Cogn Comput (2013) 5:295–306
DOI 10.1007/s12559-013-9209-0

Improved Path Integration Using a Modified Weight Combination Method

Warren A. Connors · Thomas Trappenberg

Received: 3 November 2012 / Accepted: 12 February 2013 / Published online: 10 March 2013

Abstract Dynamic neural fields have been used extensively to model brain functions. These models coupled with the mechanisms of path integration have further been used to model idiothetic updates of hippocampal head and place representations, motor functions and have recently gained interest in the field of cognitive robotics. The sustained packet of activity of a neural field combined with a mechanism for moving this activity provides an elegant representation of state using a continuous attractor network. Path integration (PI) is dependent on the modulation of the collateral weights in the neural field. This modulation introduces an asymmetry in the activity packet, which causes a movement of the packet to a new location in the field. The following work provides an analysis of the PI mechanism, with respect to the speed of the packet movement and the robustness of the field under strong rotational inputs. This analysis illustrates challenges in controlling the activity packet size under strong rotational inputs, as well as a limited speed capability using the existing PI mechanism. As a result of this analysis, we propose a novel modification to the weight combination method to provide a higher speed capability and increased robustness of the field. The results of this proposed method are an increase in over two times the existing speed capability and a resistance of the field to break down under strong rotational inputs.

W. A. Connors
Defence Research and Development Canada, Dartmouth, NS, Canada
e-mail: warren.connors@drdc-rddc.gc.ca

T. Trappenberg (✉)
Department of Computer Science, Dalhousie University, Halifax, NS, Canada
e-mail: tt@cs.dal.ca

Keywords Dynamic neural fields · Path integration · Trace learning · Stability

Bubbles in the Brain!

In 2003, John Taylor wrote a commentary titled "Bubbles in the brain?" [1] on some work one of the authors [Trappenberg] did with a group at Oxford University [2, 3]. John had shown how dynamic neural field (DNF) models as proposed and analyzed by Amari [4] can be related to many brain functions such as learning somato-sensory maps [5], learning orientation sensitivity in visual cortex [6], and illusions in visual perception [7]. The work by the Oxford group which John commented on proposed an addition to the basic DNF model to incorporate the ability to solve path integration (PI) [8]. The Oxford group related this work to hippocampal representations of head direction cells in [8], to place fields in [9], and to spatial view cells as used in the groups modeling efforts [10]. The group also applied this mechanism to motor learning to which John's comments were primarily directed.

John also wrote a foreword to the first edition of the textbook "Fundamentals of Computational Neuroscience" [11] by one of the authors of this article, where he stressed the importance of consciousness studies. Competitive mechanisms as realized by DNF have been an important ingredient in John's thinking which he outlined in "The Race for Consciousness" [12] based on his previous work with Taylor and Alavi [13]. John's commentary on bubbles in the brain showed his enthusiasm for theories that solve important information processing requirements, such as idiothetic updates of internal representations, in the neural field context. While the specific PI mechanism has demonstrated a new concept with possible biological implementation, no

attempt has been made to this day to optimize the performance of this mechanism. In this paper, we propose an optimized version of PI in dynamic neural fields.

Introduction

Wilson and Cowan introduced rate nodes as a mean field approximation of spiking neurons [14] and combined them with basic anatomical accounts of the cortex to develop a model with local excitation and long-distance inhibition [15]. Amari [4] simplified this model further and formalized it as a field model and analyzed this model formally to show possible solutions of the field. We are here specifically interested in solutions to the field equations with spatially localized field activation called activity packets or bubbles. Amari's analysis was generalized to the case of 2-D fields by Taylor [16] that more closely resemble cortical maps (but also see Doubrovinski and Herrmann [17]). DNF theory has been used to describe many biological phenomena in areas including visual cortex [18], the superior colliculus in primates [19], spatial and episodic memory [20], Hippocampus [8–10, 21] learning somatosensory maps [5], learning orientation sensitivity in visual cortex [6], and illusions in visual perception [7]. More recently, DNF models have been applied to non-biological systems in robotic sensing, motion, navigation [22–24], and cognitive robotics [23].

Dynamic neural fields are a type of recurrent neural network with lateral inhibition that has the capability to develop a localized area of activity (activity packet or bubble) through external stimuli and maintain this localized activity through local cooperation (excitation) and global competition through inhibition. These fields are also commonly referred to as continuous attractor neural networks (CANN), as they possess a continuous manifold of point attractors, allowing for stabilization of noisy inputs into a localized stable state over any node in the field [11]. The dynamics of the field are controlled by center surround kernels, a weighting function that regulates long-distance inhibition and short-distance excitation. This kernel function is typically described by a difference of Gaussians commonly known as a Mexican hat function, or a Gaussian radial basis function shifted by an inhibition constant. These weights can be self-organized through Hebbian learning [8, 9], which allows the training of nodes in the field to encode a sensitivity to a set of features, for example direction, based on external stimuli during training. This sensitivity can be thought of as tuning curves, where a specific node is sensitive to a specific external input.

Animals contain a sense of pose or direction, which suggests that a representation is maintained in the brain that is continually updated from visual or idiothetic cues. A continuous attractor can encode a continuous value such as head direction, or in 2-D, a position of the animal in a plane [25]. To provide such functionality, however, the network must be updated both through visual input and through idiothetic cues such as angular velocities through the vestibular system [25]. A common model proposed for head direction [8, 25, 26] is through the use of a DNF where each node in the continuum is sensitive or "tuned" to a specific direction. This tuning sensitivity, in concert with the dynamics of a CANN above, leads to the development of localized activity, or an activity packet, that is located over a particular portion of the field which is sensitive to the current head location of the animal. Using the above model, however, poses a challenge with respect to idiothetic input.

Self-directed movement of the activity packet caused by angular velocities from the vestibular system requires a mechanism for moving the field activity based on these angular velocities. This poses a challenge with respect to the existing DNF model. Amari [4] showed that in order to provide a stable and static activity packet in the DNF model, this packet must have symmetric excitatory inputs from neighboring nodes. To provide a capability to move this activity, asymmetries must be introduced into the activity packet, modulated by a mechanism for integrating the idiothetic input. This mechanism would allow for control of not only the movement of the packet in a specific direction, but also control over the speed of the movement.

Stringer et al. [8] have shown how asymmetric weights can be learned and how to apply these weights in combination with the neural fields to introduce a mechanism for PI. They used this mechanism to explain updating head direction and place representations from vestibular signals [8, 9]. This mechanism was also applied to motor control [2], showing learning and stable reproduction of a motor sequence. The PI mechanism as applied in these examples was used to demonstrate the functionality of motor control, but no experimental analysis of the behavior of this mechanism was conducted.

The goal of this work is to illustrate the results from an analysis of the PI mechanism and show limitations caused by the method of combining the field and PI weights. These limitations result in instability of the field under strong rotational input and limited range of speed for the packet movement. We propose an alternate method for combining symmetric and asymmetric weights to yield more stable PI over a larger speed range as compared to the original mechanism. In addition, Stringer et al. [2] introduced a stabilization mechanism, referred to as NMDA stabilization, which is examined in more detail as a mechanism to counteract noisy or partially learned field weights. We will show that the proposed weight combination and use of NMDA stabilization will provide a more robust implementation of PI

which allows a wider speed range and resistance to breakdown of the field due to activity packet growth.

Dynamic Neural Fields and Path Integration

This section briefly outlines the basic PI mechanism based on the DNF model, where the neural field $h(x, t)$ describes the activity at location x at time t. These neural fields are governed by the dynamic equation

$$\tau \frac{\partial h(x,t)}{\partial t} = -h(x,t) + \int_y w(x,y) r(y,t) \mathrm{d}y + I^{\mathrm{ext}}(x,t), \quad (1)$$

where $w(x, y)$ are the synaptic weights between node x and node y, $r(y, t)$ is the firing rate of the field at location y

$$r(x,t) = \frac{1}{1 + \mathrm{e}^{-\beta(h(x,t)-\alpha(x))}}, \quad (2)$$

and $I^{\mathrm{ext}}(x, t)$ is the external input to the field at time $t.\tau$ is a time constant, and the constants β and α describe slope and the offset of how the field h is related to the firing rates r. We are specifically interested in solutions of this system with stable localized activity packets that can encode and even memorize a pose of a system such as head directions, or in 2-D, a location on a plane. These fields are often approximated with discrete nodes that are labeled with index i, which in turn is related to the field location by

$$x = i\Delta x. \quad (3)$$

A 1-D network of such pose cells is shown in Fig. 1. This model is organized as a ring; therefore, the boundaries of the field are at 0 and 2π. The weights between these pose cells can be learned by Hebbian learning [8],

$$\tilde{w}(x,y) = \int_0^{2\pi} r(x - x^{\mathrm{p}}) r(y - x^{\mathrm{p}}) \mathrm{d}x^{\mathrm{p}}, \quad (4)$$

based on Gaussian radial basis function training patterns centered around the preferred direction, x^{p},

$$r(x - x^{\mathrm{p}}) = \mathrm{e}^{-(x-x^{\mathrm{p}})^2/2\sigma_r{}^2}. \quad (5)$$

This preferred direction, x^{p}, can be seen as the direction at which the node represents, or the direction "tuning curve." To implement the periodic boundaries, we replace the distance between x and x^{p} with,

$$|x - x^{\mathrm{p}}| \rightarrow \min(|x - x^{\mathrm{p}}|, 2\pi - |x - x^{\mathrm{p}}|). \quad (6)$$

Finally, we use a scaled and shifted version

$$w(x,y) = A_w(\tilde{w}(x,y) - C) \quad (7)$$

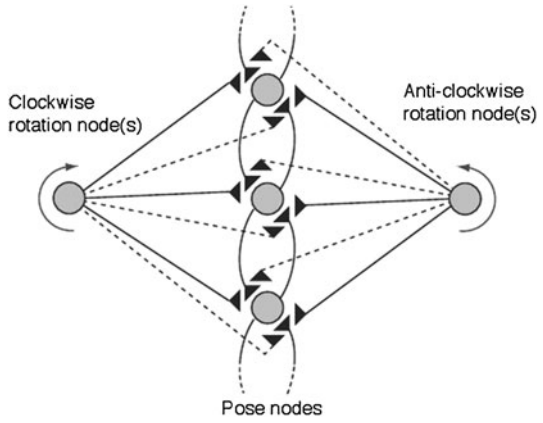

Fig. 1 The model for path integration showing a 1-D manifold of pose cells with collateral connections (such as cells encoding head directions). The rotation nodes signal rotation velocities proportional to their activity, which is propagated through synapses to modulate the activity of the field, causing an asymmetry in the activation, and moving the activity packet

of this symmetric weight kernel between the pose cells, where C is a inhibition constant, and A_w is a scaling factor for the pattern, depending on the strength of the training pattern, and the learning rate [11].

Path integration requires the integration of the neural activity from rotation cells, as shown in Fig. 1, to shift the neural activity packet. Rotation cells, denoted by r^{rot}, were proposed by Skaggs et al. [27] as an additional set of cells that control the movement of the packet through preferential synaptic connections for clockwise and anti-clockwise directions. These nodes provide preferential connectivity of clockwise and anti-clockwise rotation nodes, resulting in modulation of the collateral connections in the field based on the firing rates of the rotation nodes. These nodes represent the vestibular input, where the firing rate of the rotational node is directly proportional to the angular velocity. This modulation in effect introduces an asymmetric connectivity between pose cells, allowing the packet to move. More specifically, the model shown in Fig. 1 as proposed by Stringer et al. [8] proposes a modulatory influence from rotation nodes on the pose cell connectivity, which is equivalent to a sigma-pi network [8] as described by

$$\tau \frac{\partial h(x,t)}{\partial t} = -h(x,t) + \int_y w^{\mathrm{eff}}(x,y,r^{\mathrm{rot}}) r(y,t) \mathrm{d}y, \quad (8)$$

where $w^{\mathrm{eff}}(x, y, r^{\mathrm{rot}})$ are the effective weight kernels, which describe the effective connectivity between the pose nodes. This kernel combines the rotation weights, rotation node firing rates, and the existing field weights. The specific form chosen by Stringer et al. for this weight combination is,

$$w_{ij}^{\text{eff}} = w_{ij} \sum_k \left(1 + w_{ijk}^{\text{rot}} r_k^{\text{rot}}\right), \tag{9}$$

where r_k^{rot} is the clockwise and anti-clockwise rotation node firing rates. The rotation weights are learned with a Hebbian rule for the rotation node connectivity, incorporating a historical trace or "short-term memory" of previous movement during training. This trace allows for the association of the movement of the activity packet with the firing of the appropriate rotation node. Hebbian learning of the rotation weights using a trace for association of movement with rotation node firing rates is given by,

$$\delta w_{ijk}^{\text{rot}} = \epsilon r_i \bar{r}_j r_k^{\text{rot}}, \tag{10}$$

where $\bar{r}$ is the trace of previous activity

$$1/\eta \frac{\partial \bar{r}_i}{\partial t} = -\bar{r}_i(t) + r_i(t). \tag{11}$$

The η term controls the amount of influence earlier movements of the activity packet have on the synaptic weights, and can be used to increase the effective asymmetry of the neural field. This trace provides a short-term history of the activity packet movement, which is associated with the firing of the rotation cells and the movement of the activity packet using (10), where ϵ is a learning rate.

Earlier work with path integration models [28] was centered around the idea of pre-specified weights, specified during the development of the field. Indeed, some current applications of path integration for RatSLAM [29] robotic navigation have implemented pre-specified translation of the activity packet to ensure stable performance of the system over a widely varying set of speeds. The work by Stringer et al. [8] showed a method of learning based on a history of movements. This history provides both an ability of the system not only to learn from a short-term history or memory, but also to allow the encoding of specific motions in the movement model itself, rather than just generic motion. The learning of these asymmetries, as outlined above, provides the ability for a PI system to learn a temporal sequence of movements, as illustrated in [2].

To provide a stable activity packet, the DNF weights must provide symmetrical recurrent input. In cases of partial or noisy training, this may not be the case, where improper or inconsistent weights can cause asymmetries. These asymmetries can cause the packet to drift, therefore not providing a stable solution for a memory of an encoded feature, such as a direction. To counteract these stability issues, a nonlinear activation method proposed by Stringer et al. [8] has been applied to the activity of the pose cells. This proposal considered that the nonlinear activation could be implemented biologically through the effects of voltage dependent ion channels, such as NMDA receptors. When neurons are at rest, these receptors are blocked by magnesium ions and are inactive. An increase in the membrane potential of the neuron will remove this block, allowing the neuron to fire under lower stimulus input for subsequent time steps. Using a nonlinear activation such as NMDA receptors results in a group of neurons which can be activated with lower stimulus at time $t + 1$ if they were active at time t. The NMDA stabilization provides a resistance of the activity packet to move due to noisy weights by boosting the previous activity of the field. This method can be implemented in the model through the variation of an offset in the sigmoid activation function, depending on the field activity in the previous time step. The offset for the sigmoid activation function in Eq. 2 is described by,

$$\alpha_i = \begin{cases} \alpha^{\text{high}} & \text{if } r_i < \gamma. \\ \alpha^{\text{low}} & \text{if } r_i \geq \gamma. \end{cases}$$

where γ is a firing threshold to activate the NMDA stabilization, and α_i is the offset that is applied to activation values being input into the sigmoid. This method has the effect of boosting the activation of the nodes in the field which were firing above the threshold in the time step before and therefore causes an effective strengthening of the recurrent input to the nodes that are active. As PI requires a stable set of field weights, the qualities of NMDA stabilization are attractive in cases of noisy field weights or irregular training.

Analysis of Path Integration Methods

Analysis Model

The following analysis was based on the DNF/PI model above. The field was implemented as a ring, using periodic boundaries. This field was discretized using 360 nodes, each corresponding to one degree of head direction. The field was set to a moderate level of inhibition $C = 0.5$, and kernel weights for each node were trained using a Gaussian radial basis function, where the mean is the preferred direction, and $\sigma = 2\pi/18$. For training the rotation weights, using trace learning a window size (η) of 0.2 was used along with a learning rate (ϵ) of 0.1.

The activity packet encodes the pose or position of the system; therefore, the question is how to decode this position from the activity. A simple max decoding scheme, where the node corresponding to the maximal activity is selected, can be incorrect due to small fluctuations in the activity packet. These fluctuations in the size and shape of the packet can cause drastic shifts in the selected node. Furthermore, this decoding mechanism does not take into account the contribution of the surrounding nodes. To correct for these decoding limitations, a center of mass

calculation is used. This population decoding method is described by,

$$\text{CoM} = \sum_i r_i i \Delta x \tag{12}$$

where i is the index of the pose node, and r_i is the activity at node i. One challenge of this method of decoding for fields with periodic boundaries is calculating the center of mass when the activity packet is spanning the boundary. Simply applying the method above results in an incorrect calculation of the center of mass. To compensate for the boundaries, the activity packet is translated to the center of the field where no activity crosses the boundary, then the center of mass is calculated. The reverse translation that was applied to the activity packet is applied to the center of mass, resulting in the center of mass of the packet that is spanning the boundary.

To illustrate the difference between the decoding methods, consider Fig. 2. In this simulation, a subset of the weights were underdeveloped, having peaks that were slightly lower than the other kernel weights in the field. When the bubble travels over these weights, it causes an unintended change in shape of the activity packet. Using max activity decoding, however, results in the pulsing behavior seen in Fig. 2a, where the activity packet appears to slow down and then speed up, showing inconsistent speed over that area of the field. Using a center of mass calculation considers the entire activity packet, rather than just the node with maximum activity, therefore does not exhibit the sensitivity to the change in shape of the packet. This can be seen in Fig. 2b, where the speed is consistent throughout the simulation.

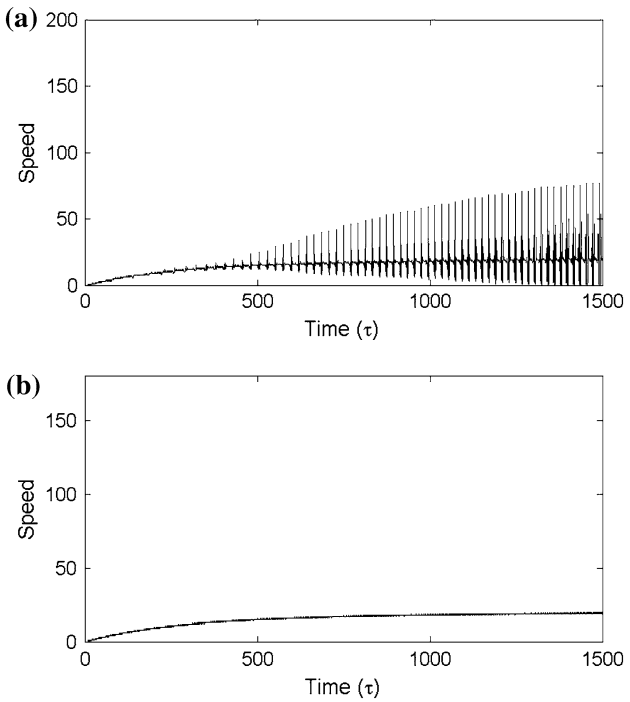

Fig. 2 Illustration of the population decoding differences for max versus center of mass decoding. Using max activity decoding (**a**) over undeveloped weights results in an inconsistent speed calculation of the activity packet due to changes in shape of the packet. Using center of mass decoding (**b**), the speed calculation is consistent over the field

For each simulation, the field was stimulated through external input to develop an activity packet at pose node 180. Once this has been formed, the external input was removed and the PI mechanism would be used to control the movement of the packet. This model was developed for the MATLAB® environment, using the "ode45" differential equation solver to compute activity of the field which is a non-stiff solver using the Runge–Kutta algorithm. Furthermore, results were verified using other ode solvers available through MATLAB®. For speed tests of the activity packet, the difference between the previous position and the current position was compared at each time step, taking into account the boundary conditions of the field. To examine the stability of the field, the simulation would increase the rotation rate input over time, causing a ramp up in the activity packet speed during the simulation.

Training

A challenge to DNFs and PI is the creation of perfectly symmetric states to sustain a stable activity packet. As already addressed by Zhang [21], noise in the weights would cause drift of the activity packet, thereby deteriorating the pose memory. To provide the perfect symmetry in these weights, the fields should be trained carefully, considering every node for the same amount of time. This allows for a fully symmetrical set of synaptic weights to be formed. These ideal learning conditions are not always present, which can lead to underdeveloped synaptic weights over parts of the neural field or noisy weight kernels as illustrated in Fig. 3. The noisy or incomplete training of the field weights can result in unwanted asymmetries in the activity packet, causing the packet to drift without rotational input. This effect is illustrated in Fig. 4, where the activity of the field drifts due to noisy weights without external stimuli or PI.

In the case where incomplete training results in underdeveloped weights, the result is a point attractor network. This network will consist of a series of point attractors, which correspond to the pose nodes with the most developed weights. When an activity packet is formed, it will drift to the nearest pose node with fully developed weights, and then settle over this pose node. Figure 5 illustrates the point attractor effect, where a simple DNF of 100 nodes was developed to show the effect of noisy weights. In this plot, a simple DNF was developed and trained with 10 sets

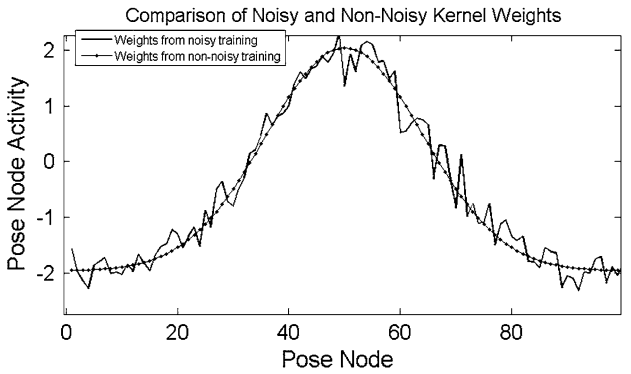

Fig. 3 Comparison of kernel weights for node 50, showing the typically symmetrical weights that the fields are trained on, and the same weights with noise added

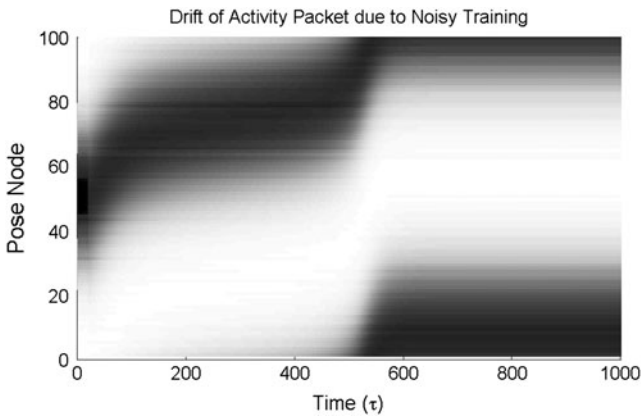

Fig. 4 Surface *plot* of the field activity for a simple, 100 node neural field when noisy weight kernels are used. The asymmetries caused by the noisy weight kernels causes drift in the activity packet, resulting in inconsistent activity in the field

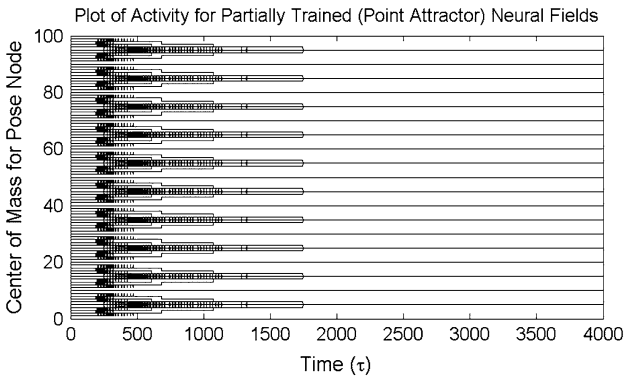

Fig. 5 The centers of mass for activity packets formed from partial training, resulting in a point attractor type field, where only 10 node weight kernels were fully developed

of fully developed weights spaced equally throughout the field. To plot the point attractor effect, 100 simulations were run, one for each node, where input was provided into each pose node of the field and the movement of the activity packet recorded. The composite plot in Fig. 5 was developed over time to show the point attractor effect, where each trace in the plot is the movement of the activity packet formed over each pose node. These traces show the activity packets formed over each node converging to the closest point attractor.

As noted above, to provide a stable neural field for path integration, a stable, non-drifting activity packet, and therefore, symmetrical neural field weights are required. Symmetrical weights can be developed through careful training where each node is considered with the same inputs and for the same time. If this is not possible, then a stabilization mechanism such as the NMDA stabilization which is considered in this work can be used to counteract the effect of asymmetrical or noisy weights.

Maintaining Appropriate Inhibition During Weight Combination

The inhibition of the DNF is critical to ensure that a stable activity packet is formed. As each node is fully connected to all other nodes, distance-dependent excitatory input is provided to all nodes when a particular pose node is excited. The inhibition in the field counteracts this effect, resulting in a single, localized packet of activity. In the case where the inhibition is insufficient or absent, this results in the uncontrolled growth of the activity packet, and a decrease in the utility of the field as it would become fully excited [4]. A careful maintenance of inhibition is key to both DNF and PI, to prevent an unstable state of the network where the activity packet grows uncontrolled. The key to any weight combination method for PI is to combine the weights in a meaningful way that allows for an asymmetry in the activity packet, while maintaining the inhibition at a rate where the packet size remains stable.

The existing method for weight combinations in Eq. 9 uses the addition of a constant to allow for the resulting effective weights to maintain inhibition, therefore maintaining some stability. This addition of a constant is required as the rotation weights have a minimum value of zero (Fig. 6), therefore do not contain inhibition. The multiplication of the DNF weights which include inhibition and PI weights which do not would result in an effective weight kernel which does not contain any inhibition. This loss of inhibition results in the field moving to a stable state of full activation. The addition of the constant to the rotation weights allows the inhibition to be maintained during the multiplication of the weights; however, it also has the undesirable effect of increasing the amplitude of the weight kernel. This increase in amplitude increases the amplitude of the neural field kernel, which will lead to activity packet growing due to the limit of the sigmoid activation function and eventually breaking down. Any

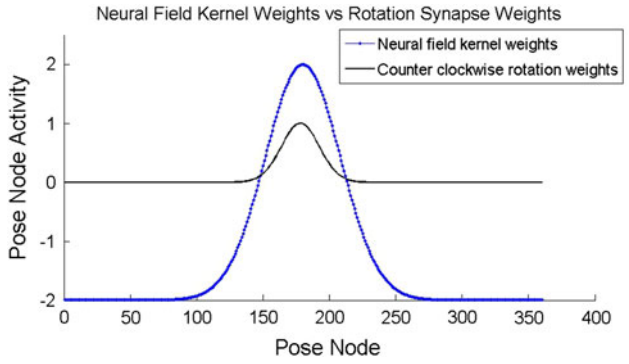

Fig. 6 Comparison of neural field kernel and rotation synapse weights. The rotational synaptic weights, which have a minimum value of 0, require care when combining multiplicatively with the kernel weights, to ensure that the inhibition is not removed from the field. Without this inhibition, the weights would result in a growing state where the entire field becomes excited

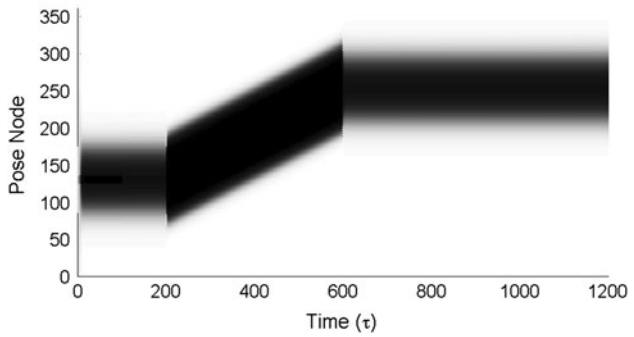

Fig. 7 Illustration of the activity packet using PI with a low rotational input between $t = 200$ and 600, showing that even under low rotational input, the activity packet will increase in width, due to the high activation rates

alternative method for weight combination must seek to minimize the growth of the activity packet, while maintaining the inhibition to ensure that the packet remains stable.

As noted above, the combination of weights and rotation rates for path integration are multiplicative; therefore, they run the risk of increasing in amplitude to orders of magnitude larger than the neural field weights. This increased amplitude due to rotational input causes a growth in the activity packet width, which if uncontrolled, can result in an inability of the field to maintain a localized solution of pose as the maximal activity is over a large area. Figure 7 illustrates this, where under a low rotational input, there is a noticeable increase in the activity packet width. This effect can also cause an unstable growth in the network during PI, resulting in a growing packet which eventually saturates the field. Figure 8 illustrates this effect, where rotation rates are increased to illustrate the growth of the activity packet.

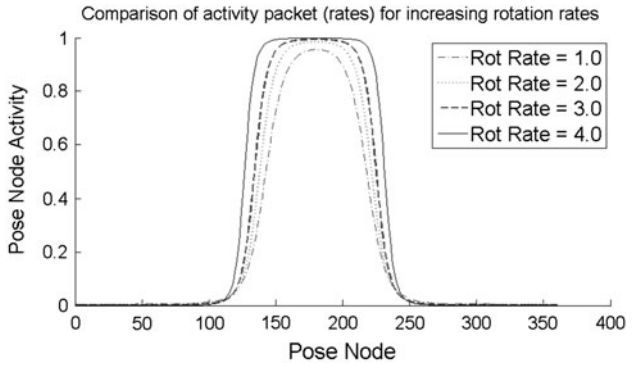

Fig. 8 Comparison of activity packet size under differing rotation rates, showing the effective growth of the activity packet under strong rotational input. This growth eventually leads to an uncontrolled excitation of the field

To provide control of the activity packet size during PI, a method to control the amplitude of the effective weight kernel is required. One possible technique would be to apply a normalization to stabilize the amplitude of these weights, when used after the rotation weights and rates are combined with the neural field. Using the existing weight combination method, however, the normalization does not have the desired effect. Consider the combination of the rotation rate and synaptic weights (Fig. 6). The PI rotation weights have no inhibition (negative component); therefore, a positive constant (1) is added to the result of the weights multiplied by the rotation rates in Eq. 9 to maintain the fields inhibition during the weight combination. This has the effect of amplifying the positive weights further, while leaving the inhibition at the same rate. When the scaled rotation weights are combined with the field's weights through multiplication, the inhibition is maintained at the fields rate; however, the excitation (positive) portion of the activation is scaled up. A simple normalization of these resulting activations to a range that is equivalent to the field's rates will shrink the peak, but it will also decrease the inhibition, effectively shifting up the kernel. As the rotation strength increases, eventually the inhibition is decreased to approach zero. The decreased inhibition causes growth of the activity packet, eventually causing the field to be fully excited and break down.

Proposed Modifications to Path Integration

Alternative Weight Combination Methods

The current weight combination method, as outlined in Eq. 9, introduces specific characteristics to the path integration mechanism which can be limiting in application. The goal of this work is to examine alternative weight combination methods to provide a more robust

implementation of the path integration method which provides more stability in the kernel weights, while increasing the overall range of speeds of the path integration with respect to rotational input. Specifically, we would like to find a method which controls the growth of the activity packet, limits any instabilities caused through learning, maintains the required inhibition in the field, and increases the overall speed range of the rotational input.

In this section, we propose a new method of combining the rotation weights and kernel weights, to eliminate or greatly limit the effects noted above, while maintaining or improving the effectiveness and computational tractability of path integration. The weight combination mechanism (Eq. 9) uses a multiplicative combination of the synaptic rotation weights, the firing rate of the rotation node, and the collateral weights of the neural field. As noted in "Dynamic Neural Fields and Path Integration" section, this combination causes a very rapid growth of the activity in the field and amplifies any inconsistencies in the rotational weights caused by the trace learning.

Trace learning results in a set of synaptic weights that are slightly skewed from the overall kernel weights in the neural field. This slight distortion, when combined with the neural field weights, results in a skew of the weight kernel in the appropriate direction. As the goal of the weight kernel is to cause an asymmetry in the activity packet, rules which increase this skew will increase not only the overall top speed at which the activity packet will move, but also the sensitivity of the asymmetry to input from the rotational nodes.

The proposed method is a modification of Eq. 9 to use the addition of the neural field kernel weights with a shifted version of the rotational weights trained from Eq. 11 and scaled by the rotation speed. This method is described by

$$w_{ij}^{\text{eff}} = w_{ij} + \sum_{k} (w_{ijk}^{\text{rot}} - \mu) r_k^{\text{rot}} \tag{13}$$

where w_{ij}^{eff} is the effective neural field weights from i to j, w_{ij} is the neural field kernel weights, w_{ijk}^{rot} is the trained rotational synaptic weights, and r_k^{rot} is the rotational synaptic strength. Any value of the inhibition constant μ can be subtracted from the weight kernel, as long as this value maintains the overall inhibition of the field at an appropriate rate. We use in the following work the mean of the rotational synaptic weight for modeling; however, biological implementations would likely use adaptive mechanisms to maintain the inhibition. This method removes the multiplication of the neural field and rotational weights, while still maintaining inhibition in the resulting effective weight kernel, through the subtraction of the mean value for the kernel. The training of both the neural field and the rotational synaptic weights is identical, and normalization is applied to regularize the weights.

Results

Weight Combination

Through addition rather than multiplication of the weights, the overall asymmetry of the kernel is increased, which is illustrated in Fig. 9. The addition supports the widening of the kernel, specifically on the appropriate side of the required rotation allowing more recurrent input to distort the activity packet, thereby moving the bubble in a more efficient manner than the multiplicative method. The existing weight combination method (Eq. 9) amplifies the distortion at high weight values (e.g. near the center of mass), but maintains a narrow profile for the weights (Fig. 9a). The modification of the effective weight kernel allows for a finer control of the activity packet speed, which results in a larger range of speeds for a given set of rotation strengths. The addition further has the effect of increasing the amplitude of the activity at a much slower rate as seen in Fig. 9 which, when run through a sigmoidal

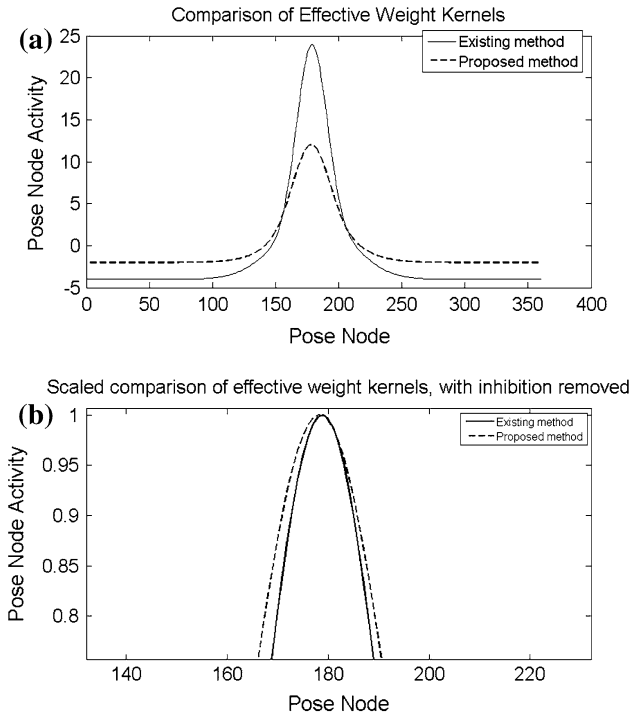

Fig. 9 Comparison of effective weight kernels of node 180 for counter clockwise rotation of a 1-D field. *Plotted* together (**a**), the amplitude difference between methods is shown, where the existing method grows rapidly, resulting in rapid growth of the activity packet leading the field to be fully excited. When scaled together and with inhibition removed, the differences in the kernel weights become more apparent, where the proposed method contains more skew in the counter clockwise direction (*left*), allowing more recurrent input in the activity packet, creating the desired asymmetry. This is illustrated in **b**, where a portion of the weights have been magnified to view the weight differences

activation function such as Eq. 2, results in a narrower activity packet, and extends the intensity of rotational input that can be maintained by the field before an uncontrolled growth of the activity packet.

The movement speed of the activity packet is controlled through the asymmetries in the effective weight kernel. This speed, which can be increased by increasing the asymmetry, has an asymptotic speed limit and can be noted in Fig. 10. As noted above, the proposed method for weight combination results in a wider, more asymmetrical input to the field, thereby allowing for a higher maximum speed as well as a wider range in inputs from the rotational nodes as shown in Fig. 10b.

Figure 10 compares the movement speeds of both the existing weight combination method (a) and the proposed new weight combination method (b). These methods were compared by increasing rotational input to the fields on kernels trained using different training speeds. The training speeds are defined by the rate of movement that is shown to the trace during training, therefore allowing the trace to learn that a given movement and rotation rate input covers a larger area of the field. The goal of this figure is to illustrate the difference in speeds across different training rates, as well as the stability of the field to high rotational input. As illustrated in Fig. 10b, the proposed method achieves a higher overall speed of more than two times the existing method. Furthermore, the proposed method shows a higher speed range for the same set of rotational inputs, allowing for more robust control of the activity packet speed. Through the additional rather than multiplicative combination of the synaptic weights with the field weights, the proposed method shows a higher resistance to the breakdown of the field due to growth of the activity packet, resulting in stability over higher rotational inputs. This effect is illustrated in Fig. 10, where the field breaks down under increasing rotational input (a). This breakdown in the field is prevented in (b), showing increased stability over (a). The improvements from the proposed weight combination method can be noted in Fig. 10, where the speed of the activity packet for fast training (32) shows the exceeding the size of the field at rotation rate input >37. This high speed results in a wrapping effect in the figure, while still showing stable activity.

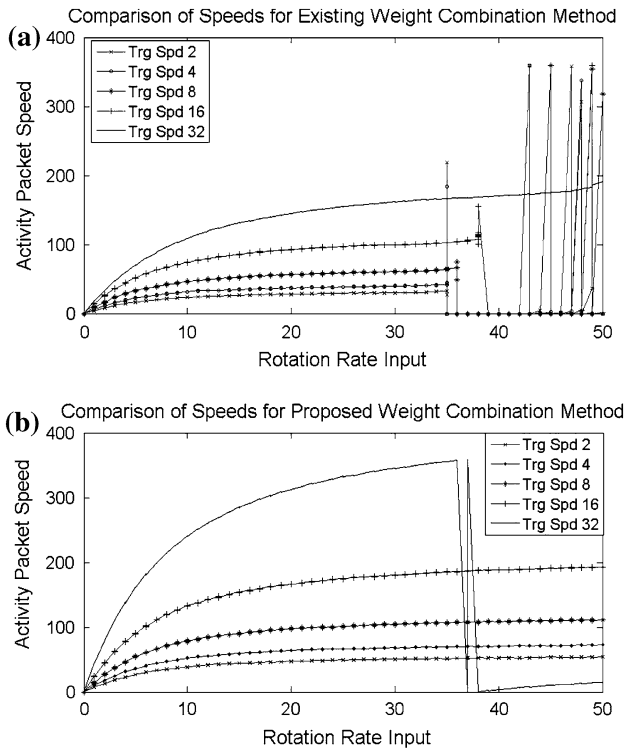

Fig. 10 Comparison of existing weight combination method (**a**) and the proposed weight combination method (**b**).This comparison illustrates the higher top speed insensitivity to the spiking behavior, higher overall speed, and stability under a larger range of rotational inputs

NMDA Stabilization of Path Integration

Finally, we want to demonstrate that NMDA stabilization is an effective stabilization technique for path integration. Intuitively, the nonlinear activation using thresholds allows for an added level of resistance to movement or "stiffness" in the activity packet. This resistance is defined by the sigmoid thresholds α^{high} and α^{low}. In the case of a noisy kernel, which results in an activity packet that drifts, this stabilization results in a fully stable activity packet. Figure 11 illustrates this stabilization effect, showing the activity when trained with noisy input, causing drift in the packet (a). The NMDA stabilization dampens the noise sufficiently to maintain a stable activity packet, as shown in (b).

As noted in "Dynamic Neural Fields and Path Integration" section, partially developed weight kernels can result in a point attractor network, as illustrated in Fig. 5. For partially developed weight kernels, a high value of α_{high} reverses the point attractor behavior of the field; however, this value creates some "stiffness" to the mobility of the activity packet, requiring strong input at alternate locations of the field to form a new activity packet. Figure 12 shows the effect of NMDA stabilization using a value of $\alpha_{high} = 10$ for the nodes with an activity under the threshold on the partially trained field in Fig. 5 which resulted in a point attractor. Using NMDA stabilization results in a reversal of the point attractor behavior, allowing for stable activity across the field.

In consideration of path integration, NMDA stabilization is effective at stabilizing the field during movement, as

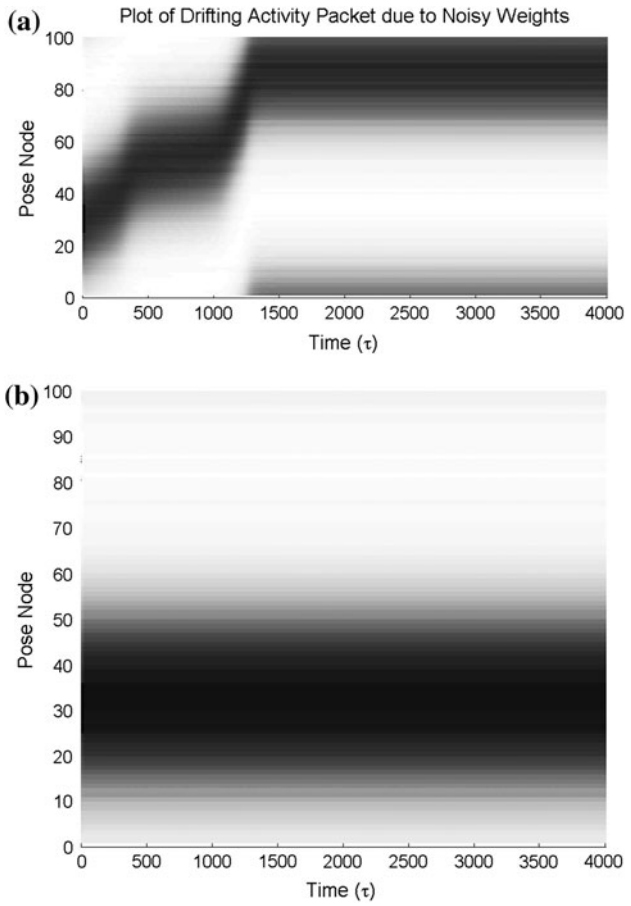

Fig. 11 Comparison of the effects of nonlinear activation through the application of a firing rate-dependent threshold. **a** The asymmetrical activity packet which resulted from noisy weights in Fig. 3, and **b**, the result of using non-linear activation to dampen the drifting through boosting the most active neurons

well as when the rotation nodes are not firing. This allows for a stabilization of the activity packet from drift when not moving, but also smoother movement of the activity packet when trained on noisy weights. Figure 13 illustrates the smoothing effect. The field was trained on noisy weights, causing both drift in the activity packet when at rest, but also inconsistent movement of the activity packet when rotational input is applied (Fig. 13a). In the figure, rotational input was applied from $t = 100$ to 500 and shows that the packet was stabilized both when rotational input was applied, and also when the rotational nodes were at rest (Fig. 13b).

This form of stabilization requires an increase in the rotation input to control the activity packet. This is a side effect of the resistance of the activity packet to move due to stabilization. This increased input is offset by the utility of the stabilized packet, as it shows stability from drift and consistent movement of the activity packet when a rotational input is supplied.

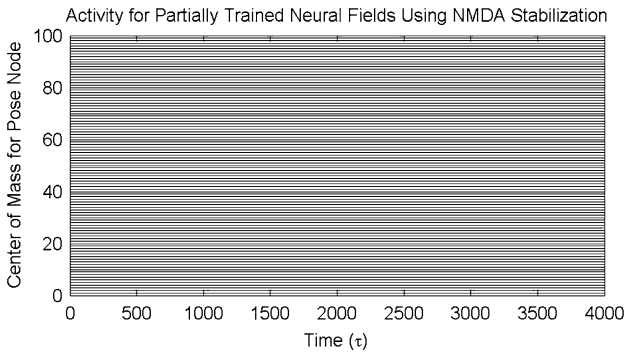

Fig. 12 Comparison of the effects of nonlinear activation through the application of a firing rate-dependent threshold on the partially trained network shown in Fig. 5. The result of using nonlinear activation with a high α_{high} value (10) to dampen the point attractor effect through boosting the most active neurons, which effectively counteracts the point attractor effect

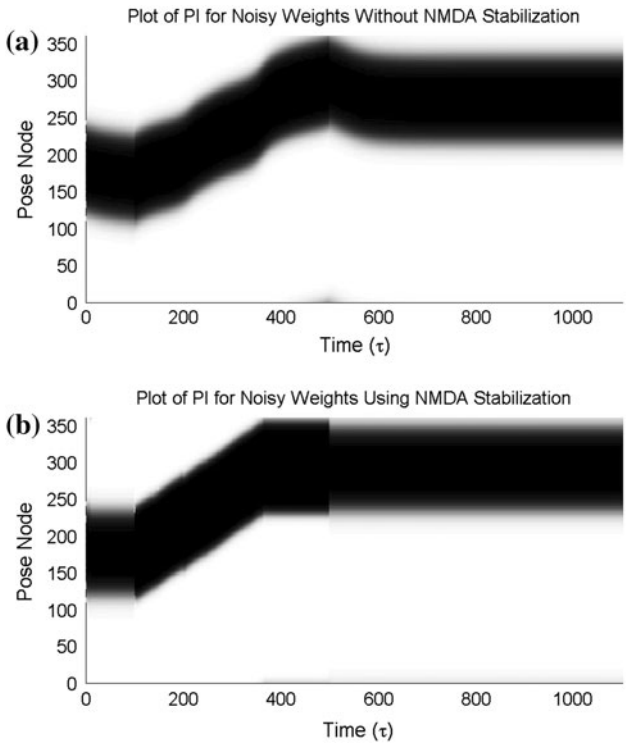

Fig. 13 Comparison of the effects of nonlinear activation through the application of an firing rate-dependent threshold on a field trained with noisy input which is used for path integration (**a**) a high α_{high} value (10) to dampen noise effects. The asymmetries introduced through the noise in the weight training results when no rotational input is supplied (t = 0–100, t = 500–1,100) and inconsistent movement when PI is applied (t = 100–500). The application of NMDA (**b**) dampens the effect of drift and allows for stable path integration speed

Discussion and Outlook

The preceding work is a result of an analysis of the existing path integration method proposed by Stringer et al. [2] and

considers issues of performance and stability. This work explored the boundaries of the performance of the path integration system, with respect to training speed and rotational input. Through the characteristics noted above, most were traced back to the method that the synaptic weights from the rotation nodes and the field weights were combined. Issues discovered in "Dynamic Neural Fields and Path Integration" section included noisy or partial training, the rapid growth of the activity packet, and the low speed range of the field to rotational inputs.

Partial or noisy training for path integration also has a large effect in the movement of the activity packet, therefore causing instability of the motion and the field. These instabilities result in a drifting packet when no rotational input is supplied to the field, and also inconsistent movement of the activity packet during movement. The application of a nonlinear activation method for the field results in a stabilization of the rates in the field, which in turn results in not only a removal of drift in the activity packet, but also stable movement of the activation when rotational input is applied. Although this method has the desirable quality of resisting noise and incomplete training, it also will limit the competition effect of the field. The selection of the threshold as well as the offsets α^{high} and α^{low} is critical to ensure that these values resist the drift in the field, while allowing for the formation of a new activity packet at a different area of the field. These thresholds will be application dependent, in consideration of the maximum activation of the field, as well as the level and amplitude of noise in the weights. Using path integration with NMDA stabilization requires higher rotational input due to the stiffness of the field, however, as path integration involves the combination of weight kernels between the field and rotation weights, this causes a smoothing effect for noisy weights. Due to this effect, the value of α^{high} can be lowered, therefore lowering the required rotation input for the packet movement.

The rapid growth of the activity packet was caused by the method at which the weights were being combined. The multiplication of the weights caused a rapid increase in the amplitude or intensity of the excitatory weights; however, it did not conversely increase the width or skewness of the distortion of the packet. This behavior results in a rapid increase in the activity of the field, but not a proportional rapid increase in the speed of the bubble movement through skew in the weight asymmetry, or sensitivity to rotational input. This mechanism, combined with the sigmoid activation function, resulted in the rapid width increase in the activity packet, thereby increasing the collateral excitatory input given to other nodes in the field. As this excitatory input increased, the field was driven to an overall higher state of activity, eventually resulting to the entire field being excited and breaking down. Although it is ideal to consider normalization to correct this, the rapid amplitude increases resulted in normalization decreasing the inhibition, unfortunately causing the very effect we were trying to prevent.

The proposed weight combination rule allows for a more robust implementation of the weight combinations for path integration. This robustness is shown in a higher overall top speed, a greater dynamic range for control of the activity packet through rotation rates, and a more robust control of the activity packet size. Computationally, this method reduces a complex convolution of weights to a simple addition, which increases the computational tractability of the rule when extended to multiple dimensions.

For modeling human or animal behavior, this method provides a technique for better control of the activity packet size and therefore stability of the simulations. For non-biological applications (e.g. robotics), this method allows for a faster movement of the packet based on input and a more consistent movement of the packet. For example, the head direction model in rodents could be extended to develop a gyrocompass functionality in a robot. To illustrate, consider an environment where a compass is not a reliable sensor, such as an area of high iron content, or potentially an area where magnetic fields are unpredictable (e.g. arctic environments). Through the use of a dynamic neural field using path integration, and a simple sensor which can output rotational speeds, this model could be used to integrate accelerations in a stable manner to produce a simple gyrocompass. Extended to 2-D, path integration allows for navigation; however, it requires fine and stable control of the field for integration of low-level accelerations. The proposed kernel not only offers a stable solution to the path integration models for animals, but could also be applied to other model tasks where a high stability and better control is required.

As noted in the current work, the inherent speed limit in the movement of the bubble is directly related to the asymmetry in the activity packet in the field. This asymmetry is controlled through the synaptic weights that are used to modulate the shape of the activity packet. In this study, the weight kernels applied were learned through self-organization and resulted in slightly skewed versions of a radial basis function. These weights were highly similar to the learned weight kernels of the neural field. In order to influence the speed and dynamic range of the activity packet with respect to rotational input, alternative kernels which increase the skewness of the packet could be considered. These kernels could increase the skewness to resemble an ex-Gaussian rather than a Gaussian distribution, using the long tail of the ex-Gaussian to influence the excitatory influence of the field. Furthermore, this could be a contrived kernel designed specifically to achieve a specific goal, potentially resembling a time varying gain curve.

Further work on path integration should consider the influence of these types of kernels, specifically for the role of non-biological implementations of neural fields for topics such as robotic navigation.

The different kernel functions can also be related to different types of learning. The trace rule is an effective method for self-organizing the synaptic weights for modulating the collateral weights in the neural field. This method uses exponential decay of the trace, which rapidly limits the influence of previous actions of the activity packet in time. This decay rate is an area where further work should be dedicated, to consider potential other methods of decay, such as a stretched exponential, or a linear decay, to potentially increase the skewness of the synaptic weights, allowing for even finer control of the activity packet.

References

1. Taylor J. Bubbles in the brain? Trends Cogn Sci. 2003;7:429–31.
2. Stringer SM, Rolls ET, Trappenberg TP, de Araujo IET. Self-organizing continuous attractor networks and motor function. Neural Netw. 2003;16:161–82.
3. Stringer SM, Rolls ET, Trappenberg TP. Self-organising continuous attractor networks with multiple activity packets, and the representation of space. Neural Netw. 2004;17:5–27.
4. Amari SI. Dynamics of pattern formation in lateral-inhibition type neural fields. Biol Cybern. 1977;27:77–87.
5. Petersen RS, Taylor JG. Reorganization of somato-sensory cortex after tactile training. In: Touretsky DS, editor. Advances in neural information processing, systems. Cambridge: MIT Press; 1996. p. 82–8.
6. Fellenz W, Taylor JG. Establishing retinotopy by lateral inhibition-type homogeneous neural fields. In: ESANN proceedings; 2000. p. 200–49.
7. Taylor JG. Perception by neural networks. Neural Netw World. 1997;4:363–95.
8. Stringer SM, Rolls ET, Trappenberg TP, de Araujo IET. Self-organising continuous attractor networks and path integration: one-dimensional models of head direction cells. Netw Comput Neural Syst. 2002;13:217–42.
9. Stringer SM, Rolls ET, Trappenberg TP, de Araujo IET. Self-organising continuous attractor networks and path integration: two-dimensional models of place cells. Netw Comput Neural Syst. 2002;13:429–46.
10. Stringer SM, Rolls ET, Trappenberg TP. Self-organising continuous attractor network models of hippocampal spatial view cells. Neurobiol Learn Mem. 2005;83:79–92.
11. Trappenberg TP. Fundamentals of computational neuroscience. Oxford: Oxford University Press; 2002.
12. Taylor JG. The race for consciousness. Bradford Books: Cambridge; 2001.
13. Taylor JG, Alavi FN. A global competitive neural network. Biol Cybern. 1995;72:233–48.
14. Wilson HR, Cowan JD. Excititory and inhibitory interactions in localized populations of model neurons. Biophys J. 1972;12:1–24.
15. Wilson HR, Cowan JD. A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. Kybernetik. 1973;13:55–80.
16. Taylor JG. Neural 'bubble' dynamics in two dimensions: foundations. Biol Cybern. 2000;80:393–409.
17. Doubrovinski K, Herrmann JM. Stability of localized patterns in neural fields. Neural Comput. 2009;21:1125–44.
18. Jancke D, Erlhagen W, Schoner G, Dinse H. Shorter latencies for motion trajectories than for flashes in population responses of a cat primary visual cortex. J Physiol. 2004;556(3):971–82.
19. Trappenberg TP, Dorris M, Munoz DP, Klein RM. A model of saccade initiation based on the competitive integration of exogenous and endogenous signals in the superior colliculus. J Cogn Neurosci. 2001;13:256–71.
20. Rolls ET, Stringer SM, Trappenberg TP. A unified model of spatial and episodic memory. Proc R Soc. 2002;269:1087–93.
21. Zhang K. Representation of spatial orientation by the intrinsic dynamics of the head direction cell ensemble: a theory. J Neurosci. 1996;16:2112–26.
22. Milford MJ, Wyeth GF, Prasser D. RatSLAM: a hippocampal model for simulataneous localization and mapping. In: Proceedings of the IEEE international conference on robotics and automation; 2004. p. 403–8.
23. Zibner SK, Faubel C, Iossifidis I, Schoner G. Dynamic neural fields as building blocks of a cortex-inspired architecture for robotic scene representation. IEEE Trans Auton Ment Dev. 2011;3:74–91.
24. Erlhagen W, Bicho E. The dynamic neural field approach to cognitive robotics. J Neural Eng. 2006;3:36–54.
25. McNaughton BL, Battaglia FP, Jensen O, Moser EI, Moser M. Path integration and the neural basis of the 'cognitive map'. Nat Rev Neurosci. 2006;7:663–78.
26. Samsonovich A, McNaughton BL. Path integration and cognitive mapping in a continous attractor neural network model. J Neurosci. 1997;17:5900–20.
27. Skaggs WE, Knierim JJ, Kudrimoti HS, McNaughton BL. 1995 A model of the neural basis of the rats sense of direction In: Tesauro G, Touretzky DS, Leen TK, editors. Advances in neural information processing systems, vol 7. Cambridge: MIT Press; 1995. p. 173–80.
28. Xie X, Hahnloser RH, Seung HS. Double-ring network model of the head-direction system. Phys Rev E Stat Nonlin Soft Matter Phys. 2002;66:041902.
29. Milford MJ, Wiles J, Wyeth GF. Solving navigational uncertainty using grid cells on robots. PLoS Comput Biol. 2010;6(11):e1000995. doi:10.1371/journal.pcbi.1000995.