

# Simultaneous planning and action: neural-dynamic sequencing of elementary behaviors in robot navigation

Adaptive Behavior  
2015, Vol. 23(5) 243–264  
© The Author(s) 2015  
Reprints and permissions:  
sagepub.co.uk/journalsPermissions.nav  
DOI: 10.1177/1059712315601188  
adb.sagepub.com



Erik Billing<sup>1</sup>, Robert Lowe<sup>1,2</sup> and Yulia Sandamirskaya<sup>3,4</sup>

## Abstract

A technique for simultaneous planning and action based on dynamic field theory is presented. The model builds on previous work on representation of sequential behavior as attractors in dynamic neural fields. Here, we demonstrate how chains of competing attractors can be used to represent dynamic plans towards a goal state. The present work can be seen as an addition to a growing body of work that demonstrates the role of dynamic field theory as a bridge between low-level reactive approaches and high-level symbol processing mechanisms. The architecture is evaluated on a set of planning problems using a simulated e-puck robot, including analysis of the system's behavior in response to noise and temporary blockages of the planned route. The system makes no explicit distinction between planning and execution phases, allowing continuous adaptation of the planned path. The proposed architecture exploits the dynamic field theory property of stability in relation to noise and changes in the environment. The neural dynamics are also exploited such that *stay-or-switch* action selection emerges where blockage of a planned path occurs; stay until the transient blockage is removed *versus* switch to an alternative route to the goal.

## Keywords

Dynamic field theory, goal directed behavior, simultaneous planning and action

## 1 Introduction

The question of how an agent represents and selects actions in order to reach a behavioral goal runs deep in the fields of cognitive science and adaptive behavior, and it is an important problem for both biological and artificial cognitive systems. This problem has been approached by breaking down the behavioral repertoire of the agent into *Elementary Behaviors (EBs)*: “motor primitives” or “schemata” that constitute the basic action vocabulary of a cognitive controller (Mataric, 2000). This idea takes root in the early work of Arbib (1985) on distributed motor control and is central in behavior-based robotic architectures (Arkin, 1998; Brooks, 1991). While the exact formulation varies significantly between different bodies of work (Billing and Hellström, 2010; Mataric, 1997; Nicolescu, 2003; Tani, Ito and Sugita 2004) almost all formulations include both sensor and motor aspects of the behavior in each elementary behavioral unit. Each EB must be initiated at the right time, and there is also a need to decide when, and if, a behavior has been successfully executed. Moreover, such a distributed controller needs to

coordinate competing behaviors and elaborate a sequence of behaviors that leads to a long-term goal.

One way to approach this problem is to formulate EBs as attractor patterns in a dynamical system that may be coupled to the agent's sensorimotor dynamics. Such an approach has an advantage that the planning decisions about selecting, activating, and terminating EBs, as well as the sensorimotor dynamics of the individual behaviors, are formulated within the same computational substrate of attractor dynamics, which leads to a more homogeneous architecture. Our work builds on one such formulation (Sandamirskaya & Schöner, 2010; Sandamirskaya, Richter, & Schöner, 2011) within the framework of *Dynamic Field Theory (DFT)*. DFT

<sup>1</sup>Interaction Laboratory, University of Skövde, Skövde, Sweden

<sup>2</sup>Department of Applied IT, University of Gothenburg, Sweden

<sup>3</sup>Institute of Neuroinformatics University of Zurich, Zurich, Switzerland

<sup>4</sup>Institute of Neuroinformatics, ETH Zurich, Zurich, Switzerland

## Corresponding author:

Erik Billing, Interaction Laboratory, University of Skövde, SE-Box 408, 541 28, Skövde, Sweden.  
Email: erik.billing@his.se

is a mathematical and conceptual framework, in which cognitive architectures may be built based on principles of neuronal dynamics. In earlier work, we considered the *serial ordering* of actions in time (Sandamirskaya & Schöner, 2010) and the flexibility in the order of actions, termed *behavior organization* (Richter, Sandamirskaya, & Schöner, 2012), introducing EBs within the DFT framework (Sandamirskaya et al., 2011). Here, we extend the architectures for sequence generation to a distributed dynamical controller capable of planning and executing a sequence of EBs that leads to a behavioral goal. To study the functionality of the new architecture, we implement a path planning scenario for a simulated robotic vehicle as an example of a more general planning and search capability. Using the extended DFT-based EB formulation, we enable path search and planning towards a given goal, robust acting out of the planned trajectory, and updating of plans if the environmental situation changes. To our knowledge, this work represents the first attempt to utilize the neural-dynamic DFT framework for planning and search.

The focus of the present work concerns linking the lower level sensorimotor dynamics to higher cognitive processes as they manifest themselves in planning and search. This is achieved through a biased competition between the currently viable action candidates and, simultaneously, between the future action candidates. With this approach, the local sensory-driven (reactive) behaviors are integrated with the “long-term” plan, represented as an attractor pattern over a number of neural fields. The approach presented here can handle the problem of shallow gradients, which is otherwise common in, e.g. potential field approaches. In the proposed architecture, attractors are sustained during action, both at a sensory-motor level and at the planning level, allowing for robust performance within a changing environment. This feature also permits long trajectories to remain stable in the face of transient sensory perturbations. Furthermore, the DFT approach used here has the potential to maintain many plans and sub-plans in parallel whilst simultaneously carrying out actions. To this end, we refer to this approach as implementing *Simultaneous Planning and Action (SPA)*. This absence of separation of planning (as well as path extraction) and execution phases, we suggest, promotes great flexibility and robustness of performance of the robotic agent whilst not severely compromising on measures of optimality (e.g. shortest path finding and maintaining safe distances from obstacles).

To the authors’ knowledge, no DFT-based system has previously been proposed that searches for a sequence of actions towards a goal, in a way that can be compared to classical artificial intelligence (AI) formulations of planning (as we do in the discussion). While efficient and optimal search may have been

overemphasized by classical AI in explaining human and animal cognition, search is without question a key component in at least higher-level cognitive functions, such as planning. For this reason, we believe that a DFT formulation of search and planning can contribute to a better understanding of the link between the lower-level sensorimotor processes and higher-level cognitive functions. It may also shed light on the pros and cons of the dynamical, sensorimotor, or embodied approaches in relation to classical AI-formulations of search.

The rest of the paper breaks down as follows. After introducing the basics of DFT in Section 2, we present the DFT architecture able to plan a sequence of actions leading towards a goal in Section 3. The presented architecture should be understood as a generalization of previous work on sequence representation (Sandamirskaya et al., 2011; Sandamirskaya & Schöner, 2010). Three test cases applying the architecture as a planning and control system for a simulated e-puck robot are presented in Section 4. While the presented work should be understood as a generic search architecture, we make an effort to present a complete system, where inputs and outputs interface directly with the world. Results from simulated robot experiments are given in Section 5. Here, we focus solely on planning within an existing state space rather than on problems of localization and mapping (i.e. construction of a cognitive map of the environment). A discussion of possible connections to animal planning and learning abilities, as well as of the relation of our work to AI planning and search algorithms are presented in Section 6. Finally, conclusions are given in Section 7.

## 2 Methodological background: dynamic neural fields

DFT (Schöner, 2008) is today a well established neurally-based framework in cognitive science, used to model various perceptual (Johnson, Spencer, & Schöner, 2008; Zibner, Faubel, Iossifidis, & Schöner, 2011), motor (Bastian, Schöner, & Riehle, 2003; Schöner, Kopecz, & Erlhagen, 1997), and cognitive functions (Sandamirskaya, Zibner, Schneegans, & Schöner, 2013; Schöner, 2008; Spencer & Schöner, 2003). DFT is often presented as a bridge between the sensorimotor levels of neural processing and levels that relate to cognitive processes (Spencer, Perone, & Johnson, 2009).

In the language of DFT, the state of a cognitive system is characterized by dynamic activation functions, *Dynamic Neural Fields (DNFs)*. A DNF is a mathematical formulation of the neuronal activation at population level, taking root in the mean-field approximation of the activation dynamics in biological neuronal networks (Amari, 1977; Ermentrout, 1998; Wilson & Cowan, 1973). DNFs are defined over behaviorally

relevant dimensions, for instance, continuous perceptual features (color, orientation, or location) or motor parameters (joint position, velocity, or force), or discrete cognitive dimensions (serial order, object labels). The activation of a DNF evolves in time according to equation (1), first analyzed by Amari (1977). This dynamical system equation has an attractor solution of a particular shape, a localized bump of positive (supra-threshold) activation. This bump, or peak, is stabilized by the recurrent interactions in the DNF. The position of the supra-threshold activation bump on the DNF's dimension specifies the content of the respective representation, i.e. which parameter values characterize the current state of the cognitive system. The strength of the activation bump expresses the certainty of the system in the current estimation of the behavioral parameter. Such localized activity bumps are units of representation in cognitive architectures, built with DNFs (Sandamirskaya et al., 2013; Schöner, 2008) they represent perceptual objects, motor intentions, or plans.

The temporal dynamics of the activation function of a DNF,  $u(x, t)$ , is defined over a continuous behavioral dimension,  $x$

$$\tau \dot{u}(x, t) = -u(x, t) + h + \int f(u(x', t)) \omega(x - x') dx' + S(x, t) \quad (1)$$

where  $\tau$  is the time constant of the field dynamics,  $t$  is time,  $h < 0$  is the resting level that ensures that without external input the field is subthreshold, i.e. at a negative

activation level, and  $S(x, t)$  represents the external stimuli. The connectivity function  $\omega(x - x')$  is a bell-shaped kernel representing a short-range excitatory ( $c_{exc}$ ), a longer-range inhibitory ( $c_{inh}$ ), and a globally inhibitory ( $c_{global}$ ) inflows from other activated locations in the field

$$\omega(x - x') = c_{exc} \exp\left[-\frac{(x - x')^2}{2\sigma_{exc}^2}\right] - c_{inh} \exp\left[-\frac{(x - x')^2}{2\sigma_{inh}^2}\right] - c_{global} \quad (2)$$

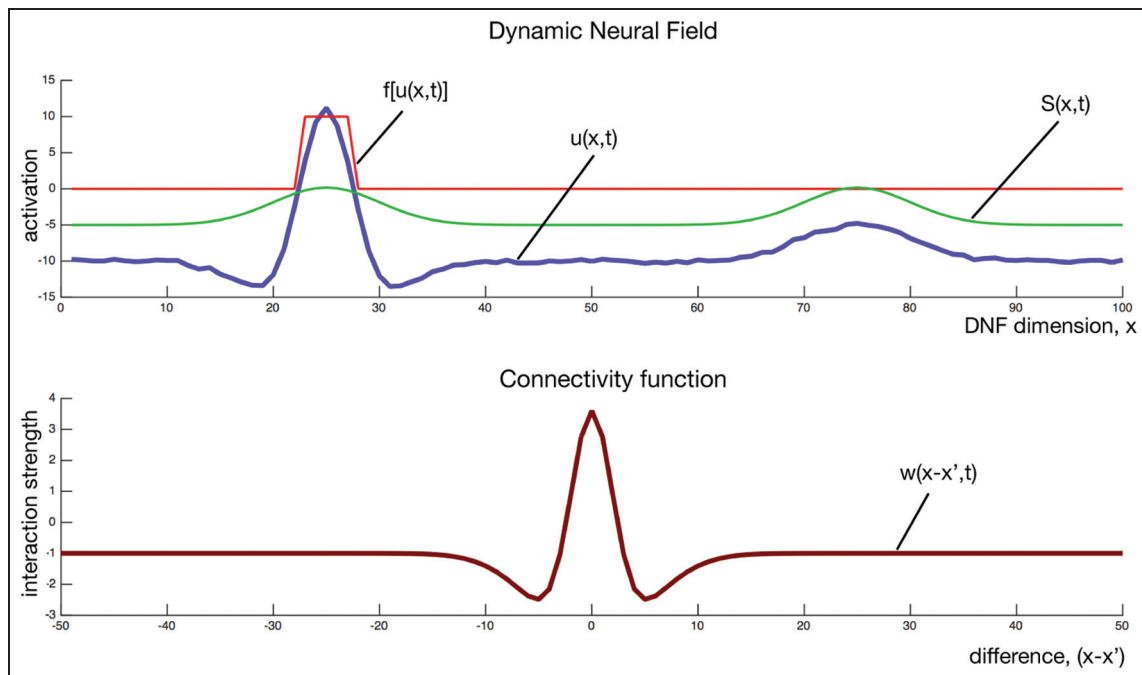
where  $\sigma$  represents the width of the kernel.

The output of the DNF is shaped by a sigmoidal non-linearity,  $f$ , that determines which locations of the DNF provide input to other locations in the field, and possibly other connected fields

$$f(u(x, t)) = \frac{1}{1 + \exp[-\beta u(x, t)]} \quad (3)$$

where  $\beta$  is the slope of the sigmoidal non-linearity.

Figure 1 shows a one-dimensional DNF, described by equation (1), and the respective lateral interactions kernel, equation (2). The external input,  $S(x, t)$ , to the field (green line in the plot) has two regions of higher strength, in which the activation level of the DNF (blue line in the plot) reaches the activation threshold. The lateral interactions of the DNF, described by the kernel level depicted in the lower part of the figure, “pull” a localized activity peak over one of the regions



**Figure 1.** An exemplary one-dimensional DNF (equation (1); top) and the respective interaction kernel (equation (2); bottom). The activation level  $u(x, t)$  (blue line), sigmoid output  $f(u(x, t))$  (red line), and external input to the field  $S(x, t)$  (green line), are shown.

with high input strength (the one that first reached the activation threshold), inhibiting the DNF at other locations. This inhibition also suppresses activation in the second region with high input strength.

Thus, through its dynamics, the DNF in Figure 1 made a *decision*, selecting one of the two regions in the input distribution and stabilizing this decision by lateral interactions. This decision will have impact on downstream structures in a DNF architecture through coupling of this field to other DNFs and, ultimately, to the sensory-motor dynamics.

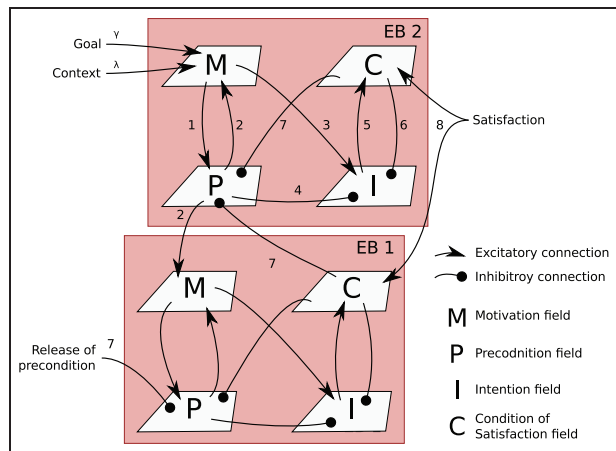
A different parameterization of the interaction kernel, with  $c_{global} = 0$ , would lead to a DNF that builds two localized activity peaks if stimulated with the same input as the one, depicted in Figure 1. If the excitatory part of the interaction kernel is strong enough, on the other hand, the activity peak may remain supra threshold even when the initial input to the field has vanished. Cognitive neural-dynamic architectures can be built with DNFs with different parameterizations and have been used both to model human cognitive behavior (Johnson, Spencer, & Schöner, 2009) and to control cognitive robots (Sandamirskaya et al., 2013).

### 3 The neural-dynamic SPA architecture

#### 3.1 The general framework

In previous work, Sandamirskaya and Schöner (2010) and Sandamirskaya et al. (2011) used the notion of EB to describe a discrete functional component representing some part of the complete behavioral repertoire of the agent; e.g. “look for an object with color X” is one possible EB, which implements search and approach (or gaze) towards an object of a given color, “move arm to position Y” is another typical EB. Each EB is characterized by two neural-dynamic structures, related to control of action initiation and termination: *intention (I)* and *condition of satisfaction (C)*, each represented by a DNF (Sandamirskaya et al., 2011). When the I-field is activated, it impacts the downstream dynamical structures, connected to the sensors and actuators of the robot, and the intended behavior is executed. Moreover, it pre-activates (or pre-shapes) the C-field to be sensitive to the desired sensory outcome of the behavior. When the sensory conditions are perceived that correspond to a successful accomplishment of the current action, the C-field becomes active, signalling that the behavioral goal is achieved. The C-field also inhibits the I-field and deactivates the EB.

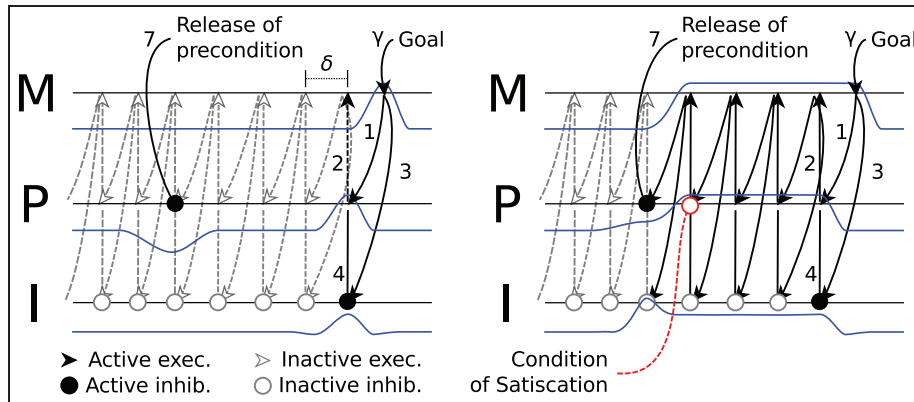
Sadamirskaya et al. (2011) demonstrated how this structure enabled autonomous organization of simple action sequences. To encode order of actions in a sequence, nodes representing rules of behavioral organization were introduced. In particular, the *precondition (P)* node inhibits the initiation of an EB that requires certain conditions to be fulfilled in order to be



**Figure 2.** Schematics of the main components of the proposed architecture. A sequence of two elementary behaviors (EB1 followed by EB2) is displayed. A goal ( $\gamma$ ) is introduced as a motivation (input to the M-field) to execute EB2, given some contextual information ( $\lambda$ ) and that some preconditions (represented as activity in the P-field) are fulfilled. Numbers on some connections are added to ease reference from the main text (Section 3).

activated. The P node is inhibited, in its turn, by an activated condition of satisfaction (C) node of a different EB. Sandamirskaya et al. (2011) have demonstrated how different action sequences may be encoded in this architecture and activated by selecting one of the task nodes. A task node boosts all EBs and precondition nodes, involved in achieving the selected behavioral goal. As a result, a sequence of behaviors unfolds, leading to accomplishment of a task, such as grasping, pointing, or lifting an object.

Here, we present a planning architecture that consists of four main parts, depicted in Figure 2 for two connected EBs. Analogous to previous work, there is *intention (I)*, *condition of satisfaction (C)*, and *precondition (P)* components, implemented as DNFs, with a fourth component, *motivation (M)*, added. M should be seen as a goal representation, feeding activity to both P (connection 1, Figure 2) and I (connection 3). Supra threshold activity in the I-field initiates execution of the EB, but also parametrization of the sensorimotor system during the particular action (e.g. location of the motor goal). There is consequently a close link between the goal representation (M) and the initiation of an action to reach that goal (I). Supra-threshold activity in the precondition field (P) suppresses activation of I (connection 4). The precondition can be released by an inhibitory input from the perceptual system (*release of precondition*, Figure 2), or internally, by C-field activity (connections 7), which signals successful execution of another part of a behavioral sequence. A *plan* emerges in the interaction between the M and P fields, where supra threshold activity in the P-field propagates to M-



**Figure 3.** Left: Initiation of a minimal plan. The plan consists of recurrent activation of a single EB, illustrated as an intersection of the two-dimensional motivation (M), precondition (P), and intention (I) fields. An external (goal) stimulus ( $\gamma$ ) activates a particular site at the M-field. The blue lines represent activity in the corresponding fields shortly after the goal stimulus is introduced. The horizontal black lines represent detection thresholds of the fields (see Section 2), allowing activity to spread to other fields. Activity in the M-field spreads, via connections 1 and 3, to P and I. As a peak forms in P, the I-field is inhibited (connection 4), preventing execution of the behavior. Activity will continue to spread, via connection 2, activating a neighboring site of the M-field. The process continues recursively, eventually forming a *plan* between M and P, until intersecting with a release of precondition. Right: Execution of minimal plan. Suppression of precondition (connection 7) allows a peak to form in the intention field (I), initiating execution of the behavior. If execution is successful, a condition of satisfaction peak will form (indicated as a red-line input), releasing precondition for the next step of the plan (compare with Figure 2).  $\delta$  denotes the size of the shift for connections 1 and 3.

fields of EBs that can fulfill that precondition (connection 2). Behavioral sequences (connections 2 and 7) are formed both between different EBs, in this case EB1 fulfills the preconditions of EB2, or internally within an EB, constituting a sequence of one EB, executed several times with different parameterizations. Finally, condition of satisfaction (C-field activity) results from simultaneous input from both the I-field (connection 5) and perceptual input (connections 8) indicating a goal state. C-field activity acts both to terminate execution (connection 6) and release preconditions (connection 7).

A plan is represented as a pattern of self-stable supra-threshold attractors (see Schöner, 2008) distributed over the M and P fields (connections 1 and 2). The active regions in the P-fields inhibit respective regions in the I-fields (connection 4), which would otherwise be activated by input from the active M-fields (connection 3). When one of the P-fields receives inhibitory input (connection 7) strong enough to inhibit the relevant location in this field and thus release the precondition, a peak rises in one of the intention fields, and the plan unfolds into action.

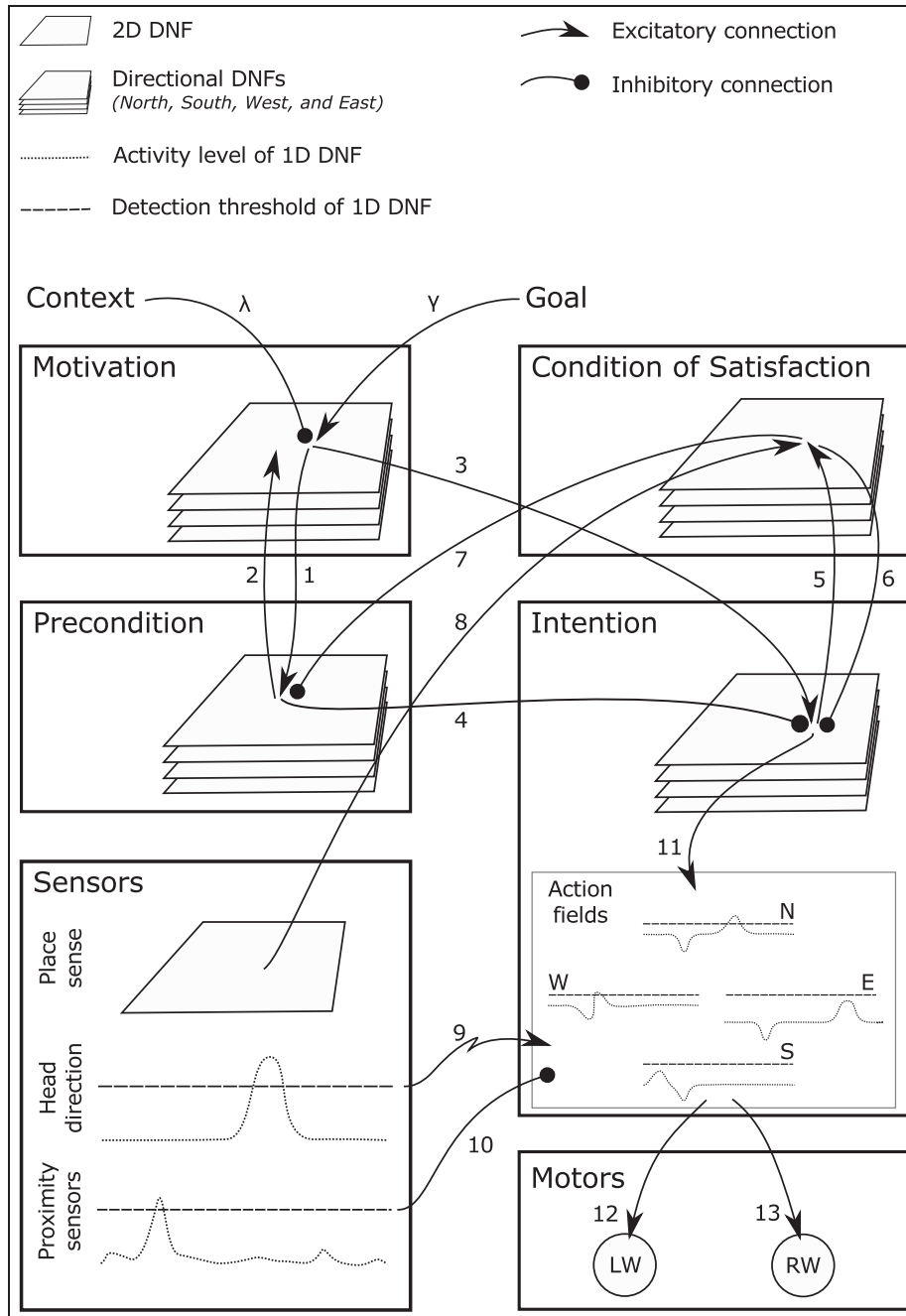
An example of a minimal planning sequence using recurrent activation of a single EB is illustrated in Figure 3. This minimal *plan* helps to illustrate the spreading of activation in time between EBs in the proposed DFT framework. Slices of the two-dimensional fields are depicted, illustrating the formation of a *plan* along a continuous dimension of a single EB. Each location on this dimension corresponds to an action of the EB with a different parametrization (e.g. movement

to a particular location in space). Active locations in the M-field excite the respective locations in the I and the P fields. If activity in the P-field is not inhibited, it suppresses the respective location in the I-field, preventing initiation of the action.

An active part of the P-field may also activate the M-field, that is, the architecture is forming a motivation to fulfill that particular precondition. A single site of P-field may activate several locations in the M-field, or even M-fields of other EBs, as illustrated in Figure 2 and later discussed in Section 3.2. Eventually, the spreading motivation may lead to activation of an M-field for which the precondition is suppressed (Figure 3, right), allowing a peak to form in the I-field, and the plan unfolds into action.

### 3.2 The SPA architecture implemented for the path planning scenario

The SPA architecture for the navigation example, considered in this work, is shown in Figure 4. The implementation of the SPA framework for this scenario requires some explanation: the individual EBs correspond to elementary movements in one of the directions in space (arbitrarily denoted as North, South, West, and East). Each EB is parameterized along two dimensions corresponding to physical space, resulting in two-dimensional DNFs implementing M, P, I, and C (compare with Figure 2). Locations in M, P, I, and C fields represent transitions between places in the environment. For instance, activity at a location A in the I-field of the *North* EB initiates a movement towards the



**Figure 4.** Overview of the complete architecture, including interactions with the robot. See text for details.

north, potentially involving the robot turning in this direction, from the location A. The robot will continue to move north as long as there is activity in the *North* I-field, but the precondition for moving north (which inhibits the respective intentions) will normally never be released unless the robot is standing at a location from which it can move north towards a motivated goal.

Figure 4 shows how the M, P, I, and C DNFs are connected with sensors and actuators of the robot. The I-fields propagate activity to four action fields (connection 11), which induce rotations of the robot's wheels

(connections 12 and 13) so that it orients and moves in the selected direction. Activity in the action fields is shifted depending on the correct heading direction of the robot (connection 9) and is inhibited by sensed obstacles on the robot's path (connection 10).

A peak in the *condition of satisfaction* C-fields forms when the robot reaches a position adjacent to the current location in the intended direction, i.e. when input from connections 5 (input from the I-field) and 8 (perceptual input from the place sense of the robot) intersect. Mathematical formulations of this network are given in Appendix A. Note that the SPA framework is

not limited to four EBs representing compass directions; this particular configuration was used to constitute an intuitive example comparable with classical planning approaches.

Some of our EBs are mutually exclusive (e.g. *South* and *North*). Competition between opposing EBs is implemented in a similar way as the motivational links (connection 2, Figures 2–4), but is represented by inhibitory rather than excitatory connections. For example, activation in the *East* M-field representing motivation to go east towards a specific position X results in activated preconditions for locations west of X (compare with Figure 3). Activity in the *East* P-field will spread, via excitatory connections, to corresponding locations in the M-fields of *North* and *South*, but will also suppress activity in the *West* M-field. This results in a cascade of competing activation between M- and P-fields, among the available set of EBs. One example of this cascading activity among all four EBs is presented in the result section (Figure 12).

The M-fields receive a *Goal* (connection  $\gamma$ ) input, essentially a place-holder for the neural activity correlated with a reinforcing input, e.g. an anticipated or actual rewarded state. In other work (Gaussier, Leprêtre, Quoi, Revel, & Joulian 2000, Hirel et al., 2013) motivations are considered in relation to drives (Tolman, 1948) and gate the valuation of the particular goal location. We take a similar approach by representing the goal as a stimulation to one or more M-fields. This stimulation constitutes the starting point for the formation of a plan. In the present work, a single Gaussian goal input is given to all M-fields, corresponding to a specific target location in the local surroundings that potentially can be reached from any direction.

Obstacles and other behavioral restrictions imposed by the environment are represented as inhibitory contextual input to the motivation fields (connection  $\lambda$ , Figure 4). The context should here be understood as a memory of the local surroundings. Both the DNF representations, and their mapping to the world through sensors, are continuous in space and time. A contextual or goal input may change at any time, sometimes resulting in dramatic changes in the activation patterns of the fields, and as a result, the robot's behavior. While plans form as competing activity in the M- and P-fields and are in this sense continuous, each EB implements a certain connection shift ( $\delta$ , see Figure 3) and has in this sense a discrete component. Plans may also involve several different EBs, e.g. execute *West* to release the precondition for going *North*. In a different scenario, this may involve moving towards a table before reaching for an object, where reaching and moving are formulated as different EBs. In this way, the architecture opens up for planning between different representational spaces (LaValle, 2006).

The agent may have many memories of different surroundings, but only one is active and imposed onto the M-fields at a single time. How these context representations may be formed and selected is discussed in Section 6, implementation of these processes is outside the scope of this work.

The SPA architecture for path planning integrates sensor information of three types: *place sense*; *head direction*; and *proximity sensors*, see Figure 4. The place sense provides sensory evidence for condition of satisfaction. In a general sense, it provides information that one EB has been successfully executed, the desired location has been reached, and the next step of the plan can be initiated. In the present implementation, the place sense field receives Gaussian input from a single noisy position sensor, see Section 4 for details.

Activity in the head direction and proximity sensor fields biases action selection (Figure 4). Each EB connects to one *action field*, a one-dimensional DNF defined over an angular dimension, representing the turning angle of the robot. The head direction field provides a single Gaussian input to each action field (connection 9).

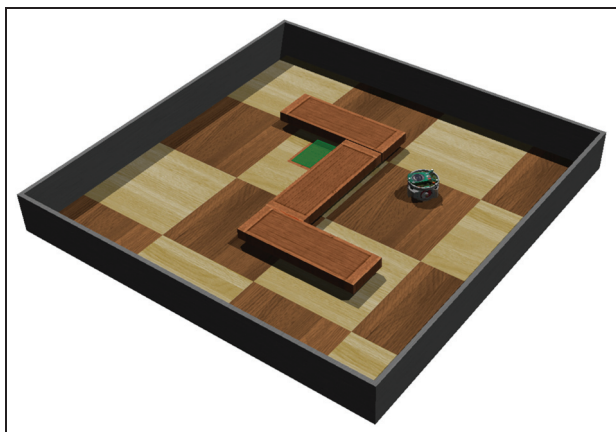
By default, the head direction input results in sub-threshold activity of the action field and hence no motor output. However, when the action field gets additional input from the I-field (connection 11), the combined input results in a supra-threshold peak in the action field, and motor output. The detailed mathematical formulation is given in Appendix A.

Each action field also gets inhibitory input from the proximity sensors (connection 10), which are activated if an obstacle is present near the robot. This input may shift or completely suppress a peak in the action field, and as a result, adjust or prevent a turn in a specific direction. This mechanism implements elementary dynamics of obstacle avoidance.

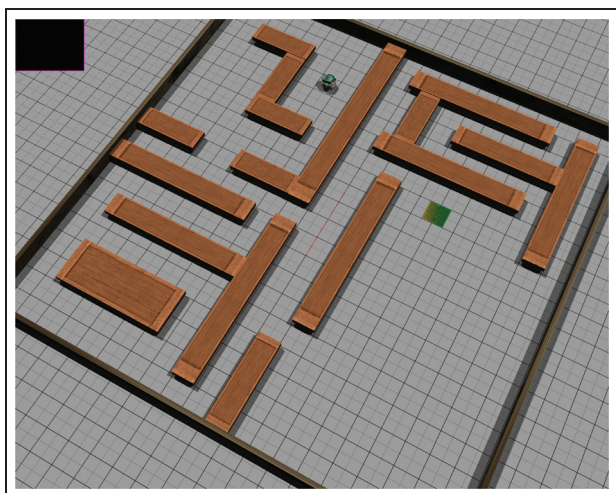
## 4 Robotic demonstration

Since the proposed architecture is to our knowledge the first DFT formulation that implements a mechanism for search, the main purpose of this demonstration was to evaluate to what degree the architecture can search for a sequence of actions towards a goal, in a way comparable to classical search algorithms. Secondary goals were to study the system's response to changes and noise during planning and plan execution. Since the neural fields constituting the search mechanism are bound directly to the sensory-motor systems of the robot (see Figure 4), it is effectively implementing both path planning and path tracking, and is therefore analyzed as a complete system.

The proposed architecture<sup>1</sup> was implemented using the Matlab framework COSIVINA (Schneegans, 2015) and evaluated in two simulated environments, using



**Figure 5.** Evaluation environment 1, the Z-maze. The e-puck robot is here standing at the starting point used for test case 3. The green area marks the goal.



**Figure 6.** Evaluation environment 2, the large maze used for test case 2.

the robot simulation software Webots (Cyberbotics, 2015). The first environment is a minimalistic Z-maze, depicted in Figure 5. The second environment, depicted in Figure 6, is a large maze introduced to test the architecture in a more complex setting. It should be noted that, despite their block-like look, these environments are continuous. To analyze behavior of the architecture, three test cases were defined;

**Case 1.** 100 pairs of start-goal locations in the Z-maze were selected randomly.

**Case 2.** 100 pairs of start-goal locations in the large maze were selected randomly.

**Case 3.** The behavior of the system was evaluated when obstacles are introduced at different times during planning and execution. Fixed start and goal locations were used (as displayed in Figure 5). Two types of

obstacles were used, a partially obstructing obstacle and a fully obstructing obstacle. The test case was executed 240 times for each obstacle type.

The purpose of the first two cases was to confirm that the architecture produces path planning and tracking behavior comparable to traditional methods, in both a small minimalistic setting (Case 1) and a larger, somewhat more demanding, environment (Case 2). Case 3 was designed to evaluate the system's response to changes in the environment. The effects of noise were studied in all test cases, using two noise levels.

Start and stop locations were selected with a minimal distance of 40 mm from walls and obstacles. The obstacle introduced in Case 3 appeared at a random time  $t_i$  and was removed at time  $t_i + t_p$ .  $t_i$  and  $t_p$  were drawn from a uniform distribution such that  $30 < t_i < 700$  and  $2 < t_p < 500$  simulation steps. These limits were selected to make the obstacle appear during planning or execution, before the robot reached the goal. When an obstacle was added, apart from introducing an object in the simulator, the map was updated with the new obstacle information, providing a new contextual inflow to the motivation fields.

Each test case was repeated with two different levels of noise, while keeping all other parameters constant. With the low noise level, normally distributed noise with an  $\text{std} = 5\%$  of the contextual stimuli strength ( $\lambda$  in Figure 4) was applied to the map. Normally distributed noise with  $\text{std}$  of 5 mm was also applied to the position sensor. With the high noise level,  $\text{std}$  of 100% of  $\lambda$  and 20 mm was applied to the map and position sensor, respectively. Noise  $\text{mean} = 0$  in all cases. Distance measures in the simulated environment are given in proportion to the physical e-puck robot (Mondada et al., 2009), with a diameter of 74 mm. Locations for the obstacles introduced in test case 3 are depicted in Figure 13.

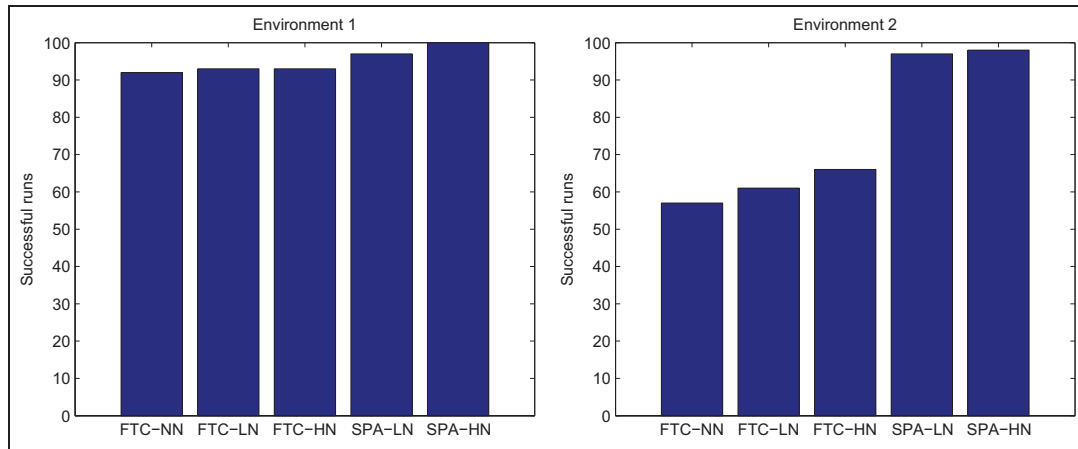
In addition to the tests using the proposed architecture, test cases 1 and 2 were executed with a reference implementation of *Follow the carrot* (Barton, 2001). For details on the implementation of *Follow the carrot*, please refer to Appendix B.

## 5 Results

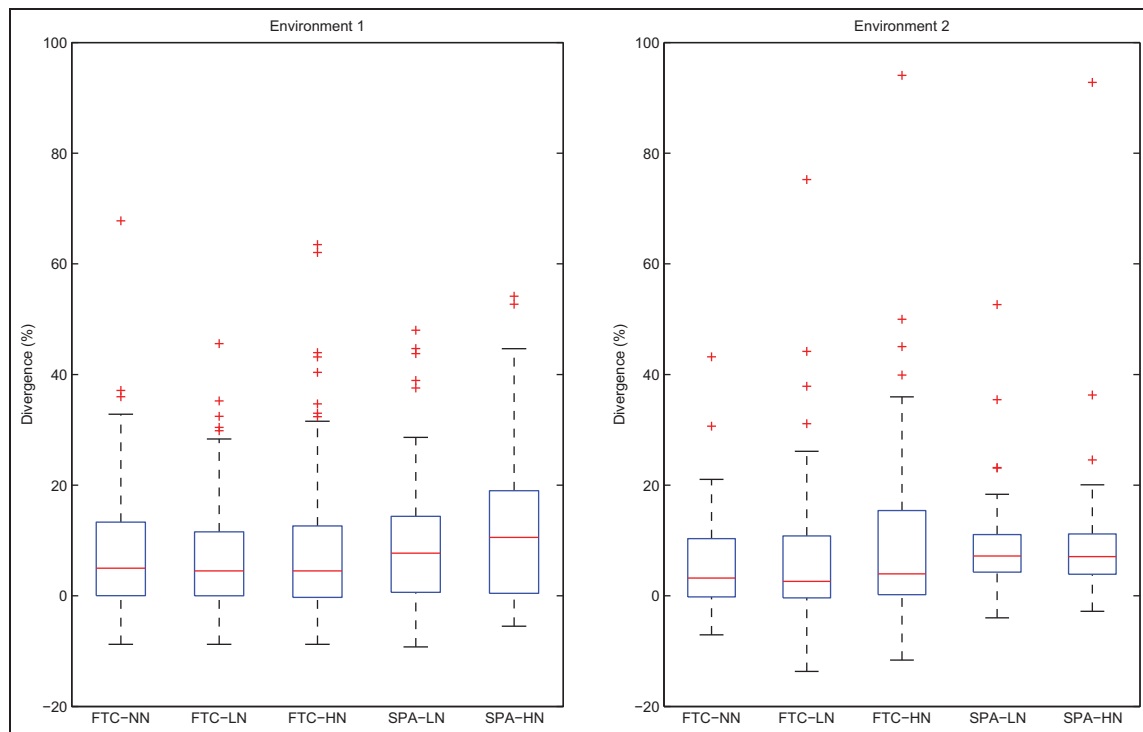
### 5.1 Results for test case 1 and 2

The proposed SPA architecture was compared to a standard path tracking approach *Follow the carrot*. Tests were executed in two environments, a minimalistic Z-maze (Case 1, Figure 5) and a large environment (Case 2, Figure 6). Test cases 1 and 2 comprised a total of five conditions each. The proposed SPA architecture with low and high noise, and *Follow the carrot* tested without noise, with low noise, and with high noise. Each condition was executed 100 times for at most 3000 simulation steps (corresponding to 192 s with a





**Figure 7.** Number of successful runs for environment 1 (left) and 2 (right). Results are given for the proposed SPA architecture tested with low noise (SPA-LN) and high noise (SPA-HN), compared to *Follow the carrot* without noise (FTC-NN), with low noise (FTC-LN), and high noise (FTC-HN).

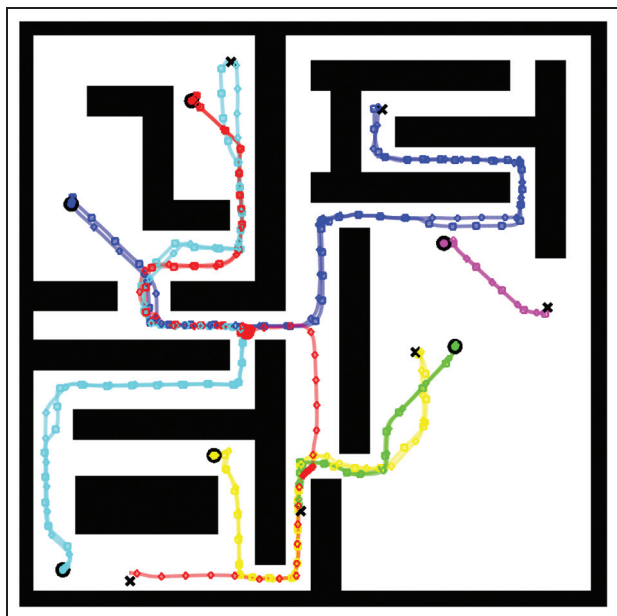


**Figure 8.** Divergence from optimal path for environment 1 (left) and 2 (right). Results are given for the proposed SPA architecture tested with low noise (SPA-LN) and high noise (SPA-HN), compared to *Follow the carrot* without noise (FTC-NN), with low noise (FTC-LN), and high noise (FTC-HN).

simulated time step of 64 ms). This limit was selected to be well above the time necessary to reach the goal, even for the longest paths.

The proportion of successful runs is presented in Figure 7. To get an overview of the stability and optimality of the proposed architecture, divergence from the optimal path was calculated as  $L_p/L_o$ .  $L_p$  represents the executed path length and  $L_o$  is the optimal path,

defined as the Euclidean distance given by A\* over an eight neighbor grid with a cell size of 1 cm and a minimum obstacle distance of 4 cm. A One-way ANOVA revealed no significant differences in path length between different conditions in environment 1 ( $F=0.97$ ) and environment 2 ( $F=0.98$ ). Executed paths produced by SPA from six representative runs of Case 2 are presented in Figure 9.



**Figure 9.** Path plot of six representative runs from Case 2, each start/stop pair plotted with a unique color. Lines are marked with diamonds and squares, representing low and high noise, respectively.

## 5.2 Results for test case 3

In order to analyze the detailed function of the architecture, activity in all fields was logged during a single run from Case 3, where a fully obstructing obstacle was introduced at  $t = 336$ . Figure 10 displays the temporal evolution of activity at four locations in the motivation field, which are defined below, for each of the four EBs (*North, East, South, and West*). The same four locations are displayed as a path plot in Figure 11. A snapshot of the activity in the motivation fields at the moment when planning is complete and the robot starts to move ( $t = 210$ ) is presented in Figure 12.

The robot starts at *location 1* ( $t = 0$ ), then continues to *location 2* ( $t = 500$ ), *location 3* ( $t = 850$ ) and, finally, *location 4* ( $t = 1250$ ). At  $t = 0$ , an excitatory input is introduced in all motivation fields, at the location of the goal. Through the interplay between the M- and the P-fields, motivation is back-chained for each action, i.e., activity spreads north in the south motivation field, and vice versa, but also from the south motivation field to the east and west fields, leading to a competitive spread of activation in all directions. The goal input first propagates to the north and east motivation fields at location 4, see in Figures 10 and 11. The south and west M-fields receive inhibitory input from north and east, respectively, leading to the dip in activity at this location. Around  $t = 150$ , the wave of propagating activity reaches location 2, activating the west motivation field, and at  $t = 210$ , the complete “plan” is present as a pattern of supra-threshold activation in the four motivation fields (Figure 12).

After  $t = 210$ , activity increases in the intention fields (Figure 10, bottom pane). As a consequence of strong activity in the north field, and competing activity in the west and east fields, the robot leaves location 1, turns north, and as the west field wins the competition, continues towards north-west (Figure 11).

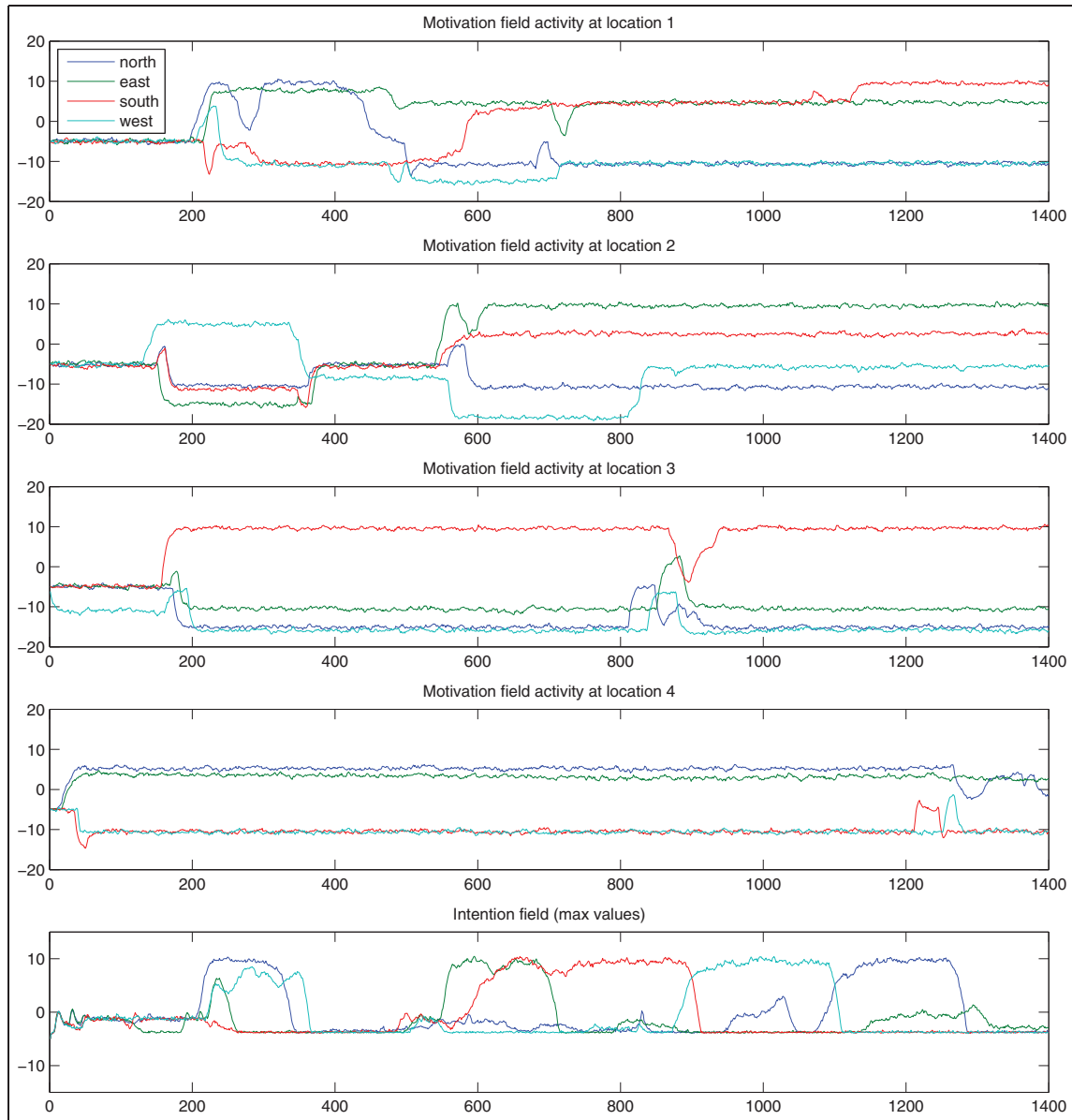
A fully obstructing obstacle is introduced during early execution ( $t = 336$ ). The following interval  $350 < t < 550$  can be seen as a replanning phase where the activity in the west motivation field dies away, and is replaced by activation in the south and east motivation fields, at location 2. The resulting sequence of actions is visible in the maximum activity of the four intention fields (Figure 10, bottom pane). Around  $t = 550$ , the east attractor stabilizes at location 2, leading high activity in the east intention field, followed by south, west, and finally, north.

This replanning behavior is dependent on the parametrization of the neural fields, resulting in a self-stable, but not self-sustained, field activation. That is, active regions of the field that no longer get the propagated support from a goal will decay. This is the case when the obstacle is introduced, cutting propagation of motivation to go west, and as a result, activity at location 2 of the west M-field (Figures 10 and 11) will decay, allowing competing motivation from the south to take over, resulting in the initiation of an alternate route.

An overview of the systems response to introduced obstacles is presented in Figure 13. Individual runs are colored according to the selected path, where green represents a direct route north of Z (shortest), blue represents a direct route south of Z (when the north path is blocked), and red represents an initial selection of the north path with a change to the south.

The time course for some representative runs from Case 3 is presented in Figure 14. If no obstacle is present, the robot would always opt for the shortest (green) path, not displayed here. When the obstacle appears during early planning, the robot opts instead for the slightly longer south (blue) route, independently of obstacle type (Figure 14, top pane). Obstacles appearing later do not have the same effect. A partly obstructing obstacle appearing during late planning or execution will not result in a change of the selected path. The robot persists on the north (green) path, which should be understood as a case of path adjustment rather than complete replanning. However, when a fully obstructing obstacle appears during late planning or early execution, the robot switches from the north to the south route (red). A period of velocities close to zero can in these cases be understood as the time of replanning.

A detailed analysis of path selection and replanning behavior is presented in Figures 15 and 16. As visible in Figure 15, the appearance of a partly obstructing obstacle during the first 200 time steps results in the selection of the south (blue) route, even if the duration of that



**Figure 10.** Field plots from the run displayed in Figure 11. The top four plots present activity from motivation fields at four locations the robot visited during execution. The bottom plot presents maximum activity in intention fields.

obstacle is short. However, the fully obstructing obstacle that persists for a short time (Figure 16) appears to have less effect on the decision-making (i.e. the robot selects the north route more often).

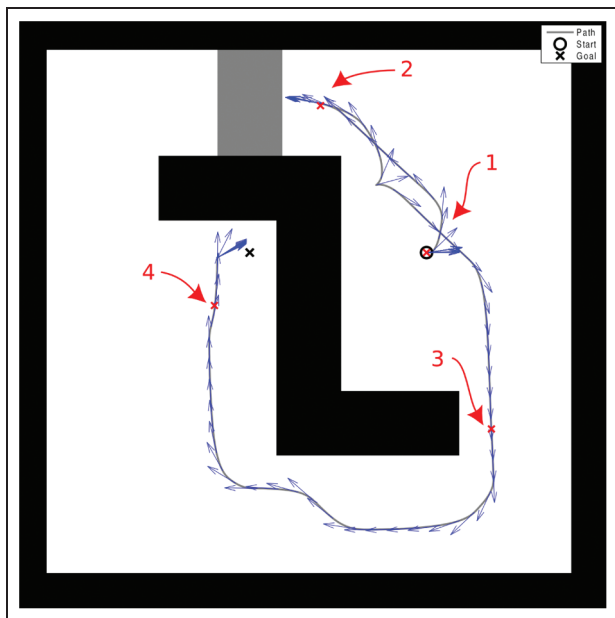
When the obstacle occurs after  $t = 200$  time steps, which roughly corresponds to the time when the robot starts moving (see Figure 14 for details), the robot makes an initial selection of the shortest, north, path in almost all cases. If a fully obstructing obstacle appears within  $200 < t < 500$ , the robot switches from the north to the south route (red). The robot is, however, more persistent in continuing on the north (green) route compared to runs where the obstacle appeared before  $t = 200$ , that is, the robot waits longer for the obstacle to disappear if the obstacle appears after

execution started. For runs where the obstacle appears after  $t = 500$ , the robot always selects the north route. In these cases, the robot has already passed the location of the obstacle at the time it appears, and it therefore does not interfere with the executed plan.

## 6 Discussion

### 6.1 Related work

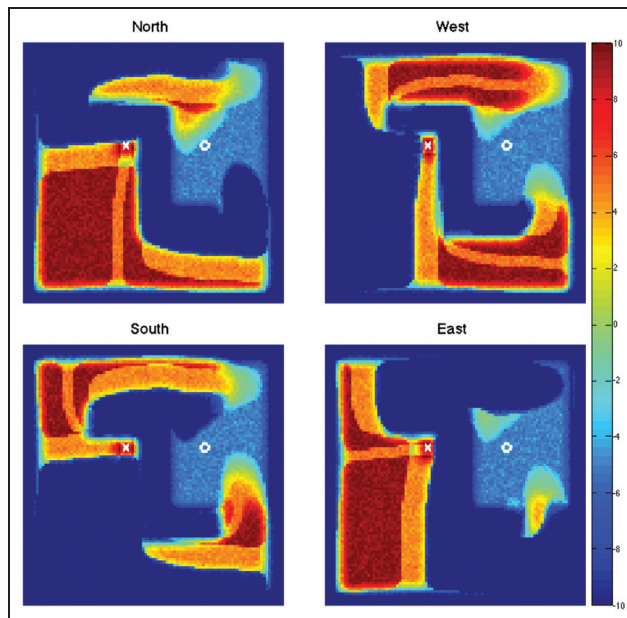
Commonly, “global” planner algorithms revolve around classical graph- and tree-based search algorithms (LaValle, 2006; Russell & Norvig, 1995). Typically, graph-based approaches entail a *search* and an *execution* component. The particular application



**Figure 11.** Path plot over a single run from test case 3. Red x marks positions that the robot passed during execution. The black x marks the goal. The grey area represents the location of a fully obstructing obstacle introduced at  $t = 336$ .

requirements constrain the choice of the approach; for example, where robust performance (avoiding collisions and maintaining accurate path integration) is concerned, constructing a Voronoi diagram that reduces likelihood of obstacle interference may be desirable. Alternatively, if the emphasis for performance is on finding optimal paths to targets, dividing the state space appropriately for computing the shortest possible path may be desirable, e.g. Distance Transform (DT) methods. In the latter case, algorithms exist that are capable of finding viable and optimal paths at low computational cost. They are used for “offline” planning, e.g. through a standard AI method for search such as  $A^*$ , that is then followed by an execution phase, or updated “online” according to perceived changes in the environment, as is the case for  $D^*$  and Anytime  $D^*$ , see Siegwart, Nourbakhsh, and Scaramuzza (2011) and Russell and Norvig (1995) for a summary of approaches. Following the search/planning phase, execution typically follows *path extraction* according to the desired optimality criteria. Such a graph-based approach is here used as a baseline reference, see Section 5.1 and Appendix B for details.

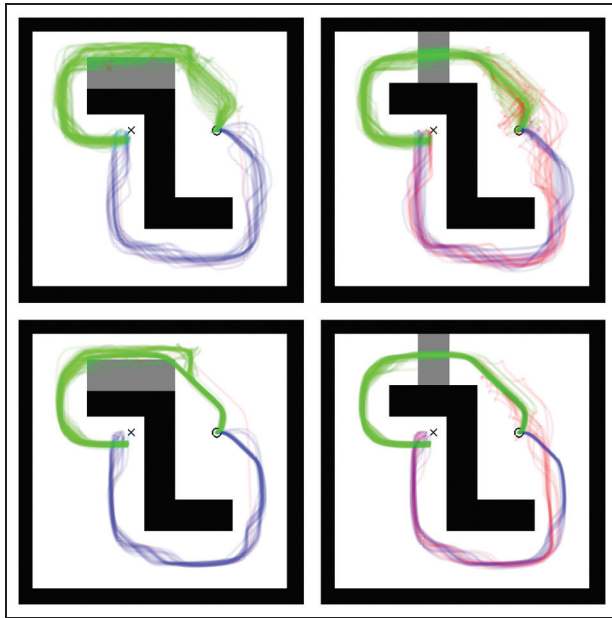
An alternative approach to graph construction/search for dealing with reactive-global navigation problems is that of potential field planning (Khatib, 1986). In this case, gradient valuations that cover the entirety of the state space are composed, allowing the robot to react to unexpected obstacles or changes in the environment while still being able to navigate towards the goal by following the gradient in relation to the new state it



**Figure 12.** Activity plot over motivation fields at  $t = 210$ . The pattern of supra-threshold activity (red) can be seen as the plan of moving north, west, south, and, finally, east from the starting point (white circle) to the goal (white cross).

finds itself in. Gaussier and colleagues (Gaussier et al., 2000; Gaussier and Zrehen, 1995; Hirel et al., 2013) have produced much work in the area of sequential behavior planning in relation to navigation including adopting a potential field guided perspective. The general approach followed by this group is that of planning the back-chains potential navigable routes (sequences of spatial behavioral transitions) from a goal state to the current state. Using this approach, the robot may explore the environment and identify landmarks, corresponding to place cells (O’Keefe & Nadel, 1978), with an activation level reflecting the robot’s distance from the identified place. By propagating activity from a goal, through the network of identified landmarks, a potential field is generated, allowing the robot to navigate towards the goal by following the gradient. This approach is shown to afford performance-wise robust responding to obstacles and temporary occlusions as robot navigation is guided by the potential field (Gaussier et al., 2000).

In contrast to potential fields that implement a static attractor basin centered at the goal, the approach presented here relies on dynamic attractors forming as activity propagates from the goal to the agent’s current state. Furthermore, while potential fields use the gradient for action selection, the framework presented here relies on several discrete elementary behaviors, competing for activation. Potential fields may produce local minima in complex environments. Although not free from local minima, the competing actions approach used here appears less prone to these problems. We aim

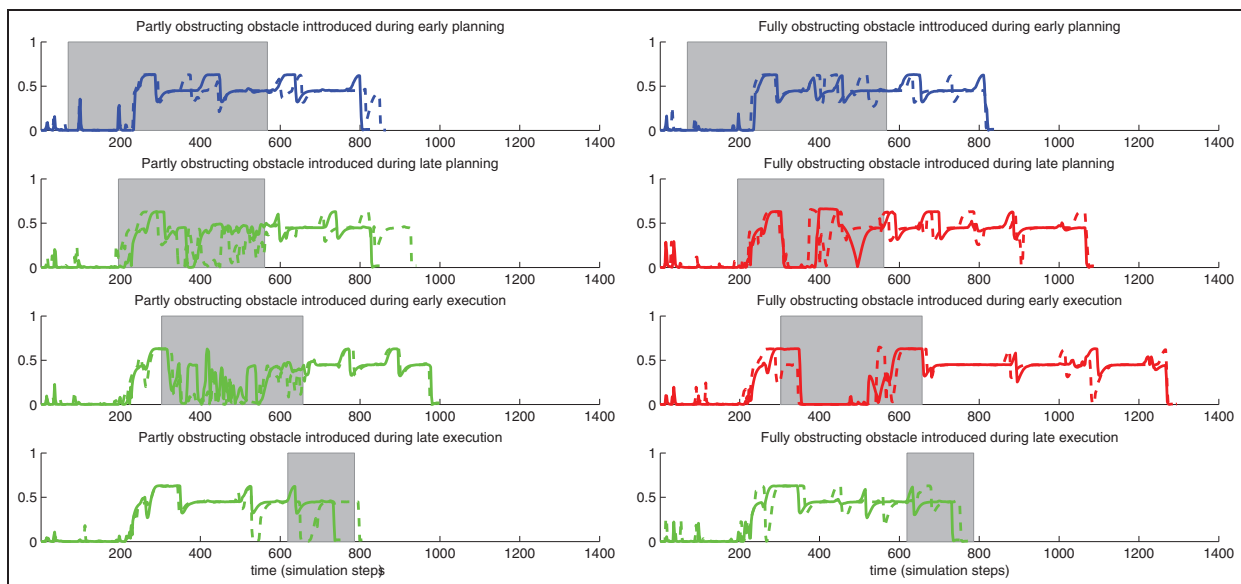


**Figure 13.** Path plots over runs from Case 3, where an obstacle was introduced at a random time. Left and right plots represent partly and fully obstructing obstacles, respectively. Upper plots show runs with high noise, and lower display runs with low noise. colors represent path selections: green represents a direct route north of Z (shortest), blue represents a direct route south of Z (when the north path is blocked), and red represents an initial selection of the north path with a change to the south.

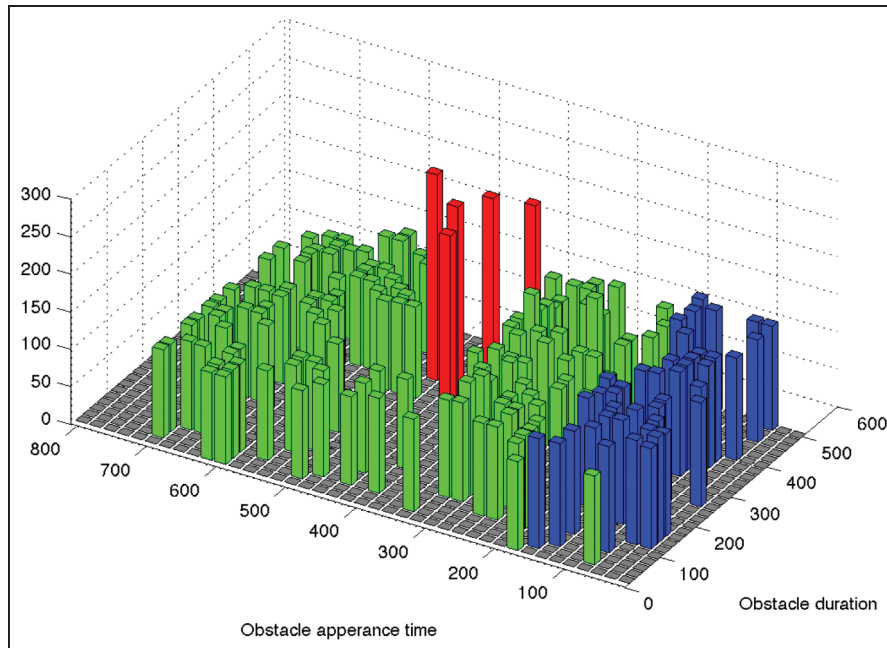
to conduct a direct comparison with the potential fields approach in future work.

Further alternatives to classical path planning approaches include reaction-diffusion methods (Adamatzky, De Lacy Costello, Melhuish, & Ratcliffe, 2003; Trevai et al., 2002; Vazquez-Otero, Faigl, & Muñuzuri, 2012) and the so-called activation-diffusion (Martinet, Passot, Fouque, Meyer, & Arleo, 2008). In the case of reaction-diffusion, a typical approach is for the path planner algorithm to generate diffusive waves of (e.g. neurochemical) activation in a map, according to the interaction of two or more variables, as guided by mapped obstacles. Some such approaches use highly biologically focused algorithms (Adamatzky et al., (2003), others seek rather to profit from certain principles of reaction-diffusion dynamics. In the latter case, Vazquez-Otero et al. (2012) used an algorithm that firstly generated diffusive activity from a start state over a map of a maze (propagation phase) and secondly, on arriving at the goal state, activated a contraction phase (dissipation of activity), and finally, prior to action execution, a path extraction phase using a standard search algorithm. They compared their approach to classical approaches - Distance Transform (DT) and Voronoi diagrams - to assess the optimality and safety-value of the paths selected. A benefit of the approach was the creation of *smooth* paths that permit more efficient online behavior.

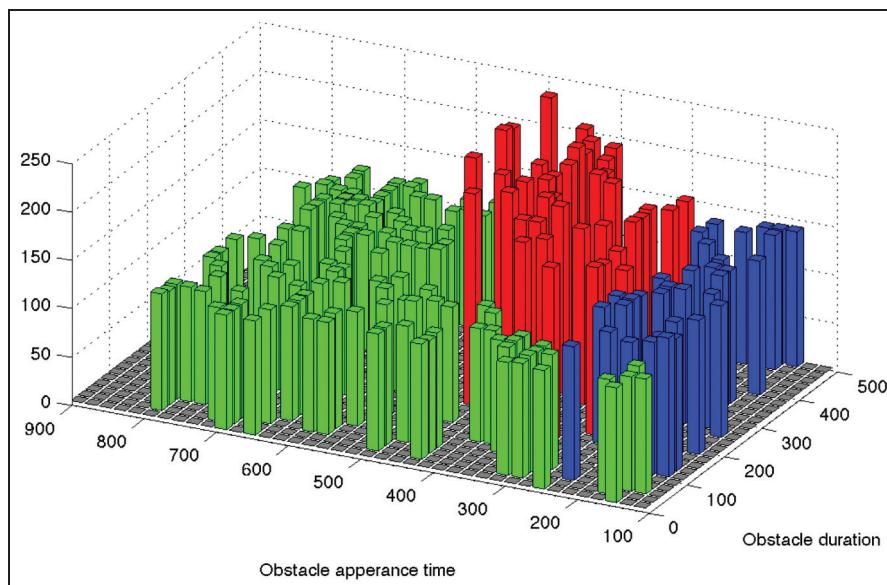
A general problem for reaction-diffusion mechanisms concerns finding appropriate mechanisms for permitting online reaction to changes in the environment (Adamatzky et al., 2003) and, relatedly, the dependence on explicit path extraction methods to guide online behavior. Martinet et al. (2008) put forward an activation-diffusion method by which simulated



**Figure 14.** Velocity plots from four representative conditions in Case 3, where an obstacle is introduced during early planning, late planning, early execution, and late execution, in order from top to bottom. The grey area represents the time that the obstacle was present. Solid and dashed lines represent the velocity over time, with low and high noise, respectively. Lines are colored according to selected path, same as Figure 13.



**Figure 15.** Path length plot over runs in Z-maze with low noise, where a partly obstructing obstacle was introduced at a random time (x-axis), for a random duration (y-axis). Bars are colored according to the selected path, as illustrated in Figure 13.



**Figure 16.** Path length plot over runs in the Z-maze with low noise, where a fully obstructing obstacle was introduced at a random time (x-axis), for a random duration (y-axis). Bars are colored according to the selected path, as illustrated in Figure 13.

cortical columns propagate back-chained activity from a “motivation”-gated column to one associated with a start state. The activation of the latter column is then forward propagated as a path representation. This activation-diffusion approach has several similarities with ours in that activation is back-chained from goal to start state and that action choices compete for activation. However, the approach presented here does not

require forward propagation or the extraction of an explicit path signal.

## 6.2 Relation to existing planning and search methods

The properties of the neural-dynamic Elementary Behaviors, which are the building blocks of the SPA

framework introduced here, are put in relation to existing planning and search algorithms:

1. *Wavefront search.* The neural dynamic nature of planning may be compared to a wavefront (i.e. multi-directional) approach. The planner is “systematic” (LaValle, 2006, p. 32). The neural dynamic nature of our planner algorithm ensures, given enough time, that all reachable states will be searched and not re-searched (it is not redundant).
2. *Shortest path search.* Initial search is Dijkstra-like. Whilst neural dynamic search is carried out, multiple candidate paths are evaluated in parallel. A short path stochastically receives greater activation faster, compared to longer paths, which leads to the inhibition of competing paths.
3. *Uni-directional.* The search is uni-directional in the sense that an active region of one motivation field is partly self-sustaining and will inhibit competing transitions, i.e. re-evaluations, as long as that location is active. The self-sustainability is however dependent on neighbouring regions being active. The introduction of an obstacle, cutting of an active path, will consequently lead to a spreading deactivation of remaining parts of that route. Deactivation also releases the inhibition of competing transitions, allowing for re-evaluations of that part of the search space. This spreading activation/deactivation mechanism results in continuous adaptation to changes in the environment.
4. *Partial updates.* Search is also online and can be compared to  $D^*$ . When changes in the environment occur (see Section 5.2), neural dynamic planning is updated locally (in relation to the changed aspect of the environment). This means that the robot is not required to completely replan when a path is required to be modified rather than “catastrophically” altered. This is very much a feature of the SPA nature of the neural-dynamic architecture. This property shows up in the relatively short replanning times (visible as periods of low velocity in Figure 14), compared to the initial planning phase (i.e. the initial propagation of activity from the goal to the starting location of the robot).
5. *Safety.* It was found that the architecture generated navigable paths equidistant between obstacles/walls. The path search, therefore, can be seen as an emergent Voronoi diagram. This can be put in relation to the reaction-diffusion algorithm of Vazquez-Otero et al. (2012), which produced navigable trajectories that were not purely dedicated to shortest-path exploration, but also included aspects of smoothness and a safe distance to obstacles. The approach presented here shows similar properties in that a path with longer distance to walls is selected, when that space is available. The south (blue) route depicted in Figure 13 shows a relatively long distance to all walls, compared to the north routes (green) of Case

3A where the partial obstacle only leaves a narrow corridor. Furthermore, the narrow passage leads to a reduction in velocity (Figure 14).

6. *Gradient search.* Search implicitly follows a gradient (in relation to the Euclidean distance from the goal state to feasible intermediate states to the current state). However, there is no explicit gradient valuation over state space, from which potential fields can be derived. One advantage of this approach is that long paths do not suffer from low gradient valuations that are susceptible to noisy evaluations or otherwise fading memory of paths that may lead to execution problems.

### 6.3 Biological mechanisms for planning and navigation

Ever since the work by O’Keefe and Nadel (1978), pyramidal neurons in hippocampus with firing patterns that correlate with the physical location of the rat, so-called *place cells*, have been a hot topic in cognitive neuroscience. A growing body of literature tells an increasingly complex story that involves not only place cells, but also cells in postsubiculum that are sensitive to direction (*head direction cells*) (Taube, Muller, & Ranck, 1990a, 1990b) and cells in medial entorhinal cortex with place sensitive firing patterns arranged in hexagonal grids (*grid cells*) (Fyhn, Molden, Witter, Moser, & Moser, 2004; Hafting, Fyhn, Molden, Moser, & Moser, 2005; Sargolini et al., 2006). Hok, Save, Lenck-Santini, and Poucet (2005) have also provided evidence for cells in medial prefrontal cortex (mPFC) that reflects the motivational salience of places (*goal cells*).

Experimental research on rodents has inspired a large body of computational investigations, e.g. Burgess, Recce, and O’Keefe (1994) and Redish and Touretzky (1997). These are also models, inspired by research on place cells, directed towards robotic applications targeting the problem of simultaneous localization and mapping (SLAM). A biologically inspired model, RatSLAM (Milford & Wyeth, 2008), has shown impressive results in mapping 66 km of urban roads using only a single camera. An algorithm like RatSLAM could potentially be linked with the architecture presented here via the *Place sense* and *Head direction* fields (Figure 4).

While place cells as a mechanism for self localization and mapping appear to be relatively well understood, their role in path planning and goal pursuit is still puzzling (Jeffery, Gilbert, Burton, & Strudwick, 2003; Poucet, Lenck-Santini, & Hok, 2004). Computational models of navigation based on place cells typically formulate the problem using reinforcement learning, using both classical artificial neural networks (Kulvicius, Tamosiunaite, Ainge, Dudchenko, & Wörgötter, 2008)

and spiking networks (Strösslin, Sheynikhovich, Chavarriaga, & Gerstner, 2005). Available paths have been modeled as transitions from one place (represented by a place cell's firing field) to another, and coded as *transition cells*, corresponding to CA3 pyramidal cells (Banquet, Gaussier, Quoy, Revel, & Burnod, 2002; Gaussier, Revel, Banquet, & Babeau, 2002). The present work takes inspiration from this view of place cells, reflected in our definition of elementary behaviors as leading to a transition in a particular direction (see Section 3 for details).

The work presented here can also be put in relation to the parallel view of perception and action proposed by Cisek and Kalaska (2010). Cisek and Kalaska summarize a large body of neurophysiological evidence speaking for a view where action selection and parameterization are processed in parallel to perception. Neural activity related to different decisions (action responses) builds up as a function of motivational value and perceptual evidence, speaking for or against different responses. While the architecture presented here is not aimed at closely resembling the biological mechanism, it constitutes one example of a computational model that captures many of these properties. Not only selection of short term action responses but also planning is in this view seen as preparation of competing actions, continuously influenced by perceptual evidence and motivational value.

#### 6.4 Effects of noise

In the gradient-based approach to planning sequential behavior used by Gaussier et al. (2000) it has been suggested that long paths may be difficult to plan –“if we want our animat to learn paths that need more than several tens of subgoals then the slope of the gradient [from goal representation to representation of current state] will be very low ... and will be very difficult to use ... There is a need to be able to structure the plans ...” (p. 86). In general, it has been suggested (Koren & Borenstein, 1991) that potential field approaches to robot navigation tasks suffer many limitations. For example, robots may a) get stuck in local minima, b) get stuck between non-passable objects/obstacles, c) oscillate/dither in the presence of obstacles or narrow passages.

Some of these limitations appear to be, at least partly, present in the architecture presented here. The robot does, on a few occasions, get stuck in local minima (see Section 5.1 for details). One such example is displayed in Figure 9, the red colored run with low noise. In this case, there are two competing attractors, one from the south transition network and one from the east. Since the distance to the goal happens to be almost exactly the same at the point of the junction, the two attractors are equally strong and, in combination with the inhibitory inflow from the obstacle avoidance, a local minimum appears.

One common way to handle local minima problems in potential fields is by introducing noise. As has been demonstrated in Section 5, the architecture presented here appears to be very robust to noise. Even though the high noise level did not lead to 100% successful runs in all cases, it did reduce the number of unsuccessful runs. The fact that no significant effect of noise was found in Cases 1 and 2 indicates that noise can be introduced with a very small cost in terms of navigation performance. These results should however be taken as preliminary, a deeper analysis of the system's behavior in response to noise is necessary to provide a full understanding of this potentially beneficial property.

#### 6.5 Learning and development

While learning is not studied experimentally here, it is still an important aspect of the proposed architecture. The architecture has many parameters and it is therefore desirable to show how these could emerge from learning. In the present context, learning can be considered on at least three levels of abstraction: (1) learning individual EBs; (2) learning relations between EBs; and (3) learning the constraints imposed by a particular environment.

The first problem may be addressed by combining reinforcement learning with the DNF framework. Based on the same principles as the ones used in the architecture presented here, Kazerounian, Luciw, Richter, and Sandamirskaya (2012) demonstrated one such example. Outside the DFT framework, we have also studied learning of EBs in a navigational context. Billing, Hellström, and Janlert (2015) applied the TD-algorithm *Predictive Sequence Learning (PSL)* to sensor and motor data from a Robosoft Kompai robot, allowing the robot to learn and generalize goal directed actions demonstrated via teleoperation.

We have also studied the second form of learning as the learning of connection weights between EBs (Luciw, Kaze-rounian, Lakhmann, Richter & Sandamirskaya, 2013; Sandamirskaya & Schöner, 2010). This should be understood as learning how behavioral consequences relate to each other and are typically environment independent. In the present work, we provide suitable connection weights between EBs, as described in Section 3.

Learning behavioral constraints for a particular environment can be seen as a SLAM problem. As discussed in Section 6.3, Milford and Wyeth (2008) propose an approach with several similarities to the present work. The technique, called RatSLAM, implements *Continuous Attractor Networks (CAN)* with similar dynamics to the DNFs used in the present architecture. In future work, we aim to combine these pieces of work in order to also learn the contextual



input, i.e. the environment map, during simultaneous planning action.

## 7 Conclusion

We have presented an approach to planning that exploits *Dynamic Field Theory (DFT)* based attractor networks to produce robust sequencing of elementary behaviors with an online updating of the planned path. The proposed approach has been implemented in an exemplary path planning and navigation scenario. The developed architecture consists of a tuple of *Dynamic Neural Fields (DNFs)*, in which activity spreads to propagate the motivational input (goal information), while prior knowledge of the environment is represented with an inhibitory flow to the motivation DNFs, and between the precondition and intention DNFs. The architecture receives direct inputs from the sensors and controls actuators of the robot, producing a system that performs neural-dynamic search in order to discover available paths towards the goal, whilst simultaneously executing selected behaviors. We refer to this property as *Simultaneous Planning and Action (SPA)*, effectively a form of online search as actions and changes in the environment perturb existing path plans.

In summary, the evaluation of our architecture in simulated robotic experiments demonstrated the following advantageous properties:

- *Scalability.* The proposed architecture scales well with the size or complexity of the environment. The activation spreads with a constant speed among DNFs and eventually reaches the current location of the agent, creating a landscape of attractors that leads towards the goal.
- *Noise Robustness.* The proposed SPA architecture appears robust to noise. Increased noise does not have any significant effect on navigation performance and does not appear to affect planning.
- *Performance robustness.* Thanks to the stability of attractor dynamics, transient changes and blockages of the environment do not result in replanning. The selected path will however be abandoned if an obstacle is blocking the way for a prolonged amount of time. Depending on the duration and extent of the blockage, an alternative route will be selected.

Of critical importance to the performance of the SPA architecture is its robust online path planning behavior. Noise and performance robustness are continuous in that minor perturbations (including partial and/or transient obstructions) do not deter the agent from following its goal path. Major perturbations, however, lead to path plan updates and following of an alternative route to the goal.

We demonstrate that the proposed architecture can produce path planning behavior comparable to traditional methods based on search (see specifically Figures 7 and 8). The number of successful runs in the larger environment is notably lower for the *Follow the carrot*, compared to the proposed SPA architecture. This is a result of the *Follow the carrot* algorithm cutting corners and therefore getting stuck at tight turns. While this is a common problem in classical approaches, the competition between actions that occur in SPA does not appear to cut corners. In addition, the high success rate for the proposed architecture partially may be attributed to tight integration of input from the proximity sensors, which biases action selection, delaying a turn-action if a wall is blocking the path. It should, however, be noted that there are other classical path tracking methods less prone to cut corners, and the problem can also be reduced using a reactive obstacle avoidance layer. Hence, these results should not be understood as a limitation of traditional search methods in general.

The approach to simultaneous planning and action presented here shows robustness on multiple levels, from emergent decision-making and high noise tolerance, to stable interactions with the world, without the need to separate the system into a deliberative and a reactive layer. While this approach still needs more analysis before it can be considered as a mature alternative to traditional planning, the present work should be seen as a proof of concept for dynamical approaches, and specifically DFT, applied as a method for planning and search.

## Acknowledgements

We would like to thank Sebastian Schneegans, Institute for Neural Computation, Ruhr-Universität Bochum, for providing the software framework COSIVINA (Schneegans, 2015) and invaluable help with technical support.

## Funding

This work was funded through the 7th framework of the EU, grant number #270247 (Neural Dynamics).

## Note

- 1 All source code required to run the presented experiments is available for download at <https://bitbucket.org/interactionlab/spa>. Recorded data is available as a specific software branch <https://bitbucket.org/interactionlab/spa/branch/data>.

## References

- Adamatzky, A., De Lacy Costello, B., Melhuish, C., & Rattcliffe, N. (2003). Experimental reaction-diffusion chemical processors for robot path planning. *Journal of Intelligent and Robotic Systems*, 37, 233–249.

- Amari, S. (1977). Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27(2), 77–87.
- Arbib, M.A. (1985). Schemas for the temporal control of behavior. *Human Neurobiology*, 4(2), 63–72.
- Arkin, R.C. (1998). *Behaviour-Based Robotics*. Cambridge, MA: MIT Press.
- Banquet, J.P., Gaussier, P., Quoy, M., Revel, A., & Burnod, Y. (2002). Cortico-hippocampal maps and navigation strategies in robots and rodents. In *Icsab proceedings of the seventh international conference on simulation of adaptive behavior on from animals to animats* (pp. 141–150). Cambridge, MA: MIT Press.
- Barton, M. J. (2001). Controller development and implementation for path planning and following in an autonomous urban vehicle. Phd thesis, The University of Sydney, Australia.
- Bastian, A., Schöner, G., & Riehle, A. (2003). Preshaping and continuous evolution of motor cortical representations during movement preparation. *European Journal of Neuroscience*, 18(7), 2047–2058.
- Billing, E. A., & Hellström, T. (2010). A formalism for learning from demonstration. *Paladyn: Journal of Behavioral Robotics*, 1(1): 1–13.
- Billing, E.A., Hellström, T., & Janlert, L.E. (2015). Simultaneous recognition and reproduction of demonstrated behavior. *Biologically Inspired Cognitive Architectures*, 12, 43–53.
- Brooks, R.A. (1991). New approaches to robotics. *Science*, 253, 1227–1232.
- Burgess, N., Recce, M., & O'Keefe, J. (1994). A model of hippocampal function. *Neural Networks*, 7, 1065–1081.
- Cisek, P., & Kalaska, J.F. (2010). Neural mechanisms for interacting with a world full of action choices. *Annual Review of Neuroscience*, 33(March), 269–298.
- Cyberbotics. (2015). *Webots*. <http://www.cyberbotics.com>.
- Ermentrout, B. (1998). Neural networks as spatio-temporal pattern-forming systems. *Reports on Progress in Physics*, 61(4), 353–430.
- Fyhn, M., Molden, S., Witter, M.P., Moser, E.I., & Moser, M.-B. (2004). Spatial representation in the entorhinal cortex. *Science (New York, N.Y.)*, 305(5688), 1258–1264.
- Gaussier, P., Leprêtre, S., Quoi, M., Revel, A., & Jouliau, C. (2000). Experiments and models about cognitive map learning and motivated navigation. In J. Demiris, & A. Birk (Eds.), *Interdisciplinary approaches to robot learning* (pp. 53–94).
- Gaussier, P., Revel, A., Banquet, J.P., & Babeau, V. (2002). From view cells and place cells to cognitive map learning: processing stages of the hippocampal system. *Biological Cybernetics*, 86(1), 15–28.
- Gaussier, P., & Zrehen, S. (1995). Perac: A neural architecture to control artificial animals. *Robotics and Autonomous Systems*, 16, 291–320.
- Hafting, T., Fyhn, M., Molden, S., Moser, M.-B., & Moser, E.I. (2005). Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052), 801–806.
- Hirel, J., Gaussier, P., Quoy, M., Banquet, J.P., Save, E., & Poucet, B. (2013). The hippocampo-cortical loop: spatio-temporal learning and goal-oriented planning in navigation. *Neural Networks*, 43, 8–21.
- Hok, V., Save, E., Lenck-Santini, P.P., & Poucet, B. (2005). Coding for spatial goals in the prelimbic/infralimbic area of the rat frontal cortex. *Proceedings of the National Academy of Sciences of the United States of America*, 102(12), 4602–4607.
- Jeffery, K.J., Gilbert, A., Burton, S., & Strudwick, A. (2003). Preserved performance in a hippocampal-dependent spatial task despite complete place cell remapping. *Hippocampus*, 13(2), 175–189.
- Johnson, J.S., Spencer, J.P., & Schöner, G. (2008). Moving to higher ground: The dynamic field theory and the dynamics of visual cognition. *New Ideas in Psychology*, 26, 227–251.
- Johnson, J.S., Spencer, J.P., & Schöner, G. (2009). A layered neural architecture for the consolidation, maintenance, and updating of representations in visual working memory. *Brain Research*, 1299, 17–32.
- Kazerounian, S., Luciw, M., Richter, M., & Sandamirskaya, Y. (2012). Autonomous reinforcement of behavioral sequences in neural dynamics. In *Proceedings of the joint IEEE international conference on development and learning & epigenetic robotics (ICDL-EPIROB)*.
- Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5, 90–98.
- Koren, Y., & Borenstein, J. (1991). Potential field methods and their inherent limitations for mobile robot navigation. In *IEEE international conference on robotics and automation* (pp. 1398–1404). Sacramento, CA: IEEE.
- Kulvicius, T., Tamosiunaite, M., Ainge, J., Dudchenko, P., & Wörgötter, F. (2008). Odor supported place cell model and goal navigation in rodents. *Journal of Computational Neuroscience*, 25(3), 481–500.
- LaValle, S.M. (2006). *Planning Algorithms*. Cambridge: Cambridge University Press. Retrieved from <http://planning.cs.uiuc.edu/>
- Luciw, M., Kazerounian, S., Lakhmann, K., Richter, M., & Sandamirskaya, Y. (2013). Learning the perceptual conditions of satisfaction of elementary behaviors. In *Robotics: science and systems (RSS), workshop "active learning in robotics: exploration, curiosity, and interaction"*.
- Martinet, L.-E., Passot, J.-B., Fouque, B., Meyer, J.-A., & Arleo, A. (2008). Map-based spatial navigation: a cortical column model for action planning. In C. Freksa, et al (Ed.), *Spatial cognition VI. Learning, reasoning, and talking about space*. (pp. 39–55). Berlin: Springer.
- Matarić, M. J. (1997). Behavior-based control: examples from navigation, learning, and group behavior. *Journal of Experimental and Theoretical Artificial Intelligence*, 9, 323–336.
- Matarić, M. J. (2000). Sensory-motor primitives as a basis for imitation: linking perception to action and biology to robotics. In K. Dautenhahn, & C. L. Nehavim (Eds.), *Imitation in animals and artifacts* (pp.391–422). Cambridge, MA, USA: MIT Press.
- Milford, M., & Wyeth, G. (2008). Mapping a suburb with a single camera using a biologically inspired SLAM system. *IEEE Transactions on Robotics*, 24(5), 1038–1053.

- Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., ... Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th conference on autonomous robot systems and competitions* (pp. 59–65).
- Nicolescu, M. (2003). *A framework for learning from demonstration, generalization and practice in human-robot domains*. PhD thesis, University of Southern California, USA.
- O'Keefe, J., & Nadel, L. (1978). *The Hippocampus as a Cognitive Map*. Oxford: Oxford University Press.
- Poucet, B., Lenck-Santini, P.P., & Hok, V. (2004). Spatial navigation and hippocampal place cell firing - the problem of goal encoding. *Reviews in the Neurosciences*, 15, 89–107.
- Redish, a.D., & Touretzky, D.S. (1997). Cognitive maps beyond the hippocampus. *Hippocampus*, 7(1), 15–35.
- Richter, M., Sandamirskaya, Y., & Schöner, G. (2012). A robotic architecture for action selection and behavioral organization inspired by human cognition. In *IEEE/RSJ international conference on intelligent robots and systems, IROS*.
- Russell, S., & Norvig, P. (1995). *Artificial Intelligence: A Modern Approach* (1st ed.). Englewood Cliffs, NJ: Prentice Hall.
- Sandamirskaya, Y., Richter, M., & Schöner, G. (2011). A neural-dynamic architecture for behavioral organization of an embodied agent. In *IEEE International Conference on Development and Learning (ICDL)*, vol. 2, pp. 1–7.
- Sandamirskaya, Y., & Schöner, G. (2010). An embodied account of serial order: How instabilities drive sequence generation. *Neural Networks*, 23(10), 1164–1179.
- Sandamirskaya, Y., Zibner, S., Schneegans, S., & Schöner, G. (2013). Using dynamic field theory to extend the embodiment stance toward higher cognition. *New Ideas in Psychology*, 31(3), 322–339.
- Sargolini, F., Fyhn, M., Hafting, T., McNaughton, B.L., Witter, M.P., Moser, M.-B., & Moser, E.I. (2006). Conjunctive representation of position, direction, and velocity in entorhinal cortex. *Science (New York, N.Y.)*, 312(5774), 758–762.
- Schneegans, S. (2015). *COSIVINA (Compose, Simulate, and Visualize Neurodynamic Architectures)*. <http://roboticsschool.ini.rub.de/software.php>.
- Schöner, G. (2008). Dynamical systems approaches to cognition. In *Cambridge handbook of computational cognitive modeling* (pp. 101–126). Cambridge, UK: Cambridge University Press.
- Schöner, G., Kopecz, K., & Erlhagen, W. (1997). The dynamic neural field theory of motor programming: Arm and eye movements. *Advances in Psychology*, 119, 271–310.
- Siegwart, R., Nourbakhsh, I., & Scaramuzza, D. (2011). *Introduction to Autonomous Mobile Robots* (2nd ed.). Cambridge, MA: The MIT Press.
- Spencer, J.P., Perone, S., & Johnson, J.S. (2009). Dynamic field theory and embodied cognitive dynamics. In J. Spencer, M. Thomas, & J. McClelland (Eds.), *Toward a unified theory of development: Connectionism and dynamic systems theory re-considered* (pp. 86–118). New York: Oxford University Press.
- Spencer, J.P., & Schöner, G. (2003). Bridging the representational gap in the dynamical systems approach to development. *Developmental Science*, 6, 392–412.
- Strösslin, T., Sheynikhovich, D., Chavarriaga, R., & Gerstner, W. (2005). Robust self-localisation and navigation based on hippocampal place cells. *Neural Networks*, 18(9), 1125–1140.
- Tani, J., Ito, M., & Sugita, Y. (2004). Self-organization of distributedly represented multiple behavior schemata in a mirror system: reviews of robot experiments using RNNPB. *Neural Networks*, 17, 1273–1289.
- Taube, J.S., Muller, R.U., & Ranck, J.B. (1990a). Head-direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis. *The Journal of Neuroscience*, 10(2), 420–435.
- Taube, J.S., Muller, R.U., & Ranck, J.B. (1990b). Head-direction cells recorded from the postsubiculum in freely moving rats. II. Effects of environmental manipulations. *The Journal of Neuroscience*, 10(2), 436–447.
- Tolman, E.C. (1948). Cognitive maps in rats and men. *The Psychological Review*, 55(4), 189–208.
- Treva, C., Fukazawa, Y., Yuasa, H., Ota, J., Arai, T., & Asama, H. (2002). Cooperative exploration path planning for mobile robots by reaction-diffusion equation on graph. In *IEEE international conference on industrial technology (ICIT)* (Vol. 2, pp. 1266–1271).
- Vazquez-Otero, A., Faigl, J., & Muñozuri, A.P. (2012). Path planning based on reaction-diffusion process. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 896–901). Vilamoura, Algarve, Portugal.
- Wilson, H.R., & Cowan, J.D. (1973). A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. *Kybernetik*, 13(2), 55–80.
- Zibner, S.K.U., Faubel, C., Iossifidis, I., & Schöner, G. (2011). Dynamic neural fields as building blocks for a cortex-inspired architecture of robotic scene representation. *IEEE Transactions on Autonomous Mental Development*, 3(1), 74–91.

## About the Authors



**Erik Billing** obtained his PhD in January 2012 at Umeå University, Umeå, Sweden, focusing on robot learning from demonstration. In 2013, Erik moved to Skövde, Sweden, and is currently affiliated as an associate senior lecturer in informatics, at the interaction laboratory, University of Skövde, conducting research within robotics and embodied cognition. He has been involved in several European projects, including NeuralDynamics and DREAM



**Robert Lowe**, docent of cognitive science, received his PhD from the University of Hertfordshire (UK) in 2007. He currently works at the ICE (interaction, cognition and emotion) laboratory at the applied IT department of the University of Gothenburg and also at the interaction laboratory at the University of Skövde. He focuses on the modelling and empirical analysis of affective brain mechanisms and their applications to cognitive robotics, in particular novel approaches to reinforcement learning. He has worked on several EU funded projects including NeuralDynamics, ICEA, and RobotDoC.



**Yulia Sandamirskaya** has obtained a Diplom degree in physics from the Belorussian State University (Minsk, Belarus) and an MSc from the Ruhr-University Bochum (RUB, Germany) in 2005 and a Dr. rer. nat. degree from Institute of Neural Computation at RUB in 2010. She was leading a group 'Autonomous Learning' as a postdoc at INI in Bochum from 2011 to 2014, involved in NeuralDynamics, joining Institute of Neuroinformatics at the University of Zurich and ETH Zurich in 2015. The focus of her current research lies in development and study of neuromorphic cognitive systems.

## Appendix A. Mathematical formulation

### Dynamic neural fields

The activation level  $u$  of the neural fields constituting the M, P, I, and C components (Section 3) evolves according to equations 4, 5, 6, and 7, respectively

$$\begin{aligned} \tau \dot{u}_M(x, t) &= -u_M(x, y, t) + h + \\ &\int f(u_M(x', y', t)) \omega(x - x', y - y') dx' dy' + \\ &S_2(x, y, t) - S_7(x, y, t) \end{aligned} \quad (4)$$

$$\begin{aligned} \tau \dot{u}_P(x, t) &= -u_P(x, y, t) + h + \\ &\int f(u_P(x', y', t)) \omega(x - x', y - y') dx' dy' + \\ &S_1(x, y, t) - S_7(x, y, t) \end{aligned} \quad (5)$$

$$\begin{aligned} \tau \dot{u}_I(x, t) &= -u_I(x, y, t) + h + \\ &\int f(u_I(x', y', t)) \omega(x - x', y - y') dx' dy' + \\ &S_3(x, y, t) - S_4(x, y, t) - S_6(x, y, t) \end{aligned} \quad (6)$$

**Table I.** Connectivity functions.

$$\begin{aligned} S_1(x, y, t) &= c_1 \int f(u_M(x', y', t)) g(x - x' + \delta x, y - y' + \delta y) dx dy \\ S_2(x, y, t) &= c_2 \int f(u_P(x', y', t)) g(x - x', y - y') dx dy \\ S_3(x, y, t) &= c_3 \int f(u_M(x', y', t)) g(x - x', y - y') dx dy \\ S_4(x, y, t) &= c_4 \int f(u_P(x', y', t)) g(x - x' + \delta x, y - y' + \delta y) dx dy \\ S_5(x, y, t) &= c_5 \int f(u_I(x', y', t)) g(x - x', y - y') dx dy \\ S_6(x, y, t) &= c_6 \int f(u_C(x', y', t)) g(x - x', y - y') dx dy \\ S_7(x, y, t) &= c_7 \int f(u_C(x', y', t)) g(x - x' + \delta x, y - y' + \delta y) dx dy \\ S_8(x, y, t) &= c_8 g(x - x_{robot}(t), y - y_{robot}(t)) \\ g(\Delta x, \Delta y) &= \exp \left[ -\frac{\Delta x^2 + \Delta y^2}{2\sigma^2} \right] \end{aligned}$$

$$\begin{aligned} \tau \dot{u}_C(x, t) &= -u_C(x, y, t) + h + \\ &\int f(u_C(x', y', t)) \omega(x - x', y - y') dx' dy' + \\ &S_5(x, y, t) + S_8(x, y, t) \end{aligned} \quad (7)$$

$\omega$  is a two-dimensional version of the standard Gaussian kernel given by equation 2. Table 1 specifies connections  $S_1$  to  $S_8$ , also visible in Figures 2 and 4.

**Table 2.** List of variables and constants. See also Figure 4 for details on field connectivity.

Symbol	Description	Value				
$u$	Activation level of a neural field					
$x$	Position in a neural field					
$c_{1-8}$	Amplitudes for connections (Table 1). $c_1:12, c_{2a}:5, c_{2b}:-4.5, c_{2c}:4.5, c_{2d}:-10, c_3:10, c_4:-10, c_5:3, c_6:-5, c_7:-15, c_8:8.$					
$t$	Time specified in simulation steps (64 ms/step)					
$\tau$	Time constant controlling field dynamics	5				
$h$	Field resting level	-5				
$\sigma$	Variance of $\omega$ , controlling the width of the excitatory part of the interaction kernel	2				
$v$	Field output to speed conversion (tics/s)	500				
$\alpha$	Constant controlling angular speed	$\pi/3$				
$c_{prox}$	Strength of e-puck proximity sensor stimuli	5				
$\sigma_{prox}$	Standard deviation of the e-puck proximity sensor stimuli	3				
$\check{x}_r$	Field position of stimuli to the proximity sensor field (equations 16 and 17): $\check{x}_0 = 0.1\pi, \check{x}_1 = 0.3\pi, \check{x}_2 = 0.8\pi, \check{x}_3 = 1.0\pi, \check{x}_4 = -1.0\pi, \check{x}_5 = -0.8\pi,$ $\check{x}_6 = -0.3\pi, \check{x}_7 = -0.1\pi.$ The angular field position reflects the physical placement of sensors on the e-puck robot.					
Constants dependent on field type: Motivation (M), precondition (P), intention (I) and condition of satisfaction (C)		M	P	I	C	
$c_{exc}$	Amplitude of the excitatory component of $\omega$	7	4	9	2	
$\sigma_{exc}$	Standard deviation of the excitatory component of $\omega$	2	2	2	2	
$c_{inh}$	Amplitude of the inhibitory component of $\omega$	2	2	0	2	
$\sigma_{inh}$	Standard deviation of the inhibitory component of $\omega$	4	4	4	4	
$c_{global}$	Global inhibition	0	0	0	0	
Constants dependent on transition direction, north (N), south (S), west (W) or east (E)		N	S	W	E	
$\delta x$	Connection shift (corresponding to the translation of the field)	0	0	5	-5	
$\delta y$		5	-5	0	0	

**Action fields**

The action field of each EB (Figure 4) is defined over an angular dimension, from  $-\pi$  to  $\pi$ , with a circular mapping, such that  $u(x, t) = u(x + 2\pi, t)$ , for all  $x$ . The connections (12 and 13) from the action field to the left and right wheels cause the robot to turn to the left ( $\mu_{lw}$ ) and to the right ( $\mu_{rw}$ ) in response to a peak on the negative and positive sides of the action field, respectively. The wheel speeds are given by

$$\mu_{lw} = \vartheta_{max}(x, \alpha) + \vartheta_{min}(x, \alpha) \tag{8}$$

$$\mu_{rw} = \vartheta_{max}(x, -\alpha) + \vartheta_{min}(x, -\alpha) \tag{9}$$

$$\vartheta_{max}(x, \alpha) = \max_{x=-\pi}^{\pi} f(u_a(x, t)) \sin(x + \alpha) v \tag{10}$$

$$\vartheta_{min}(x, \alpha) = \min_{x=-\pi}^{\pi} f(u_a(x, t)) \sin(x + \alpha) v \tag{11}$$

where  $v$  and  $\alpha$  are constants controlling the speed and turning rate of the robot, respectively.  $f$  is the field output function given by equation 3 and  $u_a$  represents the activation level of the action field, given by

$$\pi \dot{u}_a(x, t) = -u_a(x, t) + \int f(u(x', t)) \omega(x - x') dx' + h + S_I(t) + S_H(x, t) - S_O(x, t) \tag{12}$$

where

$$S_I(t) = \max_x [f(u_I(x, t))] \tag{13}$$

$$S_H(x, t) = \int f(u_H(x' + \theta, t)) \omega(x - x') dx' \tag{14}$$

$$S_O(x, t) = \int f(u_O(x', t)) \omega(x - x') dx' \tag{15}$$

$S_I, S_H,$  and  $S_O$  are the stimuli received from the corresponding I-field, the head direction field, and the obstacle field, respectively.  $u_I, u_H,$  and  $u_O$  are activation functions for the respective fields, given by equation 1.  $\theta$  is a directional constant, shifting the input in relation to action field direction. The interaction kernel  $\omega$  is given by equation 2.

**Proximity sensors**

The activation level of the proximity sensor field (Figure 4) is given by

$$-u(x, t) + h + \int f(u(x', t)) \omega(x - x') dx' + \sum_r S_r(x, t) \quad (16)$$

$S_r$  is the stimuli from each proximity sensor  $r$ , given by

$$S_r(x, t) = \log[1 + p_r(t)/50] c_{prox} \exp\left[-\frac{(x - \tilde{x}_r)^2}{2\sigma_{prox}^2}\right] \quad (17)$$

where  $p_r(t)$  is the sensor value of proximity sensor  $r$  at time  $t$ . Here  $c_{prox}$  and  $\sigma_{prox}$  are the constants controlling the strength and standard deviation of  $S_r$ , respectively. See Table 2 for details.

## Appendix B. Follow the carrot

*Follow the carrot* (Barton, 2001) is a simple method for path tracking. As a baseline implementation, *Follow*

*the carrot* was used as a comparison in the evaluation of the proposed architecture.

A steering angle  $\phi_{fic}$  is given by  $\phi_{fic} = k_{fic}(\phi_c - \phi_h)$ , where  $\phi_c$  is the angle between the robot and selected carrot point, and  $\phi_h$  is the heading angle.

The carrot point  $p_c$  is defined as

$$p_c = \min_{p \in P} [d(p, p_{look})] \quad (18)$$

where  $P$  is the set of points on the path, and  $p_{look}$  is the look ahead point at distance  $k_{look} = 1$  cm, directly in front of the robot.  $d(p, p_{look})$  denotes the Euclidean distance between  $p$  and  $p_{look}$ .

The path  $P$  is given by  $A^*$  over an eight neighbor grid with a cell size of 1 cm and a minimum obstacle distance of 4 cm. This path calculation is also used as a reference *optimal path*, Section 5.