PSOM$^+$: Parametrized Self-Organizing Maps for noisy and incomplete data

Stefan Klanke and **Helge Ritter**
Faculty of Technology
Bielefeld University
33501 Bielefeld, Germany
{sklanke,helge}@techfak.uni-bielefeld.de

Abstract - *We present an extension to the Parametrized Self-Organizing Map that allows the construction of continuous manifolds from noisy, incomplete and not necessarily grid-organized training data. All three problems are tackled by minimizing the overall smoothness of a PSOM manifold. For this, we introduce a matrix which defines a metric in the space of PSOM weights, depending only on the underlying grid layout. We demonstrate the method with several examples, including the kinematics of a PA10 robot arm.*

Key words - interpolation, function approximation, regularization, missing data

1 Introduction

The Parametrized Self-Organizing Map (PSOM) has been introduced in [5] as a continuous generalization of the standard Self-Organizing Map (SOM) [3]. It inherits the SOM's ability to create topology preserving mappings between an embedding (data) space and the nodes of a Cartesian grid in a lower dimensional space, but extends the SOM by explicitly defining a continuous manifold in the embedding space through a smooth mapping.
While the PSOM was designed to operate on the weights of a readily trained SOM, such preprocessing is not always needed. Often the training data (weights) can already be generated by sampling along the degrees of freedom of a system, for example in learning the forward and inverse kinematics of a robot arm [1, 2] or of single fingers [4] and in object recognition and pose estimation [6]. By including topology information, the PSOM permits the construction of highly accurate mappings from very few training examples [7].
However, in its original form the PSOM suffers from two restrictions. First, the algorithm features no explicit consideration of noise that might be present in the data (e.g. in physical measurements). Second, the original PSOM requires a complete set of grid-organized data to construct its mapping. This means that (i) training data lying in between the grid nodes can not be incorporated and (ii) even a single missing weight (e.g. a sample position not realizable because of physical constraints) makes the PSOM construction algorithm inapplicable.
This paper addresses these two issues by an elegant integration of rather standard smoothing techniques into the PSOM framework. Specifically, we present an approach to regularize PSOM-mappings in order to deal with noisy data in a principled manner and we provide a modification of the original algorithm, allowing to construct PSOMs from data that are not

organized in a grid topology (including as a special case grid-based data with missing elements). We will use the notation PSOM$^+$ when we wish to explicitly indicate the application of the new regularization approach.
Our method is based on measuring the overall smoothness of the PSOM mapping. For this, we integrate the square sum of all second derivatives of the PSOM mapping, the result of which can be expressed by a quadratic form. As a consequence, the problem of finding optimal (here: maximally smooth) PSOM$^+$ mappings can be solved by applying linear algebra.
The paper is organized as follows: In the next section, we briefly recall the original PSOM algorithm and introduce some necessary notation. Then, we derive a metric in the space of PSOM weights, allowing to calculate the overall smoothness of a PSOM manifold as a function of its weights. After that, we show how to tackle the problems of noisy data, missing weights and non grid-organized data and illustrate our method by toy examples. Finally, we show the performance of our method in a (simulated) robot kinematic learning task.

2 The PSOM algorithm

A conventional SOM consists of an array of formal neurons arranged on the nodes of a m-dimensional grid. Each neuron (characterized by a multi-index $\mathbf{g}$) has a d-dimensional weight $\mathbf{w_g}$ attached. The entirety of the weights, along with their respective neuron topology within the grid, form a discrete approximation of a possibly nonlinear manifold embedded in $\mathbb{R}^d$.
The PSOM is built on the same kind of neuron array, but interpolates the neuron's weights by a smooth vector-valued function $\mathbf{w}(\mathbf{s})$, thus defining a manifold parameterized by a continuous m-dimensional quantity, the coordinate $\mathbf{s}$ within the manifold sampled by the grid.

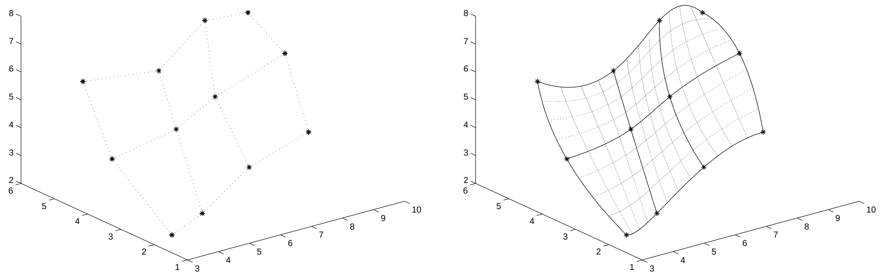


Figure 1: From SOM to PSOM. The left plot shows a 4x3 SOM with 3D weights, the right plot depicts the corresponding PSOM.

In this paper, we only work on Cartesian (but not necessarily regular) grids, which means that the set of grid coordinates $\mathbf{A} = \{\mathbf{a_g}\}$ is given by the Cartesian product of 1D coordinate sets along the different dimensions:[1]

$$\mathbf{A} = A^1 \times A^2 \times \ldots \times A^m \quad , \quad A^\mu = \{a_1^\mu, a_2^\mu, \ldots, a_{n^\mu}^\mu\} \tag{1}$$

The PSOM mapping can then be expressed by multi-dimensional Lagrange interpolation:

$$\mathbf{w}(\mathbf{s}) = \sum_{\mathbf{g}} \mathbf{w_g} b_{\mathbf{g}}(\mathbf{s}) = \sum_{\mathbf{g}} \mathbf{w_g} \prod_{\mu=1}^{m} l_{g^\mu}^\mu(s^\mu) \quad , \quad l_i^\mu(s^\mu) = \prod_{j \neq i} \frac{s^\mu - a_j^\mu}{a_i^\mu - a_j^\mu}. \tag{2}$$

Here, $\mathbf{s} = (s^1, \ldots, s^m)$ and the $l_i^\mu(s^\mu)$ are standard one-dimensional Lagrange polynomials.

[1]Throughout the paper, we denote the grid dimension by an upper Greek index.

While the standard SOM responds to an input $\mathbf{x} \in \mathbb{R}^d$ by the nearest weight $\mathbf{w}_{g^*}$ ("discrete best match"), the PSOM responds by $\mathbf{w}(\mathbf{s}^*)$ with $\mathbf{s}^*$ given as the solution of the continuous minimization problem

$$\mathbf{s}^* = \arg\min_{\mathbf{s}} d(\mathbf{w}(\mathbf{s}), \mathbf{x}), \tag{3}$$

where $d(\mathbf{x}, \mathbf{x}')$ may e.g. be chosen as the standard Euclidean distance $\|\mathbf{x} - \mathbf{x}'\|$. By including only components of $\mathbf{x}$ and $\mathbf{w}(\mathbf{s})$ from a certain index set I in the best match search, that is, using a distance function like

$$d(\mathbf{x}, \mathbf{x}') = \sum_{i \in I} (x_i - x'_i)^2, \tag{4}$$

the PSOM can be used as a "continuous associative memory" or as a "multi-map" tool, for example to unite the forward and inverse kinematics of a robot in one mapping [1, 4].

3 Derivation of smoothness metric

In this section, we derive a measure of the overall smoothness of the PSOM mapping. Hereto, we integrate the square sum of all second derivatives. The range of integration matches the allowed range for $\mathbf{s}$ — normally this is the hyper-rectangle spanned by the grid.
The different components of the mapping resp. weights are treated independently. Therefore, in the following, we can view the weights as one-dimensional.

$$\begin{aligned}
E(\{w\}) &= \int_\Omega \sum_{\mu,\nu} \left(\frac{\partial^2}{\partial s^\mu \partial s^\nu} w(\mathbf{s}) \right)^2 d^m\mathbf{s} & (5)\\
&= \sum_{\mu,\nu} \int \left(\sum_{\mathbf{g}} w_{\mathbf{g}} \frac{\partial^2}{\partial s^\mu \partial s^\nu} b_{\mathbf{g}}(\mathbf{s}) \right)^2 d^m\mathbf{s} & (6)\\
&= \sum_{\mathbf{g},\mathbf{h}} w_{\mathbf{g}} w_{\mathbf{h}} \sum_{\mu,\nu} \int \left(\frac{\partial^2}{\partial s^\mu \partial s^\nu} b_{\mathbf{g}}(\mathbf{s}) \right) \left(\frac{\partial^2}{\partial s^\mu \partial s^\nu} b_{\mathbf{h}}(\mathbf{s}) \right) d^m\mathbf{s} & (7)\\
&= \sum_{\mathbf{g},\mathbf{h}} w_{\mathbf{g}} w_{\mathbf{h}} \sum_{\mu,\nu} I^{\mu\nu}_{\mathbf{gh}} = \sum_{\mathbf{g},\mathbf{h}} w_{\mathbf{g}} w_{\mathbf{h}} M_{\mathbf{gh}} & (8)
\end{aligned}$$

Here, we used the definition

$$I^{\mu\nu}_{\mathbf{gh}} = \int \left(\frac{\partial^2}{\partial s^\mu \partial s^\nu} b_{\mathbf{g}}(\mathbf{s}) \right) \left(\frac{\partial^2}{\partial s^\mu \partial s^\nu} b_{\mathbf{h}}(\mathbf{s}) \right) d^m\mathbf{s}. \tag{9}$$

Since the basis functions $b_{\mathbf{g}}(\mathbf{s})$ are just products of one-dimensional polynomials, their derivatives are quite simple and products of one-dimensional polynomials as well. Denoting first and second derivatives by $'$ and $''$, we get

$$\frac{\partial^2}{\partial s^\mu \partial s^\nu} b_{\mathbf{g}}(\mathbf{s}) = \begin{cases} \prod_{\alpha \neq \mu} l^\alpha_g(s^\alpha) l^{\mu''}_g(s^\mu) & \mu = \nu \\ \prod_{\alpha \neq \mu,\nu} l^\alpha_g(s^\alpha) l^{\mu'}_g(s^\mu) l^{\nu'}_g(s^\nu) & \mu \neq \nu \end{cases} \tag{10}$$

To keep the notation compact, we omitted the double index of g, that is, l_g^α is to be read as an abbreviation of $l_{g^\alpha}^\alpha$. Further simplification of (9) requires a case distinction and to separate the terms depending on their dimension indices. For the first case ($\mu = \nu$) one gets

$$\begin{aligned} I_{\mathbf{gh}}^{\mu\mu} &= \int \prod_{\alpha \neq \mu} l_g^\alpha(s^\alpha) l_g^{\mu''}(s^\mu) \prod_{\beta \neq \mu} l_h^\beta(s^\beta) l_h^{\mu''}(s^\mu) d^m\mathbf{s} && (11) \\ &= \prod_{\alpha \neq \mu} \underbrace{\int l_g^\alpha(s^\alpha) l_h^\alpha(s^\alpha) ds^\alpha}_{\mathcal{A}_{gh}^\alpha} \underbrace{\int l_g^{\mu''}(s^\mu) l_h^{\mu''}(s^\mu) ds^\mu}_{\mathcal{B}_{gh}^\mu}, && (12) \end{aligned}$$

while the second case ($\mu \neq \nu$) yields

$$\begin{aligned} I_{\mathbf{gh}}^{\mu\nu} &= \int \prod_{\substack{\alpha \neq \mu \\ \alpha \neq \nu}} l_g^\alpha(s^\alpha) l_g^{\mu'}(s^\mu) l_g^{\nu'}(s^\nu) \prod_{\substack{\beta \neq \mu \\ \beta \neq \nu}} l_h^\beta(s^\beta) l_h^{\mu'}(s^\mu) l_h^{\nu'}(s^\nu) d^m\mathbf{s} && (13) \\ &= \prod_{\substack{\alpha \neq \mu \\ \alpha \neq \nu}} \underbrace{\int l_g^\alpha(s^\alpha) l_h^\alpha(s^\alpha) ds^\alpha}_{\mathcal{A}_{gh}^\alpha} \underbrace{\int l_g^{\mu'}(s^\mu) l_h^{\mu'}(s^\mu) ds^\mu}_{\mathcal{C}_{gh}^\mu} \underbrace{\int l_g^{\nu'}(s^\nu) l_h^{\nu'}(s^\nu) ds^\nu}_{\mathcal{C}_{gh}^\nu}. && (14) \end{aligned}$$

So, all we need to calculate are the symmetric matrices $\mathcal{A}, \mathcal{B}$ and $\mathcal{C}$ for each grid dimension. A convenient way to calculate the integrals is to first build a coefficient representation of the polynomials $l_g^\alpha(s^\alpha)$, which makes differentiation and integration very simple. The remaining work consists of collecting the proper summands for the matrix elements $M_{\mathbf{gh}} = \sum_{\mu,\nu} I_{\mathbf{gh}}^{\mu\nu}$. Of course, since the matrix $\mathbf{M}$ depends only on the placement of the nodes and not on the reference vectors, it has to be calculated only once for a given grid layout. Note that by construction $\mathbf{M}$ defines a symmetric positive semidefinite metric in the weight space.

4 Applying the smoothness metric

We can now use the metric $\mathbf{M}$ to tackle the problems stated in the Introduction, that is, we show how to make use of our smoothness measure $E(\{w\})$ to construct PSOM$^+$ mappings from noisy or non-grid-based data. We illustrate our approach on 1D toy data first.
Throughout the section, we drop the multi-index notation in favor of a binary representation of the indices. For a grid with N nodes, $\mathbf{M}$ thus becomes a $N \times N$ matrix, while the weights are represented by a single $N \times 1$ vector $\mathbf{w}$ for each dimension of the data space.

4.1 Noisy data

Suppose that the training data $\mathbf{w} = (w_g)$ is complete (each node has a reference vector), but noisy. For the case of Gaussian noise, this can be stated as

$$w_g = \hat{w}_g + u \qquad u \sim \mathcal{N}(0, \sigma). \tag{15}$$

Given an estimate of σ, what are the optimal ("true") weights $\hat{\mathbf{w}} = (\hat{w}_g)$? The stronger the noise, the smoother the PSOM$^+$ mapping should be, so a good strategy would be to set

$$\hat{\mathbf{w}} = \arg\min_{\hat{\mathbf{w}}} \left(\lambda E(\hat{\mathbf{w}}) + \|\mathbf{w} - \hat{\mathbf{w}}\|^2 \right) = \arg\min_{\hat{\mathbf{w}}} \left(\lambda \hat{\mathbf{w}}^T \mathbf{M} \hat{\mathbf{w}} + \|\mathbf{w} - \hat{\mathbf{w}}\|^2 \right) \tag{16}$$

$$\Leftrightarrow \quad \hat{\mathbf{w}} = (\lambda \mathbf{M} + \mathbf{I})^{-1}\mathbf{w}. \tag{17}$$

Here, λ acts as a regularization parameter that allows to balance the smoothness constraint ($E(\hat{\mathbf{w}})$) and the data constraint ($\|\mathbf{w} - \hat{\mathbf{w}}\|^2$). It is roughly proportional to the noise variance σ^2. Figure 2 shows the results on toy data for the case of a one-dimensional PSOM$^+$ with 8 nodes spaced at $a_i = i$, $(i = 1 \ldots 8)$. The right plot shows the mean square distance between de-noised and original weights as a function of the smoothing parameter.

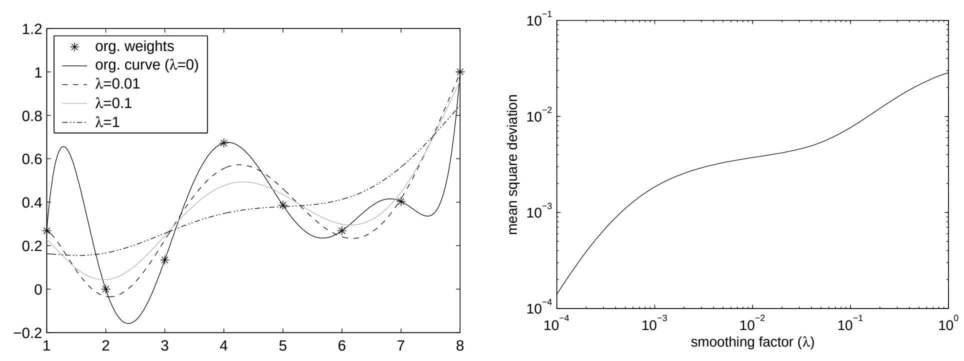


Figure 2: PSOM$^+$ from noisy data

4.2 Missing weights

Now suppose the training data to be free of noise, but instead incomplete. Given a set of indices to which the weights are known, and a complementary unknown set, what is now the optimal choice for the unknown weights ?
The less you know about a function, the simpler your estimate of it should be, so we choose the missing weights in a way that *maximizes smoothness*. Denoting sub-matrices (sub-vectors) of $\mathbf{M}$ and $\mathbf{w}$ by indices u (unknown) and k (known), the smoothness measure can be expressed as

$$E(\mathbf{w}) = \mathbf{w}_k^T \mathbf{M}_{kk} \mathbf{w}_k + \mathbf{w}_k^T \mathbf{M}_{ku} \mathbf{w}_u + \mathbf{w}_u^T \mathbf{M}_{uk} \mathbf{w}_k + \mathbf{w}_u^T \mathbf{M}_{uu} \mathbf{w}_u. \tag{18}$$

Its minimum with respect to $\mathbf{w}_u$ is given by

$$\mathbf{0} \overset{!}{=} \frac{\partial E(\mathbf{w})}{\partial \mathbf{w}_u} = 2\mathbf{M}_{uk}\mathbf{w}_k + 2\mathbf{M}_{uu}\mathbf{w}_u \tag{19}$$

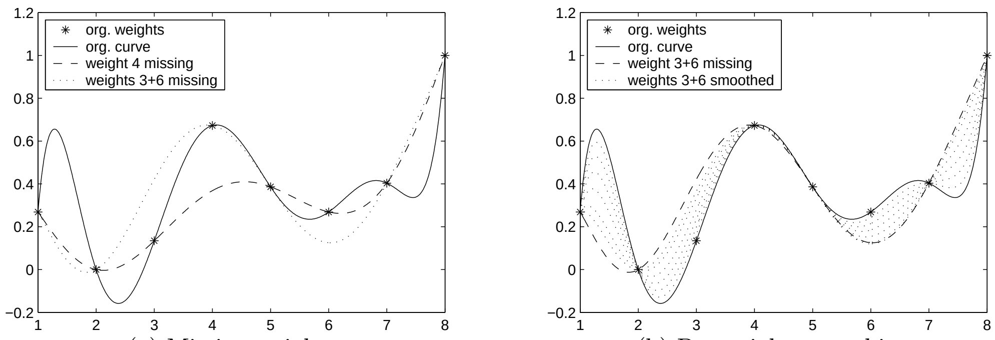
$$\mathbf{w}_u = -\left(\mathbf{M}_{uu}\right)^{-1} \mathbf{M}_{uk} \mathbf{w}_k \tag{20}$$

Figure 3a illustrates this method on a 1D PSOM$^+$ with 8 nodes.

4.3 Per-weight smoothing

If the scalar smoothing parameter λ in (17) is replaced by a diagonal matrix of per-weight smoothing parameters, the two problems described in the last sections merge into one. All exactly known weights would be assigned $\lambda = 0$, whereas missing weights would be interpreted as known, but infinitely noisy and therefore be endowed with a high value of λ.

$$\hat{\mathbf{w}} = (\mathbf{\Lambda} \mathbf{M} + \mathbf{I})^{-1}\mathbf{w} \qquad \mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots) \tag{21}$$



(a) Missing weights (b) Per-weight smoothing

Figure 3: PSOM$^+$ mappings from incomplete or partially noisy grid-organized data

Figure 3b shows an example using the same toy data as before. The weights 3 and 6 are treated as unknown or noisy and assigned a non-zero λ. All other weights are fixed ($\lambda = 0$). Note the "morphing" between the curves for the totally known and totally unknown case.

4.4 Non grid-organized training data

Suppose a set of training data given as N input-output pairs[2] $(\mathbf{s}_i, y_i)$ where the $\mathbf{s}_i$ do not match the nodes of a grid. To construct a PSOM$^+$ mapping from such data, we propose the following: First, specify a grid that spans a hyper-rectangle just large enough to embed the input data $\mathbf{s}_i$. If you have no idea how complex the mapping is, use as many nodes as is computationally feasible. The spacing of the nodes along the different axes is arbitrary.
Then, construct the smoothest mapping passing through the training data, that is

$$\text{minimize} \quad E(\mathbf{w}) \quad \text{subject to} \quad y_i \overset{!}{=} w(\mathbf{s}_i) = \sum_{\mathbf{g}} w_{\mathbf{g}} b_{\mathbf{g}}(\mathbf{s}_i) \quad i = 1 \ldots N. \tag{22}$$

By defining a matrix $\mathbf{B}$ with components $b_{ig} = b_g(\mathbf{s}_i)$, the constraints can be written as $\mathbf{y} = \mathbf{Bw}$. The optimization problem (22) only features linear equality constraints and thus can be solved by null-space methods. If the constraints are infeasible (e.g there are not enough nodes), one can use the pseudo-inverse of $\mathbf{B}$ to get an approximate solution.
In case the output data is noisy, it makes no sense to use the hard constraints of (22). Instead we proceed similar to section 4.1 and minimize a weighted sum of the smoothness measure and the distance between observed and reconstructed data:

$$\begin{aligned} \hat{\mathbf{w}} &= \arg\min_{\mathbf{w}} \left[\lambda E(\mathbf{w}) + \|\mathbf{y} - \mathbf{Bw}\|^2\right] & (23) \\ &= \arg\min_{\mathbf{w}} \left[\lambda \mathbf{w}^T \mathbf{M} \mathbf{w} + (\mathbf{y} - \mathbf{Bw})^T (\mathbf{y} - \mathbf{Bw})\right] & (24) \\ &= (\lambda \mathbf{M} + \mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T \mathbf{y} & (25) \end{aligned}$$

Figure 4 shows the resulting mappings of a 1D PSOM$^+$ with 8 nodes placed at $a_i = i = 1 \ldots 8$. The task was to interpolate 6 training data samples either (i) exactly or (ii) approximately by specification of a smoothing factor.

[2]Again, we treat the different (output) data dimensions separately.

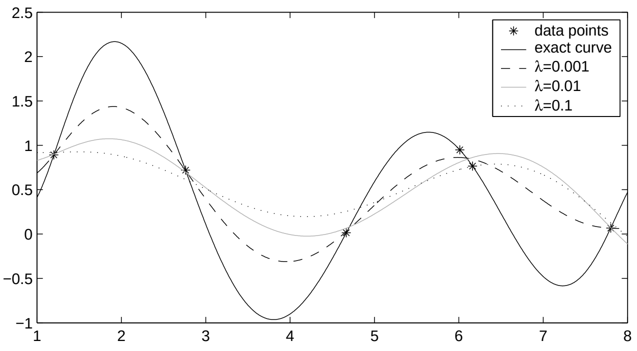


Figure 4: PSOM$^+$ from non-grid-organized data

	positional error in mm: mean (std dev)		
N_{miss}	**optimized**	local average	global average
40	**0.64 (0.45)**	7.53 (15.38)	15.65 (30.53)
100	**1.03 (1.18)**	25.86 (40.21)	56.29 (85.41)
400	**2.12 (2.21)**	62.90 (60.67)	140.32 (136.02)
0	0.54 (0.27)		

Table 1: PA10 wrist kinematics from missing data

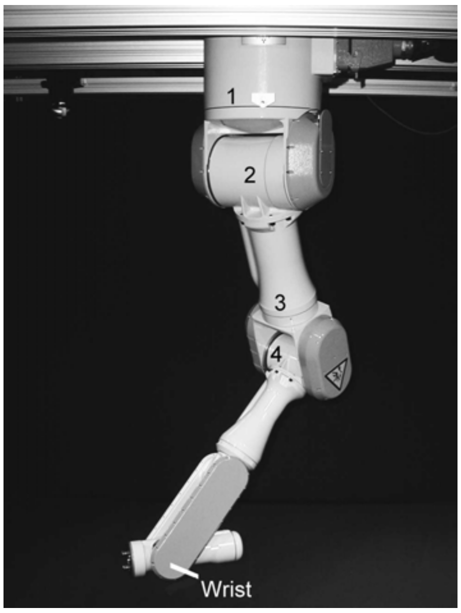


Figure 5: Mitsubishi PA10

5 Experiments

In this section, we demonstrate our PSOM extensions on a more complex dataset. We hereto chose a part of the forward kinematics of the Mitsubishi PA10, a 7 axis general purpose robot arm (see figure 5). Specifically, we simulated learning the 3D wrist position as a function of the first four joint angles. The PSOM$^+$ used is based on a 8x8x8x8 Chebyshev-spaced [8] grid, which means that the nodes are not placed regularly, but (proportionally) at the zeros of the 8th order Chebyshev polynomial. Each grid axis represents one joint and spans its respective range, so the manifold parameter $\mathbf{s}$ directly corresponds to the four joint angles.

As the basis for all experiments, we analytically calculated the “true” weights (wrist positions) for all nodes (joint angle sets). Furthermore, we generated 1000 random postures ($\mathbf{s}$) on which we compared the analytic forward kinematics to the result of the PSOM$^+$ mapping.

In our first experiment, we randomly selected $N_{miss} = 40$ (100, 400) nodes and treated their weights as missing. In addition to calculating optimal weights for these nodes (cf. section 4.2), we alternatively simply averaged (i) the neighboring weights and (ii) all weights to fill the gaps. Table 1 shows the resulting mean positional error on the set of 1000 test postures — our procedure is more than a magnitude more accurate than the two “naive” approaches.

As a second experiment, we added Gaussian noise of standard deviation $\sigma = 1$cm to all (analytically computed) weights. Then, we de-noised the PSOM$^+$ mapping with different values for λ (cf. section 4.1). Figure 6 shows the mean error on the test set, as well as the mean deviation between the original noisy and the de-noised weights.

In our third experiment, we randomly generated 2000 (3000, 4000) joint postures and their corresponding analytic forward kinematics and constructed a PSOM$^+$ mapping only from this (non-grid organized) training data, using the method of section 4.4. Table 2 depicts the mean error on the test set. Note that for 4000 training samples, the resulting mean error is about the same as for the "100 missing weights" case (cf. Table 1).

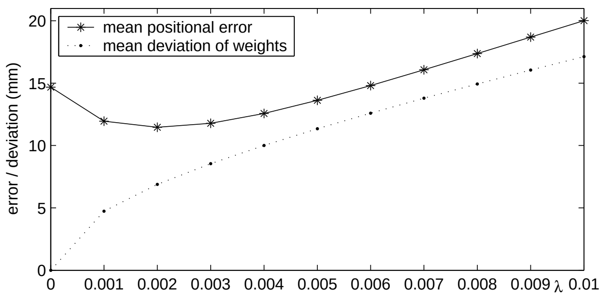


Figure 6: PA10 wrist kinematics from noisy data

Size of training set	positional error in mm mean (std dev)
2000	3.1 (5.7)
3000	1.6 (3.2)
4000	1.2 (2.7)

Table 2: PA10 wrist kinematics, non-grid-organized training data

6 Conclusion

We presented an approach to regularize PSOM mappings based on minimizing their overall smoothness. Our method allows to construct PSOMs from noisy and not necessarily grid-organized training data. As an application, we demonstrated the approach for learning the PA10 kinematics.

References

[1] Ruiz de Angulo V. and Torras C. Learning inverse kinematics via cross-point function decomposition. In *Proc. Int. Conf. on Artificial Neural Networks*, pages 856–864, 2002.

[2] A.C. Padoan Jr., G.A. Barreto, and A.F.R. Arajo. Modeling and production of robot trajectories using the temporal parametrized self-organizing map. *International Journal of Neural Systems*, 13(2):119–127, 2003.

[3] Teuvo Kohonen. *Self-Organizing Maps.* Springer, 1995.

[4] C. Nölker and H. Ritter. Parametrized SOMs for hand posture reconstruction. In S. I. Amari, C.L. Giles, M. Gori, and V. Piuri, editors, *Proc. Int. Joint Conf. on Neural Networks*, pages 139–144, 2000.

[5] Helge Ritter. Parametrized Self-Organizing Maps. In S. Gielen and B. Kappen, editors, *Proc. Int. Conf. Artificial Neural Networks*, pages 568–577, 1993.

[6] A. Saalbach, G. Heidemann, and H. Ritter. Parametrized SOMs for object recognition and pose estimation. In *Proc. Int. Conf. Artificial Neural Networks*, pages 902–907, 2002.

[7] Jörg Walter. PSOM network: Learning with few examples. In *Proc. Int. Conf. on Robotics and Automation*, pages 2054–2059, 1998.

[8] Jörg Walter and Helge Ritter. Local PSOMs and Chebyshev PSOMs – improving the Parametrised Self-Organizing Maps. In *Proc. Int. Conf. on Artificial Neural Networks, Paris*, volume 1, pages 95–102, 1995.