

# Gaussian Processes for Dynamic Movement Primitives with Application in Knowledge-based Cooperation

Yunis Fanger, Jonas Umlauf<sup>1</sup>, Sandra Hirche<sup>1</sup>

**Abstract**—Dynamic Movement Primitives (DMPs) represent stable goal-directed or periodic movements, which are learned from observations or demonstrations. They rely on proper function approximators, which are sufficiently flexible to represent arbitrary movements but also ensure goal convergence in point-to-point motions. This work shows that Gaussian Processes (GPs) are suitable as a regressor for learning movements with DMPs ensuring stability. In addition, GPs provide a measure for the uncertainty about the current movement, which we exploit by proposing a new cooperation scheme for DMPs: For better reproduction of demonstrations, we follow the intuition, that individuals with more knowledge lead towards the goal, while others follow and focus on cooperation. Along with simulation results, we validate the presented methods in a robotic cooperative object manipulation task.

## I. INTRODUCTION

Transferring skills from humans to robots is contemporary in many aspects of robotic research. Seeing that humans are able to perform multi agent tasks such as cooperative object manipulations reliably and with ease, imitating their trajectory generation capabilities and team behaviour is of great value. Thus a framework, which allows robots to extract and learn individual movements from observations and interact human-like in a team, is desirable. In setups, where individuals make separate decisions but have a common (team) goal, the coordination among the members is essential. In the example of joint object manipulation, maintaining a given formation is crucial as otherwise the object is exposed to high forces or is dropped.

In human cooperation individuals with more knowledge about the task take over leading roles, while individuals with less knowledge tend to follow the leader and act more supportively [1]. In situations with asymmetric information, this leads to a knowledge-based role assignment (leader or follower). This is advantageous as only a subset of the team members has to be instructed which then lead the less knowledgeable members, while receiving their support in the task execution [2].

Another area of ongoing research in robotics is generating flexible goal directed movements with guaranteed convergence. More recently, learning dynamical systems from demonstration has gained popularity, but comes with difficulties regarding the guaranteed stability of the reproduction. In [3] a learning approach is presented, which constrains the leaning to stable motions employing Gaussian

Mixture Models and dynamical systems in combination with Lyapunov theory. As an alternative, Dynamic Movement Primitives (DMPs), initially proposed in [4], provide a very flexible method for representing goal directed or periodic movements with dynamical systems. Trajectories are learned from demonstration and are generalized during execution. Their application ranges from sculpting [5] to tennis racket swinging [6]. DMPs have extensions for multiagent interaction, where cooperation is based on virtual or measured forces at the end-effectors [7] [8], or relies in potential field for reaching consensus [9]. In all these approaches the possibility of asymmetric distributed training data is not considered, which makes the cooperating agents always equal partners. Modelling the uncertainty of training data for DMPs is considered in [10], where the concern lies with reproducing the uncertainty. In contrast, this work aims to utilize uncertainty for parallel tasks, e.g. multi agent cooperation.

The main contribution of this paper is a new approach towards learning guaranteed converging point-to-point movements from demonstrations, while exploiting uncertainty information for the reproduction. We propose a combination of Dynamic Movement Primitives with Gaussian Processes (GPs). In order to guarantee goal convergence, i.e. stability of the underlying dynamics, a transformation of the GP's input space is introduced. Furthermore, this GP-based DMP approach is employed in a newly proposed scheme for learning cooperative tasks including role assignment from demonstrations. Therefore the uncertainty information encoded in the GP in terms of the predicted variance is exploited: following the human example the robots adapt their roles and cooperation behaviour depending on their individual knowledge. The proposed approach is validated in simulations and experiments for cooperative manipulation.

The remainder of this paper is structured as follows: Section II reviews fundamentals, namely cooperative DMPs (II-A) and Gaussian Process Regression (II-B). Section III introduces GPs for DMPs including the proof for stability, followed by the concept of dynamic cooperation in IV. Simulative and experimental results are shown in section V.

## II. BACKGROUND

### A. Cooperative DMPs

DMPs generate flexible trajectories based on a stable dynamical system. Either goal directed or periodic movements are learned from demonstration and encoded in the DMP. The work in [9] focuses on trajectories for cooperative

<sup>1</sup> Member of the Institute for Information-Oriented Control, Department of Electrical Engineering and Information Technology, Technical University of Munich, D-80290 Munich, Germany [yunis.fanger, jonas.umlauft, hirche]@tum.de

object manipulation using DMPs by having multiple agents maintaining a formation or moving synchronously.

For goal directed movements, DMPs are based on a *point attractive system*<sup>1</sup>

$$\tau \dot{z} = \alpha(\beta(\mathbf{g} - \mathbf{x}) - \mathbf{z}) + \mathbf{f}(s), \quad \tau \dot{\mathbf{x}} = \mathbf{z}, \quad (1)$$

where  $\mathbf{x}, \mathbf{z}, \mathbf{g} \in \mathbb{R}^n$  are position, velocity and goal points, respectively and  $\alpha, \beta, \tau > 0$  regulate the goal convergence of the linear part. The nonlinearity  $\mathbf{f}(s) : \mathbb{R}_+ \rightarrow \mathbb{R}^n$  adds the required flexibility for nonlinear behaviour to the system and depends on the scalar phase variable  $s$ , which is generated by the *canonical system*

$$\tau \dot{s} = -\gamma s, \quad \gamma > 0, \quad s(0) = s_0 > 0. \quad (2)$$

The canonical system removes explicit time dependency of  $\mathbf{f}$  and therefore allows to react online, e.g. to perturbation, by slowing down the movement.

To guarantee stability,  $\mathbf{f}(s) \rightarrow 0$  must hold for  $s \rightarrow 0$ . Thus, for  $t \rightarrow \infty$ , the point attractive system (1) is reduced to its linear part, which converges to  $[\mathbf{z}^* \ \mathbf{x}^*] = [\mathbf{0} \ \mathbf{g}]$ . DMPs are also capable of representing periodic movement if  $\mathbf{f}(s)$  is chosen to be a periodic function. GP are applicable in this case as well by using a periodic kernel function.

The nonlinearity  $\mathbf{f}(s)$ , which acts as a external force on the linear spring-damper system, allows DMPs to generate any smooth trajectory. It is obtained from training/observation data and usually modelled using Locally Weighted Projection Regression (LWPR) [11]. For each observation of the demonstrated movement(s)  $\mathbf{x}^{(k)}, \dot{\mathbf{x}}^{(k)}, \ddot{\mathbf{x}}^{(k)}$  for  $k = 1 \dots m$  the output training data is computed by rearranging (1) to

$$\mathbf{f}^{(k)} = \tau^2 \ddot{\mathbf{x}}^{(k)} - \alpha(\beta(\mathbf{g} - \mathbf{x}^{(k)}) - \tau \dot{\mathbf{x}}^{(k)}). \quad (3)$$

The input training data is obtained from simulating (2) and taking samples at a constant rate, resulting in  $s^{(k)}$ , concatenated in  $X \in \mathbb{R}_+^m$ . For this input-output mapping, in our approach, as described in the section III, LWPR is replaced by Gaussian Process Regression (GPR).

Cooperative DMPs, introduced in [9], extend DMPs to a network of  $i = 1 \dots N$  agents

$$\tau \dot{z}_i = \alpha_i(\beta_i(\mathbf{g}_i - \mathbf{x}_i) - \mathbf{z}_i) + \mathbf{f}_i(s_i), \quad (4)$$

$$\tau \dot{\mathbf{x}}_i = \mathbf{z}_i + \kappa_i \mathbf{c}_i(\mathbf{x}_{N_i}), \quad (5)$$

$$\tau \dot{s}_i = \frac{\gamma_i s_i}{1 + \|\eta_i \mathbf{c}_i(\mathbf{x}_{N_i})\|^2}, \quad (6)$$

where  $\mathbf{x}_{N_i}$  contains the state of all agents in set  $N_i$ , which is the set of neighbours of agent  $i$ , meaning they are capable of communicating their current state to agent  $i$ . The scalar constants  $\eta_i, \kappa_i > 0$  determine the influence of the cooperation term  $\mathbf{c}_i : \mathbb{R}^{n|N_i|} \rightarrow \mathbb{R}^n$  defined as

$$\mathbf{c}_i(\mathbf{x}_{N_i}) = - \sum_{j \in N_i} \delta_{ij} \frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|} (\|\mathbf{x}_i - \mathbf{x}_j\| - d_{ij}), \quad (7)$$

where the scalar  $d_{ij} > 0$  denotes the Euclidean desired distance between agent  $i$  and  $j$  and  $\delta_{ij} > 0$  allows to weight some links more than others. The distributed formation

<sup>1</sup>Notation: Bold symbols denote vectors. Capital letters denote matrices.  $\|\mathbf{a}\|$  denotes euclidean norm of vector  $\mathbf{a}$ .  $\|N\|$  denotes the cardinality of the set  $N$ .  $I_p$  denotes the  $p \times p$  identity matrix.  $A \succ 0$  denotes positive definiteness of the matrix  $A$ . Time dependencies of  $\mathbf{x}(t), \mathbf{z}(t), s(t)$  have been removed for simplification. In  $x$  denotes the natural logarithm of  $x$ .

control law originates from a potential field approach [12] and relies on reaching consensus on formation in distributed networks [13].

The results in [9] show that the cooperation successfully maintains a formation during execution. In the application to cooperatively manipulating an object, this leads to significant reduction of the internal stress on the manipulated object. So far only equal cooperation among all agents is assumed and therefore cases, where individual agents obtain asymmetric data about their individual trajectory are not considered, e.g.:

- For a human demonstrator, it is more convenient to only train a subset of agents. E.g. for a N dimensional movement training N manipulators (assuming enough links) is sufficient for successful reproduction of the object manipulation.
- For certain parts of the task some manipulators could be subject to tight constraints, e.g. navigating through a narrow gap, while others are not. So demonstrating critical passages to individual manipulators increases the safety during execution.
- Sensor limitation may cause missing data for individual manipulators, e.g. due to occlusion of an optical tracking system. Our cooperation scheme is able to compensate that by providing guidance to the agents with missing data.

## B. Gaussian Process Regression

Numerous methods exist for solving regression tasks but more recently non-parametric methods are gaining in popularity as their explanatory power and flexibility grows with the amount of data available [14].

The task is to model the latent function  $g(\boldsymbol{\xi})$  of which noisy measurements  $\mathcal{D} = \{\boldsymbol{\xi}_k, y_k\}_{k=1}^m$  originate from  $y_k = g(\boldsymbol{\xi}_k) + \epsilon_k$  with  $\epsilon_k \sim \mathcal{N}(0, \sigma_n^2)$ . The inputs  $\boldsymbol{\xi}_k \in \mathcal{X}$  are concatenated in  $\Xi$ , while outputs  $y_k$  are, for now, assumed to be scalar and concatenated in  $\mathbf{y}$ . A Gaussian Process is fully described by its mean  $m_g(\boldsymbol{\xi})$  and covariance function  $k_{\text{var}}(\boldsymbol{\xi}, \boldsymbol{\xi}')$ . The mean function is commonly, due to lack of prior knowledge, set to zero. A popular choice for the covariance function is the squared exponential (SE) kernel

$$k_{\text{SE}}(\boldsymbol{\xi}, \boldsymbol{\xi}') = \sigma_g^2 \exp\left(-\frac{1}{2}(\boldsymbol{\xi} - \boldsymbol{\xi}')^T \Lambda^{-2}(\boldsymbol{\xi} - \boldsymbol{\xi}')\right) \quad (8)$$

with the hyperparameters signal variance  $\sigma_g > 0$  and length-scales  $\Lambda = \text{diag}(\boldsymbol{l}) \succ 0$ , as it allows to model any smooth function. The hyperparameters are obtained from a (log) likelihood optimization, thus maximizing the probability of the seen data to occur given the current parameters [14] and input values

$$[\Lambda^*, \sigma_g^*] = \arg \max_{\Lambda, \sigma_g} \log p(\mathbf{y} | \Xi, \Lambda, \sigma_g).$$

This optimization also reliably avoids overfitting. The regression at a single test input  $\boldsymbol{\xi}_*$  returns a mean and variance prediction with  $\mathbf{k}_* = k_{\text{var}}(\Xi, \boldsymbol{\xi}_*) \in \mathbb{R}^m$  and  $K = k_{\text{var}}(\Xi, \Xi) \in \mathbb{R}^{m \times m}$

$$\mathbb{E}[g(\boldsymbol{\xi}_*)] = m_g(\boldsymbol{\xi}_*) + \mathbf{k}_*^T (K + \sigma_n^2 I_m)^{-1} \mathbf{y}, \quad (9)$$

$$\mathbb{V}[g(\boldsymbol{\xi}_*)] = k_{\text{var}}(\boldsymbol{\xi}_*, \boldsymbol{\xi}_*) - \mathbf{k}_*^T (K + \sigma_n^2 I_m)^{-1} \mathbf{k}_*. \quad (10)$$

For multiple output dimensions, multiple GPs are applied independently. For a more detailed description see [14].

### III. GAUSSIAN PROCESSES FOR DMPs

As mentioned above, previous work on DMPs apply LWPR to fit the steering term  $\mathbf{f}(s)$  in (1) to the demonstration data. It is based on dimensionality reduction and fitting a finite number of linear models to the data [11]. It comes with the advantage of low computational complexity in the number of training data  $\mathcal{O}(m)$  and is therefore applicable in real-time learning applications. But LWPR requires manually adjusted meta-parameters and many local models as the linearity assumption only holds in small regions [15]. The work in [16] formulates the problem of learning DMPs with Gaussian Mixture Regression (GMR), a probabilistic approach, but here no formal convergence guaranties are shown and the variance is exploited for stiffness modulation. GMR is also a parametric model as the number of components is set a priori.

In comparison, the hyperparameters for GPR are automatically adjusted based on marginal likelihood optimization. Its major limitation is the high computational complexity of  $\mathcal{O}(m^3)$  for training, but many recent developments have successfully tackled this problem using approximations, e.g. [15]. It also comes with the major advantage of providing an uncertainty measure with each prediction, allowing for a judgement on the model fidelity. We exploit the variance for dynamic cooperation as described in section IV.

So instead of using LWPR to fit the nonlinearity  $\mathbf{f}(s)$  to the training data, we propose to use GPR, more precisely its mean prediction (9) as a steering function for DMPs.

*a) Prior Knowledge:* We assume that  $\mathbf{f}(s)$  is a continuous differentiable function, which justifies the use of a squared exponential covariance function. Under the assumption, that no prior knowledge about the nonlinearity function is available, we employ a zero prior mean function  $m_{\mathbf{f}} = 0$ . We will later see, that any other kernel, which approaches zero for infinitely large distance between test and training points, and any prior mean converging to zero at the input origin are applicable.

*b) Logarithmic Transformation:* Taking samples from the simulation of the canonical system (2) with constant sampling time results in the input training data points becoming denser and denser close to the origin due to its exponential decay. Directly applying these training data points to the GP results in poor regression performance and additionally comes with issues regarding the systems stability guarantee as described in paragraph III-0.c. Therefore, we propose a logarithmic transformation of the input space

$$\tilde{s}^{(k)} = \ln s^{(k)} \quad (11)$$

to obtain equally spaced input training data. The input space of  $\mathbf{f}$  therefore now ranges from  $-\infty$  to  $\ln s_0$ . This allows better fitting of hyperparameters of the covariance function, as illustrated in Fig. 1. It is also assumed, that each demonstration has finite duration, thus  $s^{(k)}$  is lower bounded by a strictly positive number  $\min_k s^{(k)} > 0$  and therefore  $\min_k \tilde{s}^{(k)} > -\infty$ , i.e.,  $\tilde{s}^{(k)}$  has a finite lower bound.

*c) Stability:* One of the significant advantages of DMPs as given through equation (1) and (2) is the guaranteed goal convergence. As previously mentioned, this is achieved if the steering term vanishes over time, which is the case for the following reasons:

According to both (2) and (6) (for bounded cooperation or perturbation terms)  $s$  is, in the reproduction, guaranteed to converge to zero over time. Using the logarithmic transformation,  $\tilde{s}$  approaches  $-\infty$  over time. Thus, the distance between training samples  $\tilde{s}^{(k)}$  and phase variable during execution  $\tilde{s}$  approaches infinity for large times, due to the fact that demonstration movements have finite duration

$$\lim_{t \rightarrow \infty} \|\tilde{s} - \tilde{s}^{(k)}\| = \infty.$$

Therefore, the squared exponential kernel returns decreasing correlation between those points, expressed in  $\mathbf{k}_*$ , which is, assuming finite lengthscales, in the limit zero:

$$\lim_{\|\tilde{s} - \tilde{s}^{(k)}\| \rightarrow \infty} k_{\text{SE}}(\tilde{s}, \tilde{s}^{(k)}) = 0, \quad \forall k$$

Thus, the second term in (9) vanishes for  $t \rightarrow \infty$  under the mentioned conditions. The first part of (9),  $m_g$ , also becomes zero for  $s \rightarrow 0$  or  $\tilde{s} \rightarrow -\infty$  due to the lack of prior knowledge, as mentioned above. With a vanishing steering function, the point attractive system is linear and stable system according to the positive choice of  $\alpha, \beta, \tau$ .

Thus, by using the mean of GPR for steering DMPs, stability is preserved naturally under very mild conditions regarding the prior mean and covariance function. This is summarized in the following proposition

*Proposition 1 (Stability of DMP with GPR):* The point attractive system (1), with  $\mathbf{f}(s)$  being the GP mean prediction (9) and given

- the logarithmic transformation of the phase variable according to (11) and of the corresponding training data  $\tilde{s}^{(k)} = \ln s^{(k)}$ ,  $\forall k$  as input to the GP,
- a prior mean function  $m_{\mathbf{f}}(\tilde{s})$  which is zero in the limit  $\tilde{s} \rightarrow -\infty$ ,
- any kernel  $k_{\text{var}}$  for which holds

$$\lim_{\|\tilde{s} - \tilde{s}'\| \rightarrow \infty} k_{\text{var}}(\tilde{s}, \tilde{s}') = 0,$$

- finite demonstration time, thus  $\tilde{s}^{(k)} > -\infty$ ,  $\forall k$ ,
- strictly positive bounded parameters  $\alpha, \beta, \gamma, \tau$ ,

converges to the equilibrium point  $[\mathbf{z}^* \ \mathbf{x}^*] = [\mathbf{0} \ \mathbf{g}]$ .

Thus, by using the logarithmic transformation, the SE kernel and a zero mean prior, DMPs based on GPR are asymptotically stable.

#### A. Simulation GPR for DMPs

To validate the usage of GPR in DMPs and illustrate the stability proposition, we simulate the reproduction of an analytically generated trajectory for a goal directed single agent movement (Sim 1). Fig. 1 depicts the GPs predicted mean and variance next to the LWPRs regression result. The training data and reproduced movement with LWPR and GP, which uses the logarithmic transformation are shown in Fig. 2.

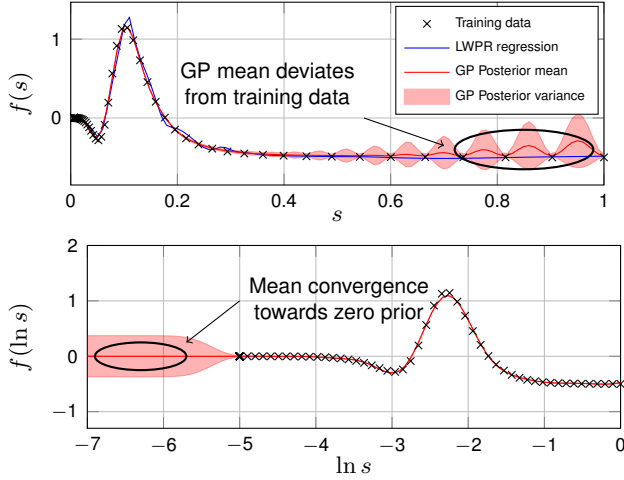


Fig. 1: Sim 1: LWPR and GP Posterior mean, before and after logarithmic transformation for nonlinear function  $f(s)$

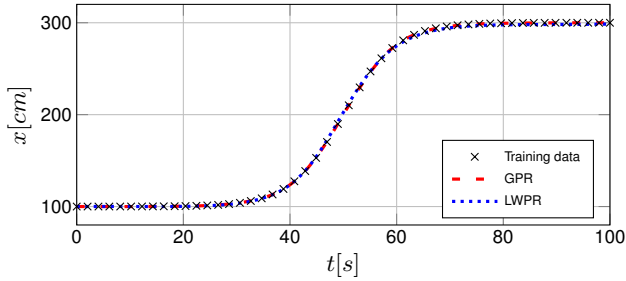


Fig. 2: Sim 1: Trajectory reproduction using GPR and LWPR

As described in section III, the logarithmic transformation spreads the training data evenly throughout the input space of  $s$ . Therefore, the GP with logarithmic transformation is able to fit the given data better than the non logarithmic variation. The hyperparameters optimization for unevenly distributed training data does fail in either the region with dense data or widely spread data, which is the case in Fig. 1. The resulting reproduced trajectories with GP and LWPR in Fig. 2 are very similar. As demonstrated both reproductions converge towards the goal point, supporting the theoretical justifications for stability made in section III.

*Remark:* The GP's variance in DMPs can be used in many different ways. Here we focus on the application to multi agent systems, e.g. variance-based impedance control [17].

#### IV. APPLICATION TO KNOWLEDGE-BASED COOPERATION

After utilising the mean prediction (9) in the previous section, we now show how the variance in (10) is exploited for dynamic cooperation. The measure of uncertainty provided by the GP reflects the proximity to training data available at the current point of the task: Variance is low if many training data points exist near the current point and high if moving in regions with sparse training data. Again, the core idea is that agents with more training data have more knowledge and therefore are more dedicated to follow their individual trajectory, while agents with less training data focus on maintaining the formation following the more knowledgeable

individuals. The amount of training data available to each agent, and with that the variance, may change throughout the task (e.g. due to occlusion of sensors in certain regions or lack of demonstrations).

With the predicted variance, GPR delivers a measure for the model fidelity and thus for availability of training data in each phase of the desired movement. Based on this measure, we vary the influence of the cooperation term on each agent's movement. More formally, rather than using constant  $\kappa_i$  in (5) and  $\eta_i$  in (6) we propose to employ variance dependent functions

$$\kappa_i(\sigma_{N_i}) : [\sigma_{n,i} \quad \sigma_{f,i}]^{|N_i|} \rightarrow [0 \quad \kappa_{\max}], \quad (12)$$

$$\eta_i(\sigma_{N_i}) : [\sigma_{n,i} \quad \sigma_{f,i}]^{|N_i|} \rightarrow [0 \quad \eta_{\max}], \quad (13)$$

where  $\kappa_{\max} > 0, \eta_{\max} < \infty$  and  $\sigma_{N_i}$  concatenates the variances of neighbouring agents  $\sigma_i, \forall i \in N_i$  and  $\sigma_i$  is obtained during reproduction from (10). The terms  $\sigma_{f,i}, \sigma_{n,i}$  denote the GP hyperparameters signal variance and signal noise of agent  $i$ , respectively. Thus, along with the state  $x_i$ , the uncertainty of agent  $i$ ,  $\sigma_i$  (as computed from GPR eq. (10)) is the only communicated quantity.

The functions (12), (13) determine the trade-off between pursuing the individual or the team goal and basically encode the team behaviour. Many choices for the functions  $\kappa_i(\sigma_{N_i}), \eta_i(\sigma_{N_i})$  can be justified, here only a few options and boundary conditions are discussed. For simplicity, we limit the discussion to  $\kappa_i$ .

- From intuition: agents increase their dominance in the team with increased knowledge about the task. Thus in our framework  $\kappa_i(\sigma_{N_i})$  must monotonically decrease with  $\sigma_i$ .
- If too many agents claim the leadership and formation is not respected agent  $i = L$  with the highest certainty acts as leader (following the demonstration accurately) while all others cooperate

$$\kappa_L = 0, \quad \kappa_i = \kappa_{\max} \quad \forall i \neq L. \quad (14)$$

Note, that the error in the formation in this way can be made arbitrarily small by increasing  $\kappa_{\max}$  but cannot be completely eliminated.

- The overall level of cooperation is kept constant, e.g.

$$\sum_{i=1}^N \kappa_i = \kappa_{\max} \quad \forall \sigma_{N_i}. \quad (15)$$

Breaking this down for two agents  $i$  and  $j$  and presuming a linear function, we derive the function

$$\kappa_i = \kappa_{\max} \frac{\frac{\sigma_i - \sigma_{n,i}}{\sigma_{i,f} - \sigma_{n,i}}}{\frac{\sigma_i - \sigma_{n,i}}{\sigma_{i,f} - \sigma_{n,i}} + \frac{\sigma_j - \sigma_{n,j}}{\sigma_{j,f} - \sigma_{n,j}}}, \quad (16)$$

which is used in the following experiments. A visualization of the function for  $\kappa_i$  is shown in Fig. 3.

$\alpha$	$\beta$	$\gamma$	$\kappa_{\max}$	$\eta$	$\tau$
0.1	0.025	0.05	$5N$	0.0005	1
	Sim 1	Sim 2	Sim 3	Exp 1	Exp 2
$\Lambda$	optimized	diag(0.5)	diag(0.1)	diag(0.01)	diag(0.01)
$\sigma_f$	optimized	1	1	0.02	0.02
$\sigma_n$	0.01	0.01	0.01	0.0001	0.0001

TABLE I: Parameters used in simulations and experiments

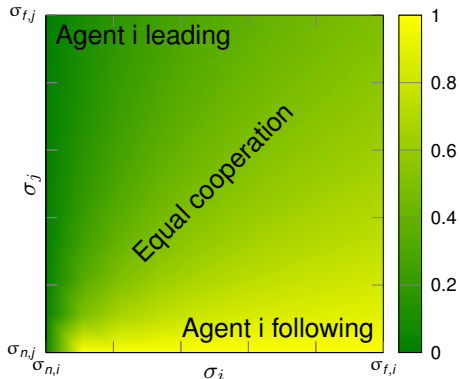


Fig. 3: According to (16), the figure shows  $\kappa_i$  as a function of  $\sigma_i$  and  $\sigma_j$  for  $\kappa_{\max} = 1$

## V. RESULTS

The parameters used in all simulations and experiments are listed in Tab. I. The DMP parameters were chosen to critically damp the linear system. Whenever hyperparameters of the GP are not optimised, it is done to save computational effort. The coupling and time scaling parameters  $\kappa_{\max}$  and  $\eta$  can be modified to affect the trade-off between trajectory tracking and formation maintenance.

### A. Multi Agent Simulations

The following simulation focuses on cooperation among multiple agents and exploits the variance values yielded from the GP.

#### 1) Two Agent Dynamic Cooperation (Sim 2):

a) *Setup and Results:* Both agents are supposed to perform a one dimensional movement with  $x_1 = 200 + 100 \tanh(\frac{t-50}{10})$ ,  $x_2 = 300 + 100 \tanh(\frac{t-50}{10})$ , while maintaining the constant distance of  $d_{1,2} = 100$ . This is an example of a one dimensional translation of an object by two manipulators with smooth acceleration and deceleration. To illustrate the difference between the dynamic and constant cooperation, training data is divided sparsely among the agents. This leads to varying uncertainty for both agents, thus making dynamic cooperation in the sense of dynamic role assignment necessary. We choose the  $\kappa$ -function to be the linear function (16). Fig. 4 shows the simulated reproduced trajectories of both cooperation methods along with the dynamic cooperation term  $\kappa$  ( $\kappa_{\max} = 10$ ).

b) *Discussion:* The dynamic cooperation reproduction follows the training data more accurately than the one simulated with constant  $\kappa$  values. The two agents switch their leader and follower roles sufficiently fast to improve the resulting joint trajectory.

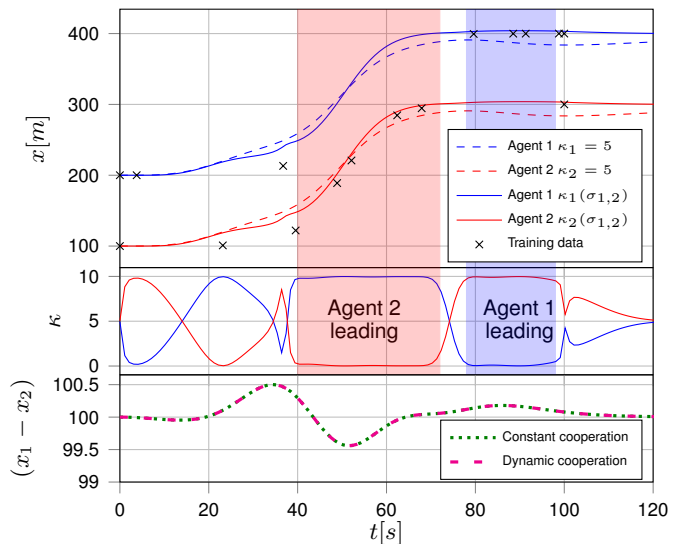


Fig. 4: Sim 2: Reproduced 1D movement for constant cooperation ( $\kappa = 5$ ) and dynamic cooperation ( $\kappa$  in lower plot) with sparse training data for both agents. The maintained distance is identically as it only depends on the sum  $\kappa_1 + \kappa_2$  which is constant, also in the dynamical cooperation case.

#### 2) Three Agent Joint Task (Sim 3):

a) *Setup and Results:* Three agents in the  $xy$ -plane are placed in an equilateral triangle formation and are taught to execute a rotational movement around the triangle's center, while translating it with constant speed in  $x$  direction. Training data is given to the first two agents, whereas the third only received start and goal points. The reproduction of the training data is performed with dynamic as well as constant cooperation. The spacial trajectories are shown in Fig. 5.

b) *Discussion:* The simulation shows, the reproduction with dynamic cooperation traces the training data closely, whereas the agents in the constant cooperation variation deviate from their desired trajectory significantly. The reproduction with constant cooperation also reduces the execution speed and therefore fails to converge to the goal in the demonstration time.

### B. Experiments

The experiments are designed to illustrate the application of our varying cooperation scheme in a real world setting. They are performed on two 7 DoF KUKA LWR robot manipulators as cooperating agents, providing measurement data on end-effector position and forces. The manipulators including an attached object are shown in Fig. 6.

a) *Goal:* The experiments aim to imitate a human motion, while maintaining the desired euclidean distance between the agents. We compare how the usage of dynamic, constant or no cooperation affects the robots performance in both retaining the formation as well as in the accuracy of the reproduced movement.

b) *Setup:* We use kinesthetic teaching by a human operator for the demonstration without the object attached.

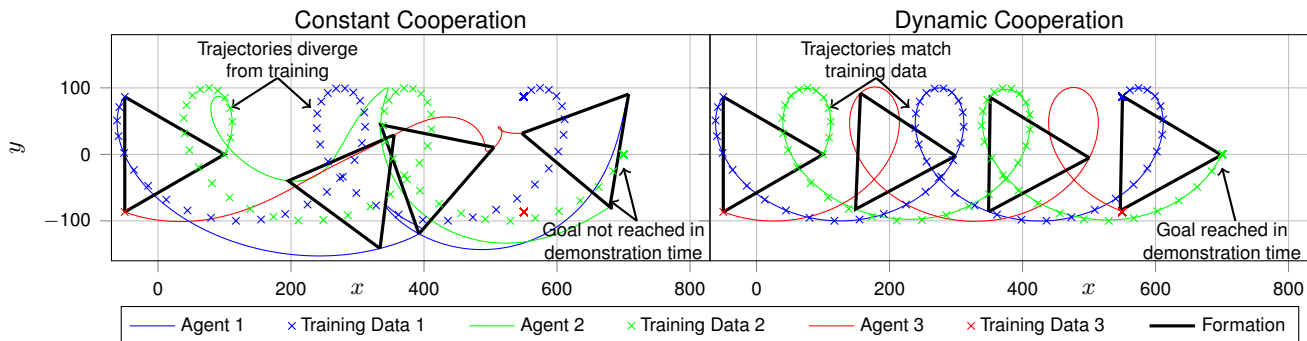


Fig. 5: Movement of three agents with asymmetric knowledge (Sim 3), constant cooperation with  $\kappa_i = 5 \forall i$ .

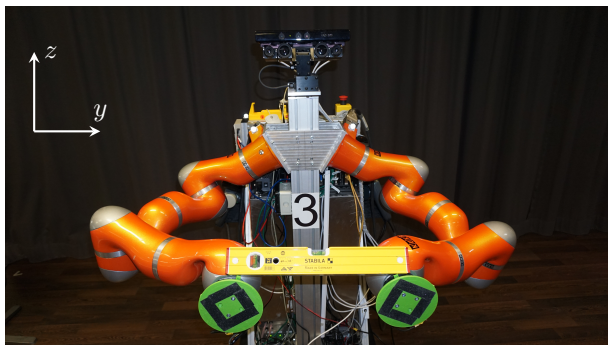


Fig. 6: KUKA manipulators with object

The employed Cartesian impedance controller are set to zero stiffness in the allowed Cartesian directions and constrained movements in all other directions as well as any rotations. After data recording, it is processed offline to keep the realtime constraints of the GPR. This processing includes down sampling, smoothing using a moving average and the numerical computation of the derivatives needed in (3). Also the computationally critical part of GPR (9), (10), the inverse  $(K + \sigma_n^2 I_m)^{-1}$  is computed offline.

Furthermore, data is only added to the training set if at the time of its recording an external force is measured. This makes only the movements, which are indeed induced by the human demonstrator, be learned by the GP and leads to a different data set for each of the agents. By demonstrating one part of the movement on only one manipulator and the second part on the other, it is determined which of them takes the lead at a certain point in time. The object is attached to the end-effectors not during the demonstration but in reproduction. Therefore the robots are required to keep the desired distance of  $d = 0.4m$ .

*c) Outcome:* In the first iteration of the experiment (Exp 1) the demonstrated motion moves only in the  $y$ -direction. Fig. 7 shows the reproductions i) without cooperation, ii) with constant cooperation and iii) with dynamic cooperation. The error in maintaining the initial distance between the manipulators is shown in Fig. 8. During the reproductions ii) and iii) an object is attached to the end-effectors, which verifies, that the formation is maintained throughout the motion, as it was not dropped. The sec-

ond part of this experiment (Exp 2), extends the allowed demonstration domain to the  $zy$ -plane. Here, we focus on the difference in spatial reconstruction of the training trajectory between the two cooperation methods, which is shown in Fig. 9. In this depiction each agent's trajectory does not lie entirely on training points since in the time periods where they don't have training data they follow their partner to keep the distance.

*d) Discussion:* These experiments demonstrate that after preprocessing the data offline, GPR is capable of keeping the real-time constraint for the 1 kHz update rate of the robot. The movements without any cooperation follow the training data almost perfectly, emphasizing that GPR is suitable to replace LWPR. However, the formation constraints are ignored and therefore joint tasks cannot be successfully executed. The comparison of different cooperation styles illustrates higher tracking precision of dynamic cooperation in contrast to constant cooperation. Only through the use of dynamic cooperation with a leader follower concept, both reconstruction accuracy and formation maintenance are satisfied. This becomes especially apparent in the reconstruction of the two dimensional movement seen in Fig. 9, where constant cooperation causes severe deviation from the desired motion. The approach is currently limited to cases where training data is provided at different time intervals to the agents. Multiple training trajectories can be handled, but the GP only considers their mean at each time step.

## VI. CONCLUSION

In this paper, we propose a new approach towards learning guaranteed converging point-to-point movements from demonstrations while exploiting uncertainty information for the reproduction. We employ a combination of Gaussian Processes with Dynamic Movement Primitives. A logarithmic transformation of the GPs input data is introduced in order to guarantee stability. Using GPs, we gain a measure of confidence at each phase of the motion, based on which we implement a dynamic leader-follower scheme as an extension to the cooperative DMP concept. It shows superior behaviour in jointly executed tasks with asymmetric data distributions. Simulations and robot experiments show the validity of this approach and demonstrate how it outperforms static cooperation in terms of tracking precision.



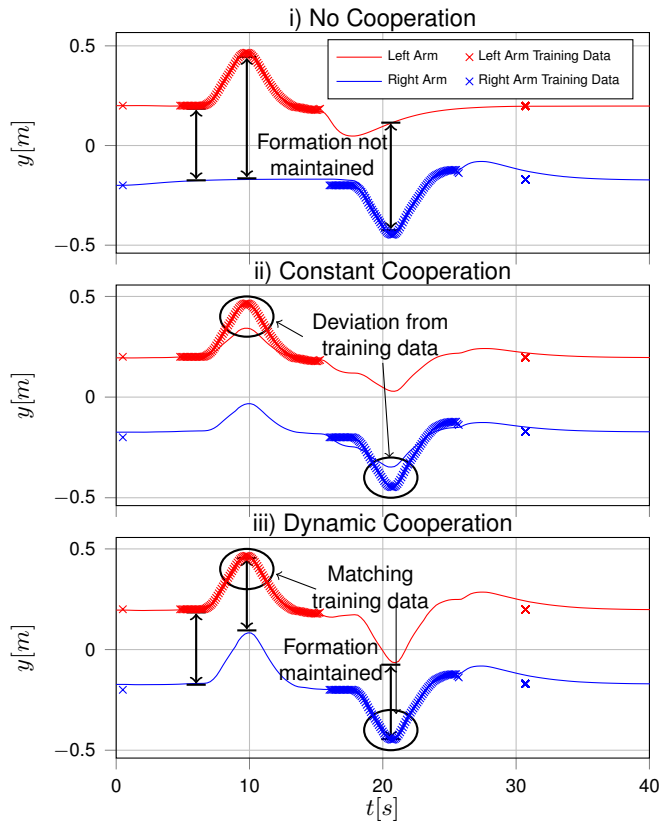


Fig. 7: Exp 1: Reproduction of a 1D movement

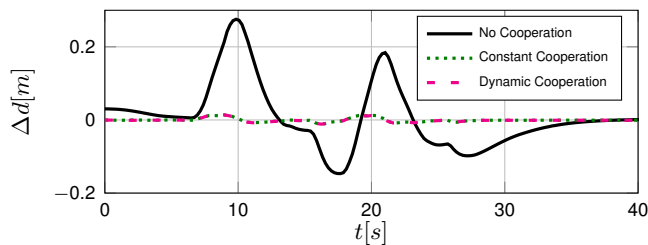


Fig. 8: Exp 1: Error in formation maintenance

#### ACKNOWLEDGMENTS

The European Research Council Starting Grant "Control based on Human Models (conhumo)" supported this work under grant agreement number 337654. Published under ©2018 IEEE with DOI 10.1109/IROS.2016.7759576 [18]

#### REFERENCES

- [1] C. Vesper and M. J. Richardson, "Strategic communication and behavioral coupling in asymmetric joint action," *Experimental brain research*, vol. 232, no. 9, pp. 2945–2956, 2014.
- [2] L. M. Sacheli, E. Tidoni, E. F. Pavone, S. M. Aglioti, and M. Candidi, "Kinematics fingerprints of leader and follower role-taking during cooperative joint actions," *Experimental brain research*, vol. 226, no. 4, pp. 473–486, 2013.
- [3] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.
- [4] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2. IEEE, 2002, pp. 1398–1403.
- [5] V. Koropouli, S. Hirche, and D. Lee, "Learning and generalizing force control policies for sculpting," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2012, pp. 1493–1498.

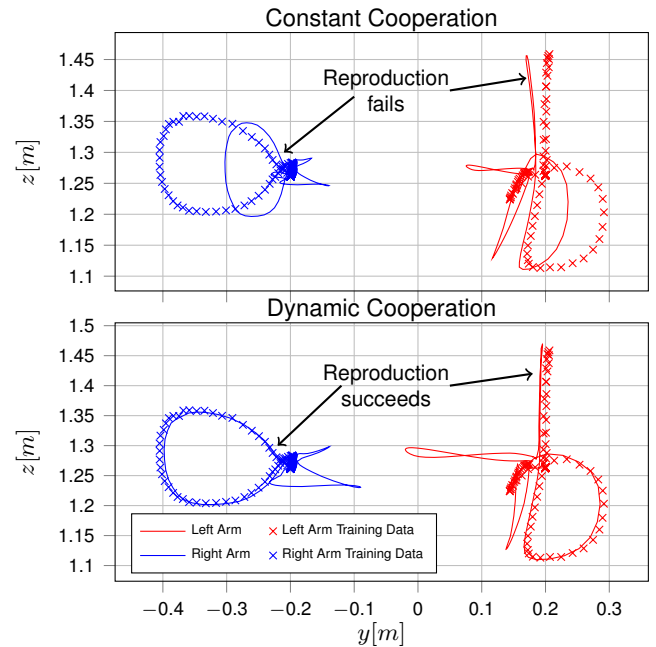


Fig. 9: Exp 2: Reproduction of a 2D movement

- [6] S. Schaal, "Dynamic movement primitives—a framework for motor control in humans and humanoid robotics," in *Adaptive Motion of Animals and Machines*. Springer, 2006, pp. 261–280.
- [7] T. Kulvicius, M. Biehl, M. J. Aein, M. Tamosiunaite, and F. Wörgötter, "Interaction learning for dynamic movement primitives used in cooperative robotic tasks," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1450–1459, 2013.
- [8] A. Gams, B. Nemeč, A. J. Ijspeert, and A. Ude, "Coupling movement primitives: Interaction with the environment and bimanual tasks," *IEEE Transactions on Robotics*, vol. 30/4, pp. 816–830, Aug. 2014.
- [9] J. Umlauf, D. Sieber, and S. Hirche, "Dynamic movement primitives for cooperative manipulation and synchronized motions," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2014, pp. 766–771.
- [10] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in Neural Information Processing Systems*, 2013, pp. 2616–2624.
- [11] S. Vijayakumar and S. Schaal, "Locally weighted projection regression: An  $o(n)$  algorithm for incremental real time learning in high dimensional space," in *International Conference on Machine Learning (ICML)*, 2000, pp. 1079–1086.
- [12] R. Saber and R. Murray, "Consensus protocols for networks of dynamic agents," in *Proceedings of the American Control Conference (ACC)*, vol. 2, June 2003, pp. 951–956.
- [13] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, March 2006.
- [14] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, ser. Adaptive Computation and Machine Learning. Cambridge, MA, USA: MIT Press, Jan. 2006.
- [15] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Model learning with local gaussian process regression," *Advanced Robotics*, vol. 23, no. 15, pp. 2015–2034, 2009.
- [16] S. Calinon, Z. Li, T. Alizadeh, N. G. Tsagarakis, and D. G. Caldwell, "Statistical dynamical systems for skills acquisition in humanoids," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, Nov. 2012, pp. 323–329.
- [17] J. R. Medina, D. Sieber, and S. Hirche, "Risk-sensitive interaction control in uncertain manipulation tasks," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 502–507.
- [18] Y. Fanger, J. Umlauf, and S. Hirche, "Gaussian processes for dynamic movement primitives with application in knowledge-based cooperation," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 3913–3919.