2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Daejeon Convention Center
October 9-14, 2016, Daejeon, Korea

Reactive Task-oriented Redundancy Resolution using Constraint-Based Programming

Yuquan Wang and Lihui Wang

Abstract— **Constraint based programming provides a versatile framework for combining several different constraints into a single robot control scheme. We take advantage of the redundancy of a robot manipulator to improve the execution of a reactive tracking task, in terms of a task-dependent measure which is a weighted sum of velocity transmissions along the current directions of motion. With inspiration from recent work, we provide analytical gradients and computable weights of the task-dependent measure, which enable us to include it in a reactive constraint based programming framework, without relying on inexact numerical approximations and manually tuning weights. The proposed approach is illustrated in a set of simulations, comparing the performance with a standard constraint based programming method.**

I. Introduction

As robots continue to play an increasingly important role in areas ranging from manufacturing to care of elderlies, the need for robust and reliable manipulation capabilities is expected to grow. Furthermore, robot systems exhibit an increasing amount of redundancy, as 6 degree-of-freedom (DOF) manipulators are often replaced by 7 DOF arms, sometimes mounted on mobile systems. Finally, as the environment of the robots become more unstructured and uncertain, there is a need for both reactive and planning systems. When collaborating with a human, a robot needs to react and respond to the actions of the human, and cannot rely on the execution of a carefully pre-planned trajectory.

Constraint based programming for reactive motion generation [1] has received a lot of attention, as it enables the execution of highly complex robot tasks. The strength of constraint based programming is that it facilitates the formulation and solution of a wide range of robot control problems, where a number of different, possibly contradicting constraints, or objectives, needs to be taken into account. In this paper, we propose to optimize a weighted sum of the force and velocity transmission ratios [2] while performing a reactive task using constraint based programming.

Since the 1980s, the velocity ellipsoid of robotic mechanisms [3] has been widely used to measure the robot manipulability in terms of the velocity transmission ratio from joint velocities to the end-effector velocity. However in the context of a specific task, the transmission ratios along the task-dependent directions are more important. For instance, in a trajectory tracing task we would like to have a large velocity transmission ratio along the trajectory tangential direction in order to enable fast motions in that direction, and at the same time have a small force transmission ratio in the surface normal direction of a rigid working piece to avoid large interaction forces. A task-dependent measure, which is a weighted sum of the translational velocity and rotational velocity transmission ratios along the task-dependent directions, is proposed in [2]. Using more recent approaches [4], [5], we can integrate constraints that are describing different tasks by formulating a Quadratic Programming (QP) problem. In order to formulate a constraint that corresponds to the task-dependent measure, we need two steps. First, we need to compute the analytical gradients of the transmission ratios. Second, we need to tune the weights correspond to the translational and rotational velocities transmission ratios in the task-dependent measure.

We obtain the analytical gradient of the transmission ratios using the analytical gradient of a spatial velocity Jacobian of a manipulator based on the product of exponentials formula [6] and the analysis of higher order differential kinematics [7], [8].

The manual selection of the weights in the task-dependent measure [2] could result in inaccuracies and requires a time-consuming tuning procedure. In light of the prioritized tasks [9], where the task with a lower priority is solved in the null space of a higher priority task, we use the minimal principal angle [10] between the subspaces spanned by the translation and rotation Jacobian to weight the translational velocity transmission ratio against the rotational velocity transmission ratio.

To summarize our contributions, we improve the reactive task-oriented redundancy resolution by including a task-dependent measure in the objective of the QP. The gradient of the task-dependent measure is found by direct differentiation of the product of exponentials formula [6]. The translational and rotational velocity transmission ratios are balanced with respect to each other by computable weights. We use simulations of a redundant robot arm curve tracing task to step by step validate the analytical gradients, the task-dependent measure and the computable weights.

The rest of the paper is organized as follows. We relate the proposed method to the state of the art in Sec. II. We introduce the mathematical preliminaries in Sec. III and formulate the reactive trajectory tracing problem in Sec. IV. We formulate the task-dependent measure as a constraint and include it in a redudancy resolution framework in Sec. V. We validate the proposed solution by simulations in Sec. VI and conclude the paper in Sec. VII.

Yuquan and Lihui are researchers from Department of Production Engineering, School of Industrial Engineering and Management (ITM) at Royal Institute of Technology (KTH), SE-100 44 Stockholm, Sweden. e-mail: {yuquan}@kth.se

II. Related work

The analytical gradient of the transmission ratios involves the eigenvalues of the product of the manipulator Jacobian and its transpose. As reported in [11] and [12], we can quantitatively define a singular region within which the the product of the manipulator Jacobian and its transpose is ill-conditioned and therefore inverse kinematics algorithms [1], [4], [5], [9] can not be applied. In order to avoid the singular region and apply the analytical gradients, we use an inequality constraint in the QP along with other constraints that specify a certain task.

Different from the cases when the subspaces are orthogonal [1], [11], [9], the minimal principal angle between the subspaces is used as a pointwise criterion to reduce conservativeness in the analysis of robustness in [13], where the conclusion is that the system remains stable as long as perturbations are smaller than the sine of the minimal principal angle.

As pointed out in [14], the physical meaning of the velocity manipulability ellipsoid is not clear in the sense that it is a sum of the translational and rotational velocities. In the analysis of the transmission ratios, we separate the translational and rotational velocity manipulability ellipsoids. A similar separation could be found in [15], where the kinematics of manipulators with spherical wrists are analyzed. The velocity manipulability ellipsoid fails another criteria raised in [14], that is, it is not invariant with respect to the change of coordinate frame. We can use the force manipulability ellipsoid in the analysis, which meets the invariance criteria [16] and has reciprocal eigenvalues to the velocity manipulability ellipsoid.

The velocity manipulability has successful applications in kinematic analysis, where derivations of the velocity ellipsoid are used to analyze closed-chain kinematics with unactuated joints [17] and mobile manipulators [18].

III. Notations and preliminaries

In order to facilitate the theoretical analysis, we list the notations used through out the paper, introduce the coordinate frames, velocity transformations and the Jacobians.

A. Notations

In this notation list we use bold symbols for vectors.

- $\boldsymbol{\theta}$ - the joint positions.
- $R \in SO(3)$, a rotation with three columns $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^3$.
- $\boldsymbol{t} \in \mathbb{R}^3$ - a translation.
- $\boldsymbol{p} \in \mathbb{R}^3$ - a point in Cartesian space.
- $g : \mathbb{R}^4 \to \mathbb{R}^4$ - a homogeneous transformation, where $g = (\boldsymbol{t}, R) \in SE(3)$. $g_{i,i-1}$ defines the Euclidean transformation of frame $i-1$ with respect to frame i.
- $Ad_g : \mathbb{R}^6 \to \mathbb{R}^6$ - an adjoint transformation. Given $g = (\boldsymbol{t}, R) \in SE(3)$, Ad_g and its inverse are:

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

where we use $s()$ to denote the skew-symmetric matrix.

- $\boldsymbol{V} = [\dot{\boldsymbol{t}}^\top \ \boldsymbol{w}^\top]^\top$ a spatial velocity, where $\dot{\boldsymbol{t}} \in \mathbb{R}^3$ denotes translational velocity and $\boldsymbol{\omega} \in \mathbb{R}^3$ denotes the rotational velocity.
- $J \in \mathbb{R}^{6\times n}$ - a Jacobian matrix of a manipulator. We use $J_t, J_\omega \in \mathbb{R}^{3\times n}$ to denote its translational and rotational part respectively.

B. Coordinate frames

In order to formulate a trajectory-tracing task for a manipulator equipped with a tool, we need to define five coordinate frames as shown in Fig. 1, where we use red-green-blue colour to denote x-y-z direction. We use the fixed robot base

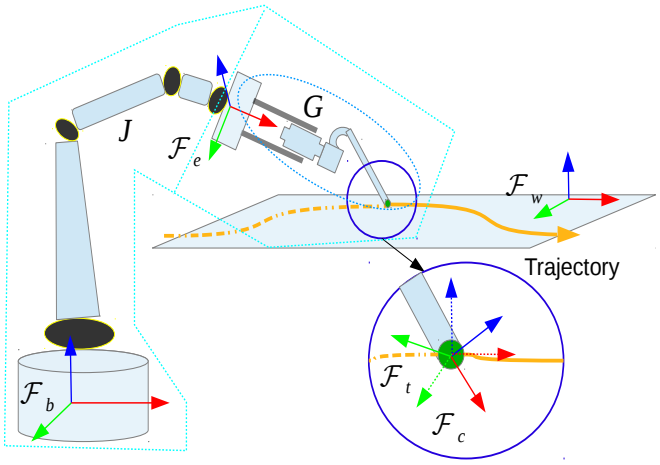

Fig. 1: Coordinate frames and Jacobians in a trajectory-tracing task

frame $\mathcal{F}_b$ as the global frame. We denote the robot end-effector frame as $\mathcal{F}_e$ and denote the tool-tip frame as $\mathcal{F}_t$. Suppose there is a trajectory on the flat surface of a working-piece, we denote the working-piece frame as $\mathcal{F}_w$ and allocate a frame $\mathcal{F}_c$ to the contact point. Note that $\boldsymbol{p}_{\mathcal{F}_c} = \boldsymbol{p}_{\mathcal{F}_t}$, whereas $R_{\mathcal{F}_c} \neq R_{\mathcal{F}_t}$. We assume that the x-direction of $\mathcal{F}_c$ is the tangential direction of the trajectory, the y-direction of $\mathcal{F}_c$ is the normal of the trajectory and then the z-direction of $\mathcal{F}_c$ is the surface normal of the working-piece.

C. Velocity transformation

We use one super-script and two sub-scripts to denote a velocity and the corresponding Jacobian. For example, $\boldsymbol{V}_{bt}^b$ denotes the velocity of $\mathcal{F}_t$ relative to $\mathcal{F}_b$ in the reference frame $\mathcal{F}_b$. In line with [19], we use the following expression to perform spatial-velocity and body-velocity transformation for different points on a rigid object:

$$\boldsymbol{V}_{bt}^b = \boldsymbol{V}_{be}^b + Ad_{g_{be}} \boldsymbol{V}_{et}^e, \tag{1}$$

D. Jacobian

In this section, we introduce the Jacobian that relates the robot joint velocities $\dot{\boldsymbol{\theta}}$ to the tip velocity $\boldsymbol{V}_{bt}^b$. Given that the tool is rigid, we have $\boldsymbol{V}_{et}^e = 0$, which simplifies (1) as:

$$\boldsymbol{V}_{bt}^b = \boldsymbol{V}_{be}^b.$$

This simplified spatial-velocity transformation indicates $J_{be}^b = J_{bt}^b$ since:

$$J_{be}^b \dot{\boldsymbol{\theta}} = \boldsymbol{V}_{be}^b = \boldsymbol{V}_{bt}^b.$$

Therefore the *grasp matrix* $G = Adg_{et}^{\top}$ that is reported in [19] does not affect the analysis of the spatial velocity $\boldsymbol{V}_{bt}^{b}$. For notation compactness, we use J to denote the spatial-velocity Jacobian J_{be}^{b}. The correspondences of J, G are marked in Fig. 1.

IV. Problem Formulation

Using the manipulability ellipsoid, we introduce the task-dependent measure, see [2], based on which we formulate a reactive task-oriented trajectory tracing problem for a redundant robot manipulator with 7 DOF.

The volume of the velocity ellipsoid [3] measures the transmission ratio from joint velocity $\dot{\boldsymbol{\theta}} \in \mathbb{R}^n$ to end-effector velocity $\boldsymbol{V} \in \mathbb{R}^6$. Suppose we have a unit sphere in $\mathbb{R}^n$, which is: $\|\dot{\boldsymbol{\theta}}\|^2 = \dot{\theta}_1^2 + \dot{\theta}_2^2 + \ldots + \dot{\theta}_n^2 \leq 1$. Using a manipulator Jacobian J , we map the unit sphere into the velocity ellipsoid in $\mathbb{R}^6$:

$$\dot{\boldsymbol{\theta}}^T \dot{\boldsymbol{\theta}} = \boldsymbol{V}^T (JJ^{\top})^{-1} \boldsymbol{V} = 1. \tag{2}$$

As reported in [3], we can use

$$\eta_0 = \sqrt{\det JJ^{\top}}, \tag{3}$$

which is proportional to the volume of (2), to measure the velocity-generating ability of the robot. As reported in [11] and [12], the robustness of inverse kinematics algorithms depends on η_0 and we can define a singular region of a manipulator configuration by constraining η_0 with a pre-specified threshold b_{η_0} as:

Definition 1 (Singular region of a manipulator):

$$\mathcal{D} = \{\boldsymbol{\theta} \mid \eta_0 < b_{\eta_0}\}. \tag{4}$$

Within the singular region of a redundant manipulator, the inverse kinematics algorithms are ill-conditioned. ■

In order to keep the manipulator configuration outside the singular region $\mathcal{D}$, we can bound η_0 from below with the inequality:

$$-\eta_0 < -b_{\eta_0}. \tag{5}$$

However in the context of a trajectory tracing task such as the one shown in Fig. 1, the measure η_0 is not able to measure the task execution ability of the robot. In Fig. 1 the robot is holding a tool and needs to trace a curve on a working piece with the tool-tip:

$$\boldsymbol{\eta}_1 = \boldsymbol{p}_t^b - \boldsymbol{p}_c^b(t) = \boldsymbol{0}. \tag{6}$$

Meanwhile the tool-tip also needs to keep an orientation constraint:

$$\eta_2 = \boldsymbol{z}_{bt}^{b\top} \boldsymbol{z}_{bc}^b \geq b_{\eta_2}, \tag{7}$$

where b_{η_2} is an orientation bound, e.g. $cos\frac{\pi}{6}$. If we keep on optimizing η_0, the optimal robot configuration shown in Fig. 2 is unfortunately not related to a given task (6-7).

In Fig. 3, we plot a schematic translational velocity ellipsoid with respect to the translation velocity $\dot{\boldsymbol{t}}_{bc}^b$ in the frame $\mathcal{F}_c$. We can see that even if η_0 is maximized, the principal axes of the manipulability ellipsoid may differ from the task-dependent directions, e.g. $\boldsymbol{x}_{bc}^b$ $\boldsymbol{y}_{bc}^b$ and $\boldsymbol{z}_{bc}^b$. As reported in

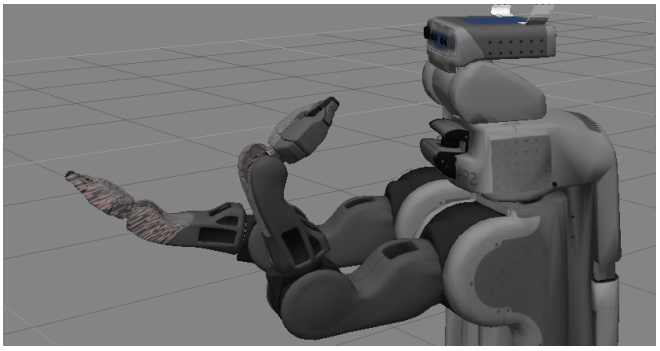

Fig. 2: The robot arm has 7 DOF, which makes it redundant. Its initial configuration corresponds to $\eta_0 = 0.023$ (the same as the back arm). If we keep on maximizing η_0, we end up with $\eta_0 = 0.062$.

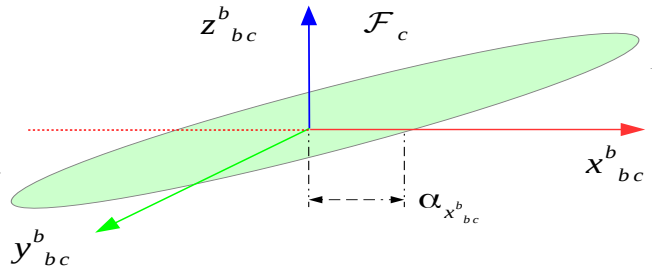

Fig. 3: Given a schematic view of the translational velocity ellipsoid defined by $(J_t J_t^{\top})^{-1}$, we denote its projection along axis $\boldsymbol{x}_{bc}^b$ as $\alpha_{\boldsymbol{x}_{bc}^b}$.

[2], the translational velocity transmission ratio $\alpha_{\boldsymbol{x}_{bc}^b}$ is given as the distance from the origin of the frame $\mathcal{F}_c$ along a task-dependent direction $\boldsymbol{x}_{bc}^b$ to the surface of the velocity ellipsoid:

$$(\alpha_{\boldsymbol{x}_{bc}^b} \boldsymbol{x}_{bc}^b)^{\top} (J_t J_t^{\top})^{-1} (\alpha_{\boldsymbol{x}_{bc}^b} \boldsymbol{x}_{bc}^b) = 1.$$

We can explicitly express $\alpha_{\boldsymbol{x}_{bc}^b}$ as:

$$\alpha_{\boldsymbol{x}_{bc}^b} = [\boldsymbol{x}_{bc}^{b\top} (J_t J_t^{\top})^{-1} \boldsymbol{x}_{bc}^b]^{-\frac{1}{2}}. \tag{8}$$

In a similar way, we can also define a rotational velocity transmission ratio as

$$\beta_{\boldsymbol{z}_{bc}^b} = [\boldsymbol{z}_{bc}^{b\top} (J_\omega J_\omega^{\top})^{-1} \boldsymbol{z}_{bc}^b]^{-\frac{1}{2}}. \tag{9}$$

Based on (8) and (9) we can define a weighted measure η_3 for the trajectory tracing task (5-7) as:

$$\eta_3 = w_\alpha \alpha_{\boldsymbol{x}_{bc}^b} + w_\beta \beta_{\boldsymbol{z}_{bc}^b}. \tag{10}$$

We can increase the velocity transmission ratios along the task-dependent directions, e.g. $\boldsymbol{x}_{bc}^b$ and $\boldsymbol{z}_{bc}^b$, by maximizing (10). In the practical implementation of maximizing (10), we can use the force and moment transmission ratios for better numerical stability and invariance under coordinate transformation, see [16].

Using the unit sphere defined by the joint torques: $\boldsymbol{\tau}^{\top}\boldsymbol{\tau} = 1$ and the kinetostatic mapping we can obtain a force transmission ratio:

$$\gamma_{\boldsymbol{u}_i^c} = [\boldsymbol{u}_i^{c\top} (J_t J_t^{\top}) \boldsymbol{u}_i^c]^{-\frac{1}{2}}, \tag{11}$$

where $\boldsymbol{u}_i^c \in \mathbb{R}^3$ denotes an interested direction in frame $\mathcal{F}_c$, along which we would like to optimize the force transmission ratio. As $\gamma_{\boldsymbol{u}_i^c}$ is invariant under the coordinate transformation and the eigenvalues of $J_t J_t^\top$ are reciprocal to the eigenvalues of $(J_t J_t^\top)^{-1}$, we can minimize $\gamma_{\boldsymbol{u}_i^c}$ instead of maximizing $\alpha_{\boldsymbol{u}_i^c}$ to improve the velocity transmission ratio of the manipulator.

Given the task-dependent measure (10), we state our problem as:

Problem 1: Reactive task-oriented trajectory tracing task for a 7 DOF redundant robot:

- Satisfy the task constraints, e.g. the trajectory tracing task (6-7).
- Keep the manipulator configuration outside the singular region (4) using the constraint (5).
- Maximize the task-dependent measure η_3 to improve the performance while fulfilling (5), (6), (7).

V. Proposed solution

We start by introducing the analytical gradients of $\frac{\partial \eta_i}{\partial \boldsymbol{\theta}}$ for $i \in 0,1,2,3$, then we explain how to calculate the weights w_α and w_β, which enabling us to integrate η_i for $i \in 0,1,2,3$ into the objective of the constraint based programming framework [5] without manual tuning.

A. The analytical gradients

Based on the analytical gradient of the spatial velocity Jacobian $\frac{\partial J}{\partial \boldsymbol{\theta}_i}$ for $i = 1, \ldots, n$, we can obtain the following gradients using straight forward calculations:

1) $\frac{\partial \eta_0}{\partial \boldsymbol{\theta}}$*:* We differentiate η_0 in (3) as:

$$\frac{\partial \eta_0}{\partial \theta_i} = \frac{\partial \det(JJ^\top)^{\frac{1}{2}}}{\partial \theta_i} = \frac{1}{2}\det(\mu)^{-\frac{1}{2}}\det(\mu)\mathrm{Tr}[\mu^{-1}\frac{\partial \mu}{\partial \theta_i}],$$

where $\mu = JJ^\top$ and

$$\frac{\partial \mu}{\partial \theta_i} = \frac{\partial J}{\partial \theta_i}J^\top + J\frac{\partial J}{\partial \theta_i}^\top.$$

2) $\frac{\partial \boldsymbol{\eta}_1}{\partial \boldsymbol{\theta}}$*:* Differentiating $\boldsymbol{\eta}_1$ with respect to $\boldsymbol{\theta}$, we get $\frac{\partial \boldsymbol{\eta}_1}{\partial \boldsymbol{\theta}} = J_{bt}^b \dot{\boldsymbol{\theta}}$. As we earlier analyzed in Sec. III-D, we know that $J_{bt}^b = J_{be}^b$. Therefore $\frac{\partial \boldsymbol{\eta}_1}{\partial \boldsymbol{\theta}}$ corresponds to the translational part of J.

3) $\frac{\partial \eta_2}{\partial \boldsymbol{\theta}}$*:* Using direct differentiation again, we get:

$$\frac{\partial \eta_2}{\partial \boldsymbol{\theta}} = -\boldsymbol{z}_{bc}^{b\,\top} s(\boldsymbol{z}_{bt}^b) J_\omega \dot{\boldsymbol{\theta}},$$

where J_ω denotes the rotational part of J.

4) $\frac{\partial \eta_3}{\partial \boldsymbol{\theta}}$*:* We calculate $\frac{\partial \eta_3}{\partial \boldsymbol{\theta}}$ by summing up the gradient $\frac{\partial \alpha}{\partial \boldsymbol{\theta}}$ and $\frac{\partial \beta}{\partial \boldsymbol{\theta}}$:

$$\frac{\partial \eta_3}{\partial \boldsymbol{\theta}} = w_\alpha \frac{\partial \alpha}{\partial \boldsymbol{\theta}} + w_\beta \frac{\partial \beta}{\partial \boldsymbol{\theta}},$$

where $\frac{\partial \alpha}{\partial \boldsymbol{\theta}}$ and $\frac{\partial \beta}{\partial \boldsymbol{\theta}}$ are obtained by differentiating (8) and (9) respectively:

$$\begin{cases} \frac{\partial \alpha}{\partial \theta_i} & = -\frac{1}{2}\mu_v^{-\frac{3}{2}}\frac{\partial \mu_v}{\partial \theta_i} \\ \frac{\partial \beta}{\partial \theta_i} & = -\frac{1}{2}\mu_\omega^{-\frac{3}{2}}\frac{\partial \mu_\omega}{\partial \theta_i} \end{cases}, \tag{12}$$

where

$$\mu_v = \boldsymbol{u}^\top (J_t J_t^\top)^{-1}\boldsymbol{u}, \quad \mu_\omega = \boldsymbol{u}^\top (J_\omega J_\omega^\top)^{-1}\boldsymbol{u}.$$

Using the fact that the force and moment transmission ellipsoid have reciprocal eigenvalues, we replace μ_v and μ_ω in (12) with the following:

$$\mu_v = -\boldsymbol{u}^\top (J_t J_t^\top)\boldsymbol{u}, \quad \mu_\omega = -\boldsymbol{u}^\top (J_\omega J_\omega^\top)\boldsymbol{u}.$$

and obtain:

$$\begin{cases} \frac{\partial \mu_v}{\partial \theta_i} = -\boldsymbol{u}^\top [\frac{\partial J_t}{\partial \theta_i}J_t^\top + J_t \frac{\partial J_t}{\partial \theta_i}^\top]\boldsymbol{u} \\ \frac{\partial \mu_\omega}{\partial \theta_i} = -\boldsymbol{u}^\top [\frac{\partial J_\omega}{\partial \theta_i}J_\omega^\top + J_\omega \frac{\partial J_\omega}{\partial \theta_i}^\top]\boldsymbol{u} \end{cases}.$$

B. Computable weights

Instead of manually tuning the weights w_α and w_β, we define w_β with respect to w_α according to the minimal principal angle between the range spaces $\mathcal{R}_{J_t}, \mathcal{R}_{J_\omega}$, which fulfills:

Definition 2: For nonzero subspaces $\mathcal{R}_1, \mathcal{R}_2 \subseteq \mathbb{R}^n$, the minimal principal angle between $\mathcal{R}_1$ and $\mathcal{R}_2$ is defined to be $0 \leq \phi \leq \pi/2$ and satisfies

$$\cos(\phi) = \max_{\boldsymbol{u},\boldsymbol{v}} \boldsymbol{u}^\top \boldsymbol{v}, \tag{13}$$

where $\boldsymbol{u} \in \mathcal{R}_1, \boldsymbol{v} \in \mathcal{R}_2$ and $\|\boldsymbol{u}\|_2 = \|\boldsymbol{v}\|_2 = 1$. Notice that $\phi = 0$ if and only if $\boldsymbol{u} = \boldsymbol{v}$, and $\phi = \pi/2$ if and only if $\mathcal{R}_1 \perp \mathcal{R}_2$. ■

While (13) defines ϕ, the calculation of ϕ relies on the orthogonal projectors $P_{\mathcal{R}_1}, P_{\mathcal{R}_2}$ as:

$$\cos(\phi) = \|P_{\mathcal{R}_1} P_{\mathcal{R}_2}\|_2 = \|P_{\mathcal{R}_2} P_{\mathcal{R}_1}\|_2, \tag{14}$$

where we can obtain the orthogonal projectors with the pseudo inverse: $P_{\mathcal{R}_J} = J^\dagger J$ and the matrix 2-norm corresponds to the biggest eigenvalue. Suppose w_α has priority over w_β, we calculate w_β as:

$$w_\beta = (1 - cos\phi) w_\alpha, \tag{15}$$

which defines w_β to be a fraction of w_α in proportion to ϕ. If $\mathcal{R}_{J_t} \perp \mathcal{R}_{J_\omega}$, we have $w_\beta = w_\alpha$ and if $\mathcal{R}_{J_t}$ and $\mathcal{R}_{J_\omega}$ share a common basis then we have $w_\beta = 0$.

C. Constraints integration

In order to solve Problem 1, we use the constraint based programming framework proposed in [5] to integrate η_0, $\boldsymbol{\eta}_1$, η_2 and η_3. Basically we formulate a quadratic programming(QP) problem to minimize $-\eta_3$ and $\|\dot{\boldsymbol{\theta}}\|_2$ while fulfilling the constraints that are based on η_0, $\boldsymbol{\eta}_1$, η_2. Concretely at each time step we formulate a QP as:

$$\begin{aligned} \min_{\dot{\boldsymbol{\theta}}, \nu_{i=0,1,2}^2} & \frac{\partial \eta_3}{\partial \boldsymbol{\theta}}^\top \dot{\boldsymbol{\theta}} + \dot{\boldsymbol{\theta}}^\top Q \dot{\boldsymbol{\theta}} + w_0 \nu_0^2 + w_1 \boldsymbol{\nu}_1^\top \boldsymbol{\nu}_1 + w_2 \nu_2^2, \\ \text{s.t.} & \frac{\partial \eta_0}{\partial \boldsymbol{\theta}}^\top \dot{\boldsymbol{\theta}} + \nu_0 \geq -k_0(\eta_0 - b_0), \\ & \frac{\partial \boldsymbol{\eta}_1}{\partial \boldsymbol{\theta}}^\top \dot{\boldsymbol{\theta}} + \boldsymbol{\nu}_1 = -\boldsymbol{k}_1(\boldsymbol{\eta}_1 - 0), \\ & \frac{\partial \eta_2}{\partial \boldsymbol{\theta}}^\top \dot{\boldsymbol{\theta}} + \nu_2 \geq -k_2(\eta_2 - b_2), \end{aligned}$$

where for $i = 0, 1, 2$, $\boldsymbol{k}_i$ sets the convergence rates, b_i denotes the bounds and the weight w_i provides us a way to weight η_0, $\boldsymbol{\eta}_1$ and η_2 with respect to each other and the other objectives.

The objective function has three aspects: (1) We always minimize the joint velocities measure $\dot{\boldsymbol{\theta}}^\top Q\dot{\boldsymbol{\theta}}$ as the robot is redundant and we want the minimum norm solution. We use the positive diagonal matrix Q to weight the joint velocities against each other and the other objectives. (2) The slack variables $\boldsymbol{\nu}_i$ fix the potential infeasibility induced by the corresponding constraints. (3) On top of the above two aspects, we optimize the weighted measure η_3.

VI. Proof of Concept Simulation

We verify the proposed solution through simulations using the PR2 simulator shown in Fig. 2. We solve the QP with *Gurobi* 6.02 at $200Hz$. We split the simulation into two parts: first we verify the analytical gradient $\frac{\partial J}{\partial \boldsymbol{\theta}}$ and show that maximizing η_0 does not result in the best transmission ratios along different task-dependent directions; then we use a Lissajous curve tracing task to show that the proposed method improves the curve-tracing performance.

A. Verification of the analytical gradients

In Fig. 4, we obtained two configurations by fixing the end-effector position $\boldsymbol{p}_t^b = constant$ and maximizing η_0. Using the analytical gradient we obtain a better measure: $\eta_0 = 0.0886037$ than 0.0558164, which is found using the numerical method with a fixed step size $\delta\theta = 0.01$. Then in Fig. 5, we show that along different task-dependent directions: $\boldsymbol{x}_{bc}^b$, $\boldsymbol{y}_{bc}^b$, $\boldsymbol{z}_{bc}^b$, we can obtain better transmission ratios using the proposed method, compared to optimizing η_0.

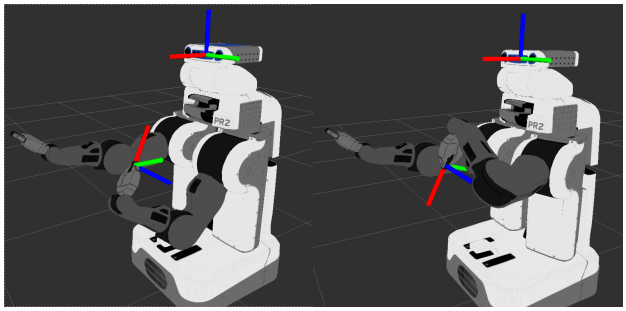

Fig. 4: We illustrate the benefit of the analytical gradient by maximizing the velocity ellipsoid volume η_0 while keeping $\boldsymbol{p}_t^b = constant$. **Left:** $max : \eta_0 = 0.0558164$ using numerical method with a fixed step size $\delta\theta = 0.01$. **Right:** $max : \eta_0 = 0.0886037$ using the analytical gradient.

B. Reactive trajectory tracing

As an example of Problem 1, we define a Lissajous curve tracing task. We choose the tool-tip position with respect to the end-effector frame $\mathcal{F}_e$ as:

$$g_{et} = ([0.05, 0.0, 0.0]^\top, R([0.0, 0.0, 1.0]^\top, \frac{\pi}{12}))^1$$

[1]We use the angle-axis representation of the rotation.

and use the *Lissajous* curve, which is plotted in Fig. 6, as the desired trajectory on the X-Y plane of the working-piece frame $\mathcal{F}_w$:

$$\boldsymbol{p}_c^w(t) = \begin{cases} x^w(t) & = a\cos(\omega_x t - \delta_x) \\ y^w(t) & = b\cos(\omega_y t - \delta_y) \\ z^w(t) & = 0 \end{cases}.$$

The parameters of the curve are selected as: $a = 0.1$, $b = 0.1$, $\omega_x = 0.03$, $\omega_y = 0.02$, $\delta_x = 0.7853$, $\delta_y = 0$. Using

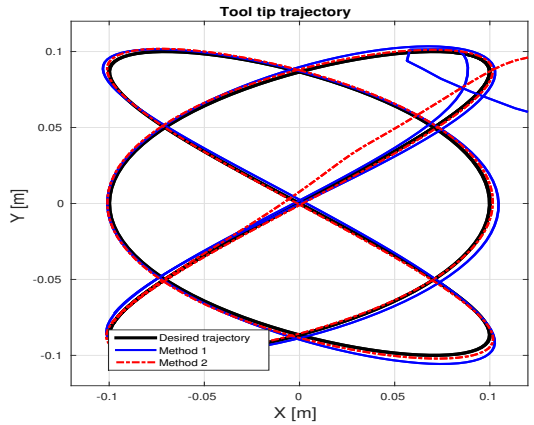

Fig. 6: The desired trajectory and the tool-tip trajectory. The primary trajectory tracing task is achieved in both cases.

the measure η_3 which is defined in (10), we compare the following three methods:
(0) Fulfilling the task constraints (5-7);
(1) Fulfilling (5-7) while minimizing η_3 with hand-chosen weights $w_\alpha = w_\beta = 200$;
(2) Fulfilling (5-7) while minimizing η_3 with $w_\alpha = 200$ and the computed w_β using (15).
In Fig. 7, we can find that in all the three cases the constraint (7) upon the orientation measure η_2 and (5) upon the velocity manipulability measure η_0 are fulfilled.

However when it comes to constraint (6), we can clearly see that by minimizing η_3, method 1 and 2 have a smaller translation error $\|\boldsymbol{\eta}_1\|_2$ than method 0. The performance of method 1 and 2 are comparable but in method 2 the weights are computed online using (15) which is an additional advantage.

According to the weight calculation (15), $\alpha_{\boldsymbol{x}_{bc}^b}$, and $\beta_{\boldsymbol{z}_{bc}^b}$ are balanced with respect to each other using $\cos(\phi) = \|P_{J_w} P_{J_t}\|_2$, which is plotted in the third row of Fig. 8. We can find that in case of method 2 we roughly have $w_\beta = 0.15 w_\alpha$, which results in a bigger $\alpha_{\boldsymbol{x}_{bc}^b}$ and a smaller $\beta_{\boldsymbol{z}_{bc}^b}$ than method 1 in the first and second row of Fig. 8.

VII. Conclusions and Future Work

We integrate the task-dependent measure with computable weights into the constraint based programming approach to perform reactive task-oriented redundancy resolution. We step by step verified the analytical gradient, and the improved trajectory tracing performance using the computable weights.

In the future, we plan to extend the proposed approach from two perspectives. By letting multiple manipulators to optimize the same task-dependent measure, we could

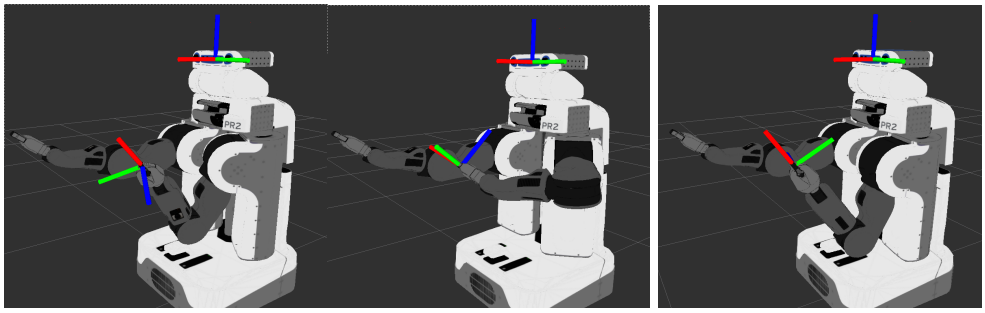

Fig. 5: **(a)**:max:$\alpha_{\boldsymbol{x}_{bc}^b}$ = 0.3225(0.2762) **(b)**: max:$\alpha_{\boldsymbol{z}_{bc}^b}$ = 0.5181(0.4736) **(c)**:max:$\beta_{\boldsymbol{z}_{bc}^b}$ = 1.8684(1.7033)

While keeping a constant distance between the left end-effector and the base frame: $\boldsymbol{p}_t^b = constant$, we optimize the joint configurations to obtain the above transmission ratios. As a comparison, we also list the corresponding values (in blue fonts) obtained by maximizing η_0 with the analytical gradient in the brackets.

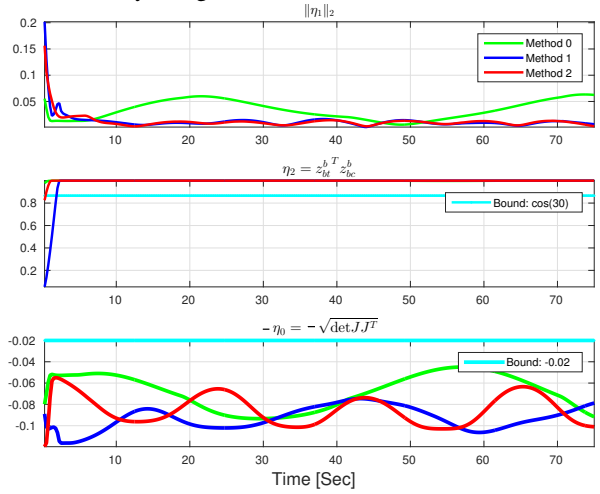

Fig. 7: In the second and third row, we can see that in all the three cases the constraints (5) and (7) are fulfilled. Compared to method 0, method 1 and 2 have a smaller error $\|\boldsymbol{\eta}_1\|_2$ by minimizing η_3.

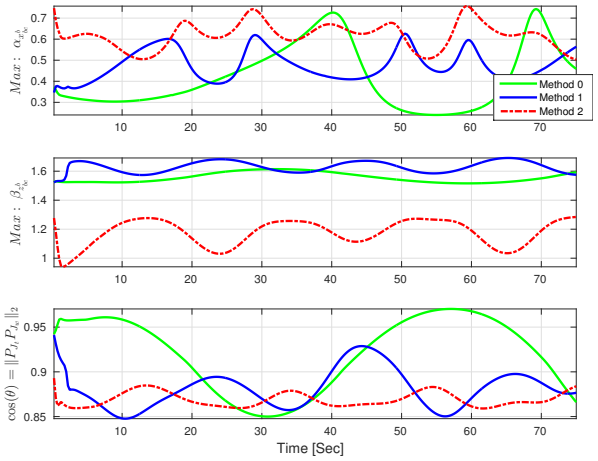

Fig. 8: In case of method 2, according to $\cos\phi$ shown in the third row, we get bigger $\alpha_{\boldsymbol{x}_{bc}^b}$ and smaller $\beta_{\boldsymbol{z}_{bc}^b}$ compared to method 1.

improve the performance of cooperative manipulation tasks using the proposed approach. In optimization based redundancy resolution frameworks, weights of different constraints are normally chosen manually. We could use the computable weights to automate the optimization problem formulation process.

REFERENCES

[1] H. Seraji, "An on-line approach to coordinated mobility and manipulation," in *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*. IEEE, 1993, pp. 28–35.

[2] S. L. Chiu, "Task compatibility of manipulator postures," *The International Journal of Robotics Research*, vol. 7, no. 5, pp. 13–21, 1988.

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

[4] W. Decré, R. Smits, H. Bruyninckx, and J. De Schutter, "Extending itasc to support inequality constraints and non-instantaneous task specification," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 964–971.

[5] 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.

[6] 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.

[7] 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.

[8] A. Müller, "Higher derivatives of the kinematic mapping and some applications," *Mechanism and Machine Theory*, vol. 76, pp. 70–85, 2014.

[9] Y. Nakamura, *Advanced robotics: redundancy and optimization*. Addison-Wesley Longman Publishing Co., Inc., 1990.

[10] I. C. Ipsen and C. D. Meyer, "The angle between complementary subspaces," *American Mathematical Monthly*, pp. 904–911, 1995.

[11] D. Oetomo and M. H. Ang Jr, "Singularity robust algorithm in serial manipulators," *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 1, pp. 122–134, 2009.

[12] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *Robotics and Automation, IEEE Transactions on*, vol. 13, no. 3, pp. 398–410, 1997.

[13] J. M. Schumacher, "A pointwise criterion for controller robustness," *Systems & control letters*, vol. 18, no. 1, pp. 1–8, 1992.

[14] H. Lipkin and J. Duffy, "Hybrid twist and wrench control for a robotic manipulator," *Journal of Mechanical Design*, vol. 110, no. 2, pp. 138–144, 1988.

[15] T. Yoshikawa, "Translational and rotational manipulability of robotic manipulators," in *Industrial Electronics, Control and Instrumentation, 1991. Proceedings. IECON'91., 1991 International Conference on*. IEEE, 1991, pp. 1170–1175.

[16] K. L. Doty, C. Melchiorri, E. M. Schwartz, and C. Bonivento, "Robot manipulability," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 3, pp. 462–468, 1995.

[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] 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.

[19] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, "A mathematical introduction to robotic manipulation," 1994.