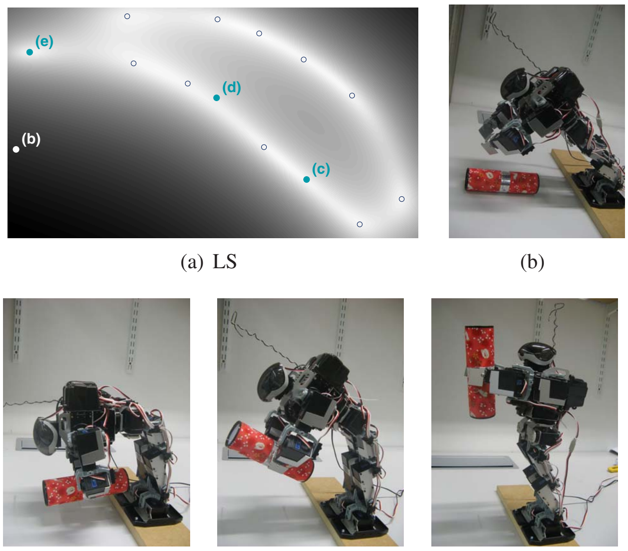
In Proc. 9th IEEE-RAS International Conference on Humanoid Robots (Humanoids '09), Dec. 7-10, Paris (2009).

Latent Spaces for Dynamic Movement Primitives

Sebastian Bitzer and Sethu Vijayakumar

***Abstract*— Dynamic movement primitives (DMPs) have been proposed as a powerful, robust and adaptive tool for planning robot trajectories based on demonstrated example movements. Adaptation of DMPs to new task requirements becomes difficult when demonstrated trajectories are only available in joint space, because their parameters do not in general correspond to variables meaningful for the task. This problem becomes more severe with increasing number of degrees of freedom and hence is particularly an issue for humanoid movements. It has been shown that DMP parameters can directly relate to task variables, when DMPs are learned in latent spaces resulting from dimensionality reduction of demonstrated trajectories. As we show here, however, standard dimensionality reduction techniques do not in general provide adequate latent spaces which need to be highly regular.**

In this work we concentrate on learning discrete (point-to-point) movements and propose a modification of a powerful nonlinear dimensionality reduction technique (Gaussian Process Latent Variable Model). Our modification makes the GPLVM more suitable for the use of DMPs by favouring latent spaces with highly regular structure. Even though in this case the user has to provide a structure hypothesis we show that its precise choice is not important in order to achieve good results. Additionally, we can overcome one of the main disadvantages of the GPLVM with this modification: its dependence on the initialisation of the latent space. We motivate our approach on data from a 7-DoF robotic arm and demonstrate its feasibility on a high-dimensional human motion capture data set.



(a) LS

(b)

(c)

(d)

(e)

Fig. 1. Example application of nonlinear DR: GPLVM learned from 12 demonstrated poses of KHR-1HV humanoid with 19 degrees of freedom on a lifting task. (a) shows the resulting latent space. (b-e) show the robot in poses generated from the depicted points in latent space. In (b), the robot drops the object, because task constraints (hands keep distance) are not fulfilled away from the data.

I. Introduction

Generating full-body humanoid motion still is an extremely difficult task. The inherent instability of an upright, 2-legged body makes it very hard to generate dynamically stable movements, but already designing efficient and natural movements for redundant bodies with many degrees of freedom *kinematically* is a challenge for the control engineer. Consequently it has been proposed to learn movements from demonstrations [1]. In this scenario we are given examples of movements which successfully solve the given task and the robot has to learn these movements and generalise them to new situations.

The Dynamic Movement Primitive (DMP) framework [11] provides representations particularly suited for robot programming by demonstration. DMPs are nonlinear dynamical systems which are learned such that an example movement is the attractor of the system. Hence, DMPs are control policies which can *robustly* replicate demonstrations. Additionally DMPs have parameters which allow to change the speed of the learned dynamics and shift and stretch it in its state space. These parameters can potentially be used to adapt a learned movement to new situations. However, when DMPs are learned in joint space, as originally proposed, it is not in general the case that a change in DMP parameters has an effect meaningful for the given task, or maintains other desirable properties of the demonstrated movements such as resolution of null space in a naturally looking way (see Fig. 2(a)). For low-DoF (degree of freedom) robots, for which inverse kinematics is easily solvable, a solution is to learn DMPs in task space [19] where DMP parameters directly relate to task variables. For full-body humanoid movements, however, learning how to resolve redundant degrees of freedom in a natural way is an important aspect of learning from demonstration.

In earlier work [3] we suggested to use dimensionality reduction (DR) to infer spaces from movement trajectories demonstrated in joint space in which DMP parameters can be related to task variables. This allows us to generate new motion which fulfils different task goals while maintaining the overall style of the demonstrated movements by a simple change of DMP parameters. While we showed that application of standard dimensionality reduction methods can lead to acceptable results (for example see Fig. 1), experiments also revealed that in many cases resulting latent spaces are not sufficiently regular to be used in combination with DMPs. This is, in particular, an issue with noisy, high-dimensional data from human motion capture, but it can already be observed for motion of a redundant 7-DoF robotic arm. In [4] we used such a system (the DLR light-

This work was funded under the EC FP6 project SENSOPAC (www.sensopac.org).

S. Bitzer (s.bitzer@ed.ac.uk) and S. Vijayakumar (sethu.vijayakumar@ed.ac.uk) are with the School of Informatics, University of Edinburgh, Informatics Forum, 10 Crichton Street, EH8 9AB, UK

weight robot arm) to extensively evaluate the capability of a range of dimensionality reduction methods to produce latent spaces which simplify point-by-point interpolation of robot poses - a problem less demanding than the one we consider here. These experiments showed that the usefulness of the latent space strongly depends on the data set, the method used and its parameters. Here we extend this analysis to the case in which we aim to interpolate whole trajectories by only changing parameters of a learned DMP and show that no standard DR method gives acceptable results. In section III, after introducing our formulation of DMPs, we consequently suggest a modification of the Gaussian Process Latent Variable Model which favours the use of DMPs in latent space by requiring it to be more regular. We present results on the DLR data set in section IV-A and finally evaluate our approach on human motion capture data in section IV-B.

II. Related work

The dynamic movement primitive framework is based on work presented by Ijspeert [11], [12]. A similar formulation with dynamical systems has been used to let a humanoid perform a drumming task [6] as a combination of discrete and periodic DMPs. However, there was no robot learning involved in this study and parameters for the DMPs had to be found by hand. DMPs have also been used as compact representations of movements for reinforcement learning [20], [9]. These approaches also aim to adapt a demonstrated movement to a new situation, but in order to do so they change the representation of the dynamical systems directly which leaves their parameters non-interpretable. This means that for each new situation the representation has to be tediously learned anew. Recently a reformulation of DMPs has been proposed [10] which is suited for applying DMPs in task space and allows for automatic obstacle avoidance. Potentially this formulation can equally be applied to latent spaces, but for this study we have chosen to use the formulation presented in section III-A instead.

So far there is only little published work on generating full-body humanoid movements based on demonstrations. One reason for this is that it is not clear where the demonstrations are supposed to come from. An apparent possibility is human motion capture. Even though there are suggestions for filtering human motion to become dynamically stable humanoid motion [18], [26], their assumptions or computational constraints make general applicability questionable. In [7] Grimes et al. propose to optimise an initial humanoid trajectory, which roughly corresponds to a human demonstration, based on predicted sensor feedback. Similarly the reinforcement learning approaches from above [20], [9] could be used to plan robot movements based on human demonstrations offline. In a recent paper Grimes et al. [8] use their single trajectory planner presented in [7] to generate movements for different task parameters. They then learn a control policy via nonparametric regression which generalises to a range of task parameters. The idea of this approach is very similar to ours except that we employ DMPs as control policies which are in our opinion better suited for robot control tasks because of their dynamical systems properties.

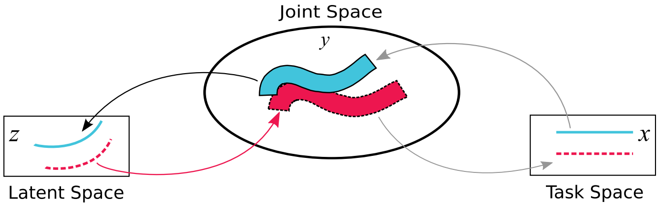


Fig. 3. Schematic of experimental setup: Given are demonstrations in joint space (blue). They have corresponding trajectories in task space, but we do not in general know them. We infer a latent space through DR in which DMPs are learned. Parameters of DMPs are changed to generate new trajectories (red) which are translated to joint space through the generative DR mapping.

Grimes et al. reduce the dimensionality of the humanoid state representation with PCA. This is a common step which has computational advantages. In our experience, however, PCA as a linear method is not able to faithfully represent joint angle motion in low dimensions. Many nonlinear DR methods have been proposed (e.g. [23], [22]), but in our application we need to generate motion from the low-dimensional (latent) space which can be done with latent variable models [14], [5], [13]. As basis for our study we chose the Gaussian Process Latent Variable Model [14], because of its powerful Gaussian Process mapping and the ease with which it can be modified. See section III-B for details on the GPLVM and its variants. We additionally compare results for the Gaussian Topographic Mapping (GTM) [2] and Unsupervised Kernel Regression (UKR) [17].

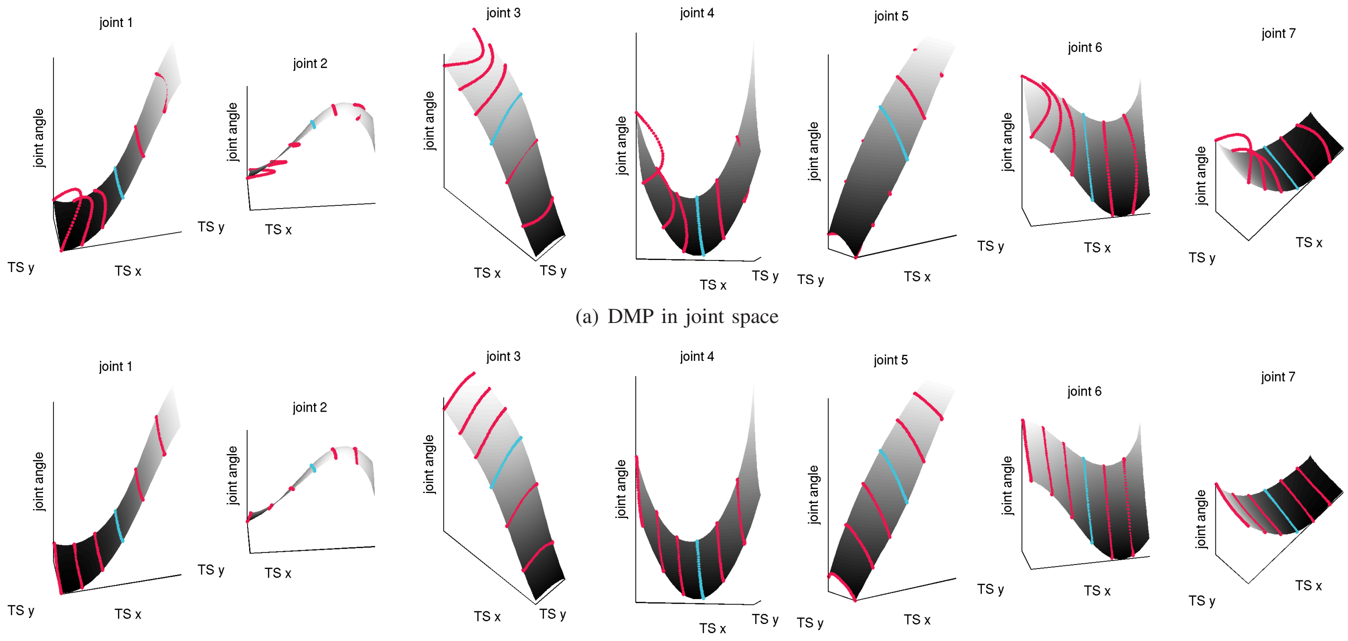
III. Methods

Fig. 3 shows a schematic of the different spaces involved. The task space positions ($\mathbf{x}$) are only known in control experiments. In general we observe joint angles ($\mathbf{y}$) and infer latent points ($\mathbf{z}$) with dimensionality reduction. We learn DMPs in latent space (blue, solid line) and change their parameters to generate new motion (red, dashed line) which should have a correspondent in task space. We present our formulation of DMPs below followed by our dimensionality reduction framework.

A. Dynamic Movement Primitives

We use discrete Dynamic Movement Primitives (DMPs) to represent goal-directed movements as attractors of nonlinear dynamical systems. We make minor adaptations of the formulation in [12] to explicitly incorporate modifiable start and end positions. Note that a DMP contains one independent dynamical system per dimension of the space in which it is learned - another argument for dimensionality reduction. The following is a description of a single dynamical system. It equally applies to all dimensions.

Discrete movements (e.g., reaching) are characterised by a starting state, z_0, some state trajectory and a goal state, g. The formalisation of such a system is shown in Table I. Ignoring the details of the modulating function f, this is a linear, two-dimensional dynamical system with a single, attracting stable point at $[g, 0]$. f is used to shape the trajectory of the dynamical system between z_0 and g. It can be represented as a weighted sum of RBF basis functions which depend on the state, ξ, of a canonical system that converges to 0. The number of basis functions, n and their width and



(a) DMP in joint space

(b) DMP in latent space

Fig. 2. Inverse kinematics for 7 DLR arm joints (function from task space x and y positions to joint angles). 2(a) The blue trajectory is generated from a DMP learned on demonstrated joint data. Red trajectories are generated from the same DMP, but with changed parameters: they do not lie on the task manifold and have a different shape. 2(b) Same as 2(a) except that DMP is learned and modulated in latent space resulting from the suggested DR method: trajectories lie on task manifold and have desired shape.

centres, h_i, c_i, are chosen a priori (wide range of choices gives good results). Given a complete movement $[\mathbf{z}, \dot{\mathbf{z}}, \ddot{\mathbf{z}}]$, the weights, w_i, of the nonlinear component are learned. Once the movement is learned (or encoded as a DMP with start state z_0^* and goal g^*), we can change the start state and goal to produce a qualitatively equivalent dynamics of motion in different parts of the state space of z.

TABLE I

DEFINITION OF DISCRETE MOVEMENT PRIMITIVES. THE DYNAMIC VARIABLE GOVERNING THE NONLINEARITY, ξ, CONVERGES TO 0. $\alpha_v, \beta_v, \alpha_\xi$ HAVE TO BE CHOSEN A PRIORI.

$$\frac{1}{\tau}\dot{v} = \alpha_v(\beta_v(g-z)-v) + \frac{g-z_0}{g^*-z_0^*}f(\xi)$$

$$\frac{1}{\tau}\dot{z} = v \qquad \frac{1}{\tau}\dot{\xi} = -\alpha_\xi \xi$$

$$f(\xi) = \xi\frac{\sum_{i=1}^{n}\Psi_i(\xi)w_i}{\sum_i \Psi_i(\xi)} \qquad \Psi_i(\xi) = \exp\left(-h_i(\xi-c_i)^2\right)$$

The parameters resulting from this formulation are τ, z_0 and g.[1] τ can be used to speed up or slow down the dynamical system. z_0 sets a desired start state and g a desired goal. By adapting z_0 and g new trajectories with the same dynamics can easily be generated without need for relearning or expensive computations. In experiments we *learn* a DMP on one trajectory, *modulate* it to fit the other trajectories in the data set by changing z_0 and g to the corresponding, known values and *interpolate* new trajectories by interpolating z_0 and g of learned and modulated trajectories.

B. *Gaussian Process Latent Variable Models*

The GPLVM learns one Gaussian Process (GP) [21] mapping from latent to data space per dimension of the data space, but with a common kernel based on the latent points. The positions of the latent points, $\mathbf{Z}$, are found by minimising the negative GP data ($\mathbf{Y}$) marginal log-likelihood

$$\mathbf{Z} = \arg\min_{\mathbf{Z}} \; -\log \mathrm{P}(\mathbf{Y}|\mathbf{Z}) \tag{1}$$

which serves as the objective function for the dimensionality reduction. Modifications of the GPLVM are mostly based on the adaptation of this function. For example, Lawrence and Moore [15] define hierarchical priors on $\mathbf{Z}$ with suitable adaptation of the likelihood to explicitly model hierarchical dependencies in the data dimensions. Wang et al. [25] introduce a dynamics prior on $\mathbf{Z}$ defined by an autoregressive GP-inspired mapping between consecutive latent points to achieve smooth transitions between points in a sequence (they call their models Gaussian Process Dynamical Models - GPDM). Unfortunately our experiments show that this kind of model is still not sufficient for generalisation of DMPs in latent space. Similarly, Lawrence and Quinonero-Candela [16] try to strengthen neighbourhood relations between points by constraining the mapping from data to latent space to be smooth as well. Consequently, the modification we present below is aimed at further constraining the latent positions in a useful way and thereby, regularising the GPLVM problem. A similar approach has been chosen in [24] where the focus is on periodic latent spaces, though.

1) Simple Sequence Prior: The basic idea of the simple sequence prior is to increase the structural constraints on the latent space by requiring that all observed sequences are

[1] $\alpha_v, \beta_v, \alpha_\xi$ are parameters governing the dynamics of the DMP which we keep fixed

translated versions of a template. This directly relates to how we use the DMPs: the dynamics of a DMP and therefore the shape of the trajectory it produces is learned on one example. Changes in z_0 and g move the DMP in its space, but its dynamics stays fixed. Consequently we amend (1) with

$$\mathbf{Z} = \arg\min_{\mathbf{Z}} \; -\log \mathrm{P}(\mathbf{Y}|\mathbf{Z})\mathrm{P}(\bar{\mathbf{Z}}) \prod_s \mathrm{P}(\mathbf{Z}_s|\bar{\mathbf{Z}}) \qquad (2)$$

where $\bar{\mathbf{Z}}$ contains the latent positions of the template sequence, $\mathrm{P}(\bar{\mathbf{Z}})$ is an autoregressive mapping as in [25] and $\mathrm{P}(\mathbf{Z}_s|\bar{\mathbf{Z}})$ is a GP mapping from template sequence to latent positions of observed sequence s. This mapping is constrained to be a translation with additive noise, i.e. GPs with mean function $\mathrm{m}_i(\bar{\mathbf{z}}) = \bar{z}_i$ and covariance function $\mathrm{k}_i(\bar{\mathbf{z}}, \bar{\mathbf{z}}') = 1 + \delta(\bar{\mathbf{z}}, \bar{\mathbf{z}}')\sigma^2$ where i is an index to latent dimensions, because we have one GP per dimension, and $\delta(\mathbf{x}, \mathbf{x}') = 1$ if and only if $\mathbf{x} = \mathbf{x}'$. This can be derived from the weight-space view of GP regression starting with $\mathbf{z} = \mathbf{A}\bar{\mathbf{z}} + \mathbf{b} = \mathbf{W}\hat{\mathbf{z}}$, $\hat{\mathbf{z}} = [\bar{\mathbf{z}}, 1]^\top$ and setting $\mathbf{A}$ to the identity matrix.

In this framework variables for the template, $\bar{\mathbf{Z}}$, and noise, σ, need to be added to the optimisation, but also the latent positions are more constrained. $\bar{\mathbf{Z}}$ can be initialised with the mean of the initialisations of the observed sequences. This formulation allows for learning of the template sequence during the optimisation process. Unfortunately we found in early experiments that this procedure does not usually lead to success, because the learned templates do not exhibit the desired temporal structure. Alternatively we suggest to fix the template to a predefined sequence. Similarly allowing σ to be adapted during optimisation does not give desired results, because the model tends to explain the data with a large amount of noise. See section IV-B.2 for a discussion on choices for $\bar{\mathbf{Z}}$ and σ. With these variables fixed the above formulation simplifies to a Gaussian prior over the latent positions. $\mathrm{P}(\bar{\mathbf{Z}})$ then is constant during optimisation and

$$\mathrm{P}(\mathbf{z}_{si}|\bar{\mathbf{z}}_{\mathbf{i}}) \sim \exp\left[-\frac{1}{2}(\mathbf{z}_{si} - \bar{\mathbf{z}}_i)^\top \mathbf{K}^{-1}(\mathbf{z}_{si} - \bar{\mathbf{z}}_i)\right] \qquad (3)$$

where $\mathbf{z}_{si}$ is a vector containing the positions of sequence s in latent dimension i and $\mathbf{K}$ is the covariance matrix described above with ones everywhere except in the diagonal where we have $1+\sigma$. For small σ this prior strongly favours latent representations of demonstrated sequences which have equal shape to the template sequence and are close to it, but their exact position and ordering among each other is not constrained.

The definition of the prior assumes that all sequences contain the same number of points as the template. This criterion could easily be relaxed by having different length templates for the different sequences (but still equal shape at the same position). In this paper we have alternatively chosen to subsample demonstrated human sequences such that all have the same number of data points. Because movements are usually sufficiently dense sampled, interpolating within a trajectory does not lead to significant errors.

In the following we shall call a GPLVM with simple sequence prior a SS-GPLVM.

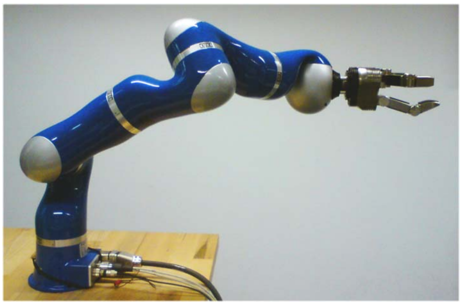


Fig. 4. 7-DoF DLR robot arm with Schunk Hand.

2) Probabilistic error feedback: Initially we also tried another modification which is based on the error between original trajectories in joint space and their correspondents generated by DMPs in latent space and then mapped to joint space. We implemented this by adding a probabilistic term based on the error to eq. (1). The idea is that the resulting gradient with respect to the latent points automatically contains a component which moves the latent points into positions compatible with the DMPs without making further assumptions. Unfortunately we found that this modification scales badly with the number of data points, but even more problematic is that instead of helping the GPLVM to overcome local optima issues through additional regularisation it makes the problem more severe and the optimisation converges prematurely.

IV. Experiments

The robotic data in the following section allows us to evaluate the generalisation capabilities of our method which motivates its application to high-dimensional movement systems such as the human motion capture data in section IV-B.

A. DLR LWR-III

These experiments are an extension of the experiments presented in [4], please see there for details of the setup. In short, we define a 2-dimensional task space for the DLR LWR-III (see Fig. 4) which consists of straight upwards movements of its end-effector along a line (see Fig. 5 for a visualisation). The DLR LWR-III is a robotic arm with 7DoFs, i.e. it has 1 redundant DoF. The redundancy is resolved in a consistent way in the inverse kinematics of the robot. We evaluate generated movements according to whether they follow the expected trajectory in the task space and whether they resolve redundancy in the defined way. If they do within a strict error margin, we say that they are successfully generated. In particular, we evaluate whether we can interpolate movements from demonstrations. In contrast to [4] interpolations are not done pose-for-pose, but by interpolating z_0 and g of a learned DMP[2] between the demonstrated trajectories. In total this data set contains 36 upward movements. To investigate how close movements need to be for interpolation to work we test our method by gradually increasing the distance between 2 demonstrated movements. This is implemented by skipping an increasing number of movements during selection of the data set. We

[2] the DMP is always learned on the 1st demonstrated trajectory in these experiments

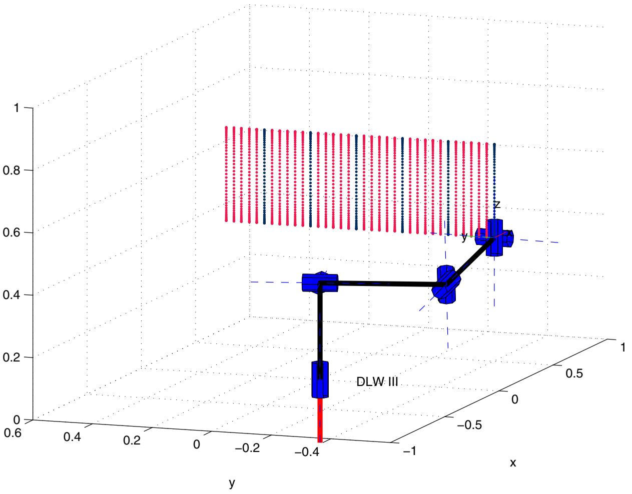


Fig. 5. DLR arm data set. Blue points: data trajectories, red points: trajectories which need to be interpolated. Data set corresponds to column 5 in Fig. 6.

call the number of skipped movements the interpolation width. Fig. 5 depicts an interpolation width of 5.

Fig. 6 summarises the accumulated results of our experiments. We tested learning and interpolation of DMPs in joint space (1st row) and in latent spaces resulting from different dimensionality reduction techniques (subsequent rows). Except for PCA and the GTM all tested techniques depend on the initialisation of latent points before optimisation. We used 6 initialisations (in that order): parallel lines[3], random, PCA, Isomap, LLE and Laplacian Eigenmaps. Our results show that no previously proposed DR technique produces latent spaces in which interpolation of DMP parameters generates trajectories which correspond to the desired trajectories in the underlying task space. Our SS-GPLVM[4], on the other hand, reaches a high percentage of successfull interpolations up to an interpolation width of 5 and partially beyond. We stress that this is a measure for how well the found latent spaces generalise to new movements of the same kind which we can only evaluate in this controlled robotic setup. Furthermore we note that while the GPLVM is known for being highly dependent on the initialisation of the latent points we achieve these results even with a random initialisation for the SS-GPLVM.

Fig. 2 illustrates the problems involved in this data set. Depicted is the inverse kinematics function for the 7 DLR joints that would need to be learned, if the task space coordinates were known. Fig. 2(a) shows the result of learning a DMP and varying its parameters in joint space. Due to the nonlinearities involved newly generated movements do not follow the task constraints and exhibit different dynamics. If the DMP is learned and modulated in a SS-GPLVM latent space as in Fig. 2(b) (interpolation width=5), however, generated movements fit very well to the desired.

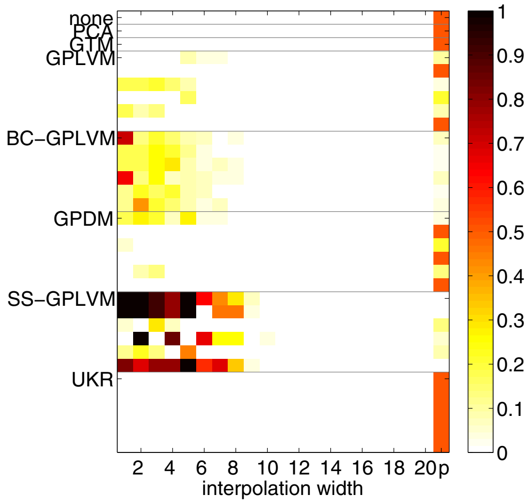


Fig. 6. Ratio of successfully interpolated trajectories using DMPs. 1 means all trajectories successfully interpolated. First line: DMPs operated in joint space, no dimensionality reduction. For GPLVM variants and UKR 6 different initialisations of latent points are tested (shown in this order): ad-hoc parallel lines, random, PCA, Isomap, LLE, LE. Last column: p-value for hypothesis that 'The mean ratio of successful interpolations is smaller or equal to the corresponding mean of the joint space interpolation'.

B. *Human motion capture data*

Encouraged by the results on the DLR data we apply our method to a data set of 3 punches executed by the same individual. We selected the punches from a motion capture database containing several martial arts movements requiring that they look similar while exhibiting variation in a prominent feature. In particular, the punches differ in the height of the trajectory of the punch hand. The movements have not been recorded with this application in mind and exhibit considerable more variation apart from the punch heights. Our aim is to generate similarly looking punches with intermediate punch heights.

The punches are represented as sequences of 57 dimensional poses consisting of 19 3D joint rotations represented with Euler angles in radians. The root node of the skeleton is fixed at the origin. We do not use constraints about physically plausible joint rotations during optimisation. Because the simple sequence prior requires sequences with equal length, we subsample 2 of the sequences with cubic spline interpolation from 27 and 37 poses, respectively, to 33 poses. This has no impact on the visual quality of the motions. The resulting 3 sequences all contain 33 poses which corresponds to 74.1, 53.5 and 60 poses per second, respectively. Consequently we do dimensionality reduction on a data set with 99 57-dimensional data points.

Again we want to interpolate between demonstrated movements. In contrast to the DLR data, however, we do not have ground truth in this case. We still can evaluate: a) The accuracy of the DR generative mapping, that is the error between a data point in joint space and its reconstruction from the corresponding latent point by means of the generative mapping (red arrow in Fig. 3). We call this the reconstruction error e_r. b) The discrepancy between a data sequence and the corresponding sequence generated from a modulated DMP, that is the DMP is learned on a different

[3] an ad-hoc initialisation motivated by the fact that we are handling discrete movements

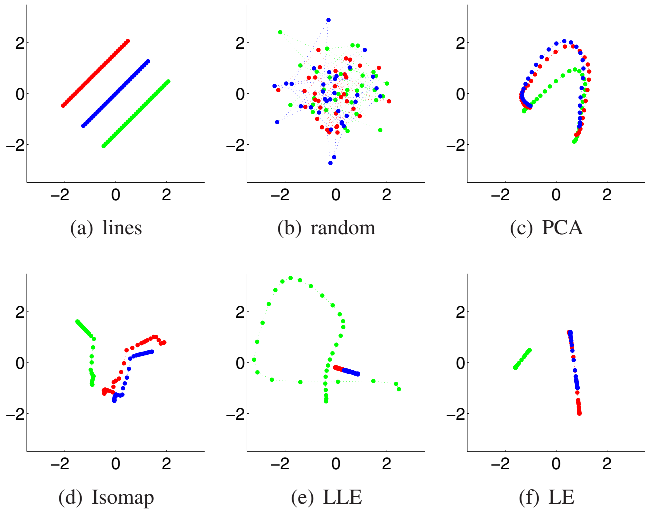
[4] $\bar{\mathbf{Z}}$ fixed to a line, $\sigma^2 = 0.001$

sequence, but its parameters z_0 and g are set to the values of the target sequence. An error occurs, when the dynamics (shape) of the learned sequence differs from the dynamics of the target sequence. We call the error the modulation error and it can be evaluated in latent space (e_m^{LS}) or joint space (e_m^{JS}) after making use of the DR generative mapping. e_m^{JS} is a combined measure of e_r and e_m^{LS}. Both, the reconsruction and modulation errors, only measure the fit of the model to the training data, but they give us a lower bound on the interpolation performance that we expect. In other words, if the modulation error is high, interpolation will not give desired results either. All errors are given as normalised mean square errors and reported as nMSE±standard deviation.

1) Existing dimensionality reduction techniques: First we evaluate the use of DMPs in joint space. The modulation error is $e_m^{\mathrm{JS}} = 0.75 \pm 1.11$. This is unacceptably high and, for example, means that some joints rotate for several periods. The resulting movement is completely unnatural. Because of its ease of use, PCA is a popular tool for preprocessing of motion capture data, but its representational power on this kind of sparse data sets is actually weak. The data reconstruction error for a 2D latent space is huge at $e_r = 11.05 \pm 16.78$. By increasing the number of latent dimensions the PCA representation becomes more accurate and e_r decreases. For a 10D latent space e_r drops to an acceptable 0.04 ± 0.05 while explaining 99% of the data variance. The modulation error, however, stays high for all latent dimensions (for 10D $e_m^{\mathrm{JS}} = 1.01 \pm 1.18$) meaning that PCA is no suitable DR method for our application.

PCA is the standard method for initialisation of the GPLVM. Additionally we use the same initialisations as with the DLR data set above. For Isomap (k=8), LLE (k=8) and LE (k=6, $\sigma_w = 0.19$) we have chosen parameters by searching for parameters leading to the smallest modulation error in latent space while requiring that the latent dimensions and sequences have comparable scale. In the random initialisation all points are distributed as N(0,1). The resulting configurations of latent points for all initialisations are depicted in Fig. 7.

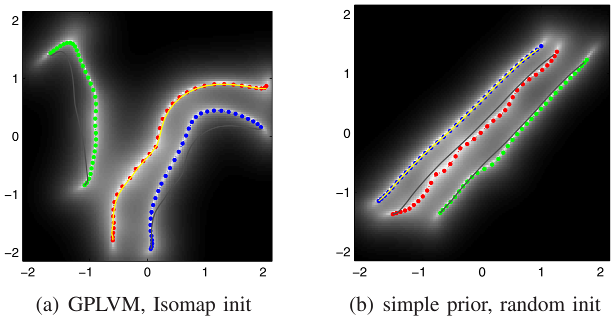
The GPLVM latent spaces are highly dependent on the chosen initialisation. We believe that the reason for this is that the capacity of the GPLVM method exceeds the complexity of the data. Even though based on probabilistic, nonparametric methods, the core procedure of the GPLVM is a maximum likelihood optimisation on, in our case, 202 highly nonlinearly related variables which is prone to run into local minima. This can be seen in Table II where we compare modulation errors e_m^{JS} for the different initialisations and methods. In our experiments, we learn the kernel parameters of the GPLVM, but do not learn output scales, because we find that this leads to bad results with some initialisations. For the same reason we do not learn the dynamics parameters in the GPDM. We define the GPDM dynamics on the differences of the latent points and use a rbf kernel with inverse width 1, variance 0.01 and white noise 1e-6. We also test a back constrained GPLVM for which we use kernel based regression (kbr) back constraints with rbf kernels with width 5. The implementations of all the GPLVM-based methods are either identical to or based on code generously provided by Neil Lawrence. We do not use sparse approximations and set the maximum number of gradient descent steps to 500 which is usually reached.



(a) lines (b) random (c) PCA

(d) Isomap (e) LLE (f) LE

Fig. 7. Different initialisations that we use on the mocap data. Green: high punch, blue: low punch, red: very low punch.



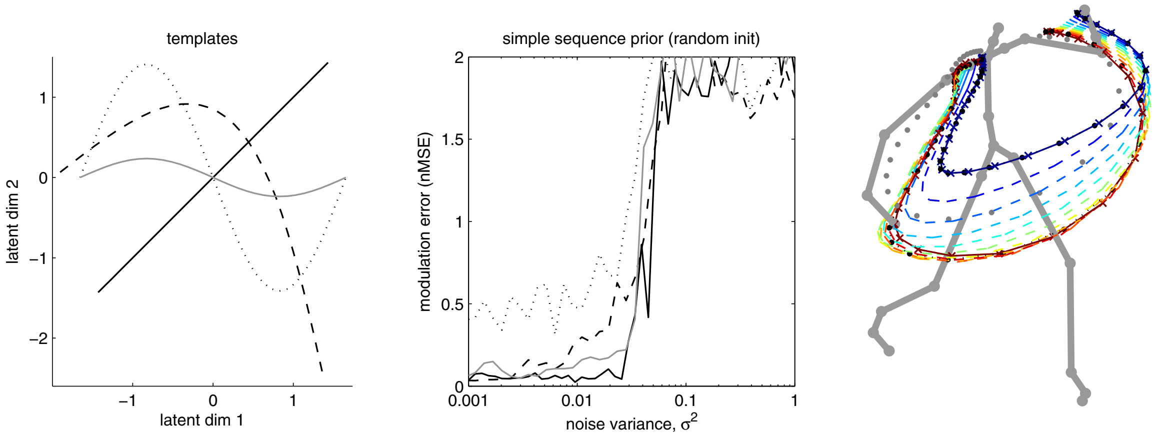
(a) GPLVM, Isomap init (b) simple prior, random init

Fig. 8. Selected latent spaces. Dots: Latent positions of data, colour coded as in Fig. 7. Solid lines: trajectories produced from DMPs, yellow: learned DMPs, dark grey: modulated DMPs. The shading corresponds to the log of the GP posterior output confidences at this point.

Although the data reconstruction error for GPLVM based approaches is usually very small (in the order of 10^{-6}), Table II shows that neither a standard GPLVM, a GPLVM with back constraints, or a GPDM can produce latent spaces sufficiently regular to allow for the representation of all punches with a single DMP. As an example, we have plotted the best latent space (GPLVM with Isomap initialisation) together with the learned and modulated DMPs in Fig. 8(a).

2) Simple Sequence Prior: Table II also shows results for the SS-GPLVM. It clearly outperforms all other methods in terms of joint space modulation error. Good results are largely independent of initialisation used as was already the case for the DLR data set. The movements generated from the modulated DMPs are very similar to the original movements. For these results we fixed template positions $\bar{\mathbf{Z}}$ to the mean of the lines initialisation and set $\sigma^2 = 0.001$. The data reconstruction error is very low at $0.003 \cdot 10^{-3} \pm 0.005 \cdot 10^{-3}$.

Not learning the parameters of the prior leaves the user with making a choice. How difficult is it to make that choice?



(a) Parameter selection

(b) Interpolation

Fig. 9. SS-GPLVM results. (a) Influence of parameter setting on performance. Left: different choices of templates for the prior. Right: Dependence of modulation error on choice of noise variance in prior for templates shown on left. (b) Trajectories of punch hand for interpolated DMP trajectories in latent space of Fig. 8(b). Shown are: original data of high and very low punches (black dots), original data of low punch (grey dots), DMP trajectory learned on high punch (blue crosses), DMP trajectory modulated to fit very low punch (red crosses) and interpolated DMP trajectories (dashed lines).

TABLE II

SUMMARY OF RESULTS (MODULATION ERROR IN JOINT SPACE: e_m^{JS}=NMSE$\pm\sigma$ OF TRAJECTORIES GENERATED FROM DMPS MODULATED IN LATENT SPACE). FOR COMPARISON, WITHOUT DR: 0.75 ± 1.11, WITH PCA: 5.38 ± 7.02

init	**GPLVM**	**GPLVM+back**	**GPDM**	**SS-GPLVM**
lines	0.233 ± 0.372	0.521 ± 0.885	0.241 ± 0.429	$\mathbf{0.022 \pm 0.038}$
random	2.080 ± 2.742	0.324 ± 0.577	12.461 ± 13.813	$\mathbf{0.037 \pm 0.061}$
pPCA	0.649 ± 1.030	0.422 ± 0.743	0.541 ± 0.831	$\mathbf{0.058 \pm 0.106}$
Isomap	0.221 ± 0.329	0.748 ± 1.349	0.411 ± 0.672	$\mathbf{0.040 \pm 0.080}$
LLE	1.304 ± 1.836	0.459 ± 0.796	1.043 ± 1.483	$\mathbf{0.019 \pm 0.033}$
LE	0.708 ± 0.963	0.894 ± 1.315	0.419 ± 0.645	$\mathbf{0.023 \pm 0.043}$

For the given template sequence we investigate the influence of the noise variance, σ^2. Fig. 9(a) shows the evolution of the modulation error for different settings of σ^2 and the random initialisation. We see that we get low errors as long as we choose σ^2 low enough. The exact choice is not important. The different graphs in the figure correspond to different choices for the template ($\bar{\mathbf{Z}}$ in eq. (2)) as shown in Fig. 9(a). Even though results with one choice of template are not acceptable, the data suggests that the exact choice of template is not important either.

A low modulation error e_m^{JS} only guarantees that we can use modulation of a DMP to accurately reconstruct all demonstrated movements, but it does not guarantee that a DMP with interpolated parameters also produces a movement that we would regard as interpolated with respect to the desired task. In particular, in our punch example, a movement interpolated between a high and low punch should result in a punch with intermediate height. Fig. 9(b) illustrates the result of such an interpolation experiment. We linearly interpolate z_0 and g between the high and very low punches and generate new trajectories in the latent space shown in Fig. 8(b). The figure depicts the punch hand trajectories of 9 interpolated punches together with the original data points and the learned and modulated DMP trajectories. The DMP is learned on the high punch.

There are a few points to note. 1) The trajectories result from equidistantly interpolated start and end points, but they are denser towards the very low punch. This means that the resulting latent space is still not linear with respect to the punch height, but it is nevertheless possible to very easily generate a continuum of punches between two examples. 2) The order of the punch sequences in latent space does not correspond to the punch heights. In particular, the very low punch is closer to the high punch than the low punch. Also none of the interpolations directly fits to the low punch. We believe that this is a consequence of the data resulting from differences in style of execution of the punches.

V. DISCUSSION

In this paper we point out that the parameters of discrete DMPs which are learned in joint space, as originally suggested, do not have a correspondence with underlying task variables. Indeed we show that a change of these parameters produces movements which neither follow the constraints set by the task, nor the consistent resolution of redundant degrees of freedom. Following previous work we suggest to use dimensionality reduction to provide a representation of demonstrated movements which overcomes these problems by making parameters of DMPs interpretable with respect to task variables while at the same time capturing regular structure in the data. Experiments, however, show that application of standard dimensionality reduction methods is not sufficient to generate meaningful new movements by only an adaptation of DMP parameters. Hence we suggest to modify the GPLVM by incorporating a simple sequence prior which strongly favours latent spaces reflecting the temporal and spatial structure of point-to-point movements (SS-GPLVM).

In a control experiment on data from a 7-DoF robot arm we demonstrate that only DMPs modulated in a SS-GPLVM latent space generate movements which accurately follow the expected trajectories as defined by task and null space constraints. As the results are obtained for interpolated DMP parameters, we can conclude that the proposed method also generalises to unseen movements which were not present in the data. The experiment also indicates that demonstrated movements can not be arbitrarily far away from each other. This is a well-known problem for dimensionality reduction methods. Note, however, that the SS-GPLVM still compares favourably to the other tested methods.

For human motion capture data we observe very similar results: we can show that a change of DMP parameters leads to unrealistic movements when the DMP is learned directly in joint space or a latent space resulting from standard dimensionality reduction methods. If the DMP is applied in a SS-GPLVM latent space, however, the original movements can be reconstructed just by changing DMP parameters. As we are missing ground truth in this data set, it is not possible to quantitatively evaluate movements generated through interpolation of DMP parameters. Instead we present interpolated trajectories which at least qualitatively correspond to our expectations.

We point out that good results can only be achieved with the simple sequence prior, if its parameters are fixed a priori. We also demonstrate, however, that making this choice is easy and that a wide range of parameter settings works well. Especially, the high tolerance to changes in the prior template is surprising, because it is the main factor determining the shape of the latent space. This suggests that the GP mapping from latent to joint space is so powerful that almost any latent space can serve as a reliable representation for the data. This hypothesis is backed up by the low data reconstruction errors for all GPLVM methods and initialisations. To control the tradeoff between highly regular structure and generalisation capabilities in latent space we suggest to set the noise parameter in the SS-GPLVM to the maximum value that gives acceptable modulation error.

In conclusion we present evidence that the SS-GPLVM provides latent spaces that allow fast, online generation of a continuum of new movements by a simple change of DMP parameters based on a small number of similar demonstrations. It is therefore a useful tool for robot programming by demonstration.

References

[1] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Chapter 59: Robot programming by demonstration. In B. Siciliano and O. Khatib, editors, *Handbook of Robotics*. Springer, 2008.

[2] C. M. Bishop, M. Svensen, and C. K. I. Williams. GTM: The generative topographic mapping. *Neural Computation*, 10(1):215–234, January 1998.

[3] S. Bitzer, I. Havoutis, and S. Vijayakumar. Synthesising novel movements through latent space modulation of scalable control policies. In *From Animals to Animats 10*. Springer, 2008.

[4] S. Bitzer, S. Klanke, and S. Vijayakumar. Does dimensionality reduction improve the quality of motion interpolation? In *Proceedings of the 17th European Symposium on Artificial Neural Networks (ESANN)*, pages 71–76, April 2009.

[5] M. A. Carreira-Perpinan and Z. Lu. The laplacian eigenmaps latent variable model. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics, AISTATS*, 2007. poster.

[6] S. Degallier, C. P. Santos, L. Righetti, and A. Ijspeert. Movement generation using dynamical systems: a humanoid robot performing a drumming task. In *IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS06)*, 2006.

[7] D. B. Grimes, R. Chalodhorn, and R. P. N. Rao. Dynamic imitation in a humanoid robot through nonparametric probabilistic inference. In *Proceedings of Robotics: Science and Systems (RSS'06)*, Cambridge, MA, 2006. MIT Press.

[8] D. B. Grimes and R. P. N. Rao. Learning nonparametric policies by imitation. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2008*, pages 2022–2028, 2008.

[9] F. Guenter, M. Hersch, S. Calinon, and A. Billard. Reinforcement learning for imitating constrained reaching movements. *Advanced Robotics*, 21(13):1521–1544, 2007.

[10] H. Hoffmann, P. Pastor, D.-H. Park, and S. Schaal. Biologically-inspired dynamical systems for movement generation: automatic real-time goal adaptation and obstacle avoidance. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[11] A. J. Ijspeert, J. Nakanishi, and S. Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1398–1403, 2002.

[12] A. J. Ijspeert, J. Nakanishi, and S. Schaal. Learning attractor landscapes for learning motor primitives. In *Advances in Neural Information Processing Systems, NIPS, 15*, 2003.

[13] A. Kanaujia, C. Sminchisescu, and D. Metaxas. Spectral latent variable models for perceptual inference. In *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV)*, 2007.

[14] N. Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816, 2005.

[15] N. D. Lawrence and A. J. Moore. Hierarchical gaussian process latent variable models. In *International Conference on Machine Learning, ICML*, 2007.

[16] N. D. Lawrence and J. Quinonero-Candela. Local distance preservation in the GP-LVM through back constraints. In *Proceedings of the International Conference in Machine Learning (ICML)*, 2006.

[17] P. Meinicke, S. Klanke, R. Memisevic, and H. Ritter. Principal surfaces from unsupervised kernel regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9):1379–1391, 2005.

[18] N. Naksuk, C. S. G. Lee, and S. Rietdyk. Whole-body human-to-humanoid motion transfer. In *Proc. 5th IEEE-RAS International Conference on Humanoid Robots*, pages 104–109, 2005.

[19] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal. Learning and generalization of motor skills by learning from demonstration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[20] J. Peters and S. Schaal. Reinforcement learning for parameterized motor primitives. In *2006 International Joint Conference on Neural Networks, IJCNN*, pages 73–80, 2006.

[21] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[22] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, Dec 2000.

[23] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, Dec 2000.

[24] R. Urtasun, D. J. Fleet, A. Geiger, J. Popovic, T. J. Darrell, and N. D. Lawrence. Topologically-constrained latent variable models. In *Proceedings of the 25th Annual International Conference on Machine Learning (ICML 2008)*, pages 1080–1087, 2008.

[25] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):283–298, Feb 2008.

[26] K. Yamane and Y. Nakamura. Dynamics filter - concept and implementation of online motion generator for human figures. *IEEE Transactions on Robotics and Automation*, 19(3):421–432, 2003.