Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Technology 00 (2016) 000–000

Procedia Technology

www.elsevier.com/locate/procedia

3rd International Conference on System-integrated Intelligence: New Challenges for Product and Production Engineering, SysInt 2016

Hybrid Mechanical and Data-driven Modeling Improves Inverse Kinematic Control of a Soft Robot

René Felix Reinhart*, Jochen Jakob Steil

Research Institute for Cognition and Robotics – CoR-Lab, Bielefeld University, Universitätsstr. 25, 33615 Bielefeld, Germany

Abstract

Feed-forward control relies on accurate knowledge about the controlled plant, e.g. models of manipulator kinematics or dynamics. However, for many plants, mechanical models do not capture all aspects of a plant or the plant's intrinsic properties, e.g. soft materials, do hardly allow for exact and efficient mechanical modeling. In this context, machine learning is a suitable technique to extract non-linear plant models from data. The paper shows that feed-forward control based on inversion of a hybrid forward model comprising a mechanical model and a learned error model can significantly improve accuracy. The proposed approach is demonstrated for inverse kinematic control of a redundant soft robot with a hybrid model that is constructed from continuum kinematics together with an efficient neural network error model.

Keywords: Machine Learning, Learning Feed-forward Control, Soft Robot, Inverse Kinematics

1. Introduction

Progress in intelligent automation systems and novel products often requires improved models for control. However, often the systems are too complex to represent or derive their action-effect relationships in simple models. Plant parameters may be time varying, or the models demand a high computational effort which makes model-based control infeasible. In this context, machine learning is a suitable technique to extract non-linear plant models from data.

Non-linear regression techniques have previously been applied to learn inverse models for feed-forward control, e.g. [1–3]. One difficulty of these approaches is that supervised training data comprising pairs of desired effects and appropriate actions for the inverse model learning need to be collected. While the acquisition of supervised data pairs is achieved in [2,3] by relying on low-gain feedback control, exploration of inverse models has been addressed more generally e.g. in [4,5]. However, in many cases rough mechanical forward models are available and also data for parameter identification is typically acquired in the modeling process. In these cases, it is often more effective to design a hybrid forward model based on approximate mechanics plus an error model that captures complex characteristics like friction, material properties, etc. In contrast to previous work on the combination of inverse models based on

* Corresponding author. Tel.: +49-521-106-5147
E-mail address: freinhart@cor-lab.uni-bielefeld.de, jsteil@cor-lab.uni-bielefeld.de

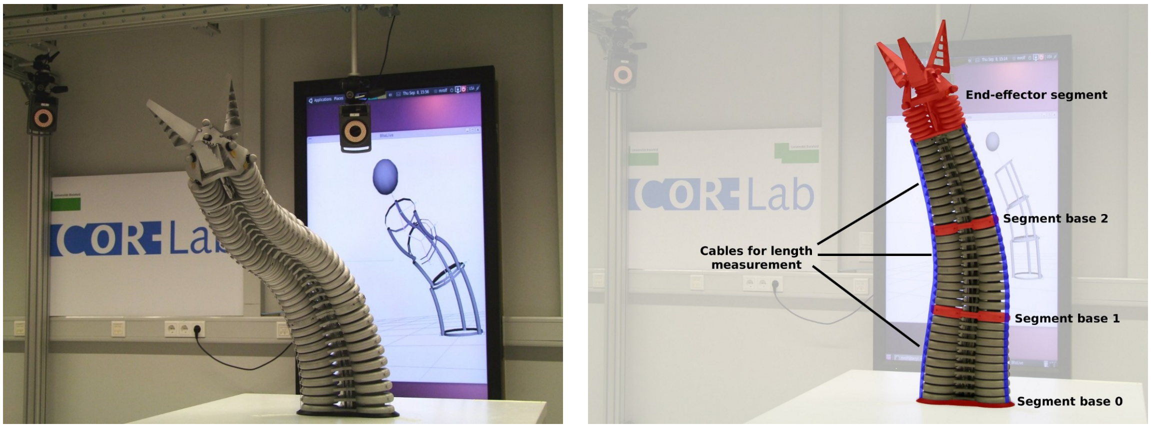

Fig. 1. Left: The Bionic Handling Assistant (BHA) is a pneumatically actuated, soft robot trunk manufactured by Festo. Right: The BHA comprises three main segments with three air chambers each and a gripper segment. The length expansion of air chambers is measured by cables running along the outer hull of each chamber.

mechanics plus learned error models, e.g. in [6], this paper focuses on the learning of hybrid forward models, which are then inverted for feed-forward control.

The paper shows that feed-forward control based on inversion of a hybrid forward model comprising a mechanical model and a learned error model can significantly improve accuracy. In particular, it is systematically pointed out that the combination of a model based on the plant mechanics plus an error model learned from data has certain advantages compared to learning the entire forward model: a) Restricting learning to the error of the mechanical model reduces the complexity of the to be learned model. b) Non-linearities captured by the mechanical model reduce the training data demand for learning the error model. c) The feed-forward control law is a combination of the inverted mechanical model, which mostly features robust extrapolation, and a black-box error model, which can be restricted to a certain degree such that unintended extrapolation artifacts of the learned error model are mitigated.

The paper demonstrates the proposed approach for inverse kinematic control of a redundant soft robot trunk (see Fig. 1), where a hybrid model is constructed from constant curvature continuum kinematics [7,8] together with an efficient neural network error model. The accuracy of the hybrid model is compared to the continuum kinematics model and an entirely learned forward kinematic model. Feed-forward control of the end effector is then achieved by inversion of the hybrid model as in Jacobian-based inverse kinematics solving for redundant robots. The paper closes with general implications of the hybrid feed-forward control scheme for improving controller design in automation systems and intelligent products.

2. Bionic Handling Assistant (BHA)

The Bionic Handling assistant (BHA, see Fig. 1) has been designed by Festo as a robotic pendant to an elephant trunk. The BHA has gathered strong interest because it belongs to a new class of continuum soft and lightweight robots based on additive manufacturing with polyamide [9]. The robot trunk is pneumatically actuated and comprises three main segments. Each segment consists of three triangular arranged air chambers. Therefore the main flexibility of the BHA is based on nine air chambers that extend their length in relation to the pressure in those chambers. A fourth end effector segment is also available but was neglected for this work. The robot has no classical, revolute joint angles. Instead each robot segment starts to bend in case that the three chambers reach different lengths. Beside pressure sensors integrated in the air valves, the BHA is equipped with cable potentiometers that allow to measure the outer length of the air chambers providing geometric information about the robot shape. The segments of the BHA together with the attached cable length sensors are depicted in Fig. 1 (right).

While the challenge of controlling the segments lengths by supplying the air chambers with appropriate pressures has been addressed in previous work [10,11], this paper focuses on the kinematic control of the end effector. The autonomous exploration of a direct inverse model for controlling the end effector has been accomplished in [5] where the main emphasis was to start model learning from very limited prior knowledge about the plant. In this paper, the idea is to apply the established scheme for solving inverse kinematics for redundant robots by local inversion of the forward model [12]. We start control design from an approximate model of the robot's forward kinematics and improve this model by learning an error model based on motion capture data.

2.1. *Constant curvature model of the forward kinematics*

In previous work, a kinematic model of the BHA was developed [7][1]. The forward kinematic model of the BHA follows the constant curvature approach that combines torus segments in order to represent continuous deformations. Fig. 1 (right) shows how the three actuators in each main segment cause a deformation between two rigid segment bases (shown in red). The three measured lengths of these actuators can be used to estimate the coordinate transformation between two platforms, which are then chained in order to get the complete forward kinematics from base to end effector. The only free parameters of the segment model are the segment radii. These radii have been estimated according to a best-fit solution by recording ground-truth data from different movements [7].

We denote the robot configuration by $\mathbf{q} \in \mathbb{R}^9$, which comprises three length measurements for each segment. The kinematic forward model based on the constant curvature approach is denoted by $\mathbf{x} = f^{cnk}(\mathbf{q})$, where we consider only end effector positions $\mathbf{x}$ in this paper for simplicity.

2.2. *Differential inverse kinematics*

The kinematic model allows 3D visualization of the robot (see Fig. 1 and Fig. 2 (left)) and also inverse kinematic control. To derive an inverse kinematics solver, we consider the differential relation between robot configurations $\mathbf{q}$ and end effector positions $\mathbf{x}$:

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \text{ where } J_{ij}(\mathbf{q}) = \frac{\partial f_i(\mathbf{q})}{\partial q_j}$$

are the entries of the Jacobian $\mathbf{J}(\mathbf{q})$ of the forward model $f(\mathbf{q})$. By inversion of this relation, we obtain the classical differential inverse kinematics control law

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}(\mathbf{q})\dot{\mathbf{x}},$$

where $\mathbf{J}^{\dagger}(\mathbf{q})$ is the pseudo-inverse of $\mathbf{J}(\mathbf{q})$ and $\dot{\mathbf{x}}$ is the desired end effector velocity. To accommodate additional constraints into the motion generation for redundant robots, e.g. to avoid joint limits, stay close to a preferred robot configuration and to avoid non-cyclic motions in configuration space, this scheme is typically extended by a term for motion generation in the null-space of the primary task:

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}(\mathbf{q})\dot{\mathbf{x}} + \left(\mathbb{1} - \mathbf{J}^{\dagger}(\mathbf{q})\mathbf{J}(\mathbf{q})\right)\mathbf{z} \tag{1}$$

In this paper, we use $\mathbf{z} = \beta(\mathbf{q}^{pref} - \mathbf{q})$ to select a redundancy solution close to a preferred robot configuration $\mathbf{q}^{pref}$.

2.3. *Error of the constant curvature kinematic model*

The assumptions made by the constant curvature model f^{cnk} are violated by the real plant [7,8]. Therefore, the plant kinematics and the kinematic model mismatch. This mismatch and calibration errors [7] result in systematic errors of the forward model and hence reduces the accuracy of the inverse kinematic controller (1). Fig. 2 (right) shows a histogram of errors

$$||\mathbf{x} - f^{cnk}(\mathbf{q})||$$

[1] The constant curvature model is available as open source `http://www.cor-lab.org/software-continuum-kinematics-simulation`.

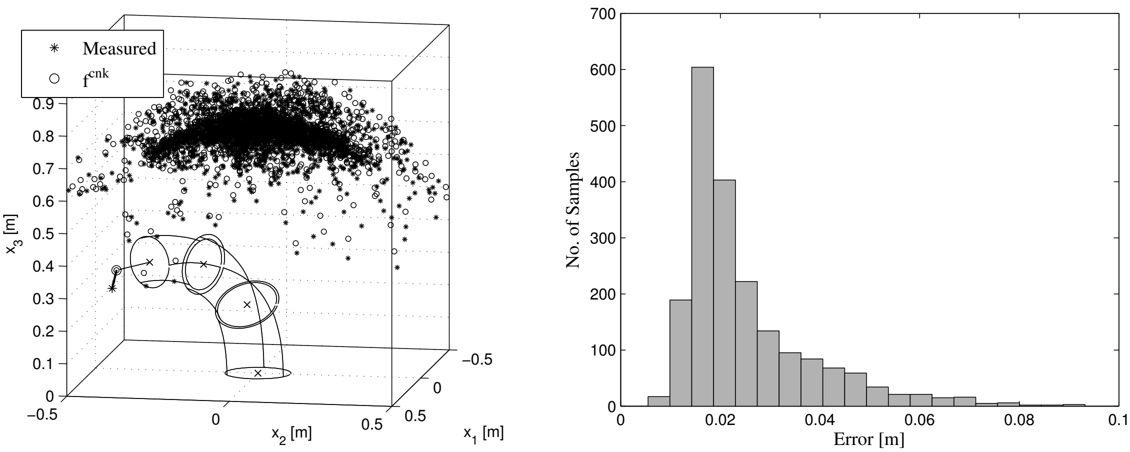

Fig. 2. Left: Data set with actual end effector positions measured by a VICON system and estimated end effector positions according to the constant curvature kinematics f^{cnk}. The deviation of the kinematic model and the measured end effector positions is shown for an exemplary robot configuration. Right: Error histogram for the constant curvature kinematics f^{cnk}.

between end effector positions $\mathbf{x}$ measured by a VICON system [13] and end effector positions estimated by the forward model $f_{cnk}(\mathbf{q})$ for a set of 2000 configurations $\mathbf{q}$. The histogram in Fig. 2 (right) has a heavy tail, which reveals that the error of the forward model is considerable. The mean error is 2.5 cm and the maximal error is 9.3 cm. Note that the accuracy of the inverse kinematic controller (1) is directly affected by the accuracy of the forward kinematic model. Hence, to improve the accuracy of end effector control, it is essential to improve the forward model.

3. Data-driven and hybrid forward models

In order to improve the accuracy of the forward kinematic model, we compare multiple, data-driven and hybrid modeling approaches: Firstly, we consider the pure data-driven modeling that fits a flexible function approximator $f^{dat}(\mathbf{q})$ to the collected data in order to predict end effector positions $\mathbf{x}$ from robot configurations $\mathbf{q}$. While this first approach has to capture the complete non-linearities of the forward kinematics, the second approach restricts learning to the error of the continuum kinematics model $f^{cnk}(\mathbf{q})$. For the hybrid model

$$f^{hyb}(\mathbf{q}) = f^{cnk}(\mathbf{q}) + \epsilon(\mathbf{q}) \tag{2}$$

solely the configuration-dependent error $\epsilon(\mathbf{q})$ of the mechanical model $f^{cnk}(\mathbf{q})$ is learned.

3.1. Linear models and Extreme Learning Machines

For learning the completely data-driven model $f^{dat}(\mathbf{q})$ or the error model $\epsilon(\mathbf{q})$, we apply two learning algorithms:

- **Linear Model**: A linear model of the form $f(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}$ is trained by linear regression as a baseline.
- **Extreme Learning Machine** (ELM, [14]): Extreme Learning Machines are feed-forward neural networks with a single hidden layer and an efficient training scheme. The output of the network is computed according to

  $$f(\mathbf{x}) = \mathbf{W}^{\text{out}}\sigma(\mathbf{W}^{\text{inp}}\mathbf{x} + \mathbf{b}),$$

  where $\sigma(a) = 1/(1 + exp(-a))$ is a sigmoid activation function applied component-wise to the synaptic summations $\mathbf{a} = \mathbf{W}^{\text{inp}}\mathbf{x} + \mathbf{b}$. In comparison to classical training of feed-forward neural networks by gradient descent through error backpropagation, ELMs restrict learning to the connections $\mathbf{W}^{\text{out}} \in \mathbb{R}^{dim(f(\mathbf{x}))\times H}$ from the hidden layer to the outputs. The number of hidden neurons H is chosen large in comparison to the number of inputs. The input weights $\mathbf{W}^{\text{inp}} \in \mathbb{R}^{H\times dim(\mathbf{x})}$ and biases $\mathbf{b} \in \mathbb{R}^H$ are initialized randomly and remain fixed. In this paper,

the components of $\mathbf{W}^{\text{inp}}$ and $\mathbf{b}$ are drawn from a uniform distribution with range $[-1, 1]$. The idea of using feed-forward neural networks with a random hidden layer has been proposed earlier, e.g. in [15].
Learning aims at minimization of the sum of squared errors with regularization term

$$\frac{1}{2}\sum_{n=1}^{N}(\mathbf{t}_n - \mathbf{W}^{\text{out}}\boldsymbol{\sigma}(\mathbf{W}^{\text{inp}}\mathbf{x}_n + \mathbf{b}))^2 + \frac{\lambda}{2}\|\mathbf{W}^{\text{out}}\|^2, \tag{3}$$

where $\mathbf{t}_n$ is the target output for input $\mathbf{x}_n$. Minimization of (3) with respect to $\mathbf{W}^{\text{out}}$ is a convex optimization problem, which can be solved efficiently by linear regression. The optimal output weight matrix is given by

$$\mathbf{W}^{\text{out}} = (\mathbf{H}^T\mathbf{H} + \lambda\mathbb{1})^{-1}\mathbf{H}^T\mathbf{T},$$

where $\mathbf{H}$ and $\mathbf{T}$ collect the hidden neuron activations $\boldsymbol{\sigma}(\mathbf{a}_n)$ and target outputs $\mathbf{t}_n$ for all $n = 1, \ldots, N$ samples row-wise. The regularization parameter $\lambda > 0$ allows for the continuous selection of the model complexity.

3.2. Learning of pure data-driven models and error models

The pure data-driven model $f^{dat}(\mathbf{q})$ is trained with the recorded data $\{(\mathbf{q}_n, \mathbf{x}_n)\}_{n=1,\ldots,N}$ comprising pairs of arm configurations $\mathbf{q}_n$ (lengths of the air chambers) as inputs and the end effector positions $\mathbf{x}_n$ measured by the VICON system as target outputs. The hybrid model (2), in contrast, is trained with pairs $\{(\mathbf{q}_n, \mathbf{e}_n)\}_{n=1,\ldots,N}$ comprising arm configurations $\mathbf{q}_n$ as inputs and errors $\mathbf{e}_n = \mathbf{x}_n - f^{cnk}(\mathbf{q}_n)$ of the mechanical forward model f^{cnk} as target outputs. Hence, the learning algorithm identifies the error model $\epsilon(\mathbf{q})$ from (2) in the latter case.

3.3. Generalization of the learned models

In this section, the ability of the different data-driven modeling approaches to generalize the learned models to new inputs, i.e. robot configurations, is compared.

We evaluate the accuracy of the data-driven approaches on a set of 2000 configurations $\mathbf{q}$ of the robot. To better cover the kinematic effects of each segment and reduce demand for data recording, configurations were sampled randomly per segment first. For each segment, resulting end effector positions have been measured when the three length values for this segment are positioned to 500 random configurations within the admissible length ranges. The remaining segment lengths were kept approximately constant in this scheme. These 1500 data pairs are complemented with 500 additional random configurations for the entire robot, i.e. configurations $\mathbf{q}$ are drawn from a nine-dimensional, uniform distribution within the admissible length ranges. The measured end effector positions are shown in Fig. 2 (left) for all 2000 samples together with a full random configuration of the robot in simulation.

For evaluation of the generalization ability, the available data is split into a training set comprising all 1500 samples with segment-wise random configurations and 400 samples with completely random configurations. The remaining 100 samples with completely random robot configurations are used as test set. Training is repeated five times for the ELM with $H = 100$ hidden neurons and $\lambda = 0.1$ in order to account for the random initialization of $\mathbf{W}^{\text{inp}}$ and $\mathbf{b}$. Errors are evaluated according to

$$\|\mathbf{x} - f^{\square}(\mathbf{q})\|$$

Table 1. Average training and test errors together with standard deviations and maximal errors for f^{dat} and f^{hyb} using Linear Models or Extreme Learning Machines for fitting.

Model	Learning Algorithm	Training Error [cm]	Test Error [cm]
Data-driven model f^{dat}	Linear Model	3.65 ± 3.97 (44)	9.8 ± 7.8 (43)
Data-driven model f^{dat}	Extreme Learning Machine	0.95 ± 1.17 (13)	3.4 ± 2.7 (21)
Hybrid model f^{hyb}	Linear Model	0.81 ± 0.65 (6.1)	1.8 ± 0.8 (4.4)
Hybrid model f^{hyb}	Extreme Learning Machine	0.57 ± 0.56 (6.5)	1.3 ± 0.05 (3.6)

where $f^{\square}$ is either the purely data-driven model f^{dat} or the hybrid model f^{hyb}.

Table 1 shows the average errors for f^{dat} and f^{hyb} when using a Linear Model or an Extreme Learning Machine for fitting. The results confirm the expectation that it is more difficult to learn the complete forward kinematics in a single model (f^{dat}). Because of the non-linear forward kinematics, the Linear Model is not suitable for this task and underfits the data (large training errors in the first row of Table 1). While the ELM can model non-linearities, generalization of f^{dat} then easily suffers from over-fitting (large test errors in the second row of Table 1). The hybrid models f^{hyb} in contrast achieve a significant improvement compared to the pure mechanical model f^{cnk} while generalizing well also to novel robot configurations. In fact, a relatively accurate error model $\epsilon(\mathbf{q})$ can already be achieved with a Linear Model (cf. third and fourth row in Table 1). This shows that the inaccurate mechanical forward model significantly reduces the complexity (non-linearity) of learning the error model $\epsilon(\mathbf{q})$. However, non-linearities still remain for the error model and the best result is achieved with the non-linear ELM in the hybrid model.

4. Inverse kinematics with a hybrid forward model

This section finally shows how to exploit the hybrid forward model with ELM learner to solve inverse kinematics. According to (1), computation of the manipulator Jacobian $\mathbf{J}(\mathbf{q})$ of the hybrid forward model $f^{hyb}(\mathbf{q})$ is required.

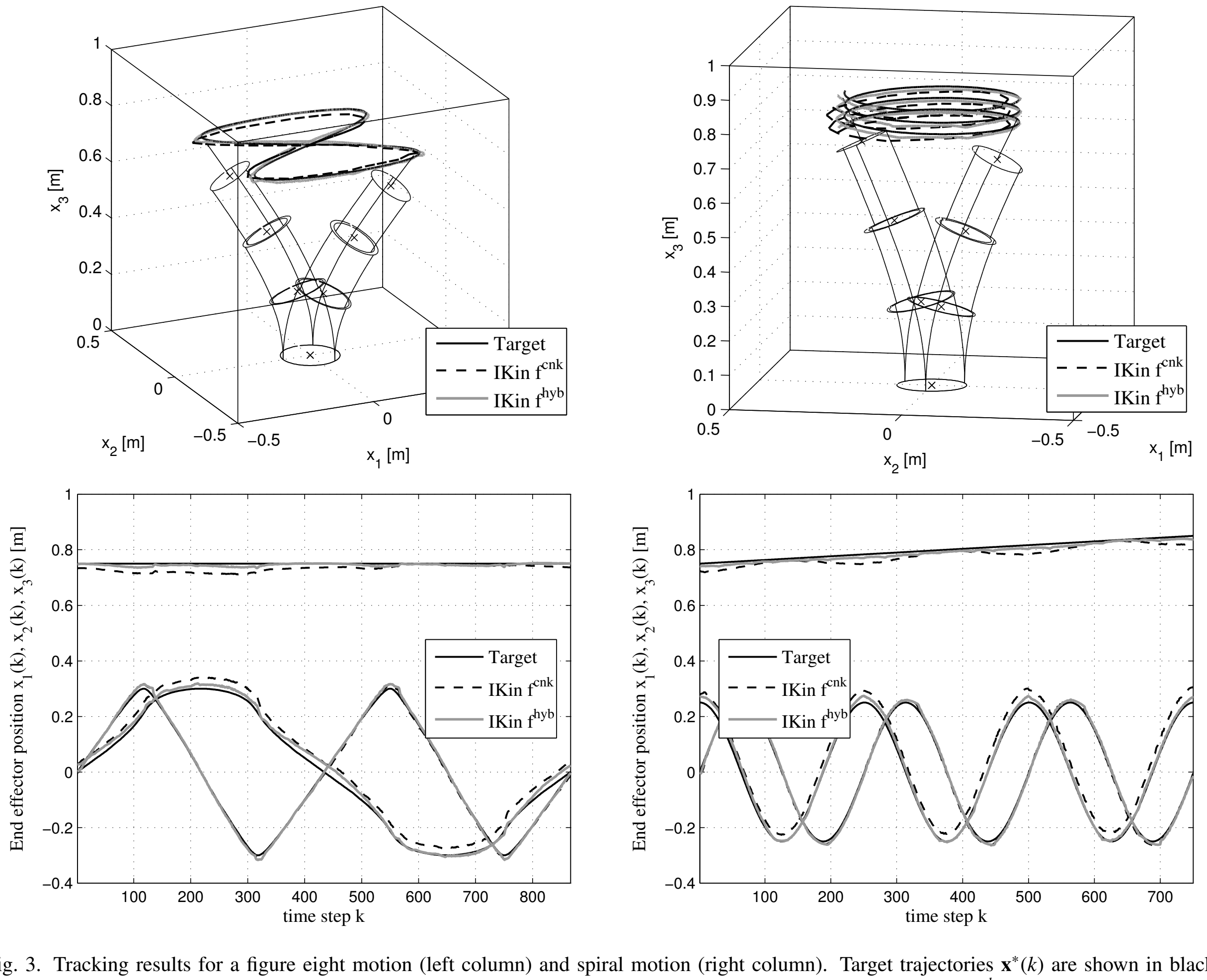

Fig. 3. Tracking results for a figure eight motion (left column) and spiral motion (right column). Target trajectories $\mathbf{x}^*(k)$ are shown in black, tracked end effector positions $\mathbf{x}(k)$ are recorded with VICON. Tracking results using (1) with continuum kinematics f^{cnk} are shown by dashed lines and for the hybrid model f^{hyb} by solid gray lines. The first row shows the trajectory in task space and the bottom row displays the task space trajectories over time steps k.

The entries

$$J_{ij}(\mathbf{q}) = \frac{\partial f_i^{hyb}(\mathbf{q})}{\partial q_j} = \frac{\partial f_i^{cnk}(\mathbf{q})}{\partial q_j} + \frac{\partial \epsilon_i(\mathbf{q})}{\partial q_j},$$

of the Jacobian now comprise a sum of partial derivatives of the mechanical forward model $f^{cnk}(\mathbf{q})$ and the learned error model $\epsilon(\mathbf{q})$. While the derivatives of the mechanical model $f^{cnk}(\mathbf{q})$ are computed numerically in this work, the partial derivatives of the error model $\epsilon(\mathbf{q})$ with ELM learner are computed analytically by

$$\frac{\partial \epsilon_i(\mathbf{q})}{\partial q_j} = \sum_{n=1}^{H} w_{in}^{out} w_{nj}^{inp} \sigma'_n \left(\sum_{k=1}^{dim(\mathbf{q})} w_{nk}^{inp} q_k + b_n \right),$$

where the derivative $\sigma'(x) = \sigma(x)(1 - \sigma(x))$ of the sigmoid function $\sigma(x) = 1/(1 + exp(-x))$ is efficient to compute.

We demonstrate the inverse kinematic control on the real BHA. We apply the inverse kinematics solver (1) with continuum kinematics f^{cnk} only and with the hybrid model (2) for two exemplary trajectories (see Fig. 3). The hybrid model achieves significantly more accurate tracking results (see Fig. 4). The overall average tracking error is reduced from 4.16 cm using only f^{cnk} to 1.67 cm using the hybrid model f^{hyb}. The latter result is close to the accuracy that can be achieved with this platform [5,7]. The effective change of the trajectory in configuration space (lengths of the air chambers) due to the learned error model is clearly visible in the right panel of Fig. 4. Note that the change of the trajectory in configuration space is not due to a drift along a redundancy manifold during inverse kinematics solving, because we set a preferred configuration $\mathbf{q}^{pref}$ (straight robot posture with medium elongation) as null-space constraint in (1). Note also that the trajectories in configuration space with the hybrid model do not display undesired discontinuities during target tracking. Inversion of the hybrid model results in well-behaved and accurate motion generation.

5. Conclusion

While the methods for classical parameter identification and machine or statistical learning approaches have methodological similarities, the main difference is the flexibility of the models used in machine learning. Many of these models are able to fit any function with arbitrary accuracy. However, in practice the selection of a suitable model complexity is the key to achieve accurate generalization to areas of the input space where no data was available for training. This paper shows that the clever combination of restricted models from a domain, e.g. constant curvature kinematics, plus a more flexible machine learning technique that models the error can significantly improve the final

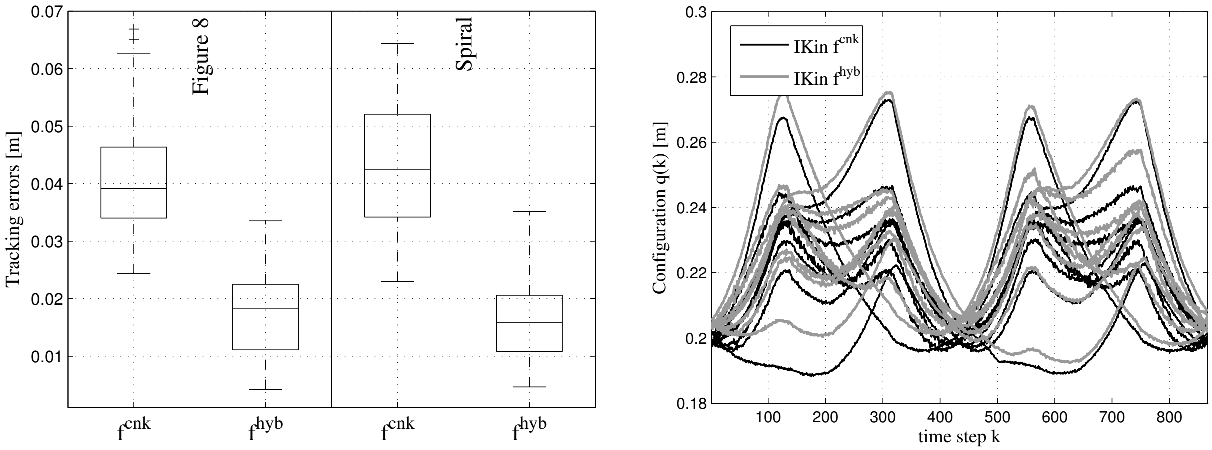

Fig. 4. Left: Tracking error statistics $\|\mathbf{x}^*(k) - \mathbf{x}(k)\|$ for both models and trajectories. Right: Trajectory in configuration space (lengths of the air chambers) for the figure eight motion over time steps k.

model accuracy. This additional model learning can be accomplished with data that is typically recorded for parameter identification or model verification anyway. The results in this paper therefore imply an additional, data-driven step in the modeling procedure that can significantly improve the final model by fully exploiting the information captured in the data. Such an integration of classical modeling approaches and novel data-driven methods on a system level also requires closer integration across disciplines in the future.

Acknowledgements

This research and development project is funded by the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Cluster Competition and managed by the Project Management Agency Karlsruhe (PTKA). The authors thank Jeffrey Queißer for technical support, in particular for recording the data set.

References

[1] Velthuis, W., de Vries, T., van Amerongen, J.. Learning feedforward control of a flexible beam. In: IEEE International Symposium on Intelligent Control. 1996, p. 103–108. doi:10.1109/ISIC.1996.556185.

[2] Schaal, S., Atkeson, C.G., Vijayakumar, S.. Scalable Techniques from Nonparametric Statistics for Real Time Robot Learning. Applied Intelligence 2002;17(1):49–60. doi:10.1023/A:1015727715131.

[3] Nguyen-Tuong, D., Seeger, M., Peters, J.. Computed torque control with nonparametric regression models. In: American Control Conference. 2008, p. 212–217. doi:10.1109/ACC.2008.4586493.

[4] Rolf, M., Steil, J., Gienger, M.. Online goal babbling for rapid bootstrapping of inverse models in high dimensions. In: IEEE International Conference on Development and Learning (ICDL); vol. 2. 2011, p. 1–8. doi:10.1109/DEVLRN.2011.6037368.

[5] Rolf, M., Steil, J.. Efficient exploratory learning of inverse kinematics on a bionic elephant trunk. IEEE Transactions on Neural Networks and Learning Systems 2014;25(6):1147–1160. doi:10.1109/TNNLS.2013.2287890.

[6] Um, T., Park, M.S., Park, J.M.. Independent Joint Learning: A novel task-to-task transfer learning scheme for robot models. In: IEEE International Conference on Robotics and Automation (ICRA). 2014, p. 5679–5684. doi:10.1109/ICRA.2014.6907694.

[7] Rolf, M., Steil, J.. Constant curvature continuum kinematics as fast approximate model for the Bionic Handling Assistant. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2012, p. 3440–3446. doi:10.1109/IROS.2012.6385596.

[8] Mahl, T., Hildebrandt, A., Sawodny, O.. A variable curvature continuum kinematics for kinematic control of the bionic handling assistant. IEEE Transactions on Robotics 2014;30(4):935–949. doi:10.1109/TRO.2014.2314777.

[9] Grzesiak, A., Becker, R., Verl, A.. The Bionic Handling Assistant: A success story of additive manufacturing. Assembly Automation 2011;31(4):329–333. doi:10.1108/01445151111172907.

[10] Neumann, K., Rolf, M., Steil, J.J.. Reliable Integration of Continuous Constraints into Extreme Learning Machines. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 2013;21:35–50. doi:10.1142/S021848851340014X.

[11] Rolf, M., Neumann, K., Queier, J., Reinhart, R., Nordmann, A., Steil, J.. A multi-level control architecture for the bionic handling assistant. Advanced Robotics 2015;29(13):847–859. doi:10.1080/01691864.2015.1037793.

[12] Spong, M., Vidyasagar, M.. Robot Dynamics and Control. John Wiley & Sons; 1989.

[13] VICON Motion Tracking Systems. Last date accessed: 2016/02/09. URL: http://www.vicon.com.

[14] Huang, G.B., Zhu, Q.Y., Siew, C.K.. Extreme learning machine: a new learning scheme of feedforward neural networks. In: IEEE International Joint Conference on Neural Networks; vol. 2. 2004, p. 985–990. doi:10.1109/IJCNN.2004.1380068.

[15] Schmidt, W., Kraaijveld, M., Duin, R.. Feedforward neural networks with random weights. In: IAPR International Conference on Pattern Recognition, Vol.II. Conference B: Pattern Recognition Methodology and Systems. 1992, p. 1–4. doi:10.1109/ICPR.1992.201708.