

Learning Coupled Forward-Inverse Models with Combined Prediction Errors

Dorothea Koert¹, Guilherme Maeda², Gerhard Neumann³ and Jan Peters^{1,4}

Abstract—Challenging tasks in unstructured environments require robots to learn complex models. Given a large amount of information, learning multiple simple models can offer an efficient alternative to a monolithic complex network. Training multiple models—that is, learning their parameters and their responsibilities—has been shown to be prohibitively hard as optimization is prone to local minima. To efficiently learn multiple models for different contexts, we thus develop a new algorithm based on expectation maximization (EM). In contrast to comparable concepts, this algorithm trains multiple modules of paired forward-inverse models by using the prediction errors of both forward and inverse models simultaneously. In particular, we show that our method yields a substantial improvement over only considering the errors of the forward models on tasks where the inverse space contains multiple solutions.

I. INTRODUCTION

In robotics, good models are crucial for planning, control and prediction. In such contexts, mostly two types of models are considered: forward models, which compute predictions for state changes and sensory signals; and inverse models, which are able to map sensory signals to appropriate actions [1], [2], [3]. Traditionally, models have been designed analytically or by hand from experts. While this design process works well for constrained environments, unexpected situations and more complex robots require more adaptive and richer models. Such models will need to be learned on demand from training data [4].

Supervised learning of models for prediction and control has long attracted the attention of the sensorimotor learning and robot learning communities. In particular, locally weighted methods [5], [6] and Gaussian Processes [7], [8] have gained popularity. The sensorimotor literature presents biologically inspired concepts for learning of coupled forward-inverse models [9], [1] and their implications for robot control. These concepts of coupled forward-inverse models have been applied to various tasks such as a framework for social interaction and learning from demonstration [10], [11]. Additionally, they have been extended in a hierarchical fashion [12]. However, learning those forward and inverse models is often hard [13] in particular for multi-valued inverse functions.

Opposed to the forward mapping, where usually a unique solution for any given input exists, learning inverse models

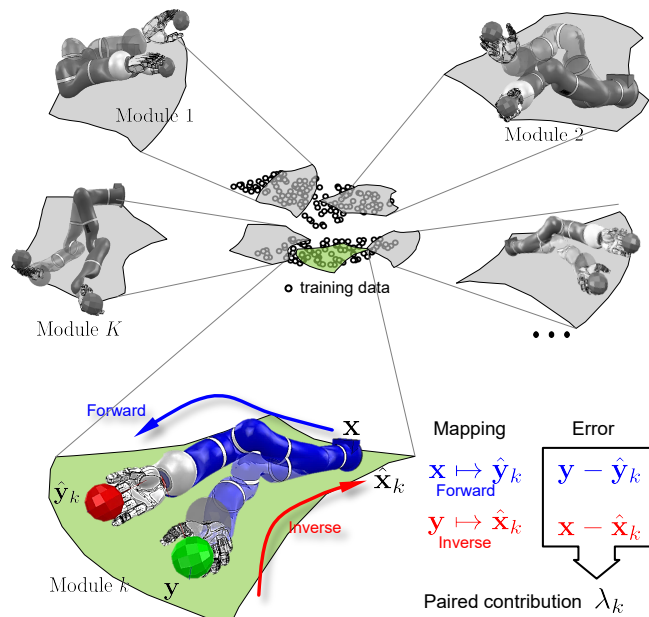


Fig. 1. Learning multiple local models from training data can enable robots to adapt to a multitude of contexts (depicted in gray). In this paper we introduce a new algorithm to simultaneously learn multiple coupled modules, each consisting of a forward model (blue) and an inverse model (red) and their paired contribution for different contexts, namely responsibilities. In contrast to existing methods, we utilize both forward and inverse model errors to train model parameters and their responsibilities.

can be problematic due to frequently occurring one-to-many mappings. This ill-posedness is reflected in robotics, for example, in the computation of the inverse kinematics functions of robot arms.

In this paper, we learn paired forward-inverse models in the context of robotics. In particular, we tackle the problem of better localizing the multiple models over a multi-valued space of solutions (such as the inverse kinematics). The proposed algorithm is inspired by MOSAIC [1], but extends the original concept of modular control to the case of general multi-valued function approximations. To this end, we present a new objective function and the corresponding EM optimization scheme. The proposed new objective function combines the prediction error of both forward and inverse models, which is used to compute the responsibilities of each model for different situations. Empirical evaluations in simulation and on a real robot demonstrate that this new algorithm can learn a better localization for forward and inverse models resulting in better models compared to the original idea from [1] of only using the forward modeling

¹TU Darmstadt, Germany

koert,peters@ias.tu-darmstadt.de

²ATR Computational Neuroscience Labs, Kyoto, Japan

g.maeda@atr.jp

³University of Lincoln, UK gneumann@lincoln.ac.uk

⁴MPI Tuebingen, Germany peters@mpi.org

errors during training. Summarizing, the contribution of this paper is a supervised learning algorithm that learns forward and inverse models, which can cover multiple inverse solutions for multiple coupled forward-inverse models.

II. BACKGROUND AND RELATED WORK

Multiple studies, in particular experiments on the adaptation of humans to changing force fields, resulted in the concept of internal models for representation of the human body and interactions with the environment [14], [15], [16]. Inspired by those biological insights, a number of approaches can be found in robotics and neuroscience literature, that are aiming for implementation of human-like learning by generating multiple internal models [17], [1], [3], [18], [19].

A direct approach for learning inverse models (referred to as “direct inverse modeling” [9]) consists in sampling input commands \mathbf{x} (motor/joint) while recording the resulting output \mathbf{y} (e.g. as end-effector forces/positions). By reversing the inputs and outputs, the mapping $\mathbf{y} \rightarrow \mathbf{x}$ can be trained via supervised learning. Although simple, this approach presents a few drawbacks. Not only must enough inputs be sampled to cover the region in which the desired targets are expected to lie, but a solution may not be found due to the one-to-many mapping.

Distal learning [9] learns a forward model to provide prediction errors. The task prediction error is propagated back through the forward model, generating an input error that can be used to learn the inverse model. While distal learning seems to be useful to learn ill-posed mappings, it is characterized by the use of a single inverse and forward model. Single, monolithic models can be problematic if they do not have sufficient model capacity to capture the complexity of the real system. Also, they must be retrained for a different task or context due to the lack of modularity.

Wolpert and Kawato introduced modularity in the learning of coupled forward-inverse models with the framework of MOSAIC [1]. The basic ideas of MOSAIC are comparable to Jacobs’ mixture of experts [20]. Modularity in the learning of models is appealing for robot control as the learned modules can be seen as motor primitives. In contrast, learning a large amount of information with a single network is challenging because new concepts can destroy what has previously been learned [21]; modularity alleviates such problem. Adaptation to new scenarios is very efficient, by simply recomputing the responsibilities of each module. MOSAIC is intrinsically an on-line algorithm and leverages on the concept that for a redundant system (a many-to-one mapping in the forward direction), a feedback controller realizes one specific input. MOSAIC then uses this feedback, in the form of a control error, to train the inverse model, which is an inherited feature from feedback-error-learning [22].

However, in the training phase MOSAIC relies only on the forward model errors to learn the decentralized gating of the models. Figure 2 shows the main structure of MOSAIC during the training of multiple models, where the information flow is depicted by the solid black arrows. In particular, if the inverse space may contain additional information for

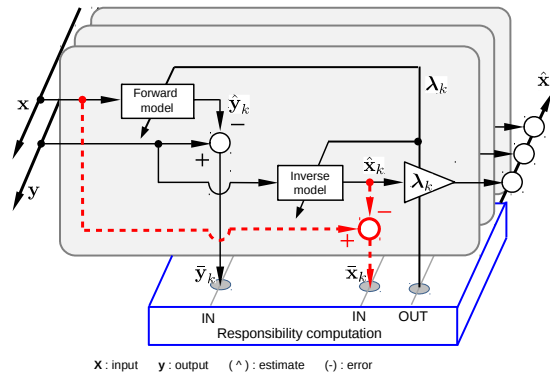


Fig. 2. In the original MOSAIC the responsibilities λ_k of the k -th module are exclusively trained on the prediction error of its forward model $\bar{\mathbf{y}}_k$. To achieve better localization of the multiple inverse models, we propose additionally incorporating the inverse prediction errors $\bar{\mathbf{x}}_k$ on the computation of the responsibilities (dashed red lines). Experiments show that the individual inverse errors $\bar{\mathbf{x}}_k$ allow for a better distribution of models even under ill-posed mappings.

localization of single models (as for multi-valued inverse functions), the forward error might not be sufficient to localize the models in a good way. Instead, a better localization of models could be achieved if the error of the inverse model is also taken into account during the learning of the gating function.

III. COUPLED FORWARD-INVERSE MODEL LEARNING

In this section, we present our new algorithm for Coupled Forward-Inverse Model Learning (CFIM). In the first part of this section, we show how to learn multiple coupled forward and inverse models through expectation maximization (EM). The EM algorithm [23], [24] is a common way to learn maximum likelihood solutions for models with latent variables. Formalizing the responsibilities of mixture components as latent variables for EM is also a well-known approach [25], [26], [27], [26].

We extend the objective, that was used in MOSAIC and also similarly in [28], to additionally minimize the error of the inverse models when updating the responsibilities in the E-Step. In the second part of this section, we derive the formulas for an application with Gaussian linear models, which we also use for experimental evaluation.

A. EM For Multiple Coupled Forward-Inverse Models

We consider a data set $\mathcal{D} = \{\mathbf{X}, \mathbf{Y}\}$ with input data $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^T$ and output data $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N]^T$. In contrast to existing approaches, CFIM learns the forward and inverse model parameters and responsibilities not only from errors of the forward models but also from errors of the inverse models. To minimize these combined model errors, we maximize a combination of the conditional log-

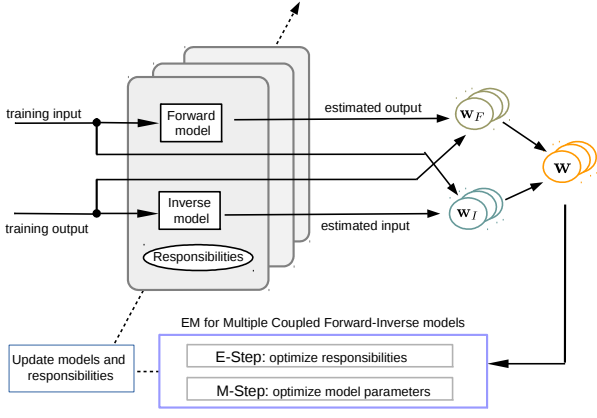


Fig. 3. We formalize the problem of learning multiple coupled forward-inverse models as an Expectation Maximization algorithm, where in the E-step we learn model responsibilities on a combination of forward and inverse modeling errors \mathbf{w}_F and \mathbf{w}_I . In the M-step, we update the model parameters of inverse and forward models through weighted regression based on the new responsibilities.

likelihoods over the data set

$$\begin{aligned} & \log \left(\prod_{n=1}^N p(\mathbf{x}_n | \mathbf{y}_n, \Theta_{I,k}) \right) + \log \left(\prod_{n=1}^N p(\mathbf{y}_n | \mathbf{x}_n, \Theta_{F,k}) \right) \\ &= \sum_{n=1}^N \log (p(\mathbf{x}_n | \mathbf{y}_n, \Theta_{I,k}) p(\mathbf{y}_n | \mathbf{x}_n, \Theta_{F,k})), \end{aligned} \quad (1)$$

where $\Theta_{F,k}$ denotes the parameters of the k -th forward model and $\Theta_{I,k}$ the parameters of the k -th inverse model. We formulate the problem of learning the parameters of K coupled forward-inverse models as a probabilistic model with observable variables $\mathbf{x}_n, \mathbf{y}_n$ and latent variables \mathbf{z}_n , where the latent variables hold information about responsibilities of the models for particular data points. For each observable data point $\{\mathbf{x}_n, \mathbf{y}_n\}$, the latent variable \mathbf{z}_n is a K -dimensional vector which represents the model assignment using the one-hot encoding. We incorporate the latent variables in Eq. (1) and introduce a distribution over the latent variables $q(z_{nk})$ to derive an EM algorithm. Hereby, we obtain the following optimization problem for the E-step:

$$\begin{aligned} & \arg \max_{q(z_{nk})} \sum_{n=1}^N \sum_{k=1}^K q(z_{nk}) \log \left(\frac{p(\mathbf{x}_n, z_{nk} | \mathbf{y}_n, \Theta_{I,k}) p(\mathbf{y}_n, z_{nk} | \mathbf{x}_n, \Theta_{F,k})}{q(z_{nk})} \right) \\ & \text{s.t.} \quad \sum_{k=1}^K q(z_{nk}) = 1. \end{aligned}$$

From here, we derive a closed form solution using the method of Lagrangian multipliers for the distribution over the latent variables, namely the responsibilities of the coupled models

$$q^*(z_{nk}) = \frac{p(\mathbf{x}_n, z_{nk} | \mathbf{y}_n, \Theta_{I,k}) p(\mathbf{y}_n, z_{nk} | \mathbf{x}_n, \Theta_{F,k})}{\sum_{k=1}^K p(\mathbf{x}_n, z_{nk} | \mathbf{y}_n, \Theta_{I,k}) p(\mathbf{y}_n, z_{nk} | \mathbf{x}_n, \Theta_{F,k})}. \quad (2)$$

The subsequent M-step computes weighted maximum likelihood solutions for the model parameters $\Theta_{I,k}$ and $\Theta_{F,k}$

respectively by solving

$$\arg \max_{\Theta} \sum_{n=1}^N \sum_{k=1}^K q^*(z_{nk}) \log \left(\frac{p(\mathbf{x}_n, z_{nk} | \mathbf{y}_n, \Theta_{I,k}) p(\mathbf{y}_n, z_{nk} | \mathbf{x}_n, \Theta_{F,k})}{q^*(z_{nk})} \right)$$

We show how to derive closed-form solutions for Gaussian linear models in the next section. Figure 3 illustrates our new algorithm.

B. Application for Gaussian Linear Models

In this subsection we derive solutions of our new algorithm for the case of Gaussian linear models.

We introduce an augmented transformation of the inputs $\tilde{\mathbf{x}}_n = [\mathbf{1}, \mathbf{x}_n]^T$ and outputs $\tilde{\mathbf{y}}_n = [\mathbf{1}, \mathbf{y}_n]^T$. We assume K components, each consisting of a forward model, which predicts $\hat{\mathbf{y}}_{k,n}$ and an inverse model, which predicts $\hat{\mathbf{x}}_{k,n}$

$$\hat{\mathbf{y}}_{nk} = \mathbf{A}_k \tilde{\mathbf{x}}_n \quad \text{and} \quad \hat{\mathbf{x}}_{nk} = \mathbf{B}_k \tilde{\mathbf{y}}_n, \quad (3)$$

where \mathbf{A}_k and \mathbf{B}_k are matrices that represent the model parameters of the k -th forward and inverse model, respectively.

For Gaussian linear models we obtain the likelihoods

$$\begin{aligned} p(\mathbf{y}_n | z_{nk}, \mathbf{x}_n, \Theta_{F,k}) &= \mathcal{N}(\mathbf{y}_n | \hat{\mathbf{y}}_{nk}, \Sigma_{F,k}) =: w_{F,nk} \quad \text{and} \\ p(\mathbf{x}_n | z_{nk}, \mathbf{y}_n, \Theta_{I,k}) &= \mathcal{N}(\mathbf{x}_n | \hat{\mathbf{x}}_{nk}, \Sigma_{I,k}) =: w_{I,nk}. \end{aligned} \quad (4)$$

We define $\Theta_{F,k} := \{\mathbf{A}_k, \Sigma_{F,k}\}$, $\Theta_{I,k} := \{\mathbf{B}_k, \Sigma_{I,k}\}$, where $\Sigma_{F,k}$ denotes the covariance of the k -th forward model and $\Sigma_{I,k}$ denotes the covariance of the k -th inverse model.

In the *E-step*, we calculate the responsibilities λ_{nk} of component k for the n -th data point by using uninformed priors in Eq. (2) which leads to

$$\lambda_{nk} = \frac{p(\mathbf{x}_n | z_{nk}, \mathbf{y}_n, \Theta_{I,k}) p(\mathbf{y}_n | z_{nk}, \mathbf{x}_n, \Theta_{F,k})}{\sum_z p(\mathbf{x}_n | z_{nk}, \mathbf{y}_n, \Theta_{I,k}) p(\mathbf{y}_n | z_{nk}, \mathbf{x}_n, \Theta_{F,k})}. \quad (5)$$

In the *M-step*, we update the model parameters of the forward and the inverse models using a weighted maximum likelihood estimate

$$\begin{aligned} \mathbf{A}_k^{\text{new}} &= (\tilde{\mathbf{X}}^T \Lambda_k \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^T \Lambda_k \mathbf{Y}, \\ \mathbf{B}_k^{\text{new}} &= (\tilde{\mathbf{Y}}^T \Lambda_k \tilde{\mathbf{Y}})^{-1} \tilde{\mathbf{Y}}^T \Lambda_k \mathbf{X}, \\ \tilde{\mathbf{X}} &= [\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_N]^T, \quad \tilde{\mathbf{Y}} = [\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2, \dots, \tilde{\mathbf{y}}_N]^T, \\ \Lambda_k &= \text{diag}(\boldsymbol{\lambda}_k), \\ \Sigma_{k,\text{fwd}}^{\text{new}} &= \frac{\sum_{n=1}^N \lambda_{n,k} (\mathbf{y}_n - \mathbf{A}_k^{\text{new}} \tilde{\mathbf{x}}_n) (\mathbf{y}_n - \mathbf{A}_k^{\text{new}} \tilde{\mathbf{x}}_n)^T}{\sum_{n=1}^N \lambda_{n,k}}, \\ \Sigma_{k,\text{inv}}^{\text{new}} &= \frac{\sum_{n=1}^N \lambda_{n,k} (\mathbf{x}_n - \mathbf{B}_k^{\text{new}} \tilde{\mathbf{y}}_n) (\mathbf{x}_n - \mathbf{B}_k^{\text{new}} \tilde{\mathbf{y}}_n)^T}{\sum_{n=1}^N \lambda_{n,k}}. \end{aligned} \quad (6)$$

To initialize the model parameters we compute the range of input and output variables as $\Delta_F = (\max(\mathbf{Y}) - \min(\mathbf{Y}))$ and $\Delta_I = (\max(\mathbf{X}) - \min(\mathbf{X}))$. The initial models are uniformly distributed over this range which results in the initial model matrices

$$\begin{aligned} \mathbf{A}_k &= \left[\mu(\mathbf{Y}) + (k-1) \frac{0.9\Delta_F}{K-1} - 0.45\Delta_F, \mathbf{0} \right], \\ \mathbf{B}_k &= \left[\mu(\mathbf{X}) + (k-1) \frac{0.9\Delta_I}{K-1} - 0.45\Delta_I, \mathbf{0} \right], \end{aligned} \quad (7)$$

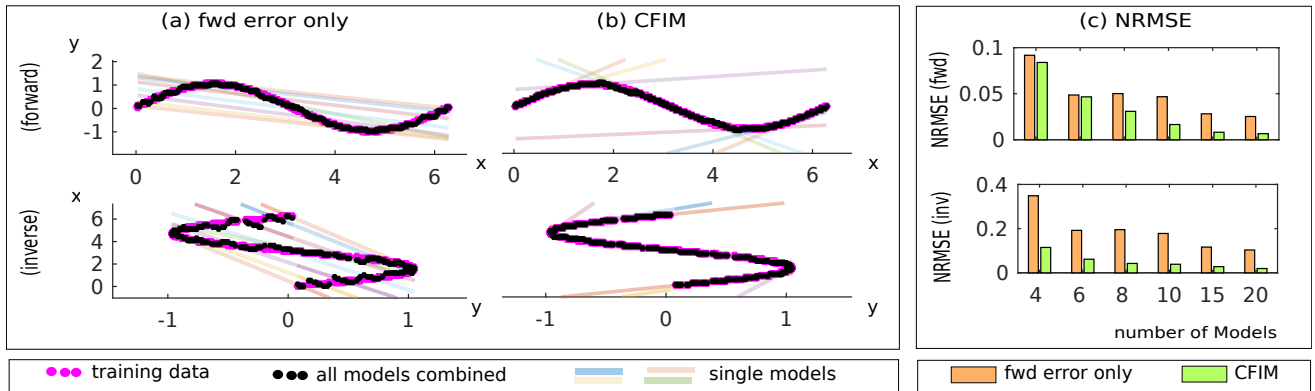


Fig. 4. We compare training responsibilities of the models only on the forward modeling errors (as proposed by MOSAIC) to CFIM, where we train responsibilities on both forward and inverse modeling errors. (a) When using only the forward error no accurate inverse function is learned for the approximation of the sine curve. (b) CFIM yields significant improvement in the approximation of the inverse function and also results in better localization of the single models. (c) The NRMSE for different numbers of models on the training data shows that CFIM achieves better results.

Algorithm 1 CFIM

Input: $\mathcal{D} = \{\mathbf{X}, \mathbf{Y}\} = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_N, \mathbf{y}_N)\}$
Initialize $\mathbf{A}_k, \mathbf{B}_k, \Sigma_{k,F}, \Sigma_{k,I}$, according to (7) and (8)
repeat
 for $k = 1$ **to** K **do**
 for $n = 1$ **to** N **do**
 compute $w_{F,nk}$ and $w_{I,nk}$ according to (4)
 end for
 end for
 compute λ_k according to (5)
 update $\mathbf{A}_k, \mathbf{B}_k, \Sigma_{k,F}, \Sigma_{k,I}$ according to (6)
until convergence of likelihood
return $\mathbf{A}_k, \mathbf{B}_k, \Sigma_{k,F}, \Sigma_{k,I}, \lambda$

where $\mu(\cdot)$ denotes the mean and the initial covariance matrices

$$\Sigma_{k,F} = \text{std}(\mathbf{Y})/N \quad \Sigma_{k,I} = \text{std}(\mathbf{X})/N, \quad (8)$$

and iterate between E-step and M-step until convergence of the combined conditional likelihood.

Finally, the combined output predictions $\hat{\mathbf{y}}_n$ and input predictions $\hat{\mathbf{x}}_n$ are computed by weighting the components predictions with their corresponding responsibilities

$$\hat{\mathbf{y}}_n = \sum_{k=1}^K \lambda_{nk} \hat{\mathbf{y}}_{nk} \quad \hat{\mathbf{x}}_n = \sum_{k=1}^K \lambda_{nk} \hat{\mathbf{x}}_{nk}. \quad (9)$$

Pseudo code for our EM based learning of coupled forward-inverse models is presented in Algorithm 1.

IV. EXPERIMENTS

In this section, we present our experimental results for learning multiple coupled forward and inverse models. First, we evaluate the performance of CFIM for learning a one-dimensional function. We compare CFIM to MOSAICs approach of only using the forward model errors during training and show that our algorithm yields significantly better results in particular for learning the inverse models.

As we consider our new algorithm in particular suited for problems where the inverse space is not unique, in the second part of this section we show results for learning forward and inverse kinematics of robot arms. While the forward kinematics is usually a unique mapping, the inverse kinematics is usually not. In this case, the inverse space might contain additional information about the context of a task and incorporating the inverse modeling error in the learning of model responsibilities can result in better localization of the single models. The learning in all experiments is done offline.

A. CFIM for 1D artificial data

The sine function is a non-linear function with one-to-many mappings in the inverse space and therefore an interesting benchmark function for learning coupled forward and inverse models. In Figure 4 we present the results of approximating a sine function with 8 linear models. The figure shows the 300 training data points (pink), the single model estimates (light colors) and the combined predictions (black) for the forward function (upper row) and the inverse function (lower row). Figure 4(a) illustrates that the approach to only use the forward errors does not learn a precise inverse model of the given data, and the single models do not adapt themselves optimally to the data set. CFIM results in significantly better inverse modeling as illustrated in Figure(b). Our new algorithm also achieves better localization of the models in both, forward and inverse space.

Figure 4(c) shows the corresponding NRMSE on the training set for different numbers of models. The figure compares the NRMSE for only leveraging on the forward error (orange) and our new algorithm CFIM (green) on the data set from Figure 4 (a),(b). We evaluate the NRMSE for the forward models (upper row) as well as the NRMSE for the inverse models (lower row). It shows that for an increasing number of models the NRMSE of both approaches decrease. However, CFIM achieves significantly lower NRMSE in particular on the inverse models. Additionally, the error of

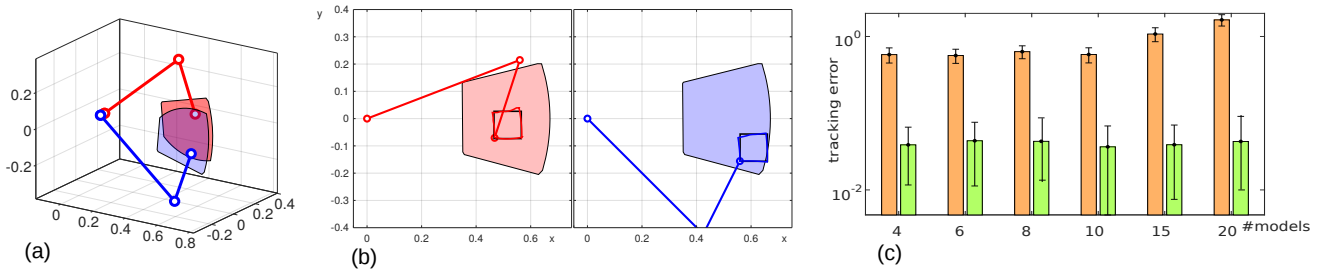


Fig. 5. (a) We consider training data that differs in the inverse space but is overlapping in the forward space, generated by two different elbow configurations (red, blue) (b) CFIM achieves accurate tracking for both configurations due to better localization of the single models in the inverse space. (c) Evaluations on 100 randomly generated test trajectories in task space show that CFIM (green) achieves better results in terms of tracking performance than the approach to only use forward model errors (orange) during training. Moreover, the mean tracking errors for 15 and 20 models show that using only the forward errors results in over fitting of the training data for a higher number of models. Please note that the y-axis is a logarithmic scale.

the forward models can be improved due to the improved localization of the coupled models.

B. Application for Learning Kinematics

We consider a given training set $D = \{(\mathbf{q}_1, \mathbf{c}_1), \dots, (\mathbf{q}_N, \mathbf{c}_N)\}$ with joint positions \mathbf{q} and Cartesian positions of the endeffector \mathbf{c} . The forward and inverse models are defined as

$$\text{fwd} : \mathbf{q} \rightarrow \mathbf{c} \quad \text{inv} : \mathbf{c} \rightarrow \mathbf{q} \quad (10)$$

We use the learned inverse models to generate point-to-point movements in task space. Therefore, the current position of the robot $(\mathbf{c}^t, \mathbf{q}^t)$ is used to predict the responsibilities for the following time step according to Equation (5) and the model estimates according to Equation (3)

$$\mathbf{c}^t, \mathbf{q}^t \rightarrow \lambda_k^t, \quad \mathbf{c}^{t+1} \rightarrow \mathbf{q}_k^{t+1}. \quad (11)$$

This responsibilities and model estimates can be used to compute the next joint positions to reach the desired Cartesian target.

$$\mathbf{q}^{t+1} = \sum_{k=1}^K \lambda_k^t \mathbf{q}_k^{t+1} \quad (12)$$

In the next parts of this section, we present results for a simulated 2-link robot and on a hand over task with a real robotic arm.

1) *2-Link-Robot*: As illustrated in Figure 5 (a), we collect 3780 training data points with two different elbow configurations (red, blue) but in an overlapping task space region. We compare CFIM to the approach from MOSAIC to train responsibilities only on the prediction errors of the forward models. When investigating the localization of the single models it shows that our new algorithm is more capable of capturing the bi-modal nature of the inverse space. This is shown in Figure 6 which illustrates an example of the localization of 10 models for the first joint (left) and the second joint (right). Figure 6 compares the localization of the models when using CFIM (upper row) against only using forward errors for learning responsibilities (lower row). While CFIM captures the two distinct regions in the inverse

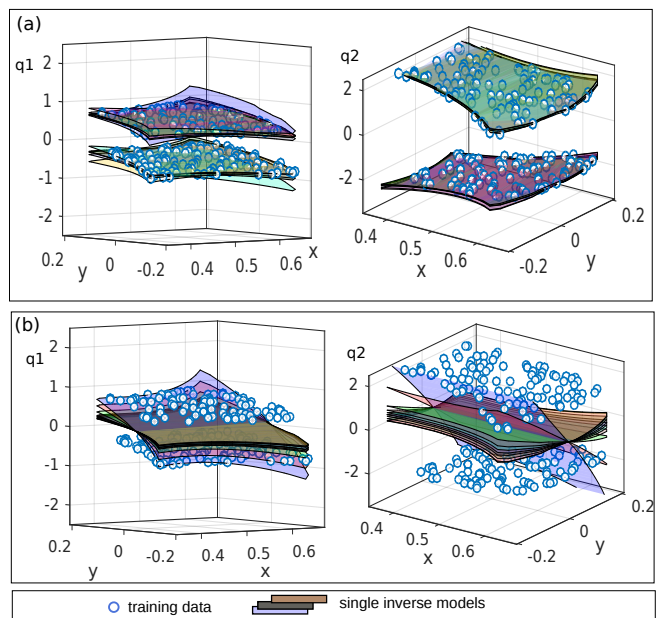


Fig. 6. Localization of the multiple models for modeling the inverse kinematics of a 2-link robot. A total of 10 local linear models, shown as hyperplanes of different colors, were used. (a) The top row shows the results for CFIM. Note that the training data are concentrated in two distinct regions. The algorithm captured these regions (two for each link) by localizing the planes according to the data. (b) Using the same training data, multiple paired models trained only on the errors of the forward models have difficulties to localize themselves over the multiple inverse solutions. Rather, many modules tend to lie at the average of the output space.

space, using only the forward model error results in a localization of the models somewhere in the average of the training data.

Figure 5 (c) and (b) show results of testing the learned inverse models for point-to-point motions. We evaluated our method on 100 randomly generated point-to-point motion trajectories for both elbow configurations. Figure 5(c) compares only using the forward error during training to CFIM. Figure 5(b) shows that CFIM is able to generate motions with both elbow configurations by activating different local models.

2) *Local IK FK Real Robot*: We demonstrate how to use CFIM to learn a handover task with a robotic arm.

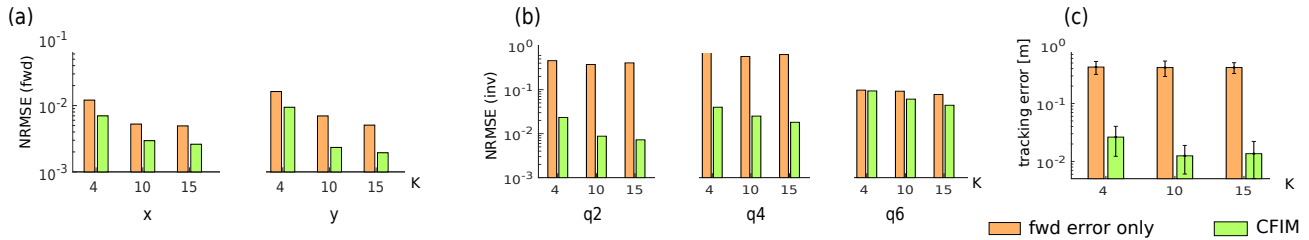


Fig. 7. On the training data CFIM (green) achieves a lower NRMSE for the forward models (a) and the inverse models (b) compared to only using the forward model error as proposed by MOSAIC (orange). (c) Evaluations of tracking 10 trajectories for each elbow configuration with the robot shows that also on the test set CFIM yields significantly better results in terms of tracking error. Please note that all y-axes are logarithmic scale.

We considered a setup, where a human co-worker can be located in two different areas of the work space, giving need to different elbow configurations of the robot. We track the exact x,y position of the robot hand, where the hand over object is located with a motion capturing marker as depicted in Figure 8. The method is used on the 3 joints of the arm which are the most significant for the task.

Figure 7 (a) and (b) show the NRMSE for forward and inverse models on the training data for 4, 10 and 15 models. CFIM (green) yields significantly better results compared to only using the forward model errors. Figure 7 (b) shows for only using the forward model errors even with increasing number of models the NRMSE is not decreasing significantly. To evaluate on test data, Figure 7 (c) shows the mean tracking errors and standard deviation for 10 randomly generated task space trajectories for each elbow context. The plot presents results for 4, 10, and 15 models where CFIM again outperforms the approach of only using the forward error.

Figure 8 illustrates that CFIM learns kinematic solutions to follow trajectories for both elbow configurations. In Table 2 we show the corresponding tracking errors for each elbow configuration and compare to results when only using the forward error. The tracking error was computed for 10 randomly generated trajectories when using 15 paired models.

V. CONCLUSION AND FUTURE WORK

In this paper, we derived a new EM based algorithm for learning multiple coupled forward-inverse models. This new algorithm trains both forward and inverse models simultaneously, and achieves significant improvements over the comparable MOSAIC due to the combined use of forward and inverse model errors. Experimental results show that this use of both inverse and forward errors can help to achieve better localization of the models. Furthermore, we demonstrated how the proposed algorithm can be used to learn local forward and inverse kinematic solutions on robot arms.

CFIM can be used whenever forward and inverse models need to be trained simultaneously. We believe the ideas which we presented in this paper can extend the scope of methods based on multiple paired forward-inverse models. In particular, efficient learning of multiple coupled forward-

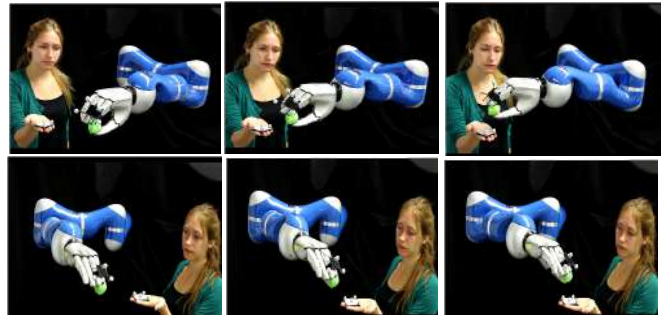


Fig. 8. CFIM can learn inverse kinematics for a hand over task where the robot needs to adapt its elbow configuration to the position of the human. The activation of single local models hereby depends on the start configuration of the elbow and trajectories for both configurations (upper row and lower row) were learned.

TABLE I
TRACKING ERROR REAL ROBOT
(15 MODELS, MEAN AND STD /M)

	EB1	EB2
only fwd error	0.2267 ± 0.1021	0.1954 ± 0.1397
CFIM	0.0075 ± 0.0056	0.0113 ± 0.0101

inverse models can help to learn advanced modular context dependent control architectures.

Transferring the insights provided by this paper to applications in high-dimensional sensorimotor learning is hereby an exciting line of future work.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community’s Seventh Framework Programmes (FP7-ICT-2013-10) under grant agreement 610878 (3rdHand) and from the European Union’s Horizon 2020 research and innovation programme under grant agreement 640554 (SKILLS4ROBOTS).

REFERENCES

- [1] D. M. Wolpert and M. Kawato, “Multiple paired forward and inverse models for motor control,” *Neural networks*, vol. 11, no. 7, pp. 1317–1329, 1998.
- [2] Y. Demiris and B. Khadhour, “Hierarchical attentive multiple models for execution and recognition of actions,” *Robotics and autonomous systems*, vol. 54, no. 5, pp. 361–369, 2006.

- [3] M. Haruno, D. M. Wolpert, and M. Kawato, "Mosaic model for sensorimotor learning and control," *Neural computation*, vol. 13, no. 10, pp. 2201–2220, 2001.
- [4] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: a survey," *Cognitive processing*, vol. 12, no. 4, pp. 319–340, 2011.
- [5] C. Atkeson, "Using locally weighted regression for robot learning," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*. IEEE, 1991, pp. 958–963.
- [6] A. D. Souza, S. Vijayakumar, and S. Schaal, "Learning inverse kinematics," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 1. IEEE, 2001, pp. 298–303.
- [7] M. P. J. Nguyen-Tuong, D.; Seeger, "Local Gaussian process regression for real time online model learning and control," in *Advances in Neural Information Processing Systems 22*, Cambridge, MA, 2009.
- [8] R. Calandra, S. Ivaldi, M. P. Deisenroth, E. Rueckert, and J. Peters, "Learning inverse dynamics models with contacts," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3186–3191.
- [9] M. I. Jordan and D. E. Rumelhart, "Forward models: Supervised learning with a distal teacher," *Cognitive science*, vol. 16, no. 3, pp. 307–354, 1992.
- [10] D. M. Wolpert, K. Doya, and M. Kawato, "A unifying computational framework for motor control and social interaction," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 358, no. 1431, pp. 593–602, 2003.
- [11] A. Tidemann and P. Öztürk, "Self-organizing multiple models for imitation: Teaching a robot to dance the ymca," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, 2007, pp. 291–302.
- [12] M. Haruno, D. M. Wolpert, and M. Kawato, "Hierarchical mosaic for movement generation," in *International congress series*, vol. 1250. Elsevier, 2003, pp. 575–590.
- [13] J. Peters, "Neural networks in robot control," 2000.
- [14] M. Ito, "Neurophysiological aspects of the cerebellar motor control system." *International journal of neurology*, vol. 7, no. 2, pp. 162–176, 1970.
- [15] J. R. Flanagan, S. King, D. M. Wolpert, and R. S. Johansson, "Sensorimotor prediction and memory in object manipulation." *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*, vol. 55, no. 2, p. 87, 2001.
- [16] D. W. Franklin, R. Osu, E. Burdet, M. Kawato, and T. E. Milner, "Adaptation to stable and unstable dynamics achieved by combined impedance control and inverse dynamics model," *Journal of neurophysiology*, vol. 90, no. 5, pp. 3270–3282, 2003.
- [17] K. S. Narendra and J. Balakrishnan, "Adaptive control using multiple models," *IEEE transactions on automatic control*, vol. 42, no. 2, pp. 171–187, 1997.
- [18] K. Doya, K. Samejima, K.-i. Katagiri, and M. Kawato, "Multiple model-based reinforcement learning," *Neural computation*, vol. 14, no. 6, pp. 1347–1369, 2002.
- [19] G. Petkos and S. Vijayakumar, "Context estimation and learning control through latent variable extraction: From discrete to continuous contexts," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 2117–2123.
- [20] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.
- [21] B. Ans, S. Rousset, R. M. French, and S. Musca, "Self-refreshing memory in artificial neural networks: Learning temporal sequences without catastrophic forgetting," *Connection Science*, vol. 16, no. 2, pp. 71–99, 2004.
- [22] M. Kawato, "Feedback-error-learning neural network for supervised motor learning," *Advanced neural computers*, vol. 6, no. 3, pp. 365–372, 1990.
- [23] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.
- [24] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*. John Wiley & Sons, 2007, vol. 382.
- [25] G. Celeux, S. Chrétien, F. Forbes, and A. Mkhadri, "A component-wise em algorithm for mixtures," *Journal of Computational and Graphical Statistics*, 2012.
- [26] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the em algorithm," *Neural computation*, vol. 6, no. 2, pp. 181–214, 1994.
- [27] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton, "Smem algorithm for mixture models," *Neural computation*, vol. 12, no. 9, pp. 2109–2128, 2000.
- [28] B. Damas and J. Santos-Victor, "An online algorithm for simultaneously learning forward and inverse kinematics," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 1499–1506.