Available online at www.sciencedirect.com
SCIENCE DIRECT®
Neurocomputing 64 (2005) 5–24
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Three-dimensional frames of references transformations using recurrent populations of neurons

Eric L. Sauser*, Aude G. Billard

Ecole Polytechnique Fédérale de Lausanne (EPFL), Swiss Federal Institute of Technology Lausanne, Autonomous Systems Laboratory, CH-1015 Lausanne, Switzerland

Available online 8 January 2005

Abstract

This work investigates whether population vector coding, a distributed computational paradigm, could be a principle mechanism for performing sensorimotor and frames of reference transformations. This paper presents a multilayer neural network that can perform arbitrary three-dimensional rotations and translations. We demonstrate, both formally and numerically, that the non-linearity of these transformations can be resolved thanks to the recurrent and concurrent activities of continuous populations of neurons.

Keywords: Population vector coding; Frames of references transformations; Gain modulation; Sensorimotor transformations

1. Introduction

When performing various sensorimotor tasks, including visuomotor and mental rotations, the brain faces the problem of transferring information across different frames of reference. For instance, visually guided movements require that visual information gathered in a retina-based frame of reference (FR) be transferred

*Corresponding author.
E-mail addresses: eric.sauser@epfl.ch (E.L. Sauser), aude.billard@epfl.ch (A.G. Billard).

0925-2312/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2004.11.019

sequentially into head-centered, body-centered, and finally hand-centered FR. Evidence that the brain encodes part of the visual and motor information in these different frames of reference is corroborated by a number of neurophysiological experiments [2,4,7]. It is, however, still unclear whether the FR transformations along the visuo-motor pathway follows such a simple, serial and linear decomposition.

The idea that sensory and motor information is represented in a distributed fashion through population of neurons has received considerable attention. Population vector coding was originally proposed as a plausible way to interpret the macroscopic effect of the joined activities of large sets of neurons [16]. It was later shown to be a computational paradigm shared by several areas of the nervous system, including proprioceptive receptors, such as muscle spindles [11], the motor cortex [7,16], and parts of the sensorimotor pathway, such as the posterior parietal cortex [4,15] and the superior temporal sulcus [2]. Population vector coding appears, thus, to be a common principle of brain organization, through which different neural populations interact and share information in the purpose of accomplishing tasks, by integrating multimodal information for distributed control across the whole body.

In this paper, we investigate how population vector coding can be used as a principle mechanism to accomplish FR transformations. The majority of related works [3,5,12] model the non-linear multiplicative response of the population by either assuming that the activation function of the neurons in the population produces a multiplicative response of the neuron's input (sigma-pi neurons) or that the synaptic strength between two neurons could be gated by a third part neuron. In contrast, following work by Salinas and Abbott [13], we derive the multiplicative property of the population output from the concurrent activity of a population of integrative neurons. Keeping the integrative properties of neurons is fundamental to remain in line with a biological account of the neural response. Prior studies considering frames of reference transformations using population codes with strictly additive synaptic inputs [6,10,13] have overlook a major effect. Indeed, the population vector, resulting from such transformation, exhibits a difference in its amplitude relative to its original value. To fill this gap, we investigate a method by which one can constrain this error within strict and acceptable bounds.

This paper is divided as follows: Section 2 describes the population vector coding paradigm and defines the notations. Then, the fundamental building block of our model will be presented. It consists in a two layers attractor network that is capable to produce a basic non-linear transformation. The next section will then explain how we applied this mechanism to perform frames of reference transformations, and finally, we will report our results on simulations analyzing the properties of the model.

2. Population vector coding

Let Ω be a continuous population of neurons where each unit participating in the population is characterized by its preferred direction $\vec{r}$. For a given population, the

preferred directions are assumed to be uniformly distributed along a N-dimensional subspace defined by

$$\Gamma(N) = \{\vec{r} \in \mathbb{R}^N | \|\vec{r}\| = 1\} \tag{1}$$

that corresponds to a unitary hypersphere of dimension N. In the rest of this paper, we will consider populations encoding vectors in either two- (2D) or three-dimensional (3D) spaces, i.e. $N \in \{2, 3\}$. Moreover, unless it is necessary to specify a value for N, we will omit it.

Let $u_{\vec{r}}$ be the neuron's membrane potential with preferred direction $\vec{r}$, and $f(u_{\vec{r}})$ its firing activity, where f is a non-linear function equal to $f(x) = [x]^+$. The neuron's firing activity is, thus, equal to its membrane potential, if it is positive, or to zero otherwise. Its dynamics follows the general form of a leaky integrator neuron, that is

$$\tau \dot{u}_{\vec{r}}(t) = -u_{\vec{r}}(t) + x_{\vec{r}}, \tag{2}$$

where τ is the neuron time constant and $x_{\vec{r}}$ is the synaptic input. The firing activity of each neuron is modeled by a cosine-tuning curve centered on the neuron's preferred direction:

$$u_{\vec{r}}(t) = \beta(t)(\vec{r} \cdot \vec{r}_{\mathrm{p}}(t)) + \alpha(t), \tag{3}$$

where $\beta(t) \geqslant 0$ is the amplitude of the cosine shape response of the population, $\vec{r}_{\mathrm{p}}(t)$ the direction encoded by the population and $\alpha(t)$ the baseline potential. The external synaptic input is defined by

$$x_{\vec{r}}(t) = \vec{r} \cdot \vec{v}(t) + h(t) = \beta_v(t)(\vec{r} \cdot \vec{r}_v(t)) + h(t), \tag{4}$$

where h is an external homogeneous input, and $\vec{v}$ is the vector encoded by the population. β_v is the norm of $\vec{v}$ and $\vec{r}_v$ the normalized direction given by $\vec{v}$. Thus, the neuron's activity is maximal when the direction coded in the external input is the closest to the neuron's preferred direction. To simplify the notation in the rest of this paper, we will omit the time variable from our equations.

Rewriting Eq. (2) using (3) and (4), we see that the neuron's potential $u_{\vec{r}}$ converges toward its synaptic input $x_{\vec{r}}$. The response of the whole population, illustrated in

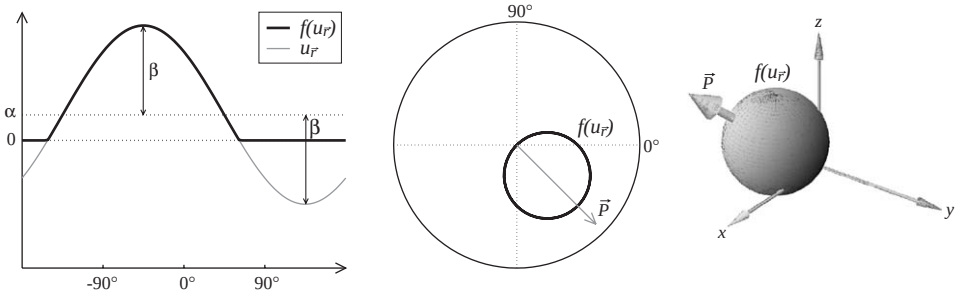

Fig. 1. (Left) Activity profile of a population with preferred directions distributed along a 2D space, such that $\vec{r} = (\cos(\theta), \sin(\theta))$. (Middle) Same as (left), but using a polar representation. The population vector $\vec{P}$ encoded by this population is superimposed on this figure. (Right) Activity profile of a 3D population represented in spherical coordinates with its population vector.

Fig. 1, called the *population vector* $\vec{P}$, is given by:

$$\vec{P} = \frac{1}{\kappa(\alpha,\beta)} \oint_{\Gamma} f(u_{\vec{r}})\vec{r}\,\mathrm{d}\vec{r}, \quad \text{where } \kappa(\alpha,\beta) > 0, \tag{5}$$

where $\kappa(\alpha,\beta)$ is a normalization factor that allows $\vec{P}$ to converge toward the input $\vec{v}$ (see Eq. (4)). The values taken by $\kappa(\alpha,\beta)$ for $N = \{2,3\}$ are summarized in Table 1. When $\beta > |\alpha|$, for $\alpha = 0$, or $\alpha/\beta = \eta$, κ becomes a constant normalization factor. η is a constant value in the interval $]0,1[$. When $\alpha = 0$, $\kappa(0,\beta) = \kappa_0 = \{\pi/2,\ 2\pi/3\}$, for $N \in \{2,3\}$ respectively. Finally, when $\kappa(\alpha,\beta) = 0$, it means that the population vector cannot be recovered because the baseline α is inhibitory and stronger than the vectorial input with amplitude β, what produces a constant and zero population activity profile, i.e. the population is silent.

In order to deal with several populations simultaneously, we add a new index to our variables, corresponding to the name of the population. For instance, the potential of a neuron with preferred direction $\vec{r}$ in a population Ω_{v} is $u^{\mathrm{v}}_{\vec{r}}$.

Let us now consider two populations Ω_{v} and $\Omega_{\mathrm{v}'}$, where the former projects its activity to the latter. Let us set the synaptic weights $w^{\mathrm{v}\to\mathrm{v}'}_{\vec{r}\to\vec{r}'}$ between neuron $u^{\mathrm{v}}_{\vec{r}}$ in Ω_{v} and neuron $u^{\mathrm{v}'}_{\vec{r}}$ in $\Omega_{\mathrm{v}'}$ by

$$w^{\mathrm{v}\to\mathrm{v}'}_{\vec{r}\to\vec{r}'} = \frac{1}{\kappa_0}(\vec{r}\cdot\vec{r}'). \tag{6}$$

The synaptic input to $u^{\mathrm{v}'}_{\vec{r}}$ is, then:

$$x^{\mathrm{v}'}_{\vec{r}'} = \oint_{\Gamma^{\mathrm{v}}} w^{\mathrm{v}\to\mathrm{v}'}_{\vec{r}\to\vec{r}'} f(u^{\mathrm{v}}_{\vec{r}})\,\mathrm{d}\vec{r}. \tag{7}$$

Let us now consider the case when the baseline of the source population is set to zero, i.e. $\alpha^{\mathrm{v}} = 0$. All neurons potential will tend toward $u^{\mathrm{v}}_{\vec{r}} = \vec{r}\cdot\vec{v}$. In that case, the synaptic input is equal to

$$x^{\mathrm{v}'}_{\vec{r}'} = \vec{r}'\cdot\vec{v}. \tag{8}$$

This implies that the source population Ω_{v} coding for $\vec{v}$ in the space defined by the set $\{\vec{r}\} = \Gamma^{\mathrm{v}}$ of its preferred directions, has projected $\vec{v}$ into the target population space defined by $\{\vec{r}'\} = \Gamma^{\mathrm{v}'}$. For instance, if both sets are equivalents, Eq. (7) becomes the identity. However, if Γ^{v} is a 3D subspace and $\Gamma^{\mathrm{v}'}$ is 2D, (7) becomes a projection of $\vec{v}$

Table 1
Expression of $\kappa(\alpha,\beta)$ under different conditions

	$N = 2$	$N = 3$
$0 < \beta \leqslant -\alpha$	0	0
$\beta > \lvert\alpha\rvert$	$\left(\frac{\alpha}{\beta}\sqrt{1-\left(\frac{\alpha}{\beta}\right)^2} + \operatorname{acos}\left(-\frac{\alpha}{\beta}\right)\right)$	$\frac{\pi}{3}\left(2 + 3\,\frac{\alpha}{\beta} - \left(\frac{\alpha}{\beta}\right)^3\right)$
$0 < \beta \leqslant \alpha$	π	$\frac{4\pi}{3}$

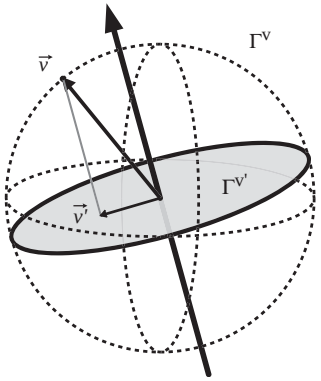

Fig. 2. Result of a projection from a 3D population with preferred direction distributed along Γ^{v} (dotted line) to a 2D one distributed along $\Gamma^{v'}$ (filled line).

on the plane defined by $\Gamma^{v'}$, as illustrated on Fig. 2. In the rest of this paper, each time, a synaptic input of the form described in Eq. (8) is mentioned, we will implicitly consider that it is the result of a projection from another population, as defined by (7).

3. Attractor network model

Let us now consider an attractor network [13,18] made of a fully connected population of neurons whose dynamics is governed by

$$\begin{aligned} \tau \dot{u}_{\vec{r}} &= -u_{\vec{r}} + \oint_{\Gamma} w_{\vec{r}' \to \vec{r}} f(u_{\vec{r}'})\, \mathrm{d}\vec{r}' + x_{\vec{r}}, \\ w_{\vec{r}' \to \vec{r}} &= \gamma(\eta)(\vec{r}' \cdot \vec{r}), \end{aligned} \tag{9}$$

where $w_{\vec{r}' \to \vec{r}}$ are the lateral weights that exhibit symmetric, rotation invariant, and center surround excitation inhibition characteristics, $x_{\vec{r}}$ is the sum over all external synaptic inputs, and $\gamma(\eta)$ is a scaling factor depending on the network parameter $\eta \in]0,1[$ that controls the influence of the lateral weights. We will see below how to define precisely this factor. Let us replace $u_{\vec{r}}$ in Eq. (9) using (3). If we put the left-hand side of this equation to zero, this gives us

$$\forall \vec{r} \in \Gamma^{N}, \quad -\alpha^{\star} + h - \beta^{\star}(\vec{r} \cdot \vec{r}_{\mathrm{p}}^{\star})(1 - \gamma(\eta)\kappa(\alpha^{\star}, \beta^{\star})) + \beta_{v}(\vec{r} \cdot \vec{r}_{v}) = 0, \tag{10}$$

where $\alpha^{\star}$, $\beta^{\star}$ and $\vec{r}_{\mathrm{p}}^{\star}$ are the values of each time-dependent variable of the population after convergence. Since (10) is true $\forall \vec{r} \in \Gamma$, we can separate each of these variables and resolve separately the following three equations:

$$\begin{aligned} \vec{r}_{v} &= \vec{r}_{\mathrm{p}}^{\star}, \\ 0 &= -\alpha^{\star} + h, \\ 0 &= -\beta^{\star}(1 - \gamma(\eta)\kappa(\alpha^{\star}, \beta^{\star})) + \beta_{v}. \end{aligned} \tag{11}$$

From this, we have, after convergence, $\vec{r}_{\mathrm{p}}^{\star} = \vec{r}_v$ and $\alpha^{\star} = h$. Since Eq. (11) does not have an analytical solution, we can only approximate the value taken by $\beta^{\star}$.

Let us first consider the case in which the external synaptic input is constant across the whole network, that is when $h > 0$ and $\beta_v = 0$ in Eq. (4). By setting $\gamma(\eta) = \kappa(\eta, 1)^{-1}$ for $\eta \in]0, 1[$ in Eq. (11), we obtain

$$0 = -\beta^{\star}\left(1 - \frac{\kappa(h, \beta^{\star})}{\kappa(\eta, 1)}\right). \tag{12}$$

One solution of this equation is $\beta^{\star} = 0$. This solution is, however, unstable. It also has the stable solution $\kappa(h, \beta^{\star}) = \kappa(\eta, 1)$. Recall that $\kappa(\alpha, \beta)$ is constant and equal to $\kappa(\eta, 1)$, when the ratio $\alpha/\beta = \eta$ is constant. Since $\kappa(h, \beta^{\star})$ follows h, after convergence, we get $\beta^{\star} = (1/\eta)h$. As an effect, the population produces a multiplicative response proportional to h, as illustrated in Fig. 3. The membrane potential of the neuron has the following stable attractor state:

$$u_{\vec{r}}^{\star} = \begin{cases} h(1 + \frac{1}{\eta}(\vec{r} \cdot \vec{r}_{\mathrm{p}})), & h > 0, \\ h, & h \leqslant 0, \end{cases} \tag{13}$$

where $\vec{r}_{\mathrm{p}}$ is strictly dependent on the initial state of the population. Similarly to [14], when the network receives a constant excitatory global activation, it will converge to an *active state*, in which the amplitude of the population vector is amplified proportionally to the external homogeneous activity. Conversely, when the external activity is inhibitory, the network population coding will be turned *off*, each neuron becoming silent. This mechanism is known as *gain modulation* [13,14].

In the opposite case, when the network receives a vectorial input, that is when $h = 0$ and $\beta_v \vec{r}_v = \vec{v} \neq \vec{0}$ in Eq. (4), we can rewrite (11) under these considerations. It is straightforward to find that the amplitude $\beta^{\star}$ after convergence

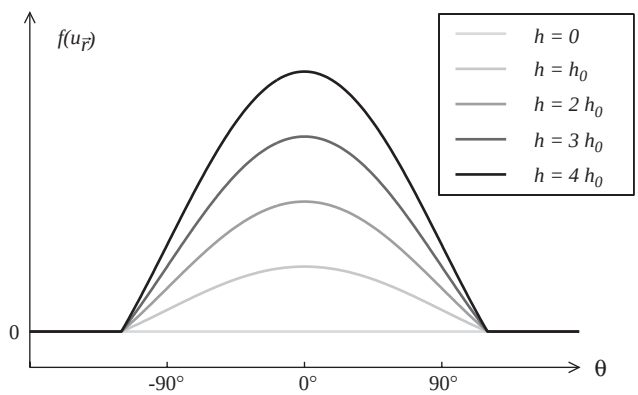

Fig. 3. Activity profile of a population of neurons, sorted relative to their preferred direction $\vec{r}(\theta)$ (see Fig. 1), which are under the influence of an homogeneous input h. By considering the global population activity, the effect of this modulatory input is a multiplicative response.

is equal to:

$$\beta^\star = \frac{1}{\chi(\eta)}\,\beta_v, \quad \text{where } \chi(\eta) = 1 - \gamma(\eta)\kappa_0, \tag{14}$$

what gives for the potential $u^\star_{\vec{r}}$

$$u^\star_{\vec{r}} = \frac{1}{\chi(\eta)}\,\beta_v(\vec{r}\cdot\vec{r}_v) = \frac{1}{\chi(\eta)}\,(\vec{r}\cdot\vec{v}). \tag{15}$$

This result means that the network tends to match its vectorial input $\vec{v}$. In order to consider now the two types of inputs discussed before, we have assumed a linear approximation of previous solutions (see Eqs. (13) and (15)) that gives

$$u^\star_{\vec{r}} \approx h + \left(\frac{1}{\eta}\,h + \frac{1}{\chi(\eta)}\,\beta_v\right)(\vec{r}\cdot\vec{r}_v). \tag{16}$$

3.1. A two layers neural network

The main contribution of this study lies in the creation of a two layers neural network that is able to produce a non-linear composition of its inputs. Indeed, frames of references transformations, described in the next section, will be based on such building block. As illustrated in Fig. 4, the first layer of our system consists of the attractor network described above. We can see that the approximation of its activity profile (Eq. (16)) reflects both the vectorial and constant inputs, plus a modulatory term in which we are interested. Thus, in order to strictly keep this multiplicative term, we build another population Ω_o (o for output) without lateral weights. It receives projections from the recurrent population using one to one synapses, and inhibitory inputs corresponding to the negative of the vectorial and constant inputs of the recurrent population with appropriate scaling, such that

$$x^o_{\vec{r}} = \eta\left(f(u_{\vec{r}}) - h - f\left(\frac{1}{\chi(\eta)}\,\beta_v(\vec{r}\cdot\vec{r}_v)\right)\right).$$

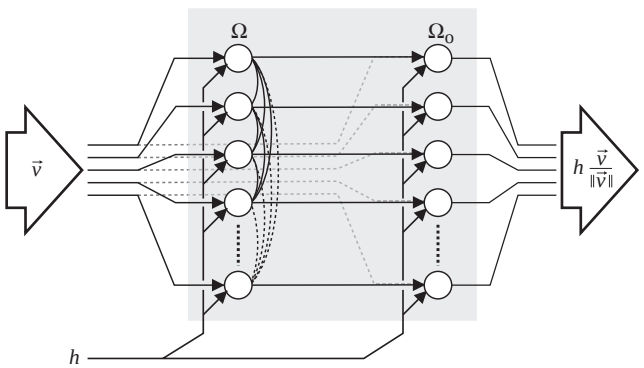

Fig. 4. The two layers neural network producing a non-linear composition of its inputs.

Considering that the neurons of Ω_o support an immediate integration mechanism such that $u^o_{\vec{r}} = x^o_{\vec{r}}$, we obtain, after a substitution in the previous equation using (16), that the firing rate is equal to

$$f(u^o_{\vec{r}}) = f\left(\eta\left(f(u_{\vec{r}}) - h - f\left(\frac{1}{\chi(\eta)}\,\beta_v(\vec{r}\cdot\vec{r}_v)\right)\right)\right) \approx \begin{cases} h(\vec{r}\cdot\vec{r}_v), & \vec{r}\cdot\vec{r}_v > 0,\ h > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

This result shows that this network is capable to encode independently two separate quantities, that are the direction $\vec{r}_v$ and the amplitude h, regardless of the intensity of the directional input β_v. In vectorial terms, this means that given a vector $\vec{v}$ and a scalar h, the output population vector will tend toward $h(\vec{v}/\|\vec{v}\|)$. Therefore, this model can be used to form a vectorial basis, but also, as will be explained in the next paragraph, to perform non-linear frames of reference transformations.

4. Frames of references transformations

As illustrated in Fig. 5 (left), the projection of a vector $\vec{v}$ from a N-dimensional referential R to a N-dimensional referential R' can be decomposed into one translation from the origin O to the origin O' and N rotations, performed serially on $\{\phi_1, \ldots, \phi_N\}$. We will show, in the next paragraphs, how to perform these operations using several populations connected serially.

4.1. Translations

Let $\vec{v}$ be a vector in referential R represented by a population Ω_v, and $\vec{v}'$ its projection in the referential R' represented by $\Omega_{v'}$. Assuming that $\vec{T}$ is the vector across the origins of both referentials, represented by Ω_T, we have $\vec{v}' = \vec{v} + \vec{T}$. To perform the translation, we consider that the population $\Omega_{v'}$ receives as inputs $x^{v'}_{\vec{r}}$,

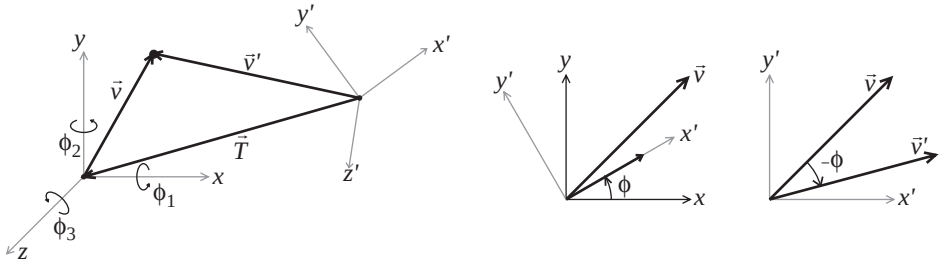

Fig. 5. (Left) 3D frames of reference transformations can be decomposed into three rotations and one translation at the origin. (Right) Frames of reference transformations between planar referentials with superposed origins.

the result of the summation of the synaptic projections (see Eq. (7)) coming from Ω_{v} and Ω_{T}. Using (8), it gives

$$\begin{aligned} x_{\vec{r}}^{\mathrm{v}'} &= \oint_{\Gamma^{\mathrm{v}}} w_{\vec{r}'\to\vec{r}}^{\mathrm{v}\to\mathrm{v}'} f(u_{\vec{r}'}^{\mathrm{v}})\,\mathrm{d}\vec{r}' + \oint_{\Gamma^{\mathrm{T}}} w_{\vec{r}'\to\vec{r}}^{\mathrm{T}\to\mathrm{v}'} f(u_{\vec{r}'}^{\mathrm{T}})\,\mathrm{d}\vec{r}' \\ &= (\vec{r}\cdot\vec{v}) + (\vec{r}\cdot\vec{T}) \\ &= \vec{r}\cdot(\vec{v}+\vec{T}) \\ &= \vec{r}\cdot\vec{v}'. \end{aligned} \tag{18}$$

By (5), $\Omega_{\mathrm{v}'}$ represents the vector $\vec{v}'$, that is the sum of two other vectors. Indeed, as illustrated in Fig. 6, translations are by nature a purely linear transformation that can, in a straightforward manner, be applied into such kind of distributed neural representation.

4.2. Rotations

Let us first consider the case of a planar rotation where the vector $\vec{v}$ in referential R is rotated by an angle $-\phi$ to project onto $\vec{v}'$. $\vec{v}$ and $\vec{v}'$ are encoded in the populations Ω_{v} and $\Omega_{\mathrm{v}'}$, respectively. ϕ corresponds to the angle between the two planar referentials R and R' (see Fig. 5 (right)), with superposed origins $O = O'$. The amount of rotation ϕ is represented in a 2D population Ω_ϕ that code for a vector in the direction ϕ with an arbitrary strictly positive amplitude β_ϕ. Its activity profile is then described by

$$u_{\vec{r}}^{\phi} = \beta_\phi(\vec{r}\cdot\vec{r}_\phi) = \vec{r}\cdot\vec{v}_\phi, \tag{19}$$

where $\vec{r}_\phi = (\cos(\phi), \sin(\phi))$.

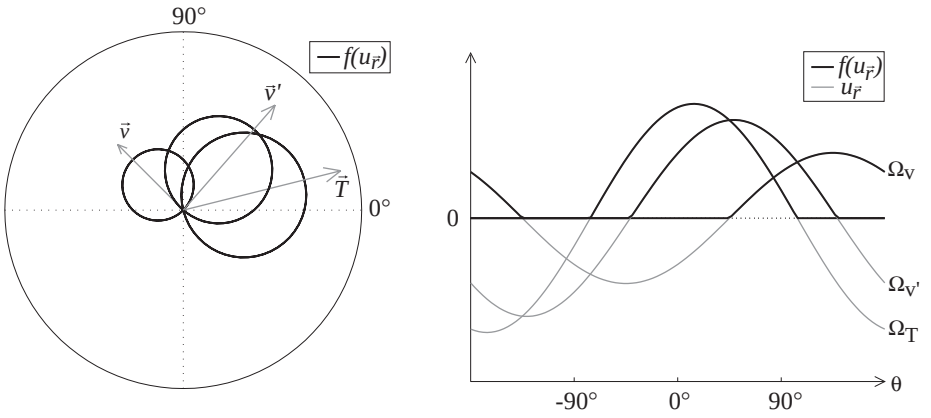

Fig. 6. (Left) Polar representation of the activity of the three populations and their corresponding population vectors that are involved in the translation such that $\vec{v}' = \vec{v} + \vec{T}$. (Right) Same as (left), but using a linear representation. Same notation as in Fig. 1 is used.

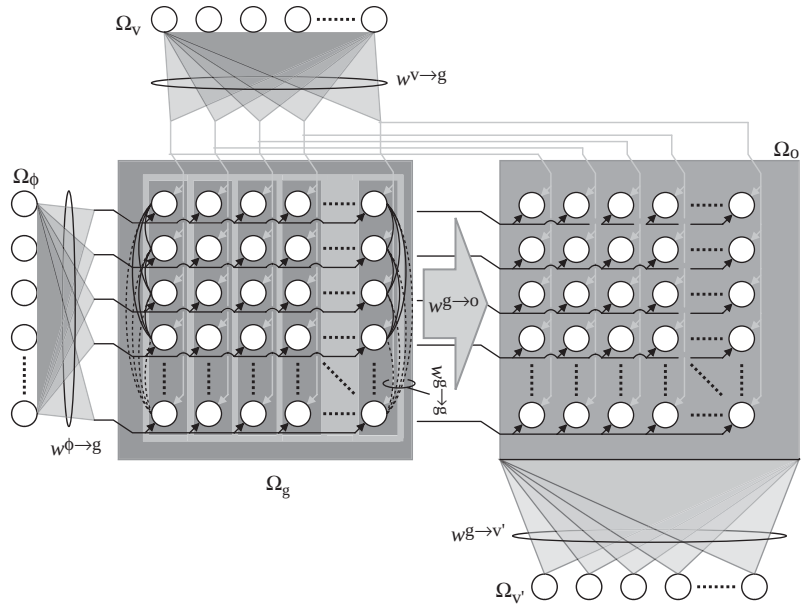

Fig. 7. Gain field architecture.

In contrast to the translation operation, a rotation is a non-linear transformation. In order to resolve the non-linearity, we need to define an intermediary population: *the gain field* Ω_g. Its architecture and connectivity are shown in Fig. 7. It consists in an assembly of building blocks that were defined in Section 3.1. As can be seen, the left array of neurons is a 2D planar population where each column (shaded areas surrounded by rectangles) denoted by a preferred direction $\vec{s}$, is composed of a recurrent population whose dynamics follows Eq. (9). Each of these neurons projects through a one-to-one connection to the right array (population Ω_o) that has the purpose to rectify their activity (see Eq. (17)). The external inputs incoming from Ω_ϕ and Ω_v are separately applied to each dimension of the gain field, $\vec{r}$ and $\vec{s}$, respectively. Hence, by (8), the input for each neuron $\vec{r}$ of each layer $\vec{s}$ in Ω_g is defined by

$$\begin{aligned} x^{g}_{(\vec{r},\vec{s})} &= \oint_{\Gamma^\phi} w^{\phi\to g}_{\vec{r}'\to\vec{r}} f(u^{\phi}_{\vec{r}'})\,\mathrm{d}\vec{r}' + \oint_{\Gamma^v} w^{v\to g}_{\vec{r}'\to\vec{s}} f(u^{v}_{\vec{r}'})\,\mathrm{d}\vec{r}' \\ &= (\vec{r}\cdot\vec{v}_\phi) + (\vec{s}\cdot\vec{v}) \\ &= \beta_\phi(\vec{r}\cdot\vec{r}_\phi) + \beta_v(\vec{s}\cdot\vec{r}_v). \end{aligned} \tag{20}$$

If we substitute (20) in Eq. (17), we obtain an output firing activity converging toward

$$f(u^{g}_{(\vec{r},\vec{s})}) \approx \beta_v(\vec{r}\cdot\vec{r}_\phi)(\vec{s}\cdot\vec{r}_v). \tag{21}$$

Similarly to [3,12], we encode the result of the rotation into the synaptic projections from the gain field to the population $\Omega_{v'}$. Using an analogy to the complex division, we express the rotation of a unitary vector $\vec{r}$ by $-\theta_s$, the angle given by the orientation of

the unitary vector $\vec{s} = (\cos\theta_s, \sin\theta_s)$ as $\vec{r}/\vec{s}$. The synaptic weights across the gain field to the population $\Omega_{v'}$ is then:

$$w^{g\to v'}_{(\vec{r},\vec{s})\to\vec{r}'} = \frac{1}{\kappa_0^2}(\vec{r}' \cdot (\vec{r}/\vec{s})). \tag{22}$$

Using (22), the input $x^{v'}_{\vec{r}'}$ to the destination population $\Omega_{v'}$ is

$$\begin{aligned} x^{v'}_{\vec{r}'} &= \oint_{\Gamma^g} w^{g\to v'}_{(\vec{r},\vec{s})\to\vec{r}'} f(u^g_{(\vec{r},\vec{s})})\,\mathrm{d}\vec{r}\,\mathrm{d}\vec{s} \\ &\approx (\vec{r}' \cdot (\vec{v}/\vec{v}_\phi)). \end{aligned} \tag{23}$$

Finally, using the population vector defined in Eq. (5), we see that $\Omega_{v'}$ encodes $\vec{v}'$, which corresponds to $\vec{v}$ rotated by $-\phi$. This mechanism is illustrated on Fig. 8.

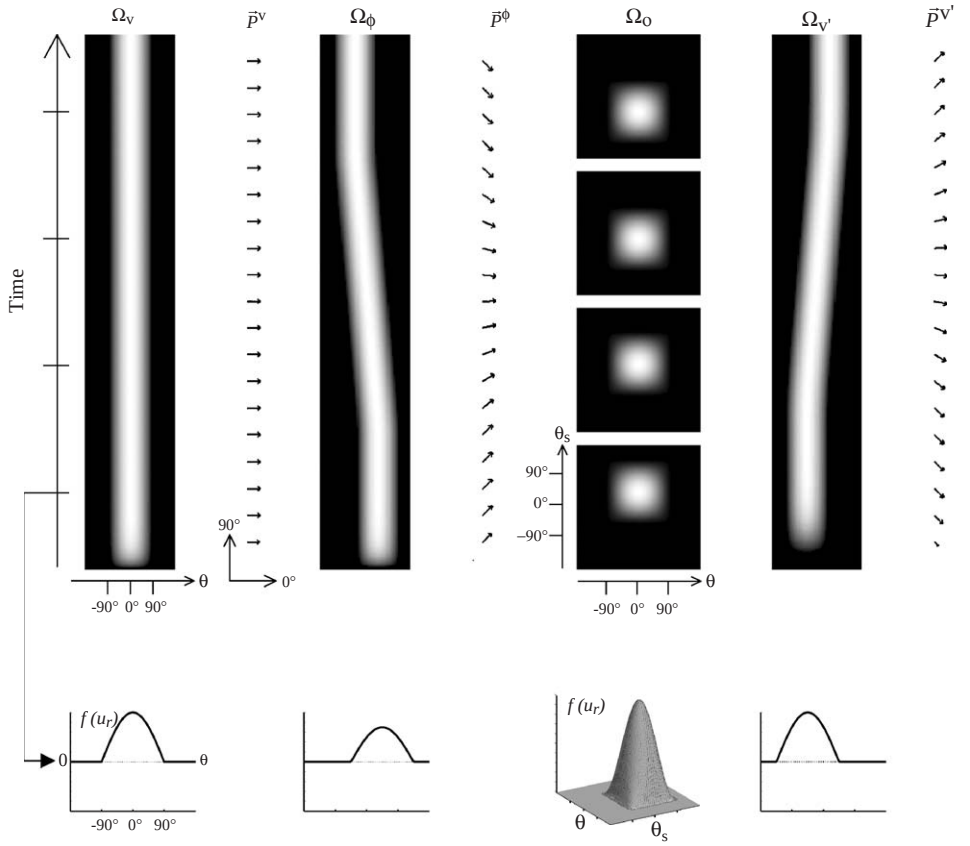

Fig. 8. (Top) Evolution over time of the activity profiles of the involved populations while performing a constantly varying rotation. The activity of the output neurons in the gain field Ω_o are shown for several time steps denoted by the marks on the time axis. On the right-hand side to each activity plot, except for Ω_o, the corresponding population vector is shown. (Bottom) Snapshot of the activities of the same populations at the time step given by the left black arrow on the time axis.

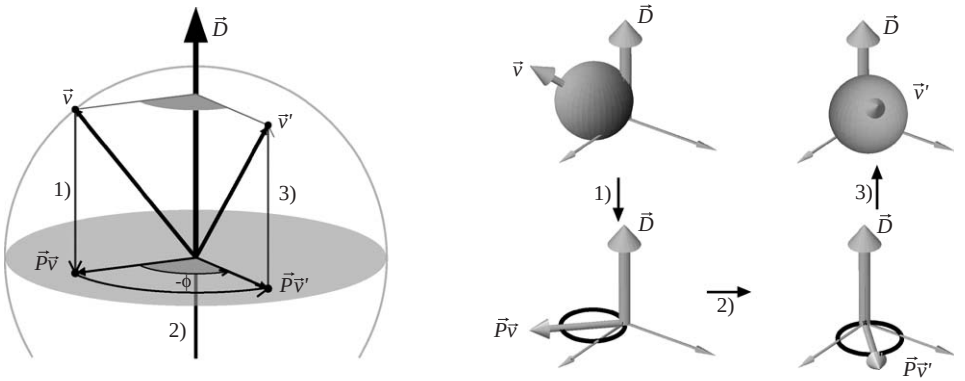

Fig. 9. (Left) A rotation in 3D space can be decomposed into three sub-transformations (see main text). (Right) Illustration in spherical coordinates of the populations activities during this transformation.

On top of the figure, we show the evolution of the activity of the involved populations while performing a constantly varying rotation. We can see that the activity profile of the gain field output is symmetric and that the peak is located at the intersection of the directions currently encoded by both source populations. On the bottom of this figure, we can see a snapshot of the activities at a given time. It shows that the amplitude of the cosine shape of the populations Ω_{v} and $\Omega_{\mathrm{v}'}$ are equal, meaning that the amplitude is preserved through the non-linear transformation.

4.3. Extension to 3D rotations

In order to perform rotations around an axis $\vec{D}, \|\vec{D}\| = 1$ in a 3D space, we decompose the rotation into three sub-transformations, (see Fig. 9). Indeed, a rotation of vector $\vec{v}$ in 3D space can be seen as a succession of three steps: (1) a projection of $\vec{v}$ on the plane perpendicular to the rotation axis $\vec{D}$, (2) a rotation of an angle $-\phi$ around $\vec{D}$ and (3) the restoration of the component parallel to $\vec{D}$ lost during the projection. This process results in a vector $\vec{v}'$ that corresponds to $\vec{v}$ rotated by an angle $-\phi$ around axis $\vec{D}$.

As mentioned earlier, a projection on a plane perpendicular to $\vec{D}$ from a 3D population encoding $\vec{v}$ to a 2D one can be realized using synaptic weights defined by Eq. (6), if and only if the set of preferred directions $\{\vec{r}'\}$ are defined on that projection plane. We can then rewrite Eqs. (20)–(23) by replacing $\vec{v}$ by $\vec{p}_{\vec{v}}$, with $\vec{p}_{\vec{v}}$ corresponding to the projection of $\vec{v}$ on the plane perpendicular to $\vec{D}$. As in Eq. (23), the inputs to the population $\Omega_{\mathrm{v}'}$ then becomes

$$x^{\mathrm{v}'}_{\vec{r}'} \approx (\vec{r}' \cdot \ (\vec{p}_{\vec{v}}/\vec{v}_{\phi})). \tag{24}$$

Up to here, this means that this population encodes the projection of $\vec{v}$ rotated by an angle $-\phi$. The last step is to recover the component of $\vec{v}$ lost during the projection

that is equal to $(\vec{v}\cdot\vec{D})\vec{D}$. As in Eq. (18), we add new synaptic links[1] from Ω_{v} to $\Omega_{\mathrm{v}'}$, defined as

$$w^{\mathrm{v}\to\mathrm{v}'}_{\vec{r}\to\vec{r}'} = \frac{1}{\kappa_0}(\vec{r}'\cdot(\vec{r}\cdot\vec{D})\vec{D}). \tag{25}$$

Using (24) this gives the total inputs for $\Omega_{\mathrm{v}'}$ that is

$$\begin{aligned} x^{\mathrm{v}'}_{\vec{r}'} &= \oint_{\Gamma^{\mathrm{g}}} w^{\mathrm{g}\to\mathrm{v}'}_{(\vec{r},\vec{s})\to\vec{r}'} f(u^{\mathrm{g}}_{(\vec{r},\vec{s})})\,\mathrm{d}\vec{r}\,\mathrm{d}\vec{s} + \oint_{\Gamma^{\mathrm{v}}} w^{\mathrm{v}\to\mathrm{v}'}_{\vec{r}\to\vec{r}'} f(u^{\mathrm{v}}_{\vec{r}})\,\mathrm{d}\vec{r} \\ &\approx (\vec{r}'\cdot\ (\vec{p}_{\vec{v}}/\vec{v}_{\phi})) + (\vec{r}'\cdot(\vec{v}\cdot\vec{D})\vec{D}) \\ &\approx (\vec{r}'\cdot(\vec{v}/\vec{v}_{\phi})) \\ &\approx (\vec{r}'\cdot\vec{v}'). \end{aligned} \tag{26}$$

Again, by (5), the population vector of $\Omega_{\mathrm{v}'}$ implies that this population encodes $\vec{v}'$ that corresponds to $\vec{v}$ rotated by $-\phi$.

5. Simulation results

5.1. Visuomotor transformations for reaching

To illustrate the rotation mechanism, let us consider a simple target reaching task, where the target $\vec{v}$ is perceived by the visual system and encoded in a population Ω_{v} in head-centered coordinates. We suppose here that the eyes are fixed in their orbit. This population is encoding both the direction and the distance to the target. Let us suppose, then, that the angle between the body and the head is encoded in a population Ω_{ϕ} receiving proprioceptive information transmitted by the corresponding muscles receptors. In order to plan a movement to reach the target, the target location $\vec{v}$ in head-centered coordinates must be transferred into a vector $\vec{v}'$ encoded in a population $\Omega_{\mathrm{v}'}$ with respect to a body-centered FR. Fig. 10 shows the firing activity over time of the three populations during the following three scenarios: In the first scenario, the head rotates to follow closely the motion of a target that moves from left to right (from -45° to 45°). In the second scenario, the head remains fixed with respect to the body, and only the target moves. Finally, in the last scenario, the head rotates with respect to the body, while the body and the target remain static. We can see that $\vec{v}$ is correctly rotated according to ϕ. There is, however, a delay in the representation of the target in body centered coordinates, which is due to the time required for the network to propagate its information.

5.2. Uniform distribution of the preferred direction in 3D space

In order to generate an uniform distribution of the preferred directions across the population in our simulations, we used an iterative algorithm, inspired by the

[1]Not shown in Fig. 7.

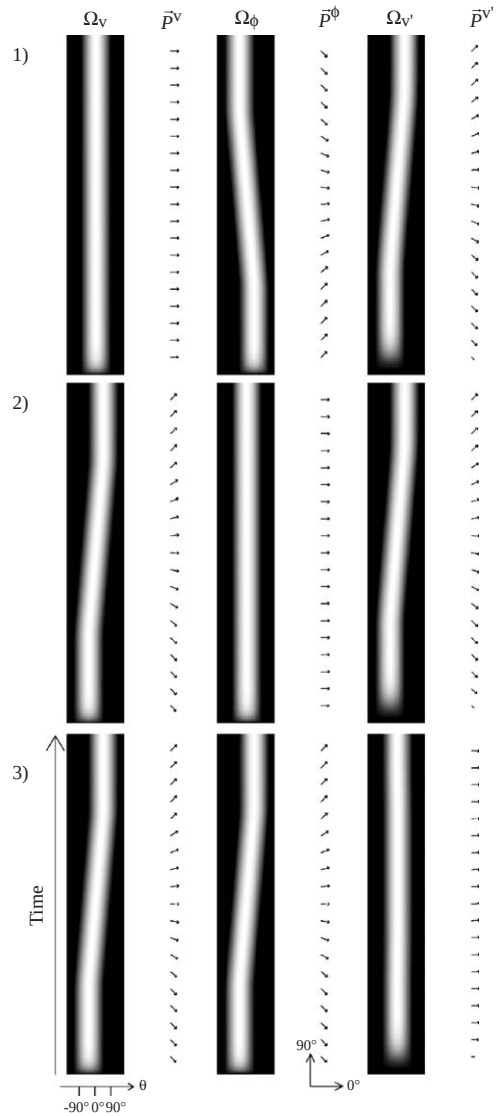

Fig. 10. Results of simulations of the network that realizes a 2D rotation. Each row corresponds to a different scenario (see the main text for explanations). Each column represents the activity of the populations Ω_v, Ω_ϕ and $\Omega_{v'}$ over time. On the right-hand side to each activity plot, the corresponding population vector is drawn.

self-organizing map [8]. In contrast to the 2D case, where a regular distribution of preferred directions is straightforward to generate, it is not possible to build a set of uniform unitary vector on a sphere, except for a finite set of number of points, that correspond to the corners of a regular polyhedra. For a population constituted of

N neurons, we generate randomly, during the initialization phase, N vectors $\vec{r}_i$, $i \in \{1 \dots N\}$ on the unit sphere by choosing two random values $\lambda_1 \in [0, 2\pi[$ and $\lambda_2 \in [-1, 1[$, such that

$$\vec{r}_i = \begin{pmatrix} \sqrt{1-(\lambda_2)^2}\cos(\lambda_1) \\ \sqrt{1-(\lambda_2)^2}\sin(\lambda_1) \\ \lambda_2 \end{pmatrix}. \tag{27}$$

This vector generation method guarantees a statistical uniform distribution on the unit sphere that does not lead to a concentration of points at the poles [9]. We then iterate for a fixed number of steps. At each time step, using the same technique as in Eq. (27), we generate a training random input vector $\vec{p}$ and update each $\vec{r}_i$ such that

$$\vec{r}_i(t+1) = \begin{cases} \vec{r}_i(t) + \delta(t)(\vec{p} - \vec{r}_i(t)) & \text{if } i = i^\star, \\ \vec{r}_i(t) & \text{if } i \neq i^\star, \end{cases} \tag{28}$$

where $\delta(t) \in]0, 1]$, a learning rate, decreases exponentially over training time, and $i^\star$ corresponds to the index of the closest preferred direction to $\vec{p}$ such that

$$\forall i, \quad \vec{r}_{i^\star}(t) \cdot \vec{p} \geqslant \vec{r}_i(t) \cdot \vec{p}.$$

This mechanism lets the set $\{\vec{r}_i\}$ of preferred directions converge toward an uniform distribution of its input space [8], which, by Eq. (27), is uniform over the unit sphere. To quantify the uniformity of the resulting distribution, we compute the error $\varepsilon = \|(1/N)\sum_i^N \vec{r}_i\|$. Fig. 11 (left) shows the evolution of ε for several trials, while Fig. 11 (right) shows an example of a resulting set of preferred directions.

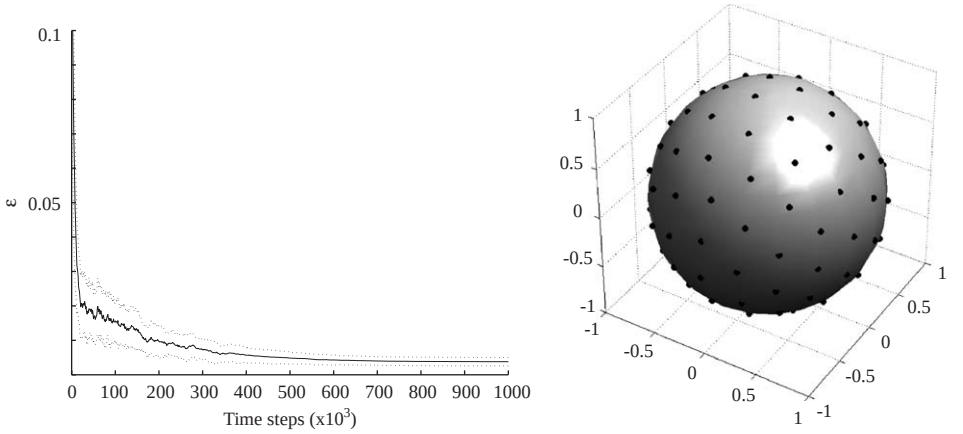

Fig. 11. Results produced by the iterative algorithm that generates a quasi-uniform distribution of points on a sphere. (Left) Evolution of the mean (filled line) and standard error (dotted line) of uniformity error ε along training time for the generation of 100 populations of 100 neurons. (Right) Hundred points distributed on a sphere using this algorithm.

5.3. Approximation errors

In addition to the errors that appear by discretizing continuous equations, the approximation we made in our mathematical development (see Eq. (16)) are also a source of systematical errors between the theoretical result vector $\vec{v}'^{\star}$, computed with classical rotation equations, and the result $\vec{v}'$ produced by our network. To quantify them, we define E_β, the error on the amplitude, and E_θ, the angular error on the direction, by

$$E_\beta(\vec{v}', \vec{v}'^{\star}) = \frac{\left| \|\vec{v}'\| - \|\vec{v}'^{\star}\| \right|}{\|\vec{v}'^{\star}\|},$$

$$E_\theta(\vec{v}', \vec{v}'^{\star}) = \operatorname{acos}\left(\frac{\vec{v}' \cdot \vec{v}'^{\star}}{\|\vec{v}'\| \, \|\vec{v}'^{\star}\|} \right),$$

that correspond to the relative difference between their norms, and to the angle they form, respectively. Fig. 12 (left) plots the error E_β for a planar rotation that uses populations having an uniform distribution of their preferred directions, and with different network parameters values. E_θ was not represented, because in this case, it can be neglected. This figure shows that the bigger η, the bigger is the error. Similarly, an increase in the amplitude β_ϕ of the population expressing the amount of rotation, generates an augmentation of the error. This result favors the use of small values for η.

In the 3D case, we applied the previously mentioned algorithm to generate several 3D populations of different sizes. We then tested the ability of these populations to faithfully represent their inputs, by comparing a large number of vectorial inputs with the resulting population vector, using the error measurements E_β and E_θ. The results of these simulations, summarized in Fig. 13, show that the size of the network considerably influences its precision. Moreover, in contrast to the planar rotation experiment mentioned above, the error decreases for increasing values for the system parameter η. This can be explained by looking at Eq. (9) describing the dynamics of the network. Indeed the factor $\gamma(\eta)$, that defines the influence of the lateral weights on the network activity decreases as η grows. Thus, for a small η, the imperfect distribution of the preferred directions along the population leads to a larger drift. In order to illustrate that this drift is mainly caused by the lateral weights, we applied an homogeneous input h to a population while keeping the vectorial contribution null, i.e. $\vec{v} = 0$ (see Eq. (4)). Using (3), we initialized the membrane potential of each neuron to a random initial direction $\vec{r}_p$. We, then, recorded the evolution of the direction coded by the population vector until convergence. The trajectories can then be visualized on a unit sphere (see Fig. 14). This shows that, for a non-uniform distribution of preferred directions, the lateral weights define specific directional attractors to which, in absence of vectorial inputs, the population vector will converge. Finally, similarly to the 2D case, we measured the drift resulting from performing a 3D rotation. The results are shown in Fig. 12 (right). The striking difference to the 2D case is that the mean error E_β becomes smaller for high values of parameter η, and bigger for small values. This interesting property appears to be the

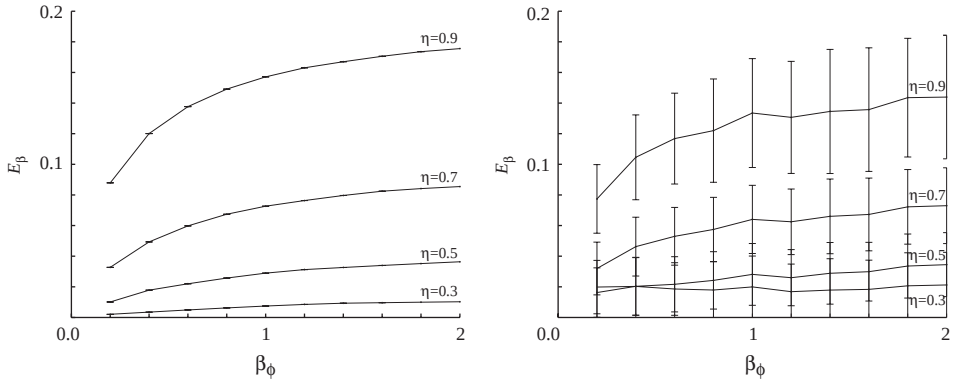

Fig. 12. (Left) Each line corresponds to the mean error E_β against β_ϕ for different values of parameter η, while simulating planar rotation. The standard error is not shown as it is too small. (Right) Same error plot, but for a 3D rotation.

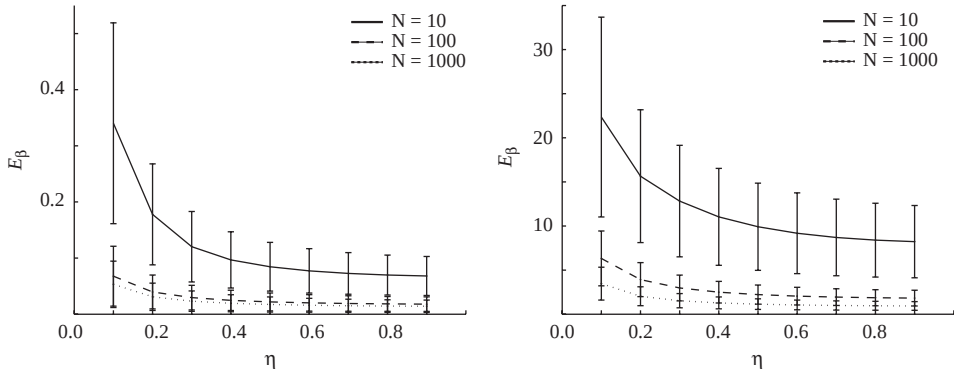

Fig. 13. Error measurements for different population sizes, expressing the ability of a 3D population with a quasi-uniform distribution of preferred directions to represent its inputs. (Left) Amplitude error E_β, (right) angular error E_θ.

result of a compromise between the two previously mentioned opposite effects that η can have on the network.

6. Conclusion

We have shown how a neural mechanism based on population vector coding can perform vectorial operations, such as translations and rotations. Our network is based on a two layers architecture that is able to produce a non-linear transformation of its inputs. It consists in a global modulatory effect of the whole population activity, as observed in several sensorimotor areas [7,15]. This work suggests that population coding could be the neural basis of transformations across frames of reference [12]. Note that the model's hypothesis that 3D frames of

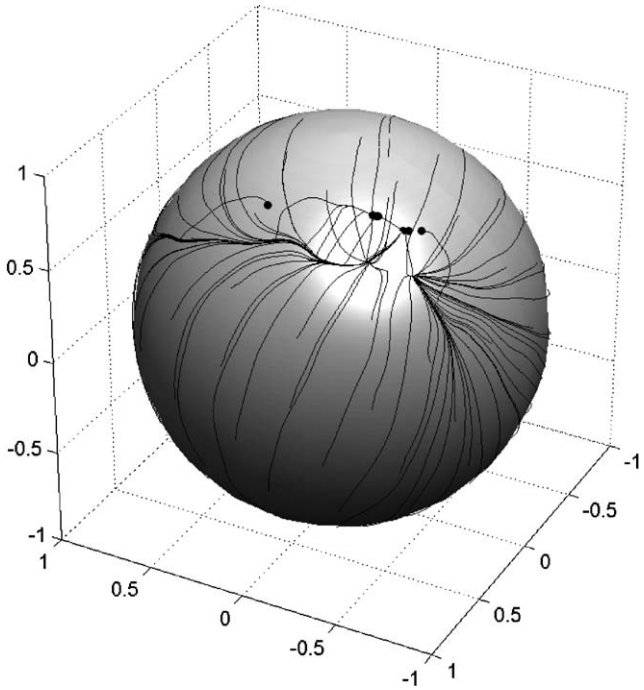

Fig. 14. Illustration of the directional attractors generated by an imperfect uniform distribution of preferred directions, while the population receives only an homogeneous input. Each line terminated by a dot on the unit sphere corresponds to a trajectory followed by the population vector of the network.

reference transformations are performed serially is in line with the *gradient hypothesis* for sensorimotor transformations [5].

Moreover, our simulations suggest that the transformation accuracy could be related to the uniformity of the distribution of the preferred directions along populations of neurons. Indeed, our results raise the hypothesis that learning to be precise may consist in recruiting more neurons in a population and in uniformizing their preferred directions. However, the model's assumption that a FR is represented by a population of neurons, having an uniform distribution of preferred direction and exhibiting a cosine tuning curve that depends on the coding direction, is not representative of all neurophysiological data [1,17]. But as proposed by Scott et al. [17], non-uniform distributions do not prevent specific brain areas to pick up uniform sub-populations or to ponderate the weights of each neuron in an inversely proportional manner relative to the distribution of preferred directions.

Acknowledgments

This work was supported by the Swiss National Science Foundation, through Grant no. 620-066127 of the SNF Professorships program.

References

[1] B. Amirikian, A.P. Georgopoulos, Directional tuning profiles of motor cortical cells, Neurosci. Res. 36 (1) (2000) 73–79.
[2] E. Ashbridge, D.I. Perrett, M.W. Oram, T. Jellema, Effect of image orientation and size on object recognition: responses of single units in the macaque monkey temporal cortex, Cog. Neuropsychol. 17 (2000) 13–34.
[3] P. Baraduc, E. Guigon, Population computation of vectorial transformations, Neural Comput. 14 (4) (2002) 845–871.
[4] A.P. Batista, C.A. Bruneo, L.H. Snyder, R.A. Andersen, Reach plans in eye-centered coordinates, Science 285 (1999) 257–260.
[5] Y. Burnod, P. Baraduc, A. Battaglia-Mayer, E. Guigon, E. Koechlin, S. Ferraina, F. Laquaniti, R. Caminiti, Parieto-frontal coding of reaching: an integrated framework, Exp. Brain Res. 129 (1999) 325–346.
[6] S. Deneve, P.E. Latham, A. Pouget, Efficient computation and cue integration with noisy populations codes, Nat. Neurosci. 4 (8) (2001) 826–831.
[7] S. Kakei, D.S. Hoffman, P.L. Strick, Muscle and movement representations in the primary motor cortex, Science 285 (1999) 2136–2139.
[8] T. Kohonen, The self-organizing map, Proc. IEEE 78 (9) (1990) 1464–1480.
[9] G. Marsaglia, Choosing a point from the surface of a sphere, Ann. Math. Stat. 43 (1972) 645–646.
[10] M.C.W. van Rossum, A. Renart, Computation with populations codes in layered networks of integrate and fire neurons, Neurocomputing 58–60 (2004) 265–270.
[11] E. Ribot-Ciscar, M. Bergenheim, F. Albert, J.-P. Roll, Proprioceptive population coding of limb position in humans, Exp. Brain Res. 149 (2003) 512–519.
[12] E. Salinas, L.F. Abbott, Transfer of coded information from sensory to motor networks, J. Neurosci. 15 (1995) 6461–6474.
[13] E. Salinas, L.F. Abbott, A model of multiplicative neural responses in parietal cortex, Proc. Natl. Acad. Sci. USA 93 (1996) 11956–11961.
[14] E. Salinas, Self-sustained activity in networks of gain-modulated neurons, Neurocomputing 52–54 (2003) 913–918.
[15] H. Scherberger, R.A. Andresen, Sensorimotor transformations, in: L.M. Chalupa, J.S. Werner (Eds.), The Visual Neurosciences, MIT Press, Cambridge, MA, 2003, pp. 1324–1336.
[16] A.B. Schwartz, R.E. Kettner, A.P. Georgopoulos, Primate motor cortex and free arm movements to visual targets in three-dimensional space. I. Relations between single cell discharge and direction of movement, J. Neurosci. 8 (8) (1988) 2913–2927.
[17] S.H. Scott, P.L. Gribble, K.M. Graham, D.W. Cabel, Dissociation between hand motion and population vectors from neural activity in motor cortex, Nature 413 (2001) 161–165.
[18] K. Zhang, Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory, J. Neurosci. 16 (6) (1996) 2112–2126.

Eric Sauser is a PhD student with the Autonomous Systems Lab at the Swiss Federal Institute of Technology, Lausanne (EPFL). He received his BSc/MSc in Computer Science from the Swiss Federal Institute of Technology, Lausanne (EPFL) in March 2003. His research interests cover Artificial Neural Networks, Computational Neuroscience and Imitation Learning.

Aude Billard is an SNF Professor at the school of Engineering at the Swiss Federal Institute of Technology (EPFL). She is also an Adjunct Assistant Professor at the Computer Science department at the University of Southern California. She received her BSc (1994) and MSc (1995) in Physics from the EPFL, with specialization in Particle Physics at the European Center for Nuclear Research (CERN). She received her MSc in Knowledge Based System (1996) and her PhD in Artificial Intelligence (1998) from the department of Artificial Intelligence at the University of Edinburgh. She, then, was a Post-doctoral Fellow at IDSIA and LAMI (EPFL, 1998-99). Aude Billard was a Research Associate (1999-2000) and a Research Assistant Professor (2000-2002) at the Computer Science department at the University of Southern California. Dr. Billard's research interests cover the fields of Artificial Neural Networks, Robotics, Neural Modeling, Computational Neuroscience and more generally Machine Learning. Her work tackles special topics, such as Programming Through Demonstration, Imitation Learning and Language Acquisition.