Improving Body Schema Learning with Kinematic Bézier Maps by Symmetry Constraints

Stefan Ulbrich and Tamim Asfour

***Abstract*— In robotics, the term *body schema* refers to the representation of the own body and its sensorimotor relations that are obtained by machine learning, ideally, from self-observation. It grants the robot a higher degree of autonomy as it allows the compensation of deformations of the body, for instance, those induced by holding a tool, without the need of a manually created explicit model. The forward kinematics is an important aspect of the body schema and the *Kinematic Bézier Maps* are a class of generic linear implicit models that can exactly encode it. Under ideal circumstances, the learned model can reliably predict the forward kinematics even for joint configurations that lie far away from those used for training (extrapolation). This does not hold, however, if the training samples lie very close together and the sensor signals are noisy—which is a common case when movement is limited when learning on a humanoid robot from pure self-observation. The model found by learning that minimizes the mean squared error (MSE) is then degenerated and not capable of extrapolation. This paper presents a solution to this problem by optimizing additional non-linear symmetry constraints in parallel to incremental learning. That way, the tolerance to noise can be increased and extrapolation be improved even under difficult conditions.**

I. INTRODUCTION

Learning forward sensorimotor maps—an important aspect of the body schema—is a supervised function regression process that creates an implicit model function of the non-linear sensorimotor relations from demonstrated examples of control signals (input) and sensor signals (output) during training. While many sensorimotor maps are important to robot control, such as dynamics and haptic, this work focuses on the forward kinematics only. Many different models and associated learning algorithms have been proposed for this task in the literature (see [1], [2] for surveys). Artificial neural networks, for instance, Kohonen maps [3], have been used successfully for robots with few degrees of freedom [4], [5], [6], [7] while, more recently, locally weighted learning techniques such as *receptive fields weighted regression (RFWR)* [8] and *Locally Weighted Projection Regression (LWPR* [9], and the related *Gaussian Mixture Regression (GMR)* [10] and the *eXtended Classifying System for Functions (XCSF)* [11] are currently the predominant methods. Their model function is a composition of linear models that partially approximate the latent function in the vicinity of observed training data. The model predicts values by evaluating their sum while each model is weighted by its distance to the query signal. Local learning usually is very fast but, on the downside, it generalizes only between samples it observed during training. To cover a large workspace of the robot generally requires an amount of linear models exponentially in the number of the robot's degrees of freedom (i.e., the number of input signals the output relates to). The algorithms mainly differ in how they create linear models. GMM and XCSF optimize their location and orientation, while LWPR reduces the local input dimensions with an incremental modification of the *Parial Least Squares* algorithm [12]. Another alternative for sensorimotor learning is *Gaussian Process Regression (GPR)* [13], [14]. In this probabilistic approach, model predictions are based on the similarity of the query input to the experienced training data. This measure of similarity is encoded in a covariance function which, for instance, results in a smooth approximation of the latent function when using the squared exponential kernel.

On the one hand, these models are very general an can be applied to any function regression problem. This means also, on the other hand, that a better performance is to be expected from more specialized methods. The recently presented *Kinematic Bézier Maps* [15], [16] are such models that are specialized to kinematic learning and permit exact encoding (although they can be modified for learning the inverse dynamics [17]). The models are global and can hence not only generalize between observed configurations but also make predictions for completely unknown configurations—given ideal training. This is the case, if during training, a wide range of joint configurations is observed, the signals are not very noisy, and a sufficient quantity of training data can be acquired. In reality and especially in the case of humanoid robots learning from self-observation, this is often not the case for the following reasons: *i.)* Self-observation is a constraint that limits possible movements as the robot's hand or tool center point (TCP) have to remain in the field of vision. *ii.)* The output signals used for learning (i.e., the position of the TCP) has to be reconstructed from stereo vision which is an error prone process. *iii.)* The arms of an anthropomorphic robot have many degrees of freedom and the latent function depends on many independent inputs as a consequence. The amount of observations required for learning exponentially grows with this number, a problem that is inherent in sensorimotor learning and is that referred to as *the curse of dimensionality*.

On a humanoid robot, it is consequently complicated to gather enough data to build a globally valid model. This may result in a degenerated implicit representation that prevents

This work was partially conducted in the project "Robots Exploring their Bodies Autonomously (REBA)" within the priority program "Autonomous Learning" founded by the German Research Foundation (DFG).

[1]S. Ulbrich, and T. Asfour are with the High Performance and Humanoid Technologies Lab (H^2T) at the Karlsruhe Institute of Technology (KIT), Germany [stefan.ulbrich,asfour] at kit.edu

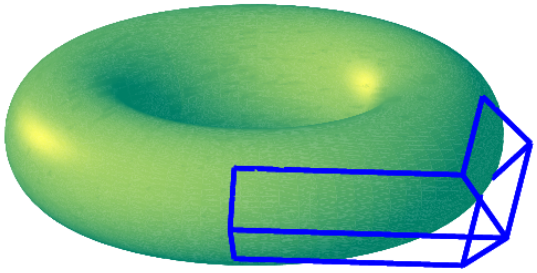

Fig. 1. The implicit KBM representation of the workspace of a simple robot with two revolute joints with perpendicular axis. The robot's workspace (all reachable positions) lies on the surface of a torus *(green)* and its implicit representation defined by the KBM forms a net of connected vertices *(blue)*.

accurate extrapolation. This problem can be addressed by decomposition techniques [18] and incremental refinement. In this work, we present a third option that compensates difficult training conditions to a certain degree. The KBM are geometric models and can be further restricted to kinematics by means of non-linear symmetry constraints that cannot be considered in the existing (linear) learning algorithms. In this work, their application in parallel to the incremental learning is presented on a theoretical basis.

This paper is structured as follows: The following section briefly presents the KBM and its learning algorithm. Afterwards, the conditioning during learning is presented and illustrated for low-dimensional cases. In the experiments section, a comparison between unconditioned and conditioned learning performed in simulation is presented. The document concludes with a summary and an outlook on future research.

II. Kinematic Bézier Maps

A. Learning the model

The KBM originate from the domain of projective geometry and are based on the observation that any conic, including circles, can be mapped onto a parabola by a projection [19]. That way, it is possible to parametrize the circular trajectories the TCP moves on when only one revolute joint is moving. In the KBM, this circular movement is represented by in a rational Bézier form. The whole kinematics—the product space of all rotations—can then be formulated the tensor product

$$f(\boldsymbol{\theta}) = \frac{\sum_{\boldsymbol{i}\in I^d} \boldsymbol{b_i} \cdot \gamma(\boldsymbol{i}) \cdot (B_{\boldsymbol{i}} \circ \tau)(\boldsymbol{\theta})}{\sum_{\boldsymbol{i}\in I^d} \gamma(\boldsymbol{i}) \cdot (B_{\boldsymbol{i}} \circ \tau)(\boldsymbol{\theta})}, \tag{1}$$

where

- d is the number of joints the forward kinematics depends on,
- $\boldsymbol{\theta}$ is the robot's joint configuration $(\theta_1, \ldots, \theta_d)$,
- $\boldsymbol{i} \in I^d$, is an index out of an index set $I^d = \{0, 1, 2\}^d$ that addresses the 3^d model parameters,
- $B_{\boldsymbol{i}}(t) = \prod_{i\in\boldsymbol{i}} B_i^2(t)$ are products of Bernstein polynomials,
- $\boldsymbol{b_i}$ are vertices of the control net—these are the model parameters determined by learning—,
- $\gamma_{\boldsymbol{i}}$ the projective weights of the control vertices,
- $\tau(\boldsymbol{\theta})$ is a tangent half-angle substitution that maps the joint angles to the necessary parameters where $= (\tau(\theta_1) \ldots \tau(\theta_d)$.

There is a-priori knowledge embedded in this formula that specializes it to kinematics: *i.)* The approximation with rational quadratic functions itself and *ii.)* the values for the weights $\gamma_{\boldsymbol{i}}$. For every three points with identical indices except for the j-th coordinate, that is,

$$\begin{aligned} \boldsymbol{i}_1 &= (\ldots, \underset{i}{0}, \ldots) \\ \boldsymbol{i}_2 &= (\ldots, \underset{i}{1}, \ldots), \\ \boldsymbol{i}_3 &= (\ldots, \underset{i}{2}, \ldots) \end{aligned} \tag{2}$$

it has to apply that the points form an isosceles triangle with a common angle α and that

$$\gamma_{\boldsymbol{i}_1} = \cos(\alpha) \cdot \gamma_{\boldsymbol{i}_0} = \cos(\alpha) \cdot \gamma_{\boldsymbol{i}_2} \tag{3}$$

holds. Only then, the resulting Bézier curve is a circular arc. That way, the only unknown parameters remaining in Eq. 1 are the control vertices $\boldsymbol{b_i}$ that appear linear in the equation. Learning consequently reduces to solving a system of linear equations. Exact batch learning can be achieved by *linear least squares* and incremental online learning by normalized least squares which is also known as *δ-rule* in machine learnig [20]. An example of a learned KBM is illustrated in Fig. 1 for a kinematics with—for the purpose of visualization—only two rotary degrees of freedom with orthogonal axes such that its workspace has the shape of a torus.

However, the first condition requiring the triangle α to be isosceles with a common angle cannot be expressed in linear terms. It can be neglected under good conditions but, if not, the circular trajectories degenerate to ellipses that produce a high extrapolation error (see Fig. 3). The novel conditioning presented in the following resolves this problem. Despite the complex mathematics, the model and its learn algorithms can be implemented and verified very easily.

B. Additional symmetry conditions, α the only parameter

The symmetry constraint that three neighboring control vertices with respect to their indices (see Eq. 2) have to form an isosceles triangles with a common angle α cannot be stated in a linear form. One possibility solution—without the need to modify the existing learning algorithms—is the parallel execution of the incremental online learning and a non-linear optimization enforcing the symmetry constraint. This optimization can be achieved with the *Gauss-Newton method*. The formulae for the isosceles constraint is derived first: For every control vertex $\boldsymbol{b_i}$, an error function $d(\boldsymbol{b_i})$ can be defined for which the minimizing control point's location

is searched for:

$$\arg\min_{\boldsymbol{b}_i} d(\boldsymbol{b_i}) = \boldsymbol{r_i}^T \cdot \boldsymbol{r_i} = \sum_{i=1}^{d} r_{i,i}^2, \tag{4}$$

where, using the following substitution for simplification,

$$\boldsymbol{a} := \boldsymbol{b}_{(\ldots,\underset{i}{0},\ldots)}, \qquad \boldsymbol{b} := \boldsymbol{b}_{(\ldots,\underset{i}{1},\ldots)}, \qquad \boldsymbol{c} := \boldsymbol{b}_{(\ldots,\underset{i}{2},\ldots)}$$

the mean of the squared sum of the residues $\boldsymbol{r_i}$

$$r_{\boldsymbol{i},i} = \|\boldsymbol{a} - \boldsymbol{b}\|^2 - \|\boldsymbol{b} - \boldsymbol{c}\|^2. \tag{5}$$

is minimized. Together with the Jacobian of $\boldsymbol{r_i}$

$$J_{\boldsymbol{i}} := \left(\frac{\partial r_{\boldsymbol{i},i}}{\partial b_{\boldsymbol{i},j}}\right)_{i,j} = \begin{cases} 2(a_j - b_j): & i_i = 0, \\ 2(c_j - a_j): & i_i = 1, , \\ 2(b_j - c_j): & i_i = 2, \end{cases} \tag{6}$$

the control vertex can be updated according to

$$\boldsymbol{b_i} = \boldsymbol{b_i} + \nu \cdot (J_{\boldsymbol{i}}^T \cdot J_{\boldsymbol{i}}) \cdot J^T \cdot \boldsymbol{r_i}, \tag{7}$$

where the factor $\nu \in (0, 1)$ regulates the step size.
In order to further ensure the that the common angle equals α, another minimization term with residuals $\bar{\boldsymbol{r}}_{\boldsymbol{i}}$ is defined:

$$\bar{r}_{\boldsymbol{i},i} = \|1/2(\boldsymbol{a} + \boldsymbol{c}) - \boldsymbol{b}\|^2 - 1/4 \tan^2(\alpha) \cdot \|1/2(\boldsymbol{a} - \boldsymbol{c})\|^2 \tag{8}$$

Analogously, the Jacobian can be derived,

$$\bar{J}_{\boldsymbol{i}} := \begin{cases} (a_j + c_j - 2b_j) - 1/2 \tan^2(\alpha)(a_j - c_j) & : i_i = 0, \\ 2 \cdot b_j - (a_j + c_j) & : i_i = 1, \\ (a_j + c_j - 2b_j) + 1/2 \tan^2(\alpha)(a_j - c_j) & : i_i = 2. \end{cases} \tag{9}$$

The concatenation of $\boldsymbol{r_i}, J$ and $\bar{\boldsymbol{r}}_{\boldsymbol{i}}, \bar{J}$ inserted in Eq. 7 yields the final formula of the optimization.

III. Evaluation

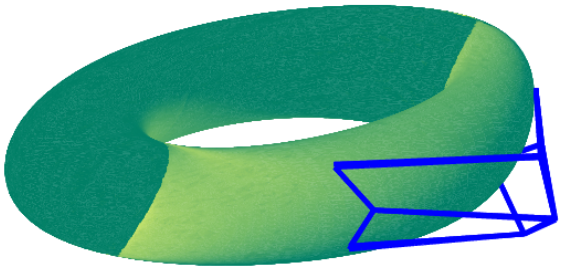

Fig. 2. Learning from noisy data with symmetry conditions. Despite the noise, the manifold that represents the expected workspace *(green)* still resembles the real workspace (torus) in Fig. 1 as the vertices of the KBM's control net *(blue)* form isosceles triangles.

We conducted two experiments to evaluate the conditioning of the incremental learning. They vary in the number of degrees of freedom the simulated kinematics used in the experiments. The first aims at the visualization of the differences and the second simulates a more lifelike application scenario.

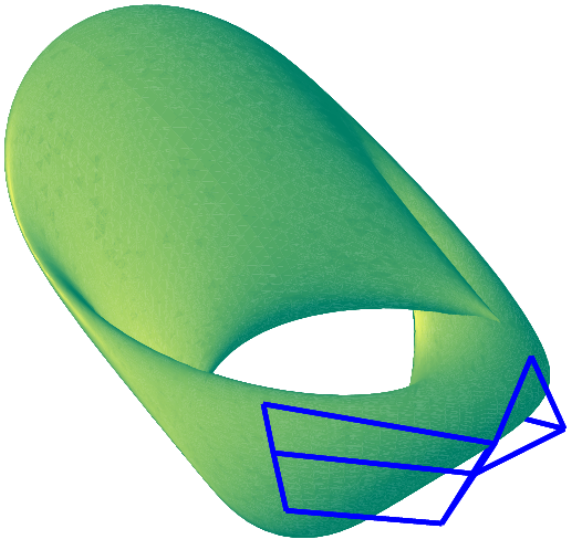

Fig. 3. Learning from noisy data without applying symmetry conditions. After learning from noisy data, the manifold that represents the expected workspace *(green)* is clearly deformed. The vertices of the KBM's control net *(blue)* do not form isosceles triangles.

A. Simulation with two degrees of freedom

In the first experiment, the same simulated kinematics used to produce Fig. 1 provided the training data for the incremental learning. Its first segment has a length of 200mm, its second is of length 75mm. The joint axes of its revolute joints are perpendicular, thus its work space is shaped like the surface of a torus. In addition, artificial noise with an standard deviation of 20mm is added to the position of the TCP. Two KBM models—only one of them fulfilling symmetry conditions—are trained with identical 30 training samples created from random joint configurations between $\pm 45°$. The models and their prediction of the complete workspace of the robot are depicted in Fig. 2 and Fig. 3. The first picture shows that the symmetry constraints leads to a good approximation of the complete workspace despite the noise. During learning, the δ-rule and the non-linear optimization were performed in parallel. It turned out that results were best for a smaller adaptation rate $\nu = 0.3$. Higher values gave a higher priority to maintaining the symmetry than the regression to the kinematics.
The second picture shows that the missing symmetry in the control net affects the extrapolation of the KBM. The manifold is deformed and has little in common with the real workspace of the robot. An important observation made during the experiments is that the errors over the training data is almost identical for both models. It can be concluded that the constraints during learning mainly affect the extrapolation of the model.

B. Simulation with five degrees of freedom

In this experiment, we used a simulated kinematics with five revolute joints with perpendicular axis and identical arm

TABLE I

RESULTS OF THE EXPERIMENTS WITH FIVE DEGREES OF FREEDOM.

Experiment		Constrained learning	Unconstrained learning	Batch learning	Expected (Noise)
Training data set	mean	129.98	136.80	75.135	79.526
	std	58.870	60.306	32.860	33.152
	median	120.84	132.80	71.055	78.182
	iqr	72.139	79.746	45.879	44.381
	max	457.20	374.74	197.48	240.97
Test data set (extrapolation)	mean	464.68	527.80	1742.8	—
	std	475.80	512.56	4854.5	—
	median	303.45	365.93	368.12	—
	iqr	407.22	462.46	1267.6	—
	max	3233.3	5024.5	6.8924e+04	—

elements of length 200mm. Again, we incrementally trained two KBM where only one used the constrained learning. The adaptation rate ν of the non-linear optimization has been set to 1.0 in this experiment as the step size decreases with the number of degrees of freedom. Training samples (800) were created using joint configurations with angles between $\pm 22.5^\circ$. Artificial noise with a standard deviation of 50mm simulates the difficult conditions expected on a real robot. After training, the models are evaluated against 600 configurations with joint angles between $\pm 90^\circ$ (without noise) and the statistics of the prediction error is computed (mean, standard deviation, median, interquartile range and maximum). Additionally, the models are compared against a KBM learned with unconstrained batch learning from the same training set. The results are shown in Table I. From the table, it can be concluded that the constrained learning performs best in terms of the extrapolation. The higher error (compared to batch learning) on the test set can be explained because of the slow final convergence rate of the δ-rule. The highest error on the extrapolation has the batch learning which can be explained by over fitting. This also explained why its errors on the training data is lower than the expected errors.

IV. CONCLUSION

In this paper, we presented a novel method to increase the performance of learning robot kinematics with KBM models under difficult conditions. By means of symmetry constraints enforced by a simultaneous non-linear optimization the extrapolation capabilities of the incremental learning can be drastically improved. That way, the algorithms can be better applied on robots learning their kinematics from pure self observation—especially humanoids.

Future research will include the investigation of the impact of the adaptation rates and the inclusion of the non-linear constraints into a batch learning algorithm.

REFERENCES

[1] M. Hoffmann, H. G. Marques, A. H. Arieta, H. Sumioka, M. Lungarella, and R. Pfeifer, "Body schema in robotics: A review," *IEEE T. Autonomous Mental Development*, vol. 2, no. 4, pp. 304–324, 2010.

[2] O. Sigaud, C. Salaun, and V. Padois, "On-line regression algorithms for learning mechanical models of robots: a survey," *Robotics and Autonomous Systems*, vol. 59, pp. 1115–1129, 2011.

[3] T. Kohonen, *Self-Organizing Maps*. Springer, 2001.

[4] J. Walter and H. Ritter, "Rapid learning with parametrized self-organizing maps," vol. 12, p. 131153, 1995.

[5] J. Walter, "PSOM network: learning with few examples," in *1998 IEEE International Conference on Robotics and Automation, 1998. Proceedings*, vol. 3, 1998, pp. 2054 –2059.

[6] S. Klanke and H. Ritter, "PSOM+: parametrized self-organizing maps for noisy and incomplete data," in *Proceedings of the 5th Workshop on Self-Organizing Maps (WSOM 05), Paris, France*, 2005.

[7] C. Gaskett and G. Cheng, "Online learning of a motor map for humanoid robot reaching," in *In Proceedings of the 2nd International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS) 2003*, 2003.

[8] S. Schaal and C. G. Atkeson, "Constructive incremental learning from only local information," *Neural Computation*, vol. 10, pp. 2047–2084, 1997.

[9] S. Vijayakumar and S. Schaal, "Locally weighted projection regression: An $O(n)$ algorithm for incremental real time learning in high dimensional space," in *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*, vol. 1, 2000, pp. 288–293.

[10] M. Lopes and B. Damas, "A learning framework for generic sensory-motor maps," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007. IROS 2007*, Nov. 2007, pp. 1533 –1538.

[11] P. O. Stalph, J. Rubinsztajn, O. Sigaud, and M. V. Butz, "A comparative study: Function approximation with LWPR and XCSF," in *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*, 2010, pp. 1863–1870.

[12] l. E. Frank and J. H. Friedman, "A statistical view of some chemometrics regression tools," *Technometrics*, vol. 35, no. 2, pp. 109–135, 1993.

[13] C. Hartmann, J. Boedecker, O. Obst, S. Ikemoto, and M. Asada, "Real-time inverse dynamics learning for musculoskeletal robots based on echo state gaussian process regression," in *Proceedings of Robotics: Science and Systems*, Sydney, Australia, 2012.

[14] J. S. de la Cruz, W. Owen, and D. Kulic, "Online learning of inverse dynamics via gaussian process regression," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2012, pp. 3583 –3590.

[15] S. Ulbrich, V. Ruiz de Angulo, T. Asfour, C. Torras, and R. Dillmann, "Rapid learning of humanoid body schemas with kinematic bzier maps," in *9th IEEE-RAS International Conference on Humanoid Robots, 2009. Humanoids 2009*, Dec. 2009, pp. 431 –438.

[16] ——, "Kinematic Bézier Maps," *IEEE Trans. Man, Systems & Cybernectics, Part B*, vol. 42, no. 4, pp. 1215–1230, Aug. 2012.

[17] S. Ulbrich, M. Bechtel, T. Asfour, and R. Dillmann, "Learning robot dynamics with Kinematic Bézier Maps," in *Proc. IEEE/RSJ Intl. Conf. Int. Robots, Syst. (IROS)*, Vilamoura, Portugal, Oct. 2012.

[18] S. Ulbrich, V. Ruiz de Angulo, T. Asfour, C. Torras, and R. Dillmann, "General robot kinematics decomposition without intermediate markers," *IEEE Trans. Neural Networks & Learning Systems*, vol. 23, no. 4, pp. 620–630, Apr. 2012.

[19] G. E. Farin, *NURBS: from projective geometry to practical use*. A.K. Peters, 1999.

[20] B. Widrow and M. Hoff, "Adaptive switching circuits." IRE, 1960, pp. 96–104.