Integrated Bi-Manual Motion Generation and Control shaped for Probabilistic Movement Primitives

Jonathan Vorndamme[1], João Carvalho[2], Riddhiman Laha[1], Dorothea Koert[2,3], Luis Figueredo[1], Jan Peters[2], and Sami Haddadin[1]

***Abstract*— This work introduces a novel cooperative control framework that allows for real-time reactiveness and adaptation whilst satisfying implicit constraints stemming from probabilistic/stochastic trajectories. Stemming from task-oriented sampling and/or task-oriented demonstrations, e.g., learning based on motion primitives, such trajectories carry additional information often neglected during real-time control deployment. In particular, methods such as probabilistic movement primitives offer the advantage to capture the inherent stochasticity in human demonstrations – which in turn reflects human's understanding about task-variability and adaption possibilities. This information, however, is often poorly exploited and, mostly, used during offline trajectory planning stage. Our work instead introduces a novel real-time motion-generation strategy that explicitly exploits such information to improve trajectories according to changes in the environmental condition and robot task-space topology. The proposed solution is particularly well-suited for bimanual and coordinated systems where the increased kinematic complexity, tightly-coupled constraints and reduced workspace have detrimental effects on the manipulability, joint-limits, and are even capable of causing unstable behavior and task-failure. Our methodology addresses these challenges, and improves performance and task-execution by taking the confidence range region explicitly into account whilst maneuvering towards better configurations. Furthermore, it can directly cope with different closed-chain kinematics and task-space topologies, resulting for instance from different grasps. Experimental evaluations on a bi-manual Franka panda robot show that the proposed method can run in the inner control loop of the robot and enables successful execution of highly constrained tasks.**

I. Introduction

Future assistive robots will be challenged by a multitude of elaborate tasks in direct contact with everyday users. Nonetheless, planning for elaborate manipulation tasks often requires experienced programmers/roboticists to interpret task goals, conditions, constraints, and preferences – which overall hampers the rapid deployment required for human-coexistence application. Here, learning modular representations such as movement primitives [1]–[3] from human demonstrations offers a convenient concept which allows also end-users to teach new tasks to a robot. Intrinsically, the advantage is that probabilistic learning by demonstration (LbD) methods, like Probabilistic Movement Primitives (ProMPs) [1] and Stable Estimator of Dynamical Systems (SEDS) [4], enable to capture the inherent stochasticity within human's demonstrations. This information, however, is often poorly exploited and, mostly, used during the offline trajectory planning stage.

This work was partially funded by the Lighthouse Initiative Geriatronics by StMWi Bayern (grant 5140951), LongLeif GaPa gGmbH (grant 5140953), by the BMBF Projects 16SV7985 (KoBo34) and 01IS20045 (IKIDA) and the Deutsche Forschungsgemeinschaft (DFG) as part of EXC 2050/1 – Project 390696704 – Cluster of Excellence "Centre for Tactile Internet with Human-in-the-Loop" (CeTI). Please note that S. Haddadin has a potential conflict of interest as shareholder of Franka Emika GmbH.

[1]Technische Universität München (TUM), TUM School of Computation, Information and Technology (CIT), Department of Computer Engineering, Munich Institute of Robotics and Machine Intelligence (MIRMI), Germany. [2]Intelligent Autonomous Systems group, Technische Universitat Darmstadt (TUD), 64285 Germany. [3]Center for Cognitive Science, TUD, Germany Email:{jonathan.vorndamme, riddhiman.laha, luis.figueredo, haddadin}@tum.de, {joao.correia_carvalho, dorothea.koert, jan.peters}@tu-darmstadt.de

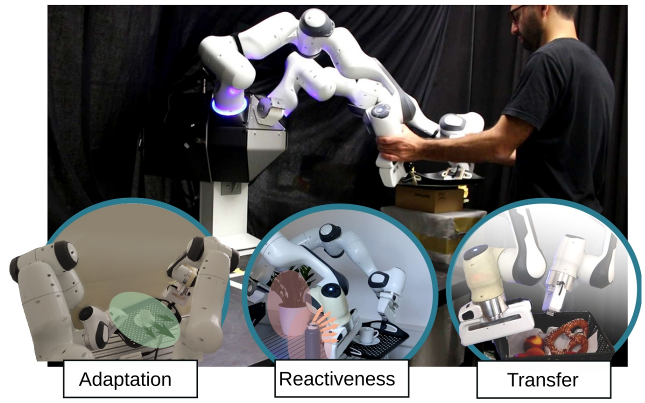

Fig. 1. Overview of our approach to human taught cooperative motions: After teaching an object-centric probabilistic movement primitive, the encoded covariance is used to adapt the trajectory in order to avoid joint limits and ensure good manipulability during task execution. In case unexpected obstacles are recognized during task execution, the motion generator automatically generates an evasive motion. Furthermore, the cooperative strategy allows for online adaptation of grasps whilst satisfying the cooperative learned trajectory and covariance-based constraints.

In this work, instead, we introduce a real-time reactive integrated motion-generation and control strategy that is attuned to probabilistic/stochastic trajectories – for instance, from the ones resulting from learning by demonstration applications. The resulting trajectory improves reactiveness, manipulability and range of motion whilst satisfying the limits imposed by the stochastic confidence interval from demonstrated motions – which implicitly accounts for human's understanding about variability and adaption possibilities of the manipulation task.

The proposed framework, although agnostic to the robot's kinematic structure – as it relies on task-space information – is well-suited for bimanual and coordinated systems which are often more challenging scenarios when it comes to LbD applications. Indeed, it remains an open challenge how to properly deploy learned trajectories in real-world human-centred scenarios where the cooperative task-space constraints, reduced workspace, and increased kinematic complexity leads to detrimental manipulability and configurations, e.g. joint-limits and singularities. These challenges are exacerbated in human-robot interaction and coexistence, particularly in terms of how to execute tasks at hand when unforeseen obstacles in the environment are introduced that need to be avoided. Here, the ability to generate reactive robot behavior whilst satisfying the limits regarding the demonstrated task-variability is a key requirement for future and assistive collaborative robots.

A. State of the Art

Probabilistic LbD methods offer a convenient approach allowing end-users to easily design new tasks whilst capturing human's implicit stochasticity in trajectories and preferences for the given task. Learning and embedding the cooperative manipulation information is nonetheless a much more complex procedure that involves both teaching in a large yet highly constrained space and extracting valuable information in an infinite range of possible combinations of decoupled or coupled task and joint spaces.

In recent years, from the pioneering work by Gams et al. [5], a few research groups have aimed at investigating the possibilities and limitations related to such challenge either in an asymmetrical manner [6], inducing – manually or learning – a dynamic coupling or force fields, [7]–[9], through joint-space compliance [10], or following task-parametrization, as in [11], [12], and learning to embed cooperative task-space information in joint-space through linear operators—or directly through the decoupled end-effectors using a single integrated GMM (Gaussian Mixture Model). The aforementioned results, despite the paramount advances to the problem, rely on an implicit coupling between arms that is highly dependent on the training setup. In other words, the random-variable coupling depends on the same initial configuration and small changes in grasp or even changing the gripper—not to mention the issues of changing to different arms—would require a new set of demonstrations as the linear operators mapping to joint-space would be completely different as well as the relative pose between arms.

Another challenge concerning cooperative manipulation are limitations towards the reactiveness and capabilities to compensate for planning inaccuracies and environment uncertainties. In this context, motion generation strategies based on reactive planners, e.g., [13]–[15], have shown to be more efficient paradigms for human-centred, dynamic and unstructured scenarios. The existing literature is however exiguous when it comes to the challenges of scaling up reactive solutions for higher DoF systems [16]–[18]. Among the scarce works, we highlight [17], [19]–[21] where cooperative task-space variables were explored with possibly task-relaxations to reduce task-dimensionality – ensuring better convergence and an improved reactive behavior. Therein, however, the task descriptions were manually defined, which required time and experienced roboticists to understand and interpret task conditions under proper representation and geometric set-based constraints. This work similarly makes explicit use of the cooperative-space formulation—which can be extended to a multitude of conditions and even multiple-arms without loss of generality [22], [23]—to improve the task execution of any LbD strategy with multiple demonstrations. Notwithstanding, most importantly, the proposed strategy also exploits the variability in user's demonstrations, which embeds important information about task flexibilities, preferences and constraints within the cooperative task-space. This allows for task-relaxations and corresponding geometric constraints to be extracted from the stochastic confidence interval from demonstrated motions in a more intuitive and practical manner.

B. Contribution

In other words, in this work, we make full use of the probabilistic information embedded into probabilistic/stochastic trajectories – resulting, for instance, from probabilistic movement primitives. The stochasticity of the demonstrations allows us to design for the first time an integrated reactive motion generation and control framework that produces improved coordinated motions and actively responds to unexpected events, such as newly introduced obstacles, in real-time. This reduces training/programming complexity for constraint satisfaction during execution, as well as informs the robotic system on the flexibilities and user-preferences—which are herein exploited in order to obtain task-space adaptation towards better manipulability and joint-limit avoidance. Furthermore, reactions to unforeseen events, e.g., dynamic obstacles are incorporated into the framework as well. The proposed adaptation and reaction schemes are also shaped according to the closed-kinematic chain and task-space topology, i.e., it will adapt for different grasps which allows the robot to overcome limitations in terms of teaching and execution with same grasps and poor manipulability and joint-space configuration—common issues whilst teaching bimanual systems. In other words, our solution, whilst satisfying the confidence space given by the demonstrations, is autonomously adapted to best fit the robot and grasps kinematics and topology.

The contributions of this work are summarized below.

1) An adaptive integrated motion generation and control scheme that is able to exploit the variance in human demonstrated trajectories for improving bi-manual manipulation, joint-limit conditions and task-dependent manipulability;
2) A method for obstacle avoidance in real-time, integrated with the original trajectory;
3) The ability to transfer the demonstrated trajectories to different grasping poses and robot topologies.

II. Reactive Cooperative Motion Generation for Probabilistic LbD

This section introduces our novel approach for generation of reactive and adaptive cooperative robot motions from and within the stochasticity of human demonstrations. First, to illustrate our concept, we will make use of the widely-known ProMPs briefly presented in Subsection II-A.[1] Second, we present the cooperative task-space, its subtasks manifolds, and how to reduce task dimensionality and enhance robot's flexibility through a cooperative set based task priority approach [19]. Third, we illustrate the robustness of our strategy to grasp generalization. Finally, we show the integrated solution that actively adapts and improves the robot motion and ensures desired reactiveness by exploiting the cooperative task variability underlying human demonstrations. The full overview of the proposed solution is depicted in Fig. 2, and detailed in subsections below.

A. Probabilistic Demonstrated Motion and its Variability

Movement Primitives (MPs) are a convenient way to represent time-based smooth robot and object movements [3]. In particular, probabilistic formulations allow to also capture variance in the demonstrations [1], [2], [24]. Here, we will use object-centric ProMPs to represent trajectories of object positions which allows constructing distributions that are conditioned on arbitrary time-steps and points inside a confidence interval of the demonstrations while, compared

[1]Notice that the proposed reactive cooperative motion generation strategy is valid for any stochastic/probabilistic trajectory with embedding information regarding the confidence range, which can be obtained through any LbD approach with multiple demonstrations.

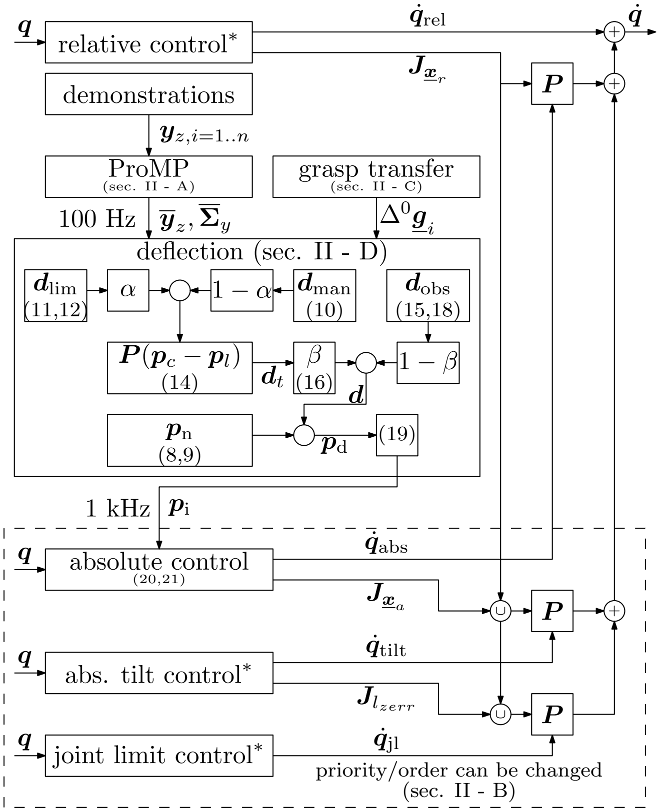

Fig. 2. Overview of our novel integrated control and motion generation approach tailored for probabilistic/stochastic trajectories. Use case stemming from the usage of ProMPs together with constant relative pose, and absolute, relative, tilt, and joint-limit controllers with Jacobians as defined in Table I and following the CoSTP (Cooperative Set-Based Task-Priority) control approach further detailed in [21] where you can also find the equations for the controllers marked with a "*". The symbol $\cup$ denotes the concatenation of the matrices input into the block. The $\boldsymbol{P}$ block depicts the nullspace projection of the input matrix $\boldsymbol{M}$ applied to the input vector $\boldsymbol{v}$, i.e. the ouput vector $\boldsymbol{o}$ would be $\boldsymbol{o} = (\boldsymbol{I} - \boldsymbol{M}^{\#}\boldsymbol{M})\boldsymbol{v}$. The # symbol denotes the robust pseudo inverse defined later in the section.

to other representations [25], relying on a small amount of data for training.

Modeling bimanual coordinated movements directly with ProMPs in joint or individual end-effector spaces results in problems due to the need to ensure correct alignment of trajectories between training and execution, as well as stable coordination in the presence of external disturbances and obstacles. Instead, in this work, we model the trajectories of the cooperative variables as a ProMP as shown in the next subsection. Note that such variables can be directly obtained from individual task-space or joints configurations, thus being easily extracted from any demonstration strategy.

Formally, an object-centric position-based ProMP is a compact representation of a trajectory, where a point $\boldsymbol{y}_z \in \mathbb{R}^3$ in the trajectory (the object's 3D position - translation part $\boldsymbol{p}$ of a dual quaternion (3)) is assumed to be a linear combination of N basis functions $\boldsymbol{y}_z = \boldsymbol{\Phi}_z^\intercal \boldsymbol{\omega}$, with $\boldsymbol{\Phi}$ a basis function matrix and $\boldsymbol{\omega}$ the learnable weights. z is a phase-variable that linearly maps time t to the $[0, 1]$ range. A distribution $p(\boldsymbol{\omega})$ over the weights is learned from multiple human demonstrations. Assuming the weights are Gaussian distributed $p(\boldsymbol{\omega}) = \mathcal{N}(\boldsymbol{\omega}; \boldsymbol{\mu}_{\boldsymbol{\omega}}, \boldsymbol{\Sigma}_{\boldsymbol{\omega}})$, the parameters of $p(\boldsymbol{\omega})$, the mean and covariance matrix, are obtained via maximum likelihood estimation. For more details on the training procedure and dimensions of all variables we refer the reader to [1], [26].

Let $\overline{\boldsymbol{y}}_z$ be a desired position to reach at (normalized) time-step $z(t)$ with covariance $\overline{\boldsymbol{\Sigma}}_y$. The conditional distribution over weights is computed with Bayes' rule for Gaussian distributions as $p(\boldsymbol{\omega}|\overline{\boldsymbol{y}}_z, \overline{\boldsymbol{\Sigma}}_y) \propto p(\overline{\boldsymbol{y}}_z|\boldsymbol{\omega}, \overline{\boldsymbol{\Sigma}}_y)p(\boldsymbol{\omega}) = \mathcal{N}\left(\boldsymbol{\omega}; \overline{\boldsymbol{\mu}}_{\boldsymbol{\omega}}, \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\omega}}\right)$, and due to the linear transformation, the resulting trajectory distribution is $p(\overline{\boldsymbol{\tau}}) = \mathcal{N}\left(\overline{\boldsymbol{\tau}}; \boldsymbol{\Phi}^\intercal \overline{\boldsymbol{\mu}}_{\boldsymbol{\omega}}, \boldsymbol{\Phi}^\intercal \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\omega}} \boldsymbol{\Phi}\right)$, with

$$\overline{\boldsymbol{\mu}}_{\boldsymbol{\omega}} = \boldsymbol{\mu}_{\boldsymbol{\omega}}^* + \mathbf{K}\left(\overline{\boldsymbol{y}}_z - \boldsymbol{\phi}_z^\intercal \boldsymbol{\mu}_{\boldsymbol{\omega}}^*\right), \quad \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\omega}} = \boldsymbol{\Sigma}_{\boldsymbol{\omega}}^* - \mathbf{K}\boldsymbol{\phi}_z^\intercal \boldsymbol{\Sigma}_{\boldsymbol{\omega}}^* \tag{1}$$

$$\mathbf{K} = \boldsymbol{\Sigma}_{\boldsymbol{\omega}}^* \boldsymbol{\phi}_z \left(\overline{\boldsymbol{\Sigma}}_y + \boldsymbol{\phi}_z^\intercal \boldsymbol{\Sigma}_{\boldsymbol{\omega}}^* \boldsymbol{\phi}_z\right)^{-1}, \quad \boldsymbol{\Phi}_z = \mathbf{I}_d \otimes \boldsymbol{\phi}_z,$$

where $\boldsymbol{\mu}_{\boldsymbol{\omega}}^*$ and $\boldsymbol{\Sigma}_{\boldsymbol{\omega}}^*$ are the learned weights of the unconditioned ProMP, and $\boldsymbol{\phi}_z$ is a vector of basis-functions activations at time-step $z(t)$. The covariance matrix for each time-step in the conditioned ProMP trajectory is obtained with

$$\boldsymbol{\Sigma}_z = \boldsymbol{\Phi}_z^\intercal \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\omega}} \boldsymbol{\Phi}_z. \tag{2}$$

B. Cooperative Set-Based Task-Priority (CoSTP)

As standard stochastic LbD methods do not encode the variables and transformations in the cooperative task-space manifold [5]–[10], we first describe how to capture the same from individual configurations. Consider a two-arm robot where $\underline{\boldsymbol{x}}_1$ and $\underline{\boldsymbol{x}}_2$ represent the pose of the left and right end-effectors. The pose is described using unit dual-quaternions as[2]

$$\underline{\boldsymbol{x}} = \boldsymbol{r} + \tfrac{1}{2}\varepsilon \boldsymbol{p}\boldsymbol{r}, \tag{3}$$

where $\boldsymbol{r} = \cos(\phi/2) + \sin(\phi/2)\boldsymbol{n}$ represents a rotation along axis $\boldsymbol{n}$ with angle ϕ [29], $\boldsymbol{p}$ is a pure quaternion that represents the translation, and ε is such that $\varepsilon{\neq}0$ but $\varepsilon^2{=}0$ [30]. Under multiplication, the set of elements $\underline{\boldsymbol{x}}$ form the unit dual-quaternion group $\mathrm{Spin}(3) \ltimes \mathbb{R}^3$ with inverse given by the conjugate $\underline{\boldsymbol{x}}^* = \boldsymbol{r}^* + \frac{1}{2}\varepsilon \boldsymbol{r}^* \boldsymbol{p}^*$. Dual quaternion elements $\underline{\boldsymbol{h}}$ can also be described by $\underline{\boldsymbol{h}} = \mathcal{P}(\underline{\boldsymbol{h}}) + \varepsilon\, \mathcal{D}(\underline{\boldsymbol{h}})$, where $\mathcal{P}(\underline{\boldsymbol{h}})$ and $\mathcal{D}(\underline{\boldsymbol{h}})$ are the primary and dual components.

The cooperative dual-task space formulation (see [18], [20], for further details) explores sequences of transformations to depict fundamental cooperative variables, i.e. the absolute pose[3] and relative pose between arms,

$$\underline{\boldsymbol{x}}_r = \underline{\boldsymbol{x}}_2^* \underline{\boldsymbol{x}}_1, \quad \text{and} \quad \underline{\boldsymbol{x}}_a = \underline{\boldsymbol{x}}_2 \underline{\boldsymbol{x}}_{r/2} \tag{4}$$

where $\underline{\boldsymbol{x}}_{r/2}$ is the transformation that corresponds to half of the angle ϕ_r around the axis $\boldsymbol{n}_r = \hat{\imath} n_x + \hat{\jmath} n_y + \hat{k} n_z$ of the $\mathcal{P}(\underline{\boldsymbol{x}}_r)$ and half of the translation between the two arms [18].

In our case, the coordinated motion is defined over desired intervals along some of these cooperative variables. For instance, while handling a ball, we often want to control just the relative distance between arms as we move along the absolute position, i.e. without orientation control. Meanwhile, in a task as shown in Fig. 8, the robot needs to control additionally the complete relative pose and the tilt angle along a given axis (to hold the tray stray). Exploring the subtasks reduces dimensionality and enlarges the robots nullspace for ensuring better manipulability, cooperative workspace, joint-space motion, and in addition, to relax constraints allowing for further reaction and adaptation to unforeseen events.

[2]Rigid bodies transformations can also be described through other non-minimal representations, e.g., homogeneous transformation matrices, yet describing the cooperative task-space using dual quaternion algebra has several advantages in terms of representation, computational complexity and, most important, its capability to extract geometric properties and primitives even in highly constrained contexts [18], [20], [23], [27], [28].

[3]The absolute pose is located between end-effectors w.r.t. to a common coordinate system yet, without loss of generality, it can be shifted by means of a constant transformation.

TABLE I

MAIN GEOMETRIC COOPERATIVE TASKS AND TASK JACOBIANS FOR BOTH RELATIVE OR ABSOLUTE VARIABLES ACCORDING TO DEFINITION 1

Task Primitive	Task Jacobian	DOFs
Rel/Abs position ($\boldsymbol{p}\in\Re_p$)	$\boldsymbol{J}_{\boldsymbol{p}_\chi}=\left[\overset{+}{\boldsymbol{H}}(\underline{\boldsymbol{x}}_2^*)\boldsymbol{J}_{\underline{\boldsymbol{x}}_1} \quad \overset{-}{\boldsymbol{H}}(\underline{\boldsymbol{x}}_1)\boldsymbol{J}^*_{\underline{\boldsymbol{x}}_2}\right]$	3
Rel/Abs orientation ($\boldsymbol{r}\in\Re_o$)	$\boldsymbol{J}_{\boldsymbol{r}_\chi}=\boldsymbol{J}_{\mathcal{P}(\underline{\boldsymbol{x}}_\chi)}$	3
Rel/Abs distance ($d\in\Re_d$)	$\boldsymbol{J}_{d_\chi}=2(\mathrm{vec}_4^T\,\boldsymbol{p}_\chi)\boldsymbol{J}_{\boldsymbol{p}_\chi}$	1
Rel/Abs tilt ($\phi_\iota\in\Re_{\phi_\iota}$)	$\boldsymbol{J}_{l_{zerr}}=-2(\boldsymbol{l}-\boldsymbol{l}_z)^T\boldsymbol{J}_{r_z}$	1
Rel/Abs singul. ($\sigma_{min}\in\mathbb{R}$)	$\boldsymbol{J}_\sigma=\frac{\partial\sigma_{\min}(\chi)}{\partial\boldsymbol{q}}$	1
Joint limits ($q_i\in\mathbb{R}^n$)	$\boldsymbol{J}_{q_i}=q_{i,c}-q_i,\ q_{i,c}=\frac{(\underline{q_i}+\overline{q_i})}{2}$	1

For brevity, herein, we focus on the following cooperative task-spaces: the absolute position, orientation, distance, and deviation angle between Plücker lines, e.g., taking the line along the z-axis, $\boldsymbol{l}_z=\boldsymbol{r}\hat{\boldsymbol{k}}\boldsymbol{r}^*$, to maintain the orientation of the manipulated object, and their correspondence in terms of the relative pose between arms as well as joint limit avoidance. These cooperative geometric subsets can be formally defined as follows.

Definition 1: For a given set $\mathfrak{S}\subseteq \mathrm{Spin}(3)\ltimes\mathbb{R}^3$, the following proper subsets can be drawn from geometric structures of interest with regard to this set,

$$\Re_p(\mathfrak{S})=\left\{\boldsymbol{p}\in\mathbb{H}_0 \mid \boldsymbol{p}=\boldsymbol{\mathcal{T}}\left(\underline{\boldsymbol{x}}_g\right),\ \underline{\boldsymbol{x}}_g\in\mathfrak{S}\right\},$$
$$\Re_o(\mathfrak{S})=\left\{\boldsymbol{r}\in\mathrm{Spin}(3) \mid \boldsymbol{r}=\mathcal{P}\left(\underline{\boldsymbol{x}}_g\right),\ \underline{\boldsymbol{x}}_g\in\mathfrak{S}\right\},$$
$$\Re_d(\mathfrak{S})=\left\{d\in\mathbb{R} \mid d=\left\|\boldsymbol{\mathcal{T}}\left(\underline{\boldsymbol{x}}_g\right)\right\|,\ \underline{\boldsymbol{x}}_g\in\mathfrak{S}\right\},$$
$$\Re_{\phi_\iota}(\mathfrak{S})=\left\{\phi_\iota\in\mathbb{R}\,|\,\phi_\iota=\cos^{-1}\left(\langle\boldsymbol{l}_z,\boldsymbol{l}\rangle\right),\boldsymbol{l}_z=\boldsymbol{r}\hat{\boldsymbol{k}}\boldsymbol{r}^*\right\},$$

where $\boldsymbol{\mathcal{T}}\left(\underline{\boldsymbol{x}}_g\right)\triangleq 2\mathcal{D}\left(\underline{\boldsymbol{x}}_g\right)\mathcal{P}\left(\underline{\boldsymbol{x}}_g\right)^*$ is the translation with $\mathbb{H}_0$ being the pure quaternions set, isomorphic to $\mathbb{R}^3$. Briefly, ϕ_ι describes the opening angle of a solid cone defined by the rotation of the body z-axis to the coordinate frame, i.e., $\boldsymbol{l}_z$ around a desired Plücker line $\boldsymbol{l}$.

It is important to highlight that human demonstrations are encoded/constrained within such cooperative subtask spaces. These define the manifolds in which the task is defined. From the robot's perspective, these tasks need to be properly mapped to joint-space actions through correspondent Jacobians, as shown in Table I.[4] The task Jacobians in Table I rely on the absolute and relative pose (4) and on the cooperative Jacobians

$$\boldsymbol{J}_{\underline{\boldsymbol{x}}_r}=\left[\overset{+}{\boldsymbol{H}}(\underline{\boldsymbol{x}}_2^*)\boldsymbol{J}_{\underline{\boldsymbol{x}}_1} \quad \overset{-}{\boldsymbol{H}}(\underline{\boldsymbol{x}}_1)\boldsymbol{J}^*_{\underline{\boldsymbol{x}}_2}\right], \tag{5a}$$

$$\boldsymbol{J}_{\underline{\boldsymbol{x}}_a}=\left[\overset{-}{\boldsymbol{H}}(\underline{\boldsymbol{x}}_{r/2})\boldsymbol{J}_{\underline{\boldsymbol{x}}_{2ext}}+\overset{+}{\boldsymbol{H}}(\underline{\boldsymbol{x}}_2)\boldsymbol{J}_{\underline{\boldsymbol{x}}_{r/2}}\right], \tag{5b}$$

where $\boldsymbol{q}=[\boldsymbol{q}_1^T \quad \boldsymbol{q}_2^T]^T\in\mathbb{R}^n$ is the augmented joint vector and $\boldsymbol{J}_{\underline{\boldsymbol{x}}_i}=\partial f_i/\partial\boldsymbol{q}_i$ is the analytical Jacobian, which can be derived using dual quaternion algebra. The matrices $\overset{+}{\mathrm{H}}$ and $\overset{-}{\mathrm{H}}$ are Hamilton operators that can be used to commute terms when performing transformations, such that $\mathrm{vec}\,\underline{\boldsymbol{z}}=\overset{+}{\mathrm{H}}\left(\underline{\boldsymbol{x}}\right)\mathrm{vec}\,\underline{\boldsymbol{y}}=\overset{-}{\mathrm{H}}\left(\underline{\boldsymbol{y}}\right)\mathrm{vec}\,\underline{\boldsymbol{x}}$ where $\mathrm{vec}:\mathcal{H}\rightarrow\mathbb{R}^8$ and $\underline{\boldsymbol{z}}=\underline{\boldsymbol{x}}\underline{\boldsymbol{y}}$, (see [22], [23], [31]).

Finally, in this paper, the cooperative motion is controlled through the *Cooperative Set-Based Task-Priority (CoSTP)* introduced in [19]. Additional task priorities and, corresponding relaxations are easily obtained from the framework in [19], nonetheless for brevity and clarity, we will focus on the control and motion generation scheme as shown in Fig. 2. Notice, additional schemes can be designed without loss of generality. The overall motion scheme from Fig. 2 leads to the closed-loop control equation

$$\dot{\boldsymbol{q}}=\dot{\boldsymbol{q}}_{\mathrm{rel}}+\boldsymbol{P}\left(\boldsymbol{J}_{\underline{\boldsymbol{x}}_r}\right)\dot{\boldsymbol{q}}_{\mathrm{abs}}+\boldsymbol{P}\left(\left[\boldsymbol{J}_{\underline{\boldsymbol{x}}_r}^T \quad \boldsymbol{J}_{\underline{\boldsymbol{x}}_a}^T\right]^T\right)\dot{\boldsymbol{q}}_{\mathrm{tilt}} + \boldsymbol{P}\left(\left[\boldsymbol{J}_{\underline{\boldsymbol{x}}_r}^T \quad \boldsymbol{J}_{\underline{\boldsymbol{x}}_a}^T \quad \boldsymbol{J}_{\underline{\boldsymbol{x}}_{l_{zerr}}}^T\right]^T\right)\dot{\boldsymbol{q}}_{\mathrm{jl}}, \tag{6a}$$

$$\dot{\boldsymbol{q}}_{\mathrm{rel}}=\boldsymbol{J}_{\underline{\boldsymbol{x}}_r}^{\#}\left(\mathrm{vec}\,\underline{\boldsymbol{x}}_{r,0}-\mathrm{vec}\,\underline{\boldsymbol{x}}_r\right),\quad \dot{\boldsymbol{q}}_{\mathrm{abs}}=\dot{\boldsymbol{q}}_{\mathrm{c}}+\dot{\boldsymbol{q}}_{\mathrm{ff}}, \tag{6b}$$

$$\dot{\boldsymbol{q}}_{\mathrm{tilt}}=\boldsymbol{J}_{\underline{\boldsymbol{x}}_{l_{zerr}}}^{\#}\left|\mathrm{vec}\left(\boldsymbol{l}-\boldsymbol{l}_z\right)\right|^2,\quad \dot{\boldsymbol{q}}_{\mathrm{jl}}=\frac{1}{4}\boldsymbol{J}_{\underline{\boldsymbol{x}}_{q_i}}^{\#}\left(\boldsymbol{J}_{\underline{\boldsymbol{x}}_{q_i}}^{\mathrm{T}}\boldsymbol{J}_{\underline{\boldsymbol{x}}_{q_i}}\right)$$

assuming that the priority order is absolute control followed by the tilt and joint limit avoidance. The # symbol defines the robust pseudo-inverse of a matrix $\boldsymbol{M}$, i.e., $\boldsymbol{M}^{\#}=\boldsymbol{M}^{\mathrm{T}}\left(\boldsymbol{M}\boldsymbol{M}^{\mathrm{T}}+\lambda_\epsilon\boldsymbol{I}\right)^{-1}$, where $\boldsymbol{I}$ is the identity matrix weighted by $\lambda_\epsilon>0$ for damping the pseudo-inverse. Furthermore, $\underline{\boldsymbol{x}}_{r,0}$ denotes the relative pose at the beginning of the task. For the definition of $\dot{\boldsymbol{q}}_{\mathrm{c}}$ and $\dot{\boldsymbol{q}}_{\mathrm{ff}}$, see sec. II-D. The resulting $\dot{\boldsymbol{q}}$ is integrated and input to a low level joint impedance controller.

Formally, each task is governed by its corresponding set $\mathcal{S}_i$ and an error $\boldsymbol{e}_i$ having an upper and lower bound $(\underline{\boldsymbol{e}_i},\overline{\boldsymbol{e}_i})$. We define regions where a particular task remains active based on these conditions: $(a)\boldsymbol{e}_i\in\left(\underline{\boldsymbol{e}_i},\overline{\boldsymbol{e}_i}\right)(b)\boldsymbol{e}_i\succeq\overline{\boldsymbol{e}_i}$ and $\dot{\boldsymbol{e}}_i\preceq 0(c)\boldsymbol{e}_i\preceq\overline{\boldsymbol{e}_i}$ and $\dot{\boldsymbol{e}}_i\succeq 0$ Therefore, if a task is close to the boundary of the set, we push it away and then it is deactivated so that it acts only in the lower priority of the self-motion, i.e. the nullspace. The stability analysis follows standard Lyapunov-Krasovskii analysis based on [32] and defines conditions for control values. Additional details can be found in [19].

Overall, the CoSTP framework allows us to shift task priority between cooperative geometric tasks according to boundary conditions – enlarging the ability of the robot to avoid dicey scenarios, and to adapt and react to unforeseen events. Among the main novelties, herein, is the ability to adapt the trajectory satisfying the user demonstrated variability in terms of the stochastic confidence interval as described in section II-D.

C. Cooperative System Generalization and Transfer

Our approach as stressed is oriented at encoding transformation and variables within the cooperative task-space. In this subsection, we will show that the consequence of this setup is that the distributions are independent of the closed kinematic chain of the robot and its configuration. Hence, it is also independent of the training conditions. Take, for instance, the trajectory taught with an initial grasp pose ${}^0\underline{\boldsymbol{x}}_1$ and ${}^0\underline{\boldsymbol{x}}_2$ as shown in Fig. 3. During execution, we may want to grasp a different object with either a different cooperative system or, at least different grasps as shown in the right side of Fig. 3. In this case, assuming a constant transformation between grippers and end-effectors, the cooperative absolute pose and its corresponding trajectory along time are respectively described according to (4) by ${}^1\underline{\boldsymbol{x}}_a(t)={}^1\underline{\boldsymbol{x}}_2^*(t)\,{}^1\underline{\boldsymbol{x}}_1(t)$ and ${}^2\underline{\boldsymbol{x}}_a(t)={}^2\underline{\boldsymbol{x}}_2^*(t)\,{}^2\underline{\boldsymbol{x}}_1(t)$ for both grasps (b) and (c).

Both cooperative variables can therefore be mapped back to the original demonstration—or resulting trajectory after ProMPs conditioning—described along ${}^0\underline{\boldsymbol{x}}_a(t)={}^0\underline{\boldsymbol{x}}_2^*(t)\,{}^0\underline{\boldsymbol{x}}_1(t)$, by means of a grasp transformation

[4] In addition to the cooperative tasks, additional subtasks are the manipulability optimization and joint limit avoidance ($\underline{q_i}$ and $\overline{q_i}$ being the lower and upper limit of the respective joint). The latter is a special case where task and joint space coincide.

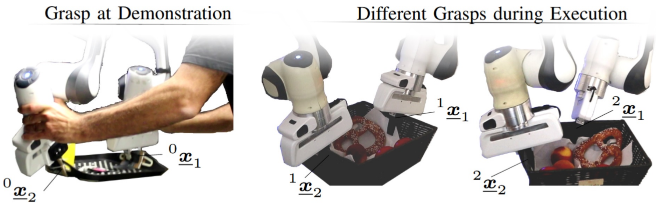

Fig. 3. Transfer generalization between demonstrations to different grasps ensures robustness in terms of shaping the cooperative variables within the encoded demonstrations.

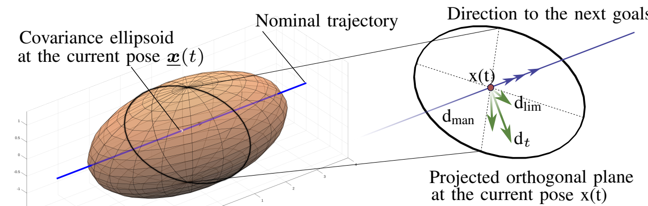

Fig. 4. Task deflection: $\boldsymbol{d}_{\mathrm{man}}$ and $\boldsymbol{d}_{\mathrm{lim}}$ are conditioned to be within limits given by the covariance ellipsoid and convexly combined (13) such that the result always respects the covariance ellipsoid.

$\Delta^0\underline{\boldsymbol{g}}_i$, $i{=}\{1,2\}$ given by $\Delta^0\underline{\boldsymbol{g}}_i \triangleq {}^i\underline{\boldsymbol{x}}_a^{*}\,{}^0\underline{\boldsymbol{x}}_a$. Now, according to (4), we have

$$\begin{aligned}\Delta^0\underline{\boldsymbol{g}}_i &= \left({}^i\underline{\boldsymbol{x}}_2\,{}^i\underline{\boldsymbol{x}}_{r/2}\right)^{*}\left({}^0\underline{\boldsymbol{x}}_2\,{}^0\underline{\boldsymbol{x}}_{r/2}\right)\\ &={}^i\underline{\boldsymbol{x}}_{r/2}^{*}\left({}^i\underline{\boldsymbol{x}}_2^{*}\,{}^0\underline{\boldsymbol{x}}_2\right){}^0\underline{\boldsymbol{x}}_{r/2},\end{aligned} \tag{7}$$

which briefly describes the frame transformation from the current frame i to the cooperative variables from the original frame (from training), i.e., 0. In other words, ${}^0\underline{\boldsymbol{x}}_a = {}^i\underline{\boldsymbol{x}}_a\Delta^0\underline{\boldsymbol{g}}_i$. In the case, one of the grippers is held constant, e.g. ${}^i\underline{\boldsymbol{x}}_2^{*} = {}^0\underline{\boldsymbol{x}}_2$, the transformation is simplified to a simple change of relative frame $\Delta^0\underline{\boldsymbol{g}}_i = {}^i\underline{\boldsymbol{x}}_{r/2}^{*}\,{}^0\underline{\boldsymbol{x}}_{r/2}$.[5]

Notice that this provides the system with a cooperative-centric perspective of the demonstration, which allows flexibility in terms of the robot motion and robustness in terms of grasping and arm configuration. The executed nominal trajectory will be always corrected by the transformation $\Delta^0\underline{\boldsymbol{g}}_i$, which fits the encoded demonstration, while the executed trajectory will be deflected to ensure better manipulability, joint-limit avoidance and overall a large region for reaction in accordance with the tightly-coupled kinematic chain system capability, i.e. according to the grasp and individual arm kinematics. These features are described in the following.

D. Adaptive Reactive Motion Generation

We now propose our cooperative motion generation framework that **(i)** follows the trajectory provided by the demonstrations; **(ii)** deflects the trajectories inside the two sigma confidence interval defined by the covariance of the ProMP to achieve higher manipulability and to avoid joint limits; and **(iii)** reactively evades unforeseen obstacles in real-time.

1) Trajectory Following Motion Generation: The trajectory defined by the ProMP is interpolated in equi-distant points that are sent to the trajectory generator one by one. The trajectory generator plans ahead for two points and requests the next trajectory point when the first point is reached based on the commanded position. The velocity v_{ts} for a trajectory segment is defined by

$$v_{\mathrm{ts}} = \min\left(\frac{|\boldsymbol{p}_{\mathrm{c}} - \boldsymbol{p}_{\mathrm{l}}|_2}{10\cdot T}, v_{\max}\right), \tag{8}$$

with $\boldsymbol{p}_l$, $\boldsymbol{p}_c$ and T denoting the last and the currently processed point of the trajectory as well as the low-level joint impedance controller and motion generator cycle time (herein, $T{=}1$ ms). The maximum Cartesian velocity is given by $v_{\max}$. The velocity bound provides an additional safety feature, required especially when the robot needs to adapt to large task deflections and large or complex obstacles to trespass. The trajectory segments are interpolated according to the current trajectory, i.e. the nominal desired position $\boldsymbol{p}_{\mathrm{n}}$ is updated according to

$$\boldsymbol{p}_{\mathrm{n}} = \boldsymbol{p}_{\mathrm{n}} + v_{\mathrm{ts}}T\frac{\boldsymbol{p}_c - \boldsymbol{p}_l}{|\boldsymbol{p}_c - \boldsymbol{p}_l|_2}, \tag{9}$$

in each time step.

2) Adaptive Deflection: Probabilistic/stochastic trajectories, as the ones stemming from probabilistic motion primitives naturally define some flexibility in task execution in terms of the observed covariance during teaching. In LbD deployment, this information is often used solely to sample a trajectory that fits the current purpose in offline stage. Instead, our approach exploits this flexibility, namely the confidence interval, to control or optimize, in real-time, different aspects concerning the manipulation task at hand by means of task-space deflection from the nominal trajectory bounded by the human's demonstrated confidence interval. This is advantageous in contrast to a fixed limit for such optimizations as we can have more space for optimization in regions where the covariance of demonstrations is high and thus, the required task precision is low but we will have all the precision that was demonstrated in regions where the demonstrations show low covariance.

In contrast to e.g. [33], where the covariance of the trajectory is used to adapt the stiffness, we consider two paramount aspects for the success of cooperative manipulation tasks, the cooperative manipulability and joint-limit avoidance.[6]

a) Cooperative Manipulability: In order to guide the robot to regions of greater manipulability, we compute the singular value decomposition of the absolute pose Jacobian and deflect the robot in accordance with the direction of the singular vector $\boldsymbol{u}_{\min}$ corresponding to the smallest singular value $\boldsymbol{\sigma}_{\min}$ as in [34] but with different scalar gains,

$$\boldsymbol{d}_{\mathrm{man}} = \lambda_{\mathrm{man}}\boldsymbol{u}_{\min}\min_{j\in\{1,2,3\}}\frac{2\boldsymbol{\Sigma}_z(j,j)}{\boldsymbol{e}_j\cdot\boldsymbol{u}_{\min}}, \tag{10}$$

$$\lambda_{\mathrm{man}} = \begin{cases}\min((1-\frac{\sigma_{\min}}{\epsilon_\sigma})k_{\mathrm{man}},1), & \text{if } \sigma_{\min} < \epsilon_\sigma;\\ 0, & \text{otherwise},\end{cases}$$

with $\boldsymbol{\Sigma}_z(j,j)$ and $\boldsymbol{e}_j$ denoting the j-th element on the diagonal of $\boldsymbol{\Sigma}_z$ and the unit vector of the j-th coordinate, respectively. As $\boldsymbol{u}_{\min}$ is only defined up to the sign, one needs to make sure, that the direction of deflection is continuous to avoid oscillations. In order to avoid leaving the two sigma region of the demonstrations, λ_{man} is capped at 1.

b) Task-space Guided Joint-limit Avoidance: In order to avoid running into joint limits during trajectory execution, the robot is commanded to move towards the Cartesian direction that a movement of all joints towards their center position

[5] The variable $\underline{\boldsymbol{x}}_{r/2}$ can be easily obtained by exponential mappings from the relative pose between arms as $\underline{\boldsymbol{x}}_{r/2} = \exp\left(\frac{1}{2}\log\underline{\boldsymbol{x}}_r\right)$, [23], [31].

[6] The task-space guided joint-limit avoidance is not to be confused with nullspace optimization of joints. In this work, we propose a task-space modification that will actively bring the cooperative system to a better pose in terms of range of motion by modifying the main task whilst keeping the deflection bounded. This should be seen as an additional robotic feature that can be used together with standard joint-limit avoidance.

would result, i.e.

$$\boldsymbol{d}_{\text{lim}} := \lambda_{\text{lim}} \frac{\boldsymbol{J}_{\boldsymbol{x}_a}(\boldsymbol{q}_c - \boldsymbol{q})}{\left|\boldsymbol{J}_{\boldsymbol{x}_a}(\boldsymbol{q}_c - \boldsymbol{q})\right|_2} \min_{j\in\{1,2,3\}} \frac{2\boldsymbol{\Sigma}_z(j,j)\left|\boldsymbol{J}_{\boldsymbol{x}_a}(\boldsymbol{q}_c - \boldsymbol{q})\right|_2}{\boldsymbol{e}_j \cdot \boldsymbol{J}_{\boldsymbol{x}_a}(\boldsymbol{q}_c - \boldsymbol{q})}, \tag{11}$$

$$\lambda_{\text{lim}} := \min\left(1, k_{\text{lim}} \max_{i=1\dots n}\left(\frac{1}{q_{\text{r}}} \operatorname{dz}\left(\frac{|q_i - q_{i,\text{c}}|}{q_{i,\max} - q_{i,\text{c}}}, 1 - q_{\text{r}}\right)\right)\right).$$

where $q_i, q_{i,\max}, q_{i,\text{c}}$ denote the current, maximum and center joint position of joint i. The minima in (11) and (10) scale the deflection such that it stays inside the 2-sigma range of the demonstrations. The parameter q_r defines the percentage of the range of motion the robot will start to deflect in order to avoid limits, e.g. when q_{r}=0.5, the robot starts deflecting as soon as one of the joints reaches the outer 50% of its range. The speed of attainment of the maximum deflection factor to one can be tuned with the parameter k_{lim}.

The deflection, in (11), maps the joint velocity that minimizes the error $(q_c-q)^T(q_c-q)$ to cooperative task-space velocities, i.e., the absolute pose, by means of the Jacobian $\boldsymbol{J}_{\boldsymbol{x}_a}$. The task-space action is normalized as scaled accordingly to the worst joint configuration after dead-zone correction through

$$\operatorname{dz}(a,b) := \begin{cases} a+b, & a < -b \\ 0, & |a| < |b| \\ a-b, & a > b. \end{cases} \tag{12}$$

These deflections are scaled such that they use only $\lambda_{\text{lim}}/\lambda_{\text{man}}$ portion of the two sigma range defined by the covariance of the conditioned ProMP (computed as in (2)), as shown in Fig. 4. Afterwards, they are convexly combined via the parameter α

$$\boldsymbol{d}_{\text{t}}^* := \alpha \boldsymbol{d}_{\text{lim}} + (1-\alpha)\boldsymbol{d}_{\text{man}}, \tag{13}$$

for the preliminary deflection $\boldsymbol{d}_{\text{t}}^*$.

The **final task deflection**, as highlighted in Fig. 4, stems from the projection of (13) onto the plane that is orthogonal to the direction towards the next goal. In other words, we have

$$\boldsymbol{d}_{\text{t}} := \boldsymbol{d}_{\text{t}}^* - \frac{\boldsymbol{d}_{\text{t}}^* \cdot (\boldsymbol{p}_c - \boldsymbol{p}_l)}{|\boldsymbol{p}_c - \boldsymbol{p}_l|_2^2}(\boldsymbol{p}_c - \boldsymbol{p}_l), \tag{14}$$

for the final task deflection $\boldsymbol{d}_{\text{t}}$. This strategy can be used on top of nullspace optimization and even for non-redundant manipulators/tasks.

3) Reactive Deflection: In the presence of obstacles, the task deflection is overwritten by a deflection that will guide us smoothly around the obstacle. We will denote this term with $\boldsymbol{d}_{\text{obs}}$. It is computed based on the distance d_{obs} to the obstacle

$$d_{\text{obs}} := \min\left(|\boldsymbol{p} - \boldsymbol{o}_{\boldsymbol{p}}|_2, |\boldsymbol{p}_{\text{n}} - \boldsymbol{o}_{\boldsymbol{p}_{\text{n}}}|_2\right), \tag{15}$$

where $\boldsymbol{o}_{\boldsymbol{x}}$ denotes the point on the obstacle surface closest to $\boldsymbol{x}$. The parameter

$$\beta := \max\left(\frac{d_{\text{obs}} - d_{\text{min}}}{d_{\text{max}} - d_{\text{min}}}, 0\right), \tag{16}$$

ensures a smooth blending between the task deflection and the obstacle deflection via a convex concurrent approach

$$\boldsymbol{d} := \beta \boldsymbol{d}_t + (1-\beta)\boldsymbol{d}_{\text{obs}}\ . \tag{17}$$

The parameter d_{max} denotes the distance at which the obstacle avoidance starts to evade and deflect the nominal trajectory. On the other hand, d_{min} depicts the point at which the task deflection is fully replaced by the obstacle avoidance deflection. Finally, $\boldsymbol{d}_{\text{obs}}$ can be computed to

$$\boldsymbol{d}_{\text{obs}} := \frac{(\boldsymbol{p}_{\text{n}}(t) - \boldsymbol{o}_{\boldsymbol{p}_{\text{n}}(t)}) \times (\boldsymbol{p}_{\text{n}}(t) - \boldsymbol{p}_{\text{n}}(t-T))}{\left|(\boldsymbol{p}_{\text{n}}(t) - \boldsymbol{o}_{\boldsymbol{p}_{\text{n}}(t)}) \times (\boldsymbol{p}_{\text{n}}(t) - \boldsymbol{p}_{\text{n}}(t-T))\right|_2}\chi, \tag{18}$$

$$\chi := \left(10\left(1 - \frac{d_{\text{obs}}}{d_{\text{max}}}\right)^3 - 15\left(1 - \frac{d_{\text{obs}}}{d_{\text{max}}}\right)^4 + 6\left(1 - \frac{d_{\text{obs}}}{d_{\text{max}}}\right)^5\right).$$

where χ is calculated as a 5th order spline that interpolates from 0 to 1 and has 0 derivative of first and second order at 0 and 1. Herein, for brevity, we are considering obstacles without velocity information, that is, unknown and unforeseen obstacles that are perceived only during execution. Nonetheless, without loss of generality, the velocity could also be integrated to the resulting deflected trajectory ensuring any motion of the obstacle is pruned.

4) Resulting Trajectory: The resulting deflection $\boldsymbol{d}$, in (17), is deployed alongside the nominal position, $\boldsymbol{p}_{\text{n}}$, yielding a

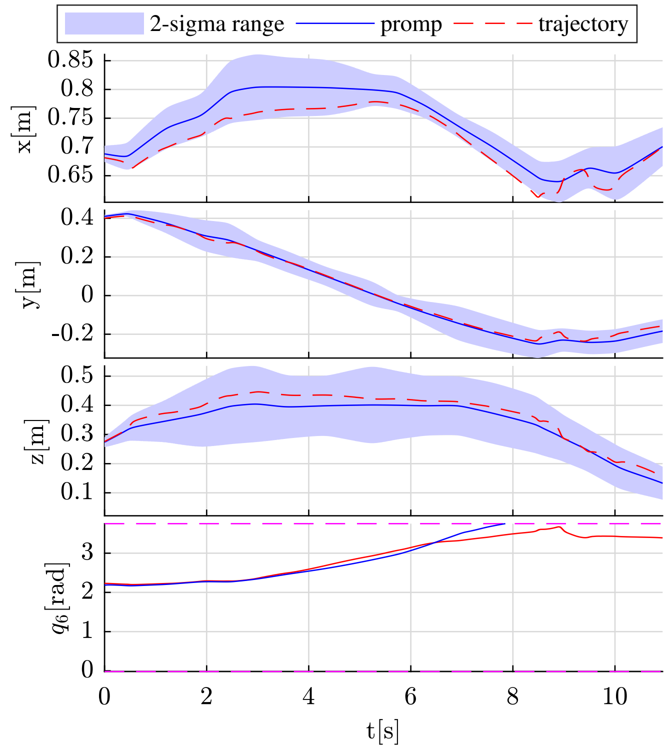

Fig. 5. Conditioned ProMP for the deflection experiment. The deflected trajectory exploits the 2-sigma region improving manipulability and joint-limit avoidance through task-space guided deflections as in (11).

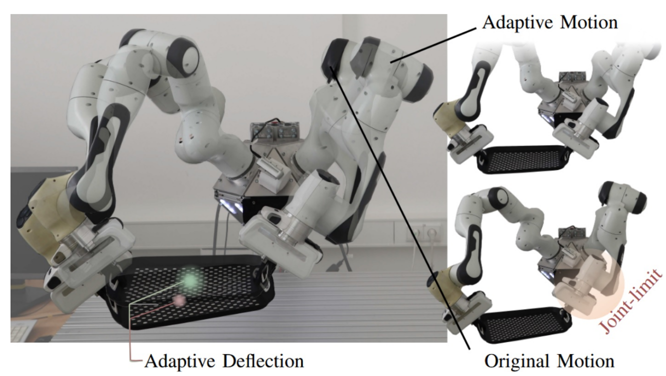

Fig. 6. Overlay of undeflected and deflected trajectory for a difficult trajectory with absolute position depicted by red and green markers. The deflection allows to avoid the joint limit in the 6-th joint of the robot's left arm that is hit during the undeflected motion. This effect is highlighted in Fig. 5.

novel desired position $\boldsymbol{p}_\mathrm{d} := \boldsymbol{p}_\mathrm{n} + \boldsymbol{d}$. As the distance between two consecutive goals can vary due to the deflection, in order to ensure a smooth movement, the instantaneous control goal $\boldsymbol{p}_\mathrm{i}$ is interpolated according to the actual velocity v_act

$$\boldsymbol{p}_\mathrm{i} := \boldsymbol{p}_\mathrm{i} + v_\mathrm{act} T \frac{\boldsymbol{p}_d - \boldsymbol{p}_\mathrm{i}}{|\boldsymbol{p}_d - \boldsymbol{p}_\mathrm{i}|_2}, \tag{19}$$

and $\boldsymbol{p}_d$ is updated whenever $|\boldsymbol{p}_d - \boldsymbol{p}_\mathrm{i}|_2 / T < v_{act}$. If the actual velocity is below the velocity for the current segment it is increased in each time step by a constant acceleration until v_ts is reached to ensure a smooth transition to higher velocities.

In order to reach the instantaneous control goal $\boldsymbol{p}_\mathrm{i}(t)$, a feed-forward joint velocity

$$\dot{\boldsymbol{q}}_\mathrm{ff} := v_\mathrm{act} \dot{\tilde{\boldsymbol{q}}}_\mathrm{ff} \frac{1}{|\dot{\tilde{\boldsymbol{q}}}_\mathrm{ff}|_2}, \tag{20}$$

$$\text{with} \quad \dot{\tilde{\boldsymbol{q}}}_\mathrm{ff} := \left(\boldsymbol{J}_{\underline{\boldsymbol{x}}_r} \boldsymbol{J}_{\underline{\boldsymbol{x}}_a}\right)^{\#} \left(\boldsymbol{0}^\mathrm{T} \quad \boldsymbol{p}_\mathrm{i}^\mathrm{T}(t+T) - \boldsymbol{p}_\mathrm{i}^\mathrm{T}(t)\right)^\mathrm{T}$$

and a control velocity

$$\dot{\boldsymbol{q}}_\mathrm{c} := k_\mathrm{pos} \dot{\tilde{\boldsymbol{q}}}_\mathrm{c} \frac{1}{T|\dot{\tilde{\boldsymbol{q}}}_\mathrm{c}|_2}, \tag{21}$$

$$\text{where} \quad \dot{\tilde{\boldsymbol{q}}}_\mathrm{c} := \boldsymbol{J}^{\#}_{\underline{\boldsymbol{x}}_a} (\boldsymbol{p}_\mathrm{i} - \boldsymbol{p})$$

are applied. The # symbol defines the robust pseudo-inverse, $\boldsymbol{J}_{\underline{\boldsymbol{x}}_r}$ and $\boldsymbol{J}_{\underline{\boldsymbol{x}}_a}$ are the relative pose and the absolute position Jacobians and $\boldsymbol{p}$ is the current position of the robot according to the commands, i.e. assuming perfect low-level control. The parameter $0 \leq k_\mathrm{pos} \leq 1$ is the position control gain. Due to the use of $\boldsymbol{J}_{\underline{\boldsymbol{x}}_r}$ in (20), the feed forward velocity respects the relative orientation of the robot and is thus not reduced when being projected into the nullspace of the relative pose controller. Fig. 2 depicts the proposed integrated strategy, summarizing the contributions within this section.

III. Experimental Evaluation

This section explores experimental aspects concerning the validation and performance of our framework on a dual-arm system, Kobo, that features two slightly modified Franka Emika Panda arms with 7-DoF each. For all experiments, demonstrations[7] were performed at the Kobo setup at Darmstadt through Kinesthetic teaching whereas execution was performed at Franka Emika headquarters in Munich.

We evaluated our framework on a cooperative task of transporting a tray with a coffee mug – building a set-based constraint for the orientation along the z-axis, i.e. the tilt angle in Table I – and constrained motion given by the coupled kinematics. In this scenario, while five demonstrated trajectories were always free of collision and trained under similar conditions, they were tested and executed in completely different settings. More specifically, we evaluate our framework in a constrained setup where the workspace is limited and the system prone to singularities and joint-limits—situations typical in dual-arm systems—in a scenario with unforeseen obstacles along the resulting trajectory and lastly, in fully different grasp-object context which needs shaping towards the original trajectory. For the sake of repeatability, for all experiments, we used the same non-optimized values of $v_\mathrm{max} = 0.1$ m/s as maximum velocity, used $q_r = 0.25$, $\alpha = 0.5$, $\epsilon_\sigma = 0.25$, $k_\mathrm{lim} = 1$, $k_\mathrm{man} = 1$, $d_\mathrm{min} = 0.2$, $d_\mathrm{max} = 0.4$ and the gains 2, -5, 0.5, 0.25 for the relative pose, end-effector tilt, ProMP following and joint limit avoidance control, respectively if not stated differently.

[7]For learning the ProMPs, we use a 20 Radial Basis Functions (RBFs) equally spaced in the $[0, 1]$ interval, where the bandwidth was chosen with cross-validation to minimize the mean-squared error between the mean trajectory and the demonstrations.

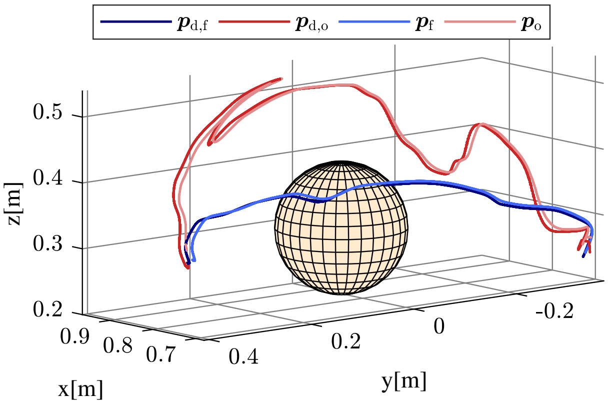

Fig. 7. Results of the obstacle evasion experiment. The depicted curves are the desired trajectories for the free ($\boldsymbol{p}_\mathrm{d,f}$) and obstructed ($\boldsymbol{p}_\mathrm{d,o}$) path as well as the real robot positions tracking these trajectories ($\boldsymbol{p}_\mathrm{f}$,$\boldsymbol{p}_\mathrm{o}$).

A. Adaptive Deflection

Firstly, we tested the capabilities of the adaptive deflection using a trajectory that was coincidentally taught with a bad initial joint configuration. Thus, without adapting the trajectory according to manipulability and joint limit avoidance, the controller was not able to follow the desired trajectory from the ProMPs after conditioning. On the other hand, through task modification ($k_\mathrm{lim} = k_\mathrm{man} = 1$) along the confidence interval of the demonstrated trajectories, the robot is able to adapt its motion and deflect the task towards a region of better manipulability and that it is not trapped in a joint-limit. We used $\alpha = 0.25$, $k_\mathrm{lim} = 5$ for the successful case and $k_\mathrm{lim} = k_\mathrm{man} = 0$ for the unsuccessful case. Fig. 6 highlights the point at which the controller fails during the execution without using the deflection—notice the joint limit at the second last joint of the robot's left arm—and an overlay of the position of the robot following the deflected trajectory for the same point. Fig. 5 depicts the resulting trajectory and the confidence interval in terms of the covariance of the demonstrated trajectory (2σ interval). As expected, the deflection explores the robot's topology changing the task trajectory, within the bounds from the demonstration, such that the joints remain further from the joint limits and retain a higher manipulability.

B. Obstacle Avoidance

In a second experiment, we evaluated our obstacle evasion using a trajectory that was trained with five demonstrations for taking a tray on the left side of the robot and putting it down on the right side. Fig. 7 shows the resulting trajectories, both for the nominal trajectory which would collide with the obstacle and the reactively deflected one. As shown in Fig. 7, even in such constrained cooperative manipulation, our deflection algorithm is able to generate deflected motions while keeping a minimum distance of 11.65 cm from the obstacle.

C. Generalization and Transferring

Finally, we explore our framework capability to generalize and transfer the encoded information from the demonstration to different grasping strategies while allowing the robot to adapt to the changed topology to improve the resulting motion. For the second grasp to execute the trajectory successfully, we needed to increase the gain of the joint limit avoidance control

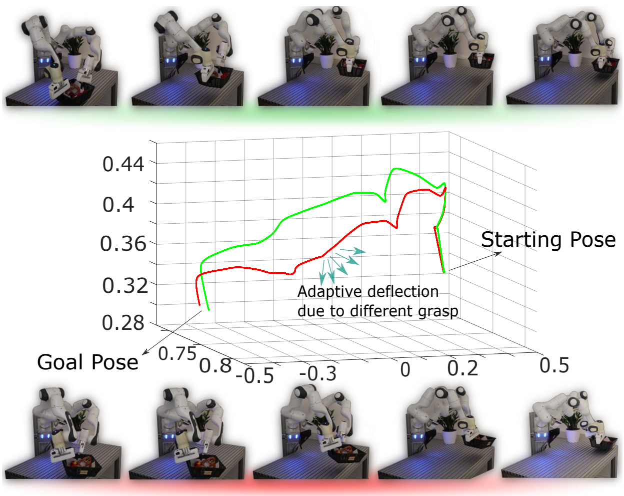

Fig. 8. Cooperative trajectory executed with different grasping poses (green denotes grasp 1, red denotes grasp 2 from Fig. 3) from an object-centric perspective view. Notice that the difference in trajectory results from different deflection strategies due to different topologies.

to 1. Fig. 8 shows the trajectories from two different grasping configurations with the same demonstrated information. The trajectories are expected to be similar but not identical due to the different adaptive deflections. Notice that the change in grasping will pose different constraints and topology for the tightly coupled kinematics. Hence, different adaptive executions are expected. Additionally, for sake of simplicity and to explore robustness of the approach, we perform transformation onto the trajectory instead of the object-frame which leads to additional differences during execution. Still, the robot was able to execute the trajectory and reach the goal with both grasps without tuning of parameters. Both trajectories are also within the confidence interval of the resulting ProMP distribution, as expected.

IV. Conclusion and Future Work

The trajectory generation from object-centric ProMPs presented in this work allows for flexible and robust bi-manual coupled motion generation and enables real-time reaction to obstacles in the workspace and cooperative task-space adaption along the confidence interval of the demonstrations' distribution. The proposed approach facilitates 1) adaptation of learned cooperative-task-space-centric motions to varying goal positions based on ProMP conditioning whilst adapting the trajectory execution at runtime to ensure sufficient manipulability and joint-limit avoidance through task-space deflections within the limits of the variance information from the user demonstrations, 2) reactive obstacle avoidance while satisfying the automatically extracted geometric task constraints and 3) adaptation of the learned motion to changed grasp configurations, which would not be feasible with existing decoupled and joint-space. In future work, we plan to encode and integrate additional human manipulability information from bimanual demonstrations to our framework by means of CoSTP nullspace optimization, and extend the demonstration strategies to include in-the-air regrasping within the cooperative dual-task space framework.

References

[1] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in Neural Information Processing Systems*, vol. 26, 2013.
[2] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2007.
[3] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.
[4] 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.
[5] A. Gams, B. Nemec, L. Zlajpah, M. Wachter, A. Ijspeert, T. Asfour, and A. Ude, "Modulation of motor primitives using force feedback: Interaction with the environment and bimanual tasks," in *IROS*, 2013.
[6] R. Lioutikov, O. Kroemer, G. Maeda, and J. Peters, "Learning manipulation by sequencing motor primitives with a Two-Armed robot," *Advances in Intelligent Systems and Computing*, 2016.
[7] N. Likar, B. Nemec, L. Žlajpah, S. Ando, and A. Ude, "Adaptation of bimanual assembly tasks using iterative learning framework," *Humanoids*, 2015.
[8] A. Gams, B. Nemec, A. J. Ijspeert, and A. Ude, "Coupling movement primitives: Interaction with the environment and bimanual tasks," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 816–830, 2014.
[9] J. Umlauft, D. Sieber, and S. Hirche, "Dynamic movement primitives for cooperative manipulation and synchronized motions," in *ICRA*, 2014.
[10] A. Batinica, B. Nemec, A. Ude, M. Raković, and A. Gams, "Compliant movement primitives in a bimanual setting," in *Humanoids*, 2017.
[11] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intelligent service robotics*, vol. 9, no. 1, pp. 1–29, 2016.
[12] J. Silvério, S. Calinon, L. Rozo, and D. G. Caldwell, "Learning task priorities from demonstrations," *IEEE Transactions on Robotics*, 2019.
[13] L. Singh, H. Stephanou, and J. Wen, "Real-time robot motion control with circulatory fields," in *ICRA*, 1996.
[14] A. Ataka, H.-K. Lam, and K. Althoefer, "Reactive magnetic-field-inspired navigation method for robots in unknown convex 3d environments," *IEEE Robotics and Automation Letters*, 2018.
[15] S. Haddadin, R. Belder, and A. Albu-Schäffer, "Dynamic motion planning for robots in partially unknown environments," *IFAC*, 2011.
[16] D. Sun and J. K. Mills, "Adaptive synchronized control for coordination of multirobot assembly tasks," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 498–510, 2002.
[17] C. K. Verginis, M. Mastellaro, and D. V. Dimarogonas, "Robust cooperative manipulation without force/torque measurements: Control design and experiments," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 3, pp. 713–729, 2019.
[18] B. V. Adorno, P. Fraisse, and S. Druon, "Dual position control strategies using the cooperative dual task-space framework," in *IROS*, 2010.
[19] R. Laha, L. F. Figueredo, J. Vrabel, A. Swikir, and S. Haddadin, "Reactive Cooperative Manipulation based on Set Primitives and Circular Fields," in *ICRA*, Xi'an, China, May 2021.
[20] L. Figueredo, B. V. Adorno, J. Y. Ishihara, and G. Borges, "Switching strategy for flexible task execution using the cooperative dual task-space framework," in *IROS*. IEEE, 2014.
[21] R. Laha, J. Vorndamme, L. F. Figueredo, Z. Qu, A. Swikir, C. Jähne, and S. Haddadin, "Coordinated motion generation and object placement: A reactive planning and landing approach," in *IROS*, 2021.
[22] B. V. Adorno, "Two-arm manipulation: From manipulators to enhanced human-robot collaboration," Ph.D. dissertation, Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM) - Université Montpellier 2, Montpellier, France, 2011.
[23] L. F. C. Figueredo, "Kinematic control based on dual quaternion algebra and its application to robot manipulators," Ph.D. dissertation, University of Brasilia, Brazil, 2016.
[24] Y. Huang, L. Rozo, J. Silvério, and D. Caldwell, "Kernelized movement primitives," *The International Journal of Robotics Research*, 2019.
[25] J. Urain, M. Ginesi, D. Tateo, and J. Peters, "Imitationflow: Learning deep stable stochastic dynamic systems by normalizing flows," in *IROS*, 2020.
[26] S. Gomez-Gonzalez, G. Neumann, B. Schölkopf, and J. Peters, "Adaptation and robust learning of probabilistic movement primitives," *IEEE Transactions on Robotics*, vol. 36, no. 2, pp. 366–379, Mar. 2020.
[27] J. Funda and R. Paul, "A computational analysis of screw transformations in robotics," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 3, pp. 348–356, 1990.
[28] E. Özgür and Y. Mezouar, "Kinematic modeling and control of a robot arm using unit dual quaternions," *Robotics and Autonomous Systems*, 2016.
[29] J. Kuipers, *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*. Princeton University Press, 1999.
[30] J. M. Selig, *Geometric Fundamentals of Robotics*, 2nd ed. Springer-Verlag New York Inc., 2005.
[31] B. V. Adorno, "Robot Kinematic Modeling and Control Based on Dual Quaternion Algebra – Part I: Fundamentals - hal-01478225," 2017.
[32] S. Moe, G. Antonelli, A. R. Teel, K. Y. Pettersen, and J. Schrimpf, "Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: General formulation, stability analysis, and experimental results," *Frontiers Robotics AI*, 2016.
[33] J. Silvério, Y. Huang, F. J. Abu-Dakka, L. Rozo, and D. G. Caldwell, "Uncertainty-aware imitation learning using kernelized movement primitives," in *IROS*, 2019.
[34] C. Qiu, Q. Cao, and S. Miao, "An on-line task modification method for singularity avoidance of robot manipulators," *Robotica*, 2009.