Robotica (2014) volume 32, E3, pp. 1–16. © Cambridge University Press 2013
doi:10.1017/S0263574713000763

Virtual mechanism approach for dual-arm manipulation

Nejc Likar†,‡,*, Bojan Nemec‡ and Leon Žlajpah‡

†*Jozef Stefan International Postgraduate School, Jamova 39, 1000 Ljubljana, Slovenia*
‡*Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia*

(Accepted June 25, 2013)

SUMMARY
We propose a novel control approach for cooperative dual-arm manipulation tasks. Our scheme has three typical features: (1) the task performed by two robots is represented as a motion of a virtual mechanism and the task execution is accomplished by controlling the virtual mechanism; (2) the two arms and the task form a joined kinematic chain; (3) the scheme allows a cooperative dual-arm system to perform the task also when robot base is moving. The calculation of the Jacobian matrix of a chained two-arm mechanisms is based on a methodology which is using the Jacobian matrices of particular robot mechanisms and their end-effector positions and orientations. The proposed algorithm for dual-arm manipulation is verified by simulations of two cooperating planar robots and by experiments on a dual-arm robot consisting of two KUKA LWR arms.

KEYWORDS: Dual-arm manipulation; Object manipulation; Virtual mechanism.

1. Introduction

Many industrial robot applications and service robotics tasks require dual-arm robot systems. Cooperative manipulation of objects with a dual-arm robot provides more flexible and versatile task execution than the manipulation of object with a single-arm robot. For example, two arms can perform independent or coordinated complex bimanual assembly tasks without the aid of fixtures or jigs, transfer voluminous and heavy objects. Control policies for cooperative manipulators can be roughly divided into two categories: symmetric formulation and task-oriented formulation. The kinetostatic symmetric formulation proposed in ref. [1] is based on mappings between forces and velocities at the tightly grasped object and their counterparts in the "virtual sticks". The major problem in this formulation is that the task-space kinetostatic variables are often unsuitable for the description of the cooperative task.

The task-oriented formulation 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.[2,3] The main advantage of defining the cooperative task-space is that the control can be applied even for tasks which do not require physical interaction of both arms with the object. The differential kinematics of the system is formulated using the relative Jacobian of two manipulators as relation between velocities of the first robot end-effector relative to the second robot end-effector and joint velocities of both robots.[4,5] However, when the robot arms are mounted on mobile or free-floating platforms, the modeling of dual-arm manipulators systems becomes more complex.

The paper presents a method for obtaining the Jacobian matrix of the kinematic chain. The derivation of the Jacobian matrix of chained robot mechanisms is already well described.[6–8] However, if the kinematics of the particular robots composing the kinematic chain is already known then using this information can simplify the calculation of the kinematics of the whole system. We derive the Jacobian matrix using already known Jacobian matrices $\mathbf{J}_i$ and end-effector homogeneous transformations $\mathbf{T}_i$ of all robot arms composing the chained mechanism. This method is especially

* Corresponding author. E-mail: nejc.likar@ijs.si

useful for the calculation of the Jacobian for the dual-arm robot system augmented with a virtual mechanism representing the task.

Several authors used the term "virtual"[9–13] for an object, which is being manipulated by the robot. Usually, this "virtual object" is used in the framework of internal forces for grasping tasks. Our paper uses the term "virtual mechanism" to denote a virtual mechanism which has the same kinematics as the task. The task is represented as a virtual mechanism, which has as many degrees of freedom (DOF) as required by the task.[14] Task execution is performed with controlling internal joints of the virtual mechanism on a position/orientation level.

In our paper, we propose an approach to control the cooperative manipulators performing a task based on a representation of two robot arms and a task as an augmented kinematic chain.[15] We modeled the augmented kinematic with chaining three serial mechanisms (both robots and the virtual mechanism). In this approach, the base frame of the augmented kinematic chain is located at the base frame of the first robot, while the end-effector of the augmented kinematic chain is located at the base frame of the second robot. The proposed formulation allows cooperative manipulators performing a task also in the case of moving bases. An example of such case is cooperative mobile manipulators, where the relative position and orientation between the robot bases is considered as the end-effector of the augmented kinematic chain. Many authors[16–20] already well described the control of the manipulator mounted on a mobile platform (vehicle). In these works, the system is modeled as a macro-/mini-structure. Major problem in these approaches is that is necessary to introduce the additional DOF of the moving platform in the system.

By introducing the proposed augmented kinematic chain, it was necessary to consider an extended task-space to control the main task defined with the virtual mechanism. The extended fingers concept presented in ref. [21] allows execution of multiple additional tasks, which are arranged in a priority order. However, these additional tasks are all executed as a secondary task in the null-space of the main task. When virtual mechanism is used then the task associated with is has to be executed together with the end-effector task as the primary task. Therefore, we used the extended task-space control as proposed in refs. [22, 23], where additional tasks are fulfilled along with the end-effector task, i.e. all tasks in the extended task-space have the same priority. Actually, when the base position of the second robot is not strictly prescribed, it can be treated as a lower-priority task and other redundancy resolution schemes can be used.

The paper is divided in four sections. Initially, in Section 2, we introduce the method for chaining of serial mechanisms. Next, in Section 3, task-space definition is presented and we introduce the virtual mechanism. We use this knowledge to derive the Jacobian of the dual-arm system which is modeled as an augmented kinematic chain in Section 4. Closed-loop velocity control for the dual-arm system executing a task is presented in Section 5. In Sections 6 and 7, simulation and experimental results to corroborate the proposed control scheme are presented. In Section 8, final remarks are given in conclusions. The conclusion section is followed by Appendices A and B.

2. Chaining of Serial Mechanisms

In this section, we propose a methodology for deriving of the Jacobian matrix of the augmented kinematic chain. The *augmented kinematic chain* is a representation of two or more serial robot manipulators connected serially. The Jacobian of the chained mechanisms is calculated based on the known Jacobian matrices $\mathbf{J}_i$ and end-effector positions/orientations $\mathbf{T}_i$ of all manipulators composing the chained mechanism

$$\mathbf{J}_{Nk} = f(\mathbf{J}_1, \mathbf{T}_1, \mathbf{J}_2, \mathbf{T}_2, \ldots, \mathbf{J}_k, \mathbf{T}_k). \tag{1}$$

2.1. Kinematics of serially chained manipulators

Consider a robot manipulator consisting of k chained manipulators as it is shown in Fig. 1, where index i denotes individual manipulators, $i = 1, \ldots, k$. In general, the position/orientation of the ith end-effector can be represented in the base frame $\mathcal{T}_i^b$ of that ith robot with the positional vector $\boldsymbol{p}_i$ and the rotation matrix $\mathbf{R}_i$, which compose the homogeneous transformation $\mathbf{T}_i$ representing the

https://doi.org/10.1017/S0263574713000763 Published online by Cambridge University Press

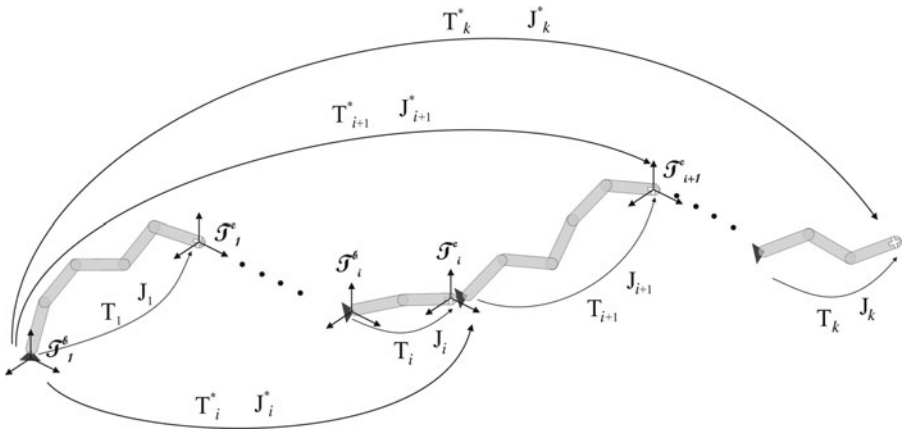

Fig. 1. With the proposed methodology, it is possible to chain serial mechanisms together and describe them as a one kinematic chain.

transformation from $\mathcal{T}_i^b$ to $\mathcal{T}_i^e$.

$$\mathbf{T}_i = \begin{bmatrix} \mathbf{R}_i & \boldsymbol{p}_i \\ \mathbf{0} & 1 \end{bmatrix}. \tag{2}$$

Let n_i denote the number of joints of the ith robot and $\boldsymbol{q}_i$ the n_i-dimensional vector of joint variables, respectively. Then, the differential kinematics representing the relation between end-effector velocities $\dot{\boldsymbol{x}}_i$ and joint velocities $\dot{\boldsymbol{q}}_i$ equals $\mathbf{J}_i(\boldsymbol{q}_i)$,

$$\dot{\boldsymbol{x}}_i = \mathbf{J}_i \dot{\boldsymbol{q}}_{.i} \tag{3}$$

In general, $\dot{\boldsymbol{x}}_i$ is a six-dimensional vector, which can be divided into two parts: the linear velocity $\boldsymbol{v}_i$ and the angular velocity $\omega_{\mathbf{i}}$ of the end-effector[6]

$$\dot{\boldsymbol{x}}_i = \begin{bmatrix} \boldsymbol{v}_i \\ \omega_{\mathbf{i}} \end{bmatrix}. \tag{4}$$

Consequently, the Jacobian matrix can be split into the position part $\mathbf{J}_{i,v}$ and the orientation part $\mathbf{J}_{i,\omega}$

$$\mathbf{J}_i = \begin{bmatrix} \mathbf{J}_{i,v} \\ \mathbf{J}_{i,\omega} \end{bmatrix} \tag{5}$$

and using Eqs. (4) and (5) in Eq. (3), we obtain the relations

$$\boldsymbol{v}_i = \mathbf{J}_{i,v} \dot{\boldsymbol{q}}_i, \tag{6}$$

$$\omega_i = \mathbf{J}_{i,\omega} \dot{\boldsymbol{q}}_i. \tag{7}$$

On the other hand, the linear velocity $\boldsymbol{v}_i$ can also be obtained as the time derivative of the positional part $\boldsymbol{p}_i$ in Eq. (2),

$$\boldsymbol{v}_i = \frac{\partial p_i}{\partial t}. \tag{8}$$

Using the time derivative of the rotation matrix[8] in Eq. (2), we obtain the angular velocity in the form

$$\dot{\mathbf{R}}_i = \mathbf{S}(\omega_i)\mathbf{R}_i, \tag{9}$$

where $\mathbf{S}(\omega_i)$ is the 3×3 skew-symmetric matrix.[8] Note that these equations are expressing the velocities of the ith robot in the base frame $\mathcal{T}_i^b$.

https://doi.org/10.1017/S0263574713000763 Published online by Cambridge University Press

As shown in Fig. 1, the end-effector frame of the ith manipulator coincides with the base frame of the $i+1$th manipulator. We can write the homogeneous transformation matrix that describes the transformation between the end-effector frame $\mathcal{T}_{i+1}^{e}$ and the base frame $\mathcal{T}_{1}^{b}$ as

$$\mathbf{T}_{i+1}^{*} = \mathbf{T}_{i}^{*}\mathbf{T}_{i+1} = \begin{bmatrix} \mathbf{R}_{i}^{*}\mathbf{R}_{i+1} & \mathbf{R}_{i}^{*}\boldsymbol{p}_{i+1} + \boldsymbol{p}_{i} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{10}$$

The vector of joint velocities of the chained mechanism is

$$\dot{\boldsymbol{q}}_{i+1}^{*} = \begin{bmatrix} \dot{\boldsymbol{q}}_{i}^{*} \\ \dot{\boldsymbol{q}}_{i+1} \end{bmatrix}. \tag{11}$$

Using Eq. (8) and the position part of $\mathbf{T}_{i+1}^{*}$ in Eq. (10) yields

$$\boldsymbol{v}_{i+1}^{*} = \dot{\mathbf{R}}_{i}^{*}\boldsymbol{p}_{i+1} + \mathbf{R}_{i}^{*}\boldsymbol{v}_{i+1} + \boldsymbol{v}_{i}^{*}. \tag{12}$$

Rearranging this equation using Eqs. (9) and (52) and using Eqs. (6) and (7) for $\boldsymbol{v}_i$, $\boldsymbol{v}_{i+1}$, ω_i, respectively, we obtain the equation

$$\mathbf{J}_{i+1,v}^{*}\dot{\boldsymbol{q}}_{i+1}^{*} = \mathbf{S}(\mathbf{R}_{i}^{*}\boldsymbol{p}_{i+1})^{T}\mathbf{J}_{i,\omega}^{*}\dot{\boldsymbol{q}}_{i}^{*} + \mathbf{R}_{i}^{*}\mathbf{J}_{i+1,v}\dot{\boldsymbol{q}}_{i+1} + \mathbf{J}_{i,v}^{*}\dot{\boldsymbol{q}}_{i}^{*}, \tag{13}$$

where we can extract the position part $\mathbf{J}_{i+1,v}^{*}$

$$\mathbf{J}_{i+1,v}^{*} = \begin{bmatrix} \mathbf{J}_{i,v}^{*} + \mathbf{S}(\mathbf{R}_{i}^{*}\boldsymbol{p}_{i+1})^{T}\mathbf{J}_{i,\omega}^{*} & \mathbf{R}_{i}^{*}\mathbf{J}_{i+1,v} \end{bmatrix}. \tag{14}$$

Similarly, differentiating the rotation part of the homogeneous transformation (10) yields

$$\dot{\mathbf{R}}_{i+1}^{*} = \dot{\mathbf{R}}_{i}^{*}\mathbf{R}_{i+1} + \mathbf{R}_{i}^{*}\dot{\mathbf{R}}_{i+1}. \tag{15}$$

Using the properties given in Appendix A and rearranging the result to a form as in Eq. (7)

$$\mathbf{J}_{i+1,\omega}^{*}\dot{\boldsymbol{q}}_{i+1}^{*} = \mathbf{J}_{i,\omega}^{*}\dot{\boldsymbol{q}}_{i}^{*} + \mathbf{R}_{i}^{*}\mathbf{J}_{i+1,\omega}\dot{\boldsymbol{q}}_{i+1}, \tag{16}$$

where we can obtain the orientation part of the Jacobian matrix $\mathbf{J}_{i+1,\omega}^{*}$ in form

$$\mathbf{J}_{i+1,\omega}^{*} = \begin{bmatrix} \mathbf{J}_{i,\omega}^{*} & \mathbf{R}_{i}^{*}\mathbf{J}_{i+1,\omega} \end{bmatrix}. \tag{17}$$

Combining Eqs. (14) and (17) yields the Jacobian matrix $\mathbf{J}_{i+1}^{*}$ of $i+1$ serially chained robots.

$$\mathbf{J}_{i+1}^{*} = \begin{bmatrix} \mathbf{J}_{i,v}^{*} + \mathbf{S}(\mathbf{R}_{i}^{*}\boldsymbol{p}_{i+1})^{T}\mathbf{J}_{i,\omega}^{*} & \mathbf{R}_{i}^{*}\mathbf{J}_{i+1,v} \\ \mathbf{J}_{i,\omega}^{*} & \mathbf{R}_{i}^{*}\mathbf{J}_{i+1,\omega} \end{bmatrix}. \tag{18}$$

Note that $\mathbf{J}_{i+1}^{*}$ is a $6 \times n_{i+1}^{*}$ dimensional matrix, where $n_{i+1}^{*} = n_{i}^{*} + n_{i+1}$.

2.2. Reverse chained manipulators

In the previous section, we presented how to obtain the Jacobian matrix when the robots are joined in the normal way, i.e. the base frame $\mathcal{T}_{i+1}^{b}$ coincides with the end-effector frame $\mathcal{T}_{i}^{e}$. However, when the robot mechanisms are joined by their end-effectors, i.e. the end-effectors frames $\mathcal{T}_{i}^{e}$ and $\mathcal{T}_{i+1}^{e}$ coincide, and the base frame $\mathcal{T}_{i+1}^{b}$ is the new end-effector frame of the chained robot (see Fig. 2), we have to use the reverse kinematics of the $i+1$ robot,

$$\mathbf{T}_{i+1}^{*} = \mathbf{T}_{i}^{*}\mathbf{T}_{i+1,}^{R} \tag{19}$$

https://doi.org/10.1017/S0263574713000763 Published online by Cambridge University Press

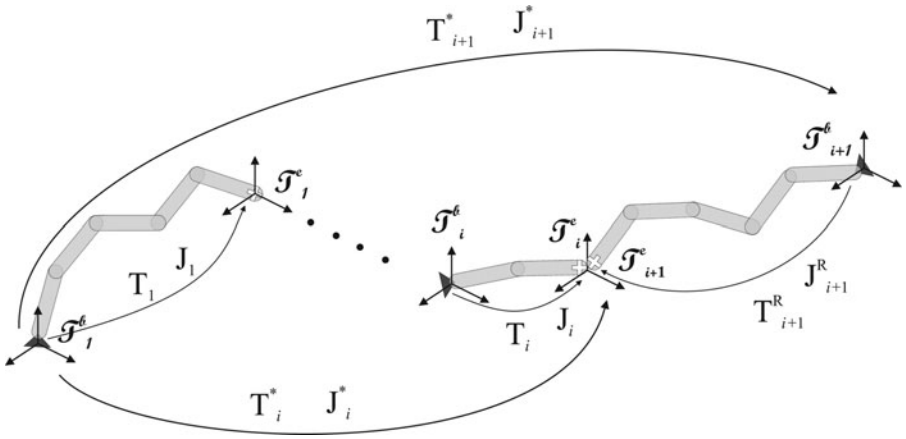

Fig. 2. As both robot arms are actually connected by the end-effectors, we have to reverse the kinematic transformation for one arm.

where $\mathbf{T}_{i+1}^{R}$ is the reversed homogeneous transformation, $\mathbf{T}_{i+1}^{R} = \mathbf{T}_{i+1}^{-1}$. Furthermore, for obtaining the Jacobian matrix $\mathbf{J}_{i+1}^{*}$ of such chained kinematic structure, the knowledge of the reversed Jacobian matrix $\mathbf{J}_{i+1}^{R}$ is necessary. The Jacobian matrix $\mathbf{J}_{i+1}^{R}$ can be obtained by separately differentiating the position and the rotation part of $\mathbf{T}_{i+1}^{-1}$ defined as

$$\mathbf{T}_{i+1}^{-1} = \left[\mathbf{R}_{i+1}^{T} \; -\mathbf{R}_{i+1}^{T}\boldsymbol{p}_{i+1}\right]. \tag{20}$$

After some calculations, we get

$$\mathbf{J}_{i+1}^{R} = \begin{bmatrix} -\mathbf{R}_{i+1}^{T}\mathbf{J}_{i+1,v} - \mathbf{S}(-\mathbf{R}_{i+1}^{T}\boldsymbol{p}_{i+1})^{T}\mathbf{J}_{i+1,\omega} \\ -\mathbf{R}_{i+1}^{T}\mathbf{J}_{i+1,\omega} \end{bmatrix}, \tag{21}$$

where the upper term is the position part and bottom term is the rotational part of the Jacobian matrix. The derivation of the Jacobian matrix $\mathbf{J}_{i+1}^{*}$ is the same as given in Section 2.1, except that we have to use $\mathbf{T}_{i+1}^{-1}$ and J_{i+1}^{R} instead of $\mathbf{T}_{i+1}$ and J_{i+1}, respectively.

3. Task-Space Definition

The motion necessary to perform a task can be represented as a transformations in the task coordinate system $\mathcal{T}_t$ with a homogeneous transformation $\mathbf{T}_t$

$$\begin{aligned} \mathbf{T}_t &= \mathbf{T}_t(t) \\ \mathbf{T}_t &= \begin{bmatrix} \mathbf{R}_t & \boldsymbol{p}_t \\ \mathbf{0} & 1 \end{bmatrix}, \end{aligned} \tag{22}$$

where $\boldsymbol{p}_t$ and $\mathbf{R}_t$ represent the task position and orientation, respectively.

Each task has its constraints and to describe it is optimal to use natural coordinates. For example, for describing dual-arm apple peeling task, the most suitable is the spherical coordinate system. We propose to represent a task with a virtual mechanism, with a kinematic structure corresponding to the nature of the task. Let assume that the task requires m_t degrees of motion for its execution. Then, it can be represented by an n_t-DOF virtual mechanism and a corresponding structure (see Fig. 3). Note that $m_t = n_t$.

The kinematics of this virtual mechanism can be expressed in general as

$$\boldsymbol{x}_t = f(\boldsymbol{q}_t), \tag{23}$$

https://doi.org/10.1017/S0263574713000763 Published online by Cambridge University Press

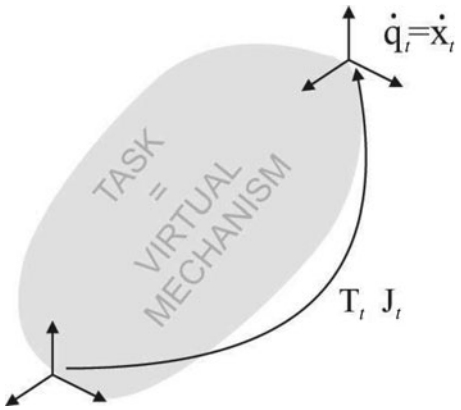

Fig. 3. Task-space definition.

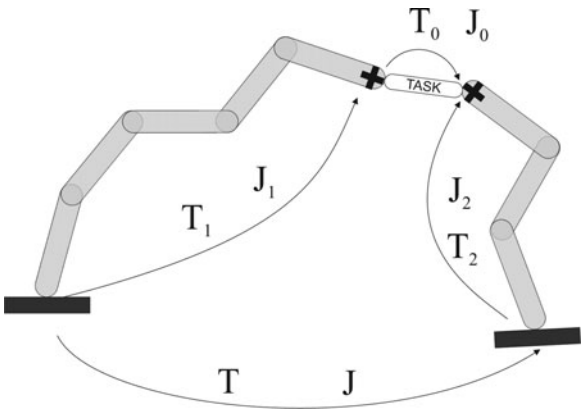

Fig. 4. Dual-arm system performing a task.

where $\boldsymbol{x}_t$ and $\boldsymbol{q}_t$ are the end-effector positions and generalized coordinates of the virtual mechanism. As this is a virtual mechanism, we select such generalized coordinates in the joint-space of the virtual mechanism that they are equal to the operational space coordinates of the task, giving

$$\boldsymbol{q}_t = \boldsymbol{x}_t. \tag{24}$$

Hence, the differential inverse kinematics is very simple

$$\dot{\boldsymbol{q}}_t = \dot{\boldsymbol{x}}_t. \tag{25}$$

4. Two Cooperating Robot Arms System

Consider, we have a dual-arm robot arm system which is executing a task as shown in Fig. 4. To exploit the advantages of a dual-arm system, a suitable kinematic formulation for such system has to be used. Most of the published formulations use the absolute and the relative position to describe the task, both expressed as a function of the positions/orientations of the two end-effectors.[2–5] In this paper, we propose to represent the task with a *virtual mechanism* (see Fig. 3). When the task is represented with a virtual mechanism, we can join both robots and the virtual mechanism in serially chained mechanism, which we denote in the following as the *augmented kinematic chain*. The augmented kinematic chain has the base in the base of one robot arm and the end-effector in the base of the other robot arm. Adding the virtual mechanism to the kinematic structure of the whole system adds m_t DOFs to the system, where m_t is the dimension of the task-space. Note also that the end-effector of the augmented system also represents a task. So, we have now actually two tasks: the first task is the

https://doi.org/10.1017/S0263574713000763 Published online by Cambridge University Press

"original" task and the second task is the position of new end-effector, i.e. the position of the robot arm base. Remarkably, this formulation is convenient for the dual-arm robot systems where one of the arms is movable (e.g. mounted on a platform).

The kinematics of this augmented kinematic chain can be solved using the methods presented in previous sections.

We can represent the homogeneous transformation for the above system as a kinematic chain from base of the first robot via the virtual mechanism to the base of the second robot as

$$\mathbf{T}_a = \mathbf{T}_1 \mathbf{T}_t \mathbf{T}_2^{-1}, \tag{26}$$

where the transformation $\mathbf{T}_t$ represents the task in its base frame $\mathcal{T}_t$.

Substituting Eqs. (2) and (22) into Eq. (26) yields

$$\mathbf{T}_a = \begin{bmatrix} \mathbf{R}_1 \mathbf{R}_t \mathbf{R}_2^T & \boldsymbol{p}_1 + \mathbf{R}_1 \boldsymbol{p}_t - \mathbf{R}_1 \mathbf{R}_t \mathbf{R}_2^T \boldsymbol{p}_2 \\ \mathbf{0} & 1 \end{bmatrix}. \tag{27}$$

Let $\dot{\boldsymbol{q}}_a$ denote the vector of joint velocities of the augmented kinematic chain,

$$\dot{\boldsymbol{q}}_a = \begin{bmatrix} \dot{\boldsymbol{q}}_1 \\ \dot{\boldsymbol{q}}_t \\ \dot{\boldsymbol{q}}_2 \end{bmatrix}, \tag{28}$$

which is composed of joint vector $\boldsymbol{q}_i$ of individual robot, and vector of internal variables of the virtual mechanism $\boldsymbol{q}_t$. If $\dot{\boldsymbol{x}}_a$ denote the end-effector velocity of the augmented kinematic chain, direct kinematics can be described as

$$\dot{\boldsymbol{x}}_a = \mathbf{J}_a \dot{\boldsymbol{q}}_a, \tag{29}$$

where $\mathbf{J}_a$ is the Jacobian matrix of the augmented kinematic chain. We obtained it as described in Section 2.

$$\mathbf{J}_a = \begin{bmatrix} \mathbf{J}_{1,v} + \mathbf{A}\mathbf{J}_{\mathbf{1},\omega} & \mathbf{J}_{1,\omega} \\ \mathbf{R}_1(\mathbf{J}_{t,v} - \mathbf{B}\mathbf{J}_{\mathbf{t},\omega}) & \mathbf{R}_1 \mathbf{J}_{t,\omega} \\ \mathbf{C}\mathbf{J}_{\mathbf{2},\omega} - \mathbf{D}\mathbf{J}_{\mathbf{2},\mathbf{v}} & \mathbf{D}\mathbf{J}_{\mathbf{2},\omega} \end{bmatrix}^T, \tag{30}$$

where

$$\begin{aligned} \mathbf{A} &= \mathbf{S}(\mathbf{R}_1 \mathbf{x}_\mathbf{t})^\mathbf{T} - \mathbf{S}(\mathbf{R}_\mathbf{1} \mathbf{R}_\mathbf{t} \mathbf{R}_\mathbf{2}^\mathbf{T} \mathbf{x}_\mathbf{2})^\mathbf{T}, \\ \mathbf{B} &= \mathbf{S}(\mathbf{R}_t \mathbf{R}_2^T \mathbf{x}_\mathbf{2})^\mathbf{T}, \\ \mathbf{C} &= \mathbf{R}_1 \mathbf{R}_t \mathbf{R}_2^T \mathbf{S}(\mathbf{x}_\mathbf{2})^\mathbf{T}, \\ \mathbf{D} &= \mathbf{R}_1 \mathbf{R}_t \mathbf{R}_2^T. \end{aligned}$$

Note that $\mathbf{J}_a$ is an $m_a \times n_a$ dimensional matrix, where $n_a = n_1 + n_t + n_2$.

Considering Eq. (28) and using the relation Eq. (25), the Jacobian matrix associated with the virtual mechanism is

$$\mathbf{J}_t = \begin{bmatrix} \mathbf{0}_1 & \mathbf{I} & \mathbf{0}_2 \end{bmatrix}, \tag{31}$$

where $\mathbf{I}$ is an $n_t \times n_t$ identity matrix and $\mathbf{0}_i$ are $m_t \times n_i$ zero matrices, where n_i is the number of DOF of the ith robot manipulator.

5. Closed-Loop Velocity Control

We are considering the tasks where the positions of the manipulators end-effectors are important, i.e. the control has to assure good position tracking. However, if the manipulators hold the same

https://doi.org/10.1017/S0263574713000763 Published online by Cambridge University Press

object, the serial link mechanisms are in a closed-loop configuration and internal forces between robot end-effectors arise, which influence the tracking accuracy. Therefore, it is necessary to control these internal forces. As the problem of internal forces is beyond the scope of this paper, we rely on a two-level control system and assume that the low-level controller[24] compensates the internal forces. In the following, we describe the proposed control algorithms at the upper level where the redundancy resolution is resolved.

In the framework of the proposed control strategy, there are actually two tasks: first, the tasks the robot arms should primary do and, second, the position of the end-effector of the augmented robot system, i.e. the base of the moving robot arm. Therefore, our approach allows a simple task motion control of a dual-arm robot system when the robot base is not fixed. In multi-task systems, there is always a question if all tasks are equally important or they can be prioritized.[25]

When both tasks, i.e. the end-effector motion and the robot base motion, have to be fulfilled simultaneously, then they have the same priority. To solve the inverse kinematics, we have to use the extended task-space formulation,[22,23] which can be defined as

$$\dot{\boldsymbol{x}}_e = \begin{bmatrix} \dot{\boldsymbol{x}}_t \\ \dot{\boldsymbol{x}}_a \end{bmatrix}. \tag{32}$$

Then, the relation between the extended task-space velocities and joint space velocities is given as

$$\dot{\boldsymbol{x}}_e = \mathbf{J}_e \dot{\boldsymbol{q}}_a, \tag{33}$$

where $\mathbf{J}_e$ is an $m_e \times n_a$ extended Jacobian matrix given in the form

$$\mathbf{J}_e = \begin{bmatrix} \mathbf{J}_t \\ \mathbf{J}_a \end{bmatrix} \tag{34}$$

and $m_e = m_t + m_a$. The solution of Eq. (33) is given in the form

$$\dot{\boldsymbol{q}}_a = \mathbf{J}_e^{\#} \dot{\boldsymbol{x}}_e + \mathbf{N}_e \xi_a, \tag{35}$$

where

$$\mathbf{J}_e^{\#} = \mathbf{W}^{-1} \mathbf{J}_e^T (\mathbf{J}_e \mathbf{W}^{-1} \mathbf{J}_e^T)^{-1} \tag{36}$$

is a weighted generalized-inverse of the $\mathbf{J}_e$, $\mathbf{W}$ is the $(n_a \times n_a)$ weighting matrix, $\mathbf{N}_e = (\mathbf{I} - \mathbf{J}_e^{\#} \mathbf{J}_e)$ is an $(n_a \times n_a)$ matrix representing the projection into the null-space of $\mathbf{J}_e$, and ξ_a is an arbitrary n_a-dimensional vector[26] of the augmented kinematic chain joint velocities. Note that all primary tasks are included in $\dot{\boldsymbol{x}}_e$ and the homogenous part of solution, $\mathbf{N}_e \dot{\boldsymbol{q}}_n$, is used in secondary tasks (e.g. obstacle avoidance) and the joint space velocities $\dot{\boldsymbol{q}}_n$ are projected into the null-space of the extended Jacobian $\mathbf{J}_e$. To avoid any drifts, a task-space controller is usually implemented for $\dot{\boldsymbol{x}}$, namely

$$\dot{\boldsymbol{x}}_e = \dot{\boldsymbol{x}}_{d,e} + \mathbf{K}\boldsymbol{e}, \tag{37}$$

where $\dot{\boldsymbol{x}}_{d,e}$ is the desired task-space velocity, $\boldsymbol{e} = \boldsymbol{x}_d - \boldsymbol{x}$ is the task-space error, and $\mathbf{K}$ is a positive-definite gain matrix.

In the cases where the robot base is movable and its position and/or orientation is not so important as the main task, i.e. the position of the end-effector of the augmented kinematic chain x_a is not strictly prescribed, then the positioning of the robot base can be treated as a lower-priority task. We propose to use the *augmented* inverse-based method proposed in ref. [27] where velocities for the lower priority are projected in the null-space of the augmented Jacobian considering all higher-priority tasks. Using this formulation, the primary task is given by Eq. (25) and the secondary tasks by Eq. (29). The inverse kinematic solution is then given in the form

$$\dot{\boldsymbol{q}}_a = \mathbf{J}_t^{\#} \dot{\boldsymbol{x}}_t + (\mathbf{J}_a \mathbf{N}_t)^{\#} (\dot{\boldsymbol{x}}_a - \mathbf{J}_a \mathbf{J}_t^{\#} \dot{\boldsymbol{x}}_t). \tag{38}$$

https://doi.org/10.1017/S0263574713000763 Published online by Cambridge University Press

The first term guarantees the joint motion necessary for the execution of the desired task with the virtual mechanism. The homogeneous solution of Eq. (38) represents the part of the joint velocity causing the motion of the augmented chain end-effector. The term $\mathbf{J}_a\mathbf{J}_t^{\#}\dot{\boldsymbol{x}}_t$ is the velocity of the augmented chain end-effector due to the virtual mechanism motion. The above solution guarantees the exact achievement of the desired motion $\dot{\boldsymbol{x}}_a$ only in the case of sufficient degree of redundancy. Note that if there are any other less-important tasks to be fulfilled together with the two main tasks, the algorithm (38) can be generalized for more tasks.[25] To avoid any drifts, the same task-space controller given by Eq. (37) is also used in Eq. (38).

6. Simulation Results

6.1. Dual-arm planar robot

In order to demonstrate the efficiency of the proposed approach, we first present simulation results of two cooperating planar 7-DOF robots. The first robot has fixed base and the second robot's base is movable, i.e. is mounted on a moving conveyor. The system is modeled as a augmented kinematic chain, where base of the first robot is the base of the augmented kinematic chain, while the base of the second robot is the end-effector of the augmented kinematic chain and its motion is constrained to the linear motion on a conveyor.

For demonstration, we have selected a balancing task, where both robots are holding together a bar. Additionally, the task requires to keep the bar in the horizontal position. Hence, the task-space is two-dimensional: the translational motion in the x direction for holding the bar, and the rotation around the z axis for maintaining the bar in horizontal position. For this task, the virtual mechanism is defined as a 1-DOF manipulator with a prismatic joint with variable a. For orientation, we have to control absolute orientation of the base of the virtual mechanism. As the virtual mechanism base is fixed to the end-effector of the first robot, the orientation task can be accomplished by controlling the orientation ω_z of the end-effector of the first robot.

Additionally, the base of the second robot has to follow the desired motion on a moving linear conveyor which defines the motion of the end-effector of the augmented kinematic chain $\dot{\boldsymbol{x}}_a$. For simplicity, the motion of the conveyor was defined as a sinusoidal function in the y direction.

Assigning both tasks the same priority and using the extended task-space definition (32), we can write

$$\dot{\boldsymbol{x}}_e = \begin{bmatrix} \dot{\boldsymbol{x}}_t \\ \dot{\boldsymbol{x}}_a \end{bmatrix}, \tag{39}$$

where

$$\dot{\boldsymbol{x}}_t = \begin{bmatrix} \dot{a} \\ \omega_z \end{bmatrix}. \tag{40}$$

The generalized coordinates $\boldsymbol{q}_a$ are defined as in Eq. (28) and the extended Jacobian matrix is defined as in Eq. (34), where

$$\mathbf{J}_t = \begin{bmatrix} \mathbf{0} & \mathbf{J}_{t1} & \mathbf{0} \\ \mathbf{J}_{t2} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{41}$$

Selecting joint coordinates as proposed by Eq. (24), the task matrices $\mathbf{J}_{t1}$ and $\mathbf{J}_{t2}$ are identity matrices. The $\mathbf{0}$ are zero matrices of corresponding dimensions.

The controller is based on the algorithm (Eq. (35)) with the task controller (Eq. (37)). The null-space motion of the augmented kinematic chain was not used in this simulation. Therefore, the velocities ξ_a in the controller (Eq. (35)) were set to zero.

Simulation results show that the cooperative robots can effectively perform the given task. The snapshots of six instances of given task are presented in Fig. 5.

In the second simulation example, these two robots are performing the same task of balancing and simultaneously avoiding a moving obstacle in their workspace. The first planar robot is mounted on a fixed support, and the second robot base is mounted on a floating platform. In this case, the position

https://doi.org/10.1017/S0263574713000763 Published online by Cambridge University Press

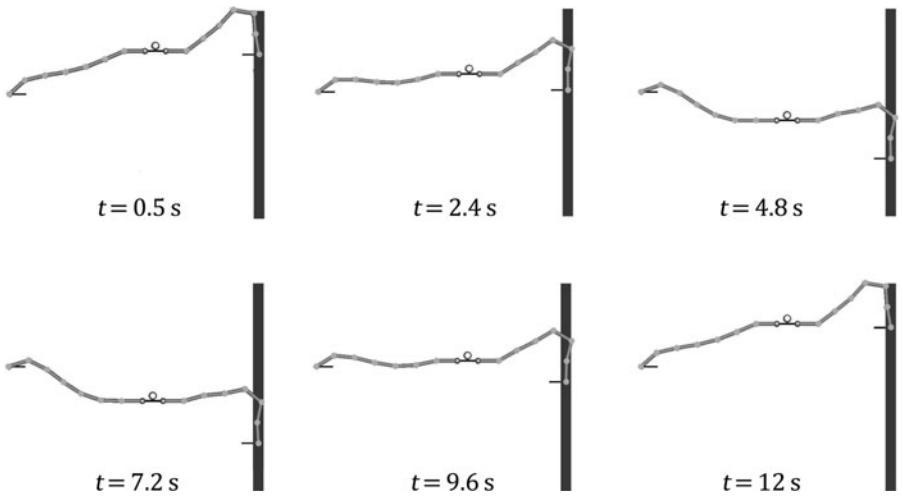

Fig. 5. The task of balancing a ball on a bar with two planar robot while one robot's base is fixed on a linear conveyor: robot configurations at six time instances.

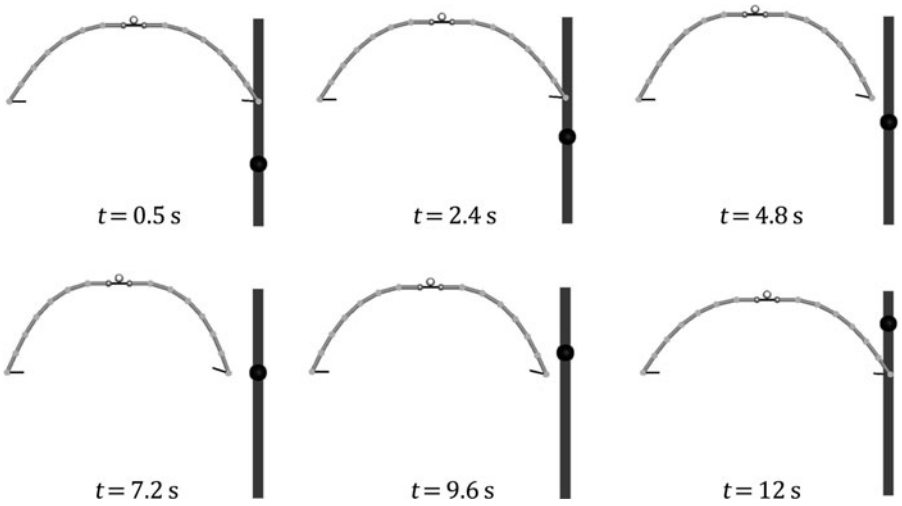

Fig. 6. The task of balancing a ball on a bar with two planar robot while avoiding the obstacle in the robot workspace: robot configurations at six time instances.

of the second robot base is not defined. The system is modeled as an augmented kinematic chain, similar as in the first simulation example.

We assumed that the obstacle is movable along the y direction (on the conveyor) and the task associated with the position of the end-effector of the augmented chain had lower priority as the balancing task.

In the null-space of higher-priority tasks,[25] the additional task was used for achieving the optimal configuration of the dual-arm planar robot system. Therefore, we have selected the redundancy resolution scheme as given by Eq. (38) generalized for more tasks,[25] with the task-space controller equation (37).

Simulation results show that the cooperative robots can effectively perform the balancing task while avoiding the obstacle. The snapshots of six instances of given task are presented in Fig. 6.

The value of the positive-definite gain matrix in Eq. (37) was chosen empirically to achieve necessary tracking accuracy.

https://doi.org/10.1017/S0263574713000763 Published online by Cambridge University Press

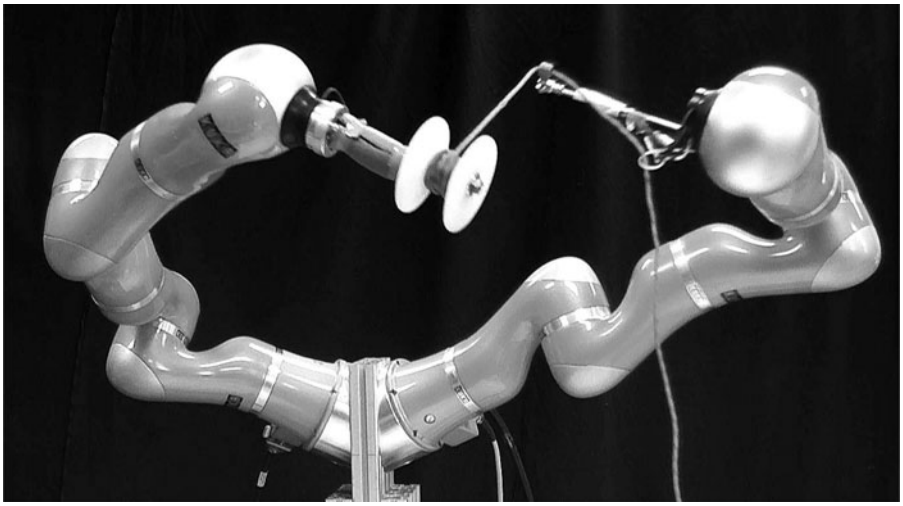

Fig. 7. Experimental setup with two Kuka LWR robot arms for task of winding a wool on a spool.

7. Experimental Results

Experiments were conducted on upper body humanoid robot torso, composed of two 7-DOF KUKA LWR robot arms as shown in Figs. 7 and 9. Both robots were controlled using Matlab/Simulink library[28] using the Fast Research Interface (FRI). The experiments were performed using joint stiffness control mode.[24] The dynamic decoupling in Kuka control unit (KRC) allowed us to use simple velocity control given by Eq. (35), where the weighting matrix was set to $\boldsymbol{W} = \boldsymbol{I}$.

7.1. Winding the wool on a spool with the dual-arm Kuka LWR robot

In the first experiment, the task is to wind the wool on a spool. This is a typical operation, where the absolute task position is not important. Namely, one robot holds in its gripper the spool and the other is rotating around the spool to wind the wool. The experimental setup is shown in Fig. 7. To describe the task, two control variables are required: the translational position along the z axis which prevents tangling of the wool, and the angle φ around the z axis. The distance r from center of the spool to the wool is not critical and can be constant during the task execution. The most suitable description of the given task is in cylindrical coordinates. So, the virtual mechanism is defined as a 2-DOF robot with one rotational and one linear joint. Our concept allows to use any type of the virtual mechanism. Therefore, we select a mechanism, where the joint coordinates are actually the task coordinates.

$$\boldsymbol{x}_0 = \begin{bmatrix} z \\ \varphi \end{bmatrix} = \boldsymbol{q}_0. \tag{42}$$

Therefore, the corresponding Jacobian matrix equals to the identity matrix

$$\mathbf{J}_0 = \frac{\partial \boldsymbol{x}_0}{\partial \boldsymbol{q}_0} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{43}$$

The trajectory for the motion is defined in the virtual mechanism coordinate frame $\mathbf{T}_0$. To perform the circular motion, the angle φ is a ramp function from 0 to 2π, while the linear movement along the z axis is a 0.5Hz sine signal with amplitude 0.4m. Figure 8 shows the reference and actual trajectories in the task of winding the wool on a spool.

It can be seen that the task can be effectively performed with configuring the movement of virtual mechanism.

7.2. Balancing the plate with the dual-arm Kuka LWR robot

The next experiment is to balance a bottle on a plate with two robot arms. Two Kuka LWR robots, with Barret-Hand grippers, are holding a plate as seen in Fig. 9. For performing this task, in addition

https://doi.org/10.1017/S0263574713000763 Published online by Cambridge University Press

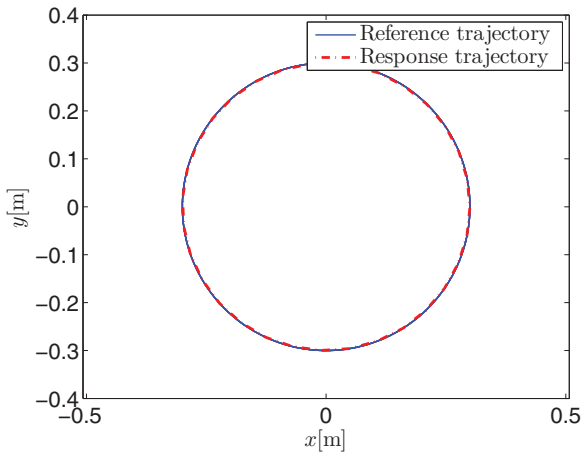

Fig. 8. (Colour online) Reference trajectories (solid lines) and responses (dotted lines) for the virtual mechanism task-space position control.

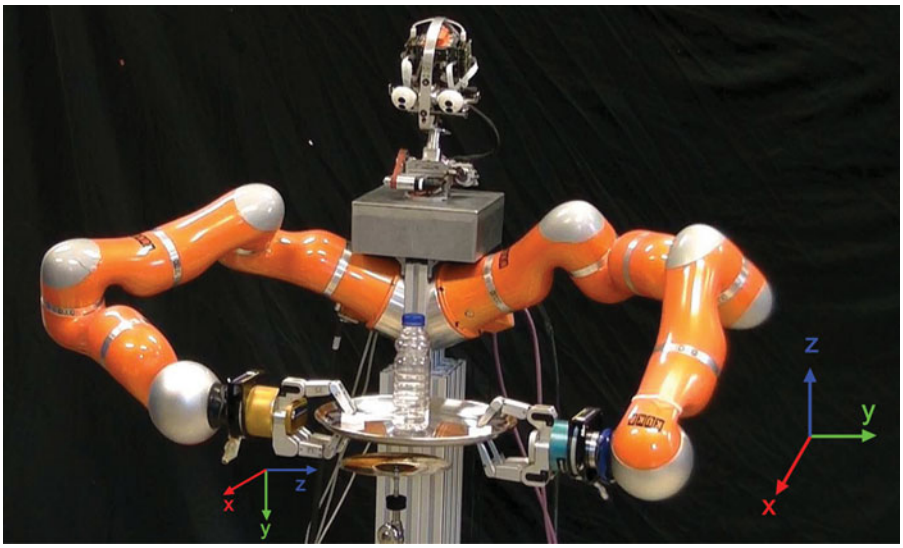

Fig. 9. (Colour online) Experimental setup with two Kuka LWR robot arms for task of balancing the plate.

to preserving the distance between the robot hands holding the plate, it is also necessary to control the orientation of the plate. The desired distance is obtained by controlling the relative positions between the grippers. For that, we defined a 3-DOF virtual mechanism with prismatic joints. For preserving the desired orientation of the plate, we have to control the absolute orientation if the plate, which is accomplished by controlling the orientation of the first (right) robot gripper. For this we need two DOFs of the first robot. So, for the whole task, the extended task-space is defined as

$$\dot{\boldsymbol{x}}_{e1} = \begin{bmatrix} \dot{\boldsymbol{x}}_{t1} \\ \dot{\boldsymbol{x}}_{t2} \\ \dot{\boldsymbol{x}}_{a} \end{bmatrix}, \tag{44}$$

where the subscript $t1$ denotes the positioning task defined with the virtual mechanism and the subscript $t2$ denotes the orientation. Consequently, the extended Jacobian matrix equals

$$\mathbf{J}_{e1} = \begin{bmatrix} \mathbf{J}_{t1} \\ \mathbf{J}_{t2} \\ \mathbf{J}_{a} \end{bmatrix}. \tag{45}$$

https://doi.org/10.1017/S0263574713000763 Published online by Cambridge University Press

We have defined the structure of the virtual mechanism in such a way that $\dot{\boldsymbol{x}}_{t1} = \dot{\boldsymbol{q}}_t = \dot{\boldsymbol{d}}$ and $\mathbf{J}_{t1} = \mathbf{I}$. The $\boldsymbol{d} = \left[d_x, d_y, d_z\right]^T$ is the three-dimensional vector, which represents the relative position between the hands. The desired position $\boldsymbol{d}_d$ for controlling the virtual mechanism task (holding the plate) depends on the diameter of the plate, $\boldsymbol{d}_d = [0, 0, 0.48\text{m}]$.

The $\dot{\boldsymbol{x}}_{t2} = \boldsymbol{\omega}_1$ is the task of preserving the orientation of the plate, and $\boldsymbol{\omega}_1$ is the angular velocity of the first robot (right robot on Fig. 9) end-effector considering only the rotation around the x and y axes,

$$\boldsymbol{\omega}_1 = \begin{bmatrix} \dot{\psi} \\ \dot{\theta} \end{bmatrix} = \mathbf{J}_{t2}\dot{\boldsymbol{q}}_a. \tag{46}$$

Here, $\dot{\psi}$ is the rotation velocity around the x axis and $\dot{\theta}$ is the rotation velocity around the y axis of the world coordinate system. The orientations $\psi_d = \pi/2$ and $\theta_d = 0$ assure a horizontal position of the plate surface. Note that the rotation of the plate around the vertical axis is free. In Eq. (46), the $\mathbf{J}_{t2}$ is obtained easily by using a corresponding part of the Jacobian matrix of the first robot

$$\mathbf{J}_{t2} = \begin{bmatrix} \mathbf{J}_{1,\omega_x} & \mathbf{0} & \mathbf{0} \\ \mathbf{J}_{1,\omega_y} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{47}$$

The $\mathbf{0}$ are zero matrices of corresponding dimensions.

In this experiment also a secondary task is defined: to place the plate on the table (denoted as $t3$). To accomplish this task, the position of plate has to be defined in the world coordinate system. For that, we defined this task as the position of the end-effector of the first robot (which is holding the plate on one side) in the world coordinate system. Using Eqs. (4) and (5), we obtain

$$\dot{\boldsymbol{x}}_{t3} = \boldsymbol{v}_1 \tag{48}$$

and

$$\mathbf{J}_{t3} = \begin{bmatrix} \mathbf{J}_{1,v} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{49}$$

Using Eqs. (44), (45), (48) and (49) in Eq. (35), we obtain the joint velocities of the whole augmented kinematic chain

$$\dot{\boldsymbol{q}}_a = \mathbf{J}_{e1}^{\#}\dot{\boldsymbol{x}}_{e1} + \mathbf{N}_{e1}\mathbf{J}_{t3}^{\#}\dot{\boldsymbol{x}}_{t3}. \tag{50}$$

These velocities were then used to control the motion of both robot arms with a compliant low-level dynamic controller. In order to show the robustness against the external disturbances, we randomly pushed one of the arms. The perturbation torques in joint space are shown in Fig. 11. Due to the compliancy, the external torques caused motion in the null-space of the extended Jacobian, i.e. the position and the orientation around the z axis of the plate changed, but the orientations around the x and y axes of the plate were almost the same (see Fig. 10).

Experimental results show that the cooperative robots can effectively perform the given task in the presence of external disturbances.

8. Conclusions

In this paper, a new control framework for a dual-robot arm system performing a cooperative task is proposed. The task is presented as a virtual mechanism with as many DOFs as needed for the task and the task is accomplished by controlling the virtual mechanism. We propose that both robots and the task form a new augmented kinematic chain. This approach is especially suitable for systems where one robot is mounted on a moving platform. As in this formulation, there are actually two tasks: the "main" task and the position of the end-effector of the augmented kinematic chain, we have to consider this when selecting the redundancy resolution method. To control both tasks simultaneously, we used the extended task-space formulation and the corresponding extended Jacobian matrix. When the base position of one robot is not determined by the task, i.e. has not highest priority, redundancy resolution schemes for prioritized tasks can be used. Additionally, we

https://doi.org/10.1017/S0263574713000763 Published online by Cambridge University Press

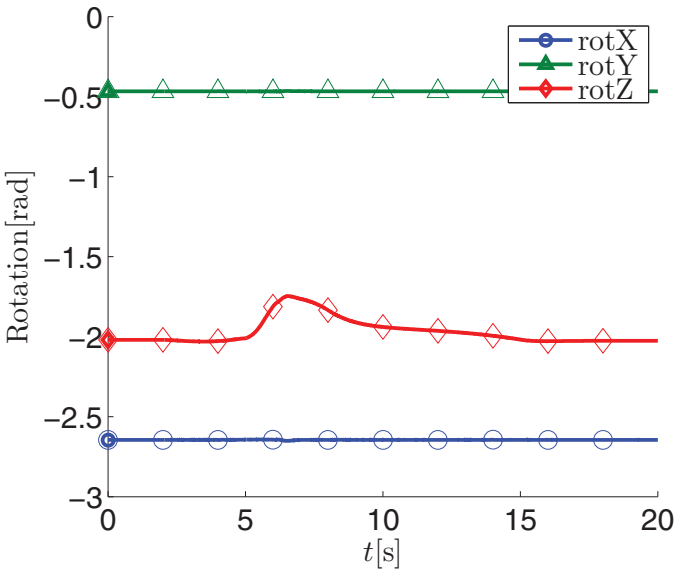

Fig. 10. (Colour online) Rotations of the plate in world coordinate space. For task of balancing the plate, only rotation around the z axis is allowed.

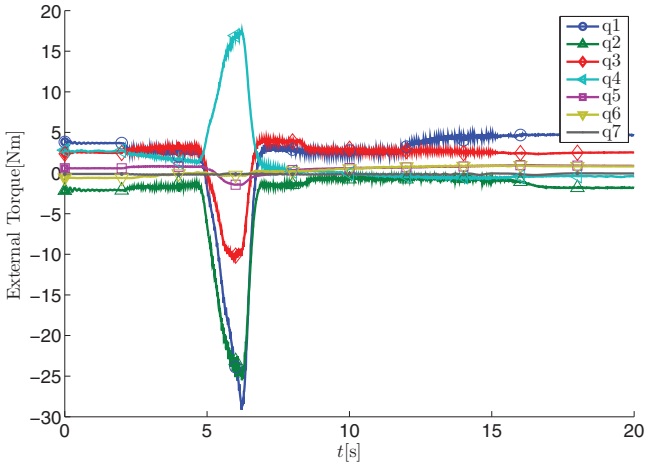

Fig. 11. (Colour online) Measured torques in joints when external perturbation was applied by human.

can easily exploit the redundancy of the augmented system for some other tasks, such as obstacle avoidance and maximizing kinematical advantage. We also present a procedure to obtain the Jacobian matrix of serially chained robot manipulators by using only the Jacobian matrices of particular robot mechanisms and their end-effector positions and orientations. The proposed approach allows chaining of serial robots together on an effective, systematic and consistent manner, which can be applied to macro–micro manipulators, reconfigurable robots, etc. Our algorithm can also cope with the systems where one (or both) of cooperative manipulator bases is moving, which is the case in many industrial and service robot applications.

References

1. M. Uchiyama and P. Dauchez, "Symmetric kinematic formulation and non-master/slave coordinated control of two-arm robots," *Adv. Robot.* **7**(4), 361–383 (1992).
2. B. V. 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), 2010 ,* Taipei, China (Oct. 2010) pp. 3955–3960.

https://doi.org/10.1017/S0263574713000763 Published online by Cambridge University Press

3. F. Caccavale, P. Chiacchio and S. Chiaverini, "Task-space regulation of cooperative manipulators," *Automatica* **36**(6), 879–887 (2000).
4. C. L. Lewis, "Trajectory Generation for Two Robots Cooperating to Perform a Task," **In**: *IEEE International Conference on Robotics and Automation, 1996*, Minneapolis, MN vol. 2 (Apr. 1996) pp. 1626–1631.
5. L. Ribeiro, R. Guenther and D. Martins, "Screw-Based Relative Jacobian for Manipulators Cooperating in a Task," **In**: *Papers presented at the Mechatronics Symposium of the 19th International Congress of Mechanical Engineering* (COBEM 2007), November 5–9, 2007, Brasilia, DF, Brazil, vol. 3 (2008) pp. 276–285.
6. B. Siciliano and L. Sciavicco, *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing, Springer-Verlag, London 2010.
7. R. M. Murray, Z. Li and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation* (CRC Press, Boca Raton, FL).
8. M. W. Spong, S. Hutchinson and M. Vidyasagar, *Robot Modeling and Control: Velocity Kinematics - The Manipulator Jacobian*, John Wiley and Sons, Inc., 2005, Hoboken, NY.
9. T. Wimboeck, C. Ott and G. Hirzinger, "Passivity-Based Object-Level Impedance Control for a Multifingered Hand," **In**: *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006,* Beijing (Oct. 2006) pp. 4621–4627
10. T. Wimbock, C. Ott and G. Hirzinger, "Analysis and Experimental Evaluation of the Intrinsically Passive Controller (IPC) for Multifingered Hands. **In**: *IEEE International Conference on Robotics and Automation, 2008,* Pasadena, CA (May 2008) pp. 278–284.
11. D. Williams and O. Khatib, "The Virtual Linkage: A Model for Internal Forces in Multi-Grasp Manipulation," **In**: *IEEE International Conference on Robotics and Automation, 1993*, Atlanta, GA vol. 1 (May 1993) pp. 1025–1030.
12. E. Usai, "Modeling and IPC control of interactive mechanical systems: A coordinate-free approach: Stefano stramigioli; lncis 266, Springer, London, 2001, ISBN 1-85233-395-2," *Automatica* **40**(1), 169–170 (2004).
13. S. Stramigioli, C. Melchiorri and S. Andreotti, A Passivity-Based Control Scheme for Robotic Grasping and Manipulation," **In**: *Proceedings of the 38th IEEE Conference on Decision and Control, 1999*, Phoenix, AZ vol. 3 (1999) pp. 2951–2956.
14. B. Nemec and L. Žlajpah, *Contemporary Robotics, Challenges and Solutions: Automatic Trajectory Generation Using Redundancy Resolution Scheme Based on Virtual Mechanism*. http://cdn.intechopen.com/pdfs/8864/InTech-Automatic_trajectory_generation_ using_redundancy_resolution_scheme_based_on_virtual_mechanism.pdf (2009) pp. 1–18.
15. N. Likar, B. Nemec and L. Žlajpah, "Virtual Mechanism Approach for Dual-Arm Manipulation," **In**: *ICINCO 2012 – Proceedings of the 9th International Conference on Informatics in Control, Automation and Robotics*, Rome, Italy, vol. 2 (Jul. 2012) pp. 321–326.
16. H. G. Tanner, K. J. Kyriakopoulos and N. J. Krikelis "Modeling of multiple mobile manipulators handling a common deformable object," *J. Robot. Syst.* **15**(11), 599–623 (1998).
17. O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg and A. Casal, "Vehicle/Arm Coordination and Multiple Mobile Manipulator Decentralized Cooperation," **In**: *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems '96, IROS 96*, Osaka, vol. 2 (Nov. 1996) pp. 546–553
18. C.-C. Wang and V. Kumar, "Velocity Control of Mobile Manipulators," **In**: *IEEE International Conference on Robotics and Automation, 1993*, Atlanta, GA vol. 2 (May 1993) pp. 713–718.
19. C. Tang, C. Xu, A. Ming and M. Shimojo, "Cooperative Control of Two Mobile Manipulators Transporting Objects on the Slope," **In**: *International Conference on Mechatronics and Automation, 2009. ICMA 2009,* Changchun (Aug. 2009), pp. 2805–2810.
20. D. Omrčen, L. Žlajpah and B. Nemec, "Autonomous motion of a mobile manipulator using a combined torque and velocity control," *Robotica* **22**(6), 623–632 (Nov. 2004).
21. F. Caccavale, V. Lippiello, G. Muscio, F. Pierri, F. Ruggiero and L. Villani, "Kinematic Control with Force Feedback for a Redundant Bimanual Manipulation System," **In**: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2011,* San Francisco, CA (Sep. 2011) pp. 4194–4200.
22. J. Baillieul, "Kinematic Programming Alternatives for Redundant Manipulators," **In**: *IEEE International Conference on Robotics and Automation. Proceedings. 1985*, St. Louis, MO, vol. 2 (Mar. 1985) pp. 722–728.
23. J. Baillieul, "Avoiding Obstacles and Resolving Kinematic Redundancy," **In**: *IEEE International Conference on Robotics and Automation. Proceedings. 1986*, San Francisco, CA, vol. 3 (Apr. 1986) pp. 1698–1704.
24. G. Schreiber, A. Stemmer and R. Bischoff, "The Fast Research Interface for the Kuka Lightweight Robot," **In**: *ICRA 2010 Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications*, Anchorage, AK (2010) pp. 15–21.
25. B. Siciliano and J.-J. E. Slotine, "A General Framework for Managing Multiple Tasks in Highly Redundant Robotic Systems," **In**: *Fifth International Conference on Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR.,* Pisa, Italy, vol. 2 (1991) pp. 1211–1216.
26. B. Nemec, L. Žlajpah and D.Omrčen, "Comparison of null-space and minimal null-space control algorithms," *Robotica* **25**(5), 511–520 (2007).
27. A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments,". *Int. J. Robot. Res.*, **4**(3), 109–117 (Sep. 1985).
28. N. Likar, Vodenje robota Kuka LWR4. Technical report, Jozef Stefan Institute (2011).

APPENDIX A

Def 1
Computing the derivative of the rotation matrix $\mathbf{R}$ is equivalent to a matrix pre-multiplication by a skew-symmetric matrix $\mathbf{S}$.[8]

$$\dot{\mathbf{R}} = \mathbf{S}(\omega)\mathbf{R}. \tag{51}$$

Def 2
Multiplying the skew-symmetric matrix with any vector $p = (p_x, p_y, p_z)^T$.[8]

$$\mathbf{S}(\omega)\mathbf{x} = \mathbf{S}(\mathbf{x})^{\mathbf{T}}\omega. \tag{52}$$

Def 3
Computing the derivative of the transposed rotation matrix $\mathbf{R}$ is equivalent to a transposed matrix post-multiplication by a skew-symmetric matrix $\mathbf{S}$.[8]

$$\dot{\mathbf{R}}^{\mathbf{T}} = -\mathbf{R}^T\mathbf{S}(\omega). \tag{53}$$

Def 4
Because rotation matrix $\mathbf{R}$ is orthogonal, it can be shown that[8]

$$\mathbf{S}(\mathbf{R}\omega) = \mathbf{R}\mathbf{S}(\omega)\mathbf{R}^T. \tag{54}$$

Def 5
An important property of the skew-symmetric matrix is linearity. For any vectors $\mathbf{a}$ and $\mathbf{b}$ belonging to $\mathbf{R}^3$, we have

$$\mathbf{S}(\mathbf{a}) + \mathbf{S}(\mathbf{b}) = \mathbf{S}(\mathbf{a} + \mathbf{b}). \tag{55}$$

APPENDIX B

Multimedia extension is available for viewing online on:
http://abr.ijs.si/interno/video.php?id=193.

https://doi.org/10.1017/S0263574713000763 Published online by Cambridge University Press