International Journal of Humanoid Robotics

WHOLE BODY CONTROL OF A DUAL-ARM MOBILE ROBOT USING A VIRTUAL KINEMATIC CHAIN

YUQUAN WANG[1], CHRISTIAN SMITH[1], YIANNIS KARAYIANNIDIS[1,2]
AND PETTER ÖGREN[1]

[1] *Computer Vision and Active Perception Lab, Centre for Autonomous Systems, School of Computer Science and Communication, Royal Institute of Technology (KTH)*
SE-100 44 Stockholm, Sweden,
{yuquan|ccs|yiankar|petter}@kth.se

[2] *Automatic Control, Automation and Mechatronics, Department of Signal and Systems,*
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
yiannis@chalmers.se

Received 12 May 2015
Revised 13 Oct 2015
Accepted 19 Oct 2015

Dual-arm manipulators have more advanced manipulation abilities compared to single-arm manipulators and manipulators mounted on a mobile base have additional mobility and a larger workspace. Combining these advantages, mobile dual-arm robots are expected to perform a variety of tasks in the future. Kinematically, the configuration of two arms that branches from the mobile base results in a serial-to-parallel kinematic structure. In order to respond to external disturbances, this serial-to-parallel kinematic structure makes inverse kinematic computations non-trivial, as the motion of the base has to take the needs of both arms into account. Instead of using the dual-arm kinematics directly, we propose to use a Virtual Kinematic Chain (VKC) to specify the common motion of the two arms. We formulate a constraint based programming solution which consists of two parts. In the first part, we use an extended serial kinematic chain including the mobile base and the VKC to formulate constraints that realize the desired orientation and translation expressed in the world frame. In the second part, we use the resolved VKC motion to constrain the common motion of the two arms. In order to explore the redundancy of the two arms in an optimization framework, we also provide a VKC-oriented manipulability measure as well as its closed-form gradient. We verify the proposed approach with simulations and experiments that are performed on a PR2 robot, which has two 7 Degrees of Freedom (DoF) arms and a 3 DoF mobile base.

Keywords: mobile manipulation; dual-arm robot; virtual kinematic chain.

1. Introduction

Compared to single arm robots, dual-arm manipulators have potential advantages in terms of higher payloads, concurrent task execution, and more advanced manipulation of a single object.[1] To increase the workspace of the dual-arm manipulators, they are often connected to a mobile base, which results in a system with a high level of kinematic redundancy.

In this paper, we use the PR2 robot as an example to study the whole-body control of such a dual-arm mobile robots. If dual-arm mobile robot needs to respond to external disturbances that are hard to model and predict accurately, e.g. when the dual-arm mobile robot co-manipulates a table with a human, we need to use a reactive robot motion control rather than a motion planning approach.[2]

Unlike the straightforward inverse kinematics solution for a single serial chain, the mobile dual arm co-manipulation behavior is not clearly partitioned into one problem for the left arm, one for the right arm and one for the mobile base. Instead, we approach the coordinated nature of the task by formulating an optimization problem under a set of linear constraints and integrate them with the constraint based programming method.

Constraint based programming allows a wide range of sub-tasks to be formulated as inequality or equality constraints.[3,4] The serial-to-parallel structure of the dual-arm mobile manipulator presents some problems in formulating these constraints. A schematic illustration of the robot kinematic structure can be found in Fig. 1. Note that as the mobile base is common for the two kinematic chains originating in each arm, the inverse kinematics solution generated for the chain consisting of the left arm and the base is different from the inverse kinematics solution generated for the right arm and the base (see Sec. 5 for a mathematical description).

We propose to use a virtual kinematic chain (VKC) for specifying a set of constraints that defines the common motion of the two arms. The proposed VKC-based method solves the coordination control with the following contributions: **(1):** The VKC separates the whole-body constraint-specification and the dual-arm constraint-specification. **(2):** Using the VKC, we are able to apply well-developed serial chain control laws/constraints to control a robot with a branching kinematic structure. **(3):** In order to utilize the redundancy of the two arms, we provide a VKC-oriented velocity manipulability measure as well as its closed-form gradient.

2. Related work

Research on cooperative manipulators has received lots of attention since the 1970s. The *operational space formulation*[5] provides dynamic modelling using the end-effector Cartesian space coordinates. With this formulation, the *augmented object model*[6] describes modelling of the dynamics of multiple fixed-base serial chain manipulators. Then it is extended to serial-to-parallel structure,[7] where the coupling between parallel structures is described with the cross terms of the dynamic model. Different dynamics and kinematics modelling methods are found in a recent book chapter[8] on cooperative manipulators. However, in certain cases it is difficult to create a dynamic model of reasonable accuracy, e.g., when a human is part of the control loop; or it may not be possible to control the motors directly, for instance, we are only allowed to control the joint velocities of most of the commercially available robots. Then, a kinematics approach can be used.

We can separate the kinematic constrains specification of the two arms and the mobile base,[9] whereas if we want to increase the manipulability[10] we need to pose the constraint simultaneously on the mobile base and the manipulator. For the inverse kinematics calculation of a dual-arm mobile manipulator, it is not clear which arm the mobile base should coordinate with. If we use a master-slave formulation,[11] which explicitly connects the mobile base to only one of the arms when formulating the kinematic constraints, the dual-arm manipulability[12] is unstable, see Sec. 6.1. Using the proposed VKC based method, we obtain a more consistent dual-arm manipulability.

Virtual mechanisms are frequently used in the robotics literature. There are both dynamic and kinematic methods. In the *virtual model control*,[13] the legs of a robot are programmed to mimic different virtual mechanical structures in order to control the dynamics along the gravitational force direction. In the *virtual mechanism approach*,[14] a VKC was used to chain serial mechanisms together. Our approach is different from the virtual model control[13] in that the VKC is part of a kinematic model, and different from the virtual mechanism approach[14] in that we use the VKC to specify the common motion shared by the two arms. Since we use a VKC to specify the relative motion rather than to physically interact with the environment, we typically use a VKC with less or equal to 6 Degrees of Freedom (DoF). By choosing a task-dependent VKC, we could explicitly specify the task-dependent motion for the two arms. For example, we could use a virtual revolute joint to specify the orientation and a virtual prismatic joint to specify the translation.[14] We could also specify task-dependent VKC's in the 6 DoF task space in a more systematic way, for example by applying methods such as the *instantaneous Task Specification using Constraints* (iTaSC)[15] but that is outside the scope of this paper.

The velocity manipulability ellipsoid proposed by Yoshikawa[16] measures the transmission ratio from the joint velocity to the end-effector velocity. We can find its extensions in case of a dual arm[12] robot or in case of a closed-loop chain.[17] We can also use the manipulability ellipsoid to optimize the manipulator configuration with respect to a given task.[18] However both the intersection of manipulability ellipsoids[12] and the Rayleigh quotient[17] are not directly differentiable, therefore we formulate a VKC-orientated measure using the generalized eigenvalue[19] and use its closed-form derivative to optimize the manipulator configuration. The closed-form derivative is obtained based on the *product of exponentials* formula[20] and analysis of higher order differential kinematics.[21]

3. Notation and preliminaries

Prior to the mathematical discussion, we define the notations, coordinate frames, kinematic chains, transformations and Jacobian matrices in this section.

3.1. *Notation*

We list most of the notations used throughout the paper, note that we use bold symbols for vectors and a preceding * to denote the desired value of a variable.

- $\boldsymbol{q}$, the joint positions.
- $\boldsymbol{\chi}$, the virtual joint positions of the VKC.
- $R \in SO(3)$, a rotation matrix. We use $R(\boldsymbol{k}, \theta)$ to denote the rotation about an axis $\boldsymbol{k}$ by an angle θ.
- $Q = \{\eta, \boldsymbol{\epsilon}\}$, a unit quaternion, where $\boldsymbol{\epsilon} \in \mathbb{R}^3$ and $\|\boldsymbol{\epsilon}\|_2 = 1$. We use $*$ to denote the quaternion multiplication.
- $\boldsymbol{t} \in \mathbb{R}^3$, a translation vector.
- $g : \mathbb{R}^4 \to \mathbb{R}^4$, a homogeneous transformation, where $g = (\boldsymbol{t}, R) \in SE(3)$. $g_{i,j}$ defines the Euclidean transformation of frame j w.r.t. frame i.
- $Ad_g : \mathbb{R}^6 \to \mathbb{R}^6$, an adjoint transformation. Given $g \in SE(3)$, we have Ad_g and its inverse defined as:

$$Ad_g = \begin{bmatrix} R & S(\boldsymbol{t})R \\ O & R \end{bmatrix}, \quad Ad_g^{-1} = \begin{bmatrix} R^\top & -R^\top S(\boldsymbol{t}) \\ O & R^\top \end{bmatrix},$$

  where $S(\cdot)$ denotes the skew-symmetric matrix.
- $\dot{\boldsymbol{t}} \in \mathbb{R}^3$, a translational velocity.
- $\boldsymbol{\omega} \in \mathbb{R}^3$, a rotational velocity.
- $\boldsymbol{V} = [\dot{\boldsymbol{t}}^\top \ \boldsymbol{\omega}^\top]^\top$, a spatial velocity.
- $J \in \mathbb{R}^{6\times n}$, a Jacobian matrix of a robot arm with n DoFs. We use ${}^{\dot{\boldsymbol{t}}}J, {}^{\boldsymbol{\omega}}J \in \mathbb{R}^{3\times n}$ to denote its translational and rotational part respectively.

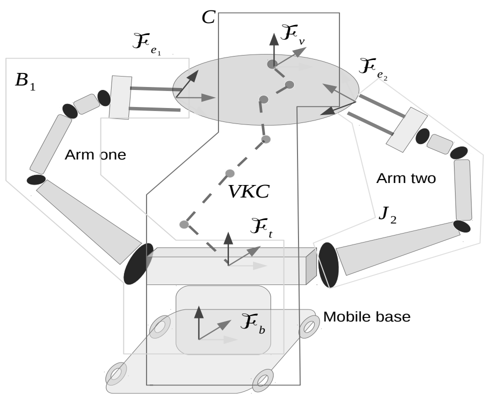


Fig. 1: A schematic drawing of the coordinate frames and the kinematic chains of a mobile dual-arm robot.

3.2. *Coordinate frames and kinematic chains*

In order to facilitate the constraints specification, we plot a schematic drawing of the kinematics of a dual-arm mobile manipulator in Fig. 1. We use a world frame $\mathcal{F}_w$ to define the pose of the mobile base. The torso frame $\mathcal{F}_t$ is defined with respect to a mobile base frame $\mathcal{F}_b$. From the torso frame $\mathcal{F}_t$ there are two end-effector frames $\mathcal{F}_{e_1}$ and $\mathcal{F}e_2$. In order to specify constraints for the common motion of the two arms, we define a virtual end-effector with frame $\mathcal{F}_v$, which is in the middle of the $\mathcal{F}_{e_1}$ and $\mathcal{F}_{e_2}$:

$$\boldsymbol{t}_{bv} = \frac{1}{2}(\boldsymbol{t}_{be_1} + \boldsymbol{t}_{be_2}), \qquad R_{bv} = R_{be_1} R_{\frac{1}{2}} \in SO(3).$$

where $R_{\frac{1}{2}} = R(\boldsymbol{k}, \frac{\theta}{2})$ and $\boldsymbol{k}, \theta$ are angle axis representations of $R_{be_1}^{\top} R_{be_2}$.

We use a superscript to denote the reference frame for a matrix or a vector. For example we denote a velocity in $\mathcal{F}_{e_i}$ as $\boldsymbol{V}^{e_i}$. For velocity, translation, rotation and Jacobian, we use two consequent subscripts as is shown in the following example: the virtual end-effector velocity relative to the base frame expressed in the base frame is denoted as $\boldsymbol{v}_{bv}^{b}$. If no superscript is used, by default it indicates that the reference frame is $\mathcal{F}_b$. Note that there are more than one kinematic chain connecting $\mathcal{F}_t$ and $\mathcal{F}_v$. Apart from the two kinematic chains of the two arms that branch from $\mathcal{F}_t$ and join at $\mathcal{F}_v$, there is also a *virtual kinematic chain* that starts from $\mathcal{F}_t$ to $\mathcal{F}_v$.

3.3. *Base-arm Jacobian and base-VKC Jacobian*

In line with the notations in the book,[20] we use the following to perform spatial-velocity transformation for different points on a rigid object:

$$\boldsymbol{V}_{be_i}^{b} = \boldsymbol{V}_{bt}^{b} + Ad_{g_{bt}} \boldsymbol{V}_{te_i}^{t}, \tag{1}$$

Given an arm-less mobile base we have a Jacobian: $J_{bt}\dot{\boldsymbol{q}}_b = \boldsymbol{V}_{bt}$ and given a fixed-base manipulator we have $J_{te_i}^{t}\dot{\boldsymbol{q}}_i = \boldsymbol{V}_{te_i}^{t}$, where $J_i = J_{te_i}^{t}$ for $i = 1, 2$. We can use the transformation (1) to combine these two parts together as:

$$B_i[\dot{\boldsymbol{q}}_b^{\top} \dot{\boldsymbol{q}}_i^{\top}]^{\top} = \boldsymbol{V}_{be_i}.$$

where $B_i = [J_{bt} \;\; Adg_{bt} J_{te_i}^{t}]$. Using (1) again, we can concatenate the mobile base joints $\boldsymbol{q}_b$ to the VKC joints $\boldsymbol{\chi}$ as:

$$C[\dot{\boldsymbol{q}}_b^{\top} \dot{\boldsymbol{\chi}}^{\top}]^{\top} = \boldsymbol{V}_{bv},$$

where we define a *base-VKC Jacobian*:

$$C = [J_{bt} \;\; Adg_{bt} J_{tv}^{t}].$$

In Fig. 1, we marked the robot components corresponding to B_1 and C respectively.

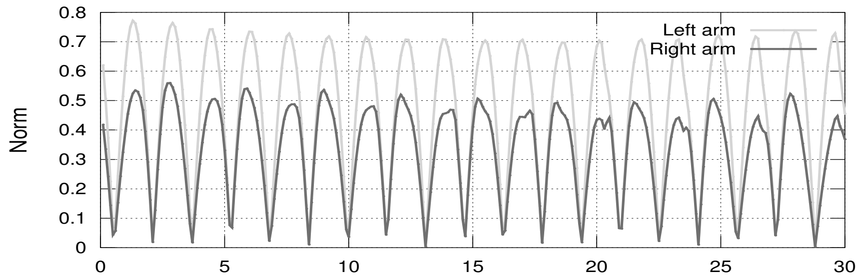


Fig. 2: The difference between $\|J_1\dot{\boldsymbol{q}}_1\|_2$ and $\|J_2\dot{\boldsymbol{q}}_2\|_2$. The data is generated with the master-slave method described in Sec. 6.1.

4. Problem formulation

Based on the base-arm Jacobian defined in the previous section, we point out the inverse kinematics problem when we pose a constraint simultaneously on the two arms and the mobile base. Suppose the virtual end-effector frame needs to realize a desired ${}^*\boldsymbol{V}_{bv}$, by directly formulating the following two constraints:

$$\left.\begin{array}{rl} B_1[\dot{\boldsymbol{q}}_b^\top \dot{\boldsymbol{q}}_1^\top]^\top = & \overbrace{J_{bt}\dot{\boldsymbol{q}}_b}^{\text{mobile base}} + Adg_{bt}J_1\dot{\boldsymbol{q}}_1 \\ B_2[\dot{\boldsymbol{q}}_b^\top \dot{\boldsymbol{q}}_2^\top]^\top = & \underbrace{J_{bt}\dot{\boldsymbol{q}}_b}_{\text{mobile base}} + Adg_{bt}J_2\dot{\boldsymbol{q}}_2 \end{array}\right\} = {}^*\boldsymbol{V}_{bv},$$

two separate kinematic chains are used, which are: **(1)** the mobile base and the left arm **(2)** the mobile base and the right arm. However the two separate constraints relate $\dot{\boldsymbol{q}}_b$ simultaneously to $\dot{\boldsymbol{q}}_1$ and $\dot{\boldsymbol{q}}_2$ such that we end up with a self-conflicting $\dot{\boldsymbol{q}}_b$ as:

$$J_{bt}\dot{\boldsymbol{q}}_b = \begin{cases} \boldsymbol{V}_{bv} - Adg_{bt}J_1\dot{\boldsymbol{q}}_1 \\ \boldsymbol{V}_{bv} - Adg_{bt}J_2\dot{\boldsymbol{q}}_2 \end{cases}, \tag{2}$$

where $J_1\dot{\boldsymbol{q}}_1$ in general is not equal to $J_2\dot{\boldsymbol{q}}_2$ due to the different arm configurations. We illustrate this difference by explicitly plotting $\|J_1\dot{\boldsymbol{q}}_1\|_2$ and $\|J_2\dot{\boldsymbol{q}}_2\|_2$ in Fig. 2. To summarize, the whole-body manipulation problem for a dual-arm mobile robot can be described as:

Problem 1 Control the two arms and the mobile base in a cooperative way (without conflicting constraints) such that the following are fulfilled:

- Track the desired virtual end-effector pose:

$$R^t_{tv} = {}^*R^t_{tv}(t) \text{ and } \boldsymbol{t}^w_{wv} = {}^*\boldsymbol{t}^w_{wv}(t). \tag{3}$$

- Keep the desired relative pose between the two arms:

$$R^{e_1}_{e_1e_2} = {}^*R^{e_1}_{e_1e_2}(t) \text{ and } \boldsymbol{t}^{e_1}_{e_1e_2} = {}^*\boldsymbol{t}^{e_1}_{e_1e_2}(t). \tag{4}$$

5. Proposed solution

We propose a constraint based programming solution to Problem 1. We formulate constraints that are able to solve (3) and (4) in Sec. 5.1. Then in order to explore the redundancy of the two arms, we introduce a VKC-oriented manipulability measure together with its closed-form gradient in Sec. 5.2. We integrate the proposed constraints using a variation of constraint-based programming in Sec. 5.3.

5.1. *Dual-arm mobile manipulator control constraints*

As stated in Problem 1, the constraints specification of the dual-arm mobile manipulator consists of two parts: in Sec. 5.1.1 we introduce the constraints that describe a mobile manipulation task that is defined in the world frame $\mathcal{F}_w$ and then in Sec. 5.1.2, we describe the constraints that keep the time-dependent relative pose between the two arms.

5.1.1. *Whole body control constraint*

In order to fulfill the constraint (3), which is defined in the world frame, we use the extended serial chain and the associated base-VKC Jacobian C to specify constraints simultaneously on the two arms and the mobile base. Basically we minimize the orientation and translation error $\Delta\boldsymbol{\epsilon}^t_{tv}, \Delta\boldsymbol{t}^w_{wv} \in \mathbb{R}^3$ with two equalities in the following form:

$$\begin{aligned}\Delta\dot{\boldsymbol{\epsilon}}^t_{tv} &= -k\Delta\boldsymbol{\epsilon}^t_{tv}\\ \Delta\dot{\boldsymbol{t}}^w_{wv} &= -k\Delta\boldsymbol{t}^w_{wv},\end{aligned}$$

which can be further developed as:

Orientation: We use the unit quaternion $Q^{t-1}_{tv} * {}^*Q^t_{tv}(t)$ to represent the orientation error ${R^t_{tv}}^\top {}^*R^t_{tv}(t)$. As stated in the book,[22] it is sufficient to represent the 3 dimensional orientation error with the vector part $\Delta\boldsymbol{\epsilon}^t_{tv}$ of $Q^{t-1}_{tv} * {}^*Q^t_{tv}(t)$:

$$\Delta\boldsymbol{\epsilon}^t_{tv} = \eta^t_{tv}\,{}^*\boldsymbol{\epsilon}^t_{tv} - {}^*\eta^t_{tv}\boldsymbol{\epsilon}^t_{tv} - S(\boldsymbol{\epsilon}^t_{tv})\,{}^*\boldsymbol{\epsilon}^t_{tv}. \tag{5}$$

For notational compactness we omit the sub-/superscripts in the following discussion. Using the *quaternion propagation*[22]

$$\begin{aligned}\dot{\eta} &= -\frac{1}{2}\boldsymbol{\epsilon}^T\boldsymbol{\omega}\\ \dot{\boldsymbol{\epsilon}} &= \frac{1}{2}(\eta\boldsymbol{I} - S(\boldsymbol{\epsilon}))\boldsymbol{\omega},\end{aligned}$$

we obtain the relationship between the time derivative of $\Delta\boldsymbol{\epsilon}$ and the virtual end-effector velocities:

$$\Delta\dot{\boldsymbol{\epsilon}} = {}^*\boldsymbol{\epsilon}\dot{\eta} - {}^*\eta\dot{\boldsymbol{\epsilon}} + S({}^*\boldsymbol{\epsilon})\dot{\boldsymbol{\epsilon}} = -\frac{1}{2}\left({}^*\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T + ({}^*\eta\boldsymbol{I} - S({}^*\boldsymbol{\epsilon}))(\eta\boldsymbol{I} - S(\boldsymbol{\epsilon}))\right)\boldsymbol{\omega}.$$

Then we can minimize $\Delta\boldsymbol{\epsilon}$ with the following equality:

$$-\frac{1}{2}\left({}^{*}\boldsymbol{\epsilon}\boldsymbol{\epsilon}^{T} + ({}^{*}\eta\boldsymbol{I} - S({}^{*}\boldsymbol{\epsilon}))(\eta\boldsymbol{I} - S(\boldsymbol{\epsilon}))\right){}^{\boldsymbol{\omega}}C[\dot{\boldsymbol{q}}_b^{\top}\dot{\boldsymbol{\chi}}^{\top}]^{\top} = -k\Delta\boldsymbol{\epsilon} \tag{6}$$

where we used the relation $\boldsymbol{\omega}_{bv}^{b} = {}^{\boldsymbol{\omega}}C[\dot{\boldsymbol{q}}_b^{\top}\dot{\boldsymbol{\chi}}^{\top}]^{\top}$.

Translation: The translation error can be defined in a straightforward way using the following difference:

$$\Delta\boldsymbol{t}_{wv}^{w} = \boldsymbol{t}_{wv}^{w} - {}^{*}\boldsymbol{t}_{wv}^{w}(t).$$

Since the base-VKC Jacobian C is defined in the base frame, we transform $\Delta\boldsymbol{t}_{wv}^{w}$ into the base frame:

$$\Delta\boldsymbol{t}_{bv}^{b} = R_{wb}^{w}{}^{\top}\Delta\boldsymbol{t}_{bv}^{b} - R_{wb}^{w}{}^{\top}\boldsymbol{t}_{wb}^{w},$$

and use the following to minimize $\Delta\boldsymbol{t}_{wv}^{w}$:

$${}^{\boldsymbol{t}}C[\dot{\boldsymbol{q}}_b^{\top}\dot{\boldsymbol{\chi}}^{\top}]^{\top} = -k\Delta\boldsymbol{t}_{bv}^{b}. \tag{7}$$

Using the constraints (6-7) we are able to generate the joint velocities $\dot{\boldsymbol{q}}_b$ and $\dot{\boldsymbol{\chi}}$. In order to simultaneously control the mobile base and the two arms, we can directly apply $\dot{\boldsymbol{q}}_b$ to the low-level joint velocity controllers and we use $\dot{\boldsymbol{\chi}}$ to constrain $\dot{\boldsymbol{q}}_1$, $\dot{\boldsymbol{q}}_2$. As we assume $\boldsymbol{V}_{e_iv}^{e_i} = \boldsymbol{0}$, we can obtain the following equation using (1):

$$\boldsymbol{V}_{tv}^{t} = \boldsymbol{V}_{te_i}^{t} + Ad_{g_{e_iv}}\boldsymbol{V}_{e_iv}^{e_i} = \boldsymbol{V}_{te_i}^{t},$$

which leads to:

$$\begin{bmatrix} J_{te_1}^{t} & 0 \\ 0 & J_{te_2}^{t} \end{bmatrix}\begin{bmatrix} \dot{\boldsymbol{q}}_1 \\ \dot{\boldsymbol{q}}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}_{te_1}^{t} \\ \boldsymbol{V}_{te_2}^{t} \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}_{tv}^{t} \\ \boldsymbol{V}_{tv}^{t} \end{bmatrix} = \begin{bmatrix} J_{tv}^{t}\dot{\boldsymbol{\chi}} \\ J_{tv}^{t}\dot{\boldsymbol{\chi}} \end{bmatrix}. \tag{8}$$

5.1.2. *Relative pose between the two arms*

The relative pose between the two arms are controlled with two equalities:

$$\begin{aligned} \frac{\partial\Delta\boldsymbol{\epsilon}_{e_1e_2}^{e_1}}{\partial\boldsymbol{q}_1} + \frac{\partial\Delta\boldsymbol{\epsilon}_{e_1e_2}^{e_1}}{\partial\boldsymbol{q}_2} &= -k\Delta\boldsymbol{\epsilon}_{e_1e_2}^{e_1} \\ \frac{\partial\Delta\boldsymbol{t}_{e_1e_2}^{e_1}}{\partial\boldsymbol{q}_1} + \frac{\partial\Delta\boldsymbol{t}_{e_1e_2}^{e_1}}{\partial\boldsymbol{q}_2} &= -k\Delta\boldsymbol{t}_{e_1e_2}^{e_1}, \end{aligned}$$

which minimize the orientation and translation error $\Delta\boldsymbol{\epsilon}_{e_1e_2}^{e_1}$, $\Delta\boldsymbol{t}_{e_1e_2}^{e_1}$ respectively. We can define the error of the relative orientation $R_{e_1e_2}^{e_1} = R_{te_1}^{t}{}^{\top}R_{te_2}^{b}$ using unit quaternion as:

$$\Delta Q_{e_1e_2}^{e_1} = Q_{te_1}^{t}{}^{-1} * Q_{te_2}^{t} - {}^{*}Q_{te_1}^{t}{}^{-1} * {}^{*}Q_{te_2}^{t},$$

then we take the vector part $\Delta\boldsymbol{\epsilon}^{e_1}_{e_1e_2}$ of $\Delta Q^{e_1}_{e_1e_2}$ and differentiate it w.r.t. to $\boldsymbol{q}_1$,$\boldsymbol{q}_2$ using the quaternion propagation:

$$\begin{aligned} &-\frac{1}{2}\left(\boldsymbol{\epsilon}_2\boldsymbol{\epsilon}_1^\top + (\eta_2\boldsymbol{I} - S(\boldsymbol{\epsilon}_2))(\eta_1\boldsymbol{I} - S(\boldsymbol{\epsilon}_1))\right){}^{\boldsymbol{\omega}}J_1\dot{\boldsymbol{q}}_1 \\ &+\frac{1}{2}\left(\boldsymbol{\epsilon}_1\boldsymbol{\epsilon}_2^\top + (\eta_1\boldsymbol{I} - S(\boldsymbol{\epsilon}_1))(\eta_2\boldsymbol{I} - S(\boldsymbol{\epsilon}_2))\right){}^{\boldsymbol{\omega}}J_2\dot{\boldsymbol{q}}_2 \\ &= -k\Delta\boldsymbol{\epsilon} \end{aligned} \tag{9}$$

where for notational compactness we omit the sub-/superscripts again. Then we define the relative translation error:

$$\Delta\boldsymbol{t}^{e_1}_{e_1e_2} = \boldsymbol{t}^{e_1}_{e_1e_2} - {}^*\boldsymbol{t}^{e_1}_{e_1e_2},$$

where $\boldsymbol{t}^{e_1}_{e_1e_2} = R^{t\,\top}_{te_1}(\boldsymbol{t}^t_{te_1} - \boldsymbol{t}^t_{te_2})$. Differentiating $\Delta\boldsymbol{t}^{e_1}_{e_1e_2}$ w.r.t $\boldsymbol{q}_1$,$\boldsymbol{q}_2$, we obtain the following equality:

$$\left(R^{t\,\top}_{te_1} + R^{t\,\top}_{te_1}S(\boldsymbol{t}^t_{te_1} - \boldsymbol{t}^t_{te_2})\right){}^{\boldsymbol{\omega}}J_1\dot{\boldsymbol{q}}_1 - R^{t\,\top}_{te_1}{}^{\boldsymbol{\omega}}J_2\dot{\boldsymbol{q}}_2 = -k\Delta\boldsymbol{t}^{e_1}_{e_1e_2}. \tag{10}$$

The equalities (9-10) are expansions of the two equalities we aforementioned in the beginning of this section that are able to constrain the relative pose between the two arms. Depending on the task specifications, the relative pose error $\Delta\boldsymbol{\epsilon}^{e_1}_{e_1e_2}$, $\Delta\boldsymbol{t}^{e_1}_{e_1e_2}$ on the right side of (9-10) can be calculated for a static or a dynamic task, e.g. they can be calculated in case of a dual-arm pan cleaning task[4] and they can also be used to define the impedance relation between the two arms.[2]

5.2. *VKC-oriented manipulability measure*

In light of the velocity manipulability ellipsoid proposed by Yoshikawa[16] as well as its extensions to the dual-arm case[12] and the closed-loop case[17], we propose a velocity transmission ratio from the joint velocity $\dot{\boldsymbol{q}}_i \in \mathbb{R}^n$ to the virtual end-effector velocity $\boldsymbol{V}^t_{tv} \in \mathbb{R}^6$. In order to use it in the optimization procedure with a better accuracy we also provide its closed-form derivative. Suppose we have a unit sphere in the joint space $\mathbb{R}^n$, which is: $\|\boldsymbol{q}\|^2 = q_1^2 + q_2^2 + \ldots + q_n^2 \leq 1$. Using a manipulator Jacobian J , we map the unit sphere into a end-effector velocity ellipsoid in $\mathbb{R}^6$:

$$\dot{\boldsymbol{q}}^\top\dot{\boldsymbol{q}} = \boldsymbol{V}^\top(JJ^\top)^{-1}\boldsymbol{V} = 1. \tag{11}$$

The determinant

$$\gamma = -\det(JJ^\top)^{\frac{1}{2}}, \tag{12}$$

is proportional to the volume of the ellipsoid (11). We could minimize (12) w.r.t. the robot configuration to increase the velocity transmission ratio.

As we specify the common dual-arm motion according to the constraint (8), the end-effector velocity of each arm $\boldsymbol{V}^t_{te_i}$ is defined by the virtual end-effector velocity $\boldsymbol{V}^t_{tv}$. Therefore instead of (11), we use the following manipulability measure in order to count in the constraint(8):

Definition[VKC-oriented manipulability measure] For the ith arm, which is constrained by the virtual end-effector velocity (8), we define the following velocity transmission ratio from $\dot{\boldsymbol{q}}_i$ to the end-effector velocity $\boldsymbol{V}^t_{te_i}$:

$$\gamma_{v_i} = -\det\left((J^t_{tv}J^t_{tv})^{-1}(J_iJ_i^\top)\right) \tag{13}$$

where γ_{v_i} is proportional to the volume of the velocity ellipsoid defined by:

$$\boldsymbol{V}^{t\ \top}_{tv}\left((J^t_{tv}J^t_{tv})^{-1}(J_iJ_i^\top)\right)^{-1}\boldsymbol{V}^t_{tv}.$$

The motivation of the above definition is as follows. According to the constraint (8), $\boldsymbol{V}^t_{te_i}$ needs to simultaneously satisfy:

$$J_i\begin{bmatrix}\dot{\boldsymbol{q}}_1\\ \dot{\boldsymbol{q}}_2\end{bmatrix} = \boldsymbol{V}^t_{te_i} \text{ and } \boldsymbol{V}^t_{te_i} = \boldsymbol{V}^t_{tv} = J^t_{tv}\dot{\boldsymbol{\chi}}.$$

The above two equations define two velocity ellipsoids for the VKC and the ith arm separately:

$$\begin{cases} \dot{\boldsymbol{q}}_i^\top\dot{\boldsymbol{q}}_i = \boldsymbol{V}^{t\ \top}_{tv}(J_iJ_i^\top)^{-1}\boldsymbol{V}^t_{tv} & = 1\\ \dot{\boldsymbol{\chi}}^\top\dot{\boldsymbol{\chi}} = \boldsymbol{V}^{t\ \top}_{tv}(J^t_{tv}J^{t\top}_{tv})^{-1}\boldsymbol{V}^t_{tv} & = 1\end{cases}$$

We define the intersection of these two ellipsoids using the product of the *generalized eigenvalues*, see Remark 1, of the matrix pair $(J_iJ_i^\top, J^t_{tv}J^{t\top}_{tv})$. The volume of the intersection is proportional[16] to γ_{v_i} defined in (13). We choose to minimize its derivative to increase the velocity transmission ratio. The closed-form derivative of (13) is given as follows:

$$\frac{\partial\gamma_{v_i}}{\partial q_j} = -\det(\mu)\mathrm{Tr}[\mu^{-1}\frac{\partial\mu}{\partial q_j}], \tag{14}$$

where $\mu_i = (J_{tv}J_{tv}^\top)^{-1}(J_iJ_i^\top)^\top$ and

$$\frac{\partial\mu_i}{\partial q_j} = (J_{tv}J_{tv}^\top)^{-1}\left(\frac{\partial J_i}{\partial q_j}J_i^\top + J_i\frac{\partial J_i}{\partial q_j}^\top\right).$$

Note that the derivative of the spatial velocity Jacobian could be calculated by differentiating the *product of exponentials* formula.[21]

Remark 1. (Generalized eigenvalue decomposition) For a matrix pair (A, B), where $A, B \in \mathbb{R}^{n\times n}$, all the scalar λ and non-zero vector $\boldsymbol{u}$ that satisfy the following equation:

$$A\boldsymbol{u} = \lambda B\boldsymbol{u}.$$

are the *generalized eigenvalue* λ and the *generalized eigenvector* for (A, B). If B is non-singular, then the generalized eigenvalue decomposition is solved by standard eigenvalue decomposition:

$$B^{-1}A\boldsymbol{u} = \lambda\boldsymbol{u}.$$

If A and B are real and symmetric, then all the generalized eigenvalues λ are real.[19] Since $(J_i J_i^\top)$ and $J^t_{tv} J^t_{tv}{}^\top$ are real and symmetric, the product of the generalized eigenvalues $-\det\left((J^t_{tv} J^t_{tv})^{-1}(J_i J_i^\top)\right)$ is real-valued. ■

5.3. *Constraint based programming implementation*

We solve two consequent Quadratic Programming problems (QP) to calculate the joint velocities $\dot{\boldsymbol{q}}_b$, $\dot{\boldsymbol{q}}_1$, $\dot{\boldsymbol{q}}_2$ and $\dot{\boldsymbol{\chi}}$. The QP is formulated according to a variation of constraint-based programming[3] Note that in order to fix the feasibility of the constraints (6-10), we use a slack variable in each row and minimize its 2-norm in the objective. In the first QP, which is summarized in Table 1, we solve for $\dot{\boldsymbol{q}}_b$ and $\dot{\boldsymbol{\chi}}$ using constraints (6-7):

$$\min_{\dot{\boldsymbol{q}}_b, \dot{\boldsymbol{\chi}}, \boldsymbol{\nu}_1, \boldsymbol{\nu}_2} \quad \dot{\boldsymbol{q}}_b^\top Q_b \dot{\boldsymbol{q}}_b + \dot{\boldsymbol{\chi}}^\top Q_{\boldsymbol{\chi}} \dot{\boldsymbol{\chi}} + \boldsymbol{\nu}_1^\top Q_1 \boldsymbol{\nu}_1 + \boldsymbol{\nu}_2^\top Q_2 \boldsymbol{\nu}_2 \tag{15}$$

$$\text{s.t.} -\tfrac{1}{2}\left({}^*\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T + ({}^*\eta \boldsymbol{I} - S({}^*\boldsymbol{\epsilon}))(\eta \boldsymbol{I} - S(\boldsymbol{\epsilon}))\right) {}^{\boldsymbol{\omega}}C[\dot{\boldsymbol{q}}_b^\top \dot{\boldsymbol{\chi}}^\top]^\top + \boldsymbol{\nu}_1 = -k\Delta\boldsymbol{\epsilon} \tag{16}$$

$${}^{\boldsymbol{t}}C[\dot{\boldsymbol{q}}_b^\top \dot{\boldsymbol{\chi}}^\top]^\top + \boldsymbol{\nu}_2 = -k\Delta \boldsymbol{t}^b_{bv}. \tag{17}$$

where $\dot{\boldsymbol{q}} \in [\dot{\boldsymbol{q}}_{min} \dot{\boldsymbol{q}}_{max}]$ and $\dot{\boldsymbol{\chi}} \in [\dot{\boldsymbol{\chi}}_{min} \dot{\boldsymbol{\chi}}_{max}]$, the slack variables $\boldsymbol{\nu}_1, \boldsymbol{\nu}_2 \in \mathbb{R}^3$ are used to fix the feasibility of (16-17), $Q_b, Q_{\boldsymbol{\chi}}, Q_1, Q_2$ denote the weights.

If the mobile base had 3 DoF, in principle we only need to choose the other 3 DoF (one translational and two rotational) for the VKC in order to specify a 6 DoF virtual end-effector motion. In order to provide a proof-of-concept verification we choose a 6 DoF VKC to fully use the two arms. If we do not have any preference of the orientation or translation along a certain direction, we can use a generic 6 DoF VKC with three prismatic joints followed by a three rotational spherical joints. This specific choice easily defines the VKC workspace as well as its forward and inverse kinematics.

Using the solved $\dot{\boldsymbol{\chi}}$ from the first QP we can calculate $\dot{\boldsymbol{q}}_1$, $\dot{\boldsymbol{q}}_2$ by formulating the second QP which is summarized in Table 2. The overall procedure is listed in Algorithm 1. Note that we can extended the two QPs listed in Table 1,2 with additional constraints. For instance if we need to add an obstacle avoidance constraint for the mobile base, we could add it to the first QP in Table 1.

Table 1: Solve for $\dot{\boldsymbol{q}}_b$ and $\dot{\boldsymbol{\chi}}$ with the extended serial chain

Objective:	**Min.**	**Max.**	**Eq.**
Joint velocities: $\Vert\dot{\boldsymbol{q}}_b\Vert_2 + \Vert\dot{\boldsymbol{\chi}}\Vert_2$	✓		
2-norm of the slack variables	✓		
Constraint:	**Equality**	**Inequality**	**Eq.**
Torso frame orientation constraint	✓		(6)
World frame translation constraint		✓	(7)
Joint limit constraints		✓	

Algorithm 1: Dual-arm mobile robot control using a virtual kinematic chain.

Goal: Calculate: $\dot{\boldsymbol{q}}_b$, $\dot{\boldsymbol{q}}_1$, $\dot{\boldsymbol{q}}_2$ and $\dot{\boldsymbol{\chi}}$.

(i) Formulate the first QP defined in Table 1.
(ii) Solve the first QP for $\dot{\boldsymbol{q}}_b$ and $\dot{\boldsymbol{\chi}}$.
(iii) Formulate the second QP defined in Table 2 using $\dot{\boldsymbol{\chi}}$.
(iv) Solve the second QP for $\dot{\boldsymbol{q}}_1$ and $\dot{\boldsymbol{q}}_2$.
(v) Apply $\dot{\boldsymbol{q}}_1$, $\dot{\boldsymbol{q}}_2$ and $\dot{\boldsymbol{q}}_b$ to the low level joint velocity controllers.
(vi) Update the VKC with $\dot{\boldsymbol{\chi}}$.

Table 2: Solve for $\dot{\boldsymbol{q}}_{1,2}$ with the solution of $\dot{\boldsymbol{\chi}}$

Objective:	**Min.**	**Max.**	**Eq.**
Configuration measure		✓	(13)
Joint velocities: $\|\dot{\boldsymbol{q}}_{1,2}\|_2$	✓		
2-norm of the slack variables	✓		
Constraint:	**Equality**	**Inequality**	**Eq.**
VKC constraint	✓		(8)
Dual-arm relative orientation constraint	✓		(9)
Dual-arm relative translation constraint	✓		(10)
Joint limit constraints		✓	

Table 3: Master/slave solution to Problem 1

Objective:	**Min.**	**Max.**	**Eq.**
Configuration measure		✓	(12)
Joint velocities: $\|\dot{\boldsymbol{q}}_{s,1,2}\|_2$	✓		
2-norm of the slack variables	✓		
Constraint:	**Equality**	**Inequality**	**Eq.**
Master chain	✓		(18)
Dual-arm relative orientation constraint	✓		(9)
Dual-arm relative translation constraint	✓		(10)
Joint limit constraints		✓	

6. Simulation and experiment evaluation

Three sets of simulations and one experiment are presented to validate the proposed solution which is summarized in Algorithm 1. The three simulations step by step verify that Algorithm 1 is a valid answer to Problem 1. Then we apply Algorithm 1 to a human robot co-manipulation experiment that is carried out with a PR2 robot.

In the first simulation, we compare the proposed method with a master-slave solution. In the second simulation, we show that the two arms obtain better measures γ_{v_1}, γ_{v_2} using the closed-form derivatives. In the last simulation, we specify an example of Problem 1 and then solve it with Algorithm 1.

The simulation is based on the *ROS* Groovy PR2 robot simulator whereas the experiment is performed on a real PR2 robot. In the simulations and experiment we use two 7 DoF redundant arms and a 3 DoF mobile base. The QPs are solved with the software *Gurobi* 6.0[23] at 200 Hz, whereas the underlying joint velocities controllers run at 1000 Hz.

6.1. *Comparison*

Instead of explicitly specifying the common motion of the two arms using the VKC constraints (6-8), we can have a master-slave solution to Problem 1, which also avoids the contradicting use of $J_{bt}\dot{\boldsymbol{q}}_b$ shown in (2). Basically we link only one of the

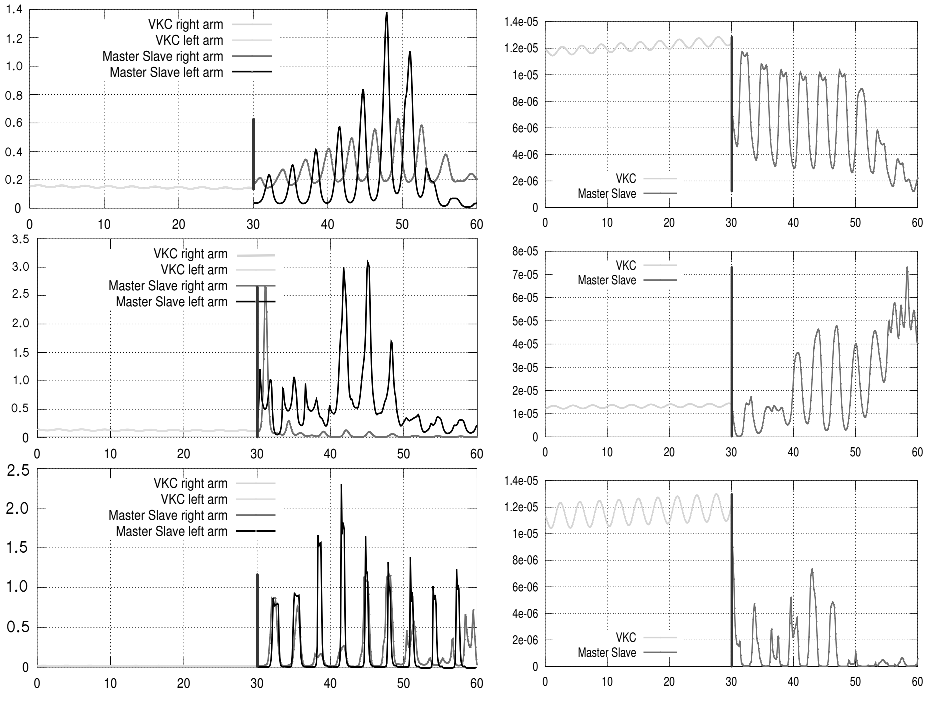


Fig. 3: Time [s] Time [s]
In the left column, we plot the velocity manipulability measures (12) for the two arms and in the right column we plot the dual-arm manipulability measure.[12] The separation line in the middle of each figure indicates that we switch from the proposed method to the master-slave method. We can tell that due to a bigger overlap of the measure (12) between the two arms, the proposed method has a more consistent dual-arm manipulability.

two arms to the mobile base with:

$$B_1[\dot{\boldsymbol{q}}_b^\top \dot{\boldsymbol{q}}_1^\top]^\top = {}^*\boldsymbol{V}_{bv}, \tag{18}$$

and let the other arm be a slave by applying the closed loop constraint (9-10). We summarize this method in Table 3. This method has a biased use of $\boldsymbol{q}_b$ in the sense that only $\dot{\boldsymbol{q}}_1$ is supported by $\dot{\boldsymbol{q}}_b$ in the constraint (18). Furthermore, different constraints are applied on $\dot{\boldsymbol{q}}_1$ and $\dot{\boldsymbol{q}}_2$ such that the configurations of the two arms ($\boldsymbol{q}_1$ and $\boldsymbol{q}_2$) differ from each other over time. This difference results in different manipulability (12) between the two arms and decreases the dual-arm manipulability[12], which is proportional to the volume of the intersection of the two velocity ellipsoids defined by $(J_1 J_1^\top)^{-1}$ and $(J_2 J_2^\top)^{-1}$.

In order to compare the proposed solution with the master-slave solution, we use a 6 dimensional sinusoidal function as the desired ${}^*\boldsymbol{V}_{bv}$ in (18). We also use the 6 dimensional sinus wave to specify the desired velocity of the virtual end-effector instead of the desired pose ${}^*R^t_{tv}(t)$ and ${}^*\boldsymbol{t}^w_{wv}(t)$. Basically we use the following to replace the constraints (6-7) in Algorithm 1:

$$C[\dot{\boldsymbol{q}}_b^\top \dot{\boldsymbol{\chi}}^\top]^\top = {}^*\boldsymbol{V}_{bv}.$$

From the first row to the third row in Fig. 3, we used the wave amplitude: $1N$, $2N$ and $3N$. In the right half of Fig. 3, we can see that the proposed solution has a better performance in the sense that its dual-arm manipulability measure does not vary over time as much as the master-slave method. This result is supported by the left half of Fig. 3, where we can see that using the proposed method the manipulability measures of the two arms have a bigger overlap.

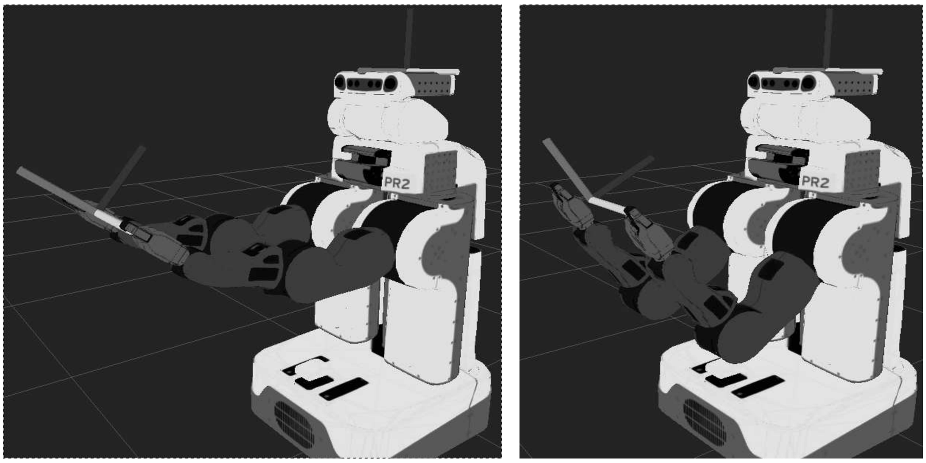


Fig. 4: Left: Initial configuration. **Right:** Optimized configuration.
We illustrate the benefit of the closed-form gradient (14) by maximizing the velocity ellipsoid volume γ_{v_1} and γ_{v_2} while keeping a constant relative pose $R^{e_1}_{e_1 e_2}$ and $\boldsymbol{t}^{e_1}_{e_1 e_2}$.

6.2. *Configuration optimization*

The closed-form gradient (14) provides an analytical form of the gradient used to improve the VKC-oriented manipulability measure γ_{v_i}. We demonstrate it by fixing the relative pose between the two arms:

$$\boldsymbol{t}^{e_1}_{e_1 e_2} = \text{constant}, \quad R^{e_1}_{e_1 e_2} = \text{constant}$$

and keep on minimizing $\frac{\partial \gamma_{v_i}}{\partial \boldsymbol{q}_1}$ and $\frac{\partial \gamma_{v_i}}{\partial \boldsymbol{q}_2}$ in the objective. On the left side of Fig. 4 we start with an initial dual-arm configuration with $\gamma_{v_1} = -0.0220276$ $\gamma_{v_2} = -0.0220434$. On the right side we plot the optimized configuration with $\gamma_{v_1} = -0.0820283$ $\gamma_{v_2} = -0.0820668$.

6.3. *World frame Lissajous curve tracing task*

In order to verify the whole-body control of the two arms and the mobile base, we define a curve tracing task, see Fig 5, that covers an area of $(0.4 \times 1.8)\ m^2$, which is way larger than the workspace of two arms. We plot the tracing error and the relative dual-arm pose error in Fig. 6 and Fig. 7 respectively. We also refer the interested readers to a video in the link below[a].

We use the Lissajous curve to define the desired world frame translation:

$$\boldsymbol{t}^w_{wv}(t) = \begin{cases} x^w_{wv}(t) = a\cos(w_x t - \delta_x) \\ y^w_{wv}(t) \ = b\cos(w_y t - \delta_y) \\ z^w_{wv}(t) \ \ = 0 \end{cases},$$

where the parameters are selected as: $a = 1$, $b = 0.2$, $w_x = 0.018$, $w_y = 0.012$, $\delta_x = 0.7853$ and $\delta_y = 0$. We plot the desired ${}^*\boldsymbol{t}^w_{wv}$ and the realized $\boldsymbol{t}^w_{wv}$ in Fig. 5.

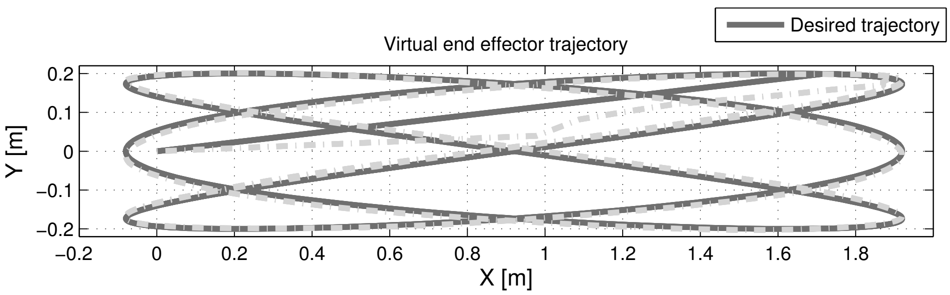


Fig. 5: The desired Lissajous curve ${}^*\boldsymbol{t}^w_{wv}$ and the achieved virtual end-effector trajectory $\boldsymbol{t}^w_{wv}$. Both are plotted in the world frame $\mathcal{F}_w$.

The error $\|\Delta \boldsymbol{t}^w_{wv}\|_2$ is plotted in the first row of Fig. 6. We can see that $\|\Delta \boldsymbol{t}^w_{wv}\|_2$ converges from the initial offset, which is 0.8 m, down to below 2 cm. We also

[a]The PR2 robot performs the Lissajous curve tracing task: `http://youtu.be/HO_amCdft-A`.

defines an orientation requirement:

$$^*Q = \{0.2\sin(0.1t), \boldsymbol{x}\},$$

where $\boldsymbol{x}$ denotes the x axis of $\mathcal{F}_v$ and we transform $^*Q$ into $\mathcal{F}_t$ as

$$^*Q^t_{tv} = Q^t_{tv} * {}^*Q.$$

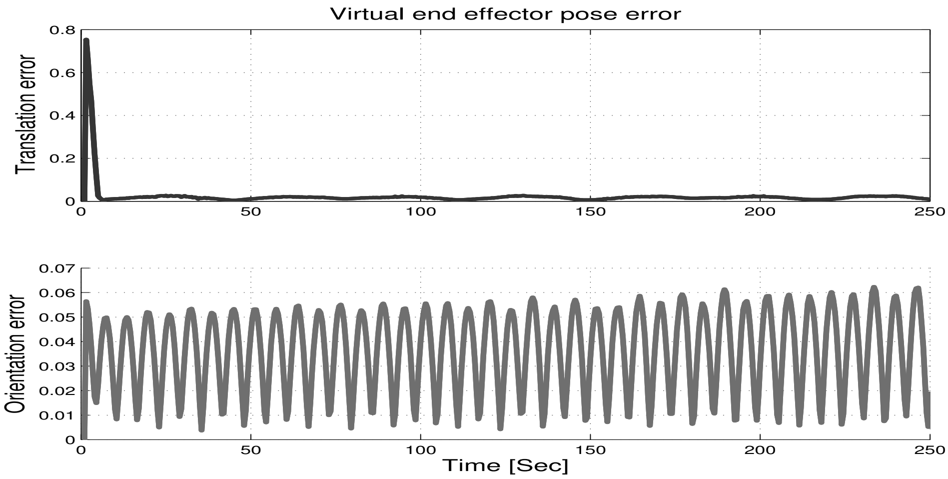


Fig. 6: In the first row we plot the world frame translation error $\|\Delta \boldsymbol{t}^w_{wv}\|_2$ and the mean is 0.0167 m after the first 10 s, then in the second row we plot the base frame orientation error $\|\Delta \boldsymbol{\epsilon}^t_{tv}\|_2$ and the mean is 0.0362.

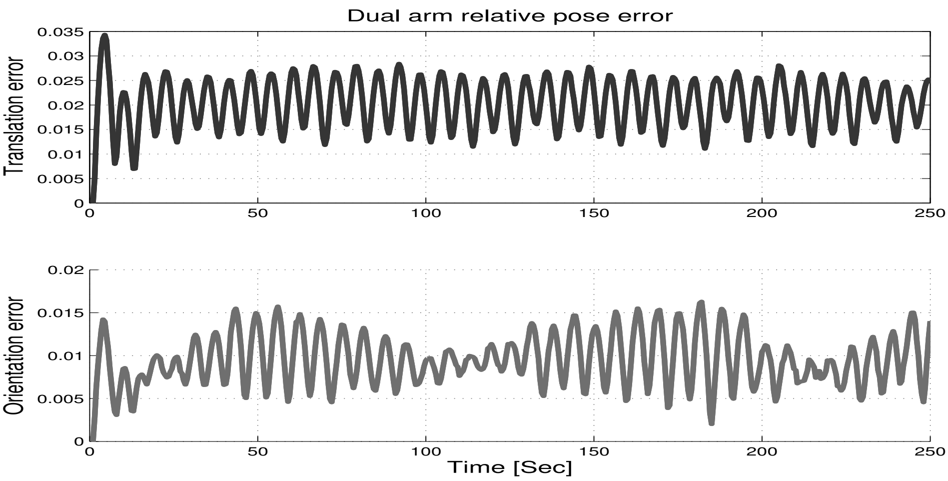


Fig. 7: We fix relative translation and orientation between the dual-arm. In the first row we plot the translation error $\|\Delta \boldsymbol{t}^{e_1}_{e_1e_2}(t)\|_2$ and the mean is 0.0204 m. In the second row we plot the orientation error $\|\Delta \boldsymbol{\epsilon}^{e_1}_{e_1e_2}\|_2$ and the mean is 0.0096.

In the second row of Fig. 6, we can see this orientation requirement is fulfilled as the the vector part of $Q^t_{tv} * {}^*Q^t_{tv}$ is close to zero. Note that there are two reasons responsible for the translation error $\|\Delta \boldsymbol{t}^w_{wv}\|_2$ and the orientation error $\|\Delta \boldsymbol{\epsilon}^t_{tv}\|_2$. First we are not using any time feed-forward term in the constraints (6-7). Second we are only using kinematics.

In this simulation, we choose to keep a constant relative dual-arm pose ${}^*R^{e_1}_{e_1 e_2}(t)$ and ${}^*\boldsymbol{t}^{e_1}_{e_1 e_2}(t)$. We can see from Fig. 7, the pose error is close to zero. If we need to reduce the error $\|\Delta \boldsymbol{t}^{e_1}_{e_1 e_2}(t)\|_2$ and $\|\Delta \boldsymbol{\epsilon}^{e_1}_{e_1 e_2}\|_2$, apart from using a slower world frame translation or orientation trajectory, we can also increase the weight of the constraints (9-10) compared to the constraint (8) in the second QP.

6.4. *Human robot co-manipulation*

Based on the accurate dual-arm mobile manipulator performance illustrated from the previous section, we can easily develop applications, for instance a human robot co-manipulation task shown in Fig. 8.

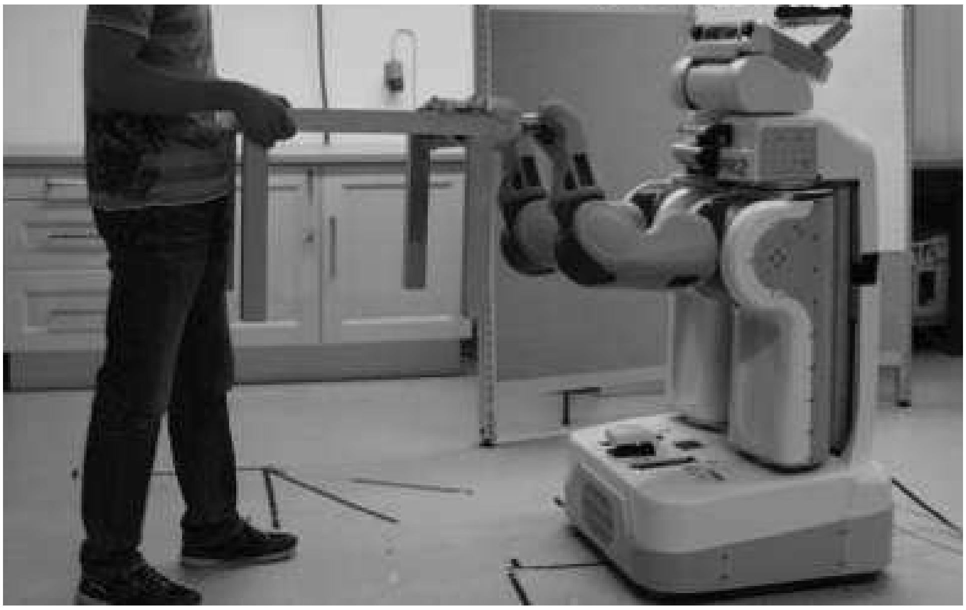


Fig. 8: Example of a human robot co-manipulation task. As the table is rigid, the robot applies fixed grasps on the table and the relative pose between the two arms is fixed.

Suppose we need the robot to follow the behavior of a human operator through an admittance control law that is given as: $\boldsymbol{V}^v_{bv} = D^{-1}\boldsymbol{h}^v$, where $\boldsymbol{h}^v \in \mathbb{R}^{6\times 1}$ denotes the force torque measured at the virtual end-effector frame $\mathcal{F}_v$ and $D \in \mathbb{R}^{6\times 6}$ could be a positive diagonal matrix that specifies the damping coefficient. Then we express this control law in $\mathcal{F}_b$ as:

$$\boldsymbol{V}_{bv} = Ad_{g_{bv}} \boldsymbol{V}^v_{bv} = Ad_{g_{bv}} D^{-1} \boldsymbol{h}^v,$$

and specify the parallel arms motion using the base-VKC Jacobian C:

$$C[\dot{\boldsymbol{q}}_b^\top \dot{\boldsymbol{\chi}}^\top]^\top = \boldsymbol{V}_{bv} = Ad_{g_{bv}} D^{-1} \boldsymbol{h}^v. \quad (19)$$

As a constraint that defines the cooperation between the mobile base and the two arms, we can use (19) to replace constraints (6-7) in Algorithm 1. We can find a video about the experiment in the link below[b], where we can see that the task is successfully executed.

7. Conclusions

We present a constraint-based programming solution to control a dual-arm mobile robot. The use of the VKC separates the whole-body constraint-specification, e.g. the admittance control constraint (19), trajectory tracing constraints (6-7), and the dual-arm constraint-specification, e.g. relative pose constraints (9-10). The manipulability of the dual-arm mobile robot is measured by a VKC-orientated manipulability measure and we integrate it into the constraint based programming implementation with its closed-form gradient.

The proposed solution is validated in a trajectory tracing simulation and a human robot co-manipulation experiment. It achieves a good coordination between the two arms and the mobile base in terms of the dual-arm manipulability.

Acknowledgment

This work is supported by the Swedish Foundation for Strategic Research (SSF) through its Centre for Autonomous Systems, the Swedish Research Council (VR) and the EU FP7 project RoboHow.Cog (FP7-ICT-288533). During the revision process, we recieved constructive comments and detailed proof-reading from Associate Professor Atsuto Maki. The authors gratefully acknowledge the support.

References

1. C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic, "Dual arm manipulationa survey," *Robotics and Autonomous Systems*, vol. 60, no. 10, pp. 1340–1353, 2012.
2. B. Talaei, F. Abdollahi, H. Talebi, and E. O. Karkani, "Whole arm manipulation planning based on feedback velocity fields and sampling-based techniques," *ISA transactions*, vol. 52, no. 5, pp. 684–691, 2013.
3. P. Ogren, C. Smith, Y. Karayiannidis, and D. Kragic, "A multi objective control approach to online dual arm manipulation," in *International IFAC Symposium on Robot Control, SyRoCo, Dubrovnik, Croatia*, 2012.

[b]The PR2 robot assists a human operator to co-manipulate a table: `youtu.be/HO_amCdft-A`.

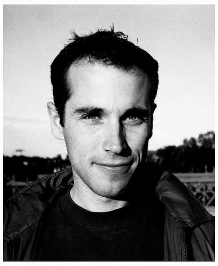
4. Y. Wang, F. Vina, Y. Karayiannidis, C. Smith, and P. Ögren, "Dual arm manipulation using constraint based programming," in *19th IFAC World Congress*, Cape Town, South Africa, August 2014.
5. J. Russakow, O. Khatib, and S. M. Rock, "Extended operational space formulation for serial-to-parallel chain (branching) manipulators," in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 1. IEEE, 1995, pp. 1056–1061.
6. O. Khatib, "Object manipulation in a multi-effector robot system," in *Proceedings of the 4th international symposium on Robotics Research.* MIT Press, 1988, pp. 137–144.
7. K.-S. Chang, R. Holmberg, and O. Khatib, "The augmented object model: Cooperative manipulation and parallel mechanism dynamics," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 1. IEEE, 2000, pp. 470–475.
8. F. Caccavale and M. Uchiyama, "Cooperative manipulators," *Springer handbook of robotics*, pp. 701–718, 2008.
9. S. Erhart, D. Sieber, and S. Hirche, "An impedance-based control architecture for multi-robot cooperative dual-arm mobile manipulation," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on.* IEEE, 2013, pp. 315–322.
10. B. Bayle, J.-Y. Fourquet, and M. Renaud, "Manipulability of wheeled mobile manipulators: Application to motion generation," *The International Journal of Robotics Research*, vol. 22, no. 7-8, pp. 565–581, 2003.
11. J.-H. Bae, S. Arimoto, R. Ozawa, M. Sekimoto, and M. Yoshida, "A unified control scheme for a whole robotic arm-fingers system in grasping and manipulation," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on.* IEEE, 2006, pp. 2131–2136.
12. S. Lee, "Dual redundant arm configuration optimization with task-oriented dual arm manipulability," *Robotics and Automation, IEEE Transactions on*, vol. 5, no. 1, pp. 78–97, 1989.
13. J. Pratt, C.-M. Chew, A. Torres, P. Dilworth, and G. Pratt, "Virtual model control: An intuitive approach for bipedal locomotion," *The International Journal of Robotics Research*, vol. 20, no. 2, pp. 129–143, 2001.
14. N. Likar, B. Nemec, and L. Žlajpah, "Virtual mechanism approach for dual-arm manipulation," *Robotica*, pp. 1–16, 2013.
15. J. De Schutter, T. De Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes, and H. Bruyninckx, "Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty," *The International Journal of Robotics Research*, vol. 26, no. 5, pp. 433–455, 2007.
16. T. Yoshikawa, "Manipulability of robotic mechanisms," *The international journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.
17. A. Bicchi and D. Prattichizzo, "Manipulability of cooperating robots with unactuated joints and closed-chain mechanisms," *Robotics and Automation, IEEE Transactions on*, vol. 16, no. 4, pp. 336–345, 2000.
18. S. L. Chiu, "Task compatibility of manipulator postures," *The International Journal of Robotics Research*, vol. 7, no. 5, pp. 13–21, 1988.
19. X. Zhang, "Matrix analysis and applications," *Tsinghua and Springer Publishing house, Beijing*, pp. 71–100, 2004.
20. R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A mathematical introduction to robotic manipulation.* CRC press, 1994.
21. A. Muller, "Closed form expressions for the sensitivity of kinematic dexterity measures

to posture changing and geometric variations," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 4831–4836.
22. B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2009.
23. I. Gurobi Optimization, "Gurobi optimizer reference manual," 2015. [Online]. Available: http://www.gurobi.com

Wang, Yuquan is a PhD candidate in robotics at the computer vision and active perception lab at KTH since 2012. His research interests include robotic manipulation and machine learning. He received his Bachelor degree in Mechanical Engineering and Automation from Beijing Jiao Tong University in 2008. From 2008 to 2010, he took the master program *Systems, control and robotics* at the Royal institute of technology (KTH). After he received his Master degree of Science in 2010, he worked as a research assistant at the Eindhoven University of Technology until the end of 2011.

Smith, Christian is an Assistant Professor in Computer Science at the Computer Vision and Active Perception Laboratory at The Royal Institute of Technology in Stockholm (KTH). He received his MSc in Engineering Physics in 2005, and PhD in Computer Science in 2009, both at KTH. He has been a post-doctorial researcher at Advanced Telecommunications Research international (ATR) in Kyoto, Japan. He is the secretary of the IEEE Robotics and Automation Society Swedish Chapter. Research interests include control and modeling for manipulation and grasping in human-centric environments and human-robot interaction.

Karayiannidis, Yiannis is an Assistant Professor with the Mechatronics research group at Chalmers University of Technology and a researcher with the Centre for Autonomous Systems at Royal Institute of Technology. He received a Diploma in Electrical and Computer Engineering and a PhD from the Aristotle University of Thessaloniki, in 2004 and 2009 respectively, working on control of kinematically uncertain robotic systems under a grant from the State Scholarships Foundation of Greece. His research interests include robot control, manipulation in human-centered environments, dual-arm manipulation, cooperative multi-agent robotic systems and physical human-robot interaction.



Ögren, Petter was born in Stockholm, Sweden, in 1974. He received the M.S. degree in engineering physics and the Ph.D. degree in applied mathematics from the Royal Institute of Technology (KTH), Stockholm, Sweden, in 1998 and 2003, respectively. In the fall of 2001, he visited the Mechanical Engineering Department, Princeton University, Princeton, NJ. From 2003 to 2012 he worked as a senior scientist and deputy research director in Autonomous Systems at the Swedish Defence Research Agency (FOI). He is currently an Associate Professor at the Computer Vision and Active Perception lab (CVAP) at KTH. His current research is focused on modular and flexible control systems inspired by computer games, with the goal of realizing robots that are robust, safe and easy to use. Application domains include search and rescue robotics, and robotic manipulation. He has also done contributions in the areas of multi agent robotics, teleoperation, navigation and obstacle avoidance.