Constrained DMPs for Feasible Skill Learning on Humanoid Robots

Anqing Duan[1], Raffaello Camoriano[2], Diego Ferigo[1],
Daniele Calandriello[2], Lorenzo Rosasco[2,3], Daniele Pucci[1]

***Abstract*— In the context of humanoid skill learning, movement primitives have gained much attention because of their compact representation and convenient combination with a myriad of optimization approaches. Among them, a well-known scheme is to use Dynamic Movement Primitives (DMPs) with reinforcement learning (RL) algorithms. While various remarkable results have been reported, skill learning with physical constraints has not been sufficiently investigated. For example, when RL is employed to optimize the robot joint trajectories, the exploration noise could drive the resulting trajectory out of the joint limits. In this paper, we focus on robot skill learning characterized by joint limit avoidance, by introducing the novel Constrained Dynamic Movement Primitives (CDMPs). By controlling a set of transformed states (called exogenous states) instead of the original DMPs states, CDMPs are capable of maintaining the joint trajectories within the safety limits. We validate CDMPs on the humanoid robot iCub, showing the applicability of our approach.**

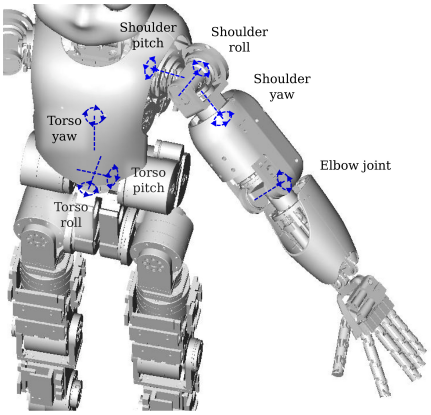


Fig. 1: Illustration of the iCub joints used in our experiments.

I. INTRODUCTION

In recent years, movement primitives as a trajectory parametrization technique remain one of the most important research topics in the area of robot skill learning. Various types of movement primitives have been developed and their development is still in progress actively [1]. With the help of movement primitives, robot skills can be encoded robustly and compactly by organizing the building blocks in parallel and/or in series. The required time and costs can usually be reduced by programing the robot from human demonstration. Actually, numerous state-of-the-art robot learning success cases rely on the usage of movement primitives. For example, in the context of imitation learning, robots can learn hard-to-engineer skills such as hitting a ball by extracting the relevant motion patterns from human demonstrations [2]. In addition, to provide robots with a new level of autonomy and flexibility, movement primitives are preferably employed in reinforcement learning (RL) algorithms since they can considerably alleviate the problem of the curse of dimensionality. Especially, the combination of Dynamic Movement Primitives (DMPs) [3] with RL algorithms has already endowed robots with various sophisticated skills such as table tennis [4], ball-in-the-cup [5], and dart throwing [6].

Despite of the remarkable achievements accomplished by optimizing the DMPs parameters with RL algorithms, skill learning with physical constraints has not been sufficiently investigated yet. This problem shall not be overlooked since multiple causes can give rise to joint limit violation. For example, when applying RL-based optimization approach to DMPs, random exploratory noise is added to the DMPs parameters in search of the optimal policy. Therefore, the generated trajectory could possibly induce unexpected overshoots or unbounded final goal position as a result of shape and goal learning. Similar concern also exists when the robot's trajectory needs to be adjusted in order to meet the additional requirements arising over the course of some task, e.g. obstacle avoidance. The typical treatment of obstacle avoidance for DMPs, in fact, is to add the coupling terms derived from the properly tuned potential field [7]. If the potential field has unduly strong intensity, the DMPs states would be driven out of the allowable limit range and thus provide the robot with unfeasible reference trajectories.

In general, joint limit avoidance is a widely studied topic in robotics. Several methods have been proposed in path planning, including damped least-squares [8], weighted least-norm solutions [9], and Lyapunov-based methods [10], among others. A limitation of path planning approaches lies in the fact that the generated safe trajectories are the desired ones, which are not guaranteed to be accurately tracked by the system. In the case of redundant manipulators, stack-of-tasks methods can be applied by introducing a lower-priority objective enforcing joint limits avoidance in the null space of higher-level tasks [11]. However, the resulting joint trajectories are not guaranteed to be safe. Gradient projection techniques have also been proposed, whose working principle is to define a function maximizing joint margin and project its gradient onto the null space projection matrix

[1] iCub Facility Department, Istituto Italiano di Tecnologia, Via Morego 30, Genoa, Italy. Email: firstname.lastname@iit.it
[2] Laboratory for Computational and Statistical Learning (IIT@MIT), Istituto Italiano di Tecnologia and Massachusetts Institute of Technology, Cambridge, MA, USA. Email: firstname.lastname@iit.it
[3] DIBRIS, Università degli Studi di Genova, Genoa, Italy.

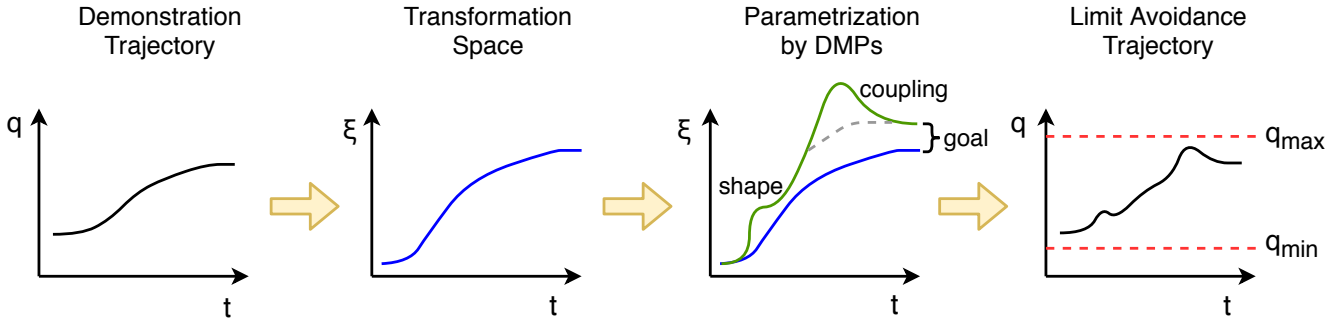


Fig. 2: Illustration of the principle of CDMPs - a joint limit avoidance scheme for DMPs. The joint space demonstration trajectory q is first transformed into the exogenous state ξ. Subsequently, the exploration noises for both shape and goal learning will be added to ξ instead. Also, potential filed based obstacle avoidance strategy will modify the dynamics of the trajectory by adding the coupling term in the transformed space. Finally, the resulting joint trajectory with guaranteed joint limit avoidance property will be obtained using the inverse transformation.

of the Jacobian [12]. This allows to drive the joints away from limits without moving the end effector, but it does not guarantee successful minimization for each joint and requires tuning additional parameters. Other approaches, introduced in whole-body humanoid motion control, are: 1) to add virtual springs or spring-dampers near joint limits [13], and 2) to introduce an inequality constraint for each joint range in the optimization problem associated with whole-body motion [14]. Both have been shown to work in practice, but lack stability and convergence guarantees. Moreover, it is not always possible to apply these methods to directly bound the evolution range of DMP-generated trajectories.

To address the aforementioned issues, we propose the novel Constrained Dynamic Movement Primitives (CDMPs). CDMPs are inspired by the work of [15], which introduced a controller with joint-limit avoidance for torque-controlled robots, based on mapping joint states to a suitable exogenous state space. This control strategy allows for convergence and asymptotic stabilization of a reference joint trajectory, and ensures that the actual joint trajectories belong to the safe ranges. Our core idea is to transform the feasible demonstration trajectory states in terms of exogenous states. By parameterizing the exogenous states instead, DMPs states are ensured to evolve within the joint limits as long as the initial policy respects the joint limits restriction.

This paper is organized as follows. Section II provides the background on robot control, obstacle avoidance and classical DMPs. Subsequently, in Section III we motivate the derivation of CDMPs and the related application settings. The performance of CDMPs is illustrated in Section IV with simulation experiments on the humanoid robot iCub [16]. Finally, Section V concludes our results and discusses potential future extensions. An overview of the proposed method is shown in Fig. 2.

II. Background

A. Robot Modeling and Control

Consider a fixed base and open chain robot with n degrees of freedom (DoF). The robot dynamics equation can be derived from the Lagrange formalism [17]:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{B}\boldsymbol{\tau} + \sum_{k=1}^{N_c} \mathbf{J}_{C_k}^{\top}\mathbf{f}_k, \qquad (1)$$

where $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^n$ are generalized positions, velocities and accelerations of the robot, respectively, $\mathbf{M} \in \mathbb{R}^{n\times n}$ is the inertia matrix, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) \in \mathbb{R}^{n\times n}$ accounts for Coriolis and centrifugal effects, $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^n$ is the gravity term, $\mathbf{B}$ is a selector matrix, $\boldsymbol{\tau} \in \mathbb{R}^n$ is a vector representing the actuation joint torques, $\mathbf{f}_k \in \mathbb{R}^6$ denotes the k-th external wrench applied by the environment on the robot, and $\mathbf{J}_{C_k}$ is the corresponding Jacobian. The term $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}+\mathbf{G}(\mathbf{q})$ together is also called *bias forces*.

A classical control strategy to calculate the desired joint torques $\boldsymbol{\tau}_d$ given the desired joint trajectory $\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d$ is called the *computed torque* control law:

$$\begin{aligned}\boldsymbol{\tau}_d = \mathbf{M}(\mathbf{q})\big(\ddot{\mathbf{q}}_d - \mathbf{K}_P(\mathbf{q}-\mathbf{q}_d) - \mathbf{K_D}(\dot{\mathbf{q}}-\dot{\mathbf{q}}_d)\big) \\ + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}), \qquad (2)\end{aligned}$$

where $\mathbf{K}_P$ (joint stiffness) and $\mathbf{K}_D$ are feedback matrices. This control law comprises both feedforward and feedback terms. By formulating the control law this way, the robot can have more compliance against the external interactions and its stability analysis is well understood [18].

It is relevant to note that applying the classic computed torque control law could sometimes cause the issue of joint limit violation from the overshoots or external disturbances. To solve this problem, a novel control law was previously proposed which incorporates the property of joint limit avoidance [15]. The novelty of the proposed control law lies in the fact that the evolution of the joint trajectory is always guaranteed to remain within the associated physical bounds $\mathbf{q}_{\text{min}}$ and $\mathbf{q}_{\text{max}}$ by parameterizing the feasible joint space in terms of the exogenous variable $\boldsymbol{\xi}$:

$$\mathbf{q}(\boldsymbol{\xi}) = \boldsymbol{\delta}\tanh(\boldsymbol{\xi}) + \mathbf{q}_o, \qquad (3)$$

where $\mathbf{q}_o = \frac{1}{2}(\mathbf{q}_{\text{min}} + \mathbf{q}_{\text{max}})$, $\boldsymbol{\delta} = \text{diag}\big(\frac{1}{2}(\mathbf{q}_{\text{max}} - \mathbf{q}_{\text{min}})\big)$, and $\tanh(\cdot)$ is a hyperbolic tangent function. The proposed

control law in the case of set points is given by:

$$\boldsymbol{\tau} = -\mathbf{K}_P\tilde{\boldsymbol{\xi}} - \mathbf{K}_D\dot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}), \tag{4}$$

where $\tilde{\boldsymbol{\xi}} = \boldsymbol{\xi} - \boldsymbol{\xi}_d$ is the tracking error and the desired trajectory $\boldsymbol{\xi}_d$ can be obtained using the inverse transformation of Eq. (3):

$$\boldsymbol{\xi}_d = \operatorname{arctanh}\big(\boldsymbol{\delta}^{-1}(\mathbf{q}_d - \mathbf{q}_o)\big). \tag{5}$$

For further details, the reader is referred to [15].

B. Obstacle Avoidance Scheme

Since the robot is directly controlled at joint level while the obstacle position is usually expressed in Cartesian space, it is therefore inevitable to introduce Cartesian space to joint space mappings. The relationship between joint space parameters $\mathbf{q} \in \mathbb{R}^n$ and Cartesian space parameters $\mathbf{x} \in \mathbb{R}^m$ at the acceleration level can be expressed as

$$\ddot{\mathbf{q}} = \mathbf{J}^\dagger(\ddot{\mathbf{x}} - \dot{\mathbf{J}}\dot{\mathbf{q}}) + (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})\mathbf{N}, \tag{6}$$

where $\mathbf{J}$ is a Jacobian and $\mathbf{J}^\dagger = \mathbf{J}^\top(\mathbf{J}\mathbf{J}^\top)^{-1}$ is its Moore-Penrose inverse providing least squares solutions; $\mathbf{I}$ represents the identity matrix of proper dimensionality and $\mathbf{N}$ corresponds to a joint space movement in the null space. From Eq. (6), the end effector movement modification $\Delta\ddot{\mathbf{x}}_e$ for obstacle avoidance can be easily obtained as

$$\Delta\ddot{\mathbf{q}}_e = \mathbf{J}^\dagger\Delta\ddot{\mathbf{x}}_e. \tag{7}$$

For link obstacle avoidance, null space movement will be exploited. Consider the closest point to the obstacle on the link is $\mathbf{x}_o$, with $\mathbf{J}_o$ the corresponding Jacobian. From inverse kinematics, we can have:

$$\ddot{\mathbf{x}}_o = \dot{\mathbf{J}}_o\dot{\mathbf{q}} + \mathbf{J}_o\ddot{\mathbf{q}}. \tag{8}$$

By combining Eq. (8) and Eq. (6) to drop $\ddot{\mathbf{q}}$ out, we can solve for $\mathbf{N}$:

$$\mathbf{N} = \big(\mathbf{J}_o(\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})\big)^\dagger\big(\ddot{\mathbf{x}}_o - \dot{\mathbf{J}}_o\dot{\mathbf{q}} - \mathbf{J}_o\mathbf{J}^\dagger(\ddot{\mathbf{x}} - \dot{\mathbf{J}}\dot{\mathbf{q}})\big). \tag{9}$$

Then, by substituting the expression for $\mathbf{N}$ back to Eq. (6), we obtain:

$$\begin{aligned}\ddot{\mathbf{q}} = \mathbf{J}^\dagger(\ddot{\mathbf{x}} - \dot{\mathbf{J}}\dot{\mathbf{q}}) + (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})\big(\mathbf{J}_o(\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})\big)^\dagger \\ \big(\ddot{\mathbf{x}}_o - \dot{\mathbf{J}}_o\dot{\mathbf{q}} - \mathbf{J}_o\mathbf{J}^\dagger(\ddot{\mathbf{x}} - \dot{\mathbf{J}}\dot{\mathbf{q}})\big).\end{aligned} \tag{10}$$

Finally, the link movement modification $\Delta\ddot{\mathbf{x}}_l$ for obstacle avoidance at joint acceleration level is given by:

$$\Delta\ddot{\mathbf{q}}_l = (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})\big(\mathbf{J}_o(\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})\big)^\dagger\Delta\ddot{\mathbf{x}}_l. \tag{11}$$

It can be observed that $\mathbf{I} - \mathbf{J}^\dagger\mathbf{J}$ is a symmetric idempotent matrix (also called projection matrix), so the results can be simplified as follows:

$$\Delta\ddot{\mathbf{q}}_l = \big(\mathbf{J}_o(\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})\big)^\dagger\Delta\ddot{\mathbf{x}}_l. \tag{12}$$

To conclude here briefly, Eq. (7) and Eq. (12) are used for the design of the CDMPs coupling terms with $\Delta\ddot{\mathbf{x}}_e$ and $\Delta\ddot{\mathbf{x}}_l$ calculated from the corresponding potential field, respectively. It will be shown later that compared with DMPs, CDMPs have the capability of handling improperly designed potential field.

C. Dynamic Movement Primitives

Here, we briefly introduce DMPs. DMPs are a flexible representation for motion primitives. It can be used to generate either discrete or rhythmic movements. In this paper, only discrete movements are considered.

Consisting of a set of linear differential equations, DMPs can be interpreted as a damped spring model with the following transformation system:

$$\tau\ddot{y} = \alpha(\beta(g - y) - \dot{y}) + f + P(y,\dot{y}), \tag{13}$$

$$\tau\dot{g} = \alpha_g(g_o - g), \tag{14}$$

where y, $\dot{y}$, g are system states; τ, α, β and α_g are positive constants; g_o is the final goal; P is called *coupling term* and its form is dependent on the choice of the potential field. Although the goal parameter is usually a constant, here we formulate it in a differential equation as Eq. (14) to allow for goal learning, as described later. Theoretically, we can obtain arbitrary continuous shapes by disturbing the dynamical system using the nonlinear forcing term f, defined as:

$$f = \mathbf{g}_t^T\boldsymbol{\Theta}, \tag{15}$$

$$[\mathbf{g}_t]_j = \frac{\Psi_j(s_t)\cdot s_t}{\sum_{k=1}^{p}\Psi_k(s_t)}(g_t - y_0), \tag{16}$$

$$\Psi_j = \exp(-0.5h_j(s_t - c_j)^2), \tag{17}$$

where $\mathbf{g}_t$ are called basis functions, composed by the initial position y_0 and the Gaussian kernel Ψ_j with parameters $h_j > 0$ and $c_j \in [0,1]$. $\boldsymbol{\Theta}$ is the vector of basis function weights or shape vector since it decides the general shape of the trajectory. Usually, it is $\boldsymbol{\Theta}$ that represents a specific policy in RL algorithms. Given a trajectory from demonstration, $\boldsymbol{\Theta}$ can be efficiently calculated by weighted linear regression. Phase variable s_t is calculated from a canonical system that drives the whole dynamic system:

$$\tau\dot{s}_t = -\alpha s_t. \tag{18}$$

Since s_t moves from one to zero, convergence to the goal g_o is guaranteed as f vanishes at the end of the movement. It should be noted that although each DoF typically requires one transformation system, only one single canonical system is enough to coordinate all DoFs.

III. Constrained Dynamic Movement Primitives

A. Constrained Dynamic Movement Primitives

As mentioned previously, the derivation of CDMPs is motivated by the novel joint limit avoidance control law introduced in [15]. The demonstrated trajectory state y is parameterized by the CDMPs state ξ in a similar way to Eq. (3), i.e.,

$$y(\xi) = y_\delta\tanh(\xi) + y_o. \tag{19}$$

As before, $y_\delta = \frac{1}{2}(y_{\min} + y_{\max})$ and $y_o = \operatorname{diag}\big(\frac{1}{2}(y_{\max} - y_{\min})\big)$ with $y_{\min}$ and $y_{\max}$ the specified trajectory limits. ξ can be simply obtained by using the inverse transformation:

$$\xi = \operatorname{arctanh}\big(y_\delta^{-1}(y - y_o)\big). \tag{20}$$

The evolution rule of the CDMPs state follows that of the DMPs with the transformation system given by:

$$\tau_\xi \ddot{\xi} = \alpha_\xi\big(\beta_\xi(g_\xi - \xi) - \dot{\xi}\big) + f_\xi + P_\xi(\xi, \dot{\xi}), \tag{21}$$

$$\tau_\xi \dot{g_\xi} = \alpha_{g_\xi}(g_{\xi_o} - g_\xi), \tag{22}$$

where τ_ξ, α_ξ, β_ξ and α_{g_ξ} are positive constants; g_{ξ_o} is the final goal; P_ξ is the coupling term of CDMPs. The nonlinear forcing term f_ξ is similarly defined as:

$$f_\xi = \mathbf{g}_t^\top \mathbf{\Theta}_\xi, \tag{23}$$

$$[\mathbf{g}_t]_j = \frac{\Psi_j(s_t) \cdot s_t}{\sum_{k=1}^{p} \Psi_k(s_t)}(g_{\xi_t} - \xi_o), \tag{24}$$

$$\Psi_j = \exp(-0.5 h_j (s_t - c_j)^2), \tag{25}$$

where $\xi_o = \operatorname{arctanh}\big(y_\delta^{-1}(y_{init} - y_o)\big)$ is the initial position of the demonstrated trajectory starting from y_{init}. Last, we have the canonical system generating the phase variable s_t:

$$\tau_\xi \dot{s}_t = -\alpha_\xi s_t. \tag{26}$$

Once the evolution of the exogenous variable ξ is obtained, the final reference joint trajectory y_{ref} can be calculated based on Eq. (19):

$$y_{\text{ref}}(\xi) = y_\delta \tanh(\xi) + y_o. \tag{27}$$

Furthermore, the velocity and acceleration of the reference trajectory is given by

$$\dot{y}_{\text{ref}}(\xi) = J(\xi)\dot{\xi}, \tag{28}$$

$$\ddot{y}_{\text{ref}}(\xi) = J(\xi)\ddot{\xi} + \dot{J}(\xi, \dot{\xi})\dot{\xi}, \tag{29}$$

where $J(\xi) = y_\delta(1 - \tanh^2(\xi))$.

As proven in [15] that the resulting joint reference trajectory is guaranteed to be bounded within the joint limit by exploiting the property of the hyperbolic function.

B. Coupling Term Design for CDMPs

For a given CDMP, sometimes it is desirable to modulate the trajectory online so that it can be flexible to different situations. As in the case of DMPs, introducing coupling terms in Eq. (21) is the typical approach to account for such complex behaviors. To realize obstacle avoidance, for instance, a spatial coupling term can be used to change the dynamics by modifying the acceleration term of CDMPs as in Eq. (21). The design of the coupling term is dependent on the form of the repelling potential field emitted by the obstacles. Taking a CDMP trajectory escaping from a board as an example, a piecewise potential field can be designed as follows. When CDMP trajectory is under the board, the coupling term is designed as:

$$P_{below} = \begin{cases} \beta_y \frac{\Delta y}{\Delta z} & \text{horizontally,} \\ \beta_z \frac{1}{\Delta z} & \text{vertically,} \end{cases} \tag{30}$$

where Δy and Δz denote the distances to the vertical edge of the board and the horizontal surface of the board, respectively and β_y and β_z are two constants. The potential field is designed in such way that CDMPs trajectory can obtain infinity high escape driving force when approaching the board. When CDMPs trajectory moves out of the region framed by the vertical edge of the board and the horizontal surface of the board, the potential field is designed as

$$P_{above} = \alpha\sqrt{\Delta z}, \tag{31}$$

where α is a constant and Δz now denotes the vertical distance to the final goal.

By adding the coupling terms to the CDMPs states directly, the joint trajectory can be bounded in spite of the possibly unduly high potential field strengths.

C. Reinforcement Learning of CDMPs

The reinforcement learning algorithm of choice for CDMPs parameters optimization is called Policy Improvement with Path Integrals (PI^2), the derivation of which is rooted in the first principles of stochastic optimal control [19]. PI^2 is a probability weighted averaging method and thus no open algorithmic tuning parameter is needed. Allegedly, PI^2 can outperform the other existing direct RL algorithms by an order of magnitude in terms of the learning speed and the final cost of the learned policy as shown in [19].

In order for PI^2 to learn both shape parameters and goal parameters of CDMPs, at each time step t of the trial k, the exploration noises $\boldsymbol{\epsilon}_{t,k}^{\Theta_\xi}$ and $\epsilon_k^{g_\xi}$ drawn from the Gaussian distributions $\mathbf{\Sigma}_{\Theta_\xi}$ and Σ_{g_ξ} shall be added, respectively. As a result, Eq. (23) and Eq. (22) now become

$$f_{\xi t} = \mathbf{g}_t^\top(\mathbf{\Theta}_\xi + \boldsymbol{\epsilon}_{t,k}^{\Theta_\xi}), \tag{32}$$

$$\tau_\xi \dot{g}_\xi = \alpha_{g_\xi}(g_{\xi_o} + \epsilon_k^{g_\xi} - g_\xi). \tag{33}$$

Like other RL algorithms, PI^2 requires the cost-to-go $S(\tau_{i,k})$ for each trial τ, starting from t_i till the final time t_N. By formulating the cost in terms of the final cost, the immediate cost and the immediate control cost, we can have

$$S(\tau_{i,k}) = \phi_{t_{N,k}} + \sum_{j=1}^{N-1} r_{t_j,k} + \frac{1}{2}\sum_{j=i+1}^{N-1}(\mathbf{\Theta}_\xi + \mathbf{M}_{t_j,k}\boldsymbol{\epsilon}_{t_j,k}^{\Theta_\xi})^\top \mathbf{R}(\mathbf{\Theta}_\xi + \mathbf{M}_{t_j,k}\boldsymbol{\epsilon}_{t_j,k}^{\Theta_\xi}), \tag{34}$$

where $\mathbf{M}_{t_j,k} = \frac{\mathbf{R}^{-1}\mathbf{g}_{t_j}\mathbf{g}_{t_j}^\top}{\mathbf{g}_{t_j}^\top \mathbf{R}^{-1}\mathbf{g}_{t_j}}$ is the matrix needed to project the exploration noises onto the parameter space with $\mathbf{R}$ a weight factor and $\mathbf{g}$ calculated from Eq. (24). Once the trajectory cost is obtained, the update rule for the shape parameters $\mathbf{\Theta}_\xi$ is given by:

$$\delta\mathbf{\Theta}_{\xi t_i} = \sum_{k=1}^{K} P(\tau_{i,k})\mathbf{M}_{t_j,k}\boldsymbol{\epsilon}_{t_j,k}^{\Theta_\xi} \tag{35}$$

$$[\delta\mathbf{\Theta}_\xi]_j = \frac{\sum_{i=0}^{N-1}(N-i)\Psi_{j,t_i}[\delta\mathbf{\Theta}_{\xi t_i}]_j}{\sum_{i=0}^{N-1}\Psi_{j,t_i}(N-i)} \tag{36}$$

$$\mathbf{\Theta}_\xi \leftarrow \mathbf{\Theta}_\xi + \delta\mathbf{\Theta}_\xi \tag{37}$$

where $P(\tau_{i,k}) = \frac{e^{-\frac{1}{\lambda}S(\tau_{i,k})}}{\sum_{k=1}^{K} e^{-\frac{1}{\lambda}S(\tau_{i,k})}}$ is the probability of each trial. The intuition behind is that lower costs should have higher probabilities.

Goal parameter can be updated in a similar yet simpler way since only the probability at the starting time of the trial k is required:

$$\delta g_\xi = \sum_{k=1}^{K} P(\tau_{0,k}) \epsilon_k^{g_\xi}, \tag{38}$$

$$g_\xi \leftarrow g_\xi + \delta g_\xi. \tag{39}$$

Since there is no interference between learning g_ξ and $\boldsymbol{\Theta}_\xi$, these parameters can be updated simultaneously using exactly the same trial.

It should be noted that by adding the exploration noises to the CDMPs parameters directly, the generated exploratory joint trajectory after the inverse transformation will respect the joint limits as well.

IV. Evaluations

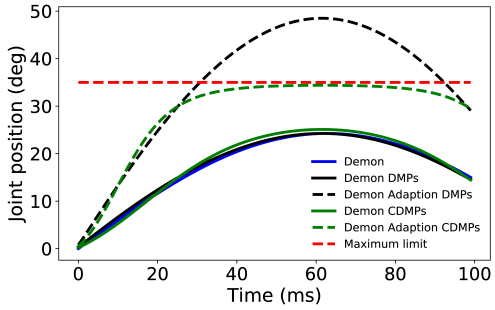
In this section, the experimental validation of the proposed method on the iCub humanoid robot is presented [20]. The iCub is a humanoid robot with a total of 53 DoFs. In our experiments, 7 of them are activated: 3 for the torso, 3 for the shoulder, and 1 for the elbow (see Fig. 1). The remaining joints are locked in the home position, and the base of the robot is fixed by setting a rigid constraint between the base link and the world frame.

We evaluate the proposed method by performing three experiments. The first one is a toy example for comparison between CDMPs and DMPs to illustrate the necessity of bounding DMPs; the second one is a reaching task with link obstacle avoidance; finally, the third one is a ball-dropping task with end-effector obstacle avoidance. All the experiments are conducted in the Pybullet simulation environment [21] with the external off-the-shelf kinematics and dynamics library iDynTree [22].

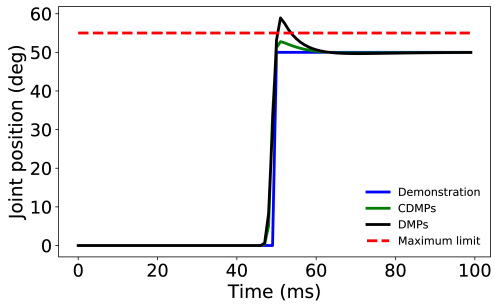
A. Toy Examples for Comparison of CDMPs and DMPs

1) Task description: In this task, the effectiveness of joint limit avoidance of CDMPs is demonstrated by comparing the evolution of the associated joint states with the ones of DMPs, whose parameters are the same ones used in [3]. Usually there are two cases that have high chance to hit the joint limits, namely goal adaption and improper acceleration. Goal adaption refers to modifying DMPs dynamics by changing the goal parameter only. Because a simple change of the goal state automatically creates a complex rescaling of the entire movement, the maximum value of the rescaled system is difficult to be bounded in DMPs. In addition, very large acceleration from the dramatic change of a trajectory could also contribute to joint limit violation as a result of overshoot. In this task, however, it will be shown that the allowable joint limit value can be easily satisfied by using CDMPs.

2) Experimental results and discussion: It can be seen in Fig. 3a that given an eligible demonstrated trajectory, both DMPs and CDMPs can accurately imitate a given trajectory. However, when adapting the final goal to a higher value, the obtained DMPs trajectory has a maximum value of 48.5 deg and it exceeds the maximum limit 35 deg. On the other hand, by using CDMPs, the range of the obtained trajectory is bounded all the time with the maximum value 34.4 deg. Moreover, when the imitated trajectory has a steep change, as shown in Fig. 3b, DMPs cannot be guaranteed to evolve within the specified limit (55 deg in the example). The maximum value due to the overshoot of DMPs is 58.9 deg, while for CDMPs it is bounded to 52.8 deg.



(a) Goal adaption



(b) Improper acceleration

Fig. 3: Toy examples for CDMPs and DMPs comparison.

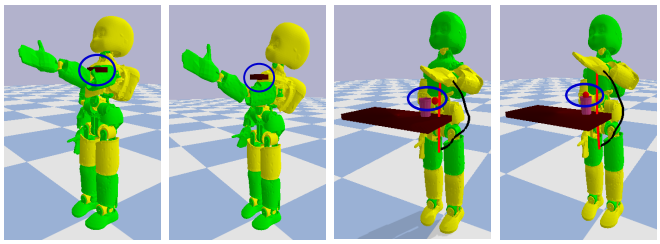
It can be concluded from the above toy examples that the proposed CDMPs can overcome the problem of joint limit violation, which is very common in the traditional DMPs.

B. Reaching Task with Link Obstacle Avoidance

1) Task description: In this task, the effects of CDMPs under the link obstacle avoidance scheme are evaluated. The robot's end effector is required to reach a desired goal positioned at $(0.350, 0.091, 0.853)$ m in Cartesian space. In the meantime, an obstacle board is placed at the final position of the upper arm link of the robot as shown in Fig. 4a. The distance between the board and the most dangerous point on the link is -1.37 cm, where a negative distance indicates overlap and therefore a collision between the link and the obstacle happens.

To prevent the dangerous link from hitting the obstacle, the redundancy of the robot will be exploited by using Eq. (12). The coupling term is designed in terms of a simple constant potential field.

2) Experimental results and discussion: By adding the coupling term and exploiting the redundancy of the robot, link collision avoidance and end effector reaching task can be satisfied simultaneously. As shown in Fig. 4b, robot



(a) (b) (c) (d)

Fig. 4: Screenshots of iCub executing different tasks using CDMPs. (a)-(b): CDMPs under the potential field for link obstacle avoidance; (c)-(d): ball dropping performance before and after learning the optimal CDMPs parameters with RL where the red line denotes the virtual trajectory of the end effector from forward kinematics neglecting the coupling terms and the black line denotes the real one.

redundancy is exploited, as it can be observed from the additional displacements of the torso and elbow joints. The resulting end effector final position is measured as $(0.354, 0.083, 0.848)$ m, which is slightly different from the requested value. The dangerous upper arm link now has a distance of 1.03 cm from the obstacle board and thus link collision is avoided. CDMPs are capable of realizing obstacle avoidance with the help of the coupling terms. It should be noted that during the task, when choosing very high strength for the potential field, CDMPs still ensure the trajectory is bounded, while DMPs cannot.

C. Ball-dropping Task with End Effector Obstacle Avoidance

1) Task description: The aim of this task is mainly to show that CDMPs can also work well with RL. The devised experiment is a manipulation task where the robot is required to successfully drop a ball (with the radius 1.2 cm) to a cup. Meanwhile, a board is put between the starting position and the final position of the end effector. The purpose of placing the board is twofold. First, the end effector obstacle avoidance capability using CDMPs will be tested. Second, the movement range of the torso pitch will be constrained by CDMPs to prevent the upper body of the robot from hitting the board. The initial policy for each joint is a straight line with respective starting and final positions. The robot will miss the cup using the initial policy, due to a realistic wrong estimation of the cup position while the real one is at (0.318, 0.113, 0.670) m.

The total cost function J by the RL algorithm is composed by two parts: the intermediate cost J_t and the final cost J_N, i.e. $J = J_t + J_N$. The intermediate cost is formulated in terms of several concerns. First, the energy used by the robot is penalized by minimizing the cumulative torques $\boldsymbol{\tau}$. Besides, the end effector accelerations $\ddot{\mathbf{q}}$ are penalized for promoting a minimum jerk trajectory. Furthermore, the norm of the CDMPs parameters is included for obtaining a smooth regularized solution. So far, the intermediate cost J_t

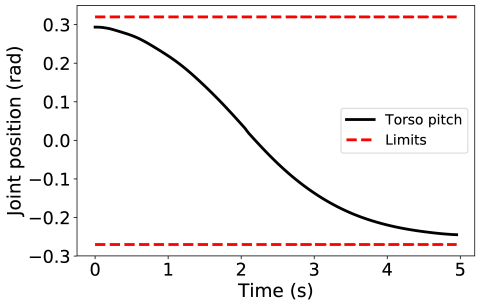


Fig. 5: Bounded torso pitch trajectory after learning.

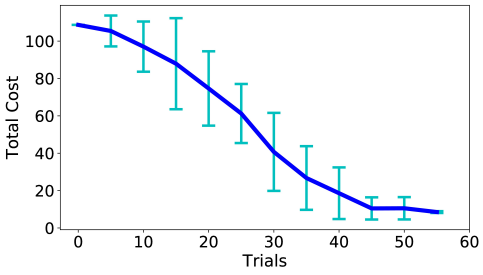


Fig. 6: Error-bar curves of cost values in the ball-dropping task. Solid curves represent the mean values averaged over 10 learning runs while the vertical bars denote the standard deviation. Each update is based on 5 trials. The first successful dropping happens after circa 12 trials.

is formulated as

$$J_t = \int_{t_0}^{t_N} \gamma_{\mathrm{E}} \boldsymbol{\tau}^\top \boldsymbol{\tau} + \gamma_{\mathrm{J}} \ddot{\mathbf{q}}^\top \ddot{\mathbf{q}} + \gamma_{\mathrm{N}} \boldsymbol{\Theta}^\top \boldsymbol{\Theta} dt, \tag{40}$$

where $\gamma_{\mathrm{E}} = 10^{-3}$, $\gamma_{\mathrm{J}} = 10^{-2}$ and $\gamma_{\mathrm{N}} = 10^{-2}$ are positive weight coefficients. The final cost J_N is formulated by the distance from the initial position and an indicator term:

$$J_N = \gamma_{\mathrm{D}} \exp\left((\mathbf{x}_{\mathrm{N}} - \mathbf{x}_0)^2\right) + \gamma_{\mathrm{I}} \mathbf{1}_{drop}, \tag{41}$$

where $\mathbf{x}_0$ and $\mathbf{x}_{\mathrm{N}}$ are the end effector initial and final position, respectively ($\mathbf{x}_0 = (0.312, 0.190, 0.790)$ m is constant throughout the training process); $\gamma_{\mathrm{D}} = 1$ and $\gamma_{\mathrm{I}} = 100$ are cost weights; $\mathbf{1}_{drop}$ is an indicator function with value 1 when missing the cup and 0 when dropping the ball successfully.

2) Experimental results and discussion: It can be seen in Fig. 4c that the robot missed the cup with the initial policy, while after training for some trials, the robot can find the correct cup position and thus drop the ball in the cup successfully as in Fig. 4d. As mentioned earlier, the movement range of the torso pitch is bounded between 0.32 rad and -0.27 rad. One of the optimized torso pitch trajectories is shown in Fig. 5 with the maximum and minimum value 0.29 rad and -0.25 rad, respectively. The proposed CDMPs offer a convenient method to set the limits for the generated trajectory, while the traditional DMPs have no such advantage. The learning results are shown as in Fig.

6, with the total cost being reduced continuously. It can be concluded that CDMPs can also work well together with RL.

V. CONCLUSIONS

In this paper we presented novel CDMPs towards bounding the original DMPs states. The derivation of CDMPs is inspired by a previously proposed joint limit avoidance control law. The core idea of CDMPs is to transform the feasible demonstrated trajectory states into the exogenous states and then control the exogenous states instead. The proposed method ensures that the resulting joint trajectory will evolve within the specified movement limits. The effectiveness of CDMPs is verified by two toy experiments, an obstacle avoidance task, and a ball dropping task, where the unbounded motion caused by the original DMP coupling terms or the exploration noises in RL is alleviated. As an extension, we plan to apply CDMPs to the whole body movement generation and optimization.

ACKNOWLEDGMENT

The authors would like to thank Silvio Traversaro for his contribution to the preparation of simulation environment. L. R. gratefully acknowledges the financial support of the AFOSR projects FA9550-17-1-0390 and BAA-AFRL-AFOSR-2016-0007 (European Office of Aerospace Research and Development), the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216, the Italian Institute of Technology, and the EU H2020-MSCA-RISE project NoMADS - DLV-777826.

REFERENCES

[1] Y. Huang, L. Rozo, J. Silvério, and D. G. Caldwell, "Kernelized movement primitives," *arXiv preprint arXiv:1708.08638*, 2017.

[2] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard, "Learning and reproduction of gestures by imitation," *IEEE Robotics & Automation Magazine*, vol. 17, no. 2, pp. 44–54, 2010.

[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] Y. Huang, D. Büchler, O. Koç, B. Schölkopf, and J. Peters, "Jointly learning trajectory generation and hitting point prediction in robot table tennis," in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*. IEEE, 2016, pp. 650–655.

[5] J. Kober and J. R. Peters, "Policy search for motor primitives in robotics," in *Advances in neural information processing systems*, 2009, pp. 849–856.

[6] J. Kober, A. Wilhelm, E. Oztop, and J. Peters, "Reinforcement learning to adjust parametrized motor primitives to new situations," *Autonomous Robots*, vol. 33, no. 4, pp. 361–379, Nov 2012.

[7] D.-H. Park, H. Hoffmann, P. Pastor, and S. Schaal, "Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields," in *Humanoid Robots (Humanoids), 2008 IEEE-RAS 8th International Conference on*. IEEE, 2008, pp. 91–98.

[8] M. Na, B. Yang, and P. Jia, "Improved damped least squares solution with joint limits, joint weights and comfortable criteria for controlling human-like figures," in *Robotics, Automation and Mechatronics, 2008 IEEE Conference on*. IEEE, 2008, pp. 1090–1095.

[9] T. F. Chan and R. V. Dubey, "A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 2, pp. 286–292, 1995.

[10] L. Chen and Y. Guo, "Hierarchical nonholonomic path planning of dual-arm space robot systems with joint limits," in *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, vol. 2. IEEE, 2006, pp. 8862–8865.

[11] L. Jamone, B. Damas, J. Santos-Victor, and A. Takanishi, "Online learning of humanoid robot kinematics under switching tools contexts," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4811–4817.

[12] M. Marey and F. Chaumette, "New strategies for avoiding robot joint limits: Application to visual servoing using a large projection operator," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 6222–6227.

[13] A. Dietrich, T. Wimböck, and A. Albu-Schäffer, "Dynamic whole-body mobile manipulation with a torque controlled humanoid robot via impedance control laws," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 3199–3206.

[14] Y. Huang, J. Silvério, L. Rozo, and D. G. Caldwell, "Hybrid probabilistic trajectory optimization using null-space exploration," in *Robotics and Automation (ICRA), 2018 IEEE International Conference on*. IEEE, 2018, pp. 7226–7232.

[15] M. Charbonneau, F. Nori, and D. Pucci, "On-line joint limit avoidance for torque controlled robots by joint space parametrization," in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*. IEEE, 2016, pp. 899–904.

[16] L. Natale, C. Bartolozzi, D. Pucci, A. Wykowska, and G. Metta, "iCub: The not-yet-finished story of building a robot child," *Science Robotics*, vol. 2, no. 13, p. eaaq1026, 2017.

[17] J. E. Marsden and T. S. Ratiu, *Introduction to mechanics and symmetry: a basic exposition of classical mechanical systems*. Springer Science & Business Media, 2013, vol. 17.

[18] G. Nava, F. Romano, F. Nori, and D. Pucci, "Stability analysis and design of momentum-based controllers for humanoid robots," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 680–687.

[19] E. Theodorou, J. Buchli, and S. Schaal, "A generalized path integral control approach to reinforcement learning," *Journal of Machine Learning Research*, vol. 11, no. Nov, pp. 3137–3181, 2010.

[20] G. Metta, L. Natale, F. Nori, G. Sandini, D. Vernon, L. Fadiga, C. Von Hofsten, K. Rosander, M. Lopes, J. Santos-Victor *et al.*, "The iCub humanoid robot: An open-systems platform for research in cognitive development," *Neural Networks*, vol. 23, no. 8-9, pp. 1125–1134, 2010.

[21] "Bullet Real-Time Physics Simulation," https://pybullet.org/wordpress/, accessed: 2018-07-11.

[22] F. Nori, S. Traversaro, J. Eljaik, F. Romano, A. Del Prete, and D. Pucci, "iCub whole-body control through force regulation on rigid non-coplanar contacts," *Frontiers in Robotics and AI*, vol. 2, p. 6, 2015.