

Learning Stable Task Sequences from Demonstration with Linear Parameter Varying Systems and Hidden Markov Models

José R. Medina and Aude Billard

Abstract: The problem of acquiring multiple tasks from demonstration is typically divided in two sequential processes: (1) the segmentation or identification of different subgoals/subtasks and (2) a separate learning process that parameterizes a control policy for each subtask. As a result, segmentation criteria typically neglect the characteristics of control policies and rely instead on simplified models. This paper aims for a single model capable of learning sequences of complex time-independent control policies that provide robust and stable behavior. To this end, we first present a novel and efficient approach to learn goal-oriented time-independent motion models by estimating *both attractor and dynamic behavior* from data guaranteeing stability using linear parameter varying (LPV) systems. This method enables learning complex task sequences with hidden Markov models (HMMs), where each state/subtask is given by a stable LPV system and where transitions are most likely around the corresponding attractor. We study the dynamics of the HMM-LPV model and propose a motion generation method that guarantees the stability of task sequences. We validate our approach in two sets of demonstrated human motions.

Keywords: Dynamical systems, Stability, Linear Parameter Varying Systems

1 Introduction

Learning from Demonstration (LfD) has become one of the most effective tools to enable robot autonomy. The majority of work in this field has been devoted to encoding demonstrations of a *single* task into a controller that synthesizes accurate, robust and generalizable robotic behavior. In this context, Dynamical Systems (DS) have become the most popular method due to their ability to provide behavioral guarantees. Specifically, if defined properly, DSs ensure convergence to a *pre-specified* attractor or goal (stability). Recently, many researchers have focused on extending these models from a single task to more complex sequences of tasks. A typical approach to this problem is the application of a two-steps: (1) a segmentation algorithm is first applied to acquire potential subtask demonstrations and (2) a stable DS is learned from the resulting segments assuming that end-points determine the attractors. However, segmentation algorithms rely on oversimplified linear models that neglect the properties of the DS considered. In addition, assuming that the end-point of a potential segment is an attractor might not hold, e.g. when data does not exhibit convergence (a bad segmentation point). An unified way of representing multiple sequential control policies is to model the currently active subtask with a discrete latent variable such as in a Hidden Markov Model (HMM): the hidden state indicates the current subtask and each subtask is given by a different parameterization of a DS. This way, subtasks are learned jointly without explicitly assuming segmentation points. Following this idea, HMMs have been successfully applied with simple Gaussian [1], linear Gaussian [2] or potentially non-stable dynamical system policies [3]. However, learning HMMs with *stable* complex nonlinear subtasks becomes more challenging. In fact, the standard learning procedure of stable goal-oriented motions prevents from learning such a unified model as it requires knowing the attractor *a priori*.

In this work, we first present a novel method to learn asymptotically stable goal-oriented motions capable of estimating the dynamic behavior and the attractor/goal from demonstrations. We param-

eterize motions as linear parameter varying (LPV) systems by means of Gaussian Mixture Models (GMM). The most likely parameters are learned applying the Expectation-Maximization (EM) algorithm and relying only on convex optimization programs. To represent complex task sequences, we formulate an HMM where each state/subtask is given by a stable LPV system and where transitions between subtasks are most likely around the corresponding attractor, which guarantees convergence of the full sequence. We validate this approach in a simulated experiment and in a set of recorded human motions.

Related work: Learning autonomous DSs has been a recurrent problem since the early works from [4] and [5]. More recently, the SEDS approach [6] ensures asymptotic stability to a previously known attractor and is also applicable to second-order systems [7] and incremental learning settings [8]. All aforementioned methods assume that the attractor is known a priori and is not part of the estimation process. In [9], the attractors of Dynamic Motor Primitives (DMPs) are estimated based on a previously trained library assuming simplified linear models. In contrast, the learning method proposed in this work estimates the attractor and dynamic behavior only from data in a time-independent manner and with no prior knowledge about the underlying primitive.

A body of work has been devoted to learning task sequences. While some early works exploit mixtures of autonomous DSs to synthesize complex behavior [10], most algorithms rely on an initial segmentation step followed by a model learning phase. In [1], a segmentation algorithm provide potential primitives which are learned and clustered by means of HMMs and their sequence is determined by a graph. Other segmentation algorithms for manipulation rely on contact information [11, 12]. A more informed segmentation that considers changes of potential constraints is presented in [13] to later fit hybrid position/force controllers with the resulting segments. Although the sequencing policy is typically represented by a graph, more sophisticated conditions can be extracted from data. In [14] a set of pre- and post-conditions is identified thereby learning a Petri net. Other works rely on linear policies [2] or directional normal distributions [15] as state emissions of HMMs [16] that consider state-dependent termination policies [17] for each state. The resulting model is later used to train movement primitives for each subtask. An interesting alternative is proposed in [18], where, given a set of potential segmentation points, the most probable ones are chosen depending on the model later used for control. In contrast to previous work, our proposed approach directly fits an HMM where each state is given by a stable LPV system. The capability of estimating the attractor from data enables a solution that considers all subtasks jointly given an initial guess.

The rest of this paper is organized as follows. Section 2 presents and evaluates the LPV model. Section 3 introduces and validates the HMM where each subtask is given by a stable LPV model. A discussion and concluding remarks are given in Section 4.

2 Learning asymptotically stable LPV systems with unknown attractor

Autonomous DSs represent motions as $\dot{\xi} = f(\xi)$, where $\xi \in \mathbb{R}^n$ represents the system's state and $f : \mathbb{R}^n \mapsto \mathbb{R}^n$. A suitable option to encode nonlinear systems is given by a mixture of C local linear DSs

$$\dot{\xi} = - \sum_{c=1}^C h_c(\alpha)(A_c \xi + b_c), \quad (1)$$

where $-A \in \mathbb{R}^{n \times n}$ is the dynamics matrix, b_c the bias and $\alpha \in \mathbb{R}^m$ is a measurable external parameter that defines the mixing coefficient $h_c(\alpha)$, $\sum_{c=1}^C h_c(\alpha) = 1$ and $h_c(\alpha) > 0$. In this case, if

$$A_c + A'_c \succeq 0, \quad \xi^* = A_c^{-1} b_c \quad \forall c = 1 \dots C \quad (2)$$

the system asymptotically converges to the unique attractor ξ^* [6]. This is a specific instance of a stable LPV system (if $\alpha = \xi$ it is denoted *quasi-LPV* system) [19] and its parameterization can be estimated from data *if the attractor is known a priori* using GMMs [20] or fuzzy models [21, 22, 23]. However, when the attractor is unknown, the estimation of ξ^* becomes a challenging problem due to the nonconvexity arising from (2). Instead of considering (1), the main idea behind the method presented in this section is to avoid this dependency by considering its inverse, i.e. the mixture of

local *inverse* linear dynamical systems

$$\xi = \sum_{c=1}^C h_c(\alpha)(\xi^* - A_c^{-1}\dot{\xi}) = \xi^* - \sum_{c=1}^C h_c(\alpha)A_c^{-1}\dot{\xi}$$

This alternative representation simplifies the identification of LPV and quasi-LPV systems when the attractor ξ^* is unknown, as it becomes a bias in the model.

Problem formulation: Given a set of M i.i.d. samples $\{\xi_i, \dot{\xi}_i, \alpha_i\}_{i=1}^M$ the problem considered in this section is the estimation of the dynamic behavior represented in the observations assuming that it is globally asymptotically stable and has a single *unknown* attractor ξ^* .

2.1 Modeling asymptotically stable LPV systems with GMMs

The model presented in this section assumes that the joint distribution of α and the observed dynamic behavior represented by $\xi, \dot{\xi}$ is given by a GMM. Specifically, observations are distributed as

$$P\left(\begin{bmatrix} \alpha \\ \dot{\xi} \end{bmatrix} \middle| \xi; \theta\right) = \sum_{c=1}^C P(c; \theta) P\left(\begin{bmatrix} \alpha \\ \dot{\xi} \end{bmatrix} \middle| \xi, c; \theta\right) \quad (3)$$

where $P(c; \theta) = \pi_c$ represents the prior and the conditional probability density function of each component

$$P\left(\begin{bmatrix} \alpha \\ \dot{\xi} \end{bmatrix} \middle| \xi, c; \theta\right) = P(\dot{\xi}|\xi, c; \theta) P(\alpha|c; \theta) \\ P(\dot{\xi}|\xi, c; \theta) = \mathcal{N}(-A_c(\xi - \xi^*), \Sigma_{c,\dot{\xi}}) \quad P(\alpha|c; \theta) = \mathcal{N}(\mu_{c,\alpha}, \Sigma_{c,\alpha}) \quad (4)$$

has dynamics matrix A_c , estimation noise $\Sigma_{c,\dot{\xi}}$ and $\mu_{c,\alpha}, \Sigma_{c,\alpha}$ denote the mean and variance of α respectively. Note that all mixture components share the same attractor ξ^* by design. The parameter set is given by $\theta = \{\xi^*, \theta_1, \dots, \theta_C\}$ where $\theta_c = \{\pi_c, \mu_{c,\alpha}, \Sigma_{c,\alpha}, A_c, \Sigma_{c,\dot{\xi}}\}$. For the sake of simplicity, in the following we will omit the dependency w.r.t θ .

For a standard LPV system, i.e. $\alpha \neq \xi$, equation (4) accordingly assumes that α is independent of the system's state or its derivative. In the case of a quasi-LPV system, i.e. $\alpha = \xi$, expression (4) reduces to the joint density $P\left(\begin{bmatrix} \dot{\xi} \\ \xi \end{bmatrix} \middle| c\right) = P(\dot{\xi}|\xi, c) P(\xi|c)$, as in the SEDS approach [6]. For later convenience, we also define the *inverse* linear dynamics of each component, which, under the asymptotic stability constraint are

$$P(\xi|\dot{\xi}, c) = \mathcal{N}(\xi^* - A_c^{-1}\dot{\xi}, \Sigma_{c,\epsilon}) \quad (5)$$

and, again, all components share the same attractor ξ^* .

2.2 Expected dynamics of the GMM

Given an observed pair $\{\alpha, \xi\}$, the dynamic behavior specified by the conditional distribution of $\dot{\xi}$ from (3) is also Gaussian, with mean

$$E[P(\dot{\xi}|\xi, \alpha)] = \sum_{c=1}^C h_c(\alpha)(-A_c(\xi - \xi^*)) \quad h_c(\alpha) = \frac{\pi_c \mathcal{N}(\alpha|\mu_{c,\alpha}, \Sigma_{c,\alpha})}{\sum_{j=1}^C \pi_j \mathcal{N}(\alpha|\mu_{j,\alpha}, \Sigma_{j,\alpha})}. \quad (6)$$

Note that $h_c(\alpha) \geq 0$ and $\sum_{c=1}^C h_c(\alpha) = 1$. Sufficient conditions for convergence of (6) are given in the following proposition.

Proposition 1 (Asymptotic stability [6]). Let $\epsilon \in \mathbb{R}$ be a small positive constant. Dynamics (6) globally asymptotically converge to the attractor ξ^* if

$$A_c + A_c^T \succ \epsilon I \quad \forall c = 1 \dots C. \quad (7)$$

2.3 Learning asymptotically stable LPV systems with the EM algorithm

Given a set of observations, the LPV parameters θ that maximize the likelihood are estimated by means of the expectation-maximization (EM) algorithm [24]. To ensure convergence of (6) we constrain the maximization problem with the set of convex constraints (7), yielding

$$\begin{aligned} \arg \max_{\theta} \sum_{i=1}^M \log P \left(\begin{bmatrix} \alpha_i \\ \dot{\xi}_i \end{bmatrix} \middle| \xi_i ; \theta \right) \\ \text{s.t.} \quad A_c + A_c^T \succ \epsilon I \quad \forall c = 1 \cdots C. \end{aligned} \quad (8)$$

The EM algorithm aims for a locally optimal solution of this problem by maximizing a lower bound of the log-likelihood in an iterative process. Let C_i be a latent variable representing the membership of the i -th observation. The *expectation step (E-step)* maximizes the distribution of C_i for current parameters θ , yielding the *responsibility* of each component

$$P(C_i = c) = \frac{P(c) P \left(\begin{bmatrix} \alpha_i \\ \dot{\xi}_i \end{bmatrix} \middle| \xi_i, c \right)}{\sum_{k=1}^C P(k) P \left(\begin{bmatrix} \alpha_i \\ \dot{\xi}_i \end{bmatrix} \middle| \xi_i, k \right)}. \quad (9)$$

Leveraging this distribution, the *maximization step (M-step)* computes the optimal parameters solving the now simplified optimization problem

$$\begin{aligned} \hat{\theta} = \arg \max_{\theta} \sum_{i=1}^M \sum_{c=1}^C P(C_i = c) \log P(c) P(\dot{\xi}_i | \xi_i, c) P(\alpha_i | c) \\ \text{s.t.} \quad A_c + A_c^T \succeq \epsilon I \quad \forall c = 1 \cdots C. \end{aligned} \quad (10)$$

The optimal $\{\hat{\pi}_c, \hat{\mu}_{c,\alpha}, \hat{\Sigma}_{c,\alpha}\}$ are computed in closed form, see [24]. To estimate attractor $\hat{\xi}^*$ and avoid nonconvexity, we approximate (10) considering inverse dynamics (5) yielding surrogate problem

$$\begin{aligned} \hat{\theta} = \arg \max_{\theta} \sum_{i=1}^M \sum_{c=1}^C P(C_i = c) \log P(c) P(\xi_i | \dot{\xi}_i, c) P(\alpha_i | c) \\ \text{s.t.} \quad A_c^{-1} + A_c^{-T} \succeq \epsilon_{\text{inv}} I \quad \forall c = 1 \cdots C, \end{aligned} \quad (11)$$

with $\epsilon_{\text{inv}} > 0$. The stationary point for the estimation noise covariance in terms of the constrained optimal inverse dynamic parameters $\hat{\xi}^*, \hat{A}_c^{-1}$ is given by

$$\hat{\Sigma}_{c,\epsilon} = \frac{\sum_{i=1}^M P(C_i = c) (\xi_i - \hat{\xi}^* + \hat{A}_c^{-1} \dot{\xi}_i) (\xi_i - \hat{\xi}^* + \hat{A}_c^{-1} \dot{\xi}_i)^T}{\sum_{i=1}^M \sum_{c=1}^C P(C_i = c)}.$$

Substituting this expression into (11), neglecting constant terms and assuming a diagonal covariance matrix, the constrained maximization step for $\hat{\xi}^*, \hat{A}_{1..C}^{-1}$ simplifies to the convex quadratic program

$$\begin{aligned} \hat{\xi}^*, \hat{A}_{1..C}^{-1} = \arg \max_{\xi^*, A_{1..C}^{-1}} \sum_{c=1}^C -\frac{1}{2} \text{tr}(\hat{\Sigma}_{c,\epsilon}) \left(\sum_{i=1}^M P(C_i = c) \right) \\ \text{s.t.} \quad A_c^{-1} + A_c^{-T} \succeq \epsilon_{\text{inv}} I \quad \forall c = 1 \cdots C. \end{aligned} \quad (12)$$

Although \hat{A}_c^{-1} is a valid estimate of the inverse of the forward dynamics matrix A_c of each component, a direct optimization of (10) considering the resulting $\hat{\xi}^*$ after solving (12) yields more accurate results. This requires an additional convex optimization step for $\hat{A}_{1..C}$ and $\hat{\Sigma}_{c,\epsilon}$ similar to (12) but considering forward dynamics (4). Given an initial θ , the EM algorithm iteratively applies the E-step (9) followed by the M-steps for $\{\hat{\pi}_c, \hat{\mu}_{c,\alpha}, \hat{\Sigma}_{c,\alpha}\}$ from (10), attractor $\hat{\xi}^*$ from (12) and linear dynamics matrix $\hat{A}_{1..C}$ and estimation noise $\hat{\Sigma}_{c,\epsilon}$ until convergence.

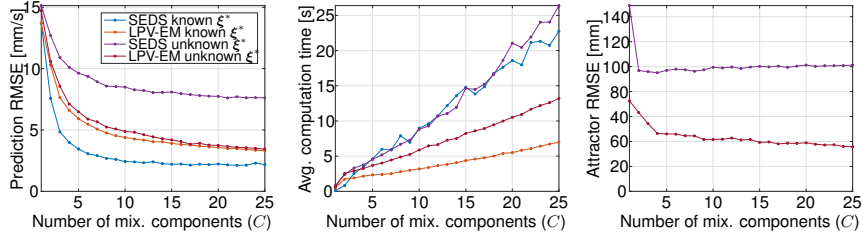


Figure 1: Results for the LASA handwriting dataset for SEDS and LPV-EM with and without knowing the attractor ξ^* . The left and center plots show the prediction RMSE and the the training time respectively for all four conditions. The right plot shows the attractor RMSE for conditions with unknown attractor.

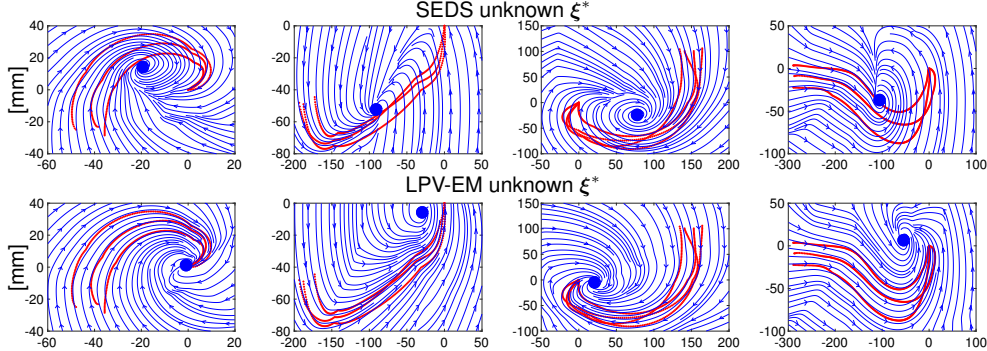


Figure 2: Streamlines and attractors of four recordings of the LASA handwriting dataset for SEDS (first row) and LPV-EM (second row) with unknown attractor ξ^* and $c = 7$ components. Red dots show the training samples and the blue marker indicates the estimated attractor.

2.4 Evaluation

We implemented our method in MATLAB using the `sedumi` [25] solver and the YALMIP [26] interface to solve (12). We compare our method *LPV-EM* with $\alpha = \xi$ to the SEDS method minimizing the mean squared error using the LASA handwriting dataset and the provided code from [6], which consists of 24 demonstrated two-dimensional trajectories of handwritten letters. We consider four conditions: SEDS and LPV-EM both with and without known attractor ξ^* . All conditions are evaluated between 1 and 25 components. For the EM algorithm, we initialize ξ^* with the average of observed states and the rest of parameters with the initial clustering given by k-means. We set $\epsilon_{\text{inv}} = 0.5$ and $\epsilon = 10^{-6}$. We evaluate our approach in terms of the root mean squared error (RMSE) of the estimated velocity computed as $\frac{1}{M} \sum_{i=1}^M \|\dot{\xi}_i - \hat{\xi}_i\|$, the RMSE of the estimation of the attractor ξ^* and the computation time.

In terms of the prediction RMSE, as shown in Fig. 1, when the attractor is known SEDS outperforms all other conditions as it directly minimizes the MSE. LPV-EM maximizes the whole likelihood including the distribution of α (which will be necessary in the next section) resulting in a slightly less accurate prediction. When the attractor is unknown, LPV-EM clearly outperforms the SEDS variant and the prediction performance of LPV-EM with and without knowing the attractor is almost identical. In terms of computation time, both LPV-EM conditions yield significantly faster computation times. Concerning the RMSE of the estimation of the attractor, the LPV-EM condition yields significantly better results. As illustrated in Fig. 2, when the attractor is unknown, the SEDS estimation might fall into local minima, while the LPV-EM variant achieves always consistent results due to the convexity of (12). In summary, the LPV-EM model is a fast alternative to SEDS when the attractor is known and it is the only reliable option when the attractor is unknown.

3 Learning Task Sequences with HMMs and stable LPV systems

In this section, to represent task sequences, we formulate an HMM where each subtask is given by an asymptotically stable quasi-LPV system. In addition, following recent work [2, 27], transitions between latent states (subtasks) depend on observations by means of a termination policy. We then study the dynamics of the *HMM-LPV* model and propose a learning and motion generation method that guarantees stability of the full sequence for left-to-right and periodic models.

Problem formulation: Given a set of demonstrated trajectories $\{\Xi_d = \{\xi_i, \dot{\xi}_i, \alpha_i\}_{i=1}^{T_d}\}_{d=1}^D$, the problem considered in this section is the acquisition of a dynamic model of the task assuming that it consists of a sequence of S goal-oriented globally asymptotically stable subtasks.

3.1 The HMM-LPV model

Let $s_i \in \{1, 2, \dots, S\}$ be a discrete latent variable representing a subtask at time step i and $b_i \in \{0, 1\}$ a *termination* binary variable that represents the event of finishing ($b_i = 1$) or not ($b_i = 0$) subtask s_i . The HMM-LPV model assumes that observed *trajectories* are distributed as

$$\begin{aligned} P(\Xi; \lambda) &= \left(\sum_{s_1=1}^S \sum_{b_{i-1}=0}^1 P(s_1; \lambda_\pi) P\left(\begin{bmatrix} \xi_1 \\ \dot{\xi}_1 \end{bmatrix} \middle| s_1; \lambda_e\right) \right) \\ &\prod_{i=1}^T \sum_{s_i=1}^S \sum_{s_{i-1}=1}^S \sum_{b_{i-1}=0}^1 P(s_i | s_{i-1}, b_{i-1}; \lambda_a) P(b_{i-1} | s_{i-1}, \xi_{i-1}; \lambda_b) P\left(\begin{bmatrix} \xi_i \\ \dot{\xi}_i \end{bmatrix} \middle| s_i; \lambda_e\right), \quad (13) \end{aligned}$$

with parameter set $\lambda = \{\lambda_\pi, \lambda_a, \lambda_b, \lambda_e\}$ and where

- initial subtask probabilities are $P(s; \lambda_\pi) = \lambda_\pi = \{\bar{\pi}_s\}$ for $1 \leq s \leq S$.
- subtask transition parameters $\lambda_a = \{a_{jk}\}$ for $1 \leq j, k \leq S$, with $j \neq k$, are such that in case of no termination $P(j|j, b=0; \lambda_a) = 1$ and in case of termination $P(j|k, b=1; \lambda_a) = a_{jk}$ and $P(j|j, b=1; \lambda_a) = 0$. As we are only considering left-to-right or periodic topology, if j precedes k then $a_{jk} = 1$.
- emission probabilities are given by a quasi-LPV systems, i.e. $\lambda_e = \{\theta_1 \dots \theta_S\}$ and $P\left(\begin{bmatrix} \xi \\ \dot{\xi} \end{bmatrix} \middle| s; \lambda_e\right) = P\left(\begin{bmatrix} \xi \\ \dot{\xi} \end{bmatrix}; \theta_s\right)$ as in (3) where $\theta_s = \{\xi_s^*, \theta_{s,1}, \dots, \theta_{s,C}\}$ and $\theta_{s,c} = \{\pi_{s,c}, \mu_{s,c,\alpha}, \Sigma_{s,c,\alpha}, A_{s,c}, \Sigma_{s,c,\xi}\}$.
- termination probabilities are given by an ξ -dependent Bernoulli distribution

$$P(b=1|s, \xi; \lambda_b) = \exp\{-(\xi - \mu_{b,s})^\top \Sigma_{b,s}^{-1} (\xi - \mu_{b,s})\} \quad (14)$$

with parameters $\lambda_b = \{\mu_{b,1}, \Sigma_{b,1} \dots \mu_{b,S}, \Sigma_{b,S}\}$. From the assumed distribution $P(b=0|s, \xi; \lambda_b) = 1 - P(b=1|s, \xi; \lambda_b)$.

3.2 Expected dynamics of the HMM-LPV model

At time step i and given t samples, i.e. $\{\xi_t\}_{t=1}^i$, the expected dynamics are

$$\dot{\xi}_{\text{hmm}} = E \left[P(\dot{\xi}_i | \{\xi_t\}_{t=1}^i; \lambda) \right] = \sum_{s=1}^S \tilde{h}_{s,i+1}(\xi_i) E \left[P(\dot{\xi}_i | \xi_i; \theta_s) \right], \quad (15)$$

where $E \left[P(\dot{\xi}_i | \xi_i; \theta_s) \right]$ is given by (6) and $\tilde{h}_{s,i+1}(\xi_i) = P(s_i | \{\xi_t\}_{t=1}^i; \lambda)$ is the forward variable computed as

$$\tilde{h}_{s,i+1}(\xi_i) = \sum_{s_i=1}^S \sum_{b_i=0}^1 \tilde{h}_{s,i}(\xi_{i-1}) P(s|s_i, b_i; \lambda_a) P(b_i | s_i, \xi_i; \lambda_b) P\left(\xi_i \middle| s_i; \lambda_e\right) \quad (16)$$

and is normalized such that $\sum_{s=1}^S \tilde{h}_{s,i+1}(\xi_i) = 1$. In this case, we only preserve the stability properties of the s -th subtask if $\tilde{h}_{s,i}(\xi_i) = 1$, when we recover dynamics (6). To ensure convergence of the whole sequence, we define s_{curr} as the *current* subtask and s_{next} as the *next* subtask with dynamics

$$s_{\text{curr},i+1} = \begin{cases} s_{\text{next},i} & \text{if } \tilde{h}_{s_{\text{next}}}(\xi_i) = 1 \\ s_{\text{curr},i} & \text{otherwise} \end{cases} \quad s_{\text{next},i+1} = \begin{cases} s_{\text{next},i} + 1 & \text{if } \tilde{h}_{s_{\text{next}}}(\xi_i) = 1 \\ s_{\text{next},i} & \text{otherwise} \end{cases}. \quad (17)$$

Conditions for the stability of transitions from s_{curr} to s_{next} are given in the following proposition.

Proposition 2 (Stability of transitions). Let $s_{\text{curr}}, s_{\text{next}}$ be the current and the next subtask respectively with dynamics (17) and let the conditional forward probability be computed considering these two states only, i.e. $S = \{s_{\text{curr}}, s_{\text{next}}\}$ in (16). Any trajectory that reaches the attractor $\xi_{s_{\text{curr}}}^*$ will converge to the subtask distribution $\tilde{h}_{s_{\text{curr}},i}(\xi_i) = 0, \tilde{h}_{s_{\text{next}},i}(\xi_i) = 1$ if

$$P(b=1|s_{\text{curr}}, \xi_{s_{\text{curr}}}^*; \lambda_b) = 1. \quad (18)$$

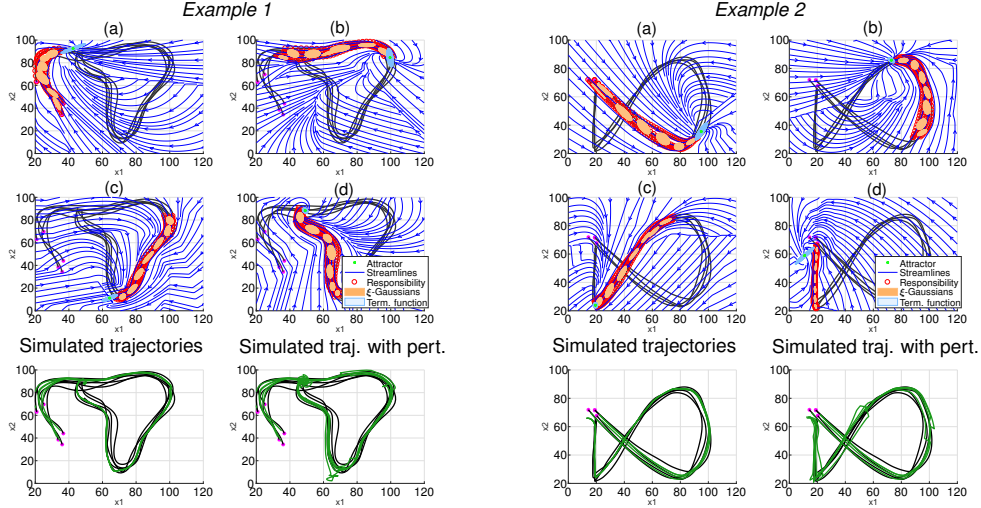


Figure 3: Resulting subtasks with sequence (a)-(b)-(c)-(d) and simulated trajectories for the HMM-LPV model with 4 subtasks and 7 GMM components for two exemplary sets of human demonstrations captured with a mouse. Black trajectories represent training samples and the pink asterisk indicates the initial positions. The first two rows depict the subtask parameters while the last row shows simulated trajectories depicted by the green solid lines generated following (19) with (left) and without (right) random perturbations every second.

Proof. Evaluating (16) for $S = \{s_{\text{curr}}, s_{\text{next}}\}$ at $\xi_i = \xi_{s_{\text{curr}}}^*$ and with (18) yields $\tilde{h}_{s_{\text{next}}, i}(\xi_i) = 1$. ■

To guarantee that $\xi_{s_{\text{curr}}}^*$ is reached, we modify the original expected dynamics (15) adding a stabilizing input [28] such that

$$\dot{\xi} = \dot{\xi}_{\text{hmm}} + \dot{\xi}_{\text{corr}} \quad \dot{\xi}_{\text{corr}} = \begin{cases} 0 & \text{if } l^T \dot{\xi}_{\text{hmm}} > 0 \\ -l^T \dot{\xi}_{\text{hmm}} l + \epsilon_{\text{corr}} l & \text{otherwise} \end{cases} \quad (19)$$

where $l = (\xi_{s_{\text{curr}}}^* - \alpha)$ and $\epsilon_{\text{corr}} > 0$. If Proposition 2 is fulfilled by every subtask, a left-to-right model following (19) is guaranteed to converge to the attractor of the last subtask, while a periodic model exhibits stable discrete limit cycle dynamics. Note that Proposition 2 provides very conservative conditions and transitions usually converge without intervention of the corrective input.

3.3 Learning an HMM-LPV model with the Baum-Welch algorithm

To estimate parameters from demonstrations we apply the Baum-Welch algorithm [29], the EM algorithm for HMMs considering conditions from Proposition 1 and 2 as

$$\arg \max_{\lambda} \sum_{d=1}^D \log P(\Xi_d; \lambda) \quad (20)$$

$$s.t. P(b = 1 | s, \xi_s^*; \lambda_b) = 1, A_{s,c} + A_{s,c}^T \succ 0 \quad \forall s = 1 \dots S, \forall c = 1 \dots C. \quad (21)$$

The E-step and the M-steps for λ_{π} , λ_{α} are detailed in [29] and [27]. The M-step for emission probabilities λ_e , is similar to (12) differing only in the responsibilities computed in the E-step. The M-step for the termination probabilities λ_b with RBF function (14) yields a similar problem structure to logistic regression [27] but with an ellipsoid boundary function and therefore a nonconvex objective. However, constraint (21) couples these two problems together and, intuitively, the maximization of λ_e drives solutions for λ_b towards regions where dynamics converge. Given an initial λ , the optimal parameters are computed applying the E- and M-step iteratively until convergence.

3.4 Validation

We implemented our approach in MATLAB using the FMINSQP solver [30] to solve the joint maximization of λ_b and λ_e from (20). We initialize the model parameters with k-means. With this initial clustering we apply the M-step of the LPV-EM to initialize λ_e while the covariance of the RBF termination function is initially set to the variance of the corresponding cluster. In our experiments, we set positive constants to $\epsilon_{\text{corr}} = 1$ in (19), and $\epsilon_{\text{inv}} = 0.5$, $\epsilon_{\text{inv}} = 10^{-6}$ in (12).

We first illustrate the capability of our model in two exemplary sets of 2-dimensional human motions captured with a mouse. As shown in Fig. 3 the HMM-LPV model is able to extract meaningful

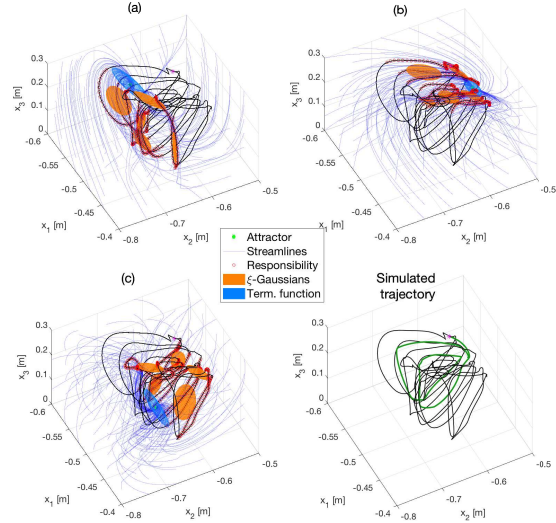
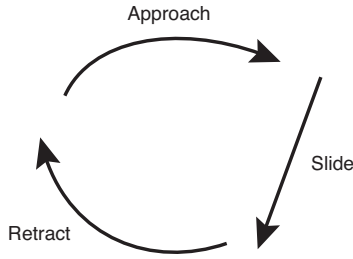


Figure 4: (left) Schematic of the repetitive task reproduced through kinaesthetic demonstrations. (right) Resulting subtasks with cyclic subtask sequence (b)-(c)-(a) and simulated trajectories for the HMM-LPV model with 3 subtasks with 5 GMM components each for the kinaesthetic teaching task. (a),(b) and (c) depict the parameters of each subtask. The bottom right plot shows the simulated trajectory generated following (19).

subtask dynamics and termination policies. Extracted attractors are typically at the end of the direction of motion, but in some cases, e.g. Example 1 (a) or Example 2 (a) they are placed further away as the corresponding data does not exhibit convergence. In both examples, generated trajectories without perturbations yielded no corrections from the stabilizing input. In simulations with disturbances the time-independent state-feedback nature of the model allows for immediate replanning and trajectories converge to the last attractor. To validate our approach in a more realistic setting, we run our approach on the motions obtained during a kinaesthetic teaching session where a human teacher drives a compliant passive robot to perform a repetitive task. The resulting subtasks are depicted in Fig. 4. The periodic topology of the model successfully captures the observed motions structure and the approach-slide-retract subtasks are represented by subtasks (b)-(c)-(a) respectively. Also, the simulated trajectory shows the state-dependent limit cycle behavior captured by the model, similar to the demonstrations. In summary, the HMM-LPV model is a suitable method to represent complex dynamic behavior with multiple attractors from data. The resulting dynamic policies are stable, time-independent and insensitive to perturbations.

4 Conclusions and Outlook

In this paper, we first presented a method to learn asymptotically stable motions from demonstrations based on the formulation of an LPV system as a GMM. The main novelty lies on the proposed LPV-EM algorithm, which robustly learns both attractor and dynamics from data. We believe that the LPV-EM algorithm has potential application in other domains that require nonlinear system identification and that has also interesting extensions, such as the addition of linear dependencies or constraints on the attractor for which the proposed M-step remains convex. This could be applied to interaction settings, where, the state of the interacting agent shifts the robot’s attractor.

The capability of the LPV-EM algorithm to estimate the attractor enabled the formulation of the HMM-LPV model, a dynamic model for complex task sequences where each subtask is given by a stable LPV system and transitions between subtasks are guaranteed by means of a state-dependent termination policy. All subtasks are learned jointly without strict segmentation bounds and the way subtasks transition is also learned from data. Although our validation shows promising results, performance strongly depend on the initialization values. In our evaluation k-means provided satisfactory results but more complex experiments would require more sophisticated initialization methods. An interesting future work is the application of our model to incremental and nonparametric settings [31], which would partially mitigate the initialization issue. Other interesting direction is the application of the HMM-LPV model as a stable but flexible closed loop policy for reinforcement learning settings [32].

Acknowledgments

We gratefully acknowledge the funding provided by the European Commission (Horizon 2020 Framework Programme, H2020-ICT-643950) for the SecondHands project. Thanks to Wissam Bejjani and Sina Miravazzi for their help with the implementation and the fruitful discussions. Thanks to Nadia Figueroa for providing the data of the zucchini experiment.

References

- [1] D. Kulić, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura. [Incremental learning of full body motion primitives and their sequencing through human motion observation](#). *The International Journal of Robotics Research*, 31(3):330–345, 2012.
- [2] O. Kroemer, C. Daniel, G. Neumann, H. Van Hoof, and J. Peters. [Towards learning hierarchical skills for multi-phase manipulation tasks](#). In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1503–1510. IEEE, 2015.
- [3] S. Calinon, F. D’Halluin, E. Sauser, D. Caldwell, and A. Billard. [Learning and Reproduction of Gestures by Imitation](#). *IEEE Robot. Autom. Mag.*, 17(2):44–54, 2010.
- [4] B. A. Pearlmutter. [Learning state space trajectories in recurrent neural networks](#). *Neural Computation*, 1(2):263–269, 1989.
- [5] J. Tani. [Model-based learning for mobile robot navigation from the dynamical systems perspective](#). *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(3):421–436, 1996.
- [6] S. M. Khansari-Zadeh and A. Billard. [Learning Stable Non-Linear Dynamical Systems with Gaussian Mixture Models](#). *IEEE Transaction on Robotics*, 2011.
- [7] S. S. M. Salehian, M. Khoramshahi, and A. Billard. [A Dynamical System Approach for Softly Catching a Flying Object: Theory and Experiment](#). *IEEE TRO*, 32(2):462–471, 2016.
- [8] K. Kronander, M. Khansari, and A. Billard. [Incremental motion learning with locally modulated dynamical systems](#). *Robotics and Autonomous Systems*, 70:52–62, 2015.
- [9] S. Manschitz, J. Kober, M. Gienger, and J. Peters. [Learning Movement Primitive Attractor Goals and Sequential Skills from Kinesthetic Demonstrations](#). *Robotics and Autonomous Systems*, 74, Part A:97–107, 2015.
- [10] J. Tani and S. Nolfi. [Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems](#). *Neural Networks*, 12(7):1131–1141, 1999.
- [11] J. Kober, M. Gienger, and J. J. Steil. [Learning movement primitives for force interaction tasks](#). In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 3192–3199, 2015.
- [12] Z. Su, O. Kroemer, G. E. Loeb, G. S. Sukhatme, and S. Schaal. [Learning to Switch Between Sensorimotor Primitives Using Multimodal Haptic Signals](#). In *International Conference on Simulation of Adaptive Behavior*, pages 170–182. Springer International Publishing, 2016.
- [13] A. Ureche, K. Umezawa, Y. Nakamura, and A. Billard. [Task Parameterization Using Continuous Constraints Extracted From Human Demonstrations](#). *Robotics, IEEE Transactions on*, 31(6):1458–1471, Dec 2015. ISSN 1552-3098. doi:10.1109/TRO.2015.2495003.
- [14] G. Chang and D. Kulić. [Robot task error recovery using Petri nets learned from demonstration](#). In *Advanced Robotics (ICAR), 2013 16th International Conference on*, pages 1–6. IEEE, 2013.
- [15] S. Manschitz, M. Gienger, J. Kober, and J. Peters. [Probabilistic decomposition of sequential force interaction tasks into Movement Primitives](#). In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 3920–3927. IEEE, 2016.

- [16] O. Kroemer, H. Van Hoof, G. Neumann, and J. Peters. [Learning to predict phases of manipulation tasks as hidden states](#). In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4009–4014. IEEE, 2014.
- [17] S. Manschitz, J. Kober, M. Gienger, and J. Peters. [Probabilistic progress prediction and sequencing of concurrent movement primitives](#). In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 449–455. IEEE, 2015.
- [18] R. Lioutikov, G. Neumann, G. Maeda, and J. Peters. [Probabilistic segmentation applied to an assembly task](#). In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 533–540. IEEE, 2015.
- [19] B. Bamieh and L. Giarre. [Identification of linear parameter varying models](#). *International journal of robust and nonlinear control*, 12(9):841–853, 2002.
- [20] X. Jin, B. Huang, and D. S. Shook. [Multiple model LPV approach to nonlinear process identification with EM algorithm](#). *Journal of Process Control*, 21(1):182–193, 2011.
- [21] G. Feng. [A survey on analysis and design of model-based fuzzy control systems](#). *IEEE Transactions on Fuzzy systems*, 14(5):676–697, 2006.
- [22] S.-G. Cao, N. W. Rees, and G. Feng. [Analysis and design for a class of complex control systems part I: fuzzy modelling and identification](#). *Automatica*, 33(6):1017–1028, 1997.
- [23] M.-T. Gan, M. Hanmandlu, and A. H. Tan. [From a Gaussian mixture model to additive fuzzy systems](#). *IEEE Transactions on Fuzzy Systems*, 13(3):303–316, 2005.
- [24] J. Bilmes. [A gentle tutorial on the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models](#). Technical report, U.C. Berkeley, 1997.
- [25] J. F. Sturm. [Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones](#). *Optimization methods and software*, 11(1-4):625–653, 1999.
- [26] J. Löfberg. [YALMIP : A Toolbox for Modeling and Optimization in MATLAB](#). In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.
- [27] C. Daniel, H. vanHoof, J. Peters, and G. Neumann. [Probabilistic inference for determining options in reinforcement learning](#). *Machine Learning*, 104(2):337–357, 2016. ISSN 1573-0565.
- [28] S. M. Khansari-Zadeh and A. Billard. [Learning Control Lyapunov Function to Ensure Stability of Dynamical System-based Robot Reaching Motions](#). *Robotics and Autonomous Systems*, 62(6):752–765, 2014.
- [29] L. Rabiner. [A tutorial on hidden Markov models and selected applications in speech recognition](#). *Proc. IEEE*, 77(2):257–286, 1989.
- [30] C. Thore. [FMINSDP a code for solving optimization problems with matrix inequality constraints](#), 2013.
- [31] I. Havoutis, A. Tanwani, and S. Calinon. [Online incremental learning of manipulation tasks for semi-autonomous teleoperation](#). In *IEEE/RSJ IROS*, 2016.
- [32] J. Rey, K. Kronander, F. Farshidian, J. Buchli, and A. Billard. [Learning motions from demonstrations and rewards with time-invariant dynamical systems based policies](#). *Autonomous Robots*, pages 1–20, 2017.