Adaptive Multi-Task Human-Robot Interaction Based on Human Behavioral Intention

JIAN FU\textsuperscript{1} (Member, IEEE), JINYU DU\textsuperscript{1}, XIANG TENG\textsuperscript{1}, YUXIANG FU\textsuperscript{2}, AND LU WU\textsuperscript{3}

\textsuperscript{1}School of Automation, Wuhan University of Technology, Wuhan 430070, China
\textsuperscript{2}Department of Computer Science, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada
\textsuperscript{3}School of Information, Wuhan University of Technology, Wuhan 430070, China

Corresponding author: Lu Wu (wulv@whut.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61773299.

ABSTRACT Learning from demonstrations with Probabilistic Movement Primitives (ProMPs) has been widely used in robot skill learning, especially in human-robot collaboration. Although ProMP has been extended to multi-task situations inspired by the Gaussian mixture model, it still treats each task independently. ProMP ignores the common scenario that robots conduct adaptive switching of the collaborative tasks in order to align with the instantaneous change of human intention. To solve this problem, we proposed an alternate learning-based parameter estimation method and an empirical minimum variation-based decomposition strategy with projection points, combining with linear interpolation strategy for weights, based on a Gaussian mixture model framework. Alternate learning of weights and parameters in multi-task ProMP (MTProMP) allows the robot to obtain a smooth composite trajectory planning which crosses expected via points. Decomposition strategy reflects how the desired via point state is projected onto the individual ProMP component, rendering the minimum total sum of deviations between each projection point with the respective prior. Linear interpolation is used to adjust the weights among sequential via points automatically. The proposed method and strategy are successfully extended to multi-task interaction ProMPs (MTiProMP). With MTProMP and MTiProMP, the robot can be applied to multiple tasks in industrial factories and collaborate with the worker to switch from one task to another according to changing intentions of the human. Classical via points trajectory planning experiments and human-robot collaboration experiments are performed on the Sawyer robot. The results of experiments show that MTProMP and MTiProMP with the proposed method and strategy perform better.

INDEX TERMS Human robot interaction, motion planning, MTProMP, MTiProMP, alternate learning, decomposition strategy.

I. INTRODUCTION

Robots are utilized in diverse occasions, not only in factories but also in homes, hospitals, etc. On these occasions, robots are usually capable of performing various tasks according to commands and context. Meanwhile, there is a trend that people enter the working environment of robots and interact with them [1]. These all indicate the urgent need for rapid robot-skill learning methods since it endows robots the ability of skill acquisition and refining akin to man. Learning from demonstrations (LfD) is a significant research paradigm in fast robot-skill learning and has been attracting widespread interest in the field of human-robot collaboration [2]. Movement Primitives (MPs) are mature methods for modeling robot movement policies in LfD. MPs were widely deployed in manipulating [3], motion [4], human-robot collaboration [5], and other robot tasks due to its generalization to new tasks, temporal modulation of the movement, co-activation of multiple primitives, and other good properties.

A probabilistic formulation of the MPs (Probabilistic Movement Primitives, ProMPs) was proposed in [6]. The modulation of a movement can be achieved by controlling the target’s positions and velocities with ProMP. It couples the multiple joints of the robot by modeling the covariance between trajectories of several degrees of freedom. Maeda et al. [7], [8] utilized Interaction ProMP, abbreviated as iProMP, to build the joint distributions between the man and the robot. iProMP can guide the robot to execute the specific task on the condition that the trajectories of a human’s multiple joints with upper limb are observed. However,
In this paper, we proposed an alternate learning parameter estimation method and projection points strategy for MTiProMP model and MTiProMP model, which are inspired by Gaussian Mixture Model (GMM) and linear basis function fitting. In MTiProMP and MTiProMP, each ProMP model or iProMP model represents a separate task. MTProMP is designed for robots that conduct multiple tasks, and MTiProMP is suitable for multi-task human-robot collaboration. During the demonstration learning phase, MTProMP builds the relationship between each ProMP, and MTiProMP establishes the relationship between each iProMP. During the predicting phase, the proposed alternate learning method based on the EM algorithm [14] is utilized to update the weight and parameter of each ProMP or iProMP. They are applied to implement fast, adaptive task switching in single-robot scenarios and human-robot collaboration scenarios, respectively.

**II. PRELIMINARY**

In this section, we briefly introduce the structure of ProMP and iProMP and their parameter estimation methods.

**A. THE STRUCTURE OF PROMP**

The ProMP model is Bayesian linear regression combined with Gaussian basis functions, and it is a movement primitive describing a trajectory in terms of a probability distribution. The form of ProMP is shown below

\[ q_t = \begin{bmatrix} q_{t,1} \\ \vdots \\ q_{t,J} \end{bmatrix} = \begin{bmatrix} \Phi_{t,1}^T \\ \vdots \\ \Phi_{t,J}^T \end{bmatrix} w + \epsilon = \Psi_{t}^T w + \epsilon, \]

where \( q_t \) includes joint position \( q_t \) and velocity \( \dot{q}_t \) at time \( t \). \( q_t \) can be extended, not limited to joint position and velocity. Parascos [15] extended the state \( q_t = [q_t, \dot{q}_t]^T \) to \( q_t = [q_t, q_t, u_t]^T \), and \( u_t \) is the control of the robot. In this paper, \( q_t \) will be extended to \( q_t = [x_t, y_t, z_t]^T \), which denotes the Cartesian coordinates. \( \varphi_t \in \mathbb{R}^N \) is the Gaussian basis function matrix (\( N \) is the number of basis functions) at time \( t \), and \( \varphi_t \in \mathbb{R}^N \) is the corresponding derivative. \( w \in \mathbb{R}^P \) is the parameter of ProMP, and it represents the weight of each basis function. \( \epsilon \sim \mathcal{N}(0, \Sigma_\epsilon) \) is gaussian noise with zero mean. Considering the problem with \( Q \) degrees of freedom (DoF), the form of ProMP can be expressed as

\[ y_t = \begin{bmatrix} y_{t,1} \\ \vdots \\ y_{t,Q} \end{bmatrix} = \begin{bmatrix} \Psi_{t,1}^T \\ \vdots \\ \Psi_{t,Q}^T \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_Q \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_Q \end{bmatrix} = H_t \omega + \epsilon_y, \]

where \( y_t \in \mathbb{R}^{2Q} \) is the observation variable with \( Q \) DoFs, \( q_{t,j} \) are position and velocity of \( j \)th DoFs. \( H_t^j = \text{diag}(\varphi_{t,1,j}, \ldots, \varphi_{t,Q,j}) \) is an observation matrix, and \( \omega \in \mathbb{R}^{NQ} \) is the parameter of multivariate ProMP. ProMP model is regarded as a kind of Policy Representation in the framework of Imitation Learning (IL), so the parameter \( \omega \) could be learned from demonstrations. For each DoF, the parameter \( w_j \) can be computed by the least square method. The learned \( w_j \) is as follows

\[ w_j = (\Psi_{j,*}^T \Psi_{j,*} + \lambda)^{-1} \Psi_{j,*} q_{j,*}, \]

where \( \Psi_{j,*}^T = [\Psi_{j,1,T}^T, \ldots, \Psi_{j,J,T}^T]^T \in \mathbb{R}^{2T \times N} \) (\( T \) is the number of time steps, and \( T \) represents the transpose of the matrix), \( \lambda \) is the regularization term, and \( q_{j,*} = [q_{j,1}, \ldots, q_{j,T}]^T \in \mathbb{R}^{2T} \). Above all, we can get parameter \( \omega \) of ProMP.

**B. THE STRUCTURE OF IPROMP**

The purpose of ProMP is to build relationships between the weights of each basis function, but only one agent has been considered. Unlike ProMP, iProMP encompasses multiple agents such as the human and the robot, connecting each agent’s parameters. During the human-robot collaboration, it assumes that the human has \( P \) DoFs, and the robot...
has \( Q \) DoFs. The form of iProMP can be described as

\[
\hat{y}_t = \left[ (q^H_{1,t})^T, \ldots, (q^H_{P,t})^T, (q^R_{P+1,t})^T, \ldots, (q^R_{P+Q,t})^T \right]^T
\]

\[
= \left[ \begin{array}{cccc}
\Psi^T_{1,t} & \cdots & \Psi^T_{P,t} & \cdots & \Psi^T_{P+Q,t}
\end{array} \right]
\left[ \begin{array}{cccc}
\mathbf{w}^H_t & \cdots & \mathbf{w}^H_t & \cdots & \mathbf{w}^H_t
\end{array} \right]
+ \left[ \begin{array}{cccc}
\epsilon_1 & \cdots & \epsilon_1 & \cdots & \epsilon_1
\end{array} \right]
\]

\[
= \bar{H}_t^T \omega + \bar{\epsilon}_y,
\]

where the variables with superscript H represent the data of the human and those with superscript R represent the data of the robot. \( \bar{H}_t^T \) indicates \( \text{diag}(\Psi^T_{1,t}, \ldots, \Psi^T_{P+Q,t}) \), which is an observation matrix with \( (P + Q) \times (P + Q) \).

So the probability distribution of \( y_t \) can be given by

\[
p(y_t) = \int p(y_t|\omega)p(\omega)d\omega
\]

\[
= \int \mathcal{N}(y_t|H_t^T \omega, \Sigma_y)\mathcal{N}(\omega|\mu_\omega, \Sigma_\omega)d\omega
\]

\[
= \mathcal{N}(y_t|H_t^T \mu_\omega, H_t^T \Sigma_\omega H_t + \Sigma_y).
\]

When \( y_t^* \) is observed with a variance \( \Sigma_\omega^* \), estimating the parameter of ProMP is to calculate the posterior probability distribution \( p(\omega|y_t^*) = \mathcal{N}(\omega|\mu_\omega^*, \Sigma_\omega^*) \) [16], and the estimation equation of \( \mu_\omega^* \) and \( \Sigma_\omega^* \) is shown below:

\[
\begin{align*}
G &= \Sigma_\omega H_t^T (\Sigma_\omega^* + H_t^T \Sigma_\omega H_t)^{-1} \\
\mu_\omega^* &= \mu_\omega + G(y_t^* - H_t^T \mu_\omega) \\
\Sigma_\omega^* &= \Sigma_\omega - G H_t^T \Sigma_\omega H_t
\end{align*}
\]

where \( G \) is the Kalman Gain, and \( H_t^\text{obs} \) is distinguished between the observable part of a human and the unobservable part of a robot.

\[
(\hat{H}_t^\text{obs})^T = \left[ \begin{array}{cccc}
0 & \cdots & 0 & 0^R
\end{array} \right]
\]

In the process of collaboration, when the human state is observed, we can estimate the expectation of parameter \( \mu_\omega^*, \) and then calculate the state of the robot, which is utilized to drive the robot.

### III. A MIXTURE OF MOVEMENT PRIMITIVES

In this section, we first introduce the defects of ProMP and iProMP. To overcome these defects, we then introduce a mixture of MP models, applying it to ProMP and iProMP, respectively.
A. DEFECTS OF PROMP AND IPROMP
The parameters of both ProMP and iProMP are learned from measurable demonstrations. Empirically, when the variance of demonstrations is too small or large, it is difficult to extract effective features from demonstrations. If the pre-collected demonstrations are close to each other and have a small variance, the demonstrated skill can be learned well, but it is challenging to generalize to the skills outside the training set. If the demonstrations are scattered and have a large deviation, the previous two models (ProMP and iProMP) cannot learn the hidden skills. The explicit result exhibits that the robot cannot pass the set via points accurately with those demonstrations.

One of the trajectory planning experiments in ProMP shows the effect of demonstrations’ variance in Fig. 1. In Fig. 1 and the following figures of joint angles, the x-axis represents time steps, each time step represents 0.05 seconds, and the y-axis represents the joint angle, expressed in radians. After learning from demonstrations, we set four via points denoted by blue dots. The new task is to make the trajectory pass these points on time. As expected, the solid black trajectory intersects all the via points. At time step 80, the via point is set outside the variance of demonstrations, which causes the entire planned trajectory to deviate far from the training set. When this planned trajectory is utilized to guide the robot, it leads to a wide range of joint position fluctuations. This phenomenon means the robot is either consuming too much energy or being confronted with the crisis of machine wear. This experiment shows the poor generalization ability of the ProMP method at low variance. The same problem exists in iProMP.

![Image](image_url)

**FIGURE 1.** The trajectory of ProMP with one joint passing four via points, one of which is outside the variance of demonstrations.

B. MTIPROMP AND MTIPROMP
In order to overcome the influence of demonstration variance, we introduce a mixture of MP. Taking the ProMP model as an example, we called the mixture of ProMP as MTProMP (multi-task ProMP), which is weighted by multiple ProMPs, and each ProMP represents one task. During the demonstration learning phase, the Gaussian Mixture Model instead of Gaussian distribution is utilized to model the parameter $\omega$. Each component of GMM represents a single ProMP. Suppose that MTProMP consists of $K$ ProMPs, so the distribution of $\omega$ is

$$p(\omega) = \sum_{k=1}^{K} p(z_k)p(\omega|z_k) = \sum_{k=1}^{K} a(k)N(\omega|\mu(k), \Sigma(k))$$

where $z$ is a $K$ dimensional binary random variable, having a 1-of-$K$ representation in which a particular $z_k$ is equal to 1, and all other elements are equal to 0. $a(k) = p(z_k = 1)$ is the weight of $k$th ProMP, $\mu(k)$ and $\Sigma(k)$ are the mean and variance of $k$th ProMP’s parameter. Thus, the purpose of the demonstration learning phase is to train a GMM model of $\omega$.

In the single ProMP case, we estimate the initial parameter with equation (6). However, in MTProMP, the EM algorithm can be an appropriate tool to estimate the parameters. During the E-step of the training phase, we introduce $\tau_{ik}$ to represent the probability that $i$th demonstration belongs to the $k$th component of GMM, also called responsibility. The derived equation of $\tau_{ik}$ is as follows

$$\tau_{ik} = p(k|\omega_i) = \frac{a(k)N(\omega_i|\mu(k), \Sigma(k))}{\sum_{j=1}^{K} a(j)N(\omega_i|\mu(j), \Sigma(j))}$$

(15)

During the M-step of the training process, we update the parameters of the GMM

$$\begin{align*}
L^{(k)} &= \sum_{i=1}^{M} \tau_{ik} \\
a^{(k)} &= \frac{1}{M}L^{(k)} \\
\mu^{(k)} &= \frac{1}{M \tau^{(k)}} \sum_{i=1}^{M} \tau_{ik} \omega_i \\
\Sigma^{(k)} &= \frac{1}{M \tau^{(k)}} \left( \sum_{i=1}^{M} \tau_{ik} (\omega_i - \mu^{(k)})(\omega_i - \mu^{(k)})^T \right).
\end{align*}$$

(16)

We can obtain the parameters $\omega$ of the GMM model by performing E-step and M-step iteratively until convergence. The GMM model of $\omega$ divides the original ProMP into $K$ classes, and each sub-ProMP has its weight $a(k)$, mean $\mu(k)$, and covariance $\Sigma(k)$. To summarize, the structure of MTProMP can be described as

$$y_t = \text{MTProMP}(a^{(1:K)}, \omega^{(1:K)}) = \sum_{k=1}^{K} a(k)\text{ProMP}_k(\omega^{(k)})$$

(17)

where $a^{(1:K)} = [a^{(1)}, a^{(2)}, \ldots, a^{(K)}]$ and $\omega^{(1:K)} = [\omega^{(1)}, \omega^{(2)}, \ldots, \omega^{(K)}]$.

As shown in equation (2), the state $y_t$ is completely governed by parameter $\omega$ when the basis function matrix $H^T$ is set in advance. We can consider that one $\omega$ uniquely determines a ProMP. In MTProMP, it categorizes the original demonstrations into $K$ classes by clustering the parameter $\omega$ using the EM method. In the multi-task case, each class
represents a unique task. When the variance of demonstrations is large in one task, each class represents a cluster of a small part of demonstrations with an appropriate covariance. In this way, richer demonstrations could be beneficial to construct the probabilistic model of the robot. The probabilistic graphical model of MTProMP is shown in Fig. 2. For every data point \( \omega_m \), there is a corresponding latent variable \( z_m \).

\[
\begin{align*}
\bar{y}_t &= \text{MTiProMP}(a^{(1:K)}, \omega^{(1:K)}) \\
&= \sum_{k=1}^{K} a^{(k)} \times \text{iProMP}_k(\omega^{(k)}).
\end{align*}
\]

![FIGURE 2. Probabilistic graphical model of MTProMP.](image)

Similar to MTProMP, the strategy of mixture of MP model could be applied to iProMP. We call the new structure as MTiProMP (multi-task iProMP). The structure of MTiProMP can be described as

\[
\tilde{y}_t = \text{MTiProMP}(a^{(1:K)}, \omega^{(1:K)}) = \sum_{k=1}^{K} a^{(k)} \times \text{iProMP}_k(\omega^{(k)}).
\]

Referring to the relationship between ProMP and iProMP in Section II, the method to learn the parameters \( a^{(k)} \) and \( \omega^{(k)} \) in MTiProMP in the demonstration learning phase can be migrated from MTProMP.

IV. PARAMETERS ESTIMATION BASED ON ALTERNATE LEARNING AND DECOMPOSITION STRATEGY WITH PROJECTION POINTS

In this section, we propose an alternate learning method and decomposition strategy to estimate parameters of MTProMP and MTiProMP in the parameter predicting phase. In the first step, we update the weight \( a^{(k)} \) in weight space and update the parameter \( \omega^{(k)} \) in parameter space of each ProMP or iProMP to pass the decomposed projection points in the second step.

A. ESTIMATION OF PARAMETERS IN MTProMP

We first discuss MTProMP, and design a trajectory planning task experiment which makes the trajectory pass the specified via points. In Section II-C, with the assumption that \( \omega \) and \( y^p \) obey Gaussian distribution, ProMP and iProMP construct a new variable \( x = [\omega^T, y^p]^T \), which obeys Gaussian distribution. It can be treated as the joint Gaussian distribution. Via Gaussian inference, the posterior distribution \( p(\omega|y^p) \) could be computed. In MTProMP, the parameter \( \omega \) has been modeled with the GMM; that is, \( \omega \) is a combination of \( K \) \( \omega \)’s. Therefore, the hypothesis of joint Gaussian distribution does not exist. The estimation method cannot be transferred from ProMP.

According to equation (2), if the parameter \( \omega \) satisfies the GMM model, state \( y \), linearly transformed from \( \omega \), also obeys the GMM model. We denote the via point at time \( t \) as \( y^p_t \). The ultimate purpose of MTProMP in mathematical form is to maximize the probability \( p(y^p_t|a^{(1:K)}, \omega^{(1:K)}) \) by optimizing weight \( a^{(1:K)} \) and parameters \( \omega^{(1:K)} \) of each ProMP. In predicting phase, when the via points \( y^p_t \) are observed, most of the effort is to maximize the log-likelihood \( p(y^p_t|a^{(1:K)}, \omega^{(1:K)}) \) concerning \( \theta = [a^{(1:K)}, \omega^{(1:K)}] \).

\[
L(\theta) = \log p(y^p_t|a^{(1:K)}, \omega^{(1:K)}) = \sum_{k=1}^{K} a^{(k)} p(y^p_t|\omega^{(k)}).
\]

Intuitively, we apply an EM algorithm to estimate the parameters. We introduce a latent variable \( z = [z_1, z_2, \ldots, z_K]^T \), and each \( z_k \in \{0, 1\} \). The sum of \( z_k \) satisfies \( \sum_{k=1}^{K} z_k = 1 \), which means that only one \( z_k \) can take the value 1, others take the value 0. In addition, we use \( p(z_k = 1) = a^{(k)} \) to represent the probability of \( k \)th ProMP, abbreviated as \( q_k(z) \). Akin to the EM algorithm, we apply the Jenson Inequality for equation (19).

\[
L(\theta) = \sum_{k=1}^{K} q_k(z) \log a^{(k)} p(y^p_t|\omega^{(k)}) = \sum_{k=1}^{K} q_k(z) \frac{a^{(k)} p(y^p_t|\omega^{(k)})}{q_k(z)}.
\]

where we denote the lower bound of log-likelihood \( L(\theta) \) as \( Q(\theta) = \sum_{k=1}^{K} q_k(z) \log \frac{a^{(k)} p(y^p_t|\omega^{(k)})}{q_k(z)} \), and \( \sum_{k=1}^{K} q_k(z) = 1 \). When \( a^{(k)} p(y^p_t|\omega^{(k)}) = c \times q_k(z) \) holds, the inequality in equation (20) takes the equal sign, where \( c \) is a constant and \( c = \sum_{k=1}^{K} a^{(k)} p(y^p_t|\omega^{(k)}) \). Hence, the distribution \( q_k(z) \) of latent variable \( z \) is as follows

\[
q_k(z) = \frac{a^{(k)} p(y^p_t|\omega^{(k)})}{\sum_{j=1}^{K} a^{(j)} p(y^p_t|\omega^{(j)})}.
\]

In the EM algorithm, it is maximizing the lower bound that guarantees the convergence iteratively. The same strategy is equally effective here. To acquire the maximization of \( Q(\theta) \), we alternately optimize the weight \( a^{(1:K)} \) and the parameter \( \omega^{(1:K)} \) of ProMP. The optimization is shown in equation (22).

\[
\arg \max_{a^{(1:K)}, \omega^{(1:K)}} Q(\theta) = \arg \max_{a^{(1:K)}, \omega^{(1:K)}} \sum_{k=1}^{K} q_k(z) \log a^{(k)} p(y^p_t|\omega^{(k)}),
\]
where \( \sum_{k=1}^{K} q_k(z) \log q_k(z) \) is omitted for simplicity. Intuitively, equation (22) can be regarded as maximizing the expectation of \( \log a(k)p(y_j|\omega(k)) \) under the \( q_k(z) \).

In E-step, we compute the distribution \( q_k(z) \) of latent variable \( z \) with the equation (21), fixing weight \( a(k) \) and \( p(y_j|\omega(k)) \).

In M-step, there are some differences from the traditional EM algorithm. We first update the weight \( a(k) \) with the same equation in (16), yet here we only have one observed data point. So \( M \) is set to 1, and we get the update equation of the new weight \( a_{new}^{(k)} \).

\[
a_{new}^{(k)} = q_k(z).
\] (23)

Then, we update the parameter \( \omega(k) \) of each ProMP with fixed \( a(k) \) to maximize \( Q(\theta) \). It’s straightforward to reckon of forcing every ProMP to pass the via points, which makes each \( p(y_j|\omega(k)) \) most probable. Consequently, the weighted average expectation \( Q(\theta) \) could be guaranteed maximum with one iteration. Although this simple strategy is feasible, the requirements are too strict.

According to equation (17), \( y_j \) obeys the GMM model and can be seen as the weighted average of multiple ProMPs or tasks. From another perspective, if each ProMP is regarded as a basis function, the via points or observed \( y_j^* \) can be viewed as a point in the space supported by multiple ProMPs. Our current strategy is to let each ProMP pass the projection point \( y_j^* \) on each ProMP basis.

The most significant task is to pinpoint the projection points \( y_j^* \). In Section III-A, the drawback of a single ProMP is analyzed. Generally speaking, we hope that each projection point \( y_j^* \) is as close as possible to the mean \( (y_j^*(\text{mean}))^{(k)} \) of each ProMP. That is, under the premise of satisfying the sum of weighted projection passing the set via point, the smaller the cumulative sum of the deviation from projection point \( y_j^*(k) \) to the mean \( (y_j^*(\text{mean}))^{(k)} \), the better decomposition strategy it is. Each ProMP just needs to pass the corresponding projection point \( y_j^*(k) \), which is close to its mean and has a high probability that it falls within the variance range. Fig. 3 shows a schematic diagram of the proposed method, the y-axis does not represent the joint angle. Red X-shape dots are the projection point of the blue via point, and they are close to the corresponding black mean point. This method ensures that each ProMP has a good, smooth planned trajectory with small fluctuation. As a result, the weighted trajectory is smooth when the via point is set between two tasks. In this way, smooth task switching is achieved attained.

As for the scenarios where \( V \) via points are required, without loss a generality, a \( V \)-dimensional key-value dictionary \( D_V \) presented as

\[
D_V = \left[ (t_1, y_j^{(\text{mean})}), \ldots, (t_V, y_j^{(\text{mean})}) \right]
\]
is proposed. Searching for the best projection point \( y_j^{(k)} \) over \( (t_j, y_j^{(\text{mean})}) \) on the \( k \)th ProMP, we define an objective function with an equality constraint to minimize the total variation of \( y_j^{(k)} \) and \( (y_j^{(mean)})^{(k)} \). The objective function is expressed as

\[
\mathcal{J}_j = \sum_{k=1}^{K} ||y_j^{(k)} - (y_j^{(mean)})^{(k)}||^2 \quad \text{s.t.} \quad y_j^* = \sum_{k=1}^{K} a_j^{(k)} y_j^{(k)},
\] (24)

where we utilize 2-Norm to measure the variation from the projection point \( y_j^{(k)} \) to each ProMP’s mean trajectory.

Three new variables coefficient \( a_j \), expected via point \( Y_j \) and experiential mean point \( (y_j^{(mean)}) \) at time point \( t \) are introduced.

\[
a_j = \begin{bmatrix} a_j^{(1)} \\ \vdots \\ a_j^{(K)} \end{bmatrix}, \quad Y_j = \begin{bmatrix} y_j^{(1)} \\ \vdots \\ y_j^{(K)} \end{bmatrix}, \quad (y_j^{(mean)}) = \begin{bmatrix} (y_j^{(mean)})^{(1)} \\ \vdots \\ (y_j^{(mean)})^{(K)} \end{bmatrix}.
\] (25)

Therefore, the equation (24) could be rewritten as

\[
\mathcal{J}_j = ||Y_j - (y_j^{(mean)})||^2 \quad \text{s.t.} \quad y_j^* = a_j^T Y_j.
\] (26)

Now, equation (24) becomes an optimization problem with equality constraints. We form the Lagrangian equation \( L(Y_j) \) with the Lagrangian multiplier \( \lambda_j \).

\[
L(Y_j) = ||Y_j - (y_j^{(mean)})||^2 + \lambda_j ||a_j^T Y_j - y_j^*||^2.
\] (27)

Our ultimate purpose is to find \( Y_j^* = \arg\min_{Y_j} L(Y_j) \). According to the characteristics of equation (27), let \( \partial L(Y_j)/\partial Y_j = 0 \), then we get \( Y_j^* \).

\[
Y_j^* = (I + \lambda_j a_j a_j^T)^{-1} (y_j^{(mean)}) + \lambda_j a_j y_j^*.
\] (28)

Previously we discussed from the perspective of multiple components of GMM at the same time point. Next, we discuss from the perspective of multiple time points along with one component of GMM. In this way, we update parameter \( \omega(k) \) to make \( K \)th ProMP pass \( y_j^{(k)} \) (\( V \) via points), one component

FIGURE 3. Schematic diagram of projection point method.
of $Y^*_{t:v}$. In contrast to equation (9), the update equation is listed as follows

$$\begin{align*}
G^{(k)}_{t:v} &= \Sigma \omega^{(k)}_t \bar{H}^T_{t:v} \left( \Sigma_y + H^T_{t:v} \Sigma \omega^{(k)}_t H_{t:v} \right)^{-1} \\
\mu_{\omega^{(k)}_t} &= \Sigma \omega^{(k)}_t \bar{y}^{(k)}_{t:v} - H^T_{t:v} \Sigma \omega^{(k)}_t \mu_{\omega^{(k)}_t} \\
\Sigma_{\omega^{(k)}_t} &= \Sigma \omega^{(k)}_t - G^{(k)}_{t:v} H^T_{t:v} \Sigma \omega^{(k)}_t.
\end{align*}$$

(29)

Different from the EM algorithm, we execute only one iteration for the speed of the algorithm, which will not exacerbate the computational burden. The later experiments demonstrate relatively good performance.

Since the coefficient $a_{ij}$ ($j \in \{1, \ldots, V\}$) are well designed at several specific via points, and they are generally different. For the smoothness of the weighted trajectory and avoidance of the sudden change, a linear interpolation strategy is utilized. For two or more tasks, the linear interpolation strategy can easily ensure the sum of weights of different tasks equals 1. Next, we will prove that $\sum_{k=1}^{K} a^{(k)} = 1$ at any time step $t$.

Since there are $V$ via points denoted as $D_V$, without loss of generality, we can select two consecutive via points at time $t_f$ and $t_{f+1}$. Then we can get the associated wights $a^{(k)}_{ij}$ and $a^{(k)}_{ij}$ of $k$th ProMP, respectively. After the use of linear interpolation between step $b$ and $c$, we assume time step $t \in (b, c)$, and get $a^{(k)}_{ij}$ with the equation

$$a^{(k)}_{ij} = a^{(k)}_{ij} + \frac{a^{(k)}_{ij} - a^{(k)}_{ij}}{c-b}(t-b).$$

So, the sum of $a^{(k)}_{ij}$ is

$$\sum_{k=1}^{K} a^{(k)}_{ij} = \sum_{k=1}^{K} \left( a^{(k)}_{ij} + \frac{a^{(k)}_{ij} - a^{(k)}_{ij}}{c-b}(t-b) \right) = \sum_{k=1}^{K} a^{(k)}_{ij} + \frac{t-b}{c-b} \left( \sum_{k=1}^{K} a^{(k)}_{ij} - \sum_{k=1}^{K} a^{(k)}_{ij} \right) = 1,$$

where $\sum_{k=1}^{K} a^{(k)}_{ij}$ and $\sum_{k=1}^{K} a^{(k)}_{ij}$ all equal 1.

**B. ESTIMATION OF PARAMETERS IN MTiProMP**

We have discussed the parameter estimation of MTProMP in the last section. In this section, we discuss MTiProMP. According to Section III-B and equation (18), The strategy of parameter estimation in MTiProMP can be migrated from MTProMP.

The EM algorithm is also utilized here. Different from MTProMP, MTiProMP predicts the robot’s state according to the observed human state, which is also called human behavioral intention. The human state is a part of the variable $y_j$. In E-step, we calculate the distribution $q_k(z)$ of latent variable $z$

$$q_k(z) = \frac{a^{(k)} \pi(z) \omega^{(k)}_t}{\sum_{j=1}^{K} a^{(j)} \pi(z) \omega^{(j)}_t}. \quad (30)$$

In M-step, we first estimate the weight $a^{(k)}$ of each iProMP with respect to human behavioral intention. Like equation (21), $a^{(k)}$ is equal to $q_k(z)$. Let’s assume the observed state $\bar{y}^{(k)}_t$ equals $[y^H_t, y^R_t]^T$, where $y^H_t = \left[ y_t^H, \ldots, y_t^H \right]^T$ and $y^R_t = \left[ y_t^R, \ldots, y_t^R \right]^T$.

We then decompose $y^H_t$ to each iProMP, which is denoted as $y^{(k)}_t$. The part of robot state remains zero vector because it is unobservable. New observation state for each iProMP is $\bar{y}^{(k)}_t = \left[ (y^{(k)}_t)^H, (y^{(k)}_t)^R \right]^T$. The predicting equation for parameters $\mu_{\omega^{(k)}_t}$ and $\Sigma_{\omega^{(k)}_t}$ in each iProMP is as follows

$$\begin{align*}
G^{(k)} &= \Sigma \omega^{(k)} \bar{H}^{(obs)} \left( \Sigma_y + \bar{H}^{(obs)} \Sigma \omega^{(k)} \bar{H}^{(obs)} \right)^{-1} \\
\mu_{\omega^{(k)}_t} &= \Sigma \omega^{(k)} \bar{y}^{(k)} - \bar{H}^{(obs)} \Sigma \omega^{(k)} \mu_{\omega^{(k)}_t} \\
\Sigma_{\omega^{(k)}_t} &= \Sigma \omega^{(k)} - G^{(k)} \bar{H}^{(obs)} \Sigma \omega^{(k)}.
\end{align*}$$

(31)

**V. EXPERIMENTS**

Several via points experiments and human-robot experiments were conducted with a 7 DoFs Sawyer robot and its camera. In the first experiment, we applied MTProMP to conduct the task, which let the robot arm to pass the set via points in joint space at the specified time. This experiment show that MTProMP has the capability to switch from task A to task B smoothly. In the second experiment, we apply MTiProMP to perform two human-robot collaboration tasks at the same time. This experiment shows that the robot switches from task A to task B when human switches behavioral intentions from task A to task B.

In all the experiments in this paper, we sample 100 time steps with a time interval of 0.05 seconds.

**A. PASSING SET VIA POINTS WITH MTProMP**

For simplicity, we first apply MTProMP to one joint to validate the feasibility. The experiment duration is 100 time steps. In Fig. 4, the light red and light green lines represent the demonstrations of task 1 and task 2 respectively, and each task has 26 demonstrations. Red and blue lines represent the mean trajectories of demonstrations. The blue dots represent the set via points at time step 1, 30, 70, 80, 100. The solid lines represent the mean trajectory for each task, and the dash lines represent the planned trajectories with ProMP for each task. Obviously, we can see that the planned trajectories have a very large fluctuation range, which is not good for robots.

Fig. 5 to Fig. 7 show the trajectory planning process of MTProMP. The black dots represent the mean joint angles of each task learned from demonstrations at time step $t$. In Fig. 5, we can see that the blue dot via point at time step 30 is decomposed into two pink X-shape dots, and the black dash weighted average trajectory passes the blue via point successfully. In Fig. 6, the blue dot coincides with one of the pink dots at time step 100. The reason for this situation is that the blue dot is in the demonstrations of task 1 and far from task 2, so the weight $a^{(1)}$ approaches 1, and $a^{(2)}$ approaches 0, which can be clearly seen in Fig. 7. The red and green dots represent the weights of two tasks corresponding to the via points at the same time step, respectively. The weight is only updated at
several specific moments. For the smoothness of the weighted trajectory and avoidance of the sudden change, a linear interpolation strategy is utilized to fit the weight curve. For two or more tasks, the linear interpolation strategy can ensure the sum of weights of different tasks equals 1. In summary, with the set via points ranging from task 2 to task 1, the weight $a^{(k)}$ changes, and the main contributor of passing blue via points switches from task 2 to task 1. Comparing the y-axis coordinates of several figures, we notice that the black dash weighted trajectory has a smaller fluctuation range.

**FIGURE 4.** Trajectories of two ProMPs with one joint.

We also apply MTProMP to the 7-DoF Sawyer robot to conduct the task of passing via points. In Fig. 8, the joints modeled by ProMP cannot pass the two set via–points completely. However, in Fig. 9, the black dash planned trajectories pass all set via points at time step 40 and 100 successfully, which validates the performance of MTProMP. MTProMP eliminates the impact of huge variance brought by demonstrations.

**B. TASK-SWITCHING BASED ON HUMAN BEHAVIORAL INTENTION WITH MTiProMP**

In multi-task human-robot collaboration experiments, we adopt the same strategy as section V-A. For simplicity, we first consider the one joint circumstance. Fig. 10 and Fig. 11 show the human-robot collaboration with one joint. In Fig. 10, human first executes task 2 and then start to switch from task 2 to task 1 at time step 35. Finally, human executes task 1 at time step 70. In MTiProMP, the robot trajectory is predicted according to the observed human trajectory. The predicted robot trajectory is shown in Fig. 11. The robot first executes along the green trajectory of task 2 and then switches to the red trajectory of task 1. The task-switching process is steady and smooth, which preliminarily proves the feasibility of MTiProMP in multi-task human-robot collaboration and task switching.

**FIGURE 5.** Trajectories of MTProMP with one joint passing via point at time step 30.

**FIGURE 6.** Trajectories of MTProMP with one joint passing via point at time step 100.

Fig. 12 to Fig. 14 depict the experiment that a human does the collaboration task with a 7-DoF Sawyer robot. Fig. 18 demonstrates the real Sawyer robot in collaboration. Here we use the Cartesian coordinates of the collaborator’s hand as mentioned in section II-A. In the demonstration learning phase, all demonstrations are collected from the collaboration
of the real Sawyer and the human. Then, we estimate the parameters of MTiProMP. At the prediction phase, we ask the human collaborator to switch from task 2 to task 1. It should be noted that human behavioral intention is limited to the demonstrations. Fig. 12 shows the coordinates of the human hand in Cartesian space, which are estimated by the
FIGURE 13. Robot trajectories switch from task 2 to task 1 in Joint space with several iProMPs.

FIGURE 14. Robot trajectories switch from task 2 to task 1 in joint space with MTiProMP.

FIGURE 15. Human and Robot trajectories switch from task 2 to task 1 in Cartesian space with several iProMPs.

FIGURE 16. Human and Robot trajectories switch from task 2 to task 1 in Cartesian space with MTiProMP.

FIGURE 17. Sawyer robot in CoppeliaSim with MTiProMP.

FIGURE 18. Real Sawyer robot with MTiProMP.

Fig 13 (method proposed in [12]) and Fig. 14 (method proposed in this paper) illustrate the angle of each joint in the robot’s joint space. Task 1 and task 2 represent the different directions of hand and robot end-effector movements, which share the same initial state. The setting of task 1 and task 2 are shown in Fig. 17 and Fig. 18. Fig. 17 show the simulation of Sawyer in CoppeliaSim, where the Sawyer robot smoothly switches from task 2 to task 1. Fig. 19 presents the movement of the camera of Sawyer. The change of hand position represents the change in the behavioral intention of the collaborator.
FIGURE 19. The movement track of Sawyer robot with MTiProMP.

track of the real Sawyer robot’s end-effector, a plastic suction cup. The collaboration is that the human hand and the robot end-effector move from respective initial position to the same target position. We can see that the actual human trajectory switches from task 2 to task 1 in Cartesian space, the corresponding task-switching of the robot is reflected in the robot’s joint space. The trajectories of robot joint angles are inferred based on the observed human behavioral intention. The trajectories in Fig. 13 have a sudden shift with the switching from task 2 to task 1, which can be considered as a hard switch. Fig. 15 and 16 show the Cartesian coordinates of robot end-effector and human hand. The actual robot trajectory in Fig. 15 has a straight part, the duration of this part is one time step. In a real robot, this is intolerable. However, the trajectory inferred with MTiProMP is smooth, realizing the smooth task-switching.

Combining the above two experiments, we evaluate the performance of MTiProMP in multi-task human-robot collaboration and task switching based on human behavioral intention.

VI. CONCLUSION

We presented MTProMP model for multi-task robot manipulation and MTiProMP model for multi-task human-robot collaboration. We also proposed an alternative learning method and decomposition strategy with projection points, combining with linear interpolation strategy for weights to estimate the weight and parameters in the previous two models. We showed that MTProMP model could effectively overcome the defects of ProMP with a huge variance or two or more tasks, and MTiProMP model is highly competent for task switching in human-robot collaboration. Classic via points experiments and human-robot collaboration experiments with the Sawyer robot have been carried out to prove the feasibility of proposed models and methods.

There are a few interesting future research directions along with this topic. First, In our experiments, a large amount of demonstrations are needed to estimate the parameters of MTProMP and MTiProMP. To improve the data efficiency, we could integrate the idea proposed in [17] with proposed algorithms. A Normal-Wishart prior and a Dirichlet prior will be employed to provide the prior information and reduce the number of the demonstrations. Second, our current innovation in this paper focuses on aspects of the ideas and theory. As for the practical operation, ordinal damped least squares and limiting functions are used for robot singularity avoidance and safe output of physical mechanisms. Although it works well, these are passive implementations, and the active optimization with our proposed algorithms by incorporating them with other restrictions will be further refined in subsequent studies.

REFERENCES

[1] S. Haddadin and E. Croft, “Physical human–robot interaction,” in Springer Handbook of Robotics. Berlin, Germany: Springer, 2016, pp. 1835–1874.
[2] J. Lee, “A survey of robot learning from demonstrations for human–robot collaboration,” 2017, arXiv:1710.08789. [Online]. Available: http://arxiv.org/abs/1710.08789
[3] F. J. Abu-Dakka and V. Kyrki, “Geometry-aware dynamic movement primitives,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2020, pp. 4421–4426.
[4] N. Wang, C. Chen, and C. Yang, “A robot learning framework based on adaptive admittance control and generalizable motion modeling with neural network controller,” Neurocomputing, vol. 390, pp. 260–267, May 2020.
[5] L. Peternel, N. Tsagarakis, D. Caldwell, and A. Ajoudani, “Robot adaptation to human physical fatigue in human–robot co-manipulation,” Auto. Robots, vol. 42, no. 5, pp. 1011–1021, Jun. 2018.
[6] A. Parachos, C. Daniel, J. Peters, and G. Neumann, “Probabilistic movement primitives,” in Proc. Adv. Neural Inf. Process. Syst., 2013, pp. 2616–2624.
[7] G. Maeda, M. Ewerton, R. Lioutikov, H. B. Amor, J. Peters, and G. Neumann, “Learning interaction for collaborative tasks with probabilistic movement primitives,” in Proc. IEEE-RAS Int. Conf. Humanoid Robots, Nov. 2014, pp. 527–534.
[8] G. Maeda, G. Neumann, M. Ewerton, R. Lioutikov, and J. Peters, A Probabilistic Framework for Semi-Autonomous Robots Based on Interaction Primitives With Phase Estimation. Cham, Switzerland: Springer, 2018, pp. 253–268.
[9] M. Khoramshahi and A. Billard, “A dynamical system approach to task-adaptation in physical human–robot interaction,” Auton. Robots, vol. 43, pp. 927–946, Apr. 2019.
[10] K. Suzuki, H. Mori, and T. Ogata, “Motion switching with sensory and instruction signals by designing dynamical systems using deep neural network,” IEEE Robot. Autom. Lett., vol. 3, no. 4, pp. 3481–3488, Oct. 2018.
[11] A. Mohseni-Kabir and M. Veloso, “Robot task interruption by learning to switch among multiple models,” in Proc. 27th Int. Joint Conf. Artif. Intell., Jul. 2018, pp. 4943–4949.
[12] G. J. Maeda, G. Neumann, M. Ewerton, R. Lioutikov, O. Kroener, and J. Peters, “Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks,” Auto. Robots, vol. 41, no. 3, pp. 593–612, Mar. 2017.
[13] E. Rueckert, J. Mundo, A. Parachos, J. Peters, and G. Neumann, “Extracting low-dimensional control variables for movement primitives,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2015, pp. 1511–1518.
[14] C. M. Bishop, Pattern Recognition and Machine Learning. New York, NY, USA: Springer, 2006.
[15] A. Parachos, E. Rueckert, J. Peters, and G. Neumann, “Probabilistic movement primitives under unknown system dynamics,” Adv. Robot., vol. 32, no. 6, pp. 297–310, Mar. 2018.
[16] T. D. Barfoot, State Estimation for Robotics. Cambridge, U.K.: Cambridge Univ. Press, 2017.
[17] T. Kulak, H. Girgin, J.-M. Odobez, and S. Calinon, “Active learning of Bayesian probabilistic movement primitives,” IEEE Robot. Autom. Lett., vol. 6, no. 2, pp. 2163–2170, Apr. 2021.
JIAN FU (Member, IEEE) received the M.Eng. degree in computer application from the Huazhong University of Science and Technology, Wuhan, China, in 1999, and the Ph.D. degree in control theory and control engineering from the University of Science and Technology Beijing, Beijing, China, in 2006. He is currently an Associate Professor with the School of Automation, Wuhan University of Technology. His current research interests include human–robot interaction, intelligent robot, and reinforcement learning.

JINYU DU received the B.S. degree from Wuhan University of Technology, China, in 2018, where he is currently pursuing the master’s degree with the Department of Control Science and Engineering. His research interests include reinforcement learning, human–robot interaction, and data-efficient robot learning.

XIAO TENG received the B.S. degree from East China University of Technology, China, in 2017, and the master’s degree from Wuhan University of Technology, China, in 2020. His research interests include deep learning and human–robot interaction.

LU WU received the M.Eng. degree in circuit and system and the Ph.D. degree in communication and information system from Wuhan University of Technology, Wuhan, China, in 2007 and 2018, respectively. She is currently a Lecturer with the School of Information Engineering, Wuhan University of Technology. Her research interests include machine learning and computer vision.

YUXIANG FU is currently pursuing the B.S. degree in computer science with The University of British Columbia, Vancouver, BC, Canada. His current research interests include machine learning and algorithmic analysis.

* * *