Dual Arm Manipulation using Constraint Based Programming ⋆

Yuquan Wang, Francisco Viña, Yiannis Karayiannidis, Christian Smith, Petter Ögren

All authors are with the Center for Autonomous Systems (CAS), at the Royal Institute of Technology (KTH), SE-100 44, Stockholm, Sweden. Email of corresponding author: `yuquan at kth.se`.

Abstract:
In this paper, we present a technique for online generation of dual arm trajectories using constraint based programming based on bound margins. Using this formulation, we take both equality and inequality constraints into account, in a way that incorporates both feedback and feedforward terms, enabling e.g. tracking of timed trajectories in a new way. The technique is applied to a dual arm manipulator performing a bi-manual task. We present experimental validation of the approach, including comparisons between simulations and real experiments of a complex bimanual tracking task. We also show how to add force feedback to the framework, to account for modeling errors in the systems. We compare the results with and without feedback, and show how the resulting trajectory is modified to achieve the prescribed interaction forces.

1. INTRODUCTION

For robotics to move from the factory floors to unstructured domestic environments, progress is needed in several areas of robotic technology. One such area is dual arm manipulation, where the human-like structure of a robot, such as the one in Figure 1, is exploited to perform tasks in environments originally intended for humans, as explained by Smith et al. [2012].

The potential benefits of endowing robots with dual arms fall into four main categories. First, using tools and workflows designed for humans is easier if the kinematic structure of the robot is similar to a human according to Kemp et al. [2007], Fuchs et al. [2009], Bloss [2010], Kr uger et al. [2011]. Second, teleoperation is easier if the robot is similar to the operator [Jau, 1988, Yoon et al., 1999, Kron and Schmidt, 2004, Buss et al., 2006, Taylor and Seward, 2010]. Third, the use of the two arms can either provide additional strength and precision by cooperating as a parallel manipulator, or provide flexibility and speed by doing two separate tasks simultaneously as discussed by Lee and Kim [1991]. Fourth, the two arms are able to perform task that are inherently bi-manual as reported by Chiacchio and Chiaverini [1998], Caccavale et al. [2000], i.e., tasks that require motion of both arms to be carried out efficiently.

In this paper we put the focus on such bi-manual tasks, which often include significant redundancies, a fact that makes them well suited to constraint based programming approaches. Examples include the following:

- Handing over a mug with one end-effector to a human and picking up a box of tea with another.

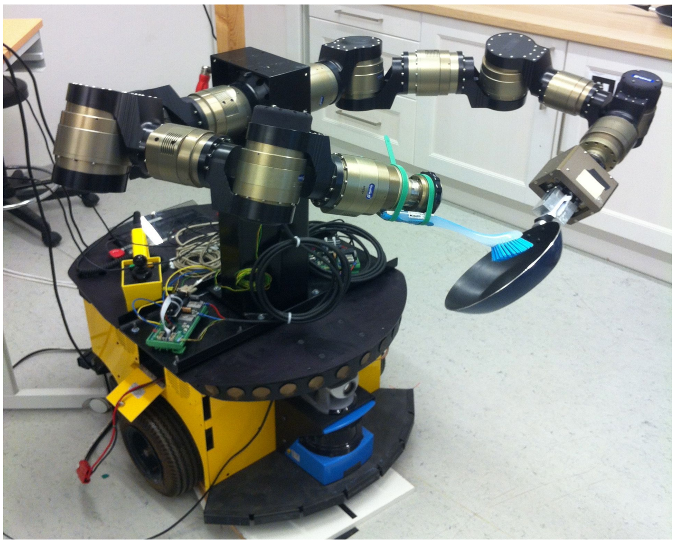

Fig. 1. The dual arm robot performing the task of cleaning a frying pan.

- While a human is lifting one side of a table the robot manipulates the other side with two grippers.
- Holding a workpiece with one hand and using some related tool with the other hand, e.g. manually cleaning a small object.

While performing bi-manual tasks, the robot need to simultaneously take the following secondary constraints into consideration: avoid internal collisions, avoid external collisions, avoid singularities, and finally keep the robot arms and the manipulated objects in the camera field of view.

Constraint based programming for robot motion generation has received a lot of attention, as it enables the execution of highly complex robot tasks. Constraint based

⋆ This work has been supported by the Swedish Research Council (VR), the Swedish Foundation for Strategic Research (SSF), and the European Union FP7 project RoboHow.Cog (FP7-ICT-288533).

programming work can be found in the early publications by Samson et al. [1991], Seraji [1989], Peng and Adachi [1993], while recently the framework iTaSC (Instantaneous Task Specification using Constraints) by De Schutter et al. [2007], Decré et al. [2009], Smits et al. [2009] has built upon it. 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 will present a new variation on constraint based programming, and apply it to do online dual arm manipulation.

The main contribution of the present work is the theoretical extension of margin based constraint based programming using inequality constraints, to also include time dependent equality constraints in a compact and uniform way. In order to demonstrate the applicability of the proposed approach to a dual arm problem, we take a bi-manual dish washing task as a proof of concept example.

We model the dish washing task with specified contact force with a set of time dependent equality constraints and one more inequality constraint. The secondary constraints are specified with another set of inequality constraints. We treat these constraints with the proposed method and the result is verified both in simulation and on a physical robot platform.

The outline of the paper is as follows. In Section 2 we relate the proposed approach to the state of the art. In Section 3 we describe the proposed version of constraint based programming and contrast it to the state of the art. Section 4 then formalizes the dual arm manipulation problem. The proposed solution is presented in Section 5. The simulation validation is followed by additional experiments on a real robot in Section 6. Finally, we conclude the paper in Section 7.

2. RELATED WORK

In general, the approaches to generating motion of redundant manipulators can be divided into two categories, global (offline) and local (online). In the offline approaches, we plan joint space trajectories such that we meet the desired objectives while fulfilling other constraints. As reported by Patel et al. [2005], these methods are computationally expensive and often require the kind of structured environments that can be found in factories, but also generate very efficient solutions, e.g., by solving minimum time problems.

Online approaches on the other hand, need to be less computationally expensive to meet real time requirements. They can handle unstructured environments with moving obstacles, such as domestic environments with humans nearby, but might produce less efficient solutions and might even fail to solve problems in really difficult cases, such as maze-like environments.

These complementary qualities may be best exploited by combining algorithms of both types. An offline algorithm may generate high-level plans and trajectories, that are then carried out by an online algorithm that performs the desired set of subtasks while at the same time ensuring that the secondary constraints are satisfied. In this paper, we put the focus on the online (local) approach. We propose a controller that can perform the assigned tasks in an efficient manner while satisfying the desired constraints. Note that the *generation* of such assigned tasks or constraints is not treated in the current work, one could specify task constraints, e.g. using the *iTask* approach described by De Schutter et al. [2007], Smits et al. [2009].

The proposed approach uses constraint based programming, which has its roots in the concepts of the *additional tasks* by Seraji [1989], the *user defined objective functions* by Peng and Adachi [1993], and the *sub-tasks* by Tatlicioglu et al. [2008]. Similar ideas were used in the *Stack of Tasks* approach by Mansard and Chaumette [2007], Mansard et al. [2009], the *iTask* approach by De Schutter et al. [2007], Smits et al. [2009], and in a variation using Quadratic programming that was proposed by Zhang et al. [2004], Zhang and Mai [2007].

However, as the approaches listed above assume that the main objective is defined with respect to the desired, possibly time varying, position and orientation of the end effector, they address all the secondary tasks using the so-called self motion, in the orthogonal space of the end-effector Jacobian.

By introducing scalar inequalities and bound margins as reported by Ögren [2008], Ögren and Robinson [2011], Ögren et al. [2012], our work can address additional tasks that are not limited to the null space of the Jacobian, and is also not limited to a number of additional constraints equal to the degree of redundancy, since inequality constraints do not reduce the dimensionality of the feasible set in joint space. The concept of using inequalities in this type of constraint based programming was earlier applied to dual arm manipulation by Ögren et al. [2012], mobile robot obstacle avoidance by Ögren [2008] and surveillance UAV control by Ögren and Robinson [2011].

Much of the work in this paper, and in Ögren [2008], Ögren and Robinson [2011], Ögren et al. [2012] is related to the strong contributions reported in Kanoun et al. [2009], Kanoun [2012]. It should however be noted that the work has been developed independently, as for example one of the major topics of Kanoun [2012], adding inequalities, was described in Ögren [2008]. This work however, goes beyond Kanoun [2012] in that we add exact tracking of timed trajectories, force feedback constraints, as well as experimental validation of the whole framework. Furthermore, different from using the *active set method* by Kanoun [2012] or iterating through inequality constraint one by one such as Kanoun et al. [2009], we solve one QP once for all. In order to ensure the solvability of this QP and prioritize different constraints, we weight the slack variables which are assigned to each constraint and minimize their weighted sum in the objective.

3. A NEW VARIATION ON CONSTRAINT BASED PROGRAMMING

In this section we will describe the proposed version of constraint based programming using bound margins. The inequalities part of the proposed approach was described in detail by Ögren [2008], Ögren and Robinson [2011]. Here we will adopt the basic ideas, as presented below, and add equality constraints to the formulation. Typographically the change is small, as the equalities are handled in a way analogous to the inequalities. However, this compact margin based formulation gives significant improvements to task execution as illustrated in the experiments of Section 6 since it includes variable softness of the bound for the inequalities, variable gains on the feedback part of the equalities, as well as feedforward to account for explicit time dependencies in the constraints.

Problem 1. Given a time interval $[t_0, t_f]$, initial state $q(t_0) = q_0$ and a control system

$$\dot{q} = h(q, u),$$

where $q \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$. Let us formulate the control objective in terms of a set of functions $f_i : \mathbb{R}^n \to \mathbb{R}$ and bounds $b_i \in \mathbb{R}$, $i \in I \subset \mathbb{N}$ as follows

$$\min_{u(\cdot)} \; f_j(q(t_f), t_f), \; j \in I \tag{1}$$

$$\text{(s.t.)} \; f_i(q(t), t) \le b_i, \; \forall i \in I_{ie}, \; t > t_0 \tag{2}$$

$$f_i(q(t), t) = b_i, \; \forall i \in I_e, \; t > t_0 \tag{3}$$

where we assume that the constraints are satisfied at t_0, i.e. $f_i(q_0, t_0) \le b_i$ for all $i \in I_{ie}$ and $f_i(q_0, t_0) = b_i$ for all $i \in I_e$ and $I_{ie}, I_e \subset I$.

Now, assuming that Problem 1 above is not solvable, either due to uncertainties or in lack of computational resources, we instead resort to the following online (local) controller to generate a new control at each time step

Problem 2.

$$\min_{u} \; \dot{f}_j(q(t), u, t) + u^T Q u, \; j \in I \tag{4}$$

$$\text{(s.t.)} \; \dot{f}_i(q, u, t) \le -k_i(f_i(q, t) - b_i), \; \forall i \in I_{ie}, \tag{5}$$

$$\dot{f}_i(q, u, t) = -k_i(f_i(q, t) - b_i), \; \forall i \in I_e, \tag{6}$$

where k_i are positive scalars and Q is a positive definite matrix.

First we look at the inequalities. It is trivial to prove that Equation (2) is fulfilled for $t > t_0$ as long as Equation (5) is satisfied. Furthermore, in the worst case, if we have equality in Equation (5) then the bounds of Equation (2) will be exponentially approached, but not violated, with time constant $1/k_i$, as presented by Ögren and Robinson [2011]. Note that the bound will *only* be approached if motion in that direction corresponds to an improvement in the objective function, or is needed with respect to some other constraint.

Looking at the equalities, we also see that as long as Equations (6) are satisfied, so will (3), for $t > t_0$. Furthermore, if we have an error in the desired equality (3), then (6) will drive that error down to zero exponentially, with time constant $1/k_i$. We will see below, in equation (17), that a feedforward term taking care of explicit time dependencies is also incorporated in (6).

Then in the objective function, we know that (1) is kept small as long as its derivative $\dot{f}_j(q(t), u, t), j \in I$ is minimized. We smooth the input u by adding a quadratic regularization term $u^T Q u$ in (4), where Q is a diagonal positive-definite matrix designed to weight elements in u.

Finally, we note that to account for changes in the index j and different bounds b_i in different parts of the state space, as well as handling the situation where the constraints are contradictory and thus impossible to satisfy simultaneously, one could apply the somewhat more complex prioritized designs of the bounds b_i presented by Ögren [2008], Ögren and Robinson [2011].

We conclude this section by noting that the use of optimization formulations in constraint based programming is well known as stated by Zhang et al. [2004], Zhang and Mai [2007], Decré et al. [2009]. The contributions of our work lies in the design of (4), (5), (6), that allow e.g., tracking of timed trajectories, adding force feedback to the framework, and the experimental validation of the approach.

4. PROBLEM FORMULATION

We first state a general dual arm manipulation (DAM) problem and subsequently define the constraints so that they can be directly embedded in the constraint based programming approach described in Section 3.

Problem 3. (DAM problem). The DAM problem consists of the following objectives and constraints.

- Satisfy the main constraint (objective) posed on the (relative) end-effector positions defined in workspace.
- Avoid singular joint positions.
- Avoid collisions.

We now formalize the problem described above. We assume that the low level controllers (e.g. PID controllers) take care of the system dynamics, then we write down the *kinematic* equations of motion for our dual arm manipulators shown in Figure 1 as

$$\dot{q}_j = u_j, \; j \in \{0, \dots 13\} \tag{7}$$

with $|u_j| \le U_j, \; |q_j| \le Q_j$, where q_j are the joint angles, u_j are the joint velocities and U_j and Q_j are bounds on velocities and angles respectively. Thus, as each arm manipulator has 7 joints, the combined state space of the robot is a subset of $\mathbb{R}^{14}$.

As an example, we now consider the main objective of manually washing a frying pan. To apply the approach described in Section 3 above, we now write the objectives in terms of scalar functions $f_i(q)$, $f_i : \mathbb{R}^{14} \to \mathbb{R}$ and then formulate the constraints with inequalities and equalities using these functions, $f_i(q) \le b_i$ and $f_i(q) = b_i$. We define the constraints on tool positions and orientations regarding the main DAM task as follows:

$$\begin{aligned} f_{1-3}(q) &= p_1(q) - p_2(q) - d(t, x_1, y_1, z_1, x_2) \\ &= 0 \\ f_4(q) &= x_1^T x_2 \le b_4, \end{aligned} \tag{8}$$

where axis $x_i, y_i, z_i \in \mathbb{R}^3$ are columns of $R_i \in SO(3)$, $p_1 \in \mathbb{R}^3$ is the center of the frying pan and $p_2 \in \mathbb{R}^3$ corresponds to the tip position of the cleaning utensil, $d(t, x_1, y_1, z_1, x_2)$ is an relative offset between p_1 and p_2, which determines what part of the frying pan is to be cleaned, see Figure 2 and below,

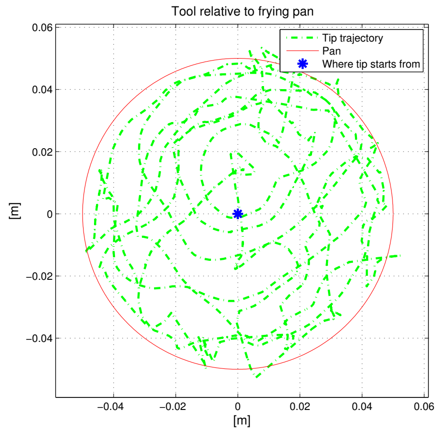

Fig. 2. A reactive pan cleaning tool trajectory $d(t, x_1, y_1, z_1, x_2)$, as defined in the frying pan coordinate frame. This kind of trajectory would be used to actually clean a pan in a reactive way. But to verify the performance of the control system, we input circular tool trajectories, as shown in Figures 4 and 6.

Note that if $f_{1-3} = 0$, the tip of the cleaning utensil p_2 touching the frying pan at $p_1 - d$, and $f_4 \le b_1 < 0$ means the cleaning utensil orientation being opposite to the normal of the frying pan and lying inside a cone around that normal (x_2 being the orientation axis of the tip of the cleaning utensil and x_1 being the surface normal of the frying pan) but both tools having arbitrary rotation around those axes.

The offset $d(t, x_1, y_1, z_1, x_2)$ could be given in real-time by some online function tracking the dirt on the frying pan, and a reactive coverage algorithm deciding where to clean next. Such an example is shown in Figure 2 following the algorithm proposed by Hussein and Stipanovic [2006]. Here however, we isolate the control problem by substituting this online signal with a circular motion with radius D in the frying pan, $d(t, x_1, y_1, z_1, x_2) = D(y_1 \cos(Lt) + z_1 \sin(Lt)) - 2D(z_1 + x_2)$, where the axis x_1, y_1, z_1 are defined above. The first part of the offset accounts for the circular motion of radius D, while the second part accounts for the frying pan handle $2Dz_1$ and the cleaning utensil handle $2Dx_2$.

We will now describe how equation (8) can be modified if we want to add force feedback to the system to ensure contact between the pan and the tool, in the presence of uncertainties regarding the grasps of the objects or the objects themselves. Let f_c be equal to the measured force $F \in \mathbb{R}^3$ projected on the pan normal direction $N \in \mathbb{R}^3$, that is $f_c = N^T F$. Suppose a nominal contact force f_c^0 is given, and let $\delta f_c = f_c^0 - f_c$. Then we can rewrite (8) as follows:

$$\begin{aligned} f_{1-3}(q) &= p_1(q) - p_2(q) - d(t, x_1, y_1, z_1, x_2) - f_F(\delta f_c)N \\ &= 0 \qquad\qquad (9) \\ f_4(q) &= x_1^T x_2 \le b_4, \end{aligned}$$

where $f_F(\cdot)$ is a contact force controller, either a P, a PI or a PID-controller, see Remark 1 below. When $f_{1-3} = 0$ we might have both the cleaning utensil tip p_2 touching the frying pan at $p_1 - d$ and the contact force $f_c^0 = f_c$, or, if modeling errors are present in the pan pose estimate, the force constraint might have forced the tool to move further in the direction of the pan to establish a contact force at its real position.

Remark 1. The contact force term $f_F(\delta f_c)N$ in (9) is orthogonal to the offset $d(t, x_1, y_1, z_1, x_2)$, and it moves the two manipulators inwards along the pan normal direction N. If the desired contact force f_c^0 is reached (9) regresses back to (8) as $f_F(f_c^0 - f_c) = 0$. The contact force controller f_F in (9) could be either a P or PI controller. If we want a damping term, we could let f_F act upon the relative velocity measurement along the contact direction.

Furthermore, we define inequality constraints for singularity and obstacle avoidance. Avoiding singular joint positions can be expressed in terms of a manipulability index, such as the one defined in Siciliano et al. [2009]:

$$f_{5i}(q) = \frac{-1}{2}\det(J_i^T J_i) \le b_5 < 0, \quad i \in \{1, 2\} \qquad (10)$$

where $J_i \triangleq \begin{bmatrix} J_{pi}^T & J_{\omega i}^T \end{bmatrix}^T$ is the manipulator Jacobian which consists of Jacobians J_{pi} and $J_{\omega i}$ related to the translational and rotational motion of the end-effector respectively. Avoiding obstacles can be formulated in terms of the minimal relative distance as

$$f_6(q) = -\min_{x_r \in X_r, x_o \in X_o} ||x_r - x_o|| \le b_6 < 0, \qquad (11)$$

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 required accuracy, 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 Patel et al. [2005].

It is clear that the constraints in equations (9)-(11) have the proper form so that Problem 3 can be formalized as Problem 1.

5. PROPOSED SOLUTION

Following the ideas described in Section 3, instead of solving Problem 1 – for the constraints (9)-(11) – to optimality, we aim to find a feasible *good enough* solution by solving Problem 2 for the aforementioned constraints. In the spirit of Zhang et al. [2004], Zhang and Mai [2007], we note that the above problem is in fact a Quadratic Programming (QP) problem as is stated in the following Lemma.

Lemma 1. Problem 2 is equivalent to the following QP

$$\min_u \; c^T u + u^T Q u \qquad (12)$$

$$\text{s.t. } A_{ie} u \le k_{ie}(b_{ie} - f_{ie}) - h_{ie} \qquad (13)$$

$$A_e u = k_e(b_e - f_e) - h_e \qquad (14)$$

where $c = \frac{df_{\hat{\jmath}}}{dq}$, and each row of $A_{ie}, A_e, b_{ie}, b_e, f_{ie}, f_e, h_{ie}, h_e$ contains the corresponding parts of $\frac{df_i}{dq}, b_i, f_i, \frac{\partial f_i}{\partial t}$ respectively.

Remark 2. Note that there are very efficient ways of solving such QPs, even in quite high dimensions. Note also that if $Q = 0$ we have a Linear Programming problem (LP) which is also easily solvable. However LPs have optimal solutions at the bound of the feasible set, which often leads to control signal chattering, a phenomenon that is removed by adding a small quadratic cost Q on the controls.

proof 1. We will now rewrite the controller with the equations of motion (7). If we differentiate the scalar functions we have

$$\dot{f}_i(q(t), u, t) = \frac{df_i}{dt} = \frac{\partial f_i}{\partial q}^T \frac{dq}{dt} + \frac{\partial f_i}{\partial t} = \frac{\partial f_i}{\partial q}^T u + \frac{\partial f_i}{\partial t}$$

which implies the following structure of the controller

$$\min_u \ \frac{\partial f_j}{\partial q}^T u + u^T Q u, \ j \in I \tag{15}$$

$$\text{s.t.} \ \frac{\partial f_i}{\partial q}^T u \leq k_i(b_i - f_i) - \frac{\partial f_i}{\partial t}, \ \forall i \in I_{ie} \tag{16}$$

$$\frac{\partial f_i}{\partial q}^T u = k_i(b_i - f_i) - \frac{\partial f_i}{\partial t}, \ \forall i \in I_e \tag{17}$$

which is equivalent to a Quadratic Programming problem in equations (12)-(14).

We furthermore note that the values of $\frac{df_i}{dq}$ can be given either in closed form, or as numerical estimates. For instance, f_{1-3} in Equation (9) can be analytically differentiated to give

$$\begin{aligned}
\frac{\partial f_{1-3}}{\partial q} =& J_{p1} - J_{p2} \\
& + D(S(y_1)J_{\omega 1}\cos Lt + S(z_1)J_{\omega 1}\sin Lt) \\
& - 2D(S(z_1)J_{\omega 1} + S(x_2)J_{\omega 2}) \\
& - f_F(\delta f_c)S(N_1)J_{\omega 1} \\
& - f_F(\delta f_c)S(N_2)J_{\omega 2} \\
\frac{\partial f_4}{\partial q} =& x_2^T(-S(x_1)J_{\omega 1}) + x_1^T(-S(x_2)J_{\omega 2}). \\
\frac{\partial f_{1-3}}{\partial t} =& KL(y_1 \sin Lt - z_1 \cos Lt)
\end{aligned} \tag{18}$$

where $S(a)$ is a skew-symmetric matrix which is used in order to produce the outer product of a vector $a \in \mathbb{R}^3$ with some vector $b \in \mathbb{R}^3$ i.e. $S(a)b = a \times b$. Note that N_1 and N_1 in (18) are defined in frame of arm 1 and 2 respectively.

To conclude, we thus propose to provide good enough, feasible, solutions to Problem 1 for the constraints (9)-(11) by iteratively solving the QP in Equations (12)-(14).

Below, we will see how the proposed approach performs in both simulation and experiments.

6. EXPERIMENTS AND SIMULATIONS

In this section we present the results of both Matlab simulations and real experiments conducted on the dual arm robot in Figure 1. We first compare the theoretical noise free (simulation) performance and the real hardware performance through a contact force free bi-manual pan cleaning task. Then we extend the hardware experiment by adding force feedback, compensating for the geometric model imperfections.

6.1 Simulation

The simulations aim to validate both the feasibility of the proposed solution and its ability to generate solutions with different convergence speed. We choose to use the Matlab Robotics Toolbox by Corke [1996] and two Puma 560 robots sharing the same workspace, see Figure 3. Even though the Puma 560 robot has only 6 DoFs, instead of 7 of the robot shown in Figure 1, there is still redundancy to be exploited in our sample bi-manual task.

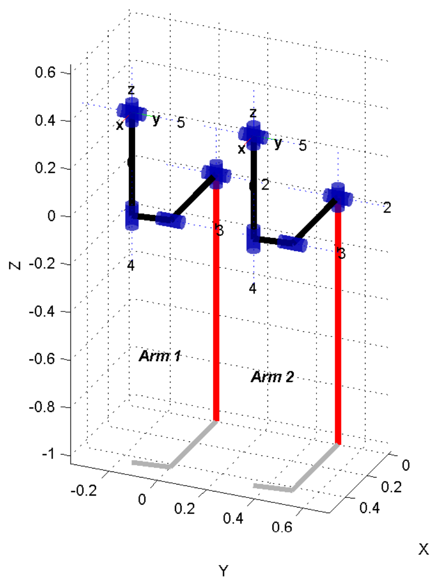

Fig. 3. We choose to simulate the dual arm manipulators with two Puma 560 manipulators sharing the same workspace.

In all examples, we let $j = 1$, $I = \{1, 2, 3, 4, 5, 6\}$, $I_e = \{1, 2, 3\}$ and $I_{ie} = \{4, 5, 6\}$. The two manipulators need to avoid a table surface type of obstacle located at height $z = -0.6$, and for simplicity we only check the minimal distance to the obstacle with the end-effector positions, i.e., $X_r = \{p_1, p_2\}$ and $X_o = \{x \in \mathbb{R}^3 : x_z = -0.6\}$, in Equation (11). Due to the lack of simulated force we apply (8) instead of (9) in the simulations. We present three different simulations in Figures 4 and 5. The difference between the three simulations is the value of the parameters k_i.

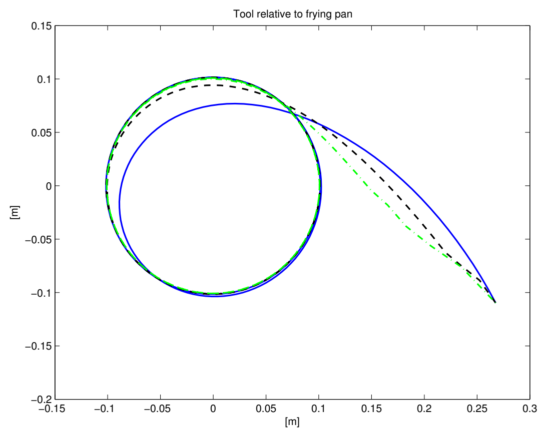

Fig. 4. The simulated motion of the cleaning utensil in the frying pan, projected onto the frying pan plane. (blue solid: $k = 0.2$, black dashed: $k = 0.5$ and green dash-dotted: $k = 1$.)

In the first simulation (green dash-dotted) we have used $k_i = 1, \forall i$ in the second (black dashed) $k_i = 0.5, \forall i$ and in the third (blue solid) $k_i = 0.2, \forall i$. The resulting motion, in all three examples, of the cleaning utensil tip relative to the frying pan is shown in Figure 4. As can be seen, due to the use of different parameters the cleaning utensil approaches the pan at different speed but traces out the same circular pattern defined by the offset d in Equation (8). The functions f_i are shown in Figure 5 (blue solid). The first plot shows tool position deviation $||f_{1-3}||$, and we can see that it converges exponentially. Then we have the tool angle f_4, which starts outside of its bound, but then approaches it and stays below it for the rest of the simulation. The singularity objectives f_{51}, f_{52} are then shown and finally the obstacle avoidance measures f_6 of each arm separately.

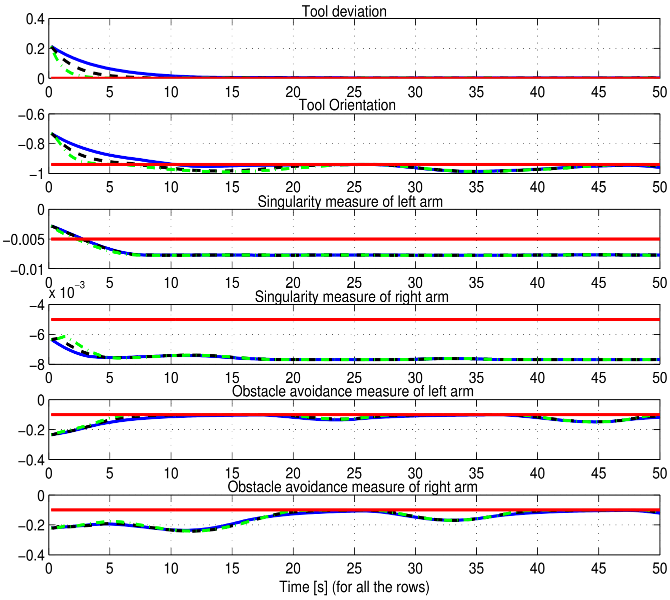

Fig. 5. Time evolution of the functions $f_i(q)$ of the simulation. Note how the scalar inequalities $f_i \leq b_i$ are satisfied for the different settings (blue solid: $k = 0.2$), (black dashed: $k = 0.5$), and (green dash-dotted: $k = 1$). The red line is the corresponding bound b_i.

The difference between the three simulations is the value of the parameters k_i. As noted in Section 3, this parameter governs how fast inequality bounds are allowed to be approached, i.e., the softness of the bound, and how fast errors in equality bounds should be decreased, i.e., the proportional gain in the feedback. The difference regarding equalities can be seen in Figure 4 and the first plot of Figure 5. The difference regarding inequalities can be seen in the second and fifth plots of Figure 5. If the inequality is not satisfied, the error is decreased exponentially, and when it is satisfied, exponential convergence is a bound for how fast it *can* be approached. Note however that inequality bounds are only approached when it is needed with respect to other constraints, or the objective function. Otherwise the system keeps executing without getting near the bound, as can be seen in plots three and four of Figure 5.

6.2 Hardware experiment without force control

Running the algorithm simulated in the above section on the dual arm manipulator of Figure 1, we got the data of Figures 6-8. All parameters were the same, except that we only used one value of the parameter $k_i = 0.6$, and moved along the circular path at a higher velocity.

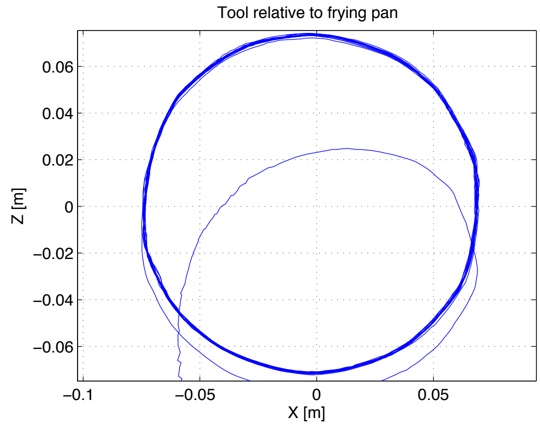

Fig. 6. The trajectory of the hardware experiment, as defined in the frying pan coordinate frame.

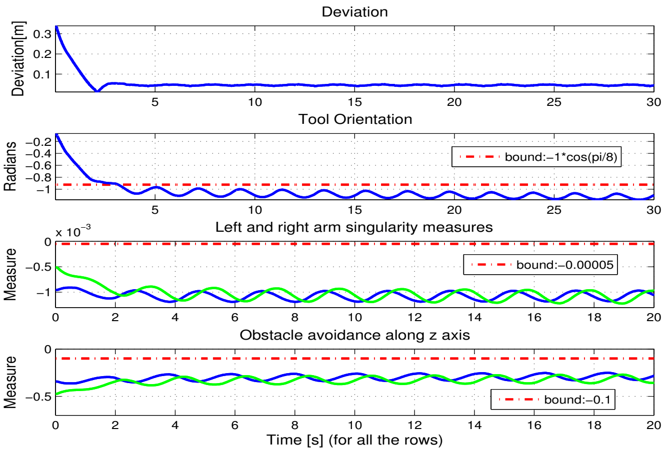

Fig. 7. Time evolution of the functions $f_i(q)$ of the hardware experiment.

The tool converges to the circular pattern in Figure 6, and the measures of Figure 7 exponentially move towards their corresponding values, or stay below their bounds. Figure 8 shows joint velocities, which can be seen to be fairly smooth, with all joint contributing to the motion. It is clear that the performance of the hardware experiment is consistent with the simulations. We also made an experimental validation of the inequality-only formulation of the problem reported by Ögren et al. [2012], where the equality constraints $f_i = b_i, \ i = 1 \dots 3$ were taken into account through the inequality constraint $||(f_1 - b_1, f_2 - b_2, f_3 - b_3)|| \leq \epsilon$ for some small ϵ. The simulation of the method by Ögren et al. [2012] show small oscillations in the tool trajectory. These turned out to be significantly amplified when run on the real hardware, as can be seen in Figure 9. Comparing with Figure 6, we see how the contributions of this paper have a major impact on performance.

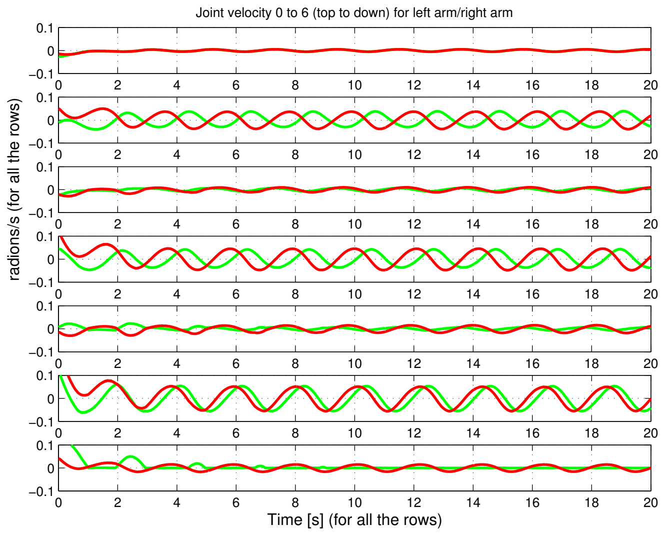

Fig. 8. All 14 joint velocities as a function of time for the hardware experiment.

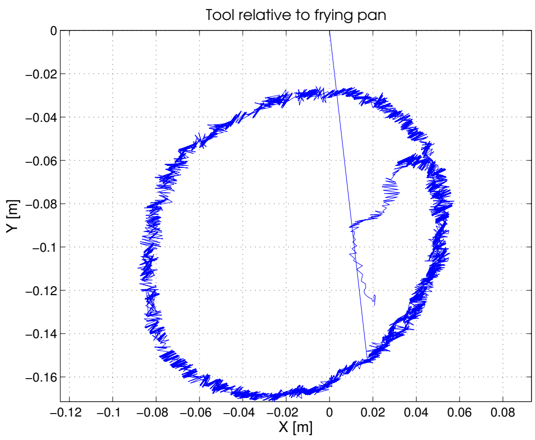

Fig. 9. Using the approach of Ögren et al. [2012], the performance is much worse than in Figure 6.

6.3 Hardware experiment with force control

We add a contact force control to the same algorithm by using (9) rather than (8) and change the parameters to achieve even faster constraint convergence [1] . The reference offset $d(t, x_1, y_1, z_1, x_2)$ used in (8) and (9) includes some modeling errors since it is defined with respect to the geometrical models of the pan and utensil. We compensate this model imperfection by ensuring the contact between the pan and the utensil with contact force control [2] . The tool tip position is estimated by the end effector position without accounting for the tool deformation. This reason, together with the aforementioned model imperfection lead to an imperfect circular cleaning trajectory in hardware experiment with guaranteed contact as shown in Figure 10.

The measures in Figure 11 have comparable performance as the corresponding measures in Figure 7 except that the

[1] $k_1 = 2.0, k_2 = 2.0, k_3 = 0.8, k_4 = 2.0$.

[2] We can verify the contact through the force curve plotted in the last row of Figure 11. The oscillation of the force curve is due to the force torque sensor noise whose standard deviation is about 1 N.

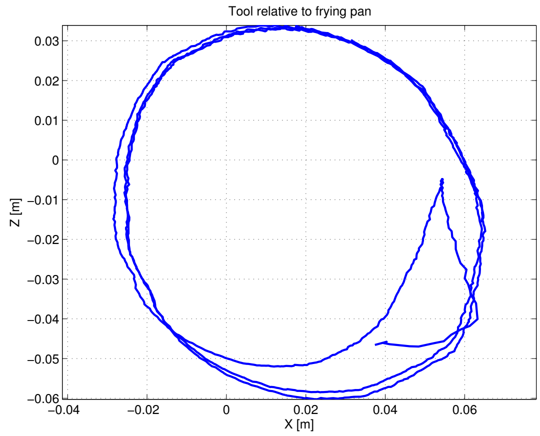

Fig. 10. The trajectory of the hardware experiment with contact force, as defined in the frying pan coordinate frame.

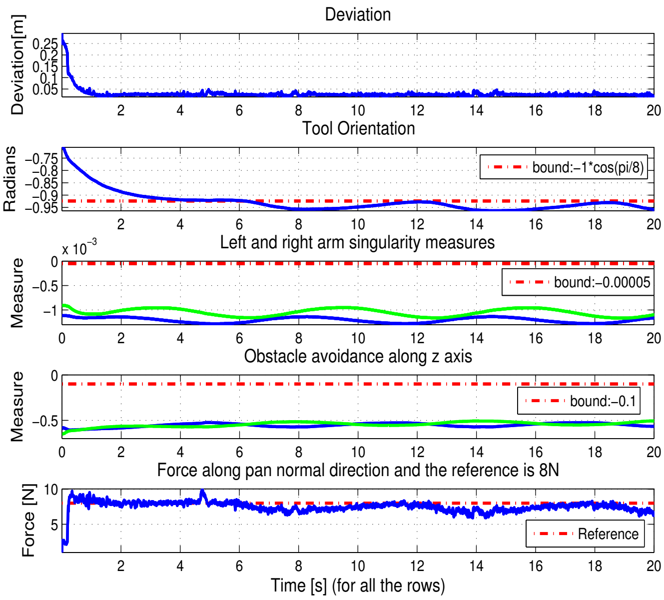

Fig. 11. Time evolution of the functions $f_i(q)$ of the hardware experiment with force control.

variations of the tool orientation angle from $6-10$s and $13-16$s are correlated with the variations of the contact force curve from $6-10$s and $13-16$s. This is because the cleaning utensil, which is a brush, has different stiffness in different directions. To fix the variations, we would need to apply an adaptive force controller. Otherwise we note that the force touching performance is similar to what we can expect from this hardware [3] , see Smith and Karayiannidis [2012].

6.4 Discussions

We note that in the contact force free case the hardware (hw) performance is quite similar to the simulation (sim). The trajectories of the tool in the pan frame, shown in Figures 4 (sim) and 6 (hw) are similar, with the latter

[3] The maximum execution frequency of the hardware experiment is about 35Hz, due to the need of synchronization of the joint position signals from 14 DoFs distributed along the two manipulators.

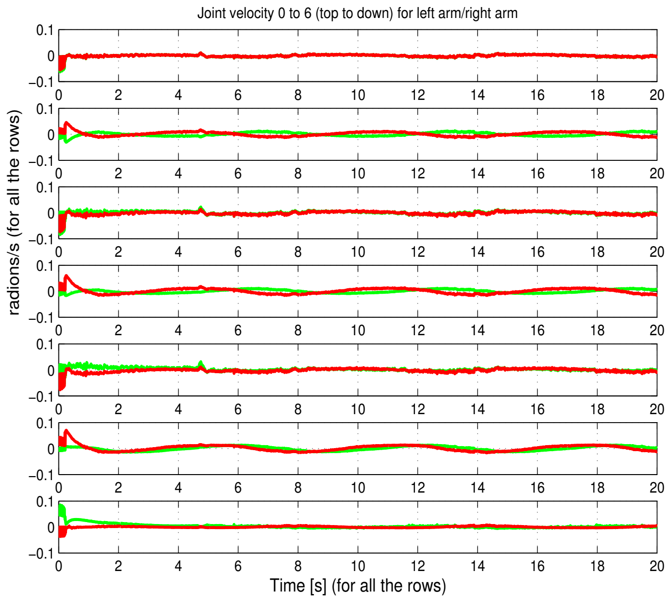

Fig. 12. All 14 joint velocities as a function of time for the hardware experiment with force control.

being somewhat less smooth. The same holds for the functions f_i in Figures 5 (sim) and 7 (hw). The use of the contact force feedback compensates for the modeling errors involved in the constraint, which makes the tool trajectory shown in Figure 10 deviate from the perfect circle shown in Figure 6. However it gives us guaranteed contact as indicated by the force curve shown in the fifth plot of Figure 11.

7. CONCLUSIONS

In this paper, we have presented a constraint based programming approach to generate online motion plans for a redundant robot manipulator. The novelty of the approach is the unified treatment of equality and inequality type constraints based on bound margins. The resulting task is formulated as a quadratic programming optimization problem. Performance is verified in experiments and simulations of a dual arm robot performing a pan cleaning task.

REFERENCES

Richard Bloss. Robotics innovations at the 2009 assembly technology expo. *Industrial Robot: An International Journal*, 37(5):427–430, 2010.

M. Buss, M. Kuschel, K.K. Lee, A. Peer, B. Stanczyk, and U. Unterhinninghofen. High fidelity telepresence systems: Design, control, and evaluation. In *Joint International COE/HAM SFB-453 Workshop on Human Adaptive Mechatronics and High-Fidelity Telepresence*, Tokyo, Japan, 2006.

F. Caccavale, P. Chiacchio, and S. Chiaverini. Task-space regulation of cooperative manipulators. *Automatica*, 36 (6):879 – 887, 2000.

Pasquale Chiacchio and Stefano Chiaverini. *Complex Robotic Systems*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edition, 1998. ISBN 3540762655.

P.I. Corke. A robotics toolbox for matlab. *Robotics & Automation Magazine, IEEE*, 3(1):24–32, 1996.

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*, 26(5):433–455, 2007.

C.C. de Wit, G. Bastin, and B. Siciliano. *Theory of robot control*. Springer-Verlag New York, Inc. Secaucus, NJ, USA, 1996.

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)*, pages 964–971. IEEE, 2009.

M. Fuchs, C. Borst, P.R. Giordano, A. Baumann, E. Kraemer, J. Langwald, R. Gruber, N. Seitz, G. Plank, K. Kunze, R. Burger, F. Schmidt, T. Wimboeck, and G. Hirzinger. Rollin' justin - design considerations and realization of a mobile platform for a humanoid upper body. In *IEEE International Conference on Robotics and Automation, 2009. ICRA '09.*, pages 4131 –4137, may 2009. doi: 10.1109/ROBOT.2009.5152464.

Islam I Hussein and Dušan M Stipanovic. Effective coverage control using dynamic sensor networks. In *Decision and Control, 2006 45th IEEE Conference on*, pages 2747–2752. IEEE, 2006.

B. M. Jau. Anthropomorphic exoskeleton dual arm/hand telerobot controller. In *International Conference on Intelligent Robots and Systems*, 1988.

Oussama Kanoun. Real-time prioritized kinematic control under inequality constraints for redundant manipulators. *Robotics: Science and Systems VII*, page 145, 2012.

Oussama Kanoun, Florent Lamiraux, Pierre-Brice Wieber, Fumio Kanehiro, Eiichi Yoshida, and Jean-Paul 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*, pages 2939–2944. IEEE, 2009.

C.C. Kemp, A. Edsinger, and E. Torres-Jara. Challenges for robot manipulation in human environments [Grand Challenges of Robotics]. *IEEE Robotics Automation Magazine*, 14(1):20–29, march 2007. ISSN 1070-9932. doi: 10.1109/MRA.2007.339604.

A. Kron and G. Schmidt. Bimanual haptic telepresence technology employed to demining operations. In *Eurohaptics2004*, pages 490–493, Munich, Germany, 2004.

J. Kr uger, G. Schreck, and D. Surdilovic. Dual arm robot for flexible and cooperative assembly. *CIRP Annals - Manufacturing Technology*, 60(1):5–8, 2011. ISSN 0007-8506. doi: DOI: 10.1016/j.cirp.2011.03.017.

Sukhan Lee and Sungbok Kim. A self-reconfigurable dual-arm system. In *IEEE International Conference on Robotics and Automation*, Apr 1991.

N. Mansard and F. Chaumette. Task sequencing for high-level sensor-based control. *Robotics, IEEE Transactions on*, 23(1):60–72, feb. 2007.

N. Mansard, O. Khatib, and A. Khedar. A unified approach to integrate unilateral constraints in the stack of tasks. *IEEE Transaction on Robotics*, 25(3), June 2009.

P. Ögren. Improved predictability of reactive robot control using control lyapunov functions. In *IEEE/RSJ Inter-*

national Conference on Intelligent Robots and Systems (IROS) 2008., pages 1274–1279. IEEE, 2008.

P. Ögren and J.W.C. Robinson. A model based approach to modular multi-objective robot control. *Journal of Intelligent & Robotic Systems*, pages 1–26, 2011.

P. Ögren, C. Smith, Y. Karayiannidis, and D. Kragic. A multi objective control approach to online dual arm manipulation. In *International IFAC Symposium on Robotic Control*, 2012.

RV Patel, F. Shadpey, F. Ranjbaran, and J. Angeles. A collision-avoidance scheme for redundant manipulators: Theory and experiments. *Journal of Robotic Systems*, 22(12):737–757, 2005.

Z.X. Peng and N. Adachi. Compliant motion control of kinematically redundant manipulators. *Robotics and Automation, IEEE Transactions on*, 9(6):831–836, 1993.

C. Samson, M. Le Borgne, and B. Espiau. *Robot control: the task function approach*, volume 22. Clarendon Press Oxford, England, 1991.

Lorenzo Sciavicco. *Robotics: modelling, planning and control*. Springer, 2009.

H. Seraji. Configuration control of redundant manipulators: Theory and implementation. *Robotics and Automation, IEEE Transactions on*, 5(4):472–490, 1989.

B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. Springer-Verlag, 2009.

Christian Smith and Yiannis Karayiannidis. Optimal command ordering for serial link manipulators. 2012.

Christian Smith, Yiannis Karayiannidis, Lazaros Nalpantidis, Xavi Gratal, Peng Qi, Dimos V. Dimarogonas, and Danica Kragic. Dual arm manipulation – a survey. *Robotics and Autonomous Systems*, 60(10):1340–1353, 2012.

R. Smits, T. Laet, K. Claes, H. Bruyninckx, and J. Schutter. itasc: A tool for multi-sensor integration in robot manipulation. *Multisensor Fusion and Integration for Intelligent Systems*, pages 235–254, 2009.

E. Tatlicioglu, D. Braganza, T.C. Burg, and D.M. Dawson. Adaptive control of redundant robot manipulators with sub-task objectives. In *American Control Conference (ACC), 2008*, pages 856–861. IEEE, 2008.

J. Taylor and D. Seward. Control of a dual–arm robotic manipulator. *Nuclear Engineering International*, 55 (673):24–26, 2010.

Woo-Keun Yoon, Yuichi Tsumaki, , and Masaru Uchiyama. An experimental system for dual-arm robot teleoperation in space with concepts of virtual grip and ball. In *International Conference on Advanced Robotics (ICAR99)*, pages 225–230, Tokyo, Japan, 1999.

Yunong Zhang and Shugen Mai. Minimum-energy redundancy resolution of robot manipulators unified by quadratic programming and its online solution. In *Mechatronics and Automation, 2007. ICMA 2007. International Conference on*, pages 3232–3237. IEEE, 2007.

Yunong Zhang, Shuzhi Sam Ge, and Tong Heng Lee. A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 34(5):2126–2132, 2004.