# Reactive Hand Movements from Arm Kinematics and EMG Signals Based on Hierarchical Gaussian Process Dynamical Models

Nick Taubert[1], Jesse St. Amand[1], Prerana Kumar[1], Leonardo Gizzi[2], and Martin A. Giese[1]

[1] Section Computational Sensomotorics, Department of Cognitive Neurology, University Clinic Tübingen, CIN, HIH and University of Tübingen, Otfried-Müller-Str. 25, 72076 Tübingen, Germany

[2] Institute for Modelling and Simulation of Biomechanical Systems, Chair for Continuum Biomechanics and Mechanobiology, University of Stuttgart, Stuttgart, Germany

{nick.taubert,jesse.st-amand,martin.giese}@uni-tuebingen.de,
leonardo.gizzi@mechbau.uni-stuttgart.de

**Abstract.** The prediction of finger kinematics from EMG signals is a difficult problem due to the high level of noise in recorded biological signals. In order to improve the quality of such predictions, we propose a Bayesian inference architecture that enables the combination of multiple sources of sensory information with an accurate and flexible model for the online prediction of high-dimensional kinematics. Our method integrates hierarchical Gaussian process latent variable models (GP-LVMs) for nonlinear dimension reduction with Gaussian process dynamical models (GPDMs) to represent movement dynamics in latent space. Using several additional approximations, we make the resulting sophisticated inference architecture real-time capable. Our results demonstrate that the prediction of hand kinematics can be substantially improved by inclusion of information from the online-measured arm kinematics, and by exploiting learned online generative models of finger kinematics. The proposed architecture provides a highly flexible framework for the integration of accurate generative models with high-dimensional motion in real-time inference and control problems.

**Keywords:** EMG, decoding, kinematics, Gaussian process, dimension reduction

## 1 Introduction

Dynamical systems have been used extensively to model human motion [18, 8]. For example, networks of coupled dynamic movement primitives or 'central pattern generators' have been shown to describe coordinated motor patterns [14]. Applications in computer graphics and robotics have resulted in powerful methods for modeling complex, high-dimensional, coordinated human motion. These models may be exploited in the context of prosthesis control applications, where often the available control signals (e.g. from electromyography (EMG)) are too low-dimensional or noisy to ensure accurate decoding of the desired actuator motion.

As a theoretical framework for this integration of neural signal decoding and synthesizing human motion, we propose probabilistic graphical models [1]. These models provide a highly flexible, systematic framework for modular model construction and realizing inference on arbitrary variables in such models. However, complex graphical models, due to the computational complexity of this inference, often do not easily transfer to online control applications or the online generation of human motion. Neither do many of the common methods for the offline synthesis of human motion, e.g. by motion capture and subsequent filtering or interpolation between stored trajectories [4], transfer easily to online synthesis applications. Based on a combination of GP-LVMs [22] with GPDMs [33], we propose here an implementation of a hierarchical probabilistic graphical model that is suitable for real-time inference. We exploit this architecture to estimate hand kinematics from EMG signals by a learned generative model of hand-arm coordination, with support by real-time kinematic data from the arm. This inferred hand motion can be used in the control of a hand orthosis for patients with unhindered arms, but impaired grasping abilities (e.g. due to stroke) [3].

## 2 Background

The industry standards and the typical benchmarks in research for online hand and arm myocontrol comprise the two-electrode "conventional control" approaches, co-contraction control (CC) and slope control (SC) [16]. Both require users to make targeted muscle contractions for switching between two single degrees of freedom (DOF) modes of control—operating a wrist rotation and opening/closing the hand. Users often consider these systems to be unintuitive compared to natural hand and arm function, difficult to use in everyday life, and to be fatiguing [16, 11]. Other approaches exploiting classification and multiple DOFs regression methods have been shown to produce better results with the same hardware [16]. More advanced techniques have effectively employed methods of auto-encoding for dimensionality reduction [31], have utilized biomechanical models for imposing realistic system constraints [10, 28], or have incorporated proprioceptive feedback or direct cortical feedback into sensory-motor brain areas [12]. Here, we explore the use of generative hierarchical Gaussian process models for establishing control by non-invasive EMG signals, augmented by predictive signals derived from the measurement of additional arm kinematic signals.

Probabilistic models for human motion have been used extensively in computer graphics for the synthesis of character motion [7], and for the editing and interpolation of motion styles [19, 2, 17]. Many of these techniques result in offline models that cannot react to online control inputs, while simple online control schemes often are not accurate enough to convey the exact motion characteristics. GP-LVMs have been used for kinematic modeling and motion interpolation [15], inverse kinematics [24], and for the learning of low-dimensional dynamical models [34].

Gaussian processes (GPs) can be interpreted as probabilistic neural networks with a particular prior on the weights and biases in the limit of infinitely many hidden units [26]. The GP-LVM have been extended by latent dynamics in [33], resulting in an architecture that has been termed the Gaussian process dynamical model (GPDM). Due to its probabilistic nature, this model is well equipped to handle the variability of natural motion data, and it is possible to model full-body human motion with just one GPDM [30, 33, 29]. Here, we use a combination of GP-LVMs and GPDM to build a hierarchical GPDM with goal/style-content separation for modeling the kinematics of coordinated hand and arm movements. To achieve real-time performance, we approximate computationally costly inference steps by learning a fast, approximate inverse of our generative model with GP, which we refer to as *back-projection*. This way of making inference tractable was inspired by the Helmholtz Machine [9], which also learns an explicit inversion of a generative model.

## 3 Model components

Our model architecture is shown in Figure 1a. It is composed of a hierarchical combination of GP-LVMs, and a top level that is implemented by a GPDM-like dynamical layer.

### 3.1 GP-LVMs

In a GP-LVM, each point $\mathbf{y}_n$ of a high dimensional data set $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^{\mathrm{T}} \in \Re^{N \times D}$ is represented by a corresponding, uncertain instance $\mathbf{x}_n$ of low dimensional latent variables $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^{\mathrm{T}} \in \Re^{N \times Q}$, where $N$ is the length of the data set, $D$ the dimension of the data space, and $Q$ the dimension of the latent space. The mapping between the latent space and the components of the data is defined by a (typically nonlinear) function $f(\mathbf{x})$ that is drawn from a Gaussian process (with $\mathbf{y} = [y_1, ..., y_D]^{\mathrm{T}}$):

$$y_d = f_d(\mathbf{x}) + \varepsilon_d, \qquad f_d(\mathbf{x}) \sim GP(0, k_{\mathbf{f}}(\mathbf{x}, \mathbf{x}')) \tag{1}$$

where $\varepsilon_d$ is white noise of the data, and $f_d(\mathbf{x})$ is drawn from a GP prior with zero mean function and kernel function $k_{\mathbf{f}}(\mathbf{x}, \mathbf{x}')$. We used composite kernel functions (see eq. 14, 15, 18) which were composed of a radial basis function (RBF) and a linear kernel [27]:

$$k_{\mathrm{RBF}}(\mathbf{x}, \mathbf{x}') = \gamma_1 \exp\left(-\frac{\gamma_2}{2}|\mathbf{x} - \mathbf{x}'|^2\right) \tag{2}$$

$$k_{\mathrm{lin}}(\mathbf{x}, \mathbf{x}') = \gamma_3 \mathbf{x}^{\mathrm{T}} \mathbf{x}' \tag{3}$$

The RBF kernel models local structure in the data and has an inverse width $\gamma_2$. The linear kernel allows linear extrapolation and interpolation between data points. The other $\gamma_i$ are positive weights.

We learned the parameters of the GP-LVMs by determining the latent variables that maximize their log-posterior, exploiting a scaled conjugate gradient (SCG) algorithm [22]. The evaluation of the log-posterior involves the inversion of a $N \times N$ kernel matrix $\mathbf{K}_{\mathbf{ff}}$ which results from computing the kernel at every pair of latent points. This inversion has computational complexity $O(N^3)$ which is prohibitive for large data-sets, such as our motion capture recordings.

### 3.2 Sparse aproximation

We solved this problem by a sparse approximation approach [6]. For this purpose, the noise free function value set $\mathbf{F} = [\mathbf{f}_1, \ldots, \mathbf{f}_N]^{\mathrm{T}}$, where $\mathrm{f}_d = f_d(\mathbf{x})$ with $\mathbf{f} = [\mathrm{f}_1, \ldots, \mathrm{f}_D]^{\mathrm{T}}$, is approximated by selection of a small set of $M \ll N$ pseudo-points $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_M]^{\mathrm{T}}$. It is assumed that training and test points of the

Gaussian process are approximately conditionally independent, if conditioned on their pseudo-points. Under this assumption, the GP prior can be approximated as follows:

$$q(\mathbf{f}|\boldsymbol{\theta}) = \int p(\mathbf{f}|\mathbf{u},\boldsymbol{\theta})p(\mathbf{u}|\boldsymbol{\theta})\,\mathrm{d}\mathbf{u}, \tag{4}$$

where $\boldsymbol{\theta}$ are the model hyperparameters and $p(\mathbf{u}|\boldsymbol{\theta})$ is the GP prior over $\mathbf{u}$. The conditional relationship between $\mathbf{u}$ and $\mathbf{f}$ is fundamental for this equation. With the GP priors $p(\mathbf{f}|\boldsymbol{\theta})$ and $p(\mathbf{u}|\boldsymbol{\theta})$ it is possible to derive from their joint probability the conditional dependency. This leads to the result $p(\mathbf{f}|\mathbf{u},\boldsymbol{\theta}) = \mathcal{N}(\mathbf{f};\mathbf{K_{fu}K_{uu}^{-1}u},\mathbf{D_{ff}})$ where $\mathbf{D_{ff}} = \mathbf{K_{ff}} - \mathbf{Q_{ff}}$ and $\mathbf{Q_{ff}} = \mathbf{K_{fu}K_{uu}^{-1}K_{uf}}$. The constructed matrices correspond to the covariance function evaluations at the pseudo-point input locations $\{[\mathbf{x_u}]_m\}_{m=1}^M$, i.e. $[\mathbf{K_{uu}}]_{m,m'} = k_{\mathbf{f}}([\mathbf{x_u}]_m,[\mathbf{x_u}]_{m'})$ and similarly covariance function evaluations between pseudo-point input and data locations $[\mathbf{K_{fu}}]_{n,m} = k_{\mathbf{f}}(\mathbf{x}_n,[\mathbf{x_u}]_m)$. Unfortunately, even this equation it is still cubic in terms of its computational complexity. However, the matrices of $p(\mathbf{f}|\mathbf{u},\boldsymbol{\theta})$ can be approximated by simpler forms, resulting in computationally tractable distributions with $q(\mathbf{f}|\mathbf{u},\boldsymbol{\theta}) \approx p(\mathbf{f}|\mathbf{u},\boldsymbol{\theta})$ [5]. We chose a subset of pseudo-inputs which parameterize the GP prior through the Deterministic Training Conditional (DTC) approximation [20] in order to ensure real-time capability for prediction and to render our approach feasible for larger data sets,

$$q(\mathbf{f}|\mathbf{u},\boldsymbol{\theta}) = \mathcal{N}(\mathbf{f}|\mathbf{K_{fu}K_{uu}^{-1}u},\mathbf{0}). \tag{5}$$

This approximation assumes that $f(\mathbf{x})$ is fully determined by the pseudo-point inputs and reduces the computational cost to $O(M^2N)$ during learning. The prior $q(\mathbf{f}|\boldsymbol{\theta})$ with DTC approximation can be combined with the likelihood $p(\mathbf{y}_d|\mathbf{f}_d,\boldsymbol{\theta}) = \mathcal{N}(\mathbf{y}_d|\mathbf{f}_d,\gamma_4^{-1}\mathbf{I})$:

$$p(\mathbf{Y}|\theta) = \prod_{d=1}^{D}\int p(\mathbf{y}_{:,d}|\mathbf{f}_{:,d},\boldsymbol{\theta})q(\mathbf{f}_{:,d}|\boldsymbol{\theta})\,\mathrm{d}\mathbf{f}_d \tag{6}$$

$$= \prod_{d=1}^{D}\mathcal{N}(\mathbf{y}_{:,d}|\mathbf{0},\mathbf{K_{fu}K_{uu}^{-1}K_{uf}} + \gamma_4^{-1}\mathbf{I}). \tag{7}$$

where $\gamma_4$ is the noise precision into which any Gaussian observation noise can be absorbed and $\mathbf{y}_{:,d},\mathbf{f}_{:,d}$ are the vectors for the $d$th dimension of $\mathbf{Y}$ and $\mathbf{F}$. Combined with priors $p(\mathbf{X})$ and $p(\boldsymbol{\theta})$ we got the log-marginal, reformulated with matrix inversion lemma [20] for $O(M^2N)$:

$$\mathcal{L} = -\frac{D(N-M)}{2}\ln 2\pi - \frac{\gamma_4}{2}\mathrm{tr}(\mathbf{YY}^{\mathrm{T}})\frac{D}{2}\ln|\mathbf{K_{uu}^{-1}}| + \frac{D}{2}\ln|\mathbf{A}| \tag{8}$$

$$-\frac{\gamma_4}{2}\mathrm{tr}(\mathbf{A}^{-1}\mathbf{K_{uf}}\mathbf{YY}^{\mathrm{T}}\mathbf{K_{fu}}) - \frac{1}{2}\sum_{n=1}^{N}\mathbf{x}_n^{\mathrm{T}}\mathbf{x}_n - \sum_{j}\ln\gamma_j, \tag{9}$$

with $\mathbf{A} = \gamma_4^{-1}\mathbf{K_{uu}} - \mathbf{K_{uf}K_{fu}}$. This implies that the computation time depends only on the number of pseudo- and test inputs, which enables real-time performance.

For the prior function of new test data $p(\tilde{\mathbf{Y}}|\boldsymbol{\theta})$, we followed the same derivations as in equations (4)-(7), replacing $\mathbf{f}$ with test function values $\tilde{\mathbf{f}}$, assumed conditional independence between $\tilde{\mathbf{f}}$ and $\mathbf{f}$. With the joint probability of the new GP priors $p(\mathbf{Y}|\boldsymbol{\theta})$ and $p(\tilde{\mathbf{Y}}|\boldsymbol{\theta})$, a posterior distribution $p(\tilde{\mathbf{y}}|\mathbf{Y}, \mathbf{X}, \mathbf{x}_*, \boldsymbol{\theta})$ for the prediction of new test outputs $\tilde{\mathbf{y}}$ with given test inputs $\mathbf{x}_*$ can be derived [21],

$$\tilde{\mathbf{y}} \sim \mathcal{N}(\mu(\mathbf{x}_*), \sigma(\mathbf{x}_*)), \tag{10}$$

where

$$\mu(\mathbf{x}_*) = \mathbf{K}_{*\mathbf{u}}\mathbf{A}^{-1}\mathbf{K}_{\mathbf{uf}}\mathbf{Y}, \tag{11}$$

$$\sigma(\mathbf{x}_*) = \mathbf{K}_{**} - \mathbf{K}_{*\mathbf{u}}(\mathbf{K}_{\mathbf{uu}} - \mathbf{A}^{-1})\mathbf{K}_{*\mathbf{u}}^{\mathrm{T}}, \tag{12}$$

where $\mathbf{K}_{**}$ is the kernel function evaluated at the test inputs, $\mathbf{K}_{*\mathbf{u}}$ is the kernel function evaluated at the test and pseudo-point inputs.

### 3.3 GPDM

In order to model the dynamics of our data in the latent space—that which is given by the temporal evolution of the columns of the inferred input matrix $\mathbf{X}$, we used a Gausian Process Dynamical Model (GPDM) [33]. This is a GP-LVM with an autoregressive dynamics for the hidden variables $\mathbf{x}_n$. These dynamics are given by a nonlinear auto-regressive model of the form:

$$\mathbf{x}_n = f_{\mathbf{x}}(\mathbf{x}_{n-1}, \ldots, \mathbf{x}_{n-z}) + \xi, \tag{13}$$

where $z$ is the order of the dynamics (we used $z = 2$), and where $f_{\mathbf{x}}(\cdot)$ is also drawn from a Gaussian process prior with the associated kernel function:

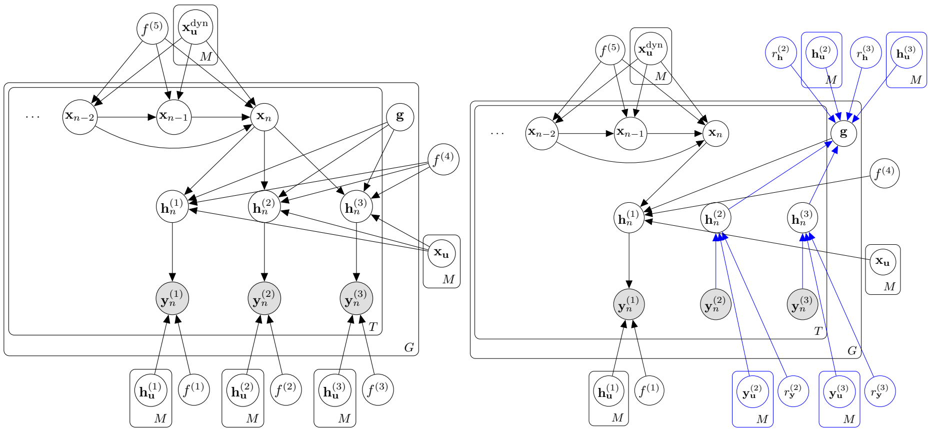
$$\begin{aligned} &k_{\mathrm{dyn}}([\mathbf{x}_{n-1}, \mathbf{x}_{n-2}], [\mathbf{x}_{\eta-1}, \mathbf{x}_{\eta-2}]) \\ &\qquad = \gamma_5 \exp\Big(-\frac{\gamma_6}{2}|\mathbf{x}_{n-1} - \mathbf{x}_{\eta-1}|^2 - \frac{\gamma_7}{2}|\mathbf{x}_{n-2} - \mathbf{x}_{\eta-2}|^2\Big) \\ &\qquad + \gamma_8 \mathbf{x}_{n-1}^{\mathrm{T}} \mathbf{x}_{\eta-1} + \gamma_9 \mathbf{x}_{n-2}^{\mathrm{T}} \mathbf{x}_{\eta-2}. \end{aligned} \tag{14}$$

This is again a combination of non-linear and linear kernel functions. This autoregressive model defines a Markov chain and generalizes a hidden Markov model (HMM) [25] in a non-linear way.

## 4 Hierarchical model architecture

For the representation of goal-dependent hand movements, we devised a hierarchical model that consists of GP-LVMs for successive dimension reduction, and of a GPDM in the highest layer for the representation of the dynamics in the latent space (cf. Fig. 1a). The model is composed of three layers, which were learned jointly. Learning includes both the bottom-up and top-down contributions at each hierarchy level.

The proposed model can be run in two different modes: (i) As a *fully generative model*, where we the complete hand-arm kinematics is generated online, dependent on an external variable that controls the goal position; (ii) as a *partially generative model*, where the goal position is inferred online from measured arm kinematics and/or EMG signals. In this case these variables are treated as observed nodes in the graphical model, and the other variables are inferred by Bayesian model inversion. For the separation of style and content [32], style referring here to the goal position, we introduced a specific goal/style variable $\mathbf{g}$ that captures specifically the goal position, which either is pre-specified in the fully generative mode, or inferred in the partially generative mode.



**a)** Graphical model of the hierarchical GPDM. **b)** Back-projection of the hierarchical GPDM.

**Fig. 1.** *Hierarchical probabilistic model.* **(a)** Three data structures $\mathbf{Y}^{(i)}, i \in \{1, 2, 3\}$ including the arm kinematic data, the hand kinematic data, and the EMG data are represented by sparse GP-LVMs, each over $T$ trials and $G$ goal positions. Each data structure is mapped by a prior mapping function $f^{(i)}(\cdot)$ drawn from a GP prior with a lower-dimensional latent variable $\mathbf{h}^{(i)}$. For each a set $M$ pseudo-input variables $\mathbf{h}_\mathbf{u}^{(i)}$ are specified for sparse approximation. The dimensionality of the variables $\mathbf{h}^{(i)}$ on the middle level of the model is further reduced by a sparse GP-LVM in the same way, with the hidden state variables $\mathbf{x}$, the corresponding inducing variables $\mathbf{x}_\mathbf{u}^{(i)}$, and the style variable $\mathbf{g}$. The temporal evolution of the hidden state variables $\mathbf{x}_n$ is modeled with second order dynamics using a GPDM also drawn from a GP prior $f^{(5)}$. Corresponding inducing variables for the sparse representation are signified by $\mathbf{x}_\mathbf{u}^{\mathrm{dyn}}$. **(b)** Back-projection mapping (blue) from data space of $\mathbf{y}^{(j)}, j \in \{2, 3\}$ to the latent spaces $\mathbf{h}^{(j)}$, using sparse GP regressions with prior functions $r_\mathbf{y}^{(j)}$ and the pseudo-inputs $\mathbf{y}_\mathbf{u}^{(j)}$. Similarly, a back-projection from $\mathbf{h}^{(j)}$ to $\mathbf{g}$ is computed with the funcion prior $r_\mathbf{h}^{(j)}$ and pseudo-input $\mathbf{h}_\mathbf{u}^{(j)}$.

### 4.1 Individual layers

**Bottom layer.** The bottom layer of our model represents the observed kinematic data of the hand, the kinematic velocity data of the arm, and the EMG data as matrices $\mathbf{Y}^{(i)}, i \in \{1; 2; 3\}$. The dimensionality of each of these data sets is reduced by a sparse GP-LVM with the same sample size ($N^{(i)} = 1440$), but different dimensionalities of the data vectors ($D_\mathbf{y}^{(1)} = 63, D_\mathbf{y}^{(2)} = 15, D_\mathbf{y}^{(3)} = 28$). For each data type, a separate set of mapping functions is used with their own GP prior $f^{(i)}(\cdot)$ to map the latent variables $\mathbf{h}^{(i)}$ onto the corresponding data vectors, according to equation (1). The dimensionalities of the hidden variable vectors are $Q_\mathbf{h}^{(1)} = 10$, $Q_\mathbf{h}^{(2)} = 4$, and $Q_\mathbf{h}^{(3)} = 15$. The kernel functions in this layer are given by a linear combination of the RBF and the linear kernel functions defined in equations (2),(3) to capture the nonlinarity of the data and support smooth blending between styles:

$$k_\mathbf{f}^{(i)}(\mathbf{h}^{(i)}, \mathbf{h}'^{(i)}) = k_{\mathrm{rbf}}(\mathbf{h}^{(i)}, \mathbf{h}'^{(i)}) + k_{\mathrm{lin}}(\mathbf{h}^{(i)}, \mathbf{h}'^{(i)}). \tag{15}$$

**Intermediate layer.** For further dimensionality reduction, an additional sparse GP-LVM was introduced that maps from the latent variable $\mathbf{x}$ ($Q_{\mathbf{x}} = 2$) onto the concatenation of the variables $\mathbf{h}^{(i)}$. We assume that $\mathbf{h}^{(i)}$ is represented in the same latent space across trials, number of goals, and time steps.

A key assumption in our approach is that the goal positions can be modeled by a separate 'style variable' $\mathbf{g}$. During the learning of the model parameters, we separate the motion content, i.e. the basic shape of the trajectories, from this motion style variable, using a factorial representation. For the partially generative model, we infer the style variable $\mathbf{g}$ online from the measured EMG and kinematic data of the arm. To promote such a factorization of the latent variables in terms of motion content and style dimensions, we applied *back-constraints* during learning [23]. This method constrains the hidden space variable to a low-dimensional manifold in a way that ensures that close points in the data space remain close in the latent space. Adjusting the gradient steps during optimization, we enforced the components of $\mathbf{x}$ to lie on a circular manifold for all styles that are given by the functions $z_1$ and $z_2$ [30]:

$$x_{n,1} = z_1([\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}]_n, \phi_n) = \sum_{\tau=1}^{N} c_{\tau,1} k_{\mathrm{rbf}}(\cos(\phi_n), \cos(\phi_\tau)) \tag{16}$$

$$x_{n,2} = z_2([\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}]_n, \phi_n) = \sum_{\tau=1}^{N} c_{\tau,2} k_{\mathrm{rbf}}(\sin(\phi_n), \sin(\phi_\tau)) \tag{17}$$

The variable $\phi_n$ specifies the motion phase, and the coefficients $c_{\tau,i}$ were chosen to ensure that the latent points lie (approximately) on a circle for all motion 'styles'.

The motion style variable $\mathbf{g}$, which specifies the reaching goal, was initially specified using 1-of-$K$ encodings [1], each goal being specified by a (eight-dimensional) unit vector (goal 1: $\mathbf{g}_1$, goal 2: $\mathbf{g}_2$, etc.) which will also be optimized. In order to blend linearly between the behaviors for the different goals, we used a linear kernel for the 'style' dimensions. The composite kernel function for the middle layer of our model was thus given by (equations (2),(3)):

$$k_{\mathbf{h}}([\mathbf{x}, \mathbf{g}], [\mathbf{x}', \mathbf{g}']) = k_{\mathrm{lin}}(\mathbf{g}, \mathbf{g}') k_{\mathrm{rbf}}(\mathbf{x}, \mathbf{x}') + k_{\mathrm{lin}}(\mathbf{x}, \mathbf{x}'). \tag{18}$$

Similar to the multi-factor model [32], the multiplication of style and latent position act similarly to a logical 'AND' operation. The resulting kernel matrix links each goal point to a pair of latent points.

**Top layer.** The top layer represents the temporal evolution of $\mathbf{x}_n$, i.e. the underlying dynamics. We used a second-order GPDM, which allowed us to model the dynamical dependencies on velocity and acceleration of $\mathbf{X}$. The evolution function for $\mathbf{x}_n$ is given by equation (13). We drew the function $f_{\mathbf{x}}(\cdot)$ from a GP with the kernel given in equation (14), where we learned the model with multiple motion sequences ($T$ trials for $G$ goals, see Fig. 1a), assuming they are i.i.d. samples from the complete model.

In order to accelerate the inference of the partially generative model, we introduced an explicit learning of the back-projections from the hidden variables $\mathbf{h}^{(2)}$ and $\mathbf{h}^{(3)}$ that represent the EMG and the arm kinematics to the variable $\mathbf{h}^{(1)}$ that represents the hand kinematics. These mappings were again represented by sparsely approximated Gaussian process regressions with the function priors $r_{\mathbf{y}}^{(j)}$, $j \in \{2,3\}$ (blue elements in Fig. 1b):

$$\mathbf{h}^{(j)} = r_{\mathbf{y}}^{(j)}(\mathbf{y}^{(j)}) + \boldsymbol{\epsilon}^{(j)}, \quad r_{\mathbf{y}}^{(j)}(\mathbf{y}^{(j)}) \sim GP(\mathbf{0}, k_{\text{back}}^{(j)}(\mathbf{y}^{(j)}, \mathbf{y}'^{(i)})), \tag{19}$$

The kernel was given by $k_{\text{back}}^{(j)}(\mathbf{y}^{(j)}, \mathbf{y}'^{(i)}) = k_{\text{rbf}}^{(j)}(\mathbf{y}^{(j)}, \mathbf{y}'^{(i)}) + k_{\text{lin}}^{(j)}(\mathbf{y}^{(j)}, \mathbf{y}'^{(i)})$, and $\boldsymbol{\epsilon}^{(j)}$ is Gaussian noise. As for a usual GP regression, only the kernel parameters are optimized during this step, while the latent variables for both layers were kept fixed. In an equivalent way, we introduced explicit back-projections for the second layer to predict the style variable $\mathbf{g}$.

### 4.2 Motion generation and prediction.

In the fully generative mode, our model, after training, predicts the hand kinematics (and also the arm kinematics and EMG signals) from a defined goal position. In the partially generative mode, it generates the hand kinematics from the arm kinematics and/or the EMG signals, estimating the likely goal position $\mathbf{g}$ from the available input data, while exploiting the back-projections for fast inference.

In both cases, the top dynamic layer ensures the generation of smooth hand motion. Given the states for the previous time steps, the GPDM predicts a novel state according to equation (10):

$$\tilde{\mathbf{x}}_n \cong \mu_{\mathbf{x}}([\tilde{\mathbf{x}}_{n-1}, \tilde{\mathbf{x}}_{n-2}]). \tag{20}$$

In the fully generative mode, the predicted state is propagated down the hierarchy to predict the original data components $\tilde{\mathbf{y}}_n^{(i)}$ according to the relationships:

$$\tilde{\mathbf{h}}_n^{(i)} \cong \mu_{\mathbf{h}}([\tilde{\mathbf{x}}_n, \mathbf{g}_*]), \tag{21}$$

$$\tilde{\mathbf{y}}_n^{(i)} \cong \mu_{\mathbf{y}}(\tilde{\mathbf{h}}_n^{(i)}). \tag{22}$$

The functions $\mu_i(.)$ are following from the posteriors of the individual layers. The goal information is added through the variable $\mathbf{g}_*$ in this equation, for which values can be chosen that interpolate between the training goals' vectors, resulting in continuously interpolated intermediate goal positions.

In the partially generative mode, the goal vector $\mathbf{g}$ is predicted from the measurements $\mathbf{y}_*^{(j)}$, $j \in \{2,3\}$, through the back-projections, according to the relationships:

$$\tilde{\mathbf{h}}_n^{(j)} \cong \mu_{\mathbf{h}}(\mathbf{y}_*^{(j)}) \tag{23}$$

$$\tilde{\mathbf{g}} \cong \mu_{\mathbf{g}}^{(j)}(\tilde{\mathbf{h}}_n^{(j)}) \tag{24}$$

For the models with two predictive variables $\tilde{\mathbf{h}}_n^{(j)}$, the estimates were combined using a maximum a posteriori framework. The style vector estimate is then combined with the predicted state $\tilde{\mathbf{x}}_n$ from the dynamics and projected down to predict the hand kinematics $\tilde{\mathbf{y}}_n^{(1)}$ according to:

$$\tilde{\mathbf{h}}_n^{(1)} \cong \mu_{\mathbf{h}}([\tilde{\mathbf{x}}_n, \tilde{\mathbf{g}}]), \tag{25}$$

$$\tilde{\mathbf{y}}_n^{(1)} \cong \mu_{\mathbf{y}}(\tilde{\mathbf{h}}_n^{(1)}). \tag{26}$$

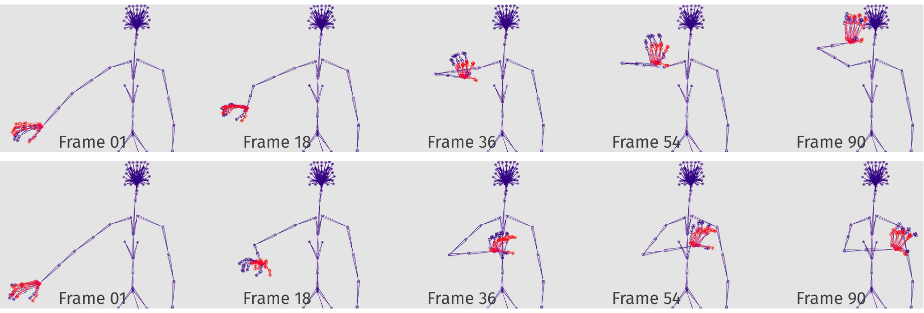


**Fig. 2.** *Frames from animations illustrating the results.* Prediction examples of hand and finger orientations over time for goal 1 (top row) and goal 09 (bottom row). Predicted hand kinematics are shown in red, and the corresponding ground truth data are in blue.

## 5 Results

### 5.1 Data set.

In order to test the performance of our approach, we applied it to a data set with grasping movements that included kinematic data from the arm and the hand, and HD-sEMG data from 3 high-density 8 x 8 surface electrode arrays (OT Bioelettronica) placed around the forearm, and from 16 bipolar sEMG electrodes on the larger muscles of the upper arm, shoulder, chest, and back. EMG recordings were done with an OT Bioelettronica Quattrocento 400 channel desktop amplifier (OTBioelettronica, Torino, IT) and three EMG pre-amplifiers. Samples were taken at 2,848 Hz, with an amplification gain of 150 V/V, and an online bandpass filter between 10 to 900 Hz. MOCAP recordings were conducted through VICON Nexus version 2.8 using eight VICON infrared cameras at 256.41 Hz.

Reaching movements were performed by a healthy, left-handed, adult male (39 years old, 178 cm tall, 82 kg) with no prior record of neuromuscular disease. Grasped objects were mounted on a vertical panel in the form of a rectangular $3 \times 3$ grid with a distance of 25 cm between the objects. All movements started from a resting position next to the hip.

EMG signals were processed by digital band-pass filtering between 20-450 Hz with a fourth-order Butterworth filter. A second-order Butterworth filter was applied as a band-stop filter between 49-51 Hz to remove power line interference. The signals were full-wave rectified and processed with a low-pass, second-order Butterworth filter at 10 Hz to extract the linear envelope.

### 5.2 Model evaluation.

We applied our method to infer the hand kinematics (63 DOFs) from either the arm kinematics (velocities) (15 DOF), the EMG signals, or from both (see Fig. 2). In total, we learned 16 motion trajectories. Due to outliers in the EMG data, we selected only two trials per goal for learning and one trial per goal for testing. Because of outliers in each trial we excluded goal two completely. Our raw training set consisted of EMG data, and arm and hand kinematics for eight goals.

| Hand NMSPR | | | |
|---|---|---|---|
| Goals | EMG + Velocity | EMG | Velocity |
| Goal 01 | 0.2425 | 0.2549 | 0.2472 |
| Goal 03 | 0.2380 | 0.2852 | 0.2477 |
| Goal 04 | 0.3692 | 0.4450 | 0.2795 |
| Goal 05 | 0.3368 | 0.3851 | 0.3109 |
| Goal 06 | 0.2865 | 0.3657 | 0.2436 |
| Goal 07 | 0.3701 | 0.4206 | 0.3191 |
| Goal 08 | 0.3920 | 0.4153 | 0.2992 |
| Goal 09 | 0.2120 | 0.2091 | 0.1802 |
| Average | 0.3059 | 0.3476 | **0.2659** |

| Goal NMSPR | | | |
|---|---|---|---|
| Goals | EMG + Velocity | EMG | Velocity |
| Goal 01 | 0.3317 | 0.2885 | 0.3377 |
| Goal 03 | 0.2964 | 0.3150 | 0.2578 |
| Goal 04 | 0.3682 | 0.4302 | 0.2860 |
| Goal 05 | 0.2488 | 0.4081 | 0.3458 |
| Goal 06 | 0.3463 | 0.4081 | 0.2631 |
| Goal 07 | 0.3508 | 0.4874 | 0.3563 |
| Goal 08 | 0.4040 | 0.4813 | 0.3170 |
| Goal 09 | 0.2864 | 0.2561 | 0.1918 |
| Average | 0.3291 | 0.3844 | **0.2944** |

Table 1: *Normalized Mean Square Prediction Error (NMSPR) for hand kinematics and goal position.* Prediction from arm velocities results in the smallest prediction error.

The bvh kinematic arm and hand data were converted from radian to exponential map [13]. Then, we extracted the velocities from exponential map representations of the arm kinematics. We trained the model with 200 pseudo-inputs for each latent space on an AMD Ryzon Threadripper 1950X 16 core processor 3.4 GHz with 64 GB RAM, which took about 10.5 hours for 1500 iteration steps.

We computed the normalized Mean Square Prediction Error (NMSPR) for predictions of the hand kinematics (joint angles). Even for this small data set, the model produced acceptable hand postures during reconstruction and the NMSPRs for the hand kinematics were smaller for predictions from the arm kinematics than for predictions from the EMG. Combining the prediction from EMG and arm kinematics reduced the prediction error substantially compared to the prediction from EMG alone.

We also analyzed the classification of the goal from the last 20 stimulus frames based on the latent space variable $\mathbf{g}$. The predicted classification was determined by the largest element of the inferred variable, which corresponds to the most likely class. The right part of Table 1 shows the NMSPR for the resulting predicted vectorial binary classification vectors relative to the ground truth, which is given by the true classes defined by the individual goals. Also for this measure, classification is best from the arm kinematics, and including the arm kinematics in the prediction improves accuracy compared to predictions derived from EMG alone.

An additional important point is that the proposed algorithm achieves *real-time performance* for online prediction of the hand kinematics. Running the algorithm on the same computer used for optimization, the prediction time per frame was on average about 23.74 ms.

## 6 Conclusion

Combining several methods from Bayesian unsupervised learning and inference, we have devised a new real-time-capable method for the simulation of reactive hand movements controlled by EMG. The high flexibility of the underlying inference framework was exploited to combine the EMG with additional kinematic data from the arm. Our model successfully predicted hand positions relative to the goal objects and reproduced hand kinematics with acceptable accuracy. Predictions derived from arm kinematics were more accurate for the tested data set than were the reconstructions from EMG. In addition, the accuracy of the reconstruction from EMG could be significantly improved by adding information from arm kinematics. The superiority of arm kinematics might be a consequence of our data set, which did not include transitions between different grip types, or supination movements. Such a variation in the dataset might give predictive power to the forearm HD-sEMG signals, enabling the disambiguation of movements that cannot be derived from the arm kinematics alone.

Despite the sophistication of the underlying probabilistic model, we have demonstrated that the algorithm is real-time-capable on a standard hardware. To our knowledge, underlying, advanced statistical methods such as GPDMs or inference on hierarchical GP-LVMs have never been tested and prepared for real-time applications. In our implementations real-time capability could be achieved by inclusion of several additional approximations, such as explicit back-projections and sparse approximations.

Future work will extend such architectures by applying variational optimization with better approximation methods to enable deep architectures with increased scalability. This will enable testing of these algorithms on much bigger data sets. Furthermore, we plan to implement the architecture with GPU computing for faster learning and inference.

**Acknowledgments** We would like to thank Albert Mukovskiy for his invaluable insight and code regarding the pre-processing of our EMG and kinematic data. This research was funded through HFSP RGP0036/2016, BMBF FKZ 01GQ1704, KONSENS-NHE BW Stiftung NEU007/1, DFG GZ: KA 1258/15-1, ERC 2019-SyG-RELEVANCE-856495.

## References

1. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2007)
2. Brand, M., Hertzmann, A.: Style machines. In: Proc. SIGGRAPH'00. pp. 183–192 (2000)
3. Brokaw, E., Black, I., Holley, R., Lum, P.: Hand spring operated movement enhancer (handsome): A portable, passive hand exoskeleton for stroke rehabilitation. Neural Systems and Rehabilitation Engineering, IEEE Transactions on 19, 391 – 399 (09 2011)
4. Bruderlin, A., Williams, L.: Motion signal processing. In: Proc. SIGGRAPH'95. pp. 97–104. ACM (1995)
5. Bui, T.D.: Efficient Deterministic Approximate Bayesian Inference for Gaussian Process models (September) (2017)
6. Quiñonero Candela, J., Rasmussen, C.E.: A unifying view of sparse approximate Gaussian process regression. J. Mach. Learn. Res. 6, 1939–1959 (2005)
7. Chai, J., Hodgins, J.K.: Performance animation from low-dimensional control signals. ACM Trans. Graph. 24(3), 686–696 (2005)
8. Churchland, M., Cunningham, J., Kaufman, M., Foster, J., Nuyujukian, P., Ryu, S., Shenoy, K.: Neural population dynamics during reaching. Nature 487, 51–6 (06 2012)
9. Dayan, P., Hinton, G., Neal, R., Zemel, R.: The Helmholtz machine. Neu. Comp. 7, 1022–1037 (1995)
10. Farina, D., Vujaklija, I., Sartori, M., Kapelner, T., Negro, F., Jiang, N., Bergmeister, K., Andalib, A., Principe, J., Aszmann, O.: Man/machine interface based on the discharge timings of spinal motor neurons after targeted muscle reinnervation. Nature Biomedical Engineering 1 (2017)
11. Franzke, A., Kristoffersen, M., Bongers, R., Murgia, A., Pobatschnig, B., Unglaube, F., Sluis, C.: Users' and therapists' perceptions of myoelectric multi-function upper

limb prostheses with conventional and pattern recognition control. PLOS ONE 14, e0220899 (08 2019)

12. Ganzer, P.D., Colachis, S.C., Schwemmer, M.A., Friedenberg, D.A., Dunlap, C.F., Swiftney, C.E., Jacobowitz, A.F., Weber, D.J., Bockbrader, M.A., Sharma, G.: Restoring the Sense of Touch Using a Sensorimotor Demultiplexing Neural Interface. Cell pp. 1–11 (2020)
13. Grassia, F.S.: Practical parameterization of rotations using the exponential map. J. Graph. Tools 3(3), 29–48 (Mar 1998)
14. Grillner, S., Wallen, P.: Central pattern generators for locomotion, with special reference to vertebrates. Ann. Rev. Neurosci. 8(1), 233–261 (1985)
15. Grochow, K., Martin, S.L., Hertzmann, A., Popovic, Z.: Style-based inverse kinematics. ACM Trans. Graph. 23(3), 522–531 (2004)
16. Hahne, J.M., Schweisfurth, M.A., Koppe, M., Farina, D.: Simultaneous control of multiple functions of bionic hand prostheses: Performance and robustness in end users. Science Robotics 3(19) (2018), https://robotics.sciencemag.org/content/3/19/eaat3630
17. Ikemoto, L., Arikan, O., Forsyth, D.A.: Generalizing motion edits with Gaussian processes. ACM Trans. Graph. 28(1) (2009)
18. Jeka, J., Kelso, S.: Manipulating symmetry in the coordination dynamics of human movement. Journal of experimental psychology. Human perception and performance 21, 360–74 (05 1995)
19. Lau, M., Bar-Joseph, Z., Kuffner, J.: Modeling spatial and temporal variation in motion data. ACM Trans. Graph. 28(5) (2009)
20. Lawrence, N.: Learning for larger datasets with the Gaussian process latent variable model. J. Mach. Learn. Res. - Proceedings Track 2, 243–250 (2007)
21. Lawrence, N.D.: Large scale learning with the Gaussian process latent variable model. Technical report cs-06-05, University of Sheffield (2006)
22. Lawrence, N.D., Moore, A.J.: Hierarchical Gaussian process latent variable models. In: Proc. ICML. pp. 481–488. Omnipress (2007)
23. Lawrence, N.D., Court, R., Science, C.: Local Distance Preservation in the GP-LVM through Back Constraints. In: ICML. pp. 513 – 520 (2006)
24. Levine, S., Wang, J.M., Haraux, A., Popović, Z., Koltun, V.: Continuous character control with low-dimensional embeddings. ACM Trans. Graph. 31(4), 1–10 (jul 2012)
25. Li, X., Parizeau, M., Plamondon, R.: Training hidden Markov models with multiple observations-a combinatorial method. IEEE Trans. Pattern Anal. Mach. Intell. 22(4), 371–377 (2000)
26. Neal, R.: Bayesian Learning for Neural Networks. Ph.D. thesis, Dept. of Computer Science, University of Toronto (1994)
27. Rasmussen, C.E., Williams, C.K.I.: Gaussian processes for machine learning. J. Am. Stat. Assoc. 103, 429–429 (2008)
28. Sartori, M., Durandau, G., Dŏsen, S., Farina, D.: Robust simultaneous myoelectric control of multiple degrees of freedom in wrist-hand prostheses by real-time neuromusculoskeletal modeling. Journal of neural engineering 15(6) (10 2018)
29. Taubert, N., Endres, D., Christensen, A., Giese, M.A.: Shaking hands in latent space. In: Adv. in AI, LNCS. vol. 7006, pp. 330–334 (2011)
30. Urtasun, R., Fleet, D.J., Lawrence, N.D.: Modeling human locomotion with topologically constrained latent variable models (2007), workshop on Human Motion: Understanding, Modeling, Capture and Animation
31. Vujaklija, I., Shalchyan, V., Kamavuako, E., Jiang, N., Marateb, H., Farina, D.: Online mapping of emg signals into kinematics by autoencoding. Journal of NeuroEngineering and Rehabilitation 15 (03 2018)
32. Wang, J.M., Fleet, D.J., Hertzmann, A.: Multifactor Gaussian process models for style-content separation. In: Proc. ICML (2007)
33. Wang, J.M., Fleet, D.J., Hertzmann, A.: Gaussian process dynamical models for human motion. IEEE Trans. Pattern Anal. Mach. Intell. 30(2), 283–298 (2008)
34. Ye, Y., Liu, C.K.: Synthesis of responsive motion using a dynamic model. Computer Graphics Forum 29(2), 555–562 (2010)