach it, but follows the target in a constant distance.

frame x-axis. The measured distances are called d_{left} and d_{right}. These distance measurements are used to calculate two values that are almost zero if the distances are large and rise up to one if the measured distances are decreasing:

$$att_{left} = \exp\left(\frac{-d_{left}^2}{\sigma^2}\right) \tag{19}$$

$$att_{right} = \exp\left(\frac{-d_{right}^2}{\sigma^2}\right) \tag{20}$$

$$\sigma \in R^+$$

The parameter σ models the obstacles' size and is set to $170cm$ in this case. An obstacle avoidance behavior can the be implemented with "AvoidLeft" and "AvoidRight" having the following response vectors:

$$\mathbf{r}_{AvoidLeft} = \begin{pmatrix} 0 \\ -200 \\ \alpha_{AvoidLeft} \end{pmatrix} \tag{21}$$

The activation of "AvoidLeft" is computed via a differential equation:

$$\dot{\alpha}_{AvoidLeft} = 10(att_{Left} - \alpha_{AvoidLeft}) \tag{22}$$

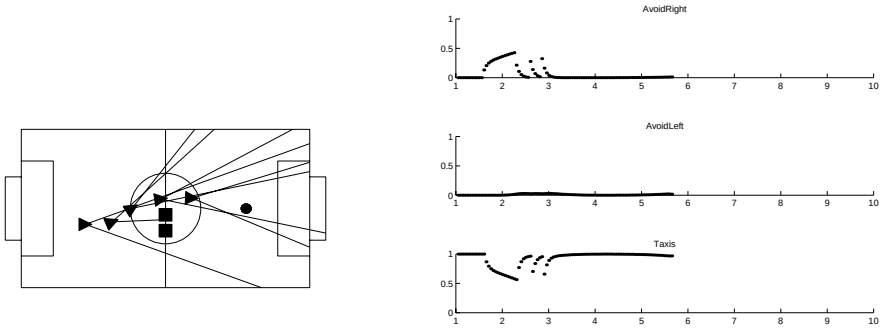

Figure 4: The robot is able to avoid the obstacles (squares) and reaches the target. At the beginning no obstacles are sensed. At this point in time "Taxis" has maximum activity, while the activation of the two "Avoid" behaviors is zero. At time-step two the right sensor detects an obstacle. The activation of "AvoidRight" rises, while "Taxis" is suppressed. According to the behavioral response of "AvoidRight" the robot turns to the left, until the obstacle is not sensed any more. The activation of "Taxis" rises again and the robot bypasses the obstacles.

The "AvoidRight" behavior is symmetrical to the "AvoidLeft" behavior:

$$\mathbf{r}_{AvoidRight} = \begin{pmatrix} 0 \\ 200 \\ \alpha_{AvoidRight} \end{pmatrix} \tag{23}$$

$$\dot{\alpha}_{AvoidRight} = 10(att_{Right} - \alpha_{AvoidRight}) \tag{24}$$

To ensure that the obstacle avoidance behaviors can take control over the robot the "Taxis" behavior is suppressed by these behaviors. This can be achieved by computing the activation of "Taxis" by:

$$\dot{\alpha}_{Taxis} = 20((1 - \alpha_{AvoidLeft})(1 - \alpha_{AvoidRight}) - \alpha_{Taxis}) \tag{25}$$

The performance of this behavior system consisting of the three behaviors "Taxis", "AvoidLeft" and "AvoidRight" is outlined in Figure 4. In the shown situation the ensemble of the three behaviors performs well. But two different situations will be presented in which the behaviors have conflicting goals and the robot gets stuck in a locally stable equilibrium. The first situation is outlined in Figure 5. After time-step four the behaviors "Taxis" and

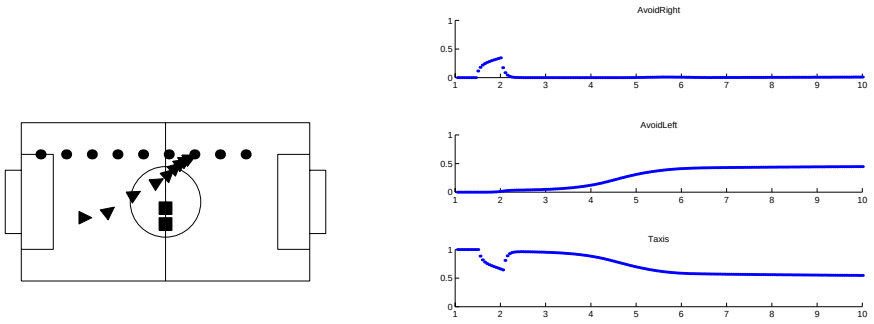

Figure 5: The robot starts from the lower left quarter of the playground. The target moves with constant velocity from right to the left. Between time-step one and two the robot senses the obstacles. The "AvoidRigth" behaviors activation rises and "Taxis" is suppressed. The robot turns to the left until the obstacles are not sensed any more. This condition is reached at time-step two. At time-step four the target and the robot reach the centerline. The "Taxis" behavior alone would turn the robot towards the target in this situation (see Figure 3). But the "AvoidLeft" behavior has an activity of about 0.2 and slightly turns the robot to the right. This is just the opposite direction the "Taxis" behavior tries to turn to. In the end the robot does not turn at all and continuous to move straight ahead.

"AvoidLeft" are in conflict. While "Taxis" produces a positive angular velocity, "AvoidLeft" tries to make a right turn and therefore produces a negative angular velocity. In the end the two behaviors cancel each other and the robot does not turn at all. At time-step nine the robot comes to a standstill. At this point the angular velocity produced by the behavior system is computed for different directions with the robot's direction at time-step nine as a reference. The result is shown in Figure 6. Due to the negative gradient of the function the equilibrium is stable. Even strong disturbances in the robot's direction cannot free the robot from this situation. A second situation in which the robot comes to a rest at a local equilibrium point is shown in Figure 7. The angular velocity produced by the behavior system at the last time-step shown in Figure 7 is plotted in Figure 8 for different directions relative to the robots direction. As in the previous example the robot is trapped in a stable equilibrium point.

When investigating properties of artificial systems, the critical reader will

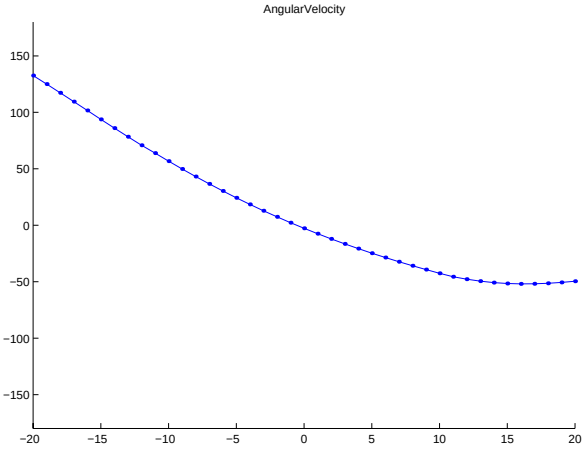

Figure 6: The angular velocity at different directions with respect to the robot's directions at time-step nine in Figure 5. The plotted function has a negative gradient at direction 0, i.e. disturbances in the robot's directions are compensated and the robot settles at direction 0.

remark, that the failure of these systems depends on errors in design, rather then on principal hindrances. This kind of critique is always true up to some point. Due to the enormous number of degrees of freedom in designing a behavior system, the situations in which the presented behavior system fails, could be solved by a different one. But all of these different behavior systems will suffer from the handicap that different behavior response vectors can cancel each other. This handicap is not only restricted to the presented algorithm for behavior merging, even though it is very clear in this case from the formulas. With control signals like desired linear velocity, desired angular velocity and activation contained in the behavior response vectors there is always a chance that one behavior tries to drive forward, while another one tries to drive backwards and the robot will stop. Of course one could do something like: "If the overall linear velocity is zero, then drive forward slowly." But such principals are not very nice, since they introduce discontinuities to the flow of control signals, and they contain hard-coded unflexible solutions, how the robot can escape from the equilibrium. The same holds for the rotational degree of freedom. One behavior tries to turn to the left and another one tries to turn to the right. In the end the robot does not

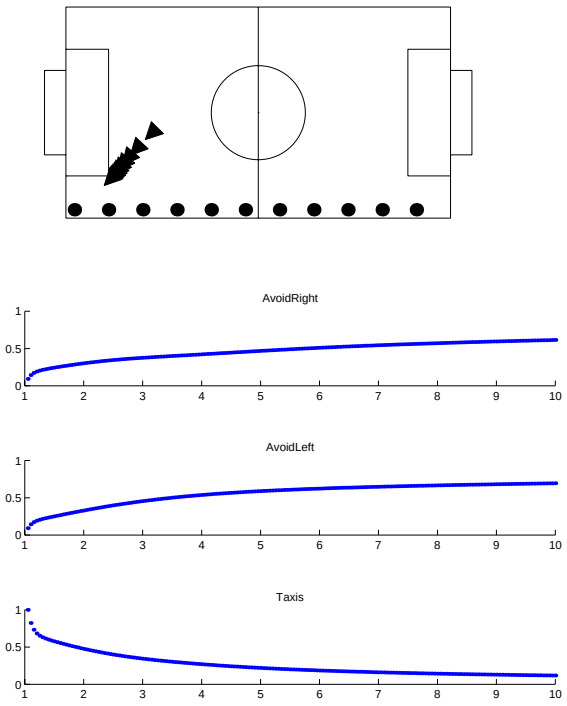

Figure 7: The robot's initial position is in the lower left quarter of the playground. It is facing the target which is positioned in the lower left corner. The target moves with constant velocity to the right. Driven by the "Taxis" behavior the robot approaches the target. As it gets nearer to the corner the two "Avoid" behaviors get active, since the distance sensors react on the playgrounds borders. The target moved to the right and "Taxis" tries to make a left turn. This rotation decreases the distance measured by the left sensor, so that "AvoidLeft" tries to compensate with a right turn. In the end both behaviors level each other and the robot does not turn at all.

turn at all, because the turning forces cancel each other. This problem is well known and designers apply turn-to-left or turn-to-right principals [14] to escape from this equilibrium.

But why are behaviors not able to cooperate in a convenient way? Why do conflicts arise in situations that should be managed easily? In the presented case the robot's intuition is to approach the the target, while bypassing obsta-

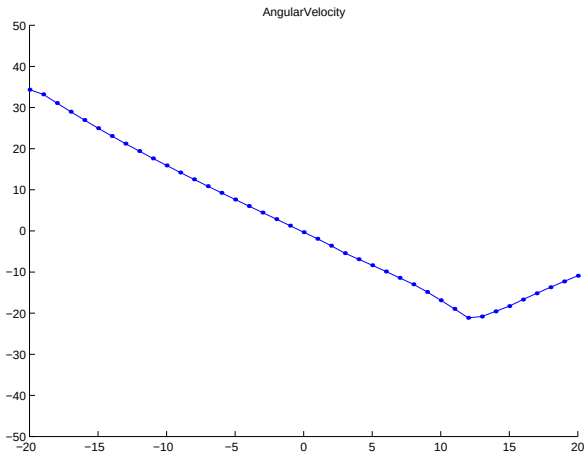

Figure 8: The angular velocity at different directions with respect to the robot's directions at the last time-step shown in Figure 7. The plotted function has a negative gradient at direction 0, i.e. disturbances in the robot's directions are compensated and the robot settles at direction 0.

cles if obstacles are sensed. The first goal of approaching the target is present all the time. The behavior "Taxis" is employed to translate this high level goal into action, i.e. generate the appropriate motor commands, so that the robot drives towards the target. Since in this case motor commands are the robot's linear velocity and angular velocity, "Taxis" is operating on a much faster time-scale as its goal is defined on. To reach the target the robot needs a time of order $10s$, but the motor commands vary on a time-scale of order $10^{-3}s$. As long as "Taxis" is the only behavior within the behavior system this discrepance does not matter, as it is shown above. By introducing the obstacle avoidance behaviors the robot has to fulfill a second goal simultaneously to the first one. The second goal is to avoid places where obstacles are detected. On this high level of goal definition the two goals of approaching the target and avoiding the obstacles are not contradictory to each other, unless the target would be placed on top of an obstacle which is not the case here. Any of the many know path planning algorithms (for an overview see [10]) can solve the presented situations easily. This is because path planning takes place on the level of abstraction where the goals of "Taxis" and the obstacle avoidance behaviors are defined. But working on this high level of

goal definition means, that path planning operates on the slow time-scale of order $10s$. This does not mean, that path planning algorithms are not able to compute new paths every few millisecond (which in fact is a matter of computational on-board power). But it turned out that classical path planners are not able to coupe with highly dynamic environments [2], because the notion of a path does not enable the robot to cope with uncertain or suddenly changing situations. By generating commands on a time-scale faster than the one on which the environment changes, a behavior based architecture should be able to ensure appropriate and smooth motions of the robot even in unforeseen situations. But by working on the fast time-scale one gets the problems outlined above that are not present when working on the slow time-scale. The task is to find a time-scale on which behaviors work, that is fast enough to handle dynamics in the environment and on the other hand slow enough to represent high level behavior goals. These two constraints interfere somehow with each other, but a reasonable solution (at least in this context) will be presented in the next chapter.

4 Linear velocity and direction as behavioral response

The kinematics of the mobile robots used in our work-group can be described by the Cartesian unicycle kinematics:

$$\begin{aligned} \dot{x} &= u\cos(\varphi) \\ \dot{y} &= u\sin(\varphi) \end{aligned} \tag{26}$$

$$\dot{\varphi} = \omega \tag{27}$$

being x and y the Cartesian coordinates of the robot, u its linear velocity in direction φ and ω its angular velocity. In the previous chapter behaviors generated linear velocities u_i and angular velocities ω_i to control the robot's motion. From the model one sees that these quantities define the robot's change in orientation via Equation 27 and the robot's change in position via Equation 26. But the robot's orientation and position can also be controlled by giving the linear velocity u and the orientation φ. Since ω is the time derivative of the robot's direction, φ changes on a much slower time-scale than ω does. This gives rise to the hope that the combination of linear velocity and orientation is more suitable for representing high level behavior goals, while ensuring that the behavior based robot controller still can handle dynamic environments. When exchanging angular velocities ω_i generated by the behaviors with reference orientations φ_i, one needs an algorithm to merge these reference directions. In this section an algorithm for merging reference directions generated by behaviors will be presented. This algorithm will be derived from the general ideas given in Section 2.

In contrast to the previous section the behavior response vector of behavior i now is given by:

$$\mathbf{r}_i = \begin{pmatrix} u_i \\ \varphi_i \\ \alpha_i \end{pmatrix} \tag{28}$$

u_i : desired linear velocity computed by behavior i

φ_i : desired direction computed by behavior i

α_i : activation of behavior i

Instead of angular velocities behaviors produce desired direction to control the rotational degree of freedom in robot motion. Therefore the membership

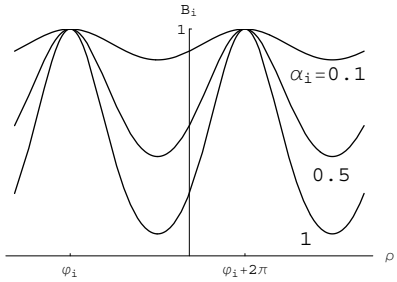

Figure 9: The membership function B is harmonic, due to the fact that the φ_i are harmonic. This leads to an infinite number of maxima at $\varphi_i + 2n\pi$ with $n = 0, \pm 1, \pm 2, \ldots$.

function B_i has to be redesigned. The most important difference between angular velocities and directions is, that directions are periodic, i.e. $\varphi_i = \varphi_i + 2n\pi$ with $n = 0, \pm 1, \pm 2, \ldots$. This means that B_i must respect the fifth property given in Table 1. A suitable form of B_i then is:

$$B_i(\rho, \alpha_i, \varphi_i) = \exp\left(-\alpha_i \sin^2\left(\frac{\rho - \varphi_i}{2}\right)\right) \tag{29}$$

The shape of this function is shown in Figure 9. The B_i are intersected by the t-norm algebraic product forming D. Due to the fact that the B_i are periodic, D is periodic too. This makes some problems for the defuzzification of D. The mean of maxima method is undefined, since D has an infinite number of maxima. The same holds for the center of gravity algorithm, since $\int_{-\infty}^{\infty} D(\rho)d\rho$ is undefined. But we can restrict our attention to the interval $]-\pi, \pi]$. If we do a mean of maxima defuzzification it is shown in §3, that the direction desired by the whole behavior system is simply given by:

$$\rho_{final} = atan2\left(\sum_{i=1}^{N} \alpha_i \sin\varphi_i, \sum_{i=1}^{N} \alpha_i \cos\varphi_i\right) \tag{30}$$

being $atan2(y, x) \in]-\pi, \pi]$ the four quadrant inverse tangent function (see §4). This formula has the simple geometric interpretation of ρ_{final} being the angle between the vector resulting from superposition of the behavior response vectors and the robot's x-axis (see Figure 10). The final mapping to the motor controller interface can be performed as shown in Equation 9.

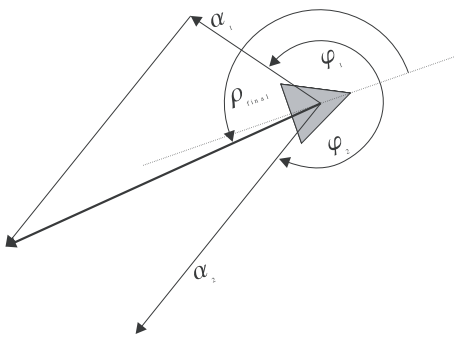

Figure 10: Two behaviors generate reference directions φ_1 and φ_2. The behaviors' activation is given by α_1 and α_2. This yields two vectors with length α_1 and α_2, including the angle φ_1 and φ_2 with the robot's x-axis. The final reference direction is computed by superimposing these two vectors, giving the final reference vector (bold drawn arrow). The final reference ρ_{final} is the angle between the final reference vector and the robot's x-axis.

4.1 Experimental results

To shown that a behavior system using directions as behavior response and the presented algorithm for behavior fusion has no chance to get trapped in local equilibriums any more, the situations investigated in the previous section are simulated again. The response vectors of the three behaviors "Taxis", "AvoidLeft" and "AvoidRight" have to be changed, so that the angular velocity is replaced by a desired direction. This yields the response vectors:

$$\mathbf{r}_{Taxis} = \begin{pmatrix} u_{Taxis} \\ \gamma \\ \alpha_{Taxis} \end{pmatrix} \tag{31}$$

The values u_{Taxis} and α_{Taxis} are calculated exactly like in Section 3.1. For the meaning of γ see Figure 2.

$$\mathbf{r}_{AvoidLeft} = \begin{pmatrix} 0 \\ -160^\circ \\ \alpha_{AvoidLeft} \end{pmatrix} \tag{32}$$

The direction desired by "AvoidLeft" is the direction in which the left sensor measures distances $+20^\circ - 180^\circ$. This means, that "AvoidLeft" desires to

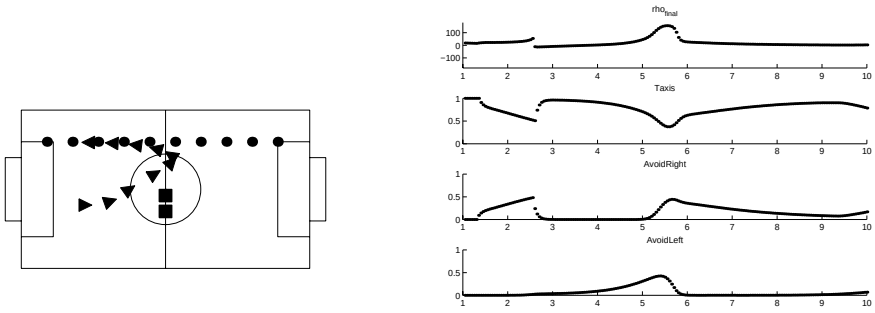

Figure 11: The initial condition of this simulation is similar to the situation shown in Figure 5. At time-step two the robot senses the obstacles. The activation of "AvoidRight" rises and the robot makes a turn to the left. Then the robot approaches the upper boundary of the playground and senses the barrier with its left sensor. Therefor the activation of "AvoidLeft" rises and "Taxis" is suppressed. But in contrast to Section 3 the robot continuous turning towards the target. This is because the reference direction generated by the behavior system ρ_{final} is positive all the time. Between time-step five and six ρ_{final} reaches a maximum of about 120°, which leads to a fast left turn. The activation of "AvoidLeft" decreases, while "AvoidRight" gets active. But the robot continuous turning towards the target and follows it in a short distance.

turn the robot away from the sensed obstacles. The activation is calculated as given in Equation 22.

$$\mathbf{r}_{AvoidRight} = \begin{pmatrix} 0 \\ +160° \\ \alpha_{AvoidRight} \end{pmatrix} \tag{33}$$

The second obstacle avoidance behavior desires to turn the robot away from the obstacles sensed by the right sensor. Therefore the desired direction is $-20° + 180°$. The activation is calculated as shown in Equation 24. Simulations using this modified behavior system are shown in Figure 11 and Figure 12. These simulations correspond to simulations done with the behavior system used in Section 3 shown in Figure 5 and Figure 7. In contrast to Section 3 now the robot shows the intended behavior. By merging directions the danger of local equilibriums is prevented, since the robot cannot

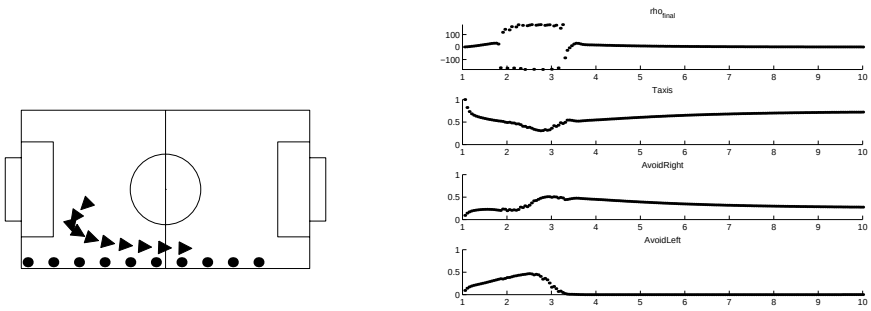

Figure 12: The initial condition of this simulation is identical to the situation shown in Figure 7. In the beginning the reference direction ρ_{final} is almost 0°, i.e. the robot drives straight ahead towards the target. By approaching the corner, the two obstacle avoidance behaviors get active and suppress the "Taxis" behavior. At time-step two the reference starts to oscillate between $-180°$ and 180°. This is because the $atan2$ function restricts ρ_{final} to $]-\pi, \pi]$. The problems that occur due to the discontinuities at the interval borders are addressed in general in [3]. Due to the target that moved to the left ρ_{final} is bend to positive directions. That leads to a left turn, so that the robot can escape from the corner and reach the target.

come to a standstill anymore. It follows the generated reference direction ρ_{final}, computing its angular velocity Ω by Equation 9. This means that an angular velocity of zero can only be reached if the target is straight ahead and no obstacles are sensed. This is reasonable, since the robot is expected to drive straight on only in this very special situation. If obstacles are sensed the reference direction ρ_{final} never can equal zero, because the "Avoid" behaviors bend the reference direction to positive or negative values. Even in the case that obstacles are sensed symmetrically to the left and to the right, as shown in Figure 12 in the corner situation, the reference direction will flip to 180°, when the activation of "AvoidLeft" and "AvoidRight" reaches a certain threshold. The robot turns on the spot and can escape from this problematic situation.

5 Conclusion

A novel approach for merging the response of behaviors has been presented in Section 2. These general results lead to merging algorithms for linear velocities, angular velocities and directions. The case study of a paradigmatic behavior system in Section 3 showed that merging angular velocities leads to locally stable equilibria, in which the robot gets trapped. This problem can be solved by replacing the angular velocity in the behaviors' response vectors by reference directions. In Section 4 a merging algorithm for directions was presented. The same behavior system that got trapped in certain situations was simulated again. It turned out that the merging of directions makes it impossible for the robot to get trapped in local equilibrium points any more.

A Appendix

§ 1 *Mean of maxima equals center of gravity. The functions A_i and B_i are both of the form:*

$$f_i(x) = \exp(-a_i(x - b_i)^2)$$

The functions C and D formed by the intersection process have the form:

$$g(x) = \prod_{i=1}^{N} f_i(x) = \exp(-\sum_{i=1}^{N} a_i(x - b_i)^2)$$

Using §7, yields:

$$g(x) = \exp(-a(x - b)^2 + k)$$

with a, b and k being constant. Therefore $g(x)$ has only one maximum at $x = b$. Because $g(x)$ is symmetric in x with respect to b the center of gravity is at $x = b$.

§ 2 *Merging linear and angular velocities give simple expressions for τ_{final} and ρ_{final}. The functions C and D formed by the intersection process have the form:*

$$g(x) = \prod_{i=1}^{N} f_i(x) = \exp(-\overbrace{\sum_{i=1}^{N} a_i(x - b_i)^2}^{h(x)})$$

The function $g(x)$ is maximum when $h(x)$ is minimum

$$\begin{aligned} h'(x_0) &= 2\sum_{i=1}^{N} a_i(x_0 - b_i) = 0 \\ &\Leftrightarrow \sum_{i=1}^{N} a_i x_0 - \sum_{i=1}^{N} a_i b_i = 0 \\ &\Leftrightarrow x_0 = \frac{\sum_{i=1}^{N} a_i b_i}{\sum_{i=1}^{N} a_i} \end{aligned}$$

It is shown in §1 that $g(x)$ has only one maximum and no other extrema. That yields that $g(x)$ is maximum at the point x_0.

§ 3 *Merging directions. Intersection of the membership functions for the rotational degree of freedom gives:*

$$D = \prod_{i=1}^{N} B_i = \prod_{i=1}^{N} e^{-\alpha_i \overbrace{\sin^2(\frac{\rho-\varphi_i}{2})}^{g(\rho)}}$$

Using the relations $\cos x = \frac{1}{2}(e^{ix} + e^{-ix})$ *and* $\sin x = \frac{1}{2i}(e^{ix} - e^{-ix})$*, conclude*

$$\begin{aligned} g(\rho) &= (\frac{1}{2i}(e^{i\frac{\rho-\varphi_i}{2}} - e^{-i\frac{\rho-\varphi_i}{2}}))^2 \\ &= -\frac{1}{4}(e^{i(\rho-\varphi_i)} - 2e^{i\frac{\rho-\varphi_i}{2}}e^{-i\frac{\rho-\varphi_i}{2}} + e^{-i(\rho-\varphi_i)}) \\ &= -\frac{1}{2}(\cos(\rho-\varphi_i) - 1) \end{aligned}$$

This yields

$$\begin{aligned} D &= \prod_{i=1}^{N} e^{\frac{1}{2}\alpha_i(\cos(\rho-\varphi_i)-1)} \\ &= e^{-\frac{1}{2}\sum_{i=1}^{N}\alpha_i} \cdot e^{\overbrace{\frac{1}{2}\sum_{i=1}^{N}\alpha_i\cos(\rho-\varphi_i)}^{h(\rho)}} \end{aligned}$$

D is maximum if $h(\rho)$ is maximum. The function $h(\rho)$ can be transformed into (see §6):

$$\begin{aligned} h(\rho) &= \sum_{i=1}^{N} \alpha_i \cos(\rho-\varphi_i) \\ &= \left(\sum_{i=1}^{N} \alpha_i \cos(\varphi_i)\right)\cos(\rho) + \left(\sum_{i=1}^{N} \alpha_i \sin(\varphi_i)\right)\sin(\rho) \end{aligned}$$

Using the relation $a\cos(x) + b\sin(x) = \sqrt{a^2+b^2}\cos(x - atan2(b,a))$*, which is proven in §4, conclude:*

$$h(\rho) = \sqrt{\left(\sum_{i=1}^{N} \alpha_i \cos(\varphi_i)\right)^2 + \left(\sum_{i=1}^{N} \alpha_i \sin(\varphi_i)\right)^2}$$

$$\times \quad \cos\left(x - atan2\left(\left(\sum_{i=1}^{N}\alpha_i \sin(\varphi_i)\right), \left(\sum_{i=1}^{N}\alpha_i \cos(\varphi_i)\right)\right)\right)$$

Since we are interested in the interval $]-\pi, \pi]$*, the argument that maximizes* D *is given by:*

$$arg_{max}(D) = atan2\left(\left(\sum_{i=1}^{N}\alpha_i \sin(\varphi_i)\right), \left(\sum_{i=1}^{N}\alpha_i \cos(\varphi_i)\right)\right)$$

§ 4 *Using Euler's equation and the trigonometric form of complex numbers, yields:* $a\cos x + b\sin x = \sqrt{a^2+b^2}\,\cos\left(x - atan2(b,a)\right)$

Proof:

$$\begin{aligned} & a\cos x + b\sin x \\ = \;& \frac{a}{2}\left(e^{ix} + e^{-ix}\right) + \frac{b}{2i}\left(e^{ix} - e^{-ix}\right) \\ = \;& e^{ix}\left(\frac{a}{2} - i\frac{b}{2}\right) + e^{-ix}\left(\frac{a}{2} + i\frac{b}{2}\right) \\ = \;& \frac{1}{2}\left(e^{ix}\sqrt{a^2+b^2}\,e^{-iatan2(b,a)} + e^{-ix}\sqrt{a^2+b^2}\,e^{iatan2(b,a)}\right) \\ = \;& \sqrt{a^2+b^2}\,\cos\left(x - atan2(b,a)\right) \end{aligned}$$

with $\sqrt{-1} = i$

§ 5 *The definition of the four quadrant inverse tangent function* $atan2$*.*

$$\begin{aligned} atan2(y,x) &= \arctan\frac{y}{x} + \gamma\pi \\ \gamma &= \begin{cases} 1 & \text{if } x < 0 \text{ and } y \geq 0 \\ 0 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \text{ and } y < 0 \end{cases} \end{aligned}$$

§ 6 $\sum_{i=1}^{N} a_i\cos(x - b_i) = \left(\sum_{i=1}^{N} a_i\cos(b_i)\right)\cos(x) + \left(\sum_{i=1}^{N} a_i\sin(b_i)\right)\sin(x)$
By induction: N=1:

$$\begin{aligned} & a_1\cos(x - b_1) \\ = \;& \frac{a_1}{2}\left(e^{i(x-b_1)} + e^{-i(x-b_1)}\right) \end{aligned}$$

$$
\begin{aligned}
&= \frac{a_1}{2}\left(e^{ix}e^{-ib_1} + e^{-ix}e^{ib_1}\right)\\
&= \frac{a_1}{2}\left(e^{ix}(\cos(b_1) - i\sin(b_1)) + e^{-ix}(\cos(b_1) + i\sin(b_1))\right)\\
&= \frac{a_1}{2}\left(e^{ix}\cos(b_1) - ie^{ix}\sin(b_1) + e^{-ix}\cos(b_1) + ie^{-ix}\sin(b_1)\right)\\
&= \frac{a_1}{2}\left(\cos(b_1)\overbrace{\left(e^{ix} + e^{-ix}\right)}^{2\cos(x)} - i\sin(b_1)\overbrace{\left(e^{ix} - e^{-ix}\right)}^{2i\sin(x)}\right)\\
&= a_1\cos(b_1)\cos(x) + a_1\sin(b_1)\sin(x)
\end{aligned}
$$

$N \geq 1$:

$$
\begin{aligned}
&\sum_{i=1}^{N+1} a_i\cos(x - b_i)\\
&= \left(\sum_{i=1}^{N} a_i\cos(x - b_i)\right) + a_{N+1}\cos(x - b_{N+1})\\
&= \left(\sum_{i=1}^{N} a_i\cos(b_i)\right)\cos(x) + \left(\sum_{i=1}^{N} a_i\sin(b_i)\right)\sin(x) + \ldots\\
&\quad a_{N+1}\cos(b_{N+1})\cos(x) + a_{N+1}\sin(b_{N+1})\sin(x)\\
&= \left(\sum_{i=1}^{N+1} a_i\cos(b_i)\right)\cos(x) + \left(\sum_{i=1}^{N+1} a_i\sin(b_i)\right)\sin(x)
\end{aligned}
$$

q.e.d.

§ 7 *For $N \geq 2$, conclude:* $\sum_{i=1}^{N} a_i(x-b_i)^2 = \left(\sum_{i=1}^{N} a_i\right)\left(x - \frac{\sum_{i=1}^{N} a_i b_i}{\sum_{i=1}^{N} a_i}\right)^2 + k_N$ *with*

$$k_N = \frac{a_N\sum_{i=1}^{N-1} a_i\left(\frac{\sum_{i=1}^{N-1} a_i b_i}{\sum_{i=1}^{N-1} a_i} - b_N\right)^2}{\sum_{i=1}^{N} a_i} + k_{N-1}$$

By indution: $N = 2$:

$$
\begin{aligned}
&a_1(x - b_1)^2 + a_2(x - b_2)^2\\
&\quad = a_1x^2 - 2a_1b_1x + a_1b_1^2 + a_2x^2 - 2a_2b_2x + a_2b_2^2\\
&\quad = x^2(a_1 + a_2) - 2x(a_1b_1 + a_2b_2) + a_1b_1^2 + a_2b_2^2
\end{aligned}
$$

$$
\begin{aligned}
&= (a_1+a_2)\left(x^2 - 2\frac{a_1b_1+a_2b_2}{a_1+a_2}x + \left(\frac{a_1b_1+a_2b_2}{a_1+a_2}\right)^2 - \left(\frac{a_1b_1+a_2b_2}{a_1+a_2}\right)^2\right) + a_1b_1^2 + a_2b_2^2 \\
&= (a_1+a_2)\left(x - \frac{a_1b_1+a_2b_2}{a_1+a_2}\right) + \overbrace{a_1b_1^2 + a_2b_2^2 - \frac{(a_1b_1+a_2b_2)^2}{a_1+a_2}}^{k_2}
\end{aligned}
$$

$$
\begin{aligned}
k_2 &= \frac{a_1b_1^2(a_1+a_2) + a_2b_2^2(a_1+a_2) - a_1^2b_1^2 - 2a_1a_2b_1b_2 - a_2^2b_2^2}{a_1+a_2} \\
&= \frac{a_1a_2(b_1-b_2)^2}{a_1+a_2} + \underbrace{k_1}_{=0}
\end{aligned}
$$

$N \geq 2$:

$$
\begin{aligned}
&\sum_{i=1}^{N+1} a_i(x-b_i)^2 \\
&= \left(\sum_{i=1}^{N} a_i(x-b_i)^2\right) + a_{N+1}(x-b_{N+1})^2 \\
&= \overbrace{\left(\sum_{i=1}^{N} a_i\right)}^{c}\left(x - \overbrace{\frac{\sum_{i=1}^{N} a_ib_i}{\sum_{i=1}^{N} a_i}}^{d}\right)^2 + k_N + a_{N+1}(x-b_N+1)^2 \\
&= (c+a_{N+1})\left(x - \frac{c\,d + a_{N+1}b_{N+1}}{c+a_{N+1}}\right)^2 + \overbrace{c\,d^2 + a_{N+1}b_{N+1}^2 - \frac{(c\,d+a_{N+1}b_{N+1})^2}{c+a_{N+1}} + k_N}^{k_{N+1}} \\
&= \left(\sum_{i=1}^{N+1} a_i\right)\left(x - \frac{\sum_{i=1}^{N+1} a_ib_i}{\sum_{i=1}^{N} a_i}\right)^2 + k_{N+1}
\end{aligned}
$$

$$
\begin{aligned}
k_{N+1} &= \frac{c\,a_{N+1}(d-b_{N+1})^2}{c+a_{N+1}} + k_N \\
&= \frac{a_{N+1}\sum_{i=1}^{N} a_i\left(\frac{\sum_{i=1}^{N} a_ib_i}{\sum_{i=1}^{N} a_i} - b_{N+1}\right)^2}{\sum_{i=1}^{N+1} a_i} + k_N
\end{aligned}
$$

q.e.d.

References

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

[2] R. C. Arkin. *Behavior-Based Robotics*. 1998.

[3] Sanjay P. Bhat and Dennis S. Bernstein. A topological obstruction to continuous global stabilization of rotational motion and the unwinding phenomenon. *Systems & Control Letters*, 39:63–70, 2000.

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

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

[6] Martin Brown and Chris Harris. *Neurofuzzy Adaptive Modelling and Control*. Prentice Hall International (UK) Limited, 1994.

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

[8] 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.

[9] David Jung and Alexander Zelinsky. An architecture for distributed cooperative planning in a behavior-based multi-robot system. *Robotics and Autonomous Systems*, 26:149–174, 1999.

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

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

[12] D. Payton, D. Keirsey, D. Kimble, J. Krozel, and J. Rosenblatt. Do whatever works: A robust approach to fault-tolerant autonomous control. *Applied Intelligence*, 2(3):225–250, September 1992.

[13] A. Saffiotti, K. Konolige, and E. Ruspini. A multivalued logic approach to integrating planning and control. *Artificial Intelligence*, 76(1-2):481–526, July 1995.

[14] W. L. Xu. A virtual target approach for resolving the limit cycle problem in navigation of a fuzzy behavior-based mobile robot. *Robotics and Autonomous Systems*, 30:315–324, 2000.