2018 26th Mediterranean Conference on Control and Automation (MED)
Zadar, Croatia, June 19-22, 2018

Human-robot Interaction Control through Demonstration

Shangke Lyu and Chien Chern Cheah

Abstract— **In human-robot interaction (HRI) tasks, robots are required to change the behaviours according to different applications. As different interaction behaviours require different task requirements, it is difficult to formulate a HRI control problem in a general or unified way that enables robots to learn a set of task requirements or behaviours through human's demonstrations and then execute the tasks by using one controller . This paper aims to solve this problem by using a dynamic potential energy function to describe a set of different task requirements so that the motion behaviours demonstrated by human can be acquired or learned by the robot systems in a unified way. A control strategy is proposed to enable the robots to perform various tasks demonstrated by human and also change the behaviours during HRI according to different applications in a stable manner. The stability of overall system is shown using Lyapunov-like analysis and experimental results are presented to illustrate the performance of the proposed control strategy.**

I. Introduction

In past decades, robotic technologies have been widely used in factory automation to improve efficiency and productivity. In a structured factory environment, most tasks can be specified in the form of desired trajectories and robots are strictly isolated from human for safety reasons. With the development of computing and sensing technologies, it is envisioned that robots would be able to share workspace with human to perform more complex tasks.

Generally, robots can perform tasks more precisely and stably than human, while human can adapt to various situations and make decisions accordingly. Therefore, human-robot interaction (HRI) can combine the strengths of both human and robot to further explore applications in other emerging areas such as military [1][2], healthcare [3][4], intelligent manufacturing [5][6]. In [7], a control strategy was developed for the collaboration of human and robot when in physical contacts. To enhance the capabilities of robot in reacting to uncertain environment, human was introduced to guide the co-manipulation task in presence of unforeseen changes in the environment [8].

In traditional robot applications, the task requirements are usually specified in advance by users and it is difficult for the robot to learn and perform a new task by itself. It is anticipated that task specification through human's demonstration would be one of the effective ways for robots to understand or imitate human's behaviours, especially during HRI. In [9], the robot was excepted to repeat the tasks based on the observation of human behaviours. In [10], a structure support vector machine algorithm was used to learn the task requirements according to humans activities observed from recorded RGB-D videos. In [11], a learning algorithm was proposed to allow the robot to acquire new motor skills based on human's demonstrations.

Most existing results on task learning are limited to learning a specific motion. For HRI applications, the task requirements or robot behaviours may change according to different situations or environments. However, few work has been done to enable the robot to learn a set of tasks or behaviours using a single task function that allows the robot to smoothly change its behaviours during HRI. An algorithm was proposed in [12] to learn the priorities of multiple tasks, but this method was mainly used in learning the weightage of each pre-defined task so as to switch from one task to another, rather than the task requirements. In [13], a single task function was used to define various HRI tasks by setting various task parameters but the task parameters were predefined rather than obtaining from demonstrations. In addition, it was in general difficult for users to set these task parameters as they may not correspond to any physical meaning.

In this paper, a control strategy is developed to enable the robot systems to learn a set of HRI tasks in a unified way through human's demonstrations. A dynamic potential energy function is used to describe different task requirements and the task parameters can be automatically obtained from human's demonstrations. An adaptive controller is proposed, which allows the robot to perform similar tasks demonstrated by human. Besides, it can also change the robot behaviours during HRI according to different human's intentions in a stable manner. In addition, rather than limiting to a single behaviour for a specific task as in [13], we also show that the various behaviours acquired through demonstrations can be combined in a sequential way to perform more complex tasks. The stability of overall system is shown using Lyapunov-like analysis and experimental results are presented to illustrate the performance of the proposed method.

II. Task learning through human's demonstration

In this section, we propose a task learning algorithm which allows the users to specify the tasks through demonstrations.

A. General task function

Generally, different task requirements lead to different potential energy function specifications and the corresponding shapes of the potential energy functions in the task-space are different as shown in Fig.1. Each shape corresponds to one task and results in one kind of robot behaviour.

S. Lyu and C. C. Cheah are with school of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798.

To describe various task requirements in a general way such that the shape of the potential energy function can be varied by simply adjusting some parameters, a task function is defined as [13][14]:

$$y(z) = S(1 - e^{-a_1(\|z\|^2 - d_1^2)})(1 - e^{-a_2(\|z\|^2 - d_2^2)}) \tag{1}$$

where $z \in \Re^n$ is a task variable, S, a_1, a_2, d_1, d_2 are task parameters. A dynamic potential function is defined as:

$$P = max[0, y(z)]^N \tag{2}$$

where $N \geq 3$ so that the potential function is at least twice differentiable. Different shapes of the dynamic potential function can be formed by adjusting the values of the task parameters S, a_1, a_2, d_1, d_2. The gradient of dynamic potential energy function $\frac{\vartheta P}{\vartheta z}$ can be integrated into robotic systems to control the robot to perform different tasks.

Among these parameters, a_1, a_2 are set as greater than 0 and thus P is always saturated due to the property of exponential function. Note that the saturated P is specified in a way that the unsaturated part ($\frac{\vartheta P}{\vartheta z} \neq 0$) of P covers the entire robot workspace. The task parameters d_1, d_2, are used to determine the size of desired region and the parameter S is set either as 1 or -1, which determines the location of the desired position or region z_d where $\frac{\vartheta P}{\vartheta z} = 0$. If $S = 1$, the desired region z_d is located between d_1 and d_2 as shown in Fig.2(a). If $S = -1$, the location of desired region z_d is changed to less than d_1 and more than d_2 as shown in Fig.2(a). For the specific S such as $S = 1$, when d_1 and d_2 are set as 0, the desired region shrinks to a point as shown in Fig. 2(b), while when d_1 and d_2 are nonzero, the desired region is shifted to the middle between d_1 and d_2 as shown in Fig.2(b). Other situations are also illustrated in Fig. 2(b).

Through adjusting the task parameters, the shape of the proposed potential energy function can be varied as illustrated in Fig.1. Therefore, the dynamic potential energy function can be used to achieve different tasks. However, as some of the task parameters do not correspond to any physical meaning, it is difficult for users to specify them directly. To solve this problem, a learning algorithm is developed in next subsection, which allows the system to directly learn the task requirements from human's demonstrations.

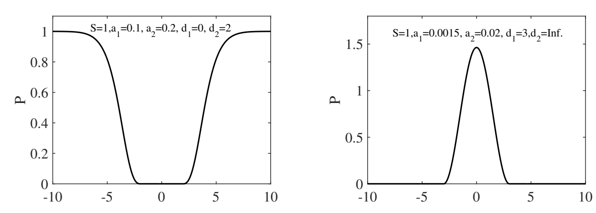
B. Reconstruction of task function from demonstrations

We present a method to estimate the shape of the potential energy based on human's demonstrations. Different motion behaviours can be described by shaping the dynamic potential energy function as shown in Fig.1 and the motion behaviour features or task requirements underlying the demonstrations can thus be obtained by reconstructing the shape of potential energy of the demonstration. Let x_i^t denote the i^{th} position information in the recorded training data when human manually moves the robot to demonstrate the task. To reconstruct the equivalent shape of potential energy in the demonstration, a task energy function P_d is defined as

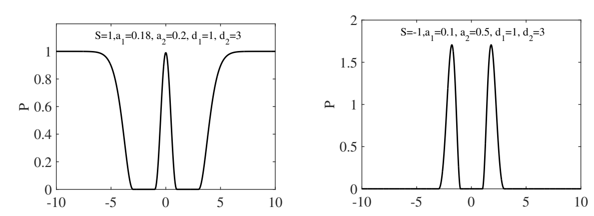
$$P_d(x_i^t - x_r^t) = \begin{cases} \|(x_i^t - x_r^t) - (x_d^t - x_r^t)\|^2 - T, & \|\dot{x}_i^t\| > 0 \\ 0, & \|\dot{x}_i^t\| = 0 \end{cases} \tag{3}$$

where $(i = 1, 2, \cdots, N)$, x_r^t is a fixed reference position and denotes the center of potential energy, x_d^t denotes the fixed target position, T is a constant value defined as $T = \min\{\|(x_i^t - x_r^t) - (x_d^t - x_r^t)\|^2\}$, $\dot{x}_i^t$ denotes the corresponding speed at position x_i^t and N denotes the number of recorded data. Therefore, when human is demonstrating a task such that $\|\dot{x}_i^t\| > 0$, the potential energy is calculated; once robots reach to the target position x_d^t and human stops moving the robot such that $\|\dot{x}_i^t\| = 0$. In the demonstration, the reference position x_r^t is arbitrarily set by human. Then, the target position x_d^t is the point where robots should finally reach and is automatically set as the position when human stops moving the robot. If the same demonstration is repeated, the average position can be used. Based on this method, the task requirements of demonstrations can be described in term of potential energy which can be learned by robots through the dynamic potential energy function.

For some tasks that are composed of two task requirements such as i)following the target and ii) keeping a safe distance, the entire task requirements can be taught to robots one by one. That is, human can teach the robot the basic motion behaviours first such as following behaviour or keeping a

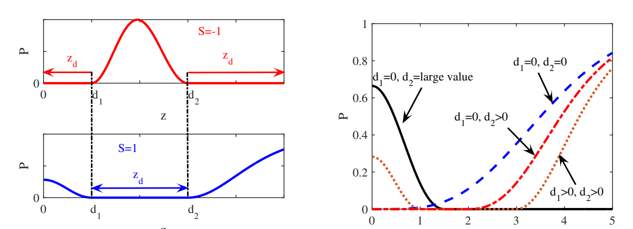


(a) Behaviour:tracking or following target (b) Behaviour:collision avoidance



(c) Behaviour:following target with a safe distance kept (d) Behaviour:trapping target

Fig. 1. Some examples of potential energy function for different tasks



(a) Locations of desired region with different S in first quadrant (b) Desired region types with different d_1 and d_2 for the case $S = 1$

Fig. 2. The effect of the task parameters S, d_1, d_2 on P

safe distance. After learning the basic motion behaviours respectively, robots can perform the task containing two basic motion behaviours by combining the corresponding task energy functions together, such as following the target with a safe distance kept as shown in Fig.3(d). Therefore, the overall energy function can be defined as:

$$P_d(x^t(i)-x^t_r)=P_{d_1}(x^t_{1_i}-x^t_{r_1})+P_{d_2}(x^t_{2_i}-x^t_{r_2}) \tag{4}$$

where $P_{d_1}(x^t_{1_i}-x^t_{r_1})$ is the task energy function for following demonstration, $P_{d_2}(x^t_{2_i}-x^t_{r_2})$ denotes the task energy function for keeping a safe distance demonstration, $x^t_{1_i}$ is the recorded data in following demonstration, $x^t_{2_i}$ is the recorded data in keeping a safe distance demonstration, $x^t_{r_1}$ is the reference position for following demonstration and is defined as $x^t_{r_1}=x^t_r+C_1$ and $x^t_{r_2}$ is the reference position for keeping a safe distance demonstration and is defined as $x^t_{r_2}=x^t_r+C_2$ and C_1,C_2 are constant value specified by users and used to shift the corresponding task energy function.

C. Task learning based on human's demonstration

The task requirements underlying the obtained task energy function P_d can be learned by the dynamic potential function P through seeking a suitable task parameter set such that the shapes of P_d and P in the task-space are similar. The task-requirement learning algorithm is composed of two parts: 1) parameter identification and 2) performance learning.

1) parameter identification: The parameters S,d_1,d_2 can determine the location and size of the desired region as shown in Fig.2 and different combinations of S,d_1,d_2 lead to different desired regions, resulting in different shapes of P and task types as shown in Fig.1. Therefore, by identifying S,d_1 and d_2 based on P_d, the task type of human's demonstrations can be determined.

Among them, the parameters d_1 and d_2 denote the boundaries of the desired region as shown in Fig.2. For the data x^t inside the desired region of P_d, there are two properties:i) its corresponding speed $\dot{x}^t$ is equal to 0 and ii) the potential energy P_d at x^t is equal to 0. Hence, the desired region Ω of P_d can be denoted as $\Omega=\{x^t \mid \dot{x}^t=\mathbf{0} \ and \ P_d(x^t-x^t_r)=0\}$. The boundaries of desired region, d_1 and d_2, can be determined as the minimum and maximum distances around the reference position as follow:

$$\begin{aligned} d_1&=min(\|x^t_k-x^t_r\|,\|x^t_{k+1}-x^t_r\|,\cdots,\|x^t_N-x^t_r\|),\\ d_2&=max(\|x^t_k-x^t_r\|,\|x^t_{k+1}-x^t_r\|,\cdots,\|x^t_N-x^t_r\|), \end{aligned} \tag{5}$$

where $\{x^t_k,x^t_{k+1},\cdots,x^t_N\}\in\Omega$. After specifying d_1 and d_2, the value of S can be determined by comparing the Euclidean distance between the normalized dynamic potential function $\frac{P(x^t-x^t_r)}{\|P(x^t-x^t_r)\|}$ and normalized task energy function $\frac{P_d(x^t-x^t_r)}{\|P_d(x^t-x^t_r)\|}$ for all $x^t\in\Omega$. Different values of S (i.e. 1 or -1) result in different locations of desired region as shown in Fig.2(a). It is straightforward that if the normalized dynamic potential function $\frac{P(x^t-x^t_r)}{\|P(x^t-x^t_r)\|}$ has the same location of desired region with the normalized task energy function $\frac{P_d(x^t-x^t_r)}{\|P_d(x^t-x^t_r)\|}$, the Euclidean distance between them is always less than the case when they have different desired region locations. Thus, the parameter S can be specified as:

$$S= \tag{6}$$

$$\begin{cases} 1, \ if \ \|\frac{P_d(x^t-x^t_r)}{\|P_d(x^t-x^t_r)\|}-\frac{P_1(x^t-x^t_r)}{\|P_1(x^t-x^t_r)\|}\|<\|\frac{P_d(x^t-x^t_r)}{\|P_d(x^t-x^t_r)\|}-\frac{P_{-1}(x^t-x^t_r)}{\|P_{-1}(x^t-x^t_r)\|}\|, \\ \quad \text{for all } x^t\in\Omega \\ -1, else, \end{cases}$$

where $P_1(x^t-x^t_r)$ and $P_{-1}(x^t-x^t_r)$ denote the dynamic potential functions with $S=1$ and $S=-1$ respectively.

Once the parameters S,d_1,d_2 are determined, the task type is determined. Manually adjusting d_1 and d_2 can enable robots to do the similar task with different desired region sizes. For example, when robots are taught to follow a target, by changing d_1 and d_2, users can adjust the following distance between the target and robots.

2) Performance learning: The remaining task parameters a_1,a_2 are used to adjust the gradient of the dynamic potential function to be steep or gradual according to the speed when human moves robots in the demonstration. More specifically, if human moves the robot fast during the demonstrations, the gradient of P should be steep so that robots quickly move after training, while if human moves robots slowly, the gradient of P is shaped to be gradual and robots slowly move after training.

To update a_1 and a_2, a residual error r is first defined as:

$$r=\frac{P_d}{\|P_d\|}-\frac{P}{\|P\|} \tag{7}$$

where $\frac{P_d}{\|P_d\|}$ is the normalized potential energy of P_d and $\frac{P}{\|P\|}$ is the normalized dynamic potential energy function. A cost function C is defined as:

$$C=0.5\cdot r^T\cdot r. \tag{8}$$

Its gradient respect to the task parameters is obtained as:

$$\Delta C=J_r^T\cdot r. \tag{9}$$

where J_r is the Jacobian matrix of r respect to task parameters. The update law for a_1 and a_2 is defined as:

$$A_{n+1}=A_n+s \tag{10}$$

where $A=\begin{bmatrix}a_1\\a_2\end{bmatrix}$ and s denotes the update step which is calculated by the following normal equation as:

$$(J_r^TJ_r)s=-J_r^T\cdot r \tag{11}$$

The update continues until $\| s \|$ and $\| \Delta C \|$, fall below a predefined tolerance value ε.

D. Combination of several sequential sub-tasks

In practical robotic applications, the given task may contain several sub-tasks and each sub-task has different task requirement. Robots should perform each sub-task in a sequential way so as to complete the whole task. In our algorithm, the combination of several sequential sub-tasks can be achieved by shaping the dynamic potential function

P sequentially such that robots would act in different motion behaviours according to different events. We define an event-based task variable $a(E_i,t)\Delta\varepsilon$ as:

$$a(E_i,t)\Delta\varepsilon \triangleq a(E_i,t)\frac{\partial P(\Delta x)}{\partial \Delta x} \tag{12}$$

where $a(E_i,t)$ is an event-based weight function, $\Delta\varepsilon$ denotes the task variable of dynamic potential energy function and $\Delta x = x(t) - x_d(t)$ is the position error between the robot position $x(t)$ and desired position $x_d(t)$. The event-based weight $a(E_i,t)$ is defined as:

$$a(E_i,t) = \begin{cases} 1, & \text{Event i detected} \\ f_{dec}(t), & \\ 0, & \text{Update } K, a_1, a_2, d_1, d_2 \text{ according to event i} \\ f_{inc}(t), & \\ 1, & \text{End of switch.} \end{cases} \tag{13}$$

where E_i denotes the event i, $f_{dec}(t)$ denotes a function to continuously decrease from 1 to 0 and $f_{inc}(t)$ denotes a function to increase from 0 to 1 continuously. $a(E_i,t)$ is at least twice differentiable and its derivative with respect to time $\dot{a}(E_i,t)$ is bounded. Hence, once an event is detected by the robots, $a(E_i,t)$ is reduced to 0 and task variable loose the control of robots. Then, the task parameters are updated according to the detected event and after that, $a(E_i,t)$ is increased from 0 to 1 again and the task variable would drive robots to perform the new task. Therefore, human can teach robots the tasks one by one based on the developed learning algorithm and specify the events in a sequential way so that robots can achieve a series of sequential tasks.

III. Controller

The proposed event-based task variable $a(E_i,t)\Delta\varepsilon$ can be integrated into the controllers of different kinds of robotic systems as a control feedback variable and thus the developed task learning algorithm can be easily implemented on various robots. For example, to implement on the robotic systems whose control command is specified as joint velocity, a joint velocity reference control signal $\dot{q}_r$ is defined as

$$\dot{q}_r = J^+(q)\dot{x}_d - J^+(q)a(E_i,t)\Delta\varepsilon \tag{14}$$

where q denotes the joint position of robots, J^+ is the pseudo-inverse of Jacobian matrix $J(q)$ from joint space to task space and $\dot{x}_d$ denotes the desired velocity.

For robot manipulators with torque control mode, the dynamics model can be specified as:

$$M(q)\ddot{q} + [\frac{1}{2}\dot{M}(q) + S(q,\dot{q})]\dot{q} + g(q) = \tau \tag{15}$$

where $M(q) \in \Re^{n\times n}$ is the inertia matrix, n denotes the degrees of freedoms of the robot, $[\frac{1}{2}\dot{M}(q) + S(q,\dot{q})]\dot{q} \in \Re^n$ denotes the centripetal and coriolis torques, and $S(q,\dot{q}) \in \Re^{n\times n}$ is a skew-symmetric matrix; $g(q) \in \Re^n$ represents the gravity vector; $\tau \in \Re^n$ denotes a vector of torque control inputs. To demonstrate the feasibility of using the event-based task variable in torque control modes, an adaptive controller is specified as:

$$\tau = -K_s s - K_p a(E_i,t)\Delta\varepsilon + Y_d(q,\dot{q},\dot{q}_r,\ddot{q}_r)\hat{\theta}_d \tag{16}$$

where $K_p \in \Re^{n\times n}$ is a positive proportional gain, $K_s \in \Re^{n\times n}$ is a gain matrix which is symmetric and positive definite , $s \in \Re^n$ is a sliding vector, $Y_d(q,\dot{q},\dot{q}_r,\ddot{q}_r) \in \Re^{n\times h}$ is the regressor matrix and $\hat{\theta}_d \in \Re^h$ is an estimate of unknown dynamics parameter vector $\theta_d \in \Re^h$ of the manipulator. The sliding vector s is defined as

$$s = \dot{q} - \dot{q}_r = \dot{q} - J^+(q)\dot{x}_d + J^+(q)a(E_i,t)\Delta\varepsilon, \tag{17}$$

and the regressor matrix in Eq.(16) is defined as:

$$M(q)\ddot{q}_r + [\frac{1}{2}\dot{M}(q) + S(q,\dot{q})]\dot{q}_r + g(q) = Y_d(q,\dot{q},\dot{q}_r,\ddot{q}_r)\theta_d. \tag{18}$$

The update law for $\hat{\theta}_d$ is given as:

$$\dot{\hat{\theta}}_d = -L_d Y_d^T(q,\dot{q},\dot{q}_r,\ddot{q}_r)s \tag{19}$$

Thus, by substituting Eq.(16), (17), (18) into Eq.(15), the closed-loop system is given as:

$$M(q)\dot{s} + [\frac{1}{2}\dot{M}(q) + S(q,\dot{q})]s + K_s s + Y_d(q,\dot{q},\dot{q}_r,\ddot{q}_r)\Delta\theta_d + [K_p J^T(q)a(E_i,t)\Delta\varepsilon] = 0. \tag{20}$$

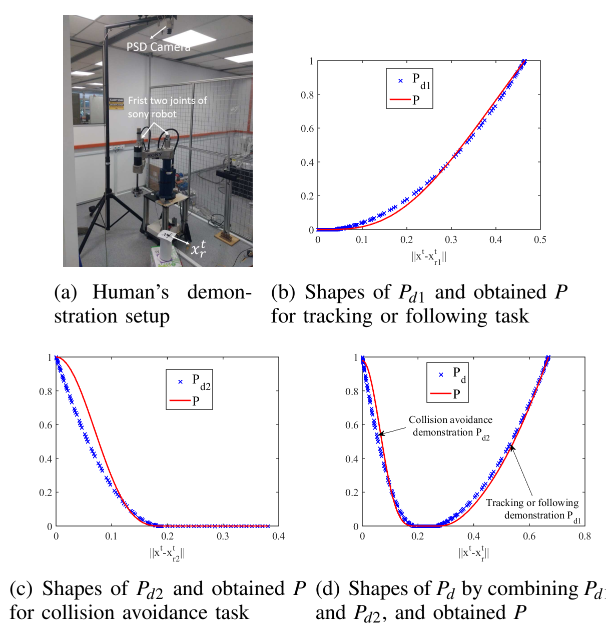
where $\Delta\theta_d = \theta_d - \hat{\theta}_d$.

To show the stability of the proposed adaptive controller, a Lyapunov-like function is proposed as:

$$V = \frac{1}{2}s^T M(q)s + a(E_i,t)P + \frac{1}{2}\Delta\theta_d^T L_d^{-1}\Delta\theta_d \tag{21}$$

Differentiating Eq.(21) and using Eq.(19) and (20), we have:

$$\dot{V} = -s^T K_s s - K_p[a(E_i,t)\Delta\varepsilon]^T[a(E_i,t)\Delta\varepsilon] + \dot{a}(E_i,t)P. \tag{22}$$



(a) Human's demonstration setup

(b) Shapes of P_{d1} and obtained P for tracking or following task

(c) Shapes of P_{d2} and obtained P for collision avoidance task

(d) Shapes of P_d by combining P_{d1} and P_{d2}, and obtained P

Fig. 3. Illustration of demonstration setup and results

Note that $\dot{a}(E_i,t)$ is not equal to 0 only during the smooth transition stage from one task to another when triggered by another event. Therefore, when robots are performing a given task, $\dot{a}(E_i,t)=0$ and thus $\dot{V}<0$, which implies that the task variable $\Delta\varepsilon$ converges to zero [15]. That is, $s\to 0, \Delta\varepsilon \to 0$ as $t\to\infty$. For the transition stage when another event is detected and $\dot{a}(E_i,t)\neq 0$, we can conclude from Eq.(22) that

$$V(t) \leq V(0) + \int_0^t \dot{a}(E_i,\sigma)P \, d\sigma \tag{23}$$

From the definition of $a(E_i,t)$ and the bounded property of P, it is clear that $\int_0^t \dot{a}(E_i,\sigma)P \, d\sigma$ is bounded and thus V is bounded. The system remains stable during the transition.

IV. Experiments

The proposed method was implemented on the first two joints of a scara robot as shown in Fig.3(a). The experiments include human's demonstrations in section IV-A and HRI tasks implementation in section IV-B.

A. Human demonstration

1) Demonstration setup: In the demonstrations, the robot was operated in off-line mode and human manually moved the robot to demonstrate the tasks. For every demonstration, human would first set an arbitrary reference position x_r^t as shown in Fig.3(a). Then, human would give the demonstration by manually moving the robot's end-effector in 2D-space, but the proposed method also works in 3D-space. The position x^t and speed $\dot{x}^t$ of the end-effector were recorded, which was used to automatically reconstruct the task energy function. Two demonstrations including tracking or following demonstration and collision avoidance demonstration, were given to illustrate the process of human's demonstrations. These two basic demonstrations can be used to describe a new task by combining them together.

2) Demonstration of tracking or following task: This demonstration was conducted as: human manually moved the end-effector towards the reference position x_r^t until reaching it, which demonstrated the behaviour of following the target.

The reconstructed task energy function P_{d1} in this demonstration was presented in Fig.3(b). Since the potential energy function is symmetric, the representative part is sufficient for learning. The task parameters were obtained as shown in Table I and thus the dynamic potential function P was shaped as shown in Fig.3(b). The desired region was specified as $\|\Delta x\| \leq 0.041(m)$ according to obtained d_1 and d_2. Hence, once the robot detected the target, the robot would move towards the target until the distance $\|\Delta x\| \leq 0.041(m)$.

3) Demonstration of collision avoidance task: This demonstration was conducted as: human manually moved the end-effector away from the reference position x_r^t until keeping certain safe distance, which demonstrated the behaviour of keeping a safe distance with the target.

The reconstructed task energy function P_{d2} in this demonstration was presented in Fig.3(c). The task parameters were obtained as shown in Table I and thus the dynamic potential function P was shaped as shown in Fig.3(c). The desired region was specified as $\|\Delta x\| \geq 0.183(m)$. Hence, once the target approached to the robot such that $\|\Delta x\| < 0.183(m)$, the robot would move away to keep a safe distance.

4) Combination of two demonstrations: following with a safe distance kept : The task of following with a safe distance kept has two different task requirements: i) following and ii) keeping a safe distance. These two task requirements had been demonstrated by human in the previous experiments and thus this new task can be achieved by combining them using Eq.(4). Here, C_1, C_2 were specified as: $C_1=0, C_2=-0.23$ and thus P_{d1} was shifted to right 0.23(m) as shown in Fig.3(d). The task parameters were obtained as shown in Table I and the dynamic potential function P was reshaped as shown in Fig.3(d). The desired region was specified as $0.183(m) \leq \|\Delta x\| \leq 0.271(m)$. Hence, the robot would follow the target once the target moved away ($\|\Delta x\| > 0.271(m)$), while keep a safe distance ($\|\Delta x\| > 0.183(m)$).

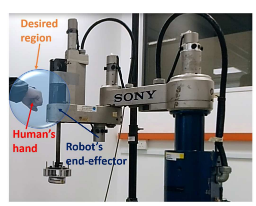
B. HRI tasks implementation with obtained P

1) Experimental results on following task: According to the task parameters obtained in section IV-A.2, the proposed adaptive controller in Eg.(16) was used to control the robot to follow human's hand motion. The control gain were set as: $K_s=0.01I, K_p=5I$. In this experiment, the center of the desired region was specified as human's hand which can be detected by a PSD camera and the robot would follow the hand motion. The illustration of this experiment was given in Fig.4 and the result in Fig.4(d) shows that when human

TABLE I

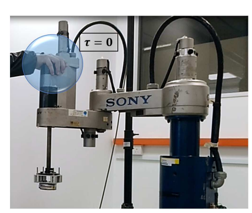
Obtained task parameters based on demonstrations

Data	S	d_1	d_2	a_1	a_2	Iteration
P_{d1}	1	0(m)	0.041(m)	1.291	13.66	2468
P_{d2}	1	0.183(m)	2(m)	0.056	25.46	2146
$P_{d1}+P_{d2}$	1	0.183(m)	0.271(m)	1.576	30.78	1628

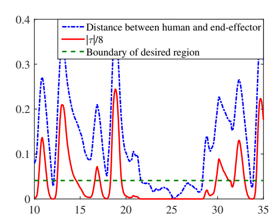


(a) Robot moved towards human's hand

(b) When human moved the hand forward, the robot followed it



(c) The robot was inside desired region, τ was reduced to 0 and human can manually move the robot



(d) τ vs t

Fig. 4. Experimental results on following human's hand motion

moved the hand away, the control input τ was increased to control the robot to follow human's hand until it reached the desired region and then τ was reduced to 0.

2) Experimental results on collision avoidance task: By using the task parameters obtained in section IV-A.3, the proposed adaptive controller was able to perform the collision avoidance task. The control gains were still the same. In this experiment, the robot should move away from human's hand to keep a safe distance when the hand was approaching. The illustration was given in Fig.5 and the results presented in Fig.5(d) shows that once $\|\Delta x\| < 0.183(m)$, τ was increased to move the robot away until $\|\Delta x\| \geq 0.183(m)$.

3) Experimental results on following with a safe distance kept: According the task parameters obtained in section IV-A.4, the robot was able to follow human's hand but keep a safe distance. The control gains were the same. The results of following human's motion with a safe distance kept are illustrated in Fig.6 and Fig.6(d) shows that when human moved the hand away from the end-effector, the control input τ was increased to control the robot to follow, while when the hand was too close to the robot, τ was increased again to move the robot to keep a safe distance.

V. Conclusion

In this paper, we have proposed a control strategy with a task learning algorithm that can be used to learn a set of different task requirements through demonstrations. The motion behaviours demonstrated by human have been formulated in a way that the behaviours can be acquired or learned by the robot systems through adjusting the task parameters of the dynamic potential function. It has been shown that the proposed control strategy allows the robots to perform similar tasks demonstrated by human and change its behaviours during HRI according to different applications in a stable manner.

References

[1] H. Kazerooni, R. Sterger, and L. Huang, "Hybrid control of the berkeley lower extremity exoskeleton (bleex)," *Int. J. Robot. Res.*, 25(5-6), pp. 561-573, 2006.

[2] C. J. Walsh, D. Paluska, K. Pasch, W. Grand, A. Valiente, and H. Herr, "Development of a lightweight, underactuated exoskeleton for loadcarrying augmentation," *IEEE Int. Conf. Robot. Autom.*, pp. 3485-3491,2006.

[3] A. M. Dollar and H. Herr, "Lower extremity exoskeletons and active orthoses: challenges and state-of-the-art," *IEEE Trans. Robot.*, 24(1), pp.144-158, 2008.

[4] J. Zhang and C.C. Cheah, "Passivity and stability of humanCrobot interaction control for upper-limb rehabilitation robots," *IEEE Trans. Robot.*, 31(2), pp.233-245, 2015.

[5] E. Gribovskaya, A. Kheddar and A. Billard, "Motion learning and adaptive impedance for robot control during physical interaction with humans," *IEEE Int. Conf. Robot. Autom.*, pp. 4326-4332, 2011.

[6] E. Noohi, M. Žefran and J. L.Patton, "A Model for Human-Human Collaborative Object Manipulation and Its Application to Human-Robot Interaction," *IEEE Trans. Robot.*, 32(4), pp.880-896, 2016.

[7] E. Magrini, F. Flacco and A. De Luca, "Control of generalized contact motion and force in physical human-robot interaction," *IEEE Int. Conf. Robot. Autom.*, pp. 2298-2304, 2015.

[8] X. Li, G. Chi, S. Vidas and C. C. Cheah, "Human-guided robotic comanipulation: two illustrative scenarios," *IEEE Trans. Contr. Syst. T.*, 24(5), pp.1751-1763, 2016.

[9] R. Dillmann, "Teaching and learning of robot tasks via observation of human performance," *Robot. Auton. Syst.*, 47(2), pp.109-116, 2004.

[10] H.S. Koppula, R. Gupta and A. Saxena, "Learning human activities and object affordances from rgb-d videos," *The Int. J. Robot. Res.*, 32(8), pp.951-970, 2013.

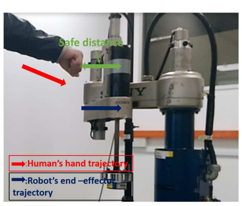
[11] P. Kormushev, S. Calinon and D. G. Caldwell, "Robot motor skill coordination with EM-based reinforcement learning," *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 3232-3237, 2010.

[12] V. Modugno, G. Neumann, E. Rueckert, G. Oriolo, J. Peters and S. Ivaldi, "Learning soft task priorities for control of redundant robots," *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp.221-226, 2016.

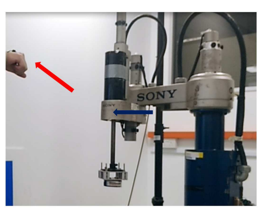
[13] S. Lyu and C. C. Cheah, "Adaptive Robot Control for Human-dominant Interactions using a General Task Function",*IEEE Int. Conf. on Real-time Comput. Robot.*, pp.39-44, 2016.

[14] R. Haghighi and C. C. Cheah, "Optical Manipulation of Multiple Groups of Microobjects Using Robotic Tweezers," *IEEE Trans. Robot.*, vol.32, no.2, pp.275-285, April 2016.39-44, 2016

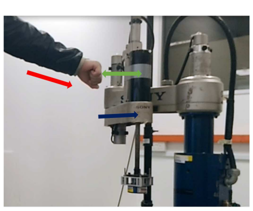
[15] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs,NJ, USA: Prentice-Hall, 1991.



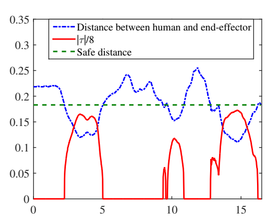
(a) Human moved the hand towards the robot and the robot moved away to keep a safe distance



(b) Human moved the hand away, the robot moved back to the position

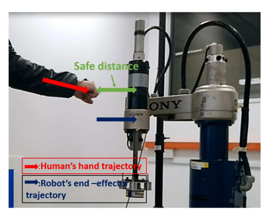


(c) Human's hand approached again and the robot moved away

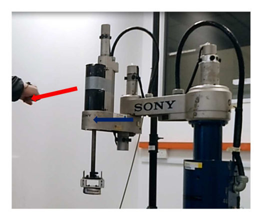


(d) τ vs t

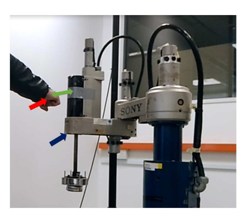
Fig. 5. Experimental results on collision avoidance with human's hand



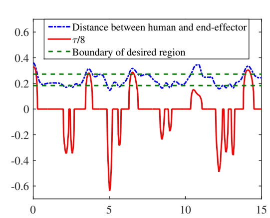
(a) Human moved the hand towards the robot and the robot moved away to keep a safe distance



(b) Human moved the hand away and the robot followed the hand's movements



(c) Human's hand approached again and the robot moved away



(d) τ vs t

Fig. 6. Experimental results on following with a safe distance kept