Resolve reactive robot control with perturbed constraints using a second order cone programming approach

Yuquan Wang and Lihui Wang

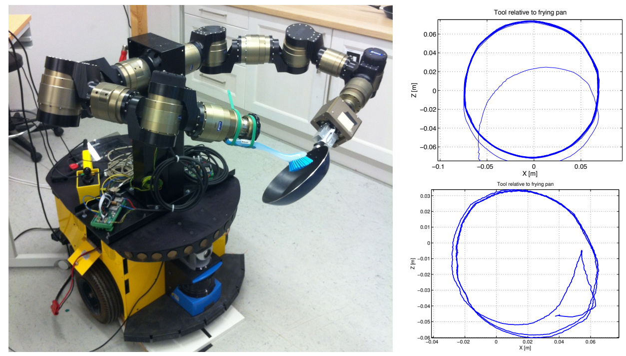
***Abstract*— As a modular and reactive control approach, constraint-based programming helps us to formulate and solve complex robotic tasks in a systematic way. In different fields ranging from industrial manipulators to humanoids, robots are supposed to work in an uncertain environment. However, how to address uncertainties is missing in the state-of-the-art of different constraint-based programming frameworks. In this paper, we introduce a Second Order Cone Programming (SOCP) approach to integrate constraints with norm bounded uncertainties. The proposed SOCP is convex and through simulations with controlled uncertainty level, we can clearly tell that the proposed approach guarantees the constraints satisfaction compared to the state-of-the-art.**

I. Introduction

From factory floors to research labs, we can find an increasing amount of robotic applications in different contexts. Different from the classical manufacturing tasks that are well-defined in a static environment, such as painting, welding and assembly, we envision a future in which robots are supposed to interact with an unpredictable and complex environment. For instance the tailor-made humanoids from Disney research are used in face-to-face entertainments with tourists and the collaborative robots from ABB and Universal robots are about to work side-by-side with human workers. These robotic applications are developed on a quick rolling basis which requires us to address control tasks using inaccurate kinematics and dynamics both from the robot side and the dynamic environment side.

Constraint-based programming is commonly used in robot task programming. It enables us to easily formulate modular and reactive online controllers for a wide range of robotic applications [1], [2], [3], [4]. We can find constraint-based programming problems that are formulated using null space projections in the earlier publications [1], [2], [3]. In recent years, optimization-based formulation received more attentions. Compared with null space projections, optimization-based formulations are able to integrate equalities and inequalities in a natural way [5], [6], resolve complex redundancies faster [4], switch smoothly from one set of constraints to another [7]. Due to its strength in formulating control problems, We could also find its applications even outside robotics, e.g. as a multi-character animation controller [8].

However none of the state-of-the-art optimization-based formulations [4] -[8] explicitly make use of the uncertainty information to integrate constraints and solve for the robot motion online. The level of uncertainties makes a difference in performance between simulated and real robots [9]. We can find such an example in Fig. 1, where the uncertain geometric model of kitchenware limited the performance of a dual-arm robot. There is a rich source of the uncertainties. From the sensing side, this includes vision measurement, friction coefficients and imperfect geometric models. From the modelling side, this include inaccurate robot kinematics, dynamics as well as properties of different emerging objects in a dynamic environment.



(a) Hardware setup includes tools with imperfect geometric models. **(b)** Brush tip trajectory relative to the frying pan

Fig. 1: We use a bi-manual pan-cleaning task to illustrate the fact that if the geometry is not perfectly modeled, we end up with undesired motion using the state-of-the-art QP approach.
In this task, the brush tip is supposed to follow a circular motion that is defined on the frying pan surface. If we do not require the contact between the brush and the frying pan, we obtained a circular relative motion, see the top right sub-figure.
When we specify the contract force to be 8N, the relative motion is far from an exact circular motion due to the imperfect geometric model, see the bottom right sub-figure.

Assuming that the uncertainties could be upper bounded, we propose to integrate the constraints and objectives with Second Order Cone Programming (SOCP), which is a class of convex optimization [10]. Taking the upper bounds of uncertain parameters into account, the proposed approach explicitly minimize and constrain the supremum of the objectives and inequalities respectively. In case of the equalities, the proposed approach minimize the worst case error, which is in a way similar to the robust approximation problem [11]. We illustrate the proposed approach through an detailed example, where the SOCP formulation is explicitly given for a bi-manual pan cleaning task. In the simulation with controlled uncertainty level, we are able to guarantee the constraints satisfaction compared to the conventional Quadratic

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

Programming (QP) formulation.

The rest of the paper is organized as follows: we relate the proposed solution to the state-of-the-art in Sec. II; then in Sec. III we mathematically states how the uncertainties propagates into constraint-based programming through task Jacobians; the proposed solution is verified by simulation in Sec. V and we conclude the paper in Sec. VI.

II. Related Work

In this section, we briefly discuss our contribution in different contexts including: constraint-based programming, control of uncertain systems and optimization in general.

Constraint-based programming effectively resolves the redundancy of a robot with a combination of objectives, equalities and inequalities. We can find its applications in visual servoing [12], dual-arm manipulation [6] and whole-body control of humanoids [3]. There is a systematic way of specifying constraints in the seminal paper [2], where we can also specify virtual joints to model the geometry of a robotic task. Optimization-based formulation allows us to easily scalarize of multiple objectives and choose from different norms. LP is able to integrate equality and inequality constraints, where the convergence of each constraint could be proved using a Lyapunov function [13], [14]. If it is needed to penalize the overuse of joint velocities and locate a smooth solution over the iterations, we need to escalate it to QP in order to apply the Euclidean norm [6]. In this paper we further escalate the optimization framework to a SOCP problem such that we can optimize over the supremum or in other words the worst case error.

Conventionally there are two general strategies to control uncertain systems: learning-based control and robust control. For systems working in a repetitive mode, we can use iterative learning approach to locate the uncertain parameters [15]. On the other hand, robust control is used when the controller has a fixed structure. In the context of constraint-based programming, though we can use reinforcement learning to find the uncertain weights of different constraints [16], how to make use of the norm bounded uncertainties to integrate constraints is missing. We can use the proposed approach to fill this gap.

When we need to optimize under uncertainties, there are different objectives that we choose from: expectation minimization, minimization of maximum costs and optimization over soft constraints, see [11]. In case of multi-stage optimization, e.g. resource allocation problems, we can use stochastic programming to minimize the expected costs. If the infeasibility of certain constraints are allowed, e.g. soft constraints, we can use probabilistic programming to optimize the chance of constraints satisfaction. These two strategies are not suitable in the context of constraint-based programming, as we need to solve hard constraints in a reactive(online) way. Assuming that we can upper bound the uncertainties, e.g. vision errors and unmodeled kinematics, we choose to minimize of the worst case error for equalities and supremum for inequalities as well as objectives.

III. Problem Formulation

By mathematically raising an abstract optimization problem, which is Problem 1, we abstract the generic structure of the state-of-the-art optimization-based frameworks. On top of Problem 1, we summarize the case when the constraints are perturbed with uncertainties in Problem 2.

Problem 1: Assuming that the kinematics $J\dot{\boldsymbol{q}} = \boldsymbol{v}$ and dynamics $\boldsymbol{\tau} = J^\top \boldsymbol{f}$ of an n degrees of freedom robot are available, where $\dot{\boldsymbol{q}}, \boldsymbol{\tau} \in \mathbb{R}^n$ denote the joint velocities and torques, $\boldsymbol{v}, \boldsymbol{F} \in \mathbb{R}^n$ denote the end-effector velocities and external forces. We can formulate the online (locally optimal) controller in terms of the gradient of a set of linear equality constraints $f_i : \mathbb{R}^n \to \mathbb{R}$ for all $i \in I_e$, inequality constraints $f_i : \mathbb{R}^n \to \mathbb{R}$ for all $i \in I_{ie}$ and objectives: $f_i : \mathbb{R}^n \to \mathbb{R}$ for all $i \in I_o$ as:

$$\min_{\boldsymbol{u}} \quad \boldsymbol{u}^\top Q \boldsymbol{u} + \sum_{i \in I_o} w_i \dot{f}_i(\boldsymbol{x}, \boldsymbol{u}) \tag{1}$$

$$\text{s.t.} \quad \dot{f}_i(\boldsymbol{x}, \boldsymbol{u}) \leq -k_i(f_i(\boldsymbol{x}) - b_i), \ \forall i \in I_{ie}, \tag{2}$$

$$\dot{f}_i(\boldsymbol{x}, \boldsymbol{u}) = -k_i(f_i(\boldsymbol{x}) - b_i), \ \forall i \in I_e, \tag{3}$$

where k_i, b_i are positive gains and bounds associated with each constraint, w_i helps us to scalarize and compromise among multiple objectives, Q is a positive definite matrix. In case of a joint velocity controlled robot, the Cartesian space variables $\boldsymbol{x}$ and robot control variables $\boldsymbol{u}$ are chosen as $\boldsymbol{v}$ and $\dot{\boldsymbol{q}}$. In case of a joint torque controlled robot, we use $\boldsymbol{f}$ and $\boldsymbol{\tau}$ alternatively. ■

Problem 1 captures the essence of different optimization-based reactive control frameworks [5], [13], [14], [4]. The parameters associated with Problem 1 helps us to define different aspects of robot control problems in a modular way. We can prioritize different objectives with different combinations of w_i [5], [13]; We can set different convergence rates and bounds using k_i, b_i [14]; and we can even swap the constraints and objectives based on the feasibility of each constraint [4]. In Lemma 1, we summarize a generic QP-based online controller (4-6) to Problem 1.

Lemma 1: As stated by the state of the art, see [14], we can obtain the solution to Problem 1 by solving:

$$\min_u \quad \boldsymbol{u}^\top Q \boldsymbol{u} + \textstyle\sum_{i \in I_o} w_i c_i^\top \boldsymbol{u} \tag{4}$$

$$\text{s.t.} \quad A_i \boldsymbol{u} \leq k_i(b_i - f_i) \ \forall i \in I_{ie}, \tag{5}$$

$$A_i \boldsymbol{u} = k_i(b_i - f_i) \ \forall i \in I_e, \tag{6}$$

where $c_i = \frac{df_i}{d\boldsymbol{u}}$, and each row of A_i, b_i, f_i contains the corresponding parts of $\frac{df_i}{d\boldsymbol{u}}, b_i, f_i$ respectively. In the robotics context, $\frac{df_i}{d\boldsymbol{u}}$ defines a task Jacobian, see [1], [17], either on a kinematics level or a dynamics level. ■

On the other hand what is missing from Problem 1, so does the state-of-the-art in constraint-based programming, is to optimize robot motion with uncertain data. Basically Problem 1 assumes that we can perfectly model the environment and the robots. This is unfortunately not always the case, as there are a complete set of papers in adaptive control and nonlinear control that aim to work well with uncertain systems, see [17].

Starting from the assumptions and conditions that are stated in Problem 1, we relax the assumption that the constraints and objectives are perfectly known. Basically we use the uncertain parameter $\boldsymbol{\xi}$ to formulate a generic optimization problem with uncertainties as the following:

Problem 2: When the robots are not perfectly modeled and/or the robotic tasks are not well defined, the gradients of the constraints and objectives are perturbed. In case as such, we can extend Problem 1 as the following:

$$\min_{\boldsymbol{u}} \quad \boldsymbol{u}^\top Q\boldsymbol{u} + \textstyle\sum_{i\in I_o} w_i \dot{f}_i(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi}_i) \tag{7}$$

$$\text{s.t.} \quad \dot{f}_i(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi}_i) \leq -k_i(f_i(\boldsymbol{x}) - b_i),\ \forall i \in I_{ie}, \tag{8}$$

$$\dot{f}_i(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi}_i) = -k_i(f_i(\boldsymbol{x}) - b_i),\ \forall i \in I_{e}, \tag{9}$$

where we use $\boldsymbol{\xi}$ to parameterize the uncertainties associated with the constraints and objectives. ■

In constraint based programming literatures [1], [17], the gradients $\dot{f}_i$ are referred as task Jacobians. We can use $\boldsymbol{\xi}$ to describe the perturbed task Jacobians in different cases, e.g. visual servoing, impedance control, trajectory following. However in Problem 2 we do not parameterize the uncertainties on the right hand side of the constraints (8-9) due to the fact that f_i defined in the Cartesian space does not relate to the optimization variables $\boldsymbol{u}$ defined in the robot configuration space.

In the next section. we draw a solution to Problem 2 based on knowledge from robust optimization.

IV. Proposed Solution

When the task Jacobians are suffering from uncertainties, which is the case described in Problem 2, we use the equivalence that is shown in Lemma 2 to reform the objectives (4), inequality constraints (5), equality constraints (6) respectively and in the end we can obtain the solution (16) to Problem 2.

Lemma 2: We assume that the uncertainties associated with matrix $\mathcal{A} \in \mathbb{R}^{m\times n}$ is restricted within a norm ball, $\mathcal{A} = \{\bar{A} + \xi \mid \|\xi\| \leq a\}$, where the norm $\|\cdot\|$ is compatible to $\mathbb{R}^{m\times n}$. For the equality constraint $\mathcal{A}x = b$, We can define the worst case approximation error as:

$$\boldsymbol{e}_{wc} = \sup_{\xi}\{\|\bar{A}x - b + \xi x\| \mid \|\xi\| \leq a\},$$

which is upper bounded by:

$$\boldsymbol{e}_{wc} = \|\bar{A}x - b\| + a\|x\|. \tag{10}$$

If we choose to use the Euclidean norm and minimize the worst case approaximation error $\boldsymbol{e}_{wc}$, it is equivalent to minimize the following second order cone programming (SOCP) problem, see [10]:

$$\begin{aligned} \min_{\boldsymbol{u}} \quad & t_1 + at_2, \\ \text{s.t.} \quad & \|\bar{A}x - b\|_2 \leq t_1, \\ & \|x\|_2 \leq t_2. \end{aligned} \tag{11}$$

■

a) Objective function: If we need to minimize an objective function, we could alternatively minimize its upper bound. In case of an objective function with uncertainty (7) we can minimize the supremum of the objective according to the problem-solution pair (1) and (4):

$$\boldsymbol{u}^\top Q\boldsymbol{u} + \sum_{i\in I_o} w_i \sup_{\boldsymbol{\xi}}\{c_i(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi})^\top \boldsymbol{u}\}. \tag{12}$$

Assuming that $c_i(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi}) = \{\bar{c}_i(\boldsymbol{x},\boldsymbol{u}) + \xi_{c_i} \mid \|\xi_{c_i}\| \leq a_{c_i}\}$, we can use Lemma 2 to obtain the fact that minimizing the upper bound of (12) is equivalent to solving the following SOCP:

$$\boldsymbol{u}^\top Q\boldsymbol{u} + \sum_{i\in I_o} w_i\big(\bar{c}_i(\boldsymbol{x},\boldsymbol{u})^\top \boldsymbol{u} + a_{c_i}\|\boldsymbol{u}\|_2\big). \tag{13}$$

b) Inequality constraint: In view of an inequality constraint-gradient pair (2) and (5), fulfilling the constraint with uncertainty (8) implies fulfilling the following:

$$A_i(\boldsymbol{\xi}_i)\boldsymbol{u} \leq k_i(b_i - f_i)\ \forall i \in I_{ie}.$$

Assuming that $A_i(\boldsymbol{\xi}_i) = \{\bar{A}_i + \xi_{A_i} \mid \|\xi_{A_i}\| \leq a_{A_i}\}$ for all $i \in I_{ie}$, we can reformulate the above inequality based on Lemma 2 and obtain the following constraint:

$$\begin{aligned} & A_i(\boldsymbol{\xi}_i)\boldsymbol{u} \leq \\ & \sup\{A_i(\boldsymbol{\xi}_i)\boldsymbol{u}\} \leq \\ & \bar{A}_i^\top \boldsymbol{u} + a_{A_i}\|\boldsymbol{u}\|_2 \leq k_i(b_i - f_i), \forall i \in I_{ie}. \end{aligned} \tag{14}$$

c) Equality constraint: Fulfilling an equality constraint perturbed with uncertainty amounts to a robust approximation problem that is stated in Lemma 2. In case of the constraint (9), we assume that $A_i(\boldsymbol{\xi}_i) = \{\bar{A}_i + \xi_{A_i} \mid \|\xi_{A_i}\| \leq a_{A_i}\}$ for all $i \in I_e$ and we minimize the worst case error of the perturbed equality constraint by solving the following SOCP:

$$\begin{aligned} \min_{\boldsymbol{u},t_{1i},t_2} \quad & \sum_{i\in I_e}(t_{1i} + a_{A_i}t_2), \\ \text{s.t.} \quad & \|\bar{A}_i^\top \boldsymbol{u} - k_i(b_i - f_i)\|_2 \leq t_{1i},\ \forall i \in I_e, \\ & \|\boldsymbol{u}\|_2 \leq t_2. \end{aligned} \tag{15}$$

Integrating (13-15) together, we can conclude that solving the SOCP problem (16) gives us the online/local solution to (7-9) that are defined in Problem 2:

$$\begin{aligned} \min_{\boldsymbol{u},t_{1i},t_2} \quad & \sum_{i\in I_o} w_i\big(\bar{c}_i(\boldsymbol{x},\boldsymbol{u})^\top \boldsymbol{u} + a_{c_i}\|\boldsymbol{u}\|_2\big) + \sum_{i\in I_e}\big(t_{1i} + a_{A_i}t_2\big), \\ \text{s.t.} \quad & \bar{A}_i^\top \boldsymbol{u} + a_{A_i}\|\boldsymbol{u}\|_2 \leq k_i(b_i - f_i), \forall i \in I_{ie}, \\ & \|\bar{A}_i^\top \boldsymbol{u} - k_i(b_i - f_i)\|_2 \leq t_{1i},\ \forall i \in I_e, \\ & \|\boldsymbol{u}\|_2 \leq t_2. \end{aligned} \tag{16}$$

Note that we removed the quadratic term $\boldsymbol{u}^\top Q\boldsymbol{u}$ from the objective (13), otherwise it overlaps with the inequality $\|\boldsymbol{u}\|_2 \leq t_2$ that is induced by the (15). In the next section, we use a dual-arm manipulation task as an example to validate the proposed solution (16).

V. Simulation Verification

Different from the example in Fig. 1, where the experiments were performed on hardware, we choose to use simulations to verify the proposed solution (16). The reason is that we can perfectly control the level of uncertainties to be introduced in the task Jacobians, which makes it straight forward for us to setup a comparison.

We choose the bi-manual pan cleaning task as a non-trivial example of constraint-based programming to validate the proposed formulation (16). We choose two Puma560 6 DoF manipulators that are simulated by the Matlab Robotics Toolbox by [18], see Fig. 2. We choose the *cvx* toolbox [19] to solve the SOCP problem. We first specify the constraints for the pan cleaning task in Sec. V-A and then we introduce the simulation setup in Sec. V-B. When the constraints are corrupted with uncertainties, we compare the results with and without the proposed solution in Sec. V-C, where we can see that the proposed solution improve the controller performance in terms of constraints satisfaction.

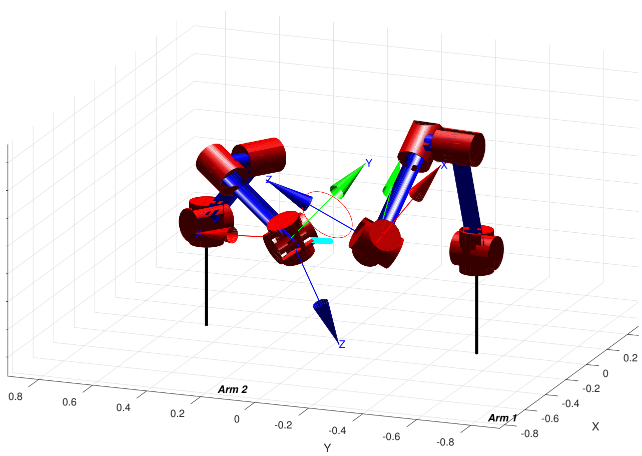


Fig. 2: Two Puma560 6DoF manipulators simulated by the Matlab Robotics Toolbox [18].

A. Controller formulation

We can use constraints to easily specify complex robotic tasks. In this paper we use a bi-manual task as an example. We choose to work with the kinematics, which means that we choose the joint velocity $\dot{\boldsymbol{q}}$ as $\boldsymbol{u}$. For each constraint, we also define the corresponding bound b_i and the convergence rate k_i that are in line with (4-6). In (20), we list the specific QP formulation corresponding to the state of the art (4-6). Then in (21), we list the specific SOCP formulation in line with the proposed solution (16).

We restrict the relative position between the two arms with three equalities and we restrict the relative orientation within a cone:

$$\begin{aligned} f_{1-3}(\boldsymbol{q}) &= p_1(\boldsymbol{q}) - p_2(\boldsymbol{q}) - d(t, \boldsymbol{x}_1, \boldsymbol{y}_1, \boldsymbol{z}_1, \boldsymbol{x}_2) = 0 \\ f_4(\boldsymbol{q}) &= x_1^T x_2 \le b_4, \end{aligned} \tag{17}$$

where axis $\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{z}_i \in \mathbb{R}^3$ are columns of $R_i \in SO(3)$, $\boldsymbol{p}_1 \in \mathbb{R}^3$ is the center of the frying pan and $\boldsymbol{p}_2 \in \mathbb{R}^3$ corresponds to the tip position of the cleaning utensil, the time varying $d(t, \boldsymbol{x}_1, \boldsymbol{y}_1, \boldsymbol{z}_1, \boldsymbol{x}_2)$ is an relative offset between $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$. In this pan cleaning task, we define the relative offset $d(t, \boldsymbol{x}_1, \boldsymbol{y}_1, \boldsymbol{z}_1, \boldsymbol{x}_2)$ as a circular motion. With straight forward calculation, we can find $A_{1-3} = [J_{p_1} + \frac{\partial d}{\partial \boldsymbol{q}_1}, -J_{p_2} + \frac{\partial d}{\partial \boldsymbol{q}_1}]$, $A_4 = [\boldsymbol{x}_2^T(-S(\boldsymbol{x}_1)J_{\omega 1}), \boldsymbol{x}_1^T(-S(\boldsymbol{x}_2)J_{\omega 2})]$, where J_{pi} and $J_{\omega i}$ correspond to the translational and rotational part of the manipulator Jacobian.

In all the simulations, we choose the same set of parameters. For simplicity, we choose the convergence gain $k_i = 1$ for $i = 1, \ldots, 5$. As we need $f_{1-3} = 0$, we set the bound $b_{1-3} = 0$. The bound of the tool orientation constraint is selected as $b_4 = -\cos(\frac{\pi}{12})$, which defines a cone for the relative orientation between the two arms.

As we need the robot to work in a collision-free manner, we can use inequalities to constrain the minimal distances between a robot and the obstacles as:

$$f_5(\boldsymbol{q}) = -||x_r - x_o||_2 \le b_5 < 0, \tag{18}$$

where X_r is the subset of the workspace occupied by the robot itself, and X_o is the subset of the workspace occupied by obstacles. Depending on the computational power and required performance, we could either apply simple conservative obstacle representations, such as spheres, or more elaborate computations of the minimal distance, e.g. using the critical points and directions as described by [20].

In this example, we define a virtual plane that $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ have to avoid with a certain bound. We select $x = -0.4m$ in the global frame as the virtual wall and $b_5 = -0.1$.

In order to efficiently generate velocities and forces, manipulators should stay away from singular configurations. We need to optimize a manipulability index, see [17]:

$$f_o(\boldsymbol{q}) = \frac{-1}{2}\det(J_i^T J_i), \quad i \in \{1, 2\} \tag{19}$$

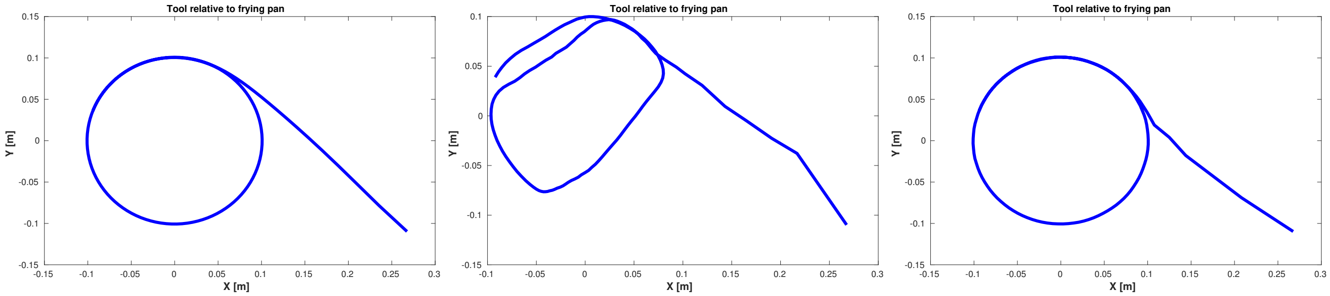
where $J_i \triangleq \begin{bmatrix} J_{pi}^T & J_{\omega i}^T \end{bmatrix}^T$ denotes the manipulator Jacobians. We choose to numerically calculate $A_5 = \frac{\partial f_5}{\partial \boldsymbol{q}}$ and $c_o = \frac{\partial f_o}{\partial \boldsymbol{q}}$.

As the manipulability measure (19) depends on the particular robot kinematics, its value range is different for different robots. This makes it empirical to define a proper bound for (19). Therefore we use (19) as an objective function such that we keep on optimizing the robot configuration while fulfilling the other constraints.

Summarizing the constraints and parameters that we specified so far, we can formulate a QP controller corresponding to the state-of-the-art (4-6) as follows:

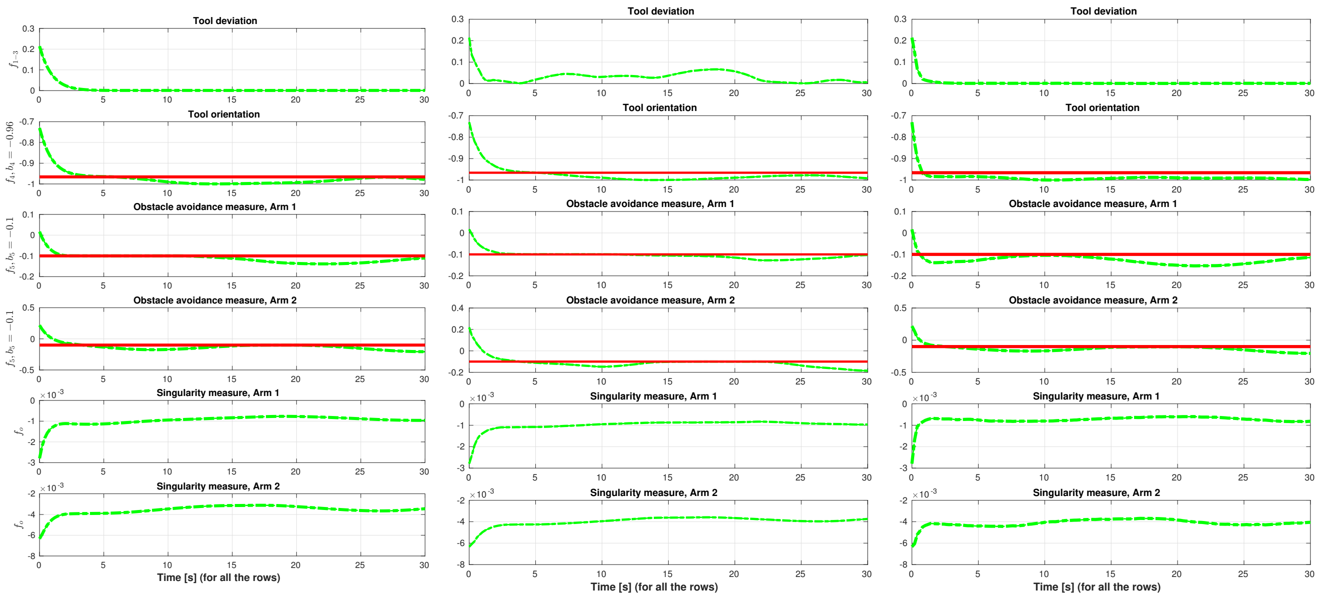
$$\begin{aligned} \min_{\dot{\boldsymbol{q}}} \quad & \dot{\boldsymbol{q}}^\top \dot{\boldsymbol{q}} + c_o^\top \dot{\boldsymbol{q}} \\ \text{s.t.} \quad & A_{1-3}\dot{\boldsymbol{q}} = (b_{1-3} - f_{1-3}) \\ & A_4 \dot{\boldsymbol{q}} \le (b_4 - f_4), \\ & A_5 \dot{\boldsymbol{q}} \le (b_5 - f_5). \end{aligned} \tag{20}$$

On top of (20), we can obtain the following SOCP



(a) QP formulation without uncertainties **(b)** QP formulation with uncertainties **(c)** Proposed formulation with uncertainties

Fig. 3: We compare the brush tip trajectories with respect to the frying pan in different simulations.



(a) QP formulation without uncertainties **(b)** QP formulation with uncertainties **(c)** Proposed formulation with uncertainties

Fig. 4: Constraints and objective measures obtained from different rounds of simulations.

controller corresponding to (16):

$$\begin{aligned} \min_{\dot{\boldsymbol{q}},t_1,t_2} \ & \left(c_o^\top \dot{\boldsymbol{q}} + a_{c_o}\|\dot{\boldsymbol{q}}\|_2\right) + \left(t_1 + a_{A_{1-3}} t_2\right), \\ \text{s.t.} \ & \|A_{1-3}^\top \dot{\boldsymbol{q}} - (b_{1-3} - f_{1-3})\|_2 \leq t_1, \\ & \|\dot{\boldsymbol{q}}\|_2 \leq t_2, \\ & A_4^\top \dot{\boldsymbol{q}} + a_{A_4}\|\dot{\boldsymbol{q}}\|_2 \leq (b_4 - f_4), \\ & A_5^\top \dot{\boldsymbol{q}} + a_{A_5}\|\dot{\boldsymbol{q}}\|_2 \leq (b_5 - f_5). \end{aligned} \tag{21}$$

Remark 5.1: Note that when the uncertainties are added to the equality constraint f_{1-3}, the following formulation

$$A_{1-3}(\boldsymbol{\xi}_{1-3})\dot{\boldsymbol{q}} = (b_{1-3} - f_{1-3}) \tag{22}$$

would make the QP (20) infeasible. In order to perform the comparison using the QP (20), we choose to minimize the worst case error of (22) by replacing (22) in (20) with the following:

$$\begin{aligned} \min_{\dot{\boldsymbol{q}},t_i} \ & \sum_{i=1,2,3} t_i^2 \\ \text{s.t.} \ & A_{1-3}(\boldsymbol{\xi}_{1-3})\dot{\boldsymbol{q}} - (b_{1-3} - f_{1-3}) \leq t_{1-3} \\ & A_{1-3}(\boldsymbol{\xi}_{1-3})\dot{\boldsymbol{q}} - (b_{1-3} - f_{1-3}) \geq -t_{1-3} \end{aligned} \tag{23}$$

The formulation in (23) uses a QP to minimize the one norm, we can find the correspondence between (23) and (22) from [10]. ■

B. Noise-free simulation

We start with the noise free simulation to verify the feasibility of the state of the art QP formulation (20) such that we have the benchmark to compare with in the next section. In Fig. 3a and Fig. 4a, we can find that the QP formulation (20) is able to generate the desired relative circular motion while satisfying all the constraints with respect to the prespecified bounds.

In the first row of Fig. 4a, in order to concretely display the deviation between the two arms, we plot $\|\boldsymbol{p}_1 - \boldsymbol{p}_2 - \boldsymbol{d}\|_2$ and we can see that in the noise-free case, $\|\boldsymbol{p}_1 - \boldsymbol{p}_2 - \boldsymbol{d}\|_2$ converges to zero quickly and stays stable.

In the second row of Fig. 4a, the tool orientation f_4 starts above the bounds but then it quickly approaches the bound and stays below in the rest of the execution. Similar behaviors of the obstacle avoidance constraints f_5 of the two arms could be found in the third and forth rows of Fig. 4a.

In the last two rows in Fig. 4a, we we plot the singularity measures f_o of the two arms. As we use f_o as the objective function, we optimize f_o if and only if the other constraints are satisfied.

C. Comparison

We perturb the constraints that are used in (20) with norm-bounded uncertainties. Basically we perturb the singular values of A_{1-5} with simulated uncertainties that are drawn from different uniform distributions. For the equality constraint f_{1-3} which define the relative offset between the two arms, we choose $\|\xi_{A_{1-3}}\| \leq a_{A_{1-3}} = 0.05$; for the inequality constraint f_4 which specifies the relative orientation between the two arms, we choose $\|\xi_{A_4}\| \leq a_{A_4} = 0.2$; for the inequality constraint f_5 which constrain the arms from obstacles, we choose $\|\xi_{A_5}\| \leq a_{A_4} = 0.1$.

From Fig. 3b and Fig. 4b, we can clearly tell that the QP formulation (20) is not able to generate desired robot motion when the task Jacobians of the constraints are corrupted.

The relative motion between the two arms is not circular any more as shown in Fig. 3b. From the first row of Fig. 4b, we can see that the deviation between the two arms could not converge effectively as compared to the first row of Fig. 4a.

Using the same perturbed constraints with the proposed SOCP formulation (21), we can find a different performance from Fig. 3c and Fig. 4c, which validates that the proposed SOCP works well with perturbed constraints.

The relative circular motion shown in Fig. 3c is comparable to Fig. 3a, which could be explained by the fast convergence of the tool deviation that is shown in the first row of Fig. 4c.

As we use the supremum in the inequality formulation (14), we can find that the inequality constraints, i.e. tool orientation f_4 and obstacle avoidance f_5, have a faster convergence from the second row to the forth row in Fig. 4c compared to the counter parts in Fig. 4a and Fig. 4b.

From the last two rows in Fig. 4a, 4b and 4c, we plot the singularity measures f_o of the two arms. In all the cases, the singularity measures have a similar profile, the reason is that f_o is the least prioritized as an objective function, see [4]. Given the constraints f_{1-5}, the robot does not have enough redundancy to make a difference of f_o.

VI. CONCLUSION

We propose to integrate constraints and resolve robot motion with a SOCP framework. Compared to the state-of-the-art QP formulations, this approach could minimize the supremum of the worst case error when the constraints are suffered from norm bounded uncertainties. We use a dual-arm pan cleaning task as a comparison example, where we perturb the constraints with simulated uncertainties. We can see that the proposed SOCP framework is able to fulfill the constraints meanwhile the state-of-the-art QP framework is not.

In the future, we can extend the proposed SOCP formulation by establishing the connection between the bounds on general task Jacobian uncertainties and the bounds on specific parameter uncertainties. Once this connection is established, the proposed SOCP is able to directly optimize the robot motion with respect to a specific uncertain parameter, such as a friction coefficient, a pose estimated from vision and the dynamics of an unknown object. There is a systematic framework of constraint specification [2], however concluding the aforementioned connection for all kinds of task Jacobians is not an easy task.

REFERENCES

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

[2] 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.

[3] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *Robotics, IEEE Transactions on*, vol. 25, no. 3, pp. 670–685, 2009.

[4] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014.

[5] W. Decré, R. Smits, H. Bruyninckx, and J. De Schutter, "Extending itasc to support inequality constraints and non-instantaneous task specification," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 964–971.

[6] 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.

[7] H. Audren, A. Kheddar, and P. Gergondet, "Stability polygons reshaping and morphing for smooth multi-contact transitions and force control of humanoid robots," in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*. IEEE, 2016, pp. 1037–1044.

[8] J. Vaillant, K. Bouyarmane, and A. Kheddar, "Multi-character physical and behavioral interactions controller," *IEEE transactions on visualization and computer graphics*, 2016.

[9] S. Lengagne, J. Vaillant, E. Yoshida, and A. Kheddar, "Generation of whole-body optimal dynamic multi-contact motions," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1104–1119, 2013.

[10] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[11] N. V. Sahinidis, "Optimization under uncertainty: state-of-the-art and opportunities," *Computers & Chemical Engineering*, vol. 28, no. 6, pp. 971–983, 2004.

[12] N. Mansard and F. Chaumette, "Task sequencing for high-level sensor-based control," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 60–72, 2007.

[13] O. Kanoun, F. Lamiraux, P.-B. Wieber, F. Kanehiro, E. Yoshida, and J.-P. Laumond, "Prioritizing linear equality and inequality systems: application to local motion planning for redundant robots," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 2939–2944.

[14] 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.

[15] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Systems*, vol. 26, no. 3, pp. 96–114, 2006.

[16] V. Modugno, G. Neumann, E. Rueckert, G. Oriolo, J. Peters, and S. Ivaldi, "Learning soft task priorities for control of redundant robots," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 221–226.

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

[18] P. I. Corke, *Robotics, Vision & Control: Fundamental Algorithms in Matlab*. Springer, 2011, iSBN 978-3-642-20143-1.

[19] M. Grant, S. Boyd, and Y. Ye, "Cvx: Matlab software for disciplined convex programming," 2008.

[20] R. Patel, F. Shadpey, F. Ranjbaran, and J. Angeles, "A collision-avoidance scheme for redundant manipulators: Theory and experiments," *Journal of Robotic Systems*, vol. 22, no. 12, pp. 737–757, 2005.