2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)
November 3 - 5, 2015, Seoul, Korea

Adaptation of Bimanual Assembly Tasks using Iterative Learning Framework

Nejc Likar, Bojan Nemec, Leon Žlajpah, Shingo Ando and Aleš Ude

***Abstract*— The paper deals with the adaptation of bimanual assembly tasks. First, the desired policy is shown by human demonstration using kinesthetic guidance, where both trajectories and interaction forces are captured. Captured entities are portioned to absolute and relative coordinates. During the execution, small discrepancies in object geometry as well as the influence of an imperfect control can result in large contact forces. Force control can diminish the above mentioned problems only to some extent. Therefore, we propose a framework that iteratively modifies the original demonstrated trajectory in order to increase the performance of the typical assembly tasks. The approach is validated on bimanual peg in a hole task using two KUKA LWR robots.**

***Index Terms*— bimanual manipulation, iterative learning control, task adaptation, real time control**

I. Introduction

Bimanual arm architecture enables the performance of a variety of assembly tasks, is essential for carrying heavy and spacious objects and enables the transfer of many human skills to robots. However, additional flexibility requires more complex control algorithms. Dual arm manipulation has been extensively investigated in the nineties. Earlier control architectures exploited master-slave approach, hybrid-force-torque and impedance control approach to synchronize motion of both arms [1], [2]. Today, most of the bimanual control architectures are based on the concept of symmetric control [3], which enables portioning of the task to so called absolute coordinates and relative coordinates and underlying internal and external forces, which are orthogonal [4]. This formalism allows the learning, demonstration and adaptation of bimanual tasks in above mentioned orthogonal subspaces. Adaptation, being the one of the key features of new generation of service and humanoid robots, can be accomplished in several ways. In most cases the adaptation is required to refine the previously demonstrated motion to different robot embodiment. Additionally, adaptation is necessary to cope with non-modelled environment constraints. Often applied paradigm for motion adaptation is reinforcement learning (RL) applying probabilistic algorithms [5], which can deal with high dimensionality spaces induced by parameterised policies [6]. Despite of these advances, learning capabilities of modern robots are still far from the learning capabilities of humans. While humans can quickly adapt to new situations, robots often have to relearn the whole policy in a lengthy exploration process, even when a good initial policy approximation is provided. Therefore, researchers are trying to find effective solutions to speed up learning. One of promising paradigms is also Iterative Learning Control (ILC). The main objective of ILC is to improve the behavior of the control system that operates repeatedly by iterative refinement of the feed-forward control input [7]. Due to its simplicity, effectiveness and robustness when dealing with repetitive operations, ILC is often applied in robotics [8]. As many tasks in industry as well as in home environments need to be executed repeatedly, it represents a natural choice for adaptation of such tasks.

In this paper we propose a new learning controller that applies to bimanual task adaptation. Bimanual adaptation was studied also in [9], where both robot arms were independent agents coupled only at the force level. In this work, robot arms are coupled both on kinematical and force level and treated as a single agent. The structure of the proposed algorithm allows easy integration into the Dynamic Motion Primitives (DMP) framework [10]. The proposed algorithm is general and can be used with both types of bimanual movements that can be represented by DMPs, i. e. discrete and periodic movements. Our experimental setup was composed of two KUKA LWR arms equipped with Barret hands. The performance of the proposed algorithm was evaluated on long poles insertion task, which is related to the classical peg-in-hole problem [11], [12], [13], [14], [15]. The paper is organized as follow. In Section II we outline kinematics and dynamics of a bimanual system. In Section III, the main contribution of the paper, we extend our previously presented trajectory adaptation scheme based on demonstrated position and force profiles to a bimanual system. We discuss also the stability of the proposed adaptation scheme. In Section IV the experimental results and the effectiveness of the proposed algorithm are given. Discussion and future work regarding bimanual adaptation are summarized in conslusion.

*This research was partially supported by EU Seventh Framework Programme grant 270273, Xperience.

Nejc Likar, Bojan Nemec, Leon Žlajpah, and Aleš Ude are with Humanoid and Cognitive Robotics Lab, Department of Automatics, Biocybernetics and Robotics, Jožef Stean Institute, Ljubljana, Slovenia,(e-mails: nejc.likar@ijs.si, bojan.nemec@ijs.si, leon.zlajpah@ijs.si, ales.ude@ijs.si) Shingo Ando is with Robotics Technology Group, Yaskawa Electric Corporation, Kitakyushu, Japan, (e-mail: shingo.ando@yaskawa.co.jp)

978-1-4799-6885-5/15/$31.00 ©2015 IEEE

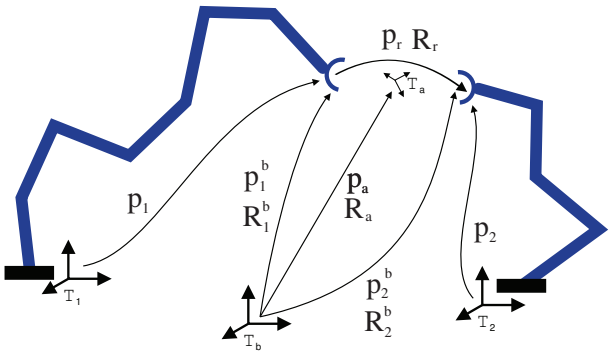

Fig. 1. Dual arm manipulator and the corresponding notation used in the paper.

II. Bimanual Task Kinematic Control

In this section we present a task-space control scheme for a bimanual system. This scheme is an extension of the previously proposed approach[16]. It fully characterizes a cooperative operational space and allows the user to specify the task in terms of geometrically meaningful motion variables defined at the position/orientation level [17], [16]. The resulting definition of the task variables in terms of the relative and absolute task motion of the cooperative system are mathematically well defined and have a clear physical meaning. Within this framework, both subspaces are orthogonal and thus decoupled - motion in relative coordinates does not affect absolute coordinates and vice versa. Consequently the control can be applied to both subsystems, relative and absolute, independently.

The key of our approach is in the definition of the common base coordinate systems for both subspaces, as illustrated in Fig. 1. According to this, the common base for the absolute coordinates is suitably chosen base $\mathcal{T}_b$ which applies to both robots, whereby the base for relative coordinates is placed in one of the robot's end effector, e.g. of the first robot. From now on we will use the notation where superscript j, $j \in \{1, 2, b\}$ denotes that the given quantity is specified relative to the coordinate system $\mathcal{T}_j$, while the subscript i, $i \in \{1, 2\}$ denotes the arm of a bimanual system and i, $i \in \{a, r\}$ denotes relative and absolute coordinates

According to this notation, absolute and relative task coordinates can be specified as

$$\begin{aligned} \mathbf{p}_r &= \mathbf{p}_2^1 = {\mathbf{R}_1^b}^T(\mathbf{p}_2^b - \mathbf{p}_1^b), && (1)\\ \mathbf{R}_r &= \mathbf{R}_2^1 = {\mathbf{R}_1^b}^T\mathbf{R}_2^b, && (2)\\ \mathbf{p}_a &= \frac{1}{2}(\mathbf{p}_1^b + \mathbf{p}_2^b), && (3)\\ \mathbf{R}_a &= \mathbf{R}_1^b\mathbf{R}_{\mathbf{k}_{21}^b}^b(\vartheta_{21}/2), && (4) \end{aligned}$$

where $\mathbf{p} \in \mathbb{R}^3$ applies to positions vector and $\mathbf{R} \in \mathbb{R}^{3\times3}$ to rotational matrices. $\mathbf{k}$ and ϑ_{21} are the axis and angle that realize the rotation $\mathbf{R}_1^b$ to $\mathbf{R}_2^b$. In quaternion notation, (2) and (4) are in the form

$$\begin{aligned} \mathbf{q}_r &= \mathbf{q}_2^1 = \bar{\mathbf{q}}_1^b * \mathbf{q}_2^b, && (5)\\ \mathbf{q}_a &= \mathbf{q}_1^b * \mathbf{q}_{\mathbf{k}_{21}^b}^b, && (6) \end{aligned}$$

where the quaternion $\mathbf{q}_1^b \in \mathbb{R}^4$ and $\mathbf{q}_2^b \in \mathbb{R}^4$ expresses the rotation of the TCP of the first and second robot in the common base coordinate frame $\mathcal{T}_b$, respectively. $\bar{\mathbf{q}}$ denotes conjugate quaternion and operator $*$ denotes quaternion product. $\mathbf{q}_{\mathbf{k}_{21}^b}^b$ denotes the unit quaternion corresponding to $\mathbf{R}_{\mathbf{k}_{21}^b}^b(\vartheta_{21}/2)$, which can be calculated from

$$\mathbf{q}_{\mathbf{k}_{21}^b}^b = \left(\cos\left(\frac{\vartheta_{21}}{4}\right), \mathbf{k}_{21}^b \sin\left(\frac{\vartheta_{21}}{4}\right)\right) \quad (7)$$

.

The corresponding relative and absolute forces and toques are

$$\begin{aligned} \mathbf{f}_r &= \frac{1}{2}(\mathbf{f}_1^1 - \mathbf{R}_r\mathbf{f}_2^2) && (8)\\ \mathbf{m}_r &= \frac{1}{2}(\mathbf{m}_1^1 - \mathbf{R}_r\mathbf{m}_2^2), && (9)\\ \mathbf{f}_a &= \mathbf{R}_1^b\mathbf{f}_1^1 + \mathbf{R}_2^b\mathbf{f}_2^2 && (10)\\ \mathbf{m}_a &= \mathbf{R}_1^b\mathbf{m}_1^1 + \mathbf{R}_2^b\mathbf{m}_2^2, && (11) \end{aligned}$$

where $\mathbf{f}_i^i \in \mathbb{R}^3$ and $\mathbf{m}_i^i \in \mathbb{R}^3$ denote the forces and torques measured at the i-th manipulator tool center point (TCP).

In order to control the robot, we have to map the desired relative and absolute task coordinates to the corresponding joint coordinates of both robots, denoted with $\boldsymbol{\theta} \in \mathbb{R}^{(N_1+N_2)}$, where N_1 and N_2 is the number of joints of the first and the second robot, respectively. This transformation is obtained through relative and absolute geometrical Jacobian, which maps the corresponding translational and angular velocities to the joint velocities

$$\begin{bmatrix} \dot{\mathbf{p}}_r \\ \boldsymbol{\omega}_r \end{bmatrix} = \mathbf{J}_r\dot{\boldsymbol{\theta}} \quad , \quad \begin{bmatrix} \dot{\mathbf{p}}_a \\ \boldsymbol{\omega}_a \end{bmatrix} = \mathbf{J}_a\dot{\boldsymbol{\theta}}. \quad (12)$$

Relative and absolute Jacobian matrices are obtained with the time derivation of the set of equations (1–4),

$$\mathbf{J}_r = \begin{bmatrix} -{\mathbf{R}_1^b}^T(\mathbf{J}_{1,p} + \mathbf{S}^T(\mathbf{p}_2 - \mathbf{p}_1)\mathbf{J}_{1,\omega}) & {\mathbf{R}_1^b}^T\mathbf{J}_{2,p} \\ -{\mathbf{R}_1^b}^T\mathbf{J}_{1,\omega} & {\mathbf{R}_1^b}^T\mathbf{J}_{2,\omega} \end{bmatrix} \quad (13)$$

and

$$\mathbf{J}_a = \begin{bmatrix} \frac{1}{2}\mathbf{J}_1 & \frac{1}{2}\mathbf{J}_2 \end{bmatrix}. \quad (14)$$

Subscript $(.)_p$ and $(.)_\omega$ denotes positional and rotational part of the Jacobian and $\mathbf{S}$ is anti-symmetric matrix [4].

If the task requires control only of the relative coordinates, the corresponding joint velocities are obtained from

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_r^+(\mathbf{v}_{r,d} + \mathbf{K}_r\mathbf{e}_r) + (\mathbf{I} - \mathbf{J}_r^+\mathbf{J}_r)\dot{\boldsymbol{\theta}}_0, \quad (15)$$

where $\mathbf{J}_r^+$ is the Moore-Penrose pseudo-inverse of the relative Jacobian $\mathbf{J}_r$, $\mathbf{v}_{r,d} \in \mathbb{R}^6$ are the desired relative translational and rotational velocities, $\mathbf{I}$ is identity matrix, $\mathbf{K}_r \in \mathbb{R}^{6\times6}$ is a diagonal matrix with the kinematic gains and $\mathbf{e}_r \in \mathbb{R}^6$ is the error between the desired and actual relative task coordinates, calculated as

$$\mathbf{e_r} = \begin{bmatrix} \mathbf{p}_{r,d} - \mathbf{p}_r \\ \log(\mathbf{q}_{r,d} * \bar{\mathbf{q}}_r) \end{bmatrix}. \quad (16)$$

The rotational part of the error is calculated using logarithmic map $\log$, which maps the quaternion describing the rotation between the desired and current pose to the rotation error vector. This mapping is defined as

$$\log(\mathbf{q}) = \log(v, \mathbf{u}) = \begin{cases} \arccos(v)\dfrac{\mathbf{u}}{\|\mathbf{u}\|}, & \mathbf{u} \neq 0 \\ [0,0,0]^{\mathrm{T}}, & \text{otherwise} \end{cases}. \quad (17)$$

Vector $\dot{\boldsymbol{\theta}}_0 \in \mathbb{R}^{(N_1+N_2)}$ is an arbitrary vector of joint velocities that is projected in the null-space of the primary task, selected in such a way that it optimizes an additional secondary task, i.e. obstacle avoidance, joint limit avoidance, singularity avoidance, etc.

If we would like to control both relative and absolute task coordinates, this can be accomplished by solving the equation

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_e^{+}(\mathbf{v}_{e,d} + \mathbf{K}_e\mathbf{e}_e) + (\mathbf{I} - \mathbf{J}_e^{+}\mathbf{J}_e)\dot{\boldsymbol{\theta}}_0, \quad (18)$$

where extended Jacobian is defined as

$$\mathbf{J}_e = \begin{bmatrix} \mathbf{J}_r \\ \mathbf{J}_a \end{bmatrix} \quad (19)$$

and $\mathbf{v}_{e,d}$ is the desired extended task space velocity composed of the desired relative and absolute task velocities. The $\mathbf{e}_e \in \mathbb{R}^6$ is the error between the desired and actual absolute task coordinates, calculated similar as in (16). Diagonal matrix $\mathbf{K}_e \in \mathbb{R}^{12\times12}$ contains suitably chosen positive kinematic control gains. Note that the dimension of the extended task defined with (19) can be ≤ 12, which allows to exploit the additional degrees of redundancy for secondary task(s).

III. Bimanual Task Adaptation

Assembly tasks performed by humans are usually accomplished with both hands. Humans are very good at performing bimanual assembly tasks that require compliance and force control and can quickly adapt to specific tools and environments. Therefore, we use human demonstration of the bimanual task as a starting point. Initial task is obtained with learning by demonstration (LbD) exploiting kinesthetic guidance. Unlike standard LbD approaches [18], we capture besides trajectories also forces and torques arising during the task execution as training data [19]. During the execution of learned skills, additional adaptation is often needed to cope with small differences induced by the environment, the inaccuracies in geometry of the manipulated objects, grasping tolerances, etc. It has been previously proved that on-line control alone can not in general provide successful adaptation in assembly tasks [20]. Therefore, various learning techniques were applied [21]. In our previous works we demonstrated that Iterative Learning Control (ILC) framework can efficiently adapt the previously demonstrated task to a new situation [9], [22], [19]. Therefore, ILC framework was applied also for bimanual task adaptation.

To learn the optimal control input, which is in our case the reference trajectory composed of position part vector $\mathbf{p}$ and rotation part quaternion $\mathbf{q}$, we applied ILC in the form

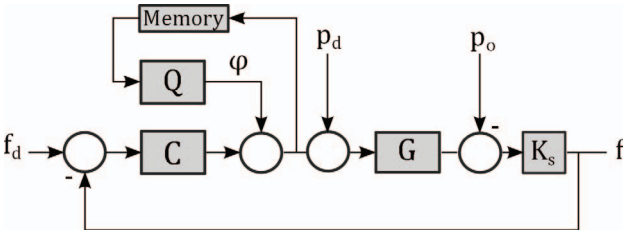

Fig. 2. Block diagram of force based adaptation scheme.

$$\mathbf{p}_l(k) = \mathbf{p}_d(k) + \boldsymbol{\varphi}_{p,l}(k) + \mathbf{C}_p(\gamma)\mathbf{e}_{f,l}(k) \quad (20)$$

$$\mathbf{q}_l(k) = \exp(\mathbf{C}_q(\gamma)\mathbf{e}_{m,l}(k)) * \exp(\boldsymbol{\varphi}_{q,l}(k)) * \mathbf{q}_d(k), \quad (21)$$

where l denotes the learning cycle, k is the time sample index, $\mathbf{C}_p(\gamma) \in \mathbb{R}^{3\times3}$ and $\mathbf{C}_q(\gamma) \in \mathbb{R}^{3\times3}$ are diagonal matrices composed of transfer function polynomials, γ is the backward shift operator, which delays a signal for one time sample and $\mathbf{e}_{f,l}(k)$ and $\mathbf{e}_{m,l}(k)$ are the force and torque tracking errors, respectively. Terms $\mathbf{p}_d(k)$ and $\mathbf{q}_d(k)$ denote the initially demonstrated trajectory in the form of the desired positions vector and rotations quaternion. Term $\boldsymbol{\varphi}_{p,l}(k)$ and $\boldsymbol{\varphi}_{q,l}(k)$ denotes position and rotation displacements, which will be learned by means of ILC. Remaining terms $\mathbf{C}_p(\gamma)\mathbf{e}_{f,l}(k)$ and $\mathbf{C}_q(\gamma)\mathbf{e}_{m,l}(k)$ belong to the admittance force controller, which minimizes force and torque tracking errors in the current iteration cycle. Therefore, the ILC scheme given by (20) and (21) is often refereed as "current iteration" ILC. Force and torque tracking errors are defined as $\mathbf{e}_{f,l}(k) = \mathbf{f}_d(k) - \mathbf{f}_l(k)$ and $\mathbf{e}_{m,l}(k) = \mathbf{m}_d(k) - \mathbf{m}_l(k)$, where $\mathbf{f}_d(k)$ and $m_d(k)$ are initially demonstrated forces and torques profiles, respectively. $\exp$ denotes the exponential map $\exp : \mathbb{R}^3 \mapsto \mathbb{S}$, defined as

$$\exp(\mathbf{r}) = \begin{cases} \cos(\|\mathbf{r}\|) + \sin(\|\mathbf{r}\|)\dfrac{\mathbf{r}}{\|\mathbf{r}\|}, & \mathbf{r} \neq 0 \\ 0, & \text{otherwise} \end{cases} \quad (22)$$

The compensation terms are learned with

$$\boldsymbol{\varphi}_{p,l}(k) = \mathbf{Q}_p(\gamma)(\boldsymbol{\varphi}_{p,l-1}(k) + \mathbf{C}_p(\gamma)\mathbf{e}_{f,l-1}(k)), \quad (23)$$

$$\boldsymbol{\varphi}_{q,l}(k) = \mathbf{Q}_q(\gamma)(\boldsymbol{\varphi}_{q,l-1}(k) + \mathbf{C}_q(\gamma)\mathbf{e}_{m,l-1}(k)), \quad (24)$$

where $\mathbf{Q}_p(\gamma) \in \mathbb{R}^{3\times3}$ and $\mathbf{Q}_q(\gamma) \in \mathbb{R}^{3\times3}$ are ILC transfer functions. Initial values of $\boldsymbol{\varphi}_{p,0}(k)$ and $\boldsymbol{\varphi}_{q,0}(k)$ are set to $0\ \forall\ k$. The same adaptation algorithm can be used for adaptation of both relative and absolute coordinates. In the above formulation we omitted indexes r and a, which define weather relative or absolute coordinates are subject of adaptation.

The stability of the proposed admittance force control law which iteratively updates the position compensation term will be analysed in the frequency domain of a time discrete system. For sake of simplicity, stability will be shown only for adaptation of the positional part of the trajectory. The overall ILC scheme for positional trajectories is outlined in

Fig. 2. Let uppercase letters denote one sided Z transform of the corresponding time-discrete signal denoted with low letter. Again, for the sake of simplicity, we will omit explicit dependence on z in transfer functions and Z transform of the signals. We define as a system output the force at the TCP of the robot, which is modelled assuming known environment stiffness $\mathbf{K}_s$,

$$\mathbf{F}_l = \mathbf{K}_s(\mathbf{G}\mathbf{P}_l - \mathbf{P}_o), \tag{25}$$

where $\mathbf{K}_s \in \mathbb{R}^{3\times3}$ is a diagonal positive definite environment stiffness matrix, $\mathbf{G} \in \mathbb{R}^{3\times3}$ is diagonal matrix containing transfer functions which map the desired position vector $\mathbf{P}_l$ into the actual position and $\mathbf{P}_o$ denotes the environment contact positions. In stability analyses we will consider the case where the robot dynamics is previously decoupled and linearized within the robot controller [23]. Therefore, $\mathbf{G}$ can be modelled as a diagonal matrix with transfer functions of second order. According to (20) and (23), Z transform of the error function $\mathbf{E}_l$, position update function $\mathbf{P}_l$ and learned offset function $\mathbf{\Phi}$ are

$$\mathbf{E}_l = \mathbf{F}_d - \mathbf{F}_l, \tag{26}$$

$$\mathbf{P}_l = \mathbf{P}_d + \mathbf{\Phi}_l + \mathbf{C}\mathbf{E}_l, \tag{27}$$

$$\mathbf{\Phi}_l = \mathbf{Q}(\mathbf{\Phi}_{l-1} + \mathbf{C}\mathbf{E}_{l-1}). \tag{28}$$

Now, let express the the error $\mathbf{E}_l$ as a function of the error in the previous learning cycle $\mathbf{E}_{l-1}$,

$$\begin{aligned}
\mathbf{E}_l &= \mathbf{F}_d - \mathbf{F}_l \qquad (29)\\
&= \mathbf{F}_d - \mathbf{K}_s(\mathbf{G}\mathbf{P}_l - \mathbf{P}_o)\\
&= \mathbf{F}_d - \mathbf{K}_s(\mathbf{G}(\mathbf{P}_d + \mathbf{\Phi}_l + \mathbf{C}\mathbf{E}_l) - \mathbf{P}_o)\\
&= \mathbf{Q}(\mathbf{F}_d - \mathbf{K}_s(\mathbf{G}(\mathbf{P}_d + \mathbf{\Phi}_{l-1} + \mathbf{C}\mathbf{E}_{l-1}) - \mathbf{P}_o)) -\\
&\quad \mathbf{K}_s\mathbf{G}\mathbf{C}\mathbf{E}_l + (\mathbf{I} - \mathbf{Q})(\mathbf{F}_d - \mathbf{K}_s(\mathbf{G}\mathbf{P}_d - \mathbf{P}_o))\\
&= \mathbf{Q}(\mathbf{F}_d - \mathbf{F}_{l-1}) - \mathbf{K}_s\mathbf{G}\mathbf{C}\mathbf{E}_l +\\
&\quad (\mathbf{I} - \mathbf{Q})(\mathbf{F}_d - \mathbf{K}_s(\mathbf{G}\mathbf{P}_d - \mathbf{P}_o))\\
&= \mathbf{Q}\mathbf{E}_{l\text{-}1} - \mathbf{K}_s\mathbf{G}\mathbf{C}\mathbf{E}_l + (\mathbf{I} - \mathbf{Q})(\mathbf{F}_d - \mathbf{K}_s(\mathbf{G}\mathbf{P}_d - \mathbf{P}_o)).
\end{aligned}$$

In the above equation we added and subtracted the term $\mathbf{Q}(\mathbf{F}_d - \mathbf{K}_s(\mathbf{G}\mathbf{P}_d - \mathbf{P}_o))$ and used (25) – (28). Rearranging (29) we obtain

$$\frac{\mathbf{E}_l}{\mathbf{E}_{l-1}} = \frac{\mathbf{Q}}{I + \mathbf{K}_s\mathbf{G}\mathbf{C}} + \frac{\mathbf{I} - \mathbf{Q}}{I + \mathbf{K}_s\mathbf{G}\mathbf{C}} \frac{\mathbf{F}_d - \mathbf{K}_s(\mathbf{G}\mathbf{P}_d - \mathbf{P}_o)}{\mathbf{E}_{l-1}} \tag{30}$$

Asymptotic stability is assured iff $\frac{\mathbf{E}_l}{\mathbf{E}_{l-1}} < 1 \ \forall l$. Inserting again the z dependence into transfer functions and signals and substituting $z = e^{j\omega}$ in (30), the condition for asymptotic stability becomes [24]

$$\frac{\mathbf{Q}(e^{j\omega})}{\mathbf{I} + \mathbf{K}_s\mathbf{G}(e^{j\omega})\mathbf{C}(e^{j\omega})} + \frac{\mathbf{I} - \mathbf{Q}(e^{j\omega})}{\mathbf{I} + \mathbf{K}_s\mathbf{G}(e^{j\omega})\mathbf{C}(e^{j\omega})}\epsilon < 1, \forall\ \omega, \tag{31}$$

where

$$\epsilon = \frac{\mathbf{F}_d(e^{j\omega}) - \mathbf{K}_s(\mathbf{G}(e^{j\omega})\mathbf{P}_d(e^{j\omega}) - \mathbf{P}_o(e^{j\omega}))}{\mathbf{E}_{l-1}(e^{j\omega})} \tag{32}$$

and $\omega = [-\pi, \pi]$ is the frequency normalised with the sampling time of our time-discrete system.

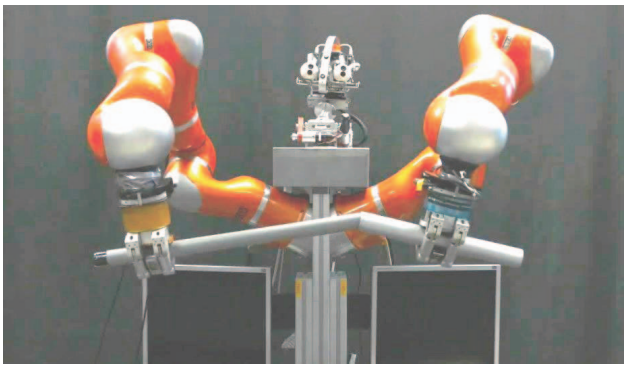

Fig. 3. Experimental platform.

Given the known transfer function $\mathbf{G}(z)$ and estimated environment stiffness $\mathbf{K}_s$ we have to design such admittance control law transfer function $\mathbf{C}(z)$ and learning function $\mathbf{Q}(z)$, that the learning error $\mathbf{E}(z)$ asymptotically decays to 0 when $l \to \infty$, Note that the nominator of the term ϵ (32) is 0, since $\mathbf{K}_s(\mathbf{G}\mathbf{P}_d - \mathbf{P}_o) = \mathbf{F}_d$, assuming ideal model for $\mathbf{K}_s$, $\mathbf{G}$ and $\mathbf{P}_o$. In practice, this is never true and the denominator of ϵ is small and bounded value, which depends only on the desired force $\mathbf{F}_d$, desired trajectory $\mathbf{P}_d$ and environment $\mathbf{P}_o$. Therefore, ϵ increases when the error $\mathbf{E}$ decreases. Consequently, zero learning error can be guaranteed only with the choice $\mathbf{Q}(z) = \mathbf{I}$. On the other hand, it is generally very hard to fulfill condition (31) with $\mathbf{Q}(z) = \mathbf{I}$. In most cases $\mathbf{Q}$ in the form of a low pass filter will assure the stability, but increase learning error [25]. Therefore, the design of $\mathbf{Q}(z)$ is a tradeoff between the robustness and stability and performance of the learning algorithm. The design of the learning algorithm can be thus summarised in the following steps:

1) calculate the upper bound of $\|\epsilon\|$ upon the admissible error $\mathbf{E}(e^{j\omega})$,
2) check if (31) is fulfilled, by e.g. Bode or Nyquist plot,
3) tune the parameters of the transfer function $\mathbf{C}$, $\mathbf{L}$ and $\mathbf{Q}$ until (31) is fulfilled.

IV. Experimental evaluation

We evaluated the performance of the proposed learning on bimanual peg in a hole tasks, where the robot has to insert one round pole into another, as show in Fig. 3. The outer diameter of one pole tightly fitted the inner diameter of the other pole. The experimental platform was composed of two KUKA LWR robot arms equipped with three finger Barret hands. The two arms were controlled with an external PC computer via FRI interface at sampling rate of 500 Hz. The Cartesian compliance of both robot arms was set to 1000 N/m for positions and to 300 Nm/rd for rotations. For this experiment, we have applied P type admittance control law with equal gains for $\mathbf{C}_p = 0.0002\ \mathbf{I}$. The KUKA built in controller decouples and linearizes the LWR robot dynamics, which can be approximated with discrete transfer function $\mathbf{G} = \frac{0.011z+0.01}{z^2-1.7z+0.7289}$. For the learning, we have chosen matrix $\mathbf{Q}$ as a diagonal matrices containing 2nd order low

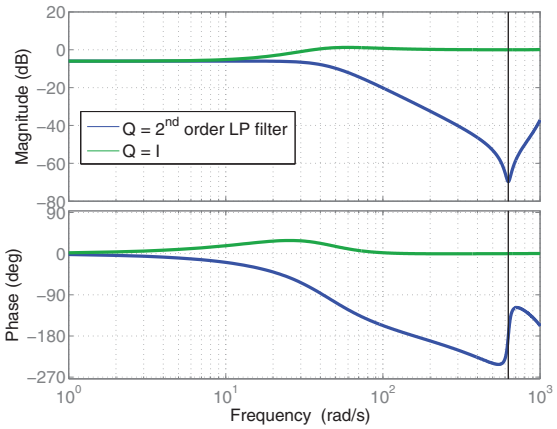

Fig. 4. Bode plot of the learning transfer functions for two choices of $\mathbf{Q}$.

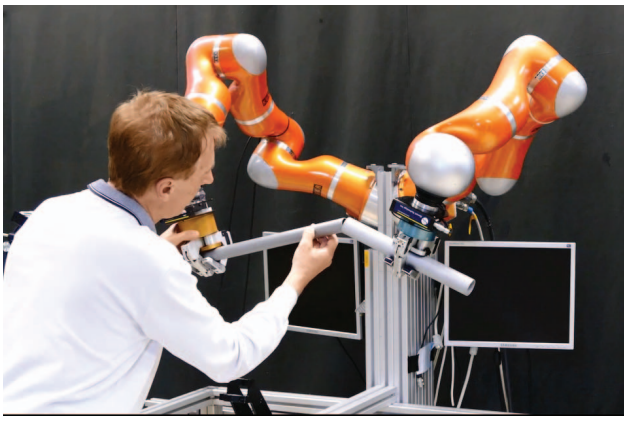

Fig. 5. Demonstration of the relative task with kinesthetic guiding of the right robot. The left robot does not move during the demonstration.

pass filter with zero at -0.9 and double poles at 0.85 in the Z plane. First, we verified the stability of the learning algorithm by drawing the Bode plot for the transfer function $\frac{\mathbf{Q}(z)}{\mathbf{I}+\mathbf{K}_s\mathbf{G}(z)\mathbf{C}(z)}$, where $\mathbf{K}_s$ was set to 500 N/m, which is exactly the mutual programmed stiffness of both robot arms. Fig. 4 shows Bode plot for two cases: in the first case $\mathbf{Q}$ is the 2nd order filter, and in the second case $\mathbf{Q}$ is unity matrix. According to (31), magnitude of the Bode plot should be below $(1-\epsilon)$ for all frequencies. It can be seen, that the Bode magnitude crosses the stability margin at high frequencies (above $20s^{-1}$) for $\mathbf{Q} = \mathbf{I}$, while it stays safe within the stability region when $\mathbf{Q}$ is in the form of a low pass filter. Note also that the magnitude plot uses logarithmic scale (dB) . The same result was verified also by the simulation, where scheme without the filtering starts to oscillate after it diminishes the tracking error in few initial learning cycles. The explanation of this phenomena is that at the beginning of the learning, low frequencies are dominant, whereas the Bode plot shows that both schemes are stable. When the algorithm diminishes tracking error, high frequencies become dominant and we have to use filtering to assure the stability of the learning.

As explained previously, small tolerances in position trajectory can result in high contact forces during assembly tasks. Force controller tries to diminish these forces instantly, while the learning minimizes them gradually during learning iterations. Force control might become inefficient or even unstable at rapid changes of the measured force and torque errors. Therefore, it is good idea to slow down the execution of the demonstrated policy when large deviation between the desired and actual forces and torques occurs. This flexibility to modulate the execution speed of the trajectory is provided by DMPs using phase stopping technique [10]. Therefore, time dependent trajectories $\mathbf{p}_d(k)$ and $\mathbf{q}_d(k)$ are replaced with the phase dependent DMP trajectories $\mathbf{p}_{DMP}(x)$ and $\mathbf{q}_{DMP}(x)$, where x is the phase variable [10]. Similarly, also the time dependent learned term $\boldsymbol{\varphi}_p(k)$ and $\boldsymbol{\varphi}_q(k)$ has to be replaced with phase dependent signals $\boldsymbol{\varphi}_{p,RBF}(x)$ and $\boldsymbol{\varphi}_{q,RBF}(x)$. Subscript RBF denotes, that the corresponding signal was encoded with radial basis functions (RBF), as explained in [19]. Accordingly, also force and torque tracking errors has to be encoded with RBF.

With the real experiment, we had to provide initial trajectories and force profiles in relative coordinates. As only the relative coordinates matter for this kind of the task execution, we can demonstrate the task by keeping one of the robots fixed, while the other was used for demonstration. This was accomplished with LbD, where we manually guided one of the robot arms using kinesthetic guidance and captured resulting relative coordinates calculation of relative coordinates from both robot poses using (1) and (5). One instance of the trajectory demonstration is show in Fig. 5. TCP forces and torques were captured from joint torque measurement, provided by the KUKA LWR. Within this setup, forces and torques can not be captured together with the position/orientation trajectories applying the kinesthetic guidance [19]. Therefore, we had to rerun the captured position/orientation trajectory in order to obtain non-distorted forces and torques. After that, we displaced one long peg in Barret hands in the local y direction for 5 mm and run 4 adaptation cycles. The results are shown in Fig. 6. Dotted line in graphs shows the insertion phase. As we can observe from plots, we have non-zero forces and torques even before contact. Part of this artefact is due to imprecise dynamics calculation when estimating TCP forces and torques from the joint torques. Another part is due to non-neglige inertia of long poles. However, the proposed adaptation algorithm effectively diminishes the force and torque tracking errors in just few learning cycles. For that reason, also the adaptation scheme with $\mathbf{Q} = \mathbf{I}$ in practice performs almost equally good as the scheme with $\mathbf{Q}$ in the form of the 2nd order filter, providing that the adaptation mechanism is switched off after few adaptation cycles.

V. Conclusions

In the paper we proposed a new force adaptation scheme for bimanual systems. The main advantage of the proposed algorithms is that adaptation act directly in relative and

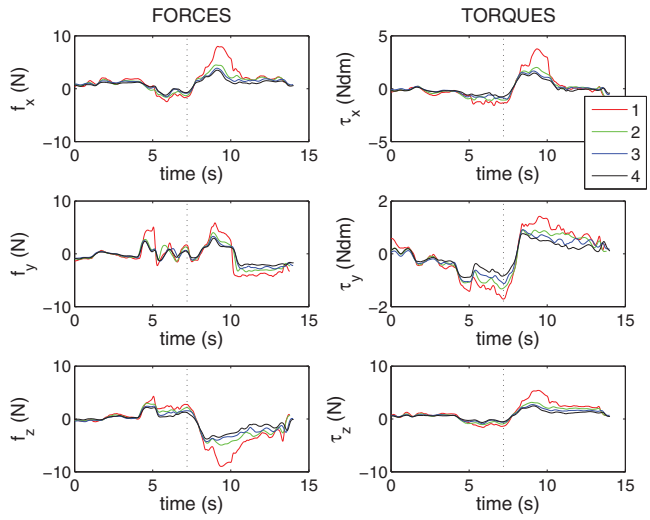

Fig. 6. Forces and torques during the long peg insertion tasks. Labels 1–4 show the adaptation cycle. We can observe extremely quick convergence of the ILC adaptation algorithm.

absolute coordinates. Subdivision of the motion of both arms to relative and absolute tasks is based on a modified definition of relative Jacobian, which assures also proper mapping to the Jacobian null-space, when additional tasks which exploit kinematic redundancy of the overall system are defined. The admittance based force adaptation is based on the ILC framework. The proposed controller belongs to a class of causal ILC controller, for which it was proved that the same steady state error can be obtained with standard feedback controller applying high gains [26]. However, high gain controllers are not suitable for robots interacting with humans [27], therefore we consider that ILC is still favourable. Stability of the proposed learning algorithm was discussed and proved together with practical notes how to choose the parameters of the learning algorithm and how to verify the stability. Theoretical results were verified with practical implementation of a dual arm peg-in-a-hole task. In our future work, we will extend the proposed algorithm to the impedance control law, which will modify control torques directly in contrast to the proposed algorithm, which adapts the position and orientation trajectory.

References

[1] Y. Zheng and J. Luh, "Joint torques for control of two coordinated moving robots," in *IEEE International Conference on Robotics and Automation. Proceedings.*, vol. 3, Apr 1986, pp. 1375–1380.

[2] T. Tarn, A. Bejczy, and X. Yun, "Coordinated control of two robot arms," in *IEEE International Conference on Robotics and Automation. Proceedings.*, vol. 3, Apr 1986, pp. 1193–1202.

[3] M. Uchiyama and P. Dauchez, "A symmetric hybrid position/force control scheme for the coordination of two robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, Philadelphia, PA, 1988, pp. 350–356.

[4] P. Chiacchio and S. Chiaverini, *Kinematic control of dual-arm systems*. Springer, 1998.

[5] J. Kober, D. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *International Journal of Robotics Research*, no. 11, pp. 1238–1274, 2013.

[6] E. A. Theodorou, J. Buchli, and S. Schaal, "A generalized path integral control approach to reinforcement learning," *Journal of Machine Learning Research*, vol. 11, pp. 3137–3181, 2010.

[7] D. Bristow, M. Tharayil, and A. Alleyne, "A survey of iterative learning control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, 2006.

[8] M. Norrlöf, "An adaptive iterative learning control algorithm with experiments on an industrial robot," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, pp. 245–251, Apr 2002.

[9] A. Gams, B. Nemec, A. Ijspeert, and A. Ude, "Coupling movement primitives: Interaction with the environment and bimanual tasks," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 816–830, 2014.

[10] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.

[11] J. F. Broenink and M. L. J. Tiernego, "Peg-in-Hole assembly using impedance control with a 6 DOF robot," in *Proceedings of the 8th European Simulation Symposium*, 1996, pp. 504–508.

[12] K. Hirana, T. Suzuki, and S. Okuma, "Optimal motion planning for assembly skill based on mixed logical dynamical system," in *7th International Workshop on Advanced Motion Control*, Maribor, Slovenia, 2002, pp. 359–364.

[13] Y. Li, "Hybrid control approach to the peg-in-hole problem," *IEEE Robotics and Automation Magazine*, vol. 4, no. 2, pp. 52–60, 1997.

[14] A. Stemmer, A. Albu-Schäffer, and G. Hirzinger, "An analytical method for the planning of robust assembly tasks of complex shaped planar parts," in *IEEE International Conference on Robotics and Automation (ICRA)*, Rome, Italy, 2007, pp. 317–323.

[15] T. Yamashita, I. Godler, Y. Takahashi, K. Wada, and R. Katoh, "Peg-and-hole task by robot with force sensor: Simulation and experiment," in *International Conference on Industrial Electronics, Control and Instrumentation (IECON)*, Kobe, Japan, 1991, pp. 980–985.

[16] F. Caccavale, P. Chiacchio, and S. Chiaverini, "Task-space regulation of cooperative manipulators," *Automatica*, vol. 36, pp. 879–887, 2000.

[17] B. Adorno, P. Fraisse, and S. Druon, "Dual position control strategies using the cooperative dual task-space framework," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 2010, pp. 3955–3960.

[18] R. Dillmann, "Teaching and learning of robot tasks via observation of human performance," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 109–116, 2004.

[19] F. Abu-Dakka, B. Nemec, J. Jørgensen, T. Savarimuthu, N. Krüger, and A. Ude, "Adaptation of manipulation skills in physical contact with the environment to reference force profiles," *Autonomous Robots*, vol. 39, no. 2, pp. 199–217, 2015.

[20] P. Pastor, L. Righetti, M. Kalakrishnan, and S. Schaal, "Online movement adaptation based on previous sensor experiences," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2011, pp. 365–371.

[21] L. Righetti, M. Kalakrishnan, P. Pastor, J. Binney, J. Kelly, R. Voorhies, G. Sukhatme, and S. Schaal, "An autonomous manipulation system based on force control and optimization," *Autonomous Robots*, vol. 36, no. 1-2, pp. 11–30, 2014.

[22] B. Nemec, F. J. Abu-Dakka, B. Ridge, J. A. Jørgensen, T. R. Savarimuthu, J. Jouffroy, H. G. Petersen, N. Krüger, and A. Ude, "Transfer of assembly operations to new workpiece poses by adaptation to the desired force profile," in *16th International Conference on Advanced Robotics (ICAR)*, Montevideo, Uruguay, 2013.

[23] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, "Operational space control: A theoretical and empirical comparison," *The International Journal of Robotics Research*, vol. 27, pp. 737–757, 2008.

[24] K. L. Moore, Y. Chen, and H.-S. Ahn, "Iterative learning control: A tutorial and big picture view," in *45th IEEE Conference on Decision and Control*, dec. 2006, pp. 2352 –2357.

[25] M. Norrlöf and S. Gunnarsson, "Experimental comparison of some classical iterative learning control algorithms," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 636–641, 2002.

[26] P. B. Goldsmith, "Brief on the equivalence of causal lti iterative learning control and feedback control," *Automatica*, vol. 38, no. 4, pp. 703–708, Apr. 2002.

[27] E. Gribovskaya, A. Kheddar, and A. Billard, "Motion learning and adaptive impedance for robot control during physical interaction with humans," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, May 2011, pp. 4326–4332.