Robot Learning

Winter Semester 2020/2021, Homework 4

Prof. Dr. J. Peters, J. Watson, J. Carvalho, J. Urain and T. Dam

TECHNISCHE UNIVERSITÄT DARMSTADT

Total points: 38
Due date: Monday, 01 February 2021 (23:59)
Übungsblatt 4

Aufgabe 4.1: Trajectory Generation with Dynamical Systems (38 Punkte)

In this exercise we will use the Dynamic Motor Primitives (DMPs), described by the following dynamical system,

$$\ddot{y} = \tau^2 \left(\alpha(\beta(g - y) - (\dot{y}/\tau)) + f_w(z)\right) \tag{1}$$

$$\dot{z} = -\tau \alpha_z z \tag{2}$$

where y is the state of the system, $\dot{y}$ and $\ddot{y}$ are the first and second time derivatives, respectively. The attractor's goal is denoted by g and the forcing function by f_w. The parameters α and β control the spring-damper system. The phase variable is denoted by z and the temporal scaling coefficient by τ. The forcing function f_w is given by

$$f_w(z) = \frac{\sum_{i=0}^{R} \phi_i(z) w_i z}{\sum_{j=0}^{R} \phi_j(z)} = \psi(z)^T w, \quad \text{with} \quad \psi_i(z) = \frac{\phi_i(z) z}{\sum_{j=1}^{K} \phi_j(z)} \tag{3}$$

where the basis functions $\phi_i(z)$ are Gaussian basis given by

$$\phi_i(z) = \exp\left(-0.5 \left(z - c_i\right)^2 / h_i\right) \tag{4}$$

where the centers c are equally distributed in the phase z, and the width h is an open parameter. For the programming exercises a basic environment of a double link pendulum is provided, as well as the computation of the $\psi_i(z)$.

4.1a) Similarities to a PD controller (2 Punkte)

Transform Equation (1) to have a similar structure to a PD-controller,

$$\ddot{y}_z = K_p \left(y_z^{\text{des}} - y_z\right) + K_D \left(\dot{y}_z^{\text{des}} - \dot{y}_z\right) + u_{ff} \tag{5}$$

and write down how the following quantities $K_p, K_d, y_z^{\text{des}}$ and $\dot{y}_z^{\text{des}}$ look like in terms of the DMP parameters. Do not expand the forcing function $f_w(z)$ at your solutions.

Lösungsvorschlag:

base on:

$$\ddot{y}_z = \tau^2 \alpha\beta(g - y_z) + (\tau\alpha)(0 - \dot{y}_z) + \tau^2 f_w(z) \tag{6}$$

* $K_p = \tau^2 \alpha\beta$
* $K_d = \tau\alpha$
* $y_z^{\text{des}} = g$
* $\dot{y}_z^{\text{des}} = 0$

Robot Learning Homework 4 Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

4.1b) Regression vs Classification (2 Punkte)

Explain why the DMPs are stable when $t \to \infty$ and what would the equilibrium point be.

Lösungsvorschlag:

$t \to \infty$ is identical as $s \to 0$ in Laplace-space, with Laplace-Transformation:

$$y(\infty) \Rightarrow \lim_{t\to\infty} y(t) = \lim_{s\to 0} s \cdot Y(s) \tag{7}$$

For DMPs:

$$\begin{aligned}
\ddot{y}_z &= \tau^2\alpha\beta(g - y_z) + (\tau\alpha)(0 - \dot{y}_z) + \tau^2\alpha f_w(z) \\
&\Downarrow \quad \textit{Laplace} \\
s^2 Y_z &= \tau^2\alpha\beta(G - Y_z) + (\tau\alpha)(0 - sY_z) + \tau^2\alpha F_w(z) \\
&\Downarrow \\
Y_z &= \frac{\tau^2\alpha\beta G + \tau^2\alpha F_w(z)}{s^2 + \tau\alpha s + \tau^2\alpha\beta} \\
&\Downarrow \\
\lim_{s\to 0} sY_z &= \frac{0}{\tau^2\alpha\beta} = 0 \\
&\Downarrow \\
&\textit{the DMPs are stable when } t \to \infty
\end{aligned}$$

On the other hand, $\dot{z} = -\tau\alpha_z z$ means the position z is an exponential decreasing function of time, which express as:

$$z = e^{-\tau\alpha\ t} \tag{8}$$

When $t \to \infty$, the translation z and the veloci ty$\dot{z}$ will be zero. Therefore the expression y under DMPs will be a total PD-controller, which is stable with positive controller quantities K_p, K_d.

In this case, the the equilibrium point will be: $\begin{pmatrix} y \\ \dot{y} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} g \\ 0 \\ 0 \end{pmatrix}$

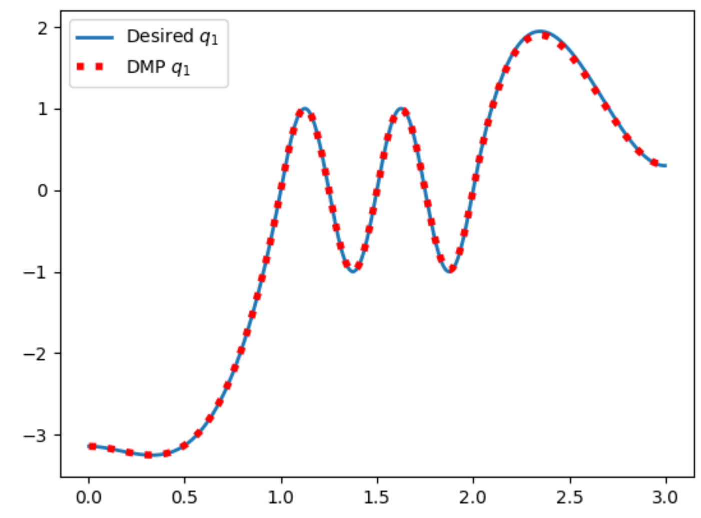
Robot Learning Homework 4 Group 200:

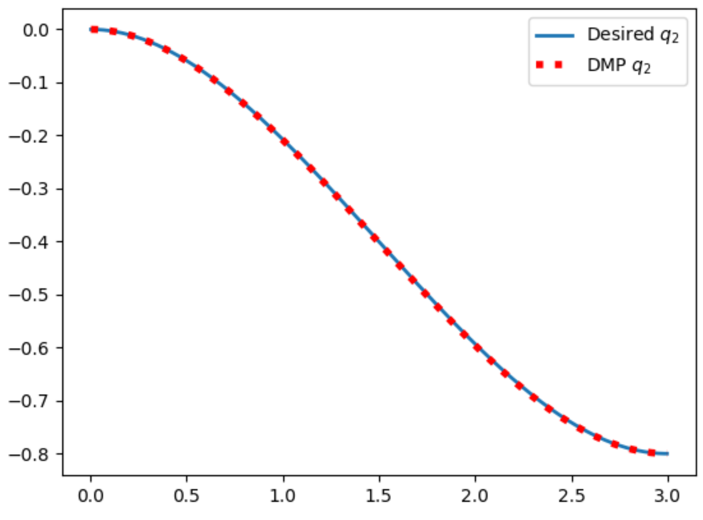
Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

4.1c) Double Pendulum - Training (12 Punkte)

Implement the DMPs and test them on the double pendulum environment. In order to train the DMPs you have to solve Equation (1) on the forcing function. Before starting the execution, set the goal g position to be the same as in the demonstration. Then, set the parameters to $\alpha = 25, \beta = 6.25, \alpha_n = 3/T, \tau = 1$. Use $N = 50$ basis functions, equally distributed in z. Use the learned DMPs to control the robot and plot in the same figure both the demonstrated trajectory and the reproduction from the DMPs. You need to implement the DMP-based controller (dmpCtl.py) and the training function for the controller parameters (dmpTrain.py). To plot your results you can use dmpcomparison. py. Refer to example.py to see how to call it.

Lösungsvorschlag:



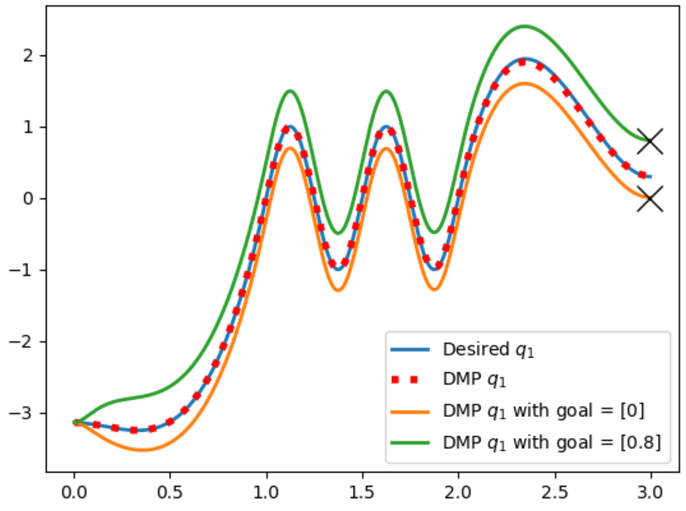


Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

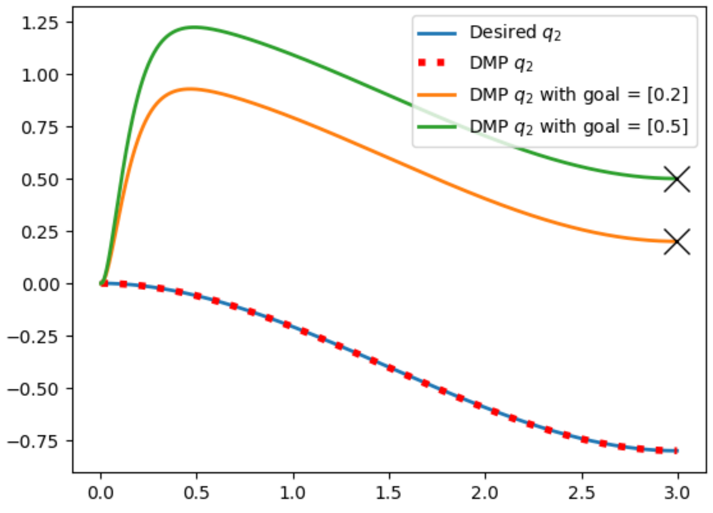
4.1d) Double Pendulum - Conditioning on the Final Position (3 Punkte)

Using the trained DMPs from the previous question, simulate the system with different goal positions: first with $q_{t=\text{ end}} = \{0, 0.2\}$ and then with $q_{t=end} = \{0.8, 0.5\}$. Generate one figure per DoF. In each figure, plot the demonstrated trajectory and the reproduced trajectories with different goal positions. How do you interpret the result?

Lösungsvorschlag:



With adapting the goal attractor g, the final position will also be changed.



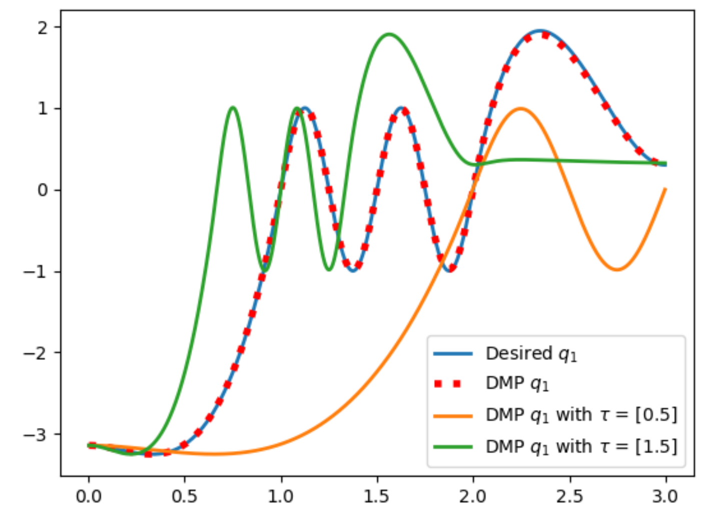
Similar as above interpretation, with different goal the trajectory will be generated also differently, which finally reach the desired goal position.

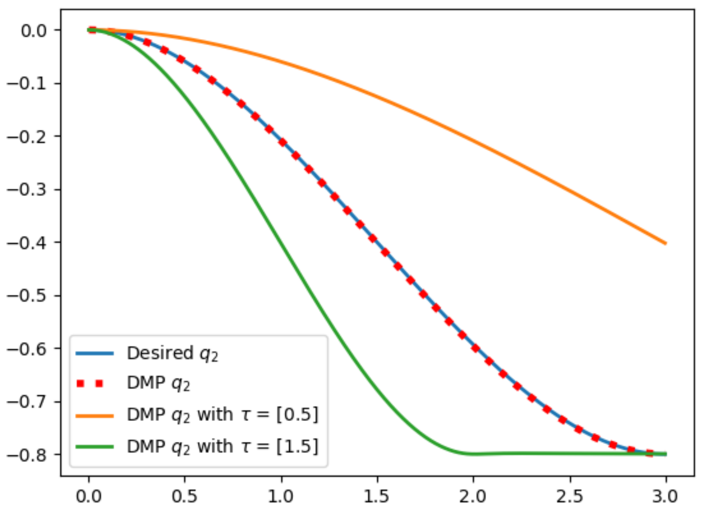
Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

4.1e) Double Pendulum - Temporal Modulation (3 Punkte)

Using the trained DMPs from the previous question, simulate the system with different Temporal Scaling $\tau = \{0.5, 1.5\}$. Generate one figure per DoF and explain the result.

Lösungsvorschlag:





Temporal scaling τ is relevant to movement duration and movement speed.
A higher τ means a higher movement speed and a shorter movement duration. With a higher τ the generated trajectory can reach the goal position earlier. (such as green line vs. yellow line above)

Robot Learning Homework 4

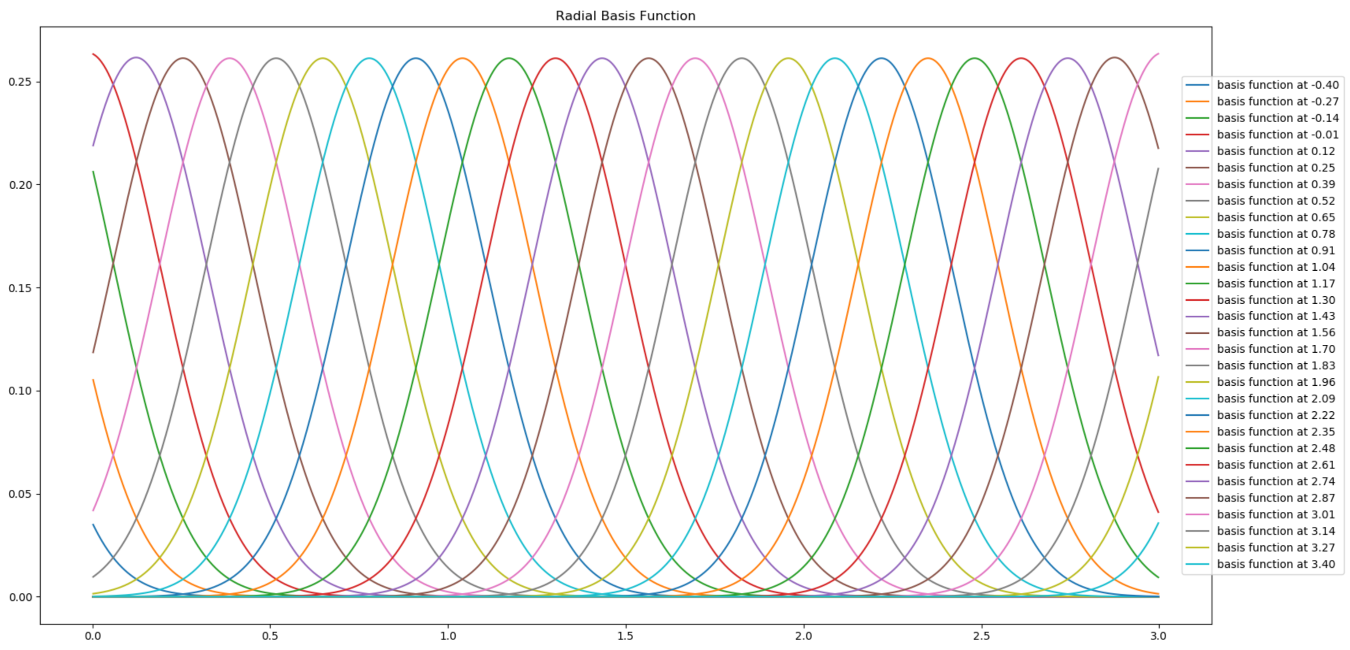
Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

4.1f) Probabilistic Movement Primitives - Radial Basis Function (3 Punkte)

We now want to use ProMPs. Before we train them, we need to define some basis functions. We decide to use $N = 30$ radial basis functions (RBFs) with centers uniformly distributed in the time interval $[0 - 2b, T + 2b]$ where T is the end time of the demonstrations. The bandwidth of the Gaussian basis (std) is set to $b = 0.2$. Implement these basis functions in getProMPBasis.py. Do not forget to normalize the basis such at every time-point they sum-up to onel Attach a plot showing the basis functions in time.

Lösungsvorschlag:



Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

4.1g) Probabilistic Movement Primitives - Training (7 Punkte)

In this exercise you will train the ProMPs using the imitation learning data from getImitationData. py and the RBFs defined in the previous question. Modify the proMP. py in order to estimate weight vectors w_i reproducing the different demonstrations. Then, fit a Gaussian using all the weight vectors. Generate a plot showing the desired trajectory distribution in time (mean and std) as well as the trajectories used for imitation.

Lösungsvorschlag:

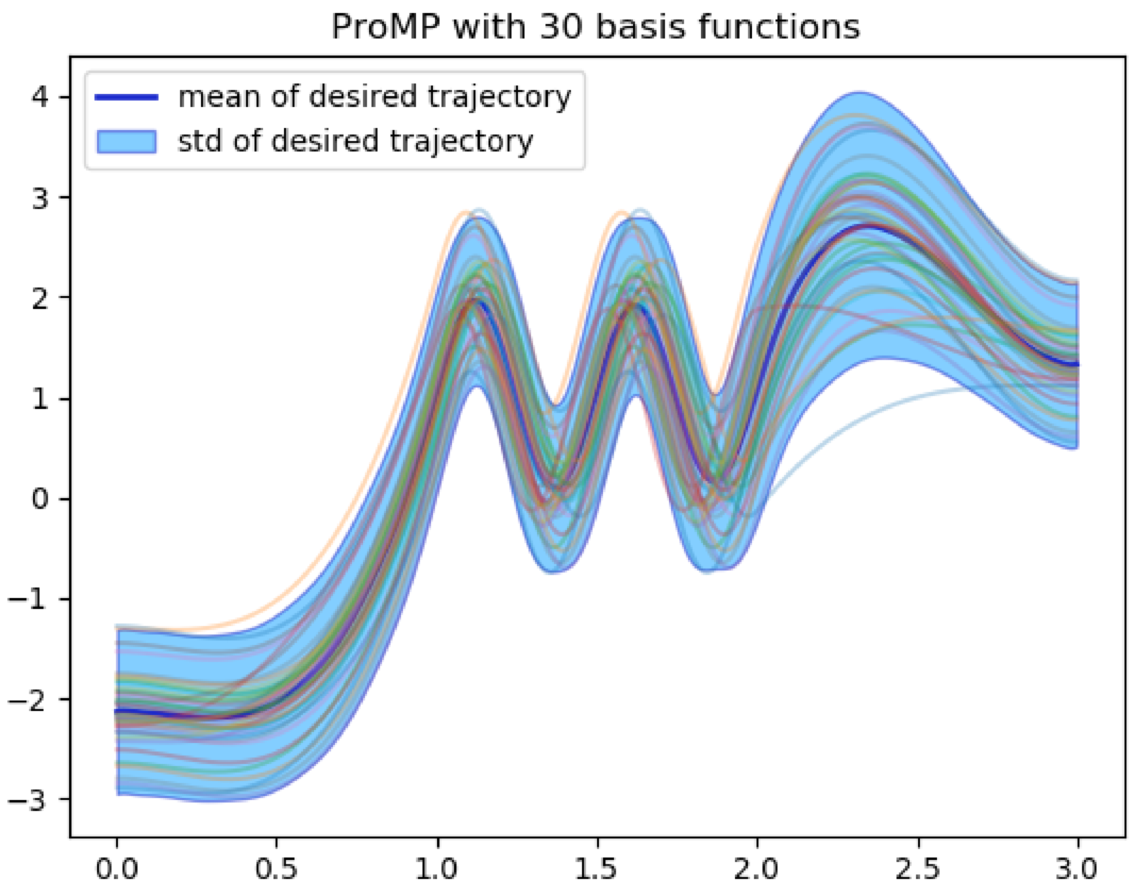
For a single time step:

$$p(\boldsymbol{y} \mid \boldsymbol{\theta}) = \mathcal{N}\left(\boldsymbol{y}_t \mid \boldsymbol{\Phi}_t \boldsymbol{\mu}_{\boldsymbol{w}}, \sigma^2 \boldsymbol{I} + \boldsymbol{\Phi}_t \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{\Phi}_t^\top\right) \tag{9}$$

Compute shape parameters $\boldsymbol{w_i}$ by linear (ridge) regression in each trajectory $\boldsymbol{y_i}$

$$\boldsymbol{w_i} = \left(\Psi^\top \Psi + \sigma^2 I\right)^{-1} \Psi^\top \boldsymbol{y_i} \tag{10}$$

Base on weight vector collection in all trajectory $\boldsymbol{w}$, the mean of weight vector $\boldsymbol{\mu_w}$ and the relevant covariance $\boldsymbol{\Sigma_w}$ could be computed.

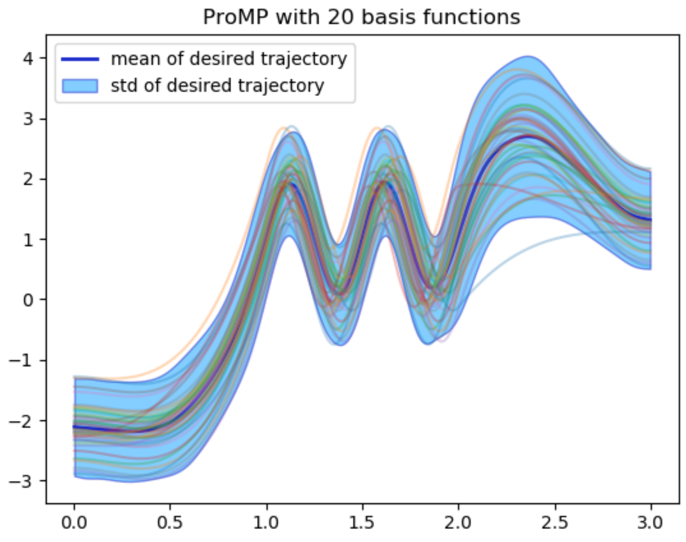


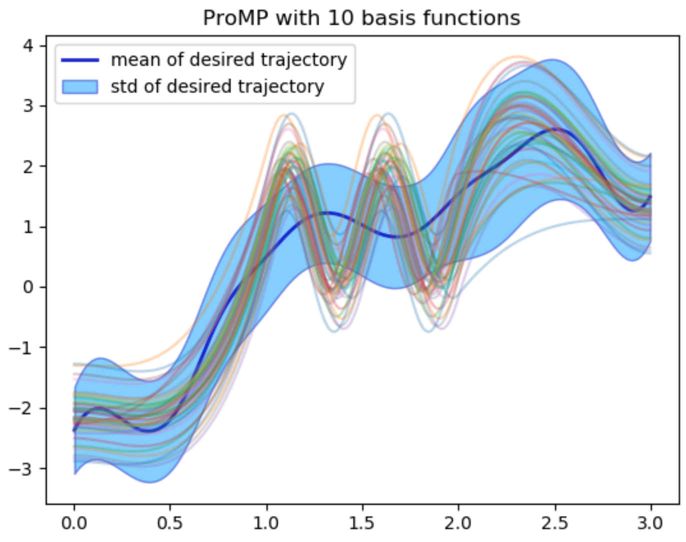
Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

4.1h) Probabilistic Movement Primitives - Number of Basis Functions (2 Punkte)

Evaluate the effects of using a reduced number of RBFs. Generate two plots showing the desired trajectory distribution and the trajectories used for imitation as in the previous exercise, but this time use $N = 20$ and $N = 10$ basis functions. Briefly analyze your results.

Lösungsvorschlag:





The number of RBFs influences the accuracy of movement process.
With less number of RBFs (such as second figure), the generated Radial Basis Function could not represent the enough trajectory details, which leads to the ”too flat”trajectory mean curve with more std.

Robot Learning Homework 4	Group 200:	Vorname	Name	Matrikel-Nr.
		Yi	Cui	2758172
		Yuting	Li	2547040
		Liaotian	Zhihao	2897965

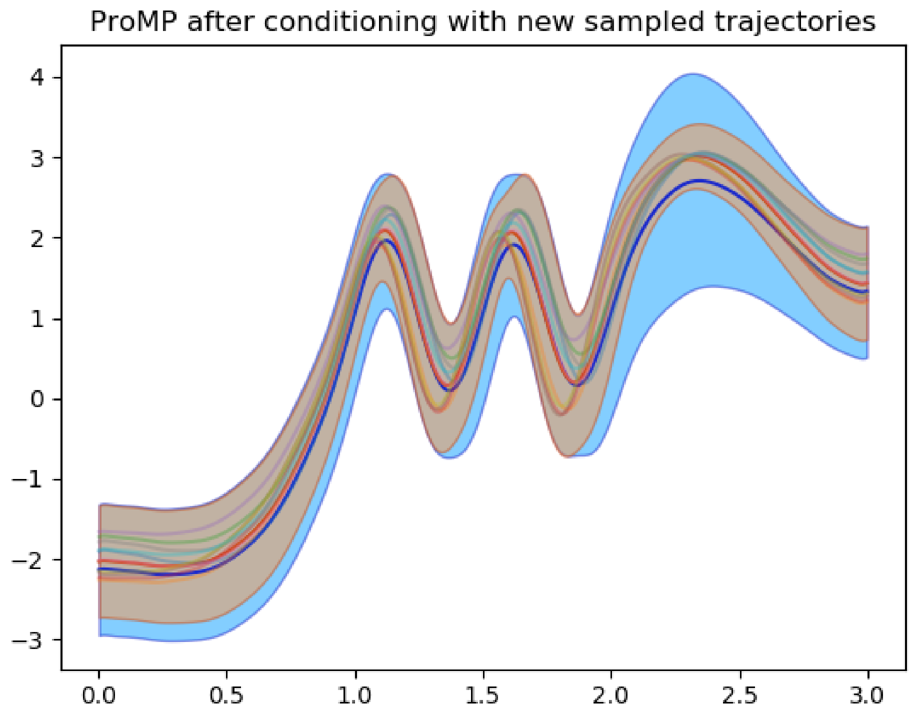
4.1i) Probabilistic Movement Primitives - Conditioning (4 Punkte)

Using Gaussian conditioning calculate the new distribution over the weight vectors w_i such as the trajectory has a via point at position $y^* = 3$ at time $t_{\text{cond}} = 1150$ with variance $\Sigma_{y^*} = 0.0002$. Use again 30 basis functions. Assuming that the probability over the weights is given by $\mathcal{N}\left(w \mid \mu_w, \Sigma_w\right)$ and the probability of being to that position is given by $\mathcal{N}\left(y^* \mid \Phi w, \Sigma_{y^*}\right)$, show how the new distribution over w is computed (how does the mean and variance look like)? Then, in a single plot, show the previous distribution (learned from imitation) and the new distribution (after conditioning). Additionally, sample $K = 10$ random weight vectors from the ProMP, compute the trajectories and plot them in the same plot. Analyze briefly your results.

Lösungsvorschlag:

The new distribution over w is updated as: [1]

$$
\begin{aligned}
\boldsymbol{\mu}_w^{|\text{new}]} &= \boldsymbol{\mu}_w + \boldsymbol{\Sigma}_w \boldsymbol{\Psi}_t \left(\boldsymbol{\Sigma}_y^* + \boldsymbol{\Psi}_t^T \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{\Psi}_t\right)^{-1} \left(\boldsymbol{y}_t^* - \boldsymbol{\Psi}_+^T \boldsymbol{\mu}_w\right) \\
&= \boldsymbol{\mu}_w^{|\text{old}]} + \boldsymbol{\Sigma}_w^{|\text{old}]} \boldsymbol{\Psi}_{t=1150} \left(0.0002 + \boldsymbol{\Psi}_{t=1150}^T \boldsymbol{\Sigma}_{\boldsymbol{w}}^{|\text{old}]} \boldsymbol{\Psi}_{t=1150}\right)^{-1} \left(3 - \boldsymbol{\Psi}_{t=1150}^T \boldsymbol{\mu}_w^{|\text{old}]}\right) \\
\boldsymbol{\Sigma}_{\boldsymbol{w}}^{|\text{new}]} &= \boldsymbol{\Sigma}_{\boldsymbol{w}} - \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{\Psi}_t \left(\boldsymbol{\Sigma}_y^* + \boldsymbol{\Psi}_t^T \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{\Psi}_{\boldsymbol{t}}\right)^{-1} \boldsymbol{\Psi}_t^T \boldsymbol{\Sigma}_{\boldsymbol{w}} \\
&= \boldsymbol{\Sigma}_{\boldsymbol{w}}^{|\text{old}]} - \boldsymbol{\Sigma}_{\boldsymbol{w}}^{|\text{old}]} \boldsymbol{\Psi}_{t=1150} \left(0.0002 + \boldsymbol{\Psi}_{t=1150}^T \boldsymbol{\Sigma}_{\boldsymbol{w}}^{|\text{old}]} \boldsymbol{\Psi}_{\boldsymbol{t=1150}}\right)^{-1} \boldsymbol{\Psi}_{t=1150}^T \boldsymbol{\Sigma}_{\boldsymbol{w}}
\end{aligned}
\tag{11}
$$



The ProMP after conditioning stays within the original distribution.[1]
the modulation is also learned from the original demonstration[1], which is represented by less std in above figure.

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

Literatur

[1] Alexandros Paraschos, Christian Daniel, Jan Peters, and Gerhard Neumann, Probabilistic Movement Primitives