The encoding of proprioceptive inputs in the brain: knowns and unknowns from a robotic perspective

Matěj Hoffmann

Istituto Italiano di Tecnologia
Via Morego 30
16163 Genova, Italy
Email: matej.hoffmann@iit.it

Naďa Bednárová

Fakulta Elektrotechnická, ČVUT v Praze
Zikova 1903/4
166 36 Praha
Email: nada.bednarova@gmail.com

Abstract

Somatosensory inputs can be grossly divided into tactile (or cutaneous) and proprioceptive – the former conveying information about skin stimulation, the latter about limb position and movement. The principal proprioceptors are constituted by muscle spindles, which deliver information about muscle length and speed. In primates, this information is relayed to the primary somatosensory cortex and eventually the posterior parietal cortex, where integrated information about body posture (postural schema) is presumably available. However, coming from robotics and seeking a biologically motivated model that could be used in a humanoid robot, we faced a number of difficulties. First, it is not clear what neurons in the ascending pathway and primary somatosensory cortex code. To an engineer, joint angles would seem the most useful variables. However, the lengths of individual muscles have nonlinear relationships with the angles at joints. Kim et al. (Neuron, 2015) found different types of proprioceptive neurons in the primary somatosensory cortex – sensitive to movement of single or multiple joints or to static postures. Second, there are indications that the somatotopic arrangement ("the homunculus") of these brain areas is to a significant extent learned. However, the mechanisms behind this developmental process are unclear. We will report first results from modeling of this process using data obtained from body babbling in the iCub humanoid robot and feeding them into a Self-Organizing Map (SOM). Our results reveal that the SOM algorithm is only suited to develop receptive fields of the posture-selective type. Furthermore, the SOM algorithm has intrinsic difficulties when combined with population code on its input and in particular with nonlinear tuning curves (sigmoids or Gaussians).

1   Introduction

Although proprioception does not belong to the commonly held "five senses"(sight, hearing, smell, touch, taste) view that dates back to Aristotle, it is, along with sense of balance for example, of outmost importance for our everyday existence. What is proprioception actu-ally? The term has its origin in the work of Sherrington (1906) who stated: "In muscular receptivity we see the body itself acting as a stimulus to its own receptors—the proprioceptors". In short, we will equate it with *sense of limb position and movement* (see (Proske and Gandevia, 2012) for an extensive survey). The reason why proprioceptive sensations are receiving so little attention in our everyray experience is mainly because we are largely unaware of them, which in turn can be attributed to their predictability. Unlike "exteroceptors", such as sight or hearing that focus on events in the external environment that are not fully under our control, the position and movement of our limbs is something that is typically under our control and thus, these sensations typically receive less (conscious) attention (Proske and Gandevia, 2012). A related term is *kinesthesia*, which strictly means movement sense, but has been sometimes equated with proprioception.

2   Representation of proprioception in biology

2.1   Receptors

The key receptors in proprioception are muscle and skeletal mechanoreceptors: muscle spindles, Golgi tendon organs, and joint capsule mechanoreceptors—an overview is provided in Table 1. They sense different physical quantities: while the muscle spindles and joint capsule mechanoreceptors are sensing kinematic variables (position and velocity), the Golgi tendons are sensing dynamic (or kinetic) quantities (forces). In addition, stretch-sensitive receptors in the skin (Ruffini endings, Merkel cells in hairy skin, and field receptors) also signal postural information Kandel et al. (2000, p. 443).[1] However, in summary, "the modern view has muscle spindles as the principal proprioceptors"(Proske and Gandevia, 2012).

---

[1]Muscle stretch can roughly be equated with muscle length.

Receptor	Fiber group / name	Submodality
Muscle spindle primary	$A\alpha/Ia$	Muscle length and speed
Muscle spindle secondary	$A\beta/II$	Muscle stretch ($\sim$ length)
Golgi tendon organ	$A\alpha/Ib$	Muscle contraction / tension ($\sim$ force)
Joint capsule mechanoreceptors	$A\beta/II$	Joint angle

Tab. 1: Muscle and skeletal mechanoreceptors.

2.1.1 Muscle spindles as principal proprioceptors

Muscle spindles respond in terms of their firing rate to changes of muscle length (that is the dynamic response – muscle spindle primary) as well to a sustained length of the muscle (static/tonic response – muscle spindle primary and secondary). "Position sense can therefore be envisaged as signaled by the mean rate of background discharge in muscle spindles, including that generated by both primary and secondary endings"(Proske and Gandevia, 2012). However, the position signal delivered by individual spindles seems to be far from perfect. For example: (i) muscle spindles have also efferent innervation (fusimotor – gamma neuron activity) and thus there are differences between active and passive muscle stretching, (ii) some spindles generate background activity at all muscle lengths, no matter how short the muscle; others fall silent at short lengths. It is also not clear how the limits are retrieved and absolute position information seems to be inaccurate. Such effects may be mitigated if population of spindles are considered and possibly also by recruiting the information from joint receptors (acting possibly as limit detectors). The interested reader is referred to (Jones et al., 2001; Macefield, 2005; Proske and Gandevia, 2012).

2.1.2 From muscle length to joint angle

The spindles deliver information about the length of a single muscle – flexor or extensor – acting over a particular joint (or often also two joints in case of biarticular muscles). Not surprisingly, every spindle afferent is directionally tuned to movement of the joint in a particular direction—as can be inferred from the arrangement of its parent muscle (Jones et al., 2001). However, in order to get information about the joint angle, one has to also consider that there is a nonlinear relationship between the muscle lengths and the angle at the joint. This is illustrated in Fig. 1 for the elbow joint (after (Shadmehr and Wise, 2005)). The length of triceps, λ, is linked to θ, the supplementary angle to the elbow angle ϕ, by the following equation:

$$\lambda = \sqrt{a^2 + b^2 - 2 \cdot a \cdot b \cdot cos(\theta)} \qquad (1)$$

Where and how (or if) the inverse of this transformation (to acquire θ and eventually ϕ from λ) is performed by spinal or cortical neurons has to our knowledge not been described in the literature.
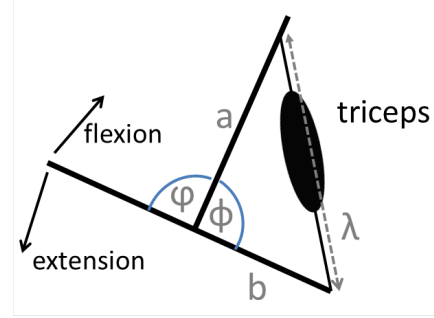

Figure 1: Geometry of elbow joint and triceps muscle. After (Shadmehr and Wise, 2005).

To encode the 3D configuration of a complex joint (such as wrist or ankle), taking into account signals from a population of spindles of different muscles acting over the joint seems indispensable (Jones et al., 2001; Bergenheim et al., 2000)—but again, it is unclear how and where would this information be combined to give an explicit representation of such 3D joint configuration.

2.2 Neural encoding

As described above, there seems to be concensus that muscle spindles are the principal proprioceptors. However, it still remains unclear what the encoded variables are (in particular whether they are the lenghts of individual muscles rather than joint angles). Second, we have also not found in the literature which neural coding type is applied to the proprioceptive modality. We will focus on rate codes—where mean neural firing rate carrries information—rather than temporal (spiking) codes. The general rate types are:

- *Intensity code.* The activity of the neuron is proportional to (or a monotonic function of) the coded variable (Mallot, 2013).

- *Local coding scheme.* The sensory space is divided into nonoverlapping areas that can be resolved by small groups of topographically arranged neurons with small receptive fields (Ghazanfar et al., 2000). Mallot (2013) calls this *channel coding* or *labeled line code without overlap*.

- *Distributed (population) coding scheme.* Neurons have relatively large and overlapping receptive

fields compared with the sensory resolution measured behaviorally (Ghazanfar et al., 2000). (Mallot, 2013) calls this *labeled line code with overlap* and the "standard way" of neural coding.

The encoding type is also intertwined with different *neural tuning curves* that relate the average firing rate of a neuron to relevant stimulus parameters. The tuning curves cited in the literature are:

- *Linear*
- *Sigmoidal*
- *Gaussian*

All of the above will be implemented and illustrated later in this article in Section 4.2. All tuning curves have specific properties which may be more suited to represent a stimulus with particular characteristics. In the linear case, firing rates are a linear function of the stimulus. Different neurons would have different equations (in terms of slope and y-intercept) (see e.g., (Zipser and Andersen, 1988)). Sigmoidal tuning curves are also monotonic but nonlinear functions of the input. Gaussian tuning curves feature a peak—are nonmonotonic functions—and may best encode the stimulus at this tuning curve peak. This may be useful if the input is noisy, since high firing rates are the neuron's most distinct response. Alternatively, stimulus may be coded not at the peak but at the high-slope regions of the tuning curve where the discrimination is the highest (see (Butts and Goldman, 2006) for an information theoretic analysis). The latter applies both to Gaussian and sigmoidal curves (not the linear functions where the slope is constant). However, we have not found in the literature what the tuning curves used for proprioceptive stimuli are.

2.3 Ascending neural pathway

The principal ascending neural pathway from the receptor sites to the cortex is common with that carrying "fine touch" and it is the so-called medial lemniscal pathway with "relay stations" at dorsal root ganglion, medulla, and thalamus, before reaching the neocortex, mostly terminating in area 3a.

2.4 Representation in the cortex

Before we briefly review the findings regarding the representation of proprioception in the cortex, it is worth noting that the findings about neuronal responses and receptive fields obtained using electrode recordings in monkeys often come from rather unspecific stimulation of so-called deep receptors: "light to moderate taps, digit and limb manipulation, and light pressure, categorized as 'deep stimulation', were used to stimulate the muscles, joints, and skin."(Seelke et al., 2011) A notable exception is (Kim et al., 2015) who have used precisely controlled proprioceptive and cutaneous (or

"tactile") stimulation. They also note that unlike cutaneous sensory processing, cortical mechanisms underlying proprioception are poorly understood.

2.4.1 Anterior parietal cortex

The anterior parietal cortex encompasses areas 3a, 3b, 1, and 2 and is also commonly referred to as primary somatosensory cortex (SI), even if that may be ambiguous (some consider only area 3b as primary). From these areas, area 3a is considered the most primary receptive site for proprioceptive inputs (while area 3b is the "most primary" cutaneous/tactile). "Area 3a ultimately receives input from group Ia muscle spindle afferents, and contains neurons that respond to the stimulation of these and other deep receptors in the skin." (Huffman and Krubitzer, 2001). (Kim et al., 2015) precisely controlling movements of individual digits (fingers) in a macaque monkey have identified three types of proprioceptive neurons:

- *Single-digit position scaled*
- *Multi-digit position scaled*
- *Posture-selective*

The first type, single-digit position scaled, means that the neuron responds in a monotonic fashion to the position (i.e. angle in a particular axis) of a single digit. This would probably be most compatible with the intensity code and a linear tuning curve. The second type, multi-digit position scaled, is sensitive to positions of mutliple digits, which is still suggestive of a rate code, but some additive mechanism for multiple joints is necessary. The posture-selective neurons, on the other hand, do not scale across specific a dimension. They are responsive to a particular posture—a specific conformation of the hand, i.e. specific position of a number of different joints. Such neurons would be compatible with either local or population coding scheme and perhaps go most naturally with Gaussian tuning curves (the peaks of tuning curves at specific positions of every joint). (Kim et al., 2015) found all of these neuron types throughout all of SI: they were most abundant in area 3a, but present also in 2, 1, and 3b. While their results were obtained for the hand receptive fields only, a similar pattern may perhaps be expected also for other body parts—arm joints, for example.

Regarding topography, (Krubitzer et al., 2004) using a macaque monkey found that area 3a contains a complete representation of deep receptors and musculature of the contralateral body and that the general organization of body part representations mirrors that of area 3b. However, the topographic organization is less precise and less orderly in 3a and some representations, such as the hand, are split or fractured like maps of motor cortex.

2.4.2 Posterior parietal areas

Whereas in the anterior parietal cortex, the receptive fields of neurons typically span individual joints or combinations of a few adjacent joints, in the posterior parietal areas, the receptive fields are in general larger. Body posture is primarily encoded in the superior parietal lobule (areas 5 and 7), a putative homologue of the so-called parietal reach region in the monkey Pellijeff et al. (2006).Seelke et al. (2011), for example, analyzed the receptive field (RF) of neurons in posterior parietal area they termed 5L (lateral) of the macaque and found representation of "deep receptors" of the limb. The RFs of individual neurons often spanned multiple joints, such as elbow and wrist or shoulder and elbow.

2.5 Development in the fetus and newborn

Somatosensation and its representation starts developing already in the womb. In fact, the somatosensory modality starts functioning over the whole body from as early as 17th gestational age and before other sensory modalities Bradley and Mistretta (1975), although connections to the neocortex are not present until the 20th gestational week. Please refer to Milh et al. (2007) and references therein for a more detailed account. Spontaneous motor activity is playing an important role. From about 22 weeks, touching the mouth is already different from touching eyes (more delicate). However, Krubitzer et al. (2004) using macaque reported that unlike 3b ($\sim$ "primary tactile") in which the topography seems to be present in neonates, area 3a (and 1 and 2) are not responsive in neonates in New and Old World monkeys. Furthermore, and again unlike 3b, the construction of area 3a is to a large extent based on the use of a particular body part rather than innervation density (as a matter of fact, about half of the muscle spindles in the body are located in the neck (Cole and Paillard, 1995)).

3 Computational and robotic modeling of proprioception

There has been some computational and robotic modeling addressing the development of somatosensory representations, even though more work has focused on the tactile rather than proprioceptive modality. The work of Kunyioshi, Mori, and colleagues has addressed prenatal development in foetus simulator including a uterine environment. Their models can also be said to be the most biologically motivated. The foetus simulator has 198 muscles and is actuated with a central pattern generator model. It is equipped with 764 tactile sensors as well as muscle spindles and Golgi tendon organs. Different works addressed the development of SI (Yamada et al., 2013) or a hierarchy of SI and SII using a denoising autoencoder (Sasaki et al., 2013). Aflalo and

Graziano (2006) investigated motor cortex development using a Self-Organizing Map. Drix and Hafner (2014) studied learning proprioceptive and motor features from joint angle values in a quadruped robot using a neural "intrinsic plasticity" rule. Finally, Žabkar and Leonardis (2016) presented a learning framework that builds compositional hierarchies of the joint angle space of the Nao robot.

4 Modeling on iCub humanoid robot and Self-Organizing Maps

We have studied the development of proprioceptive representations using the simulator of the iCub humanoid robot (Metta et al., 2010; Tikhanoff et al., 2008). Here we present an extension of the results of Bednárová (2015).

4.1 Data collection

Data was collected during a body babbling phase. However, we did not employ random body/motor babbling, which would not induce any structure in the proprioceptive space (space of joint angles). Instead, we were inspired by Aflalo and Graziano (2006) who, studying the possible motor cortex development, employed so-called ethological categories: coordinated movements that the monkey uses often (e.g., hand-to-mouth, manipulation in central space in front of the monkey). Thus, we have created a simple data set where the iCub moves the hand in a region of 20x15x15 cm in front of its face and follows it with its gaze (foveating with both eyes). That is, the positions were chosen randomly in the Cartesian space and then the iCub Cartesian and gaze controllers (Pattacini, 2011) were employed to move the hand to the position and follow it by gaze. Thus, joint positions were not randomly chosen, but corresponded to coordinated behavior inspired by a typical "ethological category" of an infant. Joints were sampled at 50 Hz and there were 7 arm joints (3 shoulder, 2 elbow, 2 wrist) of the left arm and 6 joints of the neck and eyes plant (neck pitch, roll, yaw; eyes tilt, version, vergence), giving 10 DoF (Degrees of Freedom) in total. The data set employed had a 20 min. duration.

4.2 Encoding iCub joints data

Before feeding the data to a Self-Organizing Map (SOM), we have implemented all three types of neural encoding known from biological neural systems, as reviewed in Section 2.2. In addition, we have also studied the case where the joint values are only normalized (taking the range of individual joints into account) and fed directly into the SOM.

In order to determine the number of tuning curves to encode every DoF (every DoF having a diffe-

rent range), there are two options: (a) set the number of tuning curves constant, or (b) set the offset between adjacent tuning curves constant. In both cases, the other variable can then be calculated correspondingly. In what follows, we have experimented with both variants. In all expressions below, x represents the original angle values in degrees and y represents the encoded value. The output range is in the interval $< 0, 1 >$.

4.2.1 Linear functions

Individual tuning curves were simply given by the equation

$$y = (a \cdot x) + b$$

with varying parameters for the slope a and y-intercept b. Fig.2 illustrates the whole population coding for a single DoF with 20 tuning curves: 10 with positive and 10 with negative slopes.
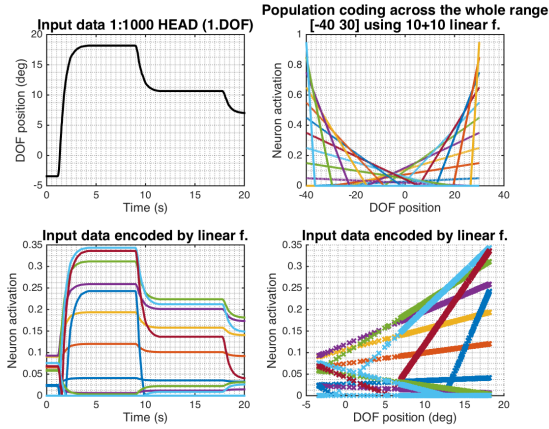

Figure 2: Movement of one DoF encoded by linear tuning curves. (Top left) Input data. Joint position over a 20 s interval. (Top right) Tuning curve activations on input space. (Bottom left) Input data encoded by individual tuning curves. (Bottom right) Close-up of part of the input space and its encoding.

4.2.2 Sigmoids

A sigmoidal tuning curve is given by the following equation:

$$y = \frac{1}{1 + e^{\,sgn \cdot (-x + offset)}}$$

where sgn sets the curve orientation (positive or negative), $offset$ represents the position of the inflexion point and x are the input values.

As noted above, there are two options for setting up tuning curve parameters for the population. If we set up a fixed number of curves encoding each DoF, the offsets for every DoF for sigmoids in one direction are cal-

culated as:

$$offset = \frac{(max\_range - min\_range)}{(number\_of\_sigmoids)}$$

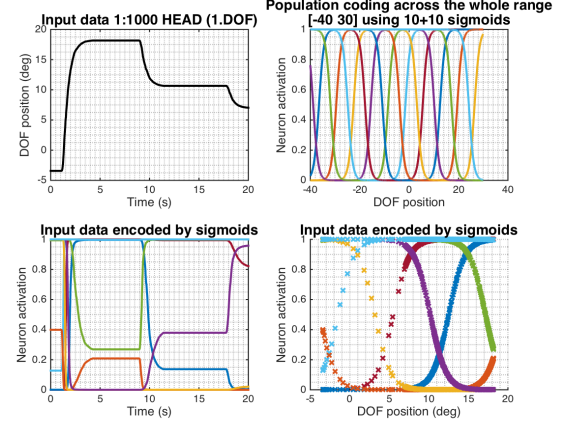An illustration of the encoding for a single DoF is depicted in Fig. 3.

Figure 3: Movement of one DoF encoded by sigmoidal tuning curves. See Fig. 2 for a description of all panels.

4.2.3 Gaussian curves

Gaussian tuning curves are described by the following equation:

$$y = e^{-\frac{(x - \mu)^2}{2\sigma^2}}$$

with the curve's center or peak at μ, σ the distance to the inflexion point and x the input.

In this case, the curves are symmetrical about the center and we will need only one set of tuning curves (no positive and negative). The offsets between curves can be determined similarly to the previous section. The additional parameters, μ and σ, for the population will then be calculated as follows:

$$\sigma = \frac{(max\_range - min\_range)}{(number\_of\_gauss - 1)}$$

$$\mu = min\_range + \sigma + offset$$

An illustration of the encoding for a single DoF is depicted in Fig. 4.

4.3 Decoding

Although the inverse mapping to the encoding may not be strictly required in all cases when stimuli are encoded by the brain, it seems that it may often be useful. In our particular application where the data are futher processed by a SOM algorithm, decoding is necessary to visualize and quantify what the network has learned. Both
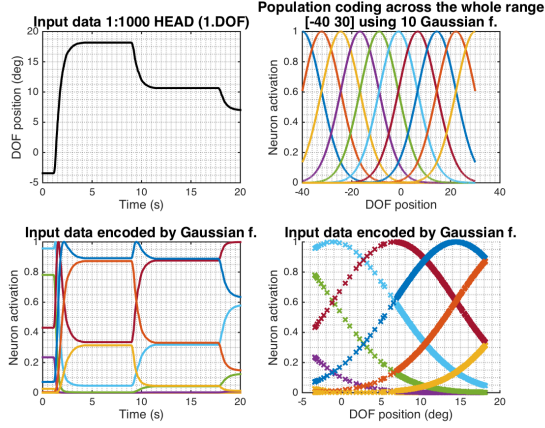
Figure 4: Movement of one DoF encoded by Gaussian tuning curves. See Fig. 2 for a description of all panels.

the linear and sigmoidal tuning curves are monotonic functions and thus a proper inverse exists. Conversely, the Gaussian tuning curve is a nonmonotonic, many-to-one function, that does not have a proper inverse. The formulas for obtaining the inverse are specified below.

4.3.1 Linear functions – decoding

$$x = \frac{(y - b)}{a}$$

4.3.2 Sigmoids – decoding

$$x = -\frac{ln(1 - y) - ln(y) - offset \cdot sgn}{sgn}$$

4.3.3 Gaussian curves – decoding

$$x = \mu + \sqrt{-ln(y) \cdot 2\sigma^2}$$

4.4 Self-organizing map on proprioceptive data

A Self-Organizing Map (SOM) (also self-organizing feature map or Kohonen map) algorithm (Kohonen, 1982, 1990) was used to process the encoded or normalized only joint angle data. A freely available SOM toolbox (Vesanto et al., 1999) was used.

4.4.1 Problems with population coding as SOM input

Let us consider one DoF with movement range from -40 to 30 degrees. Firstly, we encode the movement of this joint using 10 Gaussian curves, then we process the encoded values by a SOM, and finally we use backward transformation for decoding the result back to angles.

Let us also consider a simplified version of SOM with only one output neuron. The input space after encoding has thus 10 dimensions and there are also 10 weight vectors from the input space to the single output neuron. The range of the input variables as well as weight vector components is $< 0, 1 >$. Standard SOM random initialisation takes all input data, calculates the range of every input dimension in the dataset and randomly picks one number from that range to initialize the weights. A property of the SOM algorithm is that the weight vector of an output neuron can be interpreted as a representative point in the input space. However, recall that the 10 dimensions are a population vector derived from a single DoF value. Hence, initializing every weight vector randomly will have the consequence that this point in the input space does not correspond to any angle of the joint, as illustrated in the middle panel of Fig. 5 for the case of Gaussian curves. Therefore the initialisation of SOM had to be modified to first randomly pick an angle from the joint angle range, encode it using the respective encoding and use these values to seed the SOM weight vector—see Fig. 5 right panel.
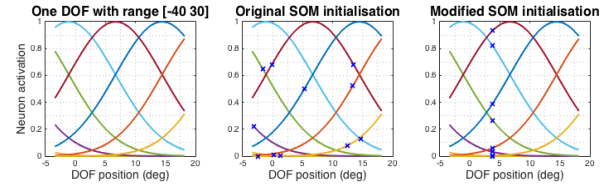

Figure 5: Modified SOM initialisation. See text for details.

However, this modification does not warrant that the representation will be consistent—in the sense that the representative vectors of the SOM output layer will correspond to joint angles—after learning. During training, the SOM algorithm iterates over vectors from the input space and everytime determines the most similar weight vector among the output layer neurons. Such neuron becomes the so-called best matching unit (BMU) which updates its weights in order to decrease the distance to the input presented (neurons in the BMU's neighborhood are also updated). This is illustrated in Fig.6 using Gaussian curves. The left panel depicts an input vector and its encoding. The middle panel a hypothetical initialisation of the SOM weight vector to a different angle. Finally, the right panel depicts the situation after a training step – weight vector update. The weights are drawn toward the input vector in the encoded space. However, since the different tuning curves have different slopes at the different activation values, the updated vector will not correspond to any real joint angle on the input, as illustrated by the green marks in the right panel. With Gaussian curves, there is the additional complication that the inverse mapping is not unique. This is not the case for sigmoids and linear tuning curves. However, the problem is still present, even for the li-

near case: even if the slope of a single curve is constant, since the pool contains lines with different slopes, the reverse transformation will also not give a unique angle of the DoF under consideration. This situation is likely to worsen with every training step.
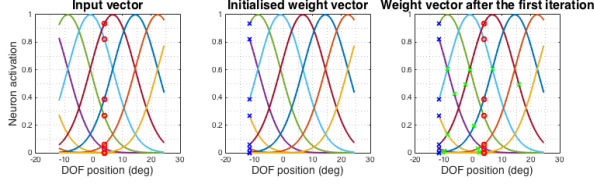

Figure 6: Inconsistency of population coding and SOM weight vector update.

In this case, if one wants to decode from the SOM's weight vector to the original space of joint angles, that is decode the values of a population, there is no straightforward solution. The method that worked the best for us was non-parametric Kernel Density Estimation [2] (KDE) that estimates the probability distribution of input data using the following expression:

$$f_h(x) = \frac{1}{nh} \sum_{i=1}^{N} K(\frac{x - x_i}{h}); \ -\infty < x < \infty$$

where h represents so called 'bandwidth' – width of the window, influencing the 'smoothness' of the curve, K_h represents normalised probability function, for example Gaussian, and $x_i, i = 1..n$ is the input data. An example is provided in Fig. 7.
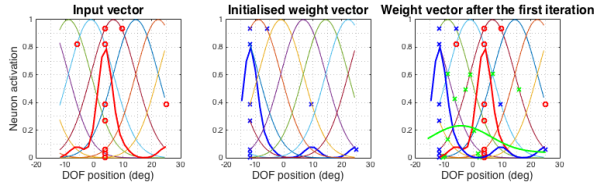

Figure 7: Kernel density estimation for decoding the population code from Fig. 6.

4.4.2 Training a 5x5 SOM from the 10 DoF data set

Using the data set specified in Section 4.1, we trained a SOM with a 5x5 output lattice. Various combinations of encoding (linear, sigmoid, Gaussian curves, or normalized inputs only), order of inputs as presented during training (ordered as in data set or randomly shuffled), and training length (number of reiterations of the whole input dataset in one map training) were tested. The connections from the input layer to the SOM were all-to-all, that is, every output neuron of the map was learning to

---

[2] http://www.mathworks.com/help/stats/kernel-distribution.html

represent a point in the space of all DoF—a complete posture of the robot in terms of arm and head.

'Random' input data order as well as longer training length always gave us better results, so we will report results for this setting only. We also tested the two variants of population coding setup—fixed number of curves per DoF or fixed offset between curves. Tests showed that the resulting map has better quality if we set up constant number of functions encoding every DoF. This guarantees that every DoF will have an equal number of input neurons to the SOM.

Various measures have been proposed to numerically assess the organization of trained SOMs (for an overview, see (Polani, 2002) and references therein). In general, they either assess the vector quantization or the topology preservation capability of the SOM algorithm. We have experimented with different measures. First, Fig. 8 provides a qualitative insight by visualizing one of the best learned maps (with Gaussian tuning curves). The representative vectors of every output neuron were decoded back to the joint angle space and visualized on the iCub simulator. Some topology preservation in the posture space is apparent.
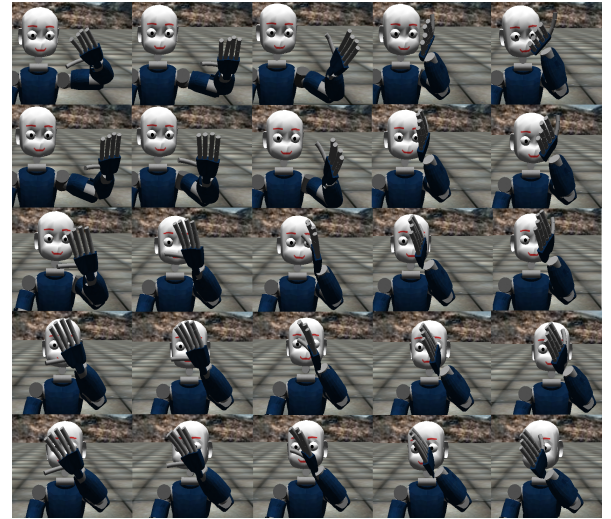

Figure 8: Visualization of SOM after training and decoding to angles. Every cell of the 5x5 matrix illustrates the posture represented by each of the output layer neurons. The decoded DoF positions were visualized on the iCub simulator.

A quantitative insight is provided in Fig. 9. Quantization error measures the quality of input data representation by calculating the average distance of every input vector with its BMU vector from the learned map. We have evaluated the criterion for different types of population coding and different number of tuning curves. In addition, we tested the case in which the joint angle values are only normalized and passed directly to the SOM. Note that in this case, the input space has only 10 dimensions (10 DoF), whereas in the other cases where

population coding is applied, the input space dimension is 10 x nr. curves encoding a single DoF.

The results clearly demonstrate that all population encoding variants lead to worse results compared to the case where normalized inputs were directly fed to the SOM. This may be surprising at first, but it is a consequence of the problems illustrated in Section 4.4.1. The problem is amplified if more tuning curves per DoF are used. Furthermore, the situation gets also worse for the nonlinear tuning curves (sigmoids and Gaussians). This confirms that there is an inherent incompatibility between population coding and the adaptation step of the SOM algorithm.
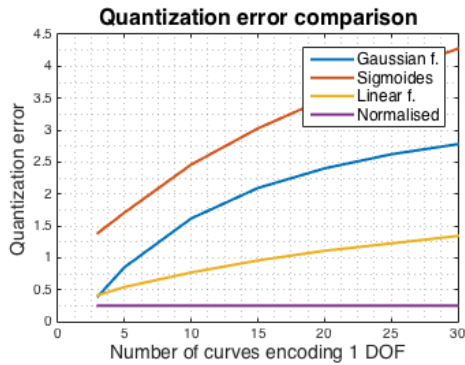

Figure 9: Comparison of quality of learned maps in terms of quantization errror. Results for different encoding types as well as different number of tuning curves are plotted. For the sigmoids and linear functions, the nr. curves per DOF corresponds to their count with one orientation—the total number was then double. Random input order and training length of 6 cycles was used in all cases.

5   Discussion, open questions, future work

This article without doubt raises more questions than provides answers. With the the goal of providing a computational model of the representation of proprioception in the brain and its development, it falls significantly short of its aspirations. However, we feel that a number of important issues have been identified, which we discuss below.

First of all, empirical knowledge regarding the workings and development of proprioception and its representation in biology is fragmented and partially contradictory. Or, perhaps, such is the information that the the neural systems dealing with proprioception are processing. Feldman (2009), for example, concludes that "ambiguity of positional information is characteristic of all afferent signals involved in kinesthesia". We want to make a number of observations. First, the view of isolated bottom-up processing of proprioceptive information is wrong. There is now sufficient evidence sug-

gesting that processing somatosensory information is a "team effort"(Saal and Bensmaia, 2014), where tactile and proprioceptive (Kim et al., 2015) and even vestibular and downstream (top-down) information (Huffman and Krubitzer, 2001) is integrated. Area 3a—even if the most "proprioceptive" and most primary—can thus hardly be considered as an isolated area processing ascending proprioceptive inputs. Second, the mechanism of formation or development of the proprioceptive representations is unclear. As pointed out earlier, there is evidence suggesting that the topography of area 3a in monkeys is to a large extent learned after birth (Krubitzer et al., 2004). However, once learned, its topography is very similar to that of neighboring area 3b receiving primary tactile inputs. This may seem surprising, since the "content" of information coming from the two modalities seems quite different. Nonetheless, the connections between these areas and probably also the frequent correlations in proprioceptive and tactile inputs from same body parts could account for this.

The computational model we presented rests on a number of simplifying assumptions. First, the data set used is preliminary and arbitrary: we used only one coordinated movement type—or ethological category (Aflalo and Graziano, 2006)—where the robot was moving the hand in front of its face and following it with gaze. The repertoire of behaviors in newborns needs to be investigated and implemented. Furthermore, newborns' coordination may in fact be severely limited and it is also possible that the appropriate stimulation type is not coordinated movement, but rather more random and isolated movements such as the muscle twitches (Khazipov et al., 2004). The correlation structure in the proprioceptive data could be induced through the agent's embodiment. This would probably require a more detailed model of the musculo-skeletal system as well as implementation of reflexes such as the stretch reflex (similar to (Kuniyoshi and Sangawa, 2006)). Second, we did not deal with all the submodalities of proprioception and restricted ourselves to "muscle spindles"—in fact, taking an additional shortcut and directly using joint angles rather than the length of (possibly also biarticular and antagonistically arranged) muscles.

In our model, we experimented with different types of population coding and the Self-Organizing Map algorithm. The most important findings are detailed in what follows. First, as detailed above in Section 2.4.1, Kim et al. (2015) identified different types of proprioceptive neurons in SI, namely position-scaled (single or multi-digit) and posture-selective. It is evident that our model can only learn to represent the posture-selective type. The SOM algorithm is in essence learning to efficiently represent points from the input space, devoting more space in the output layer to more frequent inputs and carrying over topological relationships from the input to the output space. In our case, it basically learns to pick up the most frequent postures (or

"postural synergies"). However, it cannot preserve the position-scaling property—a type of intensity code—increasing neuron firing as a monotonic function of joint position. Finding an algorithm that would preserve this feature while at the same time having the useful properties of SOM (vector quantization, topology preservation) remains the topic of our future work. Second, in our model, the connections from the input space (joint angles, possibly after population coding) to the output layer were all-to-all. That is, every output neuron of the SOM was learning to represent a configuration of all arm and head joints. This is problematic from a mathematical point of view because the topological relationships in the input space—similarity of the different postures in the configurations of all 10 DoF—are clearly of higher dimensionality than 2, while the output layer lattice of the SOM we used was only 2-dimensional and thus the algorithm was faced with the impossible task of reducing the topological manifold from the input space to mere 2 dimensions. Instead, the RFs of neurons in anterior parietal cortex typically span single or multiple joints (Kim et al., 2015), or multiple joints (e.g., 2 or 3) in the posterior parietal cortex (Seelke et al., 2011). Therefore, it seems that the connections between input and output should not be all-to-all but constrained to some regions of the input space only. A modification of the SOM algorithm constraining the maximum receptive field size (MRF-SOM) of the output neurons was presented in (Straka, 2014; Hoffmann et al., 2016). A hierarchy of such maps may give rise to similar gradual expansion of RFs as observed when going upstream in the cortex. Third, we have learned from our study that the SOM algorithm has intrinsic difficulties with input preprocessing using population code. This has to do with the fact that the input (joint angle in this case) is encoded using a population of tuning curves that have different slopes (different in different lines in the linear function case; different even at different points of single curve in sigmoids and Gaussians). After encoding, the SOM weight update occurs in this transformed space; however, this leads to inconsistency in the population if it is to be mapped back to the initial joint space. This is even more problematic if Gaussian curves are used as there is no proper inverse due to their nonmonotonicity. Perhaps, it may not be the necessity for the brain to peform this decoding; yet it seems that the encoding in combination with the SOM adaptation step does introduce undesired transformations to the information that is represented. In the robot, where joint angles are directly available, using them directly leads to superior results. In biological systems, population coding may be inevitable because the information is only available in a distributed fashion, such as from populations of muscle spindles and other receptors. As a consequence, neural learning mechanisms may need to work in different ways than the classical SOM in this case.

Acknowledgment

Literatúra

Aflalo, T. N. a Graziano, M. S. (2006). Possible origins of the complex topographic organization of motor cortex: reduction of a multidimensional space onto a two-dimensional array. *The Journal of neuroscience*, 26(23):6288–6297.

Bednárová, N. (2015). Reprezentácia proprioceptívnych vstupov humanoidného robota icub pomocou samoorganizujúcich sa máp]. Diplomová práca, Czech Technical University, Faculty of Electrical Engineering.

Bergenheim, M., Ribot-Ciscar, E. a Roll, J.-P. (2000). Proprioceptive population coding of two-dimensional limb movements in humans: I. muscle spindle feedback during spatially oriented movements. *Experimental brain research*, 134(3):301–310.

Bradley, R. M. a Mistretta, C. M. (1975). Fetal sensory receptors. *Physiological Reviews*, 55(3):352–382.

Butts, D. A. a Goldman, M. S. (2006). Tuning curves, neuronal variability, and sensory coding. *PLoS Biology*, 4(4):e92.

Cole, J. a Paillard, J. (1995). Living without touch and peripheral information about body position and movement: Studies with deafferented subjects. *The body and the self*, pp. 245–266.

Drix, D. a Hafner, V. V. (2014). Learning proprioceptive and motor features. In *Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2014 Joint IEEE International Conferences on*, pp. 374–378. IEEE.

Feldman, A. (2009). New insights into action–perception coupling. *Experimental Brain Research*, 194:39–58.

Ghazanfar, A. A., Stambaugh, C. R. a Nicolelis, M. A. (2000). Encoding of tactile stimulus location by somatosensory thalamocortical ensembles. *The Journal of Neuroscience*, 20:3761–3775.

Hoffmann, M., Straka, Z., Farkas, I., Vavrecka, M. a Metta, G. (2016). Robotic homunculus: Learning of artificial skin representation in a humanoid robot motivated by primary somatosensory cortex. *PLoS ONE [under review]*.

Huffman, K. J. a Krubitzer, L. (2001). Area 3a: Topographic organization and cortical connections in marmoset monkeys. *Cerebral Cortex*, 11:849–867.

Jones, K. E., Wessberg, J. a Vallbo, Å. B. (2001). Directional tuning of human forearm muscle afferents during voluntary wrist movements. *The Journal of Physiology*, 536(2):635–647.

Kandel, E., Schwartz, J. a Jessell, T. (2000). *Principles of Neural Science*, vol. 4. McGraw-Hill.

Khazipov, R., Sirota, A., Leinekugel, X., Holmes, G. L., Ben-Ari, Y. a Buzsáki, G. (2004). Early motor activity drives spindle bursts in the developing somatosensory cortex. *Nature*, 432(7018):758–761.

Kim, S. S., Gomez-Ramirez, M., Thakur, P. H. a Hsiao, S. S. (2015). Multimodal interactions between proprioceptive and cutaneous signals in primary somatosensory cortex. *Neuron*, 86(2):555–566.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480.

Krubitzer, L., Huffman, K. J., Disbrow, E. a Recanzone, G. (2004). Organization of area 3a in macaque monkeys: contributions to the cortical phenotype. *Journal of Comparative Neurology*, 471(1):97–111.

Kuniyoshi, Y. (2014). Learning from examples: Imitation learning and emerging cognition. In *Humanoid Robotics and Neuroscience: Science, Engineering and Society*. CRC Press.

Kuniyoshi, Y. a Sangawa, S. (2006). Early motor development from partially ordered neural-body dynamics: experiments with a. cortico-spinal-musculosleletal model. *Biol. Cybern.*, 95:589–605.

Macefield, V. G. (2005). Physiological characteristics of low-threshold mechanoreceptors in joints, muscle and skin in human subjects. *Clinical and Experimental Pharmacology and Physiology*, 32(1-2):135–144.

Mallot, H. (2013). *Computational Neuroscience: A First Course*. Springer International Publishing.

Metta, G., Natale, L., Nori, F., Sandini, G., Vernon, D., Fadiga, L., von Hofsten, C., Rosander, K., Lopes, M., Santos-Victor, J., Bernardino, A. a Montesano, L. (2010). The iCub humanoid robot: An open-systems platform for research in cognitive development. *Neural Networks*, 23(8-9):1125–1134.

Milh, M., Kaminska, A., Huon, C., Lapillonne, A., Ben-Ari, Y. a Khazipov, R. (2007). Rapid cortical oscillations and early motor activity in premature human neonate. *Cerebral Cortex*, 17(7):1582–1594.

Pattacini, U. (2011). *Modular Cartesian Controllers for Humanoid Robots: Design and Implementation on the iCub*. Dizertačná práca, Istituto Italiano di Tecnologia.

Pellijeff, A., Bonilha, L., Morgan, P. S., McKenzie, K. a Jackson, S. R. (2006). Parietal updating of limb posture: an event-related fmri study. *Neuropsychologia*, 44(13):2685–2690.

Polani, D. (2002). Measures for the organization of self-organizing maps. Seiffert, U. a Jain, L. (zost.), In *Self-Organizing Neural Networks*, vol. 78 z *Studies in Fuzziness and Soft Computing*, pp. 13–44. Springer.

Proske, U. a Gandevia, S. C. (2012). The proprioceptive senses: their roles in signaling body shape, body position and movement, and muscle force. *Physiological reviews*, 92(4):1651–1697.

Saal, H. P. a Bensmaia, S. J. (2014). Touch is a team effort: interplay of submodalities in cutaneous sensibility. *Trends in neurosciences*, 37(12):689–697.

Sasaki, R., Yamada, Y., Tsukahara, Y. a Kuniyoshi, Y. (2013). Tactile stimuli from amniotic fluid guides the development of somatosensory cortex with hierarchical structure using human fetus simulation. In *Development and Learning and Epigenetic Robotics (ICDL), 2013 IEEE Third Joint International Conference on*, pp. 1–6.

Seelke, A. M. H., Padberg, J. J., Disbrow, E., Purnell, S. M., Recanzone, G. a Krubitzer, L. (2011). Topographic maps within brodmann's area 5 of macaque monkeys. *Cerebral Cortex*.

Shadmehr, R. a Wise, S. P. (2005). *The computational neurobiology of reaching and pointing: a foundation for motor learning*. MIT press.

Sherrington, C. (1906). On the proprioceptive system, especially in its reflex aspects. *Brain*, 29:467–482.

Straka, Z. (2014). Tvorba senzorických map z taktilních vstupů robota iCub pomocí metod strojového učení. Diplomová práca, Czech Technical University, Faculty of Electrical Engineering.

Tikhanoff, V., Cangelosi, A., Fitzpatrick, P., Metta, G., Natale, L. a Nori, F. (2008). An open-source simulator for cognitive robotics research: the prototype of the iCub humanoid robot simulator. In *Proceedings of the 8th workshop on performance metrics for intelligent systems*, pp. 57–61. ACM.

Vesanto, J., Himberg, J., Alhoniemi, E. a Parhankangas, J. (1999). Self-organizing map in Matlab: the SOM toolbox. In *Proceedings of the Matlab DSP Conference*, vol. 99, pp. 16–17.

Yamada, Y., Fujii, K. a Kuniyoshi, Y. (2013). Impacts of environment, nervous system and movements of preterms on body map development: Fetus simulation with spiking neural network. In *Development and Learning and Epigenetic Robotics (ICDL), 2013 IEEE Third Joint International Conference on*, pp. 1–7.

Žabkar, J. a Leonardis, A. (2016). Motor memory: Representation, learning and consolidation. *Biologically Inspired Cognitive Architectures*, 16:64–74.

Zipser, D. a Andersen, R. A. (1988). A backpropagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature*, 331(6158):679–684.