Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 50-1 (2017) 11459–11464

A Segmentation Scheme for CLF Dynamic Movement Control Applied to Robotic Handwriting

Patrick Göttsch * **Robin Olschewski** * **Herbert Werner** *

* *Hamburg University of Technology, Institute of Control Systems, Eißendorfer Straße 40, D-21073 Hamburg, Germany (e-mail: {patrick.goettsch, robin.olschewski, h.werner}@tuhh.de).*

Abstract: Control Lyapunov Function - Dynamic Movement is a recently proposed imitation learning approach for controlling point-to-point movements. In this work some practical issues arising with this approach have been investigated by studying a particular application: robotic handwriting of German letters. One difficulty in this context is the handling of crossings. Here a novel segmentation scheme is proposed to eliminate crossings. Using a segmentation framework for splitting up the movement into a number of partial movements has additional advantages: It reduces the computational time for calculating the dynamic system that is required in this approach, as well as the time for calculating a Lyapunov function because of the simpler parametrization. This proposed segmentation scheme has been validated experimentally on a CRS A465 robotic manipulator.

Keywords: Robotics technology, Autonomous robotic systems, Robots manipulators, Lyapunov methods

1. INTRODUCTION

Bringing robots from factories to households and daily life is focus of current research in imitation learning, also known as programming by demonstration. A Dynamical System (DS) is learned from measured trajectory data. This DS can be abstracted, labelled and stored in a library and is often called Dynamical Movement/Motor Primitive (DMP) Perrin and Schlehuber-Caissier (2016). DMPs are used to asign target orientated tasks to robots Ijspeert et al. (2013).

An Advantage of such learned DS' is that they are robust against spatial perturbations because DS include not only the learned but also the unknown parts of the state space Mohammad Khansari-Zadeh and Billard (2014); Perrin and Schlehuber-Caissier (2016). Compared to classical approaches to solve a task by dividing it in two steps: planning and execution, Brock and Khatib (2000), this is done in DS-based approaches in a single step. This has the advantage that planning a new path in case of obstacle avoidance caused by a changing environment is easily done Schaal (2006); Pastor et al. (2009); Ijspeert et al. (2013); Khansari-Zadeh and Billard (2011); Mulling et al. (2013).

An important feature for the DS is that it guarantees that the target point will be reached. Two different approaches have been proposed: One was proposed by Schaal et al. and uses a second order differential equation with asymptotic stable equilibrium that has an external forcing term which is only enabled when the system moves from start point to the target point. The forcing term is cancelled out when reaching the target.

Another approach was presented in Khansari-Zadeh and Billard (2011), where stability conditions were provided when using Gaussian Mixture Model (GMM) for modelling the DS. Since for such a DS global asymptotical stability of the target point cannot be guaranteed they proposed conditions utilizing a quadratic Lyapunov function. As a result only a special class of movements can be modelled with this approach. To extend the amount of movements that can be modeled a method was presented in Mohammad Khansari-Zadeh and Billard (2014), where the constraint was removed using only GMR as regression method. Global asymptotical stability for a target point is ensured through a learned Control Lyapunov function that has its global minimum at the target point. The initial quadratic Lyapunov function is extended by asymmetric quadratic terms such that a more complex energy field can be modelled. This energy function then corrects online the trajectory.

Both methods have the drawback that they can only model low-complexity motions, which excludes e.g. rhythmic movements and movements with crossing trajectories in state space. One reason is the difficulty of finding a sufficiently parametrized Lyapunov function, another one is that the movements considered here are point-to-point methods. These methods find the next point by evaluating the DS at the current state, getting the next point in state space corrected by the Lyapunov function if necessary.

Extending the state space e.g. for the end effector coordinates of a robotic manipulator by its velocity, as proposed by Khansari-Zadeh and Billard (2011), is a solution to solve the previous problem. With this extension however it cannot be guaranteed that there is no crossing in state

2405-8963
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2017.08.1818

space, and it leads to higher computational effort because it doubles the state vector.

Movement Segmentation is also a topic in current robotics research. The aim of using Segmentation for splitting up movements are various. In Lemme et al. (2014) a self-supervised segmentation framework was proposed to setup and refine a learned MPL of decomposed DMPs. Segments can be added, adjusted or deleted based on the learned demonstrations. The OS-ELM presented in Liang et al. (2006) is used to learn DMPs from Demonstration. The segmentation of a demonstration is based on an iterative process trying to best fit MPs from MPL with parts of the demonstration. Starting from a single line MP, the approach generates new MPs by sequencing segments. These new MPs are used to be added to MPL or to adjust previously learned MPs.

For modelling more complex motions that are reproducible, different approaches have been proposed for unsupervised segmentation of point-to-point movements. Demonstrations are segmented by using different features like critical points Mohan et al. (2010), points without velocity or at minima in the velocity profile Lemme et al. (2014). More complex methods utilize for example Hidden Markov Models Asfour et al. (2008) or, if Gaussian Mixture Regression is used as regression method, each Gaussian as segment Lee et al. (2015). Here we take up the idea of using minima in the velocity profile for enabling the CLF-DM approach to model movements with crossings.

In this paper we propose a novel segmentation framework (SF). Our contributions are:

- With the proposed SF we are able to model more complex movements than proposed in Mohammad Khansari-Zadeh and Billard (2014)
- A simpler parametrization as without segmented movements
- Faster computation of the DS when learning and also while execution

This paper is structured in the following way: The idea of the CLF-DM approach is outlined in Section 2. In Section 3 the Segmentation Framework is presented. In Section 4 an experimental evaluation of how to parameterize the DS using our SF is shown. The paper is closed with a conclusion and an outlook in Section 5.

2. PRELIMINARIES

In Mohammad Khansari-Zadeh and Billard (2014) a continuous-time form of the Dynamic System Approach is presented. In this paper a discrete time version is given, because the implementation on robotic manipulators will be based on the discrete time version.

Here ξ is the state vector representing the end-effector coordinates to model a discrete-time nonlinear system as a movement primitive. The Dynamic System is formulated as

$$\xi_{k+1} = f(\xi_k) \quad f : \mathbb{R}^d \to \mathbb{R}^d \quad k \in [0 \quad T-1]\,. \tag{1}$$

The trajectory of this system can be determined recursively from (1). The starting point is given by $\xi^0 = \xi_k$ with $k = 0$, $\xi^* = \xi_k$ where $k = T-1$ denotes the target point and all points between are given by $\xi_k, k \in (0 \;\; T-1)$ where T is the number of sampling points. An advantage of CLF-DM approach is that learning can include a set $\mathcal{D}$ of N demonstrations: $\mathcal{D} = \left\{\xi^{t,n}, \dot{\xi}^{t,n}\right\}_{t=0,n=0}^{T^n,N}$. To generate an estimate of $f(\xi)$, denoted as $\hat{f}(\xi)$, any regression method can be used. In this paper GMR as a standard regression method is used for estimation of a Gaussian Mixture Model of the DS. In Calinon et al. (2006) GMM and GMR are explained in detail. The result of such an estimation is a set of parameters $\theta_{GMM}^g = \{\pi_g, \mu_g, \sigma_g\} \quad g \in [1 \quad \mathcal{G}]$. These are used in the estimated function

$$\hat{f}(\xi) = \sum_{g=1}^{\mathcal{G}} \pi_g \frac{1}{\sqrt{(2\pi)^N \cdot \|\sigma_g\|}} e^{-\frac{1}{2}\left((\xi_k-\mu_g)^\top \sigma_g^{-1}(\xi_k-\mu_g)\right)} \tag{2}$$

where $\top$ denotes the transpose; π_g, μ_g and σ_g are the prior, mean and covariance matrix for each of $\mathcal{G}$ Gaussians. The parameters are determined iteratively with an *Expectation Maximization* algorithm (Moon (1996)). $\mathcal{G}$ is a hand-tuned number that determines the accuracy of the estimate of the learned trajectory. Using GMR for estimating $\hat{f}(\xi)$ has the issue that global asymptotic stability against the target point ξ^* for $k \to T-1$ cannot be guaranteed. Such systems can run into spurious attractors and never reach the target point. Therefore the estimated system is extended by a stabilizing command input $u(\xi)$ to correct $\hat{f}(\xi)$ as

$$\xi_{k+1} = \hat{f}(\xi_k) + u(\xi_k). \tag{3}$$

Here $u(\xi, f(\xi))$ is applied only when the energy of the learned Lyapunov function $V(\xi)$ is increasing; this will be explained in detail below.

Definition 1. A Lyapunov function $V(\xi)$ is said to match the learned demonstrations, when it fulfils the following conditions:

(1) $V(\xi_k) > 0 \quad \forall \xi_k \in \mathcal{R}^d \setminus \xi^*$
(2) $V(\xi^*) = 0$
(3) $V(\xi^*) < V(\xi_k) \quad \forall k \in [0 \quad T-2]$
(4) $V(\xi_{k+1}) < V(\xi_k) \quad \forall k \in [0 \quad T-1]$

The first three statements are standard conditions for control Lyapunov functions. Statement (4) is required for ensuring that no correction is needed in the state space area that is covered by the demonstrations.

For this approach the energy function is extended by the sum of predefined $\mathcal{L}$ asymmetric quadratic terms (and is referred to as WSAQF)

$$V(\xi;\theta) = (\xi-\xi^*)^\top P^0 (\xi-\xi^*) + \sum_{l=1}^{\mathcal{L}} \beta^l(\xi;\theta)\left((\xi-\xi^*)^\top P^l (\xi-\mu^l-\xi^*)\right)^2 \tag{4}$$

where $\theta_V = \{P^0, \ldots, P^{\mathcal{L}}, \mu^1, \ldots, \mu^{\mathcal{L}}\}$ is the set of decision variables for an optimization problem. Using $\beta(\xi;\theta_V)$ as a switching term does not change that $V(\xi;\theta_V)$ is a $\mathcal{C}^1$ function. $\mu^l \in \mathbb{R}^d$ influences the shape of the asymmetric terms, $P^l \in \mathbb{R}^{d\times d} \succ 0$ are positive definite matrices, and $\beta(\xi;\theta_V)$ is defined as

$$\beta(\xi;\theta_V) = \begin{cases} 1 & \forall \xi : (\xi-\xi^*)^\top P^l (\xi-\mu^l-\xi^*) \geq 0 \\ 0 & \forall \xi : (\xi-\xi^*)^\top P^l (\xi-\mu^l-\xi^*) < 0 \end{cases} \tag{5}$$

A locally optimal solution for θ_V is determined by solving a constrained optimization problem which has two objectives: First reducing the number of data points for which (6) does not hold and second aligning the negative gradient of $V(\xi;\theta_V)$ with the direction of the trajectory.

Next $u(\xi) \in \mathbb{R}^d$ is determined such that

$$\begin{aligned} (\nabla_\xi V(\xi))^\top \left(\hat{f}(\xi) + u(\xi)\right) &< 0 \quad \forall \xi \in \mathbb{R}^d \setminus \xi^* \\ (\nabla_\xi V(\xi))^\top \left(\hat{f}(\xi) + u(\xi)\right) &= 0 \quad \xi = \xi^* \end{aligned} \tag{6}$$

where $u(\xi)$ is the solution of an optimization problem. A solution is given by

$$u^*(\xi) = \begin{cases} 0 \quad \forall \xi \in \mathbb{R}^d \setminus \xi^*, \nabla_\xi V(\xi)^\top \hat{f}(\xi) + \rho(\|\xi\|) \leq 0 \\ -\hat{f}(\xi) \quad \xi = \dot{\xi} \\ -\left(\dfrac{\nabla_\xi V(\xi)^\top \hat{f}(\xi) + \rho(\|\xi\|)}{\nabla_\xi V(\xi)^T \nabla_\xi V(\xi)}\right) \nabla_\xi V(\xi)^\top \quad \text{else} \end{cases} \tag{7}$$

Here ρ is introduced as a minimum decreasing rate for the energy function and allows to replace the strict inequality (6) with a non-strict onewhich is easier to optimize. The number of activations and the effort of u^* depend on $\hat{f}(\xi)$ and on how accurate it was estimated.

3. SEGMENTATION FRAMEWORK

When working with the DLF-DM approach for reproducing recorded lowercase handwritten German letters, it turns out that it is not possible to reproduce letters with crossing lines. A *b* was recorded five times with different starting points to extend the energy field of the learned control Lyapunov function, tackling the issue of having different connection types when connecting handwritten letters.

The results of these experiments were analysed. A problem arises when the letter contains a crossing. The reason is that at a crossing the next point is ambiguous. When the parameters of the energy function are evaluated, the algorithm drops one direction. It is not possible to create an energy function that fulfils the conditions of Def. 1 and has for the crossing point two different energy levels, which is needed to model trajectories with crossings. As shown in Fig. 1 one can see that the upper part of *b* is not reproduced. To enable MPL to store such letters, a segmentation approach was developed, which is explained detailed in this section. In Fig. 2 a flowchart illustrates the approach.

Starting with the *1. Record and Prepare Demonstrations* - Phase, N demonstrations were recorded for each letter using a tablet computer. The moving average over 20 values is calculated to first eliminate noise effects when measuring the position data of the tablet's pencil and second to smooth the trajectory. In a next step a spline for the z - coordinate, which is needed for some letters, is calculated. Currently only contact of the pencil with the screen can be measured. The last step in preprocessing is to add to each demonstration $X_n \in \mathcal{D}$ its first derivative calculated from

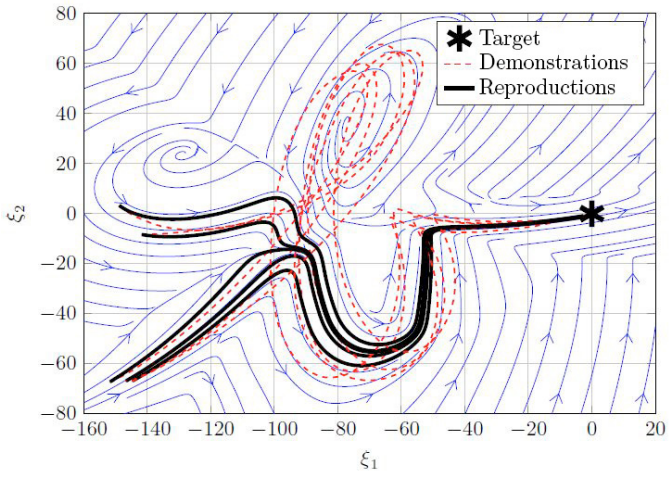


Fig. 1. Figure shows streamlines of learned GMM for five Demonstrations of a *b* without segmentation. The upper part of the *b* is not reproduced.

$$\dot{X}^n = \frac{1}{\Delta t}\left[(\xi^1 - \xi^0) \;\; (\xi^2 - \xi^1) \ldots (\xi^{T^n} - \xi^{T^n - 1})\right] \quad \forall n \in [1 \ldots \mathrm{N}], \tag{8}$$

such that we obtain $\hat{X}_n = (X_n \quad \dot{X}_n)^\top$.

Step *2. Segmentation* which can be summarized in the following three steps:

(1) Calculate the number of local minima for each demonstration
(2) Select only demonstrations with the modal value of the number of minima over all demonstrations
(3) Move the target-points of equal segments for each selected demonstration to a common target-point

For the segmentation the velocity profile for each demonstration is evaluated. The approach is based on the work by Senger et al. (2014), which has the advantage that no additional data needs to be calculated. The reason why the crossing points are not used are twofold: First from investigations in Lemme et al. (2015) it is known that human movements tend to have bell shaped velocity profiles. So they are hardly meaningful segments for the MPL. With an approach like Lemme et al. (2014) such segments can be improved and the whole library can be

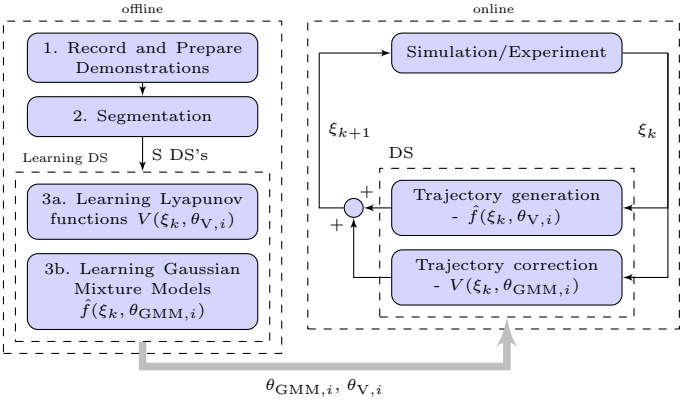


Fig. 2. Flowchart of the off-line learning part and the online reproduction part

minimized by composing segments in a different way. Second: The number of parameters needed for the GMM and the Lyapunov function can be reduced. This leads to less computational effort when calculating the parameters; this is discussed further in Section 4.

The velocity for each sampling point ξ^t is calculated by

$$v(t) = \|\dot{\xi}\| = \sqrt{\dot{\xi}^\top \dot{\xi}}. \tag{9}$$

From this the local minima are calculated. Only the first and the last minima are neglected because they are not needed for segmentation. Therefore from a demonstration $\mathcal{D} = \left\{\xi^t, \dot{\xi}^t\right\} \quad t \in [0 \quad T]$ only the interval $[R \quad T - R]$ is taken into account for determining the velocity minima. R is a constant number of measurements neglected at the beginning and at the end of each demonstration.

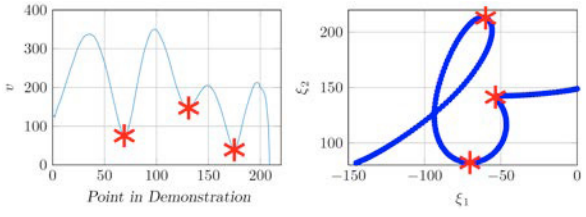
This is done for each of N demonstrations. The modal value of numbers of minima is determined over all N demonstrations. The number of segments S is one more than the number of minima. All demonstration with a number of segments not equal to S are not taken into account, because all segments should have the nearly the same shape. Based on observations this is true for demonstrations with the same S. For this number of demonstrations $\tilde{N}$ the target points are moved into the same point and no scaling is applied.

Now we have a set of $\tilde{N} \leq N$ segmented demonstrations with $S-1$ segmentation points and S segments as

$$\hat{\mathcal{D}} = \begin{pmatrix} \hat{X}_1^1 & \hat{X}_2^1 & \dots & \hat{X}_S^1 \\ \vdots & & \ddots & \vdots \\ \hat{X}_1^{\tilde{N}} & \hat{X}_2^{\tilde{N}} & \dots & \hat{X}_{S+1}^{\tilde{N}} \end{pmatrix}. \tag{10}$$

In the third step *3. Learning DS* the segmented dataset $\hat{\mathcal{D}}$ is feed to the learning algorithms to obtain for each segment the parametrization θ_V and θ_{GMM}. In Fig. 4 the result after step two is illustrated. Four segments of the letter b are shown. The two starting points are clearly to see and also moving each segment's target point to the same point can be easily seen. It also turns out, that each segment has a simple structure which is important for the choice of the parametrization of the energy function and the estimated DS.

As shown in the flowchart in Fig. 2, with step three the offline part of this approach is completed and all parameters are stored in the MPL, which now can be used in the online part to reproduce the handwritten letters.



(a) (b)

Fig. 3. Segmentation points are indicated by (∗). (a) shows the velocity profile of a demonstration of a b with segmentation points at its local minima. In (b) a reproduction of the same b as in (a) is shown.

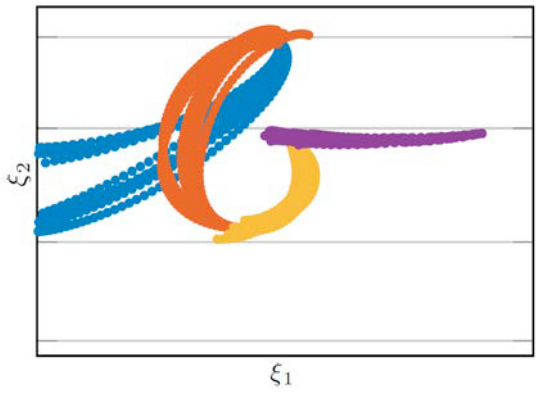


Fig. 4. $\hat{N} = 13$ out of $N = 19$ demonstrations with $S = 4$ segments. Each segment is indicated by a different colour.

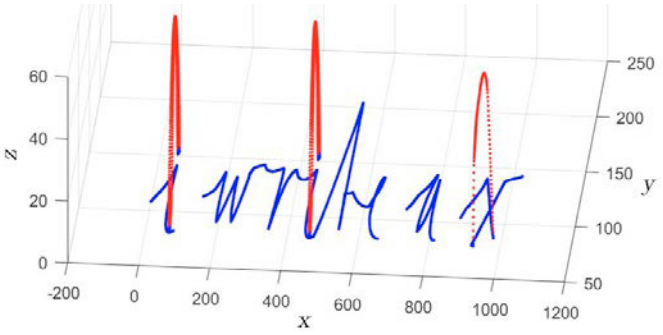


Fig. 5. 3D writing example using our SF. (····) movement in z direction, (—) written letter in xy plane.

To write a letter, the parametrization of each segment is loaded one after the other into the outer control circuit. For the first segment the width is loaded to determine the position of the target point of the segment. Then the movement is calculated by obtaining new positions from the stabilized DS until the end point ξ^* is reached. For each following segment the previous steps are repeated with an additionally coordinate transformation of ξ^0, ξ^t and ξ^* from the outer letter coordinate system to the inner segment coordinate system.

The procedure can easily be extended to write complete words and sentences by applying an additional coordinate transformation from word to letter coordinate system. An example is shown in Fig. 5 where the application of the whole framework is shown: We are not only able to write letters with crossings, but we can combine them into complete words where the connection of the letters is adjusted with respect to the height of the end point of the previous letter.

4. PARAMETER EVALUATION

As mentioned before one advantage of SF is the simpler parametrization of the DS and its stabilising energy function. In this section we show an efficient parametrization for the application of handwriting motion, and we show that using SF leads to a faster computation time.

There are three tuning parameters when applying the CLF-DM approach. These parameters are the number of Gaussian $\mathcal{G}$ in the GMM when applying GMR for

estimating $\hat{f}(\xi)$, the number of asymmetric quadratic functions $\mathcal{L}$ for modelling more complex energy levels in the Lyapunov energy function $V(\xi)$ and the number of Demonstrations N for each letter.

The influence of each parameter has been investigated. As input material all small handwritten letters were used. Each parameter was varied in the range: $\mathcal{G} \in [1\ 10]$, $\mathcal{L} \in [0\ 2]$ and N $\in \{5, 10, 20, 30\}$

As a measure of how accurate a reproduction is we calculate (as in Lemme et al. (2015)) for each demonstration a point-wise Root-Mean-Squared-Error for position and also for velocity:

$$\mathrm{RMSE}_{\mathrm{pos}} = \sqrt{\frac{1}{M}\sum_{i=0}^{M}\|x_{\mathrm{demo}}(i) - x_{\mathrm{DS}}(i)\|^2} \tag{11}$$

$$\mathrm{RMSE}_{\mathrm{vel}} = \sqrt{\frac{1}{M}\sum_{i=0}^{M}\|v_{\mathrm{demo}}(i) - v_{\mathrm{DS}}(i)\|^2}. \tag{12}$$

Because each demonstration has a different number of data points, the demonstrations and the DS are sampled with the same number M of sampling points. These points are distributed equally over time. The idea behind this that the execution time of each set of demonstrations must be the same. The starting and target point of the DS are set to the starting and endpoint of the demonstration.

With (11) the absolute distance between demonstration and reproduction is calculated. This is a measure of how accurate the letter is reproduced.

The RMSE for the velocity is calculated in (12) in the same manner to have a measure of the accuracy of velocity to be sure that the model has learned human writing motion.

After having $\mathrm{RMSE}_{\mathrm{pos}}$ and $\mathrm{RMSE}_{\mathrm{vel}}$ for each demonstration the averages over all demonstrations are determined.

From Fig. 6 it can be observed that the number of Gaussians has a significant impact on the accuracy in position an velocity. Which can be explained by the fact, that more Gaussians are need for representing more complex movements. And using SF leads to more but simple segments as depicted in Fig. 3b. If $\mathcal{G} \geq 4$ is chosen no more significant improvement can be achieved.

This observation does not change if the number of demonstrations N increases. Using $N = 5$ or $N = 10$ demonstrations is sufficient for building a MPL when applying the proposed framework.

Changing the number of weighted asymmetric functions $\mathcal{L}$ does not significantly increase the position and velocity accuracy. The results are similar to those from Fig. 6. The choice of $\mathcal{L} = 0$ leads to faster learning and evaluation of the energy function $V(\xi)$ when SF is applied in this set-up.

In Fig. 7 for all letters evaluated with the parametrization $N = 10$, $\mathcal{G} = 6$ and $\mathcal{L} = 0$ the average, the minimum and the maximum $\mathrm{RMSE}_{\mathrm{pos}}$ are illustrated. From this it can be observed that the average error for each letter is below 11 mm. The size of the written letters is between 68 to 145 mm. Thus we have an accurate representation in

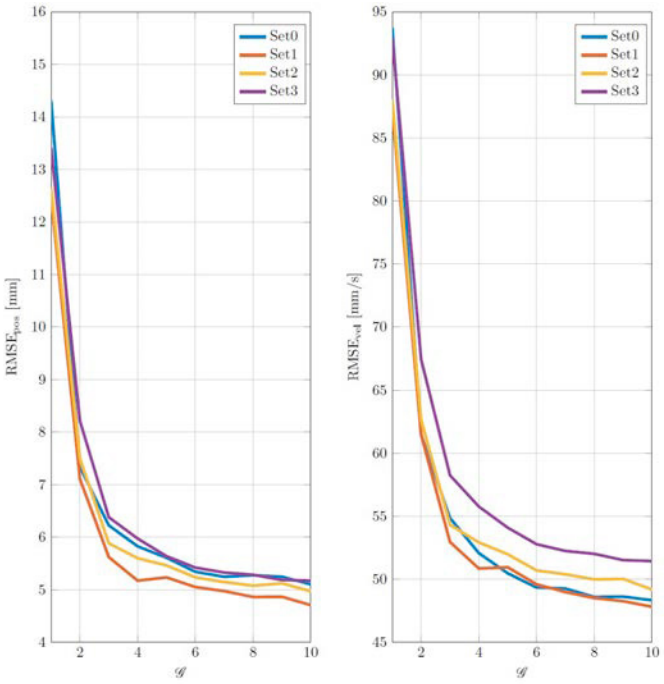


Fig. 6. Parameter evaluation for fixed $\mathcal{L} = 0$ with varied $\mathcal{G}$ on the abscissa and the RMSE of position (a) and velocity in (b) on the ordinate. (—) indicates a Set with $N = 5$, (—) $N = 10$, (—) $N = 20$ and (—) $N = 30$

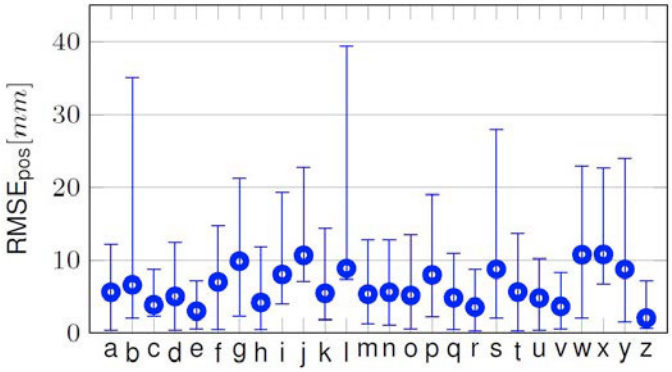


Fig. 7. For all 26 small letter the average over all demonstration, the minimal and the maximal $\mathrm{RMSE}_{\mathrm{pos}}$ are shown.

form of a library of DS, that models all small handwritten letters.

A parametrization with higher values of $\mathcal{G}$, $\mathcal{L}$ or N increases the calculation time of the parameters θ_{GMR} and θ, which is shown in Fig. 8. For both calculations a higher number of demonstrations increases the calculation time, which was performed on a *AMD Phenom(tm) II X4 965* processor with 8 GB RAM. The calculation time increases nearly linearly with the number of Gaussians and the gradient depends on N.

As shown enlarging $\mathcal{L}$ leads to a clearly higher calculation time and the gradient increases disproportionately.

Moreover, the accuracy in position and the computational time for two approaches have been investigated (Tretau (07.2016)). CLF-DM and the MoMP approach by Mulling

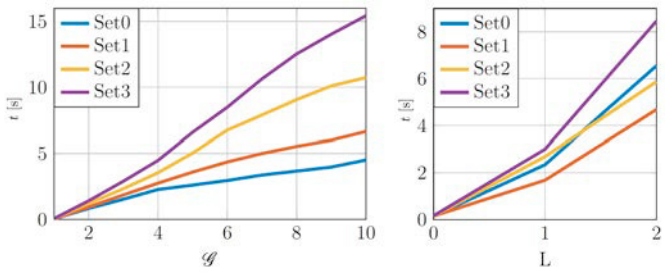


Fig. 8. Off-line calculation time for solving the optimization problem to find the parameters (a) θ_{GMM} and in (b) θ_V for the energy function.

et al. (2013) were used to calculate a reference trajectory based on a set of 80 demonstrations. The result is that the accuracy in position is 10 times better for the CLF-DM approach. The computation time for the CLF-DM approach is only 2% of that for the MoMP approach. These values were evaluated on the same system as before.

5. CONCLUSION & FUTURE WORK

Motivated by the problem that the CLFDM algorithm is not able to model movements including crossings,a Segmentation Framework was developed. This SF cuts demonstrations based on the velocity profile at their minima, inspired by human movements. This leads to a higher number of segments as only cutting at crossings, and has the advantage of a simpler parametrization for the estimation of the DS and also for the stabilizing Lyapunov energy function, which results into faster calculation time. This opens the way to model more complex movements without difficulties of finding a stabilizing energy function.

In future work force measurements and the orientation of the pencil will be integrated into the model, for example for reproducing Japanese letters. We also want to investigate how rhythmic movements can be incorporated into this approach. We will include constraints based on kinematic and dynamic limitations into the energy function to take this information into account.

Also a topic of interest is to make the segmentation framework adaptive, such that it generates more general segments which can be used by more than one MP. We are interested in applying this SF to a recently presented approach by Perrin and Schlehuber-Caissier (2016) to learn nonlinear DS by performing a new diffeomorphic matching algorithm.

REFERENCES

Asfour, T., Azad, P., Gyarfas, F., and Dillmann, R. (2008). Imitiation learning of dual-arm manipulation tasks in humanoid robots. *International Journal of Humanoid Robotics*, 05(02), 183–202.

Brock, O. and Khatib, O. (2000). Elastic strips: A framework for integrated planning and execution. *Experimental Robotics*, 250(VI), 329–338.

Calinon, S., Guenter, F., and Billard, A. (2006). On learning, representing and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man and Cybernetics, Part B. Special issue on robot learning by observation, demonstration and imitation*, 36(5).

Ijspeert, A.J., Nakanishi, J., Hoffmann, H., Pastor, P., and Schaal, S. (2013). Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2), 328–373.

Khansari-Zadeh, S.M. and Billard, A. (2011). Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Transactions on Robotics*, 27(5), 943–957.

Lee, S.H., Suh, I.H., Calinon, S., and Johansson, R. (2015). Autonomous framework for segmenting robot trajectories of manipulation task. *Autonomous Robots*, 38(2), 107–141.

Lemme, A., Meirovitch, Y., Khansari-Zadeh, M., Flash, T., Billard, A., and Steil, J.J. (2015). Open-source benchmarking for learned reaching motion generation in robotics. *Paladyn, Journal of Behavioral Robotics*, 6(1).

Lemme, A., Reinhard, R.F., and Steil, J.J. (2014). Self-supervised bootstrapping of a movement primitive library from complex trajectories*. In *14th IEEE-RAS International Conference on Humanoid Robots*.

Liang, N.Y., Huang, G.B., Saratchandran, P., and Sundararajan, N. (2006). A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 17(6), 1411–1423.

Mohammad Khansari-Zadeh, S. and Billard, A. (2014). Learning control lyapunov function to ensure stability of dynamical system-based robot reaching motions. *Robotics and Autonomous Systems*, 62(6), 752–765.

Mohan, V., Metta, G., Zenzeri, J., and Morasso, P. (2010). Teaching humanoids to imitate 'shapes' of movements. In K. Diamantaras, W. Duch, and L.S. Iliadis (eds.), *Artificial neural networks - ICANN 2010*, volume 6353 of *Lecture Notes in Computer Science*, 234–244. Springer.

Moon, T.K. (1996). The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6), 47–60.

Mulling, K., Kober, J., Kroemer, O., and Peters, J. (2013). Learning to select and generalize striking movements in robot table tennis. *The International Journal of Robotics Research*, 32(3), 263–279.

Pastor, P., Hoffmann, H., Asfour, T., and Schaal, S. (2009). Learning and generalization of motor skills by learning from demonstration. In *2009 IEEE International Conference on Robotics and Automation (ICRA)*, 763–768.

Perrin, N. and Schlehuber-Caissier, P. (2016). Fast diffeomorphic matching to learn globally asymptotically stable nonlinear dynamical systems. *Systems & Control Letters*, 96, 51–59.

Schaal, S. (2006). Dynamic movement primitives - a framework for motor control in humans and humanoid robotics. In *Adaptive Motion of Animals and Machines*, 261–280. Springer.

Senger, L., Schroer, M., Metzen, J.H., and Kirchner, E.A. (2014). Velocity-based multiple change-point inference for unsupervised segmentation of human movement behavior. In *22nd International Conference on Pattern Recognition (ICPR)*, 4564–4569.

Tretau, O. (07.2016). *Comparison of two Learning Approaches on Discrete Movements with a Classical Algorithm on Trajectory Planning for a Robotic Manipulator*. Master's thesis, Hamburg University of Technology, Hamburg.