Learning to Sequence Multiple Tasks with Competing Constraints

Anqing Duan[1,2], Raffaello Camoriano[3], Diego Ferigo[1,4], Yanlong Huang[5],
Daniele Calandriello[3], Lorenzo Rosasco[2,3] and Daniele Pucci[1]

Abstract— **Imitation learning offers a general framework where robots can efficiently acquire novel motor skills from demonstrations of a human teacher. While many promising achievements have been shown, the majority of them are only focused on single-stroke movements, without taking into account the problem of multi-tasks sequencing. Conceivably, sequencing different atomic tasks can further augment the robot's capabilities as well as avoid repetitive demonstrations. In this paper, we propose to address the issue of multi-tasks sequencing with emphasis on handling the so-called *competing constraints*, which emerge due to the existence of the concurrent constraints from Cartesian and joint trajectories. Specifically, we explore the null space of the robot from an information-theoretic perspective in order to maintain imitation fidelity during transition between consecutive tasks. The effectiveness of the proposed method is validated through simulated and real experiments on the iCub humanoid robot.**

I. Introduction

Imitation learning, also known as learning by demonstration or kinesthetic teaching, allows robots to learn new skills from a human teacher [1]. The demonstrated trajectories are usually represented through the well-established movement primitives [2], [3], [4] for later reproduction and adaptation on robot platforms. Following this paradigm, robots can learn a variety of hard-to-engineer motor skills, such as table tennis [5] and baseball batting [3].

Although imitation learning algorithms have endowed robots with quite intricate skills, most of the related research is only focused on mimicking single-stroke movements. In order to enable robots to perform more complex tasks, it seems a natural requirement to sequence several basic atomic tasks. For example, modern robots, especially general-purpose humanoid robots, usually possess several different skills in their armory. When operating in the real world, instead of executing these tasks separately, they are expected to execute one task after another sequentially. Conceivably, the ability of multi-tasks sequencing plays an important role in pushing robots towards a higher level of flexibility and autonomy.

In addition to further enhancing robots' capabilities, another benefit brought by multi-tasks sequencing is to exempt a human teacher from repetitive demonstrations. Considering the combinatorial number of basic tasks could be explosive, learning all the possible combinations exclusively through human demonstrations would be impracticable. Autonomous planning of trajectories for sequential tasks is especially meaningful when a multi-step task is difficult to demonstrate using a single demonstration or requires very lengthy demonstrations.

A. Related Work

In the literature, task sequencing can be roughly addressed from two perspectives, namely trajectory manipulation and sequences generation.

On trajectory manipulation, two methods were proposed in [6] to realize action sequencing, namely proper initialization of the third-order dynamic movement primitives or using online Gaussian kernel functions modification of the second-order dynamic movement primitives. Other alternatives for joining movement sequences include modification of the original dynamic movement primitives by overlapping kernels with a specific application to the handwriting task [7]. Sequencing of simple demonstrations can also be applied to achieve a complex bi-manual manipulation task [8].

On sequences generation, a variety of techniques can be employed. For example, in [9] a motion primitive graph was constructed to learn the temporal relationship between motion primitives. The constructed motion primitive graph was then used to generate motion consisting of sequences of motion primitives. Sequences and transitions between skills can also be realized by bifurcating dynamical systems based on continuous-time recurrent neural networks, whose output was used as activation signal for movement primitives in [10]. By representing demonstrations with a sequence graph, transitions between consecutive basic movements could also be identified by tackling a classification problem using support vector machines or Gaussian mixture models [11].

Recent research also focused on learning multi-stage tasks from a single video of a human performing the task [12].

B. Problem Statement

Although the problem of multi-tasks sequencing can be tackled from various perspectives as mentioned above, a common issue in the existing literature is the lack of methods

This work is supported by Italian Institute of Technology, Center for Brains, Minds and Machines (NSF STC CCF-1231216), and NVIDIA Corporation.

[1] Dynamic Interaction Control Lab, Istituto Italiano di Tecnologia, Via San Quirico 19D, Genoa, 16163, Italy. Email: firstname.lastname@iit.it

[2] DIBRIS, Università degli Studi di Genova, Genoa, Italy.

[3] Laboratory for Computational and Statistical Learning (IIT@MIT), Istituto Italiano di Tecnologia and Massachusetts Institute of Technology, Cambridge, MA, USA.

[4] Machine Learning and Optimization Group, School of Computer Science, University of Manchester, Manchester M13 9PL, U.K.

[5] Department of Advanced Robotics, Istituto Italiano di Tecnologia, Via Morego 30, Genova, 16163, Italy.

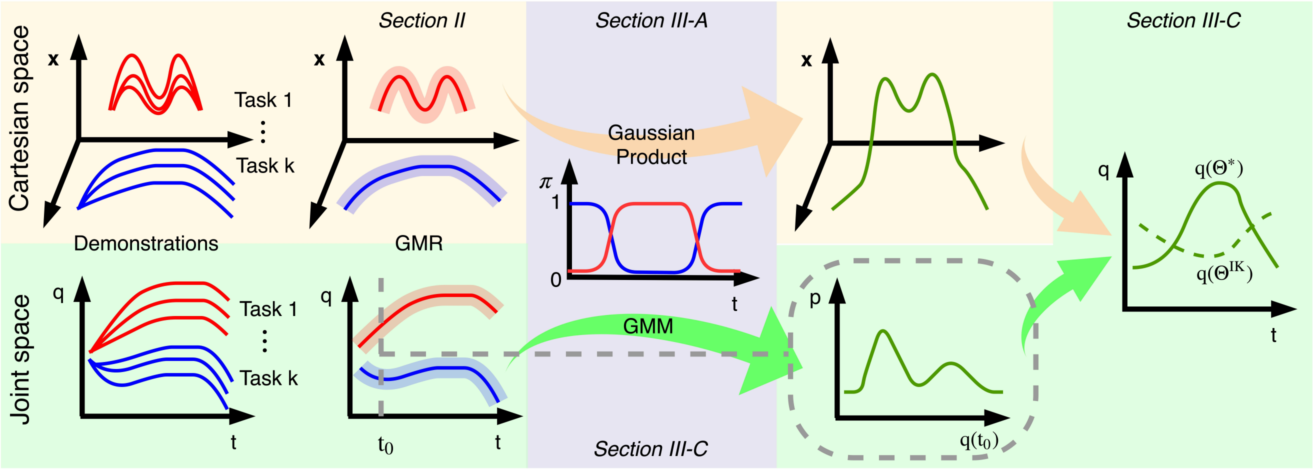

Fig. 1: Illustration of the proposed method for multi-tasks sequencing. First, both Cartesian and joint trajectories of several tasks are demonstrated, which are subsequently used for retrieving probabilistic trajectories through GMR. After that, Cartesian trajectory is blended by Gaussian product and joint trajectory is modeled by GMM, both according to the activation function π. Finally, the competing constraints are addressed by optimizing robot null-space parameter $\boldsymbol{\Theta}$.

for imitating both Cartesian and joint trajectories simultaneously. Arguably, learning in either space only is not always satisfying. For example, humanoid robots are usually required to behave resembling a human. This implies that both Cartesian space, which is responsible for the completion of a task, and joint space, which affects robot posture, are involved. In fact, relevant studies have shown that the more robots act like humans, the more willing humans are to trust and work side-by-side with them [13]. However, in order to learn Cartesian and joint trajectories simultaneously, one has to deal with the inconsistency between the two spaces encountered in multiple demonstrations. Such phenomena in imitation learning are referred to as *competing constraints* [14]. There has been relevant research effort dedicated to this topic, but without considering multi-tasks sequencing [15].

C. Scope of the Paper

In this paper, we propose to address the problem of multi-tasks sequencing with competing constraints. Specifically, we propose to exploit robot redundancy by exploring the robot null space to alleviate the conflicts resulting from Cartesian and joint space. The optimization of the null-space parameter is formulated as a reinforcement learning (RL) problem with the optimization objective designed from an information-theoretic perspective. An illustration of the proposed method is shown in Fig. 1.

This paper is organized as follows. Section II briefly reviews probabilistic modeling of multiple demonstrations. The proposed method for multi-tasks sequencing is introduced in Section III. Subsequently, the evaluative experiments are reported in Section IV. Finally, Section V concludes our results and discusses possible future work.

II. Probabilistic Modeling of Demonstrations

In order to exploit the probabilistic properties of trajectories, multiple demonstrations are required. We assume that we have M demonstrations, each of fixed length N. The dataset comprises the robot's end-effector Cartesian position $\mathbf{x} \in \mathbb{R}^3$ and joint positions $\mathbf{q} \in \mathbb{R}^d$, with d the number of active degrees of freedom. Both quantities are indexed by time t. Thus, we denote the dataset obtained from the demonstrations as $\{\{t_{n,m}, \mathbf{x}_{n,m}, \mathbf{q}_{n,m}\}_{n=1}^N\}_{m=1}^M$.

A. Gaussian Mixture Model

Upon collecting the demonstrations dataset, Gaussian mixture models (GMMs) are employed to model the joint probability distributions $p(t, \mathbf{x})$ and $p(t, \mathbf{q})$, respectively. Without loss of generality, we use $\mathbf{s}_t$ to denote either $\mathbf{x}_t$ or $\mathbf{q}_t$. A GMM with $H \in \mathbb{N}^+$ components is defined by a probability density function

$$p(t, \mathbf{s}_t) = \sum_{h=1}^{H} \eta_h \mathcal{N}(\boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h), \tag{1}$$

where

$$\boldsymbol{\mu}_h = \begin{bmatrix} \boldsymbol{\mu}_{t,h} \\ \boldsymbol{\mu}_{s,h} \end{bmatrix}, \tag{2}$$

$$\boldsymbol{\Sigma}_h = \begin{bmatrix} \boldsymbol{\Sigma}_{tt,h} & \boldsymbol{\Sigma}_{ts,h} \\ \boldsymbol{\Sigma}_{st,h} & \boldsymbol{\Sigma}_{ss,h} \end{bmatrix}. \tag{3}$$

η_h, $\boldsymbol{\mu}_h$ and $\boldsymbol{\Sigma}_h$ are the parameters of the h-th Gaussian component, defining the prior, mean, and covariance, respectively. Note that η_h is subject to $\sum_h \eta_h = 1$. Typically, there are several covariance constraints that can be used in GMM and the one used here is called *full covariance type* [16]. The mixture parameters can be obtained from maximum likelihood estimation using the standard expectation maximization algorithm [17].

B. Gaussian Mixture Regression

Gaussian mixture regression (GMR) has a simple formulation that has been employed to generate robot movements

[4]. The corresponding output $\hat{\mathbf{s}}(t)$ at each reproduction step t can be estimated in terms of conditional probability:

$$\hat{\mathbf{s}}(t) \sim \sum_{h=1}^{H} w_h(t)\mathcal{N}\big(\hat{\boldsymbol{\mu}}_h(t), \hat{\boldsymbol{\Sigma}}_h(t)\big), \tag{4}$$

where $w_h(t)$ are the activation functions defined as

$$w_h(t) = \frac{\eta_h \mathcal{N}(t \mid \boldsymbol{\mu}_{t,h}, \boldsymbol{\Sigma}_{tt,h})}{\sum_{i=1}^{H} \eta_i \mathcal{N}(t \mid \boldsymbol{\mu}_{t,i}, \boldsymbol{\Sigma}_{tt,i})}, \tag{5}$$

with

$$\hat{\boldsymbol{\mu}}_h(t) = \boldsymbol{\mu}_{s,h} + \boldsymbol{\Sigma}_{st,h}\boldsymbol{\Sigma}_{tt,h}^{-1}(t - \boldsymbol{\mu}_{t,h}), \tag{6}$$

$$\hat{\boldsymbol{\Sigma}}_h(t) = \boldsymbol{\Sigma}_{ss,h} - \boldsymbol{\Sigma}_{st,h}\boldsymbol{\Sigma}_{tt,h}^{-1}\boldsymbol{\Sigma}_{ts,h}. \tag{7}$$

Note that (4) can also be represented using a unimodal output distribution for the generated trajectory, i.e. $\hat{\mathbf{s}}(t) \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_t^s, \hat{\boldsymbol{\Sigma}}_t^s)$. By resorting to the law of total mean and variance, the approximated normal distribution can be derived as in [4]:

$$\hat{\boldsymbol{\mu}}_t^s = \sum_{h=1}^{H} w_h(t)\hat{\boldsymbol{\mu}}_h(t), \tag{8}$$

$$\hat{\boldsymbol{\Sigma}}_t^s = \sum_{h=1}^{H} w_h(t)\big(\hat{\boldsymbol{\Sigma}}_h(t) + \hat{\boldsymbol{\mu}}_h(t)\hat{\boldsymbol{\mu}}_h(t)^\top\big) - \hat{\boldsymbol{\mu}}_t^s\hat{\boldsymbol{\mu}}_t^{s\top}. \tag{9}$$

III. Multi-Tasks Sequencing

As discussed in Section I-B, behaving like a human is a highly desirable capability for robots, since it can increase human acceptance. We aim at maintaining imitation fidelity in the context of multi-tasks sequencing. In the language of *robot control*, human-like behavior implies that robots need not only follow Cartesian trajectories to accomplish the sequenced tasks (III-A), but also exploit redundancy at joint level (III-B) by optimizing null-space parameters (III-C) for human-like configuration.

A. Cartesian Trajectory Sequencing

Here, we consider the problem of multi-tasks sequencing in robot Cartesian space. Usually, in order to continuously combine and blend multiple movement primitives which are probabilistically encoded into a single movement, a common technique is to employ the *Gaussian product* [18]. By taking the product of trajectory distributions at each time step, the resulting trajectory again satisfies the Gaussian distribution. An important advantage of the Gaussian product is that it can capture the overlapping area of the activated trajectories. In general, the shape of the obtained trajectory shares higher similarity to the movement primitive that has higher probability density.

Suppose that there are in total K tasks that the robot has previously learned and for each task k the trajectory distribution $\mathbf{x}_k(t) \sim \mathcal{N}\big(\boldsymbol{\mu}_k(t), \boldsymbol{\Sigma}_k(t)\big)$ is retrieved using GMR. To allow for multi-tasks sequencing, we propose to modulate the activation status of the trajectories so that the tasks are executed one by one smoothly. Let us write the user-defined time-varying activation functions of each task k at time t as $\pi_k(t)$ with $\sum_k \pi_k(t) = 1$. The resulting trajectory $\bar{\mathbf{x}}$ can be obtained by co-activating the movement primitives:

$$p(\bar{\mathbf{x}}) \propto \prod_k p(\mathbf{x}_k)^{\pi_k}. \tag{10}$$

At each time step, the resulting distribution obeys Gaussian $\bar{\mathbf{x}}(t) \sim \mathcal{N}\big(\bar{\boldsymbol{\mu}}(t), \bar{\boldsymbol{\Sigma}}(t)\big)$, with its mean and covariance matrix given by [18]

$$\begin{aligned} \bar{\boldsymbol{\mu}}(t) &= \bar{\boldsymbol{\Sigma}}(t)\Big(\sum_k \big(\boldsymbol{\Sigma}_k(t)/\pi_k(t)\big)^{-1}\boldsymbol{\mu}_k(t)\Big), \\ \bar{\boldsymbol{\Sigma}}(t) &= \Big(\sum_k \big(\boldsymbol{\Sigma}_k(t)/\pi_k(t)\big)^{-1}\Big)^{-1}. \end{aligned} \tag{11}$$

Thus, multi-tasks sequencing in Cartesian space is accomplished by following the obtained trajectory.

B. Transformation from Cartesian Space to Joint Space

We need to transform Cartesian trajectories into joint space in order to control the robot. To do so, the Jacobian-based inverse kinematics technique $\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$ is employed, where $\mathbf{J} \in \mathbb{R}^{3\times d}$ is the robot Jacobian matrix [19]. In general, the corresponding discrete implementation incorporating null-space exploration is given by

$$\begin{aligned} &\mathbf{q}_t = \mathbf{q}_{t-1} + \mathbf{J}_{t-1}^{\dagger}(\mathbf{x}_t - \mathbf{x}_{t-1}) + (\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})\mathbf{N}(\boldsymbol{\Theta})\Delta t, \\ &\mathbf{N}(\boldsymbol{\Theta}) = \boldsymbol{\Theta}^\top\boldsymbol{\Phi}(t), \end{aligned} \tag{12}$$

where $\mathbf{J}^{\dagger} = \mathbf{J}^\top(\mathbf{J}\mathbf{J}^\top)^{-1}$ represents the Moore-Penrose pseudoinverse of $\mathbf{J}$ when $\mathbf{J}\mathbf{J}^\top$ is invertible, $\Delta t > 0$ is the time step, $\mathbf{I} \in \mathbb{R}^{d\times d}$ denotes an identity matrix, $\mathbf{N}$ corresponds to joint movement in null space, which is parameterized by $\boldsymbol{\Theta} \in \mathbb{R}^{\mathcal{O}\times d}$, and $\boldsymbol{\Phi}(t) \in \mathbb{R}^{\mathcal{O}\times 1}$ are basis functions with the total number $\mathcal{O}$ and the i-th element defined using the normalized Gaussian kernel function $\mathrm{e}^{-h_i(t-c_i)^2}/\sum_j \mathrm{e}^{-h_j(t-c_j)^2}$ with $h_i > 0$ and c_i equally spaced in the execution time interval.

When transforming the sequenced probabilistic Cartesian trajectory $\bar{\mathbf{x}}$ into joint space using (12), the obtained joint trajectory $\bar{\mathbf{q}}_t^C$ also satisfies the probabilistic distribution $\mathcal{N}(\bar{\boldsymbol{\mu}}^C(t), \bar{\boldsymbol{\Sigma}}^C(t))$ with mean $\bar{\boldsymbol{\mu}}^C(t)$ and covariance matrix $\bar{\boldsymbol{\Sigma}}^C(t)$ given by [15]

$$\begin{aligned} &\bar{\boldsymbol{\mu}}^C(t) = \bar{\mathbf{q}}_{t-1}^C + \mathbf{J}^{\dagger}(\bar{\boldsymbol{\mu}}_t - \bar{\boldsymbol{\mu}}_{t-1}) + (\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})\mathbf{N}(\boldsymbol{\Theta})\Delta t, \\ &\bar{\boldsymbol{\Sigma}}^C(t) = \mathbf{J}^{\dagger}\bar{\boldsymbol{\Sigma}}_t\mathbf{J}^{\dagger\top}. \end{aligned} \tag{13}$$

It should be noted that the obtained joint trajectory upon transformation can only guarantee that the Cartesian constraint is respected. In order to take into account joint level imitation, the parameter $\boldsymbol{\Theta}$ needs to be optimized according to the criteria proposed in the next section.

C. Optimization of Null-Space Parameters

The demonstrated joint trajectories can also be retrieved using GMR in a similar way to Cartesian trajectories. To represent different tasks, we propose to fuse the joint trajectories by GMM. It should be noted that joint trajectories are formulated in a different way from Cartesian ones, which are treated using the Gaussian product. This choice has been made since GMMs can preserve complete information on the robot's configuration for each sub-task together with their corresponding activation degree at each time step. Such information is very important, since one of the main goals for joint-level imitation is that the robot posture should be altered accordingly during the transition between consecutive tasks. In contrast, the Gaussian product only results in a single Gaussian with its mean value calculated by the summation of weighted sub-task means. In this way, the information on the original robot joint configuration for each sub-task is lost. Therefore, GMMs are a more reasonable choice for modeling multi-tasks joint trajectories, in that it can capture richer information than the Gaussian product.

For each task k, we denote the corresponding joint trajectory distribution as $\mathbf{q}_k^J(t) \sim \mathcal{N}\big(\boldsymbol{\mu}_k^J(t), \boldsymbol{\Sigma}_k^J(t)\big)$. As mentioned above, the reference joint trajectory $\bar{\mathbf{q}}^J(t)$ is formulated in terms of GMM as

$$p(\bar{\mathbf{q}}^J(t)) = \sum_k \pi_k(t)\mathcal{N}\big(\boldsymbol{\mu}_k^J(t), \boldsymbol{\Sigma}_k^J(t)\big). \tag{14}$$

One can observe that there are concurrent conflicts between Cartesian trajectory $\bar{\mathbf{q}}_t^C$ and joint trajectory $\bar{\mathbf{q}}_t^J$, which are usually referred to as *competing constraints*[1] [14]. In order to minimize such conflicts, we propose to formulate the optimization objective from an information-theoretic perspective. A widely used technique in statistics and pattern recognition to measure the distance between two probability distributions is the *Kullback-Leibler (KL) divergence*, also known as *relative entropy* [20]. Its usage in addressing competing constraints has been shown in [21], where the main purpose was to design a minimal intervention controller. By contrast, our focus is on sequencing multiple tasks. In order to minimize the inconsistency between $\bar{\mathbf{q}}_t^C$ and $\bar{\mathbf{q}}_t^J$, the null space will be explored by optimizing the null-space parameter $\boldsymbol{\Theta}$ according to the KL-divergence-based objective:

$$J_{\mathrm{KL}}(\boldsymbol{\Theta}) = \sum_t D_{\mathrm{KL}}\big(p(\bar{\mathbf{q}}_t^J) \parallel p(\bar{\mathbf{q}}_t^C; \boldsymbol{\Theta})\big), \tag{15}$$

where the expression for the KL divergence is defined as

$$D_{\mathrm{KL}}\big(p(\bar{\mathbf{q}}_t^J) \,\|\, p(\bar{\mathbf{q}}_t^C; \boldsymbol{\Theta})\big) \triangleq \int p(\bar{\mathbf{q}}_t^J) \log \frac{p(\bar{\mathbf{q}}_t^J)}{p(\bar{\mathbf{q}}_t^C; \boldsymbol{\Theta})} d\mathbf{q}_t. \tag{16}$$

It should be noted that the KL divergence is asymmetric. The choice of the direction of the KL divergence is problem-dependent. In our case, the moment projection form rather than the information projection form is employed, since during the task transition period when $p(\bar{\mathbf{q}}_t^J)$ has multiple modes, $p(\bar{\mathbf{q}}_t^C; \boldsymbol{\Theta})$ will choose to blur different modes together so as to yield more natural action for tasks switch.

Algorithm 1 Multi-tasks sequencing with competing constraints

```
Collect the dataset of $K$ tasks, each containing $M$ demonstrations having length $N$:
    $\{\{\{t_{n,m}^k, \mathbf{x}_{n,m}^k, \mathbf{q}_{n,m}^k\}_{n=1}^N\}_{m=1}^M\}_{k=1}^K$
Retrieve Cartesian and joint trajectory distribution $\{\mathbf{x}_k(t), \mathbf{q}_k(t)\}$ from demonstrations using GMR
Specify task sequencing schedule $\pi_k(t)$
Initialize RL hyperparameters $\boldsymbol{\Theta}_0$, $\boldsymbol{\Sigma}_\varepsilon$, $\lambda$, $h_i$, $c_i$, $L$
repeat
    for $l = 1$ to $L$ do
        Sample $\boldsymbol{\varepsilon}_l \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\varepsilon)$
        $\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} + \boldsymbol{\varepsilon}_l$
        for $t = 1$ to $N$ do
            Blend Cartesian trajectory using (11)
            Transform into joint trajectory using (13)
            Estimate time step cost by (17) and (18)
        end for
        Compute trial cost as (15)
    end for
    Update $\boldsymbol{\Theta}$ according to (19)
until $\boldsymbol{\Theta}$ converge
Calculate the optimal trajectory $\{\mathbf{q}^*\}_{t=1}^N$ from (12)
return $\{\mathbf{q}^*\}_{t=1}^N$
```

One issue arising from the optimization objective in (15) is that the calculation of the KL divergence between GMMs and a Gaussian is not analytically tractable. Therefore, an approximation method for (16) is necessary to facilitate solving the optimization problem. Theoretically, there are several approaches to the calculation of the KL divergence between GMMs, such as Monte Carlo sampling, unscented transformation, matched bound approximation, etc. Here, the one based on the variational lower bound to the likelihood is chosen because of its simple closed-form expression as well as its high accuracy [22]. Formally, we approximate (16) as (please refer to Appendix for the derivations)

$$\begin{aligned} &D_{\mathrm{KL}}\big(p(\bar{\mathbf{q}}_t^J) \parallel p(\bar{\mathbf{q}}_t^C; \boldsymbol{\Theta})\big) \approx \\ &\sum_k \pi_k(t) \log \frac{\sum_{k'} \pi_{k'}(t) \mathrm{e}^{-D_{\mathrm{KL}}\left(p(\mathbf{q}_{k,t}^J) \| p(\mathbf{q}_{k',t}^J)\right)}}{\mathrm{e}^{-D_{\mathrm{KL}}\left(p(\mathbf{q}_{k,t}^J) \| p(\bar{\mathbf{q}}_t^C; \boldsymbol{\Theta})\right)}}. \end{aligned} \tag{17}$$

Now the involved KL divergence calculation in (17) is only between multivariate Gaussian distributions, which has an analytical solution. Taking the KL divergence between $\mathbf{q}_{k,t}^J$ and $\mathbf{q}_t^C$ as an example, its form is given by [22]

$$\begin{aligned} &D_{\mathrm{KL}}\big(p(\mathbf{q}_{k,t}^J) \parallel p(\bar{\mathbf{q}}_t^C; \boldsymbol{\Theta})\big) = \frac{1}{2}\Big(\mathrm{Tr}(\bar{\boldsymbol{\Sigma}}_t^{C\,-1} \boldsymbol{\Sigma}_{k,t}^J) + \\ &(\bar{\boldsymbol{\mu}}_t^C - \boldsymbol{\mu}_{k,t}^J)^\top \bar{\boldsymbol{\Sigma}}_t^{C\,-1} (\bar{\boldsymbol{\mu}}_t^C - \boldsymbol{\mu}_{k,t}^J) + \log \frac{|\bar{\boldsymbol{\Sigma}}_t^C|}{|\boldsymbol{\Sigma}_{k,t}^J|} - d\Big), \end{aligned} \tag{18}$$

where $\mathrm{Tr}(\cdot)$ and $|\cdot|$ denote the trace and the determinant of a matrix, respectively. The KL divergence between other terms can be calculated similarly.

Given the complexity of the formulated optimization problem, it can be addressed with the help of RL algorithms. In this work, we apply the simplified path improvement with path integrals in search for the optimal null-space parameter $\boldsymbol{\Theta}$ [23]. The update rule for $\boldsymbol{\Theta}$ is given as

$$\boldsymbol{\Theta}_{i+1} \leftarrow \boldsymbol{\Theta}_i + \frac{\sum_{l=1}^L \boldsymbol{\varepsilon}_l \mathrm{e}^{-\frac{1}{\lambda} J(\boldsymbol{\Theta}_i + \boldsymbol{\varepsilon}_l)}}{\sum_{l=1}^L \mathrm{e}^{-\frac{1}{\lambda} J(\boldsymbol{\Theta}_i + \boldsymbol{\varepsilon}_l)}}, \tag{19}$$

[1] Time dependence is moved to subscript for simplicity.

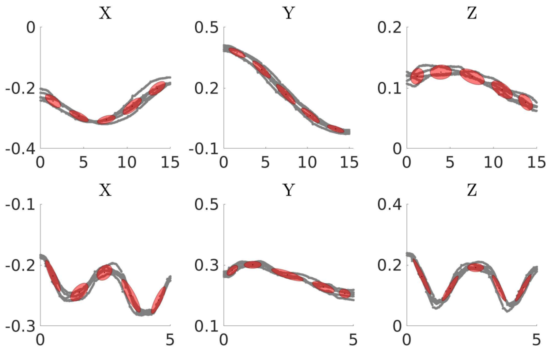

Fig. 2: GMM modeling of the demonstrated Cartesian trajectories for the pick-and-place task (*top row*) and the cleaning task (*bottom row*). The grey trajectories represent multiple demonstrations and the red ellipses are Gaussian components in GMM.

where $\mathbf{\Theta}_i$ is the initial parameter value for the i-th iteration, $\lambda > 0$ is a constant which can be considered as a factor controlling the learning rate and $\boldsymbol{\varepsilon}_l \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_\varepsilon)$ is the exploration noise for the l-th trial. The update rule keeps being executed until no further improvement on $\mathbf{\Theta}$ is observed.

The complete proposed method for multi-tasks sequencing with competing constraints is summarized in Algorithm 1.

IV. Experiments

The proposed method is validated on the iCub humanoid robot [24], both in simulation [25] as well as in real experiments. The case study we choose for concept proof is to sequence a pick-and-place task as well as a cleaning task. In this section, we will show that by teaching the robot these two tasks separately, the robot can learn by itself to sequence them together.

Firstly, the robot is taught two tasks individually with each one demonstrated for 5 times. A GMM with 5 states is used for modeling the joint probabilistic distribution between time and the corresponding output. The demonstrated Cartesian and joint trajectories with GMM modeling are shown in Fig. 2 and Fig. 3, respectively. For robot control, GMR is employed to extract the reference trajectory. The reproduction of the demonstrated tasks on the real iCub humanoid robot is shown in Fig. 4.

Next, iCub is required to sequence two demonstrated tasks together. To this end, the experimental set-up is conceived as follows: iCub will firstly pick a kitchen sponge handed over by a user, then perform the cleaning task, and, lastly, place the kitchen sponge at the destination. To start with, Cartesian trajectories of the two demonstrated tasks will be blended by Gaussian product. The extracted Cartesian trajectories for each task as well as the resulting blended trajectory are shown in the top row of Fig. 5. The corresponding weights used in the Gaussian product are given by a time-varying activation function, as shown in the bottom row of Fig. 5. According to the given activation function, the pick-and-place task will last for 15 s, during which the cleaning task will be triggered and last for 5 s.

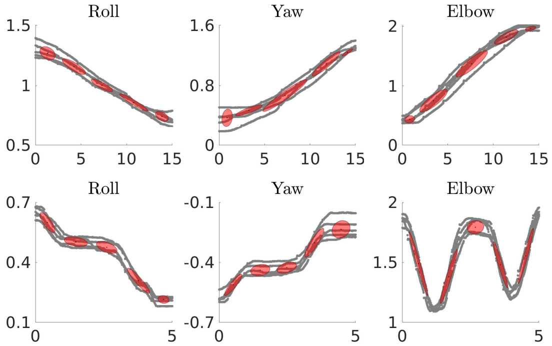

Fig. 3: GMM modeling of the demonstrated joint trajectories (shoulder roll, yaw, and elbow) for the pick-and-place task (*top row*) and the cleaning task (*bottom row*). The grey trajectories represent multiple demonstrations and the red ellipses are Gaussian components in GMM.

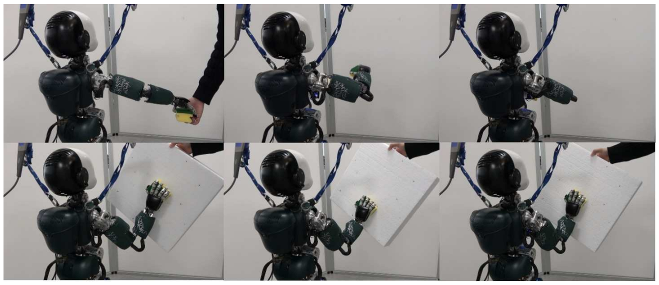

Fig. 4: Snapshots of reproduction for the pick-and-place task (*top row*) and the cleaning task (*bottom row*).

Upon the availability of the blended Cartesian trajectories, the corresponding joint trajectories shall be obtained by Jacobian-based inverse kinematics (IK) with the null-space parameters exposed. These parameters will be optimized to drive the robot configuration towards the demonstrated joint trajectories, which are modeled using GMM. The optimization is tackled by the reward-weighted RL algorithm [23] with the hyperparameters empirically set as $\mathbf{\Theta}_0 = \mathbf{0}$, $\mathbf{\Sigma}_\varepsilon = 10^{-2}\mathbf{I}$, $\lambda = 0.3$, and $L = 5$. The resulting optimized joint trajectories and their comparison with the trajectories from Jacobian-based inverse kinematics are shown in Fig. 6. The trajectories from inverse kinematics deviate from the demonstrated joint trajectories very much and therefore the imitation of the joint trajectories is broken. Instead, the trajectories with the optimal null-space parameters tend to match the corresponding demonstrated joint trajectories. The robot is trying to maintain the configuration of each demonstrated task during the process of sequencing the two tasks together. The RL algorithm performance is reported in Fig. 7. The parameters to be optimized are updated for

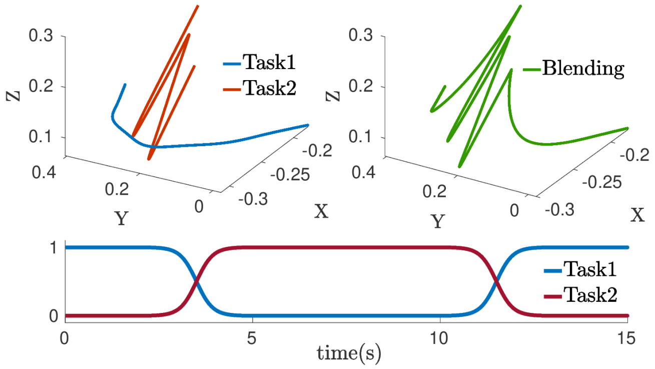

Fig. 5: Cartesian trajectories sequencing (*top row*) with their corresponding activation functions (*bottom row*). The blue trajectory represents the pick-and-place task, the red trajectory represents the cleaning task and the green trajectory shows the result of sequencing.

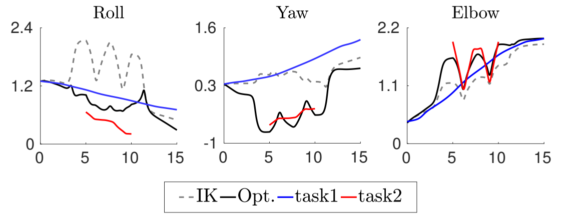

Fig. 6: Comparison of the joint trajectories with the optimal null-space parameters and inverse kinematics. The joint trajectories for two tasks are also included for reference.

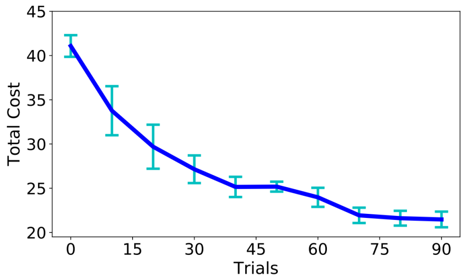

Fig. 7: The error-bar curve of the KL-divergence based cost during learning of the optimal null-space parameters. The vertical bars denote the standard deviations

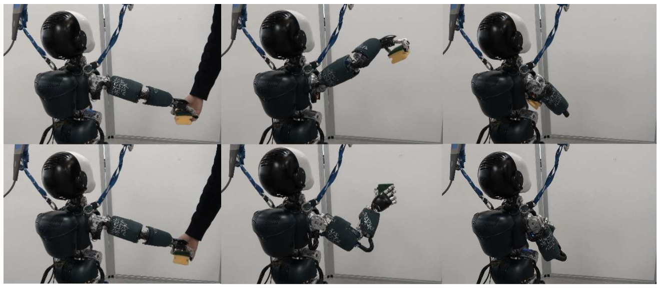

Fig. 8: Snapshots of the results of sequencing two tasks. The joint trajectories with the optimal null-space parameters (*bottom row*) result in more natural behavior than inverse kinematics solution (*top row*).

20 times with each one running 5 trials. The error bar is obtained by repeating the learning process for 5 times.

Finally, we evaluate our proposed method on the real iCub humanoid robot. The comparison between the optimized joint trajectories and the inverse-kinematics-based solution is provided in Fig. 8. It can be easily observed that the robot's movement appears more natural by running the optimized joint trajectories. This is evidenced by the fact that when the robot is undergoing task switch, the corresponding imitation emphasis also shifts from one task to another gradually.

V. Conclusions and Future Work

We addressed the issue of multi-tasks sequencing with competing constraints. To sequence Cartesian trajectories, the Gaussian product is employed for blending the trajectories from different tasks. By contrast, in joint space, the trajectories are modeled using GMMs to capture richer information. To handle competing constraints, null-space parameters are optimized according to a KL-divergence-based objective. Moreover, due to the analytical intractability of the KL divergence between GMM and a Gaussian, the optimized objective is approximated via the variational lower bound approach. We demonstrated the effectiveness of the proposed method through the case study of sequencing a pick-and-place task and a cleaning task, with experiments in simulation and on the real iCub humanoid robot.

It should be noted that, in this paper, our consideration was limited to kinematic position tasks only. For future work, this limitation could be resolved by incorporating force-based constraints for compliant interactions [26] and considering orientation issues [27]. In addition to the lack of torque learning, even though we used cleaning task as a showcase, we neither dealt with the contacts issue nor with the grasping problem. More widely applicable results could be obtained by further investigating these aspects. Also, to make robots safely operate in uncluttered environments, it can be considered to incorporate obstacle and joint limit avoidance [19]. Another interesting extension would be to learn the activation function profile π in response to the surrounding environment. With task sequences automatically generated instead of given beforehand, robots would be capable of operating more flexibly and autonomously.

Appendix

First let us denote $\mathbb{E}_{p(\bar{\mathbf{q}}_t^J)}[\log\big(p(\bar{\mathbf{q}}_t^C;\boldsymbol{\Theta})\big)]$ as $L_J(C)$. The KL-divergence from (16) can be decomposed as[2]

$$D_{\mathrm{KL}}\big(p(\bar{\mathbf{q}}^J)\,\|\,p(\bar{\mathbf{q}}^C;\boldsymbol{\Theta})\big) = L_J(J) - L_J(C). \tag{20}$$

[2] Without ambiguity, time dependence is dropped out here for simplicity.

We define variational parameters $\phi_{k'|k} > 0$ with $\sum_{k'} \phi_{k'|k} = 1$. The lower bound to $L_J(J)$ can be derived based on the Jensen's inequality [22]:

$$\begin{aligned} L_J(J) &= \mathbb{E}_{p(\bar{\mathbf{q}}^J)}[\log\big(p(\bar{\mathbf{q}}^J)\big)] \\ &= \mathbb{E}_{p(\bar{\mathbf{q}}^J)}[\log \sum_{k'} \pi_{k'} p(\mathbf{q}^J_{k'})] \\ &= \mathbb{E}_{p(\bar{\mathbf{q}}^J)}[\log \sum_{k'} \phi_{k'|k} \frac{\pi_{k'} p(\mathbf{q}^J_{k'})}{\phi_{k'|k}}] \\ &\geq \mathbb{E}_{p(\bar{\mathbf{q}}^J)}[\sum_{k'} \phi_{k'|k} \log \frac{\pi_{k'} p(\mathbf{q}^J_{k'})}{\phi_{k'|k}}] \\ &\triangleq \mathcal{L}_J(J, \phi). \end{aligned} \tag{21}$$

The obtained lower bound is maximized with respect to the variational parameters ϕ_k. Such constrained optimization problem can be solved by using Lagrangian multiplier. The maximum value is achieved when [28]

$$\phi^*_{k'|k} = \frac{\pi_{k'} \mathrm{e}^{-D_{\mathrm{KL}}\left(p(\mathbf{q}^J_k) \| p(\mathbf{q}^J_{k'})\right)}}{\sum_{\hat{k}} \pi_{\hat{k}} \mathrm{e}^{-D_{\mathrm{KL}}\left(p(\mathbf{q}^J_k) \| p(\mathbf{q}^J_{\hat{k}})\right)}}. \tag{22}$$

By substituting (22) into (21), the lower bound now becomes [28]

$$\begin{aligned} \mathcal{L}_J(J, \phi^*) = \sum_k \pi_k \log \sum_{k'} \pi_{k'} \mathrm{e}^{-D_{\mathrm{KL}}\left(p(\mathbf{q}^J_k) \| p(\mathbf{q}^J_{k'})\right)} \\ - \sum_k \pi_k H\big(p(\mathbf{q}^J_k)\big), \end{aligned} \tag{23}$$

where $H(p(\mathbf{q}^J_k))$ is the entropy of $p(\mathbf{q}^J_k)$ with its expression given as [17]

$$H\big(p(\mathbf{q}^J_k)\big) \triangleq -\int p(\mathbf{q}^J_k) \log p(\mathbf{q}^J_k) d\mathbf{q}. \tag{24}$$

For the calculation of $L_J(C)$, the exact expression is available immediately since $p(\bar{\mathbf{q}}^C)$ has only one component:

$$\begin{aligned} L_J(C) = \sum_k \pi_k \log \mathrm{e}^{-D_{\mathrm{KL}}\left(p(\mathbf{q}^J_k) \| p(\bar{\mathbf{q}}^C)\right)} \\ - \sum_k \pi_k H\big(p(\mathbf{q}^J_k)\big). \end{aligned} \tag{25}$$

By substituting (23) and (25) into (20), we can finally obtain the approximation as (17).

References

[1] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[2] Y. Huang, L. Rozo, J. Silvério, and D. G. Caldwell, "Kernelized movement primitives," *The International Journal of Robotics Research*, vol. 38, no. 7, pp. 833–852, 2019.

[3] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *2002 IEEE International Conference on Robotics and Automation*. IEEE, 2002, pp. 1398–1403.

[4] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intelligent Service Robotics*, vol. 9, no. 1, pp. 1–29, 2016.

[5] 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.

[6] B. Nemec and A. Ude, "Action sequencing using dynamic movement primitives," *Robotica*, vol. 30, no. 5, pp. 837–846, 2012.

[7] T. Kulvicius, K. Ning, M. Tamosiunaite, and F. Worgötter, "Joining movement sequences: Modified dynamic movement primitives for robotics applications exemplified on handwriting," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 145–157, 2012.

[8] R. Lioutikov, O. Kroemer, G. Maeda, and J. Peters, "Learning manipulation by sequencing motor primitives with a two-armed robot," in *Intelligent Autonomous Systems 13*. Springer, 2016, pp. 1601–1611.

[9] D. Kulić, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura, "Incremental learning of full body motion primitives and their sequencing through human motion observation," *The International Journal of Robotics Research*, vol. 31, no. 3, pp. 330–345, 2012.

[10] T. Luksch, M. Gienger, M. Mühlig, and T. Yoshiike, "Adaptive movement sequences and predictive decisions based on hierarchical dynamical systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 2082–2088.

[11] S. Manschitz, J. Kober, M. Gienger, and J. Peters, "Learning to sequence movement primitives from demonstrations," in *Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 4414–4421.

[12] T. Yu, P. Abbeel, S. Levine, and C. Finn, "One-shot hierarchical imitation learning of compound visuomotor tasks," *arXiv preprint arXiv:1810.11043*, 2018.

[13] S. You and L. P. Robert Jr, "Human-robot similarity and willingness to work with a robotic co-worker," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2018, pp. 251–260.

[14] S. Calinon and A. Billard, "Statistical learning by imitation of competing constraints in joint space and task space," *Advanced Robotics*, vol. 23, pp. 2059–2076, 2009.

[15] 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.

[16] S. Calinon and D. Lee, "Learning control," *Humanoid Robotics: a Reference*, pp. 1–52, 2016.

[17] C. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006.

[18] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced lectures on machine learning*. Springer, 2004, pp. 63–71.

[19] A. Duan, R. Camoriano, D. Ferigo, D. Calandriello, L. Rosasco, and D. Pucci, "Constrained DMPs for feasible skill learning on humanoid robots," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2018, pp. 1–6.

[20] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[21] Y. Huang, J. Silvério, and D. G. Caldwell, "Towards minimal intervention control with competing constraints," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 733–738.

[22] J. R. Hershey and P. A. Olsen, "Approximating the Kullback Leibler divergence between Gaussian mixture models," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4. IEEE, 2007, pp. IV–317.

[23] F. Stulp and O. Sigaud, "Robot skill learning: From reinforcement learning to evolution strategies," *Paladyn, Journal of Behavioral Robotics*, vol. 4, no. 1, pp. 49–61, 2013.

[24] 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.

[25] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2019.

[26] 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.

[27] Y. Huang, F. J. Abu-Dakka, J. Silvério, and D. G. Caldwell, "Generalized orientation learning in robot task space," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2531–2537.

[28] J.-L. Durrieu, J.-P. Thiran, and F. Kelly, "Lower and upper bounds for approximation of the kullback-leibler divergence between gaussian mixture models," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 4833–4836.