Cooperative control of a serial-to-parallel structure using a virtual kinematic chain in a mobile dual-arm manipulation application

Yuquan Wang, Christian Smith, Yiannis Karayiannidis and Petter Ögren

***Abstract*— In the future mobile dual-arm robots are expected to perform many tasks. Kinematically, the configuration of two manipulators that branch from the same common mobile base results in a serial-to-parallel kinematic structure, which makes inverse kinematic computations non-trivial. The motion of the base has to be decided in a trade-off, taking the needs of both arms into account. We propose to use a Virtual Kinematic Chain (VKC) to specify the common motion of the parallel manipulators, instead of using the two manipulators kinematics directly. With this VKC, we formulate a constraint based programming solution for the robot to respond to external disturbances during task execution. The proposed approach is experimentally verified both in a noise-free illustrative simulation and a real human robot co-manipulation task.**

I. Introduction and related work

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. To increase the workspace of such manipulators, they are sometimes connected to a mobile base, or a torso with articulated joints. In this paper, we use the mobile dual-arm robot shown in Fig. 1 to study the coordination problem between the parallel manipulators and the common mobile base.

Research on cooperative manipulators has received lots of attentions since the 1970s. For serial-to-parallel manipulators, the *operational space formulation* [1] provides dynamic modelling using the end-effector Cartesian space coordinates. With this formulation, the *augmented object model* [2] describes modelling and control of the dynamics of multiple fixed-base serial chain manipulators. Then it is extended to serial-to-parallel structure in [3], 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 survey on cooperative manipulators in [4]. However, sometimes it is difficult to create a dynamic model of reasonable accuracy, e.g., when a human is part of the control loop. Then, a kinematic approach can be used, and in this paper, we propose such an approach to solve the coordination control of serial-to-parallel structures.

We base our solution on constraint based programming, which is an approach for generating reactive robot motion

The authors are with the 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. Yiannis is also with Automatic Control, Automation and Mechatronics, Department of Signal and Systems, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden. e-mail: {yuquan|ccs|yiankar|petter}@kth.se

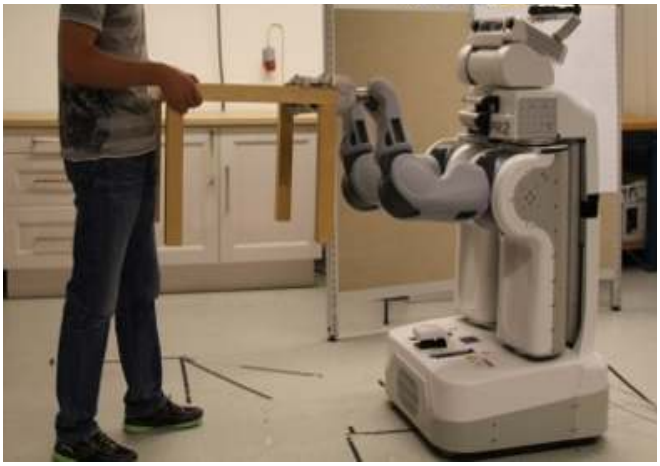

Fig. 1: Example of a human robot co-manipulation task

control for complex robot tasks [5]. By formulating an optimization problem, constraint based programming allows a wide range of sub-tasks to be formulated as inequality or equality constraints [6], [7]. However, the serial-to-parallel structure presents some problems in formulating these constraints [8]. For example, suppose we want the 17 DOF robot in Fig. 1 to co-manipulate a table with a human. Unlike the straight forward inverse kinematics solution for a serial chain, the co-manipulation behaviour is not clearly partitioned into one problem for the left arm, one for the right arm and one for the mobile base. A schematic illustration of the robot kinematic structure can be found in Fig. 2. Consider the common mobile base, its inverse kinematics solution generated using the left arm and the base is different from the inverse kinematics solution generated with the right arm and the base (see Sec. III for a mathematical description).

Therefore in these cases we propose to use a virtual kinematic chain (VKC) to replace the parallel structure when formulating these constraints. Virtual mechanisms are not new in the robotics literature. In the *virtual model control* [9], the legs of a robot are programmed to mimic different virtual mechanical structures in order to control the dynamics along the gravity force direction. In the virtual mechanism approach [10], a VKC was used to chain serial mechanisms together. Our approach is different from [9] in that the VKC is part of a kinematic model, and different from [10] in that we use the VKC to specify the common motion shared by the two parallel structures.

Since we use a VKC to specify the relative motion rather than physically interacting with the environment, we typically use a VKC with less or equal to 6DOF. By choosing a task-dependent VKC, we could explicitly specify the task-

dependent motion for the parallel structure. For example, we could use a virtual rotational joint to specify the orientation and a virtual prismatic joint to specify the translation [10]. We could also specify task-dependant VKC's in the 6 DOF task space in a more systematic way, for example by applying methods such as *instantaneous task specification using constraints* (iTaSC) [5] but that is outside the scope of the current paper.

To summarize our contributions, we use a set of VKC-based constraints to specify the parallel structure motion for robots with serial-to-parallel structures. We integrate these constraints with other different performance measures [6] into a constraint based programming framework. The proposed VKC-based method solves the coordination control with the following benefits: **(1):** Using the VKC, we are able to apply well-developed serial chain control laws/constraints to control a robot with a branching kinematic structure. **(2):** The VKC separates the constraint-specification for the parallel structure and the common serial part. This separation allows us to better explore the redundancy of the robot.

We use the human robot co-manipulation task shown in Fig. 1 as an example. In this case, the robot has 17 DOF. The two parallel arms, each has 7 DOF and branch from the common mobile base which has 3 DOF. The parallel arms apply fixed grasps to the table which is co-manipulated by a human. Note that there is a closed kinematic loop through the co-manipulated rigid object, i.e. the table.

The rest of the paper is organized as follows: in Sec. II, we introduce the notations and preliminaries and then mathematically formulate the problem in Sec. III. The constraint based programming solution is proposed in Sec. IV and validated in Sec. V; we conclude the paper in Sec. VI.

II. Notations and Preliminaries

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

A. Notations

In the following list, we define most of the notations used through out the paper. Note that we use bold symbols for vectors.

- $\boldsymbol{q}$, the joint positions.
- $\boldsymbol{\chi}$, the virtual joint positions of the VKC.
- $R \in SO(3)$, a rotation.
- $\boldsymbol{t} \in \mathbb{R}^3$, a translation.
- $g : \mathbb{R}^4 \to \mathbb{R}^4$, a homogeneous transformation, where $g = (\boldsymbol{t}, R) \in SE(3)$. $g_{i-1,i}$ defines the Euclidean transformation from frame i to frame $i-1$.
- $Ad_g : \mathbb{R}^6 \to \mathbb{R}^6$, 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 $S(\boldsymbol{t})$ denotes the skew-symmetric matrix associated with $\boldsymbol{t}$.

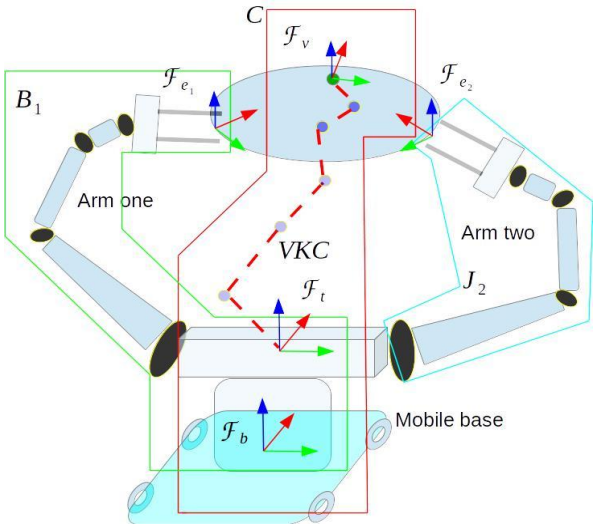

Fig. 2: Parallel kinematic chains branching from a common serial kinematic chain

- $J \in \mathbb{R}^{6\times n}$, a Jacobian matrix of a robot arm with n DOF. We use ${}^{\boldsymbol{t}}J, {}^{\boldsymbol{\omega}}J \in \mathbb{R}^{3\times n}$ to denote its translational and rotational part respectively.
- $\dot{\boldsymbol{t}} \in \mathbb{R}^3$, 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.
- $\boldsymbol{h} \in \mathbb{R}^6$, a wrench.
- D, the positive diagonal damping coefficient matrix.

B. Coordinate frames

We define the coordinate frames with a schematic sketch in Fig. 2. From the robot base to the commonly manipulated object, there are five coordinate frames: the base frame $\mathcal{F}_b$, the torso frame $\mathcal{F}_t$, the left/right end-effector frames $\mathcal{F}_{e_1}, \mathcal{F}_{e_2}$ and the virtual end-effector frame $\mathcal{F}_v$ which is in the middle of the end-effector frames $\mathcal{F}_{e_1}$ and $\mathcal{F}_{e_2}$. Similar to the *cooperative task space* variables [11] we use the following to define $\mathcal{F}_v$:

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

where $R_{\frac{1}{2}}$ is a half[1] 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 wrench measured in $\mathcal{F}_{e_i}$ as $\boldsymbol{h}_i^{e_i}$. 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 $\mathcal{F}_b$.

C. Kinematic chains

As shown in Fig. 2, there are more than one kinematic chain connecting $\mathcal{F}_t$ and $\mathcal{F}_v$. Let us clarify the kinematic chains with a list:

(a) **The common serial chain** starts from $\mathcal{F}_b$ to $\mathcal{F}_t$.

[1] Suppose R denotes the rotation about an axis $\boldsymbol{k}$ by an angle θ. Then $R_{\frac{1}{2}}$ denotes the rotation about the same axis $\boldsymbol{k}$ by an angle $\frac{1}{2}\theta$.

(b) **The arms' chains** branch from $\mathcal{F}_t$ to $\mathcal{F}_{e_i}$.
(c) **The virtual kinematic chain** starts from $\mathcal{F}_t$ to $\mathcal{F}_v$.
(d) **The extended serial chain** starts from $\mathcal{F}_b$ to $\mathcal{F}_v$, which concatenates the common serial chain and the VKC.

D. Base-arm Jacobian and base-vkc Jacobian

with the notations in the book [12], we use the following to perform spatial-velocity transformation for different points on a rigid object:

$$\boldsymbol{V}^t_{tf_v} = \boldsymbol{V}^t_{te_i} + Ad_{g_{te_i}} \boldsymbol{V}^{e_i}_{e_i f_v}, \tag{1}$$

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

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

where $B_i = [J_{bt} \;\; Adg_{bt}J^t_{te_i}]$. In a similar manner we concatenate the mobile base joints $\boldsymbol{q}_b$ to the VKC joints $\boldsymbol{\chi}$ using (1) again:

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

where we define a *base-VKC Jacobian*:

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

We marked the robot components corresponding to B_1, J_2 and C in Fig. 2.

III. Problem formulation

Assuming the robot is joint-velocity controlled, we program the robot to passively follow the human with an admittance control law. As we integrate the admittance control law with the other control aspects using the constraint based programming approach, we need to formulate constraints that coordinate the parallel and branching structure on two levels: **(1)** the close loop constraint between the two parallel arms; **(2)** the coordination between the common mobile base and the two parallel arms. We first formulate the close-loop constraint in Sec. III-A. We point out the problem of the coordination control in Sec. III-B by formulating a naive solution. Then we formulate a feasible solution, i.e. a master-slave method, for comparison purposes in Sec. III-C.

A. Close-loop constraint

In case of the fixed grasp, dual-arm manipulation requires a close chain constraint which fixes the relative translation and rotation between the parallel manipulators. We start by defining the orientation error for the relative orientation $R^{e_1}_{e_1e_2} = {R^t_{te_1}}^\top R^t_{te_2}$ as:

$$\Delta Q^{e_1}_{e_1e_2} = {Q^t_{te_1}}^{-1} * Q^t_{te_2} - {}^*Q^{t\,-1}_{te_1} * {}^*Q^t_{te_2}.$$

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:

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

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

$$\begin{aligned} &\left({R^t_{te_1}}^\top + {R^t_{te_1}}^\top S(\boldsymbol{t}^t_{te_1} - \boldsymbol{t}^t_{te_2})\right)^{\boldsymbol{t}} J_1\dot{\boldsymbol{q}}_1 - {R^t_{te_1}}^{\top\;\boldsymbol{t}} J_2\dot{\boldsymbol{q}}_2 \\ &= -k\Delta\boldsymbol{t}^{e_1}_{e_1e_2}. \end{aligned} \tag{3}$$

B. Straight-forward implementation

Let us illustrate the difficulty of coordinating the parallel arms to the mobile base with a naive candidate solution. Suppose we construct an admittance control law as: $\boldsymbol{V}^v_{bv} = D^{-1}\boldsymbol{h}^v$, where $\boldsymbol{h}^v = \boldsymbol{h}^v_1 + \boldsymbol{h}^v_2$. Then we express this control law in f_b as:

$$\boldsymbol{V}_{bv} = Ad_{g_{bv}}\boldsymbol{V}^v_{bv} = Ad_{g_{bv}}D^{-1}\boldsymbol{h}^v, \tag{4}$$

Apply the above for the two parallel arms respectively we have:

$$\left.\begin{aligned} B_1[\dot{\boldsymbol{q}}_s^\top \dot{\boldsymbol{q}}_1^\top]^\top &= \overbrace{J_{bt}\dot{\boldsymbol{q}}_s}^{\text{mobile base}} + Adg_{bt}J_1\dot{\boldsymbol{q}}_1 \\ B_2[\dot{\boldsymbol{q}}_s^\top \dot{\boldsymbol{q}}_2^\top]^\top &= \underbrace{J_{bt}\dot{\boldsymbol{q}}_s}_{\text{mobile base}} + Adg_{bt}J_2\dot{\boldsymbol{q}}_2 \end{aligned}\right\} = \boldsymbol{V}_{bv}, \tag{5}$$

If we put the above on a more explicit form:

$$J_{bt}\dot{\boldsymbol{q}}_s = \left\{\begin{array}{l} \boldsymbol{V}_{bv} - Adg_{bt}J_1\dot{\boldsymbol{q}}_1 \\ \boldsymbol{V}_{bv} - Adg_{bt}J_2\dot{\boldsymbol{q}}_2 \end{array}\right. ,$$

we can tell that (5) is self-conflicting for $J_{bt}\dot{\boldsymbol{q}}_s$, since $J_1\dot{\boldsymbol{q}}_1$ in general is not equal to $J_2\dot{\boldsymbol{q}}_2$. In Fig. 3, we illustrate this difference by plotting $\|J_1\dot{\boldsymbol{q}}_1\|_2$ and $\|J_2\dot{\boldsymbol{q}}_2\|_2$.

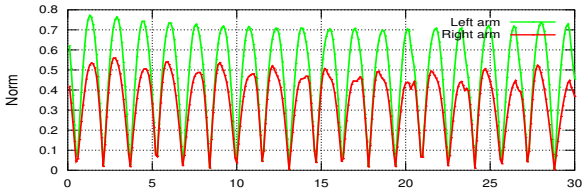

Fig. 3: The difference between $\|J_1\dot{\boldsymbol{q}}_1\|_2$ and $\|J_2\dot{\boldsymbol{q}}_2\|_2$. The data is generated with the master-slave method introduced in Sec. III-C

C. Master-slave implementation

We could apply a master-slave method to resolve the contradictive use of $J_{bt}\dot{\boldsymbol{q}}_s$. For instance we can link only one of the manipulators to the serial part with:

$$B_1[\dot{\boldsymbol{q}}_s^\top \dot{\boldsymbol{q}}_1^\top]^\top = \boldsymbol{V}_{bv} = Ad_{g_{bv}}D^{-1}\boldsymbol{h}^{f_v} \tag{6}$$

and let the other manipulator be a slave by applying the close loop constraint (2-3) on $\dot{\boldsymbol{q}}_2$. We cannot explore the full potential of the serial-to-parallel robot with this approach. First it gives a biased use of $\boldsymbol{q}_s$ in the sense that only $\dot{\boldsymbol{q}}_1$ is supported by $\dot{\boldsymbol{q}}_s$ through constraint (6). Further more, as different constraints are applied on $\dot{\boldsymbol{q}}_1$ and $\dot{\boldsymbol{q}}_2$, $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ differ over time. This difference results in different manipulabilities (9) between the dual arms, which decreases the dual-arm manipulability (12), which is proposed in [13].

Later in the simulation presented in Sec. V-B.1, we validate this weakness as compared to our proposed approach, which is to be introduced in the next section.

IV. Constraint-based programming solution with the virtual kinematic chain

We propose to use a VKC to specify the parallel arm motion, namely we use C instead of B_1 and B_2 to formulate the control law that handles the external disturbances. The proposed approach contains two consequent optimization problems: in the first one we use the control law defined for the concatenated serial chain to solve for $(\dot{\boldsymbol{q}}_s, \dot{\boldsymbol{\chi}})$. The solved common serial chain motion $\dot{\boldsymbol{q}}_s$ is ready to be applied whereas the solved virtual joint motion $\dot{\boldsymbol{\chi}}$ is used to constrain the joint velocities of the parallel structure in a second optimization problem. We summarize this two-phase procedure in Algorithm I at the end of this section. In the following subsections, we provide different functional and the associated gradients that consist of the two optimization problems.

A. Whole-body admittance control constraint

Instead of using a master-slave method (2-3) and (6) or a problematic formulation (2-3) and (5), we specify the parallel arms motion using a VKC according to the task requirement (4):

$$C[\dot{\boldsymbol{q}}_s^\top \dot{\boldsymbol{\chi}}^\top]^\top = \boldsymbol{V}_{bv} = Ad_{g_{bv}} D^{-1} \boldsymbol{h}^v. \tag{7}$$

More details about choosing a VKC for the human robot co-manipulation task (4) are found in a concrete example in Sec. V-A. In practice as different tasks require different desired robot motion, we could choose the VKC in a task-dependent way. For instance, we could use the systematic approach (iTaSC [5]) to specify a set of virtual joints as well as the corresponding Jacobian.

B. Coordination constraint

In this subsection we formulate the *coordination constraints* which let the parallel manipulators behave according to the VKC motion $\dot{\boldsymbol{\chi}}$ such that both of the parallel arms are coordinated with the common serial chain.

1) Coordination constraints: Using the constraint (7) we are able to generate the joint velocities $\dot{\boldsymbol{q}}_s$ and $\dot{\boldsymbol{\chi}}$. We can directly apply $\dot{\boldsymbol{q}}_s$ to the low-level joint velocity controllers, whereas we use $\dot{\boldsymbol{\chi}}$ to constrain $\dot{\boldsymbol{q}}_1$, $\dot{\boldsymbol{q}}_2$. As we have fixed grasps to rigid object we have: $\boldsymbol{V}_{e_i v}^{e_i} = \boldsymbol{0}$. Then we use (1) to transform $\boldsymbol{V}_{tv}^t$:

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

which leads to:

$$\left.\begin{array}{ll} J_1 \dot{\boldsymbol{q}}_1 & = \boldsymbol{V}_{te_1}^t \\ J_2 \dot{\boldsymbol{q}}_2 & = \boldsymbol{V}_{te_2}^t \end{array}\right\} = \boldsymbol{V}_{tv}^t = J_{tv}^t \dot{\boldsymbol{\chi}}, \tag{8}$$

which means that in the torso frame $\mathcal{F}_t$, the two end-effector velocities $\boldsymbol{V}_{te_i}^t$, for $i = 1, 2$ are equal to the virtual end-effector velocity $\boldsymbol{V}_{tv}^t$.

Remark 4.1: Note that the VKC typically has a different reachable workspace than the parallel arms. As we constrain the parallel arms with the VKC end-effector motion, we avoid infeasible solutions by adding a slack variable to (8). The constraint formulation with a slack variable is straightforward and the details are stated in Sec. IV-D.1. ■

2) Close-loop constraints: We use the constraint (2-3), which is defined for a general dual arm system that applies fixed grasps to a rigid object.

C. Secondary constraints

The parallel arms motion in Cartesian space has less than or equal to 6 DOF, so does the VKC. If the parallel arms have more DOF than the VKC, then they have the redundancy to perform secondary constraints listed in this section.

1) Maximizing the velocity manipulability ellipsoid: We denote the volume of the velocity manipulability ellipsoid as:

$$\gamma_m = \det(JJ^\top)^{\frac{1}{2}}, \tag{9}$$

which is defined w.r.t. $\boldsymbol{V}_{tv}^t$, namely we define γ_m using J_{tv}^t, whereas we neglect the sub-superscripts for notation compactness. Since γ_m measures the robot velocity generation ability [14], we aim to maximize γ_m with its gradient:

$$\frac{\partial \gamma_m}{\partial q_i} = \frac{1}{2} \det(\mu)^{-\frac{1}{2}} \det(\mu) \mathrm{Tr}[\mu^{-1} \frac{\partial \mu}{\partial q_i}],$$

where $\mu = JJ^\top$ and we have: $\frac{\partial \mu}{\partial q_i} = \left(\frac{\partial J}{\partial q_i} J^\top + J \frac{\partial J}{\partial q_i}^\top \right)$. The derivative $\frac{\partial J}{\partial q_i}$ is found in appendix II, where we supplement the close-form gradients for a body velocity Jacobian introduced in [15] with a spatial velocity Jacobian case.

2) Avoiding obstacle constraints: Assuming that the obstacle positions are known, avoiding obstacles can be formulated in terms of the minimal distance as

$$\gamma_o = \min ||\boldsymbol{t}_{wr}^w - \boldsymbol{t}_{wo}^w||_2$$

where $\boldsymbol{t}_{wr}^w$ denotes the robot position in a world frame $\mathcal{F}_w$ and $\boldsymbol{t}_{wo}^w$ denotes the obstacle position. We differentiate γ_o w.r.t. the joint positions $\boldsymbol{q}$:

$$\frac{\partial \gamma_o}{\partial \boldsymbol{q}} = 2 \left(\boldsymbol{t}_{wr}^w - \boldsymbol{t}_{wo}^w \right)^\top \frac{\partial \boldsymbol{t}_{wr}^w}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}}$$

where $\frac{\partial \boldsymbol{t}_{wr}^w}{\partial \boldsymbol{q}}$ corresponds to a robot Jacobian. If we want to formulate an obstacle avoidance constraint $\gamma_o \leq \gamma_{do}$ defined for the mobile base joint $\dot{\boldsymbol{q}}_s$, we have the following:

$$2 \left(\boldsymbol{t}_{wb}^w - \boldsymbol{t}_{wo}^w \right)^\top \; J_{wb}^w \dot{\boldsymbol{q}}_s \leq -k(\gamma_o - \gamma_{do}), \tag{10}$$

where k is a gain. Intuitively $2 \left(\boldsymbol{t}_{wb}^w - \boldsymbol{t}_{wo}^w \right)^\top J_{wb}^w \dot{\boldsymbol{q}}_s$ gives us the mobile base velocity towards the direction: $\boldsymbol{t}_{wb}^w - \boldsymbol{t}_{wo}^w$. Note that (10) only makes use of $\dot{\boldsymbol{q}}_s$ for base obstacle avoidance, which separates the use of joints that belong to the parallel part for other constraints. If for instance we want to limit the height of an end-effector: $\gamma_h \leq \gamma_{dh}$, we formulate the constraint as:

$$2 \left(\boldsymbol{t}_{te_i}^t - \boldsymbol{t}_{to}^t \right)^\top \frac{\partial \boldsymbol{t}_{te_i}^t}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}} \leq -k(\gamma_h - \gamma_{dh}), \tag{11}$$

where $\frac{\partial \boldsymbol{t}_{te_i}^t}{\partial \boldsymbol{q}} = J_i$ is a spatial velocity Jacobian for the ith manipulator.

D. Constraints integration

Following previous works in constraint-based programming, we integrate these constraints with the multi-objective control framework [6][7]. We formulate two consequent quadratic programming problems (QPs) on-line to solve for the joint velocities. The first QP computes $\dot{\boldsymbol{q}}_s$ and $\dot{\boldsymbol{\chi}}$. The second QP computes $\dot{\boldsymbol{q}}_1$ and $\dot{\boldsymbol{q}}_2$. The reason why we prefer QP to fast linear programming is that it facilitates the minimization of the joint velocities by penalizing its 2-norm. In this section we provide the rules to construct inequality and equality constraints with constraint gradients. We treat inequality and equality in the same form except that we specify a margin m for inequalities. We summarize the constraints used in the two QPs in Algorithm I.

1) Margin and convergence: Suppose we require $\gamma \leq \gamma_d$, we formulate the constraints by explicitly specifying the margin m and convergence rate k as:

$$\frac{\partial \gamma}{\partial q}^{\top} \dot{\boldsymbol{q}} + \nu \leq -k((\gamma - \gamma_d) + m),$$

where ν denotes the slack variable. By penalizing $w_i \nu_i^2$ in the objective, we use the weight w_i to specify the price to break the constraint. The slack variable ν also solves the infeasibility problem such that the solver always gives the best possible result.

2) Time feed-forward term: If we have a time-variant constraint and its gradient w.r.t. time, we could add a time feed-forward term to include explicit time dependency in the constraint formulation above:

$$\frac{\partial \gamma}{\partial q}^{\top} \dot{\boldsymbol{q}} + \nu \leq -k((\gamma - \gamma_d) + m) - \frac{\partial \gamma}{\partial t},$$

where we used the derivative $\dot{\gamma}(q(t), u, t) = \frac{d\gamma}{dt} = \frac{\partial \gamma}{\partial q}^{\top} \frac{dq}{dt} + \frac{\partial \gamma}{\partial t} = \frac{\partial \gamma}{\partial q}^{\top} \dot{\boldsymbol{q}} + \frac{\partial \gamma}{\partial t}$.

Algorithm 1: Dual-arm mobile manipulator control using a virtual kinematic chain method

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

1 Formulate the first QP which handles the external disturbance, e.g. (4), with constraint (7).
2 Solve the first QP for $\dot{\boldsymbol{q}}_s$ and $\dot{\boldsymbol{\chi}}$.
3 Formulate the second QP which constrains $\dot{\boldsymbol{q}}_1$ and $\dot{\boldsymbol{q}}_2$ to $\dot{\boldsymbol{\chi}}$ using the equality constraint (8).
4 Solve the second QP for $\dot{\boldsymbol{q}}_1$ and $\dot{\boldsymbol{q}}_2$.
5 Apply $\dot{\boldsymbol{q}}_1$, $\dot{\boldsymbol{q}}_2$ and $\dot{\boldsymbol{q}}_s$ to the low level joint velocity controllers.

V. Evaluation

We evaluate Algorithm I with both noise-free simulation and experiments on the real robot. In the first simulation, we can see that the proposed solution gives the robot a more consistent dual-arm manipulability compared to the master-slave method. Then in the second simulation, the robot demonstrates the ability to simultaneously fulfil constraints applied on different parts of the serial-to-parallel structure. With the same controller and the same set of parameters, we repeated the simultaneous multiple constraints satisfaction on the real robot except for a different convergence rate.

A. VKC selection

Since the PR2 robot has a three DOF mobile base, in principle we only need to choose the other 3DOF (one translational and two rotational) for the VKC in order to specify a 6DOF interaction between the robot and the human. However in order to use the mobile base and the parallel arms in a synthetic way, we choose a 6DOF VKC to fully use the parallel arms. Since in this proof-of-concept example we do not have any preference of the relative motion, we choose to use a generic 6DOF VKC with three prismatic joints followed by a three rotational spherical joint. This specific choice easily defines the VKC workspace as well as its forward and inverse kinematics. We list the two QPs required by Algorithm I in the following tables. We use *Gurobi* 6.0 to solve the QPs and the update frequency is $200Hz$.

TABLE I: Solve for $\dot{\boldsymbol{q}}_s$ and $\dot{\boldsymbol{\chi}}$ with the extended serial chain

Objective:	**Min.**	**Max.**	**Ref.**
Configuration measure		✓	(9)
Joint velocities: $\|\dot{\boldsymbol{q}}_s\|_2 + \|\dot{\boldsymbol{\chi}}\|_2$	✓		
Slacks: $\|w_5\nu_5\|_2 + \|w_8\nu_8\|_2$	✓		
Constraint:	**Equality**	**Inequality**	**Ref.**
Admittance control constraints	✓		(7)
Obstacle avoidance		✓	(10)
Joint limit constraints		✓	

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

Objective:	**Min.**	**Max.**	**Ref.**
Configuration measure		✓	(9)
Joint velocities: $\|\dot{\boldsymbol{q}}_{1,2}\|_2$	✓		
Slacks: $\|w_6\nu_6\|_2 + \|w_9\nu_9\|_2$	✓		
Constraint:	**Equality**	**Inequality**	**Ref.**
Close-loop constraints	✓		(2-3)
Coordination constraints	✓		(8)
End-effector height		✓	(11)
Joint limit constraints		✓	

B. Simulation evaluation

We separate the simulation into two parts: dual-arm manipulability comparison and multiple constraints satisfaction. In the former case, simulation results support the fact that the proposed method has a better dual-arm coordination ability compared to methods that have a biased use of the arm, e.g. the master-slave method listed in Table III. In the latter case, we have additional constraints: (10) defined for the base and (11) defined for the two arms. The satisfaction of all the constraints applied on different parts of the robots indicates the fact that the robot is indeed well coordinated.

1) Dual-arm manipulability comparison: We measure the dual-arm velocity manipulability of the two manipulators w.r.t. frame f_v using the measure proposed in [13]. Suppose we denote the volume of the velocity manipulability ellipsoid

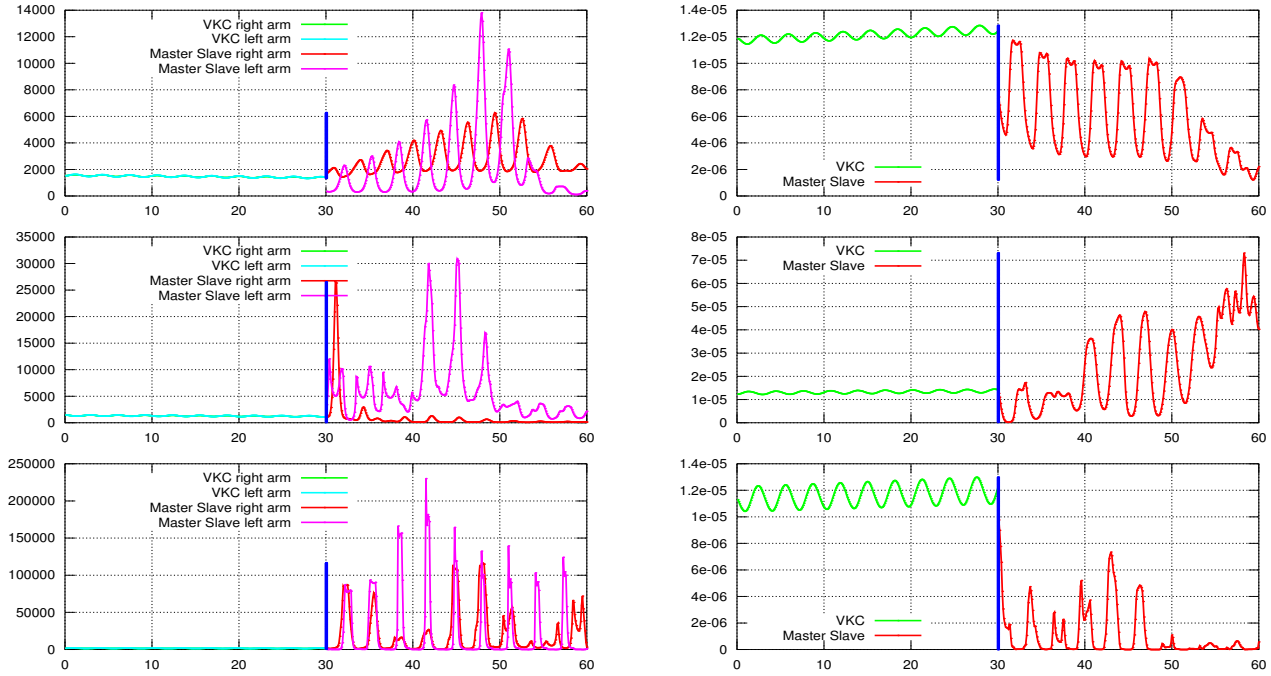

Fig. 4: **Left:** the velocity manipulability measure (9) for the two arms. **Right:** The dual arm manipulability measure (12). The blue line indicates that we switch from the proposed method to the master-slave method. We can tell that due to a bigger overlap of the measure (9) between the two arms, the proposed method has a more stable dual-arm manipulability.

TABLE III: Master/slave method

Objective:	**Min.**	**Max.**	**Ref.**
Configuration measure		✓	(9)
Joint velocities: $\|\dot{\boldsymbol{q}}_{s,1,2}\|_2$	✓		
Slacks for equalities: $\sum_i w_i \|\nu_i\|_2$	✓		
Constraint:	**Equality**	**Inequality**	**Ref.**
Master chain	✓		(6)
Close-loop constraint	✓		(2-3)
Joint limit constraints		✓	

of the ith arm as γ_{vm_i}, the dual-arm velocity manipulability measure is proportional to the intersection of γ_{vm_1} and γ_{vm_2}:

$$\gamma_{dm} \sim \gamma_{vm_1} \cap \gamma_{vm_2}. \tag{12}$$

In the noise-free simulation we simulate six dimensional sinus waves as force-torque measurements and use them as the input to the admittance control law (4). From the first row to the third row in Fig. 4, we used the wave amplitude: $1N$, $2N$ and $3N$. On right half of Fig. 4, we can see that the proposed solution is better noise-resistant in the sense that its dual-arm manipulability measure does not vary over time as much as the master-slave method. The reason is revealed by the left half of Fig. 4, as using the proposed method the manipulability ellipsoids of the two arms have a bigger overlap.

2) Multiple constraints satisfaction: We use the solution listed in Table I, II and plot the results in Fig. 5. Since (10) is defined for the base, (11) is defined for separate arm and (7)[2] is defined for the whole robot, the simultaneous satisfaction of these constraints indicate a good coordination of the robot. From the last three rows of Fig. 5, we can see that (11) and (10) are gradually satisfied even though they are not under the initial configuration. This can be verified from the second row, where we see that the constraint scores $\|w_i\nu_i\|_2$ converge to zero. On the other hand in the first row, we see that the optimizer minimizes the use of the $\dot{\boldsymbol{\chi}}$ while taking sinus-wave input to the admittance control task (4).

If we run the same controller with the same simulated wrench on the real robot, we will have a similar result, except for a different convergence rate due to a different noise level in practice. Due to the un-periodic human input, we did not plot results from the real human-robot co-manipulation task. However we refer the interested readers to the link below[3] for an experiment video, where a set of human-robot co-manipulation experiments using the proposed approach is performed.

VI. Conclusion

Using a VKC to specify the motion of the parallel arms, we treat the mobile dual-arm robot as a concatenated serial kinematic chain rather than a serial-to-parallel structure. The concatenated serial kinematic chain facilitates formulating constraints that involve the mobile base and the parallel arms, e.g. the whole-body admittance control constraint (7). If the arm has more DOF than the VKC, which has less than or equal to 6DOF, we could use this redundancy to specify additional constraint for the arms, e.g. constraint (11). If it does not conflict with the concatenated serial kinematic chain constraint, e.g. (7), we could formulate

[2] Please note that in this set of simulation, we simulated sinus-wave force in the y and z axis and sinus-wave torque about the x axis.

[3] The human-robot co-manipulation experiment video and code using the proposed approach:`http://youtu.be/HO_amCdft-A`

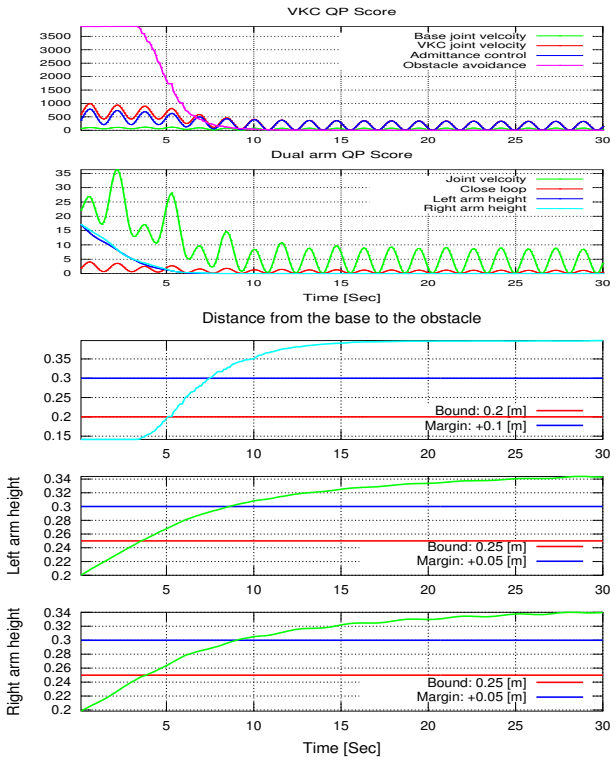

Fig. 5: Simulation validation of multiple-constraints satisfaction

additional constraint for the mobile base, e.g. (10) which realizes mobile manipulation for the robot.

In the future, we will apply the proposed solution to other mobile dual-arm manipulation tasks. For instance, if the task requires a loose dual-arm cooperation, we may specify the relative motion between the two arms with the iTaSC approach [5] rather than the close-loop constraint (2-3). If we keep (2-3) but release the fixed grasp assumption, we may introduce unactuated joints between the end-effector frame and the objects. We may also use the proposed method to formulate whole-body behaviour for other robots with a serial-to-parallel structure, e.g. a dual-arm robot with a bendable torso, or a robot arm with multiple active fingers.

APPENDIX I
BODY-SPATIAL VELOCITY TRANSFORMATION

In this paper, we apply the body velocity and spatial velocity transformation: $\boldsymbol{V}_{bv} = Adg_{bv}\boldsymbol{V}^v_{bv}$ as well as the associated jacobian transformation: $J_{bv} = Adg_{bv}J^v_{bv}$. However there is another way in [16] as the following:

$$J_{bv} = \begin{bmatrix} R_{bv} & 0 \\ O & R_{bv} \end{bmatrix} J^v_{bv}. \tag{13}$$

However this is a special case for the origin of the end-effector frame. We have the following equality:

$$\begin{bmatrix} R_{bv} & \mathbf{0} \\ O & 1 \end{bmatrix} \hat{\boldsymbol{V}}^v_{bv} \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} = \hat{\boldsymbol{V}}_{bv} \begin{bmatrix} \boldsymbol{t}_{bv} \\ 1 \end{bmatrix},$$

which could be expanded as:

$$\begin{bmatrix} R_{bv} & \mathbf{0} \\ O & 1 \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\omega}}^v_{bv} & \boldsymbol{v}^v_{bv} \\ O & 0 \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{\omega}}_{bv} & \boldsymbol{v}_{bv} \\ O & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{t}_{bv} \\ 1 \end{bmatrix} \tag{14}$$

Using the homogeneous transformation $[\boldsymbol{t}^\top_{bv} 1]^\top = g_{bv}[\mathbf{0}^\top 1]^\top$, the similarity transformation: $\hat{\boldsymbol{V}}^b_{bv} = g_{bv}\hat{\boldsymbol{V}}^v_{bv}g^{-1}_{bv}$ and the fact that

$$\begin{bmatrix} I & \boldsymbol{t}_{bv} \\ O & 1 \end{bmatrix} \hat{\boldsymbol{V}} = \hat{\boldsymbol{V}},$$

we could expand the left hand side of (14) to:

$$\begin{bmatrix} R_{bv} & \mathbf{0} \\ O & 1 \end{bmatrix} \begin{bmatrix} I & \boldsymbol{t}_{bv} \\ O & 1 \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\omega}}^v_{bv} & \boldsymbol{v}^v_{bv} \\ O & 0 \end{bmatrix} g^{-1}_{bv} \begin{bmatrix} \boldsymbol{t}_{bv} \\ 1 \end{bmatrix}$$
$$= g_{bv}\hat{\boldsymbol{V}}^v_{bv}g^{-1}_{bv} \begin{bmatrix} \boldsymbol{t}_{bv} \\ 1 \end{bmatrix} = \hat{\boldsymbol{V}}_{bv} \begin{bmatrix} \boldsymbol{t}_{bv} \\ 1 \end{bmatrix}.$$

Therefore we say (13) is a special case of

$$J_{bv} = Adg_{bv}J^v_{bv}.$$

APPENDIX II
MANIPULATOR DIFFERENTIAL KINEMATICS

For a general mapping $g(\boldsymbol{q}) \in SE(3)$, its derivative $\dot{g}(\boldsymbol{q}) \notin se(3)$. Rather, we have the instantaneous spatial velocity given by $\frac{\partial g}{\partial \boldsymbol{q}} g^{-1} \in se(3)$, where we ignore $\boldsymbol{q}$ for notation compactness. Let us expand $\frac{\partial g}{\partial \boldsymbol{q}} g^{-1}$ as: $\sum_{i=1}^{n} \left(\frac{\partial g}{\partial q_i} g^{-1}\right) \dot{q}_i$. Since $\frac{\partial g}{\partial \boldsymbol{q}_i} g^{-1}$ is matrix evaluated, we use the twist coordinate $\left(\frac{\partial g}{\partial q_i} g^{-1}\right)^\vee$ to put them into a compact form, see [12], as:

$$\boldsymbol{V} = J\dot{\boldsymbol{q}}$$

where $J = [\left(\frac{\partial g}{\partial \theta_1} g^{-1}\right)^\vee \ldots \left(\frac{\partial g}{\partial \theta_n} g^{-1}\right)^\vee]$ and $^\vee$ denotes the vee operator. In each column we have the twist coordinate $\boldsymbol{\xi}'_i = \left(\frac{\partial g}{\partial \theta_i} g^{-1}\right)^\vee$:

$$\boldsymbol{\xi}'_i = Ad_{g_{1,i-1}}\boldsymbol{\xi}_i \tag{15}$$

where $Ad_{g_{1,i-1}}\boldsymbol{\xi}_i = g_{1,i-1}\boldsymbol{\xi}_i g^{-1}_{1,i-1}$ and

$$g_{1,i-1} = e^{\hat{\boldsymbol{\xi}}_1 q_1} \ldots e^{\hat{\boldsymbol{\xi}}_{i-1} q_{i-1}}$$

, where we used the *product of exponentials formula* (POE [12]). Then using Lemma I we have the closed-form derivative $\frac{J}{\partial q_j} \in \mathbb{R}^{6\times n}$ for $j = 1, \ldots, n$ as:

$$\frac{\partial J}{\partial q_j} = [\frac{\partial \boldsymbol{\xi}'_1}{\partial q_j} \; \frac{\partial \boldsymbol{\xi}'_2}{\partial q_j} \ldots \; \frac{\partial \boldsymbol{\xi}'_n}{\partial q_j}]\dot{q}_j. \tag{16}$$

Lemma 1: For each column of the Jacobian, we have

$$\frac{\partial}{\partial q_j}\boldsymbol{\xi}'_i = \begin{cases} [\boldsymbol{\xi}'_j \; \boldsymbol{\xi}'_i], & j \le i \le n \\ \mathbf{0} & i < j \le n \end{cases},$$

where the Lie bracket $[\boldsymbol{\xi}'_j \; \boldsymbol{\xi}'_i] = (\boldsymbol{\omega}_j \times \boldsymbol{\nu}_i - \boldsymbol{\nu}_j \times \boldsymbol{\omega}_i, \boldsymbol{\omega}_j \times \boldsymbol{\omega}_i)$.

Proof 1: In case that $i < j$, using (15) we know that $\boldsymbol{\xi}'_i$ is not a function of q_j, otherwise:

$$\frac{\partial}{\partial q_j}\hat{\boldsymbol{\xi}}'_i = \frac{\partial g_{1,i-1}}{\partial q_j}\hat{\boldsymbol{\xi}}_i g^{-1}_{1,i-1} + g_{1,i-1}\hat{\boldsymbol{\xi}}_i \frac{\partial g^{-1}_{1,i-1}}{\partial q_j} \tag{17}$$

where we have

$$\begin{aligned}\frac{\partial g_{1,i-1}}{\partial q_j} &= e^{\hat{\boldsymbol{\xi}}_1 q_1} \dots \hat{\boldsymbol{\xi}}_j e^{\hat{\boldsymbol{\xi}}_j q_j} \dots e^{\hat{\boldsymbol{\xi}}_{i-1} q_{i-1}} \\ &= \hat{\boldsymbol{\xi}}_j' g_{1,i-1}\end{aligned} \tag{18}$$

$$\begin{aligned}\frac{\partial g_{1,i-1}^{-1}}{\partial q_j} &= -e^{-\hat{\boldsymbol{\xi}}_{i-1} q_{i-1}} \dots \hat{\boldsymbol{\xi}}_j e^{-\hat{\boldsymbol{\xi}}_j q_j} \dots e^{-\hat{\boldsymbol{\xi}}_1 q_1} \\ &= -e^{-\hat{\boldsymbol{\xi}}_{i-1} q_{i-1}} \dots \hat{\boldsymbol{\xi}}_j e^{-\hat{\boldsymbol{\xi}}_j q_j} \dots e^{-\hat{\boldsymbol{\xi}}_1 q_1} \\ &= -e^{-\hat{\boldsymbol{\xi}}_{i-1} q_{i-1}} \dots e^{-\hat{\boldsymbol{\xi}}_j \hat{\boldsymbol{\xi}} q_j \hat{\boldsymbol{\xi}}_j^{-1}} \hat{\boldsymbol{\xi}}_j \dots e^{-\hat{\boldsymbol{\xi}}_1 q_1} \\ &= -g_{1,i-1}^{-1} \hat{\boldsymbol{\xi}}_j'\end{aligned} \tag{19}$$

where we used the fact that $g^{-1} e^{\hat{\boldsymbol{\xi}} q} g = e^{g^{-1} \hat{\boldsymbol{\xi}} q g}$. Plug (18) and (19) into (17), we obtain:

$$\begin{aligned}\frac{\partial}{\partial q_j} \hat{\boldsymbol{\xi}}_i &= \hat{\boldsymbol{\xi}}_j' g_{1,i-1} \hat{\boldsymbol{\xi}}_i g_{1,i-1}^{-1} - g_{1,i-1} \hat{\boldsymbol{\xi}}_i g_{1,i-1}^{-1} \hat{\boldsymbol{\xi}}_j' \\ &= \hat{\boldsymbol{\xi}}_j' \hat{\boldsymbol{\xi}}_i' - \hat{\boldsymbol{\xi}}_i' \hat{\boldsymbol{\xi}}_j' \\ &= [\hat{\boldsymbol{\xi}}_j' \quad \hat{\boldsymbol{\xi}}_i'].\end{aligned}$$

■

ACKNOWLEDGEMENT

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). The authors gratefully acknowledge the support.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] F. Caccavale and M. Uchiyama, "Cooperative manipulators," *Springer handbook of robotics*, pp. 701–718, 2008.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] N. Likar, B. Nemec, and L. Žlajpah, "Virtual mechanism approach for dual-arm manipulation," *Robotica*, pp. 1–16, 2013.

[11] F. Caccavale, P. Chiacchio, and S. Chiaverini, "Task-space regulation of cooperative manipulators," *Automatica*, vol. 36, no. 6, pp. 879–887, 2000.

[12] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[13] 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.

[14] T. Yoshikawa, "Manipulability of robotic mechanisms," *The international journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[15] 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.

[16] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2009.

[17] F. C. Park, "Computational aspects of the product-of-exponentials formula for robot kinematics," *Automatic Control, IEEE Transactions on*, vol. 39, no. 3, pp. 643–647, 1994.