



Redundancy analysis of cooperative dual-arm manipulators

Alessandro Freddi¹, Sauro Longhi², Andrea Monteriù²,
and Davide Ortenzi²

Abstract

This paper presents the redundancy analysis of two cooperative manipulators, showing how they can be considered as a single redundant manipulator through the use of the relative Jacobian matrix. In this way, the kinematic redundancy can be resolved by applying the principal local optimization techniques used in the single manipulator case. We resolve the redundancy by using the Jacobian null space technique, which permits us to perform several tasks with different execution priority levels at the same time; this is a useful feature, especially when the manipulators are to be mounted on and cooperate with a mobile platform. As an illustrative example, we present a case study consisting of two planar manipulators mounted on a smart wheelchair, whose degrees of redundancy are employed to move an object along a pre-defined path, while avoiding an obstacle in the manipulator's workspace at the same time.

Keywords

Kinematic redundancy, dual-arm manipulators, wheelchair mounted robotic arms, assistive manipulators, null space, relative Jacobian, redundancy problem

Date received: 8 January 2016; accepted: 30 May 2016

Topic: Special Issue - Mobile Articulated Robots

Topic Editor: Fengfeng Xi

Introduction

In the last few years the study of dual-arm manipulation has become a subject of interest among the scientific community. Dual-arm manipulation has been defined in several ways, but it generally refers to the cooperation of two manipulation robotic systems that physically interact with an object, exerting forces on it in order to move or reshape it.¹

The dual-arm manipulation system was first introduced to replace workers in dangerous manufacturing processes. Early robotic manipulators were constructed by Goertz in the 1940s for handling radioactive goods.² Later, they were employed for marine and space exploration, where the dual-arm manipulators were considerably improved. In 1969, NASA's Johnson Space Center introduced anthropomorphic dual-arm teleoperators, since their performance is closer to that of human operators.³

Today, modern technological progress and the increasing acceptance of technology by users has encouraged the scientific community to focus on the development of

dual-arm manipulators that are adapted to work in user centered environments, for example surgery or ambient assisted living (AAL). An example of dual-arm surgical application concerns the "Active Project", which is a European project that exploits ICT and other engineering methods and technologies for the design and development of a dual-arm platform for neurosurgery.⁴ As for AAL applications, dual-arm manipulators could be mounted on a wheelchair in order to assist disabled people to reach and handle objects, or to perform more complex tasks.^{5,6,7,8,9}

¹ Università degli Studi eCampus, Via Isimbardi 10, 22060 Novedrate (CO), Italy

² Dipartimento di Ingegneria dell'Informazione, Università Politecnica delle Marche, Via Brecce Bianche, 60131 Ancona, Italy

Corresponding author:

Alessandro Freddi, Università degli Studi eCampus, Via Isimbardi 10, 22060 Novedrate (CO), Italy.

Email: alessandro.freddi@uniecampus.it



Cooperation between two robotic arms manipulating an object is not easy, since their relative motions have to be adequately controlled in order to perform the desired operation (i.e. solve a specific task). Moreover, the two arms and the object realize a closed kinematic chain, where the degrees of mobility of the system are greater than those generally required to perform the task, therefore the inverse kinematics problem admits an infinite number of solutions.¹⁰ For this reason the dual-arm manipulators become a kinematically redundant system during the cooperation, where the redundant variables are employed to perform tasks such as collision avoidance,¹¹ or satisfy specific performance criteria such as singularities avoidance,¹² mechanical joint limit avoidance,¹³ or improvement of manipulability along a chosen direction.¹⁴ There are several approaches in the literature that permit resolving the inverse kinematic problem for single redundant manipulators. They are mainly based on global and local optimization of a specific objective function. In particular, the global optimization permits calculation of the optimal path that satisfies a performance criterion (*off-line mode*),¹⁵ such as avoiding obstacles whose positions are known a priori. The local optimization permits calculation of the current desired joint velocity in order to locally satisfy the performance criterion (*real time mode*).¹⁶

The simplest local optimization technique is represented by the pseudo-inverse solution,¹⁷ which provides the joint velocity with the minimum norm among those which satisfy the task constraint. Since the obtained joint movement does not provide global velocity minimization along the whole path, this technique does not guarantee the avoidance of a kinematic singularity. Moreover, it does not permit the use of the redundant joints for any secondary task. This is instead possible by adopting a task augmentation technique.¹⁸ This technique consists of augmenting the task vector so as to tackle additional objectives by implementing two different methods: extended Jacobian^{19,20} and augmented Jacobian.²¹ The disadvantage of this technique is due to the presence of possible conflicts among the various tasks. Moreover, if the Jacobians associated with the tasks are linearly dependent, the task augmentation technique generates algorithmic singularities.¹⁹ In order to overcome these problems, the Jacobian null space technique was proposed in Dubey et al.²² It is based on a task priority strategy that projects the gradient of joint velocity tasks into the analytic Jacobian null space of the higher priority task, and is often referred to as the gradient projection scheme.²² Generally, the end effectors task is identified as the main task, namely the *primary task*, so that it has a higher priority with respect to other tasks (denominated secondary tasks), in order to obtain a hierarchical structure, as described in Antonelli et al.²³ Moreover, the Jacobian null space method can be implemented to manage the constraints on the joint velocities.²⁴ In this case, the redundancy is resolved so that the redundant joints, whose velocities do not violate the

constraints, are used to correctly execute the task. Since the velocities of the joints associated with the secondary tasks, obtained through the gradient projection method, generate self-motions in the robot, they do not affect the primary task motion. On the other hand, the Jacobian null space technique requires a relevant computing effort due to the Jacobian pseudo-inverse, which reduces the speed of data processing. An improved version of the gradient projection scheme, which does not require determination of the pseudo-inverse, can be found in Dubey et al.²⁵ For the case of rank analytic Jacobians, it is possible to implement the reduced gradient method described in De Luca and Oriolo.²⁶

In the literature, several examples of null space algorithms are implemented on single redundant manipulators that are individually controlled. Žlajpah et al. present a highly kinematic redundant planar manipulator that is able to avoid obstacles online in an unstructured environment by using a mono-dimensional obstacle avoidance task.¹¹ Park et al. propose a method to accommodate multiple tasks for redundancy utilization, which is based on a specific weighted pseudo-inverse.²⁷

Since the control of cooperative manipulators is more complex than the control of a single manipulator, it is useful to employ the method based on the relative Jacobian matrix.^{28,29} This method permits the consideration of two redundant manipulators like a unique redundant manipulator, whose number of joints is equal to the sum of the joints relative to each manipulator, while the end effector motion variables correspond to the relative motion between the two grippers. This method presents more advantages than those based on the individual control of each manipulator. First of all, a dual-arm system modeled according to the relative Jacobian method can be controlled by the same algorithms used for controlling single manipulators. Secondly, the compact expression of the relative Jacobian matrix is simple to calculate when Jacobians of the individual manipulators are known and, if a manipulator is replaced with another one, it is sufficient to change the respective Jacobian without recalculating the entire relative Jacobian matrix.²⁸

A dual manipulator system can be seen as a redundant system, and this paper is focused on the resolution of redundancy by projecting the maximum number of the secondary tasks into the Jacobian null space of the primary task, as in the single manipulator case. The main contribution is to control a dual-arm manipulation system as a single redundant manipulator through the use of the relative Jacobian, exploiting the degrees of redundancy to perform secondary tasks by using the Jacobian null space technique. The proposed approach permits the use of a dual-arm manipulation system to perform a primary task (e.g. hold an object, as presented in the case study) by using a simple control law which can account for secondary tasks as well (e.g. obstacle avoidance in the case study) as long as the system has one or more redundancy degrees. The capability of performing tasks with different execution priorities levels and the

possibility of controlling the dual-arm manipulators as a single manipulator is useful, especially in the case of cooperation with a mobile base.

The paper has the following structure. Section 2 presents the theoretical background on the resolution of the kinematic redundancy problem based on the null space of the relative Jacobian. Section 3 shows a case study, in order to evaluate the system performances with different dimensions of the null space. Conclusions and future work complete the paper in Section 4.

Redundancy resolution by Jacobian null space method

In this section we revise the main literature results that are necessary to formulate the kinematic inverse solution for cooperative manipulators. Let us first recall the definition of redundancy for robotic manipulators.

Kinematic redundancy: a manipulator is *kinematically redundant* when the number of its possible motions n (degree of motion) is higher than the number of variables m that are necessary to describe a given task (dimension of the task space), $n > m$.³⁰

Intrinsic redundancy: a manipulator is *intrinsically redundant* when its degree of motion n is greater than the dimension of the space in which the manipulator operates (dimension of the operational space s), $n > s$.

Functional redundancy: a manipulator is *functionally redundant* when its degree of motion is equal to the dimension of the operational space, that is $n = s$, and the dimension of the operational space is greater than that of the task space, $s > m$ (from which it follows $n > m$).

The first definition depends on the specific task, which implies that the same manipulator can be redundant with respect to a specific task and non-redundant with respect to another. It represents the main definition of redundancy, with the other two definitions referring to specific cases. In detail, the second definition does not depend on the task, since it is based on the intrinsic kinematic structure of the robot; it follows from the definition of kinematic redundancy by considering the operational space rather than the task space, whose dimension is always equal or less than that of the operational space. The third definition, instead, refers to the specific case where the number of possible motions of the manipulator is equal to the dimension of the operational space, and redundancy originates from the fact that the task is defined in a space with fewer dimensions than that of the operational space (e.g. the operational space has three dimensions, but the task is planar, which implies a task space in two dimensions). The following example provides a better understanding of the differences between the three redundancies. Consider a planar manipulator with 4 joints, which has to translate and rotate an object ($m = 3$) on

a plane ($s = 3$). Since in a generic open-chain manipulator the degree of motion n is equal to number of its joints, we have $n = 4$, and the considered planar manipulator results intrinsically redundant, that is $n > s$. Therefore, the manipulator can reach the desired end effector pose by assuming infinite configurations. On the contrary, a planar manipulator with 3 joints, $n = 3$, which has to translate an object on a plane ($m = 2, s = 3$) is functionally redundant, because it can perform the task by assuming an arbitrary end effector orientation.

In the following, the generalized form of the Jacobian pseudo-inverse solution for a single redundant manipulator is derived, then the analytic Jacobian is replaced by the relative Jacobian. The kinematic inverse solution for cooperative manipulators is presented at the end of the section.

Redundancy resolution for a single manipulator

Given an open chain manipulator with n joints and a task described by m variables in the operative space, it is possible to calculate the end effector velocity vector $\dot{\vec{x}}_d(t)$ necessary to perform the task by multiplying the joint velocity vector $\dot{\vec{q}}(t)$ with a linear transformation matrix, which depends on the current manipulator configuration $\vec{q}(t)$.³⁰ If the end effector velocity vector is defined in terms of linear velocity $\dot{\vec{p}}(t)$ and angular velocity $\dot{\vec{\omega}}(t)$, the transformation matrix is *geometric Jacobian* $\mathbf{J}(\vec{q}(t))$, and the end effector velocity vector is

$$\dot{\vec{x}}_d(t) = \begin{bmatrix} \dot{\vec{p}}(t) \\ \dot{\vec{\omega}}(t) \end{bmatrix} = \mathbf{J}(\vec{q}(t))\dot{\vec{q}}(t) \quad (1)$$

Otherwise, if the end effector orientation velocity $\dot{\vec{\phi}}(t)$ is expressed as the derivative of a minimal orientation representation in the operational space (i.e. Euler angles) with respect to time, the transformation matrix is *analytic Jacobian* $\mathbf{J}_A(\vec{q}(t))$,³⁰ and the end effector velocity vector is

$$\dot{\vec{x}}_d(t) = \begin{bmatrix} \dot{\vec{p}}(t) \\ \dot{\vec{\phi}}(t) \end{bmatrix} = \mathbf{J}_A(\vec{q}(t))\dot{\vec{q}}(t) \quad (2)$$

Since the dimension of the end effector velocity vector requested by task, namely $\dim(\dot{\vec{x}}_d(t))$, is equal to m (where in three-dimensional space $m \leq 6$) and the dimension of the joints velocity vector $\dot{\vec{q}}(t)$ is equal to n , the dimension of the analytic Jacobian matrix is equal to $m \times n$. In the case of non-redundant manipulators, $m = n$ and the Jacobian matrix is a square matrix. Otherwise, in the case of redundant manipulators $m < n$, the Jacobian matrix is a lower rectangular matrix. The two different Jacobian matrix forms become relevant to calculate the inverse kinematic solutions that are studied in the following.

Kinematic inverse solution for a non-redundant manipulator. Since the task variables are generally defined in the operative space, kinematic inverse algorithms are necessary to obtain the manipulator joint velocity. When the manipulator

is not redundant, the joint velocity vector $\dot{\vec{q}}(t)$ can be calculated simply by inverting the Jacobian

$$\dot{\vec{q}}(t) = \mathbf{J}_A^{-1}(\vec{q}(t))\dot{\vec{x}}_d(t) \quad (3)$$

If the initial joint position $\vec{q}(0)$ is known, it is possible to calculate the new manipulator configuration $\vec{q}(t)$ by time integration of the joint velocity

$$\vec{q}(t) = \int_0^t \dot{\vec{q}}(s)ds + \vec{q}(0) \quad (4)$$

Since the kinematic inverse algorithms generally run on digital processors, the joint position is discretized into $\vec{q}(t_k)$ (where k indicates the k th discrete time instant) by discrete-time numerical approximation of the continuous-time integral shown in equation (4). In the following the mathematical approach will be based on discrete-time. In order to obtain an accurate discrete-time approximation of equation (4), a high-order algorithm is required. However, a high-order algorithm implicates an undesired large finite time delay in real time applications, which can be reduced by shortening the time step Δt .²¹ A good choice consists of a first order algorithm as the Euler forward rectangular method, which can give an acceptable accuracy of the numerical integration with suitable Δt value. The Euler forward rectangular method is used to transform the integral shown in equation (4) in the following recursive form

$$\vec{q}_k \approx \vec{q}_{k-1} + \mathbf{J}_A^{-1}(\vec{q}_{k-1})\dot{\vec{x}}_{d_{k-1}}\Delta t \quad (5)$$

where $\dot{\vec{q}}_k$ indicates the calculated joint velocity at the time instant k , while $\dot{\vec{x}}_{d_{k-1}}$ is the desired end effector motion at the time instant $k - 1$. In spite of the kind of implemented interpolation, the end effector pose \vec{x}_k obtained by direct kinematic functions is different from the desired end effector pose \vec{x}_{d_k} , because any numerical integration is mainly affected by two sources of errors. The first consists of an unavoidable drifting error, which increases at each numerical integration step, while the second depends on uncertainty in the initial value of the joint position. Thus, the pose error \vec{e}_k can be defined as

$$\vec{e}_k = \vec{x}_{d_k} - \vec{x}_k \quad (6)$$

Defining the joint velocity vector in the function of the pose error of the end effector pose $\dot{\vec{q}}_k(\vec{e}_k)$, it is possible to ensure the convergence of the pose error to zero. This choice permits us to find several inverse kinematics algorithms which are based on the use of a feedback correction term called closed-loop inverse kinematics (CLIK).³¹ A well-known first-order kinematic algorithm is the Jacobian inverse, which permits calculation of $\dot{\vec{q}}_k$ as follows³¹

$$\dot{\vec{q}}_k = \mathbf{J}_A^{-1}(\vec{q}_k)(\dot{\vec{x}}_{d_k} + \mathbf{K}\vec{e}_k) \quad (7)$$

where \mathbf{K} is a constant positive-definite gain matrix. Since $\dot{\vec{q}}_k = \mathbf{J}_A^{-1}(\vec{q}_k)\dot{\vec{x}}_k$, equation (7) can be rewritten as follows

$$\mathbf{J}_A^{-1}(\vec{q}_k)(\dot{\vec{e}}_k + \mathbf{K}\vec{e}_k) = \mathbf{0} \rightarrow \lim_{k \rightarrow \infty} \|\vec{e}_k\|^2 = 0 \quad (8)$$

where the velocity of the pose error convergence depends on the eigenvalues of matrix \mathbf{K} , so that higher eigenvalues correspond to faster pose error convergence. However, the convergence velocity cannot be chosen arbitrarily due to the limitation of band given by the sample time Δt . Finally, if the initial error pose is equal to zero, that is $\vec{e}_{k=0} = 0$, then the feed-forward action ensures a zero error along the whole trajectory.

Kinematic inverse solution for a redundant manipulator. In the case of a redundant manipulator, the previous inverse Jacobian algorithm cannot be implemented, because the Jacobian matrix is a lower rectangular matrix ($m < n$) and its inversion admits multiple solutions. For this reason, a criterion of solution choice should be adopted. If the Jacobian matrix has full rank, the solution is based on the right pseudo-inverse of the Jacobian matrix.³⁰ This allows us to obtain the solution that locally minimizes the norm of the joint velocity, in accordance with Moore–Penrose properties.

The right pseudo-inverse of the Jacobian matrix $\mathbf{J}_A^\dagger(\vec{q}_k)$ is obtained as follows

$$\mathbf{J}_A^\dagger(\vec{q}_k) = \mathbf{J}_A^T(\vec{q}_k)(\mathbf{J}_A(\vec{q}_k)\mathbf{J}_A^T(\vec{q}_k))^{-1} \quad (9)$$

Replacing the inverse Jacobian matrix with its pseudo-inverse in equation (7), it is possible to obtain the CLIK pseudo-inverse expression

$$\dot{\vec{q}}_k^* = \mathbf{J}_A^\dagger(\vec{q}_k)(\dot{\vec{x}}_{d_k} + \mathbf{K}\vec{e}_k) \quad (10)$$

where $\dot{\vec{q}}_k^*$ is the joint velocity solution that satisfies the criteria of choice at time instant k . Such a solution presents two main advantages: it reduces the joint velocity close to singularity configuration, and it allows us to minimize the norm of the joint velocities. Thus from an engineering point of view, this saves electric energy when the manipulator is mounted on an electric mobile vehicle (e.g. a smart wheelchair).

However, the simple CLIK pseudo-inverse expression does not allow us to manage the redundant joints (joints not required by the task), which could be used to perform other secondary tasks that must not affect the performance of the primary task. A possible strategy consists of projecting the redundant joints velocity vector $\dot{\vec{q}}_k^+$ in the Jacobian null space, which is an r -dimensional orthogonal complement of the Jacobian image space. In the case of a redundant manipulator, r is non-zero and its value can be calculated via the rank-nullity theorem³⁰ as follows

$$\dim[\text{Im}(\mathbf{J}_A(\vec{q}_k))] + \dim[\text{Ker}(\mathbf{J}_A(\vec{q}_k))] = \dim[\dot{\vec{q}}_k] \quad (11)$$

where $\dim[\text{Im}(\mathbf{J}_A(\vec{q}_k))]$, $\dim[\text{Ker}(\mathbf{J}_A(\vec{q}_k))]$ and $\dim[\dot{\vec{q}}_k]$ are the dimensions of the Jacobian image space, the Jacobian null space and the joints velocity vector space, respectively. Remembering that $\dim[\text{Im}(\mathbf{J}_A(\vec{q}_k))] = \dim[\dot{\vec{x}}_d] = m$ and $\dim[\dot{\vec{q}}_k] = n$, r can be immediately calculated to be

$$\dim[\text{Ker}(\mathbf{J}_A(\vec{q}_k))] = n - m = r \quad (12)$$

The maximum number of tasks that can be executed simultaneously by the system depends on the dimensions of each task space,²³ that is if m_i is the dimension of the i th task space, then

$$\sum_{i=1}^l m_i = n \quad (13)$$

where l indicates the maximum number of tasks that can be fulfilled simultaneously by the cooperative manipulators (namely one primary task) and $l - 1$ indicates the secondary tasks executable at the same time. Since a secondary task may influence the primary task motion, it is necessary to project it into the null space of the primary task by orthogonal projector matrix \mathbf{P} , such that

$$\mathbf{J}_A(\vec{q}_k) \mathbf{P}_k \dot{\vec{q}}_k^+ = \mathbf{0} \quad (14)$$

According to the right pseudo-inverse Jacobian, an $n \times n$ matrix \mathbf{P} can be obtained at each time instant by the following expression

$$\mathbf{P}_k = \mathbf{I} - \mathbf{J}_A^\dagger(\vec{q}_k) \mathbf{J}_A(\vec{q}_k) \quad (15)$$

where \mathbf{I} indicates the identity matrix of suitable dimensions. Therefore, it is possible to add in equation (10) the term relative to the projection of $\dot{\vec{q}}_k^+$

$$\dot{\vec{q}}_k^* = \mathbf{J}_A^\dagger(\vec{q}_k)(\dot{\vec{x}}_{d_k} + \mathbf{K} \vec{e}_k) + \mathbf{P}_k \dot{\vec{q}}_k^+ \quad (16)$$

where the $\dot{\vec{q}}_k^*$ solution is different from the one obtained from equation (10), because now it is the minimal norm joint velocity solution that satisfies both the primary task and the secondary tasks. Since the $\dot{\vec{q}}_k^+$ vector can also be used to execute some objective function $w(\vec{q}_k)$, such as obstacle avoidance, it is also possible to use the gradient projection scheme.²² This method permits the projection of the gradient of a specific objective function $\nabla w(\vec{q}_k)$ (calculated in joint space) into Jacobian null space, as shown below

$$\dot{\vec{q}}_k^* = \mathbf{J}_A^\dagger(\vec{q}_k)(\dot{\vec{x}}_{d_k} + \mathbf{K} \vec{e}_k) + \mathbf{P}_k(k_a \nabla_q w(\vec{q}_k)) \quad (17)$$

where k_a is a real scalar value which indicates the gain, and is positive when $w(\vec{q}_k)$ has to be maximized. Otherwise it is a negative value if $w(\vec{q}_k)$ has to be minimized.²² However, the choice of k_a is critical for the performance of the redundancy resolution. In particular, a small step size may slow down the minimization of the performance criterion $w(\vec{q}_k)$. At the same time, a large value may even lead to an increase in it. An appropriate value of k_a could be chosen by using a simplified line search technique as described in Luenberger and Ye.³²

Kinematic inverse solution for cooperative manipulators

When two manipulators interact on an object at the same time, they make a closed-chain among the three parts. In this case, the task is generally executed by the relative end effector motions, which require fewer motion variables

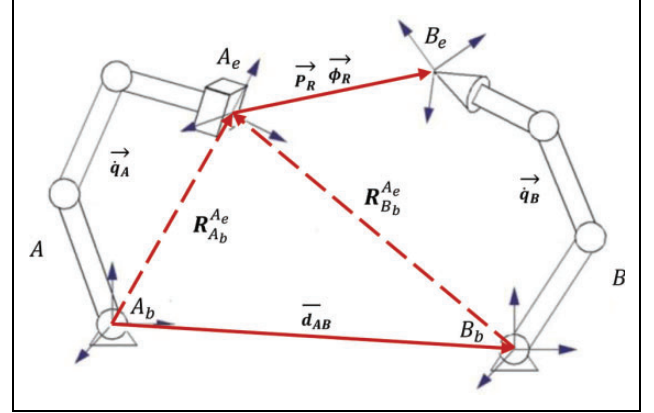


Figure 1. Two cooperative manipulators.

with than those provided by two independent manipulators. Alternatively, the two cooperative manipulators can be seen as a single redundant manipulator, having the same degree of motion of the dual arms and the same number of end effector motion variables required by the task. In this way, it is possible to implement the same control algorithm adopted in the single redundant manipulator case, with the only difference being a redefined Jacobian matrix. The Jacobian matrix associated with the equivalent manipulator depends on the two individual analytic Jacobians possessed by each manipulator, and is called the *relative Jacobian*.

The relative Jacobian matrix. Two cooperative manipulators A and B are shown in Figure 1, where each possesses a number of joints, n_a and n_b respectively. The frame integral to the base of manipulator B , namely B_b , is placed on a fixed Cartesian distance \vec{d}_{AB} from the other frame integral to the base of manipulator A_b . Their relative rotation is described by the rotation matrix $\mathbf{R}_{B_b}^{A_b}$, which describes the set of three consecutive rotations that the frame A_b must undergo to have the same orientation of the frame B_b . The frame integrals to the end effector of manipulators A and B , namely A_e and B_e , indicate the Cartesian position between the two manipulators. In particular the B_e position is defined with respect to A_e by the \vec{P}_R vector, while their relative rotation is described by the $\vec{\phi}_R$ vector. Differentiating the \vec{P}_R and $\vec{\phi}_R$ vectors with respect to time, it is possible to obtain a unique vector $\dot{\vec{x}}_{Rd_k}$, which contains the desired relative end effector motion variables at time instant k

$$\dot{\vec{x}}_{Rd_k} = \begin{bmatrix} \dot{\vec{P}}_R \\ \dot{\vec{\phi}}_R \end{bmatrix} \quad (18)$$

where $\dot{\vec{x}}_{Rd_k}$ is an m_{ab} -dimensional column vector ($m_{ab} \leq 6$ in a three-dimensional space).

Thanks to the kinematic relations between the two manipulators, it is possible to obtain a single equivalent manipulator with a number of joints equal to $n_{ab} = n_a + n_b$ and end effector motion variables defined by the $\dot{\vec{x}}_{Rd_k}$



Figure 2. Concept of the wheelchair with two manipulators within an AAL scenario.

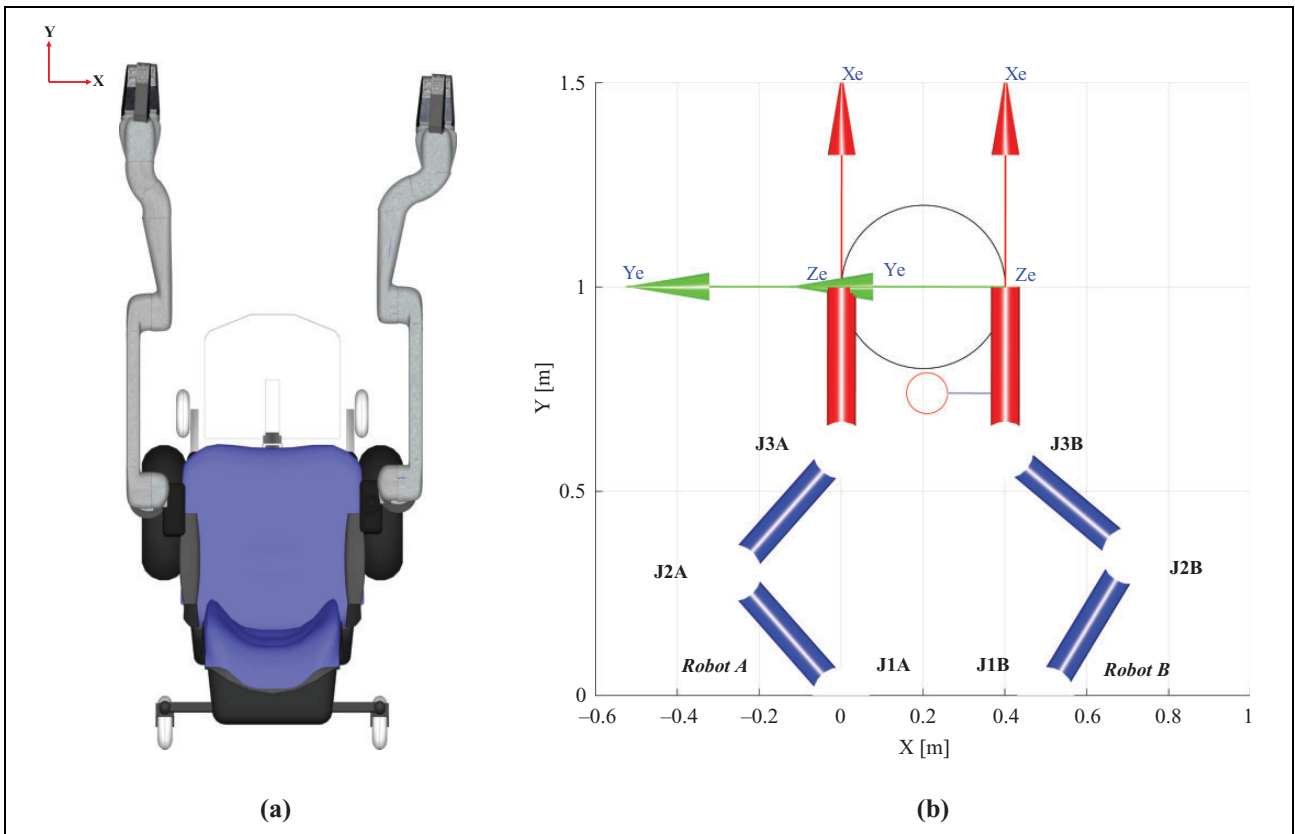


Figure 3. Top view of the system (a) and initial planar manipulators configuration (b).

vector. As described in the single manipulator case, it is possible to define a differential kinematic relation between the end effector motion variables $\dot{\vec{x}}_{Rd_k}$ and the joints velocity indicated by the n_{ab} -dimensional column vector $\vec{q}_{ab_k} = [\vec{q}_{a_k}, \vec{q}_{b_k}]^T$

$$\dot{\vec{x}}_{Rd_k} = \mathbf{J}_R(\vec{q}_{ab_k}) \dot{\vec{q}}_{ab_k} \quad (19)$$

where $\mathbf{J}_R(\vec{q}_{ab_k})$ is the $m \times n$ -dimensional relative Jacobian matrix associated to the two manipulators' cooperative

motion. In accordance with Jamisola and Roberts,²⁹ $\mathbf{J}_R(\vec{q}_{ab_k})$ can be expressed as

$$\mathbf{J}_R(\vec{q}_{ab_k}) = \begin{bmatrix} -\Psi_{B_c}^{A_c} \Omega_{A_b}^{A_c} \mathbf{J}_A(\vec{q}_{a_k}) & \Omega_{B_b}^{A_c} \mathbf{J}_B(\vec{q}_{b_k}) \end{bmatrix} \quad (20)$$

with \mathbf{J}_A and \mathbf{J}_B indicating the analytic Jacobians of the standalone manipulators, where $\Psi_{B_c}^{A_c}$ is the wrench transformation matrix,²⁹ while $\Omega_{A_b}^{A_c}$ and $\Omega_{B_b}^{A_c}$ are diagonal matrices, which respectively depend on the rotation matrices $\mathbf{R}_{A_b}^{A_c}$ and $\mathbf{R}_{B_b}^{A_c}$ evaluated at time k

$$\Omega_{A_b}^{A_c} = \begin{bmatrix} R_{A_b}^{A_c} & \mathbf{0} \\ \mathbf{0} & R_{A_b}^{A_c} \end{bmatrix} \quad (21)$$

$$\Omega_{B_b}^{A_c} = \begin{bmatrix} R_{B_b}^{A_c} & \mathbf{0} \\ \mathbf{0} & R_{B_b}^{A_c} \end{bmatrix} \quad (22)$$

The wrench transformation matrix $\Psi_{B_c}^{A_c}$ permits compensation of the relative end effectors translation velocity component due to the cross product of the end effector A orientation velocity $\vec{\phi}_A$, with the end effector's distance vector \vec{P}_R described by the following equation

$$\Psi_{B_c}^{A_c} = \begin{bmatrix} \mathbf{I} & [\vec{P}_R]_{\times} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \quad (23)$$

where $[\vec{P}_R]_{\times}$ is the skew matrix generated by vector \vec{P}_R . In particular, high values of $\vec{\phi}_A$ increase the relative end effectors position error, as demonstrated in Jamisola and Roberts.²⁹ When $\vec{\phi}_A$ is low, its contribution to the relative translational velocity is negligible, and the wrench transformation matrix is approximated with the identity matrix, as commonly assumed in the literature.^{33,34}

The relative Jacobian null space. Since the relative Jacobian matrix is associated with the cooperative dual arms, it can define its equivalent manipulator with a number of joints equal to n_{ab} . Since $n_{ab} > m_{ab}$, the equivalent manipulator results kinematically redundant with respect to the relative motion task. In accordance with equation (10), in this case the joints velocity vector \vec{q}_{abk} can be obtained by the CLIK algorithm based on the right Jacobian pseudo-inverse

$$\dot{\vec{q}}_{abk} = J_R^{\dagger}(\vec{q}_{abk})(\dot{\vec{x}}_{Rd_k} + K\vec{e}_{R_k}) \quad (24)$$

where J_A^{\dagger} has been replaced with J_R^{\dagger} , and e_{R_k} indicates the error between the desired relative end effectors position \vec{x}_{Rd_k} and the relative end effectors position \vec{x}_{R_k}

$$\vec{e}_{R_k} = \vec{x}_{Rd_k} - \vec{x}_{R_k} \quad (25)$$

In order to obtain the joints velocity vector $\dot{\vec{q}}_{abk}$, the redundant joints velocity \vec{q}_k^+ can be projected into the Jacobian null space by the orthogonal projector matrix $P_{R_k} = \mathbf{I} - J_R^{\dagger} J_R$

$$\dot{\vec{q}}_{abk} = J_R^{\dagger}(\vec{q}_{abk})(\dot{\vec{x}}_{Rd_k} + K_R \vec{e}_{R_k}) + P_{R_k} \dot{\vec{q}}_k^+ \quad (26)$$

where K_R is the gain matrix which allows the relative end effectors pose error \vec{e}_R to converge to zero, while the dimension of the relative Jacobian null space is equal to $r_{ab} = n_{ab} - m_{ab}$. However, the calculation of r_{ab} does not consider the m -dimensional end effector velocity vector, which describes the motion of each manipulator in the operative space (i.e. $m \leq 6$ in a three-dimensional space). In particular, the end effector velocity vector describes the translational and rotational motion of each end effector in the space, and these movements are very useful for many

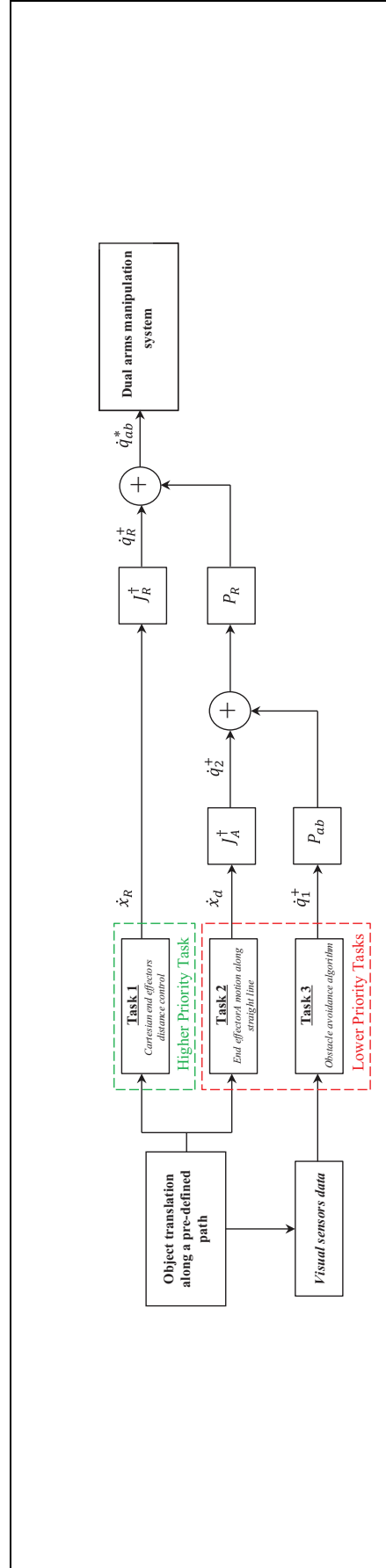


Figure 4. Block diagram of two cooperative manipulators.

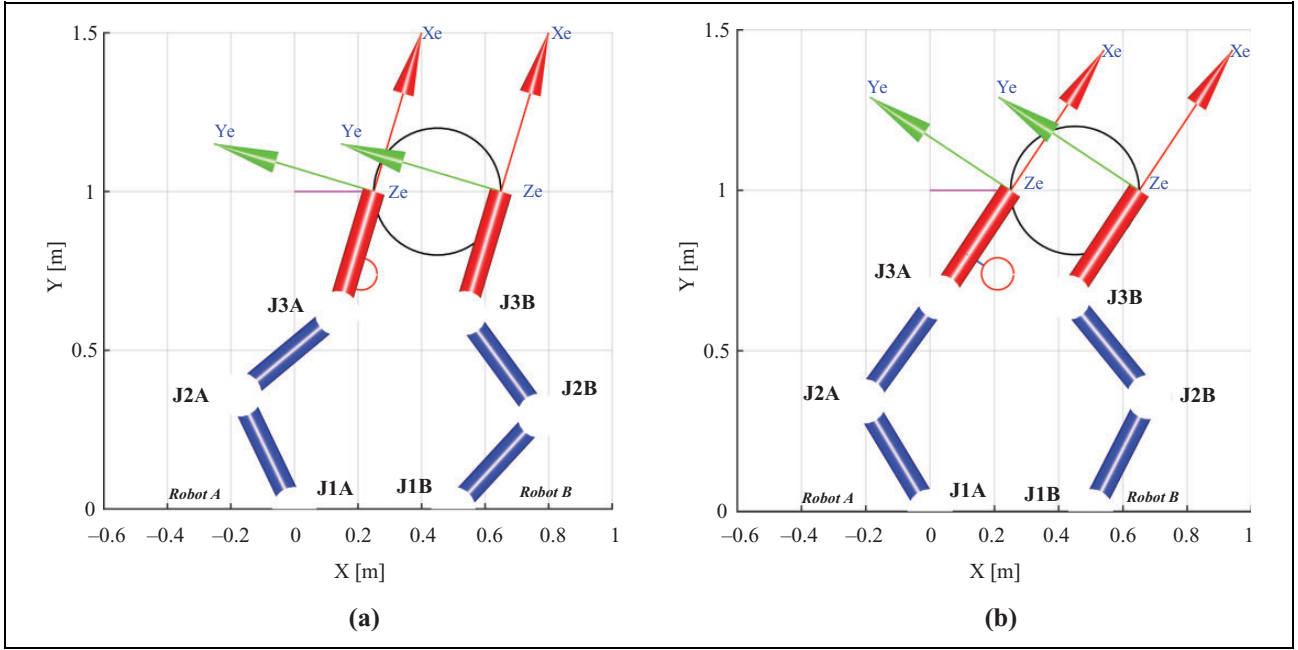


Figure 5. Final manipulators configuration without (a) and with (b) the obstacle avoidance task.

secondary tasks. Since the two end effectors are constrained by $\dot{\vec{x}}_{Rd_k}$, it is sufficient to assign $\dot{\vec{x}}_d$ to only one end effector (e.g. A manipulator) in order to move the whole dual-arm system in the space. Moreover, projecting the A manipulator joint velocity vector $\dot{\vec{q}}_{A_k}^+$ in the null space of the relative Jacobian, the primary task performance is not affected, thus

$$\dot{\vec{q}}_{ab_k} = \mathbf{J}_R^\dagger(\vec{q}_{ab_k})(\dot{\vec{x}}_{Rd_k} + K_R \vec{e}_{R_k}) + P_{R_k}(\mathbf{J}_A^\dagger(\vec{q}_{a_k}, \vec{q}_{b_k})\dot{\vec{x}}_{d_k}) \quad (27)$$

with $\mathbf{J}_A(\vec{q}_{a_k}, \vec{q}_{b_k}) = [\mathbf{J}_A(\vec{q}_{a_k}), \mathbf{J}_A(\vec{q}_{b_k})]$. In the case of $r_{ab} - (m_{ab} + m) > 0$ in accordance with equation (13), it is possible to add other lower priority tasks with maximum dimensional value equal to $m^* = r_{ab} - (m_{ab} + m)$, in order to obtain a hierarchical structure of the tasks. Generally, given several secondary tasks with different priority levels, it is possible to obtain the $\dot{\vec{q}}_{ab_k}^*$ vector by projecting the joint velocity vector $\dot{\vec{q}}_i^+$ of the i -th task into the null space relative to the higher priority task. Defining $\dot{\vec{q}}_2^+ = \mathbf{J}_A^\dagger(\vec{q}_{a_k}, \vec{q}_{b_k})\dot{\vec{x}}_{d_k}$ and a generic joint velocity task $\dot{\vec{q}}_1^+$ (having dimensions equal to n_{ab}), the general expression for the execution of the three tasks is

$$\dot{\vec{q}}_{ab_k}^* = \mathbf{J}_R^\dagger(\vec{q}_{ab_k})\dot{\vec{x}}_{Rd_k} + P_{R_k}(\dot{\vec{q}}_2^+ + P_{ab_k}\dot{\vec{q}}_1^+) \quad (28)$$

Where P_{ab_k} is a $n_{ab} \times n_{ab}$ -dimensional matrix that contains the orthogonal projectors in the null spaces of the manipulators A and B

$$P_{ab_k} = \begin{bmatrix} P_{a_k} & \mathbf{0} \\ \mathbf{0} & P_{b_k} \end{bmatrix} \quad (29)$$

Case study

Task description

This section describes a case study which consists of the development of a kinematic control based on the relative Jacobian matrix of two cooperative Jaco manipulators,³⁵ A and B , mounted on the left and right side of a smart wheelchair, as shown in Figure 2. In this scenario, the user can autonomously reach any indoor point by using the smart wheelchair, thanks to a simultaneous localization and map-building algorithm.^{36,37,38} The two cooperative manipulators add manipulation capability to the wheelchair in order to transport objects along a pre-defined path. Since the aim of this paper is focused on the analysis of the kinematic performance of a dual-manipulation system, the wheelchair motion is not considered during the execution of the manipulation tasks. Moreover the case study is faced in the planar case. The reader can refer to the concluding section for an explanation of possible future work, which takes into account cooperation between the wheelchair and the manipulators and a possible extension to the three-dimensional case.

The considered task is common in a daily scenario: during a meal the wheelchair user commands the two manipulators to move his/her dish from point \vec{p}_1 to point \vec{p}_2 of the table. Moreover, the manipulators must avoid a nearby object (i.e. a bottle placed between them). The task is modeled in the XY -plane as follows:

1. The motion of the dish is along a straight line in the XY -plane, which is a common path to choose in

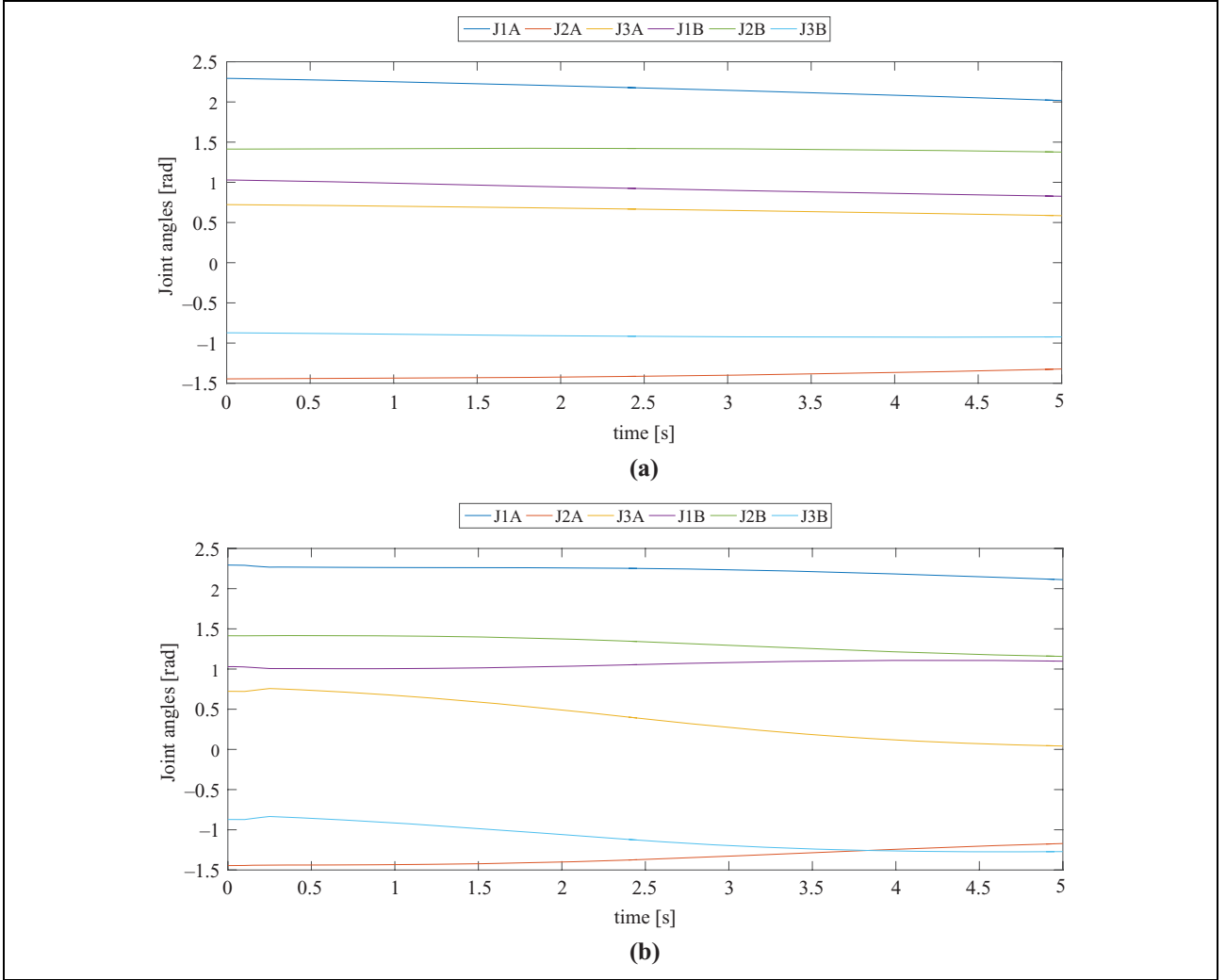


Figure 6. Joint angle position without (a) and with (b) the obstacle avoidance task.

- order to reduce both the time of the task execution and the motor energy consumption.
2. The dish is represented by a circle of ray r_1 .
 3. The bottle is represented by a circle of ray r_2 .
 4. The two anthropomorphic manipulators (Figure 3(a)) are modeled as two planar manipulators having three joints each ($n_a = n_b = 3$), which grasp the dish so that the Cartesian distance between the two end effectors is equal to the diameter of the dish (Figure 3(b)).
 5. The end effector A has a low orientation velocity, thus the wrench transformation is approximated by the identity matrix.

According to the Jacobian null space method, the proposed scenario could be decomposed into two different priority level tasks: the higher priority task (primary task) and the lower priority task (secondary task). In particular, the primary task ensures the grasping of the dish during all the motion time, that is the distance $\|\vec{p}_R\|$ has to be maintained constant and, consequently, its derivative with

respect to time has to be equal to zero, $\dot{\vec{p}}_R = [0, 0]^T$, so that the desired relative end effector's orientation velocity $\dot{\vec{\phi}}_R = 0$. Therefore, the relative end effector's velocity vector $\dot{\vec{x}}_{Rd_k}$ can be defined as

$$\dot{\vec{x}}_{Rd_k} = [0 \ 0 \ 0]^T \quad (30)$$

where $\dim[\dot{\vec{x}}_{Rd_k}] = m_{ab} = 3$. Therefore, it is possible to achieve an equivalent manipulator which possesses $n_{ab} = n_a + n_b = 6$ joints and $m_{ab} = 3$ end effector motion variables, thus the degree of redundancy is $r_{ab} = n_{ab} - m_{ab} = 3$ in accordance with equation (11).

The degree of redundancy is used to execute the first secondary task, which defines the translation of the dish between two points. It is obtained by projecting a desired Cartesian velocity $\dot{\vec{x}}_{d_k}$ relative to end effector A (or equivalently to end effector B) in the relative Jacobian null space in accordance with equation (25). In fact, since the relative motion between end effectors is established by $\dot{\vec{p}}_R = [0, 0]^T$, the motion of only one end effector implies the motion of the whole dual-arm system. Since the rotation

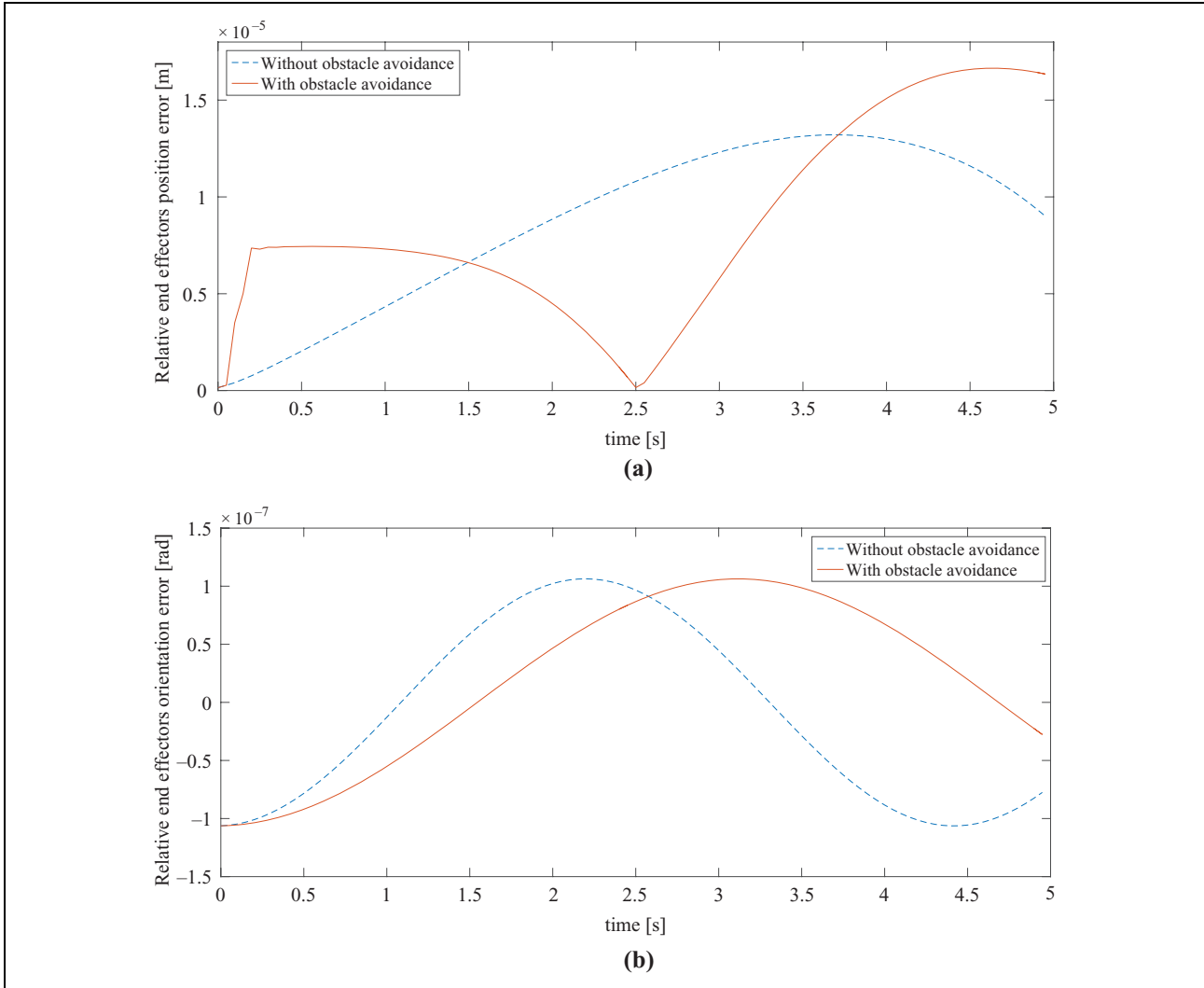


Figure 7. Relative end effectors position error (a) and orientation error (b).

of both end effectors is not important for the correct execution of this task, it is left unspecified in order to have $\dim[\dot{\vec{x}}_{d_k}] = m_a = 2$. If the end effector A is chosen, then the analytic Jacobian associated to it is a lower rectangular matrix \mathbf{J}_A ($\dim[\mathbf{J}_A] = 2 \times 3$).

Therefore, it is possible to obtain the 6-dimensional joints velocity vector $\dot{\vec{q}}_2^+$ relative to the secondary task by using the CLIK pseudo-inverse expression as described in equation (10)

$$\dot{\vec{q}}_2^+ = \mathbf{J}_A^\dagger(\vec{q}_{a_k}, \vec{q}_{b_k})(\dot{\vec{x}}_{d_k} + \mathbf{K}\vec{e}_k) \quad (31)$$

where $\mathbf{J}_A^\dagger(\vec{q}_{a_k}, \vec{q}_{b_k}) = [\mathbf{J}_A(\vec{q}_{a_k})\mathbf{0}]^\dagger$, due to the fact that the manipulator A has been chosen.

The final degree of redundancy of the dual-arm system r_{ab} results in

$$r_{ab} = n_{ab} - m_{ab} - m_a = 1 \quad (32)$$

From equation (32), the dual-arm system still has one degree of redundancy, which can be used to execute a third

secondary task such as obstacle avoidance. In particular, the dual-arm system is able to avoid an obstacle of unknown position, which could be detected, for instance, by using a vision system.

The obstacle avoidance task is obtained by using a projecting gradient method.²² This method permits increasing the distance between the obstacle and the nearest manipulator by projecting the joint space velocity relative to the third task in the null space of the secondary task. Since $m_a = 2$, the third task does not affect the translation of the end effectors, but only their orientation, which is not important for the correct execution of the higher priority task. The third task can be performed by an algorithm that operates in two steps. In the first step, the algorithm detects the arm closest to the obstacle, by iteratively calculating the minimum distance vector \vec{w} between $\vec{a}(\vec{q}_{a_k})$, $\vec{a}(\vec{q}_{b_k})$ (the position vectors of two generic points along the structure manipulators) and \vec{b} (the position vector of a suitable point on the obstacle)

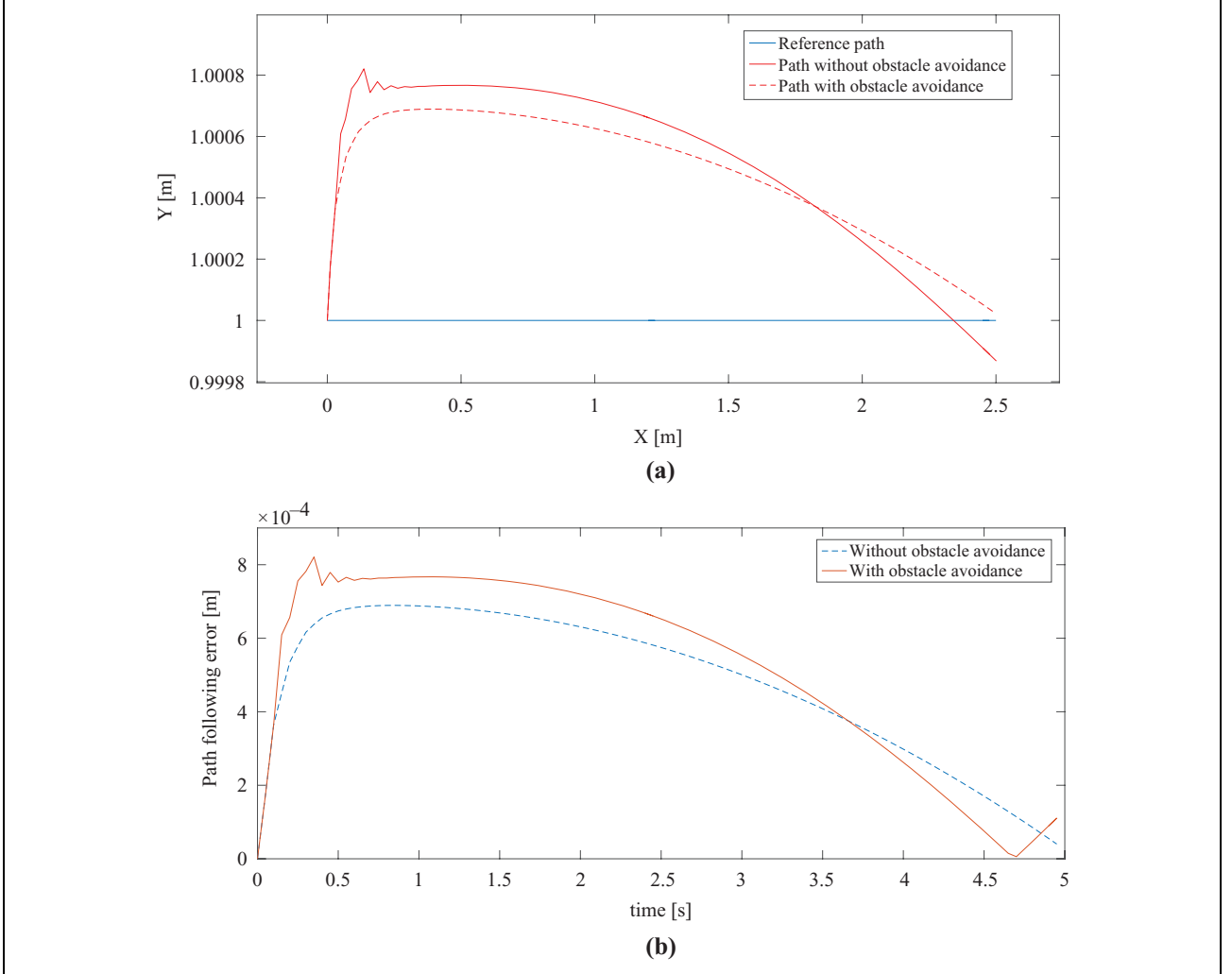


Figure 8. End-effector A position along the path (a) and the path following error (b).

$$\vec{w}_{i_k} = \min \|\vec{a}(\vec{q}_{i_k}) - \vec{b}\|^2 \quad (33)$$

where the subscript $i = a, b$ indicates the two manipulators A and B . Suppose now that A is the arm closest to the obstacle (the same applies if B is the closest). The second step consists of maximizing the distance by calculating the gradient $\nabla |\vec{w}_a|$ in the joint space. Therefore, it is possible to obtain the 6-dimensional joint velocity vector $\dot{\vec{q}}_1^+(m^* = 1)$ relative to manipulator A

$$\dot{\vec{q}}_{1k}^+ = k_a [\nabla_q |\vec{w}_{a_k}| \mathbf{0}]^T \quad (34)$$

where the obstacle avoidance gain k_a is defined as

$$\begin{cases} k_a = (1 - (|\vec{w}_a|/d_T))k_a^* & |\vec{w}_a| < d_T \\ k_a = 0 & |\vec{w}_a| \geq d_T \end{cases} \quad (35)$$

with k^* being the nominal gain value and d_T being the threshold distance where the the obstacle avoidance task is active. Finally, it is possible to calculate the final joint

velocity vector of the equivalent manipulator by substituting equations (31) and (34) into equation (29)

$$\begin{aligned} \dot{\vec{q}}_{ab_k} = & \mathbf{J}_R^\dagger (\dot{\vec{x}}_{Rd_k} + \mathbf{K}_R \vec{e}_{R_k}) + P_{R_k} ([\mathbf{J}_A(\vec{q}_{a_k}) \mathbf{0}]^\dagger (\dot{\vec{x}}_{d_k} + \mathbf{K} \vec{e}_k) \\ & + \mathbf{P}_{ab_k} k_a [\nabla_q |\vec{w}_{a_k}| \mathbf{0}]^T) \end{aligned} \quad (36)$$

Equation (36) presents a hierarchical task structure, which is summarized in the block digram shown in Figure 4.

Thus, the final degree of redundancy of the system leads to

$$r_{ab} = n_{ab} - m_{ab} - m_a - m^* = 0 \quad (37)$$

Results

In this section we simulate the movement of a dish, which has a ray $r_1 = 0.2$ m and its center placed in $p_c = [0.2 \text{ m}, 1 \text{ m}]$, along a straight line defined between the points $\vec{p}_1 = [0 \text{ m}, 1 \text{ m}]^T$ and $\vec{p}_2 = [0.25 \text{ m}, 1 \text{ m}]^T$. Moreover, a

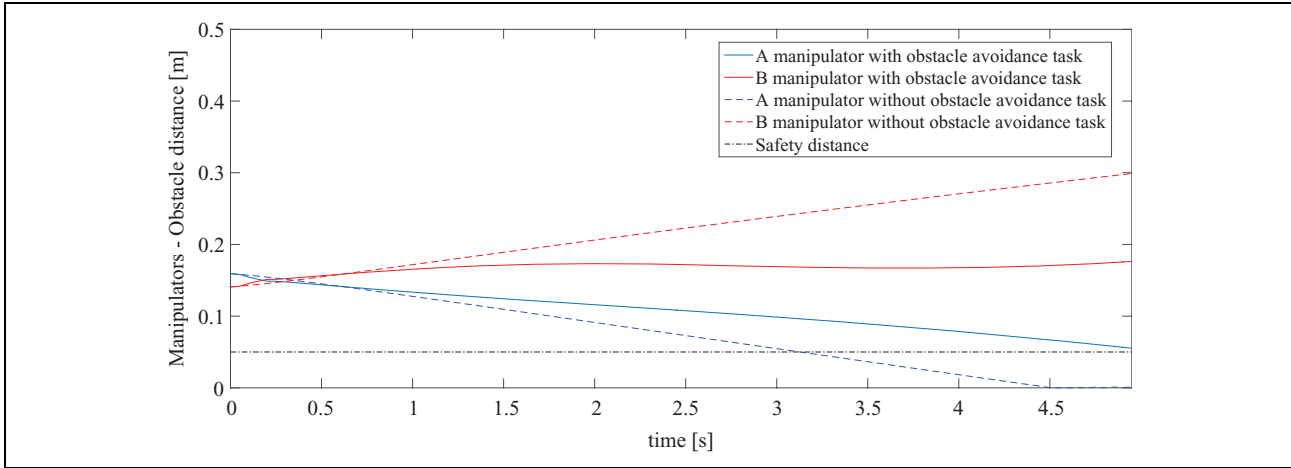


Figure 9. Minimum distance between manipulators and obstacle.

bottle centered in $\vec{c} = [0.21 \text{ m}, 0.74 \text{ m}]^T$ and of ray $r_2 = 0.05 \text{ m}$ is placed in the dual manipulation workspace,^{39,40} so that an obstacle avoidance algorithm is implemented and assigned to the manipulator nearest the obstacle. This simulation is composed by three different priority level tasks.

1. Maintaining a relative distance between the two end effectors $\|\vec{p}_R\| = 0.4 \text{ m}$.
2. Moving the A manipulator along the defined path.
3. Avoiding an obstacle (i.e. a bottle).

In order to keep a constant end effector distance, the relative Cartesian velocity vector \vec{x}_{Rd} is imposed in accordance with equation (28), while the desired Cartesian A manipulator velocity is assigned as $\vec{x}_d = [0.05 \text{ m}, 0 \text{ m}]^T$. Finally, the obstacle-avoiding velocity is obtained in the joint velocity space by $k_a \nabla_q |\vec{w}|$, when the distance between the obstacle and manipulators is less than a threshold distance d_T of 0.20 m.

The starting pose of the dual-arm system is shown in Figure 3(b), where the starting A manipulator pose is $\vec{x}_{eb} = [0 \text{ m}, 1 \text{ m}, \pi/2 \text{ rad}]^T$, the starting B manipulator pose is $\vec{x}_{eb} = [0.4 \text{ m}, 1 \text{ m}, \pi/2 \text{ rad}]^T$, while the distance between the manipulator bases d_{AB} is equal to 0.50 m in accordance with the width of the wheelchair. The black and red circles represent the sections of the dish and bottle in the XY -plane respectively, while the blue line represents the minimal distance between the obstacle point \vec{b} and the nearest manipulator point $\vec{a}(\vec{q}_{ik})$ at time k . Figures 5(a) and (b) show the final pose of the dual-arm system, without and with the obstacle avoidance task, while the purple line indicates the path tracked by manipulator A . It is worth noting that since the end effector's orientation values are not specified by the task, they can arbitrary change during the motion when the obstacle avoidance task is not applied. However, the relative end effector's orientation value is kept constant by the primary task.

The variations of joint angles without and with obstacle avoidance task are shown respectively in Figures 6(a) and (b).

The performances of the first task are shown in Figures 7(a) and (b), where the relative end effector's pose and orientation errors are reported. The results obtained indicate that the primary task is independent from the other secondary tasks. The performances of the second task are shown in Figures 8(a) and (b), which indicate respectively the end effector A 's position along the reference path and the relative error. Because no specification on the end effector's orientation is assigned, it can be calculated in order to maximize the distance between obstacle and manipulators in accordance with the obstacle avoidance task. In particular, Figure 9 shows the minimum distance between the two manipulators and obstacle, so that it is possible to note how the obstacle avoidance algorithm permits us to keep the minimum distance value higher than safety distance. Since the primary task imposes the same relative end effector orientation value during the motion, the B manipulator remains near the obstacle when the obstacle avoidance task is executed.

Conclusions and future work

This paper presents the redundancy analysis of two cooperative manipulators considered as a single redundant manipulator through the use of the relative Jacobian matrix. The kinematic redundancy is resolved through the Jacobian null space technique, which permits us to perform several tasks with different execution priority levels by projecting the lower priority tasks into the null space of the relative Jacobian matrix. Since the maximum number of lower priority tasks executable at the same time depends on the dimension of the null space of the relative Jacobian matrix, a hierarchical structure of tasks can be defined, where lower priority tasks can be added as long as the redundancy degree is not zero. The proposed approach permits us to use

a dual-arm manipulation system to perform a primary task by using a control law as simple in structure as those used for an individual manipulator, and accounting for secondary tasks as well (such as obstacle avoidance or avoiding joint limits). In the presented case study, two planar manipulators mounted on a smart wheelchair are considered, where the primary task is to keep a constant relative distance between the two end effectors, while the secondary tasks are to move one manipulator along a defined path while avoiding an obstacle. The obtained results show that the lower priority tasks do not affect the performance of the higher priority task.

Note that, due to the presence of the Jacobian pseudo-inverse, the Jacobian null space technique used in this paper may present a relevant computing effort, and thus a less computationally demanding control scheme, such as the one based on the Jacobian null space projection, should be considered for implementation on a real controller of two anthropomorphic manipulators (most suitable in the AAL scenarios). The authors are currently investigating full cooperation between the dual-arm manipulators and the mobile wheelchair in three dimensions, and are considering the extension of the case study to the three-dimensional case by using the kinematic model of real robotic manipulators (i.e. Jaco), and the cooperation of robotic arms with the smart wheelchair to operate the movement of grasped objects in three-dimensional space.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

- Smith C, Karayiannidis Y, Nalpantidis L, et al. Dual arm manipulation - A survey. *Robot Auton Syst* 2012; 60(10): 1340–1353.
- Goertz R. Fundamentals of general-purpose remote manipulators. *Nucleonics* 1952; 10(11): 36–45.
- Ambrose RO, Aldridge H, et al. Robonaut: Nasa's space humanoid. *IEEE Intell Syst App* 2000; 15(4): 57–63.
- ACTIVE. Active constraints technologies for ill-defined or volatile environments, <http://www.active-fp7.eu> (accessed 31 March 2015).
- Xu J, Grindle GG, Salatin B, et al. Enhanced bimanual manipulation assistance with the personal mobility and manipulation appliance (PerMMA). In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 18–22 October 2010, Taipei (Taiwan), pp.5042–5047.
- Cheng-Shiu C, Wang H and Cooper RA. Autonomous function of wheelchair-mounted robotic manipulators to perform daily activities. In: *IEEE international conference on rehabilitation robotics (ICORR)*, 24–26 January 2013, Seattle-WA (USA), pp.1–6.
- Cunningham A, Keddy-Hector W, Sinha U, et al. Jamster: A mobile dual-arm assistive robot with Jamboxx control. In: *IEEE international conference on automation science and engineering (CASE)*, 18–22 August 2014, Taipei (Taiwan), pp.509–514.
- Benetazzo F, Ferracuti F, Freddi A, et al. AAL technologies for independent life of elderly people. In: *Ambient assisted living*. Switzerland, Springer International Publishing, 2015, pp.329–343.
- Grindle GG, Wang H, Jeannis H, et al. Design and user evaluation of a wheelchair mounted robotic assisted transfer device. *BioMed Research International*, Hindawi Publishing Corporation, 2015, Article ID 198476, p. 9.
- Aghili F. A reconfigurable robot with lockable cylindrical joints. *IEEE T Robot* 2009; 25(4): 785–797.
- Žlajpah L and Nemeč B. Kinematic control algorithms for on-line obstacle avoidance for redundant manipulators. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 30 september–4 october 2002, Lausanne (SWI), pp.1898–1903.
- Hong MB. On the robot singularity: A novel geometric approach. *Int J Adv Robot Syst* 2012; 9: 197.
- Iossifidis I and Schoner G. Dynamical systems approach for the autonomous avoidance of obstacles and joint-limits for an redundant robot arm. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 9–15 October 2006, Beijing (China), pp.580–585.
- O'Brien JF and Wen JT. Redundant actuation for improving kinematic manipulability. In: *IEEE international conference on robotics and automation (ICRA)*, 10–15 May 1999, Detroit-MI (USA), pp.1520–1525.
- Shiller Z and Dubowsky S. On computing the global time optimal motions of robotic manipulators in the presence of obstacles. *IEEE T Robot Autom* 1991; 7(6): 785–797.
- Shiller Z, Sharma S, Stern I, et al. On-line obstacle avoidance at high speeds. *Int J Robot Res* 2013; 32(9–10): 1030–1047.
- Ben-Israel A and Greville T. *Generalized inverses: Theory and applications, volume 15*. New York: Springer-Verlag, 2013.
- Golub GH and Reinsch C. Singular value decomposition and least squares solutions. *Numer Math* 1970; 14(5): 403–420. Berlin (GER): Springer-Verlag, 1970.
- Baillieul J. Kinematic programming alternatives for redundant manipulators. In: *IEEE international conference on robotics and automation (ICRA)*, March 1985, Boston-MA (USA), pp.722–728.
- Chang PH. A closed-form solution for inverse kinematics of robot manipulators with redundancy. *IEEE T Robot Autom* 1987; 3(5): 393–403.
- Sciavicco L and Siciliano B. A solution algorithm to the inverse kinematic problem for redundant manipulators. *IEEE T Robot Autom* 1988; 4: 403–410.
- Dubey RV, Euler JA and Babcock SM. An efficient gradient projection optimization scheme for a seven-degree-of-freedom redundant robot with spherical wrist. In: *IEEE international conference on robotics and automation (ICRA)*, 24–29 April 1988, Philadelphia-PA(USA), pp.28–36.

23. Antonelli G, Arrichiello F and Chiaverini S. The null-space-based behavioral control for autonomous robotic systems. *Intel Serv Robotics* 2008; 1(1): 27–39.
24. Flacco F and De Luca A. Motion control of redundant robots under joint constraints: Saturation in the null space. In: *IEEE international conference on robotics and automation (ICRA)*, 14–18 May 2012, St. Paul – MN (USA), pp.285–292.
25. Dubey RV, Euler JA and Babcock SM. Real-time implementation of an optimization scheme for seven-degree-of-freedom redundant manipulators. *IEEE T Robot Autom* 1991; 7(5): 579–588.
26. De Luca A and Oriolo G. The reduced gradient method for solving redundancy in robot arms. *Robotersysteme*, 1991; 7(2): 117–122.
27. Park J, Choi Y, Chung WK, et al. Multiple tasks kinematics using weighted pseudo-inverse for kinematically redundant manipulators. In: *IEEE international conference on robotics and automation (ICRA)*, 21–26 May 2001, Seoul (Korea), pp.4041–4047.
28. Lewis C. Trajectory generation for two robots cooperating to perform a task. In: *IEEE international conference on robotics and automation (ICRA)*, 22–28 April 1996, Minneapolis-MN (USA), pp.1626–1631.
29. Jamisola RS and Roberts RG. A more compact expression of relative Jacobian based on individual manipulator Jacobians. *Robot Auton Syst* 2015; 63: 158–164.
30. Siciliano B, Sciavicco L, Villani L, et al. *Robotics: Modeling, planning and control*. London: Springer Science and Business Media, 2009.
31. Chiacchio P, Chiaverini S, Sciavicco L, et al. Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy. *Int J Robot Res* 1991; 10(4): 410–425.
32. Luenberger DG and Ye Y. *Linear and nonlinear programming*. Springer Science and Business Media, 2008.
33. Lee J, Chang PH and Jamisola RS. Relative impedance control for dual-arm robots performing asymmetric bimanual tasks. *IEEE T Ind Electron* 2014; 61(7): 3786–3796.
34. Cao B, Dodds GI and Irwin GW. Redundancy resolution and obstacle avoidance for cooperative industrial robots. *J Robotic Syst* 1999; 16(7): 405–417.
35. Jaco: commercial solution provided by Kinovarobotics, <http://www.kinovarobotics.com/assistive-robotics/products/manipulation/> (accessed 2016).
36. Cavanini L, Benetazzo F, Freddi A, et al. SLAM-based autonomous wheelchair navigation system for AAL scenarios. In: *IEEE/ASME 10th international conference on mechatronic and embedded systems and applications (MESA)*, 10–12 September 2014, Senigallia (Italy), pp.1–5.
37. Cimini G, Ferracuti F, Freddi A, et al. An inertial and QR code landmarks-based navigation system for impaired wheelchair users. In: *Ambient assisted living*. Switzerland: Springer International Publishing, 2014, pp.205–214.
38. Ippoliti G, Jetto L, Longhi S, et al. Mobile Robots: Perception & Navigation. *Comparative Analysis of Mobile Robot Localization Methods Based On Proprioceptive and Exteroceptive Sensors*, Germany/ARS, Austria Literatur Verlag, February, 1, 2007; 11: pp.215–237.
39. Huang MZ and Thebert J. A study of workspace and singularity characteristics for design of 3-DOF planar parallel robots. *Int J Adv Manuf Tech* 2010; 51(5-8): 789–797.
40. Huang MZ. Design of a planar parallel robot for optimal workspace and dexterity. *Int J Adv Robot Syst* 2011; 8(4): 176–183.